THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF STATISTICS

A Qualitative Approach to Predicting the Success of NFL Draftees

DREW INSLEY
SPRING 2022

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Statistics
with honors in Statistics

Reviewed and approved* by the following:

Andrew Wiesner
Associate Teaching Professor of Statistics
Thesis Supervisor

Matthew Beckman
Assistant Research Professor of Statistics
Honors Adviser

*Electronic approvals are on file in the Schreyer Honors College.

Abstract

As NFL teams continue to employ analytics to evaluate prospects, it is important to consider a holistic approach. Quantitative data have been the driving force behind the decision to draft certain players, as teams look at metrics like a player's 40-yard dash, vertical jump, and bench press. This report focuses on the use of qualitative data to evaluate players based on the strengths and weaknesses in their scouting reports for four major positions – Quarterback, Wide Receiver, Cornerback, and Defensive End. By employing a k nearest neighbors classifier, a classification tree, and a logistic regression, this study determines that the strengths in a player's scouting report is the most important factor in a player's chance of receiving a second contract with their draft team.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Dr. Andy Wiesner and Dr. Matthew Beckman for their support and assistance in the reasearch and analysis that went into this thesis. I would also like to thank my family and friends for their support during the process of writing this thesis.

Chapter 1

Introduction

1.1 The Quantitative Approach

The use of data science is swarming the National Football League as franchises use quantitative metrics to maximize their team's chance of success. Whether it be a prospect's speed at the combine, a player's chance of injury throughout the year, or the probability of converting on a 4^{th} and inches, the NFL is obsessed with the numbers game. This makes perfect sense, since everything in football is driven by numbers. From the number of yards in a game to the number of wins in a season, numbers are what determine success in football. However, quantitative data is not the only approach to predicting success.

1.2 The Qualitative Approach

A lesser used form of data analytics in the NFL is qualitative analysis. This type of analysis relies on non-numeric factors rather than quantitative data. Instead of evaluating players strictly on their numbers at the combine, this approach allows NFL teams to evaluate players without knowing a single one of a prospect's numerical stats. This can allow NFL teams to catch aspects about certain prospects that might be missed by the events of the combine alone.

1.3 Goal

The goal of this thesis is to employ qualitative analysis techniques to show a new approach in evaluating draft prospects to the NFL. Specifically, I aim to employ word analysis on the strengths and weakness sections found in player's draft reports in order to determine which factors lead to success in the NFL. I define success as a player remaining with their draft team after 4 years in the league.

Chapter 2

Collection and Creation of Data

2.1 Collection of Data

The data for this study were collected from three sources. Firstly, the scouting reports for all 122 players were scraped from walterfootball.com [2]. These reports were formatted with three sections – a list of strengths, a list of weaknesses, and a general summary of the draft prospect. Player names, draft year, and the three scouting report sections were aggregated into 4 separate datasets, one for each player position. The data were collected for years 2013 to 2017, and all completed scouting reports for players from each position were selected. Secondly, I added the players' draft team and draft round to all four datasets from pro-football-reference.com [3]. This was done by pulling all players that were drafted from 2013 to 2017, and cross-listing with the four datasets created previously. Lastly, I added a player's team after 4 years in the NFL to the four datasets from nfl.com [5]. A rookie contract in the NFL has a maximum length of 4 seasons, so a player's team after 4 years indicates whether they got a second contract with their draft team. Data were pulled for only the players already in the four datasets, and an indicator function was created to determine whether a player received a second contract with his draft team. This was coded as 1 if a player's draft team matched his team after 4 years, and 0 otherwise. Precautions were taken for when a team's name changes. If a player is still on a team with a name change after 4 years, they are still considered to have received a second contract. The four datasets were then concatenated into one master dataset, which was used for analysis.

2.2 List of Variables

Below in Table 2.1 is a list of variables in the final dataset.

Table 2.1: List of Variables in Master Dataset

Variable	Class	Description
player	Categorical	The name of the player
draft_year	Numeric	The year the player was drafted
position	Categorical	The position of the player
draft_rnd	Numeric	The round in which the player was drafted
draft_team	Categorical	The team to which the player was drafted
second_team_abb	Categorical	The team of the player after 4 years
second_contract	Binary	Whether the player received a second contract with his draft team
strengths	Text	The strengths of the player from his scouting report
weaknesses	Text	The weaknesses of the player from his scouting report
summary	Text	The summary of the player from his scouting report

2.3 Creation of New Variables

After pulling the variables above from their respective sources, I created new variables to use for analysis. The first new variable is called strengths_count. In order to create this variable, I first obtained a list of the 20 most common two-word phrases from the strengths section of the scouting reports for each of the 4 positions [4]. Two-word phrases were chosen to capture the use of qualifiers. For instance, a two-word phrase can determine the difference between ”good” and ”not good”. Below I have included a table of the top 5 most common strengths for each position from the lists of 20.

Table 2.2: List of Most Common Strengths by Position

Position	Strength Phrase	Frequency
QB	field vision	12
QB	pocket presence	9
QB	strong arm	7
QB	tight windows	6
QB	hard worker	5
WR	route runner	29
WR	play maker	23
WR	body control	19
WR	quick release	17
WR	red zone	17
DE	bull rush	22
DE	pass rusher	22
DE	run defender	21
DE	pass rushing	19
DE	shed blocks	16
CB	ball skills	26
CB	speed receivers	21
CB	cover corner	18
CB	quick feet	18
CB	loose hips	15

For each player, I then calculated how many of the 20 most common two-word phrases for their position showed up in the strengths section of their scouting report [4]. This became the strengths_count variable.

A similar process was repeated on the weaknesses section of the scouting reports to acheive a new variable called weak_count. Below is the table of the top 5 most common weaknesses for each position from the lists of 20.

Table 2.3: List of Most Common Weaknesses by Position

Position	**Weakness Phrase**	**Frequency**
QB	improve footwork	6
QB	ball placement	3
QB	pro style	3
QB	style offense	3
QB	college offense	2
WR	lacks elite	6
WR	route running	6
WR	deep threat	5
WR	nfl corners	5
WR	overly fast	5
DE	pass rushing	12
DE	rushing moves	11
DE	3 4	9
DE	4 3	7
DE	run defense	7
CB	speed receivers	4
CB	50 50	3
CB	nfl speed	3
CB	red zone	3
CB	50 passes	2

From here, for each player I calculated how many of the 20 most common two-word phrases for their position showed up in the weaknesses section of their scouting report. This became the weak_count variable.

Chapter 3

Exploratory Data Analysis

3.1 EDA

After collecting and creating my data, I took an initial look at what it showed. In the data, only 33.6% of players received a second contract with their draft team. In other words, before fitting any models, we would have around a 33.6% accuracy rating just by guessing alone. From here, I looked at the proportion of players who received a second contract with their draft team by how many strengths were in their draft report. This is shown in the figure below.

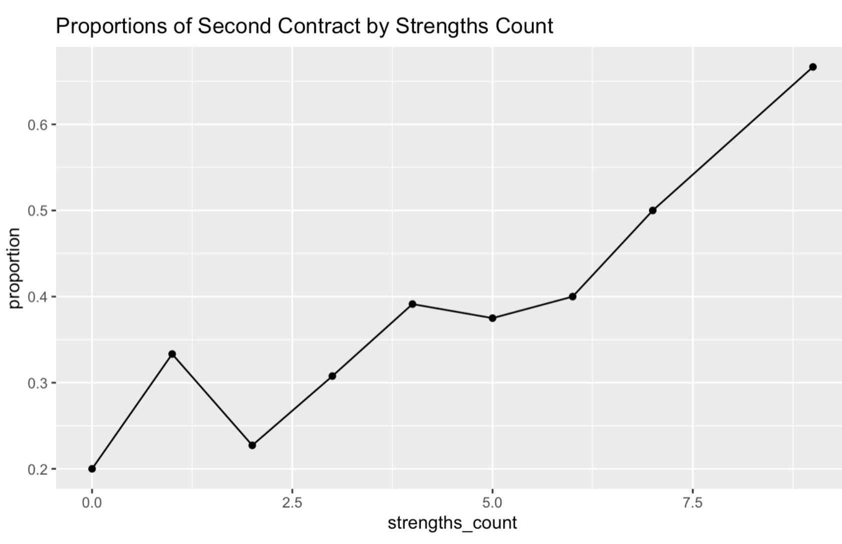

Figure 3.1: Proportion of Second Contract by Strengths Count

As one can see, there is a clear positive trend in the data. As the number of strengths increases, so does the proportion of players who received a second contract with their draft team. This could be indicative of a significant relationship between the two variables.

Likewise, I looked at the proportion of players who received a second contract with their draft team by how many weaknesses were in their draft report. This is shown in the figure below.

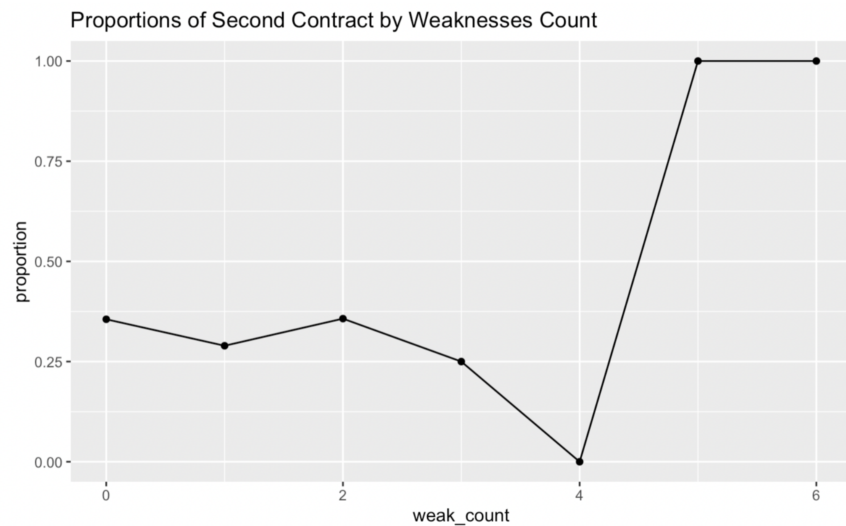

Figure 3.2: Proportion of Second Contract by Weaknesses Count

This graph does not show any clear trend in the data. In fact, it shows that the highest proportion of players that received a second contract with their draft team were those with 5 or 6 weaknesses in their draft report. This may be due to a lack of data, since only 2 players had over 5 weaknesses. However, this also may be indicative of an insignificant relationship between weak_count and second_contract.

