Stats101A, Spring 2023 - Homework 10

Luke Villanueva - 206039397

06/08/23

Problem A

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.2.3
```

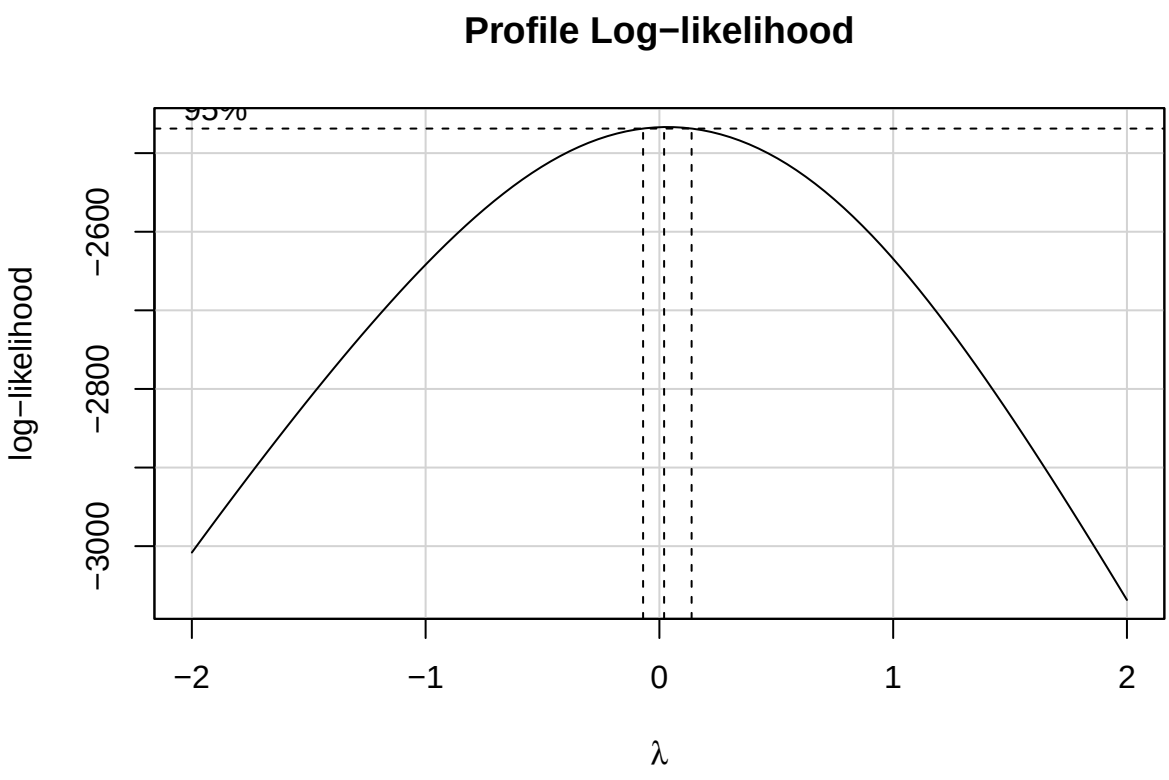
```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
pga <- read.csv(paste0(getwd(),"/pgatour2006-3.csv"))
glimpse(pga)
```

