# Time Series Analysis of Sea Ice Levels in the Artic Circle

Conor Sharpe

In this project I will be analysing a data set showing the monthly volume of sea ice in the Arctic from January 1990 to March 2011 given in 1000km3. In order to record this data, mass balance measurements are made by buoys in the sea. Sea ice volume is of great interest to scientists because melting sea ice releases CO2 as well as causing sea levels to rise.

The data is available at: https://timeseries.weebly.com/data-sets.html

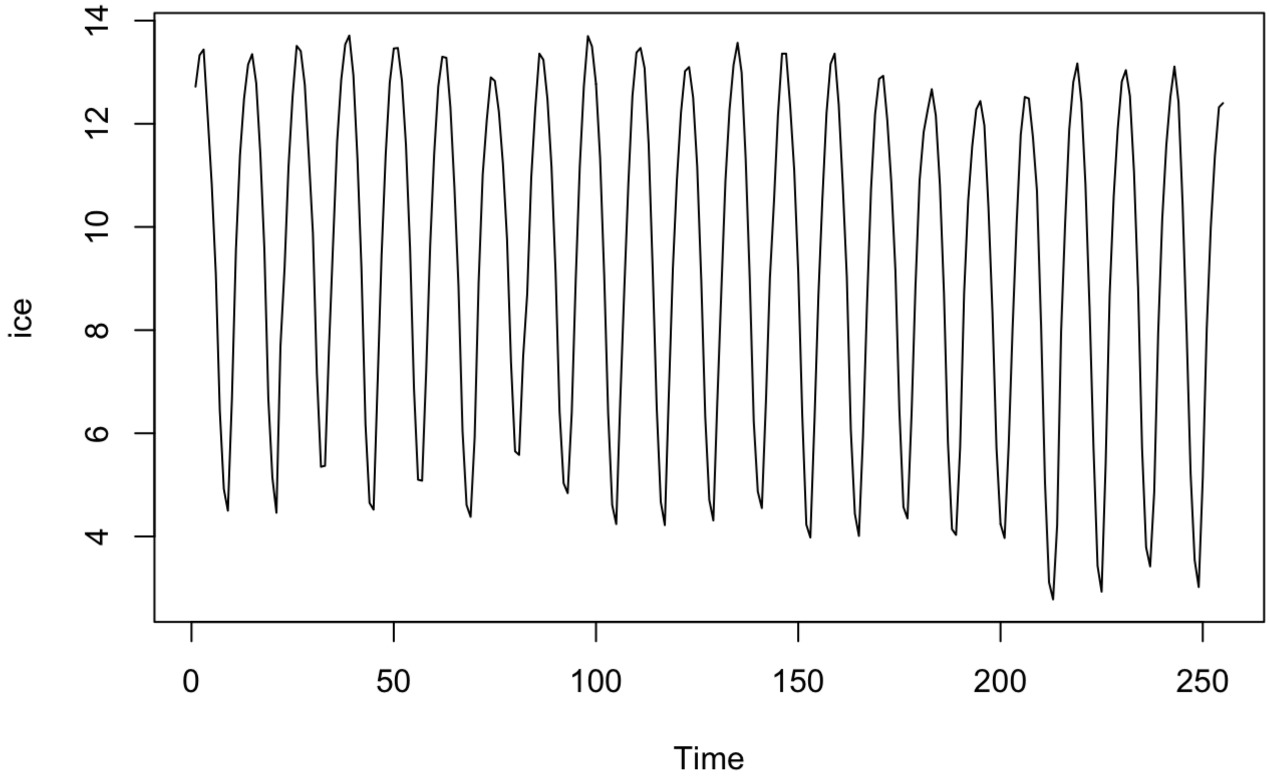
## 1. Exploratory Analyses

First we will import the data and analyse the time series plot, to do this we run the code:

```
sea_ice = read.csv("sea_ice.csv")

ice = sea_ice$Arctic

par(mfrow=c(1,1), pty="m")
ts.plot(ice)
```




Looking at the plot, there seems to be either no trend at all or a slight downward trend. The variance appears roughly constant over time and there seems to be a seasonality with a period of 12.

To investigate whether the data is stationary or not we will conduct an augmented Dickey–Fuller test. This is done using the code:

```
adf.test(ice)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ice
## Dickey-Fuller = -16.421, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
adf.test(ice , alternative = 'explosive')
```

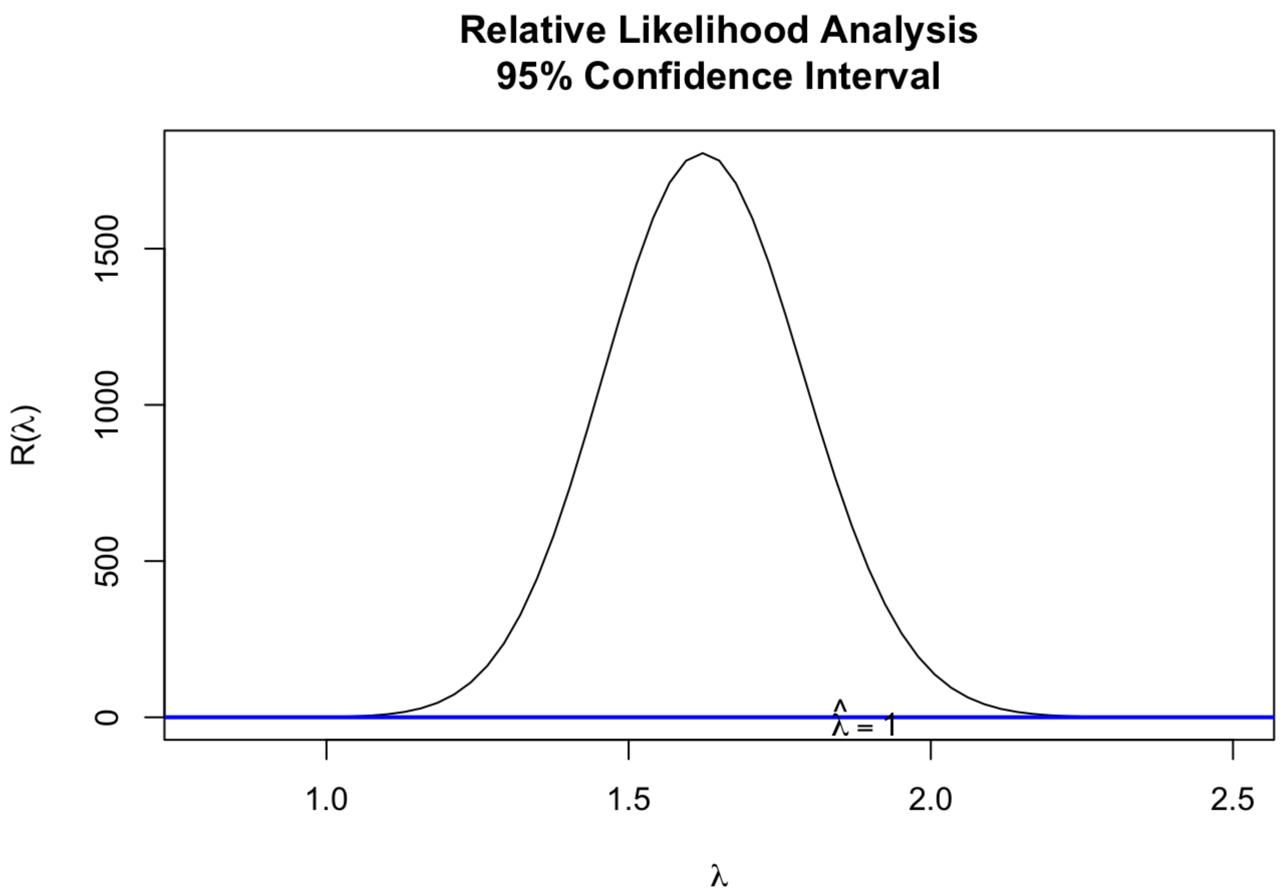
```
## Warning in adf.test(ice, alternative = "explosive"): p-value smaller than
## printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ice
## Dickey-Fuller = -16.421, Lag order = 6, p-value = 0.99
## alternative hypothesis: explosive
```

The test with the alternative hypothesis of stationary gave a test statistic of -16.421 and corresponding p-value of 0.01. So the evidence suggests that the data is stationary. The test with the alternative hypothesis of explosive gave a test statistic of -16.421 and corresponding p-value of 0.99. So the evidence suggests that the data is not explosive. Therefore, differencing will not be necessary when fitting a model.

