Lab 5: Math158

Joshua Jansen-Montoya

2022-11-22

Problem 9.3

Using the ozone data, fit a model with O3 as the response and temp, humidity and ibh as predictors. Use the Box–Cox method to determine the best transformation on the response.

Answer 9.3

First, we can fit our model as follows,

```
library(faraway)
lmod <- lm(O3 ~ temp + humidity + ibh, data = ozone)
summary(lmod)
```

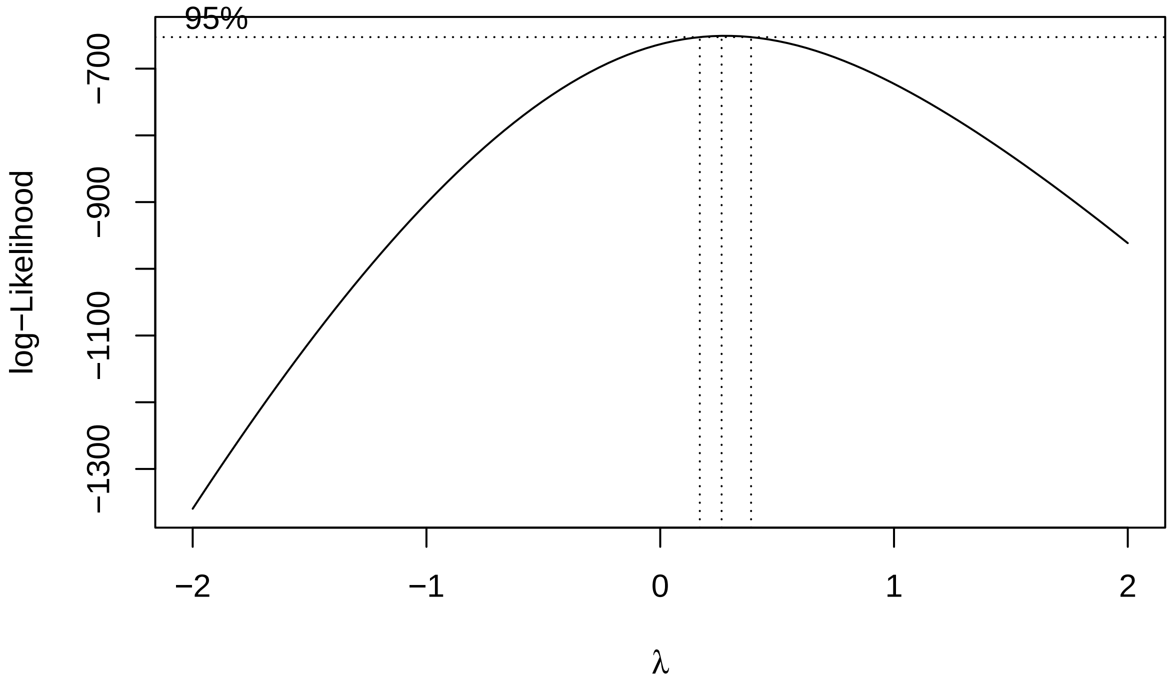
```
##
## Call:
## lm(formula = O3 ~ temp + humidity + ibh, data = ozone)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5291  -3.0137  -0.2249   2.8239  13.9303
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.049e+01  1.616e+00  -6.492 3.16e-10 ***
## temp         3.296e-01  2.109e-02  15.626  < 2e-16 ***
## humidity     7.738e-02  1.339e-02   5.777 1.77e-08 ***
## ibh         -1.004e-03  1.639e-04  -6.130 2.54e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.524 on 326 degrees of freedom
## Multiple R-squared:  0.684,  Adjusted R-squared:  0.6811
## F-statistic: 235.2 on 3 and 326 DF,  p-value: < 2.2e-16
```

Which we can see does not have a great R^2 value and thus, we can attempt a boxcox transformation as follows,

```
require(MASS)
```

```
## Loading required package: MASS
```

```
bc <- boxcox(lmod)
```



```
bc$x[which.max(bc$y)]
```

```
## [1] 0.2626263
```

Which we can see gives us a $\lambda = 0.25$. Thus, refitting our model wiht this transformation, we obtain that

```
lmod <- lm(O3^(0.25) ~ temp + humidity + ibh, data = ozone)
summary(lmod)
```

```
##
## Call:
## lm(formula = O3^(0.25) ~ temp + humidity + ibh, data = ozone)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5433 -0.1084  0.0091  0.1247  0.4933
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.137e-01  6.184e-02  14.774  < 2e-16 ***
## temp         1.307e-02  8.070e-04  16.192  < 2e-16 ***
## humidity     2.977e-03  5.124e-04   5.810 1.48e-08 ***
## ibh         -4.849e-05  6.269e-06  -7.734 1.31e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1731 on 326 degrees of freedom
## Multiple R-squared:  0.7152, Adjusted R-squared:  0.7126
## F-statistic: 272.9 on 3 and 326 DF,  p-value: < 2.2e-16
```

Which we can see improves our R^2 as well as the significance for each of our predictor variables.

Problem 9.4

Use the pressure data to fit a model with pressure as the response and temperature as the predictor using transformations to obtain a good fit.

