Predicting NFL Win Totals

Linda Ungerboeck

Executive Summary:
This code builds and evaluates a linear regression model to predict NFL team wins (W) using historical data. It involves splitting the data into training and testing sets, preprocessing with a recipe, fitting the model, making predictions, and then assessing the model's performance using RMSE and R-squared. The residual plots help to visualize how well the model's predictions align with actual outcomes.

```r
#getting websites for win total lines for each season
page2023 <- read_html("https://www.cbssports.com/nfl/news/2023-nfl-win-totals-awards-odds-lines-picks-fo

page2022 <- read_html("https://www.cbssports.com/nfl/news/2022-nfl-win-totals-futures-odds-predictions-b

page2021 <- read_html("https://www.cbssports.com/nfl/news/2021-nfl-win-totals-odds-predictions-best-bets
```

```r
#creating variables for the win total tables
team2023 <- page2023 %>%
  html_nodes("td:nth-child(1)") %>%
  html_text()

line2023 <- page2023 %>%
  html_nodes("td:nth-child(2)") %>%
  html_text() %>%
  as.numeric()
```

```
## Warning in page2023 %>% html_nodes("td:nth-child(2)") %>% html_text() %>% : NAs
## introduced by coercion
```

```r
team2022 <- page2022 %>%
  html_nodes("td:nth-child(1)") %>%
  html_text()

line2022 <- page2022 %>%
  html_nodes("td:nth-child(2)") %>%
  html_text() %>%
  as.numeric()

team2021 <- page2021 %>%
  html_nodes("td:nth-child(1)") %>%
  html_text()

line2021 <- page2021 %>%
  html_nodes("td:nth-child(2)") %>%
  html_text() %>%
  as.numeric()
```

```r
#creating tables for win totals
lines2023 <- tibble(
  team = team2023,
  line = line2023
)

lines2023 <- lines2023 %>%
  filter(team != "Team")

lines2022 <- tibble(
  team = team2022,
  line = line2022
)

lines2022 <- lines2022 %>%
  filter(team != "Team")

lines2021 <- tibble(
  team = team2021,
  line = line2021
)

lines2021 <- lines2021 %>%
  filter(team != "Team")
```

```r
#changing team names in win total line tables to make sense
lines2023$team[lines2023$team == "Buffalo   Bills   "] <- "Buffalo Bills"
lines2023$team[lines2023$team == "New   England Patriots   "] <- "New England Patriots"
lines2023$team[lines2023$team == "New   York Jets   "] <- "New York Jets"
lines2023$team[lines2023$team == "Miami   Dolphins   "] <- "Miami Dolphins"
lines2023$team[lines2023$team == "Pittsburgh   Steelers   "] <- "Pittsburgh Steelers"
lines2023$team[lines2023$team == "Baltimore   Ravens   "] <- "Baltimore Ravens"
lines2023$team[lines2023$team == "Cleveland   Browns   "] <- "Cleveland Browns"
lines2023$team[lines2023$team == "Cincinnati   Bengals   "] <- "Cincinnati Bengals"
lines2023$team[lines2023$team == "Indianapolis   Colts   "] <- "Indianapolis Colts"
lines2023$team[lines2023$team == "Tennessee   Titans   "] <- "Tennessee Titans"
lines2023$team[lines2023$team == "Jacksonville   Jaguars   "] <- "Jacksonville Jaguars"
lines2023$team[lines2023$team == "Houston   Texans   "] <- "Houston Texans"
lines2023$team[lines2023$team == "Kansas   City Chiefs   "] <- "Kansas City Chiefs"
lines2023$team[lines2023$team == "Los   Angeles Chargers  "] <- "Los Angeles Chargers"
lines2023$team[lines2023$team == "Denver   Broncos   "] <- "Denver Broncos"
lines2023$team[lines2023$team == "Las   Vegas Raiders   "] <- "Las Vegas Raiders"
lines2023$team[lines2023$team == "Philadelphia   Eagles   "] <- "Philadelphia Eagles"
lines2023$team[lines2023$team == "New   York Giants   "] <- "New York Giants"
lines2023$team[lines2023$team == "Dallas   Cowboys   "] <- "Dallas Cowboys"
lines2023$team[lines2023$team == "Washington   Commanders   "] <- "Washington Commanders"
lines2023$team[lines2023$team == "Green   Bay Packers   "] <- "Green Bay Packers"
lines2023$team[lines2023$team == "Chicago   Bears   "] <- "Chicago Bears"
lines2023$team[lines2023$team == "Detroit   Lions   "] <- "Detroit Lions"
```

```
lines2023$team[lines2023$team == "Minnesota   Vikings   "] <- "Minnesota Vikings"
lines2023$team[lines2023$team == "Tampa   Bay Buccaneers   "] <- "Tamps Bay Buccaneers"
lines2023$team[lines2023$team == "Atlanta   Falcons   "] <- "Atlanta Falcons"
lines2023$team[lines2023$team == "New   Orleans Saints   "] <- "New Orleans Saints"
lines2023$team[lines2023$team == "Carolina   Panthers   "] <- "Carolina Panthers"
lines2023$team[lines2023$team == "San   Francisco 49ers   "] <- "San Francisco 49ers"
lines2023$team[lines2023$team == "Seattle   Seahawks   "] <- "Seattle Seahawks"
lines2023$team[lines2023$team == "Arizona   Cardinals   "] <- "Arizona Cardinals"
