Forecasting of Airline Passenger Traffic

Group Id: 1

Group Members:

Dikshant Sachan (170250)
Pragyan Pandey (170478)
Rahul Bareliya (180576)
Vishnudatt Nagar (181172)

Instructor: Prof. Amit Mitra

Introduction

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. Time series are widely used for non-stationary data, like economic, weather, stock price, and retail sales in this post. We will demonstrate an approach for forecasting the Airline Passengers using its time series.

About the Data:-
We analyzed the no. of passengers in each month from the year 1949 to 1960.
The dataset can be accessed here -
https://www.kaggle.com/rakannimer/air-passengers

Roadmap to Forecasting

Our process consists of three major steps:-

1. Identification- Use the data and all related information to help select a subclass of model that may best summarize the data

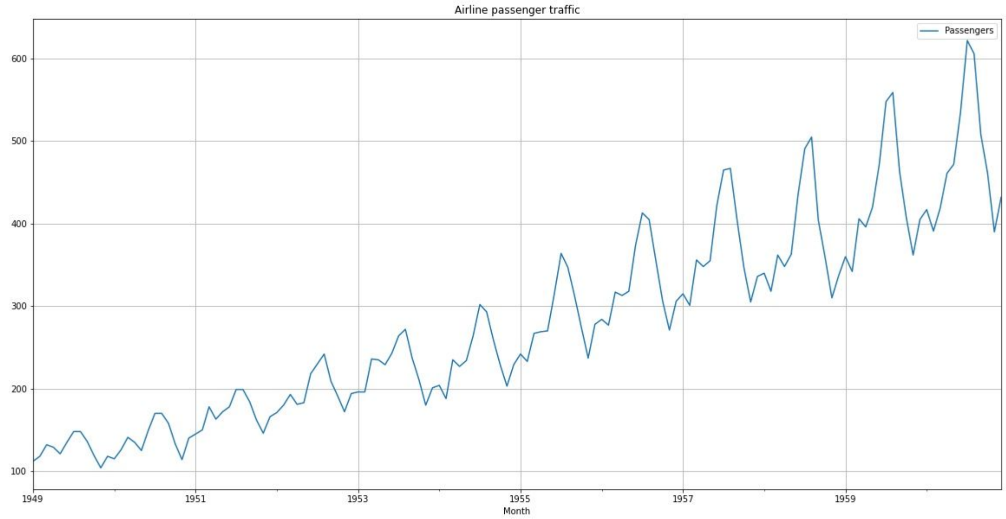
- **a.** We will apply proper transformation until the data is stationarized
- **b.** Plot the ACF and PACF of the above stationarized data for finding proper parameters of the time series process.

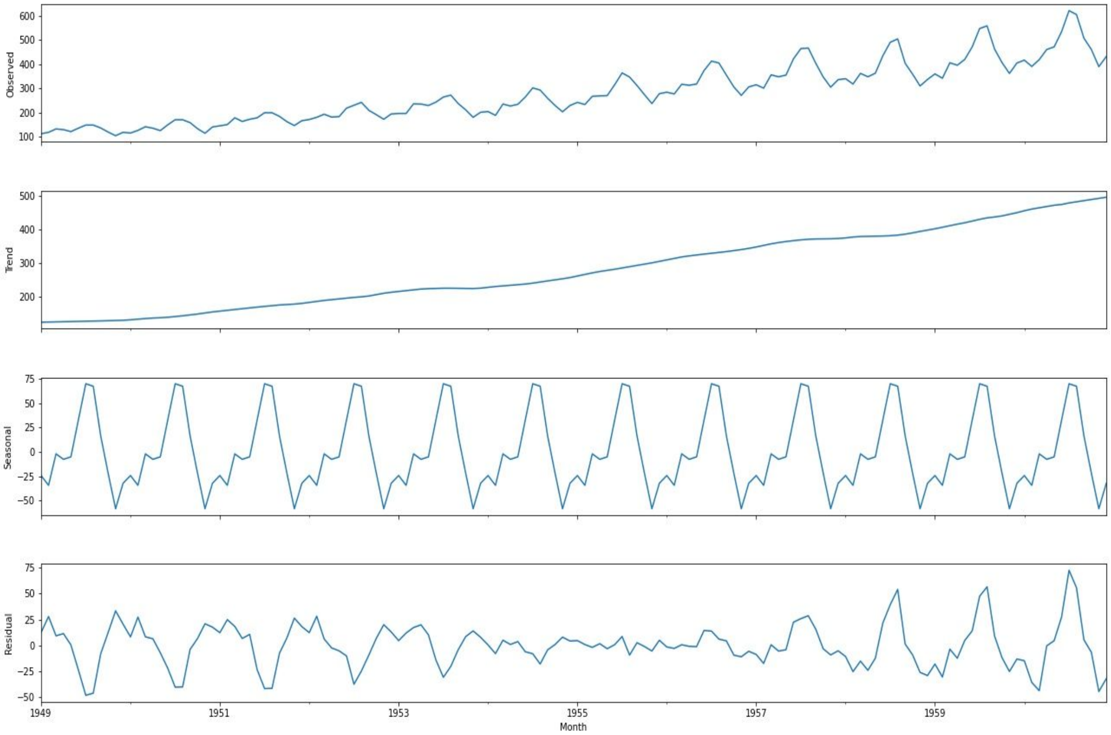
2. Estimation- Use the data to train the parameters of the model (i.e. the coefficients)