Answer 9.4

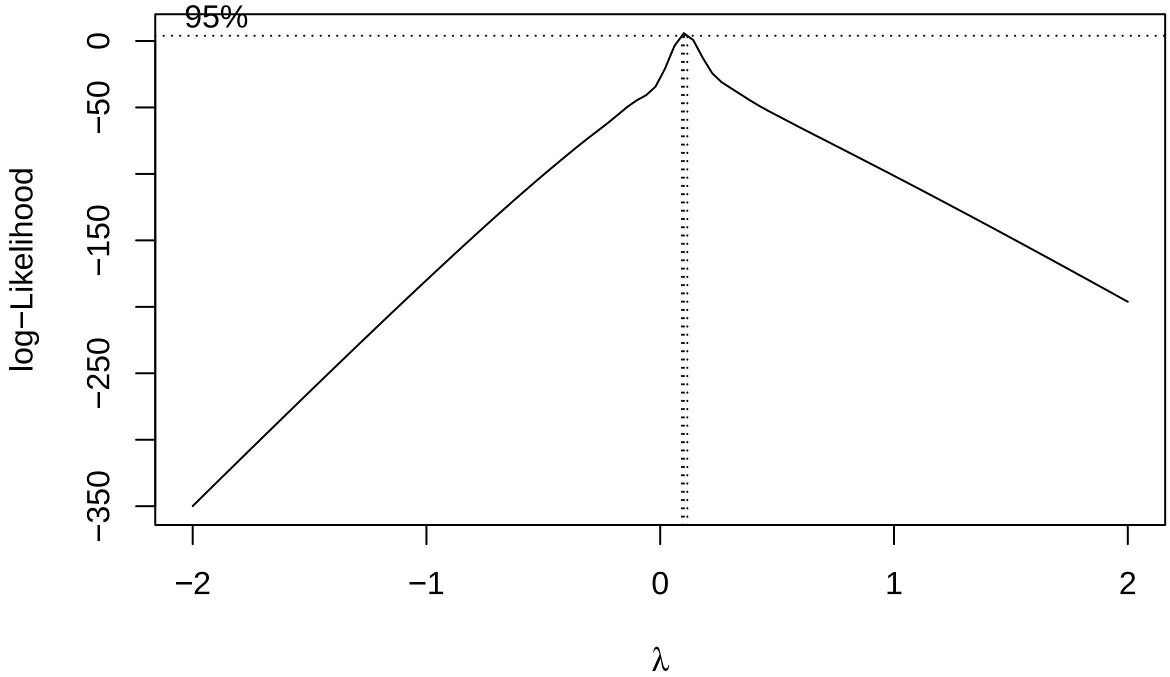
First, we can fit our model as follows,

```
library(faraway)
lmod <- lm(pressure~ temperature, data = pressure)
summary(lmod)
```

```
##
## Call:
## lm(formula = pressure ~ temperature, data = pressure)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158.08 -117.06  -32.84   72.30  409.43
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -147.8989    66.5529  -2.222 0.040124 *
## temperature    1.5124     0.3158   4.788 0.000171 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 150.8 on 17 degrees of freedom
## Multiple R-squared:  0.5742, Adjusted R-squared:  0.5492
## F-statistic: 22.93 on 1 and 17 DF,  p-value: 0.000171
```

Which we can see does not have a great R^2 value and thus, we can attempt a boxcox transformation as follows,

```
require(MASS)
bc <- boxcox(lmod)
```



```
bc$x[which.max(bc$y)]
```

```
## [1] 0.1010101
```

Thus, letting $\lambda = 0$, or in other words, using a log transformation on our response variable, we find that,

```
lmodBoxCox = lm(log(pressure)~temperature, data = pressure)
summary(lmodBoxCox)
```

```
##
## Call:
## lm(formula = log(pressure) ~ temperature, data = pressure)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4491 -0.6876  0.2866  0.8716  1.1365
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.068144   0.483831  -12.54 5.10e-10 ***
## temperature  0.039792   0.002296   17.33 3.07e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.096 on 17 degrees of freedom
## Multiple R-squared:  0.9464, Adjusted R-squared:  0.9433
## F-statistic: 300.3 on 1 and 17 DF,  p-value: 3.07e-12
```

WHich we can see gives us inmproved signficance for each of our predictor variables and reduces our R^2 and our residual standards error, giving us a better for our model.

Problem 9.5

Use transformations to find a good model for volume in terms of girth and height using the trees data

