DATA 607 - Final Project

Julian Adames-Ng

2024-12-11

1. Introduction

Initial Project Proposal:

For my project, I aim to analyze NFL statistics to uncover insights into player performance and game outcomes. The motivation stems from the growing interest in data-driven sports analytics and its impact on player evaluation and fan engagement. I also have had growing interest in NFL data as someone who was not into sports growing up. My data will be sourced from public NFL datasets, scraped directly from websites such as ESPN.com, The Football Database, and directly from NFL.com. I will try to find CSV files and other sources of data. The workflow will involve data acquisition, cleaning, and transformation (e.g., converting wide to long formats for analysis), followed by statistical analysis and visualization to highlight trends and validate my conclusions. I also plan to use topics not covered in class, such as predictive modeling for player performance.

Data Sources:

(1) NFL Team and Weather Stats: I downloaded a CSV file with NFL statistics dating back to 1960 on the following website:

https://nflsavant.com/about.php.

The raw data can be found on my Github in the following url:

https://raw.githubusercontent.com/JAdames27/DATA-607---Data-Acquisition-and-Management/refs/heads/main/DATA%20607%20-%20Final%20Project/weather_20131231.csv

(2) NFL Player Data Obtained from the nflreadr package in R

Loading the Data

The following code clears the R environment and loads essential libraries like nflreadr and tidyverse for data handling. It then imports weather and player data from CSV files into separate dataframes. Finally, it displays the structure of both datasets to examine their contents and variable types.

```r
rm(list = ls())

# Load libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
# Load the data
weather_data <- read.csv("https://raw.githubusercontent.com/JAdames27/DATA-607---Data-Acquisition-and-M

# Preview the datasets
head(weather_data)
```

```
##              id          home_team home_score           away_team away_score
## 1 196009230ram    Los Angeles Rams         21 St. Louis Cardinals         43
## 2 196009240dal      Dallas Cowboys         28 Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14       Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19     New York Giants         21
## 5 196009250clt     Baltimore Colts         20 Washington Redskins          0
## 6 196009250phi Philadelphia Eagles         24    Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                           weather      date
## 1  66 degrees- relative humidity 78%- wind 8 mph 9/23/1960
## 2 72 degrees- relative humidity 80%- wind 16 mph 9/24/1960
## 3 60 degrees- relative humidity 76%- wind 13 mph 9/25/1960
## 4 72 degrees- relative humidity 44%- wind 10 mph 9/25/1960
## 5  62 degrees- relative humidity 80%- wind 9 mph 9/25/1960
## 6  61 degrees- relative humidity 77%- wind 9 mph 9/25/1960
```

```
# Check structure
str(weather_data)
```

```
## 'data.frame':    11192 obs. of  11 variables:
##  $ id         : chr  "196009230ram" "196009240dal" "196009250gnb" "196009250sfo" ...
##  $ home_team  : chr  "Los Angeles Rams" "Dallas Cowboys" "Green Bay Packers" "San Francisco 49ers" .
##  $ home_score : int  21 28 14 19 20 24 25 28 28 42 ...
##  $ away_team  : chr  "St. Louis Cardinals" "Pittsburgh Steelers" "Chicago Bears" "New York Giants" .
##  $ away_score : int  43 35 17 21 0 41 27 9 20 7 ...
##  $ temperature: int  66 72 60 72 62 61 77 53 54 54 ...
##  $ wind_chill : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ humidity   : chr  "78%" "80%" "76%" "44%" ...
##  $ wind_mph   : int  8 16 13 10 9 9 11 16 15 9 ...
##  $ weather    : chr  "66 degrees- relative humidity 78%- wind 8 mph" "72 degrees- relative humidity 8
##  $ date       : chr  "9/23/1960" "9/24/1960" "9/25/1960" "9/25/1960" ...
```

Creating a Winner Column

A new column, winner, is created to indicate which team won each game by comparing home and away scores. If the game was a tie, it is assigned "Tie" in the column. Rows with ties are then filtered out to focus only on games with clear winners.

```r
# Create a new column for the winner
weather_data$winner <- ifelse(
  weather_data$home_score > weather_data$away_score,
  weather_data$home_team,
  ifelse(weather_data$away_score > weather_data$home_score,
         weather_data$away_team,
         "Tie")
)

weather_data <- weather_data %>%
  filter(winner != "Tie")

# View the updated dataframe
head(weather_data)
```

```
##              id           home_team home_score           away_team away_score
## 1 196009230ram    Los Angeles Rams         21 St. Louis Cardinals         43
## 2 196009240dal      Dallas Cowboys         28 Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14       Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19     New York Giants         21
## 5 196009250clt     Baltimore Colts         20 Washington Redskins          0
## 6 196009250phi Philadelphia Eagles         24    Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                           weather      date              winner
## 1  66 degrees- relative humidity 78%- wind 8 mph 9/23/1960 St. Louis Cardinals
## 2 72 degrees- relative humidity 80%- wind 16 mph 9/24/1960 Pittsburgh Steelers
## 3 60 degrees- relative humidity 76%- wind 13 mph 9/25/1960       Chicago Bears
## 4 72 degrees- relative humidity 44%- wind 10 mph 9/25/1960     New York Giants
## 5  62 degrees- relative humidity 80%- wind 9 mph 9/25/1960     Baltimore Colts
## 6  61 degrees- relative humidity 77%- wind 9 mph 9/25/1960    Cleveland Browns
```

Mapping Outdated Team Names

This chunk defines a mapping of old team names to their current names for standardization. It then updates the home_team and away_team columns in the dataset using the mapping. This ensures that all references to teams are consistent across the dataset.

```r
# Mapping historical team names to current NFL names
team_name_mapping <- c(
  # Arizona Cardinals
  "St. Louis Cardinals" = "Arizona Cardinals",
  "Phoenix Cardinals" = "Arizona Cardinals",

  # Las Vegas Raiders
  "Oakland Raiders" = "Las Vegas Raiders",
  "Los Angeles Raiders" = "Las Vegas Raiders",

  # Los Angeles Chargers
  "San Diego Chargers" = "Los Angeles Chargers",

  # Los Angeles Rams
  "St. Louis Rams" = "Los Angeles Rams",

  # Tennessee Titans
  "Houston Oilers" = "Tennessee Titans",
  "Tennessee Oilers" = "Tennessee Titans",

  # New England Patriots
  "Boston Patriots" = "New England Patriots",

  # Washington Commanders
  "Washington Redskins" = "Washington Commanders",
  "Washington Football Team" = "Washington Commanders",

  # Baltimore Ravens - Not Accounted for simplicity's sake
  # "Cleveland Browns" = "Baltimore Ravens", # Only pre-1996 Browns

  # Indianapolis Colts
  "Baltimore Colts" = "Indianapolis Colts" # Pre-1984 Colts
)

head(weather_data)
```

```
##             id           home_team home_score           away_team away_score
## 1 196009230ram    Los Angeles Rams         21 St. Louis Cardinals         43
## 2 196009240dal      Dallas Cowboys         28 Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14       Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19     New York Giants         21
## 5 196009250clt     Baltimore Colts         20 Washington Redskins          0
## 6 196009250phi Philadelphia Eagles         24    Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                           weather      date              winner
## 1  66 degrees- relative humidity 78%- wind 8 mph 9/23/1960 St. Louis Cardinals
## 2 72 degrees- relative humidity 80%- wind 16 mph 9/24/1960 Pittsburgh Steelers
## 3 60 degrees- relative humidity 76%- wind 13 mph 9/25/1960       Chicago Bears
## 4 72 degrees- relative humidity 44%- wind 10 mph 9/25/1960     New York Giants
## 5  62 degrees- relative humidity 80%- wind 9 mph 9/25/1960     Baltimore Colts
## 6  61 degrees- relative humidity 77%- wind 9 mph 9/25/1960    Cleveland Browns
```

```r
# Replace old team names in 'home_team' and 'away_team' columns
weather_data <- weather_data %>%
  mutate(
    home_team = recode(home_team, !!!team_name_mapping),
    away_team = recode(away_team, !!!team_name_mapping)
  )
```

```
head(weather_data)
```

```
##             id          home_team home_score             away_team away_score
## 1 196009230ram    Los Angeles Rams         21     Arizona Cardinals         43
## 2 196009240dal      Dallas Cowboys         28   Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14         Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19       New York Giants         21
## 5 196009250clt  Indianapolis Colts         20 Washington Commanders          0
## 6 196009250phi Philadelphia Eagles         24      Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                           weather      date              winner
## 1  66 degrees- relative humidity 78%- wind 8 mph 9/23/1960 St. Louis Cardinals
## 2 72 degrees- relative humidity 80%- wind 16 mph 9/24/1960 Pittsburgh Steelers
## 3 60 degrees- relative humidity 76%- wind 13 mph 9/25/1960       Chicago Bears
## 4 72 degrees- relative humidity 44%- wind 10 mph 9/25/1960     New York Giants
## 5  62 degrees- relative humidity 80%- wind 9 mph 9/25/1960     Baltimore Colts
## 6  61 degrees- relative humidity 77%- wind 9 mph 9/25/1960    Cleveland Browns
```

```
# Check the number of unique values in the column
num_unique_values <- length(unique(weather_data$home_team))

unique(weather_data$home_team)
```

```
##  [1] "Los Angeles Rams"      "Dallas Cowboys"        "Green Bay Packers"
##  [4] "San Francisco 49ers"   "Indianapolis Colts"    "Philadelphia Eagles"
##  [7] "Cleveland Browns"      "Arizona Cardinals"     "Detroit Lions"
## [10] "Pittsburgh Steelers"   "Washington Commanders" "Chicago Bears"
## [13] "New York Giants"       "Minnesota Vikings"     "Atlanta Falcons"
## [16] "New Orleans Saints"    "Kansas City Chiefs"    "Buffalo Bills"
## [19] "Los Angeles Chargers"  "Cincinnati Bengals"    "New England Patriots"
## [22] "Denver Broncos"        "Tennessee Titans"      "Miami Dolphins"
## [25] "New York Jets"         "Las Vegas Raiders"     "Seattle Seahawks"
## [28] "Tampa Bay Buccaneers"  "Jacksonville Jaguars"  "Carolina Panthers"
## [31] "Baltimore Ravens"      "Houston Texans"
```

```
num_unique_values
```

```
## [1] 32
```

