Unlocking Victory: A Statistical Exploration of Baseball's Complexity

Nathan Jenks

2024-02-04

Setup

Packages

Introduction

Baseball, often regarded as a straightforward game, has witnessed a profound transformation with the integration of sabermetrics. This infusion of advanced statistical methods has elevated the complexity of the sport. The proliferation of intricate formulas and novel statistics each season prompts reflection: has baseball ventured too deeply into the realm of statistics, and if so, what truly constitutes the path to victory?

This statistical investigation sets out to address a fundamental question: What truly contributes to a team's success on the baseball field? Leveraging data sourced from Retrosheets, a comprehensive repository covering every event in MLB games from 1970 to 2022, this study endeavors to identify the factors that may, or may not, play a pivotal role in securing victories. Additionally, this study aims to offer a practical and insightful 'game-winning formula' for baseball enthusiasts, players, managers, and front-office executives—equipping them with strategies to optimize their win totals and vie for championships.

Acknowledging the irony in conducting a statistical study on metrics while questioning the extent of statistical influence, I invite readers to accompany me on this journey of exploration. By navigating through the intricacies of the data, I aim to strike a nuanced balance between statistical insights and the essence of the game itself.

Adding Data/Data Cleaning

```
appearance_file_list <- list.files(path = "~/Documents/Portfolio Files/BaseballData/Appearance Files/", full.name
s = TRUE)

allplayers_data <- data.frame()

for (files in appearance_file_list) {
  alldata <- read.csv(files, colClasses = "character")

  allplayers_data <- bind_rows(allplayers_data, alldata)
}
event_file_list <- list.files(path = "~/Documents/Portfolio Files/BaseballData/Event Files/", full.names = TRUE)

allevent_data <- data.frame()

for (file in event_file_list) {
  alldata <- read.csv(file, colClasses = "character")

  allevent_data <- bind_rows(allevent_data, alldata)
}

allevent_data$event_id <- as.double(allevent_data$event_id)
allevent_data$home_score <- as.double(allevent_data$home_score)
allevent_data$vis_score <- as.double(allevent_data$vis_score)

allplayers_data <- allplayers_data %>%
  mutate(home_away = ifelse(substr(game_id, start = 1, stop = 3) == team_id, "Home", "Away")) %>%
  mutate(final_id = paste(id, game_id, sep = ""))

allevent_data <- allevent_data %>%
  dplyr::select(
    game_id, event_id, batting_team, inning, outs, balls, strikes, pitch_seq, vis_score, home_score,
    batter_id, batter_hand, pitcher_id, pitcher_hand, event_scoring, leadoff, pinch_hit,
    batt_def_pos, batt_lineup_pos, event_type, batter_event, ab, hit_val, sac_hit, sac_fly,
    event_outs, dp, tp, rbi, wild_pitch, passed_ball, fielded_by, batted_ball_type, bunt,
    foul_ground, hit_location, num_err, sb_run_1b, sb_run_2b, sb_run_3b, start_game, end_game,
    run_1b, run_2b, run_3b
  )

columns_to_convertTF <- c("batter_event", "ab", "sac_hit", "sac_fly", "dp", "tp", "wild_pitch", "passed_ball", "b
unt", "foul_ground", "sb_run_1b", "sb_run_2b", "sb_run_3b", "start_game", "end_game")
columns_to_convertLR <- c("batter_hand", "pitcher_hand")

allevent_data <- allevent_data %>%
  mutate(across(all_of(columns_to_convertTF), ~ifelse(. == "t", 1, ifelse(. == "f", 0, 0)))) %>%
  mutate(across(all_of(columns_to_convertLR), ~ifelse(. == "l", 1, ifelse(. == "r", 0, 0)))) %>%
  type.convert(allevent_data, as.is = TRUE, na.strings = "NA")

salary <- read_excel(path = "~/Documents/Portfolio Files/BaseballData/Salary File/Salary.xls")
```

