The Finances of a Winning Football Team

Spencer Ratermann

2022-12-2

Abstract:

For all 32 teams in the National Football League a common goal is shared among them, winning football games. How teams go about this mission differ from a team to team basis. Each team uses different methods of scouting, drafting, signing, and most importantly assembling a roster of 53 players of varying positions in order to achieve this goal. What is the best way to assemble a roster since different positions are more or less important than others? Which players are worth bringing on more of a financial c ommitment? With all teams having the same salary cap that is not allowed to be breached, what is the ideal way to allocate funds to an NFL roster in order to maximize a teams number of wins, and eventually make the playoffs and compete for a opportunity to win a championship? Based on the multiple different models that were created and analyzed, percentage of salary cap used is more important then looking at the overall dollars spent. It also looks as if the most telling models include the amount of money spent on active players and dead cap. However, the other models provide some clarity on how to spend money appropriately with looking at which phase of the game is the most important, in comparison to the actual position group. Lastly, looking at salary cap data it has a better ability to predict ten win teams as opposed to predicting the actual number of wins.

Introduction:

The world of professional sports provides an interesting dive into how statistics can play a role in multiple facets of the game. Whether it is tracking a player's performance or predicting the number of wins that a team achieves, statistical techniques are now being used by owners and general managers in all types of sports. A particular area in which techniques found in data science can be beneficial is in the team salary of the assembled roster. For some sports like baseball and basketball, all teams are not on a level playing field. This is because both the National Basketball Association and Major League Baseball have implemented a soft salary cap as opposed to a hard one. With a soft salary cap teams are allowed to breach the maximum salary cap if they are willing to pay a series of fines. This makes teams in big television markets like New York or Los Angeles have a significant advantage over teams in smaller markets because they are willing to pay those fines since they have the money to do so. However, in sports like football and hockey, the National Football League and National Hockey League implement hard salary caps, that when breached set off a series of punishments to the team until the team salary is under the maximum number. This makes a level playing field for all teams regardless of television market and city size. The question is if everyone can spend the same amount of money, why are certain teams more dominant than others, and why are teams consistently losing decades at a time?

The distribution of player salaries in the National Football League is an interesting topic to look at because of the number of players on the roster, as well as the massive difference in importance of players and the positions they play. For example, a Quarterback in the NFL is always going to be one of the highest paid players on the team because of the importance they play to the team, however a kicker is one of the least paid players but is usually the team's leading scorer, with the top 41 all time scoring leaders being kickers and 91 of the top 100 per pro football reference. The amount a player is valued is heavily based on the position that they play, and there are players in the league who make millions of dollars and never touch the ball their entire career. Add in the factor of a hard salary cap which creates equal spending for all, an NFL roster becomes a complicated puzzle of where do you invest your money.

This conundrum has only increased within the last 11 years in the NFL. With the evolution of offenses and defenses, certain positions that were once deemed the most important are now losing the financial backing that they once had. The question that now exists is "How do teams properly distribute finances t o certain position groups in order to maximize the number of wins?" Does it become more valuable to spend big large sums of money for flashy positions like running backs and receivers, or under the radar stars like offensive and defensive linemen? For this answer breaking down positions by grouping is important. These groups are Quarterbacks, Running Backs, Wide Receivers, Offensive Line, Defensive Line, Linebackers, Defensive Backs, and Special Teams Players like punters, kickers, and long snappers. With a set number of 53 players per NFL roster, looking at total amount of money spent per group as well as number of players in each group becomes relevant. The last hurdle to jump over is that the set salary cap number fluctuates c ompared to years past. With 2010 being the last uncapped year in the NFL, 2011 brought around a set spending limit of approximately 120 millions dollars, while the cap number for last season, 2021, was 182.5 million, an 11 year increase of 62.5 millions dollars. This means it is not possible to look at how much total salary each position group is making, but instead looking at the percentage of the salary cap each position group takes up.

The salary cap is also broken into different areas where the money might fall. The teams Cap Total is the maximum amount of money that an organization is allowed to spend, but this money can be spent in different ways. In an ideal world an NFL team would like to spend every dollar of the salary cap on active players that contribute to a teams on the field success. In reality, this will never be the case, players get injured and traded, while their salaries remain there for teams to pay while the player is not performing on the field. These dollars that are spent fall into the category of Dead cap, which is what every NFL team is trying to avoid. The three factors that all NFL team owners and general managers are looking at is how much money are we spending on the Active Cap, players on the field contributing, Dead cap, those who aren't impacting the outcome on the field, and Cap Space left after all salaries are paid.

The last contributing factor to a teams salary cap is that teams are allowed to transfer salary cap space between seasons. The hard cap that we are considering is a base number that can be increased and decreased based on the previous season. This means that in theory teams are allowed to go over the allotted cap space for a season if they believe that their team has the potential to bring an organization and city a championship. However, the amount that they do go over is then subtracted from the hard cap total set by the league the next season. This is the same case for money that is not spent. This unused millions of dollars can then be added on to a teams spending for the next season which makes NFL owners and General Managers have to play a game of do I want to spend all of my money to compete right now, or do I want to save it and build to have future success and more money. Taking all of these factors in to account is just the beginning of creating a successful team.

What is a successful team? Some fan bases might classify a successful season as a championship and everything below that is a failure, while others might be satisfied with just having more wins then the previous season. In general an NFL team has had a successful season if they have the opportunity to play in the playoffs, and compete for the Lombardi trophy at the Super Bowl at the end of the season. What makes being a playoff team so difficult in the NFL is that out of the 4 major sports in the United States (MLB, NFL, NBA, and NHL) it has the second lowest number of playoff teams. In the NFL only 14 of the 32 teams make the playoffs, which is only 43.75%. In both the NHL and NBA 50% of teams make the playoffs, with the MLB being the hardest with only 40% of teams making it. The one common factor between playoffs teams is that they consistently are the teams that have won 10 regular season games out of the 16 now 17 games completed throughout the season. In the past 11 years there have been 134 teams that have made the playoffs, with 118 of them having won 10+ games. This is 88% of the playoff teams having eclipsed double digit wins. Out of all 10+ win teams in the past 11 years only 5 of them have failed to qualify for the postseason, with these teams being in 2020, 2015, 2014, 2013, and 2012. One thing to consider is that last year, 2021, was the first year in which 14 teams made the playoffs as opposed to the traditional 12. It is safe to say that a team is successful if they achieve 10+ regular season wins and earn a birth to the playoffs, which traditionally run hand in hand.

Using data acquired from spotrac compiling lists of team's players, salaries, positions, and cap hits allows for analysis on how to produce the most financially efficient team based on the 352 rosters assembled in the last 11 years. With data acquired from every team's roster from 2011-2021 taking a dive into the new era of football becomes accessible, and predicting win totals based on their salaries becomes a viable option. Along with each team's active players salaries, a second data set has been created with the same teams from 2011-2021, their total salary cap, active cap spending, and dead cap. With this second data set it makes it possible to see how much teams should budget for salaries that do not produce results, and to see if there is an element of luck when it comes to producing winning teams based on injuries and trades. Lastly, a compiled data set of each teams wins, losses, and ties will provide the backbone of the analysis in order to predict success in the future. Assembling the perfect roster is not easy, but maybe looking at how to distribute salaries and manage injury money spent can lead to competitive football teams.

Methods:

In order to create data that is in the proper form to run statistical models a series of packages are needed, and used to complete these tests as well as complete the extensive amount of data cleaning in the model.

The Data

There was not a readily available data set that contains all of the information for the roster of each team, as well as their salary cap hit and percentage used up. This is where data mining played a huge role in acquiring data that can be molded to be used for proper analysis. Once the data is acquired, packages like tidyverse were used to perform data cleaning, with the final step of using inner joins in order to merge data sets based on the name of the NFL team and the year that they competed.

Data Mining

NFL Active Player Salaries The first of the raw data sets that was created via data mining contained the salary information of each player that has played in the NFL from 2011-2021. In order to create this data set the information had to be acquired from spotrac.com, and since it was not all in one spreadsheet it was grabbing the right information and adding it to a spreadsheet that contained all teams in all years looked at for the analysis. For this each team had to be individually gone through for each year, copying the information of their rosters one at a time. In the end there ended up being 18,416 observations with 14 variables in NFL_Active_Player_Salaries. These variables included ACTIVE PLAYER (56), POS., BASE SALARY, SIGNING BONUS, ROSTER BONUS, OPTION BONUS, WORKOUT BONUS, RESTRUC. BONUS, MISC., DEAD CAP, CAP HIT, CAP % , YEAR , and TEAM. This data set would be able to provide the base for new data sets to be created, as well as the backbone to any phase and positional analysis that will be performed. It is important to note that these are the salaries of only active players, and does not include those who contribute to the dead cap space.

NFL Cap Hits For the next data set that will be used for the analysis, it is important to also find all of the information regarding the salary cap for each team for 2011-2021. This includes finding information about each teams cap breakdown so it is possible to look at a teams Salary Cap, Dead Cap, Active Cap, and Cap room left at the end of the year. Once again, spotrac.com had all of this information but not on one spreadsheet. Once again each team was individually gone through in order to grab those numbers and put them in one singular raw data set to then be manipulated and used in the analysis. In the end the data set contained 352 observations with 8 variables in the NFL_Cap_Hits_ data set. These variables included TEAM, SIGNED, AVE AGE, ACTIVE, DEAD, TOTAL CAP, CAP SPACE (ALL), and YEAR. This data set will provide a base to perform analysis to see if the amount of active cap, dead cap, and remaining cap space play a role in projection wins.

NFL Records The last of the raw data sets created was simply a data set that contained the records of all the teams from 2011-2021. This information was publically available on NFL.com and just like the previous two data sets needed to be pieced together into one data set in order to have all of the records for all of the teams in one location. In the end there ended up being 352 observations with 8 variables in the NFL_Records data set. These variables included NFL Team, W, L, T, PCT, PF, PA, Year. PCT is winning percentage, PF is point for, and PA is points against and these variables were included because they could be beneficial for future analysis. This data set will be combined with every final data set used because wins is the variable that we are trying to predict throughout the whole analysis.

Data Cleaning

Using Tidyverse all of the raw data sets created were implemented into R. After that a series of data cleaning techniques were needed to be used in order to have the data in a form that was usable, and readable for the analysis. All code used to complete this data cleaning is included in Appendix A.

NFL Active Player Salaries Since the original raw data set came from a website there was a lot of work to do because everything came in as a character variable, when the purpose of using salary amounts is to have numeric variables. The first obstacle is that the raw data contained dollar signs ($) in the observations, which would not allow for a variable to be numeric. Using a gsub those dollar signs were removed and then all variables that needed to be numeric where changed. After tis transition to having the proper numeric variables there werre some that are unnecessary for the analysis. Using the select function the variables of ACTIVE PLAYER (56), POS, BASE_SALARY, CAP_HIT, CAP_%, YEAR, and TEAM were kept creating a cleaned data set of 7 variables instead of 14. The variables were also renamed in order to make using the variables easier. The (56) was removed from active players and all of the variables were converted to only have the first letter of each word be capitalized instead of all the letters. With all of this being done the NFL_ACTIVE_PLAYERS_SALARIES data set was ready to be used.

NFL Cap Hits For the NFL Cap Hits data set there were a lot of similar procedures that needed to be completed as the previous one. Just as before all of the variables were characters with dollar signs ($) in them which could not be used as numeric variables. The exact same technique was used to remove that dollar sign and make the variables ACTIVE, DEAD, TOTAL CAP, and CAP SPACE (ALL) numeric. The variables were also renamed to have only the first letter be capitalized and the (ALL) was removed from the variables CAP SPACE (ALL).

NFL Records NFL Records was an easy data cleaning because it was just placing a period in between NFL_TEAM instead of an underscore. With all there of the raw data sets cleaned it was just a matter of combining data set and final touches to have all of the data ready to go.

Merged Data Sets

Once again using tidyverse, data sets were merged and mutated to create the final data set that is to be used for data exploration and statistical analysis. The first thing that needed to be done is that there needed to be a way that all of the variables that needed to bee used in the analysis to be in one data set, however there needed to be some manipulation before that could be completed. The first thing looked at was the salary cap hits, whether is was the ACTIVE, DEAD, or CAP SPACE. Since the salary cap is always changing it seemed like it was important to have the percentage of the salary cap used by these groups be included in the analysis as well as the physical dollar amount. A simple mutate was used in order to take the ACTIVE, DEAD, and CAP SPace divided by the CAP TOTAL and then multiplied by 100 in order to create three new variables ACTIVEPercentage, DEADpercentage, Cap_SpacePercentage. Theoretically if the ACTIVE cap took up 70% of the total cap the number would be 70.00. After the percentage calculation were complete, the dollar amount spent, so ACTIVE, DEAD, Cap_Space where multiplied by 0.000001 in order to display 1,500,000 as 1.5 since that is the way those numbers are refered to in conversation. This dataset was then inner_joined with NFL_Records so that the number of wins were also linked to the informations about each teams salary cap. An example of the new SalaryCap data set is previewed below.

```
##                    TEAM       ACTIVE         DEAD    Total_Cap    Cap_Space YEAR
## 1 Jacksonville Jaguars 1.269054e-10 2.362507e-11 0.0001804733 2.849689e-11 2021
## 2  Philadelphia Eagles 1.016110e-10 6.376952e-11 0.0001891815 1.781732e-11 2021
## 3       Denver Broncos 1.072694e-10 3.714469e-11 0.0001883438 1.246547e-11 2021
## 4  Pittsburgh Steelers 1.241641e-10 3.038002e-11 0.0001759071 1.154292e-11 2021
## 5     Seattle Seahawks 1.379560e-10 1.392165e-11 0.0001727031 1.122811e-11 2021
## 6     Cleveland Browns 1.264739e-10 2.900308e-11 0.0002024413 9.468418e-12 2021
##   W  L T   PCT  PF  PA ACTIVEPercent DEADPercent Cap_SpacePercent
## 1 3 14 0 0.176 253 457      70.31814   13.090617        15.790087
## 2 9  8 0 0.529 444 385      53.71086   33.708127         9.418113
## 3 7 10 0 0.412 335 322      56.95402   19.721745         6.618464
## 4 9  7 1 0.559 343 398      70.58504   17.270486         6.561943
## 5 7 10 0 0.412 395 366      79.88043    8.061027         6.501394
## 6 8  9 0 0.471 349 371      62.47436   14.326663         4.677118
```

Table 1 Breakdown of Salary Cap Space Used in Millions and by Percentage of Salary Cap Consumed by Team and Year

The next data set that was created was from the NFL_Active_Player_Salaries. This new creation was looking at players not by postion, but by phase of the game. The game of football is broken into three phases, Offense, Defense, and Special Teams. Before taking a look into which positions are important it is important to look at which phases are important. Since the current NFL_Active_Players_Salaries are only looked at by position and group_by was needed to also include which phase of the game they participated in. The Offense consisted of the position groups quarterback (QB), running back (RB), wide reciever (WR), tight end (TE), tackle (T), left tackle (LT), offensive guard (G), and center (C). The Defensive phase included defensive end (DE), defensive tackle (DT), linebacker (LB), outside linebacker (OLB), inside linebacker (ILB), cornerback (CB), defensive back (DB), safety (S), and free safety (FS). Lastly the special teams phase consisted of the punter (P), kicker (K), and long snapper (LS). Now that each player had their phase of the game it was a matter of using a group_by in conjunction with a summarize to create the offensive, defensive, and special teams spending per team. The data was filtered by defense first, and then it was grouped by both team and year. After that a summarize command was piped in that included the functions to calculate the total amount spent per phase and well as the cap percentage. Appropriate conversions where made just like the first data set created in order to make the data more readable. This process was then completed with the other two phases and then an inner join was used to combine the three and create a SalariesPhases which is previewed below data set which is show below.

```
##   YEAR               TEAM OffensePercentage OffenseCap_Hit DefensePercentage
## 1 2011  Arizona Cardinals             49.88       61.04117             37.77
## 2 2011    Atlanta Falcons             54.22       65.26251             40.12
## 3 2011      Buffalo Bills             19.62       27.38810             29.17
## 4 2011  Carolina Panthers             53.98       66.48903             38.21
## 5 2011      Chicago Bears             33.70       43.19246             40.13
## 6 2011 Cincinnati Bengals             30.34       41.05799             42.44
##   DefenseCap_Hit STPercentage STCap_Hit
## 1       46.21270         3.09  3.775000
## 2       48.30053         2.60  3.121450
## 3       40.73367         1.54  2.155000
## 4       47.07111         4.18  5.157291
## 5       51.44250         4.76  6.101666
## 6       57.44272         1.37  1.858750
```

Table 2 Salaries by Millions Spent and Cap Percentage by Team and Year for the Offensive, Defensive and Special Teams Phases

