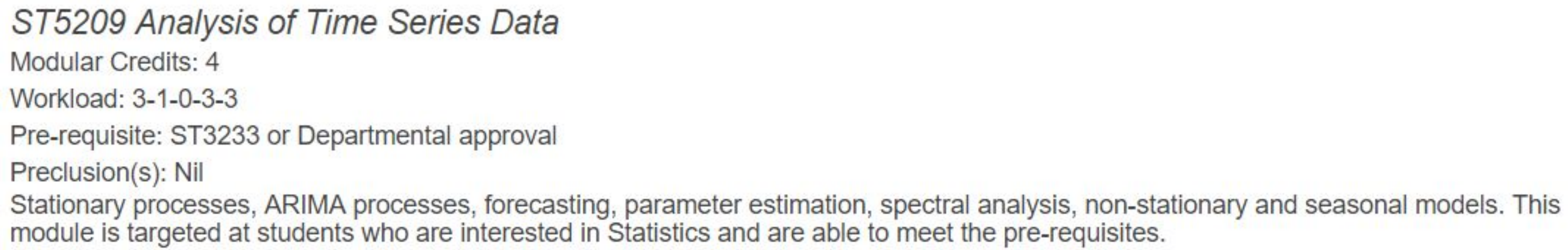
Tutorial 6
Time Series Analysis

Agenda

- Understand Time Series, Stationarity and Non-stationarity
- Time Series Decomposition: Trend, Seasonality, Residual
- Statistical tests on (non-)stationarity
- ARIMA model (Autoregressive Integrated Moving Average)
- Discussion about Programming Assignment 6
  - Time Series decomposition
  - Fit with ARIMA model and do prediction
- Python Implementation
  - Statsmodels

If you are interested in Time Series:

Register or Audit one module in Stats department:



ST5209 Analysis of Time Series Data

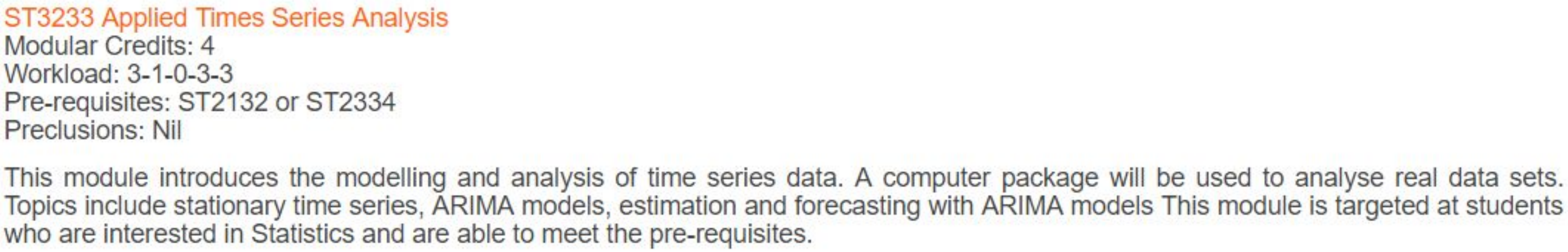
Modular Credits: 4

Workload: 3-1-0-3-3

Pre-requisite: ST3233 or Departmental approval

Preclusion(s): Nil

Stationary processes, ARIMA processes, forecasting, parameter estimation, spectral analysis, non-stationary and seasonal models. This module is targeted at students who are interested in Statistics and are able to meet the pre-requisites.

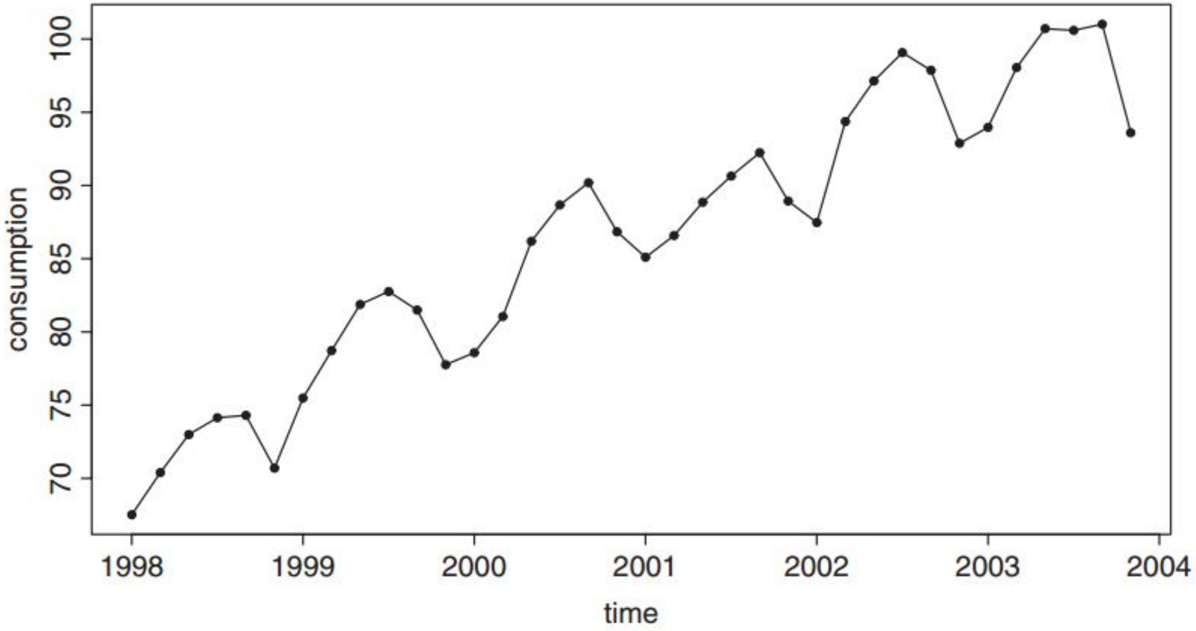


ST3233 Applied Times Series Analysis
Modular Credits: 4
Workload: 3-1-0-3-3
Pre-requisites: ST2132 or ST2334
Preclusions: Nil

This module introduces the modelling and analysis of time series data. A computer package will be used to analyse real data sets. Topics include stationary time series, ARIMA models, estimation and forecasting with ARIMA models This module is targeted at students who are interested in Statistics and are able to meet the pre-requisites.

Time series

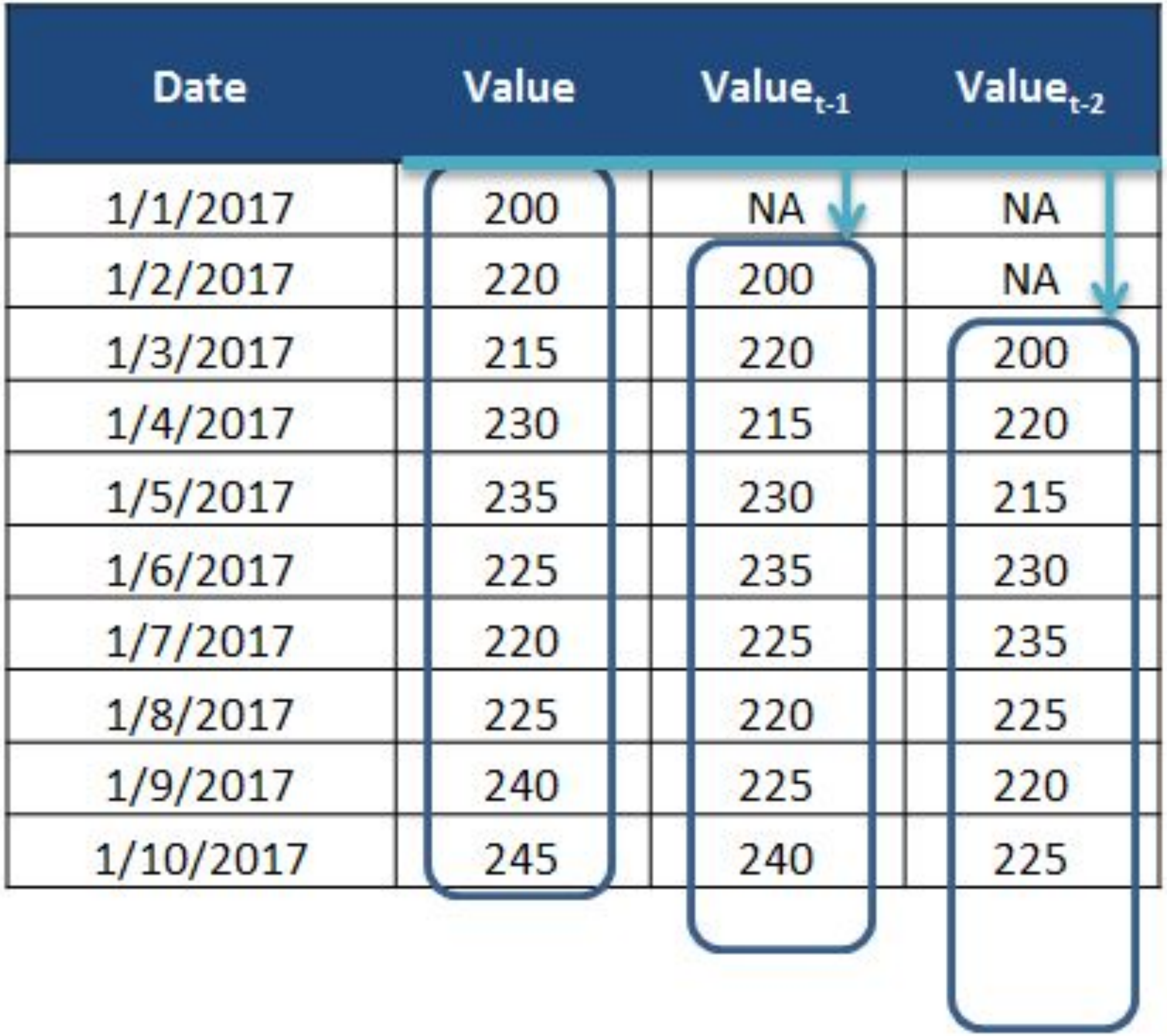
- A *time series* is a sequence $\{x_t\}$ of values assumed by a quantity of interest that can be measured at specific time periods *t*.
- *t* can be hours, days, weeks, months, quarters or years.



Time series of electricity consumption in an Italian region over 36 two-month periods

Time series

-



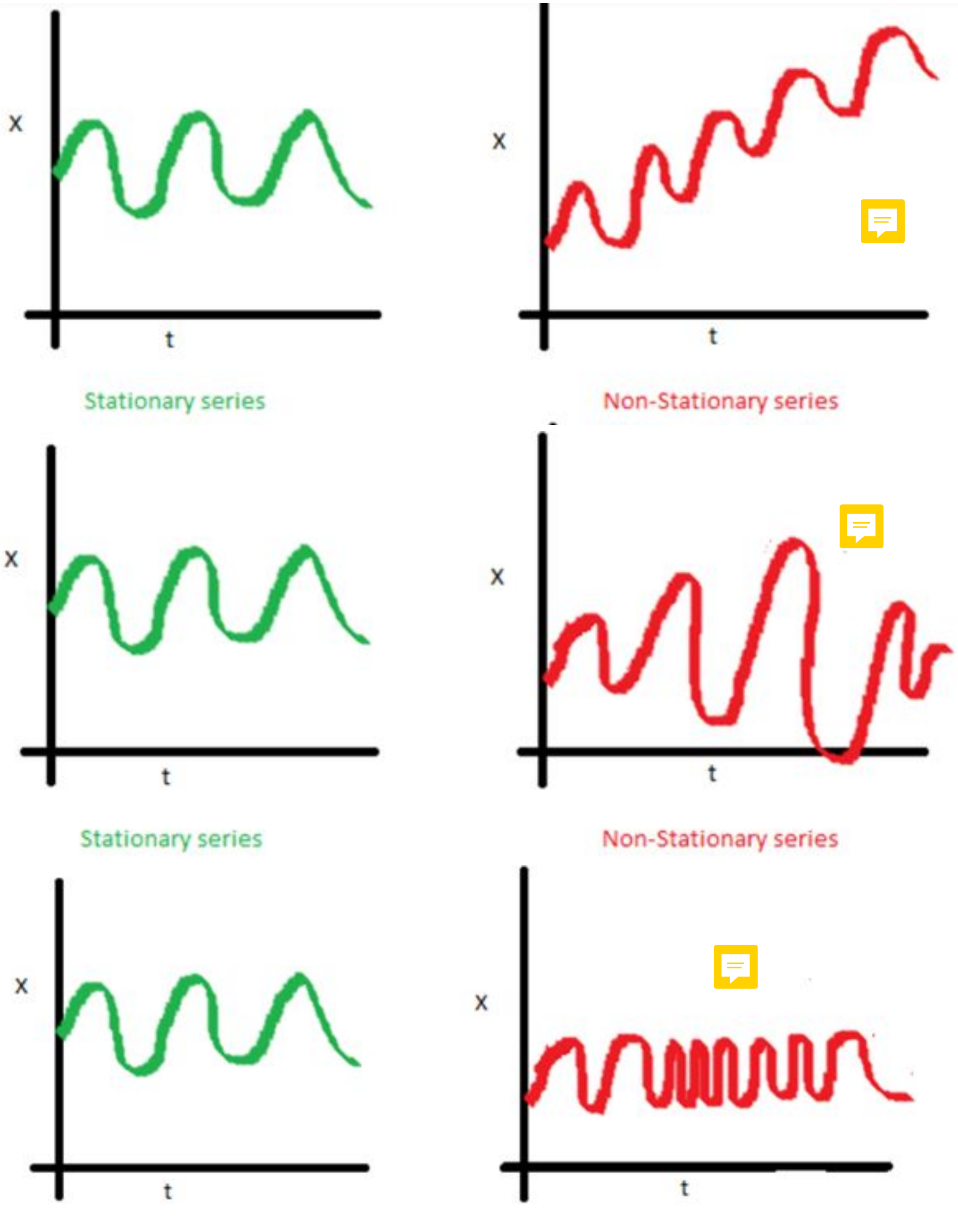
Date	Value	Value$_{t-1}$	Value$_{t-2}$
1/1/2017	200	NA	NA
1/2/2017	220	200	NA
1/3/2017	215	220	200
1/4/2017	230	215	220
1/5/2017	235	230	215
1/6/2017	225	235	230
1/7/2017	220	225	235
1/8/2017	225	220	225
1/9/2017	240	225	220
1/10/2017	245	240	225