Answer 9.5

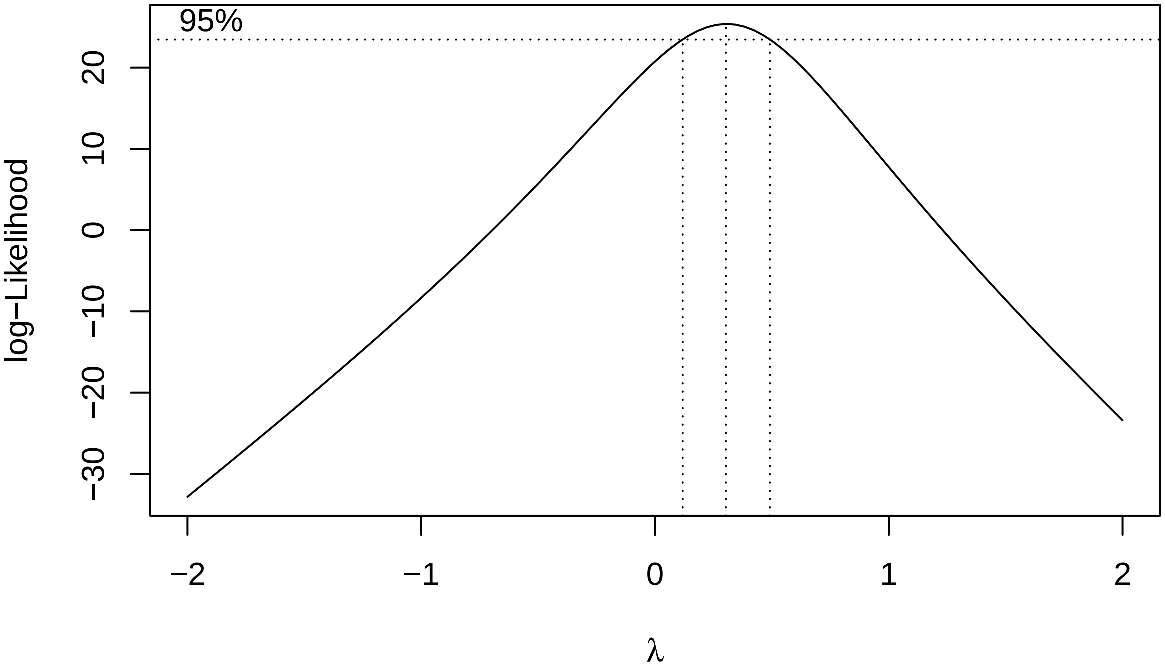
First, we can generate a non-transformed linear model for our data as follows,

```
library(faraway)
?trees
lmod <- lm(Volume ~ Girth + Height, data = trees)
summary(lmod)
```

```
##
## Call:
## lm(formula = Volume ~ Girth + Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4065 -2.6493 -0.2876  2.2003  8.4847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -57.9877     8.6382  -6.713 2.75e-07 ***
## Girth         4.7082     0.2643  17.816  < 2e-16 ***
## Height        0.3393     0.1302   2.607   0.0145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.882 on 28 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9442
## F-statistic:   255 on 2 and 28 DF,  p-value: < 2.2e-16
```

Now, using this model, we can perform a box-cox transformation as follows,

```
require(MASS)
bc <- boxcox(lmod)
```



```
bc$x[which.max(bc$y)]
```

```
## [1] 0.3030303
```

Which gives us a $\lambda = 0.303$. Thus, applying this tranformation onto our response variable, we obtain that,

```
lmod <- lm(Volume^{0.303} ~ Girth + Height, data = trees)
summary(lmod)
```

```
##
## Call:
## lm(formula = Volume^{
##     0.303
## } ~ Girth + Height, data = trees)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.129063 -0.043251 -0.004445  0.056670  0.111648
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.171886   0.151509   1.134    0.266
## Girth       0.124050   0.004635  26.764  < 2e-16 ***
## Height      0.012023   0.002283   5.267 1.34e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06808 on 28 degrees of freedom
## Multiple R-squared:  0.9775, Adjusted R-squared:  0.9759
## F-statistic: 609.6 on 2 and 28 DF,  p-value: < 2.2e-16
```

Which we can see improves our R^2.

Problem 10.1

Use the prostate data with lpsa as the response and the other variables as predictors. Implement the following variable selection methods to determine the "best" model: 1. Backward Elimination 2. AIC 3. Adjusted R^2 4. Mallows C_p ### Answer 10.1 1. First, focusing on backward elimination, we can perform this variable selection method for our "best" model as follows,

```
library(faraway)
lmod <- lm(lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason + pgg45 + lpsa, data = prostate
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 9 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
##     gleason + pgg45 + lpsa, data = prostate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7331 -0.3713 -0.0170  0.4141  1.6381
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.669337   1.296387   0.516  0.60693
## lcavol       0.587022   0.087920   6.677 2.11e-09 ***
## lweight      0.454467   0.170012   2.673  0.00896 **
## age         -0.019637   0.011173  -1.758  0.08229 .
## lbph         0.107054   0.058449   1.832  0.07040 .
## svi          0.766157   0.244309   3.136  0.00233 **
## lcp         -0.105474   0.091013  -1.159  0.24964
## gleason      0.045142   0.157465   0.287  0.77503
## pgg45        0.004525   0.004421   1.024  0.30886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7084 on 88 degrees of freedom
## Multiple R-squared:  0.6548, Adjusted R-squared:  0.6234
## F-statistic: 20.86 on 8 and 88 DF,  p-value: < 2.2e-16
```

Thus, as gleason is th least significant predictor, we can thus remove it from our model. Doing so, we now obtain,

```
lmod <- lm(lpsa ~ lcavol + lweight + age + lbph + svi + lcp + pgg45 + lpsa, data = prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 8 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
##     pgg45 + lpsa, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.73117 -0.38137 -0.01728  0.43364  1.63513
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.953926   0.829439   1.150  0.25319
## lcavol       0.591615   0.086001   6.879 8.07e-10 ***
## lweight      0.448292   0.167771   2.672  0.00897 **
## age         -0.019336   0.011066  -1.747  0.08402 .
## lbph         0.107671   0.058108   1.853  0.06720 .
## svi          0.757734   0.241282   3.140  0.00229 **
## lcp         -0.104482   0.090478  -1.155  0.25127
## pgg45        0.005318   0.003433   1.549  0.12488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7048 on 89 degrees of freedom
## Multiple R-squared:  0.6544, Adjusted R-squared:  0.6273
## F-statistic: 24.08 on 7 and 89 DF,  p-value: < 2.2e-16
```