```
## Rows: 196
## Columns: 12
## $ Name               <chr> "Aaron Baddeley", "Adam Scott", "Alex Aragon", "Ale~
## $ TigerWoods         <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ PrizeMoney         <int> 60661, 262045, 3635, 17516, 16683, 107294, 50620, 5~
## $ AveDrivingDistance <dbl> 288.3, 301.1, 302.6, 288.8, 287.7, 285.0, 282.2, 28~
## $ DrivingAccuracy    <dbl> 60.73, 62.00, 51.12, 66.40, 63.24, 62.53, 72.76, 63~
## $ GIR                <dbl> 58.26, 69.12, 59.11, 67.70, 64.04, 69.27, 68.67, 62~
## $ PuttingAverage     <dbl> 1.745, 1.767, 1.787, 1.777, 1.761, 1.775, 1.812, 1.~
## $ BirdieConversion   <dbl> 31.36, 30.39, 29.89, 29.33, 29.32, 29.20, 24.95, 32~
## $ SandSaves          <dbl> 54.80, 53.61, 37.93, 45.13, 52.44, 47.20, 41.84, 45~
## $ Scrambling         <dbl> 59.37, 57.94, 50.78, 54.82, 57.07, 57.67, 55.46, 56~
## $ BounceBack         <dbl> 19.30, 19.35, 16.80, 17.05, 18.21, 20.00, 20.85, 19~
## $ PuttsPerRound      <dbl> 27.96, 29.28, 29.20, 29.46, 28.93, 29.56, 30.06, 28~
```

a.

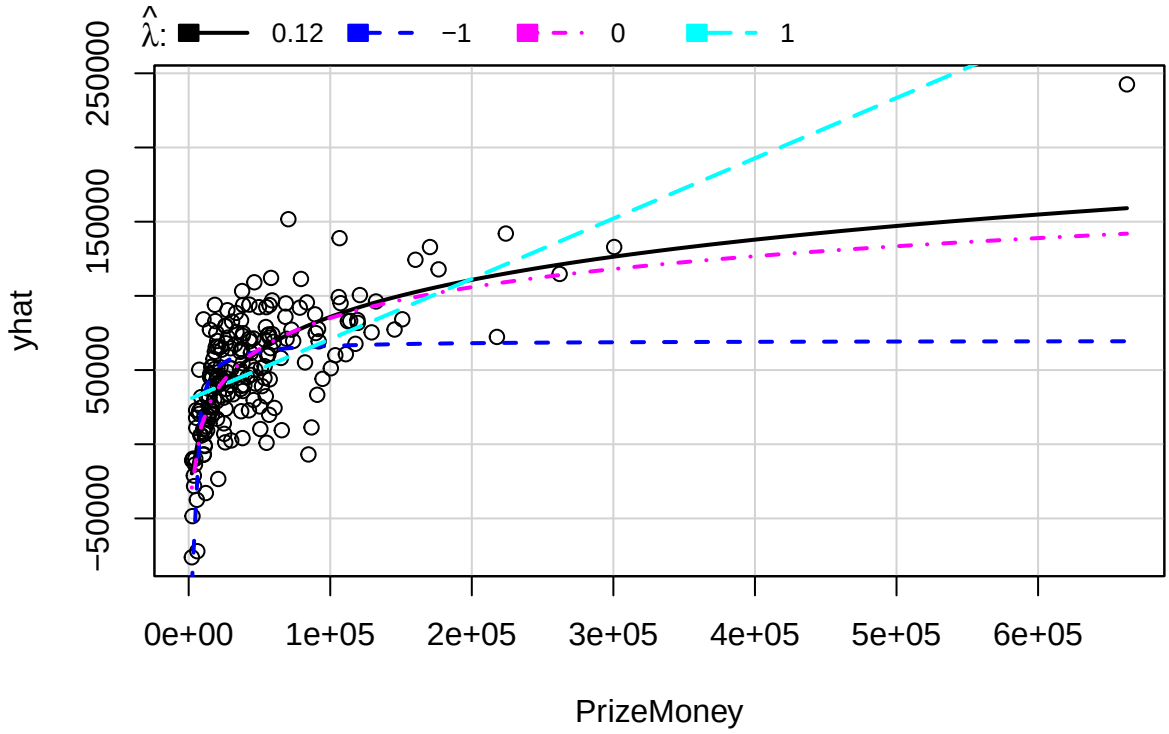
```
m <- lm(PrizeMoney ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversion + SandSaves + Scrambling

