Fantasy Football_project

Michael Robinson

2023-12-13

Introduction

In this project I combine player statistics with sentiment analysis to assist with decision-making in fantasy football. The project data-driven methodology to predict and study player performance, using sentiment scores to gain insights into player capabilities and public opinion. using data cleansing, performance analysis by position, and correlation study, using Random Forest algorithms to forecast player outcomes. This project makes a contribution by looking at how people's opinions about players relate to their fantasy football scores. This adds a new layer to how we think about player worth. It provides predictions and visual data that fans can understand and use.

Read the Data from a CSV File

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(shiny)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
library(purrr)
```

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:scales':
##
##     discard

## The following object is masked from 'package:caret':
##
##     lift
```

```r
data <- read_csv("~/OneDrive - CUNY/yearly_data_updated_08_23.csv", show_col_types = FALSE)
sentiment_data <- read.csv("~/OneDrive - CUNY/data607/sentiment.csv")
```

Data Cleaning and Preparation

Cleaning the dataset by selecting relevant columns and dividing the data into previous and current season datasets for analysis.

```
data_cleaned <- data %>%
  select(id, name, position, team, season, passing_yards, passing_tds, rushing_yards,
         rushing_tds, receptions, receiving_yards, receiving_tds, fantasy_points_ppr)

latest_season <- max(data_cleaned$season)

prev_seasons <- data_cleaned %>% filter(season < latest_season)
current_data <- data_cleaned %>% filter(season == latest_season)
```

## Top Performer by position

Visualizes the top performers in key positions such as Quarter Backs, Running Backs, and Wide Receivers.

```
Quarter_Backs <- data_cleaned %>% top_n(5, passing_yards)
Running_Backs <- data_cleaned %>% top_n(5, rushing_yards)
Wide_Receiver <- data_cleaned %>% top_n(5, receiving_yards)

# Top Quarter Backs
ggplot(Quarter_Backs, aes(x = reorder(name, passing_yards), y = passing_yards, fill = name)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 5 Passers", x = "Player", y = "Passing Yards") +
  theme_minimal() +
  scale_fill_viridis_d()
```
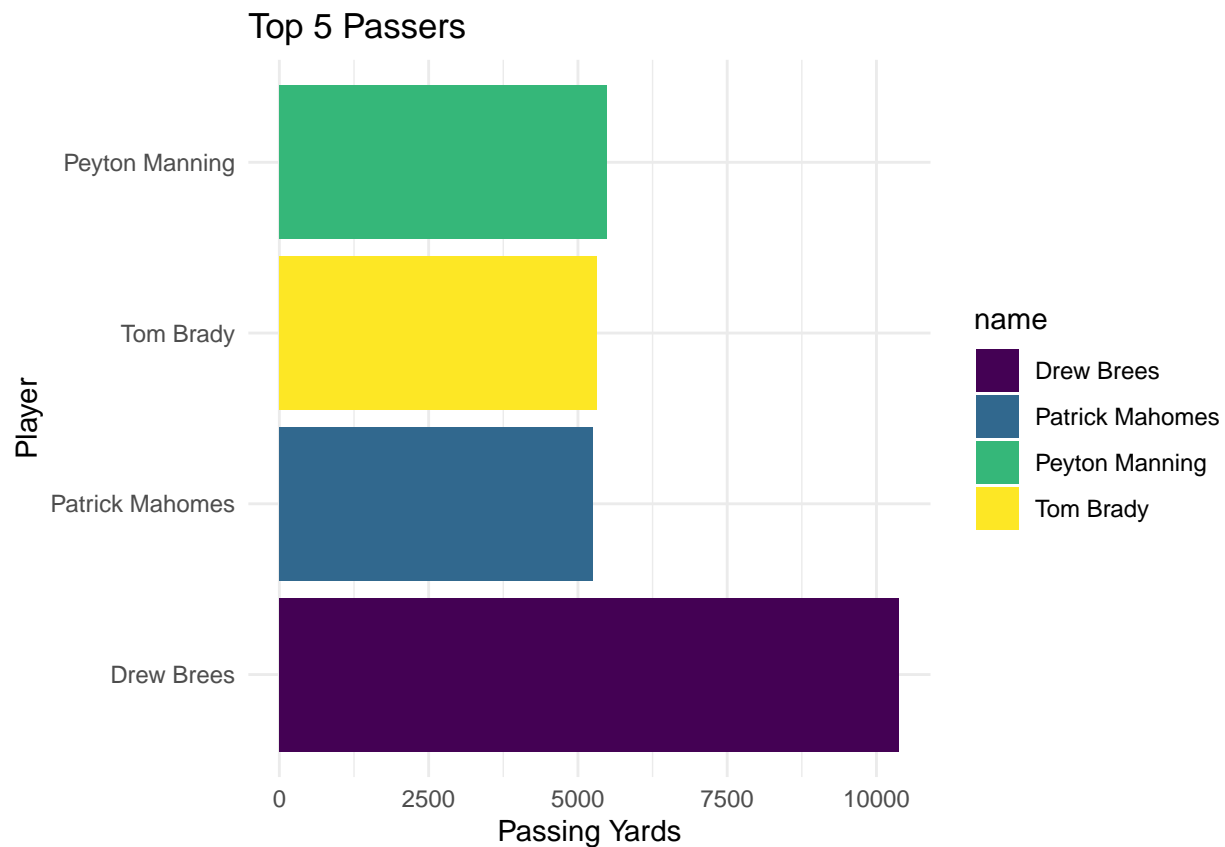
```r
#  Top Running Backs
ggplot(Running_Backs, aes(x = reorder(name, rushing_yards), y = rushing_yards, fill = name)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 5 Rushers", x = "Player", y = "Rushing Yards") +
  theme_minimal() +
  scale_fill_viridis_d()
```


