An Introduction to Time Series Analysis

With a Focus on Macroeconomic & Financial Applications

Prof. Mike Aguilar

mike.aguilar@duke.edu
https://www.linkedin.com/in/mike-aguilar-econ
Last Updated: 9Nov2026

Outline

1. Introduction
2. Data Prep
3. Explore
4. Explain
5. Forecast

Learning Objectives

- Understand what is "special" about time series.

Learning Objectives

- Understand what is "special" about time series. As such, I'll be focusing on differences to non time series settings and techniques.

Learning Objectives

- Understand what is "special" about time series. As such, I'll be focusing on differences to non time series settings and techniques.
- Learn how to adapt our typical workflow for time series data

Learning Objectives

- Understand what is "special" about time series. As such, I'll be focusing on differences to non time series settings and techniques.
- Learn how to adapt our typical workflow for time series data
- Introduce new techniques that are not feasible within non time series settings.

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Workshop Structure:

- The workshop consists of two parts

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Workshop Structure:

- The workshop consists of two parts
- Part 1: 2 hour overview of key topics

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Workshop Structure:

- The workshop consists of two parts
- Part 1: 2 hour overview of key topics
  - I will make this slide deck after the seminar (www.market-observatory.com)
  - This slide deck is a reference tool; we will not discuss every slide tonight.

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Workshop Structure:

- The workshop consists of two parts
- Part 1: 2 hour overview of key topics
  - I will make this slide deck after the seminar (www.market-observatory.com)
  - This slide deck is a reference tool; we will not discuss every slide tonight.
- Part 2: Asynchronous coding exercise; I post sample code, which you explore, complete, and execute. I will host office hours to answer questions.

Intended Audience and Seminar Structure

Intended Audience:

- Attendees have a basic understanding of statistical inference
- Attendees have a working knowledge of standard econometric and machine learning techniques

Workshop Structure:

- The workshop consists of two parts
- Part 1: 2 hour overview of key topics
  - I will make this slide deck after the seminar (www.market-observatory.com)
  - This slide deck is a reference tool; we will not discuss every slide tonight.
- Part 2: Asynchronous coding exercise; I post sample code, which you explore, complete, and execute. I will host office hours to answer questions.
- The applications will be cast within a macro and financial context, but the tools and concepts are widely applicable.

Outline

1 Introduction

2 Data Prep

3 Explore

4 Explain

5 Forecast

Taxonomy of Time Series Models

- **Linear / Econometric Models**
  - AR, MA, ARIMA, ADL, ECM
- **State Space Models**
  - Kalman Filter, Structural TS, Deep State Space
- **Machine Learning Models**
  - Tree-based, Neural Networks (LSTM, N-BEATS)
- **Global / Panel Models**
  - DeepAR, hierarchical forecasting
- **Foundation Models**
  - TimeGPT, Chronos

Types of Data

Data Types:

- Cross Sectional $i = 1, \cdots, N$
- Time Series $t = 1, \cdots, T$
- Panel $(N \times T)$

Types of Data

Data Types:

- Cross Sectional $i = 1, \cdots, N$
- Time Series $t = 1, \cdots, T$
- Panel $(N \times T)$

Modeling time series present distinct challenges from Cross Sectional or Panel data.

Types of Data

Data Types:

- Cross Sectional $i = 1, \cdots, N$
- Time Series $t = 1, \cdots, T$
- Panel $(N \times T)$

Modeling time series present distinct challenges from Cross Sectional or Panel data. Most important among those challenges is "temporal persistence".

Why model temporal persistence?

Why model temporal persistence?

1. Theory: Our theory might suggest that the underlying behavior is persistent. For instance, consider housing starts.

Why model temporal persistence?

1. Theory: Our theory might suggest that the underlying behavior is persistent. For instance, consider housing starts. Home builders often develop a neighborhood, sequentially developing house after house after house until the neighborhood is complete.

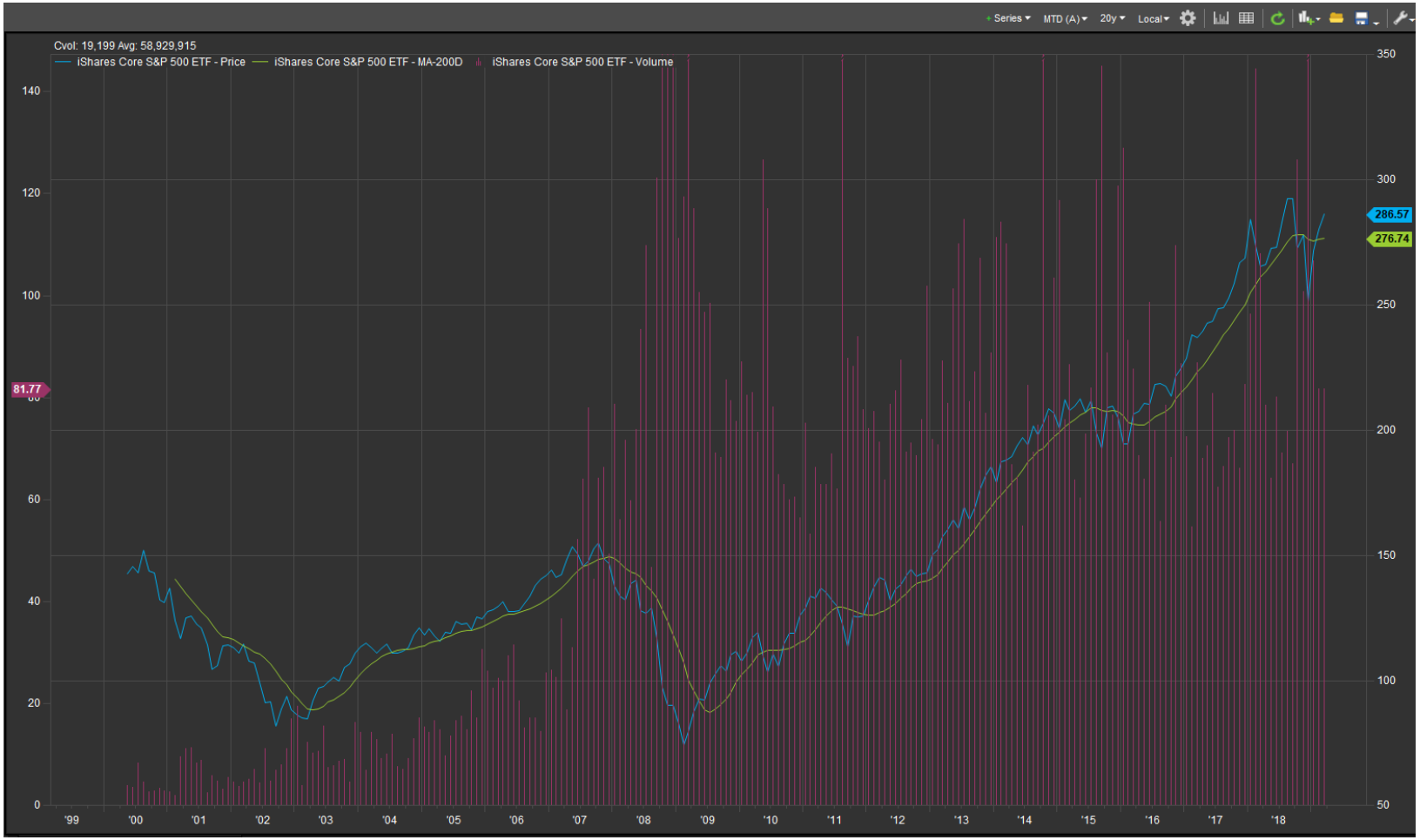
Why model temporal persistence?

1. Theory: Our theory might suggest that the underlying behavior is persistent. For instance, consider housing starts. Home builders often develop a neighborhood, sequentially developing house after house after house until the neighborhood is complete.
2. Error reduction: My error term captures relevant omitted regressors, which I don't observe or know.

Why model temporal persistence?

1. Theory: Our theory might suggest that the underlying behavior is persistent. For instance, consider housing starts. Home builders often develop a neighborhood, sequentially developing house after house after house until the neighborhood is complete.
2. Error reduction: My error term captures relevant omitted regressors, which I don't observe or know. If these omitted regressors are persistent due to the underlying behavior, then including lagged dependent variables reduces the error.

What is so special about "time series" data?



Certain features and stylized facts appear with time series data that are not apparent in cross sectional. (e.g. momentum)

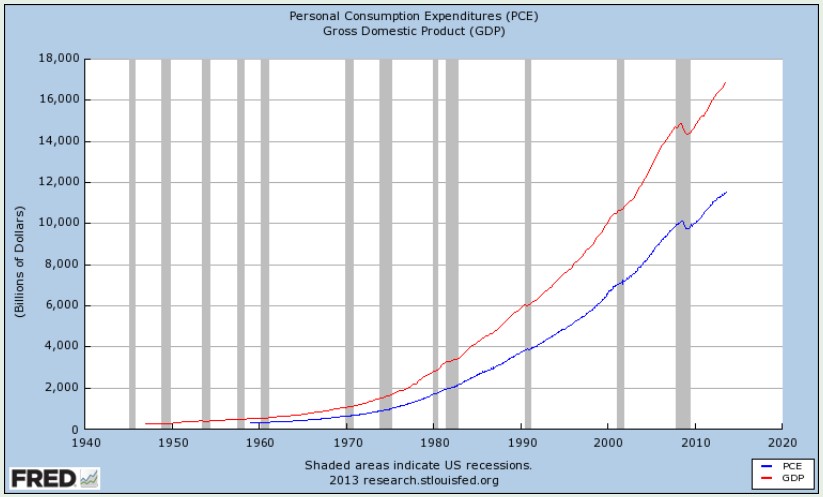
What is so special about "time series" data?

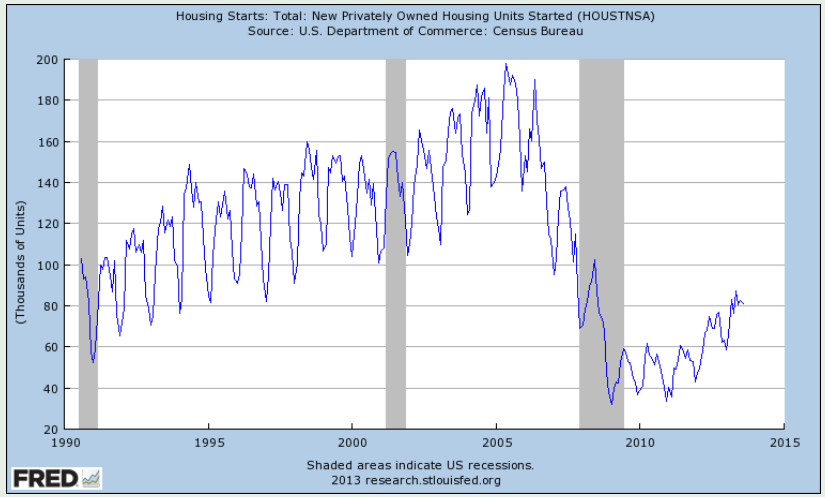
Example (Misleading Time Series Regressions)

Consider the time series models below. In each, discuss why running such a regression by OLS might be misleading.

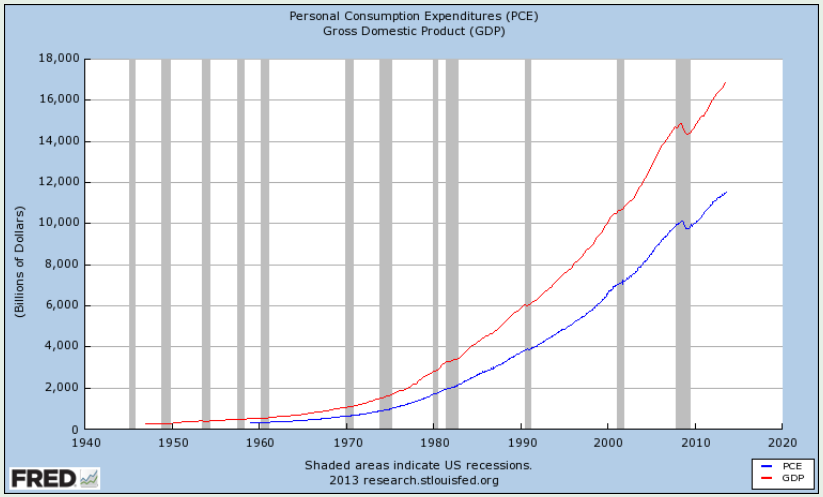
1. $C_t = \beta_0 + \beta_1 Y_t + u_t$
2. $HsngStarts_t = \beta_0 + \beta_1 MrtgRates_t + u_t$

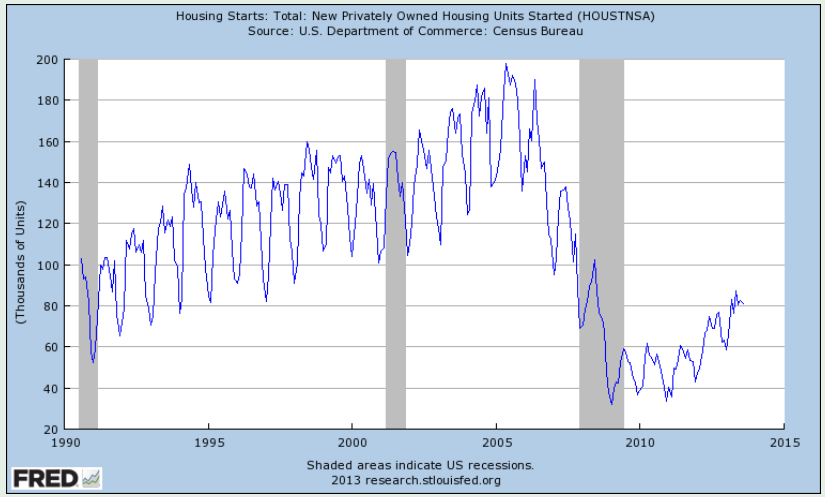
Example (Misleading Time Series Regressions)





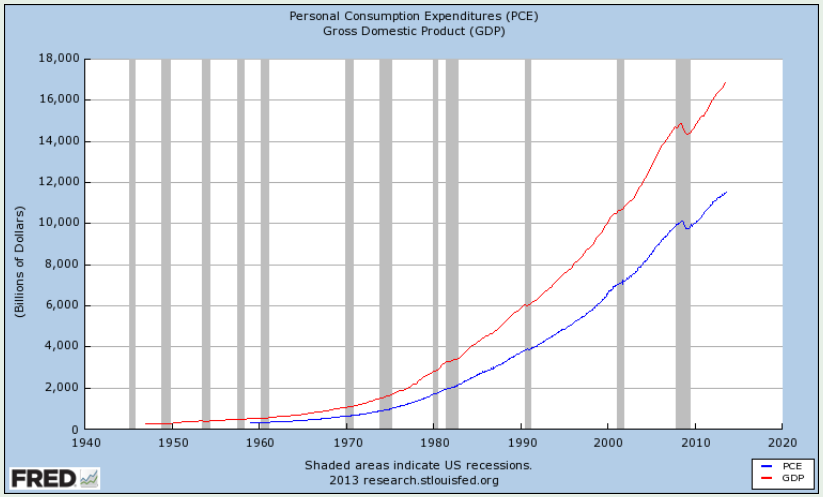
Example (Misleading Time Series Regressions)

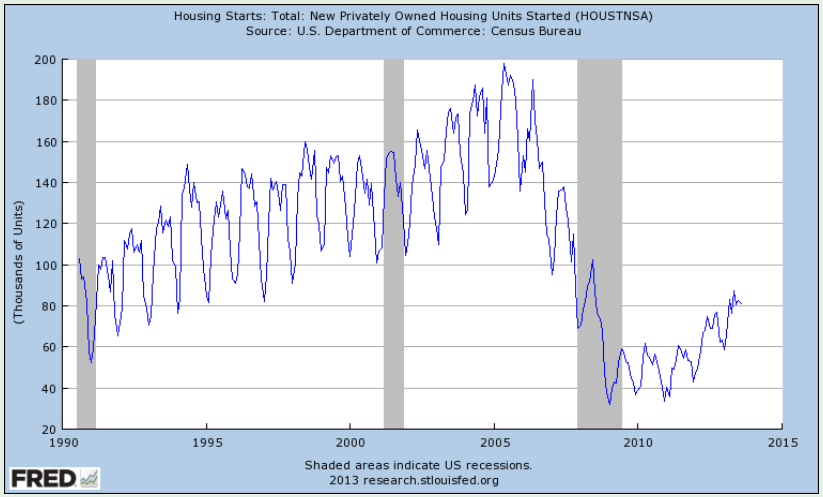




1. (Left) Spurious correlation due to omitted time trend

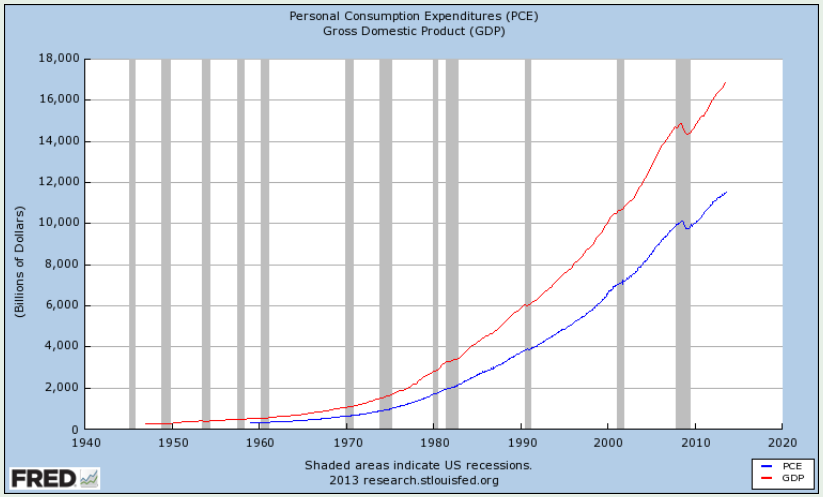
Example (Misleading Time Series Regressions)

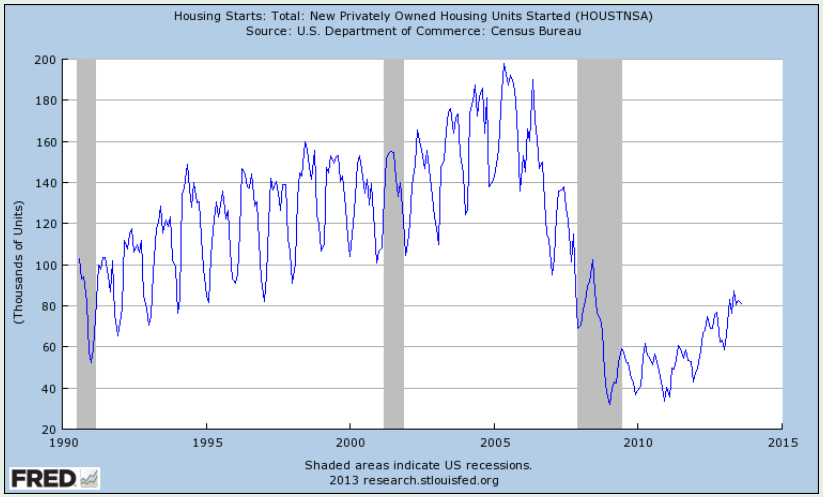




1. (Left) Spurious correlation due to omitted time trend
2. (Right) Seasonal patterns

Example (Misleading Time Series Regressions)





1. (Left) Spurious correlation due to omitted time trend
2. (Right) Seasonal patterns

We need to accommodate for these common features of time series.

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations
- **Trends** and **seasonality**

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations
- **Trends** and **seasonality**
- Risk of **spurious relationships**

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations
- **Trends** and **seasonality**
- Risk of **spurious relationships**

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations
- **Trends** and **seasonality**
- Risk of **spurious relationships**

These features imply:

- Standard cross-sectional tools may fail
- We need models that explicitly account for time

From Intuition to Modeling

Time series data exhibit features that challenge standard approaches:

- **Temporal dependence** across observations
- **Trends** and **seasonality**
- Risk of **spurious relationships**

These features imply:

- Standard cross-sectional tools may fail
- We need models that explicitly account for time

Next: How does the time series workflow adapt to these challenges?

Outline

1. Introduction
2. Data Prep
3. Explore
4. Explain
5. Forecast

Why Data Preparation Matters in Time Series

Time series data requires special care before modeling:

- **Ordering matters**
  - Incorrect ordering destroys temporal dependence
- **Missing data is not innocuous**
  - Dropping or imputing values can distort dynamics
- **Timing must be aligned**
  - Misaligned data introduces lookahead bias

Why Data Preparation Matters in Time Series

Time series data requires special care before modeling:

- **Ordering matters**
  - Incorrect ordering destroys temporal dependence
- **Missing data is not innocuous**
  - Dropping or imputing values can distort dynamics
- **Timing must be aligned**
  - Misaligned data introduces lookahead bias

Poor data preparation leads to misleading inference and invalid forecasts

Sequencing Doesn't Matter for Cross Sectional Data

Sequencing Doesn't Matter for Cross Sectional Data

Company	# Distribution Facilities	Order Fullfillment Time
ABC	10	18
DEF	12	17
GHI	13	17
JKL	15	16
MNO	19	13
PQR	20	13
STU	21	12
VWX	23	11
YZA	29	8

Sequencing Doesn't Matter for Cross Sectional Data

Company	# Distribution Facilities	Order Fullfillment Time
ABC	10	18
DEF	12	17
GHI	13	17
JKL	15	16
MNO	19	13
PQR	20	13
STU	21	12
VWX	23	11
YZA	29	8

Company	# Distribution Facilities	Order Fullfillment Time
MNO	19	13
ABC	10	18
DEF	12	17
VWX	23	11
GHI	13	17
JKL	15	16
YZA	29	8
PQR	20	13
STU	21	12

But Sequencing Does Matter for Time Series

But Sequencing Does Matter for Time Series

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
ABC	1995	6	23
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

But Sequencing Does Matter for Time Series

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
ABC	1995	6	23
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1994	6	22
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1995	6	23
ABC	1998	10	18
ABC	1996	7	21
ABC	1997	8	19

But Sequencing Does Matter for Time Series

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
ABC	1995	6	23
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1994	6	22
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1995	6	23
ABC	1998	10	18
ABC	1996	7	21
ABC	1997	8	19

The temporal dependence among observations requires that we maintain the sequencing of the time series data.

Proper Temporal Ordering

- Convention: Oldest to Youngest

Proper Temporal Ordering

- Convention: Oldest to Youngest
  - Caution: Not all data vendors deliver data in this way

Proper Temporal Ordering

- Convention: Oldest to Youngest
  - Caution: Not all data vendors deliver data in this way
- Pro Tip: Always sort your data before you model

Proper Temporal Ordering

- Convention: Oldest to Youngest
  - Caution: Not all data vendors deliver data in this way
- Pro Tip: Always sort your data before you model
- Best Practice: resort by the date/time vector after each filter/select

Proper Temporal Ordering

- Convention: Oldest to Youngest
  - Caution: Not all data vendors deliver data in this way
- Pro Tip: Always sort your data before you model
- Best Practice: resort by the date/time vector after each filter/select
- For intraday day, be sure you are sorting on the entire date/time stamp, and not just the date/day/year/hr, etc..

Date Time Stamp vs Time Index

Date Time Stamp vs Time Index

Company	Date Time Stamp	Time Index	# Distribution Facilities	Order Fullfillment Time
ABC	1990	1	2	26
ABC	1991	2	2	25
ABC	1992	3	4	23
ABC	1993	4	5	22
ABC	1994	5	6	22
ABC	1995	6	6	23
ABC	1996	7	7	21
ABC	1997	8	8	19
ABC	1998	9	10	18

Date Time Stamp vs Time Index

Company	Date Time Stamp	Time Index	# Distribution Facilities	Order Fullfillment Time
ABC	1990	1	2	26
ABC	1991	2	2	25
ABC	1992	3	4	23
ABC	1993	4	5	22
ABC	1994	5	6	22
ABC	1995	6	6	23
ABC	1996	7	7	21
ABC	1997	8	8	19
ABC	1998	9	10	18

- Using the Time Index is easy to implement and interpret for simple linear time trends on equally spaced data

Date Time Stamp vs Time Index

Company	Date Time Stamp	Time Index	# Distribution Facilities	Order Fullfillment Time
ABC	1990	1	2	26
ABC	1991	2	2	25
ABC	1992	3	4	23
ABC	1993	4	5	22
ABC	1994	5	6	22
ABC	1995	6	6	23
ABC	1996	7	7	21
ABC	1997	8	8	19
ABC	1998	9	10	18

- Using the Time Index is easy to implement and interpret for simple linear time trends on equally spaced data
- But if data has temporal patterns (e.g. seasonality, unequal spacing, etc..), then modeling directly on the Date Time Stamp is preferred.

Cleaning Time Series

Don't blindly drop rows

Cleaning Time Series

Don't blindly drop rows

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
~~ABC~~	~~1995~~	~~6~~	~~NA~~
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Cleaning Time Series

Don't blindly drop rows

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
~~ABC~~	~~1995~~	~~6~~	~~NA~~
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Cleaning Time Series

Don't blindly drop rows

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
~~ABC~~	~~1995~~	~~6~~	~~NA~~
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

Company	Year	# Distribution Facilities	Order Fullfillment Time
ABC	1990	2	26
ABC	1991	2	25
ABC	1992	4	23
ABC	1993	5	22
ABC	1994	6	22
ABC	1996	7	21
ABC	1997	8	19
ABC	1998	10	18

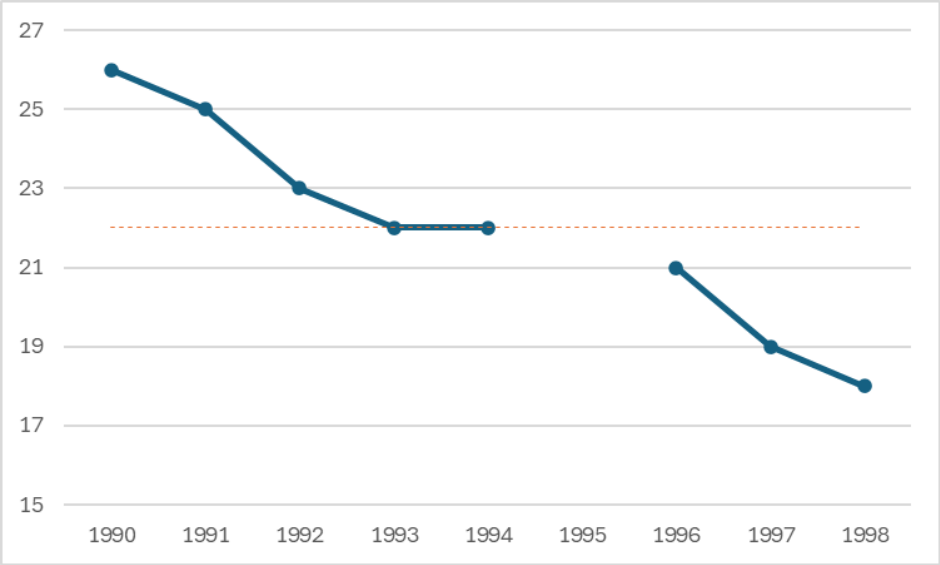
The temporal dependence of the series is disrupted when you drop rows.

Cleaning Time Series

Filling with Sample Mean can be Dangerous

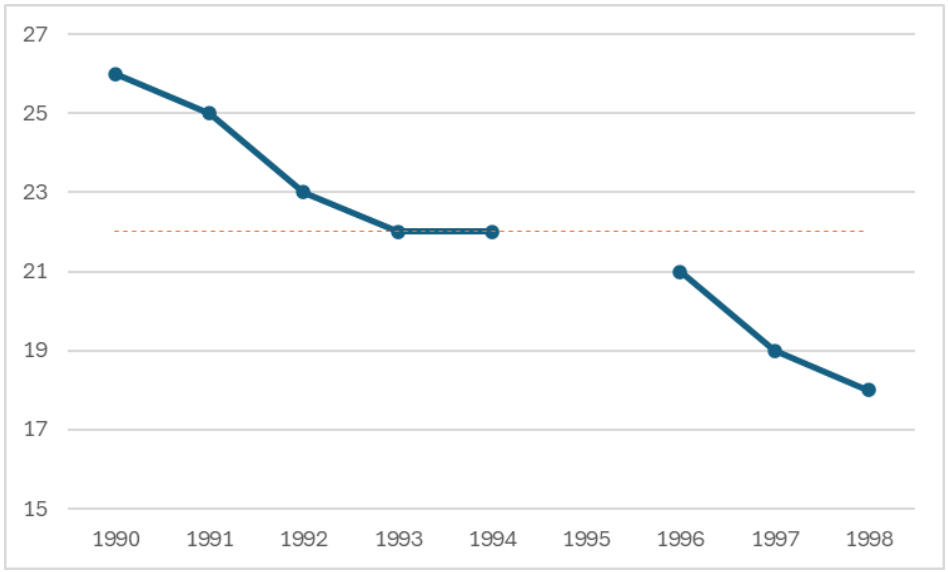
Cleaning Time Series

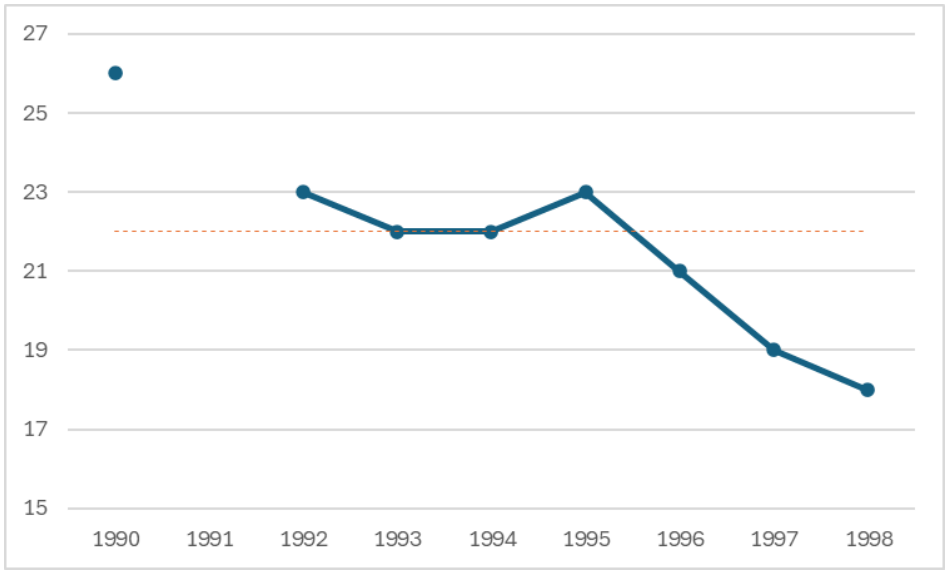
Filling with Sample Mean can be Dangerous



Cleaning Time Series

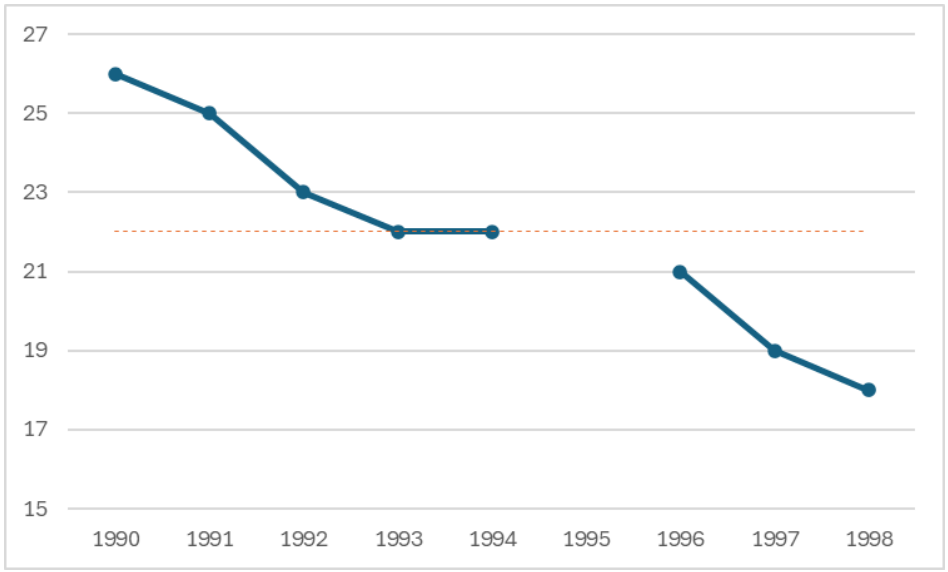
Filling with Sample Mean can be Dangerous

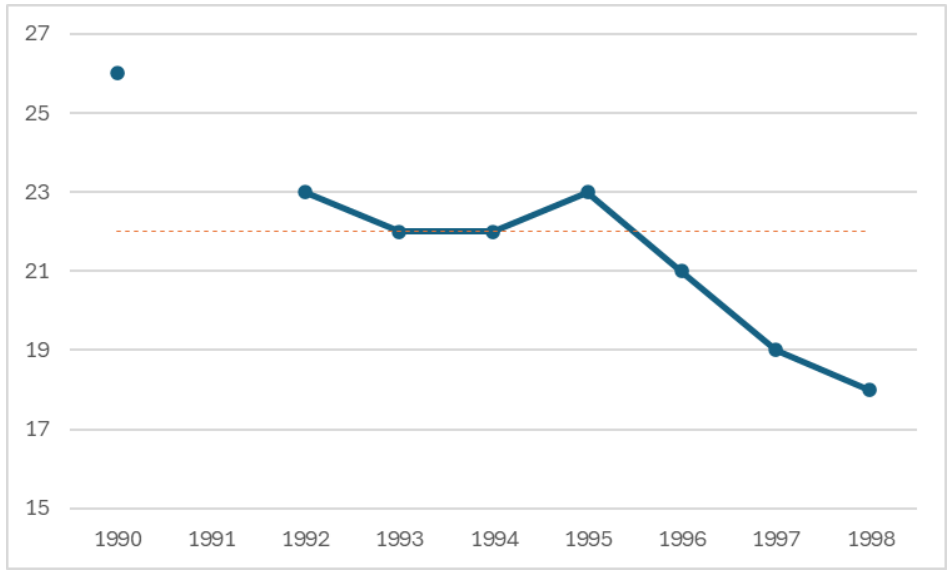




Cleaning Time Series

Filling with Sample Mean can be Dangerous

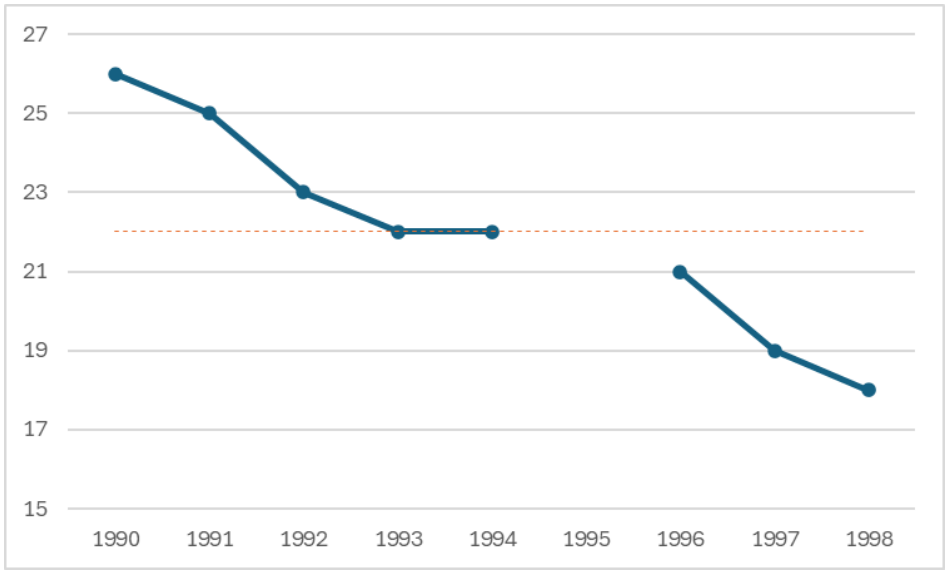


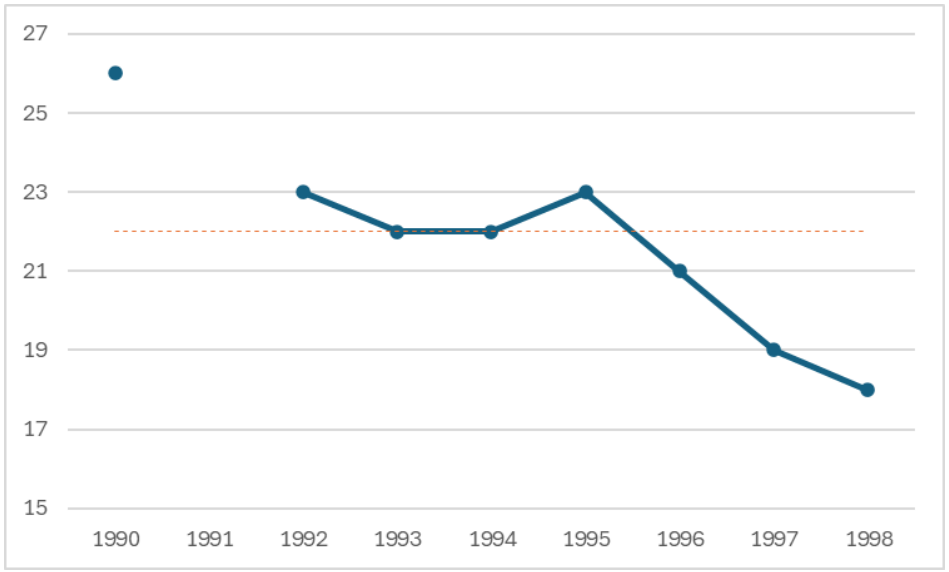


Q: Where is using the mean "dangerous" to fill missing data dangerous?

Cleaning Time Series

Filling with Sample Mean can be Dangerous



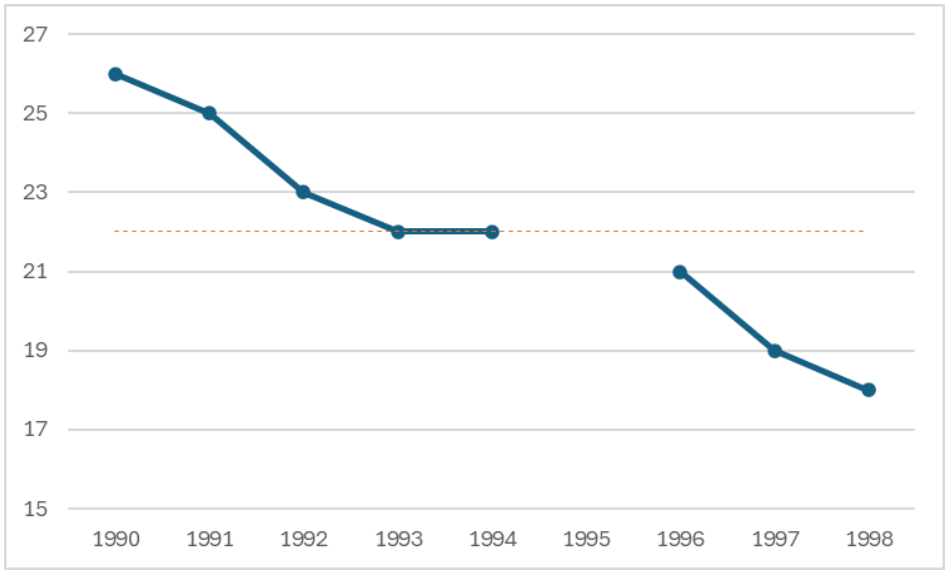


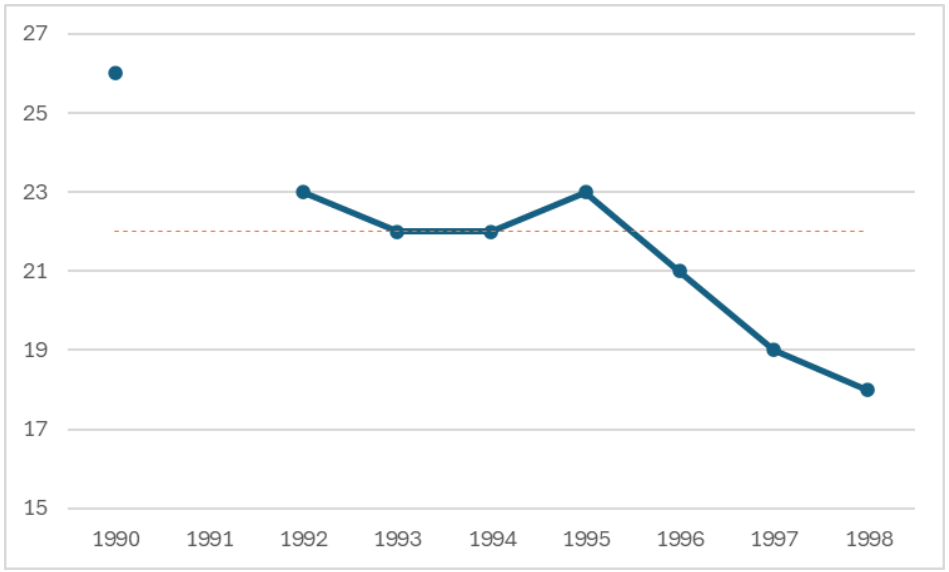
Q: Where is using the mean "dangerous" to fill missing data dangerous?

- While the mean may be a suitable proxy for the series on the left, it is not for the series on the right.

Cleaning Time Series

Filling with Sample Mean can be Dangerous





Q: Where is using the mean "dangerous" to fill missing data dangerous?

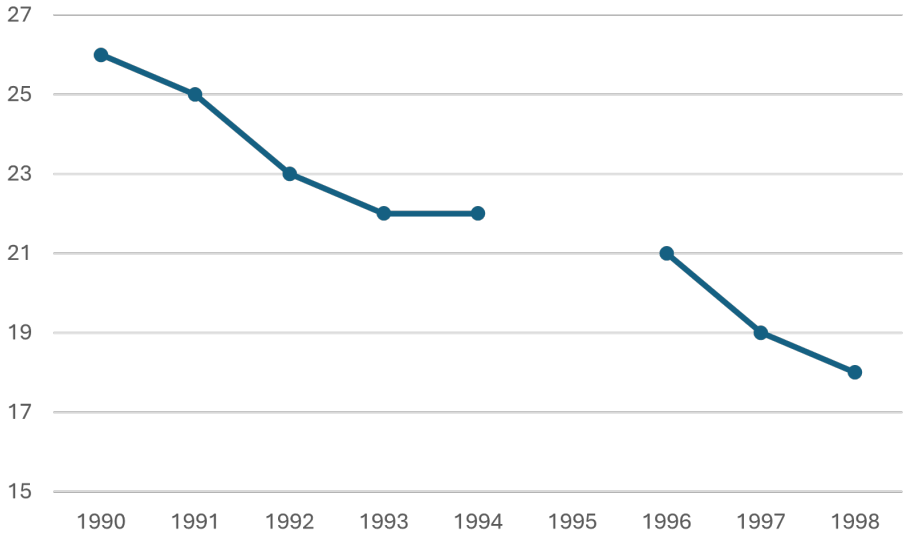
- While the mean may be a suitable proxy for the series on the left, it is not for the series on the right.
- In general, filling with the sample mean is especially dangerous on trending series

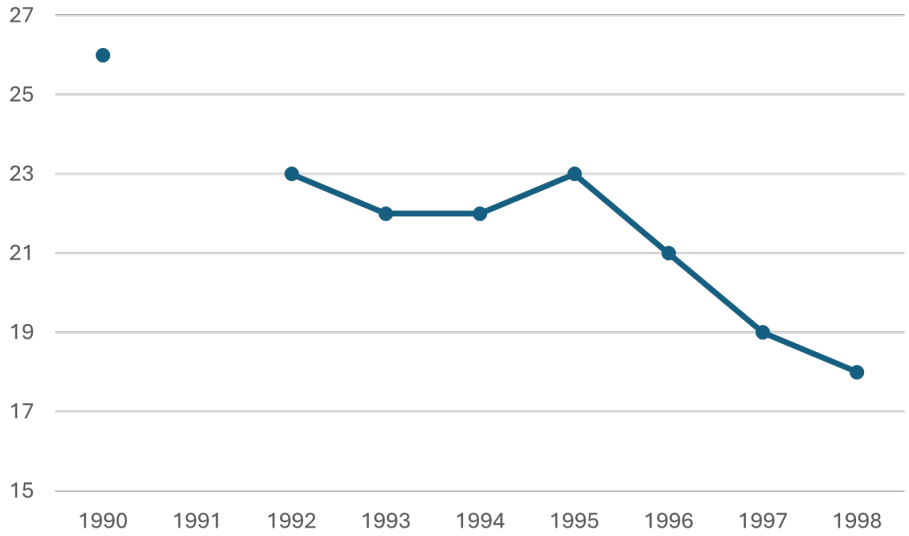
Cleaning Time Series

Interpolating is a common approach

Cleaning Time Series

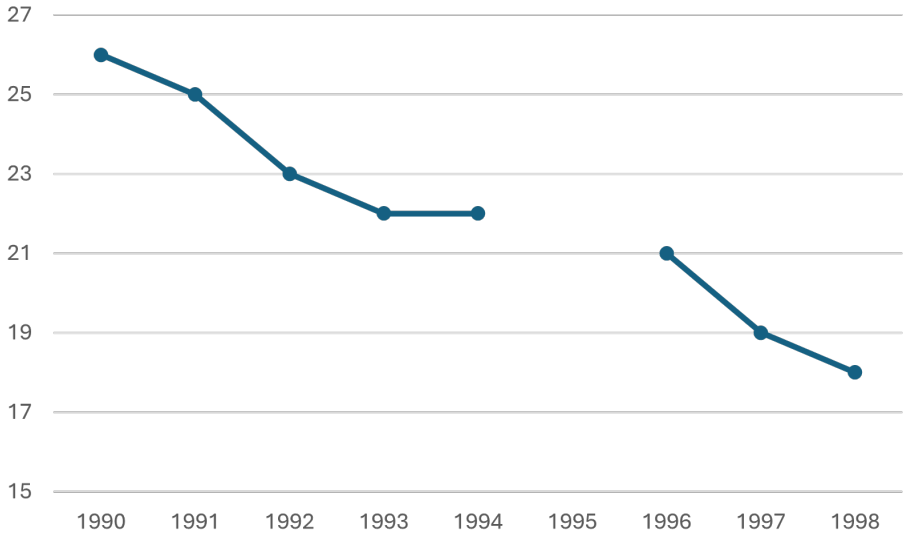
Interpolating is a common approach

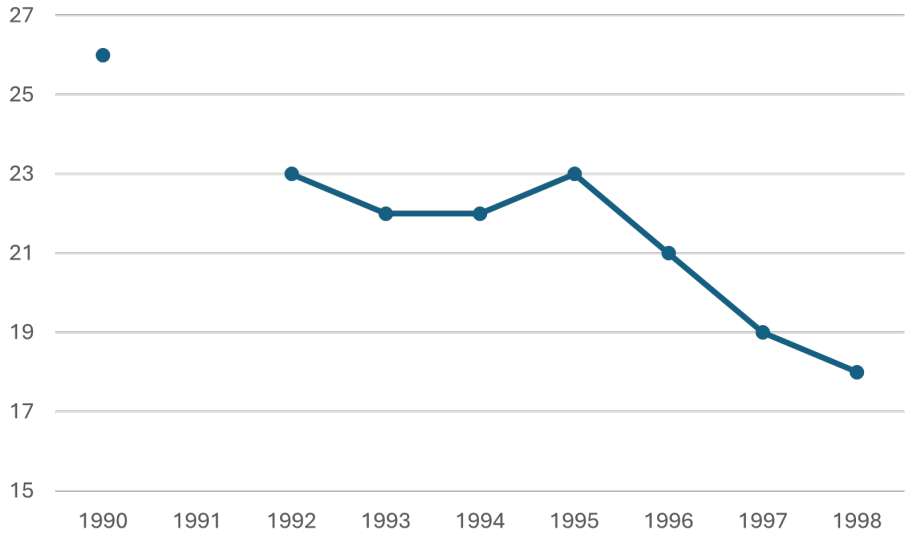




Cleaning Time Series

Interpolating is a common approach

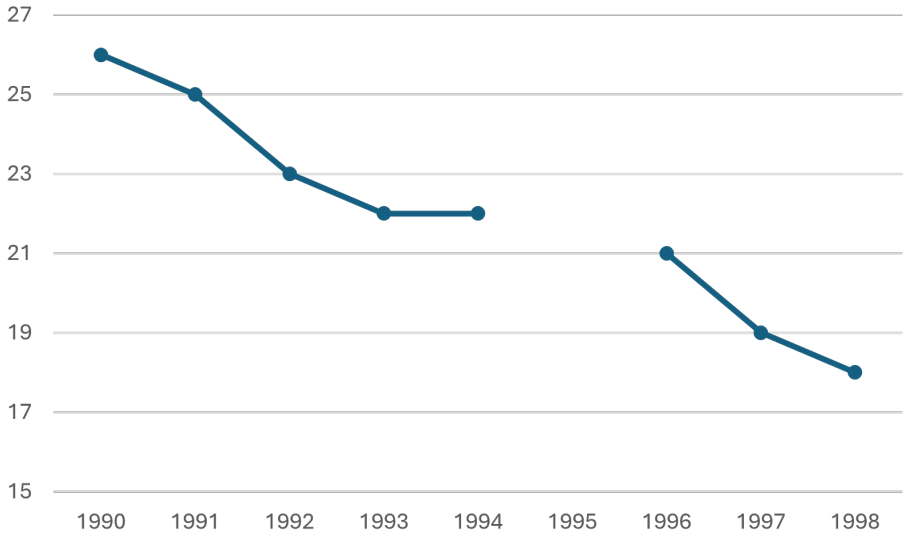


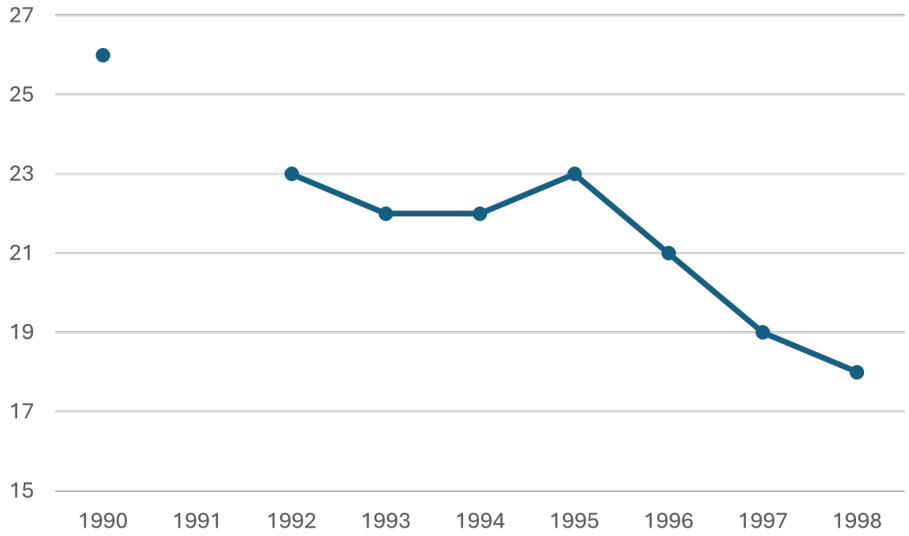


- Interpolating with a local midpoint is common

Cleaning Time Series

Interpolating is a common approach

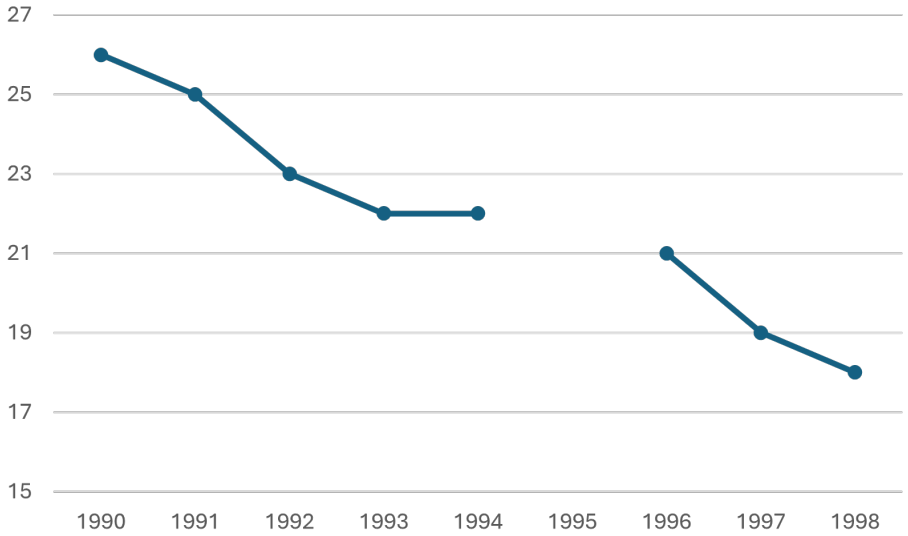


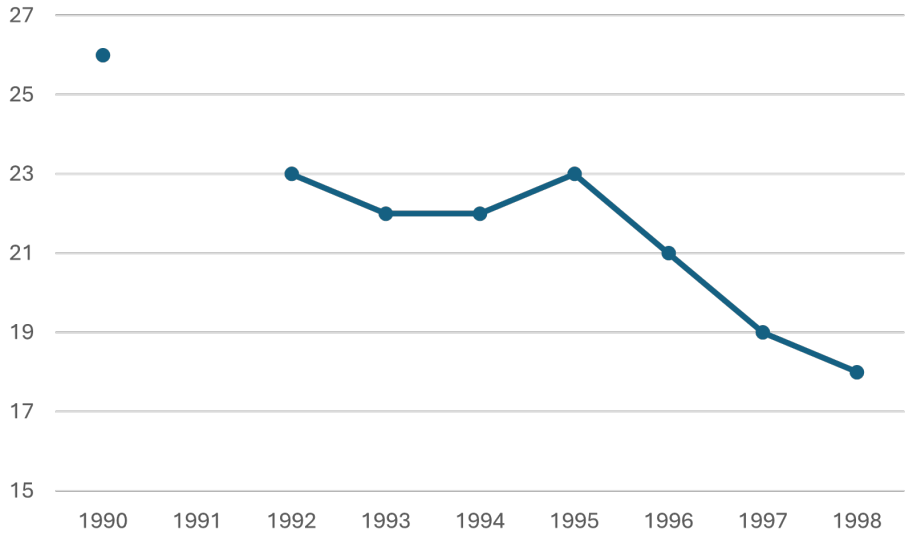


- Interpolating with a local midpoint is common
- Using sophisticated structures (e.g ARMA) is possible, but

Cleaning Time Series

Interpolating is a common approach

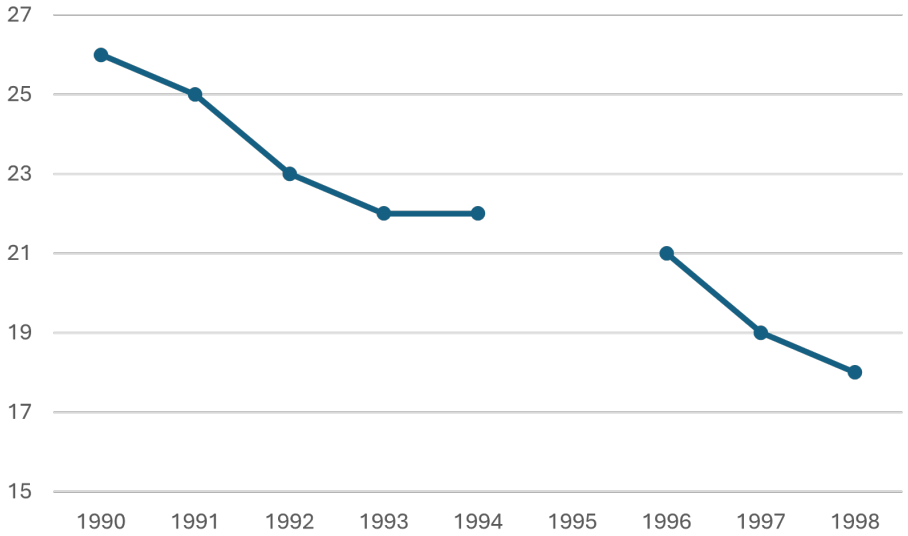


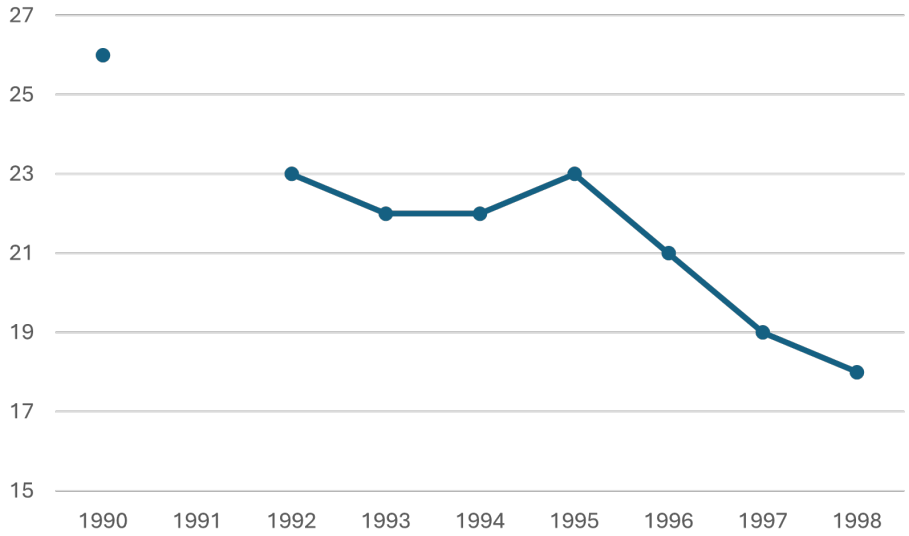


- Interpolating with a local midpoint is common
- Using sophisticated structures (e.g ARMA) is possible, but requires i) more parametric assumptions, and ii)

Cleaning Time Series

Interpolating is a common approach





- Interpolating with a local midpoint is common
- Using sophisticated structures (e.g ARMA) is possible, but requires i) more parametric assumptions, and ii) more data (e.g. modeling the missing point in the right panel is more difficult)

Cleaning Time Series

Forward Filling Values

Cleaning Time Series

Forward Filling Values

Date Time Stamp	Trade Price	Last Trade Price
1/3/2025 10:01	**25.25**	**25.25**
1/3/2025 10:02	**25.35**	**25.35**
1/3/2025 10:03	**26.05**	**26.05**
1/3/2025 10:04		**26.05**
1/3/2025 10:05	**26.10**	**26.10**
1/3/2025 10:06	**26.10**	**26.10**
1/3/2025 10:07	**26.05**	**26.05**
1/3/2025 10:08	**26.00**	**26.00**
1/3/2025 10:09	**25.95**	**25.95**

Cleaning Time Series

Forward Filling Values

Date Time Stamp	Trade Price	Last Trade Price
1/3/2025 10:01	**25.25**	**25.25**
1/3/2025 10:02	**25.35**	**25.35**
1/3/2025 10:03	**26.05**	**26.05**
1/3/2025 10:04		**26.05**
1/3/2025 10:05	**26.10**	**26.10**
1/3/2025 10:06	**26.10**	**26.10**
1/3/2025 10:07	**26.05**	**26.05**
1/3/2025 10:08	**26.00**	**26.00**
1/3/2025 10:09	**25.95**	**25.95**

- Forward filling values can be sensible in some applications

Cleaning Time Series

Forward Filling Values

Date Time Stamp	Trade Price	Last Trade Price
1/3/2025 10:01	**25.25**	**25.25**
1/3/2025 10:02	**25.35**	**25.35**
1/3/2025 10:03	**26.05**	**26.05**
1/3/2025 10:04		**26.05**
1/3/2025 10:05	**26.10**	**26.10**
1/3/2025 10:06	**26.10**	**26.10**
1/3/2025 10:07	**26.05**	**26.05**
1/3/2025 10:08	**26.00**	**26.00**
1/3/2025 10:09	**25.95**	**25.95**

- Forward filling values can be sensible in some applications
- Here, the last trade at 10:03 is the best information available at 10:04

Mixed Frequencies

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP)

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP) Caution: assumes a lot of structure

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP) Caution: assumes a lot of structure
- Aggregate the high frequency (UR)

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP) Caution: assumes a lot of structure
- Aggregate the high frequency (UR)
  1. Average of monthly data within quarter

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP) Caution: assumes a lot of structure
- Aggregate the high frequency (UR)
  1. Average of monthly data within quarter
  2. Last of monthly data within quarter

Mixed Frequencies

Date	UR	GDP
Jan-20	4.2	
Feb-20	4.3	
Mar-20	9	3.2
Apr-20	12.5	
May-20	12.6	
Jun-20	10	-5
Jul-20	9	
Aug-20	9.2	
Sep-20	8	4
Oct-20	6	
Nov-20	6.5	
Dec-20	5	3

Q: Options for data prep to facilitate modeling?

- Interpolate low frequency (GDP) Caution: assumes a lot of structure
- Aggregate the high frequency (UR)
  1. Average of monthly data within quarter
  2. Last of monthly data within quarter
- Leave as is and use mixed frequency modeling techniques (e.g. MIDAS)

Avoid Lookahead Bias

Avoid Lookahead Bias

- Goal: Predict Stock Price

Avoid Lookahead Bias

- Goal: Predict Stock Price
- Use Case:

Avoid Lookahead Bias

- Goal: Predict Stock Price
- Use Case: When earnings is released I will use my model to predict the stock price reaction in order to guide my trade.

Avoid Lookahead Bias

- Goal: Predict Stock Price
- Use Case: When earnings is released I will use my model to predict the stock price reaction in order to guide my trade.
- Simple Linear Model:

Avoid Lookahead Bias

- Goal: Predict Stock Price
- Use Case: When earnings is released I will use my model to predict the stock price reaction in order to guide my trade.
- Simple Linear Model: $\mathit{Price} = \alpha + \beta \mathit{Earnings} + u$.
  Note: We usually don't model price directly. We also need to be more precise about forecast horizon. But let us ignore for now.

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

A:

- We can't deploy this in practice. The stock price on 31-Mar is not driven by the release of Q1 earnings because that won't be known until early Q2.

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

A:

- We can't deploy this in practice. The stock price on 31-Mar is not driven by the release of Q1 earnings because that won't be known until early Q2.
- Our model estimates suffer from Lookahead Bias

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

A:

- We can't deploy this in practice. The stock price on 31-Mar is not driven by the release of Q1 earnings because that won't be known until early Q2.
- Our model estimates suffer from Lookahead Bias
- Solution:

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

A:

- We can't deploy this in practice. The stock price on 31-Mar is not driven by the release of Q1 earnings because that won't be known until early Q2.
- Our model estimates suffer from Lookahead Bias
- Solution: align release dates;

Avoid Lookahead Bias

Q: What's wrong with setting up our data this way?

Date	Quarter	EPS	Stock Price
31-Mar	Q1	6.12	225
30-Jun	Q2	6.14	236
30-Sep	Q3	6.19	212
31-Dec	Q4	6.23	249

A:

- We can't deploy this in practice. The stock price on 31-Mar is not driven by the release of Q1 earnings because that won't be known until early Q2.
- Our model estimates suffer from Lookahead Bias
- Solution: align release dates; e.g. if Q1 earnings were released on 24Apr, then put that EPS in the same row as the stock price on that day

Outline

1 Introduction

2 Data Prep

3 Explore

4 Explain

5 Forecast

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

EDA for Time Series

EDA has the same purpose in time series as it does in cross sectional settings.

EDA for Time Series

EDA has the same purpose in time series as it does in cross sectional settings.

We encounter a few special challenges due to the nature of the data.

EDA for Time Series

EDA has the same purpose in time series as it does in cross sectional settings.

We encounter a few special challenges due to the nature of the data.

We also have a few new tools.

Time Series Tools for EDA

- Autocorrelation Function
- Partial Autocorrelation Function
- Cross Correlation Function

Autocorrelation Functions

Definition

The k^{th} autocorrelation is defined as

$$\rho_y(k) = \frac{Cov(y_t, y_{t-k})}{V(y_t)}$$

Autocorrelation Functions

Definition

The k^{th} autocorrelation is defined as

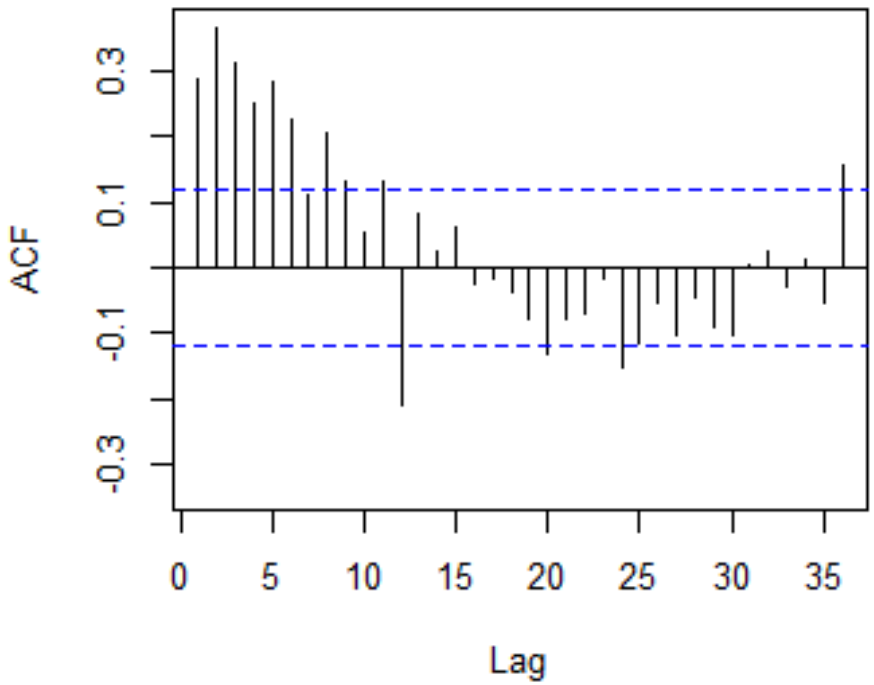
$$\rho_y(k) = \frac{Cov(y_t, y_{t-k})}{V(y_t)}$$

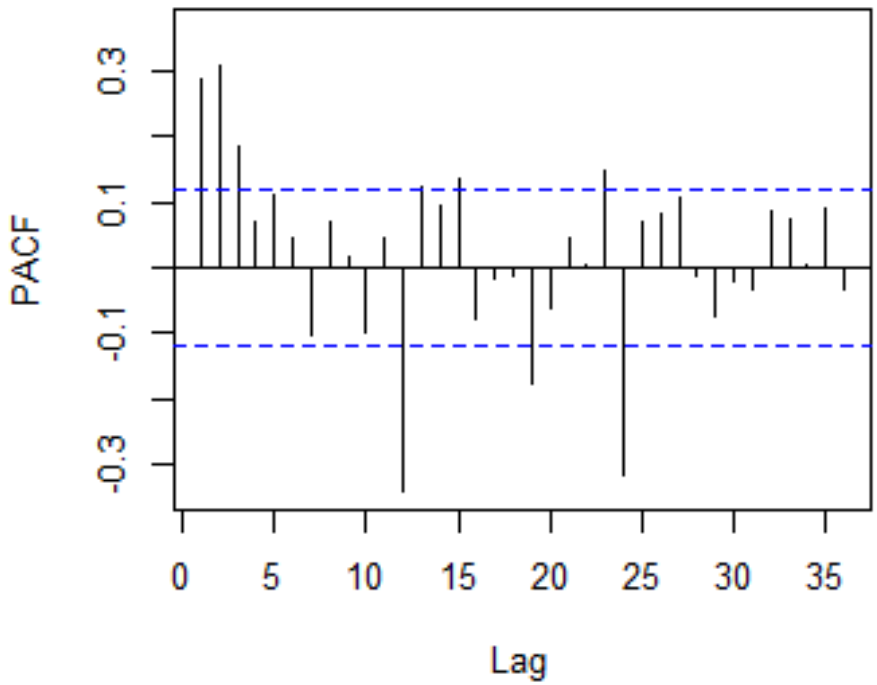
Definition

The k^{th} partial autocorrelation is defined as the estimate of θ_k in the AR(k) autoregression. For instance

- $pac_y(1) = \hat{\theta}_1$ from $y_t = \theta_1 y_{t-1} + \varepsilon_t$
- $pac_y(2) = \hat{\theta}_2$ from $y_t = \theta_1 y_{t-1} + \theta_2 y_{t-2} + \varepsilon_t$

ACF and PACF


ACF
0.3
0.1
-0.1
-0.3
0
5
10
15
20
25
30
35
Lag


PACF
0.3
0.1
-0.1
-0.3
0
5
10
15
20
25
30
35
Lag

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) $= 0.25$ and PACF(5) $= 0.10$?

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) $= 0.25$ and PACF(5) $= 0.10$?
A: Let y_t denote daily returns.

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t - 5$ to t

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \rightarrow t$

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \to t$
  - Indirect: $t-5 \to t-4 \to t-3 \to \cdots \to t$
- PACF(5) = 0.10

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \to t$
  - Indirect: $t-5 \to t-4 \to t-3 \to \cdots \to t$
- PACF(5) = 0.10
  - Keeps only the **direct path**

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \to t$
  - Indirect: $t-5 \to t-4 \to t-3 \to \cdots \to t$
- PACF(5) = 0.10
  - Keeps only the **direct path**
  - $t-5 \to t$

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \to t$
  - Indirect: $t-5 \to t-4 \to t-3 \to \cdots \to t$
- PACF(5) = 0.10
  - Keeps only the **direct path**
  - $t-5 \to t$
  - Removes indirect paths through $t-5 \to t-4 \to \cdots \to t$

Interpreting ACF vs PACF

Q: How do we interpret ACF(5) = 0.25 and PACF(5) = 0.10?

A: Let y_t denote daily returns.

- ACF(5) = 0.25
  - Returns today have a **0.25 correlation** with returns five days ago
  - Includes **all paths** from $t-5$ to t
  - Direct: $t-5 \to t$
  - Indirect: $t-5 \to t-4 \to t-3 \to \cdots \to t$
- PACF(5) = 0.10
  - Keeps only the **direct path**
  - $t-5 \to t$
  - Removes indirect paths through $t-5 \to t-4 \to \cdots \to t$

Since PACF(5) $<$ ACF(5), most of the dependence is transmitted through shorter lags rather than a direct effect.

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that x lags y.

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that x lags y.
- If $\rho_{x,y}(-1)$ significant we say that

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that x lags y.
- If $\rho_{x,y}(-1)$ significant we say that x leads y.

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that x lags y.
- If $\rho_{x,y}(-1)$ significant we say that x leads y.

Q: What does $\rho_{r_{IBM}, r_{SP500}}(2) = 0.1$ mean?

Cross Autocorrelation Functions

Definition

The k^{th} cross-correlation is defined as the as

$$\rho_{x,y}(k) = \frac{Cov(x_{t+k}, y_t)}{\sigma_x \sigma_y}$$

Notice that unlike the univariate autocorrelation function, the cross-correlation function is asymmetric.

- If $\rho_{x,y}(1)$ significant we say that x lags y.
- If $\rho_{x,y}(-1)$ significant we say that x leads y.

Q: What does $\rho_{r_{IBM}, r_{SP500}}(2) = 0.1$ mean?
A: Correlation between SP500 returns on "Monday" and IBM returns on "Wednesday"is 0.1; i.e. IBM responds with a two day lag.

Outline

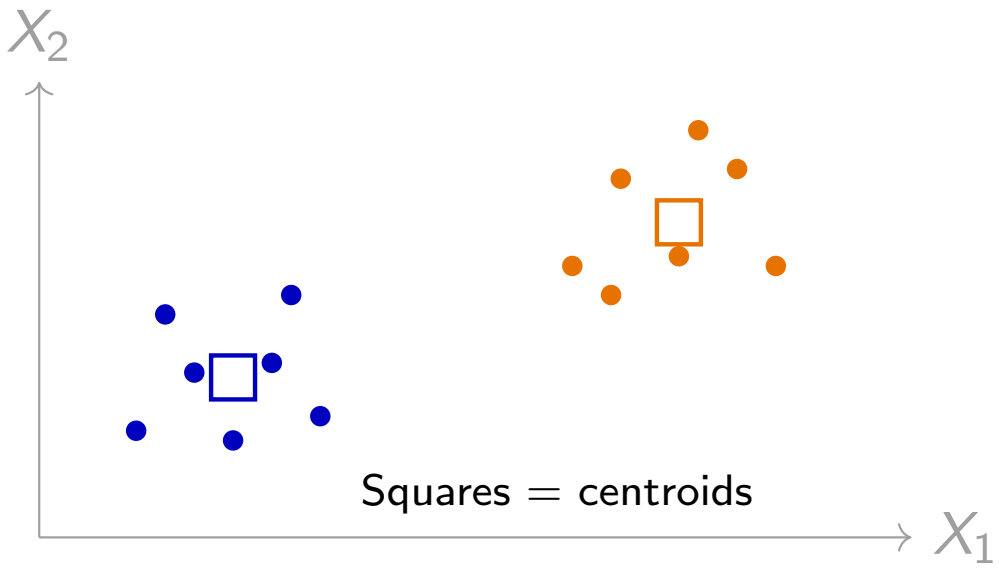
3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

k-Means on Cross-Sectional Data

Q: How did we implement k-means on cross-sectional data?

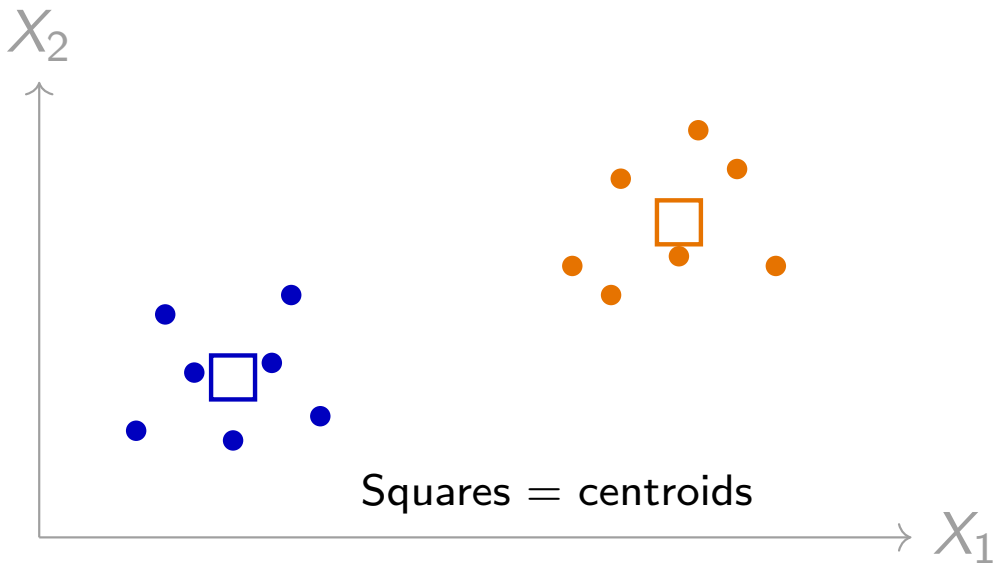
k-Means on Cross-Sectional Data

Q: How did we implement k-means on cross-sectional data?



k-Means on Cross-Sectional Data

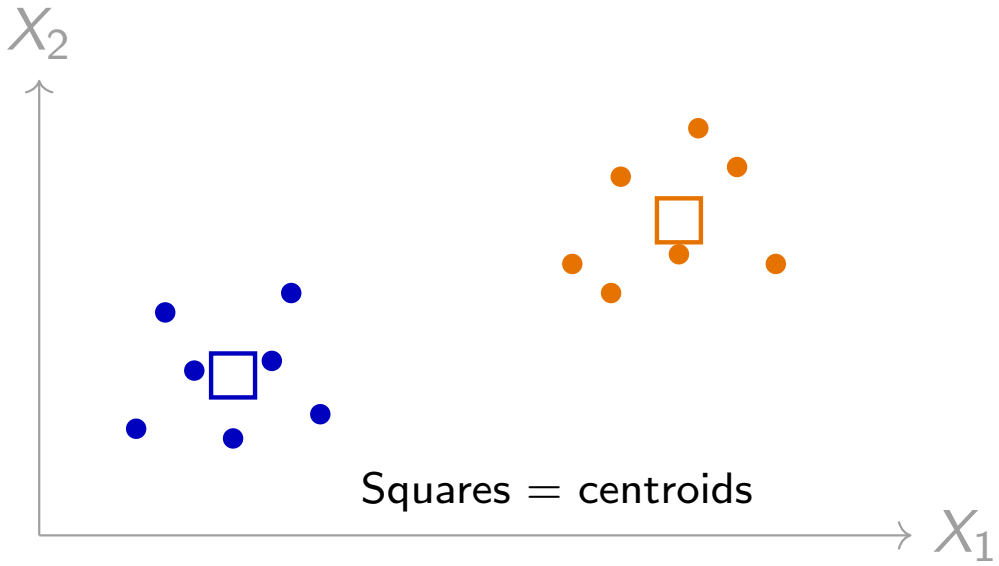
Q: How did we implement k-means on cross-sectional data?



