香港城市大學
City University of Hong Kong

IS6400:
Business Data Analytics

Time Series I: Smoothing

專業 創新 胸懷全球
Professional · Creative
For The World

Delivery mode adjustment

- A set of slides will be in companion with the Jupyter notebooks.
  - https://github.com/zhiyzuo/IS6400-Regression

- As before, you could simply run the notebook but no need to change code for the purpose of this course.

- We will discuss the code line by line in tutorial.

Your feedback is always appreciated

- Please do not hesitate to contact me if you have any confusion.

- My regular OH: 10am to 11:45am on Wednesdays
  - https://cityu.zoom.us/j/99754454964?pwd=dzVMb1kvS3VTd2UvVTgzMzh0N3hrUT09

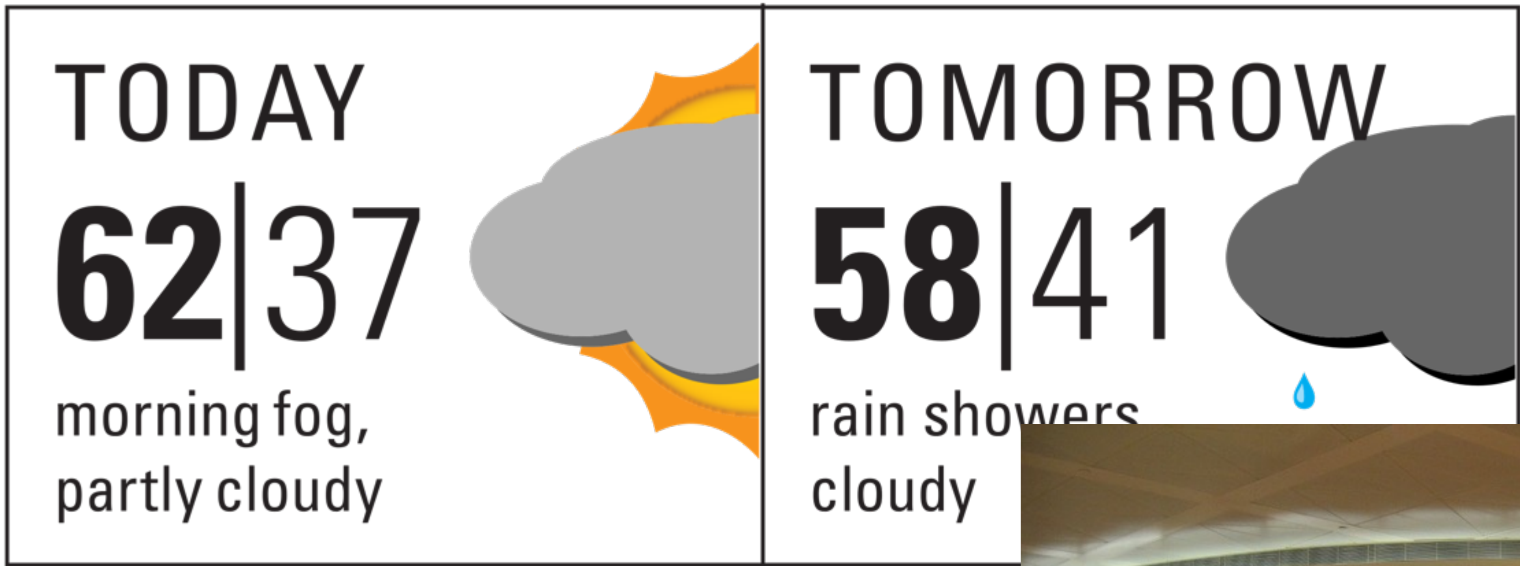
- But we can set up other times.

Week 6 Makeup

- A make-up lecture for Week 6 materials will be made on **March 5th from 2pm to 5pm.**

- The idea is for me to record the lecture and tutorial via a new presentation approach. But if any of you are interested, please join me. Recordings will be made public.
  - https://cityu.zoom.us/j/95851096414?pwd=WGU3T04vY3crTlZWc1A3MVQ5V0txUT09

Forecasting



Forecasting via Judgement

Why go this route?

- Little or limited access to data
- Important to include domain knowledge

- Example:
  - Forecasting a new product…
  - Estimating the listed price for an apartment

Australian Cigarette Package

Judgmental Forecasts

- One common approach: Delphi approach

Expert panel formation

Task distribution to panel

Forecast on proposed analogy

Assess similarity with target situation

Final forecast by weighted average

Quantitative approach: Our Focus

- When there's data!

What we will talk about

- Time series features
- Exponential Smoothing
- AutoRegresison

Week 7

Week 8

Our journey begins with…

- What is time series data?

Different types of data

- Cross-sectional: Records about different objects (collected at the same time)

- Example?
  - Student grade from IS6400 in 21/22 SemB
  - Features of hotels on Airbnb in Jan 2022
  - ...

Different types of data

- Time Series: repetitive data collected from the same object over time

- Example?
  - Xiaomi’s stock price over time
  - Hilton Garden Inn’s ratings over time

Different types of data

- Panel: Time series + Cross-sectional
  - Multiple objects
  - Multiple data collection time

- Examples?
  - Airbnb rooms features from 2018 to 2022
  - S&P500 firms' financial indicators from 2018 to 2022

We focus on time series

Where we start?

- First step: no rush into any models!
  - Know your data!

```
google_stock.describe().round()
```

	Open	High	Low	Close	Adj Close	Volume
count	1260.0	1260.0	1260.0	1260.0	1260.0	1260.0
mean	1487.0	1502.0	1473.0	1488.0	1488.0	1563063.0
std	619.0	624.0	613.0	618.0	618.0	696063.0
min	807.0	821.0	803.0	814.0	814.0	346800.0
25%	1071.0	1082.0	1059.0	1071.0	1071.0	1114875.0
50%	1207.0	1220.0	1200.0	1209.0	1209.0	1388250.0
75%	1731.0	1749.0	1717.0	1736.0	1736.0	1778525.0
max	3037.0	3042.0	2998.0	3014.0	3014.0	6207000.0

Autocorrelation

- Recall what is correlation?

- Correlation in time series: <u>autocorrelation</u>
  - Correlation with the time series itself
  - Between lagged values!

