TSA Assignment 2: U.S.A. Unemployment Rate Analysis and Forecast

TIME LORDS

1. Anurag Dinesh Karmarkar (s3829070) 2. Cleon Ozzie Rodrigues (s3826800)

3. Rahul Sanjay Halappanavar (s3815553)

Table of Contents

Introduction

Unemployment is one of the few issues that affect the socio-economic status in all nations of the world. The term unemployment refers to the individuals who are employable and are seeking a job actively but are unsuccessful in landing a job. The unemployment rate is measured by dividing the number of unemployed individuals by the total number of individuals in the workforce. This unemployment rate also acts as a factor in measuring the economic stability of a country. In this analysis, we will be talking about the unemployment rate of the United States of America.

Objective

The main objective of this report is to forecast and model the unemployment rate of the United States of America. We will be forecasting the unemployment rate for the next 12 months using different time series models. This forecast will be very vital for strategy and leaders to design and plan before time.

Data

The data that we have gathered is open source data from kaggle.com. The data consists of monthly unemployment rates from the year 1948 to 2019.

- The data has 13 columns :
  - Column 1 : Year
  - Column 2 to 13 : January to December unemployment rates

Data Preprocessing

In this section, first we import the packages that we would be needing and import the US unemployment dataset. As the year were split into 12 columns representing each month, we use the gather function to convert all the columns into one column which will consist of all the monthly data in one column called unemp_rate. Then we convert year and month into a date format. For sanity checks, we check for potential NA values in the data and we can see that there are no NA values in present. After this, we convert it to a time series object and visualise it.

```
#Import dataset and libraries
library(dplyr)
library(tidyr)
library(tseries)
library(TSA)
library(lmtest)
library(forecast)
library(nortest)
library(readr)
us_unemp <- read_csv("USUnemployment.csv")
us_unemp
```

```
## # A tibble: 72 x 13
##     Year   Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1948   3.4   3.8   4     3.9   3.5   3.6   3.6   3.9   3.8   3.7   3.8   4
```

```
##  2  1949   4.3   4.7   5     5.3   6.1   6.2   6.7   6.8   6.6   7.9   6.4   6.6
##  3  1950   6.5   6.4   6.3   5.8   5.5   5.4   5     4.5   4.4   4.2   4.2   4.3
##  4  1951   3.7   3.4   3.4   3.1   3     3.2   3.1   3.1   3.3   3.5   3.5   3.1
##  5  1952   3.2   3.1   2.9   2.9   3     3     3.2   3.4   3.1   3     2.8   2.7
##  6  1953   2.9   2.6   2.6   2.7   2.5   2.5   2.6   2.7   2.9   3.1   3.5   4.5
##  7  1954   4.9   5.2   5.7   5.9   5.9   5.6   5.8   6     6.1   5.7   5.3   5
##  8  1955   4.9   4.7   4.6   4.7   4.3   4.2   4     4.2   4.1   4.3   4.2   4.2
##  9  1956   4     3.9   4.2   4     4.3   4.3   4.4   4.1   3.9   3.9   4.3   4.2
## 10  1957   4.2   3.9   3.7   3.9   4.1   4.3   4.2   4.1   4.4   4.5   5.1   5.2
## # ... with 62 more rows
```

```
#Convert in 2 columns
us_unemp_consolidated <- us_unemp %>% gather(key = "Jan", value = "unemp_rate", c('Jan
':'Dec'))
us_unemp_consolidated <- us_unemp_consolidated %>% arrange(Year)
#Convert to date format
us_unemp_consolidated$date <- format(as.Date(paste0(us_unemp_consolidated$Jan, us_unem
p_consolidated$Year, "01"), format="%b%Y%d"), "%m-%Y")
us_unemp_ts <- us_unemp_consolidated %>% select(c(date,unemp_rate,-Year,-Jan))
us_unemp_ts
```

```
## # A tibble: 864 x 2
##    date    unemp_rate
##    <chr>        <dbl>
##  1 01-1948        3.4
##  2 02-1948        3.8
##  3 03-1948        4
##  4 04-1948        3.9
##  5 05-1948        3.5
##  6 06-1948        3.6
##  7 07-1948        3.6
##  8 08-1948        3.9
##  9 09-1948        3.8
## 10 10-1948        3.7
## # ... with 854 more rows
```

```
#Check for NA values
colSums(is.na(us_unemp_ts))
```

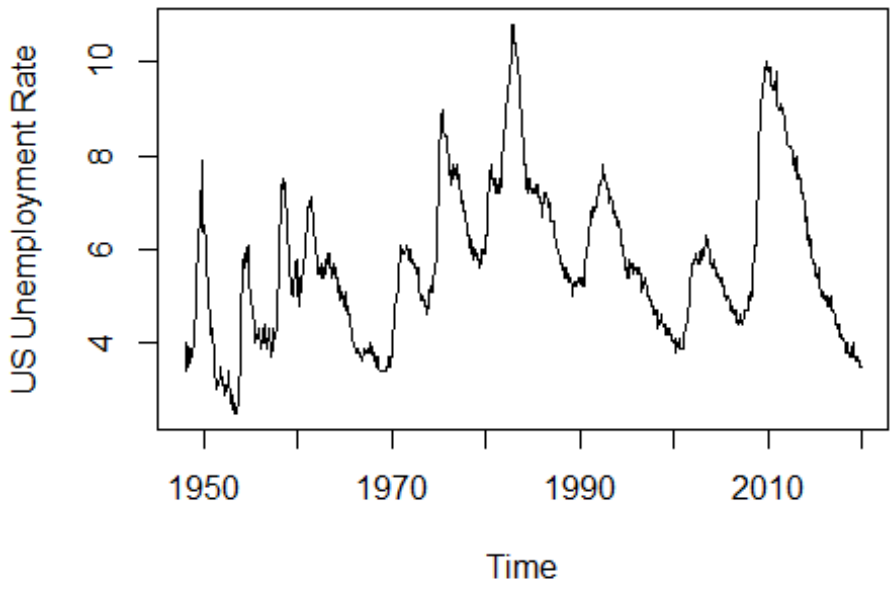
```
##       date unemp_rate
##          0          0
```

```
#Convert to time series
us_unemp_tsa <- ts(us_unemp_ts$unemp_rate, start=c(1948, 1), end=c(2019, 12), frequenc
y = 12)
class(us_unemp_tsa)
```

```
## [1] "ts"
```

```
#Plot the time series object
plot(us_unemp_tsa, ylab = 'US Unemployment Rate', main = 'Figure 1. US Unemployment Ra
te Time Series Plot')
```

Figure 1. US Unemployment Rate Time Series Plo



- From figure 1 we have following findings :
  - There is no evidence of trend but we need to confirm it using ACF plot.
  - The time series data does not appear to be seasonal.
  - There is changing variance in the time series.
  - There is moving average behavior with slight auto regressive pattern.
  - There is no change point seen.

Basic model fitting.

We try to fit the data using basic modeling techniques like liner model,quadratic model, cyclic tred model and harmoic trend mode.

Linear Model

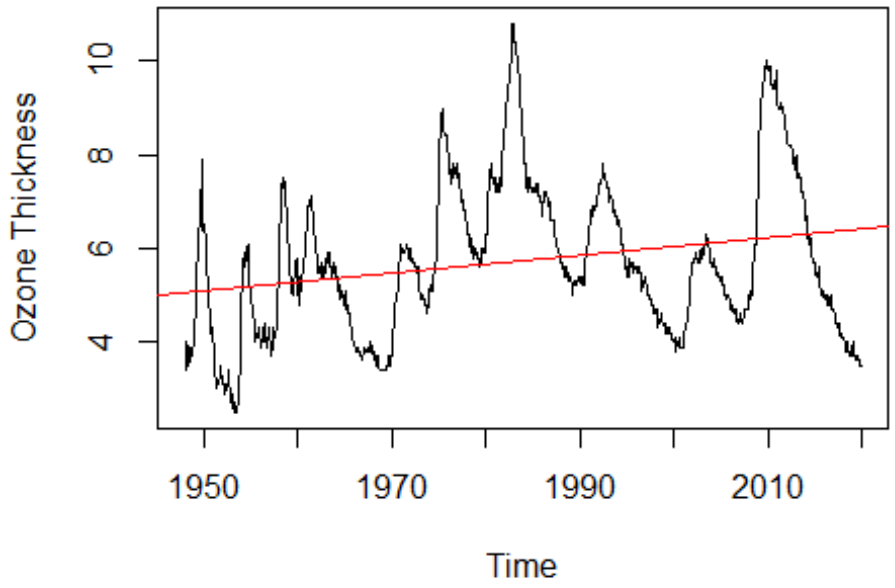
Here we fit our data into a Linear model

```
#Fitting the linear model
model_lm = lm(us_unemp_tsa~time(us_unemp_tsa))
summary(model_lm)

##
## Call:
## lm(formula = us_unemp_tsa ~ time(us_unemp_tsa))
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.9231 -1.2967 -0.2098  1.1154  5.0878
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -32.296800   5.187907  -6.225 7.49e-10 ***
## time(us_unemp_tsa)   0.019169   0.002615   7.331 5.26e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.597 on 862 degrees of freedom
## Multiple R-squared:  0.05869,    Adjusted R-squared:  0.0576
## F-statistic: 53.74 on 1 and 862 DF,  p-value: 5.261e-13
```

```
#Plotting the fitted model
plot(us_unemp_tsa,
     type='l',
     ylab='Ozone Thickness',
     main = "Figure 2. Fitted linear model to Us Unemployment Series")
abline(model_lm, col = 'red')
```

Figure 2. Fitted linear model to Us Unemployment Se



- On viewing the summary and plot we can see that :
  - The value of p is less than 0.05 level of significance.
  - Value of R-square is 0.05869
  - Value of Adjusted R-square is 0.0576
  - Ee can state that there is significance in the slope and there exists a trend but the data points are far off from the line.

Quadratic Model

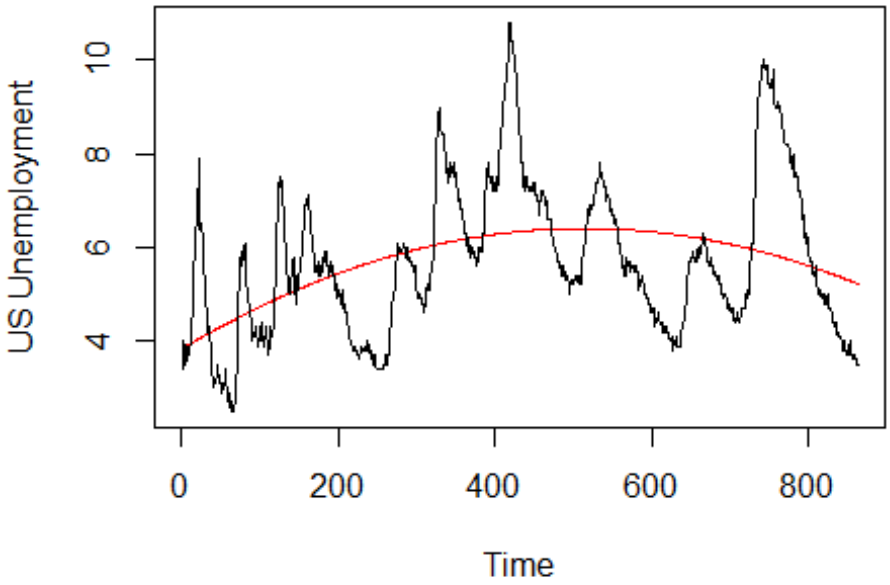
Here we fit our data into a Quadratic Model

```
#Fitting the quadratic  model
t = time(us_unemp_tsa)
t2 = t^2
model_qd = lm(us_unemp_tsa~t+t2)
summary(model_qd)
```

```
##
## Call:
## lm(formula = us_unemp_tsa ~ t + t2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.480 -1.103 -0.296  0.896  4.483
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.559e+03  5.209e+02  -10.67   <2e-16 ***
## t            5.591e+00  5.252e-01   10.65   <2e-16 ***
## t2          -1.404e-03  1.323e-04  -10.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.503 on 861 degrees of freedom
## Multiple R-squared:  0.1675, Adjusted R-squared:  0.1656
## F-statistic: 86.64 on 2 and 861 DF,  p-value: < 2.2e-16
```

```
#Plotting the built model
plot(ts(fitted(model_qd)),
     ylim = c(min(c(fitted(model_qd),
     as.vector(us_unemp_tsa))),
     max(c(fitted(model_qd),as.vector(us_unemp_tsa)))),
     ylab='US Unemployment',
     main = "Figure 3. Fitted quadratic curve to US Unemployment Series",
     col = 'red')
lines(as.vector(us_unemp_tsa), type="l")
```

gure 3. Fitted quadratic curve to US Unemployment S



- On viewing the summary and plot we can see that :
  - the value of p is less than 0.05 level of significance.
  - Value of R-square is 0.1675
  - Value of Adjusted R-square is 0.1656

  - We can even see that the Curve fits the model to certain extent but here as well the fit is not perfect as the data points are far off from the fit.

Harmonic Model

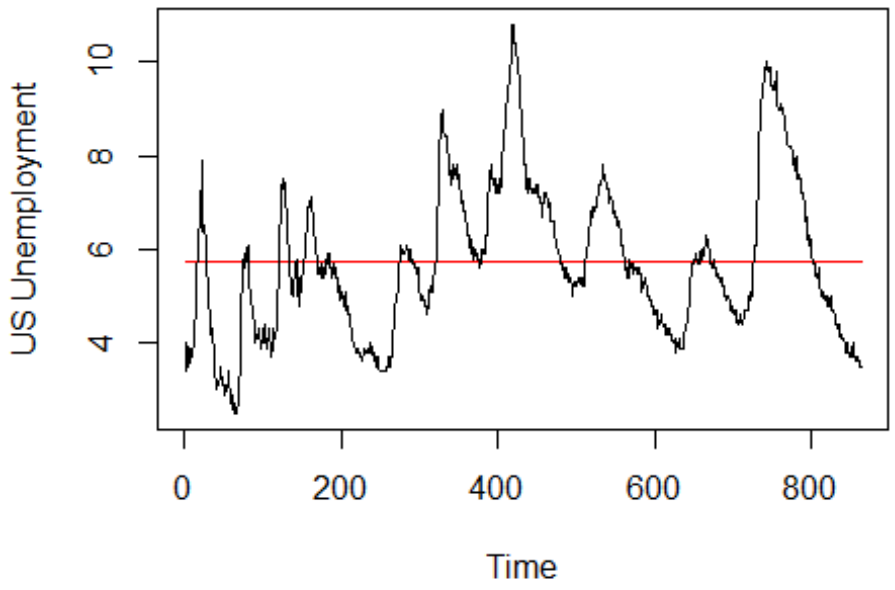
Here we fit our data into a Harmonic Model

```
#Fitting the harmonic model
har=harmonic(us_unemp_tsa,0.5)
model_cos=lm(us_unemp_tsa~har)
summary(model_cos)
```

```
##
## Call:
## lm(formula = us_unemp_tsa ~ har)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2392 -1.2314 -0.1838  1.0673  5.0716
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.733796   0.056048 102.301   <2e-16 ***
## harcos(2*pi*t) -0.002375   0.079264  -0.030    0.976
## harsin(2*pi*t)  0.004882   0.079264   0.062    0.951
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.647 on 861 degrees of freedom
## Multiple R-squared:  5.448e-06,  Adjusted R-squared:  -0.002317
## F-statistic: 0.002345 on 2 and 861 DF,  p-value: 0.9977

#Plotting the built model
plot(ts(fitted(model_cos)),
     ylab='US Unemployment',
     type='l',
     ylim=range(c(fitted(model_cos),us_unemp_tsa)),
     main="Figure 5. Fitted cosine model to US Unemployment Series",
     col = "red")
lines(as.vector(us_unemp_tsa),type="l")
```

Figure 5. Fitted cosine model to US Unemployment S



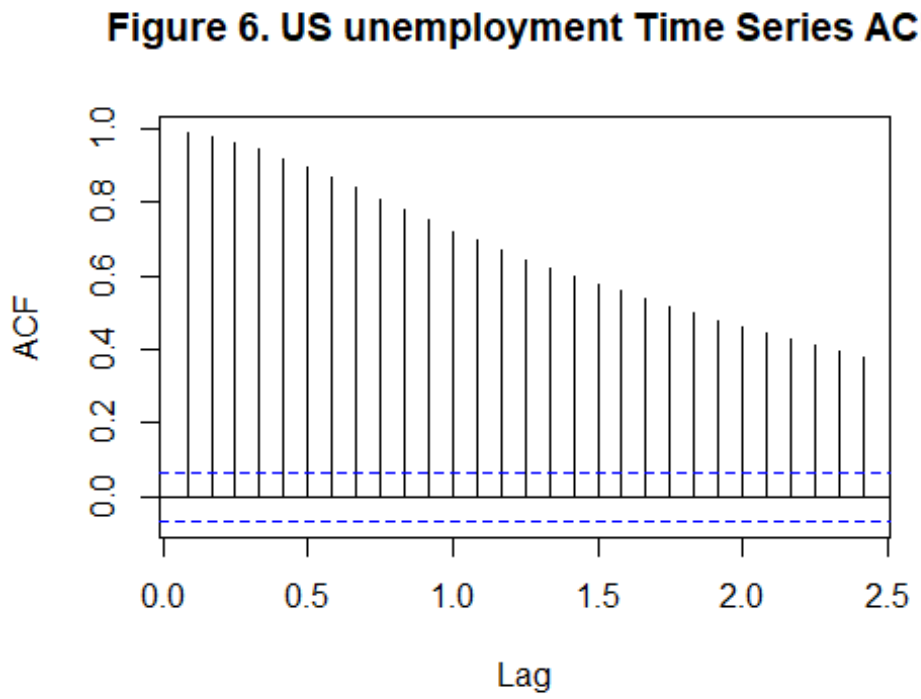
- On viewing the summary and plot we can see that :
  - the value of p is less than 0.05 level of significance.
  - Value of R-square is 0.9248.
  - Value of Adjusted R-square is 0.9237.
  - From the plot we can see that the points are not arranged along the cosine wave.