# boxcox
car::boxCox(m)
```



```
# powertransform
summary(powerTransform(m))
```

```
## 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
```

```
# inverse response
car::inverseResponsePlot(m)
```




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

BoxCox plot: confidence interval is small and centered around 0

Power Transform: fail to reject that transformation is log, reject that no transformation is needed -> there needs to be a transformation and it can be log transformation

Inverse Response Plot: the points closely followed Lambda = 0.12 or 0

In conclusion, the boxcox plot shows the optimal lambda being very close to 0. This implies that the recommended transformation by the boxcox plot is a value close to 0. In addition, the PowerTransform summary recommends that there indeed needs to be a transformation to the Y variable, and the transformation can be log. Finally, the inverse response plot shows the points closely follow the curves for $\lambda = 0.12$ and $\lambda = 0$.

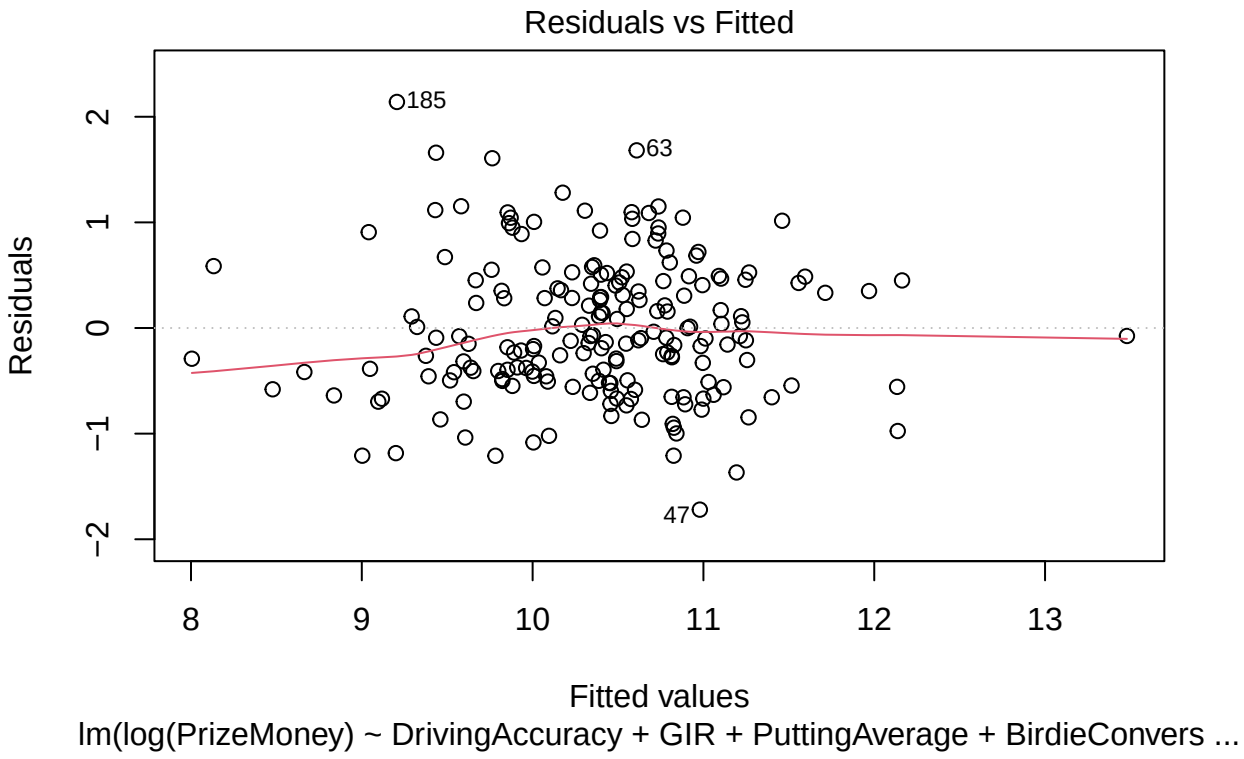
And so, all three of these tests showcase that a log transformation on the Y variable is a decent recommendation.

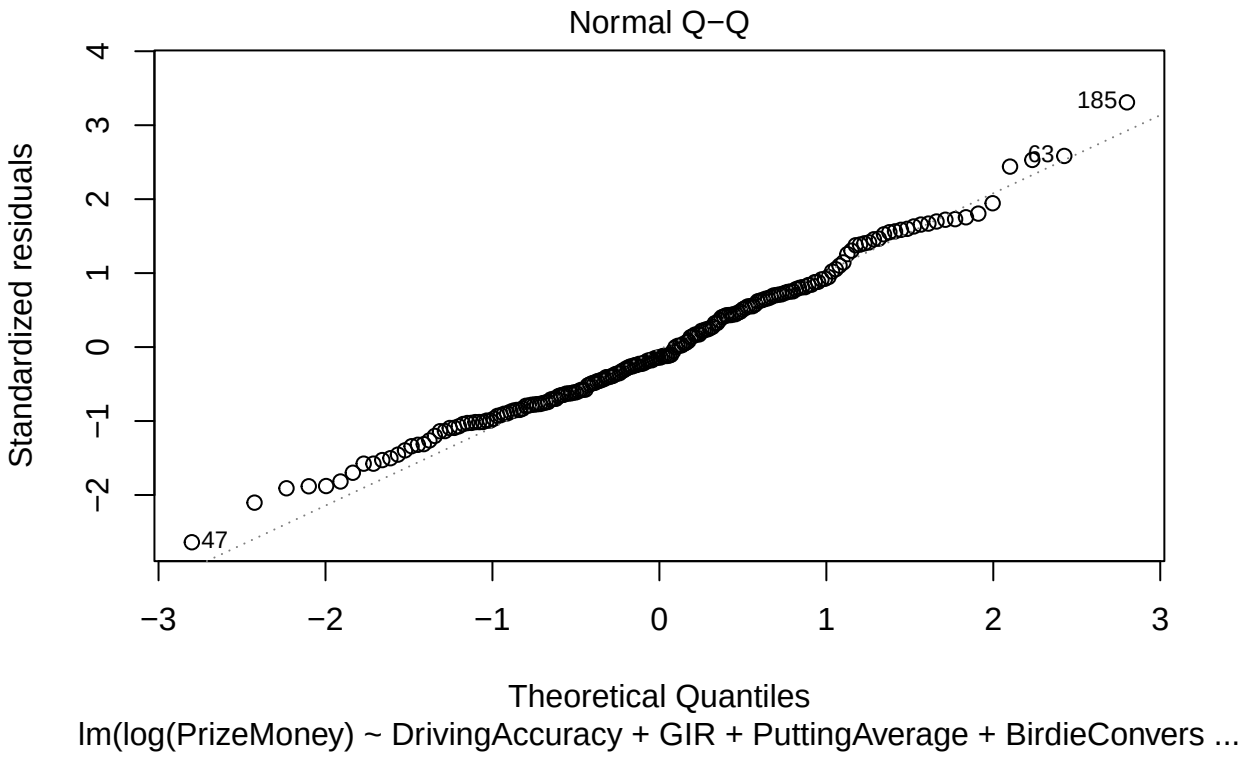
b.

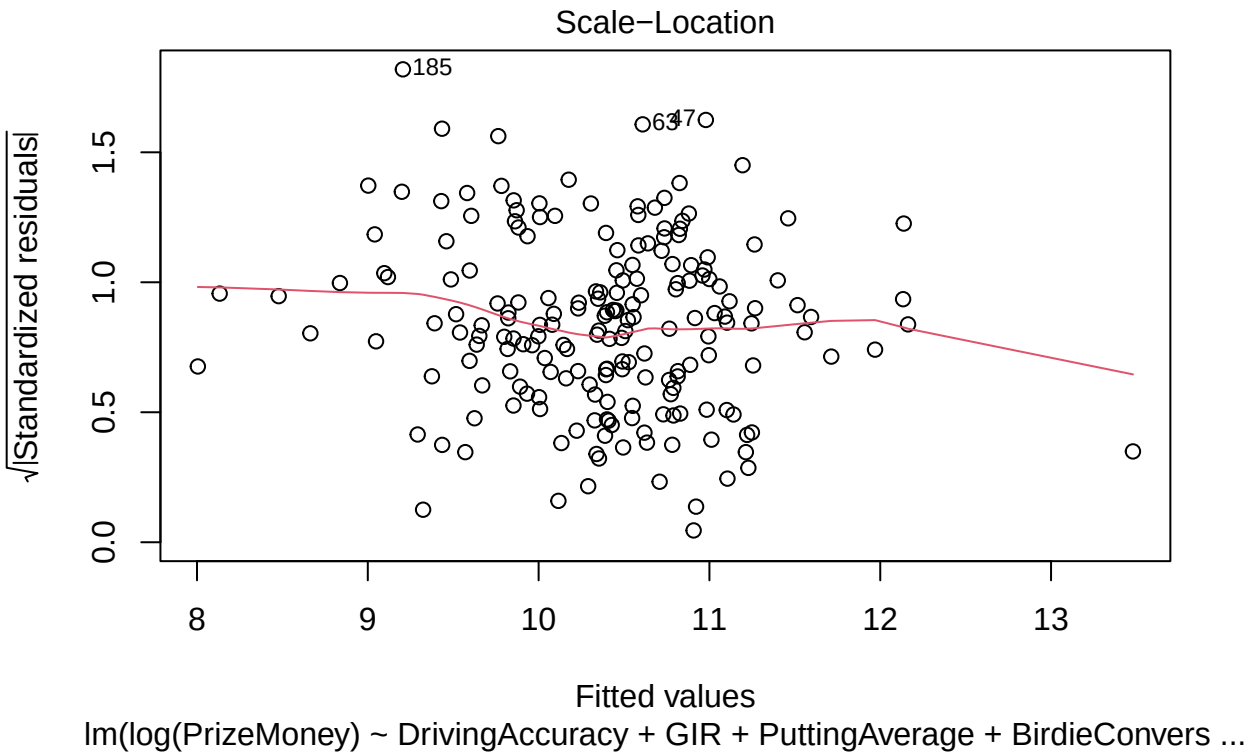
```r
# get power transformations on y and x
# check all variables
summary(powerTransform(cbind(pga$PrizeMoney, pga$DrivingAccuracy, pga$GIR, pga$PuttingAverage, pga$Birdi
```

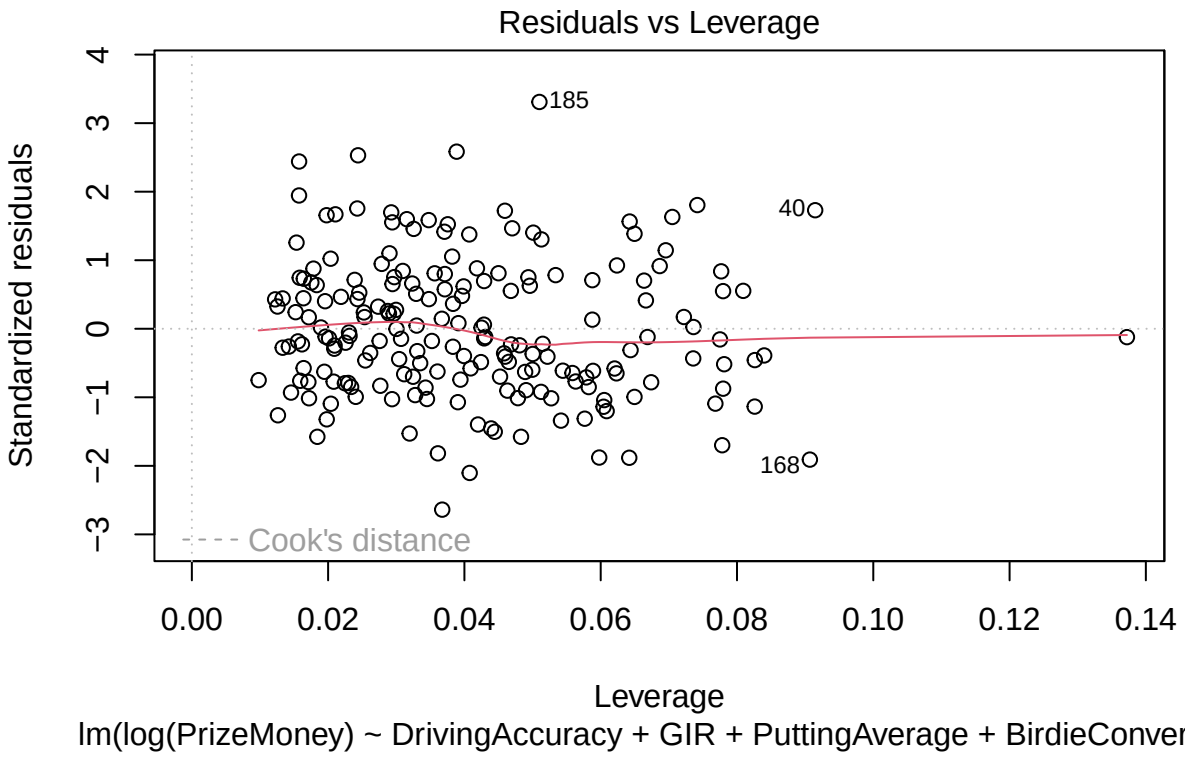
```
## bcPower Transformations to Multinormality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    0.0366           0      -0.0683       0.1415
## Y2    0.2749           1      -0.8979       1.4476
## Y3    1.5217           1       0.0893       2.9541
## Y4    1.0038           1      -3.5447       5.5524
## Y5    0.8910           1      -0.1521       1.9342
## Y6    0.9929           1       0.0543       1.9315
## Y7    0.6742           1      -0.7439       2.0923
## Y8   -0.0358           1      -3.2721       3.2004
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                           LRT df    pval
## LR test, lambda = (0 0 0 0 0 0 0 0) 12.85867  8 0.11681
##
## Likelihood ratio test that no transformations are needed
##                                           LRT df       pval
## LR test, lambda = (1 1 1 1 1 1 1 1) 338.2419  8 < 2.22e-16
```

