STATS 305A HW #6

Adhitya Venkatesh

November 7, 2017

Theoretical Problems:

1.

a.)

For a single predictor, we have the OLS solution

$$\beta_1^* = \sum_{i=1}^{n+1}(x_i - x^*)(y_i - y*)/\sum_{i=1}^{n+1}(x_i - x^*)^2$$

This represents the β_1 parameter estimate with all n + 1 points where

$$x* = \sum_{i=1}^{n+1} x_i/(n+1), x^- = \sum_{i=1}^{n} x_i/n$$

Thus, $(n+1)x^* = nx^- + x_{n+1}$ and similarly $(n+1)y^* = ny^- + \beta_1 x_{n+1} + \beta_0$

Upon substitution and some tedious algebra, we find

$$\beta_1^* = \sum_{i=1}^{n}(x_i - x^-)(y_i - y^-)/\sum_{i=1}^{n}(x_i - x^-)^2 = \beta_1$$

In other words, the estimate β_1^* with the additional data point is the same as original estimate β_1. Similarly for β_0^*, we have

$$\beta_0^* = y^* - \beta_1^* x^* = y^- - \beta_1 x^- = \beta_0$$

Thus, adding point (x_{n+1}, y_{n+1}) doesn't change parameter estimates.

b.)

Using the result from the previous part and the fact that $h_{ii} = x_i^T(X_i^T X_i)^{-1}x_i$, we have

$$x_i^T \beta_i^* = x_i^T \beta_{-i}^* + h_{ii}(y_i - x_i^T \beta_{-i}^*)$$

Thus,

$$x_i^T \beta_i^* = (1 - h_{ii})x_i^T \beta_{-i}^* + h_{ii} y_i$$

Finally, we arrive at

$$y_i^* = (1 - h_{ii})y_{-i}^* + h_{ii} y_i$$

2.

We define the loss function with LASSO regularization as

$$L(\beta) = \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda|\beta|$$

Now we proceed to take the derivative of L w.r.t β and set it to 0:

$$dL/d\beta = \sum_{i=1}^{n} -2x_i(y_i - X_i\beta_{lasso}) + \lambda sign(\beta_{lasso}) = 0$$

Equivalently,

$$\sum_{i=1}^{n} (x_i y_i - x_i^2\beta_{lasso}) - \lambda sign(\beta_{lasso})/2 = 0$$

Now, utilizing the fact that $\beta_{ols} = \sum_{i=1}^{n} x_i y_i / \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i$ since x is normalized, we have

$$\beta_{lasso} = sign(\beta_{ols}).(|\beta_{ols}| - \lambda/2)$$

3.

First, we establish that $Y \sim (p_i, p_i(1-p_i)/n)$. From Taylor Series approximation, we know that the variance of h(Y) is given by $(h'(\mu))^2 g(\mu)$. Setting this to a constant C, we obtain the differential equation

$$h'(t) = C/\sqrt{g(t)} = C/\sqrt{t(1-t)}$$

Re-arranging variables, we have

$$h(t) = \int 1/\sqrt{t(1-t)}dt$$

To evaluate this integral, we express the denominator $t(1-t) = 1/4 - (t - 1/2)^2$. Substituting $ u = t - 1/2$ and thus $du = dt$, we obtain

$$h(u) = \int 1/\sqrt{(1/2)^2 - u^2}du$$

Finally, we use an integration reference table to deduce

$$h(t) = sin^{-1}(2t-1)$$

Computational Problems:

1.

a.)

```
library(faraway)
library(car)
```

```
## Warning: package 'car' was built under R version 3.3.2
```

```
##
## Attaching package: 'car'
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

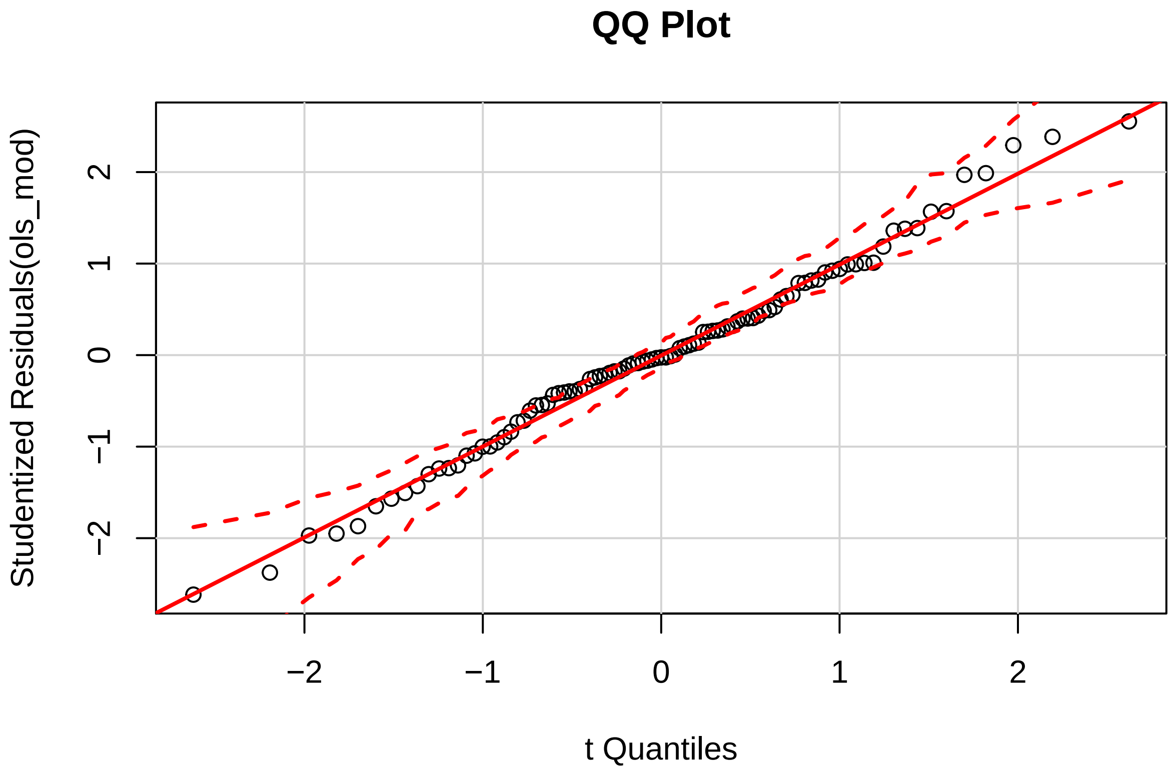
```
data_prost = faraway::prostate
ols_mod = lm(lpsa ~., data = data_prost)