Fitting ARIMA Models

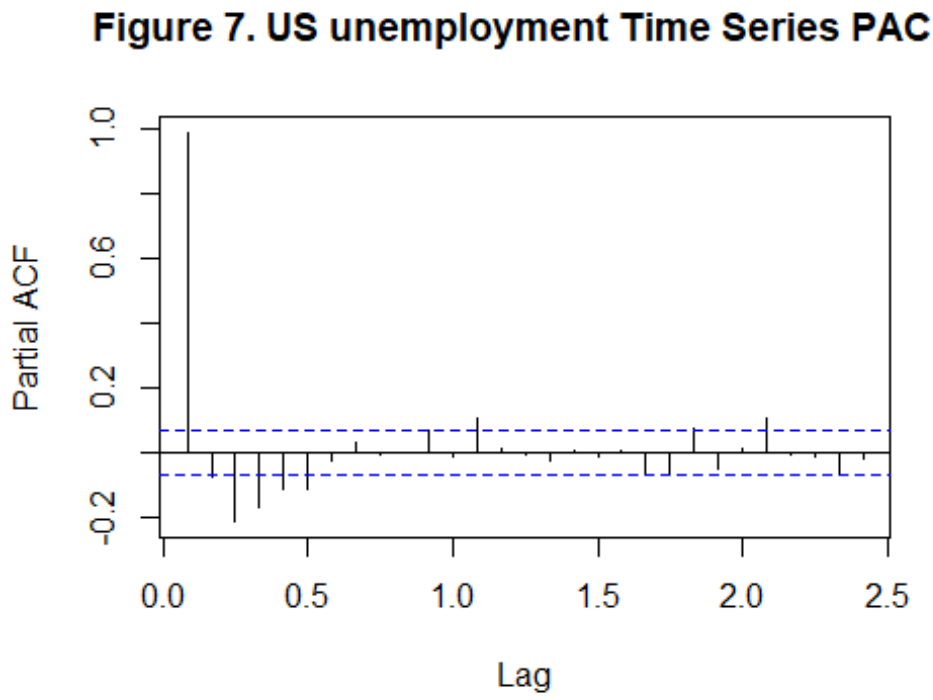
As we cannot determine any perfect fits from the above models we shift to ARIMA models to test their forecasts.

Ploting ACF and PACF

```
#ACF AND PACF tests
acf=acf(us_unemp_tsa,plot=FALSE)
plot(acf, main = "Figure 6. US unemployment Time Series ACF")
```




```
pacf=pacf(us_unemp_tsa,plot=FALSE)
plot(pacf, main = "Figure 7. US unemployment Time Series PACF")
```



- When we plotted the ACF and PACF of the time series data of US unemployment Rate, we see that
  - In the ACF plot, we can see that most of the lag data is outside the confidence bound and has a decaying pattern. This indicates that there is evidence of trend.
  - Where as for PACF, we see that the first value shows very high correlation.

ADF TEST

```
#ADF test
adf.test(us_unemp_tsa)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  us_unemp_tsa
## Dickey-Fuller = -3.7076, Lag order = 9, p-value = 0.02362
## alternative hypothesis: stationary
```

On running the Dickey-Fuller Root test, we get a p value of 0.02362 which is lesser than 0.05. Hence we can draw a conclusion that the data we have is stationary.

Now will try to use Box-Cox and Natural Logarithm transformation to get rid of the variance in the data that we have.

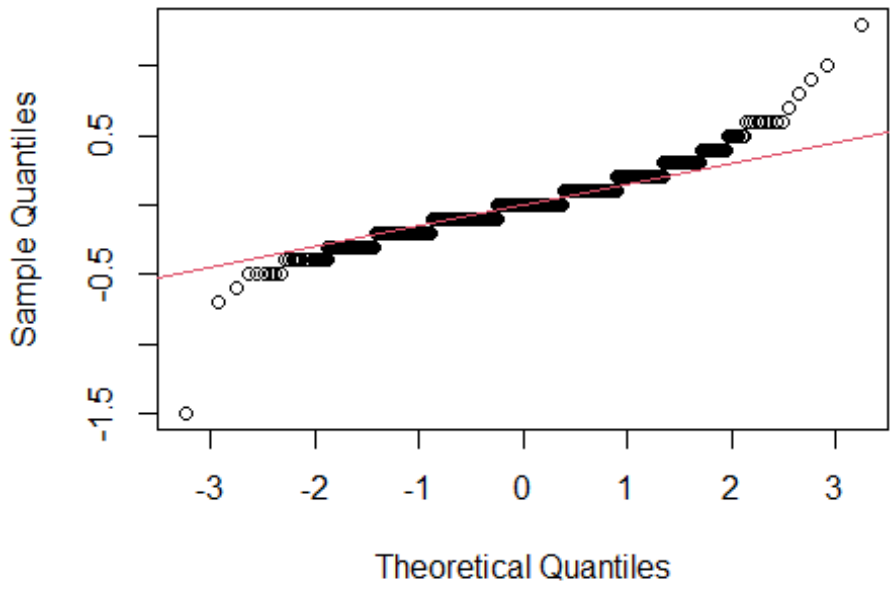
Test for stationarity on 1st differenced data

```
#Diffrencing by 1 to original data
us_unemp_tsa_diff <- diff(us_unemp_tsa, differences = 1)

#normality test and adf test for differenced data
adf.test(us_unemp_tsa_diff)
```

```
## Warning in adf.test(us_unemp_tsa_diff): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  us_unemp_tsa_diff
## Dickey-Fuller = -8.4377, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary
```

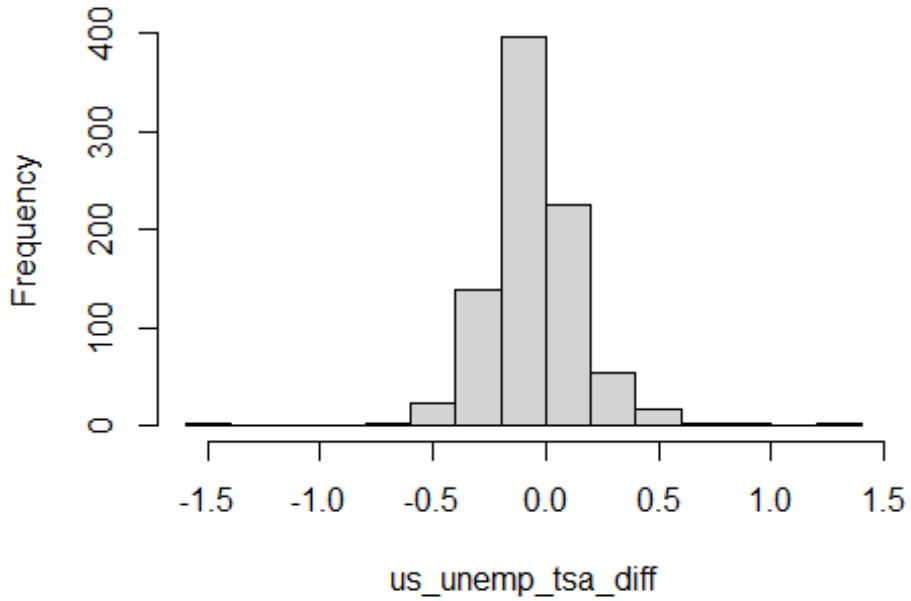
```
qq = qqnorm(us_unemp_tsa_diff, main =paste("Figure 8. QQ plot of 1st differenced data"
))
qqline(us_unemp_tsa_diff, col = 2)
```

Figure 8. QQ plot of 1st differenced data




```
hist(us_unemp_tsa_diff, main =paste("Figure 9. Histogram of 1st differenced data"))
```

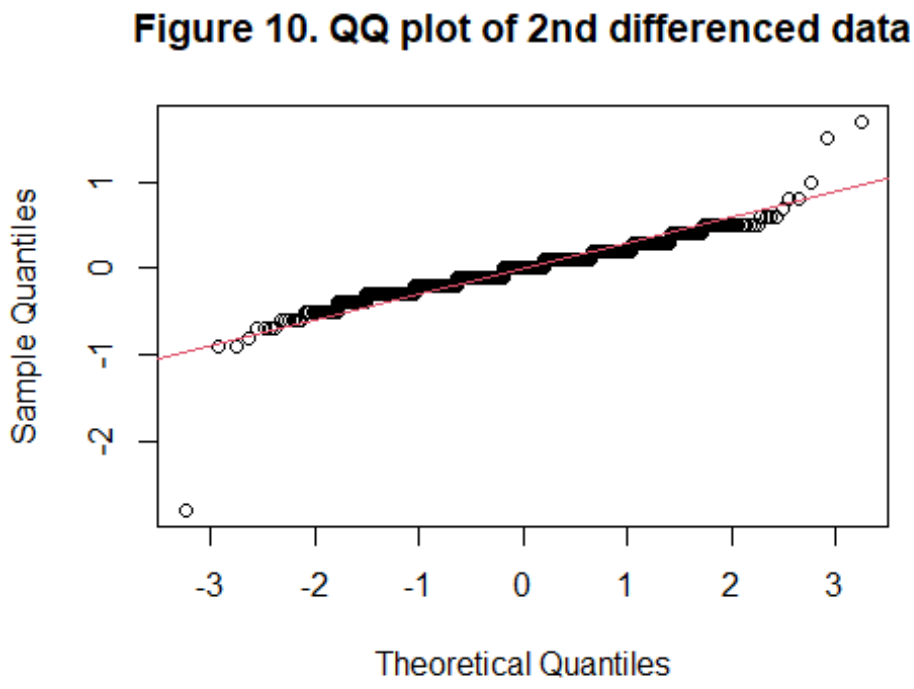
Figure 9. Histogram of 1st differenced data



- The results state that :
  - In the Dickey-Fuller Test, we get a p value of 0.01 which is greater than 0.05. Hence we can draw a conclusion that the data we have is stationary.

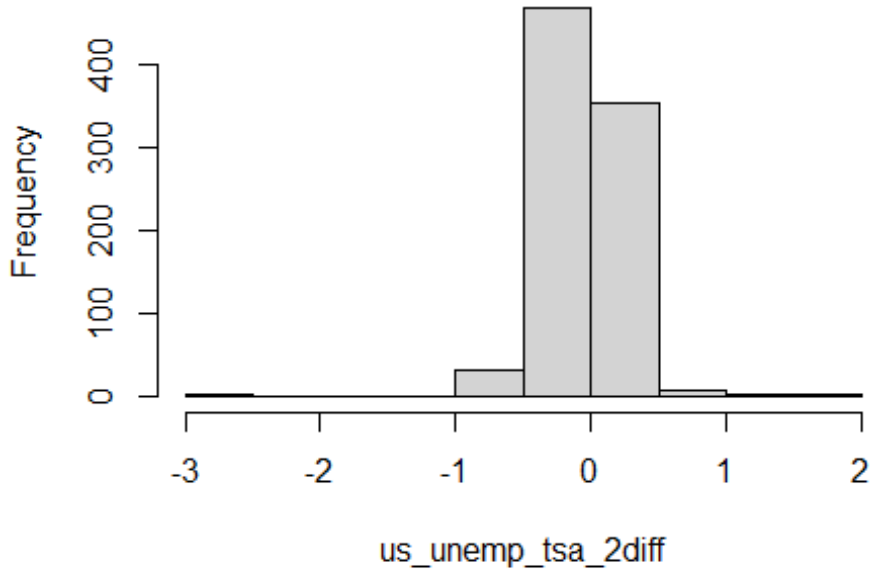
Test on for stationarity 2nd differenced data

```
#Diffrencing by 2 to original data
us_unemp_tsa_2diff <- diff(us_unemp_tsa, differences = 2)

#normality test and adf test for differenced data
adf.test(us_unemp_tsa_2diff)

## Warning in adf.test(us_unemp_tsa_2diff): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  us_unemp_tsa_2diff
## Dickey-Fuller = -12.481, Lag order = 9, p-value = 0.01
## alternative hypothesis: stationary

qq = qqnorm(us_unemp_tsa_2diff, main =paste("Figure 10. QQ plot of 2nd differenced dat
a"))
qqline(us_unemp_tsa_2diff, col = 2)
```




```
hist(us_unemp_tsa_2diff, main =paste("Figure 11. Histogram of 2nd differenced data"))
```

Figure 11. Histogram of 2nd differenced data



- The results state that
  - The QQ plot and histogram charts are visually close to noraml distribution.
  - In the Dickey-Fuller Test, we get a p value of 0.01 which is greater than 0.05. Hence we can draw a conclusion that the data we have is stationary.

Performing Normality Tests

In this step we create a funtion to plot QQ-plot, QQ-Line, histogram, Shapiro-Wilk test and Anderson-Darling test for chyecking the normality.

```
#Function for normality test
normality_test <- function(transform_used,fig) {
  qq = qqnorm(transform_used, main =paste("Figure ",fig,". QQ plot of transformed data
"))
  qqline(transform_used, col = 2)
  hist(transform_used, main =paste("Figure ",fig+1,". Histogram of transformed data"))
  sha = shapiro.test(transform_used)
  ad = ad.test(transform_used)
  testlist = list(sha, ad)
  return(testlist)
}
```

Test on base data for normality

```
#Normality test of the original data
normality_test(us_unemp_tsa,12)
```

Figure 12 . QQ plot of transformed data

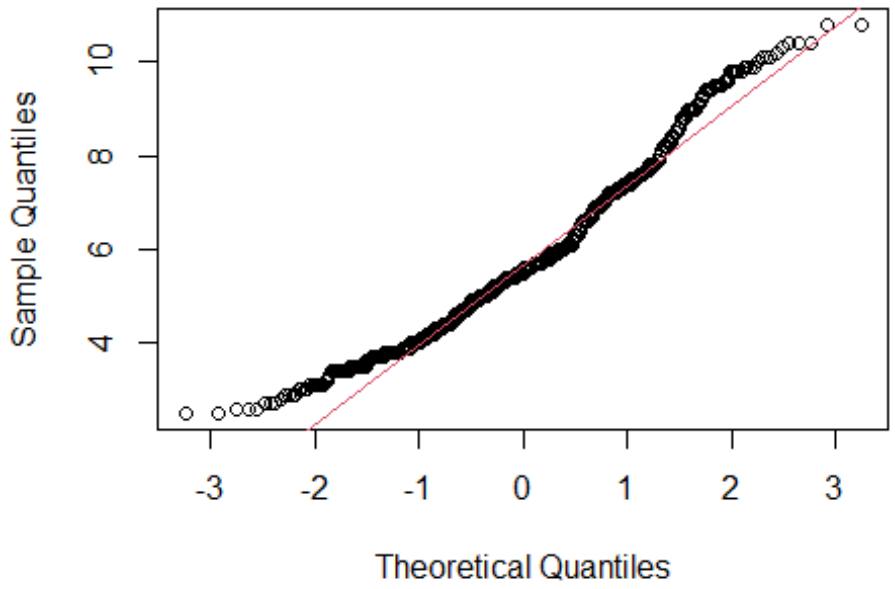
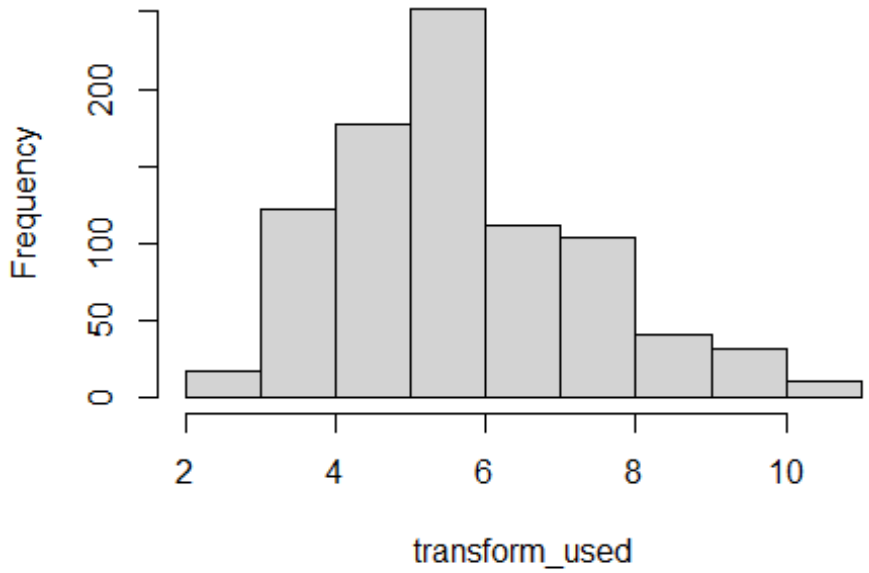


Figure 13 . Histogram of transformed data




```
## [[1]]
##
##  Shapiro-Wilk normality test
##
## data:  transform_used
## W = 0.96679, p-value = 4.029e-13
##
##
```

```
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  transform_used
## A = 7.6024, p-value < 2.2e-16
```

- The results state that :
  - The points at the start and end of the QQ plot seem to deviate a lot from the line. But the data in the middle fits in a good way on the line.
  - From the histogram, we can see that the curve for the data is a bit right skewed. The data on the right hand side of the plot has lesser values as compared to the values on the left hand side of the plot.
  - In the Shapiro-Wilk Test, the value of p is 4.029e-13, which is lesser than 0.05, we can say that we reject null hypothesis of normal distribution.
  - In the Anderson-Darling Test, the p-value is 2.2e-16 which is lesser than 0.05, we can reject the null hypothesis and conclude that we have sufficient evidence to say this data does not follow a normal distribution.

