705604096_stats101a_hw10

Jade Gregory

2023-06-06

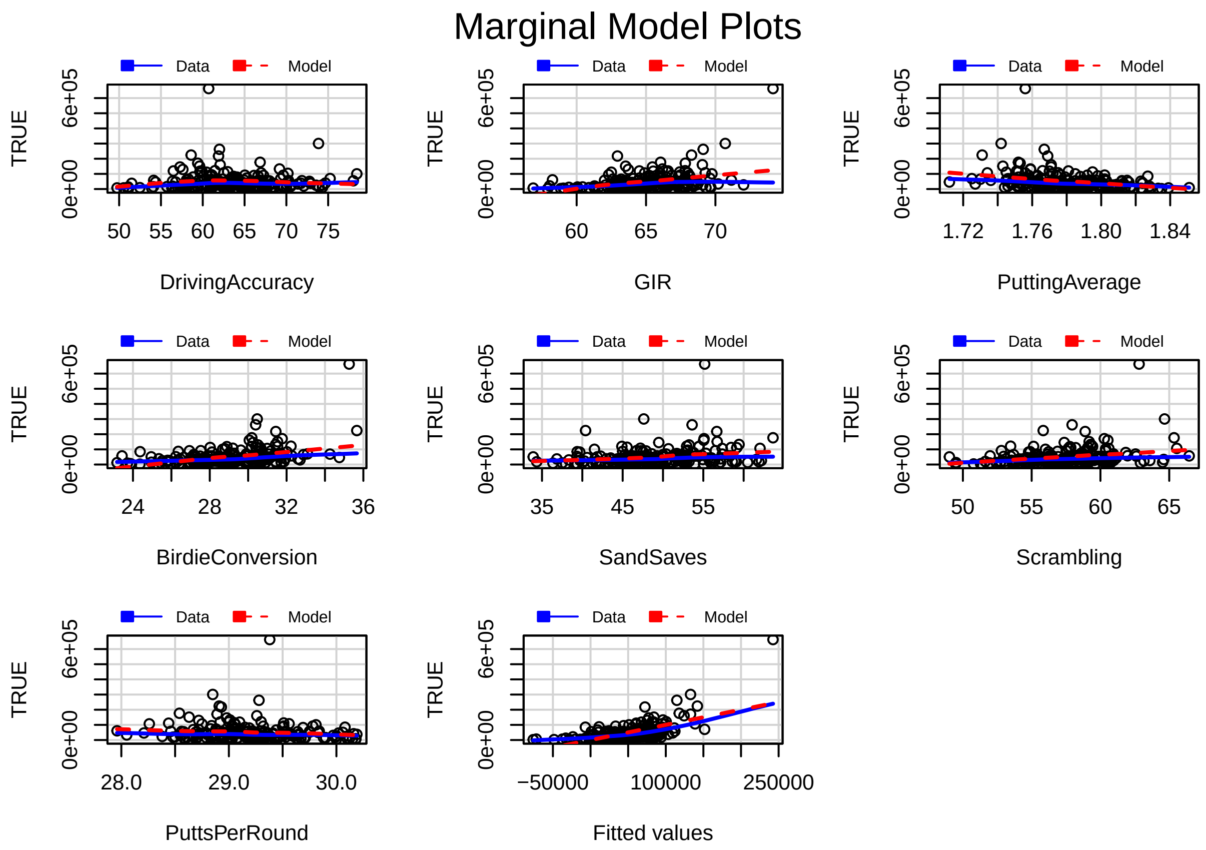
Question A

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

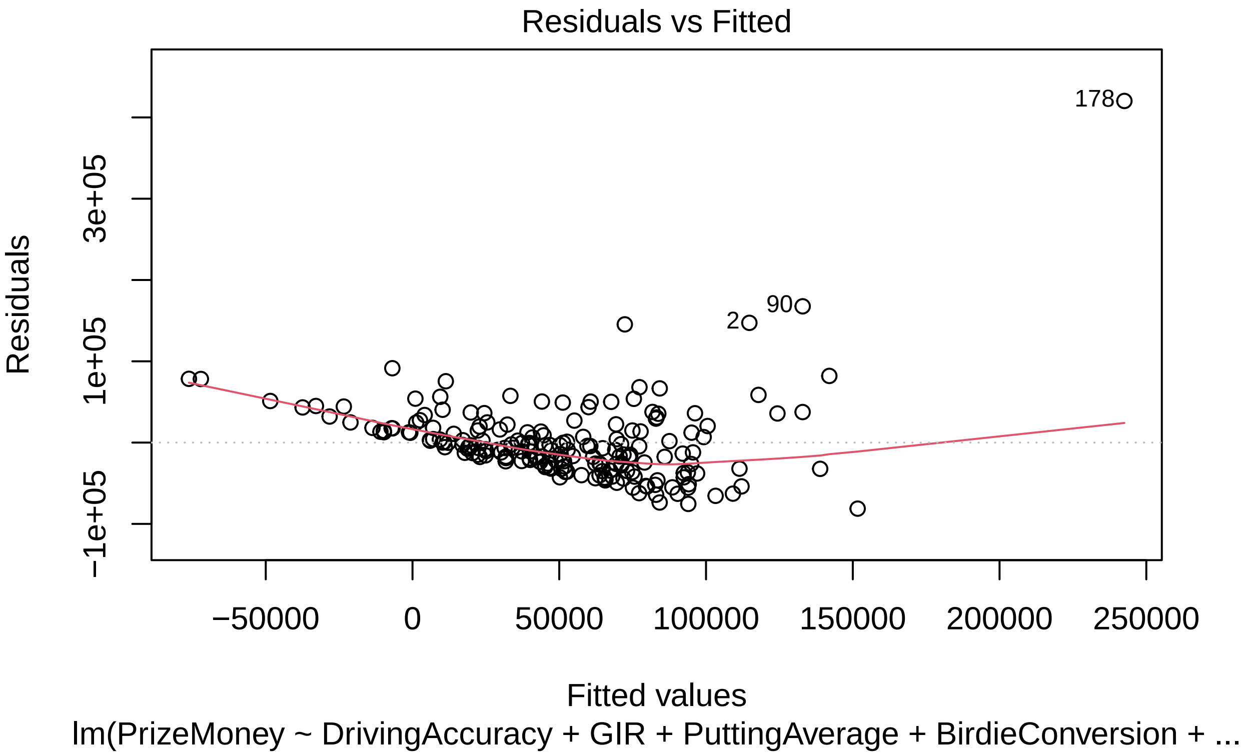
```
##                Name TigerWoods PrizeMoney AveDrivingDistance DrivingAccuracy
## 1   Aaron Baddeley          0      60661              288.3           60.73
## 2       Adam Scott          0     262045              301.1           62.00
## 3      Alex Aragon          0       3635              302.6           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
```

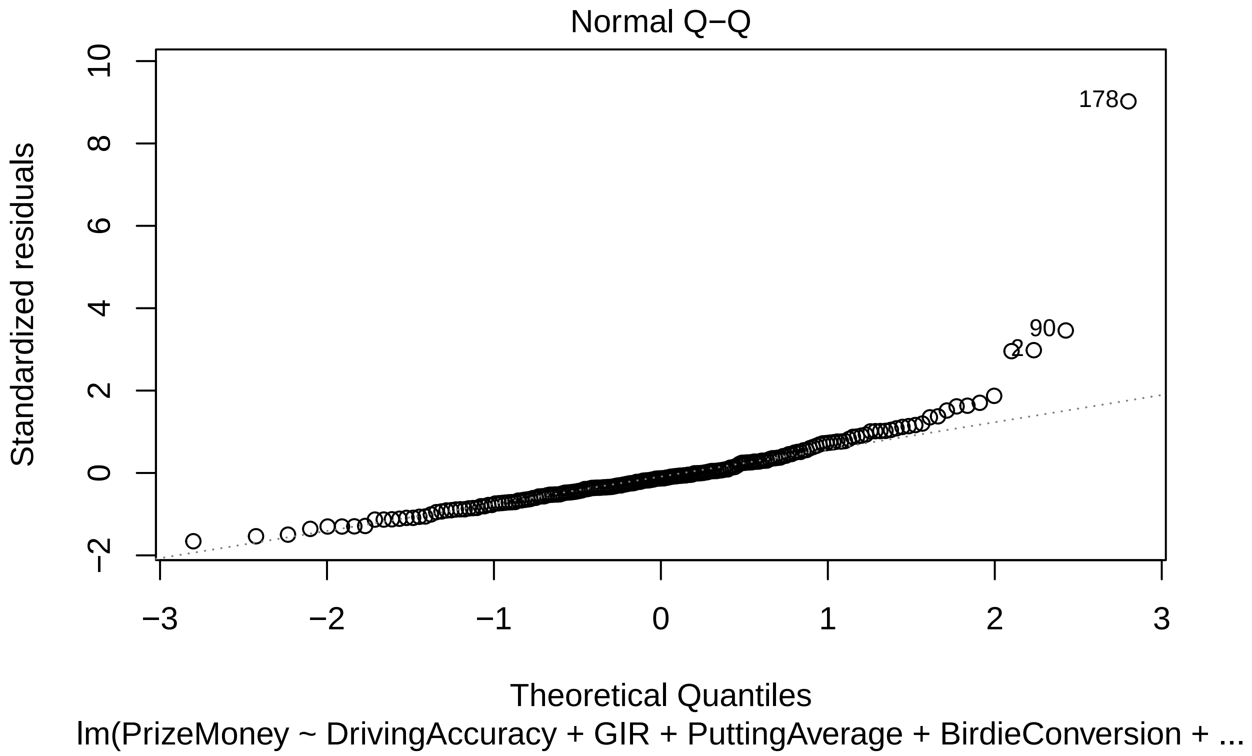
a)