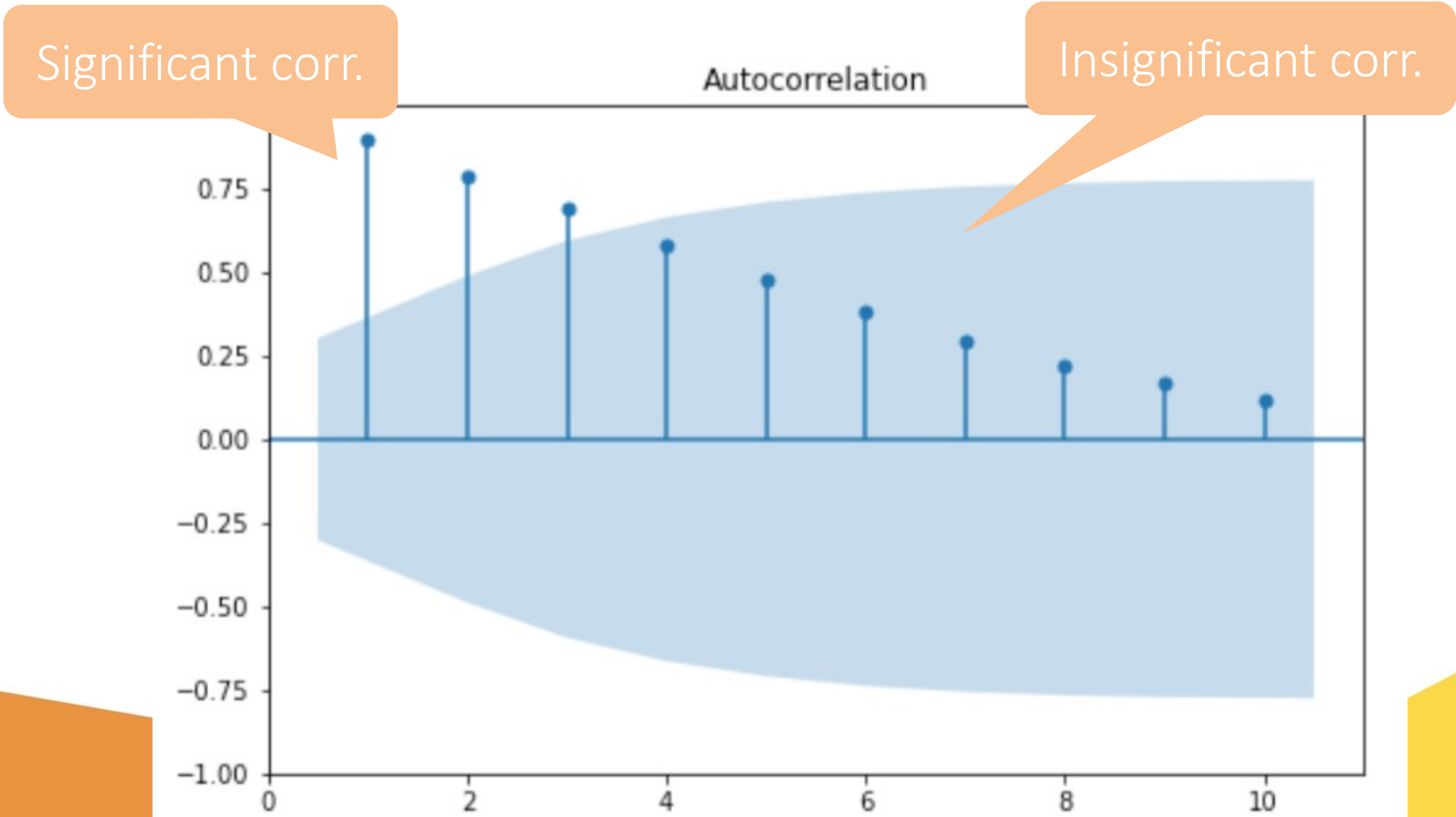
Autocorrelation

- The math?

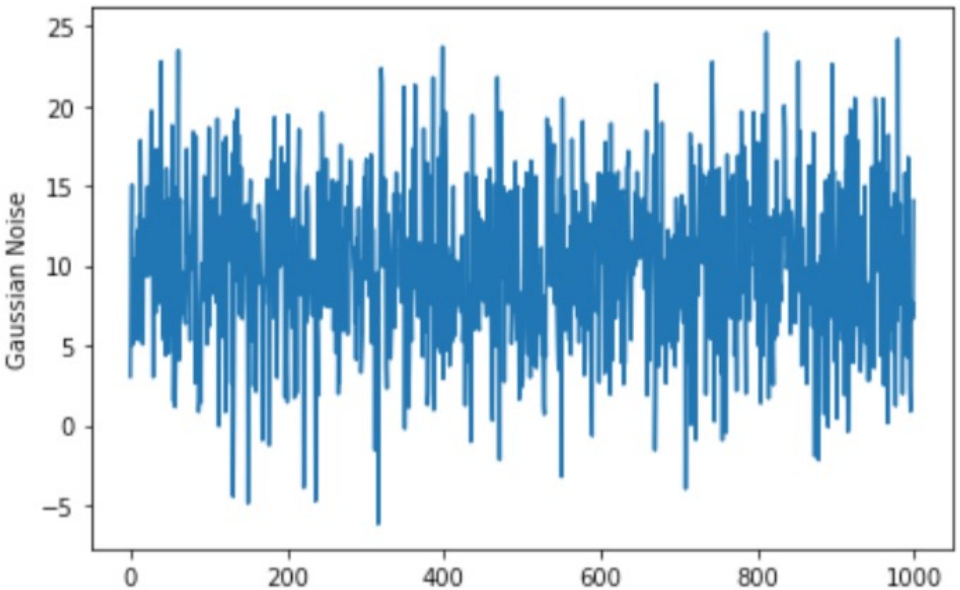
$$r_k = \frac{\sum_{t=k+1}^{T}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

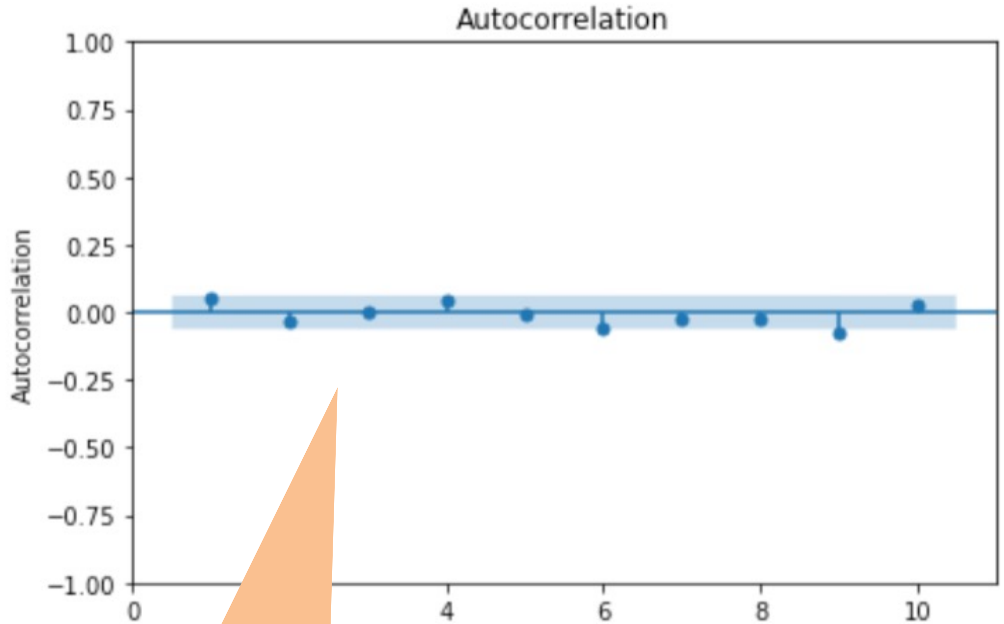
- ”k” is lag here.
  - $k = 1$: Correlation with its previous time unit