The last of the new data sets that needed to be created was the one that contained all of the pertinent information relating to each position group on the field. In the original data players were broken down to have 26 different positions which seemed like a number a little bit to large. For that reason players were broken into the categories of quarterback (QB), running back (RB), wide receiver (WR), offensive line (OL), defensive line (DL), linebackers (LB), defensive backs (DB), and tight ends (TE). Just like when looking at the phases data set creation a combination of filtering by position, grouping by team and year, and summarizing to create dollar cap hit and percentage of cap hit multiple different data sets were created with all of the positions groups. Inner join all of these position group data sets together and a completed data set named SalariesPositionGroups is created with the breakdown of dollars spent and cap percentage per position group, which is previewed below.

```
##   YEAR               TEAM OLPercentage OLCap_Hit DLPercentage DLCap_Hit
## 1 2011  Arizona Cardinals         9.90 12.126666         9.10  11.14097
## 2 2011    Atlanta Falcons         9.08 10.918886         8.37  10.06350
## 3 2011      Buffalo Bills         3.61  5.059125        13.61  19.02517
## 4 2011  Carolina Panthers        16.09 19.825426        12.90  15.87902
## 5 2011      Chicago Bears        12.17 15.601816        11.51  14.75925
## 6 2011 Cincinnati Bengals        10.20 13.787112        13.51  18.28287
##   QBPercentage QBCap_Hit RBPercentage RBCap_Hit WRPercentage WRCap_Hit
## 1         8.50 10.407912         3.01  3.677500        16.74 20.490760
## 2        13.54 16.300000         7.75  9.327750        11.19 13.482763
## 3         5.81  8.110000         2.73  3.806316         4.74  6.612801
## 4         4.44  5.467136         8.88 10.929764        10.91 13.439200
## 5         7.75  9.917215         2.90  3.713333         7.45  9.557053
## 6         2.18  2.948036         3.59  4.867730         4.96  6.721236
##   TEPercentage TECap_Hit DBPercentage DBCap_Hit LBPercentage LBCap_Hit
## 1         3.68  4.491606        18.24  22.30558        10.43 12.766149
## 2         8.04  9.664375        13.15  15.83505        18.60 22.401982
## 3         1.21  1.680882         8.67  12.09779         6.89  9.610708
## 4         5.24  6.452500        12.09  14.89921        13.22 16.292880
## 5         2.87  3.683986        13.19  16.90346        15.43 19.779796
## 6         2.29  3.101250        18.92  25.61279        10.01 13.547061
```

Table 3 Salaries by Millions Spent and Cap Percentage by Team and Year for the Position Groups

Finally to complete all of the data cleaning and data transformation all of these newly created data set were then all joined together. All of them had the variables Team and Year which then gave a teams who salary picture whether it was looking at the salary cap spending, the breakdown by position, and the breakdown by phase of the game. In total our final data set consisted of 282 observations with 39 unique variables in order to try and best predict the number of wins a team might have. A preview of the data set is seen below.

```
##   X.1 YEAR               TEAM OffensePercentage OffenseCap_Hit
## 1   1 2011  Arizona Cardinals             49.88       61.04117
## 2   2 2011    Atlanta Falcons             54.22       65.26251
## 3   3 2011      Buffalo Bills             19.62       27.38810
## 4   4 2011  Carolina Panthers             53.98       66.48903
## 5   5 2011      Chicago Bears             33.70       43.19246
## 6   6 2011 Cincinnati Bengals             30.34       41.05799
##   DefensePercentage DefenseCap_Hit STPercentage STCap_Hit OLPercentage
## 1             37.77       46.21270         3.09  3.775000         9.90
## 2             40.12       48.30053         2.60  3.121450         9.08
## 3             29.17       40.73367         1.54  2.155000         3.61
## 4             38.21       47.07111         4.18  5.157291        16.09
## 5             40.13       51.44250         4.76  6.101666        12.17
## 6             42.44       57.44272         1.37  1.858750        10.20
##   OLCap_Hit DLPercentage DLCap_Hit QBPercentage QBCap_Hit RBPercentage
## 1 12.126666         9.10  11.14097         8.50 10.407912         3.01
## 2 10.918886         8.37  10.06350        13.54 16.300000         7.75
## 3  5.059125        13.61  19.02517         5.81  8.110000         2.73
## 4 19.825426        12.90  15.87902         4.44  5.467136         8.88
## 5 15.601816        11.51  14.75925         7.75  9.917215         2.90
## 6 13.787112        13.51  18.28287         2.18  2.948036         3.59
##   RBCap_Hit WRPercentage WRCap_Hit TEPercentage TECap_Hit DBPercentage
## 1  3.677500        16.74 20.490760         3.68  4.491606        18.24
## 2  9.327750        11.19 13.482763         8.04  9.664375        13.15
## 3  3.806316         4.74  6.612801         1.21  1.680882         8.67
## 4 10.929764        10.91 13.439200         5.24  6.452500        12.09
## 5  3.713333         7.45  9.557053         2.87  3.683986        13.19
## 6  4.867730         4.96  6.721236         2.29  3.101250        18.92
##   DBCap_Hit LBPercentage LBCap_Hit    ACTIVE      DEAD Total_Cap Cap_Space  W
## 1  22.30558        10.43 12.766149 111.02887  4.742656  118.3820  3.993036  8
## 2  15.83505        18.60 22.401982 116.68449  3.908834  125.6439 -5.268878 10
## 3  12.09779         6.89  9.610708  70.27677 12.240729  108.3129 31.352071  6
## 4  14.89921        13.22 16.292880 118.71743  3.339553  123.8330 -0.657984  6
## 5  16.90346        15.43 19.779796 100.73663  7.020013  107.7566 20.358357  8
## 6  25.61279        10.01 13.547061 100.35946  4.501309  108.2490 27.125995  9
##    L T   PCT  PF  PA ACTIVEPercent DEADPercent Cap_SpacePercent TenPlus
## 1  8 0 0.500 312 348      93.78867    4.006232        3.3730104       0
## 2  6 0 0.625 402 350      92.86922    3.111042       -4.1935016       1
## 3 10 0 0.375 372 434      64.88309   11.301263       28.9458251       0
## 4 10 0 0.375 406 429      95.86899    2.696820       -0.5313479       0
## 5  8 0 0.500 353 341      93.48531    6.514692       18.8929020       0
## 6  7 0 0.563 344 323      92.71167    4.158291       25.0588862       0
```

Table 4 Completed Data set

Data Exploration

Using ggplot, within the tidyverse library, a series of visualizations were then created in order to explore the data and give grounds to the analysis and see if there are any trends that do appear within these graphs.

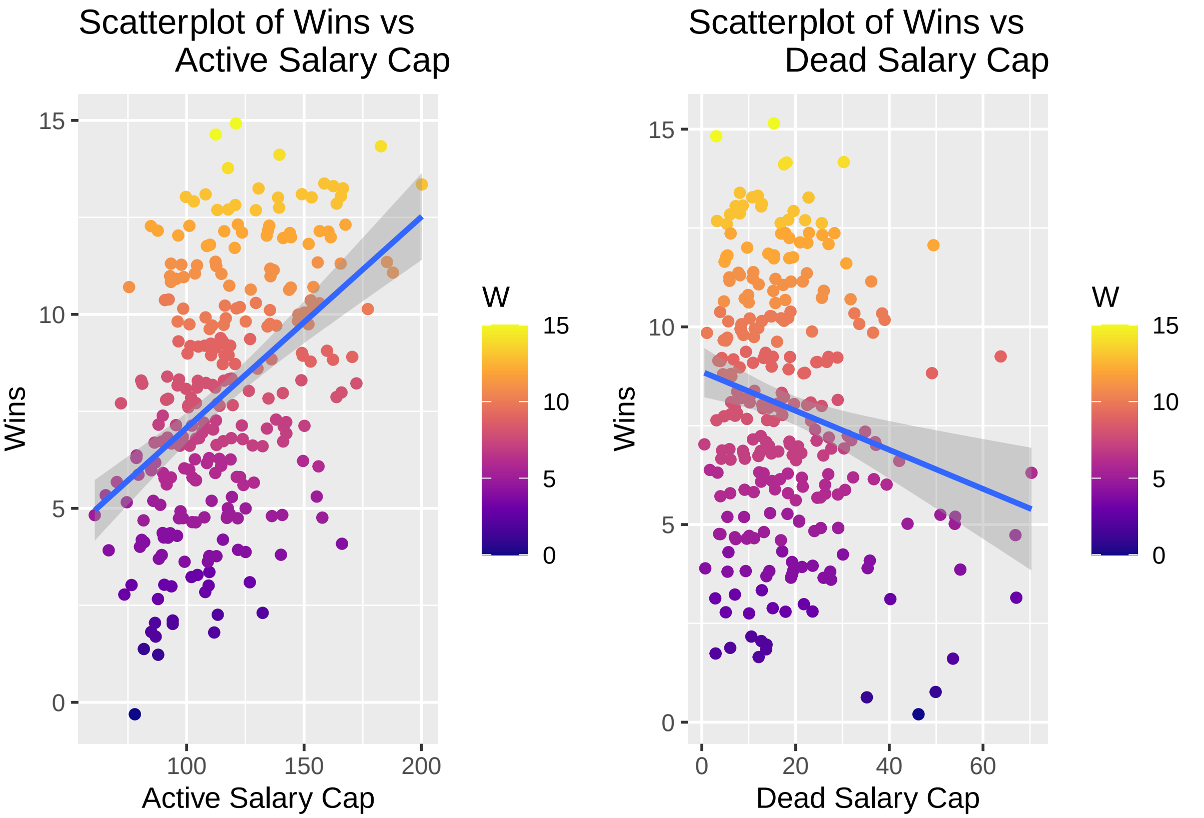


Figure 1 Scatterplot of Wins vs Active Cap and Wins vs Dead Cap

As shown by the figure above it is worth then taking a look at Active and Dead cap to see how they affect the number of wins a franchise may have that year because the more money a team spends on active players the more likey they are to win, on the dead cap could be interesting to look at because the regression line shows that there is a slight negative trend, but there is a large bunch of teams that had 0-30 million dollars in dead cap spent, with some of those teams winning up to 15 games while others only winning 2 or 3.

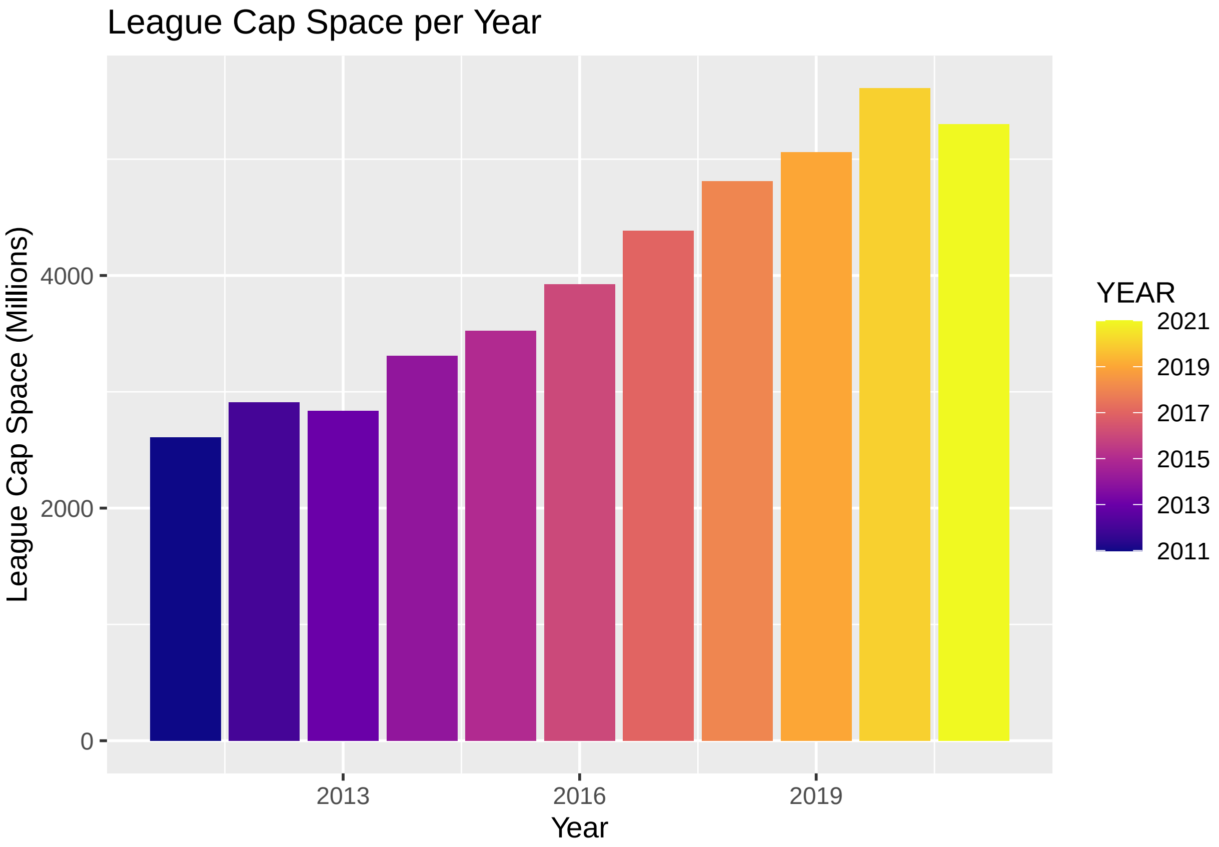


Figure 2 NFL Combined Cap Space per year for the last 11 Years

As shown by figure two dollars may not be the best way at estimating the total number of wins because of the constant change in the amount of money spent per year from 2011, 2.7 billion, to 2021, 5.2 billion, the NFL league salary cap has almost doubled meaning a large salary in 2011 would only be an average one in 2021. Another reason to only consider the cap percent being used as opposed to the dollars spend can be found in the spending per phase of the game.

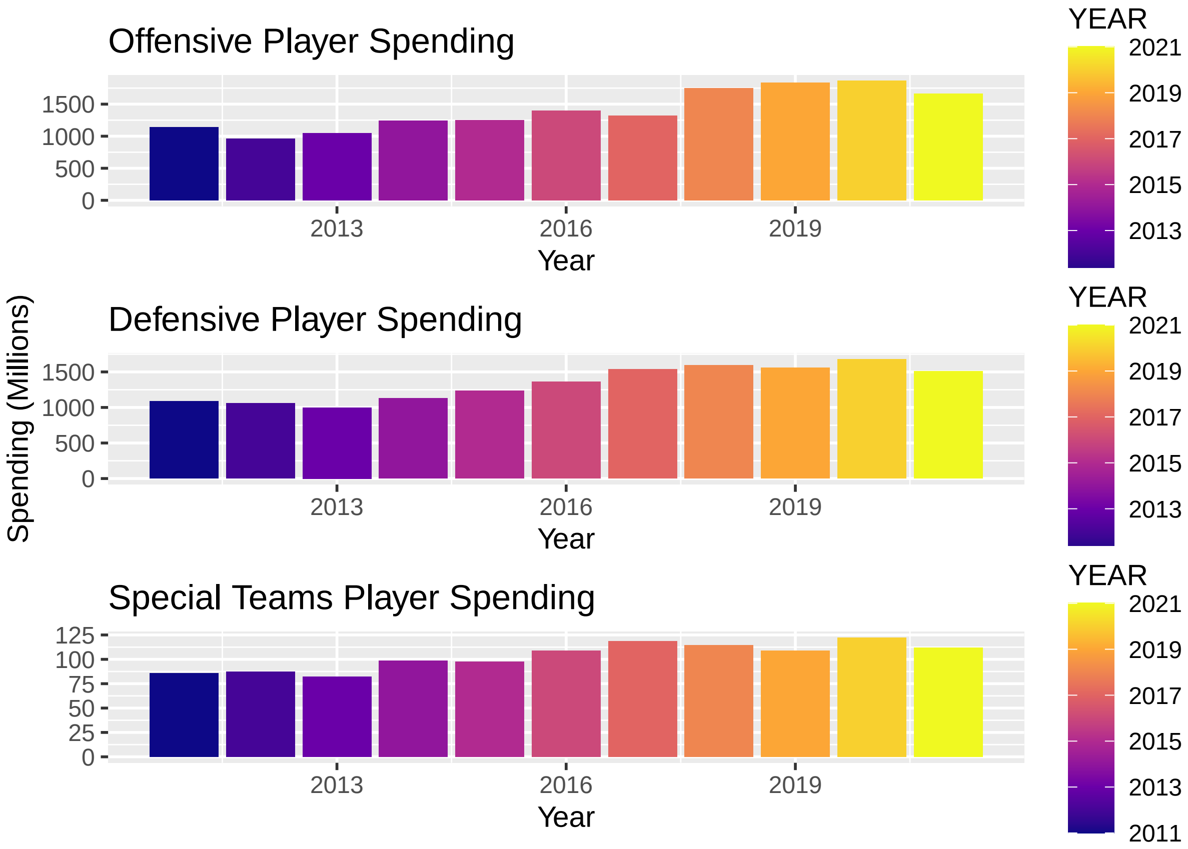


Figure 3 Spending per Phase of the Game by Year for all NFL Teams in Millions

Just like the salary cap increasing, the amount of money that is being spent on players per phase of the game is also increasing as the years go by. All of these graphs show that the use of cap percentage would be better to use to take into account the constant change in the salary cap.

