

Individual Assignment 6

Charlie Ling

10/6/2021

For Context, Refer to Problem 8 (parts a, b, c, & d).

```
set.seed(1)
X=rnorm(100)
e=rnorm(100)
Y=0.3+0.5*X+0.2*X^2+0.1*X^3+e
df=data.frame(Y,X)
train = sample(1:100, 100/2) ##default replace = FALSE
test=(-train)
y.test = Y[test]
```

- (e) Now fit a lasso model to the simulated data, again using X , X_2 , . . . , X_{10} as predictors. Use cross-validation to select the optimal value of γ . Create plots of the cross-validation error as a function of γ . Report the resulting coefficient estimates, and discuss the results obtained.

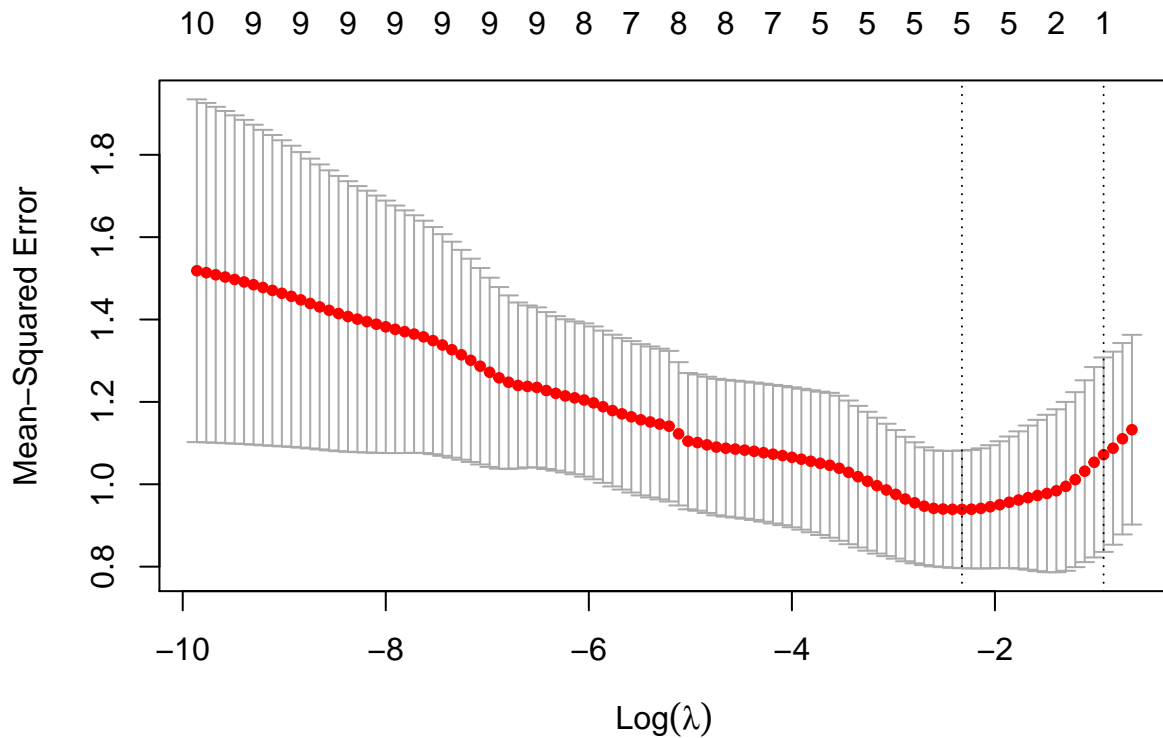
```
grid = 10^seq(10,-2,length=100)
x = model.matrix(Y~poly(X,10),df)
y=Y
library(glmnet)

## Warning: package 'glmnet' was built under R version 4.0.5

## Loading required package: Matrix

## Loaded glmnet 4.1-2

lasso.mod = glmnet(x,y,alpha=1,lambda=grid,thresh = 1e-12)
set.seed(1)
cv.out = cv.glmnet(x[train,],y[train], alpha=1) ##default is 10-fold cross-validation.
plot(cv.out)
```



```
bestlam = cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,x=x[train,],y=y[train],newx=x[test,],exact = T)
mean((lasso.pred-y.test)^2)
```

```
## [1] 1.322043
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef = predict(out,type="coefficients",s=bestlam)[-1,]
lasso.coef[lasso.coef!=0]
```

```
## poly(X, 10)1 poly(X, 10)2 poly(X, 10)3 poly(X, 10)4 poly(X, 10)5
## 5.91488743 0.25130266 0.78365817 0.27932859 0.50242195
## poly(X, 10)10
## -0.01582439
```

```
length(lasso.coef[lasso.coef!=0])
```

```
## [1] 6
```

```
length(lasso.coef[lasso.coef!=0])
```

```
## [1] 6
```

```
#LASSO reduce the original 10 predictors into 6 predictors: X1,X2,X3,X4,X5,X10
```

