Comprehensive Analysis of Jacob deGrom's Pitching Performance: A Six-Year Statcast Review (2016-2021)

By: Aryan Punjani, Bradley Parmer - Lohan, Irene Zhao, and Shrikha Dasari
STA 160
Professor Hsieh
June 11, 2024

I. Introduction

The core objective of this project is to meticulously analyze the performance evolution of Major League Baseball (MLB) pitchers over a five-year span, focusing on renowned pitcher Jacob deGrom. Utilizing extensive Statcast data accessible from "https://www.mlb.com/statcast", we aim to dissect and understand the intricate dynamics of pitching performance, which include metrics such as pitch velocity, spin rates, and the outcomes of various pitch types. Given the competitive nature of MLB and the meticulous scrutiny of player performance, understanding these metrics can provide deep insights into the changing strategies and physical conditions of pitchers across multiple seasons. The analysis will employ Statcast data, a revolutionary tracking technology that captures high-speed, detailed data on player movements and game dynamics, offering an unprecedented depth of insight into the game of baseball. This data includes detailed measures such as pitch type, velocity, spin, and resulting play outcomes, enabling a comprehensive analysis of pitcher performance. Our methodology will begin with data extraction, focusing on acquiring a clean, well-structured dataset from Statcast. Following this, we will conduct an exploratory data analysis (EDA) to identify trends, anomalies, and patterns over the five-year period. This EDA will involve various statistical tools and visualization techniques using R packages like `dplyr` for data manipulation and `ggplot2` for graphical representations. The exploration will further be detailed into comparative analyses between the two pitchers, aiming to understand individual and relative performance metrics. Ultimately, by charting the performances of Jacob deGrom, this project seeks to offer valuable insights into the factors that contribute to a pitcher's success and longevity in MLB. The findings will serve to enrich our understanding of sports analytics and could potentially inform coaching strategies, player development, and broader analytical approaches within the realm of professional baseball.

II. Data Cleaning & Summaries

In the course of preparing the dataset for comprehensive analysis, several key pre-processing steps were undertaken to ensure the integrity and usability of the data. Initially, the dataset was cleansed of deprecated columns such as `spin_dir` and `spin_rate_deprecated`, which were consistently populated with NA values and are no longer tracked by MLB's Statcast. Attention was also given to handling missing values, a crucial step considering the substantial number of entries with incomplete data. Wherever possible, fields not critical to the analysis that contained missing data were either removed or adjusted to retain only the informative portions. Furthermore, the `game_date` column was converted from a string to a date format to facilitate temporal analyses. Additional refinements included filtering out records missing essential data

like pitch type or release speed and renaming and reorganizing columns to enhance clarity and logical grouping. These pre-processing activities were essential to refine the dataset, enabling a focused and detailed exploration of Jacob deGrom's pitching performance from 2016 to 2021. The dataset encapsulates detailed pitch-level information for MLB pitcher Jacob deGrom over the period from 2016 to 2021. It consists of 42,727 observations across 94 distinct variables, highlighting the comprehensive nature of data collection in modern baseball analytics. The variables encompass a range of metrics from basic game data (game date, home and away teams, pitch type) to advanced Statcast metrics like release speed, spin rate, and launch angles. Release speeds in the dataset vary from a minimum of 59.90 mph to a maximum of 103.80 mph, demonstrating deGrom's dynamic pitching range. The pitch types recorded include sliders, curveballs, and fastballs among others, with an average release position horizontally near the center but varying vertically between 0.14 and 10.03. Additionally, player actions such as strikeouts, hits, and other in-game events are meticulously tracked, providing a rich, granular view of performance across multiple seasons. This dataset not only allows for detailed analysis of deGrom's pitching but also offers insights into the outcomes of his pitches, contributing significantly to evaluations of his techniques and strategy adjustments over the years.

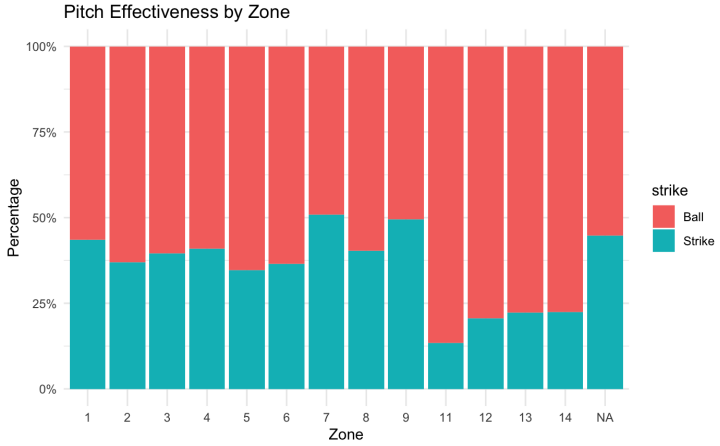
III. Methodology

In order to extract the data for deGrom for the duration time period of 2016 - 2021 various R libraries and algorithms were utilized. The application of the library(baseballr) enables access to analyzing and acquiring baseball data from various online sources such as Baseball Reference, FanGraphs, and the MLB Stats API. Furthermore, the use of library(dplyr) allows for the transformation and restructuring of tabular data to a cleaner and more readable format. In conjunction with library(dplyr), library (ggplot2) is employed in order to create graphical representations of the data. Prior to the creation of graphical representations, library(reshape2) converts data from long format to wide format. (Long format has values that do repeat in the first column, while wide format has values that do not repeat in the first column). Lastly, to aid in the time series analysis and manipulation library(zoo) was adopted, library(zoo) was extremely vital for the completion of the analysis of this project as library(zoo) is specifically curated for irregular time series of numeric vectors, matrices, and factors. It is important to note that the various R libraries were used in relation with one another, as opposed to sequentially, (and in some instances the use of certain libraries was prioritized over the use of others).