Top 5 Rushers
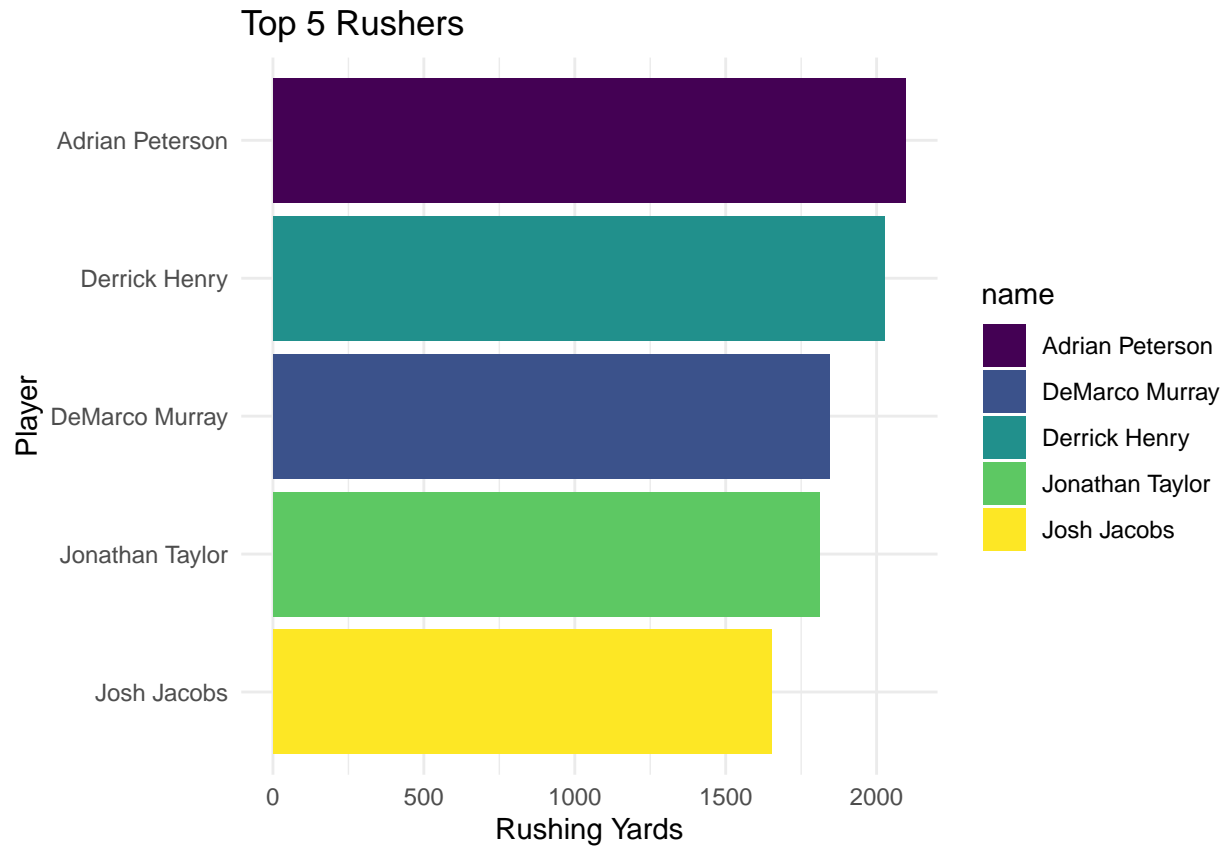
```r
# Top Wide Receivers
ggplot(Wide_Receiver, aes(x = reorder(name, receiving_yards), y = receiving_yards, fill = name)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 5 Receivers", x = "Player", y = "Receiving Yards") +
  theme_minimal() +
  scale_fill_viridis_d()
```
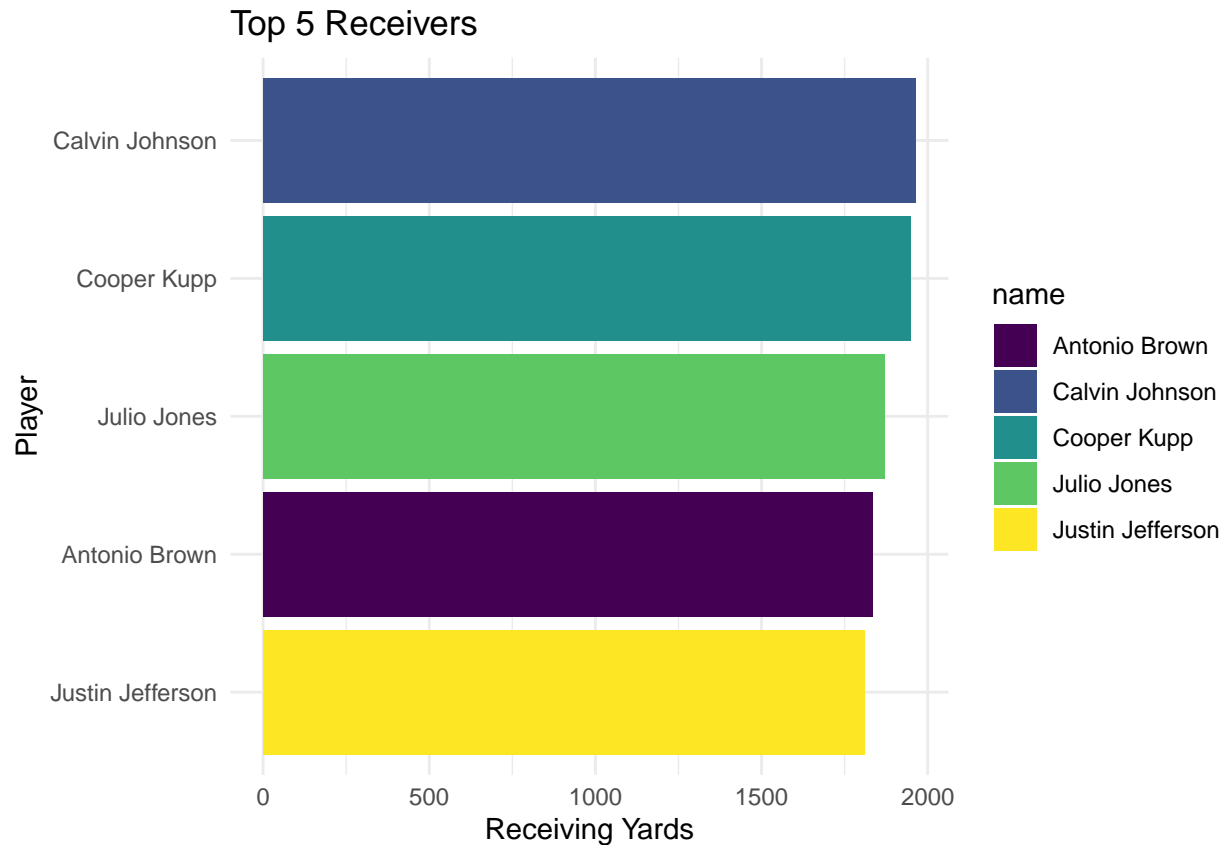
Average Points by position

