NFL Big Data Bowl

What are we looking for?

This year's competition turns to a new type of data -- what happens before the snap -- to generate creative insights and actionable predictions into what the offense or defense does after the snap.

Available Tracks and deliverables?

Metric Track (Data Analysis and Prediction)

A Jupyter Notebook (compatible with Google Colab) that contains:

- Analysis using player tracking data, ideally focusing on a narrow aspect (e.g., team or player tendencies).
- Fewer than 2,000 words and less than 10 tables/figures (important for readability).
- Code to support analysis (keep most code in an appendix to keep the main notebook readable).
- Public access to the notebook on Kaggle or a similar platform by the submission deadline.
- **Submission Format:** Ensure the notebook is public and all team members are listed as collaborators. Notebooks not using player tracking data won't be scored.

2. Coaching Presentation Track

A slide presentation (PDF format, uploaded as a Kaggle dataset) that:

- Provides an analysis tailored to coaches, such as a scouting report.
- Contains no more than 20 slides.
- Includes supporting documents and linked notebooks for any additional data/code details.
- Publicly accessible on or before the submission deadline.
- **Submission Format:** Provide a URL to the Kaggle Dataset for review.

Evaluation Criteria

The scoring across tracks is broken down as follows:

- **Football Score (30%)** - Practicality and uniqueness of insights.
- **Data Science Score (30%)** - Correctness, innovative analysis, and backed-up claims.
- **Report Score (20%)** - Clarity, structure, and purpose.
- **Data Visualization Score (20%)** - Quality and clarity of visualizations.

Plan

1. **Data Exploration & Cleaning:**
    - Understand the main characteristics of each dataset, especially focusing on the tracking, play, and player data.
    - Investigate any missing values, outliers, or inconsistencies, particularly in time-based tracking data, to ensure clean inputs for modeling.

2. **Feature Engineering:**
    - **Formation Patterns**: Analyze team formations, player movements, shifts, and alignments before the snap.
    - **Pre-snap Tendencies**: Identify features like player positioning, acceleration, and orientation (e.g., specific formations linked to pass or run plays)
    - **Situational Variables**: Capture play-level features such as down, distance, yard line, and game context (quarter, score difference).

3. **Predictive Modeling**:
    - **Play Type Prediction (Run vs. Pass)**: Model pre-snap data to predict play type, using classifiers that might handle categorical features like formation and alignment.
    - **Player Behavior Prediction**: Predict individual movements or actions, especially of key players like the quarterback and running backs, based on pre-snap configurations.

4. **Evaluation & Iteration**:
    - Optimize models based on accuracy, precision, and F1 scores to ensure actionable and realistic outputs.

5. **Documentation and Visualization**:
    - Plan for clear, engaging visuals—heatmaps, trajectory graphs, and animation-like sequences can show player positions and orientations.
    - Keep the report straightforward with charts that align with the competition's Data Visualization and Report scoring criteria.

Data Exploration(Initially done in python)

```
 # Load the datasets
import pandas as pd
games_df = pd.read_csv('nfl/Games.csv')
player_plays_df = pd.read_csv('nfl/Player-Plays.csv')
players_df = pd.read_csv('nfl/Players.csv')

# Check for missing values in each dataset
for df, name in zip([games_df, player_plays_df, players_df],
['Games', 'Player Plays', 'Players']):
    print(f"\nMissing values in {name} Dataset:")
    print(df.isnull().sum())
```

What did we find out?

For the datasets Games, Plays and Players, only the attribute date of birth had missing values.

```
birthDate      487
```

Fix?

We could either drop or impute those values but we didn't because the birth date wasn't that essential of a data.

```
# Check for duplicate rows
for df, name in zip([games_df, player_plays_df, players_df],
['Games', 'Player Plays', 'Players']):
    print(f"\nDuplicate rows in {name} Dataset:",
df.duplicated().sum())
    df.drop_duplicates(inplace=True)  # Drop duplicates if any
```

We didn't find any duplicate values for the table! As of now.

```
# Ensure no negative or implausible values in time-related columns
print("Checking for negative or implausible values in time
columns...")
time_based_cols = ['gameClock', 'timeToThrow', 'timeInTackleBox']
for col in time_based_cols:
    if col in player_plays_df.columns:
        invalid_times = player_plays_df[player_plays_df[col] < 0]
        print(f"Invalid values in {col}:", invalid_times.shape[0])
```

Valid times present!

Descriptive summary (R Code)

A descriptive summary in data analysis refers to an overview of the key characteristics or features of a dataset. It helps us understand the general structure of the data. It provides insights into its central tendencies, variability, and potential patterns. The purpose is to make it easier to analyze and interpret.

