Time Series Analysis: 780 (Tampere University)

Name: Anowar Hussain

ID: 151934177

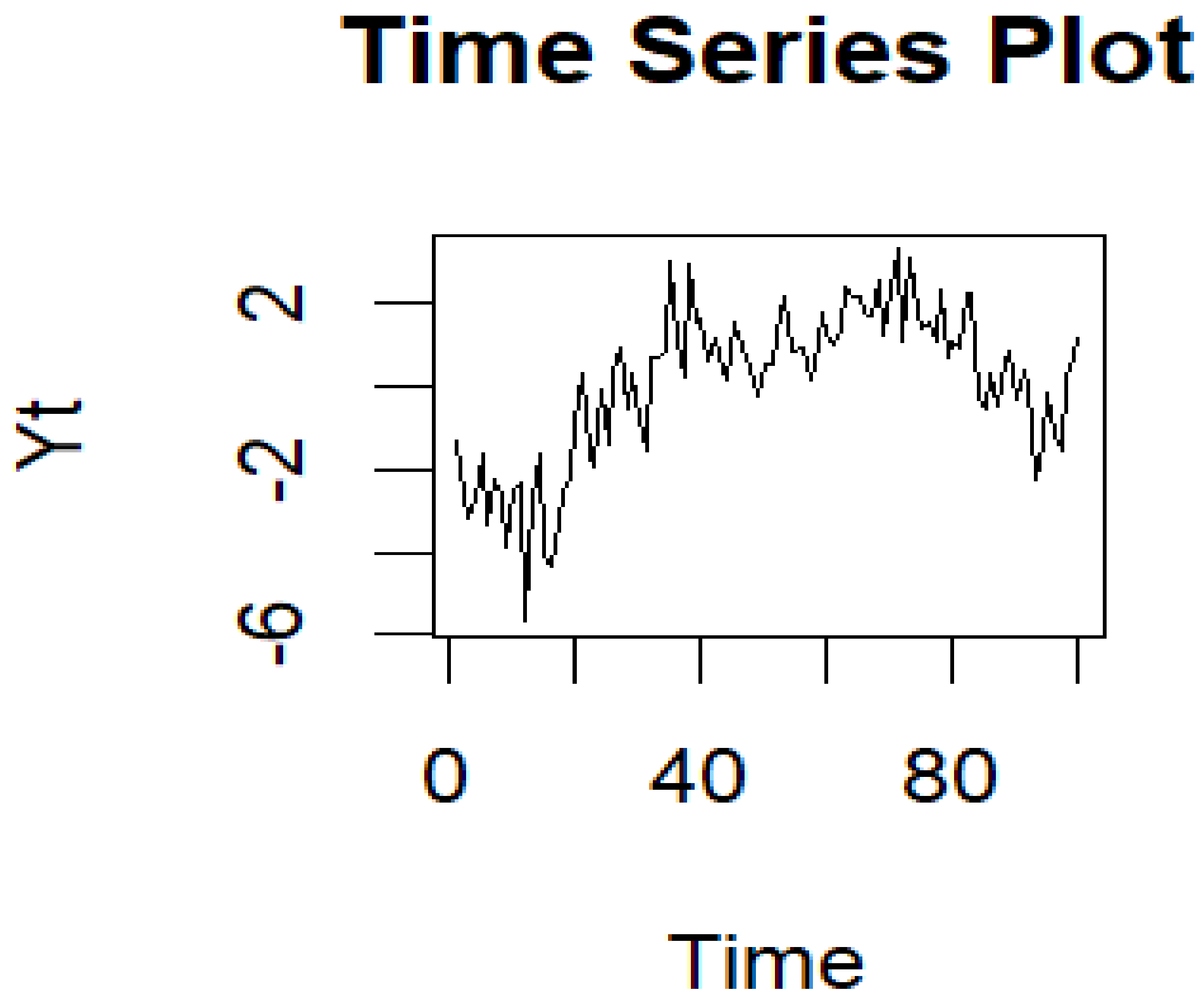
Column Assigned: V47

Assignment 1

Step 1: Preliminary analysis of Orders

1. Plot the time series (Yt):

```
plot(series, type = "l",ylab ="Yt", xlab ="Time", main = "Time Series Plot")
```



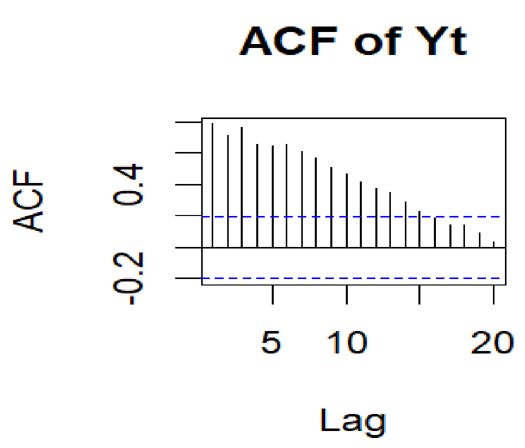
# This time series appears to have variations in mean over time, which suggests it might not be stationary. However, there is no clear trend upwards or downwards, and no obvious seasonality (regular fluctuations that follow a consistent pattern) is visible. There seems to be periods of increased and decreased variability in the values of Yt. The mean

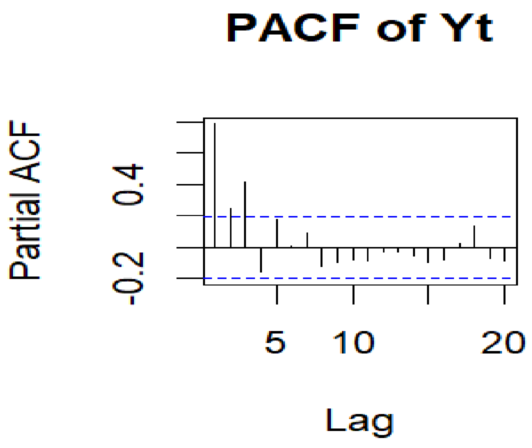
