Assignment 4

Question 1.

Ans To prove :

$$R\left(f(x), \hat{f}_n(x)\right) = bias^2 + Var$$

Risk function is the expected value of loss function.

$$E\left[(f(x) - \hat{f}_n(x))^2\right] = E\left[(f^2(x) - 2\hat{f}_n(x)f(x) + \hat{f}_n^2(x)\right]$$

$$LHS = f^2(x) - 2E\left(\hat{f}_n(x)f(x)\right) + E\left(\hat{f}_n^2(x)\right)$$

Where expectations are taken with $\hat{f}_n(x)$ random variable.

$f(x)$ is not a random variable

∴ It is equal to its own expectation with respect to any distribution.

$$RHS = Bias^2 + Var$$

$$Bias^2 = \left(E(\hat{f}_n(x)) - f(x)\right)^2$$

$$= \left[E\left(\hat{f}_n(x)\right)\right]^2 - 2E\left(\hat{f}_n(x)\right)f(x) + (f(x))^2$$

$$Var\left(\hat{f}_n(x)\right) = E\left(\hat{f}_n(x)^2\right) - \left(E\left(\hat{f}_n(x)\right)\right)^2$$

$$RHS = E\left(\hat{f}_n(x)^2\right) - 2E\left(\hat{f}_n(x)\right)f(x) + (f(x))^2$$

$$= E\left[(f(x) - \hat{f}_n(x))^2\right]$$

$$= LHS$$

$$LHS = RHS.$$

Ans

Question 2

$$R(f, \hat{f}) = E(D(f, \hat{f}))$$

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}}, \qquad \hat{\theta} = \bar{x}$$

$$\therefore \hat{f}(x) = \frac{1}{\sqrt{2\pi}} e^{-(x-\bar{x})^2/2}$$

$$E\left(\int_{-\infty}^{\infty} f(x) \log \frac{f(x)}{\hat{f}(x)} \, dx\right)$$

$$= E\left(\int_{-\infty}^{\infty} f(x) \log \frac{e^{-(x-\theta)^2/2}}{e^{-(x-\bar{x})^2/2}} \, dx\right)$$

$$= E\left(\int_{-\infty}^{\infty} f(x) \frac{(x-\bar{x})^2 - (x-\theta)^2}{2} \, dx\right)$$

$$= E_f\left[\frac{1}{2}(\bar{x} + 2(\theta - \bar{x})x - \theta^2\right]$$

$$= \frac{1}{2}\left[\bar{x}^2 + 2(\theta - \bar{x}) E_f(x) - \theta^2\right]$$

$$= \frac{1}{2}\left(\bar{x}^2 + 2(\theta - \bar{x})\theta - \theta^2\right)$$

$$= \frac{1}{2}\left(\bar{x}^2 + 2\bar{x}\theta + \theta^2\right) = \left(\frac{(\bar{x} - \theta)^2}{2}\right)$$

Assignment 4

Name: Kalpita Dapkekar

Question 3

data(CMB)

x<-CMB$ell[CMB$ell<400]

y<-CMB$Cl[CMB$ell<400]

