HW1

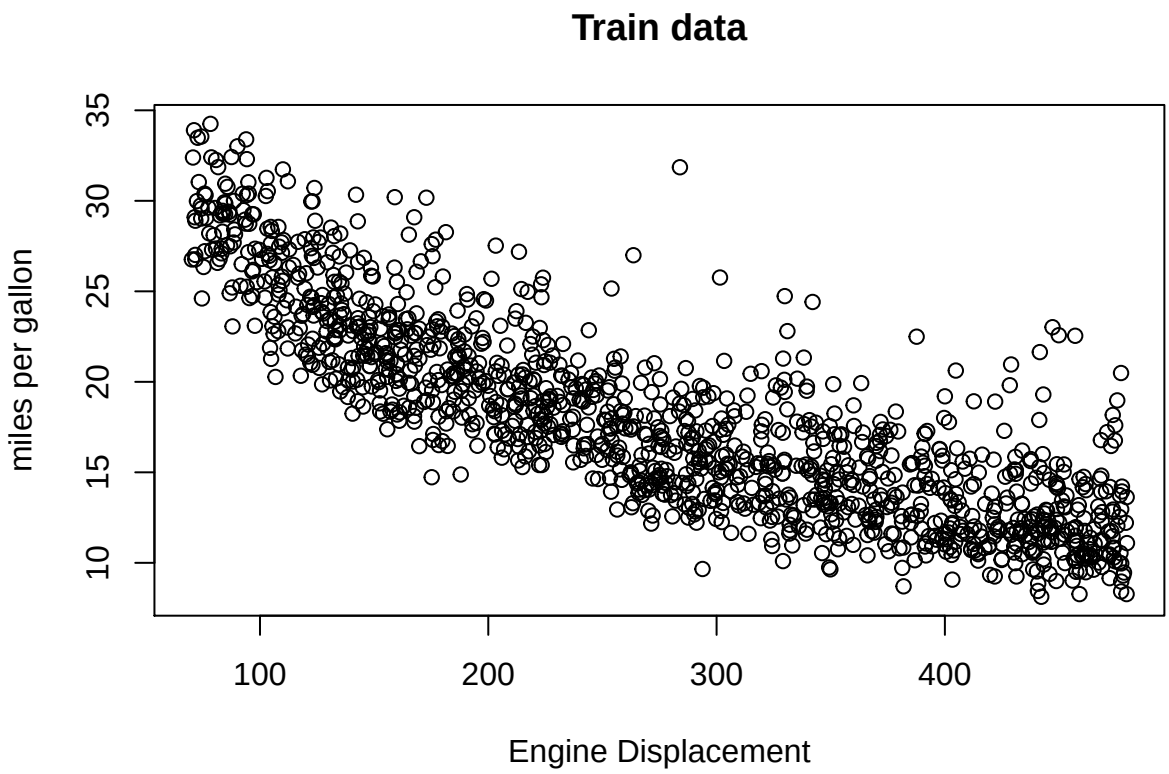
Shaojie Zhang (shaojiez)

01/23/2019

Problem 1

(a)

```
load("engine.Rdata")
x = c(engine.xtrain)
y = c(engine.ytrain)
plot(x,y,xlab = "Engine Displacement", ylab = "miles per gallon", main = "Train data")
```



We see that there seems to be a linear relationship between x and y, and the slope is downward.

(b)

```
library(np)
```

```
## Warning: package 'np' was built under R version 3.4.4
```

