Time Series Analysis: 780 (Tampere University)

Name: Maral Nourimand

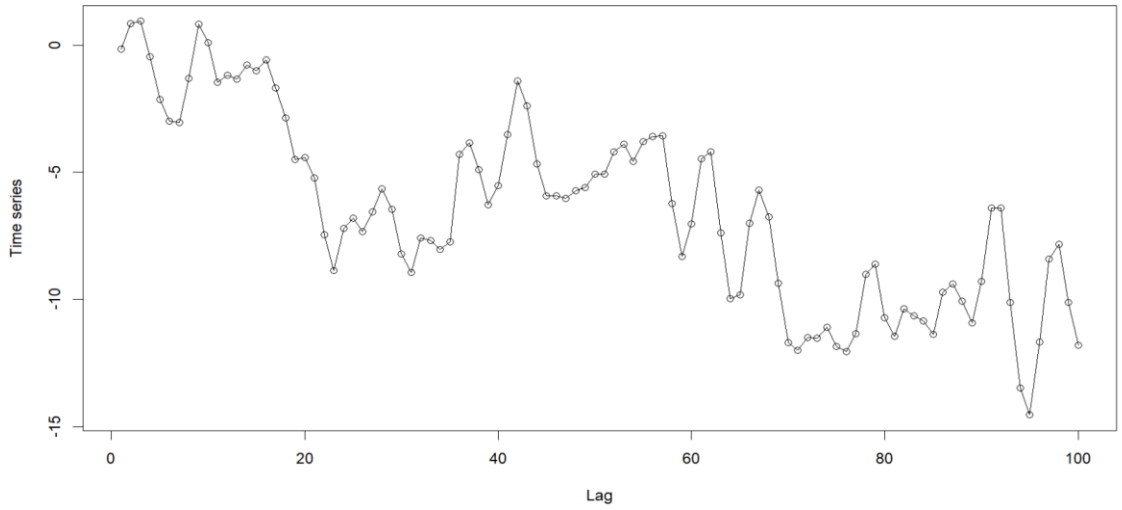
ID: 151749113

Column Assigned: V60

Assignment 1

Step 1: Preliminary analysis of Orders

1. Write your answer.



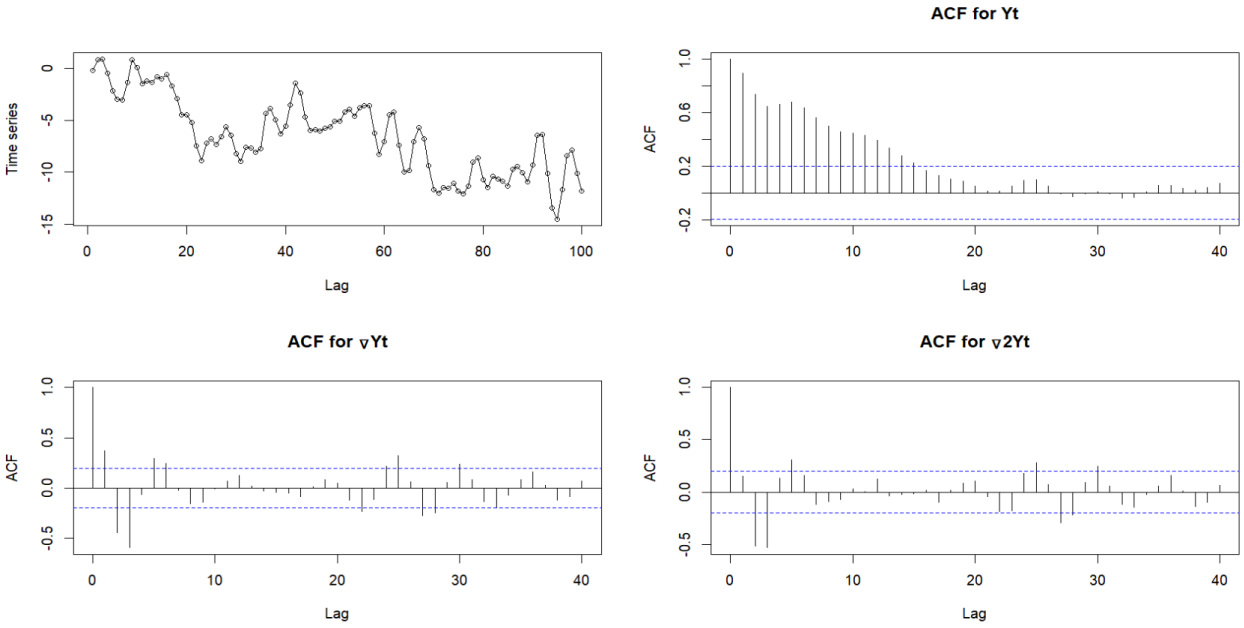

```
# read the data from the file
data <- read.csv("F:/Tampere Uni/Time Series/Project/Case_study.csv")

time_series <- data$V60 # my own column

# Plot the time series
plot(ts(time_series), ylab='Time series', xlab = 'Lag', type='o')
```

The time series consists of 100 points and it ranges between -15 to almost 0. It shows approximately a general trend of downward, however, we cannot be assured as the data includes only 100 observations and that is too small to draw any general conclusion.

