Exploring NFL Datasets

Bharath Goud Nadimpally

● Introduction

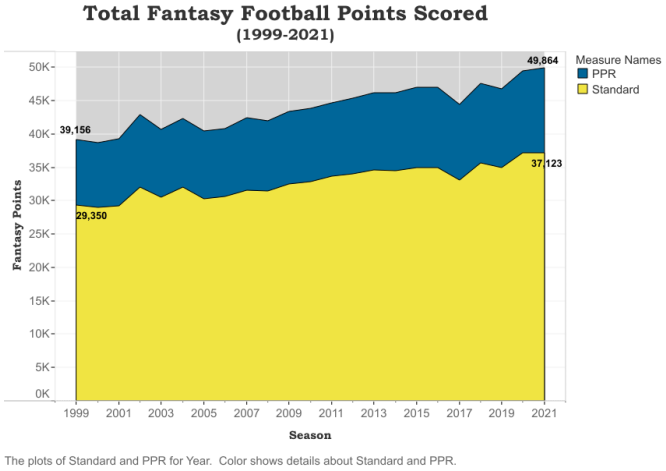
The 'NFLVerse' is a collection of data and R packages, which can be found online at https://nflverse.r-universe.dev, and is intended for NFL (National Football League) analytics. One such package, called 'nflreadr,' can be used to extract NFL data from 'GitHub' repositories, and contains 17 different vignettes of up to date data. For purposes of my exploration, I focus on the 'PBP' (Play by Play) and 'Player_Stats' datasets, which both contain columns containing ordinal, nominal, and many statistical numerical variables, and rows representing every player in every game. Separate datasets may be imported ranging from 1999 to 2022. Further details can be found at https://cran.r-project.org/web/packages/nflreadr/index.html.

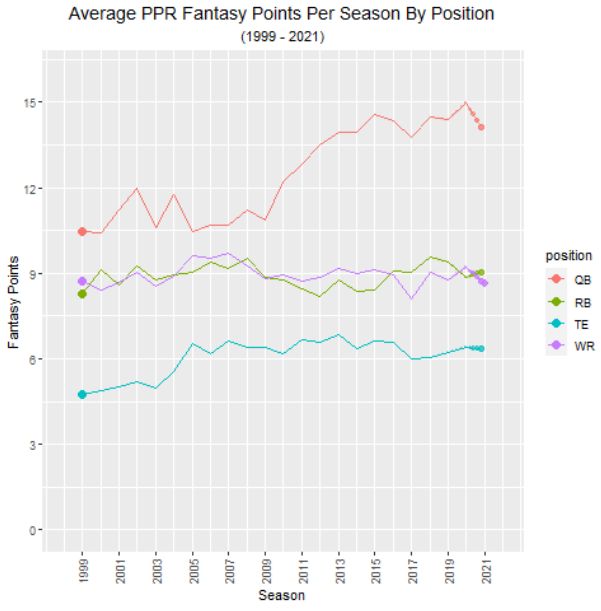
My exploration provides the average fan a visual look into popular subjects and trends regarding the NFL, both current and over the last 23 years. I begin with a subject of growing interest in the NFL community, fantasy sports, and quarterback trends of the 2022 season, a visual that could be useful for this current season. Next, I take a deep dive into the last full season, 2021, to provide a broad overview of the performance trends. Two visuals will allow the audience to see these trends: first showing the performance of each specific team, then the production in each week of the season.

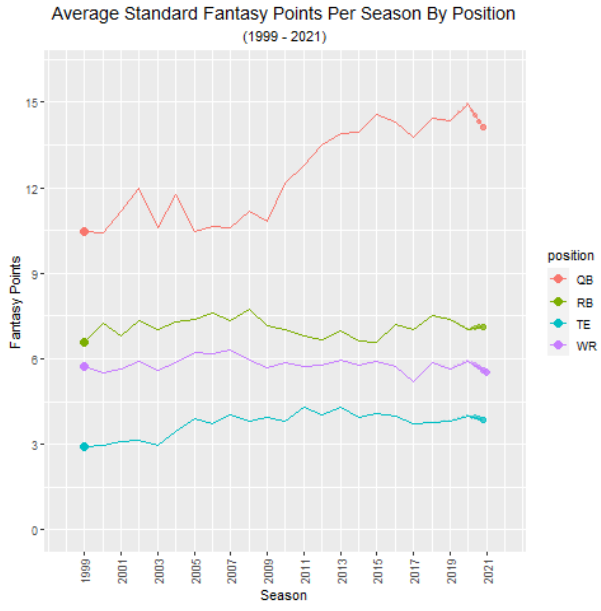
After that, I focused on return yards, which is measured by advancing the football towards the opposing team's goal, to see the pattern that is observed after the rule change that happened in 2011 for kicking (moving the kickoff line from the 30 yard line to the 35 yard line). Next, I looked at the positions for the player because to win a game every player's support is required, everyone has equal importance in a game. But I need to have an idea about the players position group in the game for in depth analysis about the players/game, to win a match. For this I did analysis for position group vs successful goals, to know which position group made more number of goals. According to the players strengths and weaknesses Coaches can change players position groups to win a match.

- **Exploratory Analysis**

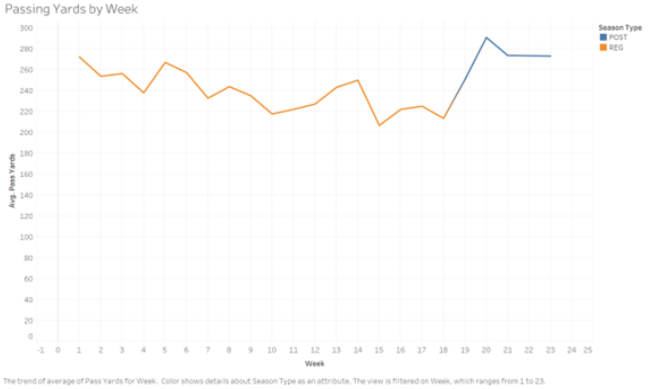
I made a handful of visualizations in order to obtain a more complete understanding of the relationships between some of the variables that were included in the dataset. The first direction, team quarterback performance in fantasy football, was decided through several exploratory graphs, beginning with a comparison of total fantasy points scored in PPR (Points Per Reception) formats and standard formats over the last 23 years. The most noticeable trend for both formats was the overall increase in total points each season, and a closer look was taken into the values of each position. Graphs of the average points scored by position each season, in both PPR and standard formats, shows a clear increase in quarterback production as opposed to the other three major offensive positions of wide receiver, running back, and tight end. I then chose to focus on the quarterback position performance of each team of the current season so it might be helpful for current fantasy football players.

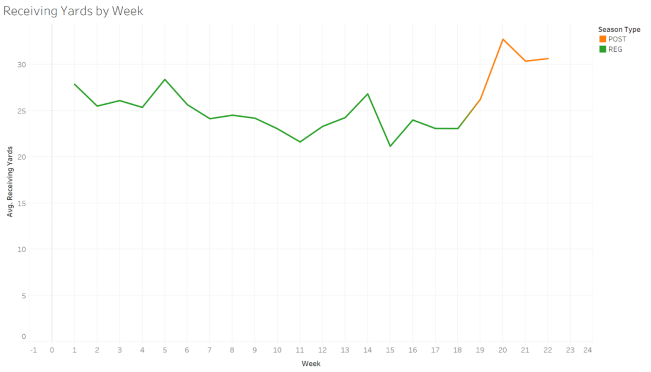


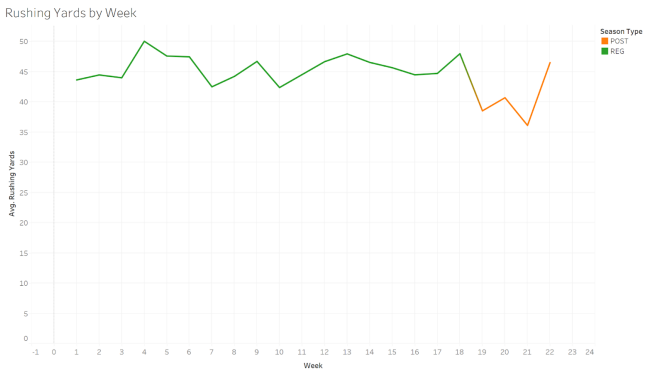


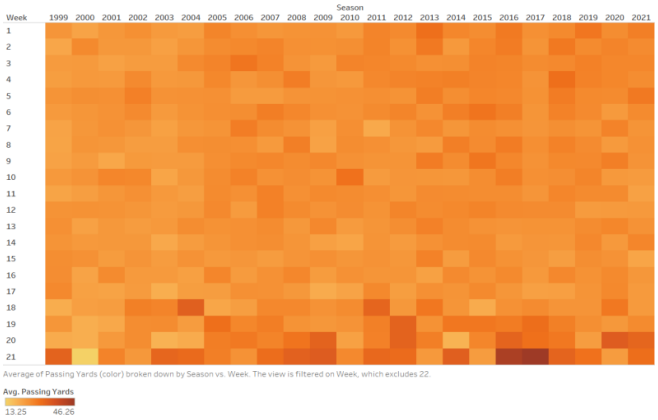


In my second direction of 2021 performance trends, ‘yards’ was explored in several different ways by comparing ‘passing yards,’ ‘return yards,’ ‘receiving yards,’ and ‘penalty yards’ with season and week for the teams that participated in the NFL to see how the trend looks like. This was done by visualizing line graphs, scatter plots and then using heat maps. As a result, i used this as one direction for my final project.