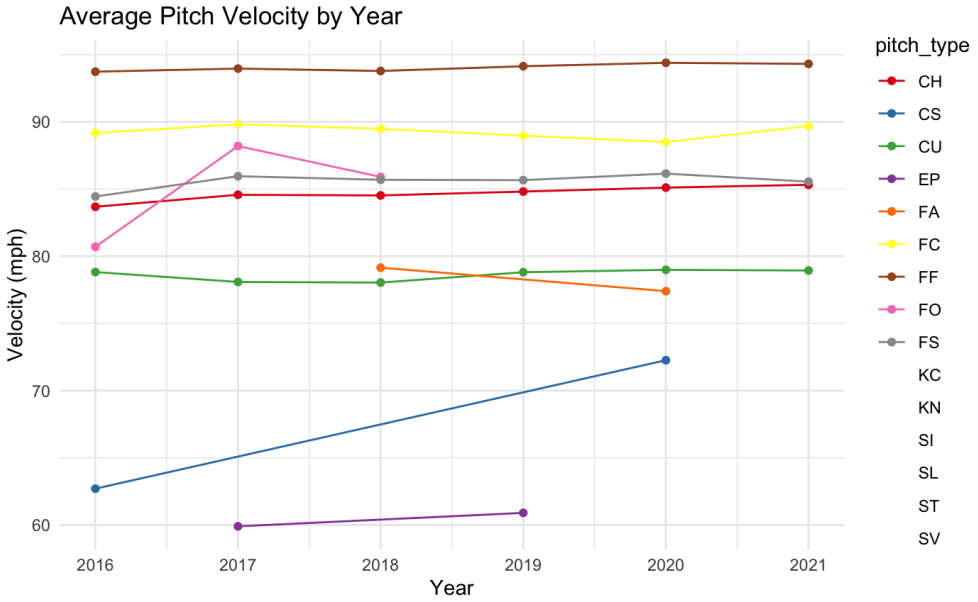
IV. Results & Discussion

This section of the report delves into the findings derived from the detailed analysis of Jacob deGrom's pitching data spanning multiple seasons. We have explored trends and patterns in pitch types, velocities, and outcomes to provide a comprehensive evaluation of deGrom's pitching dynamics and effectiveness throughout his career. Through graphical representations and statistical analysis, we discuss the implications of these results in the context of pitching

strategies and game outcomes. The insights gained not only highlight deGrom's strengths and areas for potential improvement but also play a role in enhancing the player's performance.

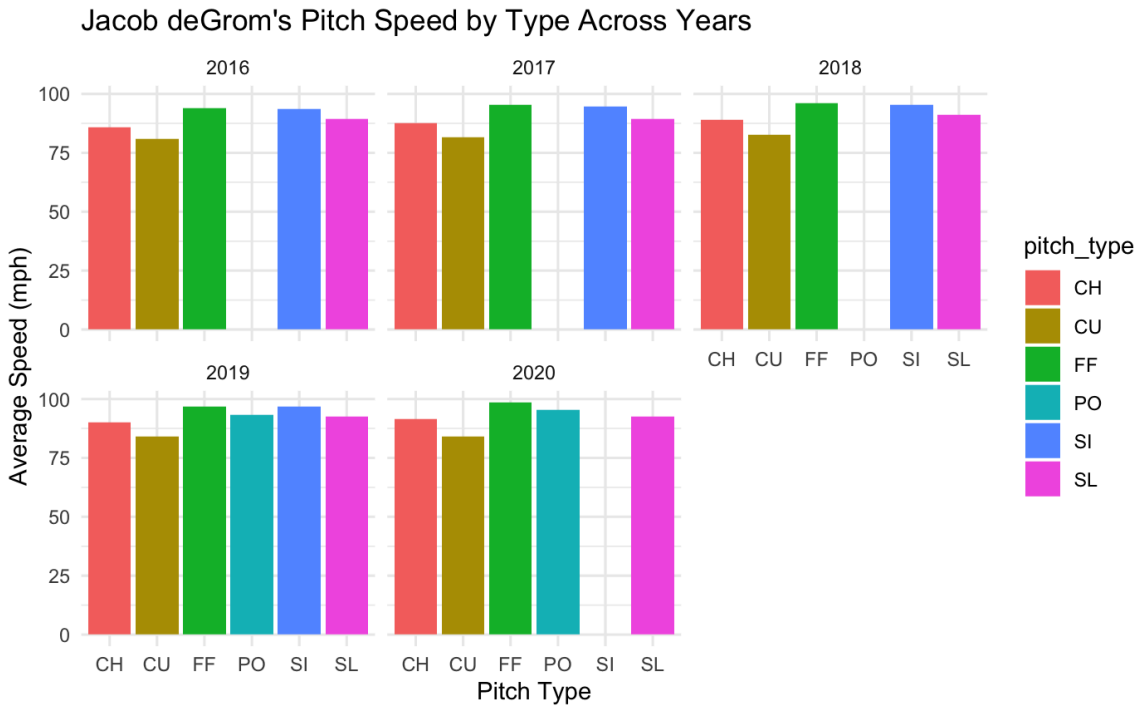


As depicted in the bar plot, the pitch effectiveness for deGrom varies drastically for the various zones. For example, in zone 1 deGrom has a strike percentage that is slightly below 50% (approximately 45%). Conversely, zone 11 has the lowest strike percentage of approximately 15%. Additionally, it is important to note that zones 1 through 9 all have comparable pitch effectivenesses centered around 50%. However, zones 11 through 14 have the lowest pitch effectiveness all below 25%. Therefore, it can be reasonably concluded that deGrom needs to train his pitch effectiveness on these zones to become a well rounded, effective pitcher.

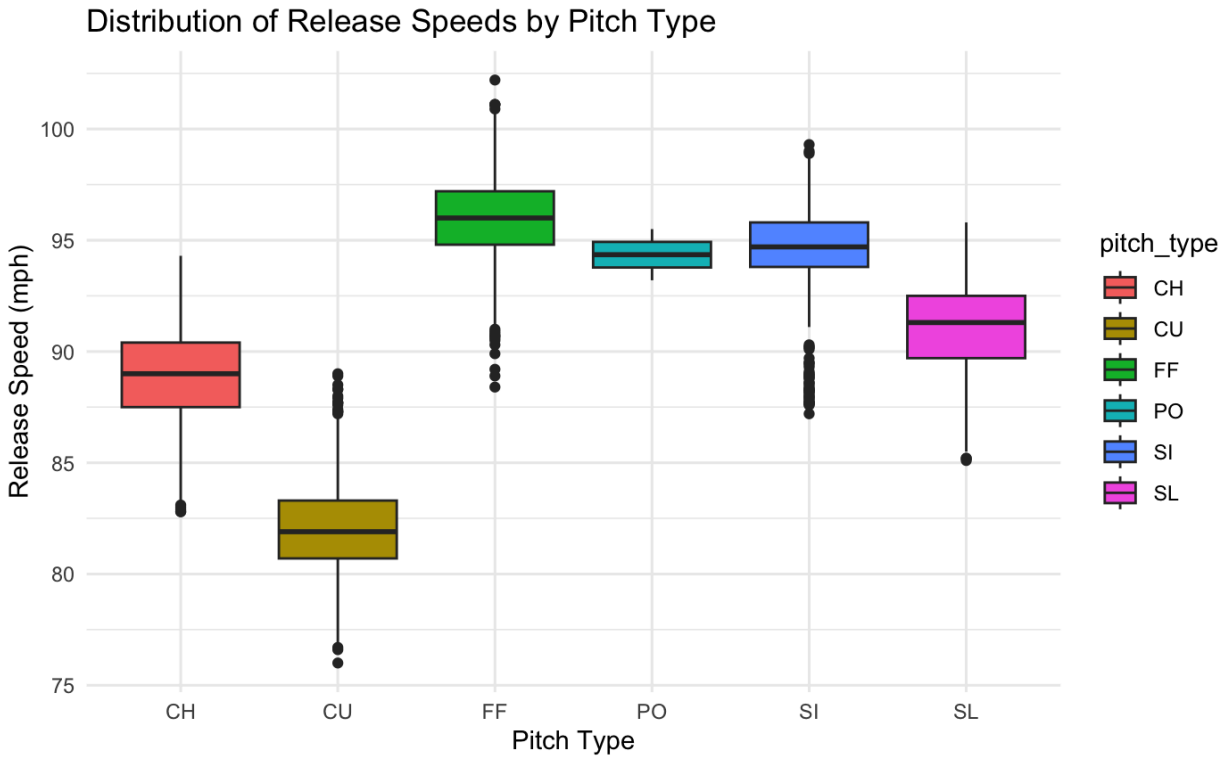


The line plot above demonstrates the various types of pitches completed by deGrom. ("**CH**: changeup pitch is a slow, off-speed pitch that's thrown with the same trajectory as a fastball, but at a slower speed. **CS**: a breaking ball designed to move in the opposite direction of just about

