Final Project COMSCI450.1

Jesse Lubell

2022-03-22

Using NFL Analytics to contextualize Player Performance & Predict Playoff Probability

Figure 1: 2021 Playoff Bracket, courtesy of CBS

Eleanor Roosevelt was quoted saying, *"If life were predictable it would cease to be life, and be without flavor."* Perhaps the same applies to football? You've heard the phrase *"On any Given Sunday"...* That's part of what makes the game great. Football is a noisy and game & subject to randomness & is needlessly difficult to predict. There are befuddling upsets every week that shake even the expert's prior assertions. It takes multiple views of a play from different angles to understand which of the 22 players on the field at any time deserves the blame for a positive play or blame for why a play failed. Understanding how a specific game script failed to materialize or was offset by opposing forces can require multiple re-watches. So while there is a preponderance of advanced analytics on the game, a flood of statistical edges one can gain, in a single-elimination tournament format it's still risky business and the goal is to find small edges here or there. Over the course of the year we can accumulate a larger sample size of data & so conclusions can be more stable but even a full season of games is a limited sample size & finding an edge to exploit can be elusive.

On a macro level, consider how the past season shook out; The '22 Super Bowl featured the Cincinnati Bengals, the team with the longest pre-season odds of winning the AFC, per Pro Football Reference *(https:*

//www.pro-football-reference.com/years/2021/preseason_odds.htm)

The Bengals were +$15,000 to get to the Super Bowl (so an event with roughly implied odds of 0.66%, whereas their opponent, the Rams were given pre-season odds of +$1200 which implies a 8.33% probability, *(a magnitude of 12.5x greater)* The odds given to the Bengals were higher than even those applied to the pre-season Jacksonville Jaguars, a team that won 3 games all season! *(For reference that means that a $100 bet placed on Cincinnati would've cashed $150,000)* A lot had to fall into place for the Bengals to rise above the odds, but they are a good case study in how a lower echelon team can become a Conference winner in a short window.

So my initial point is that even with access to advanced analytics, in any given season there's a magnitude of randomness, chaos, luck, injuries and external forces that impacts how teams and players perform over an 18 week season & trying to fit all of that into a linear or logistic regression model to predict Football success appears to be a fools errand, but with that disclaimer in hand I intend to do just that. The goal of this project is to first join a number of these independent data sets & I will then try to isolate the variables that may be semi-useful predictors of player performance and team performance & create a binary model to predict a team's likelihood of making the Playoffs, the shared goal of all 32 teams every year.

Hypothesis

Hypothetical Scenario: You are the GM of the 2021 Carolina Panthers. Your club has just finished 5-12 & in last place in the NFC South & 4 wins short of the playoffs. Your club is 31st/32 in Weighted DVOA, 32nd/32 in EPA per Play on Offense, but ranked 7th/32 in Defensive EPA per Play.

The owner of the club has informed you that while the upcoming season schedule looks a little more favorable than the previous year you must make the playoffs this year or else you will be fired. You have $45M you can free up in Salary Cap (minus dead money) to spend on Free Agent signings & you have a handful of picks to use in the the '22 draft to select players to add value to the club. Your coach, playcallers and scheme design are above average and considered for this exercise to be stable & competent.

Hypothesis: The recipe for earning the playoff berth is one that prioritizes:

1. Top 8 QB play *(Using PFF's QB Power Rating Metric)*
2. Shifting resources to having a Top 12 Passing Defense *(measured by Defensive Dropback EPA, data from NflFast)*
3. Maintaining a roughly league-average total Defense *(measured by Football Outsider's Defensive DVOA metric)*
4. Taking into account Strength of Schedule *(measured by PFF)*
5. Maintain a below-league average team variance *(measured by Football Outsider's variance metric which looks at a team's consistency ratings in their DVOA measure* and can also be extrapolated as the value added by above average coaching or scheme).

With those metrics in hand we can predict with a good deal of accuracy whether a team will make the playoffs.

Loading Libraries & Joining independent Data Sets

```
#Code Starts here and will be interspersed with blocks of analysis

library(tidyverse)
library(ggrepel)
library(ggimage)
library(nflfastR)
```

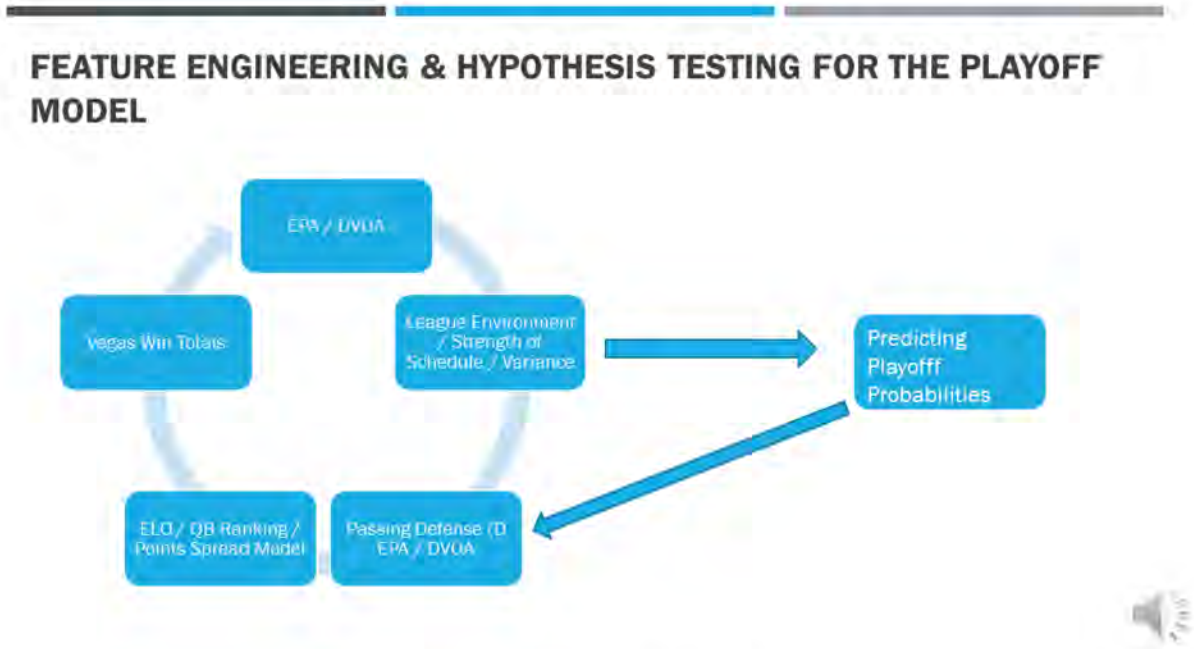


Figure 2: Model of the Hypothesis

```
library(scatterplot3d)
library(nflplotR)
library(ggplot2)
library(nflreadr)
library(dplyr, warn.conflicts = FALSE)
library(car)
library(reshape2)
library(corrplot)
library(GGally)
library(rgl)

options(scipen = 9999)
```

Data Dictionary & Notes on the Data sources used

The foundation of our data set is PFF player data. (Found at https://www.pff.com/fantasy/stats). PFF is considered the unanimous leader in advanced Anayltics for the league. They collect data from every single snap played and grade each player using a proprietary methodology to ensure consistency and limit bias. The first data set has all of the snap data, receptions, targets, rushing attempts, depth of target data etc for all of the individual players in the league. This data set consists of 594 players and 52 variables. We shrink it down to a subset of 250 players and 17 variables later in this project for ease of use. The PFF data will be joined with Fantasy Pro's ADP data for our analysis involving player performance vs their "average draft position" at the beginning of the year. We also use PFF's power ranking data and betting dashboard metrics for a good chunk of analysis.

Point Spread QB Ratings: The number of points each QB contributes to the Point Spread Team Rating

Point Spread Team Ratings: The number of points each team would be a favorite or underdog to an

average team on a neutral field

Strength of Schedule: The relative difficulty of each team's schedule based on Point Spread Team Ratings of opponents(*Projections: probabilities based off 10,000 season simulations given Team Point Spread Ratings, Strength of Schedule, and team records*)

The second component in our data set is from *Football Outsider's*, another extremely respected source for advanced NFL Analytics. Their proprietary Defense-adjusted Value Over Average (DVOA) system breaks down every single NFL play and compares a team's performance to a league baseline based on situation in order to determine value over average. (https://www.footballoutsiders.com/stats/nfl/team-efficiency/2021/regular). I have included both offensive and defensive metrics as variables in our data set. Those that factor into our models are:

Offensive and Defensive DVOA: Adjusted based on strength of opponent as well as to consider all fumbles, kept or lost, as equal value. SPECIAL TEAMS DVOA is adjusted for type of stadium (warm, cold, dome, Denver) and week of season. As always, positive numbers represent more points so DEFENSE is better when it is NEGATIVE.

Estimated Wins: Uses a statistic known as "Forest Index" that emphasizes consistency as well as DVOA in the most important specific situations: red zone defense, first quarter offense, and performance in the second half when the score is close. It then projects a number of wins adjusted to a league-average schedule and a league-average rate of recovering fumbles. Teams that have had their bye week are projected as if they had played one game per week.

Schedule: Lists average DVOA of opponents played this season, ranked from hardest schedule (#1, most positive) to easiest schedule (#32, most negative). It is not adjusted for which games are home or road. For the current season, the listing of schedule is split into past and future schedules.

Variance: Measures the statistical variance of the team's weekly DVOA performance. Teams are ranked from most consistent (#1, lowest variance) to least consistent (#32, highest variance).

```
#Code Con't
#component one of data set, PFF fantasy data.
Fantasystats <- read.csv("./data_2/PFF_week_17.csv")

summary(Fantasystats)
#Quick check for consistency... on team names/abbreviations

table(Fantasystats$team)

#second component, DVOA data
DVOA <- read.csv("./data_2/2021 Team DVOA Ratings Overall.csv")
table(DVOA$Team)


#Editing Team abbreviations so they match for the merge/join
DVOA[10,1] <- "ARZ"
DVOA[19,1] <- "BLT"
DVOA[28,1] <- "HST"
DVOA[13,1] <- "CLV"
DVOA[5,1] <- "LA"

(merge0 <- merge(Fantasystats, DVOA, by.x = "team", by.y = "Team"))

#3rd component, Defensive EPA data
DefensiveEPA <- read.csv("./data_2/Defensive EPA.csv")
names(DefensiveEPA)
```

```
DefensiveEPA <- DefensiveEPA[, 3:9]#removing unnecessary index columns and team column
#redundancies
table(DefensiveEPA$Abbr)


(merge1 <- merge(merge0, DefensiveEPA,by.x = "team", by.y = "Abbr"))
head(merge1)
tail(merge1)
table(merge1$team)

#4th component ADP Data
FFAdp <- read.csv("./data_2/FantasyPros_ADP.csv")
names(FFAdp)
dim(FFAdp)

(merge2 <- merge(merge1, FFAdp, by.x = "player", by.y = "Player", all = FALSE,
                 all.x = TRUE))


#re-ordering this data frame for easier manipulation of key variables
merge3 <- merge2[c(1,2,4,50:52,81,85,89,5:49,53:80)]
#####################################################################

Finaldive1<- merge3[order(merge3$fantasyPts, decreasing = T),]
Finaldive1
#let's clean up our environment pane and remove these files we no longer need...
remove(DefensiveEPA)
remove(DVOA)
remove(FFAdp)
remove(Fantasystats)
remove(merge0)
remove(merge1)
remove(merge2)
remove(merge3)
```

(Data Dictionary Con't)

The EPA and defensive EPA data come from NFLfast, a terrific R package(https://www.nflfastr.com/). This is a repository of play-by-play data going back to 2006. And is a terrific vessel for scraping NFL data & for generating all of the EPA data we are using in our models. They are also credited with one of the better WAR (Wins above Replacement) models to assess player value.

EPA: These terms are referring to an Expected Points Added metric (EPA) that attempts to quantitatively evaluate a play and return a magnitude and direction for the result of each play's effect on the game in relation to the mean expectation for the game state

Success Rate : An advanced metric in football that measures efficiency, but with the important context of down and distance considered. A play is defined as successful if: It gains at least 50% of the yards required to move the chains on first down. 70% of yards to gain on second down.

Defensive EPA / Defensive Success Rate: Both of these measures capture the Defense's efficiency at defending the opposing team.

ADP: Average Draft Position (Found at: https://www.fantasypros.com/nfl/adp/overall.php) ADP represents the Average Draft Position for players in fantasy football drafts. It serves as a useful draft prep tool

for understanding how players are valued. Our ADP Composite consists of consensus draft values across the most popular league hosts. * Data based on real early season drafts may be from a small sample size.

ADP Tier: I have created my own factor variable consisting of discrete values to represent the flow of a Fantasy Football draft:

Players drafted 1.01 - 2.12, The first two round of a Fantasy Draft are labeled as: "Consensus Top Pick". Players with an ADP of 25-52 are: "Second Tier". Players with an ADP of 53-83 are: considered "Middle Rounds Pick". This is and the next tier are where drafts are truly won or lost. Managers who can find ADP over-performers here typically win their leagues. Players taken at pics 84 - 130, I'm calling: "Value Rounds". Players taken 131 - 220: tier "Late Round & Relevant". And players with an ADP of 221 and beyond I'm grouping in a tier called: "Mostly Undrafted". Fantasy Drafts typically run about 18 rounds so it's rare to see ADP's beyond that number in most analysis. This is why I also created a subset for our analysis that capture the top 250 players in the '21 season.

```
#Code Con't
#5th component - NFL Logos, colors, etc.  Pulled from the NflFast library for plots
data(teams_colors_logos)
#Again checking for consistency with Team abbreviation to make sure the data aligns
#properly...
teams_colors_logos$team_abbr
table(Finaldive1$team)

#Editing these team names for the merge
teams_colors_logos[1,1] <- "ARZ"
teams_colors_logos[3,1] <- "BLT"
teams_colors_logos[13,1] <- "HST"
teams_colors_logos[8,1] <- "CLV"

(Finaldive2 <- merge(Finaldive1, teams_colors_logos, by.x = "team", by.y = "team_abbr",
                     all = FALSE, all.x = TRUE))
Finaldive2 <- Finaldive2[order(Finaldive2$fantasyPts, decreasing = T),]
Finaldive2
remove(Finaldive1)


#6th component is EPA data from either nflFast or rbsdm.com
EPA <- read.csv("./data_2/rbsdm.comstats.csv")
head(EPA)
str(EPA)

EPA <- EPA[c(2,5:10)]
table(EPA$team)


(Finaldive3 <- merge(Finaldive2, EPA, by.x = "team",all = FALSE, all.x = TRUE))
remove(Finaldive2)



#7th component is PFF ELO ranking & Super Bowl Odds from Pro Football Reference
#Final merge?! 2/10 Added SuperBowl Odds from PFR, PFF ELO & QB rating, Strength of schedule
#rank from PFF

PFR <- read.csv(("./data_2/Odds table from PFR.csv"))
```

```
table(PFR$team)

Finaldive4 <- merge(Finaldive3, PFR, by.x = "team", all = FALSE, all.x = TRUE)



teampoints <- aggregate(fantasyPts~team, data = Finaldive3, FUN = sum)
teampoints

#small merge adding aggregated sum Team fantasy points as new variable "teampoints"
names(teampoints)
names(teampoints)[2] <- "teampoints"

(Finaldive <- merge(x=Finaldive4, y=teampoints, by = "team" , all = TRUE))
summary(Finaldive)

#8th component is a table from FiveThirtyEight with Playoff Odds & Actual Wins
Five38b<- read.csv("./data_2/Five38.csv", header = TRUE)


#9th component we created manually, just a table with team name and 1 = Made Playoffs,
#0 = Didn't make playoffs.
#Data manually entered using Pro Football Reference and created in Excel.
Playoff <- read.csv("./data_2/Playoff Results.csv")

Finaldive4 <- merge(x=Finaldive, y=Playoff, by = "team" , all = TRUE)


Finaldive <- merge(x=Finaldive4, y=Five38b, by="team", all = FALSE, all.x = TRUE)
summary(Finaldive)

remove(EPA)
remove(Finaldive3)
remove(Finaldive4)
remove(PFR)
remove(teampoints)
remove(teams_colors_logos)
remove(Five38b)
remove(Playoff)

names(Finaldive) #Admin for creating the subsets which we will use for working with our models....



#Change name of "MAKE.PLAYOFFS" #188 to Playoff.odds
Finaldive <- rename(Finaldive, Playoff.odds = MAKE.PLAYOFFS)
Finaldive <- rename(Finaldive, ADP = Rank)
```

(Data Dictionary Con't)

The last two independent data sets we joined are Super Bowl odds data from Pro Football Reference (https://www.pro-football-reference.com/years/2021/preseason_odds.htm).

I also included a Playoff Model & Win model from 538 so I could use their ELO Ranking. (https://projects.fivethirtyeight.com/2021-nfl-predictions/) This forecast is based on 50,000 simulations of the season and updates after every game. Game metrics are on a 0-100 scale. Quality is determined by the harmonic mean of the teams' Elo ratings; importance measures how much the result will alter playoff projections; the overall number is the average of the quality and importance values.

ELO Ratings: A measure of strength based on head-to-head results and quality of opponent to calculate teams' chances of winning their regular-season games and advancing to and through the playoffs.