Calculates and visualizes the average fantasy points scored in each position, to providing insights into performance trends of different player roles.

```
avg_points_per_position <- data_cleaned %>%
  group_by(position) %>%
  summarise(average_fantasy_points = mean(fantasy_points_ppr, na.rm = TRUE))


ggplot(avg_points_per_position, aes(x = position, y = average_fantasy_points, fill = position)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Average Fantasy Points PPR by Position", x = "Position", y = "Average Fantasy Points")
```

Average Fantasy Points PPR by Position
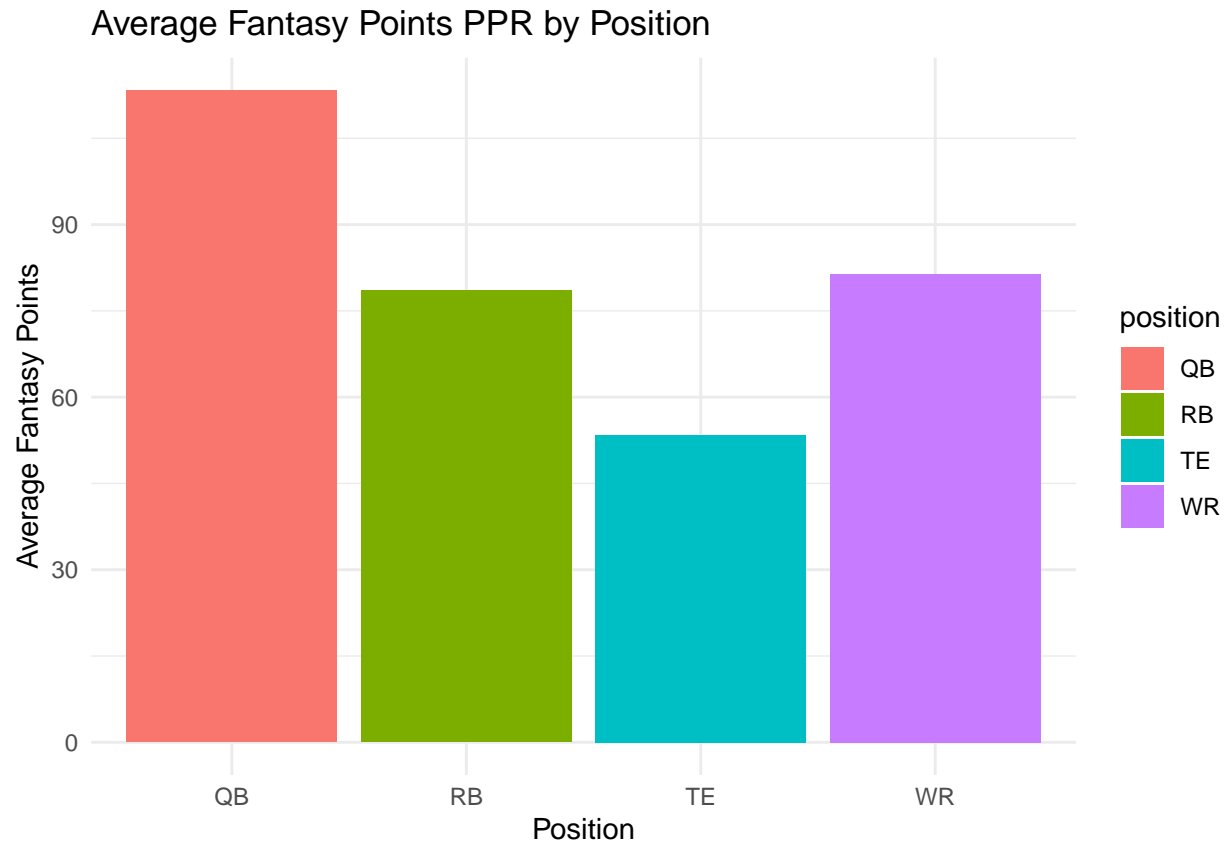
Correlation Analysis