Additionally, I looked at the proportions of players that received a second contract from their draft team by position. This is shown in the table below.

Table 3.1: Table of Proportion of Second Contracts with Draft Team in Data by Position

Position	**Proportion of Second Contracts with Draft Team in Data**
CB	0.406
WR	0.359
QB	0.313
DE	0.257

From this table, there seems to be a disparity of proportion of players who received a second contract with their draft team by postion, with a range of nearly 15%. This too may indicate a significant relationship between the two variables.

From here, I chose strength_count, weak_count, and position as my explanatory variables for analysis. Strength_count and postion were chosen because they show the potential for a significant correlation to the response second_contract, and weak_count was chosen for it's pertinence to the research question.

Chapter 4

Background on Models

4.1 General Approach

Through my analysis, I aimed to discover the important factors in determining a player's chance of receiving a second contract with his draft team. To do this, I fit three different statistical models to my data. To begin, I fit a k nearest neighbors classifier to my data. From there, I expanded my approach to more robust statistical methods through the use of classification trees and logistic regression. For all of these approaches, it is necessary to split the data into a training and a test set. Each model was created using the training set, and subsequently run on the test set. This was done to better evaluate the predictive power of the models. The training set was selected as a random 70% partition of the dataset, and the leftover 30% was designated to the test set.

4.2 K Nearest Neighbors

Firstly, I fit a k nearest neighbors classifier to my data. This is a non-parametric method to predict the value of a response variable [1]. The general approach is as follows:

1. For a given observation in the test set, determine its location on the coordinate plane of independent variables (x_1, x_2, x_3, x_p). For instance, an observation with values x_1 = 1, x_2 = 4, & x_3 = 2 would be denoted as point (1, 4, 2) on a 3-dimensional coordinate plane.

2. For each observation in the test set, identify the k closest observations on the coordinate plane from the training set.

3. The predicted value of the response variable for observations in the test set is taken as the most frequent response value of the k closest observations from the training set [1].

For the purposes of the analysis at hand, I will be using a k nearest neighbors classifier with k = 3 to my data. Though a larger k value is normally preferred, the relatively small nature of the data set and a concern for overfitting led to my decision to pick k = 3. The explanatory variables will include strengths_count and weak_count, and the response will be second_contract. I aim to determine whether the number of strengths and number of weaknesses in an individual scouting report are effective in predicting whether a player receives a second contract with his draft team.

4.3 Classification Tree

Secondly, I fit a classification tree to my data. This is also a non-parametric method of predicting the value of a response variable [1]. The general approach for this analysis is as follows:

1. Fit a tree with g terminal nodes, where splits are determined using recursive binary splitting. This is a process by which the splits are decided by minimizing $\frac{1}{n}\sum_{m=1}^{g} n_m * I_m$, where n is the total number of observations, n_m is the number of observations in the m^{th} node, and I_m is the impurity of the m^{th} node.

2. The impurity measure selected for the purposes of this analysis is the Gini index, which is defined as $G_m = \sum_{c=1}^{w} \hat{p}_{m,c}(1-\hat{p}_{m,c})$, where $\hat{p}_{m,c} = \frac{n_{m,c}}{n_m}$. $n_{m,c}$ is the amount of observations in the m^{th} node that belong in category c, and n_m is the amount of observations in the m^{th} node.

3. Find a series of nested subtrees as a function of a tuning parameter, λ. This is known as cost complexity pruning.

4. The final subtree is selected as the λ value that results in the lowest cross-validation error [1].

For the purposes of the analysis at hand, I constructed a classification tree to predict whether a player receives a second contract in the NFL with their draft team. The explanatory variables included strengths_count, weak_count, and position, and the response was second_contract.

4.4 Logistic Regression

Lastly, I fit a logistic regression to my data. This is a non-linear regression model with a binary response variable. A binary variable is defined as only having two possible outcomes, often hardcoded as 0 or 1. The logistic regression models the response with a logit link, otherwise known as the log-odds of the response event occuring [1]. The model takes the following form:

$$ln(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

[1]

where,

- π is the probability of the response event occuring
- β_0 is the value for the intercept
- β_i is the value for the i^{th} coefficient
- X_i is the value for the i^{th} explantory variable

Fitting a logistic regression with all explanatory variables provides a full model. From there, it is necessary to perform model reduction in order to find the best model according to a particular model selection technique. For my purposes, I used backwards elimination according to AIC [1]. This process is as follows:

- Fit the model with all g explanatory variables using the training set.
- Fit all models that drop one of the explanatory variables. The model with the largest R^2 is chosen.

- Repeat this process by dropping one explantory variable at a time, and selecting the model with the largest R^2 at each step.

- Choose the best model among all final models at each step using Akaike Information Criterion, or AIC [1].

AIC is calculated as $\mathrm{AIC} = 2k - 2\ln(\hat{L})$, where $\hat{L}$ is the estimated maximum log-likelihood of the model, and k is the number of parameters in the model [1].

For the purposes of the analysis at hand, I fit a full logistic regression model using strengths_count, weak_count, position, and all two-way interactions of these variables to predict the log-odds of a player receiving a second contract with his draft team. I then reduced the model using backwards elimination according to AIC. This model was then tested on the test set, and an accuracy percentage was calculated.

Chapter 5

Model Fitting and Analysis

5.1 Fitting a K Nearest Neighbors Classifier

For my analysis, I began by fitting a k nearest neighbors classifier to my data with k = 3. The explanatory variables included strengths_count and weak_count, and the response was second_contract. Position was omitted because it is qualitative in nature. Using the training set, the response values for the test set were predicted, and an accuracy percentage was calculated. In order to reduce variation, the data was split into a random partition of 70% for the training set and 30% for the test set 10 different times, and the accuracies of the knn models were averaged.

A final accuracy amount of 92.24% was obtained. This means that on average, over the ten runs of the knn model, the variables strengths_count and weak_count were able to classify second_contract correctly 92.24% of the time. This shows that the combination of the number of strengths and weaknesses in a player's draft report is useful in determining whether a player received a second contract with their draft team. From here, it was necessary to take a deeper dive into these factors.

5.2 Fitting a Classification Tree

I then fit a classifcation tree to my training data. The explanatory variables included strengths_count, weak_count, and position, and the response was once again second_contract. I decided to include the position variable due to aforementioned Table 3.1, where there seems to be a disparity of proportion of players who received a second contract with their draft team by postion. I felt that the classification tree would be a useful tool to see if this relationship is significant.

After fitting the classfication tree to my training data, I obtained the output in the figure below.

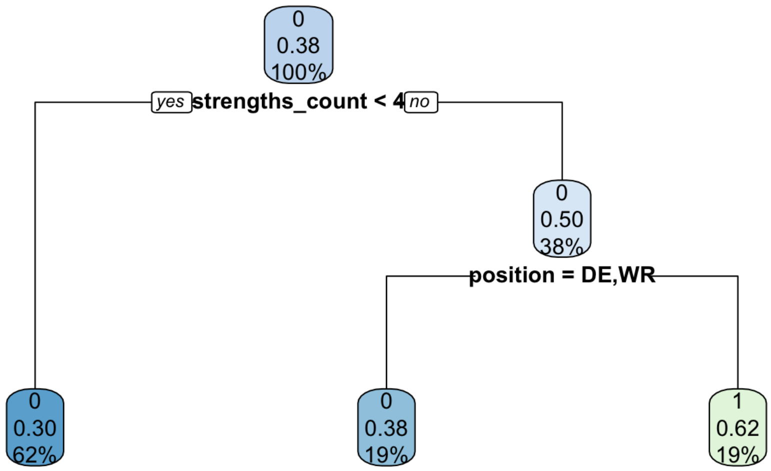

Figure 5.1: Classification Tree

Walking through this model, one can see that a player with under 4 strengths is predicted to not receive a second contract with his draft team. When a player has over 4 strengths, the predicted value depends on his position. If the player is a defensive end or wide receiver, they are predicted to not receive a second contract with their draft team. However, when a player is a quarterback or cornerback, they are predicted to receive a second contract with their draft team.

From this model, it is clear that the amount of strengths in a player's draft report, as well as his position, are important in determining whether they receive a second contract with their draft team. When using this model to predict the response in the test set, it had an accuracy percentage of 70.27%. This is an improvement over the 33.6% accuracy from random guessing. It is interesting to note that this model ended up omitting the weaknesses in a player's draft report entirely.