Autocorrelation in action



White noise: no autocorrelation

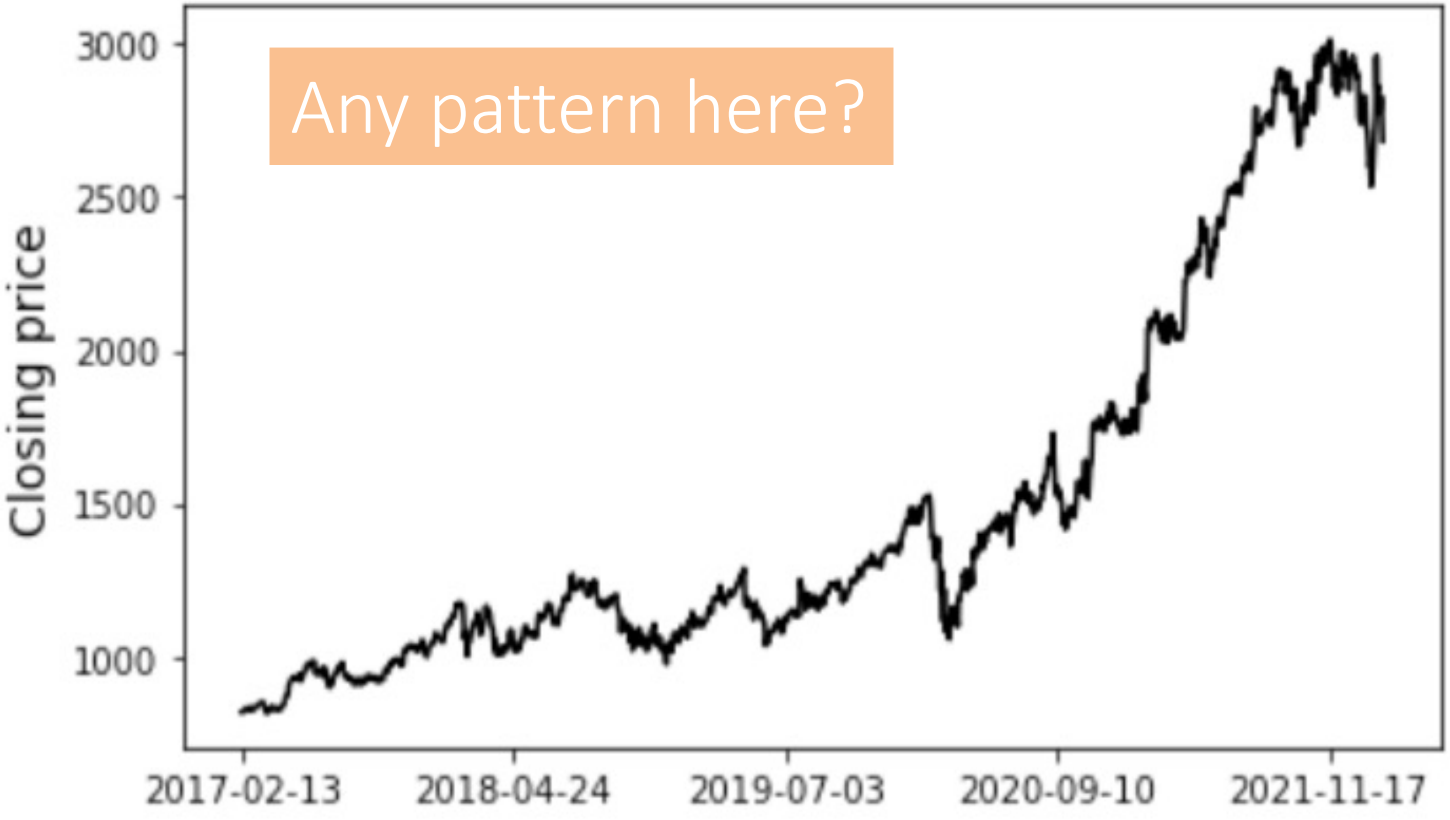




None is significant

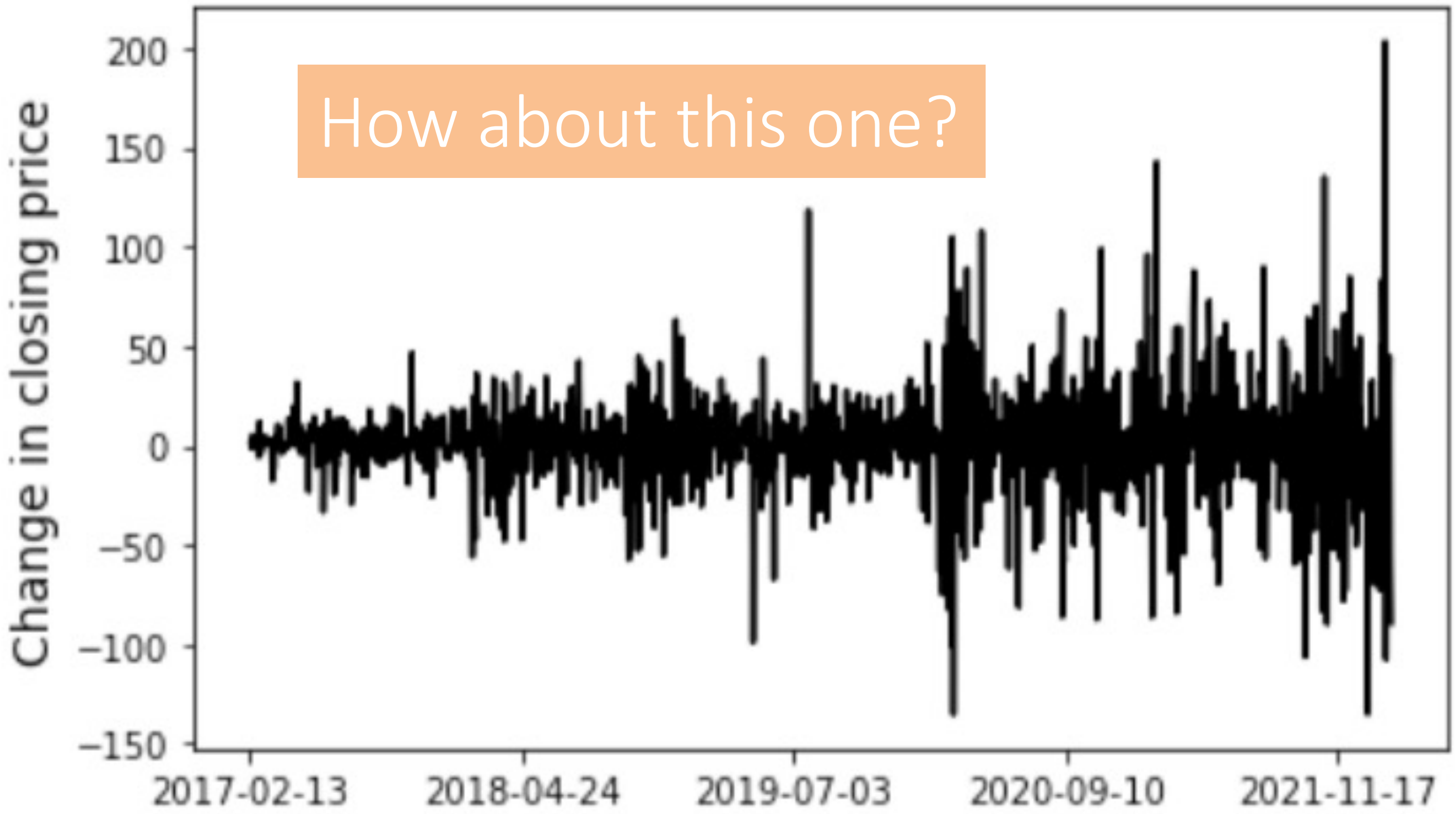
Stationarity

- A time series in non-stationary if there are any patterns of trend or seasonality.



Stationarity

- A time series is non-stationary if there are any patterns of trend or seasonality.



