Time Series Analysis of Antarctic Temperatures

Andie Donovan
PSTAT 174
Professor Raya Feldman
January 6th, 2018

Abstract

Can previous temperature recordings in the South Pole be used to better estimate future changes Antarctic climate? Has the overall surface temperature increased when holding for seasonal fluctuations? To answer these research questions, I conducted a time series analysis of mean monthly recordings of temperatures at the Faraday station in the Argentine Islands of Antarctica. By performing a Box-Cox transformation on the data and then fitting the time series using a Box-Jenkins Seasonal Autoregressive Integrated Moving Averages model, I was able to accurately forecast future temperatures for 40 months within a 95% confidence interval. Although I could not definitely conclude that the surface temperatures in Antarctica have been rising at an abnormal rate, a more in-depth analysis of other stations' recordings as well as sea levels and glacial melting could provide better insight on global warming trends.

Introduction

For the past few years, global temperatures have continued to rise drastically over the years, inducing serious natural disasters and environmental repercussions. One of the areas most affected by this global change is the Southern Pole. Increasing water and air temperatures have caused glaciers to melt and have significantly altered the ecosystem and its inhabitants such as fish, polar bears, and penguins. Furthermore, the melting of the ice caps could be linked to an increased prevalence of violent storms, flooding, and increased sea levels and temperatures.

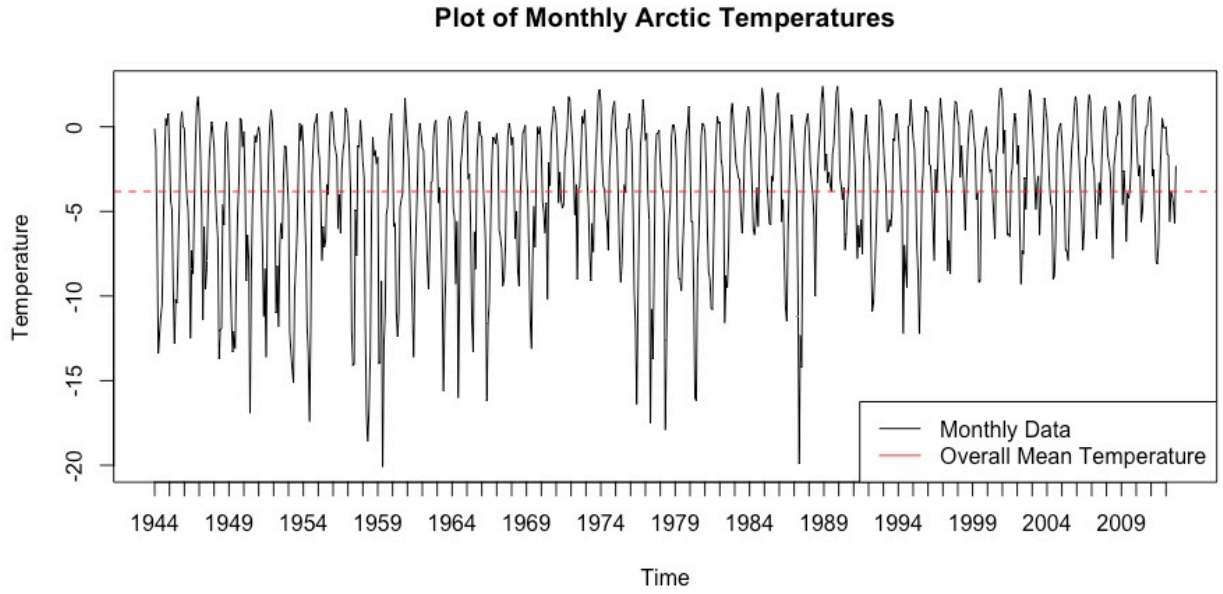
To quantify these changes, I looked at monthly surface temperature means from March of 1944 to November of 2012 at the Faraday station in Antarctica, with the ultimate goal of accurately forecasting future temperature fluctuations. The Faraday station, which was known as the Vernadsky station until 1977, is a research base on the Galindez Island of the Argentine Islands in Antarctica. The base has operated for over seventy years and continues to record meteorological, seismic, atmospheric, biological, and glacial changes in Antarctica, making the data it has collected a unique and valuable resource for mapping long-term temperature changes.

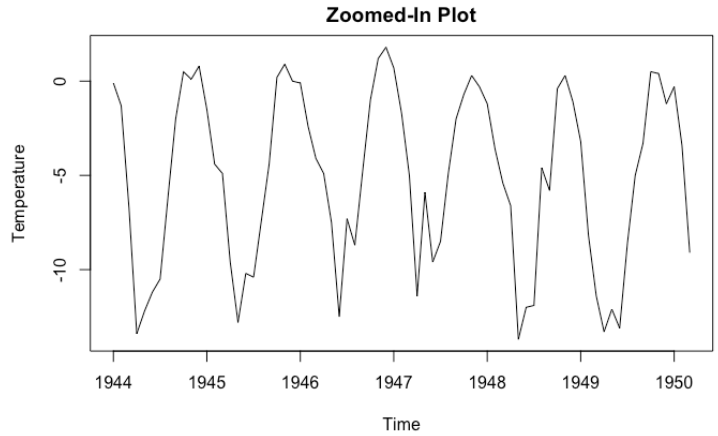
To model the data, I used Box-Jenkins seasonal and non-seasonal autoregressive integrated moving average (SARIMA and ARIMA) models to capture the changes in the average monthly measurements, properly process any homogenous non-stationarity that might be present (small clusters of similarly-behaving data points), and forecast future temperature data points. In order to utilize the Box Jenkins model, a Box-Cox transformation was applied to the time series to better achieve stationarity and Normality. Next, the optimal model was selected by examining the autocorrelation and partial autocorrelation plots of various models as well as by using pre-built functions in R to find the optimal moving averages and autoregressive orders. The models were then evaluated using the corrected Akaike's Information Criterion (AICc) and Bayesian Information Criterion (BIC) as well as tests of normality and stationarity. The best model was then used to forecast 40 observations ahead and when plotted, appears to represent the actual data very well.

The data was published by the British Antarctic Survey under the Natural Environment Research Council at www.bas.ac.uk as a means of promoting and fostering polar research, awareness, and conservation. To model, analyze and visualize the data, I used R Studio software. Furthermore, I used packages such as MASS, forecast, qpcR, and dplyer for their built-in time series and data manipulation functions. I also referenced Robert Shumway and David Stoffer's textbook *Time Series Analysis and its Applications*, 3rd Edition and greatly utilized the methods and material presented in lecture slides by Professor Raya Feldman at UC Santa Barbara.

