Homework 10

Jun Ryu, UID: 605574052

2023-06-09

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(leaps)
```

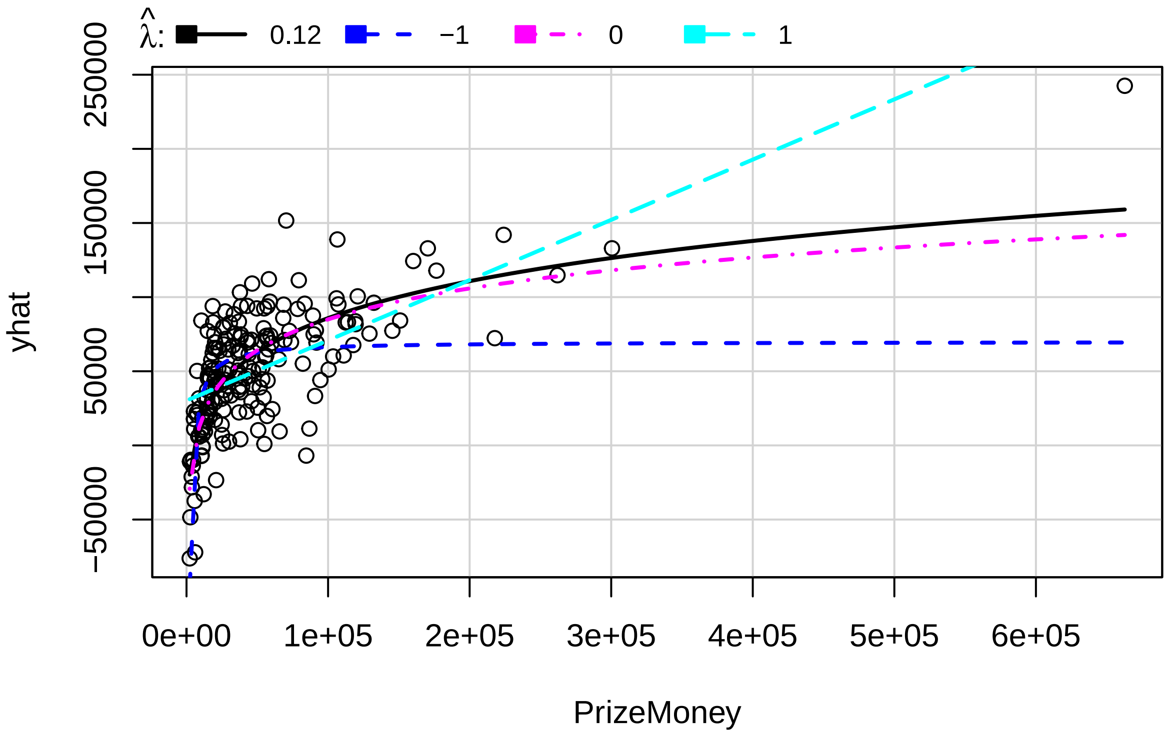
Textbook Exercise 6.5

```
pga <- read.csv("pgatour2006-3.csv")
head(pga)
```

```
##                  Name TigerWoods PrizeMoney AveDrivingDistance DrivingAccuracy
## 1   Aaron Baddeley          0      60661              288.3           60.73
## 2       Adam Scott          0     262045              301.1           62.00
## 3      Alex Aragon          0       3635              302.6           51.12
## 4       Alex Cejka          0      17516              288.8           66.40
## 5      Arjun Atwal          0      16683              287.7           63.24
## 6 Arron Oberholser          0     107294              285.0           62.53
##     GIR PuttingAverage BirdieConversion SandSaves Scrambling BounceBack
## 1 58.26          1.745            31.36     54.80      59.37      19.30
## 2 69.12          1.767            30.39     53.61      57.94      19.35
## 3 59.11          1.787            29.89     37.93      50.78      16.80
## 4 67.70          1.777            29.33     45.13      54.82      17.05
## 5 64.04          1.761            29.32     52.44      57.07      18.21
## 6 69.27          1.775            29.20     47.20      57.67      20.00
##   PuttsPerRound
## 1         27.96
## 2         29.28
## 3         29.20
## 4         29.46
## 5         28.93
## 6         29.56
```

a)

```
model <- lm(PrizeMoney ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+
              SandSaves+Scrambling+PuttsPerRound, data = pga)
invResPlot(model)
```




```
##        lambda          RSS
## 1  0.1191664 153353617043
## 2 -1.0000000 202266980718
## 3  0.0000000 154049980760
## 4  1.0000000 192096985076
```