Win/Loss Record per Team per Year

After compiling the essential data, my initial focus was to explore the potential shift towards a 'pay-to-win' dynamic within Major League Baseball. This inquiry stems from the notable surge in large contracts and the absence of a salary cap. The analysis will involve assessing each team's win percentage percentile annually, followed by a comparison through a straightforward linear regression model against both the total salary percentile and the average salary percentile.

```
allevent_data <- allevent_data %>%
  mutate(final_id = paste(batter_id, game_id, sep = ""))

allplayers_data <- allplayers_data %>%
  filter(field_pos != "ump_hp" & field_pos != "ump_1b" & field_pos != "ump_2b" & field_pos != "ump_3b" & field_po
s != "ump_lf" & field_pos != "ump_rf" & field_pos != "manager" & field_pos != "player_manager" & field_pos != "")
%>%
  mutate(year = substr(game_id, start = 4, stop = 7))

game_mapping_table <- allplayers_data %>%
  distinct(game_id, team_id) %>%
  mutate(home = ifelse(team_id == substr(game_id, 1, 3), "home_team", "away_team" )) %>%
  pivot_wider(names_from = home, values_from = team_id) %>%
  mutate(year = substr(game_id, start = 4, stop = 7))

all_datajoin_base <- left_join(allevent_data, game_mapping_table, by = "game_id")

all_datajoin <- all_datajoin_base %>%
  group_by(game_id) %>%
  filter(event_id == max(event_id)) %>%
  mutate(id = batter_id) %>%
  dplyr::select(year, id, game_id, event_id, vis_score, home_score, event_scoring, batting_team, home_team, away_
team)

all_data_long <- all_datajoin %>%
  mutate(
    vis_score = ifelse(home_score == vis_score & grepl("-H", event_scoring) & batting_team == 0, vis_score + 1, v
is_score),
    home_score = ifelse(home_score == vis_score & grepl("-H", event_scoring) & batting_team == 1, home_score + 1,
home_score)
  ) %>%
  mutate(win_team = ifelse(home_score > vis_score, home_team, away_team),
         lose_team = ifelse(home_score > vis_score, away_team, home_team)) %>%
  dplyr::select(year, home_team, away_team, win_team, lose_team) %>%
  pivot_longer(cols = c(home_team, away_team, win_team, lose_team),
               names_to = "team_type",
               values_to = "team")%>%
  mutate(result = ifelse(team_type %in% c("home_team", "away_team"), "game_played", team_type),
         result = ifelse(result == "win_team", "win", result),
         result = ifelse(result == "loss_team", "loss", result)) %>%
  distinct(game_id, year, team_type, team, result, .keep_all = TRUE)

# Summarize the win/loss records
mlb_summary <- all_data_long %>%
  group_by(year, team, result) %>%
  summarise(count = n()) %>%
  pivot_wider(names_from = result, values_from = count, values_fill = 0) %>%
  mutate(win_perc = win / game_played) %>%
  group_by(year) %>%
  mutate(win_percentile = ecdf(win_perc)(win_perc) * 100,
         joinkey = paste(team, year, sep = ""))
```

Salary-Based Models

Model 1: Percentile of Total Salary = Percentile of Number of Wins

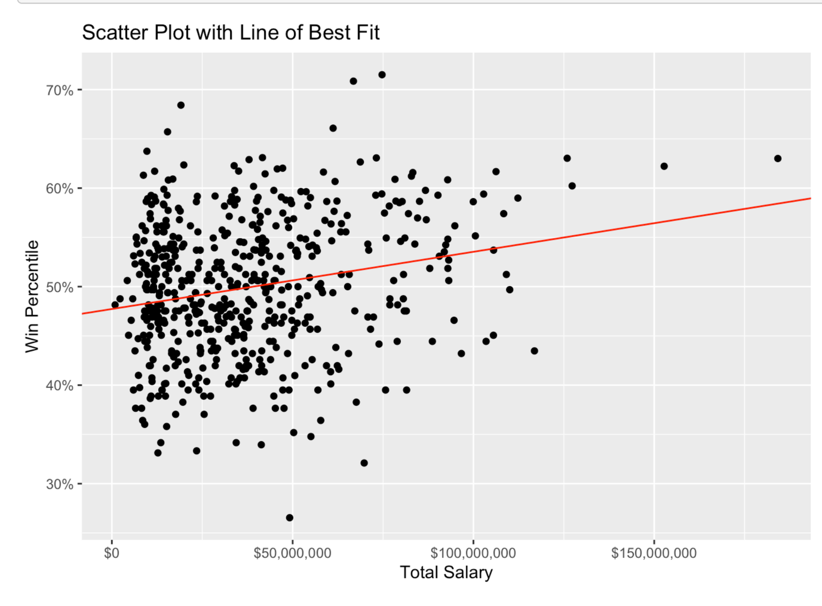
When examining the correlation between the total salary percentile of an MLB team and their win percentile per year, a subtle positive linear relationship becomes apparent, although it is not particularly robust. This observation is supported by both statistical outputs; the R-squared value, which is 0.22165, and the accompanying chart. The chart illustrates a common trend where most teams cluster within a similar range. Notably, only a handful of teams deviate by allocating significant resources to multiple large contracts, and in these cases, the investment appears to yield positive results.

