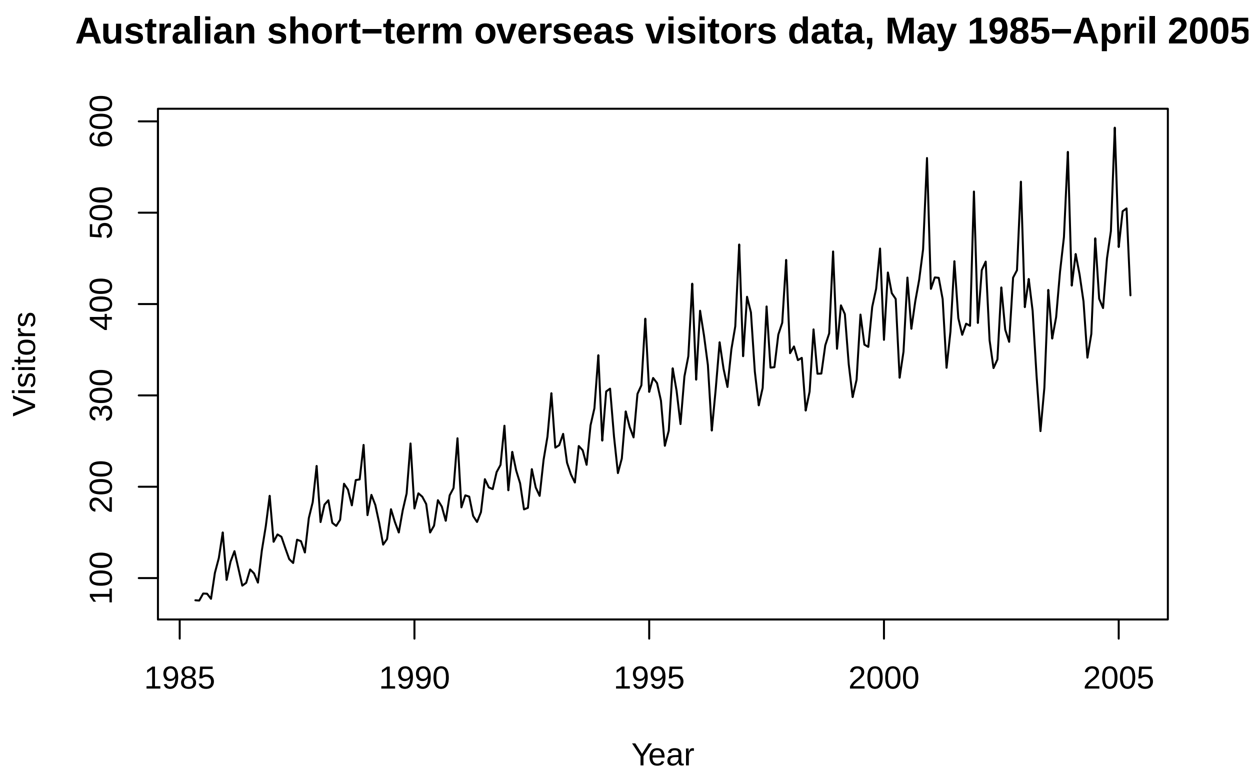
Assignment 6

Matthew Dunne

July 28, 2018

Question 1

Load and plot the visitors dataset and plot the dataset. Describe the main dataset characteristics.



This data has a pronounced upward trend and seasonality. There is some noise, meaning the pattern of moving from month to month looks the same generally but is not always exactly uniform. The amplitude of the seasonal pattern varies (is proportional to) with the average level within the season. In other words the difference between the highest point and the lowest point within a season gets bigger as time goes on.

Question 2

In this exercise you will apply the time-series cross validation method to train and test various models. Use the following values when training and testing the models:

- *Set the minimum number of samples required to train the model to 160 (i.e., this is the minimum number of samples in the sliding window and the initial number of samples in the expanding window method.)*
- *Set the number the forecast horizon, h, to 1 year (i.e., 12 months.)*

- *Recall that the period, p, is 12 months*
- *Use a single observation incrementation in each iteration (i.e., shift the training set forward by 1 observations.)*
- *Note: You are expected to have 80 iterations of cross validation*

```
k <- 160 # minimum data length for fitting a model
n <- length(data) # Number of data points
p <- 12 ### Period
H <- 12 # Forecast Horiz
st <- tsp(data)[1]+(k-2)/p #  gives the start time in time units,
#make matrices of NAs, # of rows=n-k, # of columns=H to store the errors for each model
sarima_expanding_err <- matrix(NA,n-k,H)
sarima_sliding_err <- matrix(NA,n-k,H)
ets_expanding_err <- matrix(NA,n-k,H)
ets_sliding_err <- matrix(NA,n-k,H)
#make a data frame to store the AIC values
aic<-data.frame()
```