Stationarity: Statistical test

- We may also quantitively tell whether one is stationary or not.
  - Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test
  - Null hypothesis: A time series is stationary

```
kpss_closing_price = sm.tsa.stattools.kpss(google_stock['Close'], nlags='auto')
print('Google closing price stationarity p-value: %.3f'%kpss_closing_price[1])
```

```
Google closing price stationarity p-value: 0.010
```

```
kpss_change_price = sm.tsa.stattools.kpss(google_stock['Close'].diff(1)[1:], nlags='auto')
print('Google closing price change stationarity p-value: %.3f'%kpss_change_price[1])
```

```
Google closing price change stationarity p-value: 0.100
```

What makes a pattern?

- Theoretically, a time series can be decomposed into:
  - Seasonality S_t
  - Trend T_t

Stationarity or not?

  - Remainder R_t

Noise and uncaptured pattern

Additive vs. Multiplicative

- Additive: When trend/seasonality is constant over time

$$y_t = S_t + T_t + R_t$$

- Multiplicative: varying trend/seasonality

$$y_t = S_t \times T_t \times R_t$$

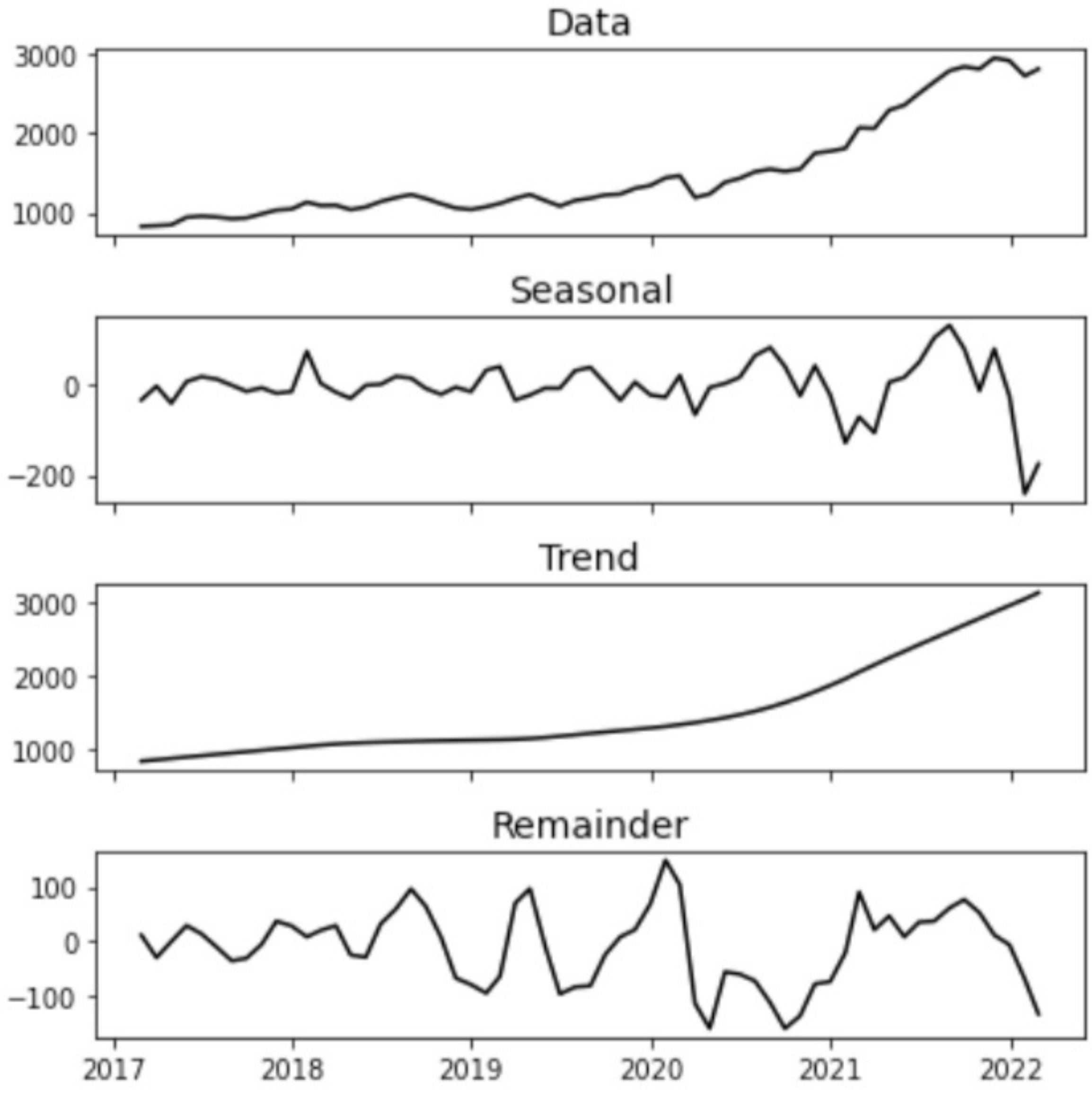
STL decomposition

- One popular approach but additive only
  - Are additive and multiplicative mutually exclusive?

$$log(y_t) = log(S_t) + log(T_t) + log(R_t)$$

- S: Seasonality
- T: Trend
- L: using LOESS (local regression)

STL decomposition: Example



Alright, now we talk about forecasting.

- What's the approach that comes to your mind if you:
  - Have historical data from time 1 to T;
  - Want to predict data at time T+1?

$$\hat{y_{t+1}} = y_t$$

Simple but…

The simple approach

- We could make use of the one-step before value for the next one

- But maybe the reliance on just one point is inaccurate.

But still...

- Hey! Let's add more: $$\hat{y_{t+1}} = \frac{\sum_{t=t_o}^{T} y_t}{T - t_o + 1}$$

