BIOST 546 HW4

Ronald Buie

Due Via Online Submission to Canvas: Sunday, March 10 at 12 PM (Noon)

Instructions: You may discuss the homework problems in small groups, but you must write up the fnal solutions and code yourself. Please turn in your code for the problems that involve coding. However, code without written answers will receive no credit. To receive credit, you must explain your answers and show your work. All plots should be appropriately labeled and legible, with axis labels, legends, etc., as needed.

1. Suppose we wish to predict a quantitative response Y using X1, which represents height (in meters) and X2, which represents weight (in pounds). We will also consider predicting Y using ~X1, which represents height (in centimeters), and X2.

(a) Prove that the residual sum of squares for the least squares model that predicts Y using X1 and X2 is the same as the residual sum of squares for the least squares model that predicts Y using ~X1 and X2.

$$RSS = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_1 - \hat{\beta}_2 X_2)^2$$

$$RSS = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 \tilde{X}_1 - \hat{\beta}_2 X_2)^2$$

Noting that $\tilde{X}_1$ is from the set inclusive of X_1 the minimum of each is equivalent.

$$RSS = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - 0.01 \times \hat{\beta}_1 X_1 - \hat{\beta}_2 X_2)^2 = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 \tilde{X}_1 - \hat{\beta}_2 X_2)^2$$

and

$$RSS = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - 100 \times \hat{\beta}_1 \tilde{X}_1 - \hat{\beta}_2 X_2)^2 = min\sum_{i=1}^{i}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_1 - \hat{\beta}_2 X_2)^2$$

(b) Let ˆ f0, ˆ f1, ˆ f2 denote the least squares regression coefcients for a model that predicts Y using X1 and X2. Derive the least squares coefcient estimates for a model that predicts Y using X1 and X2. (By derive I mean: state the coefcient estimates, and show mathematically why these are the coefcient estimates.)

(c) Prove that the ftted values for the least squares model that predicts Y using X1 and X2 are the same as the ftted values for the least squares model that predicts Y using ~X1 and X2.

Setting our coefficients to 1 given

$$f(Y) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$$

and

$$f(Y)' = \hat{\beta}_0 + \hat{\beta}_1 \tilde{X}_1 + \hat{\beta}_2 X_2$$

and

$$\hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 = \hat{\beta}_0 + 100 \times \hat{\beta}_1 \tilde{X}_1 + \hat{\beta}_2 X_2$$

In predicting Y for both models we can hold β_0 and β_2 constant, cancelling them out, leaving

$$f(Y)' = \hat{\beta}_1 X_1$$

and

$$f(Y)' = \hat{\beta}_1 \tilde{X}_1$$

give that $X_1 \times 0.01 = \tilde{X}_1$ and $\tilde{X}_1 \times 100 = X_1$ it follows that $f(Y)' = f(Y) \times 0.01$ and that $f(Y) = f(Y)' \times 100$

(d) Now, for some fxed f > 0, consider performing ridge regression to predict Y using X1 and X2, and also performing a separate ridge regression to predict Y using ~X1 and X2. Which of these ftted models will have a smaller residual sum of squares? Which of these ftted models will have a smaller value of f2 1 + f2 2? Justify your answers.

Note that β_1 coefficient is 100 times larger than the $\tilde{\beta}_1$ coefficient. For any $\lambda > 0$, ridge regression will penalize more greatly those models with the highest squared coefficients. This would be the model contraining β, and so the other would be preffered.

(e) Simulate a quantitative response Y as well as two quantitative featuresX1 and X2, each of length n = 100. Verify numerically that your answers to (a){(c) are correct.

```
y <- sample(1:30, 100, replace = TRUE)
x1 <- rnorm(100, 5)
x2 <- sample(1000:150000, 100,replace = TRUE)
x1tilde <- x1*100
```

```r
sampleData <- as.data.frame(cbind(y,x1,x1tilde, x2))

lmsmall <- lm(y~x1+x2, data = sampleData)
lmlarge <- lm(y~x1tilde + x2 , data = sampleData)
```

```r
RSSsmall <- with(summary(lmsmall), df[2] * sigma^2)
RSSlarge <- with(summary(lmlarge), df[2] * sigma^2)
```

a)

The RSS for model 1 is 7037.4972545 and for model 2 is 7037.4972545. They are the same.

b)

The coefficients for each model are:

```r
summary(lmsmall)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = sampleData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1525  -6.9071   0.7364   7.6952  13.2616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.354e+01  4.584e+00   5.135 1.45e-06 ***
## x1          -1.493e+00  8.682e-01  -1.720   0.0886 .
## x2          -7.180e-07  2.006e-05  -0.036   0.9715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.518 on 97 degrees of freedom
## Multiple R-squared:  0.03061,    Adjusted R-squared:  0.01062
## F-statistic: 1.531 on 2 and 97 DF,  p-value: 0.2214
```

```r
summary(lmlarge)
```

```
##
## Call:
## lm(formula = y ~ x1tilde + x2, data = sampleData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1525  -6.9071   0.7364   7.6952  13.2616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.354e+01  4.584e+00   5.135 1.45e-06 ***
## x1tilde     -1.493e-02  8.682e-03  -1.720   0.0886 .
## x2          -7.180e-07  2.006e-05  -0.036   0.9715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 8.518 on 97 degrees of freedom
## Multiple R-squared:  0.03061,    Adjusted R-squared:  0.01062
## F-statistic: 1.531 on 2 and 97 DF,  p-value: 0.2214
```

c)

using the above coefficients we can see that the coefficients for X1 are the same, except that X1 tilde is two orders of magnitude smaller than X1