every other breaking pitch. **CU**: a curveball, a breaking pitch that has more movement than just about any other pitch. **EP**: eephus is one of the rarest pitches thrown in baseball, and it is known for its exceptionally low speed and ability to catch a hitter off guard. **FA**: A four-seam fastball is almost always the fastest and straightest pitch a pitcher throws. It is also generally the most frequently utilized. The four-seam fastball is typically one of the easiest pitches for a pitcher to place, because of the lack of movement on the pitch. **FC**: A cutter is a version of the fastball, designed to move slightly toward the pitcher's glove side as it reaches home plate. **FF**: four-seam fastball, it is also generally the most frequently utilized. The four-seam fastball is typically one of the easiest pitches for a pitcher to place, because of the lack of movement on the pitch. **FO**: forkball One of the rarest pitches in baseball, the forkball is a variant of the splitter known for its big downward break as it approaches the plate. Because of the torque involved with snapping off a forkball, it can be one of the more taxing pitches to throw. **FS**: Splitters are often referred to as "split-finger fastballs," but because of their break and lower velocity). From the Average Pitch Velocity by Year line plot it can be asserted that the four seam fastball is deGrom's highest velocity pitch which is above 90 m/s. The pitch with the second highest velocity is the cutter which is approximately 90 m/s. Additionally, from the time period of 2016-2021 both deGrom's fastball and cutter have remained at a consistent velocity. Out of all of the pitches, deGrom's curveball has improved the most, as in 2016 deGrom's curveball had a velocity of 65 m/s while in 2020, deGrom's curveball had a velocity of 75 m/s. The line plot also indicates that deGrom stopped completing certain pitches such as the fastball and the forkball. This may be due to the reasoning that the fastball and forkball are not deGrom's fastest pitches. Additionally, as the forkball is a rare pitch and deGrom's highest velocity pitch is the four-seam fastball he does not have much need for the completion of the fastball or forkball.

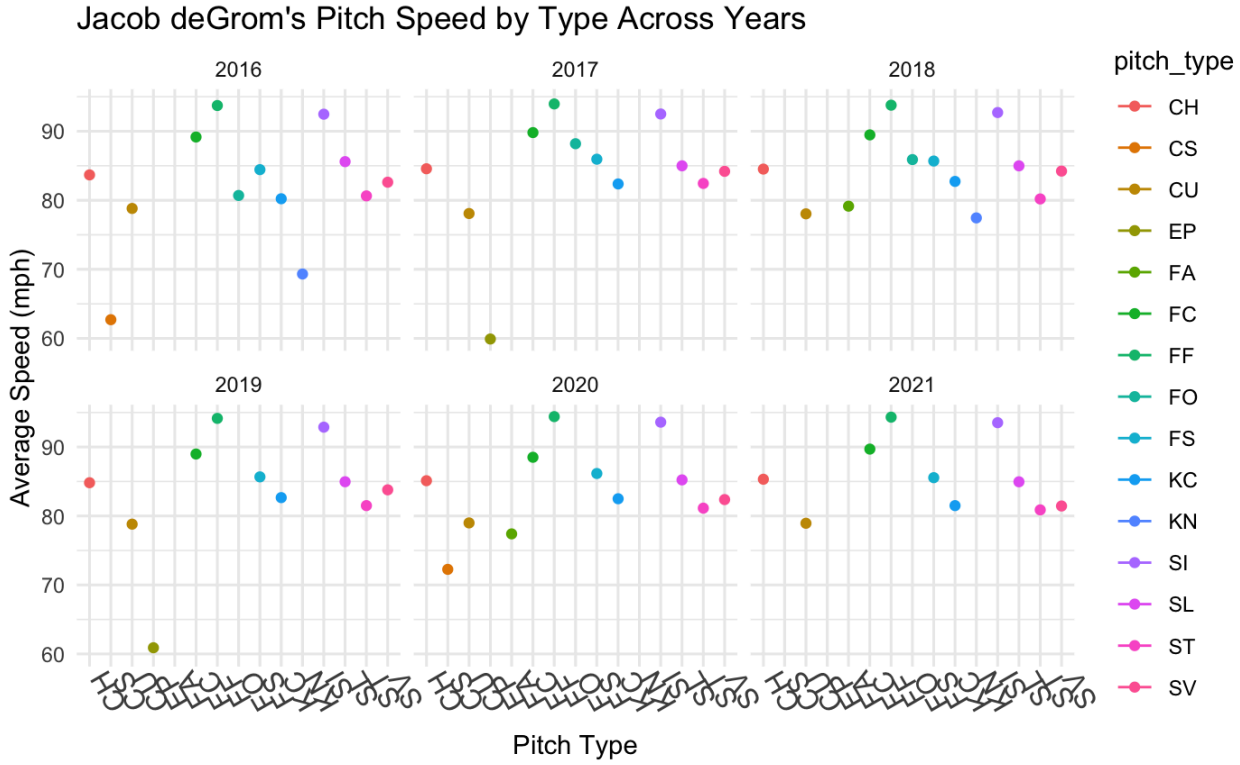


As indicated by the various barplots indicating deGrom's pitch type and their respective average speeds have remained consistent from 2016-2020. The changeup pitch has had an average speed of being between 75 and 100 mph. Similarly, the curveball has remained at a comparable average speed of being between 75 and 90 mph. A notable finding from the bar plots is that between 2016-2018 deGrom did not complete a put out. However, begins performing a put out with an average speed of being between 75 and 100 in both 2019 and 2020. Additionally, the bar plots indicate that the fast ball is deGrom's fastest pitch type, a finding that is substantiated by the *Average Pitch Velocity by Year* line plot. Furthermore, in 2020 deGrom no longer executes the sinker pitch which he completed from 2016-2019. This is a notable observation, as between 2016-2019, the sinker pitch and fastball both had comparable average speeds of being between 75 and 100 mph.

