Tools Used

1. *Programming language* : **Python**
2. *Libraries* : **pandas**, **numpy**, **matplotlib**, **statsmodels**, **sklearn**, **seaborn**
3. **Jupyter notebook**
4. **Git bash** and **Github**

Introduction

Different Types of data -

- **Cross-sectional data** : Cross sectional data can obtained by taking multiple observation from multiple individuals at same point in time.
- **Timeseries data** : Timeseries data can obtained by taking multiple observations from same source at different points of time.
- **Panel data** : Panel data is collection of multiple observations over multiple points in time. It is combination of cross-sectional data and Time-series data.

Introduction of dataset :

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1987 entries, 2010-04-01 to 2018-03-28
Data columns (total 6 columns):
Open                 1987 non-null float64
High                 1987 non-null float64
Low                  1987 non-null float64
Close                1987 non-null float64
Shares Traded        1987 non-null int64
Turnover (Rs. Cr)    1987 non-null float64
dtypes: float64(5), int64(1)
memory usage: 108.7 KB
```

- The **Nifty50** data that used is Time series data from **APR-2010** to **MAR-2018**.
- In this dataset we have
  - DatetimeIndex
  - 6 columns
  - 1987 entries in each column
  - No null values

Internal structure of time series

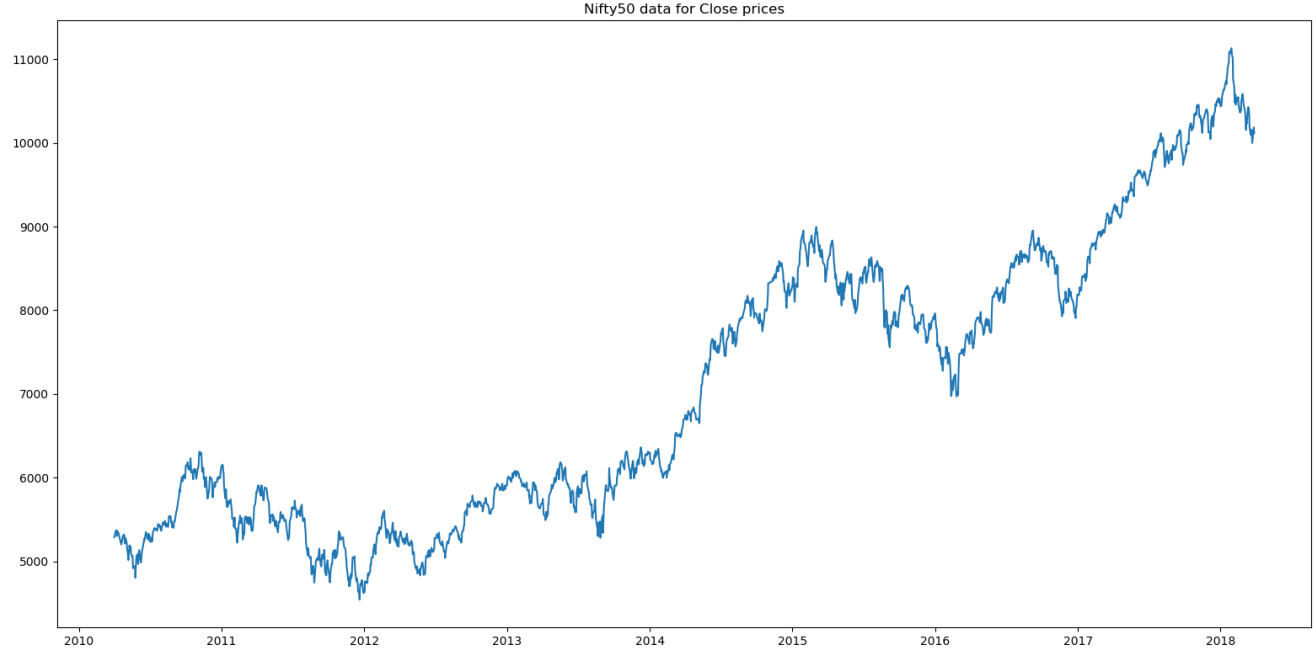
A Time series is a combination of General trend, Seasonality, Cyclic movements and Unexpected variations.

- A timeseries can be expressed as $x_t = f_t + s_t + c_t + e_t$

Where,

- f_t is Trend component
- s_t is Seasonal component
- c_t is cyclic component
- e_t is irregular component
- t is the time index at which observations about the series have been taken

- **General Trend** : When there is Upward or downward movement present in data in a long run, is Known as general trend.