(f) Now perform K-nearest-neighbors regression to predict Y using X1 and X2, and also to predict Y using ~X1 and X2. Using the data generated in (e) (or else using diferent data, if needed), show that the KNN regression approach is not scale-invariant.

```r
library(class)
trainsmall <- cbind(sampleData[1:50,]$x1, sampleData[1:50,]$x2 )
trainlarge <- cbind(sampleData[1:50,]$x1tilde, sampleData[1:50,]$x2 )
testsmall <- cbind(sampleData[51:100,]$x1, sampleData[51:100,]$x2 )
testlarge <- cbind(sampleData[51:100,]$x1tilde, sampleData[51:100,]$x2 )
trainY <- cbind(sampleData[1:50,]$y )
testY <- cbind(sampleData[51:100,]$y)
table(knn(trainsmall, testsmall,  trainY, k=4), testY)
```

```
##     testY
##      2 3 6 7 8 9 10 11 12 13 15 16 18 19 20 21 22 23 25 26 27 28 29 30
##   1  0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##   2  0 0 0 0 0 0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0  0  0
##   3  1 0 0 1 2 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##   5  0 0 0 0 0 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
##   7  0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   8  0 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   9  0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0  1  0  0  0  0  0  0
##   10 0 0 0 0 0 0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
##   11 0 1 0 0 1 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  1
##   12 0 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   13 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0
##   14 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   15 0 0 1 0 0 0  0  2  0  0  0  1  0  1  0  0  0  0  1  0  0  0  0  1
##   16 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
##   18 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   19 0 0 0 0 0 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##   21 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   22 0 0 0 0 0 1  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  0  0  0
##   23 0 0 0 1 0 0  0  0  1  0  0  0  0  0  0  0  0  0  1  0  1  1  0  0
##   24 0 0 0 0 0 0  0  0  0  1  0  0  0  1  0  1  0  0  0  0  0  0  0  0
##   26 0 0 0 1 1 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0
##   27 1 1 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##   28 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##   30 0 0 0 0 0 0  1  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
```

```r
table(knn(trainlarge, testlarge,  trainY, k=4), testY)
```

```
##     testY
##      2 3 6 7 8 9 10 11 12 13 15 16 18 19 20 21 22 23 25 26 27 28 29 30
```

```
##    1  0 0 0 0 0 1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  1  0  0
##    2  0 0 0 0 1 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
##    3  0 0 0 0 0 0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##    5  1 0 0 0 0 0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
##    7  0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    8  0 2 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    9  1 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
##    10 0 0 0 0 0 0  0  0  0  1  0  0  0  0  1  1  0  0  0  0  0  0  0  0
##    11 0 0 0 1 0 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0
##    12 0 1 0 0 0 0  0  1  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
##    13 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##    14 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##    15 0 0 0 0 1 0  0  0  0  0  0  0  1  0  0  0  0  0  1  0  0  0  0  0
##    16 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1
##    18 0 0 0 0 0 0  0  0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  0  0
##    19 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
##    21 0 0 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
##    22 0 0 0 0 0 0  0  0  0  0  0  1  1  0  0  0  0  0  0  0  1  0  0  0
##    23 0 0 1 2 0 0  0  0  1  0  0  0  0  0  0  0  0  1  1  0  0  0  0  0
##    24 0 1 0 0 0 0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##    26 0 0 0 0 1 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0
##    27 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
##    28 0 0 0 0 0 0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
##    30 0 0 0 0 0 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

we can see that, while similar, our results are not identical for our KNN predictions between X1 and X1tilde.

(g) Finally, consider ftting a ridge regression model to predict Y using just X1, for a tuning parameter value f > 0. You also consider ftting a ridge regression model to predict Y using just ~X1, for a tuning parameter value ~f 0. Is there a relationship between f and ~f that will make it so that the ftted values for the two models are equal? Justify your answer. If you answered yes, then state the relationship in the most general terms possible.

Yes.Any given lambda has a higher lambda. Ridge regression penalizes the model but does not reduce any coefficient to 0, so that for any lambda there is a $\tilde{\lambda} < \lambda$ that can decrease the fitted value to meet that resulting from the model using λ.

2. In this problem, we wish to predict a quantitative response Y using X1 andX2, where X1 is height in meters, and X2 is height in centimeters.

(a) Suppose that (ˆ f0; ˆ f1; ˆ f2) are least squares coefcient estimates for themodel that uses X1 and X2 to predict Y . Explain why this least squares solution is not unique. Derive a general expression for the set of least squares coefcient estimates (your answer should be written in terms of ˆ f0; ˆ f1; ˆ f2).

$$f(Y) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$$

β_1 and β_2 are linearly corelated such that

$$\beta_1 = \beta_2 * 100$$

and

$$\beta_2 = \beta_1 * 0.01$$

Thus, for any value of beta 1 there is a value of beta 2 that can equal it once the multiplier is applied.

(b) Now suppose that (~ f0; ~ f1; ~ f2) are ridge regression coefcient estimates for the model that uses X1 and X2 to predict Y . Is this ridge regression solution unique? In this instance, is the ridge regression solution sparse? Justify your answers.

The ridge regression is unique because changes in beta 1 and beta 2 for any identical Y will change the square of the coefficients. No ridge regression is sparse; all coeficients will be retained, but may trend towards 0.

(c) Now suppose that (f f0; f f1; f f2) are lasso coefcient estimates for the modelthat uses X1 and X2 to predict Y . Is this lasso solution unique? In this instance, is the lasso solution sparse? Justify your answers.

The lasso regression will be unqiue for the same reasons, but ridge regressions can drop a coefficient to 0 and so are sparse.

(d) Now answer questions (b) and (c) again, but this time for the model that uses X1 and X1 to predict Y . (That was not a typo: you are being asked to consider the model that uses the same predictor twice in order to predictY).

(e) Now suppose we ft a lasso model to predict Y using X1 with some fxed tuning parameter value f > 0. We also ft a lasso model to predict Y using X2, again with the same fxed tuning parameter value f > 0. Which of these two models will have a smaller value for the residual sum of squares? Which of these two models will have a smaller value of the lasso penalty term? Justify your answers.

The two models will have an identical RSS, as this is scale invariant. However for any given Y, the model with the larger coefficient for beta 2 will be penalized the most.

3. In this problem, you will analyze a (real, not simulated) dataset of your choicewith a quantitative response Y , and p f 50 quantitative predictors.

```
FIFAData <- read.csv("./data.csv")
FIFAData <- FIFAData[1:500,]
FIFAData <- FIFAData[,1:51]
names(FIFAData)[1] <- "Value"
```

(a) Describe the data. Where did you get it from? What is the meaning of the response, and what are the meanings of the predictors?