```
summary(powerTransform(model))
```

```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.0337           0      -0.0701       0.1376
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                              LRT df    pval
## LR test, lambda = (0) 0.4054804  1 0.52427
##
## Likelihood ratio test that no transformation is needed
##                             LRT df       pval
## LR test, lambda = (1) 335.2384  1 < 2.22e-16
```

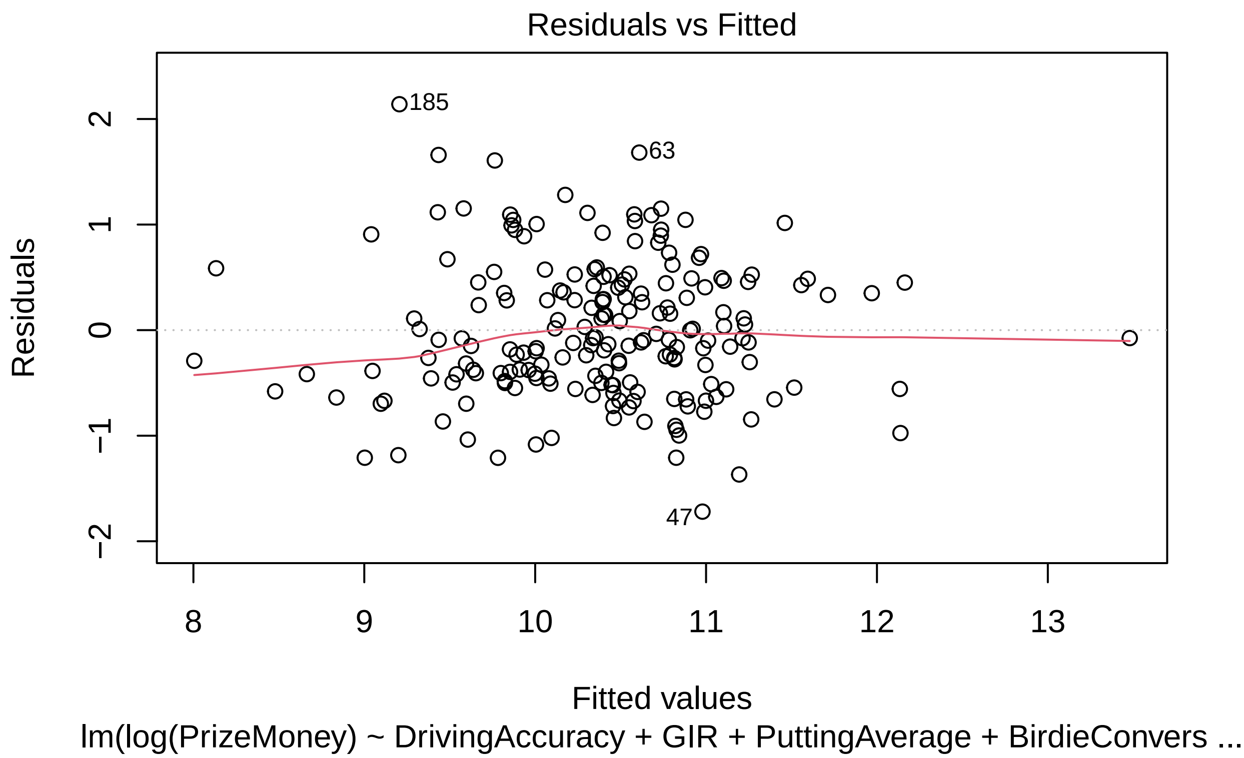
Looking at the inverse response plot, the plot suggests that a log transform would be appropriate as its plot using the "optimal" value of 0.12 is not too different from the log transform graph (lambda = 0). Moreover, checking with the power transform, because the p-value is high for the hypothesis test when lambda = 0, we do not reject the null hypothesis and conclude that a log transform of the Y variable would be appropriate.

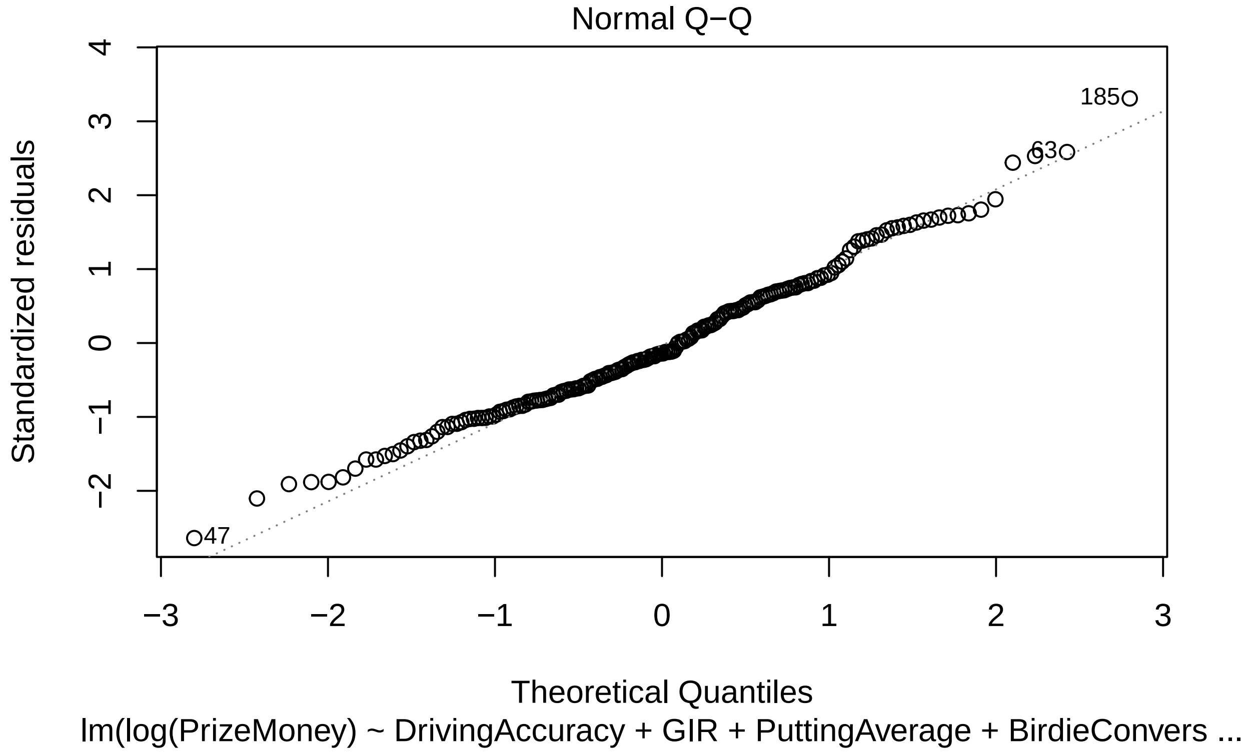
b)

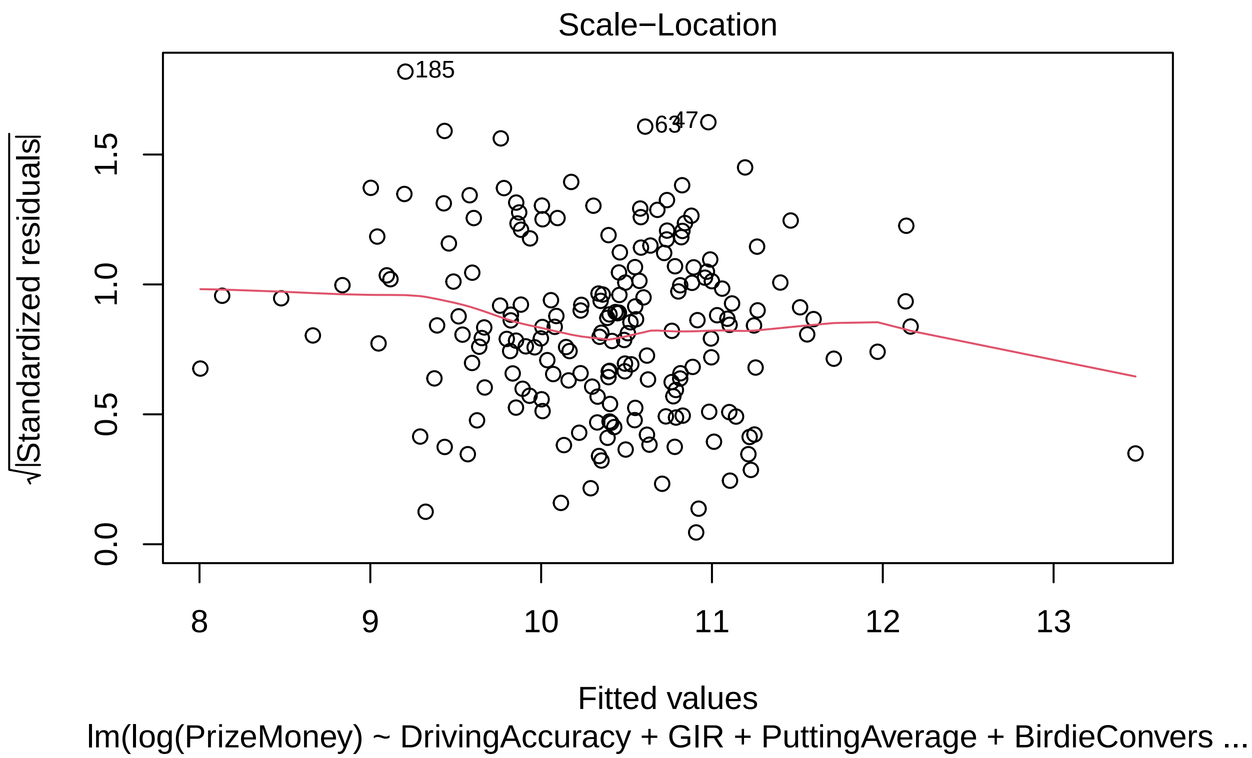
```r
# using a log transform...
log_model <- lm(log(PrizeMoney) ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+