Time series-Stationary

- A *stationary* time series is one whose statistical properties such as mean, variance, autocovariance/autocorrelation, etc. are all constant over time.
- Mean:

  $E(x_t)$ is not conditional on t
- Variance:

  $Var(x_t)$ is not conditional on t
- Autocovariance (k-order):
  $Cov(x_t, x_{t-k})$ is only conditional on k, not t



Stationary series

Non-Stationary series

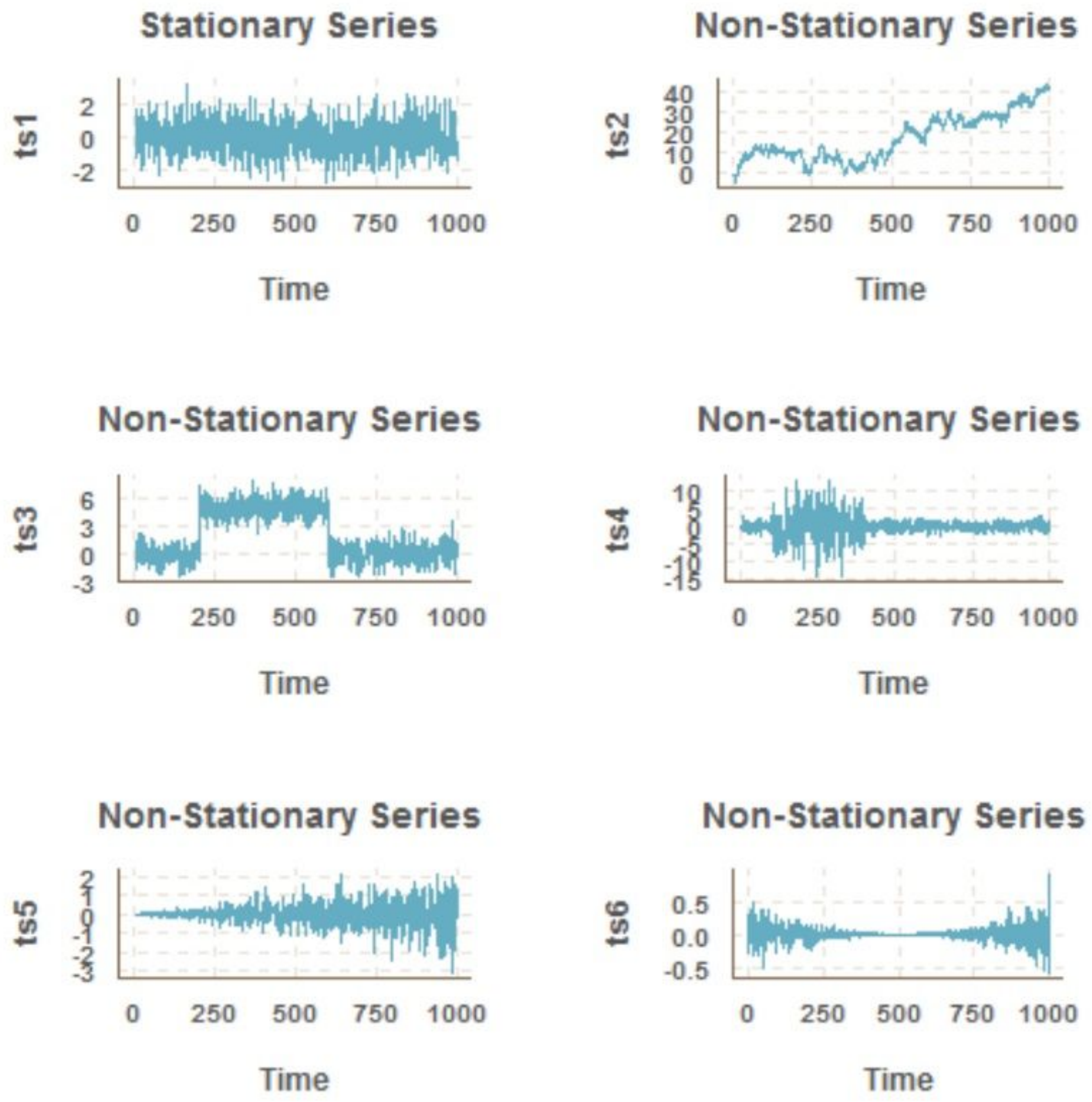
Stationary series

Non-Stationary series

Stationary series

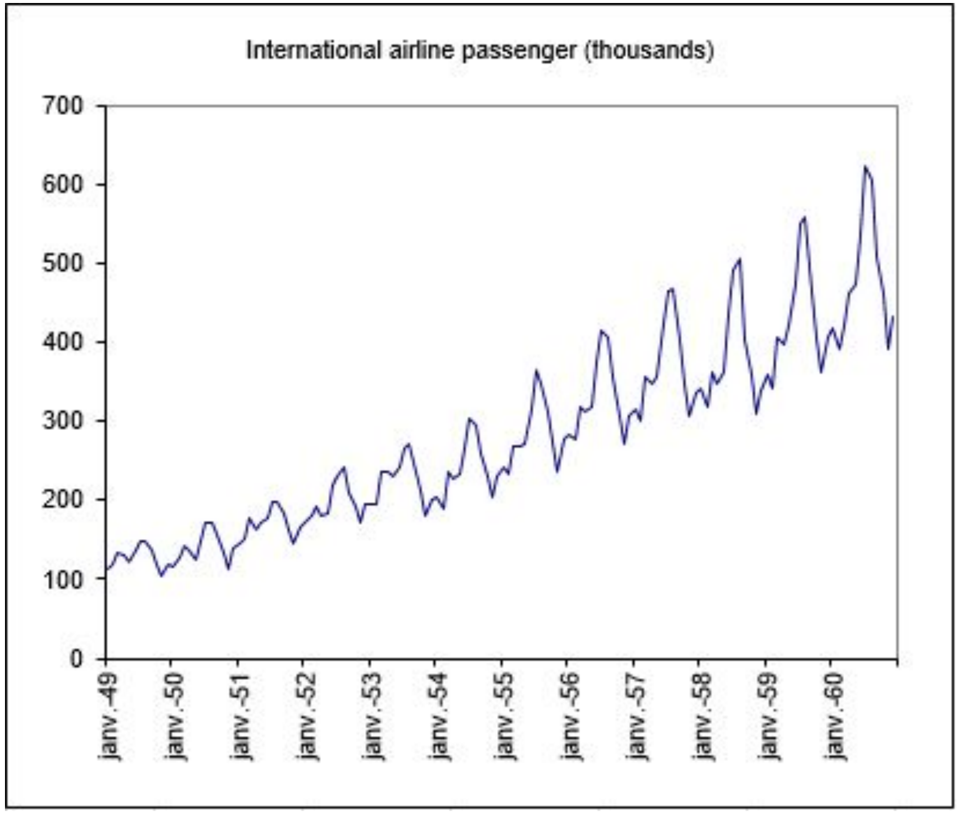
Non-Stationary series

Time series -Stationary

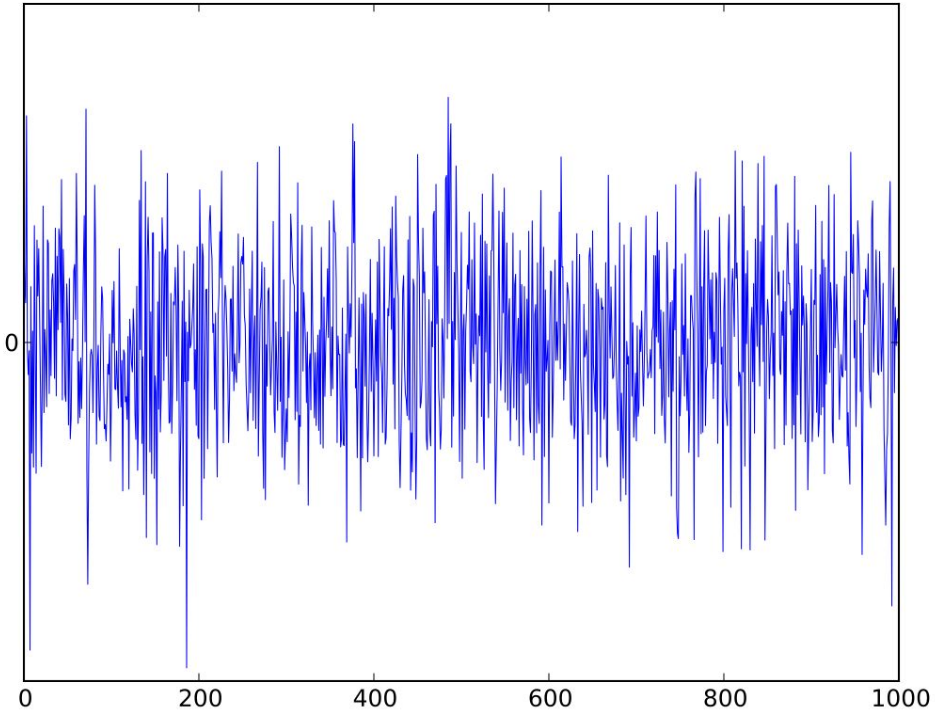


Stationarity-Example

Are they stationary and non-stationary time series?



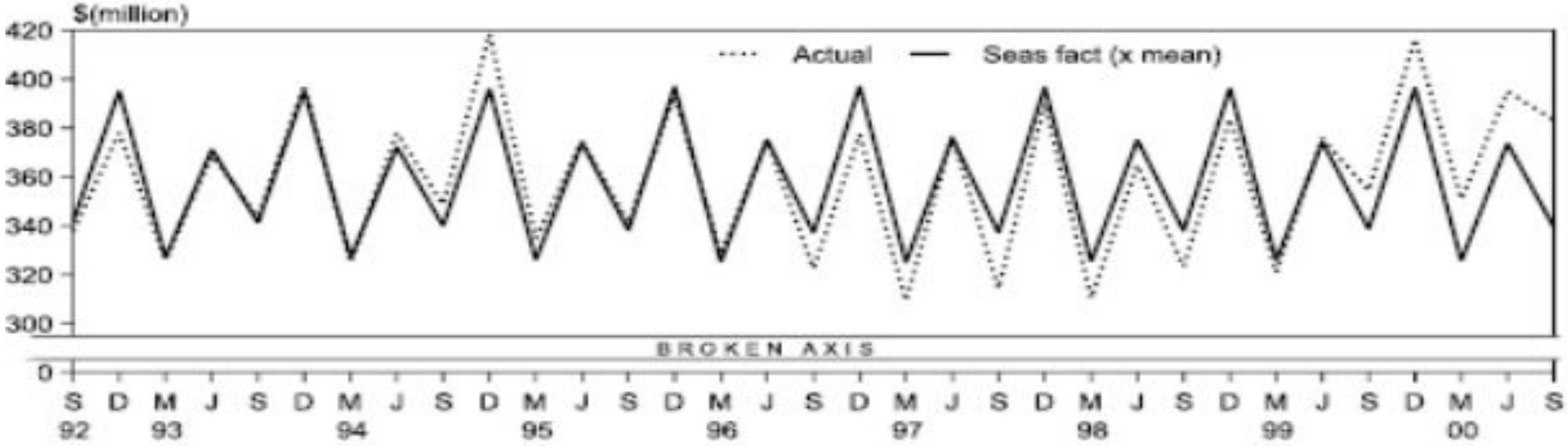
A



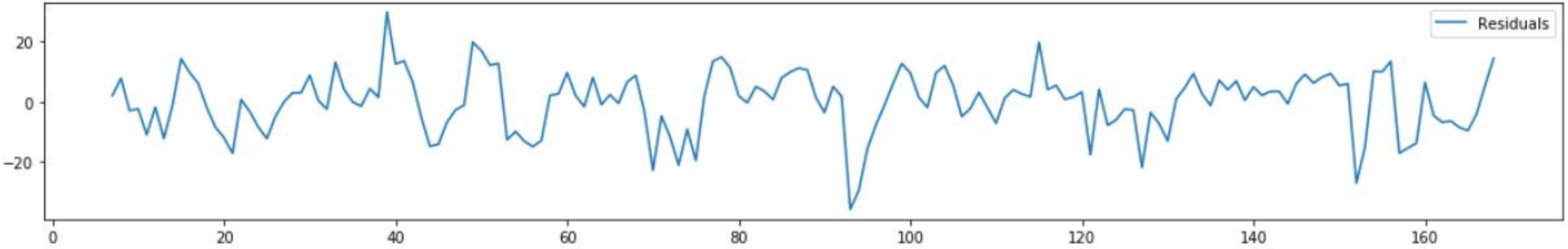
B

Stationarity-Example

Are they stationary and non-stationary time series?