Pre-Processing & Initial Data Exploration

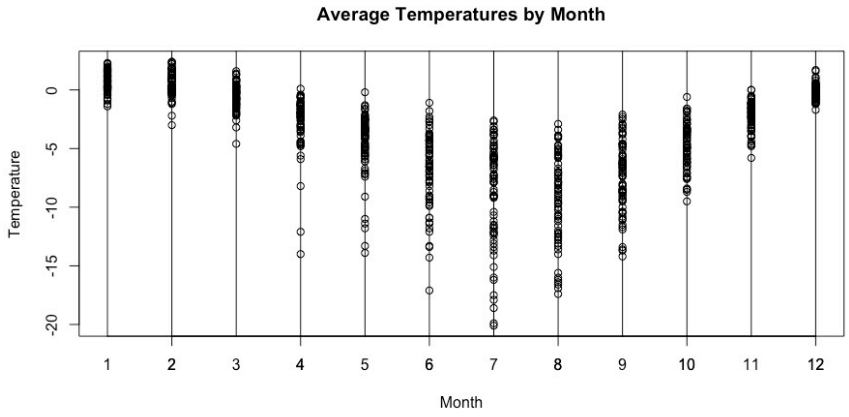
To begin analysis, I first load the data set into R and perform initial data pre-processing to restructure the data and remove missing values. I then graph the ordered data to visually demonstrate any blatant patterns or changes in variance.



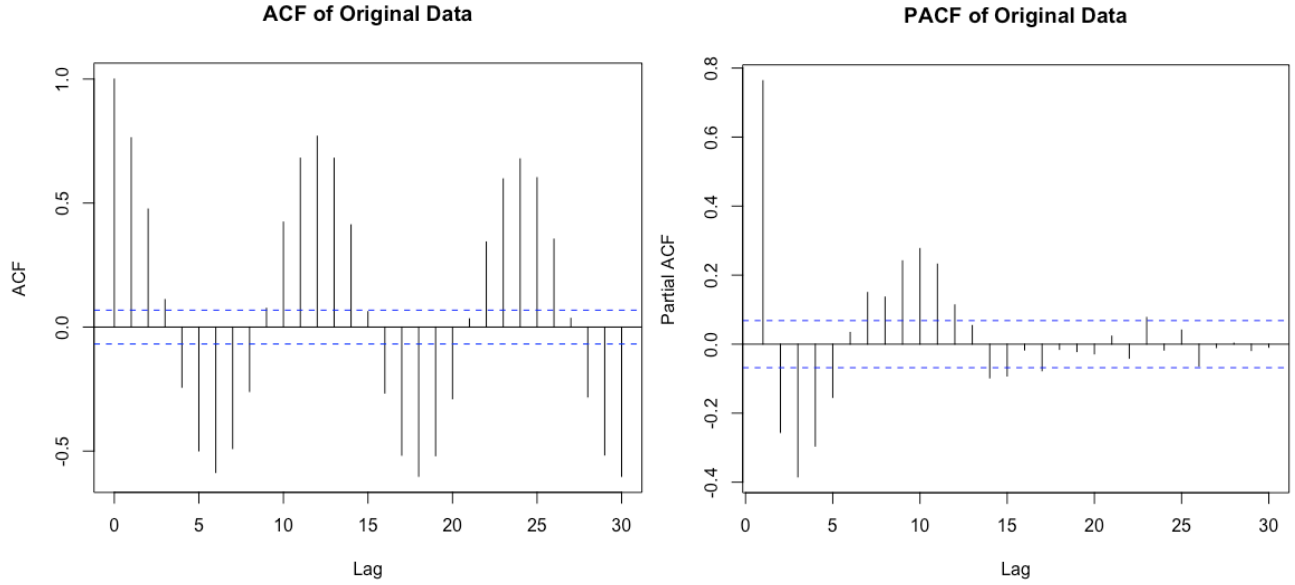


At first glance, the time series appears non-stationary due to a slight decrease in variance in the last 15 years and slight upwardly sloped trend. However, both of these effects are relatively small and will require further analysis to be confirmed. Furthermore, because of the seasonal nature of

temperature data as well as an apparent cyclical change in temperatures throughout the year (more apparent in the second graph), I suspect seasonality. To check this, I plot the temperatures by month.



The plot above shows that the mean and spread of the temperatures does in fact vary on a month-dependent basis. It appears that the mean temperature follows a concave quadratic pattern and that the spread of the temperatures is greatest in the summer months, between June and September. The lowest temperatures appear to occur in the summer months as well, which is as expected since the station is located in the Southern Hemisphere. I will later address the trend by adjusting the model for seasonality.

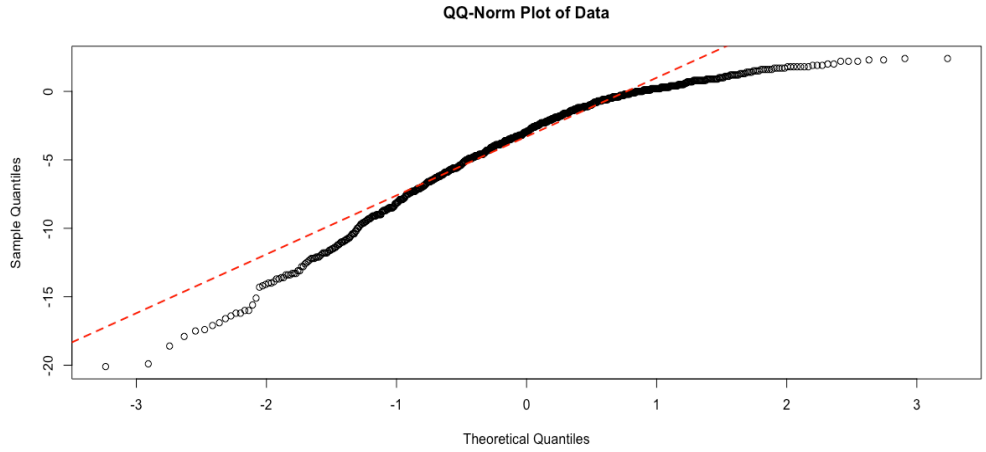


To form a preliminary estimation of possible models, I first looked at two correlograms above: the partial autocorrelation and autocorrelation plots.[1] The autocorrelation function (ACF) calculates the dependency of one lag to past lags, or in other words, the correlation of one observation to an observation *h* lags away. The ACF rotates between cycles of positive lags and negative lags every four to seven periods, indicating that the data is fairly dependent or slow to change. Additionally, the ACF does not suddenly cut off, but instead seems to decrease slightly over a long period of

1 https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

time. This slow decay in the ACF indicates possible non-stationarity.[2] Furthermore, the partial autocorrelation function (PACF) is similar to the ACF but accounts for intervening or "chain reaction" correlations.[3] The PACF plot shows oscillations that decay somewhat quickly but do not seem to cut off. The structures of the ACF and PACF suggest that the data could an autoregressive integrated moving averages (ARIMA) model with a high autoregressive order *p*.