Pairing Team to Home City

A lookup table is created to pair each team with their home location. The dataset is updated by joining this location data with the existing home_team information. This provides geographical context for each game based on where it was played.

```
# create location column based on home teams

# Lookup table with all NFL teams and their locations
team_locations <- data.frame(
  home_team = c(
```

```r
    "Arizona Cardinals", "Atlanta Falcons", "Baltimore Ravens", "Buffalo Bills",
    "Carolina Panthers", "Chicago Bears", "Cincinnati Bengals", "Cleveland Browns",
    "Dallas Cowboys", "Denver Broncos", "Detroit Lions", "Green Bay Packers",
    "Houston Texans", "Indianapolis Colts", "Jacksonville Jaguars", "Kansas City Chiefs",
    "Las Vegas Raiders", "Los Angeles Chargers", "Los Angeles Rams", "Miami Dolphins",
    "Minnesota Vikings", "New England Patriots", "New Orleans Saints", "New York Giants",
    "New York Jets", "Philadelphia Eagles", "Pittsburgh Steelers", "San Francisco 49ers",
    "Seattle Seahawks", "Tampa Bay Buccaneers", "Tennessee Titans", "Washington Commanders"
  ),
  location = c(
    "Glendale, AZ", "Atlanta, GA", "Baltimore, MD", "Buffalo, NY",
    "Charlotte, NC", "Chicago, IL", "Cincinnati, OH", "Cleveland, OH",
    "Dallas, TX", "Denver, CO", "Detroit, MI", "Green Bay, WI",
    "Houston, TX", "Indianapolis, IN", "Jacksonville, FL", "Kansas City, MO",
    "Las Vegas, NV", "Inglewood, CA", "Inglewood, CA", "Miami Gardens, FL",
    "Minneapolis, MN", "Foxborough, MA", "New Orleans, LA", "East Rutherford, NJ",
    "East Rutherford, NJ", "Philadelphia, PA", "Pittsburgh, PA", "Santa Clara, CA",
    "Seattle, WA", "Tampa, FL", "Nashville, TN", "Landover, MD"
  ),
  stringsAsFactors = FALSE
)

# Perform a join to add locations to the dataset
library(dplyr)
weather_data <- weather_data %>%
  left_join(team_locations, by = "home_team")

# Display the resulting dataframe
head(weather_data)
```

```
##              id           home_team home_score             away_team away_score
## 1 196009230ram    Los Angeles Rams         21     Arizona Cardinals         43
## 2 196009240dal      Dallas Cowboys         28   Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14         Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19       New York Giants         21
## 5 196009250clt  Indianapolis Colts         20 Washington Commanders          0
## 6 196009250phi Philadelphia Eagles         24      Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                           weather      date              winner
## 1  66 degrees- relative humidity 78%- wind 8 mph 9/23/1960 St. Louis Cardinals
## 2 72 degrees- relative humidity 80%- wind 16 mph 9/24/1960 Pittsburgh Steelers
## 3 60 degrees- relative humidity 76%- wind 13 mph 9/25/1960       Chicago Bears
## 4 72 degrees- relative humidity 44%- wind 10 mph 9/25/1960     New York Giants
## 5  62 degrees- relative humidity 80%- wind 9 mph 9/25/1960     Baltimore Colts
## 6  61 degrees- relative humidity 77%- wind 9 mph 9/25/1960    Cleveland Browns
##           location
## 1    Inglewood, CA
## 2       Dallas, TX
## 3    Green Bay, WI
## 4  Santa Clara, CA
## 5 Indianapolis, IN
## 6 Philadelphia, PA
```

```r
write_csv(weather_data, "weather_data.csv")
```

Seasonal Football Data Analysis

The following code organizes football game data by season, adjusting for games in January and February belonging to the previous year, and calculates season-level summaries including total games, average home score, and average away score.

```r
# Convert the game date column to a Date type
weather_data$date <- as.Date(weather_data$date, format = "%m/%d/%Y")

# Create a new column for the season
weather_data <- weather_data %>%
  mutate(
    season = ifelse(
      format(date, "%m") %in% c("01", "02"),
      as.numeric(format(date, "%Y")) - 1,
      as.numeric(format(date, "%Y"))
    )
  )

# Group the data by season
seasonal_totals <- weather_data %>%
  group_by(season) %>%
  summarize(
    total_games = n(),
    average_home_score = mean(home_score, na.rm = TRUE),
    average_away_score = mean(away_score, na.rm = TRUE)
  )

# Display the grouped data
head(seasonal_totals)
```

```
## # A tibble: 6 x 4
##   season total_games average_home_score average_away_score
##    <dbl>       <int>              <dbl>              <dbl>
## 1   1960          74               22.6               19.9
## 2   1961          96               23.1               20.1
## 3   1962          95               21.8               22.5
## 4   1963          94               22.8               21.4
## 5   1964          93               22.7               21.5
## 6   1965          98               22.7               23.1
```

Team Win Calculation by Season

This code processes game data to assign each game to a specific season, determine the winning team, and calculate the total wins for each team by season, excluding ties. It ensures all teams are included in the dataset, even those with no wins, and outputs a comprehensive summary of team performance across seasons.

```r
# Convert the game date column to a Date type
weather_data$date <- as.Date(weather_data$date, format = "%m/%d/%Y")
```

```r
# Create a new column for the season
weather_data <- weather_data %>%
  mutate(
    season = ifelse(
      format(date, "%m") %in% c("01", "02"),
      as.numeric(format(date, "%Y")) - 1,
      as.numeric(format(date, "%Y"))
    )
  )

# Determine the winning team for each game
weather_data <- weather_data %>%
  mutate(
    winning_team = case_when(
      home_score > away_score ~ home_team,
      home_score < away_score ~ away_team,
      TRUE ~ NA_character_  # For ties, NA is assigned
    )
  )

# Filter out games with ties or missing values in winning_team
weather_data <- weather_data %>%
  filter(!is.na(winning_team))

# Create a dataframe of all unique combinations of teams and seasons
all_teams_seasons <- weather_data %>%
  select(season, home_team, away_team) %>%
  pivot_longer(cols = c(home_team, away_team), names_to = "game_type", values_to = "team") %>%
  distinct(season, team)


# Group by season and team to calculate the number of wins
team_wins <- weather_data %>%
  group_by(season, winning_team) %>%
  summarize(
    total_wins = n(),
    .groups = "drop"
  ) %>%
  rename(team = winning_team)  # Rename for clarity

# Perform a full join with all_teams_seasons to include all combinations
team_wins <- all_teams_seasons %>%
  left_join(team_wins, by = c("season", "team")) %>%
  mutate(total_wins = replace_na(total_wins, 0))  # Replace NA values with 0

# Display the grouped data
head(team_wins)
```

```
## # A tibble: 6 x 3
##   season team                total_wins
##    <dbl> <chr>                    <int>
## 1   1960 Los Angeles Rams             4
## 2   1960 Arizona Cardinals            6
## 3   1960 Dallas Cowboys               0
## 4   1960 Pittsburgh Steelers          5
## 5   1960 Green Bay Packers            8
## 6   1960 Chicago Bears                5
```