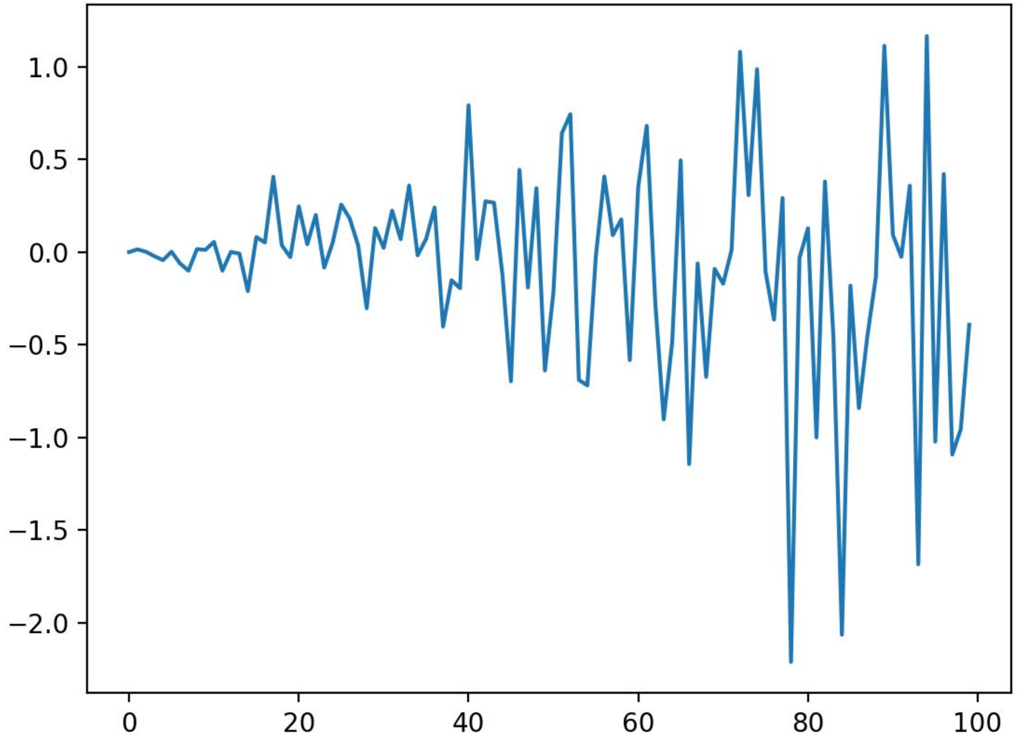
C



D

Stationarity-Example

Are they stationary and non-stationary time series?



E

Stationarity in Time Series Analysis

- We usually transform *nonstationary* time series to *stationary* time series first, and fit with time series forecasting models.

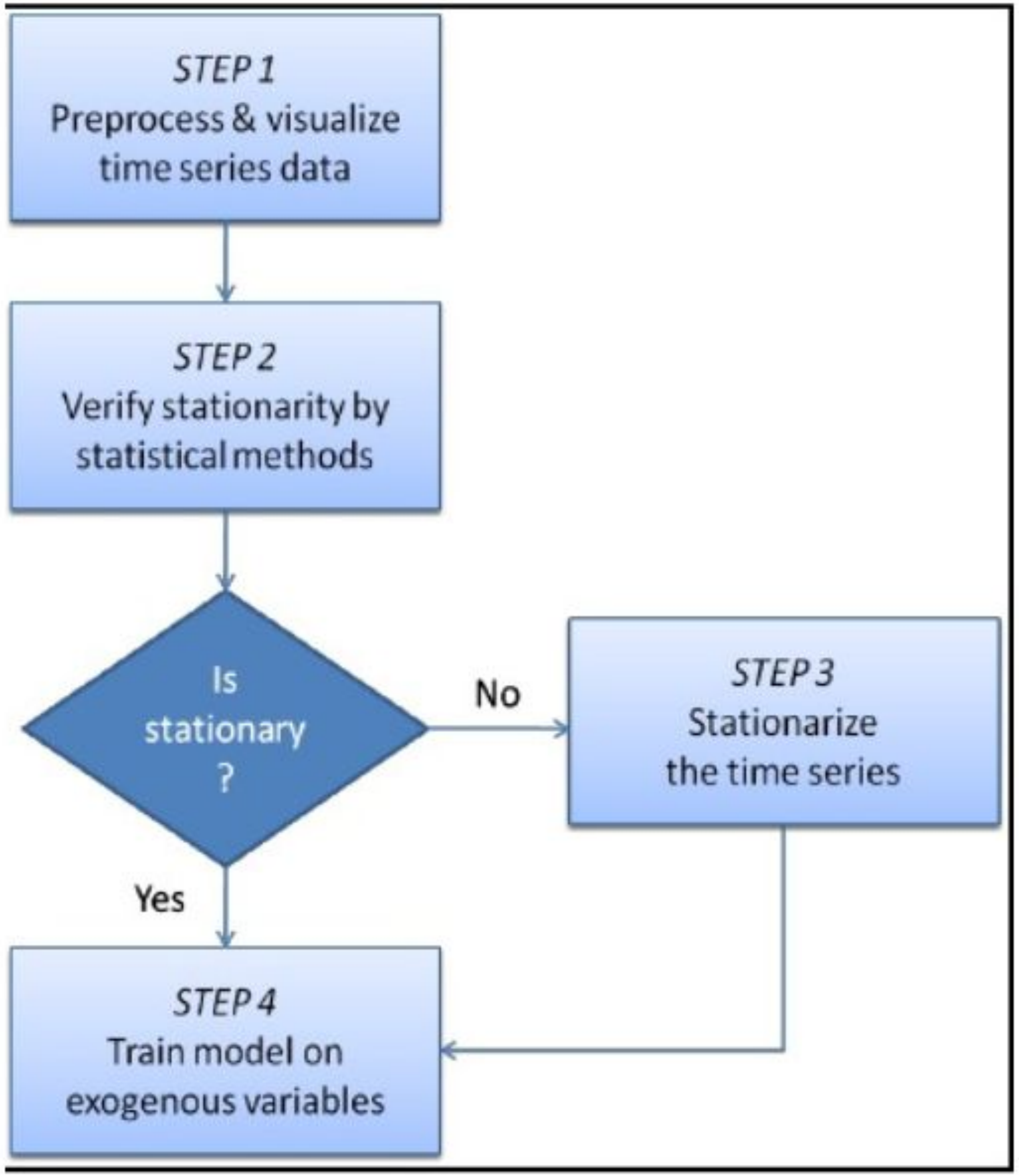

STEP 1
Preprocess & visualize time series data
STEP 2
Verify stationarity by statistical methods
Is stationary ?
No
STEP 3
Stationarize the time series
Yes
STEP 4
Train model on exogenous variables

Figure 2.1: A generic methodology of time series analysis

Stationarity in Time Series Analysis

Why do we need to transform *non-stationary* to *stationary*?

- Most statistical forecasting methods are based on the assumption that the time series can be rendered approximately stationary (i.e., "stationarized") through the use of mathematical transformations.
- A stationarized series is relatively easy to predict: you simply predict that its statistical properties will be the same in the future as they have been in the past

The predictions for the stationarized series can be "untransformed" by reversing whatever mathematical transformations were previously used, to obtain predictions for the original series.

Thus, finding the sequence of transformations needed to stationarize a time series often provides important clues in the search for an appropriate forecasting model.

Time Series Analysis – Decomposition

- A time series {Yt} can be expressed as a combination of its components

$$Yt = g(Mt, Qt, \varepsilon t)$$

- where *g* represents an appropriate function to be selected.

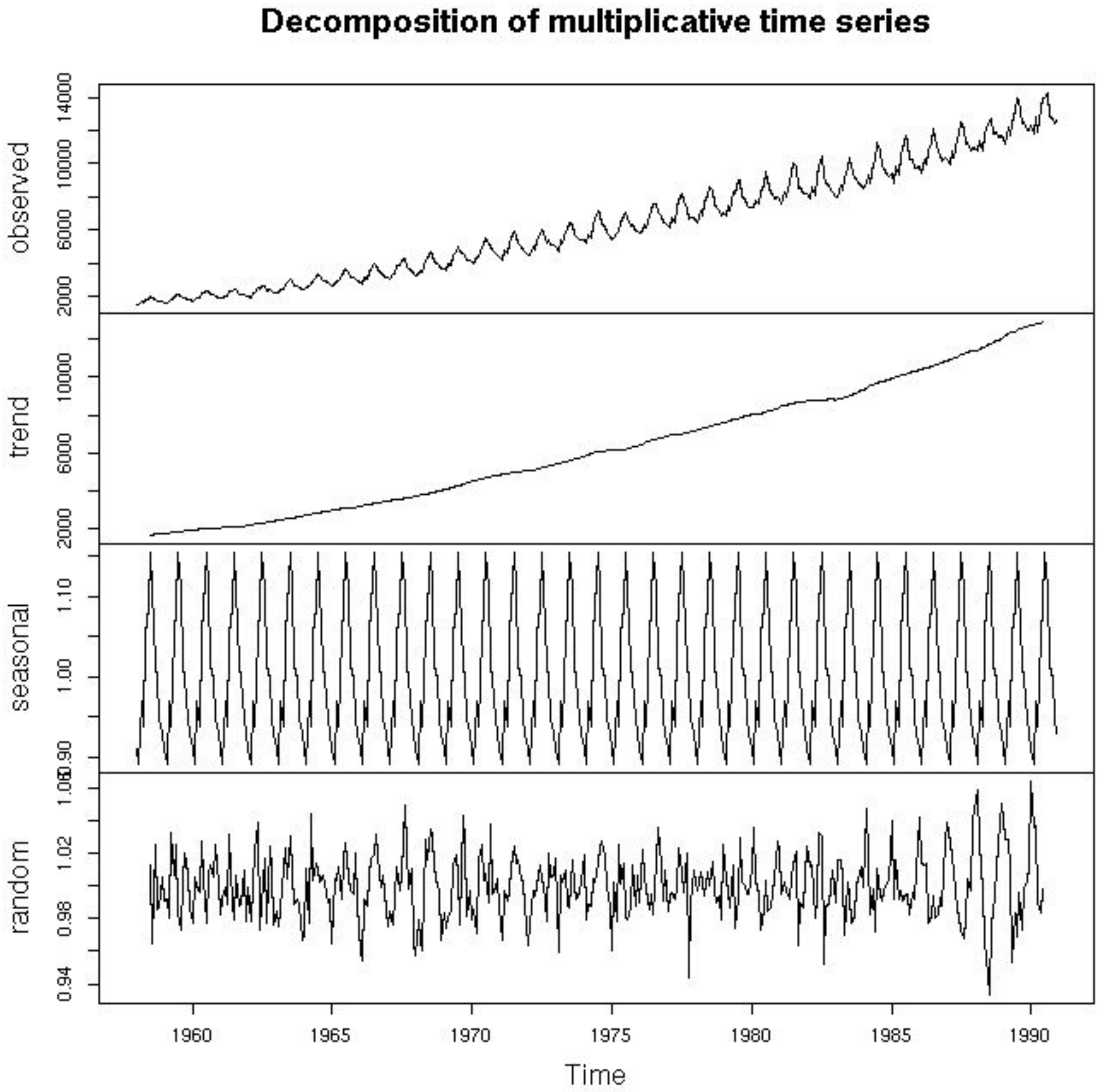
- It can be additive

$$Y_t = M_t + Q_t + \varepsilon_t,$$

- or multiplicative.

$$Y_t = M_t \times Q_t \times \varepsilon_t.$$

Time Series Analysis – Decomposition



Time Series Analysis – Decomposition

- **Trend.** A long-term *trend* component, denoted by *Mt*, describes the average behavior of a time series over time, and it can be i**ncreasing, decreasing or stationary.**
- **Seasonality.** The *seasonality* component, denoted by *Qt*, is the result of wavelike short-term fluctuations of regular frequency that appear in the values of a time series, for example corresponding to days of the week, or to months or quarters of the year.
- **Random noise.**

Time Series Analysis

- When your inputs are time series…

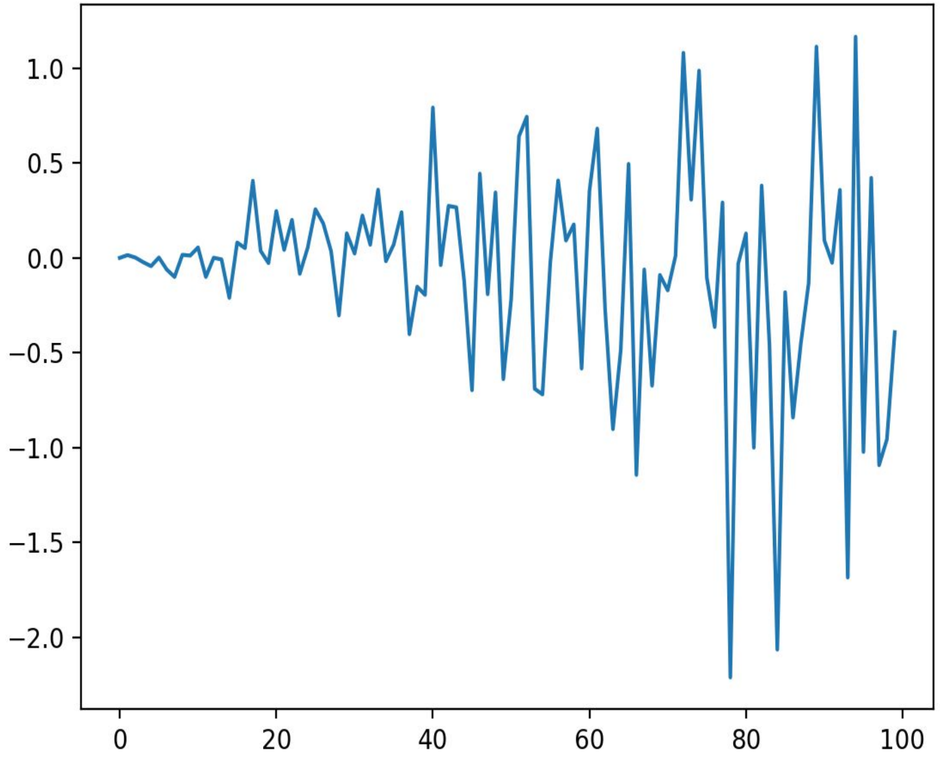
- When your outcome is a time series…
  - → need to remove trend and seasonality

- When you make prediction based on a univariate time series…
  - Reverse your transformations

First of all, What if variance is changing?