2. Write your answer.

```
par(mfrow = c(2, 2))
plot(ts(time_series), ylab='Time series', xlab = 'Lag', type='o')

# Calculate differences (∇Yt and ∇2Yt)
diff1_Yt <- diff(time_series,1)
diff2_Yt <- diff(diff1_Yt)
```

```
# Plot the autocorrelation functions
acf( time_series , main ="ACF for Yt", lag.max = 40)
# for ∇Yt
acf( diff1_Yt , main ="ACF for ∇Yt", lag.max = 40)
# for ∇Yt
acf( diff2_Yt , main ="ACF for ∇2Yt", lag.max = 40)
```



As it is obvious in the figure above, the autocorrelation plot of the time series shows slow decay over lag. It means that the data in the original time series have high correlation at least until lag 20. However, in the difference ACF plots the decay in the correlation is faster and almost after 3 or 4 we do not see any significant high correlated result. In other words, we observe a decrease in the autocorrelation pattern after differencing.

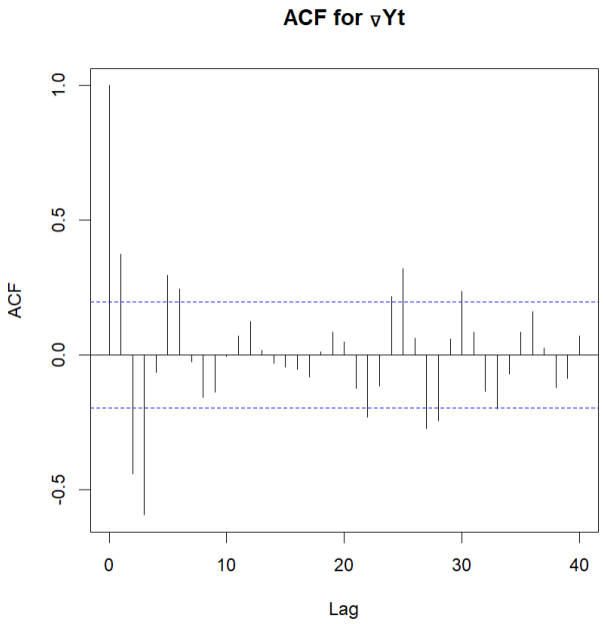
Observing the above plots, it suggests the differencing has made the series more stationary. As we don't see any noticeable alteration of autocorrelation pattern between the ∇Yt and ∇2Yt, we would conclude that only first order difference would be sufficient to reach stationary.

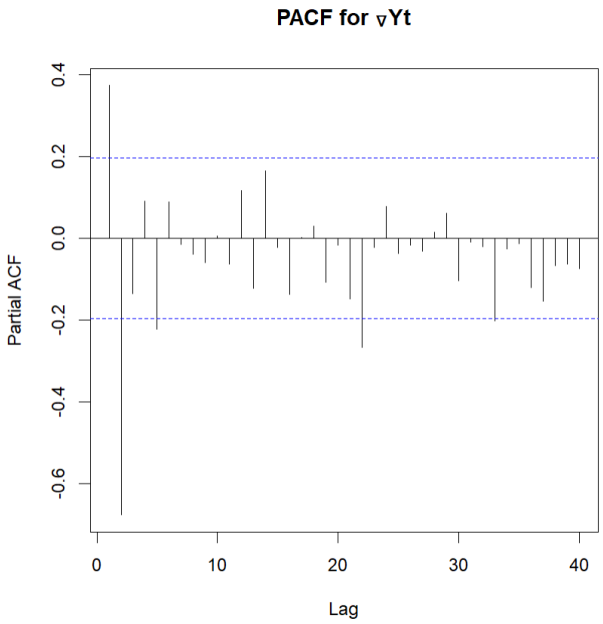
```
# Augmented Dickey - Fuller Unit Root Test
#install.packages("urca")
library ('urca')
summary (ur.df( time_series , type = "drift", lags = 1, selectlags =
"Fixed"))

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################
```

```
Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6330 -0.8252 -0.1016  0.9041  3.2331

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.89584    0.27453  -3.263  0.00153 **
z.lag.1     -0.12469    0.03717  -3.355  0.00114 **
z.diff.lag   0.43241    0.09158   4.722 8.06e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.315 on 95 degrees of freedom
Multiple R-squared:  0.2328,   Adjusted R-squared:  0.2166
F-statistic: 14.41 on 2 and 95 DF,  p-value: 3.422e-06


Value of test-statistic is: -3.3546 5.8575

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.51 -2.89 -2.58
phi1  6.70  4.71  3.86
```

