DSC 423: Data Analysis & Regression

Assignment 5: Variable Screening

```
Name: Kushal Navghare
Student ID: 2116916
Honor Statement: I, Kushal Navghare, assure that I have completed this work independently.
The solutions given are entirely my own work.
```

1.The purpose of k-fold cross validation is often misunderstood.

a. How do you use cross validation to select a final (or production) model? Note: it is not the "best" of the k models you have built using cross validation. Ans: K-Fold cross validation technique splits the data into k equal parts and considers the k-1 folds as part of training set, the remaining fold is kept aside for testing. The process builds k versions of different model and the overall accuracy metric is aggregated at the end. For each iteration, a new model is trained completely independent of the previous iteration. This way, we get to see more generalized version of the model.

However, to use this technique efficiently, we perform multiple iteration of k-fold cross validation and test multiple assumptions in such way. For example, we can try different set of features for each iteration of k-fold validation. This way, we get different performance metric for each iteration and these metrics can be compared to decide the best version of model and set of features which yields more generalized and accurate predictions on future dataset.

2. The pgatour2006.csv dataset contains data for 196 players. The variables in the dataset are:

- Player's name
- PrizeMoney = average prize money per tournament
- DrivingAccuracy = percent of times a player is able to hit the fairway with his tee shot
- GIR = percent of time a player was able to hit the green within two or less than par (Greens in Regulation)
- BirdieConversion = percentage of times a player makes a birdie or better after hitting the green in regulation
- PuttingAverage = putting performance on those holes where the green was hit in regulation.
- PuttsPerRound= average number of putts per round (shots played on the green)

```r
# read file
raw_df <- read.csv(paste0(dir.path, 'data/pgatour2006.csv'))

# summary
dim(raw_df)
```

```
## [1] 196  11
```

```r
summary(raw_df)
```

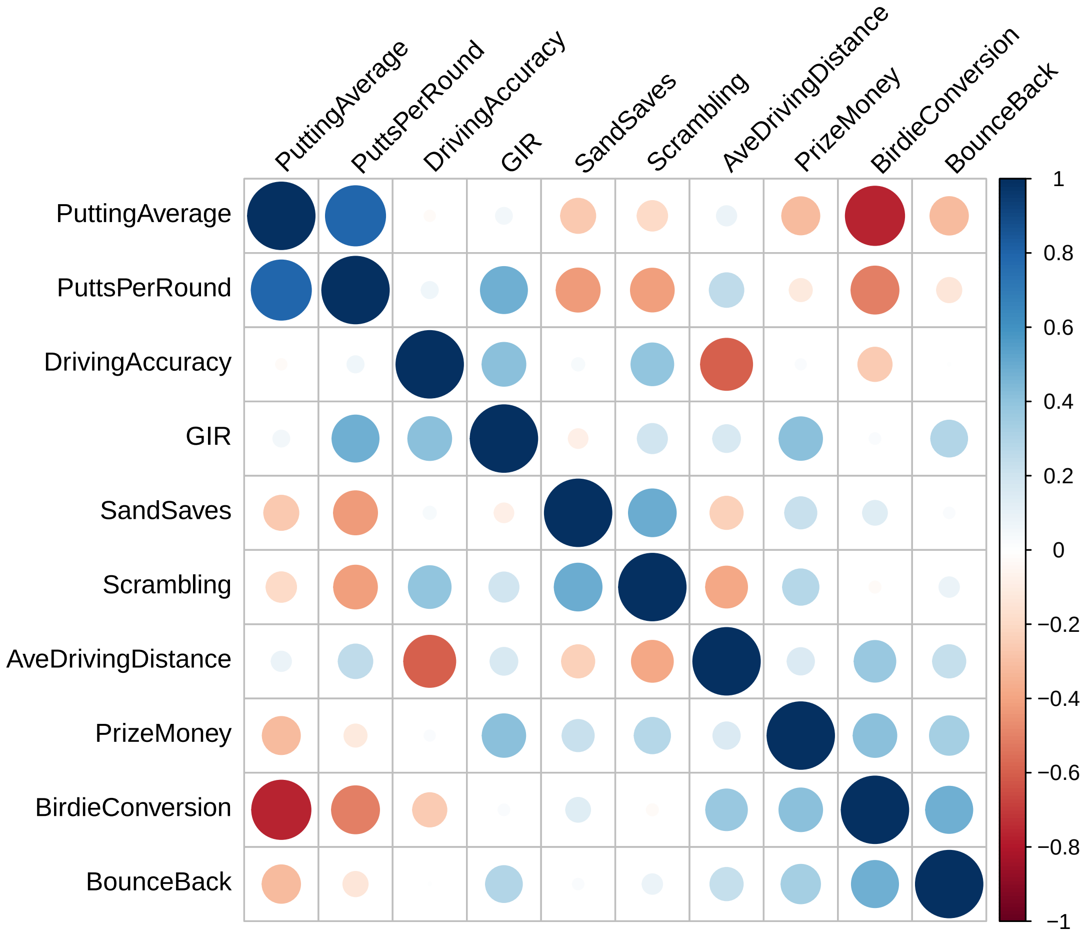
```
##      Name             PrizeMoney     AveDrivingDistance DrivingAccuracy
##  Length:196         Min.   :  2240   Min.   :265.9      Min.   :49.75
##  Class :character   1st Qu.: 17369   1st Qu.:283.6      1st Qu.:59.76
##  Mode  :character   Median : 36644   Median :288.2      Median :63.24
##                      Mean   : 50891   Mean   :289.5      Mean   :63.38
##                      3rd Qu.: 57915   3rd Qu.:295.5      3rd Qu.:66.97
##                      Max.   :662771  Max.   :319.6      Max.   :78.43
##       GIR        PuttingAverage  BirdieConversion   SandSaves
##  Min.   :56.87   Min.   :1.712   Min.   :23.17    Min.   :33.91
##  1st Qu.:63.52   1st Qu.:1.763   1st Qu.:27.51    1st Qu.:45.13
##  Median :65.36   Median :1.778   Median :29.01    Median :48.66
##  Mean   :65.19   Mean   :1.780   Mean   :28.98    Mean   :48.97
##  3rd Qu.:66.77   3rd Qu.:1.796   3rd Qu.:30.55    3rd Qu.:52.87
##  Max.   :74.15   Max.   :1.851   Max.   :35.66    Max.   :63.64
##    Scrambling      BounceBack    PuttsPerRound
##  Min.   :49.02   Min.   :12.29   Min.   :27.96
##  1st Qu.:55.26   1st Qu.:17.56   1st Qu.:28.91
##  Median :57.65   Median :19.62   Median :29.19
##  Mean   :57.49   Mean   :19.60   Mean   :29.20
##  3rd Qu.:59.46   3rd Qu.:21.31   3rd Qu.:29.48
##  Max.   :66.45   Max.   :25.93   Max.   :30.19
```

```
str(raw_df)
```