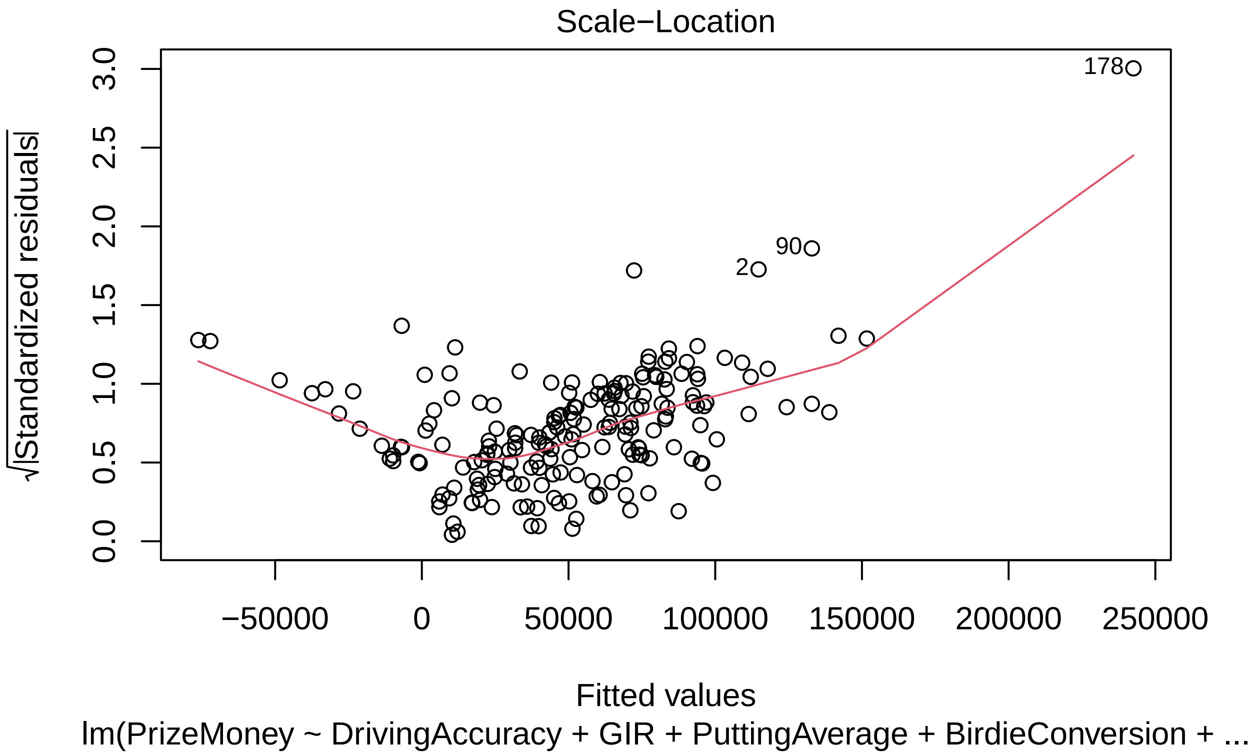
```
pga.lm <- lm(PrizeMoney ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + SandSaves + Scram
mmps(pga.lm)
```

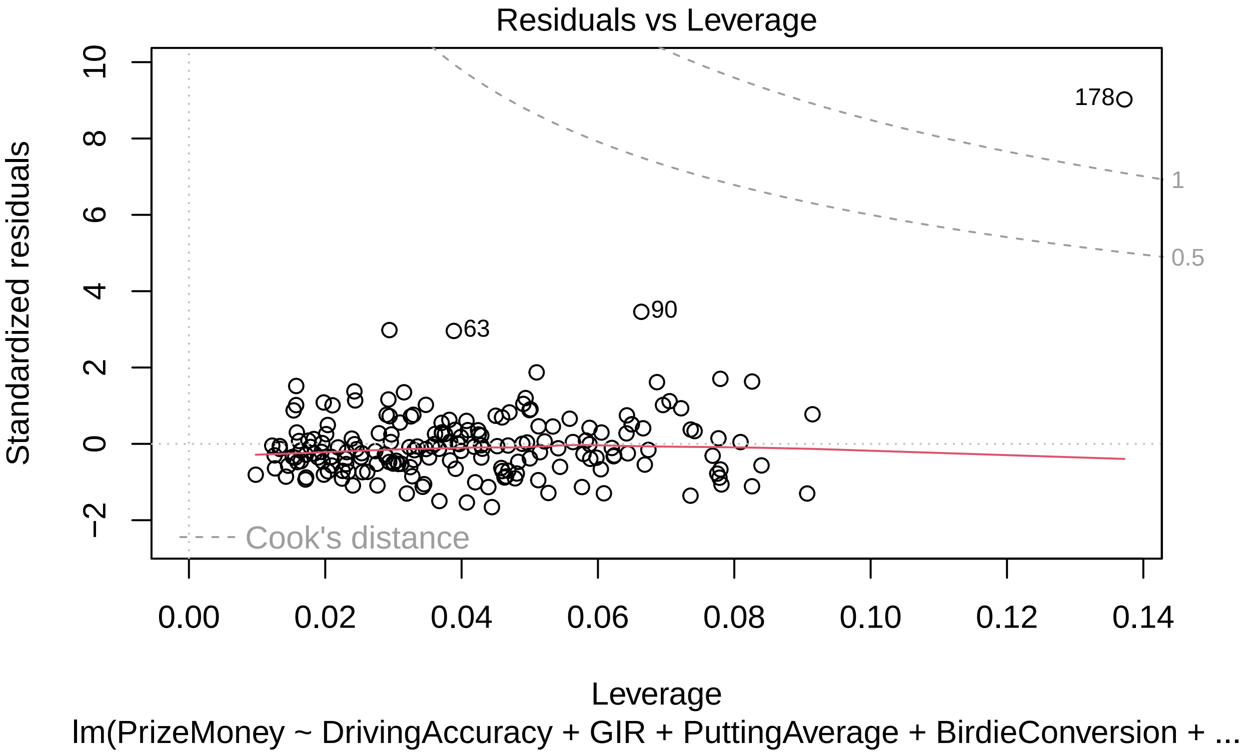

Marginal Model Plots
Data
Model
TRUE
0e+00
6e+05
50 55 60 65 70 75
DrivingAccuracy
60 65 70
GIR
1.72 1.76 1.80 1.84
PuttingAverage
24 28 32 36
BirdieConversion
35 45 55
SandSaves
50 55 60 65
Scrambling
28.0 29.0 30.0
PuttsPerRound
−50000 100000 250000
Fitted values

```
plot(pga.lm)
```