ELO QB Ratings: Our quarterback-adjusted Elo model incorporates news reports to project likely starters for every upcoming game and uses our quarterback Elo ratings to adjust win probabilities for those games. A team's current quarterback adjustment is based on its likely starter in its next game and how much better or worse that QB is than the team's top starter

```
#code con't

####### Light Cleaning & Factor Variables
#Changed "RANK" to ADP (Fantasy Pro's default PPR Rankings), Removed Team Record columns
#that imported as dates (I.e. a team with a record
#of 7-10 was showing up as "OCT-7".  Refer to new column "Actual.Wins". Created new factor variable
#for Team Wins:  "Wincat" (levels: Top Quartile -> Bottom Quartile)


Finaldive$wincat <- c()
Finaldive$wincat[Finaldive$Actual.Wins <=7] <- "Bottom Quartile"
Finaldive$wincat[Finaldive$Actual.Wins >7 & Finaldive$Actual.Wins<=9] <- "Median"
Finaldive$wincat[Finaldive$Actual.Wins >9 & Finaldive$Actual.Wins<=11] <- "3rd Quartile"
Finaldive$wincat[Finaldive$Actual.Wins >11] <- "Top Quartile"

Finaldive$wincat <- factor(Finaldive$wincat, levels =c("Top Quartile", "3rd Quartile",
                                                       "Median", "Bottom Quartile"))

table(Finaldive$wincat)
#Top Quartile       3rd Quartile      Median     Bottom Quartile
#113                    106             163             212

#Created ADPtier factor variable grouping players into draft tiers & dealing with NA's found
#in ADP data by essentially grouping them into a bucket we are calling "Mostly Undrafted",
#E.g. A.J. Dillon of the GB Packers, who went undrafted in most leagues and most platforms
#but finished with 179.9fantasy points, finishing #56 on the year.  I wanted to include the
#impactful players like these that generated a lot of production but flew under the radar
#before the season.



Finaldive$ADPtier <- c()
Finaldive$ADPtier[Finaldive$ADP<=24] <- "Consensus Top Pick"
Finaldive$ADPtier[Finaldive$ADP >=25 & Finaldive$ADP <=52] <- "Second Tier"
Finaldive$ADPtier[Finaldive$ADP >=53 & Finaldive$ADP <=84] <- "Middle Rounds Pick"
Finaldive$ADPtier[Finaldive$ADP >=85 & Finaldive$ADP <=130] <- "Value Rounds"
Finaldive$ADPtier[Finaldive$ADP>=131 & Finaldive$ADP <=220] <- "Late Round & Relevant"
Finaldive$ADPtier[Finaldive$ADP>=221] <- "Mostly Undrafted"

Finaldive$ADPtier <- factor(Finaldive$ADPtier, levels = c("Consensus Top Pick", "Second Tier",
                                                          "Middle Rounds Pick", "Value Rounds",
```

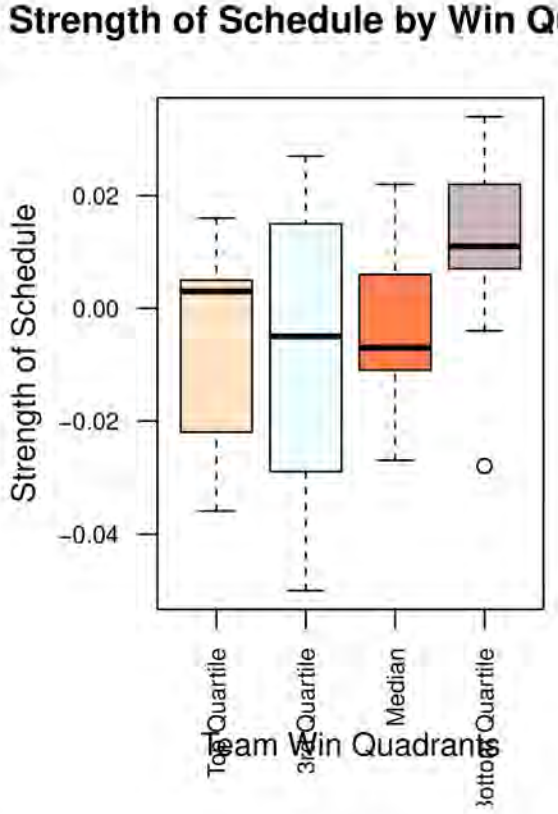
```
                                                    "Late Round & Relevant",
                                                    "Mostly Undrafted"))


#Let's set all of the NA's in the ADPTier to "Undrafted", because these are players that scored a
#reasonable amount of points to be considered relevant and also were undrafted. Most of them are
#injury-related.
Finaldive$ADPtier[is.na(Finaldive$ADPtier)] <- "Mostly Undrafted"

table(Finaldive$ADPtier)

Finaldive<- Finaldive[order(Finaldive$fantasyPts, decreasing = T),]

Finaldive <- rename(Finaldive, Vegas.Wins.DK = W.L.O.U)
Finaldive <- rename(Finaldive, SB.implied.Odds = ignore.1)
Finaldive$Opportunities <- c(Finaldive$recRec + Finaldive$rushCarries)
Finaldive <- rename(Finaldive, ELO_QB = ï..ELO.WITH.TOP.QB)
Finaldive$ELO_QB <- as.numeric(Finaldive$ELO_QB)


#Saving a version of the new data set to ensure consistency
#write.csv(Finaldive, "Finaldive.csv", row.names = FALSE)

Finaldive2 <- Finaldive



FFsubset <- Finaldive[which(Finaldive$fantasyPts >=50 |
                              Finaldive$ADPtier!="Mostly Undrafted") , ]
str(FFsubset)


#also keeping this version of the data set bc it has all of the variables.  Some of the plots
#call for 'Finaldive2'

#Now to trim doum our variables for our regression models

#For our ADP GLM Mode
FFdive<- FFsubset %>%
  dplyr::select(player, team, position, fantasyPts, teampoints, ADP, ADPtier, wincat,
                Point.Spread.Rating.QB, Vegas.Wins.DK, rac, rzRushCarries,
                Point.Spread.Rating.Points, ELO_QB, Success.Rate..SR.,
                recTarg, depth, ptsPerTouch, rzRecRec, yac,rzRecTargPct, ypc,
                Dropback.EPA, rzRecRecPct, Variance) %>%
  arrange(desc(fantasyPts))


Finaldive<- Finaldive[c(1,121,111:112,127,106,2:4,5:35,126,115:117,93,95,97,94,
                        96,98,55:57,59:68, 118,101,69:82,125,108,124,100,103:107,
                             86,87, 91,90)]
Finaldive <- Finaldive[which(Finaldive$fantasyPts >=50 |
                              Finaldive$ADPtier!="Mostly Undrafted") , ]
```

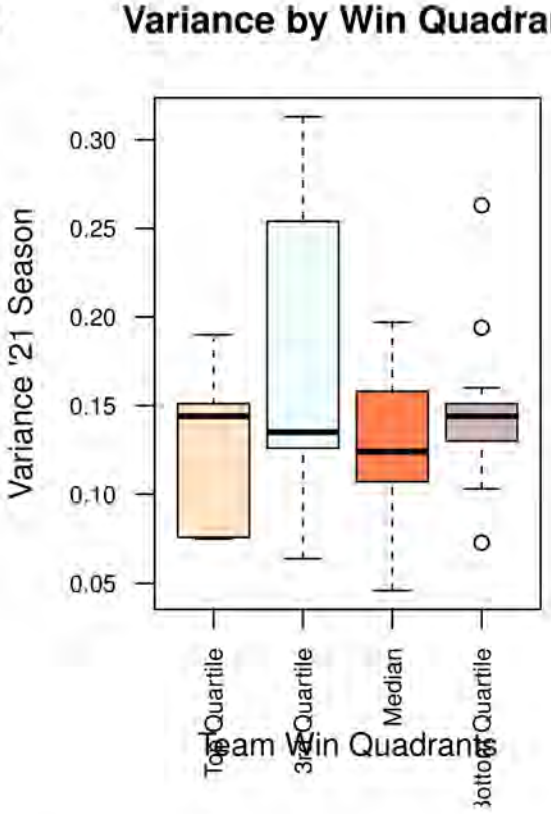
```
#Converting a couple of variables into factor variables with levels
Finaldive$team <- factor(Finaldive$team)
Finaldive$position <- factor(Finaldive$position)
remove(FFsubset)
```

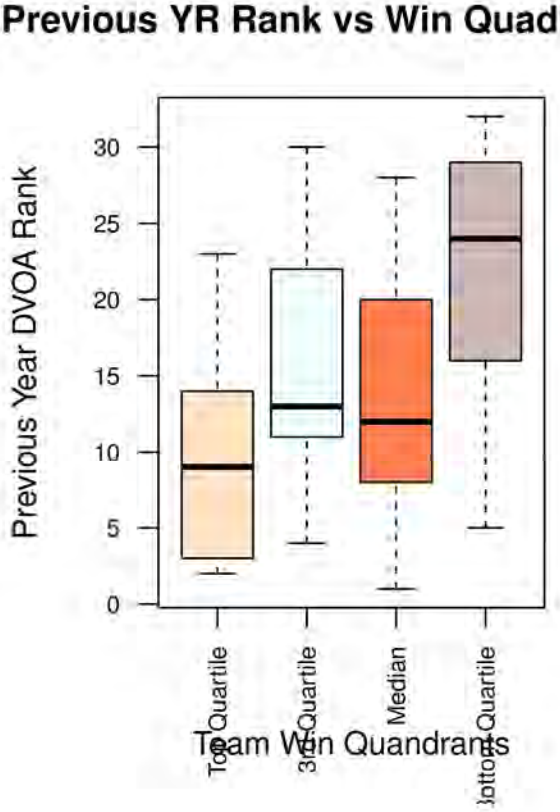
(Data Dictionary Con't)

A quick note on the FFsubset: I chose to Subset this smaller slice of this data with top approx 240 players (to mirror a 20+ Rounds in a deep draft format league. The floor here is artificially low but I'd like to see which drafted players failed to accrue a very modest 50 points in the season. Explanations for this range from suspension, to injury to "other" - See Calvin Ridley or Antonio Brown. This smaller subset removes the handful of undrafted players who scored less than 50 points & who fall in the Mostly Undrafted category (Rank >= 221) so it doesn't factor into our relevant data st.

Now that we have our clean data sets, our subsets & paired down variables to consider for feature engineering in our models let's take a look at our key variables:


Strength of Schedule by Win Quad
Strength of Schedule
0.02
0.00
−0.02
−0.04
Top Quartile
3rd Quartile
Median
Bottom Quartile
Team Win Quadrants


Variance by Win Quadrant
Variance '21 Season
0.30
0.25
0.20
0.15
0.10
0.05
Top Quartile
3rd Quartile
Median
Bottom Quartile
Team Win Quadrants


Previous YR Rank vs Win Quadrant
Previous Year DVOA Rank
30
25
20
15
10
5
0
Top Quartile
3rd Quartile
Median
Bottom Quartile
Team Win Quandrants

In the data transformation portion of this project I created a factor variable simply grouping Teams into 4 buckets for the '21 Season based on their actual win totals. Teams with 11 or more Wins I labeled as "Top Quartile". These were the elite teams during the past season: Arizona, Buffalo, Dallas, Green Bay, Kansas City, LA Rams, Tampa Bay and Tennessee. The next tier of teams were those who had more than 8 but less than 11: Cincinnati, Indy, LA Chargers, Las Vegas, Miami, Philly, Pittsburgh, San Francisco; Nearly all of these teams made the playoffs. The bottom tier consisted of the 7 teams that had 6 or less wins. And Median tier was a smattering of teams that showed promise at the beginning of the year but fell apart due to injuries (Baltimore, Cleveland, Seattle), average teams in very competitive conferences (Denver) and teams with high variance from week to week, losing many of the close games they were in (Minnesota, Washington). Let's see what similarities each tier has viewed through the lens of the feature variables that will inform our models. What we can glean from our first set of box plots is that teams with High EPA appear to be consistently in the playoffs. Strength of schedule appears to be mostly flat, but teams in the lowest tier did show to have harder schedules compared to league average. The Previous Year DVOA variable shows me that because of the parity in the league an average team or even worse than average team can still easily move into a higher tier the following year, an encouraging piece of data to inform our hypothesis.

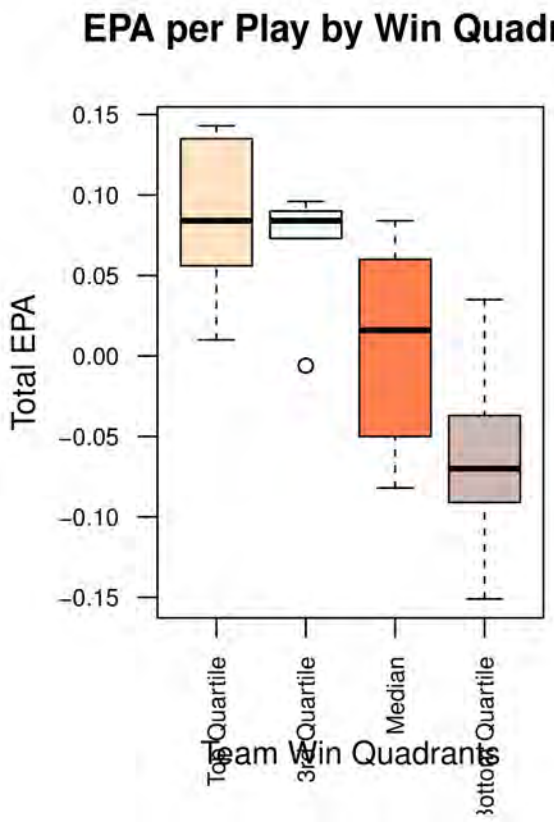
```
wincatrable <- table(Finaldive$team, Finaldive$wincat)
knitr::kable(
  wincatrable[, 1:4],
  caption = "Teams grouped by 'Win Category'"
)
```

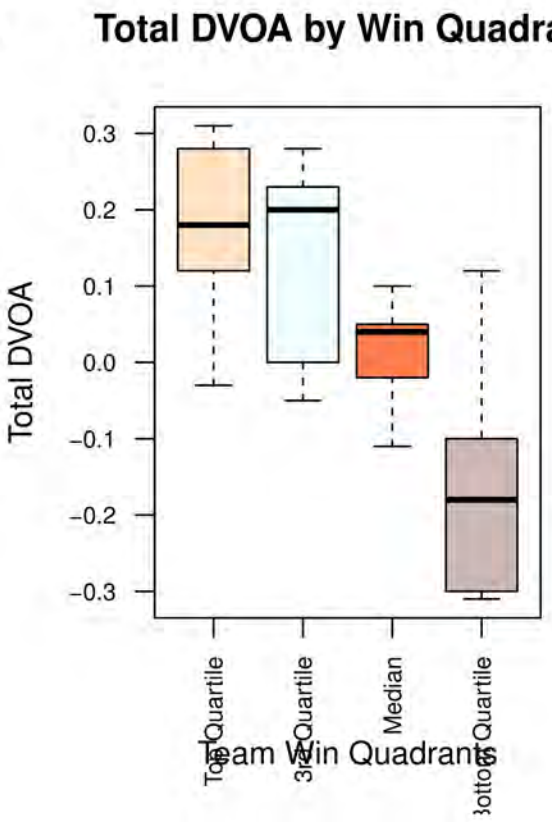
Table 1: Teams grouped by 'Win Category'

	Top Quartile	3rd Quartile	Median	Bottom Quartile
ARZ	0	8	0	0
ATL	0	0	0	7
BLT	0	0	7	0
BUF	0	8	0	0
CAR	0	0	0	5
CHI	0	0	0	8
CIN	0	6	0	0
CLV	0	0	9	0
DAL	8	0	0	0
DEN	0	0	0	7
DET	0	0	0	6
GB	7	0	0	0
HST	0	0	0	6
IND	0	0	9	0
JAX	0	0	0	8
KC	8	0	0	0
LA	7	0	0	0
LAC	0	0	8	0
LV	0	9	0	0
MIA	0	0	8	0
MIN	0	0	6	0
NE	0	9	0	0
NO	0	0	6	0
NYG	0	0	0	8
NYJ	0	0	0	9
PHI	0	0	8	0
PIT	0	0	8	0
SEA	0	0	0	10

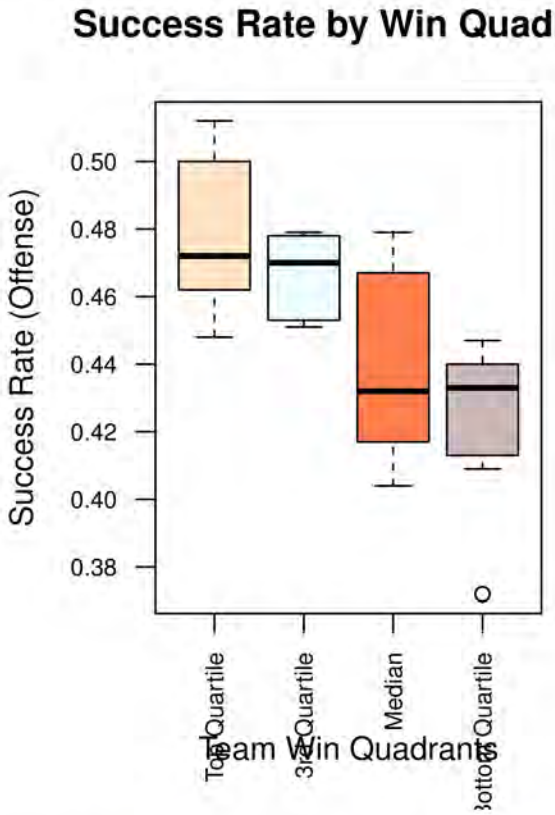
	Top Quartile	3rd Quartile	Median	Bottom Quartile
SF	0	9	0	0
TB	9	0	0	0
TEN	10	0	0	0
WAS	0	0	0	9