5.3 Fitting a Logistic Regression Model

Lastly, I fit a logistic regression model to my training data. I began by fitting a full model using strengths_count, weak_count, position, and all two-way interactions of these variables to predict the log-odds of a player receiving a second contract with his draft team. This full model was calculated as:

Table 5.1: Full Model

Variable	Coefficient	Standard Error	t-value	p-value
Intercept	-0.07717	0.26593	-0.290	0.7725
strengths_count	0.09407	0.04976	1.890	0.0627
weak_count	0.16363	0.20454	0.800	0.4264
position_DE	0.16177	0.36519	0.443	0.6591
position_QB	0.32862	0.41472	0.792	0.4307
position_WR	0.45604	0.30994	1.471	0.1455
strengths_count:weak_count	0.01911	0.03146	0.607	0.5455
strengths_count:position_DE	-0.09175	0.09609	-0.955	0.3429
strengths_count:position_QB	0.04252	0.10733	0.396	0.6932
strengths_count:position_WR	-0.11122	0.07674	-1.449	0.1516
weak_count:position_DE	-0.16422	0.17898	-0.918	0.3619
weak_count:position_QB	-0.36503	0.20228	-1.805	0.0753
weak_count:position_WR	-0.20832	0.17044	-1.222	0.2256

After performing backwards elimination using AIC as the selection criterion, I came to a final model of:

Table 5.2: Reduced Model

Variable	Coefficient	Standard Error	t-value	p-value
Intercept	0.09670	0.09713	0.996	0.32235
strengths_count	0.07881	0.02833	2.782	0.00669

As one can see, the selected final model retained only the number of strengths in a player's draft report as a predictor for the log-odds of receiving a second contract with their draft team (p-value = 0.007). A confidence interval was calculated on the strengths_count coefficient. With 95% confidence, the true impact of one additional strength on the log-odds of receiving a second contract is between (0.02328039, 0.1343357). When this model was run on the test set, it had an accuracy percentage of 56.76%. This is also an improvement over the 33.6% accuracy from random guessing.

5.4 Predictions for 2022 Draft QBs

To show the use of the models above, I gathered data on 6 quarterbacks from the upcoming 2022 NFL Draft - Matt Corral, Sam Howell, Kenny Pickett, Brock Purdy, Desmond Ridder, and Malik Willis. The final models above assigned the probabilities of a second contract with their draft team in the table below.

Table 5.3: Table of Predicted Probailities of Second Contracts with Draft Team for 2022 QBs

Draftee	**Number of Strengths**	**Classification Tree**	**Logistic Regression**
Matt Corral	5	.620	.620
Kenny Pickett	4	.300	.602
Desmond Ridder	4	.300	.602
Malik Willis	3	.300	.583
Sam Howell	1	.300	.544
Brock Purdy	1	.300	.544

The k nearest neighbors model was omitted as it does not report a probability, but rather assigns individual players a strict value for the second_contract variable. Probabilities for the classification tree were taken from Figure 5.1. Matt Corral is the only QB with over 4 strengths, so the tree gives him a 62% chance of receiving a second contract with his draft team. All other QBs are given a 30% chance. Probabilites for the logistic regression were calculated as $\hat{\pi} = \frac{\exp 0.09670+0.07881*strengths\_count}{1+\exp 0.09670+0.07881*strengths\_count}$. Unlike the classification tree, which put players in buckets, these probabilities were calculated based on unit increases in strengths. As one can see, Matt Corral is predicted to have the greatest probability of receiving a second contract with his draft team, with a probability of 62%. This is interesting due to the fact that Corral has not received as much press as other QBs like Willis, Pickett, and Ridder, who all have a lower number of strengths in their draft report, and a lower predicted probability of receiving a deal with their draft team after their rookie contract is over. This drives home the pertinence of a holistic approach to evaluating NFL Draft prospects. While the numbers may say that Willis, Pickett, and Ridder are more NFL ready, there are other immeasurables that are found in Corral's draft report that may indicate otherwise.

Chapter 6

Conclusion and Future Considerations

6.1 Results

The goal of this thesis was to provide a qualitative way to evaluate NFL draft prospects using their scouting reports. This was accomplished through the use of machine learning techniques, like k nearest neighbors, classification tress, and logisitic regression. By examining the strengths and weaknesses within a players' draft reports, I was able to determine that the strengths are significant in determining a player's chance of receiving a second contract with his draft team, while the weakneses are not significant.

6.2 Strengths

From my analysis, it is clear that as a player increases the number of strengths within their draft report, they are more likely to receive a second contract with their draft team. Using the final logistic regression model from my analysis, one can say that for every extra strength phrase a player has in his draft report, his odds of receiving a second contract with his draft team increase by a factor of 1.082. This odds ratio calculated as $\exp(\beta_1) = \exp(0.07881) = 1.082$. In simpler terms, if a player is designated as having one additional strength over a fellow draftee, he is 1.082 times more likely to receive a second contract with his draft team [1].

6.3 Weaknesses

It is also clear that the weaknesses within a player's draft report are not significant in determining whether a player receives a second contract with his draft team. This is shown by the variable's omission in both the classification tree to predict whether a player received a second contract with their draft team, and in the logistic regression to predict the log-odds of a player receiving a second contract with their draft team.

6.4 Limitations of My Analysis

While my analysis is intriguing, it does have limitations. Firstly, the data only contains 122 observations, split among the positions as shown below in Table 6.1.

Table 6.1: Table of Number of Observations by Position

Position	**Number of Observations**
WR	39
DE	35
CB	32
QB	16

Ideally, more observations for each position would lead to more accuracy in my results, particularly for the quartback position. For my analysis, I was limited to which draft reports were publicly available, and thus did not have extensive access to further data.

As well, this analysis is constrained by the fact that data were only collected from one draft analyst. While this was useful in finding my lists of most common strengths and most common weaknesses, this does limit the scope of which the results can be considered. Without the inclusion of more analysts in my data collection, it is hard to apply the results from this analysis to their draft reports. All analysts have different styles of writing, and it would not be statistically sound to say that the results from my models apply to all reports. Therefore, it is only just to say that the results from this analysis apply only to those draft reports written by Walter Cherepinsky.

6.5 Next Steps

6.5.1 Determine Pertinent Strengths

In terms of this analysis, an appropriate next step would be to determine which strengths are most integral in determining whether a player will be with the same team after 4 years. While it is good to know that strengths are what matter, it is more important to take a deeper dive and determine which strengths are valued the most. This is the information that would be truly useful to those involved in the NFL drafting process.

There are many ways for this to be accomplished, but I would recommend running another logistic regression model. The process would be as follows:

1. For the list of the 20 most common two-word strengths phrases in the data, create a dummy variable for each one. This would be coded as 1 if a player has the strength in their draft report, and 0 if they do not.

2. Fit a full logistic regression model with the response as the log-odds of a player receiving a second contract with their draft team, and with the explanatory variables as the 20 strengths dummy variables.

3. Use backwards elimination according to AIC in order to find a reduced model.

4. Analyze the dummy variables remaining. If the exponential of the dummy variable's coefficient is positive, then it is important in increasing a player's odds of attaining a second contract with their draft team. If the exponential of the dummy variable's coefficient is negative, then it actually leads to a decrease in a player's odds of attaining a second contract with their draft team.

