Baseball Analytics: A Comprehensive Analysis of Factors that Influence At-Bat Outcomes

Andersen Pickard, Aly Khanmohamed, Jacob Andreini, Enyu Li
Group 7 (STAT 405/605)

April 28, 2025

1 Introduction

We are using play by play data from the 2018-2024 MLB regular seasons and playoffs to evaluate player performances in various scenarios, such as certain foul ball or pitch totals, ballpark factors, counts, and more. We first used various graphs and plots to both provide historical context to aid the project's direction, and select and set-up our statistics for evaluation. This culminated in our direct, head-to-head comparison of batters in our killer plot, featuring a unique visualization of advanced metrics to determine the quality of two hitters side by side.

Our end goal with this project was to determine the quality of baseball players in an unbiased fashion. We believe that aspects such as market size, team performance, and teammate star power, among others, may wrongly influence national player perception. We want to eliminate any media, group-based, and non-statistical factors for any given player to build a better quantitative evaluation process of MLB players in a single season.

2 Datasets

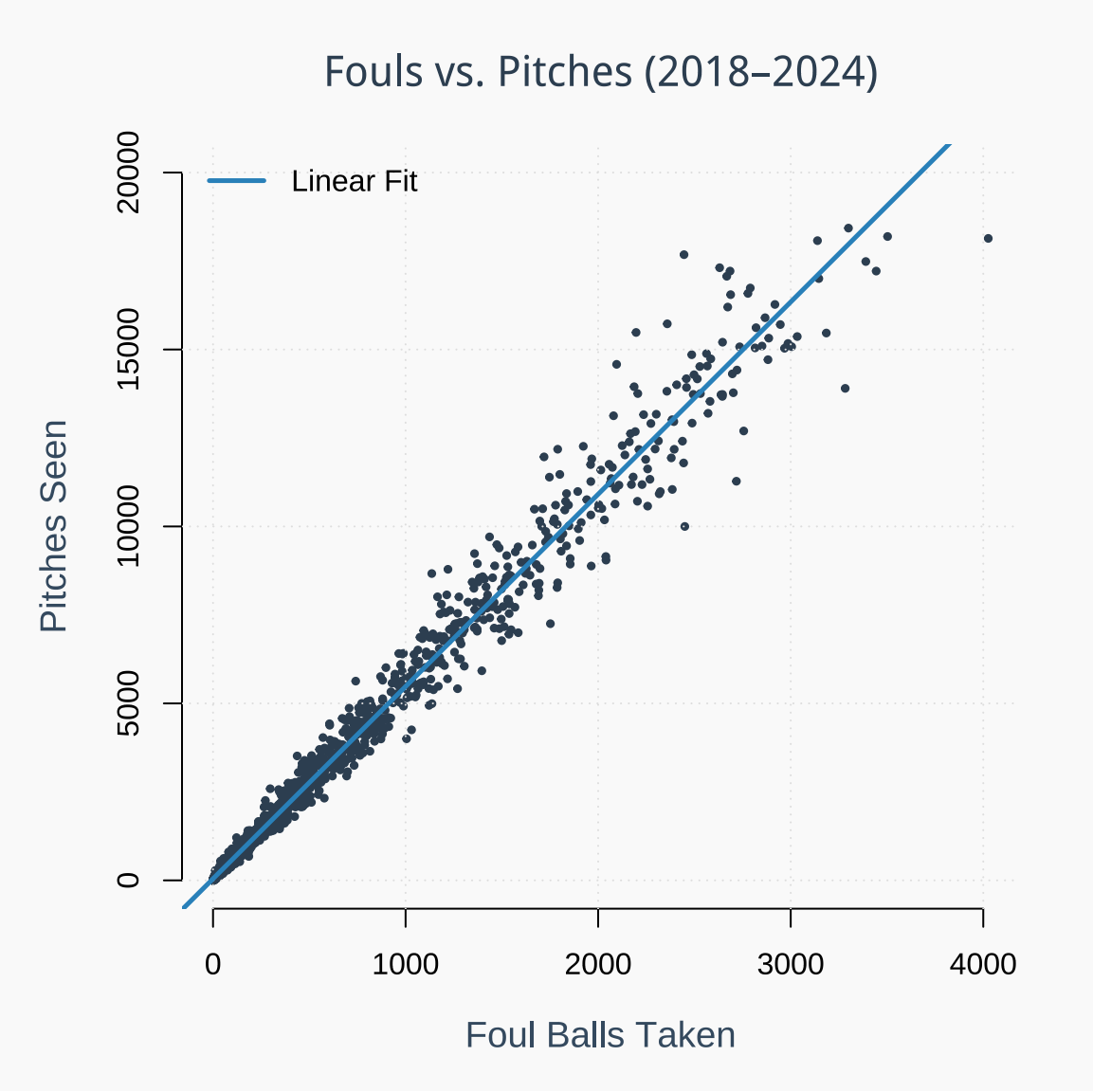
Our primary dataset contains 1,228,237 rows by 128 columns. Each row of data represents one plate appearance. Each column includes contextual, binary, and cumulative information that allows us to determine the events and overall result of the at-bat. This allows us to use situational-based data to get a better understanding of player performance, rather than just counting stats.

Our secondary dataset was predominantly used to create our killer plot. The dataset contains 617 rows by fifteen columns. It includes further specific hitting data that is not seen in our primary dataset. We were able to weight a variety of these variables to help create our hitter visualization.

3 Scatter Plot with Linear Regression

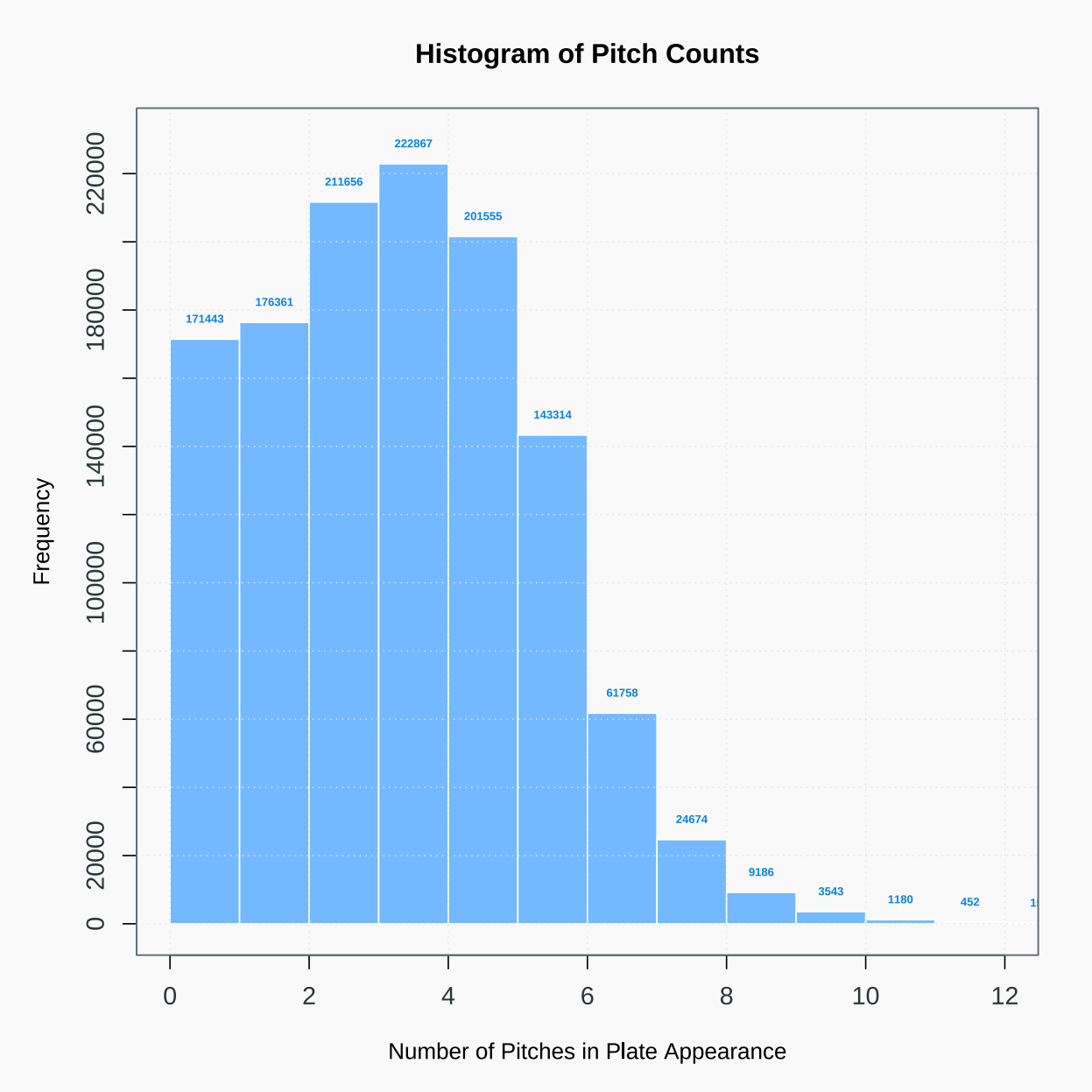
This scatter plot compares number of foul balls vs. total pitches in a single plate appearance. While the result is not necessarily surprising (it makes sense that the number of foul balls would increase as the number of pitches increases), we believe this is a very valuable result as it shows the rarity of outliers. In other words, this graph proves that there are relatively few occasions where players hit foul balls at an abnormally high or low rate.

The linear regression function shows a strong connection between the two variables, as expected. This relationship will serve as a strong foundation off which we can build our future research and analysis.