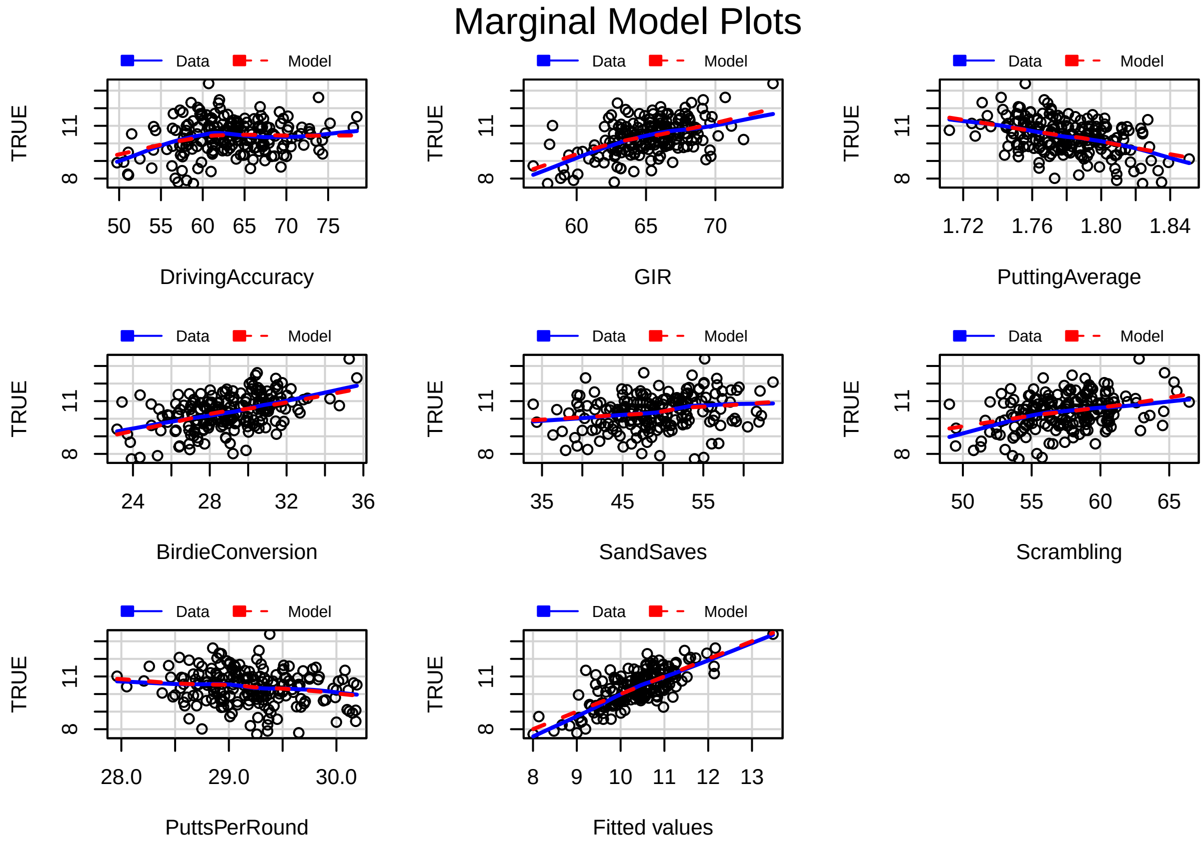

Residuals vs Fitted
Residuals
−1e+05
1e+05
3e+05
−50000
0
50000
100000
150000
200000
250000
Fitted values
lm(PrizeMoney ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + ...
178
90
2


Normal Q−Q
Standardized residuals
−2
0
2
4
6
8
10
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(PrizeMoney ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + ...
178
90
2

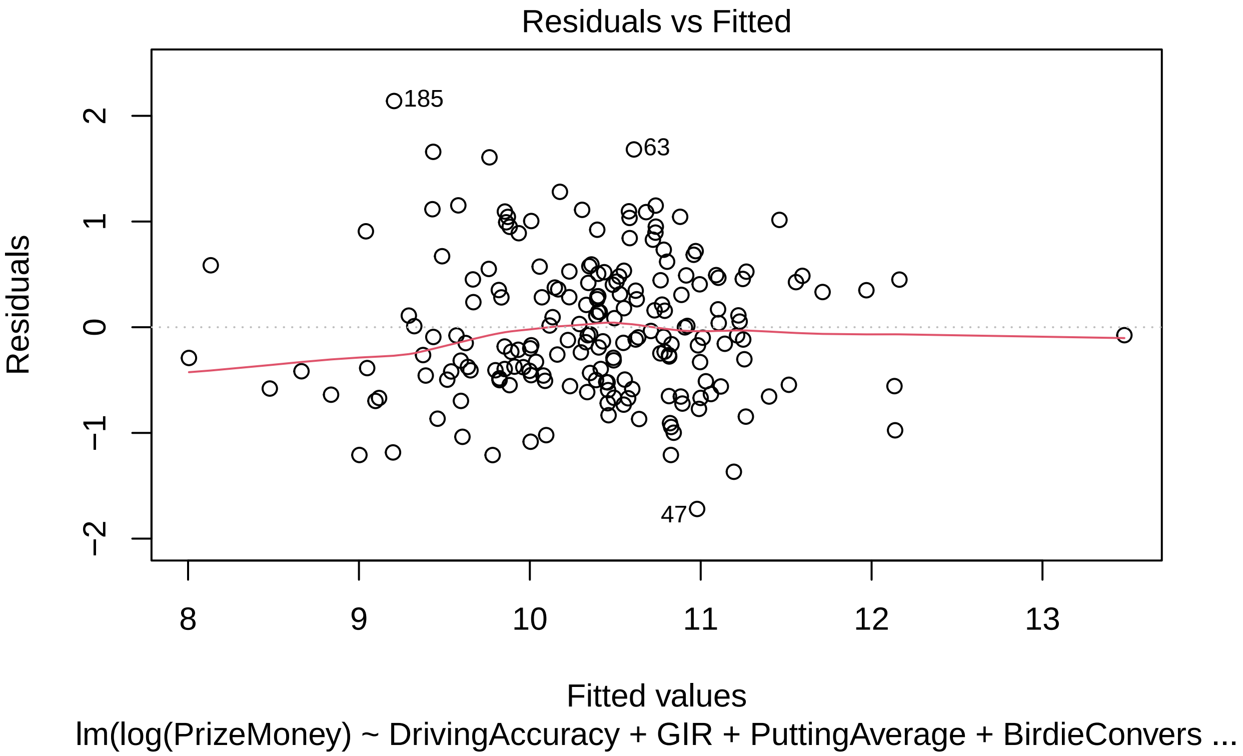

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−50000
0
50000
100000
150000
200000
250000
Fitted values
lm(PrizeMoney ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + ...
178
90
2