lines2023$team[lines2023$team == "Los   Angeles Rams   "] <- "Los Angeles Rams"

lines2022$team[lines2022$team == "Bills"] <- "Buffalo Bills"
lines2022$team[lines2022$team == "Patriots"] <- "New England Patriots"
lines2022$team[lines2022$team == "Jets"] <- "New York Jets"
lines2022$team[lines2022$team == "Dolphins"] <- "Miami Dolphins"
lines2022$team[lines2022$team == "Steelers"] <- "Pittsburgh Steelers"
lines2022$team[lines2022$team == "Ravens"] <- "Baltimore Ravens"
lines2022$team[lines2022$team == "Browns"] <- "Cleveland Browns"
lines2022$team[lines2022$team == "Bengals"] <- "Cincinnati Bengals"
lines2022$team[lines2022$team == "Colts"] <- "Indianapolis Colts"
lines2022$team[lines2022$team == "Titans"] <- "Tennessee Titans"
lines2022$team[lines2022$team == "Jaguars"] <- "Jacksonville Jaguars"
lines2022$team[lines2022$team == "Texans"] <- "Houston Texans"
lines2022$team[lines2022$team == "Chiefs"] <- "Kansas City Chiefs"
lines2022$team[lines2022$team == "Chargers"] <- "Los Angeles Chargers"
lines2022$team[lines2022$team == "Broncos"] <- "Denver Broncos"
lines2022$team[lines2022$team == "Raiders"] <- "Las Vegas Raiders"
lines2022$team[lines2022$team == "Eagles"] <- "Philadelphia Eagles"
lines2022$team[lines2022$team == "Giants"] <- "New York Giants"
lines2022$team[lines2022$team == "Cowboys"] <- "Dallas Cowboys"
lines2022$team[lines2022$team == "Commanders"] <- "Washington Commanders"
lines2022$team[lines2022$team == "Packers"] <- "Green Bay Packers"
lines2022$team[lines2022$team == "Bears"] <- "Chicago Bears"
lines2022$team[lines2022$team == "Lions"] <- "Detroit Lions"
lines2022$team[lines2022$team == "Vikings"] <- "Minnesota Vikings"
lines2022$team[lines2022$team == "Buccaneers"] <- "Tamps Bay Buccaneers"
lines2022$team[lines2022$team == "Falcons"] <- "Atlanta Falcons"
lines2022$team[lines2022$team == "Saints"] <- "New Orleans Saints"
lines2022$team[lines2022$team == "Panthers"] <- "Carolina Panthers"
lines2022$team[lines2022$team == "49ers"] <- "San Francisco 49ers"
lines2022$team[lines2022$team == "Seahawks"] <- "Seattle Seahawks"
lines2022$team[lines2022$team == "Cardinals"] <- "Arizona Cardinals"
lines2022$team[lines2022$team == "Rams"] <- "Los Angeles Rams"

lines2021$team[lines2021$team == "Bills"] <- "Buffalo Bills"
lines2021$team[lines2021$team == "Patriots"] <- "New England Patriots"
lines2021$team[lines2021$team == "Jets"] <- "New York Jets"
lines2021$team[lines2021$team == "Dolphins"] <- "Miami Dolphins"
lines2021$team[lines2021$team == "Steelers"] <- "Pittsburgh Steelers"
lines2021$team[lines2021$team == "Ravens"] <- "Baltimore Ravens"
lines2021$team[lines2021$team == "Browns"] <- "Cleveland Browns"
lines2021$team[lines2021$team == "Bengals"] <- "Cincinnati Bengals"
lines2021$team[lines2021$team == "Colts"] <- "Indianapolis Colts"
lines2021$team[lines2021$team == "Titans"] <- "Tennessee Titans"
```

```r
lines2021$team[lines2021$team == "Jaguars"] <- "Jacksonville Jaguars"
lines2021$team[lines2021$team == "Texans"] <- "Houston Texans"
lines2021$team[lines2021$team == "Chiefs"] <- "Kansas City Chiefs"
lines2021$team[lines2021$team == "Chargers"] <- "Los Angeles Chargers"
lines2021$team[lines2021$team == "Broncos"] <- "Denver Broncos"
lines2021$team[lines2021$team == "Raiders"] <- "Las Vegas Raiders"
lines2021$team[lines2021$team == "Eagles"] <- "Philadelphia Eagles"
lines2021$team[lines2021$team == "Giants"] <- "New York Giants"
lines2021$team[lines2021$team == "Cowboys"] <- "Dallas Cowboys"
lines2021$team[lines2021$team == "Washington"] <- "Washington Commanders"
lines2021$team[lines2021$team == "Packers"] <- "Green Bay Packers"
lines2021$team[lines2021$team == "Bears"] <- "Chicago Bears"
lines2021$team[lines2021$team == "Lions"] <- "Detroit Lions"
lines2021$team[lines2021$team == "Vikings"] <- "Minnesota Vikings"
lines2021$team[lines2021$team == "Buccaneers"] <- "Tamps Bay Buccaneers"
lines2021$team[lines2021$team == "Falcons"] <- "Atlanta Falcons"
lines2021$team[lines2021$team == "Saints"] <- "New Orleans Saints"
lines2021$team[lines2021$team == "Panthers"] <- "Carolina Panthers"
lines2021$team[lines2021$team == "49ers"] <- "San Francisco 49ers"
lines2021$team[lines2021$team == "Seahawks"] <- "Seattle Seahawks"
lines2021$team[lines2021$team == "Cardinals"] <- "Arizona Cardinals"
lines2021$team[lines2021$team == "Rams"] <- "Los Angeles Rams"
```

```r
#importing offense and defense stats and the records for each year

defense23 <- read.csv("/Users/lindaungerbock/Downloads/X2023defense.csv",skip = 1)
