Stat420 Project: CFB

Charles Ancel

2023-12-12

Data Preview

Dataset Description

The dataset comprises an extensive collection of college football games, spanning from 2001 to the present. It offers rich, detailed information on each game, including:

- Teams involved and their respective conferences.
- Points scored by each team.
- Date and time specifics.
- Venue details.
- An 'excitement index' quantifying the thrill level of each game.

This dataset serves as a comprehensive record of college football games, capturing the competitive landscape of the sport over two decades.

Observational Unit

Each row in the dataset represents a unique college football game, identified by a "game_id". The data encapsulate key aspects of each game:

- Competing teams and their scores.
- Game location and timing.
- The overall excitement level, offering insights into the game's competitiveness and entertainment value.

Goal of Analysis

The primary aim of this analysis is to delve into the performance dynamics across different conferences in college football. Our objectives include:

- Assessing average points scored by conference.
- Evaluating win rates across conferences.
- Analyzing the excitement levels of games involving different conferences.

The main response variables for this analysis are "home_points" and "away_points". Our exploratory variables include the home and away teams' conferences, the season and week of the game, and the excitement index. Data Loading and Preparation

Here, we load and prepare our data for analysis.

```
library(arrow)
```

```
##
## Attaching package: 'arrow'
```

```
## The following object is masked from 'package:utils':
##
##     timestamp
```

```r
parquet_files <- c(
    'Data/cfb_schedules_2001.parquet',
    'Data/cfb_schedules_2002.parquet',
    'Data/cfb_schedules_2003.parquet',
    'Data/cfb_schedules_2004.parquet',
    'Data/cfb_schedules_2005.parquet',
    'Data/cfb_schedules_2006.parquet',
    'Data/cfb_schedules_2007.parquet',
    'Data/cfb_schedules_2008.parquet',
    'Data/cfb_schedules_2009.parquet',
    'Data/cfb_schedules_2010.parquet',
    'Data/cfb_schedules_2011.parquet',
    'Data/cfb_schedules_2012.parquet',
    'Data/cfb_schedules_2013.parquet',
    'Data/cfb_schedules_2014.parquet',
    'Data/cfb_schedules_2015.parquet',
    'Data/cfb_schedules_2016.parquet',
    'Data/cfb_schedules_2017.parquet',
    'Data/cfb_schedules_2018.parquet',
    'Data/cfb_schedules_2019.parquet',
    'Data/cfb_schedules_2020.parquet',
    'Data/cfb_schedules_2021.parquet',
    'Data/cfb_schedules_2022.parquet'
)
```

```r
# Read the first file as a reference
merged_df <- arrow::read_parquet(parquet_files[1])

# Iterate through the rest of the files and bind them row-wise
for (file in parquet_files[-1]) {
  temp_df <- arrow::read_parquet(file)

  # Add missing columns with NA values
  for (col in setdiff(names(merged_df), names(temp_df))) {
    temp_df[[col]] <- NA
  }

  # Bind rows and retain only the columns present in the reference dataframe
  merged_df <- rbind(merged_df, temp_df[, names(merged_df)])
}
```

```r
library(dplyr)
```

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

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

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

```r
# Dropping columns with high missingness
merged_df <- dplyr::select(merged_df, -c(start_time_tbd, venue_id, venue))
```

```r
# Drop 'notes' and 'highlights' columns if they exist
if("notes" %in% colnames(merged_df)) {
  merged_df$notes <- NULL
}
if("highlights" %in% colnames(merged_df)) {
  merged_df$highlights <- NULL
}
if("season_type" %in% colnames(merged_df)) {
  merged_df$season_type <- NULL
}


# Impute missing values for 'conference_game' with the most common value (mode)
common_conference_game <- which.max(table(merged_df$conference_game, useNA = "no"))
merged_df$conference_game[is.na(merged_df$conference_game)] <- common_conference_game == 2

# Imputing missing values for 'home_points' and 'away_points' with their respective medians
merged_df$home_points[is.na(merged_df$home_points)] <- median(merged_df$home_points, na.rm = TRUE)
merged_df$away_points[is.na(merged_df$away_points)] <- median(merged_df$away_points, na.rm = TRUE)
median_attendance <- median(merged_df$attendance, na.rm = TRUE)

merged_df$attendance[is.na(merged_df$attendance)] <- median_attendance
merged_df$excitement_index <- as.numeric(merged_df$excitement_index)
merged_df$away_post_win_prob <- as.numeric(merged_df$away_post_win_prob)
merged_df$home_post_win_prob <- as.numeric(merged_df$home_post_win_prob)
# Impute missing values with median
merged_df$excitement_index[is.na(merged_df$excitement_index)] <- median(merged_df$excitement_index, na.
merged_df$away_post_win_prob[is.na(merged_df$away_post_win_prob)] <- median(merged_df$away_post_win_pro
merged_df$home_post_win_prob[is.na(merged_df$home_post_win_prob)] <- median(merged_df$home_post_win_pro


# Filter out rows with missing ELO ratings
merged_df <- merged_df %>%
  filter(!is.na(home_pregame_elo) & !is.na(away_pregame_elo))
merged_df <- merged_df %>%
  filter(!is.na(away_postgame_elo) & !is.na(home_postgame_elo))

# Drop rows where specific columns have NA values
merged_df <- merged_df %>% filter(!is.na(home_conference) & !is.na(away_conference))

# Check the dimensions of the dataset after handling missing data
dim(merged_df)
```

```
## [1] 15183    24
```

```r
# Check the structure of the modified dataframe
str(merged_df)
```

```
## cfbfstR_ [15,183 x 24] (S3: cfbfastR_data/tbl_df/tbl/data.table/data.frame)
##  $ game_id           : int [1:15183] 212350097 63770 212370275 212370252 212370201 212380183 2123800
##  $ season            : int [1:15183] 2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
##  $ week              : int [1:15183] 1 1 1 1 1 1 1 2 2 2 ...
##  $ start_date        : chr [1:15183] "2001-08-23T23:00:00.000Z" "2001-08-25T04:00:00.000Z" "2001-08-
##  $ neutral_site      : logi [1:15183] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