Now, we can remove lcp from our predictors to obtain,

```
lmod <- lm(lpsa ~ lcavol + lweight + age + lbph + svi + pgg45 + lpsa, data = prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped

## Warning in model.matrix.default(mt, mf, contrasts): problem with term 7 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + pgg45 +
##     lpsa, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77711 -0.41708  0.00002  0.40676  1.59681
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.980085   0.830665   1.180  0.24116
## lcavol       0.545770   0.076431   7.141 2.31e-10 ***
## lweight      0.449450   0.168078   2.674  0.00890 **
## age         -0.017470   0.010967  -1.593  0.11469
## lbph         0.105755   0.058191   1.817  0.07249 .
## svi          0.641666   0.219757   2.920  0.00442 **
## pgg45        0.003528   0.003068   1.150  0.25331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7061 on 90 degrees of freedom
## Multiple R-squared:  0.6493, Adjusted R-squared:  0.6259
## F-statistic: 27.77 on 6 and 90 DF,  p-value: < 2.2e-16
```

Now, we can remove pgg45 to obtain,

```
lmod <- lm(lpsa ~ lcavol + lweight + age + lbph + svi + lpsa, data = prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped

## Warning in model.matrix.default(mt, mf, contrasts): problem with term 6 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lpsa,
##     data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83505 -0.39396  0.00414  0.46336  1.57888
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95100    0.83175   1.143 0.255882
## lcavol       0.56561    0.07459   7.583 2.77e-11 ***
## lweight      0.42369    0.16687   2.539 0.012814 *
## age         -0.01489    0.01075  -1.385 0.169528
## lbph         0.11184    0.05805   1.927 0.057160 .
## svi          0.72095    0.20902   3.449 0.000854 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7073 on 91 degrees of freedom
## Multiple R-squared:  0.6441, Adjusted R-squared:  0.6245
## F-statistic: 32.94 on 5 and 91 DF,  p-value: < 2.2e-16
```

Now, removing age, we obtain,

```
lmod <- lm(lpsa ~ lcavol + lweight + lbph + svi + lpsa, data = prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
```

```
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 5 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + lbph + svi + lpsa, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.82653 -0.42270  0.04362  0.47041  1.48530
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14554    0.59747   0.244  0.80809
## lcavol       0.54960    0.07406   7.422 5.64e-11 ***
## lweight      0.39088    0.16600   2.355  0.02067 *
## lbph         0.09009    0.05617   1.604  0.11213
## svi          0.71174    0.20996   3.390  0.00103 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7108 on 92 degrees of freedom
## Multiple R-squared:  0.6366, Adjusted R-squared:  0.6208
## F-statistic: 40.29 on 4 and 92 DF,  p-value: < 2.2e-16
```

From which, we can remove lbph to obtain,

```
lmod <- lm(lpsa ~ lcavol + lweight + svi + lpsa, data = prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on the
## right-hand side and was dropped
## Warning in model.matrix.default(mt, mf, contrasts): problem with term 4 in
## model.matrix: no columns are assigned
```

```
summary(lmod)
```

```
##
## Call:
## lm(formula = lpsa ~ lcavol + lweight + svi + lpsa, data = prostate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72964 -0.45764  0.02812  0.46403  1.57013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26809    0.54350  -0.493  0.62298
## lcavol       0.55164    0.07467   7.388  6.3e-11 ***
## lweight      0.50854    0.15017   3.386  0.00104 **
## svi          0.66616    0.20978   3.176  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7168 on 93 degrees of freedom
## Multiple R-squared:  0.6264, Adjusted R-squared:  0.6144
## F-statistic: 51.99 on 3 and 93 DF,  p-value: < 2.2e-16
```

Which gives us all of our desired significance levels for our predictor variables.

2. Now, performing the AIC method of searching for different predictor variables, we obtain that,

```
require(leaps)
```