1. Initialize K centroids.

k-Means on Cross-Sectional Data

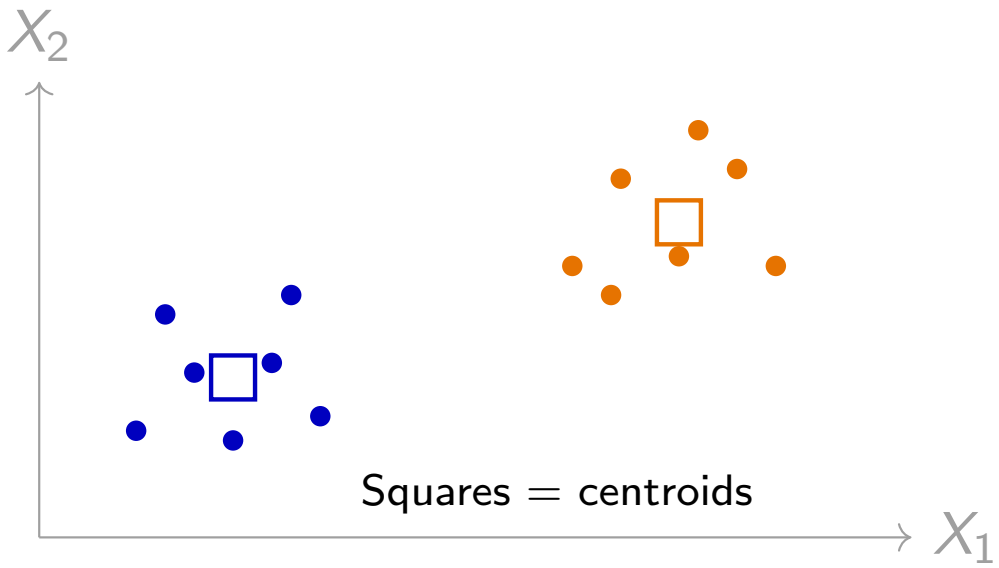
Q: How did we implement k-means on cross-sectional data?



1. Initialize K centroids.
2. Assign each observation to its nearest centroid.

k-Means on Cross-Sectional Data

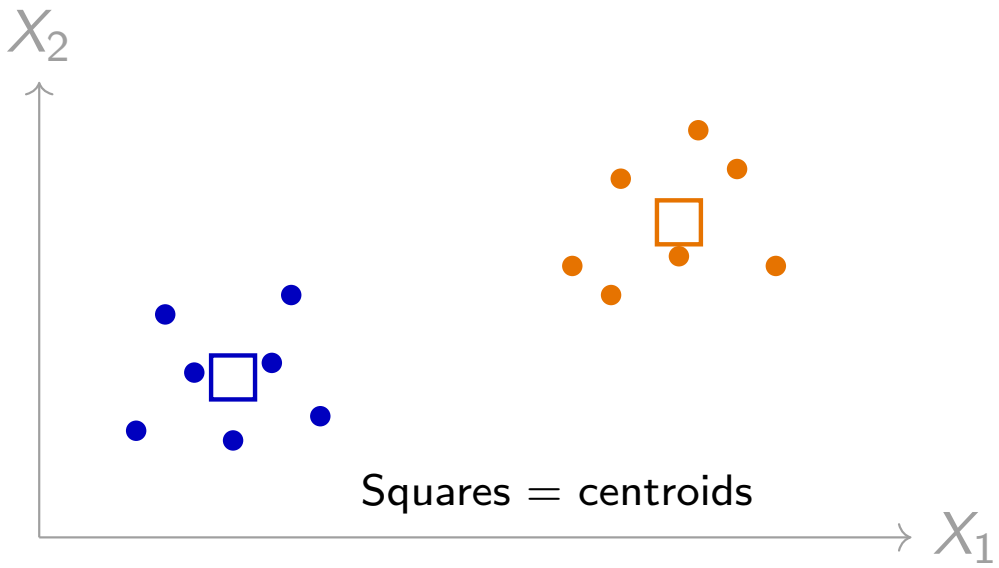
Q: How did we implement k-means on cross-sectional data?



1. Initialize K centroids.
2. Assign each observation to its nearest centroid.
3. Update each centroid to the average of its assigned observations.

k-Means on Cross-Sectional Data

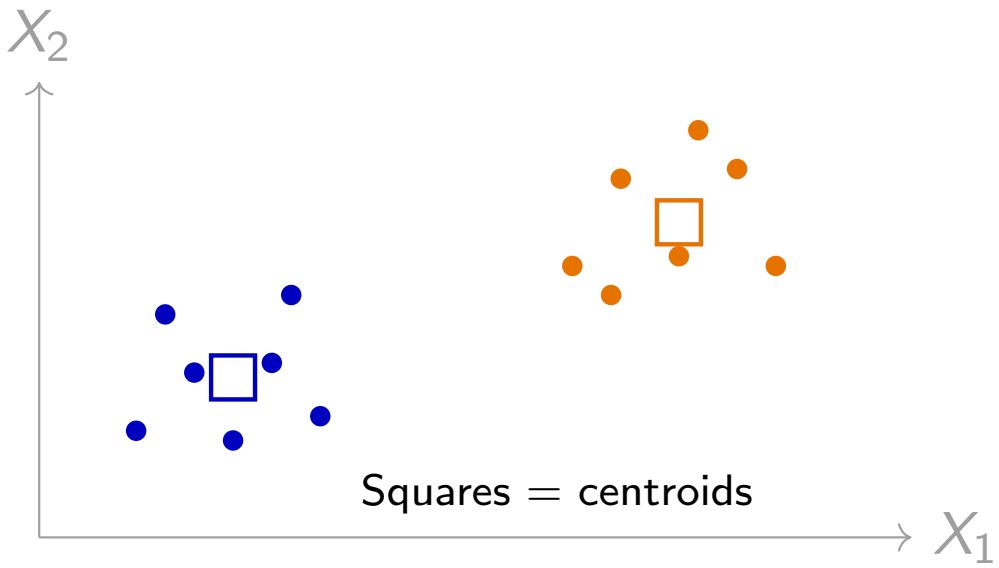
Q: How did we implement k-means on cross-sectional data?



1. Initialize K centroids.
2. Assign each observation to its nearest centroid.
3. Update each centroid to the average of its assigned observations.
4. Repeat assignment and updating until convergence.

k-Means on Cross-Sectional Data

Q: How did we implement k-means on cross-sectional data?



1. Initialize K centroids.
2. Assign each observation to its nearest centroid.
3. Update each centroid to the average of its assigned observations.
4. Repeat assignment and updating until convergence.

For time series, the key challenge is deciding what "near" means.

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.
- **Fixed timing difference:** Does one consistently follow the other?
  Measure: lagged correlation.

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.
- **Fixed timing difference:** Does one consistently follow the other?
  Measure: lagged correlation.
- **Flexible timing:** Do they follow similar paths at different speeds?
  Measure: Dynamic Time Warping (DTW).

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.
- **Fixed timing difference:** Does one consistently follow the other?
  Measure: lagged correlation.
- **Flexible timing:** Do they follow similar paths at different speeds?
  Measure: Dynamic Time Warping (DTW).
- **Statistical dynamics:** Do they show similar volatility or persistence?
  Measure: time-series features.

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.
- **Fixed timing difference:** Does one consistently follow the other?
  Measure: lagged correlation.
- **Flexible timing:** Do they follow similar paths at different speeds?
  Measure: Dynamic Time Warping (DTW).
- **Statistical dynamics:** Do they show similar volatility or persistence?
  Measure: time-series features.
- **Underlying process:** Do similar models describe their dynamics?
  Measure: model parameters or model-based similarity.

What Does Similarity Mean for Time Series?

Q: In what ways can two time series be similar?

- **Same-time movement:** Do they rise and fall together?
  Measure: contemporaneous correlation.
- **Fixed timing difference:** Does one consistently follow the other?
  Measure: lagged correlation.
- **Flexible timing:** Do they follow similar paths at different speeds?
  Measure: Dynamic Time Warping (DTW).
- **Statistical dynamics:** Do they show similar volatility or persistence?
  Measure: time-series features.
- **Underlying process:** Do similar models describe their dynamics?
  Measure: model parameters or model-based similarity.

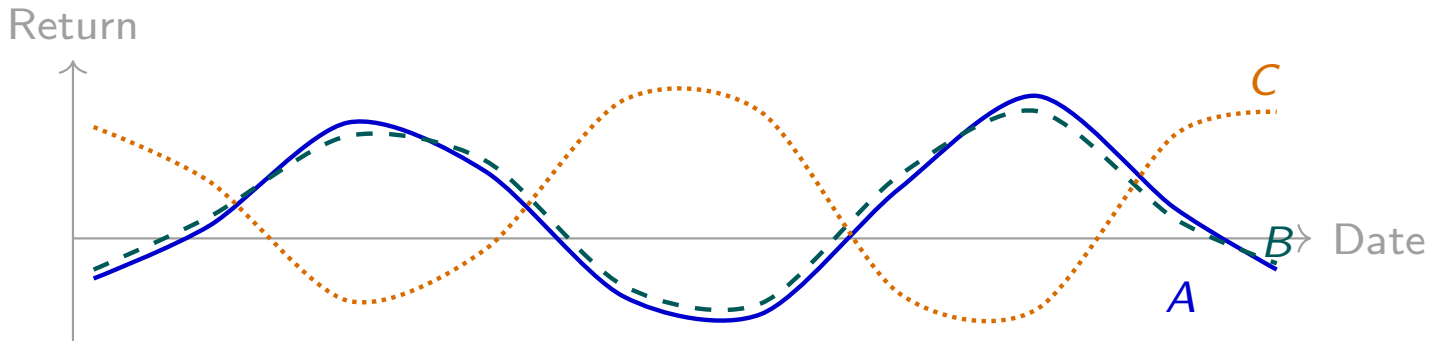
Important: These are different definitions of similarity, not successive improvements to one measure.

Clustering Aligned Time Series

Q: Which assets move together at the same time?

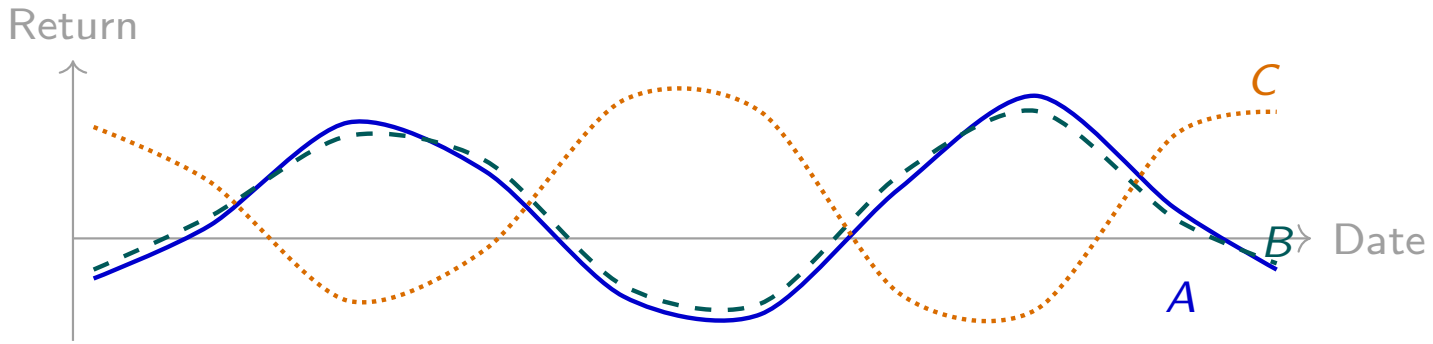
Clustering Aligned Time Series

Q: Which assets move together at the same time?



Clustering Aligned Time Series

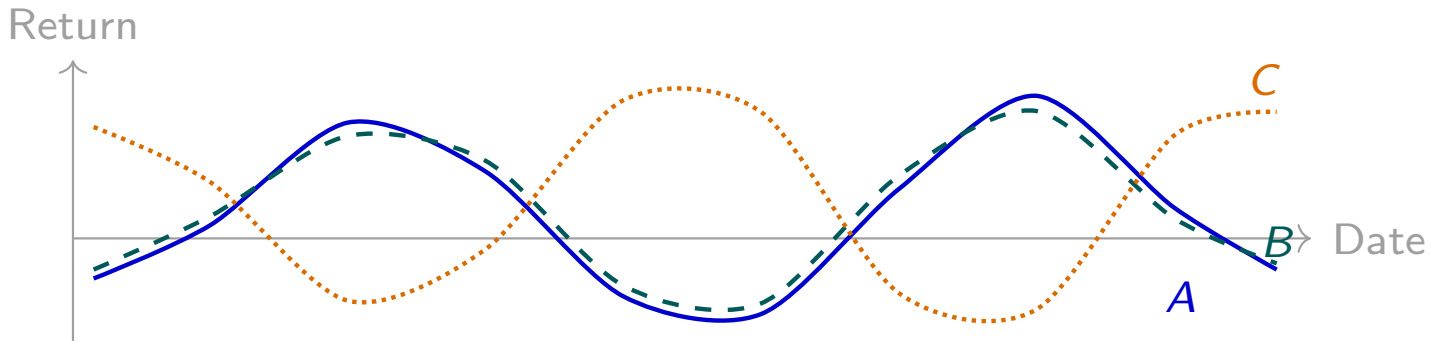
Q: Which assets move together at the same time?



- Assets A and B tend to rise and fall together.

Clustering Aligned Time Series

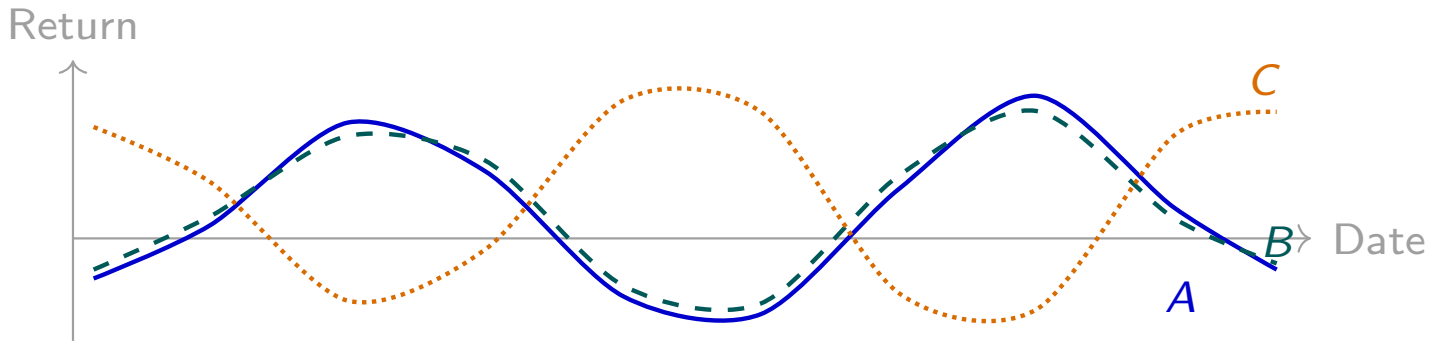
Q: Which assets move together at the same time?



- Assets A and B tend to rise and fall together.
- Asset C often moves in the opposite direction.

Clustering Aligned Time Series

Q: Which assets move together at the same time?



- Assets A and B tend to rise and fall together.
- Asset C often moves in the opposite direction.
- When dates are aligned, **contemporaneous correlation** summarizes this co-movement across entire return histories.

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

Suppose N assets each have T returns. Place their histories in columns:

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(N)} \end{bmatrix}_{T \times N}, \qquad X^{(i)} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{iT} \end{bmatrix}.$$

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

Suppose N assets each have T returns. Place their histories in columns:

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(N)} \end{bmatrix}_{T \times N}, \qquad X^{(i)} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{iT} \end{bmatrix}.$$

Compute the correlation between every pair of columns. With three assets:

$$R = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1 & .90 & -.70 \\ B & .90 & 1 & -.65 \\ C & -.70 & -.65 & 1 \end{array}$$

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

Suppose N assets each have T returns. Place their histories in columns:

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(N)} \end{bmatrix}_{T \times N}, \qquad X^{(i)} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{iT} \end{bmatrix}.$$

Compute the correlation between every pair of columns. With three assets:

$$R = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1 & .90 & -.70 \\ B & .90 & 1 & -.65 \\ C & -.70 & -.65 & 1 \end{array}$$

The $N \times N$ matrix $R = [\widehat{\rho}_{ij}]$ summarizes *pairwise similarities*.

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

Suppose N assets each have T returns. Place their histories in columns:

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(N)} \end{bmatrix}_{T \times N}, \qquad X^{(i)} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{iT} \end{bmatrix}.$$

Compute the correlation between every pair of columns. With three assets:

$$R = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1 & .90 & -.70 \\ B & .90 & 1 & -.65 \\ C & -.70 & -.65 & 1 \end{array}$$

The $N \times N$ matrix $R = [\widehat{\rho}_{ij}]$ summarizes *pairwise similarities*. K-medoids uses dissimilarities (smaller = more alike), so transform:

$$d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}, \qquad D = [d_{ij}]_{N \times N}.$$

From Time Series to Pairwise Dissimilarities

Q: How do return histories become a correlation matrix?

Suppose N assets each have T returns. Place their histories in columns:

$$X = \begin{bmatrix} X^{(1)} & X^{(2)} & \cdots & X^{(N)} \end{bmatrix}_{T \times N}, \qquad X^{(i)} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{iT} \end{bmatrix}.$$

Compute the correlation between every pair of columns. With three assets:

$$R = \begin{array}{c|ccc} & A & B & C \\ \hline A & 1 & .90 & -.70 \\ B & .90 & 1 & -.65 \\ C & -.70 & -.65 & 1 \end{array}$$

The $N \times N$ matrix $R = [\widehat{\rho}_{ij}]$ summarizes *pairwise similarities*. K-medoids uses dissimilarities (smaller = more alike), so transform:

$$d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}, \qquad D = [d_{ij}]_{N \times N}.$$

Interpretation: A and B are close; C is far from both because it tends to move in the opposite direction.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

k-Medoids: a different cluster representative

Select the series that **minimizes total within-cluster dissimilarity**:

$$m_k = \arg\min_{i \in C_k} \sum_{j \in C_k} d_{ij}.$$

Thus, the *medoid* is an actual observed series.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

k-Medoids: a different cluster representative

Select the series that **minimizes total within-cluster dissimilarity**:

$$m_k = \arg\min_{i \in C_k} \sum_{j \in C_k} d_{ij}.$$

Thus, the *medoid* is an actual observed series.

1. Initialize K medoids (observed series).

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

k-Medoids: a different cluster representative

Select the series that **minimizes total within-cluster dissimilarity**:

$$m_k = \arg\min_{i \in C_k} \sum_{j \in C_k} d_{ij}.$$

Thus, the *medoid* is an actual observed series.

1. Initialize K medoids (observed series).
2. Assign each series to its nearest medoid.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

k-Medoids: a different cluster representative

Select the series that **minimizes total within-cluster dissimilarity**:

$$m_k = \arg\min_{i \in C_k} \sum_{j \in C_k} d_{ij}.$$

Thus, the *medoid* is an actual observed series.

1. Initialize K medoids (observed series).
2. Assign each series to its nearest medoid.
3. Update each medoid using the within-cluster distances; repeat until stable.

From k-Means to k-Medoids

Q: How do we cluster from pairwise dissimilarities?

Recall: **k-means** averages observation vectors to obtain each centroid. Here we work with $D = [d_{ij}]$, containing pairwise dissimilarities. Standard k-means cannot calculate its arithmetic centroids from D alone.

k-Medoids: a different cluster representative

Select the series that **minimizes total within-cluster dissimilarity**:

$$m_k = \arg\min_{i \in C_k} \sum_{j \in C_k} d_{ij}.$$

Thus, the *medoid* is an actual observed series.

1. Initialize K medoids (observed series).
2. Assign each series to its nearest medoid.
3. Update each medoid using the within-cluster distances; repeat until stable.

Advantage: We can change the time-series dissimilarity measure without changing the clustering algorithm.

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

1. Convert correlations into dissimilarities: $d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}$.

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

1. Convert correlations into dissimilarities: $d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}$.
2. Initialize medoids among the *assets*, not among the correlation coefficients.

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

1. Convert correlations into dissimilarities: $d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}$.
2. Initialize medoids among the *assets*, not among the correlation coefficients.
3. Assign assets to their nearest medoids and update; repeat until stable.

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

1. Convert correlations into dissimilarities: $d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}$.
2. Initialize medoids among the *assets*, not among the correlation coefficients.
3. Assign assets to their nearest medoids and update; repeat until stable.
4. With $K = 2$, $\{A, B\}$ and $\{C, D\}$ are the natural groups.

Clustering Assets Using a Correlation Matrix

Q: How do pairwise correlations become clusters?

For illustration, consider four assets:

$$R = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 1 & .90 & .05 & .00 \\ B & .90 & 1 & .04 & .02 \\ C & .05 & .04 & 1 & .85 \\ D & .00 & .02 & .85 & 1 \end{array}$$

1. Convert correlations into dissimilarities: $d_{ij} = \sqrt{2(1 - \widehat{\rho}_{ij})}$.
2. Initialize medoids among the *assets*, not among the correlation coefficients.
3. Assign assets to their nearest medoids and update; repeat until stable.
4. With $K = 2$, $\{A, B\}$ and $\{C, D\}$ are the natural groups.

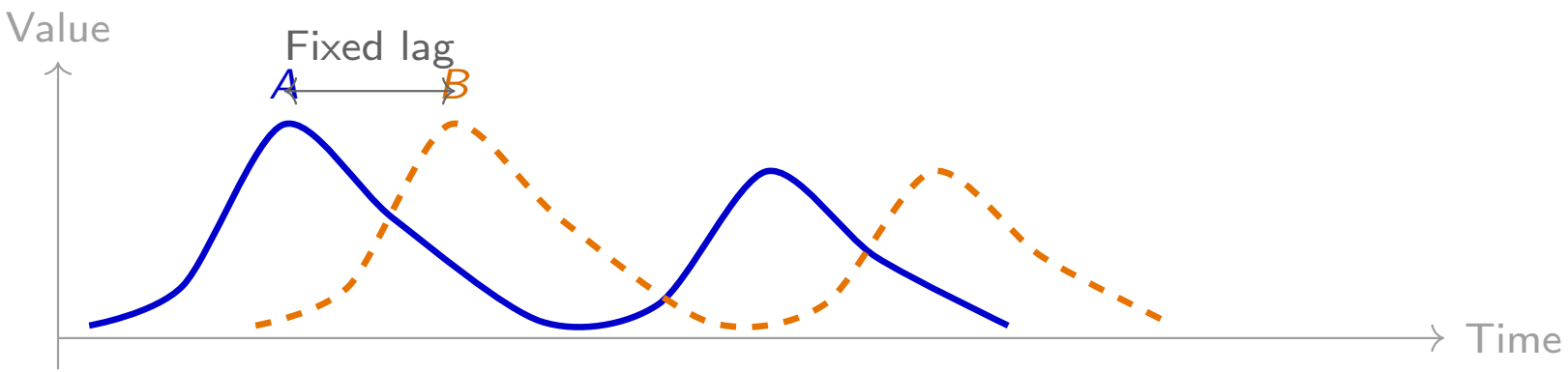
For aligned series, ordinary k-means on standardized return histories is also possible. We use k-medoids so that lagged-correlation and DTW dissimilarities can be used later without changing the clustering algorithm.

Clustering Time Series with Leads/Lags

Q: What if one asset consistently responds later than another?

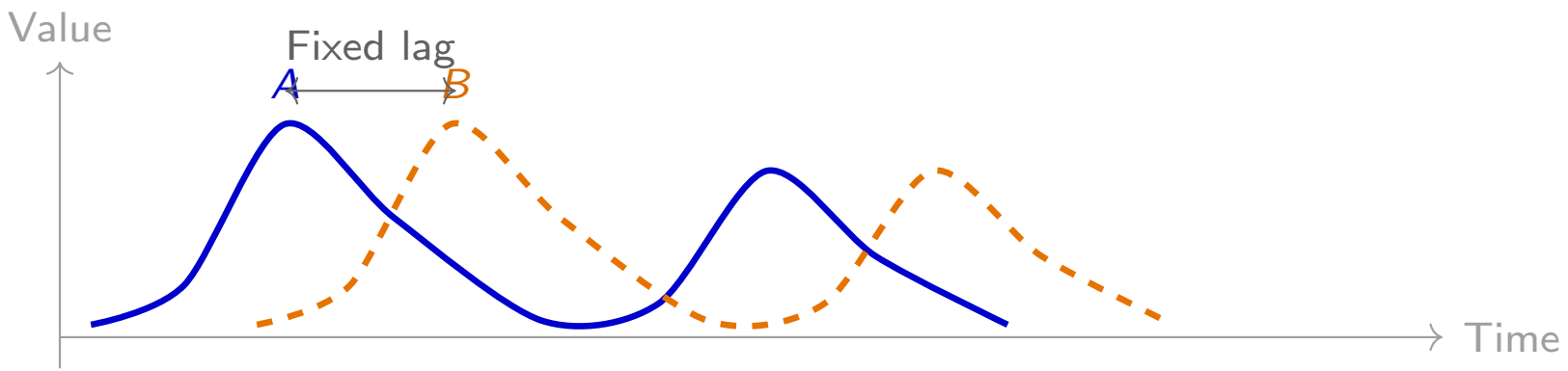
Clustering Time Series with Leads/Lags

Q: What if one asset consistently responds later than another?



Clustering Time Series with Leads/Lags

Q: What if one asset consistently responds later than another?

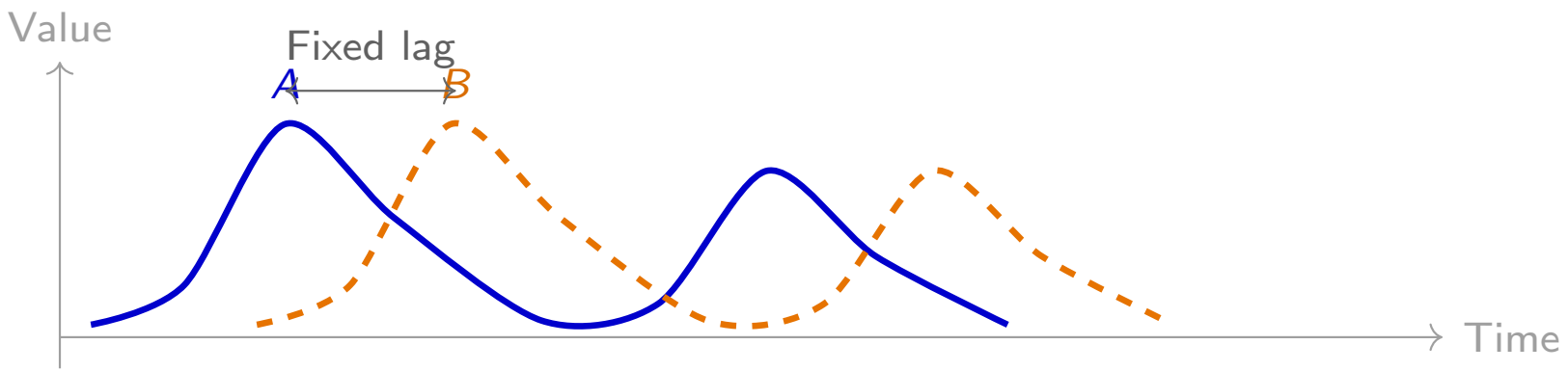


Search over plausible fixed lags $\ell \in \{-L, \ldots, L\}$:

$$\widehat{\rho}_{ij}(\ell) = \mathrm{Corr}(x_{i,t}, x_{j,t+\ell}), \qquad d_{ij}^{\mathrm{lag}} = 1 - \max_{|\ell| \le L} \widehat{\rho}_{ij}(\ell).$$

Clustering Time Series with Leads/Lags

Q: What if one asset consistently responds later than another?



Search over plausible fixed lags $\ell \in \{-L, \ldots, L\}$:

$$\widehat{\rho}_{ij}(\ell) = \mathrm{Corr}(x_{i,t}, x_{j,t+\ell}), \qquad d_{ij}^{\mathrm{lag}} = 1 - \max_{|\ell| \le L} \widehat{\rho}_{ij}(\ell).$$

Positive ℓ means j lags i. The maximum is computed over permissible lags using overlapping observations.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

3. Use the resulting $N \times N$ matrix with **k-medoids**, exactly as before.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

3. Use the resulting $N \times N$ matrix with **k-medoids**, exactly as before.

- Different pairs may have different best lags. A single common alignment across all assets may not exist.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

3. Use the resulting $N \times N$ matrix with **k-medoids**, exactly as before.

- Different pairs may have different best lags. A single common alignment across all assets may not exist.
- A wide lag search may identify chance correlations; choose L based on the application and require adequate overlap.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

3. Use the resulting $N \times N$ matrix with **k-medoids**, exactly as before.

- Different pairs may have different best lags. A single common alignment across all assets may not exist.
- A wide lag search may identify chance correlations; choose L based on the application and require adequate overlap.
- $1 - \max \widehat{\rho}$ is a useful *dissimilarity*, but need not satisfy every mathematical property of a distance metric.

Clustering with Lagged Correlations

Q: What changes relative to contemporaneous correlation clustering?

1. For each pair (i, j), calculate correlations across allowable leads and lags.
2. Retain the largest correlation and convert it to a dissimilarity:

$$d_{ij}^{\text{lag}} = 1 - \max_{|\ell| \leq L} \widehat{\rho}_{ij}(\ell).$$

3. Use the resulting $N \times N$ matrix with **k-medoids**, exactly as before.

- Different pairs may have different best lags. A single common alignment across all assets may not exist.
- A wide lag search may identify chance correlations; choose L based on the application and require adequate overlap.
- $1 - \max \widehat{\rho}$ is a useful *dissimilarity*, but need not satisfy every mathematical property of a distance metric.

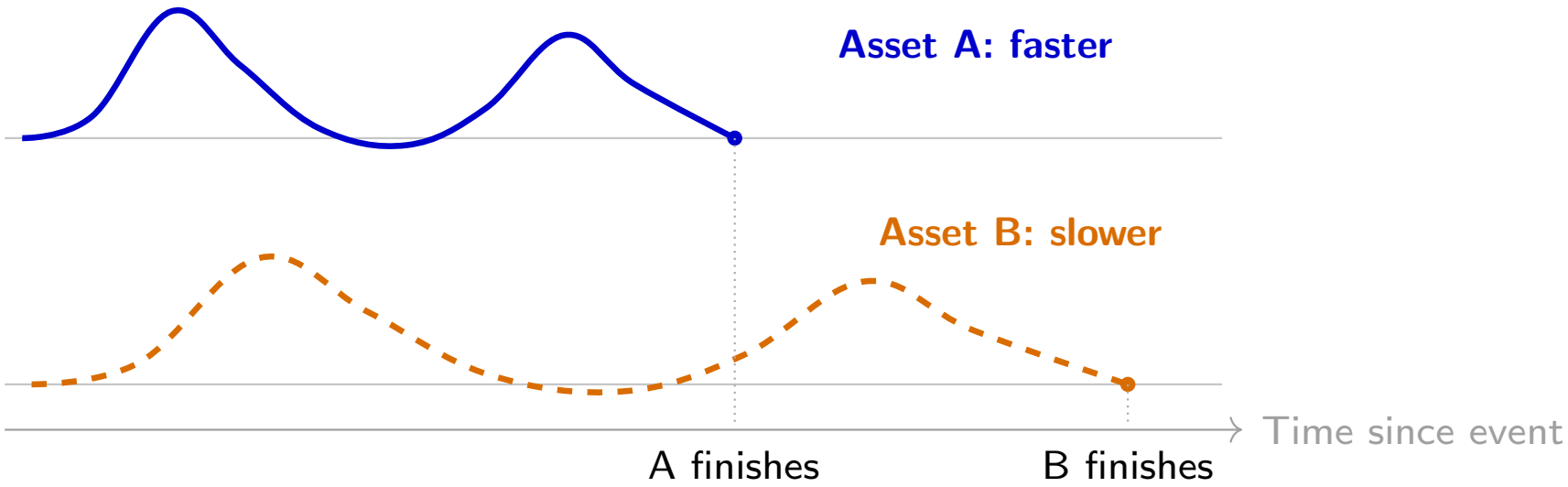
Takeaway: We changed the pairwise dissimilarities, not the k-medoids procedure.

When a Fixed Lag Is Not Enough

Q: What if two assets follow the same path, but one moves faster?

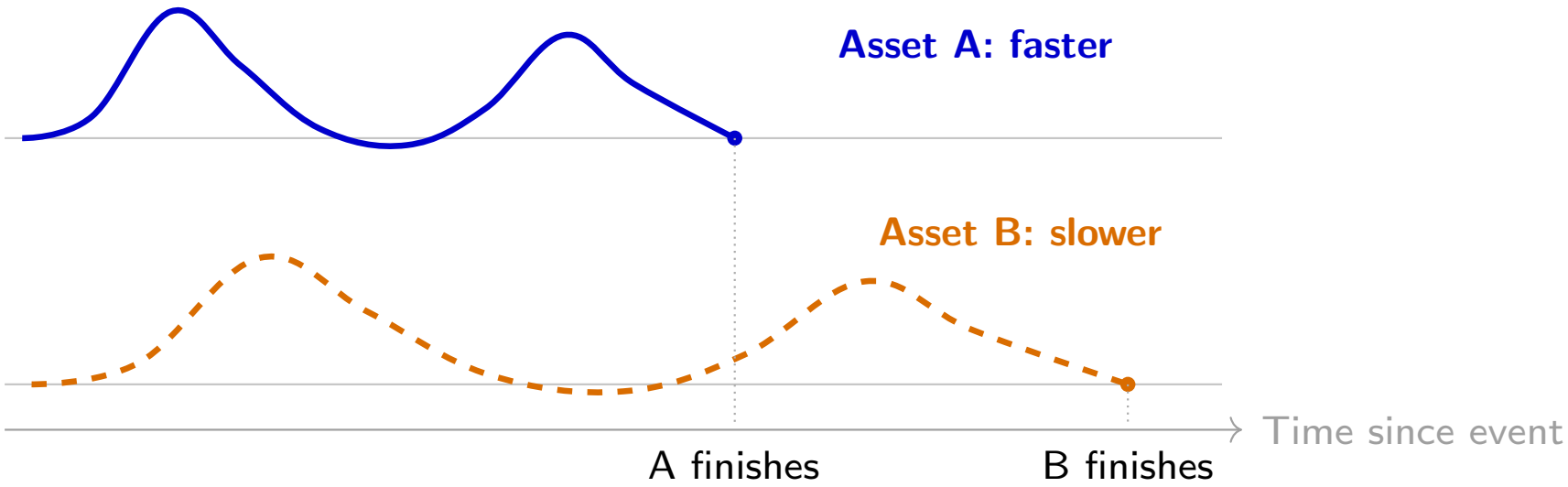
When a Fixed Lag Is Not Enough

Q: What if two assets follow the same path, but one moves faster?



When a Fixed Lag Is Not Enough

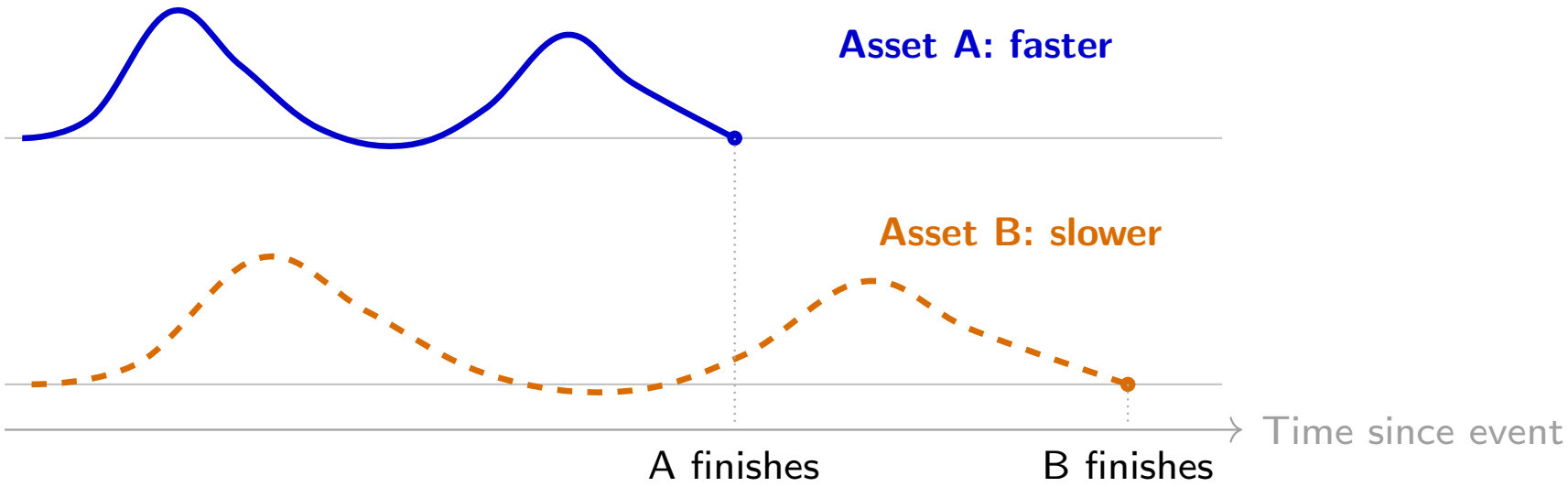
Q: What if two assets follow the same path, but one moves faster?



- Both assets follow the **same sequence of rises, falls, and recoveries**.

When a Fixed Lag Is Not Enough

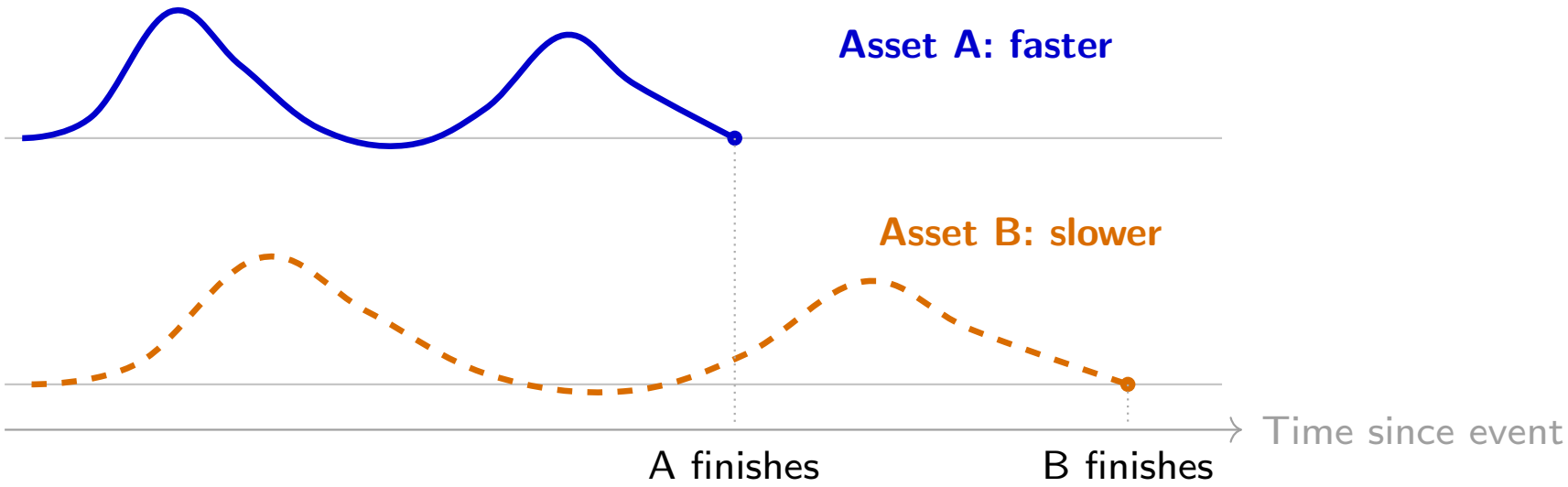
Q: What if two assets follow the same path, but one moves faster?



- Both assets follow the **same sequence of rises, falls, and recoveries**.
- Asset A completes that sequence sooner: the entire time axis is compressed.

When a Fixed Lag Is Not Enough

Q: What if two assets follow the same path, but one moves faster?



- Both assets follow the **same sequence of rises, falls, and recoveries**.
- Asset A completes that sequence sooner: the entire time axis is compressed.
- A fixed lag only *shifts* a series. It cannot *stretch or compress* time. DTW can.

Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Consider two short series:

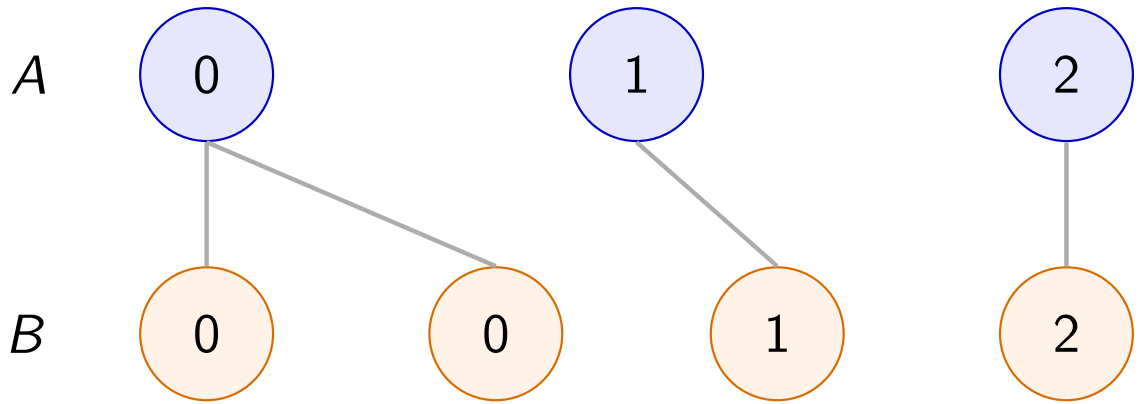
$$A = (0, 1, 2), \qquad B = (0, 0, 1, 2).$$

Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Consider two short series:

$$A = (0, 1, 2), \qquad B = (0, 0, 1, 2).$$

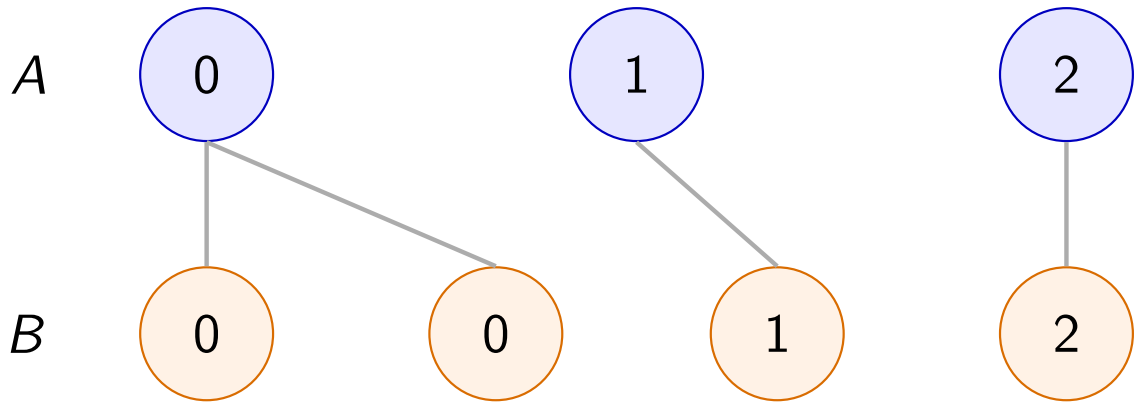


Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Consider two short series:

$$A = (0, 1, 2), \qquad B = (0, 0, 1, 2).$$



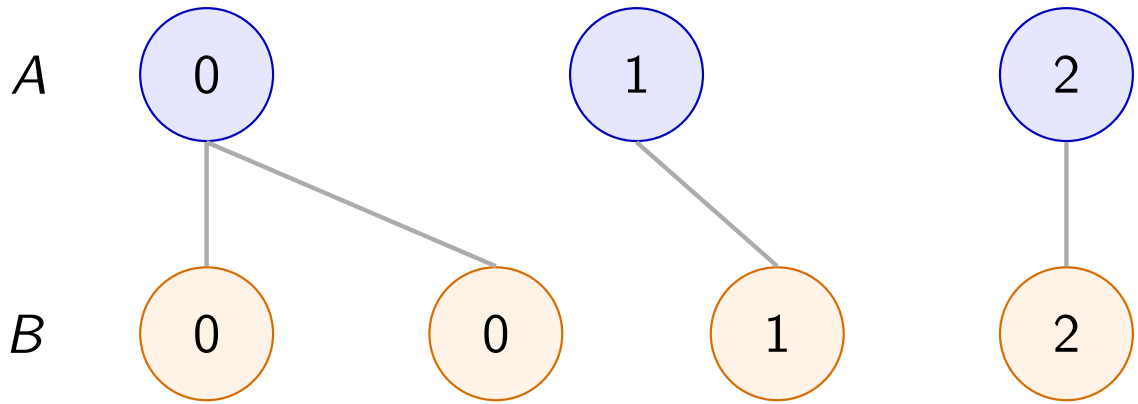
- Both sequences follow the same progression; B stays at zero for an extra observation.

Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Consider two short series:

$$A = (0, 1, 2), \qquad B = (0, 0, 1, 2).$$



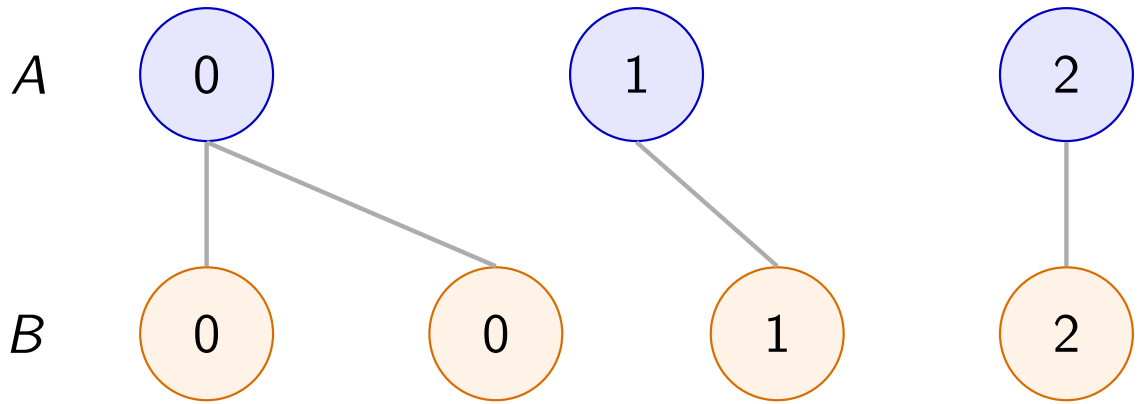
- Both sequences follow the same progression; B stays at zero for an extra observation.
- DTW can match one observation in A to multiple observations in B.

Dynamic Time Warping: Intuition

Q: How can we match observations without requiring matching dates?

Consider two short series:

$$A = (0, 1, 2), \qquad B = (0, 0, 1, 2).$$



- Both sequences follow the same progression; B stays at zero for an extra observation.
- DTW can match one observation in A to multiple observations in B.
- It finds a monotone alignment that minimizes total *value mismatch*.

DTW: The Optimization Problem

Q: What does DTW minimize?

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

- $c(a_i, b_j)$ is the mismatch between matched values (e.g., $|a_i - b_j|$ or $(a_i - b_j)^2$).

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

- $c(a_i, b_j)$ is the mismatch between matched values (e.g., $|a_i - b_j|$ or $(a_i - b_j)^2$).
- π is an alignment path from $(1, 1)$ to (T_1, T_2).

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

- $c(a_i, b_j)$ is the mismatch between matched values (e.g., $|a_i - b_j|$ or $(a_i - b_j)^2$).
- π is an alignment path from $(1, 1)$ to (T_1, T_2).
- Each step moves $(1, 0)$, $(0, 1)$, or $(1, 1)$: the path always moves forward in time, without skipping either sequence.

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

- $c(a_i, b_j)$ is the mismatch between matched values (e.g., $|a_i - b_j|$ or $(a_i - b_j)^2$).
- π is an alignment path from $(1, 1)$ to (T_1, T_2).
- Each step moves $(1, 0)$, $(0, 1)$, or $(1, 1)$: the path always moves forward in time, without skipping either sequence.
- Consequently, DTW accommodates unequal lengths and varying speeds.

DTW: The Optimization Problem

Q: What does DTW minimize?

For $A = (a_1, \ldots, a_{T_1})$ and $B = (b_1, \ldots, b_{T_2})$:

$$d^{\mathrm{DTW}}(A, B) = \min_{\pi} \sum_{(i,j)\in\pi} c(a_i, b_j)$$

- $c(a_i, b_j)$ is the mismatch between matched values (e.g., $|a_i - b_j|$ or $(a_i - b_j)^2$).
- π is an alignment path from $(1, 1)$ to (T_1, T_2).
- Each step moves $(1, 0)$, $(0, 1)$, or $(1, 1)$: the path always moves forward in time, without skipping either sequence.
- Consequently, DTW accommodates unequal lengths and varying speeds.

Important: Standard DTW penalizes *differences between matched values*, not the amount of warping itself. We may impose a maximum warping window.

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

$$G(i,j) = C(i,j) + \min \begin{cases} G(i-1,j) \\ G(i,j-1) \\ G(i-1,j-1) \end{cases}$$

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

$$G(i,j) = C(i,j) + \min \begin{cases} G(i-1,j) \\ G(i,j-1) \\ G(i-1,j-1) \end{cases}$$

- Start with $G(0,0) = 0$ and $G(i,0) = G(0,j) = \infty$ for positive indices.

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

$$G(i,j) = C(i,j) + \min \begin{cases} G(i-1,j) \\ G(i,j-1) \\ G(i-1,j-1) \end{cases}$$

- Start with $G(0,0) = 0$ and $G(i,0) = G(0,j) = \infty$ for positive indices.
- Fill the grid from the top-left to the bottom-right.

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

$$G(i,j) = C(i,j) + \min \begin{cases} G(i-1,j) \\ G(i,j-1) \\ G(i-1,j-1) \end{cases}$$

- Start with $G(0,0) = 0$ and $G(i,0) = G(0,j) = \infty$ for positive indices.
- Fill the grid from the top-left to the bottom-right.
- Final DTW cost: $G(T_1, T_2)$.

DTW: Finding the Minimum-Cost Path

Q: How does DTW find the best alignment efficiently?

Let $C(i,j) = c(a_i, b_j)$ be the *local* mismatch. Let $G(i,j)$ be the minimum *accumulated* cost of reaching (i,j).

$$G(i,j) = C(i,j) + \min \begin{cases} G(i-1,j) \\ G(i,j-1) \\ G(i-1,j-1) \end{cases}$$

- Start with $G(0,0) = 0$ and $G(i,0) = G(0,j) = \infty$ for positive indices.
- Fill the grid from the top-left to the bottom-right.
- Final DTW cost: $G(T_1, T_2)$.

This is **dynamic programming**: at each cell, use the cheapest of the three allowable preceding paths.

DTW: A Numerical Example

Q: Can two series have zero DTW cost despite different lengths?

DTW: A Numerical Example

Q: Can two series have zero DTW cost despite different lengths?

Let $A = (0, 1, 2)$, $B = (0, 0, 1, 2)$, and $C(i, j) = |a_i - b_j|$.

DTW: A Numerical Example

Q: Can two series have zero DTW cost despite different lengths?

Let $A = (0, 1, 2)$, $B = (0, 0, 1, 2)$, and $C(i, j) = |a_i - b_j|$.

$A \backslash B$	0	0	1	2
0	0	0	1	3
1	1	1	0	1
2	3	3	1	0

Accumulated cost G

Blue = optimal path

Final cost = 0

DTW: A Numerical Example

Q: Can two series have zero DTW cost despite different lengths?

Let $A = (0, 1, 2)$, $B = (0, 0, 1, 2)$, and $C(i, j) = |a_i - b_j|$.

$A \backslash B$	0	0	1	2
0	0	0	1	3
1	1	1	0	1
2	3	3	1	0

Accumulated cost G
Blue = optimal path
Final cost = 0

For example,

$$G(2, 3) = |1 - 1| + \min\{G(1, 3), G(2, 2), G(1, 2)\} = 0 + \min\{1, 1, 0\} = 0.$$

DTW: A Numerical Example

Q: Can two series have zero DTW cost despite different lengths?

Let $A = (0, 1, 2)$, $B = (0, 0, 1, 2)$, and $C(i, j) = |a_i - b_j|$.

$A \backslash B$	0	0	1	2
0	0	0	1	3
1	1	1	0	1
2	3	3	1	0

Accumulated cost G
Blue = optimal path
Final cost = 0

For example,

$$G(2, 3) = |1 - 1| + \min\{G(1, 3), G(2, 2), G(1, 2)\} = 0 + \min\{1, 1, 0\} = 0.$$

Interpretation: The extra zero in B can be matched to the first zero in A without an additional mismatch cost.

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

For N time series, compute the $N \times N$ dissimilarity matrix:

$$D = \begin{bmatrix} 0 & d_{12}^{\mathrm{DTW}} & \cdots & d_{1N}^{\mathrm{DTW}} \\ d_{21}^{\mathrm{DTW}} & 0 & \cdots & d_{2N}^{\mathrm{DTW}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{\mathrm{DTW}} & d_{N2}^{\mathrm{DTW}} & \cdots & 0 \end{bmatrix}.$$

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

For N time series, compute the $N \times N$ dissimilarity matrix:

$$D = \begin{bmatrix} 0 & d_{12}^{\mathrm{DTW}} & \cdots & d_{1N}^{\mathrm{DTW}} \\ d_{21}^{\mathrm{DTW}} & 0 & \cdots & d_{2N}^{\mathrm{DTW}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{\mathrm{DTW}} & d_{N2}^{\mathrm{DTW}} & \cdots & 0 \end{bmatrix}.$$

1. Select K initial medoids.

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

For N time series, compute the $N \times N$ dissimilarity matrix:

$$D = \begin{bmatrix} 0 & d_{12}^{\mathrm{DTW}} & \cdots & d_{1N}^{\mathrm{DTW}} \\ d_{21}^{\mathrm{DTW}} & 0 & \cdots & d_{2N}^{\mathrm{DTW}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{\mathrm{DTW}} & d_{N2}^{\mathrm{DTW}} & \cdots & 0 \end{bmatrix}.$$

1. Select K initial medoids.
2. Assign each entire series to its nearest medoid according to DTW cost.

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

For N time series, compute the $N \times N$ dissimilarity matrix:

$$D = \begin{bmatrix} 0 & d_{12}^{\mathrm{DTW}} & \cdots & d_{1N}^{\mathrm{DTW}} \\ d_{21}^{\mathrm{DTW}} & 0 & \cdots & d_{2N}^{\mathrm{DTW}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{\mathrm{DTW}} & d_{N2}^{\mathrm{DTW}} & \cdots & 0 \end{bmatrix}.$$

1. Select K initial medoids.
2. Assign each entire series to its nearest medoid according to DTW cost.
3. Update medoids and repeat until convergence.

Clustering with DTW Distances

Q: What do we do after computing DTW for every pair of series?

For N time series, compute the $N \times N$ dissimilarity matrix:

$$D = \begin{bmatrix} 0 & d_{12}^{\mathrm{DTW}} & \cdots & d_{1N}^{\mathrm{DTW}} \\ d_{21}^{\mathrm{DTW}} & 0 & \cdots & d_{2N}^{\mathrm{DTW}} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1}^{\mathrm{DTW}} & d_{N2}^{\mathrm{DTW}} & \cdots & 0 \end{bmatrix}.$$

1. Select K initial medoids.
2. Assign each entire series to its nearest medoid according to DTW cost.
3. Update medoids and repeat until convergence.

What changed? The *distance measure*, not the basic k-medoids logic.

Alternatives: hierarchical clustering; specialized centroid-based methods using DTW barycenters (DBA). k-Shape is a related shape-based method built around normalized cross-correlation.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.
- Comparable demand or customer-response trajectories of unequal length.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.
- Comparable demand or customer-response trajectories of unequal length.

Important choices

- **Preprocessing:** Raw levels, growth rates, returns, and standardized values capture different notions of similarity.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.
- Comparable demand or customer-response trajectories of unequal length.

Important choices

- **Preprocessing:** Raw levels, growth rates, returns, and standardized values capture different notions of similarity.
- **Warping limits:** Constrain alignments when very distant dates should not match.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.
- Comparable demand or customer-response trajectories of unequal length.

Important choices

- **Preprocessing:** Raw levels, growth rates, returns, and standardized values capture different notions of similarity.
- **Warping limits:** Constrain alignments when very distant dates should not match.
- **Length and computation:** For unequal lengths, consider cost normalization; all-pairs DTW can be expensive.

When Is DTW Useful — and When Is It Misleading?

Q: Which research questions justify flexible alignment?

Potential applications

- Similar economic cycles across countries that unfold at different speeds.
- Similar responses to events with different adjustment speeds.
- Comparable demand or customer-response trajectories of unequal length.

Important choices

- **Preprocessing:** Raw levels, growth rates, returns, and standardized values capture different notions of similarity.
- **Warping limits:** Constrain alignments when very distant dates should not match.
- **Length and computation:** For unequal lengths, consider cost normalization; all-pairs DTW can be expensive.
- **Interpretation:** If the exact event date matters, allowing flexible timing may erase economically important differences.

Other Time-Series Clustering Applications

Q: Must we cluster the observed trajectories themselves?

Other Time-Series Clustering Applications

Q: Must we cluster the observed trajectories themselves?

No. We can apply familiar clustering algorithms to time-series-specific inputs.

Other Time-Series Clustering Applications

Q: Must we cluster the observed trajectories themselves?

No. We can apply familiar clustering algorithms to time-series-specific inputs.

Feature-based clustering

Extract time-series properties, such as autocorrelations, volatility, volatility persistence, or seasonal strength. Standardize the resulting features across series, then apply k-means or another familiar method.

Other Time-Series Clustering Applications

Q: Must we cluster the observed trajectories themselves?

No. We can apply familiar clustering algorithms to time-series-specific inputs.

Feature-based clustering

Extract time-series properties, such as autocorrelations, volatility, volatility persistence, or seasonal strength. Standardize the resulting features across series, then apply k-means or another familiar method.

Model-based clustering

Fit a time-series model to each series; cluster estimated parameters (e.g., AR coefficients and innovation volatility), or use a model-based mixture approach.

Other Time-Series Clustering Applications

Q: Must we cluster the observed trajectories themselves?

No. We can apply familiar clustering algorithms to time-series-specific inputs.

Feature-based clustering

Extract time-series properties, such as autocorrelations, volatility, volatility persistence, or seasonal strength. Standardize the resulting features across series, then apply k-means or another familiar method.

Model-based clustering

Fit a time-series model to each series; cluster estimated parameters (e.g., AR coefficients and innovation volatility), or use a model-based mixture approach.

Example: Two assets can have weak return correlation but similar persistence or volatility dynamics. They may belong together under a feature- or model-based definition of similarity.

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

- **Across entities:** Cluster assets, countries, industries, or customers by their full series or time-series properties.

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

- **Across entities:** Cluster assets, countries, industries, or customers by their full series or time-series properties.
- **Across periods:** Cluster rolling windows of observations or window-level features to identify similar historical episodes.

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

- **Across entities:** Cluster assets, countries, industries, or customers by their full series or time-series properties.
- **Across periods:** Cluster rolling windows of observations or window-level features to identify similar historical episodes.

Financial application: Cluster 60-day market windows using realized volatility, cross-asset correlation, and credit-spread changes to explore possible market regimes.

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

- **Across entities:** Cluster assets, countries, industries, or customers by their full series or time-series properties.
- **Across periods:** Cluster rolling windows of observations or window-level features to identify similar historical episodes.

Financial application: Cluster 60-day market windows using realized volatility, cross-asset correlation, and credit-spread changes to explore possible market regimes. Hidden Markov models provide a related, explicitly temporal approach to persistent latent regimes, but need not be retaught here.

What Are We Clustering?

Q: Must each observation be an entire asset or economic series?

- **Across entities:** Cluster assets, countries, industries, or customers by their full series or time-series properties.
- **Across periods:** Cluster rolling windows of observations or window-level features to identify similar historical episodes.

Financial application: Cluster 60-day market windows using realized volatility, cross-asset correlation, and credit-spread changes to explore possible market regimes. Hidden Markov models provide a related, explicitly temporal approach to persistent latent regimes, but need not be retaught here. *For real-time applications, estimate features and assignments using only information available at the decision date.*

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.
- **Follow one another with a stable lead/lag?**
  Lag-optimized correlation dissimilarities $\rightarrow$ **k-medoids**.

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.
- **Follow one another with a stable lead/lag?**
  Lag-optimized correlation dissimilarities $\rightarrow$ **k-medoids**.
- **Follow the same shape at different speeds?**
  DTW dissimilarities $\rightarrow$ **k-medoids**.

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.
- **Follow one another with a stable lead/lag?**
  Lag-optimized correlation dissimilarities $\rightarrow$ **k-medoids**.
- **Follow the same shape at different speeds?**
  DTW dissimilarities $\rightarrow$ **k-medoids**.
- **Exhibit similar volatility or persistence?**
  Time-series features $\rightarrow$ familiar clustering algorithms.

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.
- **Follow one another with a stable lead/lag?**
  Lag-optimized correlation dissimilarities $\rightarrow$ **k-medoids**.
- **Follow the same shape at different speeds?**
  DTW dissimilarities $\rightarrow$ **k-medoids**.
- **Exhibit similar volatility or persistence?**
  Time-series features $\rightarrow$ familiar clustering algorithms.
- **Follow similar estimated dynamics?**
  Model parameters or model-based clustering.

Choosing a Time-Series Clustering Approach

Q: Which notion of similarity matches our economic question?

- **Move together on the same dates?**
  Pairwise contemporaneous correlations $\rightarrow$ **k-medoids**.
- **Follow one another with a stable lead/lag?**
  Lag-optimized correlation dissimilarities $\rightarrow$ **k-medoids**.
- **Follow the same shape at different speeds?**
  DTW dissimilarities $\rightarrow$ **k-medoids**.
- **Exhibit similar volatility or persistence?**
  Time-series features $\rightarrow$ familiar clustering algorithms.
- **Follow similar estimated dynamics?**
  Model parameters or model-based clustering.

Takeaway: First decide what makes series similar; then cluster the assets using a compatible algorithm.

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

Dimension Reduction in Time Series

Q: Why reduce dimension when working with time series?

Dimension Reduction in Time Series

Q: Why reduce dimension when working with time series?

We often construct many related signals:

- Momentum over multiple horizons
- Macroeconomic indicators (e.g., FRED panels)
- Technical indicators (moving averages, volatility, etc.)

Dimension Reduction in Time Series

Q: Why reduce dimension when working with time series?

We often construct many related signals:

- Momentum over multiple horizons
- Macroeconomic indicators (e.g., FRED panels)
- Technical indicators (moving averages, volatility, etc.)

These signals may be correlated, noisy, or redundant.

Dimension Reduction in Time Series

Q: Why reduce dimension when working with time series?

We often construct many related signals:

- Momentum over multiple horizons
- Macroeconomic indicators (e.g., FRED panels)
- Technical indicators (moving averages, volatility, etc.)

These signals may be correlated, noisy, or redundant.

Goals:

- Condense related features into a few informative factors.
- Summarize the historical behavior of a series.
- Identify common drivers of multiple evolving series.

Three Dimension-Reduction Problems

Q: What exactly are we trying to compress?

Three Dimension-Reduction Problems

Q: What exactly are we trying to compress?

1. Features at each date

- Many momentum horizons $\longrightarrow$ a few scores per asset and date.
- Methods: PCA; standard autoencoder.

Three Dimension-Reduction Problems

Q: What exactly are we trying to compress?

1. **Features at each date**
   - Many momentum horizons $\longrightarrow$ a few scores per asset and date.
   - Methods: PCA; standard autoencoder.
2. **A historical sequence**
   - A window of returns $\longrightarrow$ a compact description of its pattern.
   - Methods: lagged PCA / SSA; sequence autoencoder; DTW + embedding

Three Dimension-Reduction Problems

Q: What exactly are we trying to compress?

1. **Features at each date**
   - Many momentum horizons $\longrightarrow$ a few scores per asset and date.
   - Methods: PCA; standard autoencoder.
2. **A historical sequence**
   - A window of returns $\longrightarrow$ a compact description of its pattern.
   - Methods: lagged PCA / SSA; sequence autoencoder; DTW + embedding
3. **Multiple series evolving together**
   - A panel of asset returns $\longrightarrow$ a few common factor series.
   - Method: dynamic factor model (DFM).

Example: Momentum Signals

Q: Can we summarize momentum measured over several horizons?

Example: Momentum Signals

Q: Can we summarize momentum measured over several horizons?

For asset i at date t, collect $p = 4$ return horizons:

$$x_{i,t} = \begin{pmatrix} R_{i,t}^{(1m)} \\ R_{i,t}^{(3m)} \\ R_{i,t}^{(6m)} \\ R_{i,t}^{(12m)} \end{pmatrix} \in \mathbb{R}^p.$$

Example: Momentum Signals

Q: Can we summarize momentum measured over several horizons?

For asset i at date t, collect $p = 4$ return horizons:

$$x_{i,t} = \begin{pmatrix} R_{i,t}^{(1m)} \\ R_{i,t}^{(3m)} \\ R_{i,t}^{(6m)} \\ R_{i,t}^{(12m)} \end{pmatrix} \in \mathbb{R}^p.$$

- The horizons overlap, so their signals may share information.
- They can also differ: short- and long-horizon momentum need not agree.

Example: Momentum Signals

Q: Can we summarize momentum measured over several horizons?

For asset i at date t, collect $p = 4$ return horizons:

$$x_{i,t} = \begin{pmatrix} R_{i,t}^{(1m)} \\ R_{i,t}^{(3m)} \\ R_{i,t}^{(6m)} \\ R_{i,t}^{(12m)} \end{pmatrix} \in \mathbb{R}^p.$$

- The horizons overlap, so their signals may share information.
- They can also differ: short- and long-horizon momentum need not agree.

Goal: Produce a low-dimensional score, such as $z_{i1,t}$, that captures important common variation across horizons.

Rolling PCA: Moving the Window

Q: Which observations enter PCA as we move through time?

Rolling PCA: Moving the Window

Q: Which observations enter PCA as we move through time?

At each date s, $R_s \in \mathbb{R}^{N \times p}$ contains the p features for N assets. Let W be the number of dates in the estimation window.

Rolling PCA: Moving the Window

Q: Which observations enter PCA as we move through time?

At each date s, $R_s \in \mathbb{R}^{N \times p}$ contains the p features for N assets. Let W be the number of dates in the estimation window.

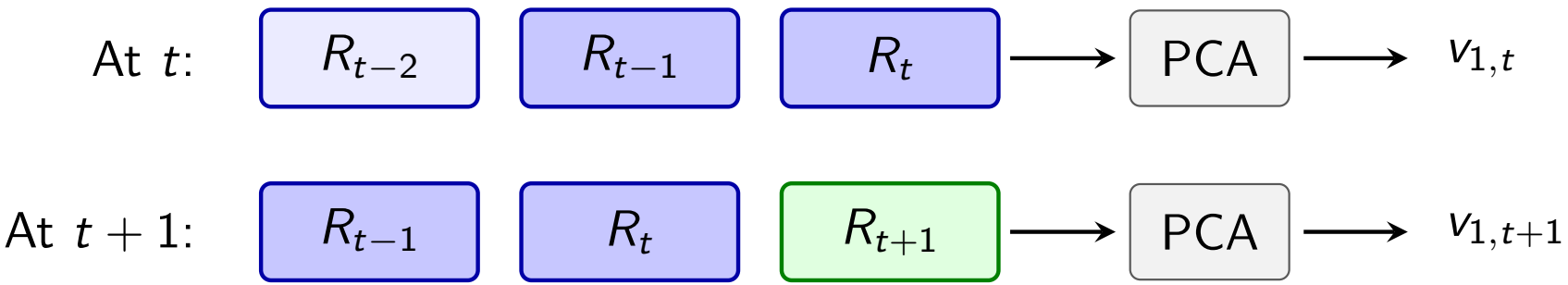


Illustration: $W = 3$ dates. Each box contains N asset observations.

Rolling PCA: Moving the Window

Q: Which observations enter PCA as we move through time?

At each date s, $R_s \in \mathbb{R}^{N \times p}$ contains the p features for N assets. Let W be the number of dates in the estimation window.

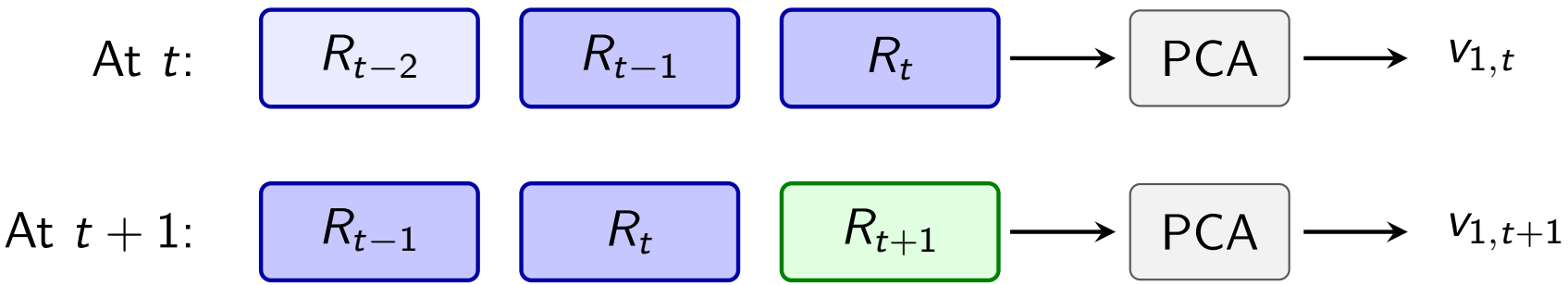


Illustration: $W = 3$ dates. Each box contains N asset observations.

- $W = 1$: Use only R_t: cross-sectional PCA at date t.

Rolling PCA: Moving the Window

Q: Which observations enter PCA as we move through time?

At each date s, $R_s \in \mathbb{R}^{N \times p}$ contains the p features for N assets. Let W be the number of dates in the estimation window.

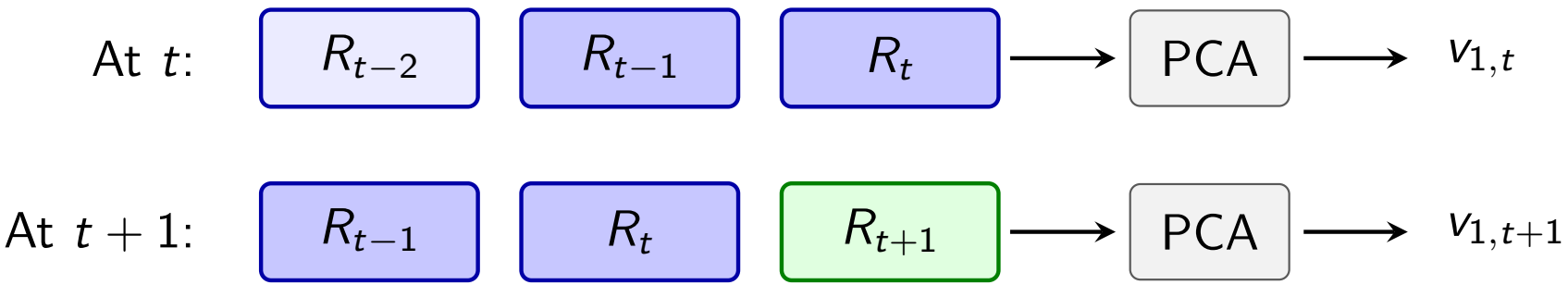


Illustration: $W = 3$ dates. Each box contains N asset observations.

- $W = 1$: Use only R_t: cross-sectional PCA at date t.
- $W > 1$: Pool W dates. At $t+1$, drop the oldest date, add the newest, and reestimate PCA.

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

1. At date t, stack $R_{t-W+1}, \ldots, R_t$ into $D_t \in \mathbb{R}^{NW \times p}$.

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

1. At date t, stack $R_{t-W+1}, \ldots, R_t$ into $D_t \in \mathbb{R}^{NW \times p}$.
2. Center each column of D_t using its window mean $\bar{x}_t$, giving $\widetilde{D}_t$. Estimate:

$$\Sigma_t = \frac{1}{NW - 1} \widetilde{D}_t^\top \widetilde{D}_t.$$

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

1. At date t, stack $R_{t-W+1}, \ldots, R_t$ into $D_t \in \mathbb{R}^{NW \times p}$.
2. Center each column of D_t using its window mean $\bar{x}_t$, giving $\widetilde{D}_t$. Estimate:

$$\Sigma_t = \frac{1}{NW - 1} \widetilde{D}_t^\top \widetilde{D}_t.$$

3. Find the first loading and compute each asset's current score:

$$\Sigma_t v_{1,t} = \lambda_{1,t} v_{1,t}, \qquad z_{i1,t} = (x_{i,t} - \bar{x}_t)^\top v_{1,t}.$$

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

1. At date t, stack $R_{t-W+1}, \ldots, R_t$ into $D_t \in \mathbb{R}^{NW \times p}$.
2. Center each column of D_t using its window mean $\bar{x}_t$, giving $\widetilde{D}_t$. Estimate:

$$\Sigma_t = \frac{1}{NW - 1} \widetilde{D}_t^\top \widetilde{D}_t.$$

3. Find the first loading and compute each asset's current score:

$$\Sigma_t v_{1,t} = \lambda_{1,t} v_{1,t}, \qquad z_{i1,t} = (x_{i,t} - \bar{x}_t)^\top v_{1,t}.$$

Limitation: Reestimating PCA updates the feature covariance structure, but does *not* explicitly model how scores evolve from $t-1$ to t.

Rolling PCA: Estimation and Interpretation

Q: How do we obtain loadings and scores in each window?

1. At date t, stack $R_{t-W+1}, \ldots, R_t$ into $D_t \in \mathbb{R}^{NW \times p}$.
2. Center each column of D_t using its window mean $\bar{x}_t$, giving $\widetilde{D}_t$. Estimate:

$$\Sigma_t = \frac{1}{NW - 1} \widetilde{D}_t^\top \widetilde{D}_t.$$

3. Find the first loading and compute each asset's current score:

$$\Sigma_t v_{1,t} = \lambda_{1,t} v_{1,t}, \qquad z_{i1,t} = (x_{i,t} - \bar{x}_t)^\top v_{1,t}.$$

Limitation: Reestimating PCA updates the feature covariance structure, but does *not* explicitly model how scores evolve from $t-1$ to t.

Practical note: Eigenvectors may switch signs across windows without changing the component; align signs before comparing scores.

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Instead of using only today's features, construct a vector of recent returns for asset i:

$$q_{i,t} = \begin{pmatrix} r_{i,t} & r_{i,t-1} & \cdots & r_{i,t-L+1} \end{pmatrix}^\top \in \mathbb{R}^L.$$

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Instead of using only today's features, construct a vector of recent returns for asset i:

$$q_{i,t} = \begin{pmatrix} r_{i,t} & r_{i,t-1} & \cdots & r_{i,t-L+1} \end{pmatrix}^\top \in \mathbb{R}^L.$$

- Stack overlapping windows $q_{i,t}$ across dates and apply PCA.

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Instead of using only today's features, construct a vector of recent returns for asset i:

$$q_{i,t} = \begin{pmatrix} r_{i,t} & r_{i,t-1} & \cdots & r_{i,t-L+1} \end{pmatrix}^\top \in \mathbb{R}^L.$$

- Stack overlapping windows $q_{i,t}$ across dates and apply PCA.
- The covariance now includes relationships across *lags*, allowing components to reflect temporal patterns.

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Instead of using only today's features, construct a vector of recent returns for asset i:

$$q_{i,t} = \begin{pmatrix} r_{i,t} & r_{i,t-1} & \cdots & r_{i,t-L+1} \end{pmatrix}^\top \in \mathbb{R}^L.$$

- Stack overlapping windows $q_{i,t}$ across dates and apply PCA.
- The covariance now includes relationships across *lags*, allowing components to reflect temporal patterns.
- This is closely related to **singular spectrum analysis (SSA)**, which also reconstructs components on the original time axis.

Lagged PCA / Singular Spectrum Analysis

Q: What if the temporal pattern itself is what matters?

Instead of using only today's features, construct a vector of recent returns for asset i:

$$q_{i,t} = \begin{pmatrix} r_{i,t} & r_{i,t-1} & \cdots & r_{i,t-L+1} \end{pmatrix}^\top \in \mathbb{R}^L.$$

- Stack overlapping windows $q_{i,t}$ across dates and apply PCA.
- The covariance now includes relationships across *lags*, allowing components to reflect temporal patterns.
- This is closely related to **singular spectrum analysis (SSA)**, which also reconstructs components on the original time axis.

Contrast: Ordinary feature PCA compresses related indicators; lagged PCA compresses information from a historical window.

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Use the same input $x_{i,t} \in \mathbb{R}^p$ as in PCA:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^k \xrightarrow{g_\phi} \widehat{x}_{i,t} \in \mathbb{R}^p, \qquad k < p.$$

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Use the same input $x_{i,t} \in \mathbb{R}^p$ as in PCA:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^k \xrightarrow{g_\phi} \widehat{x}_{i,t} \in \mathbb{R}^p, \qquad k < p.$$

- f_θ: encoder; $z_{i,t}$: compressed representation.

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Use the same input $x_{i,t} \in \mathbb{R}^p$ as in PCA:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^k \xrightarrow{g_\phi} \widehat{x}_{i,t} \in \mathbb{R}^p, \qquad k < p.$$

- f_θ: encoder; $z_{i,t}$: compressed representation.
- g_ϕ: decoder; $\widehat{x}_{i,t}$: reconstruction.

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Use the same input $x_{i,t} \in \mathbb{R}^p$ as in PCA:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^k \xrightarrow{g_\phi} \widehat{x}_{i,t} \in \mathbb{R}^p, \qquad k < p.$$

- f_θ: encoder; $z_{i,t}$: compressed representation.
- g_ϕ: decoder; $\widehat{x}_{i,t}$: reconstruction.
- Nonlinear activations permit nonlinear compression.