ncvTest(ols_mod)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.2575169    Df = 1     p = 0.6118312
```

From the results, we see the p-value is well above 0.05 and thus we fail to reject the null hypothesis of homoskedasticity.

b.)

```
qqPlot(ols_mod, main = "QQ Plot")
```

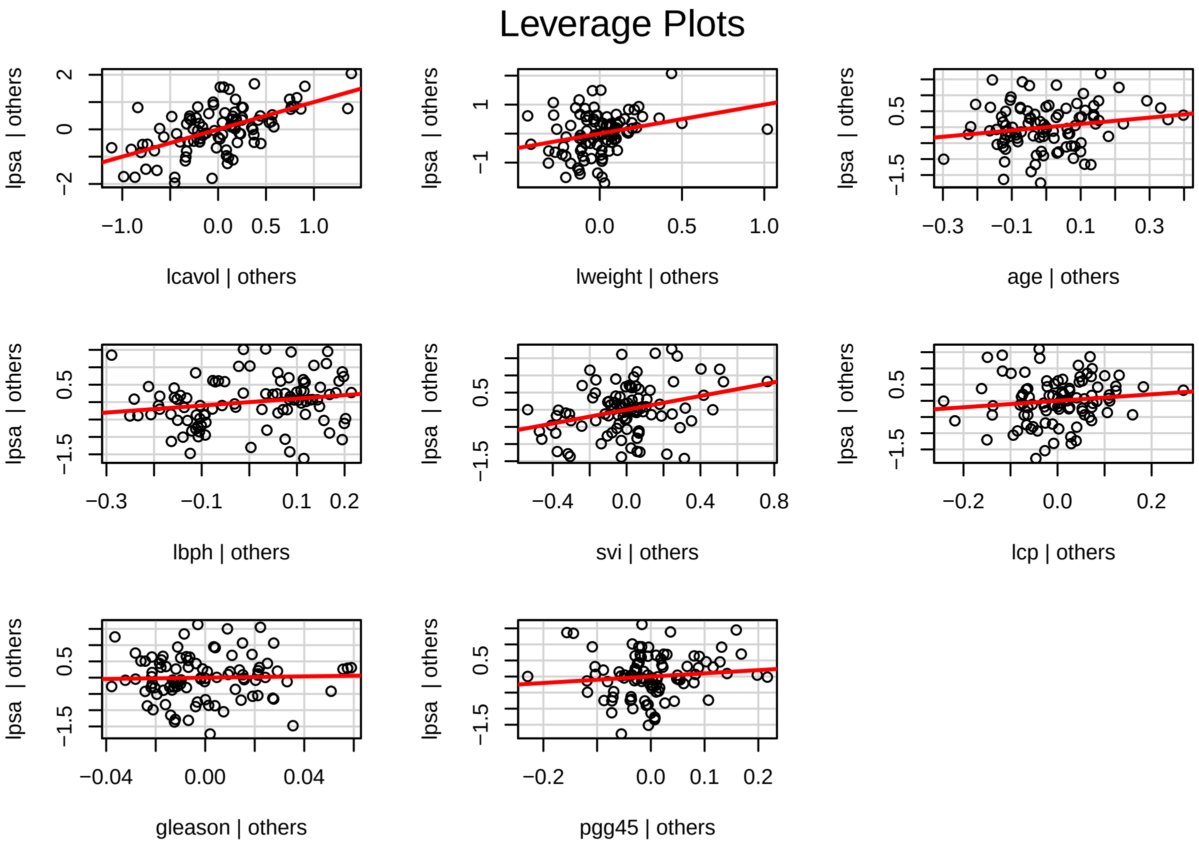


```
#Also plot the histogram
```

Clearly we see that the quantiles of the studentized residuals fit nearly perfectly with those of a standard normal distribution, indicating normality assumption is satisfied.

c.)