The null hypothesis of the ADF test is that the time series has a unit root, and a p-value less than the significance level (commonly 0.05) suggests rejecting the null hypothesis, indicating stationarity for difference 1.
It is clear from the result that for difference 1 there is a significant reason to reject the unit root. In other words, the stationary solution exists for d = 1.
The negative ADF test statistic and the small p-value suggest evidence against the presence of a unit root, supporting the stationarity of the time series after differencing.
Therefore, based on the ADF test results, we have evidence to suggest that a first-order difference (d = 1) is appropriate to make our original time series stationary.

```
par(mfrow = c(1, 2))
# for ∇Yt
acf(diff1_Yt , main ="ACF for ∇Yt", lag.max = 40) # qmax=4

pacf(diff1_Yt , main ="PACF for ∇Yt", lag.max = 40) #pmax=2
```





By looking at the PACF plot, It is shown that there's a sharp drop-off after lag 2, it suggests an autoregressive component of order pmax = 2.
Similarly, the sharp drop-off in ACF plot after lag 4 is obvious. So, we can set qmax = 4.
With greater amount of observations, we probably could state upper bounds of p and q with more certainty.

Step 2: Estimation and Selection of ARIMA Models

1. Write your answer.

```
# Set up pmax and qmax
pmax <- 4
qmax <- 4

# Create a matrix to store AIC and BIC values
AIC_results <- matrix(NA, nrow = pmax , ncol = qmax)
BIC_results <- matrix(NA, nrow = pmax , ncol = qmax)

# Loop over possible (p, q) combinations
for (p in 1:pmax) {
  for (q in 1:qmax) {

    # Fit ARIMA model
    model <- arima(time_series, order = c(p, 1, q))

    # Compute AIC and BIC
    aic <- AIC(model)
    bic <- BIC(model)


    # Store AIC and BIC values in the matrix
    AIC_results[p, q] <- aic
    BIC_results[p, q] <- bic
  }
}

# Display the AIC and BIC values for each model
AIC_results   # min AIC p=3, q=1
BIC_results   # min BIC p=3, q=1


# Flatten the AIC results into a vector
AIC_vector <- as.vector(AIC_results)

# Find the indices of the three minimum AIC values
min_aic_indices <- order(AIC_vector)[1:3]
min_aic_indices

# Display the best three models based on AIC
best_models <- matrix(NA, nrow = 3, ncol = 2)
for (i in 1:3) {
  col_index <- floor((min_aic_indices[i] - 1) / ncol(AIC_results)) + 1
  row_index <- (min_aic_indices[i] - 1) %% ncol(AIC_results) + 1
  best_models[i, ] <- c(row_index, col_index)  # Subtract 1 to get the actual
p and q values
}

print("Best three models based on AIC:")
print(best_models)


# Flatten the BIC results into a vector
BIC_vector <- as.vector(BIC_results)

# Find the indices of the three minimum BIC values
min_bic_indices <- order(BIC_vector)[1:3]
min_bic_indices
```

```
# Display the best three models based on BIC
best_models_bic <- matrix(NA, nrow = 3, ncol = 2)
for (i in 1:3) {
  col_index <- floor((min_bic_indices[i] - 1) / ncol(BIC_results)) + 1
  row_index <- (min_bic_indices[i] - 1) %% ncol(BIC_results) + 1
  best_models_bic[i, ] <- c(row_index, col_index)  # get the actual p and q
values
}

print("Best three models based on BIC:")
print(best_models_bic)
```

16 different models are build and AIC and BIC of each model is saved in a matrix. Then the 3 minimum values of each matrix are extracted.

2. Write your answer.

The result of the 3 best models from the AIC and BIC is shown in the following table.

	AIC	p	q
Min1	283.8402	3	1
Min2	285.7802	4	1
Min3	285.7938	3	2

	BIC	p	q
Min1	296.8158	3	1
Min2	300.0882	2	1
Min3	301.3509	4	1

It seems that p=3, q=1 and p=4 and q=1 are the models confirmed by both of AIC and BIC as the best models. We should decide one model between Min3 from AIC and Min2 from BIC. To compare them precisely, another table is provided.

