NHL Game and Season Prediction Models

Mark Dodd, Raymond Wong, Dustin Tang, Michael Ellsworth

2019-12-03

Introduction

The research topic of this project is an analysis of National Hockey League (NHL) statistics. Our team's objective is to determine which statistics influence the outcomes of NHL games and if these statistics can help predict an NHL team's final points total. Additionally, we want to investigate if they can be used to predict the outcome of an individual game.

Specifically, the question that we have set out to answer in this report are: Can build a model to predict the outcome of a hockey game that is better than flipping a coin?

As hockey fans, this topic is of interest for a couple of reasons. First, it is common for hockey fans to make predictions based on what they have seen in previous games or how the teams look in the standings, but often these predictions are subjective and qualitative. This group is interested to see if there is a more quantitative approach to these predictions. Secondly, we wanted to investigate if a quantitative approach to predictions could be used for sports betting. The group is interested to see if a model can be created that consistently beats the odds set by the bookmakers.

The data used for this project is taken from three key sources:

- Hockey-reference.com
- Naturalstattrick.com
- NHL Game Data sourced from Kaggle

These sites provide a combination of both the traditional hockey statistics such as goals, assists, penalty minutes etc. but also provide the advanced non-traditional hockey statistics that incorporate location information to determine shot quality. We wanted to see if, based on our techniques and methodologies from our DATA 603 course, if a model would consider both traditional hockey statistics and advanced hockey statistics.

Permission restrictions for use of data from hockey-reference.com can be found at the following link: https://www.sports-reference.com/data_use.html (https://www.sports-reference.com/data_use.html). For the purposes of this project, there are no restrictions for the dataset that we are using. Additionally, we have communicated with naturalstattrick.com to ensure use of their data is acceptable for this project. A transcript of this communication can be provided upon request. As far as the NHL Game Data from Kaggle, this is a public dataset and is not restricted for the purposes of this project.

To accomplish our tasks for this project we used a combination of R and Python. Various packages in R's extensive statistics library will be leveraged for model building, whereas Python was used for data wrangling and data frame preparation for analysis in R.

Methodology

Descriptive Statistics

Each descriptive statistics considered in both the logistic and multiple linear regression models are described below. All variables are numeric values and are either a whole number or represented as a percentage.

Descriptive Statistics - Logistic Regression

Hockey Stat Abbrev.	Hockey Stat. Name	Description
Corsi	Corsi	Any shot attempt (goals, shots on net, misses and blocks) outside of the shootout
CF	Corsi For	Count of Corsi for that team.
CA	Corsi Againist	Count of Corsi against that team.
CF%	CF%	Percentage of total Corsi in games that team played that are for that team. CF*100/(CF+CA)

Hockey Stat Abbrev.	Hockey Stat. Name	Description
Fenwick	Fenwick	Any unblocked shot attempt (goals, shots on net and misses) outside of the shootout
FF	Fenwick For	Count of Fenwick for that team
FA	Fenwick Againist	Count of Fenwick against that team.
FF%	Fenwick Pencentage	Percentage of total Fenwick in games that team played that are for that team. FF*100/(FF+FA)
Shots	Any shot attempt on net (goals and shots on net) outside of the shootout	NA
SF	Shots For	Count of Shots for that team
SA	Shots Againist	Count of Shots against that team.
Goals	Any goal, outside of the shootout.	Any goal, outside of the shootout.
GF	Goals For	Count of Goals for that team.
GA	Goals Againist	Count of Goals against that team
GF%	Goals For %	Percentage of total Goals in games that team played that are for that team. GF*100/(GF+GA)
SC	Scoring Chances	Each shot attempt (Corsi) taken in the offensive zone is assigned a value based on the area of the zone in which it was recorded
SCF	Scoring Chances For	Count of Scoring Chances for that team.
SCA	Scoring Chances againist	Count of Scoring Chances against that team
SCF%	Scoring Chances %	Percentage of total Scoring Chances in games that team played that are for that team. SCF*100/(SCF+SCA)
SH%	Shots %	Percentage of Shots for that team that were Goals. GF*100/SF
SV%	Save %	Percentage of Shots against that team that were not Goals. 100-(GA*100/SA)
PDO	Not defined	Shooting percentage plus save percentage. (GF/SF)+(GA/SA)
Hits	Hits	Count of hits for a team in a game
PIM	Penalty Minutes	The number of penalty minutes a team gets in a game
PowerPlayOpportunities	PowerPlayOpportunities	The number of times a team goes on the Power Play
powerPlayGoals	powerPlayGoals	The number of goals a team scores in a game
faceOffWinPercentage	faceOffWinPercentage	Percent of the times a team wins a faceoff
giveaways	giveaways	The number of times a team gives away the puck
takeaways	takeaways	The number of times a team take away the puck from the other team
z	Zone	For the next variables, z is a prefix for zone. There are 3 types of zones. Zone can be HighDanger (HD), Medium Danger (MD) or Low Danger (LD)
zCF	“zone” Chances For	Count of zone Scoring Chances for that team.

Hockey Stat Abbrev.	Hockey Stat. Name	Description
zCA	“zone” Chances Againist	Count of zone Scoring Chances against that team.
zCF%	“zone” Chances For Percent	Percentage of total “Zone” Scoring Chances in games that team played that are for that team. HDCF*100/(HDCF+HDCA)
zSF	“zone” Shots For	Count of Shots that are in the “Zone” Scoring Chances for that team.
zSA	“zone” Shots Againist	Count of Shots that are in the “Zone” Scoring Chances against that team.
zSF%	“zone” Shots for percent	Percentage of total Shots that are in the “Zone” Scoring Chances in games that team played that are for that team. zSF*100/(zSF+zSA)
zGF	“zone” Goals For	Count of Goals off of a “Zone” Scoring Chances for that team.
zGA	“zone” Goals Againist	Count of Goals off of a “Zone” Scoring Chances against that team.
zGF%	“zone” Goals For Percentage	Percentage of total Goals off of a “zone” Scoring Chances in games that team played that are for that team. zGF*100/(zGF+zGA)
zSH%	“zone” Shots Percentage	Percentage of total Shots for a “zone” Scoring Chances in games that team played that are for that team. zGF*100/(zGF+zGA)
zSV%	“zone” Save Percentage	Percentage of total Shots for a “zone” Shots for that team that were Goals. zGF*100/zSF

Descriptive Statistics - Multiple Linear Regression

Hockey Stat Abbrev.	Hockey Stat. Name	Description	Relation
AvAge	Average Age	Average Age of the hockey team	Lower age means a youger team
W	Wins	Number of Wins for a season	More wins in a season better the team
L (Losses)	Losses	Number of Losses for a season	More losses in a season the worse the team
OL (Overtime Losses)	Overtime Losses	Number of Overtime Losses for a season	More Overtime Losses in a season the worse the team but these losses aren’t as bad as regular losses
PTS (Points)	Points per season	The number of points for a team for a season	More points in a season better the team
GF (Goals For Team A)	Goals for Team A	Number of goals a team scores for a season	More goals in a season better the team
GA (Goals Against Team A	Goals Againist Team A	Number of goals a team lets in for a season	More goals againist in a season better the team
GD (GF - GA)	Goal Differential	The difference of goals for a team. The number of goals for minus the number of goals againist	The higher the difference the better the team
SOW	Shootout wins	Number of wins for a team in the shootout	More SOW means the team is better at the Shootout

Hockey Stat Abbrev.	Hockey Stat. Name	Description	Relation
SOL	Shootout Losses	Number of loses for a team in the shootout	More SOL means the team is worse at the Shootout
SRS	Simple Rating System	Rating for Goal Differential and Strength of Schedule	Higher SRS means better the team
SOS	Strength of Scehdule	Measurement of the opponent in the standings	Lower sos means the opponent are not good
TG/G	Total goals = (GF+GA) per game	Total goals = (GF+GA) per game	The higher TG/G means the more goals per game
EVGF	Even Strength Goals For	Even Strength Goals For a team per season	Higher EVGF means the team is better at even strength
EVGA	Even Strength Goals Against	Even Strength Goals Against per season	Lower EVGA means the team is better at even strength
PD (Penalty Differential)	Penalty Minutes For - Penalty Minutes againist per game in a season	The difference of Penalty for - Penalty againist per game in a season	The higher Penalty Differential means the team takes more penalties
SD (Shot Differential)	Shots For - Shots againist per game in a season	Shots For - Shots againist per game in a season	The higher Shots Differential means the team takes more shots than they give up
PDO	Shooting % + Save % at Even Strength	Shooting % + Save % at Even Strength	Higher pdo means the better the team at Even Strength
CF%	Corsi For % at 5 on 5	Percentage of any shot attempt (goals, shots on net, misses and blocked shots)	The higher cf% means the team is shotting more than their opponents.
FF% - Percentage of any shot attempt (goals, shots on net, misses)	Fenwick For % at 5 on 5 FF / (FF + FA)	Above 50% means the team was controlling the puck more often than not with this player on the ice in this situation. This doesn't count blocked shots	The higher FF% means the team is shooting more than their opponents.
xGF	Expected Goals For	Expected Goals For' given where shots came from, for and against, while this player was on the ice at even strength. It's based on where the shots are coming from, compared to the league-wide shooting percentage for that shot location.	The higher xGF the more goals a team is expected to score for a game
xGA	Expected Goals Against	Expected Goals Against' given where shots came from, for and against, while this player was on the ice at even strength. It's based on where the shots are coming from, compared to the league-wide shooting percentage for that shot location.	The lower xGA the more goals a team is expected to let in for a game
SCF%	Percentage Scoring Chances for	Percentage of scoring chances in this team's favor	Higher scf% means a team has more scoring chances

Hockey Stat Abbrev.	Hockey Stat. Name	Description	Relation
HDF%	High Danger Scoring Chances For Percentage	Percentage of high-danger scoring chances in this team's favor	Higher HDF% means a team has more scoring chances

Data Wrangling

The data that we have used for this project provides game by game statistics, as defined by the above table, for the 2014-2018 seasons (five seasons in total). From these years we wanted to try and use the 2014-2017 data to make predictions for the 2018 season and with this goal in mind the 2018 data was excluded from our model building. This left us with three seasons with 1230 games per season and one season with 1271 games for a total of 4,961 games of data that we could use to build our model.

Our goal was to build a logistic regression model to predict future games using previous data. Game by game data does not provide any value for our purposes, but it does provide insight into how a team plays if the data is looked at in aggregate. With this in mind we averaged all previous game data for each individual statistic for each season. The seasons were separated as teams tend to make drastic roster changes in the offseason so the statistics were only averaged for each individual season. For example, for the 42nd game that the Calgary Flames played in the 2017 season we would use an average of games 1-41 that the Flames played during the 2017 season.

Once we had the average statistics for the previous games for each team we then subtracted the Away Team statistics from the Home Team statistics (Home minus Away) to create marginal statistics for each game and we used these marginal statistics as inputs to our logistic regression model. For example, if the Home Team average shots per game was 31.3 prior to game i and the Away Team average shots per game was 29.4 prior to game i, then our model would see an average marginal shot difference of 1.9 and it is these numbers that we used in our model.

All data wrangling was performed in Python and the results were exported to csv files to analyze in R.

Methodology - Logistic Regression

In order to create a prediction model for NHL games, the group utilized the logistic regression methodology described in DATA 603. The first step in this process was to evaluate the variables in the previously described datasets by running a full model, check for multicollinearity and run individual z-tests on all variables to test variable significance.

Assumptions that will be tested in the multiple logistic regression model include:

- Multicollinearity

Methodology - Multiple Linear Regression

In order to create a prediction model for a team's next season's points totals, the group utilized the multiple linear regression methodology described in DATA 603. The individual coefficients test (t-test) is a partial model test that will be used to test the significance of the statistics from the previously described datasets. Additionally, a stepwise regression procedure will be used to compare the partial model tests to see if a more effective model can be built.

Assumptions that will be tested in the multiple linear regression model include:

- Multicollinearity
- Linearity
- Equal variance
- Normality
- Outliers

Main Results of the Analysis

Results - Logistic Regression

We calculate for Variance Inflation Factor (VIF) to confirm that our variables are not collinear. We also use ggpair to visually check if there is any type of multicollinearity.

From our data wrangling and compiling our datasets, we start out with 75 variables! 75!