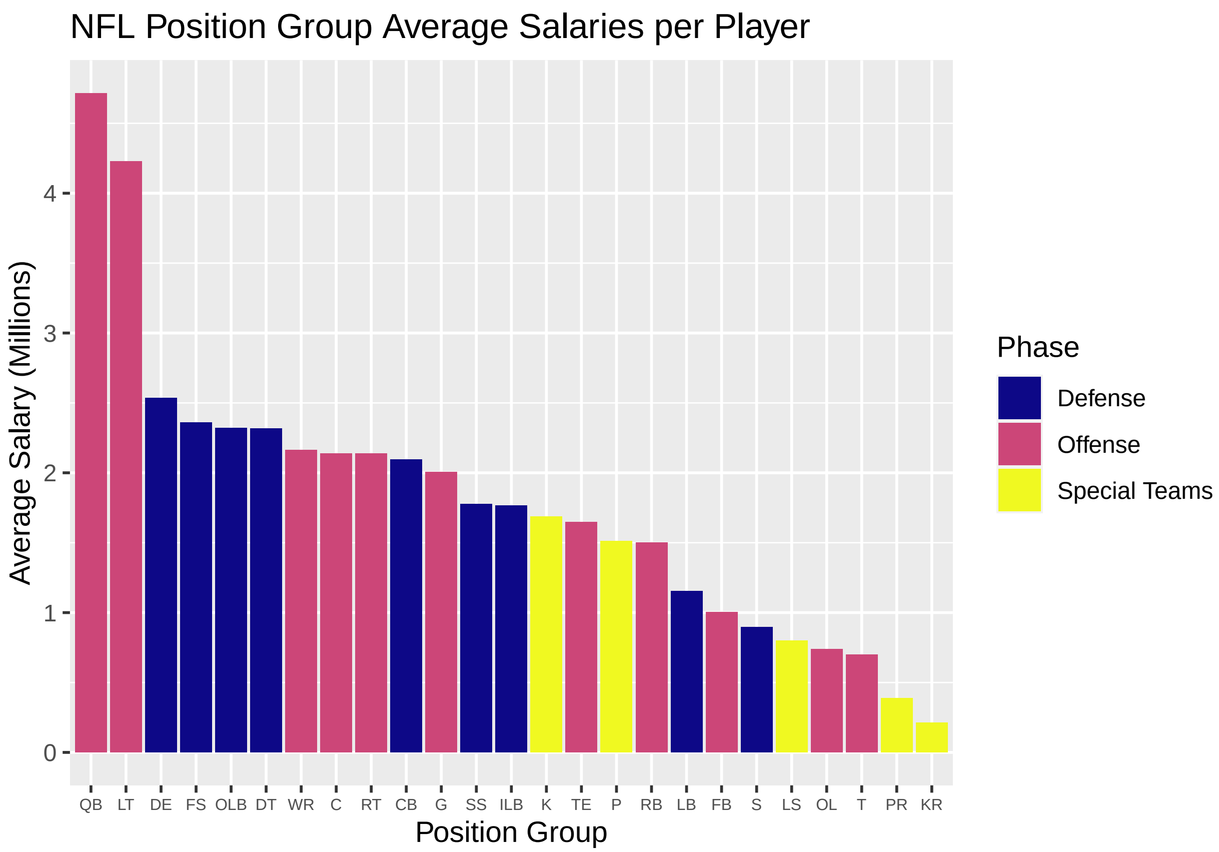


Figure 4 Average Salary in Millions of player position groups

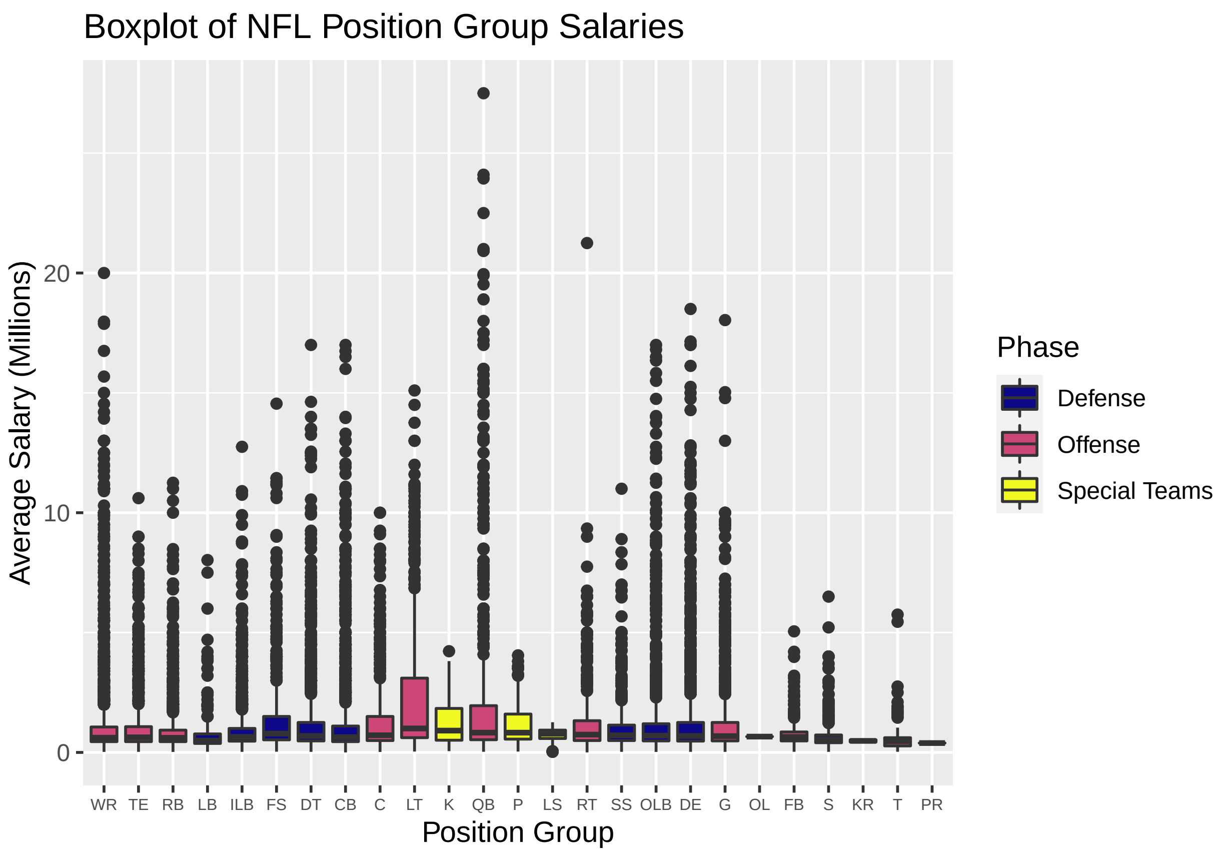


Figure 5 Boxplot of Average Salaries by position group

The last look at salaries that were considered is just looking at the positions that are traditionally paid the highest on average, as well as the highest overall salaries. Positions like Quarterback, right tackle, and wide receiver warrant higher average and individual salaries, while special teams players, linebackers, and running backs look to warrant less amount of money in their contracts.

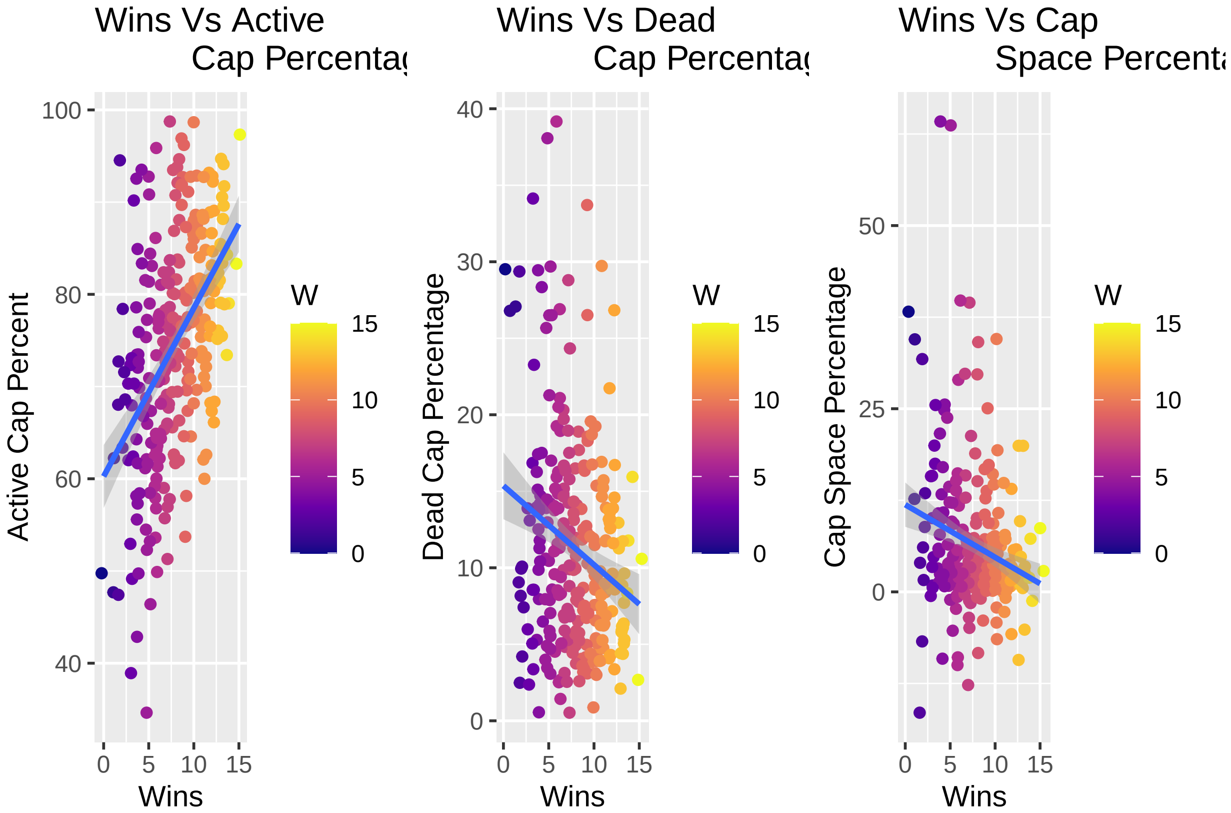


Figure 6 Scatter Plot of Wins versus Active Cap Percentage, Dead Cap Percentage, and Cap Space Percentage

Based on the figure 6 when using percentages there is still a trend that teams tend to win more games when they spend more money, while teams tend to wins less games when they have dead salary cap percentage and cap space left in their budget.

```
##   DefenseCapSpending OffenseCapSpending STCapSpending
## 1           33.35982           35.10287      2.512482
```

Table 5 Average Cap Percentages per Phase

```
##   OLCapSpending DLCapSpending QBCapSpending RBCapSpending WRCapSpending
## 1      8.899113      12.07145      7.605957       3.46656      7.869823
##   TECapSpending LBCapSpending DBCapSpending
## 1      3.359504      9.462837      11.82553
```

Table 6 Average Cap Percentages per Position Group

Tables 5 and 6 display the average cap percentages for all teams over the 11 year span in question based on phase of the game that is played and position groups.

Statistical Tests

In order to see which phases, positions, and percentages are important to the overall success of a football team a series of statistical analysis needs to be completed. The models that were created use a series of statistical techniques that allow for further analysis of the ability to predict wins. These techniques include linear regression, CART Trees, and Random Forests.

Linear Regression

Using base R, multiple regression is used in order to try and predict the number of wins based on a series of salary cap percentages. For linear regression models numeric variables are being used to predict another variable to see how changing one variable impacts the overall outcome, which in this case is the projected number of wins. These model included predicting wins by using Active Cap Percentage, Dead Cap Percentage, and Cap Space. A model that predicts wins by using the Cap percentages of all of the phases of the game. A model that uses all of the cap percentages based on position groups.

CART Trees

Using the packages rpart and rpart.plot a series of CART Trees were created. These trees are created from the same multiple regression models that were used. These CART Trees take variables that are the most important and places a set number that the variable needs to be higher or lower than. For example the top of the tree could have Active Cap with a set number of 50. If the observation has a value higher than 50 they would go one way and have a win projection, while if it goes left for being less than there would be a different win projection. This will continue to go down the tree until they reach the bottom.

Random Forests

Using the randomForest and Caret packages a series of Random forests will be used in order to predict the number of wins and if a team becomes a ten win team or not. A random forest is versatile in the sense that they can be used for numeric and categorical models. A random forest will run a variety of decision trees in order to be able to predict the overall outcome of the situation. There will be a random forests that predicts the number of wins, as well as if a team wins ten games or more. There will be models that use salary cap percentage, phase percentage, and salary cap percentage.

Results:

```
##
## Call:
## lm(formula = W ~ ACTIVE + DEAD + Cap_Space, data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.5979 -1.9344 -0.0138  2.0516  6.7964
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.164870   0.844248   3.749 0.000216 ***
## ACTIVE       0.049499   0.006419   7.711 2.24e-13 ***
## DEAD        -0.031458   0.013193  -2.384 0.017777 *
## Cap_Space   -0.042767   0.013309  -3.213 0.001466 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.728 on 278 degrees of freedom
## Multiple R-squared:  0.2488, Adjusted R-squared:  0.2407
## F-statistic:  30.7 on 3 and 278 DF,  p-value: < 2.2e-16
```

Figure 7 Linear Model of Wins Predicted by Active, Dead, and Unused Salary in Millions

We can see that the model has an adjusted R-Squared value of 0.2389, meaning that the model currently explains 23.89% of the variation in wins. For each coefficient we can see how the adjustment of 1 million dollars changes the amount of projected wins. Both Dead cap and cap space left have a negative impact on the projected number of wins, while the amount of money spent on active players positively impacts projected wins. For each million spent in Active player cap spending the team can expect to have and additional 0.04821 wins, while each million spent on dead cap subtract 0.03334 wins and cap space left 0.04686 respectably.

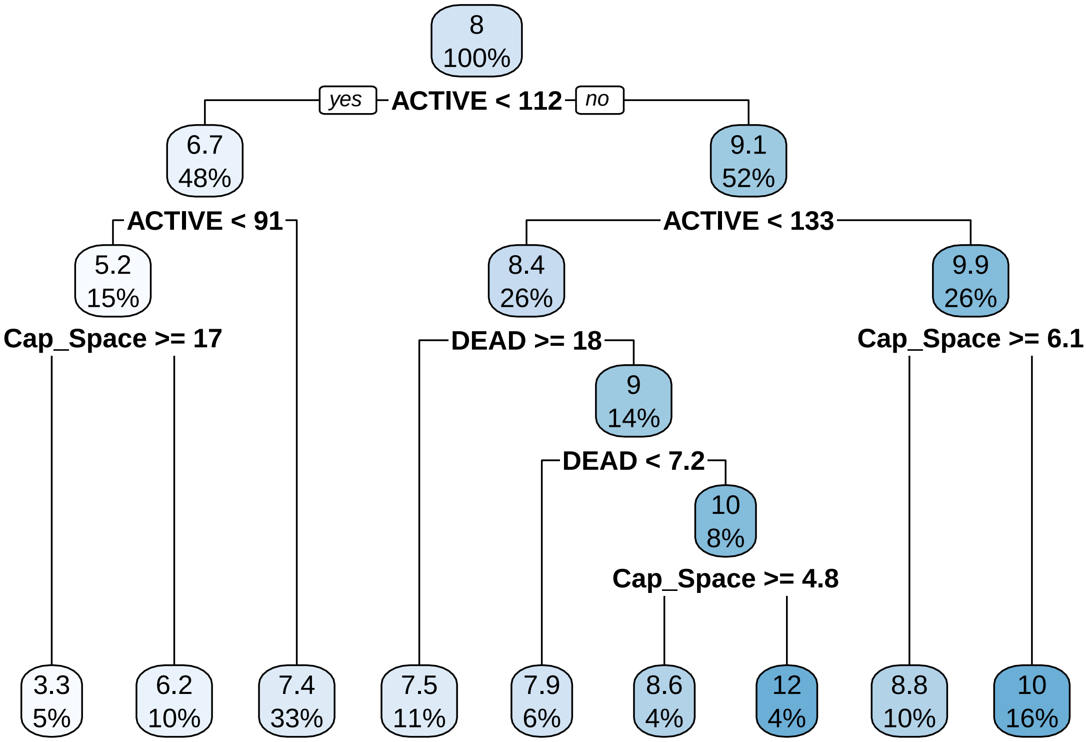


Figure 8 CART TREE of Wins Predicted by Active, Dead, and Unused Salary in Millions

Based on the CART Tree above, using the same categories in the regression model that includes all three variables, the number of wins can be predicted based on these categories. Right at the top of the tree we can see that of all of the teams in the last year, 86% have spent more than 91 million dollars on players that were active for the year and have averaged 8.6 wins. In simple terms, spending more money on players that actually play in the games will equal more wins on the field, which is what anyone would expect. The next node of the tree then shows spending more than 129 million dollars on players who are actively on the field are on average going to win 1.8 more games then those who don't, and 29% of teams over the last ten years have spent this much money on active players. Once we get to the third node that is when cap space left comes into play. teams that have less than or equal to 5.4 million dollars left on average will go on to win 11 games while those who don't on average go on to win 8.8 games which is a 2.2 win difference.

```
##
## Call:
## lm(formula = W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent,
##     data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5045 -1.7169  0.0428  1.8182  5.4516
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.74617    1.59082  -2.983  0.00310 **
## ACTIVEPercent     0.16224    0.01783   9.101  < 2e-16 ***
## DEADPercent       0.09694    0.03081   3.146  0.00183 **
## Cap_SpacePercent -0.08239    0.01630  -5.054 7.86e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.623 on 278 degrees of freedom
## Multiple R-squared:  0.3053, Adjusted R-squared:  0.2978
## F-statistic: 40.73 on 3 and 278 DF,  p-value: < 2.2e-16
```

Figure 9 Linear Model of Wins Predicted by Active, Dead, and Unused Salary in Percentage

on the same model as before but taking into consideration the cap percent as opposed to dollar spending we can see that there is a big change. When taking a look at Dead Cap percentage it affects the projected number of wins inversely. The greater the dead cap percentage is actually is equivalent to getting more projected wins. For each 1 percent of the cap taken up by dead cap a team can expect 0.09601 wins. For active cap percent for each 1 percent the projected number of wins is 0.16322. Finally for each percentage point of cap space left a team is projected to have -0.08714 wins. The model has an adjusted r-squared value of 0.3005, meaning that it explains 30.05% of the variation in wins which is greater than the previous model with dollars.

