Name : Muntaha Binte Alam

ID : 150448389

Column : v88

(*The codes are written in purple italic fonts* and the results are in teal bold fonts, The assignment follows this format : first code, then plot and results from R code, some interpretations)

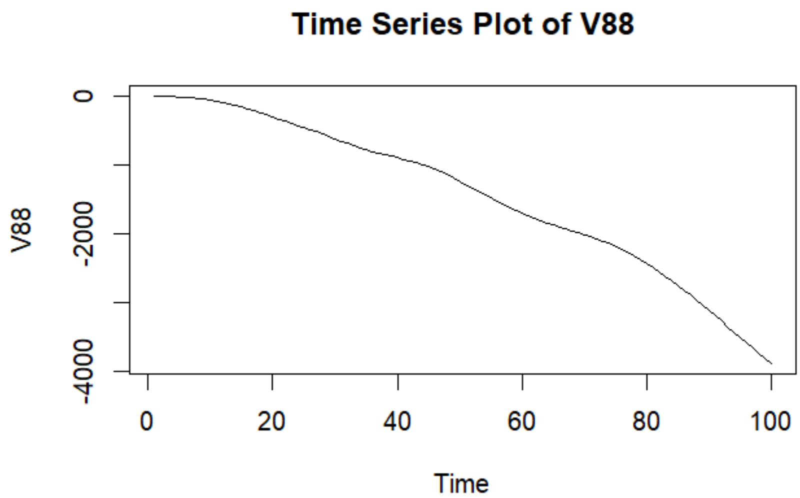
Step 1: Preliminary analysis of orders

1.

Yt <- ts(data$V88) # Retreiving v88 as its given according to my reg number

#plotting the time series

plot(Yt, main="Time Series Plot of V88", xlab="Time", ylab="V88")



Time series plot of V88 shows a decreasing trend over time. There is no evidence of seasonality or periodic fluctuations from this plot alone, but the overall trend is clearly downward.

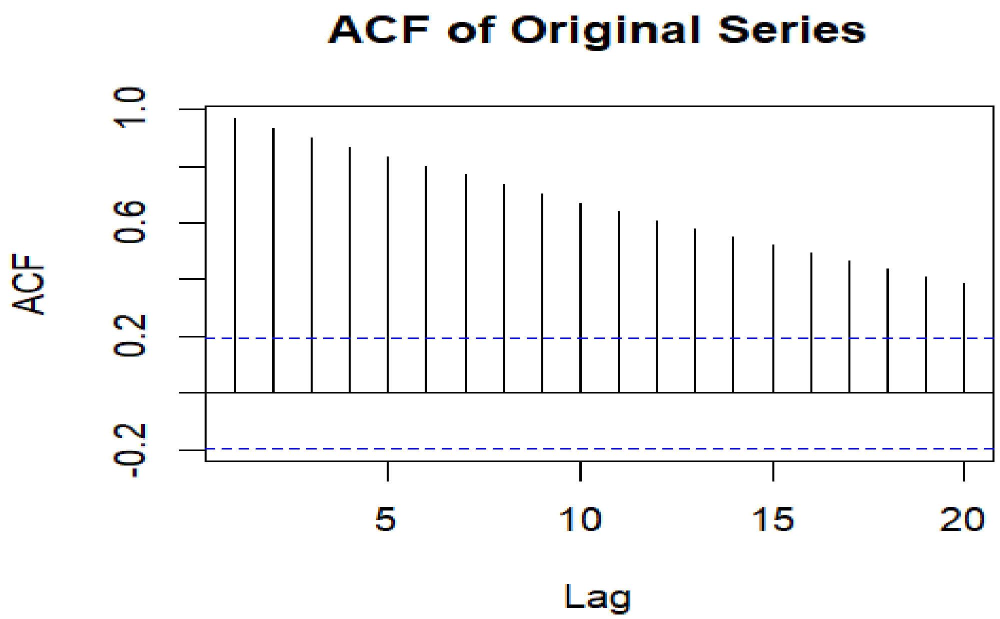
2. *# Displaying the ACF of the original series*

acf(Yt, main="ACF of Original Series")

