

HW3

Keyi Zhong

5/7/2020

1.1

```
source("test-data.R")
library("leaps")
truncated.power.design.matrix <- function(x) {
  result <- matrix(0,nrow=length(x),ncol=length(x))
  for (i in 1:length(x)) {
    for (j in 1:length(x)) {
      result[i,j] <- max(0,x[i]-x[j])
    }
  }
  for (j in 1:length(x)) {
    result[j,length(x)] <- 1
  }
  return(result)
}
```

1.2

```
regsubsets.fitted.values <- function(X, regsubsets.out, nterm) {
  out.summary <- summary(regsubsets.out)
  vorder <- regsubsets.out$vorder[1:nterm][order(regsubsets.out$vorder[1:nterm])]
  coeff <- coef(regsubsets.out,nterm)
  coefficient <- rep(0,100)
  for (i in 1:nterm) {
    coefficient[vorder[i]] <- coeff[i]
  }
  return(X%*%coefficient)
}
```

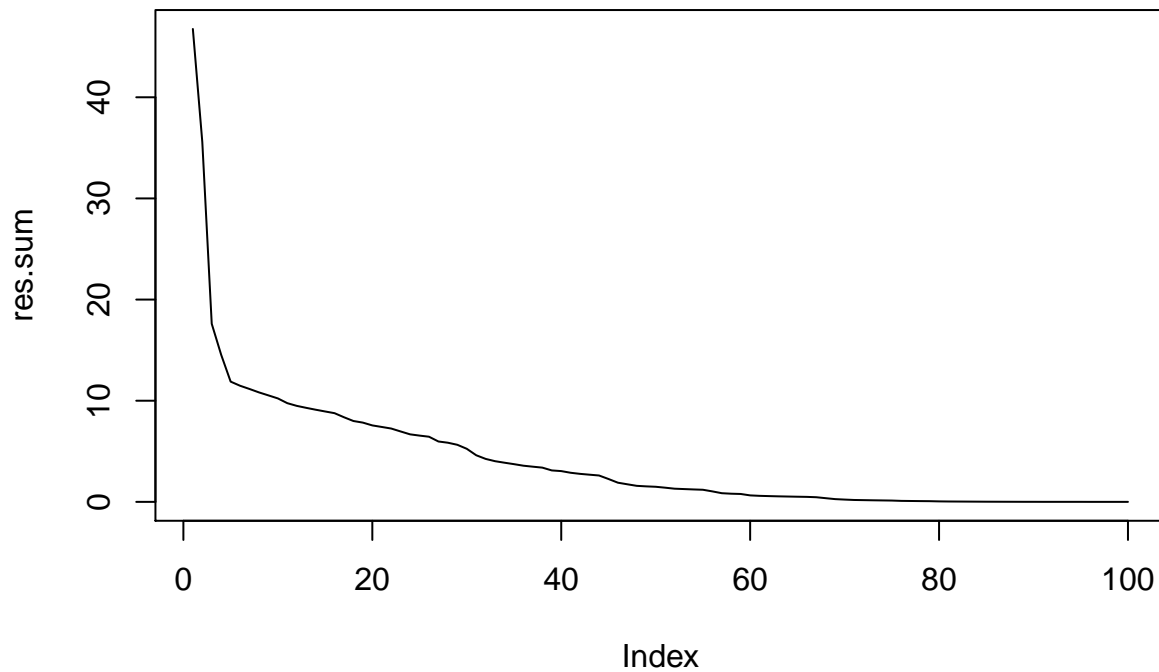
1.3

```
truncated <- truncated.power.design.matrix(x)
regsubsets.out <- regsubsets(truncated,y,method="forward",nvmax=100,intercept = FALSE)
out.summary <- summary(regsubsets.out)
res.sum <- rep(0,100)
for(k in 1:100) {
  fit.values <- regsubsets.fitted.values(truncated,regsubsets.out,k)
```

```

res.sum[k] <- sum((y-fit.values)^2)
}
plot(res.sum,type='l')