Appendix - R Code

```
# Clear working environment
rm(list = ls())

# Load necessary packages
library(rvest)
library(tidyverse)
library(ggplot2)
library(stringr)

# Create empty dataframes
playerdata_qb <- data.frame(matrix(ncol = 5, nrow = 0))
colnames <- c("name", "year", "strengths", "weaknesses", "summary")
colnames(playerdata_qb) <- colnames

playerdata_wr <- data.frame(matrix(ncol = 5, nrow = 0))
colnames <- c("name", "year", "strengths", "weaknesses", "summary")
colnames(playerdata_wr) <- colnames

playerdata_de <- data.frame(matrix(ncol = 5, nrow = 0))
colnames <- c("name", "year", "strengths", "weaknesses", "summary")
colnames(playerdata_de) <- colnames

playerdata_cb <- data.frame(matrix(ncol = 5, nrow = 0))
colnames <- c("name", "year", "strengths", "weaknesses", "summary")
colnames(playerdata_cb) <- colnames

# Create list of filepaths
filepaths_qb <- c('https://walterfootball.com/scoutingreport2012nfoles.php',
'https://walterfootball.com/scoutingreport2012rgriffin.php',
'https://walterfootball.com/scoutingreport2012aluck.php',
'https://walterfootball.com/scoutingreport2012bosweiler.php',
'https://walterfootball.com/scoutingreport2012rtannehill.php',
'https://walterfootball.com/scoutingreport2012bweeden.php',
'https://walterfootball.com/scoutingreport2013mbarkley.php',
'https://walterfootball.com/scoutingreport2013tbray.php',
'https://walterfootball.com/scoutingreport2013mglennon.php',
```

```
'https://walterfootball.com/scoutingreport2013emanuel.php',
'https://walterfootball.com/scoutingreport2013rnassib.php',
'https://walterfootball.com/scoutingreport2013gsmith.php',
'https://walterfootball.com/scoutingreport2013twilson.php',
'https://walterfootball.com/scoutingreport2014bbortles.php', 'https://walterfootbal

filepaths_wr <- c('https://walterfootball.com/scoutingreport2012jblackmon.php', 'ht
'https://walterfootball.com/scoutingreport2013kallen.php',
'https://walterfootball.com/scoutingreport2013taustin.php',
'https://walterfootball.com/scoutingreport2013sbailey.php',
'https://walterfootball.com/scoutingreport2013dhopkins.php',
'https://walterfootball.com/scoutingreport2013jhunter.php',
'https://walterfootball.com/scoutingreport2013cpatterson.php',
'https://walterfootball.com/scoutingreport2013qpatton.php',
'https://walterfootball.com/scoutingreport2013mwheaton.php',
'https://walterfootball.com/scoutingreport2013twilliams.php',
'https://walterfootball.com/scoutingreport2014jabbrederis.php', 'https://walterfoot
'https://walterfootball.com/scoutingreport2014jlandry.php',
'https://walterfootball.com/scoutingreport2014mlee.php',
'https://walterfootball.com/scoutingreport2014jmatthews.php',
'https://walterfootball.com/scoutingreport2014dmoncrief.php',
'https://walterfootball.com/scoutingreport2014arobinson.php',
'https://walterfootball.com/scoutingreport2014swatkins.php',
'https://walterfootball.com/scoutingreport2015nagholor.php', 'https://walterfootbal

filepaths_de <- c('https://walterfootball.com/scoutingreport2012abranch.php',
'https://walterfootball.com/scoutingreport2012qcoples.php',
'https://walterfootball.com/scoutingreport2012vcurry.php',
'https://walterfootball.com/scoutingreport2012mingram.php', 'https://walterfootball
'https://walterfootball.com/scoutingreport2013eansah.php',
'https://walterfootball.com/scoutingreport2013tcarradine.php',
'https://walterfootball.com/scoutingreport2013mhunt.php',
'https://walterfootball.com/scoutingreport2013djones.php',
'https://walterfootball.com/scoutingreport2013djordan.php',
'https://walterfootball.com/scoutingreport2013smontgomery.php',
'https://walterfootball.com/scoutingreport2013dmoore.php',
'https://walterfootball.com/scoutingreport2013aokafor.php',
'https://walterfootball.com/scoutingreport2013bwerner.php',
'https://walterfootball.com/scoutingreport2014jclowney.php', 'https://walterfootbal

filepaths_cb <- c('https://walterfootball.com/scoutingreport2012mclaiborne.php',
'https://walterfootball.com/scoutingreport2012adennard.php',
'https://walterfootball.com/scoutingreport2012sgilmore.php',
'https://walterfootball.com/scoutingreport2012chayward.php',
'https://walterfootball.com/scoutingreport2012jhosley.php',
'https://walterfootball.com/scoutingreport2012jjenkins.php', 'https://walterfootbal
'https://walterfootball.com/scoutingreport2013jbanks.php',
```

```
'https://walterfootball.com/scoutingreport2013tmathieu.php',
'https://walterfootball.com/scoutingreport2013dmilliner.php',
'https://walterfootball.com/scoutingreport2013jpoyer.php',
'https://walterfootball.com/scoutingreport2013xrhodes.php',
'https://walterfootball.com/scoutingreport2013lryan.php',
'https://walterfootball.com/scoutingreport2013jtaylor.php',
'https://walterfootball.com/scoutingreport2013dtrufant.php',
'https://walterfootball.com/scoutingreport2014bbreeland.php', 'https://walterfootba

# Read in qbs players into the dataframe
for (i in filepaths_qb) {
  file <- read_html(i)

list <- file %>%
  html_nodes("li␣,␣b") %>%
  html_text()

strengths <- which(list == 'Strengths:␣')
weaknesses <- which(list == 'Weaknesses:␣')
summary <- which(list == 'Summary:␣')

name <-  file %>%
  html_nodes("h1") %>%
  html_text()

name1 <- str_remove_all(name, '\r\n')
name2 <- str_remove_all(name1, '2012␣NFL␣Draft␣Scouting␣Report:|2013␣NFL␣Draft␣Scou
name3 <- str_remove_all(name2, '␣')

name4 <- str_remove_all(name, '\r\n')
name5 <- str_remove_all(name4, 'NFL␣Draft␣Scouting␣Report:|NFL␣Draft␣Player␣Previe
name6 <- str_remove_all(name5, '␣')
name7 <- str_remove_all(name6, name3)

player_name <- name3
player_year <- substr(name7,1,4)
player_strengths <- list[(strengths+1):(weaknesses-1)]
player_weaknesses <- list[(weaknesses+1):(summary-1)]

list2 <- file %>%
  html_nodes("div␣div") %>%
  html_text()
list3 <- str_replace_all(list2[17], '\r', '␣')
list3 <- str_replace_all(list3, '\n', '␣')

summary <- str_extract_all(list3, regex("Summary:.*Player␣Comparison"))
summary <- str_replace_all(summary, 'Summary:', '␣')
```

```
summary <- str_replace_all(summary, 'Player␣Comparison', '␣')

playerdata2 <- data.frame(player_name, player_year, I(list(player_strengths)), I(li
colnames(playerdata2) <- colnames
playerdata_qb <- rbind(playerdata_qb, playerdata2)
colnames(playerdata_qb) <- colnames
}

# Read in wrs players into the dataframe
for (i in filepaths_wr) {
  file <- read_html(i)

list <- file %>%
  html_nodes("li␣,␣b") %>%
  html_text()

strengths <- which(list == 'Strengths:␣')
weaknesses <- which(list == 'Weaknesses:␣')
summary <- which(list == 'Summary:␣')

name <-  file %>%
  html_nodes("h1") %>%
  html_text()

name1 <- str_remove_all(name, '\r\n')
name2 <- str_remove_all(name1, '2012␣NFL␣Draft␣Scouting␣Report:|2013␣NFL␣Draft␣Scou
name3 <- str_remove_all(name2, '␣')

name4 <- str_remove_all(name, '\r\n')
name5 <- str_remove_all(name4, 'NFL␣Draft␣Scouting␣Report:|NFL␣Draft␣Player␣Previe
name6 <- str_remove_all(name5, '␣')
name7 <- str_remove_all(name6, name3)

player_name <- name3
player_year <- substr(name7,1,4)
player_strengths <- list[(strengths+1):(weaknesses-1)]
player_weaknesses <- list[(weaknesses+1):(summary-1)]

list2 <- file %>%
  html_nodes("div␣div") %>%
  html_text()
list3 <- str_replace_all(list2[17], '\r', '␣')
list3 <- str_replace_all(list3, '\n', '␣')

summary <- str_extract_all(list3, regex("Summary:.*Player␣Comparison"))
summary <- str_replace_all(summary, 'Summary:', '␣')
summary <- str_replace_all(summary, 'Player␣Comparison', '␣')
```

```
playerdata2 <- data.frame(player_name, player_year, I(list(player_strengths)), I(li
colnames(playerdata2) <- colnames
playerdata_wr <- rbind(playerdata_wr, playerdata2)
colnames(playerdata_wr) <- colnames
}

# Read in des players into the dataframe
for (i in filepaths_de) {
  file <- read_html(i)

list <- file %>%
  html_nodes("li␣,␣b") %>%
  html_text()

strengths <- which(list == 'Strengths:␣')
weaknesses <- which(list == 'Weaknesses:␣')
summary <- which(list == 'Summary:␣')

name <-  file %>%
  html_nodes("h1") %>%
  html_text()

name1 <- str_remove_all(name, '\r\n')
name2 <- str_remove_all(name1, '2012␣NFL␣Draft␣Scouting␣Report:|2013␣NFL␣Draft␣Scou
name3 <- str_remove_all(name2, '␣')

name4 <- str_remove_all(name, '\r\n')
name5 <- str_remove_all(name4, 'NFL␣Draft␣Scouting␣Report:|NFL␣Draft␣Player␣Previe
name6 <- str_remove_all(name5, '␣')
name7 <- str_remove_all(name6, name3)

player_name <- name3
player_year <- substr(name7,1,4)
player_strengths <- list[(strengths+1):(weaknesses-1)]
player_weaknesses <- list[(weaknesses+1):(summary-1)]

list2 <- file %>%
  html_nodes("div␣div") %>%
  html_text()
list3 <- str_replace_all(list2[17], '\r', '␣')
list3 <- str_replace_all(list3, '\n', '␣')

summary <- str_extract_all(list3, regex("Summary:.*Player␣Comparison"))
summary <- str_replace_all(summary, 'Summary:', '␣')
summary <- str_replace_all(summary, 'Player␣Comparison', '␣')
```

```
playerdata2 <- data.frame(player_name, player_year, I(list(player_strengths)), I(li
colnames(playerdata2) <- colnames
playerdata_de <- rbind(playerdata_de, playerdata2)
colnames(playerdata_de) <- colnames
}

# Read in cbs players into the dataframe
for (i in filepaths_cb) {
  file <- read_html(i)

list <- file %>%
  html_nodes("li␣,␣b") %>%
  html_text()

strengths <- which(list == 'Strengths:␣')
weaknesses <- which(list == 'Weaknesses:␣')
summary <- which(list == 'Summary:␣')

name <-  file %>%
  html_nodes("h1") %>%
  html_text()

name1 <- str_remove_all(name, '\r\n')
name2 <- str_remove_all(name1, '2012␣NFL␣Draft␣Scouting␣Report:|2013␣NFL␣Draft␣Scou
name3 <- str_remove_all(name2, '␣')

name4 <- str_remove_all(name, '\r\n')
name5 <- str_remove_all(name4, 'NFL␣Draft␣Scouting␣Report:|NFL␣Draft␣Player␣Previe
name6 <- str_remove_all(name5, '␣')
name7 <- str_remove_all(name6, name3)

player_name <- name3
player_year <- substr(name7,1,4)
player_strengths <- list[(strengths+1):(weaknesses-1)]
player_weaknesses <- list[(weaknesses+1):(summary-1)]

list2 <- file %>%
  html_nodes("div␣div") %>%
  html_text()
list3 <- str_replace_all(list2[17], '\r', '␣')