Now enter simple exp. smoothing

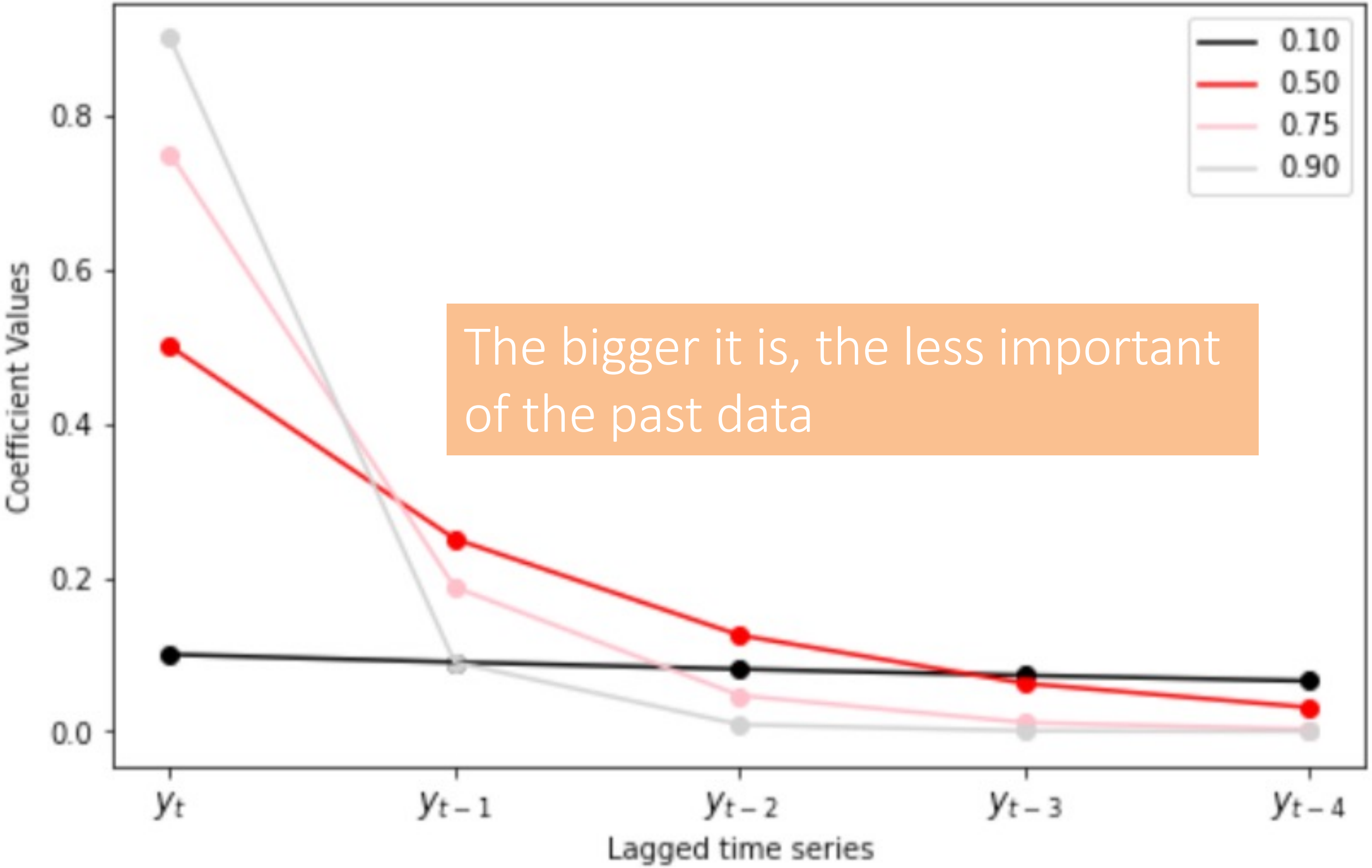
- Intuitively, the pattern which is further away is less important in the forecast

- Let's make more recent points more important:

$$\hat{y_{t+1}} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \ldots$$

Why α only? Because it's simple to infer…

The role of α



The bigger it is, the less important of the past data

The component form of this model

- Forecast equation:

$$\hat{y}_{t+1} = l_t,$$

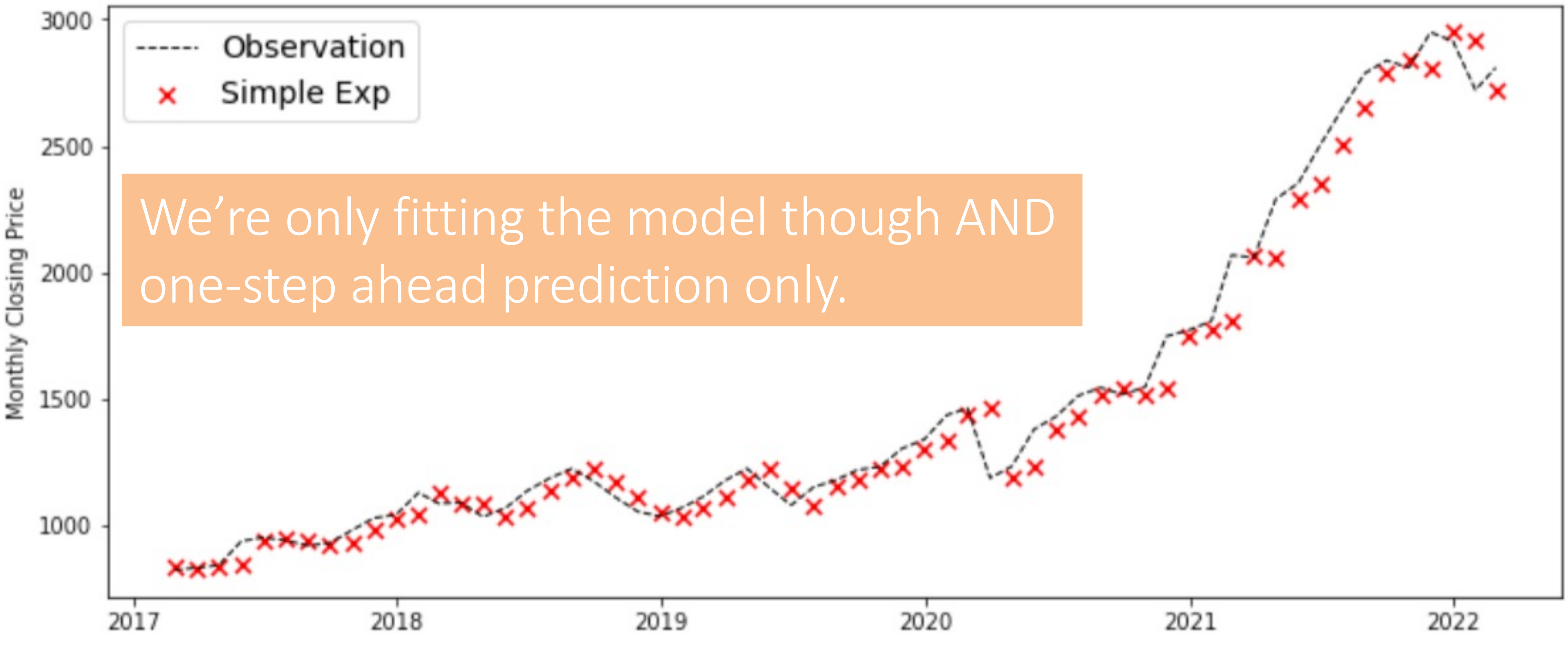
- Smoothing (level) equation:

$$l_t = \alpha y_t + (1 - \alpha) l_{t-1}$$

Math behind it (Optional)

$$\begin{aligned}
\hat{y_{t+1}} &= \alpha y_t + (1-\alpha) l_{t-1} \\
&= \alpha y_t + (1-\alpha)[\alpha y_{t-1} + (1-\alpha) l_{t-2}] \\
&= \alpha y_t + \alpha(1-\alpha) y_{t-1} + (1-\alpha)^2 [\alpha y_{t-2} + (1-\alpha) l_{t-3}] \\
&= \ldots \\
&= \sum_T \alpha (1-\alpha)^{t-T} y_T
\end{aligned}$$

Simple Exp. Smoothing: Example



We're only fitting the model though AND one-step ahead prediction only.

What's the issue?

- Flat prediction:

$$\hat{y_{t+h}} = \hat{y_{t+1}}; \quad h \geq 2$$

h: how far away we look into the future

- Solution: capturing the trend!

Holt's Method

- We need more stuff to train the model now:
  - Forecast

h: how far away we look into the future
This is the term that makes our prediction no longer flat!

$$\hat{y_{t+h}} = l_t + hb_t$$

  - Smoothing

$$l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1})$$

  - Trend

b_t: trend (or slope)

$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

Mechanism behind Holt's Method

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$l_{t-1} + b_{t-1} = \hat{y}_t$$

$$l_t = \alpha y_t + (1 - \alpha)\hat{y}_t$$

Isn't it the same as what we have before?