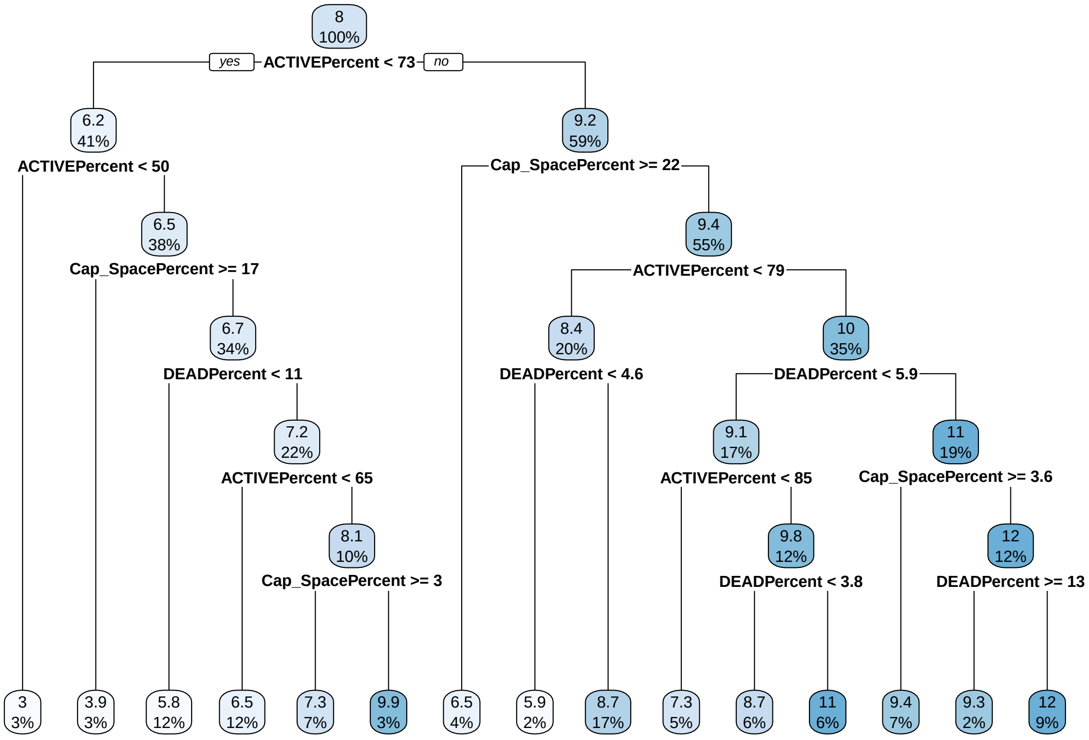


Figure 10 CART of Wins Predicted by Active, Dead, and Unused Salary in Percentages

Based on the CART Tree of the percentages model, it is clear to see that there are multiple different ways achieve double digit wins in a season based on the distribution of the salary cap. Ultimately, the most important factor is that if greater than 73% of the salary cap is taken up by Active players teams on average have 3 more wins then those that don't. This is a difference of 9.3 wins versus 6.3 wins. Only 3% of all teams in the last 10 years have finished with double digit wins while having less than 73% of the salary cap being taken up by Active players. Spending money on players on the active roster is the most important factor that contributes to projected wins, while DEAD cap percent is actually the least influential factor, with Total cap space left being more impact.

```
##
## Call:
## lm(formula = W ~ OffenseCap_Hit + DefenseCap_Hit + STCap_Hit,
##     data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9600 -2.0270  0.0427  1.9499  7.5670
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.79966    0.88600   0.903  0.36761
## OffenseCap_Hit  0.04691    0.01133   4.139 4.74e-05 ***
## DefenseCap_Hit  0.05588    0.01272   4.394 1.63e-05 ***
## STCap_Hit       0.31297    0.09644   3.245  0.00133 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.812 on 256 degrees of freedom
##   (22 observations deleted due to missingness)
## Multiple R-squared:  0.2015, Adjusted R-squared:  0.1921
## F-statistic: 21.53 on 3 and 256 DF,  p-value: 1.821e-12
```

Figure 11 Linear Model of Wins Predicted by Money Spent per Phase in Millions

Based on the model above, there is an adjusted r-squared value of 0.1921 or 19.21 percent of the variation in wins is explained. The big take away from the model is that defensive spending looks to increase the number of wins by more than the Offensive spending. For each million dollars spent on defense the team is projected 0.05588 wins and for each million spent on offense the team is projected 0.04691 wins. The other take away is that at first look special teams looks to influence wins more than any other phase of the game, but as seen below there is not as much money spent on special teams, as opposed to both offense and defense. Both offense and defense spend upwards of 50 millions dollars and special teams only have around 4 million dollars spent on.

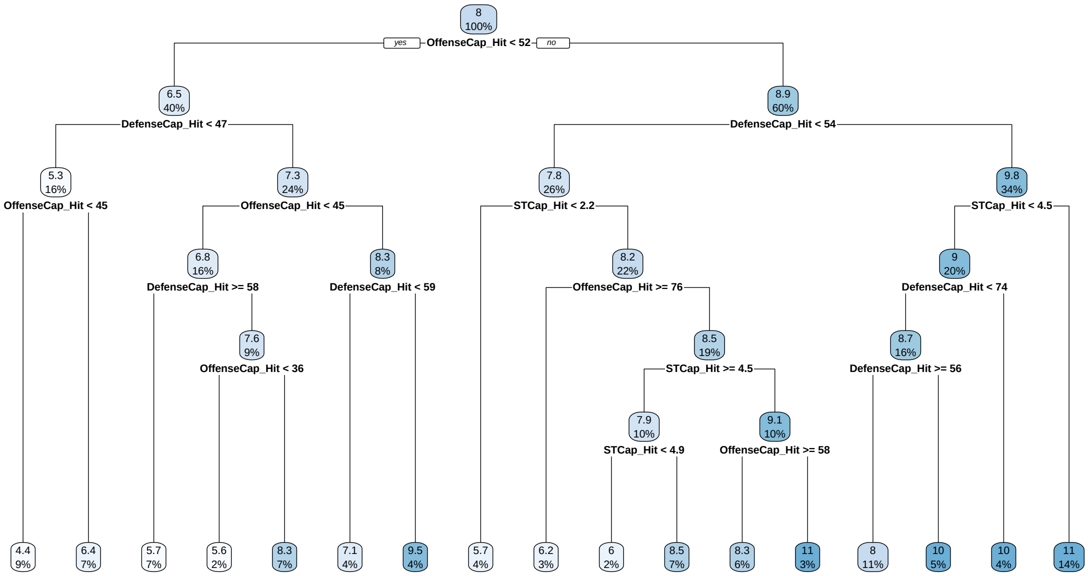


Figure 12 CART Tree of Wins Predicted by Money Spent per Phase in Millions Based on the CART Tree above it shows that the most important variable when it comes to predicting wins is the Offensive Cap hit when looking at that phase of the game. For teams that spend more than 62 million of their cap on the offense they are projected to win 8.9 games, while those who do not are projected to only have 6.5 wins. After that the defensive cap plays a strong role and in both cases on the second layer will affect the number of wins by 2. This number for both sides of the tree is the second highest factor when it comes to the predicted number of wins.

```
##
## Call:
## lm(formula = W ~ OffensePercentage + DefensePercentage + STPercentage,
##     data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8273 -1.7685  0.0592  1.9049  5.6778
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.07633    0.91793  -1.173   0.2420
## OffensePercentage  0.13342    0.01917   6.960 2.45e-11 ***
## DefensePercentage  0.11095    0.02017   5.500 8.60e-08 ***
## STPercentage       0.26397    0.14170   1.863   0.0635 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.693 on 278 degrees of freedom
## Multiple R-squared:  0.2679, Adjusted R-squared:   0.26
## F-statistic: 33.91 on 3 and 278 DF,  p-value: < 2.2e-16


##
## Call:
## lm(formula = W ~ OffensePercentage + DefensePercentage + STPercentage,
##     data = SalariesFinal)
##
## Coefficients:
##       (Intercept)  OffensePercentage  DefensePercentage       STPercentage
##           -1.0763             0.1334             0.1109             0.2640
```

Figure 13 Linear Model of Wins Predicted by Money Spent per Phase in Percentages

For the linear model that takes into account the phases of the game the adjusted R-squared is 0.26, so 26% of the variance in wins is predicted by the model. The model however is misleading because the Special Teams variable is the most influential, but teams need less special teams players and spend less money on them. However, based on the percentages the offensive percentage provides 0.133 wins and defensive percentage provides 0.111 wins per 1% spent. The percentages model predicts Offense as more important, while the dollars model predicts deffensive as more important. It also looks like the special teams money spent is not significant, probably since it is only a fraction of the cost of the total roster.

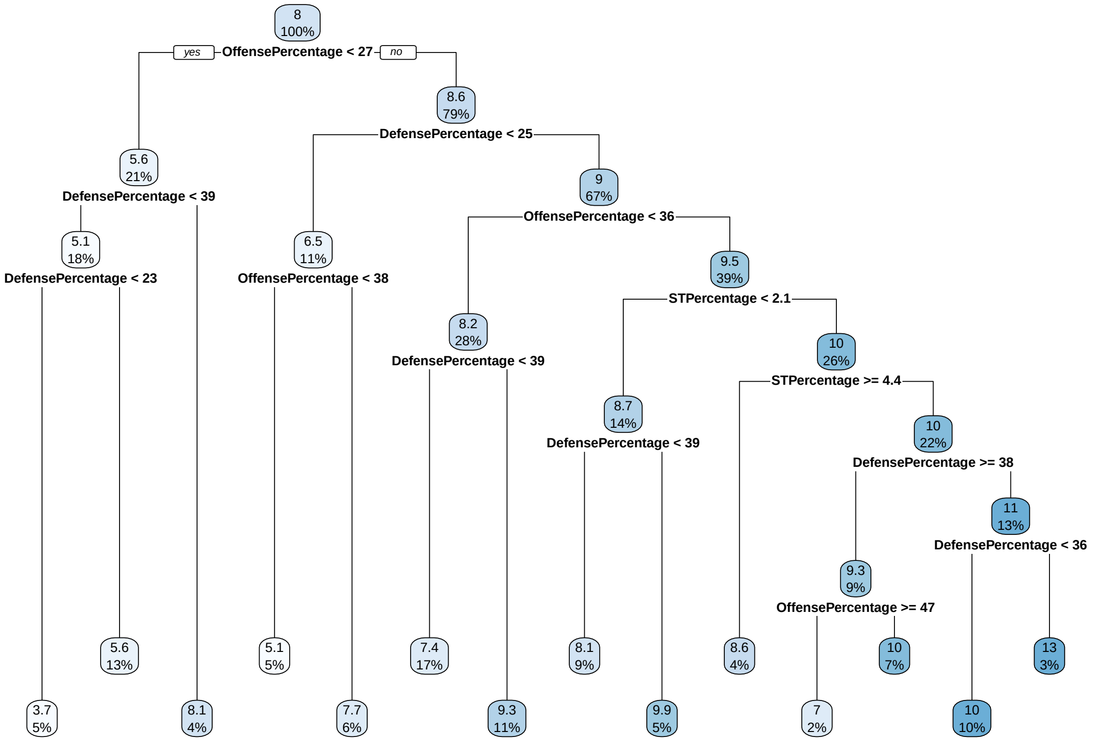


Figure 14 CART Tree of Wins Predicted by Money Spent per Phase in Percentages

Based on the CART Tree above it shows that the most important variable when it comes to predicting wins is the Offensive Cap hit when looking at that phase of the game. For teams that spend more than 27 percent of their cap on the offense they are projected to win 8.6 games, while those who do not are projected to only have 5.6 wins, a difference of 3 wins. After that the defensive cap plays a strong role and in both cases on the second layer will affect the number of wins by 2.5 to 3. This model also shows the insignificance of including Special Teams because it never showed up in the CART Tree.

```
##
## Call:
## lm(formula = W ~ OLCap_Hit + DLCap_Hit + QBCap_Hit + RBCap_Hit +
##     WRCap_Hit + TECap_Hit + LBCap_Hit + DBCap_Hit, data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7242 -2.0085 -0.1155  1.8983  7.1962
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.97361    0.86112   2.292 0.022736 *
## OLCap_Hit    0.02221    0.02805   0.792 0.429046
## DLCap_Hit    0.02661    0.01894   1.405 0.161306
## QBCap_Hit    0.07531    0.02215   3.400 0.000783 ***
## RBCap_Hit    0.06767    0.05108   1.325 0.186422
## WRCap_Hit    0.04824    0.02600   1.855 0.064743 .
## TECap_Hit    0.06533    0.05285   1.236 0.217545
## LBCap_Hit    0.07549    0.02362   3.196 0.001574 **
## DBCap_Hit    0.07981    0.02190   3.644 0.000326 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.861 on 252 degrees of freedom
##   (21 observations deleted due to missingness)
## Multiple R-squared:  0.1893, Adjusted R-squared:  0.1635
## F-statistic: 7.354 on 8 and 252 DF,  p-value: 8.659e-09
```

Figure 15 Linear Model of Wins Predicted by Money Spent per Position Group in Millions

Based on the linear model that predicts the number of wins based on the amount of money spent, in millions, on position groups, it is seen that only three of the variables are significant, which are Quarterback cap hit, Line backer cap hit, and defensive back cap hit. This model also does not do as well as the other models in predicting the amount of wins, with an adjusted r-squared of only 0.1635, meaning only 16.35% of the variance in wins is explained by the other variables. Next, looking at the percentages was completed.

```
##
## Call:
## lm(formula = W ~ OLPercentage + DLPercentage + QBPercentage +
##     RBPercentage + WRPercentage + TEPercentage + LBPercentage +
##     DBPercentage, data = SalariesFinal)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -1.7132 -0.0035  2.0072  6.4182
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.28138    0.89477  -0.314 0.753404
## OLPercentage  0.12651    0.04527   2.794 0.005567 **
## DLPercentage  0.09041    0.02987   3.027 0.002708 **
## QBPercentage  0.17120    0.03425   4.999 1.03e-06 ***
## RBPercentage  0.01509    0.07079   0.213 0.831405
## WRPercentage  0.13525    0.04006   3.376 0.000842 ***
## TEPercentage  0.17463    0.08116   2.152 0.032287 *
## LBPercentage  0.10457    0.03360   3.112 0.002056 **
## DBPercentage  0.17258    0.03558   4.851 2.07e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.719 on 273 degrees of freedom
## Multiple R-squared:  0.267,  Adjusted R-squared:  0.2455
## F-statistic: 12.43 on 8 and 273 DF,  p-value: 3.487e-15
```

Figure 16 Linear Model of Wins Predicted by Money Spent per Position Group in Percentages

When looking at the percentage of salary cap used for all of the position groups, with the subtraction of special teams, there is an adjusted r-squared value of 24.55% meaning that almost a quarter of the variation in wins can be described by position groups and how much of the salary cap is used on them.

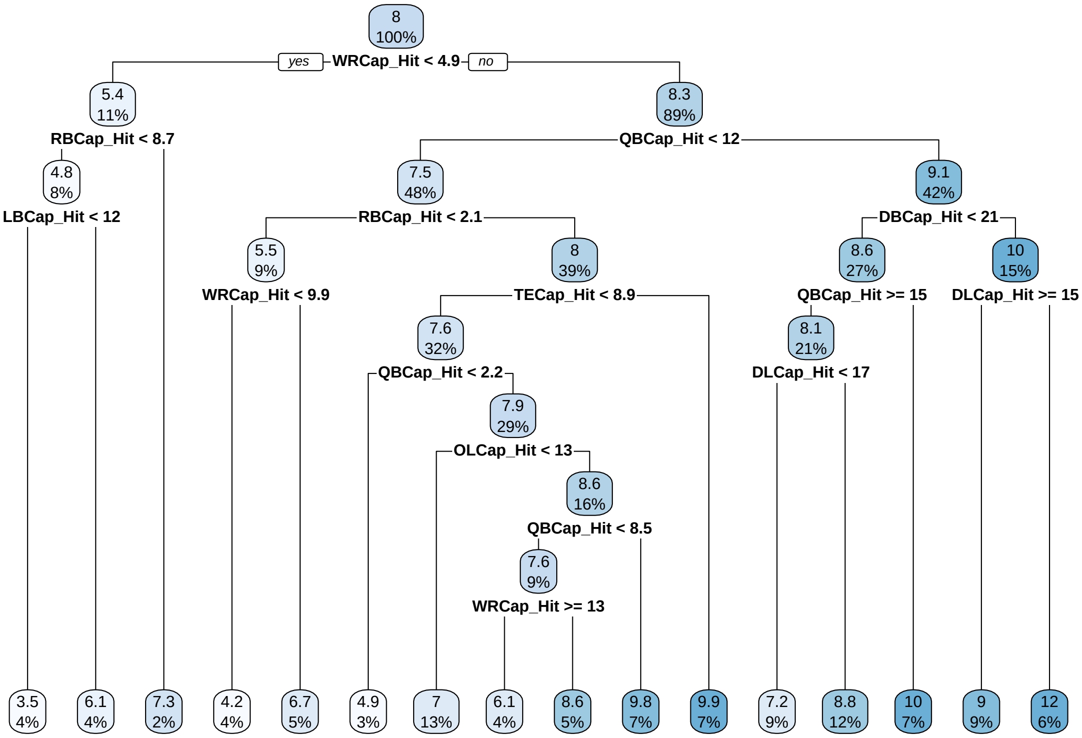


Figure 17 CART Tree of Wins Predicted by Money Spent per Position Group in Millions

Based on the CART Tree above, looking at the physical amount of dollars spent it is shown that the most important variable to look at is spending more than 4.9 million dollars on wide receiver salaries. This can take your wins from an average of 5.4 to an average of 8.3 Next is looking at the quarterback position where out of those teams that average 8.3 wins if they spend more than 12 million dollars on their quarterback they will average 9.1 wins as opposed to the 7.5 wins of those who do not.