list3 <- str_replace_all(list3, '\n', '␣')

summary <- str_extract_all(list3, regex("Summary:.*Player␣Comparison"))
summary <- str_replace_all(summary, 'Summary:', '␣')
summary <- str_replace_all(summary, 'Player␣Comparison', '␣')

playerdata2 <- data.frame(player_name, player_year, I(list(player_strengths)), I(li
```

```
colnames(playerdata2) <- colnames
playerdata_cb <- rbind(playerdata_cb, playerdata2)
colnames(playerdata_cb) <- colnames
}

# Add variable "position" to each dataframe
playerdata_qb$position <- 'QB'
playerdata_wr$position <- 'WR'
playerdata_de$position <- 'DE'
playerdata_cb$position <- 'CB'

# Reorder dataframe columns
playerdata_qb <- playerdata_qb[c(1,2,6,3,4,5)]
playerdata_wr <- playerdata_wr[c(1,2,6,3,4,5)]
playerdata_de <- playerdata_de[c(1,2,6,3,4,5)]
playerdata_cb <- playerdata_cb[c(1,2,6,3,4,5)]

# Pull in draft round data
filepath2012 <- read_html("https://www.pro-football-reference.com/years/2012/draft.
filepath2013 <- read_html("https://www.pro-football-reference.com/years/2013/draft.
filepath2014 <- read_html("https://www.pro-football-reference.com/years/2014/draft.
filepath2015 <- read_html("https://www.pro-football-reference.com/years/2015/draft.
filepath2016 <- read_html("https://www.pro-football-reference.com/years/2016/draft.
filepath2017 <- read_html("https://www.pro-football-reference.com/years/2017/draft.

draft2012 <-
  filepath2012 %>%
  html_table() %>%
  .[[1]]
names(draft2012) <- draft2012[1,]

draft2013 <-
  filepath2013 %>%
  html_table() %>%
  .[[1]]
names(draft2013) <- draft2013[1,]

draft2014 <-
  filepath2014 %>%
  html_table() %>%
  .[[1]]
names(draft2014) <- draft2014[1,]

draft2015 <-
  filepath2015 %>%
  html_table() %>%
  .[[1]]
```

```
names(draft2015) <- draft2015[1,]

draft2016 <-
  filepath2016 %>%
  html_table() %>%
  .[[1]]
names(draft2016) <- draft2016[1,]

draft2017 <-
  filepath2017 %>%
  html_table() %>%
  .[[1]]
names(draft2017) <- draft2017[1,]

# Rename variables for merging
colnames(draft2012)[4] <- "name"
draft2012$name <- gsub('␣', '', draft2012$name)

colnames(draft2013)[4] <- "name"
draft2013$name <- gsub('␣', '', draft2013$name)

colnames(draft2014)[4] <- "name"
draft2014$name <- gsub('␣', '', draft2014$name)

colnames(draft2015)[4] <- "name"
draft2015$name <- gsub('␣', '', draft2015$name)

colnames(draft2016)[4] <- "name"
draft2016$name <- gsub('␣', '', draft2016$name)

colnames(draft2017)[4] <- "name"
draft2017$name <- gsub('␣', '', draft2017$name)

draft1 <- rbind(draft2012, draft2013)
draft2 <- rbind(draft1, draft2014)
draft3 <- rbind(draft2, draft2015)
draft4 <- rbind(draft3, draft2016)
draft <- rbind(draft4, draft2017)

# Add draft round to datasets
playerdatadraft_qb <- merge(playerdata_qb,draft, by="name")[c(1:7,9)]
playerdatadraft_wr <- merge(playerdata_wr,draft, by="name")[c(1:7,9)]
playerdatadraft_de <- merge(playerdata_de,draft, by="name")[c(1:7,9)]
playerdatadraft_cb <- merge(playerdata_cb,draft, by="name")[c(1:7,9)]

playerdraftdata_qb <- playerdatadraft_qb[c(1,2,3,7,8,4,5,6)]
playerdraftdata_wr <- playerdatadraft_wr[c(1,2,3,7,8,4,5,6)]
```

```
playerdraftdata_de <- playerdatadraft_de[c(1,2,3,7,8,4,5,6)]
playerdraftdata_cb <- playerdatadraft_cb[c(1,2,3,7,8,4,5,6)]

# Create empty datasets
team2017 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2017) <- colnames

team2016 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2016) <- colnames

team2015 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2015) <- colnames

team2014 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2014) <- colnames

team2013 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2013) <- colnames

team2012 <- data.frame(matrix(ncol = 2, nrow = 0))
colnames <- c("name", "teamafter4yrs")
colnames(team2012) <- colnames

# Add team after 4 years for 2017 draftees
filenames2017 <- c("https://www.nfl.com/players/adoree-jackson/stats/career", "http
"https://www.nfl.com/players/solomon-thomas/stats/career", "https://www.nfl.com/pla

for (i in filenames2017) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2021 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2021\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', list)
  team2021 <- list[year2021 + 1]

  team2017B <- data.frame(player,team2021)
  colnames(team2017B) <- c("name", "teamafter4yrs")
  team2017B$name <- gsub('␣', '', team2017B$name)
  team2017B$teamafter4yrs <- gsub('␣', '', team2017B$teamafter4yrs)
```

```
  team2017 <- rbind(team2017, team2017B)
  colnames(team2017) <- c("name", "teamafter4yrs")
}

# Add team after 4 years for 2016 draftees
filenames2016 <- c("https://www.nfl.com/players/eli-apple/stats/career", "https://w

for (i in filenames2016) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2020 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2020\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', list)
  team2020 <- list[year2020 + 1]

  team2016B <- data.frame(player,team2020)
  colnames(team2016B) <- c("name", "teamafter4yrs")
  team2016B$name <- gsub('␣', '', team2016B$name)
  team2016B$teamafter4yrs <- gsub('␣', '', team2016B$teamafter4yrs)

  team2016 <- rbind(team2016, team2016B)
  colnames(team2016) <- c("name", "teamafter4yrs")
}

# Add team after 4 years for 2015 draftees
filenames2015 <- c("https://www.nfl.com/players/dante-fowler/stats/career", "https:

for (i in filenames2015) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2019 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2019\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', list)
  team2019 <- list[year2019 + 1]

  team2015B <- data.frame(player,team2019)
  colnames(team2015B) <- c("name", "teamafter4yrs")
  team2015B$name <- gsub('␣', '', team2015B$name)
  team2015B$teamafter4yrs <- gsub('␣', '', team2015B$teamafter4yrs)
```

```
  team2015 <- rbind(team2015, team2015B)
  colnames(team2015) <- c("name", "teamafter4yrs")
}

# Add team after 4 years for 2014 draftees
filenames2014 <- c("https://www.nfl.com/players/blake-bortles/stats/career", "https
"https://www.nfl.com/players/odell-beckham/stats/career", "https://www.nfl.com/play
"https://www.nfl.com/players/dominique-easley/stats/career", "https://www.nfl.com/p

for (i in filenames2014) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2018 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2018\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', list)
  team2018 <- list[year2018 + 1]

  team2014B <- data.frame(player,team2018)
  colnames(team2014B) <- c("name", "teamafter4yrs")
  team2014B$name <- gsub('␣', '', team2014B$name)
  team2014B$teamafter4yrs <- gsub('␣', '', team2014B$teamafter4yrs)

  team2014 <- rbind(team2014, team2014B)
  colnames(team2014) <- c("name", "teamafter4yrs")
}


# Add team after 4 years for 2013 draftees
filenames2013 <- c("https://www.nfl.com/players/matt-barkley/stats/career", "https:

for (i in filenames2013) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2017_1 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2017\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', lis
  team2017_1 <- list[year2017_1 + 1]

  team2013B <- data.frame(player,team2017_1)
  colnames(team2013B) <- c("name", "teamafter4yrs")
  team2013B$name <- gsub('␣', '', team2013B$name)
```

```
  team2013B$teamafter4yrs <- gsub('␣', '', team2013B$teamafter4yrs)

  team2013 <- rbind(team2013, team2013B)
  colnames(team2013) <- c("name", "teamafter4yrs")
}

# Add team after 4 years for 2012 draftees
filenames2012 <- c("https://www.nfl.com/players/nick-foles/stats/career",
"https://www.nfl.com/players/robert-griffin/stats/career",
"https://www.nfl.com/players/andrew-luck/stats/career",
"https://www.nfl.com/players/brock-osweiler/stats/career",
"https://www.nfl.com/players/ryan-tannehill/stats/career",
"https://www.nfl.com/players/brandon-weeden/stats/career",
"https://www.nfl.com/players/justin-blackmon/stats/career",
"https://www.nfl.com/players/michael-floyd/stats/career",
"https://www.nfl.com/players/josh-gordon/stats/career",
"https://www.nfl.com/players/stephen-hill/stats/career",
"https://www.nfl.com/players/alshon-jeffery/stats/career",
"https://www.nfl.com/players/rueben-randle/stats/career",
"https://www.nfl.com/players/mohamed-sanu/stats/career",
"https://www.nfl.com/players/kendall-wright/stats/career",
"https://www.nfl.com/players/andre-branch/stats/career",
"https://www.nfl.com/players/quinton-coples/stats/career",
"https://www.nfl.com/players/vinny-curry/stats/career",
"https://www.nfl.com/players/melvin-ingram/stats/career",
"https://www.nfl.com/players/chandler-jones/stats/career",
"https://www.nfl.com/players/nick-perry/stats/career",
"https://www.nfl.com/players/whitney-mercilus/stats/career",
"https://www.nfl.com/players/morris-claiborne/stats/career",
"https://www.nfl.com/players/alfonso-dennard/stats/career",
"https://www.nfl.com/players/stephon-gilmore/stats/career",
"https://www.nfl.com/players/casey-hayward/stats/career",
"https://www.nfl.com/players/jayron-hosley/stats/career",
"https://www.nfl.com/players/janoris-jenkins/stats/career",
"https://www.nfl.com/players/dre-kirkpatrick/stats/career")

for (i in filenames2012) {
  file <- read_html(i)
  list <- file %>%
    html_nodes("div␣h1,␣tr␣td") %>%
    html_text()

  player <- list[1]

  year2016_1 <- match('\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣2016\n␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣', lis
  team2016_1 <- list[year2016_1 + 1]
```

```
  team2012B <- data.frame(player,team2016_1)
  colnames(team2012B) <- c("name", "teamafter4yrs")
  team2012B$name <- gsub('␣', '', team2012B$name)
  team2012B$teamafter4yrs <- gsub('␣', '', team2012B$teamafter4yrs)

  team2012 <- rbind(team2012, team2012B)
  colnames(team2012) <- c("name", "teamafter4yrs")
}

# Add team after 4 years to datasets
team1 <- rbind(team2012, team2013)
team2 <- rbind(team1, team2014)
team3 <- rbind(team2, team2015)
team4 <- rbind(team3, team2016)
team <- rbind(team4, team2017)