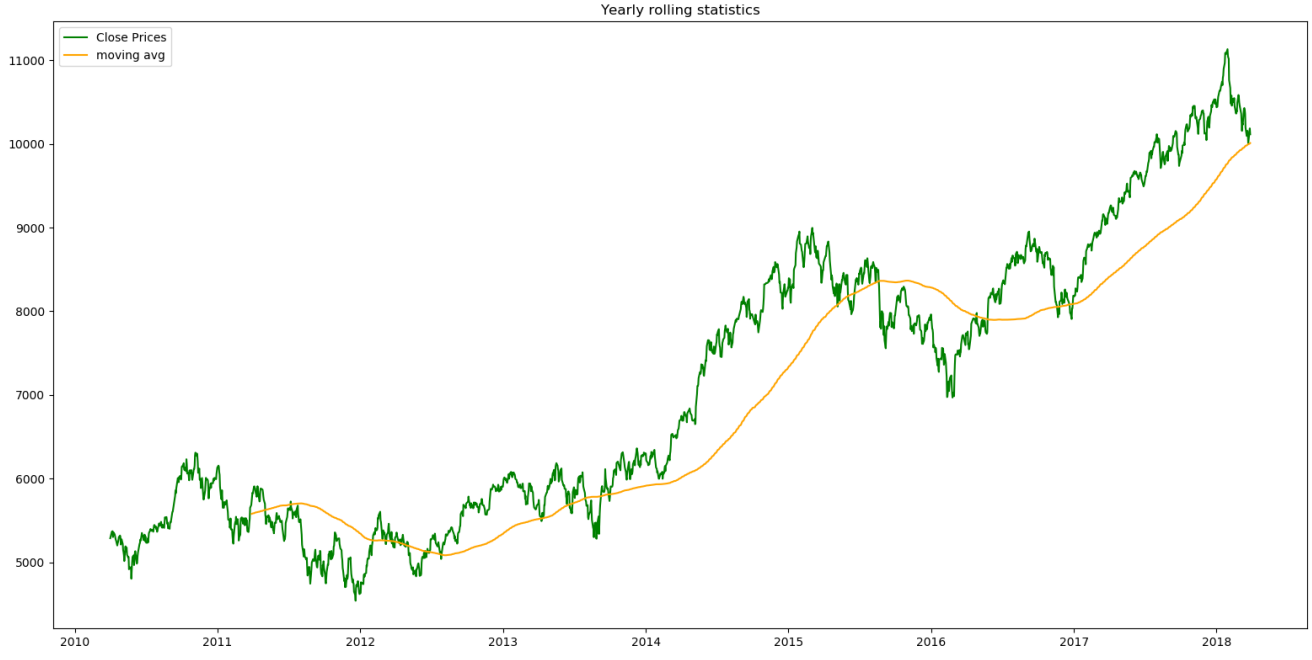
- **Seasonality** : If repetitive patterns present in data which occurs over known periods of time are known as seasonality. Mostly, presence of seasonality can be reveals by exploratory data analysis.
- **Cyclical movements** : If there are movements observes after every few units of time and do not have fixed periods of variations are known as cyclic movements.
- **Unexpected variations** : Occurance of sudden changes in time series which are rarely repeted. This component also known as residuals.

Stationary time series

A timeseries is known as stationary when it is free from Trend and seasonility. Its statistical properties like mean, variance, autocorrelation etc are constant over time.

- **check stationarity of timeseries** : To check stationarity of timeseries we can-
    i. **Plot Rolling statistics of timeseries**
    ii. **Apply Augmented Dickey Fuller test**

- By plotting Rolling statistics we can easily identify trend component.



- Augmented Dickey fuller test is statistical test to check the stationarity of timeseries. It uses null hypothesis testing where H_0 rejected if p-value is greater than 0.05.