ARIMA is a type of Box Jenkins model, in which the time series is differenced to remove trend and then fitted to account for both the moving averages and autoregressive components. Because Box Jenkins methodology assumes normality and stationarity, I checked the quantile-quantile plot for Normality, in which the dots should roughly follow the straight line.



Clearly form the plot above, our data does not appear to be Normal. The cluster of dots in the middle above the line and the tails of the dots below the dashed line indicate that our time series likely non-Gaussian tails. As an effort to obtain more stationary, Normal data with lower variance, I can apply different transformations to the data.

Data Transformation

As mentioned earlier, transformations can be a crucial step in achieving stationarity and accurately modeling a time series. As noted before, the spread of the oscillations appears to decrease towards the latter part of the time interval, indicating non-stationarity. As an effort to stabilize the variance, I applied a Box-Cox transformation to the data, using the formula:

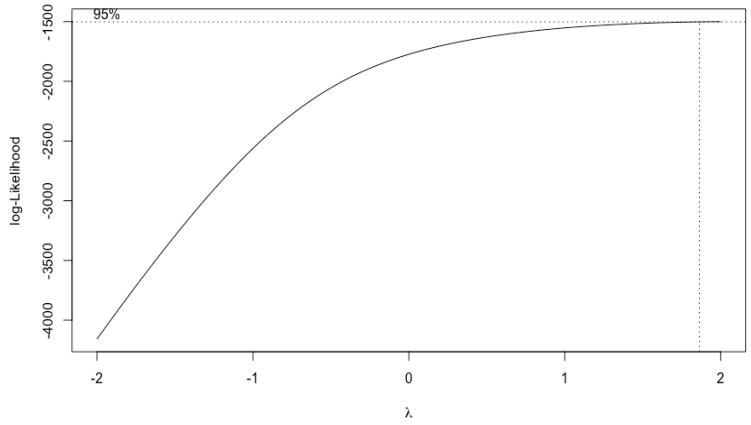
$$\begin{cases} Y_t = \frac{1}{\lambda}\left(X_t^{\lambda} - 1\right) & \text{if } \lambda \neq 0 \\ Y_t = \ln X_t & \text{if } \lambda = 0 \end{cases}$$

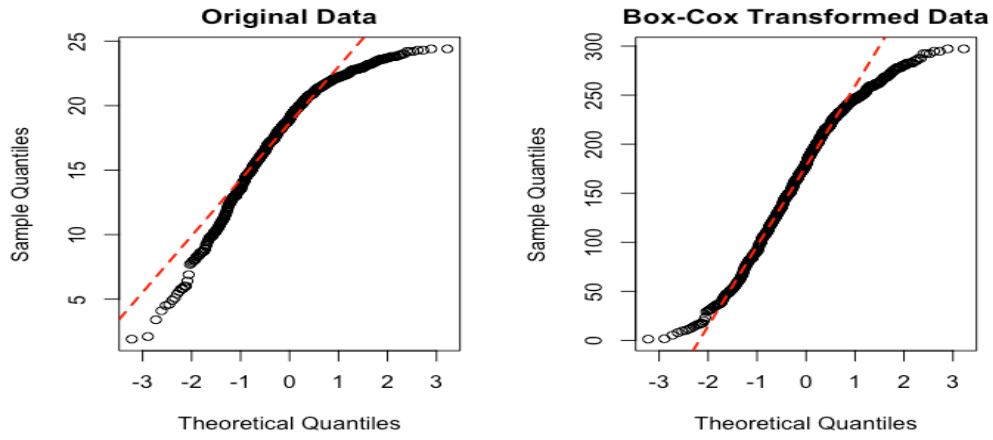
The optimal λ for the transformation is found by applying the boxcox function to display the log likelihood of the power parameter such that we can find the maximum log-likelihood estimator of λ. Because some of the data points were negative (negative degrees Celsius), I apply a shift of twenty-one degrees such that all observations are positive. The purpose of the boxcox function is to monotonically transform the data so that it roughly follows a Normal distribution, allowing for

2 https://www.otexts.org/fpp/8/1

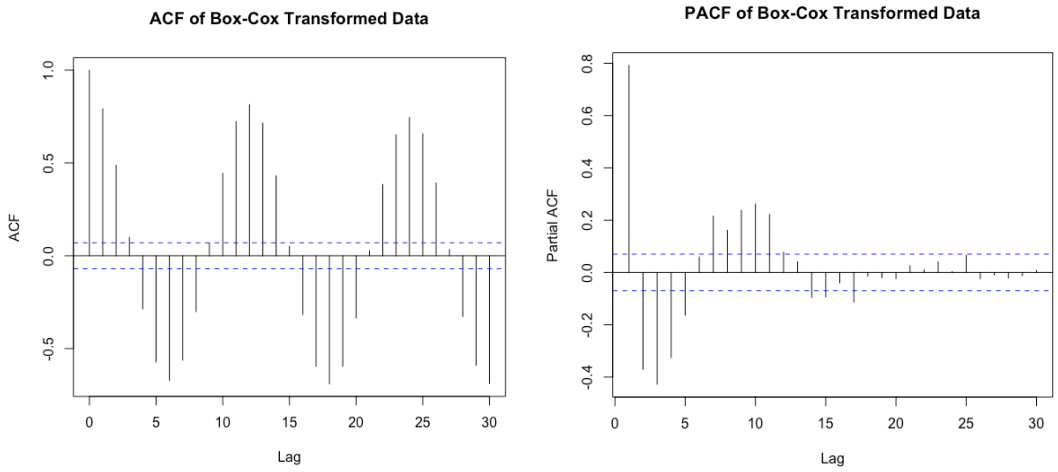
3 http://people.duke.edu/~rnau/411arim3.htm

normality assumptions and therefore more robust tests. From the plot below, it appears that $\lambda = 2$ is the optimal parameter to achieve this.