3. Diagnostic Checking- Evaluate the fitted model in the context of the available data and check for areas where the model may be improved

Visualizing the data

The first step is to visualize the data to understand what type of model we should use. We will check for the overall trend in our data. Also, look for any seasonal trends. This is important for deciding which type of model to use. We have used the predefined library functions to decompose the time series into trend, seasonal and residual components.



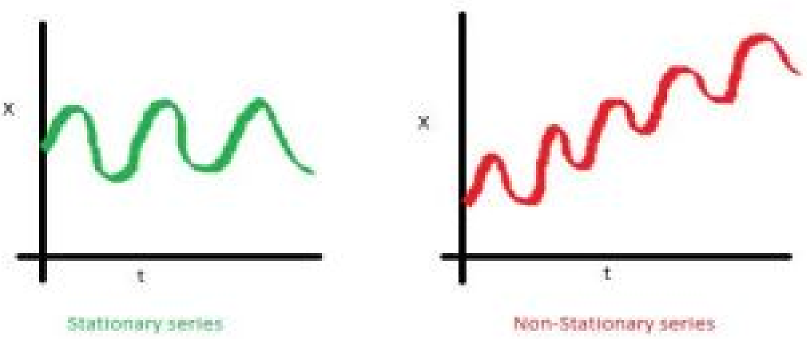


Stationarize the data:

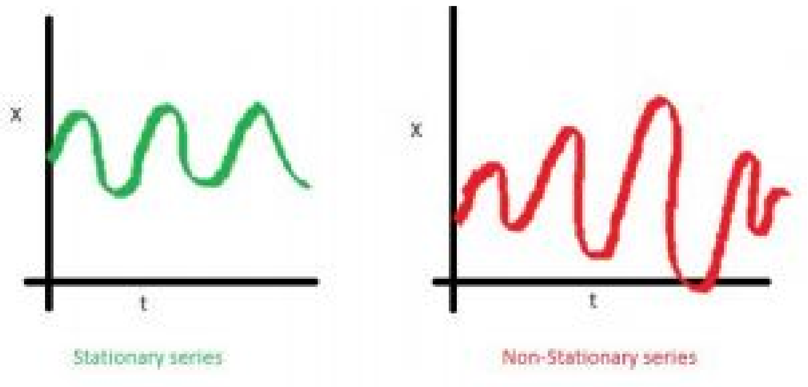
Definition- $\{X_t\}$ is said to be a stationary process if for every n, and every admissible t_1,t_2, ...t_n and any integer k, the joint distribution of $\{X_{t1},X_{t2},. . . .,X_{tn}\}$ is identical to the joint distribution of $\{X_{t1+k},X_{t2+k},. . . .,X_{tn+k}\}$.

Now, we will try to visualize what is meant by stationary time series process

1. The mean of the series should not be a function of time. The red graph below is not stationary because the mean increases over time.



2. The variance of the series should not be a function of time. This property is known as homoscedasticity. Notice in the red graph the varying spread of data over time.



