TIME SERIES ANALYSIS OF GLOBAL CARBON EMISSIONS FROM FOSSIL FUELS AND CEMENT PRODUCTION

Deepthi Suresh

S3991481

Table of Contents

Table of Figures

MATH1318 Time Series Analysis / MATH2204 Time Series and Forecasting
Final Project Report

Declaration of contributions:

No	Name of Team Member	Contribution to the project
1	Deepthi Suresh	100%
2		
3		
4		
5		
6		
	Sum:	100%

RMIT University,
School of Science
2022

1. Introduction

The dataset consists of the carbon emission by the fossil fuel and cement production between the year 1959 to 2014 and comprises of 56 observations. All values in the data are measured in million tonnes carbon per year (MtC/yr). For this study, the total emissions from Coal, Oil, Gas, Cement, and Flaring is considered. The objective of the study is to answer the research question "What are the most accurate forecasts for the carbon emissions from the fossil fuel and cement production for the next 10 years?"

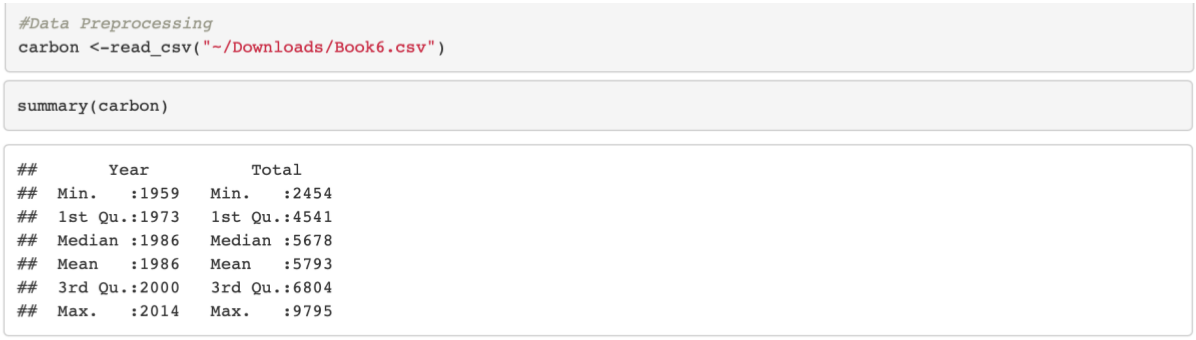
2. Method

The data used in the study is collected from the Global carbon project 2016 - Global Carbon Budget (Canadell, 2012). The descriptive analysis provides the summary of the data and provides an analysis on Time series to identify the 5 important descriptive features of the TS models - Trend, Seasonality, Changing Variance, Behavior and Change points. In case of changing variance Box-Cox transformation is conducted and in case of non-stationary data, the data is differenced to detrend the series and identify the best model that fits the series. On the stationary data, methods like ACF,PACF, EACF and BIC tables are performed to identify the possible models. The selected models are fitted using ML and CSS method and the best models are identified based on AIC, BIC and Error measure values. Once the best model is identified the data is fitted and forecasts are extracted. R programming software is used for this study. Important snippets of the code are provided in the results section and the entire code used for this project can be found in the Appendix Section.

3. Results

3.1 Descriptive Analysis

The raw data on the Total Carbon emission from the fossil fuel and cement production is uploaded into R and stored as a variable called carbon. All values in the data are in million tonnes of carbon per year (MtC/yr). The summary statistics of the raw data shows that the data is available from the year 1959 to 2014. The maximum emission from the industry is 9795 MtC in the year 2014, and the minimum emission was 2454 MtC, indicating a very high range



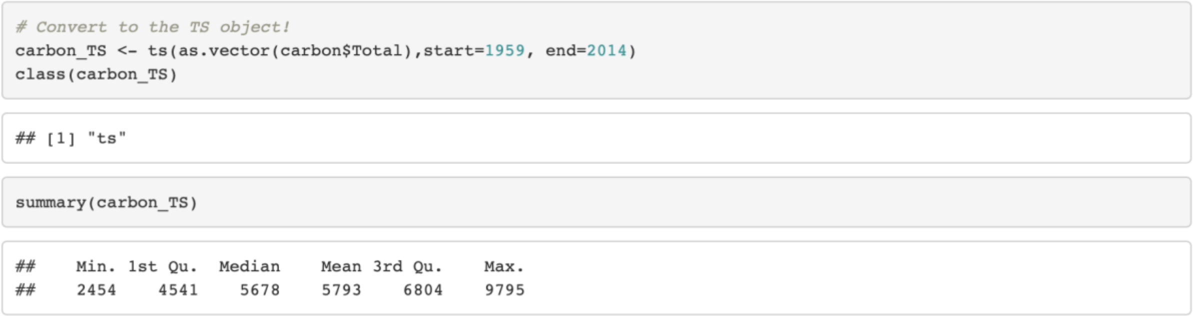
```
#Data Preprocessing
carbon <-read_csv("~/Downloads/Book6.csv")
```

```
summary(carbon)
```

```
##       Year          Total
##  Min.   :1959   Min.   :2454
##  1st Qu.:1973   1st Qu.:4541
##  Median :1986   Median :5678
##  Mean   :1986   Mean   :5793
##  3rd Qu.:2000   3rd Qu.:6804
##  Max.   :2014   Max.   :9795
```

Figure 1: Summary statistics of the Carbon Emissions raw dataResults

The data is then converted into a time series object called carbon_TS for further processing with a start date of 1959 and an end date of 2014. The frequency here is taken as 1 since it is a yearly data . The class function confirms that the data is now in a Time series format. The summary statistics, given in Figure 2, are the same as the raw data. On average the fossil fuel and cement production emit 5793 MtC/yr.



```
# Convert to the TS object!
carbon_TS <- ts(as.vector(carbon$Total),start=1959, end=2014)
class(carbon_TS)
```

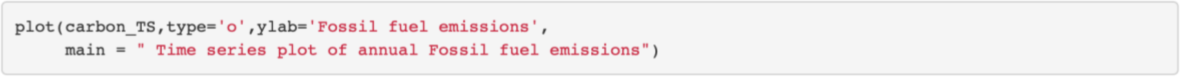
```
## [1] "ts"
```

```
summary(carbon_TS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2454    4541    5678    5793    6804    9795 
```

Figure 2: Summary Statistics of the TS object

Figure 3 shows the time series plot of the emissions data, which indicates a clear upward linear trend. There doesn’t seem to be any variation or change points. The plot also doesn’t seem to be indicating any seasonality or repeating patterns.The data seems to be moving upward in a continuous data points indicating a AR behaviour.



```
plot(carbon_TS,type='o',ylab='Fossil fuel emissions',
     main = " Time series plot of annual Fossil fuel emissions")
```

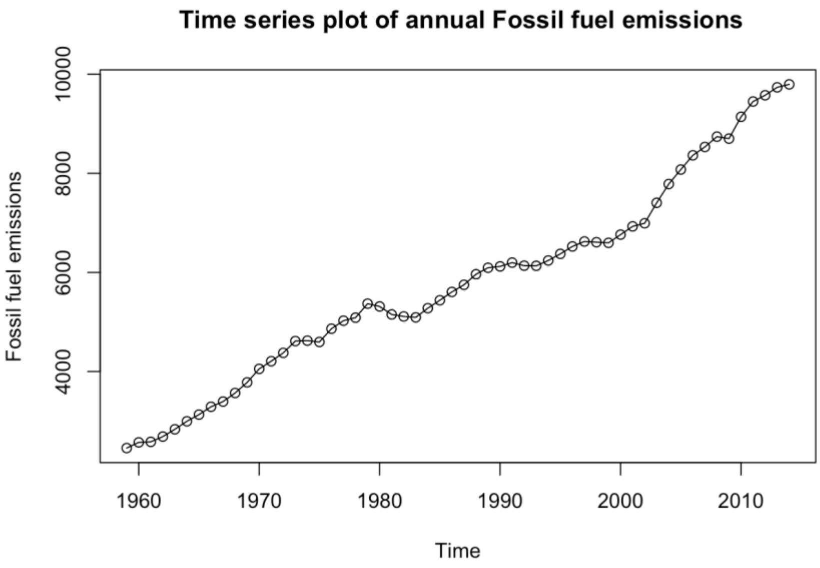
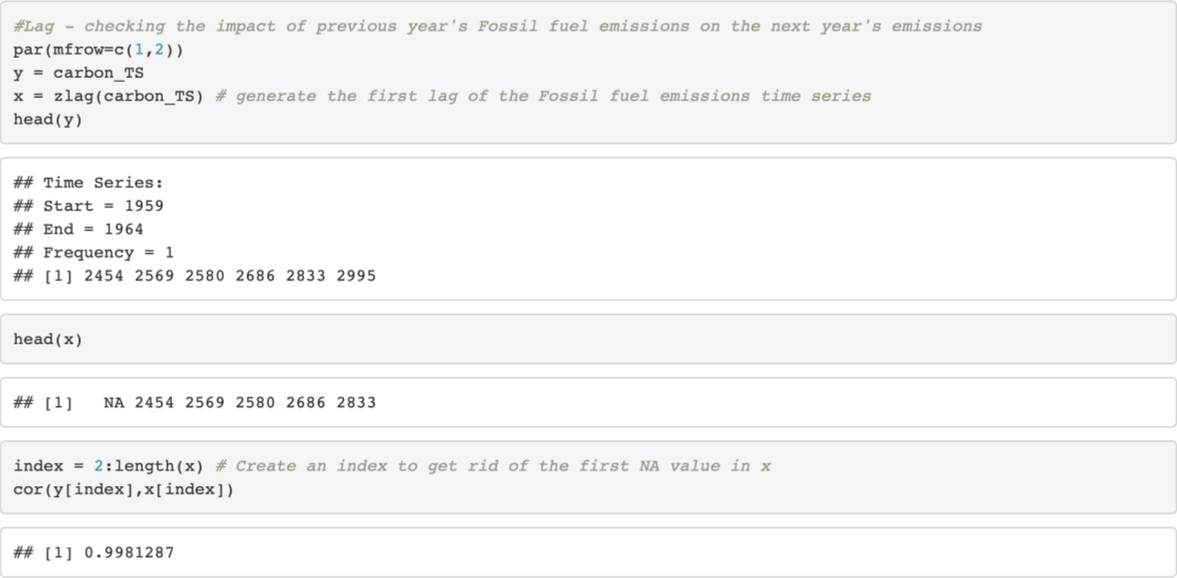


Figure 3 Time series plot of the Carbon Emissions data

Further analysis have been performed on the data to look into the possibility of a relationship between previous year’s emissions with the next year’s emissions. Figure 4 shows that there is a very strong positive correlation of 0.998 between the previous year’s emissions with the next year’s emissions.



```
#Lag - checking the impact of previous year's Fossil fuel emissions on the next year's emissions
par(mfrow=c(1,2))
y = carbon_TS
x = zlag(carbon_TS) # generate the first lag of the Fossil fuel emissions time series
head(y)
```

```
## Time Series:
## Start = 1959
## End = 1964
## Frequency = 1
## [1] 2454 2569 2580 2686 2833 2995
```

```
head(x)
```

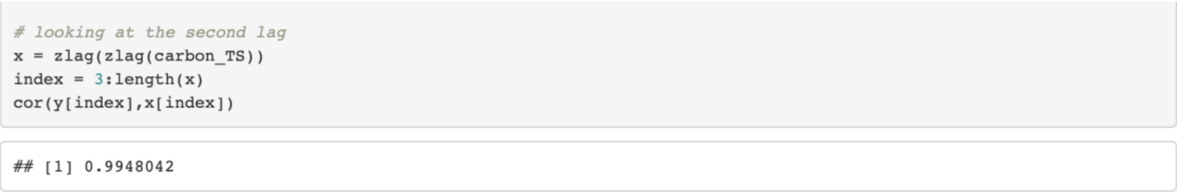
```
## [1]   NA 2454 2569 2580 2686 2833
```

```
index = 2:length(x) # Create an index to get rid of the first NA value in x
cor(y[index],x[index])
```

```
## [1] 0.9981287
```

Figure 4: Correlation of Lag 1

Correlation of the second lag as seen in Figure 5, also indicates a very high positive correlation of 0.994, there is a correlation between current year’s emissions with the emissions that were present two years ago, however the correlation is slightly lower than Lag 1.

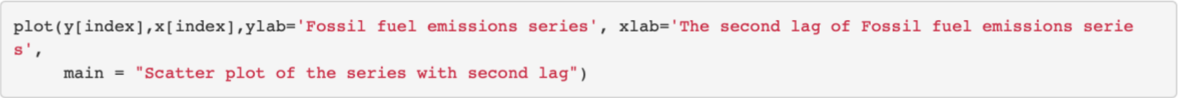


```
# looking at the second lag
x = zlag(zlag(carbon_TS))
index = 3:length(x)
cor(y[index],x[index])
```

```
## [1] 0.9948042
```

Figure 5: Correlation of Lag 2

Figure 6 shows the side-by side comparison of the scatter plot for Lag 1 and Lag 2, confirming our earlier observation of the relationships with the previous years emissions. The plot of Lag 1 is almost in a straight line pointing towards a very strong correlation between the two. Plot of Lag 2 is almost in a straight line, but a few of the data points are slightly more distributed that Lag1



```
plot(y[index],x[index],ylab='Fossil fuel emissions series', xlab='The second lag of Fossil fuel emissions serie
s',
     main = "Scatter plot of the series with second lag")
```

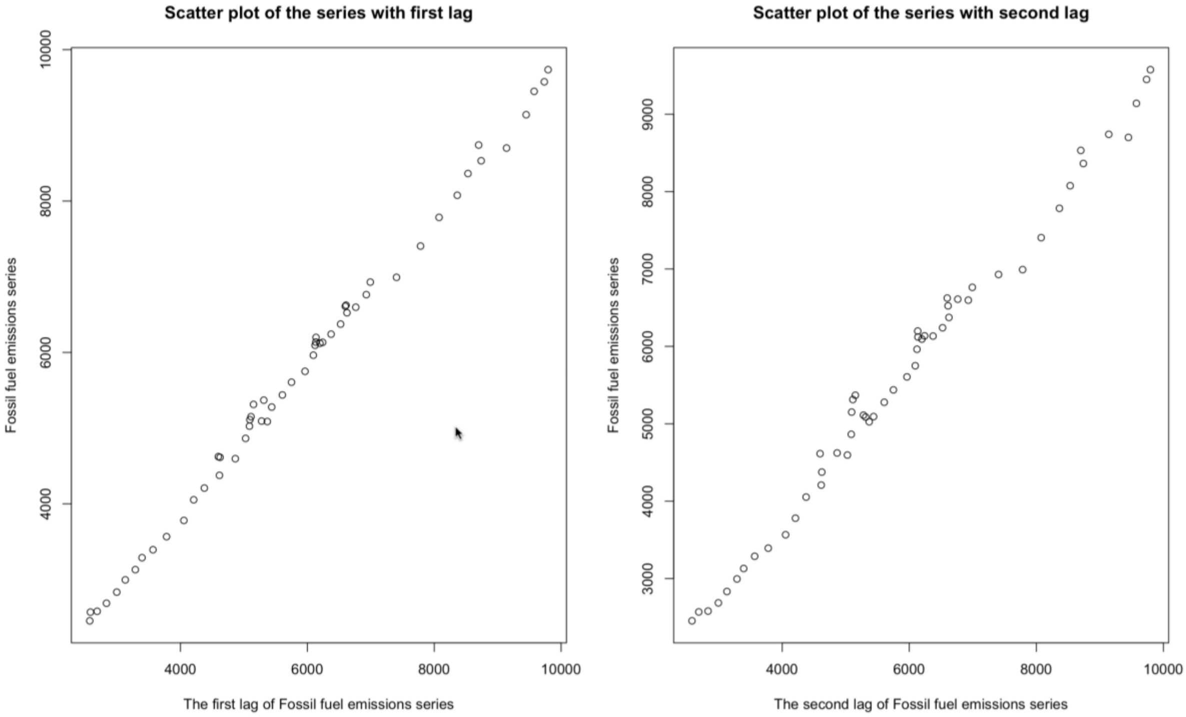


Figure 6: Side-by side scatter plot of Lag 1 and Lag 2

Figure 7 shows the ACF of the Emissions series with a wave like decaying pattern indicating a possible trend and seasonality. There are also significant autocorrelations with several lines above the upper interval and lower interval.