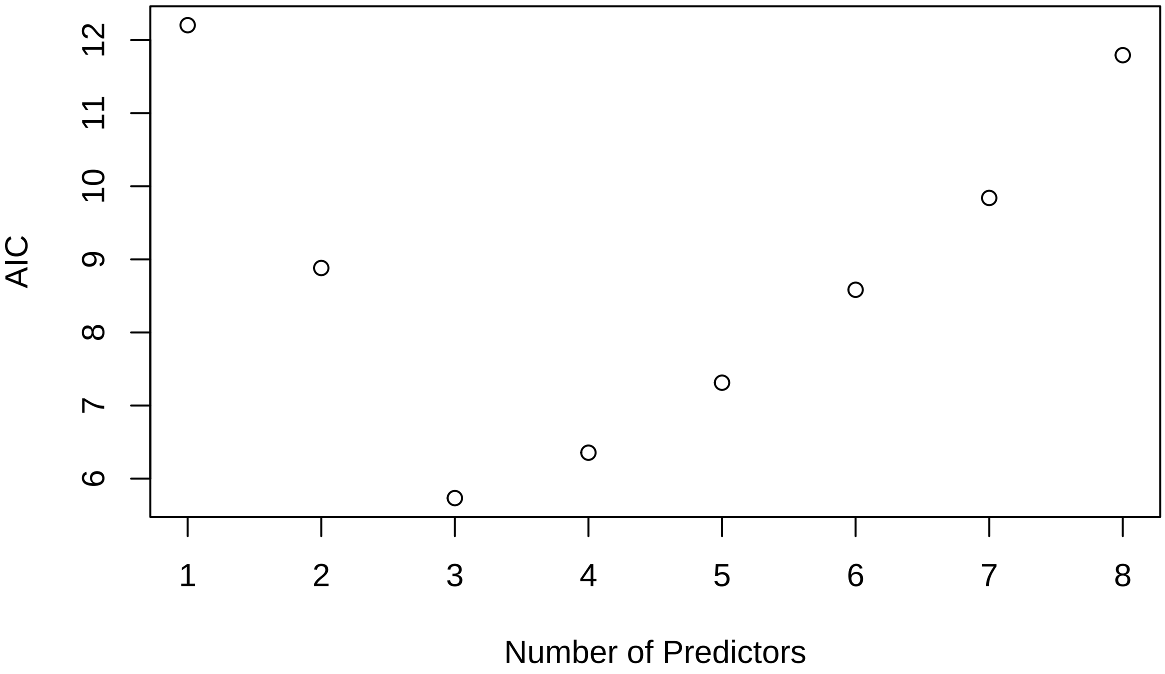
```
## Loading required package: leaps
```

```
b <- regsubsets(lpsa~.,data=prostate)
rs <- summary(b)
rs$which
```

```
##   (Intercept) lcavol lweight   age  lbph   svi   lcp gleason pgg45
## 1        TRUE   TRUE   FALSE FALSE FALSE FALSE FALSE   FALSE FALSE
## 2        TRUE   TRUE    TRUE FALSE FALSE FALSE FALSE   FALSE FALSE
## 3        TRUE   TRUE    TRUE FALSE FALSE  TRUE FALSE   FALSE FALSE
## 4        TRUE   TRUE    TRUE FALSE  TRUE  TRUE FALSE   FALSE FALSE
## 5        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE FALSE
## 6        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE FALSE   FALSE  TRUE
## 7        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE   FALSE  TRUE
## 8        TRUE   TRUE    TRUE  TRUE  TRUE  TRUE  TRUE    TRUE  TRUE
```

Which tells us that lcavol is our most significant predictor. Now, we can find what number of predictors we want as follows,

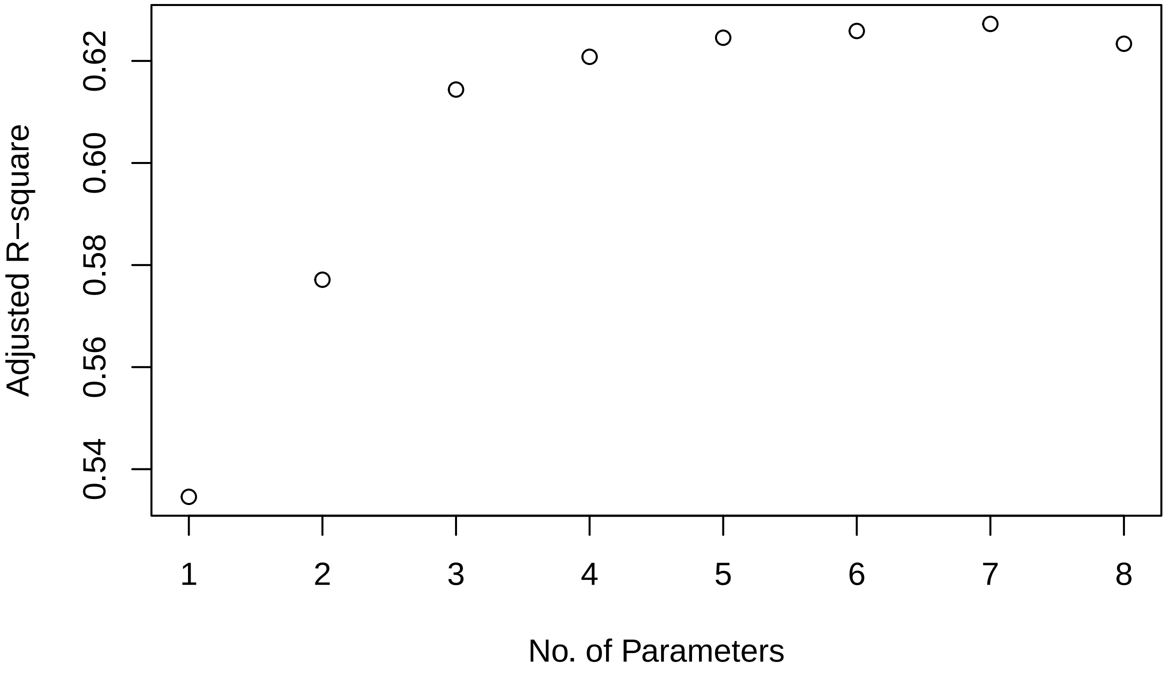
```
AIC <- 50*log(rs$rss/50) + (2:9)*2
plot(AIC ~ I(1:8), ylab="AIC", xlab="Number of Predictors")
```



Wwhich we can see that our best number of predictors is 2, and thus, our best set of predictor variables is lcavol, lweight, and svi which agrees with out backwards elimination method.