For each iteration, apply the following 4 forecasts:

1) Use the Arima() function to estimate a sARIMA([1,0,1][0,1,2]12) with drift model for a
a) expanding window
b) sliding window

```
#set empty vectors into which your AICc and iterations will go
aic_arima_exp<-vector()
aic_arima_sld<-vector()
iteration<-vector()
#loop through all observations after k
for(i in 1:(n-k))
{
  ### One Month rolling forecasting
  # Expanding Window
  train_1 <- window(data, end=st + i/p)  ## Window Length: k+i

  # Sliding Window - keep the training window of fixed length.
  # The training set always consists of k observations.
  train_2 <- window(data, start=st+(i-k+1)/p, end=st+i/p) ## Window Length: k

  test <- window(data, start=st + (i+1)/p, end=st + (i+H)/p) ## Window Length: H


  #for the expanding window
  fit_arima_exp <- Arima(train_1, order=c(1,0,1), seasonal=list(order=c(0,1,2), period=p),
                include.drift=TRUE, lambda=0, method="ML")
  #extract the AICc from the model and append to AICc vector, extract iteration too
  aic_arima_exp<-c(aic_arima_exp, fit_arima_exp$aicc)
  iteration<-c(iteration, i)
  fcast_arima_exp <- forecast(fit_arima_exp, h=H)

  #for the sliding window
  fit_arima_sld <- Arima(train_2, order=c(1,0,1), seasonal=list(order=c(0,1,2), period=p),
                include.drift=TRUE, lambda=0, method="ML")
  aic_arima_sld<-c(aic_arima_sld, fit_arima_sld$aicc)
  fcast_arima_sld <- forecast(fit_arima_sld, h=H)
```

```
  sarima_expanding_err[i,1:length(test)] <- (fcast_arima_exp[['mean']]-test)
  sarima_sliding_err[i,1:length(test)] <- (fcast_arima_sld[['mean']]-test)
}
arima_aics<-data.frame(Iteration=iteration, ARIMA_EXP_AICc=aic_arima_exp, ARIMA_SLD_AICc=aic_arima_sld)
```

2) Use the Exponential Smoothing Function ets() to estimate MAM (Multiplicative Error, Additive trend, multiplicative Season) model for a
a) expanding window
b) sliding window

```
#set empty vectors into which your AICc and iterations will go
aic_ets_exp<-vector()
aic_ets_sld<-vector()
iteration<-vector()
#loop through all observations after k
for(i in 1:(n-k))
{

  ### One Month rolling forecasting
  # Expanding Window
  train_1 <- window(data, end=st + i/p)  ## Window Length: k+i

  # Sliding Window - keep the training window of fixed length.
  # The training set always consists of k observations.
  train_2 <- window(data, start=st+(i-k+1)/p, end=st+i/p) ## Window Length: k

  test <- window(data, start=st + (i+1)/p, end=st + (i+H)/p) ## Window Length: H

  #for the expanding window
  fit_ets_exp <- ets(train_1, model = "MAM")
  fcast_ets_exp <- forecast(fit_ets_exp, h=H)
  #extract the AICc from the model and append to AICc vector, extract iteration too
  aic_ets_exp<-c(aic_ets_exp, fit_ets_exp$aicc)
  iteration<-c(iteration, i)

  #for the sliding window
  fit_ets_sld <- ets(train_2, model = "MAM")
  fcast_ets_sld <- forecast(fit_ets_sld, h=H)
  #extract the AICc from the model and append
  aic_ets_sld<-c(aic_ets_sld, fit_ets_sld$aicc)


  ets_expanding_err[i,1:length(test)] <- (fcast_ets_exp[['mean']]-test)
  ets_sliding_err[i,1:length(test)] <- (fcast_ets_sld[['mean']]-test)

}
#construct a data frame with the iteration and AICc for expanding and sliding model
ets_aics<-data.frame(Iteration=iteration, ETS_EXP_AICc=aic_ets_exp, ETS_SLD_AICc=aic_ets_sld)
```

Let us check that we have the proper number of iterations. We do this by looking at the number of rows in the data frame in which we stored the AICc values.

