Predicting Fantasy Football Draft Rankings: Unraveling the Enigma

Brian Reppeto

2023-08-7

Table of Contents

1. Introduction

Fantasy football is a popular online game where participants act as team managers and draft real players from the National Football League (NFL) to form fantasy teams. The performance of the actual stats of the players in real-life games determines the fantasy team's score. Participants compete against one another on a weekly head to head basis and these scores determine a weekly winner. One of the most important aspects of fantasy football is the drafting phase, which takes place before the NFL regular season starts. Participants select players based on their projected performance and rankings. I will use the redraft type of league along with a snake draft type of draft. Additionally, I will use the standard scoring system, which is 1 point per 25 passing yards, 4 points per passing touchdown, 1 point per 10 rushing or receiving yards, 6 points per rushing or receiving touchdown, -2 points per fumble lost or interception.

The issue I want to research is being able to help predict fantasy football drafting rankings. Specifically, drafting quarterbacks, running backs, wide receivers, and tight ends. The goal is to develop a model or methodology that can help accurately predict the ending rankings of NFL players for fantasy football drafts. This project would be of particular interest to fantasy football participants, as it can help them better draft a fantasy teams for their season. By leveraging data science techniques, I can analyze historical player performance and other relevant data to make informed predictions for the upcoming season's draft rankings.

Problem statement

Within this vibrant tapestry of fantasy football, a central enigma persists—the challenge of accurately predicting the future performance and rankings of NFL players in fantasy drafts. The nebulous interplay of player statistics, team dynamics, and unforeseen factors complicates this task. Our research aims to decipher this puzzle, focusing on forecasting draft rankings for pivotal positions: quarterbacks, running backs, wide receivers, and tight ends. This endeavor seeks to empower fantasy football enthusiasts by arming them with insights grounded in data science, ultimately enhancing their draft strategies and bolstering their chances of success in the league.

The core issue revolves around the unpredictability of fantasy football player draft rankings. The rankings and draft shape the core of the team that a player will play with during the season and understanding how to draft a better team can help win a league. This research will focus on understanding quarterbacks, running backs, wide receivers, and tight ends as these positions generate the majority of points and take up 90 of a fantasy team roster.

2. Approach

Data Collection and Cleaning

The foundation of any data-driven analysis lies in the quality and relevance of the data collected. In this phase, I embarked on a meticulous process of data collection, gathering an extensive dataset that spans multiple seasons (2019-2022). The dataset included player statistics, team standings, strength of schedule (SOS), and offensive team statistics. Each data source was carefully selected to provide a comprehensive view of player performance, team dynamics, and external factors that could influence fantasy football draft rankings.

1. Data Collection: I gathered an extensive dataset of player statistics from past seasons (2019-2022). I also gathered datasets for strength of schedule (SOS) and the final standings of the 32 NFL teams from 2022. The historical fantasy data helped tell the story of individual performances during this time.
2. Data Cleaning and Consolidation: The data was cleaned and consolidated, ensuring that each player's performance was accurately represented over time. This step was critical to ensure the integrity of our analysis and the validity of our conclusions. This step of cleaning and consolidating the datasets was very time consuming as the data has to be standardized and consolidated from multiple different sources.
3. Exploratory Analysis: Diving into the data, unveiled trends, patterns, and relationships. I investigated the relationship between player performance and variables like team SOS (2019-2022) and team standings (2022).
4. Linear Regression Analysis: To assess the predictive potential of the two variables, I employed a linear regression analysis. I tested the correlation between player fantasy points and team SOS (2019- 2022), as well as the relationship between fantasy points and team standings (2022). The analyses provided insight into the connections between player performance and these variables.
5. Model Recommendations: While my primary focus was on exploration and analysis, I was able to glean insights from my readings that a Random Forest Regression Model (RFRM) might work best with this set of data. A RFRM would help with the complex relationships between player performance and other variables that might influence the performance. Additionally, a RFRM could help with possible over fitting and outliers from individual player performances.