of the series looks to hover around zero. There may be some spikes that could be potential outliers or sudden jumps in the data.

2. Analysis of d:

   (a) Display the autocorrelation functions of Yt and ∇Yt:

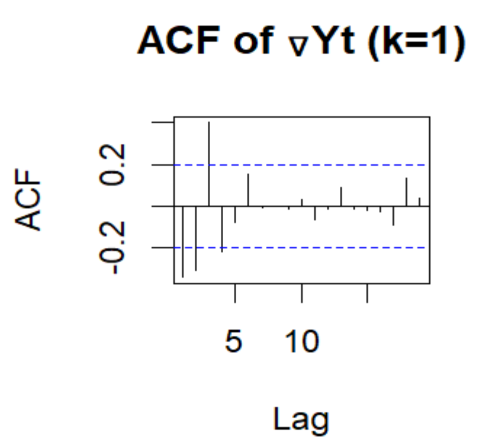
```
#display the autocorrelation function of Yt
par(mfrow=c(1,1))
acf(series, main = "ACF of Yt")
pacf(series, main = "PACF of Yt")
```

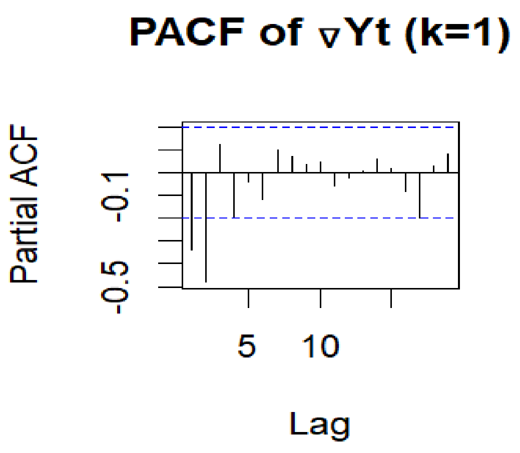




# The ACF plot shows significant autocorrelation at many lags, which slowly decays as the lags increase which suggest a non-seasonal ARIMA model and suggests that might need differencing.
# The PACF plot cuts off after the first lag, with the first lag being significant and the rest within the confidence interval (not significantly different from zero) which suggests an AR(1) model may be appropriate.