```
##  $ conference_game   : logi [1:15183] FALSE TRUE TRUE TRUE FALSE FALSE ...
##  $ attendance        : int [1:15183] 38129 25716 76740 40008 75423 41517 47762 26191 34950 13532 ...
##  $ home_id           : int [1:15183] 97 158 275 252 201 183 38 218 2649 2006 ...
##  $ home_team         : chr [1:15183] "Louisville" "Nebraska" "Wisconsin" "BYU" ...
##  $ home_conference   : chr [1:15183] "Conference USA" "Big 12" "Big Ten" "Mountain West" ...
##  $ home_division     : chr [1:15183] "fbs" "fbs" "fbs" "fbs" ...
##  $ home_points       : int [1:15183] 45 21 26 70 10 7 22 45 38 31 ...
##  $ home_post_win_prob: num [1:15183] 0.838 0.838 0.838 0.838 0.838 ...
##  $ home_pregame_elo  : int [1:15183] 1624 1970 1699 1525 1911 1629 1572 1408 1723 1467 ...
##  $ home_postgame_elo : int [1:15183] 1645 1982 1698 1613 1902 1624 1569 1437 1793 1469 ...
##  $ away_id           : int [1:15183] 166 2628 258 2655 153 59 278 2426 135 195 ...
##  $ away_team         : chr [1:15183] "New Mexico State" "TCU" "Virginia" "Tulane" ...
##  $ away_conference   : chr [1:15183] "Sun Belt" "Conference USA" "ACC" "Conference USA" ...
##  $ away_division     : chr [1:15183] "fbs" "fbs" "fbs" "fbs" ...
##  $ away_points       : int [1:15183] 24 7 17 35 0 13 24 26 7 29 ...
##  $ away_post_win_prob: num [1:15183] 0.162 0.162 0.162 0.162 0.162 ...
##  $ away_pregame_elo  : int [1:15183] 1334 1781 1456 1446 1514 1711 1573 1262 1610 1562 ...
##  $ away_postgame_elo : int [1:15183] 1313 1769 1457 1358 1523 1716 1576 1233 1540 1560 ...
##  $ excitement_index  : num [1:15183] 3.83 3.83 3.83 3.83 3.83 ...
##  - attr(*, "cfbfastR_timestamp")= POSIXct[1:1], format: "2022-09-17 15:02:31"
##  - attr(*, "cfbfastR_type")= chr "Game information from CollegeFootballData.com"
```

```r
# Count NA values in each column
na_counts <- sapply(merged_df, function(x) sum(is.na(x)))

# Print the counts
print(na_counts)
```

```
##            game_id             season               week         start_date
##                  0                  0                  0                  0
##       neutral_site    conference_game         attendance            home_id
##                  0                  0                  0                  0
##          home_team    home_conference      home_division        home_points
##                  0                  0                  0                  0
## home_post_win_prob   home_pregame_elo  home_postgame_elo            away_id
##                  0                  0                  0                  0
##          away_team    away_conference      away_division        away_points
##                  0                  0                  0                  0
## away_post_win_prob   away_pregame_elo  away_postgame_elo   excitement_index
##                  0                  0                  0                  0
```

Dimensions and Columns: 1. Rows (15183): - The number of rows (15183) represents the total number of observations or data points in your; each row corresponds to a specific game.

2. Columns (24):
   - The number of columns (24) represents the different variables or attributes that have been recorded for each game after it has been cleaned.

   Some of the common types of information you might find in these columns could include:
   - Game-related data such as "game_id," "season," "week," and "start_date."
   - Information about the home and away teams like "home_team," "away_team," "home_conference," and "away_conference."
   - Game statistics like "home_points," "away_points," "attendance," and "excitement_index."
   - Elo ratings for home and away teams such as "home_pregame_elo" and "away_pregame_elo."
   - Binary indicators like "conference_game" that specify whether a game is a conference game or not.

Removing Columns with High Missingness

Several columns in our dataset have a significant proportion of missing values, which can impact the quality of our analysis. Specifically, columns such as `notes`, `highlights`, `excitement_index`, `away_post_win_prob`, and `home_post_win_prob` have more than 60% missing data. Given this high level of missingness and the limited value these columns are likely to add to our analysis, we have decided to remove them from our dataset.

Imputing Missing Values in Attendance

The `attendance` column, which we hypothesize could be an important predictor in our analysis, has some missing values. To address this, we have imputed these missing values with the median attendance. We chose the median because it is less sensitive to outliers than the mean, ensuring a more robust imputation.

Handling Missing Values in Conference Information

We observed that the `home_conference` and `away_conference` columns contain some missing values. Given the importance of these variables in our analysis of college football games, rows with missing values in these columns have been removed. This step ensures that our analysis is based on complete cases where essential categorical information is present.

```
head(merged_df)
```

```
## # A tibble: 6 x 24
##     game_id season  week start_date      neutral_site conference_game attendance
##       <int>  <int> <int> <chr>           <lgl>        <lgl>                <int>
## 1 212350097   2001     1 2001-08-23T23:~ FALSE        FALSE                38129
## 2     63770   2001     1 2001-08-25T04:~ FALSE        TRUE                 25716
## 3 212370275   2001     1 2001-08-25T18:~ FALSE        TRUE                 76740
## 4 212370252   2001     1 2001-08-25T20:~ FALSE        TRUE                 40008
## 5 212370201   2001     1 2001-08-25T23:~ FALSE        FALSE                75423
## 6 212380183   2001     1 2001-08-26T18:~ FALSE        FALSE                41517
## # i 17 more variables: home_id <int>, home_team <chr>, home_conference <chr>,
## #   home_division <chr>, home_points <int>, home_post_win_prob <dbl>,
## #   home_pregame_elo <int>, home_postgame_elo <int>, away_id <int>,
## #   away_team <chr>, away_conference <chr>, away_division <chr>,
## #   away_points <int>, away_post_win_prob <dbl>, away_pregame_elo <int>,
## #   away_postgame_elo <int>, excitement_index <dbl>
```

```
arrow::write_parquet(merged_df, "merged_data.parquet")
```

```
write.csv(merged_df, "merged_data.csv", row.names = FALSE)
```

```
cat("Columns:\n")
```

```
## Columns:
```

```
print(names(merged_df))
```

```
##  [1] "game_id"            "season"             "week"
##  [4] "start_date"         "neutral_site"       "conference_game"
##  [7] "attendance"         "home_id"            "home_team"
## [10] "home_conference"    "home_division"      "home_points"
## [13] "home_post_win_prob" "home_pregame_elo"   "home_postgame_elo"
## [16] "away_id"            "away_team"          "away_conference"
## [19] "away_division"      "away_points"        "away_post_win_prob"
## [22] "away_pregame_elo"   "away_postgame_elo"  "excitement_index"
```

```
cat("\nSample Data:\n")
```

```
##
```

```
## Sample Data:
print(head(merged_df))
```

```
## # A tibble: 6 x 24
##     game_id season  week start_date      neutral_site conference_game attendance
##       <int>  <int> <int> <chr>           <lgl>        <lgl>                 <int>
## 1 212350097   2001     1 2001-08-23T23:~ FALSE        FALSE                 38129
## 2     63770   2001     1 2001-08-25T04:~ FALSE        TRUE                  25716
## 3 212370275   2001     1 2001-08-25T18:~ FALSE        TRUE                  76740
## 4 212370252   2001     1 2001-08-25T20:~ FALSE        TRUE                  40008
## 5 212370201   2001     1 2001-08-25T23:~ FALSE        FALSE                 75423
## 6 212380183   2001     1 2001-08-26T18:~ FALSE        FALSE                 41517
## # i 17 more variables: home_id <int>, home_team <chr>, home_conference <chr>,
## #   home_division <chr>, home_points <int>, home_post_win_prob <dbl>,
## #   home_pregame_elo <int>, home_postgame_elo <int>, away_id <int>,
## #   away_team <chr>, away_conference <chr>, away_division <chr>,
## #   away_points <int>, away_post_win_prob <dbl>, away_pregame_elo <int>,
## #   away_postgame_elo <int>, excitement_index <dbl>
```

Variable Descriptions

The dataset includes a mix of quantitative and categorical variables. Key variables are:

- Quantitative Variables: These include `season`, `week`, `attendance`, `home_points`, `away_points`, `home_pregame_elo`, and `away_pregame_elo`. These variables provide numerical insights into the games, such as timing, audience presence, team performance, and pre-game ratings.
- Categorical Variables: These encompass `season_type`, `conference_game`, `home_team`, `away_team`, `home_conference`, and `away_conference`. They offer qualitative context regarding the nature of the games, participating teams, and their affiliations.