Data

```r
head(QB_stats)
```

```
## # A tibble: 6 x 22
##    Rank Player      Season_team Current_team   CMP   ATT   PCT   YDS   Y.A    TD
##   <dbl> <chr>       <chr>       <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 Josh Allen  BUF         Buffalo Bil~   409   646  63.3  4407   6.8    36
## 2     2 Justin Her~ LAC         Los Angeles~   443   672  65.9  5014   7.5    38
## 3     3 Tom Brady   TB          NO TEAM FA     485   719  67.5  5316   7.4    43
## 4     4 Patrick Ma~ KC          Kansas City~   436   658  66.3  4828   7.3    37
## 5     5 Matthew St~ LAR         Los Angeles~   404   601  67.2  4886   8.1    41
## 6     6 Aaron Rodg~ GB          New York Je~   366   531  68.9  4115   7.7    37
```

```
## # i 12 more variables: INT <dbl>, SACKS <dbl>, ATT.1 <dbl>, YDS.1 <dbl>,
## #   TD.1 <dbl>, FL <dbl>, G <dbl>, FPTS <dbl>, FPTS.G <dbl>, ROST <dbl>,
## #   Fantasy_yr <dbl>, Posit <chr>
```

```r
head(RB_stats)
```

```
## # A tibble: 6 x 22
##    Rank Player      Season_team Current_team   ATT   YDS   Y.A    LG  X20.    TD
##   <dbl> <chr>       <chr>       <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 Jonathan T~ IND         Indianapoli~   332  1811   5.5    83    30    18
## 2     2 Austin Eke~ LAC         Los Angeles~   206   911   4.4    28     3    12
## 3     3 Joe Mixon   CIN         Cincinnati ~   292  1205   4.1    32     7    13
## 4     4 Najee Harr~ PIT         Pittsburgh ~   307  1200   3.9    37     8     7
## 5     5 James Conn~ ARI         Arizona Car~   202   752   3.7    35     4    15
## 6     6 Ezekiel El~ DAL         NO TEAM FA     237  1002   4.2    47     5    10
## # i 12 more variables: REC <dbl>, TGT <dbl>, YDS.1 <dbl>, Y.R <dbl>,
## #   TD.1 <dbl>, FL <dbl>, G <dbl>, FPTS <dbl>, FPTS.G <dbl>, ROST <dbl>,
## #   Fantasy_yr <dbl>, Posit <chr>
```

```r
head(WR_stats)
```

```
## # A tibble: 6 x 21
##    Rank Player      Season_team Current_team   REC   TGT   YDS   Y.R    LG  X20.
##   <dbl> <chr>       <chr>       <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 Cooper Kupp LAR         Los Angeles~   145   191  1947  13.4    59    57
## 2     2 Deebo Samu~ SF          San Francis~    77   121  1405  18.2    83    53
## 3     3 Ja'Marr Ch~ CIN         Cincinnati ~    81   128  1455  18      82    49
## 4     4 Justin Jef~ MIN         Minnesota V~   108   167  1616  15      56    45
## 5     5 Davante Ad~ GB          Las Vegas R~   123   169  1553  12.6    59    37
## 6     6 Mike Evans  TB          Tampa Bay B~    74   114  1035  14      46    30
## # i 11 more variables: TD <dbl>, ATT <dbl>, YDS.1 <dbl>, TD.1 <dbl>, FL <dbl>,
## #   G <dbl>, FPTS <dbl>, FPTS.G <dbl>, ROST <dbl>, Fantasy_yr <dbl>,
## #   Posit <chr>
```

```r
head(TE_stats)
```

```
## # A tibble: 6 x 21
##    Rank Player      Season_team Current_team   REC   TGT   YDS   Y.R    LG  X20.
##   <dbl> <chr>       <chr>       <chr>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 Mark Andre~ BAL         Baltimore R~   107   153  1361  12.7    43    27
## 2     2 Travis Kel~ KC          Kansas City~    92   134  1125  12.2    69    20
## 3     3 Dalton Sch~ DAL         Houston Tex~    78   104   808  10.4    32     8
## 4     4 George Kit~ SF          San Francis~    71    94   910  12.8    48    17
## 5     5 Rob Gronko~ TB          NO TEAM FA      55    88   802  14.6    42    20
## 6     6 Dawson Knox BUF         Buffalo Bil~    49    71   587  12      53    14
## # i 11 more variables: TD <dbl>, ATT <dbl>, YDS.1 <dbl>, TD.1 <dbl>, FL <dbl>,
## #   G <dbl>, FPTS <dbl>, FPTS.G <dbl>, ROST <dbl>, Fantasy_yr <dbl>,
## #   Posit <chr>
```

- footballdb (2023a)
- footballdb (2023b)

```r
tail(SOS)
```

```
## # A tibble: 6 x 7
##   ranking Team  rating    Hi   Low  Last  Year
##     <dbl> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl>
```

```
## 1       27 SF       -0.5      9     32     26  2022
## 2       28 MIN      -0.6      3     28     27  2022
## 3       29 IND      -0.7     10     30     28  2022
## 4       30 CAR      -0.8     23     32     30  2022
## 5       31 HOU      -0.9     16     32     31  2022
## 6       32 ATL      -1       15     32     32  2022
```

- Strength of schedule (2023c)

```r
head(team_stand_22)
```

```
## # A tibble: 6 x 7
##   Team      W     L     T   Pct    PF    PA
##   <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ARI       4    13     0 0.235   340   449
## 2 ATL       7    10     0 0.412   365   386
## 3 BAL      10     7     0 0.588   350   315
## 4 BUF      13     3     0 0.813   455   286
## 5 CAR       7    10     0 0.412   347   374
## 6 CHI       3    14     0 0.176   326   463
```

- Team schedule (2023d)