Mechanism behind Holt's Method

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1}$$

$$\begin{aligned} b_t &= \beta^*(l_t - l_{t-1}) + (1 - \beta^*)[\beta^*(l_{t-1} - l_{t-2}) + (1 - \beta^*)b_{t-2}] \\ &= \beta^*(l_t - l_{t-1}) + \beta^*(1 - \beta^*)(l_{t-1} - l_{t-2}) + (1 - \beta^*)^2 b_{t-2} \end{aligned}$$

Isn't this familiar to us?

$$b_t = \sum_T \beta^*(1 - \beta^*)^{t-T}(l_{t-T} - l_{t-T-1})$$

Constant or Changing Trend?

- Does trend last forever without changing at all?

Trend effect is weakened over time

- Damping parameter $\phi \in (0, 1)$
  - Forecast $\hat{y_{t+h}} = l_t + (\phi + \phi^2 + \ldots + \phi^h)b_t$
  - Smoothing $l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1})$
  - Trend $b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)\phi b_{t-1}$

Recall that there are two components

- We have trend now. What’s next?

- Seasonality!
  - Sales of certain products may exhibit seasonal patterns
  - Temperature may also feature seasonal dynamics

Holt-Winter's (or Winter's) Method

- As with the decomposition, there are two forms of Winter's method as well
  - Additive vs. Multiplicative
  - For simplicity, we talk more details about additive but skip multiplicative.
  - In using Python, it is easy to change from one to the other.

Winter's Method: Additive

- Forecast

$$\hat{y_{t+h}} = l_t + hb_t + s_{t+h-m}$$

- Smoothing

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

- Trend

$$b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$$

- Seasonality

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$$

Example: quarterly data $m = 4$

The most recent and same season

$$\hat{y_{t+1}} = l_t + b_t + s_{t-3}$$

$$l_t = \alpha(y_t - s_{t-4}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$l_{t-1} + b_{t-1} = \hat{y}_t - s_{t-4}$$

$$l_t = \alpha(y_t - s_{t-4}) + (1-\alpha)(\hat{y}_t - s_{t-4})$$

Example: quarterly data $m = 4$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1-\gamma)s_{t-4}$$

$$y_t - l_{t-1} - b_{t-1} = y_t - (\hat{y}_t - s_{t-4})$$

$$s_t = \gamma(y_t - \hat{y}_t + s_{t-4}) + (1-\gamma)s_{t-4}$$
$$= \gamma(y_t - \hat{y}_t) + s_{t-4}$$

Last seasonality plus
weighted forecast error

CityU

Multiplicative counterpart (optional)

- See the notebook:

Forecast equation: $\hat{y_{t+h}} = (l_t + hb_t)s_{t+h-m}$

Smoothing (level) equation: $l_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1})$

Trend equation: $b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1}$

Seasonality equation: $s_t = \gamma \frac{y_t}{l_{t-1} + b_{t-1}} + (1-\gamma)s_{t-m}$

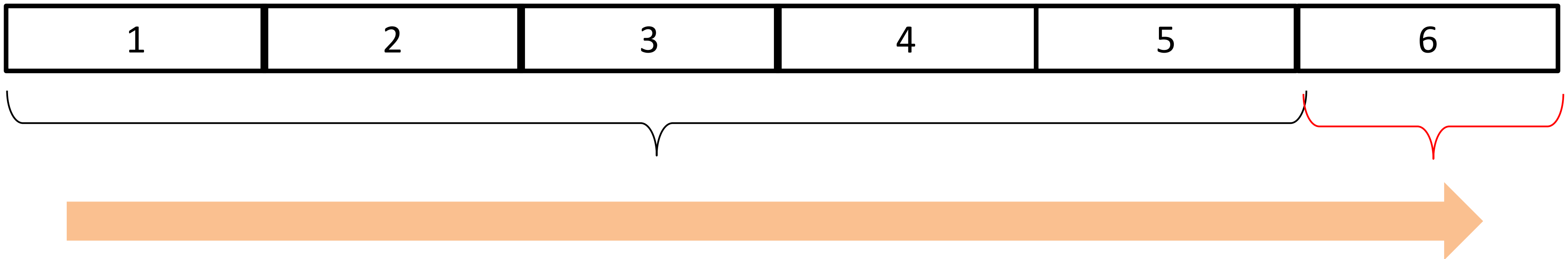
Forecasting evaluation

- What does it make a good prediction?

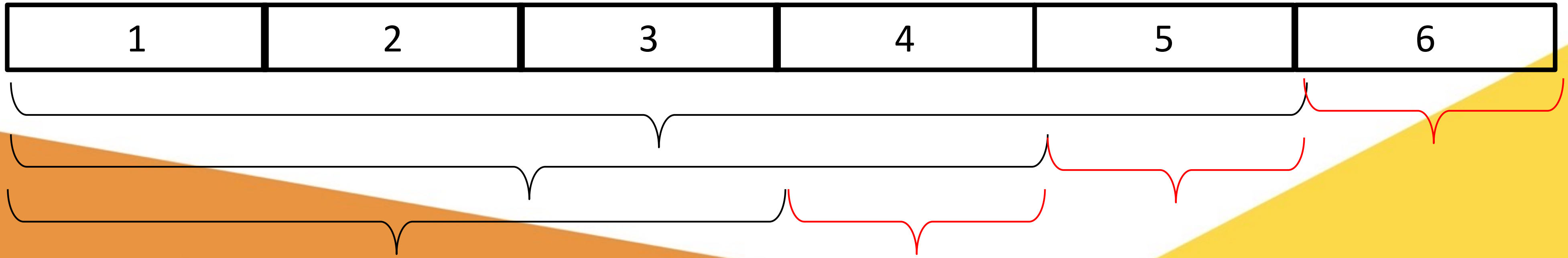
- Does cross validation work?

Data partitioning: Order Matters!