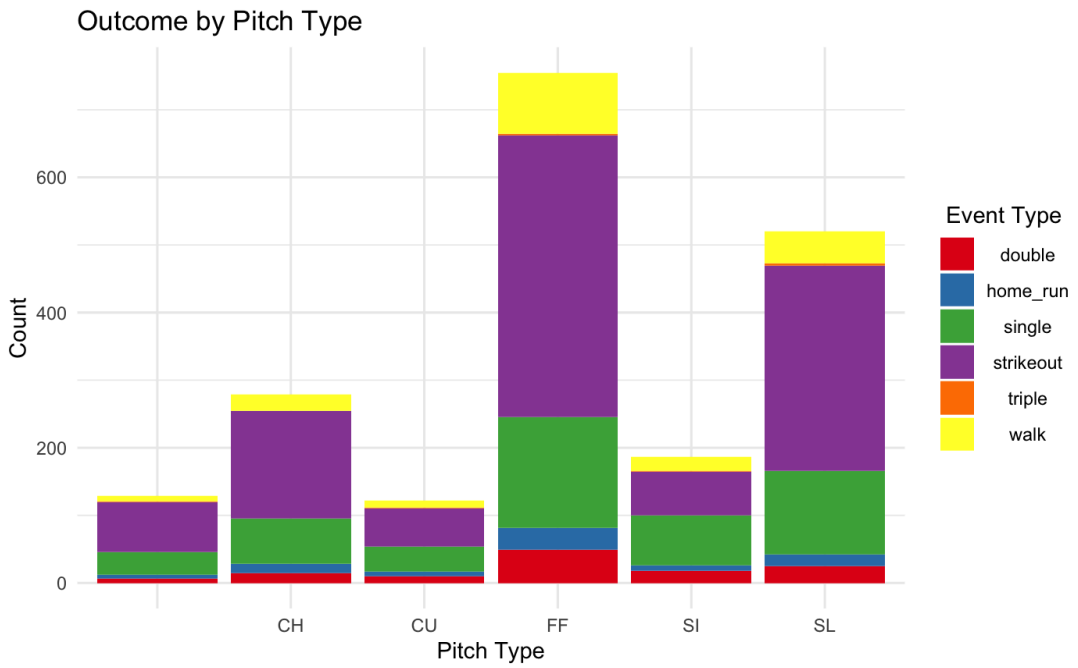


The *Distribution of Release Speeds by Pitch Type* indicates the various release speeds for the following six pitch types. CH: Changeup, CU: Curveball, FF: Fastball, PO: Put Out, SI: Sinker Pitch, and SL: slider. The release speed, otherwise known as the speed in which a ball leaves the pitcher's hand. Initial findings from the box plots indicate that the fastball has the highest release speed of being between 95 and 100 mph, (a discovery that is congruent with the findings from the earlier plots). Conversely, the curveball has the lowest release speed of being between 80 and 85 mph. Additional striking observations include: the put out pitch does not have minimum and

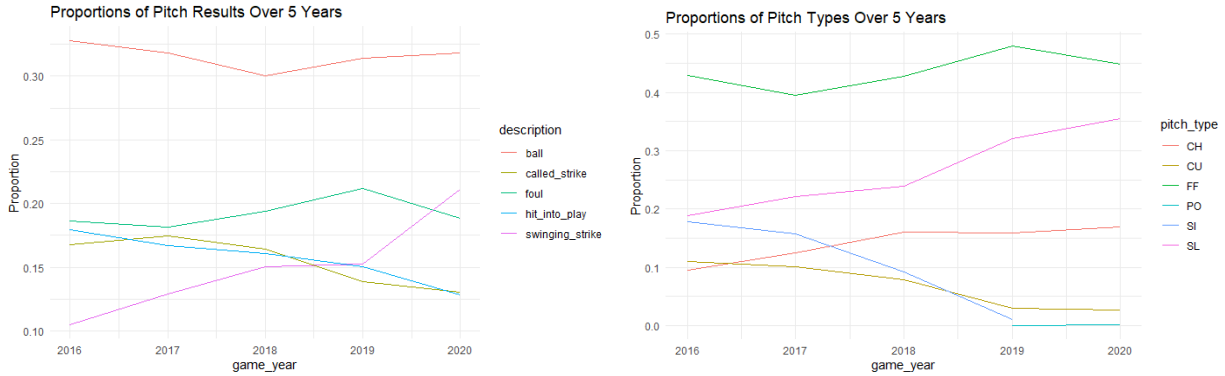
maximum average speed, indicating that the put out pitch has the smallest variation, and further indicating that the lower quartile and upper quartile values are the same as the minimum and maximum release speeds for the put out pitch.



The six scatter plots depict the various average speeds of the many different types of pitches completed by deGrom from 2016-2021. As established by the earlier plots and graphs, the four seam fastball (FF), cut fastball (FC), and forkball (FO) are the pitch types with the fastest average speeds. Throughout the 2016-2021 time period, the four seam fastball, cut fastball, and forkball all had consistent speeds of between 80-100 mph. Conversely, the curveball (CU), sweeper (ST), and slurve (SV) are all pitch types which have the slowest average speeds, in the duration of 2016 to 2021. Interestingly, the four seam fastball remained the fastest pitch type all six years, while different pitch types were the slowest during the same time frame. For instance, in 2016 the changeup pitch was the slowest, while the eephus (EP) was the slowest pitch type in 2017, 2018, and 2019. In 2020, the changeup pitch was once again the pitch type with the slowest average speed, and in 2021 the eephus was the slowest pitch type. Furthermore, in the years 2016, 2017, and 2019 the slowest pitch type had an average speed of approximately 60-65 mph. However, in the years 2018, 2020, and 2021 the slowest pitch type had an average speed of 70-80 mph. Despite the fluctuation in the average speed of the slowest pitch type, the average speed of the fastest pitch type has remained consistent throughout 2016-2021 (of being approximately 90 - 100 mph). A reasonable assertion for the fluctuation in average speed for the slowest pitch types is due to the slowest pitch types being different pitches while the fastest pitch type remained the same during 2016-2021.



The bar plot above depicts the various outcomes for the following pitch types: curveball (CU), changeup (CH), fastball (FF), sinker (SI), and slider (SL). Each of the pitch types have the six possible event types: **double** (when a batter hits the ball into play and reaches second base without the help of an intervening error or attempt to put out another baserunner), **home run** (a batter hits a fair ball and scores on the play without being put out or without the benefit of an error), **single** (when a batter hits the ball and reaches first base without the help of an intervening error or attempt to put out another baserunner), **strikeout** (when a pitcher throws any combination of three swinging or looking strikes to a hitter), **triple** (when a batter hits the ball into play and reaches third base without the help of an intervening error or attempt to put out another baserunner), and **walk** (when a pitcher throws four pitches out of the strike zone, none of which are swung at by the hitter). As established in the earlier plots and graphs, the fastball is deGrom's fastest pitch type and as demonstrated by the bar plot above has the largest amount of strikeouts compared to the other four pitch types. The curveball has the lowest amount of strikeouts, as it is deGrom's slowest pitch type. It may also be of valuable insight to point out that the most frequent outcome for deGrom's various pitch styles is a strikeout, therefore indicating that deGrom is an effective pitcher. The second common pitch outcome is a single, indicating that with deGrom pitching, the opposing team has low scoring potential. Another notable finding is that all of the pitch types, except for the sinker, have a strikeout as being the most probable outcome. Further characterizing deGrom's pitching performance as being superior against competitors.