First the sARIMA AICc:

```r
nrow(arima_aics)
```

```
## [1] 80
```

And then the ETS AICc:

```r
nrow(ets_aics)
```

```
## [1] 80
```

So we have the proper number of iterations under each model.

For each test window record the:

1) one-year forecast horizon error

Let us look at the matrices that in which we recorded the errors of our model. First the errors of the sARIMA models:

```r
head(sarima_expanding_err, 3)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] -10.23337  8.983343 19.779302 17.017919 10.300310  9.524935  6.583401
## [2,]  17.62469 28.637789 27.377710 17.956547 17.479332 14.064018 36.167413
## [3,]  14.67784 10.730639  5.772418  4.060564  1.235727 24.908501  7.317794
##          [,8]     [,9]    [,10]     [,11]     [,12]
## [1,] 29.21825 11.03950 18.49543 27.489052 14.980839
## [2,] 16.76770 24.40049 34.49941 21.064199 13.071267
## [3,] 14.64685 22.91606 10.53710  3.446214  7.468983
```

```r
head(sarima_sliding_err, 3)
```

```
##           [,1]      [,2]      [,3]      [,4]       [,5]      [,6]
## [1,] -10.23337  8.983343 19.779302 17.017919 10.3003095  9.524935
## [2,]  17.45927 28.344736 26.844402 17.421321 16.6498175 13.279627
## [3,]  14.19490  9.768555  4.511519  2.878047 -0.1833509 22.742948
##           [,7]     [,8]     [,9]     [,10]      [,11]     [,12]
## [1,]  6.583401 29.21825 11.03950 18.495428 27.4890519 14.980839
## [2,] 35.450702 16.18139 23.62989 33.437932 20.3149280 12.240049
## [3,]  5.348400 12.35509 19.90543  7.966914  0.5434467  4.382986
```

And now the ETS models:

```r
head(ets_expanding_err, 3)
```

```
##           [,1]      [,2]      [,3]       [,4]       [,5]       [,6]
## [1,] -18.98488  6.501018  16.99964   5.899986 -12.634704 -20.9605011
## [2,]  21.91862 34.364964  28.48120   4.480087  -2.466945  -0.7789178
## [3,]  18.76465  9.252670 -12.09016 -21.016279 -20.492366  -5.8049671
##           [,7]       [,8]       [,9]     [,10]      [,11]     [,12]
## [1,] -18.82746  -3.317051 -11.627987 -19.21300 -30.795467 -22.43631
## [2,]  13.10755   2.661901  -5.400114 -13.35671  -9.489569 -29.58547
## [3,] -12.44131 -21.084880 -32.087667 -24.63836 -44.659188 -32.79922
```

```r
head(ets_sliding_err, 3)
```

```
##           [,1]      [,2]      [,3]       [,4]       [,5]       [,6]
## [1,] -18.98488  6.501018  16.99964   5.899986 -12.634704 -20.960501
## [2,]  22.93116 34.862996  30.60545   3.240601  -2.423216  -2.403351
## [3,]  20.46775  7.757264 -11.32847 -17.423371 -17.134366  -2.341434
```

```
##            [,7]       [,8]      [,9]     [,10]     [,11]     [,12]
## [1,] -18.82746  -3.317051 -11.62799 -19.21300 -30.79547 -22.43631
## [2,]  11.80197   2.494217  -5.88776 -14.17098  -8.14902 -29.66616
## [3,] -11.52929 -18.879159 -30.38943 -21.87833 -42.85911 -30.42088
```

We see how the errors are different for each model, though the differences seem to be slight for any given observation.

2) estimated model AICc value

And let us also look at the estimated AICc values for each model. First the sARIMA models:

```r
head(arima_aics)
```

```
##   Iteration ARIMA_EXP_AICc ARIMA_SLD_AICc
## 1         1      -425.9020      -425.9020
## 2         2      -429.5792      -425.7699
## 3         3      -432.8148      -426.1677
## 4         4      -436.3186      -426.9698
## 5         5      -440.3406      -428.1302
## 6         6      -444.3602      -433.3648
```

