```r
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.3
```

```r
ourAuto=data.frame("mpg"=Auto$mpg,"cylinders"=factor(cut(Auto$cylinders,2)),
                   "displace"=Auto$displacement,"horsepower"=Auto$horsepower,
                   "weight"=Auto$weight,"acceleration"=Auto$acceleration,
                   "year"=Auto$year,"origin"=as.factor(Auto$origin))
colnames(ourAuto)
```

```
## [1] "mpg"          "cylinders"    "displace"     "horsepower"
## [5] "weight"       "acceleration" "year"         "origin"
```

```r
ntot=dim(ourAuto)[1]
ntot
```

```
## [1] 392
```

```r
set.seed(4268)
testids=sort(sample(1:ntot,ceiling(0.2*ntot),replace=FALSE))
ourAutoTrain=ourAuto[-testids,]
ourAutoTest=ourAuto[testids,]
```

Problem 2 - Shrinkage methods

a) Lasso and ridge regression

- Q11: λ is a parameter which draws the estimated coeffiecients towards zero in both ridge and lasso regression. For very large λ's, all the estimated coefficients are essentially zero. However, in ridge regression, none of the coefficients will be set exact to zero. In comparison , the penalty term in lasso regression may set some of the coefficient to be exactly zero when λ is large enough. In Figure 1, we can see that this is the case, while the coefficients in Figure 2 do not seem to be exactly zero for any value of λ in the given interval. Thus, Figure 1 corresponds to lasso regression while Figure 2 corresponds to ridge regression.
- Q12: As stated, we can see that the tuning parameter λ draws the β's towards zero as λ increases. In lasso regression, all of the coefficients seem to become exactly zero, while in ridge regression, the coefficients are drawn towards zero, but are not exactly zero. When $\lambda = 0$, the penalty term in both ridge and lasso regression will disappear, so the coefficients will just be the same as with least squares estimation. At this value for λ, the bias is zero, but the variance may be high. As λ increases, the variance will become lower as the coefficients are shrinked and flexibility decreases. However, the bias will increase slightly at the same time. In both ridge and lasso regression, when $\lambda \to \infty$, we have the null model, which means that all of the coefficient estimates are zero. Then the variance approaches zero, but the bias becomes large.
- Q13: Ridge regression will include all p covariates, and thus can not be used to perform model selection. However, lasso regression can be used in the same way as best subset selection, as the variables are forced to be exactly zero when λ is large. For a given value of λ, lasso regression may zero out some of the estimated coefficients and it thus performs a variable selection.

b) Finding the optimal λ

- Q14: The function cv.glmnet performs a k-fold cross-validation:

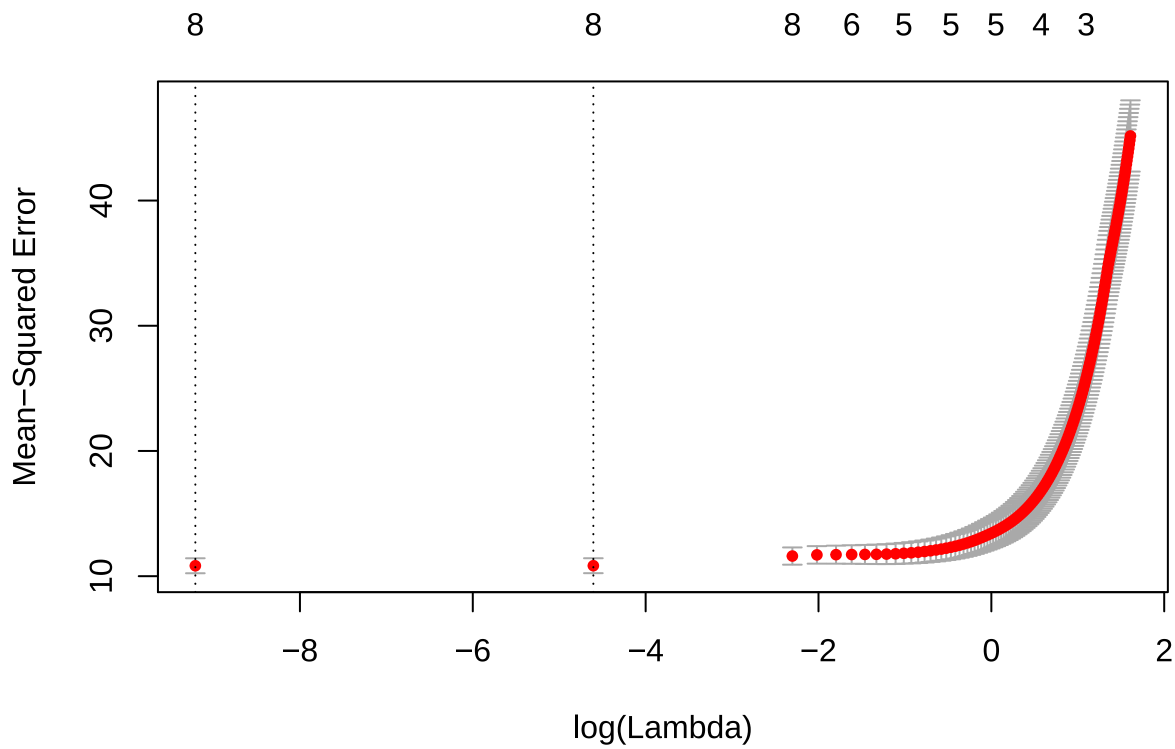
```r
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.3

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.3

## Loaded glmnet 2.0-13
```

```r
set.seed(4268)

x=model.matrix(mpg~.,ourAutoTrain)[,-1] #-1 to remove the intercept.
head(x)
```

```
##   cylinders(5.5,8.01] displace horsepower weight acceleration year origin2
## 1                   1      307        130   3504         12.0   70       0
## 2                   1      350        165   3693         11.5   70       0
## 4                   1      304        150   3433         12.0   70       0
## 5                   1      302        140   3449         10.5   70       0
## 8                   1      440        215   4312          8.5   70       0
## 9                   1      455        225   4425         10.0   70       0
##   origin3
## 1       0
## 2       0
## 4       0
## 5       0
## 8       0
## 9       0
```

```r
y=ourAutoTrain$mpg

lambda=c(seq(from=5,to=0.1,length.out=150),0.01,0.0001) #Create a set of tuning parameters, adding low 
cv.out=cv.glmnet(x,y,alpha=1,nfolds=10,lambda=lambda, standardize=TRUE) #alpha=1 gives lasso, alpha=0 g

plot(cv.out)
```



- Q15: The plot shows the cross-validation curve, including the upper and lower standard deviation curves along the sequence of λ's. The λ with the lowest cross-validated MSE in the plot can be chosen as the optimal λ:

```
cv.out$lambda.min
```

```
## [1] 1e-04
```

- Q16: The `1se-rule`, is another way to choose which λ is optimal. In the object returned by `cv.glmnet`, one of the values is `lambda.1se`, which is the largest value of λ such that the error is within one standard error of the minimum. Here, we get that this lambda is given by

```
cv.out$lambda.1se
```

```
## [1] 0.01
```

c)

- Q17: Using lasso regression with the optimal value of λ according to the `1se-rule`, $\lambda = 0.01$, we can fit the model. The coefficient estimates are given by

```
coef(cv.out,s="lambda.1se")
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                 1
## (Intercept)          -21.407540632
## cylinders(5.5,8.01]   -3.374630096
## displace               0.030954456
```

```
## horsepower          -0.035558382
## weight              -0.006071262
## acceleration         0.122203782
## year                 0.781813677
## origin2              1.675613651
## origin3              2.832358817
```