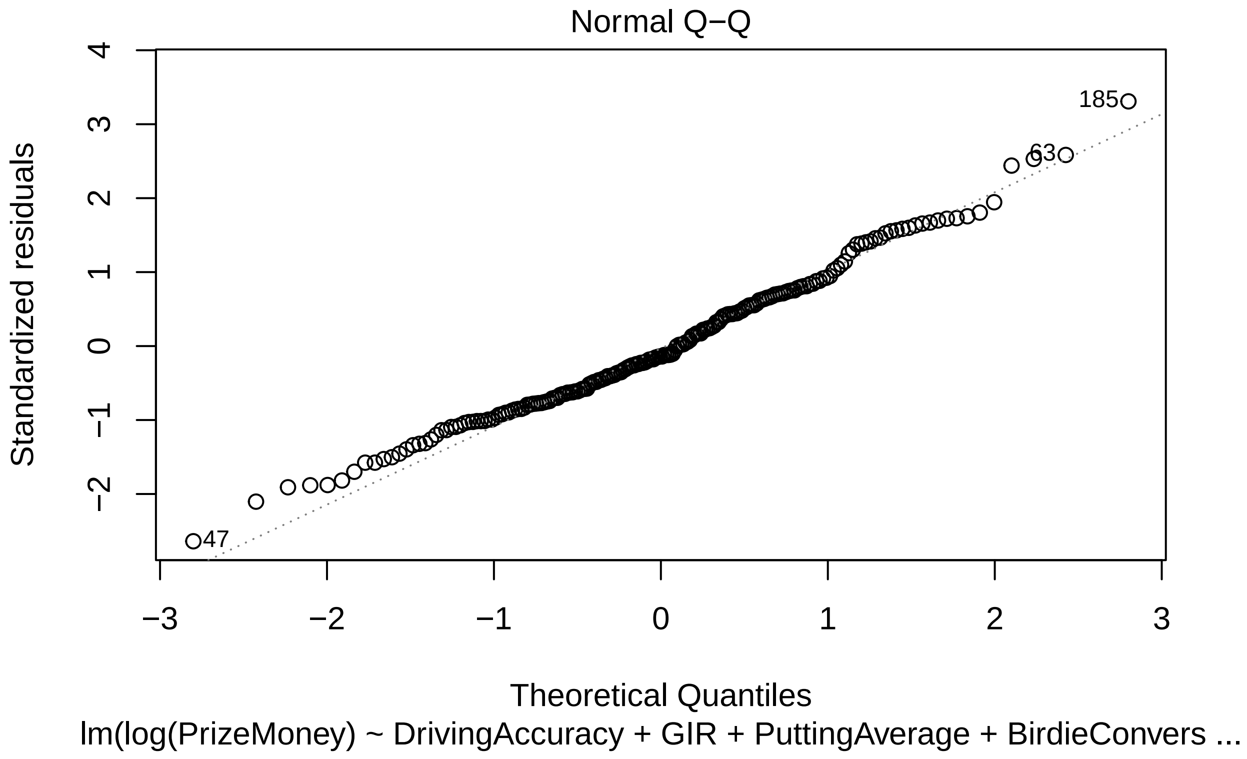


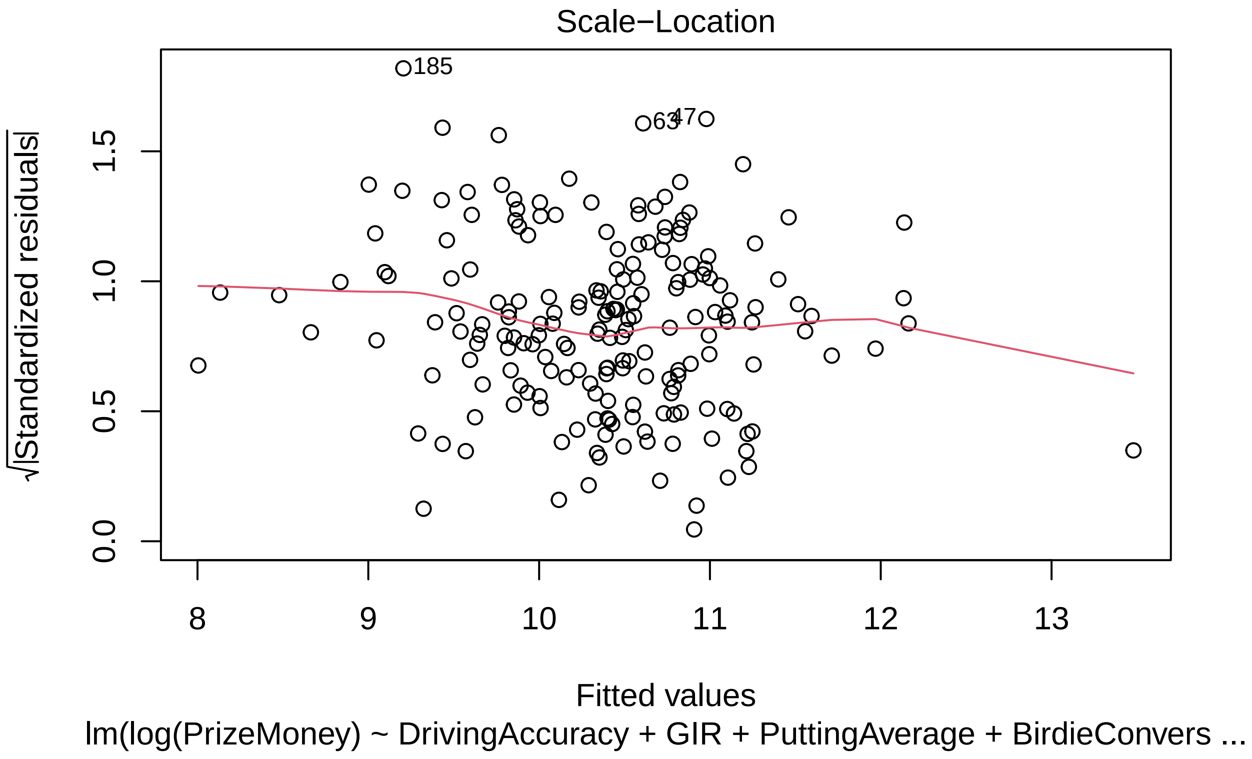
```
pga.logmodel <- lm(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + SandSa
mmps(pga.logmodel)
```

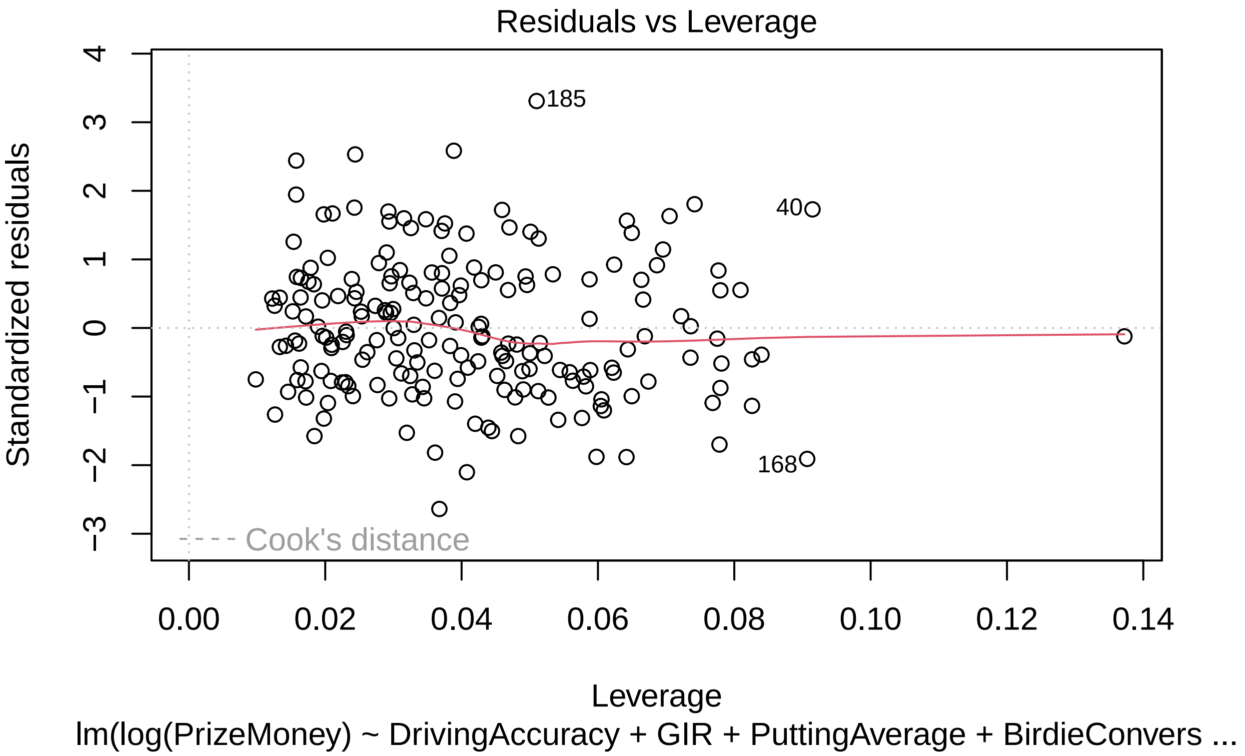


```
plot(pga.logmodel)
```


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


Scale−Location
185
63
47
√|Standardized residuals|
0.0
0.5
1.0
1.5
8
9
10
11
12
13
Fitted values
lm(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConvers ...



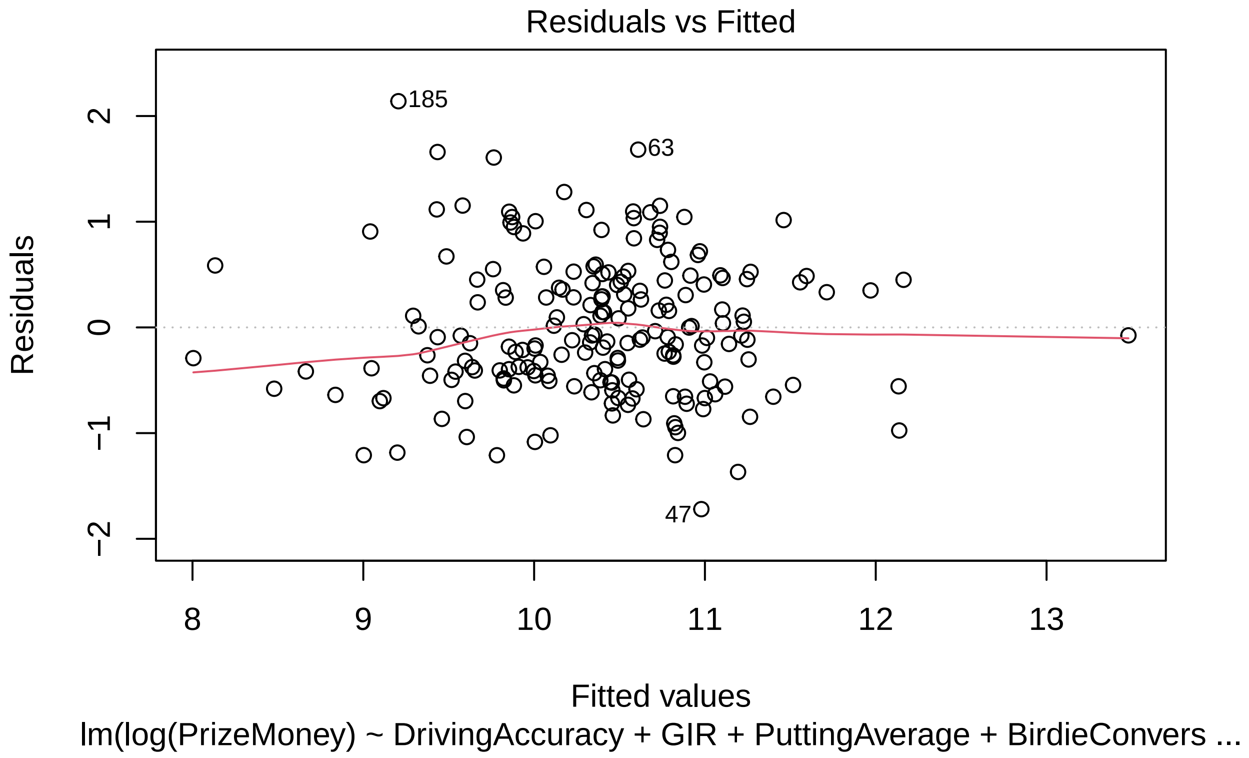
Overall, I agree with the statistician that the log transformation on the Y variable produces a better model for this data. In our mmps of the two models, we can see that the loess line in our log transformation better fits our regression line than in our non-transformed model. This indicates that our log transformation model is better for our data set. Also, we can compare the four plots of the two models. The residual plot for our non-transformed model has a fan shape, suggesting a violation of the constant variance assumption. In our residual plot for our log transformation model, though, has data points plotted evenly and horizontally across the plot, further indicating that this model is the better fit. In our QQ plots, the data points in the log transformation model more tightly follow the ashed line than in our non-transformed model, also suggesting the transformed model is the better model. Analyzing the scale location plot holds the same conclusion, as we can see that the transformed model's plot is evidence of it being a better fit for our data as the points are plotted horizontally across the plane, unlike the data in our scale location plot of our non-transformed model. So, in conclusion, I believe it is smartest to agree with the statisticians recommendation.