Autoencoders: Feature Compression

Q: What if the relationships among momentum horizons are nonlinear?

Use the same input $x_{i,t} \in \mathbb{R}^p$ as in PCA:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^k \xrightarrow{g_\phi} \widehat{x}_{i,t} \in \mathbb{R}^p, \qquad k < p.$$

- f_θ: encoder; $z_{i,t}$: compressed representation.
- g_ϕ: decoder; $\widehat{x}_{i,t}$: reconstruction.
- Nonlinear activations permit nonlinear compression.

Recall: Under suitable conditions, a linear autoencoder with squared-error loss recovers the PCA reconstruction subspace.

Autoencoder Interpretation

Q: How is it trained, and what time structure does it use?

Autoencoder Interpretation

Q: How is it trained, and what time structure does it use?

Estimate shared encoder and decoder parameters using many asset-date observations $\mathcal{D}$:

$$\min_{\theta,\phi} \sum_{(i,s)\in\mathcal{D}} \left\| x_{i,s} - g_\phi\left(f_\theta(x_{i,s})\right) \right\|^2 .$$

Autoencoder Interpretation

Q: How is it trained, and what time structure does it use?

Estimate shared encoder and decoder parameters using many asset-date observations $\mathcal{D}$:

$$\min_{\theta,\phi} \sum_{(i,s)\in\mathcal{D}} \left\| x_{i,s} - g_\phi\left(f_\theta(x_{i,s})\right) \right\|^2 .$$

- The same learned parameters are used across assets and dates.

Autoencoder Interpretation

Q: How is it trained, and what time structure does it use?

Estimate shared encoder and decoder parameters using many asset-date observations $\mathcal{D}$:

$$\min_{\theta,\phi} \sum_{(i,s)\in\mathcal{D}} \left\| x_{i,s} - g_\phi\left(f_\theta(x_{i,s})\right) \right\|^2 .$$

- The same learned parameters are used across assets and dates.
- But the code $z_{i,t} = f_\theta(x_{i,t})$ uses the feature vector *at date* t, not an explicit history of vectors.

Autoencoder Interpretation

Q: How is it trained, and what time structure does it use?

Estimate shared encoder and decoder parameters using many asset-date observations $\mathcal{D}$:

$$\min_{\theta,\phi} \sum_{(i,s)\in\mathcal{D}} \left\| x_{i,s} - g_\phi\left(f_\theta(x_{i,s})\right) \right\|^2 .$$

- The same learned parameters are used across assets and dates.
- But the code $z_{i,t} = f_\theta(x_{i,t})$ uses the feature vector *at date* t, not an explicit history of vectors.

Key distinction: A standard autoencoder can compress features nonlinearly, but its architecture does not explicitly model their sequential evolution.

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).
For each asset i, encode the full history:

$$(r_{i,1}, \ldots, r_{i,T}) \xrightarrow{f_\theta} z_i \in \mathbb{R}^k, \qquad T \approx 2500, \quad k = 5.$$

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).
For each asset i, encode the full history:

$$(r_{i,1}, \ldots, r_{i,T}) \xrightarrow{f_\theta} z_i \in \mathbb{R}^k, \qquad T \approx 2500, \quad k = 5.$$

What does $k = 5$ mean?

- Each asset's 2,500 returns become five **learned features**, not five selected dates.

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).
For each asset i, encode the full history:

$$(r_{i,1}, \ldots, r_{i,T}) \xrightarrow{f_\theta} z_i \in \mathbb{R}^k, \qquad T \approx 2500, \quad k = 5.$$

What does $k = 5$ mean?

- Each asset's 2,500 returns become five **learned features**, not five selected dates.
- These features summarize patterns in the return history that help the decoder reconstruct the sequence.

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).
For each asset i, encode the full history:

$$(r_{i,1}, \ldots, r_{i,T}) \xrightarrow{f_\theta} z_i \in \mathbb{R}^k, \qquad T \approx 2500, \quad k = 5.$$

What does $k = 5$ mean?

- Each asset's 2,500 returns become five **learned features**, not five selected dates.
- These features summarize patterns in the return history that help the decoder reconstruct the sequence.

Result:

- 500 historical return series $\rightarrow$ 500 five-dimensional vectors.
- Applications: cluster historical return paths; identify similarities in trends or volatility behavior.

Sequence Autoencoders: Motivation

Q: Can we compress an entire return trajectory?

Example: 500 assets, 10 years of daily returns (about 2,500 dates).
For each asset i, encode the full history:

$$(r_{i,1}, \ldots, r_{i,T}) \xrightarrow{f_\theta} z_i \in \mathbb{R}^k, \qquad T \approx 2500, \quad k = 5.$$

What does $k = 5$ mean?

- Each asset's 2,500 returns become five **learned features**, not five selected dates.
- These features summarize patterns in the return history that help the decoder reconstruct the sequence.

Result:

- 500 historical return series $\rightarrow$ 500 five-dimensional vectors.
- Applications: cluster historical return paths; identify similarities in trends or volatility behavior.

Limitation for forecasting: A full-history embedding summarizes that completed sample; it does not provide a historical real-time signal at earlier dates.

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Rather than encode all 10 years, use a rolling window of $L = 60$ returns:

$$q_{i,t} = (r_{i,t-59}, \ldots, r_{i,t})^\top \in \mathbb{R}^{60}.$$

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Rather than encode all 10 years, use a rolling window of $L = 60$ returns:

$$q_{i,t} = (r_{i,t-59}, \ldots, r_{i,t})^\top \in \mathbb{R}^{60}.$$

Encode the recent sequence:

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^5.$$

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Rather than encode all 10 years, use a rolling window of $L = 60$ returns:

$$q_{i,t} = (r_{i,t-59}, \ldots, r_{i,t})^\top \in \mathbb{R}^{60}.$$

Encode the recent sequence:

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^5.$$

- Each asset now has a five-dimensional code at every eligible date.

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Rather than encode all 10 years, use a rolling window of $L = 60$ returns:

$$q_{i,t} = (r_{i,t-59}, \ldots, r_{i,t})^\top \in \mathbb{R}^{60}.$$

Encode the recent sequence:

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^5.$$

- Each asset now has a five-dimensional code at every eligible date.
- Applications: cluster recent behavior, identify changing patterns, or provide features to a forecasting model.

Sequence Autoencoders: Rolling Windows

Q: How can an asset's compressed representation evolve over time?

Rather than encode all 10 years, use a rolling window of $L = 60$ returns:

$$q_{i,t} = (r_{i,t-59}, \ldots, r_{i,t})^\top \in \mathbb{R}^{60}.$$

Encode the recent sequence:

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \in \mathbb{R}^5.$$

- Each asset now has a five-dimensional code at every eligible date.
- Applications: cluster recent behavior, identify changing patterns, or provide features to a forecasting model.
- Training windows and transformations must exclude information unavailable at the date the code is used.

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence autoencoder: Instead, the input is an ordered history of returns:

$$q_{i,t} = (r_{i,t-L+1}, \ldots, r_{i,t}),$$

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{q}_{i,t}.$$

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence autoencoder: Instead, the input is an ordered history of returns:

$$q_{i,t} = (r_{i,t-L+1}, \ldots, r_{i,t}),$$

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{q}_{i,t}.$$

- The encoder learns a compact representation of patterns across dates, rather than features at one date.

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence autoencoder: Instead, the input is an ordered history of returns:

$$q_{i,t} = (r_{i,t-L+1}, \ldots, r_{i,t}),$$

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{q}_{i,t}.$$

- The encoder learns a compact representation of patterns across dates, rather than features at one date.
- To exploit temporal relationships, the encoder and decoder can use sequence-aware architectures (e.g., RNNs, temporal convolutions, or Transformers).

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence autoencoder: Instead, the input is an ordered history of returns:

$$q_{i,t} = (r_{i,t-L+1}, \ldots, r_{i,t}),$$

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{q}_{i,t}.$$

- The encoder learns a compact representation of patterns across dates, rather than features at one date.
- To exploit temporal relationships, the encoder and decoder can use sequence-aware architectures (e.g., RNNs, temporal convolutions, or Transformers).
- The same network is trained on many sequences across dates and assets, learning shared parameters.

Sequence Autoencoders: Interpretation

Q: What distinguishes a sequence autoencoder from a standard one?

Recall: A standard autoencoder compresses features observed at one date:

$$x_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{x}_{i,t}.$$

Sequence autoencoder: Instead, the input is an ordered history of returns:

$$q_{i,t} = (r_{i,t-L+1}, \ldots, r_{i,t}),$$

$$q_{i,t} \xrightarrow{f_\theta} z_{i,t} \xrightarrow{g_\phi} \hat{q}_{i,t}.$$

- The encoder learns a compact representation of patterns across dates, rather than features at one date.
- To exploit temporal relationships, the encoder and decoder can use sequence-aware architectures (e.g., RNNs, temporal convolutions, or Transformers).
- The same network is trained on many sequences across dates and assets, learning shared parameters.

Limitation: Encoding assets separately does not explicitly model their contemporaneous comovement. A multivariate sequence autoencoder can do both.

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?

Solution: PCA or Standard Autoencoder

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?
Solution: PCA or Standard Autoencoder

Problem 2: How can we summarize the historical evolution of an asset's returns using a few features?

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?
Solution: PCA or Standard Autoencoder

Problem 2: How can we summarize the historical evolution of an asset's returns using a few features?
Solution: Sequence Autoencoder

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?
Solution: PCA or Standard Autoencoder

Problem 2: How can we summarize the historical evolution of an asset's returns using a few features?
Solution: Sequence Autoencoder

Problem 3: How can we identify a few common factors driving returns across assets and explicitly model how those factors evolve?

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?
Solution: PCA or Standard Autoencoder

Problem 2: How can we summarize the historical evolution of an asset's returns using a few features?
Solution: Sequence Autoencoder

Problem 3: How can we identify a few common factors driving returns across assets and explicitly model how those factors evolve?
Solution: Dynamic Factor Model

Dynamic Factor Models: Motivation

Q: What dimension-reduction problem are we trying to solve?

Problem 1: How can we summarize many correlated features using a few components?
Solution: PCA or Standard Autoencoder

Problem 2: How can we summarize the historical evolution of an asset's returns using a few features?
Solution: Sequence Autoencoder

Problem 3: How can we identify a few common factors driving returns across assets and explicitly model how those factors evolve?
Solution: Dynamic Factor Model

Connection: PCA can identify common factors across assets. A DFM additionally specifies how those factors evolve over time.

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Observed returns and common factors:

$$\underbrace{r_t = \Lambda f_t + \epsilon_t}_{\text{Common-factor representation}} \quad , \qquad \begin{aligned} & r_t \in \mathbb{R}^N, \quad f_t \in \mathbb{R}^k, \\ & \Lambda \in \mathbb{R}^{N \times k}, \quad k \ll N. \end{aligned}$$

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Observed returns and common factors:

$$\underbrace{r_t = \Lambda f_t + \epsilon_t}_{\text{Common-factor representation}} \quad , \qquad \begin{aligned} r_t &\in \mathbb{R}^N, \quad f_t \in \mathbb{R}^k, \\ \Lambda &\in \mathbb{R}^{N \times k}, \quad k \ll N. \end{aligned}$$

Evolution of the factors:

$$\underbrace{f_t = A f_{t-1} + u_t}_{\text{Explicit time dynamics}} \ .$$

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Observed returns and common factors:

$$\underbrace{r_t = \Lambda f_t + \epsilon_t}_{\text{Common-factor representation}} \quad , \qquad \begin{aligned} &r_t \in \mathbb{R}^N, \quad f_t \in \mathbb{R}^k, \\ &\Lambda \in \mathbb{R}^{N \times k}, \quad k \ll N. \end{aligned}$$

Evolution of the factors:

$$\underbrace{f_t = A f_{t-1} + u_t}_{\text{Explicit time dynamics}} \ .$$

- Λ describes how each asset responds to the common factors.

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Observed returns and common factors:

$$\underbrace{r_t = \Lambda f_t + \epsilon_t}_{\text{Common-factor representation}} \quad , \qquad \begin{aligned} r_t &\in \mathbb{R}^N, \quad f_t \in \mathbb{R}^k, \\ \Lambda &\in \mathbb{R}^{N \times k}, \quad k \ll N. \end{aligned}$$

Evolution of the factors:

$$\underbrace{f_t = A f_{t-1} + u_t}_{\text{Explicit time dynamics}} \ .$$

- Λ describes how each asset responds to the common factors.
- A describes how current factors depend on prior factors.

Dynamic Factor Models: Structure

Q: How does a DFM extend PCA to model factor dynamics?

Observed returns and common factors:

$$\underbrace{r_t = \Lambda f_t + \epsilon_t}_{\text{Common-factor representation}} \quad , \qquad \begin{aligned} &r_t \in \mathbb{R}^N, \quad f_t \in \mathbb{R}^k, \\ &\Lambda \in \mathbb{R}^{N \times k}, \quad k \ll N. \end{aligned}$$

Evolution of the factors:

$$\underbrace{f_t = A f_{t-1} + u_t}_{\text{Explicit time dynamics}} \ .$$

- Λ describes how each asset responds to the common factors.
- A describes how current factors depend on prior factors.
- Unlike PCA, which summarizes common variation across asset returns without explicitly modeling factor dynamics, a DFM also specifies how those common factors evolve over time.

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Input: 500 assets observed daily over 10 years:

$$r_t \in \mathbb{R}^{500}, \qquad t = 1, \ldots, T.$$

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Input: 500 assets observed daily over 10 years:

$$r_t \in \mathbb{R}^{500}, \qquad t = 1, \ldots, T.$$

Output: Suppose we retain $k = 5$ common factors:

$$f_t \in \mathbb{R}^{5}, \qquad t = 1, \ldots, T.$$

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Input: 500 assets observed daily over 10 years:

$$r_t \in \mathbb{R}^{500}, \qquad t = 1, \ldots, T.$$

Output: Suppose we retain $k = 5$ common factors:

$$f_t \in \mathbb{R}^{5}, \qquad t = 1, \ldots, T.$$

- Original panel: $T \times 500$ returns.

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Input: 500 assets observed daily over 10 years:

$$r_t \in \mathbb{R}^{500}, \qquad t = 1, \ldots, T.$$

Output: Suppose we retain $k = 5$ common factors:

$$f_t \in \mathbb{R}^{5}, \qquad t = 1, \ldots, T.$$

- Original panel: $T \times 500$ returns.
- Compressed panel: $T \times 5$ factor estimates.

Dynamic Factor Models: Output

Q: What dimension does a DFM actually reduce?

Input: 500 assets observed daily over 10 years:

$$r_t \in \mathbb{R}^{500}, \qquad t = 1, \ldots, T.$$

Output: Suppose we retain $k = 5$ common factors:

$$f_t \in \mathbb{R}^{5}, \qquad t = 1, \ldots, T.$$

- Original panel: $T \times 500$ returns.
- Compressed panel: $T \times 5$ factor estimates.
- Unlike a full-sequence embedding z_i, this produces *a factor vector for each date*.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.
2. Express the DFM as a **state-space model**, with
   - an observation equation $r_t = \Lambda f_t + \epsilon_t$, and
   - a transition equation $f_t = A f_{t-1} + u_t$.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.
2. Express the DFM as a **state-space model**, with
   - an observation equation $r_t = \Lambda f_t + \epsilon_t$, and
   - a transition equation $f_t = A f_{t-1} + u_t$.
3. Estimate Λ, A, noise parameters, and the latent factors; maximum likelihood with EM is one common approach.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.
2. Express the DFM as a **state-space model**, with
   - an observation equation $r_t = \Lambda f_t + \epsilon_t$, and
   - a transition equation $f_t = A f_{t-1} + u_t$.
3. Estimate Λ, A, noise parameters, and the latent factors; maximum likelihood with EM is one common approach.

Kalman methods:

- **Filter:** Estimate f_t using observations available through t.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.
2. Express the DFM as a **state-space model**, with
   - an observation equation $r_t = \Lambda f_t + \epsilon_t$, and
   - a transition equation $f_t = A f_{t-1} + u_t$.
3. Estimate Λ, A, noise parameters, and the latent factors; maximum likelihood with EM is one common approach.

Kalman methods:

- **Filter:** Estimate f_t using observations available through t.
- **Smoother:** Reestimate past factors using later observations too.

Dynamic Factor Models: Estimation

Q: How do we estimate the unobserved factors and their dynamics?

1. PCA can provide initial estimates of factors or loadings.
2. Express the DFM as a **state-space model**, with
   - an observation equation $r_t = \Lambda f_t + \epsilon_t$, and
   - a transition equation $f_t = A f_{t-1} + u_t$.
3. Estimate Λ, A, noise parameters, and the latent factors; maximum likelihood with EM is one common approach.

Kalman methods:

- **Filter:** Estimate f_t using observations available through t.
- **Smoother:** Reestimate past factors using later observations too.

For live forecasts, avoid using smoothed historical factors that incorporate future observations.

Dimension Reduction: Choosing the Method

Q: Which structure do we need to preserve?

Dimension Reduction: Choosing the Method

Q: Which structure do we need to preserve?

Method	**Main structure**	**Typical output**
PCA	Linear feature covariance	Scores by asset/date
Standard AE	Nonlinear feature compression	Codes by asset/date
Lagged PCA / SSA	Relationships across lags	Window scores
Sequence AE	Ordered temporal patterns	Window codes
DFM	Common factors + time dynamics	Factor time series

These methods solve different compression problems; they are not simply more complex versions of one another.

Summary

Q: When would we use each method in finance?

Summary

Q: When would we use each method in finance?

- **PCA:** Can a few linear combinations summarize correlated momentum horizons or economic indicators?

Summary

Q: When would we use each method in finance?

- **PCA:** Can a few linear combinations summarize correlated momentum horizons or economic indicators?
- **Autoencoder:** Might nonlinear compression retain more of the information in those features?

Summary

Q: When would we use each method in finance?

- **PCA:** Can a few linear combinations summarize correlated momentum horizons or economic indicators?
- **Autoencoder:** Might nonlinear compression retain more of the information in those features?
- **Lagged PCA / sequence autoencoder:** Can we summarize the recent behavior of an individual asset?

Summary

Q: When would we use each method in finance?

- **PCA:** Can a few linear combinations summarize correlated momentum horizons or economic indicators?
- **Autoencoder:** Might nonlinear compression retain more of the information in those features?
- **Lagged PCA / sequence autoencoder:** Can we summarize the recent behavior of an individual asset?
- **DFM:** Can we estimate common drivers of a large asset or macroeconomic panel and model their evolution?

Summary

Q: When would we use each method in finance?

- **PCA:** Can a few linear combinations summarize correlated momentum horizons or economic indicators?
- **Autoencoder:** Might nonlinear compression retain more of the information in those features?
- **Lagged PCA / sequence autoencoder:** Can we summarize the recent behavior of an individual asset?
- **DFM:** Can we estimate common drivers of a large asset or macroeconomic panel and model their evolution?

Caution: High explained variance or low reconstruction loss need not imply better forecasts or investment signals. Evaluate any resulting signal out of sample, without look-ahead bias.

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

Outline

3 Explore
- AutoCorrelation Functions
- Clustering
- Dimension Reduction
- Stationarity
- Nonstationarity and Cointegration
- Decomposition, Filters, and Seasonality
- From Diagnostics to Models

From Diagnostics to Models

Our diagnostics guide our modeling choices:

- Autocorrelation $\rightarrow$ Autoregressive structure (AR, ARIMA)
- Trends / nonstationarity $\rightarrow$ Differencing or trend models
- Seasonality $\rightarrow$ Seasonal components (SARIMA, decomposition)
- Lagged effects $\rightarrow$ Distributed lag / ADL models
- Long-run relationships $\rightarrow$ Error Correction Models (ECM)
- Multiple series $\rightarrow$ VAR or panel models

From Diagnostics to Models

Our diagnostics guide our modeling choices:

- Autocorrelation $\rightarrow$ Autoregressive structure (AR, ARIMA)
- Trends / nonstationarity $\rightarrow$ Differencing or trend models
- Seasonality $\rightarrow$ Seasonal components (SARIMA, decomposition)
- Lagged effects $\rightarrow$ Distributed lag / ADL models
- Long-run relationships $\rightarrow$ Error Correction Models (ECM)
- Multiple series $\rightarrow$ VAR or panel models

Diagnostics are not just descriptive—they determine the model

Outline

1 Introduction

2 Data Prep

3 Explore

4 Explain

5 Forecast

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Modeling Temporal Dependence

From our analysis, we observed:

- Autocorrelation
- Persistence over time

Modeling Temporal Dependence

From our analysis, we observed:

- Autocorrelation
- Persistence over time

Key idea:

- The past helps predict the present

A Simple Dynamic Model

One way to capture dependence:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$$

- Current values depend on past values
- This introduces dynamics into the model

A Simple Dynamic Model

One way to capture dependence:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$$

- Current values depend on past values
- This introduces dynamics into the model

We now generalize this idea with richer models

Outline

4 Explain
- Modeling Dependence (why dynamics matter)
- Linear Models (ADL, FDL)
- Linear Model Assumptions
- Linear Model Violations (Serial Correlation)
- Rolling Regression
- Event Study
- Long Run Structure (ECM)
- State Space
- Paradigm Comparison

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Outline

4 Explain
- Modeling Dependence (why dynamics matter)
- Linear Models (ADL, FDL)
- Linear Model Assumptions
- Linear Model Violations (Serial Correlation)
- Rolling Regression
- Event Study
- Long Run Structure (ECM)
- State Space
- Paradigm Comparison

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Outline

4. Explain
   - Modeling Dependence (why dynamics matter)
   - Linear Models (ADL, FDL)
   - Linear Model Assumptions
   - Linear Model Violations (Serial Correlation)
   - Rolling Regression
   - Event Study
   - Long Run Structure (ECM)
   - State Space
   - Paradigm Comparison

Outline

4 Explain
- Modeling Dependence (why dynamics matter)
- Linear Models (ADL, FDL)
- Linear Model Assumptions
- Linear Model Violations (Serial Correlation)
- Rolling Regression
- Event Study
- Long Run Structure (ECM)
- State Space
- Paradigm Comparison

Comparing Time Series Modeling Approaches

Approach	**Strengths**	**Limitations**
Linear Models	Interpretable Economic structure Works with small data	Restrictive Linear dynamics
State Space Models	Flexible Captures latent components Unifying framework	More complex Requires specification
Machine Learning	Strong predictive performance Captures nonlinearities	Less interpretable Data intensive
Global / Foundation Models	Leverages many series Pretrained / scalable	Less transparent Harder to control

Outline

1. Introduction
2. Data Prep
3. Explore
4. Explain
5. Forecast

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

What is Forecasting?

- Forecasting is NOT EDA

What is Forecasting?

- Forecasting is NOT EDA
- Forecasting is NOT Causal Estimation & Inference

What is Forecasting?

- Forecasting is NOT EDA
- Forecasting is NOT Causal Estimation & Inference
- Forecasting IS a type of prediction: predicting outcomes at future dates from currently available information.

Forecasting vs Other Prediction Tasks

Q: What distinguishes a forecast from another prediction?

Forecasting vs Other Prediction Tasks

Q: What distinguishes a forecast from another prediction?

- **Forecast:** At today's close, what will TSLA's return be tomorrow using today's information?

Forecasting vs Other Prediction Tasks

Q: What distinguishes a forecast from another prediction?

- **Forecast:** At today's close, what will TSLA's return be tomorrow using today's information?
- **Conditional scenario:** What would tomorrow's expected TSLA return be *if* the market rises 2% tomorrow?

Forecasting vs Other Prediction Tasks

Q: What distinguishes a forecast from another prediction?

- **Forecast:** At today's close, what will TSLA's return be tomorrow using today's information?
- **Conditional scenario:** What would tomorrow's expected TSLA return be *if* the market rises 2% tomorrow?
- A future conditional scenario becomes a deployable forecast only if that future market movement is known, predicted, or explicitly treated as a scenario.

Forecasting vs Other Prediction Tasks

Q: What distinguishes a forecast from another prediction?

- **Forecast:** At today's close, what will TSLA's return be tomorrow using today's information?
- **Conditional scenario:** What would tomorrow's expected TSLA return be *if* the market rises 2% tomorrow?
- A future conditional scenario becomes a deployable forecast only if that future market movement is known, predicted, or explicitly treated as a scenario.
- Many ML applications predict observed outcomes that are not in the future; **time-series forecasting** emphasizes future, out-of-sample dates and information availability.

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series:

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g.

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional:

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional: data observed at a single point in time across many units. e.g.

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional: data observed at a single point in time across many units. e.g. earnings for many companies during Q1'25.

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional: data observed at a single point in time across many units. e.g. earnings for many companies during Q1'25.
- Panel:

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional: data observed at a single point in time across many units. e.g. earnings for many companies during Q1'25.
- Panel: data observed across many units and time. e.g.

What are we forecasting?

Q: Time Series vs Cross Sectional vs Panel?

- Time Series: data observed on a single series through time. e.g. quarterly earnings on a single company over the past 10yrs
- Cross Sectional: data observed at a single point in time across many units. e.g. earnings for many companies during Q1'25.
- Panel: data observed across many units and time. e.g. quarterly earnings for many companies over the past 10yrs

Forecasting Taxonomy

Forecasting Taxonomy

- **Architecture / model family:** classical time-series and econometric models; state space; tree-based ML; deep neural networks.

Forecasting Taxonomy

- **Architecture / model family:** classical time-series and econometric models; state space; tree-based ML; deep neural networks.
- **Training scope:** local (one series); global (many related series); pretrained foundation model (learned from broader corpora).

Forecasting Taxonomy

- **Architecture / model family:** classical time-series and econometric models; state space; tree-based ML; deep neural networks.
- **Training scope:** local (one series); global (many related series); pretrained foundation model (learned from broader corpora).
- **Horizon construction:** direct, recursive, or multi-output forecasts.

Forecasting Taxonomy

- **Architecture / model family:** classical time-series and econometric models; state space; tree-based ML; deep neural networks.
- **Training scope:** local (one series); global (many related series); pretrained foundation model (learned from broader corpora).
- **Horizon construction:** direct, recursive, or multi-output forecasts.
- **Additional forecast requirements:** predictive uncertainty, coherent aggregation, and updating as data arrive.

A Running Application: Monthly Company Sales

A Running Application: Monthly Company Sales

- Company total sales = sum of division sales; each division's sales = sum of its product (SKUs') sales.

A Running Application: Monthly Company Sales

- Company total sales = sum of division sales; each division's sales = sum of its product (SKUs') sales.
- Our base data: monthly dollar sales $s_t^{(i)}$ for SKU i; many SKUs belong to several divisions.

A Running Application: Monthly Company Sales

- Company total sales = sum of division sales; each division's sales = sum of its product (SKUs') sales.
- Our base data: monthly dollar sales $s_t^{(i)}$ for SKU i; many SKUs belong to several divisions.
- A recurring task is to forecast the next 12 months using observed histories through month t.

A Running Application: Monthly Company Sales

- Company total sales = sum of division sales; each division's sales = sum of its product (SKUs') sales.
- Our base data: monthly dollar sales $s_t^{(i)}$ for SKU i; many SKUs belong to several divisions.
- A recurring task is to forecast the next 12 months using observed histories through month t.
- Different problems will motivate different tools:
  - Nonlinear responses $\rightarrow$ ML.
  - Learned temporal representations $\rightarrow$ Deep learning.
  - Short individual histories $\rightarrow$ Global learning.
  - Company/Division/SKU adding-up $\rightarrow$ Hierarchical reconciliation.
  - Limited internal data or costly model training $\rightarrow$ Foundation models.

A Running Application: Monthly Company Sales

- Company total sales = sum of division sales; each division's sales = sum of its product (SKUs') sales.
- Our base data: monthly dollar sales $s_t^{(i)}$ for SKU i; many SKUs belong to several divisions.
- A recurring task is to forecast the next 12 months using observed histories through month t.
- Different problems will motivate different tools:
  - Nonlinear responses $\rightarrow$ ML.
  - Learned temporal representations $\rightarrow$ Deep learning.
  - Short individual histories $\rightarrow$ Global learning.
  - Company/Division/SKU adding-up $\rightarrow$ Hierarchical reconciliation.
  - Limited internal data or costly model training $\rightarrow$ Foundation models.
- Note: We will occasionally switch to different applications (e.g. financial returns) to illustrate specialized tasks such as volatility forecasting.

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

- t: forecast origin; $t + h$: date being forecast.

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

- t: forecast origin; $t + h$: date being forecast.
- $s_t^{(i)}$: observed monthly sales for SKU i.

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

- t: forecast origin; $t + h$: date being forecast.
- $s_t^{(i)}$: observed monthly sales for SKU i.
- $\mathcal{F}_t$: information genuinely available by the end of month t.

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

- t: forecast origin; $t + h$: date being forecast.
- $s_t^{(i)}$: observed monthly sales for SKU i.
- $\mathcal{F}_t$: information genuinely available by the end of month t.
- $\mathbf{x}_t^{(i)}$: engineered historical features known at t; $\mathbf{u}_t^{(i)}$ can denote the vector observed at a single month for recurrent models.

Notation: Forecast Origin vs Target Date

Q: What information may the model use?

- t: forecast origin; $t + h$: date being forecast.
- $s_t^{(i)}$: observed monthly sales for SKU i.
- $\mathcal{F}_t$: information genuinely available by the end of month t.
- $\mathbf{x}_t^{(i)}$: engineered historical features known at t; $\mathbf{u}_t^{(i)}$ can denote the vector observed at a single month for recurrent models.
- $\widehat{s}_{t+h|t}^{(i)}$: forecast made at t of sales at $t + h$.
- All transformations and covariates must respect information availability; future planned variables may be used if they were already known at t.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess
- Subjective with theory

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess
- Subjective with theory
- Extract from experts

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess
- Subjective with theory
- Extract from experts
- Extract from markets

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess
- Subjective with theory
- Extract from experts
- Extract from markets
- Statistical without theory

Forecasting approaches

Q: What are some approaches you can take when making a forecast?

- Subjective guess
- Subjective with theory
- Extract from experts
- Extract from markets
- Statistical without theory
- Statistical with theory

Our forecasting approach

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?
- A: A pure subjective guess. If you can't justify your forecast, it is not trustworthy.

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?
- A: A pure subjective guess. If you can't justify your forecast, it is not trustworthy.
- Statistical forecasting without theory is similarly dangerous, but often viewed as superior to a subjective guess without theory.

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?
- A: A pure subjective guess. If you can't justify your forecast, it is not trustworthy.
- Statistical forecasting without theory is similarly dangerous, but often viewed as superior to a subjective guess without theory.
- Subjectivity is sometimes necessary when information is incomplete.

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?
- A: A pure subjective guess. If you can't justify your forecast, it is not trustworthy.
- Statistical forecasting without theory is similarly dangerous, but often viewed as superior to a subjective guess without theory.
- Subjectivity is sometimes necessary when information is incomplete.
- We can combine several of the methods above in an ensemble fashion for a rich forecast.

Our forecasting approach

- Q: Which of the aforementioned approaches should you never use?
- A: A pure subjective guess. If you can't justify your forecast, it is not trustworthy.
- Statistical forecasting without theory is similarly dangerous, but often viewed as superior to a subjective guess without theory.
- Subjectivity is sometimes necessary when information is incomplete.
- We can combine several of the methods above in an ensemble fashion for a rich forecast.
- Our focus will be on Statistical forecasting grounded in theory.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Forecasting Workflow

Forecasting Workflow

1. Reconnaissance

Forecasting Workflow

1. Reconnaissance
2. Model Design

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep
4. Exploratory Data Analysis (EDA)

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep
4. Exploratory Data Analysis (EDA)
5. Iterate 1,2,3,4

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep
4. Exploratory Data Analysis (EDA)
5. Iterate 1,2,3,4
6. Forecast

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep
4. Exploratory Data Analysis (EDA)
5. Iterate 1,2,3,4
6. Forecast
7. Evaluate

Forecasting Workflow

1. Reconnaissance
2. Model Design
3. Data Prep
4. Exploratory Data Analysis (EDA)
5. Iterate 1,2,3,4
6. Forecast
7. Evaluate
8. Iterate all the above

1. Reconnaissance

Recon is done at the “whiteboard”, before your team ever touches the keyboard

1. Reconnaissance

Recon is done at the “whiteboard”, before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve.

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)
- Units (point estimate, range, probability above some threshold, etc...)

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)
- Units (point estimate, range, probability above some threshold, etc...)
- How often (re)forecast (single use case, every day, every week, etc..)

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)
- Units (point estimate, range, probability above some threshold, etc...)
- How often (re)forecast (single use case, every day, every week, etc..)
- Benchmark (absolute forecast accuracy, relative to some competitor, internal hurdle rate, etc..)

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)
- Units (point estimate, range, probability above some threshold, etc...)
- How often (re)forecast (single use case, every day, every week, etc..)
- Benchmark (absolute forecast accuracy, relative to some competitor, internal hurdle rate, etc..)
- Forecast horizon (one tick, one day, one month, five years, etc..)

1. Reconnaissance

Recon is done at the "whiteboard", before your team ever touches the keyboard

We must interview stakeholders to understand the scope of the problem we are trying to solve. Several considerations:

- Object of Interest (stock price, earnings, customer visits, etc..)
- Units (point estimate, range, probability above some threshold, etc...)
- How often (re)forecast (single use case, every day, every week, etc..)
- Benchmark (absolute forecast accuracy, relative to some competitor, internal hurdle rate, etc..)
- Forecast horizon (one tick, one day, one month, five years, etc..)
- Data Properties and availability (what can you get, regular time intervals, suffers from look ahead bias, trustworthy, etc...)

2. Model Design

2. Model Design

- The model must be aligned with reconnaissance

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need
- Model should be aligned with purpose

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need
- Model should be aligned with purpose
  - Max accuracy of object of interest e.g.

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need
- Model should be aligned with purpose
  - Max accuracy of object of interest e.g. "I want the best forecast of next month's Treasury yield"

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need
- Model should be aligned with purpose
  - Max accuracy of object of interest e.g. "I want the best forecast of next month's Treasury yield"
  - Proxy model / behavior e.g.

2. Model Design

- The model must be aligned with reconnaissance
  - The worst mistake is to give a forecast that is not what your stakeholders need
- Model should be aligned with purpose
  - Max accuracy of object of interest e.g. "I want the best forecast of next month's Treasury yield"
  - Proxy model / behavior e.g. "I want to know if the CoT data can forecast next month's Treasury yield"

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...
3. Align: i) aligning date/time stamps, ii) align activity dates or release dates?

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...
3. Align: i) aligning date/time stamps, ii) align activity dates or release dates?
4. Clean:
    - Be sure to clean AFTER you align

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...
3. Align: i) aligning date/time stamps, ii) align activity dates or release dates?
4. Clean:
   - Be sure to clean AFTER you align
   - Use caution when deleting a "row". Rows are time periods in time series forecasting. Do you really want to delete a day?

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...
3. Align: i) aligning date/time stamps, ii) align activity dates or release dates?
4. Clean:
   - Be sure to clean AFTER you align
   - Use caution when deleting a "row". Rows are time periods in time series forecasting. Do you really want to delete a day?
   - Set up data tables in wide form (human readable) or long form (machine readable)

3. Data Prep

1. Acquire: i) one time or ongoing? ii) size iii) availability (e.g. licensing, etc..)
2. Store: i) redownload and rewrite each time or smaller downloads and append to database? ii) local or cloud, etc...
3. Align: i) aligning date/time stamps, ii) align activity dates or release dates?
4. Clean:
   - Be sure to clean AFTER you align
   - Use caution when deleting a "row". Rows are time periods in time series forecasting. Do you really want to delete a day?
   - Set up data tables in wide form (human readable) or long form (machine readable)
   - Fill or delete missing, N/A, wrong values (e.g. negative price)

4. EDA

4. EDA

- Common mistake: do EDA after estimation and forecasting.

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process
- Do you see outliers? If so, why?
  - Are they due to mistakes in the database? If so, go back to the Data Prep step.

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process
- Do you see outliers? If so, why?
  - Are they due to mistakes in the database? If so, go back to the Data Prep step.
  - If they are reflective of the truth you may need to go back to Model Design

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process
- Do you see outliers? If so, why?
  - Are they due to mistakes in the database? If so, go back to the Data Prep step.
  - If they are reflective of the truth you may need to go back to Model Design
- Other considerations that may trigger you to rethink your model design
  - Do you see structural breaks?

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process
- Do you see outliers? If so, why?
  - Are they due to mistakes in the database? If so, go back to the Data Prep step.
  - If they are reflective of the truth you may need to go back to Model Design
- Other considerations that may trigger you to rethink your model design
  - Do you see structural breaks?
  - Does "X" comove with "Y"?

4. EDA

- Common mistake: do EDA after estimation and forecasting. EDA should drive model design through an iterative process
- Do you see outliers? If so, why?
  - Are they due to mistakes in the database? If so, go back to the Data Prep step.
  - If they are reflective of the truth you may need to go back to Model Design
- Other considerations that may trigger you to rethink your model design
  - Do you see structural breaks?
  - Does "X" comove with "Y"?
  - Is "Y" much more volatile than "X"?

5. Iterate

- Will the output of your model design answer the stakeholders' question?

5. Iterate

- Will the output of your model design answer the stakeholders' question?
- Is the data you have sufficient to estimate that model?

5. Iterate

- Will the output of your model design answer the stakeholders' question?
- Is the data you have sufficient to estimate that model?
- Does your EDA support the model design?

5. Iterate

- Will the output of your model design answer the stakeholders' question?
- Is the data you have sufficient to estimate that model?
- Does your EDA support the model design?

If the answer is NO to any of these questions, you need to iterate back to Steps 1,2,3

6. Forecast

- Implementation in Excel, code, etc..

6. Forecast

- Implementation in Excel, code, etc..
- Best Practices
  - Detail your process with comments

6. Forecast

- Implementation in Excel, code, etc..
- Best Practices
  - Detail your process with comments
  - Automate as much as possible

6. Forecast

- Implementation in Excel, code, etc..
- Best Practices
  - Detail your process with comments
  - Automate as much as possible
  - Store forecasts with date/time stamp for ex-post evaluation

7. Evaluate

What Evaluating?

7. Evaluate

What Evaluating?

- Your model(s) vs benchmark

7. Evaluate

What Evaluating?

- Your model(s) vs benchmark
- Object of interest & Forecast Error

7. Evaluate

What Evaluating?

- Your model(s) vs benchmark
- Object of interest & Forecast Error
- In Sample vs Out of Sample
  - in sample: data that you used to build your model
  - out of sample: data that you are using to test your model's accuracy
  - Include prediction/confidence intervals

7. Evaluate

What Evaluating?

- Your model(s) vs benchmark
- Object of interest & Forecast Error
- In Sample vs Out of Sample
  - in sample: data that you used to build your model
  - out of sample: data that you are using to test your model's accuracy
  - Include prediction/confidence intervals
- Prediction / forecast intervals

How Evaluate?

7. Evaluate

What Evaluating?

- Your model(s) vs benchmark
- Object of interest & Forecast Error
- In Sample vs Out of Sample
  - in sample: data that you used to build your model
  - out of sample: data that you are using to test your model's accuracy
  - Include prediction/confidence intervals
- Prediction / forecast intervals

How Evaluate?

- Summary metrics, e.g. MSFE (Mean squared forecast error)
- Forecast error plots with time and covariates
- Forecast error densities

8. Iterate

- Was your model sufficiently accurate?
- Is the original question answered satisfactorily?

8. Iterate

- Was your model sufficiently accurate?
- Is the original question answered satisfactorily?

If the answer is NO to any of these questions, you need to iterate back to the whole process.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Challenges with Forecasting

Challenges with Forecasting

All

Challenges with Forecasting

All Forecasts

Challenges with Forecasting

All Forecasts Are

Challenges with Forecasting

All Forecasts Are Wrong

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample
- Your parameter estimates may be wrong

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample
- Your parameter estimates may be wrong
- Data may have so much noise that you can't verify how wrong you are

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample
- Your parameter estimates may be wrong
- Data may have so much noise that you can't verify how wrong you are
- If you need to forecast an "X" to predict your "Y", your errors will compound; e.g.

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample
- Your parameter estimates may be wrong
- Data may have so much noise that you can't verify how wrong you are
- If you need to forecast an "X" to predict your "Y", your errors will compound; e.g. If I need to forecast the market's return tomorrow in order to predict my stock's return tomorrow, the errors in my market forecast will amplify my stock return forecast errors

Challenges with Forecasting

All Forecasts Are Wrong

- Your model may be wrong
  - ... in sample
  - ... out of sample
- Your parameter estimates may be wrong
- Data may have so much noise that you can't verify how wrong you are
- If you need to forecast an "X" to predict your "Y", your errors will compound; e.g. If I need to forecast the market's return tomorrow in order to predict my stock's return tomorrow, the errors in my market forecast will amplify my stock return forecast errors
- Advice: never settle for a single point forecast; provide a range of values

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

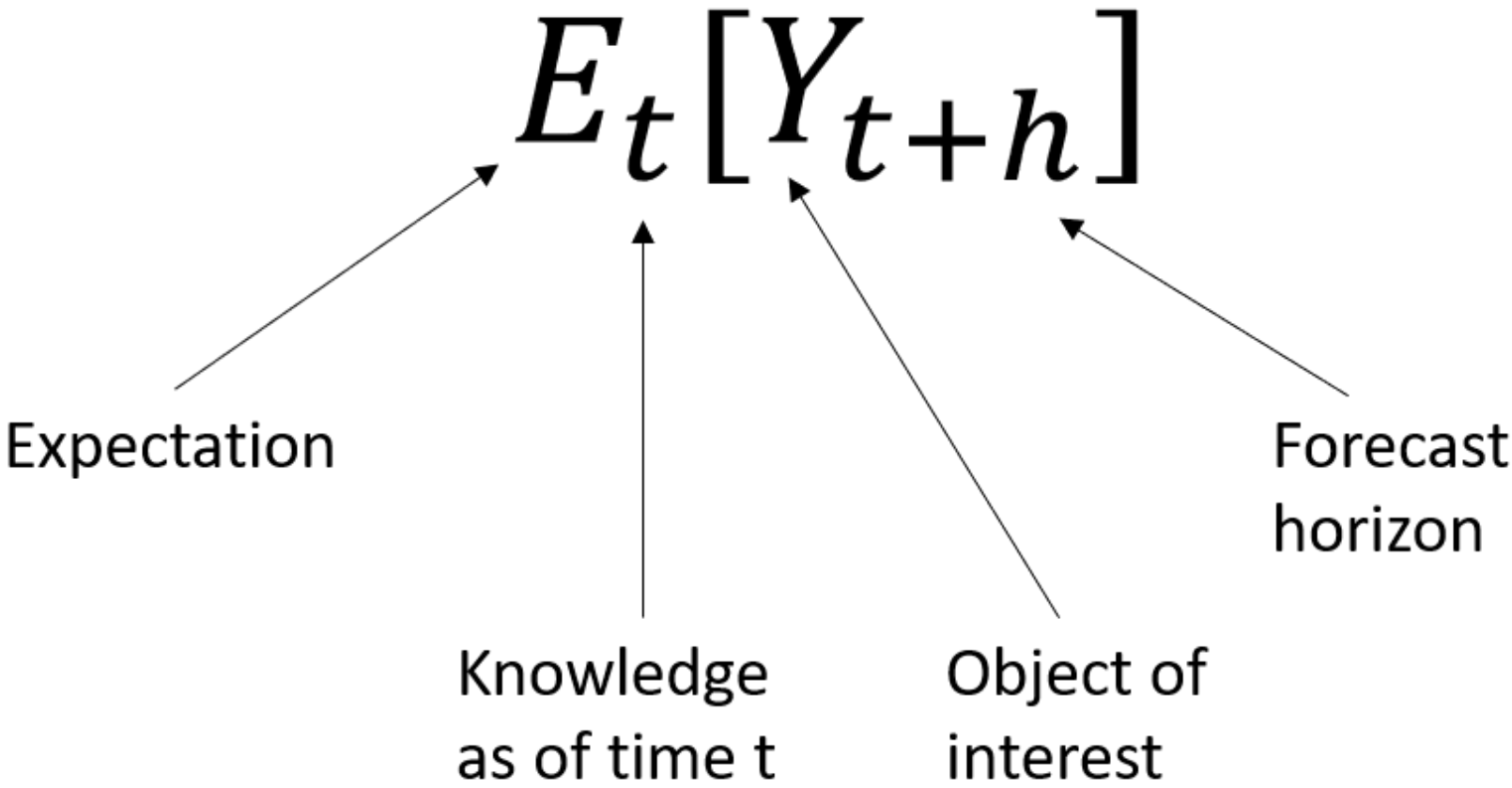
Anatomy of a Forecast

Q: Can you explain the components of this forecast?

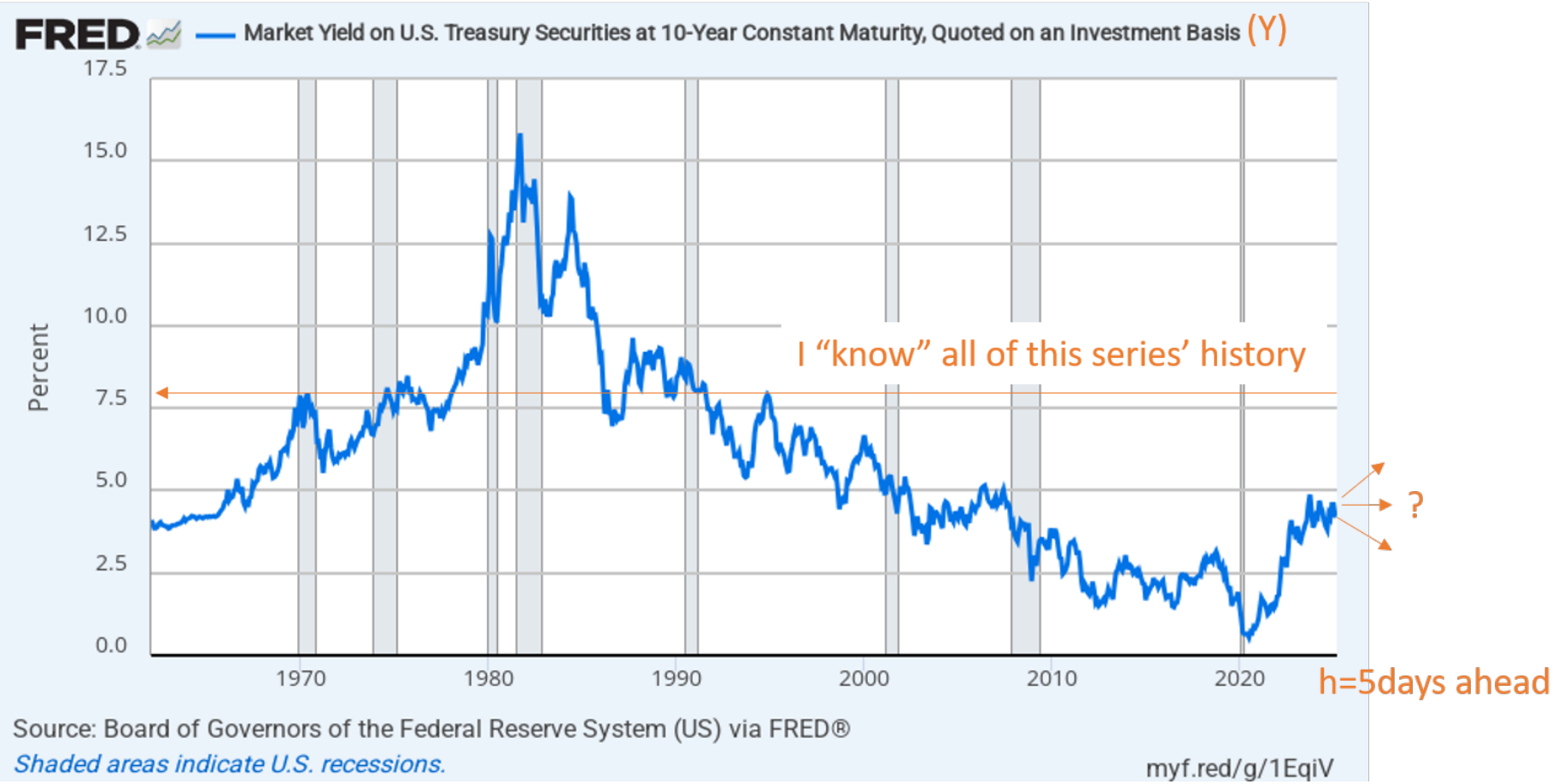
$$E_t[Y_{t+h}]$$

Anatomy of a Forecast

Q: Can you explain the components of this forecast?



5day ahead US Treasury yield forecast



Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Train / Test Split

- Training Period:

Train / Test Split

- Training Period: time period used to estimate the model parameters

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period:

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period: time period used to evaluate if the model works well

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period: time period used to evaluate if the model works well
  - aka Hold Out Sample, Pseduo Out of Sample, Evaluation Period

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period: time period used to evaluate if the model works well
  - aka Hold Out Sample, Pseduo Out of Sample, Evaluation Period
- Forecasting Period:

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period: time period used to evaluate if the model works well
  - aka Hold Out Sample, Pseduo Out of Sample, Evaluation Period
- Forecasting Period: time period for which we are making a forecast

Train / Test Split

- Training Period: time period used to estimate the model parameters
  - aka In-Sample, Estimation Period
- Testing Period: time period used to evaluate if the model works well
  - aka Hold Out Sample, Pseduo Out of Sample, Evaluation Period
- Forecasting Period: time period for which we are making a forecast
  - aka Out of Sample

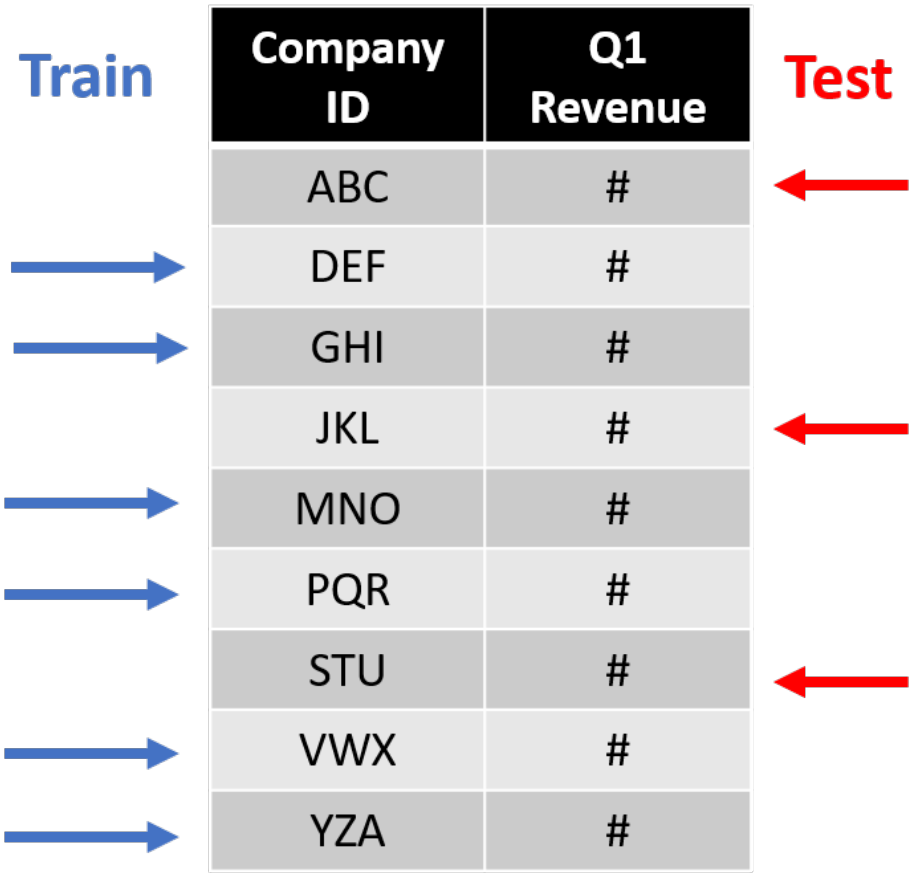
Typical Train / Test Split

Q: Consider the data below. How do we determine the train/test split?

Company ID	Q1 Revenue
ABC	#
DEF	#
GHI	#
JKL	#
MNO	#
PQR	#
STU	#
VWX	#
YZA	#

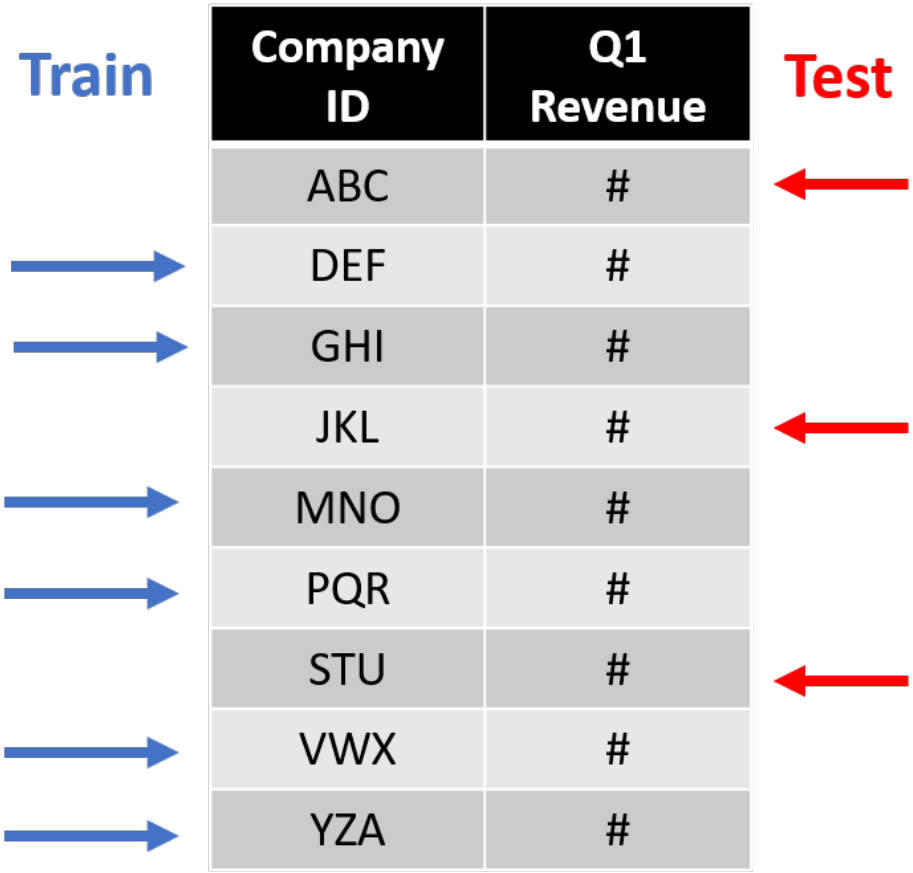
Typical Train / Test Split

A: Random assignment of each "row".



Typical Train / Test Split

A: Random assignment of each "row".



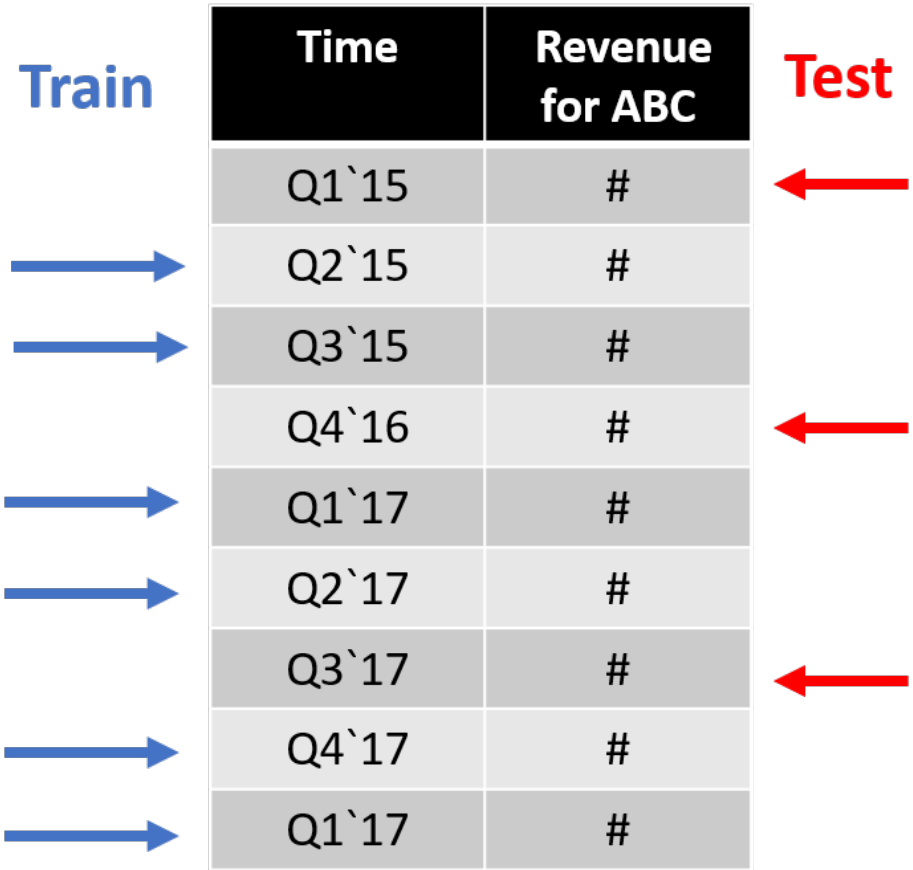
The fraction of the sample assigned to train is usually higher than to split; e.g. 80% to 20%. This is a nuisance parameter that you should stress test.

Time Series Train / Test Split

Q: Why isn't the typical approach (outlined on prior slide) appropriate for our usual forecasting applications?

Time Series Train / Test Split

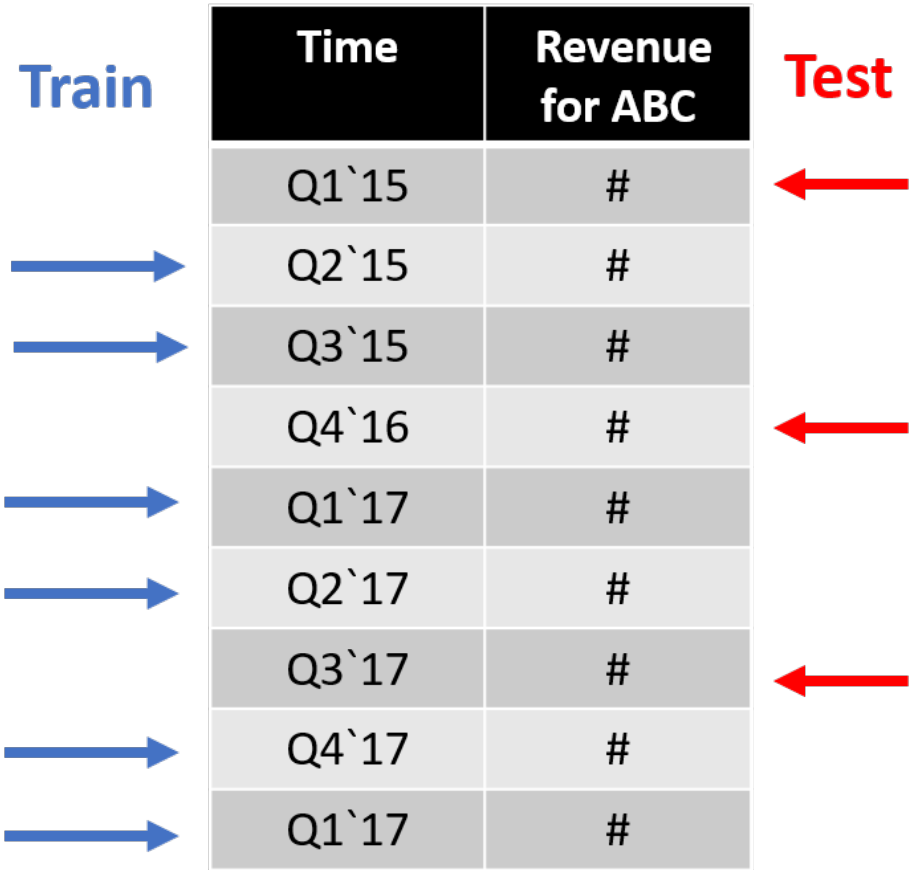
Q: Why isn't the typical approach (outlined on prior slide) appropriate for our usual forecasting applications?



Time	Revenue for ABC
Q1`15	#
Q2`15	#
Q3`15	#
Q4`16	#
Q1`17	#
Q2`17	#
Q3`17	#
Q4`17	#
Q1`17	#

Time Series Train / Test Split

Q: Why isn't the typical approach (outlined on prior slide) appropriate for our usual forecasting applications?



Time	Revenue for ABC
Q1`15	#
Q2`15	#
Q3`15	#
Q4`16	#
Q1`17	#
Q2`17	#
Q3`17	#
Q4`17	#
Q1`17	#

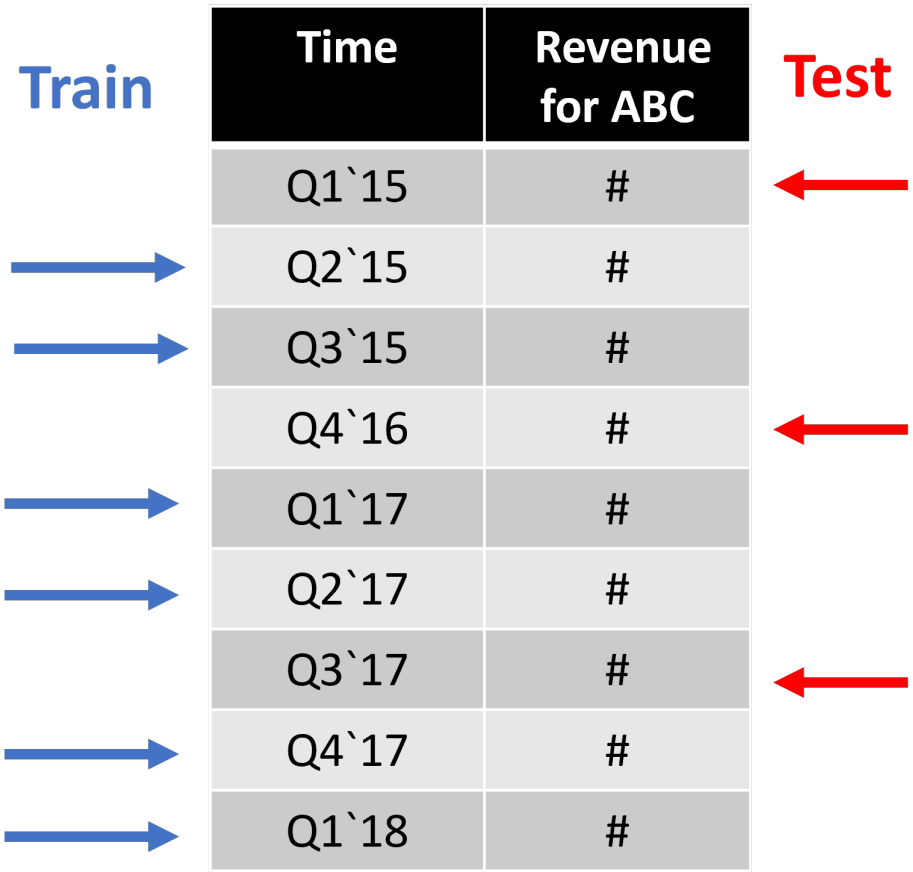
A: Randomization disrupts the temporal dependence.

Proper Splitting with Time Series data

Q: What's the proper way to split train/test with time series data?

Proper Splitting with Time Series data

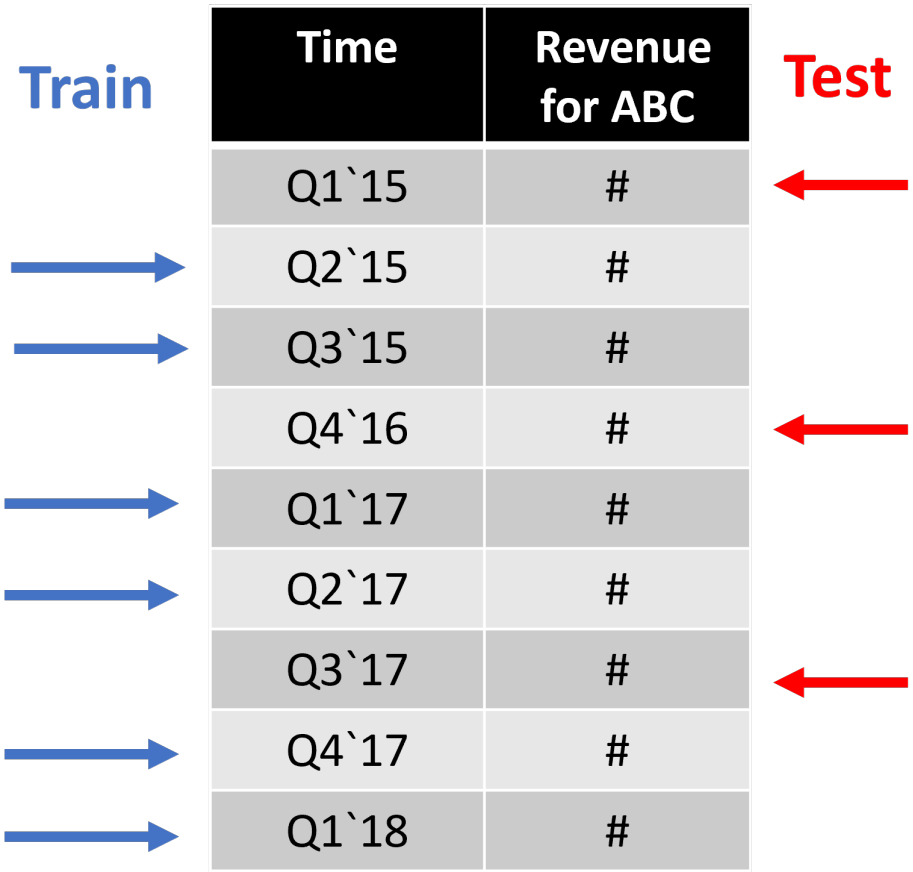
Q: What's the proper way to split train/test with time series data?



Time	Revenue for ABC
Q1`15	#
Q2`15	#
Q3`15	#
Q4`16	#
Q1`17	#
Q2`17	#
Q3`17	#
Q4`17	#
Q1`18	#

Proper Splitting with Time Series data

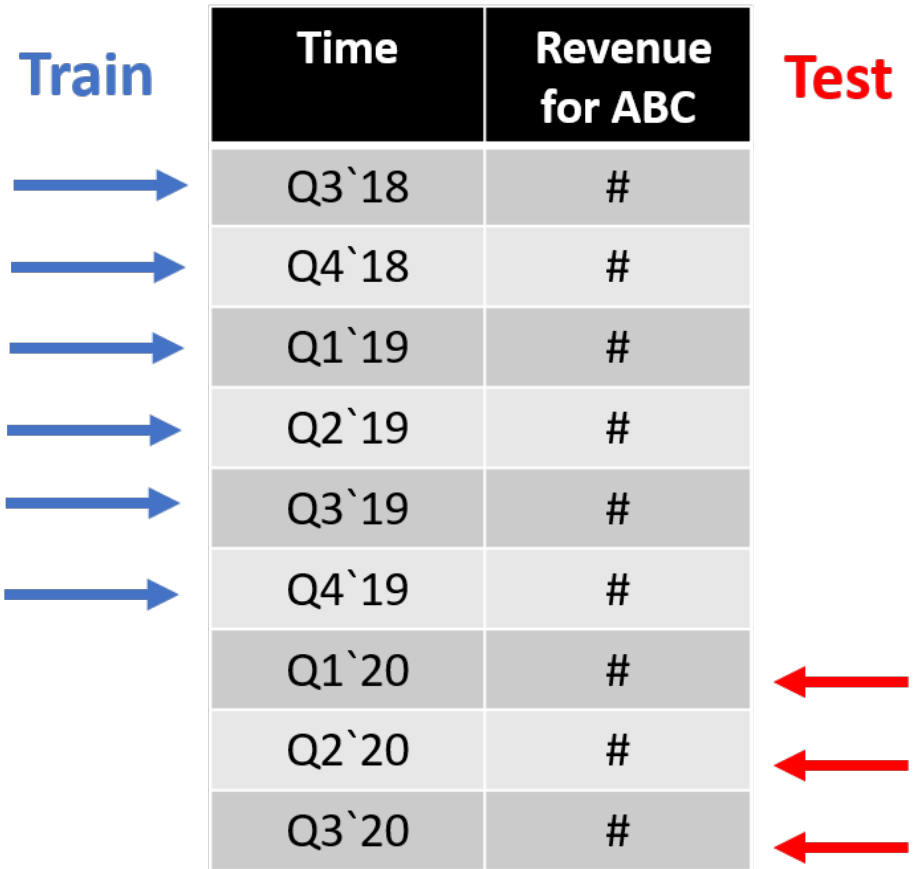
Q: What's the proper way to split train/test with time series data?



Again, the size of the split needs to be stress tested.

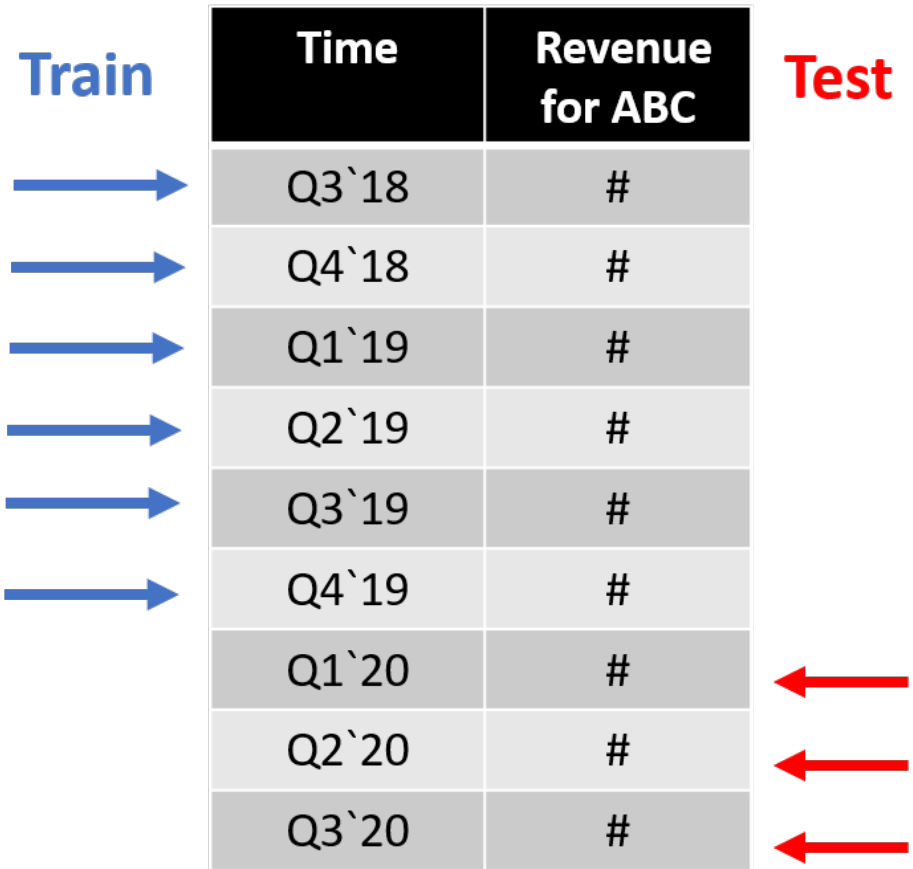
Dangers with Test/Train Splitting

Q: What's the danger in this split? (Notice the different dates)



Dangers with Test/Train Splitting

Q: What's the danger in this split? (Notice the different dates)



Time	Revenue for ABC
Q3`18	#
Q4`18	#
Q1`19	#
Q2`19	#
Q3`19	#
Q4`19	#
Q1`20	#
Q2`20	#
Q3`20	#

A: COVID (2020) likely disrupted the historical patterns. The training sample is not representative of the testing sample.

Cross Validation in Time Series

Q: What's a more stable approach for train/test splitting time series?

Cross Validation in Time Series

Q: What's a more stable approach for train/test splitting time series?

A: Step 1: Separate the data into k-folds

	Time	Revenue for ABC
Fold 1	Q3`18	#
	Q4`18	#
	Q1`19	#
Fold 2	Q2`19	#
	Q3`19	#
	Q4`19	#
Fold 3	Q1`20	#
	Q2`20	#
	Q3`20	#

Cross Validation in Time Series

Q: What's a more stable approach for train/test splitting time series?

A: Step 1: Separate the data into k-folds

	Time	Revenue for ABC
Fold 1	Q3`18	#
	Q4`18	#
	Q1`19	#
Fold 2	Q2`19	#
	Q3`19	#
	Q4`19	#
Fold 3	Q1`20	#
	Q2`20	#
	Q3`20	#

Note: we should stress test fold size.

Cross Validation in Time Series

Cross Validation in Time Series

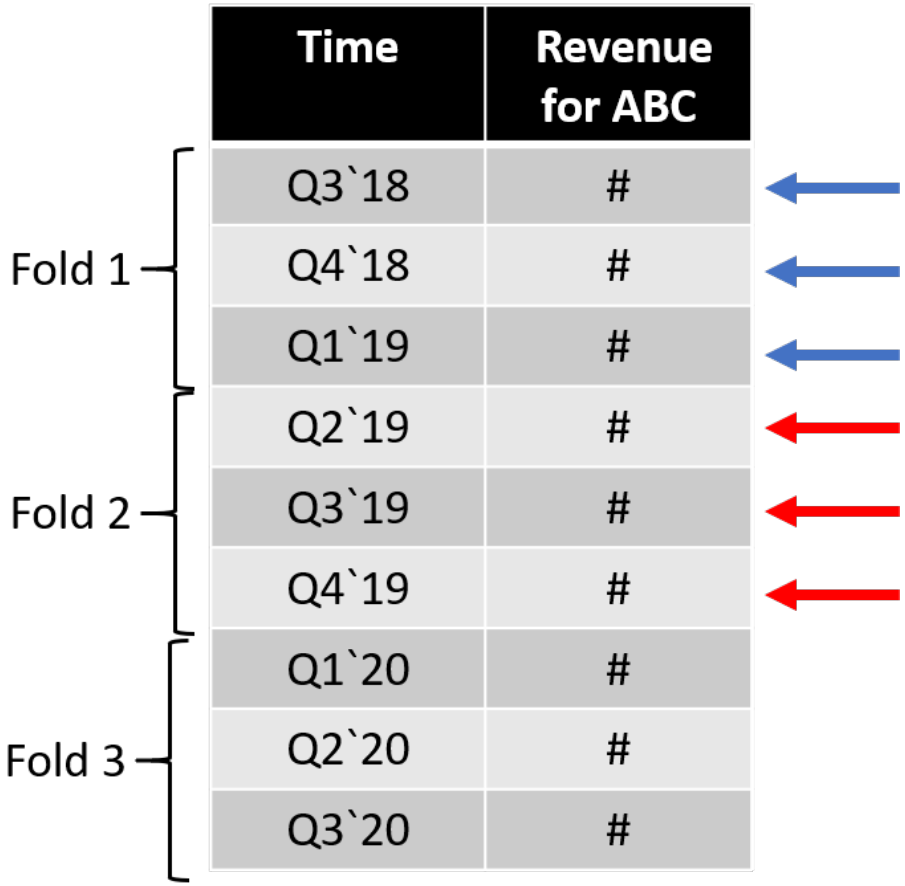
Step 2:

Cross Validation in Time Series

Step 2: Train on Fold 1 and Test on Fold 2

Cross Validation in Time Series

Step 2: Train on Fold 1 and Test on Fold 2



Note: there are many variants, which alter size and shape of testing folds, etc...