Code, Findings and Visualizations - Logistic Regression

```
nhl.na = na.omit(nhl.data)
names(nhl.na)
```

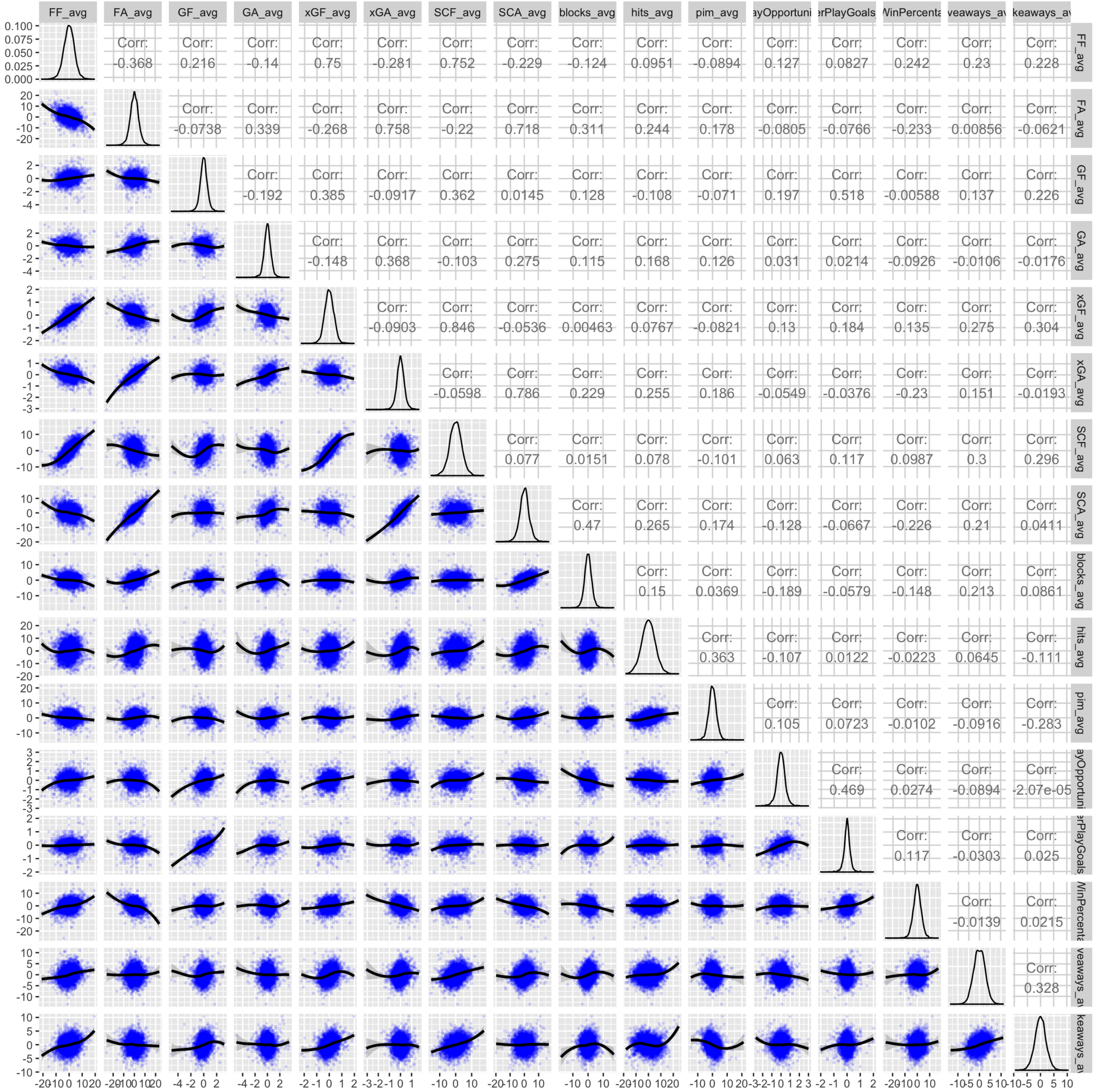
```
##  [1] "X1"                         "game_id"
##  [3] "date"                       "home"
##  [5] "home_goals"                 "away"
##  [7] "away_goals"                 "hoa"
##  [9] "result"                     "result_bool"
## [11] "Game"                       "Team"
## [13] "TOI"                        "CF_avg"
## [15] "CA_avg"                     "CF%_avg"
## [17] "FF_avg"                     "FA_avg"
## [19] "FF%_avg"                    "SF_avg"
## [21] "SA_avg"                     "SF%_avg"
## [23] "GF_avg"                     "GA_avg"
## [25] "GF%_avg"                    "xGF_avg"
## [27] "xGA_avg"                    "xGF%_avg"
## [29] "SCF_avg"                    "SCA_avg"
## [31] "SCF%_avg"                   "HDCF_avg"
## [33] "HDCA_avg"                   "HDCF%_avg"
## [35] "HDSF_avg"                   "HDSA_avg"
## [37] "HDSF%_avg"                  "HDGF_avg"
## [39] "HDGA_avg"                   "HDGF%_avg"
## [41] "HDSH%_avg"                  "HDSV%_avg"
## [43] "MDCF_avg"                   "MDCA_avg"
## [45] "MDCF%_avg"                  "MDSF_avg"
## [47] "MDSA_avg"                   "MDSF%_avg"
## [49] "MDGF_avg"                   "MDGA_avg"
## [51] "MDGF%_avg"                  "MDSH%_avg"
## [53] "MDSV%_avg"                  "LDCF_avg"
## [55] "LDCA_avg"                   "LDCF%_avg"
## [57] "LDSF_avg"                   "LDSA_avg"
## [59] "LDSF%_avg"                  "LDGF_avg"
## [61] "LDGA_avg"                   "LDGF%_avg"
## [63] "LDSH%_avg"                  "LDSV%_avg"
## [65] "SH%_avg"                    "SV%_avg"
## [67] "PDO_avg"                    "blocks_avg"
## [69] "goals_avg"                  "shots_avg"
## [71] "hits_avg"                   "pim_avg"
## [73] "powerPlayOpportunities_avg" "powerPlayGoals_avg"
## [75] "faceOffWinPercentage_avg"   "giveaways_avg"
## [77] "takeaways_avg"
```

```
nhl.stats = subset(nhl.na, select = CF_avg:takeaways_avg)

nhl.stats = nhl.stats %>% mutate(result_bool = nhl.na$result_bool)
nhl.reduced = nhl.na %>% select(-c(CF_avg:`CF%_avg`, `FF%_avg`:`SF%_avg`, `GF%_avg`, `xGF%_avg`, `SCF%_avg`, HDC
F_avg:`SV%_avg`, PDO_avg, shots_avg, goals_avg))
imcdiag(nhl.stats%>% select(-c(result_bool)), as.numeric(nhl.stats$result_bool), method="VIF")
```

```
##
## Call:
## imcdiag(x = nhl.stats %>% select(-c(result_bool)), y = as.numeric(nhl.stats$result_bool),
##     method = "VIF")
##
##
##  VIF Multicollinearity Diagnostics
##
##                                    VIF detection
## CF_avg                        334.0870         1
## CA_avg                             Inf         1
## CF%_avg                       720.8852         1
## FF_avg                        307.5446         1
## FA_avg                             Inf         1
## FF%_avg                       864.1731         1
## SF_avg                        673.3095         1
## SA_avg                        257.0018         1
## SF%_avg                       442.7898         1
## GF_avg                        370.8073         1
## GA_avg                        129.6370         1
## GF%_avg                        12.9360         1
## xGF_avg                        57.3259         1
## xGA_avg                        50.0463         1
## xGF%_avg                      133.4764         1
## SCF_avg                            Inf         1
## SCA_avg                            Inf         1
## SCF%_avg                      130.8243         1
## HDCF_avg                           Inf         1
## HDCA_avg                           Inf         1
## HDCF%_avg                      84.7504         1
## HDSF_avg                       71.0278         1
## HDSA_avg                       55.0845         1
## HDSF%_avg                      52.8002         1
## HDGF_avg                       46.3906         1
## HDGA_avg                       45.8362         1
## HDGF%_avg                       7.5802         0
## HDSH%_avg                       7.6312         0
## HDSV%_avg                       8.6963         0
## MDCF_avg                           Inf         1
## MDCA_avg                           Inf         1
## MDCF%_avg                     135.9731         1
## MDSF_avg                       48.0604         1
## MDSA_avg                       35.0233         1
## MDSF%_avg                      39.2388         1
## MDGF_avg                       29.1327         1
## MDGA_avg                       27.7779         1
## MDGF%_avg                       7.5209         0
## MDSH%_avg                       9.7202         0
## MDSV%_avg                       8.7675         0
## LDCF_avg                      110.1810         1
## LDCA_avg                      103.3410         1
## LDCF%_avg                     182.2968         1
## LDSF_avg                       59.9849         1
## LDSA_avg                       75.5658         1
## LDSF%_avg                      66.1091         1
## LDGF_avg                       22.7655         1
## LDGA_avg                       29.1783         1
## LDGF%_avg                      11.2325         1
## LDSH%_avg                      14.4888         1
## LDSV%_avg                      15.0828         1
## SH%_avg                     56065.1337         1
## SV%_avg                     49254.8411         1
## PDO_avg                    122287.7386         1
## blocks_avg                         Inf         1
## goals_avg                     252.7855         1
## shots_avg                     496.7065         1
## hits_avg                        1.7558         0
## pim_avg                          1.5844         0
## powerPlayOpportunities_avg       1.6405         0
## powerPlayGoals_avg               2.2401         0
## faceOffWinPercentage_avg         1.4877         0
## giveaways_avg                    1.8786         0
## takeaways_avg                    1.6381         0
##
## Multicollinearity may be due to CF_avg CA_avg CF%_avg FF_avg FA_avg FF%_avg SF_avg SA_avg SF%_avg GF_avg GA_a
vg GF%_avg xGF_avg xGA_avg xGF%_avg SCF_avg SCA_avg SCF%_avg HDCF_avg HDCA_avg HDCF%_avg HDSF_avg HDSA_avg HDSF%
_avg HDGF_avg HDGA_avg MDCF_avg MDCA_avg MDCF%_avg MDSF_avg MDSA_avg MDSF%_avg MDGF_avg MDGA_avg LDCF_avg LDCA_a
vg LDCF%_avg LDSF_avg LDSA_avg LDSF%_avg LDGF_avg LDGA_avg LDGF%_avg LDSH%_avg LDSV%_avg SH%_avg SV%_avg PDO_avg
blocks_avg goals_avg shots_avg regressors
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## ===================================
```

```
ggpairs(data = nhl.reduced %>% select (-c(X1:TOI)),
        lower = list(continuous = wrap("smooth_loess", alpha = 0.1, size = 0.5, color = 'blue'),
                     combo ="facethist",
                     discrete = "facetbar",
                     na = "na"))
```



After looking at the results VIF we realized we can remove a lot of the variables as a number of them returned inf and extremely high values.

In an effort to further focus our model into variables of significance, we, as Hockey Data Experts, recognize that the HD (High Danger Shots), MD (Medium Danger Shots), LD (Low Danger Shots) are all based and related to one stat, SCF (Scoring Chances For % Average). So going forward, our model does not consider those variables, and is further reduced to 16 initial variables.

After cleaning up the variables we are left with these variables:

- FF_avg
- FA_avg
- GF_avg
- GA_avg
- xGF_avg
- xGA_avg
- SCF_avg
- SCA_avg
- blocks_avg

- hits_avg
- pim_avg
- powerPlayOpportunities_avg
- powerPlayGoals_avg
- faceOffWinPercentage_avg
- giveaways_avg
- takeaways_avg

We re-test multicollinearity for these variables using the VIF to ensure that there is no presence of multicollinearity.

```
nhl.reduced = nhl.na %>% select(-c(CF_avg:`CF%_avg`, `FF%_avg`:`SF%_avg`, `GF%_avg`, `xGF%_avg`, `SCF%_avg`, HDC
F_avg:PDO_avg, shots_avg))
names(nhl.reduced)
```

```
##  [1] "X1"                         "game_id"
##  [3] "date"                       "home"
##  [5] "home_goals"                 "away"
##  [7] "away_goals"                 "hoa"
##  [9] "result"                     "result_bool"
## [11] "Game"                       "Team"
## [13] "TOI"                        "FF_avg"
## [15] "FA_avg"                     "GF_avg"
## [17] "GA_avg"                     "xGF_avg"
## [19] "xGA_avg"                    "SCF_avg"
## [21] "SCA_avg"                    "blocks_avg"
## [23] "goals_avg"                  "hits_avg"
## [25] "pim_avg"                    "powerPlayOpportunities_avg"
## [27] "powerPlayGoals_avg"         "faceOffWinPercentage_avg"
## [29] "giveaways_avg"              "takeaways_avg"
```

```
imcdiag(nhl.reduced %>% select(c(FF_avg:takeaways_avg))%>% select(-c(goals_avg)), as.numeric(nhl.reduced$result_
bool), method="VIF")
```

```
##
## Call:
## imcdiag(x = nhl.reduced %>% select(c(FF_avg:takeaways_avg)) %>%
##     select(-c(goals_avg)), y = as.numeric(nhl.reduced$result_bool),
##     method = "VIF")
##
##
##  VIF Multicollinearity Diagnostics
##
##                               VIF detection
## FF_avg                     3.4428         0
## FA_avg                     3.1664         0
## GF_avg                     1.9088         0
## GA_avg                     1.2709         0
## xGF_avg                    4.5837         0
## xGA_avg                    4.0952         0
## SCF_avg                    5.1676         0
## SCA_avg                    4.5142         0
## blocks_avg                 1.5436         0
## hits_avg                   1.3803         0
## pim_avg                    1.3069         0
## powerPlayOpportunities_avg 1.4261         0
## powerPlayGoals_avg         1.8717         0
## faceOffWinPercentage_avg   1.1493         0
## giveaways_avg              1.3181         0
## takeaways_avg              1.3129         0
##
## NOTE:  VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## ===================================
```