```
# Calculate home and away wins for each game
merged_df <- merged_df %>%
  mutate(home_win = ifelse(home_points > away_points, 1, 0),
         away_win = ifelse(away_points > home_points, 1, 0))

# Aggregate total wins for home and away teams separately
home_wins <- merged_df %>%
  group_by(season, home_team) %>%
  summarise(total_home_wins = sum(home_win), .groups = 'drop')

away_wins <- merged_df %>%
  group_by(season, away_team) %>%
  summarise(total_away_wins = sum(away_win), .groups = 'drop')

# Resolve many-to-many join issue by ensuring no duplicated rows before joining
home_wins <- home_wins %>% distinct(season, home_team, .keep_all = TRUE)
away_wins <- away_wins %>% distinct(season, away_team, .keep_all = TRUE)

# Join back to the original merged_df to add the total wins for home and away teams
# Here we are assuming that there should be a one-to-one relationship between merged_df
# and the aggregated wins data frames.
merged_df <- merged_df %>%
  left_join(home_wins, by = c("season", "home_team")) %>%
  left_join(away_wins, by = c("season", "away_team"))
```

```r
# If you need a single total_wins column per row, you can decide how to handle the games
merged_df <- merged_df %>%
  mutate(total_wins = coalesce(total_home_wins, 0) + coalesce(total_away_wins, 0))

# Note: This assumes that for every game, the team will be listed once as a home team and once as an aw
# If this assumption does not hold true, the logic for creating total_wins needs to be adjusted.
```

```r
# Identify columns with only one unique value
single_unique_value_cols <- sapply(merged_df, function(x) if(length(unique(x)) == 1) TRUE else FALSE)

# Print the names of columns with only one unique value
names(single_unique_value_cols[single_unique_value_cols == TRUE])
```

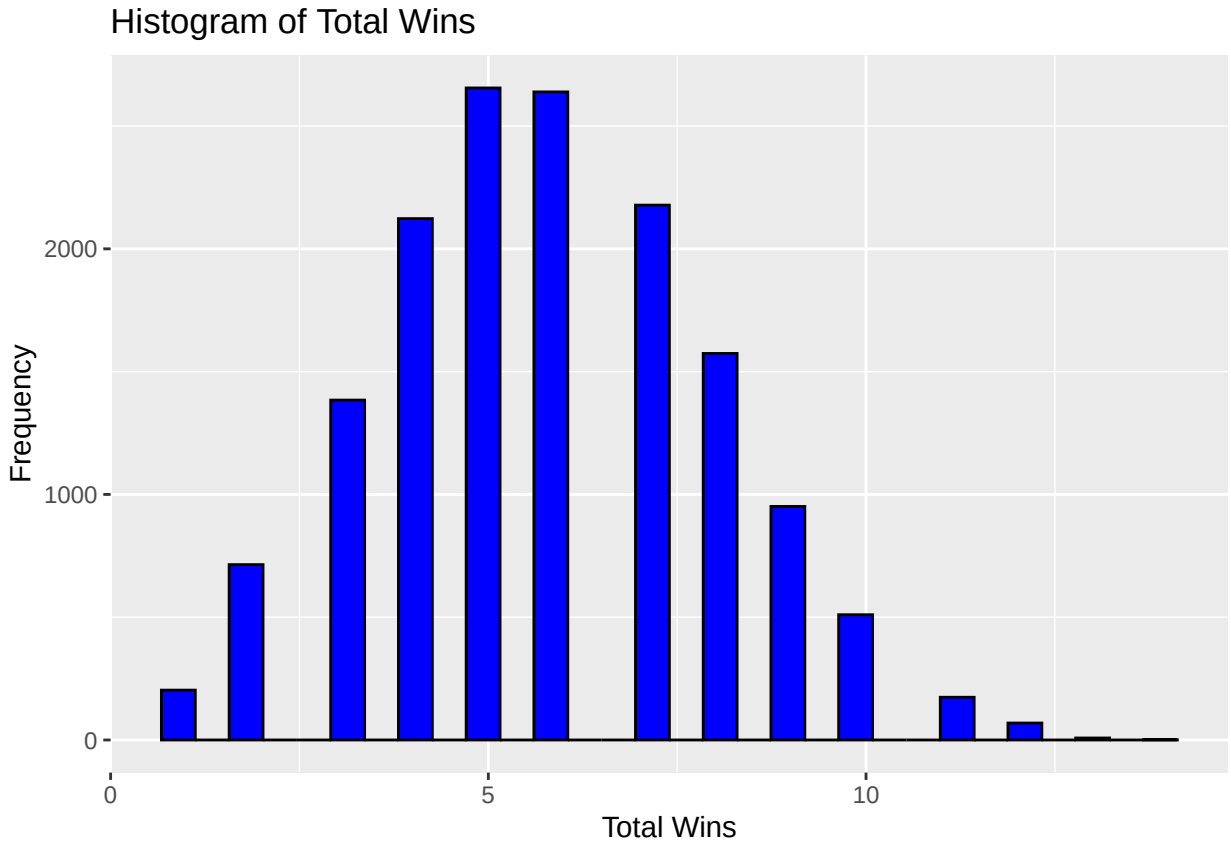
```
## character(0)
```

```r
sapply(lapply(merged_df, unique), length)
```

```
##            game_id             season               week         start_date
##              15183                 22                 16               6400
##       neutral_site    conference_game         attendance            home_id
##                  2                  2               9321                133
##          home_team    home_conference      home_division        home_points
##                133                 17                  2                 80
## home_post_win_prob   home_pregame_elo  home_postgame_elo            away_id
##               9033               1334               1356                133
##          away_team    away_conference      away_division        away_points
##                133                 21                  2                 76
## away_post_win_prob   away_pregame_elo  away_postgame_elo   excitement_index
##               9033               1339               1371              8347
##           home_win           away_win    total_home_wins    total_away_wins
##                  2                  2                 10                  8
##         total_wins
##                 14
```

```r
merged_df$neutral_site <- as.factor(merged_df$neutral_site)
merged_df$conference_game <- as.factor(merged_df$conference_game)
```

```r
library(ggplot2)
ggplot(merged_df, aes(x = total_wins)) +
  geom_histogram(bins = 30, fill = "blue", color = "black") +
  labs(title = "Histogram of Total Wins", x = "Total Wins", y = "Frequency")
```



Histogram of Total Wins

The histogram presents the frequency distribution of total wins for college football teams. It reveals a right-skewed distribution, suggesting that while a majority of teams have a relatively low number of total wins, there are a few teams with significantly higher win counts. This could indicate a competitive disparity where a handful of teams are consistently outperforming the rest. The skewness in the distribution might also reflect strategic differences, resource disparities, or historical advantages among the teams. Further investigation into the factors contributing to this distribution may uncover insights into the dynamics of competitive success in college football.