```r
# only PrizeMoney needs to be log transformed

# update model to log the Y variable
m <- update(m, log(PrizeMoney)~.)

# plot
plot(m)
```


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
−3
−2
−1
0
1
2
3
4


Scale−Location
185
63
147
√|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 ...



The model that will be used will be the linear model, but the response variable (PrizeMoney) will be log-transformed. This decision was based off of the work done in part a. and via the `summary(powerTransform())` result.

To check that the model is valid, linearity, normality, homeoscedasticity, and sampling independence must be valid.

From the plot of residuals to fitted values, there is little trend in the plot and the data is generally randomly scattered, which implies there is a solid linearity in the data.

From the QQ plot, the points follow the line, which implies normality is strong.

The scale-location plot shows a little trend in the data, but it generally shows that the variance mostly stays consistent.

Sampling independence can be assumed because each observation is a different golfer.

c.

```
plot(m)

# high leverage
2*(7 + 1)/nrow(pga)
```

Based on the leverage plot, observations on row 40 and 168 have high leverage and have around 2 standard deviations away on their standardized residuals. This means these are possible points for bad high leverage. In addition, observation 185 has is has a big standard deviation compared to the rest of the data and drifts from normality a bit, so this is a point to investigate further into.

d.

```
summary(m)
```

```
##
## 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
```

```
car::vif(m)
```

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

Based on the summary stats of the linear model, only two variables are statistically significant: GIR and BirdieConversion. This means that the rest of the predictors and the intercepts are not statistically significant enough to be different from 0.

Another weakness would be possible collinearity between variables. GIR, PuttingAverage, and PuttsPerRound have a VIF above 5. This could be the leading cause to why there are insignificant p-values for the rest of the predictor variables.

e.

Removing all the insignificant variables all off the bat is not a good approach because those p-values could be skewed by possible collinearity. Collinearity causes the biased inflation of p-values for variables, and so, this would cause certain variables to seem insignificant but in reality, they would be important to the regression of the model.

Problem B

```
# do regression algo
bestmodel <- leaps::regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConversi

# using algebra manipulation, bic conversion is this function
BICToAIC <- function(bic,n,p)
{bic - (log(n) - 2)*p}

# make function to use regsubsets obj and provide aic
# and bic analysis
aicbicanalysis <- function(regsubObj, data)
{

  # get summary
  sumModel <- summary(regsubObj)

  n <- nrow(pga)

  # convert bic to aic
  aic <- BICToAIC(sumModel$bic,n,seq_along(sumModel$bic))

  # get bic
  bic <- sumModel$bic

  aic
  bic

  # get model with lowest aic and bic
  print(paste0("Best AIC Model: ",which(aic == min(aic))))
  print(paste0("Best BIC Model: ", which(bic == min(bic))))
  print(sumModel)

}

aicbicanalysis(bestmodel, pga)
```

```
## [1] "Best AIC Model: 5"
## [1] "Best BIC Model: 3"
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, pga, method = "e")
## 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 ) "*"        "*"
```

Based on AIC, the optimal model will be using 5 predictors: GIR, BirdieConversion, SandSaves, Scrambling, and PuttsPerRound.

Based on BIC, the optimal model will be using 3 predictors: GIR, BirdieConversion, and Scrambling.

b.

```
bestmodelback <- leaps::regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieConv

aicbicanalysis(bestmodelback,pga)
```

```
## [1] "Best AIC Model: 5"
## [1] "Best BIC Model: 3"
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, pga, method = "b")
## 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 ) "*"        "*"
```

Based on AIC, the best model via backward step regression is with 5 predictors: GIR, BirdieConversion, SandSaves, Scrambling, and PuttsPerRound.

Based on BIC, the best model via backward step regression is with 3 predictors: GIR, BirdieConversion, and Scrambling.

c.