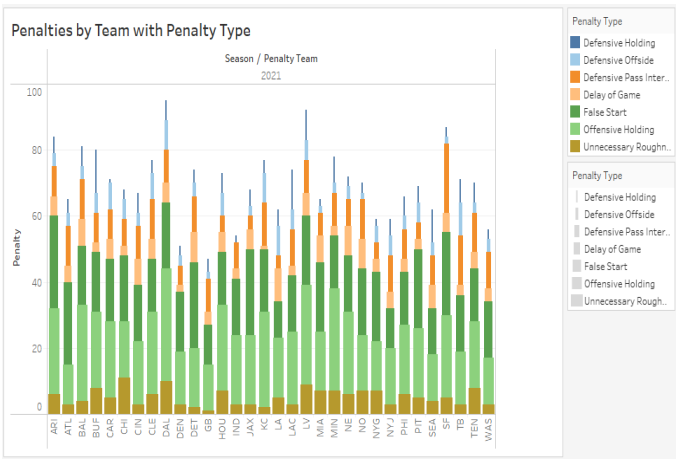


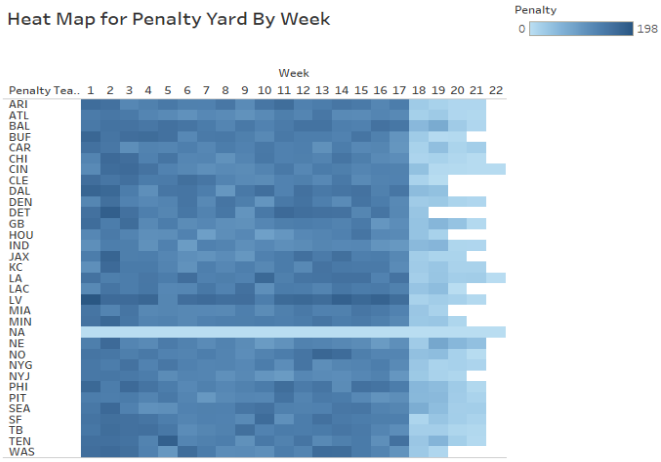




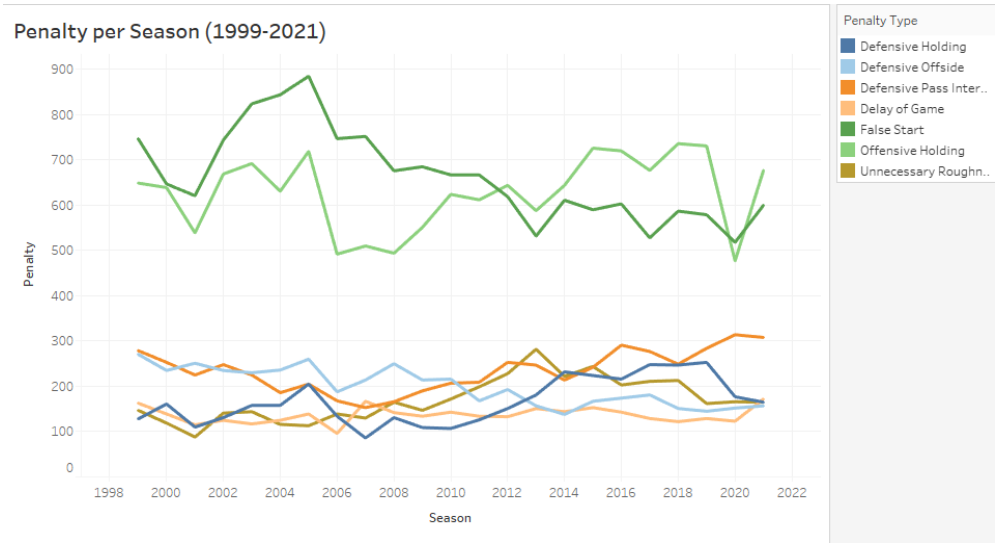


In my third direction of 2021 performance trends, the number of penalties was explored in different ways for the NFL team to see and compare how the trends look like. This was observed by first looking at the stacked bar graph to see the number of penalties sorted on the basis of penalty types that are seen higher within the teams. Not only this, a heat map was also observed that explains the relationship between penalty yards by week for each team. The graph shows how many penalties are being committed by a certain team from Week 1 through Week 22. By doing so I get to know how the performance of the team is changing as the week passes by, and clearly I can see that there is a change in the improvement when the teams played in the later week, which might also be true in general. Additionally, the divergent color used shows the two extreme values. Meaning the darker the color of each square is, the more number of penalties there will be, and the lighter the color is, lesser number of penalties there will be for a particular team in that week.



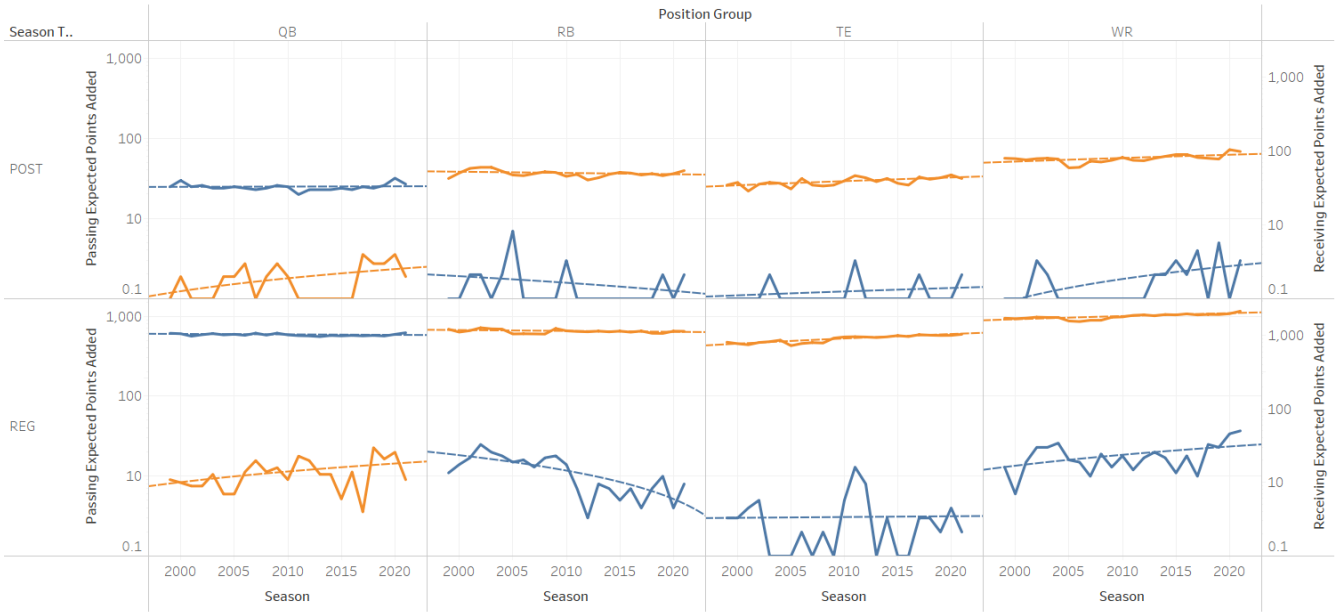


The second approach of the third direction was observed by looking at the line graph that focuses on the penalties per season from 1999 through 2021 sorted on the basis of the penalty type that are seen higher within the teams.



To overcome penalties, the team requires extra points to balance a game. In both season types (POST, REG) QB position is leading in Passing EPA whereas RB, TE, WR position is leading in receiving EPA. If the players in these positions have coordination then the team will score a lot of Extra points that may change a game.

EPA scored by Position Group in each season (1999-2021)

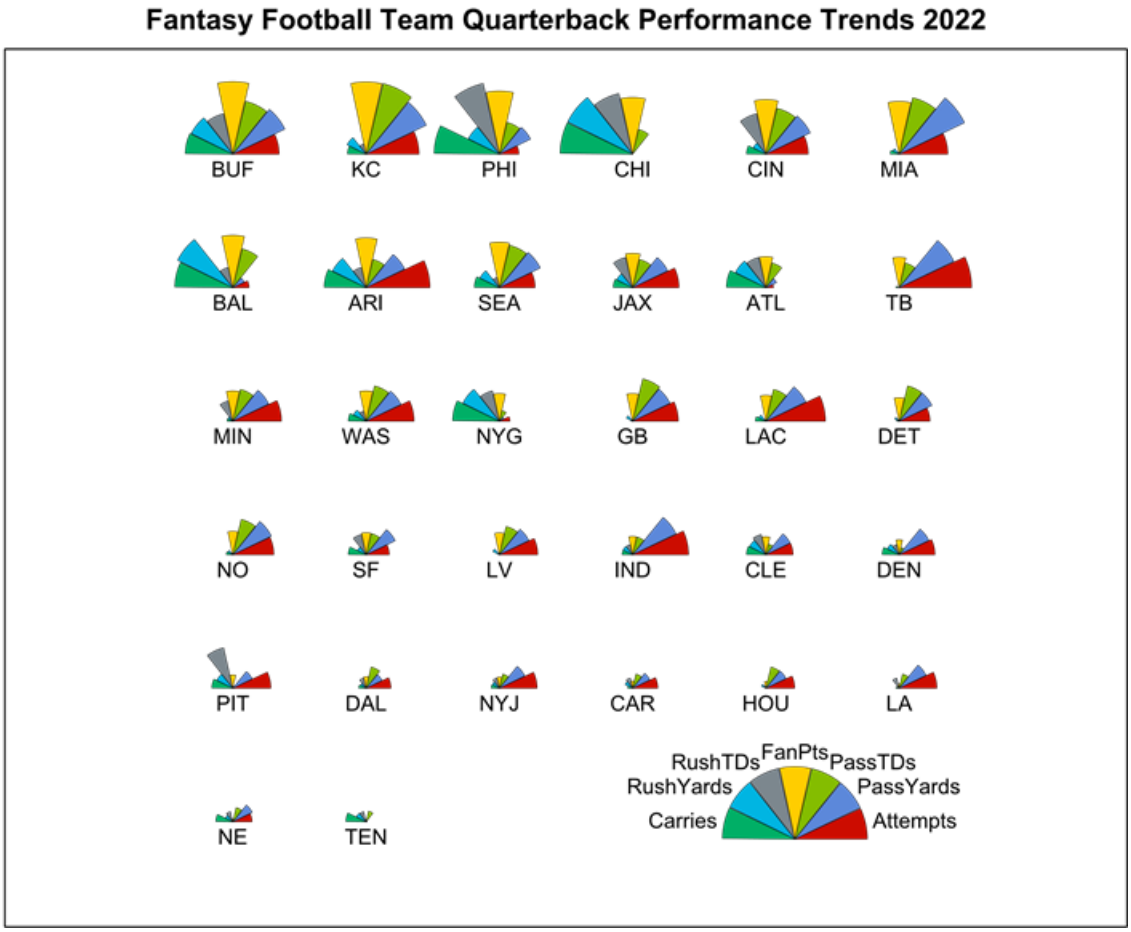


The trends of distinct count of Passing Epa and distinct count of Receiving Epa for Season broken down by Position Group vs. Season Type. Color shows details about distinct count of Passing Epa and distinct count of Receiving Epa. The view is filtered on Position Group, which keeps QB, RB, TE and WR.