              SandSaves+Scrambling+PuttsPerRound, data = pga)
summary(log_model)
```

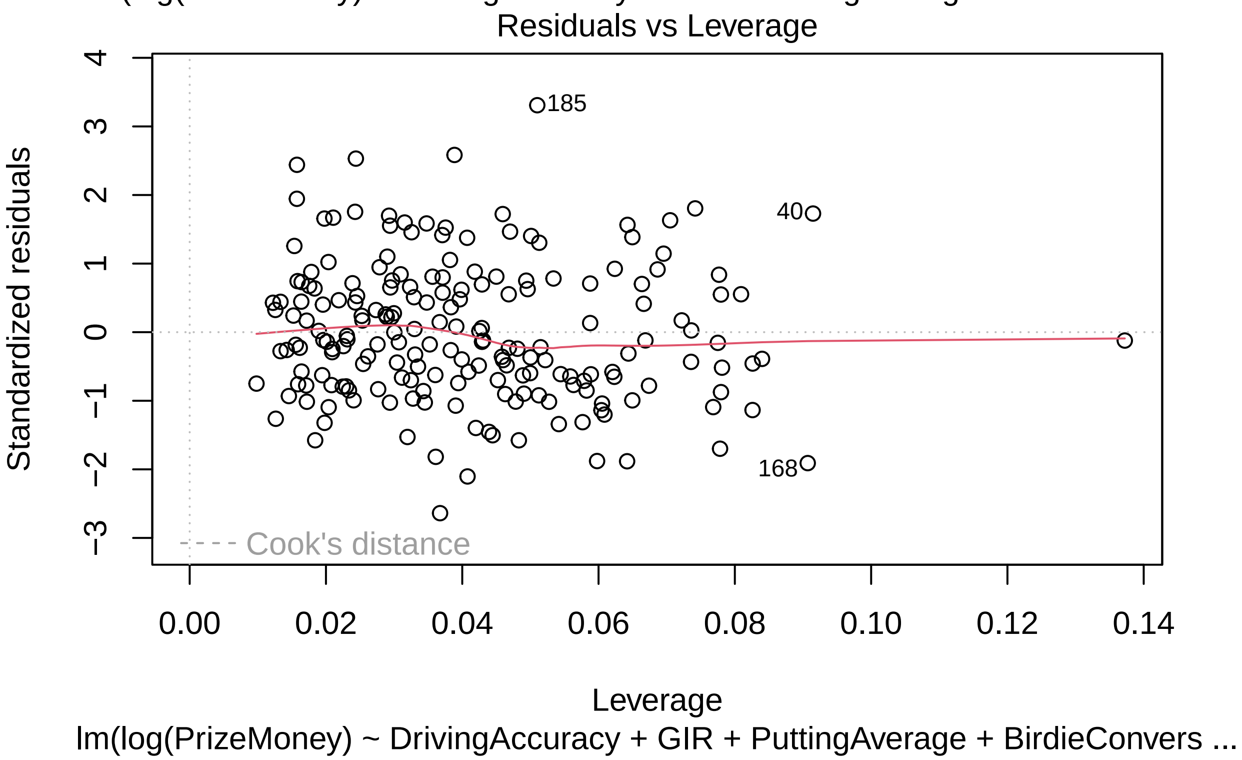
```
##
## Call:
## lm(formula = log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage +
##     BirdieConversion + SandSaves + Scrambling + PuttsPerRound,
##     data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71949 -0.48608 -0.09172  0.44561  2.14013
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.194300   7.777129   0.025 0.980095
## DrivingAccuracy  -0.003530   0.011773  -0.300 0.764636
## GIR               0.199311   0.043817   4.549 9.66e-06 ***
## PuttingAverage   -0.466304   6.905698  -0.068 0.946236
## BirdieConversion  0.157341   0.040378   3.897 0.000136 ***
## SandSaves         0.015174   0.009862   1.539 0.125551
## Scrambling        0.051514   0.031788   1.621 0.106788
## PuttsPerRound    -0.343131   0.473549  -0.725 0.469601
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6639 on 188 degrees of freedom
## Multiple R-squared:  0.5577, Adjusted R-squared:  0.5412
## F-statistic: 33.87 on 7 and 188 DF,  p-value: < 2.2e-16
```