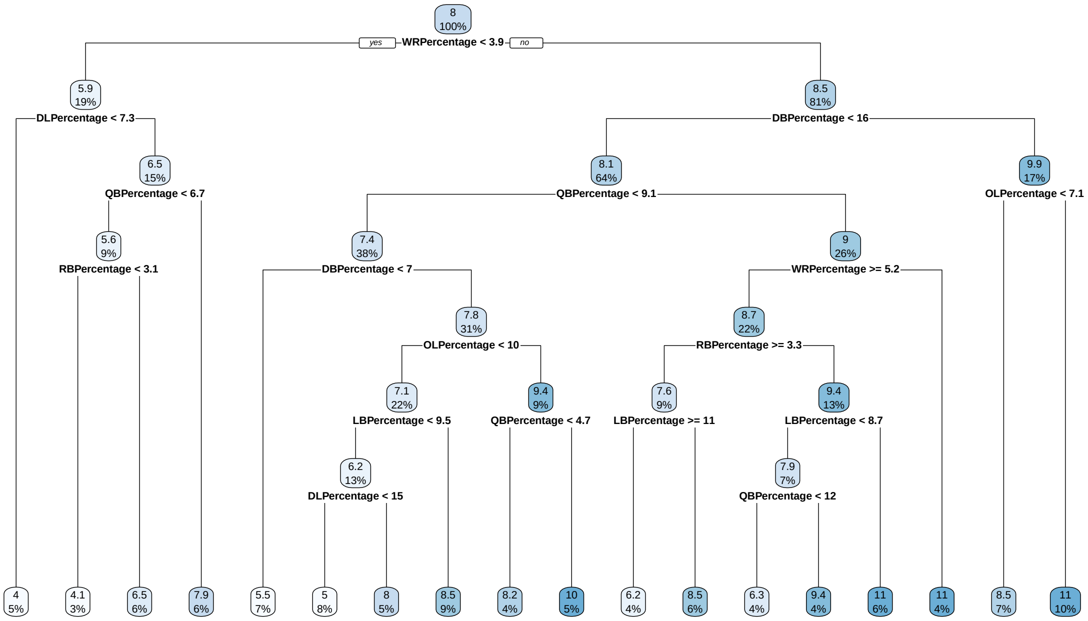


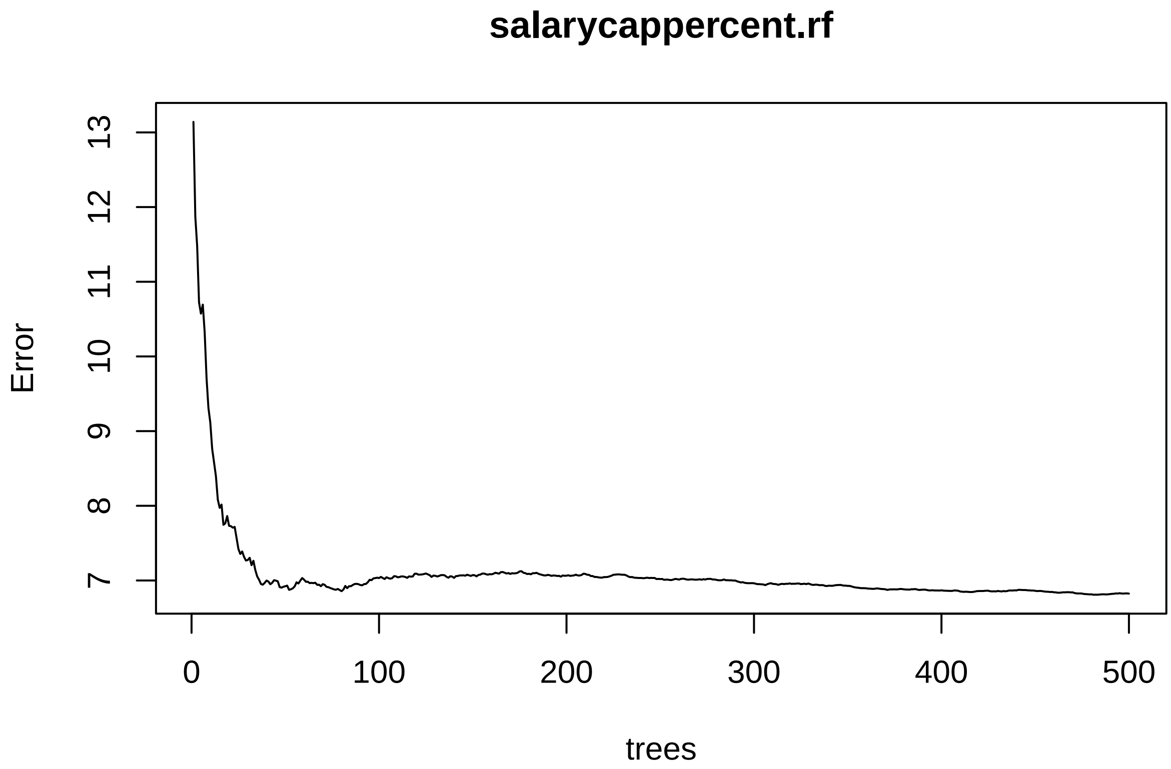
Figure 18 CART Tree of Wins Predicted by Money Spent per Position Group in Percentages

The other CART Tree is looking at the percentage of salary cap used to predict the number of wins. For this model, all teams that look to average more than 10 wins in a season need to first spend more than 3.9% of their salary cap on Wide Receivers, which 81% of the teams in the last 11 years have. The next most important position to look at is DB where teams that spend greater than 16% of the money on this position are on pace to average 9.9 wins, which is a different position then looking at the physical amount of money.

The models show that there is clearly more benefit from looking at the cap percentages as opposed to the dollar amount so the next phase of the analysis was to create random forests of the original three types of models, types of cap, phase, and position group. In order to first use a random forest a training and testing data set are needed to be created with 70% of the observations being included in the training data set and 30% of the observations being in the testing data set.

```
##
## Call:
##  randomForest(formula = W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent,      data = salariestrai
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 6.824094
##                     % Var explained: 32.1
```

Figure 19 Random Forest output for salary cap analysis The percentage of variance explained in this model is 24.44% in relation to the amount of wins a team is projected.



```
## [1] 481
```

```
## [1] 2.609402
```

Figure 20 Plot of Number of Trees vs Mean Squared Error

In order to tune the random forest model of salary cap breakdown the number of trees to produce the lowest value for the mean squared error (MSE) is 481 trees. The value for RMSE is 2.61 meaning that the average difference between predicted wins and actual wins is 2.61.

```
## 
## Call:
##  randomForest(formula = W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent,      data = salariestrai
##                Type of random forest: regression
##                      Number of trees: 481
## No. of variables tried at each split: 1
## 
##           Mean of squared residuals: 6.749606
##                     % Var explained: 32.84
```

Figure 21 Random Forest Output of Tuned Salary Cap Data

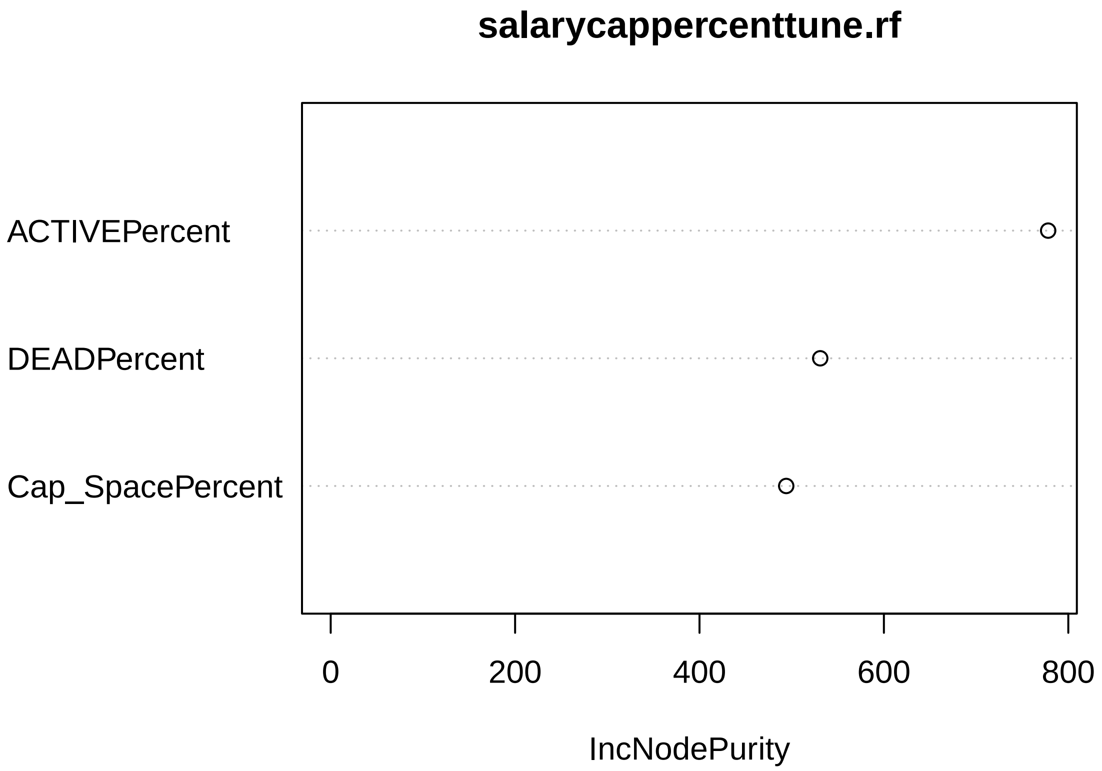


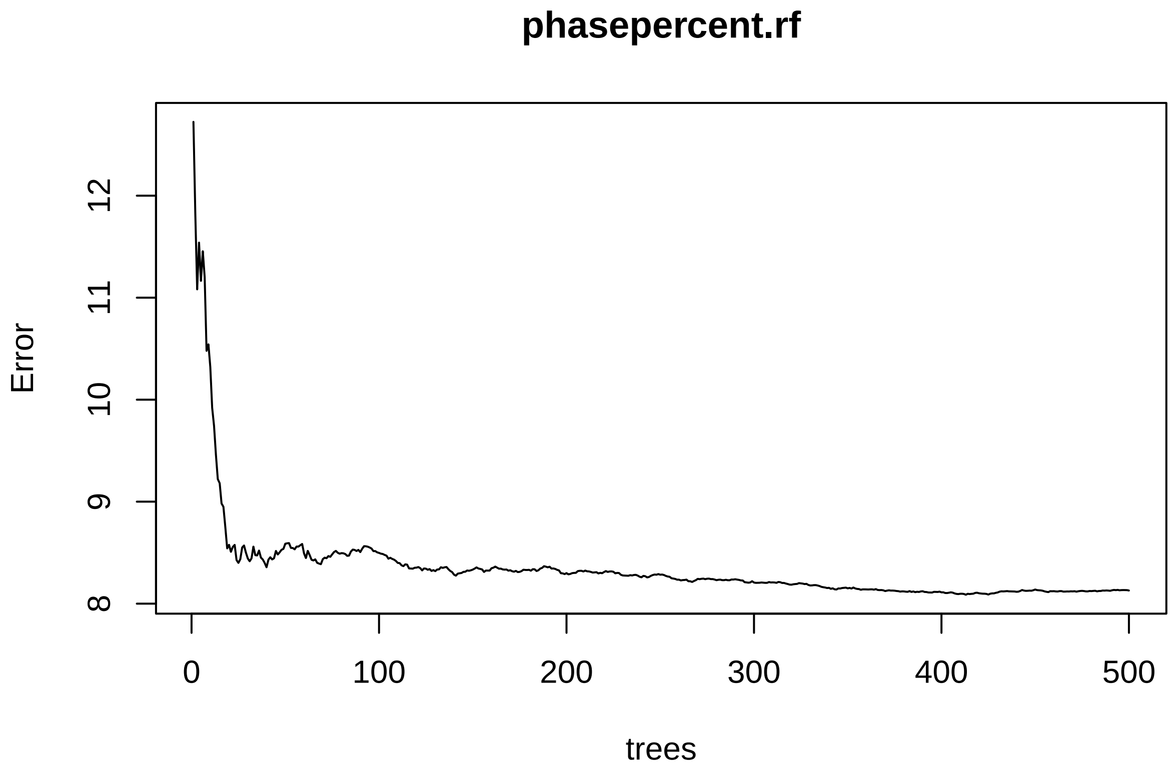
Figure 22 Variable Importance Chart for Variables used in Salary Cap Tuned Model

Next following the same process the percentage of money spent per phase of the game can be analyzed using a random forest to see how a tuned model can do predicting the amount of wins a team is projected.

```
## 
## Call:
##  randomForest(formula = W ~ OffensePercentage + DefensePercentage +      STPercentage, data = salari
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
## 
##           Mean of squared residuals: 8.128946
##                     % Var explained: 19.12
```

Figure 23 Random Forest Output for Phases Analysis

Based on the model that uses phase percentage 19.12% of the variance in wins can be explained by the percentage of money spent per phase of the game.



```
## [1] 413
```

```
## [1] 2.843852
```

Figure 24 Plot of Number of Trees vs Mean Squared Error In order to tune the random forest model of phase breakdown the number of trees to produce the lowest value for the mean squared error (MSE) is 413 trees. The value for RMSE is 2.84 meaning that the average difference between predicted wins and actual wins is 2.84.