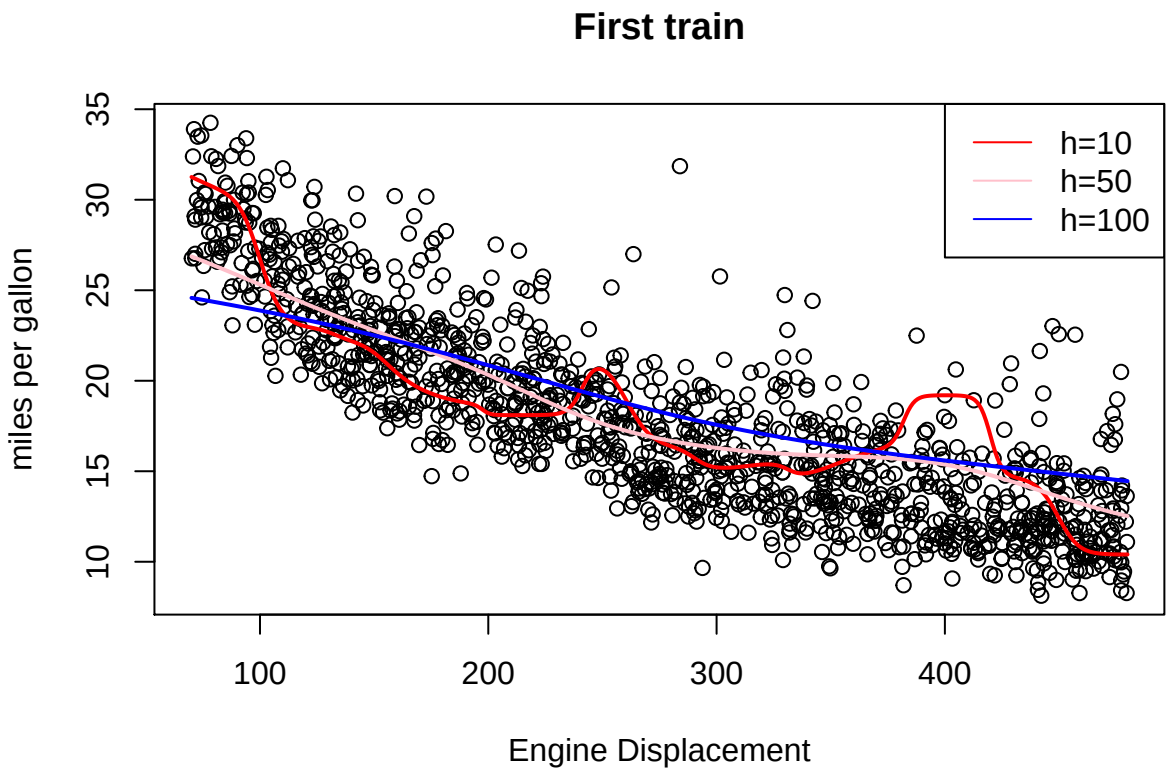
```
x1 = engine.xtrain[,1]
y1 = engine.ytrain[,1]
first=data.frame(x = x1, y = y1)

plot(x, y, xlab="Engine Displacement", ylab="miles per gallon", main="First train")
```

```
h = c(10, 50, 100)
kregobj1=npreg(y~x,data=first,bws=h[1])
kregobj2=npreg(y~x,data=first,bws=h[2])
kregobj3=npreg(y~x,data=first,bws=h[3])

lines(x0,predict(kregobj1,newdata=data.frame(x=x0)),col="red",lwd=2)
lines(x0,predict(kregobj2,newdata=data.frame(x=x0)),col="pink",lwd=2)
lines(x0,predict(kregobj3,newdata=data.frame(x=x0)),col="blue",lwd=2)

legend("topright",col=c("red","pink","blue"), lty=c(1,1,1), legend=c("h=10","h=50","h=100"))
```