```
leveragePlots(ols_mod)
```



```
#discuss results and identify any leverage points
```

It's apparent from the plots that there are leverage points, particularly noticeable in the psa vs weight graph in which there are clearly a few data points with significantly higher weight values that upon removal would noticeably change the parameter estimate.

d.)

```
outlierTest(ols_mod)
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferonni p
## 39 -2.61698            0.01046           NA
```

The result of this test indicates that no outliers exist in this data set as not even the most extreme observation occurs with Bonferroni p-value < 0.05.

e.)

```
library(olsrr)
```

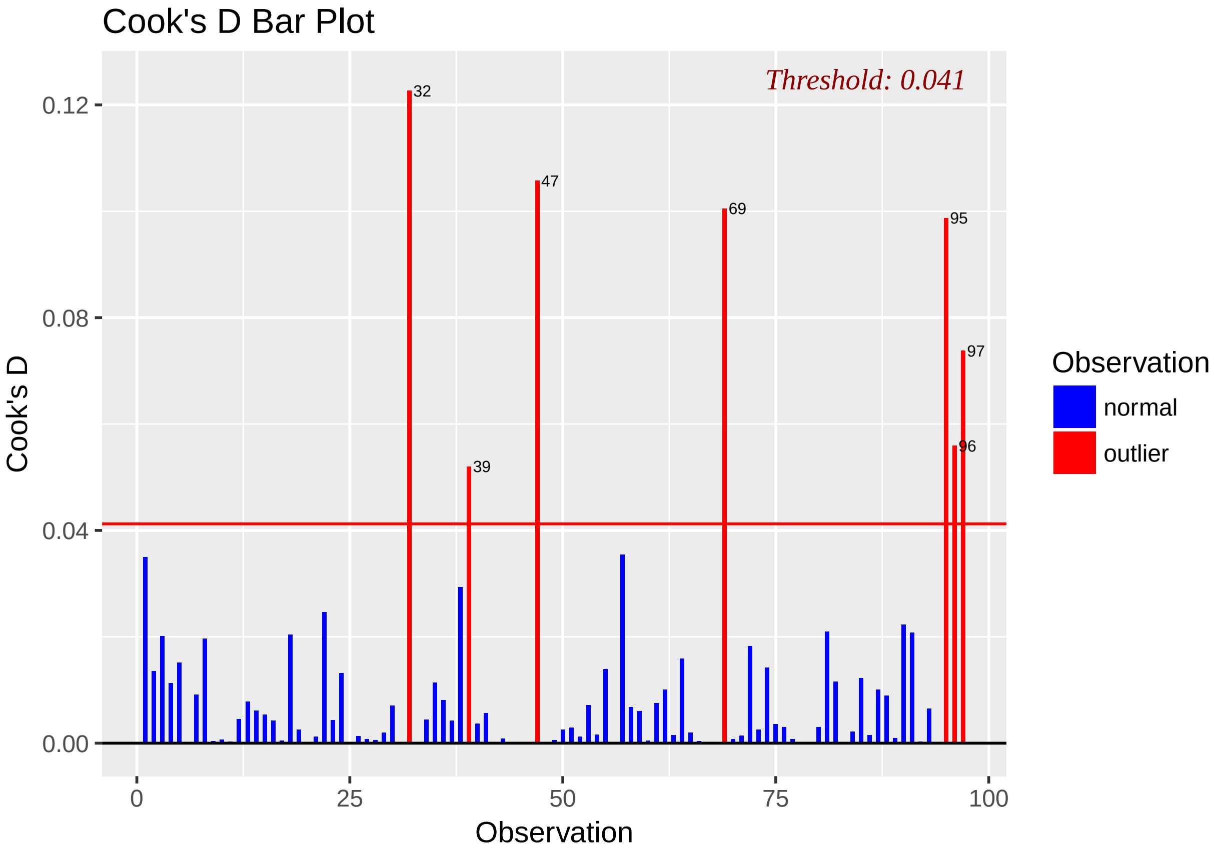
```
## Warning: package 'olsrr' was built under R version 3.3.2
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:faraway':
##
##     hsb
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
ols_cooksd_barplot(ols_mod)
```

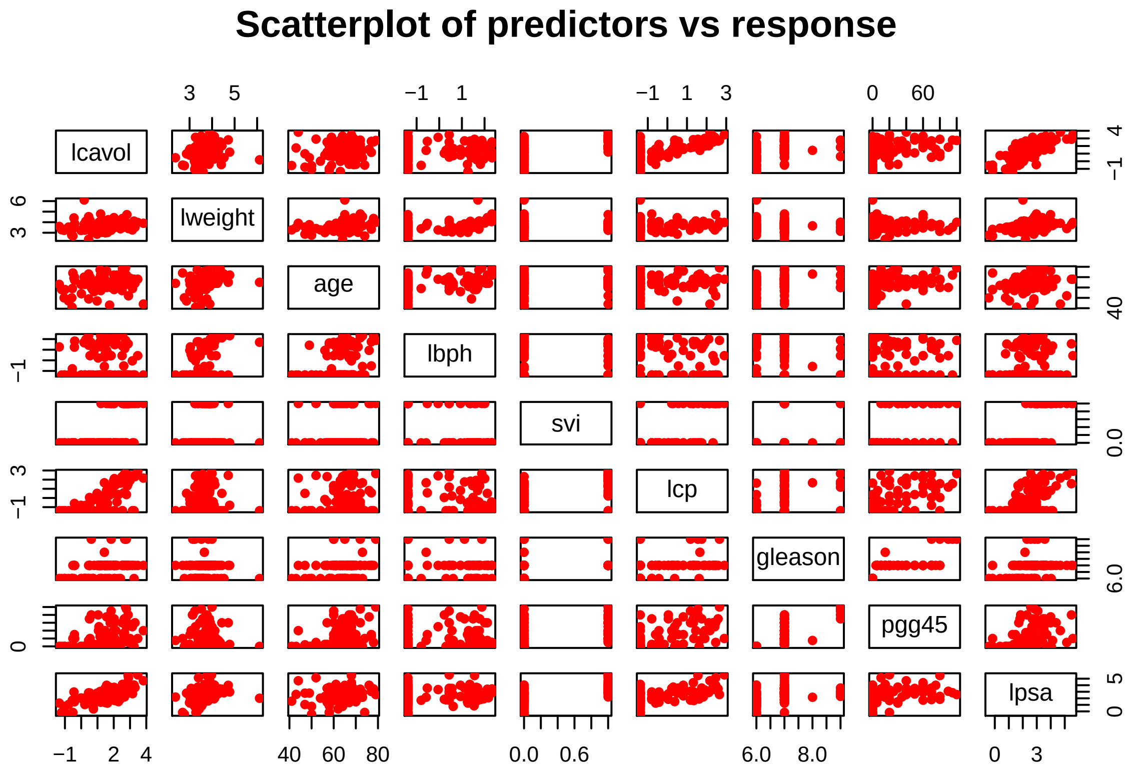
```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```



