HW6

Xinyue Lu

2020/3/3

Implement the Nadaraya-Watson kernel regression based on the Epanechnikov and Gaussian kernels. Use rule of thumb and cross-validation to determine the bandwidth parameter h. For cross validation,implement grid search around the rule of thumb estimate. Compare the results. The data set is Wage, you can find it in the library(ISLR) use attach(Wage)

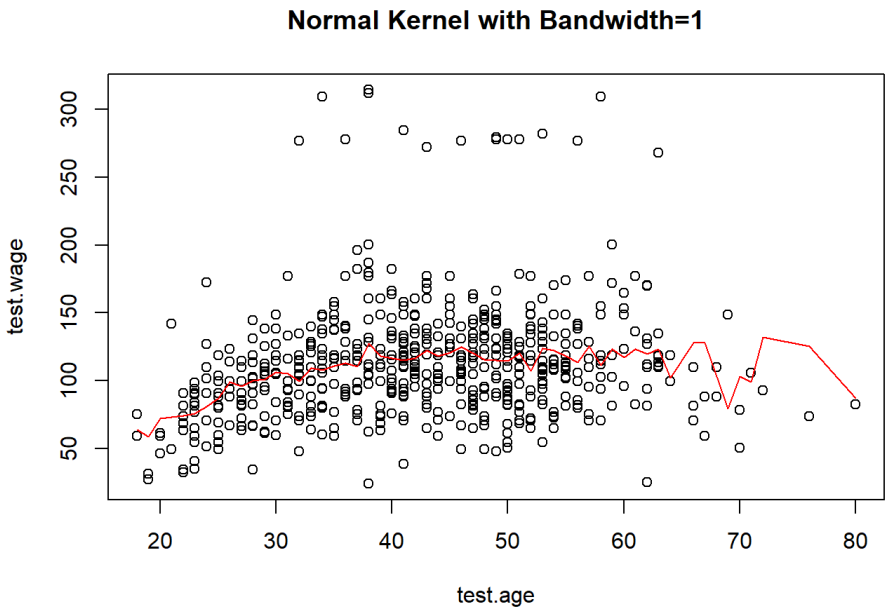
Normal Kernel by ksmooth

First, try 3 bandwidths 1, 5 and 10. Tt showed that bandwidth 5 produce the most preferable estimates, because the other two were either too rough or too smooth. Next, use cross validation to search for the best bandwidths from 5 to 9 by 0.2 a step. The best bandwidth was 7, which produced the smallest MSE 1595.884.

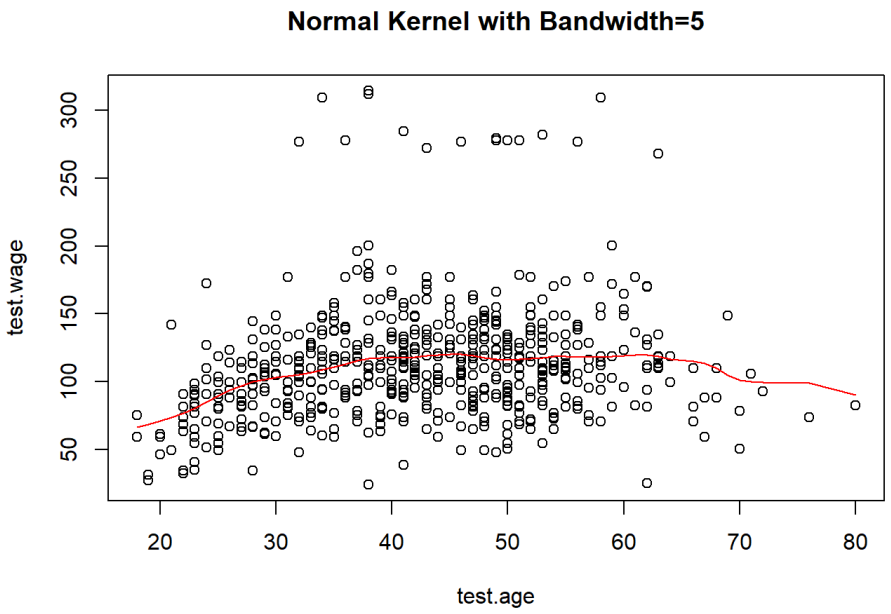
```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.2
```

