Local Smoothing

Split the data

```
df <- read.csv("utilities.csv")
y <- df$gasbill/df$billingdays
x <- df$temp
n<-length(y)
train_idx <- sample(1:n,0.8*n)
y_train <- y[train_idx]
x_train <- x[train_idx]
y_test <- y[-train_idx]
x_test <- x[-train_idx]
```

```
normal_kernel <- function(d,h){
  ##Input: d- distance (scalar), h-bandwith (scalar)
  ##Output: kernel value (scalar)
  return(1/h*dnorm(d/h,mean=0,sd=1))
}

get_W <- function(x,x_star,h,kern_func){
  W <- diag(kern_func(x-x_star,h))
  return(W/sum(W))
}

get_R <- function(x,x_star,D){
  R <- matrix(NA,ncol=D+1,nrow=length(x))
  for(i in 1:(D+1)){
    R[,i] <- (x-x_star)^(i-1)
  }
  return(R)
}

get_l <- function(R,W){
  return((solve(t(R)%*%W%*%R)%*%t(R)%*%W)[1,])
}
get_est <- function(l,y){
  return(l%*%y)
}
```

Find h that minimizes test error

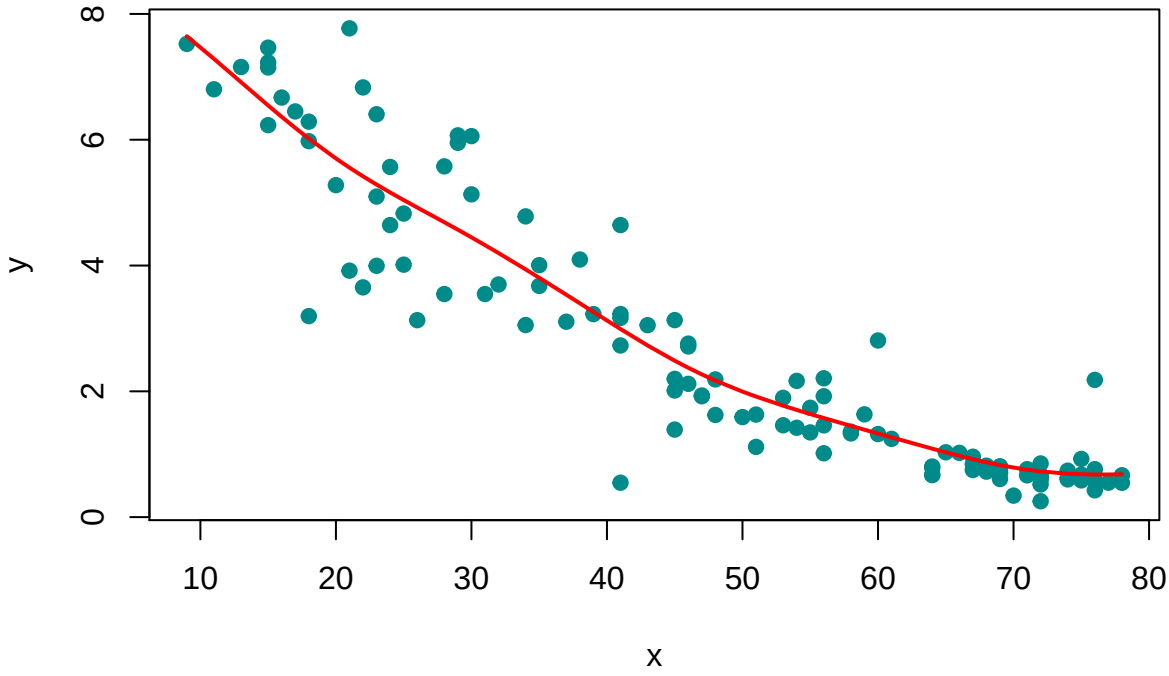
```
get_err <-function(D,h){
  err <- rep(NA,n)
  for(i in 1:n){
    x_star <- x[i]
    x_train <- x[-i]
    y_train <- y[-i]
    y_test <- y[i]
    R <- get_R(x_train,x_star,D)
    W <- get_W(x_train,x_star,h,normal_kernel)
    l <- get_l(R,W)
```

```
    y_pred <- get_est(l,y_train)
    err[i] <- (y_test-y_pred)^2
  }
  return(err)
}

h_vec <- seq(1,10,1)
h_err <- rep(NA,length(h_vec))
for(i in 1:length(h_vec)){
  h_err[i] <-mean(get_err(1,h_vec[i]))
}
h_min <-h_vec[which.min(h_err)]
h_min
```