3. Finally, the covariance of the ith term and the (i + m)th term should not be a function of time. In the following graph, you will

notice the spread becomes closer as the time increases. Hence, the covariance is not constant with time for the ‘red series’.
Now a very natural question is why do we need a stationary time series? The reason being, when we run a linear regression the assumption is that all of the observations are all independent of each other. In a time series, however, we know that observations are time dependent. It turns out that a lot of nice results that hold for independent random variables (law of large numbers and central limit theorem to name a couple) hold for stationary random variables. So by making the data stationary, we can actually apply regression techniques to this time dependent variable. There are two ways you can check the stationarity of a time series. The first is by looking at the data. By visualizing the data it should be easy to identify a changing mean or variation in the data. For a more accurate assessment there is the **Dickey-Fuller** test.

Dickey Fuller Test

In statistics, the Dickey-Fuller Test tests the null hypothesis that a unit root is present in an autoregressive model. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity.
Hence our null hypothesis H_0 is not stationary against our alternate hypothesis H_1 which is data stationary.

The Dickey-Fuller test is testing if $\phi=1$ in this model of the data:
$y_t = \alpha + \beta t + \phi y_{t-1} + e_t$
which is written as:

$\Delta y_t = y_t - y_{t-1} = \alpha + \beta t + \gamma y_{t-1} + e_t$
Where y_t is your data.

It is written this way so we can do a linear regression of Δy_t against t and y_{t-1} and test if γ is different from 0. If $\gamma=0$, then we have a random walk process. If not and $-1<1+\gamma<1$, then we have a stationary process.

We apply the test to check whether our hypothesis is true or not, we look for a p-value in the test, and if the p-value is less than a specific significant level often 0.05 or 0.01, we reject our null hypothesis and thus making our time series stationary.

So now we need to transform the data to make it more stationary. There are various transformations you can do to stationarize the data.

- Logarithmic - Converts multiplicative patterns to additive patterns and/or linearize exponential growth. Converts absolute change to percentage changes. Often stabilizes the variance of data with compound growth, regardless of whether deflation is also used.
- First Difference - Converts “levels” to “changes”. (X_t -X_{t-1})
- Seasonal Difference - Convert “levels” to “seasonal changes”.(X_t -X_{t-s})

Now we will apply various transformations recursively until we obtain a stationary time series according to the Dickey-Fuller test.

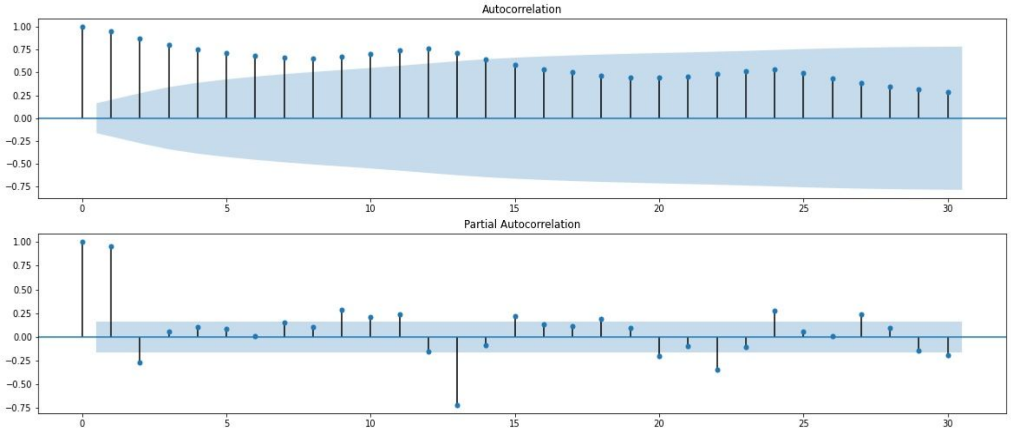
Plot the ACF and PACF charts and find the optimal parameters

- Autocorrelation Function (ACF). The plot summarizes the correlation of an observation with lag values. The x-axis shows the lag and the y-axis shows the correlation coefficient between -1 and 1 for negative and positive correlation.