offense23 <- read.csv("/Users/lindaungerbock/Downloads/X2023offense.csv",skip = 1)

defense22 <- read.csv("/Users/lindaungerbock/Downloads/X2022defense.csv",skip = 1)
offense22 <- read.csv("/Users/lindaungerbock/Downloads/X2022offense.csv",skip = 1)

defense21 <- read.csv("/Users/lindaungerbock/Downloads/X2021defense.csv",skip = 1)
offense21 <- read.csv("/Users/lindaungerbock/Downloads/X2021offense.csv",skip = 1)

AFCrecords23 <- read.csv("/Users/lindaungerbock/Downloads/X2023AFCRecords.csv") %>%
  select("Tm", "W", "L")
NFCrecords23 <- read.csv("/Users/lindaungerbock/Downloads/X2023NFCRecords.csv") %>%
  select("Tm", "W", "L")
AFCrecords23$Tm[AFCrecords23$Tm == "Buffalo Bills*"] <- "Buffalo Bills"
AFCrecords23$Tm[AFCrecords23$Tm == "Miami Dolphins+"] <- "Miami Dolphins"
AFCrecords23$Tm[AFCrecords23$Tm == "Cleveland Browns+"] <- "Cleveland Browns"
AFCrecords23$Tm[AFCrecords23$Tm == "Baltimore Ravens*"] <- "Baltimore Ravens"
AFCrecords23$Tm[AFCrecords23$Tm == "Pittsburgh Steelers+"] <- "Pittsburgh Steelers"
AFCrecords23$Tm[AFCrecords23$Tm == "Kansas City Chiefs*"] <- "Kansas City Chiefs"
AFCrecords23$Tm[AFCrecords23$Tm == "Houston Texans*"] <- "Houston Texans"
NFCrecords23$Tm[NFCrecords23$Tm == "Philadelphia Eagles+"] <- "Philadelphia Eagles"
NFCrecords23$Tm[NFCrecords23$Tm == "Dallas Cowboys*"] <- "Dallas Cowboys"
NFCrecords23$Tm[NFCrecords23$Tm == "Detroit Lions*"] <- "Detroit Lions"
NFCrecords23$Tm[NFCrecords23$Tm == "Green Bay Packers+"] <- "Green Bay Packers"
NFCrecords23$Tm[NFCrecords23$Tm == "Tampa Bay Buccaneers*"] <- "Tampa Bay Buccaneers"
NFCrecords23$Tm[NFCrecords23$Tm == "San Francisco 49ers*"] <- "San Francisco 49ers"
NFCrecords23$Tm[NFCrecords23$Tm == "Los Angeles Rams+"] <- "Los Angeles Rams"
```

```r
records23 <- AFCrecords23 %>%
  rbind(NFCrecords23) %>%
  rename(team = Tm)


AFCrecords22 <- read.csv("/Users/lindaungerbock/Downloads/X2022AFCRecords.csv") %>%
  select("Tm", "W", "L")
NFCrecords22 <- read.csv("/Users/lindaungerbock/Downloads/X2022NFCRecords.csv") %>%
  select("Tm", "W", "L")
AFCrecords22$Tm[AFCrecords22$Tm == "Buffalo Bills*"] <- "Buffalo Bills"
AFCrecords22$Tm[AFCrecords22$Tm == "Miami Dolphins+"] <- "Miami Dolphins"
AFCrecords22$Tm[AFCrecords22$Tm == "Cincinnati Bengals*"] <- "Cincinnati Bengals"
AFCrecords22$Tm[AFCrecords22$Tm == "Baltimore Ravens+"] <- "Baltimore Ravens"
AFCrecords22$Tm[AFCrecords22$Tm == "Jacksonville Jaguars*"] <- "Jacksonville Jaguars"
AFCrecords22$Tm[AFCrecords22$Tm == "Kansas City Chiefs*"] <- "Kansas City Chiefs"
AFCrecords22$Tm[AFCrecords22$Tm == "Los Angeles Chargers+"] <- "Los Angeles Chargers"
NFCrecords22$Tm[NFCrecords22$Tm == "Philadelphia Eagles*"] <- "Philadelphia Eagles"
NFCrecords22$Tm[NFCrecords22$Tm == "Dallas Cowboys+"] <- "Dallas Cowboys"
NFCrecords22$Tm[NFCrecords22$Tm == "New York Giants+"] <- "New York Giants"
NFCrecords22$Tm[NFCrecords22$Tm == "Minnesota Vikings*"] <- "Minnesota Vikings"
NFCrecords22$Tm[NFCrecords22$Tm == "Tampa Bay Buccaneers*"] <- "Tampa Bay Buccaneers"
NFCrecords22$Tm[NFCrecords22$Tm == "San Francisco 49ers*"] <- "San Francisco 49ers"
NFCrecords22$Tm[NFCrecords22$Tm == "Seattle Seahawks+"] <- "Seattle Seahawks"

records22 <- AFCrecords22 %>%
  rbind(NFCrecords22) %>%
  rename(team = Tm)



AFCrecords21 <- read.csv("/Users/lindaungerbock/Downloads/X2021AFCRecords.csv") %>%
  select("Tm", "W", "L")
NFCrecords21 <- read.csv("/Users/lindaungerbock/Downloads/X2021NFCRecords.csv") %>%
  select("Tm", "W", "L")
AFCrecords21$Tm[AFCrecords21$Tm == "Buffalo Bills*"] <- "Buffalo Bills"
AFCrecords21$Tm[AFCrecords21$Tm == "New England Patriots+"] <- "New England Patriots"
AFCrecords21$Tm[AFCrecords21$Tm == "Cincinnati Bengals*"] <- "Cincinnati Bengals"
AFCrecords21$Tm[AFCrecords21$Tm == "Pittsburgh Steelers+"] <- "Pittsburgh Steelers"
AFCrecords21$Tm[AFCrecords21$Tm == "Tennessee Titans*"] <- "Tennessee Titans"
AFCrecords21$Tm[AFCrecords21$Tm == "Kansas City Chiefs*"] <- "Kansas City Chiefs"
AFCrecords21$Tm[AFCrecords21$Tm == "Los Angeles Chargers+"] <- "Los Angeles Chargers"
AFCrecords21$Tm[AFCrecords21$Tm == "Las Vegas Raiders+"] <- "Las Vegas Raiders"
NFCrecords21$Tm[NFCrecords21$Tm == "Dallas Cowboys*"] <- "Dallas Cowboys"
NFCrecords21$Tm[NFCrecords21$Tm == "Philadelphia Eagles+"] <- "Philadelphia Eagles"
NFCrecords21$Tm[NFCrecords21$Tm == "Green Bay Packers*"] <- "Green Bay Packers"
NFCrecords21$Tm[NFCrecords21$Tm == "Tampa Bay Buccaneers*"] <- "Tampa Bay Buccaneers"
NFCrecords21$Tm[NFCrecords21$Tm == "Los Angeles Rams*"] <- "Los Angeles Rams"