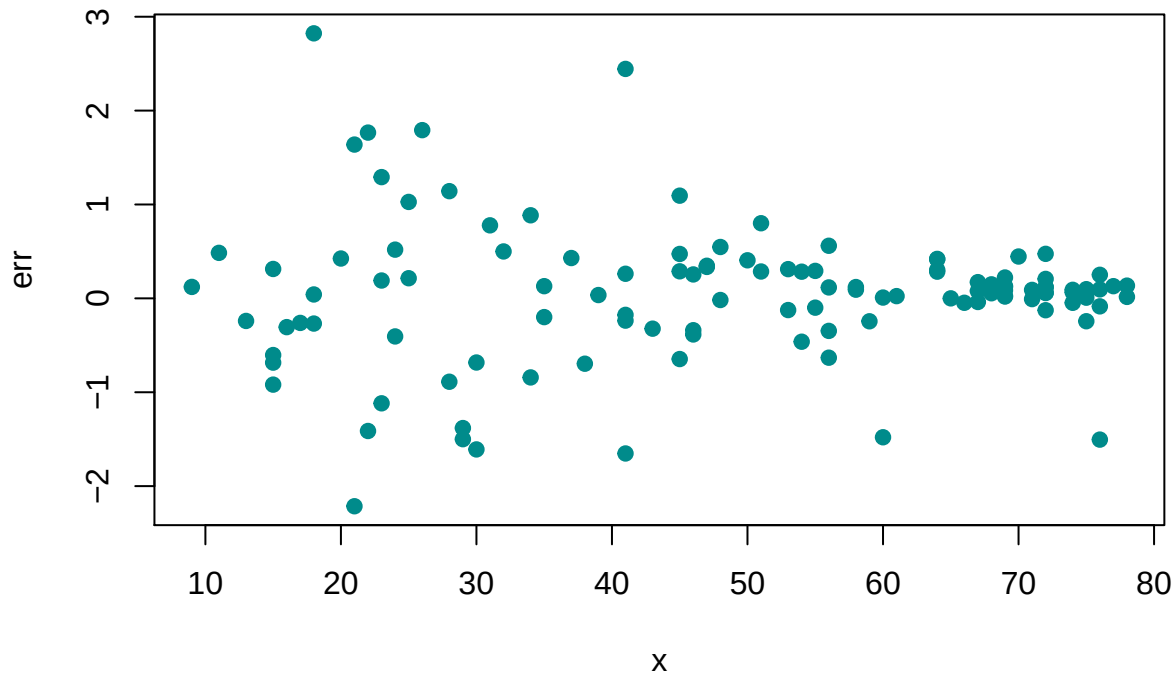
```
## [1] 7
```

Heteroscedasticity

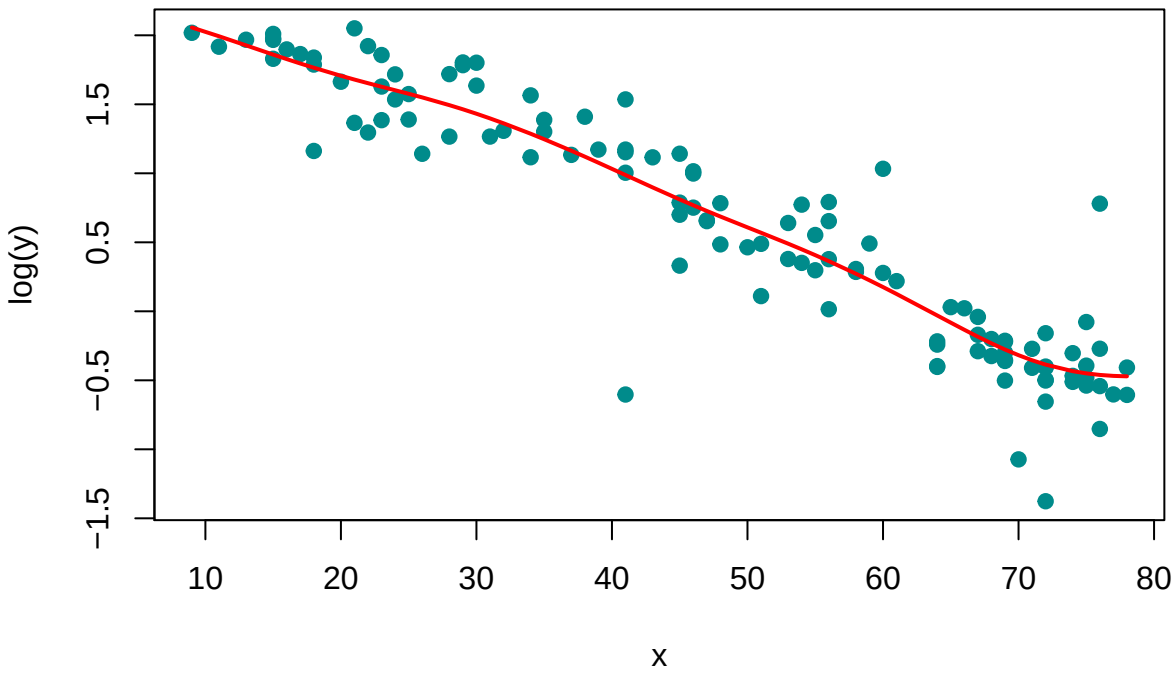
```
get_pred<-function(x_train,x_star,y,D,h,kern_func){

      R <- get_R(x_train,x_star,D)
      W <- get_W(x_train,x_star,h,kern_func)
      l <- get_l(R,W)
      y_pred <- get_est(l,y)

  return(list(y_pred,l))
}
y_fit <- rep(NA,length(x))
l <- rep(NA,length(x))
err <- rep(NA,length(x))
D<-1
H <- matrix(NA,nrow= length(x),ncol=length(x))
for(i in 1:length(x)){
  ans <- get_pred(x,x[i],y,D,h_min,normal_kernel)
  y_fit[i] <- ans[[1]]
  H[i,] <- ans[[2]]
  err[i] <- y_fit[i] - y[i]

}
x_idx <- order(x)
plot(x,y,col="cyan4",pch=19)
lines(x[x_idx],y_fit[x_idx],lwd=2,col="red")
```



