Machine Learning

Lesson 7: Time Series Modeling

Concepts Covered

- Components of a Time Series Data
- Stationarity in Time Series
- ARIMA Modelling

Learning Objectives

By the end of this lesson, you will be able to:

- Understand time series analysis
- Build time series models using ARIMA

Time Series Modeling

Topic 1: Overview

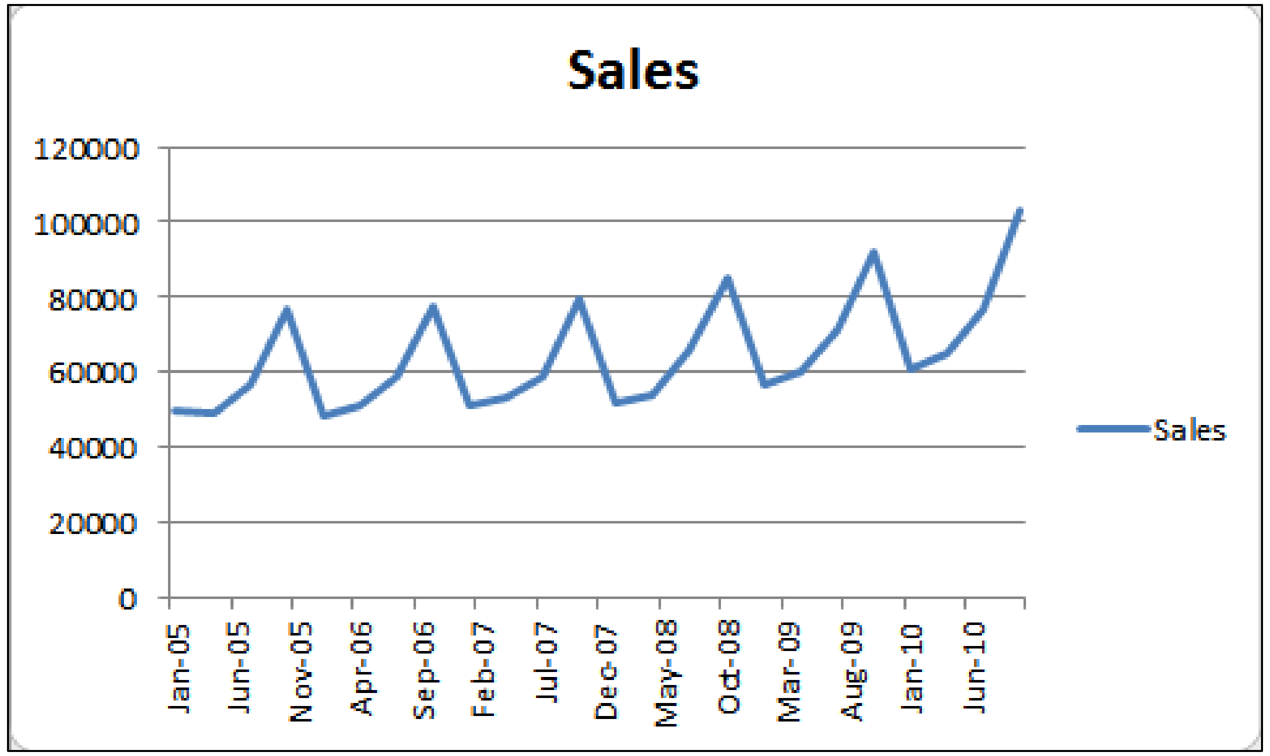
Definition

Time Series can be defined as a set of measurements of certain variable made at **regular time intervals**.

Time acts as an independent variable for estimation

A time series defined by the values Y1, Y2.. of a variable Y at times t1, t2, t3.. is given by :