Key components
Mean (Average): The arithmetic average of the data.
Median: The middle value when the data is sorted in order.
Mode: The most frequent value in the dataset.
Standard Deviation (SD): Measures the spread of data points around the mean.
Variance: The square of the standard deviation.
Range: The difference between the maximum and minimum values in the dataset.
Frequency Distribution:

```
# libraries
library(dplyr)
library(ggplot2)

# Load datasets
games <- read.csv("gdrive/MyDrive/nfl/Games.csv")
player_plays <- read.csv("gdrive/MyDrive/nfl/Player-Plays.csv")
players <- read.csv("gdrive/MyDrive/nfl/Players.csv")
```

Basic Summary Statistics

```
# Games dataset summary
cat("Summary of Games dataset:\n")
print(summary(games))

# Player-Plays dataset summary
cat("\nSummary of Player-Plays dataset:\n")
print(summary(player_plays))

# Players dataset summary
cat("\nSummary of Players dataset:\n")
print(summary(players))
```

Output-

Summary of Games dataset:

```
     gameId              season          week         gameDate
 Min.   :2.022e+09   Min.   :2022   Min.   :1.000   Length:136
 1st Qu.:2.022e+09   1st Qu.:2022   1st Qu.:3.000   Class :character
 Median :2.022e+09   Median :2022   Median :5.000   Mode  :character
 Mean   :2.022e+09   Mean   :2022   Mean   :4.846
 3rd Qu.:2.022e+09   3rd Qu.:2022   3rd Qu.:7.000
 Max.   :2.022e+09   Max.   :2022   Max.   :9.000
 gameTimeEastern    homeTeamAbbr       visitorTeamAbbr    homeFinalScore
 Length:136         Length:136         Length:136         Min.   : 3.00
 Class :character   Class :character   Class :character   1st Qu.:17.00
 Mode  :character   Mode  :character   Mode  :character   Median :22.50
                                                          Mean   :22.67
                                                          3rd Qu.:27.00
                                                          Max.   :49.00
 visitorFinalScore
 Min.   : 0.00
 1st Qu.:14.75
 Median :20.00
 Mean   :20.95
 3rd Qu.:27.00
 Max.   :48.00
```