It is important to emphasize, however, that the overall correlation lacks the strength necessary to draw conclusive statements. Higher total salaries do not consistently translate to higher win totals across all teams.

```
sum_salaries <- salary %>%
  group_by(teamID, year) %>%
  summarize(total_salary = sum(salary))

salary_percentile <- sum_salaries %>%
  group_by(year) %>%
  mutate(salary_percentile = ecdf(total_salary)(total_salary) * 100,
         joinkey = paste(teamID, year, sep = ""))

percentile_combined <- left_join(salary_percentile, mlb_summary, by = "joinkey")

final_percentile_data <- na.omit(percentile_combined)

cor(final_percentile_data$total_salary, final_percentile_data$win_perc)
```

```
## [1] 0.2216514
```

```
total_salary_model <- lm(win_perc~total_salary, data = final_percentile_data)

final_percentile_data %>%
  ggplot(aes(x = total_salary, y = win_perc)) +
  geom_point() +
  geom_abline(intercept = coef(total_salary_model)[1], slope = coef(total_salary_model)[2], color = "red") +
  ggtitle("Scatter Plot with Line of Best Fit") +
  xlab("Total Salary") +
  ylab("Win Percentile") +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(labels = scales::dollar_format(scale = 1))
```



Model 2: Percentile of Average Salary = Percentile of Number of Wins

When examining the correlation between the average salary percentile of an MLB team and its corresponding win percentile for a given year, a discernible but modest positive linear relationship emerges. While not robust, this model represents a subtle enhancement over the total salary correlation. This is evidenced by the R-squared value of 0.2369, indicating a slight improvement, and visually demonstrated through a graph showcasing a marginally more concentrated distribution of data.

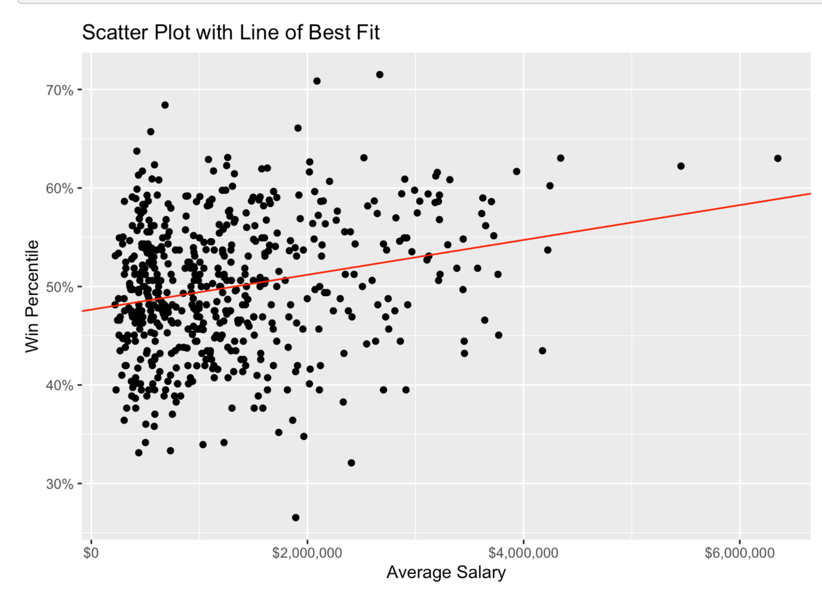
The observed pattern indicates that the majority of teams tend to adhere closely to the norm, yet there are outliers. Notably, some teams deviate from the conventional trend, and intriguingly, this departure appears to yield a higher win percentile.

