1. **Loading data and Pre-processing**

**import** packages numpy**,**pandas

#reads .csv file

fields **=** **[**'iso\_code'**,** 'location'**,**'date'**,**'date'**,**'new\_cases'**,**'new\_tests'**,**'total\_tests'**]**

df **=** pd**.**read\_csv**(**'owid-covid-data.csv'**,** skipinitialspace**=True,** usecols**=**fields**)**

#Cleaning for Location column

**for** **all** records of Location column**:**

**if(**Location has null**/**invalid value**):**

#check if iso\_code has 'MYS'

**if(**'iso\_code' **for** that row **is** "MYS"**):**

replace country name **as** "Malaysia"

**else:**

delete the record

Filter dataframe **for** records of "Malaysia"

#Clean Column Date

**if(**Date has null**/**invalid value**):**

**if(**total cases has null**/**invalid value**):**

delete the record

**else:**

Arrange data **in** ascending order of 'total\_cases'

derive **and** replace the missing date **from** the sequence**.**

#Clean new\_cases

**if(**column new\_cases has null**/**invalid value**):**

**if(**total\_cases has null**/**invalid value**):**

delete the record

**else:**

Arrange data **in** ascending order of 'total\_cases'

derive **and** replace the missing new\_cases by a difference of current **and** previous row of column total\_cases**.**

#Clean new\_tests

**if(**column new\_tests has null**/**invalid value**):**

**if(**total\_tests has null**/**invalid value**):**

delete the record

**else:**

Arrange data **in** ascending order of 'total\_tests'

derive **and** replace the missing new\_cases by a difference of current **and** previous row of column total\_tests**.**

####Data Cleaning is complete and now we can start with objective 1 & Objective 2

#Objective 1:

#Considering current trend of COVID-19 cases in Malaysia, I want to determine if Malaysia could achieve 14 days average of below 100 cases/day within the next 3 months(Cumulative count of new cases over 14 consecutive days should be less than 1400).

**import** packages warnings**,** itertools**,** numpy**,** matplotlib**,** pandas**,**statsmodels**,** pmdarima

Convert Column 'date' to DataTime **format** using Pandas

index daraframe by column date**.**

extract columns 'date' **and** 'new\_cases' **in** dataframe "df\_objective1"

df\_objective1**.**plot**(**Trend**,**Seasonality**,** **and** Noise graph to analyse the data**)**

divide df\_objective1 **in** ratio of 70**:**30 **in** train\_df\_objective1 **and** test\_df\_objective1 dataframe

**from** library pylab **import** package rcParams

using train\_df\_objective1 create a decomposition graph

plot the graph**.**

**and** show the graph

**if(for** train\_df\_objective1 Autocorrelation function Function**(**ACF**)** **is** **not** Stationary**):**

use differencing method

Verify Augmented Dickey**-**Fuller **(**ADF**)** **for** trend

Verify Partial Autocorrelation function **(**PACF**)** **for** seasonality

**if(**P value less than **or** **=** 0.05**):**

Plot Seasonality **for** lag of 14 days

Plot graph Partial Autocorrelation function **(**PACF**)**

Plot graph Autocorrelation function Function**(**ACF**)**

Extract value of Probable Autoregression **(**AR**)** **and** Probable Moving Average**(**MA**)**

Extract value of Seasonal Autoregression **(**AR**)** **and** Seasonal Moving Average**(**MA**)**

Create a SARIMA Model 'sarima\_model' using method sm**.**tsa**.**arima**.**SARIMA**().**

FIT the model

Evaluate model using Mean Squared Error**(**MSE**)** **and** Root Mean Squared Error **(**RMSE**)**

**if(**RMSE between 0.2 **and** 0.5 **):**

Accept Model

Predict result **for** new\_cases **for** each day **for** **next** 90 days**.**

Calcualte moving average **for** 14 days

**if** **sum** of consecutive 14 days **<**1400

Malaysia Can achieve average less than 100 cases per day **in** **next** three months

**else:**

Malaysia Can **not** achieve average less than 100 cases per day **in** **next** three months

**else:**

restart modelling

#Objective 2:

#Considering current trend of COVID-19 tests in Malaysia, I want to determine the number of COVID-19 test kits that would be required each week in Malaysia, over next 3 months.

extract columns 'date' **and** 'new\_tests' **in** dataframe "df\_objective2"

df\_objective2**.**plot**(**Trend**,**Seasonality**,** **and** Noise graph to analyse the data**)**

divide df\_objective2 **in** ratio of 70**:**30 **in** train\_df\_objective2 **and** test\_df\_objective2 dataframe

**from** library pylab **import** package rcParams

using train\_df\_objective2 create a decomposition graph

plot the graph**.**

**and** show the graph

**if(for** train\_df\_objective2 Autocorrelation function Function**(**ACF**)** **is** **not** Stationary**):**

use differencing method

Verify Augmented Dickey**-**Fuller **(**ADF**)** **for** trend

**if(**P value less than **or** **=** 0.05**):**

Plot graph Partial Autocorrelation function **(**PACF**)**

Plot graph Autocorrelation function Function**(**ACF**)**

Extract value of Probable Autoregression **(**AR**)** **and** Probable Moving Average**(**MA**)**

Create a ARIMA Model 'arima\_model' using method sm**.**tsa**.**arima**.**ARIMA**().**

FIT the model

Evaluate model using Mean Squared Error**(**MSE**)** **and** Root Mean Squared Error **(**RMSE**)**

**if(**RMSE between 0.2 **and** 0.5 **):**

Accept Model

Predict result **for** new\_tests **for** each day **for** **next** 90 days**.**

Calcualte number of tests lits required**:** Assuming 1 new\_test requires 1 test kit**.**

Calcualte number fo test kits required per week **for** **next** 13 weeks

**else:**

restart modelling