Measure Names
- Distinct count of Passing Epa
- Distinct count of Receiving Epa

- **Visualizations**

1. Visualization 1



The world's constant technological advancements, the internet and media's wider access, and the millions of sports fans across the globe, has become a recipe resulting in the growth of fantasy sports. "The Fantasy Sports Market is valued at USD 21.39 billion in 2021 and is expected to reach USD 44.07 billion, with an estimated CAGR (compound annual growth rate) of 12.92% over the forecast period of 2022-2027" (https://www.reportlinker.com/p06309369/?utm_source=GNW). In the United States, the largest market is fantasy football and the first visualization is a star plot that highlights the total amount of fantasy points scored by the quarterback position of each team in the NFL in 2022.

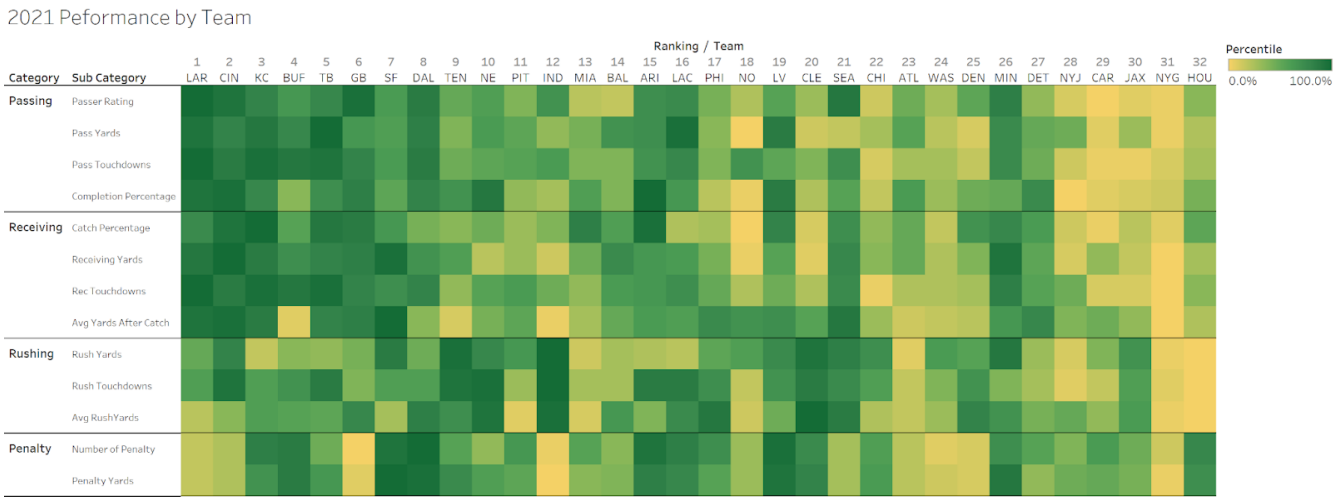
The data frame used is a subset of the 2022 offensive player stats, including categorical variables such as the team, and position. Quantitative variables were chosen to represent rushing and passing production through the number of the touchdowns scored, the amount of yardage gained, and the opportunities given. These variables help to highlight the main focus of the graph, fantasy points.

The visualization technique chosen is a star plot to show each NFL team in an organized layout. The individual star segments are manually configured to highlight the total fantasy points in the center, which is also used to sort each team in descending order. The left side of each star has three segments correlated to rushing performance, while the right side has another three segments that correlate to passing performance. The total number of touchdowns, total yardage gained, and total number of opportunities are positioned across from each other to organize each star pattern.

Further design patterns were chosen with the audience in consideration, from fantasy football players to general viewers. Because the graph was coded in R, many of the design features were able to be formatted. The ‘paletteer’ package is a collection of most color palettes in R, where the ‘star trek’ themed palette was chosen, consisting of seven discrete colors. This palette provides different hues to help distinguish the segments in the star plot, as opposed to a divergent or continuous palette that might make discerning the smaller stars difficult. The labels were imported to match each team’s star, and enlarged for the viewer. The spacing and layout of the stars were chosen to provide an organized view with no overlapping. Other aesthetics include the positioning of the legend, framing the plotted area, and keeping the stars half size as opposed to full, to better accentuate the difference in passing and rushing production.

The star plot allows the audience to easily compare not only the fantasy performance of quarterbacks for each team, but the manners in which they are productive as well. For instance, the top scoring team (Buffalo) appears to have a balanced attack of passing and rushing, while the second highest team (Kansas City) is prolific in passing, and the fourth highest scoring team (Chicago) is nearly completely rushing. Looking at a broader view, it appears that over half of the teams in the NFL have had low production from their quarterback position so far in the 2022-23 season. With more time, it would be interesting to see each specific quarterback in the league and their trends over the span of their careers, and with this create an animated time series graph in the manner of a spider or star plot.

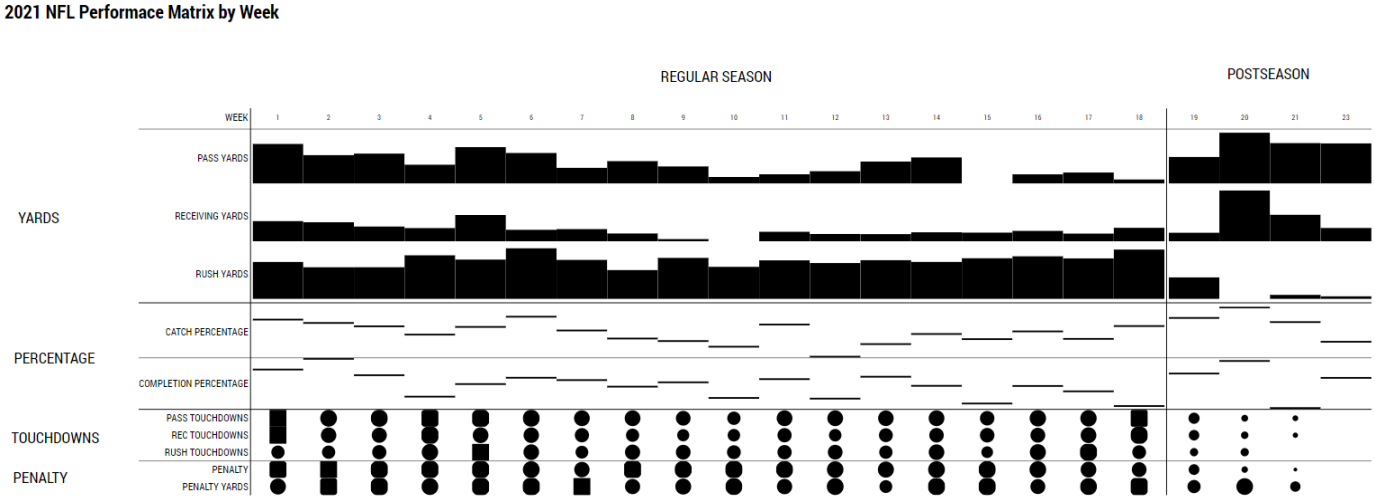
2. Visualization 2



The heatmap shows the 2021 NFL performance by Team. The x-axis is the four major matrices – Passing, Receiving, Rushing, and Penalty. The order of the x-axis is based on the correlation of the variables. When exploring the data, passing, and receiving show some position correlation, meaning the team with higher passing records will have higher receiving records. So, passing and receiving are placed together to show a strong pattern in the heatmap. Rushing and penalty are relatively randomly distributed. The y-axis is the team's name and the order is based on ranking so that the audience can see the relationship between ranking and the team's performance. The color represents the percentile of the subcategory for the team within the league. Green means high values of that variable. Continuous color palettes of adjacent colors were chosen to have high and low values stand out. I also tried the divergent color palettes since this graph shows the ranking and does not focus on the extreme values, so the presentation is a bit unclear with the divergent colors.

Based on the graph, we can observe the top-ranking teams normally have higher passing and receiving and the color becomes yellow following the ranking. The rushing seems not to have a strong pattern that the mid-ranking team has higher rushing values, and the bottom teams normally have low values with yellow colors. The top two teams have low penalties. This graph explains the performance matrix of teams based on the 2021 rankings.

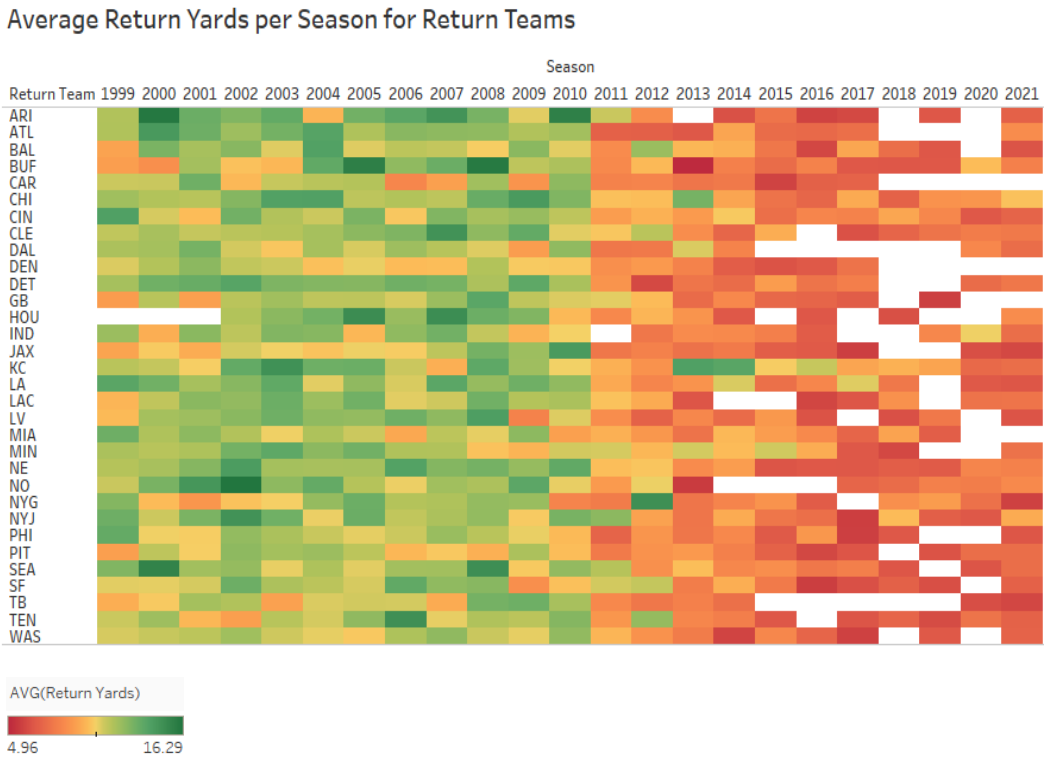
3. Visualization 3



The Bertin matrix shows the 2021 NFL performance by week. The X-axis is the number of weeks and it is divided into the regular season and postseason to show the trend of performance over the week but also compare the performance between the regular and postseason. The y-axis has four main offense matrices – yards, percentage, touchdowns, and penalty. For the yards, the bar chart is used. Although the line graph is ideal to show the trend, the line graph in bertifier.com is not a continuous line, and using the line graph cannot show the trend clearly sol decided to use the bar to present the yards for each week. The position of the black line shows the percentage record. The higher the position, the higher the value. The shape is used to show the count of touchdowns and penalties. There are some limitations of using bertifier.com such as the website does not generate the legend and hard to make labels to the graph.

