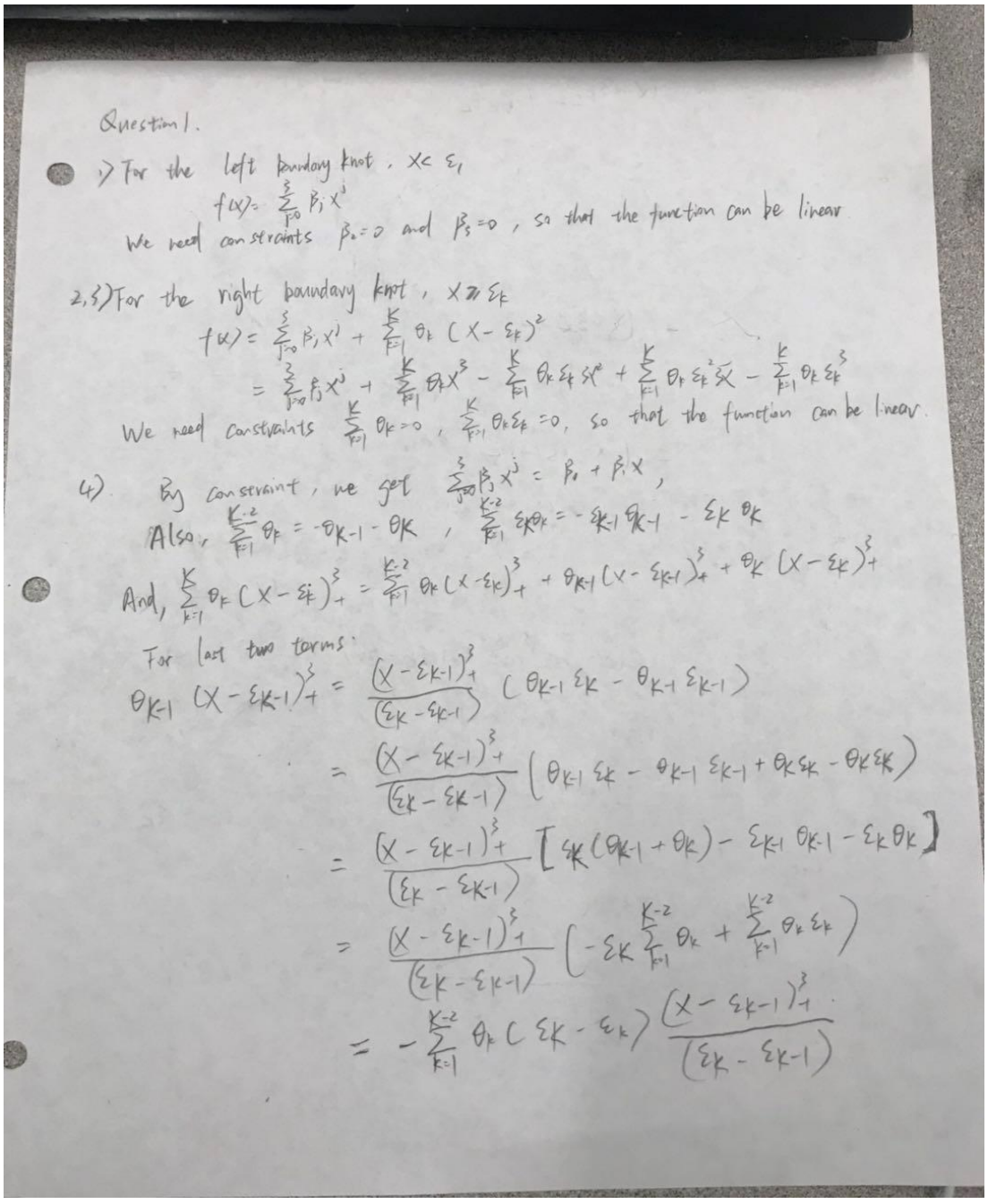
STAT 542 HW 4

Xiruo Li (xiruoli2)

Question 1



Question 1.