- Simple transformation on original data

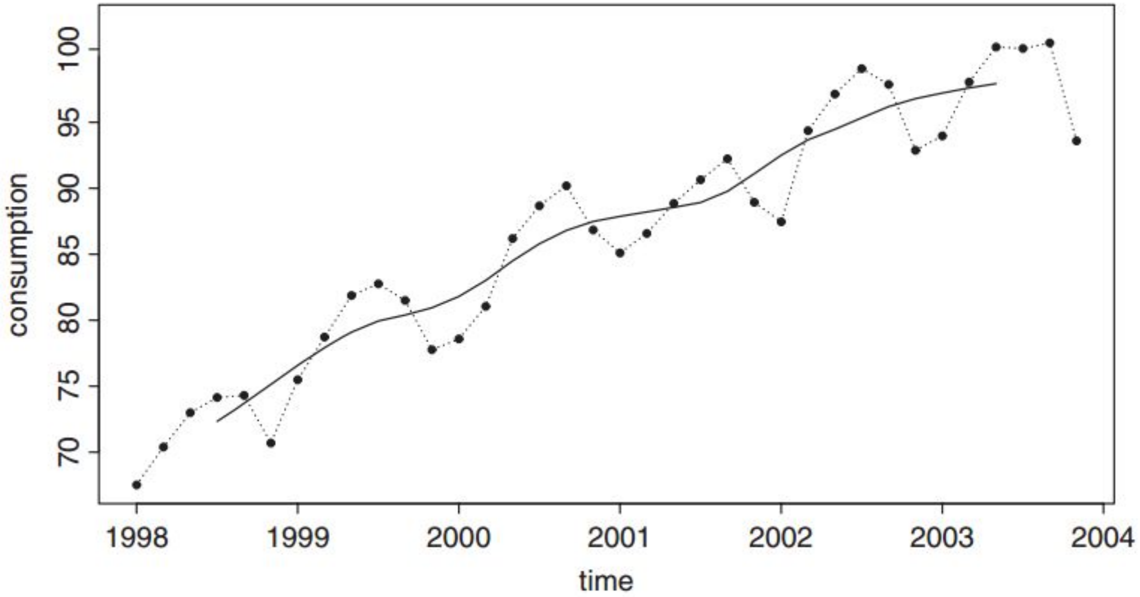
- For non-constant variance, taking the **logarithm** or **square root** of the series may stabilize the variance.
- For negative data, you can add a suitable constant to make all the data positive before applying the transformation. This constant can then be subtracted from the model to obtain predicted (i.e., the fitted) values and forecasts for future points.



https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc442.htm

Removal of trend component

- Method 1. Moving average smoothing
  Rolling time window: h



Moving average for the time series of electricity consumption

Odd

$$m_t(h) = \frac{y_{t+(h-1)/2} + y_{t+(h-1)/2-1} + \cdots + y_{t-(h-1)/2}}{h}.$$

Even

$$m_t(h) = \frac{y_{t+h/2} + y_{t+h/2-1} + \cdots + y_{t-h/2+1}}{2h} + \frac{y_{t+h/2-1} + y_{t+h/2-2} + \cdots + y_{t-h/2}}{2h},$$

A seasonality of length L, h = L

After moving average: ***Mt* ≈ *m*$_t$*(L)***

Can also be used for prediction

$$f_{t+1} = \frac{y_t + y_{t-1} + \cdots + y_{t-h+1}}{h},$$

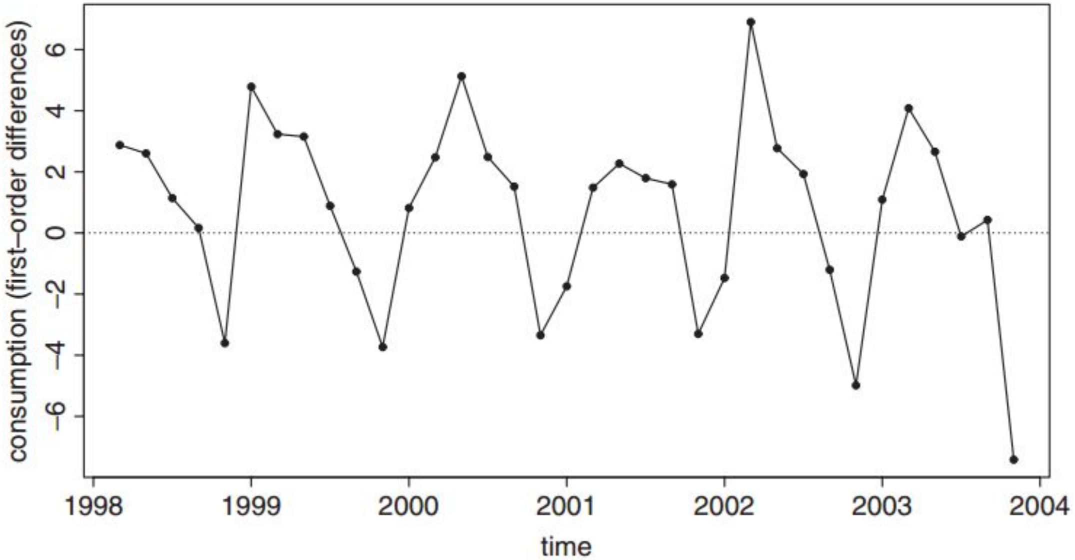
Removal of trend component

- Method 2. Differencing (e.g., first differencing)

successive differences between adjacent values of the time series

$$D_t = Y_t - Y_{t-1}, \quad t \geq 2.$$

Order =1



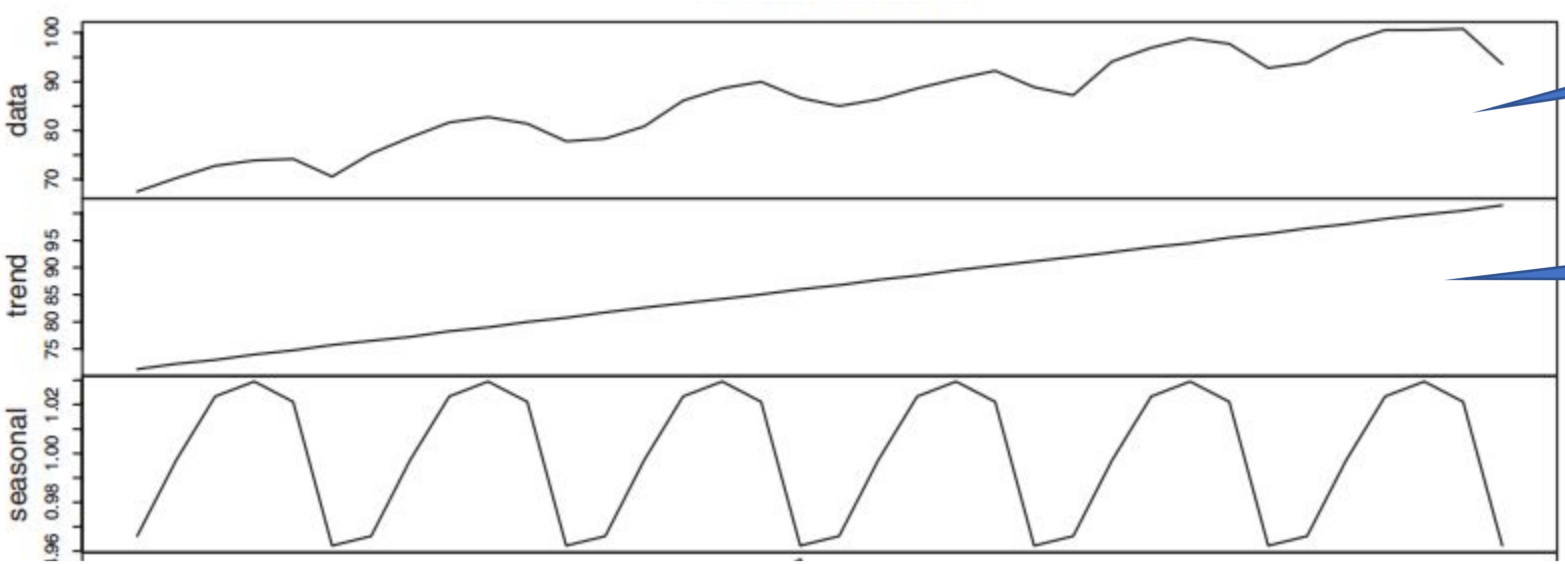
Although you can difference the data more than once, one difference is usually sufficient.

Removal of trend component

- Method 3. Trend curve

- A regression curve (linear, quadratic, exponential, logarithmic) that explains the values of the time series as a function of the time period, which plays the role of predictor variable.
- We can fit regression curve to the data and then model the residuals from that fit.

Mt = a + b*t + residual



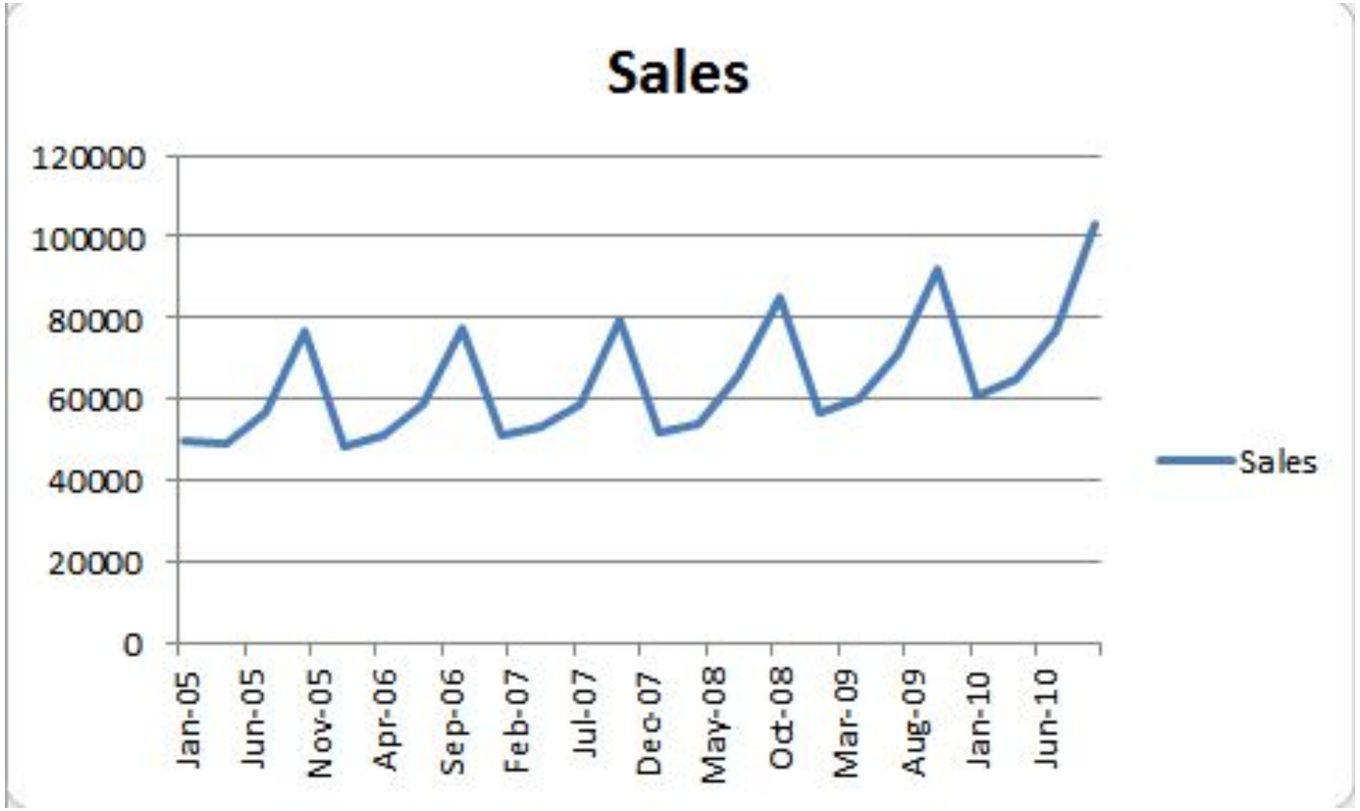
Initial data

Trend component

Remainder (e.g., including seasonality)