The graph shows the passing and receiving yards are higher compared to the yards in the regular season. For the regular season, the passing and receiving yards started strong and continuously decreased till week 10 and slightly went up. There is no significant trend for rushing yards in the regular season but the values are very low in the postseason. Regarding the percentage, the catch and passing completion rates are normally higher in the postseason. The number of touchdowns and penalties is lower in the postseason.

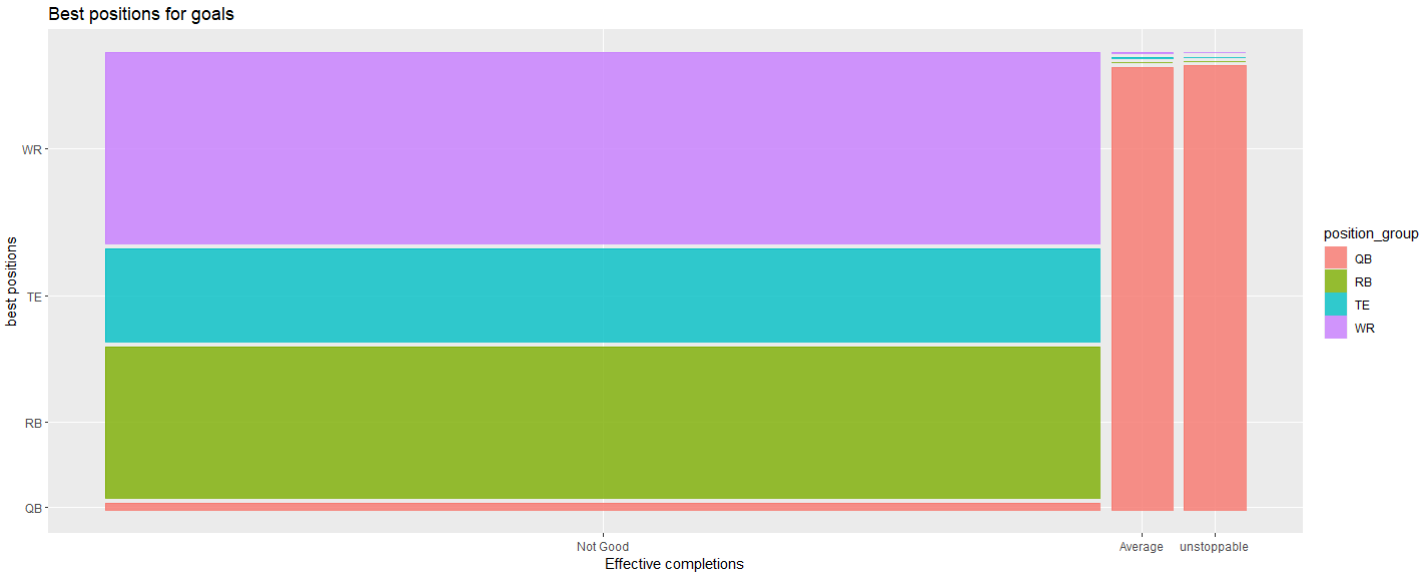
4. Visualization 4



The visualization that is used here is a Heat Map that shows the relationship between Average Return Yards per Season for the Return teams where return teams and seasons (from 1999 to 2021) are labeled and are plotted on the axis. “Return Yard” is a statistical measure that takes several forms and is an instrument that includes an offensive progress in the kick plays and is given by the sum of gains from line of scrimmage and the possession changes during the punt block or fumbles. It is a process which is measured by advancing the football towards the opposing team’s goal.

This graph is being used because it shows the rule change that happened in the year 2011 for kicking which is why it looks pretty red after 2011. The kickoff has been moved five yards forward to the 35-yard line as a result of a regulation change implemented by the National Football League in 2011. The primary goal of the modification was to reduce kickoff-related injuries, which had gotten worse over time. Additionally, the configuration for the kicking team was changed so that all players other than the kicker had to be lined up no farther than five yards from the line. Through this graph I can see that darker the green color is, more will be the return yards for the team in that season, lighter the orange-red color, less will be the return yard for the given team and this is the reason why Divergent color scheme is being used.

5. Visualization 5



The visualization that is used here is MOSAIC Plot that shows how effectively players in the position group are successfully completing a goal (from 1999-2021 season). On the X-axis the number of goals are divided into 3 parts: Not Good, Average & Unstoppable, and on Y-axis top 4 groups are displayed. QB position players are good at scoring the goals (the quarterback is usually considered the leader of the offense, and is often responsible for calling the play in the huddle. The quarterback also touches the ball on almost every offensive play, and is almost always the offensive player that throws forward passes.) Quarterbacks (QB) are also members of the attacking team. They line up directly behind the center, in the middle of the offensive team's line. A quarterback is usually the leader, and starts most plays. The quarterback is one of the most important players and quarterback is easily the toughest position to play in the NFL.

I began this visualization with just a gray-scale color pattern. I added color to the mosaic plot to make the visualization more engaging and highlight the differences between the different ordinal values. Additionally, I did not have the ordinal values positioned correctly in my rough drafts, and so I ordered the values of the variables from Not-Good to Unstopable to show the progression of the position group more clearly.

- **Analysis and Discussion**

My visualizations provide interesting information to the audience for the popular topics in the NFL. Users are able to visually observe the relationship between quarterback and fantasy points for season 2022, the pattern for performance in season 2021 from two different aspects – specific team performance for the whole season and weekly performance, the effect of kicking rule changes in 2011, and finally player contribution to the game based on the position.

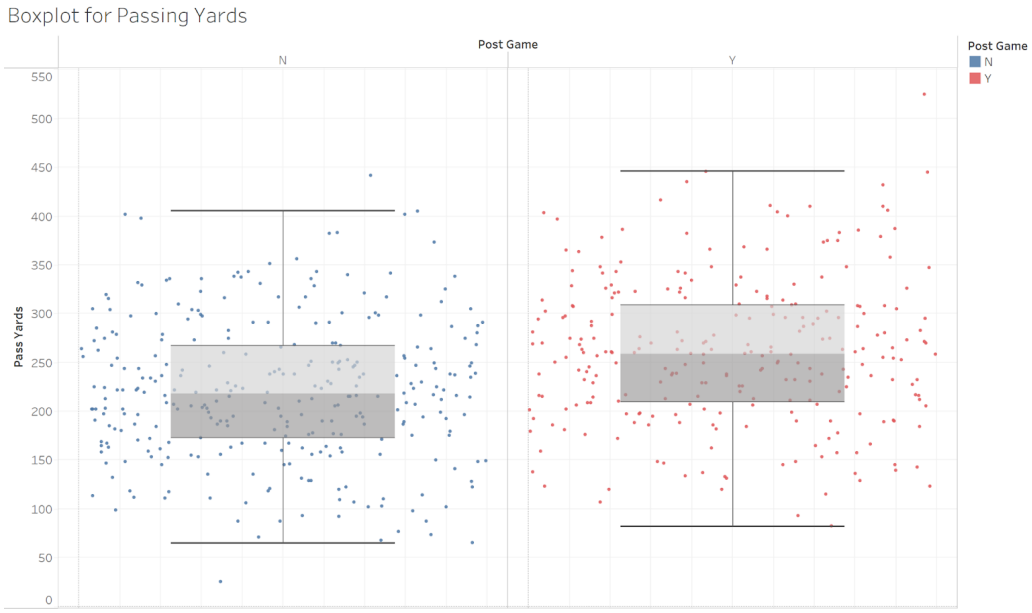
The visualizations show that some attributes play more important roles for teams or position performance compared to other attributes. For example, quarterbacks are generally considered key attack players, so the attributes related to a quarterback like passing yards and completions have a strong influence on the team's performance. There are other attributes, though conventionally considered critical to a team's performance, seem not to have strong relationships between them and game results based on the visualizations such as the number of attempts and penalty. Also, some of the visualizations present trends overtime. For example, the Bertin matrix shows that the whole league starts strong at the beginning of seasons but the passing and receiving trends went down.

Overall, the visualization reassures these key things: the importance of the quarterback, passing and receiving yards are significant matrices when evaluating performance, and the effectiveness of the rule changes.