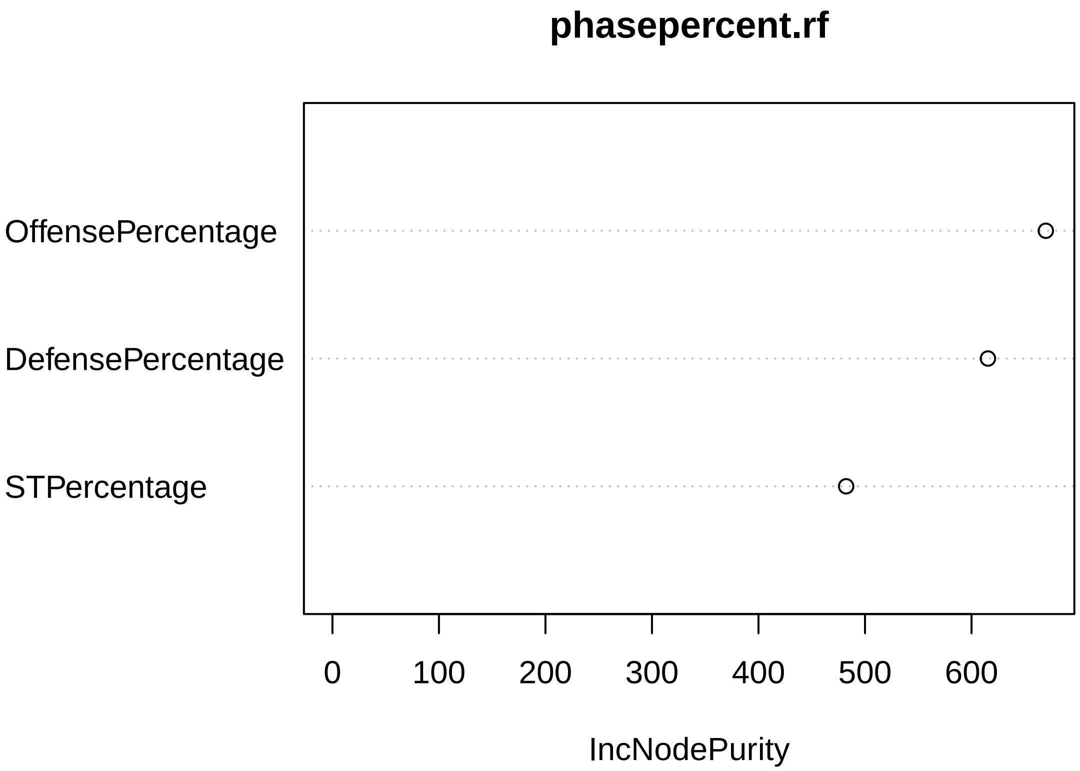
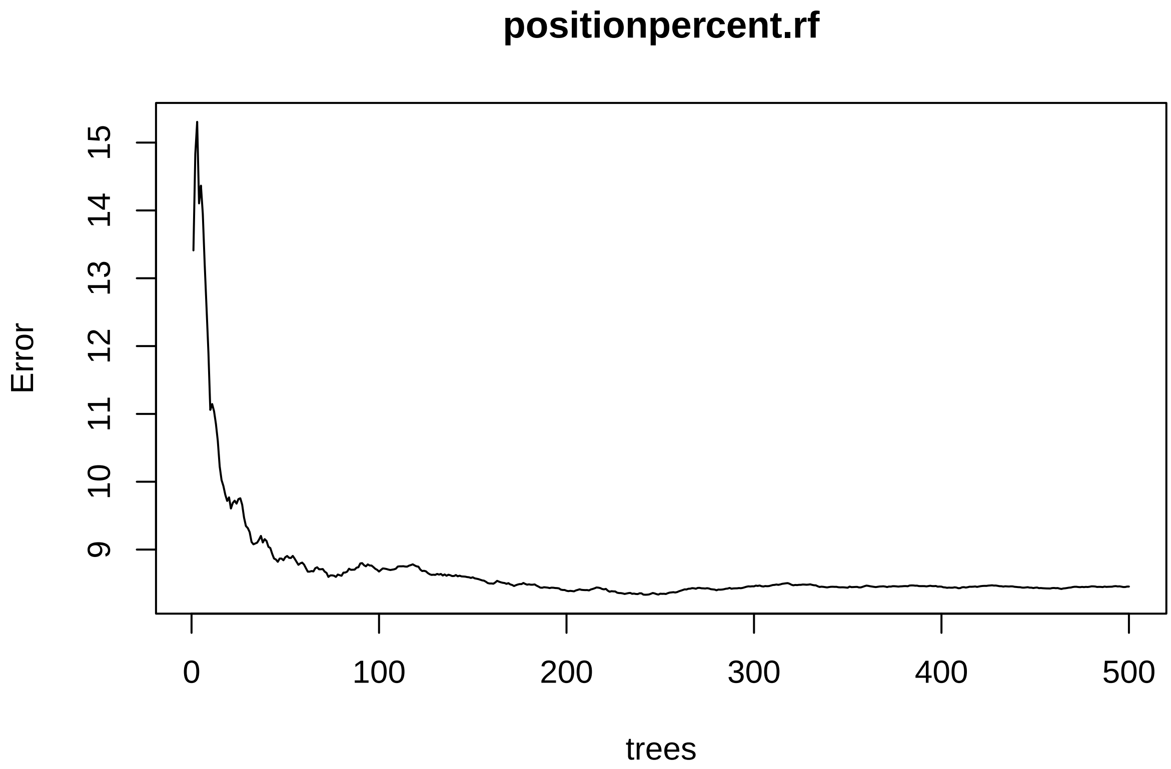


Figure 25 Variable Importance Chart for Phase Percentage Random Forest
When looking at the variable importance chart it shows that the amount of the cap used on the Offense is the most important, followed by Defense, and lastly special teams.

```
##
## Call:
##  randomForest(formula = W ~ OLPercentage + DLPercentage + QBPercentage +      RBPercentage + WRPercen
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 8.454836
##                     % Var explained: 15.88
```

Figure 26 Random Forest output for position analysis
The percentage of variance explained in this model is 15.88% in relation to the amount of wins a team is projected.



```
## [1] 242
```

```
## [1] 2.886893
```

Figure 27 Plot of Number of Trees vs Mean Squared Error

In order to tune the random forest model of salary cap breakdown the number of trees to produce the lowest value for the mean squared error (MSE) is 242 trees. The value for RMSE is 2.87 meaning that the average difference between predicted wins and actual wins is 2.87.

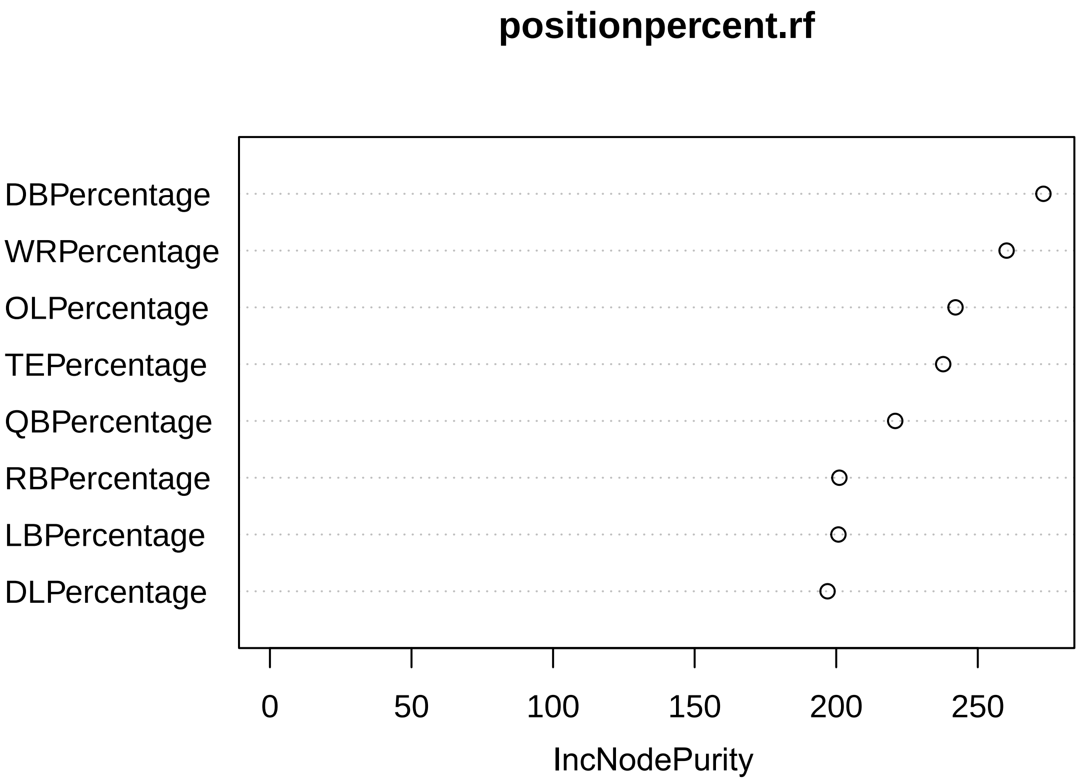


Figure 28 Variable Importance Chart for Variables used in Salary Cap Tuned Model
Based on the Variable Importance Plot of the position percentage random forest to see which positions are the most important when deciding the amount of projected wins a team might have. Based on the graph it is shown that the amount of money spent on the defensive back position is the most important followed by the Wide Receivers, Offensive Line, Tight End, Quarterback, Running Back, Linebackers, and Defensive Line.

Since the Random Forests did not do a wonderful job in predicting the number of wins, the response variable needs to be reconsidered. Random Forests do a better job in predicting the a categorical variable, as opposed to a continuous one. As stated previously a teams ultimate goal for the regular season is to achieve greater than or equal to 10 wins which can be looked at as a categorical variable. This variable of TenPlus was added to the final data set to look at a categorical variable of teams that are more than likely to make the NFL playoffs. This variable is coded as 1 = a 10+ win team and 0 = less than 10 wins.

```
## Call:
##  randomForest(formula = as.factor(TenPlus) ~ ACTIVEPercent + DEADPercent +      Cap_SpacePercent, da
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 25.38%
## Confusion matrix:
##     0  1 class.error
## 0 116 15   0.1145038
## 1  35 31   0.5303030
```

Figure 29 Random Forest Output for Salary Cap Percent Analysis Based on the Output there is an OOB

error rate of 25.38% percent which means that only 25.38 percent of the team are placed in the wrong category of 10+ wins or not.

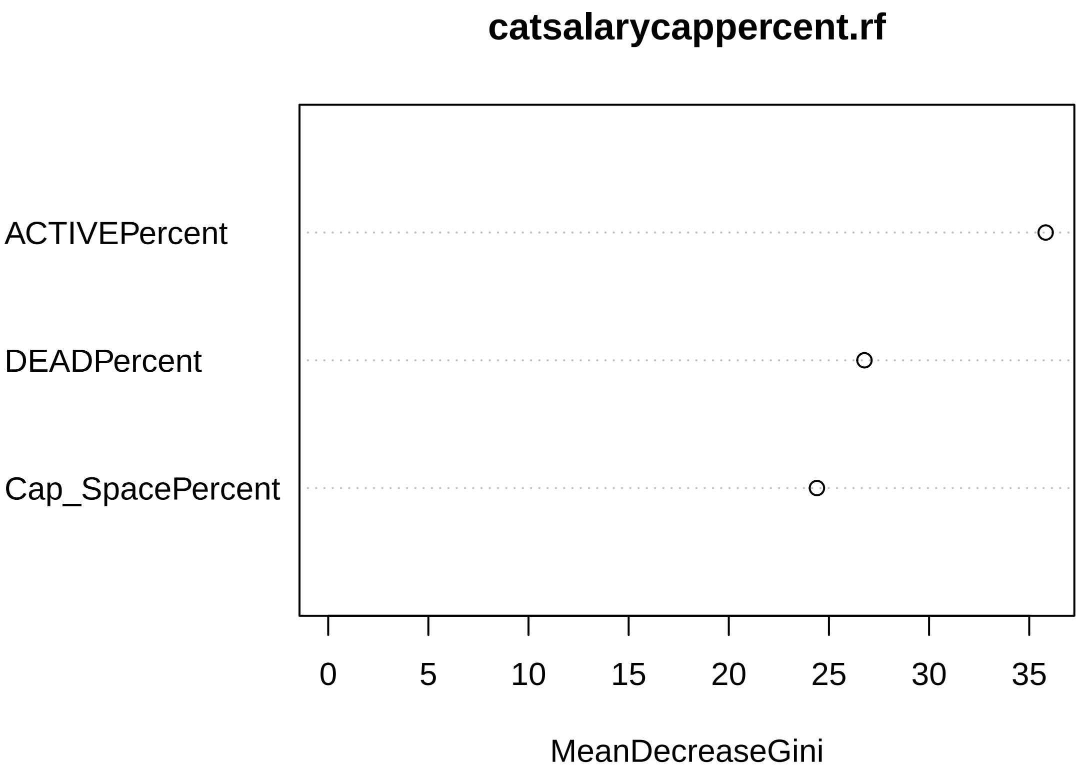


Figure 30 Variable Importance Plot of Salary Cap Percentage Analysis Based on the variable importance plot just like the previous model the variable of Active Percent spent is the most important.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 48 17
##          1  7 13
##
##                Accuracy : 0.7176
##                  95% CI : (0.6096, 0.81)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.10461
##
##                   Kappa : 0.3311
##
##  Mcnemar's Test P-Value : 0.06619
##
##             Sensitivity : 0.8727
##             Specificity : 0.4333
##          Pos Pred Value : 0.7385
##          Neg Pred Value : 0.6500
##              Prevalence : 0.6471
##          Detection Rate : 0.5647
##    Detection Prevalence : 0.7647
##       Balanced Accuracy : 0.6530
##
##        'Positive' Class : 0
##
```

Figure 31 Confusion Matrix for Salary Cap Random Forest Based on the confusion matrix when looking at how the model performs on the test data set there is a 71.76% accuracy when predicting if a team has 10+ wins or not.

```
##
## Call:
##  randomForest(formula = as.factor(TenPlus) ~ OffensePercentage +      DefensePercentage + STPercentag
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 34.52%
## Confusion matrix:
##     0  1 class.error
## 0 101 30   0.2290076
## 1  38 28   0.5757576
```

Figure 32 Random Forest Output for Position Percent Analysis Based on the Output there is an OOB error rate of 34.52 percent which means that only 34.52 percent of the team are placed in the wrong category of 10+ wins or not.

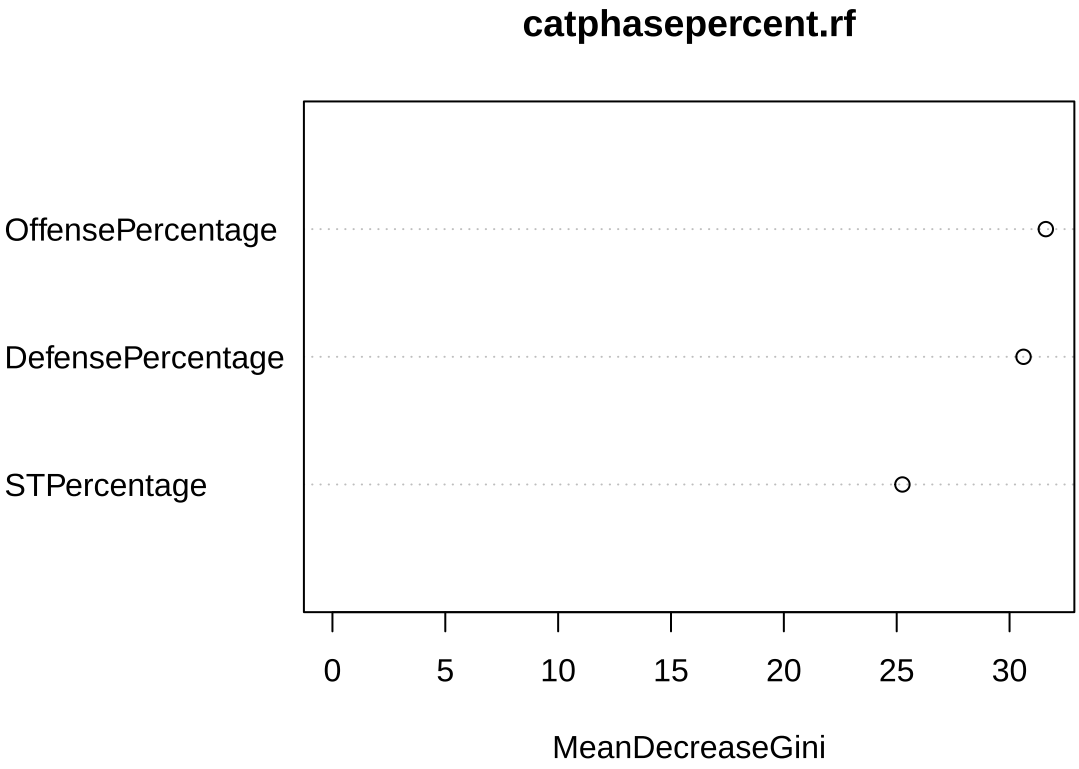


Figure 33 Variable Importance Plot of Salary Cap Percentage Analysis Based on the variable importance plot the offensive spending is the most important, while the defense spending is the second most important.

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction  0  1
##          0 43 18
##          1 12 12
##                                           
##                Accuracy : 0.6471          
##                  95% CI : (0.5359, 0.7477)
##     No Information Rate : 0.6471          
##     P-Value [Acc > NIR] : 0.5495          
##                                           
##                   Kappa : 0.1905          
##                                           
##  Mcnemar’s Test P-Value : 0.3613          
##                                           
##             Sensitivity : 0.7818          
##             Specificity : 0.4000          
##          Pos Pred Value : 0.7049          
##          Neg Pred Value : 0.5000          
##              Prevalence : 0.6471          
##          Detection Rate : 0.5059          
##    Detection Prevalence : 0.7176          
##       Balanced Accuracy : 0.5909          
##                                           
##        ’Positive’ Class : 0               
## 
```

Figure 34 Confusion Matrix for Phase Random Forest Based on the confusion matrix when looking at how the model performs on the test data set there is a 64.71% accuracy when predicting if a team has 10+ wins or not, which is not as good as the previous model.

```
## 
## Call:
##  randomForest(formula = as.factor(TenPlus) ~ OLPercentage + DLPercentage +      QBPercentage + RBPerc
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
## 
##         OOB estimate of  error rate: 36.04%
## Confusion matrix:
##     0  1 class.error
## 0 113 18   0.1374046
## 1  53 13   0.8030303
```

Figure 35 Random Forest Output for position analysis Based on the output there is an OOB estimated error rate of 36.04%, which is the lowest out of all of the models.

