TIME SERIES ANALYSIS OF GLOBAL LAND TEMPERATURE ANOMALIES

Deepthi Suresh
S3991481

Table of Contents

Table of Figures

1. Introduction

The dataset consists of the global temperature anomalies between land and ocean between the years 1850 and 2023. A time series analysis is conducted to identify the changes in the temperature over the years compared to a reference value. The global anomalies are calculated with respect to the averages of 1901-2000. The dataset comprises of 174 observations. The aim of the study is to identify the best fitting model that can be used to forecast if needed for future studies.

2. Method

To identify the best fitting time series model, a descriptive analysis is conduct. The descriptive analysis also includes a creation of TS plot which helps in identifying the 5 important descriptive features of the TS models - Trend, Seasonality, Changing Variance, Behavior and Change points. In case of a presence of changing variance and trend, the series is transformed using a Box-Cox transformation to adjust the changing variance and differencing is performed on the series to detrend the series. On the stationary series a set of model specifications methods like ACF,PACF, EACF and BIC tables are performed to identify a possible set of models. These models are then fitted using CSS and ML methods and the best models are identified based on AIC, BIC and Error measure values. R programming software is used for this study. Important snippets of the code is 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 Global Land temperature Anomalies is uploaded into R and stored in a variable called temperature. The summary statistics of the raw data shows that the data is available from the year 1850 up to 2023, and the minimum change in the mean temperature compared to the average temperature between 1901 and 2000 is -0.44 and maximum change is 0.91. The average change is the temperature within these years is 0.06. The median lies at 0 which could indicate that the data is equally distributed around 0.

```
#Data Preprocessing
temperature <- read_csv("~/OneDrive - RMIT University/Time Series Analysis/Assignment 2/assignment2Data2024.csv")
```

```r
summary(temperature) # creating Summary Statistics for the raw data
```

```
##       Year         Anomaly
##  Min.   :1850   Min.   :-0.44000
##  1st Qu.:1893   1st Qu.:-0.12750
##  Median :1936   Median : 0.00000
##  Mean   :1936   Mean   : 0.06218
##  3rd Qu.:1980   3rd Qu.: 0.23000
##  Max.   :2023   Max.   : 0.91000
```

Figure 1 The summary statistics of the land temperature Anomalies raw data

The data is then converted into a TS object called temperature_TS with a start year of 1850 and end year of 2023. Since this is an annual data, the frequency is taken as 1. The class function confirms that the data is now in a Time series format. The time series data also has the same summary statistics as the raw data as shown in Figure 2.

```r
#Converting to a TS object
temperature_TS <- ts(as.vector(temperature$Anomaly),start=1850, end=2023, frequency = 1)

class(temperature_TS)
```

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

```r
summary(temperature_TS)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.44000 -0.12750  0.00000  0.06218  0.23000  0.91000
```

Figure 2 Summary statistics of the TS data

Figure 3 shows the Time series plot for the anomalies, which indicates a slight upward trend especially after the data point at 1900. There can potentially be a quadratic trend in this case. The plot doesn't seem to indicate a distinctive seasonality or repeating patterns. There seems to be some changing variance especially between 1900 and 1950. The time series could potentially have a MA (moving average) behaviour. Two change points can be noticed in the plot – one between 1850 and 1900 and another between 1900 and 1950 where there is a sudden spike in the temperature anomalies.

```r
#creating a TS plot
plot(temperature_TS,type='o',ylab='Anamolies',
     main = " Time series plot of Global Land Temperature Anomalies")
```

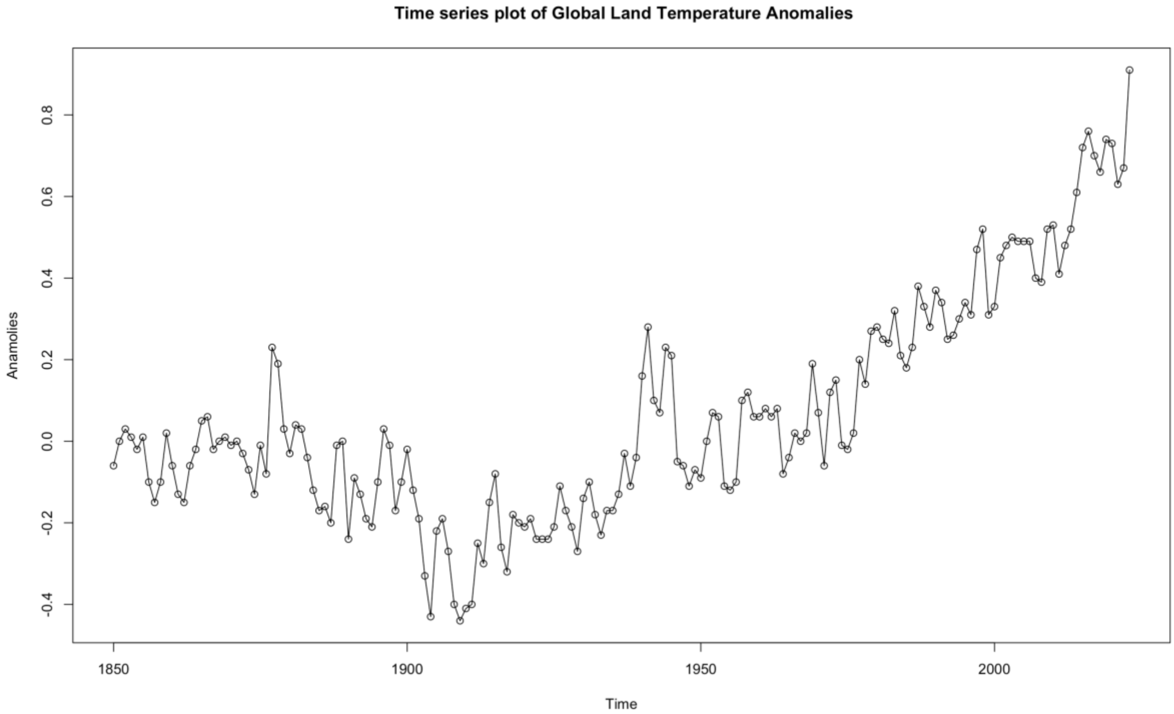


Figure 3 Time series plot of the Global Land Temperature Anomalies

Further investigations have been conducted to identify if there is any relation between the previous year's anomalies with the next year's anomalies. Figure 4 shows that Lag 1 has a correlation of 0.93 which indicates a strong positive correlation between current year's anomalies and the previous year's anomalies.

```
#Lag - checking the impact of previous year's temperature anomaly on the next year's temperature anomaly
y = temperature_TS
x = zlag(temperature_TS) # generate the first lag of the Global Land Temperature Anomalies time series
```

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

```
## [1] 0.9399931
```

Figure 4 Correlation of Lag 1

Correlation of the second lag as seen in Figure 5, shows that there is a positive correlation meaning that there is a correlation between current year's anomalies with the anomalies that were present two years ago, however the correlation is not as strong as Lag 1.

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

```
## [1] 0.8871311
```

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 anomalies. Lag 2 scatter plot is more widely distributed than lag 1 indicating a lower correlation between the years.

