Experiment No 5

Aim- Implementation of ARIMA model in R Programming.

Objective- To Understand use of Auto-Regression Integrated Moving Average Time Series Model

Theory-

In R programming , data analysis and visualization is so easy to learn the behavior of the data. Moreover, the R language is used mostly in the data science field after Python. Time series analysis is a type of analysis of data used to check the behavior of data over a period of time. The data is collected over time sequentially by the ts() function along with some parameters. It helps in analyzing the pattern of the data over a graph. There are many techniques used to forecast the time series object over the plot graph but the ARIMA model is the most widely used approach out of them.

Time Series Forecasting

Time series forecasting is a process of predicting future values with the help of some statistical tools and methods used on a data set with historical data. Some of the applications of time series forecasting are:

- Predicting stock prices
- Forecast weather
- Forecast the sales of a product

ARIMA model

ARIMA stands for AutoRegressive Integrated Moving Average and is specified by three order parameters: (p, d, q).

- AR(p) Autoregression: A regression model that utilizes the dependent relationship between a current observation and observations over a previous period.An auto regressive (AR(p)) component refers to the use of past values in the regression equation for the time series.

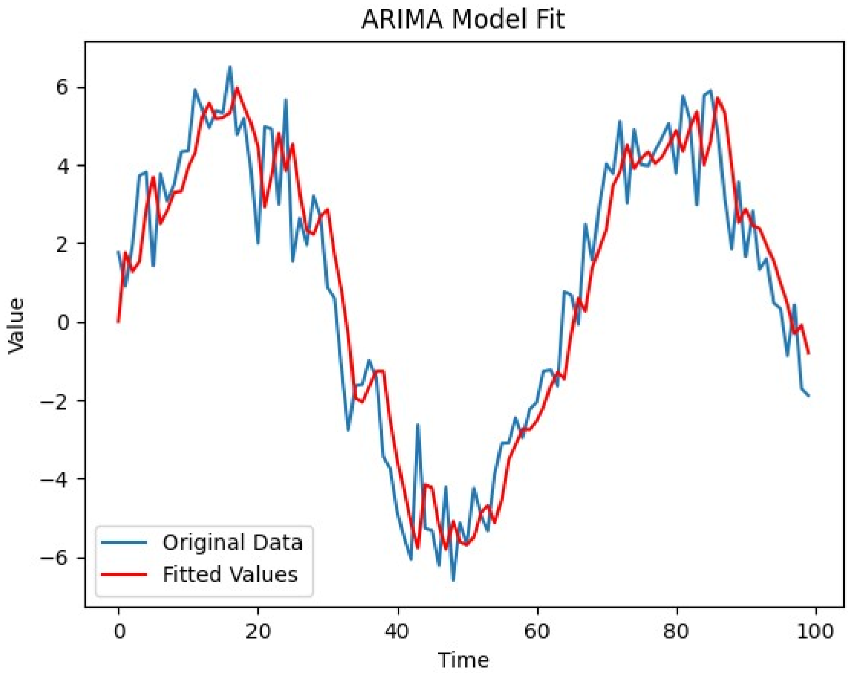
- I(d) Integration: Uses differencing of observations (subtracting an observation from observation at the previous time step) in order to make the time series stationary.

Program:

```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
# Generate synthetic time series data
np.random.seed(0)
n = 100
t = np.arange(n)
data = 5 * np.sin(0.1 * t) + np.random.normal(0, 1, n)
# Fit the ARIMA model
p = 1  # AR order
d = 1  # Differencing order
q = 1  # MA order
model = ARIMA(data, order=(p, d, q))
model_fit = model.fit()
# Print a summary of the model
print(model_fit.summary())
# Plot the original data and the fitted values
plt.plot(t, data, label='Original Data')
plt.plot(t, model_fit.fittedvalues, color='red', label='Fitted Values')
plt.xlabel('Time')
plt.ylabel('Value')
plt.title('ARIMA Model Fit')
plt.legend()
plt.show()
# Generate forecasts
n_forecast = 10
forecast = model_fit.forecast(steps=n_forecast)
# Print the forecasted values
print("Forecasted values:", forecast)
```

Output:

```
                              SARIMAX Results
==============================================================================
Dep. Variable:                  y   No. Observations:             100
Model:             ARIMA(1, 1, 1)   Log Likelihood             -164.149
Date:            Tue, 02 Apr 2024   AIC                         334.298
Time:                    18:07:00   BIC                         342.083
Sample:                         0   HQIC                        337.448
                            - 100
Covariance Type:              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.3444      0.230     -1.496      0.135      -0.796       0.107
ma.L1         -0.1109      0.236     -0.469      0.639      -0.574       0.353
sigma2         1.6098      0.275      5.864      0.000       1.072       2.148
==============================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):               1.55
Prob(Q):                              0.93   Prob(JB):                       0.46
Heteroskedasticity (H):               0.75   Skew:                          -0.09
Prob(H) (two-sided):                  0.42   Kurtosis:                       2.42
==============================================================================
```



Forecasted values: [-1.70238223 -1.76551764 -1.74377237 -1.75126194 -1.74868236 -1.74957083

-1.74926482 -1.74937021 -1.74933391 -1.74934642]

Conclusion:-

1. **The difference between the actual value of the time series and the forecasted value called** as the **forecast error** or **residual**.

2. **Use of ARIMA ()-**
The function **ARIMA()** is utilized in Python's statsmodels library to fit an ARIMA model to the time series data. It takes the order parameters (p, d, q) as arguments, representing the autoregressive order, differencing order, and moving average order, respectively.

3. **Use of forecast()-**
The function **forecast()** is used to generate forecasts for future time points based on the fitted ARIMA model. It takes the number of steps to forecast as an argument and returns the forecasted values for those steps ahead of the last observed data point.