Next we will investigate whether or not a variance stabilizing transformation is required. To do this we will run the code:

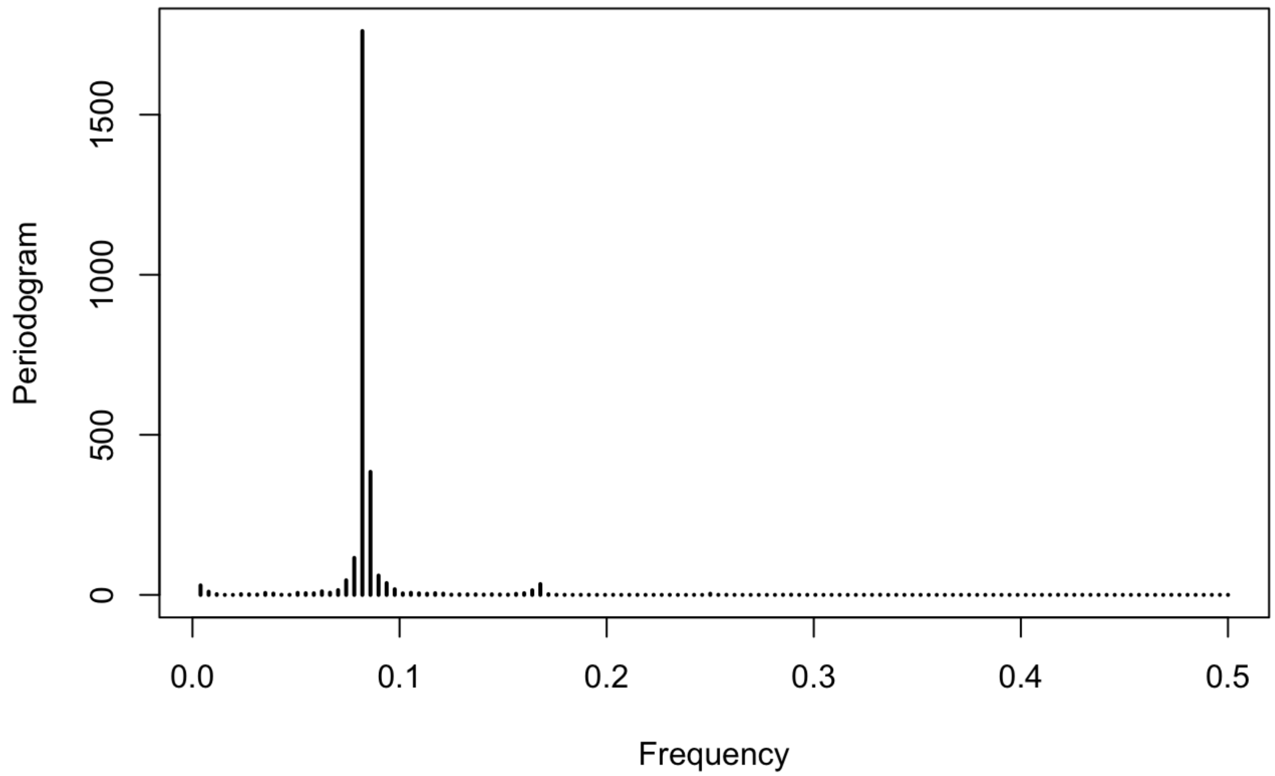
```
FitAR::BoxCox(ice)
```




The Box-Cox test recommends a lambda of 1 be used for the time series, this is equivalent to no transformation. This suggests that the data has a constant variance over time and a transformation is not required.

In order to investigate whether the data is seasonal we run the following code:

```
p=periodogram(ice)
```




We can see a clear spike around 0.8, this tells us that the time series is seasonal. In order to locate the spike we run the code:

```
p$freq[which.max(p$spec)]
```

```
## [1] 0.08203125
```

And then to see what period this corresponds to, we run the code:

```
1/p$freq[which.max(p$spec)]
```

```
## [1] 12.19048
```

This suggests that the period we should use for the model is 12.19, however using contextual knowledge that the data is recorded monthly and there are 12 months in a year we will use a period of 12 for the model.

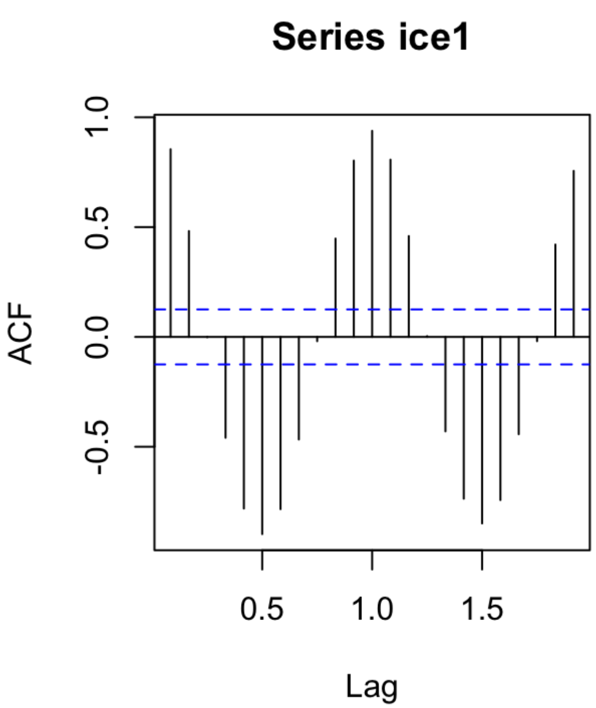
# 2. Sarima Model Fitting

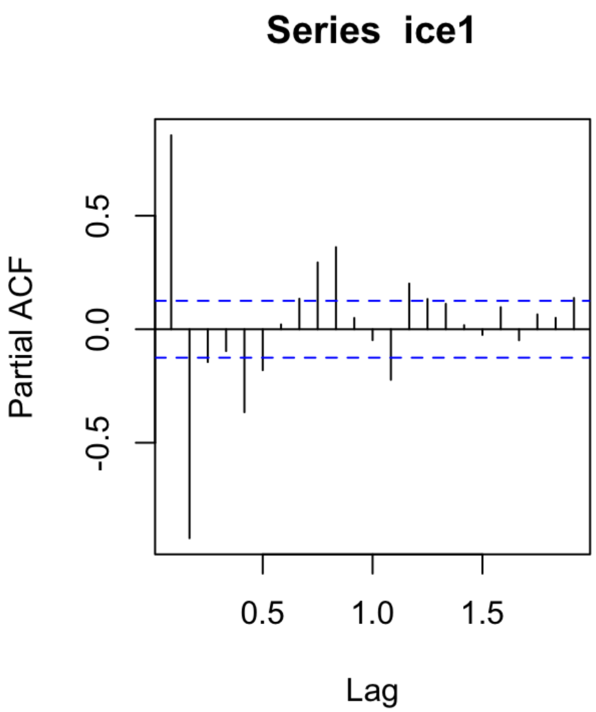
First we will split the time series into training and testing data, the first 245 values will make up the training data and the last 10 values the testing data. To do this we use the code:

```
ice1 = ts(ice[1:245] , freq=12)
ice2 = ts(ice[246:255] , freq=12 )
```

We will inspect the ACF and PACF plots in order to fit the SARIMA models, to do this we run the code:

```
par(mfrow=c(1,2),pty='s')
acf(ice1)
pacf(ice1)
```




## 2.1 Model 1

Seasonal ACF: Tails off Regular ACF: Tails off within period Seasonal PACF: Tails off Regualr PACF: Cuts off after lag 2 within period

SARIMA (2,0,0)(1,0,1)s=12

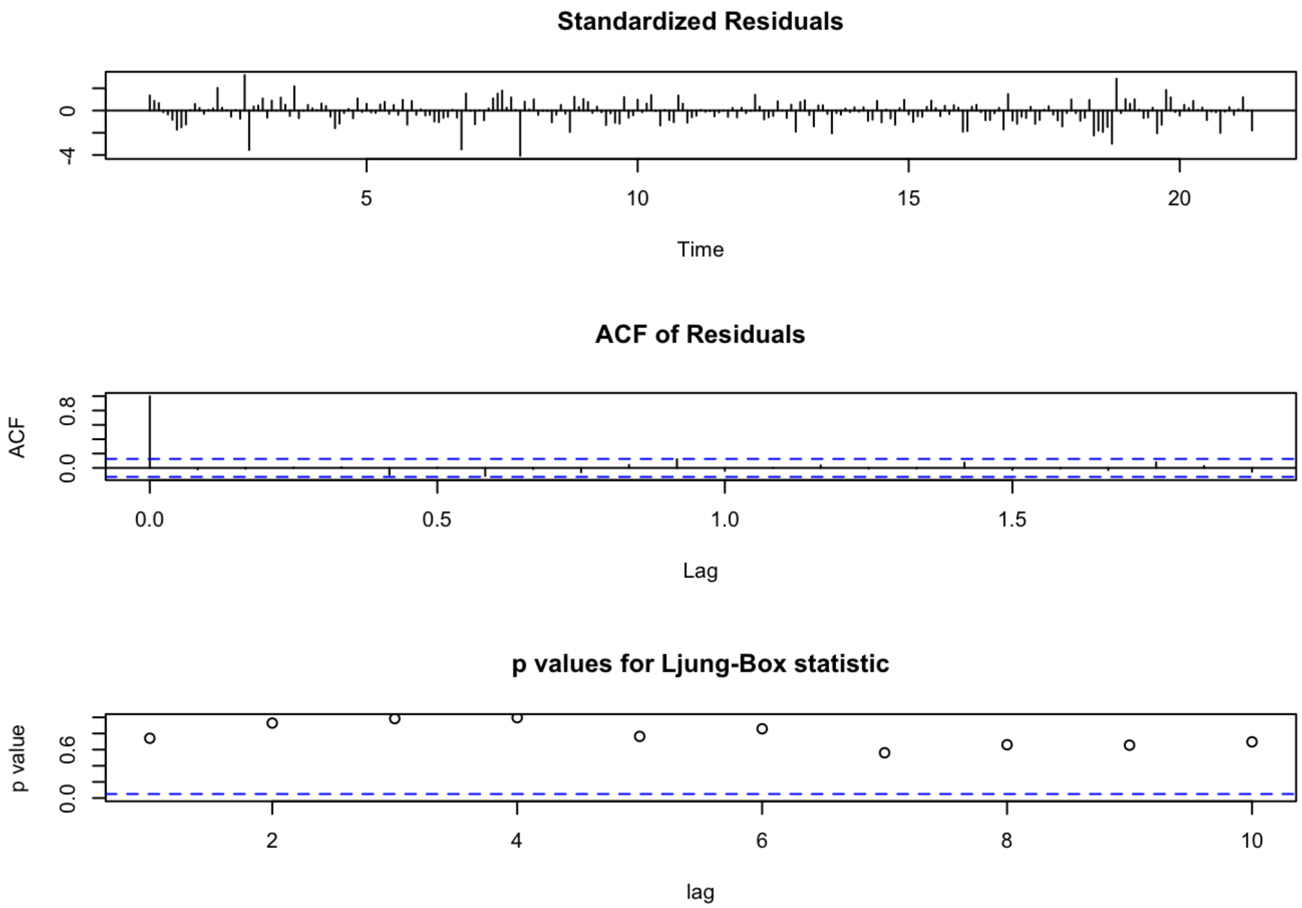
To fit this model we will use the code:

```r
m1=arima(ice1,order=c(2,0,0),seasonal=list(order=c(1,0,1),period=12), method="ML")
m1
```