Team Performance Metrics by Season

This code calculates the total games played by each team (both home and away) for every season, combines it with the total wins data, and computes each team's win rate. The results are organized by season and team, providing a comprehensive overview of team performance metrics.

```
# Calculate total games played for each team (home and away)
games_played <- weather_data %>%
  pivot_longer(
    cols = c(home_team, away_team),
    names_to = "game_type",
    values_to = "team"
  ) %>%
  group_by(season, team) %>%
  summarize(
    total_played = n(),
    .groups = "drop"
  )

#head(games_played)

team_totals <- team_wins %>%
  left_join(games_played, by = c("season", "team"))

team_totals$win_rate <- round(team_totals$total_wins / team_totals$total_played, digits = 3)

team_totals <- team_totals %>%
  arrange(season, team)

head(team_totals)
```

```
## # A tibble: 6 x 5
##   season team                total_wins total_played win_rate
##    <dbl> <chr>                    <int>        <int>    <dbl>
## 1   1960 Arizona Cardinals            6           11    0.545
## 2   1960 Chicago Bears                5           11    0.455
## 3   1960 Cleveland Browns             8           11    0.727
## 4   1960 Dallas Cowboys               0           11    0
## 5   1960 Detroit Lions                7           12    0.583
## 6   1960 Green Bay Packers            8           13    0.615
```

Game Statistics by Temperature Range

This code defines a function to filter games based on specified temperature and wind speed ranges, categorizing them into extreme weather games and ideal weather games (moderate conditions). The filtered datasets provide insights into game characteristics under varying weather conditions.

```
# Function to compute statistics by temperature range
compute_stats <- function(data, temp_filter) {
  # Filter data by temperature range
  filtered_data <- data %>%
    filter(temp_filter)
}

extreme_games <- compute_stats(weather_data, weather_data$temperature <= 32 | weather_data$temperature >
ideal_games <- compute_stats(weather_data, weather_data$temperature > 32 & weather_data$temperature < 8

# Display the first rows of each data frame
head(extreme_games)
```

```
##             id             home_team home_score           away_team away_score
## 1 196010090was Washington Commanders         26      Dallas Cowboys         14
## 2 196010230gnb     Green Bay Packers         41 San Francisco 49ers         14
## 3 196010230det         Detroit Lions         30  Indianapolis Colts         17
## 4 196010300was Washington Commanders         10    Cleveland Browns         31
## 5 196010300nyg       New York Giants         13   Arizona Cardinals         20
## 6 196010300dal        Dallas Cowboys          7  Indianapolis Colts         45
##   temperature wind_chill humidity wind_mph
## 1          59         NA      85%        1
## 2          49         NA      71%       21
## 3          53         NA      60%       22
## 4          54         NA      56%        3
## 5          54         NA      52%       27
## 6          65         NA      62%       23
##                                           weather       date              winner
## 1  59 degrees- relative humidity 85%- wind 1 mph 1960-10-09 Washington Redskins
## 2 49 degrees- relative humidity 71%- wind 21 mph 1960-10-23   Green Bay Packers
## 3 53 degrees- relative humidity 60%- wind 22 mph 1960-10-23       Detroit Lions
## 4  54 degrees- relative humidity 56%- wind 3 mph 1960-10-30    Cleveland Browns
## 5 54 degrees- relative humidity 52%- wind 27 mph 1960-10-30 St. Louis Cardinals
## 6 65 degrees- relative humidity 62%- wind 23 mph 1960-10-30     Baltimore Colts
##              location season          winning_team
## 1        Landover, MD   1960 Washington Commanders
## 2       Green Bay, WI   1960     Green Bay Packers
## 3         Detroit, MI   1960         Detroit Lions
## 4        Landover, MD   1960      Cleveland Browns
## 5 East Rutherford, NJ   1960     Arizona Cardinals
## 6          Dallas, TX   1960    Indianapolis Colts
```

```
head(ideal_games)
```

```
##             id           home_team home_score             away_team away_score
## 1 196009230ram    Los Angeles Rams         21     Arizona Cardinals         43
## 2 196009240dal      Dallas Cowboys         28   Pittsburgh Steelers         35
## 3 196009250gnb   Green Bay Packers         14         Chicago Bears         17
## 4 196009250sfo San Francisco 49ers         19       New York Giants         21
## 5 196009250clt  Indianapolis Colts         20 Washington Commanders          0
## 6 196009250phi Philadelphia Eagles         24      Cleveland Browns         41
##   temperature wind_chill humidity wind_mph
## 1          66         NA      78%        8
## 2          72         NA      80%       16
## 3          60         NA      76%       13
## 4          72         NA      44%       10
## 5          62         NA      80%        9
## 6          61         NA      77%        9
##                                          weather       date              winner
## 1  66 degrees- relative humidity 78%- wind 8 mph 1960-09-23 St. Louis Cardinals
## 2 72 degrees- relative humidity 80%- wind 16 mph 1960-09-24 Pittsburgh Steelers
## 3 60 degrees- relative humidity 76%- wind 13 mph 1960-09-25       Chicago Bears
## 4 72 degrees- relative humidity 44%- wind 10 mph 1960-09-25     New York Giants
## 5  62 degrees- relative humidity 80%- wind 9 mph 1960-09-25     Baltimore Colts
## 6  61 degrees- relative humidity 77%- wind 9 mph 1960-09-25    Cleveland Browns
##           location season        winning_team
## 1    Inglewood, CA   1960   Arizona Cardinals
## 2       Dallas, TX   1960 Pittsburgh Steelers
## 3    Green Bay, WI   1960       Chicago Bears
## 4  Santa Clara, CA   1960     New York Giants
## 5 Indianapolis, IN   1960  Indianapolis Colts
## 6 Philadelphia, PA   1960    Cleveland Browns
```

Team Performance in Extreme Weather Games

This code processes games played in extreme weather conditions to determine the winning team and calculate total wins for each team by season. It ensures that all teams, even those without wins, are included in the dataset and organizes the results by season and team for a clear overview of performance under extreme conditions.

```r
# Convert the game date column to a Date type
extreme_games$date <- as.Date(extreme_games$date, format = "%m/%d/%Y")

# Create a new column for the season
extreme_games <- extreme_games %>%
  mutate(
    season = ifelse(
      format(date, "%m") %in% c("01", "02"),
      as.numeric(format(date, "%Y")) - 1,
      as.numeric(format(date, "%Y"))
    )
  )

# Determine the winning team for each game
extreme_games <- extreme_games %>%
  mutate(
    winning_team = case_when(
      home_score > away_score ~ home_team,
      home_score < away_score ~ away_team,
      TRUE ~ NA_character_  # For ties, NA is assigned
    )
  )

# Filter out games with ties or missing values in winning_team
extreme_games <- extreme_games %>%
  filter(!is.na(winning_team))

# Create a dataframe of all unique combinations of teams and seasons
all_teams_extreme <- extreme_games %>%
  select(season, home_team, away_team) %>%
  pivot_longer(cols = c(home_team, away_team), names_to = "game_type", values_to = "team") %>%
  distinct(season, team)
```

```
unique_teams <- weather_data %>%
  select(season, home_team, away_team) %>%
  pivot_longer(cols = c(home_team, away_team), names_to = "game_type", values_to = "team") %>%
  distinct(season, team)

# Group by season and team to calculate the number of wins
extreme <- extreme_games %>%
  group_by(season, winning_team) %>%
  summarize(
    total_wins = n(),
    .groups = "drop"
  ) %>%
  rename(team = winning_team)  # Rename for clarity


# Perform a left join to map "team" and "total_wins" while assigning 0 to missing teams
extreme_wins <- unique_teams %>%
  left_join(extreme, by = c("season", "team")) %>%
  mutate(total_wins = replace_na(total_wins, 0))  # Replace NA with 0 for missing total_wins

extreme_wins <- extreme_wins %>%
   arrange(season, team)

head(extreme_wins)
```

```
## # A tibble: 6 x 3
##   season team              total_wins
##    <dbl> <chr>                   <int>
## 1   1960 Arizona Cardinals          1
## 2   1960 Chicago Bears              0
## 3   1960 Cleveland Browns           3
## 4   1960 Dallas Cowboys             0
## 5   1960 Detroit Lions              3
## 6   1960 Green Bay Packers          1
```

