

**Using Monte Carlo Algorithms for NFL Projections**

Aaron Gregory

**I. Introduction.**

In this project, I leveraged Monte Carlo simulations to predict NFL playoff game totals and spreads based on in-season statistics. I then applied these predictions to the Vegas betting lines to attempt to identify targeted bets to take advantage of. In the first phase, I used a simple regression model to predict the total score of games. This model was then used as the framework for our Monte Carlo simulations to predict individual team scores. I was then able to extract total game total and spread predictions from our simulations.

**II. The Data**

*A. Data Scraping and Information*

To begin, we scraped game-log data from pro-football-reference for every NFL team to obtain our starting data set. A sample of the game-log data is shown below. After some cleaning the resulting data set had statistics and scores for every regular season NFL game from 2017-2021. Playoff data was left out so that it could be used as the test set. I ended up settling on nine different variables including yards turnovers and expected points. Expected points is a metric that incorporates every play over a game and how that play impacts projected points scored on a given drive. The expected points is a total of every play in the game’s expected points and is split into offense, defense and special teams. The higher the positive value the better that unit has been playing. It is important to note that for a particular game if one team has an offensive EP of 5.3 then the opponent will have a defensive EP of -5.3.



In order to utilize monte carlo techniques I needed to find the means and standard deviations of each statistic for each team. I created sperate tables for each team (for the current year) that included the mean and standard deviation for each metric in the model.

*B. Specifying Variable Probability Distributions*

After parsing each team’s webpage the means of all of the statistics were stored in a dictionary keyed by each team abbreviation as shown in appendix 2. This allows us to search for the average statistic values for each individual team. We built a similar dictionary to hold each team’s standard deviations for the same statics as shown in appendix 3 and 4. Now that we have dictionaries that store the average expected points for each team and the standard deviations of these expected points, we can randomly generate the distributions of these features for each team in our simulation and key on each individual team to quickly and easily pull the numbers we need for each matchup.

*C.* *Data Overview*

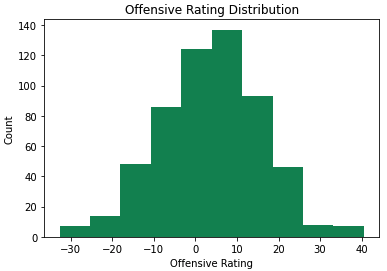
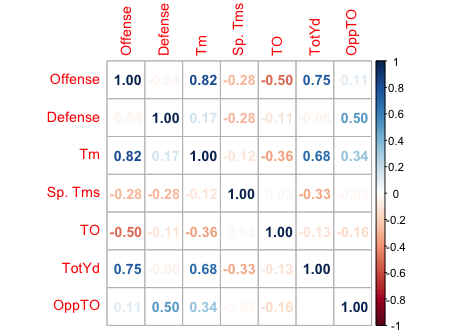
Taking a closer look at the data we see that most features are normally distributed around the mean. The only features that aren’t normally distributed are turnovers and opponents turnovers which have an exponential distribution. It should be noted that offensive and defensive expected points are centered around 4.5 and -4.5 respectively. This shows that in general offenses are more like likely to have positive plays than not.

Figure 1: DISTRIBUTION OF OFFENSIVE RATING EP

Another interesting aspect to look at is the correlation matrix and how features are correlated with one another. To the left you can see that offensive EP and total yards are most highly correlated with total team points (Tm on the graph). This is to be expected but it is also valuable to note that offensive EP and total yards are also highly correlated which will have to be accounted for in our simulations.

Figure 2: Correlation Matrix (Where Tm is total team points)

**IV. Leveraging a Linear Regression Model to Simulate Team Scores**

*A. Developing the Model*

|  |  |  |
| --- | --- | --- |
| Model Type | RMSE | R2 |
| Linear Regression | 4.71 | 0.79 |
| Ridge Regression | 4.62 | 0.80 |
| Lasso Regression | 4.62 | 0.80 |

Using the game log data from every regular season NFL game we modeled the data to predict a team’s score using linear, ridge and lasso regression techniques. By splitting the data into a training and test set we were able to look for overfitting of the data. The results are described in table 1.

Table : Modeling Results from testing data

These results show that all three regression techniques yield close to the same accuracy leading us to choose the linear regression model as it is the simplest. Looking further into the linear model we also removed all features that were not significant per the t-test as shown in appendix 6.