Examines the relationships between various statistical measures in the data, to understand the interdependencies and patterns.

```
correlation_matrix <- cor(data_cleaned %>% select(-id, -name, -team, -position))

corrplot(correlation_matrix, method = "circle", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")
```
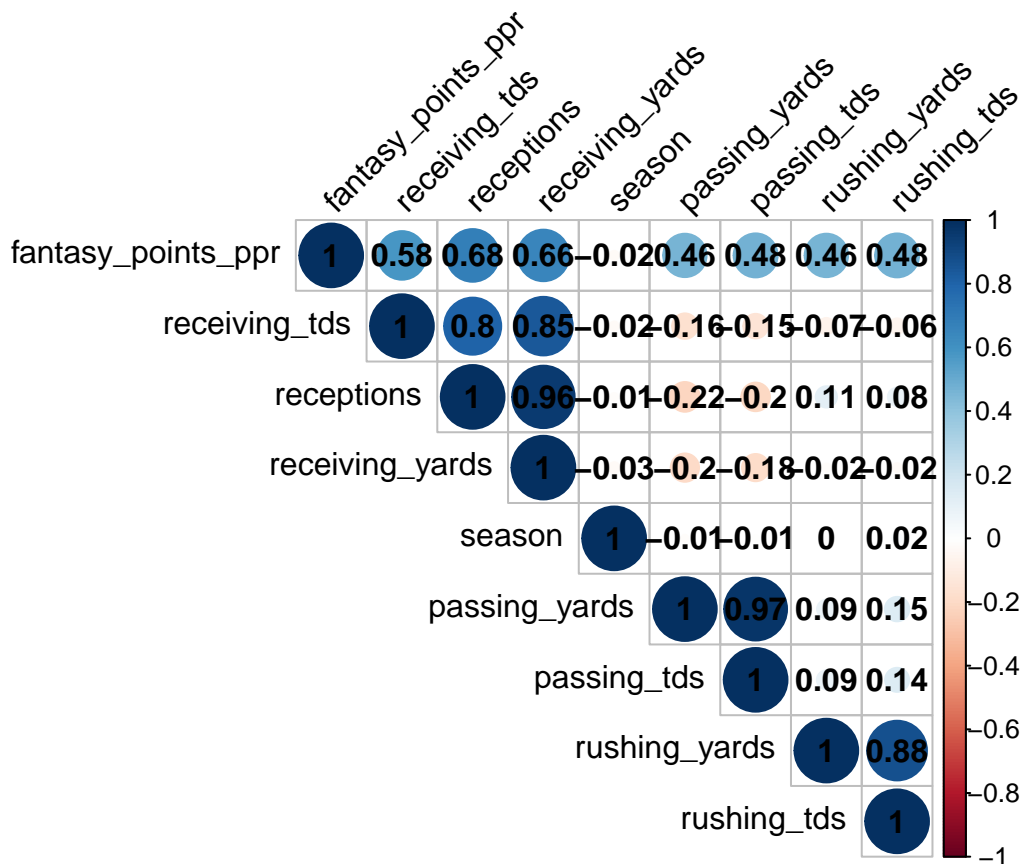
Model Training

Teaching a Random Forest model using past data, to be used making future predictions

```
model <- randomForest(fantasy_points_ppr ~ . -id -name -season, data = prev_seasons)
```

Selecting Random Players for Prediction

Randomly selects players from the current season's dataset to use for predictive modeling

```
set.seed(123)
sample_size <- 10
sampled_players <- current_data %>% sample_n(sample_size)

predict_data <- sampled_players %>% select(-fantasy_points_ppr)
```

Making and displaying Predictions

using the trained model to forecast fantasy points for the chosen players and display predictions.

```
predicted_points <- predict(model, predict_data)
```

```
predicted_points_rounded <- round(predicted_points, 2)

sampled_players$predicted_fantasy_points_ppr <- predicted_points_rounded

print(sampled_players %>% select(name, season, fantasy_points_ppr, predicted_fantasy_points_ppr))
```

```
## # A tibble: 10 x 4
##     name             season fantasy_points_ppr predicted_fantasy_points_ppr
##     <chr>             <dbl>              <dbl>                        <dbl>
##  1 Mike Gesicki        2022               98.2                         96.1
##  2 Phillip Walker      2022               39.1                         37.3
##  3 Devin Singletary    2022              178.                         187.
##  4 Tanner Conner       2022                0                            0.5
##  5 Eno Benjamin        2022                0.1                          0.46
##  6 D'Andre Swift       2022              191.                         189.
##  7 Johnny Mundt        2022               39                           40.1
##  8 Ian Thomas          2022               40.7                         41.9
##  9 Jacob Harris        2022                1.6                          1.8
## 10 Amari Cooper        2022              246                          245.
```