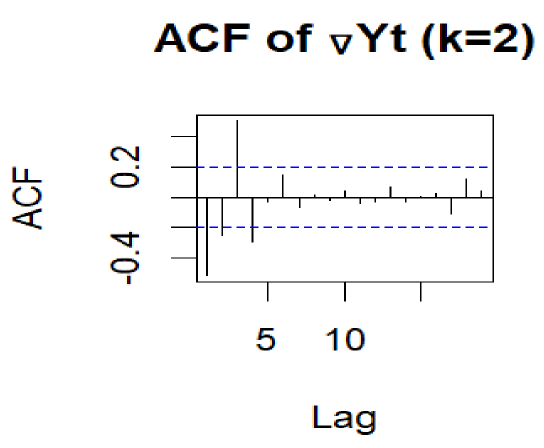
```
# Difference the series
diff_series_1 <- diff(series)
mean(diff_series_1)
[1] 0.02471119

diff_series_2 <- diff(diff_series_1)
mean(diff_series_2)
[1] 0.01699372

# Plot ACFs and PACFs after difference the series
par(mfrow=c(1,1))
acf(diff_series_1, main = "ACF of ∇Yt (k=1)")
pacf(diff_series_1, main = "PACF of ∇Yt (k=1)")
```

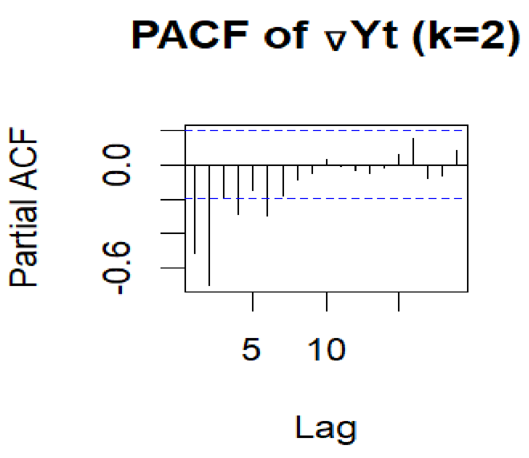




#The ACF of ∇Yt (k=1) plot shows that autocorrelations are not significant for most lags, which are within the confidence bounds, suggesting that the first differencing might be sufficient to achieve stationarity. Also, the mean of the first differenced series is close to zero, which is typical for a stationary series.
#The PACF of ∇Yt (k=1) plot shows that all partial autocorrelations are within the confidence interval after the first lag indicates that there may not be a need for an AR term once the series has been first differenced.

```
acf(diff_series_2, main = "ACF of ∇Yt (k=2)")
pacf(diff_series_2, main = "PACF of ∇Yt (k=2)")
```





#ACF and PACF plots for the second differenced series indicate that the series may be over-differenced, as typically indicated by the negative autocorrelations in the ACF plot. Which suggests that an ARIMA model with d = 1 might be appropriate.

(b) Augmented Dickey-Fuller test (ADF):

```
adf.test(series)
```

```
Augmented Dickey-Fuller Test

data:  series
Dickey-Fuller = -1.5214, Lag order = 4, p-value = 0.7749
alternative hypothesis: stationary
```

# In ADF Test on Original Series, p-value is greater than 0.05, we fail to reject the null h ypothesis of the presence of a unit root. This indicates that the original series is not stati onary.

```
adf.test(diff_series_1)
```

```
Augmented Dickey-Fuller Test

data:  diff_series_1
Dickey-Fuller = -5.8007, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

# In ADF Test on First Differenced Series, the p-value is less than 0.05, leading us to rej ect the null hypothesis. This suggests that the first differenced series is stationary.

```
adf.test(diff_series_2)
```

```
Augmented Dickey-Fuller Test

data:  diff_series_2
Dickey-Fuller = -7.9292, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

# In ADF Test on Second Differenced Series, the p-value indicates that the second differ enced series is also stationary.

(c) Order of d:

#Based on the results of ACF, PACF and ADF, it appears that the first difference is sufficient to achieve stationarity, that means the order of ARIMA model is d =1.

3. Analysis of (p, q):

```
# ACF and PACF plots after first difference the series
acf_diff <- acf(diff_series_1, main = "ACF of differenced series", lag.max = 20)
pacf_diff <- pacf(diff_series_1, main = "PACF of differenced series", lag.max = 20)

# Determine upper bounds for p and q
pmax <- which.max(acf_diff$acf[-1] ^ 2)
qmax <- which.max(pacf_diff$acf[-1] ^ 2)
```

```
# Display upper bounds
cat("Upper bound for p (AR):", pmax, "\n")

Upper bound for p (AR): 2

cat("Upper bound for q (MA):", qmax, "\n")

Upper bound for q (MA): 1

# Fit ARIMA model with determined p and q
arima_model <- auto.arima(series, d = 1, max.p = pmax, max.q = qmax, stepwise = TRUE, app
summary(arima_model)
Series: series
ARIMA(2,1,1)

Coefficients:
          ar1      ar2     ma1
      -0.8248  -0.5751  0.4348
s.e.   0.1654   0.0823  0.2159

sigma^2 = 0.969:  log likelihood = -137.77
AIC=283.54   AICc=283.97   BIC=293.92

Training set error measures:
                     ME      RMSE       MAE       MPE    MAPE      MASE        ACF1
Training set 0.04190415 0.9644677 0.7973603 -15.87299 129.693 0.8331675 -0.02577501
```