```

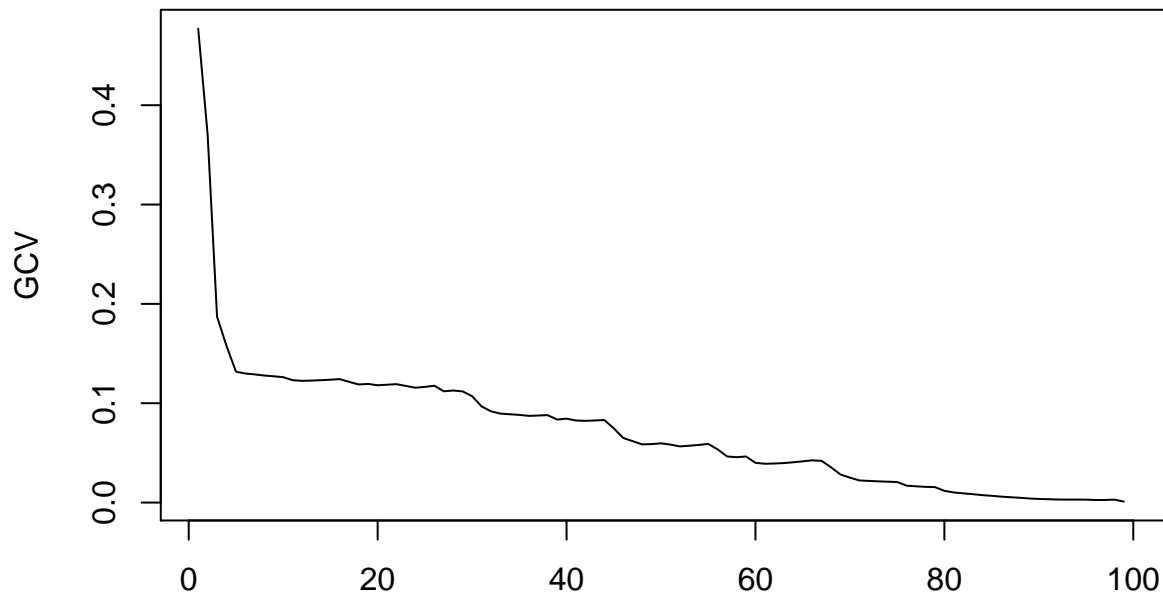


1.4

```

GCV <- rep(0,100)
for(i in 1:100) {
  fit.values <- regsubsets.fitted.values(truncated,regsubsets.out,i)
  GCV[i] <- mean((y-fit.values)^2/(1-i/100)^2)
}
plot(GCV,type='l')

