Experiment No- 4

Aim :- Implement Time Series Analysis for rainfall in R Programming. Objective:- To understand the use of time series models for prediction.

Description:-

- Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. In time series analysis, analysts record data points at consistent intervals over a set period of time rather than just recording the data points randomly.

The basic syntax for ts() function in time series analysis is -

- timeseries.object.name <- ts(data, start, end, frequency)

Following is the description of the parameters used -

- data is a vector or matrix containing the values used in the time series.
- start specifies the start time for the first observation in time series.
- end specifies the end time for the last observation in time series.
- frequency specifies the number of observations per unit time.
- Except the parameter "data" all other parameters are optional.

Different Time Intervals:

The value of the frequency parameter in the ts() function decides the time intervals at which the data points are measured. A value of 12 indicates that the time series is for 12 months. Other values and its meaning is as below -

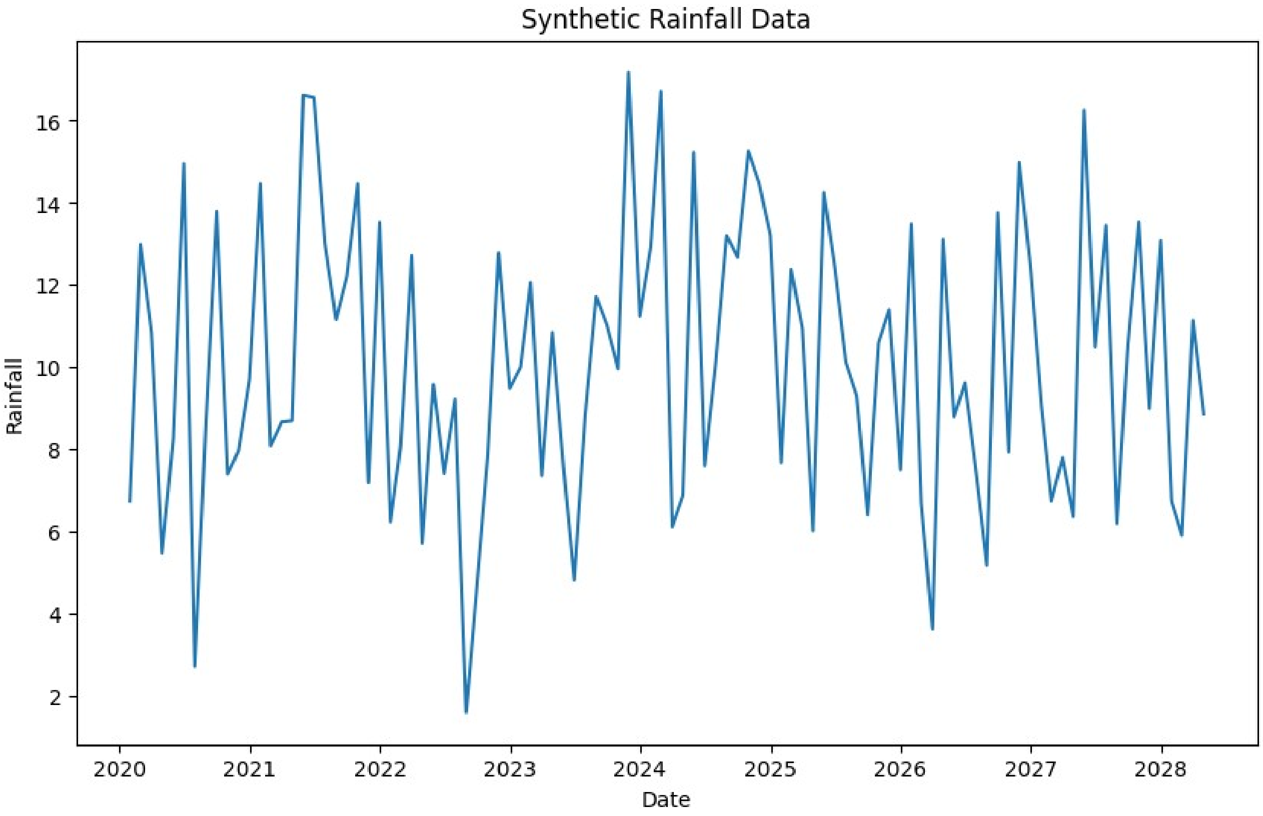
- frequency = 12 pegs the data points for every month of a year.
- frequency = 4 pegs the data points for every quarter of a year.
- frequency = 6 pegs the data points for every 10 minutes of an hour.
- frequency = 24*6 pegs the data points for every 10 minutes of a day.