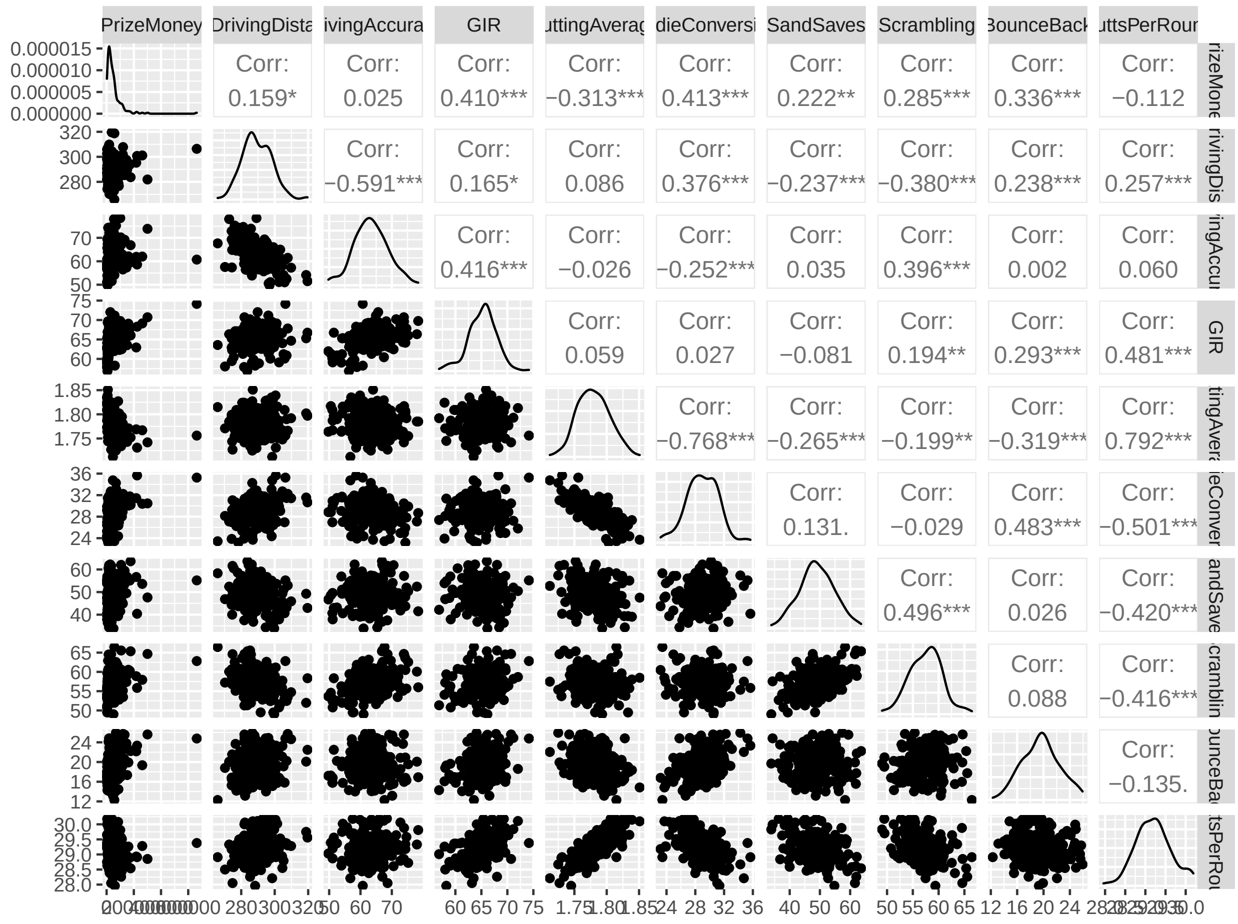
```
## ’data.frame’:    196 obs. of  11 variables:
##  $ Name              : chr  "Aaron Baddeley" "Adam Scott" "Alex Aragon" "Alex Cejka" ...
##  $ PrizeMoney        : int  60661 262045 3635 17516 16683 107294 50620 57273 86782 23396 ...
##  $ AveDrivingDistance: num  288 301 303 289 288 ...
##  $ DrivingAccuracy   : num  60.7 62 51.1 66.4 63.2 ...
##  $ GIR               : num  58.3 69.1 59.1 67.7 64 ...
##  $ PuttingAverage    : num  1.75 1.77 1.79 1.78 1.76 ...
##  $ BirdieConversion  : num  31.4 30.4 29.9 29.3 29.3 ...
##  $ SandSaves         : num  54.8 53.6 37.9 45.1 52.4 ...
##  $ Scrambling        : num  59.4 57.9 50.8 54.8 57.1 ...
##  $ BounceBack        : num  19.3 19.4 16.8 17.1 18.2 ...
##  $ PuttsPerRound     : num  28 29.3 29.2 29.5 28.9 ...
```

```
# check correlation
cor_df <- cor(raw_df %>% select_if(is.numeric))

corrplot(cor_df,
         type="full",
         order="hclust",
         tl.col="black", tl.srt=45)
```

a. Build a complete first-order model. Evaluate the model using 5-fold cross validation. If necessary, remove a non-significant variable and repeat until you have your final first-order model. Present the model.



```
# check pair plot
ggpairs(raw_df %>% select_if(is.numeric))
```



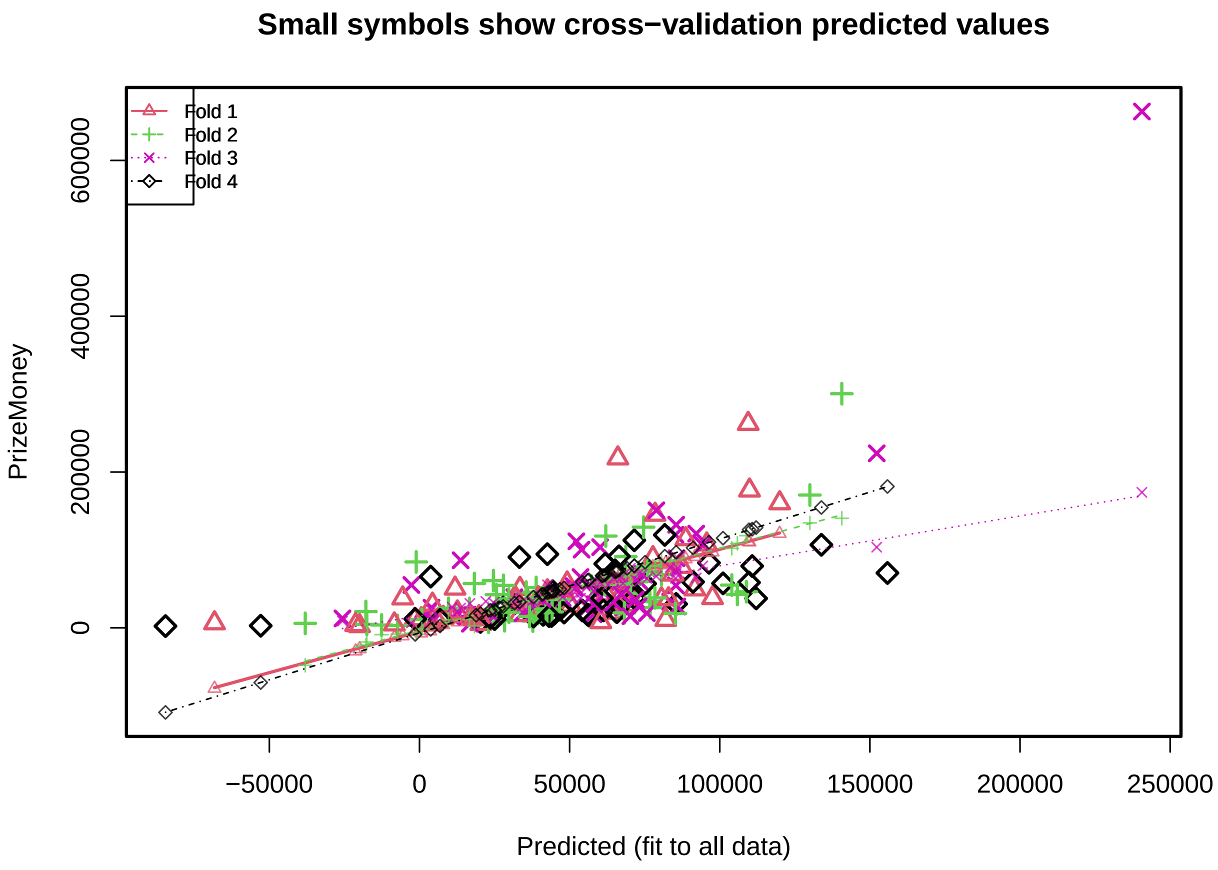
Let's start building a first-order model. here, we will try to predict based on Player's attributes, how much PrizeMoney can he make.

```
# select numeric columns
num_df <- raw_df %>%
  select_if(is.numeric)

# build a model (baseline)
base_model <- lm(PrizeMoney~DrivingAccuracy+GIR+BirdieConversion+Scrambling,
                 data = num_df)

summary(base_model)
```

```
##
## Call:
## lm(formula = PrizeMoney ~ DrivingAccuracy + GIR + BirdieConversion +
##     Scrambling, data = num_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -85429 -27959  -7833  15674 422173
##
## Coefficients:
```

```
##                    Estimate Std. Error t value             Pr(>|t|)
## (Intercept)      -1094996.9   109585.4  -9.992 < 0.0000000000000002 ***
## DrivingAccuracy     -1964.1      815.7  -2.408                0.017 *
## GIR                  9742.9     1465.9   6.646       0.000000000306 ***
## BirdieConversion    10670.5     1703.7   6.263       0.000000002439 ***
## Scrambling           5670.4     1239.4   4.575       0.000008556442 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50080 on 191 degrees of freedom
## Multiple R-squared:  0.3984, Adjusted R-squared:  0.3858
## F-statistic: 31.62 on 4 and 191 DF,  p-value: < 0.00000000000000022
```