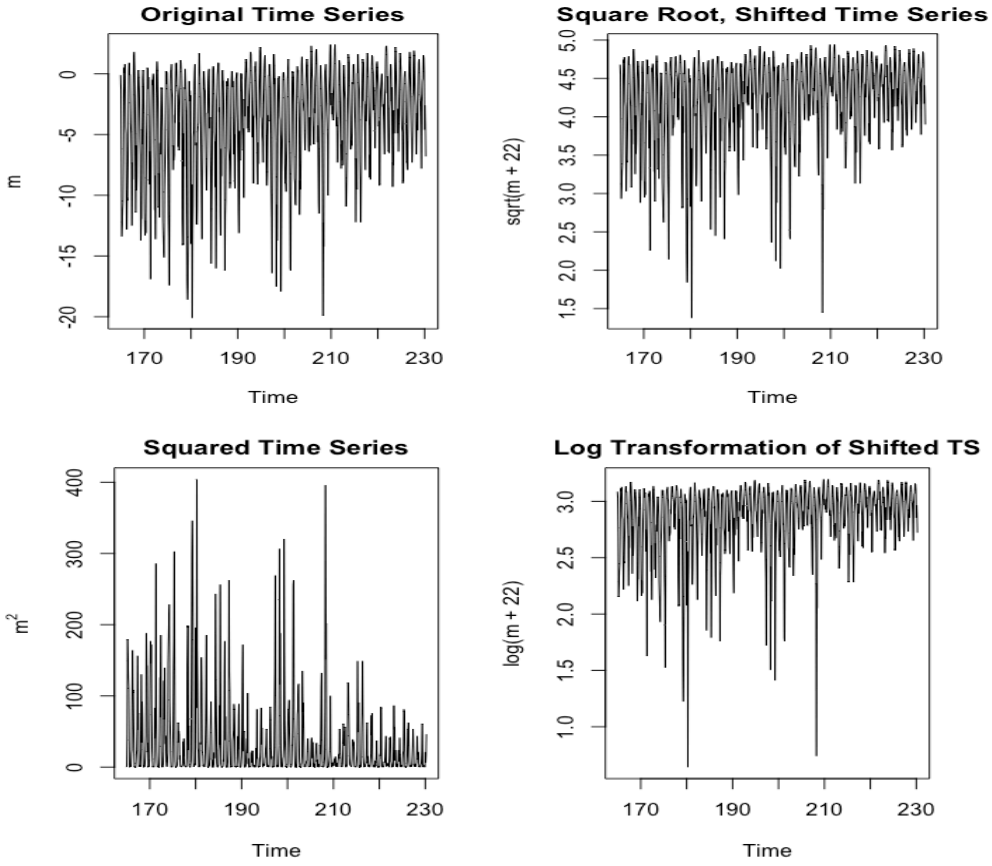


Our transformed data, Y_t, has a sample variance of 4902.394. This occurred because the lambda was greater than 1, thereby magnifying the data and its spread. The resulting ACF and PACF are as follows:



We can see that both the ACF and the PACF are the same shape as before, indicating the transformation did not alter the order or underlying form of the data. Because the Box Cox

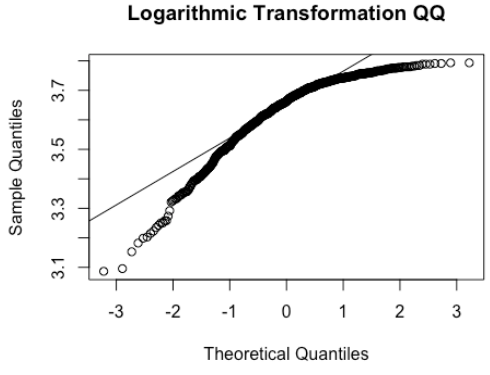
transformation increased the variance so significantly, I decided to explore other transformations below as well.



Note that when lambda equals zero in the Box Cox function, one simply takes the natural log of the data, which is equivalent to the fourth graph. From these transformations, I obtained the following samples variances:

var(X_t) = 18.66899
var($\sqrt{X_t + 22}$) = 0.1278024
var($X_t{}^2$) = 19609.58
var(log(X_t+22))= 0.01339194

The natural logarithmic transformation produces the lowest variances, however, the transformed data does not appear to follow a Gaussian distribution. Therefore I conclude to use the Box Cox transformation.



Seasonal Decomposition and Differencing

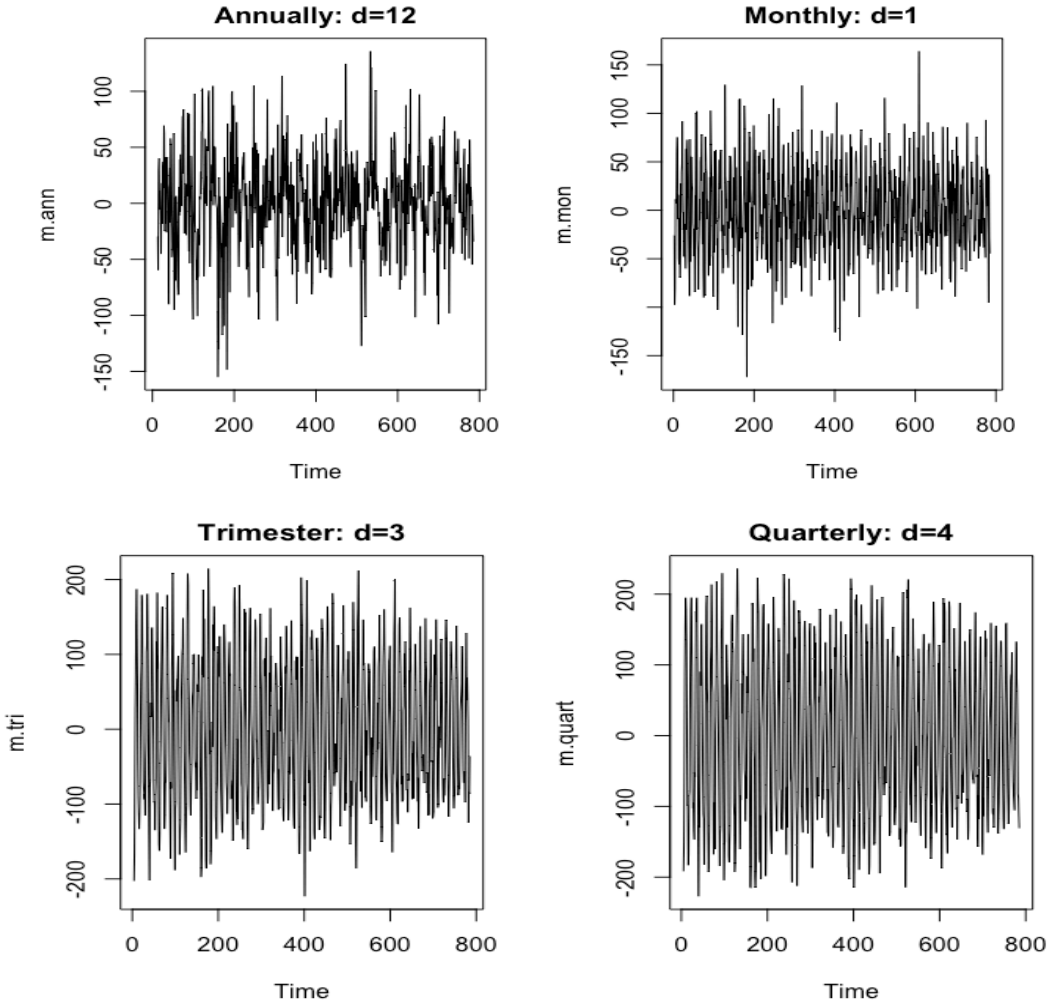
To deal with the seasonal or periodic aspect of the temperature data, I follow the classical decomposition model, which assumes that the data can be deconstructed into three separate components: trend (m_t), a seasonal component(s_t), and the stationary process (S_t).