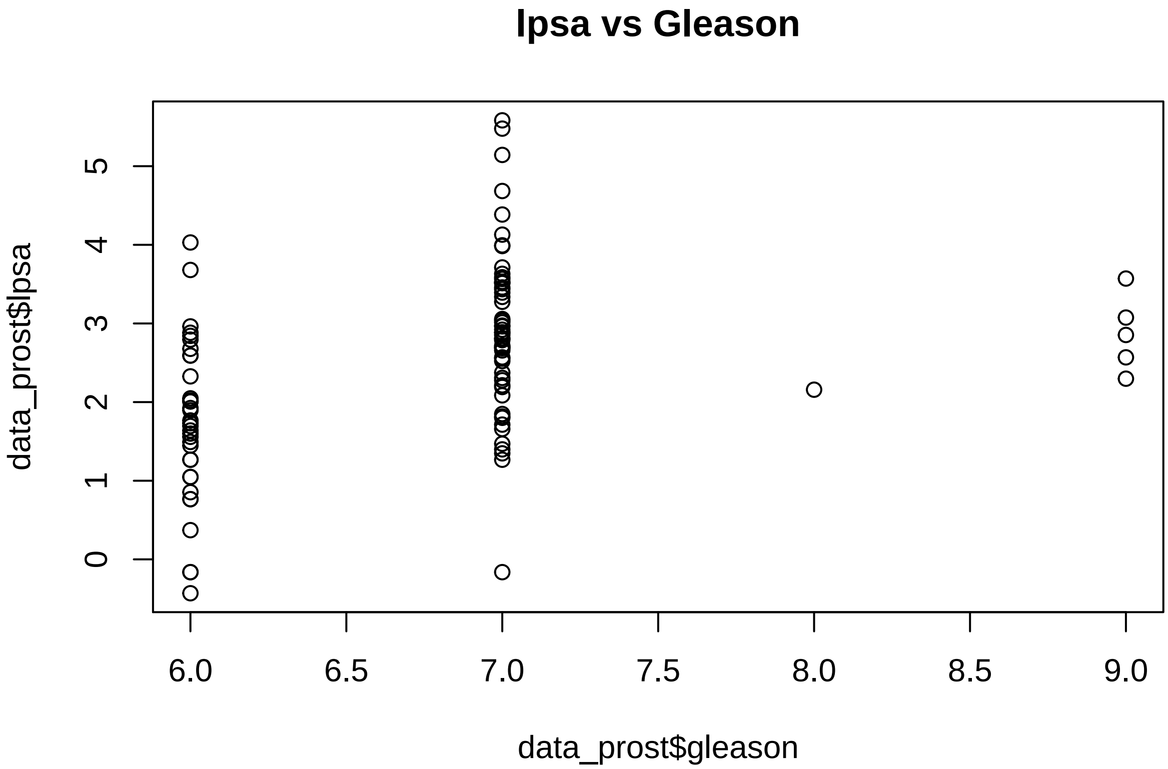
It’s clear that there are in fact influential points (labeled in red).

f.)