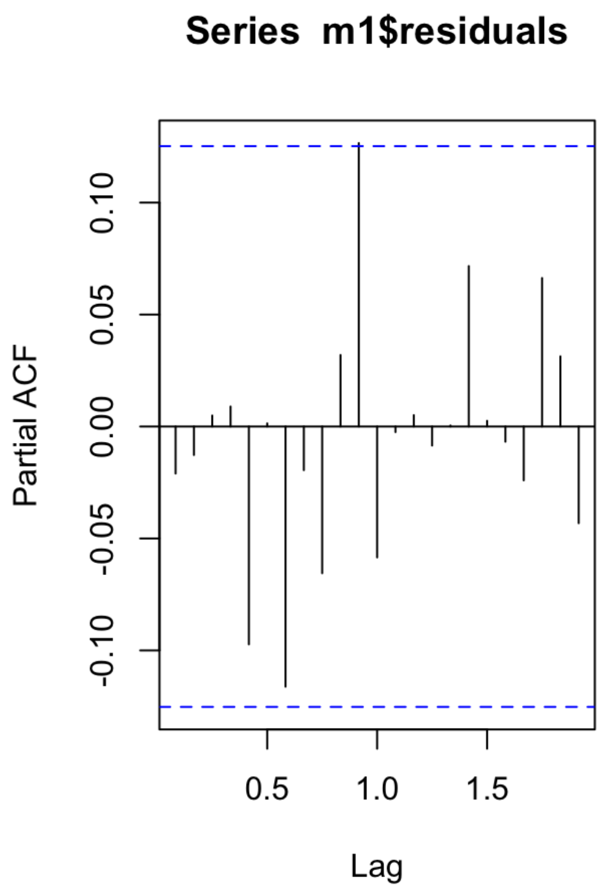
```
##
## Call:
## arima(x = ice1, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 1), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1      ar2    sar1     sma1  intercept
##       0.7889  -0.0568  0.9987  -0.6993     9.2954
## s.e.  0.0676   0.0671  0.0007   0.0528     1.7505
##
## sigma^2 estimated as 0.07674:  log likelihood = -58.91,  aic = 127.82
```

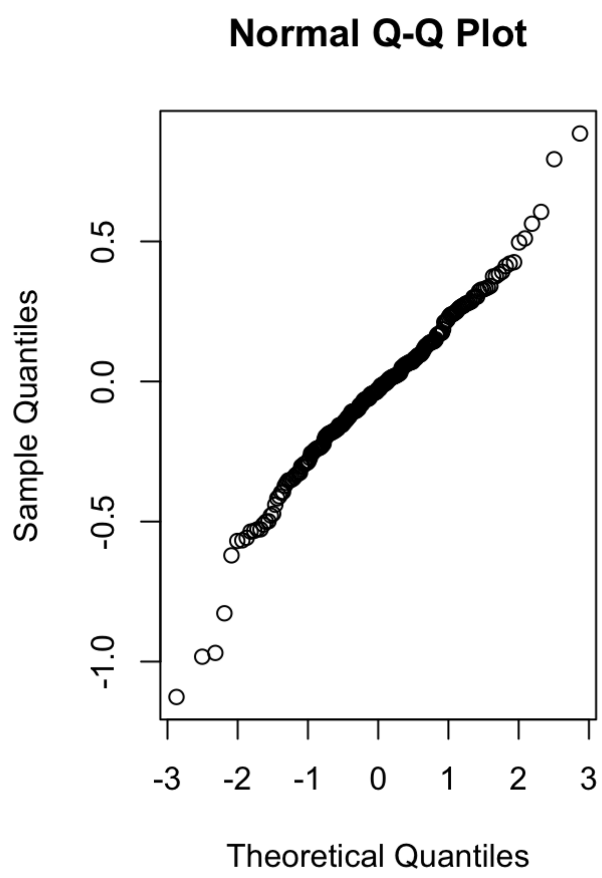
We will now run diagnostics on the model using the code:

```r
tsdiag(m1)
```




```
par(mfrow=c(1,2) , pty='m')
pacf(m1$residuals)
qqnorm(m1$residuals)
```




From the standardized residuals plot, we see most residual values lie in between -3 and 3, however there are some values close to -4 that may be considered outliers. Thus, we must continue with caution. There is no pattern and it is centered around 0. In ACF plot there are no significant values of serial correlation. Also, the p-values for Ljung-Box statistics are greater than significance level (=0.05) for all lags. So there is evidence that the SARIMA (2,0,0)(1,0,1)s=12 may be suitable for this time series data.

The PACF of residuals plot shows a significant value at lag 11, however this is only marginally significant, this suggests the model is a good fit. The Q-Q plot is a straight line, this also suggests the model is a good fit.

Using the model we will predict the next 10 data points, to do this we use the code:

```
m1.fore = predict(m1, n.ahead=10)
m1.fore
```

```
## $pred
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         8.341975  5.506141  3.680456
## 22 11.771928 12.632074 12.954104
##           Sep       Oct       Nov       Dec
## 21  3.377528  5.212983  8.284743 10.329425
## 22
##
## $se
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                        0.2770232 0.3528452 0.3860582
## 22 0.4160264 0.4165009 0.4167391
##           Sep       Oct       Nov       Dec
## 21 0.4017444 0.4093983 0.4131891 0.4150800
## 22
```

## 2.2 Model 2

Seasonal ACF: Tails off Regular ACF: Tails off within period Seasonal PACF: Tails off Regualr PACF: Tails off within period

SARIMA(1,0,1) (1,0,1)s=12

To fit this model we will use the code:

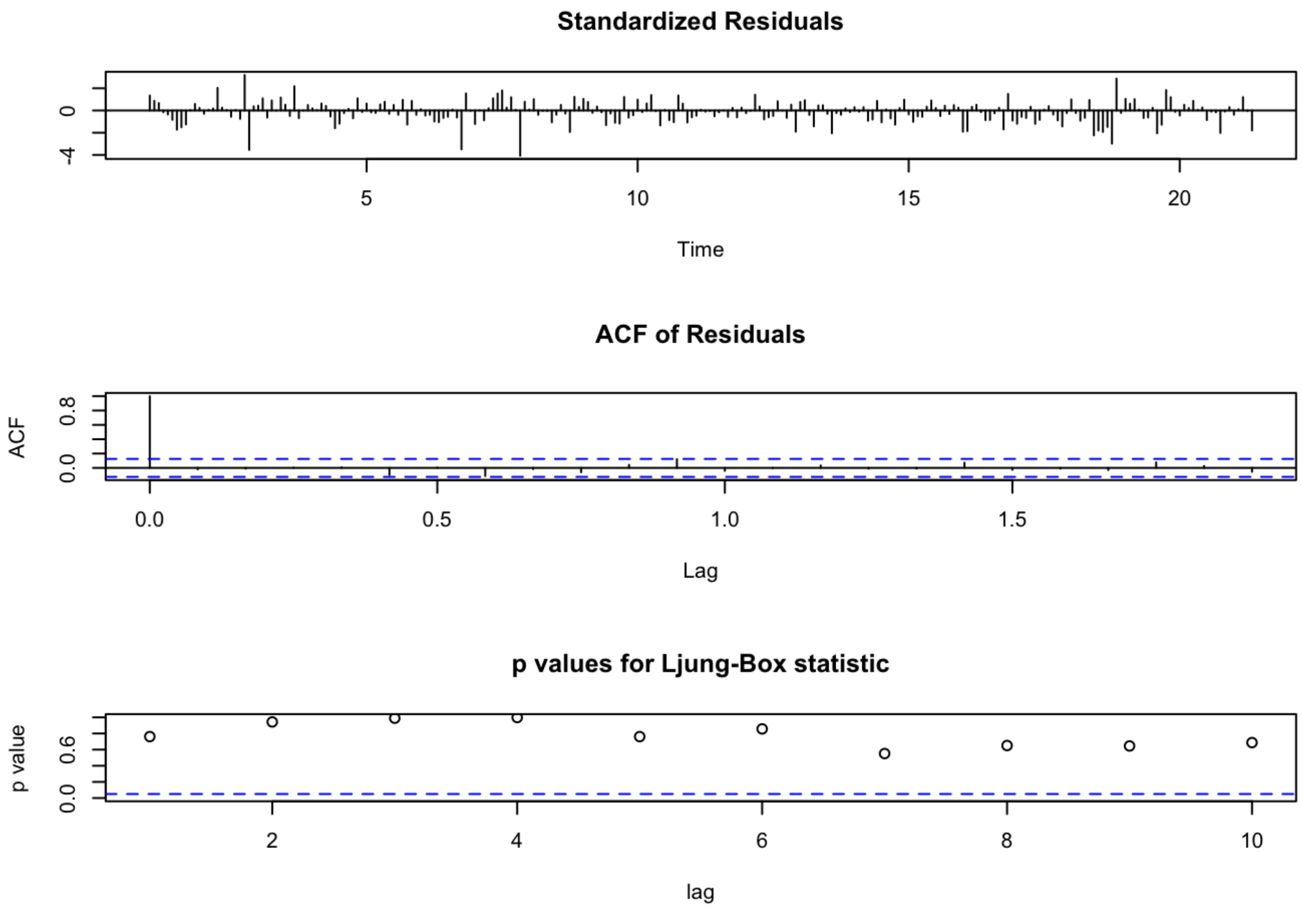
```
m2=arima(ice1,order=c(1,0,1),seasonal=list(order=c(1,0,1),period=12), method="ML")
m2
```

```
##
## Call:
## arima(x = ice1, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1), period = 12),
##     method = "ML")
##
```