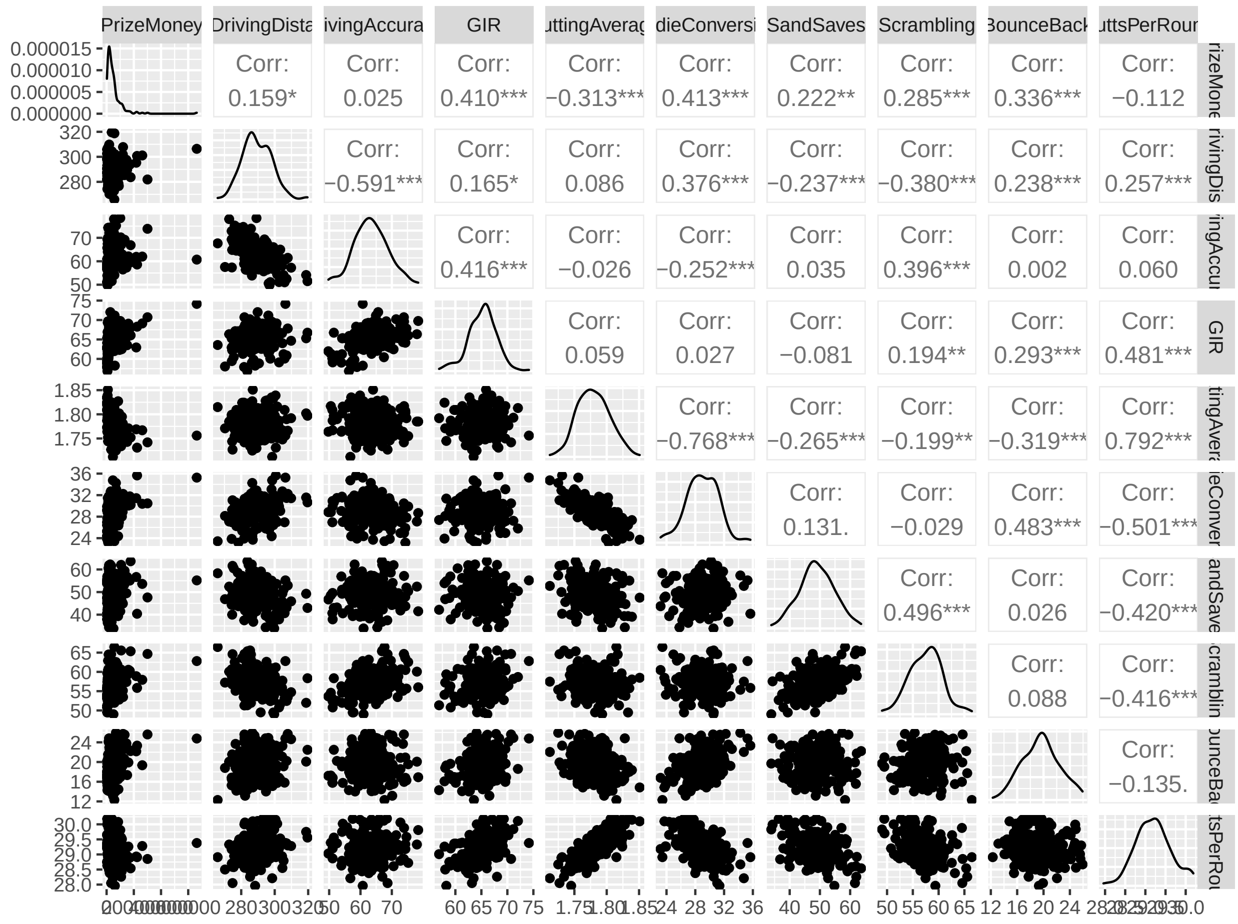
```
# cross validation
cv_model <- cv.lm(data = num_df,
                   form.lm = formula(PrizeMoney~DrivingAccuracy+GIR+BirdieConversion+Scrambling),
                  plotit = c("Observed", "Residual"), legend.pos = "topleft",
                  m = 4)
```