Diving deeper into deGrom's pitching, we look into the proportions of the pitches and their results, and how they changed year over year. deGrom stayed very consistent in terms of how many pitches were called "ball" – staying between 0.35 and 0.30 over our five year investigation period. In fact, his total proportion of balls decreased slightly from the first year we look at. His number of pitches called strikes fell from about 0.16 to 1.13, but his proportion of swinging strikes increased dramatically, from only 0.10 all the way to 0.22. This is a massive increase, and certainly points towards his gradual improvement in a pitcher, in terms of reliably throwing strikes against players. The number of pitches that were hit into play also steadily decreased from 2016 to 2020, from 0.17 down to 0.13.

His pitching style also changed slightly. While FF (fastball) was by far his favorite, remaining between 0.4 and 0.5 of all his pitches thrown, SL (slider) saw a meteoric rise, from 0.2 to 0.35, contrasting SI's (sinker) sharp decline, from 0.18 to almost 0. Interestingly, the decline in sinkers was a league-wide phenomenon. Mike Petriello reported that even when it was a popular pitch to throw, very few teams were able to actually obtain positive results from throwing it. Batters simply knew how to deal with it, and much better pitches like the slider came along. However, Petriello shares that even this is starting to lose its potency (MLB 2024). CH (changeup) also increased slightly in this time. This difference in pitching distribution could have made deGrom a more unpredictable pitcher to bat against, as players were desperate to hit a fastball out of the park, but were reluctant to swing and miss against a more complicated pitch. Given deGrom's increase in strikeouts over the years, it's reasonable to assume that he expertly varied which pitches he threw in order to maximize the confusion of batters at the plate.

V. Conclusion

The holistic study of Jacob deGrom's on field analytics and performance has culminated in the findings of various valuable insights that may have a profound impact in the further development of his ability. From the diverse analysis that has been conducted it has been established that deGrom has a multifaceted identity as a player in his on field performance. The *Pitch Effectiveness* bar plot revealed that deGrom has a strike rate of approximately 50% for a majority of the fourteen zones. Additionally, it was discovered deGrom's average pitch velocity remained consistent from 2016-2021 for most of his different pitch styles. Despite some slight fluctuations in pitches that deGrom stopped completing after a time period such as the Eephus, which is regarded as being one of the rarest pitches in baseball. Moreover, deGrom's release and pitch speed are analogous at being around 90 -100 mph for his fastball (his fastest pitch style). It can be reasonably concluded that deGrom's high strikeout rate is due to his fast release and pitch speed. The information gathered from the study has indicated deGrom's current capabilities as a pitcher, and more importantly provides a metric for which deGrom can gauge his future goals in improving. One possible area in which deGrom is able to advance as a pitcher is increasing strike out percentages in zones 11 - 14, as in these areas deGrom has a strikeout rate that is below 25%. The methodology and its applicability from this study may be utilized on other players not only in Major League Baseball, but across other sports and organizations as well, such as the National Football League, the National Basketball League, and the National Hockey League. Such research may be greatly incentivized for the various teams in the different sporting leagues, as it may translate in higher win percentages, maximizing existing player potential, and elucidating where existing and future issues lie for players as well as teams. The individual analysis of each player may culminate in how a group of players may collaborate together, and if another player's strengths may balance out the weaknesses of another, and vice versa. Furthermore, juxtaposing two opposing teams and conducting a similar analysis may identify unique information into how one team may outperform the other. Periodically, completing such an in depth analysis will enable for defining new goals for sports teams and players, as well as determining the possibility of a certain achievement. In the case of Jacob deGrom, who just won a World Series title with the Texas Rangers in 2023, the information found through the course of this project assists in comprehending how the Texas Rangers were able to win the World Series title. Possible methods of improving the existing methods and applications in this project include, looking at third party independent data in conjunction with the data found on the Major League Baseball's website as data from the Major League Baseball's database may be slightly skewed and biased. Comparing an independent, third party's data with the Major League Baseball's may also offer awareness into the Major League Baseball's data reporting and notation practices.

References

Bean, R. (2024, February 20). *Moneyball 20 years later: A progress report on data and analytics in professional sports*. Forbes. https://www.forbes.com/sites/randybean/2022/09/18/moneyball-20-years-later-a-progress-report-on-data-and-analytics-in-professional-sports/?sh=5d3fa0ed773d

Bechtold, T. (2021, April 8). State of Analytics: How the movement has forever changed baseball – for better or worse. Stats Perform. https://www.statsperform.com/resource/state-of-analytics-how-the-movement-has-forever-changed-baseball-for-better-or-worse/

Mizels, J., Erickson, B., & Chalmers, P. (2022, August). Current state of data and analytics research in baseball. Current reviews in musculoskeletal medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9276858/

Woltring, M. T., Rost, J. K., & Jubenville, C. B. (2018, October 25). Examining perceptions of baseball's eras: A statistical comparison. The Sport Journal. https://thesportjournal.org/article/examining-perceptions-of-baseballs-eras/

Wyner, A. (2024, April 25). Changing the game: How data analytics is upending baseball. Knowledge at Wharton. https://knowledge.wharton.upenn.edu/podcast/knowledge-at-wharton-podcast/analytics-in-baseball/