NFCrecords21$Tm[NFCrecords21$Tm == "San Francisco 49ers+"] <- "San Francisco 49ers"
NFCrecords21$Tm[NFCrecords21$Tm == "Arizona Cardinals+"] <- "Arizona Cardinals"
NFCrecords21$Tm[NFCrecords21$Tm == "Washington Football Team"] <- "Washington Commanders"

records21 <- AFCrecords21 %>%
  rbind(NFCrecords21) %>%
  rename(team = Tm)
```

```
str(offense23)
```

```
## 'data.frame':    35 obs. of  28 variables:
##  $ Rk      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Tm      : chr  "Dallas Cowboys" "Miami Dolphins" "San Francisco 49ers" "Baltimore Ravens" ...
##  $ G       : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ PF      : num  509 496 491 483 461 451 433 404 402 396 ...
##  $ Yds     : num  6317 6822 6773 6296 6712 ...
##  $ Ply     : num  1122 1053 1024 1076 1137 ...
##  $ Y.P     : num  5.6 6.5 6.6 5.9 5.9 5.7 5.4 5.6 5.1 5.2 ...
##  $ TO      : num  16 25 18 19 23 28 28 18 18 22 ...
##  $ FL      : num  6 10 6 12 11 10 12 5 7 12 ...
##  $ X1stD   : num  385 360 383 360 375 381 377 351 337 324 ...
##  $ Cmp     : num  428 393 336 328 408 385 369 361 406 355 ...
##  $ Att     : num  614 566 491 494 606 579 563 583 606 574 ...
##  $ Yds.1   : num  4397 4514 4384 3635 4401 ...
##  $ TD      : num  36 30 33 27 30 29 24 26 28 18 ...
##  $ Int     : num  10 15 12 7 12 18 16 13 11 10 ...
##  $ NY.A    : num  6.7 7.6 8.4 6.8 6.9 6.9 6.4 6.6 6.2 6 ...
##  $ X1stD.1: num  229 223 207 180 228 199 197 206 199 178 ...
##  $ Att.1   : num  468 456 499 541 500 512 510 477 480 479 ...
##  $ Yds.2   : num  1920 2308 2389 2661 2311 ...
##  $ TD.1    : num  14 27 27 26 27 22 22 18 13 19 ...
##  $ Y.A     : num  4.1 5.1 4.8 4.9 4.6 4.3 4.3 4.3 3.6 4.3 ...
##  $ X1stD.2: num  113 113 147 145 124 158 149 110 112 113 ...
##  $ Pen     : num  115 97 101 102 97 106 95 89 96 95 ...
##  $ Yds.3   : num  964 767 933 955 843 883 785 720 846 685 ...
##  $ X1stPy : num  43 24 29 35 23 24 31 35 26 33 ...
##  $ Sc.     : num  50.3 43.5 45.3 43.1 40.6 41.4 42.9 41.1 36.6 35.7 ...
##  $ TO.     : num  8.9 13.4 10.1 9.6 11.8 14.9 15.3 9.2 8.2 9.2 ...
##  $ EXP     : num  193 129 269 139 191 ...
```

```
#pulling only the team stats we want to use in the model
key_offense23 <- offense23 %>%
  select("Tm", "PF", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pf = PF, oyds = Yds, o_plays = Ply, o_yds_play = `Y.P`,
         turnovers = TO, o_fd = `X1stD`, o_score_pct = `Sc.`)


key_defense23 <- defense23 %>%
  select("Tm", "PA", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pa = PA, dyds = Yds, d_plays = Ply, d_yds_play = `Y.P`,
         takeaways = TO, d_fd = `X1stD`, d_score_pct = `Sc.`)

key_offense22 <- offense22 %>%
  select("Tm", "PF", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pf = PF, oyds = Yds, o_plays = Ply, o_yds_play = `Y.P`,
         turnovers = TO, o_fd = `X1stD`, o_score_pct = `Sc.`)


key_defense22 <- defense22 %>%
  select("Tm", "PA", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pa = PA, dyds = Yds, d_plays = Ply, d_yds_play = `Y.P`,
         takeaways = TO, d_fd = `X1stD`, d_score_pct = `Sc.`)

key_offense21 <- offense21 %>%
  select("Tm", "PF", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pf = PF, oyds = Yds, o_plays = Ply, o_yds_play = `Y.P`,
         turnovers = TO, o_fd = `X1stD`, o_score_pct = `Sc.`)
key_offense21$team[key_offense21$team == "Washington Football Team"] <- "Washington Commanders"

key_defense21 <- defense21 %>%
  select("Tm", "PA", "Yds", "Ply", "Y.P", "TO", "X1stD", "Sc.") %>%
  rename(team = Tm, pa = PA, dyds = Yds, d_plays = Ply, d_yds_play = `Y.P`,
         takeaways = TO, d_fd = `X1stD`, d_score_pct = `Sc.`)
key_defense21$team[key_defense21$team == "Washington Football Team"] <- "Washington Commanders"
```

```
#combining offense and defense stats for each year and then combining all three years into one table
stats_2023 <- left_join(records23, key_offense23, by = "team") %>%
  left_join(key_defense23, by = "team") %>%
  left_join(lines2023, by = "team") %>%
  mutate(year = 2023) %>%
  na.omit(stats_2023)

stats_2022 <- left_join(records22, key_offense22, by = "team") %>%
  left_join(key_defense22, by = "team") %>%
  left_join(lines2022, by = "team") %>%
  mutate(year = 2022) %>%
  na.omit(stats_2023)

stats_2021 <- left_join(records21, key_offense21, by = "team") %>%
  left_join(key_defense21, by = "team") %>%
  left_join(lines2021, by = "team") %>%
  mutate(year = 2021) %>%
  na.omit(stats_2023)

stats <- rbind(stats_2023, stats_2022, stats_2021) %>%
  mutate(ppg = pf/17, pag = pa/17)
```