1) For the left boundary knot, $X < \xi_1$

$f(x) = \sum_{j=0}^{3} \beta_j X^j$

We need constraints $\beta_2 = 0$ and $\beta_3 = 0$, so that the function can be linear.

2,3) For the right boundary knot, $X \geq \xi_K$

$$f(x) = \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k (X - \xi_k)_+^3$$

$$= \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k X^3 - \sum_{k=1}^{K} \theta_k \xi_k 3X^2 + \sum_{k=1}^{K} \theta_k \xi_k^2 3X - \sum_{k=1}^{K} \theta_k \xi_k^3$$

We need constraints $\sum_{k=1}^{K} \theta_k = 0$, $\sum_{k=1}^{K} \theta_k \xi_k = 0$, so that the function can be linear.

4) By constraint, we get $\sum_{j=0}^{3} \beta_j X^j = \beta_0 + \beta_1 X$,

Also, $\sum_{k=1}^{K-2} \theta_k = -\theta_{K-1} - \theta_K$, $\sum_{k=1}^{K-2} \xi_k \theta_k = -\xi_{K-1}\theta_{K-1} - \xi_K \theta_K$

And, $\sum_{k=1}^{K} \theta_k (X - \xi_k)_+^3 = \sum_{k=1}^{K-2} \theta_k (X - \xi_k)_+^3 + \theta_{K-1}(X - \xi_{K-1})_+^3 + \theta_K (X - \xi_K)_+^3$

For last two terms:

$$\theta_{K-1}(X - \xi_{K-1})_+^3 = \frac{(X - \xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})} (\theta_{K-1}\xi_K - \theta_{K-1}\xi_{K-1})$$

$$= \frac{(X - \xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})} (\theta_{K-1}\xi_K - \theta_{K-1}\xi_{K-1} + \theta_K \xi_K - \theta_K \xi_K)$$

$$= \frac{(X - \xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})} [\xi_K(\theta_{K-1} + \theta_K) - \xi_{K-1}\theta_{K-1} - \xi_K \theta_K]$$

$$= \frac{(X - \xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})} \left(-\xi_K \sum_{k=1}^{K-2} \theta_k + \sum_{k=1}^{K-2} \theta_k \xi_k\right)$$

$$= -\sum_{k=1}^{K-2} \theta_k (\xi_K - \xi_k) \frac{(X - \xi_{K-1})_+^3}{(\xi_K - \xi_{K-1})}$$

$$\theta_k (X-\xi_k)_+^3 = \frac{(X-\xi_k)_+^3}{\xi_k - \xi_{k-1}} (\theta_k \xi_{k-1} - \theta_k \xi_{k-1} + \theta_{k-1}\xi_{k-1} - \theta_{k-1}\xi_{k-1})$$

$$= \frac{(X-\xi_k)_+^3}{(\xi_k - \xi_{k-1})} [-\xi_{k-1}(\theta_{k-1}+\theta_k) + \xi_{k-1}\theta_{k-1} + \xi_k \theta_k]$$

$$= \frac{(X-\xi_k)_+^3}{(\xi_k - \xi_{k-1})} \left(\xi_{k-1} \sum_{k=1}^{K-2} \theta_k - \sum_{k=1}^{K-2} \theta_k \xi_k \right)$$

$$= (X-\xi_k)_+^3 \sum_{k=1}^{K-2} \theta_k \frac{\xi_{k-1} - \xi_k}{\xi_k - \xi_{k-1}}$$

$$= (X-\xi_k)_+^3 \sum_{k=1}^{K-2} \theta_k (\xi_{k-1} - \xi_k) \frac{\xi_{k-1} - \xi_k + \xi_k - \xi_k}{(\xi_k - \xi_{k-1})(\xi_k - \xi_k)}$$

$$= (X-\xi_k)_+^3 \sum_{k=1}^{K-2} \theta_k (\xi_{k-1} - \xi_k) \left(\frac{1}{\xi_k - \xi_{k-1}} - \frac{1}{\xi_k - \xi_k} \right)$$

Combine these terms.