Team Performance in Ideal Weather Games

The code below processes games played under ideal weather conditions to determine the winning team and calculate total wins for each team by season, ensuring all teams are included, even those with no wins.

```
ideal_games2 <- ideal_games

# Convert the game date column to a Date type
ideal_games2$date <- as.Date(ideal_games2$date, format = "%m/%d/%Y")

# Create a new column for the season
ideal_games <- ideal_games2 %>%
  mutate(
    season = ifelse(
      format(date, "%m") %in% c("01", "02"),
      as.numeric(format(date, "%Y")) - 1,
      as.numeric(format(date, "%Y"))
    )
  )
```

```r
# Determine the winning team for each game
ideal_games <- ideal_games %>%
  mutate(
    winning_team = case_when(
      home_score > away_score ~ home_team,
      home_score < away_score ~ away_team,
      TRUE ~ NA_character_  # For ties, NA is assigned
    )
  )

# Filter out games with ties or missing values in winning_team
ideal_games <- ideal_games %>%
  filter(!is.na(winning_team))

# Create a dataframe of all unique combinations of teams and seasons
all_teams_ideal <- ideal_games %>%
  select(season, home_team, away_team) %>%
  pivot_longer(cols = c(home_team, away_team), names_to = "game_type", values_to = "team") %>%
  distinct(season, team)

# Group by season and team to calculate the number of wins
ideal_wins <- ideal_games %>%
  group_by(season, winning_team) %>%
  summarize(
    total_wins = n(),
    .groups = "drop"
  ) %>%
  rename(team = winning_team)

# Perform a full join with all_teams_seasons to include all combinations
ideal_wins <- all_teams_ideal %>%
  left_join(ideal_wins, by = c("season", "team")) %>%
  mutate(total_wins = replace_na(total_wins, 0))  # Replace NA values with 0

# Display the grouped data
ideal_wins <- ideal_wins %>%
  arrange(season, team)

head(extreme_wins)
```

```
## # A tibble: 6 x 3
##   season team              total_wins
##    <dbl> <chr>                  <int>
## 1   1960 Arizona Cardinals          1
## 2   1960 Chicago Bears              0
## 3   1960 Cleveland Browns           3
## 4   1960 Dallas Cowboys             0
## 5   1960 Detroit Lions              3
## 6   1960 Green Bay Packers          1
```

```r
head(ideal_wins)
```

```
## # A tibble: 6 x 3
##   season team              total_wins
##    <dbl> <chr>                     <int>
## 1   1960 Arizona Cardinals             5
## 2   1960 Chicago Bears                 5
## 3   1960 Cleveland Browns              5
## 4   1960 Dallas Cowboys                0
## 5   1960 Detroit Lions                 4
## 6   1960 Green Bay Packers             7
```

Performance Metrics

This code calculates the total games played by each team in extreme weather conditions and combines it with total wins to compute win rates for each season. Missing values for wins or games played are replaced with zeros, ensuring completeness, and the data is sorted by season and team for clarity. The result provides a comprehensive overview of team performance in extreme weather.

```r
# Calculate total games played for each team (home and away)
extreme_played <- extreme_games %>%
  pivot_longer(
    cols = c(home_team, away_team),
    names_to = "game_type",
    values_to = "team"
  ) %>%
  group_by(season, team) %>%
  summarize(
    total_played = n(),
    .groups = "drop"
  )

extreme_totals <- extreme_wins %>%
  left_join(extreme_played, by = c("season", "team"))

extreme_totals$win_rate <- round(extreme_totals$total_wins / extreme_totals$total_played, digits = 3)

extreme_totals <- extreme_totals %>%
  arrange(season, team)

extreme_totals <- extreme_totals %>%
  mutate(across(c(total_wins, total_played), ~ replace_na(., 0)))

head(extreme_totals)
```

```
## # A tibble: 6 x 5
##   season team              total_wins total_played win_rate
##    <dbl> <chr>                  <int>        <int>    <dbl>
## 1   1960 Arizona Cardinals          1            1      1
## 2   1960 Chicago Bears              0            2      0
## 3   1960 Cleveland Browns           3            3      1
## 4   1960 Dallas Cowboys             0            3      0
## 5   1960 Detroit Lions              3            3      1
## 6   1960 Green Bay Packers          1            2      0.5
```

Ideal Weather Team Performance Summary

The process calculates the total games played by each team under ideal weather conditions and combines it with win data to determine seasonal win rates. Missing values are replaced with zeros for completeness, and the results are sorted by season and team for clarity. The final output highlights team performance in favorable weather scenarios.

```
# Calculate total games played for each team (home and away)
ideal_played <- ideal_games %>%
  pivot_longer(
    cols = c(home_team, away_team),
    names_to = "game_type",
    values_to = "team"
  ) %>%
  group_by(season, team) %>%
  summarize(
    total_played = n(),
    .groups = "drop"
  )

ideal_totals <- ideal_wins %>%
  left_join(ideal_played, by = c("season", "team"))

ideal_totals$win_rate <- round(ideal_totals$total_wins / ideal_totals$total_played, digits = 3)

ideal_totals <- ideal_totals %>%
  arrange(season, team)

ideal_totals <- ideal_totals %>%
  mutate(across(c(total_wins, total_played), ~ replace_na(., 0)))

head(extreme_totals)
```

```
## # A tibble: 6 x 5
##   season team              total_wins total_played win_rate
##    <dbl> <chr>                  <int>        <int>    <dbl>
## 1   1960 Arizona Cardinals          1            1      1
## 2   1960 Chicago Bears              0            2      0
## 3   1960 Cleveland Browns           3            3      1
## 4   1960 Dallas Cowboys             0            3      0
## 5   1960 Detroit Lions              3            3      1
## 6   1960 Green Bay Packers          1            2      0.5
```

```
head(ideal_totals)
```

```
## # A tibble: 6 x 5
##   season team              total_wins total_played win_rate
##    <dbl> <chr>                  <int>        <int>    <dbl>
## 1   1960 Arizona Cardinals          5           10    0.5
## 2   1960 Chicago Bears              5            9    0.556
## 3   1960 Cleveland Browns           5            8    0.625
## 4   1960 Dallas Cowboys             0            8    0
## 5   1960 Detroit Lions              4            9    0.444
## 6   1960 Green Bay Packers          7           11    0.636
```

Average Win Rate Calculation Summary

This process calculates the average win rate for each team across all seasons using data from extreme weather games. Missing values are ignored during computation, and the results are saved to a CSV file for further analysis. The output provides a clear summary of team performance trends in extreme conditions.

```
# Calculate the average win rate for each team grouped by season
mean_rate <- extreme_totals  %>%
  group_by(team) %>%
  summarize(
    mean_win_rate = mean(win_rate, na.rm = TRUE),
    .groups = "drop"
  )

# View the resulting data
head(mean_rate)
```

```
## # A tibble: 6 x 2
##   team                 mean_win_rate
##   <chr>                        <dbl>
## 1 Arizona Cardinals            0.362
## 2 Atlanta Falcons              0.253
## 3 Baltimore Ravens             0.561
## 4 Buffalo Bills                0.457
## 5 Carolina Panthers            0.559
## 6 Chicago Bears                0.480
```

```
#write_csv(mean_rate, "mean_rate.csv")
```

Lifetime Performance in Extreme Weather Summary

This analysis aggregates each team’s total games played, total wins, and average win rate across all seasons in extreme weather conditions. It also calculates an all-time win rate by dividing total wins by total games played, providing a comprehensive overview of long-term performance under challenging conditions.

```
extreme_tot <- extreme_totals %>%
  group_by(team) %>%
  summarize(
    total_games = sum(total_played, na.rm = TRUE),
    total_wins = sum(total_wins, na.rm = TRUE),
    avg_win_rate = mean(win_rate, na.rm = TRUE)
  )

head(extreme_tot)
```

```
## # A tibble: 6 x 4
##   team                 total_games total_wins avg_win_rate
##   <chr>                      <int>      <int>        <dbl>
## 1 Arizona Cardinals            134         50        0.362
## 2 Atlanta Falcons               62         20        0.253
## 3 Baltimore Ravens              85         54        0.561
## 4 Buffalo Bills                139         63        0.457
## 5 Carolina Panthers             64         36        0.559
## 6 Chicago Bears                158         82        0.480
```