Sumamry stats

```r
library(dplyr)
# Group data by Player and calculate summary statistics
QB_summary_data <- QB_stats %>%
  group_by(Player,Posit, Fantasy_yr,Season_team) %>%
  #summarize(Avg_fant_pts = round(mean(FPTS))) %>%
  summarize(FPTS)
```

```
## `summarise()` has grouped output by 'Player', 'Posit', 'Fantasy_yr'. You can
## override using the `.groups` argument.
```

```r
  #arrange(desc(Avg_fant_pts))
print(QB_summary_data)
```

```
## # A tibble: 391 x 5
## # Groups:   Player, Posit, Fantasy_yr [391]
##    Player           Posit Fantasy_yr Season_team  FPTS
##    <chr>            <chr>      <dbl> <chr>       <dbl>
##  1 AJ McCarron      QB          2019 HOU          17.9
##  2 AJ McCarron      QB          2020 HOU           0.8
##  3 Aaron Rodgers    QB          2019 GB          282
##  4 Aaron Rodgers    QB          2020 GB          387.
##  5 Aaron Rodgers    QB          2021 GB          336.
##  6 Aaron Rodgers    QB          2022 GB          251.
##  7 Adam Froman      QB          2022 <NA>          0
##  8 Adrian Martinez  QB          2022 <NA>          0
##  9 Aidan O'Connell  QB          2022 <NA>          0
## 10 Alex McGough     QB          2022 <NA>          0
## # i 381 more rows
```

```r
library(dplyr)
# Group data by Player and calculate summary statistics
RB_summary_data <- RB_stats %>%
```

```r
  group_by(Player,Posit, Fantasy_yr,Season_team) %>%
  #summarize(Avg_fant_pts = round(mean(FPTS))) %>%
   summarize(FPTS)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'Player', 'Posit', 'Fantasy_yr',
## 'Season_team'. You can override using the `.groups` argument.
```

```r
  #arrange(desc(Avg_fant_pts))
print(RB_summary_data)
```

```
## # A tibble: 847 x 5
## # Groups:   Player, Posit, Fantasy_yr, Season_team [846]
##    Player             Posit Fantasy_yr Season_team  FPTS
##    <chr>              <chr>      <dbl> <chr>       <dbl>
##  1 AJ Dillon          RB          2020 GB           38.3
##  2 AJ Dillon          RB          2021 GB          152.
##  3 AJ Dillon          RB          2022 GB          140.
##  4 Aaron Jones        RB          2019 GB          266.
##  5 Aaron Jones        RB          2020 GB          212.
##  6 Aaron Jones        RB          2021 GB          177
##  7 Aaron Jones        RB          2022 GB          190.
##  8 Adam Prentice      RB          2021 NO            2.1
##  9 Adam Prentice      RB          2022 NO            1.8
## 10 Adrian Killins Jr. RB          2020 <NA>         -1
## # i 837 more rows
```

```r
library(dplyr)
# Group data by Player and calculate summary statistics
WR_summary_data <- WR_stats %>%
  group_by(Player,Posit , Fantasy_yr, Season_team) %>%
  #summarize(Avg_fant_pts = round(mean(FPTS))) %>%
  #arrange(desc(Avg_fant_pts))
   summarize(FPTS)
```

```
## `summarise()` has grouped output by 'Player', 'Posit', 'Fantasy_yr'. You can
## override using the `.groups` argument.
```

```r
print(WR_summary_data)
```

```
## # A tibble: 1,268 x 5
## # Groups:   Player, Posit, Fantasy_yr [1,268]
##    Player         Posit Fantasy_yr Season_team  FPTS
##    <chr>          <chr>      <dbl> <chr>       <dbl>
##  1 A.J. Brown     WR          2019 TEN         165.
##  2 A.J. Brown     WR          2020 TEN         178.
##  3 A.J. Brown     WR          2021 TEN         118.
##  4 A.J. Brown     WR          2022 PHI         212.
##  5 A.J. Green     WR          2019 <NA>          0
##  6 A.J. Green     WR          2020 CIN          64.3
```

```
##   7 A.J. Green     WR          2021 ARI         103. 
##   8 A.J. Green     WR          2022 ARI          37.6
##   9 A.T. Perry     WR          2022 <NA>          0  
##  10 Adam Humphries WR          2019 TEN          49.5
## # i 1,258 more rows
```

```r
library(dplyr)
# Group data by Player and calculate summary statistics
TE_summary_data <- TE_stats %>%
  group_by(Player,Posit , Fantasy_yr, Season_team) %>%
  #summarize(Avg_fant_pts = round(mean(FPTS))) %>%
  #arrange(desc(Avg_fant_pts))
   summarize(FPTS)
```

```
## `summarise()` has grouped output by 'Player', 'Posit', 'Fantasy_yr'. You can
## override using the `.groups` argument.
```

```r
print(TE_summary_data)
```

```
## # A tibble: 715 x 5
## # Groups:   Player, Posit, Fantasy_yr [715]
##    Player              Posit Fantasy_yr Season_team  FPTS
##    <chr>               <chr>      <dbl> <chr>       <dbl>
##  1 Adam Shaheen        TE          2019 CHI           7.4
##  2 Adam Shaheen        TE          2020 MIA          33  
##  3 Adam Shaheen        TE          2021 CHI          11  
##  4 Adam Trautman       TE          2020 NO           23.1
##  5 Adam Trautman       TE          2021 NO           36.3
##  6 Adam Trautman       TE          2022 NO           26.7
##  7 Alan Cross          TE          2019 <NA>          0  
##  8 Albert Okwuegbunam TE          2020 DEN           18.1
##  9 Albert Okwuegbunam TE          2021 DEN           43  
## 10 Albert Okwuegbunam TE          2022 DEN           15.5
## # i 705 more rows
```

```r
all_players <- rbind(QB_summary_data, RB_summary_data, WR_summary_data, TE_summary_data)#%>%
#filter (Season_team == "ARI")
print(all_players)
```

```
## # A tibble: 3,221 x 5
## # Groups:   Player, Posit, Fantasy_yr [3,220]
##    Player           Posit Fantasy_yr Season_team  FPTS
##    <chr>            <chr>      <dbl> <chr>       <dbl>
##  1 AJ McCarron      QB          2019 HOU          17.9
##  2 AJ McCarron      QB          2020 HOU           0.8
##  3 Aaron Rodgers    QB          2019 GB          282  
##  4 Aaron Rodgers    QB          2020 GB          387. 
##  5 Aaron Rodgers    QB          2021 GB          336. 
##  6 Aaron Rodgers    QB          2022 GB          251. 
##  7 Adam Froman      QB          2022 <NA>          0  
##  8 Adrian Martinez QB          2022 <NA>          0  
##  9 Aidan O'Connell QB          2022 <NA>          0  
## 10 Alex McGough     QB          2022 <NA>          0  
## # i 3,211 more rows
```

```r
all_sum_team <- all_players %>%
  group_by(Season_team, Fantasy_yr)%>%
```

```r
  select(Season_team, Fantasy_yr, FPTS)%>%
  summarize(ttl_FPTS = sum(FPTS))%>%
  arrange(desc(Season_team))%>%
  filter (Season_team != "NA")
```

```
## `summarise()` has grouped output by 'Season_team'. You can override using the
## `.groups` argument.
```

```r
print(all_sum_team)
```

```
## # A tibble: 128 x 3
## # Groups:   Season_team [32]
##    Season_team Fantasy_yr ttl_FPTS
##    <chr>            <dbl>    <dbl>
##  1 WAS               2019     712.
##  2 WAS               2020     921.
##  3 WAS               2021     924.
##  4 WAS               2022    1016.
##  5 TEN               2019    1070.
##  6 TEN               2020    1288.
##  7 TEN               2021    1065.
##  8 TEN               2022     890.
##  9 TB                2019    1243
## 10 TB                2020    1158.
## # i 118 more rows
```

```r
sos_rank <- SOS %>%
  group_by(Team, Year)%>%
  select(Team, Year, Last)%>%
  arrange(Team,Year)%>%
  filter (Team != "NO TEAM FA")
  #summarize(avg_sos  = round(mean(ranking)))
# the higher the ranking the harder the schedule i.e 1 is the hardest

print(sos_rank)
```

```
## # A tibble: 128 x 3
## # Groups:   Team, Year [128]
##    Team   Year  Last
##    <chr> <dbl> <dbl>
##  1 ARI    2019    13
##  2 ARI    2020    21
##  3 ARI    2021     5
##  4 ARI    2022    13
##  5 ATL    2019     8
##  6 ATL    2020     2
##  7 ATL    2021    28
##  8 ATL    2022    32
##  9 BAL    2019     7
## 10 BAL    2020    16
## # i 118 more rows
```

```r
library(dplyr)


all_sum_team <- all_sum_team %>%
```

```r
  left_join(sos_rank, by = c("Season_team" = "Team", "Fantasy_yr" = "Year"))
  #select(-ranking, -Last.x)  # Removing duplicate columns

print(all_sum_team)
```

```
## # A tibble: 128 x 4
## # Groups:   Season_team [32]
##    Season_team Fantasy_yr ttl_FPTS  Last
##    <chr>            <dbl>    <dbl> <dbl>
##  1 WAS               2019     712.    29
##  2 WAS               2020     921.    27
##  3 WAS               2021     924.     6
##  4 WAS               2022    1016.    19
##  5 TEN               2019    1070.     2
##  6 TEN               2020    1288.    22
##  7 TEN               2021    1065.    24
##  8 TEN               2022     890.    18
##  9 TB                2019    1243     15
## 10 TB                2020    1158.     1
## # i 118 more rows
```