```
Test Statistic                  -0.371803
p-value                          0.914701
#Lags Used                       1.000000
Number of Observations Used    1985.000000
Critical Value (1%)             -3.433649
Critical Value (5%)             -2.862997
Critical Value (10%)            -2.567546
dtype: float64
Time Series is not stationary
```

Methods to detrending data

1. Differencing
2. Regression
3. Statistical function

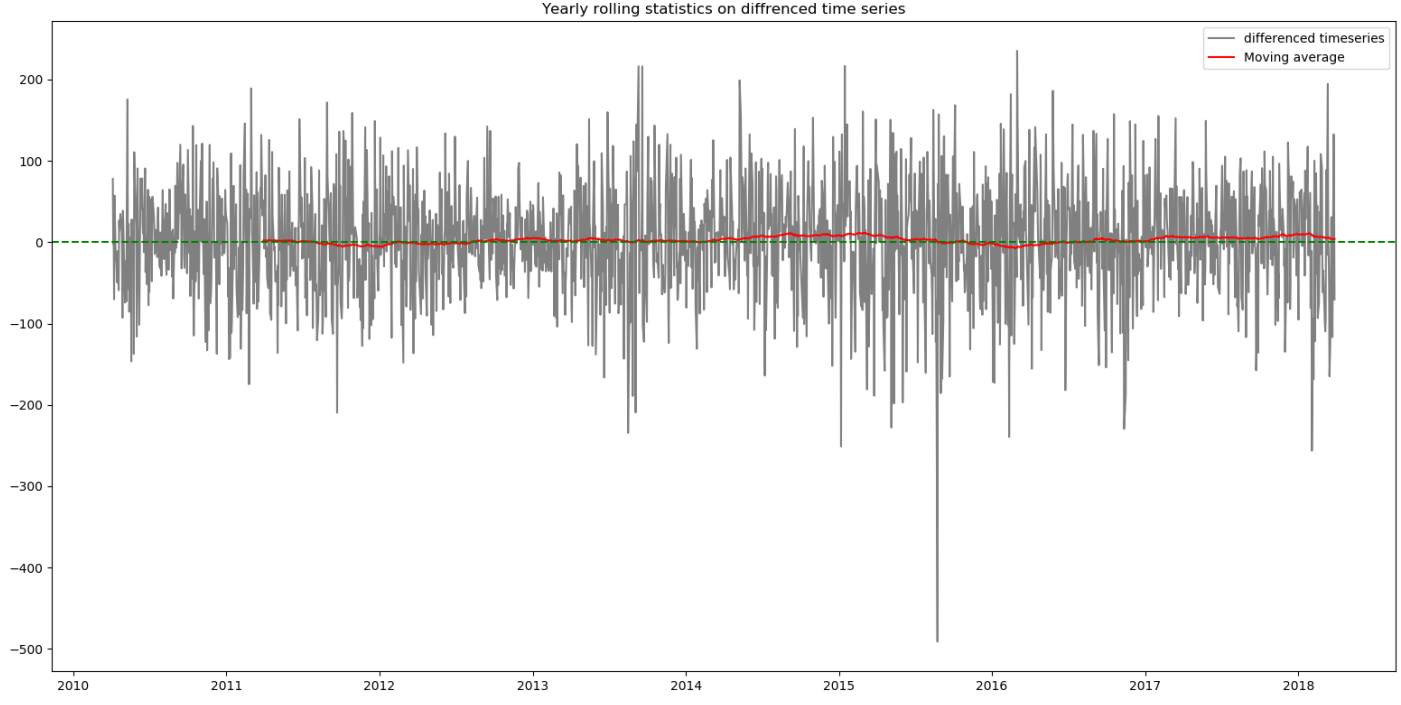
- **Differencing** : Differencing is processs of taking difference original timeseries with itself by lag.