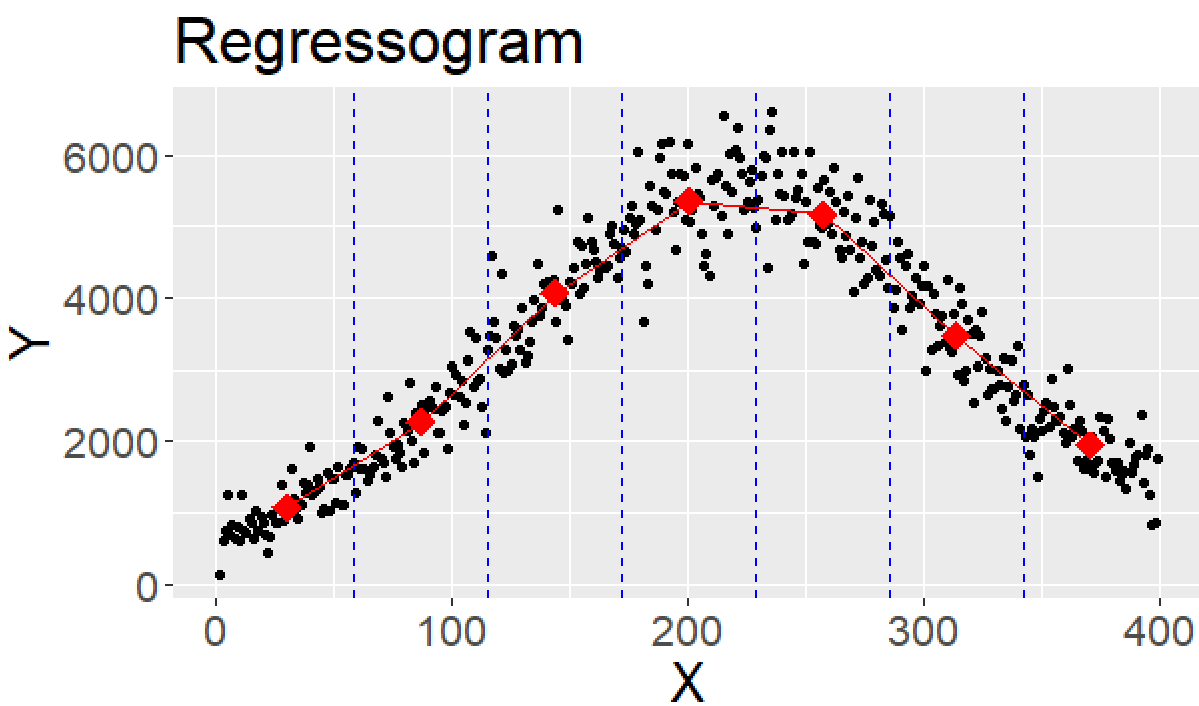
regressogram

library(HoRM)

regressogram(x, y, nbins = 7, show.bins = TRUE,

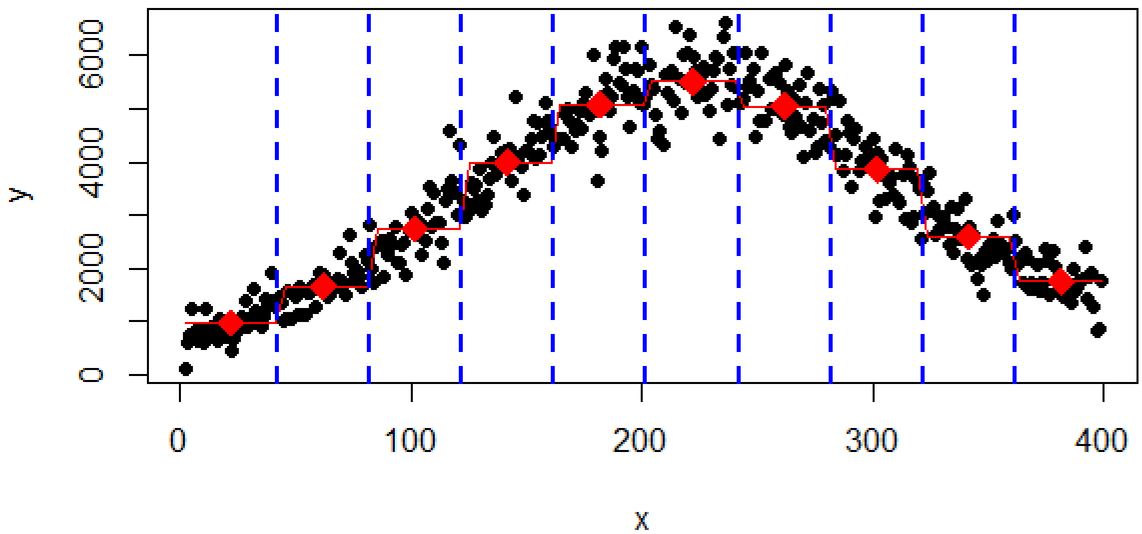
show.means = TRUE, show.lines = TRUE,

x.lab = "X", y.lab = "Y", main = "Regressogram")