At this point all values sit within low or moderate levels of multicollinearity. Now we focus on building the model. This is our first step of building our model.

```
nhl.mdl.1 = glm(result_bool ~ FF_avg +
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                hits_avg +
                                pim_avg +
                                powerPlayOpportunities_avg +
                                powerPlayGoals_avg +
                                faceOffWinPercentage_avg +
                                giveaways_avg + takeaways_avg
              , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.1)
```

```
##
## Call:
## glm(formula = result_bool ~ FF_avg + FA_avg + GF_avg + GA_avg +
##     xGF_avg + xGA_avg + SCF_avg + SCA_avg + blocks_avg + hits_avg +
##     pim_avg + powerPlayOpportunities_avg + powerPlayGoals_avg +
##     faceOffWinPercentage_avg + giveaways_avg + takeaways_avg,
##     family = "binomial", data = nhl.reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9474  -1.2174   0.9005   1.0880   1.8252
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.203171   0.029382   6.915 4.68e-12 ***
## FF_avg                     -0.002835   0.013130  -0.216 0.829022
## FA_avg                     -0.037318   0.012641  -2.952 0.003155 **
## GF_avg                      0.169986   0.068860   2.469 0.013565 *
## GA_avg                     -0.233495   0.056250  -4.151 3.31e-05 ***
## xGF_avg                    -0.380995   0.164989  -2.309 0.020932 *
## xGA_avg                     0.395389   0.166630   2.373 0.017651 *
## SCF_avg                     0.075895   0.017805   4.262 2.02e-05 ***
## SCA_avg                    -0.043680   0.018309  -2.386 0.017046 *
## blocks_avg                  0.024435   0.015116   1.617 0.105983
## hits_avg                   -0.004379   0.006339  -0.691 0.489695
## pim_avg                     0.042039   0.012484   3.367 0.000759 ***
## powerPlayOpportunities_avg -0.037577   0.072807  -0.516 0.605775
## powerPlayGoals_avg          0.173845   0.152136   1.143 0.253166
## faceOffWinPercentage_avg   -0.004055   0.009520  -0.426 0.670175
## giveaways_avg               0.004641   0.012721   0.365 0.715269
## takeaways_avg               0.013056   0.015902   0.821 0.411629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6515.3  on 4813  degrees of freedom
## AIC: 6549.3
##
## Number of Fisher Scoring iterations: 4
```

From here we decide to run individual Z-test on each variable, while considering the Residual Deviance, AIC and AUC values to further reduce our model and consider significant variables. Within our fully additive model, we decide to remove any variable where p-value > 0.05 and perform a LRTest (Likelihood Ratio Test) to see if it is safe to remove the variable from the model. This will be repeated until all of the variables are significant.

As an example, we review individual Z-test from our fully additive model, we can see that our p-value for the $\beta_{FF_{avg}}$ variable > 0.829022 . Using this parameter we run the LRTest (Likelihood Ratio Test) between the two models. This will confirm whether or not a reduced model of 15 variables is more significant than our full model with 16 variables.

$$H_0 : \beta_{FF_{avg}} = 0 \; or \; Reduced \; Model \; is \; True$$
$$H_A : \beta_{FF_{avg}} \neq 0 \; or \; Full \; Model \; is \; True$$

```
nhl.mdl.1_reduced = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                hits_avg +
                                pim_avg +
                                powerPlayOpportunities_avg +
                                powerPlayGoals_avg +
                                faceOffWinPercentage_avg +
                                giveaways_avg + takeaways_avg
              , data = nhl.reduced, family = 'binomial')

lrtest(nhl.mdl.1_reduced,nhl.mdl.1)
```

```
## Likelihood ratio test
##
## Model 1: result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg + xGA_avg +
##     SCF_avg + SCA_avg + blocks_avg + hits_avg + pim_avg + powerPlayOpportunities_avg +
##     powerPlayGoals_avg + faceOffWinPercentage_avg + giveaways_avg +
##     takeaways_avg
## Model 2: result_bool ~ FF_avg + FA_avg + GF_avg + GA_avg + xGF_avg + xGA_avg +
##     SCF_avg + SCA_avg + blocks_avg + hits_avg + pim_avg + powerPlayOpportunities_avg +
##     powerPlayGoals_avg + faceOffWinPercentage_avg + giveaways_avg +
##     takeaways_avg
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  16 -3257.7
## 2  17 -3257.6  1 0.0466      0.829
```

From the LRTest, our p-value = 0.829 > 0.05, therefore we fail to reject the H_0, we remove the ‘FF_Avg’ variable from our starting additive 16 variable model, and re-test with our now 15 variable model.

We run through the following workflow:

1. Create the model.
2. Run Summary of Additive Model and review individual z-tests of all variables.
3. Review Residual Deviance values and AIC (Akaike information criterion) values to ensure that the significance of our model does not drop.
4. Review p-value of individual z-tests of each variable in our model and remove the most insignificant variable. We continue to use a p-value < 0.05 as our threshold.
5. Perform LR Test to confirm that we are allowed to remove the variable.

Repeat. Steps 1-5.

We continue this until our model contains only significant variables and each variables’ p-value < 0.05. Our process is shown in the following r-chunks.

```
nhl.mdl.2 = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                hits_avg +
                                pim_avg +
                                powerPlayOpportunities_avg +
                                powerPlayGoals_avg +
                                faceOffWinPercentage_avg +
                                giveaways_avg + takeaways_avg
                , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.2)
```

```
##
## Call:
## glm(formula = result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + hits_avg + pim_avg +
##     powerPlayOpportunities_avg + powerPlayGoals_avg + faceOffWinPercentage_avg +
##     giveaways_avg + takeaways_avg, family = "binomial", data = nhl.reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9519  -1.2172   0.9005   1.0874   1.8093
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 0.203115   0.029380   6.913 4.73e-12 ***
## FA_avg                     -0.037510   0.012609  -2.975 0.002931 **
## GF_avg                      0.171235   0.068614   2.496 0.012574 *
## GA_avg                     -0.233957   0.056208  -4.162 3.15e-05 ***
## xGF_avg                    -0.390549   0.158937  -2.457 0.014000 *
## xGA_avg                     0.400150   0.165172   2.423 0.015409 *
## SCF_avg                     0.074329   0.016259   4.572 4.84e-06 ***
## SCA_avg                    -0.043024   0.018054  -2.383 0.017170 *
## blocks_avg                  0.024567   0.015103   1.627 0.103809
## hits_avg                   -0.004607   0.006250  -0.737 0.461022
## pim_avg                     0.042058   0.012483   3.369 0.000754 ***
## powerPlayOpportunities_avg -0.039538   0.072231  -0.547 0.584121
## powerPlayGoals_avg          0.176692   0.151561   1.166 0.243691
## faceOffWinPercentage_avg   -0.004394   0.009390  -0.468 0.639855
## giveaways_avg               0.004334   0.012641   0.343 0.731741
## takeaways_avg               0.013066   0.015901   0.822 0.411261
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6515.3  on 4814  degrees of freedom
## AIC: 6547.3
##
## Number of Fisher Scoring iterations: 4
```

Remove giveaways_avg

```
nhl.mdl.3 = glm(result_bool ~ 
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                hits_avg +
                                pim_avg +
                                powerPlayOpportunities_avg +
                                powerPlayGoals_avg +
                                takeaways_avg
                , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.3)
```

```
## 
## Call:
## glm(formula = result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg + 
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + hits_avg + pim_avg + 
##     powerPlayOpportunities_avg + powerPlayGoals_avg + takeaways_avg, 
##     family = "binomial", data = nhl.reduced)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.9381  -1.2162   0.9018   1.0876   1.8148  
## 
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                 0.203303   0.029371   6.922 4.45e-12 ***
## FA_avg                     -0.038080   0.012480  -3.051 0.002279 ** 
## GF_avg                      0.175903   0.068049   2.585 0.009740 ** 
## GA_avg                     -0.234034   0.056158  -4.167 3.08e-05 ***
## xGF_avg                    -0.394163   0.158476  -2.487 0.012875 *  
## xGA_avg                     0.414728   0.163035   2.544 0.010965 *  
## SCF_avg                     0.074692   0.016223   4.604 4.14e-06 ***
## SCA_avg                    -0.042582   0.018031  -2.362 0.018195 *  
## blocks_avg                  0.025671   0.014940   1.718 0.085741 .  
## hits_avg                   -0.004512   0.006246  -0.722 0.470023    
## pim_avg                     0.041709   0.012469   3.345 0.000823 ***
## powerPlayOpportunities_avg -0.038029   0.071966  -0.528 0.597203    
## powerPlayGoals_avg          0.164170   0.149811   1.096 0.273145    
## takeaways_avg               0.014200   0.015448   0.919 0.357990    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6515.6  on 4816  degrees of freedom
## AIC: 6543.6
## 
## Number of Fisher Scoring iterations: 4
```

Remove powerPlayOpportunities_avg

```
nhl.mdl.4 = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                hits_avg +
                                pim_avg +
                                powerPlayGoals_avg +
                                takeaways_avg
                , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.4)
```

```
##
## Call:
## glm(formula = result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + hits_avg + pim_avg +
##     powerPlayGoals_avg + takeaways_avg, family = "binomial",
##     data = nhl.reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9041  -1.2164   0.9026   1.0878   1.8059
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.203506   0.029367   6.930 4.22e-12 ***
## FA_avg             -0.038409   0.012466  -3.081 0.002062 **
## GF_avg              0.177959   0.067951   2.619 0.008821 **
## GA_avg             -0.235468   0.056098  -4.197 2.70e-05 ***
## xGF_avg            -0.401480   0.157872  -2.543 0.010988 *
## xGA_avg             0.415596   0.163051   2.549 0.010807 *
## SCF_avg             0.074930   0.016218   4.620 3.83e-06 ***
## SCA_avg            -0.042272   0.018021  -2.346 0.018993 *
## blocks_avg          0.026683   0.014815   1.801 0.071697 .
## hits_avg           -0.003969   0.006161  -0.644 0.519457
## pim_avg             0.040793   0.012348   3.304 0.000954 ***
## powerPlayGoals_avg  0.131949   0.136881   0.964 0.335062
## takeaways_avg       0.014115   0.015447   0.914 0.360862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6515.9  on 4817  degrees of freedom
## AIC: 6541.9
##
## Number of Fisher Scoring iterations: 4
```

Remove hits_avg

```
nhl.mdl.5 = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg +
                                powerPlayGoals_avg +
                                takeaways_avg
                , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.5)
```

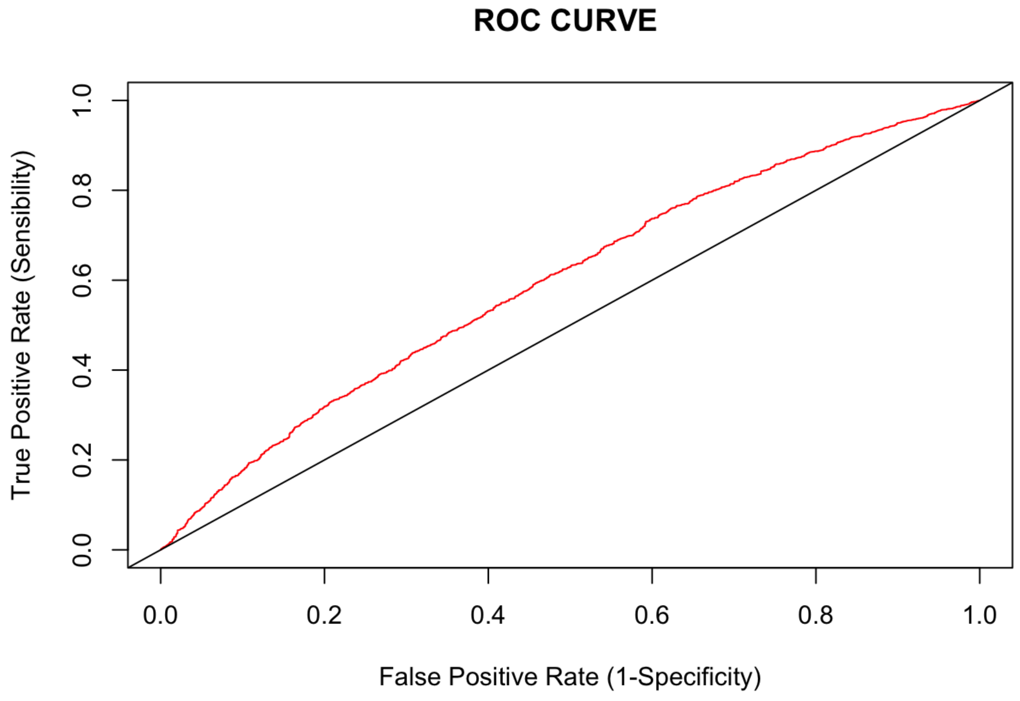
```
##
## Call:
## glm(formula = result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + pim_avg + powerPlayGoals_avg +
##     takeaways_avg, family = "binomial", data = nhl.reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9279  -1.2171   0.9046   1.0892   1.7836
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.20377    0.02936   6.940 3.93e-12 ***
## FA_avg             -0.03918    0.01241  -3.158  0.00159 **
## GF_avg              0.18517    0.06703   2.762  0.00574 **
## GA_avg             -0.23709    0.05604  -4.230 2.33e-05 ***
## xGF_avg            -0.41042    0.15725  -2.610  0.00905 **
## xGA_avg             0.41539    0.16303   2.548  0.01084 *
## SCF_avg             0.07443    0.01620   4.595 4.32e-06 ***
## SCA_avg            -0.04264    0.01801  -2.367  0.01793 *
## blocks_avg          0.02584    0.01476   1.751  0.07993 .
## pim_avg             0.03834    0.01174   3.265  0.00109 **
## powerPlayGoals_avg  0.12642    0.13658   0.926  0.35467
## takeaways_avg       0.01470    0.01542   0.953  0.34034
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6516.3  on 4818  degrees of freedom
## AIC: 6540.3
##
## Number of Fisher Scoring iterations: 4
```