```r
plot(log_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConvers ...
185
63
47
2
1
0
−1
−2
8
9
10
11
12
13


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConvers ...
185
63
47
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3





This improved model seems to be valid after analyzing the diagnostic plots. There's no visible trend in the residuals vs fitted and the scale-location plot, indicating linearity and constant variance. The qq-plot is also pretty straight, indicating normality of our model.

c)

Some points that should be investigated include outliers. In our case, observation 185 seems to be a potential outlier because not only does it have the highest residual in the residuals vs fitted plot, but the point seems to be a bad leverage point based on the residuals vs leverage plot. On the same note, observations 47 and 63 could be investigated due to their high residual values.

d)

A weakness of our model is collinearity. Observing the vif values:

```r
vif(log_model)
```

```
##  DrivingAccuracy              GIR   PuttingAverage BirdieConversion
##         1.796616         6.294969        12.900789         3.511898
##        SandSaves       Scrambling    PuttsPerRound
##         1.461506         4.470203        19.355667
```

We have that 3 of these variables have a vif value greater than 5, which is problematic. This will decrease t-statistics and thus, inflate the respective p-values.

e)

First of all, due to the issue with collinearity, we have to keep in mind that some of these p-values are inflated and thus, the t-statistics are not accurately represented. But, even beyond that, trying to remove all predictors in a single step is not a good idea as removing one variable could actually change another variable's t-statistic, perhaps making it go from insignificant to significant.

Textbook Exercise 7.3

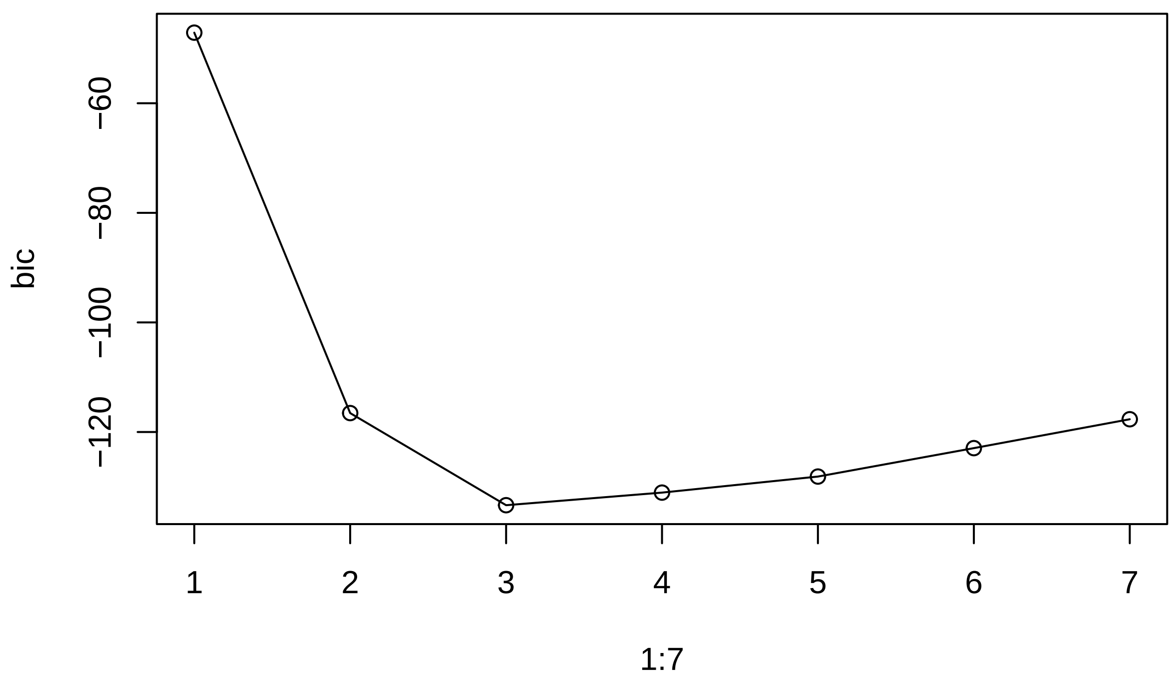
a)

```r
bestss <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+
                          SandSaves+Scrambling+PuttsPerRound, data = pga, nvmax = 7)
summary(bestss)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga, nvmax = 7)
## 7 Variables  (and intercept)
##                  Forced in Forced out
## DrivingAccuracy      FALSE      FALSE
## GIR                  FALSE      FALSE
## PuttingAverage       FALSE      FALSE
## BirdieConversion     FALSE      FALSE
## SandSaves            FALSE      FALSE
## Scrambling           FALSE      FALSE
## PuttsPerRound        FALSE      FALSE
```

```
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##          DrivingAccuracy GIR PuttingAverage BirdieConversion SandSaves
## 1  ( 1 ) " "             "*" " "            " "              " "
## 2  ( 1 ) " "             "*" " "            " "              " "
## 3  ( 1 ) " "             "*" " "            "*"              " "
## 4  ( 1 ) " "             "*" " "            "*"              "*"
## 5  ( 1 ) " "             "*" " "            "*"              "*"
## 6  ( 1 ) "*"             "*" " "            "*"              "*"
## 7  ( 1 ) "*"             "*" "*"            "*"              "*"
##          Scrambling PuttsPerRound
## 1  ( 1 ) " "        " "
## 2  ( 1 ) " "        "*"
## 3  ( 1 ) "*"        " "
## 4  ( 1 ) "*"        " "
## 5  ( 1 ) "*"        "*"
## 6  ( 1 ) "*"        "*"
## 7  ( 1 ) "*"        "*"
```

```r
# first we use the bic values:
bic <- summary(bestss)$bic
plot(1:7, bic)
lines(1:7, bic)
```