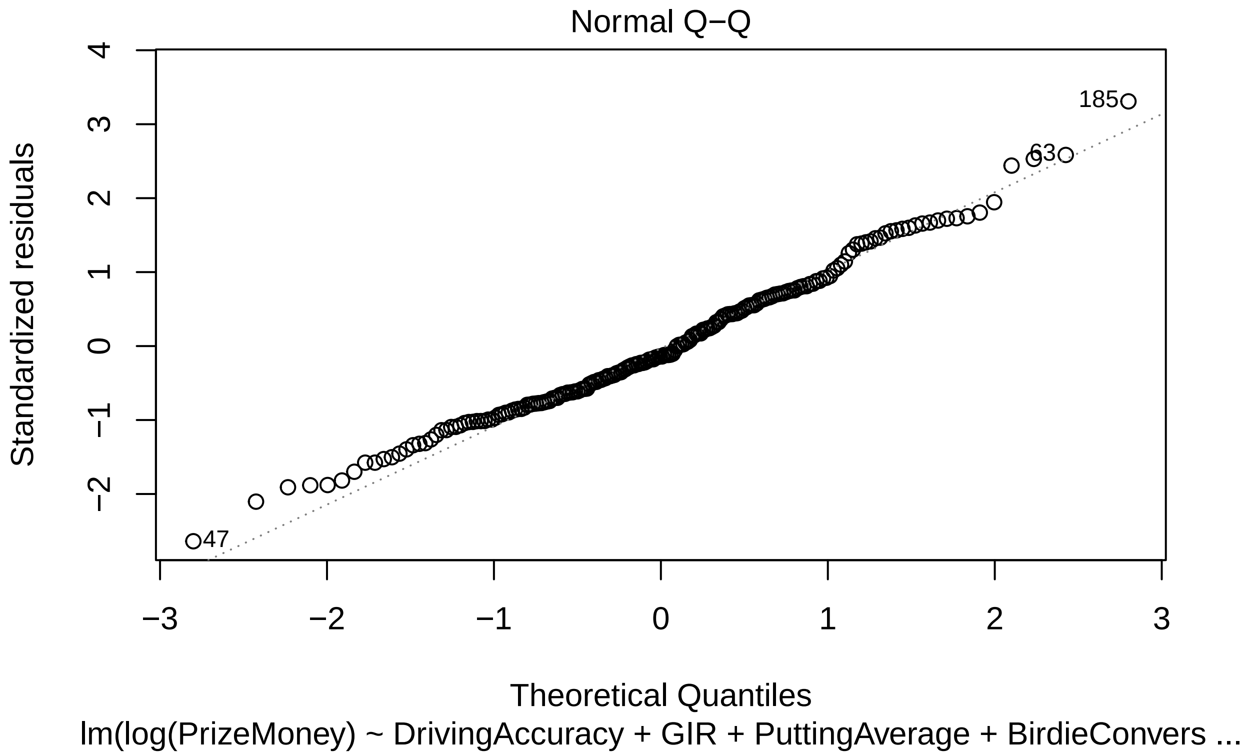
b)

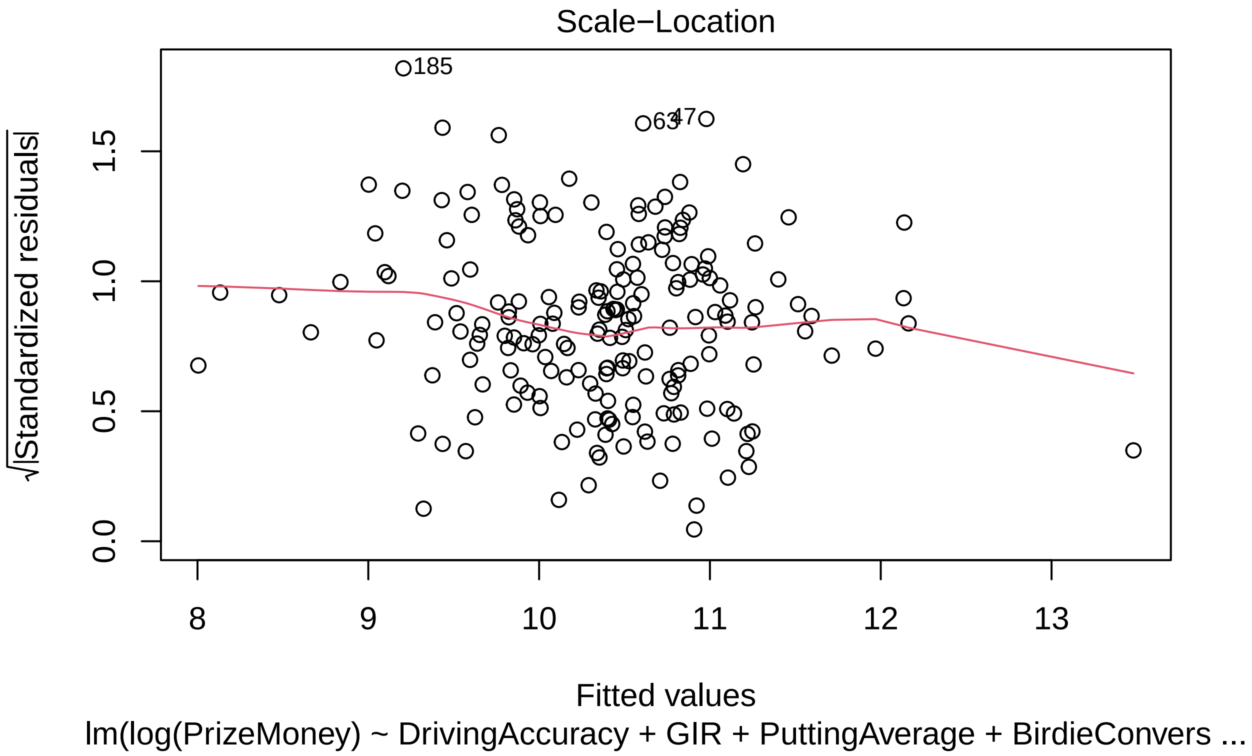
```
summary(pga.logmodel)
```

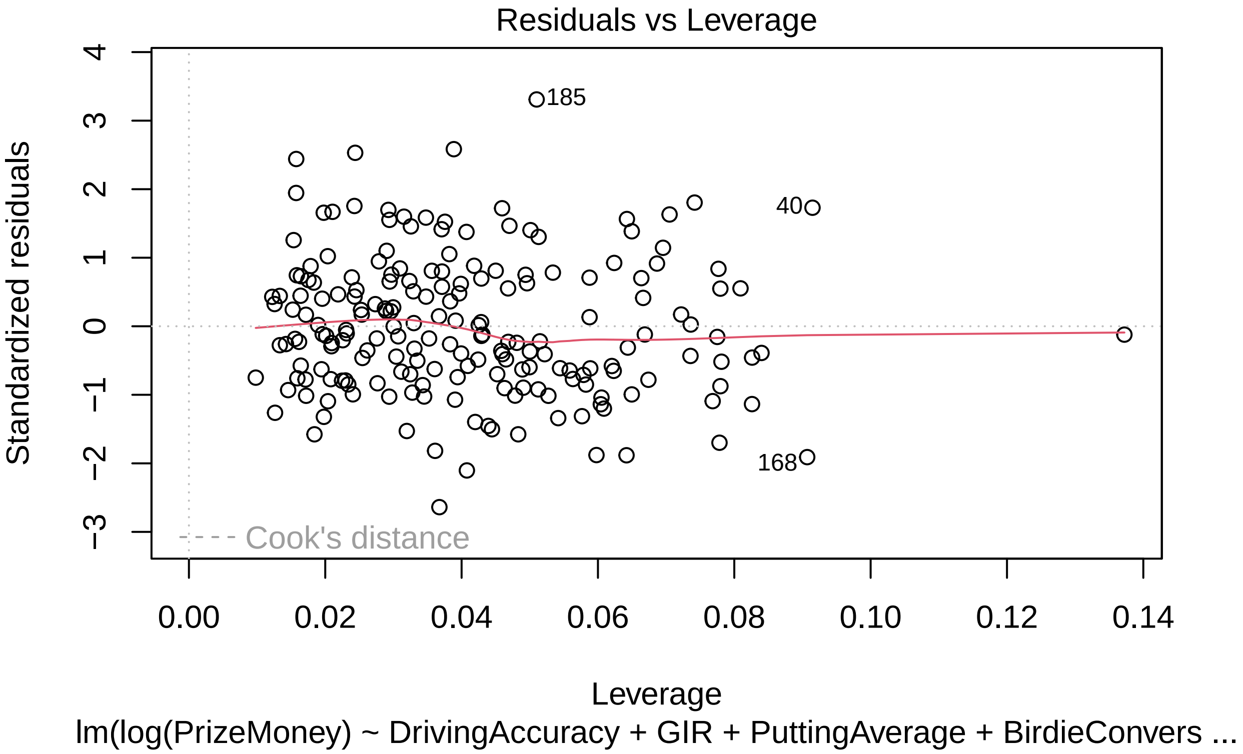
```
##
## Call:
## lm(formula = log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage +
##     BirdieConversion + SandSaves + Scrambling + PuttsPerRound,
##     data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71949 -0.48608 -0.09172  0.44561  2.14013
##
## Coefficients:
##                   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
```

```
plot(pga.logmodel)
```




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


Scale−Location
185
63
47
√|Standardized residuals|
0.0
0.5
1.0
1.5
8
9
10
11
12
13
Fitted values
lm(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConvers ...



