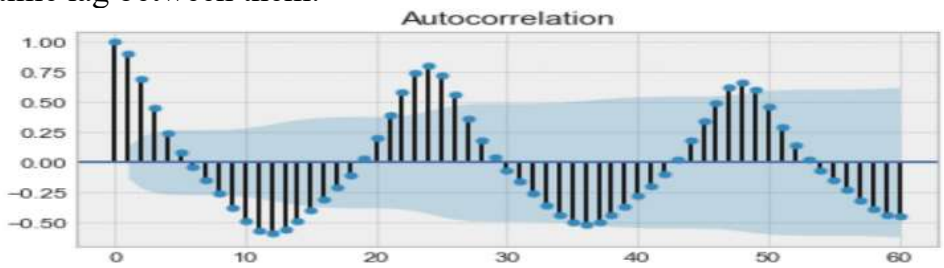
6. Check for the Autocorrelation Function and Partial Autocorrelation Function of the data.
    - (https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-min-temperatures.csv)

Aim: Understand autocorrelation, Partial Autocorrelation Function, ARMA, and apply these techniques in projects.

Autocorrelation Informally, is the similarity between observations as a function of the time lag between them.



Autocorrelation

AR, MA, ARMA, and ARIMA models are used to forecast the observation at (t+1) based on the historical data of previous time spots recorded for the same observation. However, it is necessary to make sure that the time series is stationary over the historical data of observation overtime period. If the time series is not stationary then we could apply the differencing factor on the records and see if the graph of the time series is a stationary overtime period.

ACF (Auto Correlation Function)
Auto Correlation function takes into consideration of all the past observations irrespective of its effect on the future or present time period. It calculates the correlation between the t and (t-k) time period. It includes all the ***lags*** or intervals between t and (t-k) time periods. Correlation is always calculated using the ***Pearson Correlation formula***.

PACF(Partial Correlation Function)
The PACF determines the partial correlation between time period t and t-k. It doesn't take into consideration all the time lags between t and t-k. For e.g. let's assume that today's stock price may be dependent on 3 days prior stock price but it might not take into consideration yesterday's stock price closure. Hence we consider only the time lags having a direct impact on future time period by neglecting the insignificant time lags in between the two-time slots t and t-k.

How to differentiate when to use ACF and PACF?
Eg.: Let's take an example of sweets sale and income generated in a village over a year. Under the assumption that ***every 2 months*** there is a festival in the village, we take out the historical data of sweets sale and income generated for 12 months. If we ***plot the time***

as month then we can observe that when it comes to calculating the sweets ***sale*** we are interested in only alternate months as the sale of sweets increases every two months. But if we are to consider the ***income*** generated next month then we have to take into consideration all the 12 months of last year.

So in the above situation, we will use ACF to find out the income generated in the future but we will be using PACF to find out the sweets sold in the next month.

AR (Auto-Regressive) Model

The time period at t is impacted by the observation at various slots t-1, t-2, t-3, ….., t-k. The ***impact of previous time spots is decided by the coefficient factor at that particular period of time***. The price of a share of any particular company X may depend on all the previous share prices in the time series. This kind of model calculates the regression of past time series and calculates the present or future values in the series in know as ***Auto Regression (AR) model***.

$$Yt = \beta_1 * y_{-1} + \beta_2 * y_{t-2} + \beta_3 * y_{t-3} + \ldots\ldots\ldots\ldots + \beta_k * y_{t-k}$$

Eg.:Consider an example of a milk distribution company that produces milk every month in the country. We want to calculate the amount of milk to be produced current month considering the milk generated in the last year. We begin by calculating the PACF values of all the 12 lags with respect to the current month. If the value of the ***PACF of any particular month is more than a significant value only those values will be considered for the model analysis.***

For e.g in the above figure the values 1,2, 3 up to 12 displays the direct effect(PACF) of the milk production in the current month w.r.t the given the lag t. If we consider two significant values above the threshold then the model will be termed as AR(2).

MA (Moving Average) Model

The time period at t is impacted by the ***unexpected external factors*** at various slots t-1, t-2, t-3, ….., t-k. These *unexpected impacts are* ***known as Errors or Residuals.*** The impact of previous time spots is decided by the coefficient factor α at that particular period of time. The price of a share of any particular company X may depend on some company merger that happened overnight or maybe the company resulted in shutdown due to bankruptcy. ***This kind of model calculates the residuals or errors of past time series and calculates the present or future values in the series*** in know as ***Moving Average (MA) model.***

$$Yt = \alpha_1 * \varepsilon_{t-1} + \alpha_2 * \varepsilon_{t-2} + \alpha_3 * \varepsilon_{t-3} + \ldots\ldots\ldots\ldots + \alpha_k * \varepsilon_{t-k}$$

Eg.:Consider an example of Cake distribution during birthday. Let's assume that your mom asks you to bring pastries to the party. Every year you miss judging the no of invites to the party and end upbringing more or less no of cakes as per requirement. The

difference in the actual and expected results in the error. So you want to avoid the error for this year hence we apply the moving average model on the time series and calculate the no of pastries needed this year based on past collective errors. Next, calculate the ACF values of all the lags in the time series. If the value of the ACF of any particular month is more than a significant value only those values will be considered for the model analysis.