Outputting the Predictions into a plot

A Visual representation of the predicted points, to aid in the comparison and interpretation of the model's output.

```
# Creating a bar plot with the predicted fantasy points
ggplot(sampled_players, aes(x = reorder(name, predicted_fantasy_points_ppr), y = predicted_fantasy_point
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_minimal() +
  labs(title = "Predicted Fantasy Points PPR for Selected Players", x = "Player", y = "Predicted Fantasy
  coord_flip() + # Flipping the coordinates for better readability of player names
  geom_text(aes(label = predicted_fantasy_points_ppr), hjust = -0.1) # Adding labels to the bars
```

Predicted Fantasy Points PPR for Selected Players
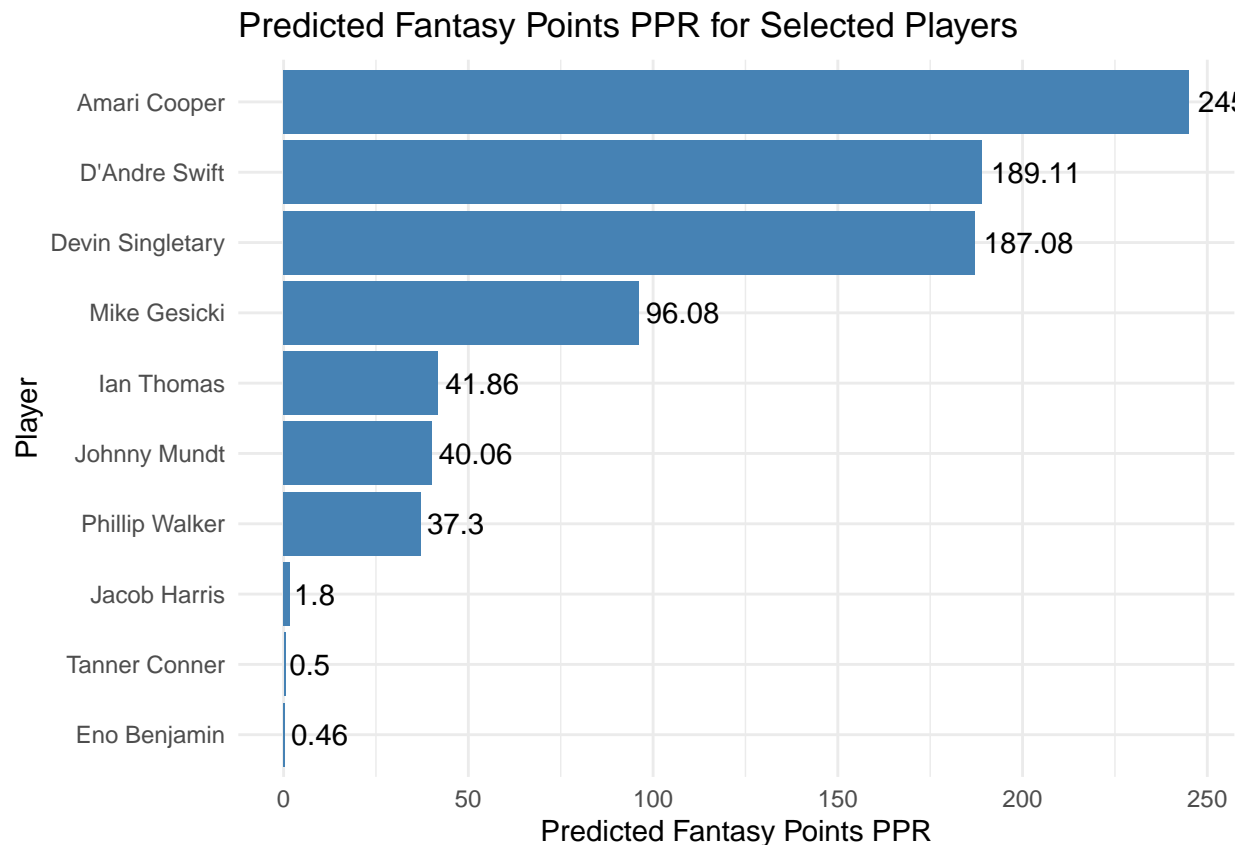
Fantasy Points vs Positive Sentiment

A visual to show the correlation between the sentiment score of players and the fantasy points

```
combined_data <- merge(data_cleaned, sentiment_data, by = "name")

# Assuming combined_data has columns for fantasy points and sentiment scores
# Calculate correlation between fantasy points and positive sentiment
correlation_pos = cor(combined_data$fantasy_points, combined_data$pos, use = "complete.obs")

# Calculate correlation between fantasy points and negative sentiment
correlation_neg = cor(combined_data$fantasy_points, combined_data$neg, use = "complete.obs")

# Scatter plot of fantasy points vs positive sentiment
ggplot(combined_data, aes(x = pos, y = fantasy_points_ppr)) +
    geom_point() +
    labs(title = "Fantasy Points vs Positive Sentiment",
         x = "Positive Sentiment Score",
         y = "Fantasy Points")
```