Remove powerPlayGoals_avg and takeaways_avg

```
nhl.mdl.6 = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg
                , data = nhl.reduced, family = 'binomial')
summary(nhl.mdl.6)
```

```
## 
## Call:
## glm(formula = result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg + 
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + pim_avg, family = "binomial", 
##     data = nhl.reduced)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.8910  -1.2193   0.9048   1.0891   1.7951  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept)  0.20330    0.02936   6.926 4.34e-12 ***
## FA_avg      -0.04014    0.01238  -3.242  0.00119 ** 
## GF_avg       0.22410    0.05612   3.993 6.52e-05 ***
## GA_avg      -0.22531    0.05512  -4.087 4.36e-05 ***
## xGF_avg     -0.38381    0.15593  -2.461  0.01384 *  
## xGA_avg      0.42169    0.16282   2.590  0.00960 ** 
## SCF_avg      0.07318    0.01611   4.544 5.53e-06 ***
## SCA_avg     -0.04238    0.01798  -2.357  0.01843 *  
## blocks_avg   0.02484    0.01466   1.695  0.09010 .  
## pim_avg      0.03650    0.01122   3.254  0.00114 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6518.0  on 4820  degrees of freedom
## AIC: 6538
## 
## Number of Fisher Scoring iterations: 4
```

```
prob=predict(nhl.mdl.6,type=c("response"))
pred<-prediction(prob,nhl.reduced$result_bool)
perf<-performance(pred,measure = "tpr",x.measure="fpr")
plot(perf,col=2,main="ROC CURVE ", xlab="False Positive Rate (1-Specificity)",ylab="True Positive Rate (Sensibil
ity)")
abline(0,1)
```



```
roc<-roc(nhl.reduced$result_bool,prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(roc)
```

```
## Area under the curve: 0.5971
```

As a baseline Area under Curve (AUC), AIC, Receiver Operating Characteristic (ROC) Curve and Residual Deviance numbers are:

Final Additive Model:

- Null deviance: 6650.8 on 4829 degrees of freedom
- Residual deviance: 6518.0 on 4820 degrees of freedom
- AIC: 6538
- AUC: 0.5971

We will discuss the significance and meaning of these numbers once we finalize our model (see Conclusion.). As we interact with our variables below, we use these numbers to help determine changes in significance in our model, as we reduce towards our best fit model.

Interesting Scenario - consideration for borderline insignificant variables:

In reducing our model, we did run into a scenario where the p-value for our “blocks_avg” was on the borderline of being insignificant. Ie: P-value = 0.0901. We run on LRTest to verify if we should keep Blocks_avg in our model.

$$H_0 : \beta_{Blocks_{avg}} = 0 \; or \; Reduced \; Model \; is \; True$$
$$H_A : \beta_{Blocks_{avg}} \neq 0 \; or \; Full \; Model \; is \; True$$

```
nhl.mdl.6 = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg
                , data = nhl.reduced, family = 'binomial')

nhl.mdl.blocks = glm(result_bool ~
                                FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                pim_avg
                , data = nhl.reduced, family = 'binomial')

lrtest(nhl.mdl.blocks, nhl.mdl.6)
```

```
## Likelihood ratio test
##
## Model 1: result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg + xGA_avg +
##     SCF_avg + SCA_avg + pim_avg
## Model 2: result_bool ~ FA_avg + GF_avg + GA_avg + xGF_avg + xGA_avg +
##     SCF_avg + SCA_avg + blocks_avg + pim_avg
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   9 -3260.4
## 2  10 -3259.0  1 2.8767    0.08987 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When running LRTest for our blocks_avg variable, our p-value for the $\beta_{blocks_{avg}}$ variable > 0.05 = 0.08987. We fail to reject our H_0, and we should take remove Blocks_avg out of our model.

However, we decide to keep this variable in our model, because our AIC value is larger with the our model with Blocks_avg, than without. As well, we want to consider the variable in our interaction models, to see if our AIC and Residual values decrease. We continue on.

Interactive Model Methods

Now that we have a final additive model, we will begin to interact our variables to determine if a better fit model exists. We will first start by explaining our process in considering interaction within our model.

Interactive process

Interacting nine variables returns a large number of interaction terms to consider, so we will describe our process here, before showing the R-Code and its outputs.

We also will not show all reduced models through our iterative process, but we will return a number of models to show how our model reduced.

1. Create the model.
2. Run Summary of interactive model and review individual z-tests of all variables.
3. Consider Residual Deviance values, AIC, and AUC values to ensure that the significance of our model does not drop, while comparing with our variables that are insignificant. We continue to use a p-value < 0.05 as our threshold.
4. Review p-value of individual z-tests of each interactive variable in our model and remove the most insignificant variable. We continue to use a p-value < 0.05 as our threshold.
5. Repeat Steps 1-4.

Here is our R-code that describes the above process:

```
nhl.int.1 = glm(result_bool ~
                                (FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg)^2
                , data = nhl.reduced, family = 'binomial')

summary(nhl.int.1)
```

```
##
## Call:
## glm(formula = result_bool ~ (FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + pim_avg)^2, family = "binomial",
##     data = nhl.reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1012  -1.2094   0.8875   1.0843   1.9329
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.430e-01  3.821e-02   6.358 2.04e-10 ***
## FA_avg             -4.106e-02  1.259e-02  -3.261 0.001110 **
## GF_avg              2.318e-01  5.738e-02   4.040 5.35e-05 ***
## GA_avg             -2.335e-01  5.712e-02  -4.087 4.37e-05 ***
## xGF_avg            -4.213e-01  1.605e-01  -2.624 0.008682 **
## xGA_avg             4.388e-01  1.665e-01   2.635 0.008413 **
## SCF_avg             7.618e-02  1.649e-02   4.620 3.83e-06 ***
## SCA_avg            -4.642e-02  1.840e-02  -2.523 0.011633 *
## blocks_avg          2.643e-02  1.491e-02   1.772 0.076348 .
## pim_avg             3.842e-02  1.147e-02   3.351 0.000806 ***
## FA_avg:GF_avg       3.153e-02  1.895e-02   1.664 0.096077 .
## FA_avg:GA_avg       5.935e-03  1.795e-02   0.331 0.740907
## FA_avg:xGF_avg     -1.434e-02  4.826e-02  -0.297 0.766314
## FA_avg:xGA_avg     -3.731e-02  3.569e-02  -1.045 0.295830
## FA_avg:SCF_avg     -4.395e-03  4.986e-03  -0.881 0.378108
## FA_avg:SCA_avg      1.280e-03  4.119e-03   0.311 0.755963
## FA_avg:blocks_avg   3.039e-03  5.520e-03   0.551 0.581882
## FA_avg:pim_avg     -2.702e-03  3.945e-03  -0.685 0.493351
## GF_avg:GA_avg       2.087e-02  6.950e-02   0.300 0.763993
## GF_avg:xGF_avg      8.227e-02  1.901e-01   0.433 0.665184
## GF_avg:xGA_avg     -3.499e-01  2.331e-01  -1.501 0.133383
## GF_avg:SCF_avg      7.134e-04  2.269e-02   0.031 0.974918
## GF_avg:SCA_avg      1.008e-02  2.852e-02   0.353 0.723814
## GF_avg:blocks_avg  -1.385e-02  2.048e-02  -0.676 0.498967
## GF_avg:pim_avg     -4.865e-03  1.713e-02  -0.284 0.776415
## GA_avg:xGF_avg      6.945e-02  1.814e-01   0.383 0.701862
## GA_avg:xGA_avg     -9.900e-02  1.865e-01  -0.531 0.595527
## GA_avg:SCF_avg      9.475e-05  2.129e-02   0.004 0.996449
## GA_avg:SCA_avg      2.095e-02  2.482e-02   0.844 0.398610
## GA_avg:blocks_avg   1.492e-02  2.271e-02   0.657 0.511138
## GA_avg:pim_avg      6.530e-03  1.642e-02   0.398 0.690878
## xGF_avg:xGA_avg    -2.784e-01  5.602e-01  -0.497 0.619197
## xGF_avg:SCF_avg    -1.446e-02  1.806e-02  -0.801 0.423081
## xGF_avg:SCA_avg     1.424e-01  6.233e-02   2.284 0.022365 *
## xGF_avg:blocks_avg -7.248e-02  6.448e-02  -1.124 0.260968
## xGF_avg:pim_avg     3.519e-02  4.545e-02   0.774 0.438761
## xGA_avg:SCF_avg     2.034e-02  6.160e-02   0.330 0.741202
## xGA_avg:SCA_avg    -3.293e-03  3.893e-02  -0.085 0.932601
## xGA_avg:blocks_avg  4.899e-02  5.812e-02   0.843 0.399305
## xGA_avg:pim_avg    -3.989e-02  4.961e-02  -0.804 0.421369
## SCF_avg:SCA_avg    -1.093e-02  6.533e-03  -1.673 0.094299 .
## SCF_avg:blocks_avg  1.111e-02  6.879e-03   1.615 0.106384
## SCF_avg:pim_avg    -3.094e-03  5.179e-03  -0.597 0.550295
## SCA_avg:blocks_avg -6.538e-03  6.011e-03  -1.088 0.276735
## SCA_avg:pim_avg     7.561e-03  5.797e-03   1.304 0.192159
## blocks_avg:pim_avg -8.427e-03  4.880e-03  -1.727 0.084199 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6486.1  on 4784  degrees of freedom
## AIC: 6578.1
```

```
##
## Number of Fisher Scoring iterations: 4
```

```
nhl.int.2 = glm(result_bool ~
                                (FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg) +
                                FA_avg:GF_avg +
                                xGF_avg:SCA_avg +
                                SCF_avg:SCA_avg +
                                blocks_avg:pim_avg

                , data = nhl.reduced, family = 'binomial')
summary(nhl.int.2)
```

```
##
## Call:
## glm(formula = result_bool ~ (FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + pim_avg) + FA_avg:GF_avg +
##     xGF_avg:SCA_avg + SCF_avg:SCA_avg + blocks_avg:pim_avg, family = "binomial",
##     data = nhl.reduced)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.829  -1.215   0.901   1.088   2.056
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.221660   0.030049   7.377 1.62e-13 ***
## FA_avg             -0.040677   0.012400  -3.280  0.00104 **
## GF_avg              0.226985   0.056255   4.035 5.46e-05 ***
## GA_avg             -0.226986   0.055352  -4.101 4.12e-05 ***
## xGF_avg            -0.397726   0.156805  -2.536  0.01120 *
## xGA_avg             0.427893   0.162918   2.626  0.00863 **
## SCF_avg             0.074394   0.016207   4.590 4.43e-06 ***
## SCA_avg            -0.043608   0.018013  -2.421  0.01548 *
## blocks_avg          0.025999   0.014710   1.767  0.07716 .
## pim_avg             0.035937   0.011273   3.188  0.00143 **
## FA_avg:GF_avg       0.003247   0.010187   0.319  0.74994
## xGF_avg:SCA_avg     0.095493   0.033381   2.861  0.00423 **
## SCF_avg:SCA_avg    -0.009696   0.003633  -2.669  0.00762 **
## blocks_avg:pim_avg -0.004348   0.003857  -1.127  0.25963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6507.7  on 4816  degrees of freedom
## AIC: 6535.7
##
## Number of Fisher Scoring iterations: 4
```

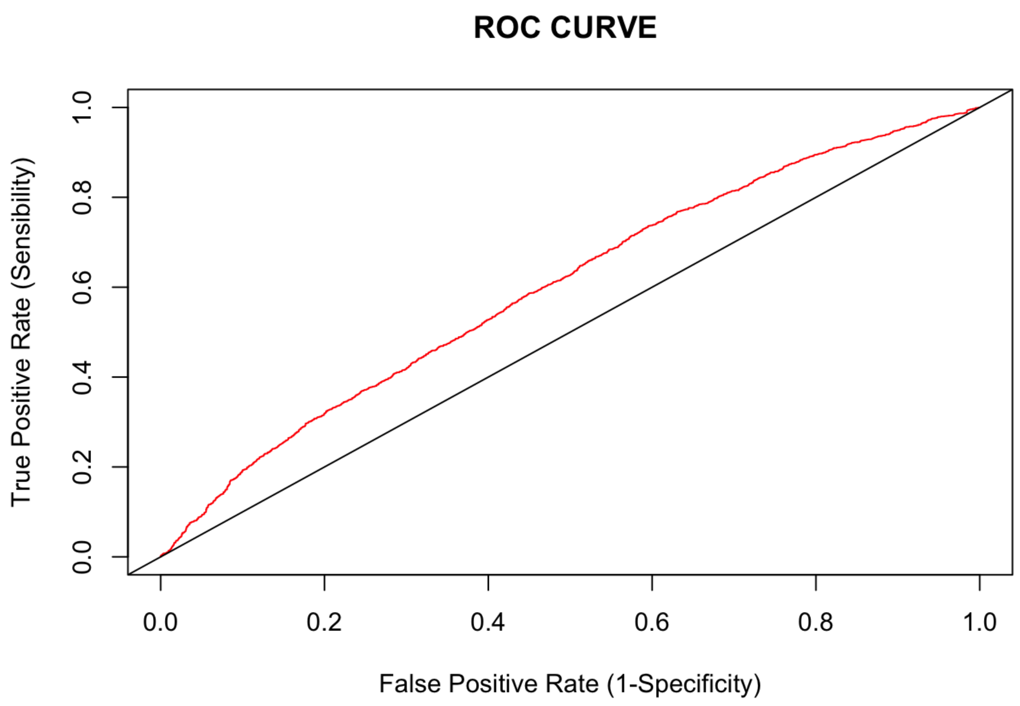
Remove FA_avg:GF_avg and blocks_avg:pim_avg