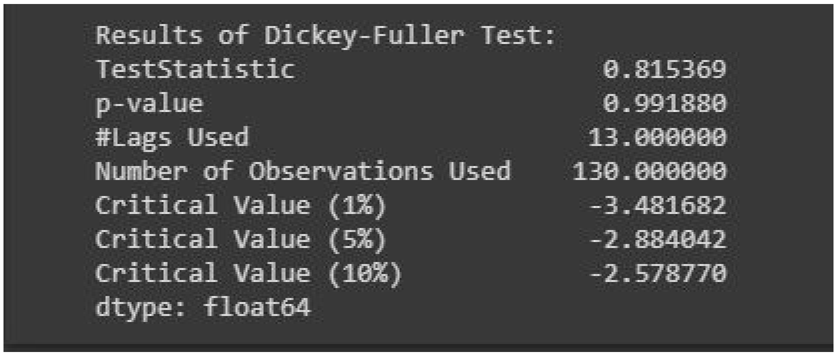
- Partial Autocorrelation Function (PACF). The plot summarizes the correlations for an observation with lag values that is not accounted for by prior lagged observations. Some useful patterns you may observe on these plots are:

  - The model is AR if the ACF trails off after a lag and has a hard cut-off in the PACF after a lag. This lag is taken as the value for p.
  - The model is MA if the PACF trails off after a lag and has a hard cut-off in the ACF after the lag. This lag value is taken as the value for q.
  - The model is a mix of AR and MA if both the ACF and PACF tail off.

We plot the ACF and PACF of the original data.



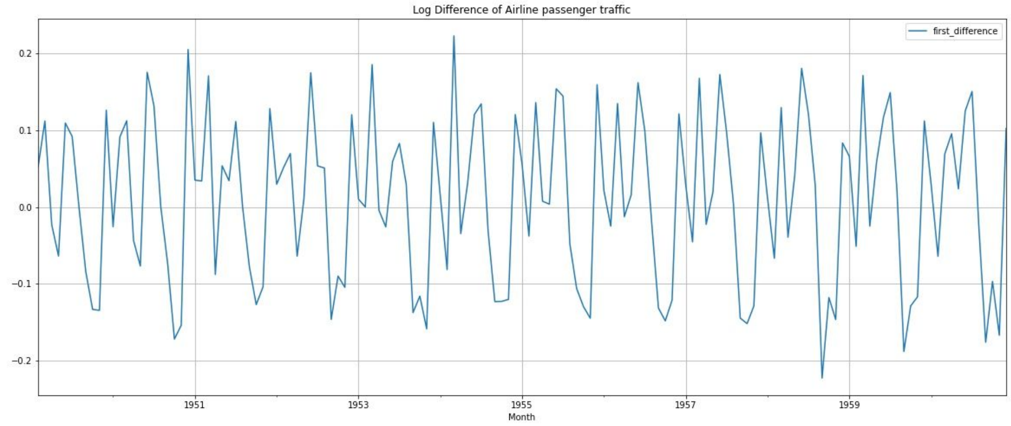
As it is evident from the ACF plots that the data is seasonal with a period of 12 as the peaks occur at lags of 12 months.



```
Results of Dickey-Fuller Test:
TestStatistic                    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
```

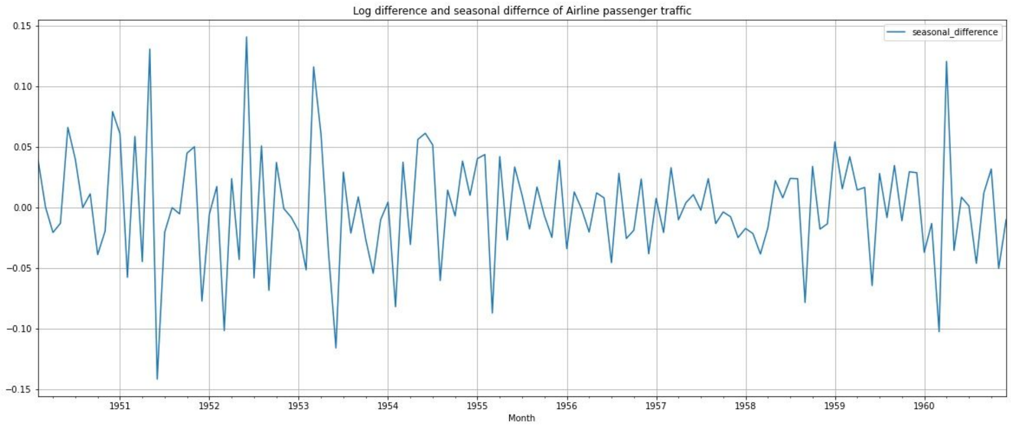
We can see that after applying the Dickey-Fuller Test on our data, the p value comes out to be significantly close to 1, as a result we accept the null hypothesis. Hence our data is not-stationary.

To make the data stationary first we apply log difference as a result in order to eliminate 1st order trend, present in the data.