Summary of Player-Plays dataset:

```
     gameId              playId         nflId         teamAbbr
 Min.   :2.022e+09   Min.   :  54   Min.   :25511   Length:354727
 1st Qu.:2.022e+09   1st Qu.: 996   1st Qu.:43426   Class :character
 Median :2.022e+09   Median :2017   Median :46457   Mode  :character
 Mean   :2.022e+09   Mean   :2024   Mean   :47437
 3rd Qu.:2.022e+09   3rd Qu.:3022   3rd Qu.:52590
 Max.   :2.022e+09   Max.   :5120   Max.   :55241

 hadRushAttempt     rushingYards       hadDropback       passingYards
 Min.   :0.00000   Min.   :-10.0000   Min.   :0.00000   Min.   :-10.0000
 1st Qu.:0.00000   1st Qu.:  0.0000   1st Qu.:0.00000   1st Qu.:  0.0000
 Median :0.00000   Median :  0.0000   Median :0.00000   Median :  0.0000
 Mean   :0.01914   Mean   :  0.0873   Mean   :0.01757   Mean   :  0.1733
```

```
3rd Qu.:0.00000   3rd Qu.:  0.0000   3rd Qu.:0.00000   3rd Qu.:  0.0000
Max.   :1.00000   Max.   : 75.0000   Max.   :1.00000   Max.   : 98.0000

sackYardsAsOffense  hadPassReception  receivingYards     wasTargettedReceiver
Min.   :-18.00000   Min.   :0.00000   Min.   :-11.0000   Min.   :0.0000
1st Qu.:  0.00000   1st Qu.:0.00000   1st Qu.:  0.0000   1st Qu.:0.0000
Median :  0.00000   Median :0.00000   Median :  0.0000   Median :0.0000
Mean   : -0.01147   Mean   :0.01586   Mean   :  0.1734   Mean   :0.0236
3rd Qu.:  0.00000   3rd Qu.:0.00000   3rd Qu.:  0.0000   3rd Qu.:0.0000
Max.   :  0.00000   Max.   :1.00000   Max.   : 98.0000   Max.   :1.0000

yardageGainedAfterTheCatch    fumbles            fumbleLost
Min.   :-7.00000           Min.   :0.0000000   Min.   :0.0000000
1st Qu.: 0.00000           1st Qu.:0.0000000   1st Qu.:0.0000000
Median : 0.00000           Median :0.0000000   Median :0.0000000
Mean   : 0.08282           Mean   :0.0007611   Mean   :0.0003186
3rd Qu.: 0.00000           3rd Qu.:0.0000000   3rd Qu.:0.0000000
Max.   :75.00000           Max.   :2.0000000   Max.   :1.0000000

fumbleOutOfBounds  assistedTackle     forcedFumbleAsDefense
Min.   :0.00e+00   Min.   :0.000000   Min.   :0.0000000
1st Qu.:0.00e+00   1st Qu.:0.000000   1st Qu.:0.0000000
Median :0.00e+00   Median :0.000000   Median :0.0000000
Mean   :5.36e-05   Mean   :0.004138   Mean   :0.0005131
3rd Qu.:0.00e+00   3rd Qu.:0.000000   3rd Qu.:0.0000000
Max.   :1.00e+00   Max.   :1.000000   Max.   :1.0000000

halfSackYardsAsDefense  passDefensed      quarterbackHit
Min.   :-18.000000     Min.   :0.000000   Min.   :0.000000
1st Qu.:  0.000000     1st Qu.:0.000000   1st Qu.:0.000000
Median :  0.000000     Median :0.000000   Median :0.000000
Mean   : -0.002154     Mean   :0.003093   Mean   :0.003986
3rd Qu.:  0.000000     3rd Qu.:0.000000   3rd Qu.:0.000000
Max.   :  0.000000     Max.   :1.000000   Max.   :1.000000

sackYardsAsDefense  safetyAsDefense      soloTackle       tackleAssist
Min.   :-17.00000   Min.   :0.00e+00   Min.   :0.00000   Min.   :0.00000
1st Qu.:  0.00000   1st Qu.:0.00e+00   1st Qu.:0.00000   1st Qu.:0.00000
Median :  0.00000   Median :0.00e+00   Median :0.00000   Median :0.00000
Mean   : -0.01036   Mean   :1.13e-05   Mean   :0.02483   Mean   :0.01544
3rd Qu.:  0.00000   3rd Qu.:0.00e+00   3rd Qu.:0.00000   3rd Qu.:0.00000
Max.   :  0.00000   Max.   :1.00e+00   Max.   :1.00000   Max.   :1.00000
```

```
 tackleForALoss    tackleForALossYardage hadInterception     interceptionYards
 Min.   :0.00000   Min.   : 0.0000       Min.   :0.0000000   Min.   :-6.00000
 1st Qu.:0.00000   1st Qu.: 0.0000       1st Qu.:0.0000000   1st Qu.: 0.00000
 Median :0.00000   Median : 0.0000       Median :0.0000000   Median : 0.00000
 Mean   :0.00327   Mean   : 0.0128       Mean   :0.0005441   Mean   : 0.00708
 3rd Qu.:0.00000   3rd Qu.: 0.0000       3rd Qu.:0.0000000   3rd Qu.: 0.00000
 Max.   :1.00000   Max.   :17.0000       Max.   :1.0000000   Max.   :99.00000

 fumbleRecoveries    fumbleRecoveryYards  penaltyYards       penaltyNames
 Min.   :0.0000000   Min.   :-15.00000   Min.   : 0.000000   Length:354727
 1st Qu.:0.0000000   1st Qu.:  0.00000   1st Qu.: 0.000000   Class :character
 Median :0.0000000   Median :  0.00000   Median : 0.000000   Mode  :character
 Mean   :0.0007076   Mean   :  0.00127   Mean   : 0.006614
 3rd Qu.:0.0000000   3rd Qu.:  0.00000   3rd Qu.: 0.000000
 Max.   :2.0000000   Max.   : 68.00000   Max.   :20.000000

 wasInitialPassRusher causedPressure  timeToPressureAsPassRusher
 Min.   :0.00         Mode :logical   Min.   : 0.8
 1st Qu.:0.00         FALSE:350420    1st Qu.: 2.2
 Median :0.00         TRUE :4307      Median : 2.7
 Mean   :0.39                         Mean   : 2.9
 3rd Qu.:1.00                         3rd Qu.: 3.2
 Max.   :1.00                         Max.   :11.6
 NA's   :247447                       NA's   :350399
 getOffTimeAsPassRusher inMotionAtBallSnap shiftSinceLineset
motionSinceLineset
 Min.   :0.00           Mode :logical      Mode :logical     Mode :logical
 1st Qu.:0.80           FALSE:103276       FALSE:172421      FALSE:84416
 Median :0.96           TRUE :4572         TRUE :3757        TRUE :5822
 Mean   :1.01           NA's :246879       NA's :178549      NA's :264489
 3rd Qu.:1.17
 Max.   :2.00
 NA's   :306695
 wasRunningRoute    routeRan         blockedPlayerNFLId1 blockedPlayerNFLId2
 Min.   :1        Length:354727      Min.   :33131       Min.   :35454
 1st Qu.:1        Class :character   1st Qu.:43316       1st Qu.:46204
 Median :1        Mode  :character   Median :46141       Median :48198
 Mean   :1                           Mean   :46504       Mean   :49704
 3rd Qu.:1                           3rd Qu.:52448       3rd Qu.:53501
 Max.   :1                           Max.   :55241       Max.   :55239
 NA's   :311948                      NA's   :305623      NA's   :350354
```

```
 blockedPlayerNFLId3 pressureAllowedAsBlocker timeToPressureAllowedAsBlocker
 Min.   :46269       Min.   :0.00             Min.   : 0.8
 1st Qu.:47786       1st Qu.:0.00             1st Qu.: 2.3
 Median :47944       Median :0.00             Median : 2.7
 Mean   :49423       Mean   :0.08             Mean   : 2.9
 3rd Qu.:50722       3rd Qu.:0.00             3rd Qu.: 3.3
 Max.   :54733       Max.   :1.00             Max.   :11.6
 NA's   :354720      NA's   :301683           NA's   :350647
 pff_defensiveCoverageAssignment pff_primaryDefensiveCoverageMatchupNflId
 Length:354727                   Min.   :29550
 Class :character                1st Qu.:44841
 Mode  :character                Median :47791
                                 Mean   :47938
                                 3rd Qu.:52608
                                 Max.   :55168
                                 NA's   :311243
 pff_secondaryDefensiveCoverageMatchupNflId
 Min.   :30842
 1st Qu.:44860
 Median :46705
 Mean   :47983
 3rd Qu.:52644
 Max.   :55157
 NA's   :352340
```

Summary of Players dataset:

```
     nflId          height              weight       birthDate
 Min.   :25511   Length:1697        Min.   :153.0   Length:1697
 1st Qu.:44830   Class :character   1st Qu.:205.0   Class :character
 Median :47874   Mode  :character   Median :236.0   Mode  :character
 Mean   :48237                      Mean   :245.8
 3rd Qu.:53476                      3rd Qu.:291.0
 Max.   :55241                      Max.   :380.0
 collegeName          position         displayName
 Length:1697        Length:1697        Length:1697
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

Summary of Week 1 dataset:

```
     gameId              playId         nflId        displayName
 Min.   :2.022e+09   Min.   :  55   Min.   :25511    Length:7104700
 1st Qu.:2.022e+09   1st Qu.: 955   1st Qu.:43384    Class :character
 Median :2.022e+09   Median :1995   Median :46214    Mode  :character
 Mean   :2.022e+09   Mean   :2024   Mean   :47186
 3rd Qu.:2.022e+09   3rd Qu.:3043   3rd Qu.:52498
 Max.   :2.022e+09   Max.   :5120   Max.   :55173
                                    NA's   :308900
    frameId        frameType             time            jerseyNumber
 Min.   :  1.00   Length:7104700     Length:7104700     Min.   : 1.00
 1st Qu.: 40.00   Class :character   Class :character   1st Qu.:21.00
 Median : 81.00   Mode  :character   Mode  :character   Median :51.00
 Mean   : 86.93                                        Mean   :48.09
 3rd Qu.:126.00                                        3rd Qu.:75.00
 Max.   :697.00                                        Max.   :99.00
                                                       NA's   :308900
    club           playDirection            x                y
 Length:7104700     Length:7104700     Min.   : -5.06   Min.   :-8.94
 Class :character   Class :character   1st Qu.: 41.10   1st Qu.:22.44
 Mode  :character   Mode  :character   Median : 61.14   Median :26.82
                                       Mean   : 60.88   Mean   :26.83
                                       3rd Qu.: 80.26   3rd Qu.:31.19
                                       Max.   :125.60   Max.   :69.47

      s                a                dis              o
 Min.   : 0.000   Min.   : 0.0000   Min.   :0.000   Min.   :  0.00
 1st Qu.: 0.060   1st Qu.: 0.0600   1st Qu.:0.010   1st Qu.: 89.33
 Median : 0.560   Median : 0.5300   Median :0.060   Median :177.32
 Mean   : 1.359   Mean   : 0.9441   Mean   :0.139   Mean   :179.17
 3rd Qu.: 1.960   3rd Qu.: 1.3500   3rd Qu.:0.200   3rd Qu.:269.22
 Max.   :29.140   Max.   :56.5800   Max.   :7.630   Max.   :360.00
                                                    NA's   :308511
     dir            event
 Min.   :  0.00   Length:7104700
 1st Qu.: 89.66   Class :character
 Median :179.65   Mode  :character
 Mean   :179.88
 3rd Qu.:270.35
 Max.   :360.00
 NA's   :308511
```

General Characteristics:

- Data contains records of 136 games from the 2022 season, spanning weeks 1 to 9.
- Scores range from 0 to 49 for both home and visitor teams.

Home vs Visitor Scores:

- Home teams: Average score of 22.67, median 22.5, with a maximum of 49.
- Visitor teams: Average score of 20.95, median 20.0, with a maximum of 48.
- Overall, home teams tend to score slightly higher than visitors.

Player-Plays Dataset

A large dataset with 354,727 rows, capturing detailed player-level statistics per play.

Key Indicators:

- Rushing yards range from -10 to 75; average is low (0.087), indicating most players did not rush.
- Passing yards range from -10 to 98; average is slightly higher (0.173).
- Receiving yards reach a maximum of 98, with a similar average of 0.173.

Defensive Highlights:

- Low rates of tackles, interceptions, sacks, and fumbles—indicating many plays without significant defensive impact.
- Mean sack yards as defense and tackles for a loss yardage are both near zero, though there are rare events with large values.

Rare Events:

Events like forced fumbles (0.05%) and safeties (0.001%) are extremely rare.

Players Dataset

- Contains 1,697 players, with attributes such as height, weight, birth date, and college name.

- Players' weights range from 153 to 380 lbs, with an average of 245.8 lbs. Positions likely influence this variation.

General Observations:

- Offense Dominates: Most plays revolve around offensive stats (rushing, passing, and receiving) with fewer defensive actions recorded.
- High Sparsity: Many binary and numeric fields have zeros or NA values, indicating sparse events (e.g., fumbles, interceptions, or blocked players).

Missing Values?

The tables showed multiple missing values. This could hinder the rendering data visualizations.

```
calculate_missing_values <- function(df) {
  # Calculate the number of missing values for each column
  missing_counts <- colSums(is.na(df))

  # Create a data frame with the results
  missing_df <- data.frame(
    Column = names(missing_counts),
    Missing_Count = missing_counts,
    Missing_Percentage = round((missing_counts / nrow(df)) * 100, 2)
  )

  # Return the data frame
  return(missing_df)
}

# Calculate missing values for each dataset
missing_games <- calculate_missing_values(games)
missing_player_plays <- calculate_missing_values(player_plays)
missing_players <- calculate_missing_values(players)
missing_week1 <- calculate_missing_values(week1)