```
summary(FIFAData)
```

```
##       Value             Age           Overall       Potential
## Min.   :  1.70   Min.   :18.00   Min.   :79.0   Min.   :79.00
```

```
##  1st Qu.: 15.50   1st Qu.:24.00   1st Qu.:80.0   1st Qu.:81.00
##  Median : 21.00   Median :27.00   Median :82.0   Median :84.00
##  Mean   : 25.48   Mean   :27.24   Mean   :82.4   Mean   :84.26
##  3rd Qu.: 30.00   3rd Qu.:30.00   3rd Qu.:84.0   3rd Qu.:87.00
##  Max.   :118.50   Max.   :37.00   Max.   :94.0   Max.   :95.00
##       Wage           Special     International.Reputation   Weak.Foot
##  Min.   :  1.00   Min.   :1582   Min.   :1.000            Min.   :2.00
##  1st Qu.: 36.75   1st Qu.:1937   1st Qu.:2.000            1st Qu.:3.00
##  Median : 68.00   Median :2038   Median :2.000            Median :3.00
##  Mean   : 88.02   Mean   :2017   Mean   :2.454            Mean   :3.37
##  3rd Qu.:115.00   3rd Qu.:2106   3rd Qu.:3.000            3rd Qu.:4.00
##  Max.   :565.00   Max.   :2346   Max.   :5.000            Max.   :5.00
##   Skill.Moves    Jersey.Number        Feet          Inches
##  Min.   :2.000   Min.   : 2.00   Min.   :5.00   Min.   : 0.000
##  1st Qu.:3.000   1st Qu.: 7.00   1st Qu.:5.00   1st Qu.: 2.000
##  Median :3.000   Median :11.00   Median :5.00   Median : 7.000
##  Mean   :3.298   Mean   :14.03   Mean   :5.47   Mean   : 5.684
##  3rd Qu.:4.000   3rd Qu.:19.00   3rd Qu.:6.00   3rd Qu.:10.000
##  Max.   :5.000   Max.   :92.00   Max.   :6.00   Max.   :11.000
##      Weight            LS              ST              RS
##  Min.   :130.0   Min.   :493.0   Min.   :493.0   Min.   :493.0
##  1st Qu.:154.0   1st Qu.:682.0   1st Qu.:682.0   1st Qu.:682.0
##  Median :168.0   Median :733.0   Median :733.0   Median :733.0
##  Mean   :167.7   Mean   :722.1   Mean   :722.1   Mean   :722.1
##  3rd Qu.:179.0   3rd Qu.:782.0   3rd Qu.:782.0   3rd Qu.:782.0
##  Max.   :220.0   Max.   :913.0   Max.   :913.0   Max.   :913.0
##       LW              LF              CF              RF
##  Min.   :443.0   Min.   :473.0   Min.   :473.0   Min.   :473.0
##  1st Qu.:693.0   1st Qu.:702.0   1st Qu.:702.0   1st Qu.:702.0
##  Median :772.0   Median :772.0   Median :772.0   Median :772.0
##  Mean   :739.8   Mean   :742.3   Mean   :742.3   Mean   :742.3
##  3rd Qu.:802.0   3rd Qu.:802.0   3rd Qu.:802.0   3rd Qu.:802.0
##  Max.   :922.0   Max.   :932.0   Max.   :932.0   Max.   :932.0
##       RW             LAM             CAM             RAM
##  Min.   :443.0   Min.   :463.0   Min.   :463.0   Min.   :463.0
##  1st Qu.:693.0   1st Qu.:712.0   1st Qu.:712.0   1st Qu.:712.0
##  Median :772.0   Median :782.0   Median :782.0   Median :782.0
##  Mean   :739.8   Mean   :748.1   Mean   :748.1   Mean   :748.1
##  3rd Qu.:802.0   3rd Qu.:803.0   3rd Qu.:803.0   3rd Qu.:803.0
##  Max.   :922.0   Max.   :932.0   Max.   :932.0   Max.   :932.0
##       LM             LCM              CM             RCM
##  Min.   :483.0   Min.   :543.0   Min.   :543.0   Min.   :543.0
##  1st Qu.:712.0   1st Qu.:703.0   1st Qu.:703.0   1st Qu.:703.0
##  Median :772.0   Median :752.0   Median :752.0   Median :752.0
##  Mean   :746.6   Mean   :742.4   Mean   :742.4   Mean   :742.4
##  3rd Qu.:802.0   3rd Qu.:783.0   3rd Qu.:783.0   3rd Qu.:783.0
##  Max.   :912.0   Max.   :883.0   Max.   :883.0   Max.   :883.0
##       RM             LWB           LDM             CDM
##  Min.   :483.0   Min.   :503   Min.   :482.0   Min.   :482.0
##  1st Qu.:712.0   1st Qu.:642   1st Qu.:612.0   1st Qu.:612.0
##  Median :772.0   Median :713   Median :742.0   Median :742.0
##  Mean   :746.6   Mean   :703   Mean   :704.4   Mean   :704.4
##  3rd Qu.:802.0   3rd Qu.:763   3rd Qu.:782.0   3rd Qu.:782.0
##  Max.   :912.0   Max.   :853   Max.   :873.0   Max.   :873.0
```