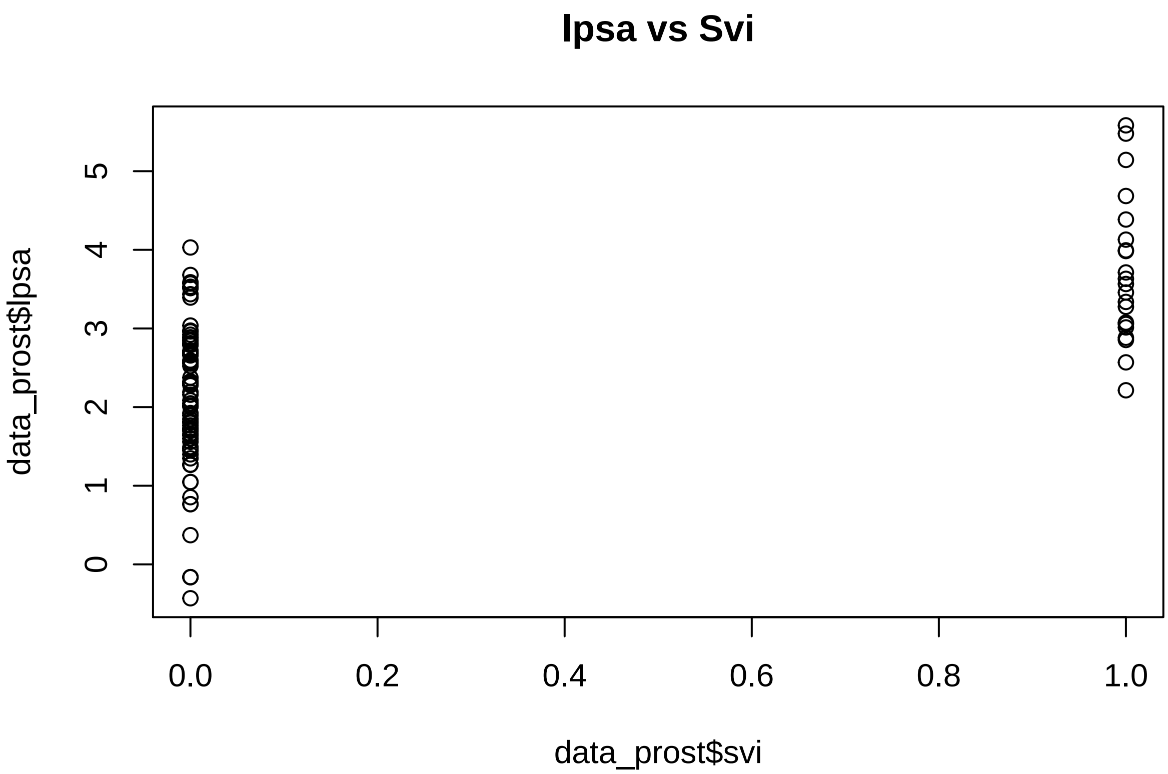
```
plot(data_prost, pch = 16, col= "red", main = "Scatterplot of predictors vs response")
```



```r
plot(data_prost$gleason, data_prost$lpsa, main = "lpsa vs Gleason")
```



```r
plot(data_prost$svi, data_prost$lpsa, main = "lpsa vs Svi")
```



We see that for all predictors bar svi and gleason, there is a relatively linear structure between the predictor value and lpsa level. However for svi and gleason, it's clear that only specific values were used (i.e. categorical variables) to measure the response.

g.)

```
step(ols_mod, data = data_prost, direction = "backward")
```

```
## Start:  AIC=-58.32
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
##
##           Df Sum of Sq    RSS     AIC
## - gleason  1    0.0412 44.204 -60.231
## - pgg45    1    0.5258 44.689 -59.174
## - lcp      1    0.6740 44.837 -58.853
## <none>                 44.163 -58.322
## - age      1    1.5503 45.713 -56.975
## - lbph     1    1.6835 45.847 -56.693
## - lweight  1    3.5861 47.749 -52.749
## - svi      1    4.9355 49.099 -50.046
## - lcavol   1   22.3721 66.535 -20.567
##
## Step:  AIC=-60.23
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - lcp      1    0.6623 44.867 -60.789
## <none>                 44.204 -60.231
## - pgg45    1    1.1920 45.396 -59.650
## - age      1    1.5166 45.721 -58.959
## - lbph     1    1.7053 45.910 -58.560
## - lweight  1    3.5462 47.750 -54.746
## - svi      1    4.8984 49.103 -52.037
## - lcavol   1   23.5039 67.708 -20.872
##
## Step:  AIC=-60.79
## lpsa ~ lcavol + lweight + age + lbph + svi + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - pgg45    1    0.6590 45.526 -61.374
## <none>                 44.867 -60.789
## - age      1    1.2649 46.131 -60.092
## - lbph     1    1.6465 46.513 -59.293
## - lweight  1    3.5647 48.431 -55.373
## - svi      1    4.2503 49.117 -54.009
## - lcavol   1   25.4189 70.285 -19.248
##
## Step:  AIC=-61.37
## lpsa ~ lcavol + lweight + age + lbph + svi
##
##           Df Sum of Sq    RSS     AIC
## <none>                 45.526 -61.374
## - age      1    0.9592 46.485 -61.352
## - lbph     1    1.8568 47.382 -59.497
## - lweight  1    3.2251 48.751 -56.735
## - svi      1    5.9517 51.477 -51.456
## - lcavol   1   28.7665 74.292 -15.871
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = data_prost)
##
## Coefficients:
## (Intercept)       lcavol      lweight          age         lbph
##     0.95100      0.56561      0.42369     -0.01489      0.11184
##         svi
##     0.72095
```

From the results, we see that the resulting model is attained by regressing on the predictors lcavol, lweight, age, lbph, and svi.

h.)