```
nhl.int.3 = glm(result_bool ~
                                (FA_avg +
                                GF_avg +
                                GA_avg +
                                xGF_avg +
                                xGA_avg +
                                SCF_avg +
                                SCA_avg +
                                blocks_avg +
                                pim_avg) +
                                xGF_avg:SCA_avg +
                                SCF_avg:SCA_avg

                , data = nhl.reduced, family = 'binomial')
summary(nhl.int.3)
```

```
##
## Call:
## glm(formula = result_bool ~ (FA_avg + GF_avg + GA_avg + xGF_avg +
##     xGA_avg + SCF_avg + SCA_avg + blocks_avg + pim_avg) + xGF_avg:SCA_avg +
##     SCF_avg:SCA_avg, family = "binomial", data = nhl.reduced)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.851  -1.216   0.902   1.090   1.964
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.220075   0.030006   7.334 2.23e-13 ***
## FA_avg          -0.040508   0.012393  -3.268  0.00108 **
## GF_avg           0.225015   0.056152   4.007 6.14e-05 ***
## GA_avg          -0.224650   0.055266  -4.065 4.81e-05 ***
## xGF_avg         -0.398329   0.156752  -2.541  0.01105 *
## xGA_avg          0.424882   0.162918   2.608  0.00911 **
## SCF_avg          0.074618   0.016202   4.605 4.12e-06 ***
## SCA_avg         -0.043757   0.018011  -2.429  0.01512 *
## blocks_avg       0.025562   0.014669   1.743  0.08141 .
## pim_avg          0.036574   0.011241   3.254  0.00114 **
## xGF_avg:SCA_avg  0.097357   0.032996   2.951  0.00317 **
## SCF_avg:SCA_avg -0.009528   0.003629  -2.626  0.00865 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6650.8  on 4829  degrees of freedom
## Residual deviance: 6509.1  on 4818  degrees of freedom
## AIC: 6533.1
##
## Number of Fisher Scoring iterations: 4
```

```
prob=predict(nhl.int.3,type=c("response"))
pred<-prediction(prob,nhl.reduced$result_bool)
perf<-performance(pred,measure = "tpr",x.measure="fpr")
plot(perf,col=2,main="ROC CURVE ", xlab="False Positive Rate (1-Specificity)",ylab="True Positive Rate (Sensibil
ity)")
abline(0,1)
```



```
roc<-roc(nhl.reduced$result_bool,prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(roc)
```

```
## Area under the curve: 0.5993
```

$$\begin{aligned}\widehat{logit}_{win} &= \beta_0 + \beta_1 X_{FA_{avg}} + \beta_2 X_{GF_{avg}} + \beta_3 X_{GA_{avg}} + \beta_4 X_{xGF_{avg}} + \beta_5 X_{xGA_{avg}} + \beta_6 X_{SCF_{avg}} + \beta_7 X_{SCA_{avg}} + \beta_8 X_{Blocks_{avg}} \\ &+ \beta_9 X_{PIM_{avg}} + \beta_{10} X_{xGF_{avg}} * X_{SCA_{avg}} + \beta_{11} X_{SCF_{avg}} * X_{SCA_{avg}}\end{aligned}$$

$$\begin{aligned}\widehat{logit}_{win} &= 0.22 - 0.04 X_{FA_{avg}} + 0.23 X_{GF_{avg}} - 0.22 X_{GA_{avg}} - 0.40 X_{xGF_{avg}} + 0.42 X_{xGA_{avg}} + 0.07 X_{SCF_{avg}} - 0.04 X_{SCA_{avg}} \\ &+ 0.03 X_{Blocks_{avg}} + 0.04 X_{PIM_{avg}} + 0.10 X_{xGF_{avg}} * X_{SCA_{avg}} - 0.01 X_{SCF_{avg}} * X_{SCA_{avg}}\end{aligned}$$

Our Logistic Regression Model is below:

$$\widehat{\pi}_{win} = \frac{e^{\beta_0+\beta_1 X_{FAavg}+\beta_2 X_{GFavg}+\beta_3 X_{GAavg}+\beta_4 X_{xGFavg}+\beta_5 X_{xGAavg}+\beta_6 X_{SCFavg}+\beta_7 X_{SCAavg}+\beta_8 X_{Blocksavg}+\beta_9 X_{PIMavg}+\beta_{10} X_{xGFavg} * X_{SCAavg}+\beta_{11} X_{SCFavg} * X_{SCAavg}}}{1+e^{\beta_0+\beta_1 X_{FAavg}+\beta_2 X_{GFavg}+\beta_3 X_{GAavg}+\beta_4 X_{xGFavg}+\beta_5 X_{xGAavg}+\beta_6 X_{SCFavg}+\beta_7 X_{SCAavg}+\beta_8 X_{Blocksavg}+\beta_9 X_{PIMavg}+\beta_{10} X_{xGFavg} * X_{SCAavg}+\beta_{11} X_{SCFavg} * X_{SCAavg}}}$$

$$\widehat{\pi}_{win} = \frac{e^{0.22-0.04 X_{FAavg}+0.23 X_{GFavg}-0.22 X_{GAavg}-0.40 X_{xGFavg}+0.42 X_{xGAavg}+0.07 X_{SCFavg}-0.04 X_{SCAavg}+0.03 X_{Blocksavg}+0.04 X_{PIMavg}+0.10 X_{xGFavg} * X_{SCAavg}-0.01 X_{SCFavg} * X_{SCAavg}}}{1+e^{0.22-0.04 X_{FAavg}+0.23 X_{GFavg}-0.22 X_{GAavg}-0.40 X_{xGFavg}+0.42 X_{xGAavg}+0.07 X_{SCFavg}-0.04 X_{SCAavg}+0.03 X_{Blocksavg}+0.04 X_{PIMavg}+0.10 X_{xGFavg} * X_{SCAavg}-0.01 X_{SCFavg} * X_{SCAavg}}}$$

Results - Multiple Linear Regression

In the early development of the logistic regression model described previously, it appeared as though a suitable model for predicting outcomes of games would not be achievable. To ensure that the group would have a regression model that would be worth using, a multiple regression model was created for predicting a team's next seasons points total. The results of this regression model building is described below.

Code, Findings and Visualizations - Multiple Linear Regression

Model with subset of variables

The first step in determining the most suitable model for predicting team's next seasons points was to take a subset of the available variables and build an additive model. Since there are a significant amount of season statistics to choose from, a number of variables were removed based on assumed significance. There was no statistical reasoning for removing the variables at this point. Instead, we assumed the statistics that were being removed were irrelevant. For example, statistics such as “Average Team Age”, were removed as we didn't believe they would significantly impact the model.

The statistics that will be used in the first full additive model are:

- PTS
- GF
- GD
- SD
- CF%
- FF%
- SCF%
- HDF%
- PDO
- PD

The first full additive model will be built with the `lm()` function using the seasons dataset from hockey-reference.com.

```
season_points_model_full <- lm(data = seasons,
                               FUTURE_PTS ~ PTS + GD + SD + `CF%` + `FF%` + `SCF%` + `HDF%` + PDO + PD)
summary(season_points_model_full)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ PTS + GD + SD + `CF%` + `FF%` + `SCF%` +
##     `HDF%` + PDO + PD, data = seasons)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.653  -8.139   1.379   6.245  24.111
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -290.19723  431.43266  -0.673   0.5042
## PTS           -0.25010    0.49086  -0.510   0.6126
## GD             0.14359    0.21981   0.653   0.5165
## SD            -0.04382    0.02620  -1.672   0.1006
## `CF%`          4.83247    2.67928   1.804   0.0772 .
## `FF%`          0.75904    3.66987   0.207   0.8370
## `SCF%`         0.57118    1.59963   0.357   0.7225
## `HDF%`         0.88400    0.79028   1.119   0.2686
## PDO            0.51670    4.11510   0.126   0.9006
## PD            -2.66338    2.53037  -1.053   0.2975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.98 on 51 degrees of freedom
## Multiple R-squared:  0.4106, Adjusted R-squared:  0.3065
## F-statistic: 3.947 on 9 and 51 DF,  p-value: 0.0007292
```

Based on the individual coefficients test (t-test) using the `summary()` function, only one variable appears to be significant and will be used in a reduced model:

- CF%

A reduced linear regression model will be built with the `lm()` function.

```
season_points_model_reduced <- lm(data = seasons, FUTURE_PTS ~ `CF%`)
summary(season_points_model_reduced)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ `CF%`, data = seasons)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.135  -8.723   1.830   6.794  31.200
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -94.4087    37.3880  -2.525   0.0143 *
## `CF%`         3.7056     0.7472   4.959 6.31e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.19 on 59 degrees of freedom
## Multiple R-squared:  0.2942, Adjusted R-squared:  0.2823
## F-statistic:  24.6 on 1 and 59 DF,  p-value: 6.314e-06
```

Based on the reduced linear regression model summary, a team's future seasons points can be predicted with the CF% variable with a model that can explain 29.4% of the variation in a team's future seasons points. This is not a significant amount of explained variation and hence, further model fitting will be attempted.

Using the same variables in the first full additive model, a stepwise regression procedure will be used to determine if a better model for a team's future seasons points can be fitted. From the `olsrr` package, the `ols_step_both_p()` function will be used to complete the stepwise regression procedure.

```
model_stepwise1_team_pts = ols_step_both_p(season_points_model_full, pent = 0.1, prem = 0.3)
```

```
## Stepwise Selection Method
## ---------------------------
## 
## Candidate Terms: 
## 
## 1. PTS 
## 2. GD 
## 3. SD 
## 4. `CF%` 
## 5. `FF%` 
## 6. `SCF%` 
## 7. `HDF%` 
## 8. PDO 
## 9. PD 
## 
## We are selecting variables based on p value...
## 
## Variables Entered/Removed: 
## 
## - `SCF%` added 
## 
## No more variables to be added/removed.
## 
## 
## Final Model Output 
## ------------------
## 
##                           Model Summary                           
## ---------------------------------------------------------------
## R                       0.548       RMSE                12.138 
## R-Squared               0.301       Coef. Var           13.360 
## Adj. R-Squared          0.289       MSE                147.320 
## Pred R-Squared          0.260       MAE                  9.491 
## ---------------------------------------------------------------
##  RMSE: Root Mean Square Error 
##  MSE: Mean Square Error 
##  MAE: Mean Absolute Error 
## 
##                                 ANOVA                                  
## ---------------------------------------------------------------------
##                 Sum of                                               
##                Squares        DF    Mean Square      F         Sig. 
## ---------------------------------------------------------------------
## Regression     3735.821         1       3735.821    25.359    0.0000 
## Residual       8691.851        59        147.320                     
## Total         12427.672        60                                    
## ---------------------------------------------------------------------
## 
##                                     Parameter Estimates                                      
## -------------------------------------------------------------------------------------------
##       model       Beta    Std. Error    Std. Beta      t        Sig        lower     upper 
## -------------------------------------------------------------------------------------------
## (Intercept)    -68.072        31.598                 -2.154    0.035    -131.299    -4.846 
##      `SCF%`      3.179         0.631        0.548     5.036    0.000       1.916     4.442 
## -------------------------------------------------------------------------------------------
```

Based on the stepwise regression procedure using a p-value entry limit of 0.1 and a p-value exit threshold of 0.3, SCF% is the only variable that was included in the model. The R-squared value increases slightly as the model using SCF% instead of CF% can explain 30.1% of the variation in a team’s future seasons points. Although the model has improved slightly, there is still plenty of room for improvement.

The next step will be to determine if more variables can be accepted using the stepwise regression procedure using a higher p-value entry and exit limit; 0.2 and 0.3 respectively.