```
##        RDM              RWB            LB              LCB
##  Min.   :482.0   Min.   :503   Min.   :463.0   Min.   :382.0
##  1st Qu.:612.0   1st Qu.:642   1st Qu.:602.0   1st Qu.:542.8
##  Median :742.0   Median :713   Median :717.5   Median :703.0
##  Mean   :704.4   Mean   :703   Mean   :688.5   Mean   :667.0
##  3rd Qu.:782.0   3rd Qu.:763   3rd Qu.:763.0   3rd Qu.:792.0
##  Max.   :873.0   Max.   :853   Max.   :843.0   Max.   :873.0
##        CB              RCB              RB            Crossing
##  Min.   :382.0   Min.   :382.0   Min.   :463.0   Min.   :17.00
##  1st Qu.:542.8   1st Qu.:542.8   1st Qu.:602.0   1st Qu.:62.00
##  Median :703.0   Median :703.0   Median :717.5   Median :74.00
##  Mean   :667.0   Mean   :667.0   Mean   :688.5   Mean   :69.61
##  3rd Qu.:792.0   3rd Qu.:792.0   3rd Qu.:763.0   3rd Qu.:79.00
##  Max.   :873.0   Max.   :873.0   Max.   :843.0   Max.   :93.00
##    Finishing     HeadingAccuracy  ShortPassing      Volleys
##  Min.   :10.00   Min.   :31.00   Min.   :59.00   Min.   :14.00
##  1st Qu.:54.00   1st Qu.:57.00   1st Qu.:76.00   1st Qu.:53.75
##  Median :70.00   Median :69.00   Median :79.00   Median :68.00
##  Mean   :65.32   Mean   :67.48   Mean   :78.93   Mean   :63.94
##  3rd Qu.:78.00   3rd Qu.:80.00   3rd Qu.:83.00   3rd Qu.:76.25
##  Max.   :95.00   Max.   :94.00   Max.   :93.00   Max.   :90.00
##    Dribbling         Curve         FKAccuracy     LongPassing
##  Min.   :42.00   Min.   :20.00   Min.   :10.00   Min.   :35.00
##  1st Qu.:72.00   1st Qu.:61.00   1st Qu.:53.00   1st Qu.:67.75
##  Median :79.00   Median :74.00   Median :66.00   Median :74.00
##  Mean   :76.73   Mean   :69.02   Mean   :62.46   Mean   :72.35
##  3rd Qu.:84.00   3rd Qu.:81.00   3rd Qu.:76.00   3rd Qu.:79.00
##  Max.   :97.00   Max.   :94.00   Max.   :94.00   Max.   :93.00
##   BallControl     Acceleration    SprintSpeed
##  Min.   :54.00   Min.   :34.00   Min.   :31.00
##  1st Qu.:77.00   1st Qu.:67.00   1st Qu.:67.00
##  Median :81.50   Median :75.00   Median :75.00
##  Mean   :79.88   Mean   :73.68   Mean   :73.94
##  3rd Qu.:84.00   3rd Qu.:84.00   3rd Qu.:82.00
##  Max.   :96.00   Max.   :97.00   Max.   :96.00
```

These are FIFA data. All categorical data have been removed and numerics containing non numeric characters have had the non numeric portion (such as euro signs) removed. The remaining values are summarized above. We are using Wage as our outcome, and all other variables as our predictors. These variables include performance statistics, heright, estimated value, age, and the cost of releasing the player. Only the first 50 features have been kept.

(b) Fit a least squares linear model to the data, and provide an estimate of the test error. (Explain how you got this estimate.)

Below we use LOO Cross validation with K=4 to estimate the MSE.

```r
library(boot)
set.seed(1000)
FIFAglm <- glm(Value~., data = FIFAData)
cv.err <- cv.glm(FIFAData, FIFAglm, K = 4)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
cv.err$delta
```

```
## [1] 24.98335 24.41619
```

(c) Fit a ridge regression model to the data, with a range of values of the tuning parameter f. Make a plot like the left-hand panel of Figure 6.4 in the textbook.

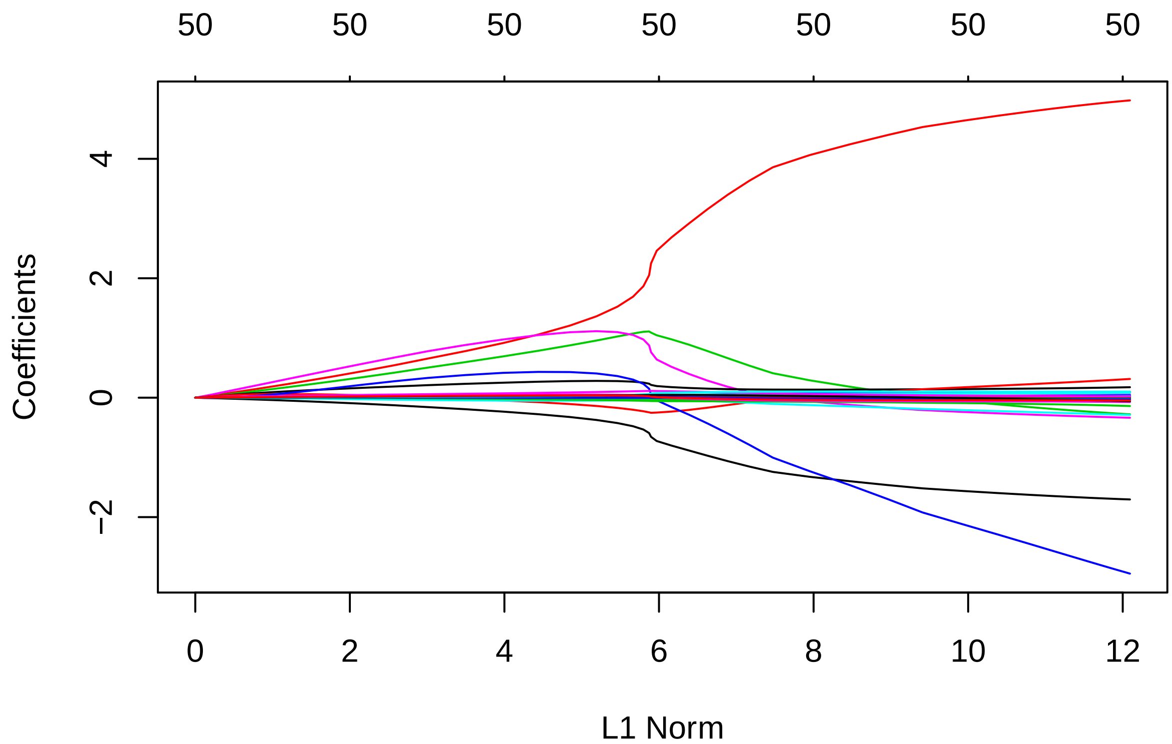
```
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
```

```
set.seed(2000)


predictors = model.matrix(Value ~ ., data = FIFAData)[,-1]
outcome = FIFAData$Value

grid = 10^seq(10, -2, length = 100)
ridge_mod = glmnet(predictors, outcome, alpha = 0, lambda = grid)
dim(coef(ridge_mod))
```