- **Code and Results of Formative, Exploratory Data Analysis**

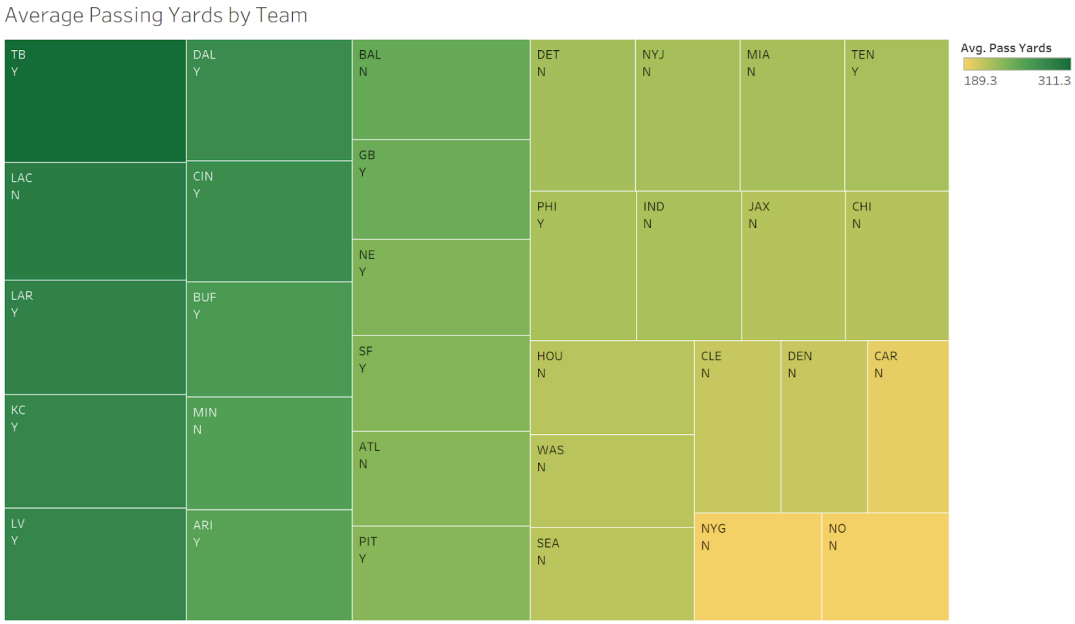
Other Exploratory Visualization

1. Boxplot - Passing yards for post-game teams vs non post-games teams

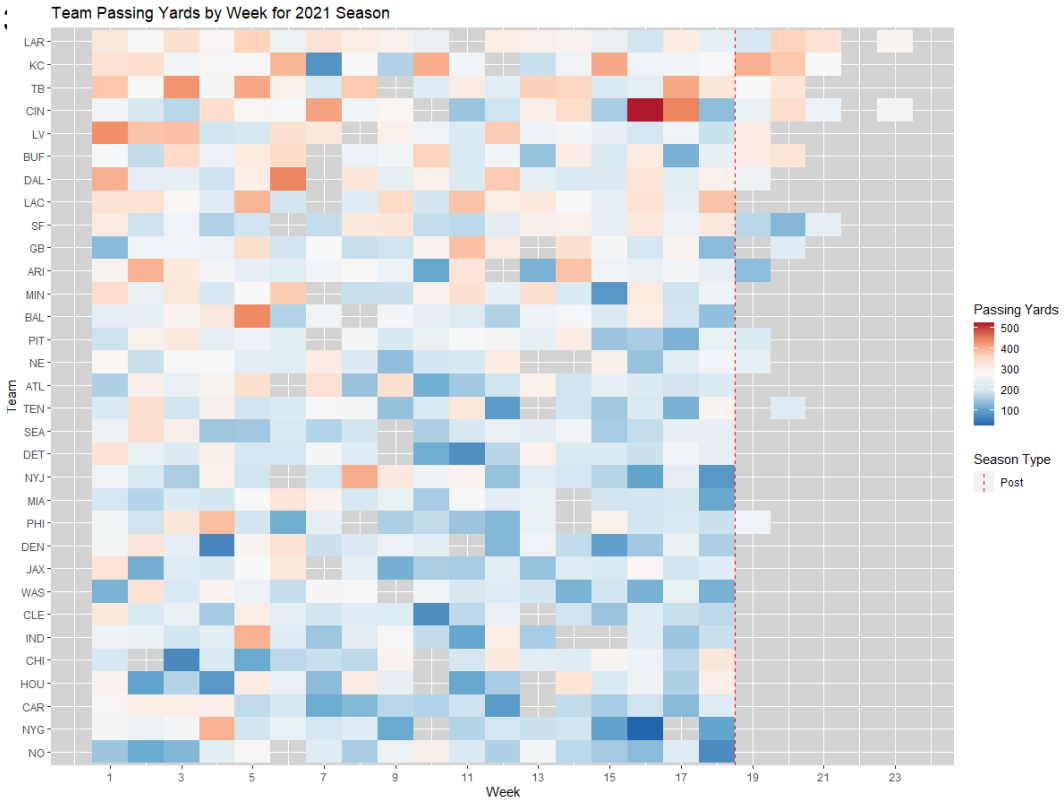


The boxplot explores the relationship between passing yards based on whether a team played in the postseason. It shows the teams playing in the postseason generally have better performance on passing yards because the box is slightly higher for post-game.

2. Heatmap - Average passing yards by team



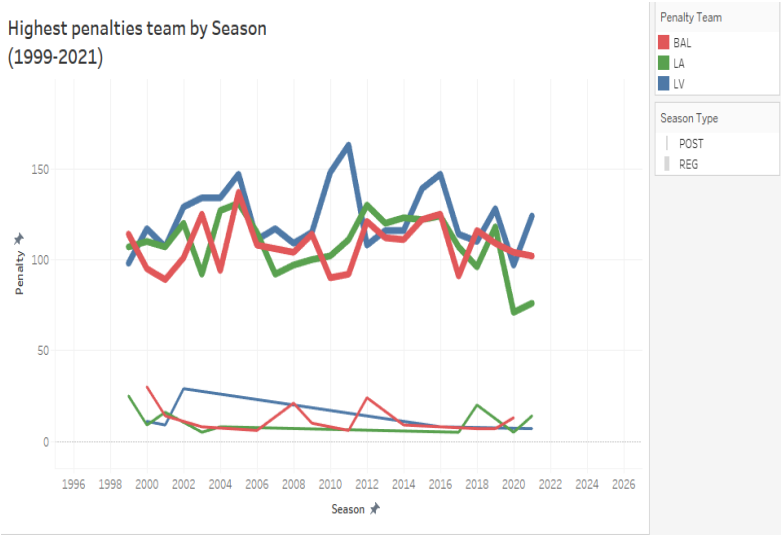
The heatmap shows the relationship between a team and the average passing yards for the 2021 season. I use this to discover the ranking of the average passing yards and see if the team with more passing yards went into the postseason. Most teams in green played in the postseason with some exceptions. For example, LAC has higher passing yards based on the color in the heatmap but the team did not play in the 2021 postseason.

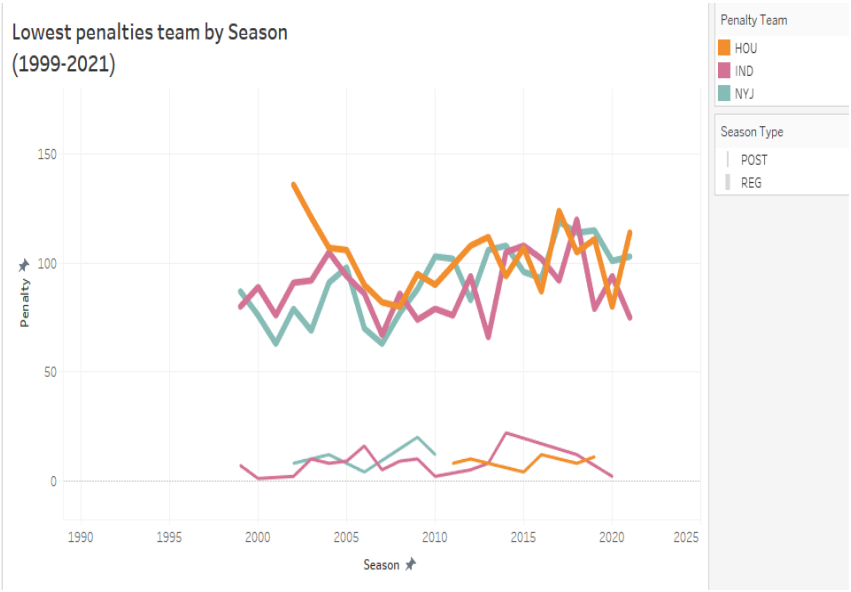


The direction of this visualization is to explain the relationship between weekly passing yards for each team. The x-axis represents the team abbreviation name and is sorted based on the average passing yards of the season in descending order. The y-axis means the week. The dashed line separates the regular season and postseason. The divergent colors are chosen to show the number of passing yards for two extreme values.

This graph simply shows that the teams that went into the post games normally the team with higher passing yards each week. The teams that did not go into the post-games performed worse with more blue throughout the season. However, there are some exceptions, for example, PIT has low average passing yards with most blue throughout the regular season but the team did make it to the postseason in 2021.

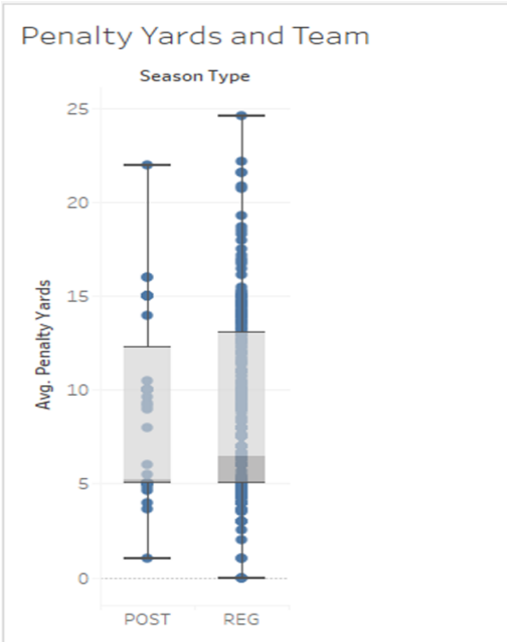
4. Line graph - Highest and Lowest penalties team by season from 1999-2021





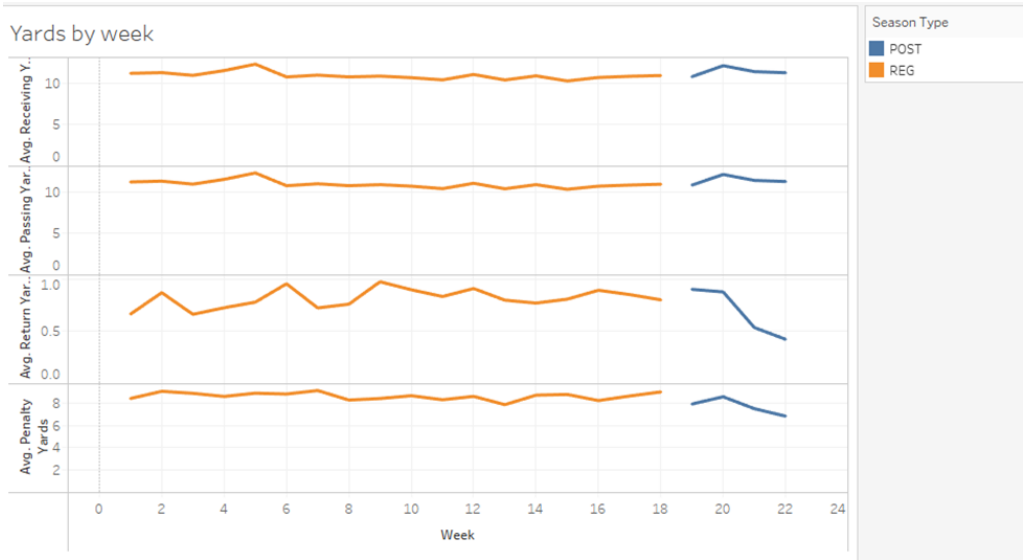
The line graph shows the top 3 three teams that have the lowest and highest number of penalties in a season from 1999 to 2021. The line graph is sorted based on Season-type as there are two season-type that is: Postseason and Regular Season.The line graph on that shows top 3 teams with highest number of penalties simply tells which teams has performed the worst in the each season and the one on that shows the top 3 teams with least number of penalties tells which teams has performed the best in each season.