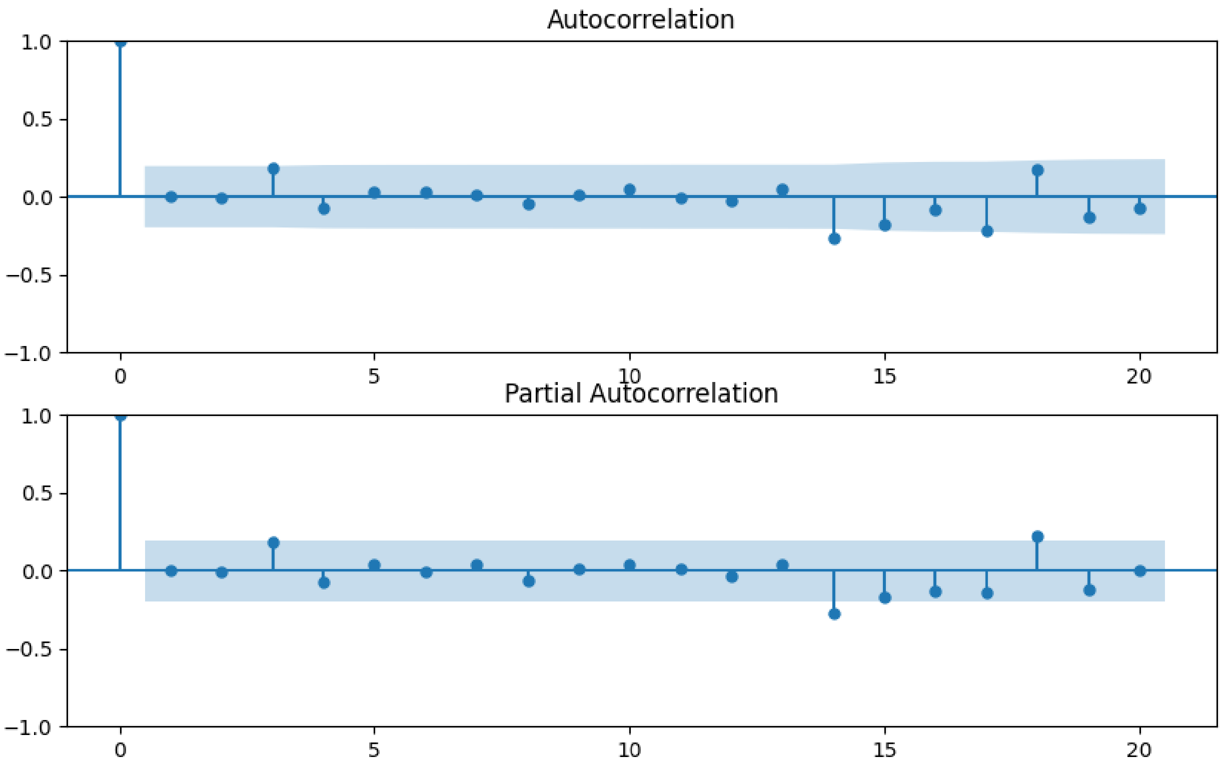
Program:

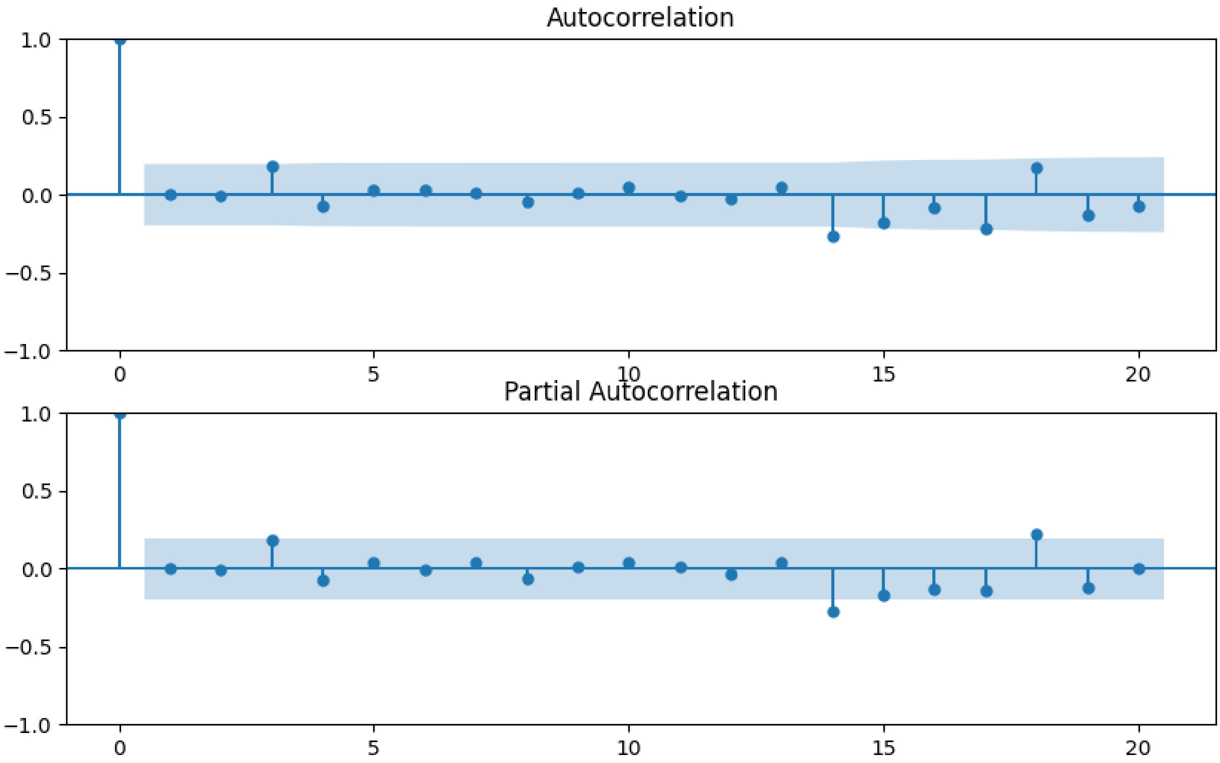
```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from statsmodels.tsa.arima.model import ARIMA

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Generate synthetic monthly rainfall data

np.random.seed(123)
```