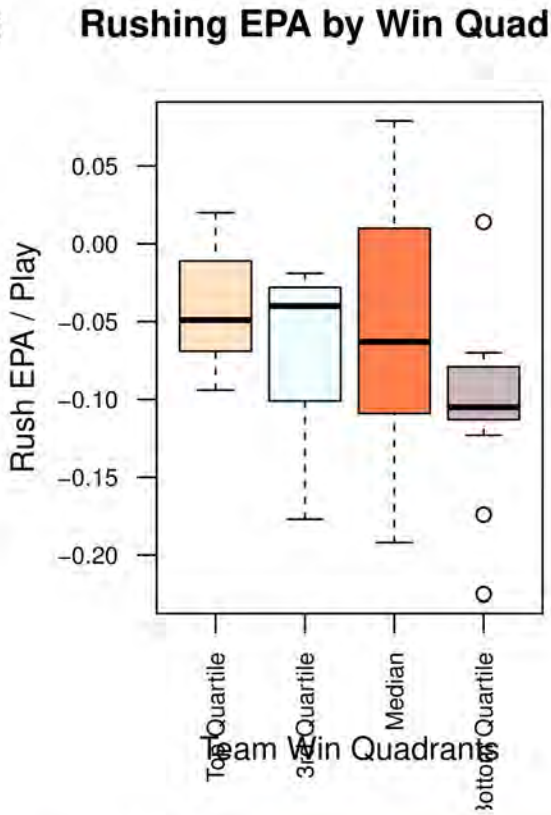
```
par(mfrow = c(1,2))
boxplot(Finaldive$EPA.play~Finaldive$wincat, main="EPA per Play by Win Quadrant",
        data=Finaldive, las = 2, cex.axis=.75, col=c("bisque","azure", "coral","mistyrose3"),
        ylab="Total EPA", xlab = "Team Win Quadrants")


boxplot(Finaldive$Total.DVOA~Finaldive$wincat, main="Total DVOA by Win Quadrant",
        data=Finaldive, las = 2, cex.axis=.75, col=c("bisque","azure", "coral","mistyrose3"),
        ylab="Total DVOA", xlab = "Team Win Quadrants")
```



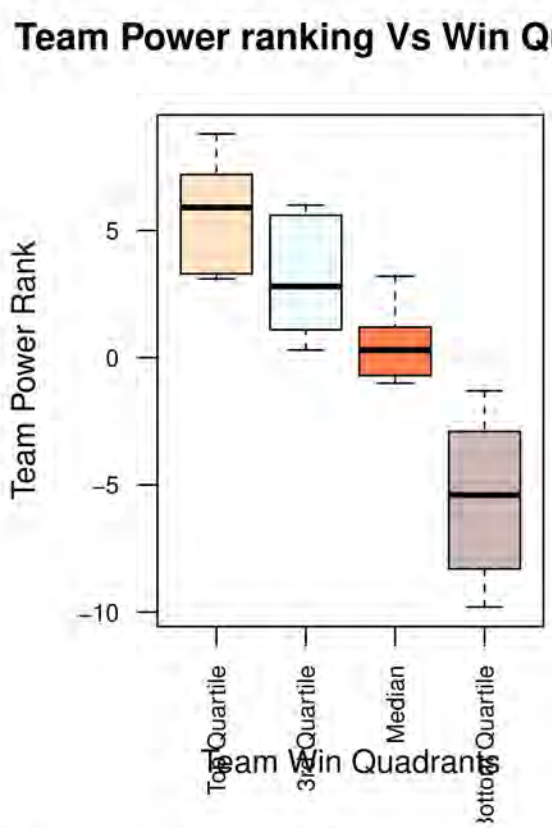



```
boxplot(Finaldive$Success.Rate..SR.~Finaldive$wincat, main="Success Rate by Win Quadrant",
        data=Finaldive, las = 2, cex.axis=.75, col=c("bisque","azure", "coral","mistyrose3"),
        ylab="Success Rate (Offense)", xlab = "Team Win Quadrants")

boxplot(Finaldive$Rush.EPA~Finaldive$wincat, main="Rushing EPA by Win Quadrant",
        data=Finaldive2, las = 2, cex.axis=.75, col=c("bisque","azure", "coral","mistyrose3"),
        ylab="Rush EPA / Play", xlab = "Team Win Quadrants")
```

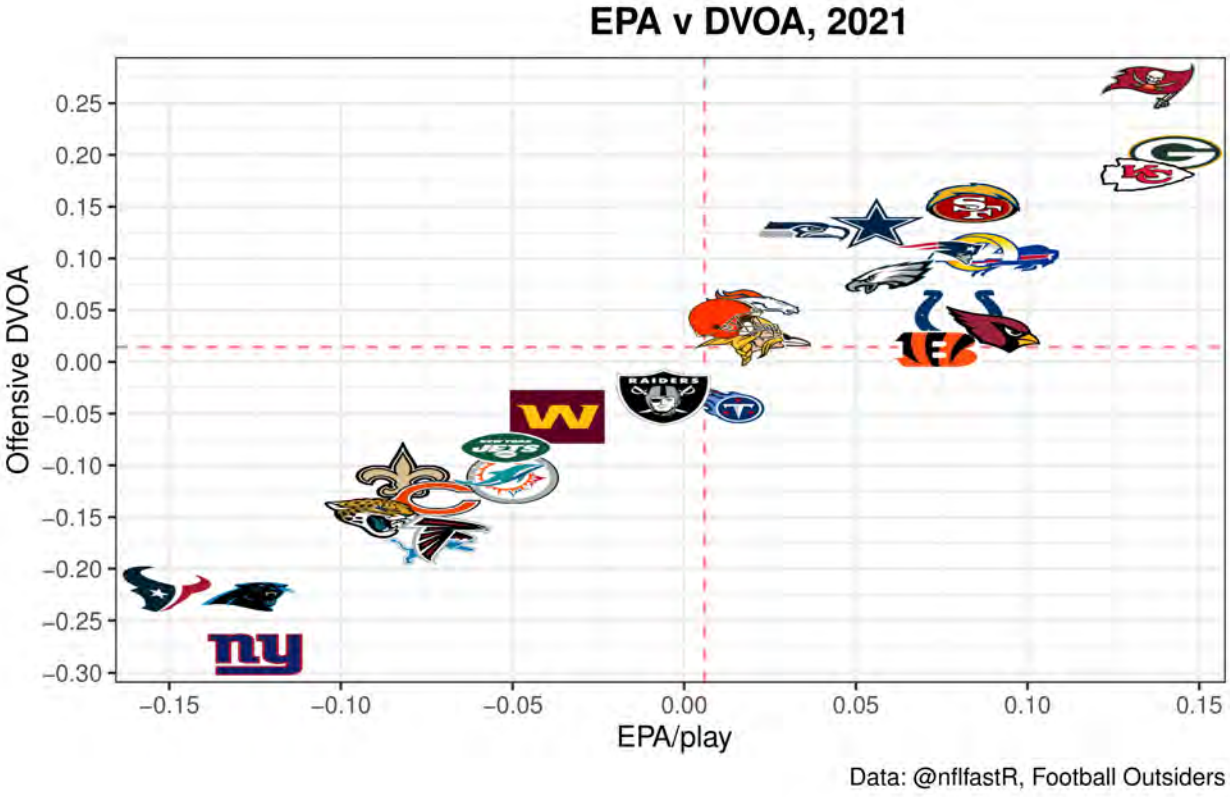


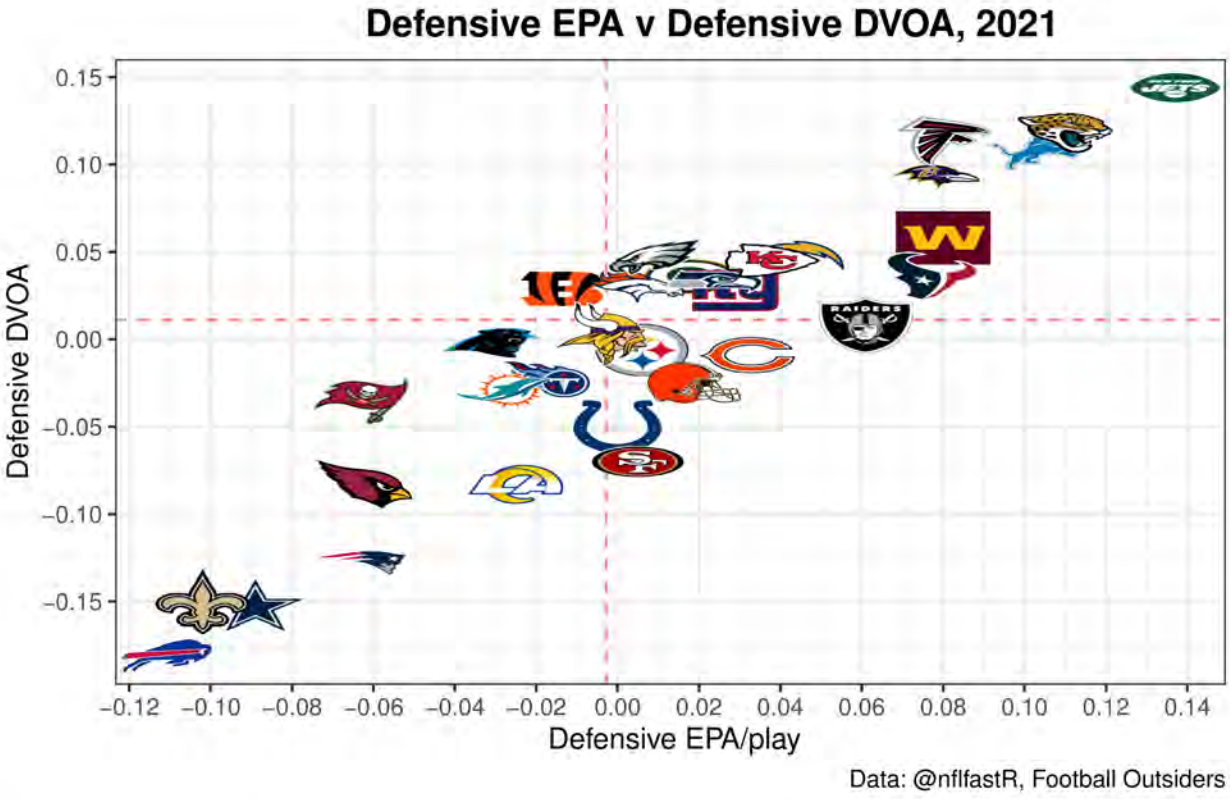



```
boxplot(Finaldive$Point.Spread.Rating.Points~Finaldive$wincat, main="Team Power ranking Vs Win Quadrant
        data=Finaldive, las = 2, cex.axis=.75, col=c("bisque","azure", "coral","mistyrose3"),
        ylab="Team Power Rank", xlab = "Team Win Quadrants")
```



From the next set of Box Plots we can see that DVOA and EPA, completely independent measures of offensive and defensive efficiency both correlate very highly by Team Win category. We will consider this when choosing which to include and which to omit from our models to avoid multicollinarity. Success Rate is a little more flat but has a similar relationship. The first evidence that we want to prioritize Pass EPA over Rush EPA is seen in the third plot. Differences between the Elite teams in the league and those in the absolute bottom tier are negligible when it comes to a team's Rushing efficiency (as measured by Rush EPA)

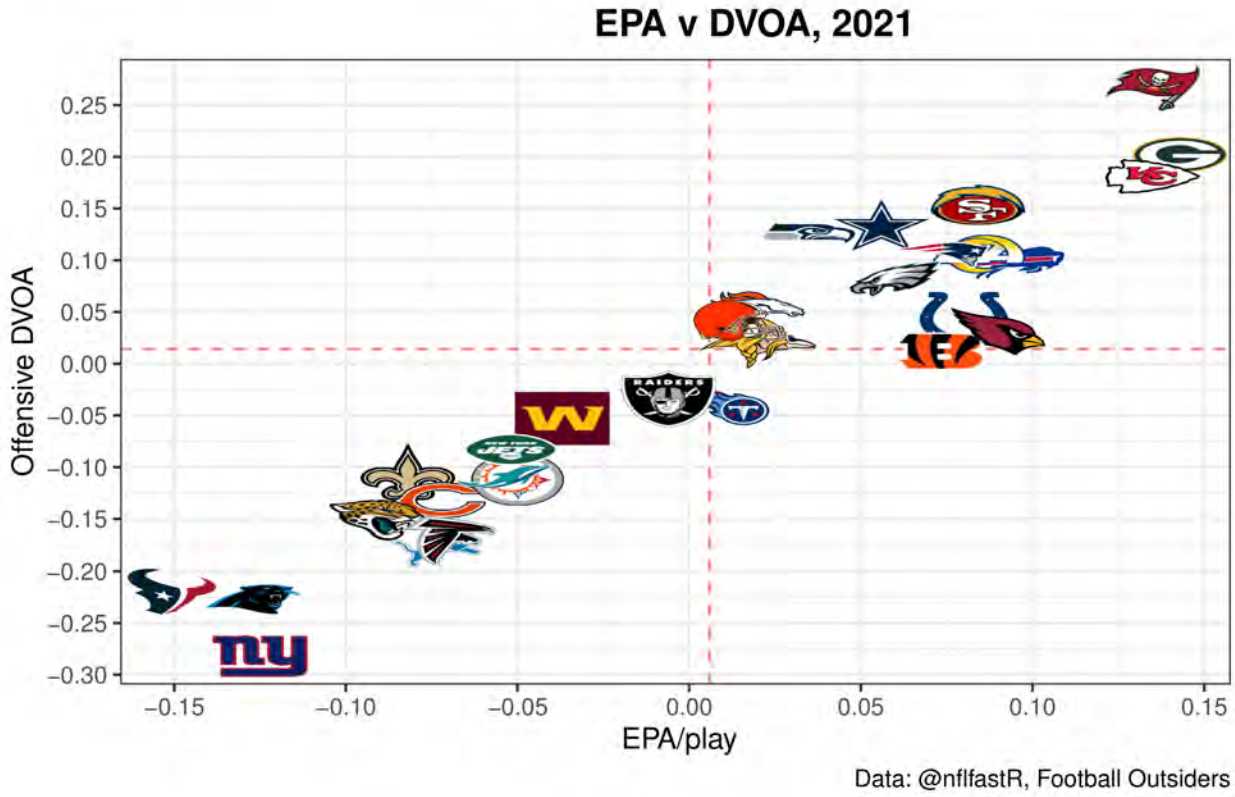

EPA v DVOA, 2021
Offensive DVOA
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
−0.30
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
EPA/play
Data: @nflfastR, Football Outsiders