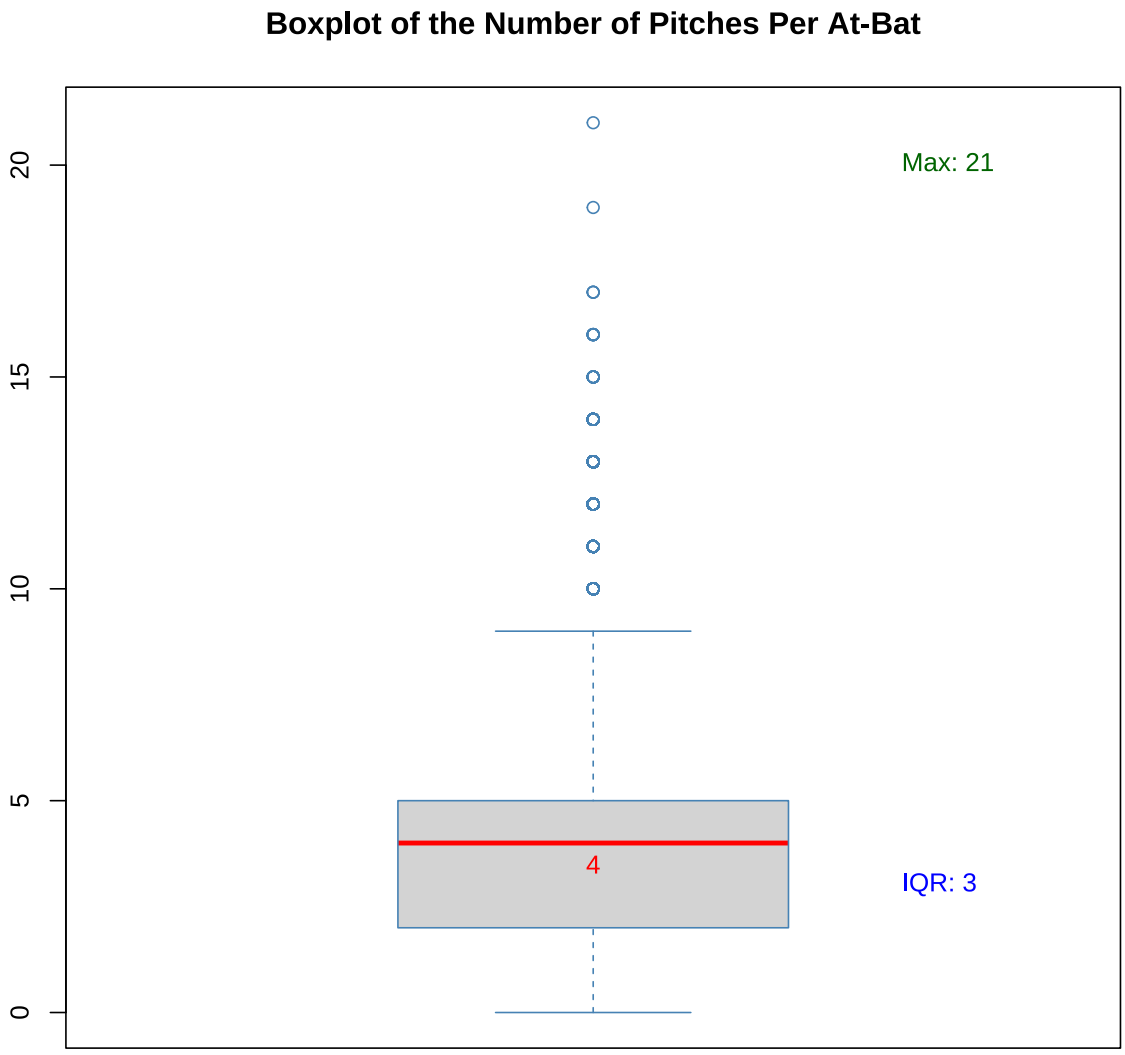
4 Histogram of Pitch Counts

This histogram shows the frequency distribution of pitches per plate appearance. As we start to look at the value of extending at bats with foul balls, we feel it is important to see the frequency of plate appearances that last a certain number of pitches. As expected, most plate appearances are five pitches or less, with four being the most common result. The data here is skewed right due to the presence of extreme cases (atypically long at-bats).



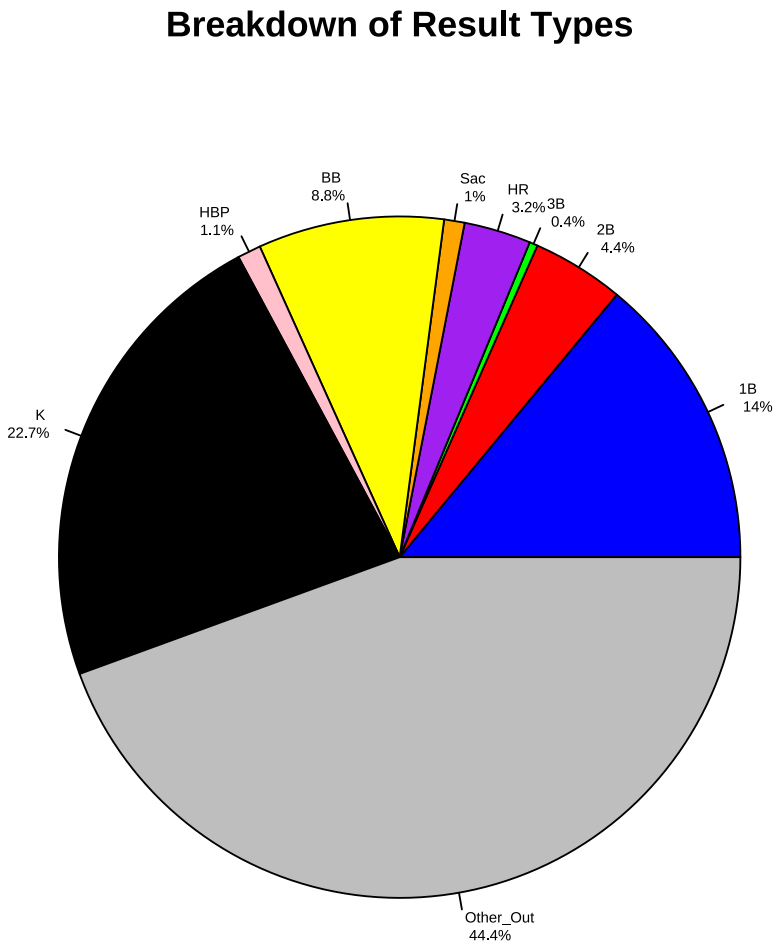
5 Boxplot of Pitches per At-Bat

This boxplot provides necessary context into our dataset and nicely complements Graph 2. The plot details the number of pitches thrown per plate appearance. We can see that any at-bat with 10+ pitches is essentially an outlier, which is interesting given that this is somewhat uncommon, but we did not feel it was necessarily rare. To expand on this plot for the future, we can divide our dataset into separate years and re-analyze the pitch counts per at-bat. That way, we can analyze for trends and differences in pitching between years.



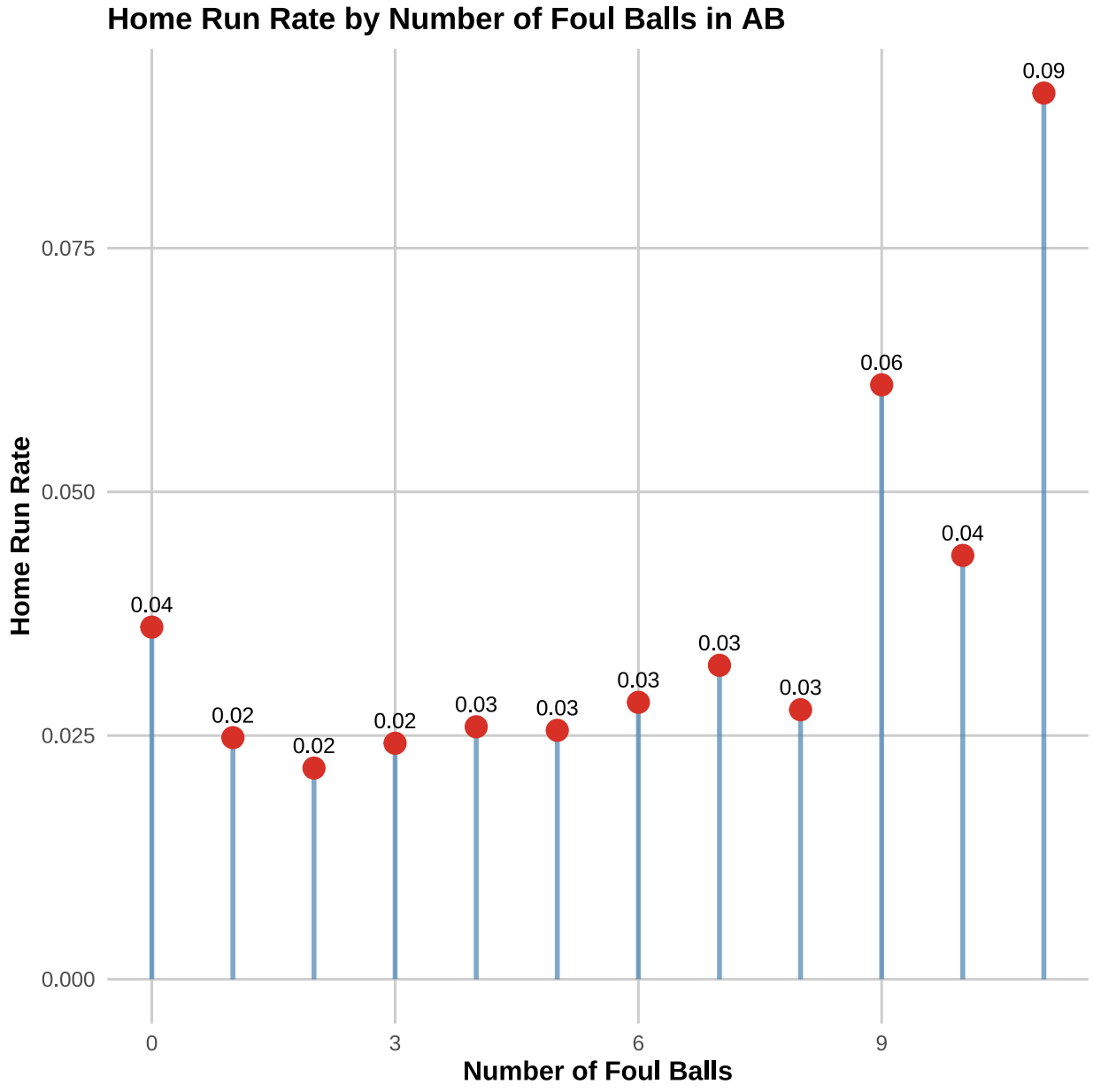
6 Pie Chart

This chart shows a breakdown of outcomes from plate appearances. These are the most common outcomes. However, it's important to note that this chart does not include every possible outcome, such as obscure results like catcher's interference. While these obscure outcomes will not be omitted from our overall research, we made the decision to omit them from this chart given their lack of relevance when comparing the more common outcomes. Rather, this chart shows the rate at which the most common outcomes occur relative to each other. This establishes an awareness the frequency of a given outcome, such as a home run, to provide additional context to future research on how foul balls impact the rate at which these outcomes occur.