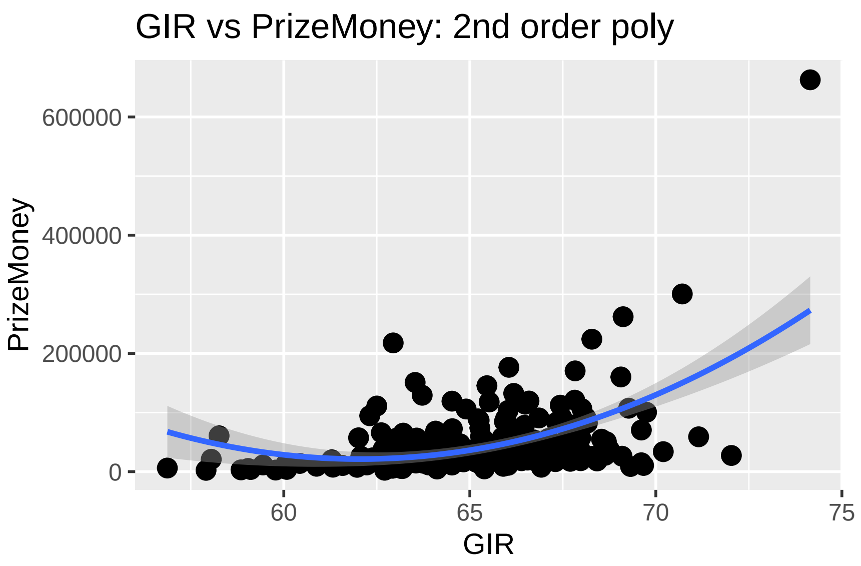
```
##
## fold 1
## Observations in test set: 49
##                    2        6        7        8       11        13        14
## Predicted   109477.3  95668.75 11848.51 49184.55  68816.83 42604.705  91689.33
## cvpred      110688.8  96458.47 12023.00 47004.96  65463.12 41791.003  91259.34
## PrizeMoney  262045.0 107294.00 50620.00 57273.00  29567.00 47172.000  49640.00
## CV residual 151356.2  10835.53 38597.00 10268.04 -35896.12  5380.997 -41619.34
##                    15        19        24        35       36        41
## Predicted    69849.63  50321.10  44308.45  97604.34 33458.09  67479.99
## cvpred       68686.54  48857.66  43583.53  97178.92 32535.57  65755.07
## PrizeMoney   53610.00  28658.00  27224.00  38455.00 50249.00  45752.00
## CV residual -15076.54 -20199.66 -16359.53 -58723.92 17713.43 -20003.07
##                    44        46        47        48        62        63
## Predicted   39605.047  33406.38  82019.24 16883.283 48247.908  66073.08
## cvpred      39856.075  29284.66  83149.91 10229.816 45768.137  63759.71
## PrizeMoney  38275.000  16630.00  10504.00 13262.000 43820.000 217748.00
## CV residual -1581.075 -12654.66 -72645.91  3032.184 -1948.137 153988.29
##                    64        65        72        77        79        81
## Predicted    3573.567 -1617.093  7933.424  80578.48 64599.621  20427.45
## cvpred      -4140.289 -4675.043  3789.004  80710.11 63335.418  15835.41
## PrizeMoney   5402.000 10528.000 13031.000  36918.00 57824.000   5265.00
## CV residual  9542.289 15203.043  9241.996 -43792.11 -5511.418 -10570.41
##                    86        88        93       101        105       108
## Predicted    62302.36  64624.61  82942.71 -19957.12  4329.9761 32159.220
## cvpred       61080.49  64835.00  82234.90 -27103.69   676.1089 32321.177
## PrizeMoney   43173.00  19594.00  37004.00   2426.00 30068.0000 37214.000
## CV residual -17907.49 -45241.00 -45230.90  29529.69 29391.8911  4892.823
##                   109       117        121       125       129        141
## Predicted    41437.30 109904.41   612.1707  60387.66 87301.233  -5594.059
## cvpred       40960.96 109999.73 -6875.2049  58143.74 87369.157 -10503.217
## PrizeMoney   26899.00 176523.00 11315.0000   7490.00 78489.000  38046.000
## CV residual -14061.96  66523.27 18190.2049 -50653.74 -8880.157  48549.217
##                   143        151       154       166      177      179
## Predicted    78469.64  -8367.754 -21266.86  88662.92 5883.315 77740.67
## cvpred       76274.45 -12614.460 -29983.60  88358.04 1881.503 78065.00
## PrizeMoney  145414.00   4667.000   3816.00 114055.00 9062.000 89770.00
## CV residual  69139.55  17281.460  33799.60  25696.96 7180.497 11705.00
##                   181       183       184      187       188       191
## Predicted   12640.251 18148.028 -68260.06 19818.04 119957.32  84465.30
## cvpred       7437.214 14798.705 -77832.22 18198.44 120988.55  84635.71
## PrizeMoney  20064.000 11309.000   6117.00 14098.00 160175.00  68613.00
## CV residual 12626.786 -3489.705  83949.22 -4100.44  39186.45 -16022.71
##
## Sum of squares = 96596380580    Mean square = 1971354706    n = 49
##
## fold 2
## Observations in test set: 49
##                    1          3        10        18       23        26
## Predicted   24622.82 -12614.814  66066.89 -17854.07 16635.15  76718.11
## cvpred      24367.73  -8732.368  65410.51 -22012.72 10726.34  79112.86
## PrizeMoney  60661.00   3635.000  23396.00  20911.00 24814.00  33782.00
## CV residual 36293.27  12367.368 -42014.51  42923.72 14087.66 -45330.86
##                    31       33        37        38        39        40
## Predicted    38505.95 38853.23  72507.12  39418.94  37750.67 18275.325
## cvpred       39414.17 35632.49  72598.35  37386.11  39375.94  2317.205
## PrizeMoney   15668.00 51770.00  59151.00  18345.00   8734.00 56873.000
## CV residual -23746.17 16137.51 -13447.35 -19041.11 -30641.94 54555.795
##                    45        57       58       60       61        66        69
## Predicted   108852.99  72816.45  74630.8 35557.39 27958.76 104011.92 25768.713
## cvpred      118360.09  77427.98  75903.9 34660.35 22674.39 102055.55 22856.158
## PrizeMoney   46377.00  43951.00 129234.0 45904.00 54477.00  54862.00 15840.000
## CV residual -71983.09 -33476.98  53330.1 11243.65 31802.61 -47193.55 -7016.158
##                    87       90        91        96       104      107       110
## Predicted    70314.13 140652.2  23010.77  28344.63  62046.51 68708.87  76838.35
## cvpred       67738.15 140599.3  24843.28  34573.45  62291.02 66804.74  73367.32
## PrizeMoney   56058.00 300555.0   7331.00   9149.00 117801.00 91406.00  25918.00
## CV residual -11680.15 159955.7 -17512.28 -25424.45  55509.98 24601.26 -47449.32
##                  111       115       119       122       127       132
## Predicted   25631.26 -7299.596 -38049.09  85227.71 -17537.14 105864.70
## cvpred      26071.30 -8031.807 -48065.02  87244.09 -18180.15 108492.78
## PrizeMoney  42589.00  3025.000   5777.00  18513.00   4444.00  42890.00
## CV residual 16517.70 11056.807  53842.02 -68731.09  22624.15 -65602.78
##                   133      139       140       144       145       146
## Predicted    9653.397 29749.07  36641.43  42158.68 57311.871  80585.81
## cvpred      10146.379 23776.13  34098.59  45635.01 55060.849  85889.20
## PrizeMoney  25135.000 37100.00  14527.00  24379.00 53634.000  68345.00
## CV residual 14988.621 13323.87 -19571.59 -21256.01 -1426.849 -17544.20
##                   149        152       158       162       164       165
## Predicted    68248.89   777.2415  29781.93  44174.86  78409.11  42447.56
## cvpred       74443.22 -1741.1376  34302.33  45515.86  78936.21  39084.07
## PrizeMoney   19200.00 10715.0000  19973.00  20502.00  38471.00  19997.00
## CV residual -55243.22 12456.1376 -14329.33 -25013.86 -40465.21 -19087.07
##                  171       185       186       189       192
## Predicted   46676.16 -1091.278 75922.392 61704.098 130009.74
## cvpred      46078.13 -6508.421 79428.195 61141.544 134662.32
## PrizeMoney  36289.00 84604.000 72623.000 55581.000 170460.00
## CV residual -9789.13 91112.421 -6805.195 -5560.544  35797.68
##
## Sum of squares = 87964361969    Mean square = 1795191061    n = 49
##
## fold 3
## Observations in test set: 49
##                     4        9        12        16        22        25
## Predicted    57997.68 13704.64  58116.01  67755.28  92159.17  83467.39
## cvpred       47083.19 25316.09  56683.00  57165.31  64136.75  77552.07
## PrizeMoney   17516.00 86782.00  44080.00  26129.00 120927.00  33471.00
## CV residual -29567.19 61465.91 -12603.00 -31036.31  56790.25 -44081.07
##                     29       49        51        52        54        55
## Predicted   73158.9954 53620.70  85483.00  85259.30  10868.55  55993.83
## cvpred      59390.6298 37954.14  69402.47  63570.95  26701.05  54386.94
## PrizeMoney  60073.0000 65174.00 132327.00 119444.00  13865.00  26301.00
## CV residual   682.3702 27219.86  62924.53  55873.05 -12836.05 -28085.94
##                    68        73       74        76        83        84
## Predicted   31592.603  60155.02 66406.01  4011.872  85878.36 -2707.036
## cvpred      31848.156  57973.17 46632.71 22372.642  77016.62  6381.382
## PrizeMoney  39356.000 103594.00 57216.00 25804.000  27361.00 55014.000
## CV residual  7507.844  45620.83 10583.29  3431.358 -49655.62 48632.618
##                    89        92        94        95       99       112
## Predicted    85391.54  53990.60 37606.494  44905.15 51334.21  62841.43
## cvpred       73421.94  45521.39 34691.538  50572.11 37957.41  55475.66
## PrizeMoney   54513.00 100398.00 27673.000  29296.00 53530.00  18494.00
## CV residual -18908.94  54876.61 -7018.538 -21276.11 15572.59 -36981.66
##                    113       114       126       128       131       134
## Predicted   -25647.112 34303.598  75723.85  16785.51  4170.275  42643.18
## cvpred        8591.756 27715.851  63314.68  31291.29 12589.382  55185.69
## PrizeMoney   12110.000 18721.000  18838.00   5285.00  8272.000  26532.00
## CV residual   3518.244 -8994.851 -44476.68 -26006.29 -4317.382 -28653.69
##                  135       136       137      142       147      148       150
## Predicted   85613.89 47703.622  22086.58 152277.5  24678.33 12734.95  52252.62
## cvpred      72113.98 39626.527  34002.48 103535.2  32263.01 20894.45  47979.35
## PrizeMoney  89312.00 37869.000  11376.00 224027.0  14558.00 16455.00 111028.00
## CV residual 17198.02 -1757.527 -22626.48 120491.8 -17705.01 -4439.45  63048.65
##                   155       156       157        159       167       168
## Predicted   51640.637  72008.45  64585.26 74200.7053  73003.79  70228.66
## cvpred      49754.164  61398.99  64197.61 68608.6551  69400.58  76351.45
## PrizeMoney  51005.000  36428.00  32843.00 69173.0000  27657.00  15012.00
## CV residual  1250.836 -24970.99 -31354.61   564.3449 -41743.58 -61339.45
##                   169       173       174       176      178       194
## Predicted    67233.81  94382.72  78884.75 41009.953 240598.2  58267.25
## cvpred       64564.30  79527.88  66143.48 35689.817 173881.2  48096.42
## PrizeMoney   42958.00 105997.00 150889.00 36861.000 662771.0  30344.00
## CV residual -21606.30  26469.12  84745.52  1171.183 488889.8 -17752.42
##
## Sum of squares = 310309714409    Mean square = 6332851314    n = 49
##
## fold 4
## Observations in test set: 49
##                     5        17        20        21        27        28
## Predicted    41197.02 18842.600  48163.74 110805.29  40706.59 112118.48
## cvpred       43253.98 16187.617  51281.29 126698.61  42840.73 128710.82
## PrizeMoney   16683.00 11989.000  19683.00  79316.00  20322.00  37751.00
## CV residual -26570.98 -4198.617 -31598.29 -47382.61 -22518.73 -90959.82
##                   30        32        34        42         43        50
## Predicted   42583.34  71535.55  69123.10  37937.75 31645.9420  43950.75
## cvpred      44966.45  79449.43  76393.68  39554.57 31745.1894  46395.10
## PrizeMoney  94571.00 112443.00  37735.00  14499.00 31371.0000  15187.00
## CV residual 49604.55  32993.57 -38658.68 -25055.57  -374.1894 -31208.10
##                    53        56       59        67         70        71
## Predicted   65130.749  54202.83 101098.8  85466.71  -84595.47  71541.16
## cvpred      71920.593  58597.32 115125.9  97001.67 -108563.34  79952.75
## PrizeMoney  73819.000  22340.00  57092.0  30656.00    2240.00  38188.00
## CV residual  1898.407 -36257.32 -58033.9 -66345.67  110803.34 -41764.75
##                   75        78        80        82        85        97
## Predicted   61852.81  20292.99 26495.839  56013.17  65542.26 -52881.74
## cvpred      67896.86  18270.79 25823.173  61037.14  72323.22 -70246.41
## PrizeMoney  82196.00   7583.00 24724.000  16927.00  20612.00   2692.00
## CV residual 14299.14 -10687.79 -1099.173 -44110.14 -51711.22  72938.41
##                    98       100       102       103       106       116
## Predicted    56401.66  91103.16  155850.0  44848.86 109741.65  96414.63
## cvpred       61685.45 102719.98  181435.3  46965.61 125906.95 110007.84
## PrizeMoney   15964.00  58953.00   70421.0  18085.00  58189.00  83483.00
## CV residual -45721.45 -43766.98 -111014.3 -28880.61 -67717.95 -26524.84
##                   118       120       123       124       130       138
## Predicted    65781.32 27882.972 44476.824  61248.14  75144.02  46727.73
## cvpred       73044.04 27512.094 46936.669  67306.88  84286.33  49557.08
## PrizeMoney   20188.00 26123.000 41390.000  22467.00  56693.00  23403.00
## CV residual -52856.04 -1389.094 -5546.669 -44839.88 -27593.33 -26154.08
##                   153         160      161       163       170       172
## Predicted    81690.36 44337.06034 66430.47  60836.73 -1424.039 133847.00
## cvpred       92037.17 47113.71067 73828.37  66619.87 -8474.102 154466.12
## PrizeMoney  119240.00 47046.00000 91808.00  56305.00 11421.000 106577.00
## CV residual  27202.83   -67.71067 17979.63 -10314.87 19895.102 -47889.12
##                   175       180       182       190      193       195      196
## Predicted    43164.21  3729.020  25025.13  6840.089 23591.71  60791.44 33277.00
## cvpred       46196.38 -2005.631  22976.60  2305.937 22542.64  66954.52 33238.05
## PrizeMoney   15098.00 65783.000  11187.00 10354.000 12803.00  38043.00 90824.00
## CV residual -31098.38 67788.631 -11789.60  8048.063 -9739.64 -28911.52 57585.95
##
## Sum of squares = 95370706618    Mean square = 1946340951    n = 49
##
## Overall (Sum over all 49 folds)
##         ms
## 3011434508
```

```r
# pair plot
ggpairs(num_df, size=.5)
```