```
season_points_stepwise1b = ols_step_both_p(season_points_model_full, pent = 0.2, prem = 0.3)
```

```
## Stepwise Selection Method
## ---------------------------
## 
## Candidate Terms: 
## 
## 1. PTS 
## 2. GD 
## 3. SD 
## 4. `CF%` 
## 5. `FF%` 
## 6. `SCF%` 
## 7. `HDF%` 
## 8. PDO 
## 9. PD 
## 
## We are selecting variables based on p value...
## 
## Variables Entered/Removed: 
## 
## - `SCF%` added 
## - `CF%` added 
## - SD added 
## 
## No more variables to be added/removed.
## 
## 
## Final Model Output 
## ------------------
## 
##                           Model Summary                           
## ---------------------------------------------------------------
## R                       0.604       RMSE                11.769 
## R-Squared               0.365       Coef. Var           12.954 
## Adj. R-Squared          0.331       MSE                138.509 
## Pred R-Squared          0.278       MAE                  8.824 
## ---------------------------------------------------------------
##  RMSE: Root Mean Square Error 
##  MSE: Mean Square Error 
##  MAE: Mean Absolute Error 
## 
##                                  ANOVA                                   
## ---------------------------------------------------------------------
##                   Sum of                                             
##                  Squares        DF    Mean Square      F         Sig. 
## ---------------------------------------------------------------------
## Regression     4532.671         3       1510.890    10.908    0.0000 
## Residual       7895.001        57        138.509                     
## Total         12427.672        60                                    
## ---------------------------------------------------------------------
## 
##                                       Parameter Estimates                                        
## ---------------------------------------------------------------------------------------------
##       model        Beta    Std. Error    Std. Beta      t        Sig        lower      upper 
## ---------------------------------------------------------------------------------------------
## (Intercept)    -233.636        80.690                 -2.895    0.005    -395.214    -72.058 
##      `SCF%`       2.386         1.154        0.412     2.067    0.043       0.075      4.697 
##       `CF%`       4.104         1.747        0.601     2.349    0.022       0.606      7.603 
##          SD      -0.031         0.016       -0.453    -1.911    0.061      -0.064      0.001 
## ---------------------------------------------------------------------------------------------
```

From this stepwise regression procedure, three variables are included in the model:

- SCF%
- CF%
- SD

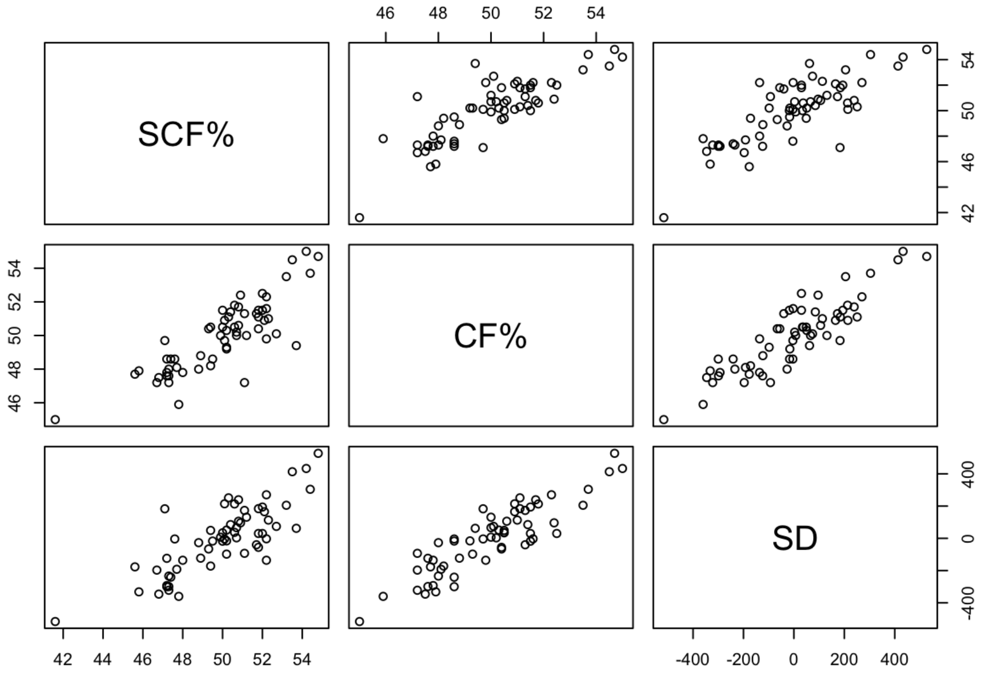
This appears to be a better model than the previous stepwise model as the adjusted R-squared value suggests that this model can explain 33.1% of the variance in a team's future seasons points. The previous best model could only explain 30.1%.

To ensure that the variables in the stepwise regression model are independent, multicollinearity will be tested using the `vif()` function from the `car` package and the `pairs()` function. The `vif()` function will calculate the variance inflation factor (VIF) and the `pairs()` function will identify multicollinearity visually.

```
vif(season_points_stepwise1b$model)
```

```
##   `SCF%`    `CF%`       SD
## 3.555168 5.868643 5.050860
```

```
pairs(~`SCF%` + `CF%` + SD, data = seasons)
```



Based on the values of VIF, SD and CF% appear to have critical levels of multicollinearity, and one of these variables should be removed from the model. This is confirmed with the pairs plots as CF% and SD appear to be linearly correlated.

Since CF% is more significant in the parameter estimates from the stepwise output, SD will be dropped and a new model will be built.

```
reduced_stepwise1b <- lm(data = seasons, FUTURE_PTS ~ `CF%` + `SCF%`)
summary(reduced_stepwise1b)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ `CF%` + `SCF%`, data = seasons)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.298  -7.833   1.032   5.262  30.053
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -95.747     36.913  -2.594    0.012 *
## `CF%`          1.907      1.345   1.418    0.162
## `SCF%`         1.826      1.142   1.599    0.115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.03 on 58 degrees of freedom
## Multiple R-squared:  0.324,  Adjusted R-squared:  0.3007
## F-statistic:  13.9 on 2 and 58 DF,  p-value: 1.169e-05
```

Based on the adjusted R-squared value from the reduced stepwise model, this model is not any better than the model with just SCF%. However, with two variables available, an interaction term can be tested.

```
reduced_interaction <- lm(data = seasons, FUTURE_PTS ~ (`CF%` + `SCF%`)**2)
summary(reduced_interaction)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ (`CF%` + `SCF%`)^2, data = seasons)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.300  -7.875   1.139   5.208  30.057
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.154e+02  6.013e+02  -0.192    0.849
## `CF%`         2.317e+00  1.261e+01   0.184    0.855
## `SCF%`        2.202e+00  1.157e+01   0.190    0.850
## `CF%`:`SCF%` -7.875e-03  2.407e-01  -0.033    0.974
##
## Residual standard error: 12.14 on 57 degrees of freedom
## Multiple R-squared:  0.324,  Adjusted R-squared:  0.2885
## F-statistic: 9.108 on 3 and 57 DF,  p-value: 5.106e-05
```

From the summary output of the model with the interaction term, none of the terms are significant according to the individual coefficients test which means the interaction model will not be used. Since the additive model with CF% and SCF% did not perform any better than the model with SCF%, the model with SCF% is the best model at this point.

```
## (Intercept)      `SCF%`
##  -68.072314    3.179121
```

```
## [1] 0.3006051
```

$$\widehat{Points}_{future} = -68.1 + 3.2 * SCF\%$$

Based on this model, with every one percent increase in SCF%, a teams next season points total will increase by ~3.2.

Model with all available variables

Since the best model found with a subset of variables had an adjusted R-squared value of only 0.301, we will attempt to find a better model utilizing all of the available variables.

The statistics that will be used in the second full additive model are:

- AvAge
- W
- L
- OL
- PTS
- GF
- GA
- GD
- SOW
- SOL
- SRS
- SOS
- TG/G
- EVGF
- EVGA
- PD
- SD
- PDO
- CF%
- FF%
- xGF
- xGA
- SCF%
- HDF%

The second full additive model will be built with the `lm()` function using the seasons dataset from hockey-reference.com. The individual coefficients test will be used to test the significance of each of the variables.

```
model_full2_team_pts <- lm(data = seasons,
                          FUTURE_PTS ~ AvAge + W + L + OL + PTS + GF + GA + GD + SOW + SOL + SRS +
                            SOS + `TG/G` + EVGF + EVGA + PD + SD + PDO + `CF%` + `FF%` + xGF + xGA +
                            `SCF%` + `HDF%`)
summary(model_full2_team_pts)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ AvAge + W + L + OL + PTS + GF + GA +
##     GD + SOW + SOL + SRS + SOS + `TG/G` + EVGF + EVGA + PD +
##     SD + PDO + `CF%` + `FF%` + xGF + xGA + `SCF%` + `HDF%`, data = seasons)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.5838  -6.1337  -0.4094   4.5808  23.5324
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.670e+02  6.680e+02  -0.549   0.5859
## AvAge       -1.996e+00  1.634e+00  -1.221   0.2294
## W            2.787e-03  1.025e+00   0.003   0.9978
## L           -5.126e-02  8.401e-01  -0.061   0.9517
## OL                  NA         NA      NA       NA
## PTS                 NA         NA      NA       NA
## GF          -1.654e+01  8.952e+00  -1.848   0.0722 .
## GA          -1.892e+01  7.967e+00  -2.375   0.0225 *
## GD                  NA         NA      NA       NA
## SOW          1.397e+00  5.075e+00   0.275   0.7845
## SOL         -9.760e-01  5.086e+00  -0.192   0.8488
## SRS         -1.093e+02  4.120e+02  -0.265   0.7922
## SOS          9.628e+01  4.004e+02   0.240   0.8112
## `TG/G`       1.485e+03  5.624e+02   2.640   0.0119 *
## EVGF        -3.256e-02  3.470e-01  -0.094   0.9257
## EVGA        -4.542e-01  3.604e-01  -1.260   0.2150
## PD          -2.468e+00  2.506e+00  -0.985   0.3306
## SD          -1.780e-02  2.884e-02  -0.617   0.5406
## PDO          1.274e+00  6.200e+00   0.206   0.8382
## `CF%`        3.069e+00  2.644e+00   1.161   0.2528
## `FF%`        5.428e-01  3.764e+00   0.144   0.8861
## xGF         -5.293e-01  3.192e-01  -1.658   0.1053
## xGA          1.514e-01  3.249e-01   0.466   0.6438
## `SCF%`       2.899e+00  2.107e+00   1.376   0.1768
## `HDF%`       7.836e-01  8.731e-01   0.897   0.3750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.7 on 39 degrees of freedom
## Multiple R-squared:  0.6406, Adjusted R-squared:  0.4471
## F-statistic:  3.31 on 21 and 39 DF,  p-value: 0.0006019
```

From the individual coefficients test, the following variables appear to be significant:

- GF
- GA
- TG/G
- xGF

From here, the stepwise regression procedure will be used to test all the available variables to see if the significant terms from the individual coefficients test are consistent with the stepwise procedure. A p-value entry limit of 0.05 and p-value exit limit of 0.3 will be used for this stepwise model.

```
model_stepwise2_team_pts <- ols_step_both_p(model_full2_team_pts, pent = 0.05, prem = 0.3)
```

```
## Stepwise Selection Method
## ---------------------------
## 
## Candidate Terms: 
## 
## 1. AvAge 
## 2. W 
## 3. L 
## 4. OL 
## 5. PTS 
## 6. GF 
## 7. GA 
## 8. GD 
## 9. SOW 
## 10. SOL 
## 11. SRS 
## 12. SOS 
## 13. `TG/G` 
## 14. EVGF 
## 15. EVGA 
## 16. PD 
## 17. SD 
## 18. PDO 
## 19. `CF%` 
## 20. `FF%` 
## 21. xGF 
## 22. xGA 
## 23. `SCF%` 
## 24. `HDF%` 
## 
## We are selecting variables based on p value...
## 
## Variables Entered/Removed: 
## 
## - `SCF%` added 
## - `TG/G` added 
## - xGF added 
## - `CF%` added 
## 
## No more variables to be added/removed.
## 
## 
## Final Model Output 
## ------------------
## 
##                           Model Summary                            
## ---------------------------------------------------------------
## R                       0.674       RMSE                11.010 
## R-Squared               0.454       Coef. Var           12.118 
## Adj. R-Squared          0.415       MSE                121.216 
## Pred R-Squared          0.355       MAE                  8.657 
## ---------------------------------------------------------------
##  RMSE: Root Mean Square Error 
##  MSE: Mean Square Error 
##  MAE: Mean Absolute Error 
## 
##                                  ANOVA                                  
## ---------------------------------------------------------------------
##                  Sum of                                                
##                 Squares        DF    Mean Square      F         Sig. 