```

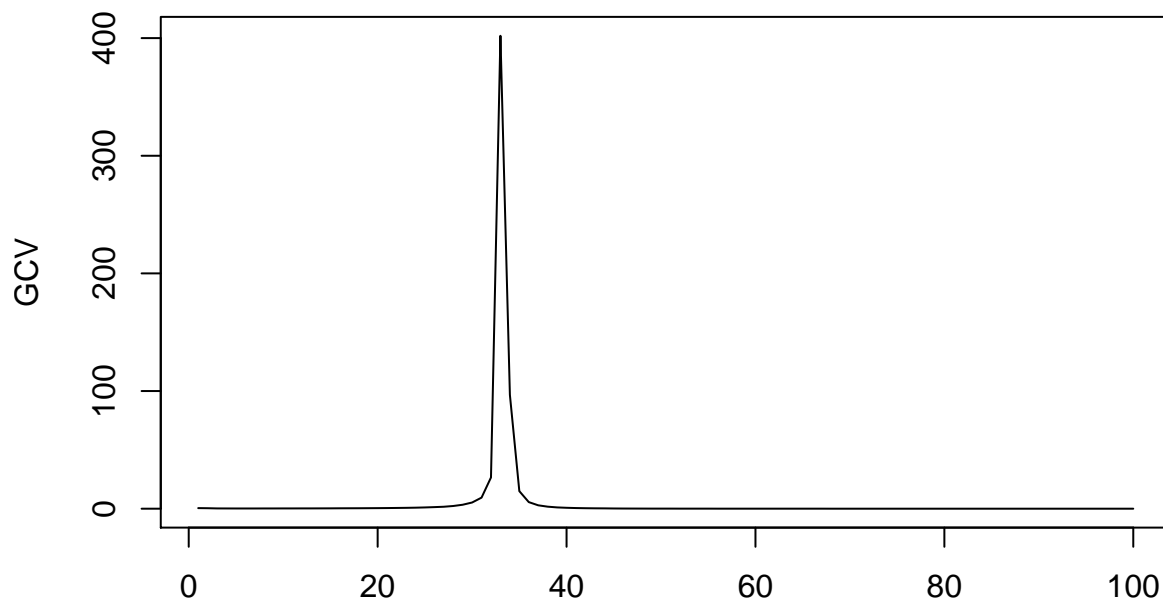


Index

Yes,

because GCV score is high with small k, because GCV is basically residual sum, $1 - \frac{k}{n}$ is decreasing ##1.5

```
GCV <- rep(0,100)
for(i in 1:100) {
  fit.values <- regsubsets.fitted.values(truncated,regsubsets.out,i)
  GCV[i] <- mean((y-fit.values)^2/(1-3*i/100)^2)
}
plot(GCV,type='l')
```



Index

No,

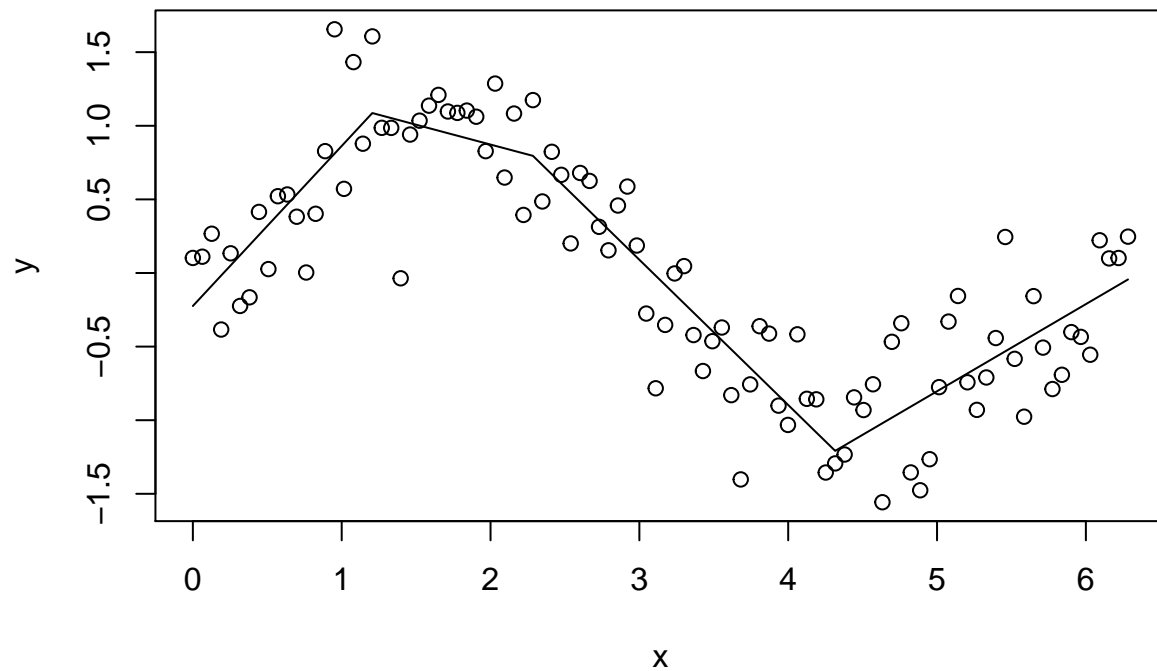
$3 \cdot k$ close to 100 when k is 33 so it gives highest score. With different degree of freedom, it gives different highest score ##1.6

```

k <- which(GCV==min(GCV[1:30]))
fit.values <- regsubsets.fitted.values(truncated,regsubsets.out,k)
plot(x,y,main='forward')
lines(x,fit.values)