Step 2: Estimation and Selection of ARIMA Models

1. Compute the AIC and BIC for the 16 remaining models:

```
# determine the value of d, pmax, qmax
d <- 1
pmax <- 4
qmax <- 4

# Initialize an empty data frame to store the results
results <- data.frame(p = integer(), q = integer(), AIC = numeric(), BIC = nu
meric())

# Loop over all combinations of p and q within the specified bounds
for (p in 1:pmax) {
  for (q in 1:qmax) {

    # Fit the ARIMA model to the data
    model <- Arima(series, order = c(p, d, q), include.constant = FALSE)

    # Store the results
    results <- rbind(results, data.frame(p = p, q = q, AIC = AIC(model), BIC
= BIC(model)))
        }
      }


# Display the results
results
      p q      AIC      BIC
      1  1 1 294.4824 302.2677
      2  1 2 285.5680 295.9484
```

```
3   1 3 286.3161 299.2917
4   1 4 287.1848 302.7556
5   2 1 283.5395 293.9200
6   2 2 283.7109 296.6865
7   2 3 285.0672 300.6380
8   2 4 286.5101 304.6760
9   3 1 284.1407 297.1163
10  3 2 285.0051 300.5758
11  3 3 286.9318 305.0976
12  3 4 288.4852 309.2461
13  4 1 284.8465 300.4172
14  4 2 287.6256 305.7914
15  4 3 288.7031 309.4641
16  4 4 283.1019 306.4580
```

2. The estimate of best three specifications:

```
Sort the results by AIC and BIC
sorted_results <- results[order(results$AIC, results$BIC),]
sorted_results

   p q      AIC      BIC
16 4 4 283.1019 306.4580
5  2 1 283.5395 293.9200
6  2 2 283.7109 296.6865
9  3 1 284.1407 297.1163
13 4 1 284.8465 300.4172
10 3 2 285.0051 300.5758
7  2 3 285.0672 300.6380
2  1 2 285.5680 295.9484
3  1 3 286.3161 299.2917
8  2 4 286.5101 304.6760
11 3 3 286.9318 305.0976
4  1 4 287.1848 302.7556
14 4 2 287.6256 305.7914
12 3 4 288.4852 309.2461
15 4 3 288.7031 309.4641
1  1 1 294.4824 302.2677
```

```
# Print the best three models based on AIC
best_by_aic <- head(results[order(results$AIC), ], 3)
print(best_by_aic)

   p q      AIC      BIC
16 4 4 283.1019 306.4580
5  2 1 283.5395 293.9200
6  2 2 283.7109 296.6865
```

```
# Print the best three models based on BIC
best_by_bic <- head(results[order(results$BIC), ], 3)
print(best_by_bic)

  p q      AIC      BIC
5 2 1 283.5395 293.9200
2 1 2 285.5680 295.9484
6 2 2 283.7109 296.6865
```

# According to lowest value of AIC, the best three models are:

ARIMA(4,1,4) with an AIC of 283.1019 and a BIC of 306.4580

ARIMA(2,1,1) with an AIC of 283.5395 and a BIC of 293.9200
ARIMA(2,1,2) with an AIC of 283.7109 and a BIC of 296.6865

According to lowest value of BIC, the best three models are:

ARIMA(2,1,1) with an AIC of 283.5395 and a BIC of 293.9200
ARIMA(1,1,2) with an AIC of 285.5680 and a BIC of 295.9484
ARIMA(2,1,2) with an AIC of 283.7109 and a BIC of 296.6865