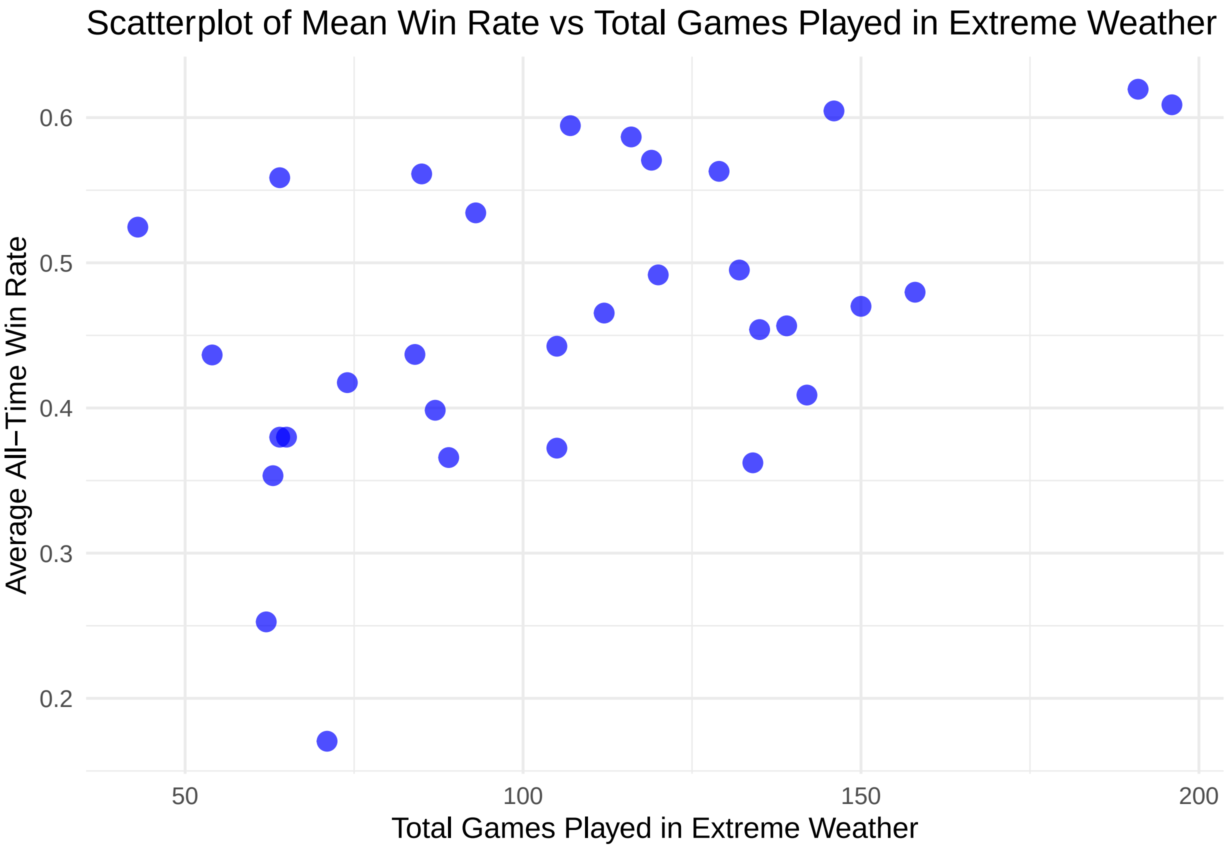
```
extreme_tot <- extreme_tot %>%
  mutate(all_time_win_rate = total_wins / total_games)

head(extreme_tot)
```

```
## # A tibble: 6 x 5
##   team                 total_games total_wins avg_win_rate all_time_win_rate
##   <chr>                  <int>      <int>        <dbl>             <dbl>
## 1 Arizona Cardinals        134         50        0.362             0.373
## 2 Atlanta Falcons           62         20        0.253             0.323
## 3 Baltimore Ravens          85         54        0.561             0.635
## 4 Buffalo Bills            139         63        0.457             0.453
## 5 Carolina Panthers         64         36        0.559             0.562
## 6 Chicago Bears            158         82        0.480             0.519
```

Average Seasonal Win Rate vs Total Number of Games Played in Extreme Weather

```r
# Create a scatterplot
ggplot(extreme_tot, aes(x = total_games, y = avg_win_rate)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  labs(
    title = "Scatterplot of Mean Win Rate vs Total Games Played in Extreme Weather",
    x = "Total Games Played in Extreme Weather",
    y = "Average All-Time Win Rate"
  ) +
  theme_minimal()
```



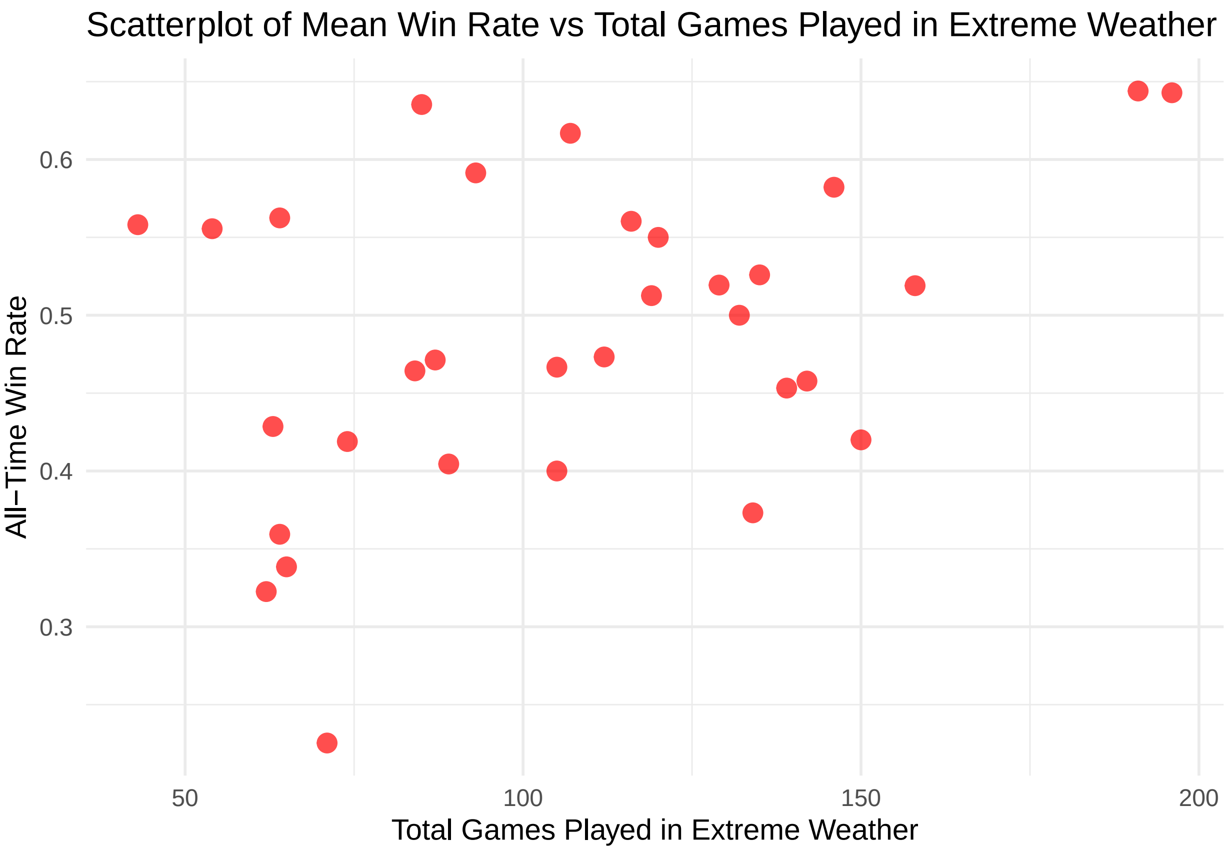
```r
# Create a linear model
lm_model <- lm(avg_win_rate ~ total_games, data = extreme_tot)

# View the summary of the model
summary(lm_model)
```

```
##
## Call:
## lm(formula = avg_win_rate ~ total_games, data = extreme_tot)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.245344 -0.048672 -0.002986  0.076339  0.151904
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3234803  0.0494657   6.539 3.12e-07 ***
## total_games 0.0013003  0.0004342   2.995  0.00546 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09401 on 30 degrees of freedom
## Multiple R-squared:  0.2302, Adjusted R-squared:  0.2045
## F-statistic: 8.969 on 1 and 30 DF,  p-value: 0.00546
```

All-Time Win Rate vs Total Number of Games Played in Extreme Weather

```r
# Create a scatterplot
ggplot(extreme_tot, aes(x = total_games, y = all_time_win_rate)) +
  geom_point(color = "red", size = 3, alpha = 0.7) +
  labs(
    title = "Scatterplot of Mean Win Rate vs Total Games Played in Extreme Weather",
    x = "Total Games Played in Extreme Weather",
    y = "All-Time Win Rate"
  ) +
  theme_minimal()
```



```
# Create a linear model
lm_model2 <- lm(all_time_win_rate ~ total_games, data = extreme_tot)

# View the summary of the model
summary(lm_model2)
```

```
##
## Call:
## lm(formula = all_time_win_rate ~ total_games, data = extreme_tot)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22356 -0.06412 -0.00460  0.06560  0.17207
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.376326   0.049453   7.610 1.74e-08 ***
## total_games 0.001022   0.000434   2.355   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09399 on 30 degrees of freedom
## Multiple R-squared:  0.1561, Adjusted R-squared:  0.1279
## F-statistic: 5.548 on 1 and 30 DF,  p-value: 0.02523
```

Conclusion

Part 1

Of the two models, the one comparing Mean Win Rate and Total Games Played in Extreme Weather is the stronger model with a higher adjusted R-squared, better predictor significance, and a good overall fit, while matching the latter model in accuracy. The results show a modest positive relationship between games played in extreme weather and average win rate, but the low R-squared suggests other factors, like team strength or weather severity, may play a bigger role. Adding more variables could improve the model's effectiveness.