AIC	p	q
289.7077	2	1
285.7938	3	2

BIC	p	q
300.0882	2	1
301.3645	3	2

The table clarifies that the BIC value of p=2, q=1 is so close to the BIC value of p=3, q=2. On the other hand, there is more significant difference for AIC values with the same p and q pairs. So, we would keep the safe side and go with the lower AIC. It

means that our 3rd model would be p=3, q=2.

In summary the 3 specifications we will consider are as follows:
model.1 = arima(3,1,1)
model.2 = arima(4,1,1)
model.3 = arima(3,1,2)

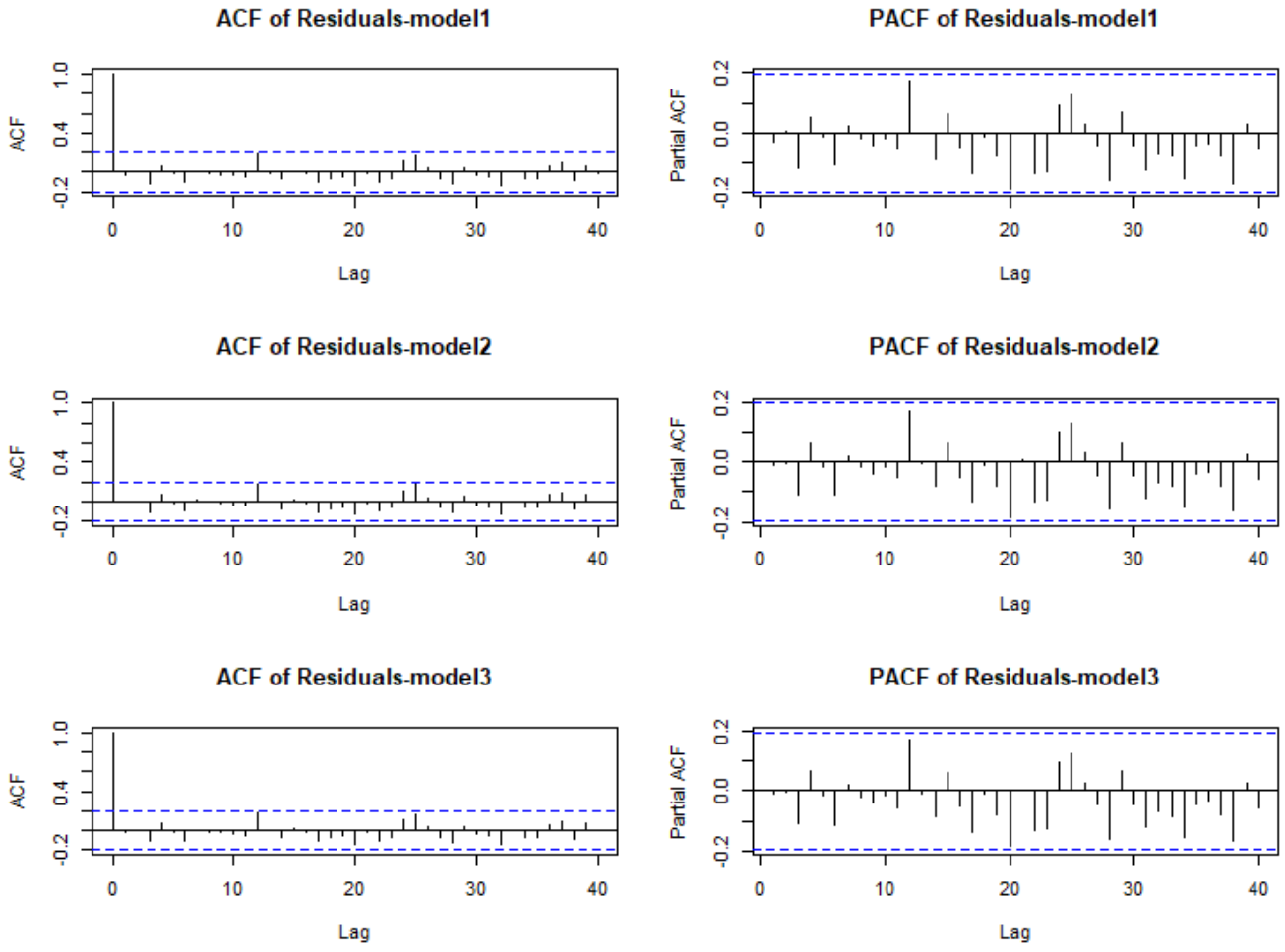
Step 3: Diagnostic tests

1. Write your answer.

```
# Fit ARIMA models
model.1 <- arima(time_series, order = c(3, 1, 1))
model.2 <- arima(time_series, order = c(4, 1, 1))
model.3 <- arima(time_series, order = c(3, 1, 2))


# Ljung-Box test
ljung_box_test <- Box.test(residuals(model.1), lag = 10, type = "Ljung-Box")
ljung_box_test  # p-value = 0.9817

ljung_box_test <- Box.test(residuals(model.2), lag = 10, type = "Ljung-Box")
ljung_box_test  # p-value = 0.9826

ljung_box_test <- Box.test(residuals(model.3), lag = 10, type = "Ljung-Box")
ljung_box_test  # p-value = 0.9827


windows(width = 8, height = 6)
# Plot ACF and PACF of residuals
par(mfrow=c(3,2))

# ACF plot of residuals
acf(residuals(model.1), lag.max = 40, main = "ACF of Residuals-model1")

# PACF plot of residuals
pacf(residuals(model.1), lag.max = 40, main = "PACF of Residuals-model1")

# ACF plot of residuals
acf(residuals(model.2), lag.max = 40, main = "ACF of Residuals-model2")

# PACF plot of residuals
pacf(residuals(model.2), lag.max = 40, main = "PACF of Residuals-model2")

# ACF plot of residuals
acf(residuals(model.3), lag.max = 40, main = "ACF of Residuals-model3")

# PACF plot of residuals
pacf(residuals(model.3), lag.max = 40, main = "PACF of Residuals-model3")

par(mfrow=c(1,1))  # Reset to a single plot
dev.off()
```

The code performs the Ljung-Box test for the first 10 lags to assess whether there are significant autocorrelations in the residuals.