playerdraftdata_cb <- merge(playerdraftdata_cb,team, by='name')
playerdraftdata_qb <- merge(playerdraftdata_qb,team, by='name')
playerdraftdata_de <- merge(playerdraftdata_de,team, by='name')
playerdraftdata_wr <- merge(playerdraftdata_wr,team, by='name')

playerdraftdata_cb <-
  playerdraftdata_cb %>%
  mutate(teamafter4yrs = ifelse(is.na(teamafter4yrs) == TRUE, 'FA', teamafter4yrs))
  mutate(teamafter4yrs = ifelse(teamafter4yrs == 'Info', 'FA', teamafter4yrs)) %>%
  mutate(teamafter4yrs = str_remove_all(teamafter4yrs, '\n'))

playerdraftdata_qb <-
  playerdraftdata_qb %>%
  mutate(teamafter4yrs = ifelse(is.na(teamafter4yrs) == TRUE, 'FA', teamafter4yrs))
  mutate(teamafter4yrs = ifelse(teamafter4yrs == 'Info', 'FA', teamafter4yrs)) %>%
  mutate(teamafter4yrs = str_remove_all(teamafter4yrs, '\n'))

playerdraftdata_de <-
  playerdraftdata_de %>%
  mutate(teamafter4yrs = ifelse(is.na(teamafter4yrs) == TRUE, 'FA', teamafter4yrs))
  mutate(teamafter4yrs = ifelse(teamafter4yrs == 'Info', 'FA', teamafter4yrs)) %>%
  mutate(teamafter4yrs = str_remove_all(teamafter4yrs, '\n'))

playerdraftdata_wr <-
  playerdraftdata_wr %>%
  mutate(teamafter4yrs = ifelse(is.na(teamafter4yrs) == TRUE, 'FA', teamafter4yrs))
  mutate(teamafter4yrs = ifelse(teamafter4yrs == 'Info', 'FA', teamafter4yrs)) %>%
  mutate(teamafter4yrs = str_remove_all(teamafter4yrs, '\n'))

# Initialize currentteam column for all datasets
playerdraftdata_cb$currentteam <- playerdraftdata_cb$teamafter4yrs
```

```r
playerdraftdata_qb$currentteam <- playerdraftdata_qb$teamafter4yrs
playerdraftdata_de$currentteam <- playerdraftdata_de$teamafter4yrs
playerdraftdata_wr$currentteam <- playerdraftdata_wr$teamafter4yrs

# Create coding system for NFL team name to abbreviation
teamtoabb <- function(x) {
  value = 0
  if (x == 'ArizonaCardinals') {
    value = 'ARI'
  }
  else if (x == 'AtlantaFalcons') {
    value = 'ATL'
  }
   else if (x == 'CarolinaPanthers') {
    value = 'CAR'
   }
   else if (x == 'ChicagoBears') {
    value = 'CHI'
   }
   else if (x == 'DallasCowboys') {
    value = 'DAL'
   }
   else if (x == 'DetroitLions') {
    value = 'DET'
   }
   else if (x == 'GreenBayPackers') {
    value = 'GNB'
   }
   else if (x == 'LosAngelesRams') {
    value = 'LAR'
   }
   else if (x == 'MinnesotaVikings') {
    value = 'MIN'
   }
   else if (x == 'NewOrleansSaints') {
    value = 'NOR'
   }
   else if (x == 'NewYorkGiants') {
    value = 'NYG'
   }
   else if (x == 'PhiladelphiaEagles') {
    value = 'PHI'
   }
   else if (x == 'SanFrancisco49ers') {
    value = 'SFO'
   }
   else if (x == 'SeattleSeahawks') {
```

```
 value = 'SEA'
}
else if (x == 'TampaBayBuccaneers') {
 value = 'TAM'
}
else if (x == 'WashingtonFootballTeam' | x == 'WashingtonRedskins') {
 value = 'WAS'
}
else if (x == 'BaltimoreRavens') {
 value = 'BAL'
}
else if (x == 'BuffaloBills') {
 value = 'BUF'
}
else if (x == 'CincinnatiBengals') {
 value = 'CIN'
}
else if (x == 'ClevelandBrowns') {
 value = 'CLE'
}
else if (x == 'DenverBroncos') {
 value = 'DEN'
}
else if (x == 'HoustonTexans') {
 value = 'HOU'
}
else if (x == 'IndianapolisColts') {
 value = 'IND'
}
else if (x == 'JacksonvilleJaguars') {
 value = 'JAX'
}
else if (x == 'KansasCityChiefs') {
 value = 'KAN'
}
else if (x == 'LasVegasRaiders') {
 value = 'LVR'
}
else if (x == 'LosAngelesChargers') {
 value = 'LAC'
}
else if (x == 'MiamiDolphins') {
 value = 'MIA'
}
else if (x == 'NewEnglandPatriots') {
 value = 'NEP'
}
```

```
    else if (x == 'NewYorkJets') {
     value = 'NYJ'
    }
    else if (x == 'PittsburghSteelers') {
     value = 'PIT'
    }
   else if (x == 'TennesseeTitans') {
     value = 'TEN'
   }
   else {
     value = 'FA'
   }
 return(value)
}

# Apply function to all datasets
playerdraftdata_cb$currentteam <- lapply(playerdraftdata_cb$currentteam,teamtoabb)
playerdraftdata_qb$currentteam <- lapply(playerdraftdata_qb$currentteam,teamtoabb)
playerdraftdata_de$currentteam <- lapply(playerdraftdata_de$currentteam,teamtoabb)
playerdraftdata_wr$currentteam <- lapply(playerdraftdata_wr$currentteam,teamtoabb)

# Set dataset secondcontract columns to NA
playerdraftdata_cb$secondcontract <- NA
playerdraftdata_qb$secondcontract <- NA
playerdraftdata_de$secondcontract <- NA
playerdraftdata_wr$secondcontract <- NA

# Create indicator function for whether someone got a second contract (take into ac
playerdraftdata_cb <-
  playerdraftdata_cb %>%
  mutate(secondcontract = ifelse(((currentteam == Tm) | (currentteam == 'LAC' & Tm

playerdraftdata_qb <-
  playerdraftdata_qb %>%
  mutate(secondcontract = ifelse(currentteam == Tm, 1, 0)) %>%
  mutate(secondcontract = ifelse(((currentteam == Tm) | (currentteam == 'LAC' & Tm

playerdraftdata_de <-
  playerdraftdata_de %>%
  mutate(secondcontract = ifelse(currentteam == Tm, 1, 0)) %>%
  mutate(secondcontract = ifelse(((currentteam == Tm) | (currentteam == 'LAC' & Tm

playerdraftdata_wr <-
  playerdraftdata_wr %>%
  mutate(secondcontract = ifelse(currentteam == Tm, 1, 0)) %>%
  mutate(secondcontract = ifelse(((currentteam == Tm) | (currentteam == 'LAC' & Tm
```

```
# Merge all datasets into master dataset
playerdraftdataA <- rbind(playerdraftdata_cb, playerdraftdata_qb)
playerdraftdataB <- rbind(playerdraftdataA, playerdraftdata_de)
playerdraftdata <- rbind(playerdraftdataB, playerdraftdata_wr)

colnames(playerdraftdata) <- c('player', 'draft_year', 'position', 'draft_rnd', 'dr
playerdraftdata <- playerdraftdata[c(1,2,3,4,5,10,11,6,7,8)]

# Clean the data before exporting
playerdraftdata <-
  playerdraftdata %>%
  arrange(draft_year, position, draft_rnd) %>%
  rowwise() %>%
  mutate(strengths = paste(strengths, collapse='␣'),
         weaknesses = paste(weaknesses, collapse='␣')) %>%
  ungroup()

# Export the data to a CSV
playerdraftdata <- apply(playerdraftdata,2,as.character)
#write.csv(playerdraftdata,"playerdraftdata.csv", row.names = FALSE)

# Load necessary packages
invisible(library(tidyverse))
invisible(library(tm))
invisible(library(SnowballC))
invisible(library(wordcloud))
invisible(library(RColorBrewer))
invisible(library(syuzhet))
invisible(library(ggplot2))
invisible(library(ISLR))
invisible(library(class))
invisible(library(RTextTools))
invisible(library(tokenizers))
invisible(library(dplyr))
invisible(library(tidytext))
invisible(library(randomForest))
invisible(library(rpart))
invisible(library(rpart.plot))
invisible(library(MASS))

# Read in the data
playerdraftdata <- read_csv("playerdraftdata.csv")
playerdraftdata <-
  playerdraftdata %>%
  filter(draft_rnd %in% c(1,2,3),
         draft_year %in% c(2013,2014,2015,2016,2017))
```

```
# Show percentage of second contract
second_contract <- sum(playerdraftdata$second_contract)/length(playerdraftdata$seco
second_contract

# Show percentage of each round and position second contracts
secondcontractper <-
  playerdraftdata %>%
  group_by(draft_rnd, position) %>%
  summarise(second_contract = sum(second_contract)/n())
secondcontractper

# Show percentage of each round second contracts
secondcontractper <-
  playerdraftdata %>%
  group_by(draft_rnd) %>%
  summarise(percentage = sum(second_contract)/length(second_contract))
secondcontractper

# Show percentage of each position second contracts
secondcontractper <-
  playerdraftdata %>%
  group_by(position) %>%
  summarise(percentage = sum(second_contract)/length(second_contract)) %>%
  arrange(desc(percentage))
secondcontractper

# Create different dataset for each position
playerdraftdata_qb <- playerdraftdata %>% filter(position == 'QB')
playerdraftdata_wr <- playerdraftdata %>% filter(position == 'WR')
playerdraftdata_cb <- playerdraftdata %>% filter(position == 'CB')
playerdraftdata_de <- playerdraftdata %>% filter(position == 'DE')

# Find most common 2 word phrases for each position's strengths variable
bigrams_qb <- playerdraftdata %>% unnest_tokens(bigram, strengths, token = "ngrams"

bigrams_wr <- playerdraftdata %>% unnest_tokens(bigram, strengths, token = "ngrams"

bigrams_de <- playerdraftdata %>% unnest_tokens(bigram, strengths, token = "ngrams"

bigrams_cb <- playerdraftdata %>% unnest_tokens(bigram, strengths, token = "ngrams"

# Filter out common stop words
bigrams_qb_separated <- bigrams_qb %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_qb_filtered <- bigrams_qb_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
```

```
# new bigram counts:
strengths_qb_list_1 <- bigrams_qb_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

strengths_qb_list_1$value <- paste(strengths_qb_list_1$word1, strengths_qb_list_1$w
strengths_qb_list <- strengths_qb_list_1 %>% dplyr::select(value)

bigrams_wr_separated <- bigrams_wr %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_wr_filtered <- bigrams_wr_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
strengths_wr_list_1 <- bigrams_wr_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

strengths_wr_list_1$value <- paste(strengths_wr_list_1$word1, strengths_wr_list_1$w
strengths_wr_list <- strengths_wr_list_1 %>% dplyr::select(value)

bigrams_cb_separated <- bigrams_cb %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_cb_filtered <- bigrams_cb_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
strengths_cb_list_1 <- bigrams_cb_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

strengths_cb_list_1$value <- paste(strengths_cb_list_1$word1, strengths_cb_list_1$w
strengths_cb_list <- strengths_cb_list_1 %>% dplyr::select(value)

bigrams_de_separated <- bigrams_de %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_de_filtered <- bigrams_de_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
strengths_de_list_1 <- bigrams_de_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

strengths_de_list_1$value <- paste(strengths_de_list_1$word1, strengths_de_list_1$w
strengths_de_list <- strengths_de_list_1 %>% dplyr::select(value)
```
```
```

```
```{r}
# Find most common 2 word phrases for each position’s weakness variable
bigrams_qb_2 <- playerdraftdata %>% unnest_tokens(bigram, weaknesses, token = "ngra