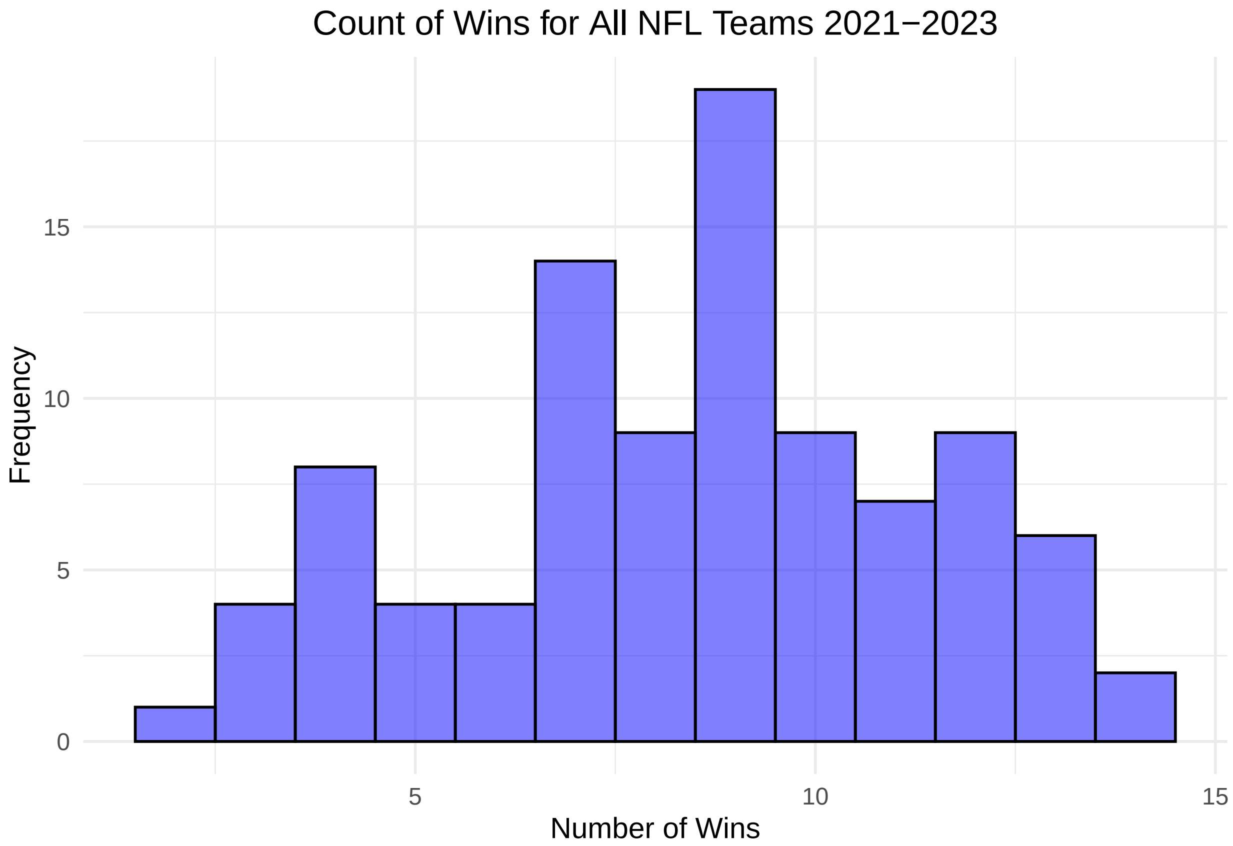
```
records23$W <- as.numeric(as.character(records23$W))
records22$W <- as.numeric(as.character(records22$W))
records21$W <- as.numeric(as.character(records21$W))

all_records <- records23 %>%
  rbind(records22) %>%
  rbind(records21)
```