Then the ETS models:

```r
head(ets_aics)
```

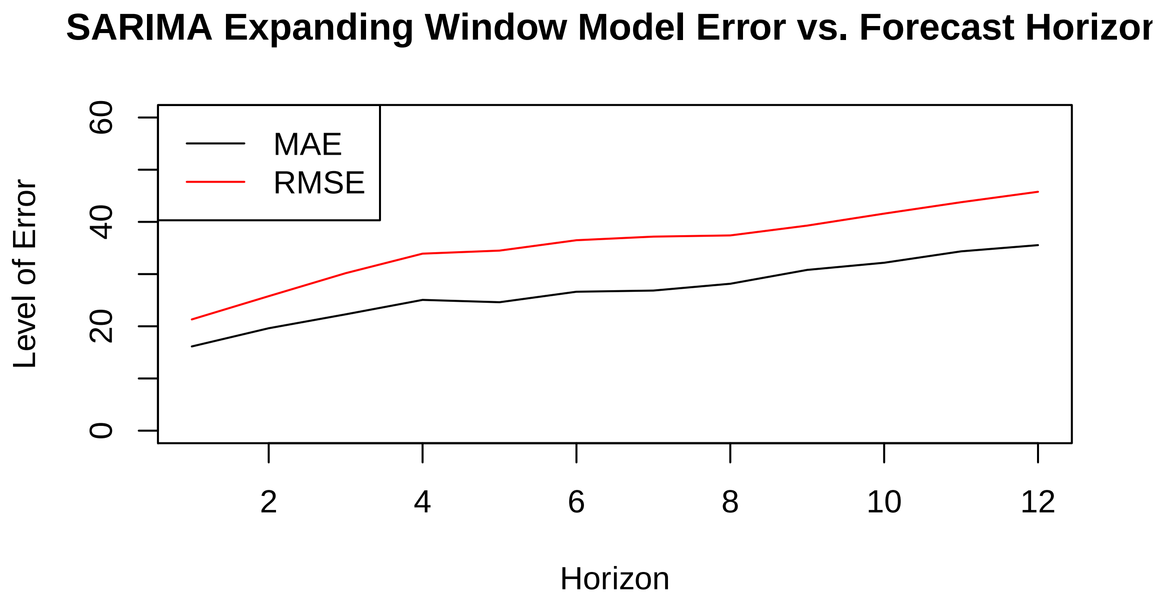
```
##   Iteration ETS_EXP_AICc ETS_SLD_AICc
## 1         1     1612.904     1612.904
## 2         2     1624.304     1611.692
## 3         3     1636.381     1615.613
## 4         4     1648.206     1616.936
## 5         5     1659.493     1620.110
## 6         6     1671.581     1614.868
```

For each of the four models above, calculate and plot the

1) Mean Absolute Forecast Error (MAE) vs forecast horizon
2) Root-square Forecast Error (RMSE) vs forecast horizon
3) AICc vs iteration number

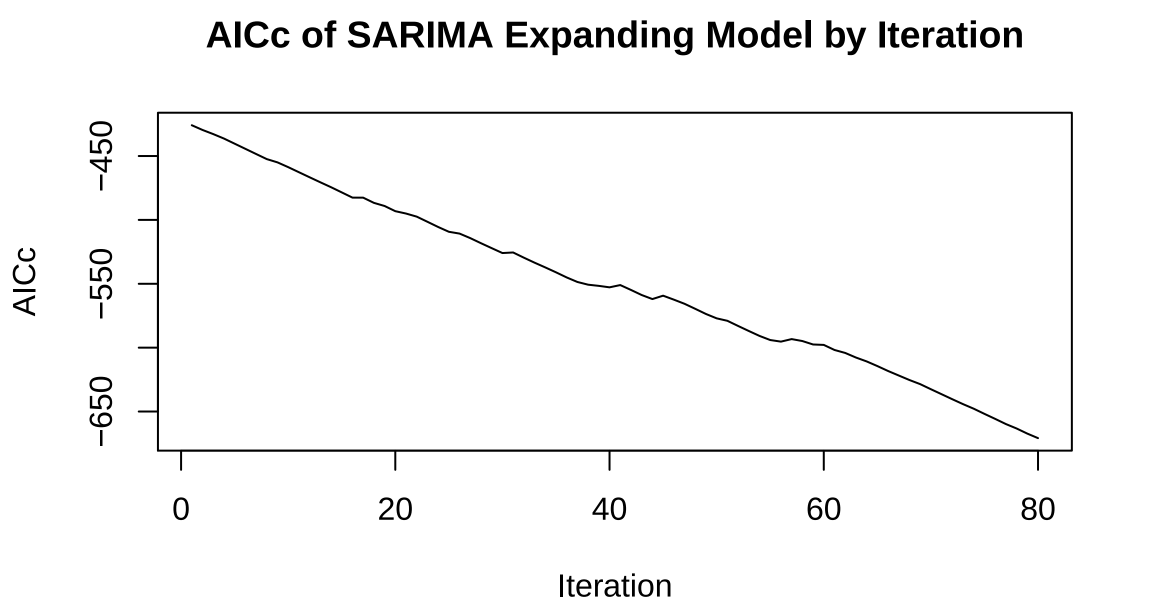
For the **sARIMA Expanding Window model** we see the following relationship of MAE and RMSE to the forecast horizon:

```r
#plot MAE
plot(1:12, colMeans(abs(sarima_expanding_err),na.rm=TRUE), type="l",col=1,xlab="Horizon", ylab="Level o
#add RMSE on top
lines(1:12, sqrt(colMeans(sarima_expanding_err^2,na.rm=TRUE)), type="l",col=2)
legend("topleft",legend=c("MAE","RMSE"),col=1:2,lty=1)
```