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:olsrr':
##
##     cement
```

```r
stepAIC(ols_mod, direction = "both")
```

```
## Start:  AIC=-58.32
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
```

```
##
##           Df Sum of Sq    RSS     AIC
## - gleason  1    0.0412 44.204 -60.231
## - pgg45    1    0.5258 44.689 -59.174
## - lcp      1    0.6740 44.837 -58.853
## <none>                 44.163 -58.322
## - age      1    1.5503 45.713 -56.975
## - lbph     1    1.6835 45.847 -56.693
## - lweight  1    3.5861 47.749 -52.749
## - svi      1    4.9355 49.099 -50.046
## - lcavol   1   22.3721 66.535 -20.567
##
## Step:  AIC=-60.23
## lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - lcp      1    0.6623 44.867 -60.789
## <none>                 44.204 -60.231
## - pgg45    1    1.1920 45.396 -59.650
## - age      1    1.5166 45.721 -58.959
## - lbph     1    1.7053 45.910 -58.560
## + gleason  1    0.0412 44.163 -58.322
## - lweight  1    3.5462 47.750 -54.746
## - svi      1    4.8984 49.103 -52.037
## - lcavol   1   23.5039 67.708 -20.872
##
## Step:  AIC=-60.79
## lpsa ~ lcavol + lweight + age + lbph + svi + pgg45
##
##           Df Sum of Sq    RSS     AIC
## - pgg45    1    0.6590 45.526 -61.374
## <none>                 44.867 -60.789
## + lcp      1    0.6623 44.204 -60.231
## - age      1    1.2649 46.131 -60.092
## - lbph     1    1.6465 46.513 -59.293
## + gleason  1    0.0296 44.837 -58.853
## - lweight  1    3.5647 48.431 -55.373
## - svi      1    4.2503 49.117 -54.009
## - lcavol   1   25.4189 70.285 -19.248
##
## Step:  AIC=-61.37
## lpsa ~ lcavol + lweight + age + lbph + svi
##
##           Df Sum of Sq    RSS     AIC
## <none>                 45.526 -61.374
## - age      1    0.9592 46.485 -61.352
## + pgg45    1    0.6590 44.867 -60.789
## + gleason  1    0.4560 45.070 -60.351
## + lcp      1    0.1293 45.396 -59.650
## - lbph     1    1.8568 47.382 -59.497
## - lweight  1    3.2251 48.751 -56.735
## - svi      1    5.9517 51.477 -51.456
## - lcavol   1   28.7665 74.292 -15.871
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi, data = data_prost)
##
## Coefficients:
## (Intercept)       lcavol      lweight          age         lbph
##     0.95100      0.56561      0.42369     -0.01489      0.11184
##         svi
##     0.72095
```

```
#discuss model
```

We see that the model yielding the minimum AIC is the one that regresses on lcavol, lweight, age, lbph, and svi.

i.)

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.3.2
```

```
linMod_cp = leaps(data_prost[,1:8], data_prost[,9], method = "Cp")
summary(linMod_cp)
```

```
##       Length Class  Mode
## which 536    -none- logical
## label   9    -none- character
## size   67    -none- numeric
## Cp     67    -none- numeric
```

```
Cp_target = nrow(data_prost)
Cp_vals = linMod_cp$Cp

#we traverse the Cp values to determine which is closest to target p= 9
i_min = 1
for(i in 2:length(Cp_vals)){
    if(abs(Cp_vals[i] - Cp_target) < abs(Cp_vals[i_min] - Cp_target ))
    {
      i_min = i
    }
}
length(Cp_vals)
```

```
## [1] 67
```

```
print(i_min)
```

```
## [1] 3
```

```
linMod_cp$which[i_min]
```

```
## [1] FALSE
```

2.)

a.)

```
fat_data = faraway::fat
test_indices = seq(10, nrow(fat_data)%/%10, length.out = 25)

test_data = fat_data[test_indices,]
```

```
train_data = fat_data[-test_indices,]
train_data = train_data[,-1]
train_data = train_data[,-2]

linReg_mod = lm(siri ~., data = train_data)
summary(linReg_mod)
```

```
##
## Call:
## lm(formula = siri ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8219 -0.6777  0.1306  0.9352  6.6199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.386303   6.388448  -1.939 0.053798 .
## age           0.007742   0.012000   0.645 0.519486
## weight        0.361971   0.023142  15.641  < 2e-16 ***
## height        0.048649   0.039867   1.220 0.223666
## adipos       -0.481622   0.108415  -4.442 1.41e-05 ***
## free         -0.566734   0.014471 -39.164  < 2e-16 ***
## neck          0.015559   0.087245   0.178 0.858626
## chest         0.119226   0.038859   3.068 0.002424 **
## abdom         0.150738   0.039719   3.795 0.000191 ***
## hip          -0.012863   0.054251  -0.237 0.812802
## thigh         0.179414   0.054168   3.312 0.001082 **
## knee          0.136978   0.091218   1.502 0.134620
## ankle         0.124643   0.080301   1.552 0.122054
## biceps        0.103620   0.063367   1.635 0.103432
## forearm       0.230062   0.072013   3.195 0.001605 **
## wrist         0.143273   0.200609   0.714 0.475868
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.531 on 220 degrees of freedom
## Multiple R-squared:  0.9697, Adjusted R-squared:  0.9677
## F-statistic: 470.1 on 15 and 220 DF,  p-value: < 2.2e-16
```