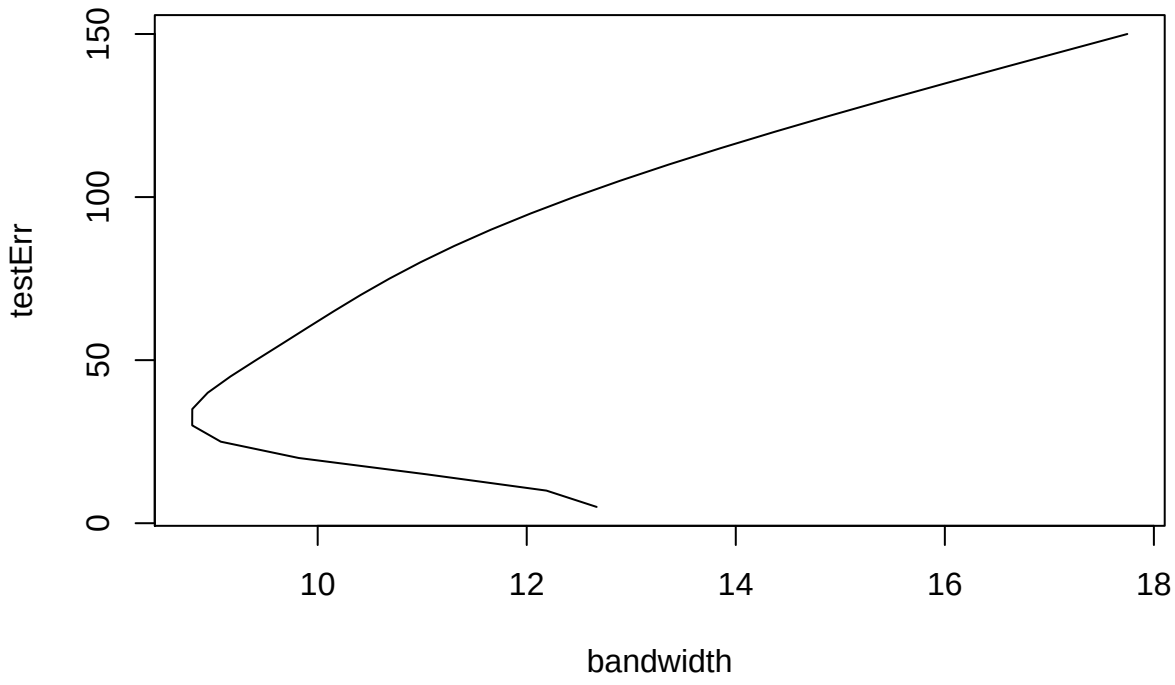


(c)

We can see from the graph that as the bandwidth gets bigger, the fitted line gets flatter because of oversmoothing. This reminds me of the k-nn demo we did in class.

(d)

```
bandwidths = c(1:30)*5
testErr = numeric(30)
for (i in (1:30)) {
  kregobj = npreg(y~x, data=first, bws=bandwidths[i])
  error = mean((engine.ytest[,1] - predict(kregobj, newdata=data.frame(x=engine.xtest[,1])))^2)
  testErr[i] = error
}
plot(testErr, bandwidths, xlab = "bandwidth", ylab = "testErr", type = "l")
```



(e)

```
index = which.min(testErr)
print(bandwidths[index])
```

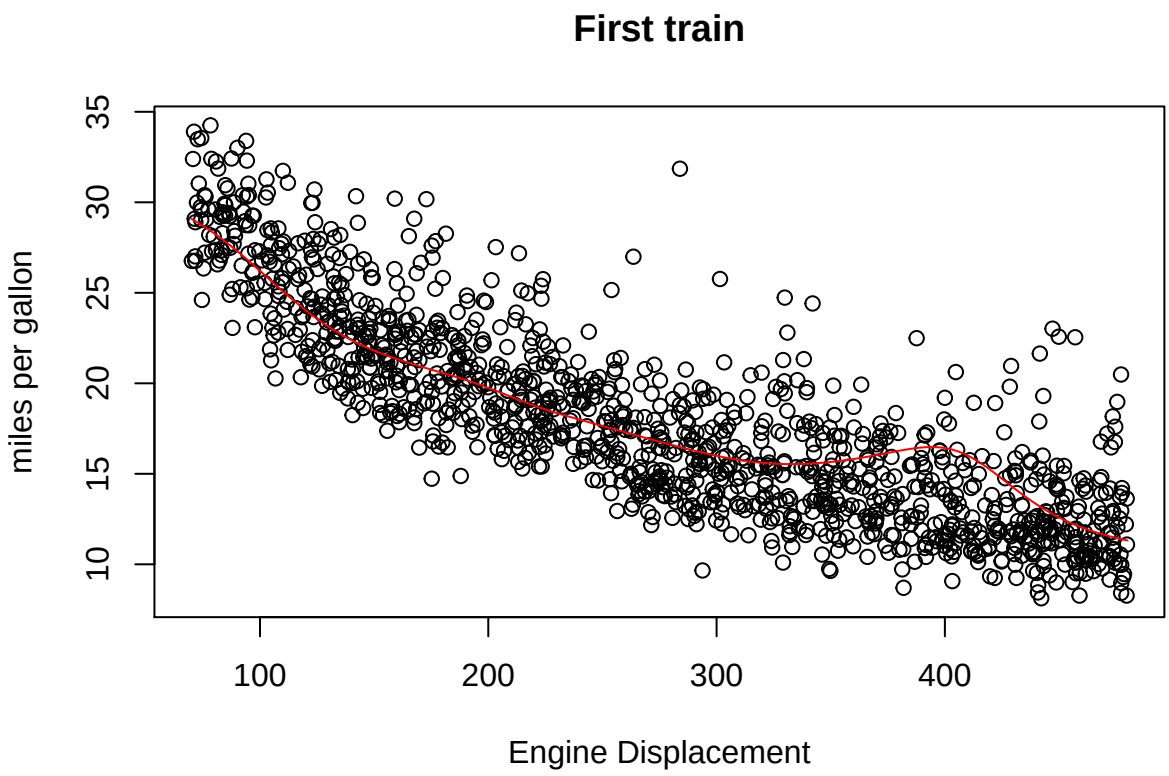
```
## [1] 30
```

```
print(testErr[index])
```

```
## [1] 8.799357
```

From above, we see that the optimal bandwidth value is 30, and the associated test error is 8.788357

```
plot(x,y,xlab="Engine Displacement", ylab="miles per gallon", main="First train")
kregobj = npreg(y~x, data=first, bws=30)
lines(x0,predict(kregobj, newdata=data.frame(x=x0)), col = "red")
```



From the plot, it seems this is a pretty good value at the beginning because it goes through all the training data points in the middle. But towards the end it seems to bump a little bit which might be off.