Code Appendix

```{r}
library(baseballr)
library(dplyr)
library(ggplot2)
library(reshape2)
library(zoo)
```

```{r}
# updated functions, old function does not work

csv_from_url <- function(...){
  data.table::fread(...)
}
#------------------------------------------------------------------------
-----------------
make_baseballr_data <- function(df, type, timestamp){
  out <- df %>%
    tidyr::as_tibble()

  class(out) <-
c("baseballr_data","tbl_df","tbl","data.table","data.frame")
  attr(out,"baseballr_timestamp") <- timestamp
  attr(out,"baseballr_type") <- type
  return(out)
}
# @title
statcast_search <- function(start_date = Sys.Date() - 1, end_date =
Sys.Date(),
                                    playerid = NULL,
                                    player_type = "batter", ...) {
  # Check for other user errors.
  if (start_date <= "2015-03-01") { # March 1, 2015 was the first
date of Spring Training.
    message("Some metrics such as Exit Velocity and Batted Ball
Events have only been compiled since 2015.")
  }
  if (start_date < "2008-03-25") { # March 25, 2008 was the first
date of the 2008 season.
    stop("The data are limited to the 2008 MLB season and after.")
    return(NULL)
  }
  if (start_date == Sys.Date()) {
```

```
    message("The data are collected daily at 3 a.m. Some of today's
games may not be included.")
  }
  if (start_date > as.Date(end_date)) {
    stop("The start date is later than the end date.")
    return(NULL)
  }

  playerid_var <- ifelse(player_type == "pitcher",
                         "pitchers_lookup%5B%5D",
"batters_lookup%5B%5D")

  vars <- tibble::tribble(
    ~var, ~value,
    "all", "true",
    "hfPT", "",
    "hfAB", "",
    "hfBBT", "",
    "hfPR", "",
    "hfZ", "",
    "stadium", "",
    "hfBBL", "",
    "hfNewZones", "",
    "hfGT", "R%7CPO%7CS%7C&hfC",
    "hfSea", paste0(lubridate::year(start_date), "%7C"),
    "hfSit", "",
    "hfOuts", "",
    "opponent", "",
    "pitcher_throws", "",
    "batter_stands", "",
    "hfSA", "",
    "player_type", player_type,
    "hfInfield", "",
    "team", "",
    "position", "",
    "hfOutfield", "",
    "hfRO", "",
    "home_road", "",
    playerid_var, ifelse(is.null(playerid), "",
as.character(playerid)),
    "game_date_gt", as.character(start_date),
    "game_date_lt", as.character(end_date),
    "hfFlag", "",
    "hfPull", "",
    "metric_1", "",
```

```
    "hfInn", "",
    "min_pitches", "0",
    "min_results", "0",
    "group_by", "name",
    "sort_col", "pitches",
    "player_event_sort", "h_launch_speed",
    "sort_order", "desc",
    "min_abs", "0",
    "type", "details") %>%
    dplyr::mutate(pairs = paste0(.data$var, "=", .data$value))

  if (is.null(playerid)) {
    # message("No playerid specified. Collecting data for all
batters/pitchers.")
    vars <- vars %>%
      dplyr::filter(!grepl("lookup", .data$var))
  }

  url_vars <- paste0(vars$pairs, collapse = "&")
  url <-
paste0("https://baseballsavant.mlb.com/statcast_search/csv?",
url_vars)
  # message(url)

  # Do a try/catch to show errors that the user may encounter while
downloading.
  tryCatch(
    {
      suppressMessages(
        suppressWarnings(
          payload <- csv_from_url(url, encoding ="UTF-8")
        )
      )
    },
    error = function(cond) {
      message(cond)
      stop("No payload acquired")
    },
    # this will never run??
    warning = function(cond) {
      message(cond)
    }
  )
  # returns 0 rows on failure but > 1 columns
  if (nrow(payload) > 1) {
```

```
    names(payload) <- c("pitch_type", "game_date", "release_speed",
"release_pos_x",
                        "release_pos_z", "player_name", "batter",
"pitcher", "events",
                        "description", "spin_dir",
"spin_rate_deprecated", "break_angle_deprecated",
                        "break_length_deprecated", "zone", "des",
"game_type", "stand",
                        "p_throws", "home_team", "away_team", "type",
"hit_location",
                        "bb_type", "balls", "strikes", "game_year",
"pfx_x", "pfx_z",
                        "plate_x", "plate_z", "on_3b", "on_2b",
"on_1b", "outs_when_up",
                        "inning", "inning_topbot", "hc_x", "hc_y",
"tfs_deprecated",
                        "tfs_zulu_deprecated", "fielder_2", "umpire",
"sv_id", "vx0",
                        "vy0", "vz0", "ax", "ay", "az", "sz_top",
"sz_bot", "hit_distance_sc",
                        "launch_speed", "launch_angle",
"effective_speed", "release_spin_rate",
                        "release_extension", "game_pk", "pitcher_1",
"fielder_2_1",
                        "fielder_3", "fielder_4", "fielder_5",
"fielder_6", "fielder_7",
                        "fielder_8", "fielder_9", "release_pos_y",
"estimated_ba_using_speedangle",
                        "estimated_woba_using_speedangle",
"woba_value", "woba_denom",
                        "babip_value", "iso_value",
"launch_speed_angle", "at_bat_number",
                        "pitch_number", "pitch_name", "home_score",
"away_score", "bat_score",
                        "fld_score", "post_away_score",
"post_home_score", "post_bat_score",
                        "post_fld_score", "if_fielding_alignment",
"of_fielding_alignment",
                        "spin_axis", "delta_home_win_exp",
"delta_run_exp", "bat_speed", "swing_length")
    payload <- process_statcast_payload(payload) %>%
      make_baseballr_data("MLB Baseball Savant Statcast Search data
from baseballsavant.mlb.com",Sys.time())
    return(payload)
```

```
  } else {
    warning("No valid data found")


# (somewhere within the statcast_search function before the payload
is searched for)
colos <- c("pitch_type", "game_date",
           "release_speed", "release_pos_x", "release_pos_z",
           "player_name", "batter", "pitcher",
           "events", "description", "spin_dir",
           "spin_rate_deprecated", "break_angle_deprecated",
           "break_length_deprecated", "zone", "des",
           "game_type", "stand", "p_throws",
           "home_team", "away_team", "type",
           "hit_location", "bb_type", "balls",
           "strikes", "game_year", "pfx_x",
           "pfx_z", "plate_x", "plate_z",
           "on_3b", "on_2b", "on_1b", "outs_when_up",
           "inning", "inning_topbot", "hc_x",
           "hc_y", "tfs_deprecated", "tfs_zulu_deprecated",
           "fielder_2", "umpire", "sv_id",
           "vx0", "vy0", "vz0", "ax",
           "ay", "az", "sz_top", "sz_bot",
           "hit_distance_sc", "launch_speed", "launch_angle",
           "effective_speed", "release_spin_rate",
           "release_extension", "game_pk", "pitcher_1",
           "fielder_2_1", "fielder_3", "fielder_4",
           "fielder_5", "fielder_6", "fielder_7",
           "fielder_8", "fielder_9", "release_pos_y",
           "estimated_ba_using_speedangle",
"estimated_woba_using_speedangle",
           "woba_value", "woba_denom", "babip_value",
           "iso_value", "launch_speed_angle", "at_bat_number",
           "pitch_number", "pitch_name", "home_score",
           "away_score", "bat_score", "fld_score",
           "post_away_score", "post_home_score",
           "post_bat_score", "post_fld_score",
"if_fielding_alignment",
           "of_fielding_alignment", "spin_axis",
           "delta_home_win_exp", "delta_run_exp")
colNumber <- ncol(payload)
if(length(colos) != colNumber){
  newCols <- paste("newStat", 1:(length(colos) - colNumber))
  colos <- c(colos, newCols)
```

```
  message("New stats detected! baseballr will be updated soon to
properly identify these stats")
}
# payload is acquired somewhere in here
# when the payload columns need to be named:
names(payload) <- colos



payload <- payload %>%
  make_baseballr_data("MLB Baseball Savant Statcast Search data from
baseballsavant.mlb.com",Sys.time())
    return(payload)
  }
}

statcast_search.default <- function(start_date = Sys.Date() - 1,
end_date = Sys.Date(),
                                                playerid = NULL,
player_type = "batter", ...) {

  message(paste0(start_date, " is not a date. Attempting to
coerce..."))
  start_Date <- as.Date(start_date)

  tryCatch(
    {
      end_Date <- as.Date(end_date)
    },
    warning = function(cond) {
      message(paste0(end_date, " was not coercible into a date. Using
today."))
      end_Date <- Sys.Date()
      message("Original warning message:")
      message(cond)
    }
  )

  statcast_search(start_Date, end_Date,
                          playerid, player_type, ...)

}

statcast_search_batters <- function(start_date, end_date, batterid =
NULL, ...) {
```

```
  statcast_search(start_date, end_date, playerid = batterid,
                          player_type = "batter", ...)
}


statcast_search_pitchers <- function(start_date, end_date, pitcherid
= NULL, ...) {
  statcast_search(start_date, end_date, playerid = pitcherid,
                  player_type = "pitcher", ...)
}
```

```{r}
# Example of using the function to fetch Jacob deGrom's data
deGrom_id <- playerid_lookup("deGrom", "Jacob")$mlbam_id
degrom_data <- statcast_search("2015-01-01", "2019-12-31", playerid =
deGrom_id, player_type = "pitcher")

