Lab 2

Maria Barouti

1/22/2020

Exercise 1 - Degrees of Flexibility. . . Regression vs Smooth Splines

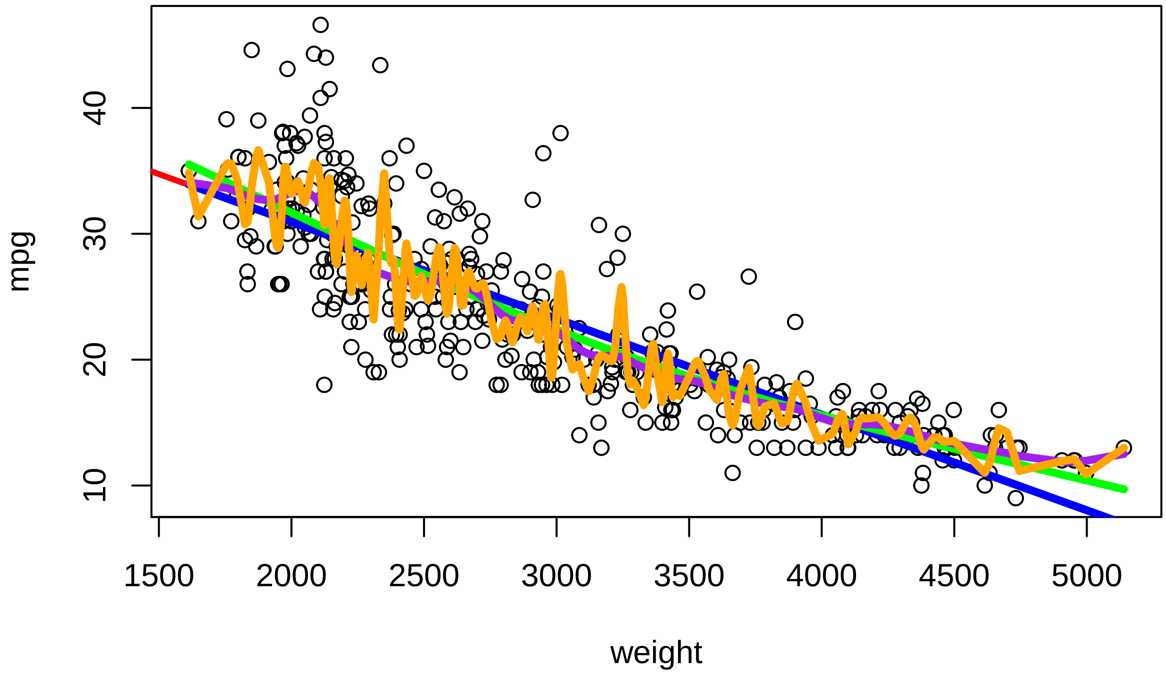
a) Fit a Regression model that predicts miles per gallon based on weight and then plot the regression line in red color with thickness=3.

b) Fit a spline with 2 degrees of freedom to our data and then plot it.What do you observe?

c) Increase the flexibility of our model by adding more d.f.(try 3,20,100 d.f.)

d) Is the last spline produced above flexible? Does it match the data well? Would this model be powerful for prediction?

```r
## Solution
setwd("C:/Users/barouti/Desktop/Statistical Machine Learning/Data")
load("Auto.rda")
attach(Auto)

reg = lm(mpg~weight)
plot(weight,mpg)
abline(reg,col="red",lwd=3)

spline2 = smooth.spline(weight,mpg,df=2)
lines(spline2,col="blue",lwd=4)

spline3 = smooth.spline(weight,mpg,df=3)
lines(spline3,col="green",lwd=4)

spline20 = smooth.spline(weight,mpg,df=20)
lines(spline20,col="purple",lwd=4)

spline100 = smooth.spline(weight,mpg,df=100)
lines(spline100,col="orange",lwd=4)
```



Exercise 2 - How to choose the optimal method? Cross-validation technique

a) Divide tha data equally into two groups (training and testing).

b) Fit a spline with 5 degrees of freedom and then compute the prediction mean squared error on the testing data.

c) Try many different splines and choose the one with the smallest prediction error (Hint: make a loop).

d) Make a plot, where the x-axis will represent the degrees of freedom and y-axis will represent the cross validation error (or prediction mean squared error).