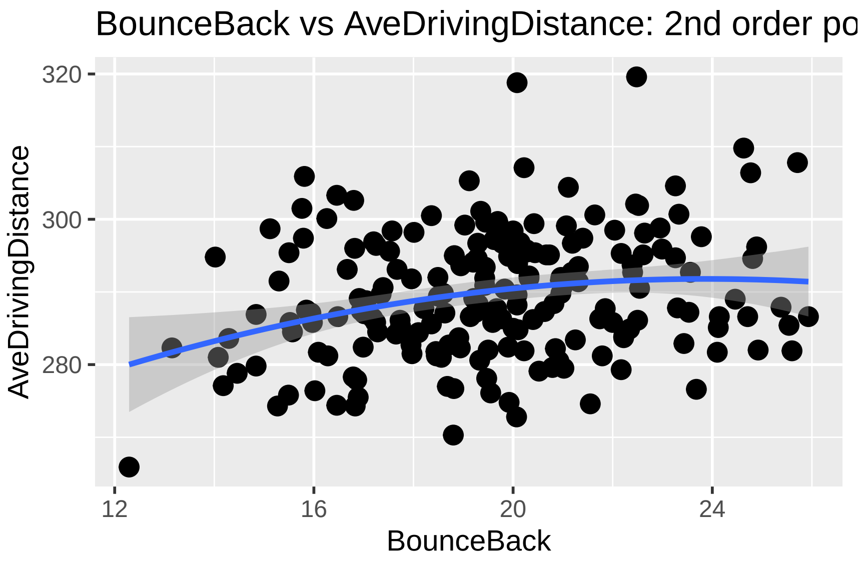
b. Evaluate scatterplots to determine which second-order terms should be tested. Test them using 5-fold cross validation and add them one-by-one until you arrive at a model you feel is appropriate. Present the model.



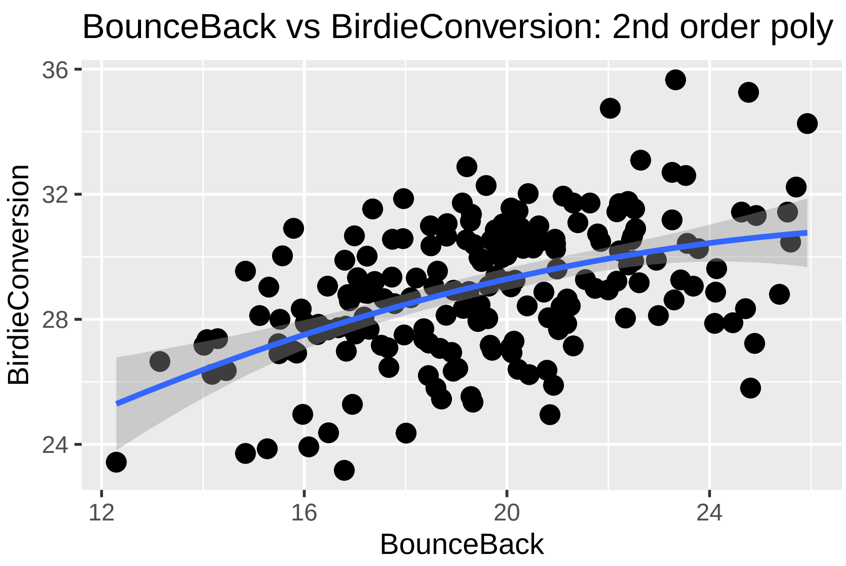
```r
# scatter plots
plt_1 <- ggplot(data = num_df, aes(x = GIR, y = PrizeMoney))
plt_1 + geom_point(size=3) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE)) +
  ggtitle("GIR vs PrizeMoney: 2nd order poly")
```



```
plt_1 <- ggplot(data = num_df, aes(x = BounceBack, y = AveDrivingDistance))
plt_1 + geom_point(size=3) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE)) +
  ggtitle("BounceBack vs AveDrivingDistance: 2nd order poly")
```



```
plt_1 <- ggplot(data = num_df, aes(x = BounceBack, y = BirdieConversion))
plt_1 + geom_point(size=3) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE)) +
  ggtitle("BounceBack vs BirdieConversion: 2nd order poly")
```



```
# try second order model
sec_ordr_model <- lm(PrizeMoney~(SandSaves+GIR+BirdieConversion)^2, data = num_df)

summary(sec_ordr_model)
```

```
##
## Call:
## lm(formula = PrizeMoney ~ (SandSaves + GIR + BirdieConversion)^2,
##     data = num_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -131379  -21075   -6305   14861  184661
##
## Coefficients:
##                              Estimate Std. Error t value            Pr(>|t|)
## (Intercept)                 8445009.3  1012279.1   8.343   0.0000000000000150
## SandSaves                    -40082.5    13989.8  -2.865             0.004640
## GIR                         -115968.5    14125.1  -8.210   0.0000000000000338
## BirdieConversion            -285772.6    30201.7  -9.462 < 0.0000000000000002
## SandSaves:GIR                   259.5      185.0   1.402             0.162445
## SandSaves:BirdieConversion      865.8      223.3   3.877             0.000146
## GIR:BirdieConversion           3891.4      415.9   9.358 < 0.0000000000000002
##
## (Intercept)                ***
## SandSaves                  **
## GIR                        ***
## BirdieConversion           ***
## SandSaves:GIR
## SandSaves:BirdieConversion ***
## GIR:BirdieConversion       ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 41370 on 189 degrees of freedom
## Multiple R-squared:  0.5938, Adjusted R-squared:  0.5809
## F-statistic: 46.04 on 6 and 189 DF,  p-value: < 0.00000000000000022
```