 # Displaying the ACF after first differencing

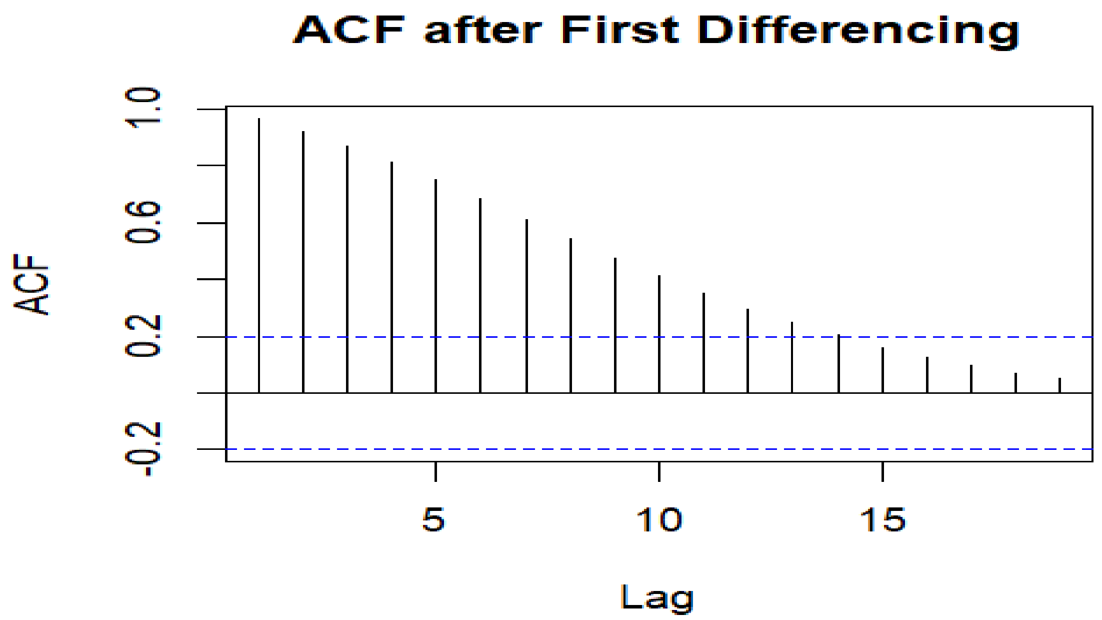
Yt_diff1 <- diff(Yt, differences=1)

```r
acf(Yt_diff1, main="ACF after First Differencing")

plot(Yt_diff1)


# Displaying the ACF after second differencing (if needed)

Yt_diff2 <- diff(Yt, differences=2)

acf(Yt_diff2, main="ACF after Second Differencing")

plot(Yt_diff2)
```



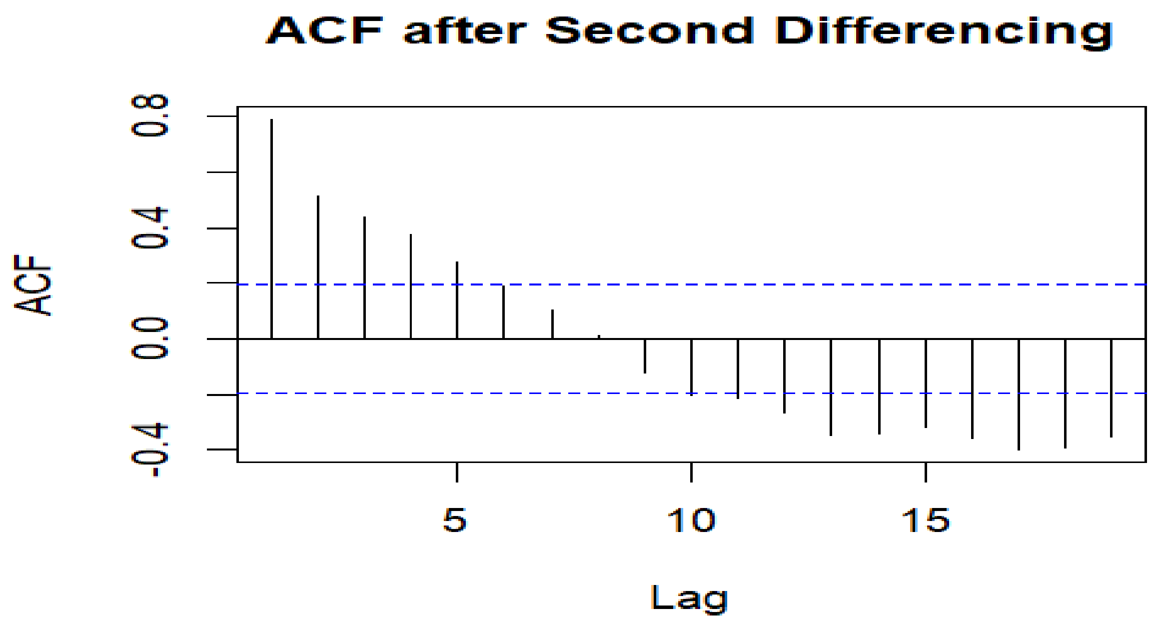
ACF of Original Series

The ACF of the original series shows a slow decay, which suggests that the data is likely non-stationary. This is a common characteristic of a time series with a trend.