bigrams_wr_2 <- playerdraftdata %>% unnest_tokens(bigram, weaknesses, token = "ngra

bigrams_de_2 <- playerdraftdata %>% unnest_tokens(bigram, weaknesses, token = "ngra

bigrams_cb_2 <- playerdraftdata %>% unnest_tokens(bigram, weaknesses, token = "ngra

# Filter out common stop words
bigrams_qb_separated <- bigrams_qb_2 %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_qb_filtered <- bigrams_qb_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
weak_qb_list_1 <- bigrams_qb_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

weak_qb_list_1$value <- paste(weak_qb_list_1$word1, weak_qb_list_1$word2)
weak_qb_list <- weak_qb_list_1 %>% dplyr::select(value)

bigrams_wr_separated <- bigrams_wr_2 %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_wr_filtered <- bigrams_wr_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
weak_wr_list_1 <- bigrams_wr_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

weak_wr_list_1$value <- paste(weak_wr_list_1$word1, weak_wr_list_1$word2)
weak_wr_list <- weak_wr_list_1 %>% dplyr::select(value)

bigrams_cb_separated <- bigrams_cb_2 %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_cb_filtered <- bigrams_cb_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
weak_cb_list_1 <- bigrams_cb_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)
```
```

```
weak_cb_list_1$value <- paste(weak_cb_list_1$word1, weak_cb_list_1$word2)
weak_cb_list <- weak_cb_list_1 %>% dplyr::select(value)

bigrams_de_separated <- bigrams_de_2 %>%
  separate(bigram, c("word1", "word2"), sep = "␣")
bigrams_de_filtered <- bigrams_de_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
# new bigram counts:
weak_de_list_1 <- bigrams_de_filtered %>%
  count(word1, word2, sort = TRUE) %>%
  head(20)

weak_de_list_1$value <- paste(weak_de_list_1$word1, weak_de_list_1$word2)
weak_de_list <- weak_de_list_1 %>% dplyr::select(value)

# Develop a weakness score for all positions
weak_check_qb <- function (x) {
  value = 0
  for (i in weak_qb_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

weak_check_wr <- function (x) {
  value = 0
  for (i in weak_wr_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

weak_check_de <- function (x) {
  value = 0
  for (i in weak_de_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}
```

```
weak_check_cb <- function (x) {
  value = 0
  for (i in weak_cb_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

word_count <- function(x) {
  value = lengths(gregexpr("\\W+", x))
  return(value)
}

playerdraftdata_qb$weak_count <- lapply(playerdraftdata_qb$weaknesses,weak_check_qb
playerdraftdata_wr$weak_count <- lapply(playerdraftdata_wr$weaknesses,weak_check_wr
playerdraftdata_de$weak_count <- lapply(playerdraftdata_de$weaknesses,weak_check_de
playerdraftdata_cb$weak_count <- lapply(playerdraftdata_cb$weaknesses,weak_check_cb

playerdraftdata_qb$length <- lapply(playerdraftdata_qb$weaknesses, word_count)
playerdraftdata_wr$length <- lapply(playerdraftdata_wr$weaknesses, word_count)
playerdraftdata_de$length <- lapply(playerdraftdata_de$weaknesses, word_count)
playerdraftdata_cb$length <- lapply(playerdraftdata_cb$weaknesses, word_count)

# Develop a strengths score for all positions
strength_check_qb <- function (x) {
  value = 0
  for (i in strengths_qb_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

strength_check_wr <- function (x) {
  value = 0
  for (i in strengths_wr_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}
```

```
strength_check_de <- function (x) {
  value = 0
  for (i in strengths_de_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

strength_check_cb <- function (x) {
  value = 0
  for (i in strengths_cb_list$value) {
    if (grepl(i, x, fixed = TRUE) == TRUE) {
      value = value + 1
    }
  }
  return(value)
}

playerdraftdata_qb$strengths_count <- lapply(playerdraftdata_qb$strengths,strength_
playerdraftdata_wr$strengths_count <- lapply(playerdraftdata_wr$strengths,strength_
playerdraftdata_de$strengths_count <- lapply(playerdraftdata_de$strengths,strength_
playerdraftdata_cb$strengths_count <- lapply(playerdraftdata_cb$strengths,strength_

playerdraftdata_qb$length2 <- lapply(playerdraftdata_qb$strengths, word_count)
playerdraftdata_wr$length2 <- lapply(playerdraftdata_wr$strengths, word_count)
playerdraftdata_de$length2 <- lapply(playerdraftdata_de$strengths, word_count)
playerdraftdata_cb$length2 <- lapply(playerdraftdata_cb$strengths, word_count)

# Clean data
fulldataA <- rbind(playerdraftdata_qb, playerdraftdata_wr)
fulldataB <- rbind(fulldataA, playerdraftdata_de)
fulldata <- rbind(fulldataB, playerdraftdata_cb)

fulldata <-
  fulldata %>%
  mutate(strengths_count = as.numeric(strengths_count),
         weak_count = as.numeric(weak_count))

# Try a KNN Classifier
final_accuracy = 0
for (i in 1:100) {
  set.seed(i)

  split <- sample(1:nrow(fulldata), size = nrow(fulldata)*0.7)
  train <- fulldata[split,]
```

```
  test <- fulldata[-split,]

  train <- train %>% dplyr::select(strengths_count, weak_count, second_contract)
  test <- test %>% dplyr::select(strengths_count, weak_count, second_contract)

  knn <- knn(train = train, test = test, cl = train$second_contract, k = 3)

  test$predicted <- knn
  test <-
    test %>%
    mutate(accuracy = ifelse(second_contract == predicted, 1, 0))
  accuracy <- sum(test$accuracy)/length(test$accuracy)
  final_accuracy = final_accuracy + accuracy
  final_accuracy
}
final_accuracy/100

# Try fitting a decision tree
set.seed(1510)

split <- sample(1:nrow(fulldata), size = nrow(fulldata)*0.7)
train <- fulldata[split,]
test <- fulldata[-split,]

train <- train %>% dplyr::select(strengths_count, weak_count, position, second_cont
test <- test %>% dplyr::select(strengths_count, weak_count, position, second_contra

tree <- rpart(second_contract~., data = train, method = 'class')
rpart.plot(tree, extra = 106)

# Predictions from decision tree
predictions <-predict(tree, test, type = 'class')
confusion_matrix <- table(test$second_contract, predictions)
confusion_matrix

sum(diag(confusion_matrix)/sum(confusion_matrix))

# Try fitting a glm
set.seed(6)

split <- sample(1:nrow(fulldata), size = nrow(fulldata)*0.7)
train <- fulldata[split,]
test <- fulldata[-split,]

glm <- stepAIC(glm(second_contract ~ (strengths_count + weak_count + as.factor(posi
summary(glm)
confint(glm)
```

```
# Predictions from decision tree
test$model_prob <- predict(glm, test, type = "response")
test <-
  test %>%
  mutate(model_pred = 1*(model_prob > .5) + 0)
test <-
  test %>%
  mutate(accurate = 1*(model_pred == second_contract))
sum(test$accurate)/nrow(test)

glm <- glm(second_contract ~ (strengths_count + weak_count + as.factor(position))^2
summary(glm)

secondcontractper <-
  playerdraftdata %>%
  group_by(position) %>%
  summarise(number = n()) %>%
  arrange(desc(number))
secondcontractper

fulldata %>%
  group_by(position) %>%
  summarise(strengths = sum(second_contract)/n())

strength <-
  fulldata %>%
  group_by(strengths_count) %>%