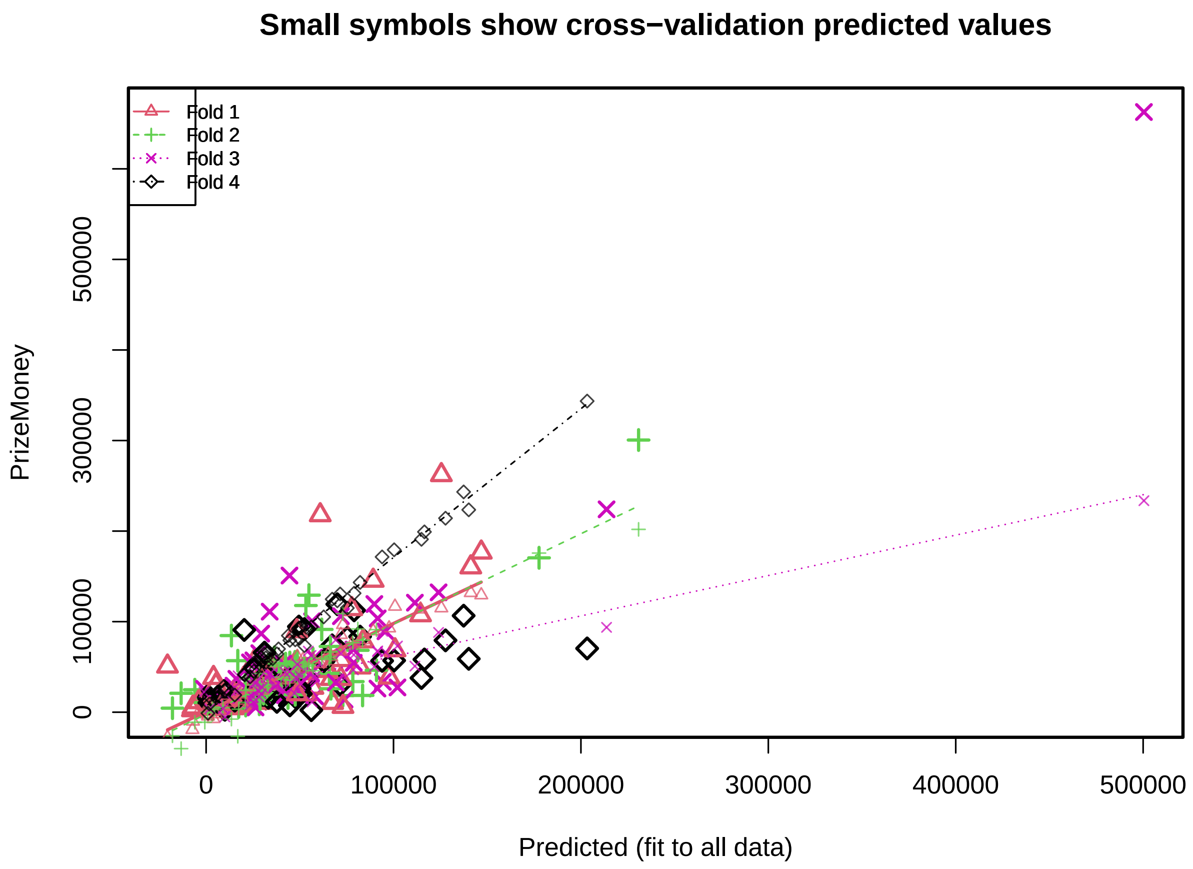
```
# add terms
sec_df <- num_df %>%
  mutate(AveDrvDistSec = AveDrivingDistance^2,
         DrvAccSec = DrivingAccuracy^2,
         GIRSec = GIR^2,
         BouncBckSec = BounceBack^2)

model_2 <- lm(PrizeMoney~BouncBckSec+GIRSec+BounceBack+BirdieConversion+GIR,
              data = sec_df)

summary(model_2)
```

```
##
## Call:
## lm(formula = PrizeMoney ~ BouncBckSec + GIRSec + BounceBack +
##     BirdieConversion + GIR, data = sec_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113508  -25116   -3667   13324  304616
##
## Coefficients:
##                   Estimate Std. Error t value       Pr(>|t|)
## (Intercept)      6290703.7  1183754.1   5.314 0.000000298288 ***
## BouncBckSec         1147.4      344.4   3.332        0.00104 **
## GIRSec              1608.6      283.6   5.672 0.000000051837 ***
## BounceBack        -44174.0    13709.9  -3.222        0.00150 **
## BirdieConversion   11596.6     1773.4   6.539 0.000000000557 ***
## GIR              -199534.9    36826.5  -5.418 0.000000180825 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 47140 on 190 degrees of freedom
## Multiple R-squared:  0.4699, Adjusted R-squared:  0.4559
## F-statistic: 33.68 on 5 and 190 DF,  p-value: < 0.00000000000000022
```

```
# add interaction terms
thr_df <- sec_df %>%
  mutate(AvgDrvD_BouncBck = AveDrivingDistance*BounceBack,
         DrvAcc_GIR = DrivingAccuracy*GIR,
         PuttAvg_Gir = PuttingAverage*GIR,
         PuttAvg_BouncBck = PuttingAverage*BounceBack,
         PuttAvg_Scrmb = PuttingAverage*Scrambling,
         Scrmb_BouncBck = Scrambling*BounceBack,
         SndSvs_Scrmb = SandSaves*Scrambling) %>%
  dplyr::select(-c(Scrambling, PuttsPerRound, BounceBack))

model_3 <- lm(PrizeMoney~., data = thr_df)

summary(model_3)
```

```
##
## Call:
## lm(formula = PrizeMoney ~ ., data = thr_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -132898  -19469   -1252   13954  162350
##
## Coefficients:
##                         Estimate   Std. Error t value      Pr(>|t|)
## (Intercept)         -21236344.23   6162605.32  -3.446      0.000710 ***
## AveDrivingDistance      29741.03     18419.43   1.615      0.108157
## DrivingAccuracy         52351.48     15224.66   3.439      0.000728 ***
## GIR                    121479.42     88612.67   1.371      0.172131
## PuttingAverage       12494032.77   2886857.42   4.328 0.00002506254 ***
## BirdieConversion         9052.88      2940.44   3.079      0.002408 **
## SandSaves              -11296.81      7317.91  -1.544      0.124432
## AveDrvDistSec             -59.25        32.93  -1.799      0.073680 .
## DrvAccSec                 347.43       104.76   3.316      0.001105 **
## GIRSec                   1967.91       312.83   6.291 0.00000000238 ***
## BouncBckSec               611.47       320.00   1.911      0.057633 .
## AvgDrvD_BouncBck          217.77       143.47   1.518      0.130815
## DrvAcc_GIR              -1506.39       305.74  -4.927 0.00000189927 ***
## PuttAvg_Gir           -153217.47     47078.24  -3.255      0.001359 **
## PuttAvg_BouncBck       -84807.79     26861.57  -3.157      0.001871 **
## PuttAvg_Scrmb          -16577.83      5245.97  -3.160      0.001854 **
## Scrmb_BouncBck           1124.35       346.55   3.244      0.001406 **
## SndSvs_Scrmb              212.28       126.21   1.682      0.094337 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 39130 on 178 degrees of freedom
## Multiple R-squared:  0.6578, Adjusted R-squared:  0.6251
## F-statistic: 20.12 on 17 and 178 DF,  p-value: < 0.00000000000000022
```

```
# cross validation
cv_sec_model <- CVlm(data = thr_df,
                     form.lm = formula(PrizeMoney~.),
                     m = 4)
```



```
##
## fold 1
## Observations in test set: 49
##                     2          6         7        8       11       13        14
## Predicted   125525.9 114423.954 -20615.93 26031.22 40232.20 35070.493  82142.73
## cvpred      115000.8 113737.133 -28097.96 29458.10 35286.23 54060.569  77070.04
## PrizeMoney  262045.0 107294.000  50620.00 57273.00 29567.00 47172.000  49640.00
## CV residual 147044.2  -6443.133  78717.96 27814.90 -5719.23 -6888.569 -27430.04
##                   15        19        24        35        36        41
## Predicted   47453.28 44068.379  56970.23  97645.69 41954.907  65158.24
## cvpred      43601.67 36304.491  58455.55  92853.37 41647.658  64109.49
## PrizeMoney  53610.00 28658.000  27224.00  38455.00 50249.000  45752.00
## CV residual 10008.33 -7646.491 -31231.55 -54398.37  8601.342 -18357.49
##                    44         46        47        48       62        63
## Predicted   32356.856   535.8664  67499.17  14358.32 29923.08  60889.14
## cvpred      33206.462 -2262.0122  56530.42  27349.05 24039.38  53400.27
## PrizeMoney  38275.000 16630.0000  10504.00  13262.00 43820.00 217748.00
## CV residual  5068.538 18892.0122 -46026.42 -14087.05 19780.62 164347.73
##                     64       65        72        77        79        81
## Predicted    -7005.432  29842.5  1031.250  71895.82 72036.846  17390.88
## cvpred      -10046.184  40079.6 -4462.864  85602.14 67171.379  21238.31
## PrizeMoney    5402.000  10528.0 13031.000  36918.00 57824.000   5265.00
## CV residual  15448.184 -29551.6 17493.864 -48684.14 -9347.379 -15973.31
##                    86         88       93       101       105      108
## Predicted    51655.72  53070.12  66793.97  -7305.663  5242.526 20068.09
## cvpred       57954.25  47833.94  61091.90 -19294.576 -1960.122 12845.57
## PrizeMoney   43173.00  19594.00  37004.00   2426.000 30068.000 37214.00
## CV residual -14781.25 -28239.94 -24087.90  21720.576 32028.122 24368.43
##                   109       117       121       125       129       141
## Predicted   35370.016 146822.51  2136.900  18752.07 83830.512  3959.518
## cvpred      29086.304 129385.82  7380.533  19768.63 79810.694 -7281.394
## PrizeMoney  26899.000 176523.00 11315.000   7490.00 78489.000 38046.000
## CV residual -2187.304  47137.18  3934.467 -12278.63 -1321.694 45327.394
##                   143       151       154       166       177      179
## Predicted    89154.48 14435.659  1564.737  77737.54  9775.817 48427.42
## cvpred       89654.58 10580.604  8259.098  75069.58 18408.556 48737.12
## PrizeMoney  145414.00  4667.000  3816.000 114055.00  9062.000 89770.00
## CV residual  55759.42 -5913.604 -4443.098  38985.42 -9346.556 41032.88
##                   181        183       184       187       188       191
## Predicted    48649.53 -3926.8235  73029.20 15505.579 141227.09 100800.37
## cvpred       61361.97  -118.7371  96588.58 16109.054 131919.75 116793.29
## PrizeMoney   20064.00 11309.0000   6117.00 14098.000 160175.00  68613.00
## CV residual -41297.97 11427.7371 -90471.58 -2011.054  28255.25 -48180.29
##
## Sum of squares = 95938616703    Mean square = 1957930953    n = 49
##
## fold 2
## Observations in test set: 49
##                     1         3        10        18         23        26
## Predicted   36443.207  8990.173  49794.58 -13328.28  -6017.608  78279.55
## cvpred      53573.944  4989.405  41177.98 -40236.94 -11130.060  76882.30
## PrizeMoney  60661.000  3635.000  23396.00  20911.00  24814.000  33782.00
## CV residual  7087.056 -1354.405 -17781.98  61147.94  35944.060 -43100.30
##                    31       33       37        38        39        40        45
## Predicted    43691.33 33103.53 57025.21  73004.63  28391.76  16900.04  90905.52
## cvpred       45430.75 30472.50 45769.88 108845.64  28199.55 -26600.05  91135.25
## PrizeMoney   15668.00 51770.00 59151.00  18345.00   8734.00  56873.00  46377.00
## CV residual -29762.75 21297.50 13381.12 -90500.64 -19465.55  83473.05 -44758.25
##                    57        58        60        61       66          69
## Predicted   46332.006  54852.64 52174.823 48115.306 44672.96    751.02360
## cvpred      49820.926  52424.94 54003.906 55556.455 35987.39    -53.90664
## PrizeMoney  43951.000 129234.00 45904.000 54477.000 54862.00 15840.00000
## CV residual -5869.926  76809.06 -8099.906 -1079.455 18874.61 15893.90664
##                   87        90        91       96       104      107       110
## Predicted   48639.80 230765.03  21130.63 4456.974  53246.11 61693.13  66623.51
## cvpred      38707.01 201895.96  23992.64 3049.633  50622.10 59495.95  61216.81
## PrizeMoney  56058.00 300555.00   7331.00 9149.000 117801.00 91406.00  25918.00
## CV residual 17350.99  98659.04 -16661.64 6099.367  67178.90 31910.05 -35298.81
##                   111       115      119       122       127       132
## Predicted   38989.786  3366.764 17397.43  83498.36 -17950.83  65797.21
## cvpred      50677.369  5515.055  3280.68  79498.29 -26063.14  58401.08
## PrizeMoney  42589.000  3025.000  5777.00  18513.00   4444.00  42890.00
## CV residual -8088.369 -2490.055  2496.32 -60985.29  30507.14 -15511.08
##                   133      139      140       144      145       146       149
## Predicted    34373.15 20899.06  28790.1 26022.553 42402.33  80952.96  47554.53
## cvpred       57694.89 10879.31  29556.6 30642.886 35874.50  91948.32  40768.97
## PrizeMoney   25135.00 37100.00  14527.0 24379.000 53634.00  68345.00  19200.00
## CV residual -32559.89 26220.69 -15029.6 -6263.886 17759.50 -23603.32 -21568.97
##                     152       158       162       164       165       171
## Predicted     -687.8284  28364.99  30630.83 47582.312  32528.86 40038.804
## cvpred      -11118.1922  42743.63  30839.58 42571.392  30242.77 38855.571
## PrizeMoney   10715.0000  19973.00  20502.00 38471.000  19997.00 36289.000
## CV residual  21833.1922 -22770.63 -10337.58 -4100.392 -10245.77 -2566.571
##                   185      186        189        192
## Predicted   13500.206 66356.28 47431.9535 177650.813
## cvpred      -7620.203 66530.14 55411.5171 175718.721
## PrizeMoney  84604.000 72623.00 55581.0000 170460.000
## CV residual 92224.203  6092.86   169.4829  -5258.721
##
## Sum of squares = 68171605068    Mean square = 1391257246    n = 49
##
## fold 3
## Observations in test set: 49
##                     4        9        12        16       22        25       29
## Predicted    58148.81 29394.50 32772.725  91434.65 111402.8  94289.34 57791.63
## cvpred       36372.34 20270.17 37959.168  66786.45  50901.8  89266.20 39143.95
## PrizeMoney   17516.00 86782.00 44080.000  26129.00 120927.0  33471.00 60073.00
## CV residual -18856.34 66511.83  6120.832 -40657.45  70025.2 -55795.20 20929.05
##                  49        51        52        54        55       68        73
## Predicted   28368.99 124045.01  89767.86  73877.49  54514.57 47287.72  91567.54
## cvpred      16771.78  88075.36  50264.01  37618.58  67433.15 20863.38  67658.73
## PrizeMoney  65174.00 132327.00 119444.00  13865.00  26301.00 39356.00 103594.00
## CV residual 48402.22  44251.64  69179.99 -23753.58 -41132.15 18492.62  35935.27
##                  74        76        83       84        89        92        94
## Predicted   25233.38 -1111.995 102104.62 23411.49 78791.248  56536.35  47980.86
## cvpred      16250.34 22261.983  73063.81  7611.11 63395.707  11280.83  38487.14
## PrizeMoney  57216.00 25804.000  27361.00 55014.00 54513.000 100398.00  27673.00
## CV residual 40965.66  3542.017 -45702.81 47402.89 -8882.707  89117.17 -10814.14
##                    95       99       112       113       114       126
## Predicted   14588.403 41256.70  50145.04 10010.135 29970.959  42018.73
## cvpred      26003.892 24342.99  40085.10 -5157.676 27184.216  44101.69
## PrizeMoney  29296.000 53530.00  18494.00 12110.000 18721.000  18838.00
## CV residual  3292.108 29187.01 -21591.10 17267.676 -8463.216 -25263.69
##                   128      131       134      135      136       137       142
## Predicted    26371.41 9025.244 26322.135 95701.25 27660.63  1888.618 213682.03
## cvpred       33288.34 4091.210 31665.454 66380.80 24254.27  3828.593  93691.68
## PrizeMoney    5285.00 8272.000 26532.000 89312.00 37869.00 11376.000 224027.00
## CV residual -28003.34 4180.790 -5133.454 22931.20 13614.73  7547.407 130335.32
##                   147       148       150      155       156      157      159
## Predicted    17066.11 10074.310  33920.67 50432.93 36641.597  69202.8 78605.69
## cvpred       40088.50 20462.979  42485.94 31366.38 27522.616  79827.9 46749.22
## PrizeMoney   14558.00 16455.000 111028.00 51005.00 36428.000  32843.0 69173.00
## CV residual -25530.50 -4007.979  68542.06 19638.62  8905.384 -46984.9 22423.78
##                   167       168       169       173       174      176      178
## Predicted    48632.30  22695.09  55995.55  71920.72  44488.28 16147.02 500420.6
## cvpred       52777.23  40649.24  62679.95  64873.25  44736.30 23523.87 233567.4
## PrizeMoney   27657.00  15012.00  42958.00 105997.00 150889.00 36861.00 662771.0
## CV residual -25120.23 -25637.24 -19721.95  41123.75 106152.70 13337.13 429203.6
##                   194
## Predicted   37253.055
## cvpred      28890.178
## PrizeMoney  30344.000
## CV residual  1453.822
##
## Sum of squares = 270559263230    Mean square = 5521617617    n = 49
##
## fold 4
## Observations in test set: 49
##                     5         17        20        21        27        28
## Predicted    23649.79   9891.795  25627.34  127742.1  23441.69  114911.8
## cvpred       50360.75  24035.508  54566.43  214215.0  38902.04  190915.3
## PrizeMoney   16683.00  11989.000  19683.00   79316.0  20322.00   37751.0
## CV residual -33677.75 -12046.508 -34883.43 -134899.0 -18580.04 -153164.3
##                    30        32        34        42        43        50
## Predicted   49505.842  78973.02  30997.38  20577.71  29456.83  31824.54
## cvpred      92901.535 131366.05  64260.31  41109.56  60568.11  59987.01
## PrizeMoney  94571.000 112443.00  37735.00  14499.00  31371.00  15187.00
## CV residual  1669.465 -18923.05 -26525.31 -26610.56 -29197.11 -44800.01
##                    53       56        59        67        70        71
## Predicted    67272.16  38644.9  100344.4  71520.10  56133.95  53488.45
## cvpred      124782.36  69875.3  179345.6 130541.51  94118.69  93841.22
## PrizeMoney   73819.00  22340.0   57092.0  30656.00   2240.00  38188.00
## CV residual -50963.36 -47535.3 -122253.6 -99885.51 -91878.69 -55653.22
##                    75        78        80        82        85        97
## Predicted    75347.36  44672.51 10932.552  47557.92  49093.49  10015.89
## cvpred      115026.57  79424.35 26322.898  79939.85  91474.86  23061.97
## PrizeMoney   82196.00   7583.00 24724.000  16927.00  20612.00   2692.00
## CV residual -32830.57 -71841.35 -1598.898 -63012.85 -70862.86 -20369.97
##                    98       100       102        103       106       116
## Predicted    33208.48  140149.4  203318.9   994.5391  116500.6  82171.18
## cvpred       66298.99  223471.0  343551.5 -1164.1277  199127.2 143314.29
## PrizeMoney   15964.00   58953.0   70421.0 18085.0000   58189.0  83483.00
## CV residual -50334.99 -164518.0 -273130.5 19249.1277 -140938.2 -59831.29
##                   118       120       123       124        130       138
## Predicted    50524.37  49885.76  31995.19  35501.80   93913.42  9275.927
## cvpred       93619.85  81590.50  55155.44  59062.78  171478.84 27955.728
## PrizeMoney   20188.00  26123.00  41390.00  22467.00   56693.00 23403.000
## CV residual -73431.85 -55467.50 -13765.44 -36595.78 -114785.84 -4552.728
##                    153       160      161       163       170       172
## Predicted    70018.444 25502.000 52457.37  62746.32  37765.32  137418.2
## cvpred      123026.409 45325.373 73473.97 105219.73  64430.14  243253.2
## PrizeMoney  119240.000 47046.000 91808.00  56305.00  11421.00  106577.0
## CV residual  -3786.409  1720.627 18334.03 -48914.73 -53009.14 -136676.2
##                   175       180       182       190       193       195
## Predicted    1612.051 31159.834  6212.218  1739.241 15292.755  43869.35
## cvpred      17648.188 67335.474 19426.545  9059.469 18912.429  84590.00
## PrizeMoney  15098.000 65783.000 11187.000 10354.000 12803.000  38043.00
## CV residual -2550.188 -1552.474 -8239.545  1294.531 -6109.429 -46547.00
##                  196
## Predicted   20344.40
## cvpred      42001.29
## PrizeMoney  90824.00
## CV residual 48822.71
##
## Sum of squares = 286208693787    Mean square = 5840993751    n = 49
##
```

```
## Overall (Sum over all 49 folds)
##         ms
## 3677949892
```

```
#
part_c_df <- thr_df %>%
  mutate(gir_sec = poly(GIR, 2, raw=TRUE),
         bird_conv_sec = poly(BirdieConversion, 2, raw=T),
         sand_saves_sec = poly(SandSaves, 2, raw=T)) %>%
  dplyr::select(c(PrizeMoney, GIR, BirdieConversion, SandSaves, gir_sec, bird_conv_sec,
                  sand_saves_sec,Scrmb_BouncBck, PuttAvg_Scrmb,
                  PuttAvg_BouncBck, PuttAvg_Gir, DrvAcc_GIR, BouncBckSec))


# second order model
model_5 <- lm(PrizeMoney~.,data = part_c_df)

summary(model_5)
```