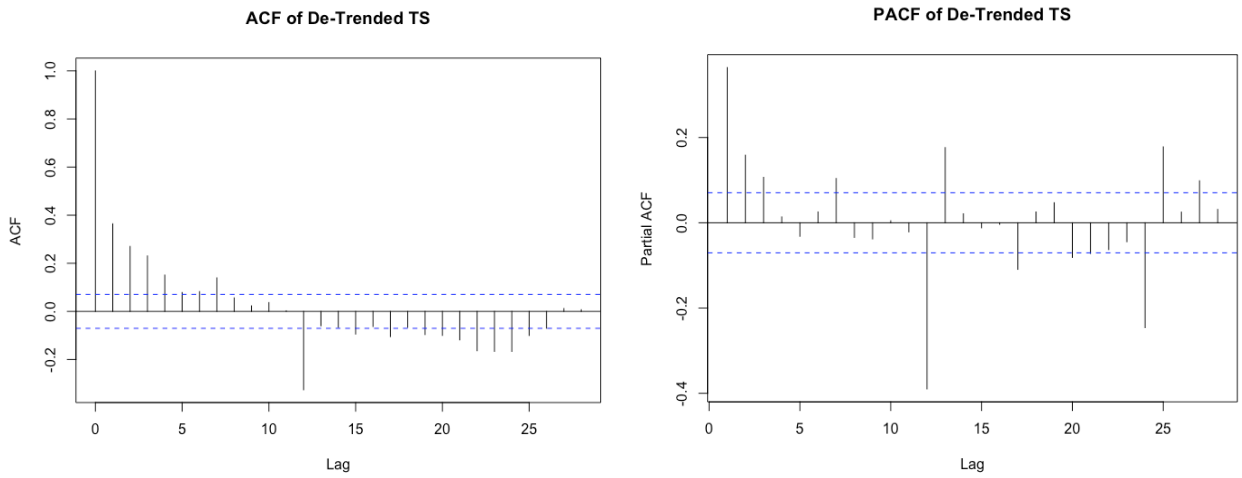
$$X_t = m_t + s_t + S_t$$

To remove the seasonal component, I difference at lag *d*, in which *d* is the suspected periodicity or seasonality:

$$\nabla_d X_t = (1 - B^d) X_t$$

After trying *d*=1,3,4 and 12 and calculating the resulting variances for each, I determine that a periodicity of 12 is the optimal value for *d*. This is surprising considering that our initial plot of the temperatures by months showed more of a quarterly pattern. Regardless, I continue with the transformed time series differenced at lag 12 for seasonality.





The ACF shows decay and then a spike at lag 12, which could point to a MA(1) component. On the other hand, the PACF tapers somewhat with spikes around 12 and 24 (which are multiples of our period). This suggests that the model has a somewhat higher AR order and likely is constituted of both seasonal and non-seasonal terms. The sample variance after differencing at lag 12 is 1679.359, which is a significant reduction from the variance of the original Box-Cox transformed time series of 4902.394. Next I want to check whether there is any trend—the m_t component—and if so, what the optimal number of differences is. To do this, I apply the *ndiffs* function and see that the time series does not need any further differencing. This makes sense as from the plots above, there does not appear to be any linear or quadratic trend and the data looks fairly stationary. To double-check these results, I look at the variances with f(t) being the Box-Cox transformation:

var($f(X_t)$) = 4902.394
var($\nabla_1 f(X_t)$) = 2035.19
var($\nabla_{12} f(X_t)$) = 1679.359
var($\nabla_{12}\nabla_1 f(Xt)$) = 2135.611

Looking at the last variance, it is clear that differencing again after removing seasonality increases the variance, indicating over-differencing. Therefore the best model is the de-trended, non-differenced model: $\nabla_{12} f(X_t)$.

Model Construction and Parameter Estimation

I now want to get a better picture of the orders of the seasonal and non-seasonal AR and MA terms. Calling the *ar* function on the box cox transformed data with the Yule Walker method produced an estimated optimal autoregressive order of 17. However, when using the same function with the Maximum Likelihood Estimator (MLE) method, I obtained an estimate of 12. Furthermore, when tested on the non-seasonal data the optimal orders were 27 and 12 respectively. I instead decide to construct many different models and use the corrected Akaike's Information Criterion(AICc) and the Bayesian Information Criterion (BIC) to evaluate them. Below are the best four, evaluated based on which had the lowest AICc and BIC scores.

Model	AICc	BIC
ARIMA(8,1,3)	7472.175	7527.806
ARIMA(17,0,1)	7516.34	7604.071
***SARIMA(8,0,1) x $(0,1,1)_{12}$**	**7387.36**	**7438.229**
SARIMA (8,1,2) x $(2,0,0)_4$	7477.504	7537.737

* best model

The fourth model, a SARIMA (8,0,1) x $(0,1,1)_{12}$, has the lowest AICc and BIC, so I will choose this one to be the primary model for testing and forecasting. Using the R *arima* function, the following model coefficient estimations are produced:

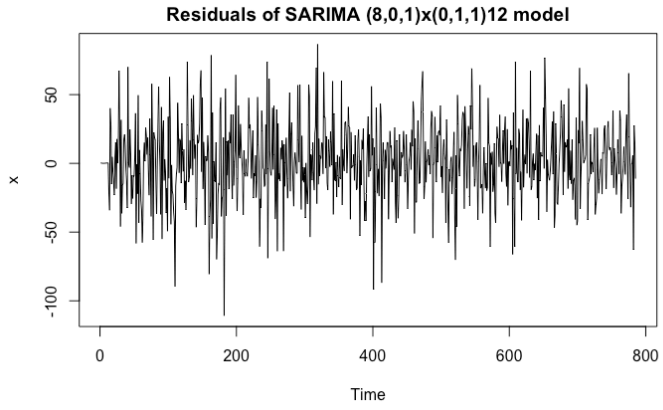
```
arima(x = m.bc, order = c(8, 0, 1), seasonal = list(order = c(0, 1, 1), period = 12))
Coefficients: ar1      ar2      ar3      ar4      ar5     ar6     ar7      ar8      ma1
              1.1456  -0.1592  -0.0284  -0.0356  -0.0712  0.0146  0.1112  -0.0310  -0.7474
s.e.          0.1013   0.0669   0.0566   0.0555   0.0551  0.0557  0.0554   0.0425   0.0936
sma1         -0.9512
s.e.          0.0156
sigma^2 estimated as 776.2:  log likelihood = -3682.54,  aic = 7387.08
```