  example of time series with lag 1 -

  > $\Delta x_t = x_t - x_{t-1}$

  Where, Δx_t is stationary time series.

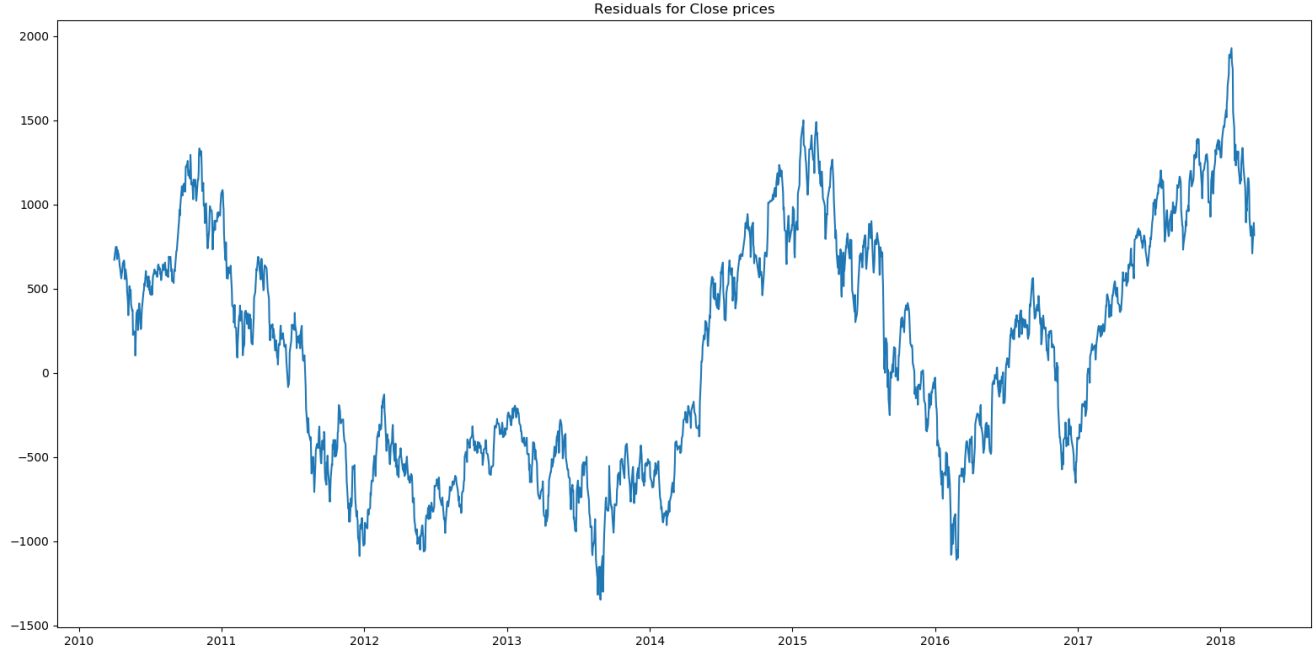
  x_t is original time series.

  x_{t-1} is time series with lag 1.

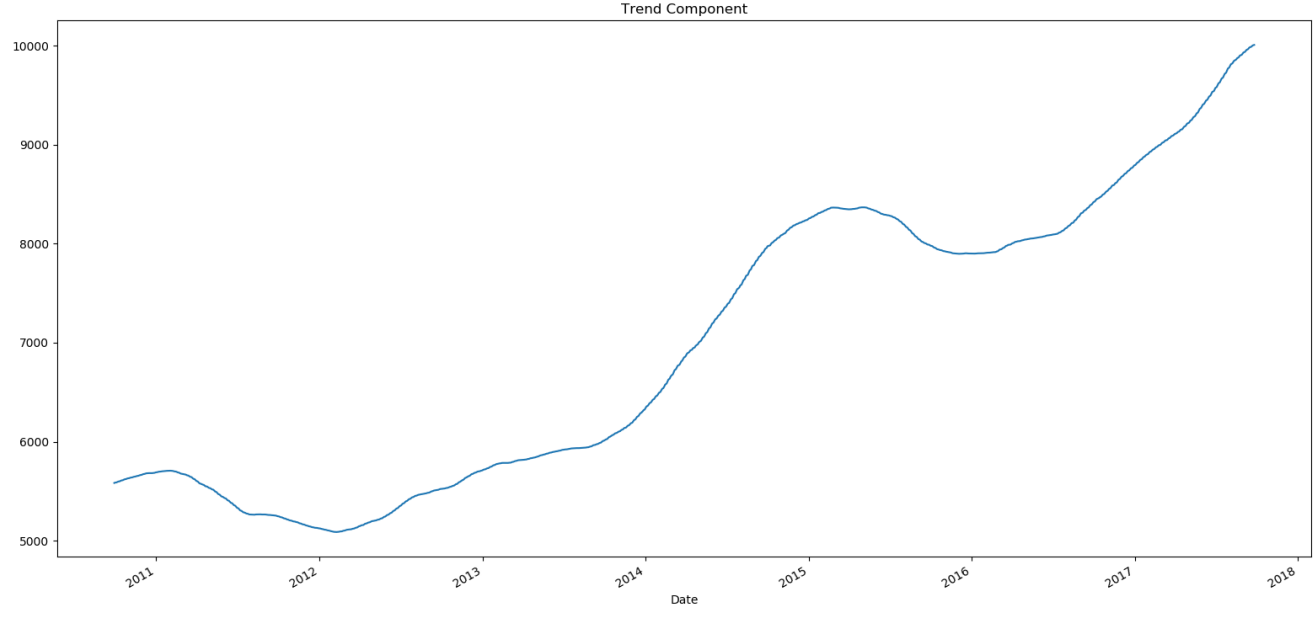

Yearly rolling statistics on diffrenced time series
differenced timeseries
Moving average
200
100
0
−100
−200
−300
−400
−500
2010
2011
2012
2013
2014
2015
2016
2017
2018

- **Regression** : Regression is useful to find trend line and to remove trend component, take difference between original time series and trend line.