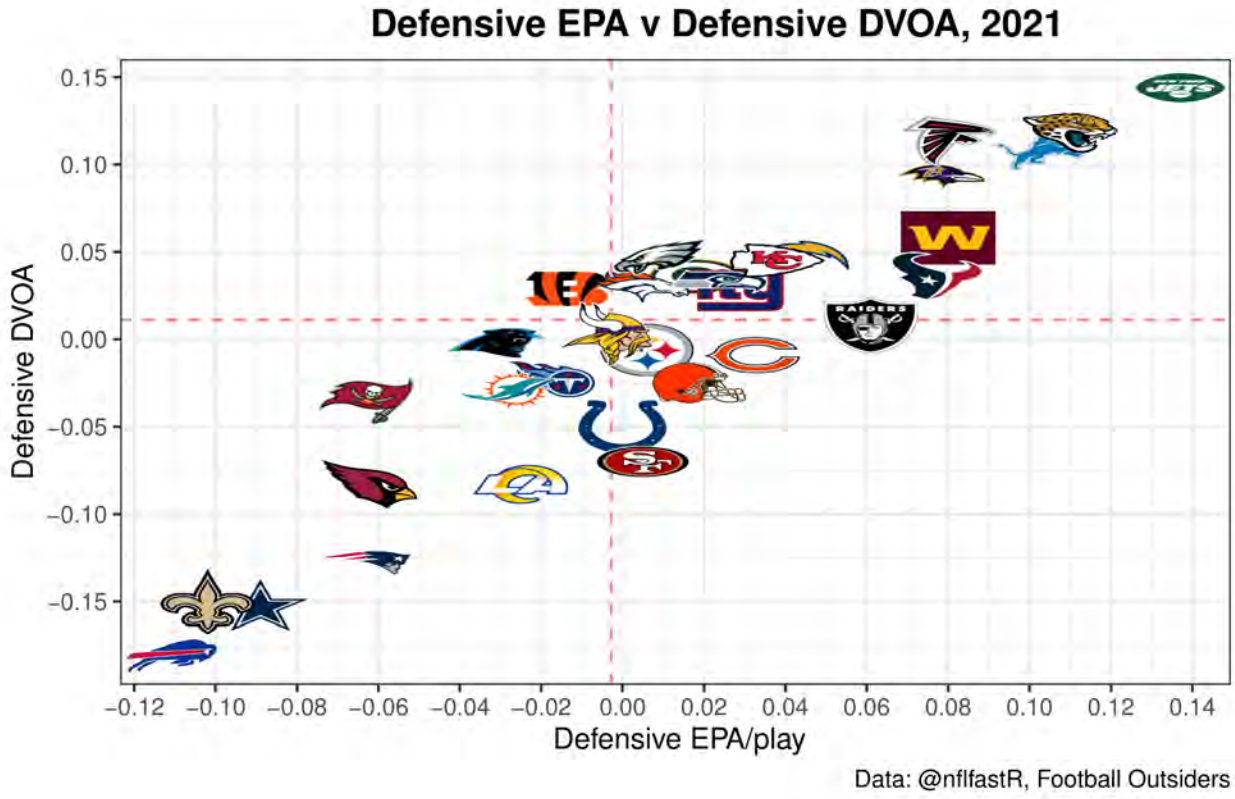
The most telling of our plots thus far as the one examining Team's with elite QB play versus those without and the impact of having a Top player at this critical position. It's no surprise that there's a massive gulf between the good teams and the bad teams in terms of their Win totals and whether they made the playoffs. Similarly the teams with the lowest Dropback EPA Defense *(a lower score/rank is better)* have nearly the same clustering as the teams with Top QB's. For our hypothesis I'd like to explore those two variables through the frame of linear / logistic regression to see if we can spot some real patterns. The strength of schedule variable clearly plays a role in our calculations but the two plots don't sharply illustrate just how. More to come on that. The lower the value on the Y axis, the harder the schedule. But there doesn't appear to be a direct correlation with strength of schedule and Team Win totals that we can suss out from this visualization.

```
Finaldive %>%
  ggplot(aes(x = EPA.play, y = Offense.DVOA)) +
  geom_hline(yintercept = mean(Finaldive$EPA.play), color = "red", linetype = "dashed", alpha=0.5) +
  geom_vline(xintercept =  mean(Finaldive$Offense.DVOA), color = "red", linetype = "dashed", alpha=0.5)
  geom_image(aes(image = team_logo_wikipedia), size=.085) +
  labs(x = " EPA/play",
       y = "Offensive DVOA",
       title = "EPA v DVOA, 2021",
       caption = "Data: @nflfastR, Football Outsiders") +
  theme_bw() +
  theme(
    aspect.ratio = 9 / 16,
    plot.title = element_text(size = 14, hjust = 0.6, face = "bold")
  ) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
```

```
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10))
```

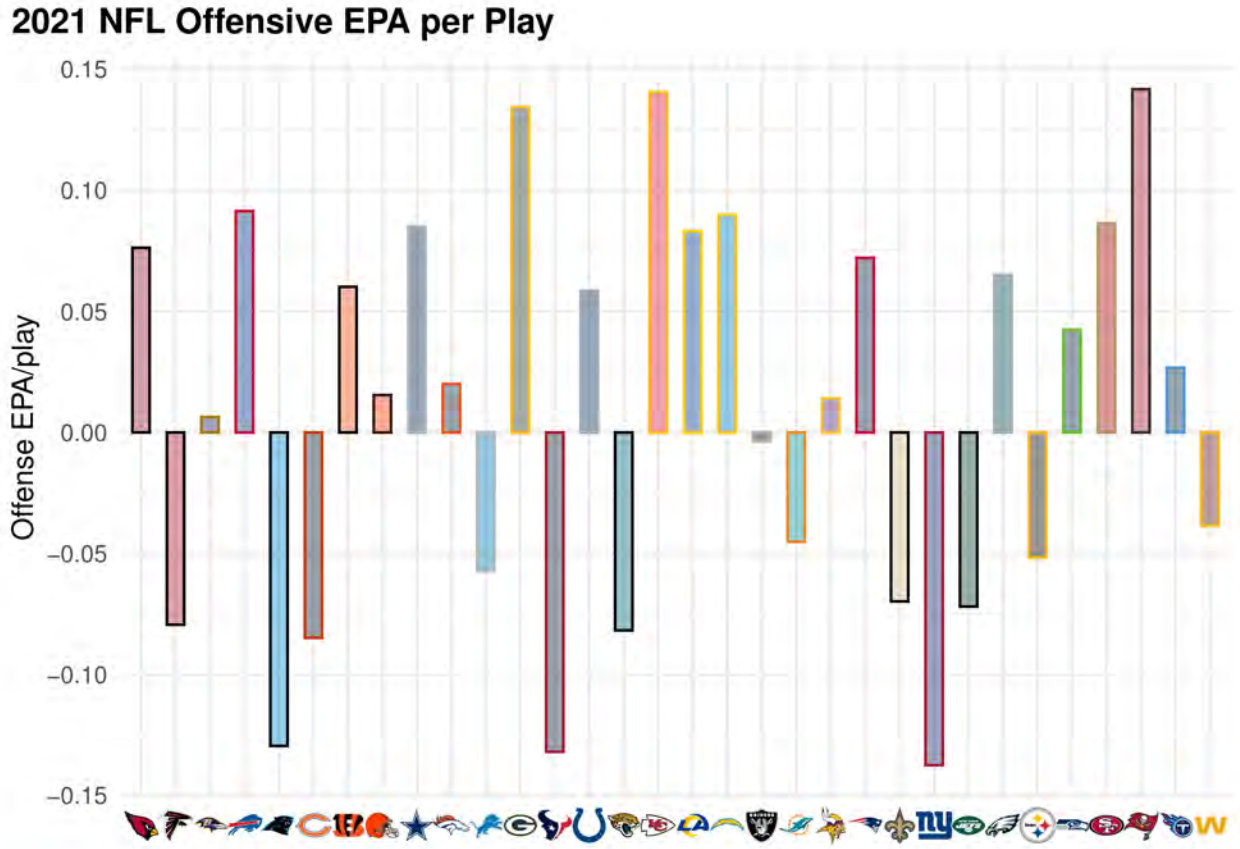



```
Finaldive %>%
  ggplot(aes(x = D.EPA.play, y = Defense.DVOA)) +
  geom_hline(yintercept = mean(Finaldive$D.EPA.play), color = "red", linetype = "dashed", alpha=0.5) +
  geom_vline(xintercept =  mean(Finaldive$Defense.DVOA), color = "red", linetype = "dashed", alpha=0.5)
  geom_image(aes(image = team_logo_wikipedia), size=.085) +
  labs(x = " Defensive EPA/play",
       y = "Defensive DVOA",
       title = "Defensive EPA v Defensive DVOA, 2021",
       caption = "Data: @nflfastR, Football Outsiders") +
  theme_bw() +
  theme(
    aspect.ratio = 9 / 16,
    plot.title = element_text(size = 14, hjust = 0.6, face = "bold")
  ) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10))
```




```
#####And Courtesy of NFLfast
pbp <- nflreadr::load_pbp(2021) %>%
  dplyr::filter(season_type == "REG") %>%
  dplyr::filter(!is.na(posteam) & (rush == 1 | pass == 1))

offense <- pbp %>%
  dplyr::group_by(team = posteam) %>%
  dplyr::summarise(off_epa = mean(epa, na.rm = TRUE))


defense <- pbp %>%
  dplyr::group_by(team = defteam) %>%
  dplyr::summarise(def_epa = mean(epa, na.rm = TRUE))

combined <- offense %>%
  dplyr::inner_join(defense, by = "team")



ggplot2::ggplot(offense, aes(x = team, y = off_epa)) +
  ggplot2::geom_col(aes(color = team, fill = team), width = 0.5) +
  nflplotR::scale_color_nfl(type = "secondary") +
  nflplotR::scale_fill_nfl(alpha = 0.4) +
  ggplot2::labs(
    title = "2021 NFL Offensive EPA per Play",
    y = "Offense EPA/play"
```

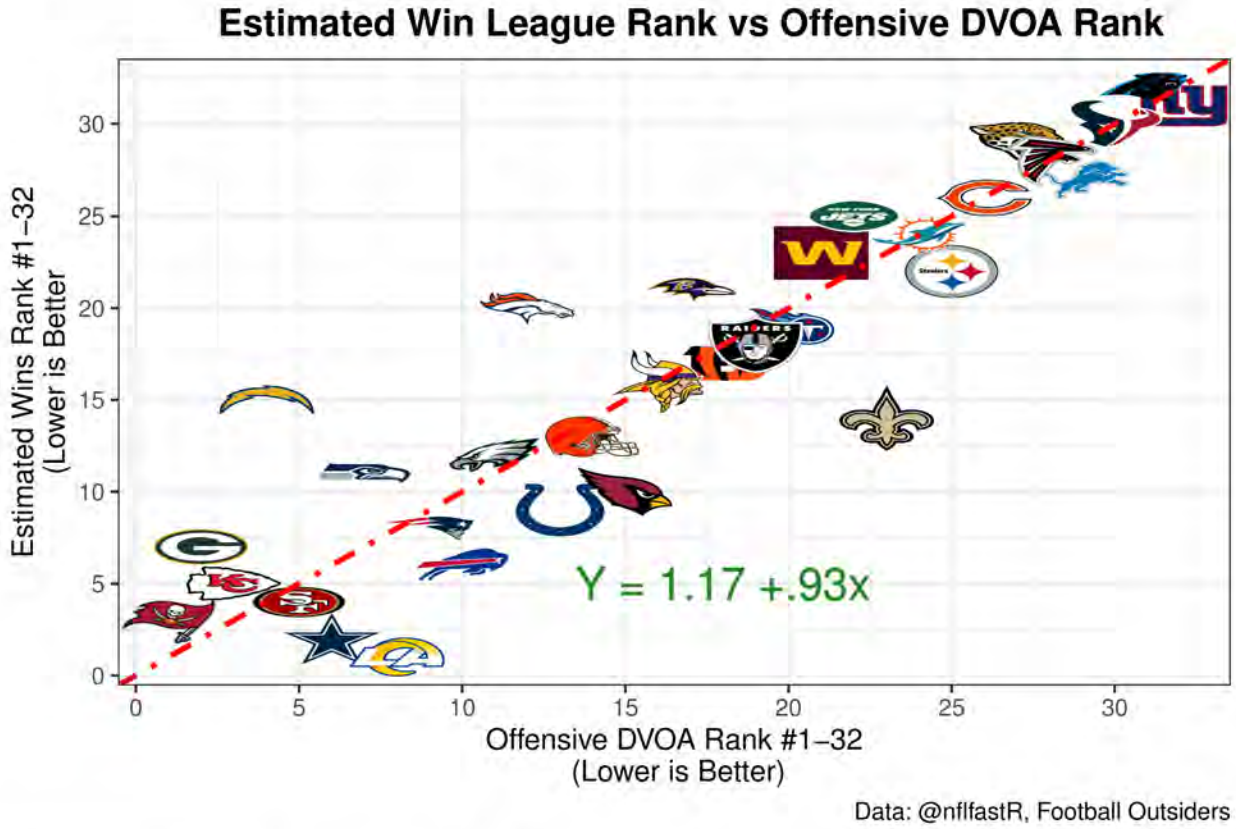
```
) +
ggplot2::theme_minimal() +
ggplot2::theme(
  plot.title = ggplot2::element_text(face = "bold"),
  plot.title.position = "plot",
  # it's obvious what the x-axis is so we remove the title
  axis.title.x = ggplot2::element_blank(),
  # this line triggers the replacement of team abbreviations with logos
  axis.text.x = element_nfl_logo()
)
```



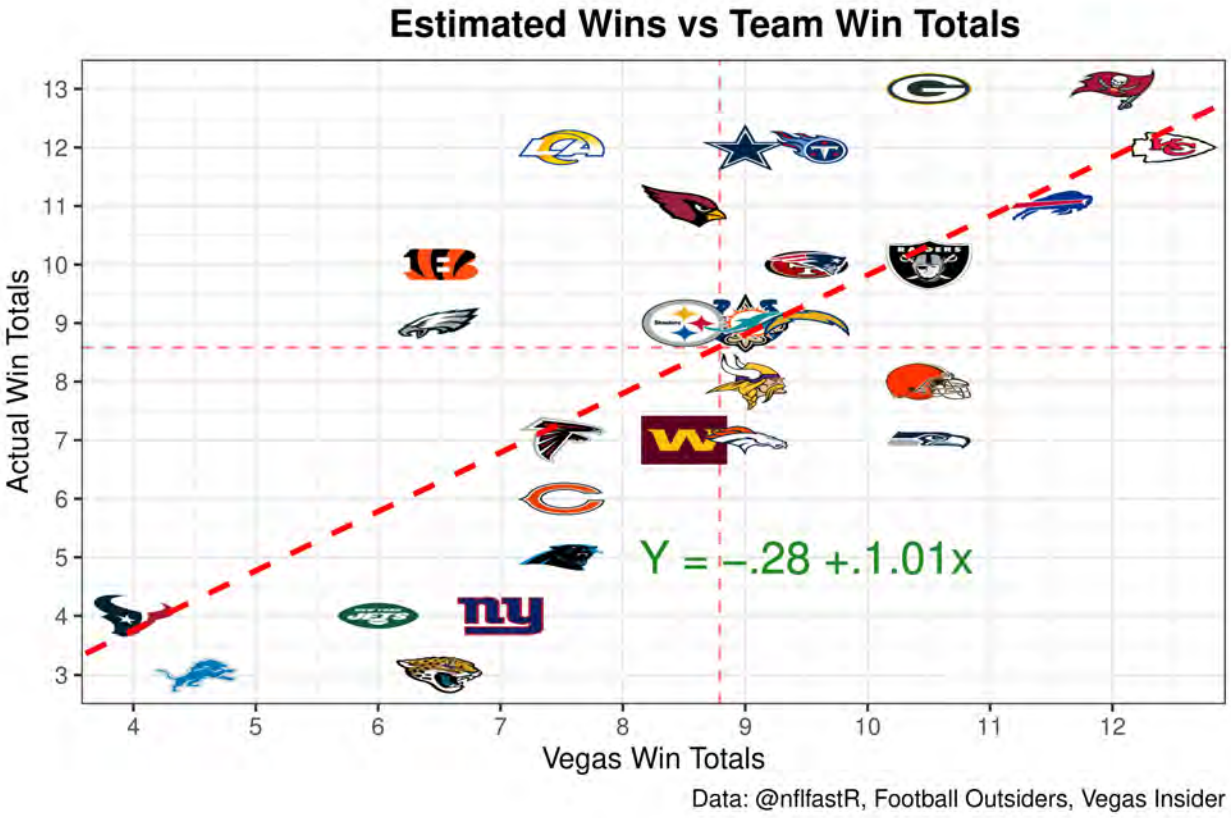

```
##### Here is back to the Linear Regression of two variables
WinRank <- Finaldive%>%
  ggplot(aes(x = Offense.DVOA.Rank, y = Estimated.Wins.Rank)) +
  #geom_hline(yintercept = mean(Finaldive$Offense.DVOA.Rank), color = "red", linetype = "dashed", alpha
  #geom_vline(xintercept =  mean(Finaldive$Estimated.Wins.Rank), color = "red", linetype = "dashed", al
  geom_image(aes(image = team_logo_wikipedia), size=.085) +
  labs(x = "Offensive DVOA Rank #1-32\n (Lower is Better)",
       y = "Estimated Wins Rank #1-32\n(Lower is Better",
       title = "Estimated Win League Rank vs Offensive DVOA Rank",
       caption = "Data: @nflfastR, Football Outsiders") +
  theme_bw() +
  theme(
    aspect.ratio = 9 / 16,
    plot.title = element_text(size = 14, hjust = 0.6, face = "bold")
```

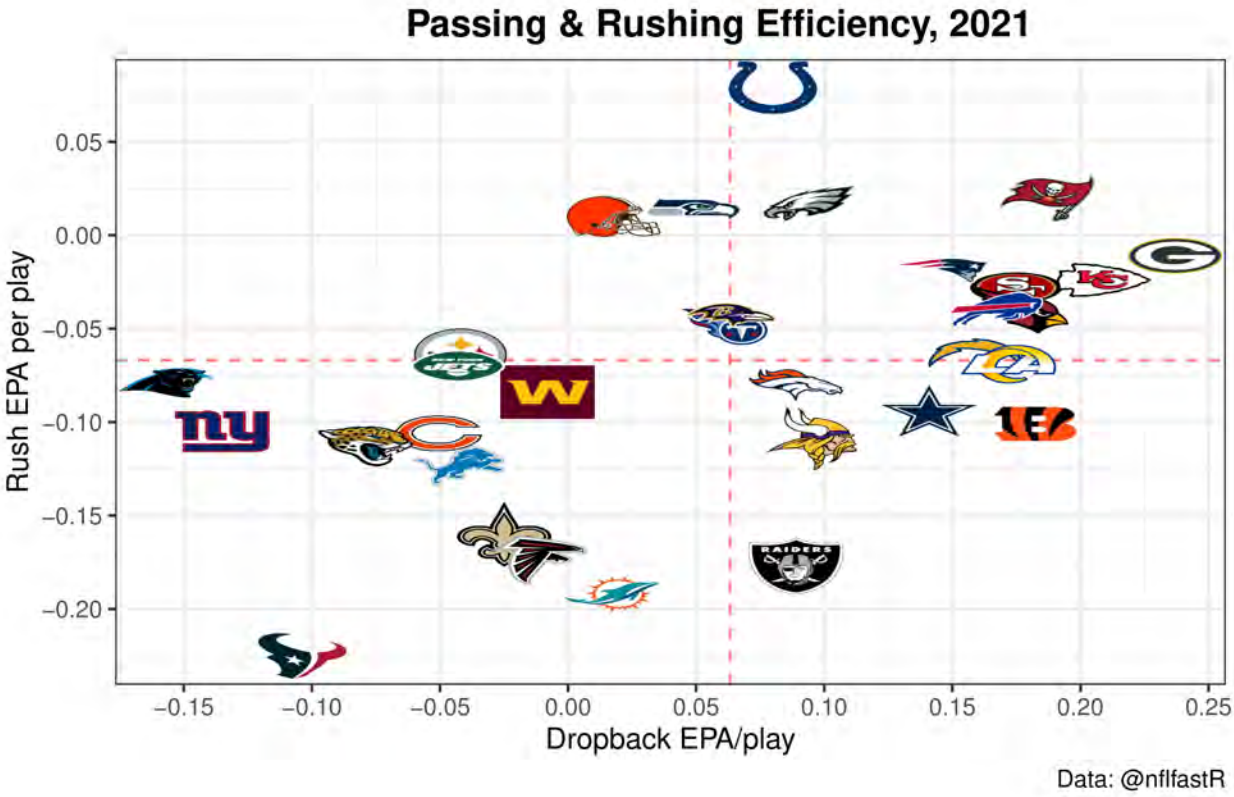
```
  ) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10))

WinRank1 <- WinRank + geom_abline(color = "red", linetype = "dotdash",
                                  size = 1)

WinRank1 + geom_text(overlap = FALSE, x=18, y=5, col = "forestgreen", size = 6,
                     label = "Y = 1.17 +.93x")
```