b.)

```
linAIC_mod = stepAIC(linReg_mod, direction = "both")
```

```
## Start:  AIC=216.42
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq   RSS    AIC
## - neck     1       0.1 515.6 214.45
## - hip      1       0.1 515.7 214.48
## - age      1       1.0 516.5 214.86
## - wrist    1       1.2 516.8 214.96
## - height   1       3.5 519.1 216.01
## <none>                 515.6 216.42
## - knee     1       5.3  520.8 216.82
## - ankle    1       5.6  521.2 216.99
## - biceps   1       6.3  521.8 217.27
## - chest    1      22.1  537.6 224.31
## - forearm  1      23.9  539.5 225.12
## - thigh    1      25.7  541.3 225.90
## - abdom    1      33.8  549.3 229.38
## - adipos   1      46.2  561.8 234.69
## - weight   1     573.3 1088.9 390.87
## - free     1    3594.5 4110.1 704.34
##
## Step:  AIC=214.45
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     hip + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - hip      1       0.2  515.8 212.54
## - age      1       1.1  516.7 212.94
## - wrist    1       1.4  517.0 213.10
## - height   1       3.5  519.2 214.07
## <none>                  515.6 214.45
## - knee     1       5.2  520.9 214.83
## - ankle    1       5.6  521.2 214.99
## - biceps   1       6.4  522.0 215.35
## + neck     1       0.1  515.6 216.42
## - chest    1      22.0  537.6 222.31
## - forearm  1      24.5  540.2 223.43
## - thigh    1      26.1  541.7 224.11
## - abdom    1      33.9  549.6 227.49
## - adipos   1      46.9  562.6 233.01
## - weight   1     593.6 1109.3 393.24
## - free     1    3660.0 4175.6 706.07
##
## Step:  AIC=212.54
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - age      1       1.1  517.0 211.05
## - wrist    1       1.5  517.3 211.21
## - height   1       3.9  519.7 212.31
## <none>                  515.8 212.54
## - knee     1       5.1  520.9 212.84
## - ankle    1       5.7  521.5 213.11
## - biceps   1       6.7  522.5 213.57
## + hip      1       0.2  515.6 214.45
## + neck     1       0.1  515.7 214.48
## - chest    1      24.1  540.0 221.33
## - forearm  1      25.4  541.2 221.88
## - thigh    1      27.1  542.9 222.62
## - abdom    1      34.3  550.2 225.74
## - adipos   1      48.4  564.2 231.71
## - weight   1     677.8 1193.6 408.53
## - free     1    3681.8 4197.6 705.31
```

```
##
## Step:  AIC=211.05
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - wrist    1       3.3  520.3 210.57
## - height   1       3.5  520.5 210.67
## <none>                  517.0 211.05
## - ankle    1       5.1  522.1 211.38
## - knee     1       6.7  523.7 212.10
## - biceps   1       7.2  524.1 212.30
## + age      1       1.1  515.8 212.54
## + neck     1       0.3  516.7 212.94
## + hip      1       0.2  516.7 212.94
## - forearm  1      24.5  541.4 219.96
## - chest    1      25.2  542.1 220.27
## - thigh    1      26.8  543.8 221.00
## - abdom    1      41.5  558.5 227.30
## - adipos   1      48.4  565.3 230.16
## - weight   1     690.7 1207.7 409.30
## - free     1    3720.9 4237.8 705.56
##
## Step:  AIC=210.57
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     knee + ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## - height   1       4.1  524.4 210.44
## <none>                  520.3 210.57
## + wrist    1       3.3  517.0 211.05
## + age      1       3.0  517.3 211.21
## - ankle    1       6.9  527.2 211.69
## + neck     1       1.1  519.2 212.07
## + hip      1       0.5  519.8 212.36
## - knee     1       8.4  528.7 212.37
## - biceps   1       8.7  529.0 212.47
## - thigh    1      23.8  544.1 219.13
## - chest    1      24.5  544.8 219.45
## - forearm  1      28.7  549.0 221.24
## - abdom    1      45.5  565.8 228.37
## - adipos   1      46.6  566.9 228.83
## - weight   1     688.6 1208.9 407.53
## - free     1    3874.9 4395.2 712.17
##
## Step:  AIC=210.44
## siri ~ weight + adipos + free + chest + abdom + thigh + knee +
##     ankle + biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## <none>                  524.4 210.44
## + height   1       4.1  520.3 210.57
## + wrist    1       3.9  520.5 210.67
## + age      1       2.6  521.8 211.25
## + neck     1       1.5  522.9 211.77
## - ankle    1       7.5  531.9 211.80
## - knee     1       7.6  532.1 211.86
## + hip      1       1.0  523.4 211.99
## - biceps   1       9.1  533.6 212.51
## - thigh    1      21.4  545.8 217.87
## - chest    1      24.9  549.3 219.40
## - forearm  1      29.8  554.3 221.50
## - abdom    1      47.8  572.2 229.02
## - adipos   1      85.9  610.3 244.24
## - weight   1     815.4 1339.8 429.80
## - free     1    3887.4 4411.9 711.06
```

```
summary(linAIC_mod)
```

```
##
## Call:
## lm(formula = siri ~ weight + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8087 -0.6210  0.1499  0.9212  6.8098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.05493    3.89173  -2.070 0.039617 *
## weight       0.36619    0.01958  18.703  < 2e-16 ***
## adipos      -0.54515    0.08980  -6.071 5.35e-09 ***
## free        -0.56181    0.01376 -40.839  < 2e-16 ***
## chest        0.12287    0.03758   3.270 0.001246 **
## abdom        0.16604    0.03667   4.527 9.68e-06 ***
## thigh        0.13876    0.04581   3.029 0.002742 **
## knee         0.15508    0.08563   1.811 0.071468 .
## ankle        0.13915    0.07752   1.795 0.074017 .
## biceps       0.12259    0.06194   1.979 0.049018 *
## forearm      0.24709    0.06905   3.578 0.000423 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.527 on 225 degrees of freedom
## Multiple R-squared:  0.9692, Adjusted R-squared:  0.9679
## F-statistic: 708.7 on 10 and 225 DF,  p-value: < 2.2e-16
```