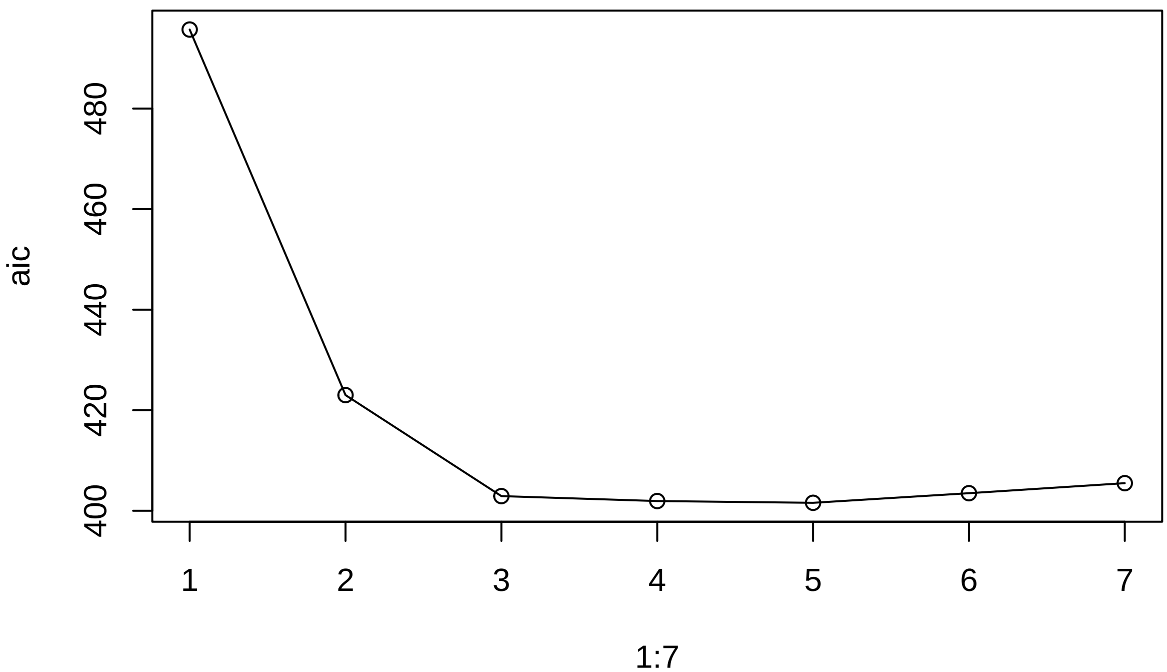


The 3-variable model has the lowest BIC, which is the model that includes the variables GIR, BirdieConversion, and Scrambling. Now we will manually compute the AIC values for comparison:

```r
m1 <- AIC(lm(log(PrizeMoney) ~ GIR, data = pga))
m2 <- AIC(lm(log(PrizeMoney) ~ GIR+PuttsPerRound, data = pga))
m3 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+Scrambling, data = pga))
m4 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling, data = pga))
m5 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling+PuttsPerRound, data = pga))
```

```r
m6 <- AIC(lm(log(PrizeMoney) ~ DrivingAccuracy+GIR+BirdieConversion+SandSaves+
          Scrambling+PuttsPerRound, data = pga))
m7 <- AIC(log_model)

plot(1:7, c(m1,m2,m3,m4,m5,m6,m7), ylab = "aic")
lines(1:7, c(m1,m2,m3,m4,m5,m6,m7))
```



The 3, 4, 5-variable model seem awfully close in their AIC values.

```r
c(m3,m4,m5)
```

```
## [1] 402.9131 401.9329 401.5823
```

We see that the 5-variable model has the lowest AIC values, which is the model that includes the variables GIR, BirdieConversion, SandSaves, Scrambling, and PuttsPerRound.