Test on Box-cox data for normality

```
#Box-cox transformation
lambda <- BoxCox.lambda(us_unemp_tsa)
print(lambda)
```

```
## [1] 0.5047327
```

```
#Normality test of transformed data
us_unemp_tsa_bc = BoxCox(us_unemp_tsa, lambda=lambda)
normality_test(us_unemp_tsa_bc,14)
```

Figure 14 . QQ plot of transformed data

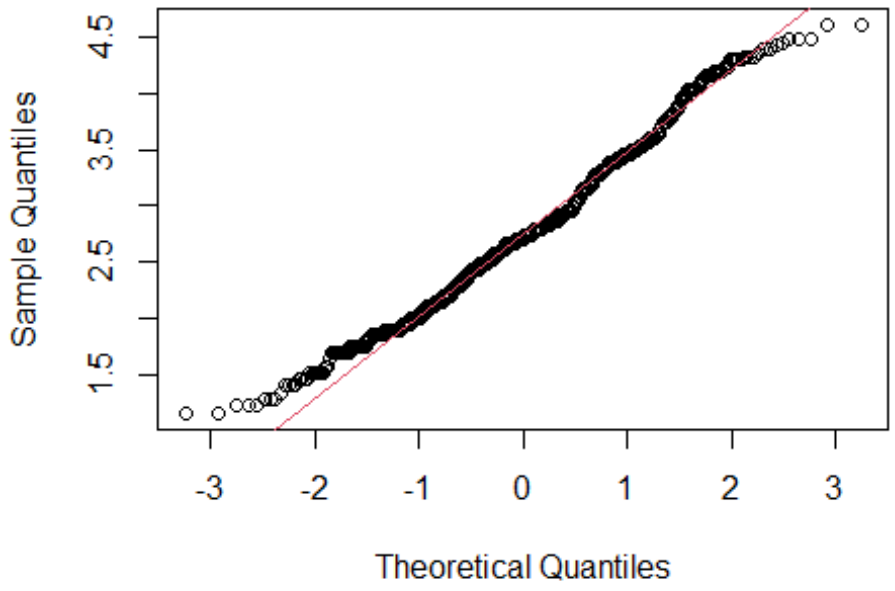
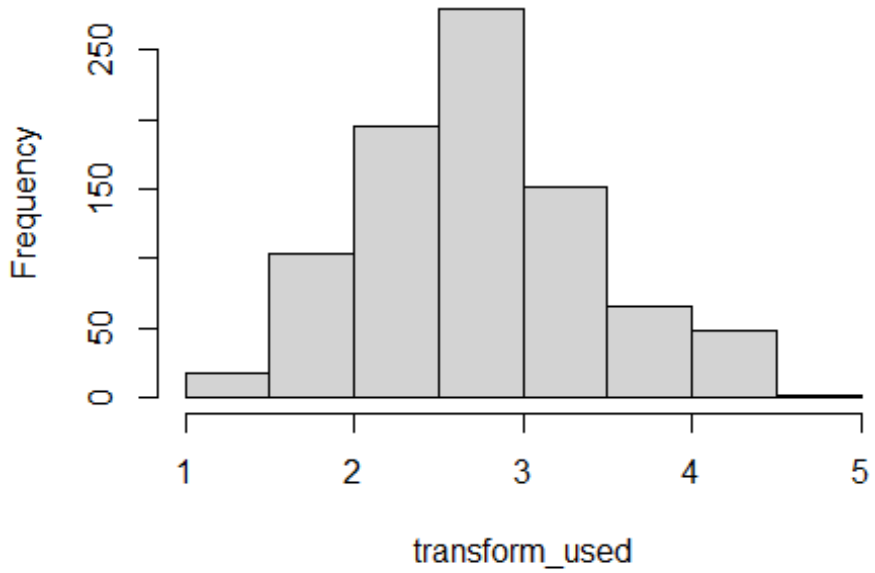


Figure 15 . Histogram of transformed data




```
## [[1]]
##
##  Shapiro-Wilk normality test
##
## data:  transform_used
## W = 0.98747, p-value = 9.667e-07
##
##
```

```
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  transform_used
## A = 2.8407, p-value = 3.784e-07
```

- The results state that :
  - The points at the start and end of the QQ plot seem to deviate a lot from the line. But the data in the middle fits in a good way on the line.
  - From the histogram, we can see that the curve for the data is a bit right skewed. The data on the right hand side of the plot has lesser values as compared to the values on the left hand side of the plot.
  - In the Shapiro-Wilk Test, the value of p is 9.667e-07, which is lesser than 0.05, we can say that we reject null hypothesis of normal distribution.
  - In the Anderson-Darling Test, the p-value is 3.784e-07 which is lesser than 0.05, we can reject the null hypothesis and conclude that we have sufficient evidence to say this data does not follow a normal distribution.

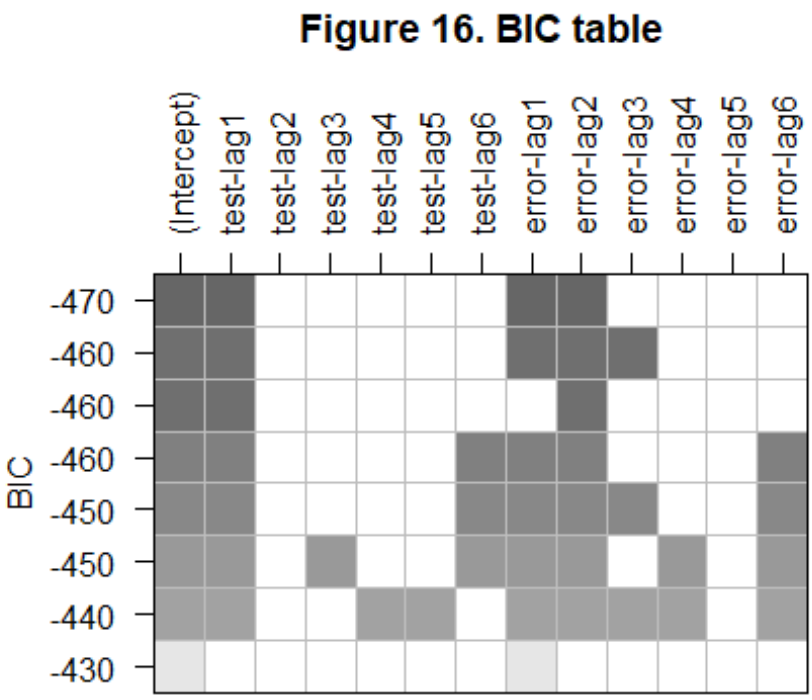
We take the second differentiation as the base data as it is close to noramlly distributed.

Model Specification

Now, we find the possible models using BIC table, ACF plot and PACF plot.

BIC table

```
#Plot BIC Table for model specification
plot(armasubsets(y=us_unemp_tsa_2diff, nar=6, nma=6, y.name='test', ar.method='ols'))
title(main = 'Figure 16. BIC table', line= 6)
```



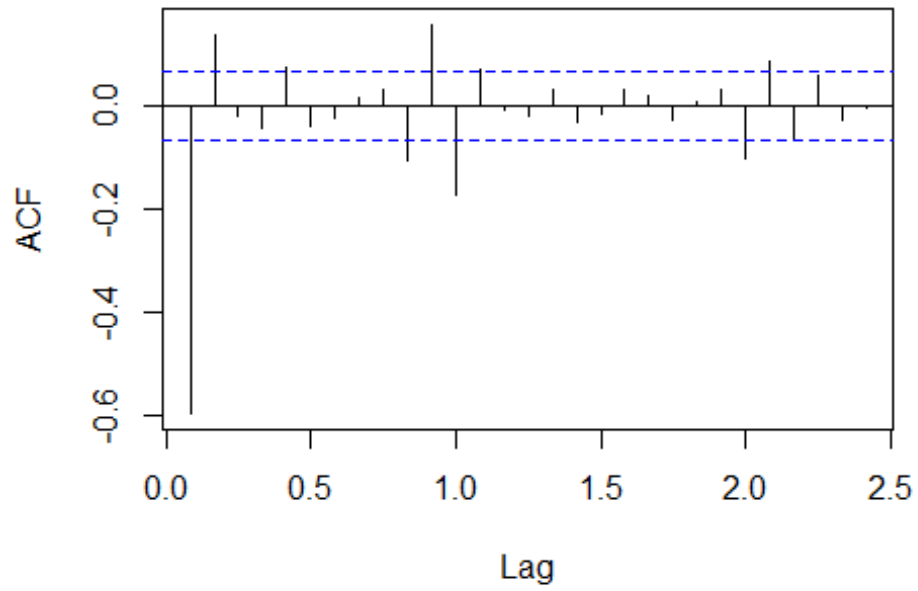
From the BIC table, we get the values p = 1 and q = 1,2,3.

Hence we can say that ARIMA(1,2,1), ARIMA(1,2,2), ARIMA(1,2,3) could be a potential model form BIC table.

ACF and PACF Plots

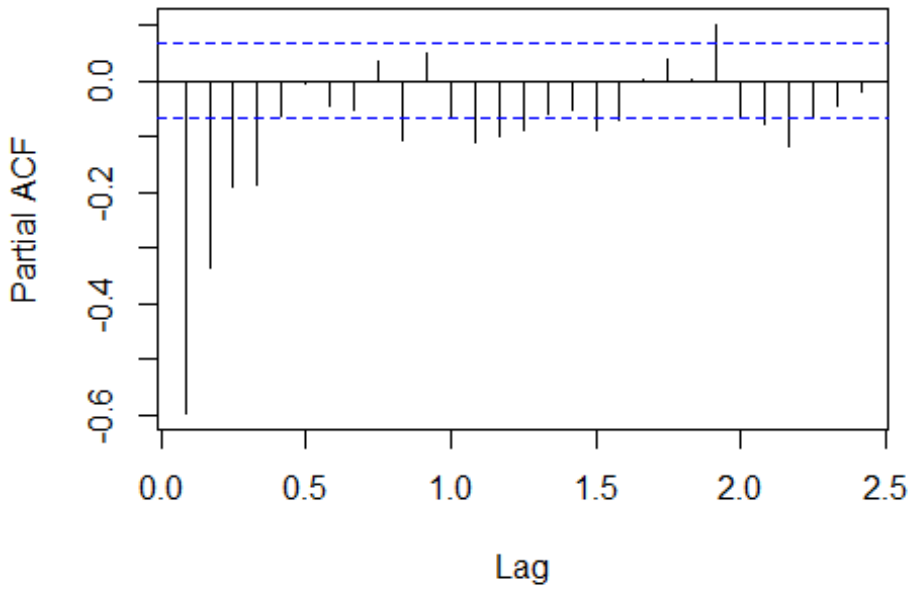
```
acf_diffrenced=acf(us_unemp_tsa_2diff,plot=FALSE)
plot(acf_diffrenced, main = "Figure 17. US unemployment Time Series diffrenced data ACF")
```

jure 17. US unemployment Time Series diffrenced da




```
pacf_diffrenced=pacf(us_unemp_tsa_2diff,plot=FALSE)
plot(pacf_diffrenced, main = "Figure 18. US unemployment Time Series diffrenced data PACF")
```

ure 18. US unemployment Time Series diffrenced dat



From the ACFand PACF, we get the values p = 1 and q = 1.

Hence possible ARIMA models are ARIMA(1,2,1) from these plots.

EACF Table

```
eacf(us_unemp_tsa_2diff)
```

```
## AR/MA
##   0 1 2 3 4 5 6 7 8 9 10 11 12 13
## 0 x x o o x o o o o x x  x  x  o
## 1 x o o o x o x o o o o  x  o  o
## 2 x x x x o o o o o o o  x  o  o
## 3 x x x x x o o o o o o  x  o  o
## 4 x x x x x o o o o o o  x  o  x
## 5 o x x x o x o o o o o  x  o  o
## 6 o x x x x x o o o o o  x  o  x
## 7 x x x o o o x o o o o  x  o  x
```

Hence, from the EACF we get the values possible ARIMA models are ARIMA(1,2,1), ARIMA(0,2,2)and ARIMA(1,2,2)

- Model Specification gave us the following possible models.
    1. ARIMA(1,2,1)
    2. ARIMA(1,2,2)
    3. ARIMA(1,2,3)
    4. ARIMA(0,2,2)

Paramter Estimation

Creating function to estimate the ARIMA() models and comparing their AIC and BIC

```r
#Function to estimate the ARIMA() models and comparing their AIC and BIC
model_estimation <- function(p, q, mtd){
  to_be_estimated <- stats::arima(us_unemp_tsa_2diff, order = c(p,0,q), method=mtd)
  warning('2nd Order Differenced data is already loaded in this function\n')
  return(to_be_estimated)
}

#Airma(1,2,1) CSS model estimation
arima121_CSS <- model_estimation(1,1,'CSS')
arima122_CSS <- model_estimation(1,2,'CSS')
arima123_CSS <- model_estimation(1,3,'CSS')
arima022_CSS <- model_estimation(0,2,'CSS')

arima121_ML <- model_estimation(1,1,'ML')
arima122_ML <- model_estimation(1,2,'ML')
arima123_ML <- model_estimation(1,3,'ML')
arima022_ML <- model_estimation(0,2,'ML')

aic_ML <- AIC(arima121_ML,
arima122_ML,
arima123_ML,
arima022_ML)

bic_ML <- BIC(arima121_ML,
arima122_ML,
arima123_ML,
arima022_ML)

sort.score <- function(x, score = c("bic", "aic")){
if (score == "aic"){
x[with(x, order(AIC)),]
} else if (score == "bic") {
x[with(x, order(BIC)),]
} else {
warning('score = "x" only accepts valid arguments ("aic","bic")')
}
}

sort.score(aic_ML, score = "aic")
```

```
##              df       AIC
## arima123_ML  6 -362.3499
## arima121_ML  4 -315.2847
## arima122_ML  5 -313.3544
## arima022_ML  4 -313.2774
```

```r
sort.score(bic_ML, score = "bic")
```

```
##              df       BIC
## arima123_ML  6 -333.7944
## arima121_ML  4 -296.2477
## arima022_ML  4 -294.2404
## arima122_ML  5 -289.5581
```

We have to select the best model for to fit.For this, we have calculated the AIC and BIC values of the models using the Maximum Likelihood method and sorted the models according to their AIC and BIC

values. We have used the sort.score() function from the lecture slides for the same purpose. The model having the smallest value for AIC and BIC is desirable. According to the AIC and BIC values, the model ARIMA(1,2,3) is the best. However, all the models have to be tested for their residuals.