```
## [1]  51 100
```

```
plot(ridge_mod)
```



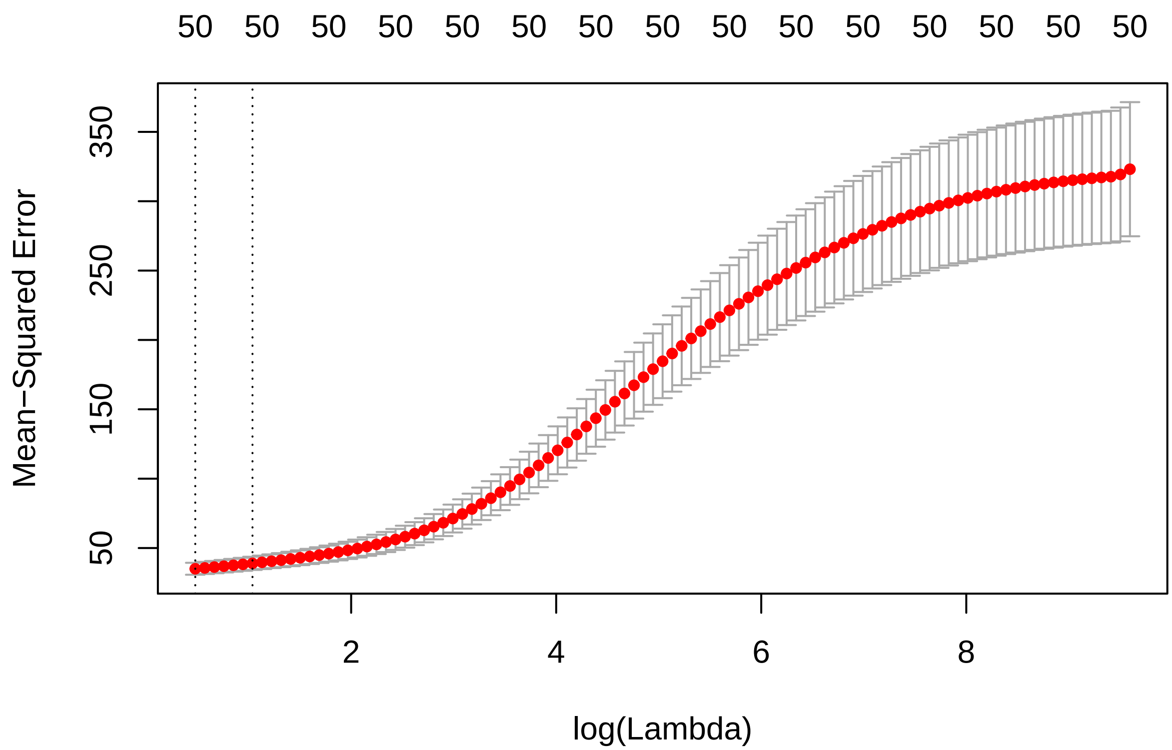
(d) What value of f in the ridge regression model provides the smallest estimated test error? Report this estimate of test error. (Also, explain how you estimated test error.)

```r
library(Momocs)
```

```
## This is Momocs 1.2.9

##
## Attaching package: 'Momocs'

## The following object is masked from 'package:stats':
##
##     filter
```

```r
library(glmnet)
set.seed(3000)

PredictTrain <- model.matrix(Value~., FIFAData[1:250,])
PredictTrain <- PredictTrain[,-1]
PredictTest <- model.matrix(Value~., FIFAData[251:500,])
PredictTest <- PredictTest[,-1]
ResponseTrain <- FIFAData[1:250,]$Value
ResponseTest <- FIFAData[251:500,]$Value

cv.out <- cv.glmnet(PredictTrain, ResponseTrain, alpha = 0)
bestlamda <- cv.out$lambda.min
```

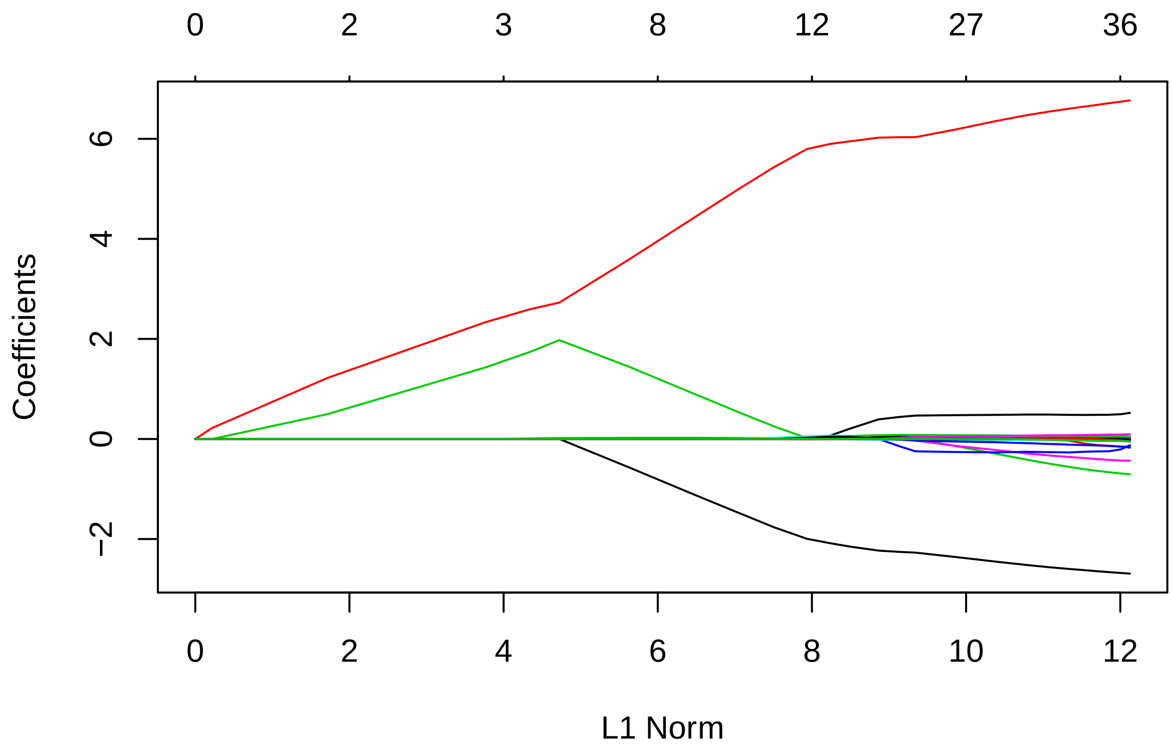
```r
plot(cv.out)
```