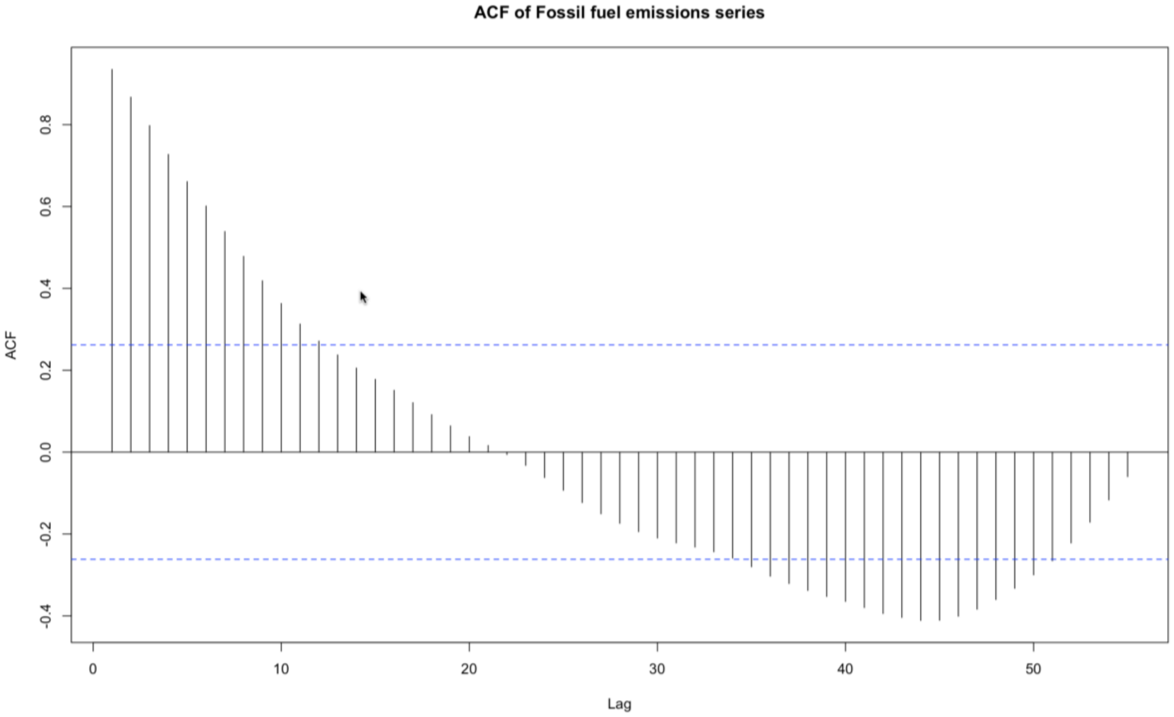
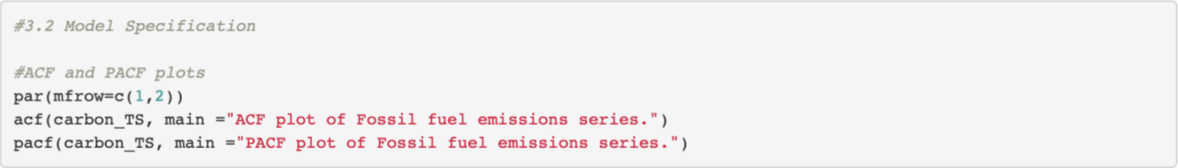


Figure 7: ACF plot of the emissions data

3.2 Model Specification

The descriptive analysis showed that the series has trend and therefore could be a non-stationary series. The ACF and the PACF plot of the emissions series is given in Figure 8. The ACF plot shows a slowly decaying pattern and the PACF plot shows one significantly high partial auto correlation in lag 1, both of which are indicators that the series is non-stationary.



```
#3.2 Model Specification

#ACF and PACF plots
par(mfrow=c(1,2))
acf(carbon_TS, main ="ACF plot of Fossil fuel emissions series.")
pacf(carbon_TS, main ="PACF plot of Fossil fuel emissions series.")
```

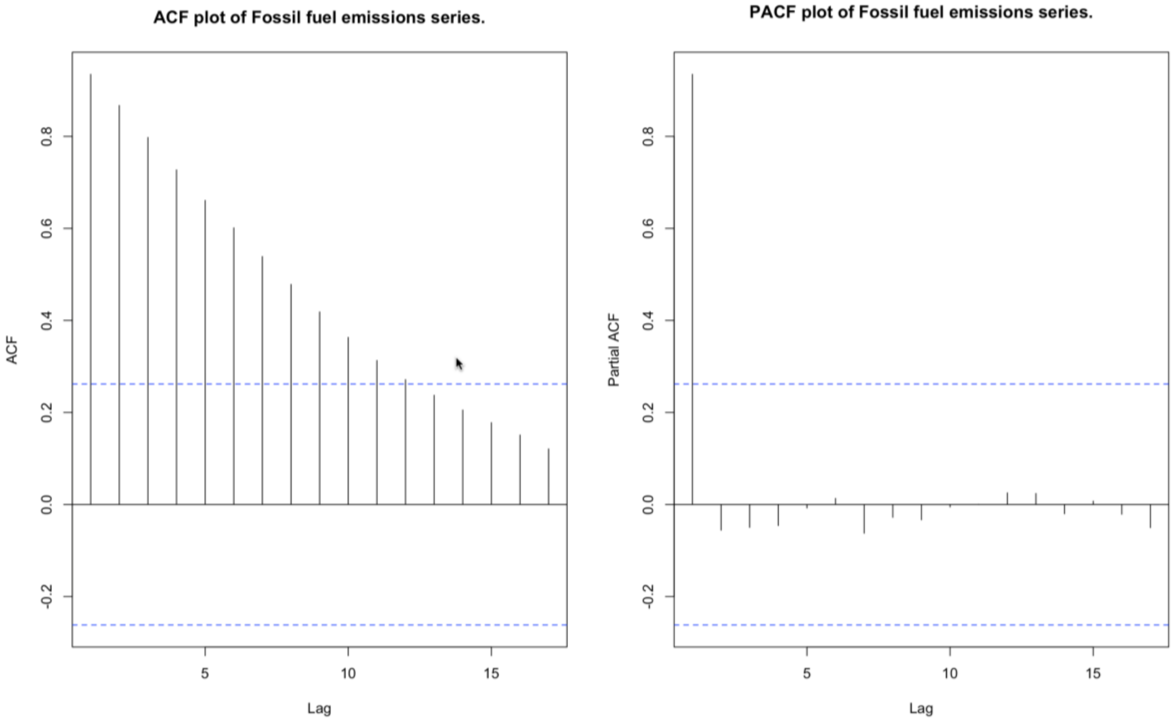
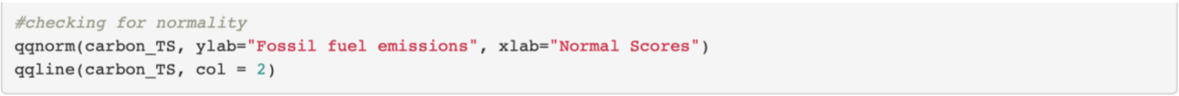


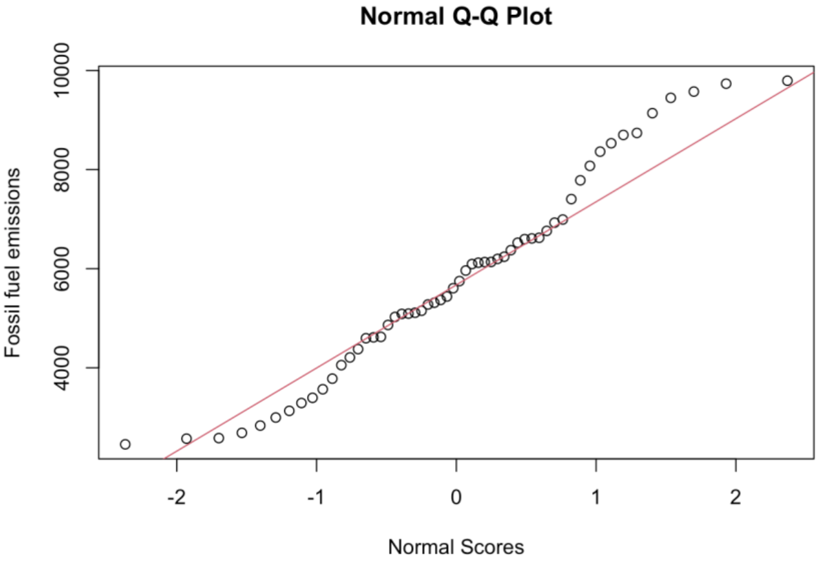
Figure 8: ACF and PACF plot of Fossil fuel emissions series

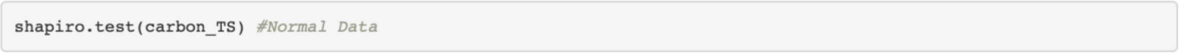
Before identifying and fitting the models, it is important to convert a non-stationary series into a stationary series.

QQ plot in Figure 9, shows that there are data points that deviate from the QQ line, indicating that the series is not normal. However, the p-value in the Shapiro Wilks test is 0.10 which is greater than 0.05 and therefore according to the test the series is normally distributed.

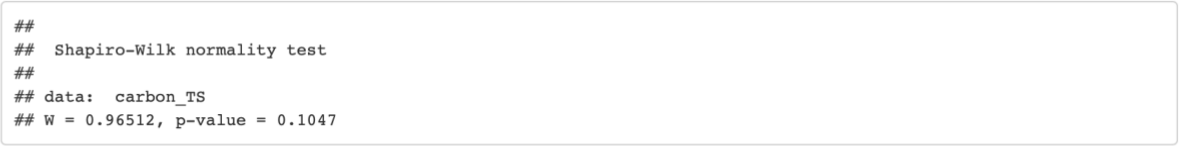


```
#checking for normality
qqnorm(carbon_TS, ylab="Fossil fuel emissions", xlab="Normal Scores")
qqline(carbon_TS, col = 2)
```





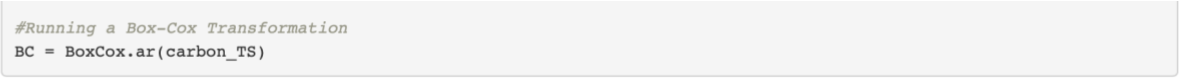
```
shapiro.test(carbon_TS) #Normal Data
```



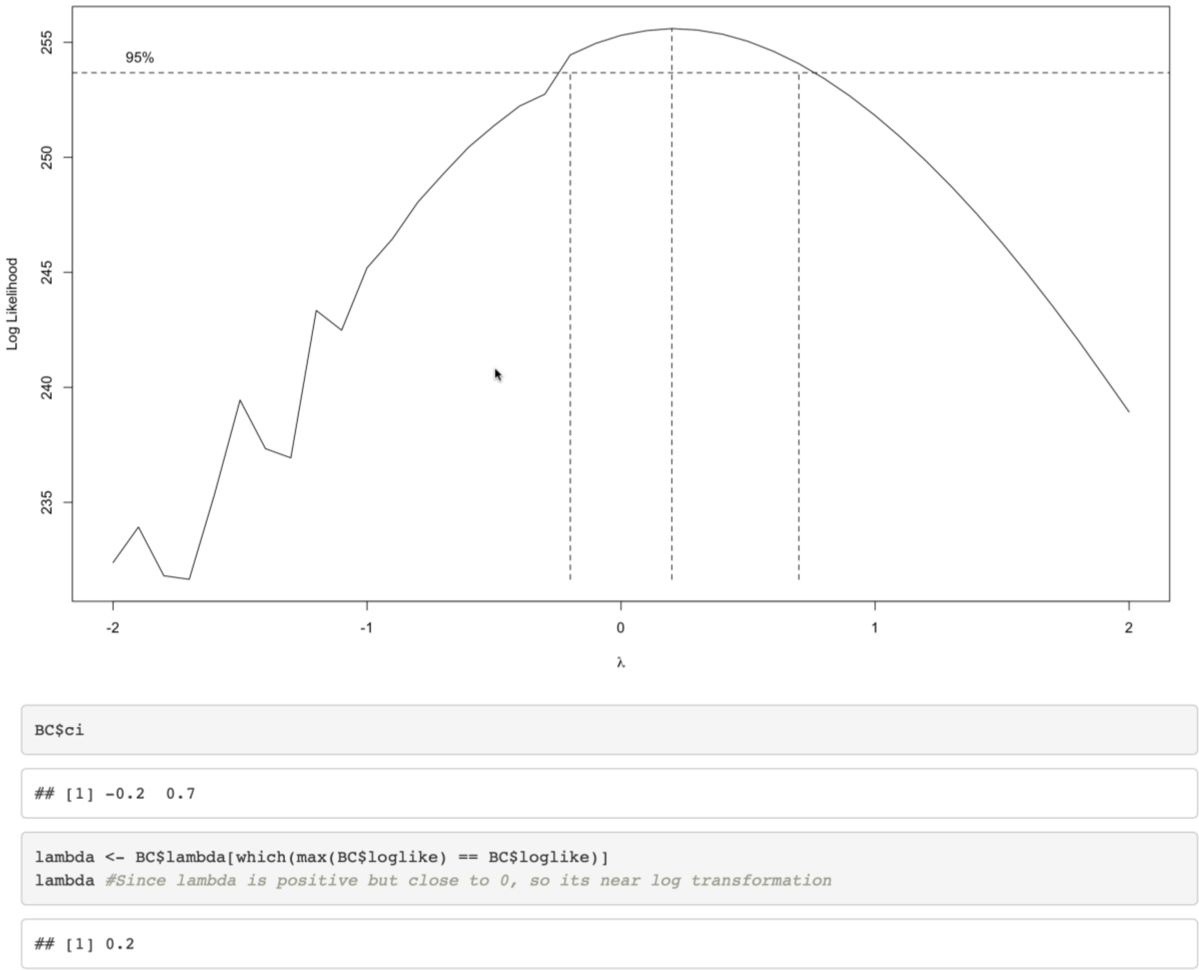
```
##
##  Shapiro-Wilk normality test
##
## data:  carbon_TS
## W = 0.96512, p-value = 0.1047
```

Figure 9: QQ plot and Shapiro- Wilk normality Test of the Emission Series

Conducting a Box-Cox Transformation, returned a Lambda of 0.2. Since lambda is positive but close to 0, it could possibly be a log transformation



```
#Running a Box-Cox Transformation
BC = BoxCox.ar(carbon_TS)
```



```
BC$ci
```

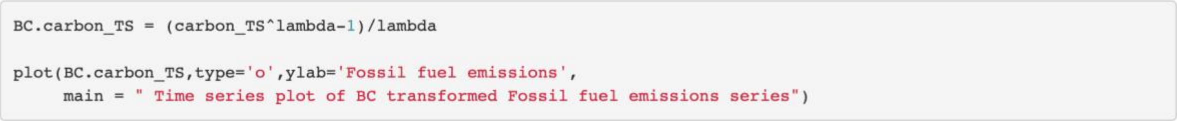
```
## [1] -0.2  0.7
```

```
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda #Since lambda is positive but close to 0, so its near log transformation
```

```
## [1] 0.2
```

Figure 10 Box Cox transformation and Lambda calculation of Emissions Series

Since there were no noticeable signs of changing variance in the initial time series plot, the BC transformation didn’t make a lot of difference in the Box-Cox transformed TS plot as shown in figure 11.



```
BC.carbon_TS = (carbon_TS^lambda-1)/lambda

plot(BC.carbon_TS,type='o',ylab='Fossil fuel emissions',
     main = " Time series plot of BC transformed Fossil fuel emissions series")
```

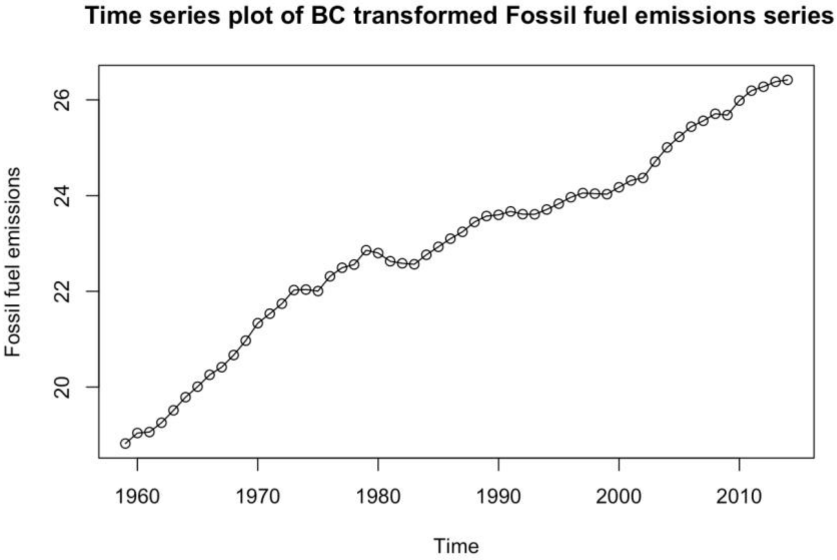
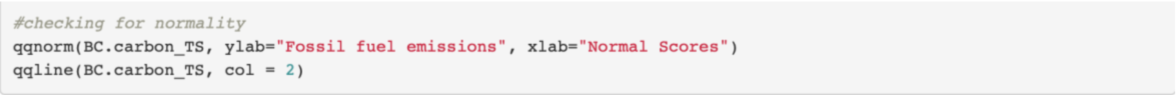
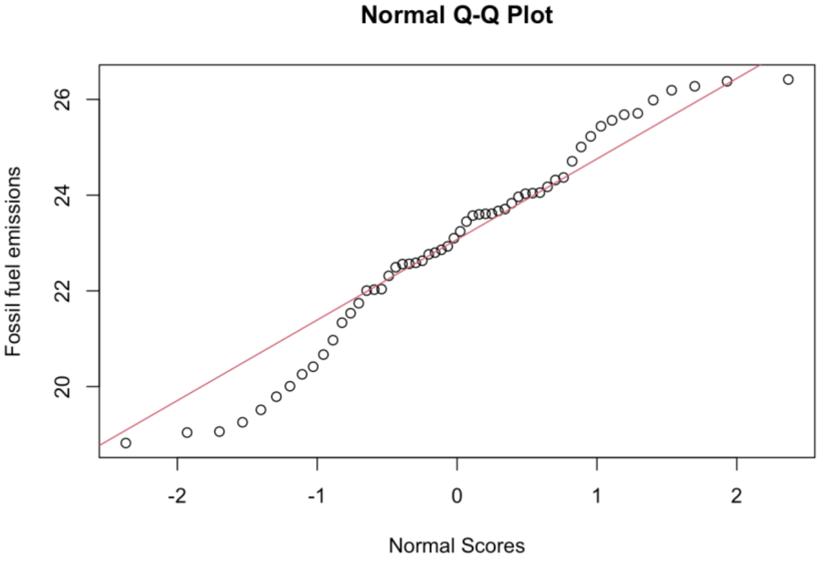


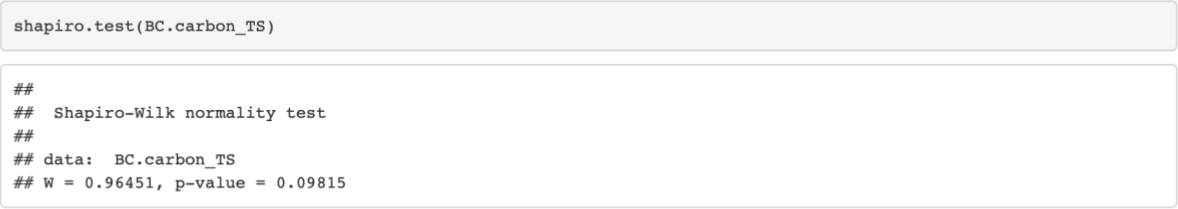
Figure 11 Time series plot of the Box Cox transformed Emissions series

Figure 12 shows the QQ plot of the BC transformed data, which shows that the data points have moved further away from the QQ line and the p-value in the Shapiro Wilks test is slightly lower that the p-value in the Shapiro Wilks test of the raw data. So, I have decided to use the raw data for further analysis.



```
#checking for normality
qqnorm(BC.carbon_TS, ylab="Fossil fuel emissions", xlab="Normal Scores")
qqline(BC.carbon_TS, col = 2)
```



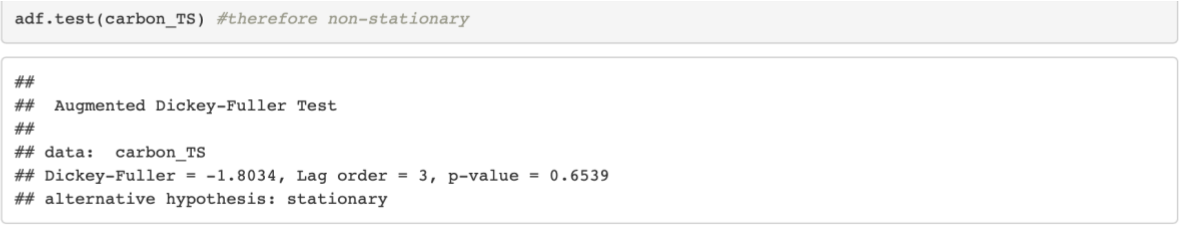


```
shapiro.test(BC.carbon_TS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BC.carbon_TS
## W = 0.96451, p-value = 0.09815
```

Figure 12 QQ plot and Shapiro-Wilk normality test for the Box-Cox transformed series

To further confirm the stationarity of the data, an ADF test has been performed, and the results are shown in figure 13. ADF test is a hypothesis test run to check the stationarity of a series. Since the p-value is greater than 0.05 we fail to reject the null hypothesis and therefore the series is non-stationary.