It is crucial to emphasize that, despite the discerned relationship, the evidence falls short of establishing a definitive link. The limited strength of the correlation precludes any confident assertion that a higher average salary percentile unequivocally leads to a commensurately higher win percentile.

```
sum_salaries <- salary %>%
  group_by(teamID, year) %>%
  summarize(average_salary = mean(salary))

salary_percentile <- sum_salaries %>%
  group_by(year) %>%
  mutate(salary_percentile = ecdf(average_salary)(average_salary) * 100,
         joinkey = paste(teamID, year, sep = ""))

percentile_combined <- left_join(salary_percentile, mlb_summary, by = "joinkey")

final_percentile_data <- na.omit(percentile_combined)

cor(final_percentile_data$average_salary, final_percentile_data$win_perc)
```

```
## [1] 0.2369004
```

```
average_salary_model <- lm(win_perc~average_salary, data = final_percentile_data)

final_percentile_data %>%
  ggplot(aes(x = average_salary, y = win_perc)) +
  geom_point() +
  geom_abline(intercept = coef(average_salary_model)[1], slope = coef(average_salary_model)[2], color = "red") +
  ggtitle("Scatter Plot with Line of Best Fit") +
  xlab("Average Salary") +
  ylab("Win Percentile") +
  scale_y_continuous(labels = scales::percent) +
  scale_x_continuous(labels = scales::dollar_format(scale = 1))
```



Salary-Based Model Conclusion

After reviewing the percentiles of total salary and average salary in relation to win percentile, we observe a modest positive correlation in both models. Notably, the average salary percentile demonstrates a slightly stronger association with win percentile compared to the total salary percentile. This suggests that prioritizing a higher average salary across the team yields a more favorable correlation than disproportionately compensating a few players.

This analysis leads to the inference that the construct of a successful baseball team leans towards emphasizing a higher average salary rather than elevating a select few players significantly above their teammates. It underscores the collective nature of baseball as a team game, where the performance of individuals may impact the season but is not the sole determinant.

In summary, while both the average salary percentile and total salary percentile exhibit positive relationships with win percentile, neither demonstrates a robust association significant enough to assert that team salary is a definitive factor in winning more baseball games.

Event-Based Model

Now that we've established that salary plays a role, albeit not a predominant one, in baseball success, let's delve into in-game events that significantly contribute to wins. The chosen metrics for this model are offensive-based and focus on the batter's perspective. Please note that all examples provided will be based on an example in which the Detroit Tigers are batting against the Chicago White Sox pitching.

- Total Hits
- Singles
- Doubles
- Triples
- Home-Runs
- Hits with 0 Outs
- Hits with 1 Out
- Hits with 2 Outs
- First Pitch Strikes: A value in this column indicates the number of first pitch strikes that the Tigers have taken while batting
- Walks
- Strikeouts: A value in this column indicates the number of strikeouts that the Tigers have while batting
- Double-Plays: A value in this column indicates the number of double-plays that the Tigers have hit into
- Triple-Plays: A value in this column indicates the number of triple-plays that the Tigers have hit into
- Errors: A value in this column indicates the number of errors that the Tigers have occurred while they are batting
- Fly-Ball Hits
- Line-Drive Hits
- Pop-Up Hits
- Stolen Bases (1st to 2nd)
- Stolen Bases (2nd to 3rd)
- Stolen Bases (3rd to Home)
- Runs Scored

After fitting the initial logistic regression model for all the listed metrics, it became evident that some adjustments were necessary before selecting a final model. Notably, there was substantial collinearity between Total Hits, Singles, Doubles, Triples, and Home Runs. To address this, Total Hits was removed from the model. Hits with 2 Outs were also excluded due to collinearity with other hit metrics.

To refine the model, I assigned weights to the hit metrics (Singles, Doubles, Triples, and Home Runs) based on the number of bases each hit would yield (1 for Single, 2 for Double, 3 for Triple, and 4 for Home Run). This decision was driven by the understanding that, in a game, the distance a base-runner starts from the box significantly impacts a team's chances of scoring, and therefore winning.

The final model also incorporates an interaction term between First Pitch Strikes and Strikeouts, recognizing their substantial contributions and close relationship.

Upon fitting the model, it is evident that all predictors included are statistically significant and should be retained in the final model for accurate predictions.