```r
prediction <- predict(ridge_mod, s = bestlamda, newx = PredictTest)
bestMSE <- mean((prediction - ResponseTest)^2)
```

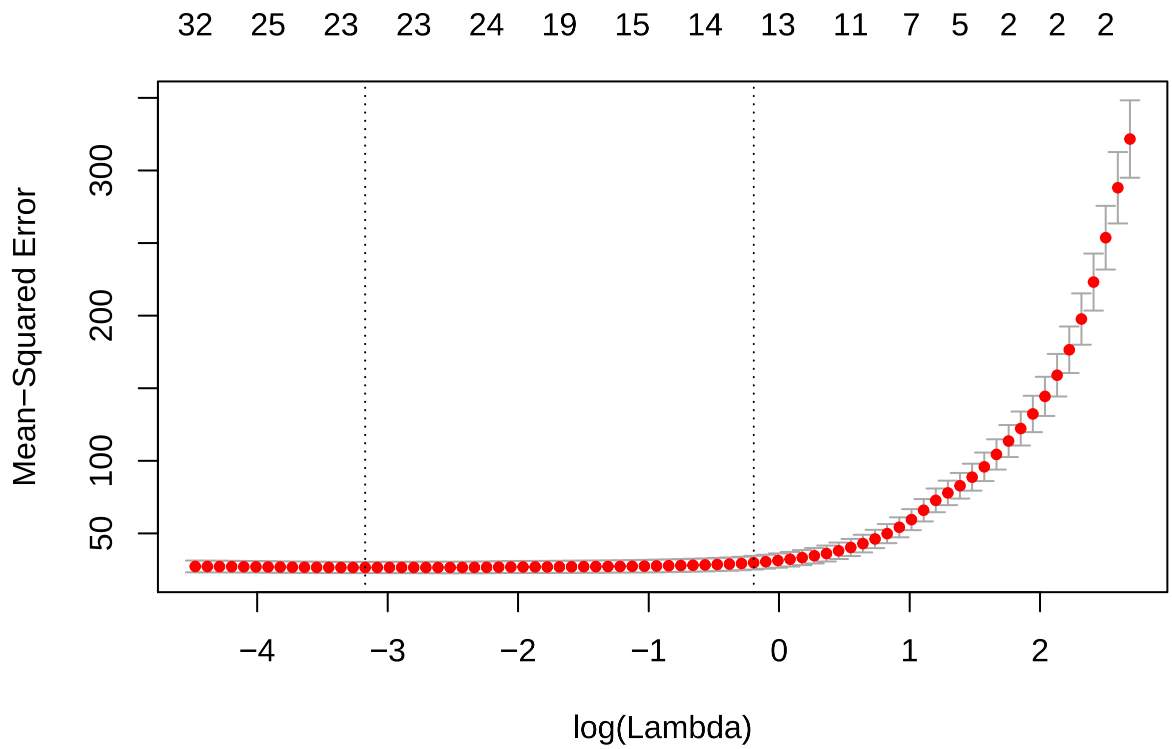
(e) Repeat (c), but for a lasso model.

```r
set.seed(4000)

FIFALasso <- glmnet(PredictTrain, ResponseTrain, alpha = 1,lambda = grid)

plot(FIFALasso)
```



(f) Repeat (d), but for a lasso model. Which features are included in this lasso model?

```
set.seed(5000)

FIFALassoCV <- cv.glmnet(PredictTrain, ResponseTrain, alpha = 1)
plot(FIFALassoCV)
```




```r
bestLASSOlamda <- FIFALassoCV$lambda.min

lasso_pred <- predict(FIFALasso, s = bestlamda, newx = PredictTest)
LASSOMSE <- mean((lasso_pred - ResponseTest)^2)


out<- glmnet(as.matrix(FIFAData[,-1]), FIFAData$Value, alpha = 1, lambda = grid)
lasso_coef <- predict(FIFALasso, type = "coefficients", s = bestLASSOlamda)[1:50,]

lasso_coef[lasso_coef != 0] # Display only non-zero coefficients
```

```
##              (Intercept)                      Age                  Overall
##            -4.077122e+02            -2.594310e+00             6.595517e+00
##                Potential                     Wage                  Special
##            -5.498375e-01             1.903937e-02             1.116756e-04
## International.Reputation                Weak.Foot              Skill.Moves
##            -3.560790e-01             4.825066e-01            -3.068076e-02
##            Jersey.Number                     Feet                   Weight
##            -1.188351e-02            -2.691004e-01            -1.152771e-02
##                       LF                       CF                       LM
##             3.008596e-04             3.263152e-06             1.766861e-02
##                       RM                       LB                      LCB
##             1.601317e-04            -1.506130e-02            -3.235854e-03
##                       CB                      RCB                       RB
##            -3.545916e-16            -2.276273e-03            -3.398733e-04
##                 Crossing                Finishing          HeadingAccuracy
```