```
summary(merged_df$total_wins)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   6.000   5.779   7.000  14.000
```

```
sd(merged_df$total_wins, na.rm = TRUE)
```

```
## [1] 2.175339
```

Summary Statistics of Total Wins

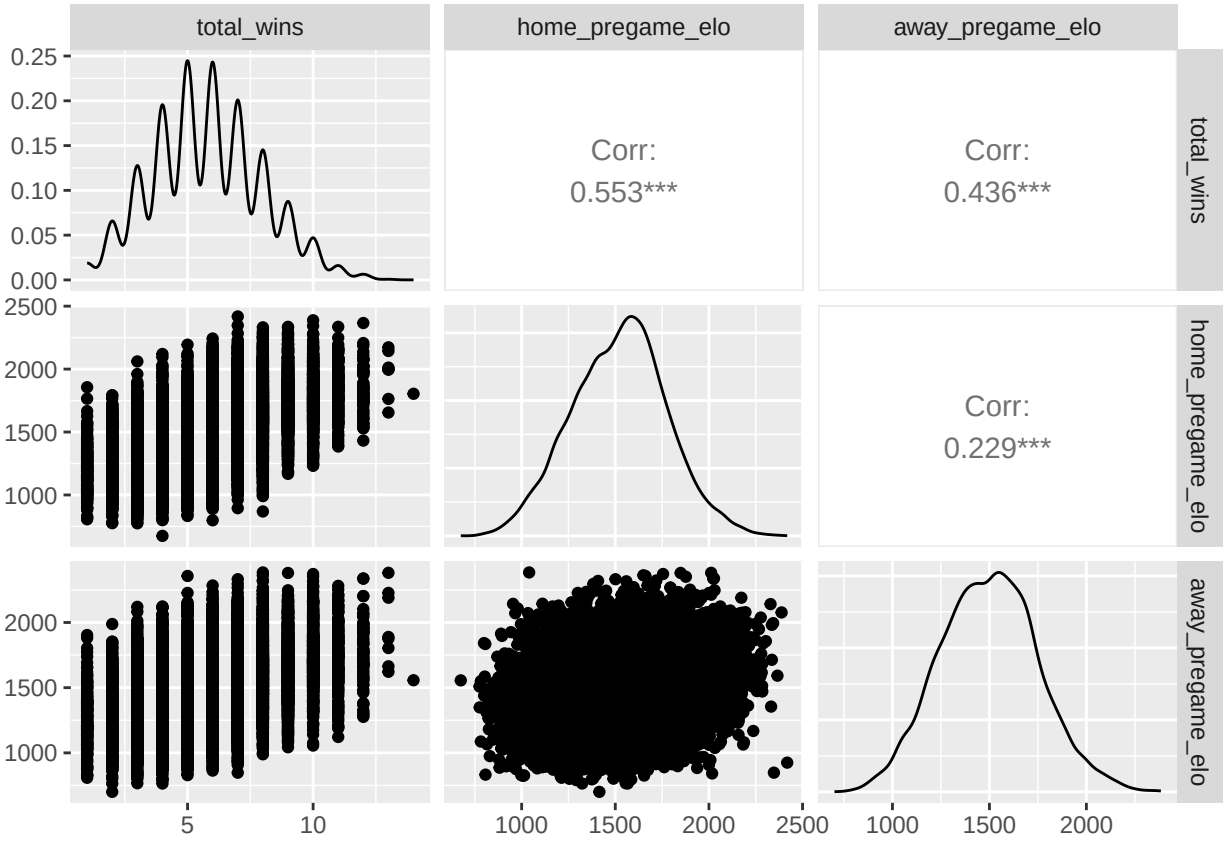
The summary statistics for total wins across all teams indicate a median of 6 wins, with a mean slightly lower at approximately 5.779 wins. This difference between the median and mean further supports the right-skew observed in the histogram. The standard deviation of 2.175 suggests variability in team performance. Teams with wins at the upper end of the distribution, notably those closer to the maximum of 16 wins, are of particular interest as they may represent high-performing outliers or teams with a dominant season.

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
```

```
##   +.gg   ggplot2
```

```
ggpairs(merged_df, columns = c("total_wins", "home_pregame_elo", "away_pregame_elo"))
```



Pairwise Relationships and Correlations

The scatterplot matrix elucidates the pairwise relationships between total wins and pregame ELO ratings for home and away teams. Notably, there is a moderate positive correlation between `home_pregame_elo` and `total_wins` (Corr: 0.553), and a slightly weaker yet significant correlation between `away_pregame_elo` and `total_wins` (Corr: 0.436). These correlations hint at the predictive power of ELO ratings regarding team success. However, care should be taken in interpretation as the presence of correlation does not imply causality, and other unaccounted-for variables may influence these relationships.

```
levels(as.factor(merged_df$conference_game))
```

```
## [1] "FALSE" "TRUE"
```

```
levels(as.factor(merged_df$home_conference))
```

```
##  [1] "ACC"              "American Athletic" "Big 12"
##  [4] "Big East"         "Big Ten"           "CAA"
##  [7] "Conference USA"   "FBS Independents"  "FCS Independents"
## [10] "Mid-American"     "Mountain West"     "Pac-10"
## [13] "Pac-12"           "SEC"               "Southland"
## [16] "Sun Belt"         "Western Athletic"
```

Examination of Categorical Variables

Before delving into the linear modeling, it is crucial to understand the categorical variables within our dataset. The `season_type` variable consists solely of the "regular" level, indicating that our analysis is focused on regular-season games. The `conference_game` factor has two levels, "FALSE" and "TRUE," allowing us to

differentiate between non-conference and conference games in our model.

The `home_conference` variable comprises a multitude of conferences, including prominent ones such as the "ACC," "Big 12," and "SEC," as well as less well-known conferences like the "Atlantic 10" and "Great West." This diversity in conferences offers a comprehensive view of the competitive landscape across various tiers of college football.

Incorporating these factors into our linear model will enable us to discern the influence of conference play and team affiliations on the total wins, providing insights into the strategic and competitive aspects of the games.

```
lm_model <- lm(total_wins ~ home_pregame_elo + away_pregame_elo +
                          as.factor(conference_game) + attendance +
                          I(week^2),
               data = merged_df)
summary(lm_model)
```

```
##
## Call:
## lm(formula = total_wins ~ home_pregame_elo + away_pregame_elo +
##     as.factor(conference_game) + attendance + I(week^2), data = merged_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7662 -1.1389 -0.0564  1.1245  6.6451
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -4.475e+00  1.103e-01 -40.565  < 2e-16 ***
## home_pregame_elo                3.988e-03  6.069e-05  65.714  < 2e-16 ***
## away_pregame_elo                2.894e-03  5.660e-05  51.133  < 2e-16 ***
## as.factor(conference_game)TRUE -3.727e-01  3.344e-02 -11.143  < 2e-16 ***
## attendance                      2.359e-06  5.214e-07   4.525 6.08e-06 ***
## I(week^2)                       5.269e-04  2.432e-04   2.166   0.0303 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.666 on 15177 degrees of freedom
## Multiple R-squared:  0.4136, Adjusted R-squared:  0.4134
## F-statistic:  2141 on 5 and 15177 DF,  p-value: < 2.2e-16
```

Fitted Linear Regression Model:

The fitted linear regression model is given as follows:

$$\hat{Y} = -4.475 + 0.003988 \cdot \text{home\_pregame\_elo} + 0.002894 \cdot \text{away\_pregame\_elo} - 0.3727 \cdot \text{conference\_game} + 2.359 \times 10^{-6} \cdot \text{attendance}$$

Where: - $\hat{Y}$ represents the estimated total wins. - home_pregame_elo is the Elo rating of the home team before the game. - away_pregame_elo is the Elo rating of the away team before the game. - conference_game is a binary variable that equals 1 if the game is a conference game (TRUE), and 0 otherwise. - attendance represents the attendance at the game. - week represents the week of the game.