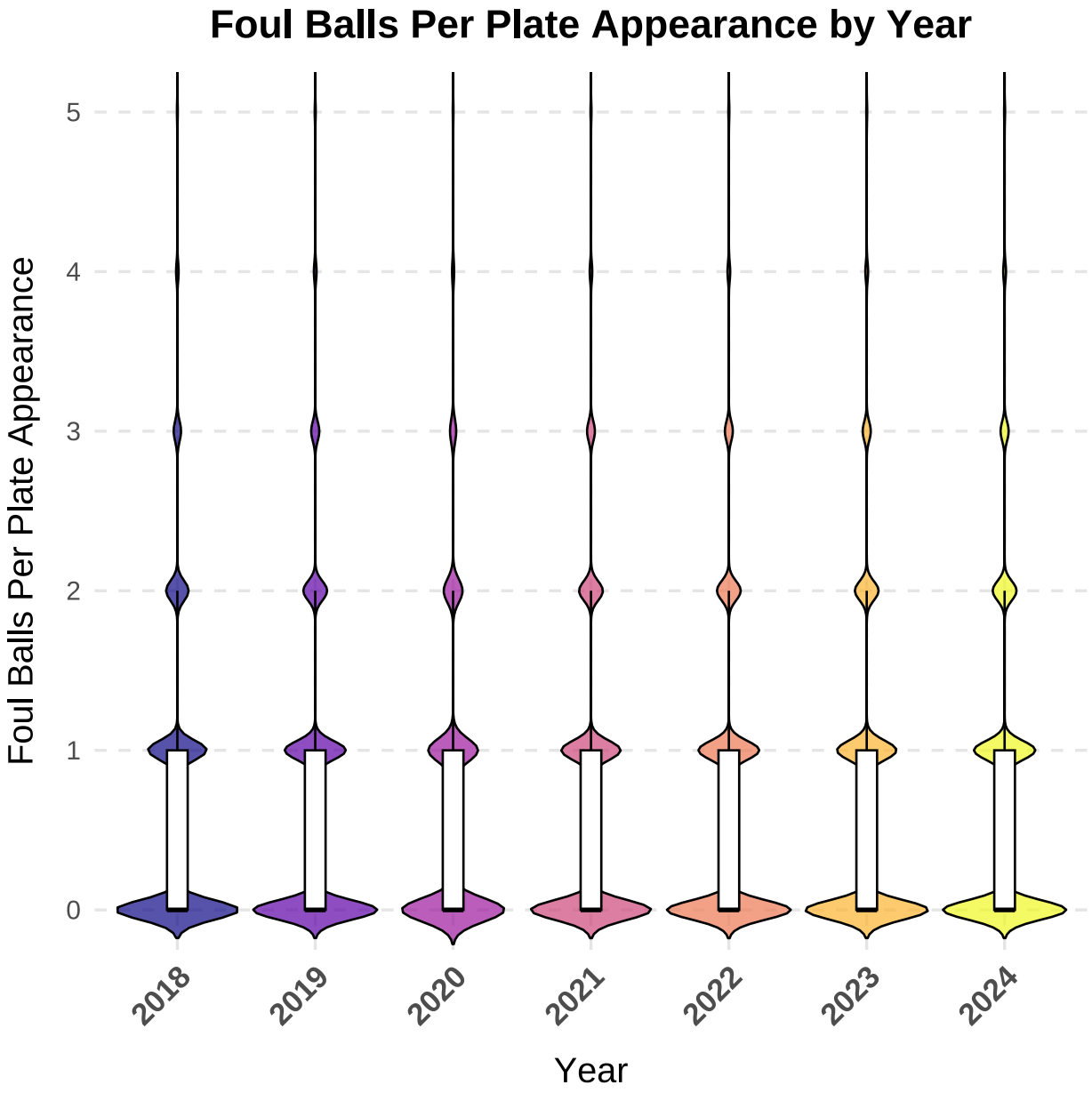
7 Lollipop Chart of HR Rate vs. Foul Count

This lollipop chart shows the rate that home runs are hit based on the length of a plate appearance. We can see that as plate appearances get longer (such as ¿8 pitches), the rate at which home runs are hit increases drastically. This is a very strong visual that provides a firm foundation for our research on the impact that a longer PA has on the likelihood of a home run. It suggests that a longer PA may benefit the batter more than the pitcher, and that a pitcher may perhaps tire faster. It is interesting to see that there is some sort of rough pattern/trend forming where number of fouls increases almost exponentially as home run rate increases.



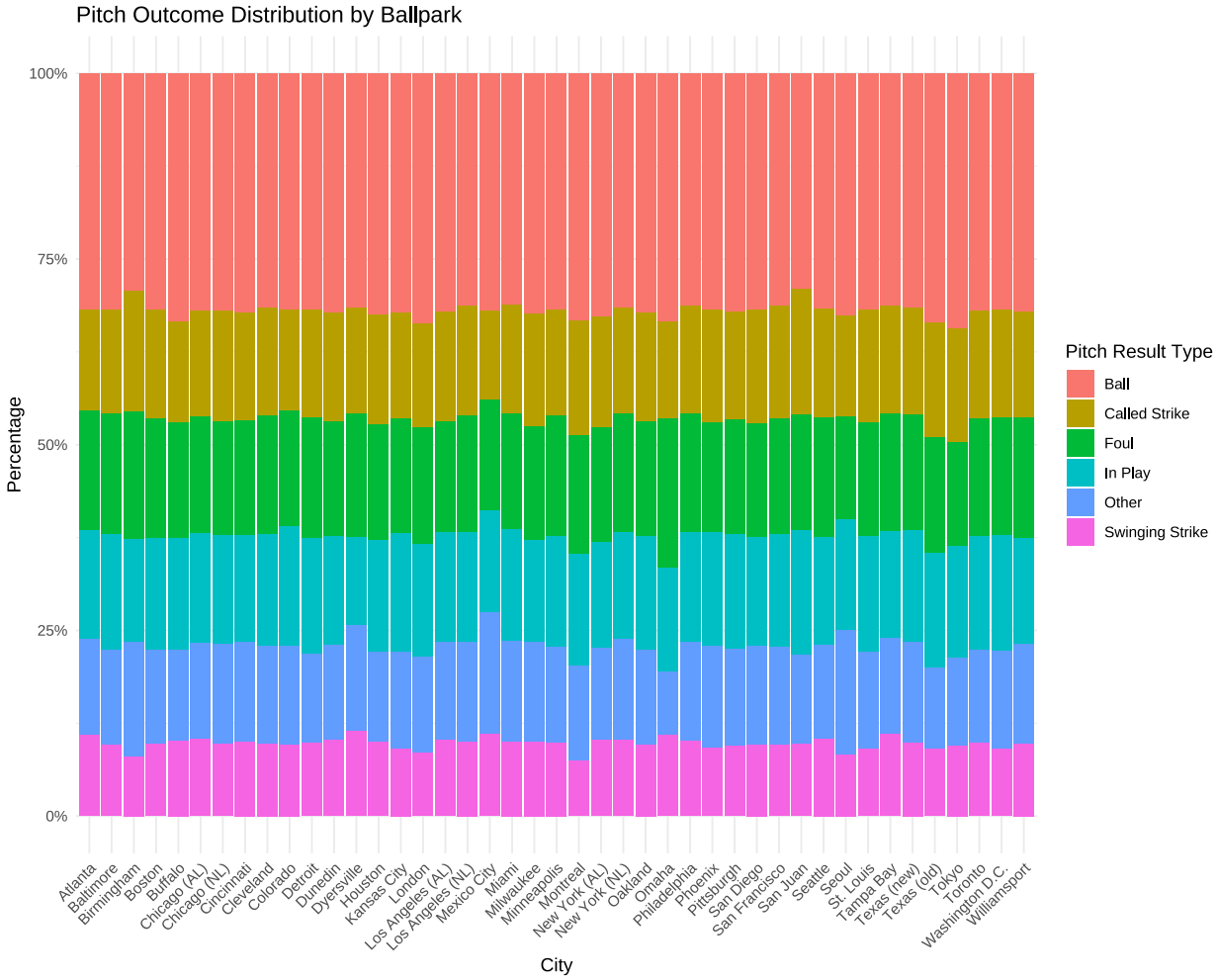
8 Violin Plot

This graph shows the distribution of foul balls per plate appearance over the seven years in our dataset. There is slight fluctuation in the densities at the individual whole numbers. 2020 naturally had fewer of each rate, for example, due to the shortened season, but for the most part, the results were consistent year over year.