If the p-value of the LjungBox test is large (typically > 0.05), it suggests that there is no evidence of autocorrelations in the residuals. The proceeding table shows the p-values of 3 models. As it is shown, all the 3 models show high p-value and reject strongly the existence of autocorrelations in the residuals.

LjungBox test result for 3 models			
	arima(3,1,1)	arima(4,1,1)	arima(3,1,2)
p-value	0.9817	0.9826	0.9827



The ACF and PACF plots show the autocorrelation and the partial autocorrelation, respectively, of residuals at different lags.
The plots do not show significant spikes outside the confidence bounds, it confirms that there are no significant autocorrelations in the residuals.

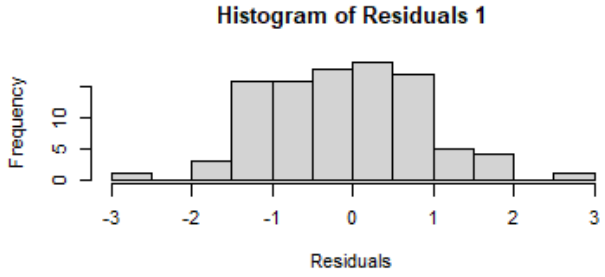
The lag 0 autocorrelation is essentially the correlation of each observation with itself, and it doesn't provide much information about the presence of a systematic pattern in the residuals.
We may conclude so far that our ARIMA(3,1,1) models are doing a good job of capturing the underlying patterns in the data.

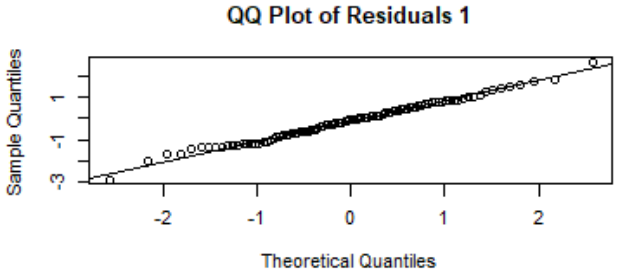
2. Write your answer.

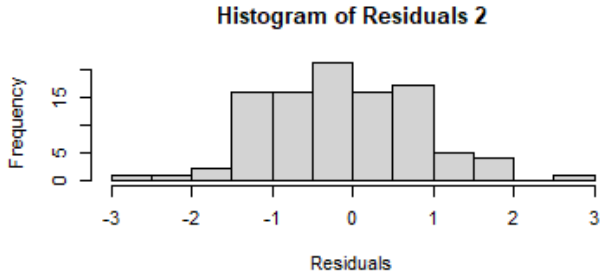
```
windows(width = 8, height = 6)
par(mfrow=c(3,2))
# Histogram of residuals
hist(residuals(model.1), main = "Histogram of Residuals 1", xlab =
"Residuals")

# QQ plot of residuals
qqnorm(residuals(model.1), main = "QQ Plot of Residuals 1")
qqline(residuals(model.1))

hist(residuals(model.2), main = "Histogram of Residuals 2", xlab =
"Residuals")

qqnorm(residuals(model.2), main = "QQ Plot of Residuals 2")
qqline(residuals(model.2))

hist(residuals(model.3), main = "Histogram of Residuals 3", xlab =
"Residuals")

qqnorm(residuals(model.3), main = "QQ Plot of Residuals 3")
qqline(residuals(model.3))

# Shapiro-Wilk test for normality
shapiro_test <- shapiro.test(residuals(model.1))
shapiro_test  #p-value = 0.8755

shapiro_test <- shapiro.test(residuals(model.2))
shapiro_test  #p-value = 0.902

shapiro_test <- shapiro.test(residuals(model.3))
shapiro_test  #p-value = 0.9065
```