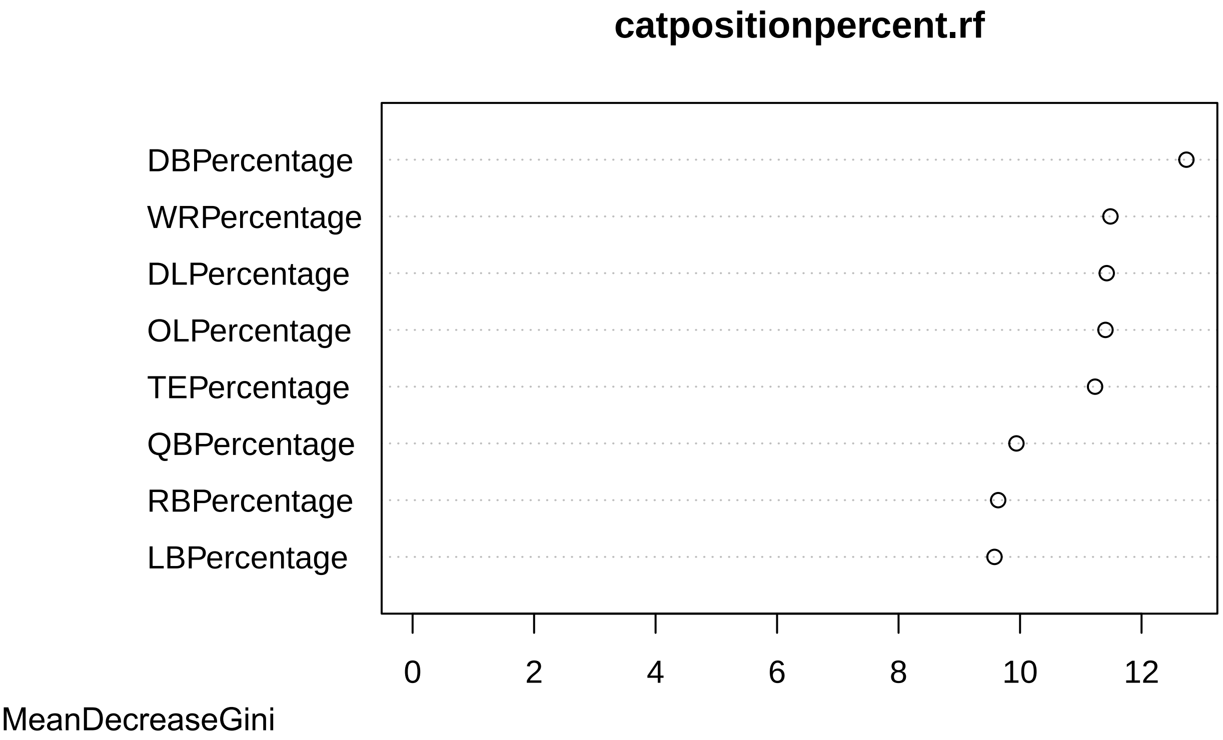


Figure 36 Variable Importance Plot for Position Analysis Based on the variable importance plot we can see that defensive back spending is the most important position followed by widereciever, defensive line, offensive line, tight end, Quarterback, Running Back, and Linebackers.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 45 20
##          1 10 10
##
##                Accuracy : 0.6471
##                  95% CI : (0.5359, 0.7477)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.5495
##
##                   Kappa : 0.1639
##
##  Mcnemar's Test P-Value : 0.1003
##
##             Sensitivity : 0.8182
##             Specificity : 0.3333
##          Pos Pred Value : 0.6923
##          Neg Pred Value : 0.5000
##              Prevalence : 0.6471
##          Detection Rate : 0.5294
##    Detection Prevalence : 0.7647
##       Balanced Accuracy : 0.5758
##
##        'Positive' Class : 0
##
```

Figure 37 Confusion Matrix of Position Analysis Based on the Confusion Matrix output there is a 60% accuracy when it comes to predicting ten win teams.

Discussion:

When looking at the results it is important to break them down by all of the different types of models to see how each piece of information contributes to the prediction of number of wins and success of the team. There was a big piece of information that came out of the data visualization portion, as well as the linear models and CART trees. When looking at Figure 2 and comparing the amount of spending over the past 11 years in the National Football League and seeing the gradual increase in spending from year to year minus a small decrease in spending from 2020 to 2021. The overall change in the salary cap has gone from 2.7 billion in 2021 to its peak of 5.2 billion in 2020. This was an important discovery because the amount of money spent might not be the best way to estimate wins, since the salary cap is always changing. This fact was then proven to be true when looking at the linear models.

For the linear models there were six in total that were created, which included a model that looked at the Active, Dead, and Remaining Salary Cap, a model that looked at the amount of spending per phase, and the amount of spending per position. For each one of these variable combinations a model that used amount of money spent in millions as well as used the amount of money in cap hit percentage. From these linear models, in all three cases the model that used the percentages out performed the one that used dollars in millions. For the model that looks at active salary cap, dead salary cap, and cap space remaining the model that uses millions had an r-squared of 0.2407 compared to the percentage model of 0.2978. For the model that looks at phase of the game the millions model had an r-squared of 0.1921 compared to the percentage model of 0.26. Lastly, the position model in millions had an adjusted r-squared of 0.1635 compared to the percentage model of 0.2455. With all of those numbers it is safe to say that looking at the percentage of the salary cap used is the better way to go about the analysis.

Breaking the models down one by one, the one that was the most effective at explaining the variance in wins was the active, dead, and remaining salary cap was the best, however each has its own benefits. However looking at this model first we can see based on the output of figure 9 that the percentage of salary used on active salary cap is the most important variable with every 1% spent on it a team can expect 0.16224 wins. With every 1% of money spent on dead salary cap a team can expect 0.09694 wins, and for every 1% of the salary cap remaining a team can expect 0.08239 wins. A simple way to look at this is a team need to be smart with the way that they manage their finances, as well as need to be a little lucky along the way. Dead cap does lead to wins, but active cap leads to wins at almost double the amount. The CART tree of figure 10 shows that if a team spend more than 73% of their cap on active salary they will average 9.2 wins then a team that doesn't who averages 6.2 wins. As well as a team that spends more than 79% of their salary cap on active players will average 10 wins, and if their dead cap percentage falls between 5.9% and 13% a team will average 12 wins a year. The random forest output of the tuned model in figure 21 that 32.1% of the variation in wins can be determined by spending. This carries a RMSE of 2.61 meaning that the model on average will miss the actual number of wins by 2.61 wins.

When looking at the model that takes into consideration the phase of the game and spending in percentage per phase the first thing to look at is table 5 . In this table the average amount of spending on the offense is 35.10%, the defense is 33.36% and special teams is 2.51%. This shows that over the past 11 years that teams have placed more value on offense, rather than defense and special teams. When running the linear model and looking at the output in figure 13 that special teams actually had the highest increase in wins per 1% at 0.26397, however this number was not significant in the linear model since a team only averages spending

2.51% of their cap in that position. Offense was the highest of the significant variables with an average of 0.13342 wins per 1% spent and the defense having an additional 0.11095 wins per 1% spent. The CART Tree in figure 14 shows that teams that spend more than 27% of their available salary cap will average 8.6 wins as opposed to 5.6 wins for those who do not. After that of those teams that have 27+% spent on offense if they then spend 26% on defense they will average 9 wins as opposed to 6.5. Overall teams that want to maximize the number of wins want an offensive spending amount between 38 and 47 percent of the salary cap and a defensive spending of over 38% with no top dollar cap that starts to become a factor. Showing that the amount of money spent on offense is more valuable within the limits, while defensive spending does not have a cap that then has negative implications. The random forest only shows that 19.12% of the variation in wins is predicted by the phase model with an RMSE of 2.84 wins as seen in figure 23. However based on the variable importance chart in figure 24 it is opposite of the linear model and CART Tree with Offense being the most important phase followed closely by the deense.

The last salary breakdown was looking at the different position groups to see which player groups are paid the most currently, and which groups should be paid the most. Table 6 shows that the order of average percentage of the salary cap taken up per team over the last 11 years is Defensive Line, Defensive Backs, Linebackers, Offensive Line, Wide Receivers, Quarterbacks, Running Backs, and lastly Tight Ends, with almost a 9% range between these players. When looking at the regression output in figure 16 there is an r-squared value of 0.2455 with only one variable not being significant which is the value of the Running Back. Actually according to the model the Tight End is the most influential variable with about 0.17 wins per 1% spent however the average percentage spent on tight ends is so low. After that it is defensive backs and quarterbacks at about 0.17 wins per 1% as well, with Wide Receivers being the next highest at about 0.13 Wins per 1%. The CART tree in figure 18 the top of the tree is the wide receiver percentage at 3.9% and teams having on average 8.5 wins compared to 5.9 wins. There are a couple easy formulas to have your team win on average 11 games. All a team would need to do is have greater than 3.9% of the salary cap for wide receivers, defensive backs greater than 16% of the salary cap, and Offensive line greater than 7.1%. Another way a team could average 11 wins is if they pay their defensive backs less than 16% then they just need to pay a quarterback greater than 9.1% of the salary. Lastly looking at the tuned random forest output in figures 26-28 we can see that only 15.88% of the variation in wins is actually explained by the model. With an RMSE of 2.89 it is actually the worst out of the three models in predicting wins, but the combination of all three could provide some backing when used in conjunction. Finally looking at the variable importance plot Defensive Backs, Wide Reveivers, Offensive Line, and Defensive Line are the most important position to pay.

All three of the models above do an all right job in projecting the number of wins, but would like to be higher and show more of what a team can expect out of their rosters. At the beginning of the study it was shown that almost every team that wins 10+ games make the playoffs, so what if the same models were then used with a categorical win ten games or not variable. The tuned random forests designs show that maybe the models are better at predicting then they initially show. When the same models for Salary Cap breakdown, phase breakdown, and position breakdown are ran some beneficial results are found. Figures 29-31 show the random forest results of the tuned model for salary cap breakdown. With an OOB Error of 26.4% that means that 26.4 % of the time the model places the team in the wrong category of 10+ win team or not. The confusion matrix shows an accuracy of 71.76% which shows that the breakdown of active, dead, and cap remaining percent correctly predict playoff teams 7 out of 10 times correctly, with the variable of Active Salary being the most important, followed by Dead and then cap remaining. The Phase analysis ran to predict playoff teams showed similar success with an OOB error of about 30.96% as shown in figure 32. Figure 33 shows that Offense is more important to pay then defense, and figure 34 shows that the confusion matrix has an accuracy of about 61%. Lastly, the model that looks at the position salary cap percentage has an OOB error of 32.49% as seen in figure 35. Figure 37 shows that the model has an accuracy of 60%, and figure 37 shows that Quarterbacks, Defensive Line, Offensive Line, and Defensive Line are the most important positions to pay. The interesting thing is that the model is actually saying the least important positions to pay if a team wants to make the payoffs is running backs, wide receivers, and tight ends. Traditionally, these positions groups are know to be the skill positions and to the common viewer are perceived as the most important players on the field. Overall, when looking at salary percentage data in multiple different forms it does an okay job in predicting the number of wins, but does an above average job in predicting teams that make the playoffs, and when used in combination could be extremely effective tools in the prediction of successful teams in the National Football League. Try to spend the most money on Active Players, minimize remaining cap, pay offense a certain amount, pay defense and equal amount with no cap, and under no circumstance pay running backs or tight ends, while valuing players in the defensive back position as well as on the offensive and defensive line. Follow these general guidelines and a team has a higher chance of making the playoffs then those who do not.

What's next? Salary Cap data can only explain so much of the parody that is the National Football League and there are multiple more steps that can be used to help refine this process. First different types of logistic regression can be used with the prediction of playoff teams that was found later in the study. There can also be other machine learning techniques like neural networks and support vector machines that when properly tuned could provide even more accurate analysis. The other big part that could be added to the analysis is the incorporation of team statistics like points scored, points allowed, and player statistics. A combination of salary data and player statistics could unlock another level to being able to assemble the most cost effective NFL roster.

References:

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. In Journal of Computational and Graphical Statistics (Vol. 15, Issue 3, pp. 651–674).

Milborrow S (2022). *rpart.plot: Plot 'rpart' Models: An Enhanced Version of 'plot.rpart'*. R package version 3.1.1, https://CRAN.R-project.org/package=rpart.plot.

NFL points scored career leaders. Pro. (n.d.). Retrieved December 12, 2022, from https://www.pro-football-reference.com/leaders/scoring_career.htm

Official site of the National Football League. NFL.com. (n.d.). Retrieved October 31, 2022, from https://www.nfl.com/

Pedersen T (2022). *patchwork: The Composer of Plots*. R package version 1.1.2, https://CRAN.R-project.org/package=patchwork.

Spotrac.com. Sports Contracts, Salaries, Caps, Bonuses, & Transactions. (1970, October 31). Retrieved October 31, 2022, from https://www.spotrac.com/

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." Journal of Open Source Software, 4(43), 1686. doi:10.21105/joss.01686.

Apendicies:

Apendix A: Code

Importing and Cleaning Data

```r
library(tidyverse)
library(readxl)
NFL_Active_Player_Salaries <- read.csv("~/PDAT 620G/NFL Active Player Salaries - Sheet1.csv")
NFL_Cap_Hits_ <- read.csv("~/PDAT 620G/NFL Cap Hits  - Sheet1.csv")
```

```
NFL_Records <- read.csv("~/PDAT 620G/NFL Records - Sheet1.csv")
options(scipen = 999)
```

Cleaning Active Salaries Dataset

```
NFL_Active_Player_Salaries <- NFL_Active_Player_Salaries %>%
  rename_at("CAP..",~"CAP.PERCENT")

NFL_Active_Player_Salaries$BASE.SALARY = gsub("\\$", "", NFL_Active_Player_Salaries$BASE.SALARY)
NFL_Active_Player_Salaries$BASE.SALARY = gsub("\\,", "", NFL_Active_Player_Salaries$BASE.SALARY)
NFL_Active_Player_Salaries$BASE.SALARY <- as.numeric(NFL_Active_Player_Salaries$BASE.SALARY)

NFL_Active_Player_Salaries$CAP.HIT = gsub("\\$", "", NFL_Active_Player_Salaries$CAP.HIT)
NFL_Active_Player_Salaries$CAP.HIT = gsub("\\,", "", NFL_Active_Player_Salaries$CAP.HIT)
NFL_Active_Player_Salaries$CAP.HIT <- as.numeric(NFL_Active_Player_Salaries$CAP.HIT)

NFL_Active_Player_Salaries <- NFL_Active_Player_Salaries %>%
  select("ACTIVE_PLAYERS_(56)", "POS.", "BASE_SALARY", "CAP_HIT", "CAP_%", "YEAR", "TEAM")
NFL_Active_Player_Salaries <- NFL_Active_Player_Salaries %>%
  rename_at('ACTIVE_PLAYERS_(56)',~'Active_Players') %>%
  rename_at("POS.",~"Pos") %>%
  rename_at("BASE_SALARY",~"Base_Salary") %>%
  rename_at("CAP_HIT",~"Cap_Hit") %>%
  rename_at("CAP_%",~"Cap_%")
```

Cleaning NFL Cap Hits

```
NFL_Cap_Hits_$ACTIVE = gsub("\\$", "", NFL_Cap_Hits_$ACTIVE)
NFL_Cap_Hits_$ACTIVE = gsub("\\,", "", NFL_Cap_Hits_$ACTIVE)
NFL_Cap_Hits_$ACTIVE <- as.numeric(NFL_Cap_Hits_$ACTIVE)

NFL_Cap_Hits_$DEAD = gsub("\\$", "", NFL_Cap_Hits_$DEAD)
NFL_Cap_Hits_$DEAD = gsub("\\,", "", NFL_Cap_Hits_$DEAD)
NFL_Cap_Hits_$DEAD <- as.numeric(NFL_Cap_Hits_$DEAD)

NFL_Cap_Hits_$TOTAL.CAP = gsub("\\$", "", NFL_Cap_Hits_$TOTAL.CAP)
NFL_Cap_Hits_$TOTAL.CAP = gsub("\\,", "", NFL_Cap_Hits_$TOTAL.CAP)
NFL_Cap_Hits_$TOTAL.CAP <- as.numeric(NFL_Cap_Hits_$TOTAL.CAP)

NFL_Cap_Hits_$CAP.SPACE = gsub("\\$", "", NFL_Cap_Hits_$CAP.SPACE)
NFL_Cap_Hits_$CAP.SPACE = gsub("\\,", "", NFL_Cap_Hits_$CAP.SPACE)
NFL_Cap_Hits_$CAP.SPACE <- as.numeric(NFL_Cap_Hits_$CAP.SPACE)

NFL_Cap_Hits_ <- NFL_Cap_Hits_ %>%
  rename_at("AVE_AGE",~"Ave_Age") %>%
  rename_at("TOTAL_CAP",~"Total_Cap") %>%
  rename_at("CAP_SPACE_(ALL)",~"Cap_Space")
```

Cleaning NFL Records

```r
names(NFL_Records) <- gsub(" ",".", names(NFL_Records))
```

Joining Datasets

```r
Salary_Cap <- NFL_Cap_Hits_ %>%
  inner_join(NFL_Records, by = c("TEAM" = "NFL.Team", "YEAR" = "Year"))

Salary_Cap <- Salary_Cap %>% mutate(ACTIVEPercent = (ACTIVE/Total_Cap) * 100,
                                 DEADPercent = (DEAD/Total_Cap) * 100,
                                 Cap_SpacePercent = (Cap_Space/Total_Cap) * 100)

Salary_Cap <- Salary_Cap %>% mutate(ACTIVE = ACTIVE*0.000001,
                                 DEAD = DEAD*0.000001,
                                 Cap_Space = Cap_Space*0.000001,
                                 Total_Cap = Total_Cap*0.000001)

Salaries <- NFL_Active_Player_Salaries %>%
  inner_join(NFL_Records, by = c("TEAM" = "NFL.Team", "YEAR" = "Year")) %>%
  mutate(Phase = case_when(POS == "C" ~ "Offense",
                           POS == "CB" ~ "Defense",
                           POS == "DE" ~ "Defense",
                           POS == "DT" ~ "Defense",
                           POS == "FB" ~ "Offense",
                           POS == "FS" ~ "Defense",
                           POS == "G" ~ "Offense",
                           POS == "ILB" ~ "Defense",
                           POS == "K" ~ "Special Teams",
                           POS == "KR" ~ "Special Teams",
                           POS == "LB" ~ "Defense",
                           POS == "LS" ~ "Special Teams",
                           POS == "LT" ~ "Offense",
                           POS == "OL" ~ "Offense",
                           POS == "OLB" ~ "Defense",
                           POS == "P" ~ "Special Teams",
                           POS == "PR" ~ "Special Teams",
                           POS == "QB" ~ "Offense",
                           POS == "RB" ~ "Offense",
                           POS == "RT" ~ "Offense",
                           POS == "S" ~ "Defense",
                           POS == "SS" ~ "Defense",
                           POS == "T" ~ "Offense",
                           POS == "TE" ~ "Offense",
                           POS == "WR" ~ "Offense"))