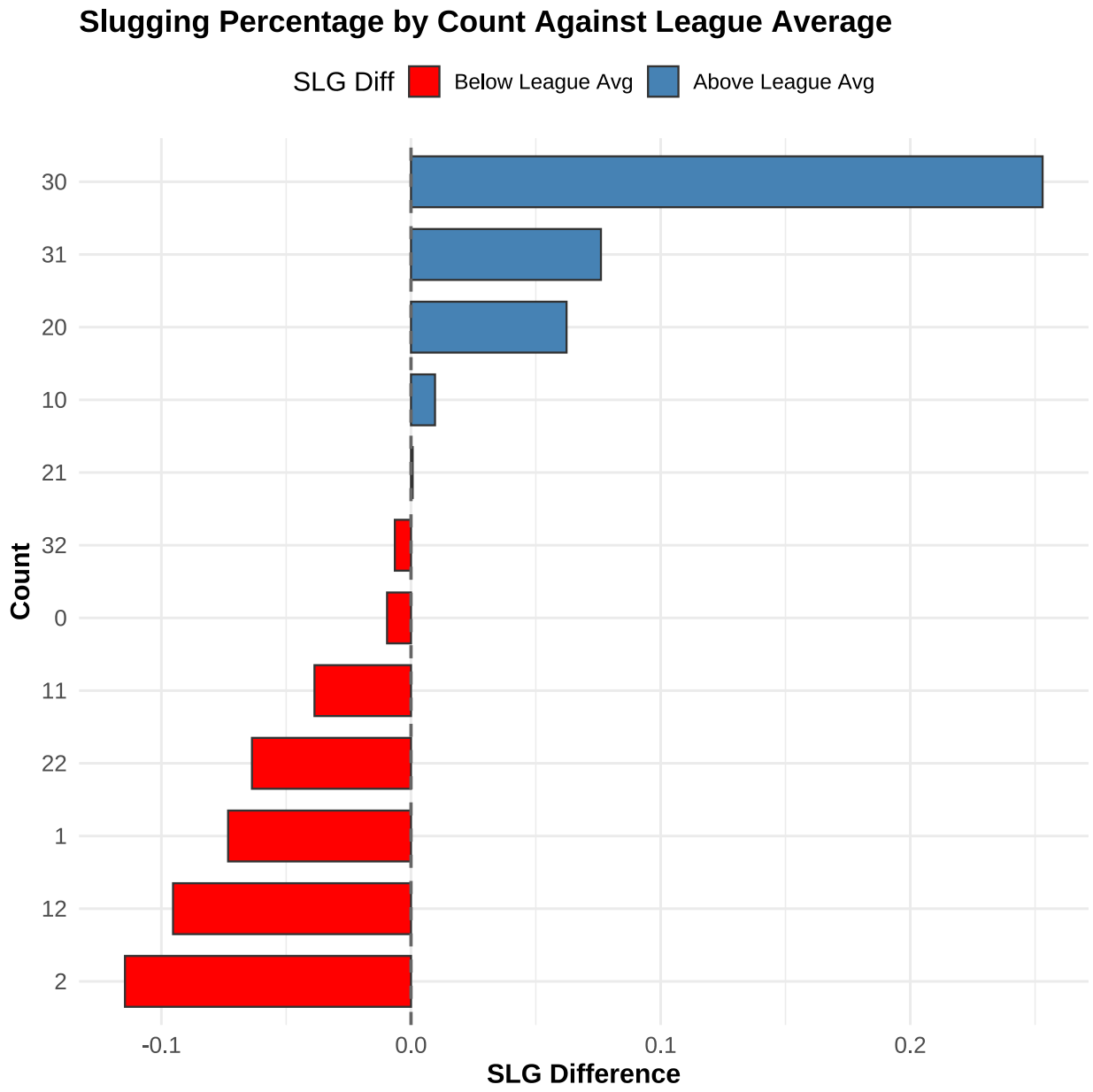
9 Categorywise Bar Chart by Ballpark

We used a categorywise bar chart to show the rate at which different outcomes occur at different ballparks. For the most part, results were consistent from park to part, but there are also some differences that we can study further. One notable difference is that the Oakland ballpark had a significantly larger ratio of foul balls. This makes sense because the Oakland ballpark has more foul territory than any of the other MLB ballparks.



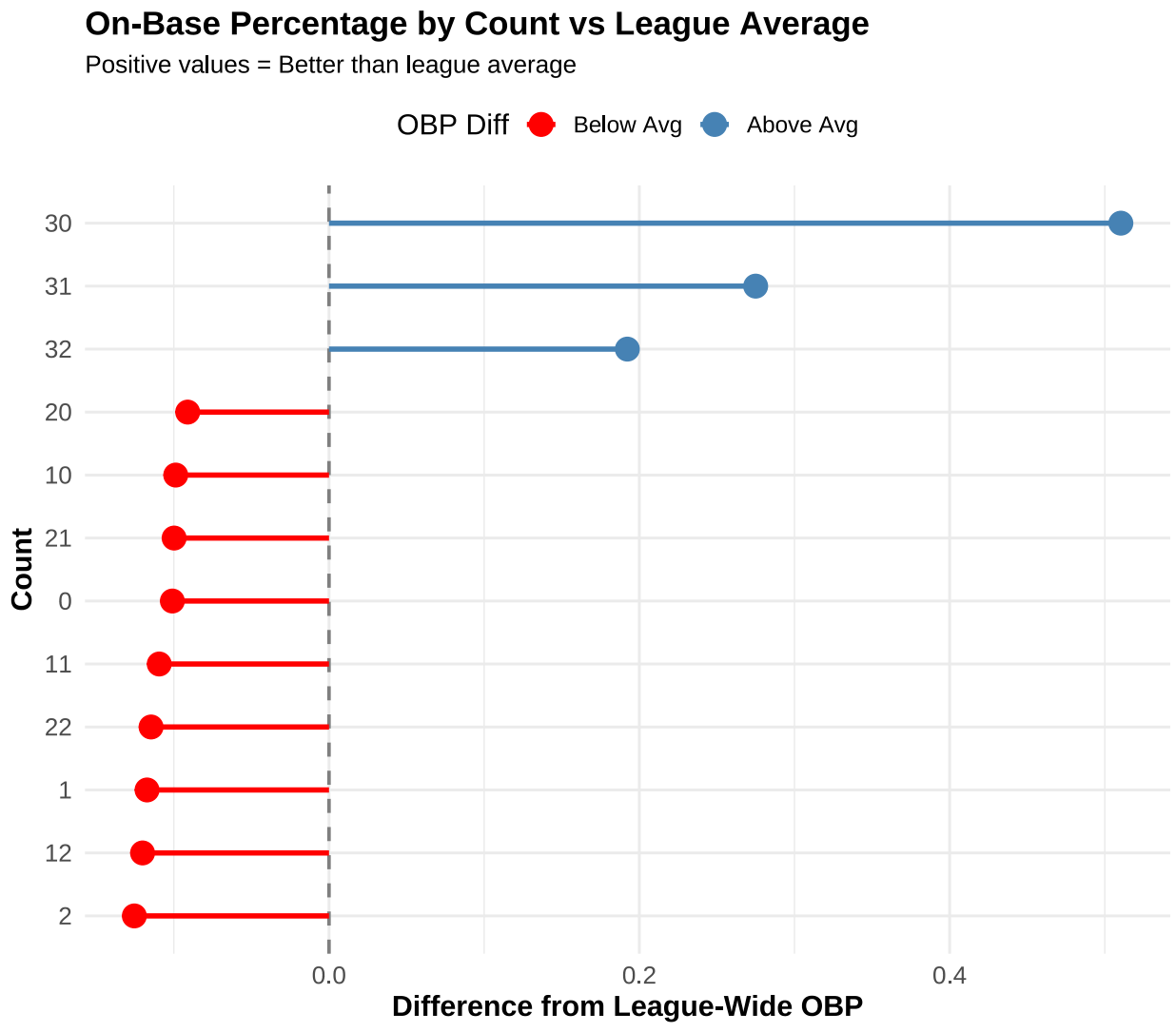
10 Diverging Bar Chart of SLG by Count

This diverging bar graph shows slugging percentage by count compared to the league average. This is a very interesting visual that allows us to gain more insight into which counts are more advantageous for pitchers or batters. It does not surprise us that counts with three balls are the best since a pitcher needs to hit the strike zone or else it's a walk, but if they hit the strike zone, the batter is more likely to put the ball in play. There is a similar pressure with a 2-0 count. Meanwhile, the lowest SLG occurs on 0-2 counts, which makes sense because the pitcher can "waste" some of their pitches and does not need to get conservative with a pitch thrown in the zone.