ACF after First Differencing

The ACF after the first differencing still shows a gradual decay, but it's not as pronounced as the original series. However, the fact that it does not cut off quickly after a few lags suggests that the first differencing might not be sufficient to achieve stationarity.



ACF after Second Differencing

The ACF after the second differencing shows a much sharper cutoff after the first lag and the subsequent lags are within the confidence interval, which indicates that the series might be stationary after second differencing.

 # ADF test on the first differenced series

adf_test_diff1 <- adf.test(Yt_diff1, alternative="stationary")

print(adf_test_diff1)

 # ADF test on the second differenced series

adf_test_diff2 <- adf.test(Yt_diff2, alternative="stationary")

print(adf_test_diff2)

ADF Results :

Augmented Dickey-Fuller Test

data: Yt_diff1

Dickey-Fuller = -3.0757, Lag order = 4, p-value = 0.1312

alternative hypothesis: stationary

Augmented Dickey-Fuller Test

data: Yt_diff2

Dickey-Fuller = -2.2484, Lag order = 4, p-value = 0.4738

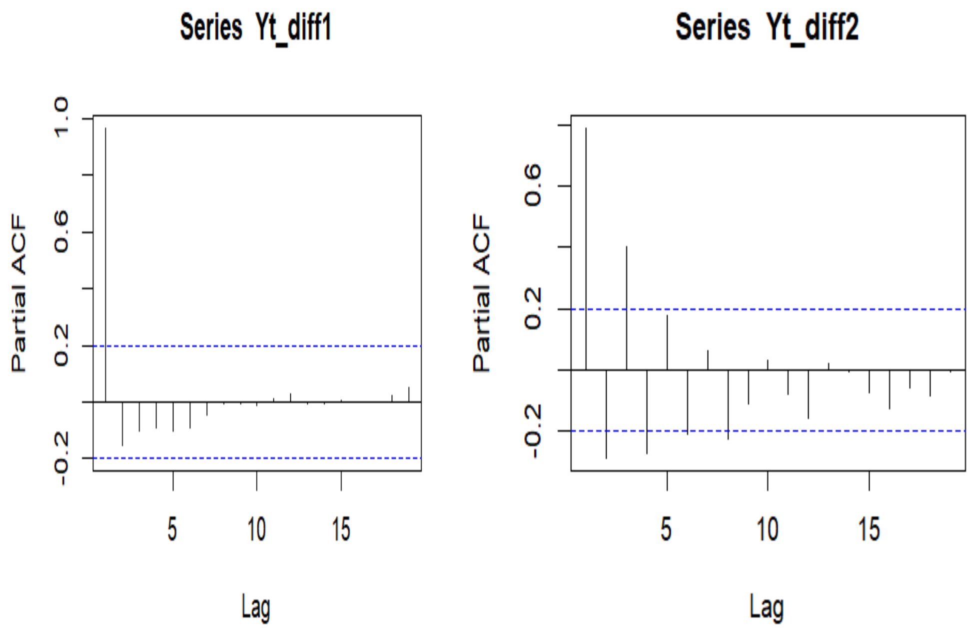
alternative hypothesis: stationary

I would suggest d = 2

3. *# Calculating and plotting the PACF for the second differenced series*

pacf(Yt_diff1)

pacf(Yt_diff2)



Based on the plots, there is no significant spikes in the PACF plot beyond the first lag for both the first and second differenced series. This would suggest a low or possibly zero order for the AR component p which can be 0 or 1.

Without the ACF plot of the residuals, it's challenging to set a bound for q precisely. But, as the ACF plot of the differenced series also cuts off early, similar to the PACF, this would suggest a low order for q as well. May be 0 or 1.

However, for the sake of model exploration and given the assignment's instructions, I will consider p(max) = 4 or q(max) = 4.

Step 2: Estimation and selection of ARIMA Models

```
# Defining the ranges for p and q
pmax <- 4
qmax <- 4
d <- 2  # Replace with the actual order of differencing determined from part 2

# Initializing a data frame to store the results
results <- data.frame(p=integer(), q=integer(), AIC=double(), BIC=double())

# Loop over the range of p and q values to fit ARIMA models and calculate AIC/BIC
for (p in 1:pmax) {
 for (q in 1:qmax) {
```

```
    # Fit the ARIMA model with the given p, d, q
    model <- Arima(Yt, order=c(p, d, q))

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

# Removing models that are not allowed (0, q) and (p, 0)
results <- subset(results, !(p == 0 & q > 0) & !(p > 0 & q == 0))

# Sorting the results by AIC and BIC
results <- results[order(results$AIC),]
results_bic <- results[order(results$BIC),]

# Selecting the best three models by AIC
best_by_aic <- head(results, 3)

# Selecting the best three models by BIC
best_by_bic <- head(results_bic, 3)

# Printing the best models by AIC
print("Best models by AIC:")
print(best_by_aic)

# Printing the best models by BIC
print("Best models by BIC:")
print(best_by_bic)
```