- After removing trend we will get Residuals.



- **Statistical function** : In python a function named `seasonal_decompose` is present in library `statsmodels.tsa.seasonal` which separate Observed data(i.e. original data), trend component, seasonal component and residuals.



Remove seasonality

- While removing trend by using differencing, seasonality was also removed.
- `seasonal_decompose()` function return seasonal component also that can be access by `Result.seasonal`.
- To remove seasonality, if regression is used to detrending data, take average of detrended data for specific season.

Forecasting

There are many Statistical models for timeseries forecasting. Among them ARIMA is widely used model which is combination of Autoregressive,Integration(differencing) and Moving average models.

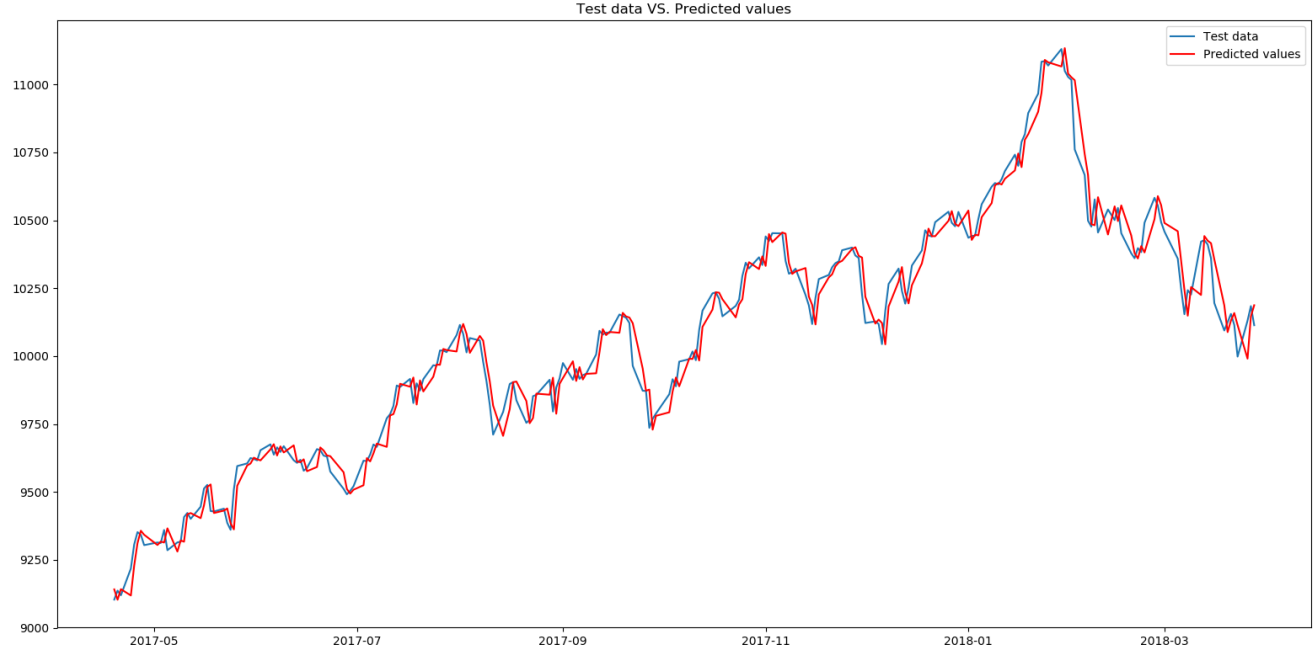
- **Autoregression** : This model gives output which depends on its own previous values.
- **Differencing** : Integration or differencing makes series sattionary.
- **Moving Average** : This model analyze data points by creating series of averages of subsets of data.

- ARIMA models are generally denoted by ARIMA(p,d,q) where,
- p is order(no. of time lags) of Autoregressive term.
- d is the order of differencing(the number of times the data have had past values subtracted).
- q is the order of moving average model.

Value of p and q is selected by autocorrelation plot partial autocorrelation plot.

- The value of p will be the lag value where the PACF chart crosses the upper confidence interval for the first time.
- The value of q will be the lag value where the ACF chart crosses the upper confidence interval for the first time.

- After fitting model .forecast() method is used.



- The accuracy of ARIMA model depends on the value of r^2.
- In statistics r^2 is known as cofficient of determination which is square of correlation Coefficient.
- Best possible r^2 can be 1.0
- r^2 can be negative because the model can be worse.
- In python r^2 can be calculated usong `.r2_score()` method which is present in `sklearn.metrics` package.