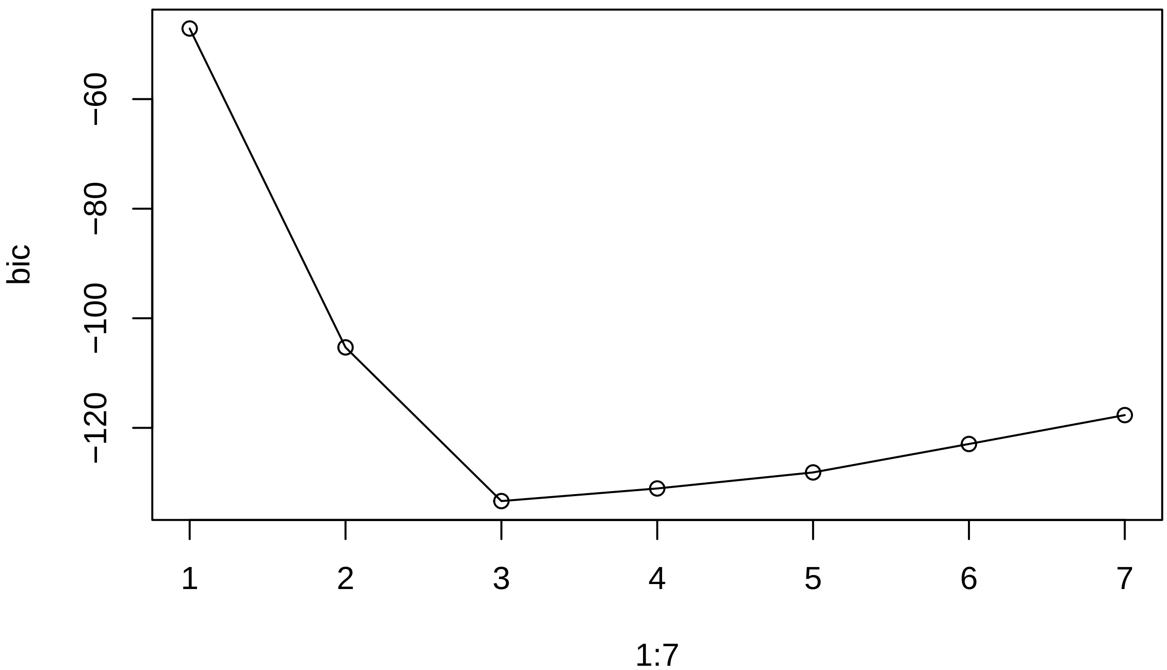
b)

```r
# using backward selection:
bestss <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+
                         SandSaves+Scrambling+PuttsPerRound, data = pga, nvmax = 7,
                     method = "backward")
summary(bestss)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga, nvmax = 7, method = "backward")
## 7 Variables  (and intercept)
```

```
##                   Forced in Forced out
## DrivingAccuracy      FALSE      FALSE
## GIR                  FALSE      FALSE
## PuttingAverage       FALSE      FALSE
## BirdieConversion     FALSE      FALSE
## SandSaves            FALSE      FALSE
## Scrambling           FALSE      FALSE
## PuttsPerRound        FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: backward
##          DrivingAccuracy GIR PuttingAverage BirdieConversion SandSaves
## 1  ( 1 ) " "             "*" " "            " "              " "
## 2  ( 1 ) " "             "*" " "            "*"              " "
## 3  ( 1 ) " "             "*" " "            "*"              " "
## 4  ( 1 ) " "             "*" " "            "*"              "*"
## 5  ( 1 ) " "             "*" " "            "*"              "*"
## 6  ( 1 ) "*"             "*" " "            "*"              "*"
## 7  ( 1 ) "*"             "*" "*"            "*"              "*"
##          Scrambling PuttsPerRound
## 1  ( 1 ) " "        " "
## 2  ( 1 ) " "        " "
## 3  ( 1 ) "*"        " "
## 4  ( 1 ) "*"        " "
## 5  ( 1 ) "*"        "*"
## 6  ( 1 ) "*"        "*"
## 7  ( 1 ) "*"        "*"
```

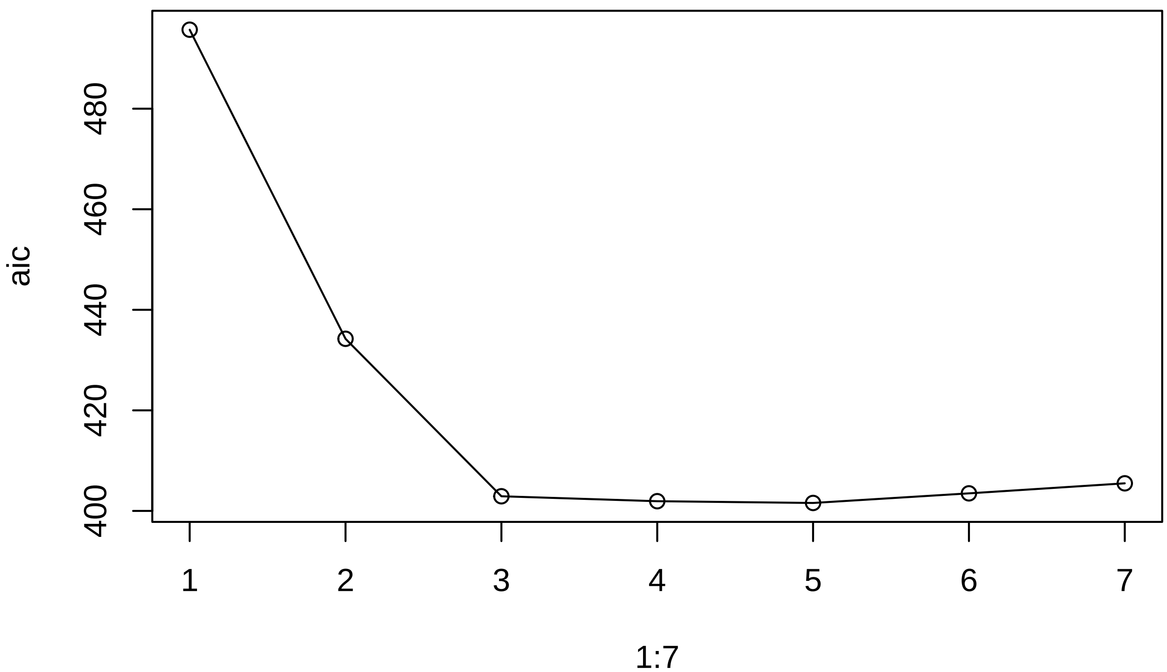
```r
bic <- summary(bestss)$bic
plot(1:7, bic)
lines(1:7, bic)
```