$$Y = F(t)$$



Series of monthly sales data

Applications

Daily sales score of E-commerce

Weekly production of a shoe manufacturing company

Yearly GDP of a developing country

Monthly tickets sold by an airline

Notice that all these datasets include time

Need

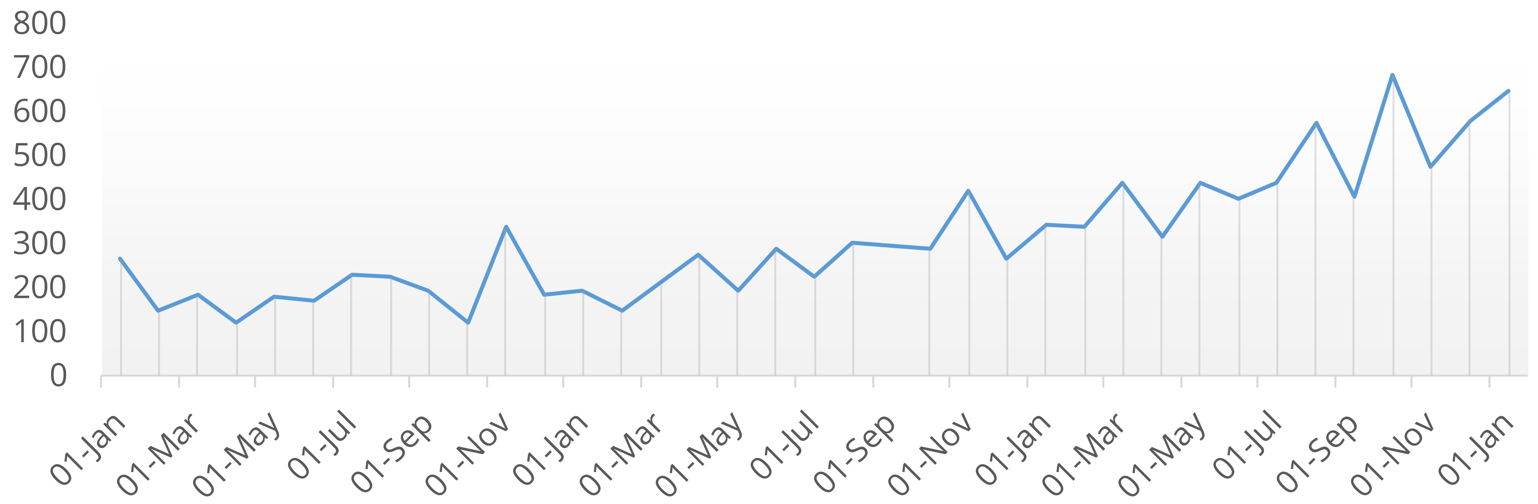
Understand seasonal patterns

Evaluate current progress

Detect unusual events

Forecasting

Time Series Pattern Types



Uptrend

Smartphone sales for a 3 year period

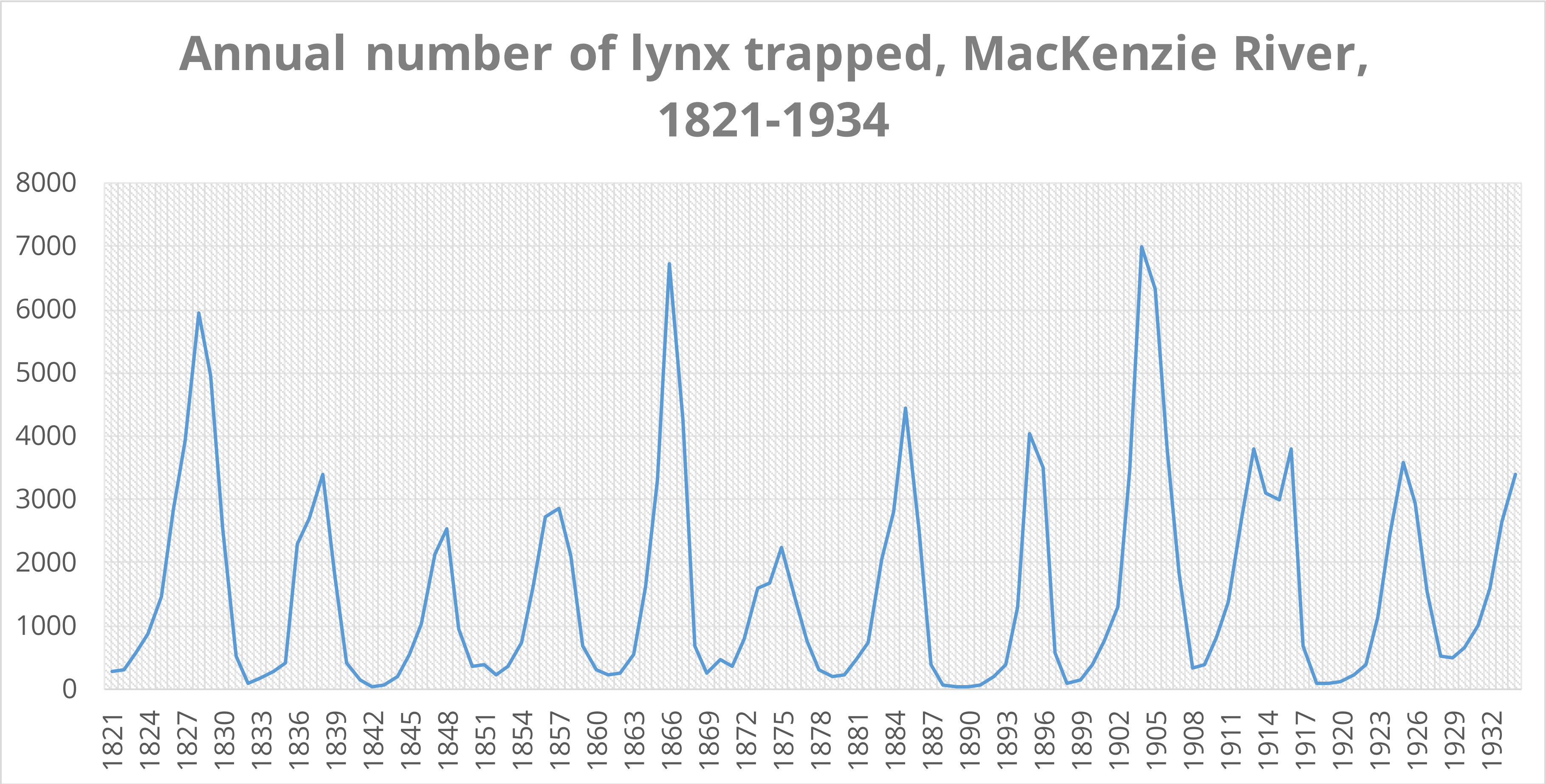
Downtrend

Stock Market price for a wall street company

Trend

A trend is a long-term increase or decrease in time series data

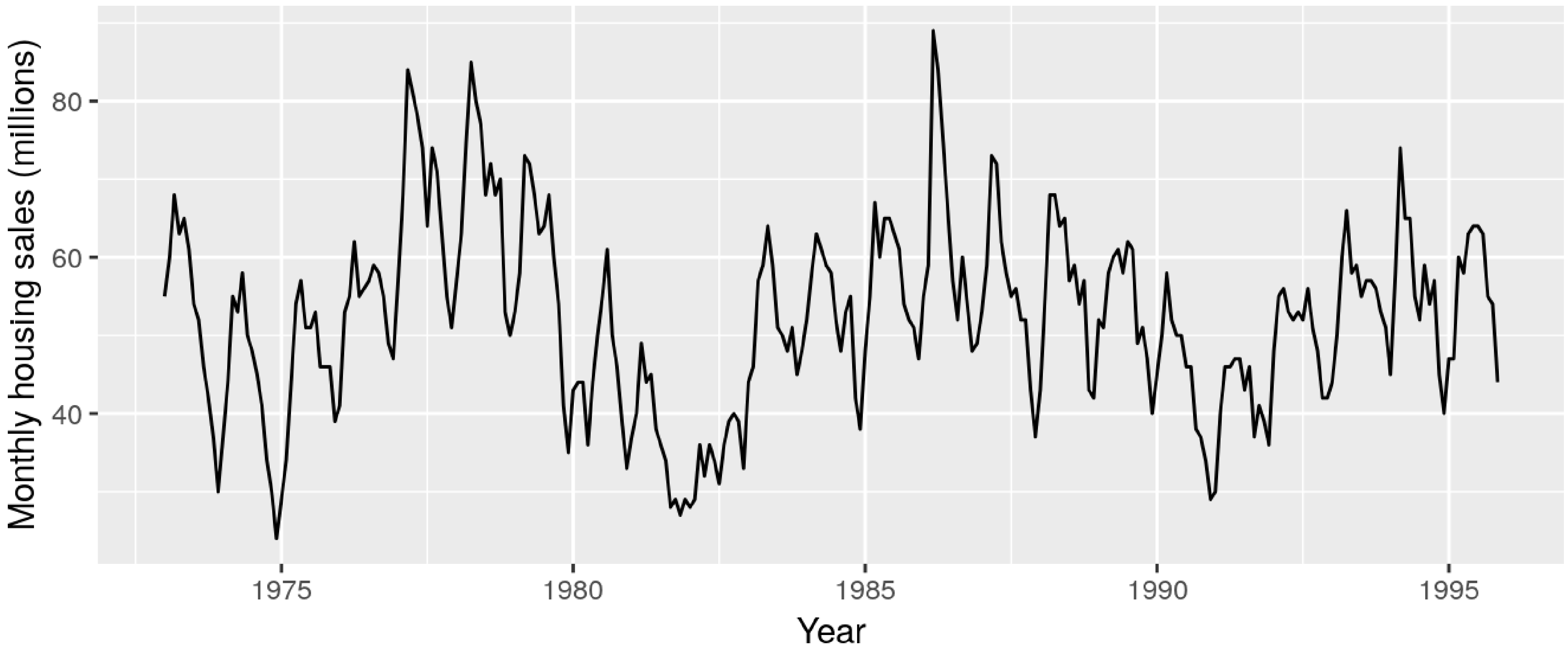
Time Series Pattern Types (Contd.)



Seasonal

- When factors such as the time of the year or the day of the week affect the dependent variable, repetitive patterns are observed in the time series
- Seasonality is always of a fixed and known frequency

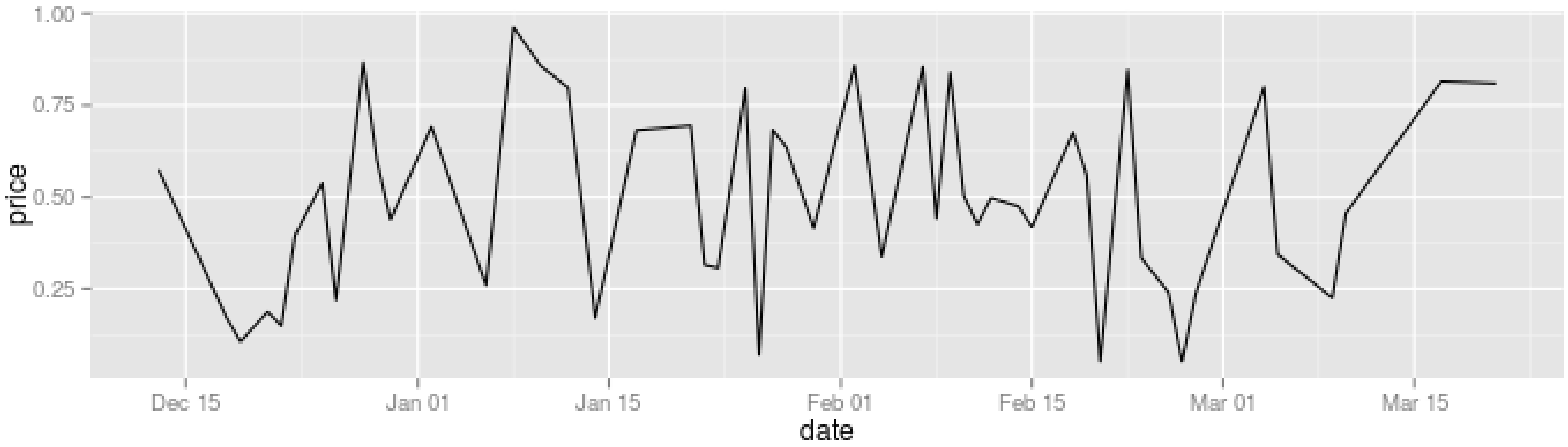
Time Series Pattern Types (Contd.)



Cyclic

- Unlike seasonal patterns, cyclic patterns exhibit rise and fall that are not of fixed period
- Duration is at least 2 years

Time Series Pattern Types (Contd.)

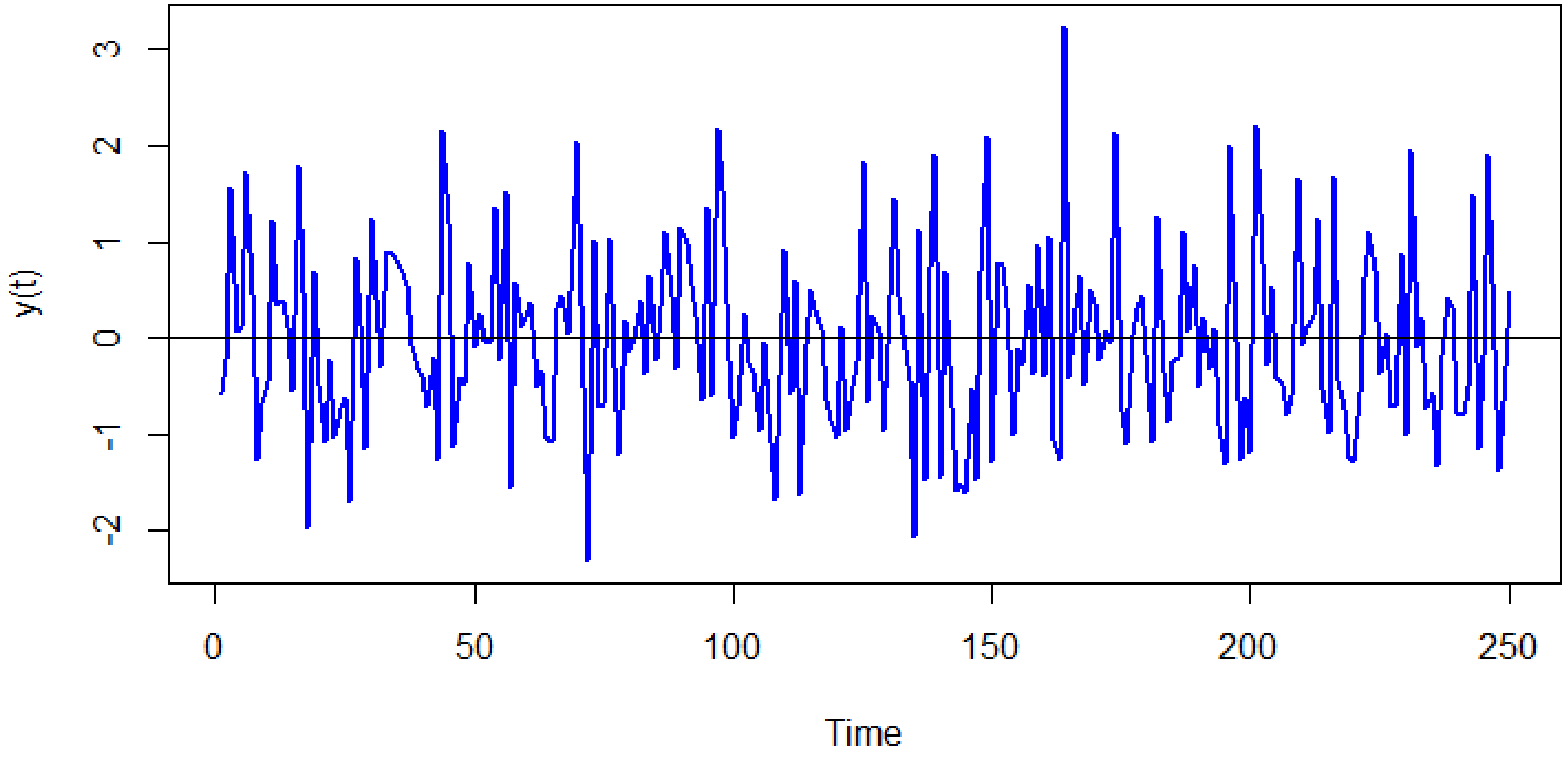


Irregular

- Irregular patterns might occur due to random or unforeseen events
- They are often of short duration and non-repeating