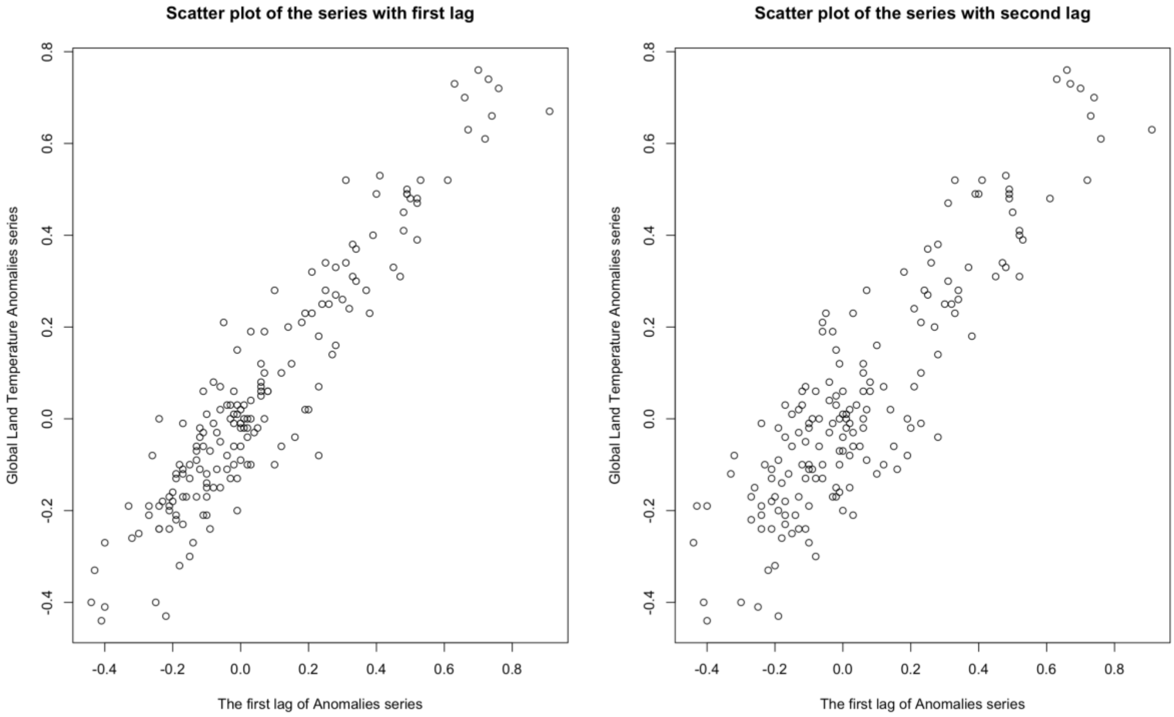


Figure 6 Side-by- side comparison of the Scatter plot for Lag 1 and Lag 2

Figure 7 shows the ACF of the Anomalies series with a decaying pattern indicating a possible trend. There are also significant autocorrelations with several lines above the upper interval

```
#Displying the ACF
acf(temperature_TS, lag.max = 60,main = "ACF of Global Land Temperature Anomalies series")
```

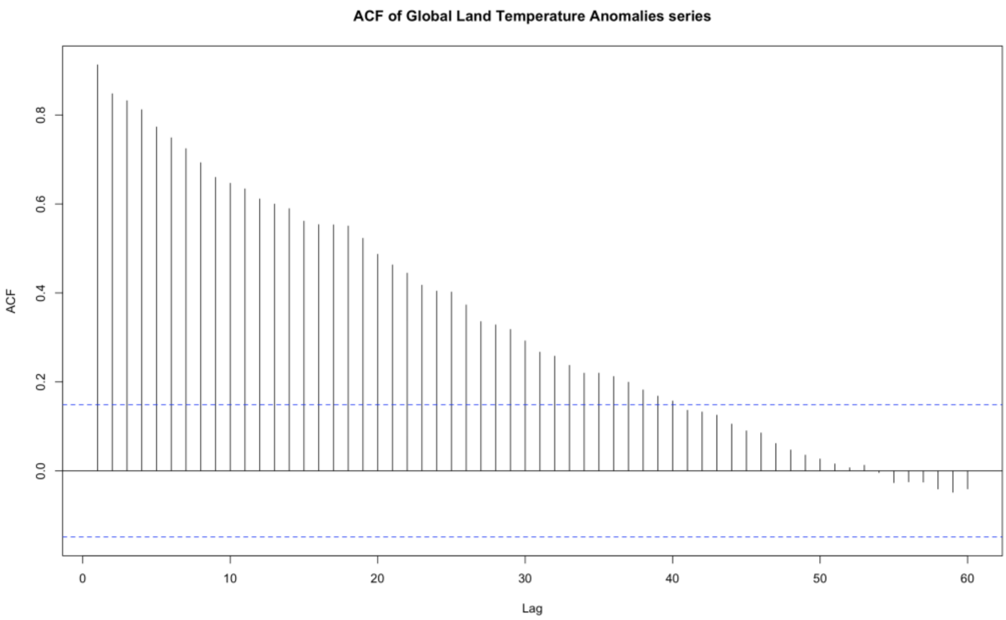


Figure 7: ACF of the Global Land temperature Anomalies series

3.2 Model Specification

From the descriptive analysis conducted in the earlier section it is clear that the series is non-stationary as it has a trend and changing variance as depicted in the time series plot as well as the ACF plots.

Figure 8 shows the ACF and PACF plots of the Anomalies series. One of the indicators of non-stationary series is a slowly decaying pattern in the ACF and a significantly high partial auto correlation in lag 1of PACF. Both these indicators are clearly present in Figure 8 and therefore confirming our assumption of non-stationarity.

```
acf(temperature_TS, main ="ACF plot of Global Land Temperature Anomalies series.")
pacf(temperature_TS, main ="PACF plot of Global Land Temperature Anomalies series.")
```

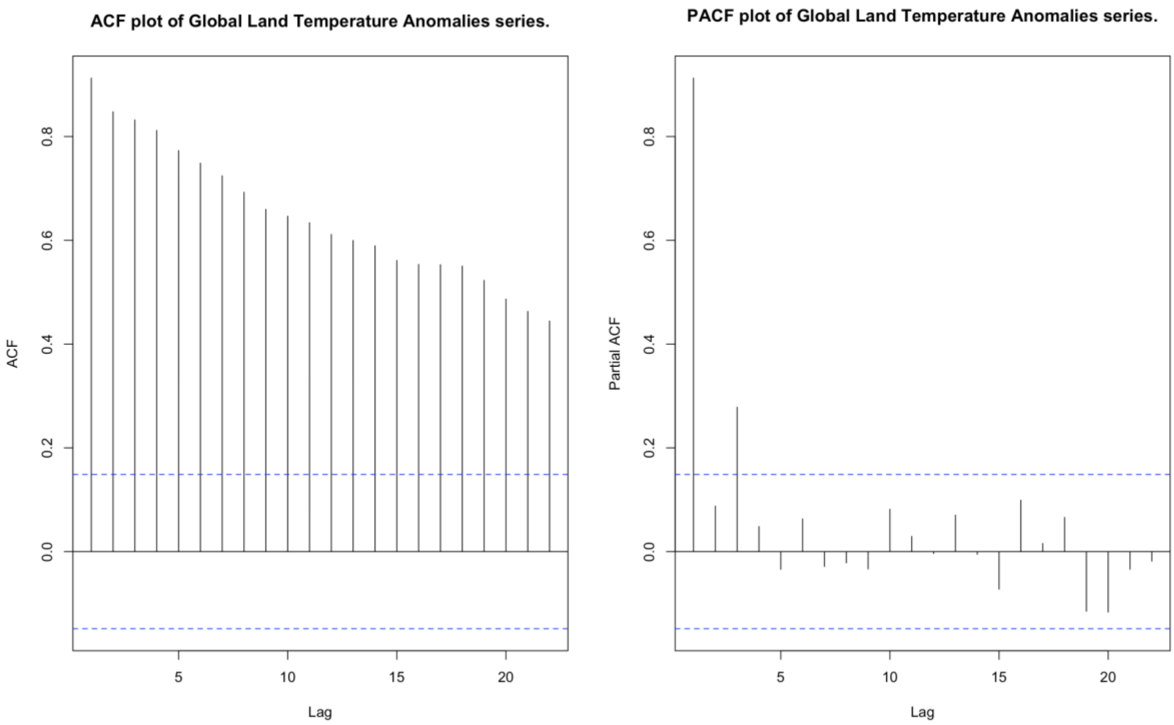
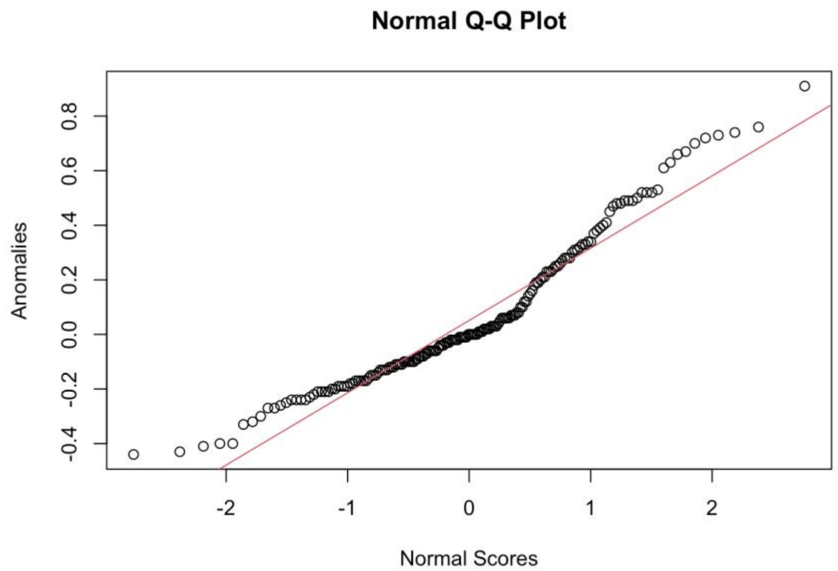