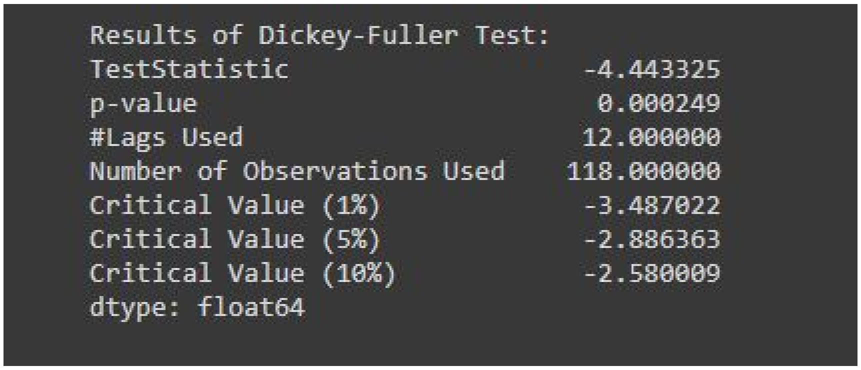


As it is evident from the differenced data that now there is no trend component present in it. We already know from the ACF plots that the data is seasonal with period of 12 months.

Next step in the process to make the time series stationary is by applying seasonal differencing.



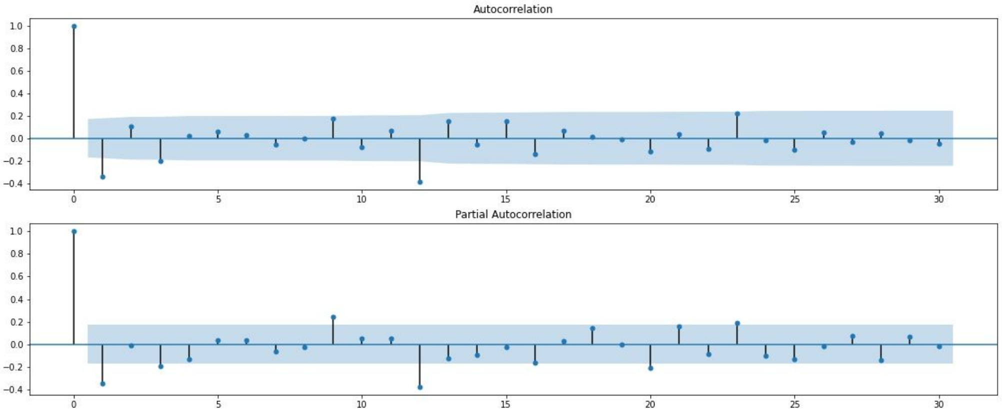
As it is evident from the time series, it pretty much looks like noise. And looks like we have achieved stationarity.
We confirm stationarity by applying the Dickey-Fuller test:



```
Results of Dickey-Fuller Test:
TestStatistic                    -4.443325
p-value                           0.000249
#Lags Used                       12.000000
Number of Observations Used     118.000000
Critical Value (1%)              -3.487022
Critical Value (5%)              -2.886363
Critical Value (10%)             -2.580009
dtype: float64
```

The p-value is small enough for us to reject the null hypothesis, and we can consider that the time series is stationary.

ACF and PACF of the stationary data:

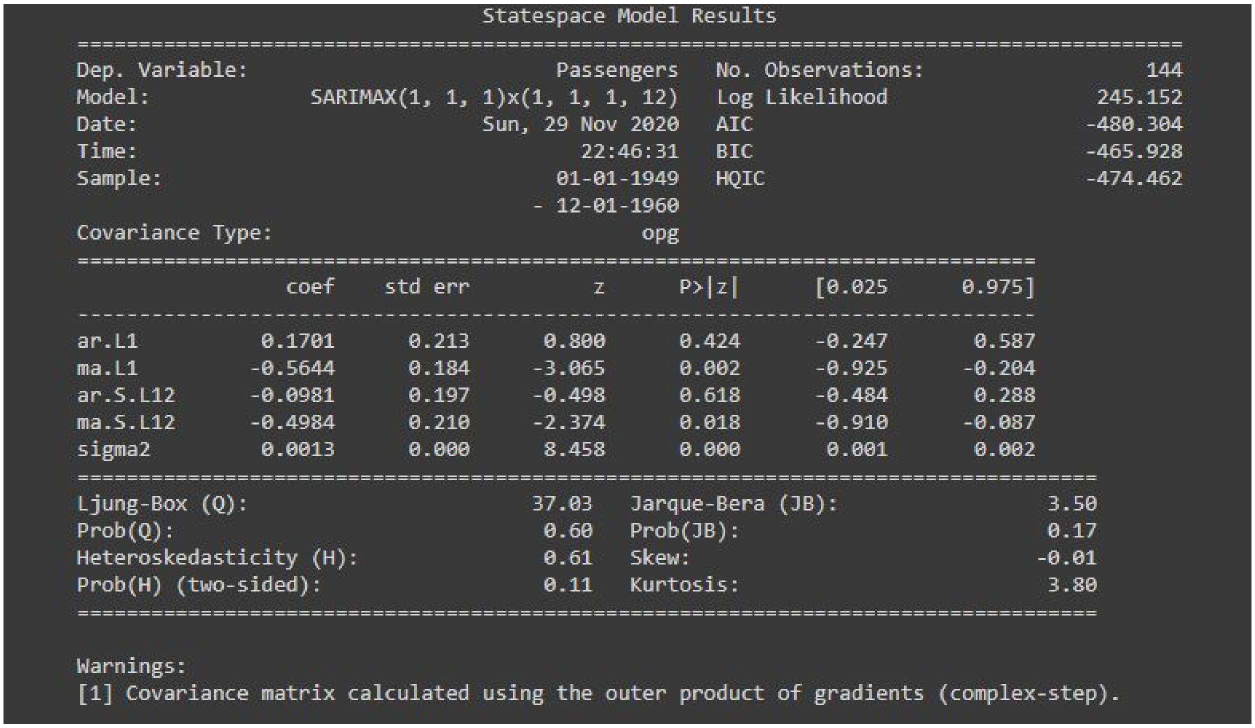


We can see from the PACF that we have a significant peak at lag 1, which suggests an AR(1) process. Also, we have another peak at lag 12, suggesting a seasonal autoregressive process of order 1 (P = 1).

Similarly looking at the ACF plot, we see a significant peak at lag 1, suggesting a MA(1) process and another peak at lag 12, suggesting a seasonal moving average process of order 1 (Q=1)

Based on the features portrayed by the plots, an initial SARIMA (1,1,1)×(1,1,1,12) model is proposed.