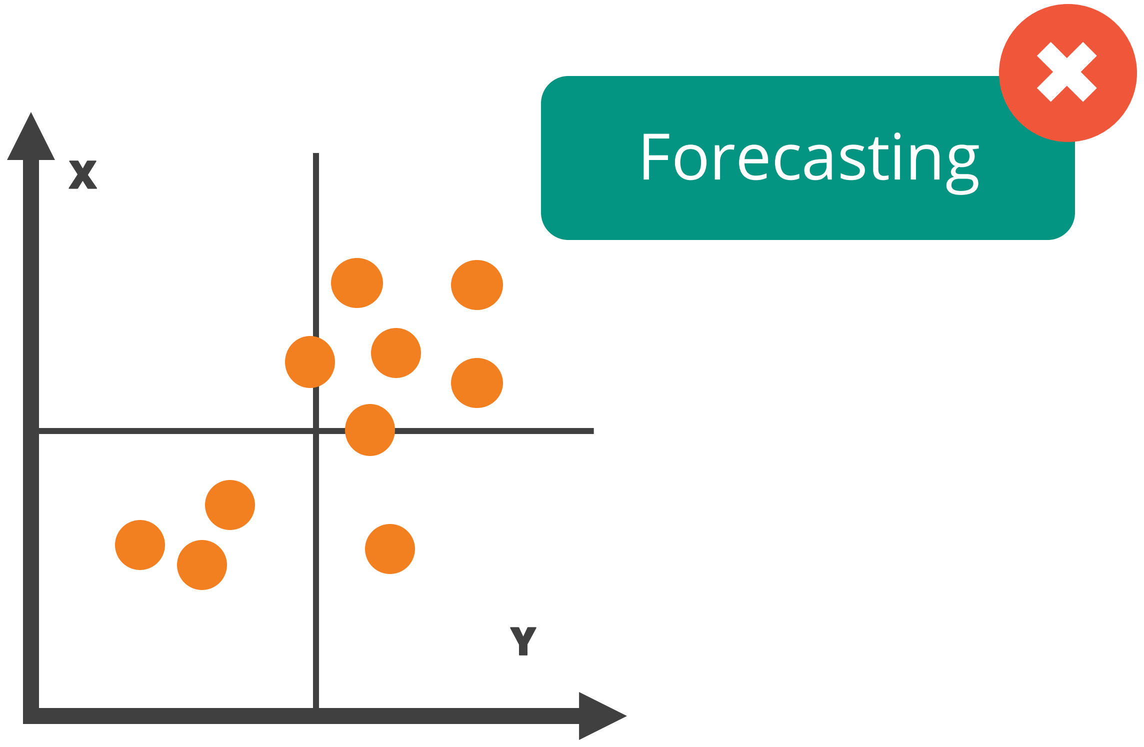
White Noise

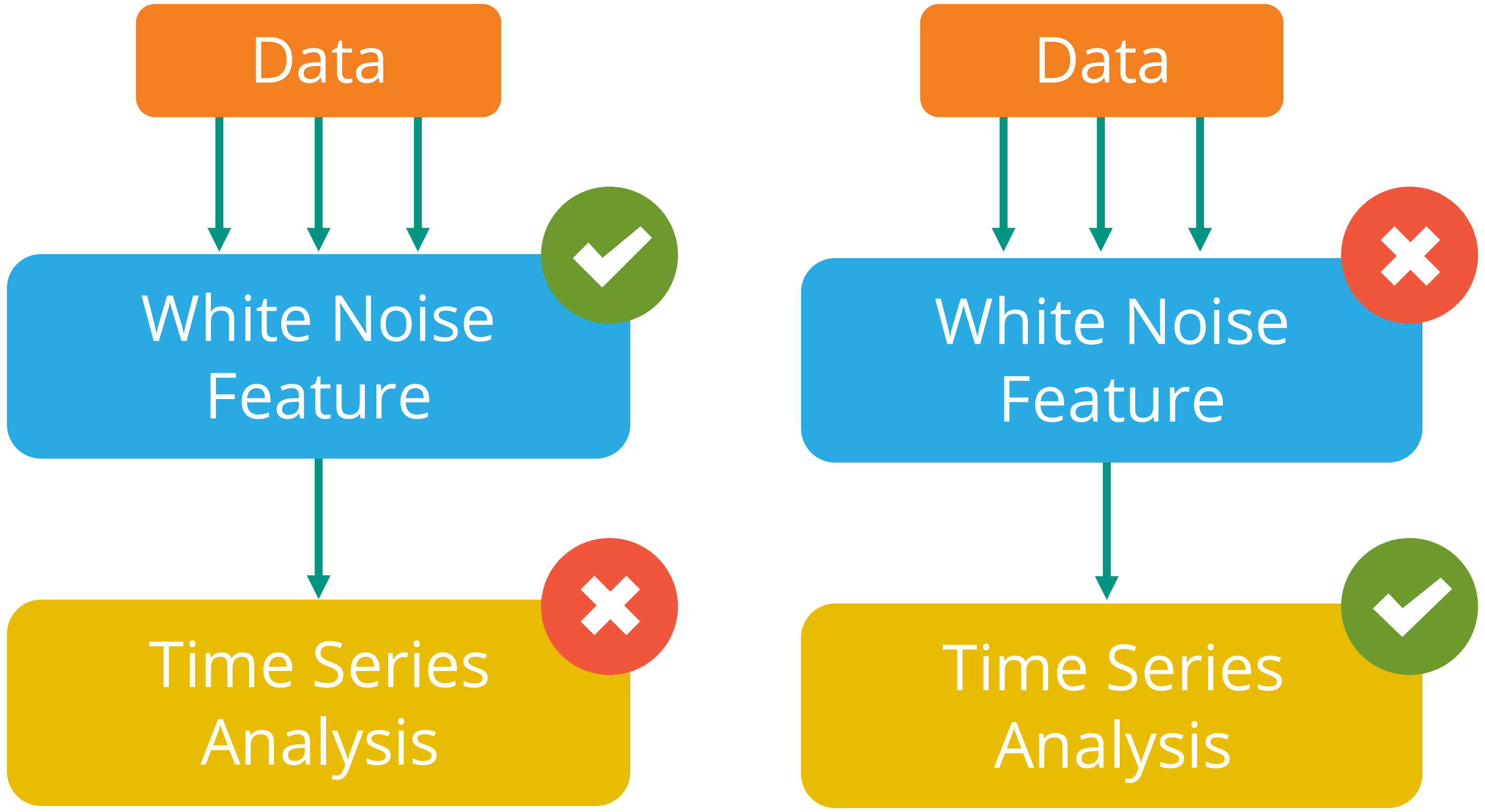
A white noise series is one with a zero mean, a constant variance, and no correlation between its values at different times.



Since values are uncorrelated, the adjacent values do not help to forecast future values

White Noise (Contd.)





Example: Stock prices of companies may vary daily and time series become uncorrelated

Time Series Modeling

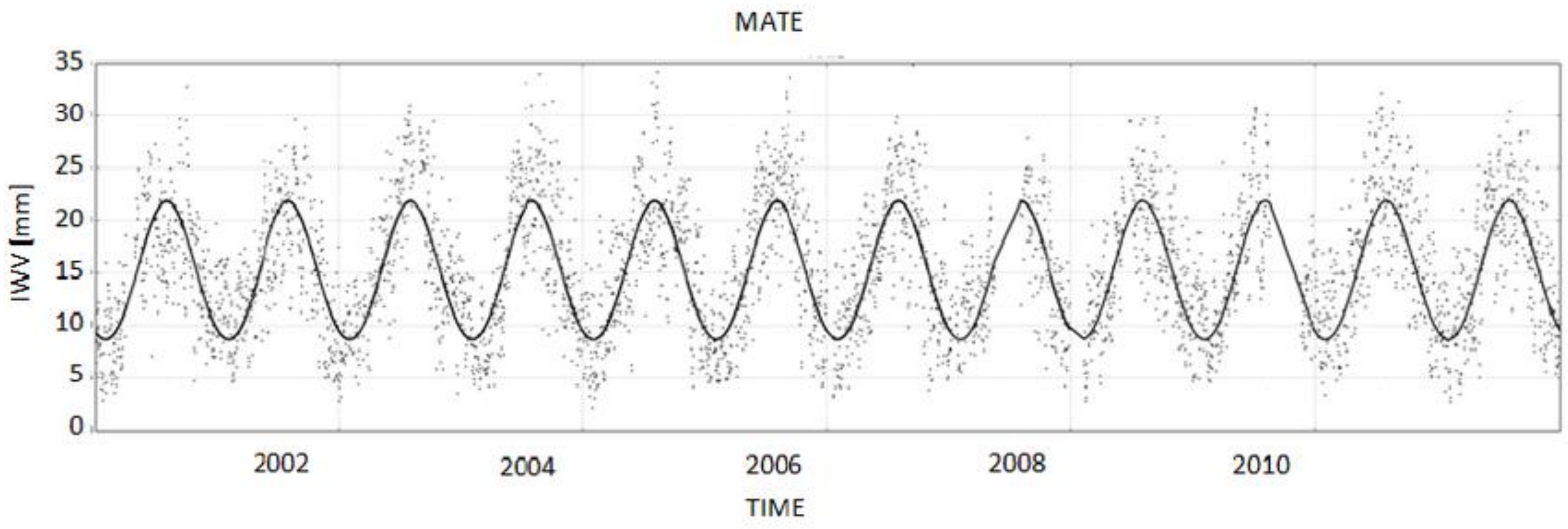
Topic 2: Stationarity

Stationarity

Criterion to classify a series as stationary

Time Invariant (constant)