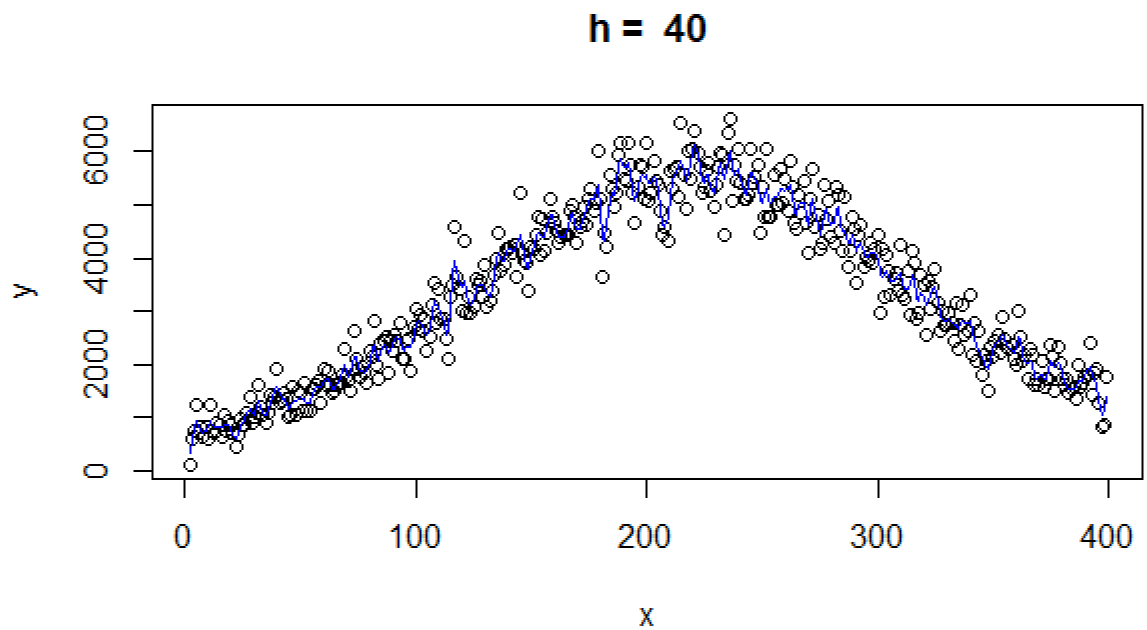


myregressogramplot<-function(x,y,h='NULL'){

```
  L<-nbin<-(max(x)-min(x))
  if(!is.numeric(h)) {nbin=10; h<-L/10}
  xx<-seq(min(x),max(x),by=round(L/100,4))
  yy<-apply(as.matrix(xx),1,function(u) myregressogram(u,x,y,h))
  plot(x,y,pch=16)
  lines(xx,yy,type='l',col='red')
  nbin<-round(L/h,0)
  if(min(x)+nbin*h<max(x)) nbin<-nbin+1
  for(k in 1:(nbin-1)){
    abline(v=min(x)+k*h,col="blue",lty=2, lwd=2)
    u<-min(x)+(k-.5)*h
    points(u,myregressogram(u,x,y,h),pch=18,col='red',cex=2)
  }
  u<-min(x)+(k+.5)*h
  points(u,myregressogram(u,x,y,h),pch=18,col='red',cex=2)
}
myregressogramplot(x,y,h=(400/10))
```

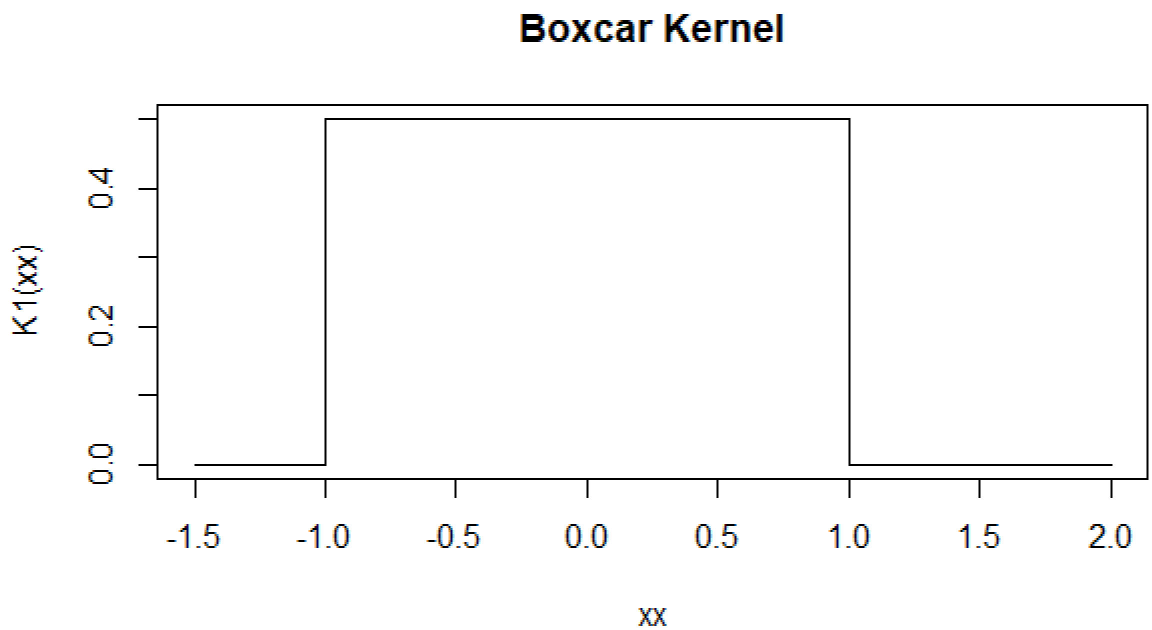



```
h=400/10
mKxy<-function(u,K,h){
  out<- sum(K((x-u)/h)*y)/sum(K((x-u)/h))
  return(out)
}
ky<-apply(as.matrix(x),1,function(x) mKxy(x,dnorm,h=1))
plot(x,y,main=paste("h = ",40))
lines(x,ky,col="blue")
```