```
bestmodelforward <- leaps::regsubsets(log(PrizeMoney) ~ DrivingAccuracy + GIR + PuttingAverage + BirdieC

aicbicanalysis(bestmodelforward,pga)
```

```
## [1] "Best AIC Model: 5"
## [1] "Best BIC Model: 4"
## Subset selection object
## Call: regsubsets.formula(log(PrizeMoney) ~ DrivingAccuracy + GIR +
##     PuttingAverage + BirdieConversion + SandSaves + Scrambling +
##     PuttsPerRound, pga, method = "f")
## 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 ) "*"         "*"
```

Based on AIC, the best model will be with 5 predictors: GIR, BirdieConversion, SandSaves, Scrambling, and PuttsPerRound.

Based on BIC, the best model will be with 4 predictors: GIR, BirdieConversion, Scrambling, and PuttsPerRound.

d.

The models differ from part c. as opposed to part a. and b. because the three parts utilize different algorithms. Part c's algorithm is forward stepping, which means that it takes the best performing model with each added variable. This can result in a different decision than the other algorithms because the BIC may let the algorithm choose a different model depending on whether starting with all the variables or starting with only 1.

e.

The final model should be a utilizing the 5-variable model. This is because all three algorithms agree on the AIC conclusion and because BIC tends to be biased toward more simplified models. Therefore, having 3 variables as the predictor variables does not feel like it would contribute much to the regression as opposed to 5 variables.

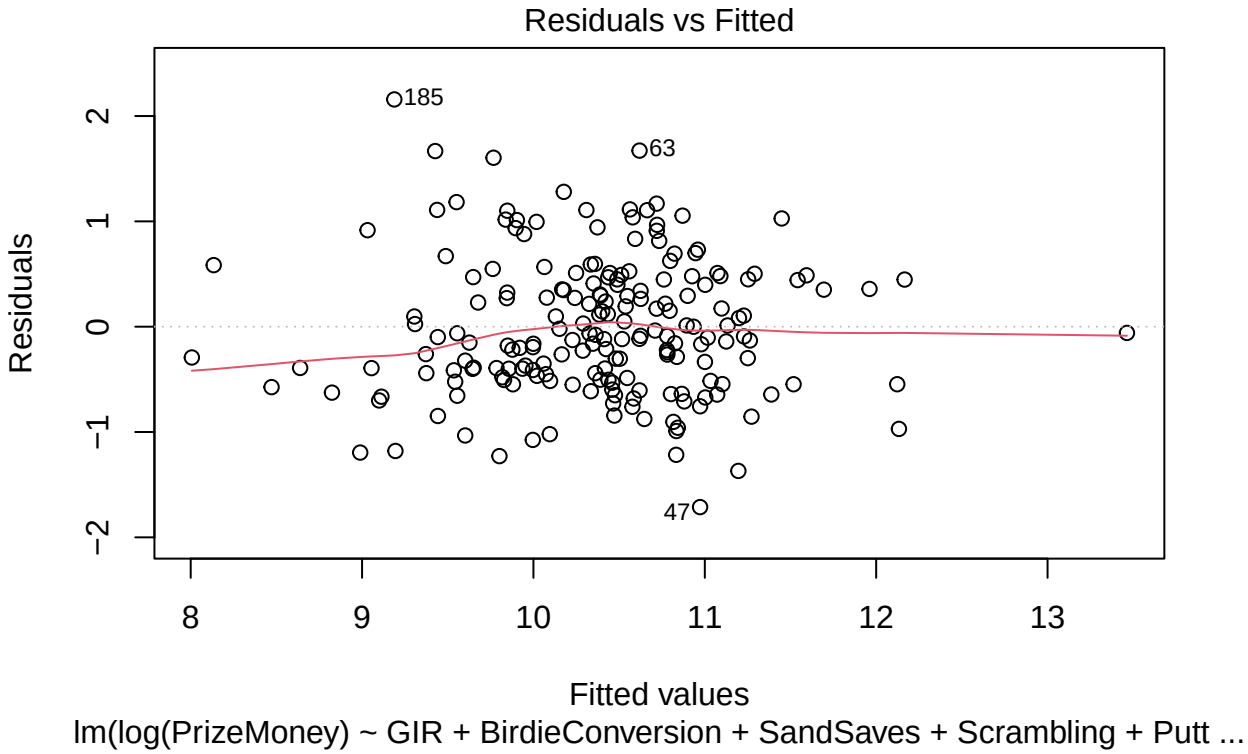
f.

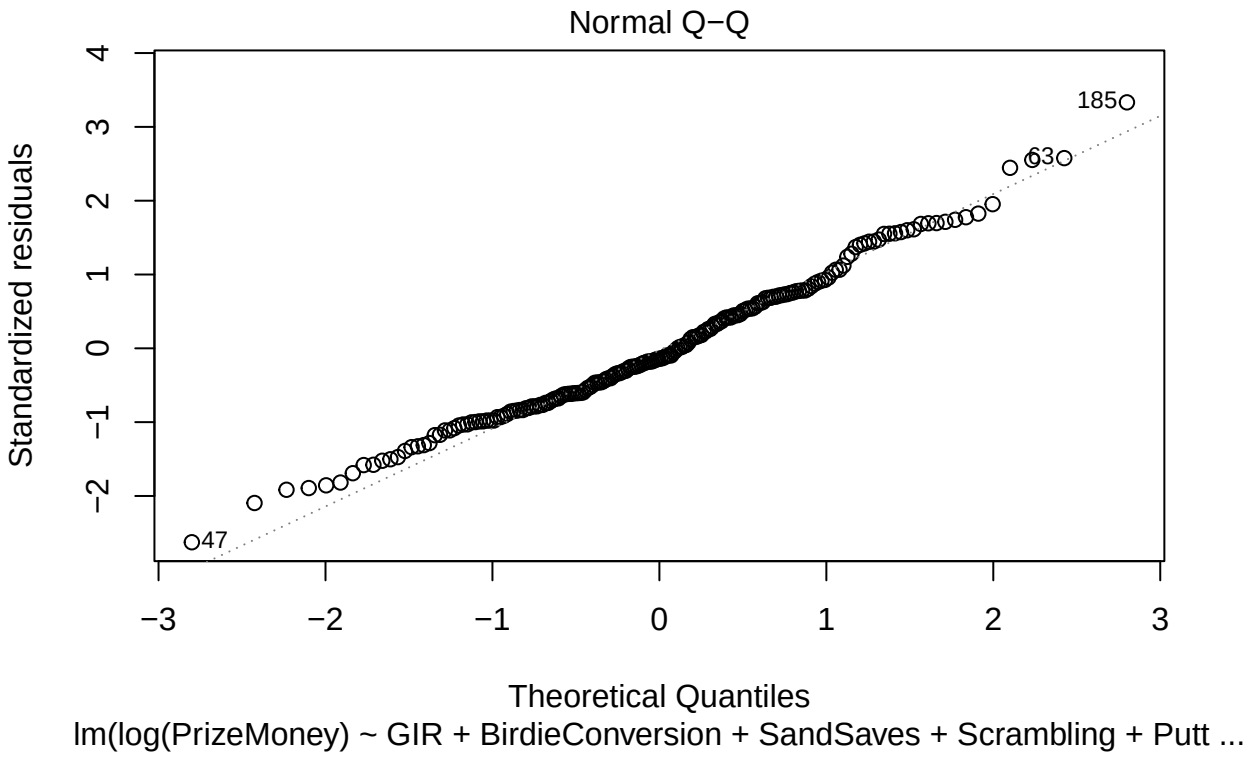
```
finalModel <- lm(log(PrizeMoney)~GIR + BirdieConversion + SandSaves + Scrambling + PuttsPerRound, pga)