$$\sum_{k=1}^{K} \theta_k (X-\xi_k)_+^3 = \sum_{k=1}^{K-2} \theta_k (X-\xi_k)_+^3 + \theta_{k-1}(X-\xi_{k-1})_+^3 + \theta_k (X-\xi_k)_+^3$$

plug in,

$$= \sum_{k=1}^{K-2} \theta_k (X-\xi_k)_+^3 - \sum_{k=1}^{K-2} \theta_k (\xi_k - \xi_k) \frac{(X-\xi_{k-1})_+^3}{(\xi_k - \xi_{k-1})}$$

$$+ (X-\xi_k)_+^3 \sum_{k=1}^{K-2} \theta_k (\xi_{k-1} - \xi_k) \left(\frac{1}{\xi_k - \xi_{k-1}} - \frac{1}{\xi_{k-1} - \xi_k} \right)$$

$$= \sum_{k=1}^{K-2} \theta_k (\xi_k - \xi_k) \left[\frac{(X-\xi_k)_+^3}{\xi_k - \xi_k} - \frac{(X-\xi_{k-1})_+^3}{\xi_k - \xi_{k-1}} + \frac{(X-\xi_k)_+^3}{\xi_k - \xi_{k-1}} - \frac{(X-\xi_k)_+^3}{\xi_k - \xi_k} \right]$$

$$= \sum_{k=1}^{K-2} \theta_k (\xi_k - \xi_k) \left[\frac{(X-\xi_k)_+^3 - (X-\xi_k)_+^3}{\xi_k - \xi_k} - \frac{(X-\xi_{k-1})_+^3 - (X-\xi_k)_+^3}{\xi_k - \xi_{k-1}} \right]$$

Therefore,

$$f(X) = \beta_0 + \beta_1 X + \sum_{k=1}^{K-2} \alpha_k (d_k(X) - d_{k-1}(X))$$

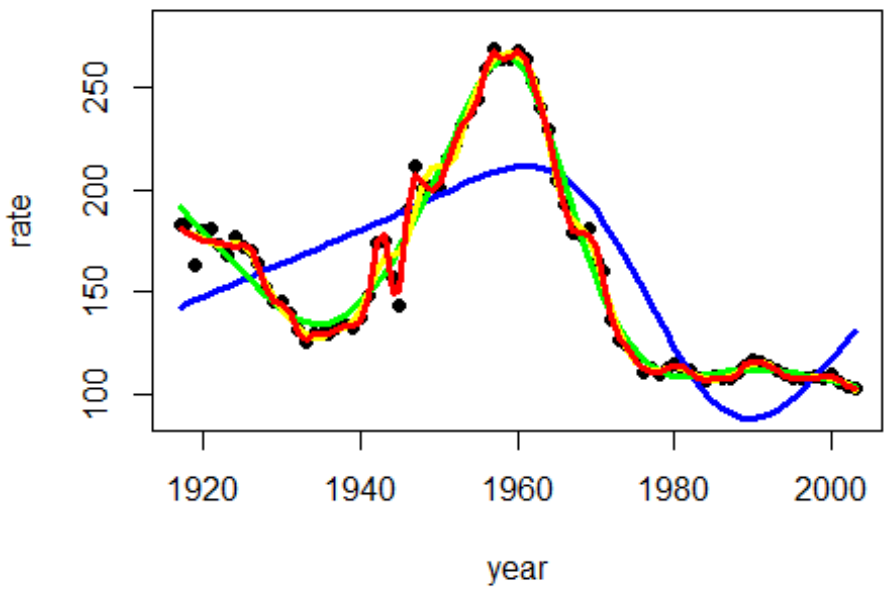
where, $d_k(X) = \dfrac{(X-\xi_k)_+^3 - (X-\xi_K)_+^3}{\xi_K - \xi_k}$ and $\alpha_k = \theta_k (\xi_K - \xi_k)$