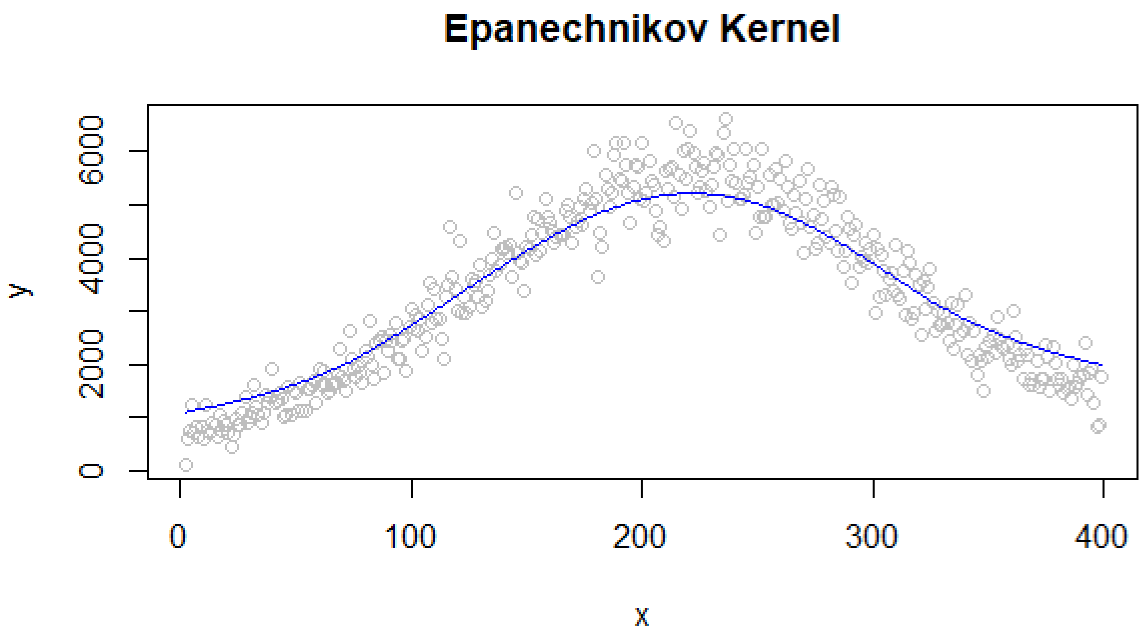
###bin smoother (boxcar)

```
Ix<-function(x){
  n<-length(x)
  temp=rep(0,n)
  temp[abs(x) <=1]<-1
  return(temp)
}
K1<-function(x){Ix(x)/2}
xx<-seq(-1.5,2,by=0.001)
par(mfrow=c(1,1))
plot(xx,K1(xx),main='Boxcar Kernel',type='l')
```



kernel (a kernel other than boxcar),

```
h=floor(.20*398)

mKxy<-function(u,K){

  out<- sum(K((x-u)/h)*y)/sum(K((x-u)/h))

  return(out)

}

K<-function(u){3/4*(1-u^2)*(abs(u)<1)}

xx<-seq(2,399,by=1)

yy<-apply(as.matrix(xx),1,function(x) mKxy(x,K))

plot(x,y,col='grey',main = 'Epanechnikov Kernel')

lines(xx,yy,col='blue')
```