Fitting Model

```
result_table <- all_data_long %>%
  filter(result != "game_played") %>%
  mutate(id = paste(game_id, team, sep = ""))

cor_table <- all_datajoin_base %>%
  pivot_longer(cols = c(home_team, away_team), names_to = "team_indicator", values_to = "team") %>%
  mutate(team_indicator = ifelse(team_indicator == "home_team", 1, 0)) %>%
  group_by(team, game_id, team_indicator, year) %>%
  summarise(
    total_hits = sum(hit_val > 0 & team_indicator == batting_team),
    singles = sum(hit_val == 1 & team_indicator == batting_team),
    doubles = sum(hit_val == 2 & team_indicator == batting_team),
    triples = sum(hit_val == 3 & team_indicator == batting_team),
    home_runs = sum(hit_val == 4 & team_indicator == batting_team),
    hits_0_outs = sum(hit_val > 0 & outs == 0 & team_indicator == batting_team),
    hits_1_out = sum(hit_val > 0 & outs == 1 & team_indicator == batting_team),
    hits_2_outs = sum(hit_val > 0 & outs == 2 & team_indicator == batting_team),
    first_pitch_strikes = sum(substr(pitch_seq, 1, 1) %in% c("A", "C", "K", "S") & team_indicator == batting_tea
m),
    walks = sum(grepl("W", event_scoring, ignore.case = TRUE) & !grepl("WP|DW", event_scoring, ignore.case = TRU
E) & team_indicator == batting_team),
    strikeouts = sum(event_type == "strikeout" & team_indicator == batting_team),
    double_play = sum(dp == 1 & team_indicator == batting_team),
    triple_play = sum(tp == 1 & team_indicator == batting_team),
    errors = sum(num_err > 0 & team_indicator != batting_team),
    fly_ball_hits = sum(batted_ball_type == "f" & hit_val > 0 & team_indicator == batting_team),
    line_drive_hits = sum(batted_ball_type == "l" & hit_val > 0 & team_indicator == batting_team),
    pop_up_hits = sum(batted_ball_type == "P" & hit_val > 0 & team_indicator == batting_team),
    ground_ball_hits = sum(batted_ball_type == "G" & hit_val > 0 & team_indicator == batting_team),
    stolen_base_1 = sum(sb_run_1b == 1 & team_indicator == batting_team),
    stolen_base_2 = sum(sb_run_2b == 1 & team_indicator == batting_team),
    stolen_base_3 = sum(sb_run_3b == 1 & team_indicator == batting_team),
    runs_scored = ifelse(team_indicator == 1, max(home_score), max(vis_score))
  ) %>%
  mutate(id = paste(game_id, team, sep = ""))

final_cor_table <- left_join(cor_table, result_table, by = "id") %>%
  dplyr::select(result, total_hits, singles, doubles, triples, home_runs, hits_0_outs, hits_1_out, hits_2_outs, f
irst_pitch_strikes, walks, strikeouts, double_play, triple_play, errors, fly_ball_hits, line_drive_hits, pop_up_h
its, ground_ball_hits, stolen_base_1, stolen_base_2, stolen_base_3, runs_scored) %>%
  mutate(result = ifelse(result == "win", 1, 0))

model_1 <- glm(result ~ singles + I(doubles*2) + I(triples*3) + I(home_runs*4) + hits_0_outs + hits_1_out + first
_pitch_strikes*strikeouts + walks + double_play + triple_play + errors + fly_ball_hits + line_drive_hits + ground
_ball_hits + stolen_base_1 + stolen_base_2 + stolen_base_3 + runs_scored, data = final_cor_table, family = binomi
al)

summary(model_1)
```

```
##
## Call:
## glm(formula = result ~ singles + I(doubles * 2) + I(triples *
##     3) + I(home_runs * 4) + hits_0_outs + hits_1_out + first_pitch_strikes *
##     strikeouts + walks + double_play + triple_play + errors +
##     fly_ball_hits + line_drive_hits + ground_ball_hits + stolen_base_1 +
##     stolen_base_2 + stolen_base_3 + runs_scored, family = binomial,
##     data = final_cor_table)
##
## Coefficients:
##                                Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                  -1.254e+00  2.973e-03 -421.860   <2e-16 ***
## singles                      -3.440e-02  3.837e-04  -89.668   <2e-16 ***
## I(doubles * 2)               -1.459e-02  2.923e-04  -49.933   <2e-16 ***
## I(triples * 3)                8.941e-03  4.527e-04   19.753   <2e-16 ***
## I(home_runs * 4)             -1.653e-02  2.091e-04  -79.051   <2e-16 ***
## hits_0_outs                   4.064e-03  4.235e-04    9.598   <2e-16 ***
## hits_1_out                    1.519e-02  4.192e-04   36.233   <2e-16 ***
## first_pitch_strikes          -2.812e-02  2.405e-04 -116.922   <2e-16 ***
## strikeouts                   -7.190e-02  3.480e-04 -206.600   <2e-16 ***
## walks                         4.510e-02  3.006e-04  150.008   <2e-16 ***
## double_play                  -1.182e-01  6.617e-04 -178.574   <2e-16 ***
## triple_play                  -2.419e-01  1.921e-02  -12.596   <2e-16 ***
## errors                       -5.435e-01  6.915e-04 -785.945   <2e-16 ***
## fly_ball_hits                 1.095e-02  5.374e-04   20.367   <2e-16 ***
## line_drive_hits               1.319e-02  3.019e-04   43.685   <2e-16 ***
## ground_ball_hits              3.285e-02  3.779e-04   86.919   <2e-16 ***
## stolen_base_1                 2.304e-01  7.273e-04  316.776   <2e-16 ***
## stolen_base_2                 1.322e-01  2.117e-03   62.469   <2e-16 ***
## stolen_base_3                 3.296e-01  9.711e-03   33.937   <2e-16 ***
## runs_scored                   5.268e-01  4.055e-04 1299.077   <2e-16 ***
## first_pitch_strikes:strikeouts  5.044e-04  3.006e-05   16.782   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 25796864  on 18608503  degrees of freedom
## Residual deviance: 18510743  on 18608483  degrees of freedom
## AIC: 18510785
##
## Number of Fisher Scoring iterations: 5
```