Model Building



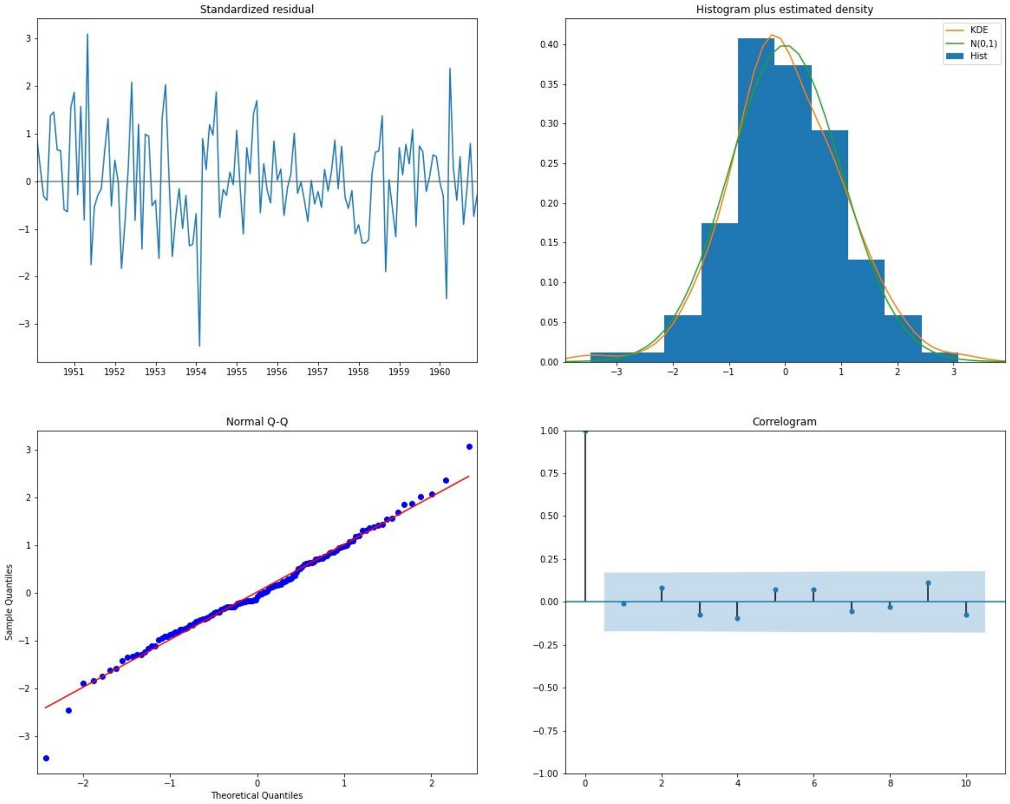
```
                                     Statespace Model Results
==========================================================================================
Dep. Variable:                         Passengers   No. Observations:                  144
Model:             SARIMAX(1, 1, 1)x(1, 1, 1, 12)   Log Likelihood                 245.152
Date:                            Sun, 29 Nov 2020   AIC                           -480.304
Time:                                    22:46:31   BIC                           -465.928
Sample:                                01-01-1949   HQIC                          -474.462
                                     - 12-01-1960
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.1701      0.213      0.800      0.424      -0.247       0.587
ma.L1         -0.5644      0.184     -3.065      0.002      -0.925      -0.204
ar.S.L12      -0.0981      0.197     -0.498      0.618      -0.484       0.288
ma.S.L12      -0.4984      0.210     -2.374      0.018      -0.910      -0.087
sigma2         0.0013      0.000      8.458      0.000       0.001       0.002
===================================================================================
Ljung-Box (Q):                       37.03   Jarque-Bera (JB):                 3.50
Prob(Q):                              0.60   Prob(JB):                         0.17
Heteroskedasticity (H):               0.61   Skew:                            -0.01
Prob(H) (two-sided):                  0.11   Kurtosis:                         3.80
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

We took a set of integers and tried for different values of p,q and i and found that for the above shown process, the AIC and BIC values came out to be small and hence it is the required model which came out to be same as we anticipated using ACF and PACF plots.

The above table also displays the values of parameters.

Residual Analysis



- Histogram of residuals are used to show whether variance is normally distributed or not, a symmetric well shaped histogram which is evenly distributed around zero indicates that the normality assumption about the residuals is likely to be true. If the histogram indicates that random error is not normally distributed then it suggests that the model underlying assumptions may have been violated.

- Normal Q-Q , or quantile-quantile plot is a graphical tool to help us assess if a set of data plausibly came from some theoretical distribution such as a Normal or exponential. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, we should see the points forming a line that's roughly straight. We can observe that for our Q-Q plots both the quantiles are overlapping.

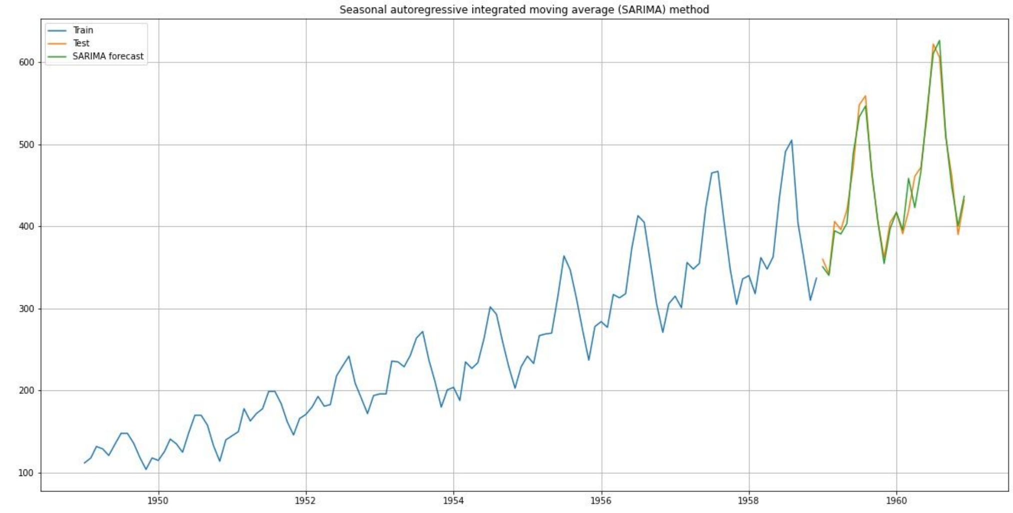
- In the standardized residual plot we can see that the residuals appear to be just fluctuating about x-axis without any kind of trend or pattern and hence showing the randomness of this residuals.

- A correlogram (also called Auto Correlation Function ACF Plot or Autocorrelation plot) is a visual way to show serial correlation in data that changes over time (i.e. time series data). Serial correlation (also called autocorrelation) is where an error at one point in time travels to a subsequent point in time. Correlograms can give you a good idea of whether or not pairs of data show autocorrelation. We can observe that our residual data follows a plot similar to WN sequence.

Hence based on these plots, we can conclude that there are no further trend that can be extracted from these residuals and that it follows a distribution similar to a normal WN process.

Model Predictions-

Now that we have a model built, we want to use it to make forecasts. First we use the model to forecast for time periods that we already have data for, so we can understand how accurate the forecasts are.



As can be seen from the plot above our model fits the time series very accurately and hence it is one of the optimum models for this prediction.

Code-