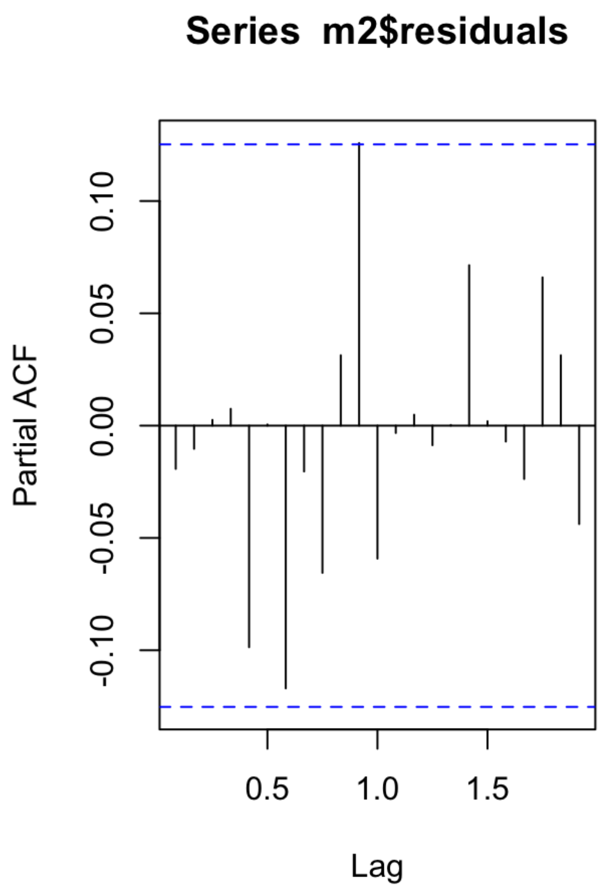
```
## Coefficients:
##          ar1     ma1    sar1     sma1  intercept
##       0.7142  0.0728  0.9987  -0.6996     9.2953
## s.e.  0.0621  0.0872  0.0007   0.0528     1.7714
##
## sigma^2 estimated as 0.0767:  log likelihood = -58.93,  aic = 127.85
```

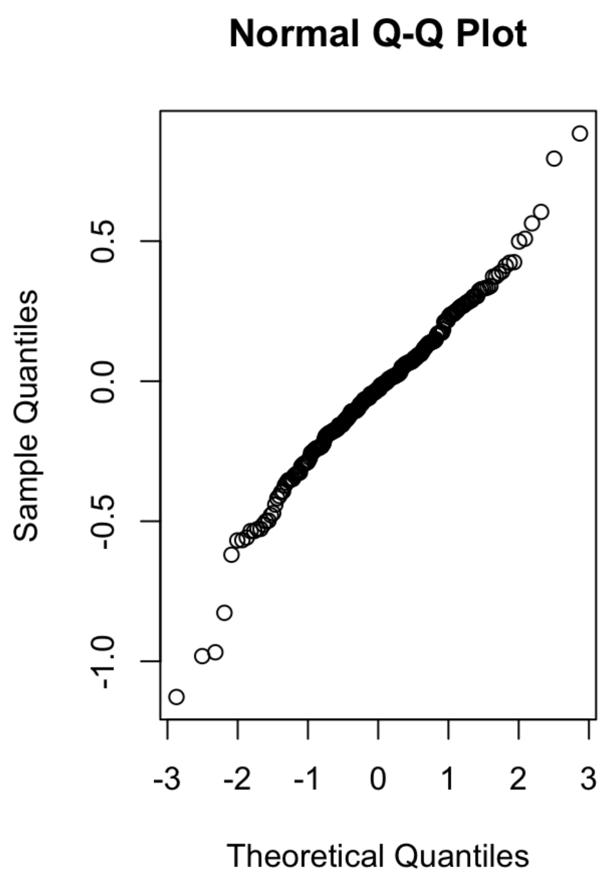
We will now run diagnostics on the model using the code:

```
tsdiag(m2)
```




```
par(mfrow=c(1,2) , pty='m')
pacf(m2$residuals)
qqnorm(m2$residuals)
```




From the standardized residuals plot, most residual values lie in between -3 and 3, however there are some values close to -4 that may be considered outliers. Thus, we must continue with caution. There is no pattern and it is centered around 0. In ACF plot, there are no significant values of serial correlation. Also, the p-values for Ljung-Box statistics are greater than significance level (=0.05) for all lags . So there is evidence that theSARIMA (1,0,1)(1,0,1)s=12 may be suitable for this time series data.

The PACF of residuals plot shows a significant value at lag 11 however this is only marginally significant, this suggests the model is a good fit. The Q-Q plot is a straight line, this also suggests the model is a good fit.

Using the model we will predict the next 10 data points, to do this we use the code:

```
m2.fore = predict(m1, n.ahead=10)
m2.fore
```

```
## $pred
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                       8.341975  5.506141  3.680456
## 22 11.771928 12.632074 12.954104
##           Sep       Oct       Nov       Dec
## 21  3.377528  5.212983  8.284743 10.329425
## 22
##
## $se
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                      0.2770232 0.3528452 0.3860582
## 22 0.4160264 0.4165009 0.4167391
##           Sep       Oct       Nov       Dec
## 21 0.4017444 0.4093983 0.4131891 0.4150800
## 22
```

## 2.3 Model 3

For our third model we will use the same model as Model 1 however we will add regular differencing, this will give the model SARIMA (2,1,0)(1,0,1)s=12. Although the ADF test suggested differencing was not necessary, a slight downward trend is visible on the time series plot and therefore it may be worth investigating. To fit the model we used the following code:

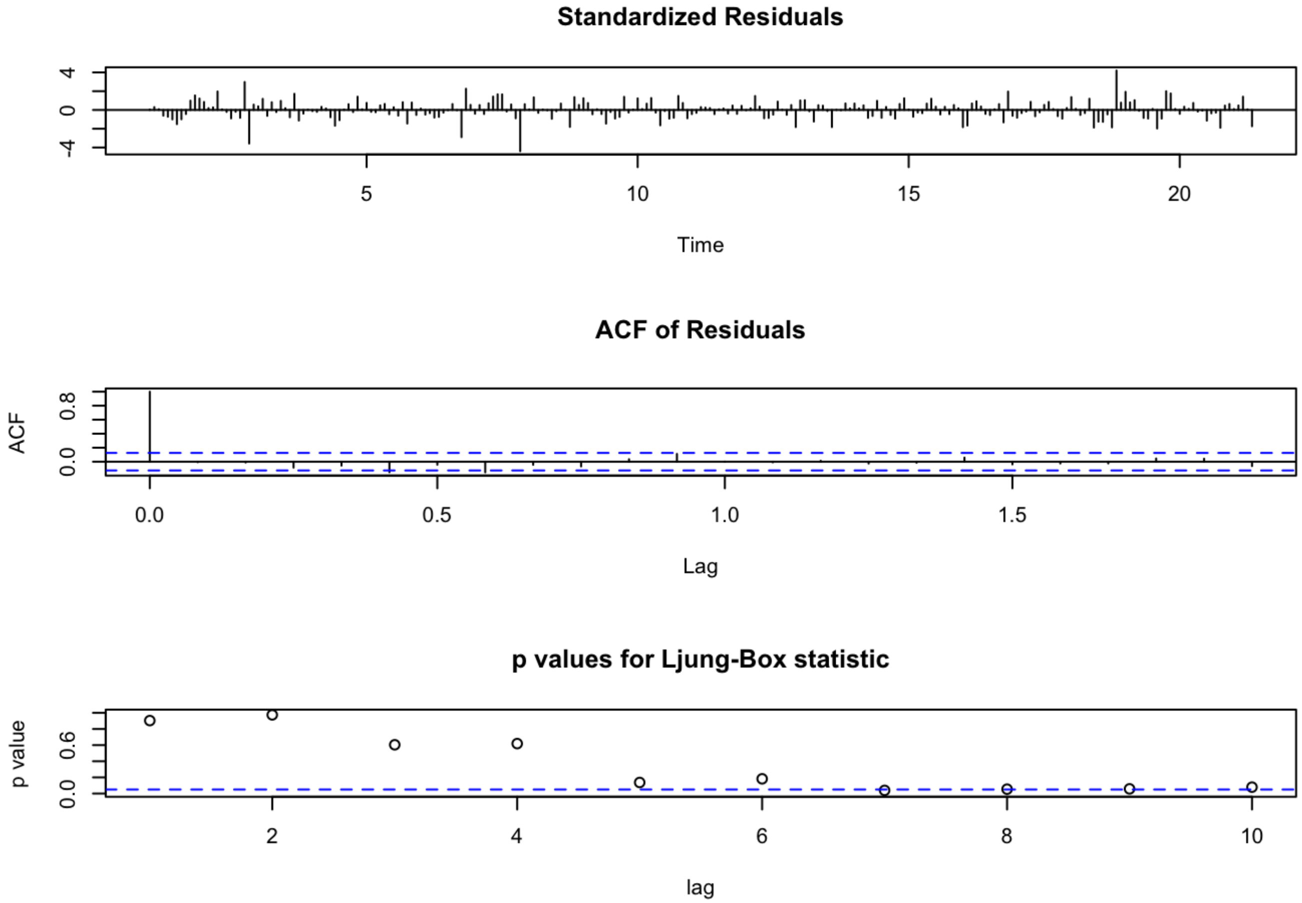
```
m3=arima(ice1,order=c(2,1,0),seasonal=list(order=c(1,0,1), period=12), method="ML")
m3
```

```
##
## Call:
## arima(x = ice1, order = c(2, 1, 0), seasonal = list(order = c(1, 0, 1), period = 12),
##     method = "ML")
##
## Coefficients:
```