Model Diagnostics

We have written a function for performing the diagnostic checking of the proposed models. The function includes the z test of model coefficients and the Anderson Darling Normality test of the model parameters. The function also includes the Ljung Box test, Box Pierce test, time series plot, histogram, QQ Plot and autocorrelation function of the residuals. The maximum likelihood (ML) and conditional sum of squares (CSS) methods were used to build the models.

```
diagnostic_checking <- function(model_used,fig) {

model_Res = rstandard(model_used)
plot(model_Res, xlab='Time',
ylab='Standardized Residuals',type='l', main = paste("Figure ",fig,". Time series plot
of standardised residuals. "))

hist(model_Res, ylab='Standardized Residuals',
main = paste("Figure ",fig+1,". Histogram of standardised residuals."))

qqnorm(model_Res, main = paste("Figure ",fig+2,". QQ plot of standardised residuals.")
)
qqline(model_Res, col = 2, lwd = 1, lty = 2)

ct = coeftest(model_used)

ad = ad.test(model_Res)

acf(model_Res, main = paste("Figure ",fig+3,". ACF plot of standardised residuals."))

lj = Box.test(model_Res, type = "Ljung-Box")

bp = Box.test(model_Res, type = "Box-Pierce")

tsdiag(model_used, gof=15, omit.initial = F)

testlist <- list(ct, ad, lj, bp)

return(testlist)
}
```

The residual analysis of the four ARIMA models is as follows:

```
diagnostic_checking(arima123_ML,19)
```

Figure 19 . Time series plot of standardised residua

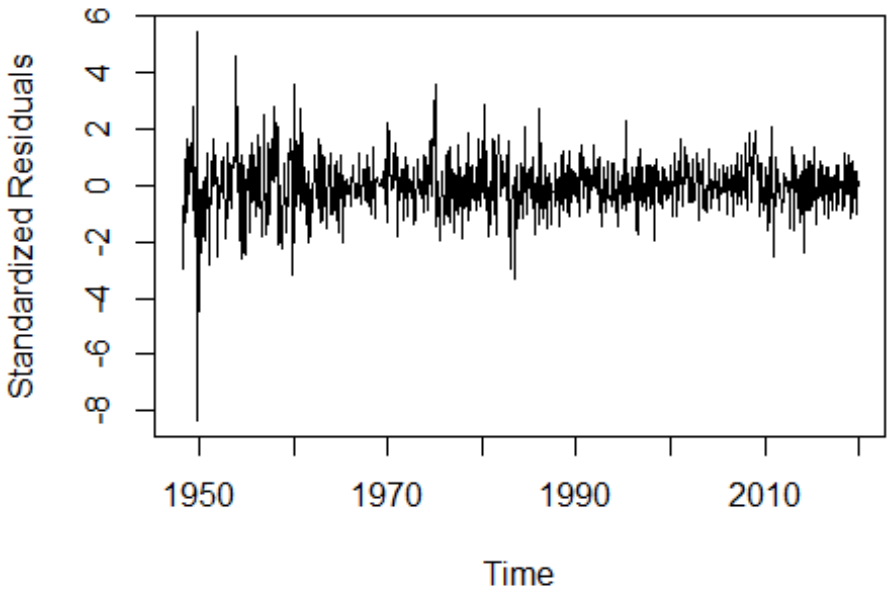


Figure 20 . Histogram of standardised residuals.

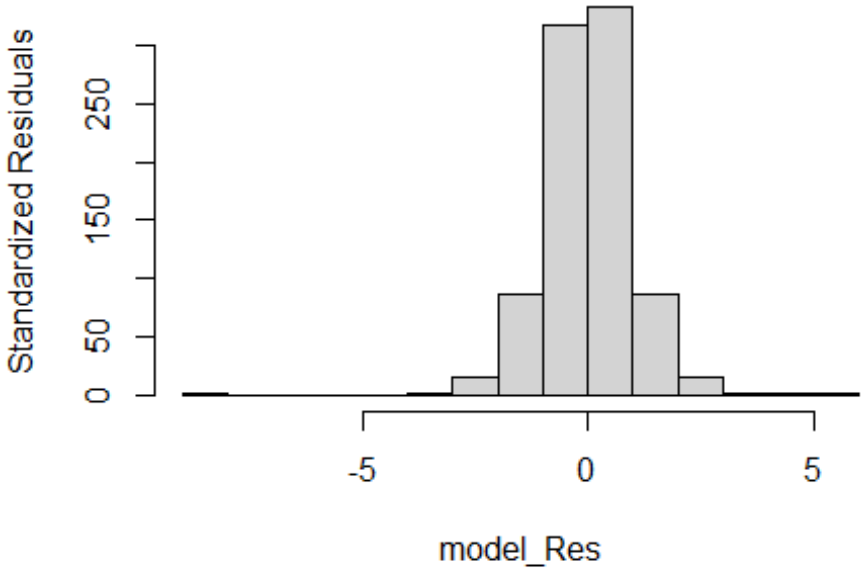


Figure 21 . QQ plot of standardised residuals.

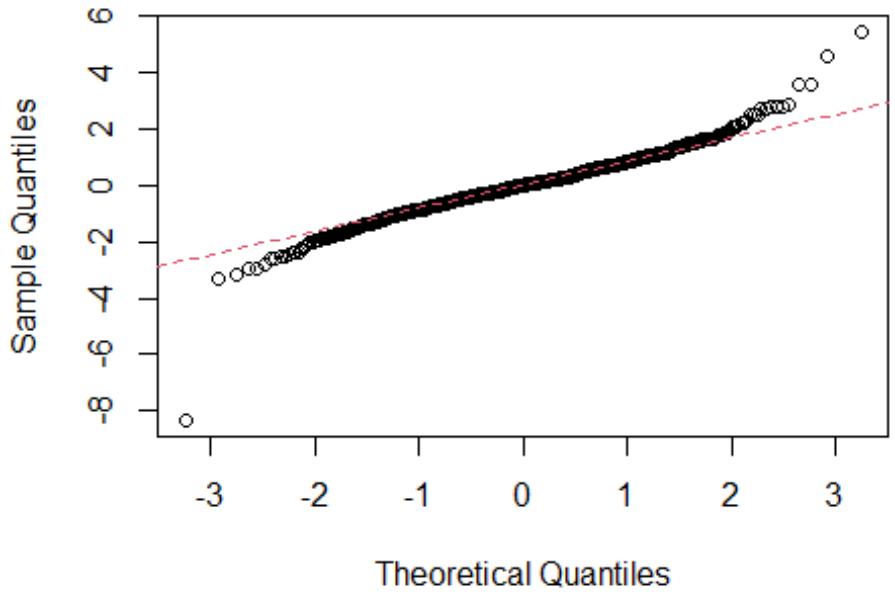
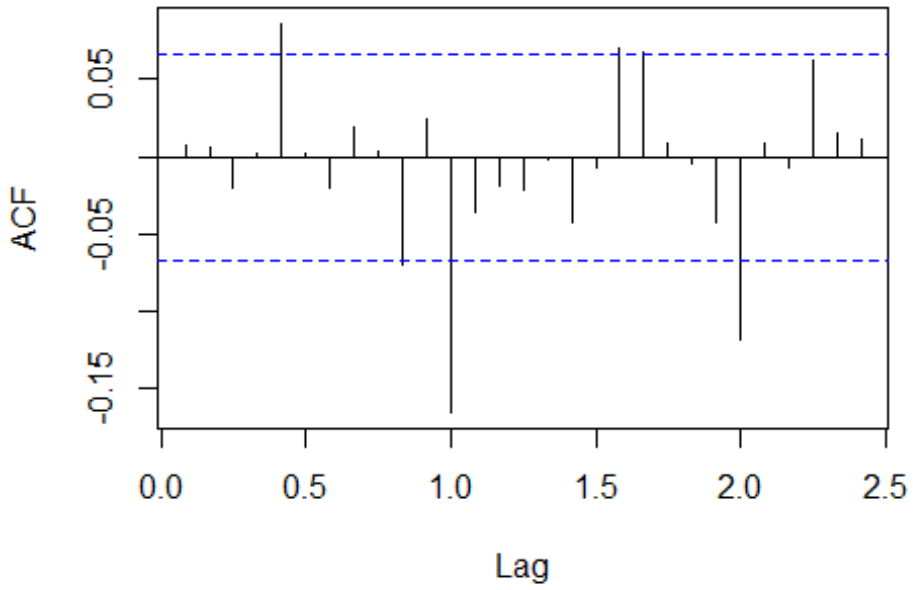
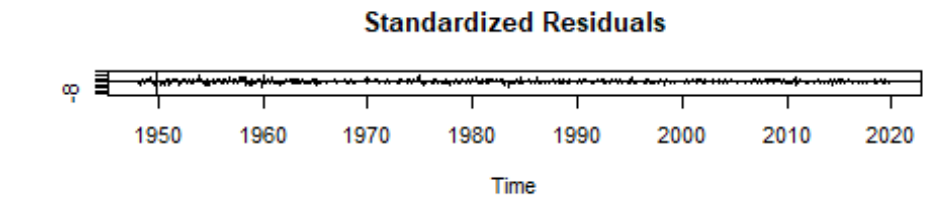
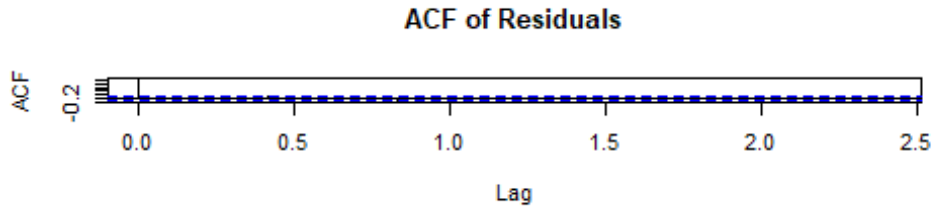
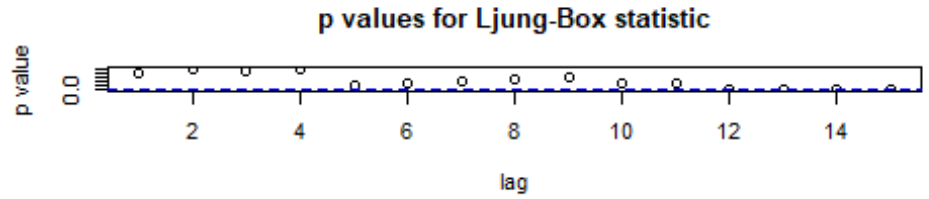


Figure 22 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1        8.1872e-01  3.6586e-02  22.3779 < 2.2e-16 ***
## ma1       -1.8129e+00  4.8083e-02 -37.7039 < 2.2e-16 ***
## ma2        1.0322e+00  6.8616e-02  15.0431 < 2.2e-16 ***
## ma3       -2.1927e-01  3.3613e-02  -6.5234 6.875e-11 ***
## intercept -4.0868e-05  6.7947e-05  -0.6015    0.5475
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.0027, p-value = 2.238e-12
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.044025, df = 1, p-value = 0.8338
##
##
## [[4]]
```

```
##
##  Box-Pierce test
##
## data:  model_Res
## X-squared = 0.043872, df = 1, p-value = 0.8341
```

```
diagnostic_checking(arima123_CSS,23)
```

Figure 23 . Time series plot of standardised residua

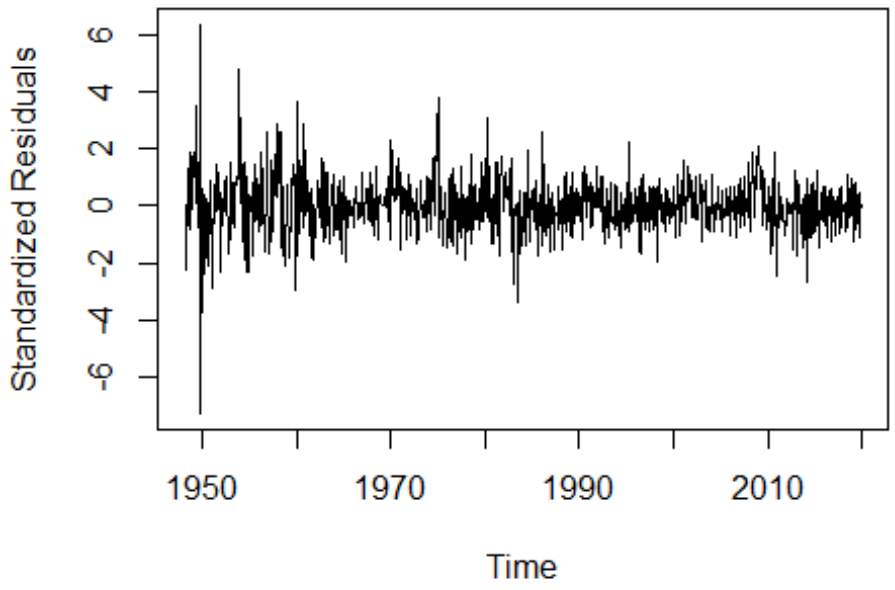


Figure 24 . Histogram of standardised residuals.

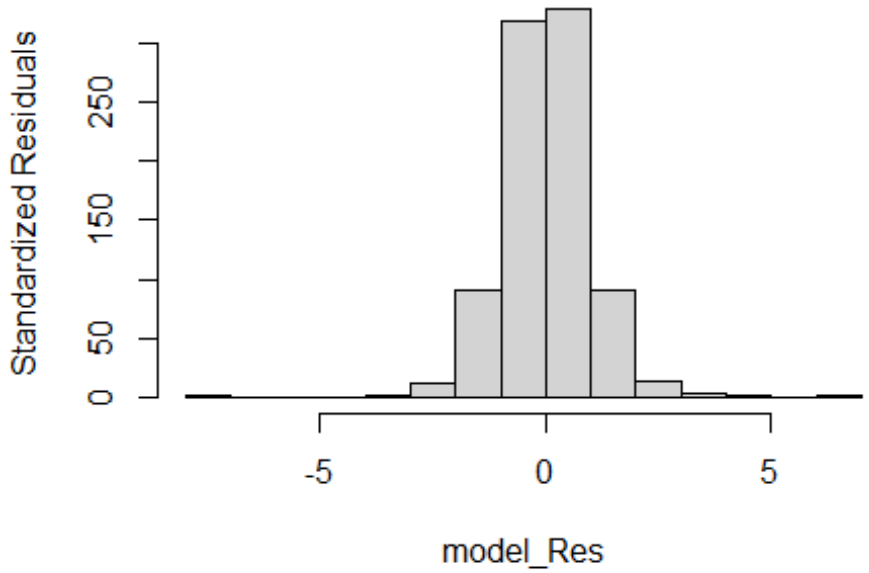


Figure 25 . QQ plot of standardised residuals.

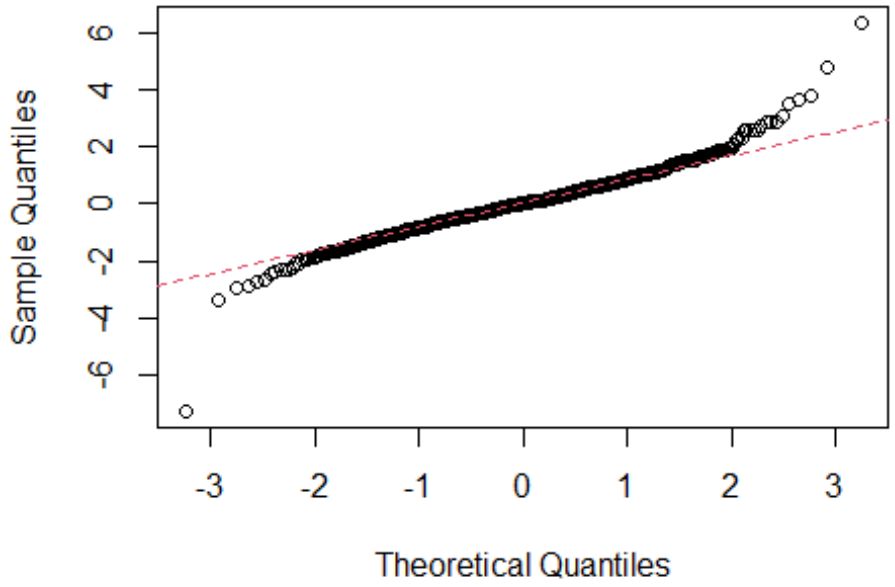
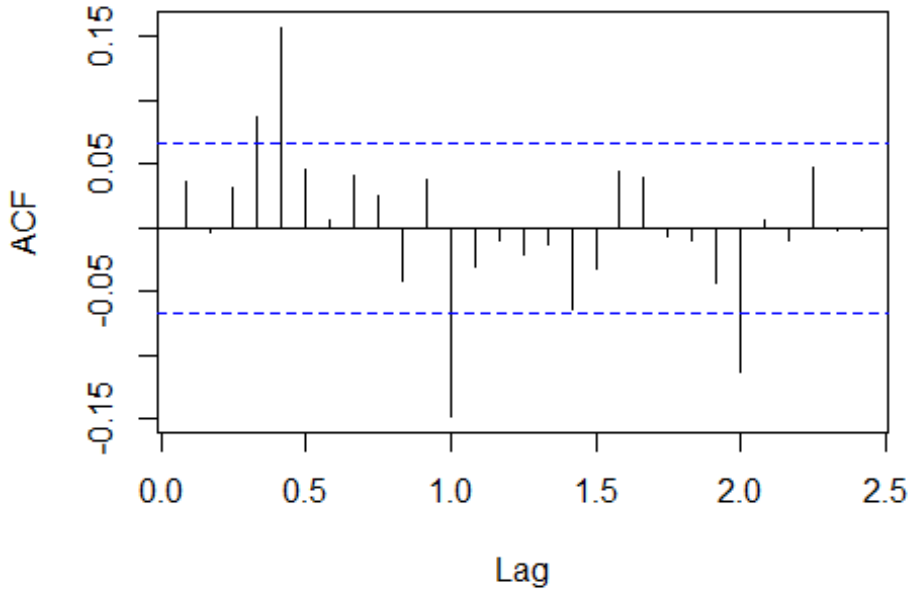
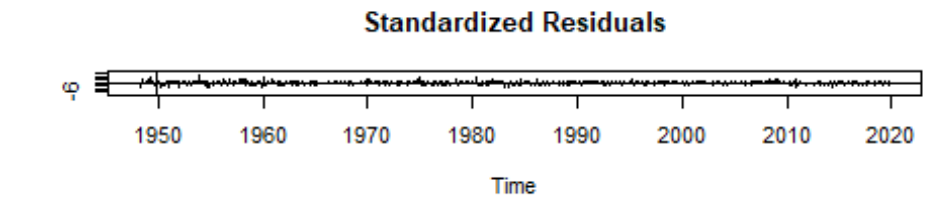
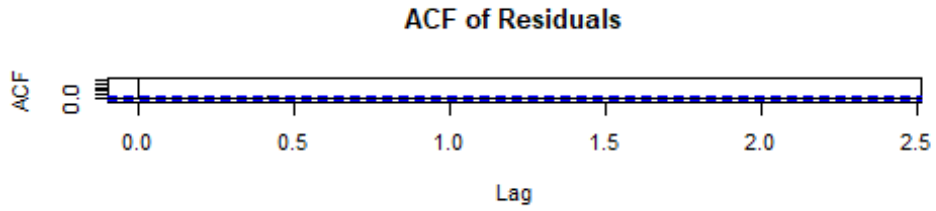
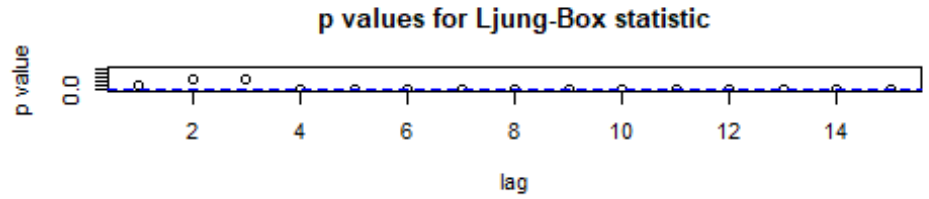


Figure 26 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1        5.3950e-01  4.8776e-02  11.0609 < 2.2e-16 ***
## ma1       -1.5434e+00  6.0172e-02 -25.6497 < 2.2e-16 ***
## ma2        7.9287e-01  7.8055e-02  10.1578 < 2.2e-16 ***
## ma3       -2.4803e-01  3.0932e-02  -8.0185 1.071e-15 ***
## intercept  2.4751e-05  6.7205e-05   0.3683    0.7127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 4.717, p-value = 1.088e-11
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 1.1848, df = 1, p-value = 0.2764
##
##
## [[4]]
```

```
##
##  Box-Pierce test
##
## data:  model_Res
## X-squared = 1.1807, df = 1, p-value = 0.2772
```