```
import pandas as pd
import numpy as np
import io
import matplotlib.pyplot as plt
import datetime
import statsmodels.api as sm
from statsmodels.tsa.stattools import acf
from statsmodels.tsa.stattools import pacf
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.statespace.sarimax import SARIMAX
from itertools import product
from tqdm.notebook import tqdm
from google.colab import files


data =
pd.read_csv(io.StringIO(uploaded['AirPassengers.csv'].decode('utf-8')))
data['Month'] = pd.to_datetime(data['Month'], format='%Y-%m')
data = data.set_index('Month')


data.plot(figsize=(20, 10))
plt.grid()
plt.legend(loc='best')
plt.title('Airline passenger traffic')


decomposition = seasonal_decompose(data.Passengers, freq=12,
extrapolate_trend = 1)
fig = plt.figure()
fig = decomposition.plot()
fig.set_size_inches(20, 12)


fig = plt.figure(figsize=(20,8))
ax1 = fig.add_subplot(211)
```

```
fig = sm.graphics.tsa.plot_acf(data.Passengers.iloc[:], lags=30, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(data.Passengers.iloc[:], lags=30, ax=ax2)




def dicky_fuller_test(timeseries):
    print('Results of Dickey-Fuller Test:')
    dftest = adfuller(timeseries)
    dfoutput = pd.Series(dftest[0:4],
index=['TestStatistic','p-value','#Lags Used','Number of Observations
Used'])
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print(dfoutput)


data['Passengers'] = np.log(data['Passengers'])
data['first_difference'] = data.Passengers - data.Passengers.shift()


data.first_difference.dropna(inplace=False).plot(figsize=(20, 8))
plt.grid()
plt.legend(loc='best')
plt.title('Log Difference of Airline passenger traffic')


data['seasonal_difference'] = data.first_difference -
data.first_difference.shift(12)


data.seasonal_difference.dropna(inplace=False).plot(figsize=(20, 8))
plt.grid()
plt.legend(loc='best')
plt.title('Log difference and seasonal differnce of Airline passenger
traffic')


dicky_fuller_test(data.seasonal_difference.dropna(inplace=False))
```

```
fig = plt.figure(figsize=(20,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(data['seasonal_difference'].iloc[13:],
lags=30, ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(data['seasonal_difference'].iloc[13:],
lags=30, ax=ax2)


model = SARIMAX(data.Passengers, order=(1, 1, 1), seasonal_order=(1, 1, 1,
12))
model_fit = model.fit()


y_hat_sarima = data['first_difference'].copy()
y_hat_sarima['sarima_forecast_tmp'] =
model_fit.predict(data['first_difference'].index.min(),
data['first_difference'].index.max())
y_hat_sarima['sarima_forecast'] =
np.exp(y_hat_sarima['sarima_forecast_tmp'])


plt.figure(figsize=(20,10))
plt.grid()
plt.plot(np.exp(data.Passengers[:120]), label='Train')
plt.plot(np.exp(data.Passengers[120:]), label='Test')
plt.plot(y_hat_sarima['sarima_forecast'][data.Passengers[120:].index.min()
:], label='SARIMA forecast')
plt.legend(loc='best')
plt.title('Seasonal autoregressive integrated moving average (SARIMA)
method')


best_model = SARIMAX(data['Passengers'], order=(1, 1, 1),
seasonal_order=(1, 1, 1, 12)).fit(dis=-1)
print(best_model.summary())
```

```
best_model.plot_diagnostics(figsize=(20, 16))
```