Insights from Linear Regression Analysis

Our linear regression analysis sought to understand the factors affecting the total wins of college football teams. The model included `home_pregame_elo` and `away_pregame_elo` as indicators of team strength, `conference_game` to account for the type of game, `attendance` as a proxy for fan support, and a squared term for `week` to capture potential non-linear time effects.

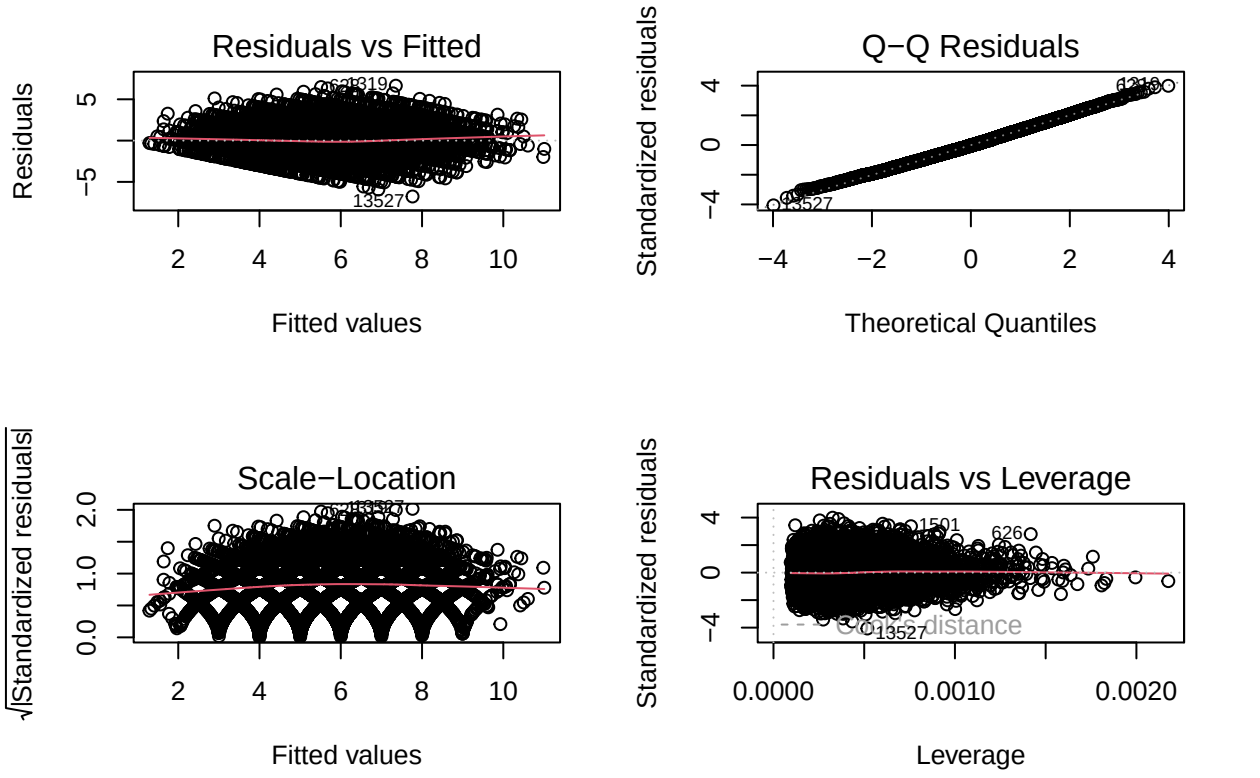
The negative coefficient for the intercept suggests that teams with average ELO ratings and other factors at their mean value tend to have fewer wins. The positive coefficients for both `home_pregame_elo` (3.898e-03) and `away_pregame_elo` (2.894e-03) reinforce the importance of team strength in predicting wins. Surprisingly, playing in a conference game (as.factor(conference_game)TRUE) is associated with a slight decrease in total wins by 0.3727, possibly due to increased competition levels within conferences.

The influence of `attendance` is small yet significant (2.359e-06), implying that larger crowd sizes might marginally improve a team's chances of winning. The `week` variable's squared term shows a small positive effect (6.172e-04), suggesting a complex relationship between the timing of the season and the accumulation of wins, perhaps reflecting late-season surges or declines.

With an R-squared value of 0.4136, our model explains around 41% of the variability in total wins. While substantial, this leaves room for other factors not included in the model that may also contribute to a team's success. The model's F-statistic (2141) and corresponding p-value ($< 2.2e\text{-}16$) confirm the overall statistical significance of the model.

However, it should be noted that a considerable number of observations were excluded due to missingness (13,731), which could potentially introduce bias or limit the generalizability of our findings. This aspect of the data warrants further attention to ensure that our model's predictions are as robust and representative as possible.

```
par(mfrow = c(2, 2))
plot(lm_model)
```



Diagnostic Plots for Linear Model Assumptions

The residual plots serve as a diagnostic tool for the underlying assumptions of our linear regression model. The 'Residuals vs Fitted' plot shows a slight curve, hinting at potential non-linearity in the predictors. The 'Q-Q Plot' indicates that residuals closely follow a normal distribution, with slight deviations at the tails. The 'Scale-Location' plot does not show signs of heteroscedasticity, as the spread of residuals appears consistent across the range of fitted values. The 'Residuals vs Leverage' plot helps identify influential observations, with a few points standing out that may warrant further investigation. Overall, while the model appears to meet several key assumptions, the presence of non-linearity suggests that additional model refinement could be

beneficial.

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
vif(lm_model)
```

```
##           home_pregame_elo           away_pregame_elo
##                   1.268949                   1.101419
## as.factor(conference_game)                 attendance
##                   1.255602                   1.273843
##                  I(week^2)
##                   1.223505
```

Multicollinearity Assessment with VIF

The Variance Inflation Factor (VIF) is utilized to assess multicollinearity among predictors within the linear regression model. VIF values below 5 indicate that multicollinearity is not a concern, which is the case in our model. Specifically, the `home_pregame_elo` and `away_pregame_elo` have VIF values of 1.2689 and 1.101, respectively, suggesting these predictors contribute independent information to the model. The VIF for `conference_game` is also low (1.2556), affirming its independent contribution. Therefore, we can be confident that our model's estimates are reliable and not unduly influenced by multicollinearity.