SalariesPosition <- Salaries %>%
  group_by(TEAM, POS, YEAR) %>%
  summarise(Percentage = sum(`CAP.PERCENT`),
            Cap_Hit = sum(CAP.HIT),
            .groups = "drop")
```

```
SalariesPosition <- SalariesPosition %>%
  inner_join(NFL_Records, by = c("TEAM" = "NFL.Team", "YEAR" = "Year"))

SalariesPosition <- SalariesPosition %>%
  mutate(Phase = case_when(POS == "C" ~ "Offense",
                           POS == "CB" ~ "Defense",
                           POS == "DE" ~ "Defense",
                           POS == "DT" ~ "Defense",
                           POS == "FB" ~ "Offense",
                           POS == "FS" ~ "Defense",
                           POS == "G" ~ "Offense",
                           POS == "ILB" ~ "Defense",
                           POS == "K" ~ "Special Teams",
                           POS == "KR" ~ "Special Teams",
                           POS == "LB" ~ "Defense",
                           POS == "LS" ~ "Special Teams",
                           POS == "LT" ~ "Offense",
                           POS == "OL" ~ "Offense",
                           POS == "OLB" ~ "Defense",
                           POS == "P" ~ "Special Teams",
                           POS == "PR" ~ "Special Teams",
                           POS == "QB" ~ "Offense",
                           POS == "RB" ~ "Offense",
                           POS == "RT" ~ "Offense",
                           POS == "S" ~ "Defense",
                           POS == "SS" ~ "Defense",
                           POS == "T" ~ "Offense",
                           POS == "TE" ~ "Offense",
                           POS == "WR" ~ "Offense"))

SalariesPosition <- SalariesPosition %>%
  inner_join(NFL_Cap_Hits_, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

write.csv(SalariesPosition, "C:\\Users\\spenc\\Documents\\PDAT 620G\\SalariesPosition.csv")

write.csv(Salary_Cap, "C:\\Users\\spenc\\Documents\\PDAT 620G\\SalaryCap.csv")
```

Joining Datasets for Phase Analysis

```
SalariesDefense <- SalariesPosition %>% filter(Phase == "Defense") %>%
  group_by(YEAR, TEAM) %>%
  summarise(DefensePercentage = sum(Percentage),
            DefenseCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesOfffense <- SalariesPosition %>% filter(Phase == "Offense") %>%
  group_by(YEAR, TEAM) %>%
  summarise(OffensePercentage = sum(Percentage),
            OffenseCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")
```

```r
SalariesST <- SalariesPosition %>% filter(Phase == "Special Teams") %>%
  group_by(YEAR, TEAM) %>%
  summarise(STPercentage = sum(Percentage),
            STCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesPhases <- SalariesOffense %>%
  inner_join(SalariesDefense, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

SalariesPhases <- SalariesPhases %>%
  inner_join(SalariesST, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

SalariesPhases <- SalariesPhases %>% select(!X)

write.csv(SalariesPhases, "C:\\Users\\spenc\\Documents\\PDAT 620G\\SalariesPhases.csv")
```

Cleaning Data For Position Analysis

```r
SalariesOL <- SalariesPosition %>% filter(Pos %in% c("C","G","RT","T")) %>%
  group_by(YEAR, TEAM) %>%
  summarise(OLPercentage = sum(Percentage),
            OLCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesDL <- SalariesPosition %>% filter(Pos %in% c("DT","DE")) %>%
  group_by(YEAR, TEAM) %>%
  summarise(DLPercentage = sum(Percentage),
            DLCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesQB <- SalariesPosition %>% filter(Pos == "QB") %>%
  group_by(YEAR, TEAM) %>%
  summarise(QBPercentage = sum(Percentage),
            QBCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesRB <- SalariesPosition %>% filter(Pos == "RB") %>%
  group_by(YEAR, TEAM) %>%
  summarise(RBPercentage = sum(Percentage),
            RBCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesWR <- SalariesPosition %>% filter(Pos == "WR") %>%
  group_by(YEAR, TEAM) %>%
  summarise(WRPercentage = sum(Percentage),
            WRCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesTE <- SalariesPosition %>% filter(Pos == "TE") %>%
  group_by(YEAR, TEAM) %>%
  summarise(TEPercentage = sum(Percentage),
            TECap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesDB <- SalariesPosition %>% filter(Pos %in% c("CB","FS","S","SS")) %>%
  group_by(YEAR, TEAM) %>%
  summarise(DBPercentage = sum(Percentage),
            DBCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesLB <- SalariesPosition %>% filter(Pos %in% c("LB","ILB","OLB")) %>%
  group_by(YEAR, TEAM) %>%
  summarise(LBPercentage = sum(Percentage),
            LBCap_Hit = sum(Cap_Hit) * 0.000001,
            .groups = "drop")

SalariesPositionGroup <- SalariesOL %>%
  inner_join(SalariesDL, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesQB, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesRB, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesWR, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesTE, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesDB, by = c("TEAM" = "TEAM", "YEAR" = "YEAR")) %>%
  inner_join(SalariesLB, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

write.csv(SalariesPositionGroup, "C:\\Users\\spenc\\Documents\\PDAT 620G\\SalariesPositionGroup.csv")
```

Creating the Final Dataset

```
SalariesFinal <- SalariesPhases %>%
  inner_join(SalariesPositionGroup, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

SalariesFinal <- SalariesFinal %>%
  inner_join(SalaryCap, by = c("TEAM" = "TEAM", "YEAR" = "YEAR"))

SalariesFinal <- SalariesFinal %>% select(!c(X.x,X.y)) %>%
  mutate(Total_Cap = Total_Cap * 0.000001)

SalariesFinal <- SalariesFinal %>%
  mutate(Total_Cap = Total_Cap * 0.000001)

SalariesFinal <- SalariesFinal %>%
  mutate(TenPlus = case_when(W >= 10 ~ 1,
                             W <= 9 ~ 0))

write.csv(SalariesFinal, "C:\\Users\\spenc\\Documents\\PDAT 620G\\SalariesFinal.csv")
```

Data Visualizations

```
library(patchwork)
Salary1 <- ggplot(data = SalariesFinal, aes(x = ACTIVE, y = W, color = W)) +
  geom_jitter() +
  ggtitle("Scatterplot of Wins vs
          Active Salary Cap") +
  xlab("Active Salary Cap") +
  ylab("Wins") +
  stat_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma")

Salary2 <- ggplot(data = SalariesFinal, aes(x = DEAD, y = W, color = W)) +
  geom_jitter() +
  ggtitle("Scatterplot of Wins vs
          Dead Salary Cap") +
  xlab("Dead Salary Cap") +
  ylab("Wins") +
  stat_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma")

Salary3 <- ggplot(data = SalariesFinal, aes(x = W, y = Cap_Space, color = W)) +
  geom_jitter() +
  ggtitle("Scatterplot of Wins vs Cap Space Left") +
  xlab("Wins") +
  ylab("Active Salary Cap") +
  stat_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma")

Salary1 + Salary2
```

```
ggplot(data = SalariesFinal, aes(x = YEAR, y = Total_Cap, fill = YEAR)) +
  geom_col() +
  scale_fill_viridis_c(option = "plasma") +
  ylab("League Cap Space (Millions)") +
  xlab("Year") +
  ggtitle("League Cap Space per Year")
```

```
Off <- ggplot(data = SalariesFinal, aes(x = YEAR, y = OffenseCap_Hit, fill = YEAR)) +
  geom_col() +
  scale_fill_viridis_c(option = "plasma") +
  ylab(" ") +
  xlab("Year") +
  ggtitle("Offensive Player Spending")

Dff <- ggplot(data = SalariesFinal, aes(x = YEAR, y = DefenseCap_Hit, fill = YEAR)) +
  geom_col() +
  scale_fill_viridis_c(option = "plasma") +
  ylab("Spending (Millions)") +
  xlab("Year") +
  ggtitle("Defensive Player Spending")

ST <- ggplot(data = SalariesFinal, aes(x = YEAR, y = STCap_Hit, fill = YEAR)) +
  geom_col() +
  scale_fill_viridis_c(option = "plasma") +
  xlab("Year") +
  ylab(" ") +
  ggtitle("Special Teams Player Spending")

Off / Dff / ST
```

```
ggplot(Positions) +
  geom_col(aes(fct_rev(fct_reorder(Pos, Average_Salary)), Average_Salary, fill = Phase)) +
  ggtitle("NFL Position Group Average Salaries per Player") +
  xlab("Position Group") +
  ylab("Average Salary") +
  theme(axis.text.x = element_text(size = 6))
```

```
ggplot(Positions) +
  geom_col(aes(fct_rev(fct_reorder(Pos, Average_Salary)), Average_Salary*0.000001, fill = Phase)) +
  ggtitle("NFL Position Group Average Salaries per Player") +
  xlab("Position Group") +
  ylab("Average Salary (Millions)") +
  theme(axis.text.x = element_text(size = 6)) +
  scale_fill_viridis_d(option = "plasma")
```

```
ggplot(data = Salaries) +
  geom_boxplot(aes(fct_rev(fct_reorder(Pos, Base_Salary)), Base_Salary*0.000001, fill = Phase)) +
  ggtitle("Boxplot of NFL Position Group Salaries") +
  xlab("Position Group") +
  ylab("Average Salary (Millions)") +
  theme(axis.text.x = element_text(size = 6)) +
  scale_fill_viridis_d(option = "plasma")
```

```
p1 <- ggplot(SalariesFinal, mapping = aes(x = W, y = ACTIVEPercent, color = W)) +
  geom_jitter() +
  ggtitle("Wins Vs Active
          Cap Percentage") +
  geom_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma") +
  xlab("Wins") +
  ylab("Active Cap Percent")
```

```
p2 <- ggplot(SalariesFinal, mapping = aes(x = W, y = DEADPercent, color = W)) +
  geom_jitter() +
  ggtitle("Wins Vs Dead
          Cap Percentage") +
  geom_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma") +
  xlab("Wins") +
  ylab("Dead Cap Percentage")
```

```
p3 <- ggplot(SalariesFinal, mapping = aes(x = W, y = Cap_SpacePercent, color = W)) +
  geom_jitter() +
  ggtitle("Wins Vs Cap
          Space Percentage") +
  geom_smooth(method = "lm") +
  scale_colour_viridis_c(option = "plasma") +
  xlab("Wins") +
  ylab("Cap Space Percentage")
```

```
p1 + p2 + p3
```

```
DefenseCapSpending = mean(SalariesFinal$DefensePercentage, na.rm = TRUE)
OffenseCapSpending = mean(SalariesFinal$OffensePercentage, na.rm = TRUE)
STCapSpending = mean(SalariesFinal$STPercentage, na.rm = TRUE)

Spending <- data.frame(DefenseCapSpending, OffenseCapSpending, STCapSpending)
Spending
```

```
OLCapSpending = mean(SalariesFinal$OLPercentage, na.rm = TRUE)
DLCapSpending = mean(SalariesFinal$DLPercentage, na.rm = TRUE)
QBCapSpending = mean(SalariesFinal$QBPercentage, na.rm = TRUE)
RBCapSpending = mean(SalariesFinal$RBPercentage, na.rm = TRUE)
WRCapSpending = mean(SalariesFinal$WRPercentage, na.rm = TRUE)
TECapSpending = mean(SalariesFinal$TEPercentage, na.rm = TRUE)
LBCapSpending = mean(SalariesFinal$LBPercentage, na.rm = TRUE)
DBCapSpending = mean(SalariesFinal$DBPercentage, na.rm = TRUE)

SpendingPositionPercentage <- data.frame(OLCapSpending, DLCapSpending, QBCapSpending, RBCapSpending, WR
SpendingPositionPercentage
```

```
salarycap.lm <- lm(W ~ ACTIVE + DEAD + Cap_Space, data = SalariesFinal)
summary(salarycap2.lm)
salarycap.lm
```

```
salarycap.rpart <- rpart(W ~ ACTIVE + DEAD + Cap_Space, data = SalaryCap)
rpart.plot(salarycap.rpart)
```

```
salarycappercent.lm <- lm(W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent, data = SalariesFinal)
summary(salarycappercent.lm)
salarycappercent.lm
```

```
salarycappercent.rpart <- rpart(W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent, data = SalaryCap)
rpart.plot(salarycappercent.rpart)
```

```
Phase.lm <- lm(W ~ OffenseCap_Hit + DefenseCap_Hit + STCap_Hit, data = SalariesFinal)
summary(Phase.lm)
Phase.lm
```

```
Phase.rpart <- rpart(W ~ OffenseCap_Hit + DefenseCap_Hit + STCap_Hit, data = SalariesPhases)
rpart.plot(Phase.rpart)
```

```
PhasePercent.lm <- lm(W ~ OffensePercentage + DefensePercentage + STPercentage, data = SalariesPhases)
summary(PhasePercent.lm)
PhasePercent.lm
```

```r
PhasePercent.rpart <- rpart(W ~ OffenseCap_Hit + DefenseCap_Hit + STCap_Hit, data = SalariesFinal)
rpart.plot(PhasePercent.rpart)
```

Include Description

```r
PhasePercent2.rpart <- rpart(W ~ OffenseCap_Hit + DefenseCap_Hit, data = SalariesFinal)
rpart.plot(PhasePercent2.rpart)
```

Include Description

```r
Position.lm <- lm(W ~ OLCap_Hit + DLCap_Hit + QBCap_Hit + RBCap_Hit + WRCap_Hit + TECap_Hit + LBCap_Hit
summary(Position.lm)
Position.lm
```

```r
PositionPercentage.lm <- lm(W ~ OLPercentage + DLPercentage + QBPercentage + RBPercentage + WRPercentag
summary(PositionPercentage.lm)
PositionPercentage.lm
```

```r
Position.rpart <- rpart(W ~ OLCap_Hit + DLCap_Hit + QBCap_Hit + RBCap_Hit + WRCap_Hit + TECap_Hit + LBCa
rpart.plot(Position.rpart)
```

```r
PositionPercentage.rpart <- rpart(W ~ OLPercentage + DLPercentage + QBPercentage + RBPercentage + WRPerc
rpart.plot(PositionPercentage.rpart)
```

```r
salariestrain <- SalariesFinal %>% dplyr::sample_frac(0.70)
salariestest  <- dplyr::anti_join(SalariesFinal, salariestrain)
```

```r
salarycappercent.rf <- randomForest(W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent, data = salariest
salarycappercent.rf
```

```r
plot(salarycappercent.rf)
which.min(salarycappercent.rf$mse)
sqrt(salarycappercent.rf$mse[which.min(salarycappercent.rf$mse)])
```

```r
salarycappercenttune.rf <- randomForest(W ~ ACTIVEPercent + DEADPercent + Cap_SpacePercent,
                                        data = salariestrain,
                                        ntree = 255)
salarycappercenttune.rf
```

```r
varImpPlot(salarycappercenttune.rf)
```

```r
phasepercent.rf <- randomForest(W ~ OffensePercentage + DefensePercentage + STPercentage, data = salarie
phasepercent.rf
```

```r
plot(phasepercent.rf)
which.min(phasepercent.rf$mse)
sqrt(phasepercent.rf$mse[which.min(phasepercent.rf$mse)])
```

```
varImpPlot(phasepercent.rf)
```

```
positionpercent.rf <- randomForest(W ~ OLPercentage + DLPercentage + QBPercentage + RBPercentage + WRPe
positionpercent.rf
```

```
plot(positionpercent.rf)
which.min(positionpercent.rf$mse)
sqrt(positionpercent.rf$mse[which.min(positionpercent.rf$mse)])
```

```
varImpPlot(positionpercent.rf)
```

```
catsalarycappercent.rf <- randomForest(as.factor(TenPlus) ~ ACTIVEPercent + DEADPercent + Cap_SpacePerc
catsalarycappercent.rf
```

```
varImpPlot(catsalarycappercent.rf)
```

```
salarypredictions <- predict(catsalarycappercent.rf, newdata = salariestest)
confusionMatrix(salarypredictions, as.factor(salariestest$TenPlus))
```

```
catphasepercent.rf <- randomForest(as.factor(TenPlus) ~ OffensePercentage + DefensePercentage + STPerce
catphasepercent.rf
```

```
varImpPlot(catphasepercent.rf)
```

```
phasepredictions <- predict(catphasepercent.rf, newdata = salariestest)
confusionMatrix(phasepredictions, as.factor(salariestest$TenPlus))
```

```
catpositionpercent.rf <- randomForest(as.factor(TenPlus) ~ OLPercentage + DLPercentage + QBPercentage +
catpositionpercent.rf
```

```
varImpPlot(catpositionpercent.rf)
```

```
positionpredictions <- predict(catpositionpercent.rf, newdata = salariestest)
confusionMatrix(positionpredictions, as.factor(salariestest$TenPlus))
```