The final model has six features including all three expected points statistics, turnovers for both teams and total offense yards. As discussed earlier there is some correlation between total yards and offensive expected points but there was no overfitting or negative effects seen in the testing data to give concern.

****

The a plot of the residuals (appendix 7) shows that the model is most accurate around the mean but over projects games in which a team scores very little and under projects games in which a team scores a lot of points. The reason that this is not a problem for us is that we want to compare our results to the what the gambling line is which is not as extreme as these results. Most betting line for team totals range from 15-30 points. If a team goes out and scores 55 points we just want our model to have a projection that is higher than the betting line so we don’t necessarily need to predict 55 points as long as we predict higher than the betting line. Another thing to consider is that football has a huge variance and lots of unpredictable events, this is why people love it so much. Our goal is to make a profit not be perfect and in order to make a profit gambling you only need to hit 52.4% of the bets you place.

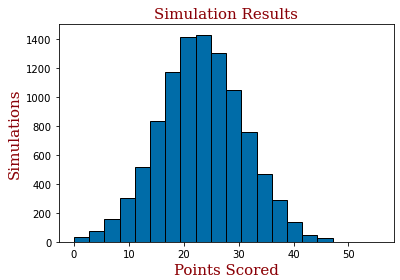
*B. Building out the Simulation*

Once we had selected our model we could begin building out the monte carlo simulation. For each feature in the model we calculated a random number from the distribution and substituted it into our model to generate a total team score. Table 2 shows the variables needed, their distributions and what the inputs needs to calculate a random number in that distribution.

|  |  |  |
| --- | --- | --- |
| Variable | Distribution | Inputs |
| Offensive EP | Correlated Normal | Average, Covariance Matrix |
| Defensive EP | Normal | Average, Standard Deviation |
| Sp. Teams EP | Normal | Average, Standard Deviation |
| Turnovers | Exponential | Average, Standard Deviation |
| Opponents Turnovers | Exponential | Average, Standard Deviation |
| Total Yards | Correlated Normal | Average, Covariance Matrix |

Table : Distributions and inputs for random number generation

As discussed previously offensive EP and total yards have a correlation of 0.75. To account for this in our simulation we used the covariance matrix of the two to help generate correlated values for each simulation. Because we only used one year of game logs we decided that it would be best to use the covariance matrix of the entire NFL in junction with the teams specific average offensive EP and total yards to generate each feature variable. Although using the covariance matrix from all NFL games does add bias we believe that it is the best option due to having such a small sample size to work with.

A key feature that was added to our simulation that was not in the model was the consideration of defensive impact on offensive EP and visa-versa. Because offensive EP and defensive EP are inverses of each other the average of the offensive EP and opponents defensive EP was taken for a team’s offensive EP projection. The opposite was done for their defensive EP. This alongside opponent turnovers are the main differences in the simulation between different teams. The code for this calculation is in appendix 8.

In summary this model took two teams that were playing each other and then simulated the score 10,000 times outputting a histogram of each teams expected points along with the projected total and the projected spread (as shown in appendix 9). The spread was calculated by taking the team with the higher projected score and subtracting the lower projected score.

Figure 4: Output histogram of team scoring

*C. Assessing the Model’s Results*

The usefulness of our model is dependent on our ability to use its predictions to bet against betting lines and totals. We used the model with the 2021 NFL regular season data to simulate every playoff matchup from this past year as shown in table 3. The model did a better job predicting the total score than the predicted spread. The three games that the model failed to project the total for either had overtime or were a blowout. The models record predicting against the spread improves to 5-4 if you implement a threshold of 3 points (meaning the difference between the betting line is more than 3 points). By using this threshold and betting on both the total and spread a bettor betting $100 bets would have profited $730 or a 37% ROI. This sounds great but in reality this is still a very small sample size so it may be prudent to try this over multiple years before making any decisions on the effectiveness of the model.