saving data to use in python

```
# Save the all_records data frame to a CSV file
write.csv(all_records, "all_records.csv", row.names = FALSE)
write.csv(stats, "stats_combined.csv", row.names = FALSE)
```

Before fitting the model, we analyze the distribution of wins (W) in the 2021-2023 NFL seasons. The histogram below shows the count of wins across 96 data points, revealing a roughly normal distribution with nine wins being the most common. This aligns with the fact that nine wins is approximately half of the 17-game season. Given this normal distribution, no data transformation is needed, and a linear regression model is suitable

```r
ggplot(all_records, aes(x = W)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.5) +
  labs(title = "Count of Wins for All NFL Teams 2021-2023", x = "Number of Wins", y = "Frequency") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```



Predict future win totals based on historical data using linear regression

A linear regression model was initially fitted to all variables, with the data split into 75% training and 25% testing sets. The training set produced an intercept of 42.57—far exceeding the season's maximum wins of 17—and many predictors were not significant at common alpha levels, suggesting overfitting. Additionally, several offensive and defensive predictors behaved counterintuitively, further indicating model issues. Despite this, the model showed a low RMSE of about 1 and a relatively high R-squared of 0.85 on the training data, suggesting good initial performance.

```r
#Split the data into training (75%) and testing (25%)
set.seed(1122)
stats_split <- initial_split(stats)
stats_train <- training(stats_split)
stats_test <- testing(stats_split)
```

Splitting the data into a training set (75%) and a testing set (25%)

```
#Set to linear model
stats_mod <- linear_reg() %>%
  set_engine("lm")
```

Set up a linear regression model using the lm engine and the model will predict the W variable (wins).

```
#Create a recipe
stats_rec <- recipe(W ~ ., data = stats_train) %>%
  step_rm(L, year, pf, pa) %>%
  update_role(team, new_role = "id")
```

Creating a recipe. A recipe defines the data processing steps for the model: W ~ .: This specifies that you're predicting the W variable using all other variables in the dataset (.). step_rm(L, year, pf, pa): Removes unnecessary variables (L, year, pf, and pa) from the recipe that aren't needed for the prediction. update_role(team, new_role = "id"): Sets team as an identifier, meaning it's not included as a predictor but just for reference.

```
#Build a workflow for fitting the model
stats_wflow <- workflow() %>%
  add_model(stats_mod) %>%
  add_recipe(stats_rec)
```

```
#Fit the model to training data
stats_fit <- stats_wflow %>%
  fit(data = stats_train)

tidy(stats_fit)
```

```
## # A tibble: 16 x 5
##    term        estimate std.error statistic p.value
##    <chr>          <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept) 42.6      39.8         1.07  0.290
##  2 oyds         0.00198   0.00480     0.412 0.682
##  3 o_plays     -0.0138    0.0263     -0.526 0.601
##  4 o_yds_play  -2.50      5.08       -0.491 0.625
##  5 turnovers   -0.131     0.0444     -2.96  0.00457
##  6 o_fd         0.0294    0.0178      1.65  0.106
##  7 o_score_pct  0.111     0.0809      1.37  0.177
##  8 dyds         0.00676   0.00519     1.30  0.199
##  9 d_plays     -0.0161    0.0278     -0.582 0.563
## 10 d_yds_play  -7.58      5.56       -1.36  0.179
## 11 takeaways    0.0441    0.0426      1.04  0.305
## 12 d_fd        -0.0411    0.0163     -2.52  0.0147
## 13 d_score_pct  0.0386    0.0952      0.405 0.687
## 14 line         0.246     0.103       2.38  0.0212
## 15 ppg          0.0958    0.164       0.584 0.562
## 16 pag         -0.0926    0.169      -0.549 0.585
```

```
stats_train_pred <- predict(stats_fit,
                                new_data = stats_train) %>%
  bind_cols(stats_train %>% select(W, team, year, line)) %>%
  mutate(residual = W - .pred)
```

```
## # A tibble: 16 x 5
##    term        estimate std.error statistic p.value
##    <chr>          <dbl>     <dbl>     <dbl>   <dbl>
##  1 (Intercept) 42.6      39.8          1.07  0.290
##  2 oyds         0.00198   0.00480      0.412 0.682
##  3 o_plays     -0.0138    0.0263      -0.526 0.601
##  4 o_yds_play  -2.50      5.08        -0.491 0.625
##  5 turnovers   -0.131     0.0444      -2.96  0.00457
##  6 o_fd         0.0294    0.0178       1.65  0.106
##  7 o_score_pct  0.111     0.0809       1.37  0.177
##  8 dyds         0.00676   0.00519      1.30  0.199
##  9 d_plays     -0.0161    0.0278      -0.582 0.563
## 10 d_yds_play  -7.58      5.56        -1.36  0.179
## 11 takeaways    0.0441    0.0426       1.04  0.305
## 12 d_fd        -0.0411    0.0163      -2.52  0.0147
## 13 d_score_pct  0.0386    0.0952       0.405 0.687
## 14 line         0.246     0.103        2.38  0.0212
## 15 ppg          0.0958    0.164        0.584 0.562
## 16 pag         -0.0926    0.169       -0.549 0.585
```