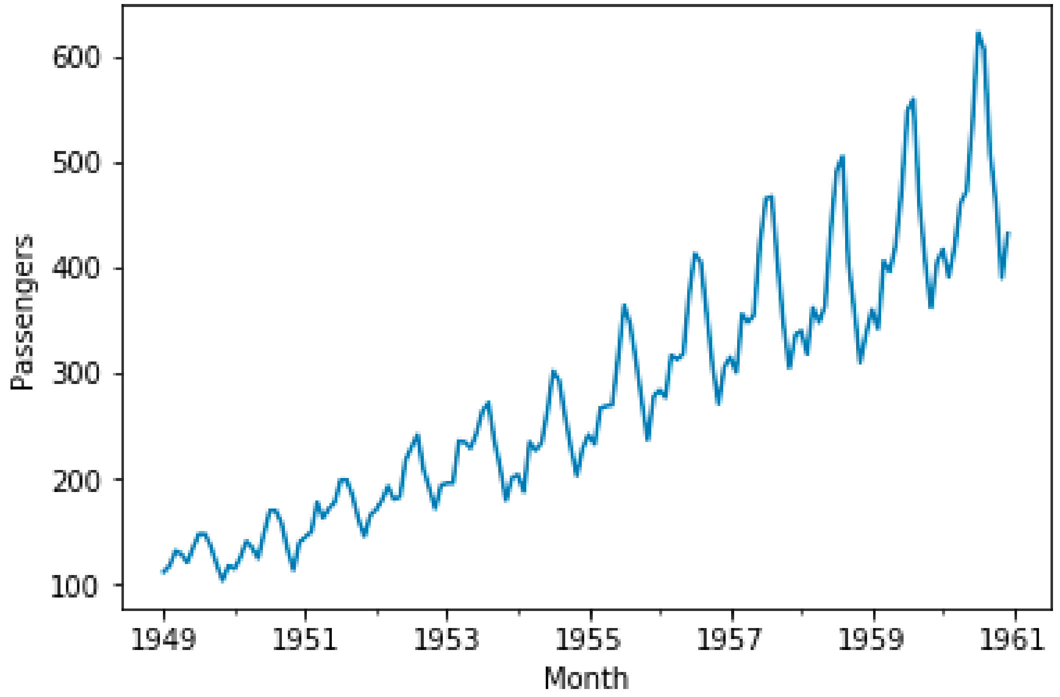
- Mean
- Variance
- Covariance



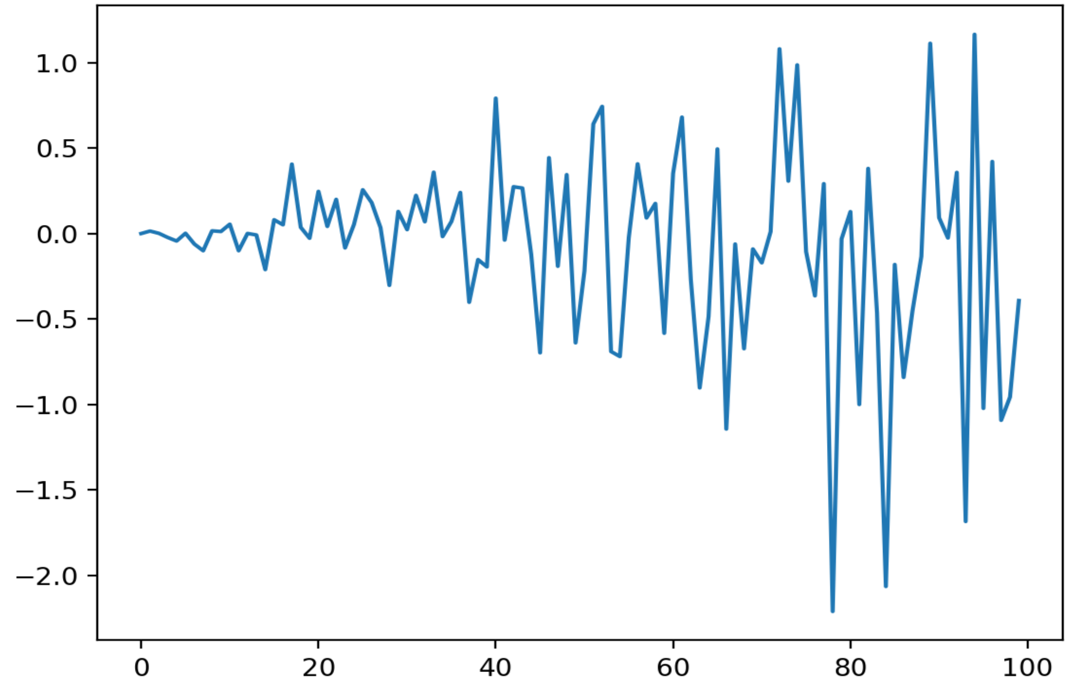
Stationary series

The time series should be stationary to build the model

Non-Stationary Series

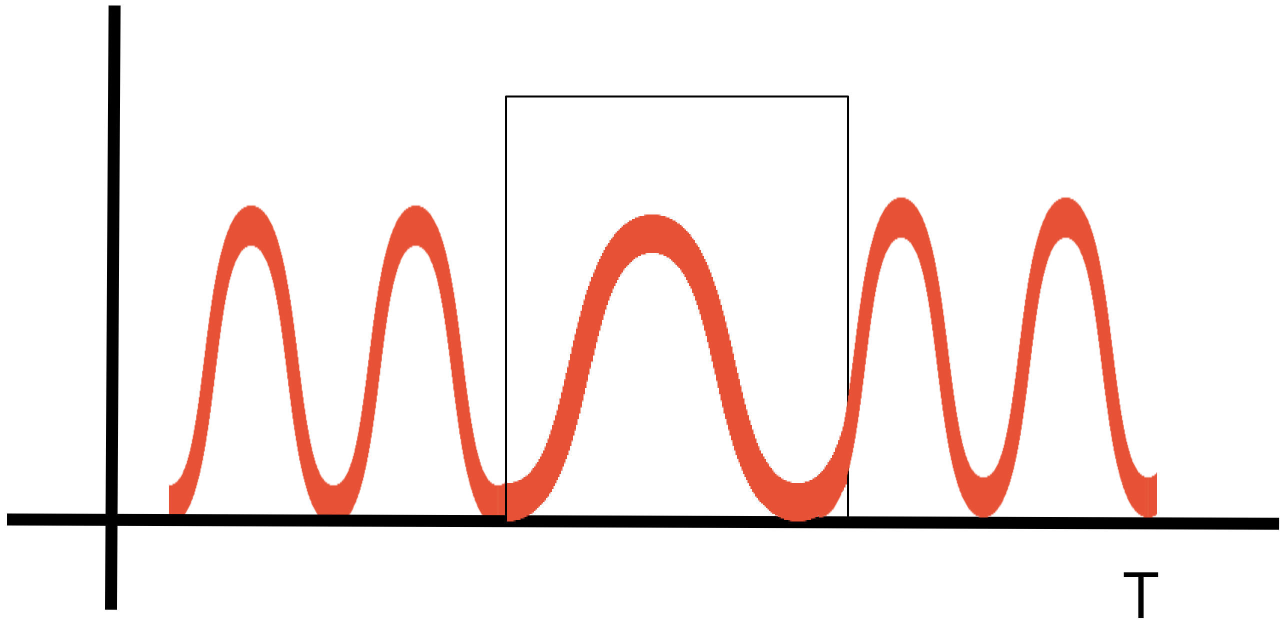


Increasing trend or non-constant **mean**



Non-constant **variance**

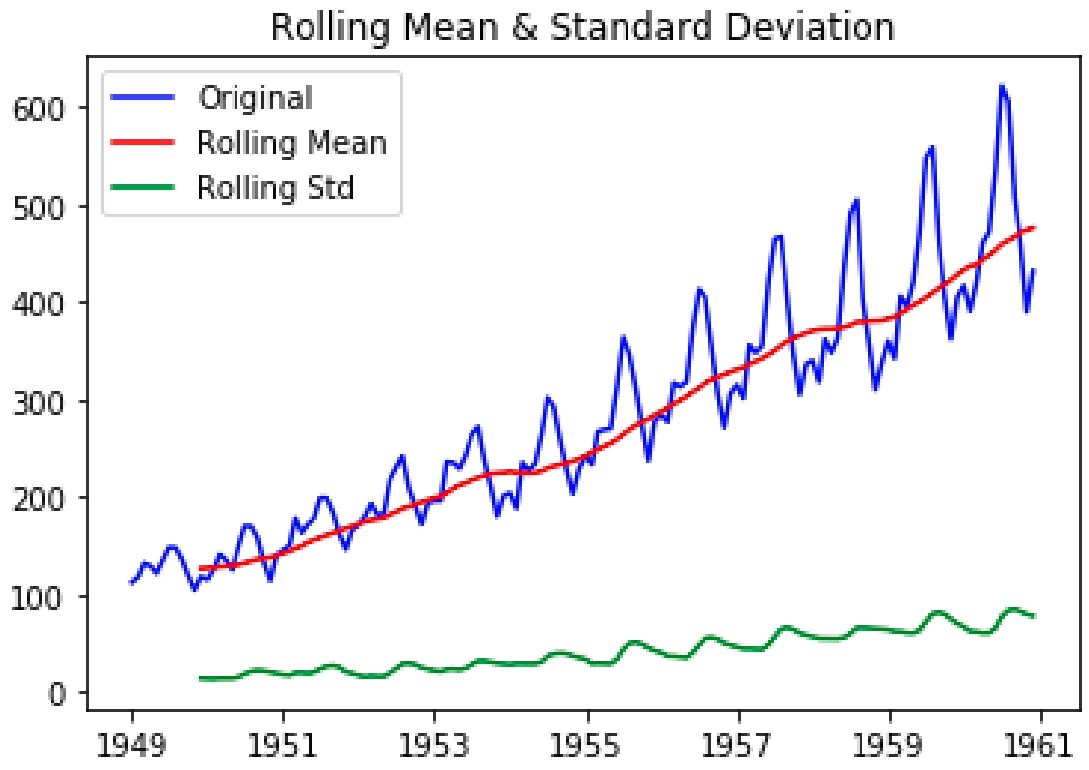
Co-variance is not constant with time



simplilearn

Stationarity Check

1 Rolling Statistics (Visual)



Plot the moving average or moving variance to check if it varies with time.

Notice the mean and variance **increase** constantly

Dickey Fuller test (Statistical) 2

```
Test Statistic                   0.815369
p-value                          0.991880
#Lags Used                      13.000000
Number of Observations Used    130.000000
Critical Value (1%)             -3.481682
Critical Value (5%)             -2.884042
Critical Value (10%)            -2.578770
dtype: float64
```

Null Hypothesis = TS is non-stationary

If 'Test Statistic' < 'Critical Value', **Reject** the null hypothesis

Removal of Non-Stationarity

Differencing

Decomposition

Getting a TS perfectly stationary is desirable but not practical, so it is made as close as possible using these **statistical techniques**

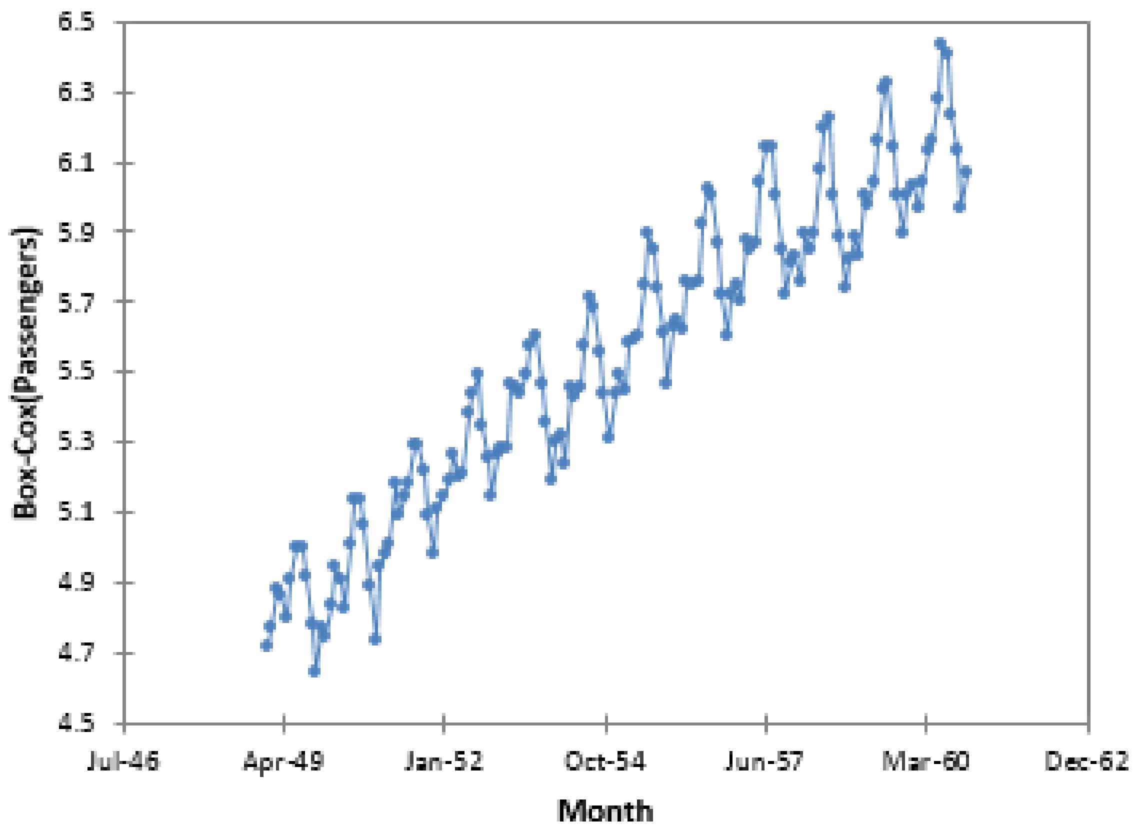
Differencing

Differencing is performed by subtracting the previous observation from the current observation.

