```
---
title: "Unit 11 asynch work"
author: "Todd Garner"
date: "2023-03-13"
output: html_document
editor_options:
  chunk_output_type: console
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(dev = c('pdf', 'png'),
        fig.align = 'center', fig.height = 5, fig.width = 8.5,
        pdf.options(encoding = "ISOLatin9.enc"))

library(class)
library(caret)
library(e1071)
library(dplyr)
library(jsonlite)
library(ggplot2)
library(ggthemes)
library(tidyverse)
library(gridExtra)
```


```{r}
#install.packages("fpp")
library(fpp)

# 1. SES MODEL FOR AUS AIR
data(ausair)

#returns a ts object
air = window(ausair, start = 1990, end = 2004)

# Always plot the data first!
plot(air,ylab = "Airline Passegners", xlab = "Year", main = "Airline Passengers")

#fit 3 different simple exponential smoothing models ... how are they different?
# what does the h paramter do?
fit1 = ses(air, initial = "simple",alpha = .9,h = 10)
fit2 = ses(air,initial = "simple",alpha = .8, h = 12)
fit3 = ses(air, h = 3) #defaults

# the forecast package has a nice accuracy funciton with various metrics just pass it the
# the model and the data!  (This is the "training" data)
accuracy(fit1, ausair)
accuracy(fit2, ausair)
accuracy(fit3, ausair)

#Reset the plot
plot(air,ylab = "Airline Passegners", xlab = "Year", type = "o", xlim = c(1990, 2008),ylim
= c(15,50), main = "Airline Passengers")

#Plot the estimated values from the models .. the "fitted" values are the training values.
lines(fitted(fit1), col = "blue", type = "o")
lines(fitted(fit2), col = "red", type = "o")
lines(fitted(fit3), col = "green", type = "o")

# the  $mean values are the forecasts.
lines(fit1$mean, col = "blue", type = "o")
lines(fit2$mean, col = "red", type = "o")
```
```

```
lines(fit3$mean, col = "green", type = "o")

# These are the actual values!  Compare visually with the forecasts!
air2008 = window(ausair, start = 1990, end = 2007)
points(air2008, type = "o")

# Compare the forecasts with the actual values with various fit metrics.
accuracy(fit1, air2008)
accuracy(fit2, air2008)
accuracy(fit3, air2008)



```

Activity 2 - Using the Holt-Winters model. This provides more details on weighting toward the front, seasonality and such. I think it may be challenging as we only have one data point per year. Perhaps the predict function will work?

```{r}
air = window(ausair, start = 1990, end = 2004)

#2 Holt's Linear Trend Model for AUS AIR
fit1h = holt(air, alpha = .8, beta = .2, initial = "simple", h = 5)
fit2h = holt(air, alpha = .8, beta = .2, initial = "simple", exponential = TRUE, h = 5)

# Check out estiamted values of the "training" data from the first holt model
fitted(fit1h)
# Check out the forecast value (h of them)
fit1h$mean

# Reset the Plot!
plot(air,ylab = "Airline Passegners", xlab = "Year", type = "o", xlim = c(1990, 2009),ylim
= c(15,60))
#Plot each models estimated values of the training data (Do these one by one to see the
differences)
lines(fitted(fit1h),col = "blue", type= "o")
lines(fitted(fit2h), col = "red", type= "o")
#Plot each models forecasts (Do these one by one to see the differences)
lines(fit1h$mean, col = "blue", type= "o")
lines(fit2h$mean,col = "red", type= "o")

# Fit another model ... damped!
fit3h = holt(air, alpha = .8, beta = .2, damped = TRUE, initial = "optimal", h = 5)
# Plot the fitted value (estimated from triaining data)
lines(fitted(fit3h), col = "darkgreen", type= "o")
# Plot the forecasts
lines(fit3h$mean,col = "darkgreen", type= "o")

# Fit another model ... what is the difference?
fit4h = holt(air, alpha = .8, beta = .5, damped = TRUE, initial = "optimal", exponential =
TRUE, h = 5)
# Plot the fitted value (estimated from triaining data)
lines(fitted(fit4h), col = "cyan", type= "o")
#Plot the forecasts
lines(fit4h$mean,col = "cyan", type= "o")