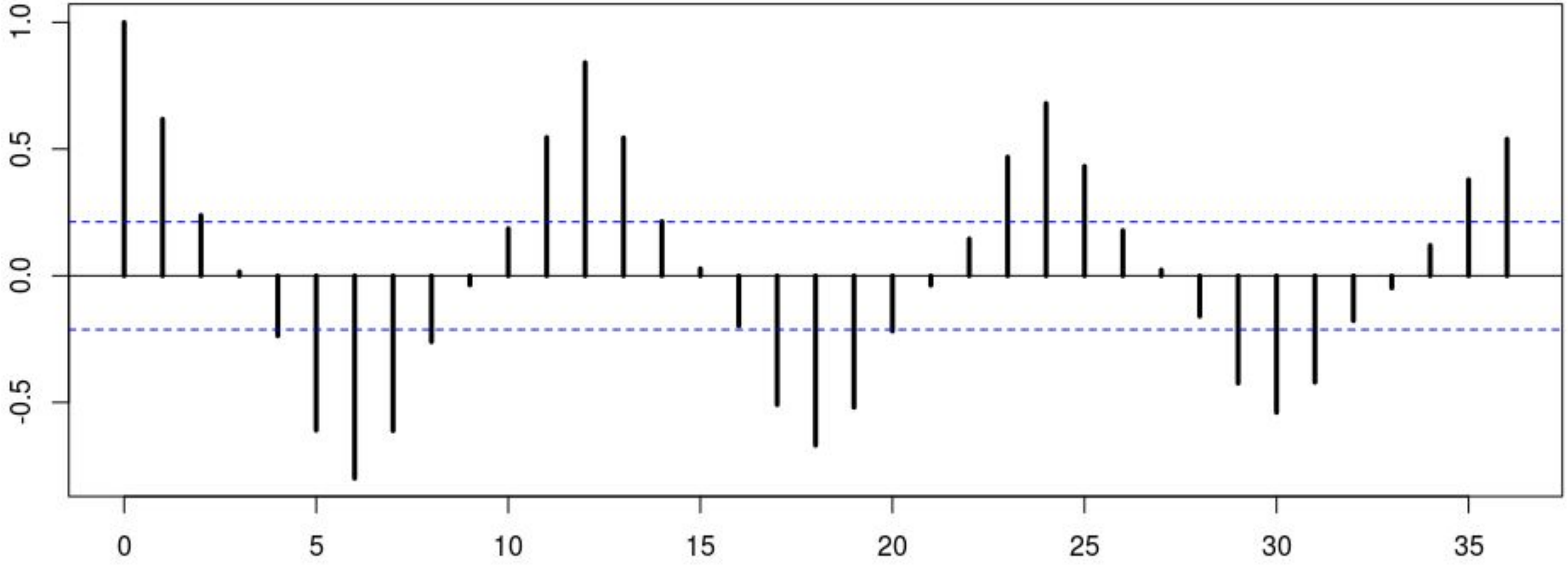
Seasonality

- Identifying Seasonality
  - Observing from time series plots



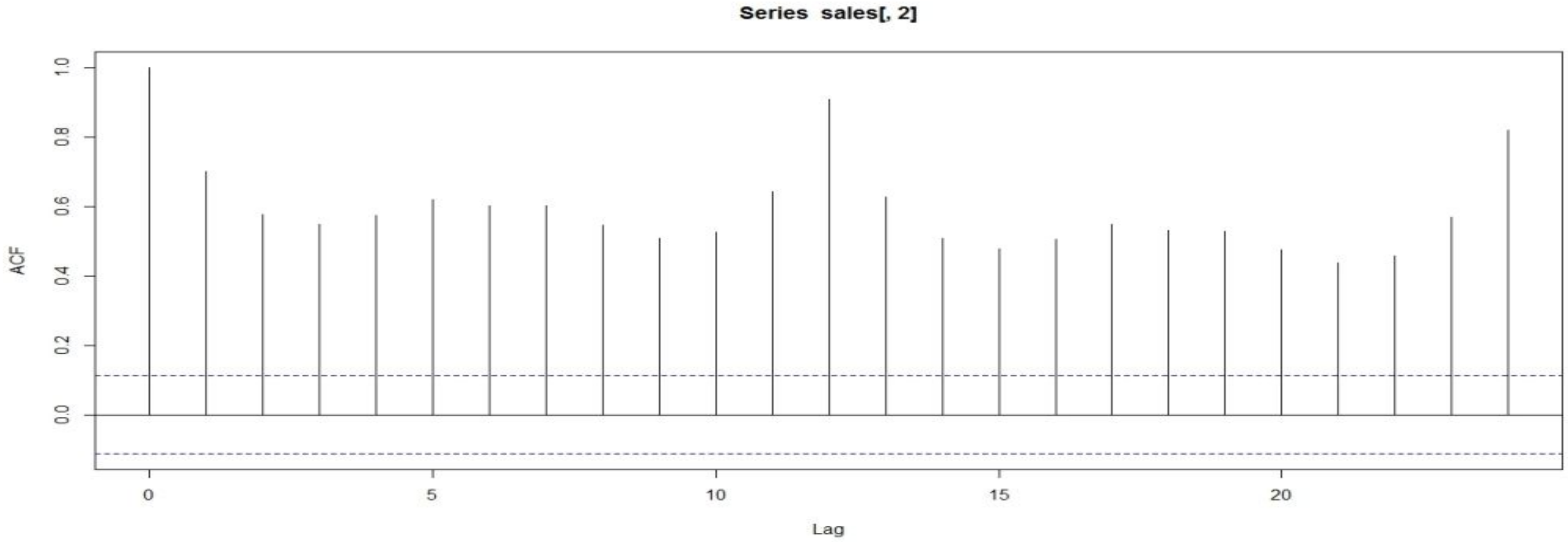
Seasonality

- Identifying Seasonality
  - Observing from time series plots
  - Observing from autocorrelation plot (ACF)



Seasonality

- Identifying Seasonality
  - Observing from time series plots
  - Observing from autocorrelation plot (ACF)



Removal of seasonality

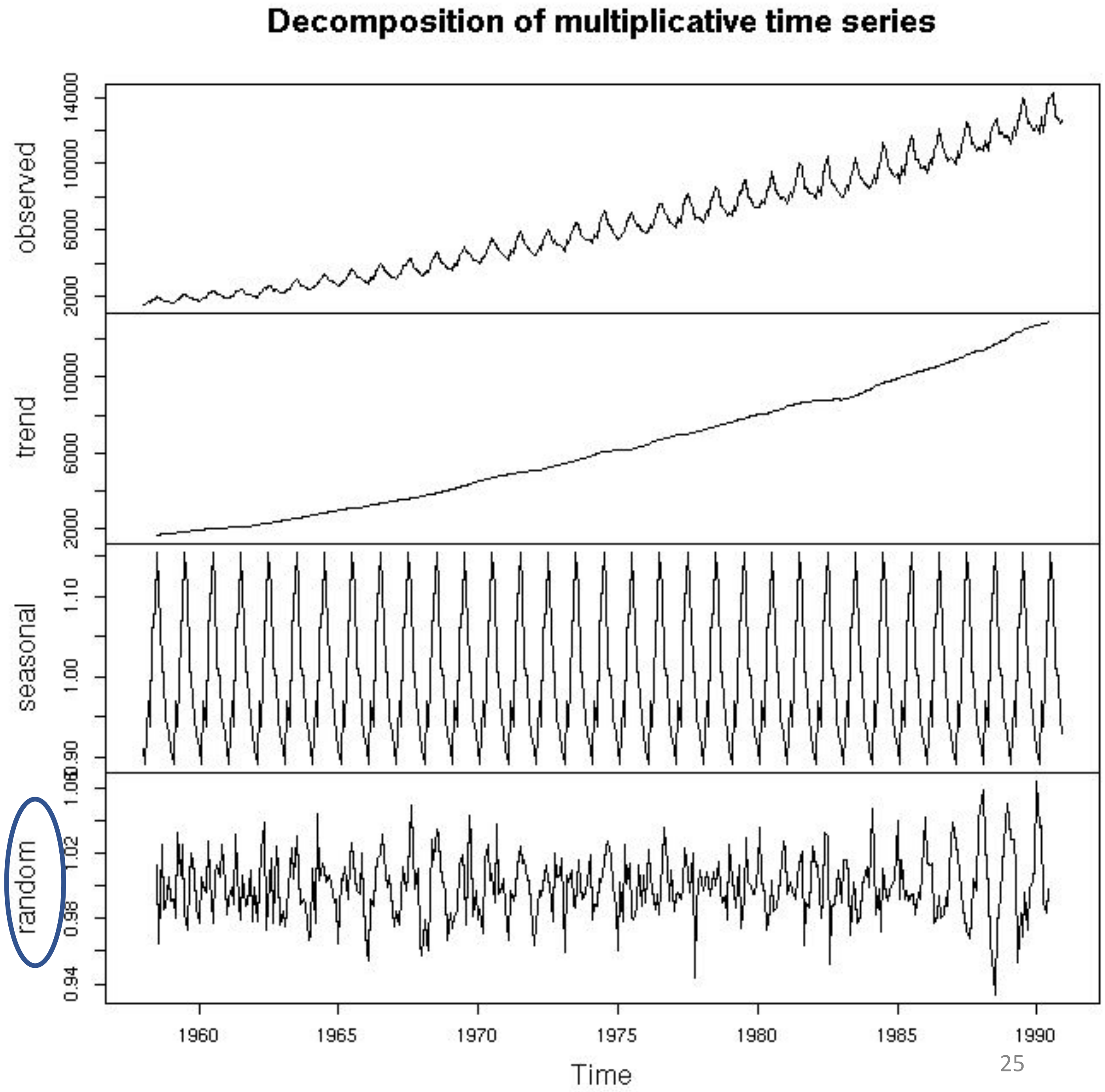
- Method 1: First differencing on seasonal data y_t
  - If seasonal period is 12 months for monthly data (i.e., yearly seasonality): $\Delta x_t = y_t - y_{t-12}$
  - You need to determine seasonal period (e.g., 1 month? 3 months? 12 months? etc.)
  - https://www.stat.berkeley.edu/~gido/Removal%20of%20Trend%20and%20Seasonality.pdf

- Method 2: Including Seasonal dummy variables
  - https://www.ssc.wisc.edu/~bhansen/390/390Lecture14.pdf

- Method 3: Fit SARIMA model
  - Let computer automatically fits the seasonality
  - https://onlinecourses.science.psu.edu/stat510/node/67/
  - https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html

After removal of trend and seasonality

- Fit time series model on the residuals if residuals are not random noise
  - Autoregressive and Moving Average (ARMA)
  - AutoRegressive Integrated Moving Average (ARIMA)
  - ARCH or GARCH



Dickey-Fuller Test on Stationarity

Intuition on Dickey-Fuller Test:

- It is a test against random-walk time series model.
- Random-walk model:

$$y_t = \phi y_{t-1} + \varepsilon_t$$

$$where\ \phi = 1 \qquad {}^{2})$$

http://www.ams.sunysb.edu/~zhu/ams586/UnitRoot_ADF.pdf

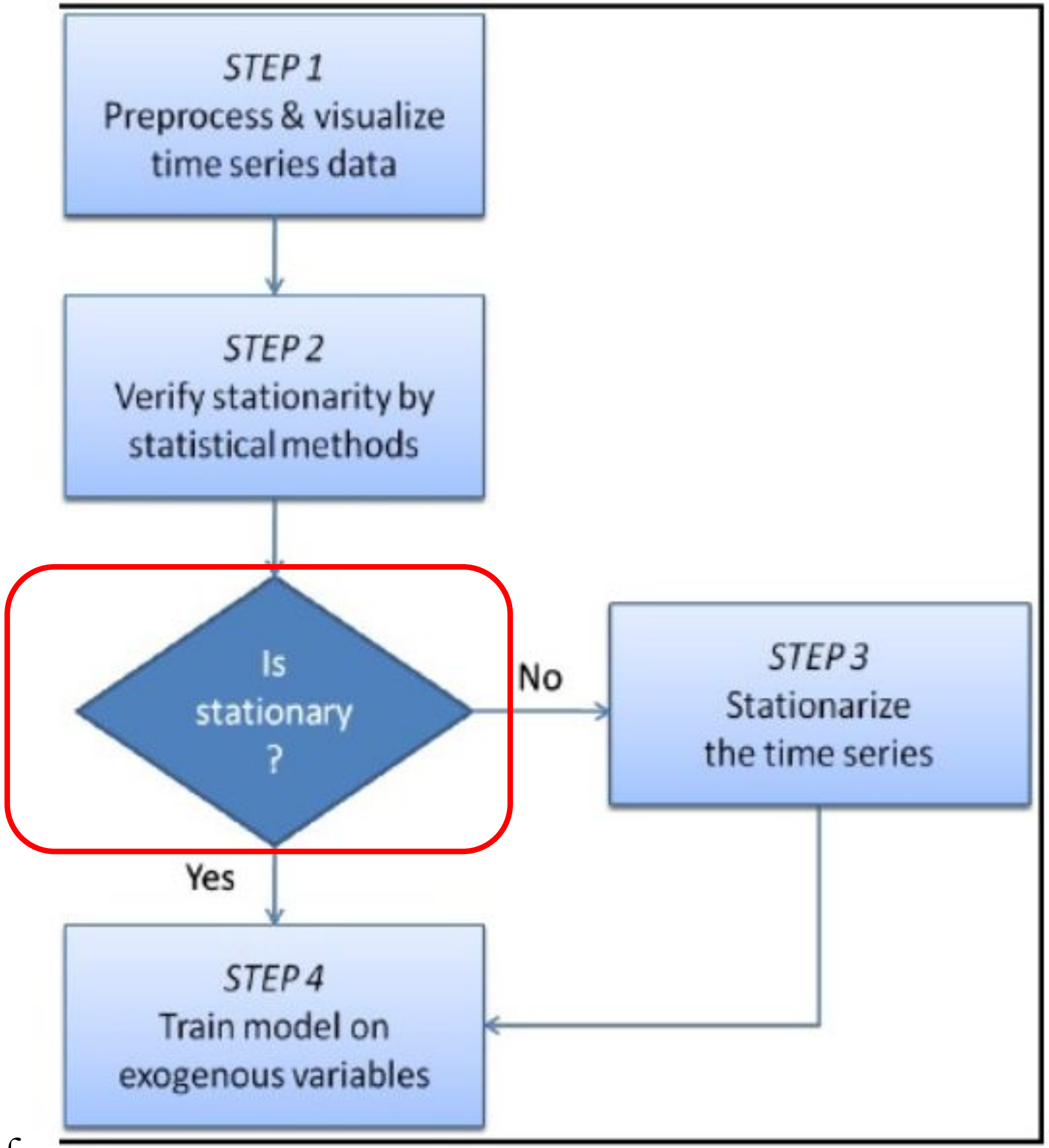

STEP 1
Preprocess & visualize time series data
STEP 2
Verify stationarity by statistical methods
Is stationary ?
No
Yes
STEP 3
Stationarize the time series
STEP 4
Train model on exogenous variables