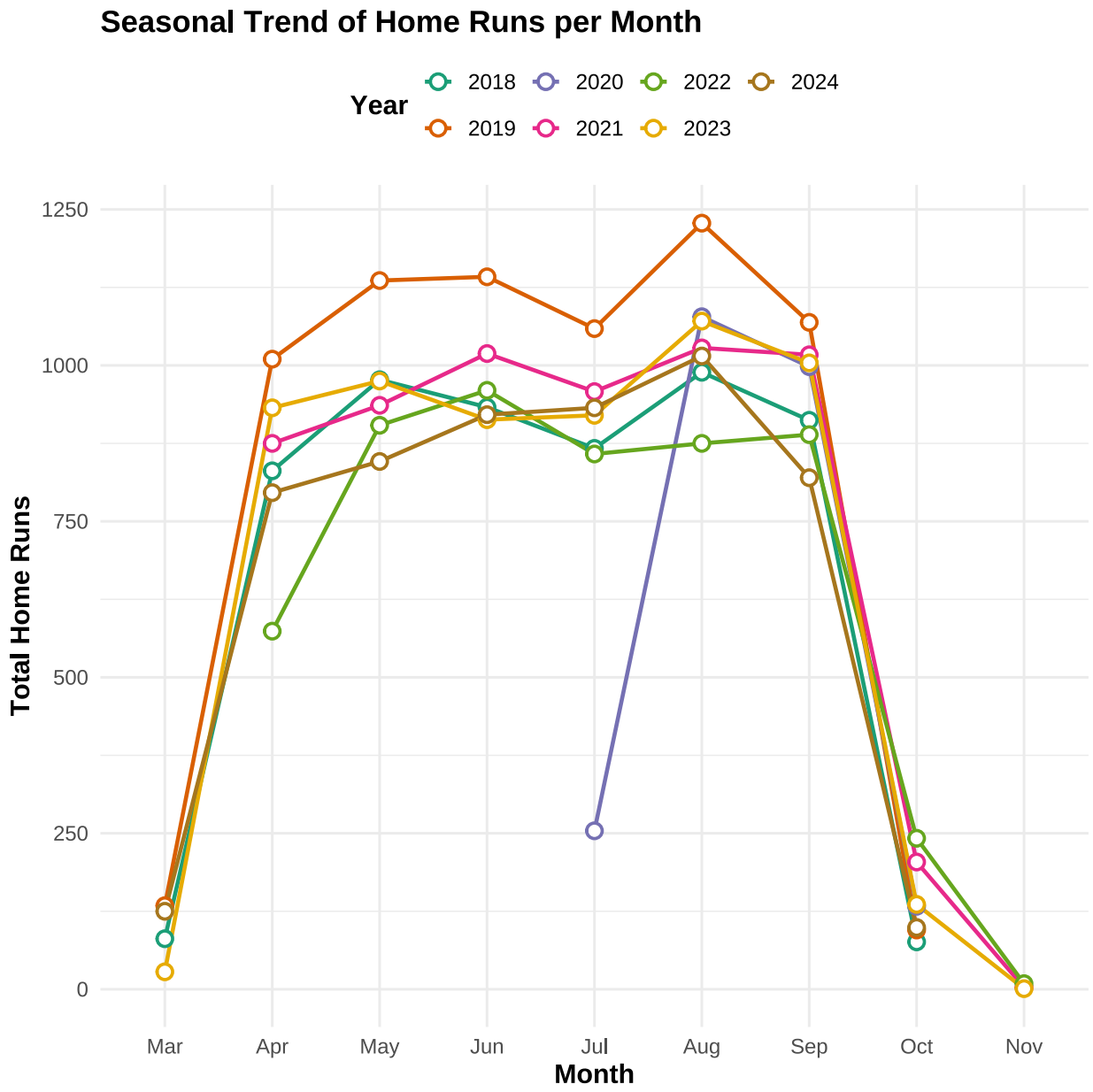
11 Diverging Lollipop Chart

This diverging lollipop chart shows on-base percentage by count compared to the league average. This is a very interesting visual that allows us to gain more insight into which counts are more advantageous for pitchers or batters. It does not surprise us that 3-ball counts are the best since a pitcher needs to hit the strike zone or else it's a walk, but if they hit the strike zone, the batter is more likely to put the ball in play. There's similar pressure with a 2-0 count. Meanwhile, the lowest OBP occurs on 0-2 counts, which makes sense because the pitcher can "waste" some of their pitches and does not need to get conservative with a pitch thrown in the zone. This chart is similar to our diverging bar graph, which focused on SLG. However, the visual looks a little different for a few reasons. Most notably, the effect of 3-ball counts is extremely drastic and skews both the data and the visual. This is because of the likelihood that a walk follows the pitch in a 3-ball count. A walk does not count toward the SLG, so it does not skew that graph, but it does count toward the OBP here.



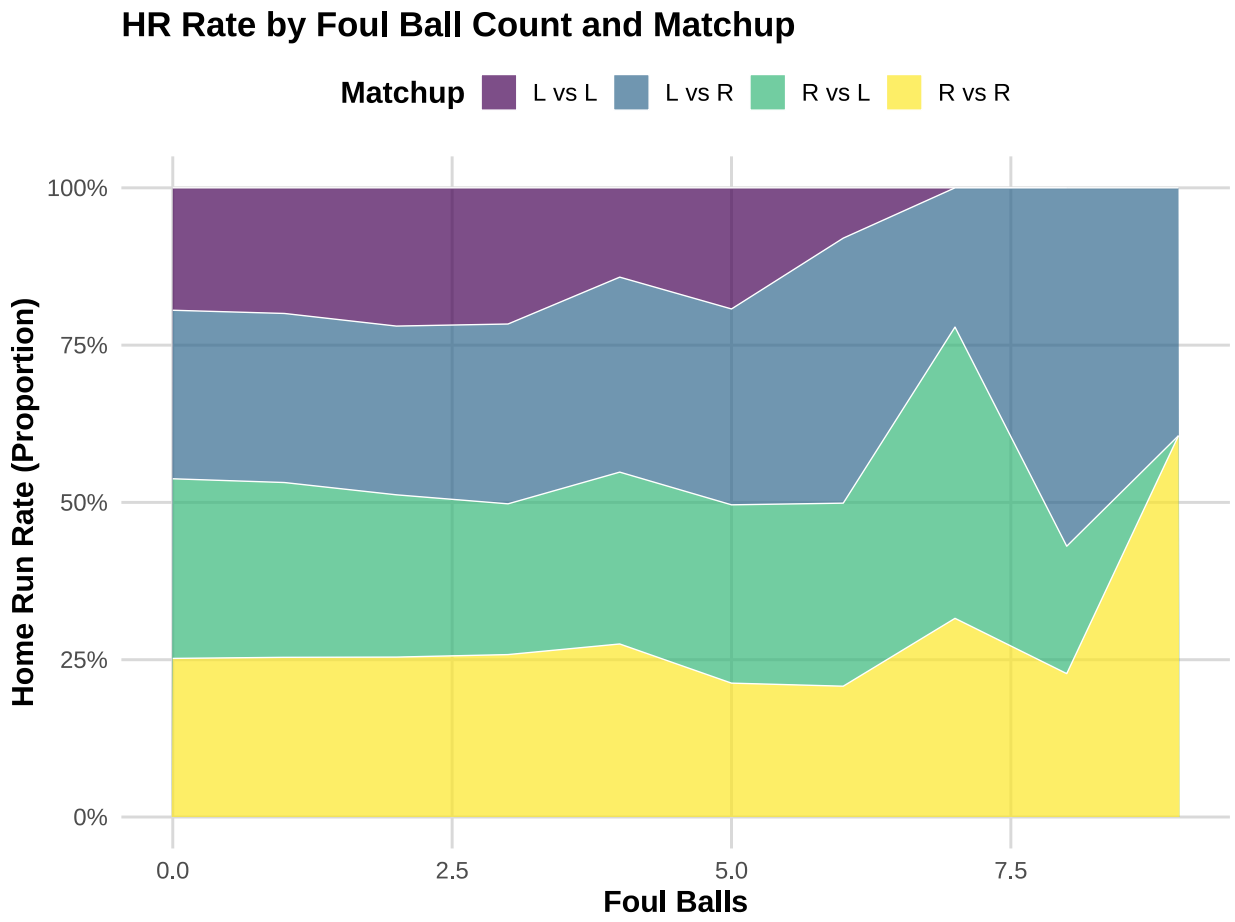
12 Seasonal Plot of Home Runs Per Month

This seasonal plot shows the trend of home runs over the months within a given season. This visual gives us insight into a few different trends, including the potential of a "juiced" ball in 2019, the effect of the hot summer months on power output, the general in-season fluctuation, and the effects of months with reduced game schedules. Ultimately, this visual allows us to begin drawing conclusions or conducting further research on certain years.



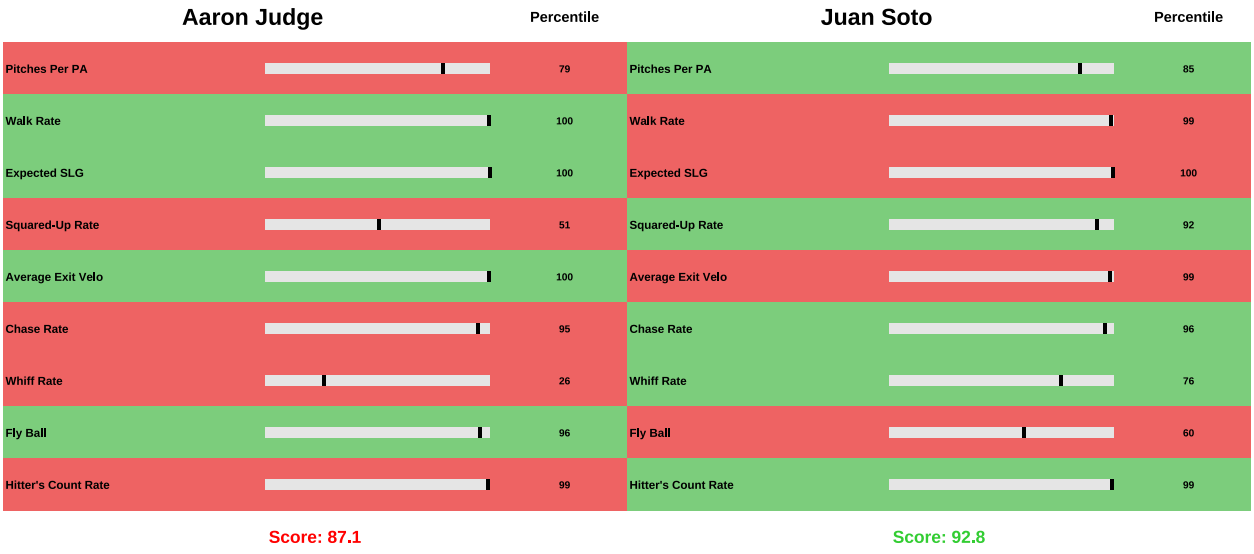
13 Area Graph of HR Rate by Matchup and Foul Count

This area graph shows the proportions of total home runs that are hit in four different matchups: LHP vs. LHH; LHP vs. RHH; RHP vs. LHH; RHP vs. LHP. This visualization provide insight into which matchups may be more advantageous for the hitter or pitcher as the at-bat either lasts longer or shorter. We can see that there is a roughly even split with a low number of foul balls, but it is almost unheard of for R vs. L or L vs. L home runs to occur once you get closer to, for example, ten fouls in an at-bat.



