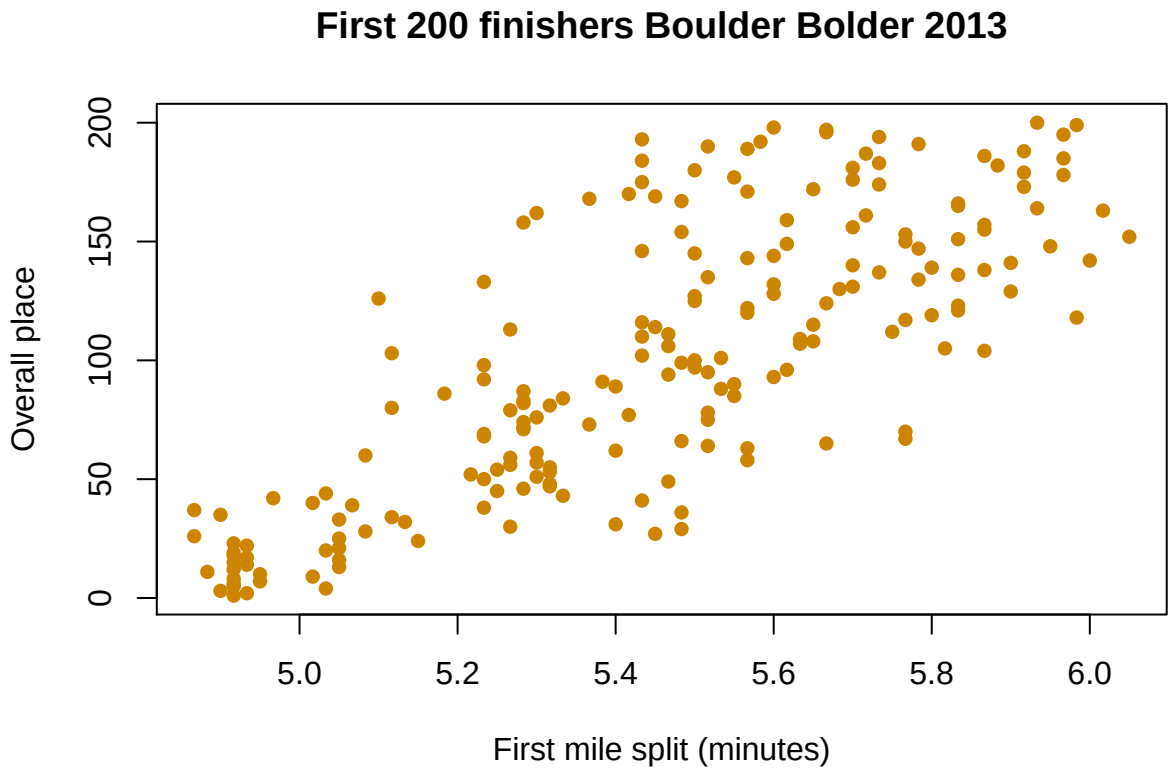
APPM2720 HW5

(1)

```
BB200<- read.table( "BB200.txt", skip=2,
                    stringsAsFactors=FALSE,
                    header=TRUE)
```

(2)

```
source("HW5.R")
timeMinutes<- convertTime(BB200$MILE1)
plot( timeMinutes, BB200$PLACE,
 xlab = "First mile split (minutes)" ,
 ylab = "Overall place", pch=16, col="orange3" )
title("First 200 finishers Boulder Bolder 2013")
```