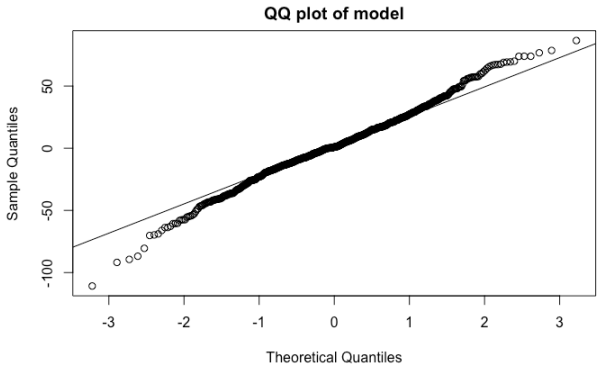
So, rounded, the final model is:

$$\mathbf{1.45X_t - 0.16X_{t-1} - 0.03X_{t-2} - 0.04X_{t-3} - 0.07X_{t-4} + 0.01X_{t-5} - 0.11X_{t-6} - 0.03X_{t-7} = -0.75Z_t - 0.95Z_{t-12}}$$

Diagnostic Checking:

Diagnostic checking aims to test the adequacy of the model in representing the actual data. To ensure the performance of the previous defined SARIMA model, I performed several diagnostic checks, such as the Shapiro Wilks test of Normality, the Ljung Box test, the Box Pierce test, the McLeod-Li test, and examined a qq-norm plot to check the Normality of the residuals.





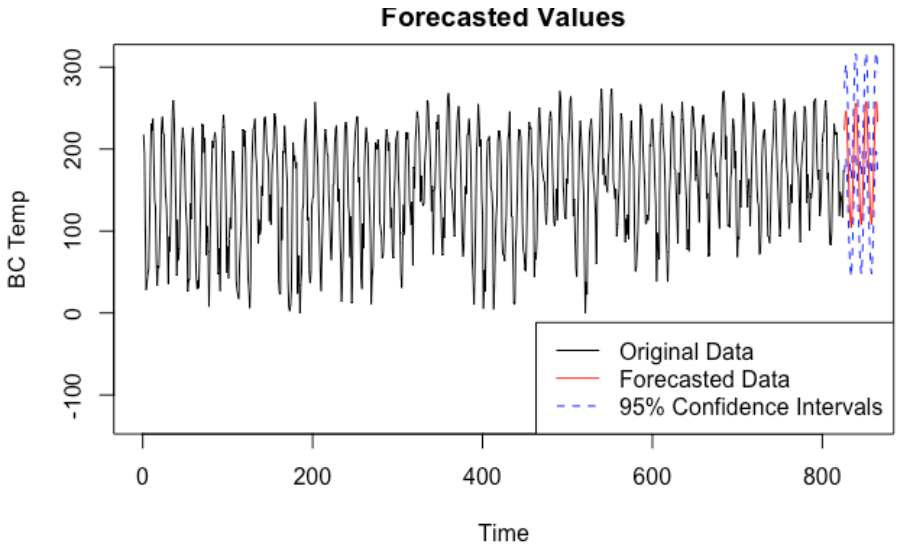

```
Shapiro-Wilk normality test
data:  x W = 0.9918, p-value = 0.0002421

Box-Ljung test  data:  x
X-squared = 0.0043247, df = 1, p-value = 0.9476

Box-Pierce test  data:  x
X-squared = 0.0043082, df = 1, p-value = 0.9477

Box-Pierce test  data:  (x)^2
X-squared = 4.2362, df = 1, p-value = 0.03957
```

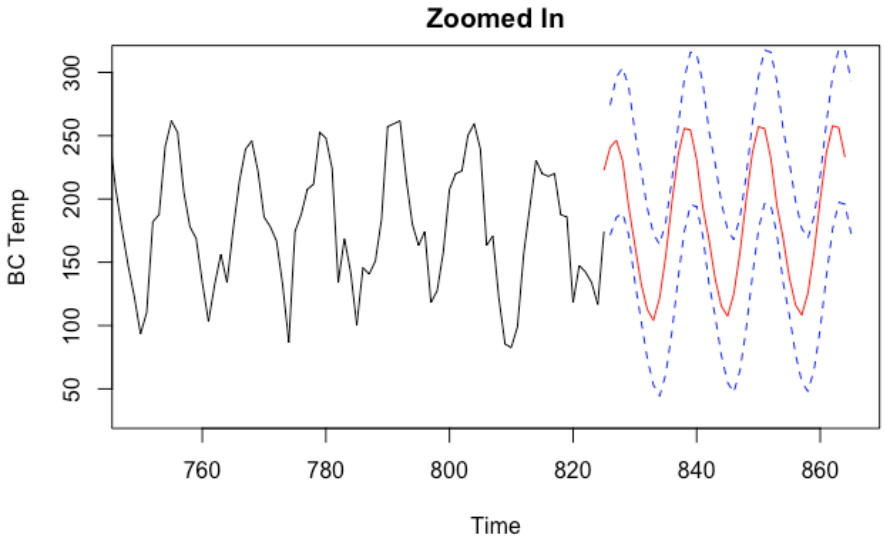
The residuals do not show any distinct pattern or change in variability and therefore can be succinctly concluded as to resemble a White Noise process. Additionally, the quantile-quantile plot shows that the residuals appear to be roughly Normally distributed. Furthermore, the Box Ljung and Box Pierce (jointly known as Portmanteau tests) both show p-values far larger than 0.05, leading us to fail to reject the null hypothesis that the residuals follow a White Noise process. Lastly, it is important to also check the squares of the residuals, as sometimes these are not caught by the previous two tests. Here, unfortunately, the test fails, as the p-value of 0.03957 is less than 0.05. Although the final diagnostic test failed, the other evaluations all passed, so I will continue with the forecasting and remark on the outcomes in the conclusion.

Forecasting

To test the model constructed, I forecast 40 months into the future and then compare the predictions to the test data that was excluded from the model building process. To make the predictions, I used the predict function in R with our fitted model and specified the steps ahead to 40. To create the confidence interval for the predictions, I used the following formula:

$$Confidence\ Interval = pred \pm 2 * se(pred)$$





Conclusion

After a in-depth analysis of the time series data as well as evaluation of a large selection of potential representative models, I chose a SARIMA $(8,0,1) \times (0,1,1)_{12}$ model to fit the data:

$$\mathbf{1.45X_t - 0.16X_{t-1} - 0.03X_{t-2} - 0.04X_{t-3} - 0.07X_{t-4} + 0.01X_{t-5} - 0.11X_{t-6} - 0.03X_{t-7} = -0.75Z_t - 0.95Z_{t-12}}$$

The model did not pass the McLeod-Li test for squared residuals—however—the model did pass the rest of the diagnostic tests. In terms of forecasting, the model clearly produced forecasts that were slightly shifted up from where the actual data left off. This could either be due to a fault in the indexing or the model itself. Despite the offset, the forecast seems to follow the sinuous shape of the real data very well. On a different note, because there was no upward linear trend present after differencing for seasonality, I was not able to conclude that the mean monthly surface temperatures recorded at the station have increased significantly over the past 70 years. However, more recent data as well as data from other stations in the South Pole could be extremely useful in furthering such research.