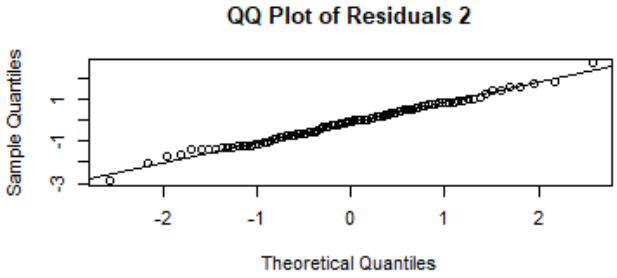
This code generates a histogram of residuals, a QQ plot, and performs the Shapiro-Wilk test for normality.
The histogram looks approximately symmetric and the QQ plot shows the residuals roughly following a straight line. So, it suggests normality.
The Shapiro-Wilk test provides a p-value, and a high p-value (> 0.05) suggests that the residuals do not deviate significantly from normality.
In these 3 models, the p-value is much greater than 0.05(it is shown in the table), it suggests that there is no strong evidence to reject the null hypothesis of normality for the residuals. Therefore, we may conclude that the residuals are normally distributed.

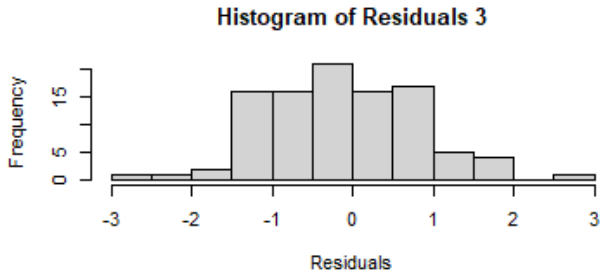
Shapiro-Wilk's test result for 3 models			
	arima(3,1,1)	arima(4,1,1)	arima(3,1,2)
p-value	0.8755	0.902	0.9065

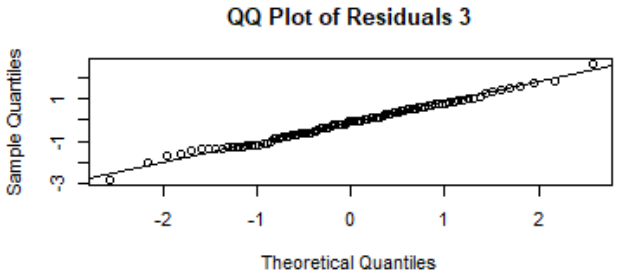












3. Write your answer.

Observing the diagnostic test results such as LjungBox and Shapiro-Wilk's test confirms the fact of normality that the residual plots are showing. All three models results in the normally distributed residuals and capture the underlying data sufficiently appropriate.