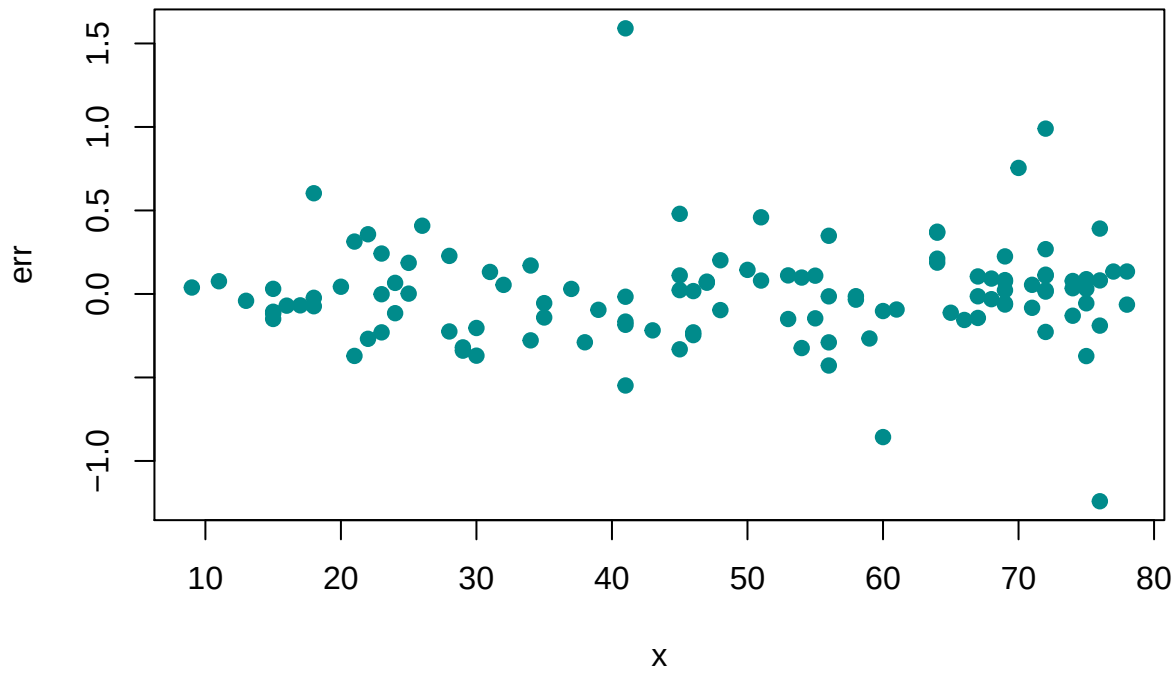

```
plot(x,err,pch=19,col="cyan4")
```




```
y_fit <- rep(NA,length(x))
l <- rep(NA,length(x))
err <- rep(NA,length(x))
D<-1
H <- matrix(NA,nrow= length(x),ncol=length(x))
for(i in 1:length(x)){
  ans <- get_pred(x,x[i],log(y),D,h_min,normal_kernel)
  y_fit[i] <- ans[[1]]
  H[i,] <- ans[[2]]
  err[i] <- y_fit[i] - log(y[i])

}
x_idx <- order(x)
plot(x,log(y),col="cyan4",pch=19)
lines(x[x_idx],y_fit[x_idx],lwd=2,col="red")
```