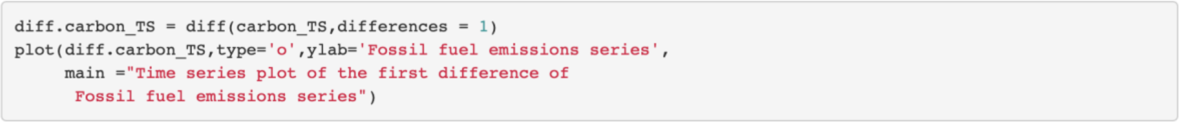


```
adf.test(carbon_TS) #therefore non-stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  carbon_TS
## Dickey-Fuller = -1.8034, Lag order = 3, p-value = 0.6539
## alternative hypothesis: stationary
```

Figure 13 ADF test of the raw emission series

To convert the non-stationary data into a stationary form, differencing is performed on the data. Here the raw data is considered as the BC transformation did not make much of a difference to the series. In this case differencing once seems to handle most of the trend. The time series plot, as shown in figure 14, indicates a likely stationary data series, with no trend.



```
diff.carbon_TS = diff(carbon_TS,differences = 1)
plot(diff.carbon_TS,type='o',ylab='Fossil fuel emissions series',
     main ="Time series plot of the first difference of
      Fossil fuel emissions series")
```

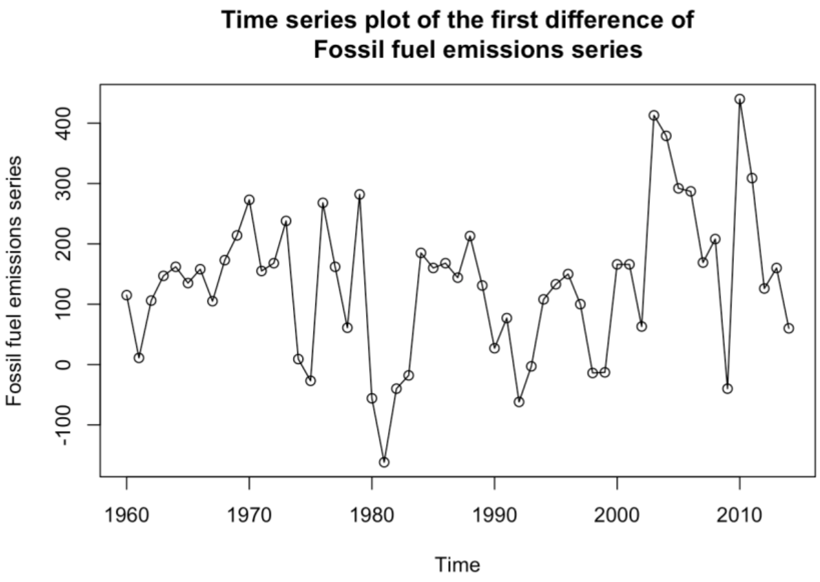
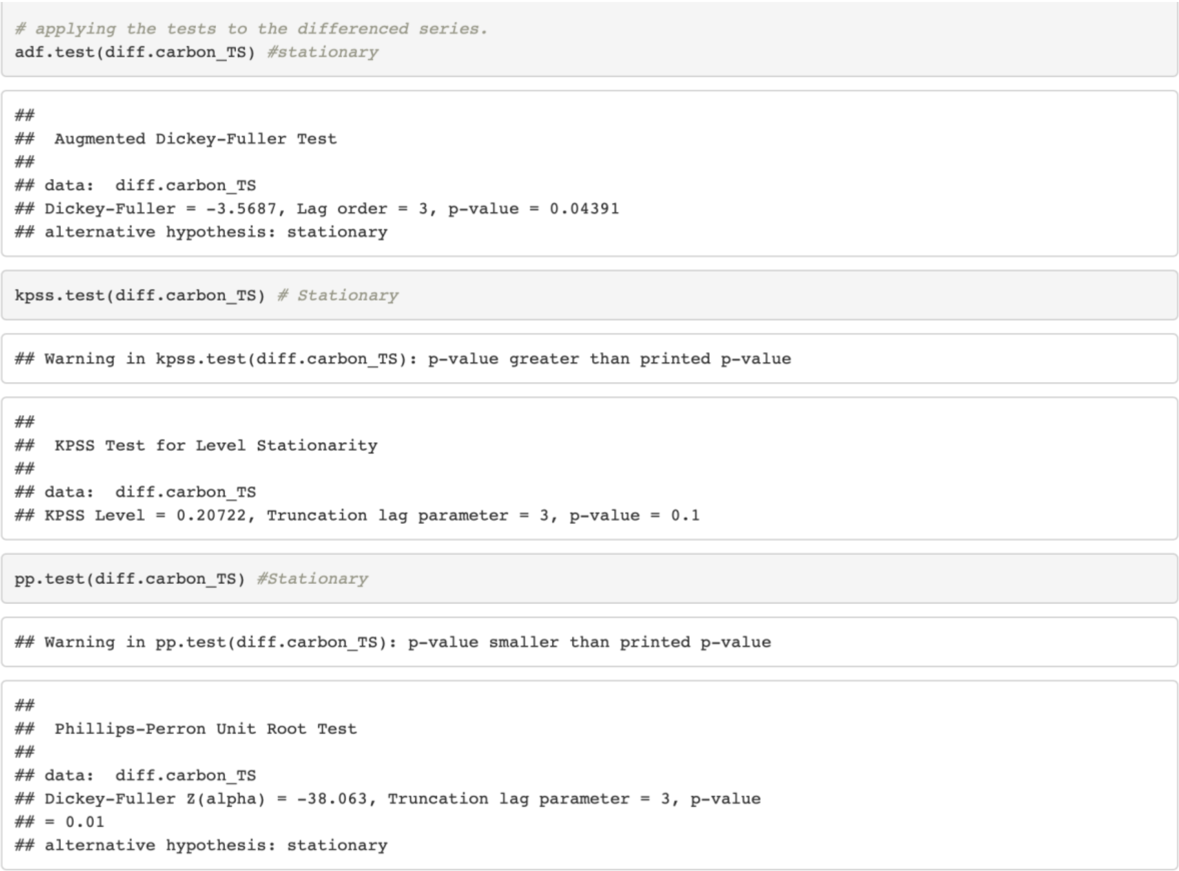


Figure 14 Time series plot of first difference of the Emissions Series

The assumption of stationarity can be further confirmed by running multiple hypothesis tests such as the ADF test, PP test and KPSS Test, the results of which are in Figure 15. The null hypothesis in the ADF and PP test states that the series is non-stationary and the null hypothesis in KPSS test states that the series is stationary. Here the p-value of the ADF test and PP test is less than 0.05 and therefore we can reject the null hypothesis, meaning that the series in stationary. Similarly, the KPSS test has a p-value of 0.1, which is greater than 0.05 and therefore we fail to reject the null hypothesis and the series is confirmed to be stationary. We can therefore conclude that the differencing once converted the series from non-stationary into a stationary series.



```
# applying the tests to the differenced series.
adf.test(diff.carbon_TS) #stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.carbon_TS
## Dickey-Fuller = -3.5687, Lag order = 3, p-value = 0.04391
## alternative hypothesis: stationary
```

```
kpss.test(diff.carbon_TS) # Stationary
```

```
## Warning in kpss.test(diff.carbon_TS): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff.carbon_TS
## KPSS Level = 0.20722, Truncation lag parameter = 3, p-value = 0.1
```

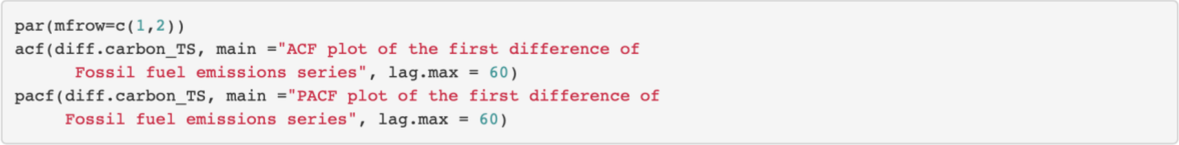
```
pp.test(diff.carbon_TS) #Stationary
```

```
## Warning in pp.test(diff.carbon_TS): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  diff.carbon_TS
## Dickey-Fuller Z(alpha) = -38.063, Truncation lag parameter = 3, p-value
## = 0.01
## alternative hypothesis: stationary
```

Figure 15 Hypothesis testing conducted on the differenced series indicating stationarity

Figure 16 shows the ACF and PACF of the first differenced Emissions series. It can be noticed that the ACF plot no longer has a slowly decaying pattern and PACF no longer has as high of a significant autocorrelation in the first lag. Therefore confirming that the series is converted into a stationary series



```
par(mfrow=c(1,2))
acf(diff.carbon_TS, main ="ACF plot of the first difference of
      Fossil fuel emissions series", lag.max = 60)
pacf(diff.carbon_TS, main ="PACF plot of the first difference of
     Fossil fuel emissions series", lag.max = 60)
```

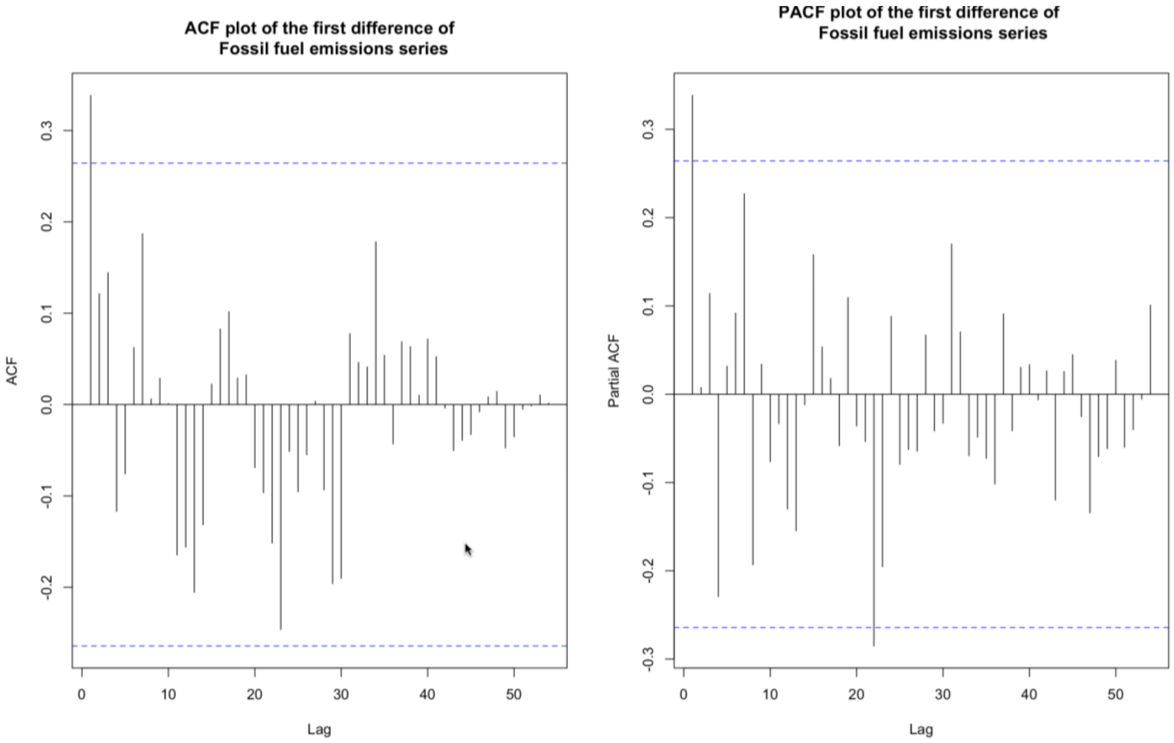
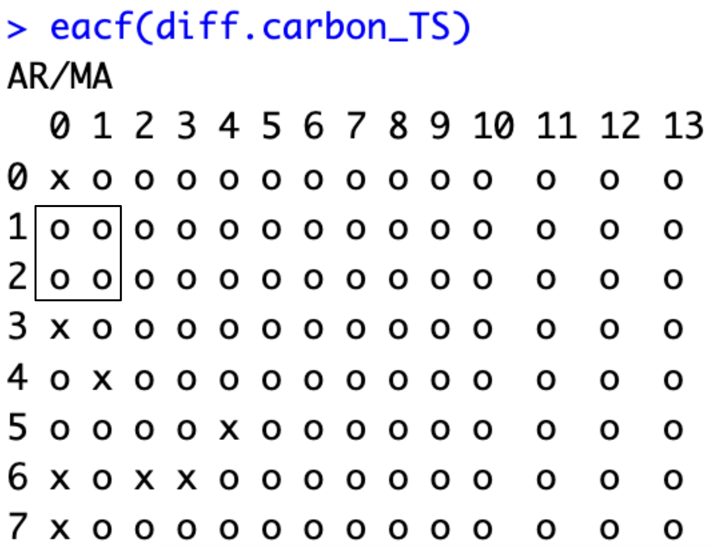


Figure 16 ACF and PACF of the first differenced emissions Series

From the ACF and PACF plot, we can determine p as 1, as there is one significant lag in PACF and q as 1 as as there is one significant lag in ACF and d as 1 as the data is differenced once. The significant lag in PACF after year 20 is not considered as it is a late lag. So, the possible model from ACF and PACF is

ARIMA{1,1,1}

Figure 17 shows the EACF plot of the Emissions series. The top left corner is considered as (1,0) as it is the left most point without being interrupted by x.



```
> eacf(diff.carbon_TS)
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x o o o o o o o o o o  o  o  o
1 o o o o o o o o o o o  o  o  o
2 o o o o o o o o o o o  o  o  o
3 x o o o o o o o o o o  o  o  o
4 o x o o o o o o o o o  o  o  o
5 o o o o x o o o o o o  o  o  o
6 x o x x o o o o o o o  o  o  o
7 x o o o o o o o o o o  o  o  o
```

Figure 17 EACF of the differenced Emissions Series

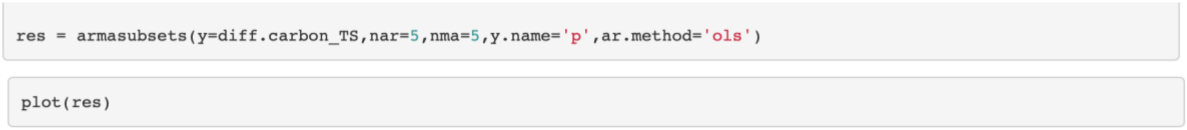
The set of possible models from the EACF are:

ARIMA {1,1,0}, ARIMA {1,1,1}, ARIMA {2,1,0}, ARIMA {2,1,1}

Figure 18 gives the BIC table of the Emissions series. Nar and Nma is considered as 5 since the models from the previous methods are on the smaller side. The best model from BIC Table is ARIMA {1,1,0} as the p of 1 is supported by all the subsequent models.The second best model is ARIMA {4,1,0}. The p of 4 is supported by all models after second except fourth.