Fantasy Points vs Positive Sentiment
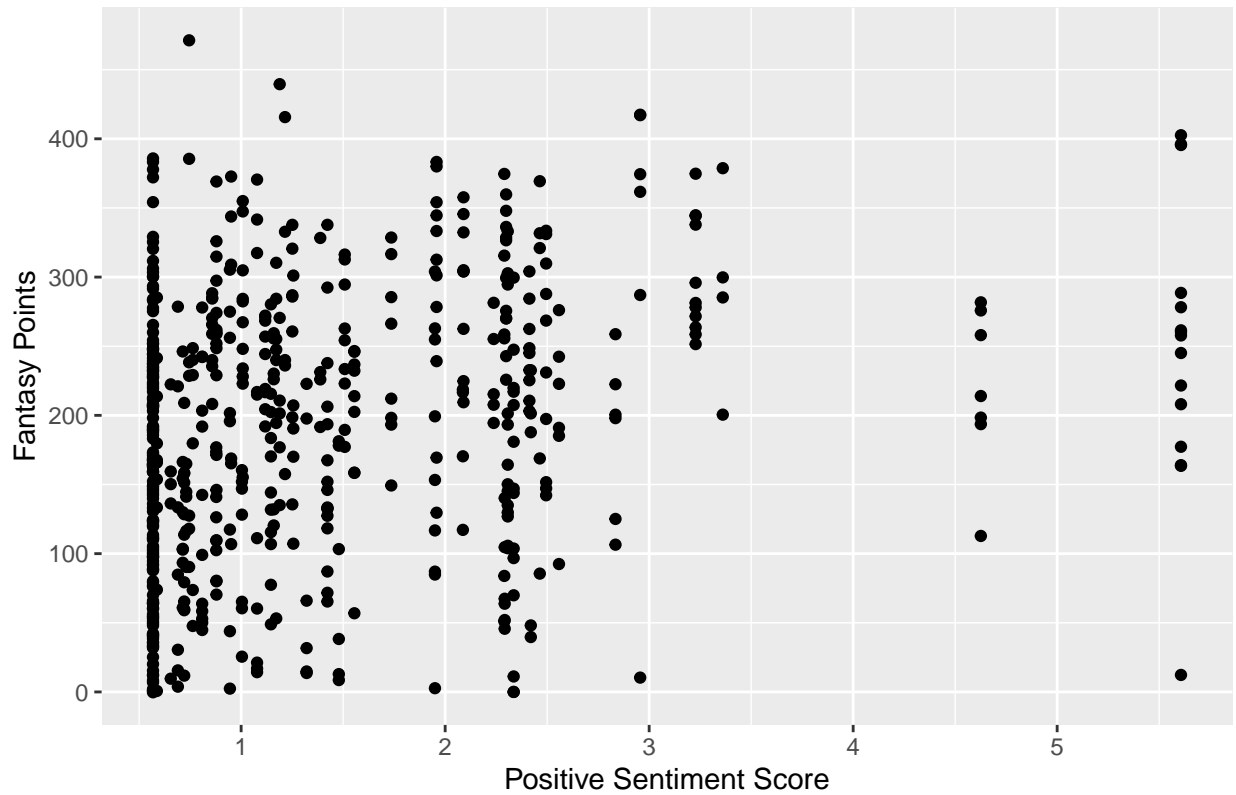
Sentiment Scores by Position

The visual displays the distribution of positive sentiment scores across four different positions in a sport, which are labeled as QB (Quarterback), RB (Running Back), TE (Tight End), and WR (Wide Receiver).

```r
ggplot(combined_data, aes(x = Pos, y = pos)) +
    geom_boxplot() +
    labs(title = "Positive Sentiment Scores by Position", x = "Position", y = "Positive Sentiment Score"
```