# with implicit Test set... it figures out by the time which are training and which are
test.
accuracy(fit1h, ausair)
accuracy(fit2h, ausair)
accuracy(fit3h, ausair)
```

```
#with explicit Test set ... (same output)
airTest = window(ausair, start = 2005)
accuracy(fit1h, airTest)
accuracy(fit2h, airTest)
accuracy(fit3h, airTest)

#Add the actual values to visually compare forecasts to actual values
air2008 = window(ausair, start = 1990, end = 2009)
points(air2008, type = "o")

```

Activity 3

```{r}
#3. Seasonal Trend

#Load the data
data("austourists")
# Read about the dataset!
?austourists


# Always plot the data first!
plot(austourists)

# returns a ts object.
aust = window(austourists,start = 1999, end = 2004)

#fit an additive and multiplicative model
fit1s = hw(aust,seasonal = "additive",h = 40)
fit2s = hw(aust,seasonal = "multiplicative",h = 40)

#Plot the original data
plot(aust,ylab = "Australian Tourists", xlab = "Year", type = "o", xlim = c(1999,
2014),ylim = c(15,60))
#add the fitted values from the model (of the training data)
lines(fitted(fit1s),col = "blue", type= "o")
lines(fitted(fit2s), col = "red", type= "o")

#Now add the forecasts (add these one at a time)
lines(fit1s$mean, col = "blue", type= "o")
lines(fit2s$mean,col = "red", type= "o")

#Compare the accuracy
accuracy(fit1s,austourists)
accuracy(fit2s,austourists)

#add the actual values to visually compare the forecasts to the actual values.
points(austourists, type = "o")

```
Activity 4
```{r}
#install.packages("fpp2")
library(fpp2)
```
```{r}
data(maxtemp)
#View(maxtemp)
summary(maxtemp)
```

1. Eliminate data before 1990

````
```{r}
maxtemp1 <- window(maxtemp, start = 1990, end = 2016)
View(maxtemp1)

```
````

2. Utilize SES to predict the next five years of maximum temperatures in Melbourne. Plot the information, including the prior data, the SES predictions and the forecast. Add the predicted value line across 1990-present as a separate line, preferably blue. So, to review, you should have the data, the predicted value line overlaying it, and a forecast through 2021, all on one plot. Find the AICc and BIC of this fitted model. You will use that information later.

````
```{r}
temp1 <- maxtemp[20:46]
#temp1

temp <- ts(temp1,frequency=12,start=1990, end = 2016)

temp2 <- window(temp, start = 1990, end = 2016)
View(temp2)

# Always plot the data first!
plot(temp2,ylab = "Temperature degrees Celsius", xlab = "Year", main = "Maximum temperatur
in Melbourne, Australia in degrees Celsius by year")

#fit 3 different simple exponential smoothing models ... how are they different?
# what does the h paramter do?
fit1 = ses(temp2, initial = "simple",alpha = .9,h = 26)
fit2 = ses(temp2,initial = "simple",alpha = .8, h = 31)
fit3 = ses(temp2, h = 26) #defaults

# the forecast package has a nice accuracy funciton with various metrics just pass it the
# the model and the data!  (This is the "training" data)
#accuracy(fit1, temp2)
#accuracy(fit2, temp2)
#accuracy(fit3, temp2)


#Reset the plot
plot(temp2,ylab = "Temperature in degrees Celsius", xlab = "Year", type = "o", xlim =
c(1971, 2021),ylim = c(15,50), main = "Airline Passengers")

#Plot the estimated values from the models .. the "fitted" values are the training values.
lines(fitted(fit1), col = "blue", type = "o")
lines(fitted(fit2), col = "red", type = "o")
lines(fitted(fit3), col = "green", type = "o")

# the  $mean values are the forecasts.
lines(fit1$mean, col = "blue", type = "o")
lines(fit2$mean, col = "red", type = "o")
lines(fit3$mean, col = "green", type = "o")

# These are the actual values!  Compare visually with the forecasts!
temp_real = window(maxtemp, start = 1971, end = 2016)
points(temp_real, type = "o")

# Compare the forecasts with the actual values with various fit metrics.
accuracy(fit1, temp_real)
accuracy(fit2, temp_real)
accuracy(fit3, temp_real)


```
````