Online References

Data:

1. http://www.nerc-bas.ac.uk/icd/gjma/faraday.temps.html
2. www.bas.ac.uk

Time Series Learning Material:

3. https://www.otexts.org/fpp/8/1
4. http://people.duke.edu/~rnau/411arim3.htm
5. https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

Appendix

```
library(qpcR) # modeling and analysis; AICC

library(dplyr) # data manipulation

library(MASS) # boxcox, confidence interval
library(forecast) # forecasting
library(astsa) # times series


df<-read.csv('arctictemps.txt', header=TRUE, sep = ",") # read in data set
sdf<-stack(df, drop=FALSE) # stack columns
sdf$year<- seq(from = 1944, to= 2012, by=1) # create columns with years

missVal<-c(1,70,828) # mark rows with missing values (3 NAs)
sdf<-sdf[-missVal,] # remove rows with missing values

monthsAbb <- list(jan=1,feb=2,mar=3,apr=4,may=5,jun=6,jul=7,aug=8,sep=9,oct=10,
nov=11,dec=12) # translate month abbreviations to numbers for easier sorting
monthNum <- function(x) {x <- tolower(x)
  sapply(x,function(x) monthsAbb[[x]])
} # function to turn month abbreviations to lower case and then replace each wi
th it's corresponding number
sdf$ind<-monthNum(sdf$ind)

sdf<-sdf[with(sdf, order(ind)),] # order the data frame by month
sdf<-sdf[with(sdf, order(year)),] # order the data frame by year
head(sdf)

##     values ind year
## 139   -0.1   3 1944
## 208   -1.3   4 1944
## 277   -6.7   5 1944
## 346  -13.4   6 1944
## 415  -12.2   7 1944
## 484  -11.2   8 1944
```

```r
mon.ts<- ts(sdf$values) # create the time series object
length(mon.ts) # 825 observations total
```

```
## [1] 825
```

```r
plot(y=sdf$values, x=sdf$ind, xlab="Month", ylab="Temperature", main="Average T
emperatures by Month") # monthly averages and spread
xticks <- seq(1, 12, 1)
axis(1, at = xticks, labels = xticks,  tck=1)
```

```r
qqnorm(sdf$values, main="QQ-Norm Plot of Data") # qq plot
qqline(sdf$values, col=2,lty=2, lwd=2)
```

```r
# initial time series plotting
plot(mon.ts, xlab="Time", ylab="Temperature", type='l', main='Plot of Monthly A
rctic Temperatures', xaxt="n")
xaxis2<- seq(1, 825, 12)
axis(1, at = xaxis2, labels = seq(1944,2012, by=1))
abline(h=mean(mon.ts), col=2, lty=2, lwd=1)
legend("bottomright", c("Monthly Data", "Overall Mean Temperature"), col=c("bla
ck", "red"), lty=1)
# Not stationary, there appears to be sudden spikes, as well a decrease in vari
ability and a slight overall increase in temperature with time

plot(mon.ts[1:75], xlab="Time", ylab="Temperature", type='l', main='Zoomed-In P
lot', xaxt="n") # zoomed in plot to show seasonality
xaxis2<- seq(1, 75, 12)
axis(1, at = xaxis2, labels = seq(1944,1950, by=1))

# initial ACF and PACF
acf(mon.ts, lag.max=30, main="ACF of Original Data") # the ACF tails off very s
lowly; this could indicate a long memory process

pacf(mon.ts, lag.max=30, main="PACF of Original Data") # the PACF tails off som
wehat after lag 15 with a few significant lags occurring after (i.e.: 33, 38)

# Because both the ACF and PACFs tail off slowly, the times series likely fallo
ws an ARIMA model
```

```r
m.shft <-m.train$values+21 # shift by 21 degrees so that all data is positive
int <- as.numeric(1:length(m.shft)) # time interval
m <-ts(m.shft) # create time series object of all positive values
length(m) # 785 observations in training set
```

```
## [1] 785
```

```r
span = 1:length(m) # span of time interval
bc = boxcox(m~span, plotit=TRUE) # computes and plots optimal lambda for transf
ormation

lmbda = bc$x[which(bc$y == max(bc$y))] # 2
m.bc = (m^lmbda - 1 ) / lmbda # lambda must be less than 1 to reduce variance
var(m.bc)
```

```
## [1] 4324.837
```

```r
var(m)
```

```
## [1] 18.66899
```

```r
par(mfrow = c(1,2))
qqnorm(m, main="Original Data")
qqline(m, col=2,lty=2, lwd=2)
qqnorm(m.bc, main="Box-Cox Transformed Data")
qqline(m.bc, col=2,lty=2, lwd=2)

par(mfrow = c(1,2))
plot(m.bc, main="Box Cox Transformed")
plot(m, main="Original Time Series")

par(mfrow = c(1,1))
acf(m.bc, lag.max=30, main="ACF of Box-Cox Transformed Data")

pacf(m.bc, lag.max=30, main="PACF of Box-Cox Transformed Data")

par(mfrow = c(1,2))
## other transformations
plot(m, main= "Original Time Series")
plot(sqrt(m+21), main= "Square Root, Shifted Time Series" )

plot(m^2, main= "Squared Time Series", ylab=expression(m^2))
plot(log(m+21), main= "Log Transformation of Shifted TS")

var(m) #18.66899
```

```
## [1] 18.66899
```

```r
var(sqrt(m+21)) # 0.3474679
```

```
## [1] 0.131452
```

```r
var(m^2) # 3284.823
```

```
## [1] 17299.35
```

```r
var(log(m+21)) # 0.1225263
```

```
## [1] 0.01498488
```

```r
par(mfrow = c(1,1))
qqnorm(log(m+21), main="Logarithmic Transformation QQ")
qqline(log(m+21))
```

```r
# difference at different lags
m.ann <-diff(m.bc, lag=12)
m.mon <-diff(m.bc, lag=1)
m.tri <-diff(m.bc, lag=3)
m.quart <-diff(m.bc, lag=4)
m.biann <-diff(m.bc, lag=6)

# check all the variances
var(m.bc)
```

```
## [1] 4324.837
```

```r
var(m.ann)
```

```
## [1] 1461.19
```

```r
var(m.mon)
```

```
## [1] 1785.135
```

```r
var(m.tri)
```

```
## [1] 7815.015
```

```r
var(m.quart)
```

```
## [1] 11167.56
```

```r
var(m.biann)
```

```
## [1] 14514.93
```

```r
# plot
par(mfrow = c(1,2))