3. Analysis

Performance Patterns Across Positions

The initial phase of analysis revolved around understanding the performance patterns across different player positions. The heatmap of positional scores provided a visual representation of the average fantasy points earned by quarterbacks (QBs), running backs (RBs), wide receivers (WRs), and tight ends (TEs) over the seasons 2019-2022. This visualization revealed distinct patterns in point distribution among these positions. Notably, it underscored the scarcity of high-scoring RBs, indicating a potential draft strategy of prioritizing top-performing RBs early due to their scarcity.

Influence of Team Success and Strength of Schedule

One of the key insights we uncovered was the profound influence of team dynamics on player performance. By examining the relationship between team success and player fantasy points, we observed that players belonging to successful teams tended to generate higher fantasy point outputs. This phenomenon highlights the interplay between team success and individual player performance, indicating that drafting players from teams with higher winning percentages could be a strategic advantage.

However, the relationship between player performance and strength of schedule (SOS) was more nuanced. While there was a degree of correlation between SOS and fantasy points, the predictive power of SOS alone was limited. This indicated that the difficulty of opponents, as represented by SOS, is not a definitive predictor of player performance. Fantasy managers should consider other variables, such as team success, when making drafting decisions.

Insights from Linear Regression Analysis

We conducted linear regression analyses to quantify the relationships between fantasy points, team success, and SOS. The regression model for team standings exhibited a substantial R-squared value, indicating that team success (measured by winning percentage) is a significant predictor of player performance. Fantasy football managers should take note of this relationship, as players from successful teams are more likely to deliver higher fantasy point outputs.

In contrast, the regression model for SOS demonstrated a weaker R-squared value, implying that SOS alone has limited predictive power for fantasy points. This underscores the complexity of player performance, suggesting that factors beyond SOS contribute significantly to a player's fantasy point output.