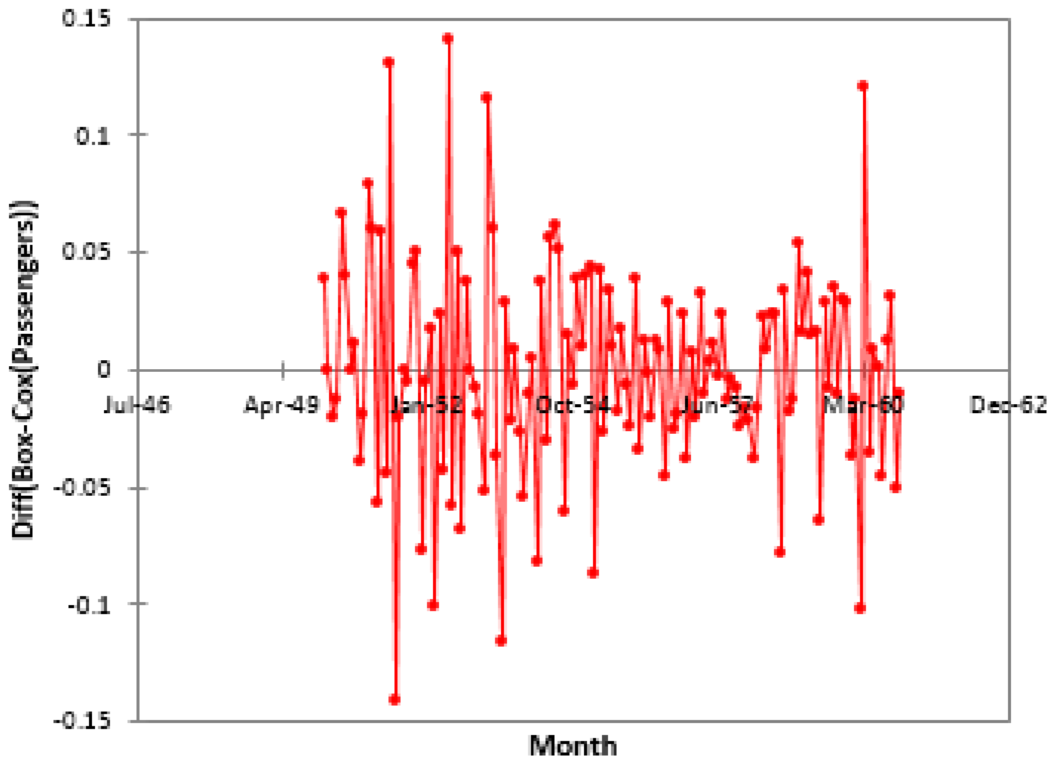
$$\Delta y_t = y_t - y_{t-1}$$

Δy_t is the difference between two successive values

Y_t is the value of y at t and Y_{t-1} is the value preceding Y_t



Non-stationary series



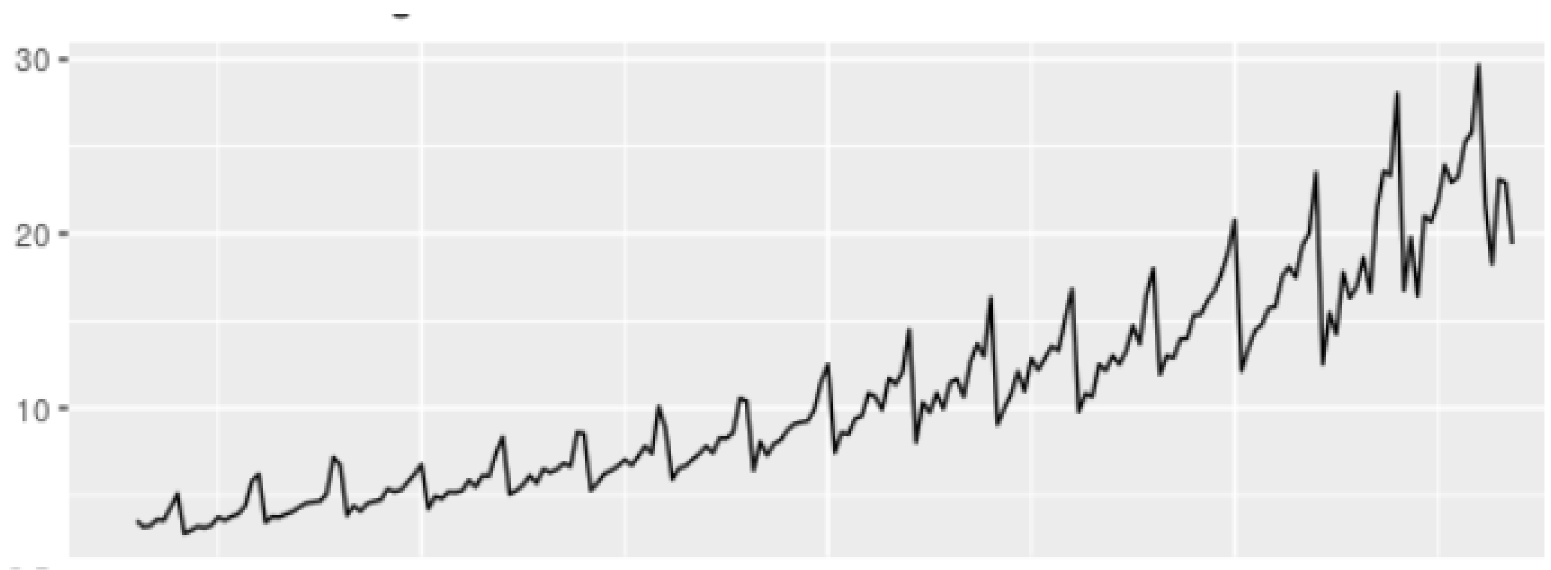
On differencing the series on left

Decomposition

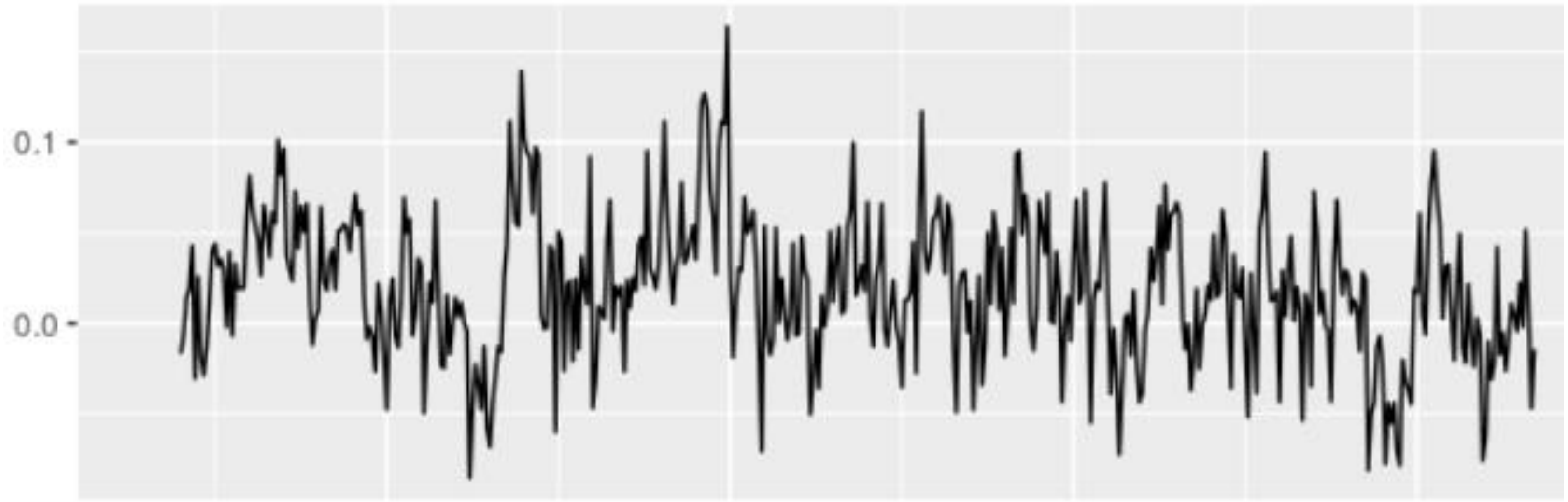
Detrending or de-seasonalizing eliminates the trend and seasonality respectively.

Decomposition is performed on the original series by regressing the series on time and taking the residuals from the regression.

$$y_t = \mu + \beta t + \epsilon_t$$



Seasonality with increasing trend



Seasonally decomposed series

i You can also use techniques like **transformation** which penalize higher values more than lower values. Example: square root, cube root, log.

Assisted Practice

Stationarity

Duration: 15 mins.

Problem Statement: The Air Passenger dataset provides monthly total of US airline passengers, from 1949 to 1960. This dataset is of a time series class.

Objective:

- Check for the stationarity of your data using Rolling Statistics and Dickey fuller test
- If stationarity is present, remove it using differencing in Python

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Unassisted Practice

Stationarity

Duration: 20 mins.

Problem Statement: The Beer production dataset provides a time series data for monthly beer production in Australia, for the period Jan 1956 – Aug 1995.

Objective:

- Check for the stationarity of your data using Rolling Statistics and Dickey fuller test
- if stationarity is present, remove it using differencing in Python

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Step 1: Data Import

Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import rcParams
from datetime import datetime
%matplotlib inline
df = pd.read_csv('monthly-beer-production-in-austr.csv')
df.head()
```

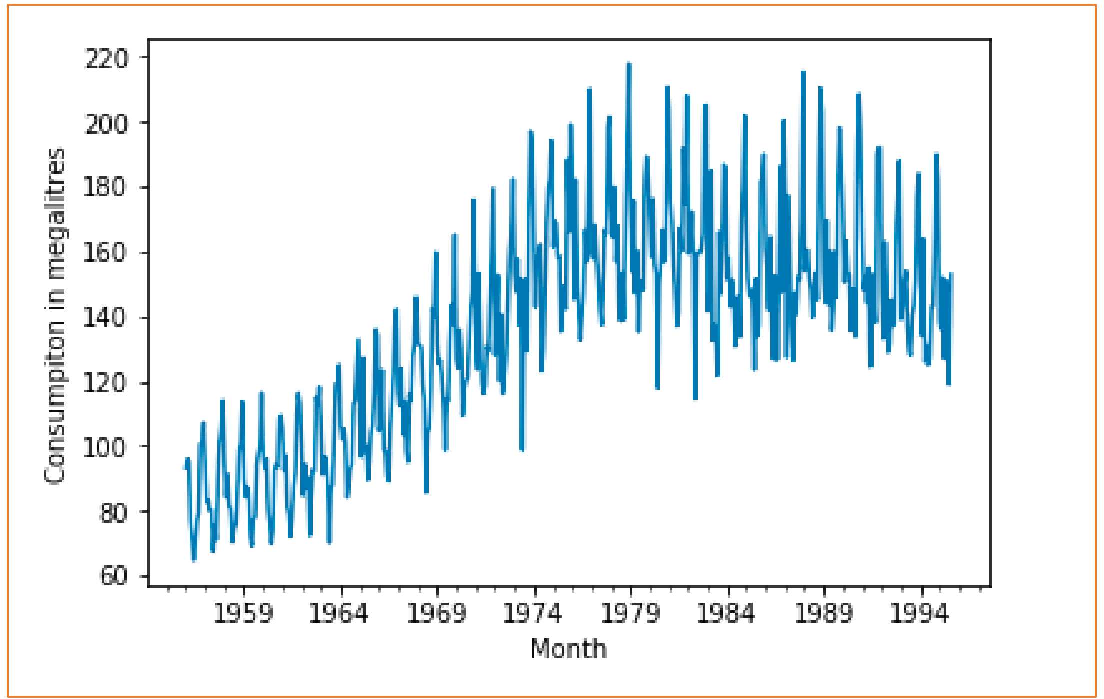
	Month	Monthly beer production in Australia
0	1956-01	93.2
1	1956-02	96.0
2	1956-03	95.2
3	1956-04	77.1
4	1956-05	70.9

Step 2: Parse and Plot

Code