```

forward

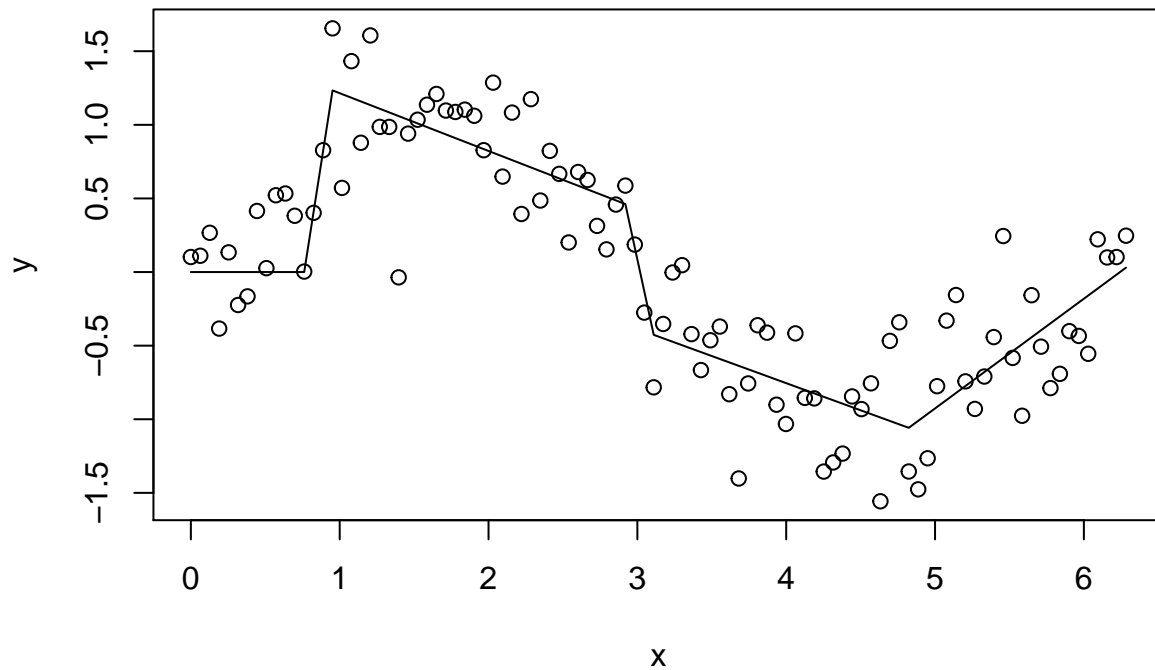


```

regsubsets.out <- regsubsets(truncated,y,method="backward",nvmax=100,intercept = FALSE)
fit.values <- regsubsets.fitted.values(truncated,regsubsets.out,k)
plot(x,y,main='backward')
lines(x,fit.values)