14 Killer Plot

Our killer plot is a visualization that allows MLB teams to compare two players side by side. While there are many different ways to compare and quantify players, we believe our visualization accurately and efficiently provides an analysis of players' abilities to work deep into counts and deliver quality contact, two components of being a successful hitter. Managers or front office executives can select two players to compare, and a weighted equation provides a composite score for each player. The player with the higher composite score is viewed as the "better" hitter. The interactive version of our Killer Plot allows you to select any two players with 10+ plate appearances in 2024. For the sake of this report, we will compare Aaron Judge and Juan Soto, who were two players with very high composite scores.



15 Conclusion

We have learned much about baseball analytics throughout our work on this project, using patterns from real season data to formulate new proofs and unlock previously undiscovered trends. We have found a few key takeaways from our work.

First, we conducted basic, preliminary background research into common baseball claims. For example, we expected that there would be a linear relationship between pitches and foul balls, but we finally proved that statistically so that we could advance with our analysis. Similarly, we developed a histogram of pitch counts showing that the vast majority of at-bats end within five pitches; a boxplot showing another distribution of pitches per at-bat; and a pie chart of at-bat outcomes that showed that batters get out nearly 70 percent of the time and that more than two-thirds of batters who reach base do so via a walk or single. This research allows us to proceed with further, much deeper analysis into topics such as at-bat length and batted ball outcomes.

Next, we found that batters perform far better as at-bats increase in length. One of the best indicators of this determination is the lollipop chart of HR rate vs. foul counts. We saw that home run rates stay consistent for the first eight pitches of an at-bat, but then the rate increases exponentially after that point. This could be due to various factors, with the most prominent reason being that the pitcher grows tired and

the batter becomes more comfortable as plate appearances last longer. Longer PAs lead to more balls going into play. This stresses the importance of hitters protecting the plate, being selective with their pitch, and continuing to foul off unwanted pitches.

We decided to take this research one step further by looking at the distribution of foul balls per plate appearance and the breakdown of ballparks by outcome. We wondered if these visualizations would show us anything different than what we already understood to be true about baseball hitting. There were no major breakthroughs, but we did develop some interesting conclusions from the categorywise outcome chart broken down by ballpark. This chart showed interesting ballpark trends, such as the fact that Colorado's park had an abnormally high in-play rate, likely due to the thinner air in Denver creating more batter-friendly conditions. In the future, we could explore the impact of Denver's thin air over longer at-bats compared to the impact of conditions at pitcher-friendly parks.

Furthermore, we examined the impact of handedness matchups over the course of a long at-bat. This was an important topic to research given that opposite handedness matchups are typically seen as a major advantage for the batter. However, we wondered if extending an at-bat beyond 5-7 pitches could compensate for (or even outweigh) the potential disadvantage created by a same-handedness matchup (i.e., LHP vs. LHH). Ultimately, we found that working into deeper at-bats does, at least partially, make up for unfavorable handedness matchups. This means that, although we still prefer a LHP vs. RHH matchup than a same-handedness matchup, there is a benefit to working longer at-bats and reaching hitter's counts so that it substantially compensates for the disadvantage of a same-handedness matchup. Therefore, we believe that managers preparing to make a substitution should not simply look at handedness splits, but also look at their ability to work quality at-bats.

Additionally, we found that a batter's stats vastly improve when facing a hitter's count (a 3-ball or 2-0 count). For example, batters have a significantly higher than league-average slugging percentage and on-base percentage in these scenarios. We believe this is because the dual threat of a walk or a hit forces pitchers to be more conservative, while allowing batters to be more aggressive. This showcases the necessity for batters to work their way into positive positioning, as the work before the hit is just as important as the final pitch. However, for pitchers counts, where there are more strikes than balls, slugging percentage is far lower, as expected. This underscores the importance of a hitter working deeper into count and getting ahead early, as well as the importance of a pitcher getting quick outcomes and not falling behind early.

Finally, our killer plot provides important support to managers and front office executives as they try to compare two players. Although our plot intentionally ignores defensive and baserunning metrics, it can be used in situations where substitution decisions must be made. For example, in the late innings of a close game with runners on base, the best choice for a pinch-hitter is someone who works deep into counts, challenges the opposing pitcher, squares up the ball well, and makes quality contact that typically results in a base hit rather than an out. The presentation of our killer plot allows for a context-based comparison to be made; we're not just suggesting a better vs. worse hitter, but we're also showing where they rank relative to the rest of the league, as well as the exact quantifiable difference between certain categories.

One potential avenue for us to explore if we wanted to take this project further in the future would be the sequence of pitch result, particularly looking at whether batters or pitchers wear down faster in plate appearances with many foul balls, and therefore giving a quantitative advantage to batters or pitchers for facing more pitches.

Additionally, we would like to improve our killer plot during future studies. Visually, we would like to add player pictures and options to sort players by their team. Furthermore, we would like to add a comparison between pitchers and batters, evaluating, showcasing, and predicting who would win an at-bat between them.

Ultimately, though, we are confident in our research, findings, and analysis. In a league where ratings are driven by swinging at the first pitch and flashy plays, we have shown that being patient and smart at the plate actually results in far more advantageous outcomes for the batter.