# Print the results without repeating column names
cat("Missing values in Games.csv:\n")
```

```
print(missing_games, row.names = FALSE)  # Suppress row names

cat("\nMissing values in Player-Plays.csv:\n")
print(missing_player_plays, row.names = FALSE)

cat("\nMissing values in Players.csv:\n")
print(missing_players, row.names = FALSE)

cat("\nMissing values in Week1.csv:\n")
print(missing_week1, row.names = FALSE)
```

Output

Missing values in Games.csv:

Column	Missing_Count	Missing_Percentage
gameId	0	0
season	0	0
week	0	0
gameDate	0	0
gameTimeEastern	0	0
homeTeamAbbr	0	0
visitorTeamAbbr	0	0
homeFinalScore	0	0
visitorFinalScore	0	0

Missing values in Player-Plays.csv:

Column	Missing_Count	Missing_Percentage
gameId	0	0.00
playId	0	0.00
nflId	0	0.00
teamAbbr	0	0.00
hadRushAttempt	0	0.00
rushingYards	0	0.00
hadDropback	0	0.00
passingYards	0	0.00
sackYardsAsOffense	0	0.00
hadPassReception	0	0.00
receivingYards	0	0.00
wasTargettedReceiver	0	0.00
yardageGainedAfterTheCatch	0	0.00
fumbles	0	0.00
fumbleLost	0	0.00
fumbleOutOfBounds	0	0.00
assistedTackle	0	0.00
forcedFumbleAsDefense	0	0.00
halfSackYardsAsDefense	0	0.00
passDefensed	0	0.00

quarterbackHit	0	0.00
sackYardsAsDefense	0	0.00
safetyAsDefense	0	0.00
soloTackle	0	0.00
tackleAssist	0	0.00
tackleForALoss	0	0.00
tackleForALossYardage	0	0.00
hadInterception	0	0.00
interceptionYards	0	0.00
fumbleRecoveries	0	0.00
fumbleRecoveryYards	0	0.00
penaltyYards	0	0.00
penaltyNames	354351	99.89
wasInitialPassRusher	247447	69.76
causedPressure	0	0.00
timeToPressureAsPassRusher	350399	98.78
getOffTimeAsPassRusher	306695	86.46
inMotionAtBallSnap	246879	69.60
shiftSinceLineset	178549	50.33
motionSinceLineset	264489	74.56
wasRunningRoute	311948	87.94
routeRan	311948	87.94
blockedPlayerNFLId1	305623	86.16
blockedPlayerNFLId2	350354	98.77
blockedPlayerNFLId3	354720	100.00
pressureAllowedAsBlocker	301683	85.05
timeToPressureAllowedAsBlocker	350647	98.85
pff_defensiveCoverageAssignment	288953	81.46
pff_primaryDefensiveCoverageMatchupNflId	311243	87.74
pff_secondaryDefensiveCoverageMatchupNflId	352340	99.33

Missing values in Players.csv:

Column	Missing_Count	Missing_Percentage
nflId	0	0.0
height	0	0.0
weight	0	0.0
birthDate	487	28.7
collegeName	0	0.0
position	0	0.0
displayName	0	0.0

Missing values in Week1.csv:

Column	Missing_Count	Missing_Percentage
gameId	0	0.00
playId	0	0.00
nflId	308900	4.35
displayName	0	0.00
frameId	0	0.00
frameType	0	0.00
time	0	0.00
jerseyNumber	308900	4.35
club	0	0.00

```
 playDirection             0              0.00
             x             0              0.00
             y             0              0.00
             s             0              0.00
             a             0              0.00
           dis             0              0.00
             o        308511              4.34
           dir        308511              4.34
         event       6795189             95.64
```

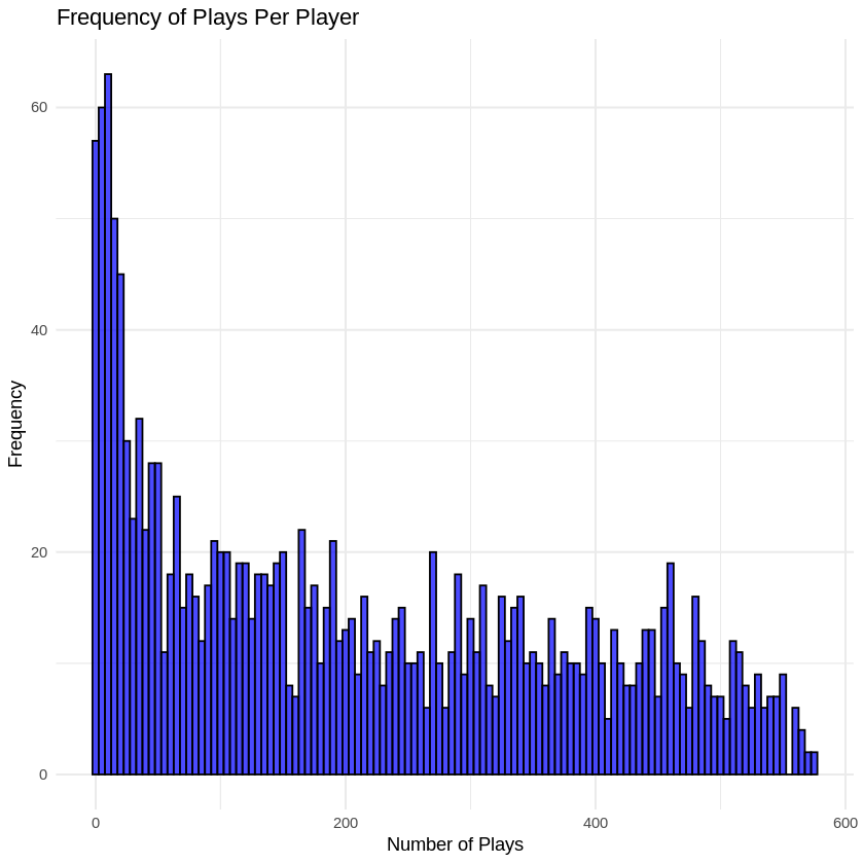
We find that some columns have almost no values. Columns such as blockedPlayerNFLId3, penaltynames have 99.9% missing value rate. Games.csv is doesn't have any NA values. The player's data has the column “birthDate” with 28.7% missing values.

Frequency Visualization