Second Data Analysis - Data Set 2 (Using NFLfastR and NFLreadR Libraries)

Quarterback Stats Data Compilation

This script uses the nflfastR package to load play-by-play data for NFL seasons from 2000 to 2022 and filters it for quarterback-specific stats. It loops through each season, processes the data to extract relevant columns (e.g., passing yards, touchdowns, interceptions), and combines the results into a single dataframe. The final dataset provides a comprehensive view of quarterback performance across multiple seasons.

```r
#install.packages('nflfastR')
library(nflfastR)
library(tidyverse)

# Define the range of seasons
seasons <- 2000:2022

# Initialize an empty list to store data for each season
qb_stats_list <- list()

# Loop through each season and load the play-by-play data
for (season in seasons) {
  cat("Loading data for season:", season, "\n") # Print progress
  # Load play-by-play data
  pbp <- load_pbp(season)

  # Filter for quarterback stats
  qb_stats <- pbp %>%
    filter(!is.na(passer_player_name) | qb_scramble == 1) %>%
    select(
      season,
      week,
      play_id,
      game_id,
      passer_player_name,
      qb_scramble,
      pass_attempt,
      pass_touchdown,
      interception,
      sack,
      rushing_yards,
      passing_yards,
      complete_pass,
      incomplete_pass,
      two_point_conv_result,
      fumble_lost
    )

  # Store the filtered data
  qb_stats_list[[as.character(season)]] <- qb_stats
}
```

```
## Loading data for season: 2000
## Loading data for season: 2001
## Loading data for season: 2002
## Loading data for season: 2003
## Loading data for season: 2004
## Loading data for season: 2005
## Loading data for season: 2006
## Loading data for season: 2007
## Loading data for season: 2008
## Loading data for season: 2009
## Loading data for season: 2010
## Loading data for season: 2011
## Loading data for season: 2012
## Loading data for season: 2013
## Loading data for season: 2014
## Loading data for season: 2015
## Loading data for season: 2016
## Loading data for season: 2017
## Loading data for season: 2018
## Loading data for season: 2019
## Loading data for season: 2020
## Loading data for season: 2021
## Loading data for season: 2022
```

```
# Combine all seasons into a single dataframe
qb_stats_data <- bind_rows(qb_stats_list)

# Preview the combined dataframe
head(qb_stats_data)
```

```
## -- nflverse play by play data --------------------------------------------------

## i Data updated: 2024-08-13 11:30:22 EDT

## # A tibble: 6 x 16
##   season  week play_id game_id       passer_player_name qb_scramble pass_attempt
##    <int> <int>   <dbl> <chr>         <chr>                    <dbl>        <dbl>
## 1   2000     1     131 2000_01_ARI_~ J.Plummer                    0            1
## 2   2000     1     148 2000_01_ARI_~ J.Plummer                    0            1
## 3   2000     1     253 2000_01_ARI_~ K.Collins                    0            1
## 4   2000     1     274 2000_01_ARI_~ K.Collins                    0            1
## 5   2000     1     295 2000_01_ARI_~ K.Collins                    0            1
## 6   2000     1     333 2000_01_ARI_~ K.Collins                    0            1
## # i 9 more variables: pass_touchdown <dbl>, interception <dbl>, sack <dbl>,
## #   rushing_yards <dbl>, passing_yards <dbl>, complete_pass <dbl>,
## #   incomplete_pass <dbl>, two_point_conv_result <chr>, fumble_lost <dbl>
```

```
dim(qb_stats_data)
```

```
## [1] 464938     16
```

```
#write_csv(qb_stats_data, "qb_stats_data.csv")
```

Processing and Cleaning Play-by-Play Data

This section loads and cleans play-by-play data for each season, filtering for relevant columns and quarterback-specific plays, while categorizing weather conditions.

```
# Initialize an empty list to store cleaned data for each season
cleaned_pbp_list <- list()

# Loop through each season to load, clean, and prepare the data
for (season in seasons) {
  cat("Processing data for season:", season, "\n") # Print progress

  # Load play-by-play data for the current season
  pbp_data <- load_pbp(season)

  # Clean and prepare the data
  cleaned_pbp <- pbp_data %>%
    # Select relevant columns
    select(
      game_id, play_id, posteam, defteam, play_type, rush_attempt,
      pass_attempt, yards_gained, weather, down, yardline_100, passer, passer_jersey_number
    ) %>%
    # Remove rows with missing weather or play type information
    filter(!is.na(weather), !is.na(play_type)) %>%
    # Categorize weather conditions (e.g., temperature, wind, etc., if available)
    mutate(
      weather_condition = case_when(
        grepl("snow|cold", weather, ignore.case = TRUE) ~ "Cold",
        grepl("rain|wet", weather, ignore.case = TRUE) ~ "Rain",
        grepl("hot|warm", weather, ignore.case = TRUE) ~ "Hot",
        TRUE ~ "Clear"
      )
    ) %>%
    # Focus on quarterback-specific plays
    filter(!is.na(passer)) %>%
    group_by(weather_condition, passer) %>%
    summarize(
      total_pass_attempts = sum(pass_attempt, na.rm = TRUE),
      total_yards_gained = sum(yards_gained, na.rm = TRUE),
      avg_yards_per_attempt = mean(yards_gained / pass_attempt, na.rm = TRUE),
      total_plays = n(),
      .groups = "drop"
    )

  # Store cleaned data for the current season
  cleaned_pbp_list[[as.character(season)]] <- cleaned_pbp
}
```

```
## Processing data for season: 2000
## Processing data for season: 2001
## Processing data for season: 2002
## Processing data for season: 2003
## Processing data for season: 2004
## Processing data for season: 2005
## Processing data for season: 2006
## Processing data for season: 2007
## Processing data for season: 2008
## Processing data for season: 2009
## Processing data for season: 2010
## Processing data for season: 2011
## Processing data for season: 2012
## Processing data for season: 2013
## Processing data for season: 2014
## Processing data for season: 2015
## Processing data for season: 2016
## Processing data for season: 2017
## Processing data for season: 2018
## Processing data for season: 2019
## Processing data for season: 2020
## Processing data for season: 2021
## Processing data for season: 2022
```

```r
# Combine cleaned data from all seasons into a single dataframe
cleaned_pbp_data <- bind_rows(cleaned_pbp_list, .id = "season")

# Preview the cleaned and combined data
head(cleaned_pbp_data)
```

```
## # A tibble: 6 x 7
##   season weather_condition passer     total_pass_attempts total_yards_gained
##   <chr>  <chr>             <chr>                    <dbl>              <dbl>
## 1 2001   Clear             A.Brooks                   510               3288
## 2 2001   Clear             A.Feeley                    14                143
## 3 2001   Clear             A.Hakim                      1                 51
## 4 2001   Clear             A.Smith                      9                 40
## 5 2001   Clear             A.Van Pelt                 294               1798
## 6 2001   Clear             A.Wright                    78                517
## # i 2 more variables: avg_yards_per_attempt <dbl>, total_plays <int>
```

```r
# Save the cleaned data to a CSV file (optional)
write.csv(cleaned_pbp_data, "cleaned_pbp_data_quarterbacks_2000_2022.csv", row.names = FALSE)
```