c. Beginning from scratch, engineer all possible second-order terms and add them to your dataset. From this dataset, produce a model using backward selection. Evaluate this model using 5-fold cross validation. Do you arrive at the same model as above? Explain.

```
##
## Call:
## lm(formula = PrizeMoney ~ ., data = part_c_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125736  -20205   -2208   15527  238778
##
## Coefficients: (3 not defined because of singularities)
##                    Estimate Std. Error t value      Pr(>|t|)
## (Intercept)      6656717.34 1104322.14   6.028 0.00000000896 ***
## GIR              -176664.40   34768.66  -5.081 0.00000092100 ***
## BirdieConversion  -88272.69   26125.62  -3.379      0.000890 ***
## SandSaves          -1567.58    6645.50  -0.236      0.813785
## gir_sec1                 NA         NA      NA            NA
## gir_sec2            1152.99     274.61   4.199 0.00004186107 ***
## bird_conv_sec1           NA         NA      NA            NA
## bird_conv_sec2      1674.61     447.27   3.744      0.000242 ***
## sand_saves_sec1          NA         NA      NA            NA
## sand_saves_sec2       30.37      67.63   0.449      0.653952
## Scrmb_BouncBck      1098.05     325.08   3.378      0.000893 ***
## PuttAvg_Scrmb     -10141.07    3652.70  -2.776      0.006070 **
## PuttAvg_BouncBck  -49555.66   12607.44  -3.931      0.000120 ***
## PuttAvg_Gir        21654.82    7494.30   2.890      0.004325 **
## DrvAcc_GIR           -21.42      11.66  -1.836      0.067956 .
## BouncBckSec          643.00     309.79   2.076      0.039333 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 42510 on 183 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5574
## F-statistic: 21.47 on 12 and 183 DF,  p-value: < 0.00000000000000022
```