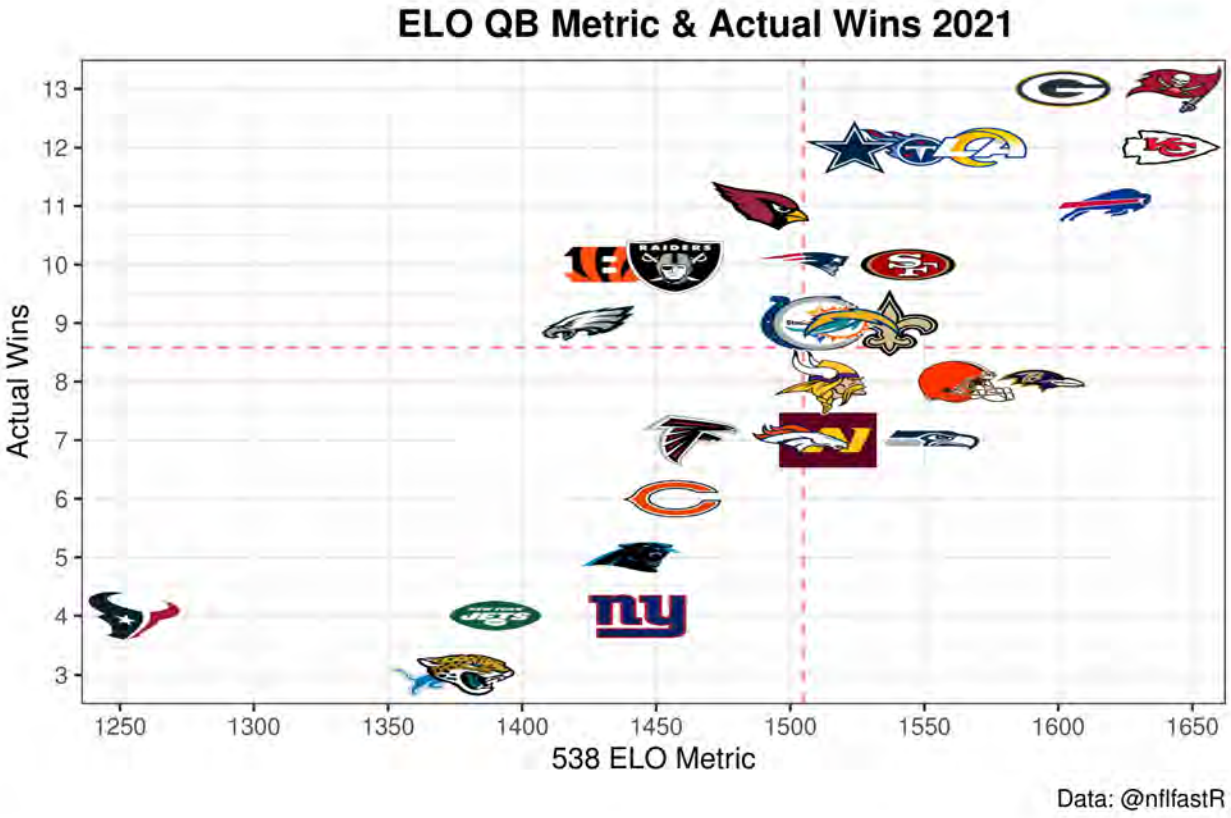
```
## Warning: Ignoring unknown parameters: overlap
```



While EPA and DVOA measure the same thing while applying different methodologies it's clear that we can just select one of them on either side of the ball for our model building. Defensive DVOA and EPA paint a slightly different picture. But also be aware of that a negative DVOA is actually the goal. The teams in the bottom left quadrant of the second plot shows the best defensive teams in the league this past year. But as you can see with the Bengals sitting at the intersection of league mean, that some years defense is less important than offense as it relates to post season success.


Estimated Wins vs Team Win Totals
Actual Win Totals
Vegas Win Totals
Y = −.28 +.1.01x
Data: @nflfastR, Football Outsiders, Vegas Insider


Passing & Rushing Efficiency, 2021
Rush EPA per play
0.05
0.00
−0.05
−0.10
−0.15
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
Dropback EPA/play
Data: @nflfastR

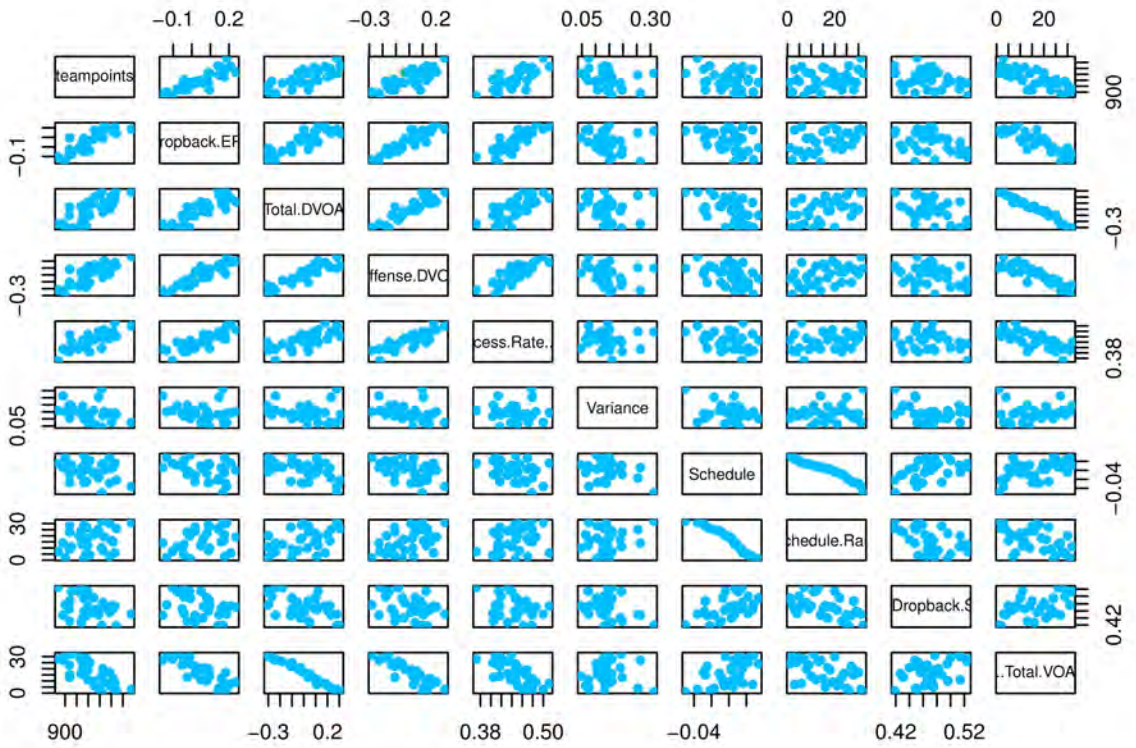


What's also evident here is that Vegas has shown to be fairly efficient in giving us Win Totals. You can see that above the regression line are the teams that overachieved based on their pre-season win totals. The teams below din't hit their win totals. The point of this is at once this data is refined in late July / August, sharp bettors can begin building models based on LY data, the new win totals and improvement index values to get a credible silhouette of how the season will look.

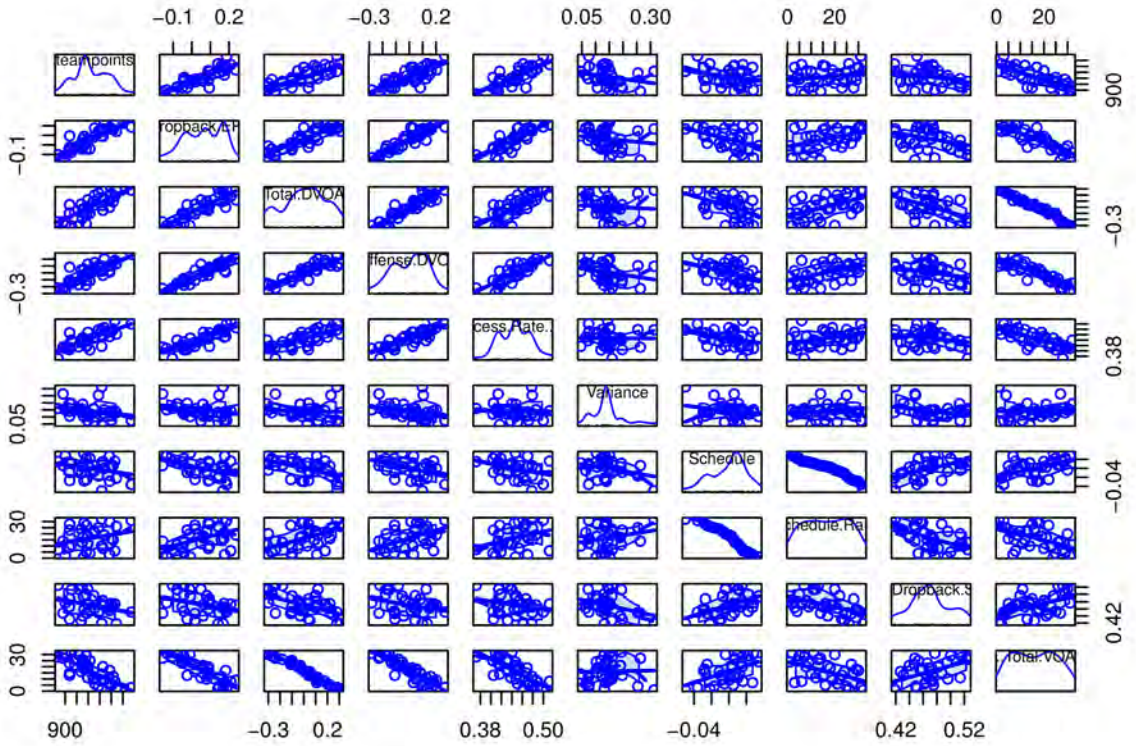
Now for our Model Building and Evaluations

Looking at our correlation matrix here, it's clear that we're going to run into some multi-colinearity issues down the road but at this point I just want to make sure we're not dealing with crazy outliers or oddities & as long as win totals project as "normal distributions" we can reduce variables later when we build the models.

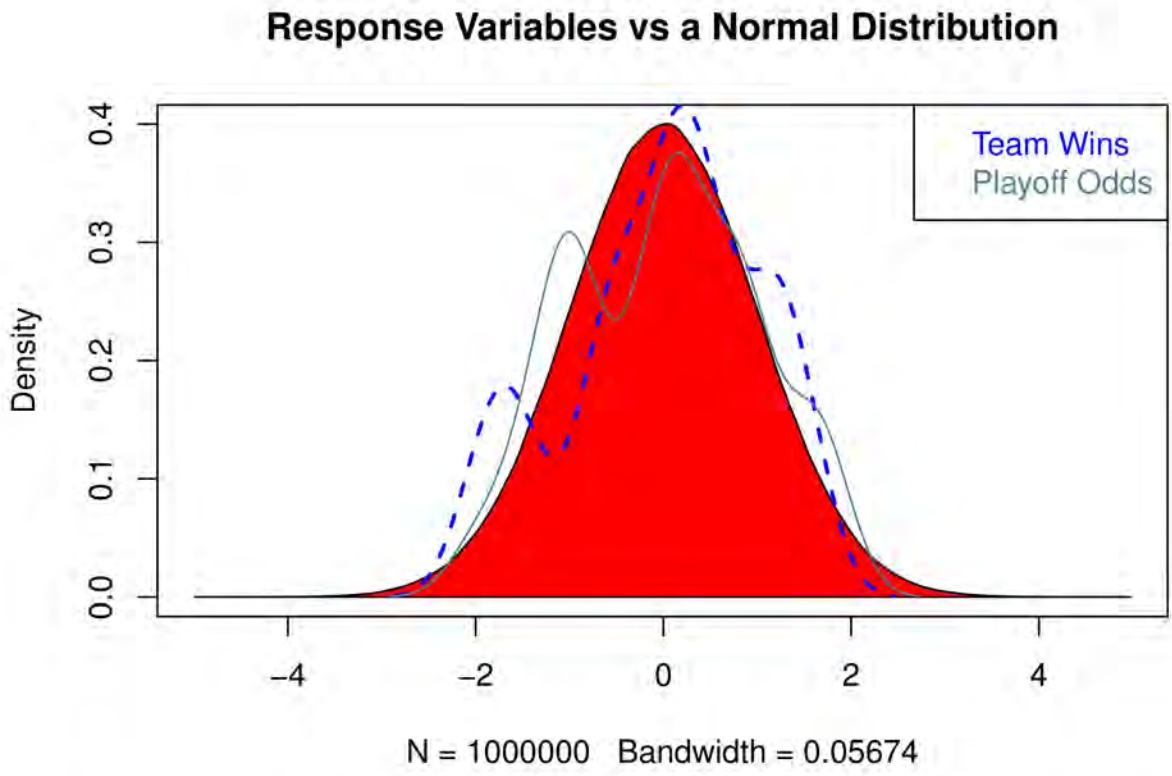
```
#R Code Con't

attach(Finaldive)

pairs(~teampoints+Dropback.EPA+Total.DVOA+Offense.DVOA+Success.Rate..SR.+
        Variance+Schedule+Schedule.Rank+D.Dropback.SR+
        Unadj..Total.VOA.Rank,
      pch = 16, col = "deepskyblue")
```




```
scatterplotMatrix(~teampoints+Dropback.EPA+Total.DVOA+Offense.DVOA+Success.Rate..SR.+
                  Variance+Schedule+Schedule.Rank+D.Dropback.SR+
                  Unadj..Total.VOA.Rank)
```




```
detach(Finaldive)

par(mfrow = c(1,1))
zwin <- scale(Finaldive$Actual.Wins)
zpoffs <- scale(Finaldive$Playoff.odds)
zbin <- scale(Finaldive$Made.Playoffs)
normalpoints <- rnorm(1000000)


plot(density(normalpoints), main ="Response Variables vs a Normal Distribution" )
polygon(density(normalpoints), col = "red")
lines(density(zwin), col="blue", lty = 2, lwd =2)
lines(density(zpoffs), col = "cadetblue4")
legend("topright", c("Team Wins", "Playoff Odds"),
text.col = c("blue", "cadetblue4"))
```



Let's first consider whether these variables are good predictors of whether a team will make the playoffs or not, determining if linear regression is our best fit.

