APPM2720 HW5

The file **BB200.txt** in the Homework Directory contains results on the top 200 finishers from the 2014 Bolder Boulder 10K (6.2 miles) footrace. The columns are: place, names, sex/age division (M or F and age cateogry), place in the division, times for each mile, average mile time and the total time.

In the file HW5.R is a handy function **convertTime** to convert the times in the data set into the number of minutes e.g. `Mile1<- convertTime(results[,6])`

Don't forget to source this function in order to use it!

(1) Read in the data **BB200.txt** as a data frame.

(2) Convert the times to numerical values. Explain whether the first mile split time (MILE1) is predictive or not predictive of the place (PLACE). Include a figure to support your answer.

(3) Use the **substr** and **as.numeric** functions to make a new variable, AGE, from the division (DIV) variable. Also make another variable SEX based on the M/F division code. Make a plot of TIME as a function of AGE and color the points based on the SEX.

(4) The file **BB200Raw.txt** is closer to the raw data that was extracted from the web. `read.table` does not work on this file. What are the three separate aspects that need to be corrected for this file to be read in by **read.table** ?

(5) This is a review of smoothing from *Week04* and an example of cross-validation for investigating a data set. In the file HW5.R is a handy function **mySmooth3** that is a modification of the one that was written in class. How is it different?

Take a look at the WorldBankCO2 data set and work with the log10 GDP as the "x" and log10 CO2 as the "y". I.e. `x<- log10(WorldBankCO2$GDP.cap)` and `y <- log10(WorldBankCO2$CO2.cap)`

We want to determine a good choice for the **span** of the smoother.
For the spans .2, .5 and 1.0 smooth the data and find the mean sum of squared differences (MSE) between the smooth values and the data points.

For example,

```
yHat<- mySmooth3(x,y, xnew = x, span=.2) # note: two values returned x and y
MSE<- mean( (y- yHat[2])^2, na.rm = TRUE)
```

Now write a **for** loop where you omit each data point, smooth the remaining data but then predict at the omitted point. A handy way to do this in R is to use the convention that a minus index omitted a value. I.e. for the 10th data value and span = .2.

```
yHatCV<- matrix( NA, 75, 2) # CV stands for cross-validation
yHatCV[10,] <- mySmooth3(x[-10], y[-10], xnew = x[10], span=.2)
```

Using this procedure find the MSECV for the different spans, `MSECV<-mean( (y- yHatCV[,2])^2, na.rm = TRUE)` , How does it compare to the first way?

EXTRA CREDIT: Use a whole sequence of spans instead of 3 and make a plot of MSE and MSECV to compare them. Where is the minimum value for MSECV?