```

```{r}
# Analysis of deGrom's pitching data
degrom_data %>%
  filter(!is.na(pitch_type), !is.na(release_speed)) %>%
  group_by(year = lubridate::year(game_date), pitch_type) %>%
  summarise(
    Average_Speed = mean(release_speed, na.rm = TRUE),
    Total_Strikeouts = sum(events == "strikeout", na.rm = TRUE),
    .groups = 'drop'  # This will drop all grouping after
summarisation
  ) %>%
  ggplot(aes(x = pitch_type, y = Average_Speed, fill = pitch_type)) +
  geom_bar(stat = "identity") +
  facet_wrap(~year) +
  labs(title = "Jacob deGrom's Pitch Speed by Type Across Years", x =
"Pitch Type", y = "Average Speed (mph)") +
  theme_minimal()
```

```{r}
library(dplyr)
library(baseballr)

fetch_deGrom_data <- function(start_year, end_year) {
```

```
  deGrom_id <- playerid_lookup("deGrom", "Jacob")$key_mlbam[1] #
Fetch deGrom's player ID

  all_data <- lapply(start_year:end_year, function(year) {
    # Statcast search for each year
    statcast_search(start_date = paste0(year, "-01-01"),
                    end_date = paste0(year, "-12-31"),
                    playerid = deGrom_id,
                    player_type = "pitcher")
  }) %>%
    bind_rows() %>%
    filter(launch_speed > 0) # Filter out pitches with zero batted
ball speed

  return(all_data)
}

deGrom_data <- fetch_deGrom_data(2016, 2021)
```

#SUMMARY
```{r}
summary(deGrom_data)
str(deGrom_data)
```

```{r}
data <- deGrom_data %>%
  filter(!is.na(pitch_type) & pitch_type != "" &
!is.na(release_speed))
```

```{r}
deGrom_velocity <- data %>%
  group_by(year = year(game_date), pitch_type) %>%
  summarise(average_velocity = mean(release_speed, na.rm = TRUE),
.groups = 'drop')

ggplot(deGrom_velocity, aes(x = year, y = average_velocity, color =
pitch_type)) +
  geom_line() +
  geom_point() +
  labs(title = "Average Pitch Velocity by Year", x = "Year", y =
"Velocity (mph)") +
  theme_minimal() +
```

```
  scale_color_brewer(palette = "Set1")

```
#STRIKE OUT TRENDS
```{r}
library(lubridate)
library(dplyr)
library(ggplot2)

# Ensure 'game_date' is in date format
data <- data %>%
  mutate(game_date = as.Date(game_date, format = "%Y-%m-%d"),
         year = year(game_date))

# Now perform your summarization
summary_data <- data %>%
  group_by(year, pitch_type) %>%
  summarise(
    Average_Speed = mean(release_speed, na.rm = TRUE),
    .groups = 'drop' # This controls the creation of an extra grouped
layer
  )

ggplot(summary_data, aes(x = pitch_type, y = Average_Speed, color =
pitch_type)) +
  geom_line() +
  geom_point() +
  facet_wrap(~year) +
  labs(title = "Jacob deGrom's Pitch Speed by Type Across Years", x =
"Pitch Type", y = "Average Speed (mph)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 120, hjust = 1, vjust = 1,
size = 12))  # Rotate and adjust the size of x-axis labels


```



```{r}
# Print the names of all columns in the dataset
print(names(deGrom_data))
```



```{r}
```

```
library(baseballr)

# Look up Jacob deGrom's player ID
degrom_lookup <- playerid_lookup("deGrom", "Jacob")
deGrom_id <- degrom_lookup$mlbam_id

# Print IDs to ensure they are loaded
print(deGrom_id)
```

```{r}
# Function to fetch data for a specific pitcher over the years
2016-2021
get_pitcher_data <- function(player_id, start_year, end_year) {
  do.call(rbind, lapply(start_year:end_year, function(year) {
    statcast_search(start_date = paste0(year, "-01-01"), end_date =
paste0(year, "-12-31"),
                    playerid = player_id, player_type = "pitcher")
  }))
}

# Fetching data for Jacob deGrom
degrom_data <- get_pitcher_data(deGrom_id, 2016, 2021)
```

```{r}
degrom_data$outs_recorded <- degrom_data$events %in% c("strikeout",
"field_out", "force_out", "double_play", "grounded_into_double_play",
"strikeout_double_play", "sac_bunt", "sac_fly",
"fielders_choice_out")
outs_total <- sum(degrom_data$outs_recorded)

# Convert total outs to innings pitched (3 outs per inning)
innings_pitched <- outs_total / 3
```

```{r}
# Function to calculate ERA
custom_era_calculation <- function(data) {
  earned_runs <- sum(data$events %in% c("home_run", "scored_by_hit"),
na.rm = TRUE)
  innings_pitched <- sum(data$outs_recorded, na.rm = TRUE) / 3
  if (innings_pitched == 0) return(NA)  # Avoid division by zero
```

```
  (earned_runs / innings_pitched) * 9
}
```

```{r}
custom_whip_calculation <- function(data) {
  walks_hits <- sum(data$events %in% c("walk", "single", "double",
"triple", "home_run"), na.rm = TRUE)
  innings_pitched <- sum(data$innings_pitched, na.rm = TRUE)
  if (innings_pitched == 0) return(NA)  # Avoid division by zero
  walks_hits / innings_pitched
}

```



```{r}
analyze_data <- function(data) {
  data %>%
    group_by(year = lubridate::year(game_date)) %>%
    summarise(
      Average_Velocity = mean(release_speed, na.rm = TRUE),
      Total_Strikeouts = sum(events == "strikeout", na.rm = TRUE),
      ERA = custom_era_calculation(cur_data()),
      WHIP = custom_whip_calculation(cur_data())
    )
}
```



```{r}
degrom_stats <- analyze_data(degrom_data)
print(degrom_stats)
```

```{r}
library(ggplot2)

# Convert 'year' to numeric
degrom_stats$year <- as.numeric(as.character(degrom_stats$year))

# Check and convert 'ERA' and 'WHIP' to numeric
degrom_stats$ERA <- as.numeric(as.character(degrom_stats$ERA))
degrom_stats$WHIP <- as.numeric(as.character(degrom_stats$WHIP))
```

```
# Plotting ERA and WHIP over time
ggplot(degrom_stats, aes(x = year)) +
  geom_line(aes(y = ERA, color = "ERA"), size = 1) +  # Specify color
within aes() for legends
  geom_line(aes(y = WHIP, color = "WHIP"), size = 1) +  # Specify
color within aes() for legends
  scale_color_manual(values = c("ERA" = "blue", "WHIP" = "red")) +
  labs(
    title = "Jacob deGrom: ERA and WHIP Trends (2016-2020)",
    y = "Value",
    x = "Year",
    color = "Metric"  # Proper label for legend
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  #
Improves readability of x-axis labels

