Team Members: Tiffany, J.R., Tom, Bao

March 27, 2021

1. Introduction
   1. For this project, we are continuing our quantitative analysis of Fantasy (*Pandasy*) Football Data. Specifically, we are building a comprehensive SQL Database derived from the last five seasons of player and scoring statistics. Our goal is to assemble a cohesive database in which the average fantasy football fanatic can gain utility and knowledge from simply clicking around!
2. Data
   1. Description

The original data was obtained from two sources: an API from Sports Data, and CSV files from Fantasy Football Data Pros. There is a plethora of data from both sources. Focusing on the individual players, the data from the API included fantasy data and positional data, but also included physical attributes like age, height, weight. It also included non-fantasy relevant data such as experience. The CSV data contained fantasy and positional data for the NFL seasons from 2015-2019.

There are four positions that contained the most data for fantasy football: Quarterback, Running Back, Wide Receiver, and Tight End. Each position has specific statistics that contribute to their fantasy football score i.e. Quarterbacks and passing yards, Running Back and rushing touchdowns, or Wide Receivers and yards after the catch. There is some crossover in the statistics as some positional players are proficient enough to gain fantasy points outside of the fields generally attributed to the position.

* 1. Limitations

While the API file contained a unique Player ID for each player, the csv files did not. Therefore, we had to merge the data on the players’ names. This became problematic when there was more than one player with the same first and last names. Due to this limitation, we had to drop “duplicate” players, when in actuality, they were not the same person, leaving us unable to use the data from those particular players.

Fantasy football teams are generally composed of a mix of the following positions: Quarterback, Running Back, Wide Receiver, Tight End, Kicker, Defense Teams. The CSV files did not contain information on Defensive Players. The API contained data on Defensive players. On the Join, Defensive Players and Kickers were excluded from the results.

The data is also from a five year span, but does not include the most recent data from the 2020 season. Due to the global pandemic, the fantasy data is skewed for that year because players sat out of the season and teams were quarantined, forcing games to be shifted. Being an atypical season, the most recent data used for this project is two years old.

Some historical data, positional data, and unique ids were not accessed by this project due to cost restrictions.

1. Analysis/Methodology
   1. Initially, we used Jupyter Notebook as a means of importing, cleaning and organizing our data. We imported 5 CSVs of Fantasy-specific football data and called an API for other player-specific data points. Given that we were combining five seasons of NFL Player & Scoring data, it was imperative that our tables were intuitive and accurate.

First and foremost, we had to clean our overabundance of data. While we had access to thousands of individual data-points via our API, we decided to hone in on simple quantitative figures like experience, height, and weight. We supplemented those metrics with qualitative attributes like “Team” and “Position”. However, we noticed as players retire, change teams, and get hurt, their statistics are generally rendered insignificant. To avoid distortions in our analysis, we removed all rows with “NaN” values. Similarly, some players had a position equal to “0”. That was clearly an error, so we went ahead and removed those data points as well.

Next, our imported API data had an odd method of listing players’ height. In fact, player height was separated into two columns: “Feet” & “Inches”, making it difficult to maneuver and compare with other data points. By multiplying our “Feet” column by 12 and adding it to our “Inches” column, we were able to create one singular “Height” column. We would go ahead and clean this segment of our data by deleting the two original “Feet” and “Inches” columns.

Finally, we built a connection to Postgres in order to view our tables. Within Postgres, we had built tables featuring a player’s team, position, age and total fantasy points for each NFL season dating back to 2015. Additionally, we built a separate table of physical player attributes featuring data on a player’s height, weight, experience, and their average draft position. Using these tables, we amalgamated our datasets into one singular view, “player\_alldata”. In joining the player data from the API with the Fantasy-specific data found in the CSV files, we removed all defensive (non-skill) players from our dataset.

1. Results
   1. By piecing together this mammoth dataset, we were able to create descriptive “Views” of our dataset through a query of “player\_alldata”. These views enabled us to easily sift through our dataset by combining all five seasons of fantasy data. The cardinal use of this tool would allow us to sort the dataset by player, to display that player’s fantasy football statistics from the last five seasons. As our queries show, we were able to find descriptive statistics from the last five years such as the average height of quarterbacks and the maximum and minimum age of wide receivers.
2. Conclusions
   1. Theoretically, anyone with this tool could locate or calculate any statistic of the last five fantasy football seasons. The competitive fantasy football player may find utility in this tool by running queries to identify major trends and outliers. Given what we’ve learned in our original analysis, a competitive fantasy football player may seek taller wide receivers, or more weathered quarterbacks, during their upcoming fantasy football drafts.
3. Appendix

Sports Data API:

*https://api.sportsdata.io/v3/nfl/scores/json/Players?key=*

Fantasy Football Data Pros:

*https://www.fantasyfootballdatapros.com/csv\_files*