For e.g in the above figure the values 1,2, 3 up to 12 displays the total error(ACF) of count in pastries current month w.r.t the given the lag t by considering all the in-between lags between time t and current month. If we consider two significant values above the threshold then the model will be termed as MA(2).

ARMA (Auto Regressive Moving Average) Model

This is a model that is combined from the AR and MA models. In this model, *the impact of previous lags along with the residuals is considered for forecasting the future values of the time series.* Here β represents the coefficients of the AR model and α represents the coefficients of the MA model.

$Yt = \beta_1 * y_{t-1} + \alpha_1 * \varepsilon_{t-1} + \beta_2 * y_{t-2} + \alpha_2 * \varepsilon_{t-2} + \beta_3 * y_{t-3} + \alpha_3 * \varepsilon_{t-3} + \ldots\ldots\ldots\ldots + \beta_k * y_{t-k} + \alpha_k * \varepsilon_{t-k}$

Eg.: Consider the above graphs where the MA and AR values are plotted with their respective significant values. Let's assume that we consider only 1 significant value from the AR model and likewise 1 significant value from the MA model. So the ARMA model will be obtained from the combined values of the other two models will be of the order of ARMA(1,1).

We know that in order to apply the various models we must in the beginning convert the series into Stationary Time Series. So, apply the differencing or Integrated method where the t-1 value is subtracting from t values of time series. After applying the first differencing if we are still unable to get the Stationary time series then, again apply the second-order differencing.

The ARIMA model is quite similar to the ARMA model other than the fact that it includes one more factor known as Integrated(I) i.e. differencing which stands for I in the ARIMA model. So in short ARIMA model ***is a combination of a number of differences already applied on the model in order to make it stationary, the number of previous lags along with residuals errors in order to forecast future values.***

Let's assume that we consider only 1 significant value from the AR model and likewise 1 significant value from the MA model. Also, the graph was initially non-stationary and we had to perform differencing operation once in order to convert into a stationary set. Hence the ARIMA model which will be obtained from the combined values of the other two models along with the Integral operator can be displayed as ARIMA(1,1,1).

Autocorrelation and partial autocorrelation plots are heavily used in time series analysis and forecasting.

The autocorrelation plots that graphically summarize the strength of a relationship with an observation in a time series with observations at prior time steps. The difference between autocorrelation and partial autocorrelation can be difficult and confusing unless carefully discern.

Aim –

- To plot and review the partial autocorrelation function for a time series.
- To find the difference between autocorrelation and partial autocorrelation functions for time series analysis.

Dataset – Minimum Daily Temperatures Dataset

- This dataset describes the minimum daily temperatures over 10 years (1981-1990) in the city Melbourne, Australia. The units are in degrees Celsius and there are 3,650 observations. [Source: Australian Bureau of Meteorology].

Step 1 : Download the dataset and place it in your current working directory with the filename "daily-minimum-temperatures.csv'".

```
from pandas import read_csv
from matplotlib import pyplot
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
series.plot()
pyplot.show()
```

Correlation and Autocorrelation

Statistical correlation summarizes the strength of the relationship between two variables. Assume the distribution of each variable fits a Gaussian (bell curve) distribution. If this is the case, use the Pearson's correlation coefficient to summarize the correlation between the variables. The Pearson's correlation coefficient is a number between -1 and 1 that describes a negative or positive correlation respectively. A value of zero indicates no correlation.

Calculate the correlation for time series observations with previous time steps, called lags. Because the correlation of the time series observations is *calculated with values of the same series at previous times*, this ***is called a serial correlation, or an autocorrelation***. A plot of the autocorrelation of a time series by lag is called the

AutoCorrelation Function, or the acronym ACF. This plot is sometimes called a ***correlogram or an autocorrelation*** plot.

Below is an example of calculating and plotting the autocorrelation plot for the Minimum Daily Temperatures using the plot_acf() function from the statsmodels library.

```
from pandas import read_csv
from matplotlib import pyplot
from statsmodels.graphics.tsaplots import plot_acf
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
plot_acf(series)
pyplot.show()
```

Running the example creates a 2D plot showing the lag value along the x-axis and the correlation on the y-axis between -1 and 1. Confidence intervals are drawn as a cone. By default, this is set to a 95% confidence interval, suggesting that correlation values outside of this code are very likely a correlation and not a statistical fluke. By default, all lag values are printed, which makes the plot noisy. We can limit the number of lags on the x-axis to some 'n' to make the plot easier to read.

Partial Autocorrelation Function

A partial autocorrelation is a summary of the relationship between an observation in a time series with observations at prior time steps with the relationships of intervening observations removed. The partial autocorrelation at lag k is the correlation that results after removing the effect of any correlations due to the terms at shorter lags.