Selecting Variables for feature engineering, Starting with a Correlation Matrix

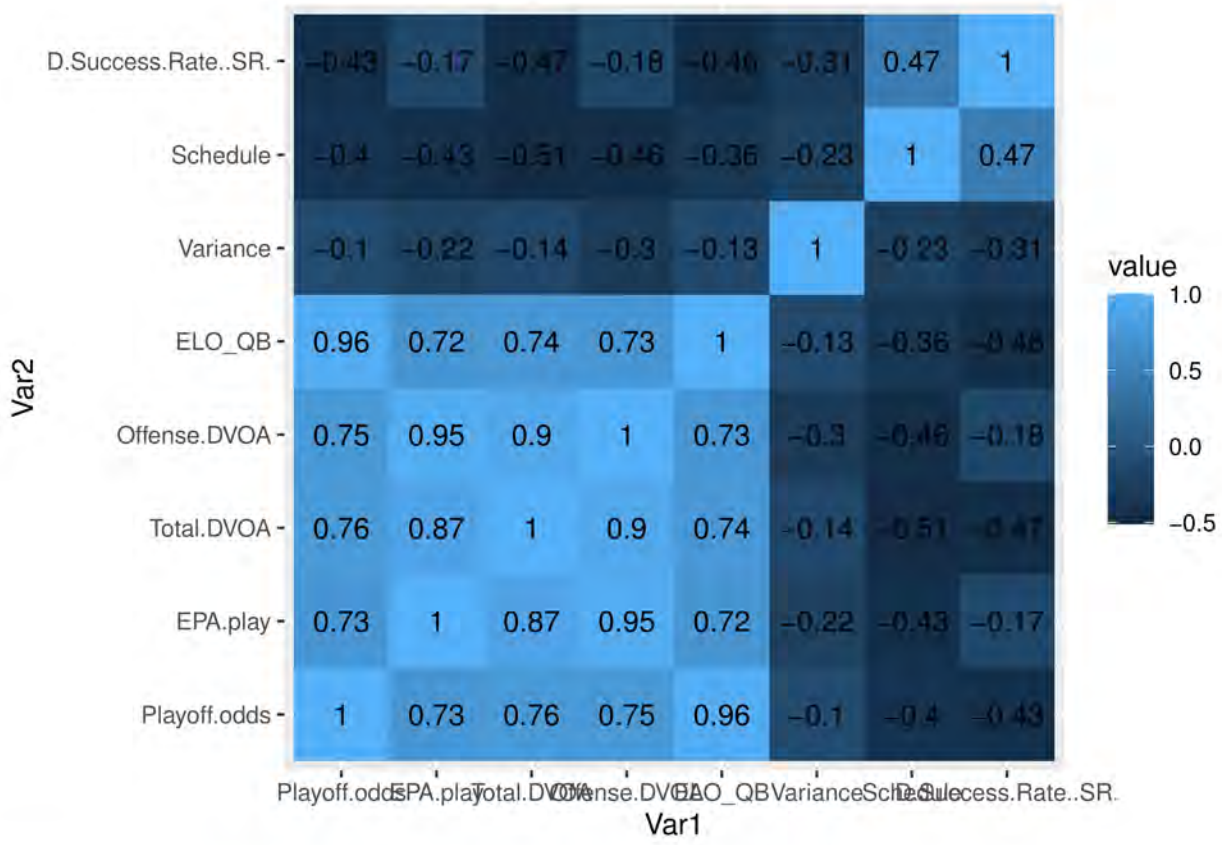
#121 - Playoff.odds; #93 - EPA per play; #56 - Total.DVOA; #62 - Offense.DVOA #112 - ELO QB Metric ; #76 - Variance ; #72 - Schedule; #78 - D.Success.Rate..SR.

```
#Code Con't

#Model 1
cor(Finaldive2[ , c(121,93,56,62,112,76,72,78) ], method = "pearson")
```

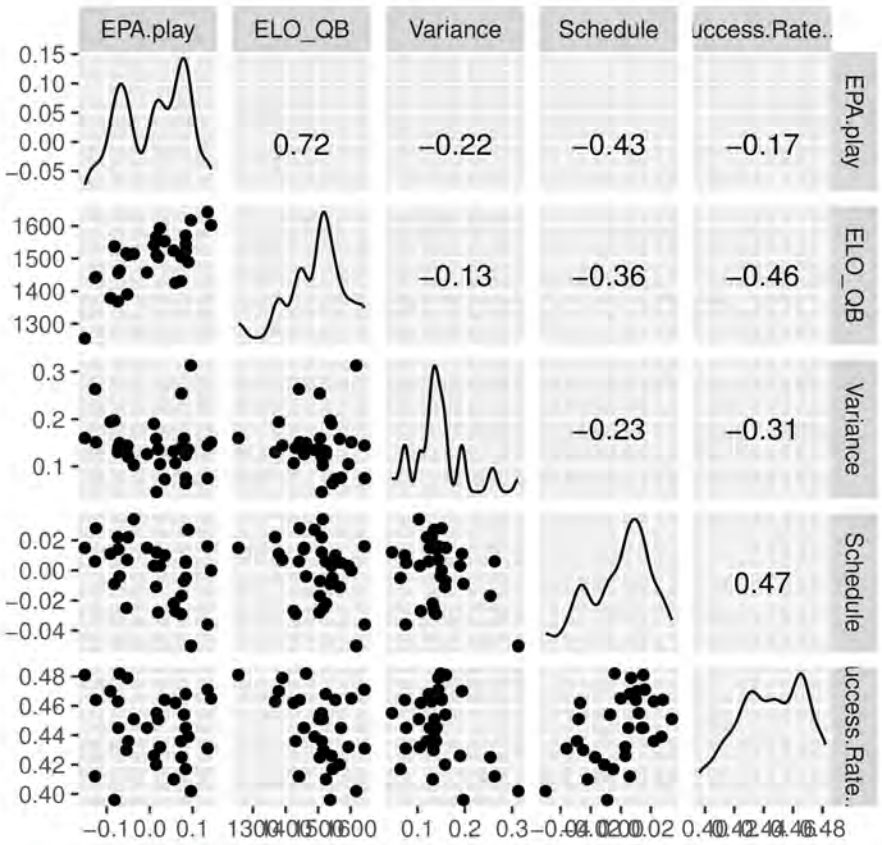
```
##                     Playoff.odds   EPA.play Total.DVOA Offense.DVOA     ELO_QB
## Playoff.odds          1.00000000  0.7281900  0.7631457    0.7527578  0.9600288
## EPA.play              0.72819003  1.0000000  0.8740376    0.9521638  0.7234144
## Total.DVOA            0.76314571  0.8740376  1.0000000    0.9049453  0.7396916
## Offense.DVOA          0.75275782  0.9521638  0.9049453    1.0000000  0.7285725
## ELO_QB                0.96002884  0.7234144  0.7396916    0.7285725  1.0000000
## Variance             -0.09690735 -0.2234071 -0.1350010   -0.2963135 -0.1284486
## Schedule             -0.40477253 -0.4284179 -0.5125962   -0.4626578 -0.3555486
## D.Success.Rate..SR.  -0.42926598 -0.1738325 -0.4695450   -0.1842658 -0.4611156
##                        Variance   Schedule D.Success.Rate..SR.
## Playoff.odds        -0.09690735 -0.4047725          -0.4292660
## EPA.play            -0.22340710 -0.4284179          -0.1738325
```

```
## Total.DVOA          -0.13500096 -0.5125962          -0.4695450
## Offense.DVOA        -0.29631353 -0.4626578          -0.1842658
## ELO_QB              -0.12844856 -0.3555486          -0.4611156
## Variance             1.00000000 -0.2318658          -0.3111972
## Schedule            -0.23186582  1.0000000           0.4735672
## D.Success.Rate..SR. -0.31119724  0.4735672           1.0000000
```

```
cormat<-round(cor(Finaldive2[ , c(121,93,56,62,112,76,72,78) ], method = "pearson"),2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2,
                                 fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value),
            color = "black", size = 4)
```




```
# Removing a couple variables
#93 - EPA per play;
#112 - ELO QB Metric ;
#76 - Variance ;
#72 - Schedule;
#78 - D.Success.Rate..SR.

ggscatmat(Finaldive2, columns=c(93,112,76,72,78))
```




```
##
## T.test to reject or accept null hypothesis
##
## Null hypothesis -  variables have no impact whether the team is going to make the playoff's
#or not...
## Alternative hypothesis - all the variables listed have an impact on the outcome.

options(scipen = 9999)

t.test(Finaldive2$EPA.play+Finaldive2$ELO_QB+Finaldive2$D.Success.Rate..SR.+
         Finaldive2$Variance, var.equal=TRUE, paired=FALSE)
```

```
##
##  One Sample t-test
##
## data:  Finaldive2$EPA.play + Finaldive2$ELO_QB + Finaldive2$D.Success.Rate..SR. + Finaldive2$Varianc
## t = 428.5, df = 593, p-value < 0.00000000000000022
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1490.894 1504.623
## sample estimates:
## mean of x
##  1497.758
```

```
#We accept the Alternative hypothesis.  Now let's see if this model fits a linear regression

m2 <- lm(Actual.Wins~EPA.play+ELO_QB+Point.Spread.Rating.QB+
           D.Success.Rate..SR.+Variance+Schedule, data = Finaldive2)
summary(m2)
```

```
##
## Call:
## lm(formula = Actual.Wins ~ EPA.play + ELO_QB + Point.Spread.Rating.QB +
##     D.Success.Rate..SR. + Variance + Schedule, data = Finaldive2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.35728 -0.86214  0.06765  1.16484  2.11416
##
## Coefficients:
##                           Estimate Std. Error t value             Pr(>|t|)
## (Intercept)             24.072086   2.650620   9.082 <0.0000000000000002 ***
## EPA.play                18.307782   1.130548  16.194 <0.0000000000000002 ***
## ELO_QB                   0.001869   0.001085   1.723               0.0855 .
## Point.Spread.Rating.QB   0.378831   0.034915  10.850 <0.0000000000000002 ***
## D.Success.Rate..SR.    -44.407444   3.106573 -14.295 <0.0000000000000002 ***
## Variance                -2.293857   1.077724  -2.128               0.0337 *
## Schedule                 6.786185   3.355125   2.023               0.0436 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.247 on 587 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.8223
## F-statistic: 458.2 on 6 and 587 DF,  p-value: < 0.00000000000000022
```

So our first stab at figuring out which variables are significant predictors for increasing Team Wins (with basic logic, more wins increases chances of getting to the playoffs - in our CAR example we need to increase our Win Total by 4 to secure the 7th Seed). We will also test this model using logistic regression and a binary classifier.

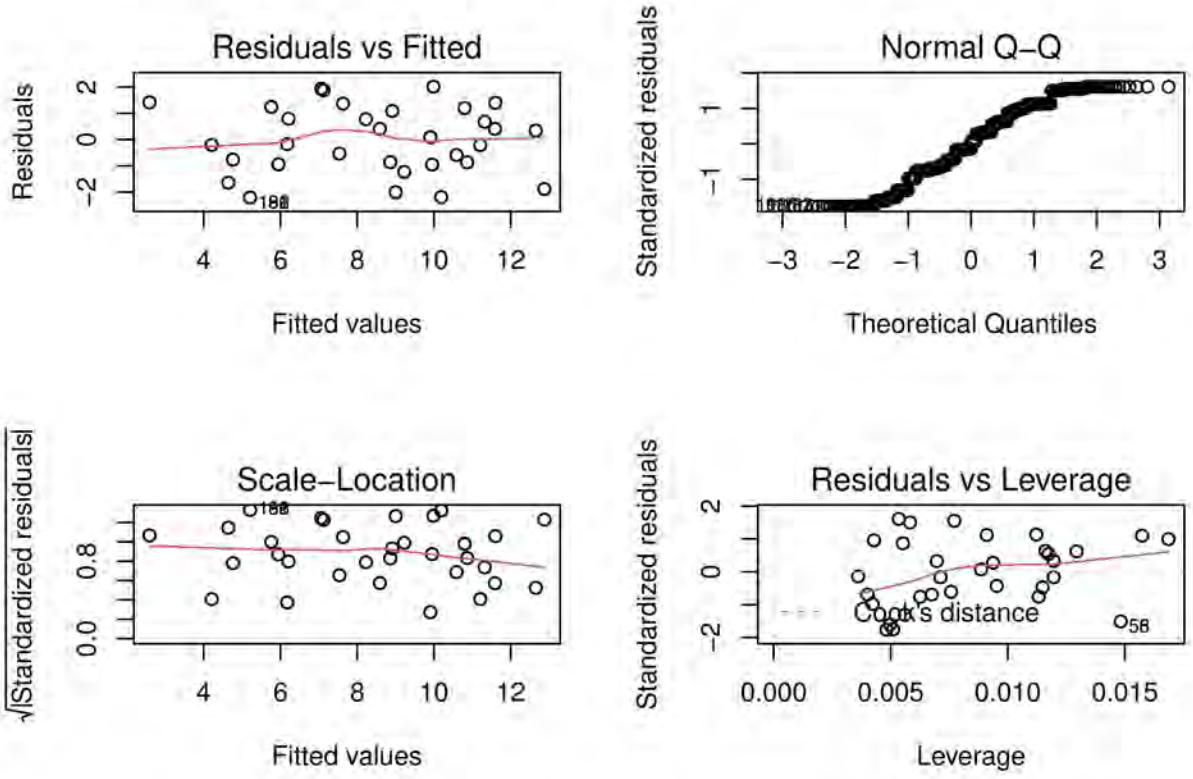
We can see that our model is pretty good, evidence by the Multiple R-squared value of .8083 & a p-value of less than .05. Based on the regression output let's take out ELO_QB & Variance and run it again & then view our diagnostic plots

```
m2.1 <- lm(Actual.Wins~EPA.play+Point.Spread.Rating.QB+
           D.Success.Rate..SR.+Schedule, data = Finaldive2)
summary(m2.1)
```

```
##
## Call:
## lm(formula = Actual.Wins ~ EPA.play + Point.Spread.Rating.QB +
##     D.Success.Rate..SR. + Schedule, data = Finaldive2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20592 -0.94380 -0.04087  1.08679  2.01313
##
```

```
## Coefficients:
##                          Estimate Std. Error t value             Pr(>|t|)
## (Intercept)              27.13735    1.12020  24.226 < 0.0000000000000002 ***
## EPA.play                 19.81829    0.97080  20.414 < 0.0000000000000002 ***
## Point.Spread.Rating.QB    0.40011    0.03323  12.039 < 0.0000000000000002 ***
## D.Success.Rate..SR.     -45.95876    2.51954 -18.241 < 0.0000000000000002 ***
## Schedule                  9.72095    3.21759   3.021              0.00263 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.254 on 589 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8203
## F-statistic: 677.7 on 4 and 589 DF,  p-value: < 0.00000000000000022
```

```
par(mfrow = c(2,2))
plot(m2.1)
```



This model checks out. Now let's just see if it's grounded in reality. Using '21 data let's see if our model predicts whether CAR would have made the playoffs.

```
#current CAR stats!
predict(m2.1, list(EPA.play = -.127, ELO_QB = 1441, Point.Spread.Rating.QB =0.5,  D.Success.Rate..SR. =
                   Variance =.263, Schedule = .006))
```

```
##        1
## 5.943799
```

```
#   5.94
##So No, 6 Wins does not get us to the playoffs.  Let's test it against our Playoff
##Binary classifier.

m2.1log <- glm(Made.Playoffs~EPA.play+Point.Spread.Rating.QB+
          D.Success.Rate..SR.+Schedule, data = Finaldive, family = binomial)

summary(m2.1log)
```

```
##
## Call:
## glm(formula = Made.Playoffs ~ EPA.play + Point.Spread.Rating.QB +
##     D.Success.Rate..SR. + Schedule, family = binomial, data = Finaldive)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.49542  -0.52629  -0.09093   0.51932   1.85186
##
## Coefficients:
##                          Estimate Std. Error z value      Pr(>|z|)
## (Intercept)              27.31026    5.42090   5.038 0.00000047053 ***
## EPA.play                 28.99604    4.80678   6.032 0.00000000162 ***
## Point.Spread.Rating.QB    0.09525    0.10785   0.883        0.3771
## D.Success.Rate..SR.     -64.67991   12.27621  -5.269 0.00000013738 ***
## Schedule                 28.78512   13.75487   2.093        0.0364 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 344.64  on 249  degrees of freedom
## Residual deviance: 185.68  on 245  degrees of freedom
## AIC: 195.68
##
## Number of Fisher Scoring iterations: 6
```

```
predict(m2.1log, list(EPA.play = -.127, ELO_QB = 1441, Point.Spread.Rating.QB =0.5,  D.Success.Rate..SR
                Variance =.263, Schedule = .006))
```

```
##         1
## -2.800026
```

```
# 0 = DID NOT MAKE PLAYOFFS / 1 = MADE PLAYOFFS
```