Appendix: Full R Code

The following is the full R code used in this report. This code is displayed for your viewing, but it will not be executed when compiling the document.

```
# Document Set-Up
library(DBI)
library(RSQLite)
library(dplyr)
library(ggplot2)
library(viridis)
library(showtext)
library(tidyr)
library(stringr)
font_add_google("Montserrat", "montserrat")
showtext_auto()
dcon <- dbConnect(RSQLite::SQLite(), "405baseball.sqlite")

# Graph 1
query1 <- "SELECT
  f.batter,
  f.total_fouls,
  f.plate_appearances AS pa_foul,
  f.total_fouls * 1.0 / f.plate_appearances AS fouls_per_pa,
  n.total_nump,
  n.plate_appearances AS pa_nump,
  n.total_nump * 1.0 / n.plate_appearances AS nump_per_pa
FROM
  (SELECT
     batter,
     SUM(LENGTH(pitches) - LENGTH(REPLACE(pitches, 'F', ''))) AS total_fouls,
     COUNT(*) AS plate_appearances
   FROM stat405baseball
   GROUP BY batter) f
LEFT JOIN
  (SELECT
     batter,
     SUM(nump) AS total_nump,
     COUNT(*) AS plate_appearances
   FROM stat405baseball
   GROUP BY batter) n
ON f.batter = n.batter"
result <- dbGetQuery(dcon, query1)
lm_model <- lm(total_nump ~ total_fouls, data = result)
par(bg = "#f9f9f9", mar = c(5, 5, 4, 2))
plot(
  result$total_fouls, result$total_nump,
  xlab = "Foul Balls Taken",
  ylab = "Pitches Seen",
  main = "Fouls vs. Pitches (2018{2024)",
  ylim = c(0, 20000),
  col = "#2c3e50",
  pch = 19,
```

```
  cex = 0.5,
  cex.lab = 1.2,
  cex.main = 1.4,
  col.lab = "#34495e",
  col.main = "#2c3e50",
  bty = "n"
)
grid(col = "#dddddd", lty = "dotted")
abline(lm_model, col = "#2980b9", lwd = 2.5)
legend("topleft", legend = "Linear Fit", col = "#2980b9", lwd = 2.5, bty = "n")



# Graph 2
baseballdata <- dbGetQuery(dcon, "SELECT * FROM stat405baseball")
par(bg = "#f9f9f9", mar = c(5, 5, 4, 2))
h <- hist(
  baseballdata$nump,
  breaks = 20,
  col = "#74b9ff",
  border = "white",
  ylim = c(0, 230000),
  xlim = c(0, 12),
  xlab = "Number of Pitches in Plate Appearance",
  ylab = "Frequency",
  main = "Histogram of Pitch Counts",
  axes = FALSE
)

axis(1, at = seq(0, 12, 2), labels = seq(0, 12, 2), col.axis = "#2d3436",
     cex.axis = 1.1)
axis(2, at = seq(0, 230000, 20000), col.axis = "#2d3436", cex.axis = 1.1)

abline(h = seq(0, 230000, 20000), col = "#dfe6e9", lty = "dotted")
abline(v = seq(0, 12, 2), col = "#dfe6e9", lty = "dotted")

text(
  x = h$mids[h$counts > 0],
  y = h$counts[h$counts > 0] + 6000,
  labels = h$counts[h$counts > 0],
  col = "#0984e3",
  font = 2,
  cex = .5
)

box(col = "#636e72", lwd = 1.2)


# Graph 3
overall_median <- median(baseballdata$nump, na.rm = TRUE)
overall_iqr <- IQR(baseballdata$nump, na.rm = TRUE)
boxplot(baseballdata$nump, main = "Boxplot of the Number of Pitches Per At-Bat",
        border = "steelblue", medcol = "red")
text(x = 1.3, y = 20, labels = "Max: 21", pos = 4, col = "darkgreen")
```

```
text(x = 1, y = overall_median, labels = round(overall_median, 2), pos = 1,
     col = "red")
text(x = 1.3, y = overall_iqr, labels = paste("IQR:", round(overall_iqr, 2)),
     pos = 4, col = "blue")



# Graph 4
library(dplyr)
result_counts <- baseballdata %>%
  summarise(
    "1B" = sum(single),
    "2B" = sum(double),
    "3B" = sum(triple),
    HR = sum(hr),
    Sac = sum(sf) + sum(sh),
    BB = sum(walk) + sum(iw),
    HBP = sum(hbp),
    K = sum(k),
    OthOut = sum(othout)

  )

result_vector <- as.numeric(result_counts)
names(result_vector) <- colnames(result_counts)

totals <- sum(result_vector)
percent_labels <- paste0(round(100 * result_vector / totals, 1), "%")

pie(result_vector,
    labels = paste(names(result_vector), "\n", percent_labels),
    col = c("blue", "red", "green", "purple", "orange", "yellow",
            "pink", "black", "gray"),
    main = "Breakdown of Result Types (Single, Double, Triple, HR)",
    cex = .5)



# Graph 5
query3 <- "WITH foul_counts AS (
  SELECT *,
         LENGTH(pitches) - LENGTH(REPLACE(pitches, 'F', '')) AS foul_ball_count
  FROM stat405baseball
),
aggregated AS (
  SELECT foul_ball_count,
         COUNT(*) AS total_at_bats,
         AVG(hr) AS hr_rate
  FROM foul_counts
  GROUP BY foul_ball_count
)
SELECT *
FROM aggregated
WHERE total_at_bats >= 10
```

```
ORDER BY foul_ball_count;"
filtered_data <- dbGetQuery(dcon, query3)

ggplot(filtered_data, aes(x = foul_ball_count, y = hr_rate)) +
  geom_segment(aes(xend = foul_ball_count, y = 0, yend = hr_rate),
               color = "#4682B4", linewidth = 1.2, alpha = 0.7) +
  geom_point(color = "#D73027", size = 5) +
  geom_text(aes(label = round(hr_rate, 2)), vjust = -1, size = 4,
            color = "black") +
  labs(
    title = "Home Run Rate by Number of Foul Balls in AB",
    x = "Number of Foul Balls",
    y = "Home Run Rate"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold"),
    panel.grid.major = element_line(color = "grey80"),
    panel.grid.minor = element_blank()
  )



# Graph 6
library(ggplot2)
library(dplyr)
library(stringr)
library(viridis)

baseballdata <- baseballdata %>%
  mutate(year = as.factor(substr(gid, 4, 7)),
         foul_count = str_count(pitches, "F"))

ggplot(data = baseballdata, aes(x = year, y = foul_count, fill = year)) +
  geom_violin(alpha = 0.7, color = "black", trim = FALSE) +
  geom_boxplot(width = 0.15, fill = "white", color = "black", outlier.shape = NA) +
  scale_fill_viridis_d(option = "C") +
  coord_cartesian(ylim = c(0, 5)) +
  labs(title = "Foul Balls Per Plate Appearance by Year",
       x = "Year",
       y = "Foul Balls Per Plate Appearance") +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 18,
    margin = margin(b = 12)),  # Centered title
    axis.title.x = element_text(size = 16, margin = margin(t = 12)),
    axis.title.y = element_text(size = 16, margin = margin(r = 12)),
    axis.text.x = element_text(size = 14, face = "bold", angle = 45, hjust = 1),
    axis.text.y = element_text(size = 12),
    legend.position = "none",
    panel.grid.major = element_line(color = "gray90", linetype = "dashed"),
    panel.grid.minor = element_blank()
```

```
  )



# Graph 7
query5 <- "
WITH RECURSIVE split_pitches(gid, ballpark, pitch, rest, n) AS (
  SELECT gid, ballpark, SUBSTR(pitches, 1, 1), SUBSTR(pitches, 2), 1
  FROM stat405baseball
  WHERE pitches IS NOT NULL AND LENGTH(pitches) > 0
  UNION ALL
  SELECT gid, ballpark, SUBSTR(rest, 1, 1), SUBSTR(rest, 2), n + 1
  FROM split_pitches
  WHERE LENGTH(rest) > 0
),
filtered AS (
  SELECT ballpark,
         CASE pitch
           WHEN 'F' THEN 'Foul'
           WHEN 'B' THEN 'Ball'
           WHEN 'S' THEN 'Swinging Strike'
           WHEN 'C' THEN 'Called Strike'
           WHEN 'X' THEN 'In Play'
           ELSE 'Other'
         END AS pitch_label
  FROM split_pitches
),
counts AS (
  SELECT ballpark, pitch_label, COUNT(*) AS count
  FROM filtered
  GROUP BY ballpark, pitch_label
),
percentages AS (
  SELECT ballpark, pitch_label, count,
         ROUND(100.0 * count / SUM(count) OVER (PARTITION BY ballpark), 2)
         AS percentage
  FROM counts
)
SELECT * FROM percentages
ORDER BY ballpark, pitch_label;
"

df_pitches <- dbGetQuery(dcon, query5)

ballpark_city <- c(
  "ANA01" = "Los Angeles (AL)",
  "ARL02" = "Texas (old)",
  "ARL03" = "Texas (new)",
  "ATL03" = "Atlanta",
  "BAL12" = "Baltimore",
  "BIR01" = "Birmingham",
  "BOS07" = "Boston",
  "BUF05" = "Buffalo",
  "CHI11" = "Chicago (NL)",
```

```
  "CHI12" = "Chicago (AL)",
  "CIN09" = "Cincinnati",
  "CLE08" = "Cleveland",
  "DEN02" = "Colorado",
  "DET05" = "Detroit",
  "DUN01" = "Dunedin",
  "DYE01" = "Dyersville",
  "HOU03" = "Houston",
  "KAN06" = "Kansas City",
  "LON01" = "London",
  "LOS03" = "Los Angeles (NL)",
  "MEX02" = "Mexico City",
  "MIA02" = "Miami",
  "MIL06" = "Milwaukee",
  "MIN04" = "Minneapolis",
  "MNT01" = "Montreal",
  "NYC20" = "New York (NL)",
  "NYC21" = "New York (AL)",
  "OAK01" = "Oakland",
  "OMA01" = "Omaha",
  "PHI13" = "Philadelphia",
  "PHO01" = "Phoenix",
  "PIT08" = "Pittsburgh",
  "SAN02" = "San Diego",
  "SEA03" = "Seattle",
  "SEO01" = "Seoul",
  "SFO03" = "San Francisco",
  "SJU01" = "San Juan",
  "STL10" = "St. Louis",
  "STP01" = "Tampa Bay",
  "TOK01" = "Tokyo",
  "TOR02" = "Toronto",
  "WAS11" = "Washington D.C.",
  "WIL02" = "Williamsport"
)

df_pitches <- df_pitches %>%
  mutate(city = ballpark_city[ballpark])

ggplot(df_pitches, aes(x = city, y = percentage, fill = pitch_label)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(fill = "Pitch Result Type",
       title = "Pitch Outcome Distribution by Ballpark",
       x = "City", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


# Graph 8

base_values <- c(single = 1, double = 2, triple = 3, hr = 4, walk = 0, iw = 0, xi = 0)

slugging_data <- baseballdata %>%
```

```
  filter(!(count %in% c(40, 41, 42, 43, 3, 13, 23, 33))) %>%
  mutate(
    total_bases = single * base_values["single"] +
      double * base_values["double"] +
      triple * base_values["triple"] +
      hr * base_values["hr"],
    total_at_bats = (single + double + triple + hr + othout)
  ) %>%
  group_by(count) %>%
  summarise(
    total_bases = sum(total_bases),
    total_at_bats = sum(total_at_bats),
    slugging_pct = total_bases / total_at_bats
  ) %>%
  mutate(slugging_diff = slugging_pct - mean(slugging_pct))

ggplot(slugging_data, aes(x = reorder(count, slugging_diff), y = slugging_diff,
                          fill = slugging_diff > 0)) +
  geom_bar(stat = "identity", width = 0.7, color = "gray20") +
  coord_flip() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray40",
             linewidth = 0.8) +
  scale_fill_manual(
    values = c("red", "steelblue"),
    labels = c("Below League Avg", "Above League Avg"),
    name = "SLG Diff"
  ) +