Cross Validation in Time Series

Cross Validation in Time Series

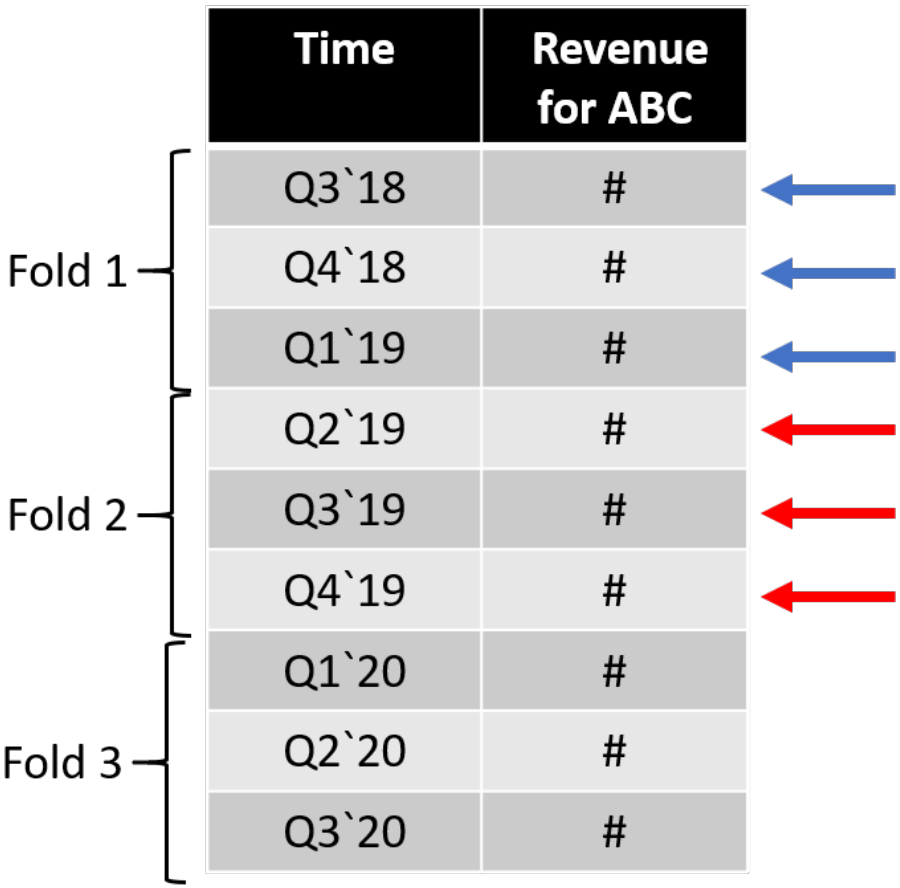
Step 3:

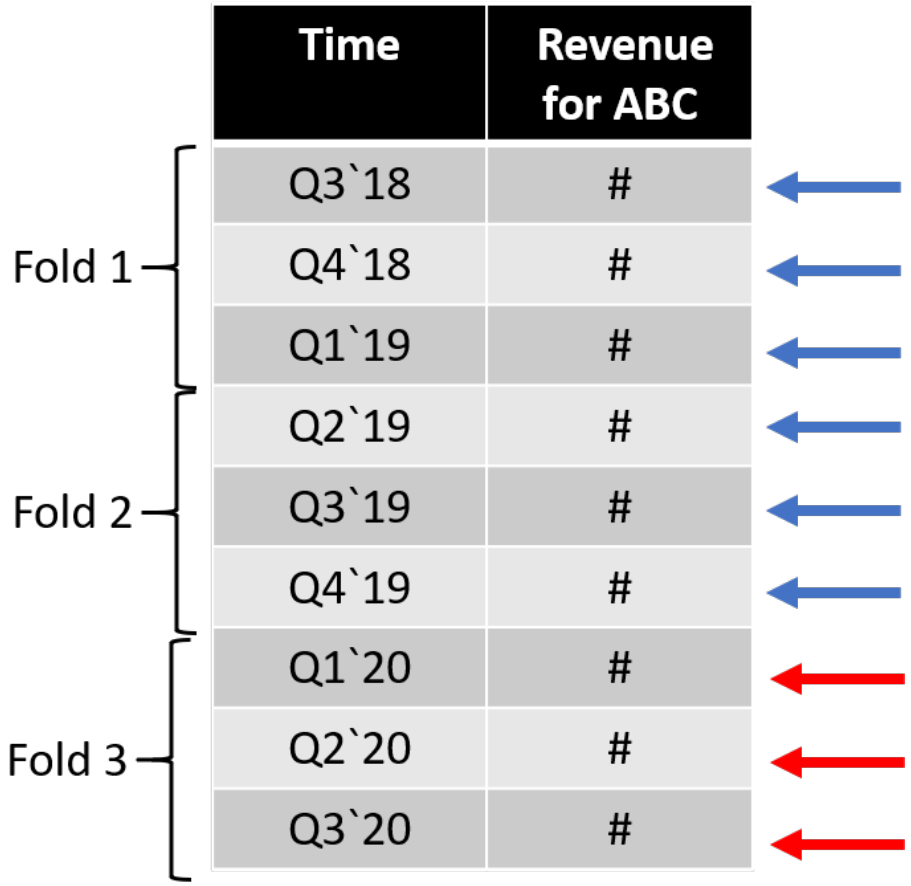
Cross Validation in Time Series

Step 3: Train on Folds 1 and 2 combined, Test on Fold 3

Cross Validation in Time Series

Step 3: Train on Folds 1 and 2 combined, Test on Fold 3





Repeat for all of your k folds.

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds
- Compute $\bar{\hat{\beta}} = \frac{1}{F} \sum_{f=1}^{F} \hat{\beta}_f$ as the

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds
- Compute $\bar{\hat{\beta}} = \frac{1}{F} \sum_{f=1}^{F} \hat{\beta}_f$ as the central tendency of the estimates

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds
- Compute $\bar{\hat{\beta}} = \frac{1}{F}\sum_{f=1}^{F}\hat{\beta}_f$ as the central tendency of the estimates
- Compute $\sqrt{\frac{1}{F}\sum_{f=1}^{F}(\hat{\beta}_f - \bar{\hat{\beta}})^2}$ as the

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds
- Compute $\bar{\hat{\beta}} = \frac{1}{F}\sum_{f=1}^{F}\hat{\beta}_f$ as the central tendency of the estimates
- Compute $\sqrt{\frac{1}{F}\sum_{f=1}^{F}(\hat{\beta}_f - \bar{\hat{\beta}})^2}$ as the standard deviation of the estimates

Summarizing Cross Validation

Q: What do you do with all of these Cross Validation estimates?

- Let's define $\hat{\beta}_f$ as the estimate of interest for fold f
- Suppose there are F folds
- Compute $\bar{\hat{\beta}} = \frac{1}{F}\sum_{f=1}^{F}\hat{\beta}_f$ as the central tendency of the estimates
- Compute $\sqrt{\frac{1}{F}\sum_{f=1}^{F}(\hat{\beta}_f - \bar{\hat{\beta}})^2}$ as the standard deviation of the estimates
- Other sample metrics are computed easily.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Forecast Evaluation Measures

$$
\begin{aligned}
Y_{t+h|t} &= \beta_1 + \beta_2 X_t + u_{t+h} \\
\hat{Y}_{t+h|t} &= \hat{\beta}_1 + \hat{\beta}_2 X_t \\
\hat{u}_{t+h|t} &= Y_{t+h} - \hat{Y}_{t+h|t}
\end{aligned}
$$

Forecast Evaluation Measures

$$
\begin{aligned}
Y_{t+h|t} &= \beta_1 + \beta_2 X_t + u_{t+h} \\
\hat{Y}_{t+h|t} &= \hat{\beta}_1 + \hat{\beta}_2 X_t \\
\hat{u}_{t+h|t} &= Y_{t+h} - \hat{Y}_{t+h|t}
\end{aligned}
$$

- $ME = \frac{1}{T} \sum_{t=1}^{T} \hat{u}_{t+h|t}$
- $MSE = \frac{1}{T} \sum_{t=1}^{T} \hat{u}_{t+h|t}^2$
- $RMSE = \sqrt{MSE}$
- $MAE = \frac{1}{T} \sum_{t=1}^{T} |\hat{u}_{t+h|t}|$
- $MAPE = \frac{1}{T} \sum_{t=1}^{T} |\frac{\hat{u}_{t+h|t}}{Y_{t+h}}|$

Forecast Evaluation Test

Diebold-Mariano

Consider the errors from forecasting strategies: $\hat{u}^{(A)}_{t+h|t}$ and $\hat{u}^{(B)}_{t+h|t}$.

Impose a loss function: $L(\hat{u}^{(i)}_{t+h|t}) = (\hat{u}^{(i)}_{t+h|t})^2$

Define $\bar{d} = \frac{1}{T}\sum_{t=1}^{T}\left[L_t^{(A)} - L_t^{(B)}\right]$

$$H_0 : E[d] = 0$$
$$H_a : E[d] \neq 0$$

$$DM = \frac{\bar{d}}{V(\bar{d})^{1/2}} \sim N(0,1)$$

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X
  - Histogram of Errors

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X
  - Histogram of Errors

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X
  - Histogram of Errors
- Repeat the above diagnostics for squared errors to detect changes in the variability of the forecast accuracy

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X
  - Histogram of Errors
- Repeat the above diagnostics for squared errors to detect changes in the variability of the forecast accuracy
- Q: If you see a pattern in the error squared but not the errors, what does that signal?

Evaluating Errors

- Statistically summarize and compare errors in the training and testing samples
- Diagnostic plots of the errors are very useful. If you spot any type of pattern, that implies your model is missing something.
  - Plot Errors vs Time
  - Plot Errors vs Y
  - Plot Errors vs (each) X
  - Histogram of Errors
- Repeat the above diagnostics for squared errors to detect changes in the variability of the forecast accuracy
- Q: If you see a pattern in the error squared but not the errors, what does that signal?
- A: The model forecasts may not be biased, but the uncertainty of those forecasts vary through time or across Y or X

How Forecast?

Q: You've found the model with the best error properties. How forecast out of sample?

How Forecast?

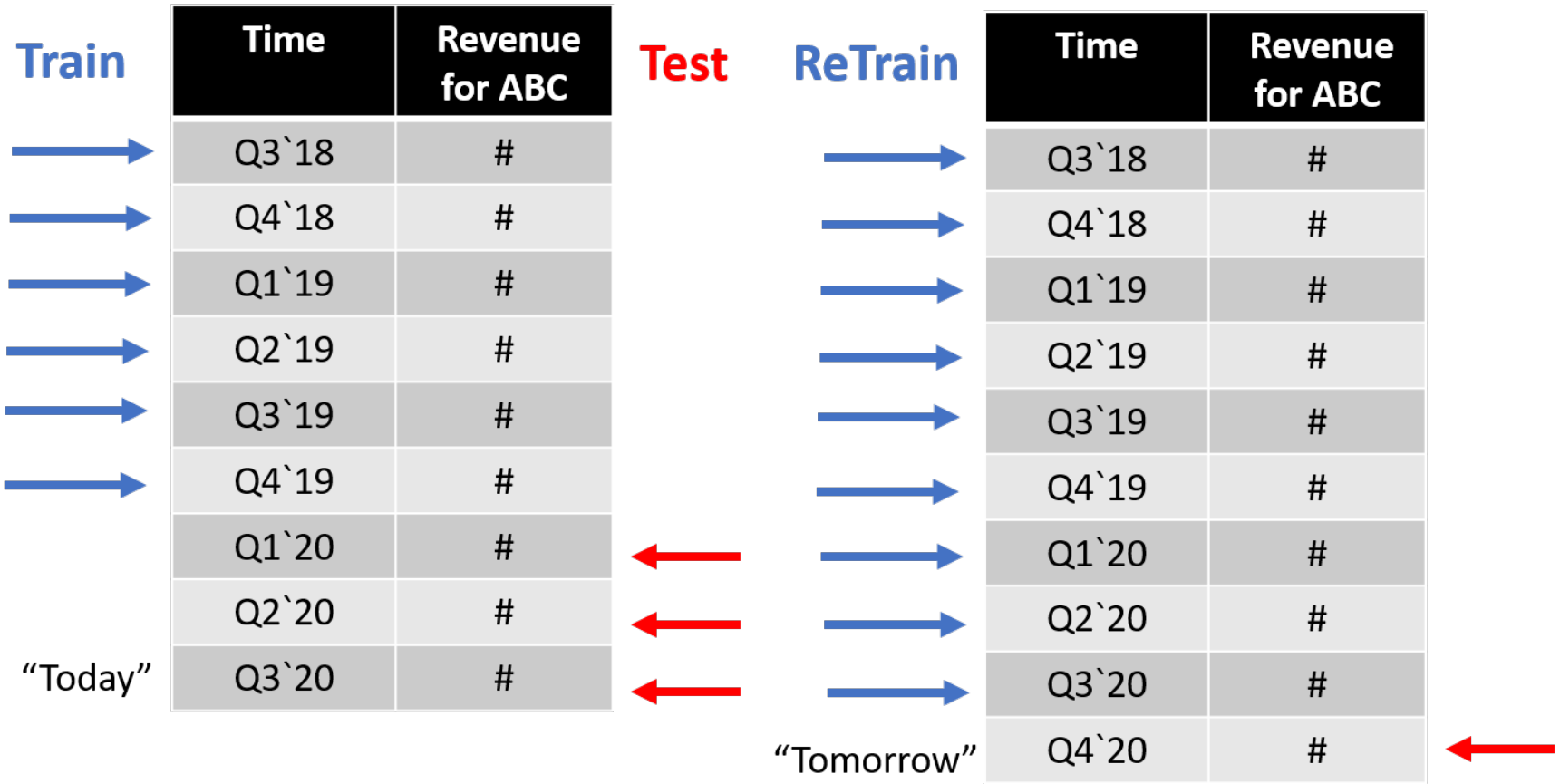
Q: You've found the model with the best error properties. How forecast out of sample?

A: We typically retrain the model on the full sample in order to use as much data as possible to inform the forecast.

How Forecast?

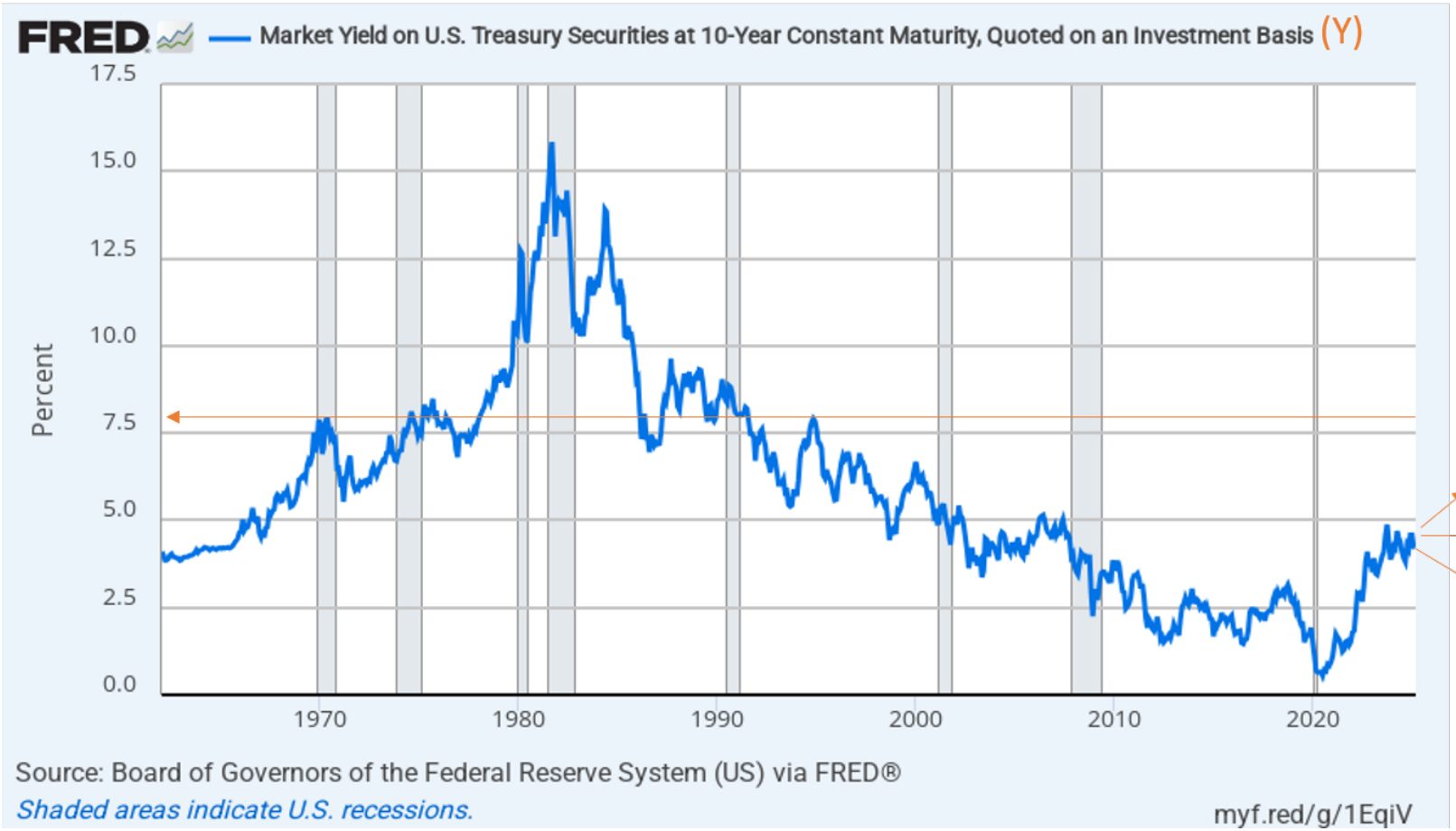
Q: You've found the model with the best error properties. How forecast out of sample?

A: We typically retrain the model on the full sample in order to use as much data as possible to inform the forecast.



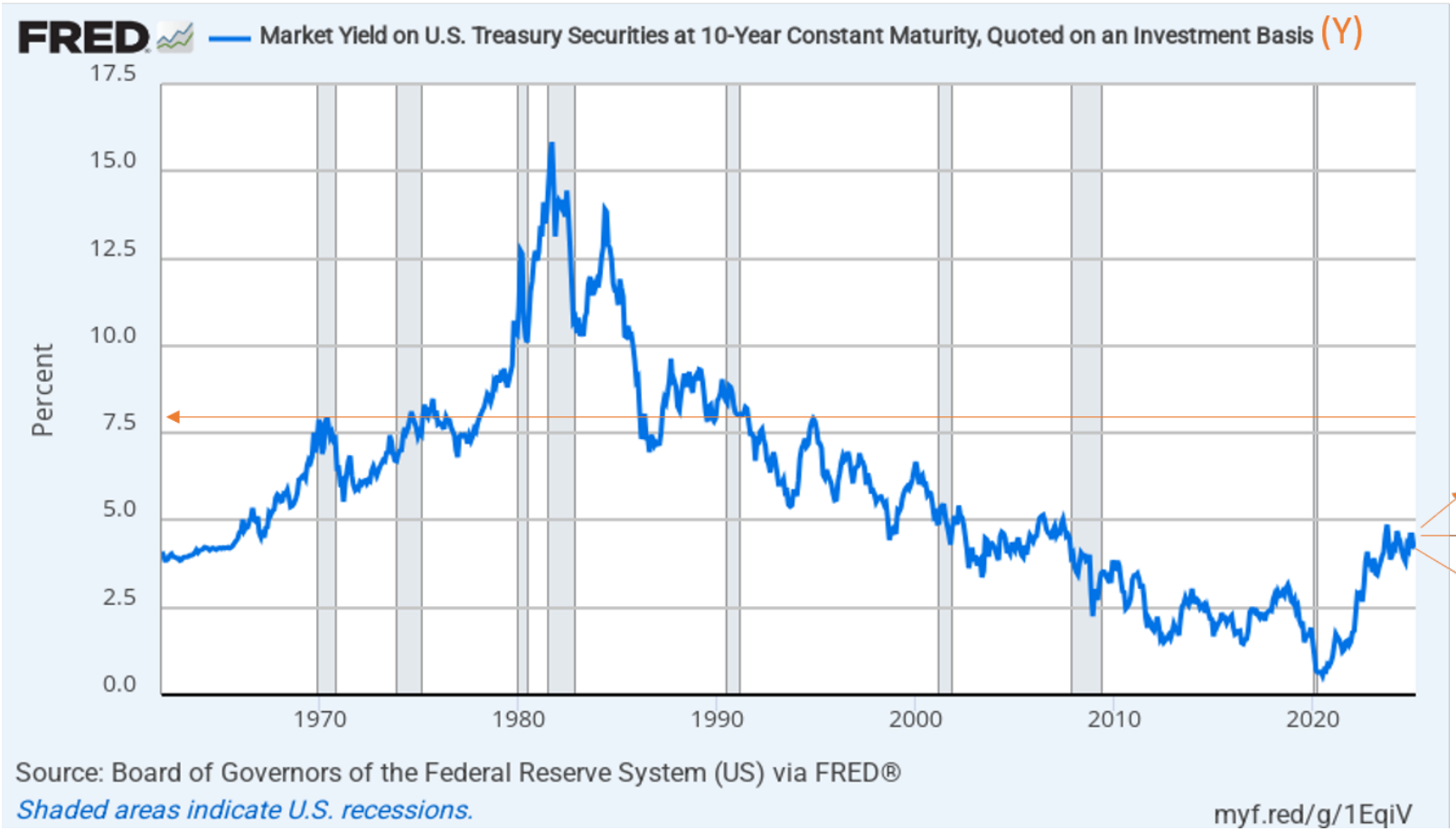
Prediction Intervals

Q: You've forecasted the 10yr yield at 4.75%. What does a 95% prediction interval of .75 mean?



Prediction Intervals

Q: You've forecasted the 10yr yield at 4.75%. What does a 95% prediction interval of .75 mean?



A: There is a 95% chance that the interval (4-5.5) will cover the future observation.

Forecast vs Confidence Intervals

Q: Which is wider?

Forecast vs Confidence Intervals

Q: Which is wider?

A: The confidence interval is:

$$\hat{Y} \pm t_{\alpha/2,n-2} \cdot se_{errors} \cdot \sqrt{\left(\frac{(x-\bar{X})^2}{SS_X} + \frac{1}{n}\right)}$$

Forecast vs Confidence Intervals

Q: Which is wider?

A: The confidence interval is:

$$\hat{Y} \pm t_{\alpha/2,n-2} \cdot se_{errors} \cdot \sqrt{\left(\frac{(x-\bar{X})^2}{SS_X} + \frac{1}{n}\right)}$$

The forecast interval is:

$$\hat{Y} \pm t_{\alpha/2,n-2} \cdot se_{errors} \cdot \sqrt{\left(\frac{(x-\bar{X})^2}{SS_X} + \frac{1}{n} + 1\right)}$$

Forecast vs Confidence Intervals

Q: Which is wider?

A: The confidence interval is:

$$\hat{Y} \pm t_{\alpha/2,n-2} \cdot se_{errors} \cdot \sqrt{\left(\frac{(x-\bar{X})^2}{SS_X} + \frac{1}{n}\right)}$$

The forecast interval is:

$$\hat{Y} \pm t_{\alpha/2,n-2} \cdot se_{errors} \cdot \sqrt{\left(\frac{(x-\bar{X})^2}{SS_X} + \frac{1}{n} + 1\right)}$$

- Forecast Intervals are wider (notice the extra "1"). Confidence intervals reflects uncertainty about the mean prediction. Forecast intervals reflects uncertainty about the mean and the individual observation for which you are forecasting.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

One Step Ahead Forecasting

One Step Ahead could literally mean tomorrow...

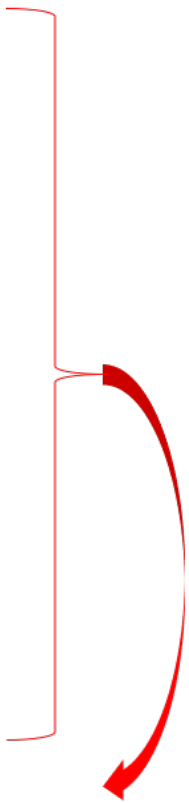
One Step Ahead Forecasting

One Step Ahead could literally mean tomorrow...

Time	Revenue for ABC
1Jan	#
2Jan	#
3Jan	#
4Jan	#
5Jan	#
6Jan	#
7Jan	#
8Jan	#
9Jan	#
10Jan	#

"Today"

"Tomorrow"



One Step Ahead Forecasting

...or could mean the next period

One Step Ahead Forecasting

...or could mean the next period

	Time	Revenue for ABC
	1Jan	#
	2Jan	#
	3Jan	#
	4Jan	#
	5Jan	#
	6Jan	#
	7Jan	#
	8Jan	#
"Today"	9Jan	#
"Tomorrow"	10Jan	#

Time	Revenue for ABC
Jan	#
Feb	#
Mar	#
Apr	#
May	#
Jun	#
Jul	#
Aug	#
Sep	#
Oct	#

Time	Revenue for ABC
Q3`18	#
Q4`18	#
Q1`19	#
Q2`19	#
Q3`19	#
Q4`19	#
Q1`20	#
Q2`20	#
Q3`20	#
Q4`20	#

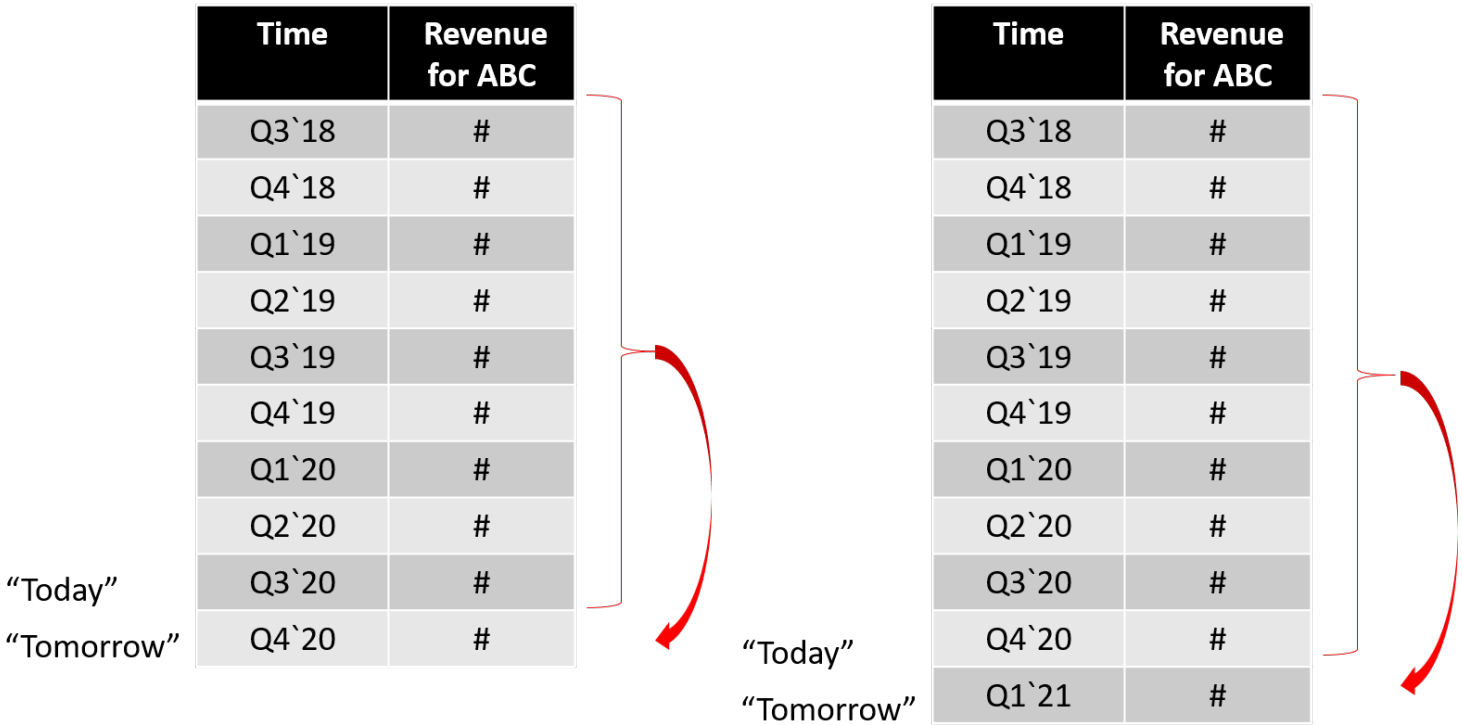
Rolling Forecasts

Q: What is a one step ahead rolling forecast?

Rolling Forecasts

Q: What is a one step ahead rolling forecast?

A: Each period we re-estimate the model to produce a one step ahead forecast.



Note: The training windows can i) retain a fixed length and slide forward or ii) grow (as shown here).

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q4'20}$$

which

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q4'20}$$

which uses the old estimates but feeds in the new data for NGDP.

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q4'20}$$

which uses the old estimates but feeds in the new data for NGDP.

Or we could re-estimate

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q4'20} NGDP_{Q4'20}$$

which

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q4'20}$$

which uses the old estimates but feeds in the new data for NGDP.

Or we could re-estimate

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q4'20} NGDP_{Q4'20}$$

which re-estimates the parameters on the new training period and then feeds in the new data for NGDP.

Rolling via Re-Forecasting or Re-Estimating

Suppose we had a model to forecast revenues $Rev_{Q4'20} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q3'20}$

To roll this forward one step ahead we could re-forecast

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q3'20} NGDP_{Q4'20}$$

which uses the old estimates but feeds in the new data for NGDP.

Or we could re-estimate

$$Rev_{Q1'21} = \hat{\beta}^{Q3'18-Q4'20} NGDP_{Q4'20}$$

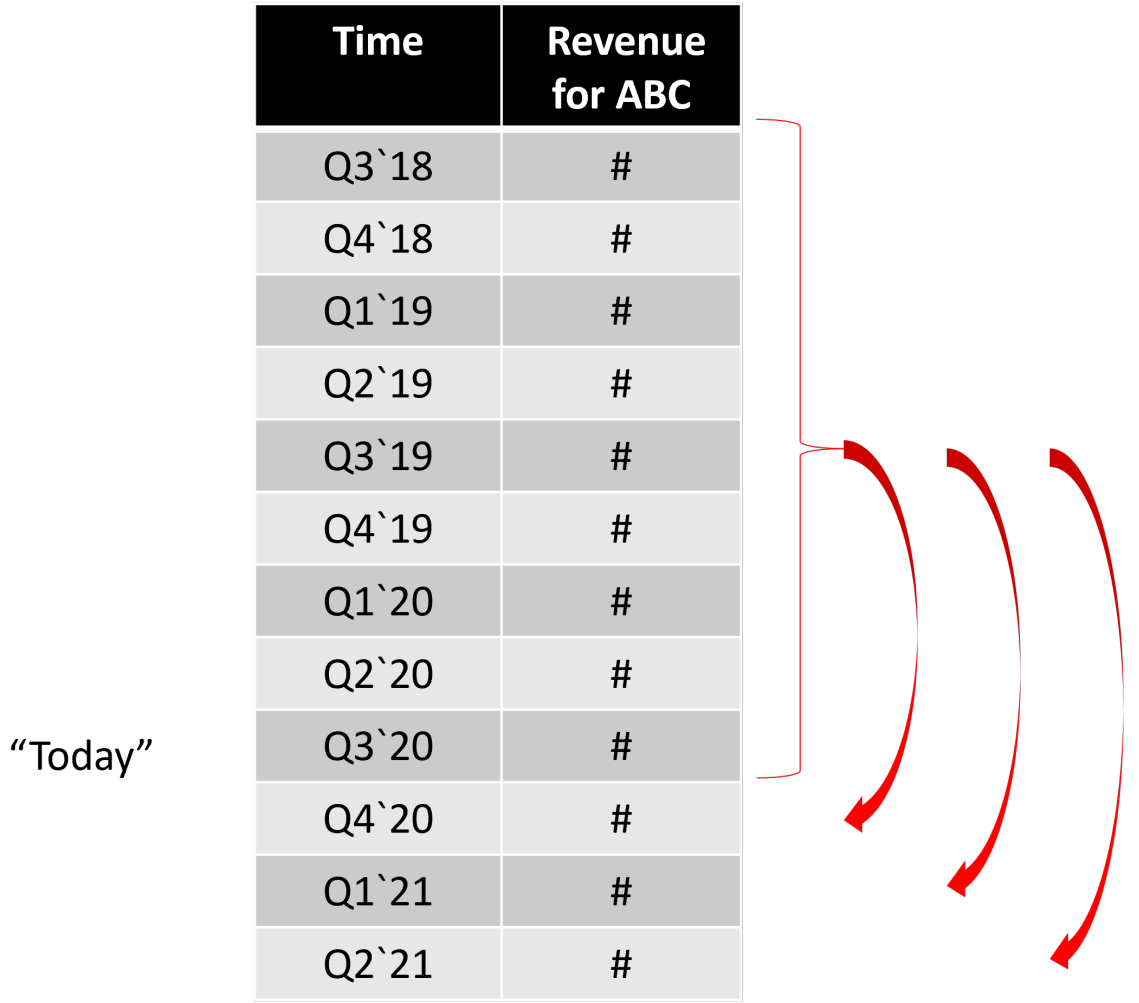
which re-estimates the parameters on the new training period and then feeds in the new data for NGDP.
If the model is stable, these should yield similar results.

Multi Step Ahead Forecasts - Static

Static Multi Step Ahead forecasts use the same set of information to forecast multiple horizons

Multi Step Ahead Forecasts - Static

Static Multi Step Ahead forecasts use the same set of information to forecast multiple horizons

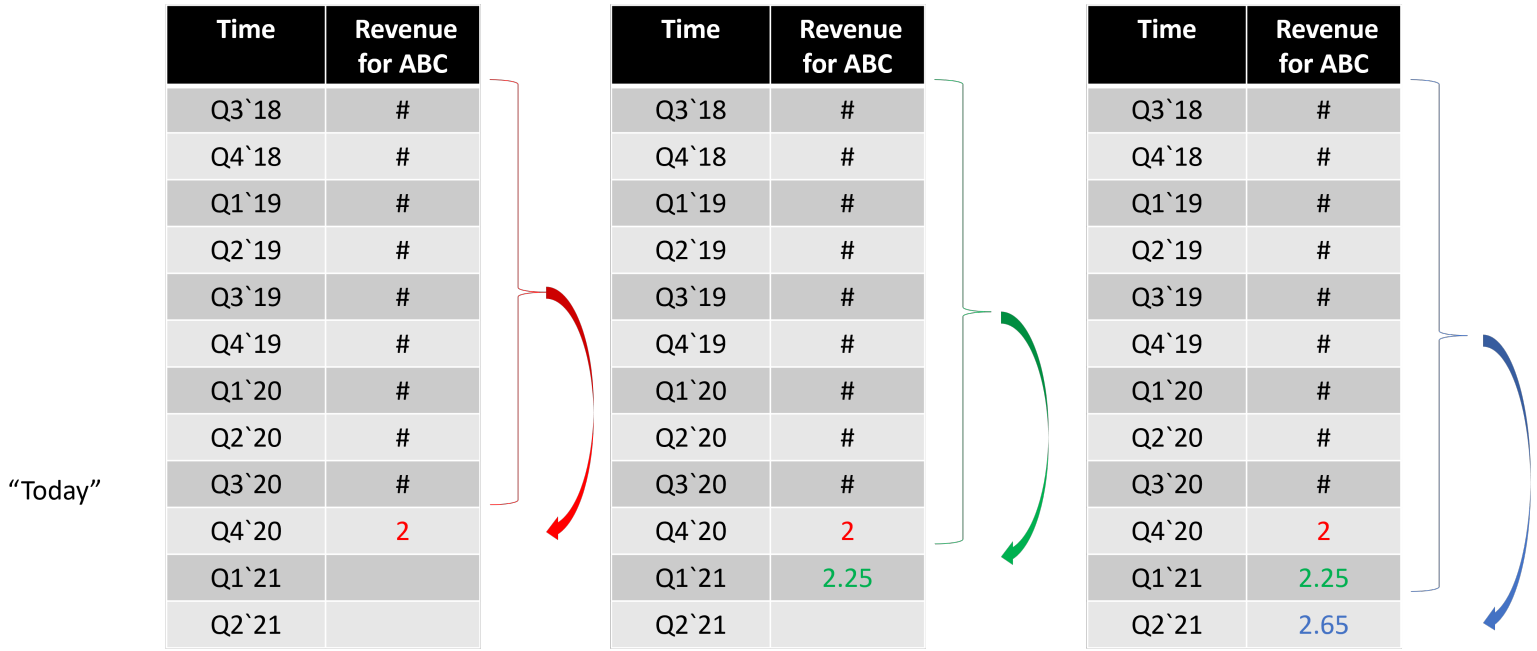


Multi Step Ahead Forecasts - Dynamic

Dynamic forecasts use the one step ahead estimates to forecast two steps ahead, etc..

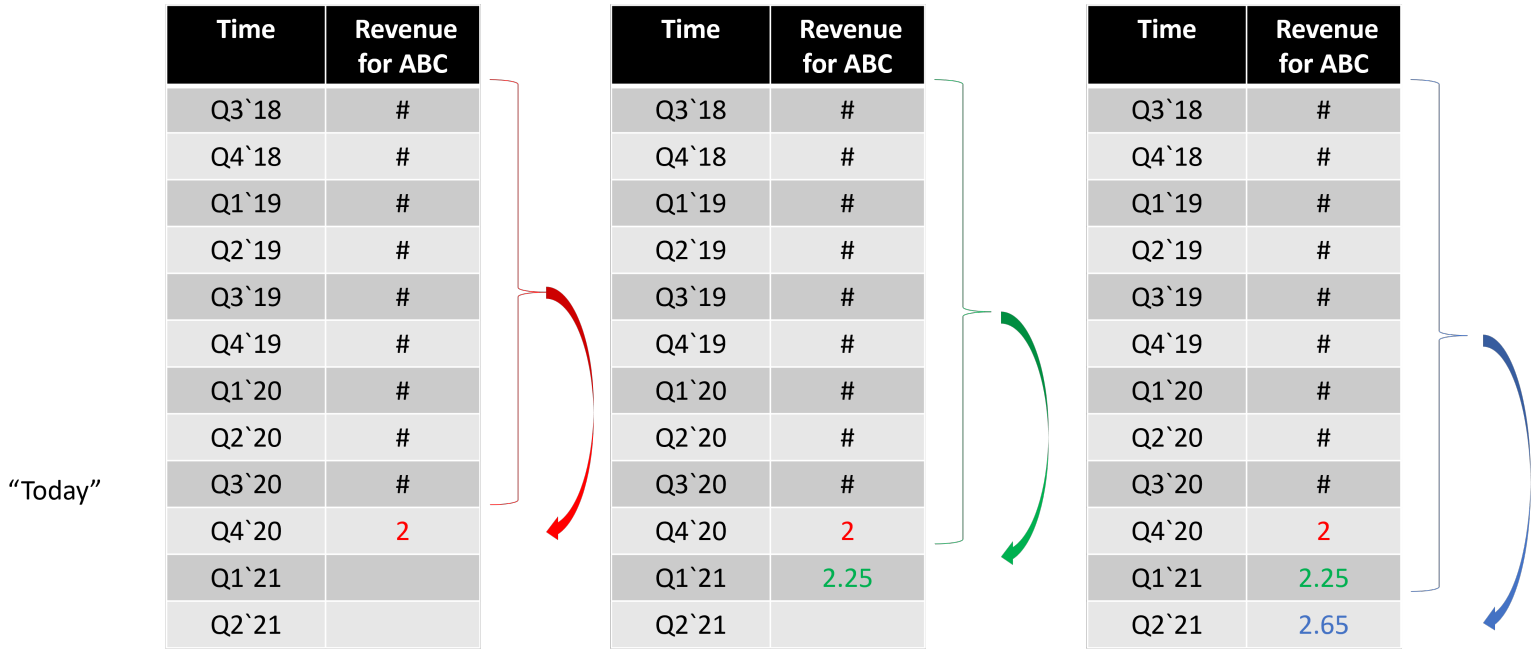
Multi Step Ahead Forecasts - Dynamic

Dynamic forecasts use the one step ahead estimates to forecast two steps ahead, etc..



Multi Step Ahead Forecasts - Dynamic

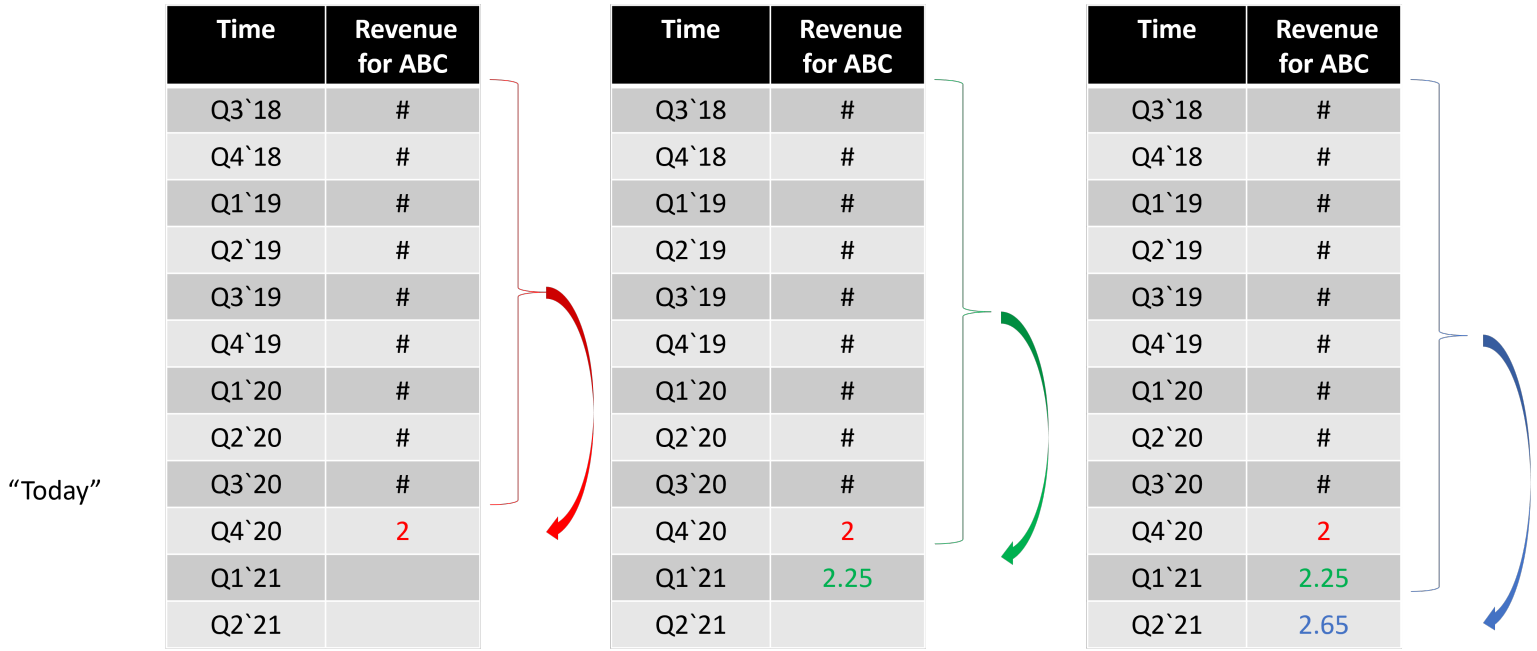
Dynamic forecasts use the one step ahead estimates to forecast two steps ahead, etc..



Q: What are (dis)advantages of Dynamic relative to Static?

Multi Step Ahead Forecasts - Dynamic

Dynamic forecasts use the one step ahead estimates to forecast two steps ahead, etc..



	Time	Revenue for ABC
	Q3`18	#
	Q4`18	#
	Q1`19	#
	Q2`19	#
	Q3`19	#
	Q4`19	#
	Q1`20	#
	Q2`20	#
"Today"	Q3`20	#
	Q4`20	2
	Q1`21	
	Q2`21	

Time	Revenue for ABC
Q3`18	#
Q4`18	#
Q1`19	#
Q2`19	#
Q3`19	#
Q4`19	#
Q1`20	#
Q2`20	#
Q3`20	#
Q4`20	2
Q1`21	2.25
Q2`21	

Time	Revenue for ABC
Q3`18	#
Q4`18	#
Q1`19	#
Q2`19	#
Q3`19	#
Q4`19	#
Q1`20	#
Q2`20	#
Q3`20	#
Q4`20	2
Q1`21	2.25
Q2`21	2.65

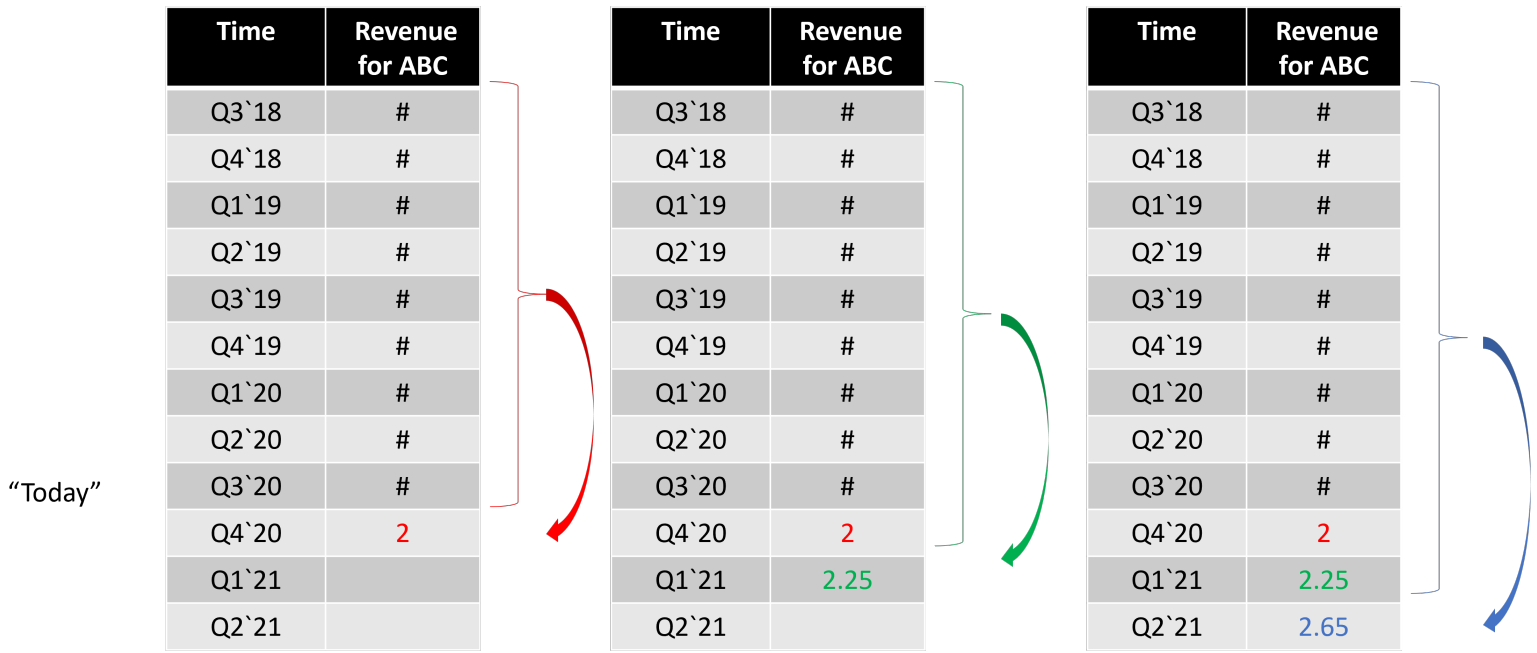
Q: What are (dis)advantages of Dynamic relative to Static?

A:

- Advantages: Can potentially extend far into the future; adapt quickly to new settings

Multi Step Ahead Forecasts - Dynamic

Dynamic forecasts use the one step ahead estimates to forecast two steps ahead, etc..



Q: What are (dis)advantages of Dynamic relative to Static?

A:

- Advantages: Can potentially extend far into the future; adapt quickly to new settings
- Disadvantage: Errors compound on each other

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Mean and Linear Time Trends

Mean and Linear Time Trends

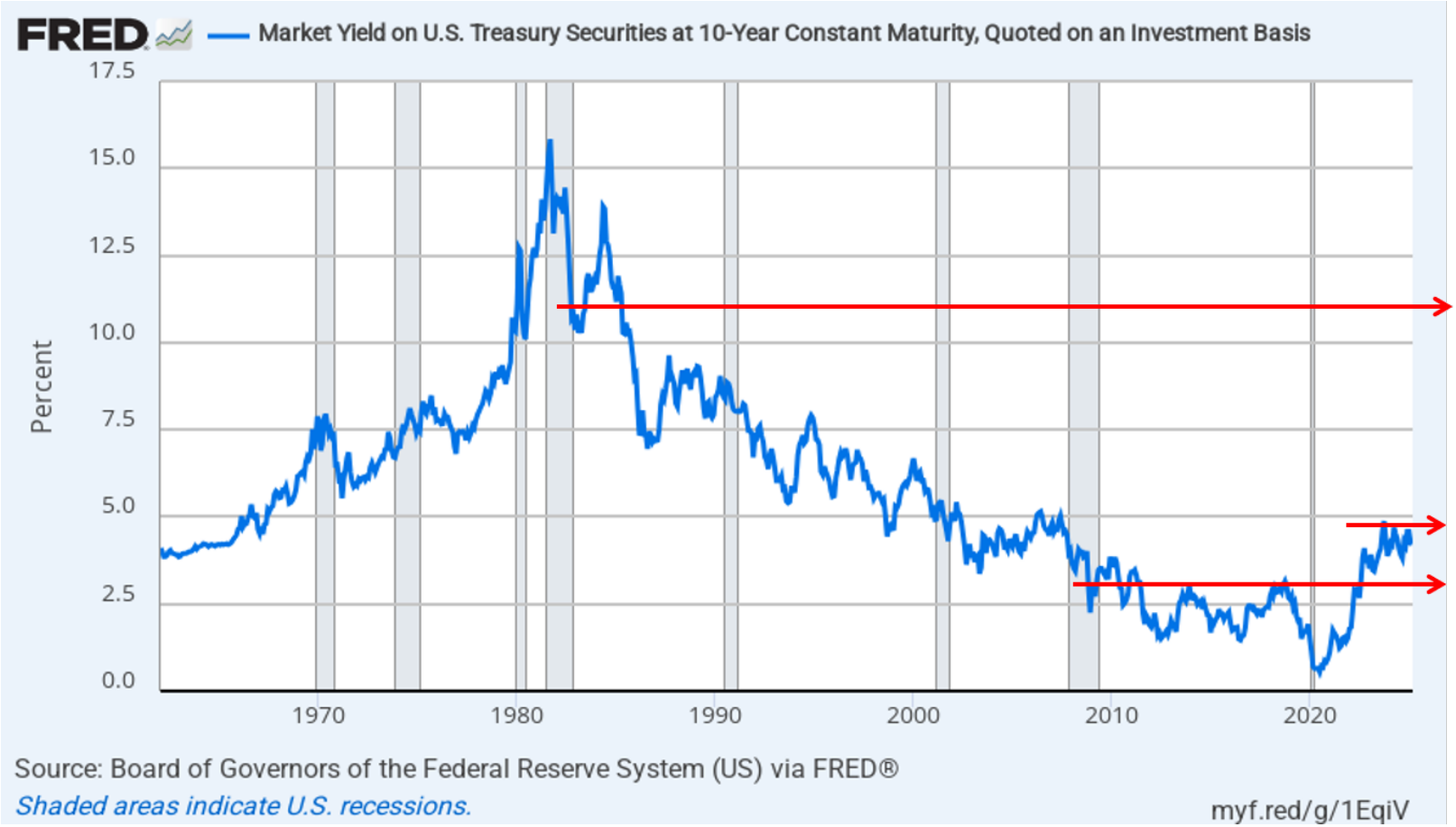
The two "simplest" models can serve as naive benchmarks:

$$\text{Mean Model: } E_t\left[Y_{t+h}\right] = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

$$\text{Linear Time Trend Model: } E_t\left[Y_{t+h}\right] = \hat{\alpha} + \hat{\beta} \times Time_{t+h}$$

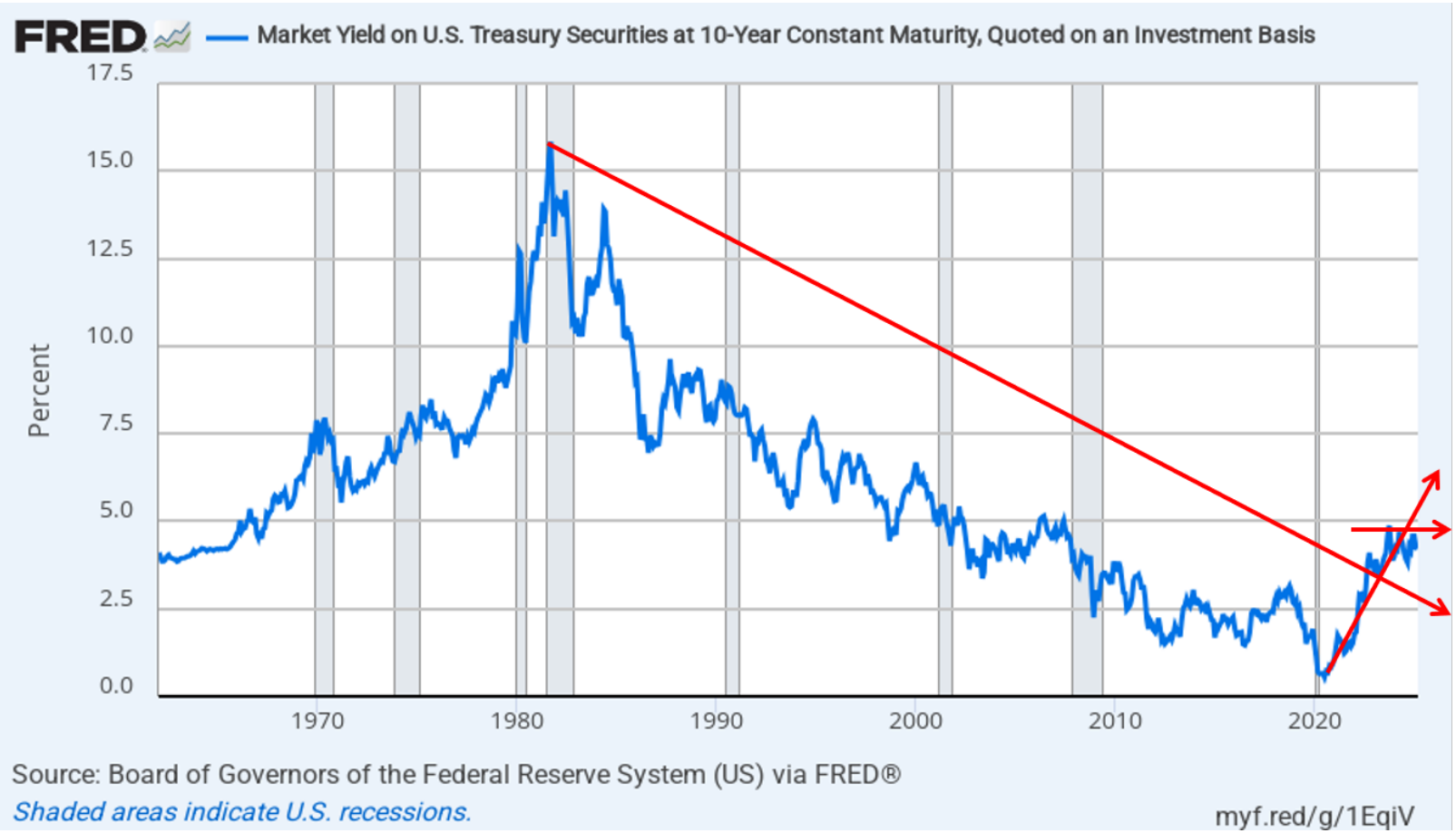
Mean Model

Though easy to compute, the size of training sample must be tuned



Linear Time Trend Model

Though easy to compute, the size of training sample must be tuned



Training Sample Selection

A few best practices:

Training Sample Selection

A few best practices:

- Training Sample should be larger than Testing Sample

Training Sample Selection

A few best practices:

- Training Sample should be larger than Testing Sample
- In time series, Training Sample should come before Testing Sample

Training Sample Selection

A few best practices:

- Training Sample should be larger than Testing Sample
- In time series, Training Sample should come before Testing Sample
- Ideally, Training Sample would "look like" the current period

Training Sample Selection

A few best practices:

- Training Sample should be larger than Testing Sample
- In time series, Training Sample should come before Testing Sample
- Ideally, Training Sample would "look like" the current period
- Ideally, Testing Sample would "look like" the out of sample period

Training Sample Selection

A few best practices:

- Training Sample should be larger than Testing Sample
- In time series, Training Sample should come before Testing Sample
- Ideally, Training Sample would "look like" the current period
- Ideally, Testing Sample would "look like" the out of sample period
- Try several sample splits and examine the range of forecasts for stability

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e.

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...
- ...they are univariate (i.e. Y_t is a vector, not matrix)

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...
- ...they are univariate (i.e. Y_t is a vector, not matrix)
- There are two ways in which this temporal persistence can manifest: AR and MA, or combinations thereof

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...
- ...they are univariate (i.e. Y_t is a vector, not matrix)
- There are two ways in which this temporal persistence can manifest: AR and MA, or combinations thereof
- Our job is to determine which type and what parameterization

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...
- ...they are univariate (i.e. Y_t is a vector, not matrix)
- There are two ways in which this temporal persistence can manifest: AR and MA, or combinations thereof
- Our job is to determine which type and what parameterization
- Imagery: every series has its own unique temporal "fingerprint" defined by the mean, variance, and autocorrelation of the series.

Classical Time Series Models

- Classical Time series models rely upon the temporal persistence of the outcome variable (i.e. there are no exogenous features), and...
- ...they are univariate (i.e. Y_t is a vector, not matrix)
- There are two ways in which this temporal persistence can manifest: AR and MA, or combinations thereof
- Our job is to determine which type and what parameterization
- Imagery: every series has its own unique temporal "fingerprint" defined by the mean, variance, and autocorrelation of the series.
- If we can identify the fingerprint, we can model the series.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Moving Average Process

$$Y_t = \mu + \varepsilon_t + \alpha\varepsilon_{t-1} \tag{1}$$

where $\varepsilon_t \sim N(0, \sigma^2)$ iid.

Moving Average Process

$$Y_t = \mu + \varepsilon_t + \alpha\varepsilon_{t-1} \tag{1}$$

where $\varepsilon_t \sim N(0, \sigma^2)$ iid.

Example (MA(1))

- Find the $E[Y_t]$.
- Find the $Var(Y_t)$.
- Find the autocovariance of Y over 1 lag i.e. $Cov(Y_t, Y_{t-1})$.
- Find the autocovariance of Y over 2 lags i.e. $Cov(Y_t, Y_{t-2})$.

Example (MA(1))

$$E[Y_t] = E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}]$$

Example (MA(1))

$$\begin{aligned} E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\ &= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \end{aligned}$$

Example (MA(1))

$$\begin{aligned} E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\ &= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \\ &= \mu \end{aligned}$$

Example (MA(1))

$$\begin{aligned} E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\ &= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \\ &= \mu \end{aligned}$$

$$Var(Y_t) = E[(\mu + \varepsilon_t + \alpha\varepsilon_{t-1} - \mu - \bar{\varepsilon}_t - \alpha\bar{\varepsilon}_{t-1})^2]$$

Example (MA(1))

$$\begin{aligned} E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\ &= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \\ &= \mu \end{aligned}$$

$$\begin{aligned} Var(Y_t) &= E[(\mu + \varepsilon_t + \alpha\varepsilon_{t-1} - \mu - \bar{\varepsilon}_t - \alpha\bar{\varepsilon}_{t-1})^2] \\ &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})^2] \end{aligned}$$

Example (MA(1))

$$\begin{aligned} E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\ &= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \\ &= \mu \end{aligned}$$

$$\begin{aligned} Var(Y_t) &= E[(\mu + \varepsilon_t + \alpha\varepsilon_{t-1} - \mu - \bar{\varepsilon}_t - \alpha\bar{\varepsilon}_{t-1})^2] \\ &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})^2] \\ &= E[\varepsilon_t^2] + \alpha^2 E[\varepsilon_{t-1}^2] \end{aligned}$$

Example (MA(1))

$$
\begin{aligned}
E[Y_t] &= E[\mu + \varepsilon_t + \alpha\varepsilon_{t-1}] \\
&= \mu + E[\varepsilon_t] + \alpha E[\varepsilon_{t-1}] \\
&= \mu
\end{aligned}
$$

$$
\begin{aligned}
Var(Y_t) &= E[(\mu + \varepsilon_t + \alpha\varepsilon_{t-1} - \mu - \bar{\varepsilon}_t - \alpha\bar{\varepsilon}_{t-1})^2] \\
&= E[(\varepsilon_t + \alpha\varepsilon_{t-1})^2] \\
&= E[\varepsilon_t^2] + \alpha^2 E[\varepsilon_{t-1}^2] \\
&= (1 + \alpha^2)\sigma^2
\end{aligned}
$$

Example (MA(1))

Example (MA(1))

$$Cov(Y_t, Y_{t-1}) = E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-1} + \alpha\varepsilon_{t-2})]$$

Example (MA(1))

$$\begin{aligned} Cov(Y_t, Y_{t-1}) &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-1} + \alpha\varepsilon_{t-2})] \\ &= \alpha E[\varepsilon_t^2] \end{aligned}$$

Example (MA(1))

$$\begin{aligned} Cov(Y_t, Y_{t-1}) &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-1} + \alpha\varepsilon_{t-2})] \\ &= \alpha E[\varepsilon_t^2] \\ &= \alpha\sigma^2 \end{aligned}$$

Example (MA(1))

$$\begin{aligned} Cov(Y_t, Y_{t-1}) &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-1} + \alpha\varepsilon_{t-2})] \\ &= \alpha E[\varepsilon_t^2] \\ &= \alpha\sigma^2 \end{aligned}$$

$$Cov(Y_t, Y_{t-2}) = E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-2} + \alpha\varepsilon_{t-3})]$$

Example (MA(1))

$$\begin{aligned} Cov(Y_t, Y_{t-1}) &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-1} + \alpha\varepsilon_{t-2})] \\ &= \alpha E[\varepsilon_t^2] \\ &= \alpha\sigma^2 \end{aligned}$$

$$\begin{aligned} Cov(Y_t, Y_{t-2}) &= E[(\varepsilon_t + \alpha\varepsilon_{t-1})(\varepsilon_{t-2} + \alpha\varepsilon_{t-3})] \\ &= 0 \end{aligned}$$

Moving Average Process

Definition

A moving average process of order q [MA(q)] can be represented by

$$Y_t = \mu + \varepsilon_t + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}$$

where $\varepsilon \sim N(0, \sigma^2)$ iid

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Autoregressive Process

Consider the $MA(\infty)$

$$Y_t = \mu + \sum_{j=0}^{\infty} \alpha_j \varepsilon_{t-j}$$

where $\alpha_0 \equiv 1$, and $\varepsilon \sim N(0, \sigma^2)$ iid. Assume that the α weights converge to zero as j gets large.

Autoregressive Process

Consider the $MA(\infty)$

$$Y_t = \mu + \sum_{j=0}^{\infty} \alpha_j \varepsilon_{t-j}$$

where $\alpha_0 \equiv 1$, and $\varepsilon \sim N(0, \sigma^2)$ iid. Assume that the α weights converge to zero as j gets large. Rewrite $\alpha_j = \theta^j$ for some θ, $|\theta| < 1$.

$$Y_t = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j}$$

Autoregressive Process

Notice from the last line above that

$$Y_{t-1} = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

Autoregressive Process

Notice from the last line above that

$$Y_{t-1} = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

and so we can write

$$Y_t - \theta Y_{t-1} = \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \theta \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

Autoregressive Process

Notice from the last line above that

$$Y_{t-1} = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

and so we can write

$$\begin{aligned} Y_t - \theta Y_{t-1} &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \theta \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1} \\ &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \sum_{j=1}^{\infty} \theta^j \varepsilon_{t-j} \end{aligned}$$

Autoregressive Process

Notice from the last line above that

$$Y_{t-1} = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

and so we can write

$$\begin{aligned} Y_t - \theta Y_{t-1} &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \theta \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1} \\ &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \sum_{j=1}^{\infty} \theta^j \varepsilon_{t-j} \\ Y_t &= \delta + \theta Y_{t-1} + \varepsilon_t \end{aligned}$$

where $\delta = \mu - \theta\mu$.

Autoregressive Process

Notice from the last line above that

$$Y_{t-1} = \mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}$$

and so we can write

$$\begin{aligned} Y_t - \theta Y_{t-1} &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \theta \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1} \\ &= \mu - \theta\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j} - \sum_{j=1}^{\infty} \theta^j \varepsilon_{t-j} \\ Y_t &= \delta + \theta Y_{t-1} + \varepsilon_t \end{aligned}$$

where $\delta = \mu - \theta\mu$. Defining $y_t = Y_t - \mu$ yields an AR(1) $y_t = \theta y_{t-1} + \varepsilon_t$

Example (AR(1))

Consider the AR(1) $y_t = \theta y_{t-1} + \varepsilon_t$, where $\varepsilon \sim N(0, \sigma^2)$ iid.

- Find the $E[y_t]$.
- Find the $Var(y_t)$.
- Find the autocovariance for Y_t and Y_{t-1}.
- Find the autocovariance for Y_t and Y_{t-2}.

Example (AR(1))

Example (AR(1))

$$E[y_t] = E[\theta y_{t-1} + \varepsilon_t]$$

Example (AR(1))

$$\begin{aligned} E[y_t] &= E[\theta y_{t-1} + \varepsilon_t] \\ &= \theta E[y_{t-1}] + E[\varepsilon_t] \end{aligned}$$

Example (AR(1))

$$\begin{aligned} E[y_t] &= E[\theta y_{t-1} + \varepsilon_t] \\ &= \theta E[y_{t-1}] + E[\varepsilon_t] \\ &= 0 \end{aligned}$$

Example (AR(1))

$$Var(y_t) = Var(y_t + \mu) = Var(Y_t)$$

Example (AR(1))

$$\begin{aligned} Var(y_t) &= Var(y_t + \mu) = Var(Y_t) \\ &= Var(\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \end{aligned}$$

Example (AR(1))

$$
\begin{aligned}
Var(y_t) &= Var(y_t + \mu) = Var(Y_t) \\
&= Var(\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= Var(\sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1})
\end{aligned}
$$

Example (AR(1))

$$
\begin{aligned}
Var(y_t) &= Var(y_t + \mu) = Var(Y_t) \\
&= Var(\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= Var(\sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= \sum_{j=0}^{\infty} Var(\theta^j \varepsilon_{t-j-1}) = \sum_{j=0}^{\infty} \theta^{2j} Var(\varepsilon_{t-j-1})
\end{aligned}
$$

Example (AR(1))

$$
\begin{aligned}
Var(y_t) &= Var(y_t + \mu) = Var(Y_t) \\
&= Var(\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= Var(\sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= \sum_{j=0}^{\infty} Var(\theta^j \varepsilon_{t-j-1}) = \sum_{j=0}^{\infty} \theta^{2j} Var(\varepsilon_{t-j-1}) \\
&= \sigma^2 \sum_{j=0}^{\infty} \theta^{2j}
\end{aligned}
$$

Example (AR(1))

$$
\begin{aligned}
Var(y_t) &= Var(y_t + \mu) = Var(Y_t) \\
&= Var(\mu + \sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= Var(\sum_{j=0}^{\infty} \theta^j \varepsilon_{t-j-1}) \\
&= \sum_{j=0}^{\infty} Var(\theta^j \varepsilon_{t-j-1}) = \sum_{j=0}^{\infty} \theta^{2j} Var(\varepsilon_{t-j-1}) \\
&= \sigma^2 \sum_{j=0}^{\infty} \theta^{2j} \\
&= \frac{\sigma^2}{1-\theta^2}
\end{aligned}
$$

Example (AR(1))

$$Cov(y_t, y_{t-1}) = E[y_t y_{t-1}]$$

Example (AR(1))

$$\begin{aligned} Cov(y_t, y_{t-1}) &= E[y_t y_{t-1}] \\ &= E[(\theta y_{t-1} + \varepsilon_t) y_{t-1}] \end{aligned}$$

Example (AR(1))

$$\begin{aligned} Cov(y_t, y_{t-1}) &= E[y_t y_{t-1}] \\ &= E[(\theta y_{t-1} + \varepsilon_t) y_{t-1}] \\ &= \theta E[y_{t-1}^2] + E[\varepsilon_t y_{t-1}] \end{aligned}$$

Example (AR(1))

$$
\begin{aligned}
Cov(y_t, y_{t-1}) &= E[y_t y_{t-1}] \\
&= E[(\theta y_{t-1} + \varepsilon_t) y_{t-1}] \\
&= \theta E[y_{t-1}^2] + E[\varepsilon_t y_{t-1}] \\
&= \theta Var(y_{t-1})
\end{aligned}
$$

Example (AR(1))

$$
\begin{aligned}
Cov(y_t, y_{t-1}) &= E[y_t y_{t-1}] \\
&= E[(\theta y_{t-1} + \varepsilon_t) y_{t-1}] \\
&= \theta E[y_{t-1}^2] + E[\varepsilon_t y_{t-1}] \\
&= \theta Var(y_{t-1}) \\
&= \theta \frac{\sigma^2}{1-\theta^2}
\end{aligned}
$$

Example (AR(1))

$$Cov(y_t, y_{t-2}) = E[y_t y_{t-2}]$$

Example (AR(1))

$$\begin{aligned} Cov(y_t, y_{t-2}) &= E[y_t y_{t-2}] \\ &= E[(\theta y_{t-1} + \varepsilon_t) y_{t-2}] \end{aligned}$$

Example (AR(1))

$$\begin{aligned} Cov(y_t, y_{t-2}) &= E[y_t y_{t-2}] \\ &= E[(\theta y_{t-1} + \varepsilon_t) y_{t-2}] \\ &= \theta E[y_{t-1} y_{t-2}] + E[\varepsilon_t y_{t-2}] \end{aligned}$$

Example (AR(1))

$$
\begin{aligned}
Cov(y_t, y_{t-2}) &= E[y_t y_{t-2}] \\
&= E[(\theta y_{t-1} + \varepsilon_t) y_{t-2}] \\
&= \theta E[y_{t-1} y_{t-2}] + E[\varepsilon_t y_{t-2}] \\
&= \theta \left\{ \theta \frac{\sigma^2}{1-\theta^2} \right\}
\end{aligned}
$$

Example (AR(1))

$$
\begin{aligned}
Cov(y_t, y_{t-2}) &= E[y_t y_{t-2}] \\
&= E[(\theta y_{t-1} + \varepsilon_t) y_{t-2}] \\
&= \theta E[y_{t-1} y_{t-2}] + E[\varepsilon_t y_{t-2}] \\
&= \theta \left\{ \theta \frac{\sigma^2}{1-\theta^2} \right\} \\
&= \theta^2 \frac{\sigma^2}{1-\theta^2}
\end{aligned}
$$

Autoregressive Process

Definition

An autoregressive process of order p [AR(p)] can be represented by

$$y_t = \sum_{j=1}^{p} \theta_j y_{t-j} + \varepsilon_t$$

where $\varepsilon \sim N(0, \sigma^2)$ iid

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

ARMA

Definition

An ARMA(p,q) process can be represented by

$$y_t = \sum_{j=1}^{p} \theta_j y_{t-j} + \varepsilon_t + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}$$

Estimation

Typical OLS estimation is difficult in MA or ARMA models.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.
But ε_{t-1} is not observed.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.
But ε_{t-1} is not observed.
Under certain (invertibility) conditions, we can write
$\varepsilon_{t-1} = \sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1}$, with a new objective function
$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1})^2$.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.
But ε_{t-1} is not observed.
Under certain (invertibility) conditions, we can write
$\varepsilon_{t-1} = \sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1}$, with a new objective function
$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1})^2$.
In practice, $y_0, y_{-1}, \ldots$ are not observed. So, we cut off the infinite sum and optimize

$$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{t-2}(-\alpha)^j y_{t-j-1})^2$$

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.
But ε_{t-1} is not observed.
Under certain (invertibility) conditions, we can write
$\varepsilon_{t-1} = \sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1}$, with a new objective function
$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1})^2$.
In practice, $y_0, y_{-1}, ...$ are not observed. So, we cut off the infinite sum and optimize

$$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{t-2}(-\alpha)^j y_{t-j-1})^2$$

Notice this is non linear.

Estimation

Typical OLS estimation is difficult in MA or ARMA models.
Consider an MA(1): $y_t = \varepsilon_t + \alpha\varepsilon_{t-1}$.
OLS would minimize $S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\varepsilon_{t-1})^2$.
But ε_{t-1} is not observed.
Under certain (invertibility) conditions, we can write
$\varepsilon_{t-1} = \sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1}$, with a new objective function
$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{\infty}(-\alpha)^j y_{t-j-1})^2$.
In practice, $y_0, y_{-1}, ...$ are not observed. So, we cut off the infinite sum and optimize

$$S(\alpha) = \sum_{t=2}^{T}(y_t - \alpha\sum_{j=0}^{t-2}(-\alpha)^j y_{t-j-1})^2$$

Notice this is non linear. Maximum Likelihood is the common alternative.

Model Choice via Box Jenkins

Rules of Thumb

Model Choice via Box Jenkins

Rules of Thumb

An AR(p) process is described by

1. An ACF with slow decay
2. A PACF that is (close to) zero for lags larger than p

Model Choice via Box Jenkins

Rules of Thumb

An AR(p) process is described by

1. An ACF with slow decay
2. A PACF that is (close to) zero for lags larger than p

An MA(q) process is described by

1. An ACF that is (close to) zero for lags larger than q
2. A PACF with slow decay

Model Choice via Box Jenkins

Rules of Thumb

An AR(p) process is described by

1. An ACF with slow decay
2. A PACF that is (close to) zero for lags larger than p

An MA(q) process is described by

1. An ACF that is (close to) zero for lags larger than q
2. A PACF with slow decay

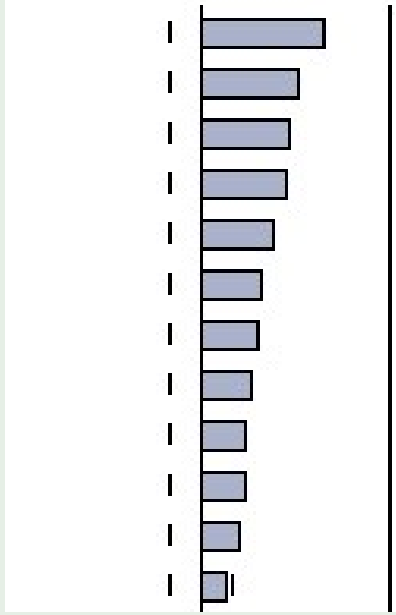
An ARMA process is described by

- Slow decay in both ACF and PACF

Example (Reading the Correlogram)

Q: What model specification seems appropriate given the correlogram below?

Date: 07/11/07 Time: 11:27
Sample: 1996M04 2007M06
Included observations: 135

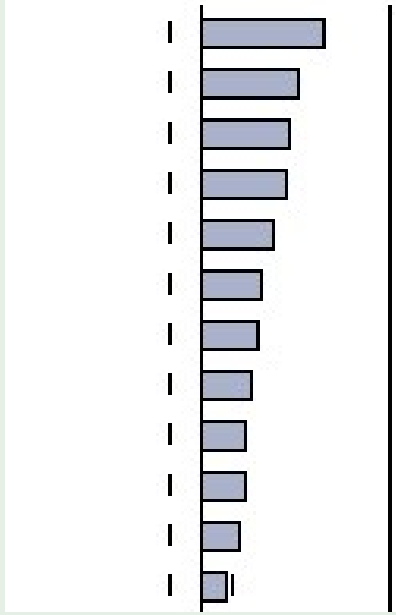


Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.680	0.680	63.865	0.000
		2	0.549	0.160	105.71	0.000
		3	0.495	0.139	140.06	0.000
		4	0.466	0.108	170.72	0.000
		5	0.402	0.001	193.72	0.000
		6	0.331	-0.031	209.45	0.000
		7	0.305	0.034	222.91	0.000
		8	0.271	-0.003	233.64	0.000
		9	0.248	0.021	242.68	0.000
		10	0.239	0.042	251.10	0.000
		11	0.213	-0.005	257.89	0.000
		12	0.143	-0.096	260.98	0.000

Example (Reading the Correlogram)

Q: What model specification seems appropriate given the correlogram below?

Date: 07/11/07 Time: 11:27
Sample: 1996M04 2007M06
Included observations: 135



Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
		1	0.680	0.680	63.865	0.000
		2	0.549	0.160	105.71	0.000
		3	0.495	0.139	140.06	0.000
		4	0.466	0.108	170.72	0.000
		5	0.402	0.001	193.72	0.000
		6	0.331	-0.031	209.45	0.000
		7	0.305	0.034	222.91	0.000
		8	0.271	-0.003	233.64	0.000
		9	0.248	0.021	242.68	0.000
		10	0.239	0.042	251.10	0.000
		11	0.213	-0.005	257.89	0.000
		12	0.143	-0.096	260.98	0.000

A: The slowly decaying ACF and the spike at 1 lag for the PACF suggest an AR(1) might be appropriate.

Diagnostic Checking

Diagnostic Checking

1. After fitting a potential model, conduct the "usual" overfitting and parameter tests of added variables (t and F tests).

Diagnostic Checking

1. After fitting a potential model, conduct the "usual" overfitting and parameter tests of added variables (t and F tests).
2. Plot the residuals and look for patterns as a sign of misspecification.

Diagnostic Checking

1. After fitting a potential model, conduct the "usual" overfitting and parameter tests of added variables (t and F tests).
2. Plot the residuals and look for patterns as a sign of misspecification.
3. Conduct a test of white noise on the residuals
   - H_0 : Residuals are W.N. vs H_a :Residuals are not W.N.
   - Test Statistic: $Q_k = T(T+2)\sum_{k=1}^{K} \frac{1}{T-k} r_k^2$, where r_k are the estimated autocorrelation coefficients of the residuals $\hat{\varepsilon}_t$.
   - Critical Value: $Q_k \sim \chi^2(k-p-q)$

Diagnostic Checking

1. After fitting a potential model, conduct the "usual" overfitting and parameter tests of added variables (t and F tests).
2. Plot the residuals and look for patterns as a sign of misspecification.
3. Conduct a test of white noise on the residuals
   - H_0 : Residuals are W.N. vs H_a :Residuals are not W.N.
   - Test Statistic: $Q_k = T(T+2)\sum_{k=1}^{K} \frac{1}{T-k} r_k^2$, where r_k are the estimated autocorrelation coefficients of the residuals $\hat{\varepsilon}_t$.
   - Critical Value: $Q_k \sim \chi^2(k-p-q)$
4. Search for the smallest AIC and/or BIC values
   - $AIC = \log(\hat{\sigma}^2) + 2(p+q)/T$
   - $BIC = \log(\hat{\sigma}^2) + log(T)(p+q)/T$

ARIMA

ARIMA is an extension of the ARMA structure,

AR: Autoregressive

I: Integrated

MA: Moving Average

The order of integration indicates the number of times the series must by differenced in order to achieve stationarity. A process that is $I(0)$ is stationary. A process that is $I(1)$ is said to be "integrated of order 1", implying that it must be differenced once to achieve stationarity.

ARIMA

ARIMA is an extension of the ARMA structure,

AR: Autoregressive

I: Integrated

MA: Moving Average

The order of integration indicates the number of times the series must by differenced in order to achieve stationarity. A process that is $I(0)$ is stationary. A process that is $I(1)$ is said to be "integrated of order 1", implying that it must be differenced once to achieve stationarity.

ARIMA

ARIMA is an extension of the ARMA structure,

AR: Autoregressive

I: Integrated

MA: Moving Average

The order of integration indicates the number of times the series must by differenced in order to achieve stationarity. A process that is $I(0)$ is stationary. A process that is $I(1)$ is said to be "integrated of order 1", implying that it must be differenced once to achieve stationarity.