After evaluating 16 ARIMA models,

Best Models Based on AIC:

ARIMA (3,2,1): AIC = 292.65, BIC = 305.57

ARIMA (1,2,4): AIC = 294.23, BIC = 309.74

ARIMA (1,2,2): AIC = 294.37, BIC = 304.71

Best Models Based on BIC:

ARIMA (1,2,1): AIC = 295.13, BIC = 302.89

ARIMA (1,2,2): AIC = 294.37, BIC = 304.71

ARIMA (3,2,1): AIC = 292.65, BIC = 305.57

In a table,

Model	AIC	BIC
ARIMA (3,2,1)	292.65	305.57
ARIMA (1,2,4)	294.23	309.74
ARIMA (1,2,2)	294.37	304.71
ARIMA (1,2,1)	295.13	302.89
ARIMA (1,2,2)	294.37	304.71
ARIMA (3,2,1)	292.65	305.57

Observations:

The ARIMA (3,2,1) model shows the lowest AIC, suggesting a good fit with sufficient complexity.

The ARIMA (1,2,1) model has the lowest BIC, indicating a preference for simpler models.

ARIMA (1,2,2) ranks well under both criteria, making it a balanced choice.

Considering AIC and BIC values, ARIMA(1,2,2) is quite a good choice, balancing model fit and simplicity.

Step 3: Diagnostic tests

1. # Fit ARIMA(3,2,1)

model_321 <- Arima(Yt, order=c(3,2,1))

 # Fit ARIMA(1,2,1)

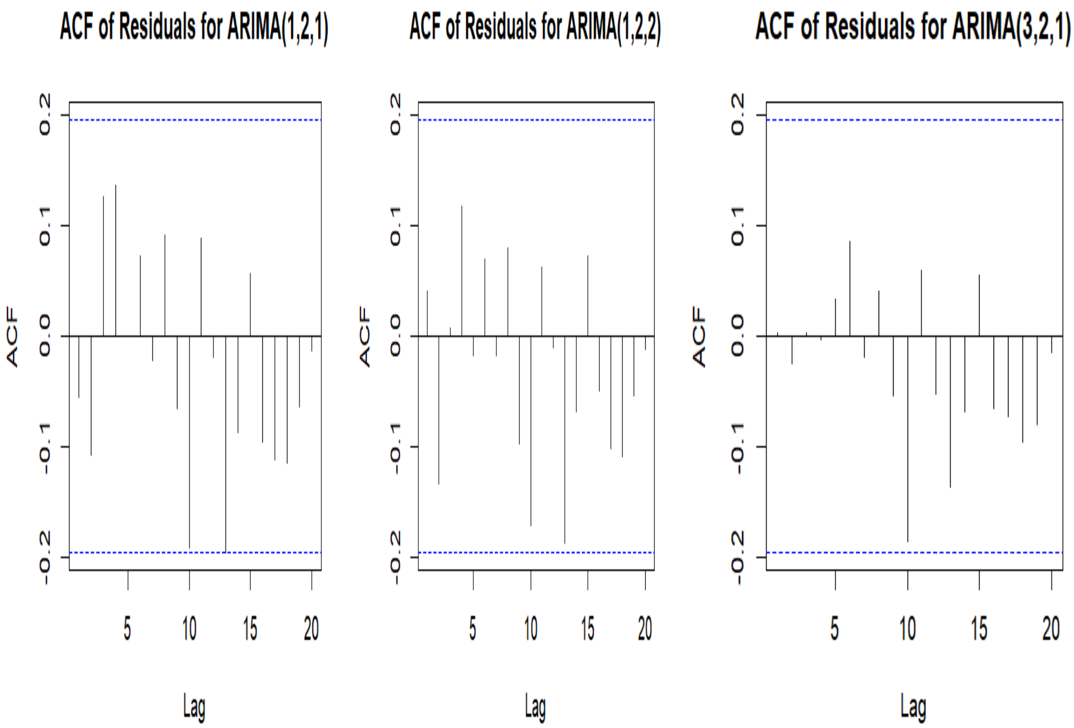
model_121 <- Arima(Yt, order=c(1,2,1))