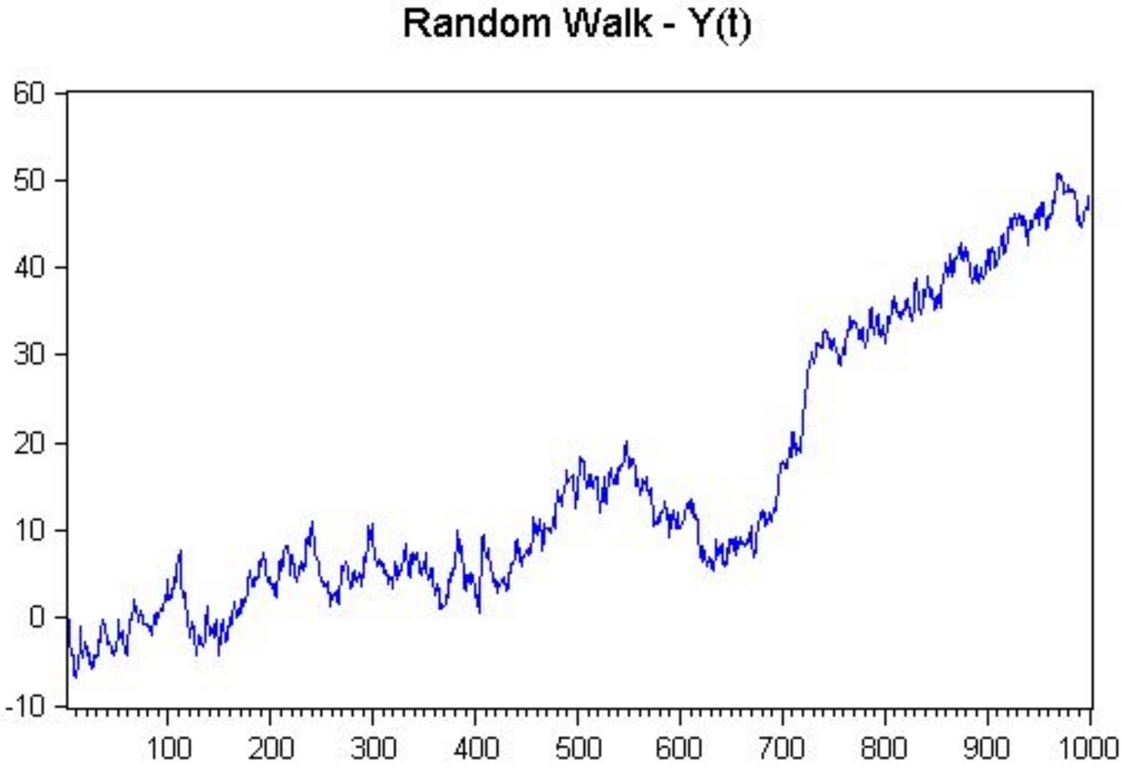
Figure 2.1: A generic methodology of time series analysis

Dickey-Fuller Test on Stationarity

Intuition on Dickey-Fuller Test:

- What is wrong with random-walk model:
- $y_t = y_{t-1} + \varepsilon_t = \ldots = \varepsilon_t + \varepsilon_{t-1} + \varepsilon_{t-2} + \ldots + \varepsilon_1$
  - Mean: $E(y_t) = t^*\mu$
  - Variance: $Var(y_t) = t^*\sigma^2$
  - Autocovariance: $Cov(y_t, y_{t+k}) = t^*\sigma^2$

Random walk series never revert back to mean value



http://www.ams.sunysb.edu/~zhu/ams586/UnitRoot_ADF.pdf

Dickey-Fuller Test on Stationarity

Intuition on Dickey-Fuller Test:

- Test against random-walk model:
- $y_t = φ*y_{t-1}+ε_t$
- $Δy_t = (φ-1)*y_{t-1}+ε_t$
- Null hypothesis: $φ=1$ (We don't care about explosive process: $φ>1$)
- Alternative hypothesis: $φ<1$
- Left-tailed test:
  - When reject null hypothesis, then stationary
  - Fail to reject null hypothesis, then non-stationary

http://www.ams.sunysb.edu/~zhu/ams586/UnitRoot_ADF.pdf

Statistical Tests on Stationarity

Alternative Statistical Tests on Stationarity:

- Augmented Dickey-Fuller Test: Consider multiple lags
  - http://www.ams.sunysb.edu/~zhu/ams586/UnitRoot_ADF.pdf
  - https://en.wikipedia.org/wiki/Augmented_Dickey%E2%80%93Fuller_test

- Phillips–Perron Test
  - https://en.wikipedia.org/wiki/Phillips%E2%80%93Perron_test

- KPSS Test
  - https://en.wikipedia.org/wiki/KPSS_test
  - https://www.bauer.uh.edu/rsusmel/phd/ec2-5.pdf

Statistical Tests on Autocorrelation

If you worry about autocovariance/autocorrelation:

Statistical Tests on Autocorrelation:

- Portmanteau Test
  - https://en.wikipedia.org/wiki/Portmanteau_test
  - Ljung-Box Test
  - https://en.wikipedia.org/wiki/Ljung%E2%80%93Box_test

Statistical Tests on Heteroskedasticity

If you worry about changing variance over time (i.e., heteroskedasticity):

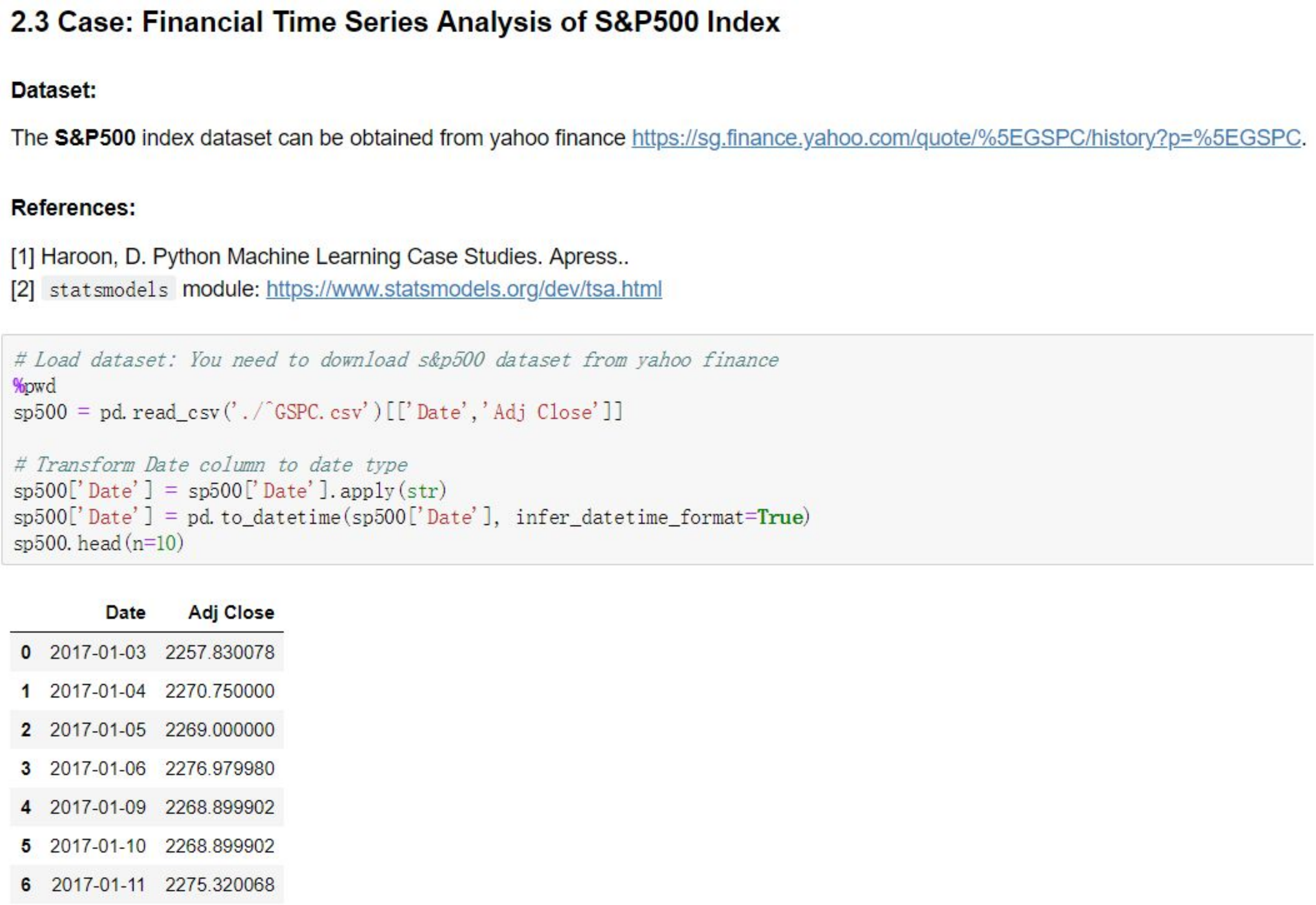
Statistical Tests on heteroskedasticity:

- Breusch-Pagan Test
  - https://en.wikipedia.org/wiki/Breusch%E2%80%93Pagan_test
- McLeod-Li Test and some others
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.553.7228&rep=rep1&type=pdf

Tips for Time Series Analysis

1. Change in variances: (1) log-transformation; (2) or sqrt-transformation
2. Trend: (1) simple regression; (2) or first differencing; (3) or moving average smoothing
3. Seasonality: (1) add dummies; (2) or differencing on seasonal data
4. Run Dickey-Fuller test after each step, until you get stationary residuals
5. Fit with time series model
    - ARMA, ARIMA, SARIMA, ARCH, GARCH, etc.
6. Time series forecasting
    - If you transform the original data, do remember to reverse back
7. Real-time updating/re-training (optional)

Case Analysis: S&P 500 Index



2.3 Case: Financial Time Series Analysis of S&P500 Index

Dataset:

The **S&P500** index dataset can be obtained from yahoo finance https://sg.finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC.

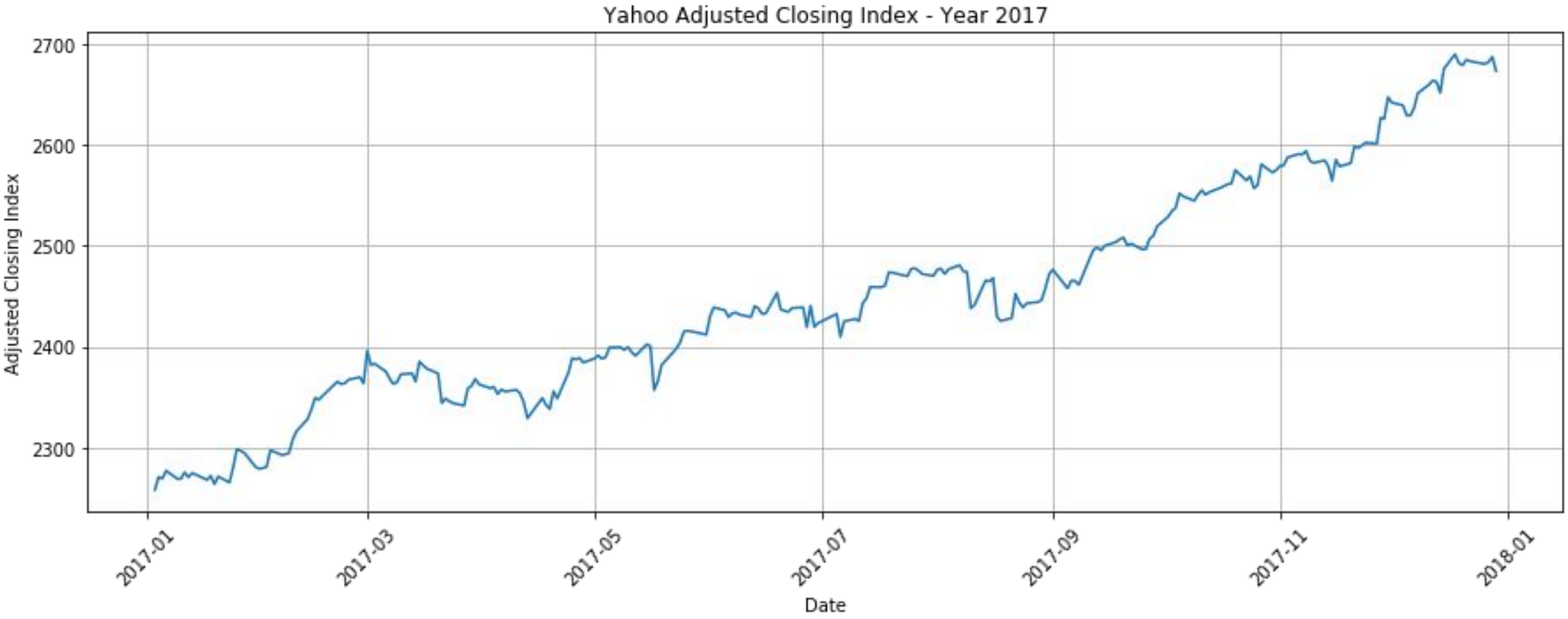
References:

[1] Haroon, D. Python Machine Learning Case Studies. Apress..
[2] `statsmodels` module: https://www.statsmodels.org/dev/tsa.html

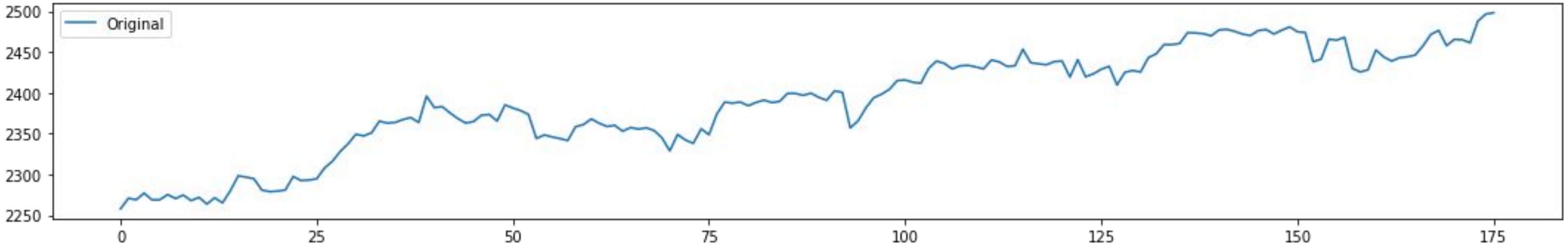
```
# Load dataset: You need to download s&p500 dataset from yahoo finance
%pwd
sp500 = pd.read_csv('./^GSPC.csv')[['Date','Adj Close']]

# Transform Date column to date type
sp500['Date'] = sp500['Date'].apply(str)
sp500['Date'] = pd.to_datetime(sp500['Date'], infer_datetime_format=True)
sp500.head(n=10)
```