Question 2

```
source("C:/Users/Xiruo Li/Desktop/STAT542/HW/hw4/birthrates.txt");
birthrates = as.data.frame(birthrates)
names(birthrates) = c("year", "rate")

plot(birthrates$year, birthrates$rate, ylim=c(90,280), xlab = "year", y
lab = "rate", pch = 19)
color_list=c('blue','green','yellow','red')
n=1
pos <- function(x) x*(x>0)
for (K in c(5,10,30,50))
{
  myknots = quantile(birthrates$year, seq(1/K,1,1/K))
  mybasis=birthrates$year
  for (i in 1:(K-2))
  {
    dk=(pos(birthrates$year - myknots[i])^3-pos(birthrates$year - mykno
ts[K])^3)/(myknots[K]-myknots[i])
    dK=(pos(birthrates$year - myknots[K-1])^3-pos(birthrates$year - myk
nots[K])^3)/(myknots[K]-myknots[K-1])
    Nk=dk-dK
    mybasis=cbind(mybasis,Nk)
  }
  mydata = data.frame(mybasis, rate = birthrates$rate)
  myfit = lm(rate ~., data = mydata)

  lines(birthrates$year, predict(myfit, mydata), col=color_list[n],lty=
1, lwd = 3)
  n=n+1
}
```



I choose number of knots in (5,10,30,50), which colored by ('blue','green','yellow','red'). When knots number is 5, the splines don't fit well. When the knots number is 10, the splines fit well, which is the best model. When the knots number is 30 or 50, the splines become overfitting.

NCS vs B splines: NCS don't have local control, while B splines have it. Also, NCS have interpolation, while B splines don't have it. The advantage of NCS is it force the second and third derivatives to be 0 at the boundaries, so that the splines become smooth at the boundary. But, the problem is that this method lose some information at the boundaries.

Question 3

(a)

```
library(randomForest)

## Warning: package 'randomForest' was built under R version 3.3.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

library(MASS)
#generate data
set.seed(1)
```

```
n=200
p=20
X=mvrnorm(n , mu = rep(0,p), Sigma = diag(p))
mean_Y=1+0.5*apply(X[,1:4],1,sum)
y_pred=matrix(nrow=20,ncol=200)
#Tune mtry and nodesize
mtry=c(1,5,10,15)
nodesize=c(1,10,30,50)
n_mtry=length(mtry)
n_nodesize=length(nodesize)
dof=matrix(0,n_nodesize,n_mtry)
for ( i in 1:n_nodesize)
{
  for (j in 1:n_mtry)
  {
    y=t(matrix(rep(mean_Y,20),nrow=200,ncol=20))
    for (k in 1:20)
    {
      y[k,]=y[k,]+rnorm(200)
      RF=randomForest(X,y[k,],mtry=mtry[j],nodesize = nodesize[i])
      y_pred[k,]=predict(RF,newdata=X)
    }
    for (l in 1:200)
    {
      dof[i,j]=dof[i,j]+cov(y_pred[,l],y[,l])
    }
  }
}
dof
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 123.84346 123.95381 127.02452 128.32683
## [2,]  83.08434  99.98397 107.21929 107.94043
## [3,]  46.16436  61.10353  64.83338  68.18175
## [4,]  32.88567  40.34110  41.51469  47.12007
```