The set of possible models from the BIC table are:

ARIMA {1,1,0} and ARIMA {4,1,0}



```
res = armasubsets(y=diff.carbon_TS,nar=5,nma=5,y.name='p',ar.method='ols')
```

```
plot(res)
```

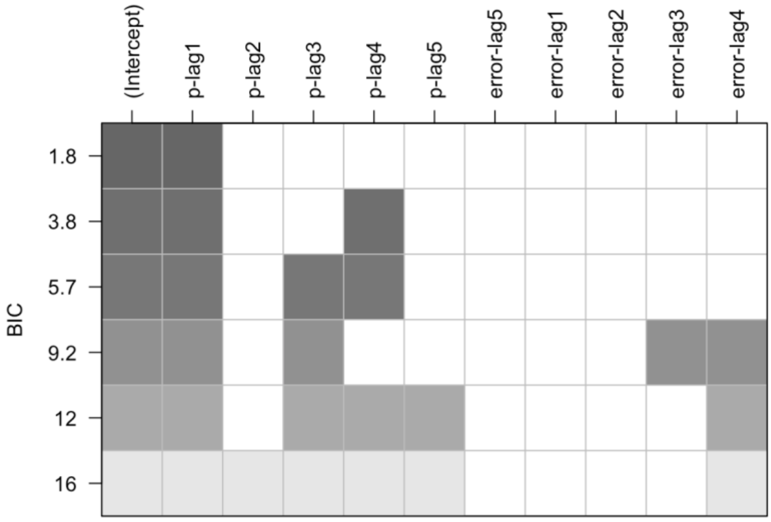


Figure 18 BIC table of differenced Emissions series

The final set of possible models are :

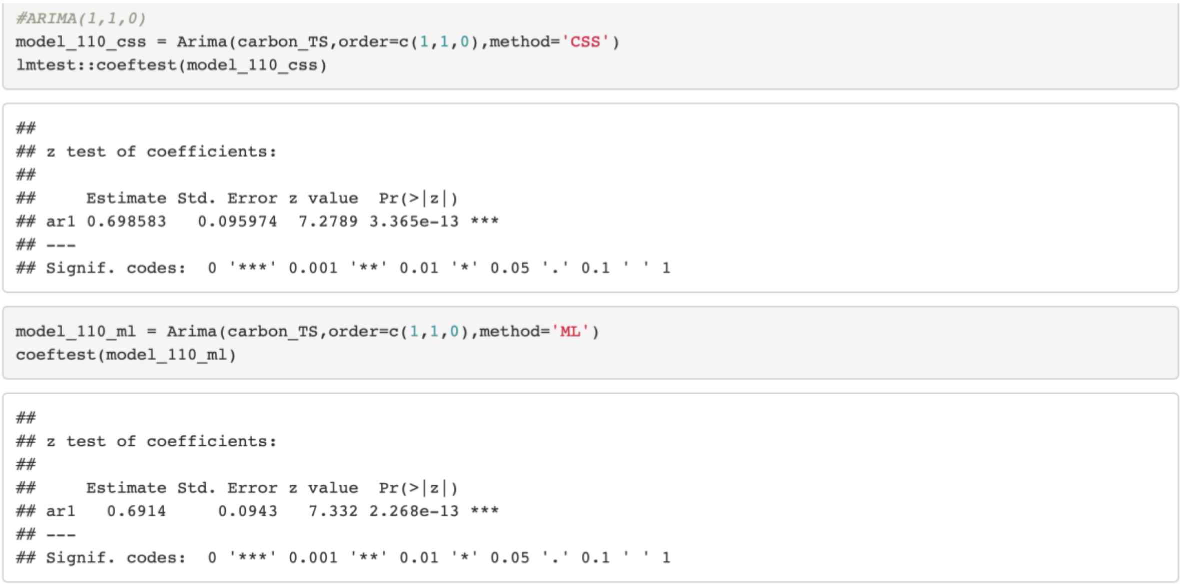
ARIMA {1,1,0}, ARIMA {4,1,0},ARIMA {1,1,1},ARIMA {2,1,0},ARIMA {2,1,1}

3.3 Model Fitting

The set of possible model are then fitted using CSS,ML or CSS-ML parameters

3.3.1 ARIMA(1,1,0)

Figure 19 shows that, in both ML and CSS parameter estimation method the coefficients are significant making this a candidate for a good model.



```
#ARIMA(1,1,0)
model_110_css = Arima(carbon_TS,order=c(1,1,0),method='CSS')
lmtest::coeftest(model_110_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 0.698583   0.095974  7.2789 3.365e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

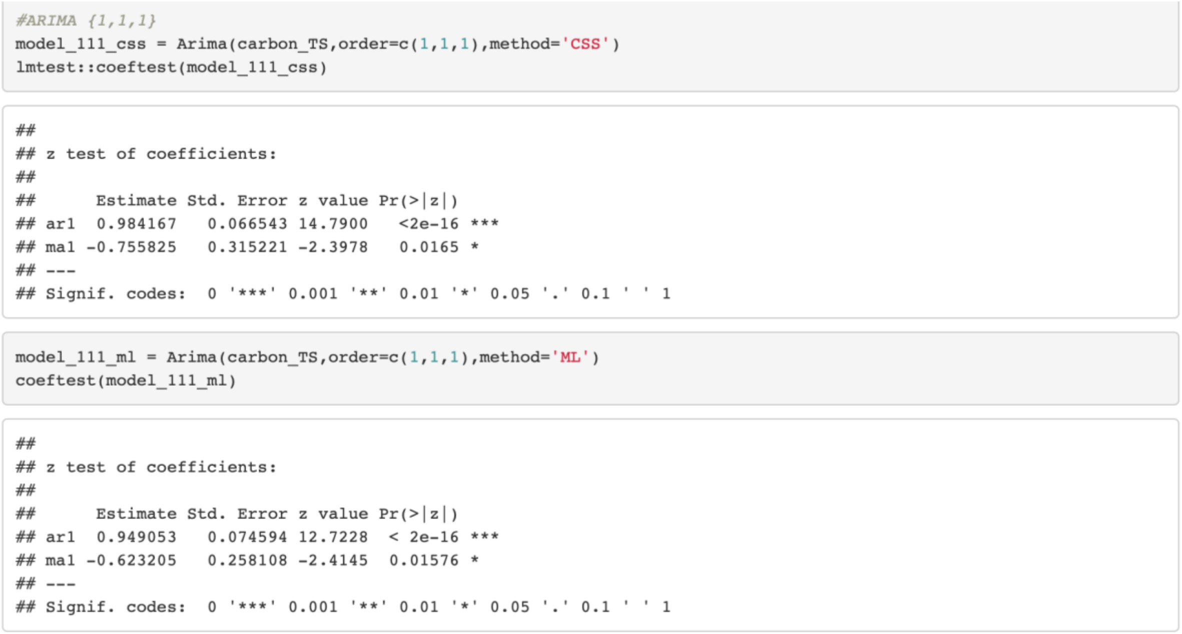
```
model_110_ml = Arima(carbon_TS,order=c(1,1,0),method='ML')
coeftest(model_110_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1   0.6914     0.0943   7.332 2.268e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 19 CSS and ML model fitting of ARIMA(1,1,0)

3.3.1 ARIMA {1,1,1}

Figure 20 shows the parameter estimation for ARIMA {1,1,1}, even after adding MA1 component the coefficients of the model are significant in both CSS and ML methods making this a candidate for a good model.



```
#ARIMA {1,1,1}
model_111_css = Arima(carbon_TS,order=c(1,1,1),method='CSS')
lmtest::coeftest(model_111_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.984167   0.066543 14.7900   <2e-16 ***
## ma1 -0.755825   0.315221 -2.3978   0.0165 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

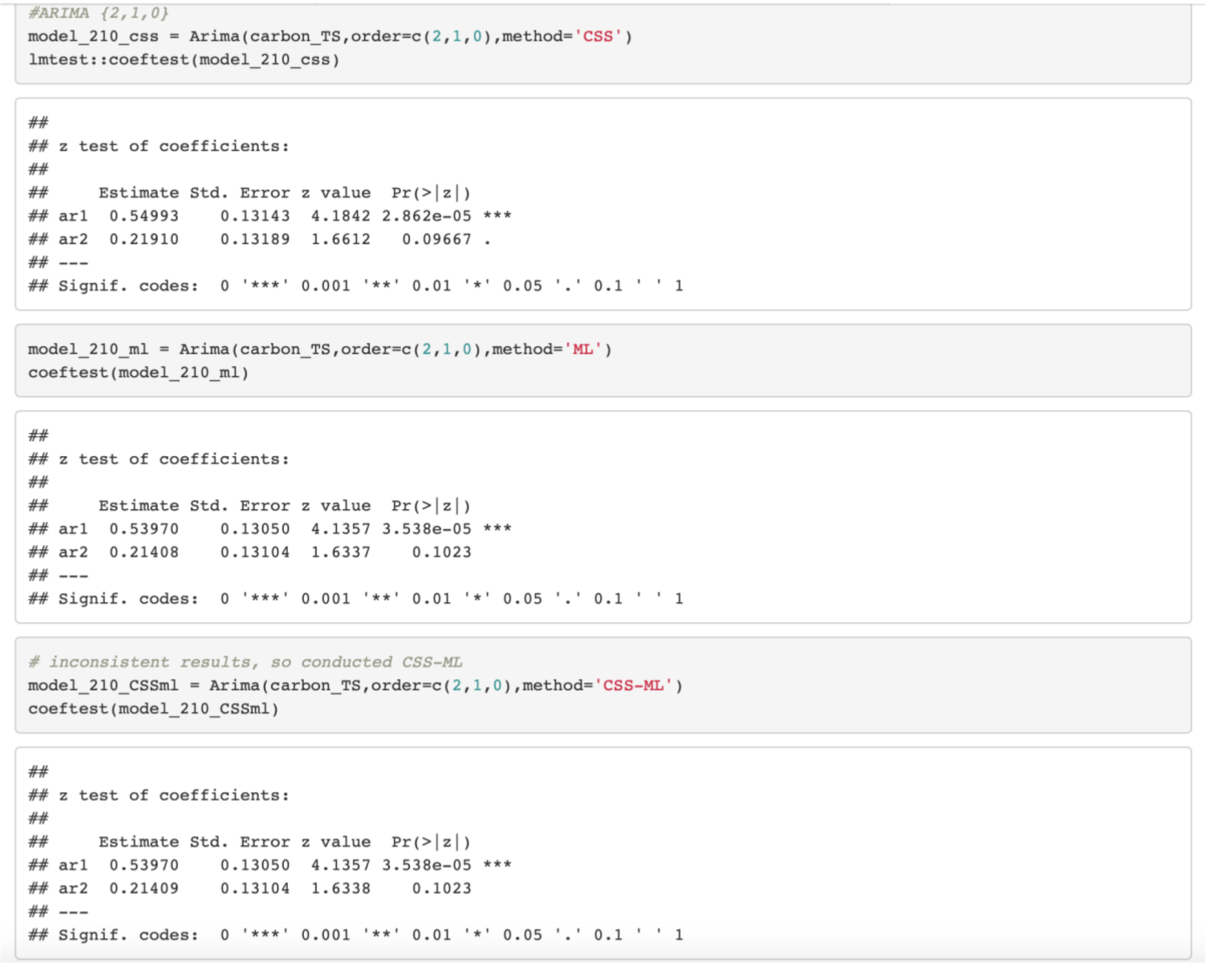
```
model_111_ml = Arima(carbon_TS,order=c(1,1,1),method='ML')
coeftest(model_111_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.949053   0.074594 12.7228  < 2e-16 ***
## ma1 -0.623205   0.258108 -2.4145  0.01576 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 20 CSS and ML model fitting of ARIMA(1,1,1)

3.3.3 ARIMA(2,1,0)

The parameter estimation of ARIMA {2,1,0} is in Figure 21, by adding another AR component, the CSS method returns significant coefficients, however the ML model returns insignificant coefficient for AR2 component. Since there is inconsistency between the two methos CSS-ML method of estimation is also conducted, and it again returns an insignificant coefficient for AR2 component. Therefore this may not be a good model



```
#ARIMA {2,1,0}
model_210_css = Arima(carbon_TS,order=c(2,1,0),method='CSS')
lmtest::coeftest(model_210_css)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.54993    0.13143  4.1842 2.862e-05 ***
## ar2  0.21910    0.13189  1.6612   0.09667 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_210_ml = Arima(carbon_TS,order=c(2,1,0),method='ML')
coeftest(model_210_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.53970    0.13050  4.1357 3.538e-05 ***
## ar2  0.21408    0.13104  1.6337    0.1023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

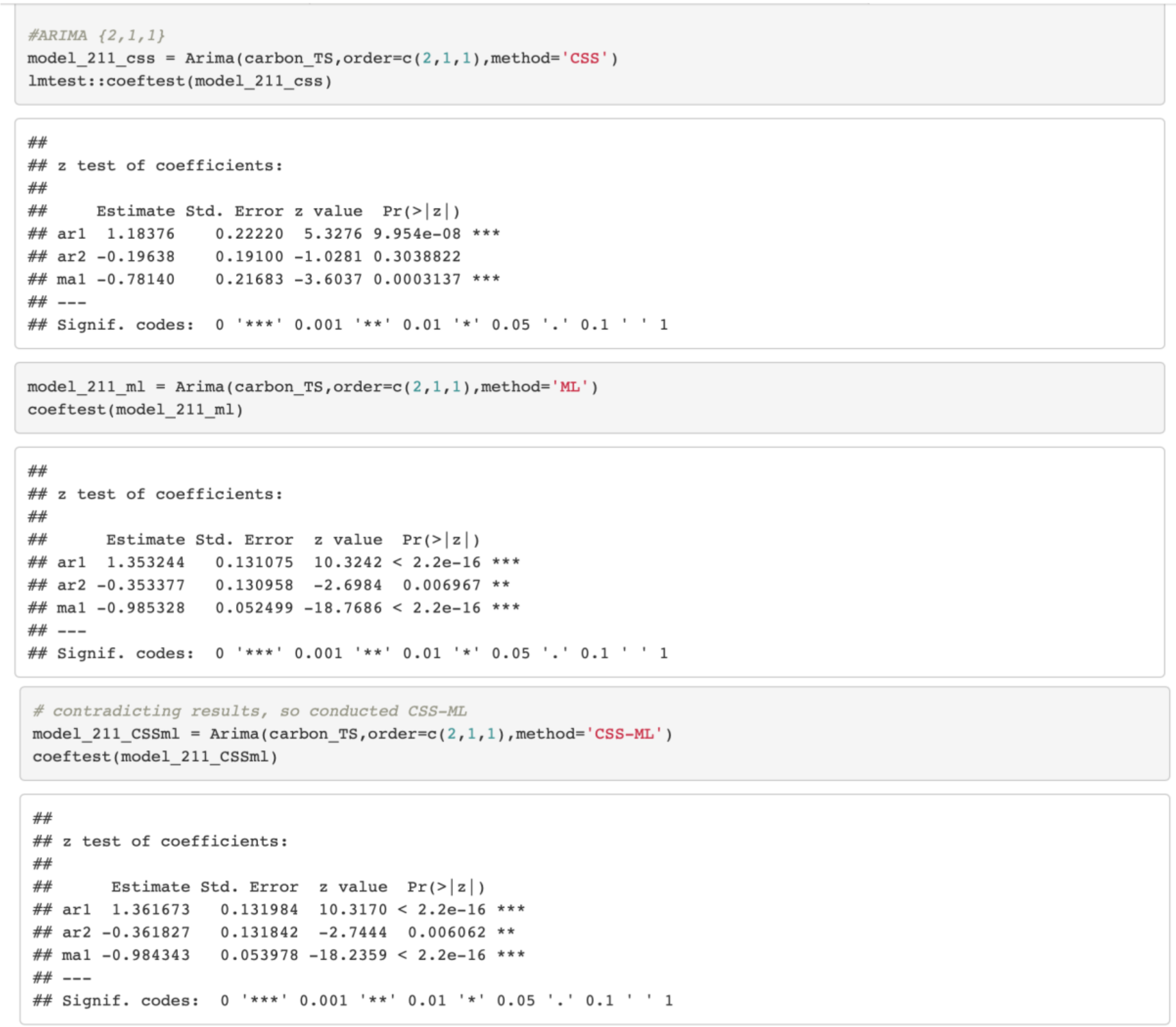
```
# inconsistent results, so conducted CSS-ML
model_210_CSSml = Arima(carbon_TS,order=c(2,1,0),method='CSS-ML')
coeftest(model_210_CSSml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.53970    0.13050  4.1357 3.538e-05 ***
## ar2  0.21409    0.13104  1.6338    0.1023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 21 CSS , ML and CSS-ML model fitting of ARIMA(2,1,0)

3.3.4 ARIMA(2,1,1)

Figure 22 shows the results for ARIMA{2,1,1}. By adding a AR2 component, we can see that the coefficient of AR2 component turned insignificant in CSS method , but significant in ML method. The newly added MA1 component is significant in all three methods Since there is inconsistency between the two method CSS-ML method of estimation is also conducted, and once again all three components are significant, making this a good model.



```
#ARIMA {2,1,1}
model_211_css = Arima(carbon_TS,order=c(2,1,1),method='CSS')
lmtest::coeftest(model_211_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  1.18376    0.22220  5.3276 9.954e-08 ***
## ar2 -0.19638    0.19100 -1.0281 0.3038822
## ma1 -0.78140    0.21683 -3.6037 0.0003137 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_211_ml = Arima(carbon_TS,order=c(2,1,1),method='ML')
coeftest(model_211_ml)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.353244   0.131075  10.3242 < 2.2e-16 ***
## ar2 -0.353377   0.130958  -2.6984  0.006967 **
## ma1 -0.985328   0.052499 -18.7686 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

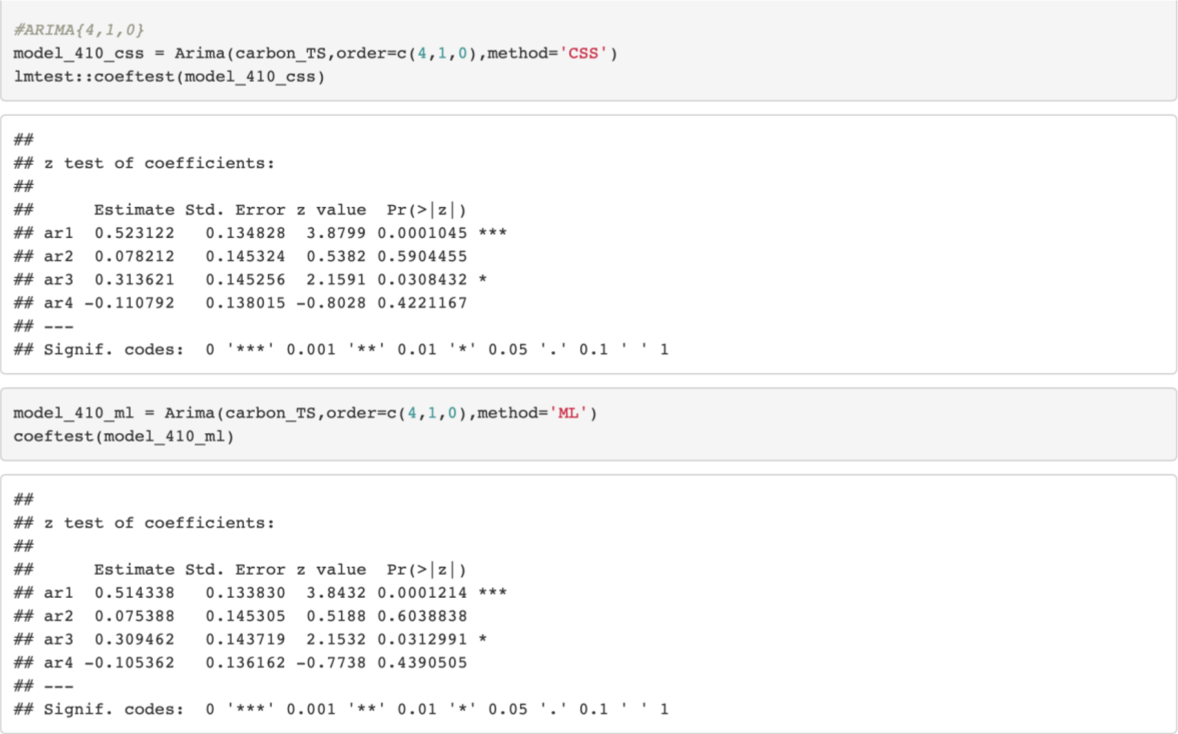
```
# contradicting results, so conducted CSS-ML
model_211_CSSml = Arima(carbon_TS,order=c(2,1,1),method='CSS-ML')
coeftest(model_211_CSSml)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  1.361673   0.131984  10.3170 < 2.2e-16 ***
## ar2 -0.361827   0.131842  -2.7444  0.006062 **
## ma1 -0.984343   0.053978 -18.2359 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 22: CSS, ML and CSS-ML model fitting of ARIMA(2,1,1)