Extension: Seasonal ARIMA

- SARIMA$(p, d, q) \times (P, D, Q)_s$
- Captures both short-run dynamics and seasonal patterns
- Seasonal differencing: $\Delta_s Y_t = Y_t - Y_{t-s}$
- Connection: Seasonal ARIMA underlies many decomposition tools (e.g. X-13 ARIMA-SEATS)

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

ARIMAX Motivation: Diagnosing Missing Structure

- Start with a simple AR(1) model:

$$r_t = \alpha + \phi r_{t-1} + \varepsilon_t$$

ARIMAX Motivation: Diagnosing Missing Structure

- Start with a simple AR(1) model:

$$r_t = \alpha + \phi r_{t-1} + \varepsilon_t$$

- Extract residuals $\hat{\varepsilon}_t$ and examine their structure

ARIMAX Motivation: Diagnosing Missing Structure

- Start with a simple AR(1) model:

$$r_t = \alpha + \phi r_{t-1} + \varepsilon_t$$

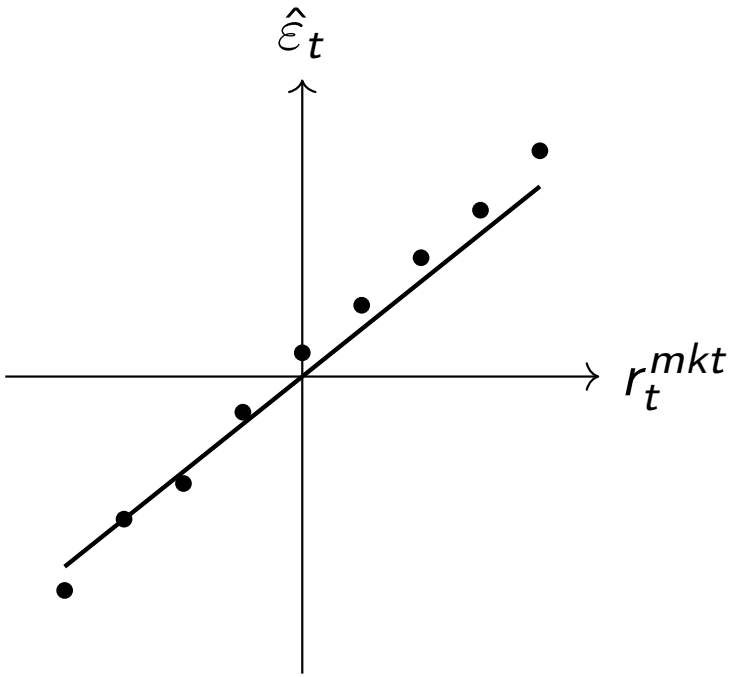
- Extract residuals $\hat{\varepsilon}_t$ and examine their structure
- Suppose we observe:
  - Residuals comove with market returns r_t^{mkt}
  - Suggests missing explanatory variable

ARIMAX Motivation: Diagnosing Missing Structure

- Start with a simple AR(1) model:

$$r_t = \alpha + \phi r_{t-1} + \varepsilon_t$$

- Extract residuals $\hat{\varepsilon}_t$ and examine their structure
- Suppose we observe:
  - Residuals comove with market returns r_t^{mkt}
  - Suggests missing explanatory variable

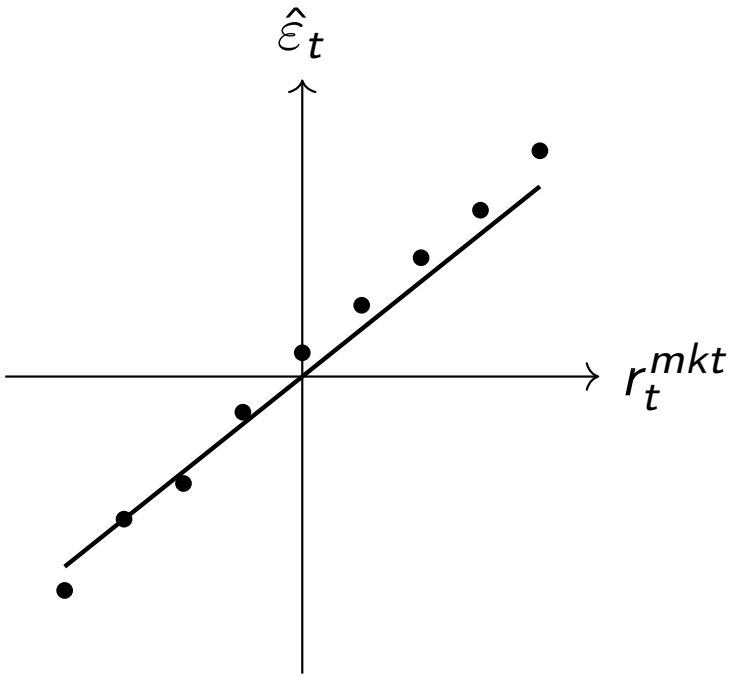


ARIMAX Motivation: Diagnosing Missing Structure

- Start with a simple AR(1) model:

$$r_t = \alpha + \phi r_{t-1} + \varepsilon_t$$

- Extract residuals $\hat{\varepsilon}_t$ and examine their structure
- Suppose we observe:
  - Residuals comove with market returns r_t^{mkt}
  - Suggests missing explanatory variable



- If residuals contain structure $\Rightarrow$ model is misspecified
- Add X_t to capture this variation
- $\Rightarrow$ ARIMAX

ARIMAX Model

- Extend ARIMA to include exogenous variables

ARIMAX Model

- Extend ARIMA to include exogenous variables

$$r_t = \alpha + \phi r_{t-1} + \beta X_t + \varepsilon_t$$

ARIMAX Model

- Extend ARIMA to include exogenous variables

$$r_t = \alpha + \phi r_{t-1} + \beta X_t + \varepsilon_t$$

- Examples of X_t:
  - Market returns
  - Interest rates
  - Macro indicators

ARIMAX Model

- Extend ARIMA to include exogenous variables

$$r_t = \alpha + \phi r_{t-1} + \beta X_t + \varepsilon_t$$

- Examples of X_t:
  - Market returns
  - Interest rates
  - Macro indicators
- Interpretation:
  - AR term: internal dynamics
  - X_t: external information

ARIMAX Model

- Extend ARIMA to include exogenous variables

$$r_t = \alpha + \phi r_{t-1} + \beta X_t + \varepsilon_t$$

- Examples of X_t:
  - Market returns
  - Interest rates
  - Macro indicators
- Interpretation:
  - AR term: internal dynamics
  - X_t: external information
- Important:
  - Must forecast X_{t+h} or use only lagged X
  - If forecasting X_{t+h}, then forecast errors may compound

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Case Study Background

Consider the Phillips Curve, which relates inflation (π) to unemployment (UR). Suppose each follows an AR(1), and by theory is related to the other.

$$\begin{aligned}\pi_t &= \delta_1 + \theta_{11}\pi_{t-1} + \theta_{12}UR_{t-1} + \epsilon_{1t} \\ UR_t &= \delta_2 + \theta_{21}\pi_{t-1} + \theta_{22}UR_{t-1} + \epsilon_{2t}\end{aligned}$$

where ϵ_{1t} and ϵ_{2t} are potentially correlated white noise processes, each independent of the history of UR and π.

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.
We can estimate each equation via OLS, but there are advantages to estimating simultaneously.

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.
We can estimate each equation via OLS, but there are advantages to estimating simultaneously.

- May be more parsimonious (e.g. by including more information you may need fewer lags of a single series)

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.
We can estimate each equation via OLS, but there are advantages to estimating simultaneously.

- May be more parsimonious (e.g. by including more information you may need fewer lags of a single series)
- Sims (1980) advocates the use of VAR's because there is less of a need to distinguish endogenous and exogenous variables apriori, which might otherwise force the econometrician to impose arbitrary restrictions.

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.
We can estimate each equation via OLS, but there are advantages to estimating simultaneously.

- May be more parsimonious (e.g. by including more information you may need fewer lags of a single series)
- Sims (1980) advocates the use of VAR's because there is less of a need to distinguish endogenous and exogenous variables apriori, which might otherwise force the econometrician to impose arbitrary restrictions.
- Possibly lead to more accurate forecasts.

The Case for VAR

The Phillips Curve theory suggests that $\theta_{12} \neq 0$ (or $\theta_{21} \neq 0$), implying that inflation is dependent upon the unemployment rate.
We can estimate each equation via OLS, but there are advantages to estimating simultaneously.

- May be more parsimonious (e.g. by including more information you may need fewer lags of a single series)
- Sims (1980) advocates the use of VAR's because there is less of a need to distinguish endogenous and exogenous variables apriori, which might otherwise force the econometrician to impose arbitrary restrictions.
- Possibly lead to more accurate forecasts.
- Greene (2000) suggests that simultaneous estimates might be more efficient in some situations.

A VAR(1) System

The VAR(1) specification can be written as

$$\begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ X_{t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

A VAR(1) System

The VAR(1) specification can be written as

$$\begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ X_{t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

or written compactly as

$$\overrightarrow{Y}_t = \delta + \Phi_1 \overrightarrow{Y}_{t-1} + \overrightarrow{\epsilon}_t$$

where $\overrightarrow{Y}_t = (Y_t, X_t)'$ and $\overrightarrow{\epsilon}_t = (\epsilon_{1t}, \epsilon_{2t})'$.

A VAR(p) System

Define $\overrightarrow{Y}_t$ as a $t \times k$ matrix. Then the VAR(p) model can be written as

$$\overrightarrow{Y}_t = \delta + \Phi_1 \overrightarrow{Y}_{t-1} + \cdots + \Phi_p \overrightarrow{Y}_{t-p} + \overrightarrow{\epsilon}_t$$

where each parameter matrix Φ is a $k \times k$ and the white noise errors $\overrightarrow{\epsilon}$ is a $k-$dimensional vector with covariance matrix Σ.

Expected Value

The expected value can be written as

$$E\{\overrightarrow{Y}_t\} = \delta + \Phi_1 E\{\overrightarrow{Y}_{t-1}\} + \cdots + \Phi_p E\{\overrightarrow{Y}_{t-p}\}$$

If we define the lag operator L and I_k as the $k-$dimensional Identity matrix, then we can rewrite as

$$\mu = E\{\overrightarrow{Y}_t\} = (I_k - \Phi_1 - \cdots - \Phi_p)^{-1}\delta = \Phi(1)^{-1}\delta$$

Note that stationarity requires that $\Phi(1)$ is invertible.

Covariance Estimation

Denote $e_{1t}, \cdots, e_{kt}$ as the residuals from each of the $k-$equations in the system. Then we can estimate the (i,j)-element of Σ as

$$\hat{\sigma}_{ij} = \frac{1}{T-p} \Sigma_{t=p+1}^{T} e_{it} e_{jt}$$

This permits us to form

$$\hat{\Sigma} = \frac{1}{T-p} \Sigma_{t=p+1}^{T} \overrightarrow{e}_t \overrightarrow{e}'_t$$

where $\overrightarrow{e}_t = (e_{1t}, \cdots, e_{kt})$.

Forecasting 1-step ahead

If we demean via $\overrightarrow{y}_t = \overrightarrow{Y}_t - \mu$, we can form

$$\overrightarrow{y}_t = \Phi_1 \overrightarrow{y}_{t-1} + \cdots + \Phi_p \overrightarrow{y}_{t-p} + \overrightarrow{\epsilon}_t$$

then we can form optimal one-period ahead forecasts via

$$\overrightarrow{y}_{T+1|T} = E\{\overrightarrow{y}_{T+1} | \overrightarrow{y}_T, \overrightarrow{y}_{T-1}, ...\} = \Phi_1 \overrightarrow{y}_T + \cdots + \Phi_p \overrightarrow{y}_{T-p+1}$$

with the accompanying forecast error variance

$$V\{\overrightarrow{y}_{T+1} | \overrightarrow{y}_T, \overrightarrow{y}_{T-1}, \cdots\} = \Sigma$$

Forecasting multiple steps ahead

Forecasting is accomplished via recursive substitution. For instance, for a

$$\begin{aligned}\overrightarrow{y}_{T+2|T} &= \Phi_1 \overrightarrow{y}_{T+1|T} + \cdots + \Phi_p \overrightarrow{y}_{T-p-2} \\ &= \Phi_1(\Phi_1 \overrightarrow{y}_T + \cdots + \Phi_p \overrightarrow{y}_{T-p+1}) + \cdots + \Phi_p \overrightarrow{y}_{T-p+2}\end{aligned}$$

Determining the Lag Length (p)

Determining the lag length (p) is not easy. We cannot use autocorrelograms (see Canova 1995).

Determining the Lag Length (p)

Determining the lag length (p) is not easy. We cannot use autocorrelograms (see Canova 1995).
A reasonable strategy (see Verbeek 2000) is to

1. Estimate the VAR over a range of p values
2. Pick the specification with the smallest AIC/BIC.

Impulse Response Function (IRF)

If $\Phi(1)$ is invertible, we can rewrite the VAR as a Vector Moving Average (VMA)

$$\overrightarrow{Y}_t = \mu + \Phi(L)^{-1}\overrightarrow{\epsilon}_t$$

Writing $\Phi(L)^{-1} = I_k + A_1 L + A_2 L^2 + \cdots$, we can see

$$\overrightarrow{Y}_t = \mu + \overrightarrow{\epsilon}_t + A_1\overrightarrow{\epsilon}_{t-1} + A_2\overrightarrow{\epsilon}_{t-2} + \cdots$$

Each element of any A matrix can be thought of as a marginal effect of a shock to the residual upon data in Y.

Impulse Response Function (IRF)

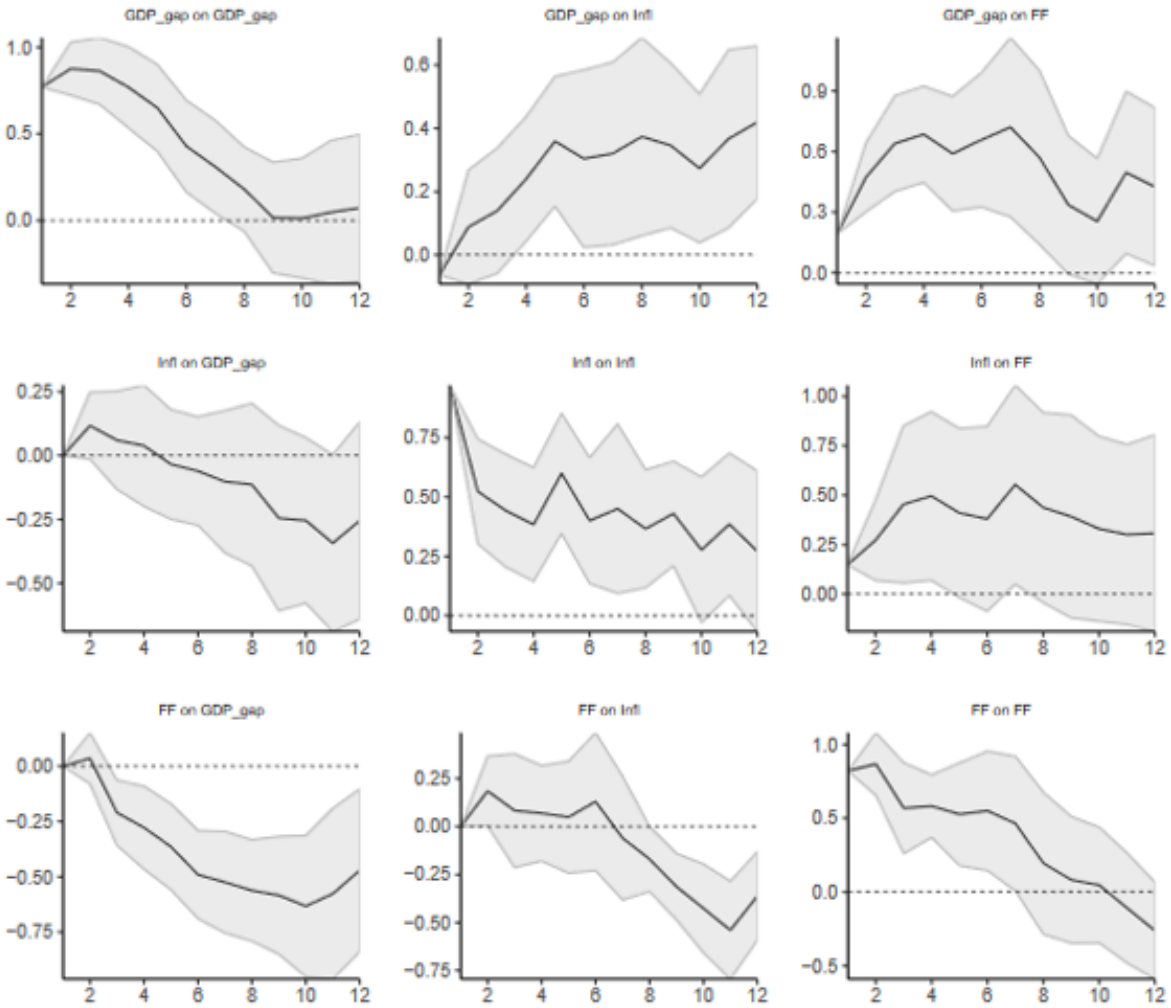
Consider again the Phillips Curve example

$$\begin{pmatrix} UR_t \\ \pi_t \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{pmatrix} \begin{pmatrix} UR_{t-1} \\ \pi_{t-1} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

Rewriting as a VMA we have $\overrightarrow{Y}_t = \mu + \overrightarrow{\epsilon}_t + A_1 \overrightarrow{\epsilon}_{t-1} + A_2 \overrightarrow{\epsilon}_{t-2} + \cdots$.
To examine the impact of a unit-one shock of unemployment upon inflation we consider the dynamics of the marginal effects

- $\frac{\partial Y_{2,t+1}}{\partial \epsilon_{1,t}} = A_{1,(1,2)}$
- $\frac{\partial Y_{2,t+2}}{\partial \epsilon_{1,t}} = A_{2,(1,2)}$
- $\cdots$

Impulse Response Function (IRF)



Multivariate Cointegration

Consider a binary VAR(1) model with variables integrated of order one ($\overrightarrow{Y}_t \sim I(1)$), we can write it as:

$$\overrightarrow{Y}_t = \delta + \Phi_1 \overrightarrow{Y}_{t-1} + \overrightarrow{\epsilon}_t$$

where matrix Φ_1 is a 2×2 and the white noise errors $\overrightarrow{\epsilon}$ is a $2-$dimensional vector with covariance matrix Σ.
By subtracting the lagged vectors $\overrightarrow{Y}_{t-1}$ from both side, we get:

$$\overrightarrow{Y}_t - \overrightarrow{Y}_{t-1} = \delta + \Phi_1 \overrightarrow{Y}_{t-1} - \overrightarrow{Y}_{t-1} + \epsilon_t$$
$$\Delta \overrightarrow{Y}_t = \delta + (\Phi_1 - I_2) \overrightarrow{Y}_{t-1} + \epsilon_t$$

Multivariate Cointegration

- Since we know Y_t is integrated of order one, so we have an $I(0)$ on the left hand side.
- The right hand side is the combination of two $I(0)$ (one constant vector δ and white noise vestor $\overrightarrow{\epsilon}$) and $(\Phi_1 - I_2)\overrightarrow{Y}_{t-1}$.
- If all variables are not cointegrated, then we must have:

$$(\Phi_1 - I_2) = \overrightarrow{0} \Rightarrow \Phi_1 = I_2$$

- If the variables are cointegrated, then there exists $r(0 < r \leq 2)$ cointegrated relations, $(\Phi_1 - I_2)\overrightarrow{Y}_{t-1}$ can be written as a $I(0)$ variable:

$$(\Phi_1 - I_2)\overrightarrow{Y}_{t-1} = \lambda\gamma'\overrightarrow{Y}_{t-1}$$

where γ' is the $(r \times 2)$ matrix of cointegration coefficients and λ is a $(2 \times r)$ matrix. Specifically, if $r = 2$, then all variables in $\overrightarrow{Y}_t$ must be stationary.

Multivariate Cointegration

The above structure can be extended to VAR(p) model with k variables with integrated of order d ($\overrightarrow{Y}_t \sim I(d)$):

$$\overrightarrow{Y}_t = \delta + \Phi_1 \overrightarrow{Y}_{t-1} + \cdots + \Phi_p \overrightarrow{Y}_{t-p} + \overrightarrow{\epsilon}_t$$

where each parameter matrix Φ is a $k \times k$ and the white noise errors $\overrightarrow{\epsilon}$ is a $k-$dimensional vector with covariance matrix Σ.
Apply difference operator $\Delta = 1 - L$ to both side, we get:

$$\begin{aligned}\Delta \overrightarrow{Y}_t =& \delta + (\sum_{j=1}^{p} \Phi_i - I_k) \overrightarrow{Y}_{t-1} + \Gamma_1 \Delta \overrightarrow{Y}_{t-1} + \cdots + \Delta \Gamma_p \overrightarrow{Y}_{t-p-1} + \overrightarrow{\epsilon}_t \\ =& \delta + \Pi \overrightarrow{Y}_{t-1} + \sum_{j=1}^{p-1} \Gamma_j \Delta \overrightarrow{Y}_{t-j} + \overrightarrow{\epsilon}_t\end{aligned}$$

This equation also called vector error correction model (VECM).

Multivariate Cointegration

$$\Delta \overrightarrow{Y}_t = \delta + \Pi \overrightarrow{Y}_{t-1} + \sum_{j=1}^{p-1} \Gamma_j \Delta \overrightarrow{Y}_{t-j} + \overrightarrow{\epsilon}_t$$

The rank of matrix Π determines number of cointegration vectors:

- If rank$(\Pi) = 0$, all variables in vector $\overrightarrow{Y}_t$ are non-stationary. There is no linear combination of variables that leads to stationary
- If Π has full rank k, then all variables in $\overrightarrow{Y}_t$ must be stationary
- If Π has reduce rank r $(0 < r < k)$, then there are cointegration relations among $\overrightarrow{Y}_t$

Vector Error Correction Model

Consider VECM equation:

$$\Delta \overrightarrow{Y}_t = \delta + \Pi \overrightarrow{Y}_{t-1} + \sum_{j=1}^{p-1} \Gamma_j \Delta \overrightarrow{Y}_{t-j} + \overrightarrow{\epsilon}_t$$

If Π has rank $r > 0$, then Π can be written as production of A and B':

$$\Delta \overrightarrow{Y}_t = \delta + AB' \overrightarrow{Y}_{t-1} + \sum_{j=1}^{p-1} \Gamma_j \Delta \overrightarrow{Y}_{t-j} + \overrightarrow{\epsilon}_t$$

where A is a $(k \times r)$ matrix contains the cointegration vectors and B is a $(k \times r)$ matrix which contains the speed of adjustment parameters.

We call this new version VECM as restricted regression in the sense that it imposes the hypothesis that rank(Π)$>$ 0, so the original VECM is unrestricted regression.

Johansen Test

As mentioned before, the Engle and Granger two-step test cannot be used to test cointegration involves more than two variables, so here we use Johansen test for multivariate cointegration testing.

Johansen test involves trace test and maximum eigenvalue test:

- Trace test:

$$\lambda_{trace} = -T \sum_{i=r+1}^{k} log(1 - \lambda_i)$$

$H_0 : r < k \Longleftrightarrow H_1 : r = k$

- Maximum eigenvalue test:

$$\lambda_{max} = -Tlog(1 - \lambda_{r+1})$$

$H_0 : r = q, q = 1, 2, \ldots, k \Longleftrightarrow H_1 : r \leq q + 1$

where λ_i is the eigenvalue value of matrix Π.

Granger Causality

In most cases, it is hard to tell the causality among variables from regression. For example:

$$y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

the significance of coefficient β_1 indicates that there exists relationship between variable y and X, but says nothing about X cause y.

Granger Causality

For causality, we have the following concepts:

- If X causes y, the changes of X happened before and were followed by the changes of y
- If X causes y, including X in regression helps improve forecasting of y, while reverse is wrong

Granger(1969) and Sims(1972) use VAR to test for the "Causality" in above sense, thus this causal relationship is called *Granger*(or *Granger-Sims*) *causality*.

Granger Causality

Consider a binary (y and X) system to keep things simple. Based on the "Causality" concepts:

- X is said to Granger-cause y if the current or lagged term of X help improve forecast X
- X fails to Granger-cause y if predict accuracy of $y_{t+s}(s > 0)$ based on information $(y_t, y_{t-1}, \dots)$ is the same as predict based on information $(y_t, y_{t-1}, \dots, X_t, X_{t-1}, \dots)$
- If we restrict the relationship between X and y are linear, we may use MSE to measure forecasting performance and the above condition can be rewrite as: X fails to Granger-cause y if

$$MSE[\hat{E}(y_{t+s}|y_t, y_{t-1}, \dots)] = MSE[\hat{E}(y_{t+s}|y_t, y_{t-1}, \dots X_t, X_{t-1}, \dots)]$$

Granger Causality

Example (VAR Formulation)

The binary VAR(p) specification can be written as

$$\begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} + \begin{pmatrix} \Phi_{11}^1 & \Phi_{12}^1 \\ \Phi_{21}^1 & \Phi_{22}^1 \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ X_{t-1} \end{pmatrix} + \cdots + \begin{pmatrix} \Phi_{11}^p & \Phi_{12}^p \\ \Phi_{21}^p & \Phi_{22}^p \end{pmatrix} \begin{pmatrix} Y_{t-p} \\ X_{t-p} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}$$

If X fails to Granger-cause y, then we have:

$$\Phi_{12}^1 = \Phi_{12}^2 = \cdots = \Phi_{12}^p = 0$$

that is this regression has a lower triangle coefficient matrix.

Granger Causality

Consider a linear regression of X_t on the past, current and future value of y's:

$$X_t = c + \sum_{j=0}^{\infty} b_j y_{t-j} + \sum_{j=0}^{\infty} d_j y_{t+j} + \epsilon_t$$

where the exogenous assumption is satisfied ($E[\epsilon_t y_s] = 0$ for all t and s), then X fails to Granger-cause y if and only if $d_j = 0$ for $j = 1, 2, \ldots$

Similarity, we can use this regression to test whether y Granger-cause X:

$$y_t = c + \sum_{j=0}^{\infty} b_j X_{t-j} + \sum_{j=0}^{\infty} d_j X_{t+j} + v_t$$

where the exogenous assumption is satisfied ($E[v_t X_s] = 0$ for all t and s), then y fails to Granger-cause X if and only if $d_j = 0$ for $j = 1, 2, \ldots$

Granger Causality

Causality model:

$$y_t = c_0 + \sum_{i=0}^{p_y} \alpha_i y_{t-i} + \sum_{j=0}^{p_x} \beta_j x_{t-j} + u_t$$

where p_x and p_y are the order of AR model of each series which are determined by F-test and/or AIC, BIC.

Granger-Causality test:

- Test: H_0 (X fail to Granger-cause y): $\beta_1 = \cdots = \beta_{p_x} = 0$
- Estimate the unrestricted regression, keep RSS_u
- Estimate the restricted regression ($\beta_1 = \cdots = \beta_{p_x} = 0$), keep RSS_R
- Construct F-test: $F = [(T-k)/q] \times [(RSS_R - RSS_U)/RSS_U]$, where $k = (1 + p_x + p_y)$ and $q = (1 + p_y)$

(Similar test can be conducted to test whether y Granger-cause X)

Granger Causality

There are four possible outcome for Granger-Causality test:

- X Granger-cause y, but y fails to Granger-cause X
- y Granger-cause X, but X fails to Granger-cause y
- X Granger-cause y and y Granger-cause X
- X fails to Granger-cause y and y fails to Granger-cause X

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

- Earlier, we applied dynamic econometric models to financial returns. The same logic applies to monthly sales of a product (SKU).

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

- Earlier, we applied dynamic econometric models to financial returns. The same logic applies to monthly sales of a product (SKU).
- Let $s_t^{(i)}$ be monthly dollar sales of SKU i, and let $p_t^{(i)}$ be promotional spending already observed at the end of month t.

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

- Earlier, we applied dynamic econometric models to financial returns. The same logic applies to monthly sales of a product (SKU).
- Let $s_t^{(i)}$ be monthly dollar sales of SKU i, and let $p_t^{(i)}$ be promotional spending already observed at the end of month t.
- A one-month-ahead ADL specification is

$$s_{t+1}^{(i)} = \alpha + \phi s_t^{(i)} + \beta p_t^{(i)} + u_{t+1}^{(i)}.$$

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

- Earlier, we applied dynamic econometric models to financial returns. The same logic applies to monthly sales of a product (SKU).
- Let $s_t^{(i)}$ be monthly dollar sales of SKU i, and let $p_t^{(i)}$ be promotional spending already observed at the end of month t.
- A one-month-ahead ADL specification is

$$s_{t+1}^{(i)} = \alpha + \phi s_t^{(i)} + \beta p_t^{(i)} + u_{t+1}^{(i)}.$$

- The lagged-sales term accommodates **persistence**; the promotion term introduces an additional sales driver.

A Familiar Starting Point: ADL Forecasting

Q: How might we forecast monthly sales using familiar econometric methods?

- Earlier, we applied dynamic econometric models to financial returns. The same logic applies to monthly sales of a product (SKU).
- Let $s_t^{(i)}$ be monthly dollar sales of SKU i, and let $p_t^{(i)}$ be promotional spending already observed at the end of month t.
- A one-month-ahead ADL specification is

$$s_{t+1}^{(i)} = \alpha + \phi s_t^{(i)} + \beta p_t^{(i)} + u_{t+1}^{(i)}.$$

- The lagged-sales term accommodates **persistence**; the promotion term introduces an additional sales driver.
- This model imposes a **linear, additive conditional mean** with constant coefficients, unless we enrich the specification.

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

- **Thresholds:** Very small promotions might have almost no effect. Once spending crosses a threshold, sales could respond sharply.

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

- **Thresholds:** Very small promotions might have almost no effect. Once spending crosses a threshold, sales could respond sharply.
- **Diminishing returns:** Doubling promotional spending may not double its impact on sales.

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

- **Thresholds:** Very small promotions might have almost no effect. Once spending crosses a threshold, sales could respond sharply.
- **Diminishing returns:** Doubling promotional spending may not double its impact on sales.
- **Interactions:** The effect of a promotion may differ between holiday months and ordinary months, or between rising and falling sales regimes.

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

- **Thresholds:** Very small promotions might have almost no effect. Once spending crosses a threshold, sales could respond sharply.
- **Diminishing returns:** Doubling promotional spending may not double its impact on sales.
- **Interactions:** The effect of a promotion may differ between holiday months and ordinary months, or between rising and falling sales regimes.
- Our simple ADL assumes that one additional dollar of promotion changes conditional expected sales by the same β, holding other inputs fixed.

Where Might the ADL Be Too Restrictive?

Q: What if the response of sales to promotions is nonlinear?

- **Thresholds:** Very small promotions might have almost no effect. Once spending crosses a threshold, sales could respond sharply.
- **Diminishing returns:** Doubling promotional spending may not double its impact on sales.
- **Interactions:** The effect of a promotion may differ between holiday months and ordinary months, or between rising and falling sales regimes.
- Our simple ADL assumes that one additional dollar of promotion changes conditional expected sales by the same β, holding other inputs fixed.
- **Important:** We could extend regression with interactions, dummies, polynomial terms, or splines. ML becomes attractive when many potential nonlinearities are difficult to specify in advance.

From a Linear ADL to Flexible Machine Learning

Q: Can familiar ML models learn the nonlinear response?

From a Linear ADL to Flexible Machine Learning

Q: Can familiar ML models learn the nonlinear response?

- Replace the restricted linear conditional-mean function with

$$s_{t+1}^{(i)} = f\left(s_t^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \ldots\right) + u_{t+1}^{(i)}.$$

Here f need not be linear or additive.

From a Linear ADL to Flexible Machine Learning

Q: Can familiar ML models learn the nonlinear response?

- Replace the restricted linear conditional-mean function with

$$s_{t+1}^{(i)} = f\left(s_t^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \ldots\right) + u_{t+1}^{(i)}.$$

  Here f need not be linear or additive.

- A regression tree can identify **thresholds**; a forest can stabilize trees; boosting can build a more flexible prediction function. Feedforward neural networks are another option.

From a Linear ADL to Flexible Machine Learning

Q: Can familiar ML models learn the nonlinear response?

- Replace the restricted linear conditional-mean function with

$$s_{t+1}^{(i)} = f\left(s_t^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \ldots\right) + u_{t+1}^{(i)}.$$

  Here f need not be linear or additive.

- A regression tree can identify **thresholds**; a forest can stabilize trees; boosting can build a more flexible prediction function. Feedforward neural networks are another option.

- For example, a tree may learn a large promotional response only when recent sales have weakened *and* a holiday is approaching.

From a Linear ADL to Flexible Machine Learning

Q: Can familiar ML models learn the nonlinear response?

- Replace the restricted linear conditional-mean function with

$$s_{t+1}^{(i)} = f\left(s_t^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \ldots\right) + u_{t+1}^{(i)}.$$

  Here f need not be linear or additive.

- A regression tree can identify **thresholds**; a forest can stabilize trees; boosting can build a more flexible prediction function. Feedforward neural networks are another option.

- For example, a tree may learn a large promotional response only when recent sales have weakened *and* a holiday is approaching.

- **But:** Flexible ML does not automatically know what happened in previous months. That depends on the information we give it.

Why Ordinary Tabular ML Is Not Yet Enough

Q: Can a tree capture sales persistence if we omit sales history?

Why Ordinary Tabular ML Is Not Yet Enough

Q: Can a tree capture sales persistence if we omit sales history?

- Suppose we train a regression tree using current sales, promotional spending, price, and the month of the year.

Why Ordinary Tabular ML Is Not Yet Enough

Q: Can a tree capture sales persistence if we omit sales history?

- Suppose we train a regression tree using current sales, promotional spending, price, and the month of the year.
- Consider two hypothetical sales histories. Current sales and all other model inputs are identical, but the preceding sales trajectories differ.

	Not given to tree		**Given to tree**			
Scenario	s_{t-2}	s_{t-1}	s_t	Promotion	Price	Month
A (rising)	70	80	90	20	10	June
B (falling)	110	100	90	20	10	June

Why Ordinary Tabular ML Is Not Yet Enough

Q: Can a tree capture sales persistence if we omit sales history?

- Suppose we train a regression tree using current sales, promotional spending, price, and the month of the year.
- Consider two hypothetical sales histories. Current sales and all other model inputs are identical, but the preceding sales trajectories differ.

	Not given to tree		**Given to tree**			
Scenario	s_{t-2}	s_{t-1}	s_t	Promotion	Price	Month
A (rising)	70	80	90	20	10	June
B (falling)	110	100	90	20	10	June

- Because the tree sees identical inputs in both scenarios, it must produce the **same forecast**, despite their different histories.

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

- Construct the model's input vector from information available at the forecast origin t:

$$\mathbf{x}_t^{(i)} = \left(s_t^{(i)}, s_{t-1}^{(i)}, s_{t-2}^{(i)}, s_{t-12}^{(i)}, \bar{s}_{t,3}^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \dots \right).$$

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

- Construct the model's input vector from information available at the forecast origin t:

$$\mathbf{x}_t^{(i)} = \left(s_t^{(i)}, s_{t-1}^{(i)}, s_{t-2}^{(i)}, s_{t-12}^{(i)}, \bar{s}_{t,3}^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \dots\right).$$

- Here $\bar{s}_{t,3}^{(i)} = (s_t^{(i)} + s_{t-1}^{(i)} + s_{t-2}^{(i)})/3$ is the trailing three-month average.

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

- Construct the model's input vector from information available at the forecast origin t:

$$\mathbf{x}_t^{(i)} = \left(s_t^{(i)}, s_{t-1}^{(i)}, s_{t-2}^{(i)}, s_{t-12}^{(i)}, \bar{s}_{t,3}^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \dots\right).$$

- Here $\bar{s}_{t,3}^{(i)} = (s_t^{(i)} + s_{t-1}^{(i)} + s_{t-2}^{(i)})/3$ is the trailing three-month average.
- We can also use growth rates, trailing volatility, seasonal lags, observed prices, or *already announced* future promotions.

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

- Construct the model's input vector from information available at the forecast origin t:

$$\mathbf{x}_t^{(i)} = \left(s_t^{(i)}, s_{t-1}^{(i)}, s_{t-2}^{(i)}, s_{t-12}^{(i)}, \bar{s}_{t,3}^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \dots\right).$$

- Here $\bar{s}_{t,3}^{(i)} = (s_t^{(i)} + s_{t-1}^{(i)} + s_{t-2}^{(i)})/3$ is the trailing three-month average.
- We can also use growth rates, trailing volatility, seasonal lags, observed prices, or *already announced* future promotions.
- This is **feature engineering**: we encode aspects of the history so that standard ML algorithms can learn relationships involving time.

Adapting ML to Time Series

Q: How do we make temporal persistence available to the algorithm?

- Construct the model's input vector from information available at the forecast origin t:

$$\mathbf{x}_t^{(i)} = \left(s_t^{(i)}, s_{t-1}^{(i)}, s_{t-2}^{(i)}, s_{t-12}^{(i)}, \bar{s}_{t,3}^{(i)}, p_t^{(i)}, \text{month}_{t+1}, \dots\right).$$

- Here $\bar{s}_{t,3}^{(i)} = (s_t^{(i)} + s_{t-1}^{(i)} + s_{t-2}^{(i)})/3$ is the trailing three-month average.
- We can also use growth rates, trailing volatility, seasonal lags, observed prices, or *already announced* future promotions.
- This is **feature engineering**: we encode aspects of the history so that standard ML algorithms can learn relationships involving time.
- The tree, forest, or boosting algorithm itself usually needs no new time-series-specific optimization procedure.

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

- At the end of month t, forecast SKU i's sales h months ahead:

$$\widehat{s}^{(i)}_{t+h|t} = f_h\left(\mathbf{x}^{(i)}_t; \widehat{\theta}_h\right).$$

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

- At the end of month t, forecast SKU i's sales h months ahead:

$$\widehat{s}^{(i)}_{t+h|t} = f_h\left(\mathbf{x}^{(i)}_t; \widehat{\theta}_h\right).$$

- Every input must be known at t (measurable with respect to $\mathcal{F}_t$, the information set at the forecast origin).

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

- At the end of month t, forecast SKU i's sales h months ahead:

$$\widehat{s}^{(i)}_{t+h|t} = f_h\left(\mathbf{x}^{(i)}_t; \widehat{\theta}_h\right).$$

- Every input must be known at t (measurable with respect to $\mathcal{F}_t$, the information set at the forecast origin).
- Under squared-error loss, the optimal population prediction from these inputs is the conditional mean:

$$f^*_h(\mathbf{x}^{(i)}_t) = E\left[s^{(i)}_{t+h} \mid \mathbf{x}^{(i)}_t\right].$$

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

- At the end of month t, forecast SKU i's sales h months ahead:

$$\widehat{s}^{(i)}_{t+h|t} = f_h\left(\mathbf{x}^{(i)}_t; \widehat{\theta}_h\right).$$

- Every input must be known at t (measurable with respect to $\mathcal{F}_t$, the information set at the forecast origin).
- Under squared-error loss, the optimal population prediction from these inputs is the conditional mean:

$$f^*_h(\mathbf{x}^{(i)}_t) = E\left[s^{(i)}_{t+h} \mid \mathbf{x}^{(i)}_t\right].$$

- The target depends on the *loss function*: absolute-error loss targets a conditional median; quantile loss targets a specified conditional quantile.

Formalizing the ML Forecast

Q: What function are we estimating at a forecast origin?

- At the end of month t, forecast SKU i's sales h months ahead:

$$\widehat{s}^{(i)}_{t+h|t} = f_h\left(\mathbf{x}^{(i)}_t; \widehat{\theta}_h\right).$$

- Every input must be known at t (measurable with respect to $\mathcal{F}_t$, the information set at the forecast origin).
- Under squared-error loss, the optimal population prediction from these inputs is the conditional mean:

$$f^*_h(\mathbf{x}^{(i)}_t) = E\left[s^{(i)}_{t+h} \mid \mathbf{x}^{(i)}_t\right].$$

- The target depends on the *loss function*: absolute-error loss targets a conditional median; quantile loss targets a specified conditional quantile.
- Unlike a linear ADL, f_h can capture nonlinearities and interactions without us specifying their exact functional forms.

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

- Each historical forecast origin contributes an input–target pair:

$$\left(\mathbf{x}_t^{(i)},\ s_{t+h}^{(i)}\right).$$

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

- Each historical forecast origin contributes an input–target pair:

$$\left(\mathbf{x}_t^{(i)},\ s_{t+h}^{(i)}\right).$$

- For $h = 1$, features assembled using information through December 2022 are paired with January 2023 sales. Shift the origin forward one month and repeat.

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

- Each historical forecast origin contributes an input–target pair:

$$\left(\mathbf{x}_t^{(i)},\ s_{t+h}^{(i)}\right).$$

- For $h = 1$, features assembled using information through December 2022 are paired with January 2023 sales. Shift the origin forward one month and repeat.
- For $h = 3$, those same December 2022 features are instead paired with March 2023 sales.

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

- Each historical forecast origin contributes an input–target pair:

$$\left(\mathbf{x}_t^{(i)},\ s_{t+h}^{(i)}\right).$$

- For $h = 1$, features assembled using information through December 2022 are paired with January 2023 sales. Shift the origin forward one month and repeat.
- For $h = 3$, those same December 2022 features are instead paired with March 2023 sales.
- Consecutive examples have overlapping feature windows; hence training rows need not be independent observations.

Turning a Time Series into Supervised Learning

Q: What counts as one training observation?

- Each historical forecast origin contributes an input–target pair:

$$\left(\mathbf{x}_t^{(i)},\ s_{t+h}^{(i)}\right).$$

- For $h = 1$, features assembled using information through December 2022 are paired with January 2023 sales. Shift the origin forward one month and repeat.
- For $h = 3$, those same December 2022 features are instead paired with March 2023 sales.
- Consecutive examples have overlapping feature windows; hence training rows need not be independent observations.
- Decide the target horizon and the feature-availability rule *before* building the supervised dataset.

Sliding Forecast Origins: An Illustration

Q: How do histories, features, and future targets line up?

Date	Y	X	YL1	YL2	XL1	XL2	XL3
Jan-00	10	~~100~~					
Feb-00	11	~~112~~	11		100		
Mar-00	12	~~115~~	12	10	112	100	
Apr-00	13	~~118~~	13	11	115	112	100
May-00	14	~~120~~	14	12	118	115	112
Jun-00	15	~~130~~	15	13	120	118	115

Sliding Forecast Origins: An Illustration

Q: How do histories, features, and future targets line up?

Date	Y	X	YL1	YL2	XL1	XL2	XL3
Jan-00	10	~~100~~					
Feb-00	11	~~112~~	11		100		
Mar-00	12	~~115~~	12	10	112	100	
Apr-00	13	~~118~~	13	11	115	112	100
May-00	14	~~120~~	14	12	118	115	112
Jun-00	15	~~130~~	15	13	120	118	115

- Every training feature window ends at its own forecast origin; its associated target occurs h months later.

Sliding Forecast Origins: An Illustration

Q: How do histories, features, and future targets line up?

Date	Y	X	YL1	YL2	XL1	XL2	XL3
Jan-00	10	~~100~~					
Feb-00	11	~~112~~	11		100		
Mar-00	12	~~115~~	12	10	112	100	
Apr-00	13	~~118~~	13	11	115	112	100
May-00	14	~~120~~	14	12	118	115	112
Jun-00	15	~~130~~	15	13	120	118	115

- Every training feature window ends at its own forecast origin; its associated target occurs h months later.
- Adjacent windows overlap. More rows do *not* necessarily mean an equivalent increase in independent information.

Regression Trees: From Tabular ML to Time Series

Q: How do we adapt a familiar regression tree to forecast sales?

Regression Trees: From Tabular ML to Time Series

Q: How do we adapt a familiar regression tree to forecast sales?

- **Recall a standard regression tree:**
  - Partition predictor space into terminal regions $R_1, \ldots, R_J$.
  - Predict the mean observed target for training observations in each leaf.

Regression Trees: From Tabular ML to Time Series

Q: How do we adapt a familiar regression tree to forecast sales?

- **Recall a standard regression tree:**
  - Partition predictor space into terminal regions $R_1, \ldots, R_J$.
  - Predict the mean observed target for training observations in each leaf.
- **Now consider our time-series application:**
  - Each training observation corresponds to a historical forecast origin.
  - The features include **lagged** sales and other information known at that origin.
  - The target is sales h months later.

Regression Trees: From Tabular ML to Time Series

Q: How do we adapt a familiar regression tree to forecast sales?

- **Recall a standard regression tree:**
  - Partition predictor space into terminal regions $R_1, \ldots, R_J$.
  - Predict the mean observed target for training observations in each leaf.
- **Now consider our time-series application:**
  - Each training observation corresponds to a historical forecast origin.
  - The features include **lagged** sales and other information known at that origin.
  - The target is sales h months later.
- **We apply the same tree algorithm** to these newly constructed input–target pairs.

Regression Trees: From Tabular ML to Time Series

Q: How do we adapt a familiar regression tree to forecast sales?

- **Recall a standard regression tree:**
  - Partition predictor space into terminal regions $R_1, \ldots, R_J$.
  - Predict the mean observed target for training observations in each leaf.
- **Now consider our time-series application:**
  - Each training observation corresponds to a historical forecast origin.
  - The features include **lagged** sales and other information known at that origin.
  - The target is sales h months later.
- **We apply the same tree algorithm** to these newly constructed input–target pairs.
- The tree can now split on lagged sales, recent growth, seasonality, and interactions among these features.

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

- For a new forecast origin t, construct $\mathbf{x}_t^{(i)}$ using the available sales history.

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

- For a new forecast origin t, construct $\mathbf{x}_t^{(i)}$ using the available sales history.
- Pass this feature vector through the fitted tree to identify its terminal leaf.

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

- For a new forecast origin t, construct $\mathbf{x}_t^{(i)}$ using the available sales history.
- Pass this feature vector through the fitted tree to identify its terminal leaf.
- **The forecast is the mean future sales of historical training observations in that leaf:**

$$\widehat{s}_{t+h|t}^{(i)} = \sum_{j=1}^{J} \overline{s}_{j,h} \, \mathbf{1}\{\mathbf{x}_t^{(i)} \in R_j\}.$$

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

- For a new forecast origin t, construct $\mathbf{x}_t^{(i)}$ using the available sales history.
- Pass this feature vector through the fitted tree to identify its terminal leaf.
- **The forecast is the mean future sales of historical training observations in that leaf:**

$$\widehat{s}_{t+h|t}^{(i)} = \sum_{j=1}^{J} \overline{s}_{j,h} \, \mathbf{1}\{\mathbf{x}_t^{(i)} \in R_j\}.$$

- **Example:** Suppose the selected leaf contains historical observations characterized by rising sales and a scheduled promotion.

  The average sales h months after those historical origins becomes our forecast.

Regression Trees: Generating a Forecast

Q: How does the fitted tree forecast future sales?

- For a new forecast origin t, construct $\mathbf{x}_t^{(i)}$ using the available sales history.
- Pass this feature vector through the fitted tree to identify its terminal leaf.
- **The forecast is the mean future sales of historical training observations in that leaf:**

$$\widehat{s}_{t+h|t}^{(i)} = \sum_{j=1}^{J} \overline{s}_{j,h} \, \mathbf{1}\{\mathbf{x}_t^{(i)} \in R_j\}.$$

- **Example:** Suppose the selected leaf contains historical observations characterized by rising sales and a scheduled promotion.

  The average sales h months after those historical origins becomes our forecast.

- **Key insight:** The tree algorithm is unchanged. By supplying historical features, we allow it to learn nonlinear relationships involving temporal persistence.

Bagging and Random Forests

Q: Doesn't bootstrap resampling violate temporal ordering?

Bagging and Random Forests

Q: Doesn't bootstrap resampling violate temporal ordering?

- **Recall bagging:** Fit trees on bootstrapped training rows and average their predictions:

$$\widehat{f}_h(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}_{h,b}(\mathbf{x}).$$

Bagging and Random Forests

Q: Doesn't bootstrap resampling violate temporal ordering?

- **Recall bagging:** Fit trees on bootstrapped training rows and average their predictions:

$$\widehat{f}_h(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}_{h,b}(\mathbf{x}).$$

- **Recall random forests:** Also randomize the candidate features considered at each split.

Bagging and Random Forests

Q: Doesn't bootstrap resampling violate temporal ordering?

- **Recall bagging:** Fit trees on bootstrapped training rows and average their predictions:

$$\widehat{f}_h(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}_{h,b}(\mathbf{x}).$$

- **Recall random forests:** Also randomize the candidate features considered at each split.

- **If we randomly resample these training rows, aren't we destroying their temporal ordering?**

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

A: Because each training row already contains the correctly ordered historical features and future target.

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

A: Because each training row already contains the correctly ordered historical features and future target.

Example: One possible bootstrap sample

Origin	s_{t-2}	s_{t-1}	s_t	**Target** s_{t+1}
June	95	105	110	120
April	90	100	95	105
June	95	105	110	120
May	100	95	105	110

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

A: Because each training row already contains the correctly ordered historical features and future target.

Example: One possible bootstrap sample

Origin	s_{t-2}	s_{t-1}	s_t	**Target** s_{t+1}
June	95	105	110	120
April	90	100	95	105
June	95	105	110	120
May	100	95	105	110

- The bootstrap changes which **complete training rows** are selected, including duplicates.

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

A: Because each training row already contains the correctly ordered historical features and future target.

Example: One possible bootstrap sample

Origin	s_{t-2}	s_{t-1}	s_t	**Target** s_{t+1}
June	95	105	110	120
April	90	100	95	105
June	95	105	110	120
May	100	95	105	110

- The bootstrap changes which **complete training rows** are selected, including duplicates.
- It does **not** change the ordering of lagged sales *within* any row.

Bootstrapping Time-Series Training Observations

Q: Why can we resample rows without destroying temporal information?

A: Because each training row already contains the correctly ordered historical features and future target.

Example: One possible bootstrap sample

Origin	s_{t-2}	s_{t-1}	s_t	**Target** s_{t+1}
June	95	105	110	120
April	90	100	95	105
June	95	105	110	120
May	100	95	105	110

- The bootstrap changes which **complete training rows** are selected, including duplicates.
- It does **not** change the ordering of lagged sales *within* any row.
- A regression tree partitions the feature space; it does not require training rows to arrive in date order.

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

- Recall: A tree's **out-of-bag (OOB)** observations are those not selected in its bootstrap training sample.

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

- Recall: A tree's **out-of-bag (OOB)** observations are those not selected in its bootstrap training sample.
- Suppose we build a forest using monthly sales observations from 2018–2023.

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

- Recall: A tree's **out-of-bag (OOB)** observations are those not selected in its bootstrap training sample.
- Suppose we build a forest using monthly sales observations from 2018–2023.
- Consider predicting July 2021 sales using features available at the end of June 2021.
- One tree's bootstrap sample might include:

Forecast origin	**Selected for tree?**	**After June 2021?**
March 2020	Yes	No
June 2021	No (OOB)	—
April 2022	Yes	Yes
September 2023	Yes	Yes

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

- Recall: A tree's **out-of-bag (OOB)** observations are those not selected in its bootstrap training sample.
- Suppose we build a forest using monthly sales observations from 2018–2023.
- Consider predicting July 2021 sales using features available at the end of June 2021.
- One tree's bootstrap sample might include:

Forecast origin	Selected for tree?	After June 2021?
March 2020	Yes	No
June 2021	No (OOB)	—
April 2022	Yes	Yes
September 2023	Yes	Yes

- **Problem:** Although June 2021 is OOB, the tree was trained using observations from 2022 and 2023.

Random Forests: Out-of-Bag Evaluation

Q: Does an out-of-bag observation constitute a genuine forecast?

- Recall: A tree's **out-of-bag (OOB)** observations are those not selected in its bootstrap training sample.
- Suppose we build a forest using monthly sales observations from 2018–2023.
- Consider predicting July 2021 sales using features available at the end of June 2021.
- One tree's bootstrap sample might include:

Forecast origin	**Selected for tree?**	**After June 2021?**
March 2020	Yes	No
June 2021	No (OOB)	—
April 2022	Yes	Yes
September 2023	Yes	Yes

- **Problem:** Although June 2021 is OOB, the tree was trained using observations from 2022 and 2023.
- Its OOB prediction for July 2021 sales therefore does **not** replicate a forecast we could have made in June 2021.

Random Forests: Chronological Forecast Evaluation

Q: How can we evaluate forecasts without using future information?

Random Forests: Chronological Forecast Evaluation

Q: How can we evaluate forecasts without using future information?

Suppose we have monthly sales from 2018–2024 and want to evaluate one-month-ahead forecasts for 2024.

Random Forests: Chronological Forecast Evaluation

Q: How can we evaluate forecasts without using future information?

Suppose we have monthly sales from 2018–2024 and want to evaluate one-month-ahead forecasts for 2024.

Separate training and evaluation chronologically:

Stage	**Forecast origin**	**Target month**
Training	Through Nov. 2023	Through Dec. 2023
Test 1	Dec. 2023	Jan. 2024
Test 2	Jan. 2024	Feb. 2024
⋮	⋮	⋮

Random Forests: Chronological Forecast Evaluation

Q: How can we evaluate forecasts without using future information?

Suppose we have monthly sales from 2018–2024 and want to evaluate one-month-ahead forecasts for 2024.

Separate training and evaluation chronologically:

Stage	**Forecast origin**	**Target month**
Training	Through Nov. 2023	Through Dec. 2023
Test 1	Dec. 2023	Jan. 2024
Test 2	Jan. 2024	Feb. 2024
⋮	⋮	⋮

- At each test origin, use only information available at that date. If the model is re-estimated, use only outcomes that have already been observed.

Random Forests: Chronological Forecast Evaluation

Q: How can we evaluate forecasts without using future information?

Suppose we have monthly sales from 2018–2024 and want to evaluate one-month-ahead forecasts for 2024.

Separate training and evaluation chronologically:

Stage	**Forecast origin**	**Target month**
Training	Through Nov. 2023	Through Dec. 2023
Test 1	Dec. 2023	Jan. 2024
Test 2	Jan. 2024	Feb. 2024
⋮	⋮	⋮

- At each test origin, use only information available at that date. If the model is re-estimated, use only outcomes that have already been observed.
- **Key distinction:**
  - **Bootstrap resampling:** Randomly selects rows within the eligible training sample to fit trees.
  - **Forecast evaluation:** Tests the fitted model on subsequent observations, preserving chronological order.

Gradient Boosting for Forecasting

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.
2. For squared-error loss, compute training residuals:

$$r_m^{(k)} = s_{t_m+h}^{(i_m)} - \widehat{f}_{k-1}(\mathbf{x}_{t_m}^{(i_m)}).$$

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.
2. For squared-error loss, compute training residuals:

$$r_m^{(k)} = s_{t_m+h}^{(i_m)} - \widehat{f}_{k-1}(\mathbf{x}_{t_m}^{(i_m)}).$$

3. Fit a small tree $g_k(\mathbf{x})$ to these residuals (or to a negative gradient for a different loss).

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.
2. For squared-error loss, compute training residuals:

$$r_m^{(k)} = s_{t_m+h}^{(i_m)} - \widehat{f}_{k-1}(\mathbf{x}_{t_m}^{(i_m)}).$$

3. Fit a small tree $g_k(\mathbf{x})$ to these residuals (or to a negative gradient for a different loss).
4. Update the forecast function:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}), \quad 0 < \eta \leq 1.$$

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.
2. For squared-error loss, compute training residuals:

$$r_m^{(k)} = s_{t_m+h}^{(i_m)} - \widehat{f}_{k-1}(\mathbf{x}_{t_m}^{(i_m)}).$$

3. Fit a small tree $g_k(\mathbf{x})$ to these residuals (or to a negative gradient for a different loss).
4. Update the forecast function:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}), \quad 0 < \eta \leq 1.$$

5. After K iterations: $\widehat{f}_K = \widehat{f}_0 + \eta \sum_{k=1}^{K} g_k$.

Gradient Boosting for Forecasting

1. Start with an initial forecast, e.g. $\widehat{f}_0(\mathbf{x}) = \overline{s}$.
2. For squared-error loss, compute training residuals:

$$r_m^{(k)} = s_{t_m+h}^{(i_m)} - \widehat{f}_{k-1}(\mathbf{x}_{t_m}^{(i_m)}).$$

3. Fit a small tree $g_k(\mathbf{x})$ to these residuals (or to a negative gradient for a different loss).
4. Update the forecast function:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}), \quad 0 < \eta \leq 1.$$

5. After K iterations: $\widehat{f}_K = \widehat{f}_0 + \eta \sum_{k=1}^{K} g_k$.

- Tune learning rate, tree complexity, and iteration count with chronological validation and early stopping (e.g., XGBoost).

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

- Recall: Gradient boosting fits a "sequence" of trees, each correcting errors from the preceding ensemble:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}).$$

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

- Recall: Gradient boosting fits a "sequence" of trees, each correcting errors from the preceding ensemble:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}).$$

- However, the "sequence" of fitted trees is **not the "sequence" of monthly sales observations**.

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

- Recall: Gradient boosting fits a "sequence" of trees, each correcting errors from the preceding ensemble:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}).$$

- However, the "sequence" of fitted trees is **not the "sequence" of monthly sales observations**.
- As with regression trees and random forests, boosting learns temporal relationships only if we supply relevant historical information:

$$\mathbf{x}_t = (s_t, s_{t-1}, s_{t-2}, \ldots).$$

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

- Recall: Gradient boosting fits a "sequence" of trees, each correcting errors from the preceding ensemble:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}).$$

- However, the "sequence" of fitted trees is **not the "sequence" of monthly sales observations**.
- As with regression trees and random forests, boosting learns temporal relationships only if we supply relevant historical information:

$$\mathbf{x}_t = (s_t, s_{t-1}, s_{t-2}, \ldots).$$

- **Example:** Boosting might learn that a promotion has a greater effect when sales have been declining for several consecutive months.

Gradient Boosting and Temporal Dependence

Q: Does "sequentially" fitting trees automatically capture persistence?

- Recall: Gradient boosting fits a "sequence" of trees, each correcting errors from the preceding ensemble:

$$\widehat{f}_k(\mathbf{x}) = \widehat{f}_{k-1}(\mathbf{x}) + \eta g_k(\mathbf{x}).$$

- However, the "sequence" of fitted trees is **not the "sequence" of monthly sales observations**.
- As with regression trees and random forests, boosting learns temporal relationships only if we supply relevant historical information:

$$\mathbf{x}_t = (s_t, s_{t-1}, s_{t-2}, \ldots).$$

- **Example:** Boosting might learn that a promotion has a greater effect when sales have been declining for several consecutive months.
- **Key insight:** Boosting changes how we *estimate* the prediction function, not how temporal information enters the model.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

- **Recall:** Neural networks learn nonlinear relationships through interconnected layers of trainable transformations.

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

- **Recall:** Neural networks learn nonlinear relationships through interconnected layers of trainable transformations.
- **Deep learning:** A family of ML methods centered on **deep neural networks (DNNs)**, which learn representations through multiple hidden layers.

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

- **Recall:** Neural networks learn nonlinear relationships through interconnected layers of trainable transformations.
- **Deep learning:** A family of ML methods centered on **deep neural networks (DNNs)**, which learn representations through multiple hidden layers.
    - A standard **feedforward neural network** can already be *deep*: its layers successively transform the inputs.

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

- **Recall:** Neural networks learn nonlinear relationships through interconnected layers of trainable transformations.
- **Deep learning:** A family of ML methods centered on **deep neural networks (DNNs)**, which learn representations through multiple hidden layers.
  - A standard **feedforward neural network** can already be *deep*: its layers successively transform the inputs.
  - **Architecture** is a separate design choice: how connections and computations are organized determines how information flows and interacts within a network.

Deep Learning for Time-Series Forecasting

Q: What is deep learning, and why does architecture matter for forecasting?

- **Recall:** Neural networks learn nonlinear relationships through interconnected layers of trainable transformations.
- **Deep learning:** A family of ML methods centered on **deep neural networks (DNNs)**, which learn representations through multiple hidden layers.
  - A standard **feedforward neural network** can already be *deep*: its layers successively transform the inputs.
  - **Architecture** is a separate design choice: how connections and computations are organized determines how information flows and interacts within a network.
- **For forecasting:** We will investigate different architectures for representing and processing historical observations. Depth alone does not determine how a network handles time.

Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Construct a fixed input vector and map it to future sales:

$$\mathbf{x}_t = (s_t, s_{t-1}, \ldots, s_{t-35}, \mathbf{z}_t), \qquad \widehat{\mathbf{s}}_{t+1:t+12|t} = f_\theta(\mathbf{x}_t).$$

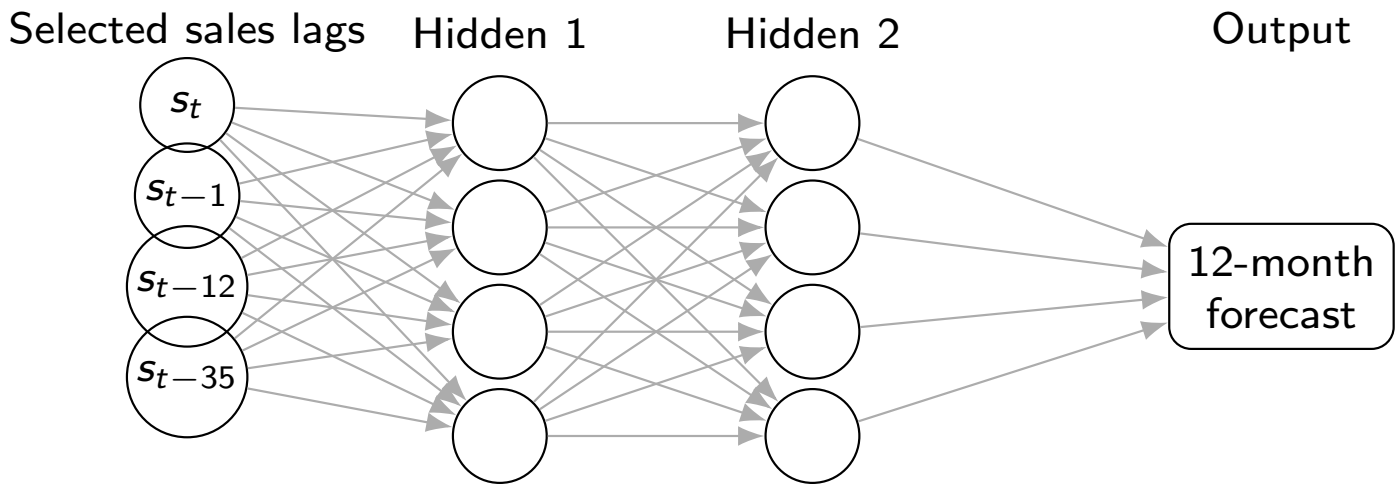
Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Construct a fixed input vector and map it to future sales:

$$\mathbf{x}_t = (s_t, s_{t-1}, \ldots, s_{t-35}, \mathbf{z}_t), \qquad \widehat{\mathbf{s}}_{t+1:t+12|t} = f_\theta(\mathbf{x}_t).$$



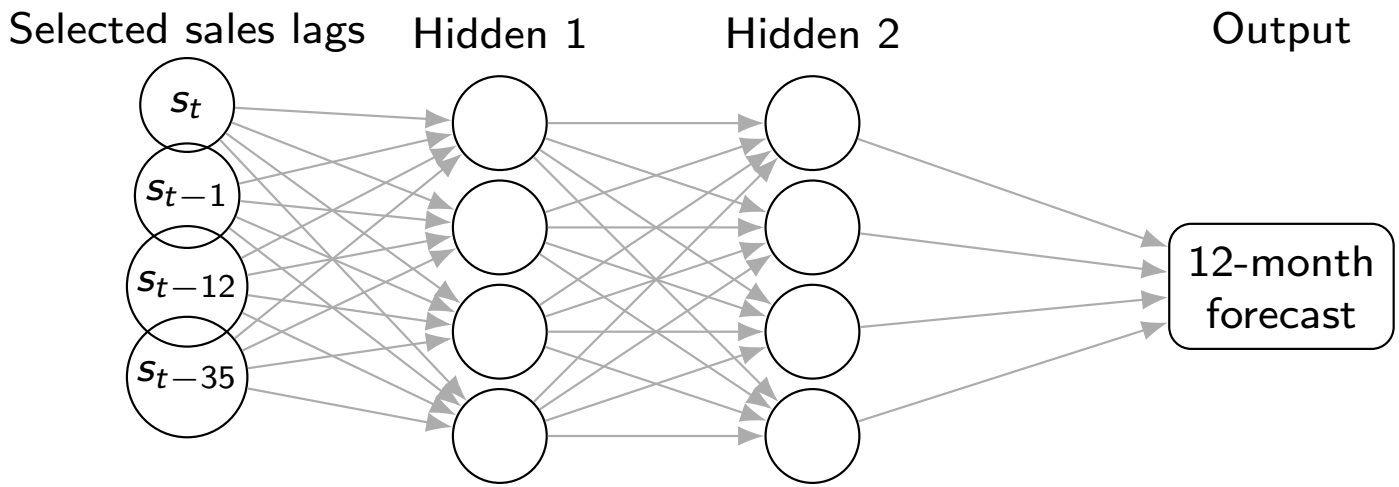
Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Construct a fixed input vector and map it to future sales:

$$\mathbf{x}_t = (s_t, s_{t-1}, \ldots, s_{t-35}, \mathbf{z}_t), \qquad \widehat{\mathbf{s}}_{t+1:t+12|t} = f_\theta(\mathbf{x}_t).$$



- Historical lags occupy **fixed input positions**; the network can learn nonlinear lag effects and interactions.

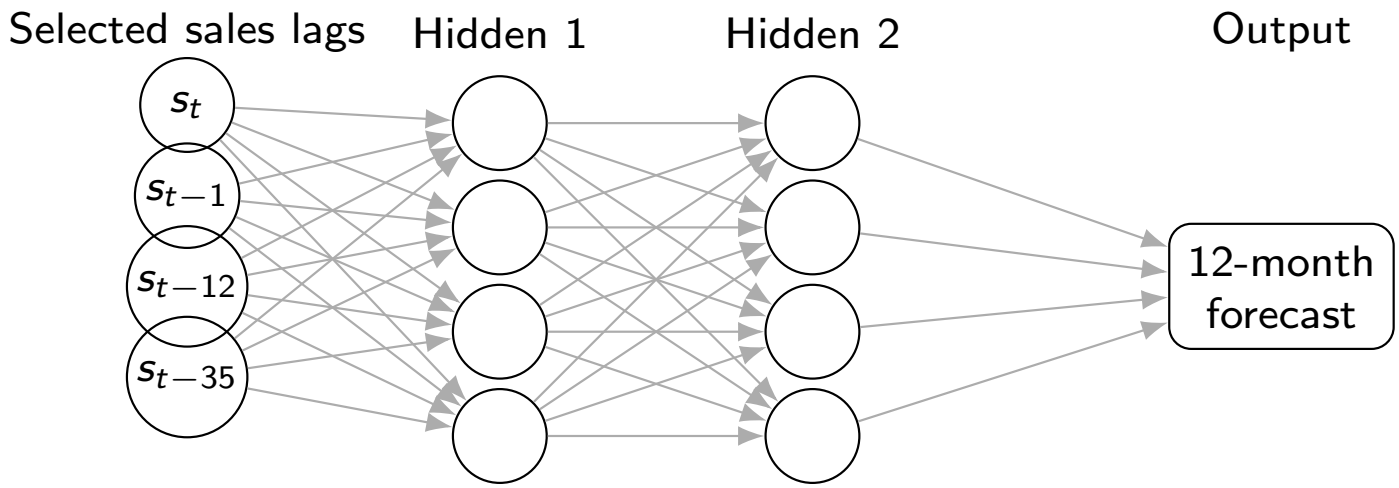
Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Construct a fixed input vector and map it to future sales:

$$\mathbf{x}_t = (s_t, s_{t-1}, \ldots, s_{t-35}, \mathbf{z}_t), \qquad \widehat{\mathbf{s}}_{t+1:t+12|t} = f_\theta(\mathbf{x}_t).$$



- Historical lags occupy **fixed input positions**; the network can learn nonlinear lag effects and interactions.
- The **36-month window** is one forecasting input, not our entire estimation sample. We train on many historical input–target windows, possibly pooled across SKUs.

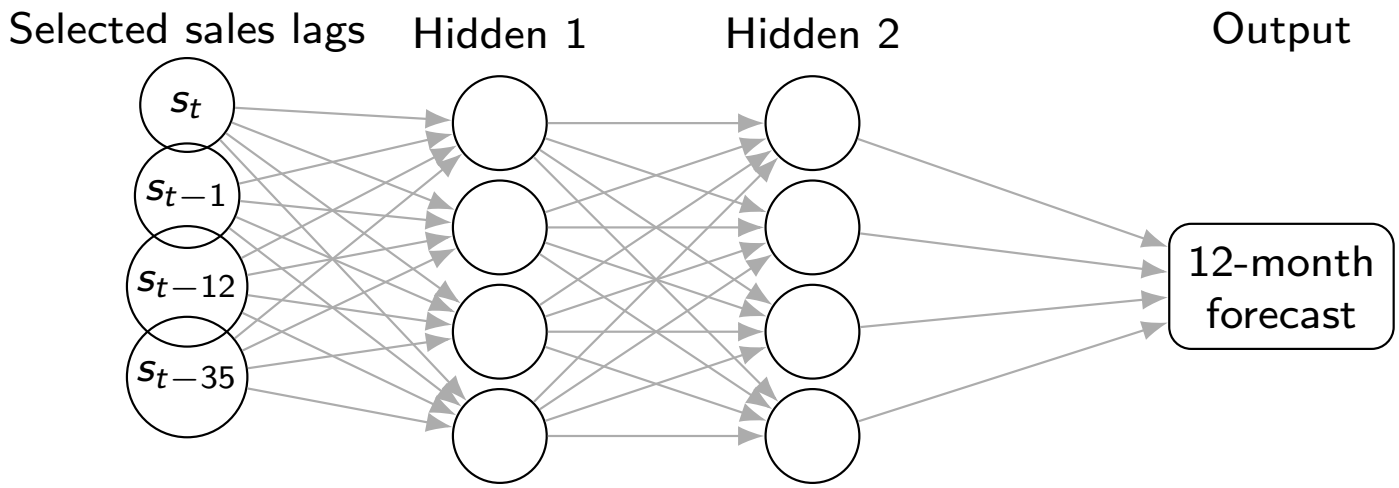
Feedforward Neural Networks for Forecasting

Q: How does a familiar neural network process historical sales?

Suppose we use **36 months of sales** for a SKU to forecast the **next 12 months**.

Construct a fixed input vector and map it to future sales:

$$\mathbf{x}_t = (s_t, s_{t-1}, \ldots, s_{t-35}, \mathbf{z}_t), \qquad \widehat{\mathbf{s}}_{t+1:t+12|t} = f_\theta(\mathbf{x}_t).$$



- Historical lags occupy **fixed input positions**; the network can learn nonlinear lag effects and interactions.
- The **36-month window** is one forecasting input, not our entire estimation sample. We train on many historical input–target windows, possibly pooled across SKUs.
- **Limitation:** No hidden state is carried from one month to the next. History outside the supplied window is unavailable unless included in additional features.

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.
- But we can introduce **forecasting-specific structure** into how the network processes that window.

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.
- But we can introduce **forecasting-specific structure** into how the network processes that window.
- Consider two alternatives:

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.
- But we can introduce **forecasting-specific structure** into how the network processes that window.
- Consider two alternatives:
  - **N-BEATS:** Organizes the network into successive blocks. Each block explains part of the historical window and contributes to the forecast.

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.
- But we can introduce **forecasting-specific structure** into how the network processes that window.
- Consider two alternatives:
  - **N-BEATS:** Organizes the network into successive blocks. Each block explains part of the historical window and contributes to the forecast.
  - **N-HiTS:** Processes historical information at multiple resolutions to represent patterns operating at different timescales.

Specialized Feedforward Forecasting Networks

Q: Can we design a feedforward architecture specifically for forecasting?

- Our standard feedforward network learns a nonlinear mapping from a historical window to future sales.
- But we can introduce **forecasting-specific structure** into how the network processes that window.
- Consider two alternatives:
  - **N-BEATS:** Organizes the network into successive blocks. Each block explains part of the historical window and contributes to the forecast.
  - **N-HiTS:** Processes historical information at multiple resolutions to represent patterns operating at different timescales.
- Both remain feedforward architectures: neither requires recurrent memory or transformer attention.

N-BEATS: Inside One Block

Q: Is an N-BEATS block a layer, or an entire neural network?

N-BEATS: Inside One Block

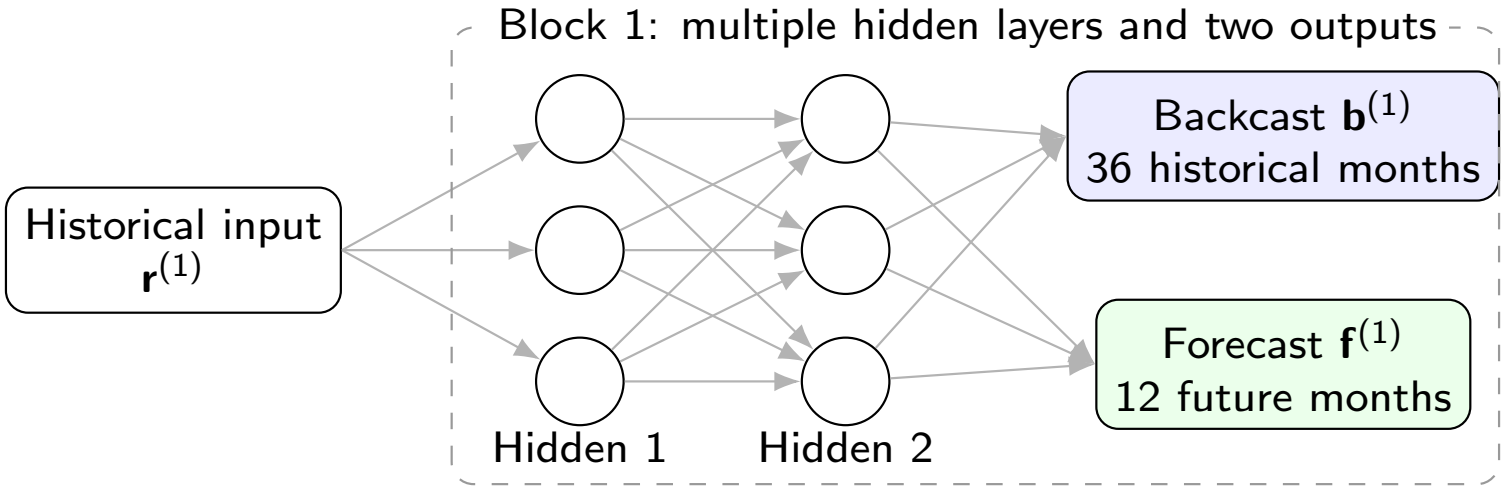
Q: Is an N-BEATS block a layer, or an entire neural network?

A block is a small feedforward neural network containing multiple hidden layers. It takes historical input and produces *two different outputs*.

N-BEATS: Inside One Block

Q: Is an N-BEATS block a layer, or an entire neural network?

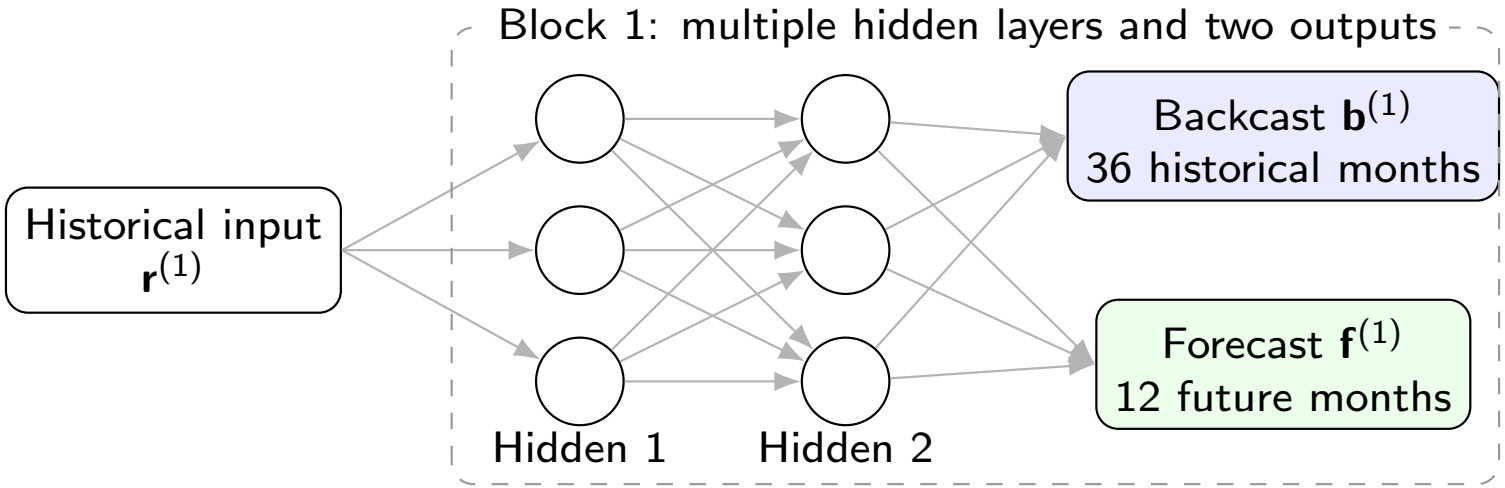
A block is a small feedforward neural network containing multiple hidden layers. It takes historical input and produces *two different outputs*.



N-BEATS: Inside One Block

Q: Is an N-BEATS block a layer, or an entire neural network?