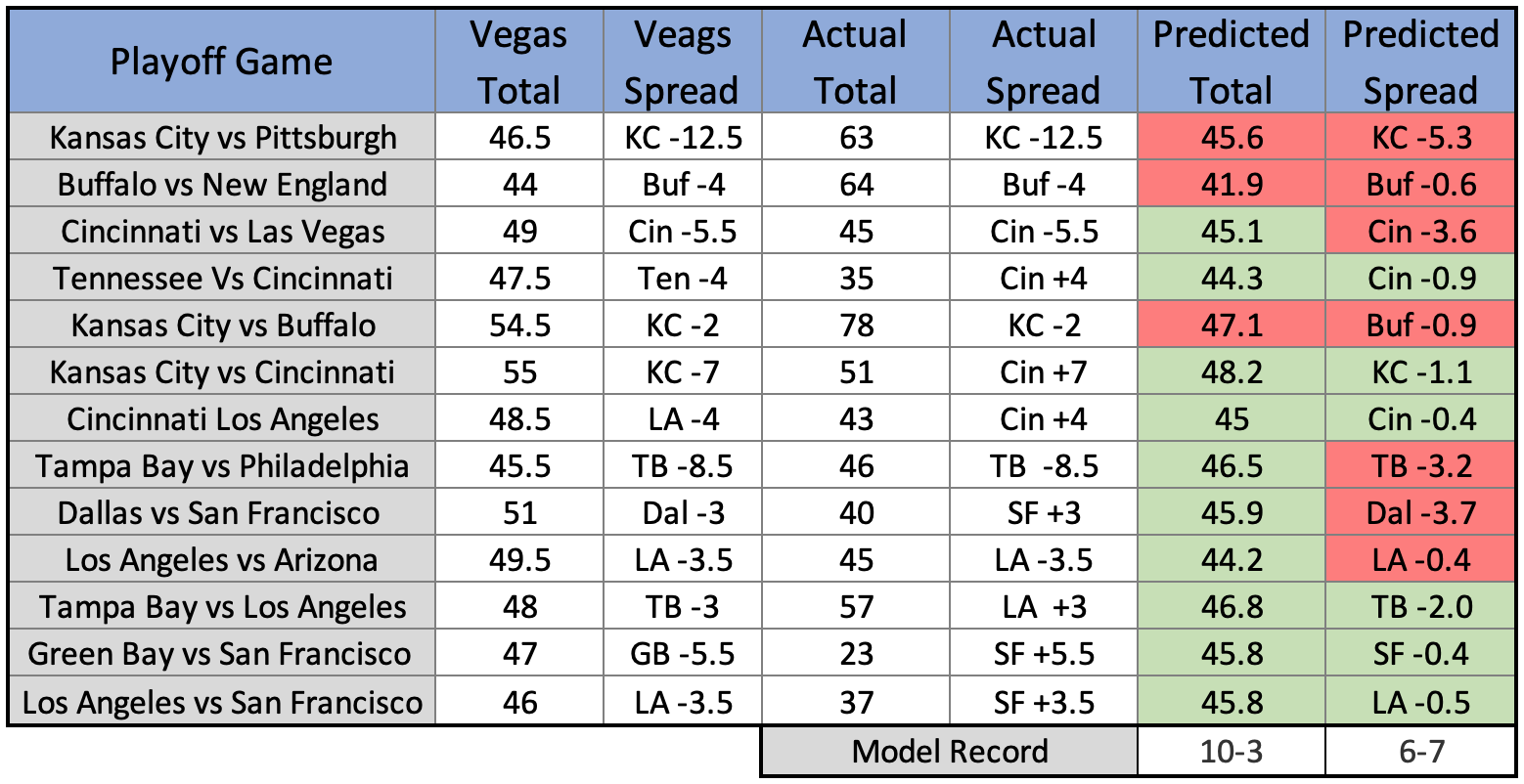


Table 3:Models results for the 2021-22 NFL Playoffs

*D. Possible Model Improvements*

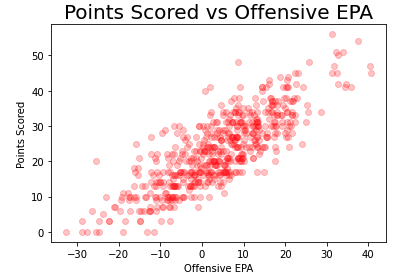
One clear issue with using this model is that it is not particularly usable until very late in the season since the averages and standard deviations will vary greatly early in the year. NFL teams have high player turnover which greatly increases the variability of team performance from one year to the next but if you were able to adjust a teams expected performance maybe this model would be able to be used earlier. Another issue with the model goes back to decreasing the larger residuals at the scoring extremes. One possible way to do this would be to try and implement a cubic or exponential aspect to the model in regards to offensive EP. The scatter chart to the right shows that very high and very low offensive EP do correlate with the extreme scores and is a possible opportunity for improvement. Lastly we could go back and simulate more postseason for a bigger sample size to judge the model on.