1. ARIMA(1,2,3) :- The model was built using ML and CSS methods. Both models were tested for the diagnostic_checking() function. The QQ plot and the histogram appears normally distributed but the Anderson Darling Normality test suggests that the plot is not normally distributed. Hence, we cannot use the model built using the ML method as the ML method assumes normality in data. When we analyse the model built using the CSS method, the AR and MA coefficients are significant. However, the resulting Ljung Box test has most of the lags less than 10 below the zero line. Hence, we cannot assume the model to be a good fit. We have rejected this model.

```
diagnostic_checking(arima121_ML,27)
```

Figure 27 . Time series plot of standardised residua

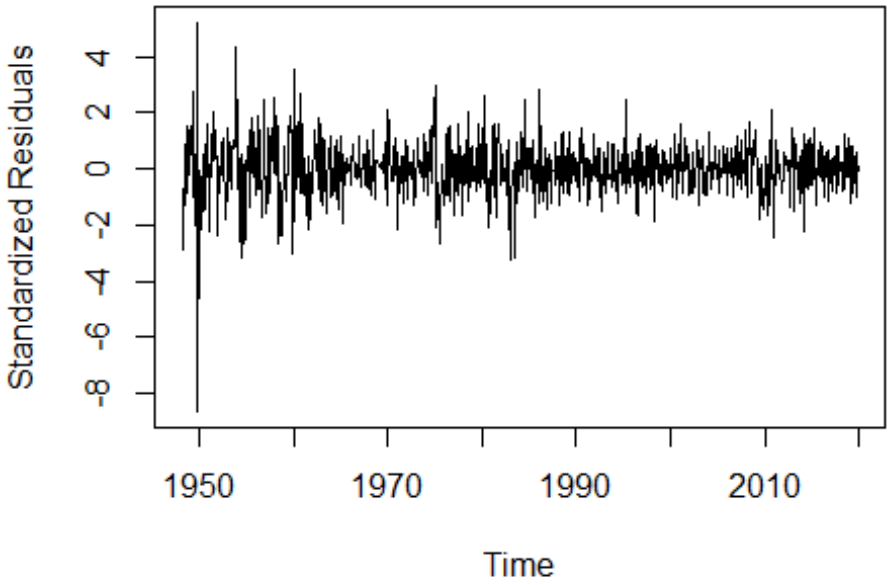


Figure 28 . Histogram of standardised residuals.

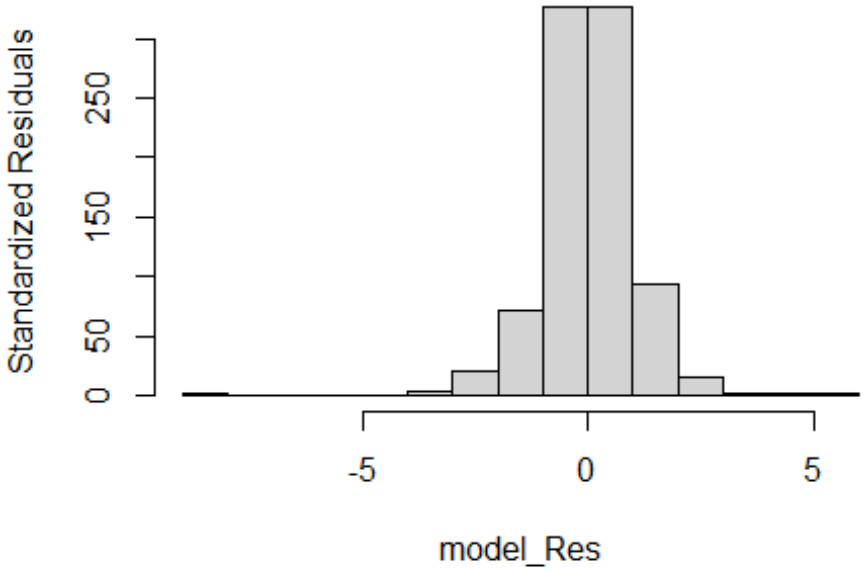


Figure 29 . QQ plot of standardised residuals.

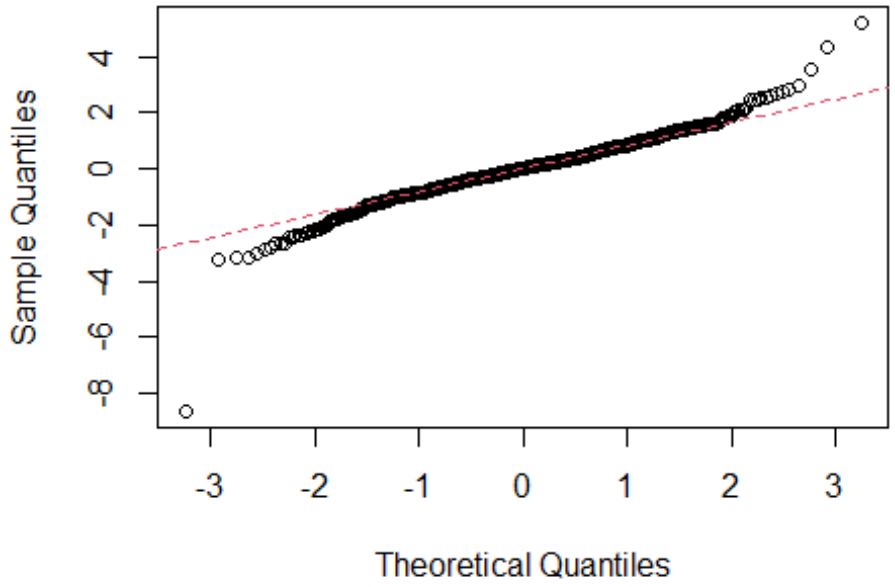
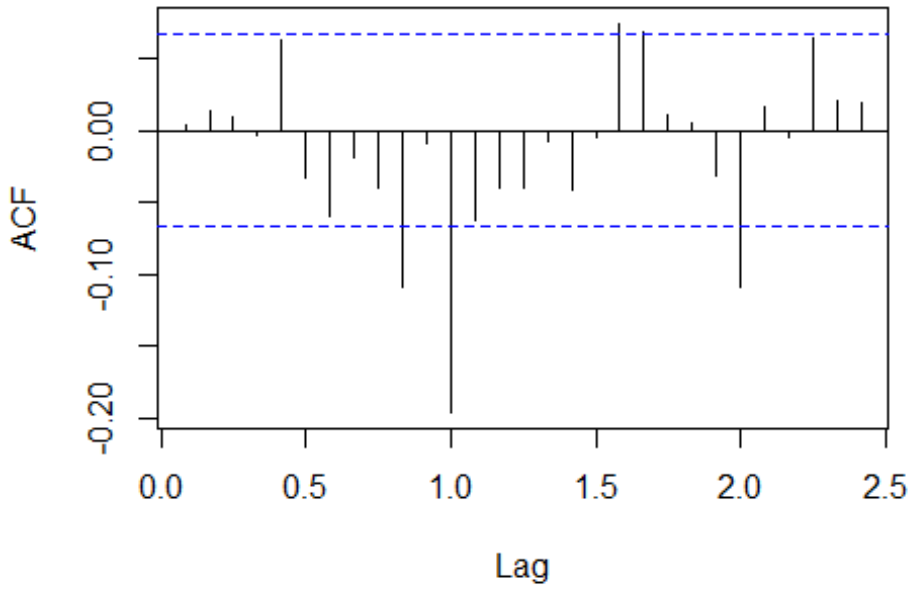
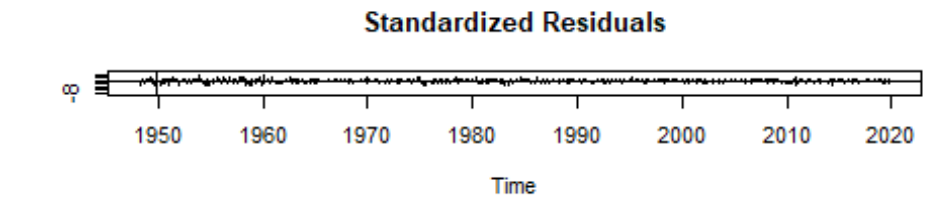
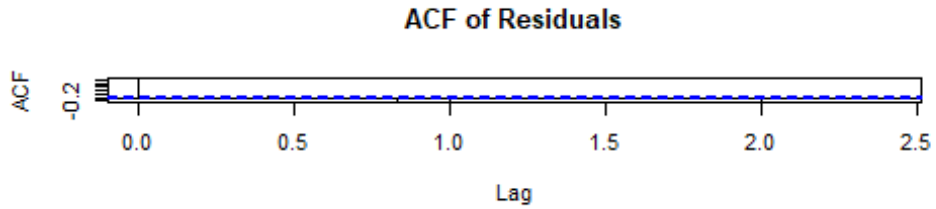
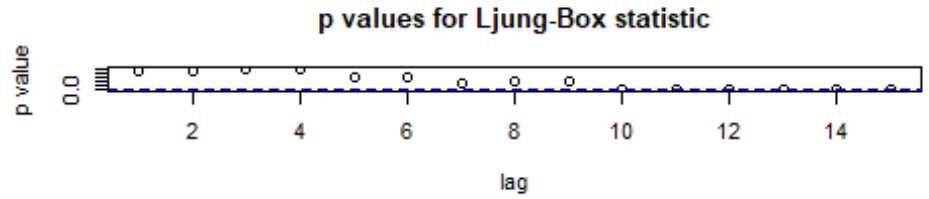


Figure 30 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1       -0.22880629  0.04436621  -5.1572 2.506e-07 ***
## ma1       -0.69683452  0.03603558 -19.3374 < 2.2e-16 ***
## intercept -0.00016266  0.00168991  -0.0963    0.9233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.1028, p-value = 1.288e-12
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.014811, df = 1, p-value = 0.9031
##
##
## [[4]]
##
##  Box-Pierce test
```