The autocorrelation for an observation and an observation at a prior time step is comprised of both the **direct correlation and indirect correlation**s. *These indirect correlations are a linear function of the correlation of the observation, with observations at intervening time steps*. It is these indirect correlations that the partial autocorrelation function seeks to remove. This is the intuition for the ***partial autocorrelation***.

The example below calculates and plots a partial autocorrelation function for the first 50 lags in the Minimum Daily Temperatures dataset using the plot_pacf() from the statsmodels library.

```
from pandas import read_csv
from matplotlib import pyplot
from statsmodels.graphics.tsaplots import plot_pacf
series = read_csv('daily-minimum-temperatures.csv', header=0, index_col=0)
plot_pacf(series, lags=50)
pyplot.show()
```

- Running the example creates a 2D plot of the partial autocorrelation for the first 50 lags.

Autoregression Intuition
Consider a time series that was generated by an autoregression (AR) process with a lag of k. The ACF describes the autocorrelation between an observation and another observation at a prior time step that includes direct and indirect dependence information. This means, would expect the ACF for the AR(k) time series to be strong to a lag of k and the inertia of that relationship would carry on to subsequent lag values, trailing off at some point as the effect was weakened.

The PACF only describes the direct relationship between an observation and its lag. This would suggest that there would be no correlation for lag values beyond k. This is exactly the expectation of the ACF and PACF plots for an AR(k) process.

Moving Average Intuition
Consider a time series that was generated by a moving average (MA) process with a lag of k. The moving average process is an autoregression model of the time series of residual errors from prior predictions. Another way to think about the moving average model is that it corrects future forecasts based on errors made on recent forecasts. We would expect the ACF for the MA(k) process to show a strong correlation with recent values up to the lag of k, then a sharp decline to low or no correlation. By definition, this is how the process was generated. For the PACF, we would expect the plot to show a strong relationship to the lag and a trailing off of correlation from the lag onwards.

Durbin-Watson test
The test gives an output ranging from 0 to 4. The autocorrelation will be

Closer to 0: Stronger and positive
Middle: Low
Closer to 4: Negative

Ljung–Box test
It is also known as the modified Box-Pierce, Ljung–Box Q test, Box test, or Portmanteau test. It tests for the absence of serial autocorrelation for a given lag “k.” In addition, it tests for randomness and independence. If the autocorrelations of the residuals are very small, the model is fine; that is, the model does not exhibit a significant lack of fit.

Calculating Autocorrelation and Partial Autocorrelation

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.tsatools import lagmat

import matplotlib.pyplot as plt

# settings
plt.style.use("seaborn")
plt.rcParams["figure.figsize"] = (16, 8)
%config InlineBackend.figure_format = "retina"
```

Data

```
df = pd.read_csv("../data/air_passengers.csv", index_col=0)
df.index = pd.to_datetime(df.index)
y = df["#Passengers"]
y
```

```
fig, ax = plt.subplots(2, 1)
plot_acf(df, ax=ax[0])
plot_pacf(df, ax=ax[1], method="ols")
```

Autocorrelation Function (ACF)

```
acf(y, nlags=10)
```

Replicating it the wrong way

```
acf_df = pd.DataFrame()
for lag in range(0, 11):
    acf_df[f"y_lag_{lag}"] = y.shift(lag)

acf_df
```

```
acf_df.corr()["y_lag_0"].values
```

Replicating it the right way

```
# benchmark
acf(y, nlags=10)
# replicating the acf function
acf_list = []
mu = y.mean()
for lag in range(0, 11):
    acf_list.append(np.dot((y - mu).iloc[lag:], (y.shift(lag) - mu).iloc[lag:]) / sum((y - mu)
** 2))
np.array(acf_list)
# alternative way to write the same using sum instead of the dot product
acf_list = []
mu = y.mean()

for lag in range(0, 11):
    acf_list.append(sum((y - mu).iloc[lag:] * (y.shift(lag) - mu).iloc[lag:]) / sum((y - mu)
** 2))

np.array(acf_list)
```

Partial Autocorrelation Function (PACF)

```
pacf(df, nlags=10, method="ols")
N_LAGS = 10

# the first partial autocorrelation is always equal to 1
pacf_list = [1]

X = pd.DataFrame(lagmat(y, N_LAGS))
X.columns = [f"lag_{lag+1}" for lag in range(10)]

for k in range(1, N_LAGS + 1):
    fitted_model = LinearRegression().fit(X.iloc[k:, :k],
                                          y.iloc[k:])
    pacf_list.append(fitted_model.coef_[-1])

np.array(pacf_list)
```

Below you can see how the PACF values change when including more lags using the inefficient method

```
pacf(y, 2, method="ols-inefficient")
pacf(y, 4, method="ols-inefficient")
```

For comparison's sake, we run the same two calls to the pacf function using the efficient method.

```
pacf(y, 2, method="ols")
pacf(y, 4, method="ols")
```