Figure 8: ACF and PACF plot of Global Land Temperature Anomalies Series

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

QQ Plot in Figure 9, shows that the data is not normally distributed, as there is no alignment with the QQ line, which is further confirmed by the Shapiro-Wilk test with a p-value of <0.05.

```r
  #checking for normality
qqnorm(temperature_TS, ylab="Anomalies", xlab="Normal Scores")
qqline(temperature_TS, col = 2)
```



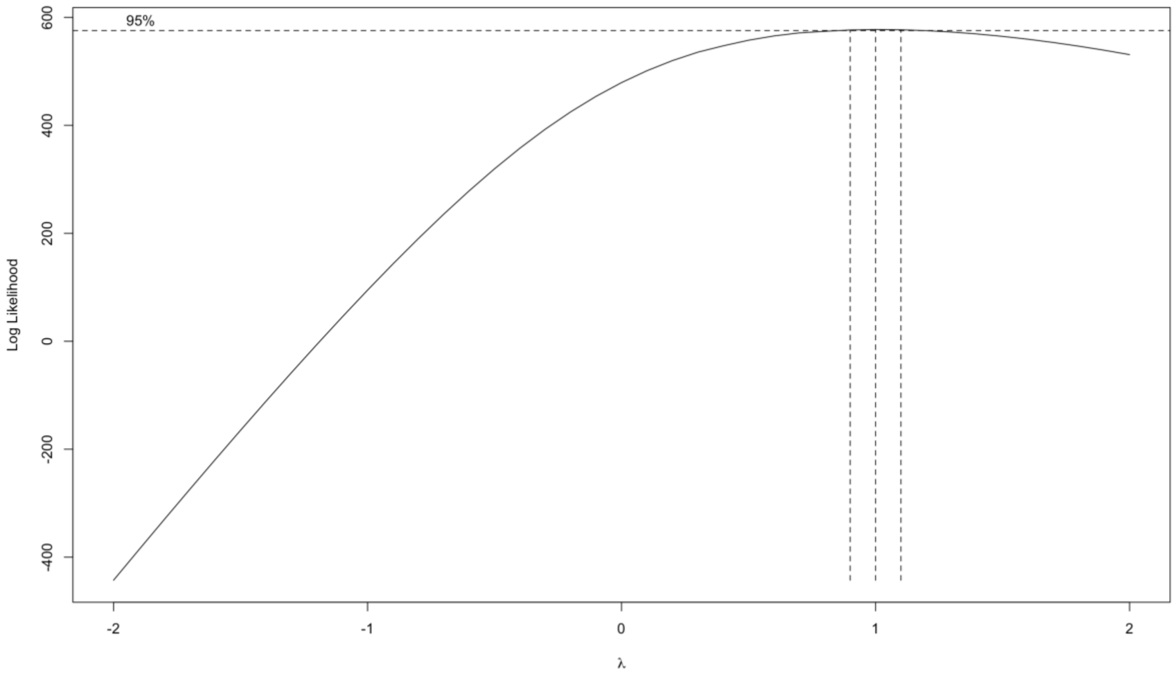

```r
shapiro.test(temperature_TS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  temperature_TS
## W = 0.94279, p-value = 1.897e-06
```

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

In order to handle the changing variance in the time series data, a Box-cox transformation is performed. Upon initial attempt, the BC function couldn't be run as there are negative values in the data. This was rectified by adding a constant value of 0.01 and then the BC function returned a lambda of 1 as shown in Figure 10, which indicates that no transformations were done.

```r
  #Running a Box-Cox Transformation
#BC = BoxCox.ar(temperature_TS)
# We get an error message as there are negative values
temperature_TS2 <- temperature_TS + abs(min(temperature_TS)) + 0.01 #To remove negative values
BC = BoxCox.ar(temperature_TS2)
```




```
BC$ci
```

```
## [1] 0.9 1.1
```

```
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda
```

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

Figure 10 Box Cox transformation and Lambda calculation of Anomalies Series

The time series plot of the BC transformed data is shown in figure 11, which indicates that there has been no changes in the time series plot.

```
BC.temperature_TS = (temperature_TS2^lambda-1)/lambda

plot(BC.temperature_TS,type='o',ylab='Global Land Temperature Anomalies', main = " Time series plot of BC transfo
rmed Global Land Temperature Anomalies series")
```