```
##
## data:  model_Res
## X-squared = 0.01476, df = 1, p-value = 0.9033

diagnostic_checking(arima121_CSS,31)
```

Figure 31 . Time series plot of standardised residua

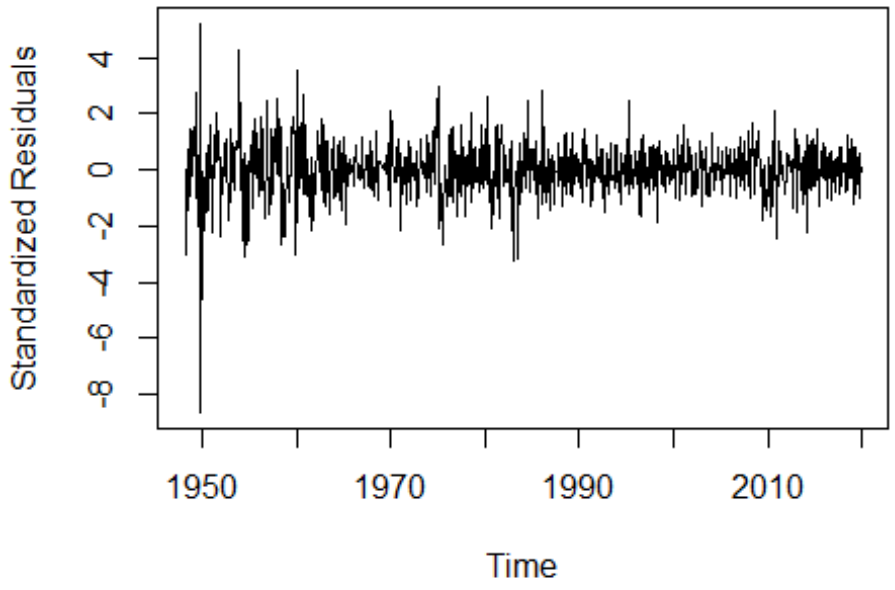


Figure 32 . Histogram of standardised residuals

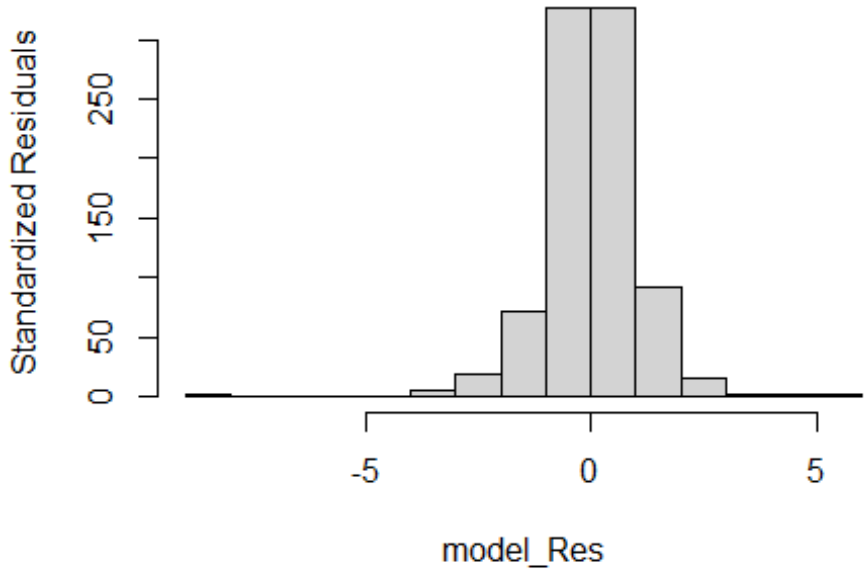


Figure 33 . QQ plot of standardised residuals.

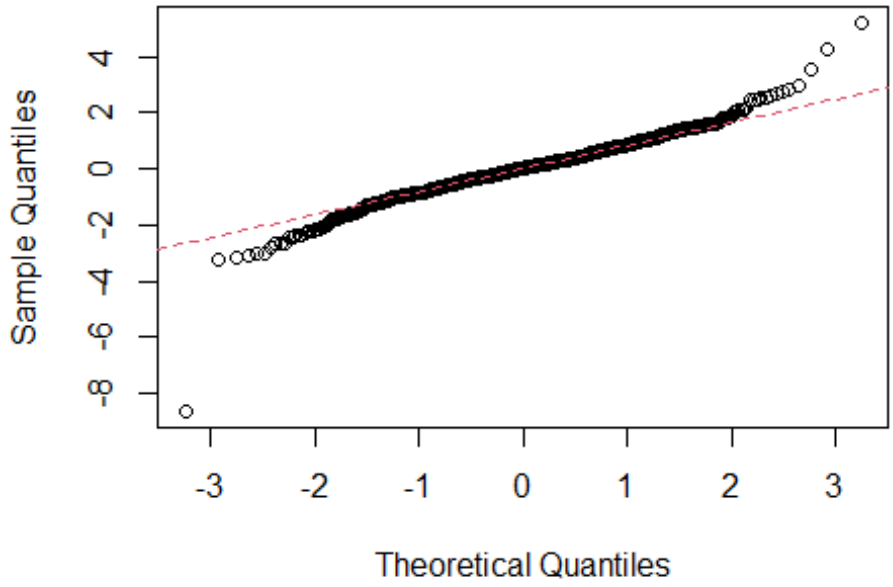
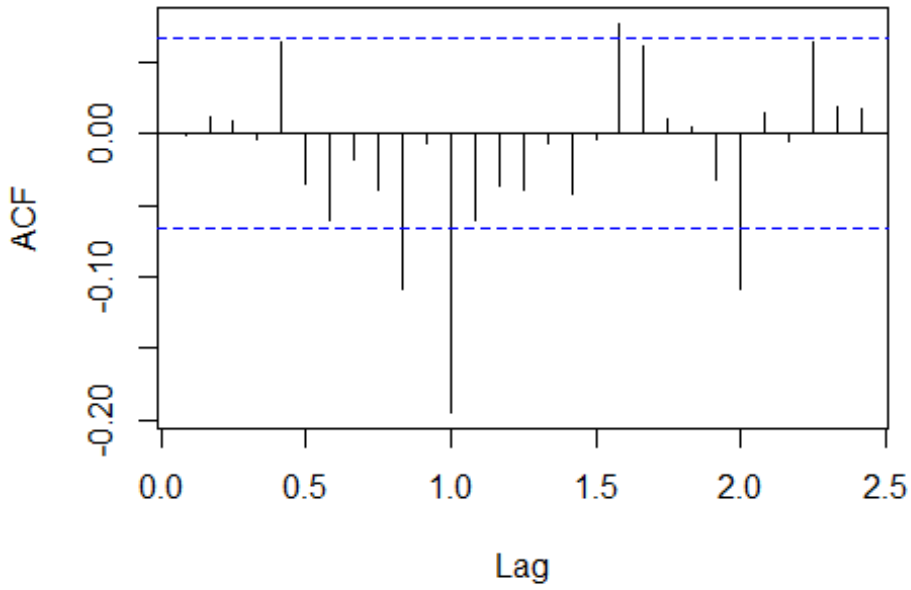
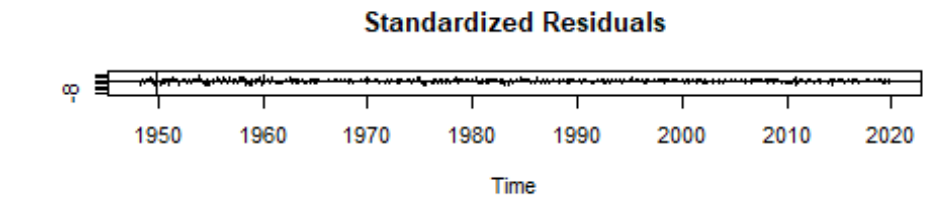
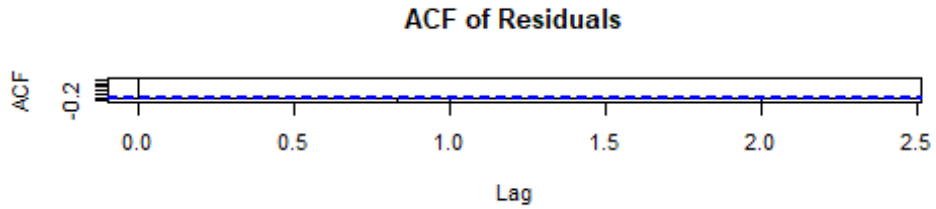
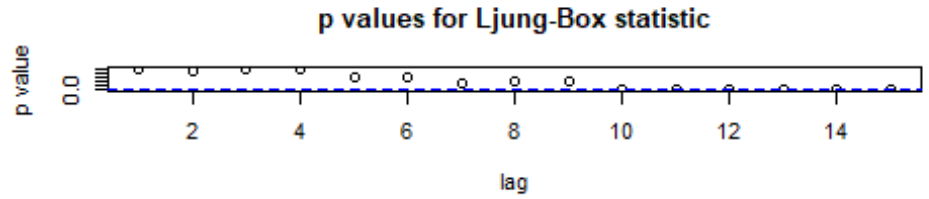


Figure 34 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ar1       -0.22583898  0.04458484  -5.0654 4.076e-07 ***
## ma1       -0.69599554  0.03631442 -19.1658 < 2.2e-16 ***
## intercept -0.00017178  0.00169779  -0.1012    0.9194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.1743, p-value = 8.675e-13
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.0014227, df = 1, p-value = 0.9699
##
##
## [[4]]
##
##  Box-Pierce test
```

```
##
## data:  model_Res
## X-squared = 0.0014177, df = 1, p-value = 0.97
```

2. ARIMA(1,2,2) :- The model was built using ML and CSS methods. Both models were tested for the diagnostic_checking() function. The QQ plot and the histogram appears normally distributed but the Anderson Darling Normality test suggests that the plot is not normally distributed. Hence, we cannot use the model built using the ML method as the ML method assumes normality in data. When we analyse the model built using the CSS method, we find that the AR1 and MA2 coefficients are not significant. Also, resulting Ljung Box test has most of the lags under 10 below the zero line. Hence, we cannot assume the model to be a good fit. We have rejected this model.

```
diagnostic_checking(arima022_ML,35)
```

Figure 35 . Time series plot of standardised residua

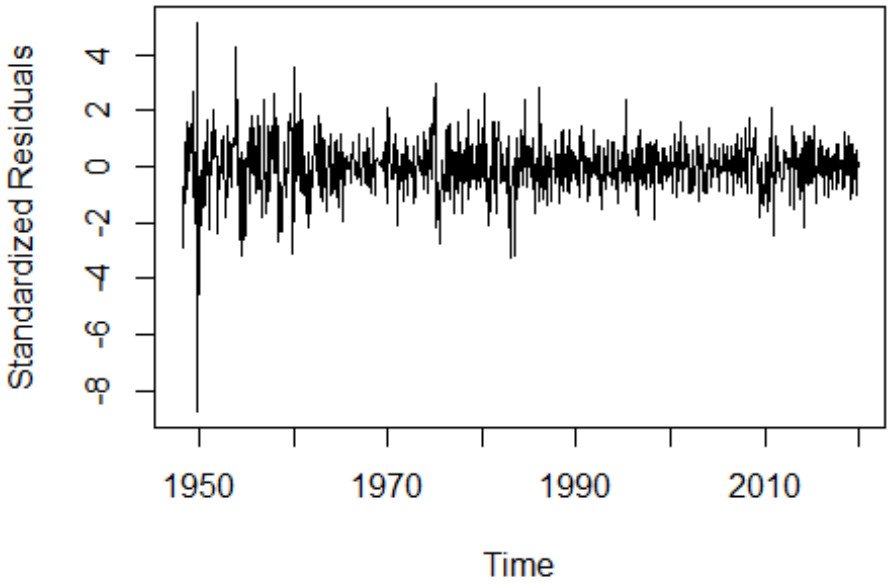


Figure 36 . Histogram of standardised residuals.

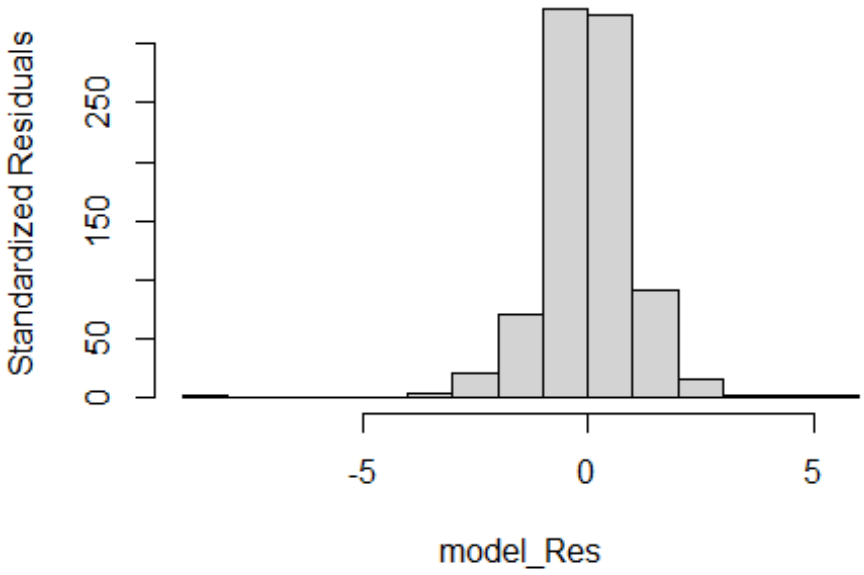


Figure 37 . QQ plot of standardised residuals.

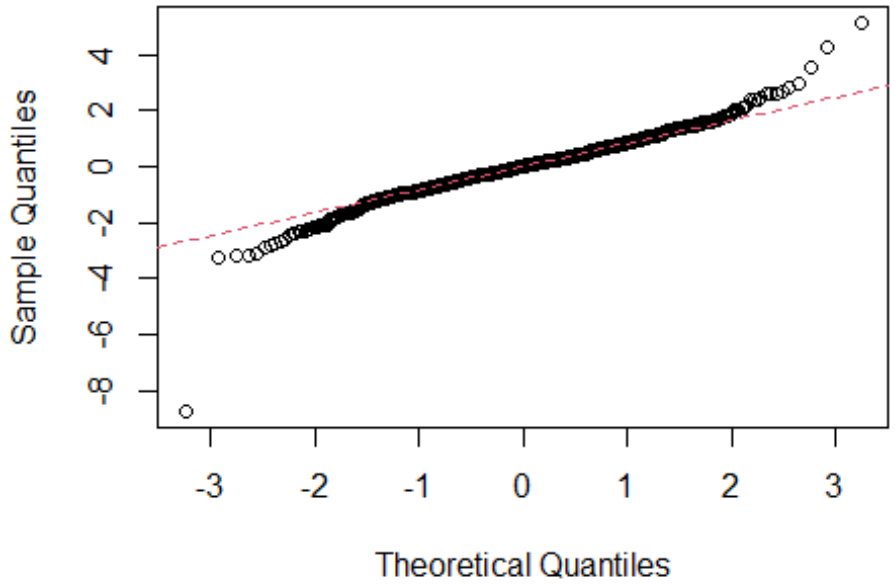
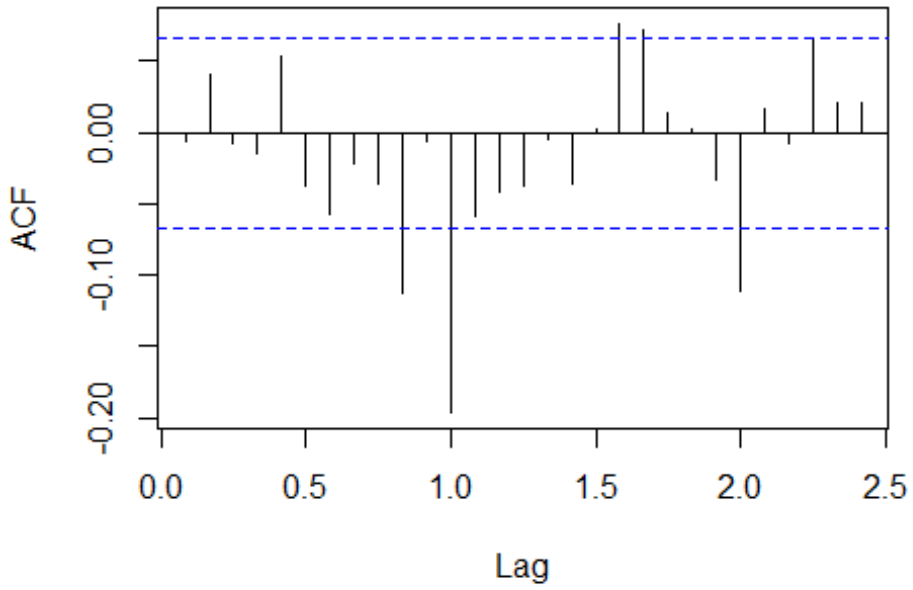
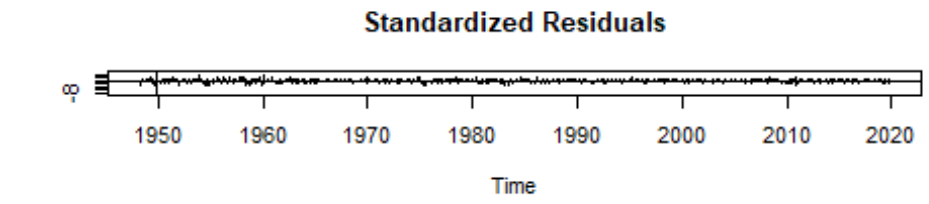
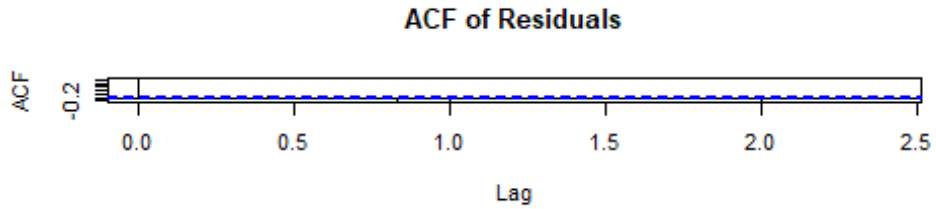
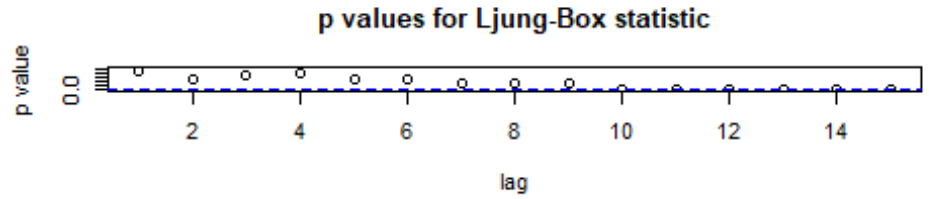


Figure 38 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ma1       -0.91491119  0.03247497 -28.1728 < 2.2e-16 ***
## ma2        0.17556547  0.03699057   4.7462 2.073e-06 ***
## intercept -0.00018709  0.00178683  -0.1047    0.9166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.086, p-value = 1.412e-12
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.033809, df = 1, p-value = 0.8541
##
##
## [[4]]
##
##  Box-Pierce test
```