5. **Boxplot - Average Penalty Yards from Postseason and Regular-Season**



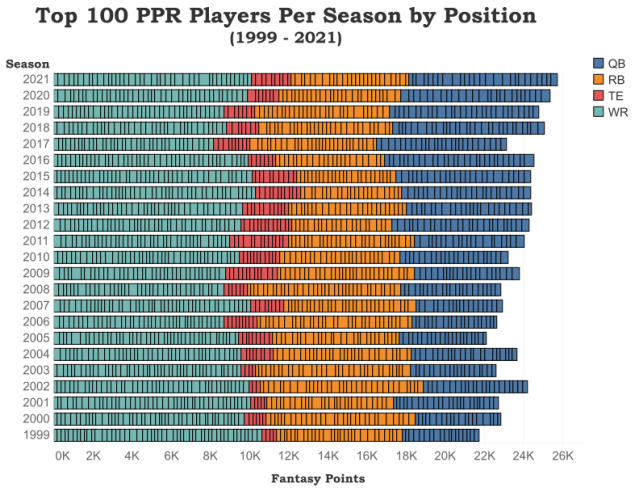
The boxplot shows the average number of penalty yards for both the seasons that are: the postseason and regular-season. Not much information was seen from the graph which is why I considered heatmap and picked that to show the relationship and used that as one of the main directions to observe NFL performance trends for the year 2021.

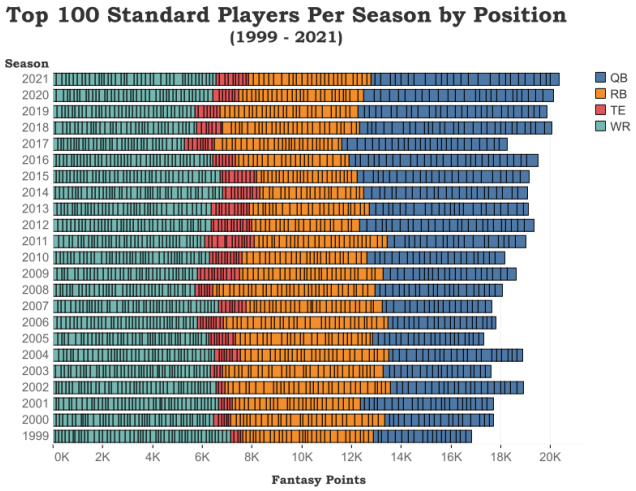
6. **Line graph showing penalty, return, passing and receiving yards by week for postseason and regular season.**



7. Top 100 Fantasy Players Per Season By Position

In fantasy football, the most common leagues have 12 players with a roster setup that uses 7 offensive players, so the number of offensive players that would start in a given week is 84. The next two graphs show the distribution of the top 100 highest scoring players per year regarding fantasy points. They concentrate on the players that are most likely to be starting on fantasy football teams, and disregard players who are less likely to make an impact. Between the years 1999 and 2021, the quarterback group has had the largest increase in number of players in the top 100, whereas tight ends have had a slight increase, the number of running backs appears to be decreasing, and the number of wide receivers has stayed relatively the same. This highlights the trend that quarterbacks have been increasing in their performance over the years.





R Code

```
#load data
Passing21 <- nextGenPassing21
#filter data
Passing21 <-nextGenPassing21 %>%
  filter(week!=0) %>%
  select(-c())%>%
  as.data.frame()
#define season type
Passing21$Season_f = factor(Passing21$season_type, levels=c('REG','POST'))
#generate heatmap
nfl_plot2<-ggplot(Passing21, aes(x=week, y=team_abbr, fill=pass_yards), na.rm=TRUE)+
  labs(title="Team Passing Yards by Week for 2021 Season",fill="Passing Yards")+
  ylab("Team")+
  xlab("Week")+
  theme(strip.background = element_rect(fill = "grey"))+
  scale_x_continuous(breaks=seq(1, 23, by=2))+
  geom_tile()+
  geom_vline(aes(xintercept=18.5, color="Post"), linetype="dashed", size=0.4) +
  scale_color_manual(name = "Season Type",values = c("Post"="red"))+
  scale_y_discrete(limits = c("NO", "NYG",
      "CAR","HOU","CHI","IND","CLE","WAS","JAX","DEN","PHI","MIA","NYJ","DET","SEA","TEN",'ATL',"NE","
      PIT","BAL","MIN","ARI","GB","SF","LAC","DAL","BUF","LV","CIN","TB","KC","LAR"))

nfl_plot2+
  scale_fill_distiller(palette = 'RdBu')+
  theme(panel.background=element_rect(fill='lightgrey'))

#Libraries used for the NFL dataset
library(nflreadr)
data = load_pbp(2021)
View(data)
library(nflfastR)
library(nflplotR)
library(ggrepel)

data_2020 = load_pbp(2020)
View(data_2020)

#Loading the entire season dataset from 1999 to 2021
data_multipleSeasons = load_pbp(1999:2021)
View(data_multipleSeasons)
write.csv(data_multipleSeasons, "C:/Users/hbali/OneDrive/Desktop/NFL DATA/NFL_DATA.csv", row.names
      = TRUE)

#Filtering the data based on the Penalty columns needed

new_data <- data_multipleSeasons[,c('penalty_team', 'penalty_player_name', 'penalty_yards', 'penalty_type',
      'season', 'week', 'penalty', 'season_type')]
```

```
write.csv(new_data, "C:/Users/tgupt/OneDrive/Desktop/NFL DATA/NFL_DATA_FILTER.csv", row.names =
    FALSE)

passing_data <- data_multipleSeasons[,c('passing_yards', 'penalty_type', 'penalty_team', 'season', 'week',
    'season_type')]
write.csv(new_data, "C:/Users/tgupt/OneDrive/Desktop/NFL DATA/Passing.csv", row.names = FALSE)

data_multipleSeasons %>%
  group_by(play_type) %>%
  summarize(n = n())

data_multipleSeasons %>%
  group_by(season) %>%
  summarize(n = n())

data_multipleSeasons %>%
  group_by(season_type) %>%
  summarize(n = n())

data_multipleSeasons %>%
  qqplot (data_multipleSeasons, aes(x = season, y = yardline_100)) + geom_point()
```