Figure 5:Offensive EP vs Team score

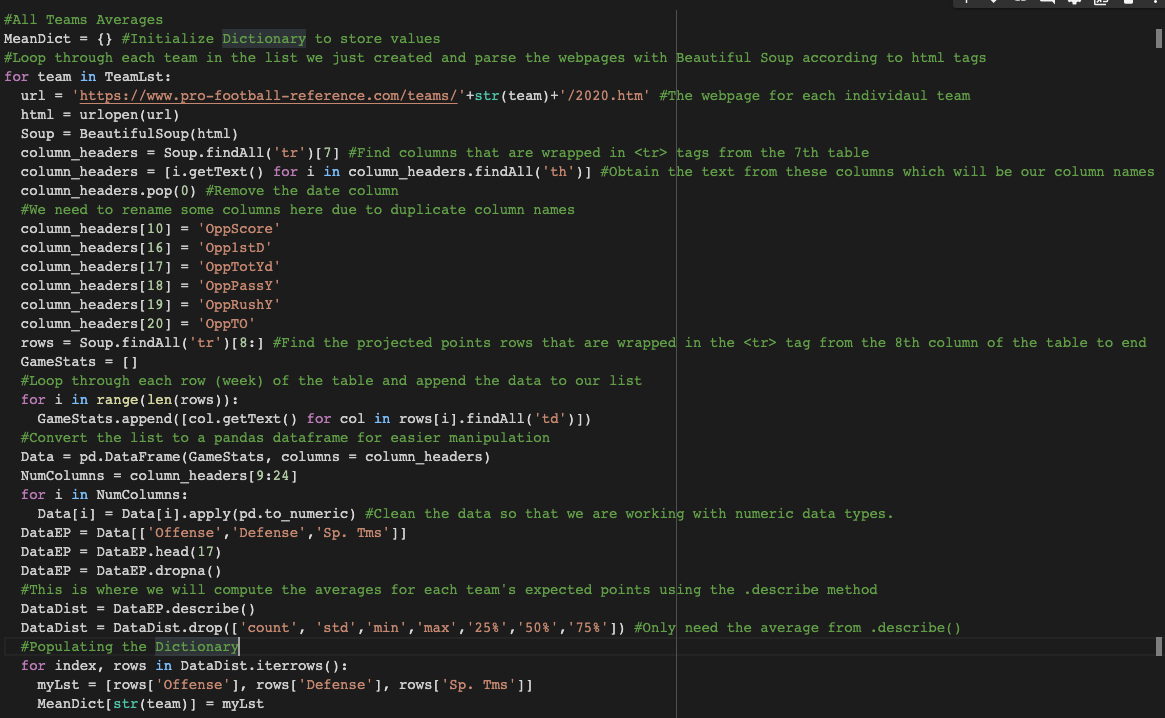
**V. Future Areas of Analysis**

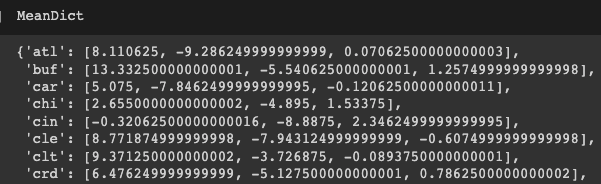
Expected points could be broken down to expected points average by positional group. This could perhaps unlock additional predictive power in our initial machine learning model along with increasing the model’s early season predication ability. Including specific player data and including home/away effects as features in the model could also introduce additional complexity to this model which could have additive effects to it’s predictive power.

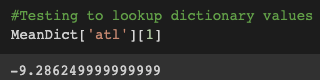
Regarding the simulation of Team Win Totals, the simulated matchups for each team are independent of one another. I.e., there is not one loss for every win and vice versa across team simulations. This causes less correlation in results and is evidenced with most team win predictions centering around the midpoint of 8-9 wins for the season. This model also does not take into account offseason improvements or deteriorations made by each team between the end of the previous season and the beginning of the predicted season. It is simply taking the expected points distribution from the prior season which is great for model simplicity and the interpretability of results but additional complexity would most likely be beneficial here.