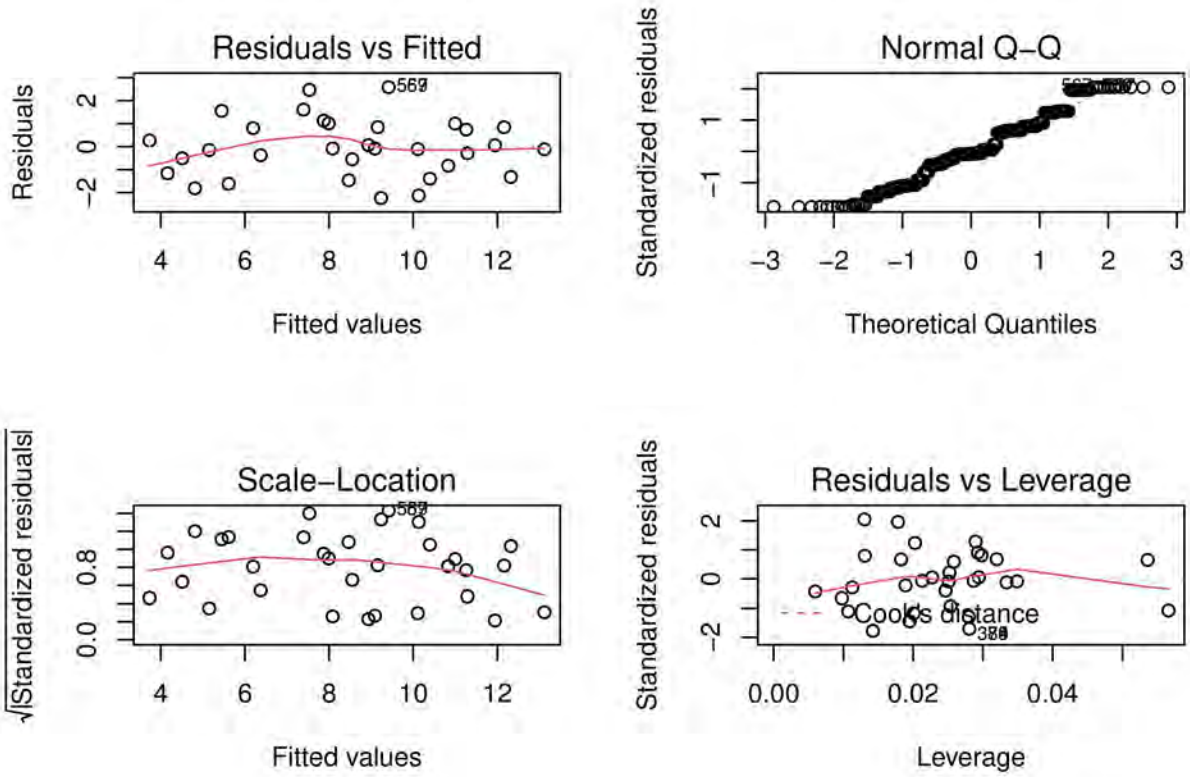
This model checks out. Seeing a value of -2.8 where a value of 0 = Didn't make Playoffs, and values of 1 = Made Playoffs, indicates that the current build of the team didn't meet the threshold to make the playoffs. And using the "Win Model" version of it, the team falls about 3 Wins short of the total needed to get in.

```
#Model #3
#Top 8 QB Play = Point.Spread.Rating.QB.1 = 6, ELO_QB = 1550, EPA.play =0.74, D.Dropback.EPA = .02,
#Strength.of.Schedule.To.Date = 24, Variance = .105
```

```r
#options(scipen = 999)
m3 <- lm(Actual.Wins~EPA.play+Point.Spread.Rating.QB+
           D.Dropback.EPA+Variance+Strength.of.Schedule.To.Date, data = Finaldive)
summary(m3)
```

```
##
## Call:
## lm(formula = Actual.Wins ~ EPA.play + Point.Spread.Rating.QB +
##     D.Dropback.EPA + Variance + Strength.of.Schedule.To.Date,
##     data = Finaldive)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23663 -0.83273 -0.08952  0.84450  2.58376
##
## Coefficients:
##                                Estimate Std. Error t value             Pr(>|t|)
## (Intercept)                    8.355651   0.364799  22.905 < 0.0000000000000002
## EPA.play                      16.947092   1.501264  11.289 < 0.0000000000000002
## Point.Spread.Rating.QB         0.294861   0.049061   6.010       0.00000000671
## D.Dropback.EPA               -11.468091   1.068481 -10.733 < 0.0000000000000002
## Variance                      -3.492122   1.679718  -2.079               0.0387
## Strength.of.Schedule.To.Date   0.002361   0.010197   0.232               0.8171
##
## (Intercept)                  ***
## EPA.play                     ***
## Point.Spread.Rating.QB       ***
## D.Dropback.EPA               ***
## Variance                     *
## Strength.of.Schedule.To.Date
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.268 on 244 degrees of freedom
## Multiple R-squared:  0.8032, Adjusted R-squared:  0.7991
## F-statistic: 199.1 on 5 and 244 DF,  p-value: < 0.00000000000000022
```

```r
par(mfrow = c(2,2))
plot(m3)
```




```
#Hypothesis Testing
predict(m3, list(Point.Spread.Rating.QB = 6, ELO_QB = 1550, EPA.play =0.74, D.Dropback.EPA = .02,
                 Strength.of.Schedule.To.Date = 24, Variance = .105))
```

```
##       1
## 22.1263
```

```
m3log <- glm(Made.Playoffs~EPA.play+Point.Spread.Rating.QB+
          D.Dropback.EPA+Variance+Strength.of.Schedule.To.Date, data = Finaldive, family = binomial)

summary(m3log)
```

```
##
## Call:
## glm(formula = Made.Playoffs ~ EPA.play + Point.Spread.Rating.QB +
##     D.Dropback.EPA + Variance + Strength.of.Schedule.To.Date,
##     family = binomial, data = Finaldive)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.70026  -0.41390  -0.04356   0.43072   1.89736
##
## Coefficients:
##                                 Estimate Std. Error z value      Pr(>|z|)
## (Intercept)                     0.358906   0.936616   0.383       0.70158
```

```
## EPA.play                      29.810175   4.962931   6.007 0.00000000189 ***
## Point.Spread.Rating.QB         0.003344   0.115041   0.029       0.97681
## D.Dropback.EPA               -16.372882   3.300005  -4.961 0.00000069961 ***
## Variance                      16.906079   5.897774   2.867       0.00415 **
## Strength.of.Schedule.To.Date  -0.156930   0.033900  -4.629 0.00000366974 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 344.64  on 249  degrees of freedom
## Residual deviance: 172.89  on 244  degrees of freedom
## AIC: 184.89
##
## Number of Fisher Scoring iterations: 6
```

```
# Split train and test data - 70% train and 30% Test.
split_dummy <- sample(c(rep(0,0.7*nrow(Finaldive)),
                        rep(1,0.3*nrow(Finaldive))))
table(split_dummy)
```

```
## split_dummy
##   0   1
## 175  75
```

```
Finaldive[,'Made.Playoffs'] <- as.factor(Finaldive[,'Made.Playoffs'])

# Split input file into two sets
Finaldive_Train <- Finaldive[split_dummy==0,]
Finaldive_Test <- Finaldive[split_dummy==1,]
# Prepare playoff odd variable. Use 538 playoff odd variable to populate.
# If a team has probability of more than 50%, the odd to make playoff is 1
# else set it as 0.
Playoffs.binary <- ifelse(Finaldive_Train$Playoff.odds >0.5,1,0)
Playoffs.binary
```

```
##   [1] 1 1 1 0 0 0 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 1 0 0 1 0 0 1 1 0 0 0 0 0 1 1
##  [38] 1 0 1 0 0 1 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 1
##  [75] 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0
## [112] 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 1 1 1 1 0 0
## [149] 0 0 1 0 1 0 0 0 1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 0 1 0
```

```
attach(Finaldive_Train)
# Prepare model
m3log.2 <- glm(Made.Playoffs~EPA.play+Point.Spread.Rating.QB+
      D.Dropback.EPA+Variance+Strength.of.Schedule.To.Date, data = Finaldive_Train, family = binomial
      (link = "logit"))

summary(m3log.2)
```

```
##
## Call:
```

```
## glm(formula = Made.Playoffs ~ EPA.play + Point.Spread.Rating.QB +
##     D.Dropback.EPA + Variance + Strength.of.Schedule.To.Date,
##     family = binomial(link = "logit"), data = Finaldive_Train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.57342  -0.36279  -0.05085   0.30441   2.12164
##
## Coefficients:
##                               Estimate Std. Error z value    Pr(>|z|)
## (Intercept)                   -0.45104    1.17068  -0.385     0.70003
## EPA.play                      31.34172    6.27335   4.996 0.000000585 ***
## Point.Spread.Rating.QB         0.04882    0.14218   0.343     0.73130
## D.Dropback.EPA               -21.11690    5.10115  -4.140 0.000034786 ***
## Variance                      19.91436    7.52426   2.647     0.00813 **
## Strength.of.Schedule.To.Date  -0.15030    0.04610  -3.261     0.00111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 240.08  on 174  degrees of freedom
## Residual deviance: 107.64  on 169  degrees of freedom
## AIC: 119.64
##
## Number of Fisher Scoring iterations: 6
```

```
m2log.3 <- glm(Made.Playoffs~EPA.play+ELO_QB+Point.Spread.Rating.QB+
                 D.Success.Rate..SR.+Schedule, data = Finaldive, family = binomial
          (link = "logit"))
summary(m2log.3)
```

```
##
## Call:
## glm(formula = Made.Playoffs ~ EPA.play + ELO_QB + Point.Spread.Rating.QB +
##     D.Success.Rate..SR. + Schedule, family = binomial(link = "logit"),
##     data = Finaldive)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.57135  -0.39148  -0.02567   0.43315   2.11952
##
## Coefficients:
##                            Estimate  Std. Error z value     Pr(>|z|)
## (Intercept)              123.151486   22.563562   5.458 0.0000000482 ***
## EPA.play                  61.285086   10.781739   5.684 0.0000000131 ***
## ELO_QB                    -0.038575    0.007601  -5.075 0.0000003880 ***
## Point.Spread.Rating.QB     0.370654    0.153641   2.412     0.015845 *
## D.Success.Rate..SR.     -153.053944   28.379549  -5.393 0.0000000692 ***
## Schedule                  93.403230   25.326630   3.688     0.000226 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 344.64  on 249  degrees of freedom
## Residual deviance: 142.57  on 244  degrees of freedom
## AIC: 154.57
##
## Number of Fisher Scoring iterations: 7
```

```
# Run model with Test data
P<-predict(m2log.3,newdata=Finaldive_Test, type="response")

# Get odds of playoff vs not playoffs
table_mat<-table(Finaldive_Test$Made.Playoffs, P>0.45)
table_mat
```

```
##
##     FALSE TRUE
##   0    35    3
##   1     9   28
```

```
accuracy_test <- sum(diag(table_mat))/sum(table_mat)
# Get accuracy percentage
# 0.84
accuracy_test
```

```
## [1] 0.84
```

```
p.Made.Playoffs <-round(P)

# Following code is to test how much model came close to our Test data
#
# Use input file to output our playoff prediction
p.Made.Playoffs <- round(predict(m2log.3,Finaldive, type="response"))
data <- data.frame(nflid = Finaldive$team, Predict.Playoffs = p.Made.Playoffs)
write.csv(data,"Playoff Model 2.3.csv",row.names=TRUE)
```

This model correctly classified 28/32 teams! 88% Accuracy. It correctly identified all 14 teams that didn't make the playoffs. And the false negatives and false positives were all "bubble teams". IND had they not lost a couple of fluke games were a shoe-in for the playoffs. PHI and PIT were both the 7th seeds in their respective divisions and got blown out in dramatic fashion in wildcard weekend. And by all accounts really The LA Chargers were better than their record & were one play away from getting to the playoffs in their week 18 game.

```
M2log.3results <- read.csv("./data_2/Playoff Model 2.3.csv")
knitr::kable(
  M2log.3results[ , 2:7],
  caption = "Model 2 Prediction Results"
)
```

Table 2: Model 2 Prediction Results

nflid	Predict.Playoffs	Positive	False.Positive	Negative	False.Negative
ARZ	1	1	NA	NA	NA
ATL	0	NA	NA	1	NA
BLT	0	NA	NA	1	NA
BUF	1	1	NA	NA	NA
CAR	0	NA	NA	1	NA
CHI	0	NA	NA	1	NA
CIN	1	1	NA	NA	NA
CLV	0	NA	NA	1	NA
DAL	1	1	NA	NA	NA
DEN	0	NA	NA	1	NA
DET	0	NA	NA	1	NA
GB	1	1	NA	NA	NA
HST	0	NA	NA	1	NA
IND	1	NA	1	NA	NA
JAX	0	NA	NA	1	NA
KC	1	1	NA	NA	NA
LA	1	1	NA	NA	NA
LAC	1	NA	1	NA	NA
LV	1	1	NA	NA	NA
MIA	0	NA	NA	1	NA
MIN	0	NA	NA	1	NA
NE	1	1	NA	NA	NA
NO	0	NA	NA	1	NA
NYG	0	NA	NA	1	NA
NYJ	0	NA	NA	1	NA
PHI	0	NA	NA	NA	1
PIT	0	NA	NA	NA	1
SEA	0	NA	NA	1	NA
SF	1	1	NA	NA	NA
TB	1	1	NA	NA	NA
TEN	1	1	NA	NA	NA
WAS	0	NA	NA	1	NA

Conclusions and Future Work on this Project

A preview of the next progression of this project: Using some of these same variables can we project which fantasy football prospects are better bets to achieve value relative to their draft capital? For instance, Hunter Renfrow of the Las Vegas Raiders & Cooper Kupp of the LA Rams this past year...

Which positional groups represent value come draft time? Which are more likely to bust or overachieve?

```
ADP2 <- read.csv("./data_2/FinalDF_2.26.22-3.csv")
ADP2 <- ADP2[c(4,7,10:11)]
names(ADP2)

FFdive <- merge(x=FFdive, y=ADP2, by="player", all = FALSE, all.x = TRUE)

FFdive$ADP_bin <- rep(0,250)
FFdive$ADP_bin[FFdive$Finished_As_Delta >= 1] <- 1
FFdive$ADP_bin[FFdive$Finished_As_Delta <= 0] <- 0
```

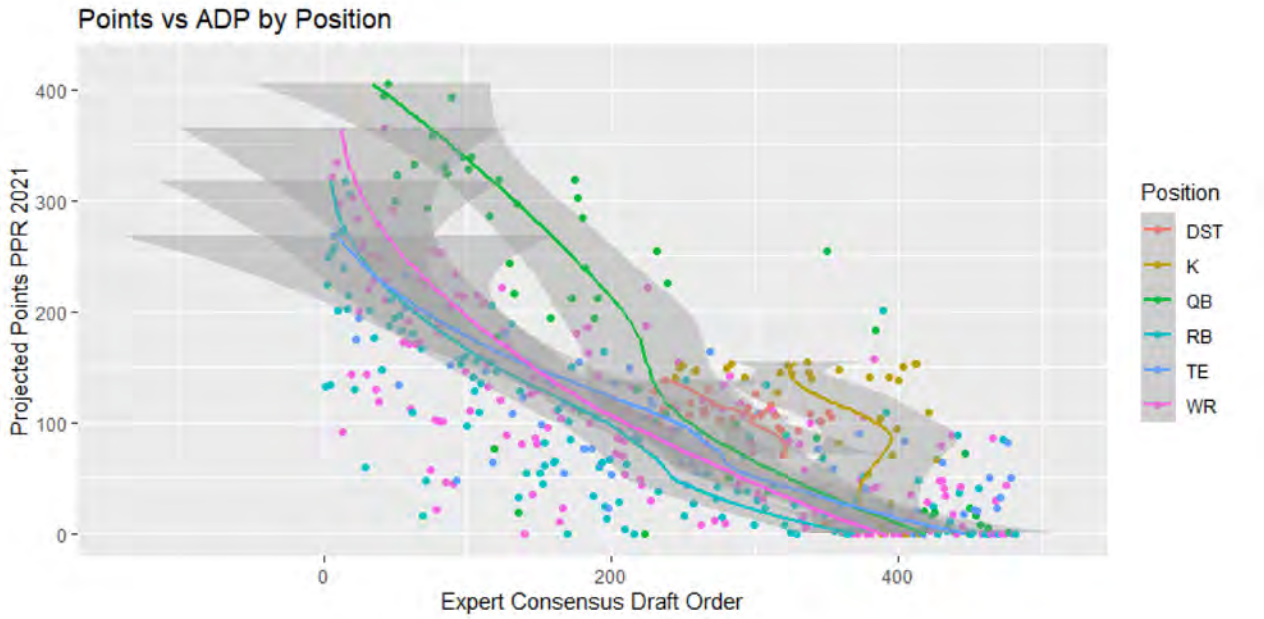
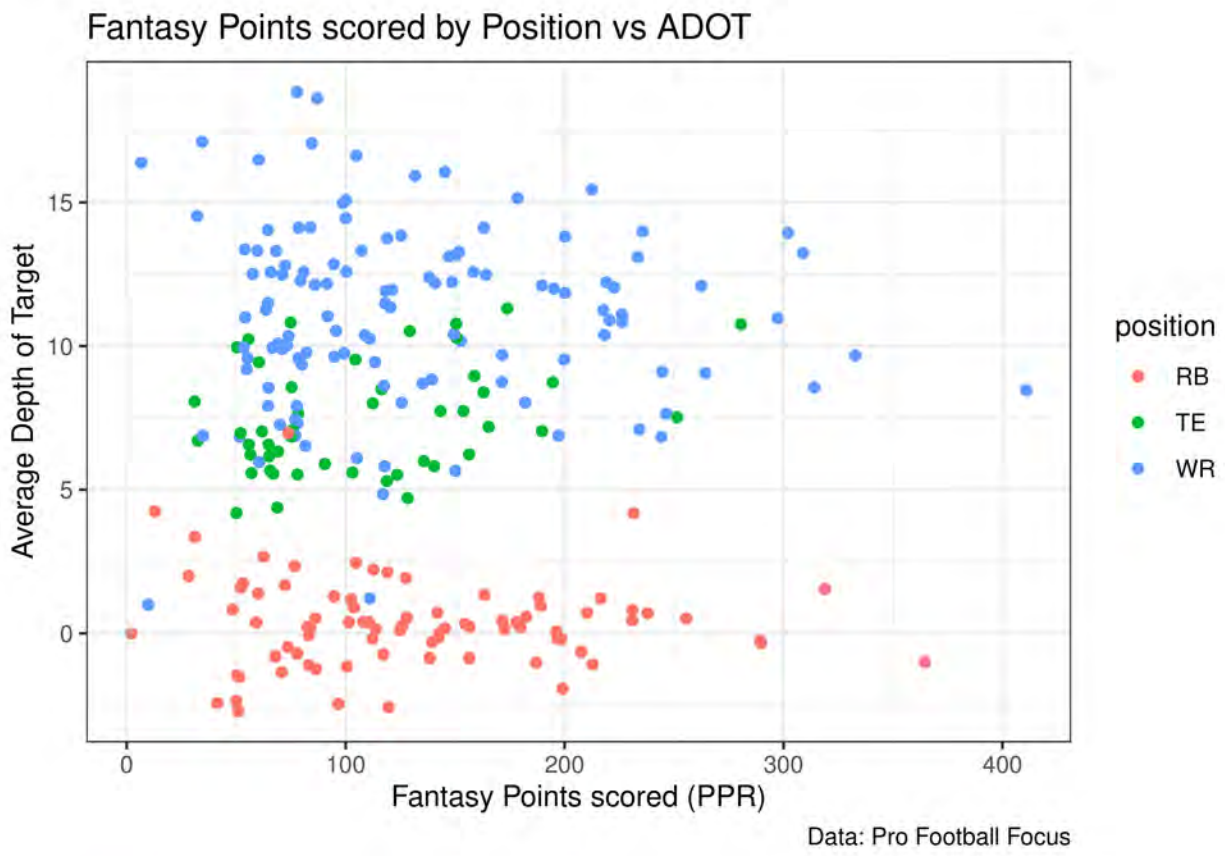
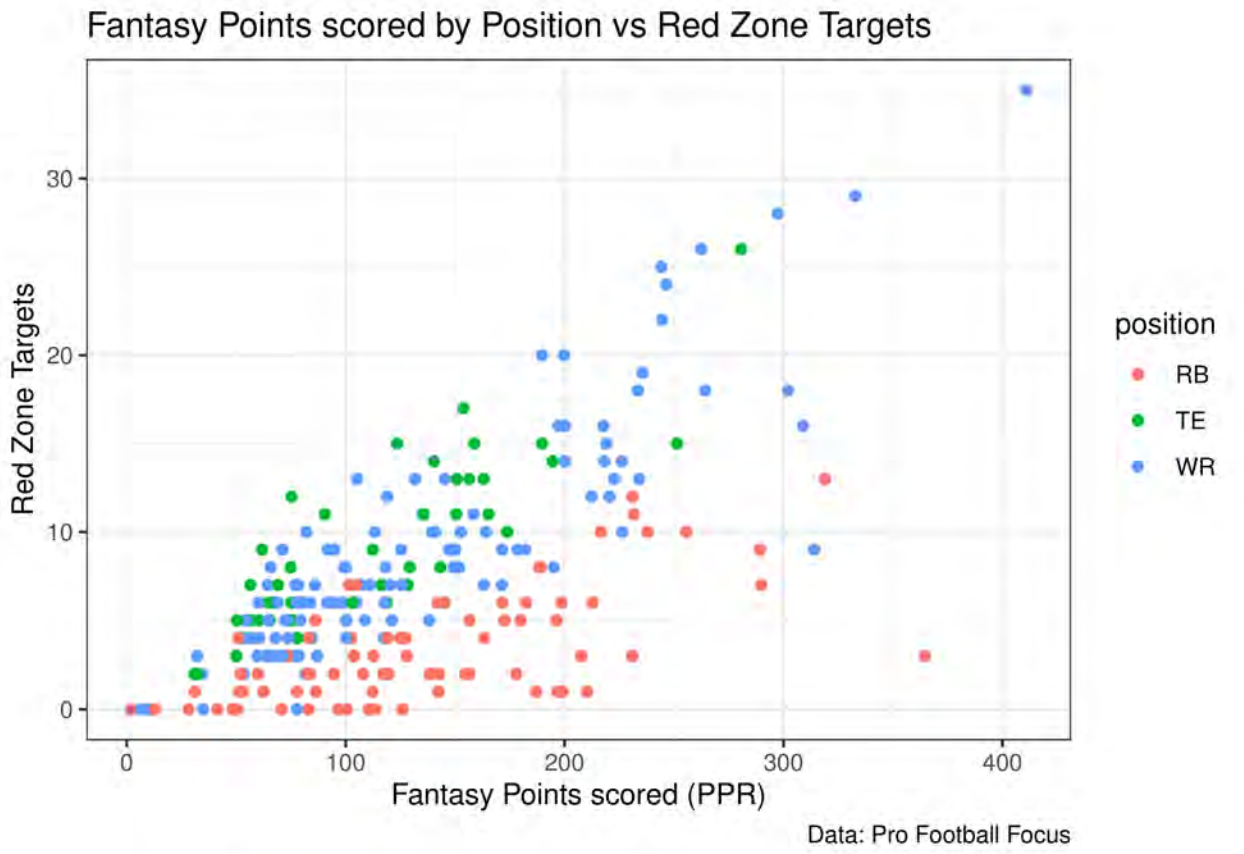