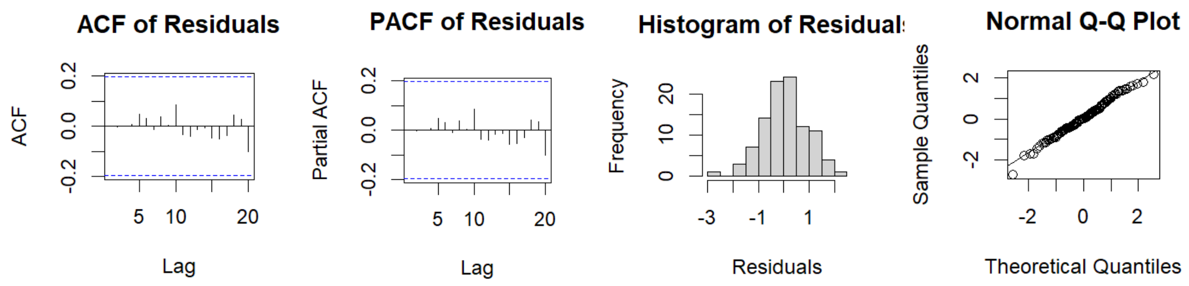
In both cases, the ARIMA(2,1,1) model is considered one of the best models. It is the best model according to BIC and the second-best according to AIC. The ARIMA(4,1,4) model has the lowest AIC, suggesting it might have the best fit among the models considered, but it is also the most complex model with the highest number of parameters. The ARIMA(1,1,2) model appears in BIC top-three lists, which indicates it might also be a good compromise between model complexity and fit for the number of low parameters.

Step 3: Diagnostic tests

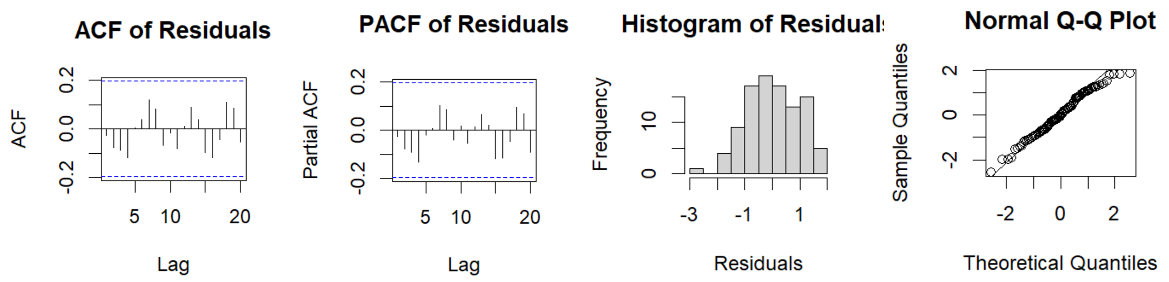
1. & 2. To test the absence of correlation of residuals terms, here I used LjungBox test, ACF and PACF plots, to test the normality I used histogram, QQplots, Shapiri-Wilk's test for the selected three models based on AIC and BIC.

```
# Define the diagnostic checking function
run_diagnostics <- function(model) {

+   #LjungBox test for residuals
+   print(Box.test(model$residuals, lag=10, type="Ljung-Box"))
+

+   #ACF and PACF plots for residuals
+   acf(model$residuals, main="ACF of Residuals")
+   pacf(model$residuals, main="PACF of Residuals")
+

+   #histogram test for residuals
+   hist(model$residuals, main="Histogram of Residuals", xlab="Residuals")
+

+   #Q-Q plot for residuals
+   qqnorm(model$residuals)
+   qqline(model$residuals)
+

+   #Shapiro normality test for residuals
+   print(shapiro.test(model$residuals))
+ }

# Run diagnostics for the best model based on AIC
best_model_aic <- Arima(series, order=c(4,1,4), include.constant=FALSE)
run_diagnostics(best_model_aic)
```