c.)

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-5
```

```
train_data = as.matrix(train_data)
linRidge_mod = cv.glmnet(train_data[,-1], train_data[,1], family = 'gaussian', alpha = 0)
```

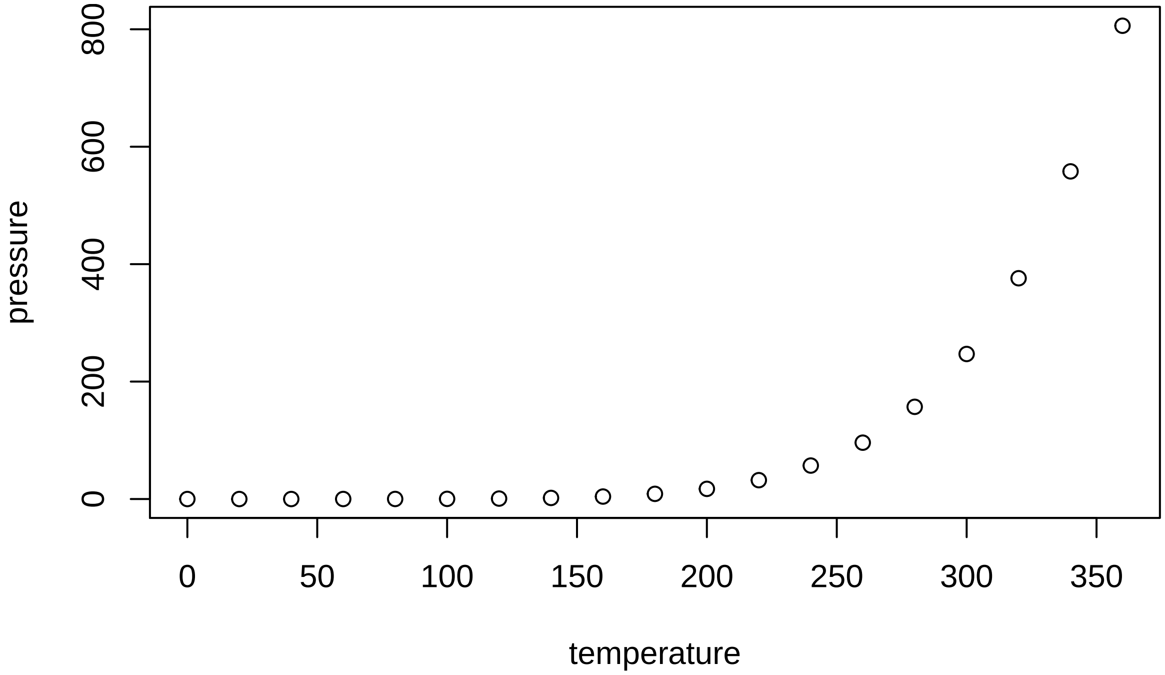
```r
summary(linRidge_mod)
```

```
##            Length Class  Mode     
## lambda     99     -none- numeric  
## cvm        99     -none- numeric  
## cvsd       99     -none- numeric  
## cvup       99     -none- numeric  
## cvlo       99     -none- numeric  
## nzero      99     -none- numeric  
## name        1     -none- character
## glmnet.fit 12     elnet  list     
## lambda.min  1     -none- numeric  
## lambda.1se  1     -none- numeric  
```

d.)

```r
linLasso_mod = cv.glmnet(train_data[,-1], train_data[,1], family = 'gaussian', alpha = 1)
summary(linLasso_mod)
```

```
##            Length Class  Mode     
## lambda     67     -none- numeric  
## cvm        67     -none- numeric  
## cvsd       67     -none- numeric  
## cvup       67     -none- numeric  
## cvlo       67     -none- numeric  
## nzero      67     -none- numeric  
## name        1     -none- character
## glmnet.fit 12     elnet  list     
## lambda.min  1     -none- numeric  
## lambda.1se  1     -none- numeric  
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.