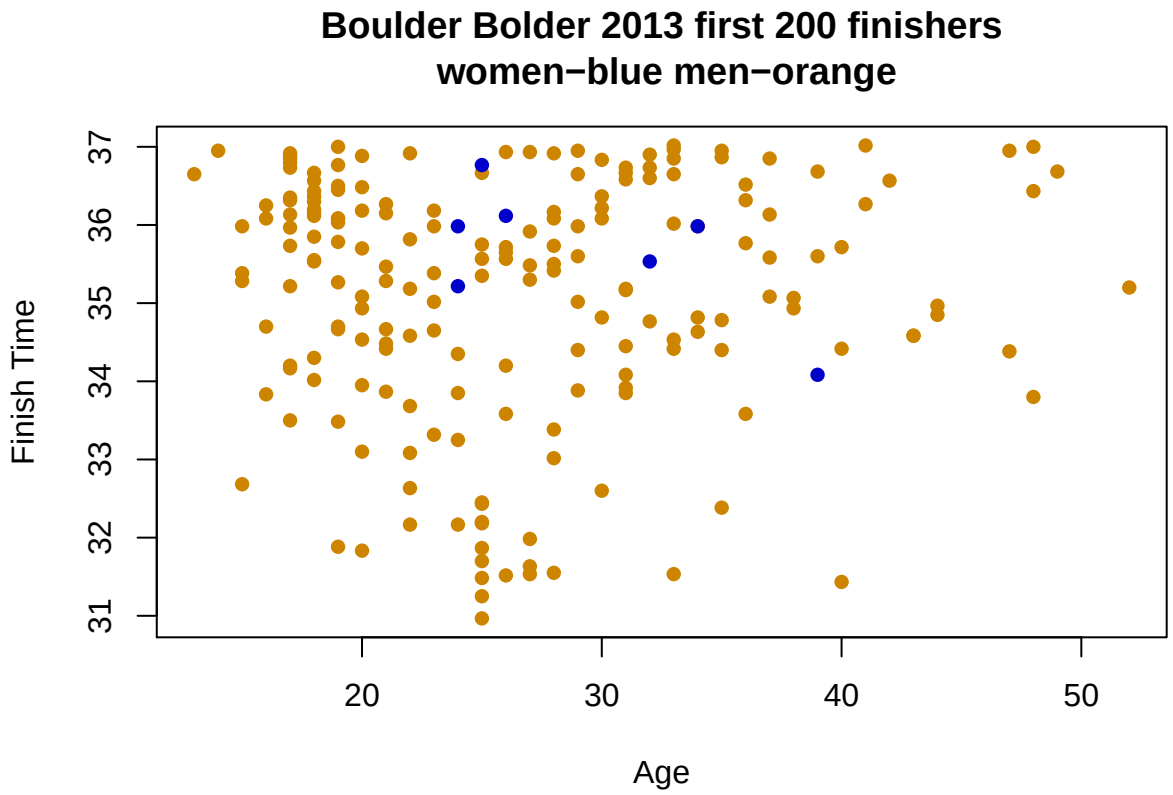


Yes there is some increasing relationship but not particularly strong. The main pattern is whether the runners split is above or below 5.1 minutes. Below they tend to place in the top 50 and above there would variability from ranks < 50 all the way to close to 200. There is also another break around 5.8 where runner above this split did not place in the top 100. Notice that some of the fastest runners appeared to be running in a tight pack all having about the same mile split (4.92).

(3)

```
SEX<- substr(BB200$DIV,1,1)
AGE<- as.numeric(  substr(BB200$DIV,2,3) )
# One way
colCode<- ifelse( SEX=="M","black","red")
# another way
colCode<- rep( NA, length(SEX))
colCode[ SEX=="M"] <-"orange3"
```

```
colCode[ SEX=="F"] <-"blue3"
Time<- convertTime(BB200$TIME )
plot(AGE, Time, col=colCode, xlab="Age", ylab="Finish Time", pch=16 )
title("Boulder Bolder 2013 first 200 finishers\n women-blue men-orange")
```




```
# A tricksy way using indices:
# colTable<- c(orange3, "blue3")
# names( colTable)<- c("M","F")
# plot(AGE, Time, col=colTable[SEX])
```

It looks like men in mid 20s do the best and increse in time for younger men. But the response of the 7 women is more even among age. Here is a useful set of boxplots for men grouping by ages.

```
library( fields)
```

```
## Loading required package: spam
```

```
## Loading required package: grid
```