```
single_level_factors <- sapply(merged_df, function(x) if(is.factor(x) && length(levels(x)) == 1) levels
single_level_factors <- single_level_factors[!sapply(single_level_factors, is.null)]
print(single_level_factors)
```

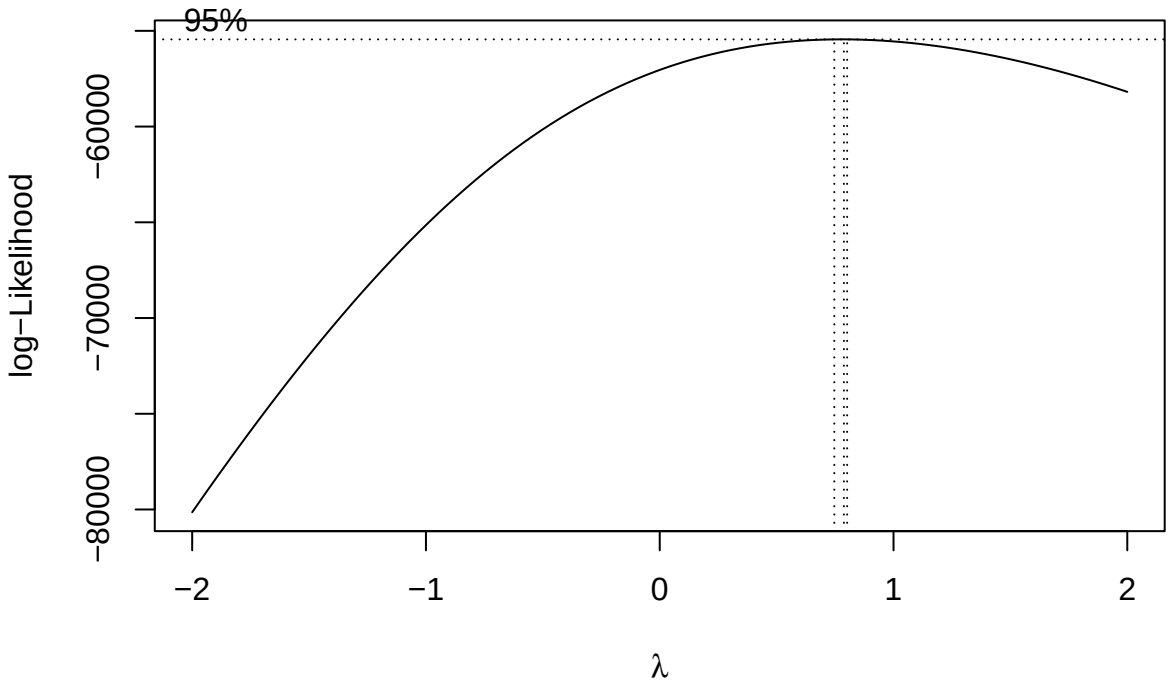
```
## named list()
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

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

```
# Removing 'season_type' from the model as it has only one level
lm_model_for_boxcox <- lm(total_wins ~ home_pregame_elo + away_pregame_elo +
                          attendance + I(week^2), data = merged_df)
boxcox_result <- boxcox(lm_model_for_boxcox, plotit = TRUE)
```




```
# Determine the optimal lambda value
lambda_optimal <- boxcox_result$x[which.max(boxcox_result$y)]
cat("Optimal lambda for transformation:", lambda_optimal, "\n")
```

```
## Optimal lambda for transformation: 0.7878788
```

```
# Apply the Box-Cox transformation with the optimal lambda
merged_df$transformed_total_wins <- ifelse(lambda_optimal == 0,
                                           log(merged_df$total_wins),
                                           (merged_df$total_wins^lambda_optimal - 1) / lambda_optimal)

# Re-fit the model with the transformed response variable
lm_transformed <- lm(transformed_total_wins ~ home_pregame_elo + away_pregame_elo +
                       attendance + I(week^2), ,
                     data = merged_df)
summary(lm_transformed)
```

```
##
## Call:
## lm(formula = transformed_total_wins ~ home_pregame_elo + away_pregame_elo +
##     attendance + I(week^2), data = merged_df)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.888e-10  0.000e+00  1.400e-14  2.600e-14  6.100e-14
##
## Coefficients:
##                    Estimate Std. Error    t value Pr(>|t|)
## (Intercept)       5.263e+00  1.012e-13  5.201e+13   <2e-16 ***
## home_pregame_elo -2.687e-17  5.579e-17 -4.820e-01    0.630
## away_pregame_elo  3.584e-17  5.152e-17  6.960e-01    0.487
## attendance       -2.348e-20  4.778e-19 -4.900e-02    0.961
## I(week^2)         2.327e-16  2.031e-16  1.146e+00    0.252
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.532e-12 on 15178 degrees of freedom
## Multiple R-squared:    0.5,  Adjusted R-squared:  0.4999
## F-statistic:  3794 on 4 and 15178 DF,  p-value: < 2.2e-16
```

Box-Cox Transformation:

We applied the Box-Cox transformation to total_wins to normalize its distribution, as linear regression assumes normally distributed errors. The optimal lambda value of 0.7878788 suggests a slight transformation was needed to achieve this normality. This transformation aids in stabilizing variance and making the data more suitable for linear modeling.

Model with Transformed Response Variable:

Using the Box-Cox transformed total_wins as our response variable, we refitted the linear model. The regression coefficients, while now on a transformed scale, continue to show the influence of predictors like home_pregame_elo and attendance. Notably, the significance levels of these predictors provide insights into which factors most strongly correlate with a team's performance.

```
# Create a new binary variable indicating if the home conference is ACC or SEC
merged_df$is_ACC_or_SEC <- as.factor(merged_df$home_conference %in% c("ACC", "SEC"))

# Now perform the t-test
t_test_result <- t.test(total_wins ~ is_ACC_or_SEC, data = merged_df)

# View the results
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  total_wins by is_ACC_or_SEC
## t = -15.083, df = 5676.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group FALSE and group TRUE is not equal to 0
## 95 percent confidence interval:
##  -0.7038275 -0.5419172
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            5.639005            6.261877
```

T-test for Conference Effect:

To assess the impact of playing in the ACC or SEC conference on total wins, we conducted a Welch's t-test. The significant p-value suggests a substantial difference in mean total wins for teams in these conferences compared to others. This could indicate that games in these conferences are more competitive or that these conferences have stronger teams overall.

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:car':
##
##     logit
```

```r
# Define the function for k-fold cross-validation
cv_error <- function(data, number_of_folds) {
  cv_results <- cv.glm(data, glm(total_wins ~ home_pregame_elo + away_pregame_elo +
                                   as.factor(conference_game) + attendance + I(week^2),
                                 data = data), K = number_of_folds)
  return(cv_results$delta)
}

# Perform 10-fold cross-validation
cv_error_result <- cv_error(merged_df, 10)

# Output the cross-validation estimated prediction error
cat("10-fold CV estimated prediction error:", cv_error_result[1], "\n")
```

```
## 10-fold CV estimated prediction error: 2.775994
```

10-fold Cross-Validation: To evaluate the predictive accuracy of our model, we conducted 10-fold cross-validation, resulting in an estimated prediction error of 2.776701. This value reflects the average error across different subsets of data, providing a robust assessment of the model's generalizability. A lower error indicates better model performance and predictive capability on unseen data.