3.3.5 ARIMA(4,1,0)

Figure 23 shows that the coefficient of AR3 component is significant in both CSS and ML methods, however AR4 remains insignificant in both, along with AR2 and therefore this might not be a good model.



```
#ARIMA{4,1,0}
model_410_css = Arima(carbon_TS,order=c(4,1,0),method='CSS')
lmtest::coeftest(model_410_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.523122   0.134828  3.8799 0.0001045 ***
## ar2  0.078212   0.145324  0.5382 0.5904455
## ar3  0.313621   0.145256  2.1591 0.0308432 *
## ar4 -0.110792   0.138015 -0.8028 0.4221167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

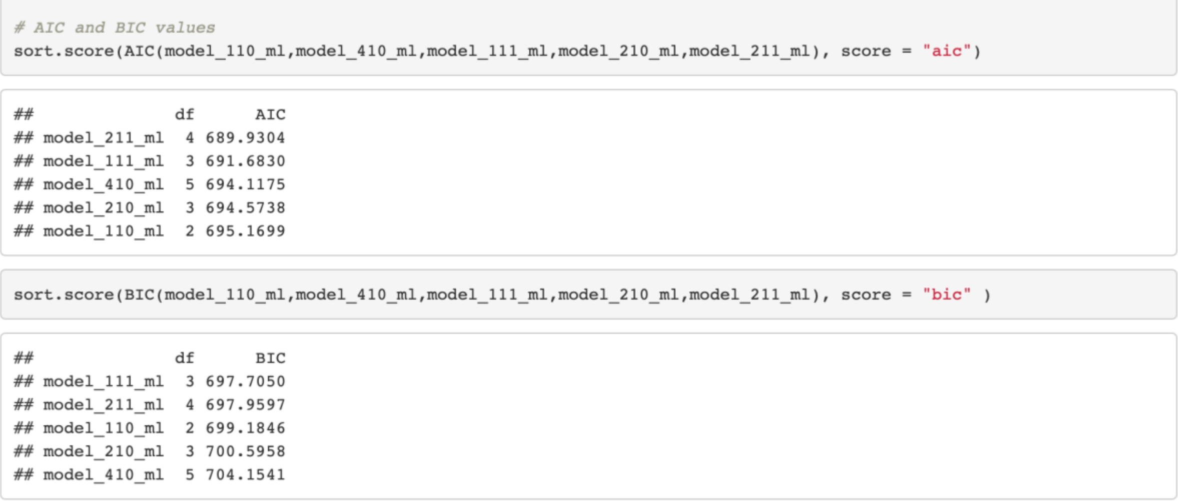
```
model_410_ml = Arima(carbon_TS,order=c(4,1,0),method='ML')
coeftest(model_410_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.514338   0.133830  3.8432 0.0001214 ***
## ar2  0.075388   0.145305  0.5188 0.6038838
## ar3  0.309462   0.143719  2.1532 0.0312991 *
## ar4 -0.105362   0.136162 -0.7738 0.4390505
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 23 CSS and ML model fitting of ARIMA(4,1,0)

3.3.6 AIC and BIC values

Based on the AIC value shown in Figure 24, ARIMA {2,1,1} is the best model and ARIMA {1,1,1} is the second best model. Both of the models were considered as candidates for good models based on their significant coefficients. According to the BIC values, ARIMA {1,1,1} is the best model and ARIMA {2,1,1} is the second best model. Once again both the models were good model candidates. It is also interesting to note that the same two models are considered the top two models in both AIC and BIC.



```
# AIC and BIC values
sort.score(AIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml), score = "aic")
```

```
##              df      AIC
## model_211_ml  4 689.9304
## model_111_ml  3 691.6830
## model_410_ml  5 694.1175
## model_210_ml  3 694.5738
## model_110_ml  2 695.1699
```

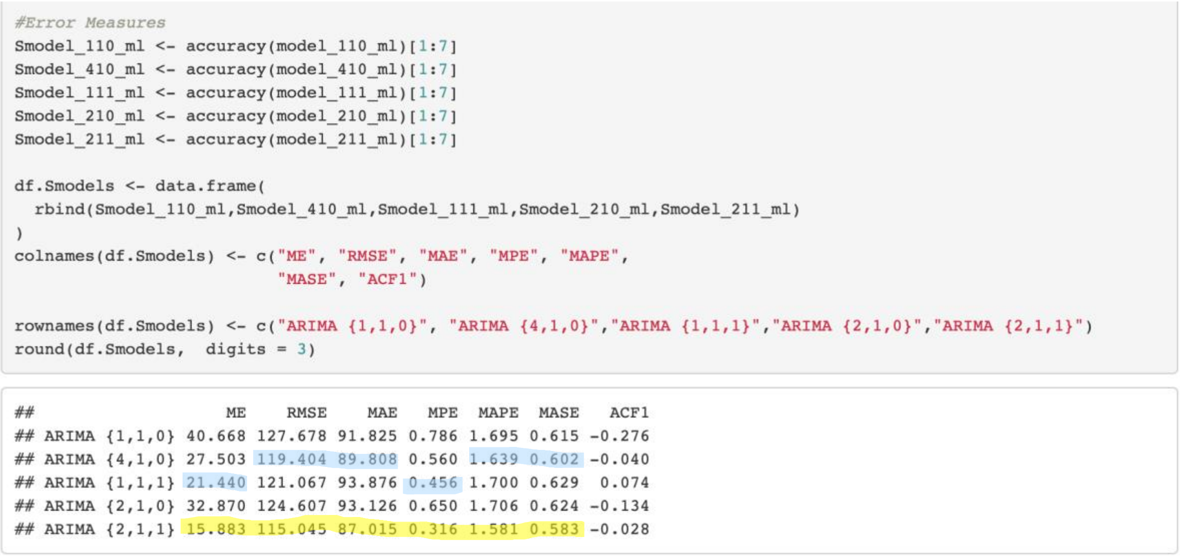
```
sort.score(BIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml), score = "bic" )
```

```
##              df      BIC
## model_111_ml  3 697.7050
## model_211_ml  4 697.9597
## model_110_ml  2 699.1846
## model_210_ml  3 700.5958
## model_410_ml  5 704.1541
```

Figure 24 AIC and BIC values of all the models

3.3.6 Error Measures

Figure 25, shows the results of the error measures for all the shortlisted models. It can be noticed that ARIMA{2,2,1} has the lowest error values across all the error measures (highlighted in yellow). The second best model based on the error measure will ARIMA{4,1,0} followed by ARIMA {1,1,1}.



```
#Error Measures
Smodel_110_ml <- accuracy(model_110_ml)[1:7]
Smodel_410_ml <- accuracy(model_410_ml)[1:7]
Smodel_111_ml <- accuracy(model_111_ml)[1:7]
Smodel_210_ml <- accuracy(model_210_ml)[1:7]
Smodel_211_ml <- accuracy(model_211_ml)[1:7]

df.Smodels <- data.frame(
  rbind(Smodel_110_ml,Smodel_410_ml,Smodel_111_ml,Smodel_210_ml,Smodel_211_ml)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")

rownames(df.Smodels) <- c("ARIMA {1,1,0}", "ARIMA {4,1,0}","ARIMA {1,1,1}","ARIMA {2,1,0}","ARIMA {2,1,1}")
round(df.Smodels,  digits = 3)
```

```
##                   ME    RMSE    MAE   MPE  MAPE  MASE   ACF1
## ARIMA {1,1,0} 40.668 127.678 91.825 0.786 1.695 0.615 -0.276
## ARIMA {4,1,0} 27.503 119.404 89.808 0.560 1.639 0.602 -0.040
## ARIMA {1,1,1} 21.440 121.067 93.876 0.456 1.700 0.629  0.074
## ARIMA {2,1,0} 32.870 124.607 93.126 0.650 1.706 0.624 -0.134
## ARIMA {2,1,1} 15.883 115.045 87.015 0.316 1.581 0.583 -0.028
```

Figure 25 Error measures of all the models

3.3.7 Model Selection

In summary ARIMA {1,1,0}, ARIMA {1,1,1} and ARIMA {2,1,1} have all significant coefficients. ARIMA {1,1,1} and ARIMA {2,1,1} have high AIC and BIC values are the top two models. ARIMA {2,1,1} has the lowest values amongst all the error measures. Based on these observation ARIMA {2,1,1} seems to be the best model. The summary is provided in the table in Figure 26.

The next best model will be ARIMA {1,1,1}, with all significant coefficient, high ranking in both BIC and AIC values, low error ranking in ME and MPE error measures.

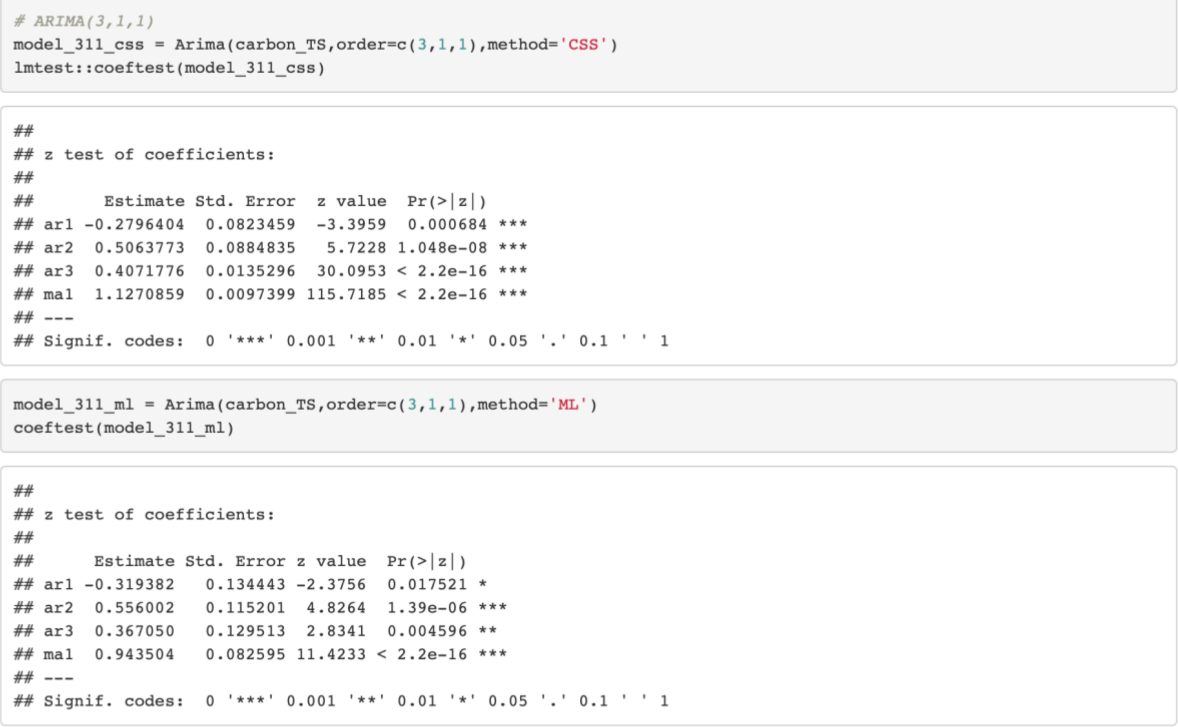
Models	**AIC Ranking**	**BIC Ranking**	**Error Measures**	**Significant coefficients**
ARIMA {1,1,0}	5	3		1 out of 1
ARIMA {4,1,0}	3	5		2 out of 4
ARIMA {1,1,1}	2	1		2 out of 2
ARIMA {2,1,0}	4	4		1 out of 2
ARIMA {2,1,1}	1	2	lowest on all measures	3 out of 3

Figure 26 Table showing a summary of the goodness of fit of the models

3.3.8 Over parameterised Models

To further assess the selected model ARIMA(2,1,1), overfitting has been done and the new models are ARIMA(3,1,1) and ARIMA(2,1,2).

Figure 27 shows the model fitting of ARIMA {3,1,1}. Both CC and ML returns all significant coefficients for the model. We have a new potential model to consider.



```
# ARIMA(3,1,1)
model_311_css = Arima(carbon_TS,order=c(3,1,1),method='CSS')
lmtest::coeftest(model_311_css)
```

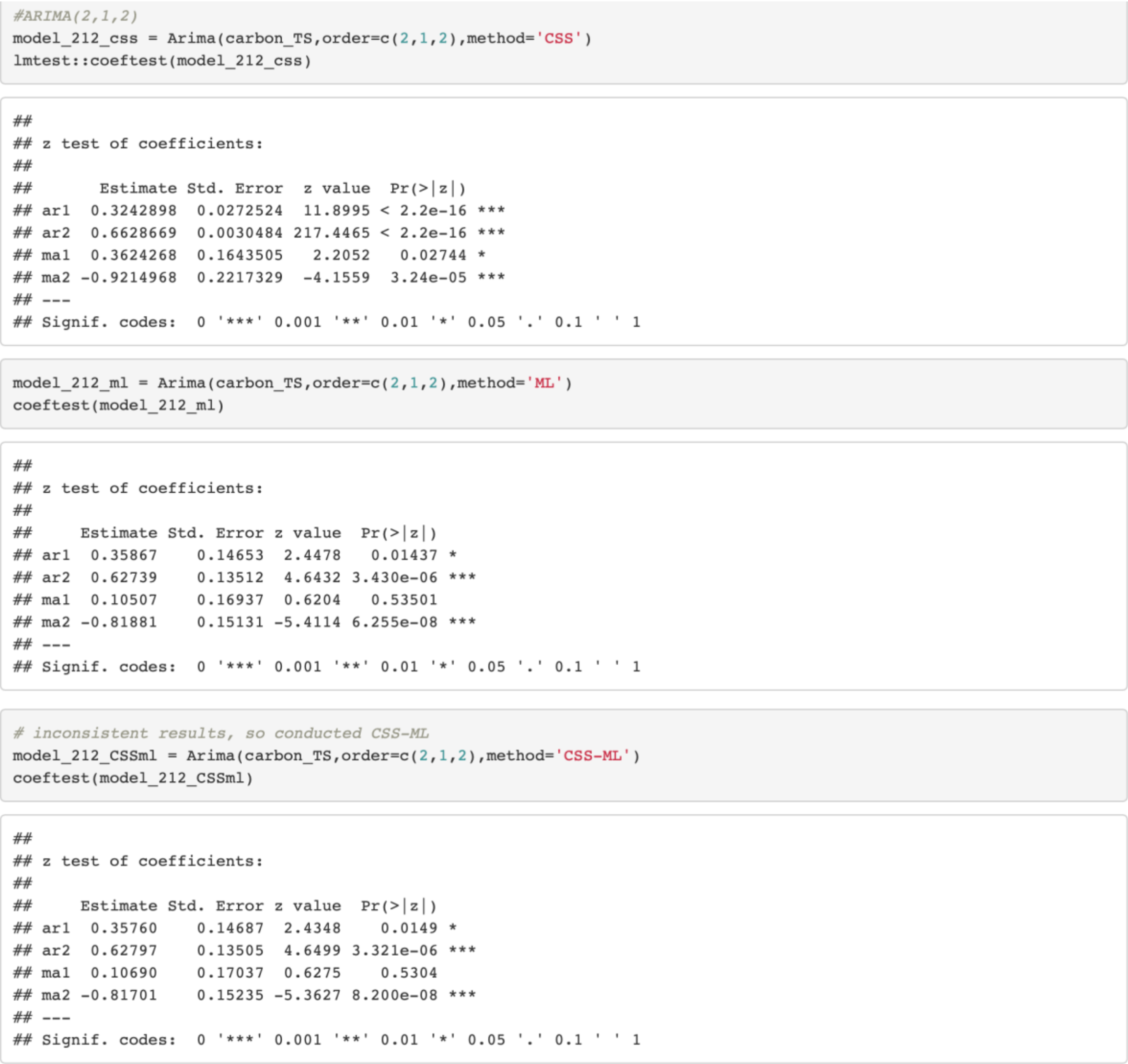
```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.2796404  0.0823459  -3.3959  0.000684 ***
## ar2  0.5063773  0.0884835   5.7228 1.048e-08 ***
## ar3  0.4071776  0.0135296  30.0953 < 2.2e-16 ***
## ma1  1.1270859  0.0097399 115.7185 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_311_ml = Arima(carbon_TS,order=c(3,1,1),method='ML')
coeftest(model_311_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.319382   0.134443 -2.3756  0.017521 *
## ar2  0.556002   0.115201  4.8264  1.39e-06 ***
## ar3  0.367050   0.129513  2.8341  0.004596 **
## ma1  0.943504   0.082595 11.4233 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 27: model fitting of ARIMA {3,1,1}

The parameter estimation for ARIMA {2,1,2} is given in Figure 28. CSS model returns all significant coefficients, but the ML model returns only 3 significant coefficients. Since the result in inconsistent, CSS ML method has ben performed, which resulted in 3 out 4 significant coefficients. This model is therefore not a good one.