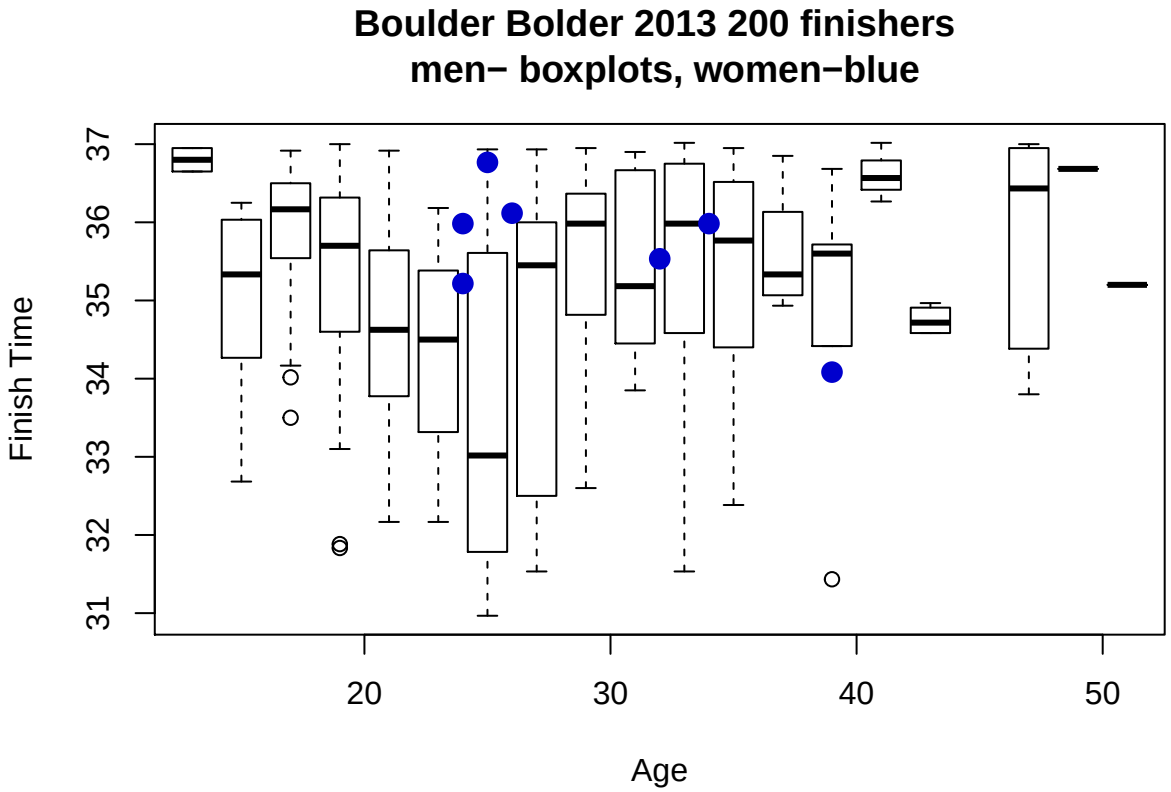
```
## Spam version 1.4-0 (2016-08-29) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

```
ind<- SEX=="M"
bplot.xy(AGE[ind], Time[ind], xlab="Age", ylab="Finish Time",N=15 )
points( AGE[!ind], Time[!ind], pch=16, col="blue3", cex=1.5)
title("Boulder Bolder 2013 200 finishers \n men- boxplots, women-blue")
```



Looking at the medians and the overlap of the boxes, the effect of age is less obvious except at 20-25. Note that the young women are slower than the men, but not the 3 older women.

(4) What are the three separate aspects that need to be corrected for this file to be read in by **read.table** ?

- Number 169 Jesse Parker Jr for example will be read as three strings not two
- Several runners have blank spaces where times were not entered. (Fill these in with NAs)
- The heading DIV PL will be read as the name for two heading not combined (I changed to DIV/PL in the cleaned up file.) If you are a UNIX person the quick way to find the differences is in UNIX `diff BB200Raw.txt BB200.txt`

(5)

```
library(dataWorkshop)
data(WorldBankCO2)
x<- log10(WorldBankCO2$GDP.cap)
y <- log10(WorldBankCO2$CO2.cap)
```

MSE for smoothing full data

```
yHat<-  mySmooth3(x,y, xnew =  x, .2) # want x in same order as y
mean( (y - yHat[,2])^2, na.rm=TRUE)
```

```
## [1] 0.2844177
```

```
yHat<-  mySmooth3(x,y, xnew =  x, .5)
mean( (y - yHat[,2])^2, na.rm=TRUE)
```

```
## [1] 0.3075332
```

```
yHat<-  mySmooth3(x,y, xnew =  x, 1.0)
mean( (y - yHat[,2])^2, na.rm=TRUE)
```

```
## [1] 0.3529773
```

It is helpful to write a function that handles the case of one span and does the cross-validation. Note that this is hard-wired for 75 obs and x and y as the data set.