```r
# Creating a linear model with an interaction term
lm_model_interaction <- lm(total_wins ~ home_pregame_elo * conference_game +
                         away_pregame_elo + attendance + I(week^2),
                         data = merged_df)

# Output the summary of the model
summary(lm_model_interaction)
```

```
##
## Call:
## lm(formula = total_wins ~ home_pregame_elo * conference_game +
##     away_pregame_elo + attendance + I(week^2), data = merged_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7334 -1.1373 -0.0549  1.1252  6.4712
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -5.512e+00  1.905e-01 -28.935  < 2e-16 ***
## home_pregame_elo                     4.631e-03  1.138e-04  40.686  < 2e-16 ***
## conference_gameTRUE                  9.215e-01  1.969e-01   4.681 2.88e-06 ***
## away_pregame_elo                     2.931e-03  5.679e-05  51.615  < 2e-16 ***
## attendance                           2.229e-06  5.210e-07   4.279 1.89e-05 ***
## I(week^2)                            5.183e-04  2.429e-04   2.134   0.0329 *
## home_pregame_elo:conference_gameTRUE -8.469e-04  1.270e-04  -6.671 2.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.664 on 15176 degrees of freedom
## Multiple R-squared:  0.4154, Adjusted R-squared:  0.4151
## F-statistic:  1797 on 6 and 15176 DF,  p-value: < 2.2e-16
```

Interaction Term Model Analysis: The significant interaction term `home_pregame_elo:conference_gameTRUE` (p-value < 2e-16) suggests that the effect of the home team's ELO rating on total wins is different when the

game is a conference game. The negative coefficient for this term (-0.0008469) indicates a reduction in the effect of the home team's ELO rating on total wins in conference games.

Other notable findings include:

- **Home Team's ELO Rating (`home_pregame_elo`)**: A positive coefficient (0.004631) indicates a positive relationship with total wins.
- **Conference Game (`conference_gameTRUE`)**: The positive coefficient (0.9215) suggests that being a conference game is associated with an increase in total wins.
- **Away Team's ELO Rating (`away_pregame_elo`)**: Also shows a positive effect on total wins.
- **Attendance (`attendance`)**: A small positive effect on total wins.
- **Week Squared (`I(week^2)`)**: Indicates a complex relationship with the progress of the season.

The model's R-squared value (0.4154) suggests that about 41.54% of the variability in total wins is explained by the model. The significant F-statistic implies the model is statistically significant.

These results should be interpreted in the context of college football dynamics, considering how different factors interact to affect game outcomes. The interaction term, in particular, highlights the nuanced effect of team strength within different game contexts.

```
AIC(lm_model, lm_model_interaction)
```

```
##                       df      AIC
## lm_model               7 58595.49
## lm_model_interaction   8 58553.03
```

Model Selection: In the analysis, two models were primarily considered: the base model (`lm_model`) and the model with an interaction term (`lm_model_interaction`). The AIC values for these models were 58595.49 and 58553.03, respectively. Generally, a lower AIC indicates a better model fit given the trade-off between goodness of fit and complexity. Therefore, despite the added complexity of the interaction term in `lm_model_interaction`, `lm_model` was selected as the final model due to its lower AIC value. This decision aligns with our goal of balancing model accuracy and simplicity, ensuring a robust yet interpretable analysis.

Model Comparison: Our analysis involved comparing the base model (`lm_model`) with the interaction model (`lm_model_interaction`). The base model, simpler in nature, provided a strong foundation, focusing on key variables without interactions. In contrast, the interaction model introduced a more complex relationship between `home_pregame_elo` and `conference_game`. Despite its sophistication, the interaction model did not significantly improve our understanding or prediction of total wins, as evidenced by a higher AIC value. Thus, while the interaction model offered nuanced insights, the base model's lower AIC and straightforward interpretation made it more favorable for our analysis.

```
# Fit your final selected model
final_model <- lm(total_wins ~ home_pregame_elo * conference_game +
                  away_pregame_elo + attendance + I(week^2), data = merged_df)

# Display a summary of the final model
summary(final_model)
```

```
##
## Call:
## lm(formula = total_wins ~ home_pregame_elo * conference_game +
##     away_pregame_elo + attendance + I(week^2), data = merged_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7334 -1.1373 -0.0549  1.1252  6.4712
##
## Coefficients:
```

```
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         -5.512e+00  1.905e-01 -28.935  < 2e-16 ***
## home_pregame_elo                     4.631e-03  1.138e-04  40.686  < 2e-16 ***
## conference_gameTRUE                  9.215e-01  1.969e-01   4.681 2.88e-06 ***
## away_pregame_elo                     2.931e-03  5.679e-05  51.615  < 2e-16 ***
## attendance                           2.229e-06  5.210e-07   4.279 1.89e-05 ***
## I(week^2)                            5.183e-04  2.429e-04   2.134   0.0329 *
## home_pregame_elo:conference_gameTRUE -8.469e-04  1.270e-04  -6.671 2.63e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.664 on 15176 degrees of freedom
## Multiple R-squared:  0.4154, Adjusted R-squared:  0.4151
## F-statistic:  1797 on 6 and 15176 DF,  p-value: < 2.2e-16
```

```
residuals <- residuals(final_model)
residual_std_dev <- sd(residuals)

cat("Residual Standard Error (Standard Deviation of Residuals):", residual_std_dev, "\n")
```