Code for Direction 1 (File Included in Submission):

```
install.packages("nflreadr")
install.packages("readr")
install.packages("tidyverse")
install.packages("viridis")
install.packages("gganimate")
install.packages("gifski")
install.packages("paletteer")

library(nflreadr)
library(readr)
library(dplyr)
library(ggplot2)
library(viridis)
library(gganimate)
library(gifski)
library(paletteer)

help("nflreadr")
help("readr")

# Load offense and kicking statistics for 2021 (1999-2022)

load_player_stats(seasons = 2021)
offense21 = load_player_stats(seasons = 2021,
                stat_type = c("offense"))
offense21
```

```
kicking21 <- data.frame(load_player_stats(seasons = 2021,
                stat_type = c("kicking")))

# Load next gen stats from 2021 (2016-2022)
nextGenPassing21 <- data.frame(load_nextgen_stats(seasons = 2021,
                                      stat_type = c("passing")))
nextGenRushing21 <- data.frame(load_nextgen_stats(seasons = 2021,
                                      stat_type = c("rushing")))
nextGenReceiving21 <- data.frame(load_nextgen_stats(seasons = 2021,
                                      stat_type = c("receiving")))

# Export 2021 statistics to csv file

write.csv(offense21, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense21.csv",
          row.names = FALSE)
write.csv(kicking21, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\kicking21.csv",
          row.names = FALSE)
write.csv(nextGenPassing21, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\nextGenPassing21.cs
      v",
          row.names = FALSE)
write.csv(nextGenRushing21, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\nextGenRushing21.cs
      v",
          row.names = FALSE)
write.csv(nextGenReceiving21, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\nextGenReceiving21.
      csv",
          row.names = FALSE)

# Graph - compare QBs and weekly passer rating
qbRating2021 <- nextGenPassing21 %>%
  select(week, player_display_name, passer_rating)
qbRating2021

qbRatingPlot2021 <- ggplot(qbRating2021, aes(x = week, y = passer_rating, color = player_display_name))
      +
  geom_line()
qbRatingPlot2021

# Graph average QB rating of QB's in 2021
aveQBRating <- qbRating2021 %>%
  group_by(player_display_name) %>%
  summarise(QBRating = mean(passer_rating, na.rm = TRUE))
aveQBRating <-aveQBRating[order(aveQBRating$QBRating), decreasing = TRUE]
aveQBRating
```

```
aveQBRatingPlot <- ggplot(aveQBRating, aes(x = player_display_name, y = QBRating, color =
    player_display_name)) +
  geom_col() +
  ggtitle("Average Quarterback Passer Rating in 2021") +
  labs(x = "Quarterback", y = "Passer Rating") +
  guides(col= guide_legend(title= "Quarterbacks")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
      legend.position = "bottom") +
  scale_x_discrete(limits = aveQBRating$player_display_name[order(-aveQBRating$QBRating)]) +
  scale_y_continuous(limits = c(0, 120), breaks = c(20, 40, 60, 80, 100, 120, 140))
aveQBRatingPlot

# Graph average quarterback passer rating for the entire nfl from 2016 to 2022
nextGenPassing16 <- data.frame(load_nextgen_stats(seasons = 2016,
                                    stat_type = c("passing")))
nextGenPassing17 <- data.frame(load_nextgen_stats(seasons = 2017,
                                    stat_type = c("passing")))
nextGenPassing18 <- data.frame(load_nextgen_stats(seasons = 2018,
                                    stat_type = c("passing")))
nextGenPassing19 <- data.frame(load_nextgen_stats(seasons = 2019,
                                    stat_type = c("passing")))
nextGenPassing20 <- data.frame(load_nextgen_stats(seasons = 2020,
                                    stat_type = c("passing")))
nextGenPassing21 <- data.frame(load_nextgen_stats(seasons = 2021,
                                    stat_type = c("passing")))
nextGenPassing22 <- data.frame(load_nextgen_stats(seasons = 2022,
                                    stat_type = c("passing")))
max_ln1 <- max(c(length(nextGenPassing16$passer_rating),
            length(nextGenPassing17$passer_rating)))
max_ln2 <- max(c(length(nextGenPassing18$passer_rating),
            length(nextGenPassing19$passer_rating)))
max_ln3 <- max(c(length(nextGenPassing20$passer_rating),
            length(nextGenPassing21$passer_rating)))
max_ln4 <- max(max_ln2, max_ln1)
max_ln5 <- max(max_ln3, length(nextGenPassing21$passer_rating))
max_ln <- max(max_ln4, max_ln5)
max_ln
#nextGenPassing < data.frame(QBR2016 = c(nextGenPassing16$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing16$passer_rating)))),
#                       QBR2017 = c(nextGenPassing17$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing17$passer_rating)))),
#                       QBR2018 = c(nextGenPassing18$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing18$passer_rating)))),
#                       QBR2019 = c(nextGenPassing19$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing19$passer_rating)))),
#                       QBR2020 = c(nextGenPassing20$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing20$passer_rating)))),
#                       QBR2021 = c(nextGenPassing21$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing21$passer_rating)))),
```

```
#                       QBR2022 = c(nextGenPassing22$passer_rating, rep(NA, abs(max_ln -
    length(nextGenPassing22$passer_rating)))))
QBR2016 <- c(nextGenPassing16$passer_rating)
QBR2017 <- c(nextGenPassing17$passer_rating)
QBR2018 <- c(nextGenPassing18$passer_rating)
QBR2019 <- c(nextGenPassing19$passer_rating)
QBR2020 <- c(nextGenPassing20$passer_rating)
QBR2021 <- c(nextGenPassing21$passer_rating)
QBR2022 <- c(nextGenPassing22$passer_rating)

QBR2016
QBR2017
QBR2018
QBR2019
QBR2020
QBR2021
QBR2022

length(QBR2016) <- max_ln
length(QBR2017) <- max_ln
length(QBR2018) <- max_ln
length(QBR2019) <- max_ln
length(QBR2020) <- max_ln
length(QBR2021) <- max_ln
length(QBR2022) <- max_ln

QBR16 <- as.data.frame(QBR2016) %>%
  mutate(QBRYear = "2016")
QBR17 <- as.data.frame(QBR2017) %>%
  mutate(QBRYear = "2017")
QBR18 <- as.data.frame(QBR2018) %>%
  mutate(QBRYear = "2018")
QBR19 <- as.data.frame(QBR2019) %>%
  mutate(QBRYear = "2019")
QBR20 <- as.data.frame(QBR2020) %>%
  mutate(QBRYear = "2020")
QBR21 <- as.data.frame(QBR2021) %>%
  mutate(QBRYear = "2021")
QBR22 <- as.data.frame(QBR2022) %>%
  mutate(QBRYear = "2022")

nextGenPassing <- data.frame(QBR2016,
                             QBR2017,
                             QBR2018,
                             QBR2019,
                             QBR2020,
                             QBR2021,
                             QBR2022)
nextGenPassing
```

```
QBR <- data.frame(c(QBR2016, QBR2017, QBR2018, QBR2019, QBR2020, QBR2021, QBR2022),
                  c(QBR16$QBRYear, QBR17$QBRYear, QBR18$QBRYear, QBR19$QBRYear, QBR20$QBRYear,
      QBR21$QBRYear, QBR22$QBRYear))
colnames(QBR) <- c("QuarterbackRating", "Year")
QBR

QB <- c(QBR2016, QBR2017, QBR2018, QBR2019, QBR2020, QBR2021, QBR2022)
QB

QBRPlot <- ggplot(QBR, aes(x =  Year, y = QuarterbackRating)) +
  geom_violin(aes(color = Year)) +
  geom_boxplot(width = .1) +
  geom_jitter(aes(color = Year), height = 0.01, shape = 13) +
  scale_color_viridis(discrete = TRUE) +
  ggtitle("Distribution of Quarterback Ratings Between 2016 - 2022") +
  labs(x = "Year", y = "Quarterback Passer Rating (Max 158.3)") +
  theme(panel.background = element_rect((fill = "#B2B2B2")),
        legend.background = element_rect(fill = "#B2B2B2"))
QBRPlot


# Touchdowns and fantasy points per position 2021

posTD <- offense21 %>%
  select(position, passing_tds, rushing_tds, receiving_tds, fantasy_points, fantasy_points_ppr) %>%
  summarise(passingTD = sum(passing_tds, na.rm = TRUE),
            rushingTD = sum(rushing_tds, na.rm = TRUE),
            receivingTD = sum(receiving_tds, na.rm = TRUE),
            fantasyPoints = sum(fantasy_points, na.rm = TRUE),
            fantasPointsPPR = sum(fantasy_points_ppr, na.rm = TRUE))
posTD

posTDPlot <- ggplot(posTD) +
  geom_col()
posTDPlot


offense99 = load_player_stats(seasons = 1999,
                              stat_type = c("offense"))
offense00 = load_player_stats(seasons = 2000,
                              stat_type = c("offense"))
offense01 = load_player_stats(seasons = 2001,
                              stat_type = c("offense"))
offense02 = load_player_stats(seasons = 2002,
                              stat_type = c("offense"))
offense03 = load_player_stats(seasons = 2003,
                              stat_type = c("offense"))
offense04 = load_player_stats(seasons = 2004,
                              stat_type = c("offense"))
```

```
offense05 = load_player_stats(seasons = 2005,
                        stat_type = c("offense"))
offense06 = load_player_stats(seasons = 2006,
                        stat_type = c("offense"))
offense07 = load_player_stats(seasons = 2007,
                        stat_type = c("offense"))
offense08 = load_player_stats(seasons = 2008,
                        stat_type = c("offense"))
offense09 = load_player_stats(seasons = 2009,
                        stat_type = c("offense"))
offense10 = load_player_stats(seasons = 2010,
                        stat_type = c("offense"))
offense11 = load_player_stats(seasons = 2011,
                        stat_type = c("offense"))
offense12 = load_player_stats(seasons = 2012,
                        stat_type = c("offense"))
offense13 = load_player_stats(seasons = 2013,
                        stat_type = c("offense"))
offense14 = load_player_stats(seasons = 2014,
                        stat_type = c("offense"))
offense15 = load_player_stats(seasons = 2015,
                        stat_type = c("offense"))
offense16 = load_player_stats(seasons = 2016,
                        stat_type = c("offense"))
offense17 = load_player_stats(seasons = 2017,
                        stat_type = c("offense"))
offense18 = load_player_stats(seasons = 2018,
                        stat_type = c("offense"))
offense19 = load_player_stats(seasons = 2019,
                        stat_type = c("offense"))
offense20 = load_player_stats(seasons = 2020,
                        stat_type = c("offense"))
offense21 = load_player_stats(seasons = 2021,
                        stat_type = c("offense"))
offense22 = load_player_stats(seasons = 2022,
                        stat_type = c("offense"))



write.csv(offense99, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense99.csv",
      row.names = FALSE)
write.csv(offense00, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense00.csv",
      row.names = FALSE)
write.csv(offense01, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense01.csv",
      row.names = FALSE)
write.csv(offense02, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense02.csv",
```

```
        row.names = FALSE)
write.csv(offense03, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense03.csv",
        row.names = FALSE)
write.csv(offense04, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense04.csv",
        row.names = FALSE)
write.csv(offense05, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense05.csv",
        row.names = FALSE)
write.csv(offense06, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense06.csv",
        row.names = FALSE)
write.csv(offense07, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense07.csv",
        row.names = FALSE)
write.csv(offense08, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense08.csv",
        row.names = FALSE)
write.csv(offense09, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense09.csv",
        row.names = FALSE)
write.csv(offense10, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense10.csv",
        row.names = FALSE)
write.csv(offense11, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense11.csv",
        row.names = FALSE)
write.csv(offense12, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense12.csv",
        row.names = FALSE)
write.csv(offense13, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense13.csv",
        row.names = FALSE)
write.csv(offense14, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense14.csv",
        row.names = FALSE)
write.csv(offense15, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense15.csv",
        row.names = FALSE)