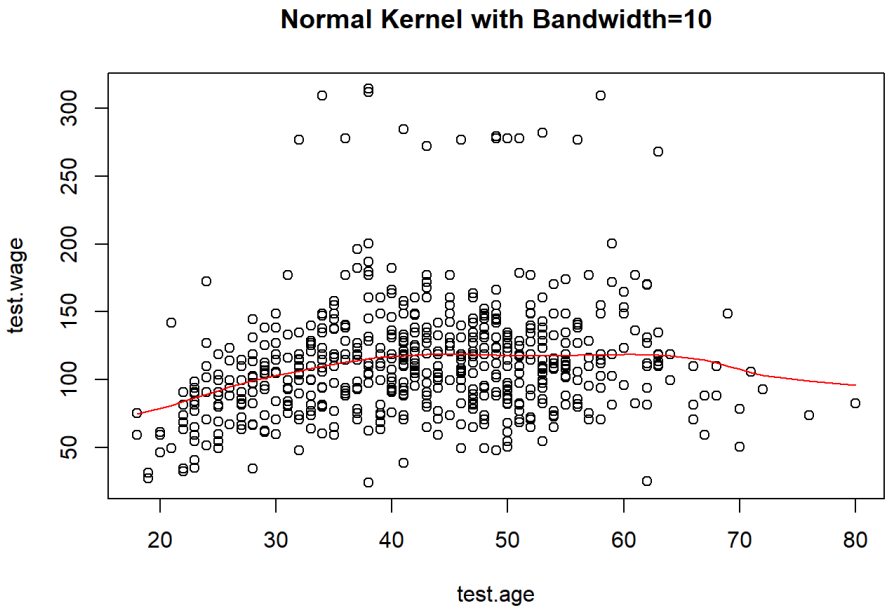
```
attach(Wage)

# set cross validation set
set.seed(666)
idx = sample(seq(1:3000))
index = cbind(idx[1:600],idx[601:1200],idx[1201:1800],idx[1801:2400],idx[2401:3000])

# choose the bandwidth by plots
test = index[,1]
train = (-test)
train.age = age[train]
train.wage = wage[train]
test.age = age[test]
test.wage = wage[test]
# bandwidth =1
pred = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = 1, range.x = range(train.age),
        n.points = length(train.age),x.points = test.age)
plot(test.age,test.wage)
lines(pred$x,pred$y,col="red")
title("Normal Kernel with Bandwidth=1")
```




```
# bandwidth =5
pred = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = 5, range.x = range(train.age),
        n.points = length(train.age),x.points = test.age)
plot(test.age,test.wage)
lines(pred$x,pred$y,col="red")
title("Normal Kernel with Bandwidth=5")
```




```
# bandwidth =10
pred = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = 10, range.x = range(train.age),
       n.points = length(train.age),x.points = test.age)
plot(test.age,test.wage)
lines(pred$x,pred$y,col="red")
title("Normal Kernel with Bandwidth=10")
```



```
##############################################
# use cross validation to find best bandwidth
mse = matrix(0,20,5)
for(h in 1:20){
b=h*0.2+5
for(j in (1:5)){
test = index[,j]
train = (-test)
train.age = age[train]
train.wage = wage[train]
test.age = age[test]
test.wage = wage[test]

#fit = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = b, range.x = range(train.age), n.points = length(train.age))
#plot(train.age,train.wage)
#lines(fit$x,fit$y,col="red")

pred = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = b, range.x = range(train.age),
        n.points = length(train.age),x.points = test.age)
#plot(test.age,test.wage)
#lines(pred$x,pred$y,col="red")

mat = cbind(test.age,test.wage)
o = order(mat[,"test.age"])
mat = mat[o,]
mse[h,j]=mean((mat[,"test.wage"]-pred$y)^2)
}}

# MSE of the cross validation with bandwidth [5 to 9]
MSE = rowMeans(mse)
MSE
```

```
##  [1] 1596.903 1596.716 1596.556 1596.422 1596.314 1596.233 1596.175
##  [8] 1596.142 1596.134 1596.147 1596.182 1596.238 1596.321 1596.420
## [15] 1596.538 1596.681 1596.839 1597.015 1597.213 1597.425
```

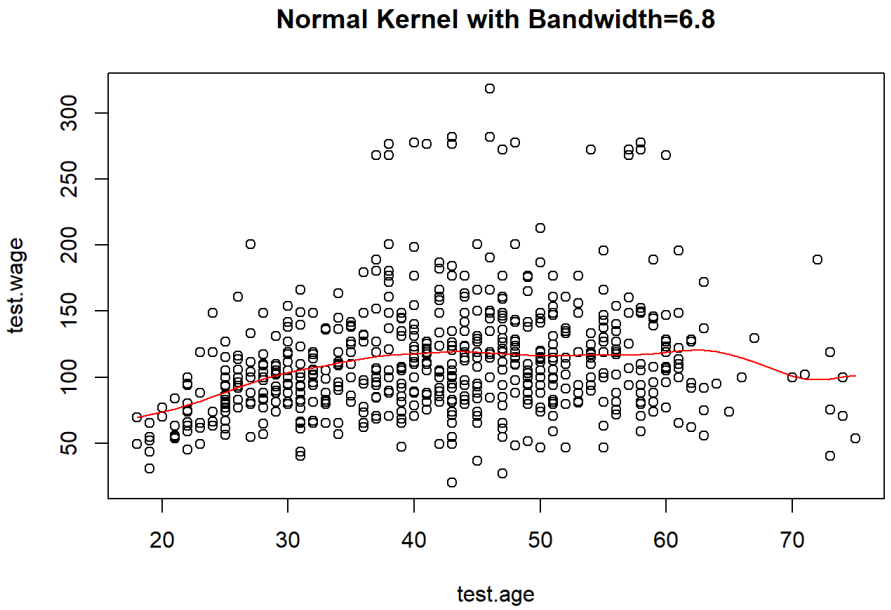
```
min(MSE)
```

```
## [1] 1596.134
```

```
# Find the Bandwidth with Minimal MSE
index = which(MSE == min(MSE))
best.bandwidth = index*0.2+5
best.bandwidth
```

```
## [1] 6.8
```

```
pred = ksmooth(train.age, train.wage, kernel = "normal", bandwidth = 6.8, range.x = range(train.age),
        n.points = length(train.age),x.points = test.age)
plot(test.age,test.wage)
lines(pred$x,pred$y,col="red")
title("Normal Kernel with Bandwidth=6.8")
```


Normal Kernel with Bandwidth=6.8
test.wage
50
100
150
200
250
300
20
30
40
50
60
70
test.age