And we see the following regarding the AICc vs. iteration number:

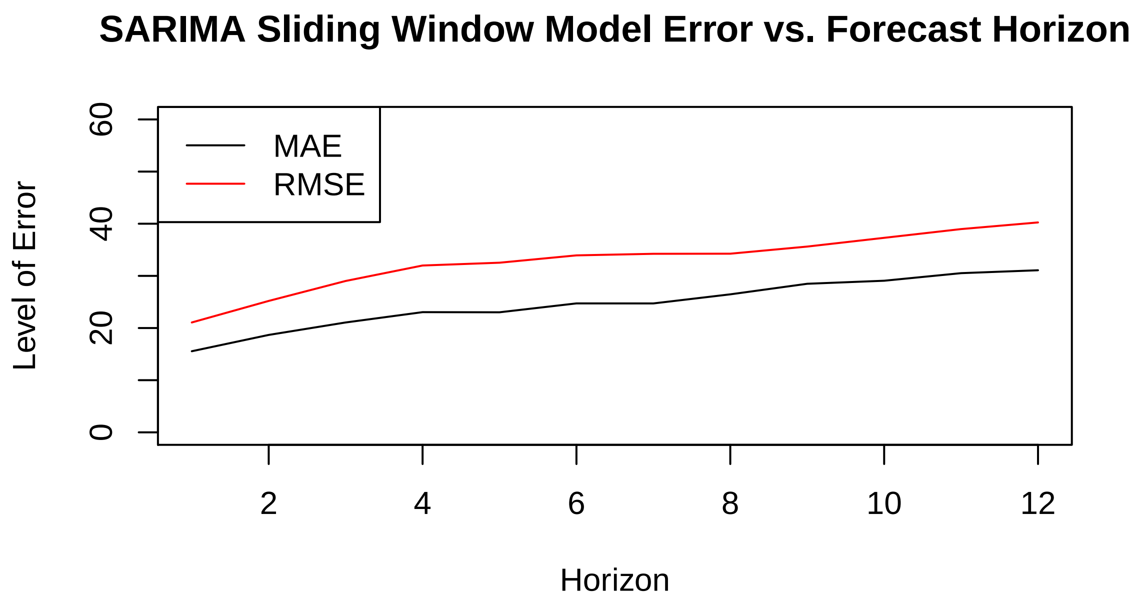
```
plot(arima_aics$Iteration, arima_aics$ARIMA_EXP_AICc, type = "l", xlab="Iteration", yla="AICc", main =
```



For the **sARIMA Sliding Window model** we see the following relationship of MAE and RMSE to the forecast horizon:

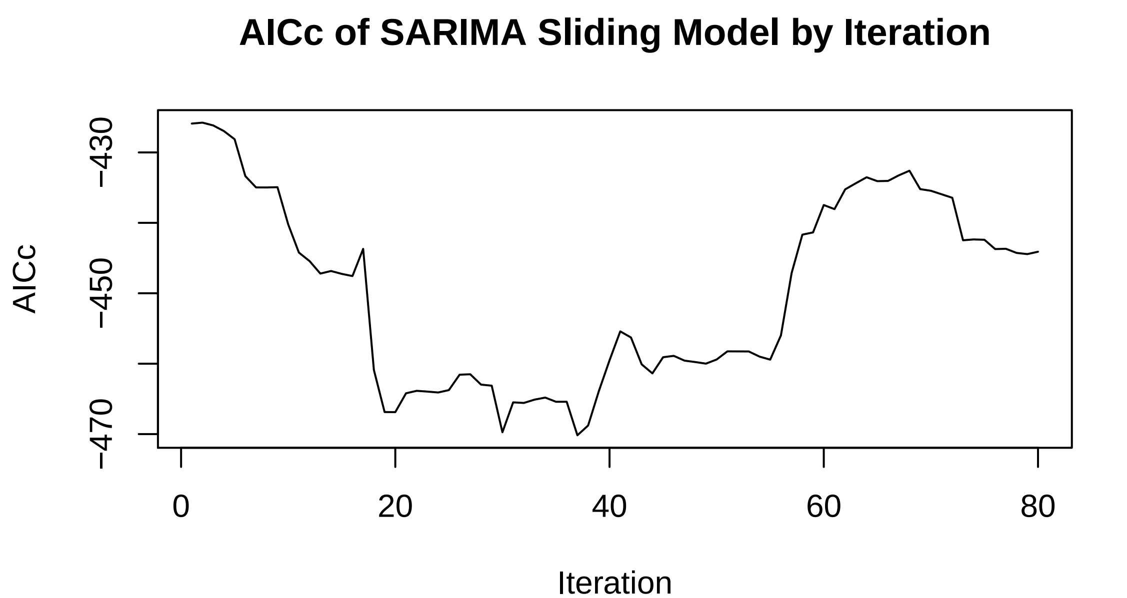
```
#plot MAE
plot(1:12, colMeans(abs(sarima_sliding_err),na.rm=TRUE), type="l",col=1,xlab="Horizon", ylab="Level of E
#add RMSE on top
lines(1:12, sqrt(colMeans(sarima_sliding_err^2,na.rm=TRUE)), type="l",col=2)
```

```r
legend("topleft",legend=c("MAE","RMSE"),col=1:2,lty=1)
```



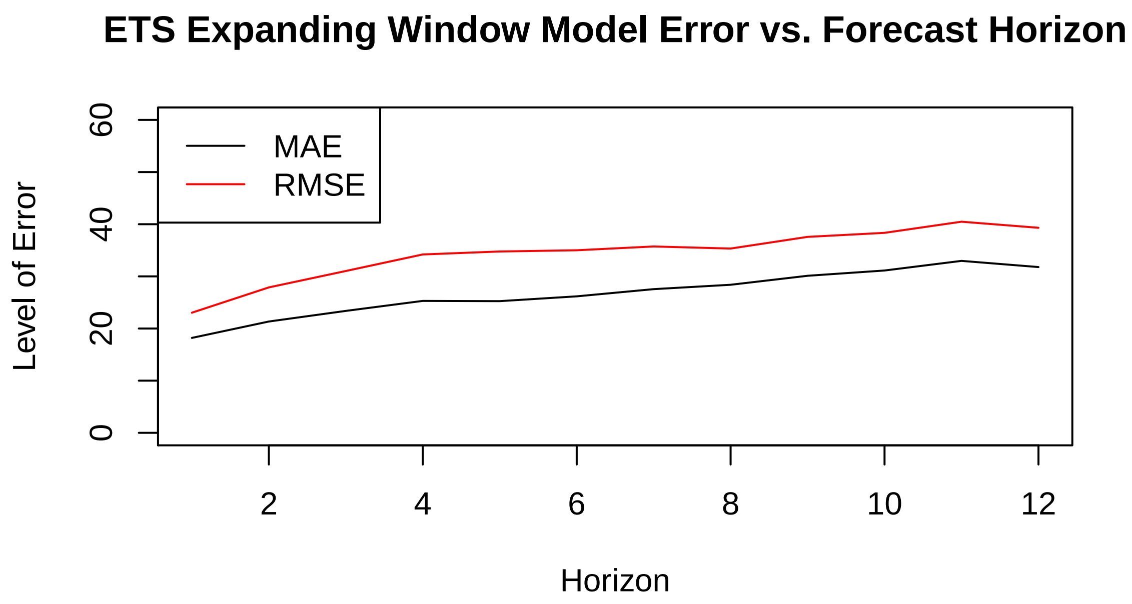
And we see the following regarding the AICc vs. iteration number:

```r
plot(arima_aics$Iteration, arima_aics$ARIMA_SLD_AICc, type = "l", xlab="Iteration", yla="AICc", main =
```