	Date	Adj Close
0	2017-01-03	2257.830078
1	2017-01-04	2270.750000
2	2017-01-05	2269.000000
3	2017-01-06	2276.979980
4	2017-01-09	2268.899902
5	2017-01-10	2268.899902
6	2017-01-11	2275.320068

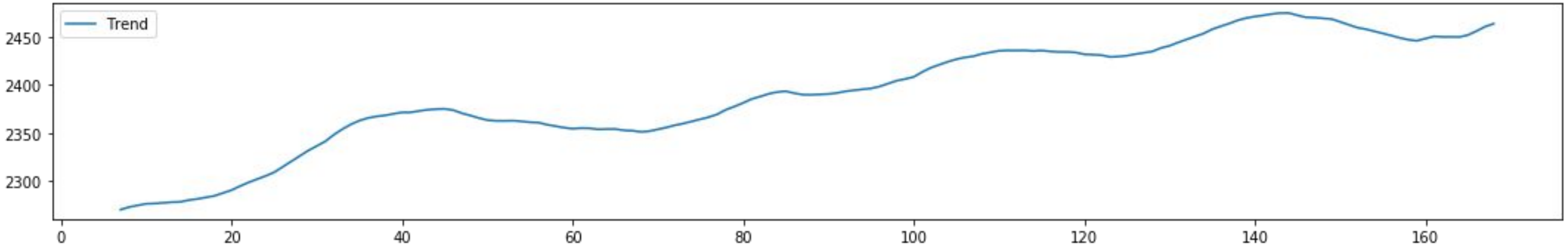
Case Analysis: S&P 500 Index



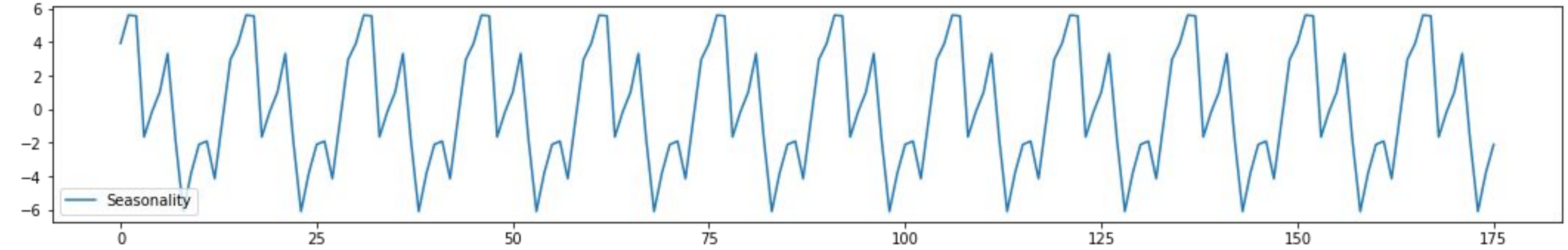
Case Analysis: S&P 500 Index



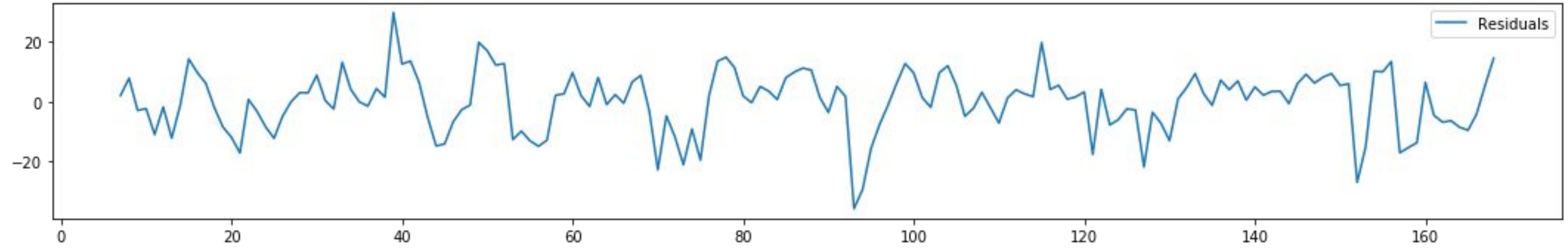
Original Data



Trend



Seasonality



Residuals

Case Analysis: S&P 500 Index

Transforming to Stationary Time Series:

(1) Estimating Trend and Removing It from the Original Series: Using Moving Average Smoothing
  - Dickey Fuller Test: p-value=0.006431

(2) First Differencing
  - Dickey Fuller Test: p-value=0.016438

Case Analysis: S&P 500 Index

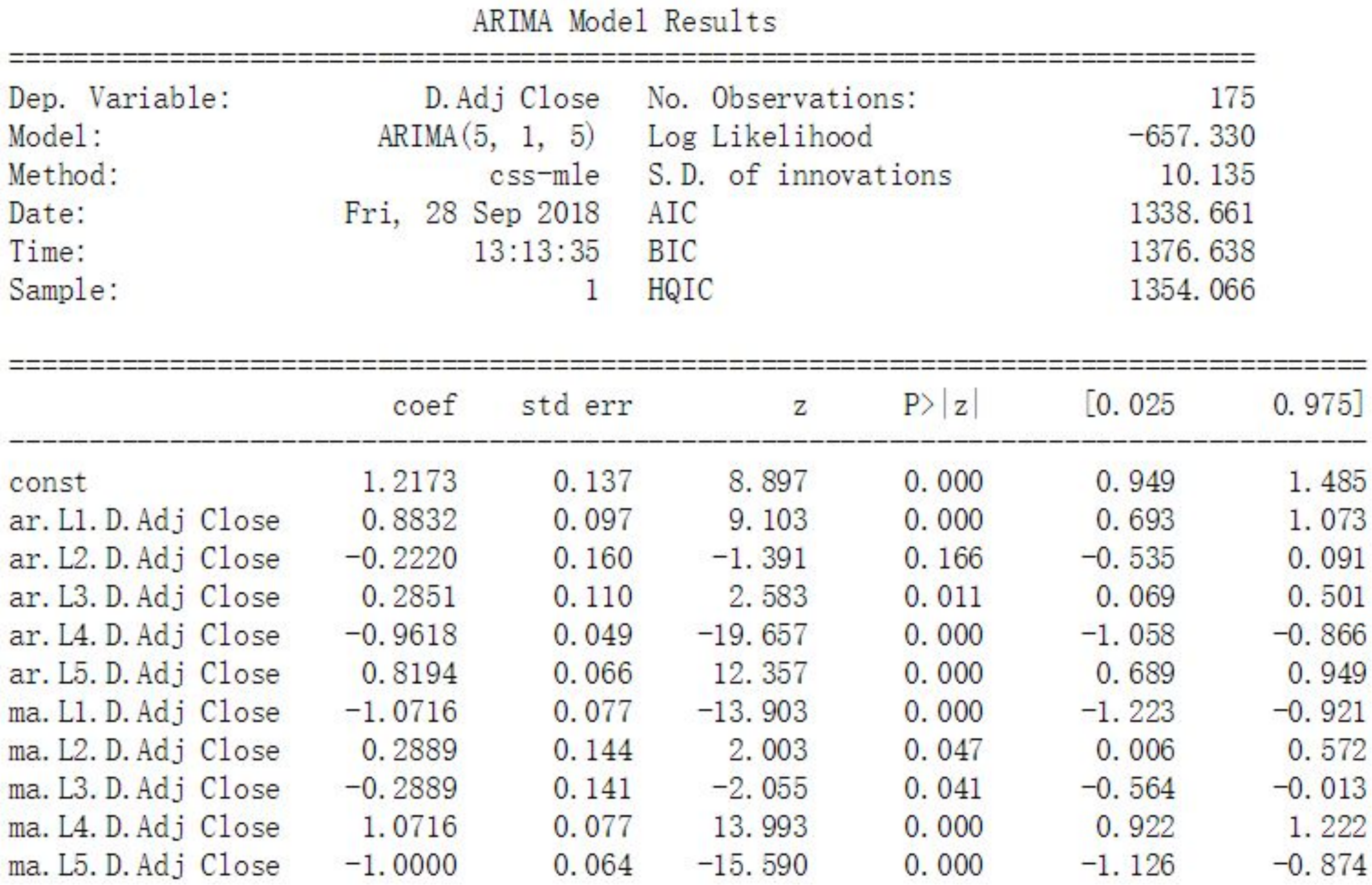
Fit Time Series Model ARIMA:

AIC: Akaike information criterion

- Measures model performance
- The lower the better

$$\mathrm{AIC} = 2k - 2\ln(\hat{L})$$

https://en.wikipedia.org/wiki/Akaike_information_criterion

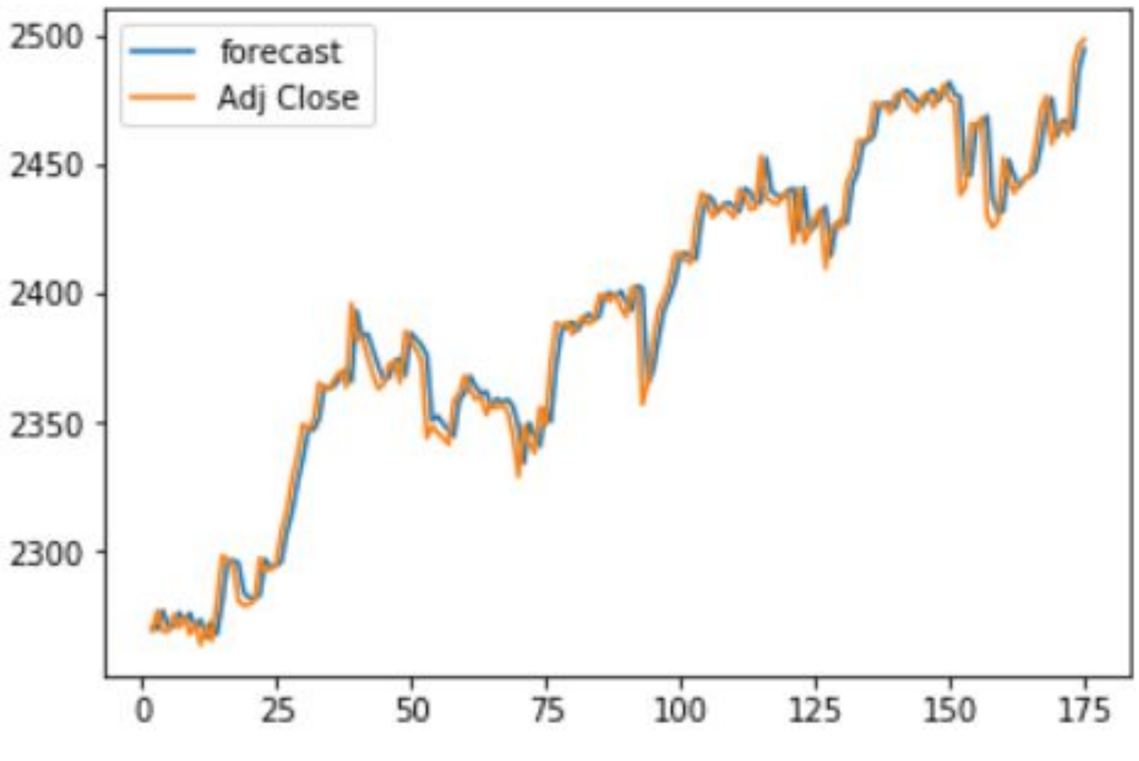


```
                              ARIMA Model Results
==============================================================================
Dep. Variable:             D.Adj Close   No. Observations:                  175
Model:                  ARIMA(5, 1, 5)   Log Likelihood                -657.330
Method:                        css-mle   S.D. of innovations             10.135
Date:                 Fri, 28 Sep 2018   AIC                           1338.661
Time:                         13:13:35   BIC                           1376.638
Sample:                              1   HQIC                          1354.066

=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
const                 1.2173      0.137      8.897      0.000       0.949       1.485
ar.L1.D.Adj Close     0.8832      0.097      9.103      0.000       0.693       1.073
ar.L2.D.Adj Close    -0.2220      0.160     -1.391      0.166      -0.535       0.091
ar.L3.D.Adj Close     0.2851      0.110      2.583      0.011       0.069       0.501
ar.L4.D.Adj Close    -0.9618      0.049    -19.657      0.000      -1.058      -0.866
ar.L5.D.Adj Close     0.8194      0.066     12.357      0.000       0.689       0.949
ma.L1.D.Adj Close    -1.0716      0.077    -13.903      0.000      -1.223      -0.921
ma.L2.D.Adj Close     0.2889      0.144      2.003      0.047       0.006       0.572
ma.L3.D.Adj Close    -0.2889      0.141     -2.055      0.041      -0.564      -0.013
ma.L4.D.Adj Close     1.0716      0.077     13.993      0.000       0.922       1.222
ma.L5.D.Adj Close    -1.0000      0.064    -15.590      0.000      -1.126      -0.874
```

Case Analysis: S&P 500 Index

Predictions:

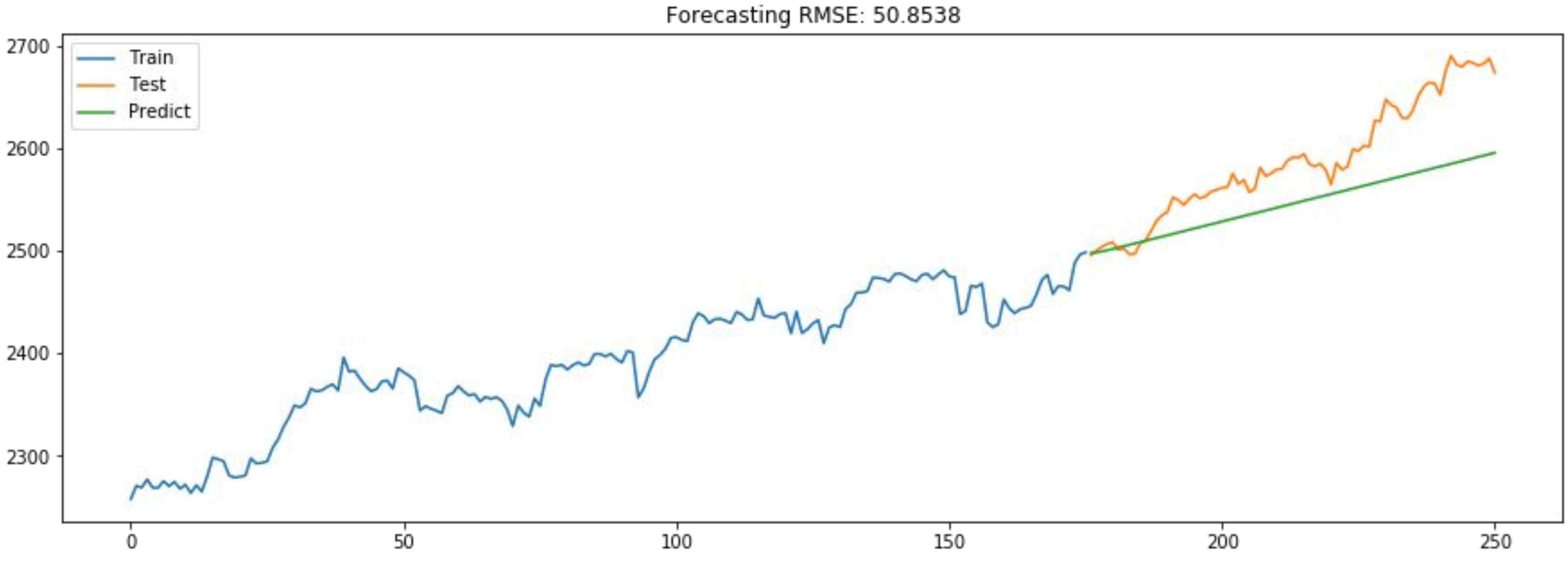
In-sample forecasting: Whether model fits training data well



Case Analysis: S&P 500 Index

Predictions:

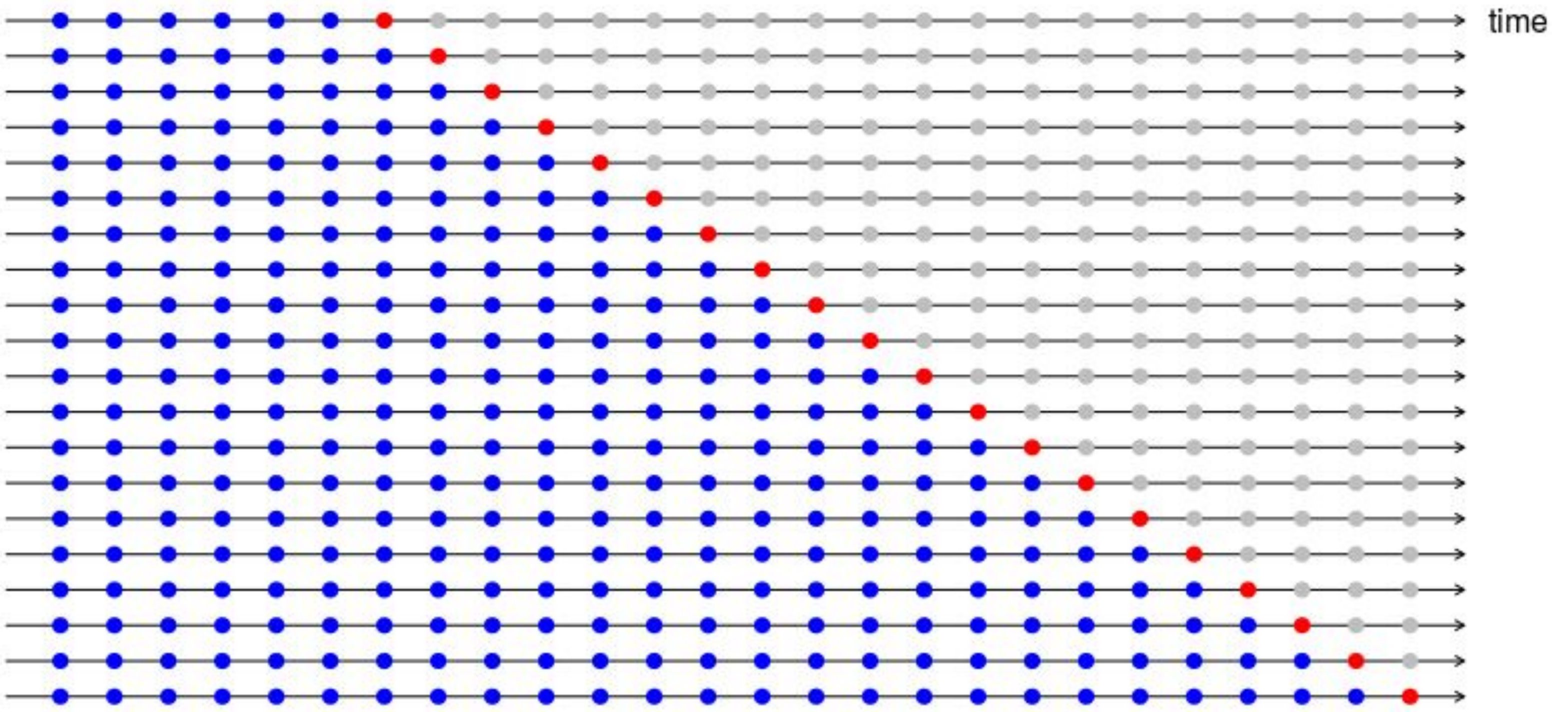
Out-of-sample Forecasting: Prediction using test data



Case Analysis: S&P 500 Index

Predictions:

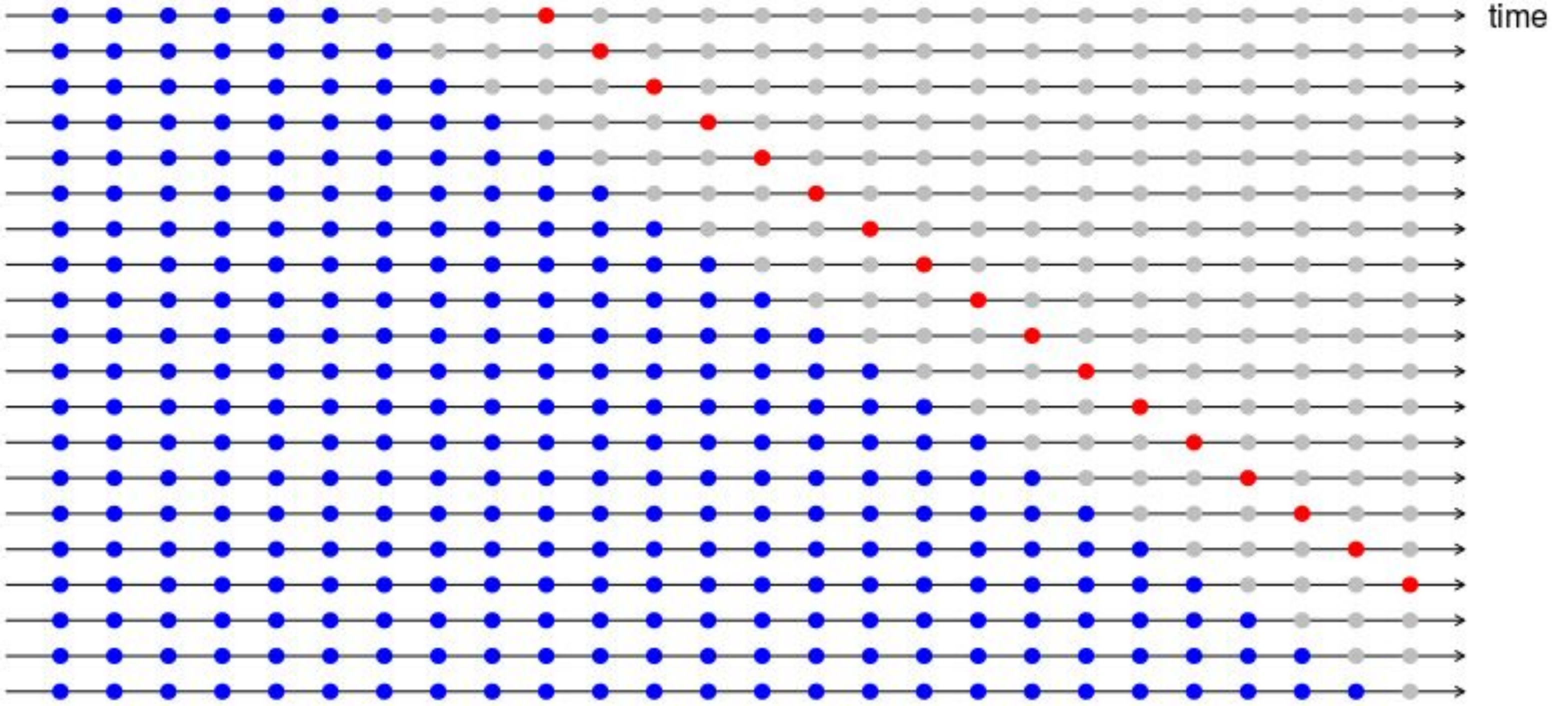
Out-of-sample Forecasting: Rolling Forecasting (One step: Only predict next instance)



Case Analysis: S&P 500 Index

Predictions:

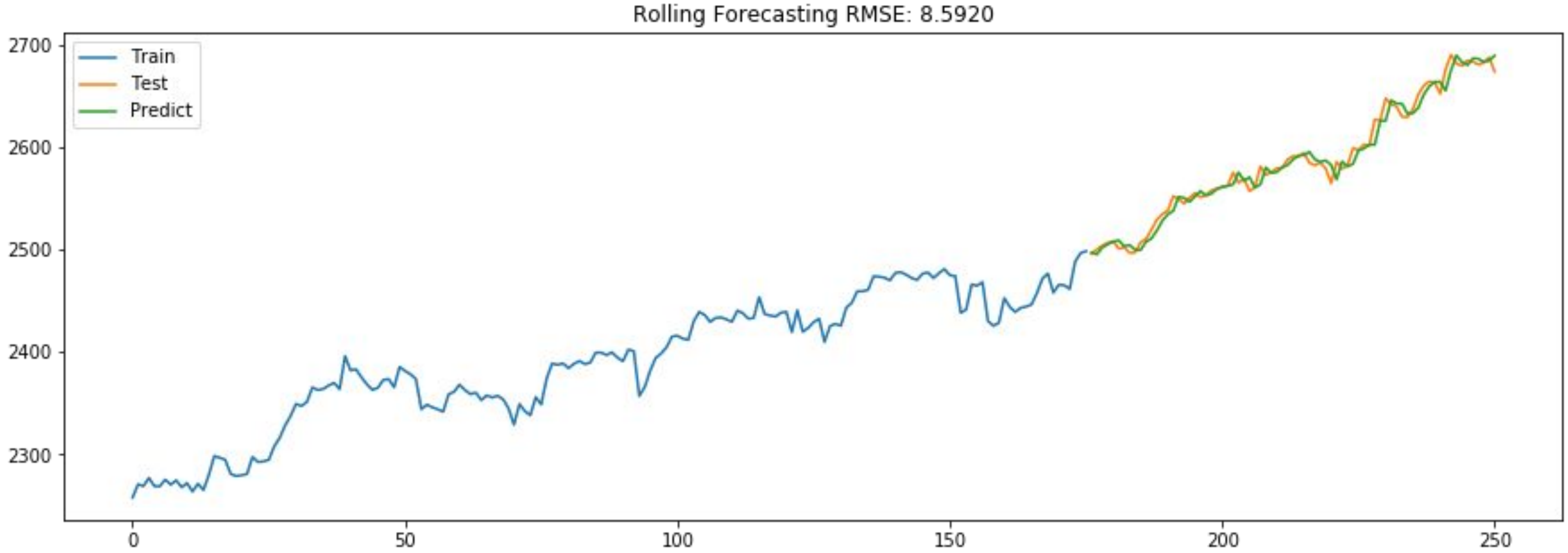
Out-of-sample Forecasting: Rolling Forecasting (Multi-step)



Case Analysis: S&P 500 Index

Predictions:

Out-of-sample Forecasting: Rolling Forecasting (One step)



Case Analysis: S&P 500 Index

Alternative Time Series Model:

SARIMA (Seasonal Autoregressive Integrated Moving Average):

As introduced above, this method addresses seasonality issue when seasonablity is a significant reason for non-stationarity.

ARCH (Autoregressive conditional heteroskedasticity) and **GARCH** (Generalized autoregressive conditional heteroskedasticity):

https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity and
https://arch.readthedocs.io/en/latest/index.html

Case Analysis: S&P 500 Index

Predictions:

How to improve prediction accuracy?

Ensemble method:

https://arxiv.org/ftp/arxiv/papers/1302/1302.6595.pdf

https://ieeexplore.ieee.org/document/6011011

Programming Assignment 6

Using the BT2101 Tutorial 6 Notebook (Time Series Analysis.ipynb), please answer the questions in the jupyter notebook

Answer all in the jupyter notebook.

Instructions

Submit Python Notebook to the submission folder and Named:
AXXXX_T6_program.ipynb

Include your answers in the jupyter notebook

- You need to show outputs, instead of just showing functions.

Submit by Tuesday OCT-16 (by 12:00pm noon)

- Based on Time Series Analysis.ipynb

Thank You!