```r
rmse(stats_train_pred,
     truth = W,
     estimate = .pred)
```

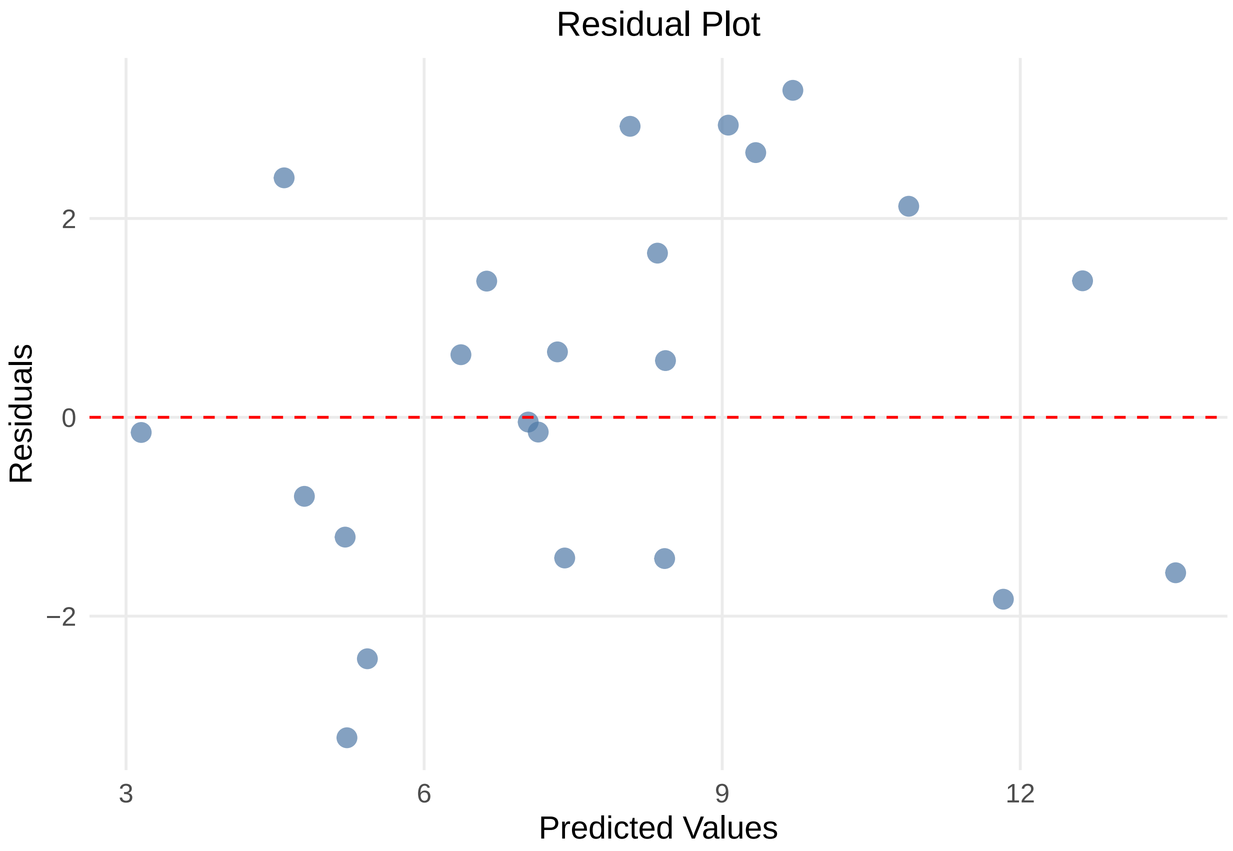
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        1.01
```

```r
rsq(stats_train_pred,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.853
```

The model, though likely overfit, was used to predict the testing dataset. The RMSE increased slightly to 1.88, and R-squared dropped to 0.73. The residual plot shows most predictions were within two wins of the actual values, indicating a reasonably accurate estimate.

```r
#Make predictions with test data
stats_test_pred <- predict(stats_fit,
                            new_data = stats_test) %>%
  bind_cols(stats_test %>% select(W, team, year, line)) %>%
  mutate(residual = W - .pred)

ggplot(stats_test_pred, aes(x = .pred, y = residual)) +
  geom_point(color = "#4e79a7", alpha = 0.7, size = 3) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Predicted Values", y = "Residuals", title = "Residual Plot") +
  theme_minimal() +  # Minimalist theme
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        legend.position = "none")
```



```
rmse(stats_test_pred,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        1.88
```

```
rsq(stats_test_pred,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.730
```

To address overfitting, many predictors were removed to improve model quality. While the initial model suggested turnovers, first downs allowed, and win total line as significant predictors, points per game and points allowed per game were intuitively selected along with win total line. This refinement yielded a well-fitting model with meaningful coefficients and small p-values. The intercept, now 4.9, falls within a realistic range of 0 to 16. Points per game positively correlates with wins, while points allowed per game negatively correlates, aligning with expectations. Coefficients and intercept details are shown below

```
stats_rec2 <- recipe(W ~ ppg + pag + line, data = stats_train)

stats_wflow2 <- workflow() %>%
  add_model(stats_mod) %>%
  add_recipe(stats_rec2)

stats_fit2 <- stats_wflow2 %>%
  fit(data = stats_train)
tidy(stats_fit2)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    4.92     2.08        2.37 2.09e- 2
## 2 ppg            0.433    0.0409     10.6  8.34e-16
## 3 pag           -0.336    0.0605     -5.56 5.54e- 7
## 4 line           0.159    0.100       1.59 1.18e- 1
```

```
stats_train_pred2 <- predict(stats_fit2, new_data = stats_train) %>%
  bind_cols(stats_train %>% select(W, team, year, line))%>%
  mutate(residual = W - .pred)

glance(stats_fit2)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.792         0.783  1.24      82.7 3.81e-22     3  -111.  231.  243.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
rmse(stats_train_pred2,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        1.20
```