## ---------------------------------------------------------------------
## Regression     5639.585         4       1409.896    11.631    0.0000 
## Residual       6788.087        56        121.216                       
## Total         12427.672        60                                      
## ---------------------------------------------------------------------
## 
##                                       Parameter Estimates
## ------------------------------------------------------------------------------------------
##       model        Beta    Std. Error    Std. Beta       t        Sig        lower        upper
## ------------------------------------------------------------------------------------------
## (Intercept)    -208.858        46.299                  -4.511    0.000    -301.606    -116.109
##      `SCF%`       2.209         1.203        0.381      1.836    0.072      -0.201       4.620
##      `TG/G`      18.174         4.994        0.462      3.639    0.001       8.171      28.178
##         xGF      -0.340         0.148       -0.346     -2.301    0.025      -0.635      -0.044
##       `CF%`       2.888         1.292        0.423      2.235    0.029       0.299       5.476
## ------------------------------------------------------------------------------------------
```

From the stepwise regression procedure, the following variables appear to be significant:

- SCF%
- TG/G
- xGF
- CF%

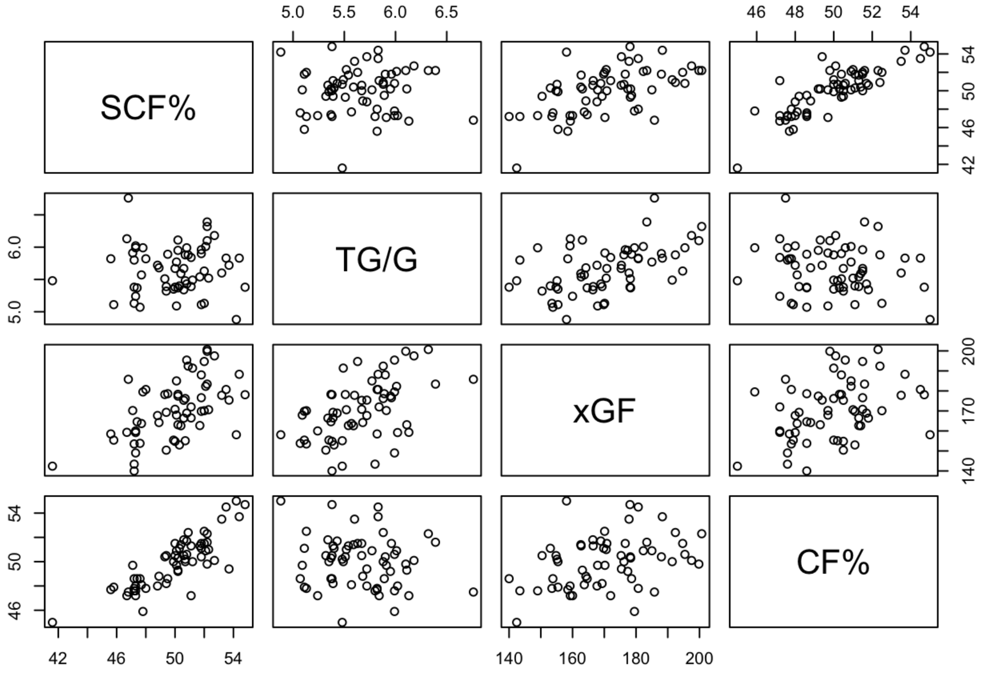
Although the variables are slightly different between the individual coefficients test and the stepwise regression procedure, the stepwise regression model will be used going forward. One of the reasons for choosing the stepwise regression model is the fact that SCF%, which is a variable that was found significant in the previous model, is found in the stepwise regression model but not the model from the individual t-test.

To ensure that the variables in the stepwise regression model are independent, multicollinearity will be tested using the `vif()` function from the `car` package and the `pairs()` function. The `vif()` function will calculate the variance inflation factor (VIF) and the `pairs()` function will identify multicollinearity visually.

```
vif(model_stepwise2_team_pts$model)
```

```
##   `SCF%`   `TG/G`      xGF    `CF%`
## 4.415788 1.655211 2.315742 3.668395
```

```
pairs(~`SCF%` + `TG/G` + xGF + `CF%`, data = seasons)
```



Based on the values of VIF, SCF% and CF% appear to have moderate levels of multicollinearity, bordering on significant. From the pairs visualization, SCF% and CF% appears to be linearly correlated.

Since CF% is more significant in the parameter estimates from the stepwise output, but SCF% was the first variable chosen in the stepwise regression procedure, both variables will be dropped independently and two models will be tested.

```
model_stepwise2_reduced_team_pts_a <- lm(data = seasons, FUTURE_PTS ~ `TG/G` + xGF + `CF%`)
# Adjusted R-squared of model without SCF%
summary(model_stepwise2_reduced_team_pts_a)$adj.r.squared
```

```
## [1] 0.39044
```

```
model_stepwise2_reduced_team_pts_b <- lm(data = seasons, FUTURE_PTS ~ `TG/G` + xGF + `SCF%`)
# Adjusted R-squared of model without CF%
summary(model_stepwise2_reduced_team_pts_b)$adj.r.squared
```

```
## [1] 0.3737718
```

Since the adjusted R squared value is higher in the model without SCF%, this will be the model that is used going forward. As there has been a change to the model, the individual coefficients test will be used to ensure all variables continue to be significant.

```
model_stepwise2_reduced_team_pts <- lm(data = seasons, FUTURE_PTS ~ `TG/G` + xGF + `CF%`)
summary(model_stepwise2_reduced_team_pts)
```

```
## 
## Call:
## lm(formula = FUTURE_PTS ~ `TG/G` + xGF + `CF%`, data = seasons)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -24.4354  -6.8007  -0.1749   7.0765  22.1030 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -208.4509    47.2515  -4.412 4.61e-05 ***
## `TG/G`        17.6714     5.0889   3.473 0.000991 ***
## xGF           -0.2219     0.1356  -1.636 0.107384    
## `CF%`          4.7437     0.8215   5.775 3.36e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 11.24 on 57 degrees of freedom
## Multiple R-squared:  0.4209, Adjusted R-squared:  0.3904 
## F-statistic: 13.81 on 3 and 57 DF,  p-value: 7.007e-07
```

Since the p-value of xGF is higher than 0.05, it will be removed from the reduced model.

```
model_stepwise2_reduced_team_pts <- lm(data = seasons, FUTURE_PTS ~ `TG/G` + `CF%`)
summary(model_stepwise2_reduced_team_pts)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ `TG/G` + `CF%`, data = seasons)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.3165  -7.2437  -0.1601   5.9414  21.6286
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -181.6257    44.9504  -4.041 0.000159 ***
## `TG/G`        12.5383     4.0638   3.085 0.003113 **
## `CF%`          4.0309     0.7064   5.707 4.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.4 on 58 degrees of freedom
## Multiple R-squared:  0.3937, Adjusted R-squared:  0.3728
## F-statistic: 18.83 on 2 and 58 DF,  p-value: 4.981e-07
```

Two variables remain in the best fit model which means we have the opportunity to test an interaction term.

```
model_stepwise2_interact_team_pts <- lm(data = seasons, FUTURE_PTS ~ (`TG/G` + `CF%`)**2)
summary(model_stepwise2_interact_team_pts)
```

```
##
## Call:
## lm(formula = FUTURE_PTS ~ (`TG/G` + `CF%`)^2, data = seasons)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.186  -7.299  -0.161   5.958  21.500
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -149.7448   492.3278  -0.304    0.762
## `TG/G`          6.8607    87.4011   0.078    0.938
## `CF%`           3.3923     9.8450   0.345    0.732
## `TG/G`:`CF%`    0.1138     1.7494   0.065    0.948
##
## Residual standard error: 11.5 on 57 degrees of freedom
## Multiple R-squared:  0.3938, Adjusted R-squared:  0.3619
## F-statistic: 12.34 on 3 and 57 DF,  p-value: 2.507e-06
```

Since neither of the interaction terms are significant, the best fit model is as follows:

```
coefficients(model_stepwise2_reduced_team_pts)
```

```
## (Intercept)      `TG/G`       `CF%`
## -181.625735   12.538322    4.030869
```

$$\widehat{Points_{future}} = -181.63 + 12.5 * TG/G + 4.0 * CF\%$$

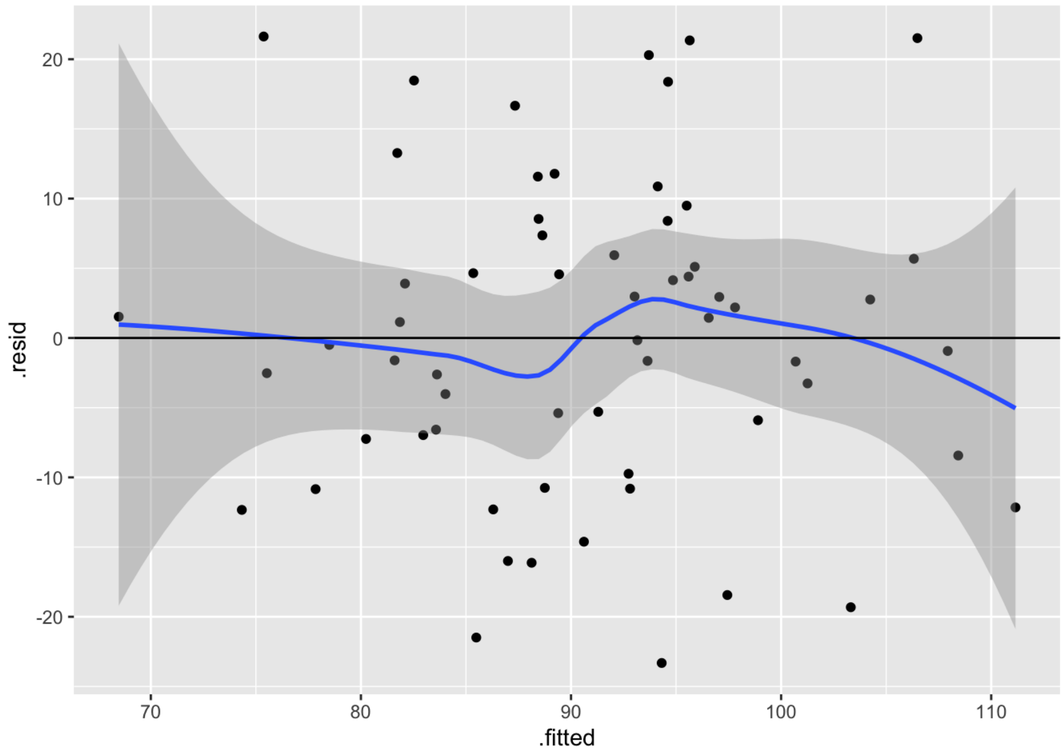
Based on this model, with every one percent increase in CF%, a teams next season points total will increase by ~4. Similarly, with every one unit increase in TG/G, a teams next season points total will increase by ~12.5.

In order to ensure that this model is acceptable, a few additional assumptions will be tested:

- Linearity
- Equal variance
- Normality
- Outlier

The linearity and equal variance assumptions will be tested by plotting the fitted versus the residual values on a scatter plot and visually determining if a trend can be observed.

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Since the plot above displays no observable pattern between the residuals and the fitted values of the model, the linearity and equal variance assumptions hold. Additionally, the Breusch-Pagan test can be used to confirm the equal variance assumption quantitatively.

```
bptest(model_stepwise2_reduced_team_pts)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model_stepwise2_reduced_team_pts
## BP = 0.1035, df = 2, p-value = 0.9496
```

Since the p-value from the Breusch-Pagan test is greater than 0.05, we can confirm that heteroscedasticity is not present.

The normality assumption will be tested using the Shapiro-Wilk test.

```
shapiro.test(residuals(model_stepwise2_reduced_team_pts))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(model_stepwise2_reduced_team_pts)
## W = 0.98268, p-value = 0.5406
```

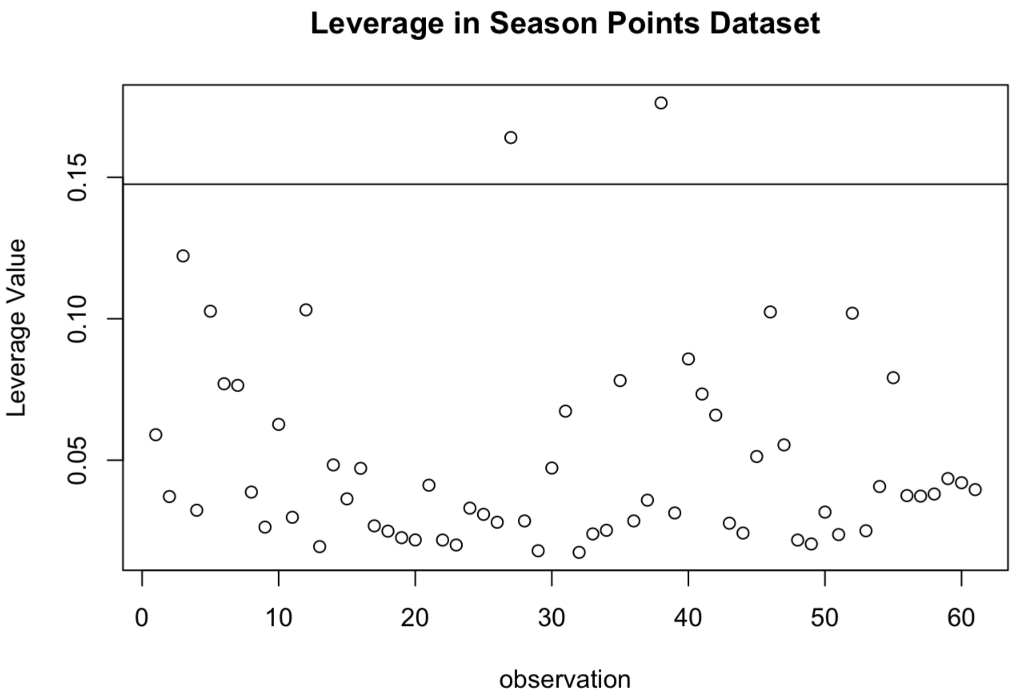
Since the p-value from the Shapiro-Wilk test is greater than 0.05, we can say that the sample data are significantly normally distributed, thus confirming the normality assumption.

Lastly, the outlier assumption will be tested using the leverage points methodology.

```
lev <- hatvalues(model_stepwise2_reduced_team_pts)
p <- length(coef(model_stepwise2_reduced_team_pts))
n <- nrow(seasons)
outlier <- lev[lev > (3 * p / n)]
outlier
```

```
##        27        38 
## 0.1640350 0.1762808
```

```
plot(rownames(seasons), lev, main = "Leverage in Season Points Dataset",
     xlab="observation",
     ylab = "Leverage Value")
abline(h = 3 *p/n, lty = 1)
```



Since there are two outliers that affect the model, a new model will be tested without these outliers.