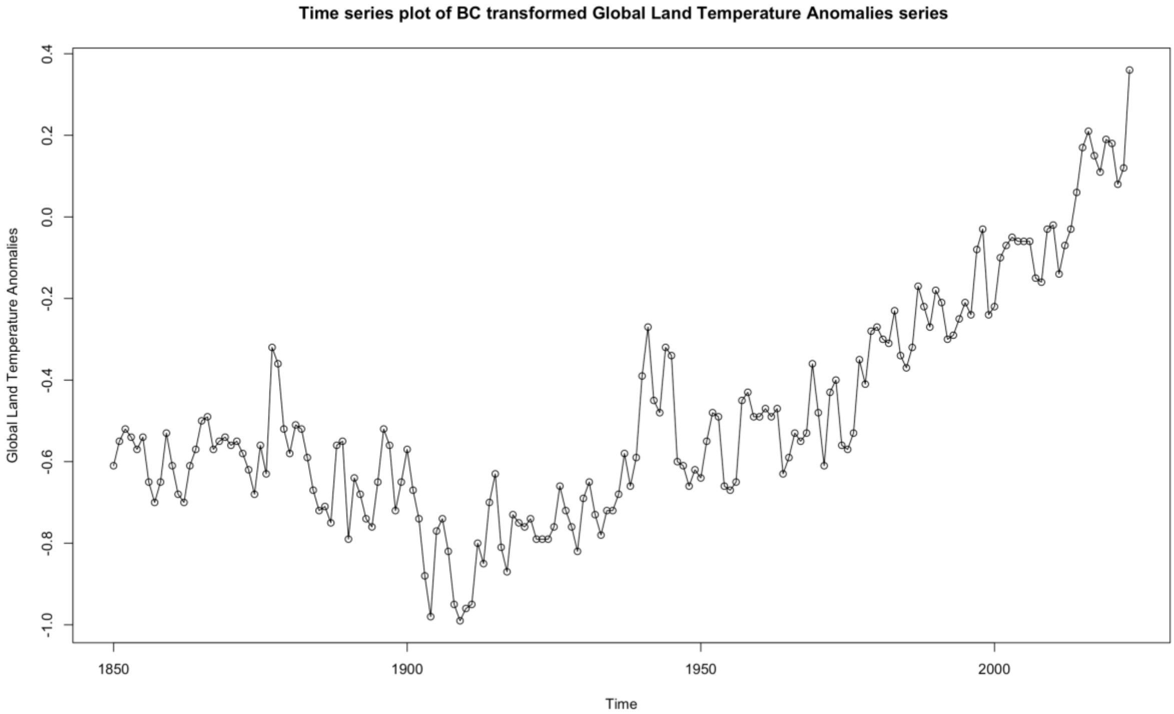
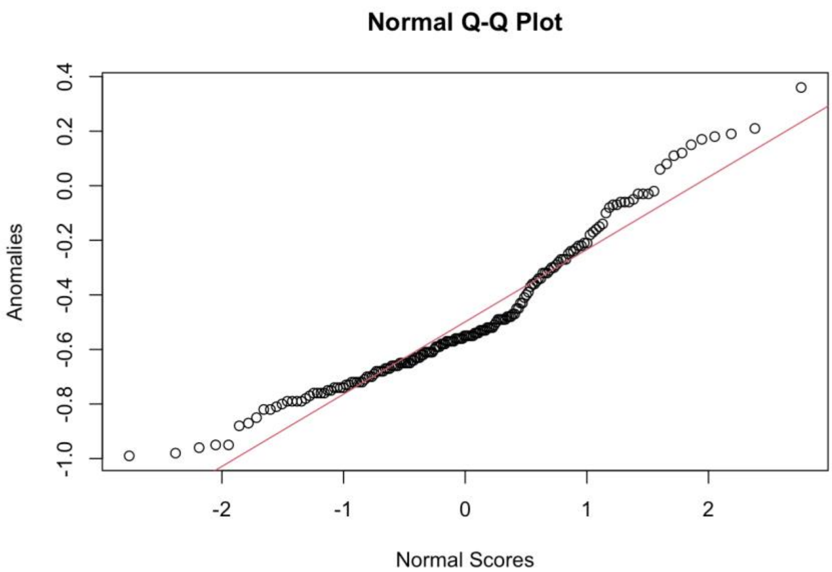


Figure 11 Time series plot of the Box Cox transformed Anomalies series

Figure 12 shows the QQ plot of the BC transformed data, which indicated that the series continues to be not normal, which is further confirmed by the Shapiro-Wilk normality test with a p-value < 0.05. There has been no changes from the original normality plot.

```
qqnorm(BC.temperature_TS,ylab="Anomalies", xlab="Normal Scores")
qqline(BC.temperature_TS, col = 2)
```



```
shapiro.test(BC.temperature_TS)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  BC.temperature_TS
## W = 0.94279, p-value = 1.897e-06
```

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

Figure 13 shows the result of the ADF test. ADF test is a hypothesis test run to check the stationarity of a series. Since the p-value in greater than 0.05 we fail to reject the null hypothesis and the series is non-stationary.

```
adf.test(BC.temperature_TS) #therefore non-stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  BC.temperature_TS
## Dickey-Fuller = -1.1974, Lag order = 5, p-value = 0.9044
## alternative hypothesis: stationary
```

Figure 13 ADF Test on the box-cox transformed series

We can continue to perform differencing on the Temperature_TS2 data (the data adjusted with a constant of 0.01 to avoid any InF or Nan errors) to detrend the series and convert the series from a non-stationary one to a stationary series. Here the raw data is considered as conducting a BC transformation did not develop any changes to the normality of the series. In this case differencing once has handled the trend. The time series plot, as shown in figure 14, indicates a stationary data series.

```
  #Performing differencing
#continuing with the raw data
diff.temperature_TS = diff(temperature_TS2)
plot(diff.temperature_TS,type='o',ylab='First difference of Anomalies', main ="Time series plot of the first diff
erence of
     the Global Land Temperature Anomalies series.")
```

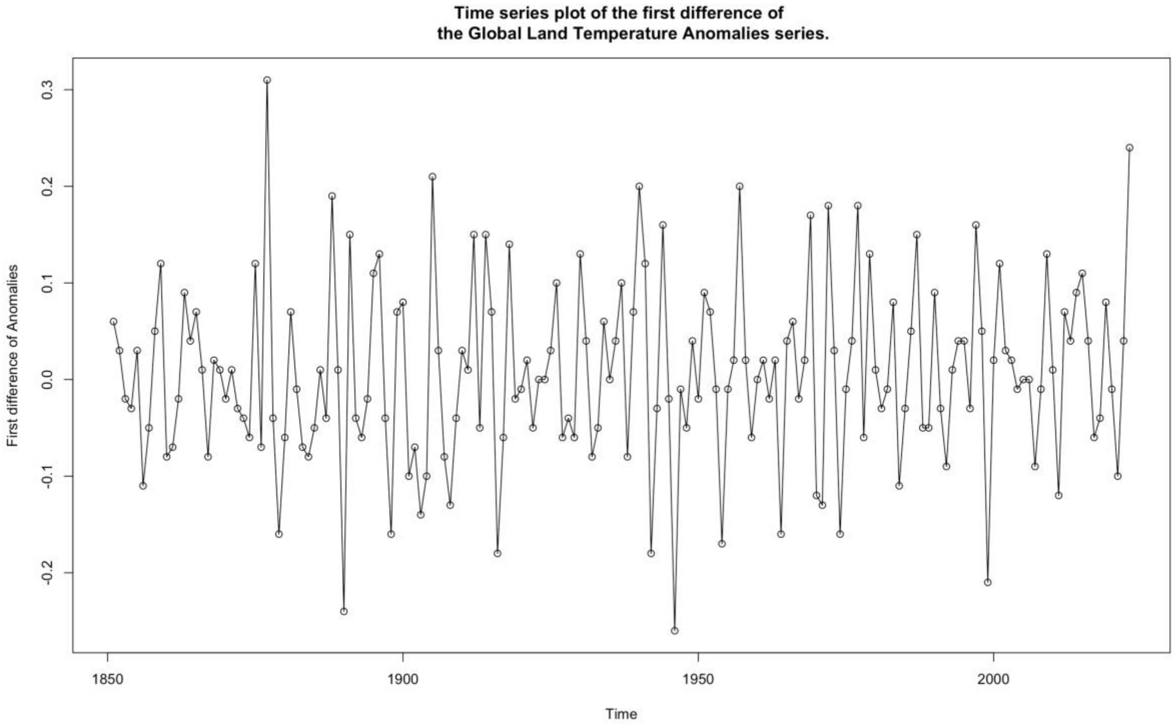


Figure 14 : Time series plot of the Box-Cox transformed and first difference of the Anomalies Series

The stationarity of the data can be further confirmed by running multiple hypothesis tests such as the ADF test, PP test and KPSS Test. 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. The results are shown in Figure 15, which shows that the p-value in ADF and PP tests are less than 0.05 and therefore we reject the null hypothesis. In the case KPSS the p-value is greater than 0.05 and therefore we accept the null hypothesis. We can therefore conclude that the differencing converted the series into a stationary series.

```r
# applying the tests to the differenced series.
adf.test(diff.temperature_TS)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff.temperature_TS
## Dickey-Fuller = -7.0878, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
kpss.test(diff.temperature_TS)
```

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

```
##
##  KPSS Test for Level Stationarity
##
## data:  diff.temperature_TS
## KPSS Level = 0.24108, Truncation lag parameter = 4, p-value = 0.1
```

```r
pp.test(diff.temperature_TS)
```

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

```
##
##  Phillips-Perron Unit Root Test
##
## data:  diff.temperature_TS
## Dickey-Fuller Z(alpha) = -136.61, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

Figure 15 Hypothesis testing conducted on the differenced series

Figure 16 shows the ACF and PACF of the transformed and first differenced Anomalies series. The ACF plot no longer has a decaying pattern and PACF no longer has a high significant autocorrelation In the first lag. Therefore we can further confirm that the data series has been transformed to a stationary series.