summary(finalModel)
```

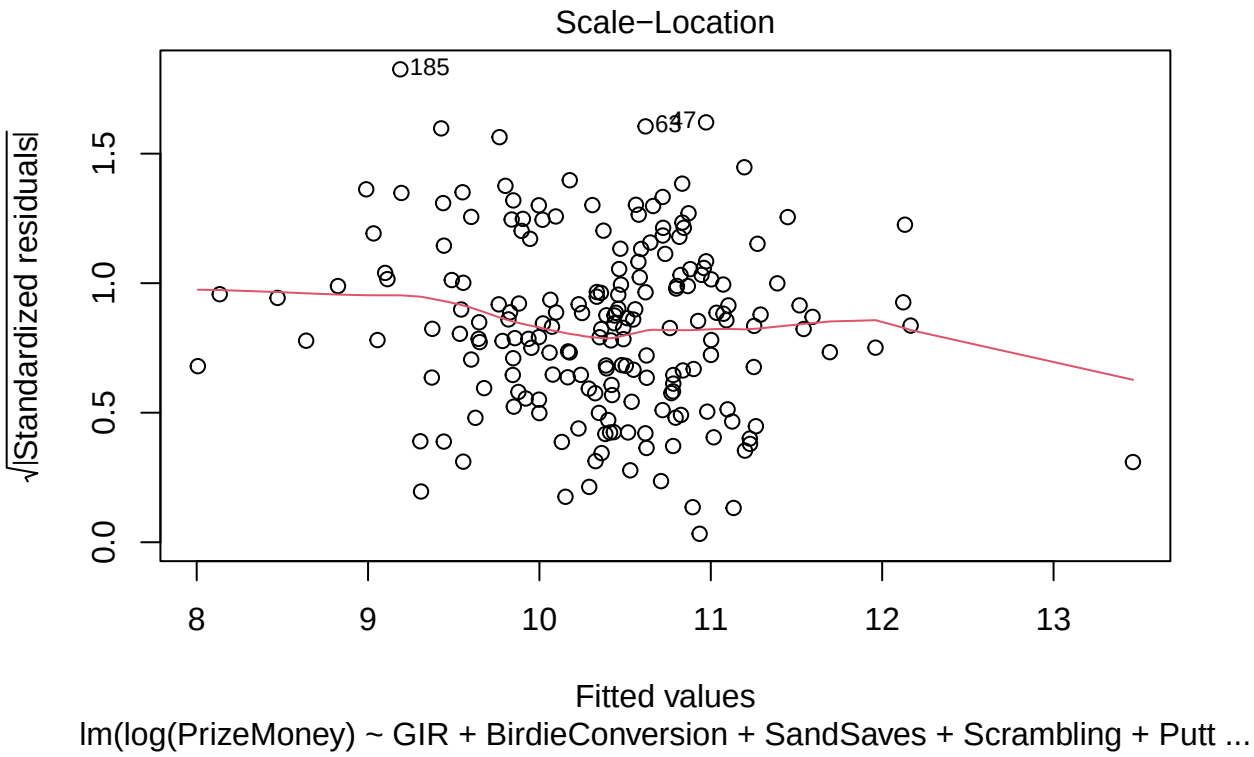
```
##
## Call:
## lm(formula = log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves +
##     Scrambling + PuttsPerRound, data = pga)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71291 -0.48168 -0.09097  0.44843  2.15763
##
## Coefficients:
##                   Estimate Std. 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
```

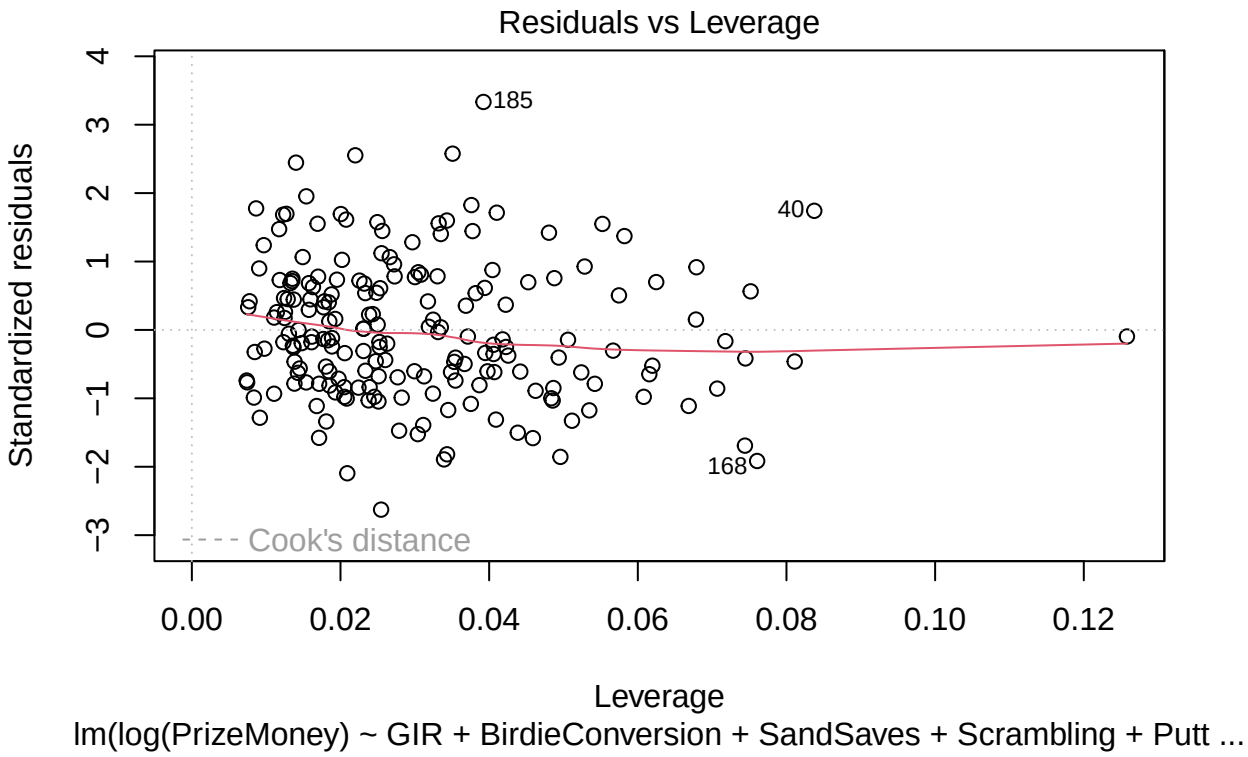
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6606 on 190 degrees of freedom
## Multiple R-squared:  0.5575, Adjusted R-squared:  0.5459
## F-statistic: 47.88 on 5 and 190 DF,  p-value: < 2.2e-16
```

```
plot(finalModel)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves + Scrambling + Putt ...
185
63
47
−3
−2
−1
0
1
2
3
4


Scale−Location
185
63
147
√|Standardized residuals|
0.0
0.5
1.0
1.5
8
9
10
11
12
13
Fitted values
lm(log(PrizeMoney) ~ GIR + BirdieConversion + SandSaves + Scrambling + Putt ...




```
car::vif(finalModel)
```

```
##              GIR BirdieConversion        SandSaves       Scrambling
##         2.730165         2.322693         1.441054         2.734765
##    PuttsPerRound
##         4.652336
```

The intercept is the average log of PrizeMoney a player makes assuming all the rest of the other variables are the same.

The rest of the variables describe the rate of the log of PrizeMoney as the variables increase. (I.e. log of PrizeMoney increases by about 0.197 and GIR increases by 1 unit and the other variables are controlled.) This interpretation can be applied and is similar for the rest of the variables.

Yes, it is still necessary to take these statistical results with extreme caution because the estimated slopes of some of the variables aren't statistically significant enough to differ from 0. In addition, there are a few possibly bad high leverage points that could have negatively contributed to the regression model. However, it seems that collinearity does not seem to be a prominent problem anymore compared to the 7 variable model. So, we can rule out that these variables are not drastically negatively affecting each others' significance.