A block is a small feedforward neural network containing multiple hidden layers. It takes historical input and produces *two different outputs*.



- **Backcast:** the block's approximation of its *historical input*.
- **Forecast:** the block's contribution to *future sales*.

N-BEATS: Two Outputs, Two Different Jobs

Q: What do we do with a block's backcast and forecast?

N-BEATS: Two Outputs, Two Different Jobs

Q: What do we do with a block's backcast and forecast?

1. Backcast: subtract it from the block's historical input. The remainder becomes the input to the next block:

$$\boxed{\mathbf{r}^{(2)} = \mathbf{r}^{(1)} - \mathbf{b}^{(1)}}$$

N-BEATS: Two Outputs, Two Different Jobs

Q: What do we do with a block's backcast and forecast?

1. Backcast: subtract it from the block's historical input. The remainder becomes the input to the next block:

$$\boxed{\mathbf{r}^{(2)} = \mathbf{r}^{(1)} - \mathbf{b}^{(1)}}$$

Toy example (3 observed months):

$$\underbrace{(90, 100, 110)}_{\mathbf{r}^{(1)}} - \underbrace{(85, 95, 102)}_{\mathbf{b}^{(1)}} = \underbrace{(5, 5, 8)}_{\mathbf{r}^{(2)}}$$

N-BEATS: Two Outputs, Two Different Jobs

Q: What do we do with a block's backcast and forecast?

1. Backcast: subtract it from the block's historical input. The remainder becomes the input to the next block:

$$\boxed{\mathbf{r}^{(2)} = \mathbf{r}^{(1)} - \mathbf{b}^{(1)}}$$

Toy example (3 observed months):

$$\underbrace{(90, 100, 110)}_{\mathbf{r}^{(1)}} - \underbrace{(85, 95, 102)}_{\mathbf{b}^{(1)}} = \underbrace{(5, 5, 8)}_{\mathbf{r}^{(2)}}$$

2. Forecast: add it to the other blocks' forecast contributions:

$$\boxed{\widehat{\mathbf{s}}_{t+1:t+H|t} = \mathbf{f}^{(1)} + \mathbf{f}^{(2)}}$$

N-BEATS: Two Outputs, Two Different Jobs

Q: What do we do with a block's backcast and forecast?

1. Backcast: subtract it from the block's historical input. The remainder becomes the input to the next block:

$$\boxed{\mathbf{r}^{(2)} = \mathbf{r}^{(1)} - \mathbf{b}^{(1)}}$$

Toy example (3 observed months):

$$\underbrace{(90, 100, 110)}_{\mathbf{r}^{(1)}} - \underbrace{(85, 95, 102)}_{\mathbf{b}^{(1)}} = \underbrace{(5, 5, 8)}_{\mathbf{r}^{(2)}}$$

2. Forecast: add it to the other blocks' forecast contributions:

$$\boxed{\widehat{\mathbf{s}}_{t+1:t+H|t} = \mathbf{f}^{(1)} + \mathbf{f}^{(2)}}$$

Toy example (2 future months):

$$\underbrace{(115, 120)}_{\mathbf{f}^{(1)}} + \underbrace{(3, -2)}_{\mathbf{f}^{(2)}} = \underbrace{(118, 118)}_{\text{final forecast}}$$

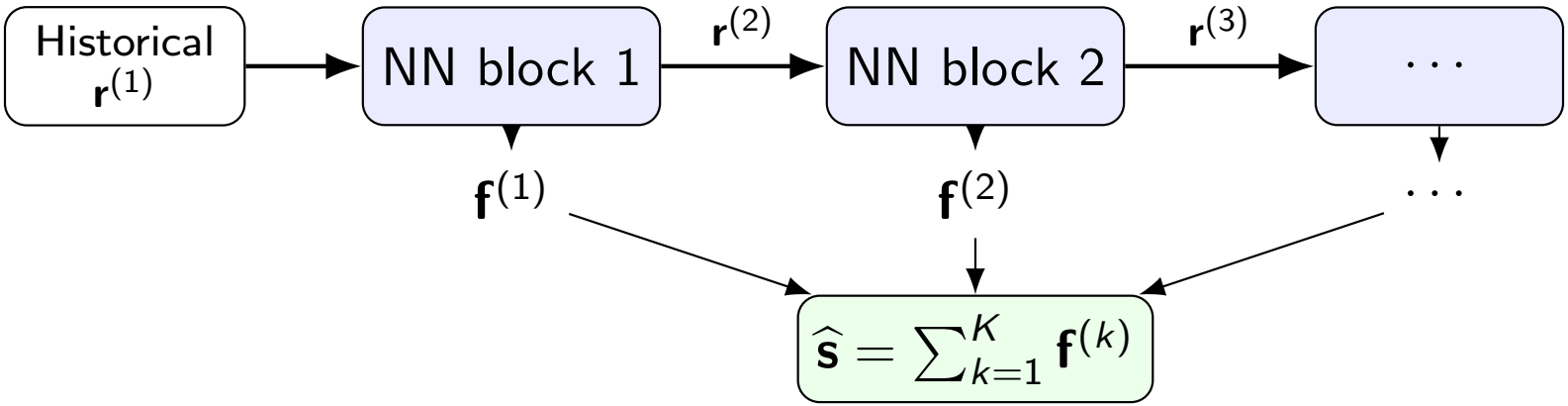
N-BEATS: Connecting the Blocks

Q: Is this just gradient boosting with neural networks?

N-BEATS: Connecting the Blocks

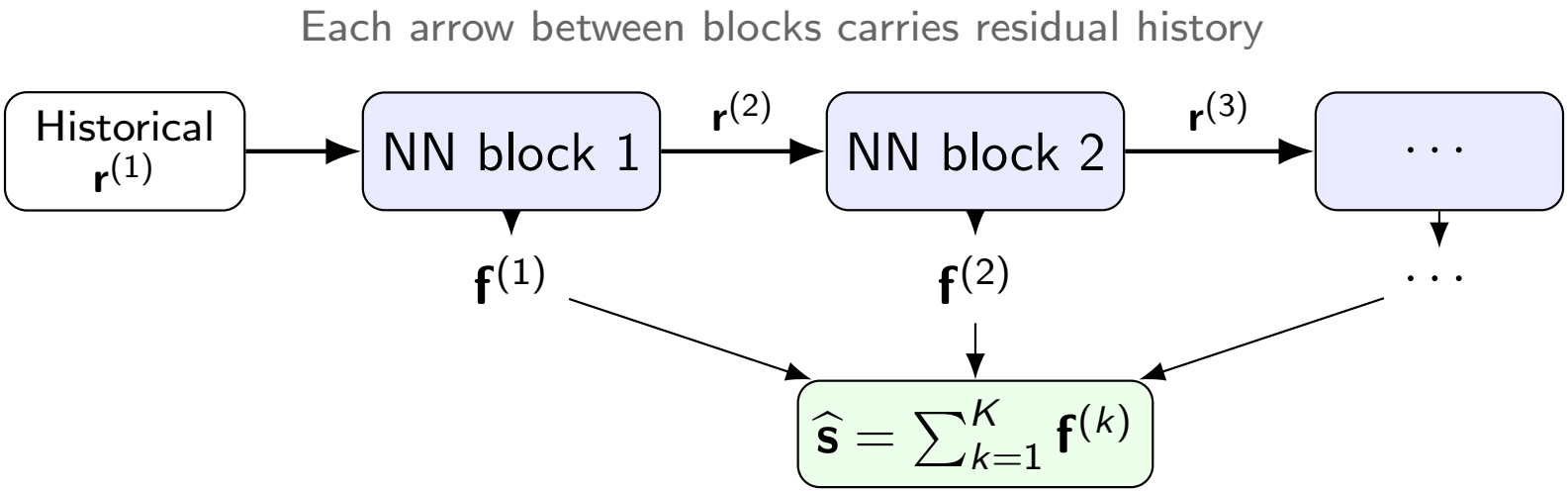
Q: Is this just gradient boosting with neural networks?

Each arrow between blocks carries residual history



N-BEATS: Connecting the Blocks

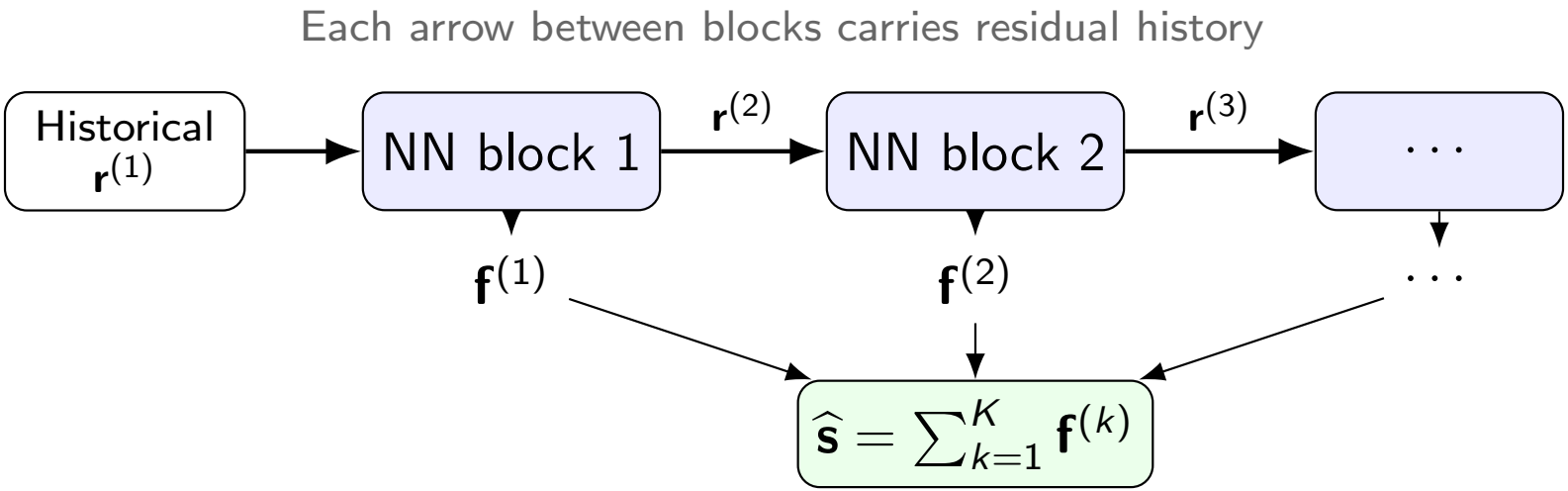
Q: Is this just gradient boosting with neural networks?



- **Like boosting:** contributions from successive components are added to form the final prediction.

N-BEATS: Connecting the Blocks

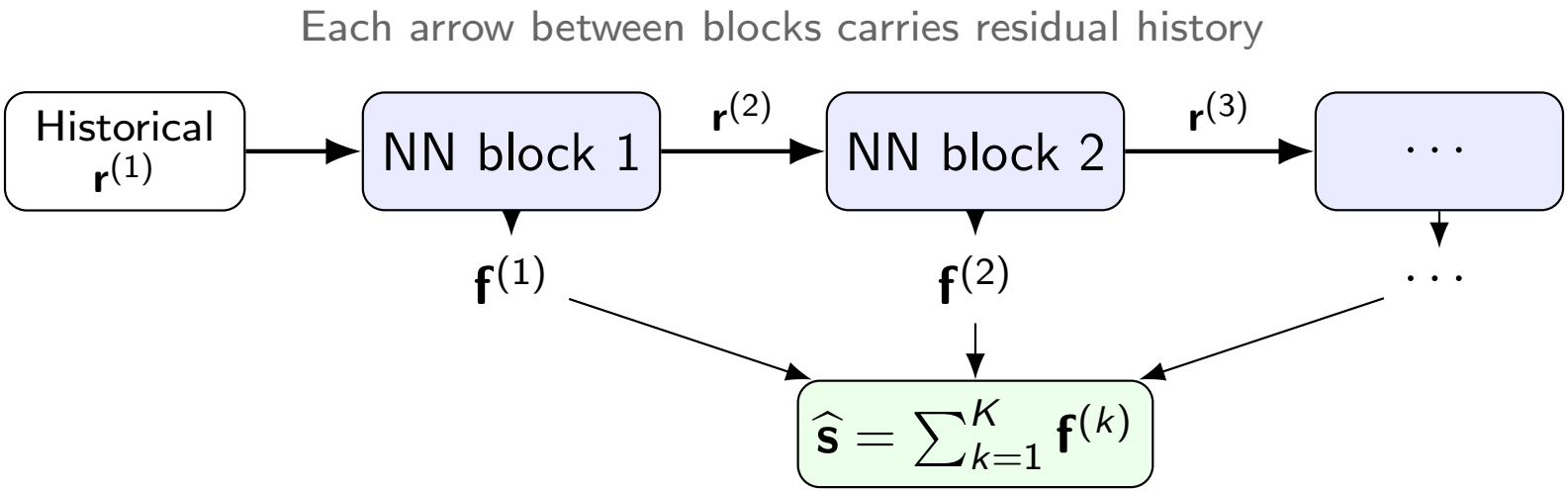
Q: Is this just gradient boosting with neural networks?



- **Like boosting:** contributions from successive components are added to form the final prediction.
- **Unlike boosting:** N-BEATS subtracts a *backcast from historical input*, not a prediction from an observed future target.

N-BEATS: Connecting the Blocks

Q: Is this just gradient boosting with neural networks?



- **Like boosting:** contributions from successive components are added to form the final prediction.
- **Unlike boosting:** N-BEATS subtracts a *backcast from historical input*, not a prediction from an observed future target.
- **Training distinction:** N-BEATS blocks are optimized *jointly* by backpropagation on the final forecasting loss; boosting typically fits its trees sequentially to target errors.

N-HiTS: Multiple Timescales

Q: Why might coarse and fine views of sales history both help?

N-HiTS: Multiple Timescales

Q: Why might coarse and fine views of sales history both help?

- Consider forecasting a SKU's next 12 months: the broad sales level may evolve slowly, while promotions create short-lived fluctuations.

N-HiTS: Multiple Timescales

Q: Why might coarse and fine views of sales history both help?

- Consider forecasting a SKU's next 12 months: the broad sales level may evolve slowly, while promotions create short-lived fluctuations.
- **N-HiTS** retains the feedforward/backcast–forecast idea but adds:
  - **Multi-rate sampling:** blocks analyze historical inputs at different temporal resolutions.
  - **Hierarchical interpolation:** blocks predict future coefficients on grids of different resolutions and interpolate to the target horizon.

N-HiTS: Multiple Timescales

Q: Why might coarse and fine views of sales history both help?

- Consider forecasting a SKU's next 12 months: the broad sales level may evolve slowly, while promotions create short-lived fluctuations.
- **N-HiTS** retains the feedforward/backcast–forecast idea but adds:
  - **Multi-rate sampling:** blocks analyze historical inputs at different temporal resolutions.
  - **Hierarchical interpolation:** blocks predict future coefficients on grids of different resolutions and interpolate to the target horizon.
- The design can model broad patterns with coarse representations and finer changes with more detailed ones.

N-HiTS: Multiple Timescales

Q: Why might coarse and fine views of sales history both help?

- Consider forecasting a SKU's next 12 months: the broad sales level may evolve slowly, while promotions create short-lived fluctuations.
- **N-HiTS** retains the feedforward/backcast–forecast idea but adds:
  - **Multi-rate sampling:** blocks analyze historical inputs at different temporal resolutions.
  - **Hierarchical interpolation:** blocks predict future coefficients on grids of different resolutions and interpolate to the target horizon.
- The design can model broad patterns with coarse representations and finer changes with more detailed ones.
- **Distinction:** N-BEATS emphasizes blockwise residual decomposition; N-HiTS also organizes information across resolutions. Both are feedforward architectures.

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

- **Feedforward NN – a snapshot:** At each forecast origin, provide the network with a vector of historical sales, such as the preceding 36 months. The network processes this vector without carrying a learned state from the previous month.

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

- **Feedforward NN – a snapshot:** At each forecast origin, provide the network with a vector of historical sales, such as the preceding 36 months. The network processes this vector without carrying a learned state from the previous month.
- **RNN – a running summary:** Instead, process observations sequentially, updating an internal representation as each new month arrives.

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

- **Feedforward NN – a snapshot:** At each forecast origin, provide the network with a vector of historical sales, such as the preceding 36 months. The network processes this vector without carrying a learned state from the previous month.
- **RNN – a running summary:** Instead, process observations sequentially, updating an internal representation as each new month arrives.
- Let $\mathbf{u}_t = (s_t, \mathbf{z}_t)$ contain the observations for month t. A basic RNN computes

$$\boxed{\mathbf{h}_t = \tanh(W_x \mathbf{u}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}).}$$

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

- **Feedforward NN – a snapshot:** At each forecast origin, provide the network with a vector of historical sales, such as the preceding 36 months. The network processes this vector without carrying a learned state from the previous month.
- **RNN – a running summary:** Instead, process observations sequentially, updating an internal representation as each new month arrives.
- Let $\mathbf{u}_t = (s_t, \mathbf{z}_t)$ contain the observations for month t. A basic RNN computes

$$\boxed{\mathbf{h}_t = \tanh(W_x \mathbf{u}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}).}$$

- **The crucial difference:**
  - In a feedforward NN, hidden activations depend on the *preceding layer*.
  - In an RNN, the hidden state also depends on the *preceding month's hidden state*, $\mathbf{h}_{t-1}$.

From Feedforward to Recurrent Networks

Q: What changes when a neural network maintains a memory of the past?

- **Feedforward NN – a snapshot:** At each forecast origin, provide the network with a vector of historical sales, such as the preceding 36 months. The network processes this vector without carrying a learned state from the previous month.
- **RNN – a running summary:** Instead, process observations sequentially, updating an internal representation as each new month arrives.
- Let $\mathbf{u}_t = (s_t, \mathbf{z}_t)$ contain the observations for month t. A basic RNN computes

$$\boxed{\mathbf{h}_t = \tanh(W_x \mathbf{u}_t + W_h \mathbf{h}_{t-1} + \mathbf{b}).}$$

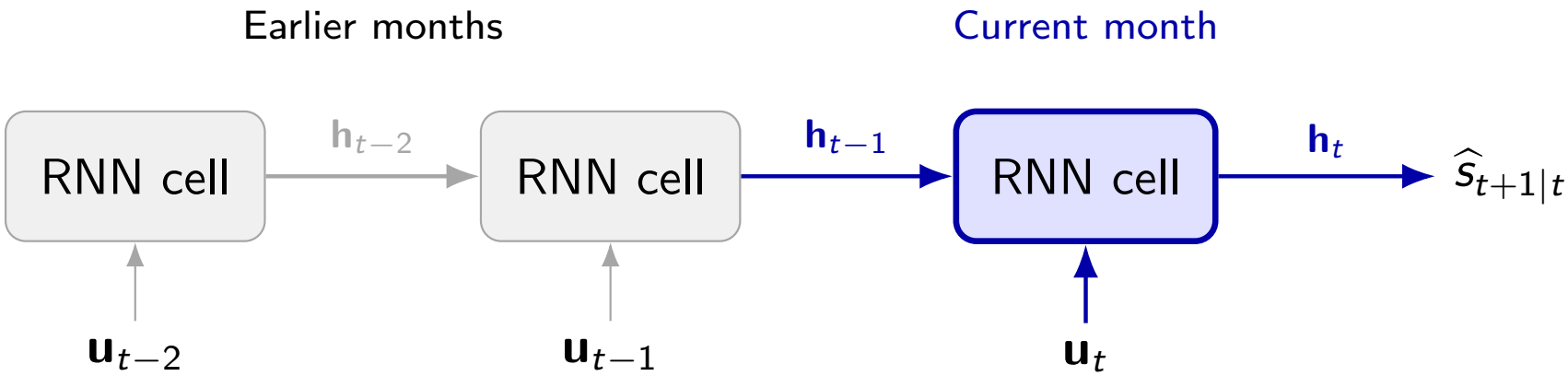
- **The crucial difference:**
  - In a feedforward NN, hidden activations depend on the *preceding layer*.
  - In an RNN, the hidden state also depends on the *preceding month's hidden state*, $\mathbf{h}_{t-1}$.
- The same state-update rule is reused each month. The network can learn what historical information to carry forward, although its effective memory is limited.

Recurrent Neural Networks: Unrolling Over Time

Q: How does a trained RNN use successive observations to forecast next month's sales?

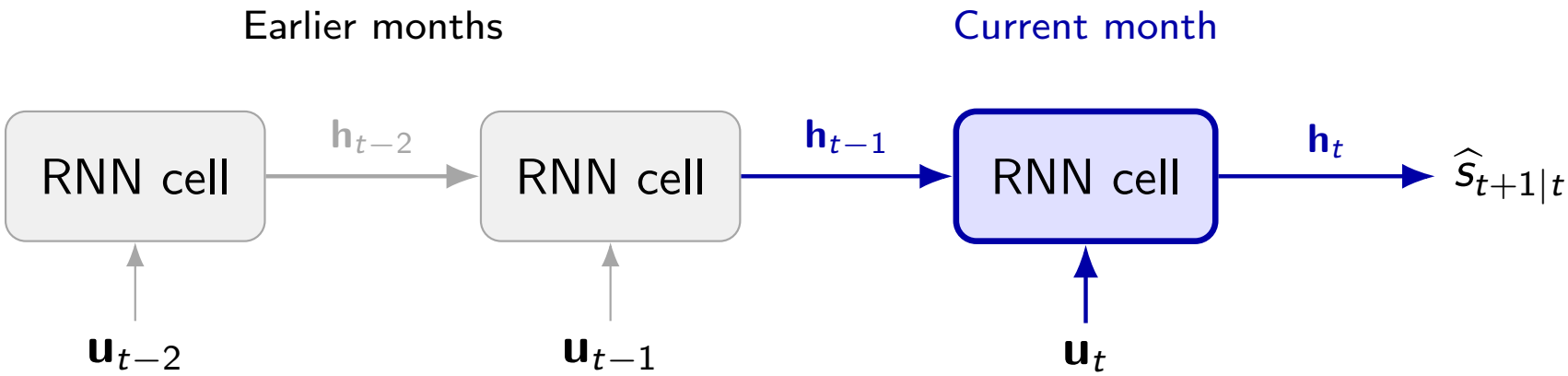
Recurrent Neural Networks: Unrolling Over Time

Q: How does a trained RNN use successive observations to forecast next month's sales?



Recurrent Neural Networks: Unrolling Over Time

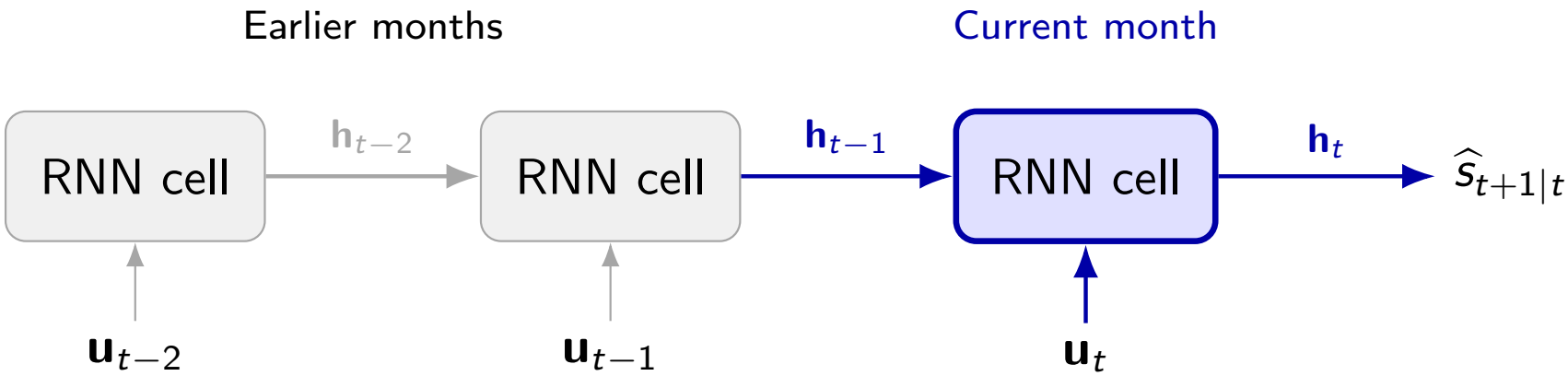
Q: How does a trained RNN use successive observations to forecast next month's sales?



- **Updating the hidden state:** At month t, the network combines the new observation $\mathbf{u}_t$ with the previous hidden state $\mathbf{h}_{t-1}$. The resulting $\mathbf{h}_t$ summarizes the history processed so far.

Recurrent Neural Networks: Unrolling Over Time

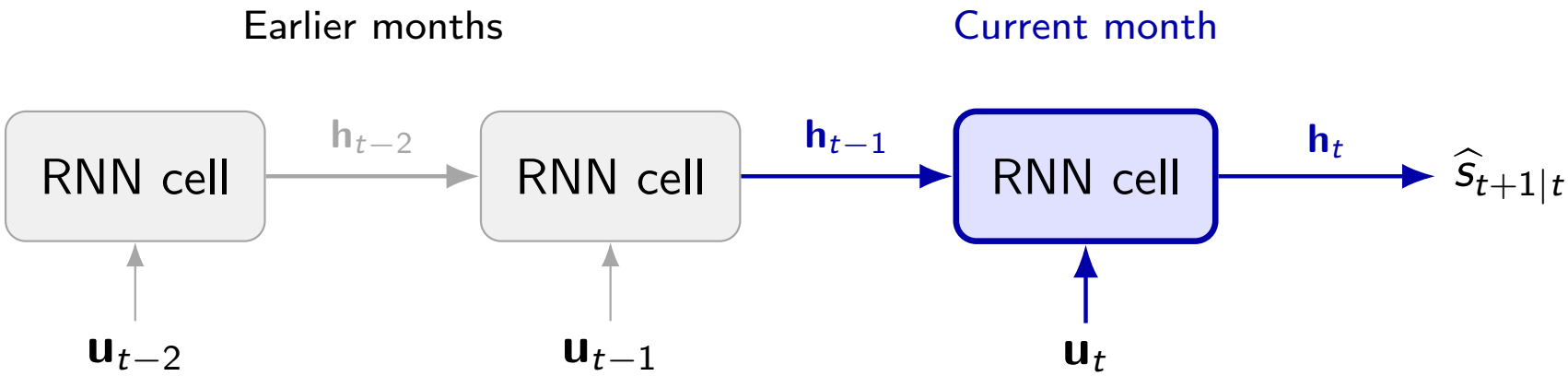
Q: How does a trained RNN use successive observations to forecast next month's sales?



- **Updating the hidden state:** At month t, the network combines the new observation $\mathbf{u}_t$ with the previous hidden state $\mathbf{h}_{t-1}$. The resulting $\mathbf{h}_t$ summarizes the history processed so far.
- **Forecasting with a trained RNN:** The weights are fixed and reused at every month. The diagram shows successive computations using the same trained network, not successive rounds of estimation. The final state generates the forecast.

Recurrent Neural Networks: Unrolling Over Time

Q: How does a trained RNN use successive observations to forecast next month's sales?



- **Updating the hidden state:** At month t, the network combines the new observation $\mathbf{u}_t$ with the previous hidden state $\mathbf{h}_{t-1}$. The resulting $\mathbf{h}_t$ summarizes the history processed so far.
- **Forecasting with a trained RNN:** The weights are fixed and reused at every month. The diagram shows successive computations using the same trained network, not successive rounds of estimation. The final state generates the forecast.
- **Contrast with feedforward NN:** A standard feedforward network processes a supplied historical window without passing a hidden state from one month to the next.

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

- Suppose last December's sales contain a seasonal signal relevant to the forecast we make this November.

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

- Suppose last December's sales contain a seasonal signal relevant to the forecast we make this November.
- In an RNN, that information influences November's state only through **successive updates** during the intervening months.

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

- Suppose last December's sales contain a seasonal signal relevant to the forecast we make this November.
- In an RNN, that information influences November's state only through **successive updates** during the intervening months.
- During training, gradients likewise propagate through successive state transitions:

$$\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-k}} = \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{h}_{t-2}} \cdots \frac{\partial \mathbf{h}_{t-k+1}}{\partial \mathbf{h}_{t-k}}.$$

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

- Suppose last December's sales contain a seasonal signal relevant to the forecast we make this November.
- In an RNN, that information influences November's state only through **successive updates** during the intervening months.
- During training, gradients likewise propagate through successive state transitions:

$$\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-k}} = \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{h}_{t-2}} \cdots \frac{\partial \mathbf{h}_{t-k+1}}{\partial \mathbf{h}_{t-k}}.$$

- Repeated derivatives can cause gradients to **vanish** or **explode**, making distant dependencies difficult to learn.

The Limitation of a Basic RNN

Q: Why might a useful observation from last year be forgotten?

- Suppose last December's sales contain a seasonal signal relevant to the forecast we make this November.
- In an RNN, that information influences November's state only through **successive updates** during the intervening months.
- During training, gradients likewise propagate through successive state transitions:

$$\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-k}} = \frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}} \frac{\partial \mathbf{h}_{t-1}}{\partial \mathbf{h}_{t-2}} \cdots \frac{\partial \mathbf{h}_{t-k+1}}{\partial \mathbf{h}_{t-k}}.$$

- Repeated derivatives can cause gradients to **vanish** or **explode**, making distant dependencies difficult to learn.
- **Motivation for LSTM:** Provide a more controlled memory pathway so useful information can be retained, revised, or forgotten.

LSTM: Selective Memory

Q: How can a recurrent network decide what to remember?

LSTM: Selective Memory

Q: How can a recurrent network decide what to remember?

- An LSTM maintains two distinct states:
  - $\mathbf{c}_t$: **cell state**, a pathway for retaining information over time.
  - $\mathbf{h}_t$: **hidden state**, passed to the next time step and used to generate forecasts.

LSTM: Selective Memory

Q: How can a recurrent network decide what to remember?

- An LSTM maintains two distinct states:
  - $\mathbf{c}_t$: **cell state**, a pathway for retaining information over time.
  - $\mathbf{h}_t$: **hidden state**, passed to the next time step and used to generate forecasts.
- Three learned gates control the information flow:
  - **Forget gate $\mathbf{f}_t$**: how much prior cell memory to retain.
  - **Input gate $\mathbf{i}_t$**: how much candidate memory to add.
  - **Output gate $\mathbf{o}_t$**: how much updated memory to expose.

LSTM: Selective Memory

Q: How can a recurrent network decide what to remember?

- An LSTM maintains two distinct states:
  - $\mathbf{c}_t$: **cell state**, a pathway for retaining information over time.
  - $\mathbf{h}_t$: **hidden state**, passed to the next time step and used to generate forecasts.
- Three learned gates control the information flow:
  - **Forget gate $\mathbf{f}_t$**: how much prior cell memory to retain.
  - **Input gate $\mathbf{i}_t$**: how much candidate memory to add.
  - **Output gate $\mathbf{o}_t$**: how much updated memory to expose.
- **Sales example:** The model could learn to retain a recurring December demand pattern while discounting an isolated one-month sales spike.

LSTM: Selective Memory

Q: How can a recurrent network decide what to remember?

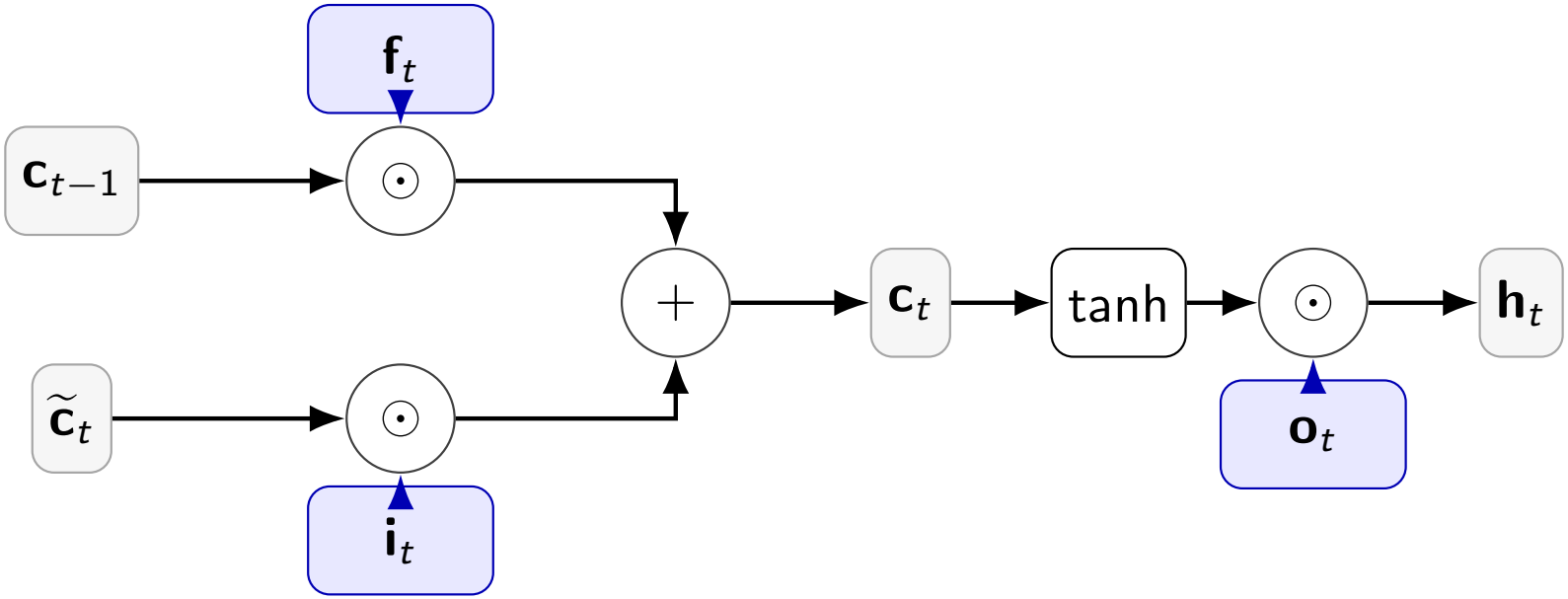
- An LSTM maintains two distinct states:
  - $\mathbf{c}_t$: **cell state**, a pathway for retaining information over time.
  - $\mathbf{h}_t$: **hidden state**, passed to the next time step and used to generate forecasts.
- Three learned gates control the information flow:
  - **Forget gate $\mathbf{f}_t$**: how much prior cell memory to retain.
  - **Input gate $\mathbf{i}_t$**: how much candidate memory to add.
  - **Output gate $\mathbf{o}_t$**: how much updated memory to expose.
- **Sales example:** The model could learn to retain a recurring December demand pattern while discounting an isolated one-month sales spike.
- Gates *facilitate* long-range learning; they do not guarantee the network will discover the correct pattern.

LSTM: Inside One Time Step

Q: Where do the forget, input, and output gates act?

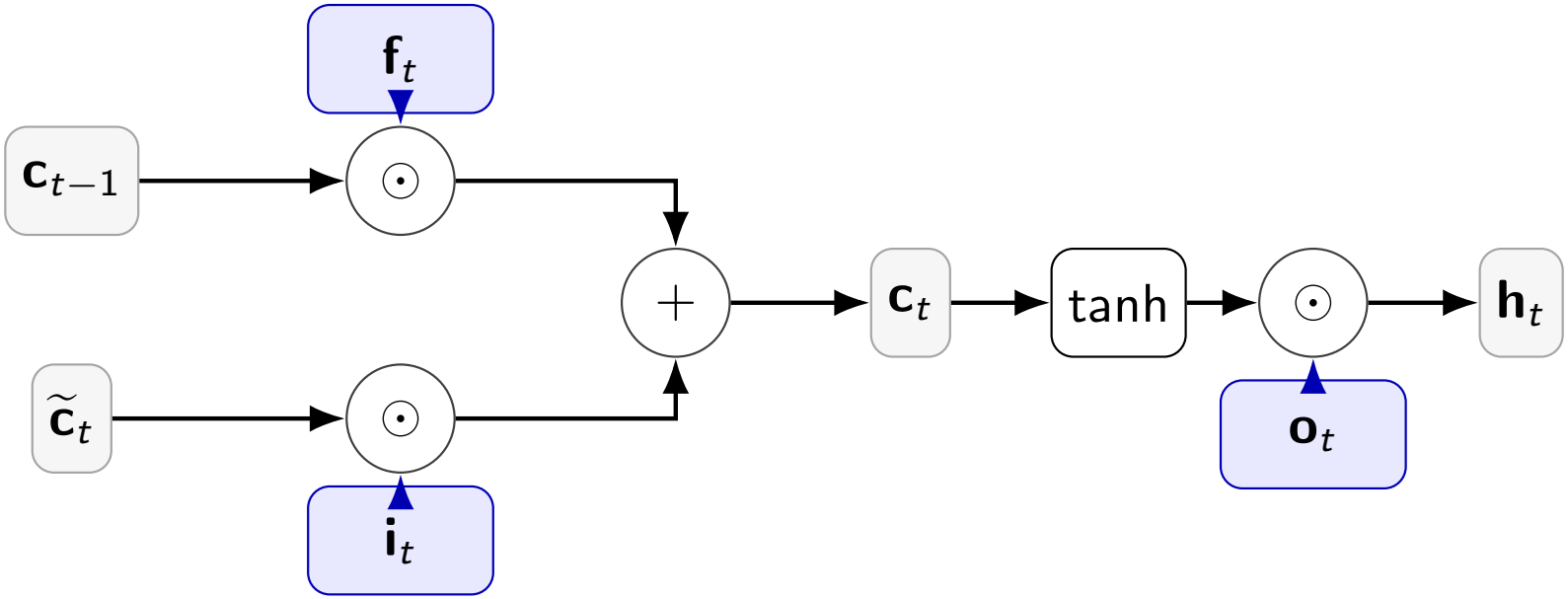
LSTM: Inside One Time Step

Q: Where do the forget, input, and output gates act?



LSTM: Inside One Time Step

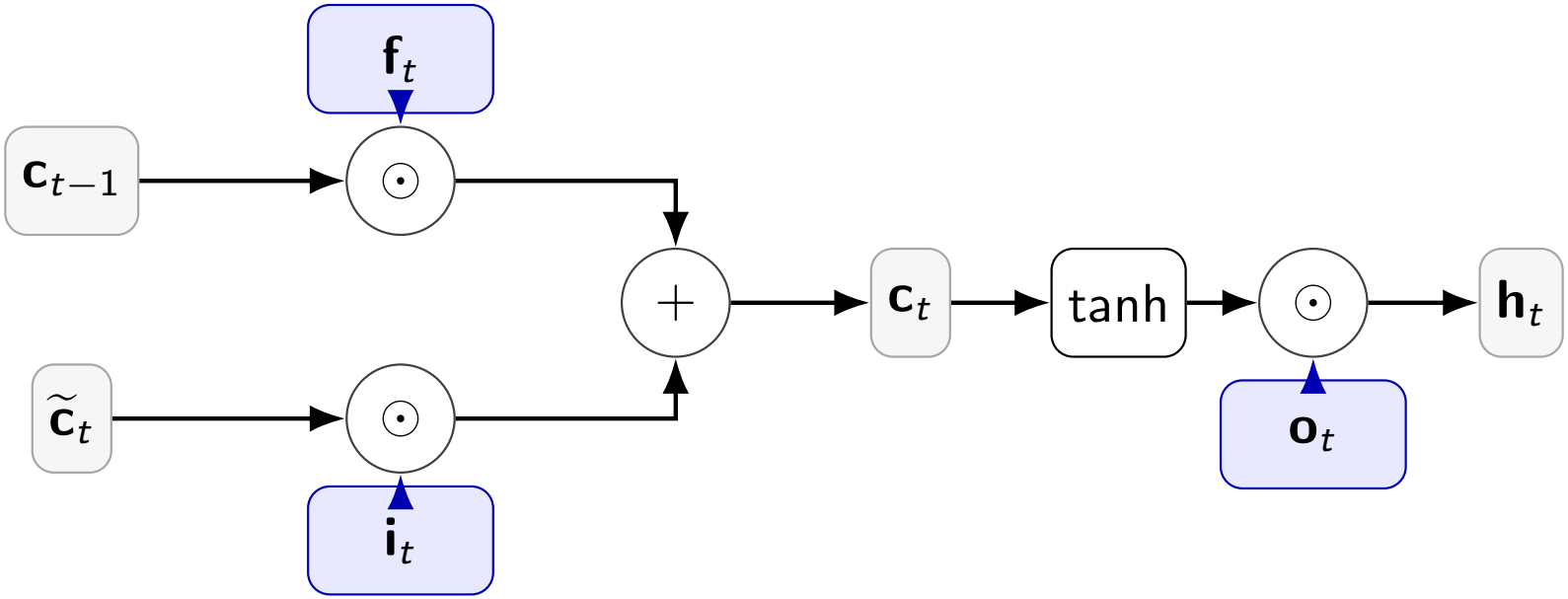
Q: Where do the forget, input, and output gates act?



- **Update cell memory:** $\mathbf{f}_t$ controls how much of $\mathbf{c}_{t-1}$ to retain; $\mathbf{i}_t$ controls how much candidate memory $\widetilde{\mathbf{c}}_t$ to add.

LSTM: Inside One Time Step

Q: Where do the forget, input, and output gates act?



- **Update cell memory:** $\mathbf{f}_t$ controls how much of $\mathbf{c}_{t-1}$ to retain; $\mathbf{i}_t$ controls how much candidate memory $\widetilde{\mathbf{c}}_t$ to add.
- **Produce hidden state:** $\mathbf{o}_t$ controls how much of the updated memory is exposed as $\mathbf{h}_t$. Both $\mathbf{c}_t$ and $\mathbf{h}_t$ pass to the next month.

The gate values and candidate memory are computed from $(\mathbf{u}_t, \mathbf{h}_{t-1})$ using learned parameters.

LSTM Gates: Mathematical Definition

Q: How are the three gates and candidate memory computed?

LSTM Gates: Mathematical Definition

Q: How are the three gates and candidate memory computed?

Using the current input $\mathbf{u}_t$ and previous hidden state $\mathbf{h}_{t-1}$:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{u}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(forget gate)}, \\
\mathbf{i}_t &= \sigma(W_i \mathbf{u}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(input gate)}, \\
\mathbf{o}_t &= \sigma(W_o \mathbf{u}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(output gate)}, \\
\widetilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{u}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(candidate memory)}.
\end{aligned}$$

LSTM Gates: Mathematical Definition

Q: How are the three gates and candidate memory computed?

Using the current input $\mathbf{u}_t$ and previous hidden state $\mathbf{h}_{t-1}$:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{u}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(forget gate)}, \\
\mathbf{i}_t &= \sigma(W_i \mathbf{u}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(input gate)}, \\
\mathbf{o}_t &= \sigma(W_o \mathbf{u}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(output gate)}, \\
\widetilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{u}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(candidate memory)}.
\end{aligned}$$

- $\sigma(z) = 1/(1 + e^{-z})$ produces gate values in $(0, 1)$.

LSTM Gates: Mathematical Definition

Q: How are the three gates and candidate memory computed?

Using the current input $\mathbf{u}_t$ and previous hidden state $\mathbf{h}_{t-1}$:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{u}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(forget gate)},\\
\mathbf{i}_t &= \sigma(W_i \mathbf{u}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(input gate)},\\
\mathbf{o}_t &= \sigma(W_o \mathbf{u}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(output gate)},\\
\widetilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{u}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(candidate memory)}.
\end{aligned}$$

- $\sigma(z) = 1/(1 + e^{-z})$ produces gate values in $(0, 1)$.
- The weights and biases are **learned** and reused at every month.

LSTM Gates: Mathematical Definition

Q: How are the three gates and candidate memory computed?

Using the current input $\mathbf{u}_t$ and previous hidden state $\mathbf{h}_{t-1}$:

$$\begin{aligned}
\mathbf{f}_t &= \sigma(W_f \mathbf{u}_t + U_f \mathbf{h}_{t-1} + \mathbf{b}_f) && \text{(forget gate)},\\
\mathbf{i}_t &= \sigma(W_i \mathbf{u}_t + U_i \mathbf{h}_{t-1} + \mathbf{b}_i) && \text{(input gate)},\\
\mathbf{o}_t &= \sigma(W_o \mathbf{u}_t + U_o \mathbf{h}_{t-1} + \mathbf{b}_o) && \text{(output gate)},\\
\widetilde{\mathbf{c}}_t &= \tanh(W_c \mathbf{u}_t + U_c \mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(candidate memory)}.
\end{aligned}$$

- $\sigma(z) = 1/(1+e^{-z})$ produces gate values in $(0, 1)$.
- The weights and biases are **learned** and reused at every month.
- These equations determine the gate values; the next slide shows how the gates actually update memory.

LSTM: Updating Memory and Producing a Forecast

Q: How do gate values affect what the network retains?

LSTM: Updating Memory and Producing a Forecast

Q: How do gate values affect what the network retains?

$$\mathbf{c}_t = \underbrace{\mathbf{f}_t \odot \mathbf{c}_{t-1}}_{\text{retain prior memory}} + \underbrace{\mathbf{i}_t \odot \widetilde{\mathbf{c}}_t}_{\text{add candidate memory}},$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

$$\widehat{s}_{t+1|t} = g(\mathbf{h}_t).$$

LSTM: Updating Memory and Producing a Forecast

Q: How do gate values affect what the network retains?

$$\mathbf{c}_t = \underbrace{\mathbf{f}_t \odot \mathbf{c}_{t-1}}_{\text{retain prior memory}} + \underbrace{\mathbf{i}_t \odot \widetilde{\mathbf{c}}_t}_{\text{add candidate memory}},$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

$$\widehat{s}_{t+1|t} = g(\mathbf{h}_t).$$

- $\odot$ indicates **componentwise multiplication**; the gates act separately on each component of the memory vector.

LSTM: Updating Memory and Producing a Forecast

Q: How do gate values affect what the network retains?

$$\mathbf{c}_t = \underbrace{\mathbf{f}_t \odot \mathbf{c}_{t-1}}_{\text{retain prior memory}} + \underbrace{\mathbf{i}_t \odot \widetilde{\mathbf{c}}_t}_{\text{add candidate memory}},$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

$$\widehat{s}_{t+1|t} = g(\mathbf{h}_t).$$

- $\odot$ indicates **componentwise multiplication**; the gates act separately on each component of the memory vector.
- If a forget-gate component is near one, much of that memory component persists. If an input-gate component is near zero, little new candidate information is added.

LSTM: Updating Memory and Producing a Forecast

Q: How do gate values affect what the network retains?

$$\mathbf{c}_t = \underbrace{\mathbf{f}_t \odot \mathbf{c}_{t-1}}_{\text{retain prior memory}} + \underbrace{\mathbf{i}_t \odot \widetilde{\mathbf{c}}_t}_{\text{add candidate memory}},$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t),$$

$$\widehat{s}_{t+1|t} = g(\mathbf{h}_t).$$

- $\odot$ indicates **componentwise multiplication**; the gates act separately on each component of the memory vector.
- If a forget-gate component is near one, much of that memory component persists. If an input-gate component is near zero, little new candidate information is added.
- g is an output layer that maps the hidden state to a sales forecast (or parameters of a predictive distribution).

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

- Read observed inputs $\mathbf{u}_{t-L+1}, \ldots, \mathbf{u}_t$ to obtain the final states $(\mathbf{h}_t, \mathbf{c}_t)$.

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

- Read observed inputs $\mathbf{u}_{t-L+1}, \ldots, \mathbf{u}_t$ to obtain the final states $(\mathbf{h}_t, \mathbf{c}_t)$.
- **Direct multi-output forecast:** map the final state to all future months in one step:

$$(\widehat{s}_{t+1|t}, \ldots, \widehat{s}_{t+12|t}) = g_{1:12}(\mathbf{h}_t).$$

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

- Read observed inputs $\mathbf{u}_{t-L+1}, \ldots, \mathbf{u}_t$ to obtain the final states $(\mathbf{h}_t, \mathbf{c}_t)$.
- **Direct multi-output forecast:** map the final state to all future months in one step:

$$(\widehat{s}_{t+1|t}, \ldots, \widehat{s}_{t+12|t}) = g_{1:12}(\mathbf{h}_t).$$

- **Recursive forecast:** predict month $t+1$, feed its predicted value back into subsequent recurrent steps, and continue to month $t+12$.

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

- Read observed inputs $\mathbf{u}_{t-L+1}, \ldots, \mathbf{u}_t$ to obtain the final states $(\mathbf{h}_t, \mathbf{c}_t)$.
- **Direct multi-output forecast:** map the final state to all future months in one step:

$$(\widehat{s}_{t+1|t}, \ldots, \widehat{s}_{t+12|t}) = g_{1:12}(\mathbf{h}_t).$$

- **Recursive forecast:** predict month $t+1$, feed its predicted value back into subsequent recurrent steps, and continue to month $t+12$.
- Announced promotions and calendar information may be known for future months; unknown future drivers need scenarios or separate forecasts.

LSTM: From States to Multi-Step Sales Forecasts

Q: How can one recurrent representation produce the next 12 months?

- Read observed inputs $\mathbf{u}_{t-L+1}, \ldots, \mathbf{u}_t$ to obtain the final states $(\mathbf{h}_t, \mathbf{c}_t)$.
- **Direct multi-output forecast:** map the final state to all future months in one step:

$$(\widehat{s}_{t+1|t}, \ldots, \widehat{s}_{t+12|t}) = g_{1:12}(\mathbf{h}_t).$$

- **Recursive forecast:** predict month $t+1$, feed its predicted value back into subsequent recurrent steps, and continue to month $t+12$.
- Announced promotions and calendar information may be known for future months; unknown future drivers need scenarios or separate forecasts.
- **Tradeoff:** direct outputs avoid recursive error propagation; recursive decoding reuses a sequential prediction rule, but errors can accumulate.

From LSTMs to Transformers

Q: How can two architectures use distant observations differently?

From LSTMs to Transformers

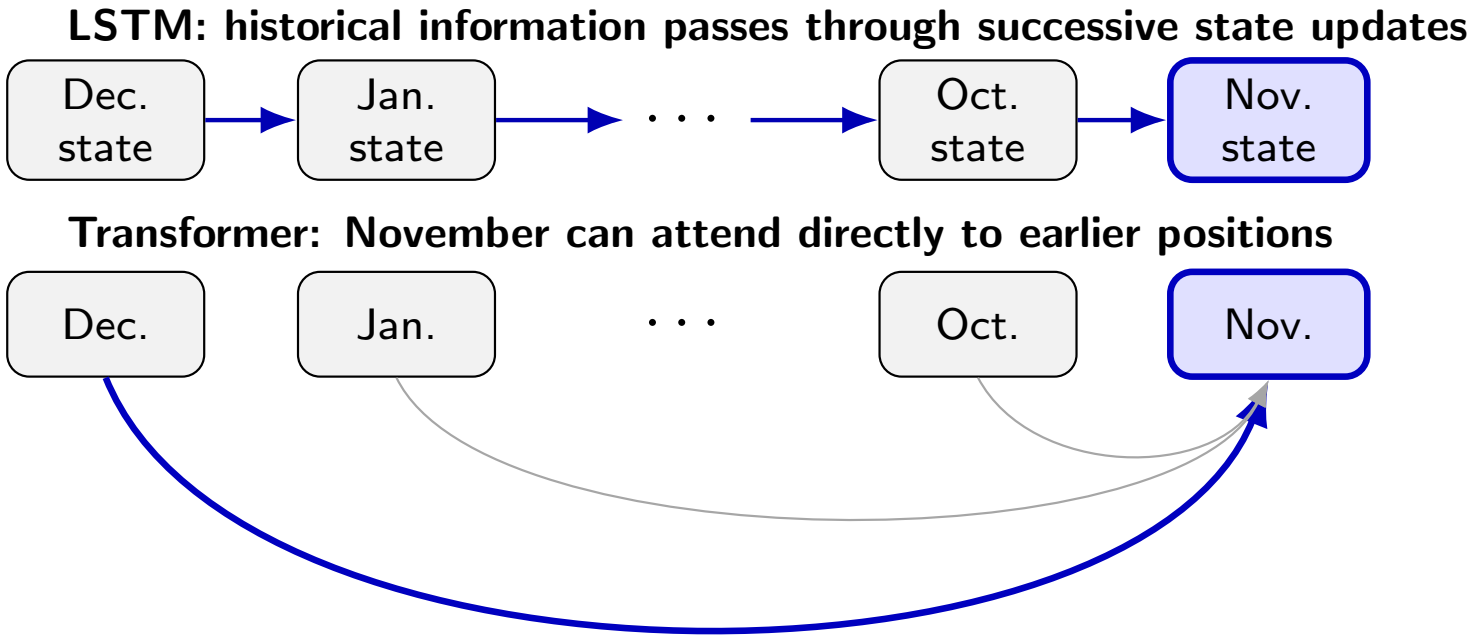
Q: How can two architectures use distant observations differently?

Two architectures: LSTMs use *recurrence*; transformers use *attention*.

From LSTMs to Transformers

Q: How can two architectures use distant observations differently?

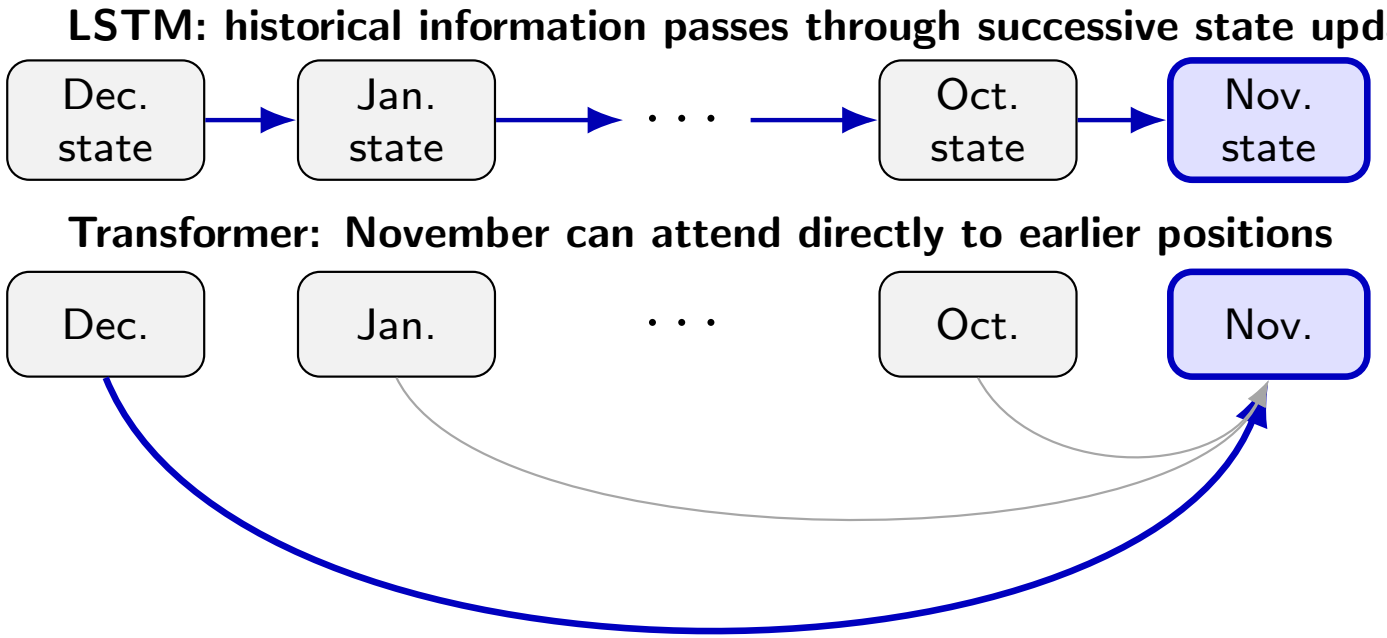
Two architectures: LSTMs use *recurrence*; transformers use *attention*.



From LSTMs to Transformers

Q: How can two architectures use distant observations differently?

Two architectures: LSTMs use *recurrence*; transformers use *attention*.



- **Both can use last December's information.** An LSTM carries it forward through state updates; attention lets November's representation combine information from December directly.

Attention: Queries, Keys, and Values – Intuition

Q: How does attention decide which historical observations matter?

Attention: Queries, Keys, and Values – Intuition

Q: How does attention decide which historical observations matter?

Econometric analogy: Attention resembles a learned, observation-dependent weighting scheme, similar in spirit to kernel regression. A *representation* is a learned feature vector, **not** a prediction.

Attention: Queries, Keys, and Values – Intuition

Q: How does attention decide which historical observations matter?

Econometric analogy: Attention resembles a learned, observation-dependent weighting scheme, similar in spirit to kernel regression. A *representation* is a learned feature vector, **not** a prediction.

Concept	**Intuition**	**Sales example**
Query	What characteristics am I looking for?	A learned vector from November's observed features used to assess which past months are relevant.
Key	How well does this observation match what I am looking for?	Each month supplies a learned matching vector compared with November's query.
Value	What information does the observation contribute?	Each month supplies learned information combined using those weights.

Attention: Queries, Keys, and Values – Intuition

Q: How does attention decide which historical observations matter?

Econometric analogy: Attention resembles a learned, observation-dependent weighting scheme, similar in spirit to kernel regression. A *representation* is a learned feature vector, **not** a prediction.

Concept	**Intuition**	**Sales example**
Query	What characteristics am I looking for?	A learned vector from November's observed features used to assess which past months are relevant.
Key	How well does this observation match what I am looking for?	Each month supplies a learned matching vector compared with November's query.
Value	What information does the observation contribute?	Each month supplies learned information combined using those weights.

- **Key distinction:** Queries and keys determine the *weights*; values supply the *information being weighted*.

Attention: Queries, Keys, and Values – Intuition

Q: How does attention decide which historical observations matter?

Econometric analogy: Attention resembles a learned, observation-dependent weighting scheme, similar in spirit to kernel regression. A *representation* is a learned feature vector, **not** a prediction.

Concept	**Intuition**	**Sales example**
Query	What characteristics am I looking for?	A learned vector from November's observed features used to assess which past months are relevant.
Key	How well does this observation match what I am looking for?	Each month supplies a learned matching vector compared with November's query.
Value	What information does the observation contribute?	Each month supplies learned information combined using those weights.

- **Key distinction:** Queries and keys determine the *weights*; values supply the *information being weighted*.

Transformer Attention: Direct Historical Comparisons

Q: What does it mean for November to attend to earlier months?

Transformer Attention: Direct Historical Comparisons

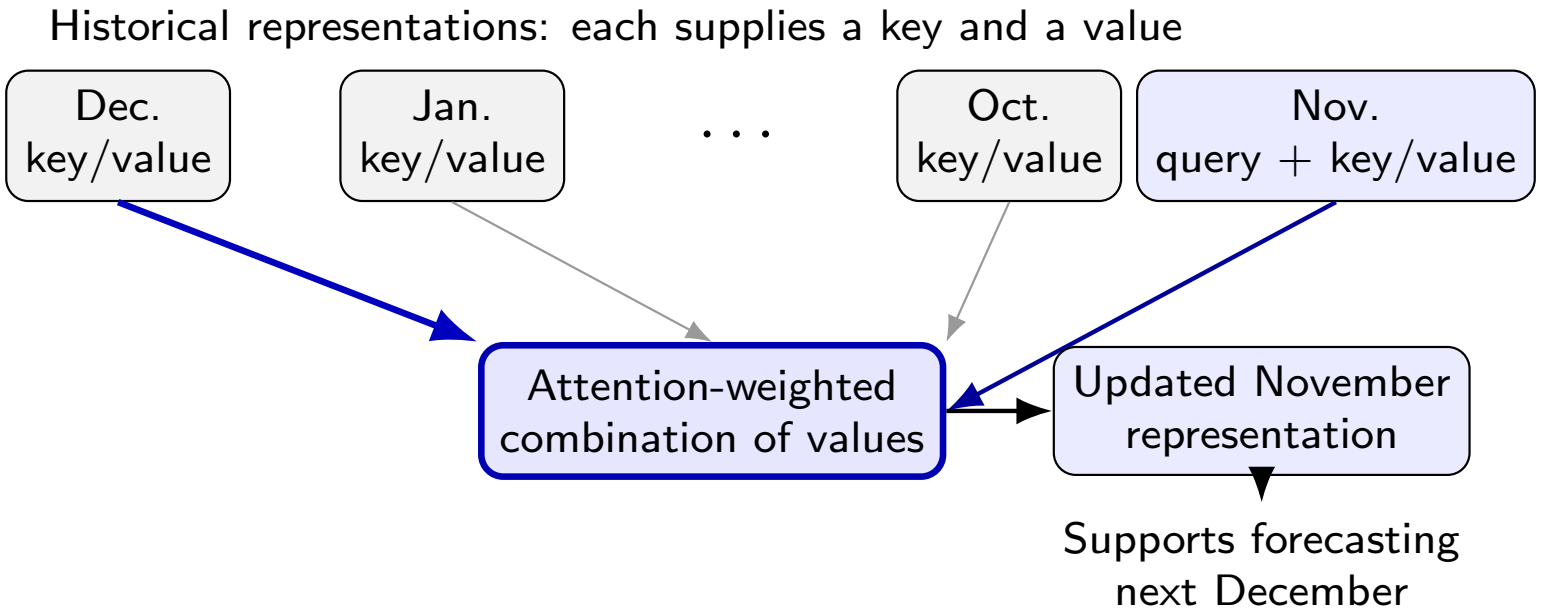
Q: What does it mean for November to attend to earlier months?

Example: To update November's representation, the model compares a *query* from November with *keys* from observed months.

Transformer Attention: Direct Historical Comparisons

Q: What does it mean for November to attend to earlier months?

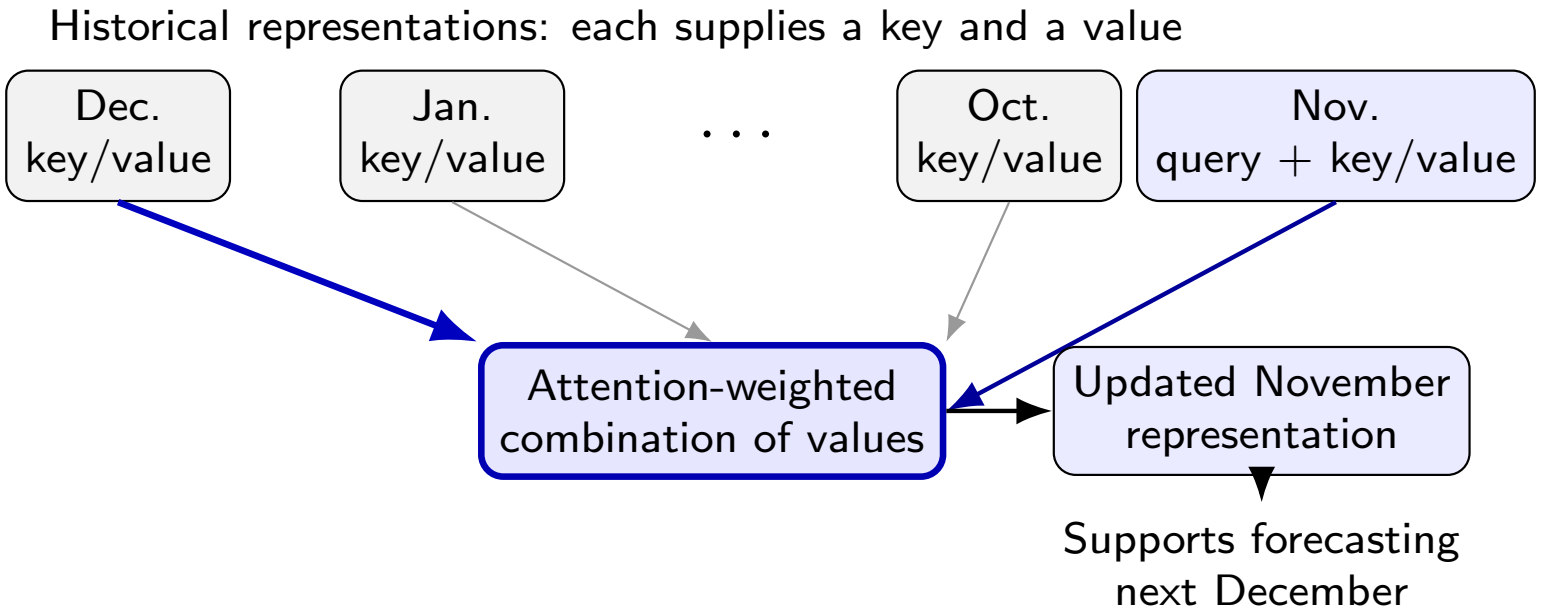
Example: To update November's representation, the model compares a *query* from November with *keys* from observed months.



Transformer Attention: Direct Historical Comparisons

Q: What does it mean for November to attend to earlier months?

Example: To update November's representation, the model compares a *query* from November with *keys* from observed months.



- **Attention:** November's query is compared with historical keys to determine weights. Those weights combine historical values into an updated November representation, which can support future forecasts.

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

- Form three learned projections:

$$Q = EW_Q, \qquad K = EW_K, \qquad V = EW_V.$$

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

- Form three learned projections:

$$Q = EW_Q, \qquad K = EW_K, \qquad V = EW_V.$$

- Compute scaled dot-product attention:

$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK'}{\sqrt{d_k}} + M\right) V.}$$

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

- Form three learned projections:

$$Q = EW_Q, \qquad K = EW_K, \qquad V = EW_V.$$

- Compute scaled dot-product attention:

$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK'}{\sqrt{d_k}} + M\right) V.}$$

- **Queries and keys:** QK' produces relevance scores between positions. Dividing by $\sqrt{d_k}$ stabilizes their scale.

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

- Form three learned projections:

$$Q = EW_Q, \qquad K = EW_K, \qquad V = EW_V.$$

- Compute scaled dot-product attention:

$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK'}{\sqrt{d_k}} + M\right) V.}$$

- **Queries and keys:** QK' produces relevance scores between positions. Dividing by $\sqrt{d_k}$ stabilizes their scale.
- **Softmax:** turns each query's scores into nonnegative weights summing to one. **Values:** V contains the information those weights combine.

Attention: Queries, Keys, and Values – Mathematics

Q: How does the model compute relevance between historical positions?

Let $E \in \mathbb{R}^{L \times d}$ contain L learned, position-aware historical representations.

- Form three learned projections:

$$Q = EW_Q, \qquad K = EW_K, \qquad V = EW_V.$$

- Compute scaled dot-product attention:

$$\boxed{\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK'}{\sqrt{d_k}} + M\right) V.}$$

- **Queries and keys:** QK' produces relevance scores between positions. Dividing by $\sqrt{d_k}$ stabilizes their scale.
- **Softmax:** turns each query's scores into nonnegative weights summing to one. **Values:** V contains the information those weights combine.
- M optionally masks positions that should not be visible.

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

- **Recall LSTM:** Processing observations in sequence makes their order affect the evolving hidden and cell states.

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

- **Recall LSTM:** Processing observations in sequence makes their order affect the evolving hidden and cell states.
- **Self-attention alone is different:** Without positional information or order-dependent masking, it compares content without encoding which month came first.

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

- **Recall LSTM:** Processing observations in sequence makes their order affect the evolving hidden and cell states.
- **Self-attention alone is different:** Without positional information or order-dependent masking, it compares content without encoding which month came first.
- Consider the same sales observations in different orders:

$$\underbrace{(90, 100, 110)}_{\text{Rising sales}} \qquad \underbrace{(110, 100, 90)}_{\text{Declining sales}}.$$

The distinction is *when* each value was observed.

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

- **Recall LSTM:** Processing observations in sequence makes their order affect the evolving hidden and cell states.
- **Self-attention alone is different:** Without positional information or order-dependent masking, it compares content without encoding which month came first.
- Consider the same sales observations in different orders:

$$\underbrace{(90, 100, 110)}_{\text{Rising sales}} \qquad \underbrace{(110, 100, 90)}_{\text{Declining sales}}.$$

  The distinction is *when* each value was observed.
- **Solution:** Incorporate the position of each month into its learned input features, for example:

$$\mathbf{e}_j = \text{Embed}(\mathbf{u}_j) + \text{PE}(j),$$

  where $\text{PE}(j)$ identifies position j in the window.

Transformers and Temporal Ordering

Q: How does a transformer distinguish rising from falling sales?

- **Recall LSTM:** Processing observations in sequence makes their order affect the evolving hidden and cell states.
- **Self-attention alone is different:** Without positional information or order-dependent masking, it compares content without encoding which month came first.
- Consider the same sales observations in different orders:

$$\underbrace{(90, 100, 110)}_{\text{Rising sales}} \qquad \underbrace{(110, 100, 90)}_{\text{Declining sales}}.$$

  The distinction is *when* each value was observed.
- **Solution:** Incorporate the position of each month into its learned input features, for example:

$$\mathbf{e}_j = \text{Embed}(\mathbf{u}_j) + \text{PE}(j),$$

  where $\text{PE}(j)$ identifies position j in the window.
- **Distinction:** Position encodes sequence order; calendar variables (e.g., month of year) can encode seasonality.

Transformer Architecture: Stacking Blocks

Q: Where does depth enter a transformer?

Transformer Architecture: Stacking Blocks

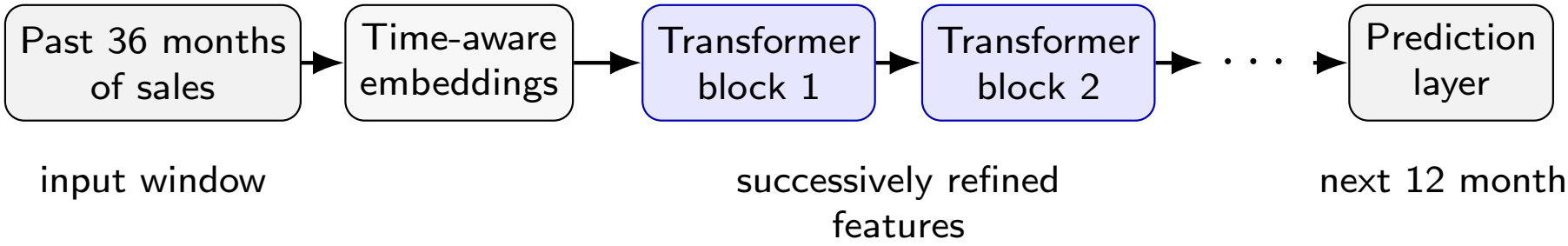
Q: Where does depth enter a transformer?

A transformer is a neural network built from repeated blocks, not a single attention calculation.

Transformer Architecture: Stacking Blocks

Q: Where does depth enter a transformer?

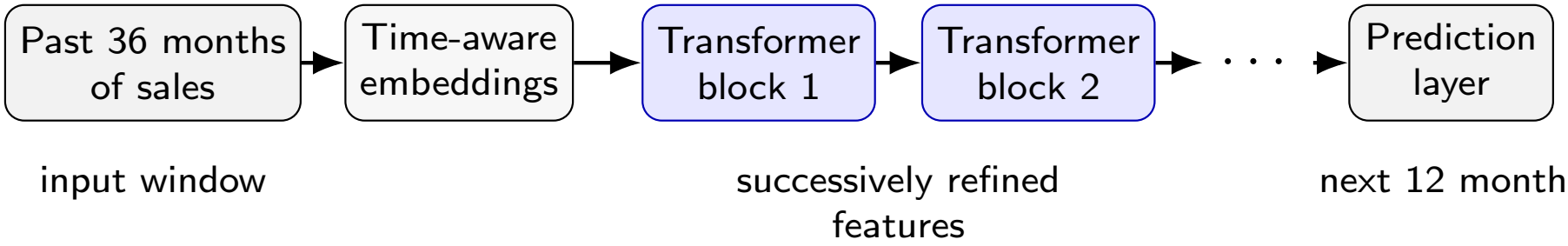
A transformer is a neural network built from repeated blocks, not a single attention calculation.



Transformer Architecture: Stacking Blocks

Q: Where does depth enter a transformer?

A transformer is a neural network built from repeated blocks, not a single attention calculation.

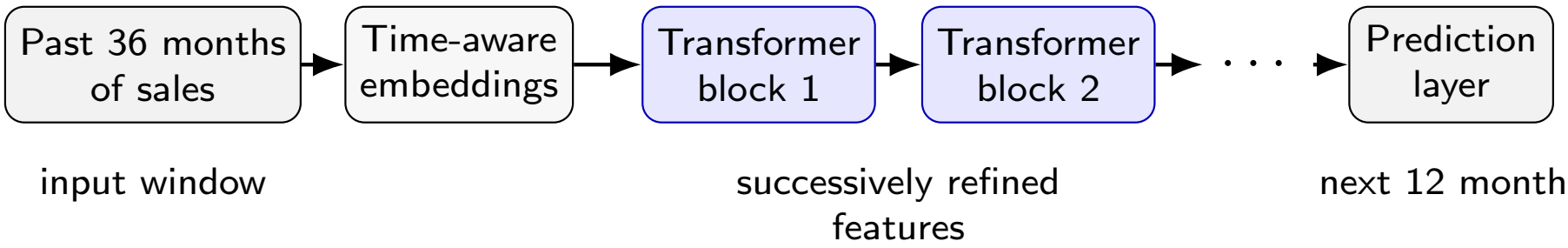


- **Each block performs learned feature engineering:** It takes the features constructed by the preceding block and transforms them, using attention to incorporate information across historical observations.

Transformer Architecture: Stacking Blocks

Q: Where does depth enter a transformer?

A transformer is a neural network built from repeated blocks, not a single attention calculation.



- **Each block performs learned feature engineering:** It takes the features constructed by the preceding block and transforms them, using attention to incorporate information across historical observations.
- **Stacking blocks creates depth:** Successive blocks learn increasingly refined features. A final prediction layer converts these features into sales forecasts.

Inside a Transformer Block

Q: What happens to historical information within one block?

Inside a Transformer Block

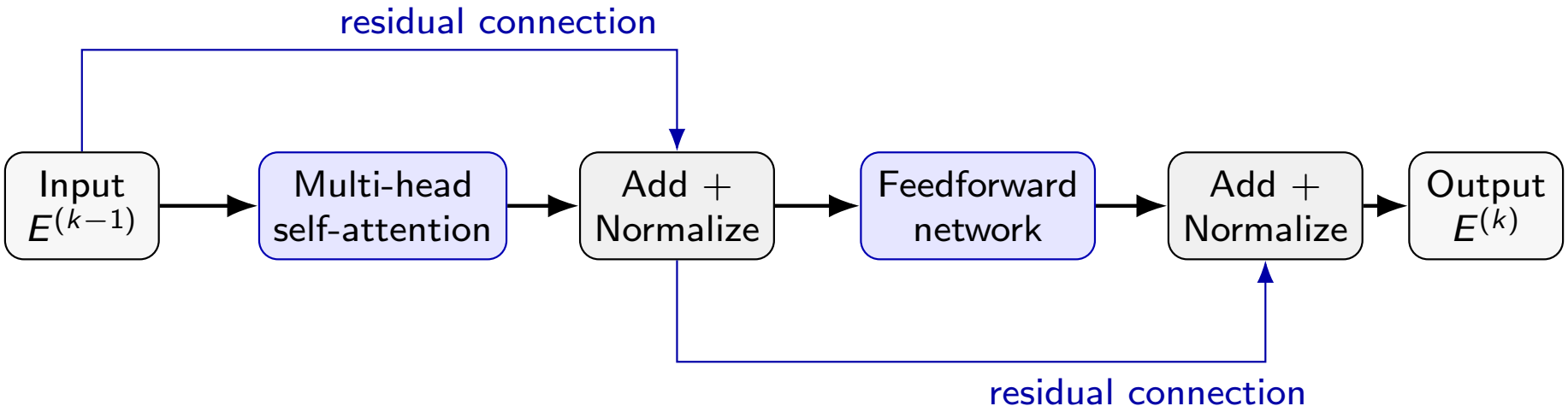
Q: What happens to historical information within one block?

A **block** is a subnetwork, not a single hidden layer. Unlike N-BEATS blocks, which contribute forecasts, transformer blocks produce updated learned features.

Inside a Transformer Block

Q: What happens to historical information within one block?

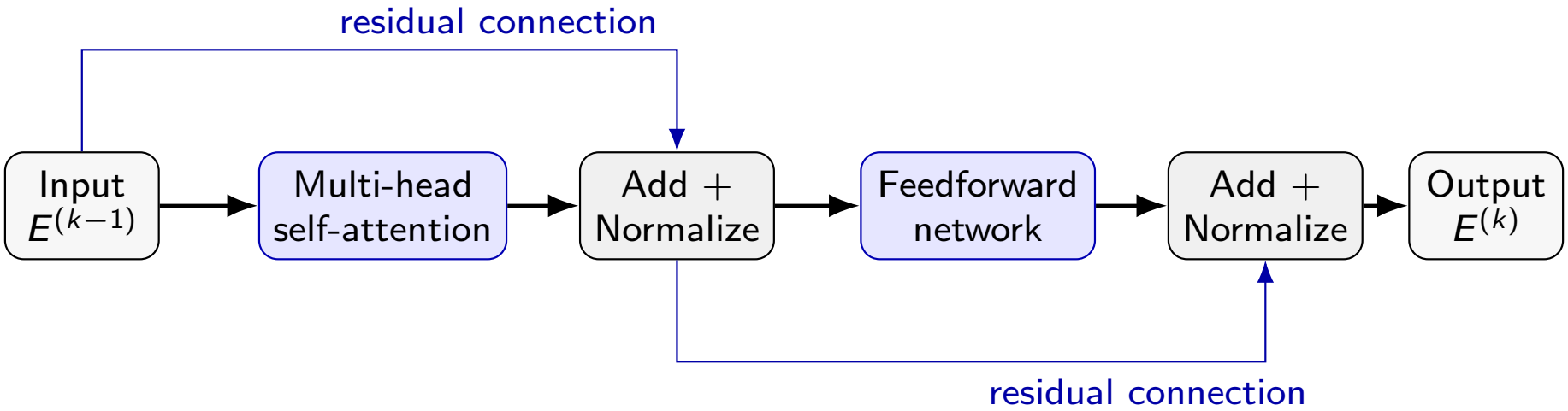
A **block** is a subnetwork, not a single hidden layer. Unlike N-BEATS blocks, which contribute forecasts, transformer blocks produce updated learned features.