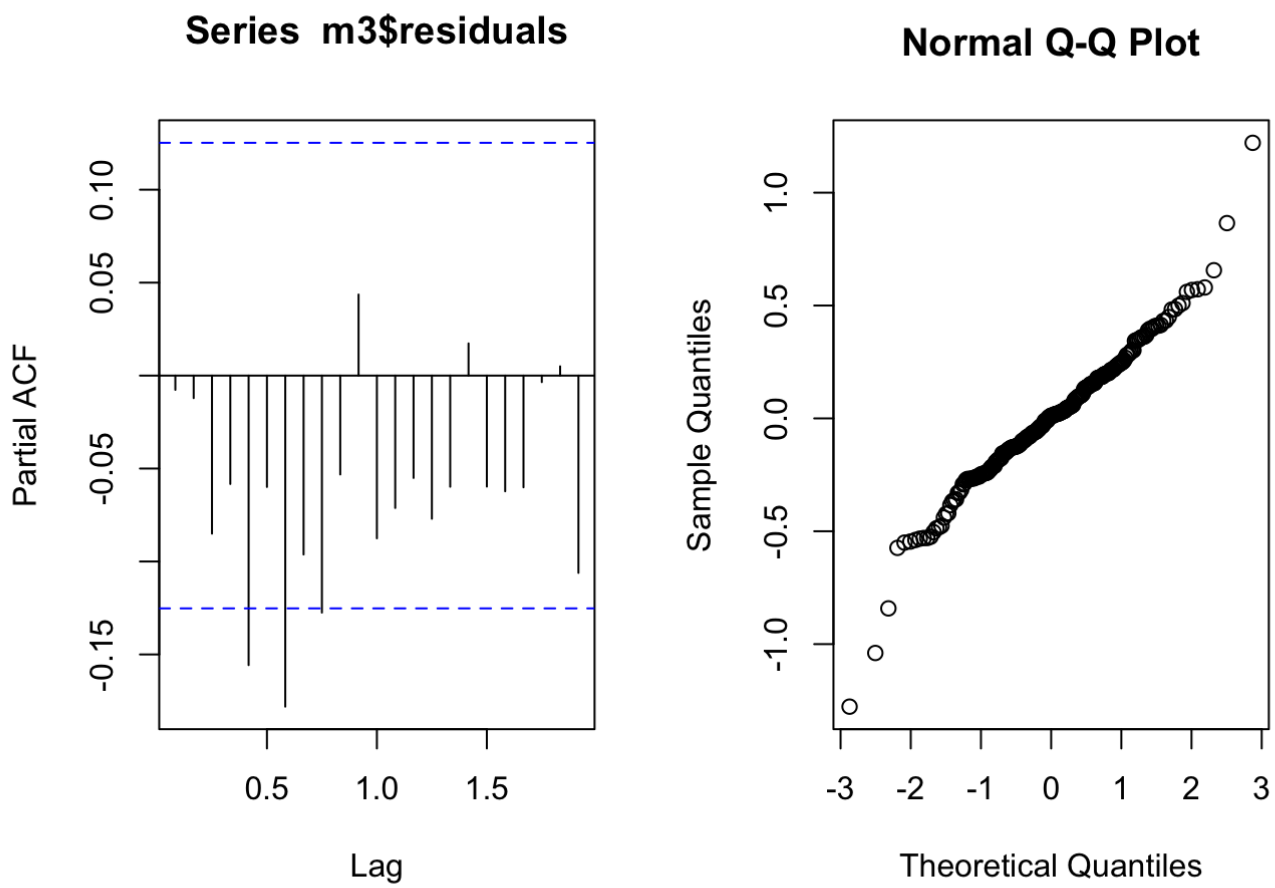
```
##            ar1      ar2    sar1     sma1
##        -0.0932  -0.1112  0.9993  -0.7537
## s.e.    0.0665   0.0651  0.0004   0.0509
##
## sigma^2 estimated as 0.0844:  log likelihood = -72.6,  aic = 153.19
```

We will now run diagnostics on the model using the code:

```
tsdiag(m3)
```



```
par(mfrow=c(1,2) , pty='m')
pacf(m3$residuals)
qqnorm(m3$residuals)
```




From the standardized residuals plot, it seems that there is no outlier since all residual values lie in between -4 and 4. However, the residuals are more spread than in other models. Also, there is no pattern and it is centered around 0. In ACF plot there is one significant value of serial correlation at lag 7. Also, the p-values for Ljung-Box statistics are not greater than significance level (=0.05) for lags greater than 7. So there is evidence that the SARIMA(2,1,0) (1,0,1)s=12 may not be suitable for this time series data.

The PACF of residuals plot shows a significant value at lag 5, 7, 9, this suggests the model is not a good fit. The Q-Q plot is not a straight line, this also suggests the model is not a good fit.

Using the model we will predict the next 10 data points, to do this we use the code:

```
m3.fore = predict(m3, n.ahead=10)
m3.fore
```

```
## $pred
##            Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                          8.343800  5.491738  3.614733
## 22 11.538733 12.381506 12.668638
##            Sep       Oct       Nov       Dec
## 21  3.269495  5.106431  8.091937 10.109686
## 22
##
## $se
##            Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         0.2905214 0.3921810 0.4565122
## 22 0.7063681 0.7464230 0.7844308
##            Sep       Oct       Nov       Dec
## 21 0.5154985 0.5695697 0.6185825 0.6639058
## 22
```

##2.4 Comparison

We will now compare the three sarima models using the following code:

```
AIC(m1)
```

```
## [1] 129.8234
```

```
BIC(m1)
```

```
## [1] 150.8309
```

```
AIC(m2)
```

```
## [1] 129.8524
```

```
BIC(m2)
```

```
## [1] 150.8599
```

```
AIC(m3)
```

```
## [1] 155.1927
```

```
BIC(m3)
```

```
## [1] 172.6786
```

```
(MAPE1=mean(abs((ice2[1:10]-m1.fore$pred)/ice2[1:10])))
```

```
## [1] 0.04396618
```

```
(MAPE2=mean(abs((ice2[1:10]-m2.fore$pred)/ice2[1:10])))
```

```
## [1] 0.04396618
```

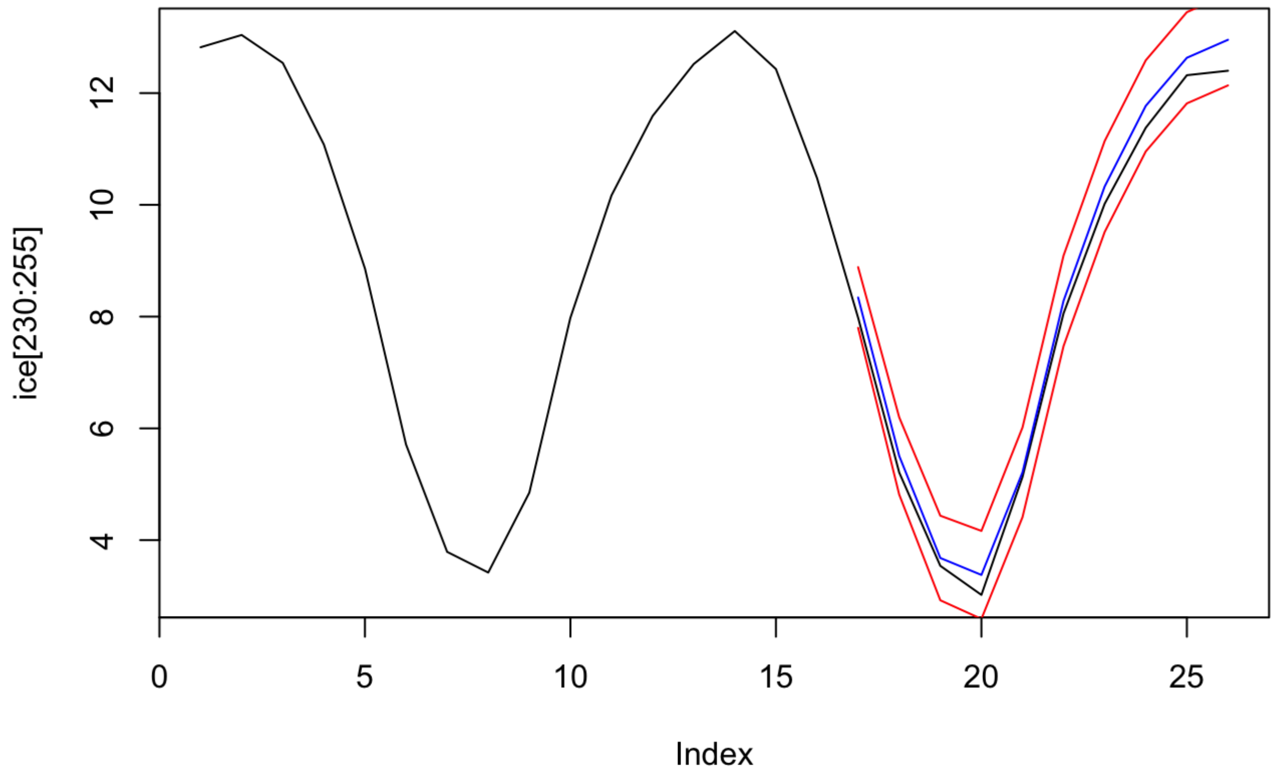
```
(MAPE3=mean(abs((ice2[1:10]-m3.fore$pred)/ice2[1:10])))
```

```
## [1] 0.02615032
```

All of the models have the same number of parameters. Model 3 has the value closest to zero for testing MAPE, however as we saw in the diagnostics above it is not a suitable model so we will not be using it. The next best MAPE is Model 1, this model also has the lowest AIC and BIC so this model should be use

The plot below shows the last 26 time points of the ice times eires plotted alongside the Model 1 forecast, the blue line shows the mean and the red line shows a 95% confidence interval. In order to generate this, the following code was used:

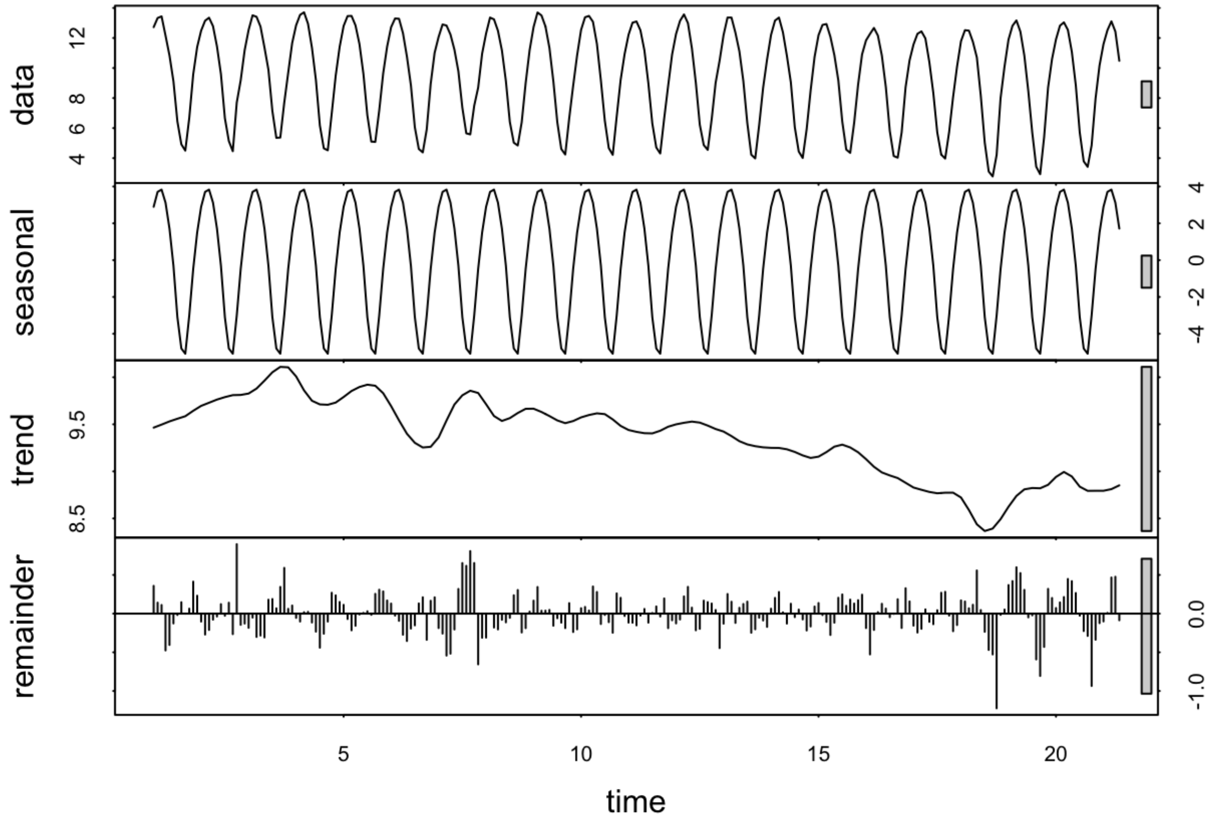
```
plot(ice[230:255] , type="l")
lines(17:26, m1.fore$pred , col="blue")
lines(17:26 , m1.fore$pred-(1.96*m1.fore$se) , col="red")
lines(17:26 , m1.fore$pred+(1.96*m1.fore$se) , col="red")
```




# 3. Decomposition

We will now analyse the LOWESS decomposition plot, to generate this plot we use the code:

```
lowess=stl(ice1,"periodic")
plot(lowess)
```




The LOWESS decomposition plot shows a decreasing trend, however if we look at the scale on the axis we see this is only minor. We see the seasonal effect is large. The remainder has no pattern and is centered around 0, it seems that there are no outliers since all values lie in between -1 and 1.

# 3.1 Model 4

First we will Forecast the next 10 values using the LOWESS arima method, to do this we use the code:

```
m4=forecast(lowess,method="arima", h=10)
m4$mean
```

```
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         8.449954  5.665421  3.981186
## 22 11.699118 12.510971 12.635380
##           Sep       Oct       Nov       Dec
## 21  3.705154  5.786950  8.322097 10.300068
## 22
```

## 3.2 Model 5

We will now forecast the next 10 values using the LOWESS ets method, to do this we use the code:

```
m5=forecast(lowess,method="ets", h=10)
m5$mean
```

```
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         8.544542  5.741192  4.042649
## 22 11.739319 12.550398 12.674334
##           Sep       Oct       Nov       Dec
## 21  3.757607  5.833870  8.365632 10.341533
## 22
```

## 3.3 Model 6

We will now forecast the next 10 values using the LOWESS naive method, to do this we use the code:

```
m6=forecast(lowess,method="naive", h=10)
m6$mean
```

```
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         8.467684  5.664333  3.965791
## 22 11.662460 12.473539 12.597475
##           Sep       Oct       Nov       Dec
## 21  3.680748  5.757011  8.288773 10.264675
## 22
```

## 3.4 Model 7

We will now forecast the next 10 values using the LOWESS rwdrift method, to do this we use the code:

```
m7=forecast(lowess,method="rwdrift", h=10)
m7$mean
```

```
##           Jan       Feb       Mar Apr May       Jun       Jul       Aug
## 21                                         8.463349  5.655664  3.952788
## 22 11.627786 12.434531 12.554133
##           Sep       Oct       Nov       Dec
## 21  3.663412  5.735340  8.262768 10.234335
## 22
```

## 3.5 Comparison

In order to compare the four LOWESS models we will find the testing MAPE values, to do this we use the code:

```
(MAPE4=mean(abs((ice2-m4$mean[1:10])/ice2)))
```

```
## [1] 0.074886
```

```
(MAPE5=mean(abs((ice2-m5$mean[1:10])/ice2)))
```

```
## [1] 0.08385476
```

```
(MAPE6=mean(abs((ice2-m6$mean[1:10])/ice2)))
```

```
## [1] 0.07156234
```

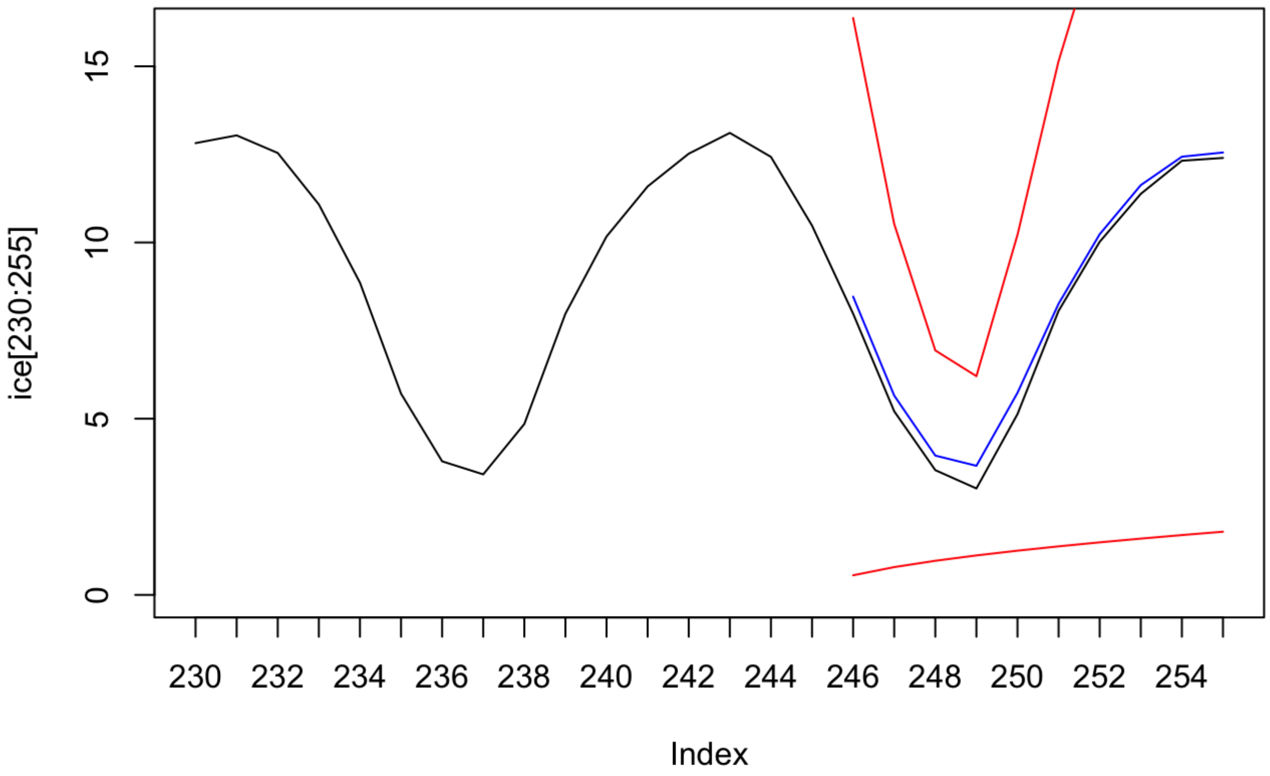
```
(MAPE7=mean(abs((ice2-m7$mean[1:10])/ice2)))
```

```
## [1] 0.06838158
```

Of the four LOWESS models, Model 7 (rw drift) has the MAPE closest to zero and therefore should be used.