Additionally, the heat map of historical player performance provides a roadmap for which direction related to player position when drafting in a snake draft. There were 35 QB's whom had more than 300 points since 2019 (per year), there were 3 RB's whom had more than 300 points since 2019 (per year), there were 14 WR's who had more than 200 points since 2019 (per year), and only 6 TE's whom had more than 150 points (per year). These facts point out that the shortage of a good RB would create a higher demand and should be drafted earlier than all other positions assuming it is one of the top 3. Next import would be one of the top 5 QB's whom were over 400 points (per year), or the top WR whom was just shy of 300 point (per year). Once these players are gone, the distinction of position to draft next becomes a little less clear, as there are only a handful of second tier RB's to take with just two hovering around the 300 mark.

```r
library(lm.beta)
model_1 <- lm( ttl_FPTS ~ Last, data = all_sum_team)
summary(model_1)
```

```
##
## Call:
## lm(formula = ttl_FPTS ~ Last, data = all_sum_team)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -434.37 -115.28   -5.48  124.81  400.25
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1139.176     30.949  36.809  < 2e-16 ***
## Last          -4.922      1.637  -3.007  0.00319 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 171 on 126 degrees of freedom
## Multiple R-squared:  0.06696,    Adjusted R-squared:  0.05955
## F-statistic: 9.042 on 1 and 126 DF,  p-value: 0.003185
```

```r
standardized_betas_model_1 <- lm.beta(model_1)
#all_sum_team$std_beta_sos_rank <- standardized_betas_model_1[2]
print(standardized_betas_model_1)
```

```
##
## Call:
## lm(formula = ttl_FPTS ~ Last, data = all_sum_team)
##
## Standardized Coefficients::
## (Intercept)        Last
##          NA  -0.2587657
```

```r
#print(all_sum_team)

conf_intervals_model_1 <- confint(model_1)



print(conf_intervals_model_1)
```

```
##                   2.5 %      97.5 %
## (Intercept) 1077.929782 1200.422839
```

```
## Last          -8.161292   -1.682807
```

The R-squared values are low, but suggest that the model can explain only a small portion of variance. While there is a relationship between the SOS and Fantasy point, it is probably not a good indicator of future fantasy points.

```r
all_22_team <- all_players %>%
  filter (Fantasy_yr == "2022", !is.na(Season_team))%>%
  group_by(Season_team) %>%
  summarise(total_FPTS = sum(FPTS))%>%
  left_join(team_stand_22, by = c("Season_team" = "Team"))



  print(all_22_team)
```

```
## # A tibble: 32 x 8
##    Season_team total_FPTS     W     L     T   Pct    PF    PA
##    <chr>            <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 ARI               738.     4    13     0 0.235   340   449
##  2 ATL               937.     7    10     0 0.412   365   386
##  3 BAL              1000.    10     7     0 0.588   350   315
##  4 BUF              1266     13     3     0 0.813   455   286
##  5 CAR               715      7    10     0 0.412   347   374
##  6 CHI               979.     3    14     0 0.176   326   463
##  7 CIN              1226.    12     4     0 0.75    418   322
##  8 CLE              1060.     7    10     0 0.412   361   381
##  9 DAL              1173.    12     5     0 0.706   467   342
## 10 DEN              1015.     5    12     0 0.294   287   359
## # i 22 more rows
```

```r
library(lm.beta)
model_2 <- lm( total_FPTS ~ Pct, data = all_22_team)
summary(model_2)
```

```
##
## Call:
## lm(formula = total_FPTS ~ Pct, data = all_22_team)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -283.86  -81.86   15.23   88.00  193.37
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   743.87      63.68  11.681 1.09e-12 ***
## Pct           618.92     119.53   5.178 1.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 122.4 on 30 degrees of freedom
## Multiple R-squared:  0.4719, Adjusted R-squared:  0.4543
## F-statistic: 26.81 on 1 and 30 DF,  p-value: 1.409e-05
```

```r
standardized_betas_model_2 <- lm.beta(model_2)
#all_sum_team$std_beta_sos_rank <- standardized_betas_model_1[2]
```

```r
print(standardized_betas_model_2)
```

```
##
## Call:
## lm(formula = total_FPTS ~ Pct, data = all_22_team)
##
## Standardized Coefficients::
## (Intercept)         Pct
##          NA   0.6869811
```

```r
#print(all_sum_team)

conf_intervals_model_2 <- confint(model_2)



print(conf_intervals_model_2)
```