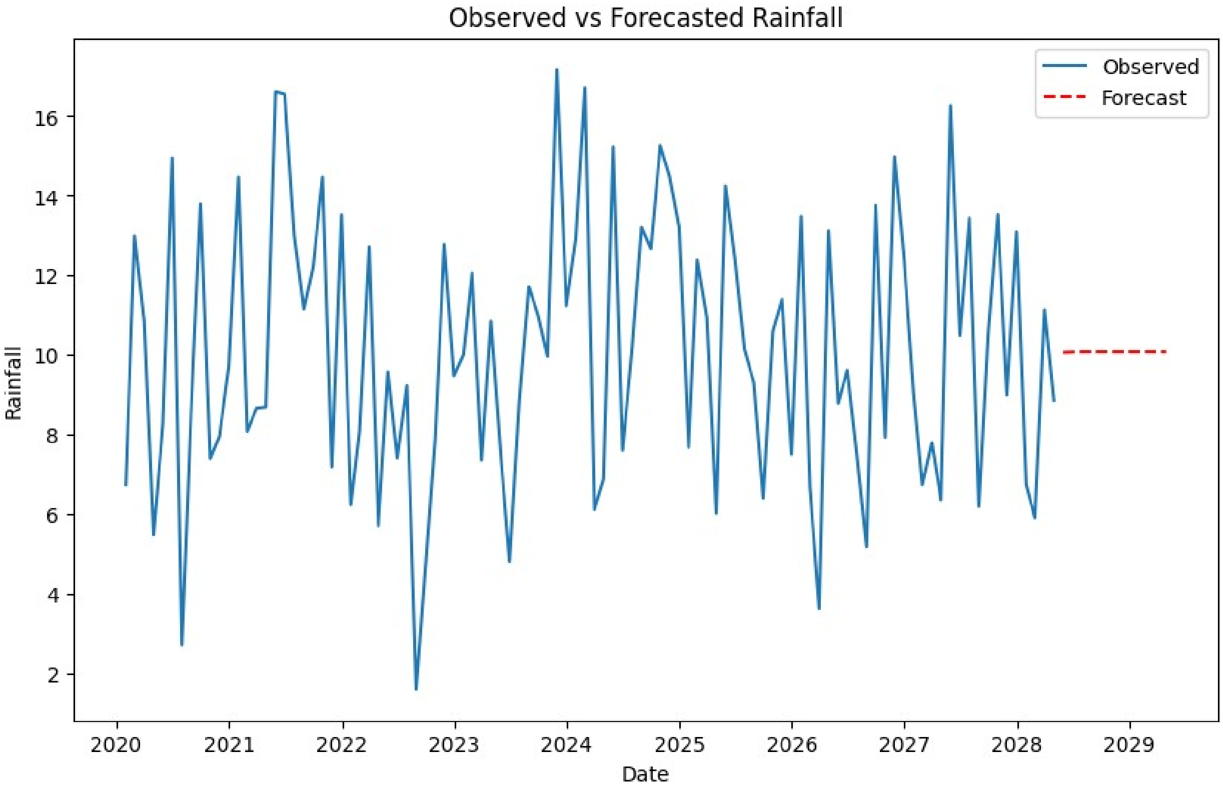
```
dates = pd.date_range(start='2020-01-01', periods=100, freq='M')
rainfall = np.random.normal(loc=10, scale=3, size=len(dates))
# Create a DataFrame
data = pd.DataFrame({'date': dates, 'rainfall': rainfall})
data.set_index('date', inplace=True)
# Visualize the data
plt.figure(figsize=(10, 6))
plt.plot(data)
plt.title('Synthetic Rainfall Data')
plt.xlabel('Date')
plt.ylabel('Rainfall')
plt.show()
# Plot ACF and PACF to determine the order of AR and MA components
fig, ax = plt.subplots(2, figsize=(10, 6))
plot_acf(data, ax=ax[0])
plot_pacf(data, ax=ax[1])
plt.show()
# Fit the ARIMA model
model = ARIMA(data, order=(2, 1, 1))
fit_model = model.fit()
# Summary of the model
print(fit_model.summary())
# Plot diagnostics of the model
fit_model.plot_diagnostics(figsize=(10, 8))
plt.show()
# Plot the original data and the forecasted values
plt.figure(figsize=(10, 6))
plt.plot(data, label='Observed')
plt.plot(forecast, color='red', linestyle='--', label='Forecast')

plt.title('Observed vs Forecasted Rainfall')

plt.xlabel('Date')

plt.ylabel('Rainfall')

plt.legend()

plt.show()

# Forecast future values

forecast = fit_model.forecast(steps=12)  # Forecasting 12 steps ahead

print("Forecasted values:", forecast)
```

Output:




Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
Partial Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20


Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
Partial Autocorrelation
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20



```
Forecasted values:
2028-05-31    10.064656
2028-06-30    10.074053
2028-07-31    10.080200
2028-08-31    10.080353
2028-09-30    10.080386
2028-10-31    10.080388
2028-11-30    10.080388
2028-12-31    10.080388
2029-01-31    10.080388
2029-02-28    10.080388
2029-03-31    10.080388
2029-04-30    10.080388
Freq: M, Name: predicted_mean, dtype: float64
```

Conclusion:

1. **An orderly set of data arranged in accordance with their time of occurrence is called** Time Series.
2. **The graph of time series is called** a time series plot.
3. **Use of Matrix()-**

   The use of Matrix() is not explicitly mentioned in the provided experiment. However, if you're referring to the use of matrices in time series analysis, they are often employed in various computations and transformations involved in data preparation and modeling, such as in calculating autocorrelation functions or in matrix representations of linear models.