The 3-variable model has the lowest BIC, the same model as the one in part a). Now the AIC values for backward selection:

```r
m1 <- AIC(lm(log(PrizeMoney) ~ GIR, data = pga))
m2 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion, data = pga))
m3 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+Scrambling, data = pga))
m4 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling, data = pga))
m5 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling+PuttsPerRound, data = pga))
m6 <- AIC(lm(log(PrizeMoney) ~ DrivingAccuracy+GIR+BirdieConversion+SandSaves+
           Scrambling+PuttsPerRound, data = pga))
m7 <- AIC(log_model)

plot(1:7, c(m1,m2,m3,m4,m5,m6,m7), ylab = "aic")
lines(1:7, c(m1,m2,m3,m4,m5,m6,m7))
```



Again, we observe the same result as in part a). Since the variables we used for model 3, 4 and 5 did not change with backward selection, model 5 will still have the lowest AIC value.

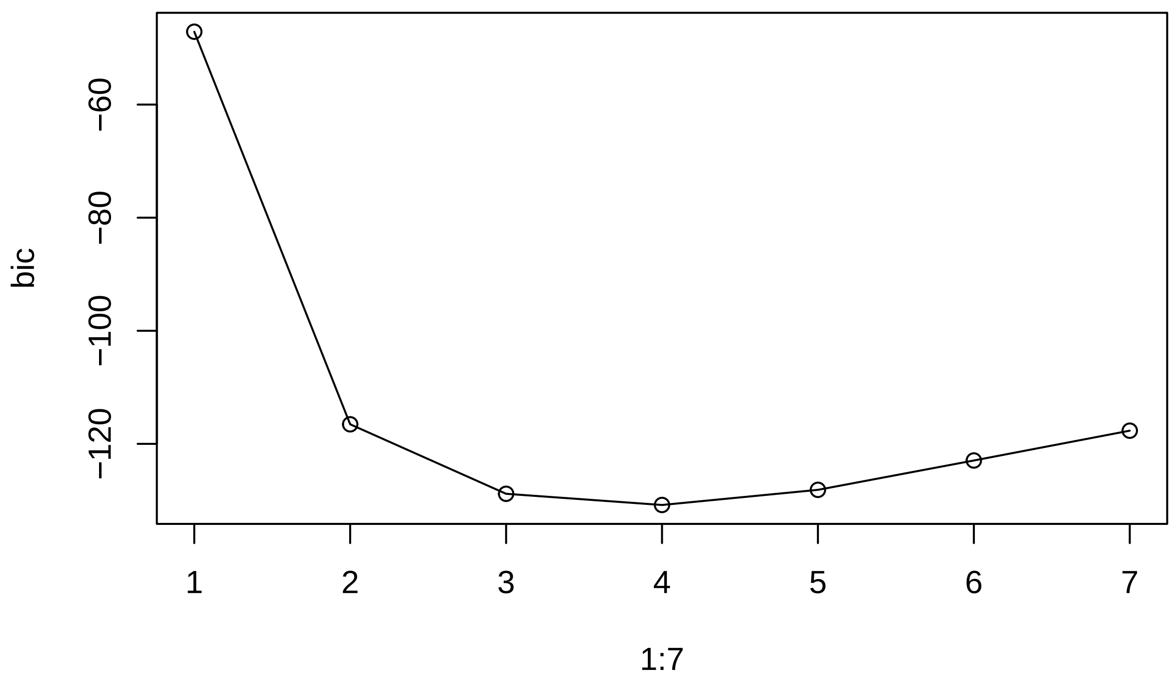
c)

```r
# using forward selection:
bestss <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy+GIR+PuttingAverage+BirdieConversion+
                       SandSaves+Scrambling+PuttsPerRound, data = pga, nvmax = 7,
                     method = "forward")
summary(bestss)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga, nvmax = 7, method = "forward")
## 7 Variables  (and intercept)
##                  Forced in Forced out
## DrivingAccuracy      FALSE      FALSE
```

```
## GIR                    FALSE      FALSE
## PuttingAverage         FALSE      FALSE
## BirdieConversion       FALSE      FALSE
## SandSaves              FALSE      FALSE
## Scrambling             FALSE      FALSE
## PuttsPerRound          FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: forward
##          DrivingAccuracy GIR PuttingAverage BirdieConversion SandSaves
## 1  ( 1 ) " "             "*" " "            " "              " "
## 2  ( 1 ) " "             "*" " "            " "              " "
## 3  ( 1 ) " "             "*" " "            "*"              " "
## 4  ( 1 ) " "             "*" " "            "*"              " "
## 5  ( 1 ) " "             "*" " "            "*"              "*"
## 6  ( 1 ) "*"             "*" " "            "*"              "*"
## 7  ( 1 ) "*"             "*" "*"            "*"              "*"
##          Scrambling PuttsPerRound
## 1  ( 1 ) " "        " "
## 2  ( 1 ) " "        "*"
## 3  ( 1 ) " "        "*"
## 4  ( 1 ) "*"        "*"
## 5  ( 1 ) "*"        "*"
## 6  ( 1 ) "*"        "*"
## 7  ( 1 ) "*"        "*"
```