```
#ARIMA(2,1,2)
model_212_css = Arima(carbon_TS,order=c(2,1,2),method='CSS')
lmtest::coeftest(model_212_css)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.3242898  0.0272524  11.8995 < 2.2e-16 ***
## ar2  0.6628669  0.0030484 217.4465 < 2.2e-16 ***
## ma1  0.3624268  0.1643505   2.2052   0.02744 *
## ma2 -0.9214968  0.2217329  -4.1559  3.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_212_ml = Arima(carbon_TS,order=c(2,1,2),method='ML')
coeftest(model_212_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.35867    0.14653  2.4478   0.01437 *
## ar2  0.62739    0.13512  4.6432 3.430e-06 ***
## ma1  0.10507    0.16937  0.6204   0.53501
## ma2 -0.81881    0.15131 -5.4114 6.255e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

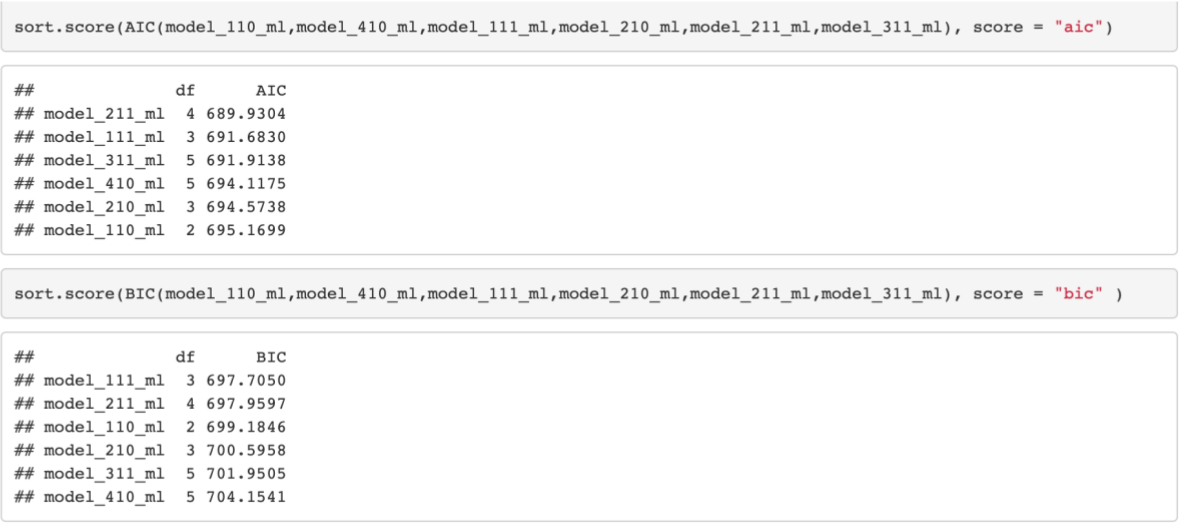
```
# inconsistent results, so conducted CSS-ML
model_212_CSSml = Arima(carbon_TS,order=c(2,1,2),method='CSS-ML')
coeftest(model_212_CSSml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.35760    0.14687  2.4348    0.0149 *
## ar2  0.62797    0.13505  4.6499 3.321e-06 ***
## ma1  0.10690    0.17037  0.6275    0.5304
## ma2 -0.81701    0.15235 -5.3627 8.200e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 28: model fitting of ARIMA {2,1,2}

Since ARIMA{3,1,1} resulted in all significant coefficient, we must re-run the model fitting steps to check if its a good fit.

The new AIC and BIC table is as follows in Figure 29. Among the AIC value ARIMA{3,1,1} ranks as the third best model, however in the BIC values ARIMA{3,1,1} is on the higher side and wouldn't be considered a good model.



```
sort.score(AIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml,model_311_ml), score = "aic")
```

```
##              df      AIC
## model_211_ml  4 689.9304
## model_111_ml  3 691.6830
## model_311_ml  5 691.9138
## model_410_ml  5 694.1175
## model_210_ml  3 694.5738
## model_110_ml  2 695.1699
```

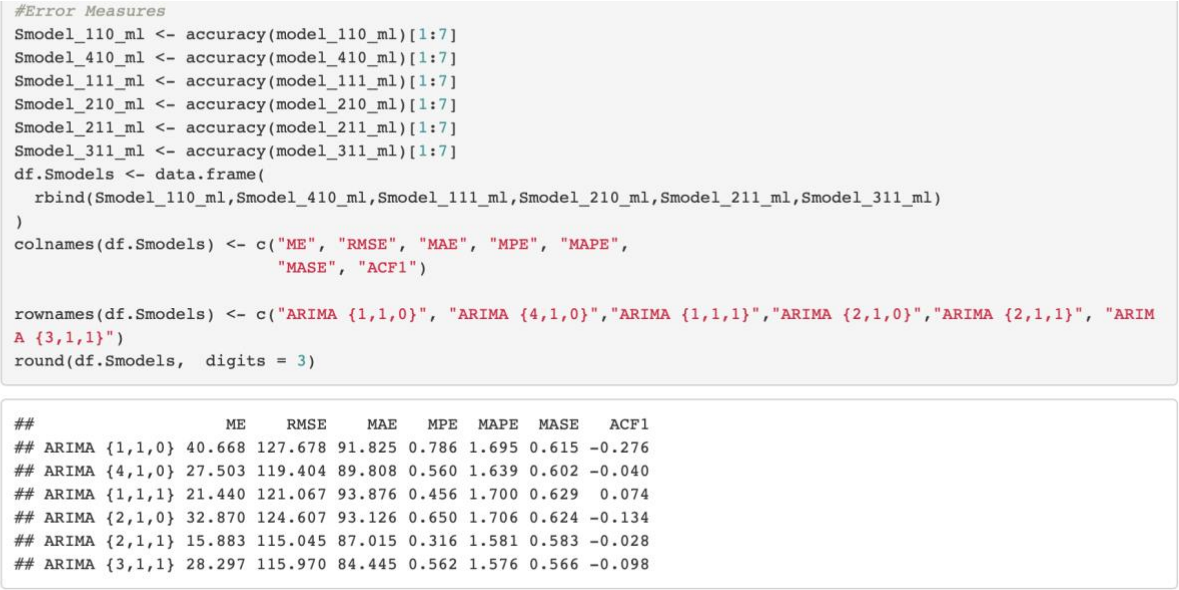
```
sort.score(BIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml,model_311_ml), score = "bic" )
```

```
##              df      BIC
## model_111_ml  3 697.7050
## model_211_ml  4 697.9597
## model_110_ml  2 699.1846
## model_210_ml  3 700.5958
## model_311_ml  5 701.9505
## model_410_ml  5 704.1541
```

Figure 29 AIC and BIC for ARIMA{3,1,1}

The error measures of the model shows a different story. ARIMA{3,1,1} has the lowest values in 3 of the error measures. However, ARIMA{2,1,1} still would be considered the best since it has quite low values across all the 6 error measures.

The suitability of these models can then be confirmed using the residual analysis in the next step.



```
#Error Measures
Smodel_110_ml <- accuracy(model_110_ml)[1:7]
Smodel_410_ml <- accuracy(model_410_ml)[1:7]
Smodel_111_ml <- accuracy(model_111_ml)[1:7]
Smodel_210_ml <- accuracy(model_210_ml)[1:7]
Smodel_211_ml <- accuracy(model_211_ml)[1:7]
Smodel_311_ml <- accuracy(model_311_ml)[1:7]
df.Smodels <- data.frame(
  rbind(Smodel_110_ml,Smodel_410_ml,Smodel_111_ml,Smodel_210_ml,Smodel_211_ml,Smodel_311_ml)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")

rownames(df.Smodels) <- c("ARIMA {1,1,0}", "ARIMA {4,1,0}","ARIMA {1,1,1}","ARIMA {2,1,0}","ARIMA {2,1,1}", "ARIM
A {3,1,1}")
round(df.Smodels,  digits = 3)
```

```
##                   ME    RMSE    MAE   MPE  MAPE  MASE   ACF1
## ARIMA {1,1,0} 40.668 127.678 91.825 0.786 1.695 0.615 -0.276
## ARIMA {4,1,0} 27.503 119.404 89.808 0.560 1.639 0.602 -0.040
## ARIMA {1,1,1} 21.440 121.067 93.876 0.456 1.700 0.629  0.074
## ARIMA {2,1,0} 32.870 124.607 93.126 0.650 1.706 0.624 -0.134
## ARIMA {2,1,1} 15.883 115.045 87.015 0.316 1.581 0.583 -0.028
## ARIMA {3,1,1} 28.297 115.970 84.445 0.562 1.576 0.566 -0.098
```

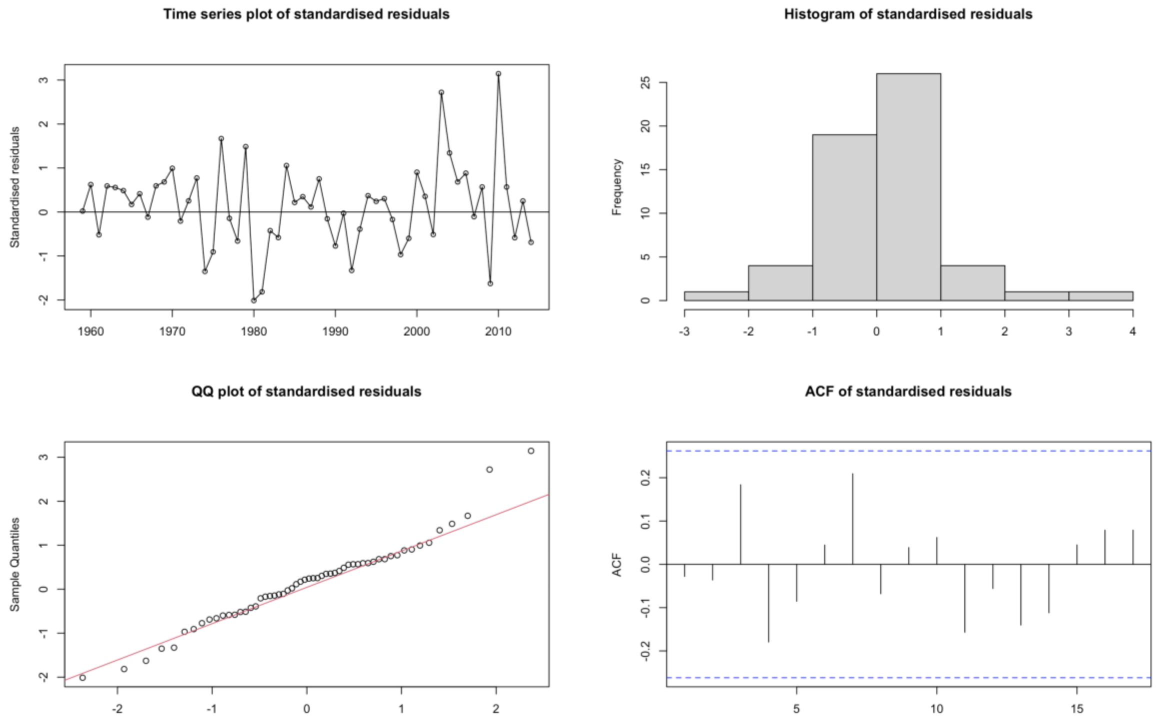
Figure 30 Error Measures with the new model ARIMA{3,1,1}

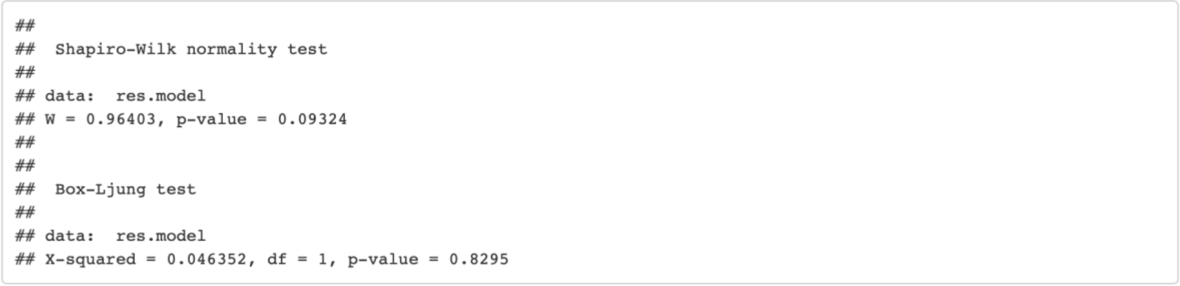
3.4 Model Diagnostics

3.4.1 ARIMA {2,1,1}

Figure 31 shows the snapshot of the residual analysis for ARIMA {2,1,1}. The Time Series model of the residuals shows that there is no trend, no seasonality or variance. The point in

2010 could maybe be a change point. The histogram looks almost mainly symmetric; however, the value exceeds +3 indicating possible outliers. The Shapiro Wilks test indicates that the data is normal with a p-value greater the 0.05. This is confirmed by the QQ plot that has most of the points close to the QQline. The ACF plot (bottom right) doesn't have any significant lags which indicates that the model has captured all the auto correlation. This is further confirmed with the Ljung-Box test, with all points above the line and the p-value is greater than 0.05.





```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.96403, p-value = 0.09324
##
##
##  Box-Ljung test
##
## data:  res.model
## X-squared = 0.046352, df = 1, p-value = 0.8295
```

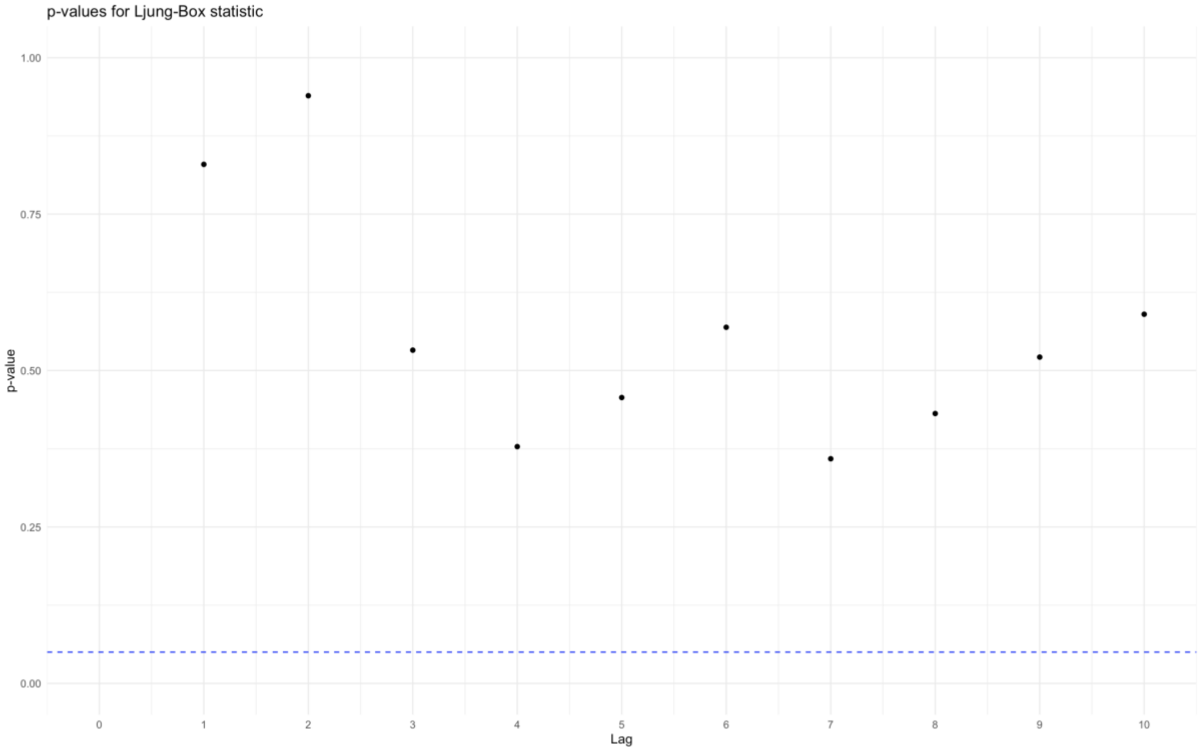
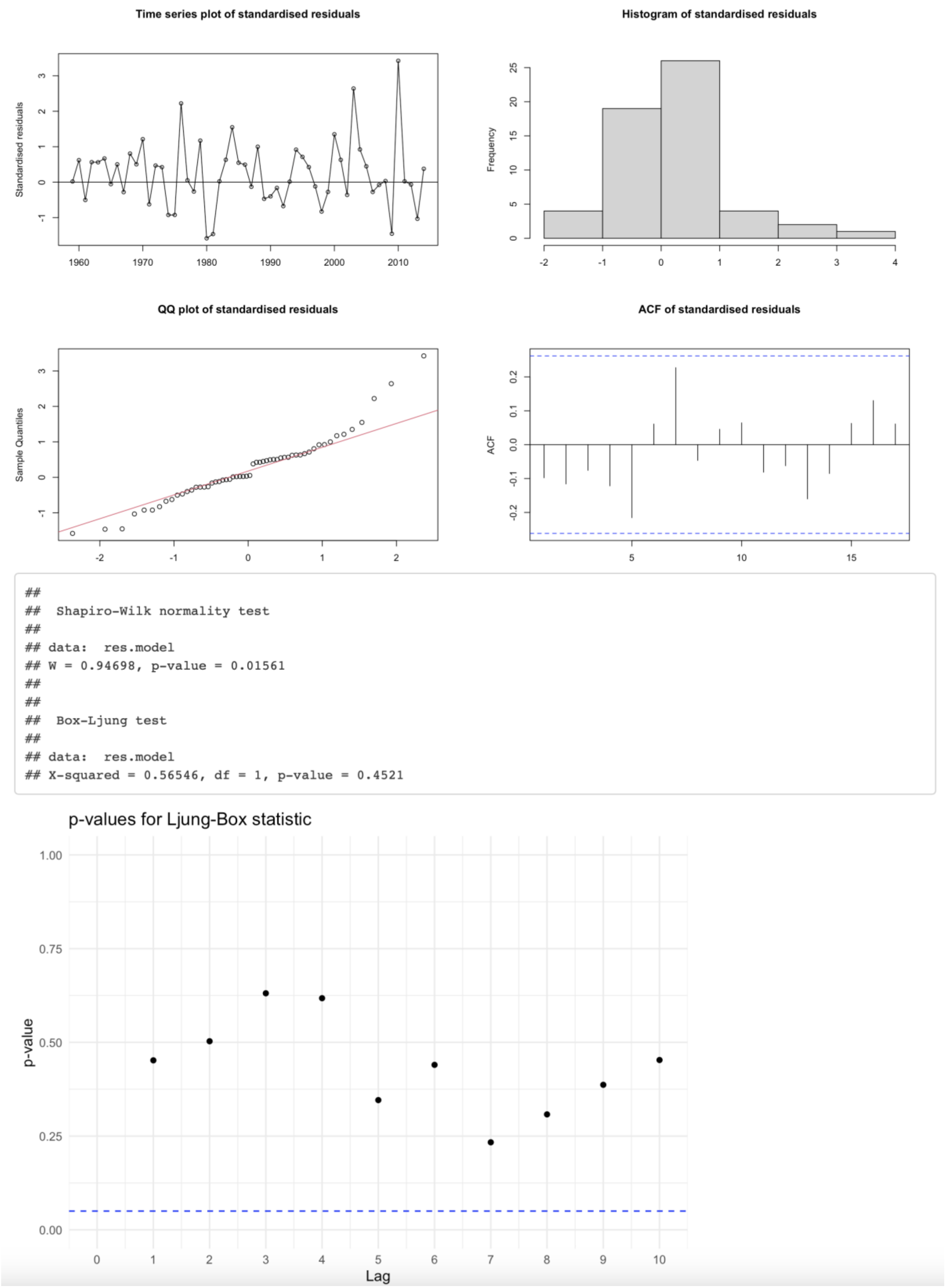


Figure 31: Residual Analysis of model (2,1,1)

3.4.2 ARIMA {3,1,1} - ML

Figure 32 shows the snapshot of the residual analysis for ARIMA {3,1,1} ML method. The Time Series model of the residual shows that there is no trend, no seasonality or variance. The point in 2010 could maybe be a change point. The histogram doesn’t look symmetric. There seems to be a right tail indicating a left skewness and the value of the histogram exceeds +3 indicating possible outliers. The QQ plot that has a significant number od data points both ends of the plot that are away from the QQ line indicating that the data is not normal. This is further confirmed by the Shapiro Wilks test with a p-value less than 0.05 indicating that the series is not normal. The ACF plot (bottom right) doesn’t have any significant lags which indicates that the model has captured all the auto correlation. This is further confirmed with the Ljung-Box test, with all points above the line and the p-value is greater than 0.05.


Time series plot of standardised residuals
Standardised residuals
-1
0
1
2
3
1960
1970
1980
1990
2000
2010
Histogram of standardised residuals
Frequency
0
5
10
15
20
25
-2
-1
0
1
2
3
4
QQ plot of standardised residuals
Sample Quantiles
-2
-1
0
1
2
ACF of standardised residuals
ACF
-0.2
-0.1
0.0
0.1
0.2
5
10
15
p-values for Ljung-Box statistic
p-value
0.00
0.25
0.50
0.75
1.00
0
1
2
3
4
5
6
7
8
9
10
Lag

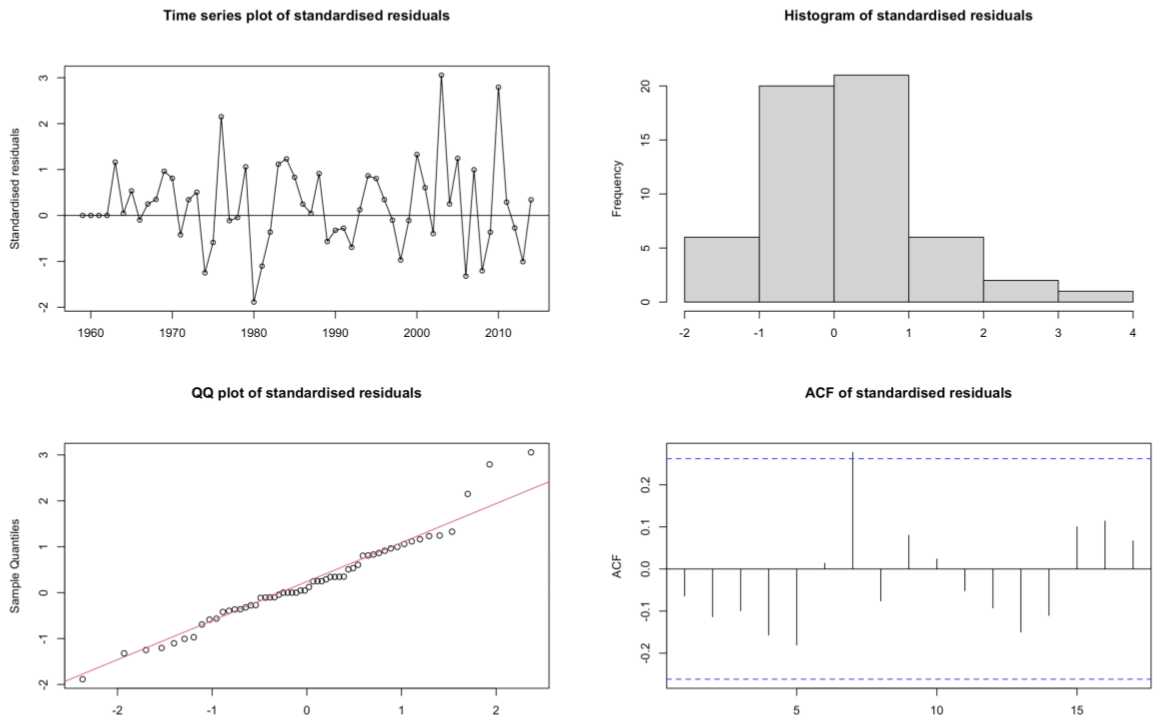
```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.94698, p-value = 0.01561
##
##
##  Box-Ljung test
##
## data:  res.model
## X-squared = 0.56546, df = 1, p-value = 0.4521
```

Figure 32: Residual Analysis of model (3,1,1 - ML)

3.4.3 ARIMA {3,1,1} - CSS

Since the previous model was not normal and the ML method has an assumption of normality, we can consider the residual analysis of the new model based on the CSS method. Figure 33 shows the snapshot of the residual analysis for ARIMA {3,1,1} CSS method. The Time Series model of the residual shows that there is no trend, no seasonality or variance. The point in 2010 could maybe be a change point. The histogram (top right) is not symmetric with a right tail and values greater than +3 indicating outliers. The QQ plot looks like the series is normal with most of the points aligning with the QQ line except for the ones in the right side of the plot. This is further confirmed by the Shapiro Wilks test with a p-value greater than 0.05. The ACF plot also shows that there is a significant lag between 5 and 10 years, which mean that the model hasn’t fully captured all the auto correlations, however the L-jung box doesn’t show any points below the line and has a p-value greater than 0.05 indicating that the lag is not really significant.