3. Now, focusing on the R^2 statistic, we can see that we obtain,

```
plot(1:8,rs$adjr2,xlab="No. of Parameters",ylab="Adjusted R-square")
```



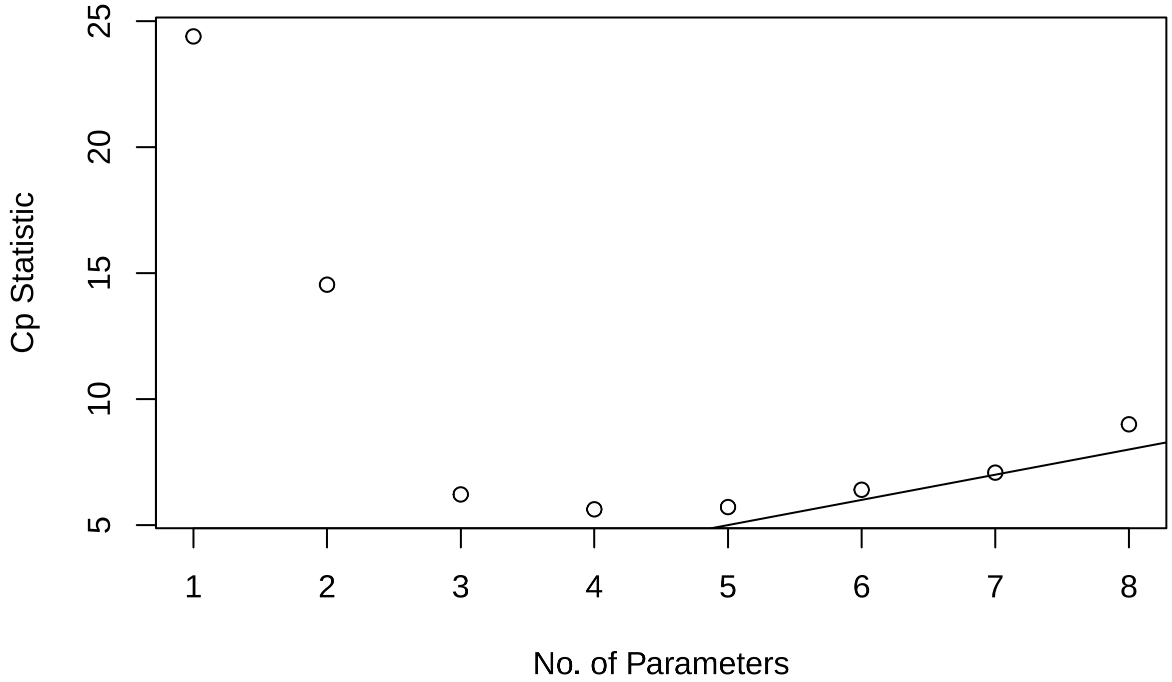
```
which.max(rs$adjr2)
```

```
## [1] 7
```

Which we can see that our 7 parameter model gives us the highest adjusted R^2, which is the set of predictor variables of all remaining variables without gleason.

4. Now, focusing on the C_p criterion for model selection, we can find our best model as follows,

```
plot(1:8,rs$cp,xlab="No. of Parameters",ylab="Cp Statistic")
abline(0,1)
```



From which, we can see that our set of 7 predictors is the best for our criteria, and thus, we will select the set, lcavol, lweight, age, lbph, svi, lcph, and pgg45.

Problem 10.3

Using the divusa data set with divorce as the response and the other variables as predictors, repeat the work of the first question.

Answer 10.3

1. Beginning with backward elimination, we will begin with our set of all predictors as follows,

```
library(faraway)
?divusa
lmod <- lm(divorce ~ year + unemployed + femlab + marriage + birth + military, data = divusa)
summary(lmod)
```

```
##
## Call:
## lm(formula = divorce ~ year + unemployed + femlab + marriage +
##     birth + military, data = divusa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9087 -0.9212 -0.0935  0.7447  3.4689
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 380.14761   99.20371   3.832 0.000274 ***
## year         -0.20312    0.05333  -3.809 0.000297 ***
## unemployed   -0.04933    0.05378  -0.917 0.362171    
## femlab        0.80793    0.11487   7.033 1.09e-09 ***
## marriage      0.14977    0.02382   6.287 2.42e-08 ***
## birth        -0.11695    0.01470  -7.957 2.19e-11 ***
## military     -0.04276    0.01372  -3.117 0.002652 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.513 on 70 degrees of freedom
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.9288 
## F-statistic: 166.2 on 6 and 70 DF,  p-value: < 2.2e-16
```