```

backward



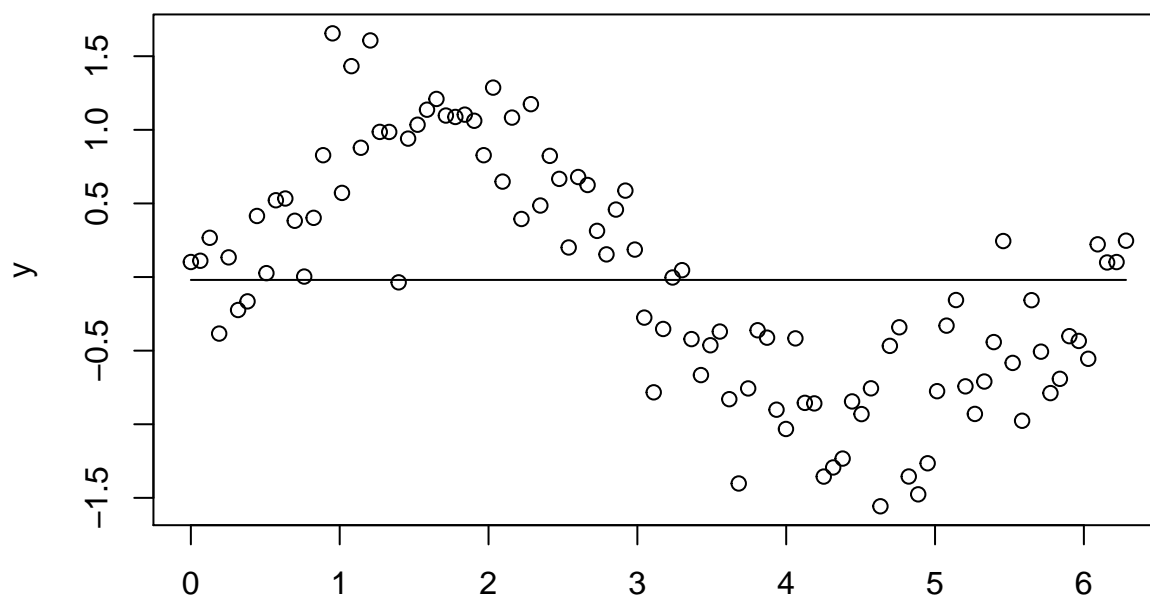
2.2

```
library('glmnet')

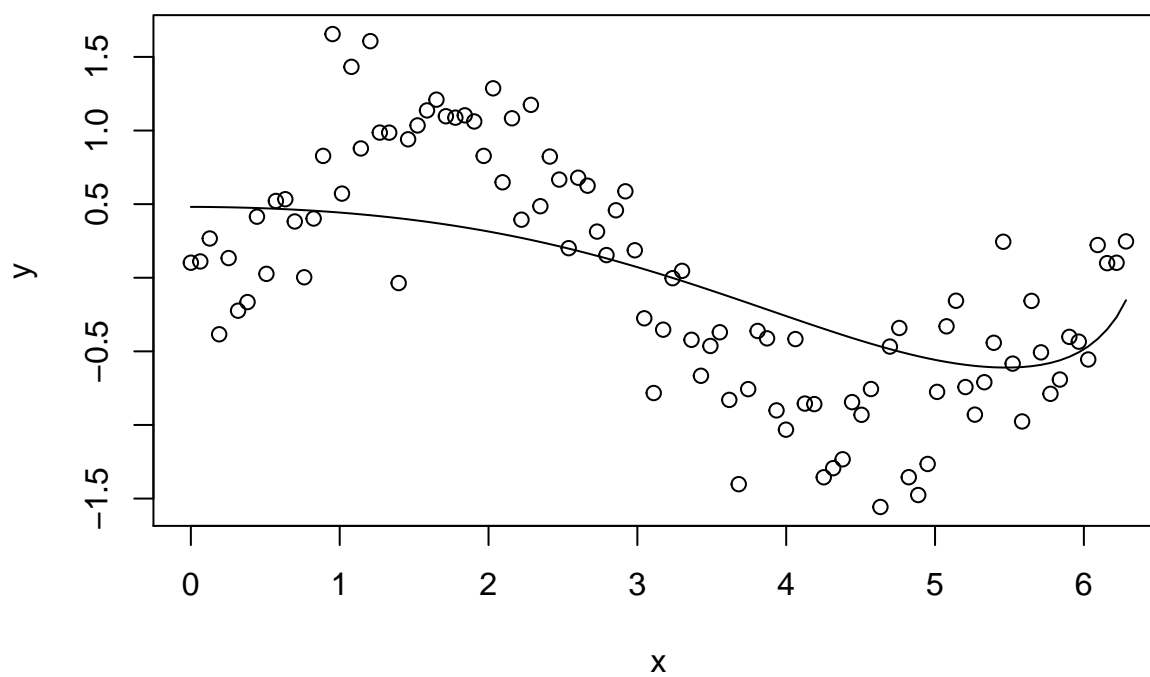
## Loading required package: Matrix
## Loaded glmnet 3.0-2

lambda <- c(0,1,10,1000000)
ridge.mod <- glmnet(truncated,y,alpha=0,lambda = c(0,1,10,1000000))
coeff <- coef(ridge.mod)
coeff[101,]=coeff[1,]
for (i in 1:4) {
  plot(x,y,main = paste0("lambda=",lambda[i]))
  lines(x,truncated%%coeff[,i][2:101])
}
```