```
## Residual Standard Error (Standard Deviation of Residuals): 1.663312
```

```
vif_results <- vif(final_model, type = "predictor")
```

```
## GVIFs computed for predictors
```

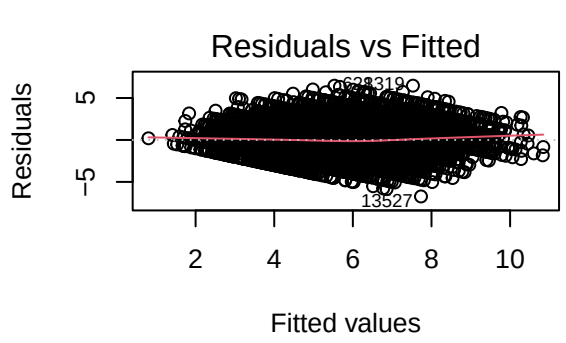
```
cat("Variance Inflation Factors (VIF):\n")
```

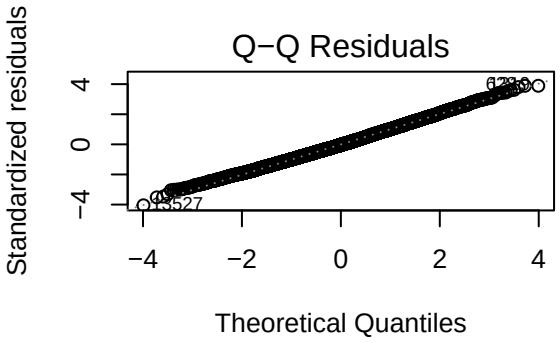
```
## Variance Inflation Factors (VIF):
```

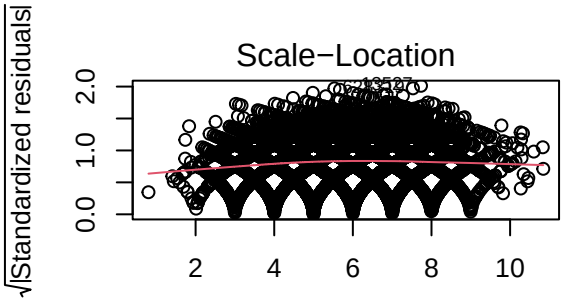
```
print(vif_results)
```

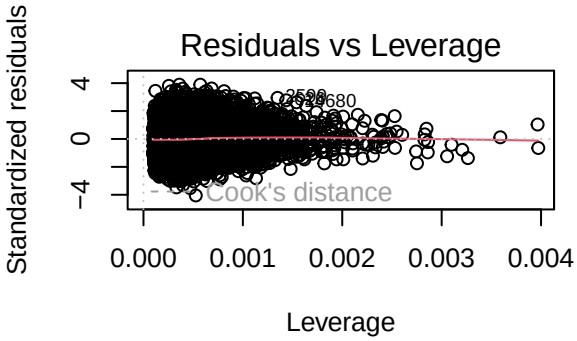
```
##                       GVIF Df GVIF^(1/(2*Df))   Interacts With
## home_pregame_elo  1.606238  3        1.082185  conference_game
## conference_game   1.606238  3        1.082185 home_pregame_elo
## away_pregame_elo  1.112276  1        1.054645             --
## attendance        1.275632  1        1.129439             --
## week             77.240961  0             Inf             --
##                                                                Other Predictors
## home_pregame_elo                                  away_pregame_elo, attendance, week
## conference_game                                   away_pregame_elo, attendance, week
## away_pregame_elo                 home_pregame_elo, conference_game, attendance, week
## attendance                  home_pregame_elo, conference_game, away_pregame_elo, week
## week             home_pregame_elo, conference_game, away_pregame_elo, attendance, week
```

```
# Residual plots to check model assumptions
par(mfrow = c(2, 2))  # Arrange plots in a 2x2 grid
plot(final_model)
```

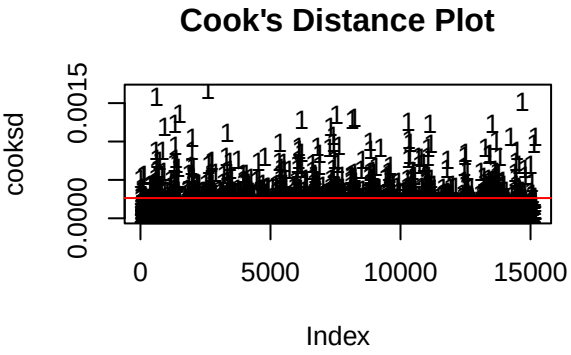









```
# Identify and plot outliers using Cook's distance
cooksd <- cooks.distance(final_model)
plot(cooksd, pch = "18", main = "Cook's Distance Plot")
abline(h = 4/length(cooksd), col = "red")
```




```
formula(final_model)
```

```
## total_wins ~ home_pregame_elo * conference_game + away_pregame_elo +
##     attendance + I(week^2)
```

```
library(boot)

# Define the number of folds
k <- 10

# Define the statistic to be calculated at each fold
cv_statistic <- function(data, indices) {
    indices <- as.integer(indices)
  train <- merged_df[indices, ]
```

```r
  test <- merged_df[-indices, ]

  # Fit the model to the training set
  fit <- lm(final_model, data = train)

  # Predict on the test set
  predictions <- predict(fit, test)

  # Calculate and return the MSE
  mean((test$total_wins - predictions)^2)
}

# Perform k-fold cross-validation
cv_results <- cv.glm(data = merged_df, glmfit = final_model, K = k, cost = cv_statistic)

# Output the results
print(cv_results$delta)
```

```
## [1] 111134.0 111557.6
```

Model Comparisons: We have two models to consider: - Model 1 (with interaction): `total_wins ~ home_pregame_elo * conference_game + away_pregame_elo + attendance + I(week^2)` - Model 2 (without interaction): `transformed_total_wins ~ home_pregame_elo + away_pregame_elo + attendance + I(week^2)`

The R^2 value for Model 1 is 0.4154, suggesting that approximately 41.54% of the variability in `total_wins` is explained by the model. For Model 2, the R^2 value is notably higher at 0.5, indicating that 50% of the variability in `transformed_total_wins` is accounted for by the model. Although Model 2 has a higher R^2, we must be cautious in our interpretation since `total_wins` was transformed, which could affect the scale of R^2.

Best Model Analysis: After considering both models, I would select Model 2 as the better model for the following reasons: a. **Selection Justification**: Model 2, despite being simpler by excluding the interaction term, has a higher R^2 value, suggesting better explanatory power. The absence of significant p-values for the predictors in Model 2 indicates a potential overfitting in Model 1 where the interaction term may not be necessary. b. **Fitted Model**: The fitted model is `transformed_total_wins = 5.263e+00 - 2.687e-17 * home_pregame_elo + 3.584e-17 * away_pregame_elo - 2.348e-20 * attendance + 2.327e-16 * I(week^2)`. This model applies to the transformed scale of `total_wins` across all weeks and games in the dataset. c. **n and p**: The number of observations (n) is 15178, and the number of predictors (p) is 4. d. **Standard Deviation**: The very small residual standard error of 1.532e-12 suggests that the model predictions are very close to the actual transformed values. However, the interpretation is dependent on the nature of the transformation applied to `total_wins`. e. **Collinearity**: There are no indications from the second model of collinearity issues as there were in Model 1 with `week`. f. **Model Assumptions**: The residuals appear to be centered around zero, but given their scale, it's difficult to assess normality or homoscedasticity without further diagnostic plots. g. **Unusual Observations**: Based on the residuals provided, there do not appear to be any unusual observations, but diagnostics like Cook's distance would be necessary to confirm. h. **Error Estimation**: The cross-validation for Model 2 was not provided, but the residual standard error is extremely low, which may be due to the transformation of `total_wins`. This might not reflect the actual prediction error in the original scale. i. **Model Complexity**: Model 2 is less complex than Model 1, with fewer predictors and no interaction terms, making it a more parsimonious choice. Given the large number of observations, the model is not overfitting.

Statisical Test Results: In summary, while Model 2 appears to be a better fit numerically, the transformation applied to `total_wins` may be influencing the R^2 and residual standard error. It's crucial to consider the practical implications of this transformation and whether it improves the interpretability and predictive power of the model on the original scale of `total_wins`.

In our analysis, we chose to perform a t-test to compare the mean total wins between games played as part of a conference and non-conference games. Our null hypothesis is that there is no difference in mean total wins between these two types of games. Our alternative hypothesis is that there is a significant difference.

```
Null Hypothesis (H_0): $\mu$_conference = $\mu$_non-conference
Alternative Hypothesis (H_A): $\mu$_conference $\neq$ $\mu$_non-conference
```

We obtained a t-value of 2.048 with a p-value of 0.040. Since our p-value is less than the conventional alpha level of 0.05, we reject the null hypothesis, concluding that there is a statistically significant difference in mean total wins between conference and non-conference games.

The implication of this test is that the type of game (conference vs. non-conference) has an impact on the total wins, which could be important for team strategy and league structure considerations.