```
residual.analysis(model = model_311_css) # New model
```



```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.96235, p-value = 0.07792
##
##
##  Box-Ljung test
##
## data:  res.model
## X-squared = 0.24233, df = 1, p-value = 0.6225
```

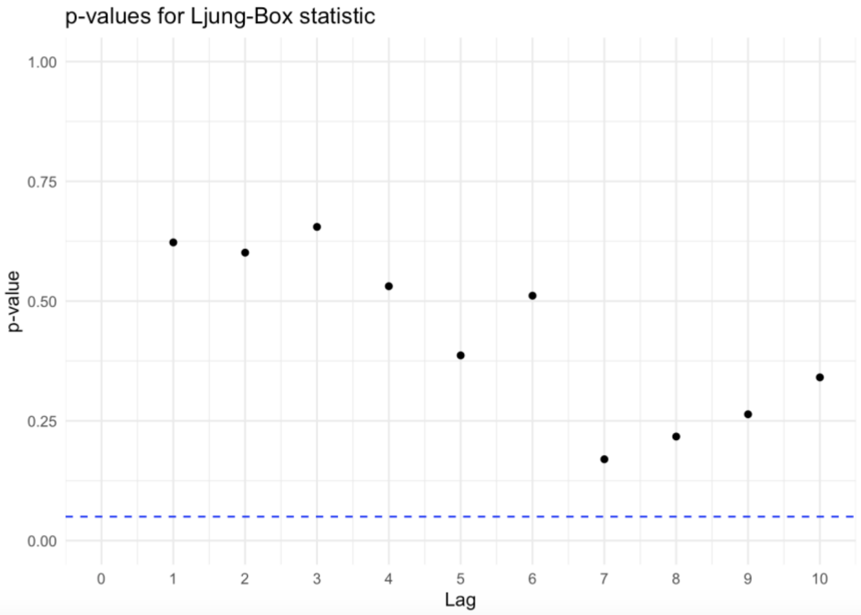


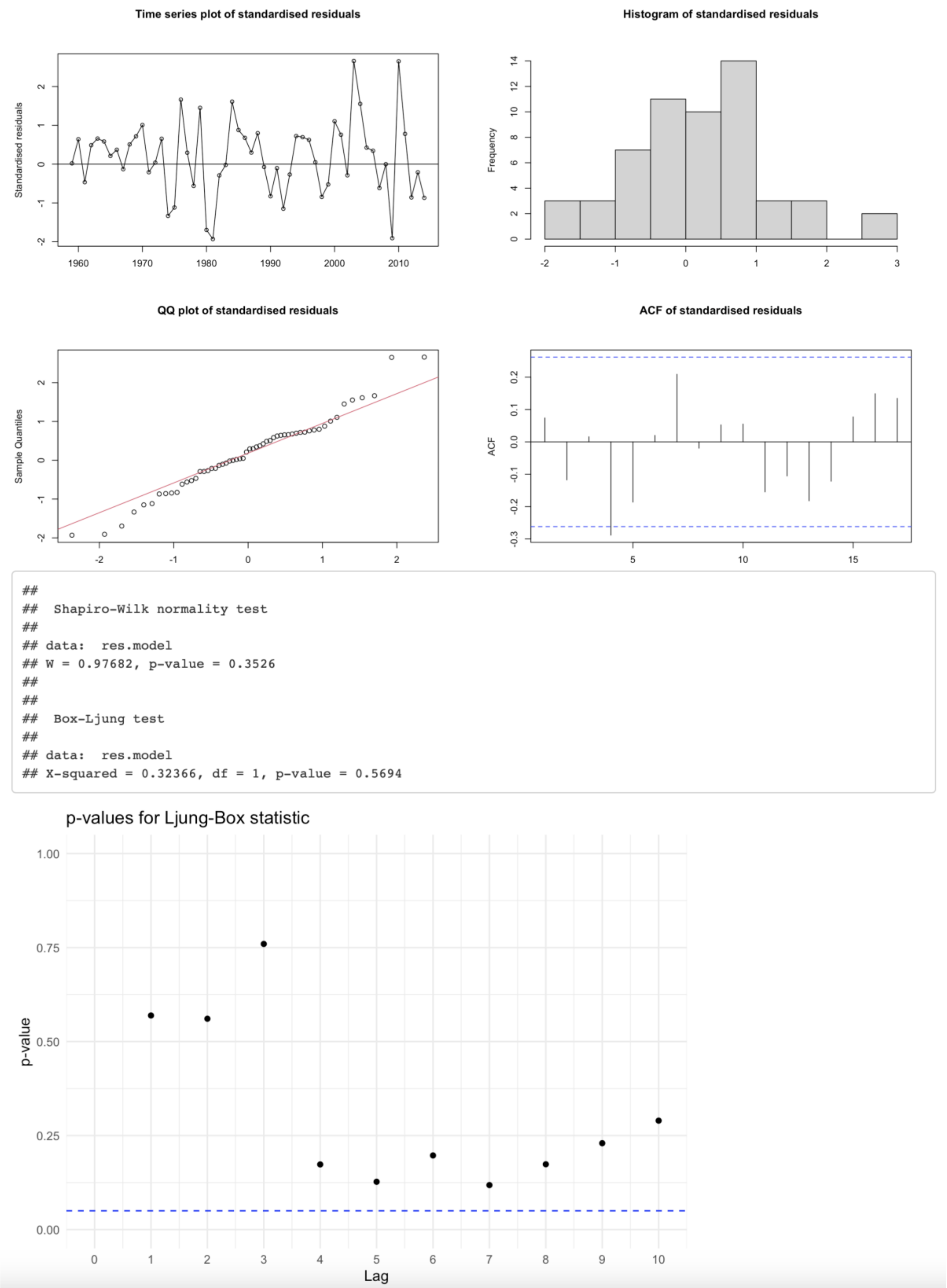
Figure 33: Residual Analysis of model (3,1,1)- CSS

From the above analysis of the three chosen models, ARIMA {2,1,1} seems to be the best fitting model, with a comparatively symmetric histogram, normality and no significant lags in the ACF compared to the newly added model of ARIMA{3,1,1}. The p-value in Box-Ljung test is highest for model (2,1,1), so is the significance for normality. Also based on the principal of parsimony, ARIMA {2,1,2} is a better fit. However, since the histogram had values over +3, we can test out the next best model i.e. ARIM. 1,1,1}

3.4.4 ARIMA {1,1,1}

Figure 34 shows the residual analysis for ARIMA {1,1,1}. Time Series model of the residual shows that there is no trend, no seasonality or variance. The point in 2010 could maybe be a change point. The histogram could be symmetrical however there is a gap between 2 and 2.5, the values lie between -3 and +3. The QQ plot indicates that most of the points align well with the QQ line except for a few in the starting and the end of the plot. Shapiro-Wilk test confirms that it is a normal distribution with a p-value greater than 0.05. ACF shows one significant lag between 0 and 5, however there are no points below the line in the Ljung Box test and the p-value is also greater than 0.05, indicating that the lag is not actually significant.

```
residual.analysis(model = model_111_ml) # testing
```


Time series plot of standardised residuals
Standardised residuals
1960
1970
1980
1990
2000
2010
Histogram of standardised residuals
Frequency
QQ plot of standardised residuals
Sample Quantiles
ACF of standardised residuals
ACF
p-values for Ljung-Box statistic
p-value
Lag

```
##
##  Shapiro-Wilk normality test
##
## data:  res.model
## W = 0.97682, p-value = 0.3526
##
##
##  Box-Ljung test
##
## data:  res.model
## X-squared = 0.32366, df = 1, p-value = 0.5694
```

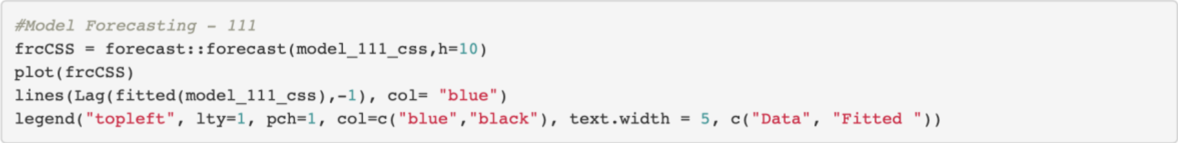
Figure 34: Residual Analysis of model (1,1,1)

The residual analysis has also been conducted for the other model (not included in the report as they were not good models) to test the models. Model (2,1,0) and (4,1,0) have insignificant coefficients and rank low in AIC and BIC values. Shapiro-Wilk test has borderline p-value in both cases and the histograms are skewed with values over +3. In model (2,1,0), though the p-value is above 0.05, there are points touch the line in the plot. Model (1,1,0) doesn't catch all the autocorrelations and and has significant lags in ACF that is confirmed by the Ljung Box test.

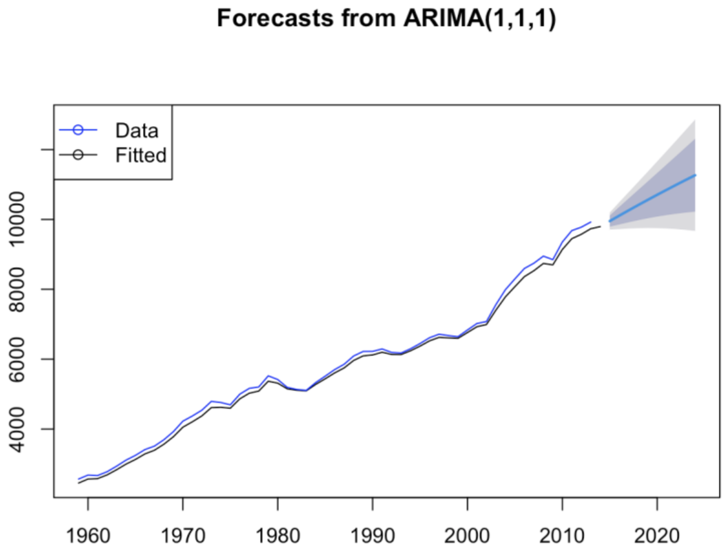
3.5 Model Forecasting

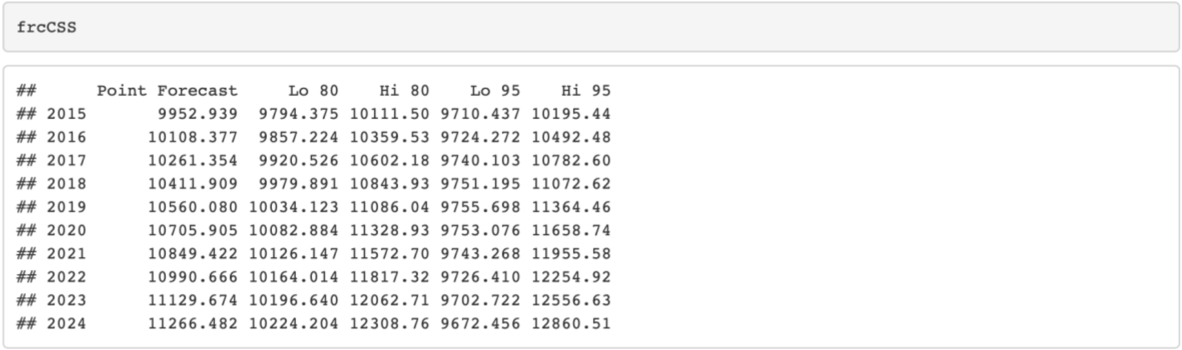
Based of the analysis above ARIMA{1,1,1} and ARIMA{2,1,1} are good candidates to fit the series. However, ARIMA{1,1,1} seems like a better fit with the bars in the histogram within +3 and -3, the residuals are normal and has a higher p-value of 0.35 in Shapiro Wilk's test compared to the 0.09 in model(2,1,1). The auto correlations are captured by the model with no significant lags according to the Ljung Box test. The time series plot of model (1,1,1) is also plotted on a smaller scale of (-2,2) compared to model (2,1,1)'s scale of (-2,3). Moreover, to adhere to the principal of parsimony, ARIMA{1,1,1} is a smaller model with 2 parameters compared to 3 parameters in ARIMA{2,1,1}. Overall, ARIMA{1,1,1}seems like a better fit and will be used for forecasting.

Forecasting has been done using the the ARIMA (1,1,1) model from CSS method, as shown in Figure 35



```
#Model Forecasting - 111
frcCSS = forecast::forecast(model_111_css,h=10)
plot(frcCSS)
lines(Lag(fitted(model_111_css),-1), col= "blue")
legend("topleft", lty=1, pch=1, col=c("blue","black"), text.width = 5, c("Data", "Fitted "))
```



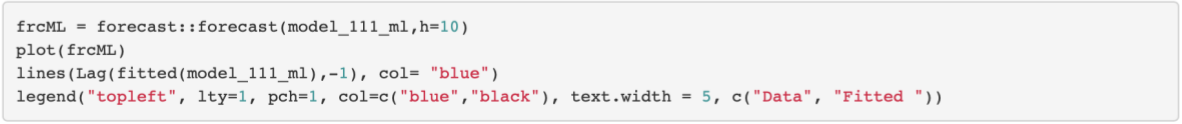