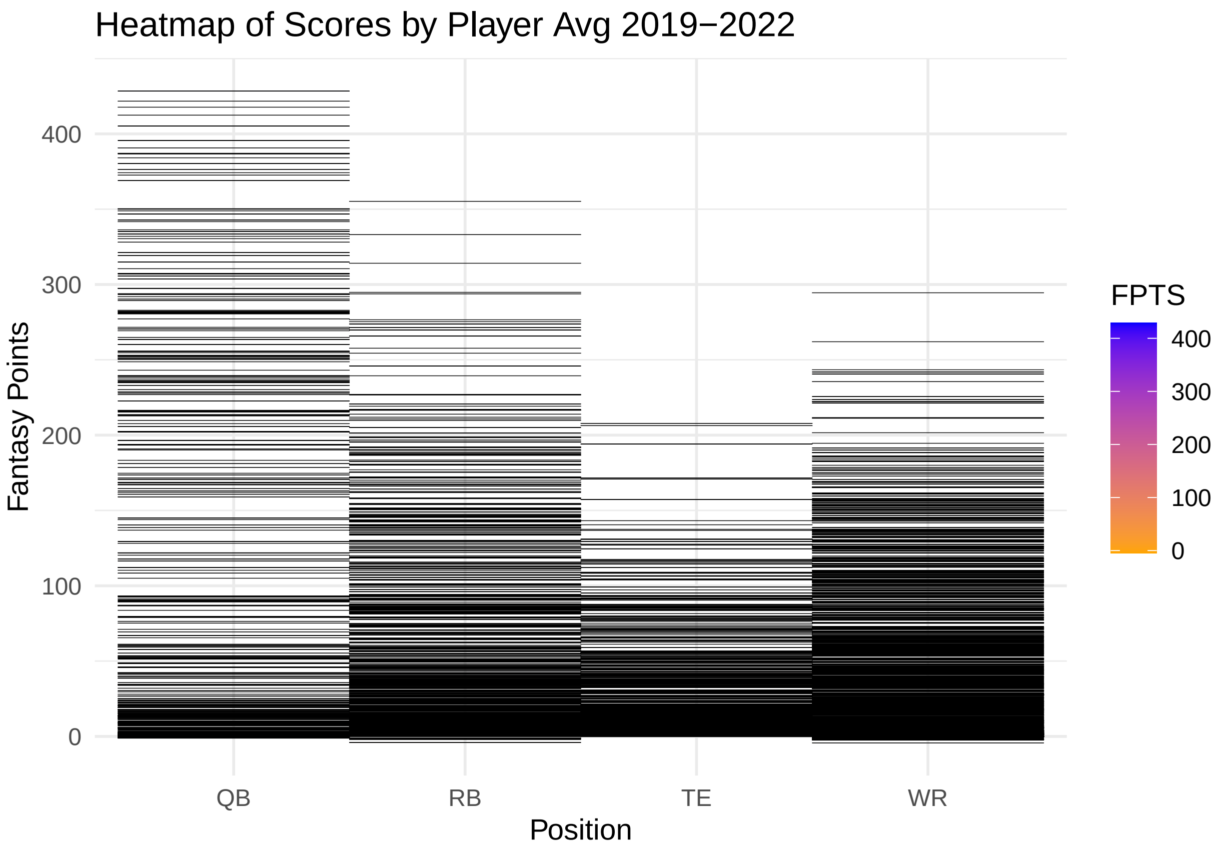
```
##                2.5 %   97.5 %
## (Intercept) 613.8085 873.9265
## Pct         374.8098 863.0250
```

Overall, the winning percentage variable indicates that the "Pct" variable has a significant impact on predicting the "total_FPTS," and the model explains a substantial portion of the variability in the dependent variable. This variable will be good to help predict the more valuable players. If your team has a higher winning percentage the player will be more valuable as they will score more points.