The plot below shows the last 26 time points of the ice time series plotted alongside the Model 7 forecast, the blue line shows the mean and the red line shows a 95% confidence interval. In order to generate this, the following code was used:

```
plot(ice[230:255], xaxt = "n" , type="l" , ylim=c(0,16))
axis(1 , at=1:26, labels=230:255)
lines(17:26, m7$mean , col="blue")
lines(17:26 , m7$mean-(m7$lower[11:20]) , col="red")
lines(17:26 , m7$mean+(m7$lower[11:20]) , col="red")
```




# 4. Smoothing

## 4.1 Model 8

First we will use the simple exponential smoothing method to forecast 10 data points, to do this we use the code:

```
m8=ses(ice1, h=10)
summary(m8)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = ice1, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 12.7183
##
##   sigma:  1.7157
##
##      AIC     AICc      BIC
## 1616.325 1616.425 1626.829
##
## Error measures:
##                        ME     RMSE     MAE       MPE     MAPE     MASE
## Training set -0.009135916 1.708728 1.47263 -2.820665 18.91594 4.266106
##                   ACF1
## Training set 0.7884932
##
## Forecasts:
##        Point Forecast    Lo 80    Hi 80      Lo 95    Hi 95
## Jun 21        10.4802 8.281379 12.67901  7.1173960 13.84299
## Jul 21        10.4802 7.370755 13.58964  5.7247169 15.23567
## Aug 21        10.4802 6.671988 14.28840  4.6560447 16.30435
## Sep 21        10.4802 6.082893 14.87750  3.7551015 17.20529
## Oct 21        10.4802 5.563886 15.39650  2.9613495 17.99904
## Nov 21        10.4802 5.094667 15.86572  2.2437399 18.71665
## Dec 21        10.4802 4.663173 16.29722  1.5838279 19.37656
## Jan 22        10.4802 4.261548 16.69884  0.9695955 19.99079
```

```
## Feb 22        10.4802 3.884333 17.07606  0.3926949 20.56770
## Mar 22        10.4802 3.527554 17.43284 -0.1529519 21.11334
```

The simple exponential smoothing method recommends the smoothing parameter of alpha = 0.9999, this is very close to 1 so only the most recent values influence the forecasts.

# 4.2 Model 9

We will use Holt's method to forecast 10 data points, to do this we use the code:

```
m9=holt(ice1, h=10)
summary(m9)
```

```
## 
## Forecast method: Holt's method
## 
## Model Information:
## Holt's method
## 
## Call:
##  holt(y = ice1, h = 10) 
## 
##   Smoothing parameters:
##     alpha = 0.9903 
##     beta  = 0.9903 
## 
##   Initial states:
##     l = 12.9857 
##     b = 0.7485 
## 
##   sigma:  1.1244
## 
##      AIC     AICc      BIC 
## 1411.246 1411.497 1428.752 
## 
```

```
## Error measures:
##                       ME     RMSE       MAE      MPE     MAPE     MASE
## Training set -0.01112146 1.115226 0.9394781 3.996374 13.24397 2.721601
##                   ACF1
## Training set 0.4714317
##
## Forecasts:
##        Point Forecast       Lo 80     Hi 80      Lo 95    Hi 95
## Jun 21      8.5427331   7.1017018  9.983764   6.338866 10.74660
## Jul 21      6.5929232   3.3956926  9.790154   1.703181 11.48267
## Aug 21      4.6431134  -0.7001195  9.986346  -3.528656 12.81488
## Sep 21      2.6933035  -5.1254553 10.512062  -9.264456 14.65106
## Oct 21      0.7434936  -9.8415758 11.328563 -15.444973 16.93196
## Nov 21     -1.2063162 -14.8208345 12.408202 -22.027925 19.61529
## Dec 21     -3.1561261 -20.0422177 13.729965 -28.981175 22.66892
## Jan 22     -5.1059360 -25.4890538 15.277182 -36.279225 26.06735
## Feb 22     -7.0557459 -31.1476983 17.036207 -43.901208 29.78972
## Mar 22     -9.0055557 -37.0067163 18.995605 -51.829635 33.81852
```

Holt's method recommends the smoothing parameter of alpha = 0.9903 and beta = 0.9903.

## 4.3 Model 10

We will use the Holt-Winters method to forecast 10 data points, to do this we use the code:

```
m10=hw(ice1,seasonal="additive",h=10)
summary(m10)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = ice1, h = 10, seasonal = "additive")
```

```
##
##   Smoothing parameters:
##     alpha = 0.8532
##     beta  = 1e-04
##     gamma = 0.1086
##
##   Initial states:
##     l = 9.8474
##     b = -0.0076
##     s = 1.5493 -0.5695 -2.8612 -5.1087 -4.8127 -3.1733
##            -0.3286 1.5905 3.1345 3.8418 3.8305 2.9076
##
##   sigma:  0.295
##
##      AIC     AICc      BIC
## 767.0122 769.7082 826.5336
##
## Error measures:
##                       ME      RMSE       MAE        MPE    MAPE      MASE
## Training set 0.004008657 0.2851654 0.2156623 -0.1649453 2.74747 0.6247585
##                   ACF1
## Training set 0.0145229
##
## Forecasts:
##        Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Jun 21       8.431817  8.053811  8.809822  7.853707  9.009926
## Jul 21       5.649952  5.153040  6.146865  4.889990  6.409915
## Aug 21       3.942557  3.350120  4.534995  3.036502  4.848612
## Sep 21       3.716834  3.042247  4.391421  2.685142  4.748526
## Oct 21       5.726371  4.978589  6.474154  4.582737  6.870006
## Nov 21       8.566454  7.752012  9.380895  7.320873  9.812035
## Dec 21      10.356250  9.480192 11.232308  9.016435 11.696066
## Jan 22      11.688513 10.754883 12.622143 10.260648 13.116378
## Feb 22      12.492397 11.504531 13.480264 10.981586 14.003209
## Mar 22      12.676977 11.637689 13.716264 11.087523 14.266430
```

Holt's method recommends the smoothing parameter of alpha = 0.8532, beta = 1e-04 and gamma = 0.1086

## 4.4 Comparison

Before conducting numerical analysis we see that Model 8 predicts the same value for all time points and Model 9 predicts negative values for some time points (which is clearly not possible when the unit is volume of sea ice). This suggests these models may not be a good fit.

In order to compare the three smoothing models we will find the testing MAPE, AIC and BIC to do this we use the code:

```
summary(m8)
```

```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
##  ses(y = ice1, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.9999
##
##   Initial states:
##     l = 12.7183
##
##   sigma:  1.7157
##
##      AIC     AICc      BIC
## 1616.325 1616.425 1626.829
##
## Error measures:
##                        ME     RMSE     MAE       MPE     MAPE     MASE
## Training set -0.009135916 1.708728 1.47263 -2.820665 18.91594 4.266106
##                   ACF1
## Training set 0.7884932
##
## Forecasts:
```

```
##         Point Forecast    Lo 80    Hi 80      Lo 95    Hi 95
## Jun 21        10.4802 8.281379 12.67901  7.1173960 13.84299
## Jul 21        10.4802 7.370755 13.58964  5.7247169 15.23567
## Aug 21        10.4802 6.671988 14.28840  4.6560447 16.30435
## Sep 21        10.4802 6.082893 14.87750  3.7551015 17.20529
## Oct 21        10.4802 5.563886 15.39650  2.9613495 17.99904
## Nov 21        10.4802 5.094667 15.86572  2.2437399 18.71665
## Dec 21        10.4802 4.663173 16.29722  1.5838279 19.37656
## Jan 22        10.4802 4.261548 16.69884  0.9695955 19.99079
## Feb 22        10.4802 3.884333 17.07606  0.3926949 20.56770
## Mar 22        10.4802 3.527554 17.43284 -0.1529519 21.11334
```

```
summary(m9)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
##  holt(y = ice1, h = 10)
##
##   Smoothing parameters:
##     alpha = 0.9903
##     beta  = 0.9903
##
##   Initial states:
##     l = 12.9857
##     b = 0.7485
##
##   sigma:  1.1244
##
##      AIC     AICc      BIC
## 1411.246 1411.497 1428.752
##
```

```
## Error measures:
##                       ME     RMSE       MAE      MPE     MAPE     MASE
## Training set -0.01112146 1.115226 0.9394781 3.996374 13.24397 2.721601
##                   ACF1
## Training set 0.4714317
##
## Forecasts:
##        Point Forecast       Lo 80     Hi 80      Lo 95    Hi 95
## Jun 21      8.5427331   7.1017018  9.983764   6.338866 10.74660
## Jul 21      6.5929232   3.3956926  9.790154   1.703181 11.48267
## Aug 21      4.6431134  -0.7001195  9.986346  -3.528656 12.81488
## Sep 21      2.6933035  -5.1254553 10.512062  -9.264456 14.65106
## Oct 21      0.7434936  -9.8415758 11.328563 -15.444973 16.93196
## Nov 21     -1.2063162 -14.8208345 12.408202 -22.027925 19.61529
## Dec 21     -3.1561261 -20.0422177 13.729965 -28.981175 22.66892
## Jan 22     -5.1059360 -25.4890538 15.277182 -36.279225 26.06735
## Feb 22     -7.0557459 -31.1476983 17.036207 -43.901208 29.78972
## Mar 22     -9.0055557 -37.0067163 18.995605 -51.829635 33.81852
```

```
summary(m10)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = ice1, h = 10, seasonal = "additive")
##
##   Smoothing parameters:
##     alpha = 0.8532
##     beta  = 1e-04
##     gamma = 0.1086
##
##   Initial states:
```

```
##     l = 9.8474
##     b = -0.0076
##     s = 1.5493 -0.5695 -2.8612 -5.1087 -4.8127 -3.1733
##            -0.3286 1.5905 3.1345 3.8418 3.8305 2.9076
##
##   sigma:  0.295
##
##      AIC     AICc      BIC
## 767.0122 769.7082 826.5336
##
## Error measures:
##                       ME      RMSE       MAE        MPE    MAPE      MASE
## Training set 0.004008657 0.2851654 0.2156623 -0.1649453 2.74747 0.6247585
##                   ACF1
## Training set 0.0145229
##
## Forecasts:
##        Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## Jun 21       8.431817  8.053811  8.809822  7.853707  9.009926
## Jul 21       5.649952  5.153040  6.146865  4.889990  6.409915
## Aug 21       3.942557  3.350120  4.534995  3.036502  4.848612
## Sep 21       3.716834  3.042247  4.391421  2.685142  4.748526
## Oct 21       5.726371  4.978589  6.474154  4.582737  6.870006
## Nov 21       8.566454  7.752012  9.380895  7.320873  9.812035
## Dec 21      10.356250  9.480192 11.232308  9.016435 11.696066
## Jan 22      11.688513 10.754883 12.622143 10.260648 13.116378
## Feb 22      12.492397 11.504531 13.480264 10.981586 14.003209
## Mar 22      12.676977 11.637689 13.716264 11.087523 14.266430
```

```
(MAPE8=mean(abs((ice2-m8$mean[1:10])/ice2)))
```

```
## [1] 0.7527982
```

```
(MAPE9=mean(abs((ice2-m9$mean[1:10])/ice2)))
```

```
## [1] 0.8823102
```