After fitting the model, I delved into assessing its accuracy in predicting baseball game outcomes, specifically focusing on whether a particular team would win or lose. The methodology involved processing each game through the model, projecting the output as a probability of a team winning. If the probability exceeded 0.5, the prediction was labeled as 'win'; otherwise, it was categorized as 'loss.' This approach was mirrored with the actual results.

Upon analyzing the data, a comparison revealed that 75.57% of the predictions aligned with the actual game outcomes across a dataset of 117,322 total games. This robust percentage underscores the model's effectiveness in predicting game results.

Testing Model Accuracy

```
predicted_value <- model_1 %>% predict(final_cor_table, type = "response")

predicted.classes <- ifelse(predicted_value > 0.5, "win", "loss")
actual.classes <- ifelse(final_cor_table$result > 0.5, "win", "loss")
mean(predicted.classes == actual.classes)
```

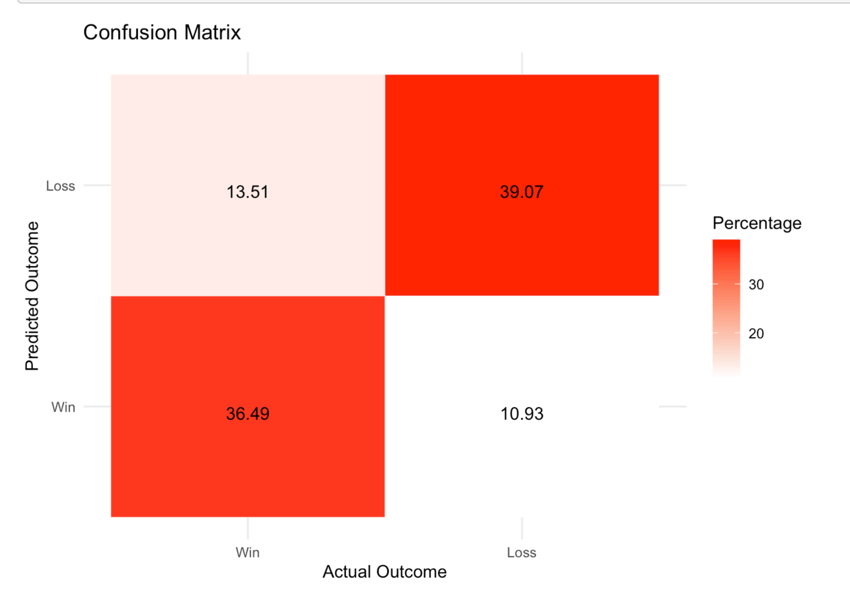
```
## [1] 0.7556637
```

Now that I've developed a model demonstrating an accuracy rate of over 75% in predicting baseball game outcomes from 1970-2022, I sought to understand the nature of the inaccuracies to enhance the model's reliability. Specifically, I examined whether the missed predictions were a result of overestimating or underestimating the data, aiming to identify potential systemic issues.

To investigate this, I constructed a confusion matrix for the values that were inaccurately predicted. The analysis revealed a relatively even distribution between overpredicted outcomes and underpredicted outcomes. There was a slight bias towards values predicted as losses that turned out to be wins. In baseball terminology, this scenario resembles a team securing a victory despite not statistically deserving it, often seen in games with narrow margins such as a 2-1 win. These instances highlight situations where exceptional defensive play and minimal offensive output still lead to a favorable outcome.

Finding Confusion Matrix for Model

```
final_cor_table$pred <- predict(model_1, type = "response")
final_cor_table$predicted_value <- ifelse(final_cor_table$pred > 0.5, "Win", "Loss")
final_cor_table2 <- final_cor_table %>%
  mutate(actual_value = ifelse(result > 0.5, "Win", "Loss"))

final_cor_table2$predicted_value <- factor(final_cor_table2$predicted_value, levels = c('Win', 'Loss'))
final_cor_table2$actual_value <- factor(final_cor_table2$actual_value, levels = c('Win', 'Loss'))

conf_matrix <- confusionMatrix(final_cor_table2$predicted_value, final_cor_table2$actual_value)

conf_matrix_table <- as.data.frame(as.table(conf_matrix))

conf_matrix_table$Percentage <- conf_matrix_table$Freq / sum(conf_matrix_table$Freq) * 100