```
##
## data:  model_Res
## X-squared = 0.033691, df = 1, p-value = 0.8544
```

```
diagnostic_checking(arima022_CSS,39)
```

Figure 39 . Time series plot of standardised residua

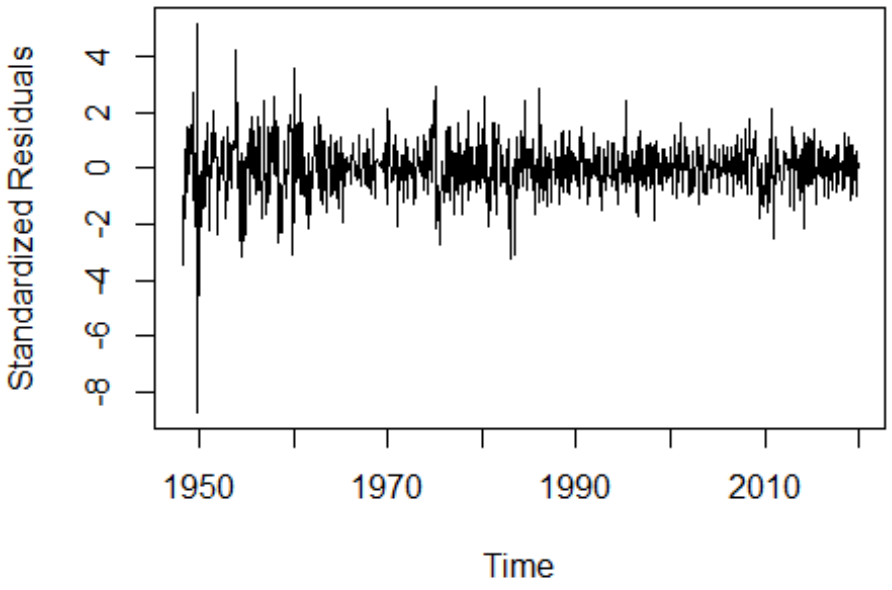


Figure 40 . Histogram of standardised residuals

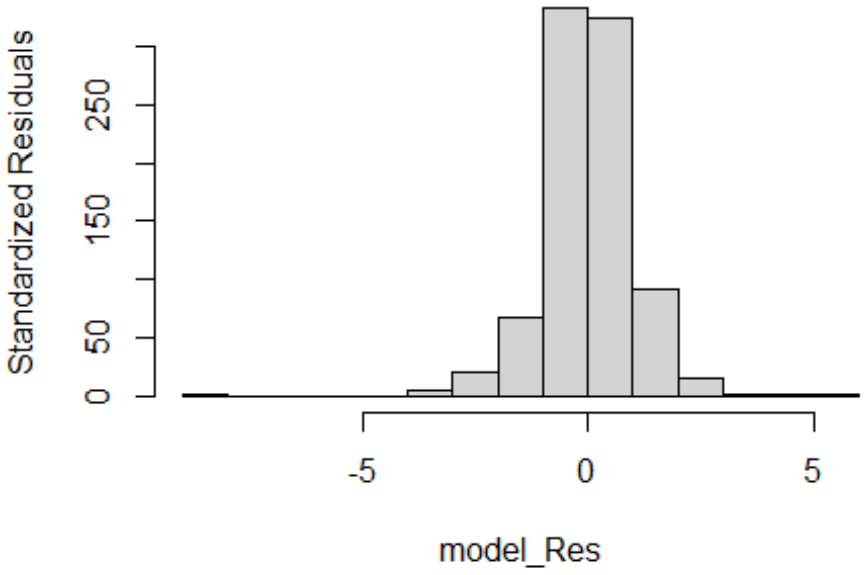


Figure 41 . QQ plot of standardised residuals.

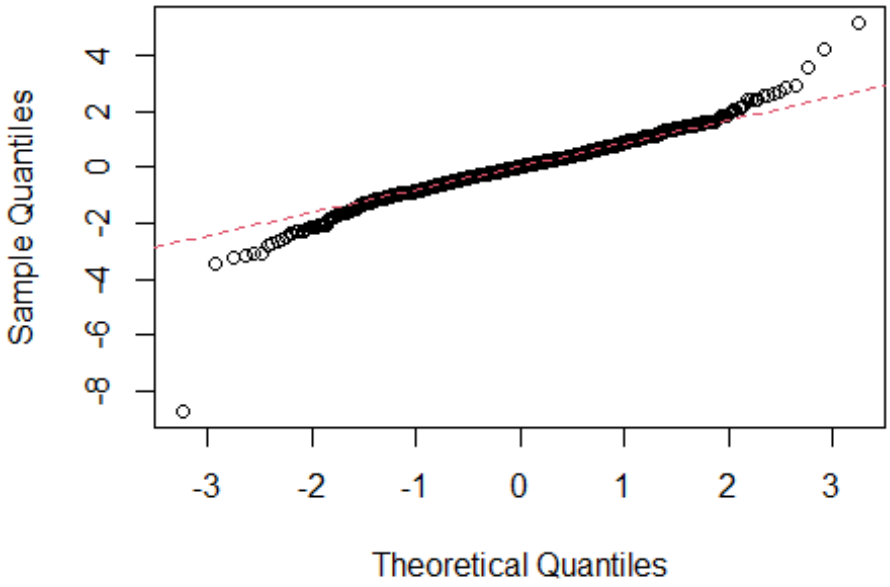
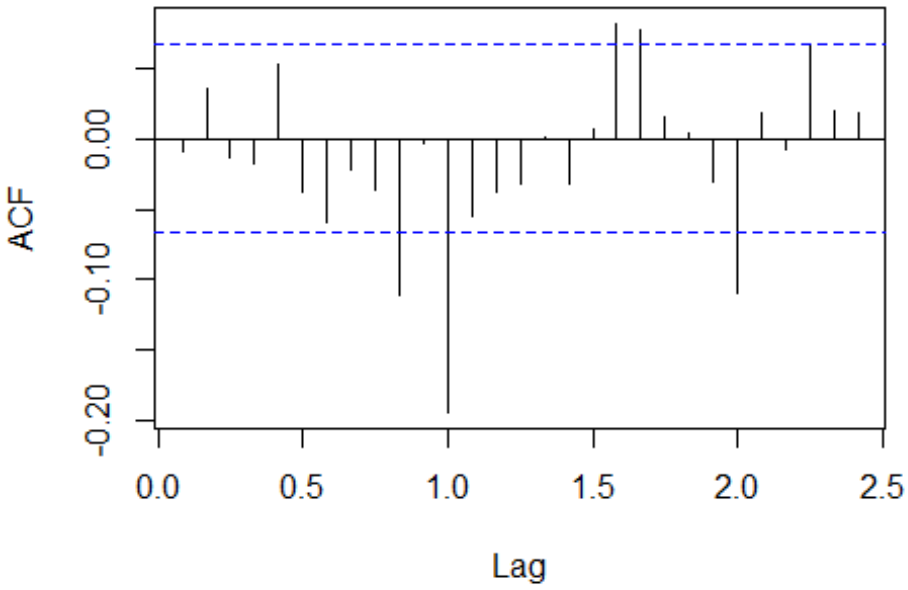
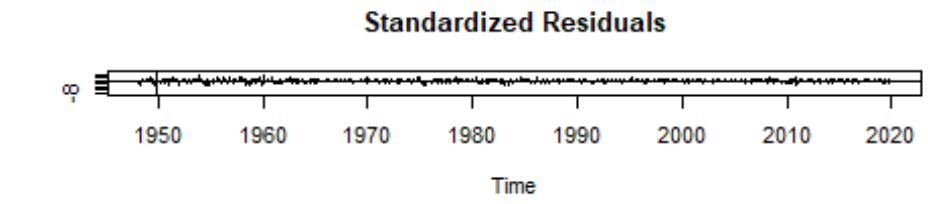
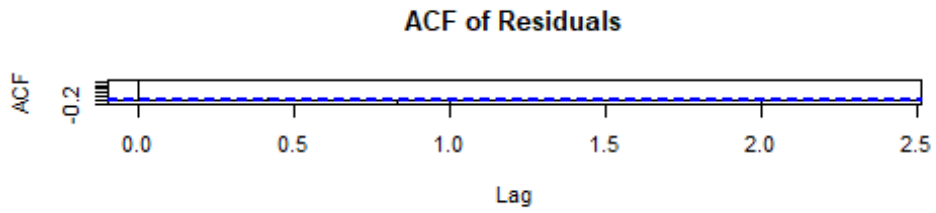
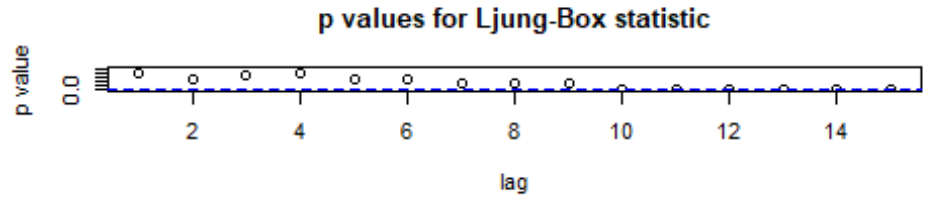


Figure 42 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error  z value  Pr(>|z|)
## ma1       -0.90634790  0.03232770 -28.0363 < 2.2e-16 ***
## ma2        0.17638048  0.03655084   4.8256 1.396e-06 ***
## intercept -0.00031336  0.00185648  -0.1688     0.866
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.1856, p-value = 8.148e-13
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.067804, df = 1, p-value = 0.7946
##
##
## [[4]]
##
##  Box-Pierce test
```

```
##
## data:  model_Res
## X-squared = 0.067568, df = 1, p-value = 0.7949
```

3. ARIMA(0,2,2) :- The model was built using ML and CSS methods. Both models were tested for the diagnostic_checking() function. The QQ plot and the histogram appears normally distributed but the Anderson Darling Normality test suggests that the plot is not normally distributed. Hence, we cannot use the model built using the ML method as the ML method assumes normality in data. When we analyse the model built using the CSS method, the AR and MA coefficients are significant. The Ljung Box test has all of the lags less than 10 above the zero line. Hence, we can assume the model to be a good fit.

```
diagnostic_checking(arima122_ML,43)
```

Figure 43 . Time series plot of standardised residua

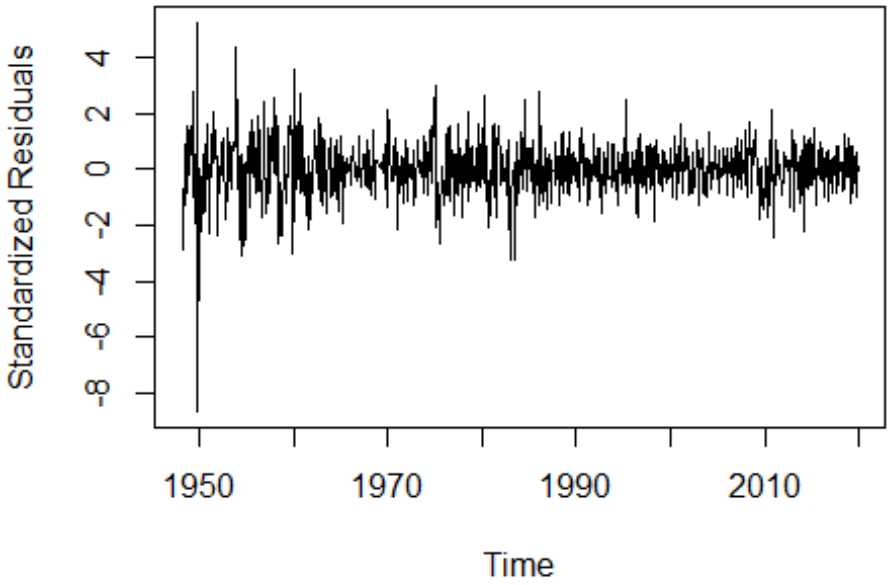


Figure 44 . Histogram of standardised residuals.

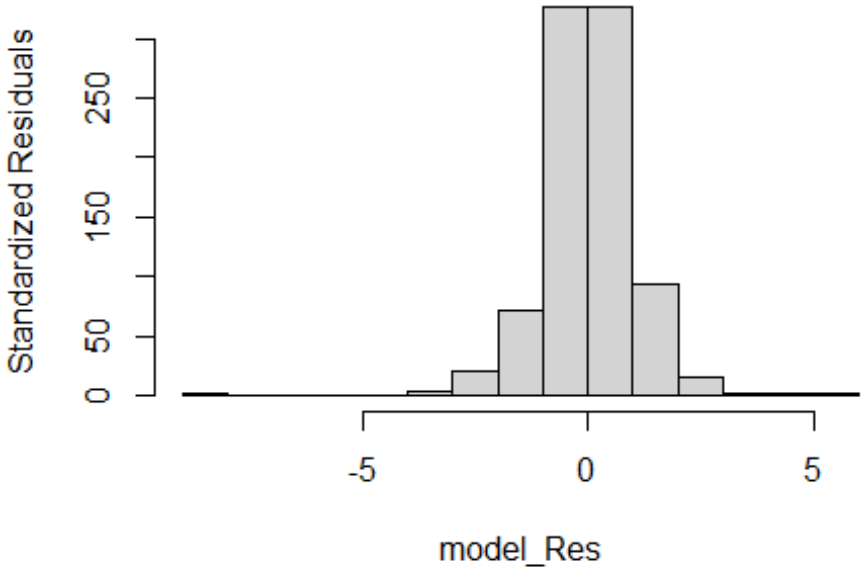


Figure 45 . QQ plot of standardised residuals.

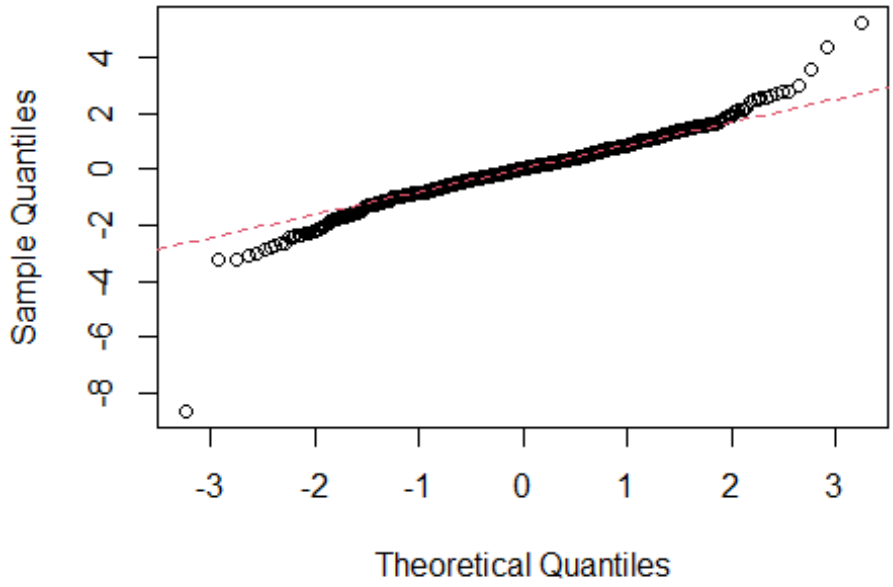
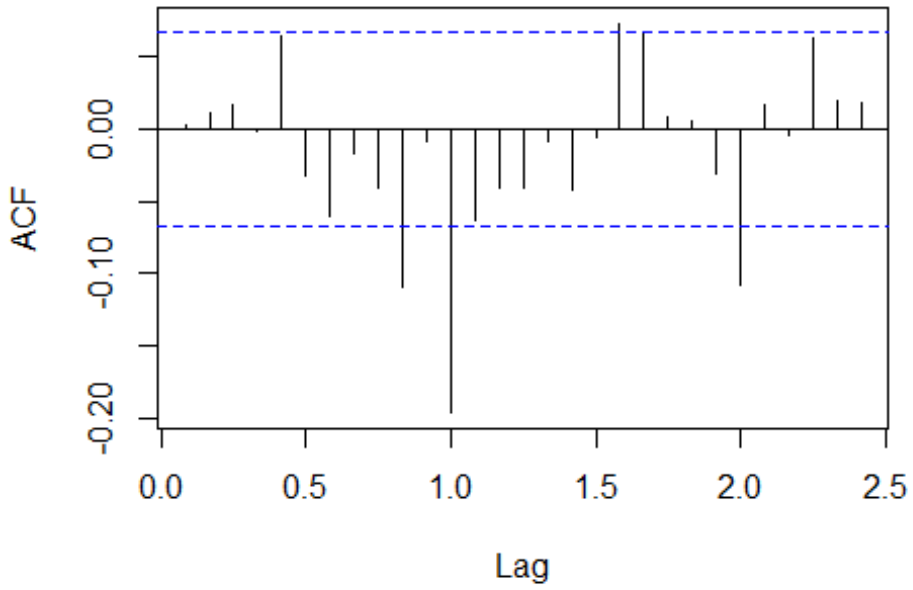
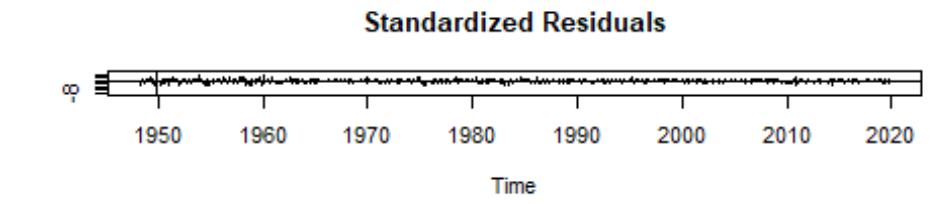
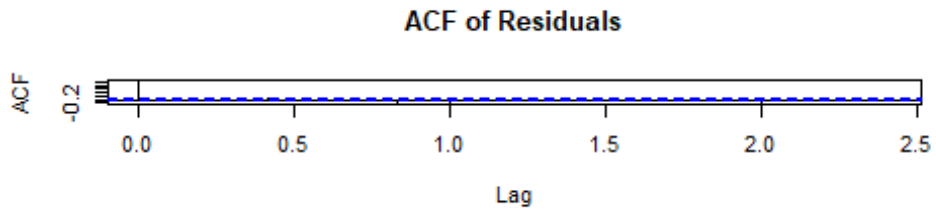
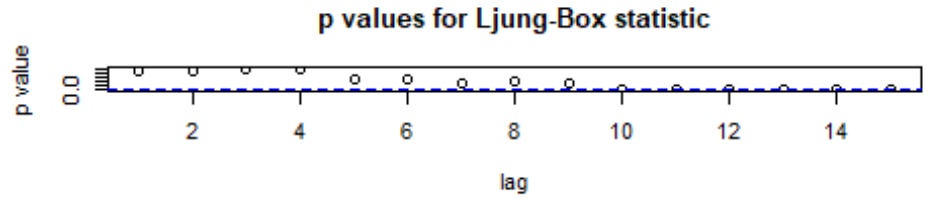


Figure 46 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error z value  Pr(>|z|)
## ar1       -0.27287497  0.17237427 -1.5830 0.1134129
## ma1       -0.65173374  0.17483393 -3.7277 0.0001932 ***
## ma2       -0.03920388  0.14956377 -0.2621 0.7932278
## intercept -0.00016115  0.00166325 -0.0969 0.9228169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.1189, p-value = 1.178e-12
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.0070617, df = 1, p-value = 0.933
##
##
## [[4]]
##
```