```
dateparse = lambda dates: pd.datetime.strptime(dates, '%Y-%m')
data = pd.read_csv('monthly-beer-production-in-austr.csv',
parse_dates=['Month'], index_col='Month', date_parser=dateparse)

ts = data['Monthly beer production in Australia']
ts.plot()
plt.ylabel("Consumption in megalitres")
```



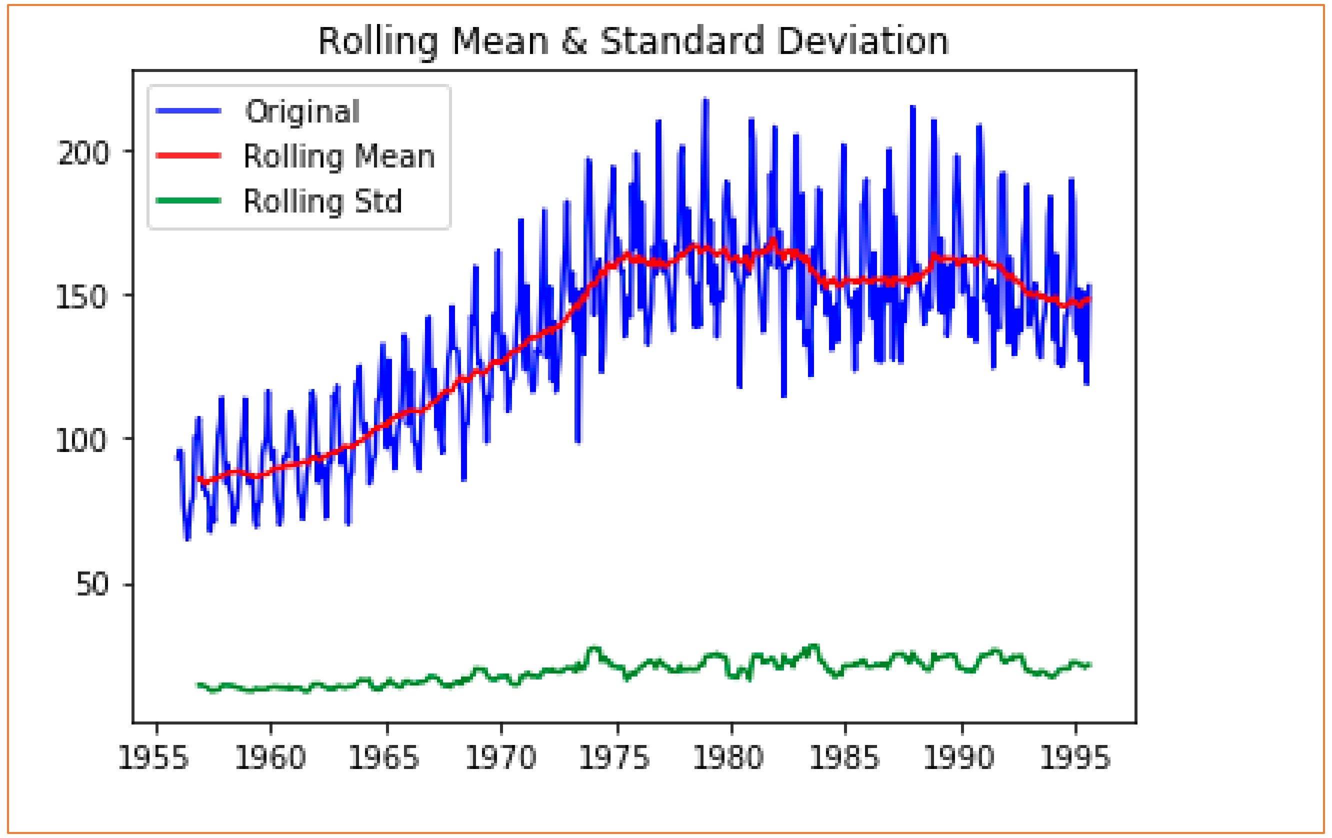
Step 3: Stationarity Check

```
from statsmodels.tsa.stattools import adfuller
def test_stationarity(timeseries):

    #Determing rolling statistics
    rolmean = timeseries.rolling(window=52,center=False).mean()
    rolstd = timeseries.rolling(window=52,center=False).std()
    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)
    #Perform Dickey-Fuller test:
    print ('Results of Dickey-Fuller Test:')
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags
                                             Used','Number of Observations Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print (dfoutput)

test_stationarity(data['Monthly beer production in Australia'])
```

Output



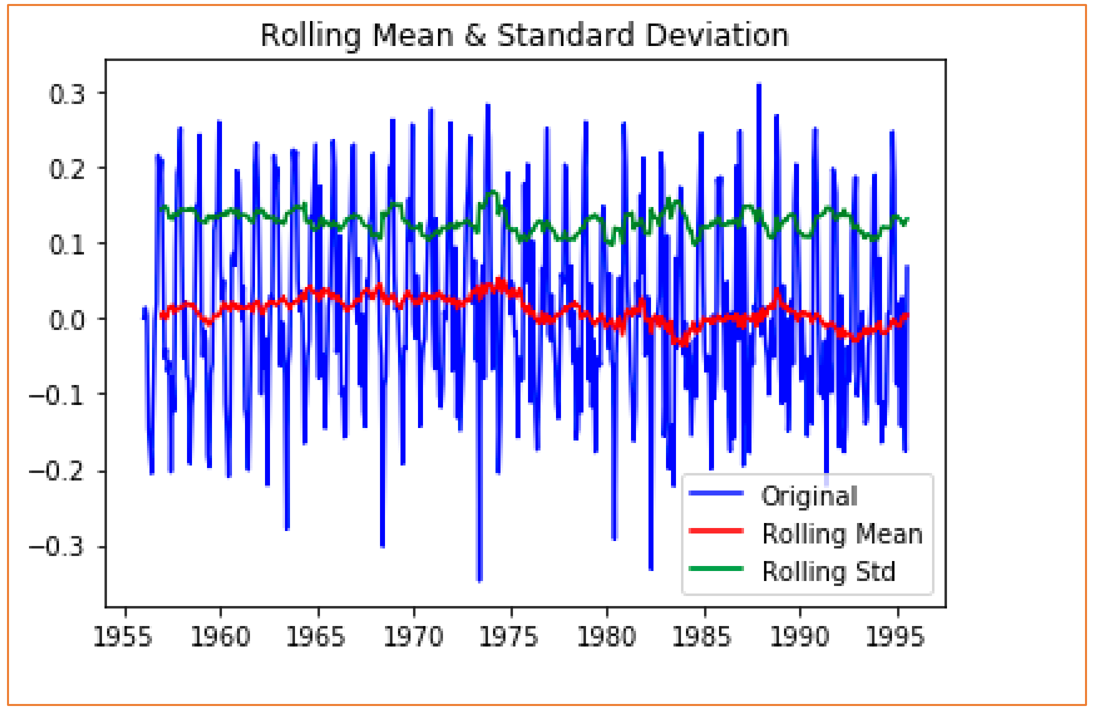

```
Test Statistic                  -2.282661
p-value                          0.177621
#Lags Used                      17.000000
Number of Observations Used    458.000000
Critical Value (1%)             -3.444709
Critical Value (5%)             -2.867871
Critical Value (10%)            -2.570142
dtype: float64
```

The test statistic is more than critical value and the moving average is not constant over time.

So, the null hypothesis of the Dickey-Fuller test cannot be rejected. This shows that the time series is not stationary.

Step 4: Stationarize

```
ts_log_mv_diff = pd.rolling_mean(data['Monthly beer production in
Australia].apply(lambda x: math.log(x)),2).diff(1)
ts_log_mv_diff.dropna(inplace=True)
ts_log_mv_diff.plot()
```



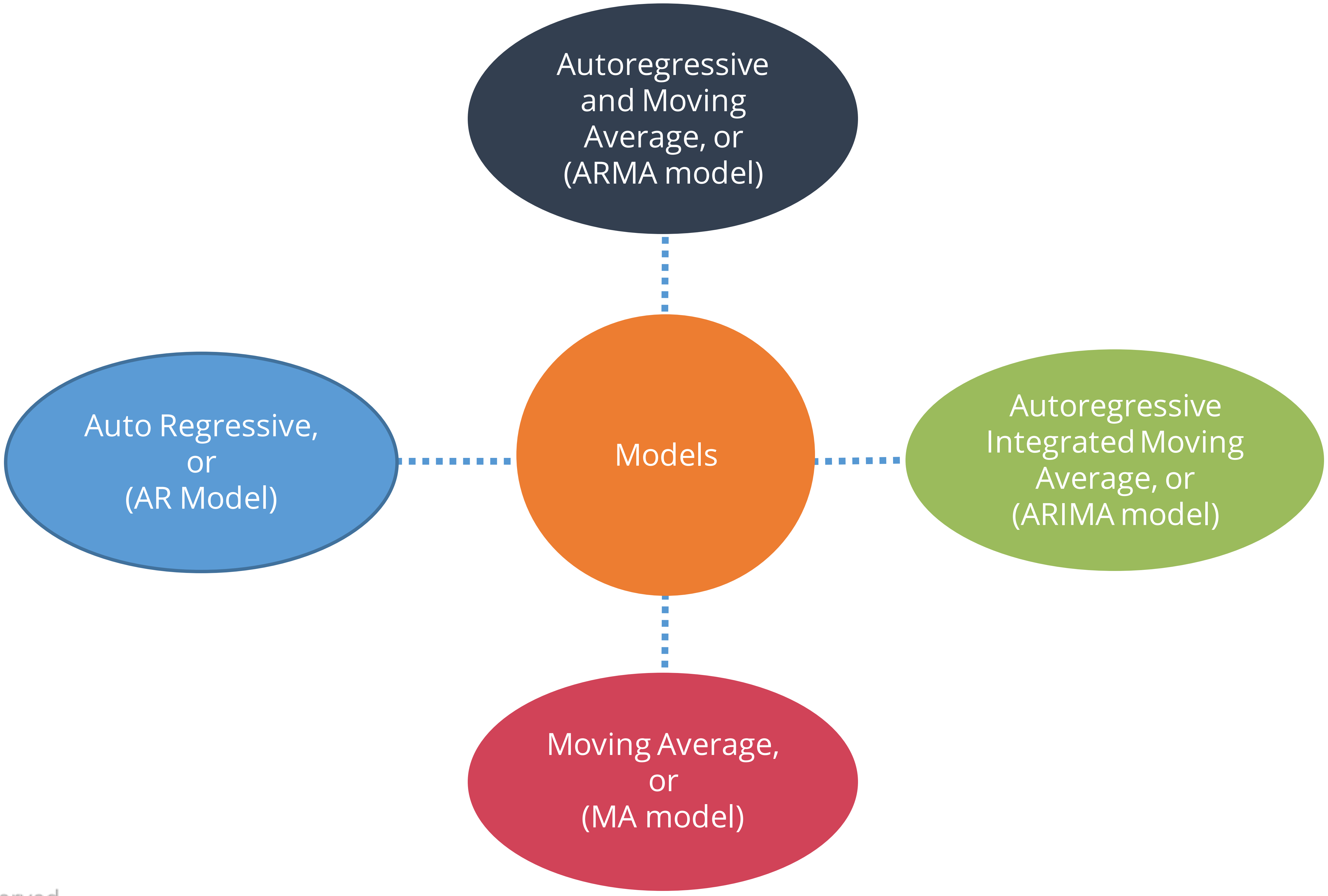