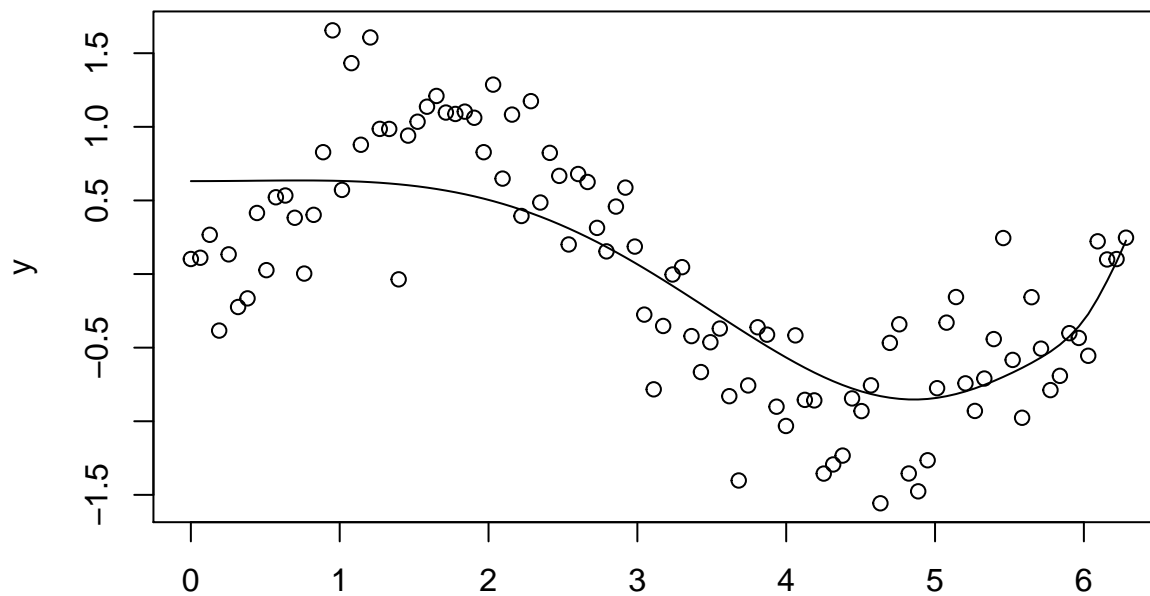
lambda=0



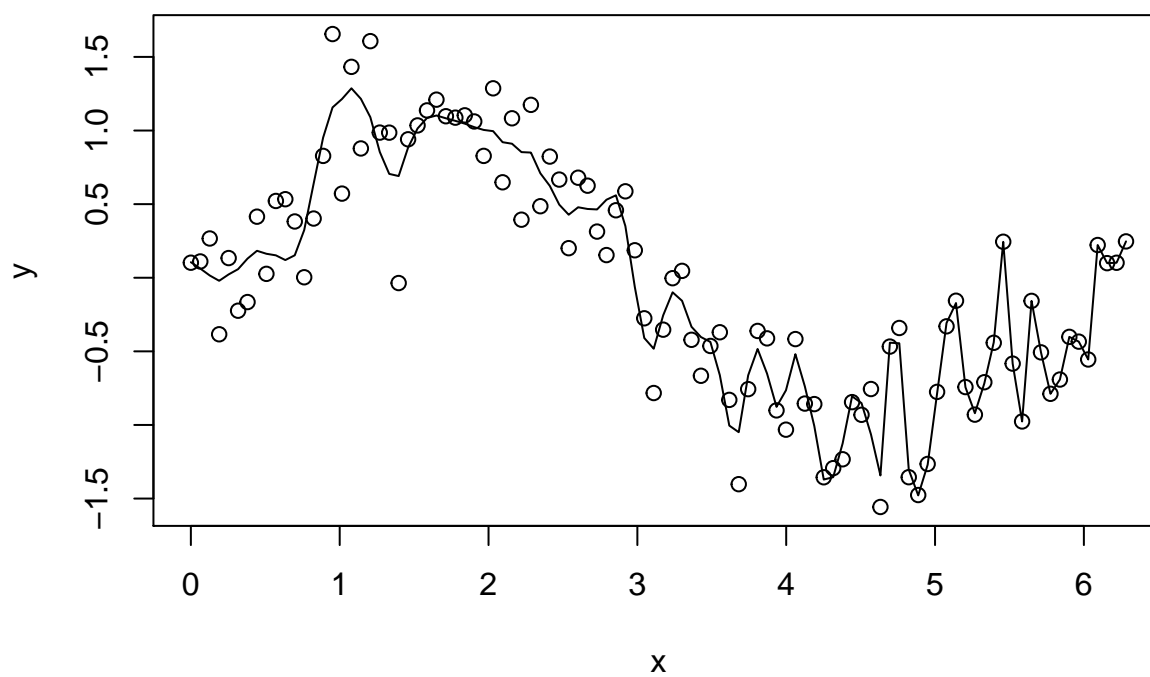
lambda=1



lambda=10

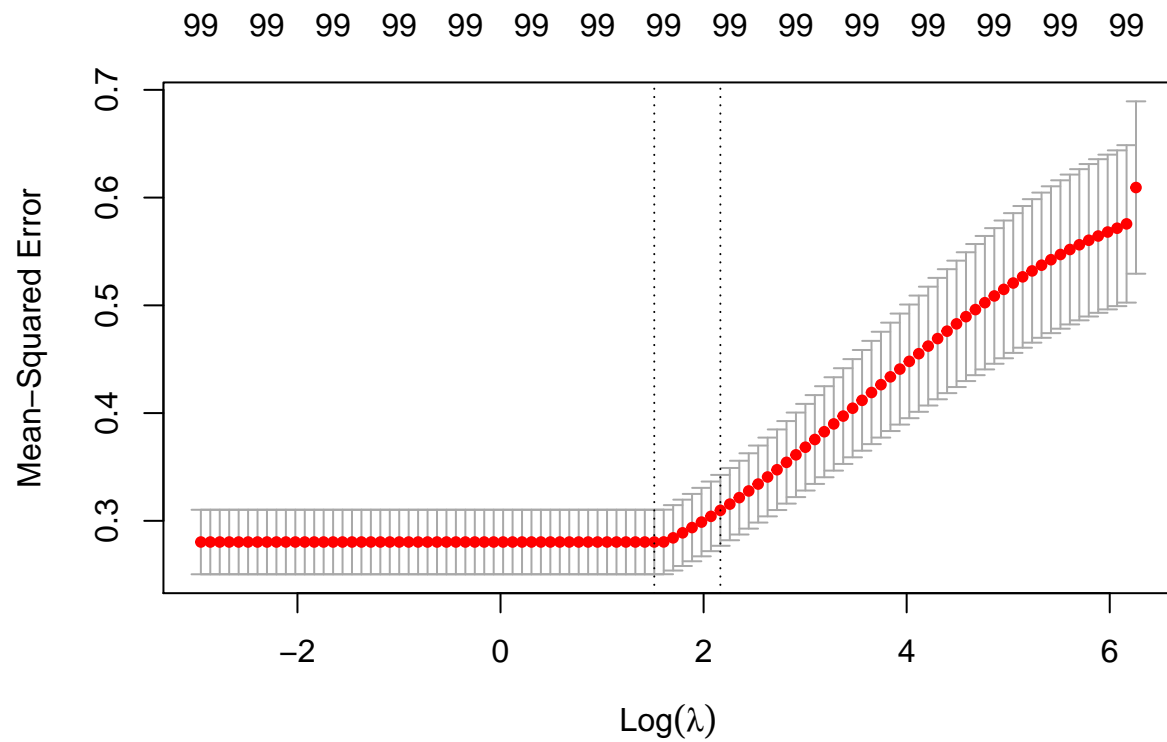


lambda=1e+06



2.3

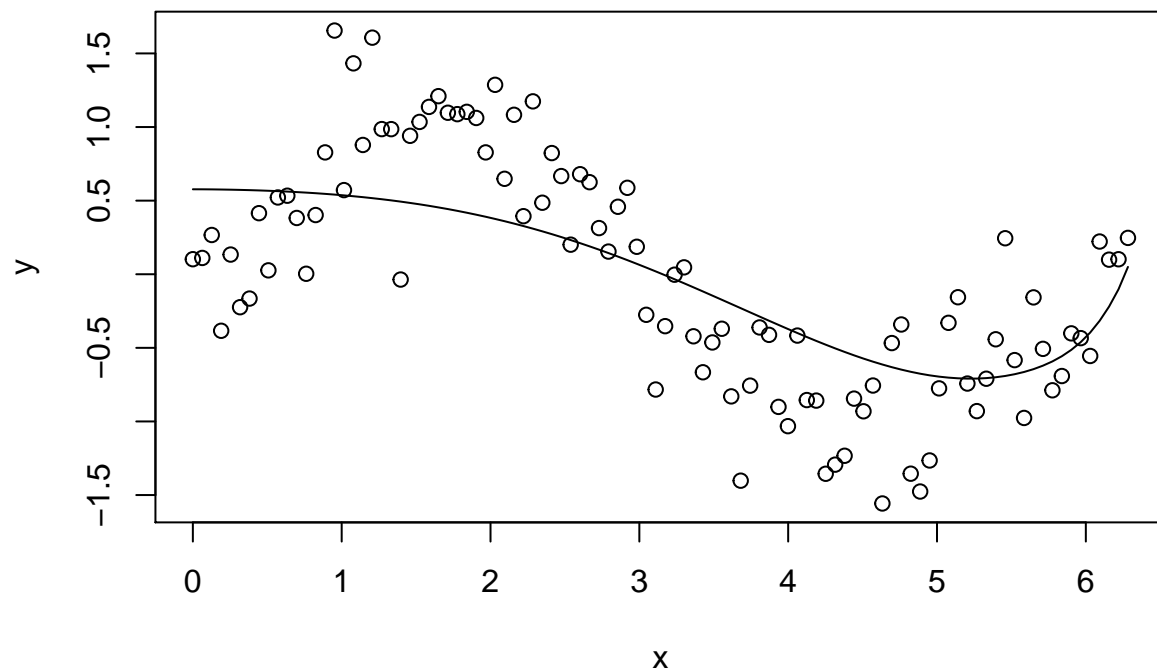
```
cv.out <- cv.glmnet(truncated,y,alpha=0)
plot(cv.out)
```



```

optlam <- cv.out$lambda.min
ridge.mod <- glmnet(truncated,y,alpha=0,lambda = optlam)
coeff <- coef(ridge.mod)
coeff[101,]=coeff[1,]
y_hat <- truncated%*%coeff[2:101]
plot(x,y)
lines(x,y_hat)

```



3

a

Best subset gives the smallest training RSS, because it chooses the best k among all predictors as backward and forward both choose k from top to bottom or vice versa.

b

Also best subset, because we don't have control on test set, in general it gives the best prediction.

c

1

True, because forward choose k from beginning to the end, so a $k+1$ model must include.

2

True, same reason as last question.

3

False, forward and backward choose from different direction.

4

False, same as last question.

5

False, the additional variable could take one of the predictor's place.

4

a

Increase, as λ increase, the model is more off, because of the $-\lambda \sum \beta^2$ part.

b

Decrease then increase, first, increasing λ can make the model close to the true model, but it will eventually off the model.

c

Decrease, it lessen the variance of x_{ij}

d

Increase, it makes the model off the predict model.

e

Not change, irreducible error does not depend on the model