Inside a Transformer Block

Q: What happens to historical information within one block?

A **block** is a subnetwork, not a single hidden layer. Unlike N-BEATS blocks, which contribute forecasts, transformer blocks produce updated learned features.

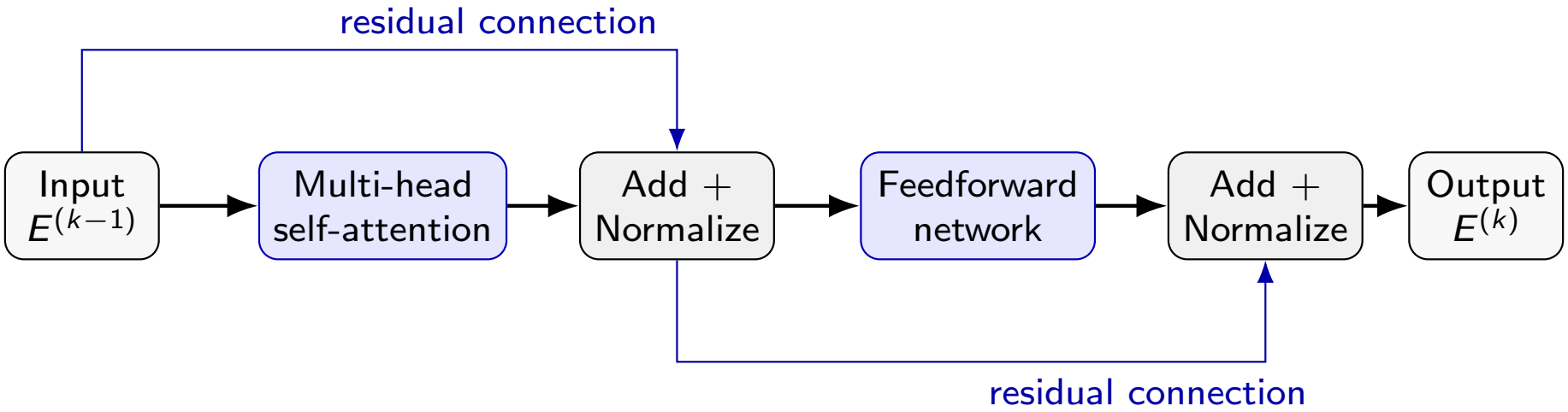


- **Self-attention** lets representations at different historical positions exchange information.

Inside a Transformer Block

Q: What happens to historical information within one block?

A **block** is a subnetwork, not a single hidden layer. Unlike N-BEATS blocks, which contribute forecasts, transformer blocks produce updated learned features.

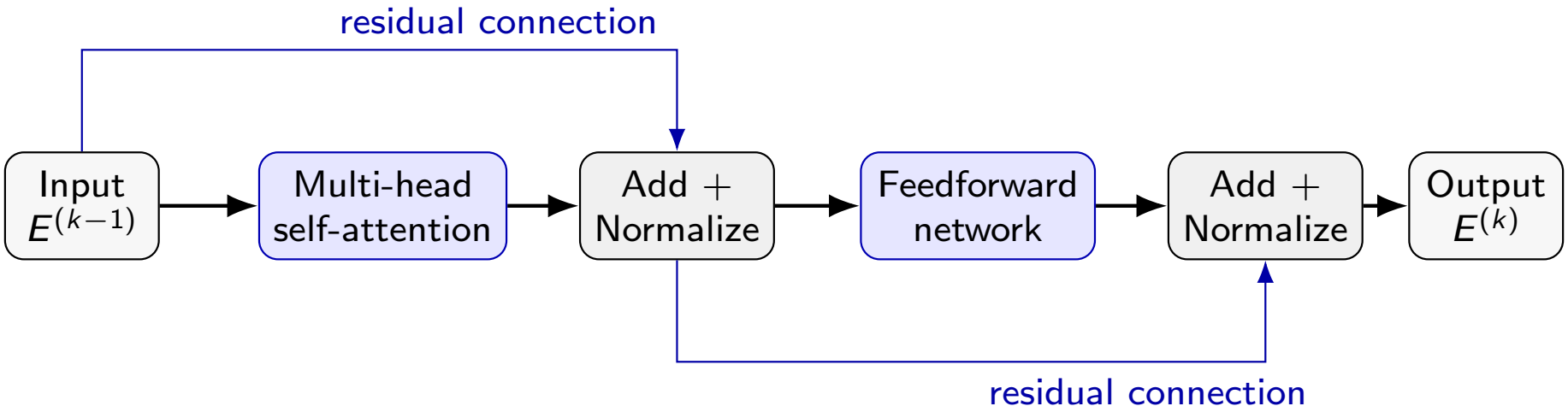


- **Self-attention** lets representations at different historical positions exchange information.
- The **feedforward network** applies learned nonlinear transformations to each position's updated representation.

Inside a Transformer Block

Q: What happens to historical information within one block?

A **block** is a subnetwork, not a single hidden layer. Unlike N-BEATS blocks, which contribute forecasts, transformer blocks produce updated learned features.



- **Self-attention** lets representations at different historical positions exchange information.
- The **feedforward network** applies learned nonlinear transformations to each position's updated representation.
- **Residual connections and normalization** support training when many blocks are stacked.

Transformer Forecasts: From History to Future Sales

Q: How are learned historical features converted into 12 forecasts?

Transformer Forecasts: From History to Future Sales

Q: How are learned historical features converted into 12 forecasts?

- Transformer blocks construct **context-dependent features** from the observed history. A prediction layer maps them into sales forecasts.

Transformer Forecasts: From History to Future Sales

Q: How are learned historical features converted into 12 forecasts?

- Transformer blocks construct **context-dependent features** from the observed history. A prediction layer maps them into sales forecasts.
- As with LSTMs, there are two common output strategies:
  - **Direct multi-horizon:** Predict all 12 future months at once.
  - **Recursive:** Generate forecasts sequentially, using earlier predictions when forecasting later months.

Transformer Forecasts: From History to Future Sales

Q: How are learned historical features converted into 12 forecasts?

- Transformer blocks construct **context-dependent features** from the observed history. A prediction layer maps them into sales forecasts.
- As with LSTMs, there are two common output strategies:
  - **Direct multi-horizon:** Predict all 12 future months at once.
  - **Recursive:** Generate forecasts sequentially, using earlier predictions when forecasting later months.
- Known future calendar variables or announced promotions can be supplied where supported; unknown drivers require scenarios or separate forecasts.

Transformer Forecasts: From History to Future Sales

Q: How are learned historical features converted into 12 forecasts?

- Transformer blocks construct **context-dependent features** from the observed history. A prediction layer maps them into sales forecasts.
- As with LSTMs, there are two common output strategies:
  - **Direct multi-horizon:** Predict all 12 future months at once.
  - **Recursive:** Generate forecasts sequentially, using earlier predictions when forecasting later months.
- Known future calendar variables or announced promotions can be supplied where supported; unknown drivers require scenarios or separate forecasts.
- **Distinction:** Attention determines how the historical features are constructed. The output strategy determines how future predictions are generated.

Comparing Neural-Network Forecasting Architectures

Q: How does each architecture represent sales history?

Comparing Neural-Network Forecasting Architectures

Q: How does each architecture represent sales history?

Architecture	**How historical information is processed**
Feedforward NN	A fixed vector of lagged sales passes through nonlinear transformations.
N-BEATS / N-HiTS	A fixed window passes through specialized forecasting blocks; N-HiTS uses multiple resolutions.
RNN / LSTM	A hidden state evolves as each month's observations are processed; LSTM adds gated memory.
Transformer	Attention combines representations of historical positions or patches, using positional information.

Comparing Neural-Network Forecasting Architectures

Q: How does each architecture represent sales history?

Architecture	**How historical information is processed**
Feedforward NN	A fixed vector of lagged sales passes through nonlinear transformations.
N-BEATS / N-HiTS	A fixed window passes through specialized forecasting blocks; N-HiTS uses multiple resolutions.
RNN / LSTM	A hidden state evolves as each month's observations are processed; LSTM adds gated memory.
Transformer	Attention combines representations of historical positions or patches, using positional information.

Takeaway: These architectures differ in *how they process temporal information*. Network depth is a separate modeling choice.

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

- **Fixed windows and relatively simple patterns:** A feedforward network is a useful baseline; extra architectural complexity may not pay off.

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

- **Fixed windows and relatively simple patterns:** A feedforward network is a useful baseline; extra architectural complexity may not pay off.
- **Multi-horizon structure and several temporal scales:** N-BEATS/N-HiTS offer forecasting-specific feedforward designs.

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

- **Fixed windows and relatively simple patterns:** A feedforward network is a useful baseline; extra architectural complexity may not pay off.
- **Multi-horizon structure and several temporal scales:** N-BEATS/N-HiTS offer forecasting-specific feedforward designs.
- **Sequential updates and learned state:** LSTMs can be attractive when it is useful to update a representation as new observations arrive.

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

- **Fixed windows and relatively simple patterns:** A feedforward network is a useful baseline; extra architectural complexity may not pay off.
- **Multi-horizon structure and several temporal scales:** N-BEATS/N-HiTS offer forecasting-specific feedforward designs.
- **Sequential updates and learned state:** LSTMs can be attractive when it is useful to update a representation as new observations arrive.
- **Relationships across positions in a long context:** Attention-based models may help, but their data and computation requirements must be considered.

Selecting an Architecture for Sales Forecasting

Q: What considerations might favor one network over another?

- **Fixed windows and relatively simple patterns:** A feedforward network is a useful baseline; extra architectural complexity may not pay off.
- **Multi-horizon structure and several temporal scales:** N-BEATS/N-HiTS offer forecasting-specific feedforward designs.
- **Sequential updates and learned state:** LSTMs can be attractive when it is useful to update a representation as new observations arrive.
- **Relationships across positions in a long context:** Attention-based models may help, but their data and computation requirements must be considered.
- **Empirical decision:** Compare against simple benchmarks using genuinely out-of-sample forecasts. No architecture consistently wins merely because it is newer or deeper.

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

- **No.** Feedforward networks, LSTMs, and transformers can be trained on one SKU's history (*local*) or across many related SKUs (*global*).

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

- **No.** Feedforward networks, LSTMs, and transformers can be trained on one SKU's history (*local*) or across many related SKUs (*global*).
- **Architecture** specifies how temporal information is processed.

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

- **No.** Feedforward networks, LSTMs, and transformers can be trained on one SKU's history (*local*) or across many related SKUs (*global*).
- **Architecture** specifies how temporal information is processed.
- **Training scope** specifies which SKU histories contribute to estimating shared parameters.

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

- **No.** Feedforward networks, LSTMs, and transformers can be trained on one SKU's history (*local*) or across many related SKUs (*global*).
- **Architecture** specifies how temporal information is processed.
- **Training scope** specifies which SKU histories contribute to estimating shared parameters.
- **Pretraining** is yet another decision: A model may first learn from a broad collection of series, then be adapted or used for our sales forecasts.

Architecture Is Not Training Scope

Q: Must an LSTM or transformer be trained separately for every SKU?

- **No.** Feedforward networks, LSTMs, and transformers can be trained on one SKU's history (*local*) or across many related SKUs (*global*).
- **Architecture** specifies how temporal information is processed.
- **Training scope** specifies which SKU histories contribute to estimating shared parameters.
- **Pretraining** is yet another decision: A model may first learn from a broad collection of series, then be adapted or used for our sales forecasts.
- **Next:** How can we exploit shared patterns across many SKUs and divisions without assuming every series behaves identically?

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

From Local to Global Models

Q: What if each SKU has too little history to estimate a rich forecasting model?

From Local to Global Models

Q: What if each SKU has too little history to estimate a rich forecasting model?

- A company sells many SKUs across several divisions. Let $s_t^{(i)}$ denote monthly dollar sales of SKU i.

From Local to Global Models

Q: What if each SKU has too little history to estimate a rich forecasting model?

- A company sells many SKUs across several divisions. Let $s_t^{(i)}$ denote monthly dollar sales of SKU i.
- **Local approach:** one model for every SKU, estimated only on its own history.
- **Global approach:** one set of parameters trained using forecast examples from many SKUs.

From Local to Global Models

Q: What if each SKU has too little history to estimate a rich forecasting model?

- A company sells many SKUs across several divisions. Let $s_t^{(i)}$ denote monthly dollar sales of SKU i.
- **Local approach:** one model for every SKU, estimated only on its own history.
- **Global approach:** one set of parameters trained using forecast examples from many SKUs.
- Related SKUs may share seasonality, pricing responses, and promotion patterns.
- But

From Local to Global Models

Q: What if each SKU has too little history to estimate a rich forecasting model?

- A company sells many SKUs across several divisions. Let $s_t^{(i)}$ denote monthly dollar sales of SKU i.
- **Local approach:** one model for every SKU, estimated only on its own history.
- **Global approach:** one set of parameters trained using forecast examples from many SKUs.
- Related SKUs may share seasonality, pricing responses, and promotion patterns.
- But pooling dissimilar products can also hurt: common parameters are not automatically appropriate for every SKU.

Local vs Pooled AR: A First Example

Q: What does parameter sharing mean in familiar regression notation?

Local vs Pooled AR: A First Example

Q: What does parameter sharing mean in familiar regression notation?

- **Separate local AR(1) models:**

$$s_t^{(i)} = a_i + \phi_i s_{t-1}^{(i)} + \varepsilon_t^{(i)}.$$

Local vs Pooled AR: A First Example

Q: What does parameter sharing mean in familiar regression notation?

- **Separate local AR(1) models:**

$$s_t^{(i)} = a_i + \phi_i s_{t-1}^{(i)} + \varepsilon_t^{(i)}.$$

- **Fully pooled AR(1):**

$$s_t^{(i)} = a + \phi s_{t-1}^{(i)} + \varepsilon_t^{(i)}.$$

Local vs Pooled AR: A First Example

Q: What does parameter sharing mean in familiar regression notation?

- **Separate local AR(1) models:**

$$s_t^{(i)} = a_i + \phi_i s_{t-1}^{(i)} + \varepsilon_t^{(i)}.$$

- **Fully pooled AR(1):**

$$s_t^{(i)} = a + \phi s_{t-1}^{(i)} + \varepsilon_t^{(i)}.$$

- All SKU–month rows contribute to estimation of the same a and ϕ.
- Pooling may improve precision when histories are short and common dynamics are credible; it may introduce bias when they are not.

Pooled AR: Estimation and Tradeoff

Q: What exactly changes when we stack SKU histories?

Pooled AR: Estimation and Tradeoff

Q: What exactly changes when we stack SKU histories?

- Estimate a shared slope by minimizing a joint objective:

$$(\widehat{a}, \widehat{\phi}) = \arg\min_{a,\phi} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \left(s_t^{(i)} - a - \phi s_{t-1}^{(i)} \right)^2.$$

Pooled AR: Estimation and Tradeoff

Q: What exactly changes when we stack SKU histories?

- Estimate a shared slope by minimizing a joint objective:

$$(\widehat{a}, \widehat{\phi}) = \arg\min_{a,\phi} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \left(s_t^{(i)} - a - \phi s_{t-1}^{(i)} \right)^2.$$

- Pooled AR is already a **global model** even though its functional form is classical and linear.

Pooled AR: Estimation and Tradeoff

Q: What exactly changes when we stack SKU histories?

- Estimate a shared slope by minimizing a joint objective:

$$(\widehat{a}, \widehat{\phi}) = \arg\min_{a,\phi} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \left(s_t^{(i)} - a - \phi s_{t-1}^{(i)} \right)^2.$$

- Pooled AR is already a **global model** even though its functional form is classical and linear.
- Its effective information depends on within-SKU dependence and cross-SKU correlation: $N \times T$ rows are not $N \times T$ independent observations.

Pooled AR: Estimation and Tradeoff

Q: What exactly changes when we stack SKU histories?

- Estimate a shared slope by minimizing a joint objective:

$$(\widehat{a}, \widehat{\phi}) = \arg\min_{a,\phi} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \left(s_t^{(i)} - a - \phi s_{t-1}^{(i)} \right)^2.$$

- Pooled AR is already a **global model** even though its functional form is classical and linear.
- Its effective information depends on within-SKU dependence and cross-SKU correlation: $N \times T$ rows are not $N \times T$ independent observations.
- Alternatives: allow SKU intercepts a_i, division-specific slopes ϕ_d, partial pooling, or regularized hierarchical coefficients.

From Pooled AR to Flexible Global Models

Q: Can SKUs share patterns without assuming one identical sales response?

From Pooled AR to Flexible Global Models

Q: Can SKUs share patterns without assuming one identical sales response?

- General model:

$$\widehat{s}_{t+h|t}^{(i)} = f_h\left(\mathbf{x}_t^{(i)}, \mathbf{z}^{(i)}; \theta\right),$$

where θ is shared; $\mathbf{z}^{(i)}$ contains known SKU/division characteristics.

From Pooled AR to Flexible Global Models

Q: Can SKUs share patterns without assuming one identical sales response?

- General model:

$$\widehat{s}_{t+h|t}^{(i)} = f_h\left(\mathbf{x}_t^{(i)}, \mathbf{z}^{(i)}; \theta\right),$$

  where θ is shared; $\mathbf{z}^{(i)}$ contains known SKU/division characteristics.
- f_h can be linear, boosted trees, feedforward, recurrent, or transformer-based.

From Pooled AR to Flexible Global Models

Q: Can SKUs share patterns without assuming one identical sales response?

- General model:

$$\widehat{s}_{t+h|t}^{(i)} = f_h\left(\mathbf{x}_t^{(i)}, \mathbf{z}^{(i)}; \theta\right),$$

  where θ is shared; $\mathbf{z}^{(i)}$ contains known SKU/division characteristics.
- f_h can be linear, boosted trees, feedforward, recurrent, or transformer-based.
- **Nonlinearity** lets the response depend on the state; **SKU characteristics** help distinguish structurally different products.
- A flexible model cannot distinguish otherwise identical inputs unless relevant differences are supplied or represented.

Global Training Dataset

Q: Where do the supervised examples come from?

Global Training Dataset

Q: Where do the supervised examples come from?

- Combine valid historical origin–target pairs from every SKU:

$$\mathcal{D} = \left\{ \left(i, t, \mathbf{x}_t^{(i)}, s_{t+h}^{(i)} \right) : i = 1, \ldots, N;\ t \in \mathcal{T}_i \right\}.$$

Global Training Dataset

Q: Where do the supervised examples come from?

- Combine valid historical origin–target pairs from every SKU:

$$\mathcal{D} = \left\{ \left(i, t, \mathbf{x}_t^{(i)}, s_{t+h}^{(i)} \right) : i = 1, \ldots, N;\ t \in \mathcal{T}_i \right\}.$$

- Each SKU contributes many rolling-origin examples; windows typically overlap.

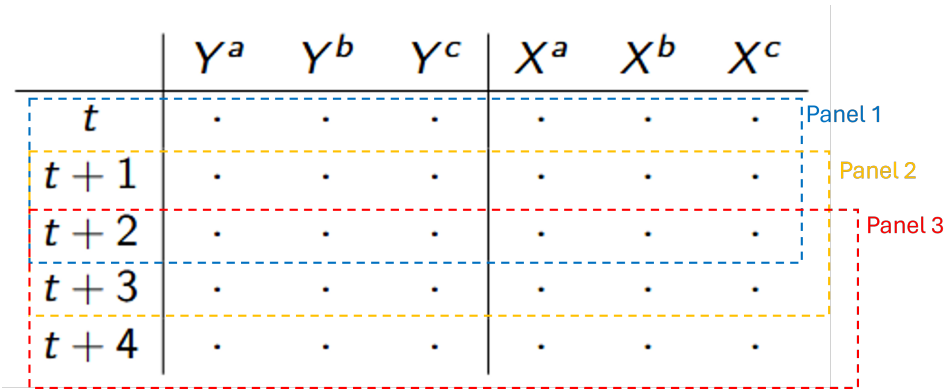
Global Training Dataset

Q: Where do the supervised examples come from?

- Combine valid historical origin–target pairs from every SKU:

$$\mathcal{D} = \left\{ \left(i, t, \mathbf{x}_t^{(i)}, s_{t+h}^{(i)} \right) : i = 1, \ldots, N;\ t \in \mathcal{T}_i \right\}.$$

- Each SKU contributes many rolling-origin examples; windows typically overlap.
- Sharing data across SKU histories is the *additional* source of information in global learning, not the existence of rolling windows alone.



Two Levels of Pooling

Q: Should we share information within divisions or across all divisions?

Two Levels of Pooling

Q: Should we share information within divisions or across all divisions?

- **Within-division:** train one global model for each division using its SKUs.

Two Levels of Pooling

Q: Should we share information within divisions or across all divisions?

- **Within-division:** train one global model for each division using its SKUs.
- **Across-division:** train one model for all SKUs across the company.

Two Levels of Pooling

Q: Should we share information within divisions or across all divisions?

- **Within-division:** train one global model for each division using its SKUs.
- **Across-division:** train one model for all SKUs across the company.
- A broader pool increases available examples but may also combine different sales dynamics and scales.

Two Levels of Pooling

Q: Should we share information within divisions or across all divisions?

- **Within-division:** train one global model for each division using its SKUs.
- **Across-division:** train one model for all SKUs across the company.
- A broader pool increases available examples but may also combine different sales dynamics and scales.
- Compare both using date-based, held-out forecast origins and metrics appropriate to the decision (e.g. SKU-level and division-level errors).
- Training only on some SKUs and evaluating on entirely unseen SKUs tests *cross-series transfer*; evaluating later months on previously seen SKUs tests *future forecasting*. These are different experiments.

Scale and Heterogeneity in Global Forecasting

Q: What if one SKU sells 10 units and another sells 10,000?

Scale and Heterogeneity in Global Forecasting

Q: What if one SKU sells 10 units and another sells 10,000?

- Unscaled squared-error training can be dominated by high-volume SKUs.

Scale and Heterogeneity in Global Forecasting

Q: What if one SKU sells 10 units and another sells 10,000?

- Unscaled squared-error training can be dominated by high-volume SKUs.
- Possible responses include training-only normalization, per-series scale factors, weighted loss, or scale-free evaluation metrics.

Scale and Heterogeneity in Global Forecasting

Q: What if one SKU sells 10 units and another sells 10,000?

- Unscaled squared-error training can be dominated by high-volume SKUs.
- Possible responses include training-only normalization, per-series scale factors, weighted loss, or scale-free evaluation metrics.
- Include SKU/division identifiers (or learned embeddings), calendar indicators, known prices, and other covariates when useful.

Scale and Heterogeneity in Global Forecasting

Q: What if one SKU sells 10 units and another sells 10,000?

- Unscaled squared-error training can be dominated by high-volume SKUs.
- Possible responses include training-only normalization, per-series scale factors, weighted loss, or scale-free evaluation metrics.
- Include SKU/division identifiers (or learned embeddings), calendar indicators, known prices, and other covariates when useful.
- Preserve the meaning of the target: we forecast *dollar sales* so that SKU values add to divisional and company revenue; modeling physical unit counts across unlike products poses a different aggregation problem.
- A global model learns shared patterns, but maintaining heterogeneity is a modeling design problem.

Introducing DeepAR

Q: Can we combine global pooling with learned recurrent memory?

Introducing DeepAR

Q: Can we combine global pooling with learned recurrent memory?

- DeepAR is a globally trained autoregressive recurrent network for **probabilistic** time-series forecasting.

Introducing DeepAR

Q: Can we combine global pooling with learned recurrent memory?

- DeepAR is a globally trained autoregressive recurrent network for **probabilistic** time-series forecasting.
- Every SKU has its own evolving hidden state, but *the recurrent weights and distribution head are shared*.

Introducing DeepAR

Q: Can we combine global pooling with learned recurrent memory?

- DeepAR is a globally trained autoregressive recurrent network for **probabilistic** time-series forecasting.
- Every SKU has its own evolving hidden state, but *the recurrent weights and distribution head are shared*.
- A basic representation is

$$\mathbf{h}_t^{(i)} = F_\theta\big(\mathbf{h}_{t-1}^{(i)}, s_{t-1}^{(i)}, \mathbf{z}_t^{(i)}\big), \quad \psi_t^{(i)} = G_\theta(\mathbf{h}_t^{(i)}).$$

Introducing DeepAR

Q: Can we combine global pooling with learned recurrent memory?

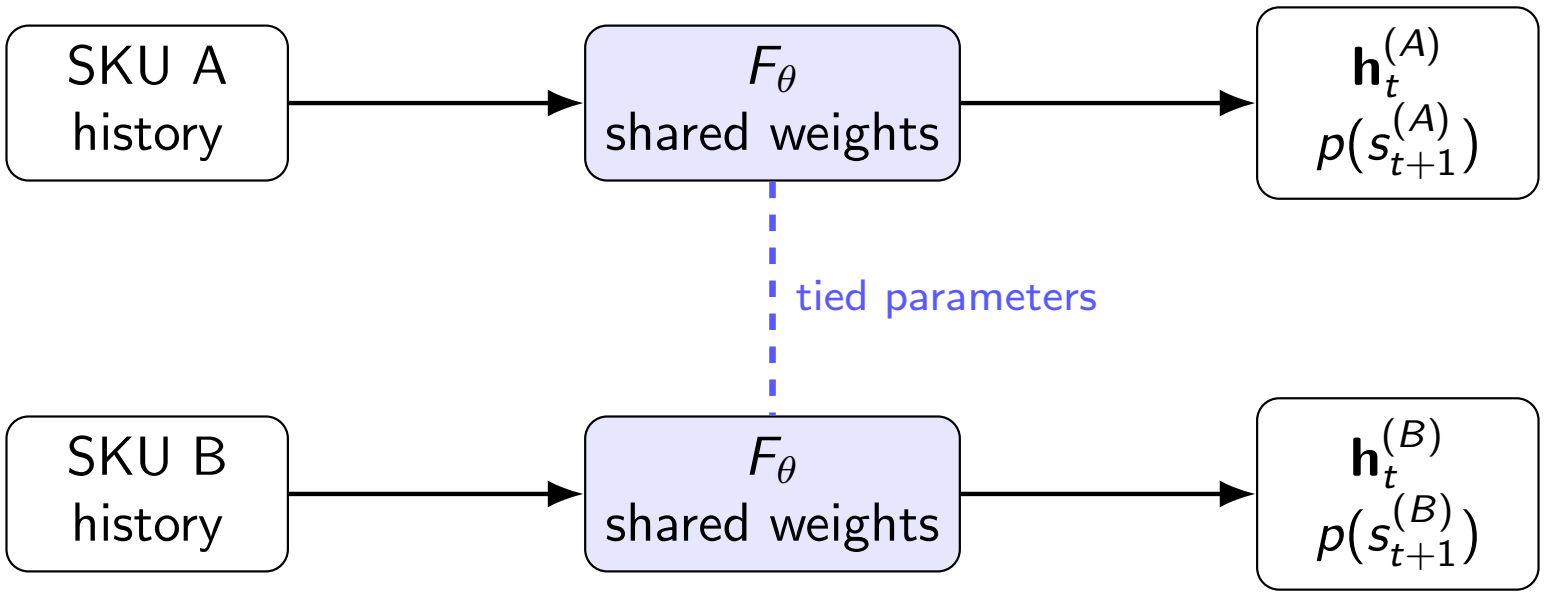
- DeepAR is a globally trained autoregressive recurrent network for **probabilistic** time-series forecasting.
- Every SKU has its own evolving hidden state, but *the recurrent weights and distribution head are shared*.
- A basic representation is

$$\mathbf{h}_t^{(i)} = F_\theta\big(\mathbf{h}_{t-1}^{(i)}, s_{t-1}^{(i)}, \mathbf{z}_t^{(i)}\big), \quad \psi_t^{(i)} = G_\theta(\mathbf{h}_t^{(i)}).$$

- $\psi_t^{(i)}$ parametrizes a predictive distribution $p(s_t^{(i)} \mid \psi_t^{(i)})$.
- This separates *shared parameters* (θ) from each SKU's *time-varying state* ($\mathbf{h}_t^{(i)}$).

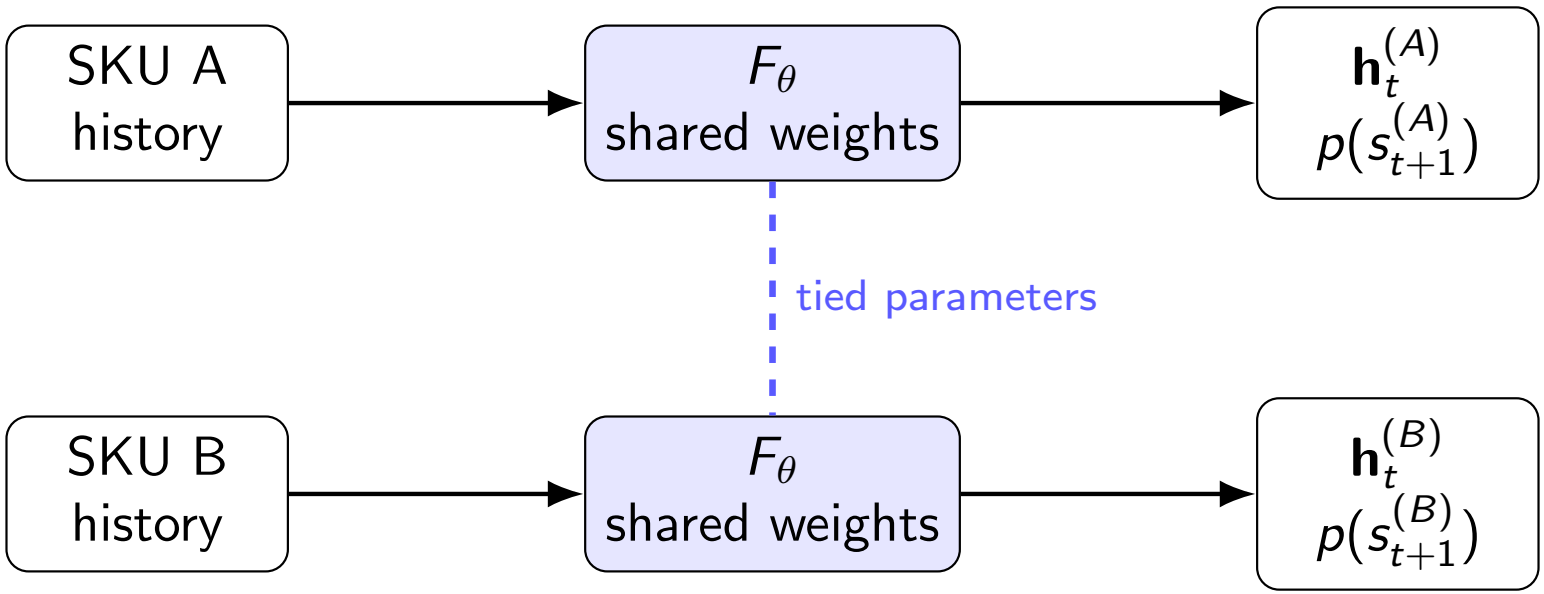
DeepAR: Shared Parameters, Different States

Q: How can the same network accommodate different SKU histories?



DeepAR: Shared Parameters, Different States

Q: How can the same network accommodate different SKU histories?



- The network parameters are pooled; SKU-specific states reflect their distinct observed histories.
- Covariates and scale normalization can allow additional differences across products.

DeepAR: A Probability Model

Q: What does the network predict instead of one sales number?

DeepAR: A Probability Model

Q: What does the network predict instead of one sales number?

- It outputs parameters of a chosen likelihood family, not an arbitrary distribution:

$$p(s_t^{(i)} \mid \mathcal{F}_{t-1}) = p\big(s_t^{(i)} \mid \psi_t^{(i)}\big).$$

DeepAR: A Probability Model

Q: What does the network predict instead of one sales number?

- It outputs parameters of a chosen likelihood family, not an arbitrary distribution:

$$p(s_t^{(i)} \mid \mathcal{F}_{t-1}) = p\big(s_t^{(i)} \mid \psi_t^{(i)}\big).$$

- For real-valued targets, a Gaussian head can predict $(\mu_t^{(i)}, \sigma_t^{(i)})$:

$$s_t^{(i)} \mid \mathcal{F}_{t-1} \sim N\big(\mu_t^{(i)}, (\sigma_t^{(i)})^2\big), \quad \sigma_t^{(i)} > 0.$$

DeepAR: A Probability Model

Q: What does the network predict instead of one sales number?

- It outputs parameters of a chosen likelihood family, not an arbitrary distribution:

$$p(s_t^{(i)} \mid \mathcal{F}_{t-1}) = p\big(s_t^{(i)} \mid \psi_t^{(i)}\big).$$

- For real-valued targets, a Gaussian head can predict $(\mu_t^{(i)}, \sigma_t^{(i)})$:

$$s_t^{(i)} \mid \mathcal{F}_{t-1} \sim N\big(\mu_t^{(i)}, (\sigma_t^{(i)})^2\big), \quad \sigma_t^{(i)} > 0.$$

- If instead modeling units sold as counts, a negative-binomial head can allow overdispersion.
- The likelihood must match the observed target, especially whether sales are counts, continuous revenue, intermittent, or bounded.

DeepAR: Estimation

Q: How do we fit shared parameters to many SKU sequences?

DeepAR: Estimation

Q: How do we fit shared parameters to many SKU sequences?

- Given a training set of SKU histories, maximize the joint conditional log likelihood:

$$\widehat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \log p\big(s_t^{(i)} \mid \psi_t^{(i)}(\theta)\big).$$

DeepAR: Estimation

Q: How do we fit shared parameters to many SKU sequences?

- Given a training set of SKU histories, maximize the joint conditional log likelihood:

$$\widehat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \log p\big(s_t^{(i)} \mid \psi_t^{(i)}(\theta)\big).$$

- During training, observed prior sales enter the recurrent update (often called *teacher forcing*).

DeepAR: Estimation

Q: How do we fit shared parameters to many SKU sequences?

- Given a training set of SKU histories, maximize the joint conditional log likelihood:

$$\widehat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \log p\big(s_t^{(i)} \mid \psi_t^{(i)}(\theta)\big).$$

- During training, observed prior sales enter the recurrent update (often called *teacher forcing*).
- During future simulation, observed values are unavailable, so sampled or predicted sales enter the next recurrent step.

DeepAR: Estimation

Q: How do we fit shared parameters to many SKU sequences?

- Given a training set of SKU histories, maximize the joint conditional log likelihood:

$$\widehat{\theta} = \arg\max_{\theta} \sum_{i=1}^{N} \sum_{t \in \mathcal{T}_i} \log p\big(s_t^{(i)} \mid \psi_t^{(i)}(\theta)\big).$$

- During training, observed prior sales enter the recurrent update (often called *teacher forcing*).
- During future simulation, observed values are unavailable, so sampled or predicted sales enter the next recurrent step.
- Model estimation may use sampled subsequences; the objective uses shared parameters while hidden states remain SKU-specific.

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

1. Process observed SKU history through month t to obtain its recurrent state.

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

1. Process observed SKU history through month t to obtain its recurrent state.
2. Generate a draw $s_{t+1}^{(b)} \sim p(\cdot \mid \psi_{t+1})$.

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

1. Process observed SKU history through month t to obtain its recurrent state.
2. Generate a draw $s_{t+1}^{(b)} \sim p(\cdot \mid \psi_{t+1})$.
3. Feed that draw and known future covariates back into the network; draw $s_{t+2}^{(b)}$, and continue to $t + H$.

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

1. Process observed SKU history through month t to obtain its recurrent state.
2. Generate a draw $s_{t+1}^{(b)} \sim p(\cdot \mid \psi_{t+1})$.
3. Feed that draw and known future covariates back into the network; draw $s_{t+2}^{(b)}$, and continue to $t + H$.
4. Repeat for paths $b = 1, \ldots, B$; use path distributions for quantiles, prediction intervals, and scenario summaries.

DeepAR: Simulating Multi-Month Paths

Q: How do we turn one-step conditional distributions into 12-month uncertainty?

1. Process observed SKU history through month t to obtain its recurrent state.
2. Generate a draw $s_{t+1}^{(b)} \sim p(\cdot \mid \psi_{t+1})$.
3. Feed that draw and known future covariates back into the network; draw $s_{t+2}^{(b)}$, and continue to $t + H$.
4. Repeat for paths $b = 1, \ldots, B$; use path distributions for quantiles, prediction intervals, and scenario summaries.

- This is autoregressive *forecast generation*; it does not mean an ordinary linear AR model.
- Forecast paths reflect the assumed likelihood and trained model, and can be miscalibrated when assumptions or dynamics are wrong.

What a Global Model Can and Cannot Do

Q: Is pooling automatically an improvement?

What a Global Model Can and Cannot Do

Q: Is pooling automatically an improvement?

- **Potential gains:** learn shared seasonality and responses; improve training where SKU histories are limited.

What a Global Model Can and Cannot Do

Q: Is pooling automatically an improvement?

- **Potential gains:** learn shared seasonality and responses; improve training where SKU histories are limited.
- **Potential costs:** negative transfer, scale imbalance, shared misspecification, and difficult-to-explain differences between products.

What a Global Model Can and Cannot Do

Q: Is pooling automatically an improvement?

- **Potential gains:** learn shared seasonality and responses; improve training where SKU histories are limited.
- **Potential costs:** negative transfer, scale imbalance, shared misspecification, and difficult-to-explain differences between products.
- A global model may use any architecture: pooled regression, boosted trees, N-BEATS, DeepAR, or transformers/TFT.

What a Global Model Can and Cannot Do

Q: Is pooling automatically an improvement?

- **Potential gains:** learn shared seasonality and responses; improve training where SKU histories are limited.
- **Potential costs:** negative transfer, scale imbalance, shared misspecification, and difficult-to-explain differences between products.
- A global model may use any architecture: pooled regression, boosted trees, N-BEATS, DeepAR, or transformers/TFT.
- Local/global describes *training scope*; it says nothing by itself about whether forecasts are coherent across the company hierarchy.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Hierarchical Time Series: The Problem

Q: What if our SKU forecasts disagree with the company's total-sales forecast?

Hierarchical Time Series: The Problem

Q: What if our SKU forecasts disagree with the company's total-sales forecast?

- Monthly dollar sales are additive across the company's hierarchy:

$$\text{Company} \longrightarrow \text{Division} \longrightarrow \text{SKU}.$$

Hierarchical Time Series: The Problem

Q: What if our SKU forecasts disagree with the company's total-sales forecast?

- Monthly dollar sales are additive across the company's hierarchy:

$$\text{Company} \longrightarrow \text{Division} \longrightarrow \text{SKU}.$$

- We may forecast each SKU, each division, and the company independently.

Hierarchical Time Series: The Problem

Q: What if our SKU forecasts disagree with the company's total-sales forecast?

- Monthly dollar sales are additive across the company's hierarchy:

$$\text{Company} \longrightarrow \text{Division} \longrightarrow \text{SKU}.$$

- We may forecast each SKU, each division, and the company independently.
- Independent *base forecasts* need not agree:

$$\widehat{s}^{\text{Company}}_{t+h|t} \neq \sum_{d} \widehat{s}^{\text{Division } d}_{t+h|t}.$$

Hierarchical Time Series: The Problem

Q: What if our SKU forecasts disagree with the company's total-sales forecast?

- Monthly dollar sales are additive across the company's hierarchy:

$$\text{Company} \longrightarrow \text{Division} \longrightarrow \text{SKU}.$$

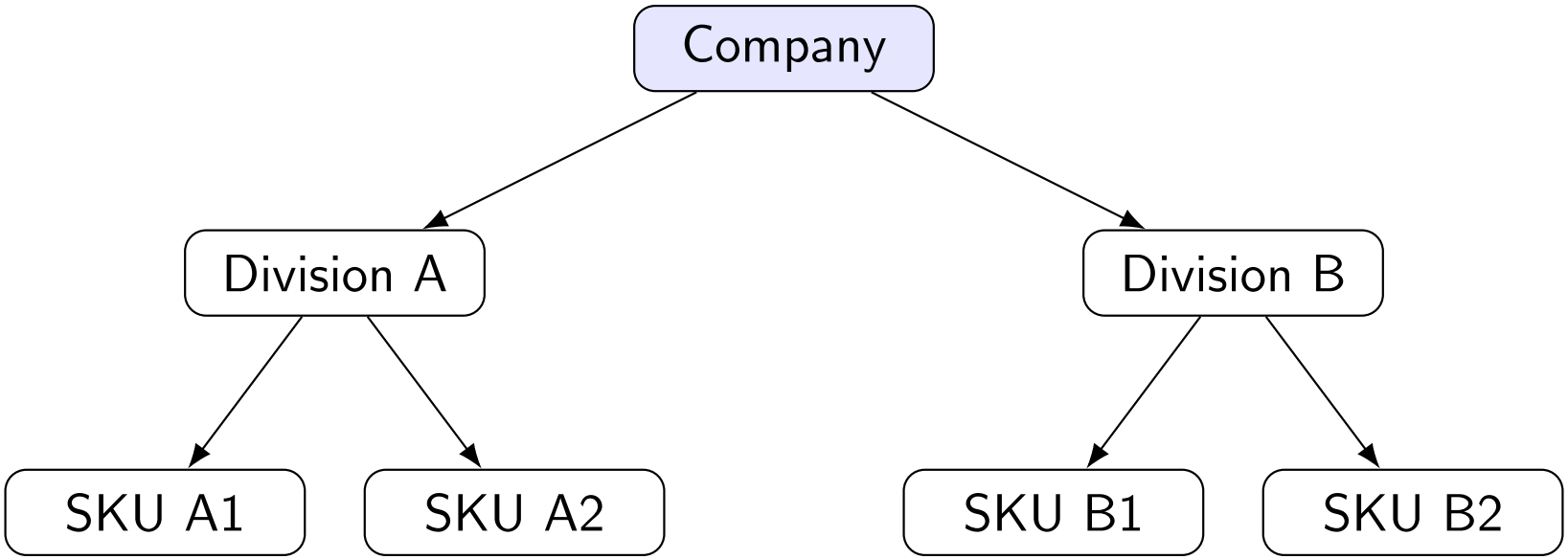
- We may forecast each SKU, each division, and the company independently.
- Independent *base forecasts* need not agree:

$$\widehat{s}^{\text{Company}}_{t+h|t} \neq \sum_{d} \widehat{s}^{\text{Division } d}_{t+h|t}.$$

- The issue is **coherence**: forecasts violate identities obeyed by the actual data.
- Even globally trained models do not generally enforce those identities automatically.

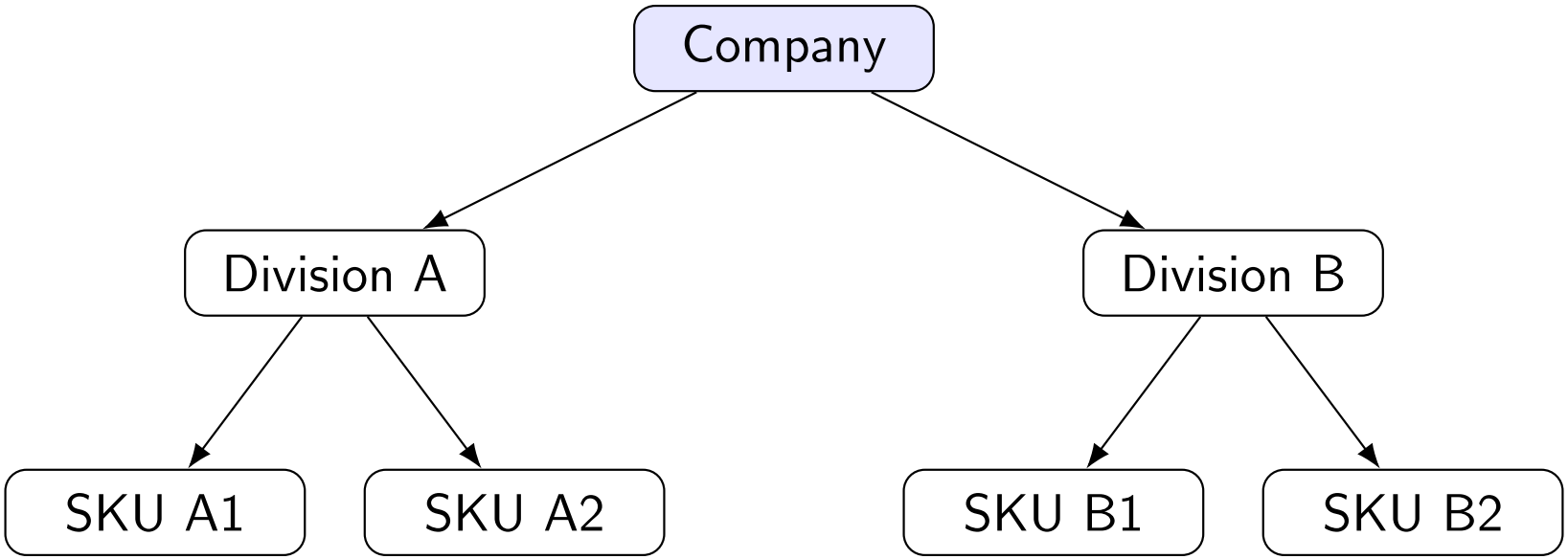
Hierarchy: One Company, Two Divisions, Four SKUs

Q: Which aggregation identities must forecasts respect?



Hierarchy: One Company, Two Divisions, Four SKUs

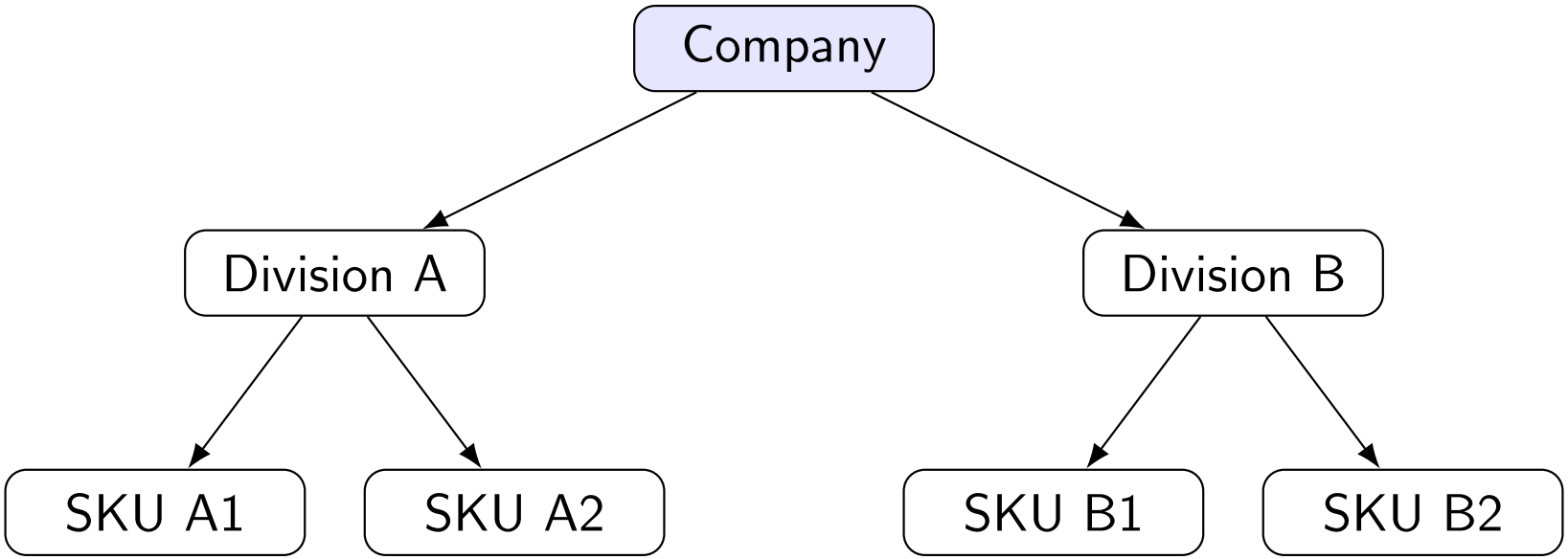
Q: Which aggregation identities must forecasts respect?



$$s^{\text{Company}} = s^{A} + s^{B}, \quad s^{A} = s^{A1} + s^{A2}, \quad s^{B} = s^{B1} + s^{B2}.$$

Hierarchy: One Company, Two Divisions, Four SKUs

Q: Which aggregation identities must forecasts respect?



$$s^{\text{Company}} = s^{A} + s^{B}, \quad s^{A} = s^{A1} + s^{A2}, \quad s^{B} = s^{B1} + s^{B2}.$$

- These are identities in *additive dollar sales*, not optional statistical relationships.

Three Basic Reconciliation Strategies

Q: Which level's base forecasts should we trust?

Three Basic Reconciliation Strategies

Q: Which level's base forecasts should we trust?

- **Bottom-up:** forecast SKU sales; sum to divisions and company.

Three Basic Reconciliation Strategies

Q: Which level's base forecasts should we trust?

- **Bottom-up:** forecast SKU sales; sum to divisions and company.
- **Top-down:** forecast the company total; allocate downward using proportions (often historical or forecast shares).

Three Basic Reconciliation Strategies

Q: Which level's base forecasts should we trust?

- **Bottom-up:** forecast SKU sales; sum to divisions and company.
- **Top-down:** forecast the company total; allocate downward using proportions (often historical or forecast shares).
- **Middle-out:** forecast divisions; sum up to company and allocate down to SKUs.

Three Basic Reconciliation Strategies

Q: Which level's base forecasts should we trust?

- **Bottom-up:** forecast SKU sales; sum to divisions and company.
- **Top-down:** forecast the company total; allocate downward using proportions (often historical or forecast shares).
- **Middle-out:** forecast divisions; sum up to company and allocate down to SKUs.
- Each enforces adding-up, but uses different information and imposes different assumptions.
- More general reconciliation can use base forecasts from *every* level rather than selecting one preferred level.

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

$$\widehat{s}^{A1} = 60, \qquad \widehat{s}^{A2} = 50, \qquad \widehat{s}^{A} = 120.$$

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

$$\widehat{s}^{A1} = 60, \qquad \widehat{s}^{A2} = 50, \qquad \widehat{s}^{A} = 120.$$

- **Conflict:** $60 + 50 = 110 \neq 120$.

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

$$\widehat{s}^{A1} = 60, \qquad \widehat{s}^{A2} = 50, \qquad \widehat{s}^{A} = 120.$$

- **Conflict:** $60 + 50 = 110 \neq 120$.
- **Bottom-up:** retain SKU forecasts and set $\widetilde{s}^{A} = 110$.

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

$$\widehat{s}^{A1} = 60, \qquad \widehat{s}^{A2} = 50, \qquad \widehat{s}^{A} = 120.$$

- **Conflict:** $60 + 50 = 110 \neq 120$.
- **Bottom-up:** retain SKU forecasts and set $\widetilde{s}^{A} = 110$.
- **Top-down:** retain the 120 division forecast. For shares 55% and 45%:

$$\widetilde{s}^{A1} = 0.55(120) = 66, \quad \widetilde{s}^{A2} = 0.45(120) = 54.$$

Bottom-Up and Top-Down: A Numerical Example

Q: Suppose two SKUs are forecast at 60 and 50, but their division at 120.

$$\widehat{s}^{A1} = 60, \qquad \widehat{s}^{A2} = 50, \qquad \widehat{s}^{A} = 120.$$

- **Conflict:** $60 + 50 = 110 \neq 120$.
- **Bottom-up:** retain SKU forecasts and set $\widetilde{s}^{A} = 110$.
- **Top-down:** retain the 120 division forecast. For shares 55% and 45%:

$$\widetilde{s}^{A1} = 0.55(120) = 66, \quad \widetilde{s}^{A2} = 0.45(120) = 54.$$

- Historical allocations can be poor when SKU mix changes, products are launched, or demand shifts.

Middle-Out: A Numerical Example

Q: What if divisional forecasts are more reliable than SKU forecasts?

Middle-Out: A Numerical Example

Q: What if divisional forecasts are more reliable than SKU forecasts?

- Trust division forecasts: $\widehat{s}^A = 120$, $\widehat{s}^B = 80$.

Middle-Out: A Numerical Example

Q: What if divisional forecasts are more reliable than SKU forecasts?

- Trust division forecasts: $\widehat{s}^{A} = 120$, $\widehat{s}^{B} = 80$.
- Aggregate upward: $\widetilde{s}^{\text{Company}} = 120 + 80 = 200$.

Middle-Out: A Numerical Example

Q: What if divisional forecasts are more reliable than SKU forecasts?

- Trust division forecasts: $\widehat{s}^{A} = 120$, $\widehat{s}^{B} = 80$.
- Aggregate upward: $\widetilde{s}^{\text{Company}} = 120 + 80 = 200$.
- Allocate downward using division-specific shares:

	share	reconciled SKU sales
$A1$	55% of A	66
$A2$	45% of A	54
$B1$	75% of B	60
$B2$	25% of B	20

Middle-Out: A Numerical Example

Q: What if divisional forecasts are more reliable than SKU forecasts?

- Trust division forecasts: $\widehat{s}^{A} = 120$, $\widehat{s}^{B} = 80$.
- Aggregate upward: $\widetilde{s}^{\text{Company}} = 120 + 80 = 200$.
- Allocate downward using division-specific shares:

	share	reconciled SKU sales
$A1$	55% of A	66
$A2$	45% of A	54
$B1$	75% of B	60
$B2$	25% of B	20

- Check: $66 + 54 + 60 + 20 = 200$. Every level is coherent.

Aggregation as a Matrix

Q: How can we express every adding-up constraint at once?

Aggregation as a Matrix

Q: How can we express every adding-up constraint at once?

- Collect bottom-level sales in $\mathbf{b} = (s^{A1}, s^{A2}, s^{B1}, s^{B2})'$.
- With ordering (Company, A, B, A1, A2, B1, B2),

$$\mathbf{y} = S\mathbf{b}, \qquad S = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Aggregation as a Matrix

Q: How can we express every adding-up constraint at once?

- Collect bottom-level sales in $\mathbf{b} = (s^{A1}, s^{A2}, s^{B1}, s^{B2})'$.
- With ordering (Company, A, B, A1, A2, B1, B2),

$$\mathbf{y} = S\mathbf{b}, \qquad S = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- S is a fixed summing matrix. Any vector of the form $S\mathbf{b}$ is coherent.

General Forecast Reconciliation

Q: What if base forecasts are available for all seven series?

General Forecast Reconciliation

Q: What if base forecasts are available for all seven series?

- Let $\widehat{\mathbf{y}}_{t+h|t}$ be the vector of *base forecasts at every hierarchy level*, not just SKUs.

General Forecast Reconciliation

Q: What if base forecasts are available for all seven series?

- Let $\widehat{\mathbf{y}}_{t+h|t}$ be the vector of *base forecasts at every hierarchy level*, not just SKUs.
- First map these base forecasts to adjusted bottom-level forecasts via G_h, then aggregate:

$$\boxed{\widetilde{\mathbf{y}}_{t+h|t} = S\, G_h\, \widehat{\mathbf{y}}_{t+h|t}.}$$

General Forecast Reconciliation

Q: What if base forecasts are available for all seven series?

- Let $\widehat{\mathbf{y}}_{t+h|t}$ be the vector of *base forecasts at every hierarchy level*, not just SKUs.
- First map these base forecasts to adjusted bottom-level forecasts via G_h, then aggregate:

$$\boxed{\widetilde{\mathbf{y}}_{t+h|t} = S\, G_h\, \widehat{\mathbf{y}}_{t+h|t}.}$$

- G_h is a 4×7 reconciliation map for our example; S is 7×4.
- If only four bottom-level forecasts are supplied and retained, the familiar special case is $\widetilde{\mathbf{y}} = S\widehat{\mathbf{b}}$.

General Forecast Reconciliation

Q: What if base forecasts are available for all seven series?

- Let $\widehat{\mathbf{y}}_{t+h|t}$ be the vector of *base forecasts at every hierarchy level*, not just SKUs.
- First map these base forecasts to adjusted bottom-level forecasts via G_h, then aggregate:

$$\boxed{\widetilde{\mathbf{y}}_{t+h|t} = S\, G_h\, \widehat{\mathbf{y}}_{t+h|t}.}$$

- G_h is a 4×7 reconciliation map for our example; S is 7×4.
- If only four bottom-level forecasts are supplied and retained, the familiar special case is $\widetilde{\mathbf{y}} = S\widehat{\mathbf{b}}$.
- Different reconciliation methods correspond to different choices of G_h.

Why Consider Minimum-Trace (MinT) Reconciliation?

Q: Could we combine evidence from all levels rather than choose only one?

Why Consider Minimum-Trace (MinT) Reconciliation?

Q: Could we combine evidence from all levels rather than choose only one?

- SKU forecasts may be noisy while aggregate forecasts are stable, but aggregates may miss changing product mix.

Why Consider Minimum-Trace (MinT) Reconciliation?

Q: Could we combine evidence from all levels rather than choose only one?

- SKU forecasts may be noisy while aggregate forecasts are stable, but aggregates may miss changing product mix.
- Historical base-forecast errors reveal which series are more uncertain and how their errors co-move.

Why Consider Minimum-Trace (MinT) Reconciliation?

Q: Could we combine evidence from all levels rather than choose only one?

- SKU forecasts may be noisy while aggregate forecasts are stable, but aggregates may miss changing product mix.
- Historical base-forecast errors reveal which series are more uncertain and how their errors co-move.
- **MinT** chooses a coherent linear reconciliation that minimizes the sum of reconciled forecast-error variances, under an unbiased-base-forecast framework.
- It uses a covariance matrix rather than assuming every level is equally reliable.

MinT: Mathematical Formulation

Q: What weight matrix reconciles forecast errors efficiently?

MinT: Mathematical Formulation

Q: What weight matrix reconciles forecast errors efficiently?

- Let $W_h = \text{Var}(\mathbf{y}_{t+h} - \widehat{\mathbf{y}}_{t+h|t})$ be the covariance of h-step base-forecast errors.

MinT: Mathematical Formulation

Q: What weight matrix reconciles forecast errors efficiently?

- Let $W_h = \mathrm{Var}(\mathbf{y}_{t+h} - \widehat{\mathbf{y}}_{t+h|t})$ be the covariance of h-step base-forecast errors.
- Under the usual unbiasedness/coherence constraints and invertible W_h,

$$G_h^{\mathrm{MinT}} = \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1}.$$

MinT: Mathematical Formulation

Q: What weight matrix reconciles forecast errors efficiently?

- Let $W_h = \mathrm{Var}(\mathbf{y}_{t+h} - \widehat{\mathbf{y}}_{t+h|t})$ be the covariance of h-step base-forecast errors.
- Under the usual unbiasedness/coherence constraints and invertible W_h,

$$G_h^{\mathrm{MinT}} = \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1}.$$

- Reconciled forecasts are

$$\boxed{\widetilde{\mathbf{y}}_{t+h|t} = S \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1} \widehat{\mathbf{y}}_{t+h|t}.}$$

MinT: Mathematical Formulation

Q: What weight matrix reconciles forecast errors efficiently?

- Let $W_h = \mathrm{Var}(\mathbf{y}_{t+h} - \widehat{\mathbf{y}}_{t+h|t})$ be the covariance of h-step base-forecast errors.
- Under the usual unbiasedness/coherence constraints and invertible W_h,

$$G_h^{\mathrm{MinT}} = \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1}.$$

- Reconciled forecasts are

$$\boxed{\widetilde{\mathbf{y}}_{t+h|t} = S \left(S' W_h^{-1} S\right)^{-1} S' W_h^{-1} \widehat{\mathbf{y}}_{t+h|t}.}$$

- Practical issues: estimate W_h from legitimate forecast errors; use shrinkage/approximations if the covariance estimate is noisy or singular.
- Linear coherence does not automatically guarantee nonnegative forecasts or coherent predictive *distributions*.

Hierarchical Forecasting Is Model-Agnostic

Q: Must we choose between global models and reconciliation?

Hierarchical Forecasting Is Model-Agnostic

Q: Must we choose between global models and reconciliation?

- **No.** A base model may be ARIMA, boosted trees, DeepAR, a global transformer, or a foundation model.

Hierarchical Forecasting Is Model-Agnostic

Q: Must we choose between global models and reconciliation?

- **No.** A base model may be ARIMA, boosted trees, DeepAR, a global transformer, or a foundation model.
- Reconciliation concerns *constraints across series*; local/global concerns *how the model is trained*.

Hierarchical Forecasting Is Model-Agnostic

Q: Must we choose between global models and reconciliation?

- **No.** A base model may be ARIMA, boosted trees, DeepAR, a global transformer, or a foundation model.
- Reconciliation concerns *constraints across series*; local/global concerns *how the model is trained*.
- For the same company, we might (i) train global SKU models, (ii) independently forecast divisions and company totals, and (iii) reconcile all outputs.

Hierarchical Forecasting Is Model-Agnostic

Q: Must we choose between global models and reconciliation?

- **No.** A base model may be ARIMA, boosted trees, DeepAR, a global transformer, or a foundation model.
- Reconciliation concerns *constraints across series*; local/global concerns *how the model is trained*.
- For the same company, we might (i) train global SKU models, (ii) independently forecast divisions and company totals, and (iii) reconcile all outputs.
- Forecast accuracy and coherence should both be evaluated; coherence alone does not imply better forecast accuracy.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

From Global Models to Foundation Models

Q: What if our company does not have enough data to train a rich model?

From Global Models to Foundation Models

Q: What if our company does not have enough data to train a rich model?

- A global model pools many SKUs within our company, then estimates θ from those histories.

From Global Models to Foundation Models

Q: What if our company does not have enough data to train a rich model?

- A global model pools many SKUs within our company, then estimates θ from those histories.
- A foundation model is **pretrained** on a much larger, varied collection of time-series tasks and datasets; we can reuse its parameters on new series.

From Global Models to Foundation Models

Q: What if our company does not have enough data to train a rich model?

- A global model pools many SKUs within our company, then estimates θ from those histories.
- A foundation model is **pretrained** on a much larger, varied collection of time-series tasks and datasets; we can reuse its parameters on new series.
- It may capture broad patterns such as seasonality, persistence, changes in scale, and recurring temporal shapes.

From Global Models to Foundation Models

Q: What if our company does not have enough data to train a rich model?

- A global model pools many SKUs within our company, then estimates θ from those histories.
- A foundation model is **pretrained** on a much larger, varied collection of time-series tasks and datasets; we can reuse its parameters on new series.
- It may capture broad patterns such as seasonality, persistence, changes in scale, and recurring temporal shapes.
- Pretraining does not guarantee transfer: unfamiliar products, intermittent sales, structural breaks, and specialized covariates may still be challenging.

What Makes a Forecasting Model a Foundation Model?

Q: Is a foundation model simply a very large global model?

What Makes a Forecasting Model a Foundation Model?

Q: Is a foundation model simply a very large global model?

- There is conceptual continuity: both learn parameters from multiple time series.

What Makes a Forecasting Model a Foundation Model?

Q: Is a foundation model simply a very large global model?

- There is conceptual continuity: both learn parameters from multiple time series.
- The distinctive deployment idea is **pretrain then reuse** across tasks or datasets, rather than necessarily fit a new model for each dataset.

What Makes a Forecasting Model a Foundation Model?

Q: Is a foundation model simply a very large global model?

- There is conceptual continuity: both learn parameters from multiple time series.
- The distinctive deployment idea is **pretrain then reuse** across tasks or datasets, rather than necessarily fit a new model for each dataset.
- **Zero-shot:** forecast a new SKU without updating pretrained weights.
- **Fine-tuning:** adapt some or all weights using eligible company histories.

What Makes a Forecasting Model a Foundation Model?

Q: Is a foundation model simply a very large global model?

- There is conceptual continuity: both learn parameters from multiple time series.
- The distinctive deployment idea is **pretrain then reuse** across tasks or datasets, rather than necessarily fit a new model for each dataset.
- **Zero-shot:** forecast a new SKU without updating pretrained weights.
- **Fine-tuning:** adapt some or all weights using eligible company histories.
- Pretrained models differ in their input representations, architectures, forecast outputs, and covariate support.

Time-Series Representations for Pretraining

Q: How can a transformer consume numerical observations?

Time-Series Representations for Pretraining

Q: How can a transformer consume numerical observations?

- **Discrete tokenization:** rescale numbers and quantize them into a finite vocabulary (original Chronos).

Time-Series Representations for Pretraining

Q: How can a transformer consume numerical observations?

- **Discrete tokenization:** rescale numbers and quantize them into a finite vocabulary (original Chronos).
- **Continuous patches:** group several adjacent numeric observations and embed the patch as a vector (e.g. original TimesFM).

Time-Series Representations for Pretraining

Q: How can a transformer consume numerical observations?

- **Discrete tokenization:** rescale numbers and quantize them into a finite vocabulary (original Chronos).
- **Continuous patches:** group several adjacent numeric observations and embed the patch as a vector (e.g. original TimesFM).
- Both feed *positioned representations* into a transformer; tokenization into discrete bins is *not* universal.

Time-Series Representations for Pretraining

Q: How can a transformer consume numerical observations?

- **Discrete tokenization:** rescale numbers and quantize them into a finite vocabulary (original Chronos).
- **Continuous patches:** group several adjacent numeric observations and embed the patch as a vector (e.g. original TimesFM).
- Both feed *positioned representations* into a transformer; tokenization into discrete bins is *not* universal.
- We will develop **original Chronos** carefully as one example, then contrast alternative designs.

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

1. Select historical context, e.g. the previous 36 monthly sales observations.

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

1. Select historical context, e.g. the previous 36 monthly sales observations.
2. Scale the context and quantize each scaled value into a discrete token.

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

1. Select historical context, e.g. the previous 36 monthly sales observations.
2. Scale the context and quantize each scaled value into a discrete token.
3. Use a transformer-based model (original Chronos includes T5 encoder–decoder variants) to learn conditional probabilities for future tokens.

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

1. Select historical context, e.g. the previous 36 monthly sales observations.
2. Scale the context and quantize each scaled value into a discrete token.
3. Use a transformer-based model (original Chronos includes T5 encoder–decoder variants) to learn conditional probabilities for future tokens.
4. Generate multiple plausible future-token sequences, map them back to numerical values, and undo the scale transformation.

Original Chronos: Training and Forecasting Pipeline

Q: How do continuous sales numbers become a forecasting task for a language-model architecture?

1. Select historical context, e.g. the previous 36 monthly sales observations.
2. Scale the context and quantize each scaled value into a discrete token.
3. Use a transformer-based model (original Chronos includes T5 encoder–decoder variants) to learn conditional probabilities for future tokens.
4. Generate multiple plausible future-token sequences, map them back to numerical values, and undo the scale transformation.
5. Summarize simulated paths as means, medians, quantiles, and prediction intervals.

Original Chronos: Scale Normalization

Q: How can one model handle SKUs with widely different monthly sales levels?

Original Chronos: Scale Normalization

Q: How can one model handle SKUs with widely different monthly sales levels?

- Let the observed context be $s_{t-L+1}, \ldots, s_t$.

Original Chronos: Scale Normalization

Q: How can one model handle SKUs with widely different monthly sales levels?

- Let the observed context be $s_{t-L+1}, \ldots, s_t$.
- Compute a context-based positive scale, for example

$$a_t = \frac{1}{L} \sum_{j=0}^{L-1} |s_{t-j}|, \quad z_\tau = \frac{s_\tau}{a_t}, \qquad a_t > 0.$$

Original Chronos: Scale Normalization

Q: How can one model handle SKUs with widely different monthly sales levels?

- Let the observed context be $s_{t-L+1}, \ldots, s_t$.
- Compute a context-based positive scale, for example

$$a_t = \frac{1}{L} \sum_{j=0}^{L-1} |s_{t-j}|, \quad z_\tau = \frac{s_\tau}{a_t}, \qquad a_t > 0.$$

- Scaling makes values across disparate series more comparable before quantization.
- Handle all-zero histories using an explicit safeguard; do not divide by zero.

Original Chronos: Scale Normalization

Q: How can one model handle SKUs with widely different monthly sales levels?

- Let the observed context be $s_{t-L+1}, \ldots, s_t$.
- Compute a context-based positive scale, for example

$$a_t = \frac{1}{L} \sum_{j=0}^{L-1} |s_{t-j}|, \quad z_\tau = \frac{s_\tau}{a_t}, \qquad a_t > 0.$$

- Scaling makes values across disparate series more comparable before quantization.
- Handle all-zero histories using an explicit safeguard; do not divide by zero.
- The scale must depend only on the historical context at the forecast origin; including future target values would leak information.

Original Chronos: Quantization into Tokens

Q: How does a continuous scaled value become a discrete symbol?

Original Chronos: Quantization into Tokens

Q: How does a continuous scaled value become a discrete symbol?

- Partition a finite range of scaled values into K bins with representative centers $m_1, \ldots, m_K$.

Original Chronos: Quantization into Tokens

Q: How does a continuous scaled value become a discrete symbol?

- Partition a finite range of scaled values into K bins with representative centers $m_1, \ldots, m_K$.
- For a scaled sales value z_τ, define a token

$$q_\tau = Q(z_\tau) \in \{1, \ldots, K\}.$$

Original Chronos: Quantization into Tokens

Q: How does a continuous scaled value become a discrete symbol?

- Partition a finite range of scaled values into K bins with representative centers $m_1, \ldots, m_K$.
- For a scaled sales value z_τ, define a token

$$q_\tau = Q(z_\tau) \in \{1, \ldots, K\}.$$

- Illustration: 0.8 maps to one bin, 1.0 to another, and 1.2 to another; the exact bins are a model design choice.

Original Chronos: Quantization into Tokens

Q: How does a continuous scaled value become a discrete symbol?

- Partition a finite range of scaled values into K bins with representative centers $m_1, \ldots, m_K$.
- For a scaled sales value z_τ, define a token

$$q_\tau = Q(z_\tau) \in \{1, \ldots, K\}.$$

- Illustration: 0.8 maps to one bin, 1.0 to another, and 1.2 to another; the exact bins are a model design choice.
- Quantization loses precision and may clip extreme values, but permits a categorical prediction objective.
- Original Chronos adds special tokens where needed; the toy three-bin example is only intuition, not the actual vocabulary.

Original Chronos: Why a Transformer?

Q: What does the pretrained model learn from these tokens?

Original Chronos: Why a Transformer?

Q: What does the pretrained model learn from these tokens?

- The context tokens provide a representation of the recent sales history.

Original Chronos: Why a Transformer?

Q: What does the pretrained model learn from these tokens?

- The context tokens provide a representation of the recent sales history.
- The model learns a conditional distribution of future tokens given this context:

$$p_\theta(q_{t+1:t+H} \mid q_{t-L+1:t}).$$

Original Chronos: Why a Transformer?

Q: What does the pretrained model learn from these tokens?

- The context tokens provide a representation of the recent sales history.
- The model learns a conditional distribution of future tokens given this context:

$$p_\theta(q_{t+1:t+H} \mid q_{t-L+1:t}).$$

- In the autoregressive sequence-generation formulation, use

$$p_\theta(q_{t+1:t+H} \mid \text{context}) = \prod_{j=1}^{H} p_\theta(q_{t+j} \mid q_{t+1:t+j-1}, \text{context}).$$

Original Chronos: Why a Transformer?

Q: What does the pretrained model learn from these tokens?

- The context tokens provide a representation of the recent sales history.
- The model learns a conditional distribution of future tokens given this context:

$$p_\theta(q_{t+1:t+H} \mid q_{t-L+1:t}).$$

- In the autoregressive sequence-generation formulation, use

$$p_\theta(q_{t+1:t+H} \mid \text{context}) = \prod_{j=1}^{H} p_\theta(q_{t+j} \mid q_{t+1:t+j-1}, \text{context}).$$

- The transformer represents contextual relationships; it does not literally identify the structural data-generating process.

Original Chronos: Training Objective

Q: How are probabilities over future tokens learned?

Original Chronos: Training Objective

Q: How are probabilities over future tokens learned?

- Build many historical context–future pairs from many datasets.

Original Chronos: Training Objective

Q: How are probabilities over future tokens learned?

- Build many historical context–future pairs from many datasets.
- Estimate transformer weights by cross-entropy, i.e. negative conditional log likelihood of the observed future tokens:

$$\widehat{\theta} = \arg\min_{\theta} \left[- \sum_{(c,f)\in\mathcal{D}} \sum_{j=1}^{H} \log p_\theta(q_{t+j}^{(f)} \mid q_{t+1:t+j-1}^{(f)}, q_{t-L+1:t}^{(c)}) \right].$$

Original Chronos: Training Objective

Q: How are probabilities over future tokens learned?

- Build many historical context–future pairs from many datasets.
- Estimate transformer weights by cross-entropy, i.e. negative conditional log likelihood of the observed future tokens:

$$\widehat{\theta} = \arg\min_{\theta} \left[- \sum_{(c,f)\in\mathcal{D}} \sum_{j=1}^{H} \log p_\theta(q_{t+j}^{(f)} \mid q_{t+1:t+j-1}^{(f)}, q_{t-L+1:t}^{(c)}) \right].$$

- During training, known target tokens can condition later decoding steps; at inference they are replaced by generated tokens.
- This objective is *specific to the discretized generative example*; other foundation models may use continuous forecasts or quantile losses.

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

1. Condition on the 36-month observed context and sample future tokens $q_{t+1}^{(b)}, \ldots, q_{t+12}^{(b)}$.

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

1. Condition on the 36-month observed context and sample future tokens $q_{t+1}^{(b)}, \ldots, q_{t+12}^{(b)}$.
2. Map each token to its bin representative: $\widetilde{z}_{t+j}^{(b)} = m_{q_{t+j}^{(b)}}$.

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

1. Condition on the 36-month observed context and sample future tokens $q_{t+1}^{(b)}, \ldots, q_{t+12}^{(b)}$.
2. Map each token to its bin representative: $\widetilde{z}_{t+j}^{(b)} = m_{q_{t+j}^{(b)}}$.
3. Undo scaling: $\widetilde{s}_{t+j}^{(b)} = a_t \widetilde{z}_{t+j}^{(b)}$.

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

1. Condition on the 36-month observed context and sample future tokens $q_{t+1}^{(b)}, \ldots, q_{t+12}^{(b)}$.
2. Map each token to its bin representative: $\tilde{z}_{t+j}^{(b)} = m_{q_{t+j}^{(b)}}$.
3. Undo scaling: $\tilde{s}_{t+j}^{(b)} = a_t \tilde{z}_{t+j}^{(b)}$.
4. Repeat over sampled paths $b = 1, \ldots, B$; calculate empirical quantiles or point summaries for each future month.

Original Chronos: From Tokens Back to Sales

Q: How is a probabilistic 12-month forecast produced?

1. Condition on the 36-month observed context and sample future tokens $q_{t+1}^{(b)}, \ldots, q_{t+12}^{(b)}$.
2. Map each token to its bin representative: $\tilde{z}_{t+j}^{(b)} = m_{q_{t+j}^{(b)}}$.
3. Undo scaling: $\tilde{s}_{t+j}^{(b)} = a_t \tilde{z}_{t+j}^{(b)}$.
4. Repeat over sampled paths $b = 1, \ldots, B$; calculate empirical quantiles or point summaries for each future month.

- Sampling introduces Monte Carlo variation; increasing the sample count reduces that numerical variation but not model misspecification.

Using Chronos for Our Company's SKUs

Q: What changes compared with training DeepAR on our own sales histories?

Using Chronos for Our Company's SKUs

Q: What changes compared with training DeepAR on our own sales histories?

- **DeepAR:** estimate shared recurrent weights on the company's SKU histories.

Using Chronos for Our Company's SKUs

Q: What changes compared with training DeepAR on our own sales histories?

- **DeepAR:** estimate shared recurrent weights on the company's SKU histories.
- **Zero-shot Chronos:** supply each SKU's observed historical context to pretrained weights and generate forecasts without training on company data.

Using Chronos for Our Company's SKUs

Q: What changes compared with training DeepAR on our own sales histories?

- **DeepAR:** estimate shared recurrent weights on the company's SKU histories.
- **Zero-shot Chronos:** supply each SKU's observed historical context to pretrained weights and generate forecasts without training on company data.
- Both can forecast many SKUs; their *training provenance* is different.

Using Chronos for Our Company's SKUs

Q: What changes compared with training DeepAR on our own sales histories?

- **DeepAR:** estimate shared recurrent weights on the company's SKU histories.
- **Zero-shot Chronos:** supply each SKU's observed historical context to pretrained weights and generate forecasts without training on company data.
- Both can forecast many SKUs; their *training provenance* is different.
- Evaluate on identical forecast origins, horizons, and targets; include seasonal-naive and other simple baselines.
- The model's default representation of time series may omit promotion plans or other company-specific features; check the particular model's covariate support.

Alternative Foundation Model Architectures

Q: Do all pretrained time-series models use Chronos-style quantization?

Alternative Foundation Model Architectures

Q: Do all pretrained time-series models use Chronos-style quantization?

Representative	**Distinguishing design idea**
Original Chronos	Scale and quantize observations; transformer predicts categorical future tokens.
Chronos-Bolt	Patch-based encoder with direct multi-step quantile forecasts; not the original token-by-token generation pipeline.
Original TimesFM	Continuous-valued patches processed with a decoder-only transformer.
Moirai	Pretrained probabilistic forecasting with masked/patch-based representations and flexible series/frequency handling.

Alternative Foundation Model Architectures

Q: Do all pretrained time-series models use Chronos-style quantization?

Representative	Distinguishing design idea
Original Chronos	Scale and quantize observations; transformer predicts categorical future tokens.
Chronos-Bolt	Patch-based encoder with direct multi-step quantile forecasts; not the original token-by-token generation pipeline.
Original TimesFM	Continuous-valued patches processed with a decoder-only transformer.
Moirai	Pretrained probabilistic forecasting with masked/patch-based representations and flexible series/frequency handling.

These are *illustrative architecture families*; names, versions, and available input modalities evolve.

Foundation Model Adaptation

Q: What if zero-shot forecasts are not good enough for our SKU data?

Foundation Model Adaptation

Q: What if zero-shot forecasts are not good enough for our SKU data?

- **Zero-shot:** evaluate pretrained weights θ_0 without updating them.