- Fixed partitioning

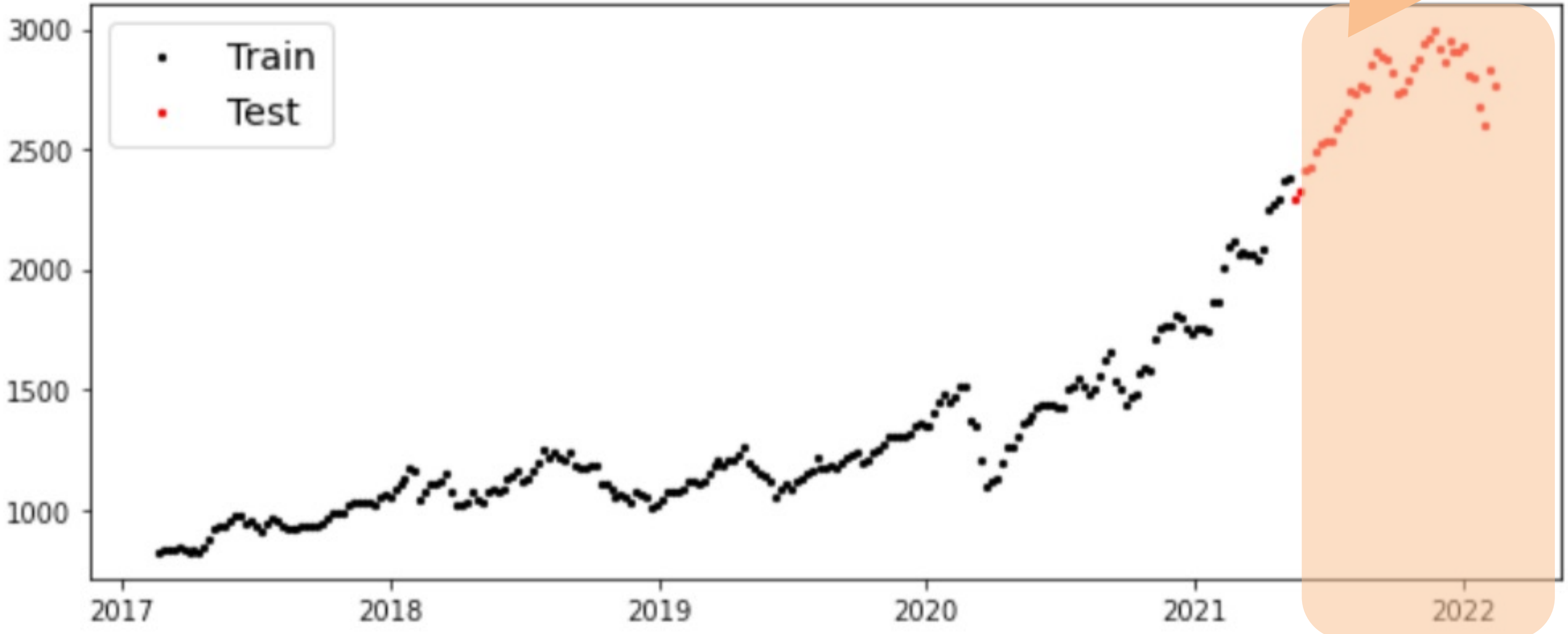


- Roll forward

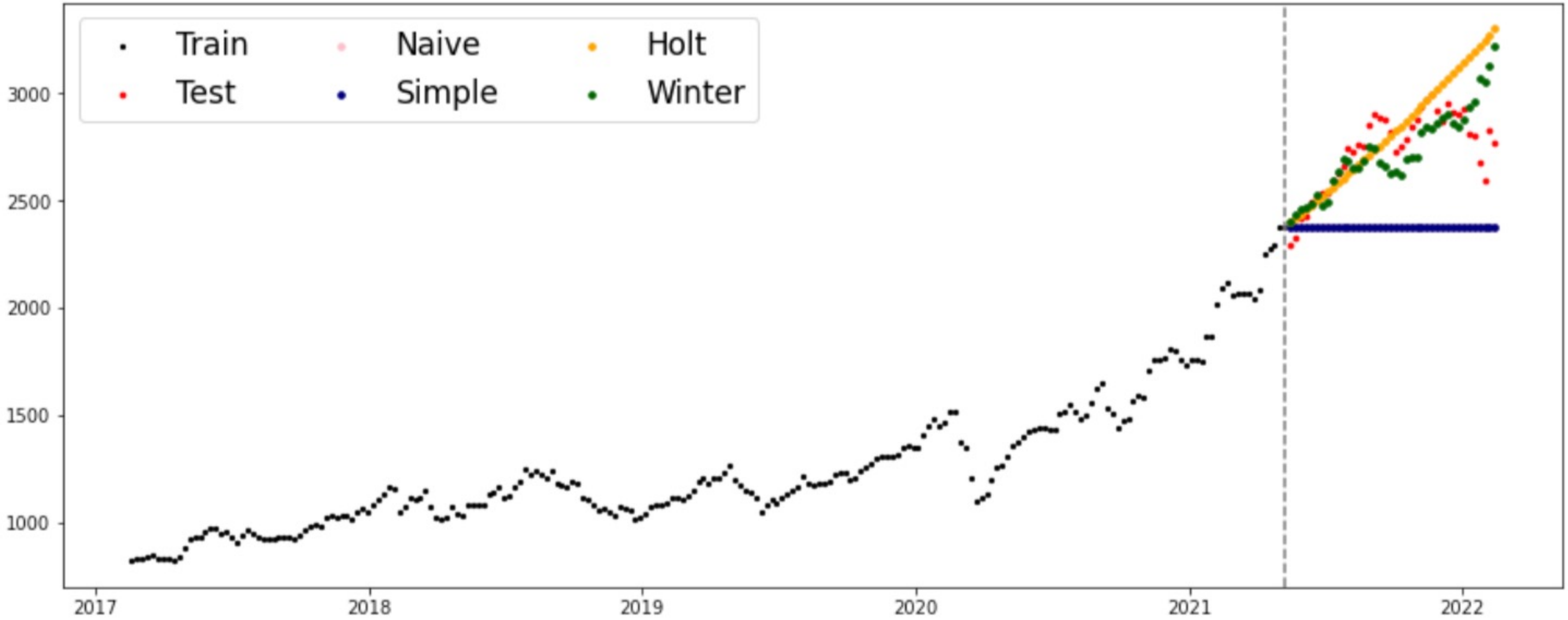


Fixed partitioning

One-time split with:
- Training set as the history
- Testing set as the future



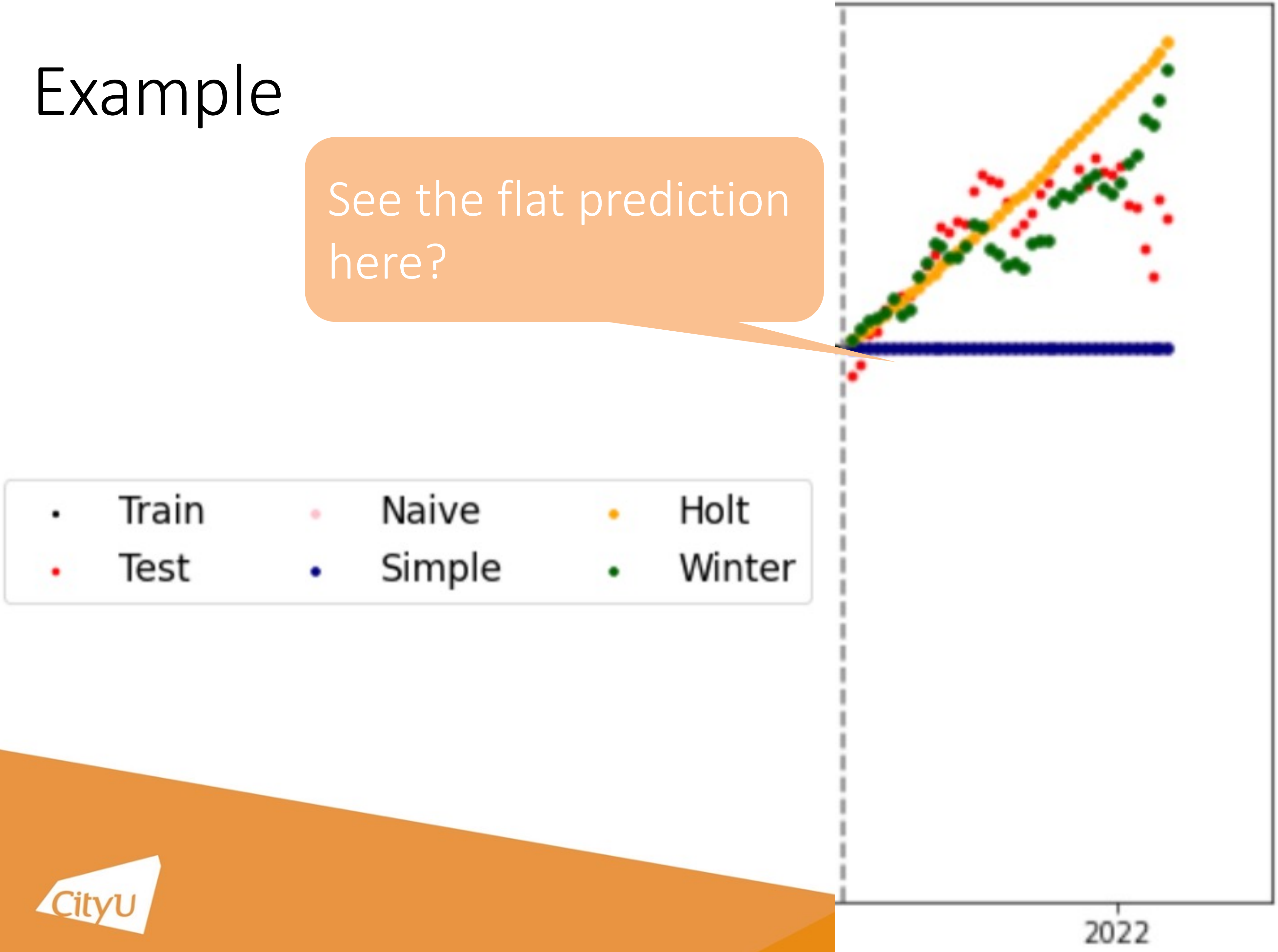
Example



Train
Test
Naive
Simple
Holt
Winter
3000
2500
2000
1500
1000
2017
2018
2019
2020
2021
2022

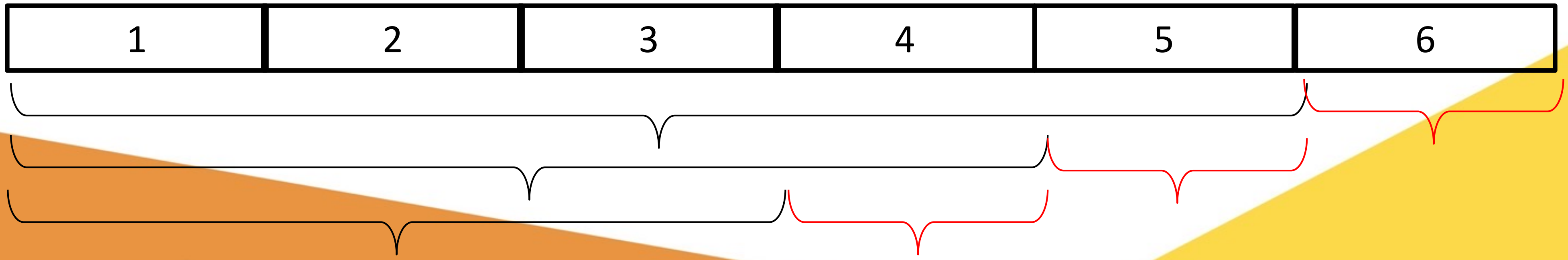
Example

See the flat prediction here?

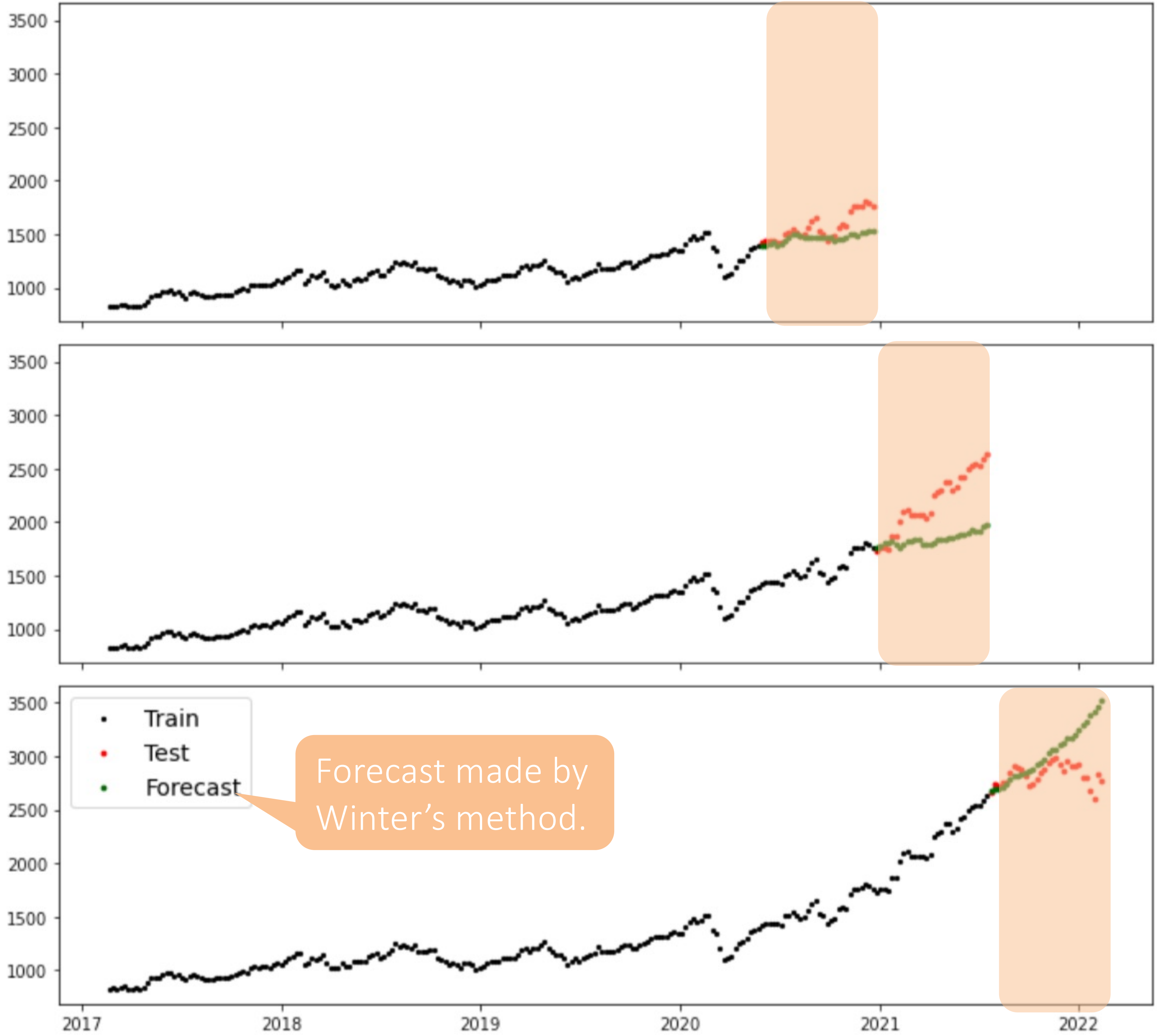


Roll forward

- The time- series version of cross validation
- Iteratively shift the time series splitting for training and testing



Example


3500
3000
2500
2000
1500
1000
Train
Test
Forecast
Forecast made by Winter's method.
2017
2018
2019
2020
2021
2022

A summary

- Time series features/characteristics
- Forecasts via exponential smoothing
  - What time series components are we thinking about?

A summary of smoothing

Trend Component	Seasonal Component		
	N	**A**	**M**
	(None)	**(Additive)**	**(Multiplicative)**
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)

From Table 8.5 of Hyndman and Athanasopoulos (2018)

Suggested References

- Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3.

- Business analytics using forecasting: https://youtube.com/playlist?list=PLoK4oIB1jeK0LHLbZW3DTT05e4srDYxFq

香港城市大學
City University of Hong Kong

The End

專業 創新 胸懷全球
Professional · Creative
For The World