ggplot(conf_matrix_table, aes(x = Reference, y = Prediction, fill = Percentage)) +
  geom_tile(color = "white") +
  geom_text(aes(label = round(Percentage, 2), vjust = 1)) +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Confusion Matrix",
       x = "Actual Outcome",
       y = "Predicted Outcome") +
  theme_minimal()
```



Conclusion

In conclusion, the exploration of salary-based dynamics unveiled interesting nuances in the correlation between team salaries and win percentile. Two models, one based on the total salary percentile and the other on the average salary percentile, were constructed. While both models exhibited a positive correlation with win percentile, the strength of the relationships was not robust enough to definitively assert that higher salaries lead to more wins. Interestingly, the model based on average salary percentile demonstrated a slightly stronger association, suggesting a potential emphasis on team cohesion over individual compensation.

In essence, the findings highlight the collective nature of baseball, where the team's overall dynamics, rather than individual salaries, may play a more pivotal role in achieving success.

Turning our attention to in-game events, the analysis delved into offensive metrics from the batter's perspective. A logistic regression model was constructed, incorporating various metrics such as hits, strikeouts, walks, double-plays, and more. The model demonstrated significant predictive power, achieving an impressive accuracy rate of over 75% in predicting game outcomes. This success underscores the efficacy of the chosen offensive metrics in capturing the essence of winning baseball games.

Despite the high accuracy rate, an exploration of model inaccuracies revealed a balanced distribution between overpredicted and underpredicted outcomes. Notably, there was a slight bias towards predicting losses that turned out to be wins. This phenomenon mirrors instances in baseball where exceptional defensive play and minimal offensive output can lead to unexpected victories, highlighting the intricate and unpredictable nature of the sport.

In final, this statistical journey sought to bridge the gap between the intricate world of sabermetrics and the essence of baseball. The findings underscore the importance of a holistic team approach over individual player salaries. While salary dynamics play a role, their influence is nuanced, and success is more likely when teams prioritize a balanced and collective effort.

This study contributes valuable insights to the ongoing dialogue about the factors that truly define success in baseball. As the sport continues to evolve, this exploration lays the foundation for further investigations, inviting enthusiasts and experts alike to delve deeper into the statistical nuances that shape the game we love.