```r
par(mfrow=c(1,2))
acf(diff.temperature_TS, main ="ACF plot of the first difference of the Global Land Temperature Anomalies serie
s.", lag.max = 60)
pacf(diff.temperature_TS, main ="PACF plot of the first difference of Global Land Temperature Anomalies series.",
lag.max = 60)
```

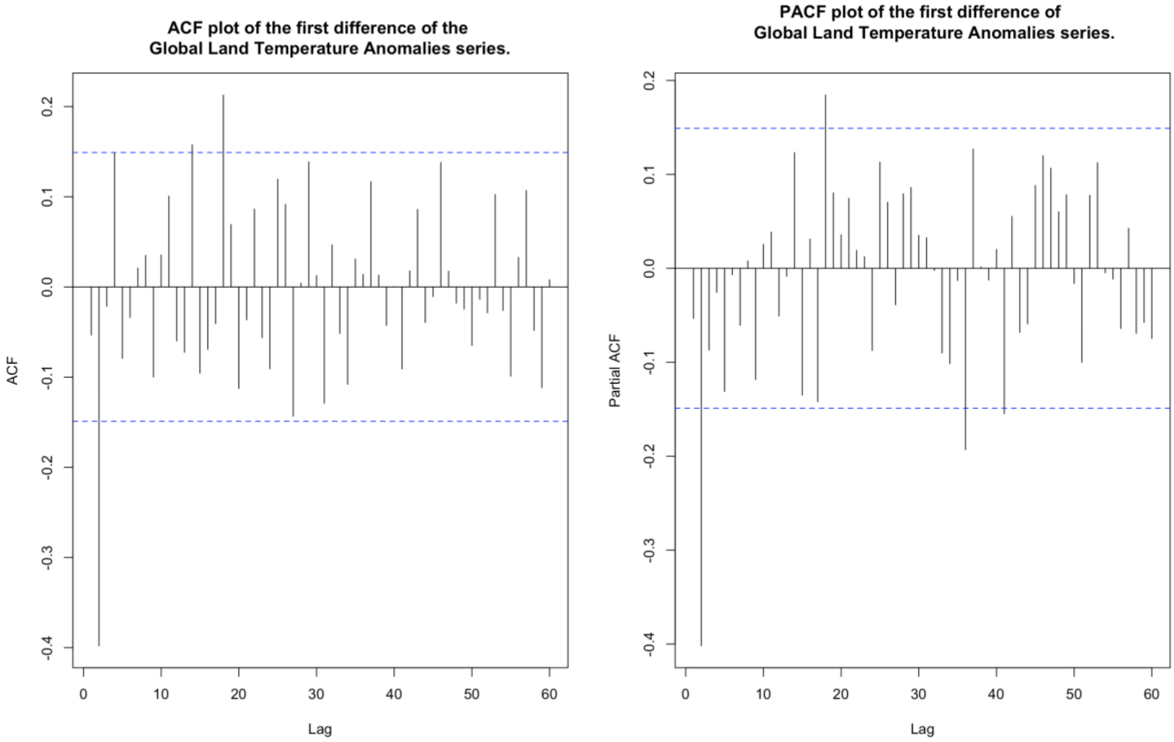


Figure 16 ACF and PACF of the Box-Cox transformed and first differenced Anomalies Series

From the above ACF and PACF plots in Figure 16, we can determine p as 1, as there is only 1 significant lag in PACF. q in this case can be determined from the ACF plot as 1 (obvious significant lag) or 2 (a not so obvious significant lag under 10 year lag). The set of possible models are:

{ARIMA(1,1,1) , ARIMA(1,1,2)}

Note: in this case any significant lags after year 10 is considered as late lags

Figure 17 shows the EACF plot of the Anomalies series. The top left corner is at (0,2) as there are no x's in the row.

```
#identifying possible models
eacf(diff.temperature_TS)
```

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

Figure 17 EACF of the differenced Anomalies Series

The set of possible models from the EACF are:

{ARIMA(0,1,2) , ARIMA(0,1,3), ARIMA(1,1,2), ARIMA(1,1,3)}

Figure 18 gives the BIC table of the Anomalies series. Nar and NMA is considered as 5 since the models from the previous methods are on the smaller side. The best model in this case is (2,0). The second best model has a q of 4, however it is not supported by 3rd or 4th best models and therefore ignored. The third and 4th best models, gives a p of 1 and q of 1 as therefore considered for further analysis.

The set of possible models from the BIC table are:

{ARIMA(1,1,0) , ARIMA(1,1,1), ARIMA(2,1,0), ARIMA(2,1,1)}

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

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 3 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
plot(res)
```

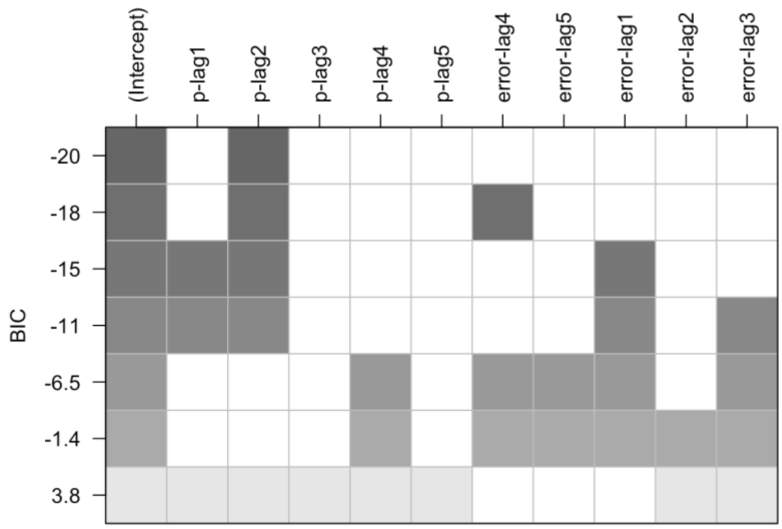


Figure 18: BIC table of differenced Anomalies series

The final set of possible models are :

{ARIMA(1,1,1) , ARIMA(1,1,2), ARIMA(0,1,2) , ARIMA(0,1,3), ARIMA(1,1,3), ARIMA(1,1,0) , 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(0,1,2)

Figure 19 shows that, in both ML and CSS parameter estimation method the coefficients are significant which could indicate that this is a good model.

```
#The final set of possible models are :

#  {ARIMA(1,1,1) , ARIMA(1,1,2), ARIMA(0,1,2) , ARIMA(0,1,3),
#    ARIMA(1,1,3), ARIMA(1,1,0) , ARIMA(2,1,0), ARIMA(2,1,1)}
```

```
#3.3 Model Fitting

#ARIMA(0,1,2)
model_012_css = Arima(temperature_TS2,order=c(0,1,2),method='CSS')
lmtest::coeftest(model_012_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.155660   0.067130 -2.3188   0.02041 *
## ma2 -0.380127   0.062131 -6.1182 9.466e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_012_ml = Arima(temperature_TS2,order=c(0,1,2),method='ML')
coeftest(model_012_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.154653   0.067355 -2.2961   0.02167 *
## ma2 -0.376908   0.062077 -6.0716 1.267e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 19 CSS and ML model fitting of ARIMA(0,1,2)

3.3.2 ARIMA(0,1,3)

In a ARIMA(0,1,3) model as shown in Figure 20, by adding a MA3 component, 2 out of 3 coefficients turns insignificant in both CSS and ML methods, indicating that this might not be a good model.