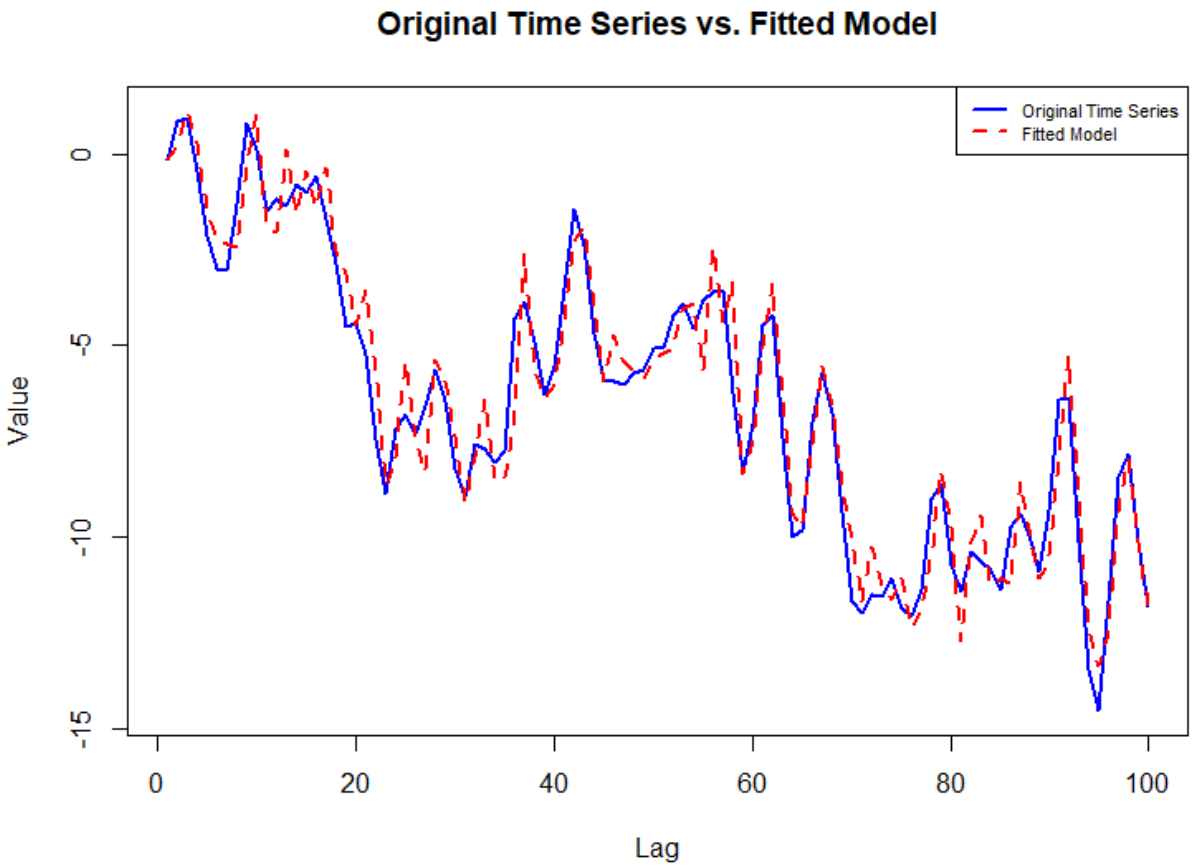
However we need to go with one model, and to decide wisely the principle of parsimony helps.
Selecting the ARIMA model with fewer parameters aligns with the principle of parsimony, which favours simplicity and avoids overfitting. Model.1 (3,1,1) has fewer parameters compared to Model.2 (4,1,1) and Model.3 (3,1,2). The ARIMA model (3,1,1) strikes a balance between capturing the underlying temporal dependencies in the data and avoiding excessive complexity. This choice ensures a more interpretable and efficient model, making it a prudent selection in accordance with the principle of parsimony.

4. Write your answer.

```
# Generate the best-fitted values from the ARIMA model(3,1,1)
library(forecast)
fitted_values1 <- fitted(model.1)

par(mfrow=c(1,1))

# Plot original time series
plot(time_series, type = "l", col = "blue", lwd = 2, main = "Original Time
Series vs. Fitted Model",
     xlab = "Lag", ylab = "Value",ylim = range(c(time_series,
fitted_values1)))

# Add best-fitted values to the plot
lines(fitted_values1, col = "red", lty = 2, lwd = 2)


# Add legend
legend("topright", legend = c("Original Time Series", "Fitted Model"),
       col = c("blue", "red"), lty = 1:2, cex = 0.7, lwd = 2)
```

The close overlap between the original time series and the fitted values suggests that the ARIMA(3,1,1) model has effectively captured the underlying patterns and temporal dependencies in the data. The minor discrepancies between the two may stem from inherent randomness or unmodeled nuances in the time series.

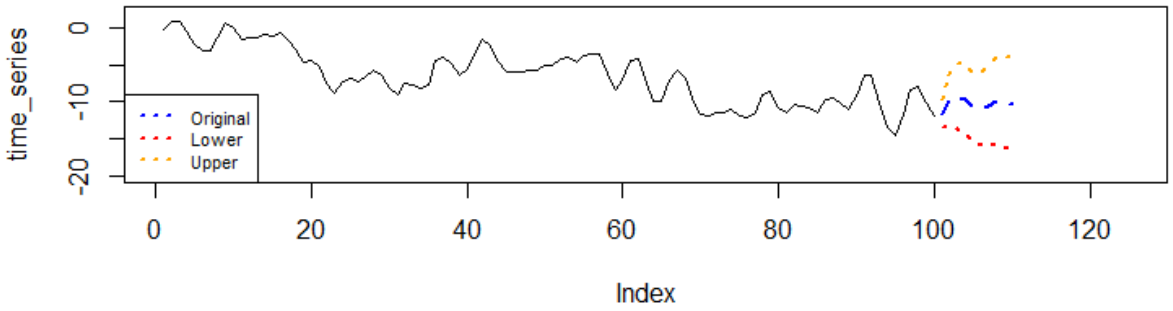
Overall, the strong alignment between the observed and fitted values affirms the model's ability to replicate the observed behaviour and reinforces its reliability in capturing the essential features of the original time series.