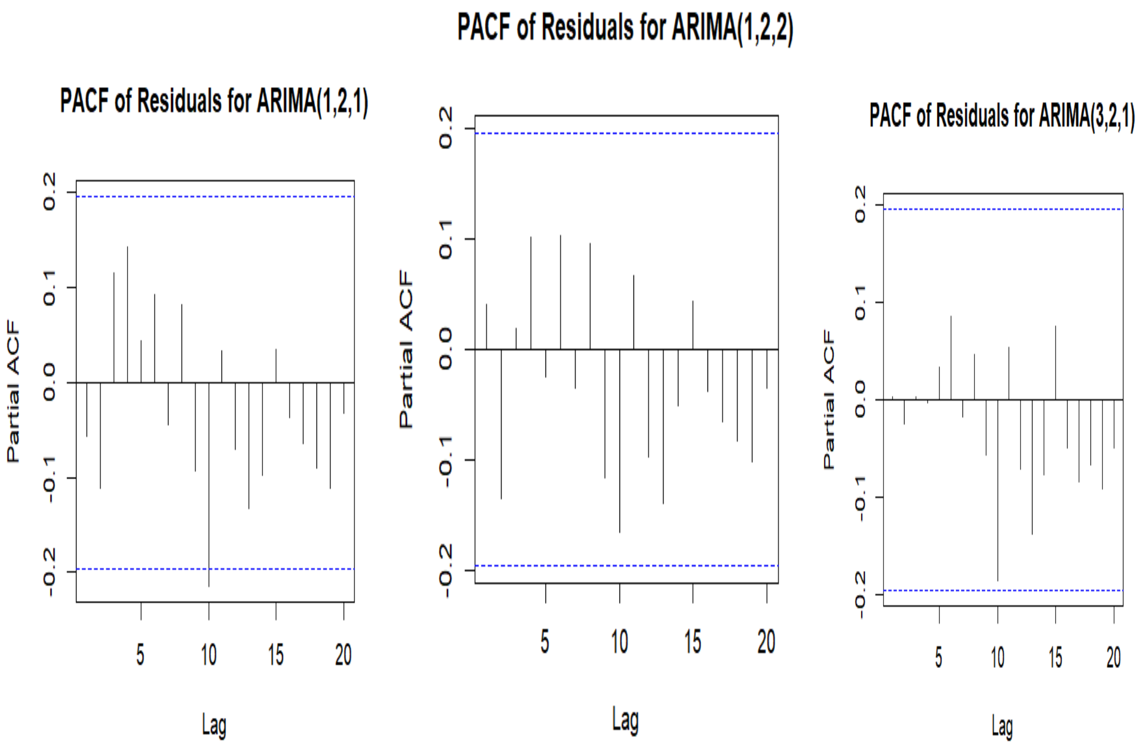
 # Fit ARIMA(1,2,2)

model_122 <- Arima(Yt, order=c(1,2,2))

#for ARIMA(3,2,1)

```
Box.test(residuals(model_321), lag=10, type="Ljung-Box")
acf(residuals(model_321), main="ACF of Residuals for ARIMA(3,2,1)")
pacf(residuals(model_321), main="PACF of Residuals for ARIMA(3,2,1)")

# for ARIMA(1,2,1)
Box.test(residuals(model_121), lag=10, type="Ljung-Box")
acf(residuals(model_121), main="ACF of Residuals for ARIMA(1,2,1)")
pacf(residuals(model_121), main="PACF of Residuals for ARIMA(1,2,1)")

# for ARIMA(1,2,2)
Box.test(residuals(model_122), lag=10, type="Ljung-Box")
acf(residuals(model_122), main="ACF of Residuals for ARIMA(1,2,2)")
pacf(residuals(model_122), main="PACF of Residuals for ARIMA(1,2,2)")
```





LjungBox test Results :

Box-Ljung test

data: residuals(model_321)

X-squared = 5.4357, df = 10, p-value = 0.8602

Box-Ljung test

data: residuals(model_121)

X-squared = 11.359, df = 10, p-value = 0.3302

Box-Ljung test

data: residuals(model_122)

X-squared = 9.1597, df = 10, p-value = 0.517

The ACF plots for ARIMA (1,2,1), ARIMA (1,2,2), and ARIMA (3,2,1), all the autocorrelations at different lags are within the confidence bounds, indicating no significant autocorrelation.

Similarly, the PACF plots show that partial autocorrelations are within the confidence bounds, suggesting no significant autoregressive structure is left uncaptured by the model.

Ljung-Box Test Results:

The p-values for the Ljung-Box test for all three models are well above the typical significance level of 0.05. This suggests that there is no significant evidence of autocorrelation in the residuals at the first 10 lags for any of the models, implying that the models are capturing the time series' structure adequately.