```r
#ARIMA(0,1,3)
model_013_css = Arima(temperature_TS2,order=c(0,1,3),method='CSS')
lmtest::coeftest(model_013_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.120979   0.087093 -1.3891    0.1648
## ma2 -0.380308   0.061678 -6.1660 7.002e-10 ***
## ma3 -0.053368   0.083661 -0.6379    0.5235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_013_ml = Arima(temperature_TS2,order=c(0,1,3),method='ML')
coeftest(model_013_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.121311   0.086910 -1.3958    0.1628
## ma2 -0.377091   0.061659 -6.1158 9.609e-10 ***
## ma3 -0.051400   0.083107 -0.6185    0.5363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 20 CSS and ML model fitting of ARIMA(0,1,3)

3.3.3 ARIMA(1,1,0)

Figure 21 shows the CSS and ML model for ARIMA(1,1,0), which indicates that the coefficient AR1 is insignificant in both CSS and ML methods and therefore may not be a good model

```r
#ARIMA(1,1,0)
model_110_css = Arima(temperature_TS2,order=c(1,1,0),method='CSS')
lmtest::coeftest(model_110_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.052700   0.077314 -0.6816   0.4955
```

```r
model_110_ml = Arima(temperature_TS2,order=c(1,1,0),method='ML')
coeftest(model_110_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.052514   0.077271 -0.6796   0.4968
```

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

3.3.4 ARIMA(1,1,1)

Figure 22 shows that when we added a ma component both ar1 and ma1 coefficient turned significant, making this a potentially good model.

```r
#ARIMA(1,1,1)
model_111_css = Arima(temperature_TS2,order=c(1,1,1),method='CSS')
lmtest::coeftest(model_111_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.483274   0.110230   4.3842 1.164e-05 ***
## ma1 -0.772055   0.075674 -10.2024 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_111_ml = Arima(temperature_TS2,order=c(1,1,1),method='ML')
coeftest(model_111_ml)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error  z value  Pr(>|z|)
## ar1  0.50331    0.11182   4.5012 6.757e-06 ***
## ma1 -0.78487    0.07365 -10.6569 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

3.3.5 ARIMA(1,1,2)

Figure 23 shows that when we added an additional ma component 2 out of the 3 coefficient turned insignificant and therefore this might not be a good model.

```
#ARIMA(1,1,2)
model_112_css = Arima(temperature_TS2,order=c(1,1,2),method='CSS')
lmtest::coeftest(model_112_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.063647   0.153936  0.4135   0.6793
## ma1 -0.202697   0.134390 -1.5083   0.1315
## ma2 -0.370220   0.067526 -5.4826 4.19e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_112_ml = Arima(temperature_TS2,order=c(1,1,2),method='ML')
coeftest(model_112_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.070015   0.155948  0.4490    0.6535
## ma1 -0.207702   0.136249 -1.5244    0.1274
## ma2 -0.366988   0.067420 -5.4433 5.231e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 23 CSS and ML model fitting of ARIMA(1,1,2)

3.3.6 ARIMA(1,1,3)

In this model, another MA component is being added and seen in figure 24, all components are significant, making this a potentially good model.

```r
#ARIMA(1,1,3)
model_113_css = Arima(temperature_TS2,order=c(1,1,3),method='CSS')
lmtest::coeftest(model_113_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.942774   0.047472 -19.860 < 2.2e-16 ***
## ma1  0.847097   0.079913  10.600 < 2.2e-16 ***
## ma2 -0.514501   0.081849  -6.286 3.258e-10 ***
## ma3 -0.440719   0.067183  -6.560 5.382e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model_113_ml = Arima(temperature_TS2,order=c(1,1,3),method='ML')
coeftest(model_113_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error  z value  Pr(>|z|)
## ar1 -0.958601   0.039214 -24.4452 < 2.2e-16 ***
## ma1  0.869860   0.077785  11.1828 < 2.2e-16 ***
## ma2 -0.510864   0.082772  -6.1720 6.745e-10 ***
## ma3 -0.449871   0.067228  -6.6918 2.205e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 24 CSS and ML model fitting of ARIMA(1,1,3)

3.3.7 ARIMA(2,1,0)

Figure 25 shows the CSS and ML methods on ARIMA model (2,1,0). Here AR1 is insignificant in both methods and therefore cannot be considered as a really good model

```
#ARIMA(2,1,0)
model_210_css = Arima(temperature_TS2,order=c(2,1,0),method='CSS')
lmtest::coeftest(model_210_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.078685   0.070804 -1.1113    0.2664
## ar2 -0.414783   0.070755 -5.8622 4.567e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_210_ml = Arima(temperature_TS2,order=c(2,1,0),method='ML')
coeftest(model_210_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.077561   0.070832 -1.0950    0.2735
## ar2 -0.411121   0.070413 -5.8387 5.261e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 25 CSS and ML model fitting of ARIMA(2,1,0)

3.3.8 ARIMA(2,1,1)

Figure 26, shows that adding a MA component did not really help and 2 out of 3 coefficients in this model are still insignificant therefore this model cannot really be considered a good model.

```
#ARIMA(2,1,1)
model_211_css = Arima(temperature_TS2,order=c(2,1,1),method='CSS')
lmtest::coeftest(model_211_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.147460   0.192159  0.7674   0.4429
## ar2 -0.399352   0.077451 -5.1562 2.52e-07 ***
## ma1 -0.273855   0.214609 -1.2761   0.2019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_211_ml = Arima(temperature_TS2,order=c(2,1,1),method='ML')
coeftest(model_211_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.148792   0.194411  0.7653    0.4441
## ar2 -0.395596   0.077386 -5.1120 3.188e-07 ***
## ma1 -0.273818   0.217564 -1.2586    0.2082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 26 CSS and ML model fitting of ARIMA(2,1,1)

3.3.9 AIC and BIC values

Based on the AIC value shown in Figure 27, ARIMA(1,1,3) is the preferred model. This can also be backed by the ML and CSS methods which returned all significant coefficients. However BIC value (figure 26) selected ARMIA(2,1,0) as the preferred model, but only one of the coefficient is significant here. ARMIA(2,1,0) can also be considered as the second best model according to the AIC value

```r
# AIC and BIC values
sort.score(AIC(model_012_ml,model_013_ml,model_110_ml,model_111_ml,model_112_ml,model_113_ml,model_210_ml,model_2
11_ml), score = "aic")
```

```
##              df       AIC
## model_113_ml  5 -354.6224
## model_210_ml  3 -353.2793
## model_211_ml  4 -352.8980
## model_012_ml  3 -352.3256
## model_013_ml  4 -350.7104
## model_112_ml  4 -350.5250
## model_111_ml  3 -338.5562
## model_110_ml  2 -324.3068
```

```r
sort.score(BIC(model_012_ml,model_013_ml,model_110_ml,model_111_ml,model_112_ml,model_113_ml,model_210_ml,model_2
11_ml), score = "bic" )
```

```
##              df       BIC
## model_210_ml  3 -343.8194
## model_012_ml  3 -342.8657
## model_211_ml  4 -340.2848
## model_113_ml  5 -338.8560
## model_013_ml  4 -338.0972
## model_112_ml  4 -337.9119
## model_111_ml  3 -329.0963
## model_110_ml  2 -318.0002
```

Figure 27 AIC and BIC values of all the models

3.3.9 Error Measures

Figure 28, shows the error measures of the different models. As highlighted in figure 28, ARIMA(2,1,0) has the lowest error value in both MAE and MASE and ARIMA (1,1,0) has the lowest error values in ME and MAPE. Other models with the lowest error values are ARIMA(1,1,1) and ARIMA(1,1,3).