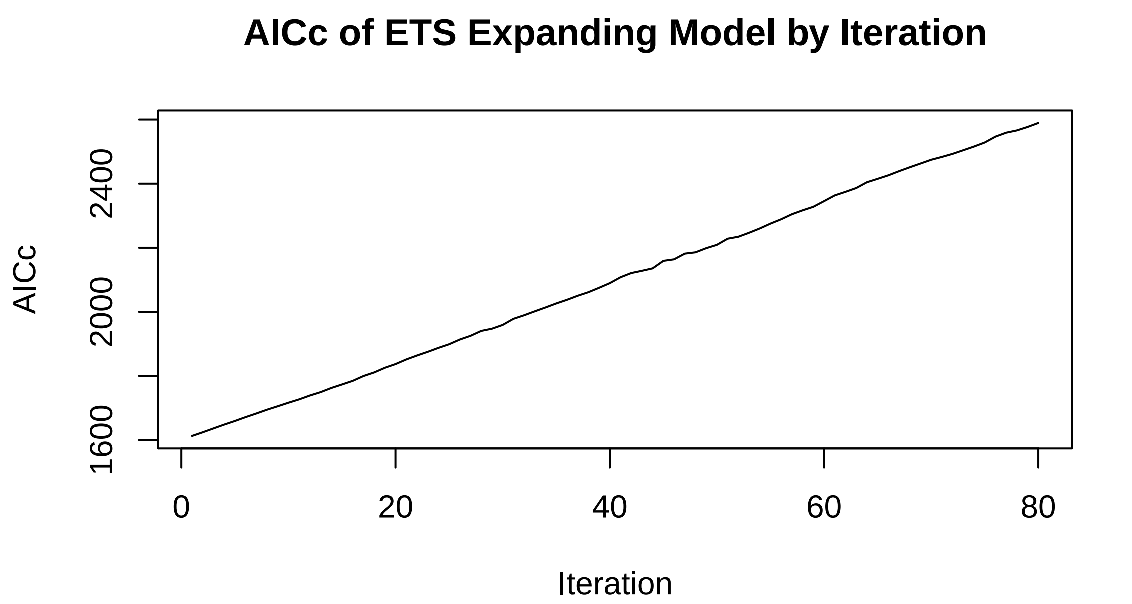
For the **ETS Expanding Window model** we see the following relationship of MAE and RMSE to the forecast horizon:

```
#plot MAE
plot(1:12, colMeans(abs(ets_expanding_err),na.rm=TRUE), type="l",col=1,xlab="Horizon", ylab="Level of E
#add RMSE on top
lines(1:12, sqrt(colMeans(ets_expanding_err^2,na.rm=TRUE)), type="l",col=2)
legend("topleft",legend=c("MAE","RMSE"),col=1:2,lty=1)
```



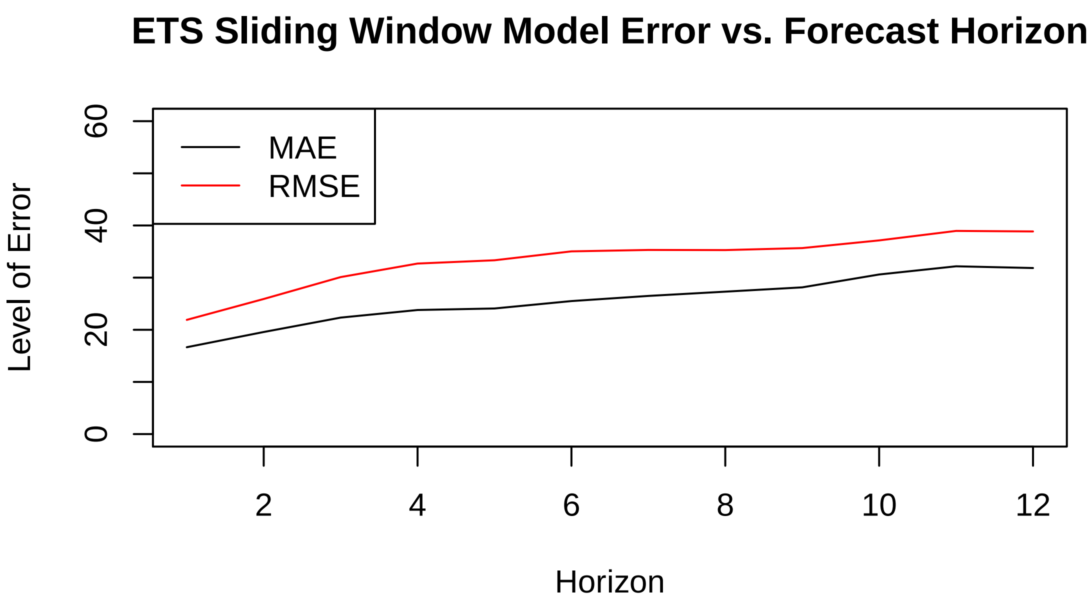
And we see the following regarding the AICc vs. iteration number:

```
plot(ets_aics$Iteration, ets_aics$ETS_EXP_AICc, type = "l", xlab="Iteration", yla="AICc", main = "AICc o
```