```
##  Box-Pierce test
##
## data:  model_Res
## X-squared = 0.0070372, df = 1, p-value = 0.9331
```

```
diagnostic_checking(arima122_CSS,47)
```

Figure 47 . Time series plot of standardised residua

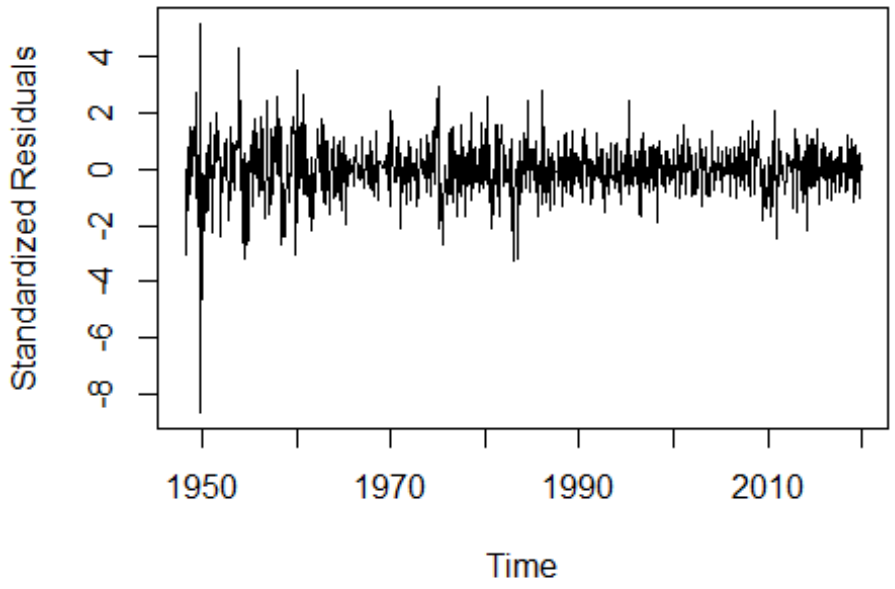


Figure 48 . Histogram of standardised residuals.

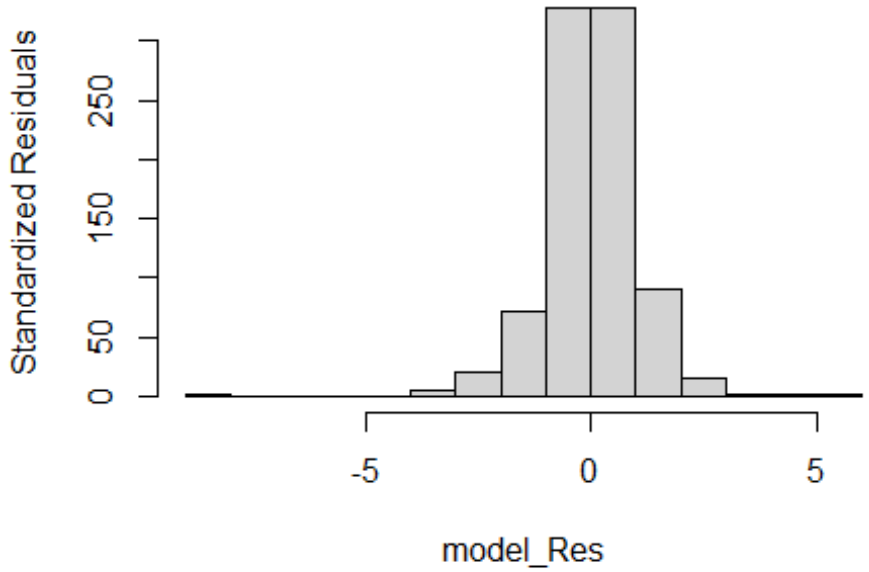


Figure 49 . QQ plot of standardised residuals.

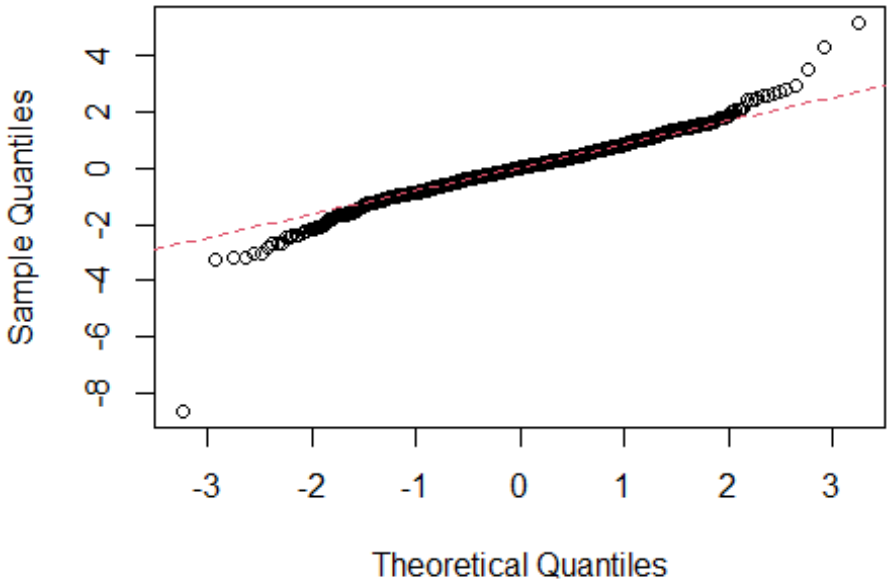
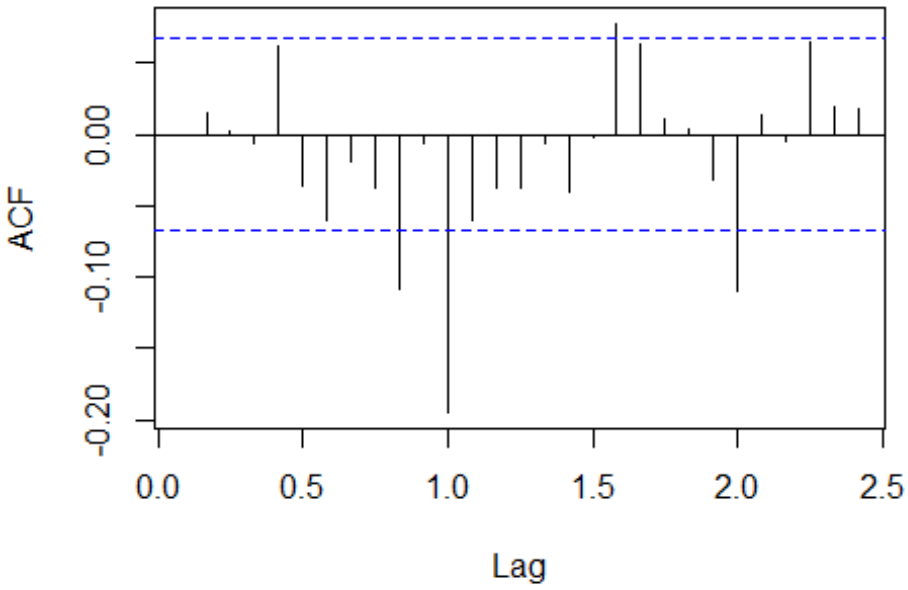
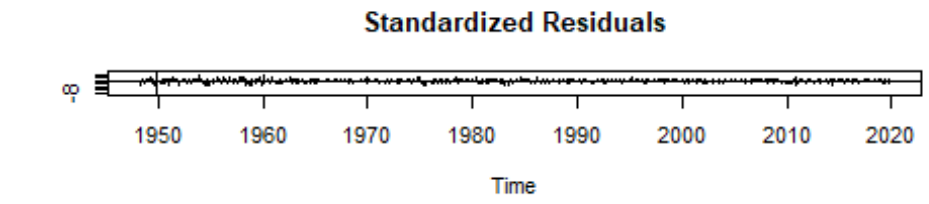
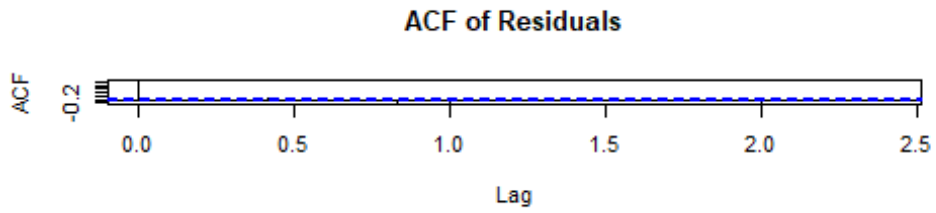
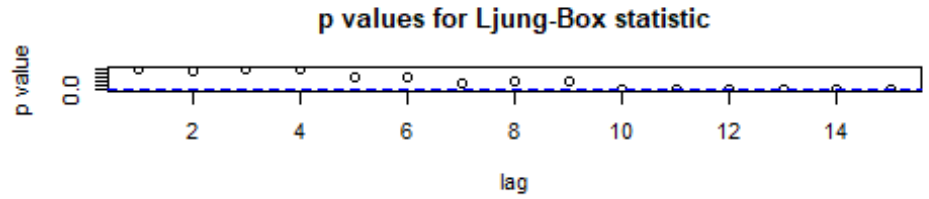


Figure 50 . ACF plot of standardised residuals.










```
## [[1]]
##
## z test of coefficients:
##
##              Estimate  Std. Error z value  Pr(>|z|)
## ar1       -0.17372045  0.20348244 -0.8537 0.3932508
## ma1       -0.74843760  0.20156196 -3.7132 0.0002047 ***
## ma2        0.04473986  0.16861653  0.2653 0.7907514
## intercept -0.00016421  0.00172839 -0.0950 0.9243073
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## [[2]]
##
##  Anderson-Darling normality test
##
## data:  model_Res
## A = 5.1444, p-value = 1.023e-12
##
##
## [[3]]
##
##  Box-Ljung test
##
## data:  model_Res
## X-squared = 0.00084751, df = 1, p-value = 0.9768
##
##
## [[4]]
##
```

```
##  Box-Pierce test
##
## data:  model_Res
## X-squared = 0.00084457, df = 1, p-value = 0.9768
```

4. ARIMA(1,2,1) :- The model was built using ML and CSS methods. Both models were tested for the diagnostic_checking() function. The QQ plot and the histogram appears normally distributed but the Anderson Darling Normality test suggests that the plot is not normally distributed. Hence, we cannot use the model built using the ML method as the ML method assumes normality in data. When we analyse the model built using the CSS method, the AR and MA coefficients are significant. The Ljung Box test has all of the lags less than 10 above the zero line.

We have found the models ARIMA(1,2,1) and ARIMA(0,2,2) to be favourable models after residual analysis. We have eliminated the models ARIMA(1,2,3) and ARIMA(1,2,2). We then compared the selected models upon their AIC and BIC values. The AIC and BIC values for ARIMA(1,2,1) were less than those for ARIMA(0,2,2). Hence, we selected the model ARIMA(1,2,1).

Forecasting

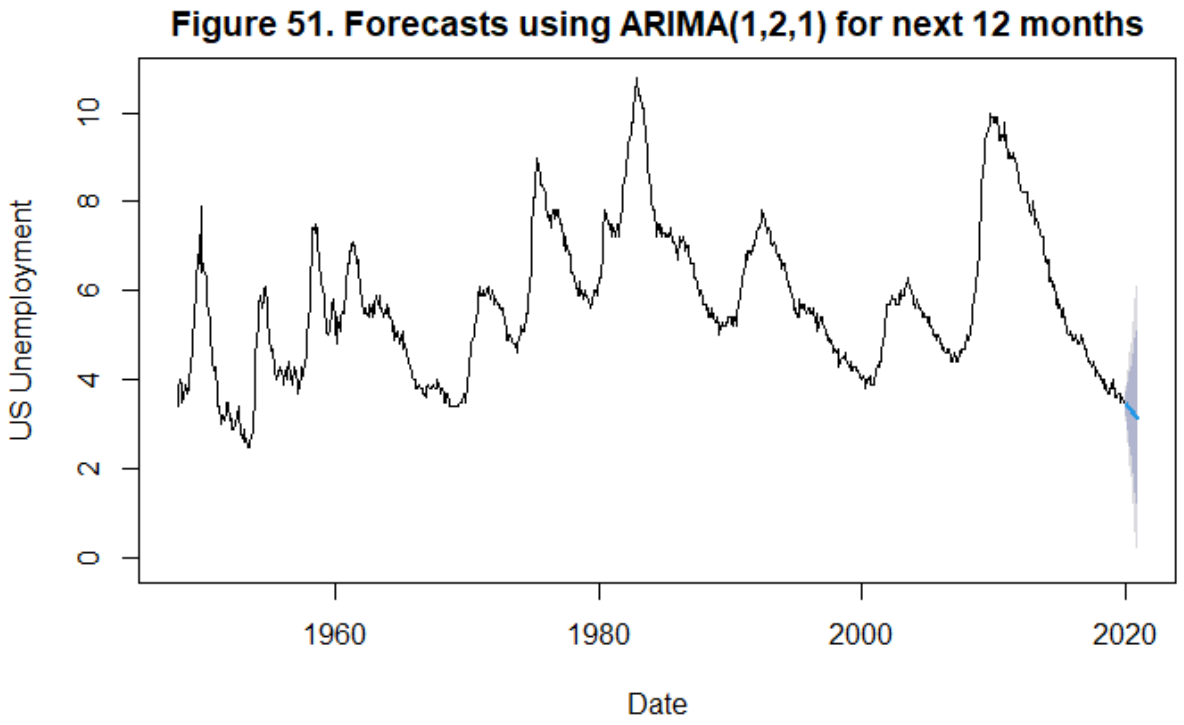
The border models for ARIMA(1,2,1) are ARIMA(0,2,1), ARIMA(2,2,1), ARIMA(1,2,0) and ARIMA(1,2,2). We have rejected ARIMA(0,2,1), ARIMA(2,2,1) and ARIMA(1,2,0) from the BIC table and the EACF. We have rejected ARIMA(1,2,2) from the residual analysis. Hence these border models are eliminated and **the final model selected for forecasting is ARIMA(1,2,1).**

We have forecasted the values for the next 12 months i.e. Jan 2020 to Dec 2020 using Arima() from the forecast package.

```
model_121A = Arima(us_unemp_tsa,order=c(1,2,1),method='CSS')
model_121Afrc = forecast::forecast(model_121A, h = 12)
model_121Afrc
```

```
##          Point Forecast    Lo 80    Hi 80      Lo 95    Hi 95
## Jan 2020       3.463666 3.206524 3.720807  3.0704020 3.856930
## Feb 2020       3.435536 3.057399 3.813673  2.8572248 4.013847
## Mar 2020       3.405553 2.889705 3.921401  2.6166318 4.194474
## Apr 2020       3.375989 2.715546 4.036431  2.3659295 4.386048
## May 2020       3.346330 2.531830 4.160830  2.1006592 4.592001
## Jun 2020       3.316692 2.339260 4.294125  1.8218384 4.811547
## Jul 2020       3.287050 2.137987 4.436113  1.5297094 5.044391
## Aug 2020       3.257409 1.928340 4.586478  1.2247734 5.290044
## Sep 2020       3.227767 1.710616 4.744919  0.9074844 5.548051
## Oct 2020       3.198126 1.485102 4.911150  0.5782822 5.817970
## Nov 2020       3.168485 1.252065 5.084904  0.2375744 6.099395
## Dec 2020       3.138843 1.011752 5.265935 -0.1142621 6.391949
```

```
plot(model_121Afrc, main = "Figure 51. Forecasts using ARIMA(1,2,1) for next 12 months
",ylab='US Unemployment',xlab='Date')
```



Conclusion

The United States unemployment data were first analyzed by using the linear models. We were unable to find a perfect fit for the data in the linear models, so we then tried to propose an ARIMA model as we saw that the time series had autoregressive and moving average behaviour. Stationarity was confirmed by checking the data. The data was differenced to the second order to improve stationarity. This differenced data was then used to propose a set of ARIMA models by using various statistical analysis techniques taught in the course. The data did not show any seasonal behaviour, so we did not proceed with SARIMA. The best model I.e. ARIMA(1,2,1) was selected to forecast future values. As we found good fits with ARIMA models we did not experiment further with ARCH and GARCH models. The model forecasted that the unemployment rate for the next 12 months will decrease consistently. This is a good sign for the population of the United States as a decrease in the unemployment rate will result in economic growth. ARIMA(1,2,1) model can be used to forecast more values in the future. This prediction can be helpful for the policymakers to plan for the future of the country.

References

1.Tunguz, B., 2020. US Monthly Unemployment Rate 1948 - Present. [online] Kaggle.com. Available at: https://www.kaggle.com/tunguz/us-monthly-unemployment-rate-1948-present [Accessed 8 May 2021].

2. Lecture Notes. 2021. Time Series Analysis. [online] Canvas.com [Accessed 9 June 2021].