```r
# Error Measures
Smodel_012_css <- accuracy(model_012_css)[1:7]
Smodel_013_css <- accuracy(model_013_css)[1:7]
Smodel_110_css <- accuracy(model_110_css)[1:7]
Smodel_111_css <- accuracy(model_111_css)[1:7]
Smodel_112_css <- accuracy(model_112_css)[1:7]
Smodel_113_css <- accuracy(model_113_css)[1:7]
Smodel_210_css <- accuracy(model_210_css)[1:7]
Smodel_211_css <- accuracy(model_211_css)[1:7]
df.Smodels <- data.frame(
  rbind(Smodel_012_css,Smodel_013_css,Smodel_110_css,Smodel_111_css,Smodel_112_css,
        Smodel_113_css,Smodel_210_css, Smodel_211_css)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                          "MASE", "ACF1")
rownames(df.Smodels) <- c("ARIMA(0,1,2)", "ARIMA(0,1,3)", "ARIMA(1,1,0)",
                          "ARIMA(1,1,1)", "ARIMA(1,1,2)", "ARIMA(1,1,3)", "ARIMA(2,1,
0)", "ARIMA(2,1,1)")
round(df.Smodels,  digits = 3)
```

```
##                 ME  RMSE   MAE     MPE   MAPE  MASE   ACF1
## ARIMA(0,1,2) 0.010 0.086 0.065 -11.664 26.198 0.899  0.004
## ARIMA(0,1,3) 0.011 0.085 0.065 -11.392 26.026 0.901 -0.023
## ARIMA(1,1,0) 0.005 0.093 0.071  -8.254 24.914 0.992 -0.025
## ARIMA(1,1,1) 0.009 0.089 0.068 -13.166 28.064 0.953  0.102
## ARIMA(1,1,2) 0.010 0.085 0.064 -11.699 26.029 0.895 -0.008
## ARIMA(1,1,3) 0.010 0.084 0.064 -11.190 25.707 0.894 -0.022
## ARIMA(2,1,0) 0.007 0.085 0.062 -11.598 25.553 0.867 -0.042
## ARIMA(2,1,1) 0.008 0.085 0.063 -12.057 26.012 0.873 -0.001
```

Figure 28 Error measures of all the models

3.3.10 Model Selection

In summary, ARIMA(1,1,1) , ARIMA(0,1,2) , ARIMA(1,1,3) have all significant coefficients. The top 4 models based on AIC are ARIMA(1,1,3), ARIMA(2,1,0), ARIMA(2,1,1) and ARIMA(0,1,2). The top 4 models based on BIC values are ARIMA(2,1,0), ARIMA(0,1,2), ARIMA(2,1,1) and ARIMA(1,1,3). Based on the error measures ARIMA(2,1,0) and ARIMA (1,1,0) has the lowest error in two of the measures whereas ARIMA(1,1,3) and ARIMA(1,1,1) has the lowest errors in one measure each.

From the table in Figure 29, ARIMA(2,1,0) has 2 low error measures, is the top model based on BIC and the second most preferred model in AIC. However only one of its coefficient is significant. On the other hand, ARIMA(1,1,3) has all significant coefficients and is the top most model according to AIC values. However it lies in the top four preferred models according to BIC values and has the lowest RMSE value.

Based on the parsimony principle, ARIMA(2,1,0) is considered to be the best model as it has only 2 parameters in total while ARIMA(1,1,3) has 4.

Models	AIC ranking	BIC ranking	Error Measures	Significant coefficients
ARIMA(1,1,1)	7	7	1 lowest measure - MPE	2 out of 2
ARIMA(1,1,2)	6	6		1 out of 3
ARIMA(0,1,2)	4	2		2 out of 2
ARIMA(0,1,3)	5	5		1 out of 3
ARIMA(1,1,3)	1	4	1 lowest measure - RMSE	4 out of 4
ARIMA(1,1,0)	8	8	2 lowest measure – ME and MAPE	0 out of 1
ARIMA(2,1,0)	2	1	2 lowest measures – MAE and MASE	1 out of 2
ARIMA(2,1,1)}	3	3		1 out of 3

Figure 29 Table showing a summary of the goodness of fit of the models

3.3.11 Over parameterised Models

ARIMA(3,1,0) and ARIMA(2,1,1) are over parametrised models for ARIMA(2,1,0)

```
#Best Model ARIMA(2,1,0)

#ARIMA(3,1,0) and ARIMA(2,1,1) are over parametrised models for ARIMA(2,1,0)

# ARIMA(3,1,0)
model_310_css = Arima(temperature_TS2,order=c(3,1,0),method='CSS')
coeftest(model_310_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.114459   0.076995 -1.4866    0.1371
## ar2 -0.422290   0.070851 -5.9602 2.519e-09 ***
## ar3 -0.089112   0.077166 -1.1548    0.2482
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_310_ml = Arima(temperature_TS2,order=c(3,1,0),method='ML')
coeftest(model_310_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.112964   0.077047 -1.4662    0.1426
## ar2 -0.418351   0.070420 -5.9408 2.837e-09 ***
## ar3 -0.087789   0.076714 -1.1444    0.2525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 30: Model fitting of over parametrised models ARIMA(3,1,0)

Figure 30, shows the CSS and ML methods on ARIMA model (3,1,0). Here, the additional parameter ar3 is insignificant.

```
# ARIMA(2,1,1)
model_211_css = Arima(temperature_TS2,order=c(2,1,1),method='CSS')
coeftest(model_211_css)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1  0.147460   0.192159  0.7674   0.4429
## ar2 -0.399352   0.077451 -5.1562 2.52e-07 ***
## ma1 -0.273855   0.214609 -1.2761   0.2019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model_211_ml = Arima(temperature_TS2,order=c(2,1,1),method='ML')
coeftest(model_211_ml)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1  0.148792   0.194411  0.7653    0.4441
## ar2 -0.395596   0.077386 -5.1120 3.188e-07 ***
## ma1 -0.273818   0.217564 -1.2586    0.2082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 31 Model fitting of over parametrised models ARIMA(2,1,2)

Figure 31 shows the CSS and ML methods on ARIMA model (2,1,1). Here, the additional parameter MA1 is insignificant.

Therefore, the best model here is ARIMA(2,1,0)

4. Conclusion

From the above model fitting strategy, we can conclude that ARIMA (2,1,0) is the best model. However, conducting a residual analysis is out of scope for this study, and therefore we cannot confirm this assumption. In case the residual analysis returns a negative result, the second-best model to consider is ARIMA(1,1,3).

5. Appendix

```
library("TSA") # calling the Time Series Package
library("readr")
library("dplyr")
library(fUnitRoots)
library(lmtest)
library(tseries)
library(forecast)

#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")')
 }
}

#Data Pre-processing
temperature <- read_csv("~/OneDrive - RMIT University/Time Series
Analysis/Assignment 2/assignment2Data2024.csv")
class(temperature)
summary(temperature) # creating Summary Statistics for the raw data

#Converting to a TS object
temperature_TS <- ts(as.vector(temperature$Anomaly),start=1850, end=2023,
frequency = 1)

class(temperature_TS)
summary(temperature_TS)

#creating a TS plot
plot(temperature_TS,type='o',ylab='Anamolies',
   main = " Time series plot of Global Land Temperature Anomalies")

#Lag - checking the impact of previous year's temperature anomaly on the next year's
temperature anomaly
par(mfrow=c(1,2))
y = temperature_TS
x = zlag(temperature_TS) # generate the first lag of the Global Land Temperature
Anomalies 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='Global Land Temperature Anomalies series', xlab='The first
lag of Anomalies series',
    main = "Scatter plot of the series with first lag")