2.

```
# Histogram of residuals
hist(residuals(model_321), main="Histogram of Residuals (model 321)", xlab="Residuals")
hist(residuals(model_121), main="Histogram of Residuals (model 121)", xlab="Residuals")
hist(residuals(model_122), main="Histogram of Residuals (model 122)", xlab="Residuals")

# QQ plot of residuals
qqnorm(residuals(model_321))
qqline(residuals(model_321), col = "red")

qqnorm(residuals(model_121))
qqline(residuals(model_121), col = "blue")

qqnorm(residuals(model_122))
qqline(residuals(model_122), col = "green")

# Shapiro-Wilk normality test
shapiro.test(residuals(model_321))
shapiro.test(residuals(model_121))
shapiro.test(residuals(model_122))
```

Shapiro-Wilk normality test

data: residuals(model_321)

W = 0.97605, p-value = 0.06525

Shapiro-Wilk normality test

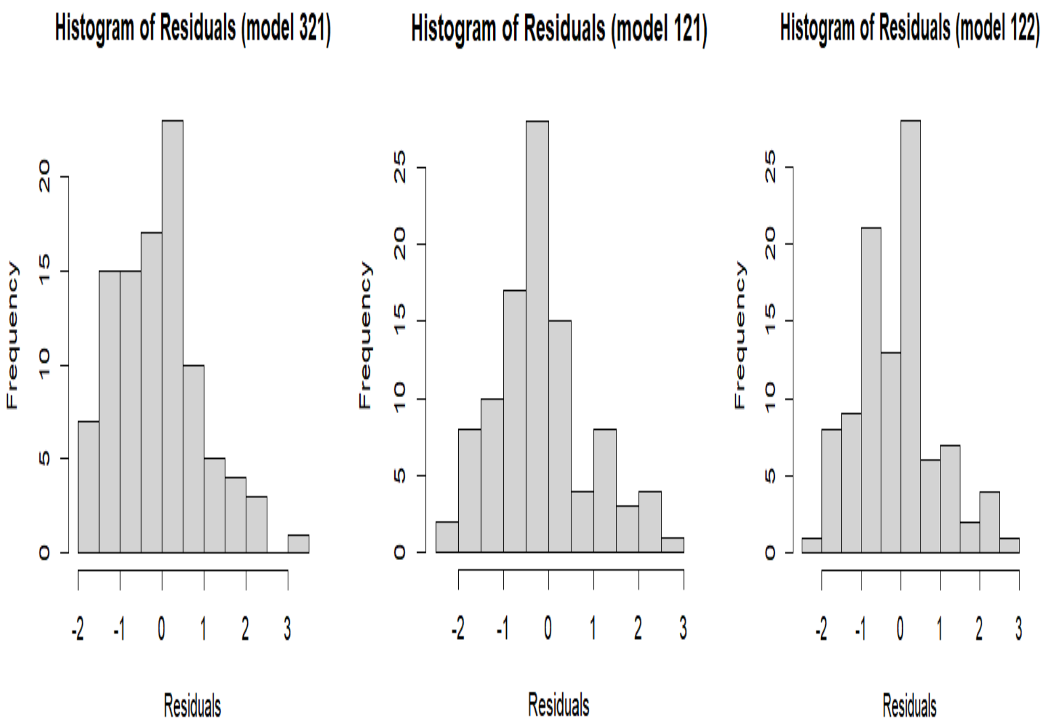
data: residuals(model_121)

