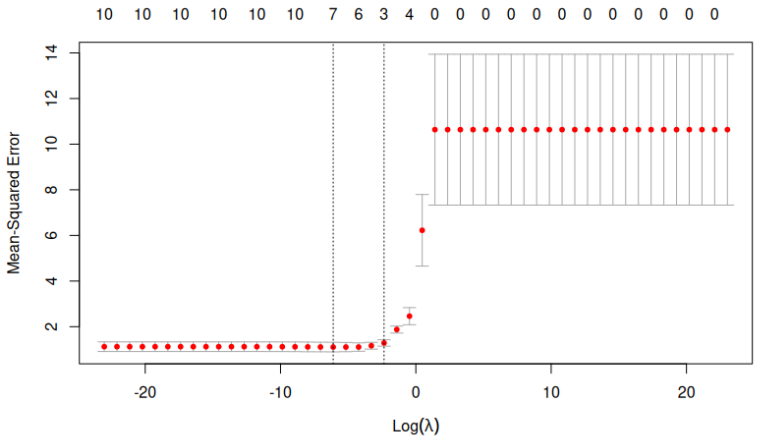
Question 8

e) I did parts a and b, including setting the seed, X, e vectors, and creating a response Y vector. I then followed the book's example to create a model matrix of the x vector with poly(X, 10, raw = TRUE) and created a grid of lambdas (like the book did, but I adjusted them to create smaller lambdas that performed better). I split into training and test at a 4/5 ratio

```
# e)
las_x = model.matrix(Y ~ poly(X, 10, raw = TRUE))
las_y = Y
grid = 10^seq(10, -10, length = 50)
# do training split
train = sample(1:nrow(las_x), nrow(las_x) * 4 / 5)
test = (-train)
y.test = las_y[test]
```

After this I created a lasso plot and plotted it. The best lambda (after extracting it from the model) was .0022. The final lasso coefficients are as follows:




```
12 x 1 sparse Matrix of class "dgCMatrix
                                       s1
(Intercept)                  2.9933080798
(Intercept)                  .
poly(X, 10, raw = TRUE)1     1.6414317942
poly(X, 10, raw = TRUE)2     0.9501646581
poly(X, 10, raw = TRUE)3    -0.5247975760
poly(X, 10, raw = TRUE)4     0.0361152754
poly(X, 10, raw = TRUE)5    -0.1006707505
poly(X, 10, raw = TRUE)6     .
poly(X, 10, raw = TRUE)7     .
poly(X, 10, raw = TRUE)8     .
poly(X, 10, raw = TRUE)9     0.0002690818
poly(X, 10, raw = TRUE)10 -0.0002834468
```

This is interesting, because the intercept coefficient is incredibly close, as well as the X^2 term, but nothing else is really close to the true values.

Question 9

c) I copied a lot of code from Question 8 and made appropriate changes to use the College dataset. Here are my coefficients for Ridge regression, as well as the mean squared test error. Best lambda is 20.565

```
> mean((ridge.pred - test_y)^2)
[1] 961804.2
```

```
> print(ridge.pred)
19 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) -106.39428756
(Intercept)    .
PrivateYes  -504.18561187
Accept         1.60405529
Enroll        -0.99560853
Top10perc     54.50512306
Top25perc    -15.34374263
F.Undergrad    0.07079106
P.Undergrad    0.06230940
Outstate      -0.07759786
Room.Board     0.12526083
Books          0.11525097
Personal       0.01339509
PhD           -7.81887237
Terminal      -4.29150836
S.F.Ratio      4.45005866
perc.alumni   -2.16344284
Expend         0.07531675
Grad.Rate      6.47420947
```

d) Same as c, but used lasso. Test error:

```
> mean((lasso.pred - test_y)^2)
[1] 944898.7
```

Coefficients:

best lambda is 11.498

```
19 x 1 sparse Matrix of class "dgCMatrix"
                      s1
(Intercept) -303.32368155
(Intercept)    .
PrivateYes  -462.19224161
Accept         1.50266950
Enroll        -0.34137543
Top10perc     41.92783440
Top25perc     -6.05571642
F.Undergrad    .
P.Undergrad    0.04552757
Outstate      -0.05524841
Room.Board     0.10222414
Books          0.06537131
Personal       .
PhD           -5.76946439
Terminal      -4.02216122
S.F.Ratio      .
perc.alumni   -2.54163079
Expend         0.06982113
Grad.Rate      3.83488269
```

Supp 1)

a) I adjusted the code to make a 100 * 200 matrix of data

```
# a)
set.seed(2)
variables = matrix(rnorm(100*200),nrow=200, dimnames=list(NULL, paste(1:100)))
```

b, c, d) I made a rnorm of 200 for the response variable, then framed the data, made a linear model, and ran it's summary. There were only 2 predictors that were significant at the .05 level, X9 and X87

```
# b)
y = rnorm(200)
data = data.frame(y, variables)

# c)
lm = lm(y ~ ., data = data)

# d)
print(summary(lm))
# 2 significant
```

e) I then created a separate lm with the 3 most significant predictors, and here are the results of their summary. The X9 variable is again significant, but the other variables are not. This is expected as having many predictors can influence false relationships with the response variable. There is evidence that at least one coefficient is non-zero, X9

```
Call:
lm(formula = y ~ X9 + X74 + X87, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.67446 -0.60908 -0.01428  0.65318  2.89760

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03597    0.06694   0.537  0.59168
X9           0.17106    0.06429   2.661  0.00844 **
X74         -0.03030    0.06619  -0.458  0.64762
X87         -0.12719    0.07008  -1.815  0.07107 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9399 on 196 degrees of freedom
Multiple R-squared:  0.04582,   Adjusted R-squared:  0.03121
F-statistic: 3.137 on 3 and 196 DF,  p-value: 0.02655
```

Supp 2

a) The R^2 for the new model is .9804, while the adjusted R^2 is .0256. The R^2 is the measure of change in Y that is accounted for in X, so the change in all of the X variables together account for 98% of the variability or change in Y.

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9841 on 4 degrees of freedom
Multiple R-squared:  0.9804,    Adjusted R-squared:  0.02555
F-statistic: 1.027 on 195 and 4 DF,  p-value: 0.5771
```

b) The last model has no R^2, and all of the coefficients from 200-300 are blank. This is because there is no unique solution to regular least squares when p > n

```
X242                NA         NA       NA       NA
X243                NA         NA       NA       NA
X244                NA         NA       NA       NA
X245                NA         NA       NA       NA
X246                NA         NA       NA       NA
X247                NA         NA       NA       NA
X248                NA         NA       NA       NA
X249                NA         NA       NA       NA
 [ reached getOption("max.print") -- omitted 51 rows ]

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:    NaN
F-statistic:   NaN on 199 and 0 DF,  p-value: NA
```