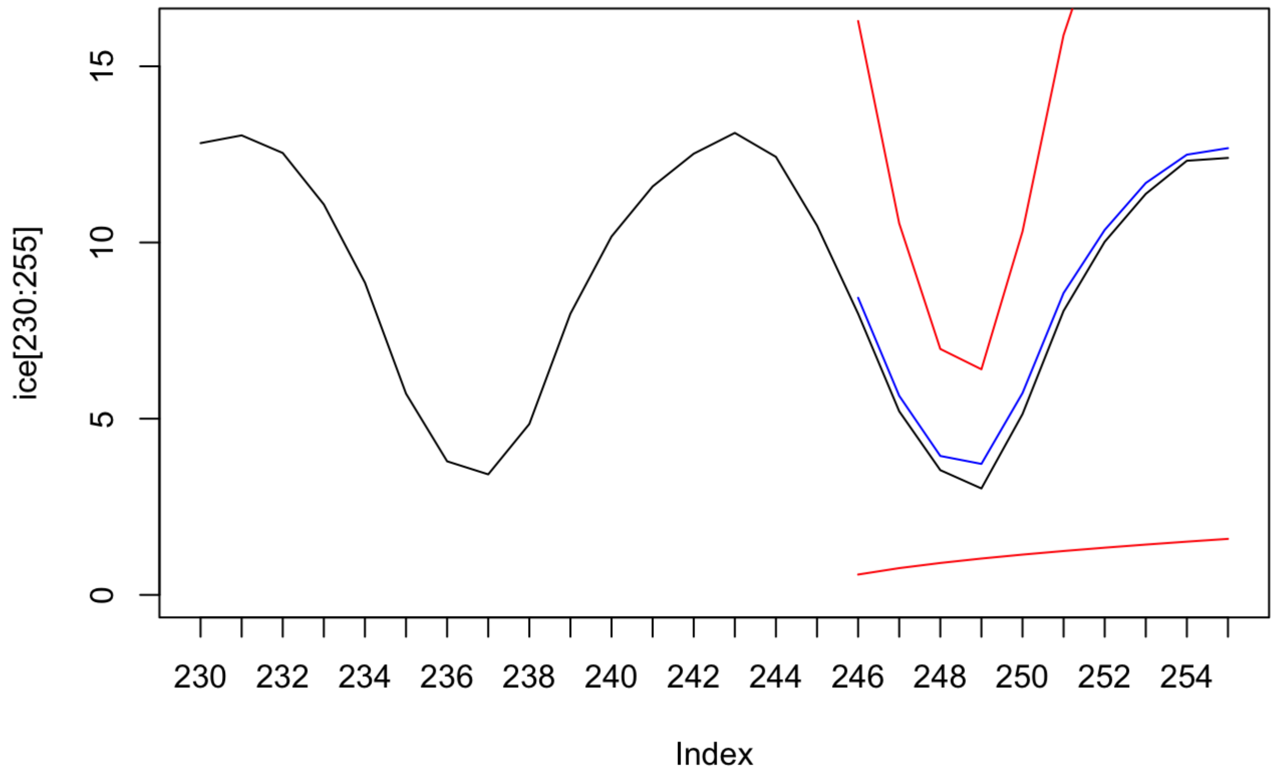
```
(MAPE10=mean(abs((ice2-m10$mean[1:10])/ice2)))
```

```
## [1] 0.07616042
```

Model 10 (Holt-Winters) has the MAPE closest to zero as well as the lowest AIC and BIC, therefore this model should be used. This is supported by our original visual inspection of the forecast predictions.

The plot below shows the last 26 time points of the ice time series plotted alongside the Model 10 forecast, the blue line shows the mean and the red line shows a 95% confidence interval. In order to generate this, the following code was used:

```
plot(ice[230:255], xaxt = "n" , type="l" , ylim=c(0,16))
axis(1 , at=1:26, labels=230:255)
lines(17:26, m10$mean , col="blue")
lines(17:26 , m10$mean-(m10$lower[11:20]) , col="red")
lines(17:26 , m10$mean+(m10$lower[11:20]) , col="red")
```




## 5. Conclusion

To conclude, by running two augmented Dickey–Fuller tests we saw the time series was not stationary or explosive and therefore differencing was not necessary. Next, we ran a Box-Cox test which showed that a variance stabilizing transformation was also not necessary. Analysing the periodogram showed that the time series has a frequency of 12.

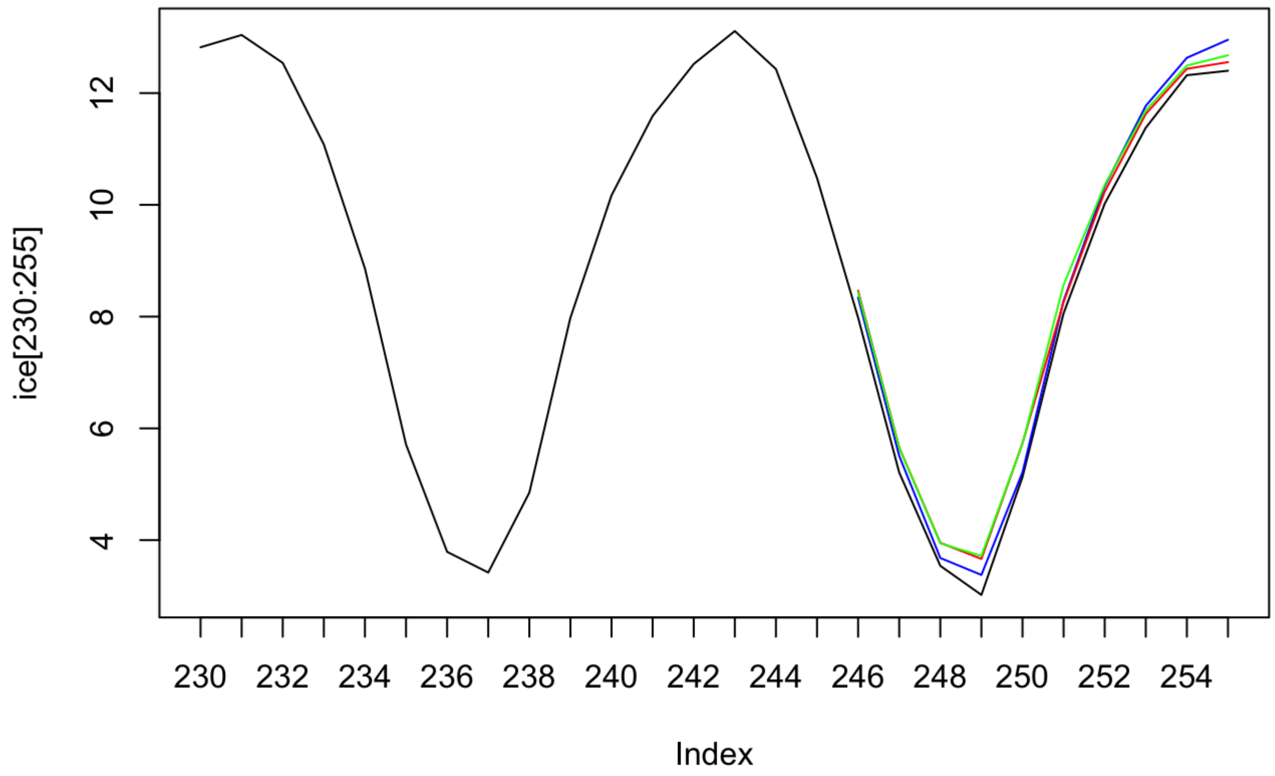
By analysing the ACF and PACF plots, we then fitted 3 SARIMA models. The best of these was Model 1, SARIMA (2,0,0)(1,0,1)s=12 .

Using LOWESS decomposition we saw the data had a slight decreasing trend as well as a strong seasonal effect with a frequency of 12. Next we fitted 4 models: LOWESS arima, LOWESS ets, LOWESS naive and LOWESS rwdrift. The best of these was Model 7, the LOWESS rwdrift.

Finally we used fit 4 smoothing methods to the data, these were: Simple Exponential Smoothing, Holt's Method and the Holt-Winters Method. The best of these was Model 10, the Holt-Winters method.

The plot below shows the last 26 time points of the ice time series plotted alongside the best model from each section above. Model 1 forecast (blue), Model 7 forecast (red), Model 10 forecast (green). In order to generate this, the following code was used:

```
plot(ice[230:255], xaxt = "n" , type="l")
axis(1 , at=1:26, labels=230:255)
lines(17:26 , m1.fore$pred , col="blue")
lines(17:26 , m7$mean , col="red")
lines(17:26 , m10$mean , col="green")
```




Using visual inspection all the models appear to fit very well so we will compare the testing MAPE values (the code for generating these values is shown earlier in the report).

Model 1: 0.04396618 Model 2: 0.06838158 Model 3: 0.07616042

Here we see Model 1, SARIMA (2,0,0)(1,0,1)s=12 , has the testing MAPE closest to zero and therefore is the best fit for the data. The final model for the data is given by:

(1-0.7889B+0.0568B2)(1-0.9987B12)XT=(1-0.6993B12)+9.2954

We will now use our final model to predict the volume of sea ice (in 1000km3) in the arctic for the six month period after what is given in the dataset. This is done using the code:

```
m1.fore = predict(m1, n.ahead=16)
m1.fore$pred[11:16]
```

```
## [1] 12.309979 10.744333  8.559546  5.666626  3.798187  3.463502
```