4. Plots and Visualizations

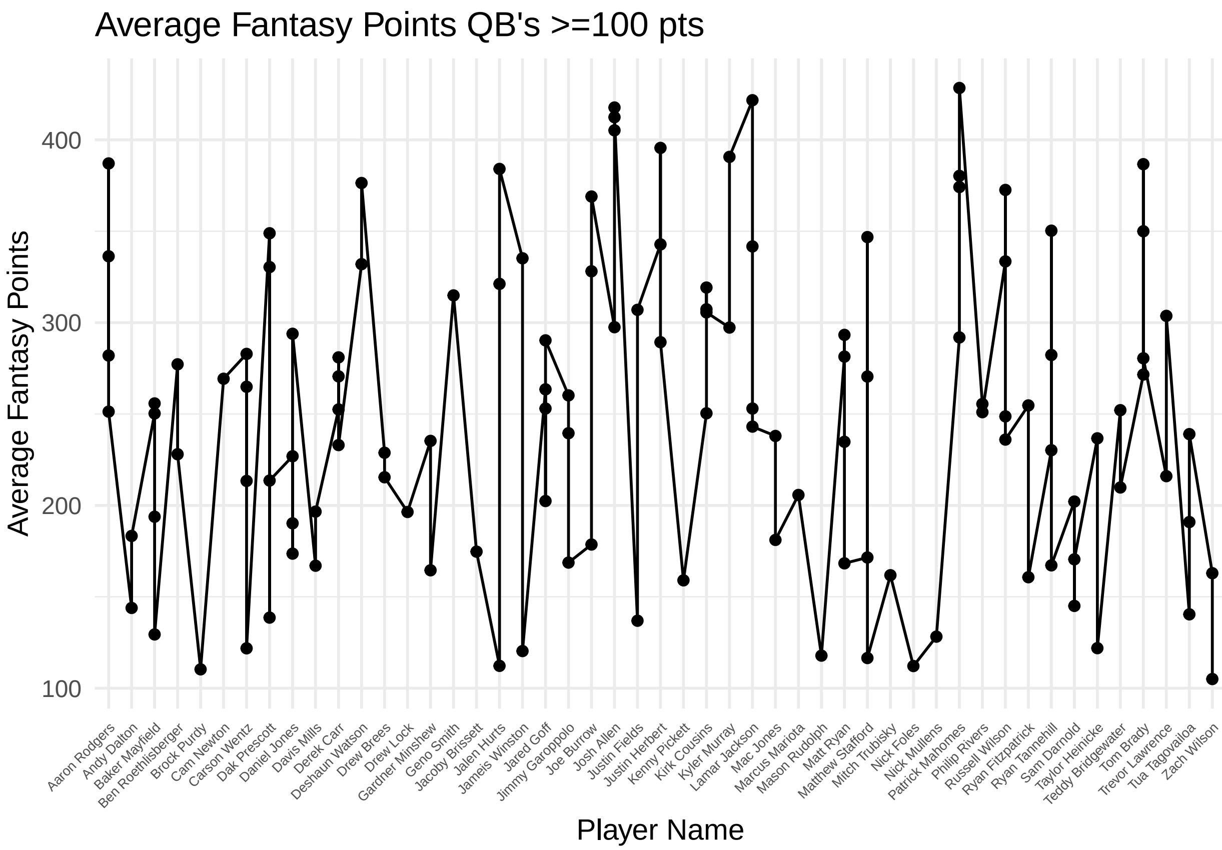
Plots and tables needed

1. Line plots to visualize trends in player performance over time.
2. Tables to display the top-performing players based on historical fantasy points.
3. Heat map to help understand which positions are more important to draft due to outliers in scoring by position.



The heat map shows some very important features when it comes to drafting analysis. There are a significant number of Qb's with more than 300 points, and only a small handful of Rb's that reach this milestone.

```
QB_filtered_players <-
  subset(QB_summary_data, Season_team != "NO TEAM FA" &
           FPTS >= 100)
RB_filtered_players <- subset(RB_summary_data, Season_team != "NO TEAM FA")
WR_filtered_players <- subset(WR_summary_data, Season_team != "NO TEAM FA")
TE_filtered_players <- subset(TE_summary_data, Season_team != "NO TEAM FA")
```



5. Implications

The analysis of fantasy football data, provided several intriguing insights on the intricate dynamics of player performance and their relationship with team attributes:

Team Success and Player Performance: A compelling connection was unearthed between team success and player performance. Players belonging to teams with higher winning percentages tended to produce higher fantasy points. This revelation challenges the notion that player performance exists in isolation and underscores the profound influence of team dynamics on individual output. Limited Predictive Power of SOS: While the strength of schedule (SOS) metric demonstrated a correlation with player fantasy points, its predictive power proved to be limited. The notion that a harder or easier schedule directly translates to better or worse player performance was debunked. This insight prompts a reconsideration of the common belief in SOS as a definitive predictor. Positional Differences: The heat map analysis illustrated the stark contrast in fantasy points distribution across positions. Quarterbacks boasted a wider range of fantasy points, with numerous players crossing the 300-point threshold. Conversely, running backs exhibited fewer high-scoring players, suggesting a position-specific variability in performance. Draft Strategy Implications: The revelation of team dynamics' impact on player performance introduces a strategic edge in drafting. Managers who prioritize players from successful teams might enhance their fantasy point accumulation. This insight empowers managers to strategize beyond individual player statistics and delve into team statistics for a more holistic drafting approach. Foundation for Future Predictive Models: While I didn't construct an extensive predictive model in this phase, the relationships uncovered lay the groundwork for future modeling endeavors. The insights guide the selection of features and the creation of models that can better predict fantasy football draft rankings. Complexity of Player Performance: The analysis highlighted the multifaceted nature of player performance prediction. Player statistics alone cannot encapsulate the intricate web of factors affecting fantasy output. Team dynamics, SOS, and other external variables collectively shape player performance. These insights collectively redefine the narrative of fantasy football drafting, emphasizing the significance of team success, the nuanced interplay between variables, and the need for predictive models that capture these complexities. As enthusiasts navigate the realm of fantasy football, armed with this new found understanding, they're poised to make more informed decisions, thereby elevating their chances of championship glory.

6. Concluding remarks

While the analysis provides valuable insights, it's essential to recognize its limitations and opportunities for improvement:

Limited Sample Size: The analysis spanned a few years of historical data (2019-2022). This short period of analysis might not encompass all possible scenarios and trends. Expanding the data set to include more seasons could provide a broader perspective on player performance. Model Development: While the analysis explored relationships between variables, I didn't construct a predictive model. Developing and fine-tuning a predictive model or models could provide more insights for fantasy football managers. Advanced statistics: My analysis mainly focused on existing features an did not incorporating any additional features, such as player performance ratios or advanced statistics. External Factors: I didn't delve into external factors that could influence player performance, such as injuries, player trades, coaching changes or weather. Model Selection: While Random Forest Regression is a strong contender, other advanced models like Gradient Boosting, Neural Networks, or time series models could be explored. A model comparison study could help identify the most suitable model for this specific problem. Real-time Updates: The analysis used historical data, limiting its real-time relevance. Creating a system that updates predictions during the NFL season based on current player performance and team dynamics could provide actionable insights for fantasy managers.

In summary, while the analysis has provided several key insights, its potential can be unlocked by addressing these limitations. Future iterations could integrate advanced modeling techniques, more comprehensive datasets, and a multidimensional approach that considers player statistics, and external factors. By addressing these challenges, the analysis could evolve from an exploratory exercise into a powerful tool guiding fantasy football managers towards successful drafting strategies.