My full regression of the model is log(PrizeMoney) = 0.194300 + (-0.003530) * DrivingAccuracy + 0.199311 * GIR + (-0.466304) * PuttingAverage + 0.157341 * BirdieConversion + 0.015174 * SandSaves + 0.051514 * Scrambling + (-0.343131) * PuttsPerRound. From our plots, we can conclude that this is a valid, well fitting model for our data. In our residual plot, the data is plotted evenly and horizontally across the plane suggesting that the constant variance assumption is held in our model. In the QQ norm plot, the data follows the dashed line suggesting that the normality assumption is held in our model. From this analysis, we can conclude that this is a good fitting model for our data.

c)

```
(1 + 7) / nrow(pga)
```

```
## [1] 0.04081633
```

Our qualification for bad leverage points are points with leverage greater than 0.04081633 and have a standardized residual outside of [-2,2]. In our leverage plot we can see that the points 185 and 168 fit the criteria for bad leverage points. Also, point 40 is hard to determine if it is out of the bounds [-2,2], so it may be worth looking into.

d) In our scale location plot, we are able to see a slight negative association suggesting that the constant variance condition may not hold in our model. Also, there are a few bad leverage points that we need to explore as well as a handful of high leverage points.

e) I would not recommend this approach because in our marginal plots we can see that all of the variables belong in this model. Removing a variable would change how we analyze the rest of the variables, because these response variables deemed insignificant in from the summary still have an effect on the other variables and our model as a whole. Therefore, it would not be in best practice to remove these predictors with insignificant t-values.

Question B

a)

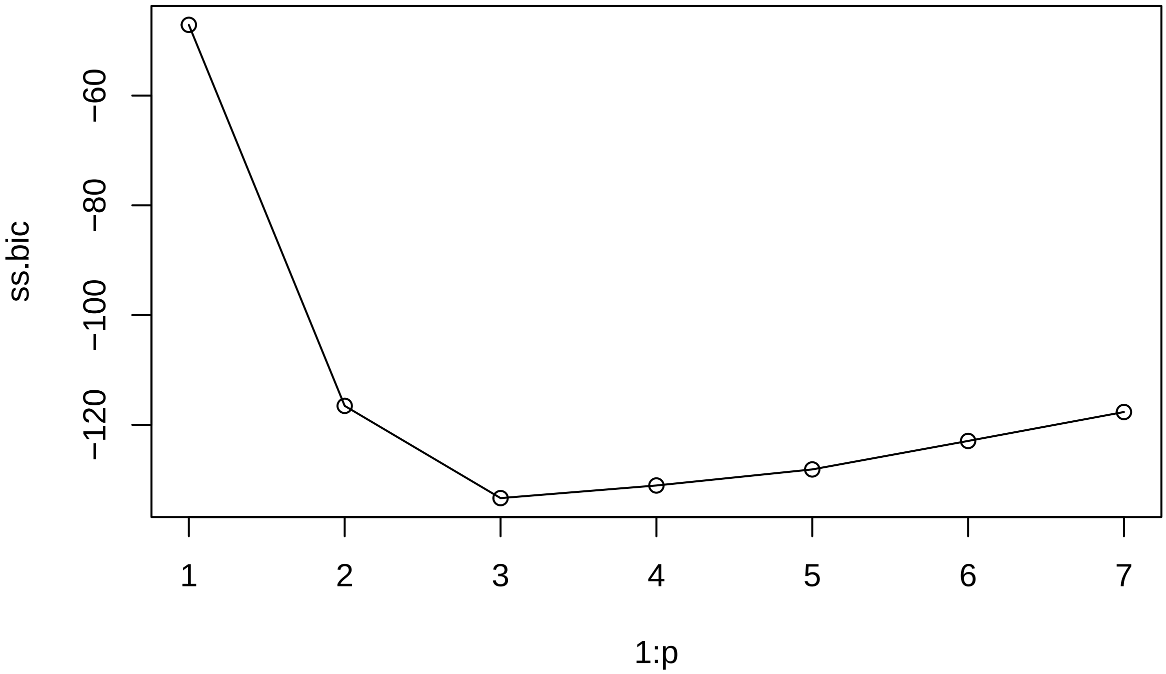
```
require(leaps)
```

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

```
bestss <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + SandS
summary(bestss)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga)
## 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 ) "*"        "*"
```

```
n <- nrow(pga)
ss.bic <- summary(bestss)$bic
p <- length(ss.bic)
plot(1:p, ss.bic)
lines(1:p, ss.bic)
```



```
summary(bestss)$which[which.min(ss.bic),] |> which()
```

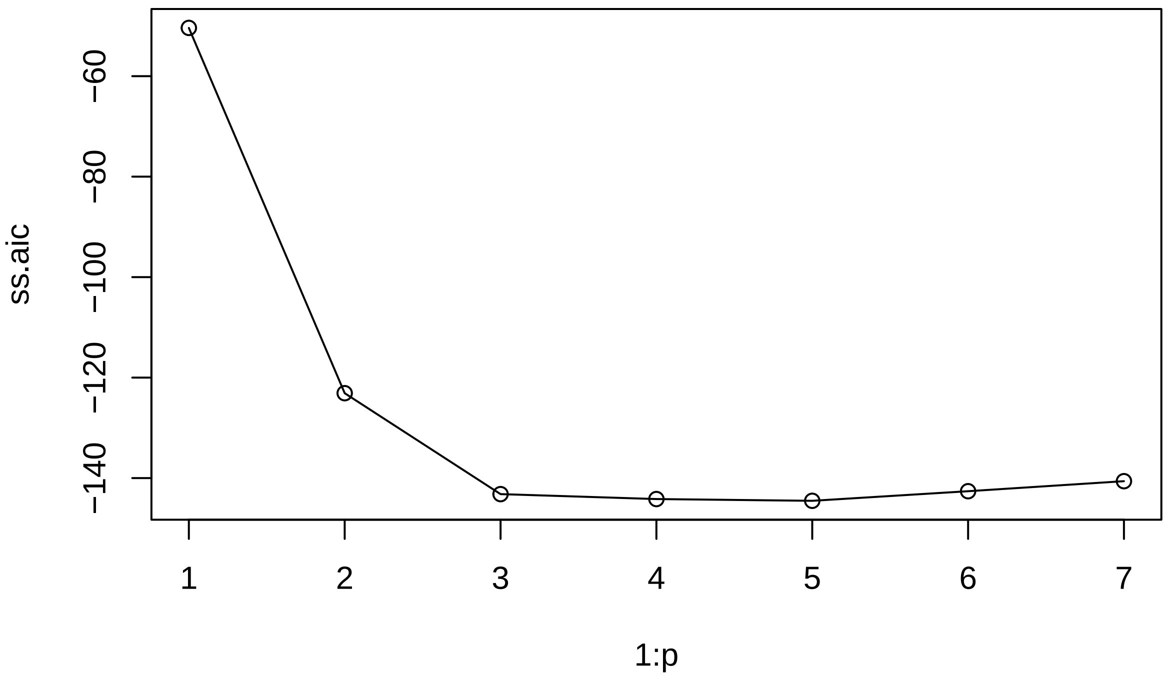
```
##      (Intercept)              GIR BirdieConversion       Scrambling
##                1                3                5                7
```

```
ss.bestmodelBIC <- lm(log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling, data = pga)
summary(ss.bestmodelBIC)
```

```
##
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling,
##     data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71081 -0.50717 -0.06683  0.41975  2.04147
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -11.08314    1.45712  -7.606 1.23e-12 ***
## GIR                0.15658    0.01787   8.761 1.01e-15 ***
## BirdieConversion   0.20625    0.02164   9.531  < 2e-16 ***
## Scrambling         0.09178    0.01539   5.965 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6661 on 192 degrees of freedom
## Multiple R-squared:  0.5453, Adjusted R-squared:  0.5382
## F-statistic: 76.75 on 3 and 192 DF,  p-value: < 2.2e-16
```