**VI. Appendix**

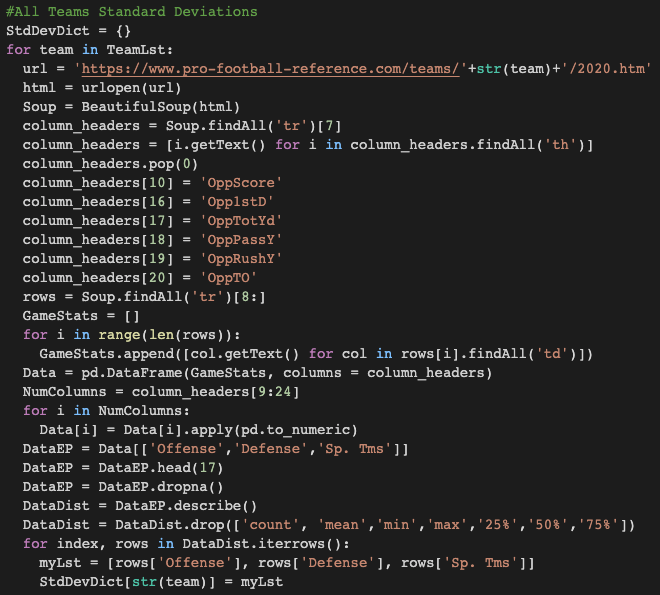
1. Example of Beautiful Soap to Gather Team Data

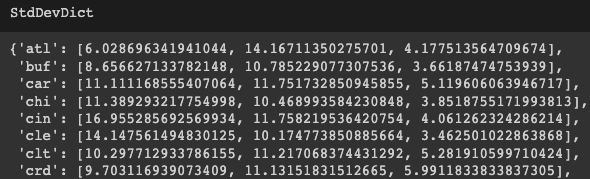
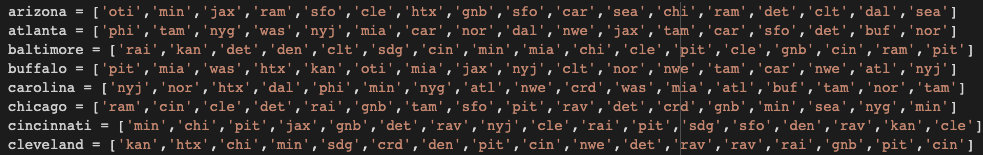