7. Data collection and cleaning

Datsets

Strength of schedule (as of 9/1 each year):

https://www.teamrankings.com/nfl/ranking/schedule-strength-by-other?date=2022-09-01

Fantasy scoring points by two types of systems:

https://www.footballdb.com/fantasy-football/index.html

Team standings: https://www.nfl.com/standings/division/2022/REG

Player seasonal teams: https://fantasydata.com/nfl/fantasy-football-leaders

Required Packages

'tidyverse' 'ggplot2' 'dplyr' 'lm.beta'

Fantasy stats

The data had to have positions added to the data field, in addition to the fantasy year. Additionally, each position file had to be downloaded separately and combined into one one fantasy stats table per position for 2019-2022. Moreover, the fields also had to have the team for each player added. Since the fantasy stats reflected the players current team, these had to added to the file by inserting them into each of the roles.

Team standings

The data had to be put into a tibble format removing the division and conference. Additionally, the teams moving on to the playoffs had to have the signifying symbols removed from the team name. Moreover, the fields also were standardized by unbolding, removing the underlines, and hyperlinks. The team names had to be standardized to match the abbreviation used in the fantasy numbers files.

Team offensive stats

The offensive team stats had to have the fields standardized by unbolding, removing the underlines, and hyperlinks.

Strength of schedule (SOS)

The SOS data set had to have the date added to the file. Each file was downloaded separately and appended to each file. The file is as of 2/28 for the year that the season started (2019-2022). This allows for the final SOS to have been calculated. The team names had to be standardized to match the abbreviation used in the fantasy numbers files.

8. Appendix

Cleaning data

```r
# Load the required library
library(tidyverse)

# Step 1: Read the CSV file into a data frame
setwd("~/DSC520/Week 9/csv")
# Read all the CSV files and bind them row-wise
data1 <- read.csv("FantasyPros_Fantasy_Football_Statistics_QB-2_2021.csv")
data2 <- read.csv("FantasyPros_Fantasy_Football_Statistics_QB_2022.csv")
data3 <- read.csv("FantasyPros_Fantasy_Football_Statistics_QB-3_2020.csv")
data4 <- read.csv("FantasyPros_Fantasy_Football_Statistics_QB-4_2019.csv")

# Combine all data frames into one
combined_data <-
  rbind(data1,data2,data3,data4)

# Write the merged data frame to a new CSV file
write.csv(combined_data, file = "QB_Merged_Fantasy_Football_Statistics.csv"
          , row.names = FALSE)
```

```r
# Load the required library
library(tidyverse)

# Step 1: Read the CSV file into a data frame
setwd("~/DSC520/Week 9/csv")
# Read all the CSV files and bind them row-wise

data5 <- read.csv("FantasyPros_Fantasy_Football_Statistics_RB-2_2021.csv")
data6 <- read.csv("FantasyPros_Fantasy_Football_Statistics_RB-3_2020.csv")
data7 <- read.csv("FantasyPros_Fantasy_Football_Statistics_RB-4_2019.csv")
data8 <- read.csv("FantasyPros_Fantasy_Football_Statistics_RB_2022.csv")
```

```r
# Combine all data frames into one
combined_data2 <-
  rbind(data5,data6,data7,data8)

# Write the merged data frame to a new CSV file
write.csv(combined_data2, file = "RB_Merged_Fantasy_Football_Statistics.csv"
          , row.names = FALSE)
```

```r
# Load the required library
library(tidyverse)

# Step 1: Read the CSV file into a data frame
setwd("~/DSC520/Week 9/csv")
# Read all the CSV files and bind them row-wise

data9 <- read.csv("FantasyPros_Fantasy_Football_Statistics_WR-2_2021.csv")
data10 <- read.csv("FantasyPros_Fantasy_Football_Statistics_WR-3_2020.csv")
data11 <- read.csv("FantasyPros_Fantasy_Football_Statistics_WR_2022.csv")
data12 <- read.csv("FantasyPros_Fantasy_Football_Statistics_WR-4_2019.csv")

# Combine all data frames into one
combined_data3 <-
  rbind(data9,data10,data11,data12)

# Write the merged data frame to a new CSV file
write.csv(combined_data3, file = "WR_Merged_Fantasy_Football_Statistics.csv"
          , row.names = FALSE)
```

```r
# Load the required library
library(tidyverse)

# Step 1: Read the CSV file into a data frame
setwd("~/DSC520/Week 9/csv")
# Read all the CSV files and bind them row-wise

data13 <- read.csv("FantasyPros_Fantasy_Football_Statistics_TE-2_2021.csv")
data14 <- read.csv("FantasyPros_Fantasy_Football_Statistics_TE-4_2019.csv")
data15 <- read.csv("FantasyPros_Fantasy_Football_Statistics_TE_2022.csv")
data16 <- read.csv("FantasyPros_Fantasy_Football_Statistics_TE-3_2020.csv")

# Combine all data frames into one
combined_data4 <-
  rbind(data13,data14,data15,data16)

# Write the merged data frame to a new CSV file
write.csv(combined_data4, file = "TE_Merged_Fantasy_Football_Statistics.csv"
          , row.names = FALSE)
```

9. References

2023b. https://fantasydata.com/nfl/fantasy-football-leaders.

———. 2023d. https://www.nfl.com/standings/division/2022/REG.

———. 2023a. https://www.footballdb.com/fantasy-football/index.html.

———. 2023c. https://www.teamrankings.com/nfl/ranking/schedule-strength-by-other?date=2020-02-28.