(f) Now generate a response vector Y according to the model

$Y=B_0+B_7X_7+e$,

and perform best subset selection and the lasso. Discuss the results obtained.

```

Y=0.1+0.7*X^7+e
df=data.frame(Y,X)
y.test = Y[test]

library(leaps)

## Warning: package 'leaps' was built under R version 4.0.5

regfit.full = regsubsets(Y~poly(X,10),data=df)
reg.summary = summary(regfit.full)
plot(reg.summary$adjr2,xlab="Number of Predictors", ylab = "Adjusted RSq", type = "l")
which.max(reg.summary$adjr2)

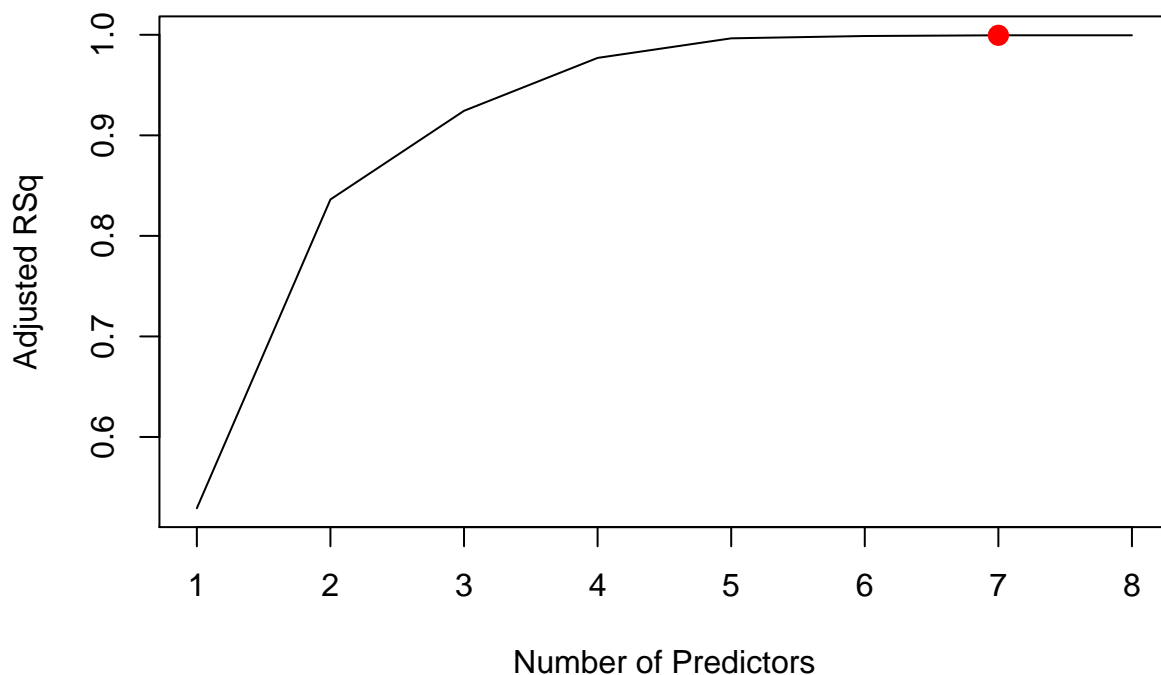
## [1] 7

coef(regfit.full,7)

## (Intercept) poly(X, 10)1 poly(X, 10)2 poly(X, 10)3 poly(X, 10)4 poly(X, 10)5
##      3.016442   234.852169    96.188141   310.430686    58.111708   125.412481
## poly(X, 10)6 poly(X, 10)7
##      11.029615    19.970811

points(7,reg.summary$adjr2[7],col="red",cex=2,pch=20)