(f)

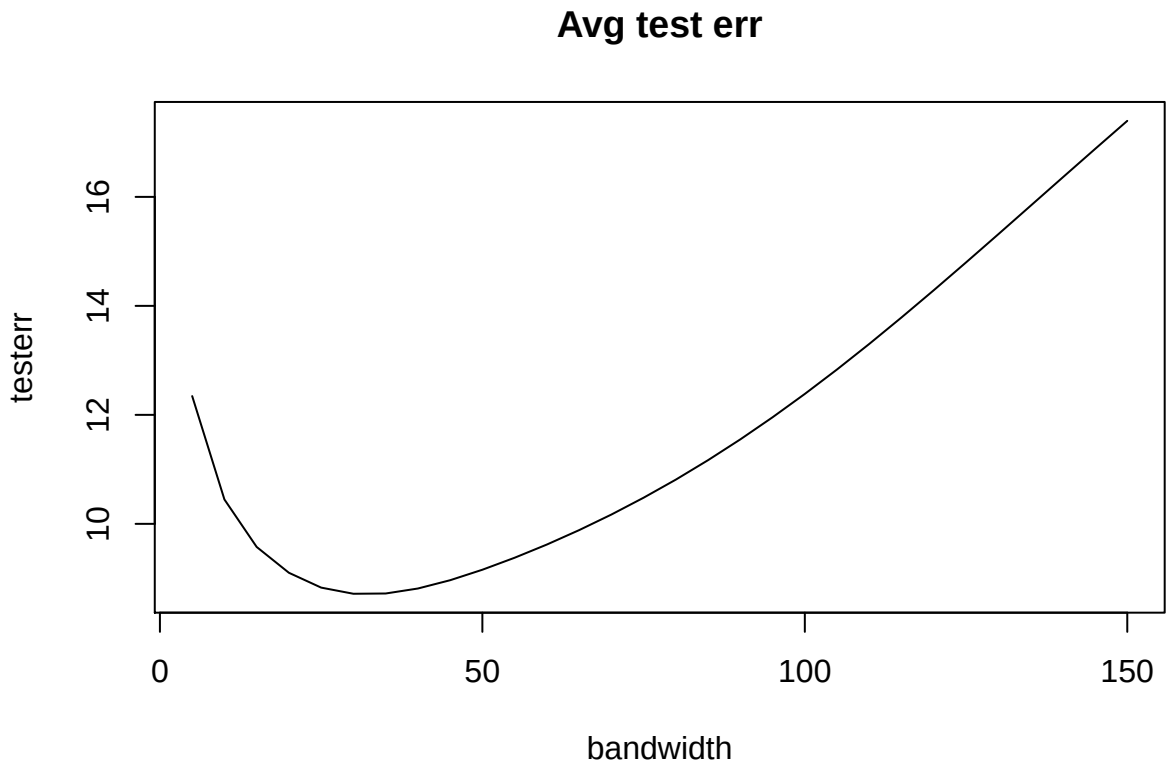
```
testErrMatrix = matrix(0, 30, 40)
for (i in 1:40) {
  x = engine.xtrain[,i]
  y = engine.ytrain[,i]
  ithdata = data.frame(x=x, y=y)

for (j in 1:30){
  kregobj = npreg(y~x, data=ithdata, bws = bandwidths[j])
  error = mean((engine.ytest[,i] - predict(kregobj, newdata=data.frame(x=engine.xtest[,i])))^2)
  testErrMatrix[j, i] = error
}
}

testErrMean = numeric(30)
for (k in 1:30){
  testErrMean[k] = mean(testErrMatrix[k,])
}
```

Now plot the test error against bandwidths

```
plot(bandwidths, testErrMean, type = "l", xlab = "bandwidth", ylab = "testerr", main = "Avg test err")
```



I now see the graph that is close to the one we saw in the lecture.

```
newindex = which.min(testErrMean)
print(bandwidths[newindex])
```

```
## [1] 30
```

```
print(testErrMean[newindex])
```

```
## [1] 8.716932
```

So the optimal value of bandwidth does not change, and the associated test error is a little bit smaller than before.

Problem 2

Please see a seperate scan pdf file