NFL Football Analysis Report

By Jackson Downs

In my work, I have analyzed data from the 2019 NFL season. My goal for analyzing this data was to look for significant trends or predictors in what makes a team good; then, I wanted to create my own ranking system to rank how good these teams were actually. In the end, I found that teams with more of an intent to run and pass deep were successful in my rankings. In my report, I will begin by explaining my data collection process. Next, I’ll talk about my expectations, results and limitations from my analysis. Additionally, I’ll cover any other things I found interesting during my analysis that I thought was noteworthy. Finally, I’ll talk about what I would expect for future analysis on this subject. For my analysis I’m referring to, you can access the code through the link to Deep Note on the bottom or the Jupyter Notebook (final\_project.ipynb) on my GitHub Repository. The link to that will also be at the bottom of this report.

To begin, I got all of my data from Football Reference. However, most of the data was separated into different tables. I saved each table as an Excel Workbook, and then saved the file to my laptop as a CSV, but you can also export the tables as CSV as well (All of the CSV’s used in my analysis are in my repository). The only change I made to any of the Excel Workbook was removing the designation of playoff berths, from the AFC and NFC workbooks, as it led to me having trouble merging my data frames later. After this, I loaded my CSV into Deep Note and used the pd.read\_csv() function to begin our analysis on them. The only CSV not from Deep Note not used in our analysis was the Excel Regression CSV, which came from my own analysis. This CSV was a product of me using Regression of Excel to answer my original goal. I loaded it into my analysis just to show some preliminary research.

For expectations from our analysis, I kind of expected what any logical NFL fan would expect. The team with the most passing yards, least amount of bad throws, and most QB Pressures would be the most successful. While some teams are more one dimensional, I would expect the teams with the most wins and expected wins to lead in most key statistics.

Pivoting to the actual results, I’ll first explain about Expected Wins. The specific formula for expected wins is in my analysis, but basically it looks at a team's points scored and points allowed to calculate what a team’s winning percentage should be; we can then multiply that by 16, the amount of games in a NFL season, to get a team’s number of expected wins. For our regression using Pingouin, we used Expected Wins as our response variable. For both our heatmaps, we included Expected Wins. Overall, Expected Wins was a critical part of our analysis.

So how was Expected Wins affected by other variables? The correlation in our heatmaps and linear regression from Pingouin showed us the same thing. Expected Wins increased when teams ran the ball more, and passed the ball less; passing generally didn’t have an effect on Expected Wins. We can even see the correlations between Pass Attempts, Rushing Attempts and Expected wins on our dashboard (FinalP.py in my repository, also linked below). For my Final Ranking System, I included all variables that were statistically significant in the Pingouin regression:

* Rushing Yards Per Game
* Rushing Attempts
* Attempts Per Broken Tackle (negative coefficient)
* Receptions (negative coefficient)
* Drops (negative coefficient)
* Yards Before Contact Per Reception (negative coefficient)
* Missed Tackles (negative coefficient)
* Pass Attempts (negative coefficient)
* Expected Wins (add in due to prior importance)

For my points system, a team with the highest of a statistic with a positive coefficient from regression would earn 32 points, while the lowest would earn 1. This was switched when the variable coefficient was negative, so a team with the most missed tackles would only get 1 point, while a team with the least would get 32, for example. We can take a look at the top 6 teams in our analysis and see how they compare to the actual NFL Standings

|  |  |  |
| --- | --- | --- |
| Team Ranking | My Ranking System | Actual NFL (most wins) |
| 1st | Ravens (14) | Ravens (14) |
| 2nd | Vikings (10) | 49ers (13) |
| 3rd | Colts (7) | Saints (13) |
| 4th | 49ers (13) | Packers (13) |
| 5th | Titans (9) | Patriots (12) |
| 6th | Broncos (7) | Chiefs (12) |

In comparing my model to the actual results of the NFL’s, we can see some differences. While my model has the top 2 teams, it only includes two other playoff teams. However, those playoff teams (the Vikings and Titans) were pretty successful in the playoffs. Overall, we could chalk the differences down to small sample sizes, and the Broncos or Colts getting unlucky. Furthermore, those teams could have followed the right strategy but just not had the best personnel to succeed. The Colts did have Jacoby Brissett at Quarterback and the Broncos switched to Rookie QB Drew Lock midseason. However, I must acknowledge that for regular season success predictions, my model has some limitations.

For the first limitation, the sample size is small at 16. For example, the Titans may be as good as the Saints, but bad luck and injuries could cost them a few wins. This is even more difficult due to the fact that teams change every season, so comparing over season is difficult as well.

Additionally, like most analytics, there are a few things my model can’t take into account. My model would assume a team should run the ball or throw the ball deep, which makes sense logically, but short yardage options can be effective. What if the defense is aware of this strategy and plays two deep safeties? Some teams are far more successful with short yardage options like the Patriots with Wes Welker and Super Bowl MVP Julian Edelman. Another reason could be that a team's QB just isn’t very successful throwing the ball deep. There are a multitude of reasons why a team couldn’t execute the strategy I’ve presented. In that case, they need to make a football decision about whether to try and follow this blueprint regardless or utilize their players to their strengths and try and win that way.

One team I particularly wanted to focus on was the 2019 Super Bowl Champion Kansas City Chiefs who finished tied for 5th in the regular season for wins with 12, but the 19th ranked team in my model. I wanted to see how it was possible such a good team could finish so low in my rankings, and whether it was through a fault of my model or an anomaly.

I dove deeper into the statistics regarding the Chiefs, and found some interesting in their passing statistics. Unsurprisingly, they have the 5th most passing yards in the league, but only on the 15th most attempts. This goes along with our prior point that the Chiefs are successful with mainly deep passing plays.

However, the Chiefs low ranking comes to the fact due to the fact they’re in the bottom third in the league in rushing attempts. However, when you watch Chiefs you realize that this is because they have Patrick Mahomes. Patrick is a Super Bowl Champion and NFL MVP at the age of 25; I could list all sorts of he has broken records and is on pace to break, but it’s easiest to say he is an incredible player. When you have a player as incredible as Patrick Mahomes, it’s usually best to play to his strengths. Therefore, the Chiefs success is an anomaly due to a generational talent rather than the precedent.

In the future, I would probably condense the amount of variables I used and increase the sample size (or number of games played). While teams do change often, I feel a data size of 4 years would be long enough to get an accurate depiction of success in the NFL without being too large for league play changes to occur. Furthermore, while all the variables I included were important in my opinion, I feel that over 70 columns was excessive and made it difficult for me to focus on one area of the data; this is even after I condensed the number of variables from my original Excel regression.

Link to Repository: https://github.com/jxdowns/Final-Project.git

Link to Deep Note: https://deepnote.com/project/b32d4f71-4554-4678-a924-018c77579a5d

Link to Dashboard: http://localhost:8501