```



```{r}
library(baseballr)
library(dplyr)

degrom_lookup <- playerid_lookup("deGrom", "Jacob")
deGrom_id <- degrom_lookup$mlbam_id

get_pitcher_data <- function(player_id, start_year, end_year) {
  do.call(rbind, lapply(start_year:end_year, function(year) {
    statcast_search(start_date = paste0(year, "-01-01"), end_date =
paste0(year, "-12-31"),
                    playerid = player_id, player_type = "pitcher")
  }))
}

# Fetching data for Jacob deGrom
degrom_data <- get_pitcher_data(deGrom_id, 2016, 2021)

```

```{r}
# Convert game_date to Date type
degrom_data$game_date <- as.Date(degrom_data$game_date)
```

```
#Summary
summary_stats <- degrom_data%>%
  group_by(pitch_type) %>%
  summarise(
    Average_Speed = mean(release_speed, na.rm = TRUE),
    Strikeouts = sum(events == "strikeout", na.rm = TRUE),
    Total_Pitches = n()
  )
```

```{r}
# Plotting average pitch speed by pitch type
ggplot(summary_stats, aes(x = pitch_type, y = Average_Speed, fill =
pitch_type)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Pitch Speed by Pitch Type", x = "Pitch Type",
y = "Average Speed (mph)")

# Strikeout rate by pitch type
summary_stats$Strikeout_Rate <- summary_stats$Strikeouts /
summary_stats$Total_Pitches
ggplot(summary_stats, aes(x = pitch_type, y = Strikeout_Rate, fill =
pitch_type)) +
  geom_bar(stat = "identity") +
  labs(title = "Strikeout Rate by Pitch Type", x = "Pitch Type", y =
"Strikeout Rate")
```

```{r}
#Pitch Velocity Over Time
#This visualization will help observe changes in Jacob deGrom's pitch
velocity across games.
ggplot(degrom_data, aes(x = game_date, y = release_speed, color =
pitch_type)) +
  geom_line() +
  labs(title = "Pitch Velocity Over Time", x = "Game Date", y =
"Velocity (mph)") +
  theme_minimal()

```

```{r}
#Distribution of Pitch Types
```

```
#See the frequency of different pitch types used by deGrom.

ggplot(degrom_data, aes(x = pitch_type, fill = pitch_type)) +
  geom_bar() +
  labs(title = "Distribution of Pitch Types", x = "Pitch Type", y =
"Count") +
  theme_minimal()

```


```{r}
library(dplyr)
library(ggplot2)

significant_outcomes <- c("strikeout", "home_run", "single",
"double", "triple", "walk")

# Filter data for significant outcomes and plot
degrom_data %>%
  filter(events %in% significant_outcomes) %>%
  group_by(pitch_type, events) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ggplot(aes(x = pitch_type, y = count, fill = events)) +
  geom_bar(stat = "identity", position = position_stack(reverse =
TRUE)) +
  scale_fill_brewer(palette = "Set1") +  # Using a color palette that
is easy to distinguish
  labs(title = "Outcome by Pitch Type", x = "Pitch Type", y =
"Count") +
  guides(fill = guide_legend(title = "Event Type", title.position =
"top", title.hjust = 0.5, label.hjust = 0.5)) +
  theme_minimal() +
  theme(legend.position = "right", legend.title.align = 0.5)  #
Adjusting legend position and alignment
```


```{r}
degrom_data %>%
  mutate(strike = ifelse(description %in% c("called_strike",
"swinging_strike"), "Strike", "Ball")) %>%
  ggplot(aes(x = factor(zone), fill = strike)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
```

```
  labs(title = "Pitch Effectiveness by Zone", x = "Zone", y =
"Percentage") +
  theme_minimal()

```

```{r}
# Check structure of both columns
str(degrom_data$release_speed)
str(degrom_data$spin_rate_deprecated)

# Count NA values in the spin_rate_deprecated
sum(is.na(degrom_data$spin_rate_deprecated))

summary(degrom_data$spin_rate_deprecated)
```

```{r}
degrom_data <- degrom_data %>%
  filter(pitch_type != "")

# Visualize the distribution of release speeds by pitch type
ggplot(degrom_data, aes(x = pitch_type, y = release_speed, fill =
pitch_type)) +
  geom_boxplot() +
  labs(title = "Distribution of Release Speeds by Pitch Type", x =
"Pitch Type", y = "Release Speed (mph)") +
  theme_minimal()

# Examine outcomes by pitch type
degrom_data %>%
  group_by(pitch_type, events) %>%
  summarise(count = n(), .groups = 'drop') %>%
  ggplot(aes(x = pitch_type, y = count, fill = events)) +
  geom_bar(stat = "identity", position = position_stack()) +
  labs(title = "Outcomes by Pitch Type", x = "Pitch Type", y =
"Count")

```

```{r}
# calculate the total number of pitches
total_pitches_per_year <- df %>%
  group_by(game_year) %>%
  summarize(total_pitches = n())
```

```
# calculate the proportions at which each description occurs
description_proportions <- df %>%
  group_by(game_year, description) %>%
  # we do a count
  summarize(count = n()) %>%
  ungroup() %>%
  # and observe over each year
  left_join(total_pitches_per_year, by = "game_year") %>%
  mutate(proportion = count / total_pitches) %>%
  select(game_year, description, proportion)

# and filter out anything less than a rate of 0.05, as it is too
small for meaningful conclusions
description_proportions_filtered <- description_proportions %>%
  filter(proportion >= 0.05)

# Calculate proportions for "pitch_type"
# and follow the same process as with desctiptions
pitch_type_proportions <- df %>%
  group_by(game_year, pitch_type) %>%
  summarize(count = n()) %>%
  ungroup() %>%
  left_join(total_pitches_per_year, by = "game_year") %>%
  mutate(proportion = count / total_pitches) %>%
  select(game_year, pitch_type, proportion)

# Plot for "description" proportions
ggplot(description_proportions_filtered, aes(x = game_year, y =
proportion, color = description)) +
  geom_line() +
  labs(title = "Proportions of Pitch Results Over 5 Years",
       x = "game_year",
       y = "Proportion") +
  theme_minimal()

# Plot for "pitch_type" proportions
ggplot(pitch_type_proportions, aes(x = game_year, y = proportion,
color = pitch_type)) +
  geom_line() +
  labs(title = "Proportions of Pitch Types Over 5 Years",
       x = "game_year",
       y = "Proportion") +
  theme_minimal()
```