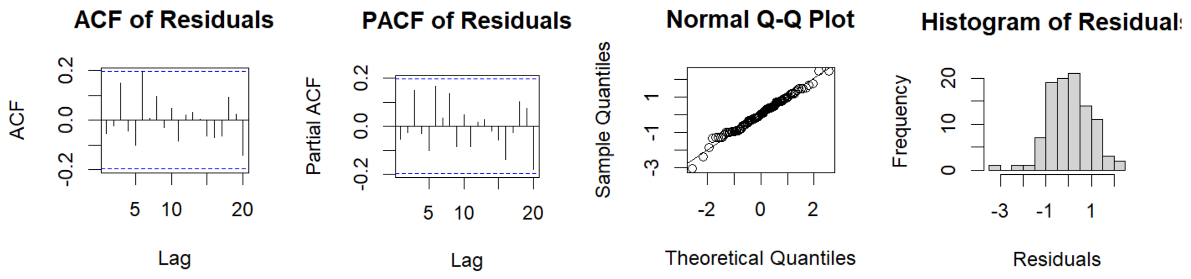
```
        Box-Ljung test

data:  model$residuals
X-squared = 1.3677, df = 10, p-value = 0.9993


        Shapiro-Wilk normality test

data:  model$residuals
W = 0.99317, p-value = 0.8981
```



```
# Run diagnostics for the best two models based on BIC
best_model_bic1 <- Arima(series, order=c(2,1,1), include.constant=FALSE)
run_diagnostics(best_model_bic1)

        Box-Ljung test

data:  model$residuals
X-squared = 6.0325, df = 10, p-value = 0.8125


        Shapiro-Wilk normality test

data:  model$residuals
W = 0.98504, p-value = 0.3198
```



```
best_model_bic2 <- Arima(series, order=c(1,1,2), include.constant=FALSE)
run_diagnostics(best_model_bic2)

        Box-Ljung test

data:  model$residuals
X-squared = 9.4925, df = 10, p-value = 0.4861


        Shapiro-Wilk normality test

data:  model$residuals
W = 0.98796, p-value = 0.5052
```



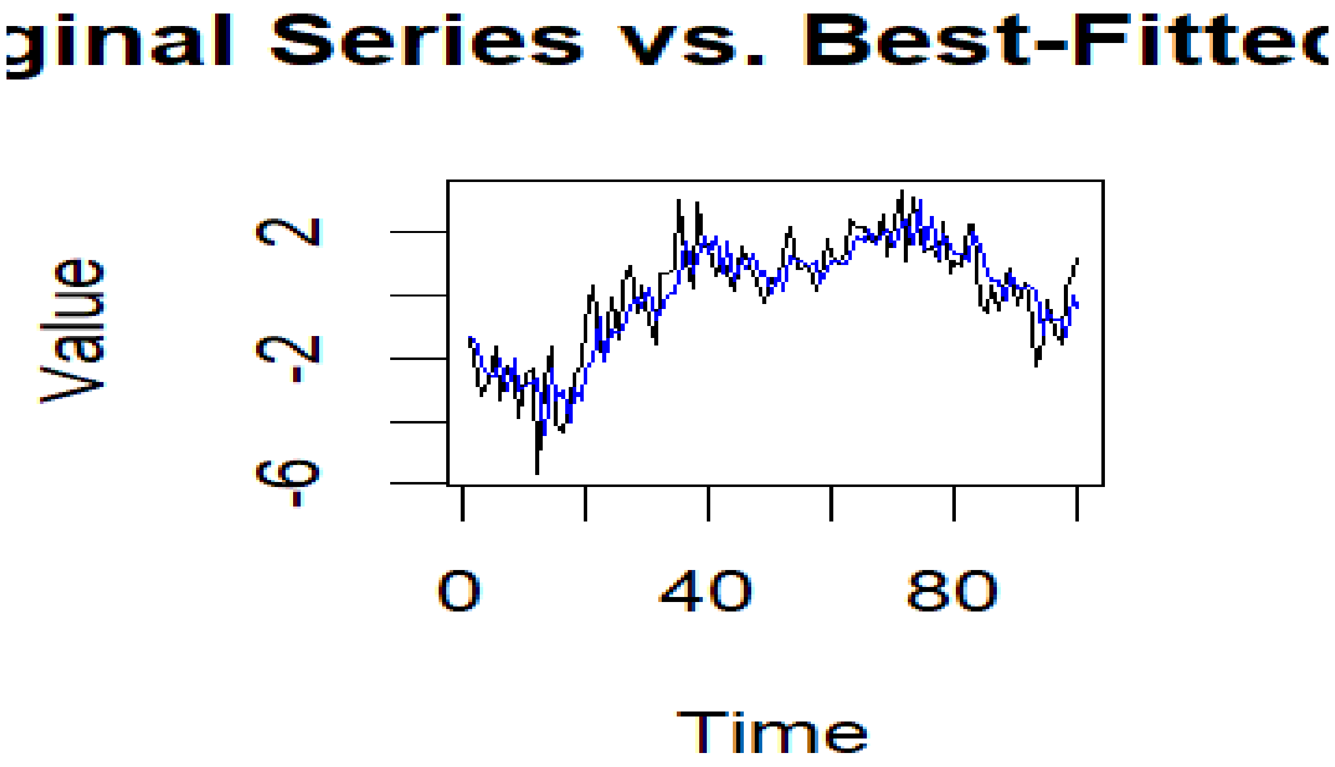
#Ljung-Box Test: The p-values for the Ljung-Box test for all models were high (well above 0.05), indicating that there is no significant autocorrelation in the residuals for any of the models. This is good as it suggests that all the models are capturing the underlying process well.
#Shapiro-Wilk Test: The p-values were also high for the Shapiro-Wilk test, indicating that the residuals from all models could be considered normally distributed.
#ACF and PACF of Residuals: The plots for the ACF and PACF of the residuals for each model showed that all autocorrelation coefficients were within the confidence bounds, indicating that the residuals are white noise.
#Histogram and QQ Plot: The histogram and QQ plots of the residuals for each model showed a roughly normal distribution.