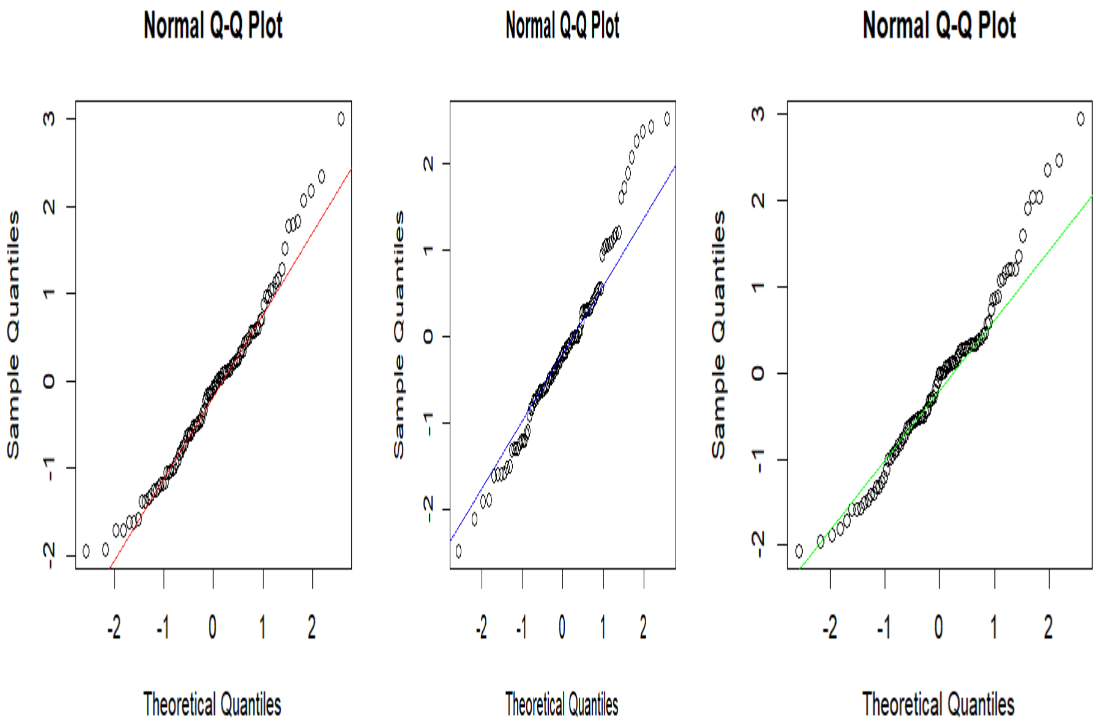
W = 0.97259, p-value = 0.03498

Shapiro-Wilk normality test

data: residuals(model_122)

W = 0.97553, p-value = 0.05937





Histograms: The histograms of residuals for all three models (ARIMA (3,2,1), ARIMA (1,2,1), and ARIMA (1,2,2)) show a roughly bell-shaped distribution, which is consistent with normality. However, some slight asymmetry is visible, particularly for the ARIMA (1,2,1) and ARIMA (1,2,2) models.

Q-Q Plots: The Q-Q plots for the residuals of the ARIMA (3,2,1) and ARIMA (1,2,2) models show a reasonably good fit to the line, especially towards the centre of the distribution, which is a sign of normality. The plot for the ARIMA (1,2,1) model shows some deviation from the line, suggesting potential departures from normality.

Shapiro-Wilk Test:

For the ARIMA (3,2,1) model, the p-value is 0.06525, which is slightly above the 0.05 threshold, suggesting that the residuals could be considered normally distributed at a 5% significance level.

The ARIMA (1,2,1) model has a p-value of 0.03498, indicating the residuals may not be normally distributed since the p-value is below 0.05.

The ARIMA (1,2,2) model's p-value is 0.05937, which is just above 0.05, again suggesting that the residuals could be considered normally distributed at the 5% significance level.

the visual inspections and statistical tests suggest that the residuals from the ARIMA (3,2,1) and ARIMA (1,2,2) models are approximately normally distributed, while there is some indication of non-normality.

3.

To determine the preferred specification, let's go through the findings again:

AIC and BIC Values: The ARIMA (3,2,1) model had the lowest AIC, suggesting the best fit among the models, while the ARIMA (1,2,1) model had the lowest BIC, indicating a preference for its simplicity.

Ljung-Box Test: All models demonstrated a lack of autocorrelation in the residuals, which is good. However, no model stood out as superior based on this test alone.

Normality of Residuals: The residuals of both ARIMA (3,2,1) and ARIMA (1,2,2) were approximately normal based on the Shapiro-Wilk test, while the ARIMA (1,2,1) model showed some evidence against normality.

Principle of Parsimony: This principle suggests that we should prefer simpler models when possible. In this context, the ARIMA (1,2,1) model is simpler than the ARIMA (3,2,1) model because it has fewer parameters.

Given these considerations, the ARIMA (1,2,2) model appears to be a balanced choice as:

It ranks well by both AIC and BIC, indicating a good fit with reasonable complexity.

The residuals are approximately normal, which is satisfactory for many types of analysis.

It is more parsimonious than the ARIMA (3,2,1) model, having fewer parameters, which is in line with the principle of parsimony.