Analyzing Play Types and Quarterback Performance

Analyzes the relationship between play types, quarterback performance, and weather conditions, categorizing plays based on weather types.

```r
options(timeout = 300) # Set timeout to 300 seconds

# Initialize an empty list to store play type and quarterback analysis for each season
play_type_qb_analysis_list <- list()

# Loop through each season to analyze play types and quarterbacks
for (season in seasons) {
  cat("Analyzing play types and quarterbacks for season:", season, "\n") # Print progress
  # Load play-by-play data for the current season
  pbp_data <- load_pbp(season)

  # Analyze play types and quarterbacks
  play_type_qb_analysis <- pbp_data %>%
    select(
      game_id, play_id, posteam, defteam, play_type, rush_attempt, pass_attempt,
      yards_gained, weather, passer, passer_jersey_number
    ) %>%
    # Filter out rows with missing play type, passer, or weather information
    filter(!is.na(play_type), !is.na(weather), !is.na(passer)) %>%
    # Categorize weather conditions (if not already categorized)
    mutate(
      weather_condition = case_when(
        grepl("snow|cold", weather, ignore.case = TRUE) ~ "Cold",
        grepl("rain|wet", weather, ignore.case = TRUE) ~ "Rain",
        grepl("hot|warm", weather, ignore.case = TRUE) ~ "Hot",
        grepl("clear|sunny", weather, ignore.case = TRUE) ~ "Clear",
        TRUE ~ "Other"
      )
    ) %>%
    # Group by weather condition, play type, and passer
    group_by(weather_condition, play_type, passer) %>%
    summarize(
      total_plays = n(),
      avg_yards_gained = mean(yards_gained, na.rm = TRUE),
      total_rush_attempts = sum(rush_attempt, na.rm = TRUE),
      total_pass_attempts = sum(pass_attempt, na.rm = TRUE),
      avg_yards_per_attempt = mean(yards_gained / pass_attempt, na.rm = TRUE),
      .groups = "drop"
    )

  # Store play type and quarterback analysis for the current season
  play_type_qb_analysis_list[[as.character(season)]] <- play_type_qb_analysis
}
```

```
## Analyzing play types and quarterbacks for season: 2000
## Analyzing play types and quarterbacks for season: 2001
## Analyzing play types and quarterbacks for season: 2002
## Analyzing play types and quarterbacks for season: 2003
## Analyzing play types and quarterbacks for season: 2004
## Analyzing play types and quarterbacks for season: 2005
## Analyzing play types and quarterbacks for season: 2006
## Analyzing play types and quarterbacks for season: 2007
## Analyzing play types and quarterbacks for season: 2008
## Analyzing play types and quarterbacks for season: 2009
## Analyzing play types and quarterbacks for season: 2010
## Analyzing play types and quarterbacks for season: 2011
## Analyzing play types and quarterbacks for season: 2012
## Analyzing play types and quarterbacks for season: 2013
## Analyzing play types and quarterbacks for season: 2014
## Analyzing play types and quarterbacks for season: 2015
## Analyzing play types and quarterbacks for season: 2016
## Analyzing play types and quarterbacks for season: 2017
## Analyzing play types and quarterbacks for season: 2018
## Analyzing play types and quarterbacks for season: 2019
## Analyzing play types and quarterbacks for season: 2020
## Analyzing play types and quarterbacks for season: 2021
## Analyzing play types and quarterbacks for season: 2022
```

```
# Combine play type and quarterback analysis data from all seasons into a single dataframe
play_type_qb_analysis_data1 <- bind_rows(play_type_qb_analysis_list, .id = "season")

# Preview the play type and quarterback analysis data
head(play_type_qb_analysis_data1)
```

```
## # A tibble: 6 x 9
##   season weather_condition play_type passer     total_plays avg_yards_gained
##   <chr>  <chr>             <chr>     <chr>            <int>            <dbl>
## 1 2001   Clear             no_play   A.Brooks             1                0
## 2 2001   Clear             no_play   A.Van Pelt           7                0
## 3 2001   Clear             no_play   A.Wright             2                0
## 4 2001   Clear             no_play   B.Favre              7                0
## 5 2001   Clear             no_play   B.Griese            15                0
## 6 2001   Clear             no_play   B.Johnson           13                0
## # i 3 more variables: total_rush_attempts <dbl>, total_pass_attempts <dbl>,
## #   avg_yards_per_attempt <dbl>
```

```
play_type_qb_analysis_data2 <- subset(play_type_qb_analysis_data1, grepl("hot|cold|rain|weather", weathe

play_type_qb_analysis_data3 <- subset(play_type_qb_analysis_data2, grepl("pass|run|qb_spike", play_type

# View the filtered data
head(play_type_qb_analysis_data3)
```

```
## # A tibble: 6 x 9
##   season weather_condition play_type passer    total_plays avg_yards_gained
##   <chr>  <chr>             <chr>     <chr>           <int>            <dbl>
## 1 2001   Cold              pass      B.Favre            28             4.96
## 2 2001   Cold              pass      C.Dillon            1             0
## 3 2001   Cold              pass      J.Fiedler          39             7.95
## 4 2001   Cold              pass      J.Kitna            48             7
## 5 2001   Cold              pass      K.Faulk             1            23
## 6 2001   Cold              pass      K.Stewart          17             3.65
## # i 3 more variables: total_rush_attempts <dbl>, total_pass_attempts <dbl>,
## #   avg_yards_per_attempt <dbl>
```

Regression Analysis and Scatterplots

Performs regression analysis to explore the impact of weather conditions on average yards per attempt, complemented by scatterplots with regression lines for visualization.

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```r
data1_clean <- na.omit(play_type_qb_analysis_data3[, c("weather_condition", "avg_yards_per_attempt")])
data2_clean <- na.omit(play_type_qb_analysis_data1[, c("weather_condition", "avg_yards_per_attempt")])

# Clean the datasets (remove rows with NA, NaN, or Inf in avg_yards_per_attempt)
data1_clean <- data1_clean %>%
  filter(!is.na(avg_yards_per_attempt) & !is.infinite(avg_yards_per_attempt))

data2_clean <- data2_clean %>%
  filter(!is.na(avg_yards_per_attempt) & !is.infinite(avg_yards_per_attempt))

# Convert weather_condition to a factor for regression
data1_clean$weather_condition <- as.factor(data1_clean$weather_condition)
data2_clean$weather_condition <- as.factor(data2_clean$weather_condition)

# Regression model for the first dataset
model1 <- lm(avg_yards_per_attempt ~ weather_condition, data = data1_clean)
summary(model1)
```

```
##
## Call:
## lm(formula = avg_yards_per_attempt ~ weather_condition, data = data1_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.756  -1.528  -0.008   1.213  45.244
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.7564     0.2690  21.400   <2e-16 ***
## weather_conditionHot     0.5473     0.5312   1.030    0.303
## weather_conditionRain   -0.3917     0.3213  -1.219    0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.346 on 959 degrees of freedom
## Multiple R-squared:  0.00457,    Adjusted R-squared:  0.002494
## F-statistic: 2.201 on 2 and 959 DF,  p-value: 0.1112
```

```r
# Add regression lines to the scatterplots
plot1 <- ggplot(data1_clean, aes(x = weather_condition, y = avg_yards_per_attempt)) +
  geom_jitter(width = 0.2, height = 0, alpha = 0.7, color = "blue") +
  labs(
    title = "Extreme Weather",
    subtitle = "Avg Yards Per Attempt vs Weather",
    x = "Weather Condition",
    y = "Average Yards Per Attempt"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  geom_smooth(method = "lm", se = FALSE, color = "black")

# Regression model for the second dataset
model2 <- lm(avg_yards_per_attempt ~ weather_condition, data = data2_clean)
summary(model2)
```

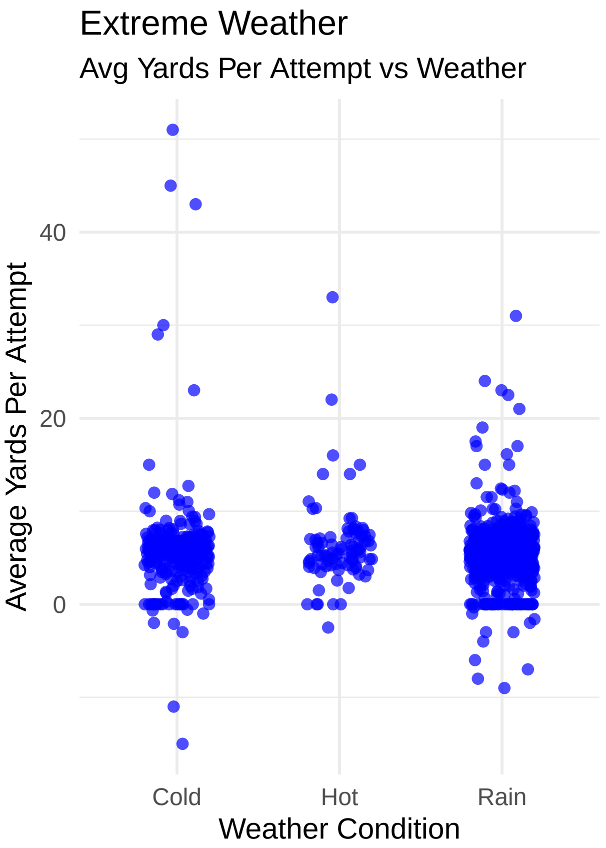
```
##
## Call:
## lm(formula = avg_yards_per_attempt ~ weather_condition, data = data2_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.838  -2.006  -0.250   0.968  75.162
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.20606    0.16050  38.668  < 2e-16 ***
## weather_conditionCold    -0.44963    0.42417  -1.060  0.28920
## weather_conditionHot      0.09765    0.68762   0.142  0.88708
## weather_conditionOther   -0.36814    0.21464  -1.715  0.08639 .
## weather_conditionRain    -0.84680    0.30250  -2.799  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.343 on 4501 degrees of freedom
## Multiple R-squared:  0.001944,   Adjusted R-squared:  0.001057
## F-statistic: 2.192 on 4 and 4501 DF,  p-value: 0.06732
```

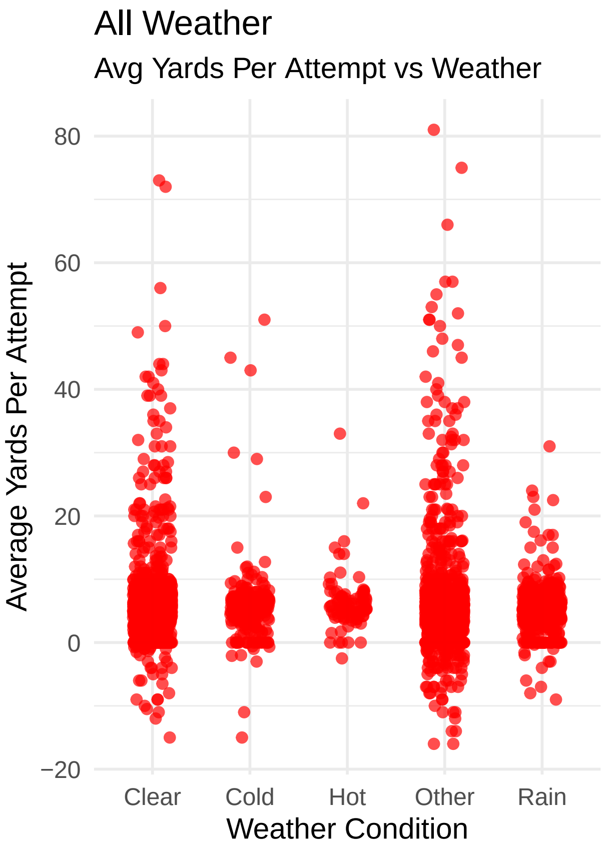
```r
plot2 <- ggplot(data2_clean, aes(x = weather_condition, y = avg_yards_per_attempt)) +
  geom_jitter(width = 0.2, height = 0, alpha = 0.7, color = "red") +
  labs(
    title = "All Weather",
    subtitle = "Avg Yards Per Attempt vs Weather",
    x = "Weather Condition",
    y = "Average Yards Per Attempt"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  geom_smooth(method = "lm", se = FALSE, color = "black")