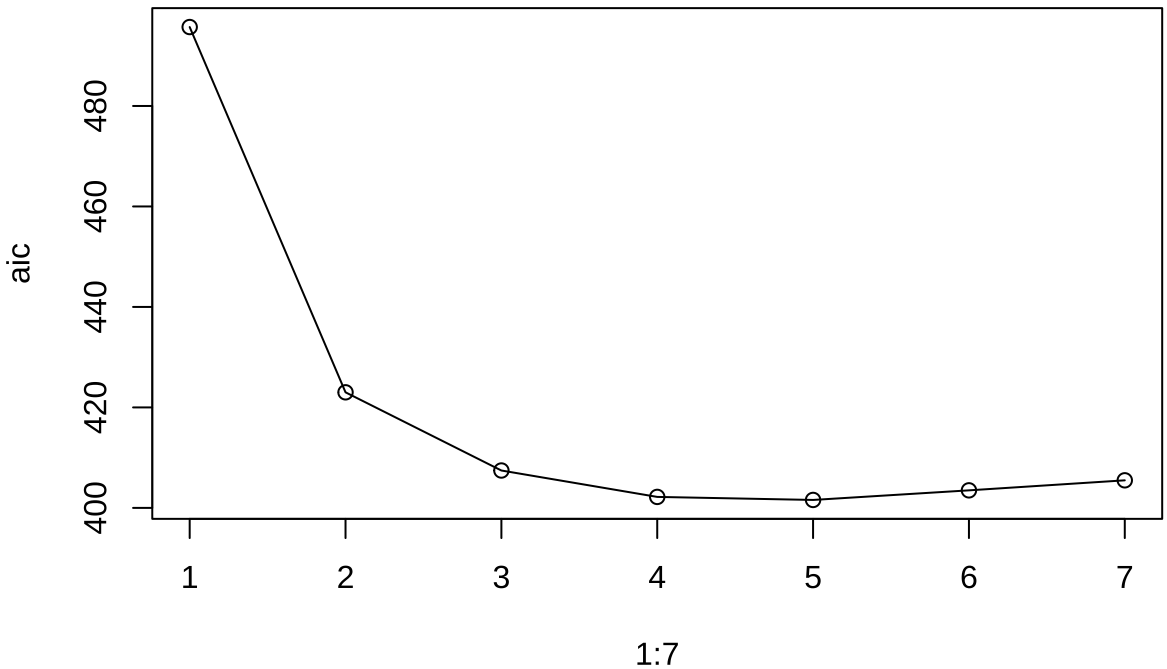
```r
bic <- summary(bestss)$bic
plot(1:7, bic)
lines(1:7, bic)
```



Now, the 4-variable model has the lowest BIC value. This model includes the variables GIR, BirdieConversion, Scrambling, and PuttsPerRound.

```r
m1 <- AIC(lm(log(PrizeMoney) ~ GIR, data = pga))
m2 <- AIC(lm(log(PrizeMoney) ~ GIR+PuttsPerRound, data = pga))
m3 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+PuttsPerRound, data = pga))
m4 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+Scrambling+PuttsPerRound, data = pga))
m5 <- AIC(lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling+PuttsPerRound, data = pga))
m6 <- AIC(lm(log(PrizeMoney) ~ DrivingAccuracy+GIR+BirdieConversion+SandSaves+
           Scrambling+PuttsPerRound, data = pga))
m7 <- AIC(log_model)

plot(1:7, c(m1,m2,m3,m4,m5,m6,m7), ylab = "aic")
lines(1:7, c(m1,m2,m3,m4,m5,m6,m7))
```




```r
c(m3,m4,m5)
```

```
## [1] 407.4398 402.1839 401.5823
```

Here, the same 5-variable model from the previous two parts has the lowest AIC value.

d)

Professor told us that we could skip this portion.

e)

The final recommended model is the 5-variable model with variables GIR, BirdieConversion, SandSaves, Scrambling, and PuttsPerRound because this model had the lowest AIC value across all three methods (exhaustive, backward, and forward).

f)

```r
rec_model <- lm(log(PrizeMoney) ~ GIR+BirdieConversion+SandSaves+Scrambling+PuttsPerRound, data = pga)
summary(rec_model)
```

```
##
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves +
##     Scrambling + PuttsPerRound, data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71291 -0.48168 -0.09097  0.44843  2.15763
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.583181   7.158721  -0.081   0.9352
## GIR               0.197022   0.028711   6.862 9.31e-11 ***
## BirdieConversion  0.162752   0.032672   4.981 1.41e-06 ***
## SandSaves         0.015524   0.009743   1.593   0.1127
## Scrambling        0.049635   0.024738   2.006   0.0462 *
## PuttsPerRound    -0.349738   0.230995  -1.514   0.1317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6606 on 190 degrees of freedom
## Multiple R-squared:  0.5575, Adjusted R-squared:  0.5459
## F-statistic: 47.88 on 5 and 190 DF,  p-value: < 2.2e-16
```

Here, the intercept represents the average prize value when all other coefficients are fixed to 0. For the slope estimates for each variable, they represent the average increase/decrease in percentage of the prize value when that particular variable is increased by one. It is important to be cautious when taking these results literally because this model's adjusted R-squared value is 0.5459, which means there is still a considerable amount of variation left to be explained.