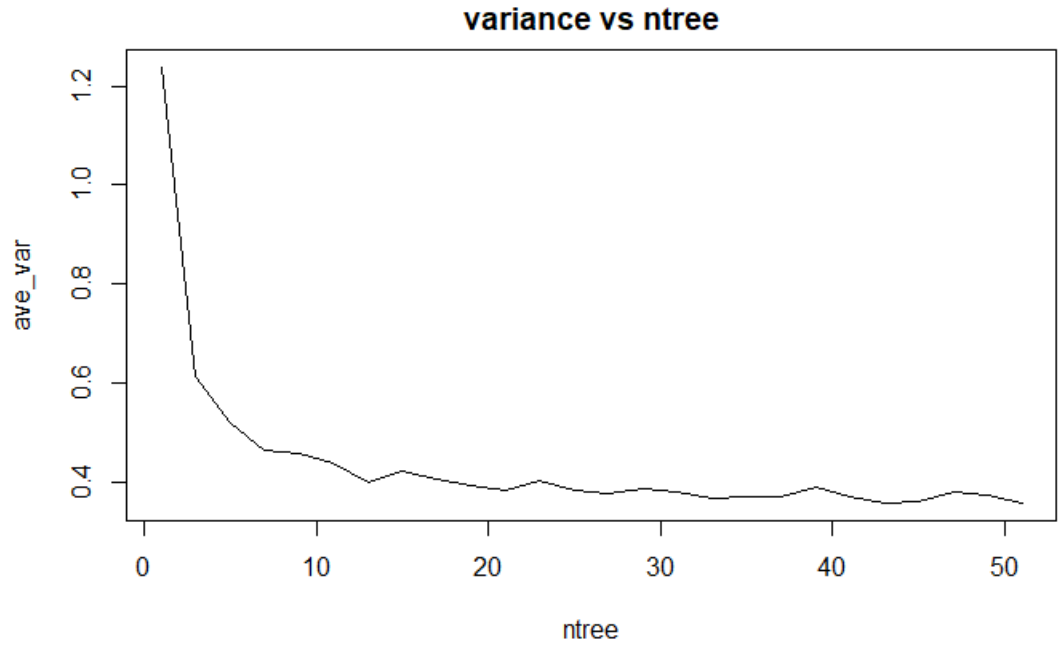
In the degree freedom matrix, nodesize increase by rows and mtry increase by columns. When mtry increases, the degree of freedom increase. When nodesize increase, the degree of freedom will decrease and it has larger effect on dof compared with mrty.

(b)

```
#generate data
ntree=seq(1,51,2)
ave_var=rep(0,length(ntree))
for (i in 1:length(ntree))
{
  y_pred=matrix(nrow=20,ncol=200)
```

```
  y=t(matrix(rep(mean_Y,20), nrow=20,ncol=200))
  var=0
  for (j in 1:20)
  {

    y[j,]=y[j,]+rnorm(200)
    RF=randomForest(X, y[j,],ntree=ntree[i])
    y_pred[j,]=predict(RF,X)
  }
  for (k in 1:200)
  {
    var=var+mean((y_pred[,k]-mean(y_pred[,k]))^2)
  }
  ave_var[i]=var/n
}
plot(ntree,ave_var,main="variance vs ntree",type="l")
```



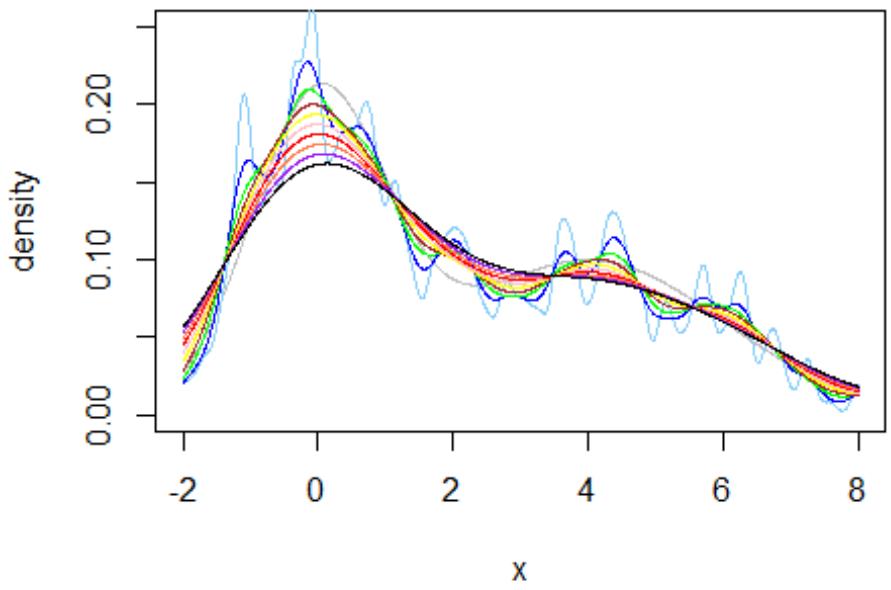
When the number of tree increase, variance will decrease. Since random forest make the result stable by get the average results from multiple trees.