```
final_model_no_outliers <- lm(data = seasons[-c(27, 38),], FUTURE_PTS ~ `TG/G` + `CF%`)
summary(final_model_no_outliers)
```

```
## 
## Call:
## lm(formula = FUTURE_PTS ~ `TG/G` + `CF%`, data = seasons[-c(27, 
##     38), ])
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -24.616  -7.415   0.858   5.556  22.737 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -178.7886    45.8873  -3.896 0.000263 ***
## `TG/G`        10.6470     4.6153   2.307 0.024781 *  
## `CF%`          4.1864     0.7513   5.572 7.46e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 11.51 on 56 degrees of freedom
## Multiple R-squared:  0.3925, Adjusted R-squared:  0.3708 
## F-statistic: 18.09 on 2 and 56 DF,  p-value: 8.678e-07
```

Now that the outliers have been removed and all other assumptions have been tested, the final best fit model becomes:

$$\widehat{Points_{future}} = -178.8 + 10.6 * TG/G + 4.2 * CF\%$$

Based on this model, with every one percent increase in CF%, a teams next season points total will increase by ~4.2. Similarly, with every one unit increase in TG/G, a teams next season points total will increase by ~10.6.

Conclusion

Conclusion - Logistic Regression

The Best Model from our study is

$$\hat{\pi}_{win} = \frac{e^{0.22-0.04X_{FA_{avg}}+0.23X_{GF_{avg}}-0.22X_{GA_{avg}}-0.40X_{xGF_{avg}}+0.42X_{xGA_{avg}}+0.07X_{SCF_{avg}}-0.04X_{SCA_{avg}}+0.03X_{Blocks_{avg}}+0.04X_{PIM_{avg}}+0.10X_{xGF_{avg}}*X_{SCA_{avg}}-0.01X_{SCF_{avg}}*X_{SCA_{avg}}}}{1+e^{0.22-0.04X_{FA_{avg}}+0.23X_{GF_{avg}}-0.22X_{GA_{avg}}-0.40X_{xGF_{avg}}+0.42X_{xGA_{avg}}+0.07X_{SCF_{avg}}-0.04X_{SCA_{avg}}+0.03X_{Blocks_{avg}}+0.04X_{PIM_{avg}}+0.10X_{xGF_{avg}}*X_{SCA_{avg}}-0.01X_{SCF_{avg}}*X_{SCA_{avg}}}}$$

Interpretations of AIC, Residual Deviance, ROC

To recap, AIC, Residual Deviance, AUC, ROC Curve for the final additive model and our final interactive models are:

Final Additive Model:

- Null deviance: 6650.8 on 4829 degrees of freedom
- Residual deviance: 6518.0 on 4820 degrees of freedom
- AIC: 6538
- AUC: 0.5971

Final Interactive Model:

- Null deviance: 6650.8 on 4829 degrees of freedom
- Residual deviance: 6509.1 on 4818 degrees of freedom
- AIC: 6533.1
- AUC: 0.5993

For the purposes of picking the best model, we selected the final interactive model because of the higher AUC, lower AIC and lower residual deviance.

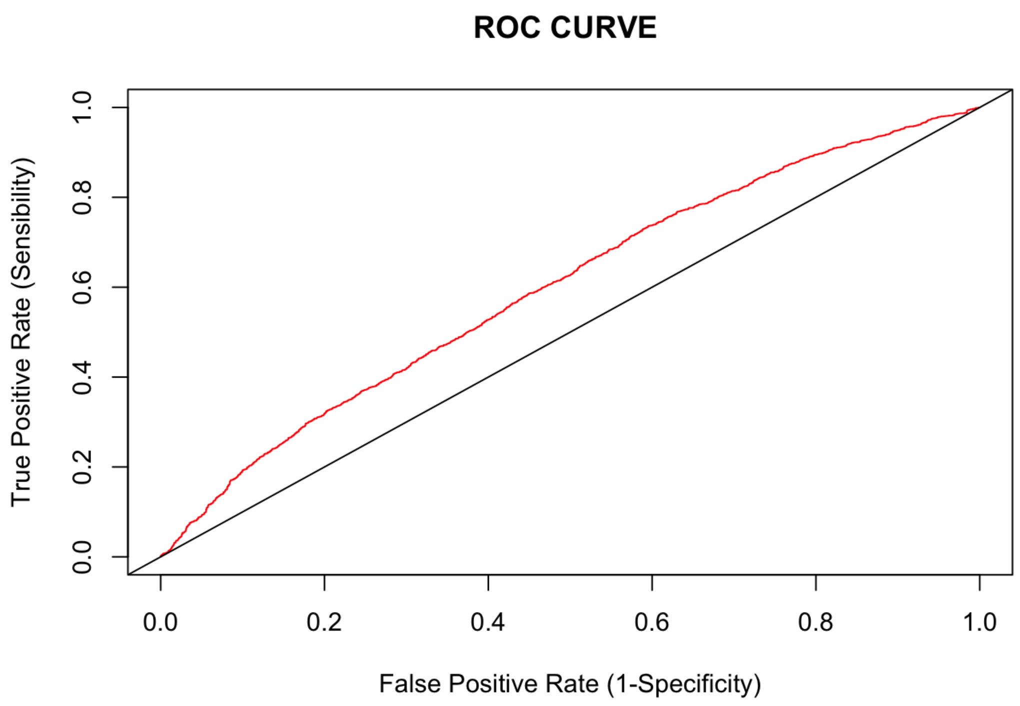
The residual deviance describes how well our response variables is predicted by the independent variables in our model. The values residual deviance values between both our additive model and interactive model show a decreased value. This indicates that our interactive model increases the significance and precision of predicting the probability of a win over our additive model and is the better model. The Residual Deviance has reduced by 8.9 points with a loss of two degrees of freedom.

The AIC is another value we can utilize to help compare one model to another model. A model with a smaller AIC number is the model to choose when selecting between two models. In comparing between the AIC numbers between our additive and interactive model, the AIC value has decreased and also indicates that our interactive model is better than our additive model. The AIC value has decreased by 4.9 points.

The ROC plots us the Sensitivity of our model against Specificity. In short, we are plotting the probability whether the true values are true and false values as false. The AUC (Area Under Curve) quantitatively tells us this relationship; the higher the AUC value, the better the model predicts the probability of our response variable. Looking at our AUC value, the value has increased by 0.0022, which indicates that our interactive model is a better predictive model.

The ROC chart is given:

```
plot(perf,col=2,main="ROC CURVE ", xlab="False Positive Rate (1-Specificity)",ylab="True Positive Rate (Sensibility)")
abline(0,1)
```



```
roc<-roc(nhl.reduced$result_bool,prob)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc(roc)
```

```
## Area under the curve: 0.5993
```

The effects of each independent variable on the model

β_1 for $X_{FA_{avg}}$ means that for every 1 increases for $X_{FA_{avg}}$, we estimate the odds of winning to be multiplied by -0.040508

β_2 for $X_{GF_{avg}}$ means that for every 1 increases for $X_{GF_{avg}}$, we estimate the odds of winning to be multiplied by 0.225015

β_3 for $X_{xGA_{avg}}$ means that for every 1 increases for $X_{xGA_{avg}}$, we estimate the odds of winning to be multiplied by - 0.224650

β_4 for $X_{xGF_{avg}}$ means that for every 1 increases for $X_{xGF_{avg}}$, we estimate the odds of winning to be multiplied by - 0.398329

β_5 for $X_{xGA_{avg}} $ means that for every 1 increases for $X_{xGA_{avg}}$, we estimate the odds of winning to be multiplied by + 0.424882

β_6 for $X_{SCF_{avg}}$ means that for every 1 increases for $X_{SCF_{avg}}$, we estimate the odds of winning to be multiplied by + 0.074618

β_7 for $X_{SCA_{avg}}$ means that for every 1 increases for $X_{SCA_{avg}}$, we estimate the odds of winning to be multiplied by - 0.043757

β_8 for $X_{Blocks_{avg}}$ means that for every 1 increases for $X_{Blocks_{avg}}$, we estimate the odds of winning to be multiplied by 0.025562

β_9 for $X_{PIM_{avg}}$ means that for every 1 increases for $X_{PIM_{avg}}$, we estimate the odds of winning to be multiplied by 0.036574

β_{10} for $X_{xGF_{avg}} * X_{SCA_{avg}}$ means that for every 1 increases for $X_{xGF_{avg}} * X_{SCA_{avg}}$, we estimate the odds of winning to be multiplied by 0.097357

β_{11} for $X_{SCF_{avg}} * X_{SCA_{avg}}$ means that for every 1 increases for $X_{SCF_{avg}} * X_{SCA_{avg}}$, we estimate the odds of winning to be multiplied by - 0.009528

```
head(results, 4)
```

```
##   prediction result    game_id       date  test
## 1  0.7221502      1 2018020029 2018-10-07  TRUE
## 2  0.4587589      0 2018020030 2018-10-07  TRUE
## 3  0.2857085      1 2018020032 2018-10-08 FALSE
## 4  0.2585069      1 2018020033 2018-10-08 FALSE
```

```
mean(results$test)
```

```
## [1] 0.5743134
```

Now that we have both our equations and deviance, AIC, AUC numbers, we now want to answer two questions that we introduce in the introduction.

Recapping, the question that we have set out to answer in this report was:

1. Can build a model to predict the outcome of a hockey game that is better than flipping a coin?

Recapping, our model was built built off of 2014-2017 data. In our R-Chunk above, we've taken 2018 results into a separate dataframe. To test, we want to run our model against 2018 results, to see if we've accurately predicted results of 2018 games against the actual outcome of those 2018 games. In the end, we were correct 57.43% of the time. Success!

Conclusion - Multiple Linear Regression

The best fit multiple linear regression model for predicting team's points totals is as follows:

$$\widehat{Points}_{future} = -178.8 + 10.6 * TG/G + 4.2 * CF\%$$

This model is the best fit model for a number of reasons:

- The model does not have any insignificant variables
- The model has the highest adjusted R-squared out of the models without insignificant variables
- The model includes independent variables that do not provide redundant information
- The data used in the model is significantly normal
- The linearity assumption holds
- There are no significant outliers that impact the model significantly
- The error terms of the model have a constant variance

Based on the best fit model, we can say that with every one percent increase in CF%, a teams next season points total will increase by ~4.2. Similarly, with every one unit increase in TG/G, a teams next season points total will increase by ~10.6.

Based on an adjusted R-squared value of 0.3708, this model explains 37% of the variation in a team's points total. Additionally, based on the RMSE value of 11.51, the standard deviation of the unexplained variance in the model is 11.51.

Discussion

Discussion - Logistic Regression

In creating our logistic regression model, the first point of interest involved data wrangling. Hockey, once started as a pastime obsession, has slowly evolved to a mathematical study to quantify victory beyond the score between two teams. As such, traditional hockey statistics have now spawned a number of modern statistics and further derivations of those statistics. Shots on Goal begat Shot Attempts. Shot Attempts begat Corsi. Corsi begat Fenwick and so on and so on. The interesting lessons learned from our model really indicated how many variables correlate, to where it was imperative that the model had to be reduced immediately by checking multicollinearity with VIF tests. Alternatively, the lesson learned was that the possibility of creating models based on all hockey stats seems almost limitless. (We will one day, create a hockey stat or model called, "The Thuntida")

In some aspects, the results of our model were both expected and unexpected. Our model considered both traditional hockey statistics (e.g. GF_avg, blocks_avg) and modern analytical statistics (e.g. SCF, FA_avg). This indicated to us that neither category of statistics dominates the other. The unexpected result of our model were the different kinds of statistics that were included and significant to our model. A statistic like penalty minute average was significant in our model, and more significant than a blocked shot average.

One big lesson learned was that the combination of data wrangling along with the variations statistical modelling method considerations ultimately made predicting hockey wins/losses limitless. Another lesson learned were the range of statistics that we used in creating our model. We considered all statistics and data gathered from the outset of a season. The randomness of the first 10-15 games may have affected our model more than we originally would have thought. In reading other similar studies done, it is not uncommon to use a predictive model against a rolling window of games to predict outcomes, and this is something that we would have liked to try against our model.

Discussion - Multiple Linear Regression

The results of the best fit multiple linear regression model are in line with what was expected going into this model building exercise. Although there are a significant number of variables available to predict a team's points totals, it is challenging to predict future results based on past results. In the NHL, there is usually significant change that happens between seasons and to say that one team will perform similarly to how they performed in a previous season, is flawed. With that being said, 37% explained variance is surprisingly high. It would be interesting to use this model to predict the current NHL season to see how the prediction compared to the actual results.

One of the surprising outcomes of the best fit model was the fact that past seasons points totals are not significant in predicting future seasons points total. If a team had 100 points in the previous season, that would not be a good indication that the team would also achieve around 100 points in the next season.

A more interesting and effective model for predicting future seasons points totals would be to incorporate a value of the current roster of players. As discussed previously, the previous season's team is not going to look exactly like the future season's team which is why the best fit model is flawed. Incorporating an aggregation of current player value would help accurately predict the results of the upcoming season.