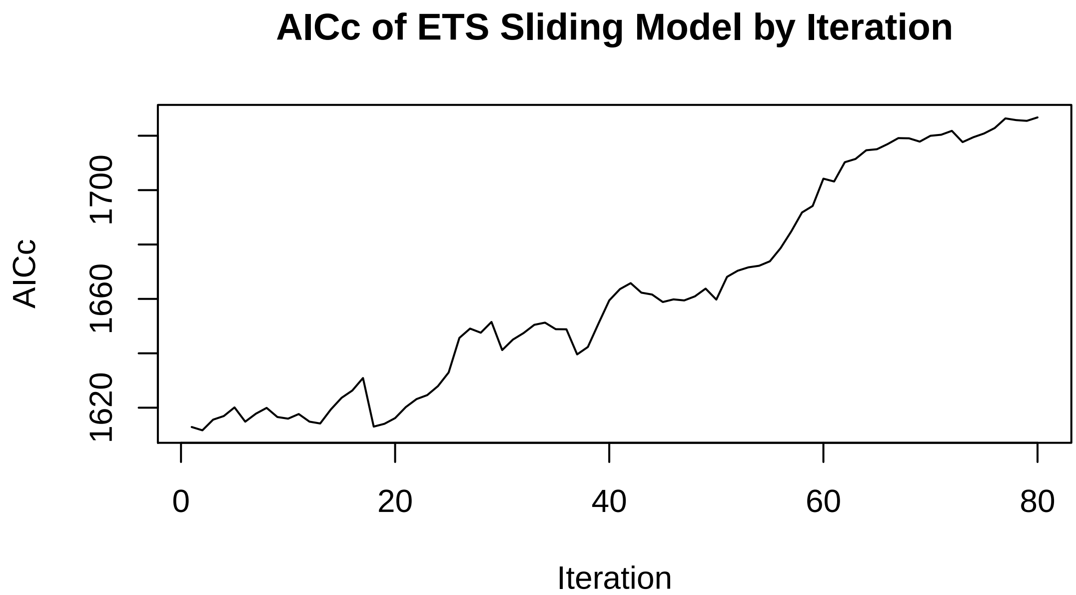
For the **ETS Sliding Window model** we see the following relationship of MAE and RMSE to the forecast horizon:

```
#plot MAE
plot(1:12, colMeans(abs(ets_sliding_err),na.rm=TRUE), type="l",col=1,xlab="Horizon", ylab="Level of Erro
#add RMSE on top
lines(1:12, sqrt(colMeans(ets_sliding_err^2,na.rm=TRUE)), type="l",col=2)
legend("topleft",legend=c("MAE","RMSE"),col=1:2,lty=1)
```



And we see the following regarding the AICc vs. iteration number:

```
plot(ets_aics$Iteration, ets_aics$ETS_SLD_AICc, type = "l", xlab="Iteration", yla="AICc", main = "AICc
```



Discuss your results.

The MAE and RMSE are largely the same between model types (sARIMA, ETS) and between window types (expanding, sliding).

Of the expanding window models, the AICc for sARIMA goes down by roughly four with every iteration. For the ETS is goes up by about 12.

Of the sliding window models, the AICc for the sARIMA model falls, almost in a stepwise fashion, as the number of iterations goes to 40. Then it rises again. For the ETS model there is a noticable, if irregular, trend upwards.

Question 3

What are the disadvantages of the above methods. What would be a better approach to estimate the models?

Hint: How were the sArima and exponential time series models determined in question 2?

The first thing I would change is to set k (minimum number of observations in expanding window and total observations in sliding window) to some multiple of 12, given that we have monthly data. By setting it at 160 we are using 13 years and 4 months as our training data for the sliding window, rather than full years, e.g. August to July.

Also, given the nature of the data we may want to increment by 12 rather than 1. That way you are using some number of full years to forecast 1 to 12 months ahead.