  summarise(proportion = sum(second_contract)/n())

strength %>%
  ggplot(aes(x = strengths_count, y = proportion)) +
  geom_line() +
  geom_point() +
  ggtitle('Proportions of Second Contract by Strengths Count')

weak <-
  fulldata %>%
  group_by(weak_count) %>%
  summarise(proportion = sum(second_contract)/n())

weak %>%
  ggplot(aes(x = weak_count, y = proportion)) +
  geom_line() +
  geom_point() +
  ggtitle('Proportions of Second Contract by Weaknesses Count')
```

```
willis_strengths <- c('Powerful␣arm␣␣Can␣make␣every␣throw␣␣Easy␣arm␣strength␣to␣pus

pickett_strengths <- c('Accurate␣passer
Tremendous␣pocket␣composure
Excellent␣decision␣maker
Fits␣passes␣into␣tight␣windows
Superb␣ball␣placement
Throws␣a␣very␣catchable␣ball
Can␣throw␣receivers␣open
Beats␣good␣coverage␣with␣accuracy,␣placement
Excellent␣timing
Leads␣receivers␣for␣more␣yardage␣after␣the␣catch
Able␣to␣loft␣in␣touch␣passes
Advanced␣field␣vision
Moves␣eyes␣through␣progressions
Quality␣arm
Can␣push␣the␣ball␣vertically
Phenomenal␣deep-ball␣accuracy
Can␣fire␣fastballs␣into␣tight␣windows
Good␣internal␣clock
Mastered␣his␣offense
Ball␣security
Mobility
Difficult␣to␣sack
Keeps␣his␣eyes␣downfield␣while␣scrambling
Can␣hurt␣defenses␣on␣the␣ground
Dangerous␣to␣pick␣up␣yards␣on␣the␣ground
Good␣height
Thick␣build␣for␣the␣next␣level
Good␣fit␣for␣a␣west␣coast␣offense
Rhythm␣thrower
Intangibles
Leadership
Student␣of␣the␣game
Leadership␣personality
Great␣preparation␣skills
Tough;␣plays␣injured
Hard␣worker')

ridder_strengths <- c('Good␣arm␣strength
Can␣make␣beautiful␣throws␣downfield
Can␣make␣all␣the␣throws␣required
Throws␣a␣catchable␣ball
Can␣pick␣up␣yards␣on␣the␣ground
Has␣some␣pocket␣presence
Can␣make␣superb␣throws␣off␣platform
Flashes␣tremendous␣accuracy␣on␣some␣throws
```

```
Good␣ball␣placement␣at␣times
Can␣throw␣touch␣passes
Mobility
Can␣hurt␣defenses␣on␣the␣ground
Difficult␣to␣sack
Athletic
Upside')

corral_strengths <- c('Accurate␣passer
Vastly␣improved␣decision␣maker␣in␣2021
Quality␣ball␣placement
Throws␣a␣very␣catchable␣ball
Can␣throw␣receivers␣open
Beats␣good␣coverage␣with␣accuracy,␣placement
Has␣feel;␣shows␣some␣passing␣instincts
Able␣to␣loft␣in␣touch␣passes
Advanced␣field␣vision
Moves␣eyes␣through␣progressions
Quality␣arm
Can␣push␣the␣ball␣vertically
Can␣fire␣fastballs␣into␣tight␣windows
Good␣internal␣clock
Mastered␣his␣offense
Ball␣security␣was␣very␣improved␣in␣2021
Mobility
Dual-threat␣to␣hurt␣defenses␣on␣the␣ground
Dangerous␣to␣pick␣up␣yards␣on␣the␣ground
Will␣juke␣defenders␣in␣the␣open␣field
Can␣be␣difficult␣to␣sack
Tough;␣plays␣injured
Said␣to␣be␣a␣hard␣worker␣in␣2021')

howell_strengths <- c('Play-maker
Gunslinger␣attitude
Instincts
Accurate␣passer␣with␣good␣ball␣placement
Throws␣receivers␣open
Excellent␣touch␣passer
Throws␣a␣very␣catchable␣ball
Deep-ball␣accuracy
Natural␣feel␣as␣a␣passer
Superb␣ball␣placement␣to␣lead␣receivers␣for␣yards␣after␣the␣catch
Throws␣with␣good␣timing
Mobility
Escapability␣to␣extend␣plays
Difficult␣to␣sack
Good␣at␣throwing␣on␣the␣run
```

```
Can␣pick␣up␣yards␣on␣the␣ground
Enough␣arm␣to␣make␣all␣the␣throws␣in␣the␣NFL
Leadership␣potential
Intelligent')

purdy_strengths <- c('Play-maker
Gunslinger␣attitude
Instincts
Accurate␣passer␣with␣good␣ball␣placement
Throws␣receivers␣open
Excellent␣touch␣passer
Throws␣a␣very␣catchable␣ball
Deep-ball␣accuracy
Natural␣feel␣as␣a␣passer
Superb␣ball␣placement␣to␣lead␣receivers␣for␣yards␣after␣the␣catch
Throws␣with␣good␣timing
Mobility
Escapability␣to␣extend␣plays
Difficult␣to␣sack
Good␣at␣throwing␣on␣the␣run
Can␣pick␣up␣yards␣on␣the␣ground
Enough␣arm␣to␣make␣all␣the␣throws␣in␣the␣NFL
Leadership␣potential
Intelligent')

strength_check_qb(corral_strengths)
strength_check_qb(pickett_strengths)
strength_check_qb(ridder_strengths)
strength_check_qb(willis_strengths)
strength_check_qb(howell_strengths)
strength_check_qb(purdy_strengths)
```

Bibliography

[1] Nicholas J. Horton Benjamin S. Baumer, Daniel T. Kaplan. *Modern Data Science with R*. Chapman and Hall, 2017.

[2] Walter Cherepinsky. 2022 nfl draft scouting reports. `https://walterfootball.com/scoutingreports.php`, 1999.

[3] Sean Forman. Drafted players. `https://www.pro-football-reference.com`, 2012.

[4] David Robinson Julia Silge. *Text Mining with R*. Creative Commons, 2016.

[5] NFL Enterprises LLC. Follow players. `https://www.nfl.com/players`, 2022.

Academic Vita

DREW INSLEY
dki5019@psu.edu

EDUCATION

The Pennsylvania State University, University Park, PA — **Expected Graduation:** May 2022
Schreyer Honors College, Eberly College of Science
Bachelor of Science in Statistics, *Actuarial Option* — Minor in Mathematics

PROFESSIONAL EXAMS/VEE

- Exam SRM (Statistics for Risk Modeling) — Passed: January 2022
- VEE Accounting and Finance — Passed: June 2021
- VEE Economics — Passed: June 2021
- VEE Mathematical Statistics — Passed: June 2021
- Exam FM (Financial Mathematics) — Passed: December 2020
- Exam P (Probability) — Passed: July 2020

TECHNICAL SKILLS

- An aptitude towards data wrangling, data visualization, statistical analysis/modeling, & machine learning
- Software: Microsoft Excel, Minitab
- Languages: R, Python, SAS, VBA, MATLAB

WORK EXPERIENCE

Actuarial Analyst – Philadelphia Health — May 2022 – Present
Milliman, Wayne, PA

- Draft, revise, and finalize client correspondence

- Assist Consultants in the development, checking, and utilization of actuarial models and tools in Microsoft Excel
- Project work on Medicare Part D pricing as part of CMS's Medicare bidding process

Actuarial Intern – Philadelphia Health May 2021 – August 2021
Milliman, Wayne, PA

- Assist Analysts and Consultants in the development, checking, and utilization of actuarial models and tools in Microsoft Excel
- Project work on the development of Python code in DataBricks to upload, process, and summarize Medicare member data

Intern – Broadband Navigation Systems (BBNS) January 2020 – August 2020
Applied Research Lab (ARL), University Park, PA

- Gather and analyze BBNS data
- Project work on determining correlation between data parameters in a univariate sense, and covariance between data parameters in a multivariate sense

Learning Assistant (LA) - Calculus with Analytic Geometry II August 2019 – December 2019
Wartik Lab, University Park, PA

- Attend a science pedagogy class for training
- Facilitate in-class small group interaction
- Address student difficulties with course content

Intern - Inertial Navigation Systems (INS) May 2019 – July 2019
Applied Research Lab (ARL), Warminster, PA

- Gather, analyze, and simulate INS error data
- Project work on setting INS performance thresholds to balance buyer/seller risk
- Presentations on Gaussian Copula method of simulation

HONORS & ACCOLADES

- Member of the Mu Sigma Rho Honor Society April 2021 – Present
- Member of the Phi Kappa Phi Honor Society March 2021 - Present
- Member of the Phi Beta Kappa Honor Society October 2020 – Present
- Penn State Dean's List Fall 2018 – Fall 2021
- The Penn State President's Freshman Award March 2019