Figure 3: Points Scored vs ADP by Position

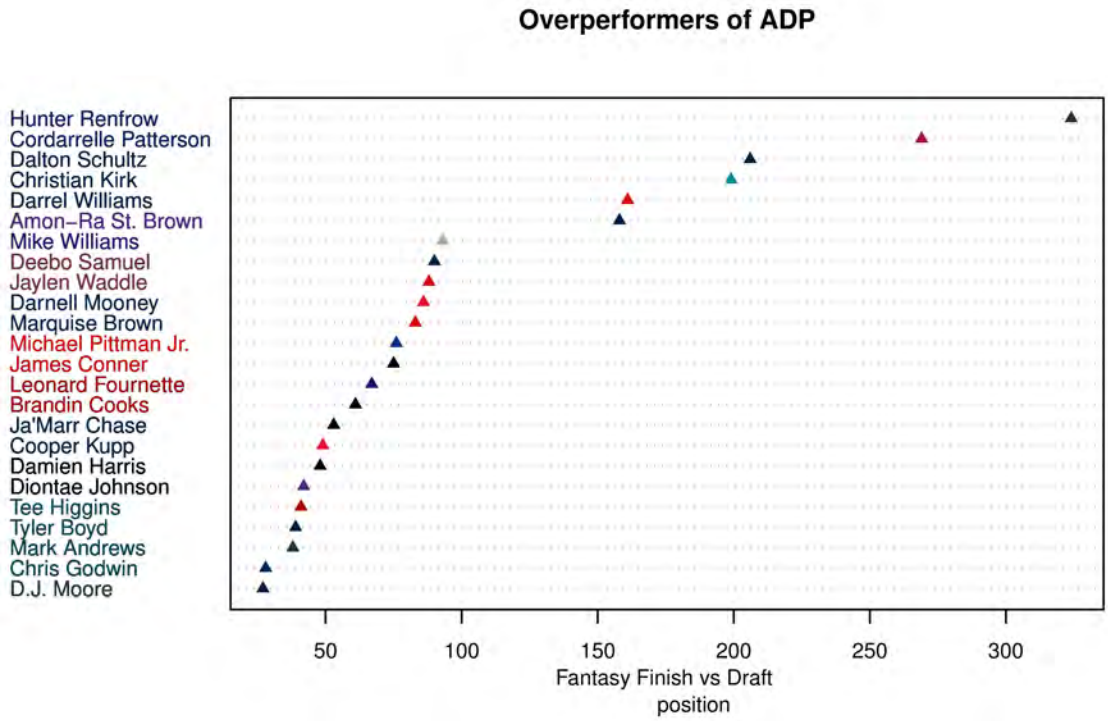
```
table(FFdive$ADP_bin)
# 0   1
#117 133

ggplot(data=FFdive) +
  geom_point(mapping = aes(x=fantasyPts, y=depth,
                           color=position)) +
  labs(x = "Fantasy Points scored (PPR)",
       y = "Average Depth of Target",
       title = "Fantasy Points scored by Position vs ADOT",
       caption = "Data: Pro Football Focus") +
  theme_bw()
```



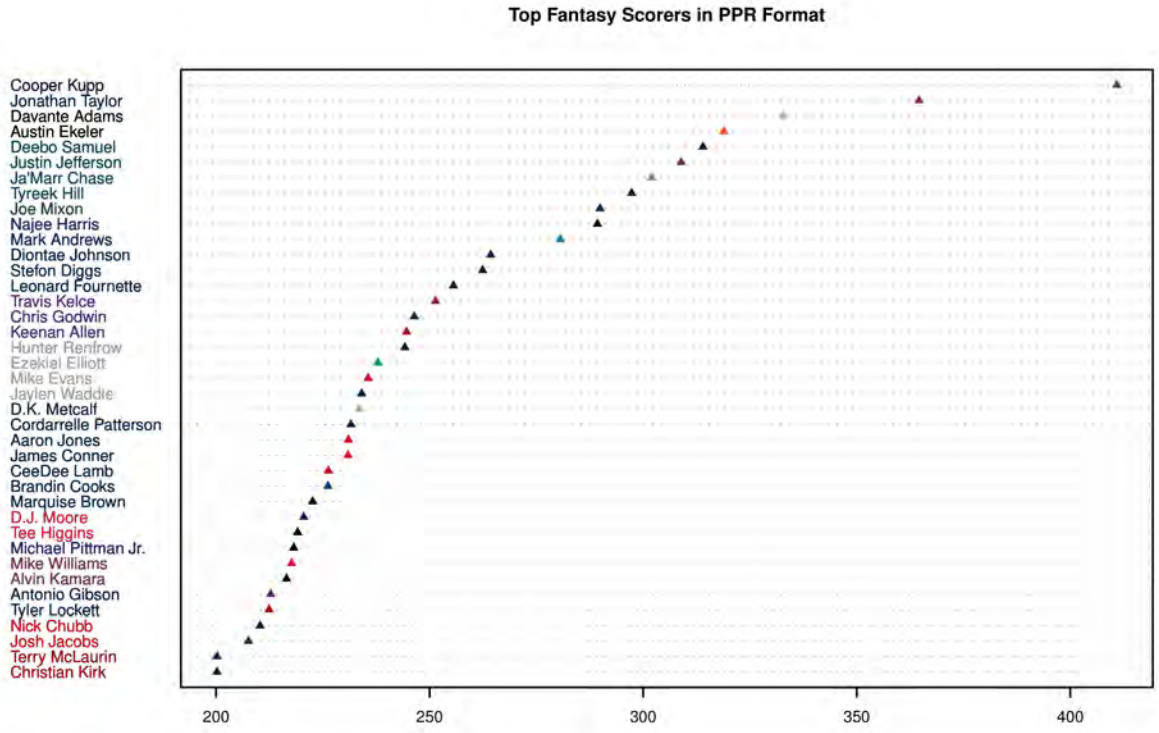

```
ggplot(data=FFdive) +
  geom_point(mapping = aes(x=fantasyPts, y=rzRecRec,
                           color=position)) +
  labs(x = "Fantasy Points scored (PPR)",
       y = "Red Zone Targets",
       title = "Fantasy Points scored by Position vs Red Zone Targets",
       caption = "Data: Pro Football Focus") +
  theme_bw()
```




```
delta <- FFdive[, c(1:7,28,29)]

delta$Finished_As_Delta <- as.numeric(delta$Finished_As_Delta)

## Warning: NAs introduced by coercion

delta <- delta[order(delta$Finished_As_Delta, decreasing = TRUE), ]


delta2 <- subset(delta, fantasyPts >= 180 & Finished_As_Delta >=25)
(delta2 <- delta2[order(delta2$Finished_As_Delta),])


dotchart(delta2$Finished_As_Delta, labels = delta2$player, cex = .7,
         main = "Overperformers of ADP", xlab = "Fantasy Finish vs Draft
        position", pch = 17, color = Finaldive2$team_color)
```




```
fpts <- subset(delta, fantasyPts > 200)
(fpts <- fpts[order(fpts$fantasyPts),])
dotchart(fpts$fantasyPts, labels = fpts$player, cex = .5,
         main = "Top Fantasy Scorers in PPR Format", pch = 17,
         color = Finaldive2$team_color)
```



A longer look into a player like Hunter Renfrow is warranted; A player drafted at ADP #342 but finishing at #18 (at all positions, scoring 244pts in a PPR Format!). This is the proverbial league-winner!

So what variables should be considered as predictors of Fantasy production? Let's look at what I believe our response variable should be first, "Fantasy Points", I.e. points scored throughout the '21 season in a PPR format. Also I want to see if it correlates with QB play, Success Rate of these offenses, Depth of Target, Recieving Targets, Points Per Touch for starters.

```
#Interesting look at summary stats for players based on where they were drafted & whether or not they
#exceeded their ADP value & how positional value interacts here...

FFL <- aggregate(fantasyPts~ADP_bin+position, data = fpts, FUN = summary)
(FFLtier <- aggregate(fantasyPts~ADPtier+ADP_bin+position, data = fpts, FUN = summary))
```

```
##                ADPtier ADP_bin position fantasyPts.Min. fantasyPts.1st Qu.
## 1  Consensus Top Pick       0       RB         210.300            212.800
## 2          Second Tier       0       RB         207.600            207.600
## 3  Consensus Top Pick       1       RB         289.300            289.750
## 4  Middle Rounds Pick       1       RB         255.600            255.600
## 5         Value Rounds       1       RB         230.900            230.900
## 6    Mostly Undrafted       1       RB         231.600            231.600
## 7  Consensus Top Pick       0       TE         251.400            251.400
## 8          Second Tier       1       TE         280.600            280.600
## 9  Consensus Top Pick       0       WR         233.500            233.500
## 10         Second Tier       0       WR         200.200            200.200
```

```
## 11 Consensus Top Pick       1       WR         262.400            288.575
## 12         Second Tier       1       WR         226.300            235.600
## 13 Middle Rounds Pick       1       WR         212.400            219.100
## 14        Value Rounds       1       WR         217.700            219.300
## 15   Mostly Undrafted       1       WR         200.200            211.200
##    fantasyPts.Median fantasyPts.Mean fantasyPts.3rd Qu. fantasyPts.Max.
## 1            216.500         221.700            231.000         237.900
## 2            207.600         207.600            207.600         207.600
## 3            304.400         315.675            330.325         364.600
## 4            255.600         255.600            255.600         255.600
## 5            230.900         230.900            230.900         230.900
## 6            231.600         231.600            231.600         231.600
## 7            251.400         251.400            251.400         251.400
## 8            280.600         280.600            280.600         280.600
## 9            233.500         233.500            233.500         233.500
## 10           200.200         200.200            200.200         200.200
## 11           303.100         300.350            314.875         332.800
## 12           244.600         272.760            246.400         410.900
## 13           220.500         243.660            264.300         302.000
## 14           224.400         238.800            232.125         314.000
## 15           222.200         222.200            233.200         244.200
```

```
#maxdata <- aggregate(score~gender+id+test, data=longdata, FUN = max)
```

```
ADPbin <- glm(ADP_bin~ADP+Point.Spread.Rating.QB+Success.Rate..SR.+recTarg+depth+ptsPerTouch,
              data = FFdive, family = binomial)
summary(ADPbin)
```

```
##
## Call:
## glm(formula = ADP_bin ~ ADP + Point.Spread.Rating.QB + Success.Rate..SR. +
##     recTarg + depth + ptsPerTouch, family = binomial, data = FFdive)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.33105  -0.35332   0.08442   0.37924   2.36085
##
## Coefficients:
##                          Estimate Std. Error z value        Pr(>|z|)
## (Intercept)           -12.432437   4.489973  -2.769         0.00562 **
## ADP                     0.031524   0.004841   6.512 0.0000000000739 ***
## Point.Spread.Rating.QB  0.094024   0.126435   0.744         0.45708
## Success.Rate..SR.      13.343946   9.960023   1.340         0.18033
## recTarg                 0.073243   0.012944   5.658 0.0000000152792 ***
## depth                  -0.205833   0.109786  -1.875         0.06081 .
## ptsPerTouch            -0.139534   0.652831  -0.214         0.83075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 269.91  on 206  degrees of freedom
```

```
## Residual deviance: 128.64  on 200  degrees of freedom
##   (43 observations deleted due to missingness)
## AIC: 142.64
##
## Number of Fisher Scoring iterations: 6

ADPbin2 <- lm(fantasyPts~ADP+Point.Spread.Rating.QB+Success.Rate..SR.+recTarg+depth+ptsPerTouch,
              data = FFdive)
summary(ADPbin2)


##
## Call:
## lm(formula = fantasyPts ~ ADP + Point.Spread.Rating.QB + Success.Rate..SR. +
##     recTarg + depth + ptsPerTouch, data = FFdive)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -74.974 -26.597  -4.402  19.455 204.573
##
## Coefficients:
##                         Estimate Std. Error t value             Pr(>|t|)
## (Intercept)             16.02372   49.61440   0.323             0.747059
## ADP                     -0.10081    0.02655  -3.796             0.000195 ***
## Point.Spread.Rating.QB   2.91564    1.60916   1.812             0.071500 .
## Success.Rate..SR.      124.34122  118.73622   1.047             0.296269
## recTarg                  1.71391    0.09327  18.376 < 0.0000000000000002 ***
## depth                   -5.20426    1.19018  -4.373            0.0000197 ***
## ptsPerTouch              0.82103    7.30924   0.112             0.910676
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.54 on 200 degrees of freedom
##   (43 observations deleted due to missingness)
## Multiple R-squared:  0.7364, Adjusted R-squared:  0.7285
## F-statistic: 93.11 on 6 and 200 DF,  p-value: < 0.00000000000000022
```

"' In conclusion, there's quite a bit more to do on the latter half of this project but we've charted quite a bit of ground for the purposes of this assignment so I'm going to conclude this exploration here. Updates to come!