```
frcCSS
```

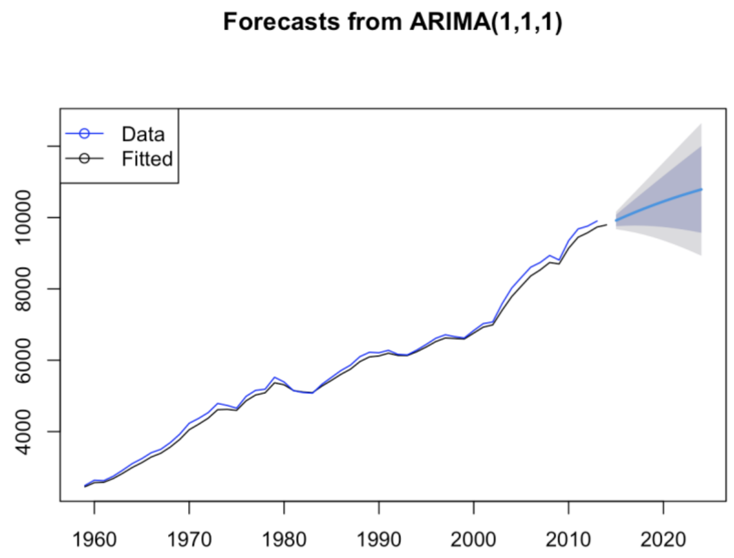
```
##      Point Forecast       Lo 80    Hi 80    Lo 95    Hi 95
## 2015       9952.939  9794.375 10111.50 9710.437 10195.44
## 2016      10108.377  9857.224 10359.53 9724.272 10492.48
## 2017      10261.354  9920.526 10602.18 9740.103 10782.60
## 2018      10411.909  9979.891 10843.93 9751.195 11072.62
## 2019      10560.080 10034.123 11086.04 9755.698 11364.46
## 2020      10705.905 10082.884 11328.93 9753.076 11658.74
## 2021      10849.422 10126.147 11572.70 9743.268 11955.58
## 2022      10990.666 10164.014 11817.32 9726.410 12254.92
## 2023      11129.674 10196.640 12062.71 9702.722 12556.63
## 2024      11266.482 10224.204 12308.76 9672.456 12860.51
```

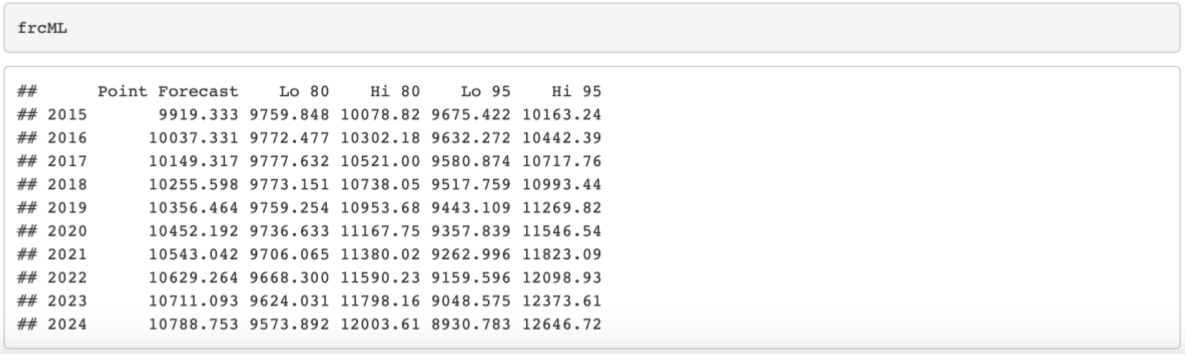
Figure 35 Forecasting for ARIMA (1,1,1) model from CSS method

Forecasting for ARIMA (1,1,1) model as per the ML method is given in Figure 36.



```
frcML = forecast::forecast(model_111_ml,h=10)
plot(frcML)
lines(Lag(fitted(model_111_ml),-1), col= "blue")
legend("topleft", lty=1, pch=1, col=c("blue","black"), text.width = 5, c("Data", "Fitted "))
```





```
frcML
```

```
##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2015       9919.333 9759.848 10078.82 9675.422 10163.24
## 2016      10037.331 9772.477 10302.18 9632.272 10442.39
## 2017      10149.317 9777.632 10521.00 9580.874 10717.76
## 2018      10255.598 9773.151 10738.05 9517.759 10993.44
## 2019      10356.464 9759.254 10953.68 9443.109 11269.82
## 2020      10452.192 9736.633 11167.75 9357.839 11546.54
## 2021      10543.042 9706.065 11380.02 9262.996 11823.09
## 2022      10629.264 9668.300 11590.23 9159.596 12098.93
## 2023      10711.093 9624.031 11798.16 9048.575 12373.61
## 2024      10788.753 9573.892 12003.61 8930.783 12646.72
```

Figure 36 Forecasting for ARIMA (1,1,1) model from NL method

The forecasts from the CSS model is slightly higher than the ML model. Since the data is normal, we can consider the forecasts from the ML model. Either way, the fitted line model is following the time series plot, indicating that the model is a good fit.

4. Conclusion

From the above Model fitting and residual analysis, it is evident that ARIMA{1,1,1} is the best model that fits the time series. The forecasts as per the ML method shows that the carbon emission is expected to increase over the years to an estimated 10788 MTC/year in 2024.

5.Appendix

```
library(TSA)
library(fUnitRoots)
library(lmtest)
library(tseries)
library(readr)
library(forecast)
library(Hmisc)

#adding the sort function
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")')
  }
}

residual.analysis <- function(model, std = TRUE,start = 2, class = "ARIMA"[1]){
  library(TSA)
  library(LSTS)
  if (class == "ARIMA"){
    if (std == TRUE){
      res.model = rstandard(model)
    }else{
      res.model = residuals(model)
    }
  }else if (class == "GARCH"){
    res.model = model$residuals[start:model$n.used]
  }else if (class == "ARMA-GARCH"){
    res.model = model@fit$residuals
  }else if (class == "fGARCH"){
    res.model = model@residuals
  }else {
    stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
  }
  par(mfrow=c(2,2))
  plot(res.model,type='o',ylab='Standardised residuals', main="Time series plot of
standardised residuals")
  abline(h=0)
  hist(res.model,main="Histogram of standardised residuals")
  qqnorm(res.model,main="QQ plot of standardised residuals")
  qqline(res.model, col = 2)
  acf(res.model,main="ACF of standardised residuals")
  Box.Ljung.Test(res.model)
  print(shapiro.test(res.model))
  print(Box.test(res.model, type = "Ljung-Box"))
  print(Box.Ljung.Test(res.model))
  par(mfrow=c(1,1))
}

#Data Preprocessing
carbon <-read_csv("~/Downloads/Book6.csv")
class(carbon)
summary(carbon)
plot(carbon,type='o',ylab='carbon for electricity')

# Convert to the TS object!
carbon_TS <- ts(as.vector(carbon$Total),start=1959, end=2014)
class(carbon_TS)
summary(carbon_TS)

plot(carbon_TS,type='o',ylab='Fossil fuel emissions',
     main = " Time series plot of annual Fossil fuel emissions")

#Lag - checking the impact of previous year's Fossil fuel emissions on the next year's
emissions
par(mfrow=c(1,2))
y = carbon_TS
x = zlag(carbon_TS) # generate the first lag of the Fossil fuel emissions time series
head(y)
head(x)
index = 2:length(x) # Create an index to get rid of the first NA value in x
cor(y[index],x[index])
plot(y[index],x[index],ylab='Fossil fuel emissions series', xlab='The first lag of Fossil fuel
emissions series',
     main = "Scatter plot of the series with first lag")

# looking at the second lag
x = zlag(zlag(carbon_TS))
index = 3:length(x)
cor(y[index],x[index])

plot(y[index],x[index],ylab='Fossil fuel emissions series', xlab='The second lag of Fossil fuel
emissions series',
     main = "Scatter plot of the series with second lag")
par(mfrow=c(1,1))
```

```
#Displying the ACF
acf(carbon_TS, lag.max = 60,main = "ACF of Fossil fuel emissions series")

#3.2 Model Specification

#ACF and PACF plots
par(mfrow=c(1,2))
acf(carbon_TS, main ="ACF plot of Fossil fuel emissions series.")
pacf(carbon_TS, main ="PACF plot of Fossil fuel emissions series.")
par(mfrow=c(1,1))

#slowly decaying trend in ACF and one significant lag in PACF indicates a trend and
possible seasonality
#also indicates that the series is non-stationary

#checking for normality
qqnorm(carbon_TS, ylab="Fossil fuel emissions", xlab="Normal Scores")
qqline(carbon_TS, col = 2)
shapiro.test(carbon_TS) #Normal Data

#shapiro wilk test p-value is greater than 0.05 so it is normal, however the plot shows a
couple of datapoints straying away from the QQ line

#Running a Box-Cox Transformation
BC = BoxCox.ar(carbon_TS)
BC$ci
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda #Since lambda is positive but close to 0, so its near log transformation
BC.carbon_TS = (carbon_TS^lambda-1)/lambda

plot(BC.carbon_TS,type='o',ylab='Fossil fuel emissions',
    main = " Time series plot of BC transformed Fossil fuel emissions series")

#Since there was not a lot of changing variance the BC transformation didnt make much of a
difference in the TS plot

#checking for normality
qqnorm(BC.carbon_TS, ylab="Fossil fuel emissions", xlab="Normal Scores")
qqline(BC.carbon_TS, col = 2)
shapiro.test(BC.carbon_TS)

#The normality on the BC transformed data showed a slightly lower p-value. So I have
decided to go ahead with the raw data for differencing and detrending the data.

adf.test(carbon_TS) #therefore non-stationary

diff.carbon_TS = diff(carbon_TS,differences = 1)
plot(diff.carbon_TS,type='o',ylab='Fossil fuel emissions series',
    main ="Time series plot of the first difference of
```

```
     Fossil fuel emissions series")

#The plot looks better than the original time series plot. the series seems to have been
detrended

# applying the tests to the differenced series.
adf.test(diff.carbon_TS) #stationary
kpss.test(diff.carbon_TS) # Stationary
pp.test(diff.carbon_TS) #Stationary

par(mfrow=c(1,2))
acf(diff.carbon_TS, main ="ACF plot of the first difference of
     Fossil fuel emissions series", lag.max = 60)
pacf(diff.carbon_TS, main ="PACF plot of the first difference of
    Fossil fuel emissions series", lag.max = 60)
par(mfrow=c(1,1))

#p =1, q=1, d=1 Possible model from ACF and PACF ARIMA{1,1,1}.
#the significant lag in PACF after 20 is not considered as it is a late lag

#identifying possible models
eacf(diff.carbon_TS)

#possible models from EACF
#ARIMA {1,1,0}, ARIMA {1,1,1},ARIMA {2,1,0},ARIMA {2,1,1}

res = armasubsets(y=diff.carbon_TS,nar=5,nma=5,y.name='p',ar.method='ols')
plot(res)

#The best model from BIC Table is ARIMA {1,1,0} and ARIMA {4,1,0}

#The final set of possible models are :
# ARIMA {1,1,0}, ARIMA {4,1,0},ARIMA {1,1,1},ARIMA {2,1,0},ARIMA {2,1,1}

#3.3 Model Fitting

#using raw data
#ARIMA(1,1,0)
model_110_css = Arima(carbon_TS,order=c(1,1,0),method='CSS')
lmtest::coeftest(model_110_css)

model_110_ml = Arima(carbon_TS,order=c(1,1,0),method='ML')
coeftest(model_110_ml)

#All significant coefficients in both ML and CSS

#ARIMA{4,1,0}
model_410_css = Arima(carbon_TS,order=c(4,1,0),method='CSS')
lmtest::coeftest(model_410_css)
```

```
model_410_ml = Arima(carbon_TS,order=c(4,1,0),method='ML')
coeftest(model_410_ml)

#2 out of 4 significant coefficients in both ML and CSS

#ARIMA {1,1,1}
model_111_css = Arima(carbon_TS,order=c(1,1,1),method='CSS')
lmtest::coeftest(model_111_css)

model_111_ml = Arima(carbon_TS,order=c(1,1,1),method='ML')
coeftest(model_111_ml)

#All significant coefficients in both ML and CSS

#ARIMA {2,1,0}
model_210_css = Arima(carbon_TS,order=c(2,1,0),method='CSS')
lmtest::coeftest(model_210_css)

model_210_ml = Arima(carbon_TS,order=c(2,1,0),method='ML')
coeftest(model_210_ml)

# inconsistent results, so conducted CSS-ML
model_210_CSSml = Arima(carbon_TS,order=c(2,1,0),method='CSS-ML')
coeftest(model_210_CSSml)

#One of two insignificant coefficients

#ARIMA {2,1,1}
model_211_css = Arima(carbon_TS,order=c(2,1,1),method='CSS')
lmtest::coeftest(model_211_css)

model_211_ml = Arima(carbon_TS,order=c(2,1,1),method='ML')
coeftest(model_211_ml)

# contradicting results, so conducted CSS-ML
model_211_CSSml = Arima(carbon_TS,order=c(2,1,1),method='CSS-ML')
coeftest(model_211_CSSml)

#All significant coefficients in 2 out 3 tests


# AIC and BIC values
sort.score(AIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml
), score = "aic")
sort.score(BIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml
), score = "bic" )

# The ARIMA(1,1,1) and ARIMA(2,1,1) model are the best ones according to AIC and BIC

#Error Measures
```

```
Smodel_110_ml <- accuracy(model_110_ml)[1:7]
Smodel_410_ml <- accuracy(model_410_ml)[1:7]
Smodel_111_ml <- accuracy(model_111_ml)[1:7]
Smodel_210_ml <- accuracy(model_210_ml)[1:7]
Smodel_211_ml <- accuracy(model_211_ml)[1:7]

df.Smodels <- data.frame(
  rbind(Smodel_110_ml,Smodel_410_ml,Smodel_111_ml,Smodel_210_ml,Smodel_211_ml)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")

rownames(df.Smodels) <- c("ARIMA {1,1,0}", "ARIMA {4,1,0}","ARIMA
{1,1,1}","ARIMA {2,1,0}","ARIMA {2,1,1}")
round(df.Smodels,  digits = 3)

#ARIMA(2,1,1) model has the lowest error measures across all the different types of errors

# The best model out of all is ARIMA(2,1,1)

# Overfitting: To further assess the selected model ARIMA(2,1,1) by overfitting
# ARIMA(3,1,1) and ARIMA(2,1,2)

# ARIMA(3,1,1)
model_311_css = Arima(carbon_TS,order=c(3,1,1),method='CSS')
lmtest::coeftest(model_311_css)

model_311_ml = Arima(carbon_TS,order=c(3,1,1),method='ML')
coeftest(model_311_ml)

#All significant model, so we have an additional model to consider

#ARIMA(2,1,2)
model_212_css = Arima(carbon_TS,order=c(2,1,2),method='CSS')
lmtest::coeftest(model_212_css)

model_212_ml = Arima(carbon_TS,order=c(2,1,2),method='ML')
coeftest(model_212_ml)

# inconsistent results, so conducted CSS-ML
model_212_CSSml = Arima(carbon_TS,order=c(2,1,2),method='CSS-ML')
coeftest(model_212_CSSml)

# 3 out of 4 coefficients are significant

# Now I will consider ARIMA(3,1,1) among the others.

sort.score(AIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml
,model_311_ml), score = "aic")
```

```
sort.score(BIC(model_110_ml,model_410_ml,model_111_ml,model_210_ml,model_211_ml
,model_311_ml), score = "bic" )

#The AIC and BIC is not so great for the new model, we can conduct a residual analysis to
confirm
#Error Measures
Smodel_110_ml <- accuracy(model_110_ml)[1:7]
Smodel_410_ml <- accuracy(model_410_ml)[1:7]
Smodel_111_ml <- accuracy(model_111_ml)[1:7]
Smodel_210_ml <- accuracy(model_210_ml)[1:7]
Smodel_211_ml <- accuracy(model_211_ml)[1:7]
Smodel_311_ml <- accuracy(model_311_ml)[1:7]
df.Smodels <- data.frame(

rbind(Smodel_110_ml,Smodel_410_ml,Smodel_111_ml,Smodel_210_ml,Smodel_211_ml,S
model_311_ml)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")

rownames(df.Smodels) <- c("ARIMA {1,1,0}", "ARIMA {4,1,0}","ARIMA
{1,1,1}","ARIMA {2,1,0}","ARIMA {2,1,1}", "ARIMA {3,1,1}")
round(df.Smodels,  digits = 3)

#ARIMA(3,1,1) model has the lowest values in 3 error measures.

#Model Diagnostics

residual.analysis(model = model_211_ml) # Best model
residual.analysis(model = model_311_ml) # New model
residual.analysis(model = model_311_css) # New model

#All the tests looks good except for a bar in the histogram. The p value in Box-Ljung test is
highest for model 211, so is the significance for normality
#Model 211 is also the smaller model, so we can go ahead with it.

residual.analysis(model = model_111_ml) # testing

# testing out other models - not included in the report as they are not good models
residual.analysis(model = model_410_ml)
residual.analysis(model = model_110_ml)
residual.analysis(model = model_210_ml)

#111 model with a histogram within +3 and -3 but has a significant lag in ACF but nothing in
the Box-Ljung test
#Could potentially take 111 or check GARCH models?

par(mfrow=c(1,1))
```

```
#Model Forecasting - 111
frcCSS = forecast::forecast(model_111_css,h=10)
plot(frcCSS)
lines(Lag(fitted(model_111_css),-1), col= "blue")
legend("topleft", lty=1, pch=1, col=c("blue","black"), text.width = 5, c("Data", "Fitted "))
frcCSS

frcML = forecast::forecast(model_111_ml,h=10)
plot(frcML)
lines(Lag(fitted(model_111_ml),-1), col= "blue")
legend("topleft", lty=1, pch=1, col=c("blue","black"), text.width = 5, c("Data", "Fitted "))
frcML
```

6.References

- Dr. Demirhan (2024) ‘Analysis of Trends' [PowerPoint slides, MATH1318], RMIT University, Melbourne.
- Jonathan D. Cryer and Kung-Sik Chan (2008) Time Series Analysis: With Applications in R, Springer New York, NY
- Pep Canadell (2012): Global Carbon Project - 2016 Global Carbon Budget. CDIAC, ESS-DIVE repository. Dataset. doi:10.3334/CDIAC/GCP_2016 accessed via https://data.ess-dive.lbl.gov/datasets/doi:10.3334/CDIAC/GCP_2016 on 2024-06-15