```
# Frequency of plays per player
play_count <- player_plays %>%
  group_by(nflId) %>%
  summarize(Play_Count = n())

# Plot frequency of plays per player
ggplot(play_count, aes(x = Play_Count)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black", alpha =
0.7) +
  theme_minimal() +
  labs(
    title = "Frequency of Plays Per Player",
    x = "Number of Plays",
    y = "Frequency"
  )
```

Output:-



Key Statistics per Dataset

```
# Example: Player-Plays dataset
player_stats <- player_plays %>%
  summarise(
    Mean_Rushing_Yards = mean(rushingYards, na.rm = TRUE),
    Median_Rushing_Yards = median(rushingYards, na.rm = TRUE),
    SD_Rushing_Yards = sd(rushingYards, na.rm = TRUE),
    Mode_Rushing_Yards = as.numeric(names(sort(table(rushingYards),
decreasing = TRUE)[1]))
  )

cat("\nKey Statistics for Player-Plays dataset:\n")
print(player_stats)
```

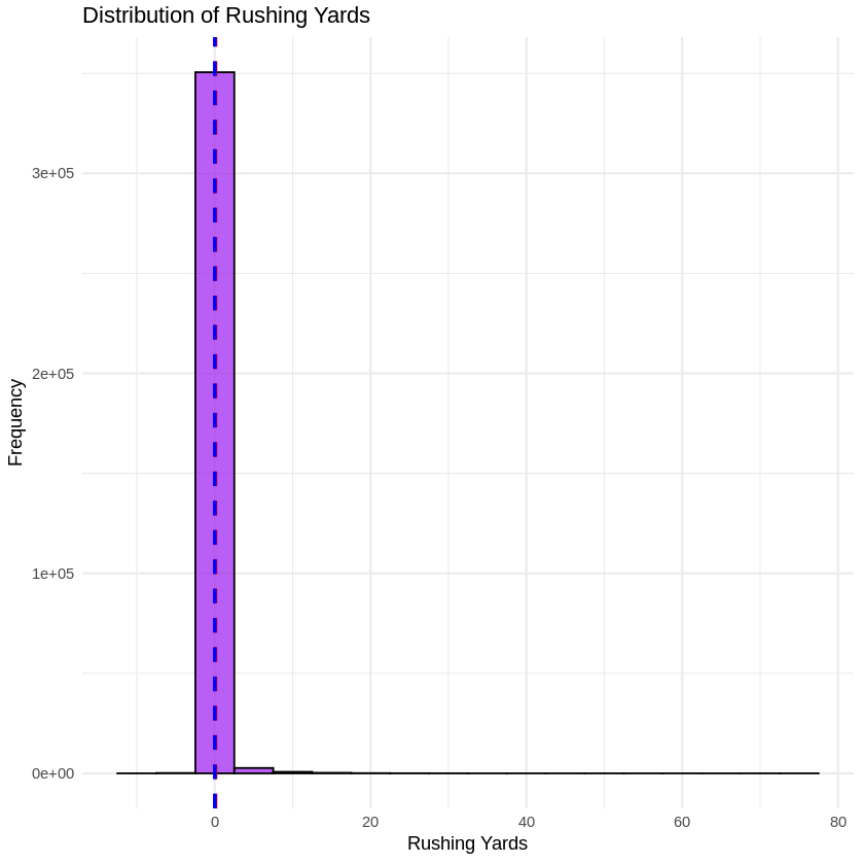
Output:-

```
Key Statistics for Player-Plays dataset:
  Mean_Rushing_Yards Median_Rushing_Yards SD_Rushing_Yards
Mode_Rushing_Yards
1         0.08730376                   0         1.098183
0
```

Rushing Yards Distribution