Now, we can removed unemployed from our predictor variables to obtain,

```
lmod <- lm(divorce ~ year + femlab + marriage + birth + military, data = divusa)
summary(lmod)
```

```
## 
## Call:
## lm(formula = divorce ~ year + femlab + marriage + birth + military, 
##     data = divusa)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.7586 -1.0494 -0.0424  0.7201  3.3075 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 405.61670   95.13189   4.264 6.09e-05 ***
## year         -0.21790    0.05078  -4.291 5.52e-05 ***
## femlab        0.85480    0.10276   8.318 4.29e-12 ***
## marriage      0.15934    0.02140   7.447 1.76e-10 ***
## birth        -0.11012    0.01266  -8.700 8.43e-13 ***
## military     -0.04120    0.01360  -3.030  0.00341 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.511 on 71 degrees of freedom
## Multiple R-squared:  0.9336, Adjusted R-squared:  0.929 
## F-statistic: 199.7 on 5 and 71 DF,  p-value: < 2.2e-16
```

At this point, our set of predictor variables appear to all be relevant, and thus, we can say that the best set of predictor variables for our model appears to be year, femlab, marriage, birth, and military.

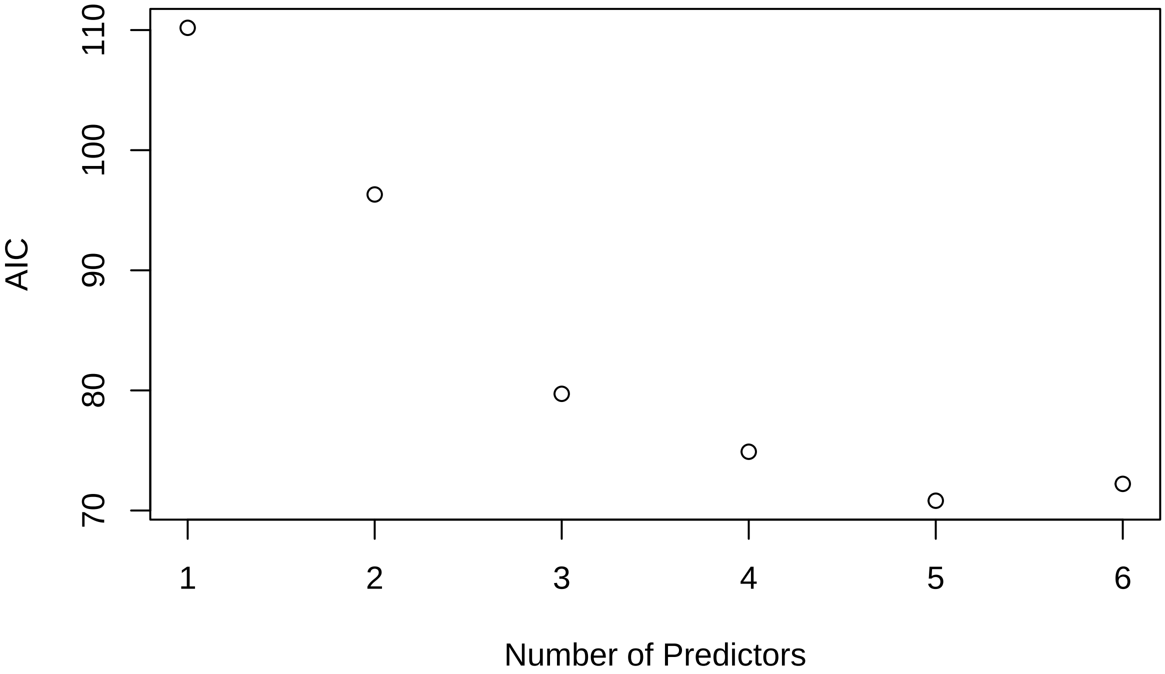
2. Now, performing the AIC method of searching for different predictor variables, we obtain that,

```
require(leaps)
b <- regsubsets(divorce~.,data=divusa)
rs <- summary(b)
rs$which
```

```
##   (Intercept)  year unemployed femlab marriage birth military
## 1        TRUE FALSE      FALSE   TRUE    FALSE FALSE    FALSE
## 2        TRUE FALSE      FALSE   TRUE    FALSE  TRUE    FALSE
## 3        TRUE FALSE      FALSE   TRUE     TRUE  TRUE    FALSE
## 4        TRUE  TRUE      FALSE   TRUE     TRUE  TRUE    FALSE
## 5        TRUE  TRUE      FALSE   TRUE     TRUE  TRUE     TRUE
## 6        TRUE  TRUE       TRUE   TRUE     TRUE  TRUE     TRUE
```

Which tells us that femlab is our most significant predictor. Now, we can find what number of predictors we want as follows,