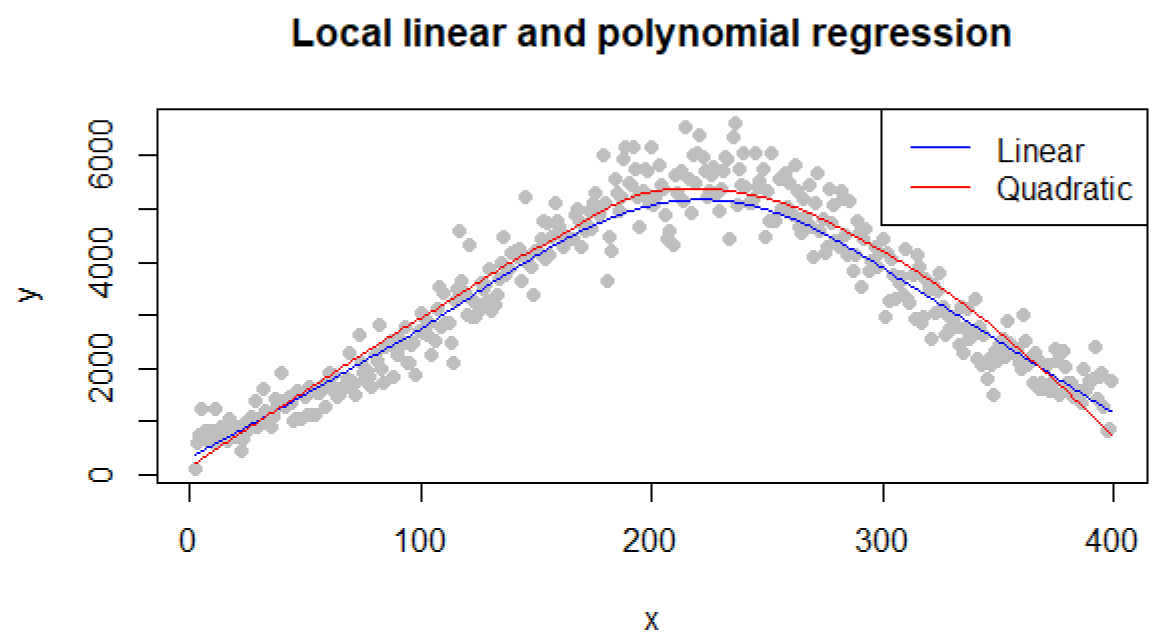


###local linear regression, loess deg=1

```
plot(x,y,pch=19,main = 'Local linear and polynomial regression',col="gray")
fit <- loess(y~x,degree=1,span=.5)
lines(xx,predict(fit,newdata=xx),col="blue")
```

###local quadratic polynomial regression. loess deg =2

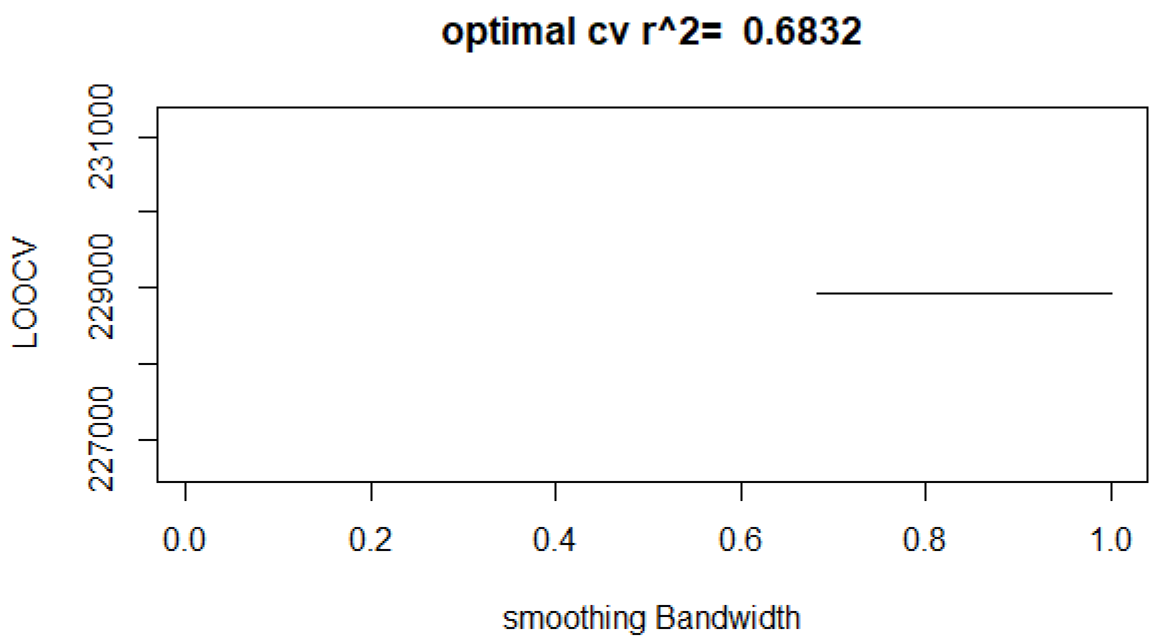
```
fit <- loess(y~x,degree=2,span=.8)
lines(xx,predict(fit,newdata=xx),col="red")
legend("topright",lty=1,c("Linear","Quadratic"),col=c("blue","red"))
```