3.Selecting the preferred model:

Considering all these diagnostic tests, each model appears to perform well. However, when applying the principle of parsimony, which favors simpler models (models with fewer parameters), the ARIMA(1,1,2) model stands out as the best model. It has fewer parameters than the ARIMA(4,1,4), potentially leading to a simpler and more generalizable model without overfitting. The ARIMA(1,1,2) model also had a competitive BIC value compared to the other models. Since BIC penalizes model complexity more heavily than AIC, a lower BIC can be a strong indicator of a better model fit when comparing models with a different number of parameters.
That's I considered that the best model would be ARIMA(1,1,2).

4.Plot the time series and best fitted model:

```
plot(series, main="Original Series vs. Best-Fitted Model",xlab = "Time", ylab
= "Value")
lines(fitted(best_model_bic2), col="blue")
```

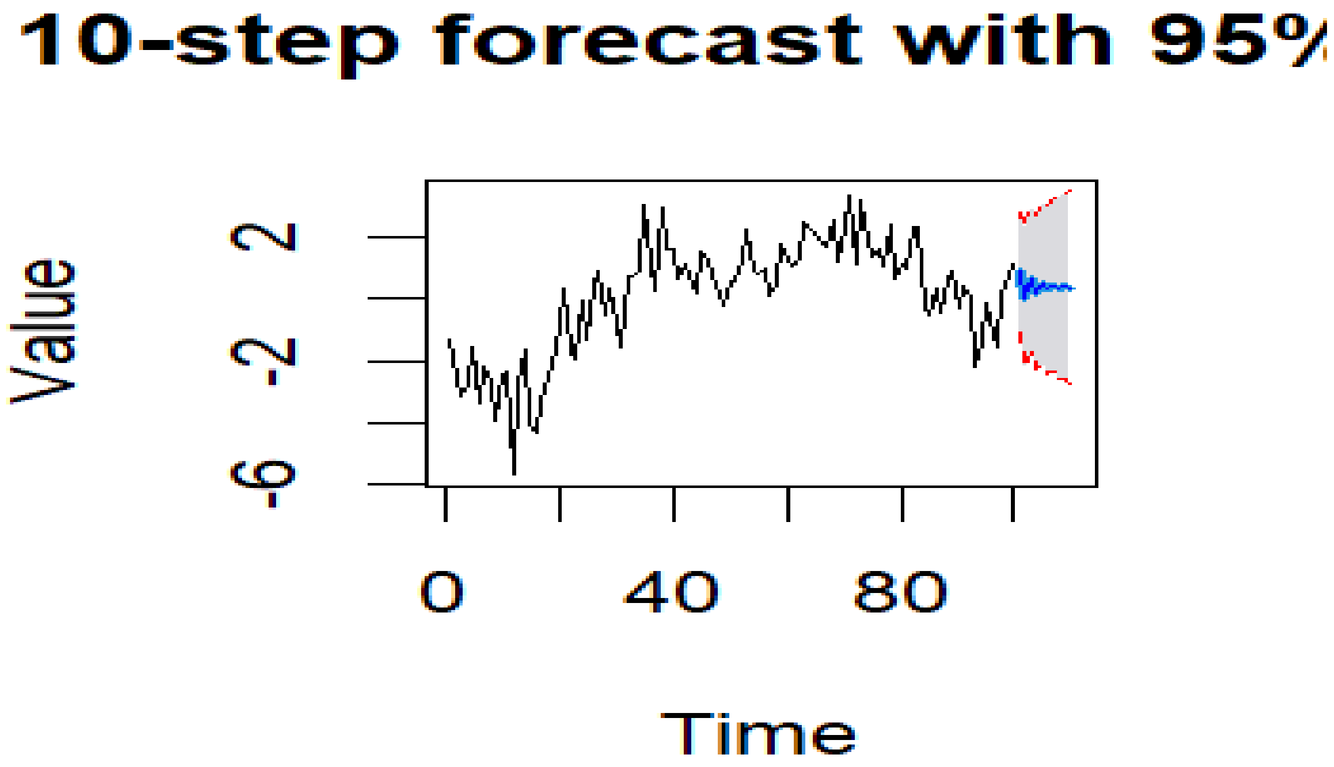


#This plot shows that it is a well-fitted ARIMA model closely tracking the historical time series data, indicating a good fit with minimal residuals. There are no visible trends or seasonal patterns in the data.