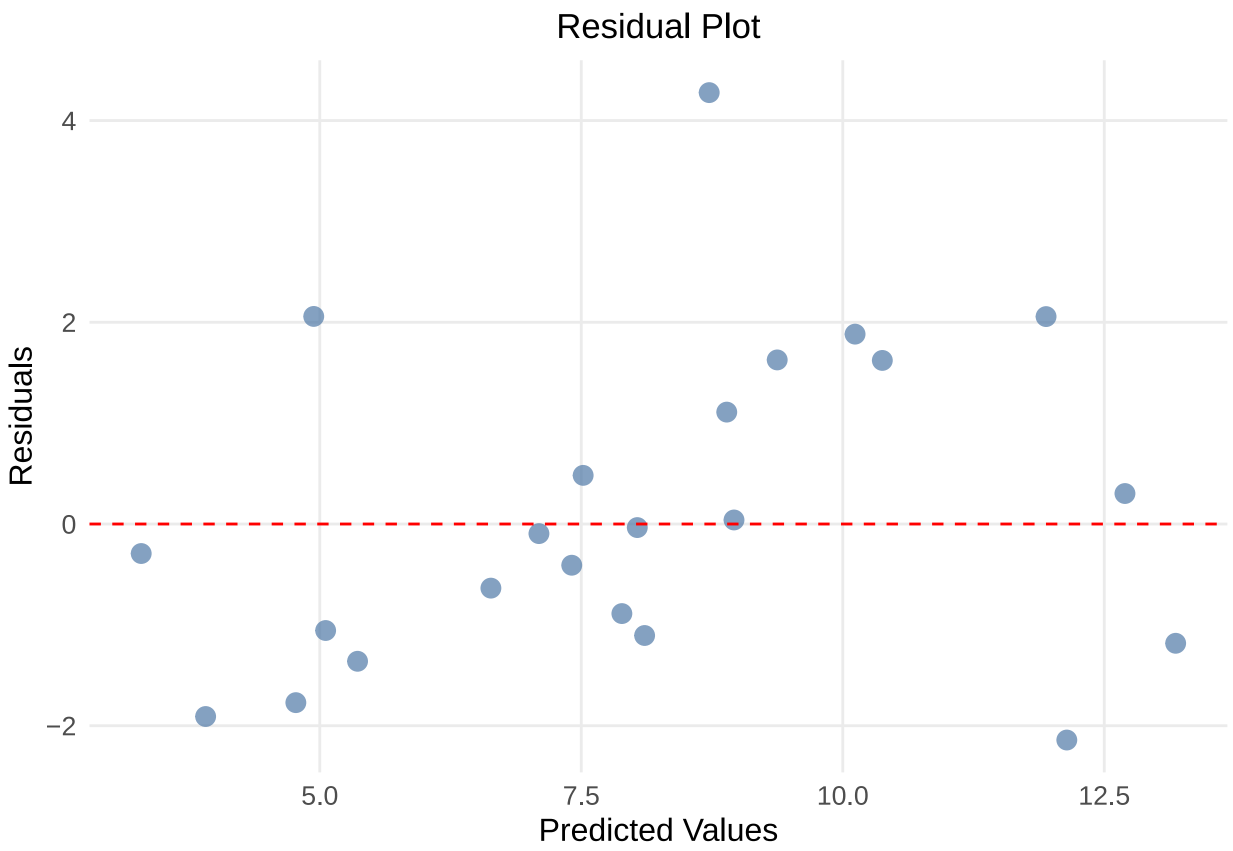
```
rsq(stats_train_pred2,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.792
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    4.92     2.08        2.37 2.09e- 2
## 2 ppg            0.433    0.0409     10.6  8.34e-16
## 3 pag           -0.336    0.0605     -5.56 5.54e- 7
## 4 line           0.159    0.100       1.59 1.18e- 1
```

As expected, removing predictors slightly increased the RMSE to 1.2 and decreased the R-squared to 0.79, but these values remain strong, with predictions typically within one win of actual totals. On the testing dataset, performance was consistent, with an RMSE of 1.5 and an improved R-squared of 0.81. The residual plot below highlights the differences between predicted and actual wins in the training set, showing most predictions within two wins of the true values.

```
stats_test_pred2 <- predict(stats_fit2,
                            new_data = stats_test) %>%
  bind_cols(stats_test %>% select(W, team, year, line)) %>%
  mutate(residual = W - .pred)

ggplot(stats_test_pred2, aes(x = .pred, y = residual)) +
  geom_point(color = "#4e79a7", alpha = 0.7, size = 3) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Predicted Values", y = "Residuals", title = "Residual Plot") +
  theme_minimal() +  # Minimalist theme
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        legend.position = "none")
```



```
rmse(stats_test_pred2,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        1.55
```

```
rsq(stats_test_pred2,
     truth = W,
     estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rsq     standard       0.814
```