Positive Sentiment Scores by Position
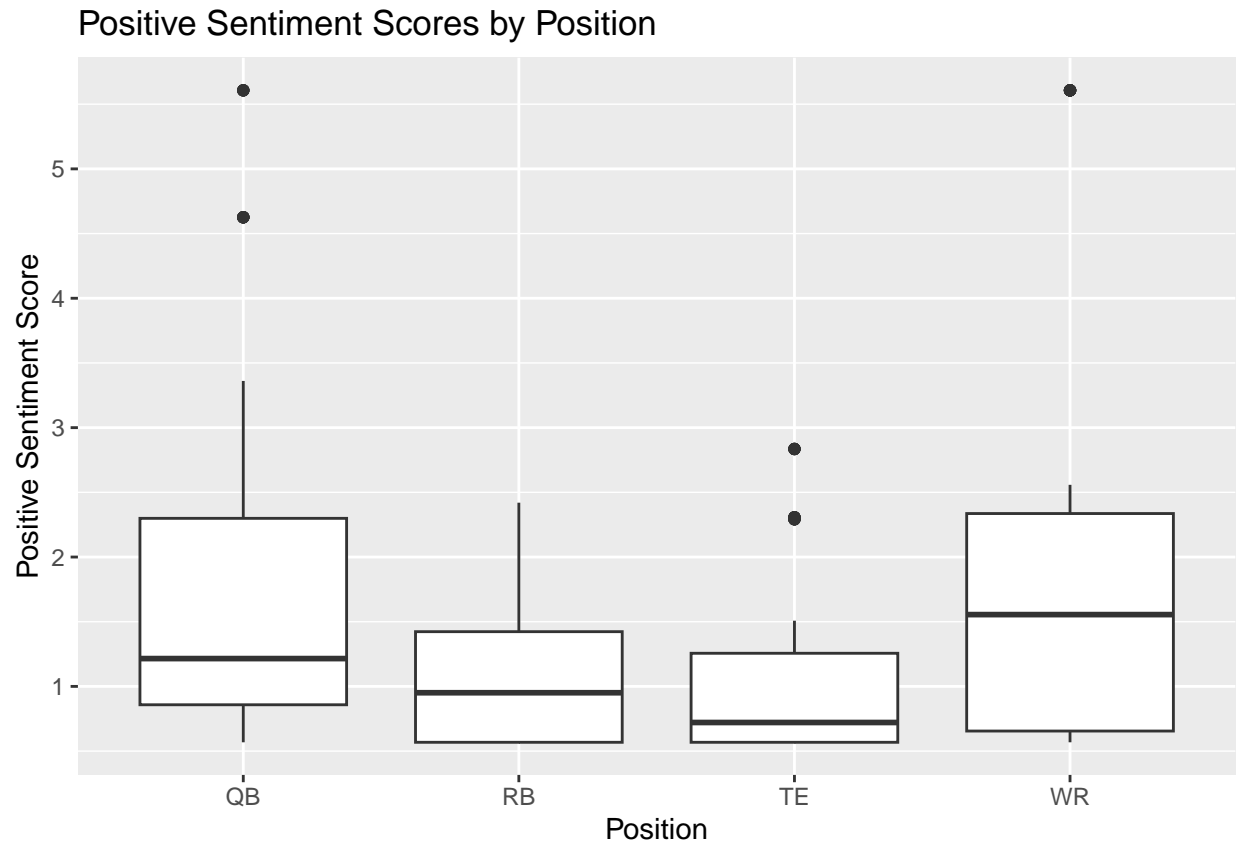
Sentiment Score over Time

The chart displays the average positive sentiment score for each season, with seasons

```r
# Plotting the average positive sentiment score for each season
ggplot(combined_data, aes(x = season, y = pos)) +
    geom_bar(stat = "summary", fun = "mean", fill = "blue") +
    labs(title = "Average Positive Sentiment Scores Over Seasons", x = "Season", y = "Average Positive S
```