```
findCVMSE<- function( span){
hold<- rep( NA, 75)
for( i in 1:75){
    output<- mySmooth3(x[-i], y[-i],x[i],span)
    hold[i]<- ( y[i] - output[2])^2
  }
  return(  mean( hold, na.rm=TRUE) )
}
```

CV MSE for the three spans.

```
findCVMSE(.2)
```

```
## [1] 0.347754
```

```
findCVMSE(.5)
```

```
## [1] 0.3378149
```

```
findCVMSE(1.0)
```

```
## [1] 0.3802338
```

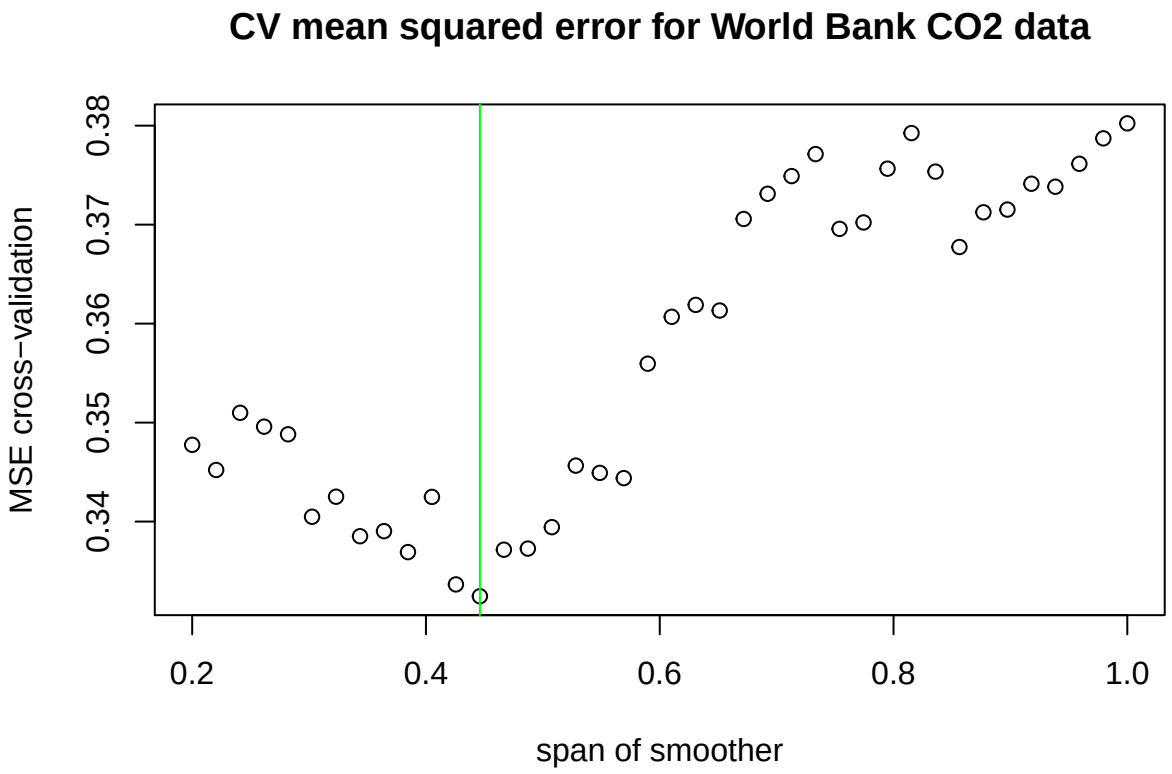
MSE decrease with span and be less than the CV MSE. This is because the MSE uses the data that is also predicting to.

Extra credit:

```
  span<- seq( .2, 1.0, length.out=40)
  CVMSE<- matrix( NA,40)
  MSE<- matrix( NA,40)
for( k in 1:40){
  CVMSE[k] <- findCVMSE(span[k])
  output<- mySmooth3(x, y ,x,span[k] )
  MSE[k]<- mean( (y- output[,2])^2 , na.rm=TRUE)
}
```

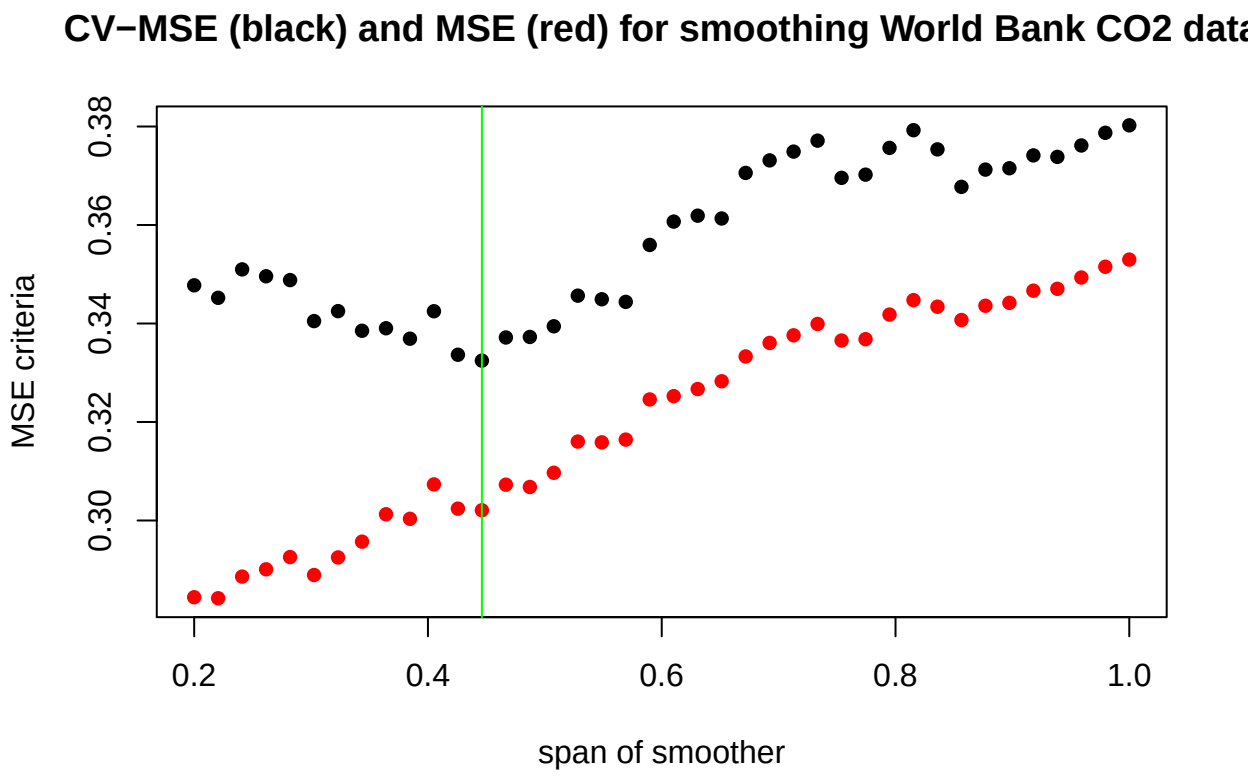
OK the plot you have all been waiting for! Minimum around .45 so suggests to use that size of span for best smoothing results.

```
  plot( span, CVMSE, xlab="span of smoother", ylab= "MSE cross-validation")
  title( "CV mean squared error for World Bank CO2 data")
  best<- span[ which.min(CVMSE)]
  abline( v= best, col="green")
```



Here is the plot adding the MSE too. This will just decrease steady as span decreases and so is not useful for picking the span.

```
matplot( span, cbind(CVMSE,MSE) , xlab="span of smoother", ylab= "MSE criteria", type="p", pch=16)
title( "CV-MSE (black) and MSE (red) for smoothing World Bank CO2 data")
best<- span[ which.min(CVMSE)]
abline( v= best, col="green")
```


CV−MSE (black) and MSE (red) for smoothing World Bank CO2 data
MSE criteria
0.30
0.32
0.34
0.36
0.38
0.2
0.4
0.6
0.8
1.0
span of smoother