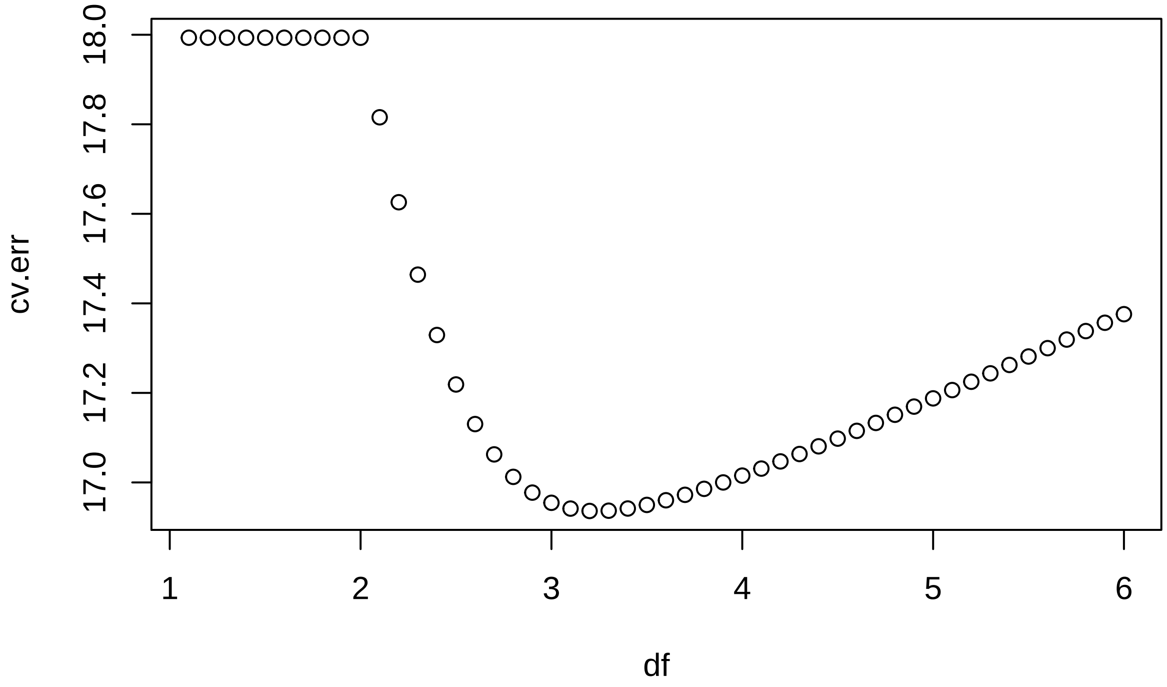
```
## Solution

n = length(mpg)
Z = sample(n,n/2)
attach(Auto[Z,])
ss5 = smooth.spline(weight,mpg,df=5)

attach(Auto)
Yhat = predict(ss5,x=weight)
names(Yhat)
```

```
## [1] "x" "y"
```

```
mean((Yhat$y[-Z]-mpg[-Z])^2)
```

```
## [1] 17.18777
```

```
cv.err = rep(0,50)
for (p in 1:50){
  attach(Auto[Z,])
  ss = smooth.spline(weight, mpg, df=1+p/10)
  attach(Auto)
  Yhat = predict(ss,weight)
  cv.err[p] = mean((Yhat$y[-Z]-mpg[-Z])^2)
}
df = 1+(1:50)/10
plot(df,cv.err)
```



```
which.min(cv.err)
```

```
## [1] 22
```

```
df[which.min(cv.err)]
```

```
## [1] 3.2
```