Now, let's try backward selection.

```
# backward selection
bckwrd_selctn <- stepAIC(model_5, direction = "backward")
```

```
## Start:  AIC=4190.3
## PrizeMoney ~ GIR + BirdieConversion + SandSaves + gir_sec + bird_conv_sec +
##     sand_saves_sec + Scrmb_BouncBck + PuttAvg_Scrmb + PuttAvg_BouncBck +
##     PuttAvg_Gir + DrvAcc_GIR + BouncBckSec
##
##
## Step:  AIC=4190.3
## PrizeMoney ~ GIR + BirdieConversion + gir_sec + bird_conv_sec +
##     sand_saves_sec + Scrmb_BouncBck + PuttAvg_Scrmb + PuttAvg_BouncBck +
##     PuttAvg_Gir + DrvAcc_GIR + BouncBckSec
##
##
## Step:  AIC=4190.3
## PrizeMoney ~ GIR + gir_sec + bird_conv_sec + sand_saves_sec +
##     Scrmb_BouncBck + PuttAvg_Scrmb + PuttAvg_BouncBck + PuttAvg_Gir +
##     DrvAcc_GIR + BouncBckSec
##
##
## Step:  AIC=4190.3
## PrizeMoney ~ gir_sec + bird_conv_sec + sand_saves_sec + Scrmb_BouncBck +
##     PuttAvg_Scrmb + PuttAvg_BouncBck + PuttAvg_Gir + DrvAcc_GIR +
##     BouncBckSec
##
##                    Df   Sum of Sq          RSS    AIC
## <none>                              330726784077 4190.3
## - sand_saves_sec    2  9087838897 339814622974 4191.6
## - DrvAcc_GIR        1  6093120367 336819904444 4191.9
## - BouncBckSec       1  7785553418 338512337495 4192.9
## - PuttAvg_Scrmb     1 13930246485 344657030562 4196.4
## - PuttAvg_Gir       1 15089169972 345815954049 4197.0
## - Scrmb_BouncBck    1 20619348233 351346132310 4200.2
## - PuttAvg_BouncBck  1 27922307163 358649091240 4204.2
## - bird_conv_sec     2 44350444173 375077228250 4211.0
## - gir_sec           2 49812186522 380538970599 4213.8
```

```
print(bckwrd_selctn$anova)
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## PrizeMoney ~ GIR + BirdieConversion + SandSaves + gir_sec + bird_conv_sec +
##     sand_saves_sec + Scrmb_BouncBck + PuttAvg_Scrmb + PuttAvg_BouncBck +
##     PuttAvg_Gir + DrvAcc_GIR + BouncBckSec
##
## Final Model:
## PrizeMoney ~ gir_sec + bird_conv_sec + sand_saves_sec + Scrmb_BouncBck +
##     PuttAvg_Scrmb + PuttAvg_BouncBck + PuttAvg_Gir + DrvAcc_GIR +
##     BouncBckSec
##
##
##                  Step Df     Deviance Resid. Df   Resid. Dev      AIC
## 1                                           183 330726784077 4190.303
## 2        - SandSaves  0 0.0000000000       183 330726784077 4190.303
## 3 - BirdieConversion  0 0.0001220703       183 330726784077 4190.303
## 4              - GIR  0 0.0000000000       183 330726784077 4190.303
```

Let's see how the final model is performing.

```
model_final <- lm(PrizeMoney ~ gir_sec + bird_conv_sec + sand_saves_sec + Scrmb_BouncBck +
    PuttAvg_Scrmb + PuttAvg_BouncBck + PuttAvg_Gir + DrvAcc_GIR +
    BouncBckSec, data = part_c_df)

summary(model_final)
```

```
##
## Call:
## lm(formula = PrizeMoney ~ gir_sec + bird_conv_sec + sand_saves_sec +
##     Scrmb_BouncBck + PuttAvg_Scrmb + PuttAvg_BouncBck + PuttAvg_Gir +
##     DrvAcc_GIR + BouncBckSec, data = part_c_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -125736  -20205   -2208   15527  238778
##
## Coefficients:
##                    Estimate Std. Error t value      Pr(>|t|)
## (Intercept)      6656717.34 1104322.14   6.028 0.00000000896 ***
## gir_sec1         -176664.40   34768.66  -5.081 0.00000092100 ***
## gir_sec2            1152.99     274.61   4.199 0.00004186107 ***
## bird_conv_sec1    -88272.69   26125.62  -3.379      0.000890 ***
## bird_conv_sec2      1674.61     447.27   3.744      0.000242 ***
## sand_saves_sec1    -1567.58    6645.50  -0.236      0.813785
## sand_saves_sec2       30.37      67.63   0.449      0.653952
## Scrmb_BouncBck      1098.05     325.08   3.378      0.000893 ***
## PuttAvg_Scrmb     -10141.07    3652.70  -2.776      0.006070 **
## PuttAvg_BouncBck  -49555.66   12607.44  -3.931      0.000120 ***
## PuttAvg_Gir        21654.82    7494.30   2.890      0.004325 **
## DrvAcc_GIR           -21.42      11.66  -1.836      0.067956 .
## BouncBckSec          643.00     309.79   2.076      0.039333 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 42510 on 183 degrees of freedom
## Multiple R-squared:  0.5847, Adjusted R-squared:  0.5574
## F-statistic: 21.47 on 12 and 183 DF,  p-value: < 0.0000000000000022
```

From comparison, it is clear that we've arrived the model from earlier. This is because of the data used to build the model. We've used the data which yields the best model possible with features available to us.

As the same data has been used to perform stepwise model selection, it will not be able to achieve local maxima or minima of the metric. Yet, it will follow a particular path by adding or removing the variables from the iteration.

d. You have used two procedures to build a second-order model. Compare these two procedures. Which do you think is "best"? Explain. In the first method, we first identify the features that are significant by building a full model with all the features. Then, gradually, we remove the features which are not significant. This is a iterative process where we remove features one-by-one to get the best version of the model. However, in the second method, we build a model using stepwise selection where we pass in a full model object and select the direction for stepwise search. In this method, we build a multiple versions of model based on its AIC (prediction error, similar to adj R-squared) and features.

Building a model using backward or forward selection method gives you more flexibility in terms of manual efforts. This way, we can build multiple versions of model and pick one of our choice which is accurate and less complex. Also, it tries all the combination of model from null model (model with no features) to full model (with all features) by defining the scope. Therefore, stepwise selection is best for building a model.