  labs(
    title = "Slugging Percentage by Count Against League Average",
    x = "Count",
    y = "SLG Difference"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 12),
    legend.position = "top"
  )

# Graph 9
library(dplyr)
library(ggplot2)
library(DBI)
library(RSQLite)
query7 <- "WITH filtered AS (
  SELECT *
  FROM stat405baseball
  WHERE count NOT IN (40, 41, 42, 43, 3, 13, 23, 33)
),
with_obp AS (
  SELECT
    count,
```

```
    (single + double + triple + hr + walk + iw + xi) AS reached_base,
    (single + double + triple + hr + walk + iw + xi + othout) AS total_pa
  FROM filtered
),
grouped AS (
  SELECT
    count,
    SUM(reached_base) AS total_reached_base,
    SUM(total_pa) AS total_pa,
    SUM(reached_base) * 1.0 / SUM(total_pa) AS obp
  FROM with_obp
  GROUP BY count
),
with_league_avg AS (
  SELECT
    *,
    AVG(obp) OVER () AS league_obp
  FROM grouped
)
SELECT
  count,
  total_reached_base,
  total_pa,
  obp,
  obp - league_obp AS obp_diff
FROM with_league_avg
ORDER BY obp_diff DESC;
"

library(ggplot2)

obp_data <- dbGetQuery(dcon, query7)
obp_data$count <- factor(obp_data$count)

ggplot(obp_data, aes(x = reorder(count, obp_diff), y = obp_diff,
                     color = obp_diff > 0)) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray50",
             linewidth = 0.8) +
  geom_segment(aes(xend = count, y = 0, yend = obp_diff), linewidth = 1.2) +
  geom_point(size = 5) +
  scale_color_manual(
    values = c("red", "steelblue"),
    labels = c("Below Avg", "Above Avg"),
    name = "OBP Diff"
  ) +
  labs(
    title = "On-Base Percentage by Count vs League Average",
    subtitle = "Positive values = Better than league average",
    x = "Count",
    y = "Difference from League-Wide OBP"
  ) +
  coord_flip() +
  theme_minimal(base_size = 14) +
  theme(
```

```
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 12),
    legend.position = "top"
  )



# Graph 10
library(dplyr)
library(ggplot2)
library(DBI)
library(RSQLite)

query8 <- "SELECT
  SUBSTR(gid, 4, 4) AS year,
  SUBSTR(gid, 8, 2) AS month,
  SUM(hr) AS total_home_runs
FROM stat405baseball
GROUP BY year, month
ORDER BY year, month;
"

library(ggplot2)
library(RColorBrewer)

home_runs_per_month <- dbGetQuery(dcon, query8)

home_runs_per_month$month <- factor(
  home_runs_per_month$month,
  levels = sprintf("%02d", 1:12),
  labels = month.abb
)

ggplot(home_runs_per_month, aes(x = month, y = total_home_runs, group = year,
                                color = factor(year))) +
  geom_line(linewidth = 1) +
  geom_point(size = 3, shape = 21, fill = "white", stroke = 1.2) +
  scale_color_brewer(palette = "Dark2", name = "Year") +
  labs(
    title = "Seasonal Trend of Home Runs per Month",
    x = "Month",
    y = "Total Home Runs"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 11),
    legend.title = element_text(face = "bold"),
    legend.position = "top"
  )
```

```
# Graph 11
query9 <- "WITH foul_balls AS (
  SELECT *,
         LENGTH(pitches) - LENGTH(REPLACE(pitches, 'F', '')) AS foul_ball_count,
         CASE
           WHEN bathand = 'R' AND pithand = 'R' THEN 'R vs R'
           WHEN bathand = 'L' AND pithand = 'L' THEN 'L vs L'
           WHEN bathand = 'R' AND pithand = 'L' THEN 'R vs L'
           WHEN bathand = 'L' AND pithand = 'R' THEN 'L vs R'
           ELSE NULL
         END AS handedness_matchup
  FROM stat405baseball
),
filtered AS (
  SELECT * FROM foul_balls WHERE handedness_matchup IS NOT NULL
),
grouped AS (
  SELECT foul_ball_count, handedness_matchup, COUNT(*) AS total_at_bats,
         AVG(hr * 1.0) AS hr_rate
  FROM filtered
  GROUP BY foul_ball_count, handedness_matchup
),
filtered_min10 AS (
  SELECT * FROM grouped WHERE total_at_bats >= 10
),
sum_hr_rate AS (
  SELECT foul_ball_count, SUM(hr_rate) AS total_hr_rate
  FROM filtered_min10
  GROUP BY foul_ball_count
)
SELECT f.foul_ball_count, f.handedness_matchup, f.total_at_bats, f.hr_rate,
       f.hr_rate / s.total_hr_rate AS prop_hr_rate
FROM filtered_min10 f
JOIN sum_hr_rate s ON f.foul_ball_count = s.foul_ball_count
ORDER BY f.foul_ball_count, f.handedness_matchup;"
library(ggplot2)
library(scales)
library(viridis)

hrmatchup <- dbGetQuery(dcon, query9)

ggplot(hrmatchup, aes(x = foul_ball_count, y = prop_hr_rate,
                      fill = handedness_matchup)) +
  geom_area(alpha = 0.7, color = "white", linewidth = 0.3) +
  scale_fill_viridis_d(name = "Matchup", option = "D") +
  scale_y_continuous(labels = percent_format(accuracy = 1)) +
  labs(
    title = "HR Rate by Foul Ball Count and Matchup",
    x = "Foul Balls",
    y = "Home Run Rate (Proportion)"
  ) +
```

```
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(size = 12),
    axis.title = element_text(face = "bold"),
    axis.text = element_text(size = 11),
    panel.grid.major = element_line(color = "grey85"),
    panel.grid.minor = element_blank(),
    legend.position = "top",
    legend.title = element_text(face = "bold")
  )

#Killer Plot
library(knitr)
library(dplyr)
library(grid)
library(gridExtra)
library(tidyr)

killerdata <- read.csv("~/Downloads/killer_data.csv")

stat_labels <- c(
  ppa = "Pitches Per PA",
  bb_percent = "Walk Rate",
  xslg = "Expected SLG",
  squared_up_swing = "Squared-Up Rate",
  exit_velocity_avg = "Average Exit Velo",
  oz_swing_percent = "Chase Rate",
  whiff_percent = "Whiff Rate",
  flyballs_percent = "Fly Ball",
  hitcountrate = "Hitter's Count Rate"
)

batter_score_weights <- c(
  xslg = 0.20,
  bb_percent = 0.15,
  whiff_percent = 0.10,
  oz_swing_percent = 0.10,
  ppa = 0.10,
  exit_velocity_avg = 0.15,
  flyballs_percent = 0.05,
  squared_up_swing = 0.05,
  hitcountrate = 0.10
)

normalize_stat <- function(x) {
  ecdf(x)(x) * 100
}

draw_profile_card <- function(player_row, player_percentiles,
    opponent_percentiles, score_color) {
  stats <- names(player_percentiles)
  stat_labels_clean <- stat_labels[stats]
  player_score <- sum(player_percentiles[stats] * batter_score_weights[stats],
```

```
    na.rm = TRUE)

  pushViewport(viewport(layout = grid.layout(length(stats) + 3, 3,
                                             widths = unit(c(0.4, 0.4, 0.2),
                                               "npc"))))
  grid.text(player_row$last_first,
            vp = viewport(layout.pos.row = 1, layout.pos.col = 1:2),
            gp = gpar(fontsize = 16, fontface = "bold"))
  grid.text("Percentile",
            vp = viewport(layout.pos.row = 1, layout.pos.col = 3),
            gp = gpar(fontsize = 10, fontface = "bold"))
  for (i in seq_along(stats)) {
    stat <- stats[i]
    p_val <- player_percentiles[stat]
    o_val <- opponent_percentiles[stat]
    p_val_clean <- ifelse(is.na(p_val), 0, p_val)
    o_val_clean <- ifelse(is.na(o_val), 0, o_val)
    p_val_clean <- max(min(p_val_clean, 100), 0)
    o_val_clean <- max(min(o_val_clean, 100), 0)
    pct_to_pos <- function(pct) 0.05 + 0.9 * (pct / 100)
    bg_color <- if (p_val_clean > o_val_clean) "palegreen3"
      else if (p_val_clean < o_val_clean) "indianred2" else "gray80"
    row_index <- i + 1
    for (col in 1:3) {
      grid.rect(gp = gpar(col = NA, fill = bg_color),
                vp = viewport(layout.pos.row = row_index, layout.pos.col = col))
    }
    grid.text(stat_labels_clean[i],
              just = "left", x = unit(0.01, "npc"),
              gp = gpar(fontsize = 8, fontface = "bold"),
              vp = viewport(layout.pos.row = row_index, layout.pos.col = 1))
    grid.rect(x = unit(0.5, "npc"), width = unit(0.9, "npc"), height =
                unit(0.5, "lines"),
              gp = gpar(col = NA, fill = "gray90"),
              vp = viewport(layout.pos.row = row_index, layout.pos.col = 2))
    grid.rect(x = unit(pct_to_pos(p_val_clean), "npc"),
              width = unit(0.015, "npc"), height = unit(0.5, "lines"),
              gp = gpar(col = NA, fill = "black"),
              vp = viewport(layout.pos.row = row_index, layout.pos.col = 2))
    grid.text(round(p_val_clean),
              x = unit(0.5, "npc"), y = unit(0.5, "npc"),
              gp = gpar(fontsize = 7, fontface = "bold"),
              vp = viewport(layout.pos.row = row_index, layout.pos.col = 3))
  }
  grid.text(paste("Score:", round(player_score, 1)),
            gp = gpar(fontsize = 12, fontface = "bold", col = score_color),
            vp = viewport(layout.pos.row = length(stats) + 2, layout.pos.col = 1:3))

  upViewport()
}
stats <- names(stat_labels)
percentiles_df <- killerdata
for (stat in stats) {
  percentiles_df[[paste0(stat, "_pct")]] <- normalize_stat(percentiles_df[[stat]])
```

```
}

batter1 <- "Aaron Judge"
batter2 <- "Juan Soto"

p1 <- killerdata %>% filter(last_first == batter1)
p2 <- killerdata %>% filter(last_first == batter2)

p1_percentiles <- percentiles_df %>%
  filter(last_first == batter1) %>%
  select(ends_with("_pct")) %>%
  unlist(use.names = FALSE)
names(p1_percentiles) <- stats

p2_percentiles <- percentiles_df %>%
  filter(last_first == batter2) %>%
  select(ends_with("_pct")) %>%
  unlist(use.names = FALSE)
names(p2_percentiles) <- stats

score1 <- sum(p1_percentiles * batter_score_weights[stats], na.rm = TRUE)
score2 <- sum(p2_percentiles * batter_score_weights[stats], na.rm = TRUE)

color1 <- if (score1 > score2) "limegreen" else "red"
color2 <- if (score2 > score1) "limegreen" else "red"

grid.newpage()
pushViewport(viewport(layout = grid.layout(1, 2)))

pushViewport(viewport(layout.pos.row = 1, layout.pos.col = 1))
draw_profile_card(p1, p1_percentiles, p2_percentiles, color1)
upViewport()

pushViewport(viewport(layout.pos.row = 1, layout.pos.col = 2))
draw_profile_card(p2, p2_percentiles, p1_percentiles, color2)
upViewport(2)
```