```
##          -1.115715e-01            2.091775e-02            1.715707e-02
##                Volleys               Dribbling              FKAccuracy
##           2.729860e-02            6.608360e-02           -6.824695e-04
##            LongPassing
##           7.587595e-02
```

The best MSE is 24.4843767 and is provided by a lamda of 0.0419142.

4. Consider predicting a quantitative response using p features, using a linear regression modelt via least squares. LetMBSSk k denote the best feature models in the best subset, forward stepwise, and backward stepwise selection procedures. Recall that the training set residual sum of squares (or RSS for short) is ll in the blank with one of the following: s than or equalto, greater than or equal toequal toot enough information totell if it is not possible to complete the sentence as given. Justify your answers.

(a) Claim: The RSS of MBWD p is the RSS of MBSS p .

equal. For any value of p, BWD will optimall choose this first model. BSS will optimally choose model for any of p = p to p = 0

(b) Claim: The RSS of MBWD p??1 is the RSS of MBSS p??1 .

equal or greater than. While BSS will always choose the optimal model. BWD may fail to by keeping the choice of dropped parameter in model p. it may be that p-1 should include that parameter.

(c) Claim: The RSS of MBWD 4 is the RSS of MBSS4 .

equal, this is the same as (a)

(d) Claim: The RSS of MBWD4 is the RSS of MFWD4 .

equal. in the case that p = max(p), both models will use all features.

(e) Claim: The RSS of MFWD 1 is the RSS of MBWD 1 .

equal. FWD will optimally choose the first model above p=0 (p=1), and BWD will optimally choose the first model of p. In this example, those are the same.

(f) Claim: The RSS of MFWD 0 is the RSS of MBWD 0 .

equal, in the case of p=0, both algorithms will choose the empty model

(g) Claim: The RSS of MFWD1 is the RSS of MBSS1 .

equal. in the case of P=1, FWD will choose the optimal (there is no prior guess to retain) and BSS will always choose the optimal. It is also the case that this is the set of all choices

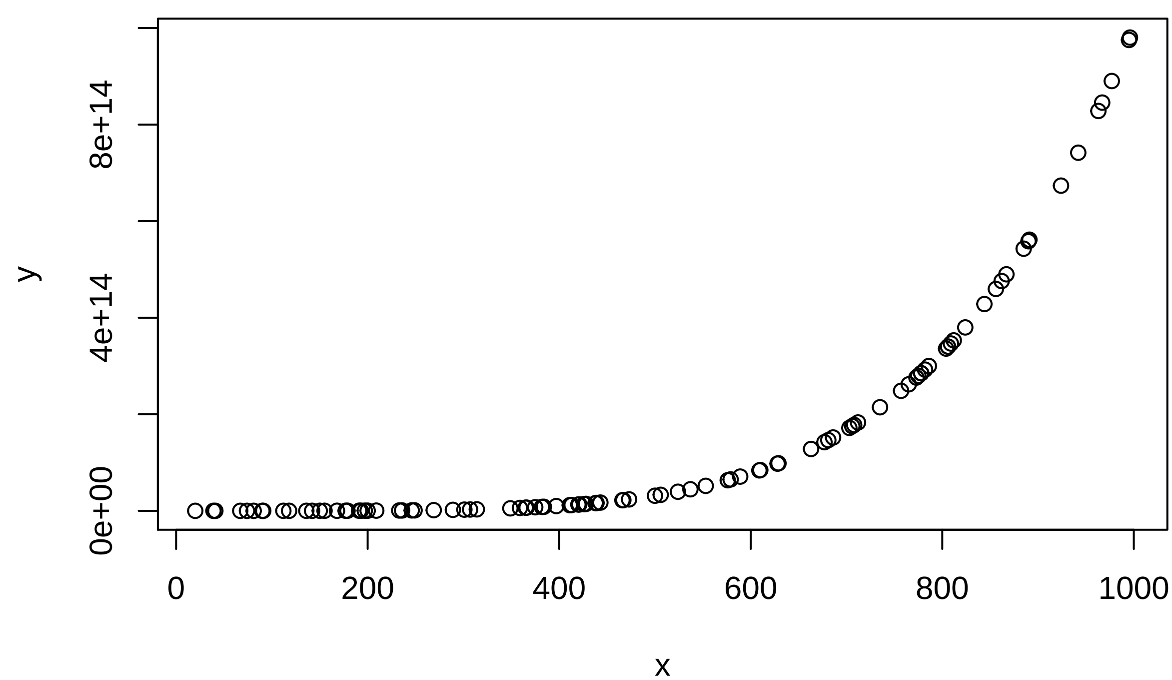
(h) Claim: The RSS of MBWD 1 is the RSS of MBSS 1 .

equal. BWD will start with the full model, BSS will always choose the optimal model. In this case those are the same.

5. In this problem, you will simulate some data, and then will carry out forward and backward stepwise selection on the simulated data.

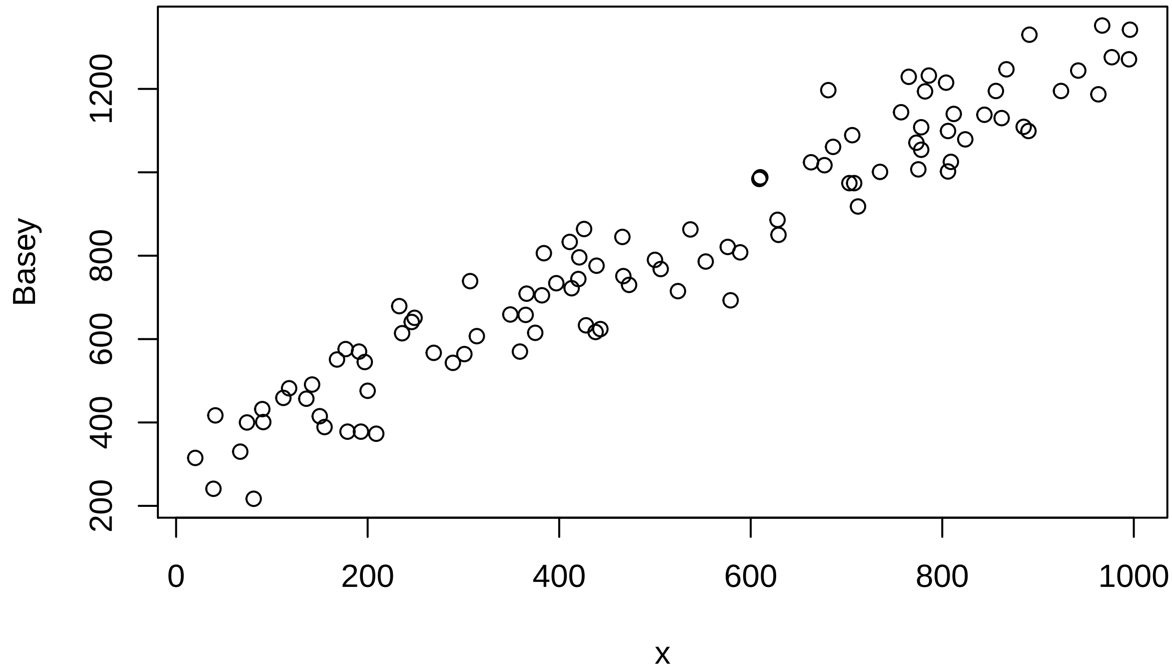
(a) Simulate a quantitative predictor X with n = 100. Then, generate aresponse Y according to the modelY = f0 + f1X + f2X2 + f3X5 + e:Provide details of how you generated X, how you chose f0; : : : ; f3, and how you generated e.

```
set.seed(6000)

betafunction <- function(x) {
 #(x^2)/400
  x+1
}
b0 = 300

x <- sample(1:1000,100, replace = TRUE)
y<- 1:100
for (i in 1:100) {
 y[i] <- b0 + betafunction(x[i]) + betafunction(x[i]^2) + betafunction(x[i]^5) + round(rnorm(1,0,80),0)
}
plot(y~x)
```




```
Basey <- 1:100

for (i in 1:100) {
 Basey[i] <- b0 + betafunction(x[i]) + round(rnorm(1,0,80),0)
```

```
}

Q5Data <- as.data.frame(cbind(y= Basey,x))
plot(Basey~x)
```



Our predictors are a set of random numbers from 20 to 996. The beta coefficient for β_0 was generated as an arbitrary number 300. Our β_1 coefficient was generated using a static formula $(X^2)/400$ so that the β_2 and β_3 of this formula are the squared and pentagonal powers of this formula. Our error term was generated as an integer from a normal distribution with a mean of 0 and standard deviation of 800.

(b) Fit a least squares linear model to predict Y using X;X2; : : : ;X10, andreport the coefcient estimates obtained, as well as the p-values corresponding to null hypotheses of the form H0j : fj = 0. Comment on your results, in light of the way you generated the data in (a).