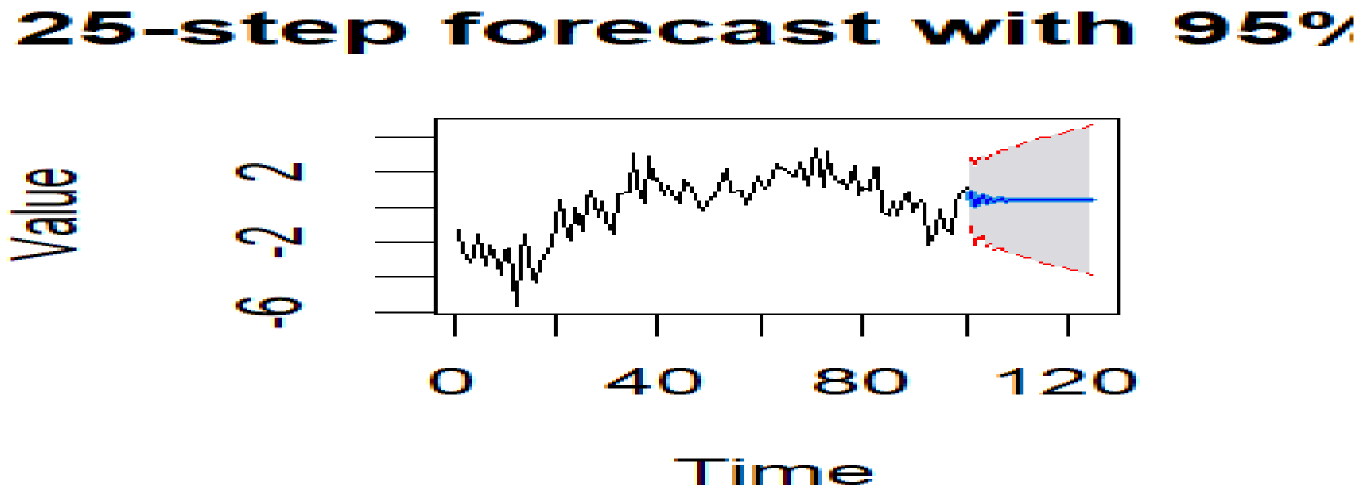
Step 4: Forecast

1. **Generate a forecast plot for a time series, incorporating a 95% CI:**

```
#Fit the preferred ARIMA model
preferred_model <- Arima(series, order =c(1,1,2), include.constant = FALSE)

#Generate forecast
forecast_10 <- forecast(preferred_model, h = 10, level = 95)
forecast_25 <- forecast(preferred_model, h = 25, level = 95)

#Plot the original series with forecasts and confidence intervals for h = 10
plot(forecast_10, main = "10-step forecast with 95% CI", xlab = "Time", ylab = "Value")
lines(forecast_10$mean, col = "blue")
lines(forecast_10$lower, col = "red", lty = 2)
lines(forecast_10$upper, col = "red", lty = 2)
```



```
#Plot the original series with forecasts and confidence intervals for h = 10
plot(forecast_25, main = "25-step ahead forecast with 95% CI", xlab = "Time", ylab = "Val
lines(forecast_25$mean, col = "blue")
lines(forecast_25$lower, col = "red", lty = 2)
lines(forecast_25$upper, col = "red", lty = 2)
```



#10-Step Ahead Forecast starts immediately after the last observed data point and extends for 10-time units into the future. The confidence interval appears to be widening as the forecast horizon increases, which is typical since uncertainty tends to grow with the length of the forecast period.
#25-Step Ahead Forecast also starts immediately after the last observed point and provides a longer view into the future.
#In both cases, the plots suggest that the model expects the series to continue following the pattern established in the historical data. However, because the confidence intervals increase over time, the precision of the forecast decreases the further out you go.