Foundation Model Adaptation

Q: What if zero-shot forecasts are not good enough for our SKU data?

- **Zero-shot:** evaluate pretrained weights θ_0 without updating them.
- **Fine-tuning:** initialize at θ_0 and minimize a model-appropriate forecasting loss on historical company data:

$$\widehat{\theta} = \arg\min_{\theta} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \ell(y, f_\theta(\mathbf{x})).$$

Foundation Model Adaptation

Q: What if zero-shot forecasts are not good enough for our SKU data?

- **Zero-shot:** evaluate pretrained weights θ_0 without updating them.
- **Fine-tuning:** initialize at θ_0 and minimize a model-appropriate forecasting loss on historical company data:

$$\widehat{\theta} = \arg\min_{\theta} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \ell(y, f_\theta(\mathbf{x})).$$

- Options include adapting all weights, selected layers, or lightweight adapters where supported.

Foundation Model Adaptation

Q: What if zero-shot forecasts are not good enough for our SKU data?

- **Zero-shot:** evaluate pretrained weights θ_0 without updating them.
- **Fine-tuning:** initialize at θ_0 and minimize a model-appropriate forecasting loss on historical company data:

$$\widehat{\theta} = \arg\min_{\theta} \sum_{(\mathbf{x},y)\in\mathcal{D}_{\text{train}}} \ell(y, f_\theta(\mathbf{x})).$$

- Options include adapting all weights, selected layers, or lightweight adapters where supported.
- Some models permit exogenous covariates, while others require separate mechanisms; check model-specific interfaces.
- Any fine-tuning and hyperparameter selection must use data preceding the held-out forecast origins.

Zero-Shot Forecasting: Evaluation Pitfalls

Q: Does strong performance elsewhere guarantee strong sales forecasts here?

Zero-Shot Forecasting: Evaluation Pitfalls

Q: Does strong performance elsewhere guarantee strong sales forecasts here?

- No: pretraining data and our demand process may differ (domain shift).

Zero-Shot Forecasting: Evaluation Pitfalls

Q: Does strong performance elsewhere guarantee strong sales forecasts here?

- No: pretraining data and our demand process may differ (domain shift).
- Forecast origins in a benchmark may overlap publicly available pretraining data; evaluate carefully for potential contamination.

Zero-Shot Forecasting: Evaluation Pitfalls

Q: Does strong performance elsewhere guarantee strong sales forecasts here?

- No: pretraining data and our demand process may differ (domain shift).
- Forecast origins in a benchmark may overlap publicly available pretraining data; evaluate carefully for potential contamination.
- Check seasonality, scale, intermittent/zero demand, forecast horizon, compute cost, and available covariates.

Zero-Shot Forecasting: Evaluation Pitfalls

Q: Does strong performance elsewhere guarantee strong sales forecasts here?

- No: pretraining data and our demand process may differ (domain shift).
- Forecast origins in a benchmark may overlap publicly available pretraining data; evaluate carefully for potential contamination.
- Check seasonality, scale, intermittent/zero demand, forecast horizon, compute cost, and available covariates.
- Compare against seasonal naive, ETS/ARIMA, boosting, and global models using the *same* backtest design.
- Predictive distributions and hierarchical reconciliation can still require additional validation and processing.

Global vs Foundation: The Key Distinction

Q: Are these mutually exclusive categories?

Global vs Foundation: The Key Distinction

Q: Are these mutually exclusive categories?

	Company-trained global	**Pretrained foundation**
Training corpus	Company's many SKU histories	Broad pretraining corpus
Use on our SKUs	Estimate weights from our data	Reuse pretrained weights; optionally adapt
Architecture	AR, boosting, RNN, transformer, etc.	Usually deep architecture; design varies
Risks	Negative transfer within company	Domain shift, pretraining mismatch, cost

Global vs Foundation: The Key Distinction

Q: Are these mutually exclusive categories?

	Company-trained global	**Pretrained foundation**
Training corpus	Company's many SKU histories	Broad pretraining corpus
Use on our SKUs	Estimate weights from our data	Reuse pretrained weights; optionally adapt
Architecture	AR, boosting, RNN, transformer, etc.	Usually deep architecture; design varies
Risks	Negative transfer within company	Domain shift, pretraining mismatch, cost

Takeaway: the distinction is predominantly *where parameters were learned and how they are reused*, not whether a transformer is involved.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

What Is Forecast Uncertainty?

Q: Why is one point forecast not enough?

What Is Forecast Uncertainty?

Q: Why is one point forecast not enough?

- A point forecast may target the conditional mean:

$$\widehat{s}_{t+h|t} = E[s_{t+h} \mid \mathcal{F}_t].$$

What Is Forecast Uncertainty?

Q: Why is one point forecast not enough?

- A point forecast may target the conditional mean:

$$\widehat{s}_{t+h|t} = E[s_{t+h} \mid \mathcal{F}_t].$$

- The realized outcome differs from the forecast:

$$s_{t+h} = \widehat{s}_{t+h|t} + e_{t+h|t}.$$

What Is Forecast Uncertainty?

Q: Why is one point forecast not enough?

- A point forecast may target the conditional mean:

$$\widehat{s}_{t+h|t} = E[s_{t+h} \mid \mathcal{F}_t].$$

- The realized outcome differs from the forecast:

$$s_{t+h} = \widehat{s}_{t+h|t} + e_{t+h|t}.$$

- A **prediction interval** $[L_{t+h|t}, U_{t+h|t}]$ describes plausible future *observations*; a **confidence interval** addresses uncertainty about a fixed parameter or conditional-mean estimate.

What Is Forecast Uncertainty?

Q: Why is one point forecast not enough?

- A point forecast may target the conditional mean:

$$\widehat{s}_{t+h|t} = E[s_{t+h} \mid \mathcal{F}_t].$$

- The realized outcome differs from the forecast:

$$s_{t+h} = \widehat{s}_{t+h|t} + e_{t+h|t}.$$

- A **prediction interval** $[L_{t+h|t}, U_{t+h|t}]$ describes plausible future *observations*; a **confidence interval** addresses uncertainty about a fixed parameter or conditional-mean estimate.
- These intervals answer different questions. Forecast uncertainty typically includes future shocks and sometimes parameter uncertainty.

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

- **Analytical/model-based:** derive forecast error variance from an ARIMA or state-space specification.

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

- **Analytical/model-based:** derive forecast error variance from an ARIMA or state-space specification.
- **Simulation-based:** draw many possible future paths from a fitted probability model (Monte Carlo).

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

- **Analytical/model-based:** derive forecast error variance from an ARIMA or state-space specification.
- **Simulation-based:** draw many possible future paths from a fitted probability model (Monte Carlo).
- **Resampling:** resample model errors or dependent blocks to approximate uncertainty (bootstrap).

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

- **Analytical/model-based:** derive forecast error variance from an ARIMA or state-space specification.
- **Simulation-based:** draw many possible future paths from a fitted probability model (Monte Carlo).
- **Resampling:** resample model errors or dependent blocks to approximate uncertainty (bootstrap).
- **Calibration-based:** use held-out forecast errors to calibrate intervals (conformal methods).

How Do We Quantify Forecast Uncertainty?

Q: What information supports an interval forecast?

- **Analytical/model-based:** derive forecast error variance from an ARIMA or state-space specification.
- **Simulation-based:** draw many possible future paths from a fitted probability model (Monte Carlo).
- **Resampling:** resample model errors or dependent blocks to approximate uncertainty (bootstrap).
- **Calibration-based:** use held-out forecast errors to calibrate intervals (conformal methods).
- All methods require assumptions for trustworthy coverage; no procedure is universally assumption-free under arbitrary time dependence.

Analytical Prediction Intervals: AR(1)

Q: Can we derive future uncertainty without simulations?

Analytical Prediction Intervals: AR(1)

Q: Can we derive future uncertainty without simulations?

- Suppose $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $E[\varepsilon_t] = 0$, independent shocks and $\text{Var}(\varepsilon_t) = \sigma^2$.

Analytical Prediction Intervals: AR(1)

Q: Can we derive future uncertainty without simulations?

- Suppose $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $E[\varepsilon_t] = 0$, independent shocks and $\text{Var}(\varepsilon_t) = \sigma^2$.
- Iterating h steps gives the forecast error

$$y_{t+h} - \widehat{y}_{t+h|t} = \sum_{j=0}^{h-1} \phi^j \varepsilon_{t+h-j}.$$

Analytical Prediction Intervals: AR(1)

Q: Can we derive future uncertainty without simulations?

- Suppose $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $E[\varepsilon_t] = 0$, independent shocks and $\text{Var}(\varepsilon_t) = \sigma^2$.
- Iterating h steps gives the forecast error

$$y_{t+h} - \widehat{y}_{t+h|t} = \sum_{j=0}^{h-1} \phi^j \varepsilon_{t+h-j}.$$

- With known parameters, forecast-error variance is

$$\sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \phi^{2j}.$$

Analytical Prediction Intervals: AR(1)

Q: Can we derive future uncertainty without simulations?

- Suppose $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $E[\varepsilon_t] = 0$, independent shocks and $\text{Var}(\varepsilon_t) = \sigma^2$.
- Iterating h steps gives the forecast error

$$y_{t+h} - \widehat{y}_{t+h|t} = \sum_{j=0}^{h-1} \phi^j \varepsilon_{t+h-j}.$$

- With known parameters, forecast-error variance is

$$\sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \phi^{2j}.$$

- Under Gaussian shocks, a nominal 95% prediction interval is $\widehat{y}_{t+h|t} \pm 1.96\sigma_h$.
- Parameter estimation adds uncertainty; model-based intervals can be miscalibrated after structural changes.

Simulation-Based Uncertainty

Q: What if the forecast distribution has no convenient formula?

Simulation-Based Uncertainty

Q: What if the forecast distribution has no convenient formula?

- Specify a fitted forecast mechanism and simulate many possible future paths conditional on information at t.

Simulation-Based Uncertainty

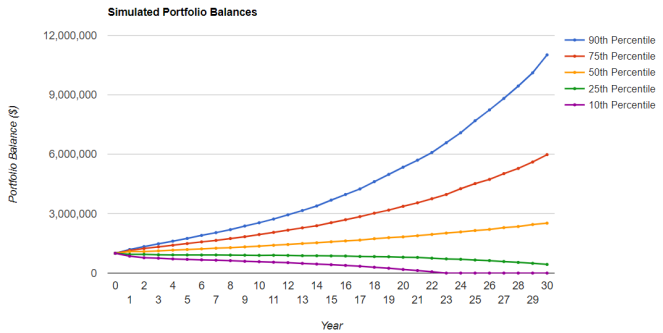
Q: What if the forecast distribution has no convenient formula?

- Specify a fitted forecast mechanism and simulate many possible future paths conditional on information at t.
- At each horizon, the empirical distribution across paths approximates the model's predictive distribution.

Simulation-Based Uncertainty

Q: What if the forecast distribution has no convenient formula?

- Specify a fitted forecast mechanism and simulate many possible future paths conditional on information at t.
- At each horizon, the empirical distribution across paths approximates the model's predictive distribution.
- The median and 5th/95th percentiles give a point summary and a central 90% **prediction interval**.
- Simulation also supports stress testing, nonlinear transformations, and aggregate outcomes.



Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

1. Specify $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.

Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

1. Specify $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.
2. Choose (a, ϕ, σ^2) from a calibrated model or a scenario.

Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

1. Specify $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.
2. Choose (a, ϕ, σ^2) from a calibrated model or a scenario.
3. Initialize from the latest observed y_t for a *conditional forecast*; for an unconditional stationary simulation one may initialize at $a/(1-\phi)$ if $|\phi| < 1$.

Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

1. Specify $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.
2. Choose (a, ϕ, σ^2) from a calibrated model or a scenario.
3. Initialize from the latest observed y_t for a *conditional forecast*; for an unconditional stationary simulation one may initialize at $a/(1-\phi)$ if $|\phi| < 1$.
4. Draw independent future shocks and propagate the model forward H steps.

Monte Carlo: A Simple Dynamic Example

Q: How do we generate plausible paths from an AR(1)?

1. Specify $y_t = a + \phi y_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$.
2. Choose (a, ϕ, σ^2) from a calibrated model or a scenario.
3. Initialize from the latest observed y_t for a *conditional forecast*; for an unconditional stationary simulation one may initialize at $a/(1-\phi)$ if $|\phi| < 1$.
4. Draw independent future shocks and propagate the model forward H steps.
5. Repeat B times, setting and recording a pseudorandom seed for reproducibility.

A Financial Example: Geometric Random Walk

Q: How do we simulate positive prices?

A Financial Example: Geometric Random Walk

Q: How do we simulate positive prices?

- Simple return $R_t = P_t/P_{t-1} - 1$; log return $r_t = \ln(P_t/P_{t-1}) = \ln(1 + R_t)$.

A Financial Example: Geometric Random Walk

Q: How do we simulate positive prices?

- Simple return $R_t = P_t/P_{t-1} - 1$; log return $r_t = \ln(P_t/P_{t-1}) = \ln(1 + R_t)$.
- If $r_t \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$, then

$$P_t = P_0 \exp\left(\sum_{j=1}^{t} r_j\right) > 0.$$

A Financial Example: Geometric Random Walk

Q: How do we simulate positive prices?

- Simple return $R_t = P_t/P_{t-1} - 1$; log return $r_t = \ln(P_t/P_{t-1}) = \ln(1 + R_t)$.
- If $r_t \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$, then

$$P_t = P_0 \exp\left(\sum_{j=1}^{t} r_j\right) > 0.$$

- Log prices are Gaussian at each horizon, hence prices are lognormal.
- This is an illustrative parametric simulation model, not a claim that actual asset returns are Gaussian or iid.

Why Use Log Returns Rather Than Gaussian Price Increments?

Q: What goes wrong if price changes are modeled additively?

Why Use Log Returns Rather Than Gaussian Price Increments?

Q: What goes wrong if price changes are modeled additively?

- An additive price random walk $P_t = P_{t-1} + u_t$ with Gaussian u_t can produce negative prices.

Why Use Log Returns Rather Than Gaussian Price Increments?

Q: What goes wrong if price changes are modeled additively?

- An additive price random walk $P_t = P_{t-1} + u_t$ with Gaussian u_t can produce negative prices.
- Modeling *gross returns* as lognormal gives

$$1 + R_t = e^{r_t} > 0, \qquad P_t = P_{t-1} e^{r_t} > 0.$$

Why Use Log Returns Rather Than Gaussian Price Increments?

Q: What goes wrong if price changes are modeled additively?

- An additive price random walk $P_t = P_{t-1} + u_t$ with Gaussian u_t can produce negative prices.
- Modeling *gross returns* as lognormal gives

$$1 + R_t = e^{r_t} > 0, \qquad P_t = P_{t-1} e^{r_t} > 0.$$

- Log returns add across time:

$$\ln\big(1 + R_t(H)\big) = \sum_{j=0}^{H-1} r_{t-j}, \quad 1 + R_t(H) = \prod_{j=0}^{H-1} (1 + R_{t-j}).$$

Why Use Log Returns Rather Than Gaussian Price Increments?

Q: What goes wrong if price changes are modeled additively?

- An additive price random walk $P_t = P_{t-1} + u_t$ with Gaussian u_t can produce negative prices.
- Modeling *gross returns* as lognormal gives

$$1 + R_t = e^{r_t} > 0, \qquad P_t = P_{t-1} e^{r_t} > 0.$$

- Log returns add across time:

$$\ln\left(1 + R_t(H)\right) = \sum_{j=0}^{H-1} r_{t-j}, \quad 1 + R_t(H) = \prod_{j=0}^{H-1} (1 + R_{t-j}).$$

- If independent log returns are Gaussian, their sum is Gaussian; the compounded *simple* return is generally not Gaussian.

The Lognormal Distribution

Q: What price distribution follows from Gaussian log returns?

The Lognormal Distribution

Q: What price distribution follows from Gaussian log returns?

- If $X > 0$ and $\ln X \sim N(\mu, \sigma^2)$, then

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], \qquad x > 0.$$

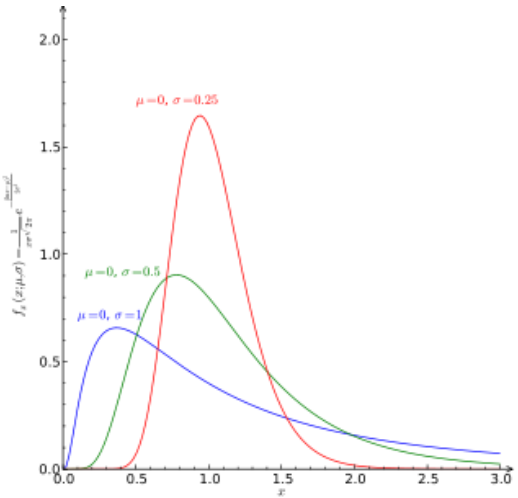
The Lognormal Distribution

Q: What price distribution follows from Gaussian log returns?

- If $X > 0$ and $\ln X \sim N(\mu, \sigma^2)$, then

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], \qquad x > 0.$$

- The support is positive, and the level distribution is right-skewed.
- This ensures positive simulated prices but does not remove tail risk or guarantee the lognormal model is empirically correct.



Bootstrap Methods: Two Different Goals

Q: What can we learn by resampling observed data?

Bootstrap Methods: Two Different Goals

Q: What can we learn by resampling observed data?

- **Parameter inference:** resample data and re-estimate a statistic or model to approximate sampling variability (e.g. standard error of a coefficient).

Bootstrap Methods: Two Different Goals

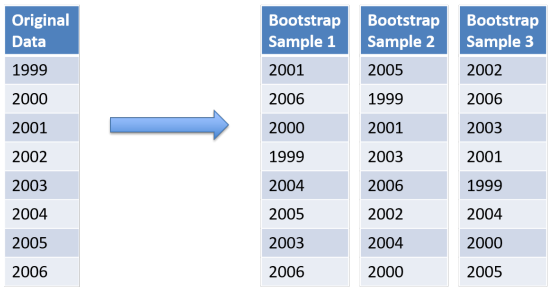
Q: What can we learn by resampling observed data?

- **Parameter inference:** resample data and re-estimate a statistic or model to approximate sampling variability (e.g. standard error of a coefficient).
- **Forecast uncertainty:** resample innovations/residuals or blocks, propagate the forecasting model, and form an empirical predictive distribution.

Bootstrap Methods: Two Different Goals

Q: What can we learn by resampling observed data?

- **Parameter inference:** resample data and re-estimate a statistic or model to approximate sampling variability (e.g. standard error of a coefficient).
- **Forecast uncertainty:** resample innovations/residuals or blocks, propagate the forecasting model, and form an empirical predictive distribution.
- These differ: a confidence interval for the *mean* is not the same as an interval for a *new observation*.
- The resampling design must reflect the dependence and forecast target.



Original Data
1999
2000
2001
2002
2003
2004
2005
2006

Bootstrap Sample 1	Bootstrap Sample 2	Bootstrap Sample 3
2001	2005	2002
2006	1999	2006
2000	2001	2003
1999	2003	2001
2004	2006	1999
2005	2002	2004
2003	2004	2000
2006	2000	2005

Why a Block Bootstrap?

Q: Why not resample individual time observations indiscriminately?

Why a Block Bootstrap?

Q: Why not resample individual time observations indiscriminately?

- Serially dependent observations need not remain dependent after independent row resampling.

Why a Block Bootstrap?

Q: Why not resample individual time observations indiscriminately?

- Serially dependent observations need not remain dependent after independent row resampling.
- **Block bootstrap:** draw contiguous observed segments and concatenate them to preserve local temporal patterns.

Why a Block Bootstrap?

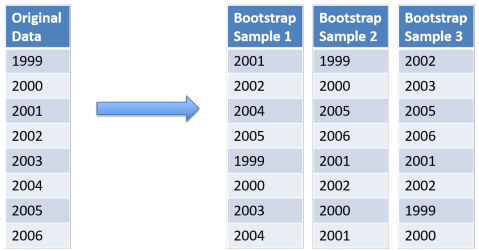
Q: Why not resample individual time observations indiscriminately?

- Serially dependent observations need not remain dependent after independent row resampling.
- **Block bootstrap:** draw contiguous observed segments and concatenate them to preserve local temporal patterns.
- Examples include moving-block, circular-block, and stationary bootstraps.

Why a Block Bootstrap?

Q: Why not resample individual time observations indiscriminately?

- Serially dependent observations need not remain dependent after independent row resampling.
- **Block bootstrap:** draw contiguous observed segments and concatenate them to preserve local temporal patterns.
- Examples include moving-block, circular-block, and stationary bootstraps.
- Block length trades retaining dependence against having enough distinct resampling units.
- A block bootstrap requires suitable assumptions (often weak dependence/stationarity); it is not a universal fix for trends, structural breaks, or nonstationarity.



Original Data
1999
2000
2001
2002
2003
2004
2005
2006

Bootstrap Sample 1	Bootstrap Sample 2	Bootstrap Sample 3
2001	1999	2002
2002	2000	2003
2004	2005	2005
2005	2006	2006
1999	2001	2001
2000	2002	2002
2003	2000	1999
2004	2001	2000

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

1. Build each time-series-compatible resample $\mathcal{D}^{*(b)}$.

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

1. Build each time-series-compatible resample $\mathcal{D}^{*(b)}$.
2. Re-estimate the statistic or model parameter $\widehat{\theta}^{*(b)}$.

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

1. Build each time-series-compatible resample $\mathcal{D}^{*(b)}$.
2. Re-estimate the statistic or model parameter $\widehat{\theta}^{*(b)}$.
3. Repeat for $b = 1, \ldots, B$.

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

1. Build each time-series-compatible resample $\mathcal{D}^{*(b)}$.
2. Re-estimate the statistic or model parameter $\widehat{\theta}^{*(b)}$.
3. Repeat for $b = 1, \ldots, B$.
4. Estimate its bootstrap standard error:

$$\widehat{\mathrm{se}}_{\mathrm{boot}}(\widehat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\theta}^{*(b)} - \overline{\widehat{\theta}^{*}}\right)^2}.$$

Bootstrap for Estimator Uncertainty

Q: How would we estimate the sampling variation of a statistic?

1. Build each time-series-compatible resample $\mathcal{D}^{*(b)}$.
2. Re-estimate the statistic or model parameter $\widehat{\theta}^{*(b)}$.
3. Repeat for $b = 1, \ldots, B$.
4. Estimate its bootstrap standard error:

$$\widehat{\mathrm{se}}_{\mathrm{boot}}(\widehat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\widehat{\theta}^{*(b)} - \overline{\widehat{\theta}^{*}}\right)^2}.$$

- To estimate forecast uncertainty, instead simulate future innovations and propagate them through the forecasting system.

Conformal Prediction: Core Idea

Q: Can historical forecast errors calibrate intervals?

Conformal Prediction: Core Idea

Q: Can historical forecast errors calibrate intervals?

- Split chronologically into **training**, **calibration**, and **test** periods.

Conformal Prediction: Core Idea

Q: Can historical forecast errors calibrate intervals?

- Split chronologically into **training**, **calibration**, and **test** periods.
- Fit a point model on training data, then freeze it when computing calibration errors at valid historical forecast origins.

Conformal Prediction: Core Idea

Q: Can historical forecast errors calibrate intervals?

- Split chronologically into **training**, **calibration**, and **test** periods.
- Fit a point model on training data, then freeze it when computing calibration errors at valid historical forecast origins.
- For horizon h, compute calibration nonconformity scores

$$e_{t,h} = \left| s_{t+h} - \widehat{s}_{t+h|t} \right| .$$

Conformal Prediction: Core Idea

Q: Can historical forecast errors calibrate intervals?

- Split chronologically into **training**, **calibration**, and **test** periods.
- Fit a point model on training data, then freeze it when computing calibration errors at valid historical forecast origins.
- For horizon h, compute calibration nonconformity scores

$$e_{t,h} = \left| s_{t+h} - \widehat{s}_{t+h|t} \right| .$$

- Use a high empirical score quantile $q_{h,1-\alpha}$ to construct

$$\left[\widehat{s}_{T+h|T} - q_{h,1-\alpha},\ \widehat{s}_{T+h|T} + q_{h,1-\alpha} \right].$$

Conformal Quantile: Mathematical Detail

Q: Which empirical calibration quantile yields the nominal interval?

Conformal Quantile: Mathematical Detail

Q: Which empirical calibration quantile yields the nominal interval?

- Suppose n calibration errors for a fixed horizon are ordered:

$$e_{(1),h} \leq e_{(2),h} \leq \cdots \leq e_{(n),h}.$$

Conformal Quantile: Mathematical Detail

Q: Which empirical calibration quantile yields the nominal interval?

- Suppose n calibration errors for a fixed horizon are ordered:

$$e_{(1),h} \leq e_{(2),h} \leq \cdots \leq e_{(n),h}.$$

- Standard split conformal uses the order-statistic index

$$k = \lceil (n+1)(1-\alpha) \rceil, \qquad q_h = e_{(k),h},$$

with $q_h = +\infty$ if $k > n$.

Conformal Quantile: Mathematical Detail

Q: Which empirical calibration quantile yields the nominal interval?

- Suppose n calibration errors for a fixed horizon are ordered:

$$e_{(1),h} \leq e_{(2),h} \leq \cdots \leq e_{(n),h}.$$

- Standard split conformal uses the order-statistic index

$$k = \lceil (n+1)(1-\alpha) \rceil, \qquad q_h = e_{(k),h},$$

  with $q_h = +\infty$ if $k > n$.
- Then use $[\widehat{s}_{T+h|T} - q_h,\ \widehat{s}_{T+h|T} + q_h]$.

Conformal Quantile: Mathematical Detail

Q: Which empirical calibration quantile yields the nominal interval?

- Suppose n calibration errors for a fixed horizon are ordered:

$$e_{(1),h} \leq e_{(2),h} \leq \cdots \leq e_{(n),h}.$$

- Standard split conformal uses the order-statistic index

$$k = \lceil (n+1)(1-\alpha) \rceil, \qquad q_h = e_{(k),h},$$

  with $q_h = +\infty$ if $k > n$.
- Then use $[\widehat{s}_{T+h|T} - q_h,\ \widehat{s}_{T+h|T} + q_h]$.
- The familiar finite-sample marginal coverage argument assumes calibration and test errors are exchangeable; dependent time-series scores generally require more care.

Conformal Prediction: Coverage and Assumptions

Q: When can we expect these intervals to cover the future?

Conformal Prediction: Coverage and Assumptions

Q: When can we expect these intervals to cover the future?

- Under exchangeability of calibration and future scores, standard split conformal methods provide marginal finite-sample coverage using a finite-sample-adjusted quantile.

Conformal Prediction: Coverage and Assumptions

Q: When can we expect these intervals to cover the future?

- Under exchangeability of calibration and future scores, standard split conformal methods provide marginal finite-sample coverage using a finite-sample-adjusted quantile.
- Time-series forecast errors can be serially dependent, heteroskedastic, or subject to distribution shifts; exchangeability may fail.

Conformal Prediction: Coverage and Assumptions

Q: When can we expect these intervals to cover the future?

- Under exchangeability of calibration and future scores, standard split conformal methods provide marginal finite-sample coverage using a finite-sample-adjusted quantile.
- Time-series forecast errors can be serially dependent, heteroskedastic, or subject to distribution shifts; exchangeability may fail.
- Practical adaptations include rolling calibration, horizon-specific errors, and dependence-aware procedures, but their guarantees need their own assumptions.

Conformal Prediction: Coverage and Assumptions

Q: When can we expect these intervals to cover the future?

- Under exchangeability of calibration and future scores, standard split conformal methods provide marginal finite-sample coverage using a finite-sample-adjusted quantile.
- Time-series forecast errors can be serially dependent, heteroskedastic, or subject to distribution shifts; exchangeability may fail.
- Practical adaptations include rolling calibration, horizon-specific errors, and dependence-aware procedures, but their guarantees need their own assumptions.
- Evaluate empirical coverage *and width* on later test origins; do not claim assumption-free time-series coverage.

Evaluating Probabilistic Forecasts

Q: Is a wider interval automatically better?

Evaluating Probabilistic Forecasts

Q: Is a wider interval automatically better?

- Calibration: does a nominal 90% prediction interval contain roughly 90% of outcomes over valid test forecast origins?

Evaluating Probabilistic Forecasts

Q: Is a wider interval automatically better?

- Calibration: does a nominal 90% prediction interval contain roughly 90% of outcomes over valid test forecast origins?
- Sharpness: among intervals with appropriate coverage, narrower is more informative.

Evaluating Probabilistic Forecasts

Q: Is a wider interval automatically better?

- Calibration: does a nominal 90% prediction interval contain roughly 90% of outcomes over valid test forecast origins?
- Sharpness: among intervals with appropriate coverage, narrower is more informative.
- Quantile regression can estimate selected conditional quantiles directly; *pinball loss* evaluates them.

Evaluating Probabilistic Forecasts

Q: Is a wider interval automatically better?

- Calibration: does a nominal 90% prediction interval contain roughly 90% of outcomes over valid test forecast origins?
- Sharpness: among intervals with appropriate coverage, narrower is more informative.
- Quantile regression can estimate selected conditional quantiles directly; *pinball loss* evaluates them.
- Distributional scoring rules such as CRPS evaluate an entire predictive distribution, balancing calibration and sharpness.
- Always assess uncertainty by forecast horizon, period, and relevant SKU groups.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Use Case: Forecast Updating

Q: How should our forecast of December company sales evolve over the year?

Use Case: Forecast Updating

Q: How should our forecast of December company sales evolve over the year?

- In June, forecast **December** sales using information available in June.

Use Case: Forecast Updating

Q: How should our forecast of December company sales evolve over the year?

- In June, forecast **December** sales using information available in June.
- In July, observe new information and revise the forecast for the *same December target*.

Use Case: Forecast Updating

Q: How should our forecast of December company sales evolve over the year?

- In June, forecast **December** sales using information available in June.
- In July, observe new information and revise the forecast for the *same December target*.
- Continue in August, September, ... while the target month remains fixed.

Use Case: Forecast Updating

Q: How should our forecast of December company sales evolve over the year?

- In June, forecast **December** sales using information available in June.
- In July, observe new information and revise the forecast for the *same December target*.
- Continue in August, September, ... while the target month remains fixed.
- Write the forecast as $\widehat{s}_{T|t} = E[s_T \mid \mathcal{F}_t]$ with fixed target date T and expanding information sets $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$.

Three Different Ways a Forecast Can Change

Q: Does every forecast revision require retraining?

Three Different Ways a Forecast Can Change

Q: Does every forecast revision require retraining?

1. **Forecast revision:** recompute the target-date prediction using newly available inputs, keeping fitted parameters fixed.

Three Different Ways a Forecast Can Change

Q: Does every forecast revision require retraining?

1. **Forecast revision:** recompute the target-date prediction using newly available inputs, keeping fitted parameters fixed.
2. **State update:** update an evolving latent or recurrent state using new observations, with fitted parameters fixed.

Three Different Ways a Forecast Can Change

Q: Does every forecast revision require retraining?

1. **Forecast revision:** recompute the target-date prediction using newly available inputs, keeping fitted parameters fixed.
2. **State update:** update an evolving latent or recurrent state using new observations, with fitted parameters fixed.
3. **Parameter re-estimation:** refit or fine-tune the model using a longer training sample.

Three Different Ways a Forecast Can Change

Q: Does every forecast revision require retraining?

1. **Forecast revision:** recompute the target-date prediction using newly available inputs, keeping fitted parameters fixed.
2. **State update:** update an evolving latent or recurrent state using new observations, with fitted parameters fixed.
3. **Parameter re-estimation:** refit or fine-tune the model using a longer training sample.

- These operations can occur separately or together; none is equivalent to the other two.
- As the target approaches, the horizon $h = T - t$ also decreases.

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

- **ARIMA/ADL:** use the newest observations and the same estimated coefficients to calculate an updated forecast from the new origin.

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

- **ARIMA/ADL:** use the newest observations and the same estimated coefficients to calculate an updated forecast from the new origin.
- **Boosted trees:** refresh the available feature vector and apply a previously trained horizon-specific or multi-output model.

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

- **ARIMA/ADL:** use the newest observations and the same estimated coefficients to calculate an updated forecast from the new origin.
- **Boosted trees:** refresh the available feature vector and apply a previously trained horizon-specific or multi-output model.
- **LSTM/DeepAR:** process the new observation to update its recurrent state, then generate a new forecast; weights need not change.

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

- **ARIMA/ADL:** use the newest observations and the same estimated coefficients to calculate an updated forecast from the new origin.
- **Boosted trees:** refresh the available feature vector and apply a previously trained horizon-specific or multi-output model.
- **LSTM/DeepAR:** process the new observation to update its recurrent state, then generate a new forecast; weights need not change.
- **Transformer/foundation model:** provide the updated historical context and generate a revised forecast using fixed pretrained weights.

Updating an ARIMA, ML, or Deep-Learning Forecast

Q: Can ordinary forecasting models revise a fixed-date target?

- **ARIMA/ADL:** use the newest observations and the same estimated coefficients to calculate an updated forecast from the new origin.
- **Boosted trees:** refresh the available feature vector and apply a previously trained horizon-specific or multi-output model.
- **LSTM/DeepAR:** process the new observation to update its recurrent state, then generate a new forecast; weights need not change.
- **Transformer/foundation model:** provide the updated historical context and generate a revised forecast using fixed pretrained weights.
- These methods can all update coherently *with respect to their own specifications*; state-space modeling is one particularly explicit updating framework.

Why State Space Models Are Natural for Updating

Q: What if sales are driven by evolving trend and seasonality that we cannot observe directly?

Why State Space Models Are Natural for Updating

Q: What if sales are driven by evolving trend and seasonality that we cannot observe directly?

- Let $\mathbf{x}_t$ collect latent trend, seasonal, or other dynamic components.

Why State Space Models Are Natural for Updating

Q: What if sales are driven by evolving trend and seasonality that we cannot observe directly?

- Let $\mathbf{x}_t$ collect latent trend, seasonal, or other dynamic components.
- Linear state-space representation:

$$\underbrace{\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t}_{\text{state transition}}, \qquad \underbrace{s_t = H\mathbf{x}_t + \varepsilon_t}_{\text{observation}}.$$

Why State Space Models Are Natural for Updating

Q: What if sales are driven by evolving trend and seasonality that we cannot observe directly?

- Let $\mathbf{x}_t$ collect latent trend, seasonal, or other dynamic components.
- Linear state-space representation:

$$\underbrace{\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t}_{\text{state transition}}, \qquad \underbrace{s_t = H\mathbf{x}_t + \varepsilon_t}_{\text{observation}}.$$

- F, H specify how hidden states evolve and generate the observed sales; $\boldsymbol{\eta}_t$ and ε_t are process and measurement innovations.

Why State Space Models Are Natural for Updating

Q: What if sales are driven by evolving trend and seasonality that we cannot observe directly?

- Let $\mathbf{x}_t$ collect latent trend, seasonal, or other dynamic components.
- Linear state-space representation:

$$\underbrace{\mathbf{x}_t = F\mathbf{x}_{t-1} + \boldsymbol{\eta}_t}_{\text{state transition}}, \qquad \underbrace{s_t = H\mathbf{x}_t + \varepsilon_t}_{\text{observation}}.$$

- F, H specify how hidden states evolve and generate the observed sales; $\boldsymbol{\eta}_t$ and ε_t are process and measurement innovations.
- The Kalman filter updates estimates of hidden states as new sales observations arrive.

State Space: Forecast From the Current State

Q: How do latent trend and seasonality generate a December forecast?

State Space: Forecast From the Current State

Q: How do latent trend and seasonality generate a December forecast?

- Given filtered state $\widehat{\mathbf{x}}_{t|t}$ and zero-mean innovations,

$$\widehat{\mathbf{x}}_{T|t} = F^{T-t}\widehat{\mathbf{x}}_{t|t}, \qquad \widehat{s}_{T|t} = HF^{T-t}\widehat{\mathbf{x}}_{t|t}.$$

State Space: Forecast From the Current State

Q: How do latent trend and seasonality generate a December forecast?

- Given filtered state $\widehat{\mathbf{x}}_{t|t}$ and zero-mean innovations,

$$\widehat{\mathbf{x}}_{T|t} = F^{T-t}\widehat{\mathbf{x}}_{t|t}, \qquad \widehat{s}_{T|t} = HF^{T-t}\widehat{\mathbf{x}}_{t|t}.$$

- With nonzero intercepts, deterministic inputs, or changing transition matrices, include those terms rather than simply using F^{T-t}.

State Space: Forecast From the Current State

Q: How do latent trend and seasonality generate a December forecast?

- Given filtered state $\widehat{\mathbf{x}}_{t|t}$ and zero-mean innovations,

$$\widehat{\mathbf{x}}_{T|t} = F^{T-t}\widehat{\mathbf{x}}_{t|t}, \qquad \widehat{s}_{T|t} = HF^{T-t}\widehat{\mathbf{x}}_{t|t}.$$

- With nonzero intercepts, deterministic inputs, or changing transition matrices, include those terms rather than simply using F^{T-t}.
- New data can change our estimate of the *current state* even if the estimated transition and observation matrices remain fixed.

Kalman Filter: Predict, Observe, Update

Q: How does a new monthly observation revise our belief about hidden sales drivers?

Kalman Filter: Predict, Observe, Update

Q: How does a new monthly observation revise our belief about hidden sales drivers?

$$\begin{aligned}
\text{Predict state:} \quad \widehat{\mathbf{x}}_{t|t-1} &= F\widehat{\mathbf{x}}_{t-1|t-1}, \\
\text{Innovation:} \quad v_t &= s_t - H\widehat{\mathbf{x}}_{t|t-1}, \\
\text{Update state:} \quad \widehat{\mathbf{x}}_{t|t} &= \widehat{\mathbf{x}}_{t|t-1} + K_t v_t.
\end{aligned}$$

Kalman Filter: Predict, Observe, Update

Q: How does a new monthly observation revise our belief about hidden sales drivers?

$$\begin{aligned}
\text{Predict state:} \quad \widehat{\mathbf{x}}_{t|t-1} &= F\widehat{\mathbf{x}}_{t-1|t-1}, \\
\text{Innovation:} \quad v_t &= s_t - H\widehat{\mathbf{x}}_{t|t-1}, \\
\text{Update state:} \quad \widehat{\mathbf{x}}_{t|t} &= \widehat{\mathbf{x}}_{t|t-1} + K_t v_t.
\end{aligned}$$

- v_t is the surprise relative to the model's prediction of newly observed sales.
- K_t controls how strongly the surprise revises the hidden-state estimate.
- This is *filtering*, not re-estimation of F or H.

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

- Let $P_{t|t-1}$ be predicted state-estimation-error covariance and R measurement-noise covariance.

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

- Let $P_{t|t-1}$ be predicted state-estimation-error covariance and R measurement-noise covariance.
- Linear-Gaussian gain:

$$K_t = P_{t|t-1}H'\left(HP_{t|t-1}H' + R\right)^{-1}.$$

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

- Let $P_{t|t-1}$ be predicted state-estimation-error covariance and R measurement-noise covariance.
- Linear-Gaussian gain:

$$K_t = P_{t|t-1}H'\left(HP_{t|t-1}H' + R\right)^{-1}.$$

- Large measurement noise R generally reduces the weight placed on new observed sales.

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

- Let $P_{t|t-1}$ be predicted state-estimation-error covariance and R measurement-noise covariance.
- Linear-Gaussian gain:

$$K_t = P_{t|t-1}H'\left(HP_{t|t-1}H' + R\right)^{-1}.$$

- Large measurement noise R generally reduces the weight placed on new observed sales.
- Greater uncertainty about the state generally raises the scope for learning from a new observation (the exact effect depends on H and the covariance structure).

Kalman Gain: How Much Do We Trust New Information?

Q: Why does one sales surprise cause a large revision while another does not?

- Let $P_{t|t-1}$ be predicted state-estimation-error covariance and R measurement-noise covariance.
- Linear-Gaussian gain:

$$K_t = P_{t|t-1}H'\left(HP_{t|t-1}H' + R\right)^{-1}.$$

- Large measurement noise R generally reduces the weight placed on new observed sales.
- Greater uncertainty about the state generally raises the scope for learning from a new observation (the exact effect depends on H and the covariance structure).
- Under correct linear-Gaussian assumptions, this gain gives the minimum mean-squared-error state update.

Forecast Revisions and News

Q: What precisely triggers a revision to the same December target?

Forecast Revisions and News

Q: What precisely triggers a revision to the same December target?

- Suppose the model parameters remain fixed and no new deterministic inputs arrive.

Forecast Revisions and News

Q: What precisely triggers a revision to the same December target?

- Suppose the model parameters remain fixed and no new deterministic inputs arrive.
- The part of a newly observed s_{t+1} that the model already expected does not constitute news; the innovation does.

Forecast Revisions and News

Q: What precisely triggers a revision to the same December target?

- Suppose the model parameters remain fixed and no new deterministic inputs arrive.
- The part of a newly observed s_{t+1} that the model already expected does not constitute news; the innovation does.
- For a fixed future target $T > t+1$, the linear state-space forecast revision is

$$\widehat{s}_{T|t+1} - \widehat{s}_{T|t} = HF^{T-t-1}K_{t+1}v_{t+1}.$$

Forecast Revisions and News

Q: What precisely triggers a revision to the same December target?

- Suppose the model parameters remain fixed and no new deterministic inputs arrive.
- The part of a newly observed s_{t+1} that the model already expected does not constitute news; the innovation does.
- For a fixed future target $T > t+1$, the linear state-space forecast revision is

$$\widehat{s}_{T|t+1} - \widehat{s}_{T|t} = HF^{T-t-1}K_{t+1}v_{t+1}.$$

- A small news shock may produce a small revision; a large or informative surprise may produce a substantial revision.
- Revisions are not necessarily *smooth* or monotone.

Forecast Updating and the Fair-Game Property

Q: Should the forecast revision itself be predictable?

Forecast Updating and the Fair-Game Property

Q: Should the forecast revision itself be predictable?

- Under a correctly specified conditional-mean forecast,

$$\widehat{s}_{T|t} = E[s_T \mid \mathcal{F}_t].$$

Forecast Updating and the Fair-Game Property

Q: Should the forecast revision itself be predictable?

- Under a correctly specified conditional-mean forecast,

$$\widehat{s}_{T|t} = E[s_T \mid \mathcal{F}_t].$$

- Then the tower property implies

$$E\left[\widehat{s}_{T|t+1} - \widehat{s}_{T|t} \mid \mathcal{F}_t\right] = 0.$$

Forecast Updating and the Fair-Game Property

Q: Should the forecast revision itself be predictable?

- Under a correctly specified conditional-mean forecast,

$$\widehat{s}_{T|t} = E[s_T \mid \mathcal{F}_t].$$

- Then the tower property implies

$$E\left[\widehat{s}_{T|t+1} - \widehat{s}_{T|t} \mid \mathcal{F}_t\right] = 0.$$

- In this idealized sense, the *new information component* of the forecast revision should be unpredictable from yesterday's information.

Forecast Updating and the Fair-Game Property

Q: Should the forecast revision itself be predictable?

- Under a correctly specified conditional-mean forecast,

$$\widehat{s}_{T|t} = E[s_T \mid \mathcal{F}_t].$$

- Then the tower property implies

$$E\left[\widehat{s}_{T|t+1} - \widehat{s}_{T|t} \mid \mathcal{F}_t\right] = 0.$$

- In this idealized sense, the *new information component* of the forecast revision should be unpredictable from yesterday's information.
- Systematically predictable revisions can indicate misspecification, insufficient information processing, or deliberate forecast smoothing.

When to Update States, Inputs, or Parameters

Q: How should a forecasting system operate through time?

When to Update States, Inputs, or Parameters

Q: How should a forecasting system operate through time?

Operation	**What changes**	**Example**
Refresh features	New known sales, prices, calendar covariates	Boosted-tree forecast
Filter hidden state	Posterior estimate of current trend/seasonality	Kalman filter
Update recurrent state	Current hidden/cell state	LSTM / DeepAR
Re-estimate weights	Model coefficients or network parameters	Rolling AR fit or DL fine-tuning

When to Update States, Inputs, or Parameters

Q: How should a forecasting system operate through time?

Operation	What changes	Example
Refresh features	New known sales, prices, calendar covariates	Boosted-tree forecast
Filter hidden state	Posterior estimate of current trend/seasonality	Kalman filter
Update recurrent state	Current hidden/cell state	LSTM / DeepAR
Re-estimate weights	Model coefficients or network parameters	Rolling AR fit or DL fine-tuning

Takeaway: forecast updating is universal; state space provides a particularly explicit decomposition of new data, uncertainty, and revisions.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

A Different Forecasting Target: Volatility

Q: What if predicting uncertainty matters more than predicting the mean?

A Different Forecasting Target: Volatility

Q: What if predicting uncertainty matters more than predicting the mean?

- For financial returns, conditional mean predictability can be weak while the magnitude of future movements is persistent.

A Different Forecasting Target: Volatility

Q: What if predicting uncertainty matters more than predicting the mean?

- For financial returns, conditional mean predictability can be weak while the magnitude of future movements is persistent.
- Instead of forecasting $E[r_{t+1} \mid \mathcal{F}_t]$, forecast conditional variance

$$h_{t+1|t} = \mathrm{Var}(r_{t+1} \mid \mathcal{F}_t).$$

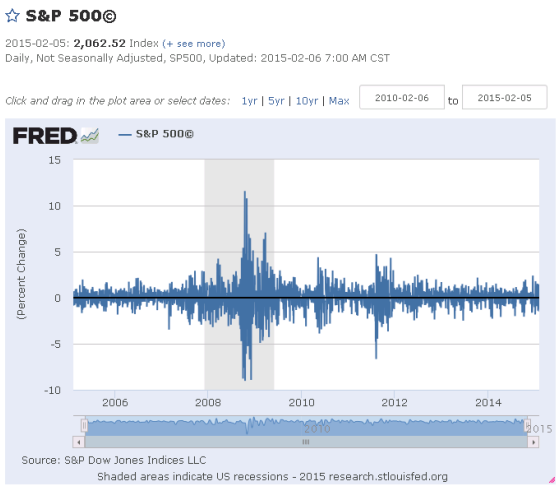
A Different Forecasting Target: Volatility

Q: What if predicting uncertainty matters more than predicting the mean?

- For financial returns, conditional mean predictability can be weak while the magnitude of future movements is persistent.
- Instead of forecasting $E[r_{t+1} \mid \mathcal{F}_t]$, forecast conditional variance

$$h_{t+1|t} = \mathrm{Var}(r_{t+1} \mid \mathcal{F}_t).$$

- Volatility is the square root of variance; forecasting one is not identical to forecasting the other.



Volatility Clustering

Q: Why can the next period's risk be predictable even when returns are not?

Volatility Clustering

Q: Why can the next period's risk be predictable even when returns are not?

- Raw returns may display little autocorrelation, yet squared or absolute returns often display substantial positive autocorrelation.

Volatility Clustering

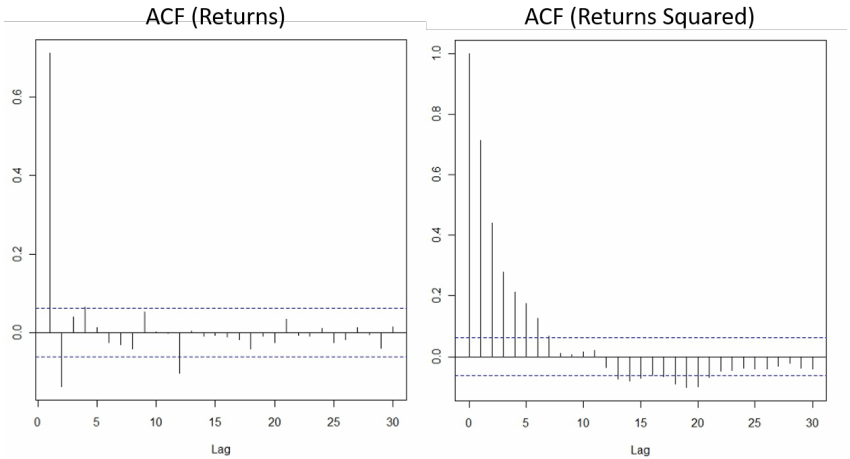
Q: Why can the next period's risk be predictable even when returns are not?

- Raw returns may display little autocorrelation, yet squared or absolute returns often display substantial positive autocorrelation.
- Large movements (of either sign) tend to be followed by large movements; quiet periods tend to remain quiet.

Volatility Clustering

Q: Why can the next period's risk be predictable even when returns are not?

- Raw returns may display little autocorrelation, yet squared or absolute returns often display substantial positive autocorrelation.
- Large movements (of either sign) tend to be followed by large movements; quiet periods tend to remain quiet.
- This motivates models for *conditional variance* rather than only conditional mean.



Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

- **Heavy tails:** large absolute returns occur more often than a simple normal model would imply.

Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

- **Heavy tails:** large absolute returns occur more often than a simple normal model would imply.
- **Volatility clustering:** persistence in $|r_t|$ and r_t^2.

Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

- **Heavy tails:** large absolute returns occur more often than a simple normal model would imply.
- **Volatility clustering:** persistence in $|r_t|$ and r_t^2.
- **Asymmetry/leverage:** negative returns can be associated with stronger subsequent volatility increases than equally large positive returns.

Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

- **Heavy tails:** large absolute returns occur more often than a simple normal model would imply.
- **Volatility clustering:** persistence in $|r_t|$ and r_t^2.
- **Asymmetry/leverage:** negative returns can be associated with stronger subsequent volatility increases than equally large positive returns.
- **Co-movement:** volatility can rise together across assets and markets.

Stylized Facts About Financial Volatility

Q: Which properties should a volatility model potentially explain?

- **Heavy tails:** large absolute returns occur more often than a simple normal model would imply.
- **Volatility clustering:** persistence in $|r_t|$ and r_t^2.
- **Asymmetry/leverage:** negative returns can be associated with stronger subsequent volatility increases than equally large positive returns.
- **Co-movement:** volatility can rise together across assets and markets.
- GARCH(1,1) targets persistence and clustering; extensions address asymmetry and cross-series dynamics.

Unconditional Dispersion Measures

Q: How can we measure variability using a historical sample?

Unconditional Dispersion Measures

Q: How can we measure variability using a historical sample?

- Sample variance of returns:

$$\widehat{\sigma}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \bar{r})^2.$$

Unconditional Dispersion Measures

Q: How can we measure variability using a historical sample?

- Sample variance of returns:

$$\widehat{\sigma}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \bar{r})^2.$$

- Mean absolute deviation (a different dispersion measure):

$$\text{MAD} = \frac{1}{T}\sum_{t=1}^{T}|r_t - \bar{r}|.$$

Unconditional Dispersion Measures

Q: How can we measure variability using a historical sample?

- Sample variance of returns:

$$\widehat{\sigma}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \bar{r})^2.$$

- Mean absolute deviation (a different dispersion measure):

$$\text{MAD} = \frac{1}{T}\sum_{t=1}^{T}|r_t - \bar{r}|.$$

Range-Based Volatility Measures

Q: Can daily high and low prices improve on squared close-to-close returns?

Range-Based Volatility Measures

Q: Can daily high and low prices improve on squared close-to-close returns?

- Parkinson daily range-based *variance proxy*:

$$\widehat{\sigma}^2_{P,t} = \frac{\left(\ln H_t - \ln L_t\right)^2}{4 \ln 2}.$$

Range-Based Volatility Measures

Q: Can daily high and low prices improve on squared close-to-close returns?

- Parkinson daily range-based *variance proxy*:

$$\widehat{\sigma}^2_{P,t} = \frac{\left(\ln H_t - \ln L_t\right)^2}{4 \ln 2}.$$

- Garman–Klass daily range-based variance proxy:

$$\widehat{\sigma}^2_{GK,t} = \frac{1}{2}\left(\ln \frac{H_t}{L_t}\right)^2 - (2\ln 2 - 1)\left(\ln \frac{C_t}{O_t}\right)^2.$$

Range-Based Volatility Measures

Q: Can daily high and low prices improve on squared close-to-close returns?

- Parkinson daily range-based *variance proxy*:

$$\widehat{\sigma}^2_{P,t} = \frac{\left(\ln H_t - \ln L_t\right)^2}{4 \ln 2}.$$

- Garman–Klass daily range-based variance proxy:

$$\widehat{\sigma}^2_{GK,t} = \frac{1}{2}\left(\ln \frac{H_t}{L_t}\right)^2 - (2\ln 2 - 1)\left(\ln \frac{C_t}{O_t}\right)^2.$$

- O, H, L, C are open, high, low, close. Range estimators require assumptions (e.g. limited drift and no overnight jumps for the canonical formulas).

Realized Variance from Intraday Returns

Q: Can high-frequency data measure the volatility of a trading day?

Realized Variance from Intraday Returns

Q: Can high-frequency data measure the volatility of a trading day?

- Let $p_{t,j}$ be the log price at intraday observation j on day t.

Realized Variance from Intraday Returns

Q: Can high-frequency data measure the volatility of a trading day?

- Let $p_{t,j}$ be the log price at intraday observation j on day t.
- For M intraday intervals, define $r_{t,j} = p_{t,j} - p_{t,j-1}$ and

$$RV_t^{(d)} = \sum_{j=1}^{M} r_{t,j}^2.$$

Realized Variance from Intraday Returns

Q: Can high-frequency data measure the volatility of a trading day?

- Let $p_{t,j}$ be the log price at intraday observation j on day t.
- For M intraday intervals, define $r_{t,j} = p_{t,j} - p_{t,j-1}$ and

$$RV_t^{(d)} = \sum_{j=1}^{M} r_{t,j}^2.$$

- With 30-minute returns during a 6.5-hour regular US stock-market session, there are 13 intervals.

Realized Variance from Intraday Returns

Q: Can high-frequency data measure the volatility of a trading day?

- Let $p_{t,j}$ be the log price at intraday observation j on day t.
- For M intraday intervals, define $r_{t,j} = p_{t,j} - p_{t,j-1}$ and

$$RV_t^{(d)} = \sum_{j=1}^{M} r_{t,j}^2.$$

- With 30-minute returns during a 6.5-hour regular US stock-market session, there are 13 intervals.
- Realized *variance* is an ex-post measure/proxy, not by itself a forecasting model.
- Microstructure noise, jumps, sampling frequency, and overnight returns can complicate the relationship to latent integrated variance.

Why Not Just Use One Historical Variance?

Q: What changes when volatility varies through time?

Why Not Just Use One Historical Variance?

Q: What changes when volatility varies through time?

- An unconditional historical variance treats all dates as having the same expected variance.

Why Not Just Use One Historical Variance?

Q: What changes when volatility varies through time?

- An unconditional historical variance treats all dates as having the same expected variance.
- But if turbulence clusters, a large recent shock may increase the next period's *conditional* variance.

Why Not Just Use One Historical Variance?

Q: What changes when volatility varies through time?

- An unconditional historical variance treats all dates as having the same expected variance.
- But if turbulence clusters, a large recent shock may increase the next period's *conditional* variance.
- Two complementary strategies:
  - **GARCH:** model the latent conditional variance as a function of past shocks and variance.
  - **HAR-RV:** predict tomorrow's observed realized variance using realized volatility measures averaged across daily, weekly, and monthly scales.

ARCH(1): Conditional Heteroskedasticity

Q: Can yesterday's return surprise explain today's variance?

ARCH(1): Conditional Heteroskedasticity

Q: Can yesterday's return surprise explain today's variance?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$

$$E[z_t \mid \mathcal{F}_{t-1}] = 0, \quad E[z_t^2 \mid \mathcal{F}_{t-1}] = 1,$$

$$h_t = \omega + \alpha \varepsilon_{t-1}^2, \qquad \omega > 0,\ \alpha \geq 0.$$

ARCH(1): Conditional Heteroskedasticity

Q: Can yesterday's return surprise explain today's variance?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$
$$E[z_t \mid \mathcal{F}_{t-1}] = 0, \quad E[z_t^2 \mid \mathcal{F}_{t-1}] = 1,$$
$$h_t = \omega + \alpha \varepsilon_{t-1}^2, \qquad \omega > 0,\ \alpha \geq 0.$$

- Conditional mean of ε_t is zero; conditional variance is h_t.

ARCH(1): Conditional Heteroskedasticity

Q: Can yesterday's return surprise explain today's variance?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$
$$E[z_t \mid \mathcal{F}_{t-1}] = 0, \quad E[z_t^2 \mid \mathcal{F}_{t-1}] = 1,$$
$$h_t = \omega + \alpha \varepsilon_{t-1}^2, \qquad \omega > 0,\ \alpha \geq 0.$$

- Conditional mean of ε_t is zero; conditional variance is h_t.
- A large squared innovation ε_{t-1}^2 raises h_t.

ARCH(1): Conditional Heteroskedasticity

Q: Can yesterday's return surprise explain today's variance?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$
$$E[z_t \mid \mathcal{F}_{t-1}] = 0, \quad E[z_t^2 \mid \mathcal{F}_{t-1}] = 1,$$
$$h_t = \omega + \alpha \varepsilon_{t-1}^2, \qquad \omega > 0,\ \alpha \geq 0.$$

- Conditional mean of ε_t is zero; conditional variance is h_t.
- A large squared innovation ε_{t-1}^2 raises h_t.
- ARCH $=$ *autoregressive conditional heteroskedasticity*; the variation is in conditional variance, not necessarily the unconditional variance.

ARCH(1): Conditional vs Unconditional Variance

Q: Can volatility change every day while having a stable long-run variance?

ARCH(1): Conditional vs Unconditional Variance

Q: Can volatility change every day while having a stable long-run variance?

- Conditional on information at $t-1$,

$$\mathrm{Var}(\varepsilon_t \mid \mathcal{F}_{t-1}) = \omega + \alpha\varepsilon_{t-1}^2.$$

ARCH(1): Conditional vs Unconditional Variance

Q: Can volatility change every day while having a stable long-run variance?

- Conditional on information at $t-1$,

$$\mathrm{Var}(\varepsilon_t \mid \mathcal{F}_{t-1}) = \omega + \alpha\varepsilon_{t-1}^2.$$

- If $0 \leq \alpha < 1$ and a stationary second moment exists, then

$$\mathrm{Var}(\varepsilon_t) = E[h_t] = \omega + \alpha E[\varepsilon_{t-1}^2] = \boxed{\frac{\omega}{1-\alpha}}.$$

ARCH(1): Conditional vs Unconditional Variance

Q: Can volatility change every day while having a stable long-run variance?

- Conditional on information at $t-1$,

$$\mathrm{Var}(\varepsilon_t \mid \mathcal{F}_{t-1}) = \omega + \alpha\varepsilon_{t-1}^2.$$

- If $0 \leq \alpha < 1$ and a stationary second moment exists, then

$$\mathrm{Var}(\varepsilon_t) = E[h_t] = \omega + \alpha E[\varepsilon_{t-1}^2] = \boxed{\frac{\omega}{1-\alpha}}.$$

- Thus a changing *conditional* variance can coexist with a finite, constant *unconditional* variance.

GARCH: Let Variance Have Its Own Persistence

Q: Why should volatility depend on its previous forecast as well as its latest shock?

GARCH: Let Variance Have Its Own Persistence

Q: Why should volatility depend on its previous forecast as well as its latest shock?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$

$$h_t = \omega + \sum_{j=1}^{p} \beta_j h_{t-j} + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2.$$

GARCH: Let Variance Have Its Own Persistence

Q: Why should volatility depend on its previous forecast as well as its latest shock?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$

$$h_t = \omega + \sum_{j=1}^{p} \beta_j h_{t-j} + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2.$$

- β_j coefficients capture lagged conditional variance (GARCH terms); α_i capture lagged squared surprises (ARCH terms).

GARCH: Let Variance Have Its Own Persistence

Q: Why should volatility depend on its previous forecast as well as its latest shock?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$

$$h_t = \omega + \sum_{j=1}^{p} \beta_j h_{t-j} + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2.$$

- β_j coefficients capture lagged conditional variance (GARCH terms); α_i capture lagged squared surprises (ARCH terms).
- In this *GARCH(p, q)* convention, p counts variance lags and q counts squared-shock lags. Check software conventions.

GARCH: Let Variance Have Its Own Persistence

Q: Why should volatility depend on its previous forecast as well as its latest shock?

$$r_t = \mu + \varepsilon_t, \qquad \varepsilon_t = \sqrt{h_t}\, z_t,$$

$$h_t = \omega + \sum_{j=1}^{p} \beta_j h_{t-j} + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2.$$

- β_j coefficients capture lagged conditional variance (GARCH terms); α_i capture lagged squared surprises (ARCH terms).
- In this *GARCH(p, q)* convention, p counts variance lags and q counts squared-shock lags. Check software conventions.
- Standard sufficient nonnegativity conditions: $\omega > 0$, $\alpha_i, \beta_j \geq 0$; $\sum_i \alpha_i + \sum_j \beta_j < 1$ is a familiar sufficient condition for finite long-run variance in the standard specification.

GARCH(1,1): The Workhorse

Q: What do α and β do?

GARCH(1,1): The Workhorse

Q: What do α and β do?

$$\boxed{h_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta h_{t-1}}$$

GARCH(1,1): The Workhorse

Q: What do α and β do?

$$\boxed{h_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta h_{t-1}}$$

- α: impact of newly observed squared return shocks.

GARCH(1,1): The Workhorse

Q: What do α and β do?

$$\boxed{h_t = \omega + \alpha \varepsilon_{t-1}^2 + \beta h_{t-1}}$$

- α: impact of newly observed squared return shocks.
- β: persistence of yesterday's conditional variance.

GARCH(1,1): The Workhorse

Q: What do α and β do?

$$\boxed{h_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta h_{t-1}}$$

- α: impact of newly observed squared return shocks.
- β: persistence of yesterday's conditional variance.
- If $\alpha + \beta < 1$, long-run variance is

$$\bar{h} = \frac{\omega}{1 - \alpha - \beta}.$$

GARCH(1,1): The Workhorse

Q: What do α and β do?

$$\boxed{h_t = \omega + \alpha\varepsilon_{t-1}^2 + \beta h_{t-1}}$$

- α: impact of newly observed squared return shocks.
- β: persistence of yesterday's conditional variance.
- If $\alpha + \beta < 1$, long-run variance is

$$\bar{h} = \frac{\omega}{1 - \alpha - \beta}.$$

- When $\alpha + \beta$ is near one, shocks to conditional variance decay slowly; this does not guarantee a stationary process in every more general specification.

GARCH(1,1): Multi-Step Volatility Forecasting

Q: How does today's variance shock affect forecasts several days ahead?

GARCH(1,1): Multi-Step Volatility Forecasting

Q: How does today's variance shock affect forecasts several days ahead?

- Next period, ε_t and h_t are observed/filtered:

$$\widehat{h}_{t+1|t} = \omega + \alpha\varepsilon_t^2 + \beta h_t.$$

GARCH(1,1): Multi-Step Volatility Forecasting

Q: How does today's variance shock affect forecasts several days ahead?

- Next period, ε_t and h_t are observed/filtered:

$$\widehat{h}_{t+1|t} = \omega + \alpha\varepsilon_t^2 + \beta h_t.$$

- For $h \geq 2$, use $E_t[\varepsilon_{t+h-1}^2] = \widehat{h}_{t+h-1|t}$:

$$\widehat{h}_{t+h|t} = \omega + (\alpha + \beta)\widehat{h}_{t+h-1|t}.$$

GARCH(1,1): Multi-Step Volatility Forecasting

Q: How does today's variance shock affect forecasts several days ahead?

- Next period, ε_t and h_t are observed/filtered:

$$\widehat{h}_{t+1|t} = \omega + \alpha\varepsilon_t^2 + \beta h_t.$$

- For $h \geq 2$, use $E_t[\varepsilon_{t+h-1}^2] = \widehat{h}_{t+h-1|t}$:

$$\widehat{h}_{t+h|t} = \omega + (\alpha + \beta)\widehat{h}_{t+h-1|t}.$$

- With $\alpha + \beta < 1$, variance forecasts mean-revert toward $\bar{h} = \omega/(1 - \alpha - \beta)$.
- For a return prediction interval, forecast variance must be combined with distributional assumptions (e.g. conditional Gaussian or Student-t innovations).

Testing for ARCH Effects

Q: Is there evidence that past squared shocks predict current squared shocks?

Testing for ARCH Effects

Q: Is there evidence that past squared shocks predict current squared shocks?

1. Estimate an appropriate conditional-mean model and save residuals $\widehat{\varepsilon}_t$.

Testing for ARCH Effects

Q: Is there evidence that past squared shocks predict current squared shocks?

1. Estimate an appropriate conditional-mean model and save residuals $\widehat{\varepsilon}_t$.
2. Regress residual squares on a constant and q of their own lags:

$$\widehat{\varepsilon}_t^2 = a + \sum_{j=1}^{q} \gamma_j \widehat{\varepsilon}_{t-j}^2 + u_t.$$

Testing for ARCH Effects

Q: Is there evidence that past squared shocks predict current squared shocks?

1. Estimate an appropriate conditional-mean model and save residuals $\widehat{\varepsilon}_t$.
2. Regress residual squares on a constant and q of their own lags:

$$\widehat{\varepsilon}_t^2 = a + \sum_{j=1}^{q} \gamma_j \widehat{\varepsilon}_{t-j}^2 + u_t.$$

3. Test $H_0 : \gamma_1 = \cdots = \gamma_q = 0$ using Engle's ARCH LM statistic $nR^2 \overset{a}{\sim} \chi_q^2$ under regularity conditions.

Testing for ARCH Effects

Q: Is there evidence that past squared shocks predict current squared shocks?

1. Estimate an appropriate conditional-mean model and save residuals $\widehat{\varepsilon}_t$.
2. Regress residual squares on a constant and q of their own lags:

$$\widehat{\varepsilon}_t^2 = a + \sum_{j=1}^{q} \gamma_j \widehat{\varepsilon}_{t-j}^2 + u_t.$$

3. Test $H_0 : \gamma_1 = \cdots = \gamma_q = 0$ using Engle's ARCH LM statistic $nR^2 \overset{a}{\sim} \chi_q^2$ under regularity conditions.
4. Rejecting suggests conditional heteroskedasticity remains in the residuals; it does not by itself select the ideal GARCH order.

HAR-RV: A Different Approach to Persistence

Q: Can daily, weekly, and monthly volatility have different predictive effects?

HAR-RV: A Different Approach to Persistence

Q: Can daily, weekly, and monthly volatility have different predictive effects?

- HAR-RV (heterogeneous autoregressive realized volatility) models *observed realized variance* using averages over several timescales.

HAR-RV: A Different Approach to Persistence

Q: Can daily, weekly, and monthly volatility have different predictive effects?

- HAR-RV (heterogeneous autoregressive realized volatility) models *observed realized variance* using averages over several timescales.
- Daily, trailing-week, and trailing-month information may represent different persistence horizons.

HAR-RV: A Different Approach to Persistence

Q: Can daily, weekly, and monthly volatility have different predictive effects?

- HAR-RV (heterogeneous autoregressive realized volatility) models *observed realized variance* using averages over several timescales.
- Daily, trailing-week, and trailing-month information may represent different persistence horizons.
- This is a relatively simple regression-based forecasting model, rather than a latent conditional-variance recursion.
- It illustrates an important idea from earlier ML: carefully constructed lag summaries can capture rich temporal structure without a deep neural network.

HAR-RV: Constructing Daily, Weekly, Monthly Features

Q: What exactly are the inputs to the HAR-RV regression?

HAR-RV: Constructing Daily, Weekly, Monthly Features

Q: What exactly are the inputs to the HAR-RV regression?

$$RV_t^{(d)} = \text{realized variance during day } t,$$

$$RV_t^{(w)} = \frac{1}{5}\sum_{j=0}^{4} RV_{t-j}^{(d)},$$

$$RV_t^{(m)} = \frac{1}{22}\sum_{j=0}^{21} RV_{t-j}^{(d)}.$$

HAR-RV: Constructing Daily, Weekly, Monthly Features

Q: What exactly are the inputs to the HAR-RV regression?

$$RV_t^{(d)} = \text{realized variance during day } t,$$

$$RV_t^{(w)} = \frac{1}{5}\sum_{j=0}^{4} RV_{t-j}^{(d)},$$

$$RV_t^{(m)} = \frac{1}{22}\sum_{j=0}^{21} RV_{t-j}^{(d)}.$$

- All three predictors use information through day t, which is available at the forecast origin after the day ends.

HAR-RV: Constructing Daily, Weekly, Monthly Features

Q: What exactly are the inputs to the HAR-RV regression?

$$RV_t^{(d)} = \text{realized variance during day } t,$$

$$RV_t^{(w)} = \frac{1}{5}\sum_{j=0}^{4} RV_{t-j}^{(d)},$$

$$RV_t^{(m)} = \frac{1}{22}\sum_{j=0}^{21} RV_{t-j}^{(d)}.$$

- All three predictors use information through day t, which is available at the forecast origin after the day ends.
- The windows overlap; each coefficient reflects its predictor conditional on the other two.
- Here 5 and 22 are illustrative trading-day approximations to a week and a month.

HAR-RV: One-Day-Ahead Forecast Model

Q: How are the timescales combined?

HAR-RV: One-Day-Ahead Forecast Model

Q: How are the timescales combined?

$$\boxed{RV_{t+1}^{(d)} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} + u_{t+1}.}$$

HAR-RV: One-Day-Ahead Forecast Model

Q: How are the timescales combined?

$$\boxed{RV_{t+1}^{(d)} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} + u_{t+1}.}$$

- Estimate coefficients by regression on available daily realized-variance history.

HAR-RV: One-Day-Ahead Forecast Model

Q: How are the timescales combined?

$$\boxed{RV_{t+1}^{(d)} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} + u_{t+1}.}$$

- Estimate coefficients by regression on available daily realized-variance history.
- β_d, β_w, and β_m capture the association of next-day realized variance with short-, medium-, and longer-window persistence.

HAR-RV: One-Day-Ahead Forecast Model

Q: How are the timescales combined?

$$RV_{t+1}^{(d)} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} + u_{t+1}.$$

- Estimate coefficients by regression on available daily realized-variance history.
- β_d, β_w, and β_m capture the association of next-day realized variance with short-, medium-, and longer-window persistence.
- A linear HAR-RV forecast can be negative; practitioners may use log-RV or other positive transformations/model constraints.
- Re-estimation, prediction intervals, and backtests should respect chronological ordering and measurement conventions.

GARCH vs HAR-RV

Q: Which volatility process are we modeling?

GARCH vs HAR-RV

Q: Which volatility process are we modeling?

	GARCH(1,1)	**HAR-RV**
Target	Conditional variance of innovations	Next-period realized variance
Main inputs	Lagged squared shocks and conditional variance	Daily, weekly, monthly realized-variance averages
Data	Returns (often daily)	Intraday-derived realized-variance history
Estimation	Likelihood / QML commonly used	OLS/regression variants common
Key mechanism	Recursion with volatility persistence	Linear combination of multiple timescales

GARCH vs HAR-RV

Q: Which volatility process are we modeling?

	GARCH(1,1)	**HAR-RV**
Target	Conditional variance of innovations	Next-period realized variance
Main inputs	Lagged squared shocks and conditional variance	Daily, weekly, monthly realized-variance averages
Data	Returns (often daily)	Intraday-derived realized-variance history
Estimation	Likelihood / QML commonly used	OLS/regression variants common
Key mechanism	Recursion with volatility persistence	Linear combination of multiple timescales

Takeaway: complementary tools; compare on the same evaluation target and horizon.

Extensions: Asymmetric and Multivariate Volatility

Q: What do these two workhorses leave out?

Extensions: Asymmetric and Multivariate Volatility

Q: What do these two workhorses leave out?

- **GJR-GARCH / EGARCH:** allow volatility reactions to depend on the sign as well as magnitude of shocks.

Extensions: Asymmetric and Multivariate Volatility

Q: What do these two workhorses leave out?

- **GJR-GARCH / EGARCH:** allow volatility reactions to depend on the sign as well as magnitude of shocks.
- **Stochastic volatility:** treat log variance as a separate latent stochastic process, often estimated with filtering or Bayesian methods.

Extensions: Asymmetric and Multivariate Volatility

Q: What do these two workhorses leave out?

- **GJR-GARCH / EGARCH:** allow volatility reactions to depend on the sign as well as magnitude of shocks.
- **Stochastic volatility:** treat log variance as a separate latent stochastic process, often estimated with filtering or Bayesian methods.
- **Multivariate models:** represent co-movement and time-varying covariance across assets.

Extensions: Asymmetric and Multivariate Volatility

Q: What do these two workhorses leave out?

- **GJR-GARCH / EGARCH:** allow volatility reactions to depend on the sign as well as magnitude of shocks.
- **Stochastic volatility:** treat log variance as a separate latent stochastic process, often estimated with filtering or Bayesian methods.
- **Multivariate models:** represent co-movement and time-varying covariance across assets.
- **ML extensions:** predict realized variance with flexible lag features and covariates, or combine realized measures with parametric variance models.

Outline

5 Forecast
- Forecasting Framework
- Setup & Evaluation
- Classical Time Series Models
- Econometric Models
- Machine Learning Models
- Deep Learning
- Global / Panel Models
- Hierarchical
- Foundation Models
- Uncertainty and Prediction Intervals
- Forecast Updating
- Volatility Forecasting
- Paradigm Comparison

Forecasting Choices: Four Separate Dimensions

Q: How should we assemble a forecasting system?

Forecasting Choices: Four Separate Dimensions

Q: How should we assemble a forecasting system?

Dimension	**Choice**	**Question answered**
Architecture	ARIMA/ADL, state space, tree ML, LSTM, transformer	How does the model represent dynamics?
Training scope	Local, pooled/global, pretrained foundation	Which histories inform estimation?
Horizon strategy	Direct, recursive, multi-output	How are future dates generated?
Output constraints	Prediction intervals, hierarchy, updates	What must the forecasts satisfy?

Forecasting Choices: Four Separate Dimensions

Q: How should we assemble a forecasting system?

Dimension	**Choice**	**Question answered**
Architecture	ARIMA/ADL, state space, tree ML, LSTM, transformer	How does the model represent dynamics?
Training scope	Local, pooled/global, pretrained foundation	Which histories inform estimation?
Horizon strategy	Direct, recursive, multi-output	How are future dates generated?
Output constraints	Prediction intervals, hierarchy, updates	What must the forecasts satisfy?

Example: global DeepAR with autoregressive forecasts and uncertainty, followed by hierarchical reconciliation.

Forecasting: Model Selection Map (I)

Q: Which basic model is a reasonable starting point?

Forecasting: Model Selection Map (I)

Q: Which basic model is a reasonable starting point?

Data / forecasting challenge	**Candidates to benchmark**
One series, short history	Naive/seasonal naive, ETS, ARIMA
One series plus economic drivers	ADL, ARIMAX, ECM if supported by long-run structure
Several interacting series	VAR; VECM when cointegrated
Nonlinear lag and feature effects	Regression trees, random forests, boosted trees

Forecasting: Model Selection Map (I)

Q: Which basic model is a reasonable starting point?

Data / forecasting challenge	**Candidates to benchmark**
One series, short history	Naive/seasonal naive, ETS, ARIMA
One series plus economic drivers	ADL, ARIMAX, ECM if supported by long-run structure
Several interacting series	VAR; VECM when cointegrated
Nonlinear lag and feature effects	Regression trees, random forests, boosted trees

Principle: start with a strong, appropriately chosen baseline; escalate complexity when out-of-sample evidence warrants it.

Forecasting: Model Selection Map (II)

Q: When might learned representations or shared training help?

Forecasting: Model Selection Map (II)

Q: When might learned representations or shared training help?

Data / forecasting challenge	**Candidates to benchmark**
Rich temporal patterns	Feedforward/N-BEATS, LSTM, transformer models
Many related series; limited history each	Pooled models, global boosting, DeepAR, global transformers
Pretraining potentially useful	Foundation models; check transfer against task-specific baselines
Required aggregate consistency	Reconcile SKU, division, and company forecasts after producing base forecasts

Forecasting: Model Selection Map (II)

Q: When might learned representations or shared training help?

Data / forecasting challenge	Candidates to benchmark
Rich temporal patterns	Feedforward/N-BEATS, LSTM, transformer models
Many related series; limited history each	Pooled models, global boosting, DeepAR, global transformers
Pretraining potentially useful	Foundation models; check transfer against task-specific baselines
Required aggregate consistency	Reconcile SKU, division, and company forecasts after producing base forecasts

Important: these are candidates, not a performance ranking. Use rolling-origin evaluation with a common target, horizon, and loss.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.
- **Training scope:** local, within-division pooled, company-wide global, or pretrained/fine-tuned.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.
- **Training scope:** local, within-division pooled, company-wide global, or pretrained/fine-tuned.
- **Risk and uncertainty:** analytic intervals, Monte Carlo/bootstrap paths, or calibrated conformal procedures.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.
- **Training scope:** local, within-division pooled, company-wide global, or pretrained/fine-tuned.
- **Risk and uncertainty:** analytic intervals, Monte Carlo/bootstrap paths, or calibrated conformal procedures.
- **Aggregation:** reconcile company, division, and SKU forecasts when identities must hold.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.
- **Training scope:** local, within-division pooled, company-wide global, or pretrained/fine-tuned.
- **Risk and uncertainty:** analytic intervals, Monte Carlo/bootstrap paths, or calibrated conformal procedures.
- **Aggregation:** reconcile company, division, and SKU forecasts when identities must hold.
- **Updating:** refresh inputs or latent states as information arrives; separately decide when to re-estimate parameters.

Forecasting: Additional Requirements

Q: What decisions remain after choosing the forecasting model?

- **Forecast horizon:** direct, recursive, or joint multi-output, appropriate to the required target dates.
- **Training scope:** local, within-division pooled, company-wide global, or pretrained/fine-tuned.
- **Risk and uncertainty:** analytic intervals, Monte Carlo/bootstrap paths, or calibrated conformal procedures.
- **Aggregation:** reconcile company, division, and SKU forecasts when identities must hold.
- **Updating:** refresh inputs or latent states as information arrives; separately decide when to re-estimate parameters.
- **Evaluation:** preserve chronology, forecast-origin information, benchmark comparisons, and the loss that matters for the business decision.