```
Q5DataExp <- as.data.frame(cbind(Q5Data, Q5Data$x^2, Q5Data$x^3, Q5Data$x^4, Q5Data$x^5, Q5Data$x^6, Q5
test <- lm(y~., data = Q5DataExp)
summary(test)
```

```
##
## Call:
## lm(formula = y ~ ., data = Q5DataExp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -153.383  -64.494    0.638   55.744  188.613
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.195e+02  2.308e+02   1.817   0.0726 .
## x             -7.895e+00  1.256e+01  -0.628   0.5313
## `Q5Data$x^2`   1.907e-01  2.423e-01   0.787   0.4334
## `Q5Data$x^3`  -1.902e-03  2.297e-03  -0.828   0.4099
## `Q5Data$x^4`   1.054e-05  1.237e-05   0.852   0.3962
## `Q5Data$x^5`  -3.508e-08  4.073e-08  -0.861   0.3914
## `Q5Data$x^6`   7.258e-11  8.489e-11   0.855   0.3949
## `Q5Data$x^7`  -9.394e-14  1.124e-13  -0.836   0.4054
## `Q5Data$x^8`   7.390e-17  9.152e-17   0.808   0.4215
## `Q5Data$x^9`  -3.230e-20  4.179e-20  -0.773   0.4415
## `Q5Data$x^10`  6.018e-24  8.186e-24   0.735   0.4642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.34 on 89 degrees of freedom
## Multiple R-squared:  0.9314, Adjusted R-squared:  0.9237
## F-statistic: 120.8 on 10 and 89 DF,  p-value: < 2.2e-16
```

Because we know the true relationship of x and y is 1 to 1, we know that any amount of exponentiation would create a reduced fit. We see this in our regression where, including polynomials up to 10, reducing all of them to insignicance.

(c) Using the leaps package in R, perform forward stepwise selection. Write out the least squares linear model that you found using forward stepwise selection (specify both the predictors and the coefcients in thismodel). Comment on your results, in light of the way you generated the data in (a).

```r
library(leaps)
set.seed(7000)
forwardStep <- regsubsets(y~., data = Q5DataExp, method = "forward", nvmax = 10)


results <-summary(forwardStep)

results$which[which.min(results$adjr2),]
```

```
##   (Intercept)             x  `Q5Data$x^2`  `Q5Data$x^3`  `Q5Data$x^4`
##          TRUE          TRUE          TRUE         FALSE         FALSE
##  `Q5Data$x^5`  `Q5Data$x^6`  `Q5Data$x^7`  `Q5Data$x^8`  `Q5Data$x^9`
##          TRUE         FALSE         FALSE         FALSE         FALSE
## `Q5Data$x^10`
##          TRUE
```

```r
coef(forwardStep,which.min(results$adjr2))
```

```
##   (Intercept)             x  `Q5Data$x^2`  `Q5Data$x^5` `Q5Data$x^10`
##  2.874432e+02  1.184400e+00 -3.552056e-04  3.942729e-13 -2.263039e-28
```

This model was identified as the best because it ha the lowest adjusted R squared. Surprisngly it also includes three of the polynomial terms. This is surprising because the true association is linier.

(d) Using the leaps package in R, perform backward stepwise selection. Write out the least squares linear model that you found using backward stepwise selection (specify both the predictors and the coefcients in this model). Comment on your results, in light of the way you generated thedata in (a). In what sense is this model

```
library(leaps)
set.seed(7000)
backwardStep <- regsubsets(y~., data = Q5DataExp, method = "backward", nvmax = 10)


results <-summary(backwardStep)

results$which[which.min(results$adjr2),]
```

```
##   (Intercept)             x  `Q5Data$x^2`  `Q5Data$x^3`  `Q5Data$x^4`
##          TRUE         FALSE          TRUE         FALSE         FALSE
##  `Q5Data$x^5`  `Q5Data$x^6`  `Q5Data$x^7`  `Q5Data$x^8`  `Q5Data$x^9`
##         FALSE         FALSE         FALSE         FALSE         FALSE
## `Q5Data$x^10`
##         FALSE
```

```
coef(backwardStep,which.min(results$adjr2))
```

```
##  (Intercept) `Q5Data$x^2`
## 4.998876e+02 9.349381e-04
```

using the adjusted r squared to identify the best model in the backward step algorithm removed all but one coefficient, keeping the 2nd polynomial. It is somewhat surprising that the first coeffiient was not chosen. However, it was the first one removed y the algorithm, preventing its selection as higher order polynomials were removed.

(e) Now generate n = 100 test observations (you can do this using the exact same data-generating set-up used in (a)). Compute the mean squared error of the models obtained in (b){(d) on this test set. Comment on your results.

```
set.seed(8000)

Basey <- 1:100

for (i in 1:100) {
 Basey[i] <- b0 + betafunction(x[i]) + round(rnorm(1,0,80),0)
}

Q5DataTest <- as.data.frame(cbind(y= Basey,x))

Q5DataExpTest <- as.data.frame(cbind(Q5DataTest, Q5DataTest$x^2, Q5DataTest$x^3, Q5DataTest$x^4, Q5DataT

#lr
testlr <- lm(y~., data = Q5DataExpTest)
testSummary <- summary(testlr)

mean(testSummary$residuals^2)
```

```
## [1] 5265.997
```

```r
#fwdlr
testfwdlr <-lm(y~x+ `Q5DataTest$x^2` + `Q5DataTest$x^5` + `Q5DataTest$x^10`, data = Q5DataExpTest)
fwdtestSummary <- summary(testfwdlr)

mean(fwdtestSummary$residuals^2)
```

```
## [1] 5342.888
```

```r
#bwdlr
testbwdlr <-lm(y~ `Q5DataTest$x^2`, data = Q5DataExpTest)
bwdtestSummary <- summary(testbwdlr)

mean(bwdtestSummary$residuals^2)
```

```
## [1] 12123.95
```

The model calculated by our initial regression was the highest performing.