Question 4

(a)

```
x <- c(rnorm(5000), rnorm(5000, 4, 2))
grid = seq(-2, 8, 0.01)
lambda <- seq(0.1,1,0.1)
sample_size=500
sample_x=sample(x,sample_size)
```

```
density = matrix(NA, sample_size, length(grid))
color_list=c('lightskyblue','blue','green','brown','yellow','pink','red
','coral','purple','black')
counter=1
plot(grid, 0.5*dnorm(grid) + 0.5*dnorm(grid,4,2), type = "l",col = "gre
y", xlab = "x", ylab = "density",ylim=c(0,0.25))
for (i in 1:length(lambda))
{
  for (j in 1:sample_size)
  {
    density[j,]=dnorm((grid - sample_x[j])/lambda[i])/length(sample_x)/
lambda[i]
  }
  lines(grid, colSums(density), type = "l", col = color_list[counter])
  counter=counter+1
}
```

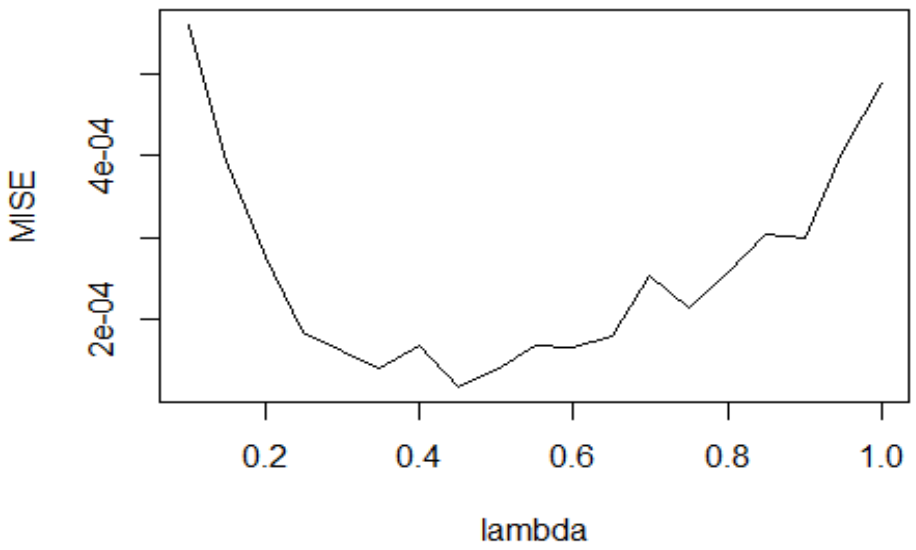


I tune the lambda in seq(0.1,1,0.1), which represented by color ('lightskyblue','blue','green','brown','yellow','pink','red','coral','purple','black'). At the boundary,when lambda increase: the estimation curve become smoother, which means variance decreae; however, the estimation curve deviate more away from real density curve, which means the bias increase.

(b)

```
x <- c(rnorm(5000), rnorm(5000, 4, 2))
grid = seq(-2, 8, 0.01)
```

```
lambda <- seq(0.1,1,0.05)
sample_size=500
density = matrix(NA, sample_size, length(grid))
MISE=rep(0,length(lambda))
f=(0.5*dnorm(grid) + 0.5*dnorm(grid,4,2))
for (i in 1:length(lambda))
{
  Ef_err=0
  for (k in 1:10)
  {
    sample_x=sample(x,sample_size)
    for (j in 1:sample_size)
    {
      density[j,]=dnorm((grid - sample_x[j])/lambda[i])/length(sample_x
)/lambda[i]
    }
    f_hat=colSums(density)
    Ef_err=Ef_err+sum(((f_hat - f)^2))
  }
  Ef_err=Ef_err/10
  MISE[i]=1/length(grid)*Ef_err
}
plot(lambda, MISE, type = "l",xlab = "lambda", ylab = "MISE")
```



MISE is the combination of bias and variance. When bandwidth increase, MISE first decreae then increase. When lambda is around 0.4, it got the lowest MISE and the best performance. This is corresponding the bias-variance trade off I mentioned in part (a).