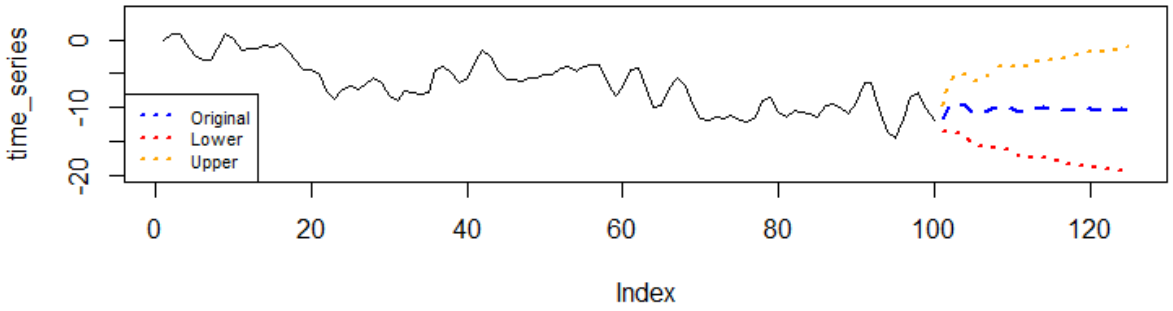

Original Time Series vs. Fitted Model
Original Time Series
Fitted Model
Value
0
-5
-10
-15
0
20
40
60
80
100
Lag

Step 4: Forecast

1. Write your answer.

```
# Forecast h=10 steps ahead
forecast_10 <- predict(model.1, n.ahead = 10)

# Forecast h=25 steps ahead
forecast_25 <- predict(model.1, n.ahead = 25)


windows(width = 8, height = 6)
par(mfrow=c(2,1))

x_axis <- seq_along(time_series)
# Plot original time series and forecast
plot(time_series, type = "l", xlim = c(1, length(time_series) + 25), ylim =
range(-20, 2))
lines(x_axis[length(time_series)] + (1:10), forecast_10$pred, lty = 2, col =
"blue", lwd = 2)
lines(x_axis[length(time_series)] + (1:10), forecast_10$pred - 1.96 *
forecast_10$se, lty = 3, col = "red", lwd = 2)
lines(x_axis[length(time_series)] + (1:10), forecast_10$pred + 1.96 *
forecast_10$se, lty = 3, col = "orange", lwd = 2)

# Add a legend to the plot
legend("bottomleft",
       legend=c("Original", "Lower", "Upper"),
       col=c("blue", "red", "orange"), lty=3, cex=0.7,lwd = 2)

plot(time_series, type = "l", xlim = c(1, length(time_series) + 25), ylim =
range(-20, 4))
lines(x_axis[length(time_series)] + (1:25), forecast_25$pred, lty = 2, col =
"blue", lwd = 2)
lines(x_axis[length(time_series)] + (1:25), forecast_25$pred - 1.96 *
forecast_25$se, lty = 3, col = "red", lwd = 2)
lines(x_axis[length(time_series)] + (1:25), forecast_25$pred + 1.96 *
forecast_25$se, lty = 3, col = "orange", lwd = 2)
```





The solid line represents the original time series data. The dashed pattern represents the forecasted values. Blue dashed pattern is the forecasted data based on the model itself, while the red and the orange ones are the lower bound and the upper bound of 95% confidence intervals. The 95% confidence interval is calculated based on the formula:
(mean – z* (std_dev), mean + z* (std_dev)).
Where z* = 1.96 for 95%.

The above plot represents the forecasted values for a horizon of 10 time steps (h=10). The forecast appears to follow the general trend of the original data. However, there is some deviation from the actual data, indicating uncertainty. The 95% confidence intervals (red and orange dashed lines) widen as we move into the future, emphasizing the increasing uncertainty in longer-term predictions.

The lower plot line represents the forecast for a longer horizon (h=25).
As expected, the confidence intervals widen further.

The forecast becomes less accurate due to the cumulative effect of uncertainty over a longer period. It's crucial to consider this uncertainty when making decisions based on the forecast.

In summary, while the ARIMA(3,1,1) model provides forecasts, the widening confidence intervals highlight the inherent uncertainty in predicting further into the future. Analysts should be cautious and consider the confidence intervals when interpreting and using these forecasts for decision-making.