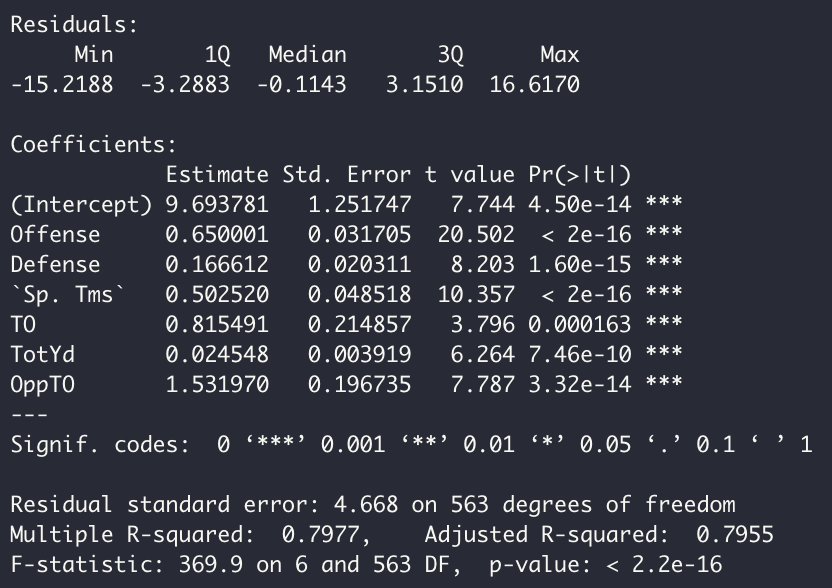
1. Example of Means Dictionary



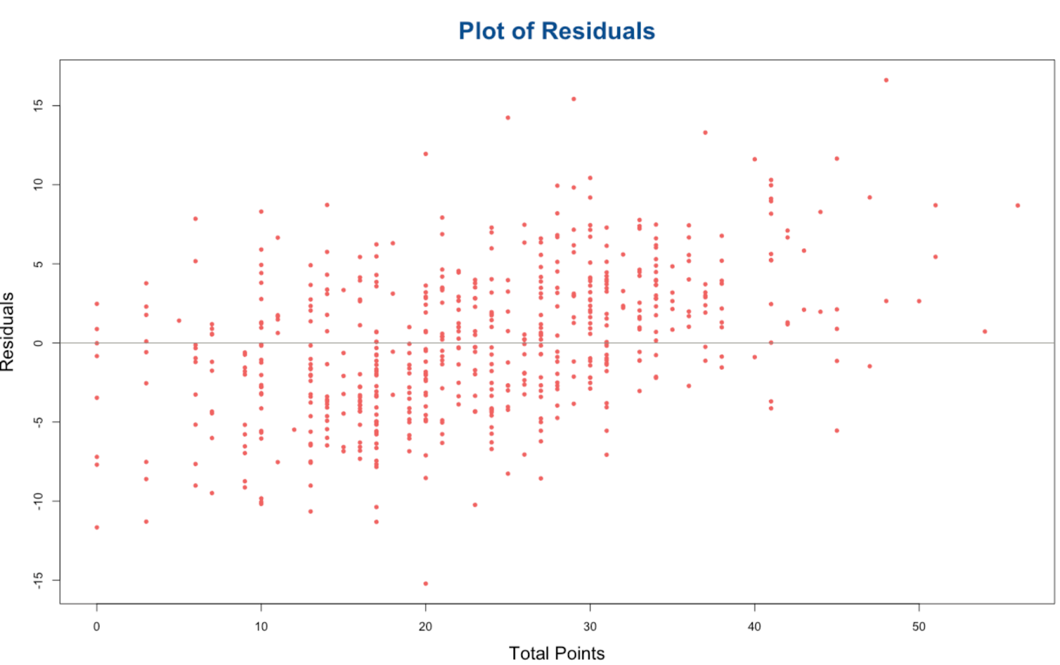
1. Example Code to Create Standard Deviations Dictionary



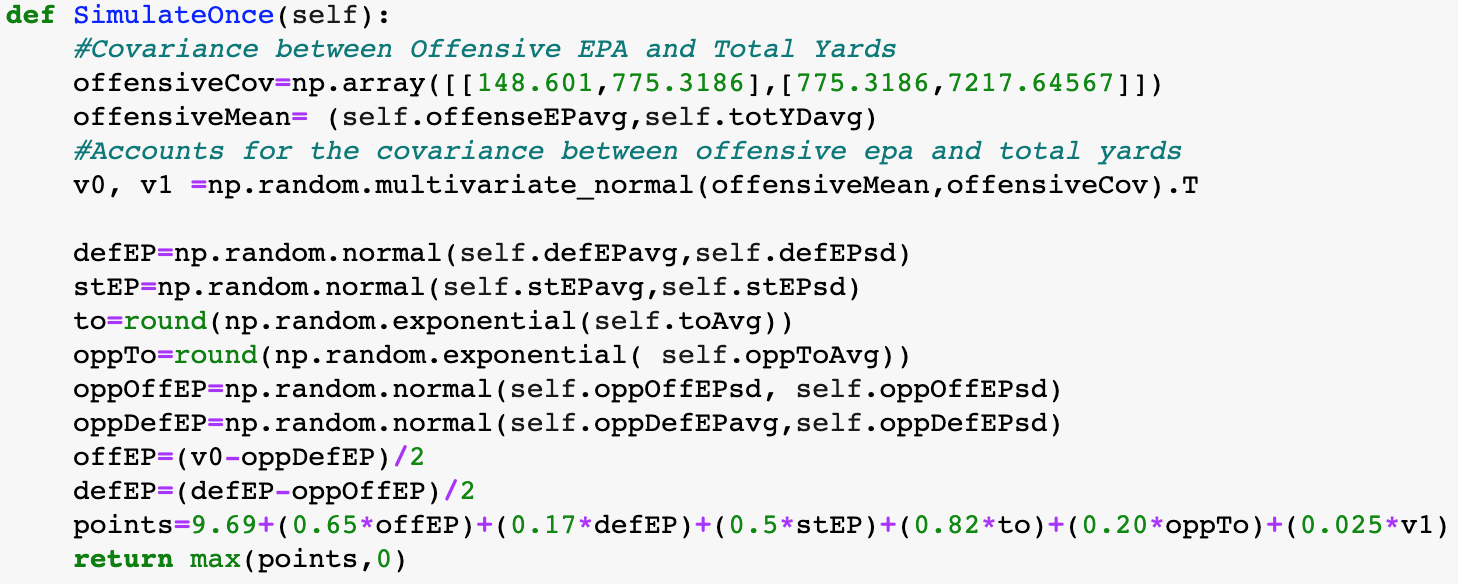
1. Standard Deviations Dictionary Output
2. Example of Team Schedules
3. Linear Regression Output From R



1. Plot of Residuals vs Total Team Points



1. Code for simulateOnce



1. Code for a Game Simulation