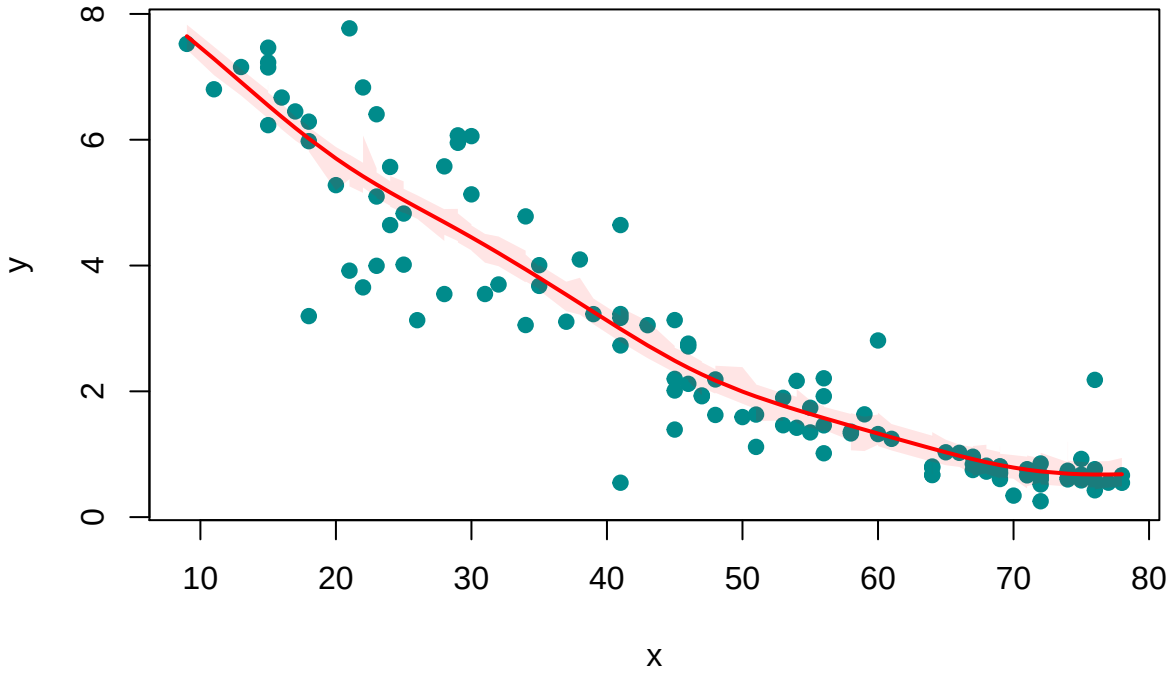



```
plot(x,err,col="cyan4",pch=19)
```



Confidence Band

```
y_fit <- rep(NA,length(x))
l <- rep(NA,length(x))
err <- rep(NA,length(x))
D<-1
H <- matrix(NA,nrow= length(x),ncol=length(x))
for(i in 1:length(x)){
  ans <- get_pred(x,x[i],y,D,h_min,normal_kernel)
  y_fit[i] <- ans[[1]]
  H[i,] <- ans[[2]]
  err[i] <- y_fit[i] - y[i]

}
sigma_2 <- sum(err^2)/(length(x)-2*sum(diag(H))+sum(diag(t(H)%*%H)))
sd_vec <- rep(NA,length(x))
for(i in 1:dim(H)[1]){
  l <- H[i,]
  sd_vec[i] <- sqrt(sigma_2*t(l)%*%l)
}

x_idx <- order(x)
plot(x,y,col="cyan4",pch=19)
lines(x[x_idx],y_fit[x_idx],lwd=2,col="red")
polygon(c(x[x_idx],rev(x[x_idx])), c(y_fit[x_idx]-sd_vec*qnorm(0.95,0,1), rev(y_fit[x_idx])+sd_vec*qnor
```


y
0
2
4
6
8
10
20
30
40
50
60
70
80
x