```
if("rushingYards" %in% colnames(player_plays)) {
  ggplot(player_plays, aes(x = rushingYards)) +
    geom_histogram(binwidth = 5, fill = "purple", color = "black",
alpha = 0.7) +
    theme_minimal() +
    labs(
      title = "Distribution of Rushing Yards",
      x = "Rushing Yards",
      y = "Frequency"
    ) +
    geom_vline(aes(xintercept = mean(rushingYards, na.rm = TRUE)),
color = "red", linetype = "dashed", linewidth = 1) +
    geom_vline(aes(xintercept = median(rushingYards, na.rm = TRUE)),
color = "blue", linetype = "dashed", linewidth = 1)
} else {
  cat("Column 'rushingYards' not found in player_plays dataset.\n")
}
```

Output:-

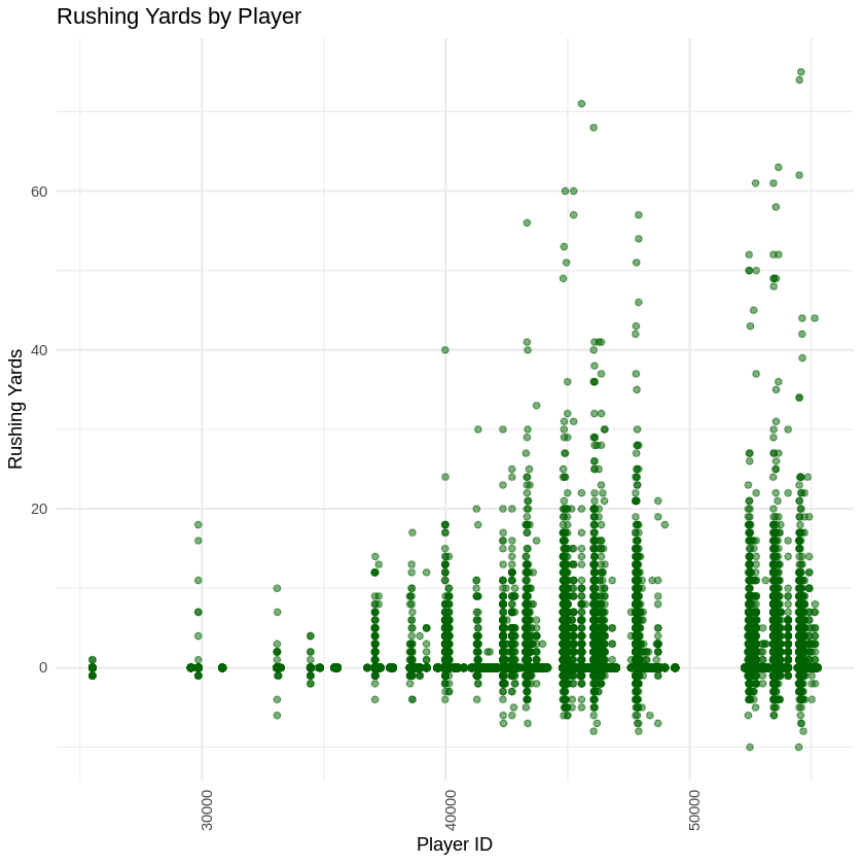


Player Statistics Scatter Plot

```
if("rushingYards" %in% colnames(player_plays) && "nflId" %in%
colnames(player_plays)) {
  ggplot(player_plays, aes(x = nflId, y = rushingYards)) +
    geom_point(color = "darkgreen", alpha = 0.5) +
    theme_minimal() +
    labs(
      title = "Rushing Yards by Player",
      x = "Player ID",
```

```
      y = "Rushing Yards"
    ) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
} else {
  cat("Column 'rushingYards' or 'nflId' not found in player_plays
dataset.\n")
}
```

Output:-



Save Visualizations

```
# Save histogram of play counts
ggsave("Play_Count_Histogram.png")

# Save distribution of rushing yards (if the column exists)
```

```
if("rushingYards" %in% colnames(player_plays)) {
  ggsave("Rushing_Yards_Distribution.png")
}
```

Classification

Grouping by teams and calculating the average scores. Calculate the average and total scores for each team type (home and visitor) across games to get a sense of team scoring tendencies.
Features Created:

- Average home and visitor scores
- Total home and visitor scores

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# `games_df` is already loaded from Games.csv

# Group by team type to calculate average and total scores across
games
team_scores = games_df.groupby(['homeTeamAbbr',
'visitorTeamAbbr']).agg(
    avg_home_score=('homeFinalScore', 'mean'),
    avg_visitor_score=('visitorFinalScore', 'mean'),
    total_home_score=('homeFinalScore', 'sum'),
    total_visitor_score=('visitorFinalScore', 'sum')
).reset_index()

# Displaying the results for inspection
print(team_scores.head())
```

```
homeTeamAbbr visitorTeamAbbr  avg_home_score  avg_visitor_score  \
0          ARI              KC            21.0               44.0
1          ARI              LA            12.0               20.0
2          ARI              NO            42.0               34.0
3          ARI             PHI            17.0               20.0
4          ARI             SEA            21.0               31.0

   total_home_score  total_visitor_score
0                21                   44
1                12                   20
2                42                   34
3                17                   20
4                21                   31
```

I also tried to make a bar visualization for this. We can break down the home and visitor average scores and compare them side-by-side for each team. This way, we'll easily see how scoring differs between home and visitor scenarios. The data is grouped by team visualizations