# looking at the second lag

x = zlag(zlag(temperature_TS))
index = 3:length(x)
cor(y[index],x[index])

plot(y[index],x[index],ylab='Global Land Temperature Anomalies series', xlab='The first
lag of Anomalies series',
    main = "Scatter plot of the series with second lag")
par(mfrow=c(1,1))

#Displying the ACF
acf(temperature_TS, lag.max = 60,main = "ACF of Global Land Temperature Anomalies
series")

#-------------------------------------------------------------

#3.2 Model Specification
  #Non-Stationary Models

par(mfrow=c(1,2))
acf(temperature_TS, main ="ACF plot of Global Land Temperature Anomalies series.")
pacf(temperature_TS, main ="PACF plot of Global Land Temperature Anomalies series.")
par(mfrow=c(1,1))

  #checking for normality
qqnorm(temperature_TS, ylab="Anomalies", xlab="Normal Scores")
qqline(temperature_TS, col = 2)
shapiro.test(temperature_TS)

  #Running a Box-Cox Transformation
#BC = BoxCox.ar(temperature_TS)
# We get an error message as there are negative values
temperature_TS2 <- temperature_TS + abs(min(temperature_TS)) + 0.01 #To remove
negative values
BC = BoxCox.ar(temperature_TS2)
BC$ci
lambda <- BC$lambda[which(max(BC$loglike) == BC$loglike)]
lambda
BC.temperature_TS = (temperature_TS2^lambda-1)/lambda
```

```
plot(BC.temperature_TS,type='o',ylab='Global Land Temperature Anomalies', main = "
Time series plot of BC transformed Global Land Temperature Anomalies series")

qqnorm(BC.temperature_TS,ylab="Anomalies", xlab="Normal Scores")
qqline(BC.temperature_TS, col = 2)
shapiro.test(BC.temperature_TS)

adf.test(BC.temperature_TS) #therefore non-stationary

 #Performing differencing
#continuing with the raw data
diff.temperature_TS = diff(temperature_TS2)
plot(diff.temperature_TS,type='o',ylab='First difference of Anomalies', main ="Time
series plot of the first difference of
   the Global Land Temperature Anomalies series.")

 # applying the tests to the differenced series.
adf.test(diff.temperature_TS)
kpss.test(diff.temperature_TS)
pp.test(diff.temperature_TS)


par(mfrow=c(1,2))
acf(diff.temperature_TS, main ="ACF plot of the first difference of the
  Global Land Temperature Anomalies series.", lag.max = 60)
pacf(diff.temperature_TS, main ="PACF plot of the first difference of
   Global Land Temperature Anomalies series.", lag.max = 60)
par(mfrow=c(1,1))

 #identifying possible models
eacf(diff.temperature_TS)

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

#The final set of possible models are :

# {ARIMA(1,1,1) , ARIMA(1,1,2), ARIMA(0,1,2) , ARIMA(0,1,3),
#   ARIMA(1,1,3), ARIMA(1,1,0) , ARIMA(2,1,0), ARIMA(2,1,1)}


#----------------------------------------------------------

#3.3 Model Fitting

#ARIMA(0,1,2)
model_012_css = Arima(temperature_TS2,order=c(0,1,2),method='CSS')
```

```
lmtest::coeftest(model_012_css)

model_012_ml = Arima(temperature_TS2,order=c(0,1,2),method='ML')
coeftest(model_012_ml)

#ARIMA(0,1,3)
model_013_css = Arima(temperature_TS2,order=c(0,1,3),method='CSS')
lmtest::coeftest(model_013_css)

model_013_ml = Arima(temperature_TS2,order=c(0,1,3),method='ML')
coeftest(model_013_ml)

#ARIMA(1,1,0)
model_110_css = Arima(temperature_TS2,order=c(1,1,0),method='CSS')
lmtest::coeftest(model_110_css)

model_110_ml = Arima(temperature_TS2,order=c(1,1,0),method='ML')
coeftest(model_110_ml)

#ARIMA(1,1,1)
model_111_css = Arima(temperature_TS2,order=c(1,1,1),method='CSS')
lmtest::coeftest(model_111_css)

model_111_ml = Arima(temperature_TS2,order=c(1,1,1),method='ML')
coeftest(model_111_ml)

#ARIMA(1,1,2)
model_112_css = Arima(temperature_TS2,order=c(1,1,2),method='CSS')
lmtest::coeftest(model_112_css)

model_112_ml = Arima(temperature_TS2,order=c(1,1,2),method='ML')
coeftest(model_112_ml)

#ARIMA(1,1,3)
model_113_css = Arima(temperature_TS2,order=c(1,1,3),method='CSS')
lmtest::coeftest(model_113_css)

model_113_ml = Arima(temperature_TS2,order=c(1,1,3),method='ML')
coeftest(model_113_ml)

#ARIMA(2,1,0)
model_210_css = Arima(temperature_TS2,order=c(2,1,0),method='CSS')
lmtest::coeftest(model_210_css)

model_210_ml = Arima(temperature_TS2,order=c(2,1,0),method='ML')
coeftest(model_210_ml)
```

```
#ARIMA(2,1,1)
model_211_css = Arima(temperature_TS2,order=c(2,1,1),method='CSS')
lmtest::coeftest(model_211_css)

model_211_ml = Arima(temperature_TS2,order=c(2,1,1),method='ML')
coeftest(model_211_ml)

# AIC and BIC values
sort.score(AIC(model_012_ml,model_013_ml,model_110_ml,model_111_ml,model_11
2_ml,model_113_ml,model_210_ml,model_211_ml), score = "aic")
sort.score(BIC(model_012_ml,model_013_ml,model_110_ml,model_111_ml,model_11
2_ml,model_113_ml,model_210_ml,model_211_ml), score = "bic" )

# Error Measures
Smodel_012_css <- accuracy(model_012_css)[1:7]
Smodel_013_css <- accuracy(model_013_css)[1:7]
Smodel_110_css <- accuracy(model_110_css)[1:7]
Smodel_111_css <- accuracy(model_111_css)[1:7]
Smodel_112_css <- accuracy(model_112_css)[1:7]
Smodel_113_css <- accuracy(model_113_css)[1:7]
Smodel_210_css <- accuracy(model_210_css)[1:7]
Smodel_211_css <- accuracy(model_211_css)[1:7]
df.Smodels <- data.frame(

rbind(Smodel_012_css,Smodel_013_css,Smodel_110_css,Smodel_111_css,Smodel_1
12_css,
    Smodel_113_css,Smodel_210_css, Smodel_211_css)
)
colnames(df.Smodels) <- c("ME", "RMSE", "MAE", "MPE", "MAPE",
                  "MASE", "ACF1")
rownames(df.Smodels) <- c("ARIMA(0,1,2)", "ARIMA(0,1,3)", "ARIMA(1,1,0)",
                  "ARIMA(1,1,1)", "ARIMA(1,1,2)", "ARIMA(1,1,3)", "ARIMA(2,1,0)",
"ARIMA(2,1,1)")
round(df.Smodels,  digits = 3)

#Best Model ARIMA(2,1,0)

#ARIMA(3,1,0) and ARIMA(2,1,1) are over parametrised models for ARIMA(2,1,0)

# ARIMA(3,1,0)
model_310_css = Arima(temperature_TS2,order=c(3,1,0),method='CSS')
coeftest(model_310_css)

model_310_ml = Arima(temperature_TS2,order=c(3,1,0),method='ML')
coeftest(model_310_ml)

# ARIMA(2,1,1)
```

```
model_211_css = Arima(temperature_TS2,order=c(2,1,1),method='CSS')
coeftest(model_211_css)

model_211_ml = Arima(temperature_TS2,order=c(2,1,1),method='ML')
coeftest(model_211_ml)

#Best Model ARIMA(2,1,0)
```

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