```
Test Statistic                  -3.303161
p-value                          0.014738
#Lags Used                      18.000000
Number of Observations Used    452.000000
Critical Value (1%)             -3.444900
Critical Value (5%)             -2.867956
Critical Value (10%)            -2.570187
dtype: float64
```

Test statistic < 5 % of critical value. **Reject** null hypothesis

Time Series Modeling

Topic 3: Various Time Series Models

Time Series Models



Auto Regressive (AR) Model

In an AR model, you predict future values based on a weighted sum of past values.

Equation for the auto regressive model :

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p Y_{t-p} + e_t$$

Y_t is the function of different past values of the same variable
e_t is the error term
c is a constant
φ_1 to φ_p are the parameters

AR(1) is a model whose current value is based on the preceding value

AR(2) is based on the preceding two values

Day	Price	
1	21	y_{t-p}
2	22	.
3	23	.
4	24	.
5	23	.
6	26	.
7	27	.
8	27	.
9	29	y_{t-3}
10	30	y_{t-2}
11	32	y_{t-1}
12	?	y_t

Moving Average (MA) Model

MA model is used to forecast time series if Y_t depends only on the random error terms.

Equation for the MA model :

$$Y_t = \mu + \varphi_1 E_{t-1} + \varphi_2 E_{t-2} + \cdots + \varphi_p E_{t-p}$$

Y_t is the function of different past error terms
μ is the mean of the series
E_t is the error term
φ_1 to φ_p are the parameters

The error terms here are assumed to be white noise processes with mean zero and constant variance.

Year	Units	Moving Avg.
1994	2	—
1995	5	3
1996	2	3
1997	2	3.67
1998	7	5
1999	6	—

ARMA Model

ARMA model is used to forecast time series using both the past values and the error terms.

Equation for the ARMA model :

$$\underbrace{Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p Y_{t-p} + e}_{\text{Autoregressive part}} + \underbrace{\mu + E_t + \varphi_1 E_{t-1} + \varphi_2 E_{t-2} + \text{---------} + \varphi_p E_{t-p}}_{\text{Moving Average part}}$$

Autoregressive part ⊕ Moving Average part = ARMA

It is referred as ARMA (p, q), where p is autoregressive terms and q is moving average terms

ARIMA Model

ARIMA model predicts a value in a response time series as a linear combination of its own past values, past errors, also current and past values of other time series.

ARIMA (p, d, q)

- p is the order of the autoregressive part
- d is the order of the differencing
- q is the order of the moving-average process

If no differencing is done (d = 0), the models are usually referred to as ARMA(p, q) models

ACF and PACF

Autocorrelation refers to the way the observations in a time series are related to each other.

Autocorrelation Function (ACF)

ACF is the coefficient of correlation between the value of a point at a current time and its value at lag p, that is, correlation between Y(t) and Y(t-p)

ACF will identify the order of MA process

Partial Autocorrelation Function (PACF)

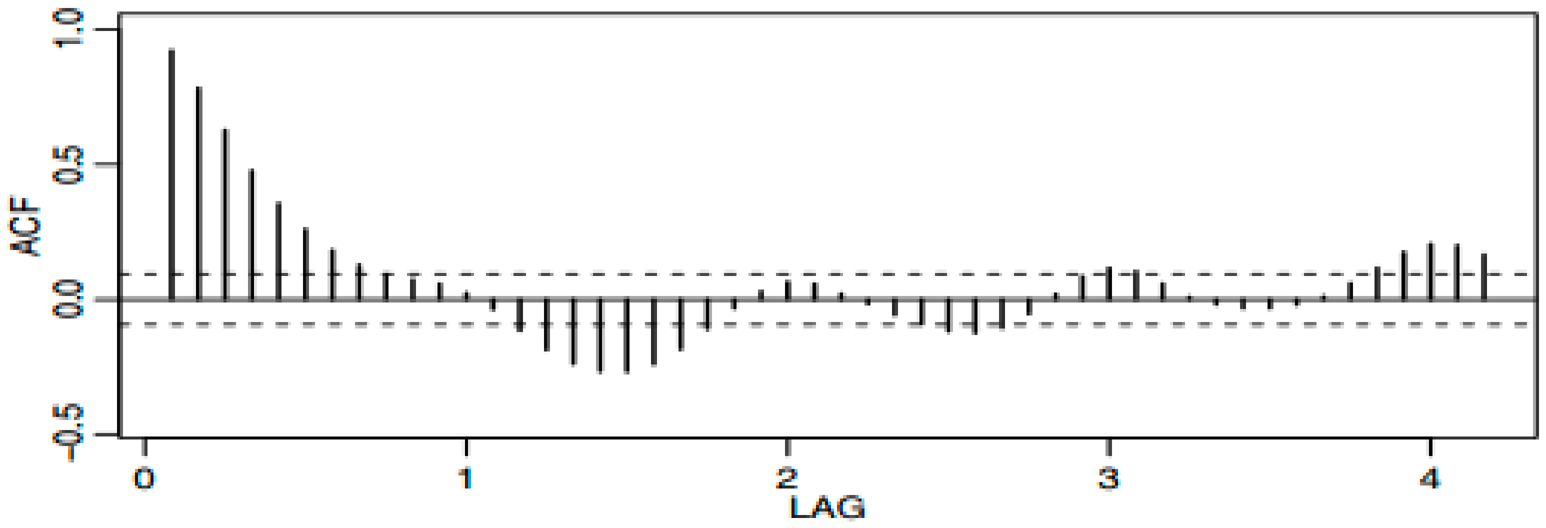
PACF is similar to ACF, but the intermediate lags between t and t-p are removed, that is, correlation between Y(t) and Y(t-p) with p-1 lags excluded.

PACF will identify the order of AR process

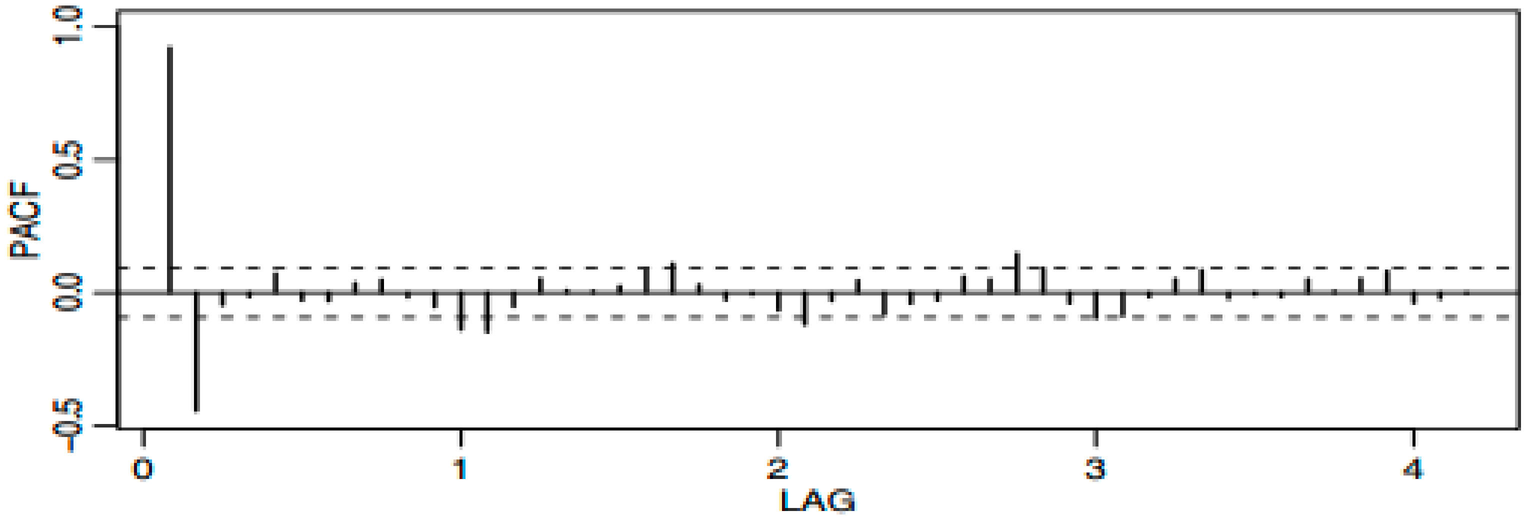
ACF and PACF are used to determine the value of p and q

Characteristics of ACF and PACF

MODEL	ACF	PACF
AR(p)	Spikes decay towards zero	Spikes cutoff to zero
MA(q)	Spikes cutoff to zero	Spikes decay towards zero
ARMA(p,q)	Spikes decay towards zero	Spikes decay towards zero



ACF “decays" to zero



PACF "cuts off“ to zero after the 2nd lag

Steps in Time Series Forecasting

Step 01	Visualize the time series – check for trend, seasonality, or random patterns
Step 02	Stationarize the series using decomposition or differencing techniques
Step 03	Plot ACF / PACF and find (p, d, q) parameters
Step 04	Build ARIMA model
Step 05	Make predictions using final ARIMA model

Assisted Practice

Modeling

Duration: 15 mins.

Problem Statement: The Air Passenger dataset provides monthly total of US airline passengers, from 1949 to 1960. This dataset is of a time series class

Objective:

- Perform ARIMA modeling in Python after obtaining ACF and PACF plots

Access: Click on the **Labs** tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

simplilearn

Unassisted Practice

Modeling

Duration: 15 mins.

Problem Statement: : The Beer production dataset provides a time series data for monthly beer production in Australia, for the period Jan 1956 – Aug 1995

Objective:

- Perform ARIMA modeling in Python after obtaining ACF and PACF plots

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

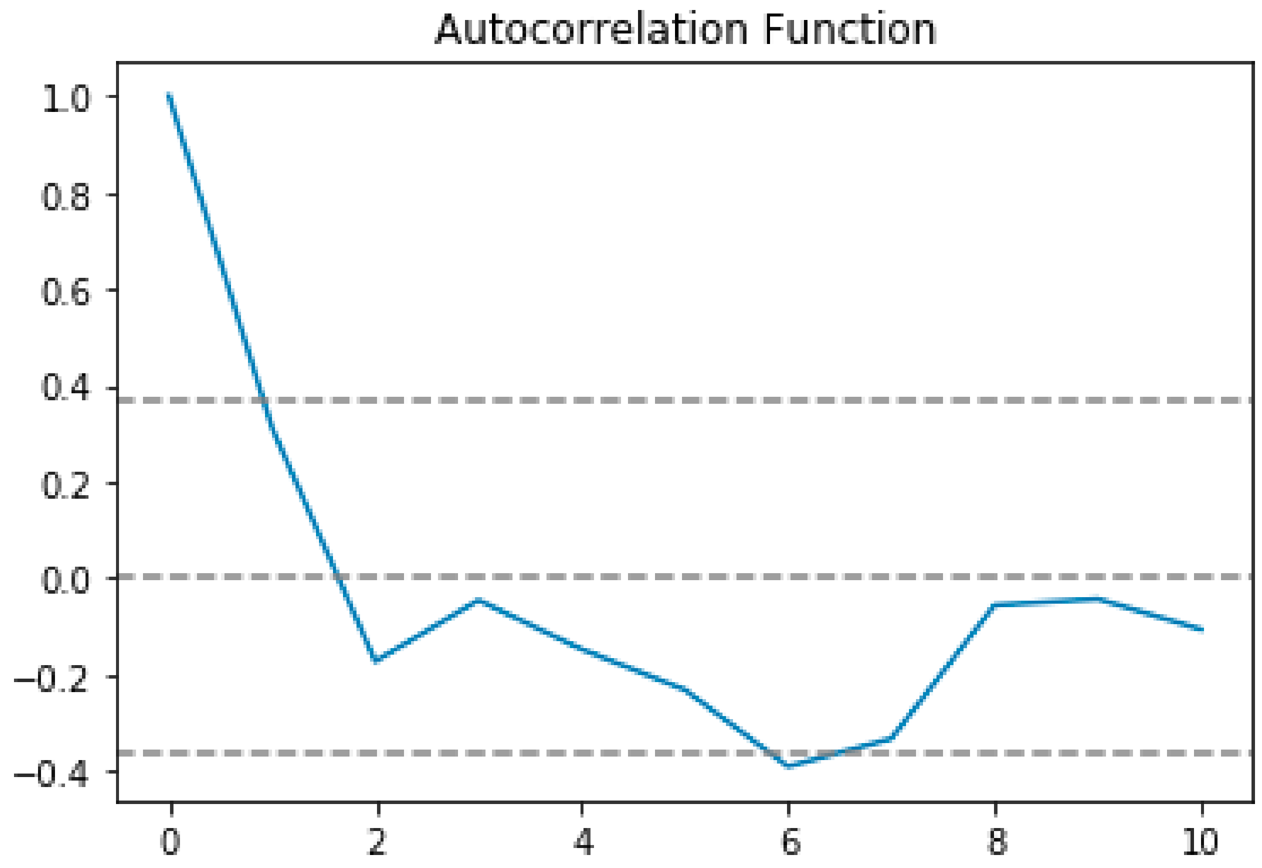
ACF and PACF

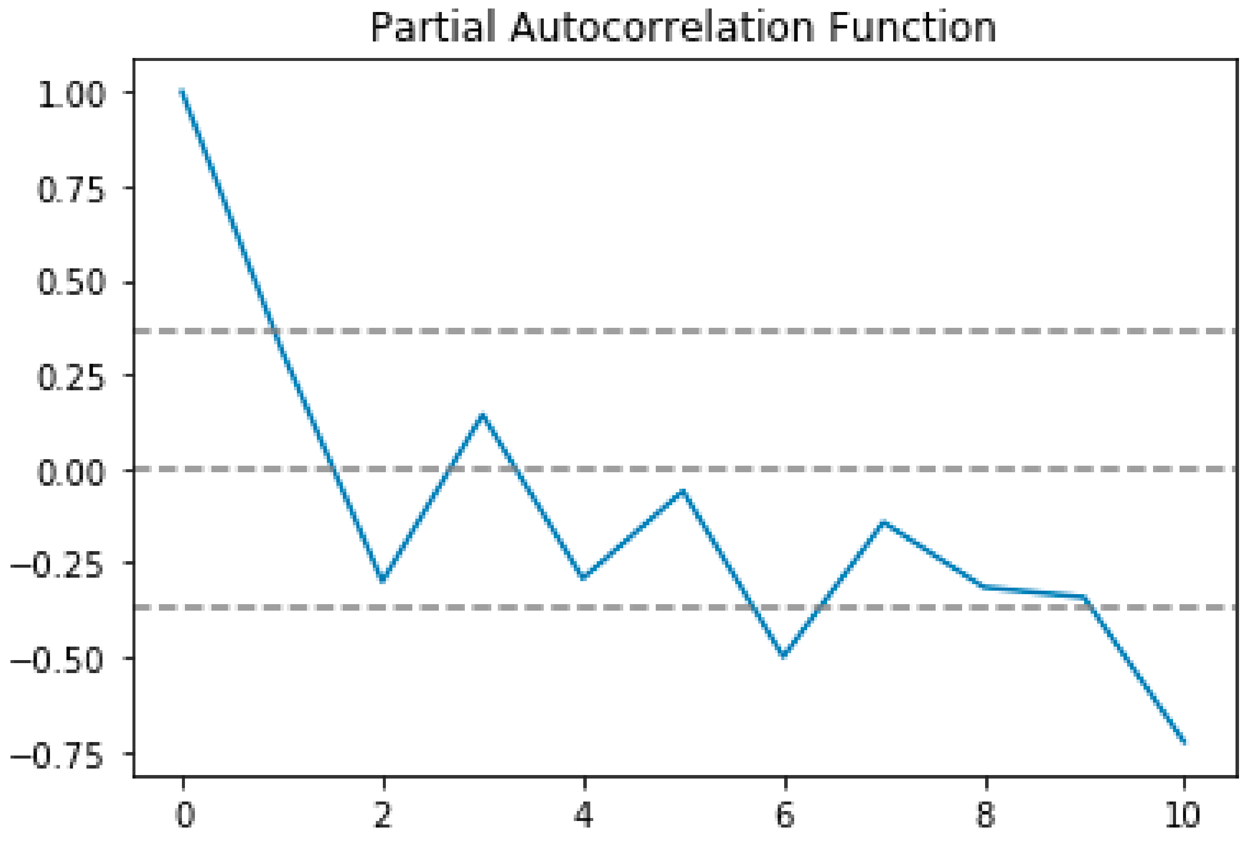
Code