Average Positive Sentiment Scores Over Seasons
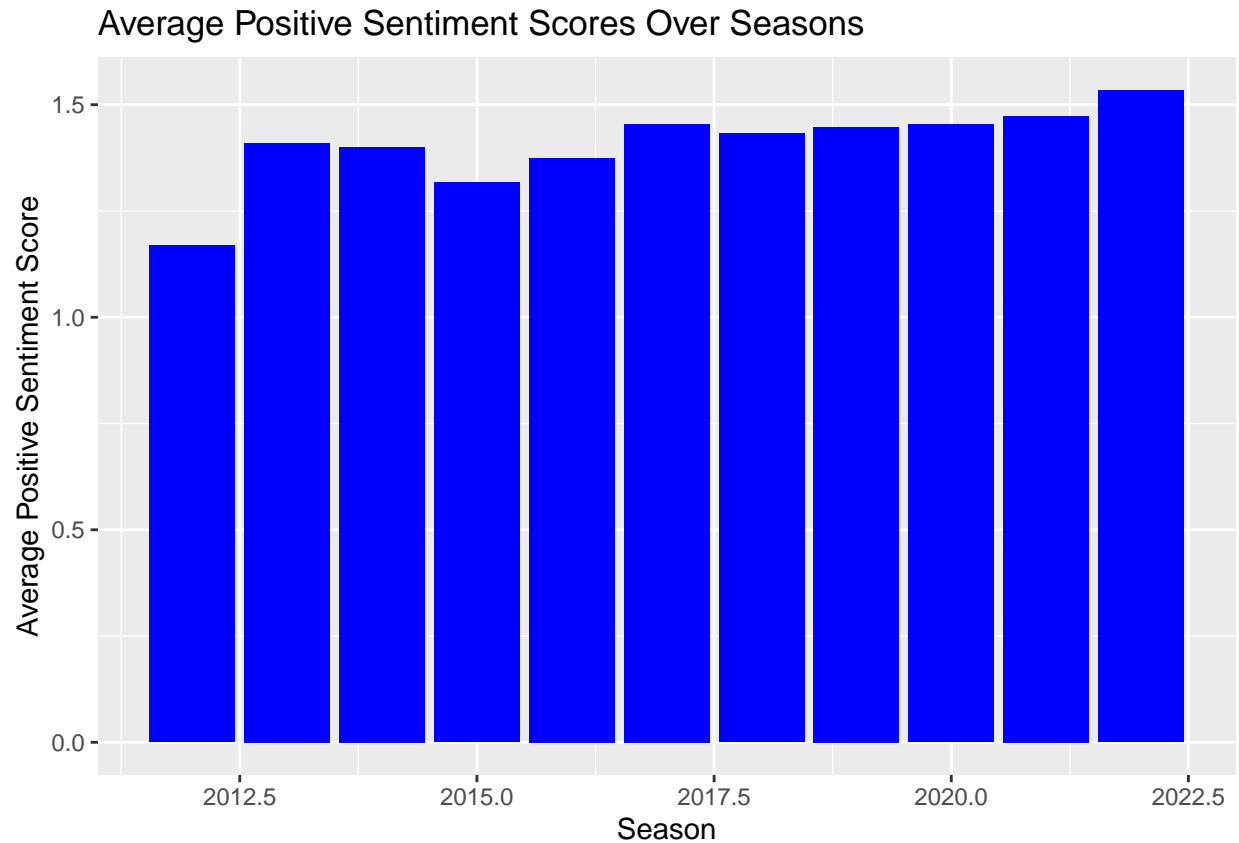
Identify Top Players

```r
combined_data$combined_score <- combined_data$pos + combined_data$fantasy_points_ppr

top_players <- combined_data[order(combined_data$combined_score, decreasing = TRUE), ][1:10, ]

ggplot(top_players, aes(x = reorder(name, combined_score))) +
    geom_bar(aes(y = pos), stat = "identity", position = position_dodge(), fill = "blue") +
    geom_line(aes(y = fantasy_points_ppr/10, group = 1), color = "red", size = 1) + # scaling down fant
    scale_y_continuous(sec.axis = sec_axis(~.*10, name = "Fantasy Points PPR")) +
    labs(title = "Top Players by Sentiment Scores and Fantasy Points",
        x = "Player",
        y = "Positive Sentiment Score") +
    coord_flip()
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

Top Players by Sentiment Scores and Fantasy Points
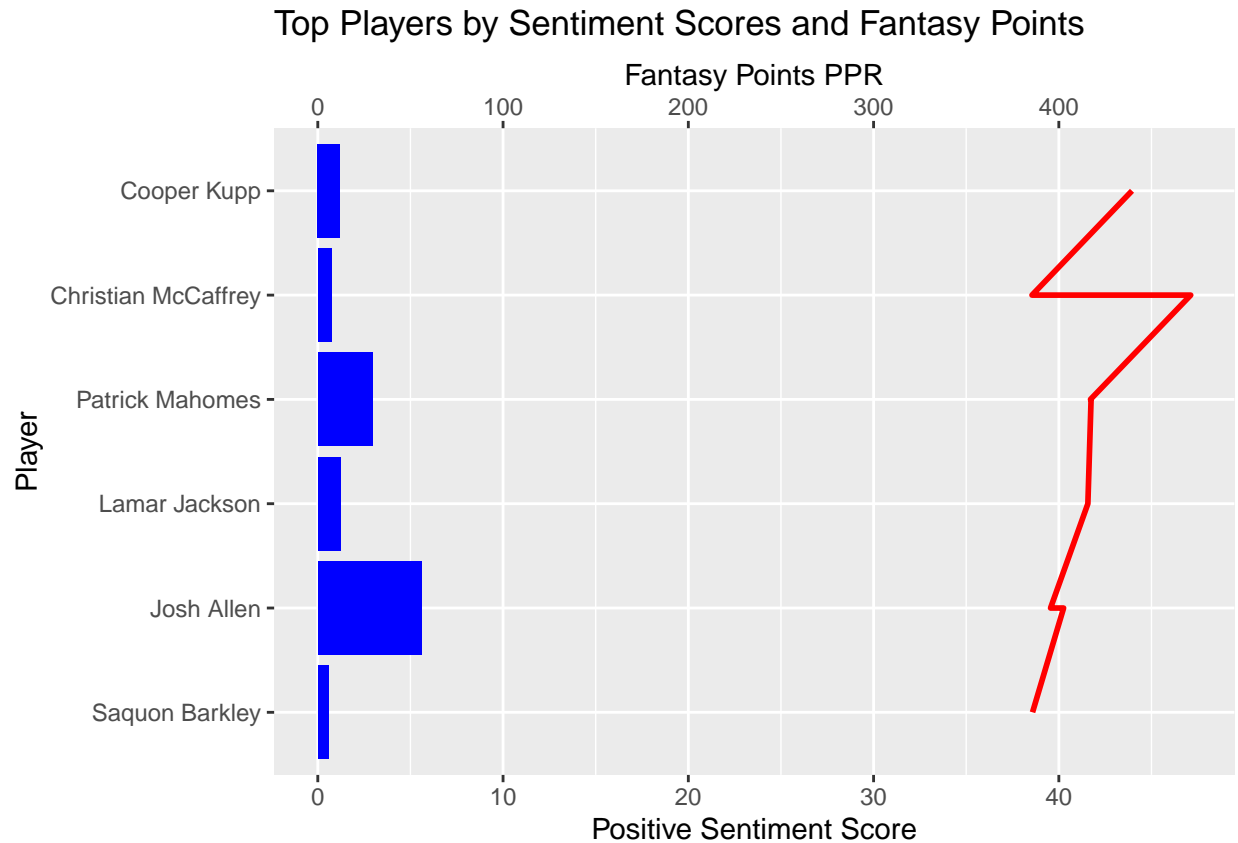
Conclusion

This project showcases the intersection of data analytics and fantasy football through player performance and public sentiment analysis. The project highlights the importance of statistical data and player sentiment to determining fantasy player value, offering a multifaceted perspective of the game. These insights are not only important for current strategy but will also shape future approaches in fantasy football, helping enthusiasts and analysts make informed decisions and gain a competitive advantage.