# Combine the two plots side-by-side for comparison
grid.arrange(plot1, plot2, ncol = 2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
```





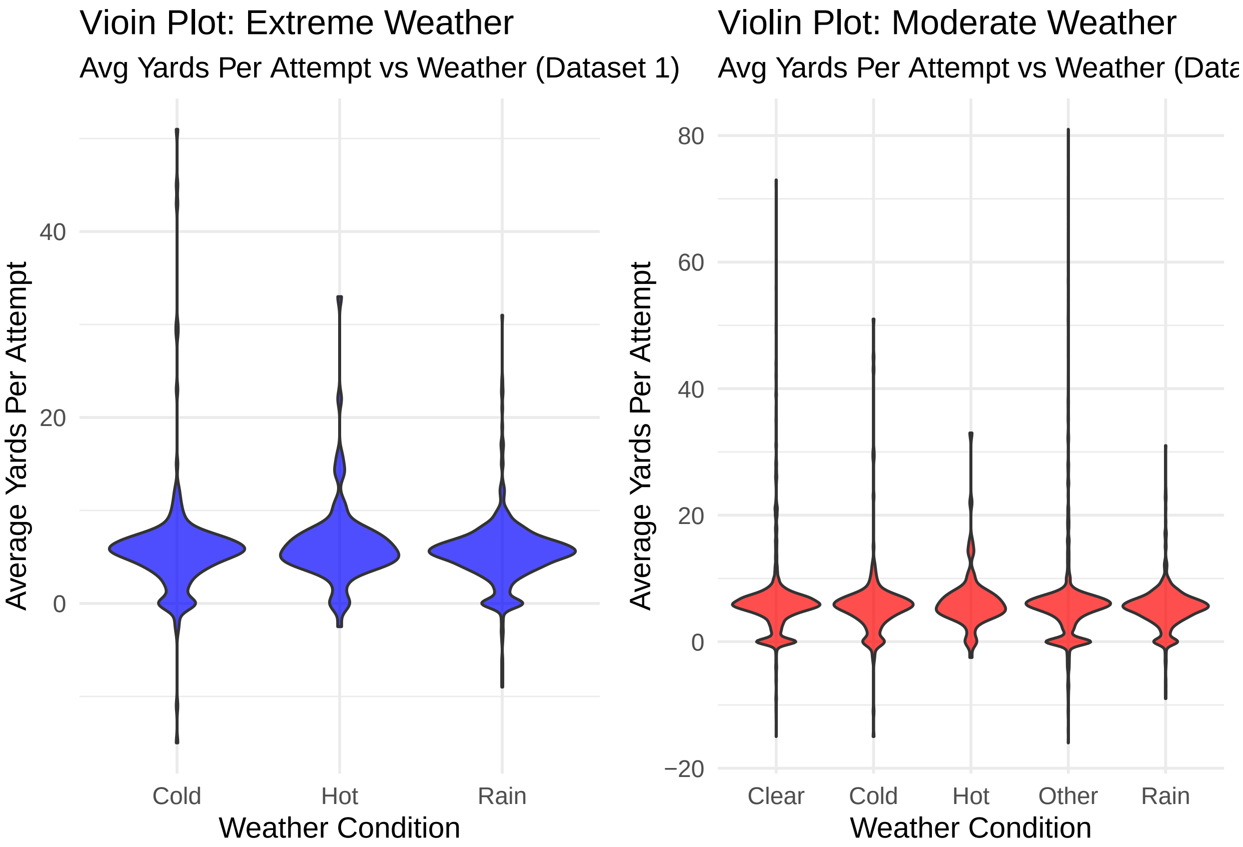
Violin Plots for Weather Impact

The code below creates violin plots to compare the distributions of average yards per attempt across weather conditions, highlighting differences between datasets for extreme and moderate weather.

```r
# Create a violin plot for Dataset 1
violinplot1 <- ggplot(data1_clean, aes(x = weather_condition, y = avg_yards_per_attempt)) +
  geom_violin(fill = "blue", alpha = 0.7) +
  labs(
    title = "Vioin Plot: Extreme Weather",
    subtitle = "Avg Yards Per Attempt vs Weather (Dataset 1)",
    x = "Weather Condition",
    y = "Average Yards Per Attempt"
  ) +
  theme_minimal()

# Create a violin plot for Dataset 2
violinplot2 <- ggplot(data2_clean, aes(x = weather_condition, y = avg_yards_per_attempt)) +
  geom_violin(fill = "red", alpha = 0.7) +
  labs(
    title = "Violin Plot: Moderate Weather",
    subtitle = "Avg Yards Per Attempt vs Weather (Dataset 2)",
    x = "Weather Condition",
    y = "Average Yards Per Attempt"
  ) +
  theme_minimal()
```

```r
grid.arrange(violinplot1, violinplot2, ncol = 2)
```



```r
# Statistical summaries using lm()
lm_summary1 <- lm(avg_yards_per_attempt ~ weather_condition, data = data1_clean)
summary(lm_summary1)
```

```
##
## Call:
## lm(formula = avg_yards_per_attempt ~ weather_condition, data = data1_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.756  -1.528  -0.008   1.213  45.244
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.7564     0.2690  21.400   <2e-16 ***
## weather_conditionHot    0.5473     0.5312   1.030    0.303
## weather_conditionRain  -0.3917     0.3213  -1.219    0.223
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.346 on 959 degrees of freedom
## Multiple R-squared:  0.00457,    Adjusted R-squared:  0.002494
## F-statistic: 2.201 on 2 and 959 DF,  p-value: 0.1112
```

```
lm_summary2 <- lm(avg_yards_per_attempt ~ weather_condition, data = data2_clean)
summary(lm_summary2)
```

```
##
## Call:
## lm(formula = avg_yards_per_attempt ~ weather_condition, data = data2_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.838  -2.006  -0.250   0.968  75.162
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              6.20606    0.16050  38.668  < 2e-16 ***
## weather_conditionCold   -0.44963    0.42417  -1.060  0.28920
## weather_conditionHot     0.09765    0.68762   0.142  0.88708
## weather_conditionOther  -0.36814    0.21464  -1.715  0.08639 .
## weather_conditionRain   -0.84680    0.30250  -2.799  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.343 on 4501 degrees of freedom
## Multiple R-squared:  0.001944,   Adjusted R-squared:  0.001057
## F-statistic: 2.192 on 4 and 4501 DF,  p-value: 0.06732
```

Conclusion

Part 2

The violin plots show that weather conditions generally have minimal impact on average yards per attempt, as the distributions overlap significantly across categories. Rain stands out slightly, with lower performance compared to other conditions, which aligns with earlier regression results showing a modest negative effect. Overall, weather seems to play a secondary role, with other factors like team or player performance likely having a greater influence. The regression results confirm that weather has little impact on average yards per attempt, with both models showing very low explanatory power (adjusted R-squared values near zero). While Rain in Dataset 2 shows a small, statistically significant negative effect, other weather conditions are not significant. This aligns with the violin plots, highlighting that performance is likely influenced more by other factors like team or player quality than by weather. These findings are consistent with the conclusion that while extreme weather may modestly impact win rates, the overall effect of weather is secondary to other factors, suggesting the need for additional variables in future models. Although there was no evidence of a strong correlation, these findings highlight the strategic importance of understanding how teams and players adapt to different weather conditions, such as how cold-weather teams might perform better in freezing conditions or how indoor teams might struggle outdoors. By identifying patterns in performance under adverse weather, such as shifts in game strategy (ie., run-heavy approaches in snow) or the challenges faced by key positions like quarterbacks and kickers, this research can help teams refine their preparation and playcalling. Additionally, it underscores the potential advantage "home" teams may have in adverse weather, offering further opportunities for teams to gain a competitive edge.

Challenges and Solutions

While working on this project, I encountered several challenges, including managing large datasets, integrating data sources, and optimizing R code. Working with play-by-play data across multiple NFL seasons and combining it with weather data resulted in long run times. To address this, I filtered out irrelevant data early on, cached intermediate results, and used tools like dplyr to streamline processing. Finding and merging compatible weather data also proved difficult due to mismatched formats, but I resolved this by standardizing fields such as team names and weather descriptions.