```
plt.plot(np.arange(0,11), acf(ts_log_mv_diff, nlags = 10))
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-7.96/np.sqrt(len(ts_log_mv_diff)),linestyle='--',color='gray')
plt.axhline(y=7.96/np.sqrt(len(ts_log_mv_diff)),linestyle='--',color='gray')
plt.title('Autocorrelation Function')
plt.show()

plt.plot(np.arange(0,11), pacf(ts_log_mv_diff, nlags = 10))
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-7.96/np.sqrt(len(ts_log_mv_diff)),linestyle='--',color='gray')
plt.axhline(y=7.96/np.sqrt(len(ts_log_mv_diff)),linestyle='--',color='gray')
plt.title('Partial Autocorrelation Function')
plt.show()
```

Output





ACF curve crosses the upper confidence value when the lag value is between 0 and 1
Thus, optimal value of q in the ARIMA model must be 0 or 1

The **PACF** curve drops to 0 between lag values 1 and 2
Thus, optimal value of p in the ARIMA model is 1 or 2

ARIMA

Code

```
model = ARIMA(ts_log, order=(1, 1, 0)) results_ARIMA = model.fit(disp=-1)
plt.plot(ts_log_mv_diff) plt.plot(results_ARIMA.fittedvalues, color='red')
plt.title('RSS: %.4f’% (((results_ARIMA.fittedvalues[1:] -
ts_log_mv_diff)**2).mean()))
predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
predictions_ARIMA_diff.head()
predictions_ARIMA_diff_cumsum = predictions_ARIMA_diff.cumsum()
predictions_ARIMA_diff_cumsum.head()
predictions_ARIMA_log = pd.Series(ts_log.ix[0], index=ts_log.index)
predictions_ARIMA_log =
predictions_ARIMA_log.add(predictions_ARIMA_diff_cumsum,fill_value=0)
predictions_ARIMA_log.head()
```

```
Month
1956-02-15    0.000936
1956-03-15   -0.005458
1956-04-15    0.003012
1956-05-15    0.048189
1956-06-15    0.019847
dtype: float64
```

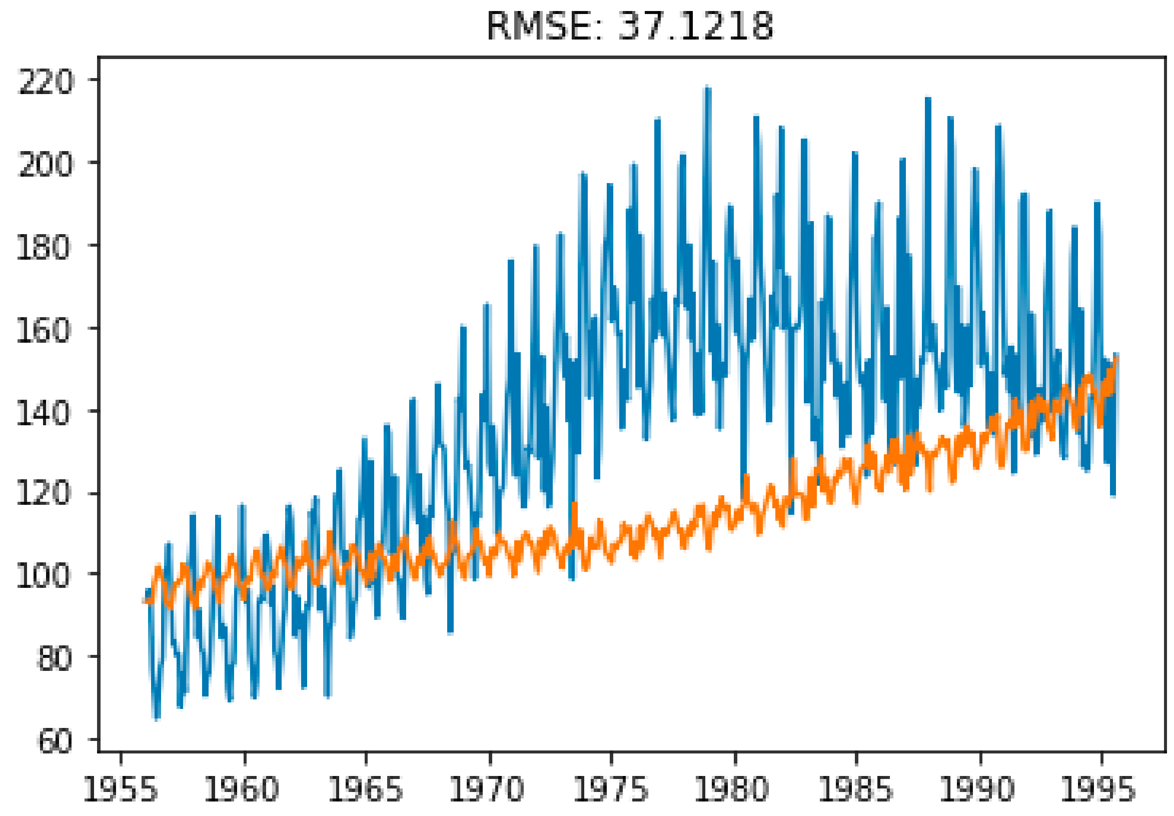
```
Month
1956-02-15    0.000936
1956-03-15   -0.004522
1956-04-15   -0.001510
1956-05-15    0.046680
1956-06-15    0.066527
dtype: float64
```

```
Month
1956-01-15    4.534748
1956-02-15    4.535684
1956-03-15    4.530226
1956-04-15    4.533238
1956-05-15    4.581428
dtype: float64
```

ARIMA

Code

```
predictions_ARIMA = np.exp(predictions_ARIMA_log)
plt.plot(ts)
plt.plot(predictions_ARIMA)
plt.title('RMSE: %.4f'% np.sqrt(((predictions_ARIMA-ts)**2)/(ts)).mean())
```



Key Takeaways

Now, you are able to:

- Understand time series analysis
- Build time series models using ARIMA

Knowledge Check

Knowledge Check

1

Which of the following cannot be a part of time series data?

a. Trend

b. Seasonality

c. Noise

d. None of the above

Knowledge Check 1	Which of the following cannot be a part of time series data?

a. Trend

b. Seasonality

c. Noise

d. None of the above

The correct answer is **d. None of the above**

Options a, b, c are time series components.

Knowledge Check 2

Which of the following techniques can be used to make a series stationary?

a. Transformation

b. Differencing

c. Decomposition

d. All of the above

Knowledge Check
2

Which of the following techniques can be used to make a series stationary?

a. Transformation

b. Differencing

c. Decomposition

d. All of the above

The correct answer is **d. All of the above**

All of these techniques are used to stationarize a time series

Lesson-End Project

IMF Commodity Price Forecast

Duration: 20 mins.

Problem Statement: You are provided with a dataset which consists of Zinc prices for the period Jan 1980 – Feb 2016

Objective:

- Visualize the time series
- Check for the stationarity of your data using Rolling Statistics and Dickey fuller test and if present, remove it using stationarity removal techniques
- Plot ACF and PACF plots. Find p, d, q values
- Perform ARIMA modeling
- Forecast the prices using the new model

Access: Click on the Labs tab on the left side panel of the LMS. Copy or note the username and password that are generated. Click on the Launch Lab button. On the page that appears, enter the username and password in the respective fields, and click Login.

Thank You