Utilizing the best subsets method with BIC, our optimal model is log(PrizeMoney) = -11.08314 + 0.15658 * GIR + 0.20625 * BirdieConversion + 0.09178 * Scrambling.

```
ss.aic <- ss.bic
for(i in 1:p){
  ss.aic[i] <- ss.bic[i] - (log(n) * i) + (2 * i)
}
plot(1:p, ss.aic)
lines(1:p, ss.aic)
```



```
summary(bestss)$which[which.min(ss.aic),] |> which()
```

```
##      (Intercept)              GIR BirdieConversion        SandSaves
##                1                3                5                6
##       Scrambling    PuttsPerRound
##                7                8
```

```
ss.bestmodelAIC <- lm(log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves + Scrambling + PuttsPerRound
summary(ss.bestmodelAIC)
```

```
##
## 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
```

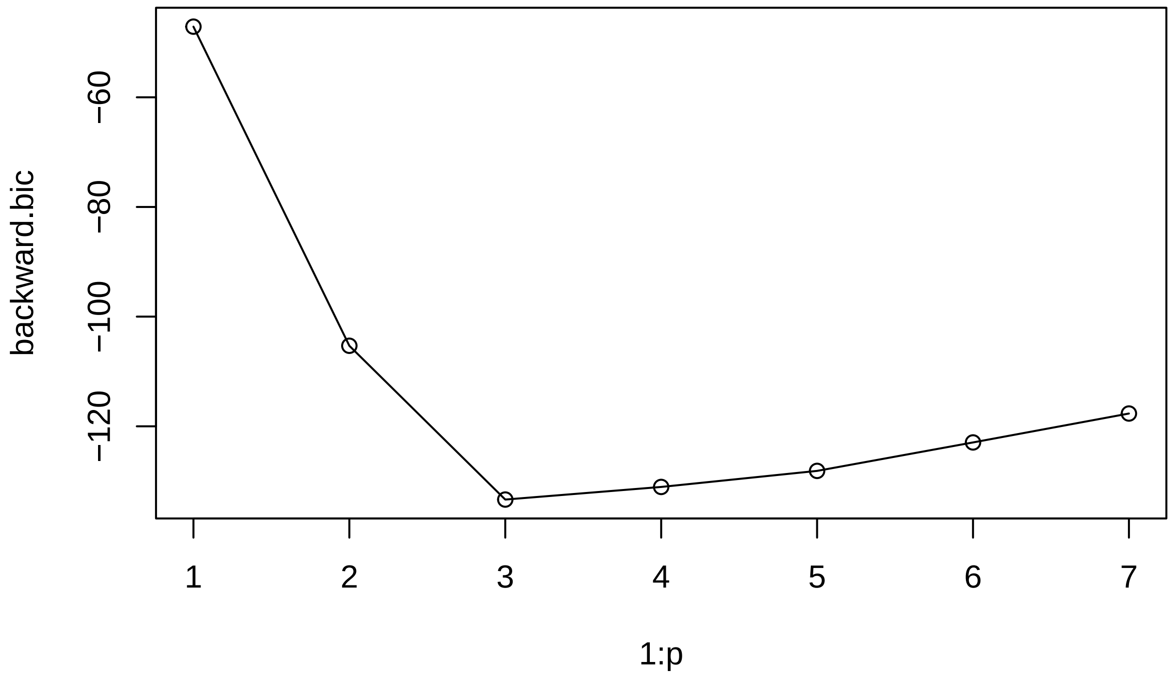
Using the same method with AIC, our optimal model is log(PrizeMoney) = -0.583181 + 0.197022 * GIR + 0.162752 * BirdieConversion + 0.015524 * SandSaves + 0.049635 * Scrambling + (-0.349738) * PuttsPerRound.

b)

```
backward <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + Sa
summary(backward)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga, 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 ) "*"        "*"
```

```
backward.bic <- summary(backward)$bic
plot(1:p, backward.bic)
lines(1:p, backward.bic)
```



```
summary(backward)$which[which.min(backward.bic),] |> which()
```

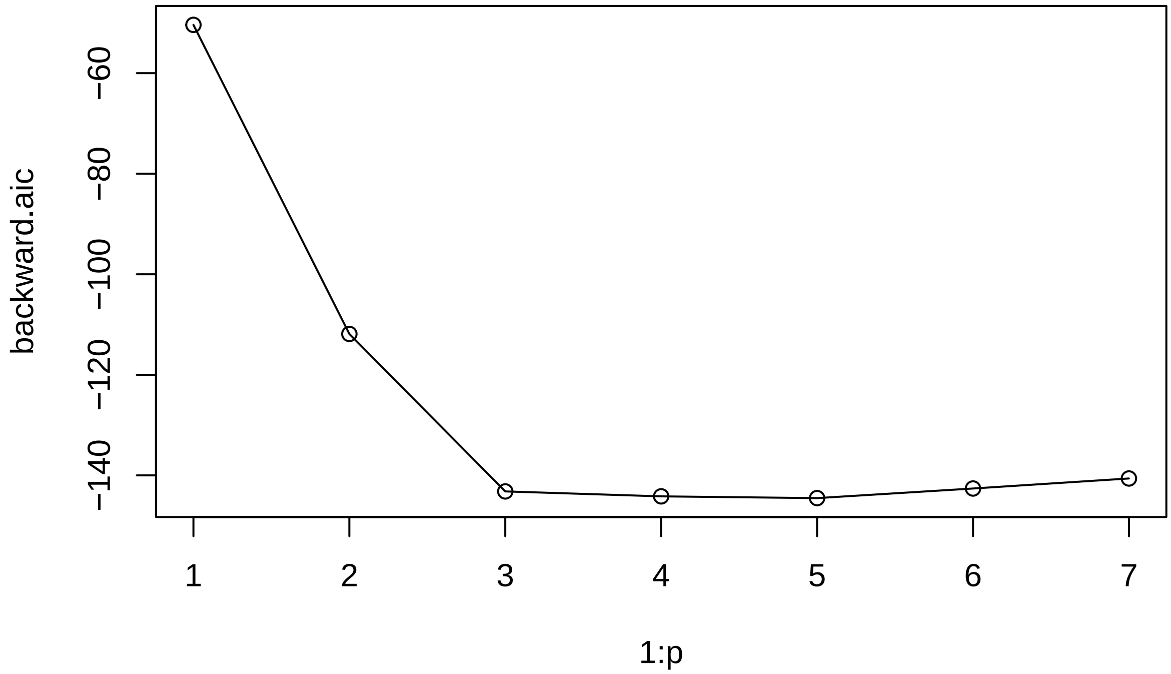
```
##      (Intercept)              GIR BirdieConversion       Scrambling
##                1                3                5                7
```

```
backward.bestmodelBIC <- lm(log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling, data = pga)
summary(backward.bestmodelBIC)
```

```
##
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling,
##     data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71081 -0.50717 -0.06683  0.41975  2.04147
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -11.08314    1.45712  -7.606 1.23e-12 ***
## GIR                0.15658    0.01787   8.761 1.01e-15 ***
## BirdieConversion   0.20625    0.02164   9.531  < 2e-16 ***
## Scrambling         0.09178    0.01539   5.965 1.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6661 on 192 degrees of freedom
## Multiple R-squared:  0.5453, Adjusted R-squared:  0.5382
## F-statistic: 76.75 on 3 and 192 DF,  p-value: < 2.2e-16
```

Utilizing the backward selection method with BIC, our optimal model is log(PrizeMoney) = -11.08314 + 0.15658 * GIR + 0.20625 * BirdieConversion + 0.09178 * Scrambling.

```
backward.aic <- backward.bic
for(i in 1:p){
  backward.aic[i] <- backward.bic[i] - (log(n) * i) + (2 * i)
}
plot(1:p, backward.aic)
lines(1:p, backward.aic)
```



```
summary(backward)$which[which.min(backward.aic),] |> which()
```

```
##      (Intercept)              GIR BirdieConversion        SandSaves 
##                1                3                5                6 
##       Scrambling    PuttsPerRound 
##                7                8
```

```
backward.bestmodelAIC <- lm(log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves + Scrambling + PuttsPe
summary(backward.bestmodelAIC)
```

```
## 
## 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
```