```



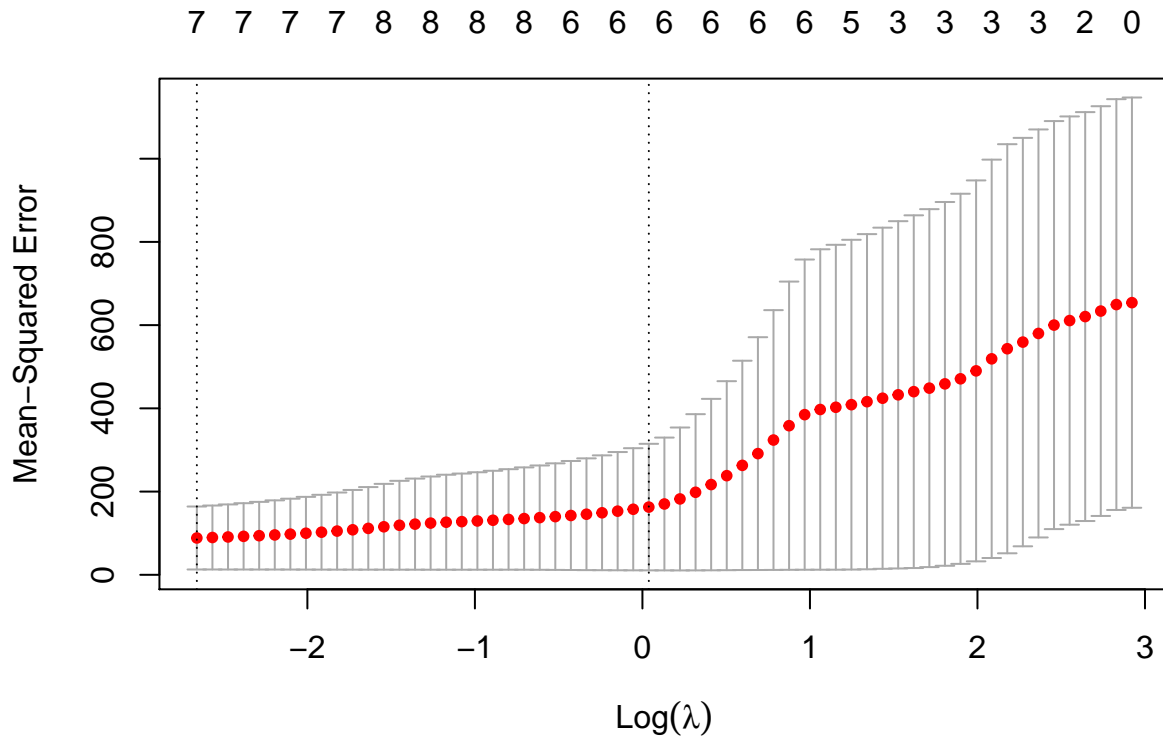
```

#Model 7 is the best model obtained according to adjr2, which has 7 predictors: X1,X2,X3,X4,X5,X6,X7

x = model.matrix(Y~poly(X,10),df)
y=Y

```

```
lasso.mod = glmnet(x,y,alpha=1,lambda=grid,thresh = 1e-12)
set.seed(1)
cv.out = cv.glmnet(x[train,],y[train], alpha=1) ##default is 10-fold cross-validation.
plot(cv.out)
```



```
bestlam = cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,x=x[train,],y=y[train],newx=x[test,],exact = T)
mean((lasso.pred-y.test)^2)
```

```
## [1] 2665.106
```

```
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef = predict(out,type="coefficients",s=bestlam)[-1,]
lasso.coef[lasso.coef!=0]
```

```
## poly(X, 10)1 poly(X, 10)2 poly(X, 10)3 poly(X, 10)4 poly(X, 10)5
## 234.1525849 95.4885569 309.7311016 57.4121236 124.7128965
## poly(X, 10)6 poly(X, 10)7 poly(X, 10)10
## 10.3300308 19.2712268 -0.2516451
```

```
length(lasso.coef[lasso.coef!=0])
```

```
## [1] 8
```

```
#LASSO reduce the original 10 predictors into 8 predictors: X1,X2,X3,X4,X5,X6,X7,X10
```