Step 1: Restructure the data for visualization

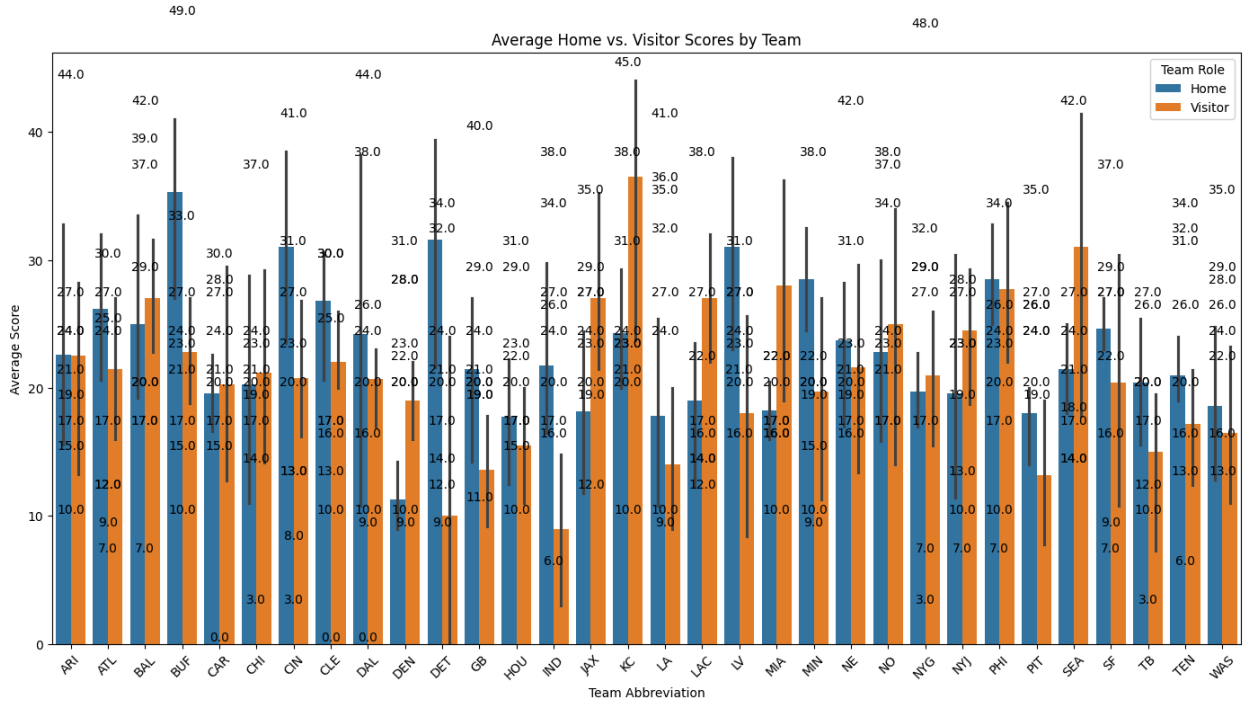
```
team_scores_melted = pd.melt(
    team_scores,
    id_vars=['homeTeamAbbr', 'visitorTeamAbbr'],
    value_vars=['avg_home_score', 'avg_visitor_score'],
    var_name='Score Type',
    value_name='Average Score'
)

# Step 2: Map score types for clarity in labeling
team_scores_melted['Team Role'] = team_scores_melted['Score
Type'].apply(lambda x: 'Home' if 'home' in x else 'Visitor')
team_scores_melted['Team Abbr'] = np.where(team_scores_melted['Team
Role'] == 'Home', team_scores_melted['homeTeamAbbr'],
team_scores_melted['visitorTeamAbbr'])

# Step 3: Plotting
plt.figure(figsize=(14, 8))
sns.barplot(data=team_scores_melted, x='Team Abbr', y='Average
Score', hue='Team Role')
plt.title("Average Home vs. Visitor Scores by Team")
plt.xlabel("Team Abbreviation")
plt.ylabel("Average Score")
plt.legend(title="Team Role", loc="upper right")

# Annotate to show values
for i, row in team_scores_melted.iterrows():
    plt.text(
        i % len(team_scores_melted['Team Abbr'].unique()),
        row['Average Score'] + 0.2,
        f"{row['Average Score']:.1f}",
        ha='center',
        color='black'
    )

plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Modeling by focusing first on basic statistical features, which will give us an initial sense of player and play patterns. **Average yards gained per play** for each player, broken down by rushing and passing plays. This can help us assess performance differences across players and between different types of plays. I'll guide you through the code and add some visualizations as well.

```
# Imports

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Calculate average yards gained per play (rushing and passing)

avg_yards_gained = player_plays.groupby(['nflId']).agg({

    'rushingYards': 'mean',

    'passingYards': 'mean'

}).reset_index()
```

```
# Rename columns for clarity

avg_yards_gained.columns = ['Player ID', 'Avg Rushing Yards', 'Avg
Passing Yards']

# Display top 10 players in each category

top_rushers = avg_yards_gained.nlargest(10, 'Avg Rushing Yards')

top_passers = avg_yards_gained.nlargest(10, 'Avg Passing Yards')

# Plotting top rushers and passers

plt.figure(figsize=(16, 8))

# Top Rushers

plt.subplot(1, 2, 1)

sns.barplot(data=top_rushers, x='Avg Rushing Yards', y='Player ID',
palette="Blues_d")

plt.title('Top 10 Players by Average Rushing Yards')

plt.xlabel('Average Rushing Yards')

plt.ylabel('Player ID')

# Annotate bars with values

for index, value in enumerate(top_rushers['Avg Rushing Yards']):

    plt.text(value, index, f"{value:.2f}", color='black',
va="center")

# Top Passers

plt.subplot(1, 2, 2)

sns.barplot(data=top_passers, x='Avg Passing Yards', y='Player ID',
palette="Greens_d")

plt.title('Top 10 Players by Average Passing Yards')

plt.xlabel('Average Passing Yards')

plt.ylabel('Player ID')
```

```
# Annotate bars with values

for index, value in enumerate(top_passers['Avg Passing Yards']):

    plt.text(value, index, f"{value:.2f}", color='black',
va="center")


plt.tight_layout()

plt.show()
```