write.csv(offense16, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense16.csv",
        row.names = FALSE)
write.csv(offense17, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense17.csv",
        row.names = FALSE)
write.csv(offense18, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense18.csv",
        row.names = FALSE)
```

```
write.csv(offense19, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense19.csv",
      row.names = FALSE)
write.csv(offense20, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense20.csv",
      row.names = FALSE)
write.csv(offense21, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense21.csv",
      row.names = FALSE)
write.csv(offense22, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\offense22.csv",
      row.names = FALSE)

# Create new dataframe for time series
# Create columns for fantasy points per year (regular and ppr)

year <- c("1999", "2000", "2001", "2002", "2003", "2004", "2005", "2006",
          "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014",
          "2015", "2016", "2017", "2018", "2019", "2020", "2021")
yearlyFF <- c(sum(offense99$fantasy_points), sum(offense00$fantasy_points),
           sum(offense01$fantasy_points), sum(offense02$fantasy_points),
           sum(offense03$fantasy_points), sum(offense04$fantasy_points),
           sum(offense05$fantasy_points), sum(offense06$fantasy_points),
           sum(offense07$fantasy_points), sum(offense08$fantasy_points),
           sum(offense09$fantasy_points), sum(offense10$fantasy_points),
           sum(offense11$fantasy_points), sum(offense12$fantasy_points),
           sum(offense13$fantasy_points), sum(offense14$fantasy_points),
           sum(offense15$fantasy_points), sum(offense16$fantasy_points),
           sum(offense17$fantasy_points), sum(offense18$fantasy_points),
           sum(offense19$fantasy_points), sum(offense20$fantasy_points),
           sum(offense21$fantasy_points))
yearlyFFPPR <- c(sum(offense99$fantasy_points_ppr), sum(offense00$fantasy_points_ppr),
                 sum(offense01$fantasy_points_ppr), sum(offense02$fantasy_points_ppr),
                 sum(offense03$fantasy_points_ppr), sum(offense04$fantasy_points_ppr),
                 sum(offense05$fantasy_points_ppr), sum(offense06$fantasy_points_ppr),
                 sum(offense07$fantasy_points_ppr), sum(offense08$fantasy_points_ppr),
                 sum(offense09$fantasy_points_ppr), sum(offense10$fantasy_points_ppr),
                 sum(offense11$fantasy_points_ppr), sum(offense12$fantasy_points_ppr),
                 sum(offense13$fantasy_points_ppr), sum(offense14$fantasy_points_ppr),
                 sum(offense15$fantasy_points_ppr), sum(offense16$fantasy_points_ppr),
                 sum(offense17$fantasy_points_ppr), sum(offense18$fantasy_points_ppr),
                 sum(offense19$fantasy_points_ppr), sum(offense20$fantasy_points_ppr),
                 sum(offense21$fantasy_points_ppr))


NFLFF <- data.frame(year, yearlyFF, yearlyFFPPR)

write.csv(NFLFF, file =
    "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\yearlyFF.csv",
      row.names = FALSE)
```

```
# Combine all offense datasheets

allOffense <- rbind(offense99, offense00, offense01, offense02, offense03, offense04, offense05,
                    offense06, offense07, offense08, offense09, offense10, offense11, offense12,
                    offense13, offense14, offense15, offense16, offense17, offense18, offense19,
                    offense20, offense21)
allOffense
write.csv(allOffense, file =
      "C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\AllOffense.csv",
          row.names = FALSE)


# Animation of fantasy points time series

posPoints <- allOffense %>%
  select(season, position, fantasy_points, fantasy_points_ppr) %>%
  filter(position == "QB" | position == "WR" | position == "RB" | position =="TE") %>%
  group_by(season, position) %>%
  summarise(average = mean(fantasy_points, na.rm =  TRUE))
posPoints

posPointsPlot <- ggplot(posPoints, aes(x = season, y = average, color = position)) +
  geom_point(size = 3) +
  geom_line() +
  ggtitle(label = "Average Standard Fantasy Points Per Season By Position",
          subtitle = "(1999 - 2021)") +
  labs(x = "Season",
       y = "Fantasy Points") +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(1998, 2022),
                     breaks = c(1999, 2001, 2003, 2005, 2007, 2009, 2011,
                                2013, 2015, 2017, 2019, 2021)) +
  scale_y_continuous(limits = c(0, 16),
                     breaks = c(0, 3, 6, 9, 12, 15)) +
  guides(x = guide_axis(angle = 90)) +
  transition_reveal(season) +
  shadow_wake(0.2)
posPointsPlot

posPoints2 <- allOffense %>%
  select(season, position, fantasy_points, fantasy_points_ppr) %>%
  filter(position == "QB" | position == "WR" | position == "RB" | position =="TE") %>%
  group_by(season, position) %>%
  summarise(total = sum(fantasy_points, na.rm =  TRUE))
posPoints2

posPointsPlot2 <- ggplot(posPoints2, aes(x = season, y = total, color = position)) +
  geom_point(size = 3) +
```

```
  geom_line() +
  ggtitle(label = "Total Standard Fantasy Points Per Season By Position",
          subtitle = "(1999 - 2021)") +
  labs(x = "Season",
       y = "Fantasy Points") +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(1998, 2022),
                     breaks = c(1999, 2001, 2003, 2005, 2007, 2009, 2011,
                                2013, 2015, 2017, 2019, 2021)) +
  scale_y_continuous(limits = c(0, 13000),
                     breaks = c(0, 2500, 5000, 7500, 10000, 12500)) +
  guides(x = guide_axis(angle = 90)) +
  transition_reveal(season) +
  shadow_wake(0.2)
posPointsPlot2

posPointsPPR <- allOffense %>%
  select(season, position, fantasy_points, fantasy_points_ppr) %>%
  filter(position == "QB" | position == "WR" | position == "RB" | position =="TE") %>%
  group_by(season, position) %>%
  summarise(average = mean(fantasy_points_ppr, na.rm =  TRUE))
posPointsPPR

posPointsPPRPlot <- ggplot(posPointsPPR, aes(x = season, y = average, color = position)) +
  geom_point(size = 3) +
  geom_line() +
  geom_col() +
  ggtitle(label = "Average PPR Fantasy Points Per Season By Position",
          subtitle = "(1999 - 2021)") +
  labs(x = "Season",
       y = "Fantasy Points") +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(1998, 2022),
                     breaks = c(1999, 2001, 2003, 2005, 2007, 2009, 2011,
                                2013, 2015, 2017, 2019, 2021)) +
  scale_y_continuous(limits = c(0, 16),
                     breaks = c(0, 3, 6, 9, 12, 15)) +
  guides(x = guide_axis(angle = 90)) +
  transition_reveal(season) +
  shadow_wake(0.2)
posPointsPPRPlot

posPointsPPR2 <- allOffense %>%
  select(season, position, fantasy_points, fantasy_points_ppr) %>%
  filter(position == "QB" | position == "WR" | position == "RB" | position =="TE") %>%
  group_by(season, position) %>%
  summarise(total = sum(fantasy_points_ppr, na.rm =  TRUE))
posPointsPPR2
```

```
posPointsPPRPlot2 <- ggplot(posPointsPPR2, aes(x = season, y = total, color = position)) +
  geom_point(size = 3) +
  geom_line() +
  ggtitle(label = "Total PPR Fantasy Points Per Season By Position",
          subtitle = "(1999 - 2021)") +
  labs(x = "Season",
       y = "Fantasy Points") +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  scale_x_continuous(limits = c(1998, 2022),
                     breaks = c(1999, 2001, 2003, 2005, 2007, 2009, 2011,
                                2013, 2015, 2017, 2019, 2021)) +
  scale_y_continuous(limits = c(0, 20000),
                     breaks = c(0, 3000, 6000, 9000, 12000, 15000, 18000)) +
  guides(x = guide_axis(angle = 90)) +
  transition_reveal(season) +
  shadow_wake(0.2)
posPointsPPRPlot2


anim_save("C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\AveStandardP
      oints.gif", posPointsPlot)
anim_save("C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\AvePPRPoint
      s.gif", posPointsPPRPlot)
anim_save("C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\TotalStandard
      Points.gif", posPointsPlot2)
anim_save("C:\\Users\\dongh\\Desktop\\Depaul\\Masters\\DSC465-DataVisualization\\Project\\TotalPPRPoint
      s.gif", posPointsPPRPlot2)




# Milestone 3 Quick plots

posPointsPlot2b <- ggplot(posPoints2, aes(x = season, y = total, color = position)) +
  geom_col() +
  #geom_line() +
  ggtitle(label = "Total Standard Fantasy Points Per Season By Position",
          subtitle = "(1999 - 2021)") +
  labs(x = "Season",
       y = "Fantasy Points") +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  #scale_x_continuous(limits = c(1998, 2022),
  #                   breaks = c(1999, 2001, 2003, 2005, 2007, 2009, 2011,
  #                              2013, 2015, 2017, 2019, 2021)) +
  #scale_y_continuous(limits = c(0, 13000),
  #                   breaks = c(0, 2500, 5000, 7500, 10000, 12500)) +
  guides(x = guide_axis(angle = 90))
```

```
  #transition_reveal(season) +
  #shadow_wake(0.2)
posPointsPlot2b

# Stars plot

stars22 <- offense22 %>%
  select(recent_team, season_type, completions, attempts,
         passing_yards, passing_tds, interceptions, passing_air_yards,
         passing_first_downs, passing_epa, pacr, dakota) %>%
  filter(season_type == "REG") %>%
  group_by(recent_team) %>%
  summarise(completions = sum(completions, na.rm = TRUE),
            attempts = sum(attempts, na.rm = TRUE),
            passing_yards = sum(passing_yards, na.rm = TRUE),
            passing_tds = sum(passing_tds, na.rm = TRUE),
            #interceptions = sum(interceptions, na.rm = TRUE),
            #passing_air_yards = sum(passing_air_yards, na.rm = TRUE),
            passing_first_downs = sum(passing_first_downs, na.rm = TRUE),
            passing_epa = sum(passing_epa, na.rm = TRUE))
            #pacr = sum(pacr, na.rm = TRUE),
            #dakota = sum(dakota, na.rm = TRUE)) %>%

stars22

stars(stars22[, 2:7], full = FALSE, scale = TRUE,
      labels = stars22$recent_team, len = 1,
      draw.segments = TRUE, key.loc = c(12,2),
      main = "Quarterback Passing Trends 2022") +
  plotStarLegend(main = "Legend")


# Quarterback stars plot 2022

offense22 = load_player_stats(seasons = 2022,
                              stat_type = c("offense"))

qbStars22 <- offense22 %>%
  select(recent_team, position, player_display_name, season_type, headshot_url,
         attempts, passing_yards, passing_tds,
         fantasy_points,
         rushing_tds, rushing_yards, carries) %>%
  filter(season_type == "REG", position == "QB") %>%
  group_by(recent_team) %>%
  summarise(Attempts = sum(attempts, na.rm = TRUE),
            PassYards = sum(passing_yards, na.rm = TRUE),
            PassTDs = sum(passing_tds, na.rm = TRUE),
            FanPts = sum(fantasy_points, na.rm = TRUE),
            RushTDs = sum(rushing_tds, na.rm = TRUE),
            RushYards = sum(rushing_yards, na.rm = TRUE),
```

```
          Carries = sum(carries, na.rm = TRUE)) %>%
  arrange(desc(FanPts))

qbStars22
palette(paletteer_d("ggsci::uniform_startrek"))
stars(qbStars22[, 2:8],
      full = FALSE,
      scale = TRUE,
      labels = qbStars22$recent_team,
      frame.plot = TRUE,
      len = 1.25,
      locations = NULL,
      draw.segments = TRUE,
      key.loc = c(12,2),
      cex = 0.9,
      main = "Fantasy Football Team Quarterback Performance Trends 2022",
      sub = "Breakdown of Fantasy Points")
```

Code For Mosaic Plot

```
library(plyr)
library(dplyr)
library(ggplot2)
library(ggmosaic)

nfl <- read.csv("C:\\Users\\CDMStudent14\\Desktop\\DSC-465\\Project\\nfl_data.csv")

data_2 <- nfl[complete.cases(nfl$position_group), ]

df1 <- data_2[!(data_2$position_group=="SPEC" | data_2$position_group=="OL" |
                data_2$position_group=="DB" | data_2$position_group=="DL" | data_2$position_group=="LB" ),]

df1$completions <- as.numeric(df1$completions)

breaks <- c(0,5,20,50)
tags <- c("Not Good","Average","unstoppable")

com_rate <- cut(df1$completions,
                breaks = breaks,
                include.lowest = TRUE,
                right = FALSE,
                labels = tags)

mosaic_ex <- ggplot(data = df1)+
  geom_mosaic(aes(x=product(com_rate), fill = position_group))+
  labs(y="best positions", x="Effective completions", title = "Best positions for goals")

mosaic_ex
```