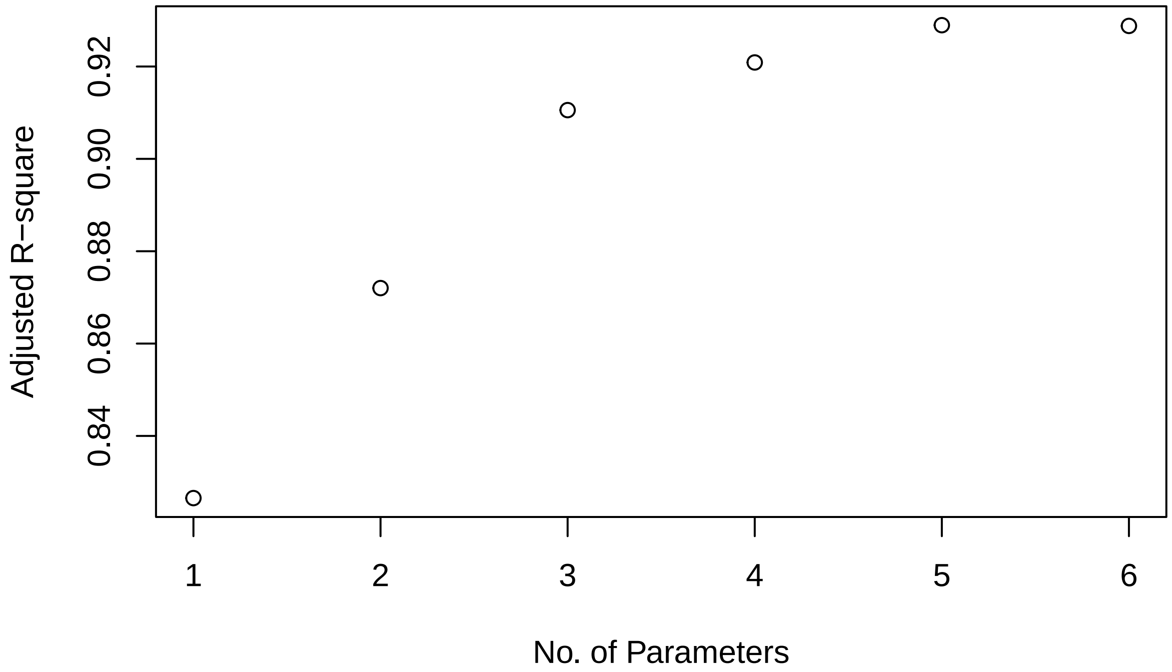
```
AIC <- 50*log(rs$rss/50) + (2:7)*2
plot(AIC ~ I(1:6), ylab="AIC", xlab="Number of Predictors")
```



From which we can see that our best model in terms of AIC is our set of 5 predictors, or the set year, femlab, marriage, birth, and military.

3. Now, we can use our adjusted R^2 statistic to determine our best model as follows, Now, focusing on the R^2 statistic, we can see that we obtain,

```
plot(1:6,rs$adjr2,xlab="No. of Parameters",ylab="Adjusted R-square")
```



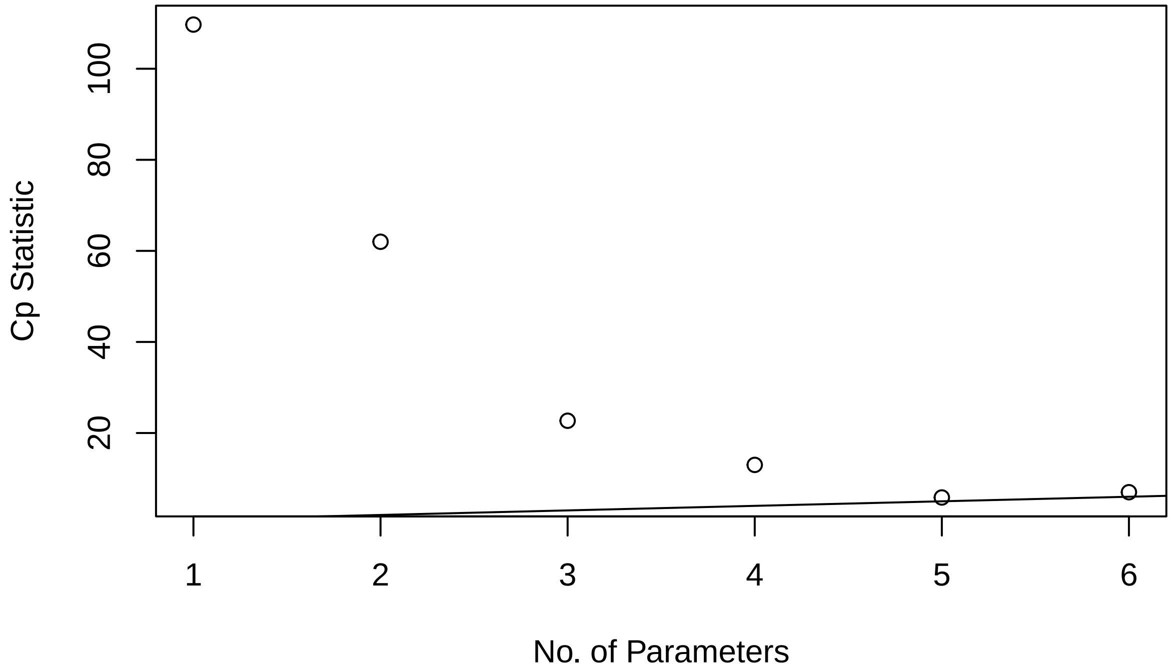
```
which.max(rs$adjr2)
```

```
## [1] 5
```

From which we can see that our best model in terms of adjusted R^2 is our set of 5 predictors, or the set year, femlab, marriage, birth, and military.

4. Now focusing on C_p, we can see that we obtain,

```
plot(1:6,rs$cp,xlab="No. of Parameters",ylab="Cp Statistic")
abline(0,1)
```



Which as both 5 and 6 are along the line, we will defer for 5 and thus use the set of predictor variables: year, femlab, marriage, birth, and military.