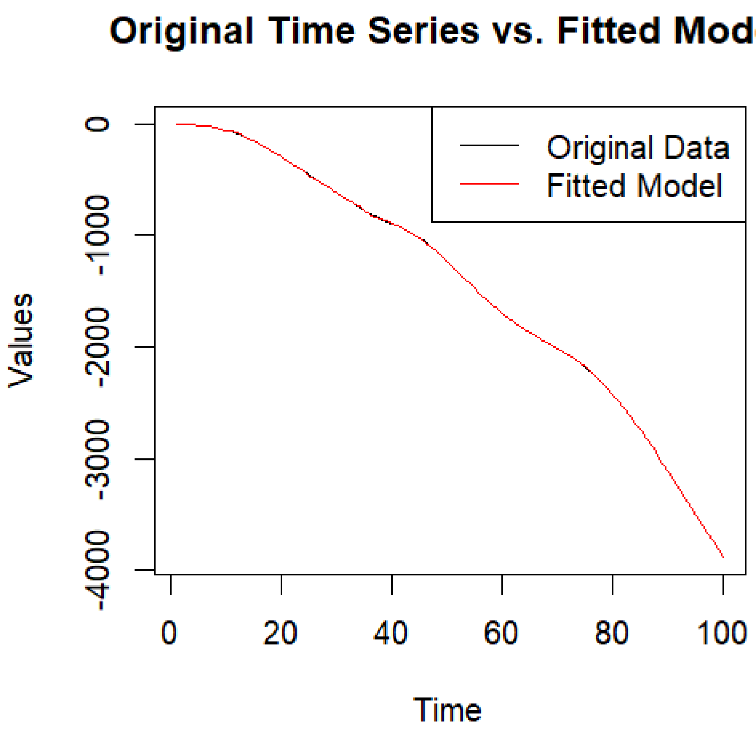
4.

```
# BY Choosing ARIMA(1,2,2) as best model

best_model <- Arima(Yt, order=c(1,2,2))


# Fitted values from the model

fitted_values <- fitted(best_model)


# Plot the original time series

plot(Yt, main="Original Time Series vs. Fitted Model", xlab="Time", ylab="Values")


# Lines function to add the fitted values to the plot

lines(fitted_values, col="red")
```

 # Add a legend to distinguish the data

legend("topright", legend=c("Original Data", "Fitted Model"), col=c("black", "red"), lty=1)

summary(best_model)



Series: Yt

ARIMA(1,2,2)

Coefficients:

	ar1	ma1	ma2
	0.8243	0.4739	-0.3605
s.e.	0.0962	0.1607	0.1572

sigma^2 = 1.094: log likelihood = -143.19

AIC=294.37   AICc=294.8   BIC=304.71

Training set error measures:

ME   RMSE   MAE   MPE   MAPE   MASE   ACF1

Training set -0.09850995 1.019547 0.7921779 0.3064836 0.5155947 0.02019752 0.04052256

Step 4: Forecast

 # 'model_122' is the fitted ARIMA(1,2,2) model and 'Yt' is original time series data

 # Forecast h=10 steps ahead

forecast_10 <- forecast(model_122, h=10)

 # Forecast h=25 steps ahead

forecast_25 <- forecast(model_122, h=25)

 # Plot the original time series and h-step predictions

par(mfrow=c(2, 1)) # Set up the plot area for two plots

 # Plot for h=10

plot(forecast_10, main="10-step Forecast with 95% Confidence Interval")

lines(Yt, col='blue')

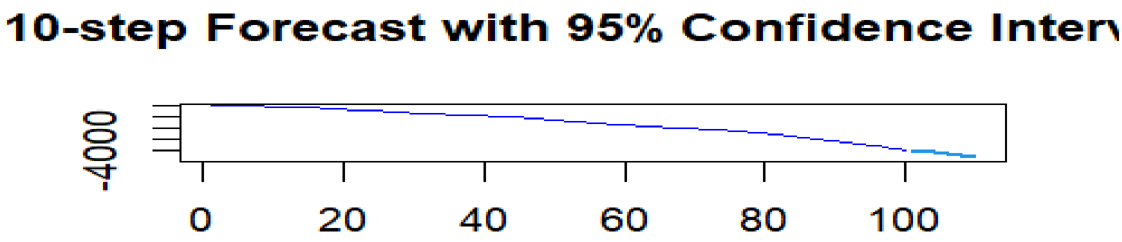
 # Plot for h=25

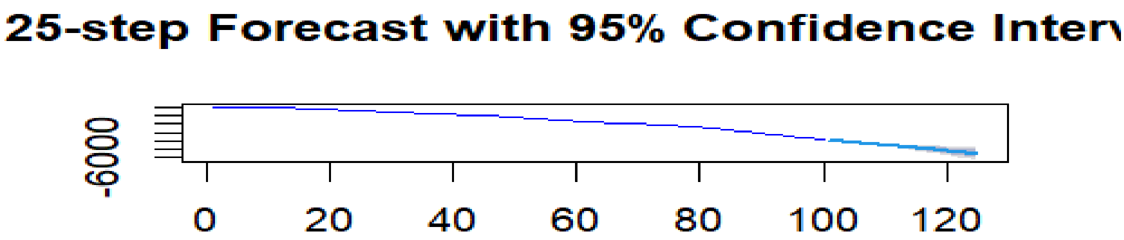
plot(forecast_25, main="25-step Forecast with 95% Confidence Interval")

lines(Yt, col='blue')

 # Reset plot area to default

par(mfrow=c(1, 1))





Both plots indicate that the model makes the series to continue its current trend into the future, with the 10-step forecast showing a slight decline and the 25-step forecast projecting a continued decline over a longer period.