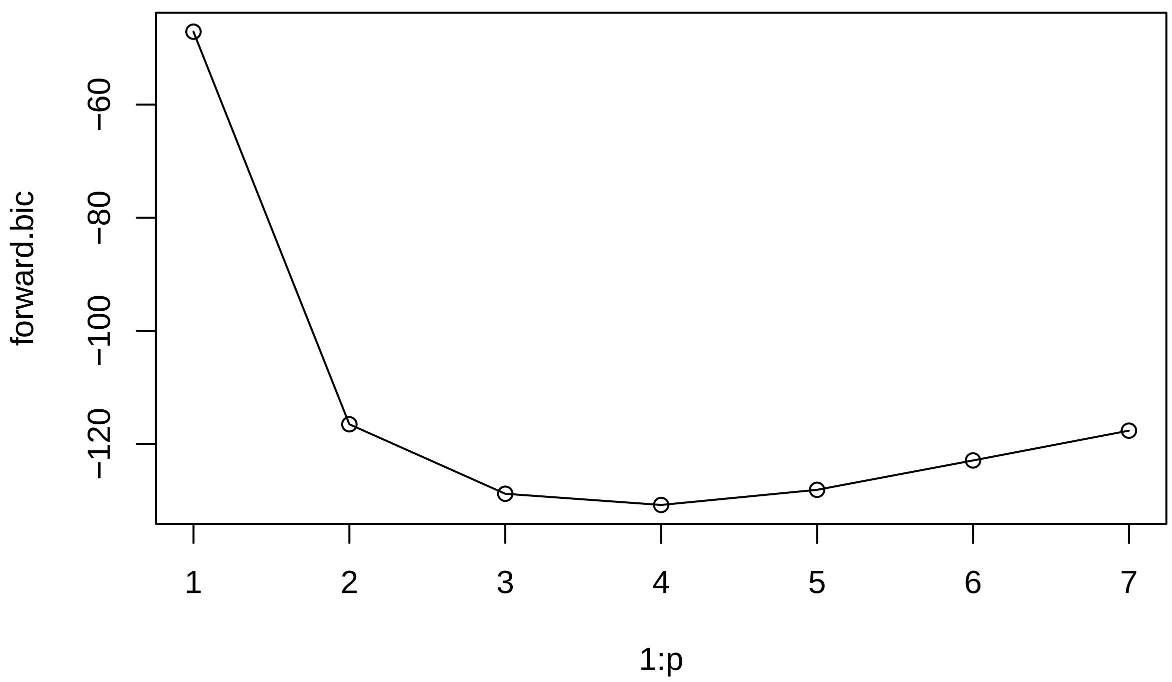
Using the backward selection method with AIC, our optimal model is log(PrizeMoney) = -0.583181 + 0.197022 * GIR + 0.162752 * BirdieConversion + 0.015524 * SandSaves + 0.049635 * Scrambling + (-0.349738) * PuttsPerRound.

c)

```
forward <- regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + Sand
summary(forward)
```

```
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, data = pga, 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 ) "*"        "*"
```

```
forward.bic <- summary(forward)$bic
plot(1:p, forward.bic)
lines(1:p, forward.bic)
```



```
summary(forward)$which[which.min(forward.bic),] |> which()
```

```
##      (Intercept)              GIR BirdieConversion       Scrambling 
##                1                3                5                7 
##    PuttsPerRound 
##                8
```

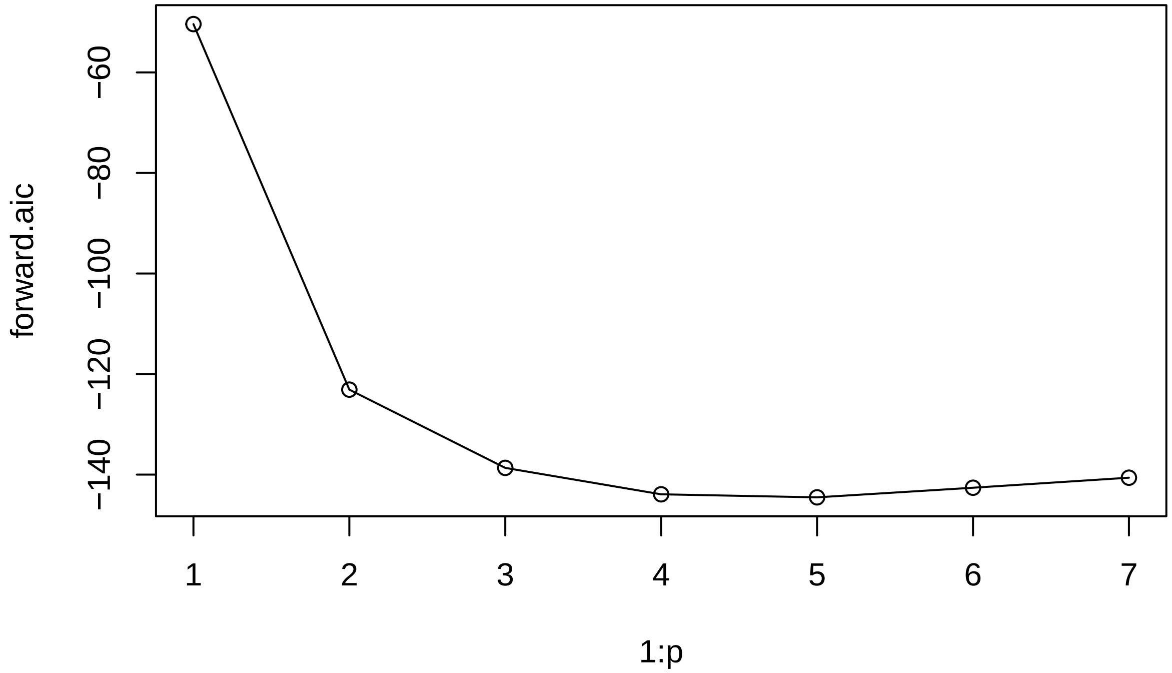
```
forward.bestmodelBIC <- lm(log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling + PuttsPerRound, data
summary(forward.bestmodelBIC)
```

```
## 
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling + 
##     PuttsPerRound, data = pga)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.68884 -0.49753 -0.07461  0.43648  2.08504 
## 
## Coefficients:
```

```
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.39320    7.16112   0.055  0.95627
## GIR                 0.19352    0.02874   6.733 1.89e-10 ***
## BirdieConversion    0.16589    0.03274   5.066 9.52e-07 ***
## Scrambling          0.06282    0.02341   2.684  0.00792 **
## PuttsPerRound      -0.37840    0.23122  -1.637  0.10338
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6632 on 191 degrees of freedom
## Multiple R-squared:  0.5516, Adjusted R-squared:  0.5422
## F-statistic: 58.74 on 4 and 191 DF,  p-value: < 2.2e-16
```

Utilizing the forward selection method with BIC, our optimal model is log(PrizeMoney) = 0.39320 + 0.19352 * GIR + 0.16589 * BirdieConversion + 0.06282 * Scrambling + (-0.37840) * PuttsPerRound.

```
forward.aic <- forward.bic
for(i in 1:p){
  forward.aic[i] <- forward.bic[i] - (log(n) * i) + (2 * i)
}
plot(1:p, forward.aic)
lines(1:p, forward.aic)
```



```
summary(forward)$which[which.min(forward.aic),] |> which()
```

```
##      (Intercept)              GIR BirdieConversion        SandSaves
##                1                3                5                6
##       Scrambling    PuttsPerRound
##                7                8
```

```
forward.bestmodelAIC <-lm(log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves + Scrambling + PuttsPerR
summary(forward.bestmodelAIC)
```

```
##
## 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
```

Utilizing the forward selection method with AIC, our optimal model is log(PrizeMoney) = -0.583181 + 0.197022 * GIR + 0.162752 * BirdieConversion + 0.015524 * SandSaves + 0.049635 * Scrambling + (-0.349738) * PuttsPerRound.

e)

```
AIC(ss.bestmodelBIC)
```

```
## [1] 402.9131
```

```
BIC(ss.bestmodelBIC)
```

```
## [1] 419.3037
```

```
AIC(ss.bestmodelAIC)
```

```
## [1] 401.5823
```

```
BIC(ss.bestmodelAIC)
```

```
## [1] 424.5291
```

```
AIC(forward.bestmodelBIC)
```

```
## [1] 402.1839
```

```
BIC(forward.bestmodelBIC)
```

```
## [1] 421.8526
```

In my previous work, I can see that the most optimal models have either 3, 4, or 5 variables included. To determine the most efficient model, I compared the AIC and BIC of the models to one another and concluded that the model with 5 variables has the smallest AIC and the largest BIC, the model with 4 variables has the second smallest AIC and BIc, and the model with 3 variables has the largest AIC and the smallest BIC. From this, I would recommend the simplest model given since none of them have the absolute smallest AIC and BIC both. My final model is log(PrizeMoney) = -11.08314 + 0.15658 * GIR + 0.20625 * BirdieConversion + 0.09178 * Scrambling.

f)

```
summary(ss.bestmodelBIC)
```

```
##
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + Scrambling,
##     data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71081 -0.50717 -0.06683  0.41975  2.04147
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -11.08314    1.45712  -7.606 1.23e-12 ***
## GIR                0.15658    0.01787   8.761 1.01e-15 ***
## BirdieConversion   0.20625    0.02164   9.531  < 2e-16 ***
## Scrambling         0.09178    0.01539   5.965 1.16e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.6661 on 192 degrees of freedom
## Multiple R-squared:  0.5453, Adjusted R-squared:  0.5382
## F-statistic: 76.75 on 3 and 192 DF,  p-value: < 2.2e-16
```

Undoing the log conversion on log(PrizeMoney), we can see that PrizeMoney is 1.536928e-05 dollars when the GIR, BirdieConversion, and Scrambling variables are 0. The PrizeMoney increases by 1.169504 dollars on average for each increase in the GIR percentage. The PrizeMoney increases by 1.22906 dollars on average for each increase in the BirdieConversion percentage. The PrizeMoney increases by 1.096124 dollars on average for each increase in the Scrambling percentage. It is best to be cautious when interpreting these results, because real life does not always strictly follow predicted models. Also, we know that in real life it would not make sense for any of the variables measured with percentages to be negative or have values greater than 100.