plot(m.ann, main="Annually: d=12")
plot(m.mon, main="Monthly: d=1")

plot(m.tri, main="Trimester: d=3")
plot(m.quart, main="Quarterly: d=4")

# acf and pacf of d = 12 differenced, box-cox transformed time series
acf(m.ann, main="ACF of De-Trended TS")
pacf(m.ann, main="PACF of De-Trended TS")
```

```r
var(m.bc)
```

```
## [1] 4324.837

var(diff(m.bc, lag=1))

## [1] 1785.135

var(diff(m.bc, lag=12))

## [1] 1461.19

var(diff(m.ann, lag=1))

## [1] 1853.393

# using yule walker & AIC
yw.order <- ar(m.bc, aic = TRUE, order.max = NULL, method="yule-walker")
yw.order # order 17

## Order selected 17  sigma^2 estimated as  852.7

# using MLE & AIC
mle.order <- ar(m.bc, aic = TRUE, order.max = NULL, method="mle")

## Order selected 12  sigma^2 estimated as  715.6

ar(m.ann, aic = TRUE, order.max = NULL, method="yule-walker") # 27
## Order selected 27  sigma^2 estimated as  890.1

ar(m.ann, aic = TRUE, order.max = NULL, method="mle") # 12

## Order selected 12  sigma^2 estimated as  1010

ndiffs(m.bc) # difference for trend?

## [1] 1

ndiffs(m.ann) # difference for trend?

## [1] 0

### Model Construction and order estimation

## Series: m.bc
## ARIMA(2,1,2)
## sigma^2 estimated as 902.9:  log likelihood=-3780.5
## AIC=7571.01   AICc=7571.08   BIC=7594.33

### arima models
mod1 <- arima(m.bc, order=c(1,0,1))
mod2 <- arima(m.bc, order=c(8,0,2))
mod3 <- arima(m.bc, order=c(2,1,2))
mod4 <- arima(m.bc, order=c(17,1,1))
mod7 <- arima(m.ann, order=c(1,0,1))
mod8 <- arima(m.bc, order=c(8,1,3))
```

```r
mod9 <- arima(m.bc, order=c(2,1,2))


## [1] 7361.436

#### sarima models

s_mod1 <- arima(m.bc, order=c(2,0,0), seasonal = list(order = c(2, 0, 0), perio
d=0))
s_mod2 <- arima(m.bc, order=c(1,0,1), seasonal = list(order = c(0, 0, 0), perio
d = 12))
s_mod3<- arima(m.bc, order=c(8,1,2), seasonal= list(order=c(0,0,0), period=12))
s_mod4<- arima(m.bc, order=c(8,1,1), seasonal= list(order=c(0,1,1), period=12))
s_mod7<- arima(m.bc, order=c(8,0,1), seasonal= list(order=c(0,1,1), period=12))

s_mod5<- arima(m.bc, order=c(8,1,2), seasonal= list(order=c(2,0,0), period=4))
s_mod6<- sarima(m.bc, p=1, d=0, q=1, S=12)

### model evaluation using BIC and corrected AIC
BIC(mod1) # 8090.312

## [1] 7985.874

BIC(mod2) # 7545.894

## [1] 7434.933

BIC(mod3) # 7703.859

## [1] 7594.329

BIC(mod4) # 7604.071

## [1] 7494.45

BIC(mod7) # 7822.367

## [1] 7713.472

BIC(mod8) # 7527.806 ###

## [1] 7417.067

AICc(mod1) # 8071.68

## [1] 7967.242

AICc(mod2) # 7490.248

## [1] 7379.287

AICc(mod3) # 8136.858

## [1] 7571.058
```

```
AICc(mod4) # 7516.34
## [1] 7406.719
AICc(mod7) # 7803.797
## [1] 7694.902
AICc(mod8) # 7472.175 ###
## [1] 7361.436
AICc(s_mod1) # 8016.013
## [1] 7910.846
AICc(s_mod2) # 8071.68
## [1] 7967.242
AICc(s_mod3) # 7495.494
## [1] 7384.251
AICc(s_mod4) # 7392.929 ###
## [1] 7283.129
AICc(s_mod5) # 7555.23
## [1] 7366.664
AICc(s_mod7) # 7387.36
## [1] 7277.105
BIC(s_mod4)
## [1] 7333.983
BIC(s_mod3)
## [1] 7435.275
BIC(s_mod5)
## [1] 7426.898
BIC(s_mod7)
## [1] 7327.973


#### Model Diagnostics
myModel<-arima(m.bc, order=c(8,0,1), seasonal= list(order=c(0,1,1), period=12))
```

```
#### tests & diagnostic checking
x= myModel$residuals

plot(x, main="Residuals of SARIMA (8,0,1)x(0,1,1)12 model")

shapiro.test(x) # Normality of residuals

##
##  Shapiro-Wilk normality test
##
## data:  x
## W = 0.99289, p-value = 0.0008522

Box.test(x, type="Ljung-Box") # Ljung Box Test: reject white noise hypothesis i
f p-value <0.05

##
##  Box-Ljung test
##
## data:  x
## X-squared = 0.004424, df = 1, p-value = 0.947

Box.test(x, type="Box-Pierce") #

##
##  Box-Pierce test
##
## data:  x
## X-squared = 0.0044071, df = 1, p-value = 0.9471

Box.test((x)^2, type="Box-Pierce") # McLeod-Li: check if squares of residuals a
re correlated

##
##  Box-Pierce test
##
## data:  (x)^2
## X-squared = 3.9461, df = 1, p-value = 0.04698

acf(x, main="ACF of Residuals")

pacf(x, main="PACF of Residuals")

qqnorm(x, main="QQ plot of model")
qqline(x)

# Apply model to data

myModel<-arima(m.bc, order=c(8,0,1), seasonal=list(order=c(0,1,1), period=12),
 method="ML", xreg= 1:length(m.bc))
```

```
pred = predict(myModel, n.ahead=40, newxreg=(length(m.bc)+1) : (length(m.bc)
+40)) # use prediction function to forecast next 40 months (about 5% of observa
tions)

up_ci = pred$pred + 2*pred$se # upper bound for the C.I. for transformed data

low_ci = pred$pred - 2*pred$se # lower bound for the C.I. for transformed data

ts.plot(m.bc,xlim=c(0,850),ylim=c(-130, 310), main="Forecasted Values", ylab="
BC Temp") # plot time series

lines(up_ci, col="blue", lty="dashed") # plot CIlines(low_ci, col="blue", lty=
"dashed") # plot CI

lines((length(m.bc)+1):(length(mon.ts)+40), pred$pred, col="red") # plot pre
dictions at end of real observations

legend("bottomright", c("Original Data", "Forecasted Data", "95% Confidence In
tervals"), col=c("black", "red", "blue"), lty=c(1,1,2)) # put in a legend



## zoomed in:

ts.plot(m.bc,xlim=c(750,865), ylim=c(30,310), main="Zoomed In", ylab="BC Temp"
)

lines(up_ci, col="blue", lty="dashed")

lines(low_ci, col="blue", lty="dashed")

lines((length(m.bc)):(length(m.bc)+39), pred$pred, col="red")
```