cross-validation to choose the amount of smoothing

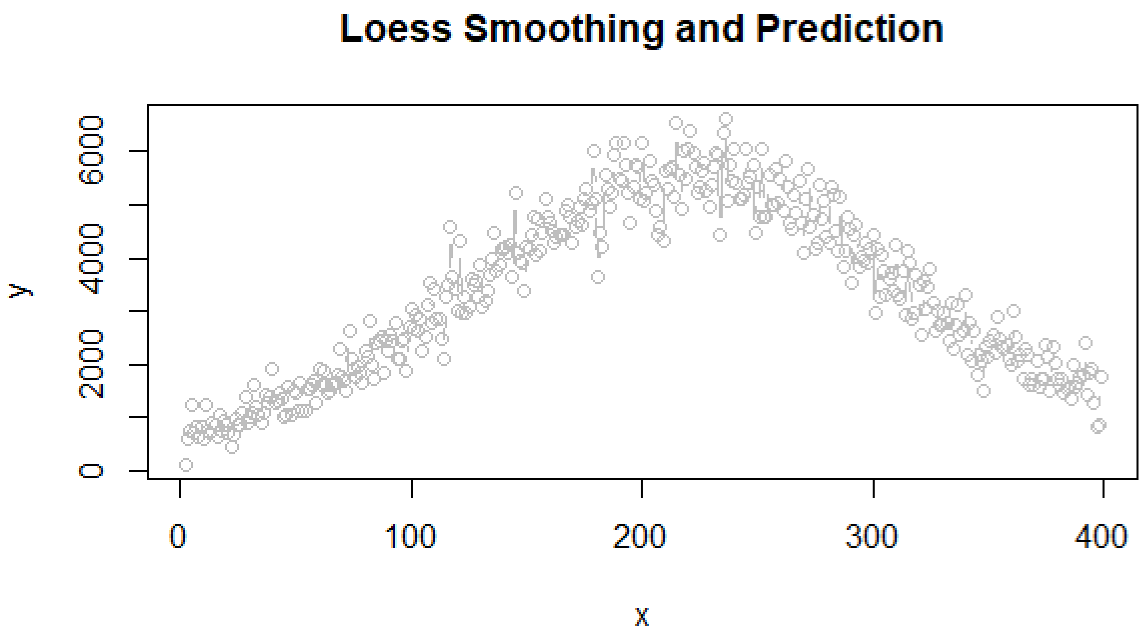
```
bw <- seq(.01,1,len = 51)
r <- matrix(NA,nrow=length(bw),ncol=length(y))   #nrow(m)
for (i in 1:length(bw))
{
  for (j in 1:length(y))  #nrow(y)
  {
    yhat <- ksmooth(x[-j],y[-j],kernel="normal",bandwidth=bw[i],x.points=x[j])$y
    r[i,j] <- y[j] - yhat
    r[i,j]
  }
}
CV <- apply(r^2,1,mean)
```

```
plot(bw, apply(r^2,1,mean),type="l",ylab="LOOCV",xlab="smoothing
Bandwidth",main=paste("optimal cv r^2= ",bw[which.min(CV)]))
```



###Loess Smoothing and Prediction

```
loessMod10 <- loess(x ~ y,span=bw[which.min(CV)])

smoothed10 <- predict(loessMod10)

plot(x,y,type="b", main="Loess Smoothing and Prediction",col = 'gray')
```


Loess Smoothing and Prediction
y
0
2000
4000
6000
x
0
100
200
300
400