

# Catch probability data

Tim White

2025-01-28

```
library(tidyverse)
library(rjson)
library(baseballr)
library(rvest)
```

We scrape individual play data for every fly ball hit to an outfielder during a particular season.

```
scrape_of_catch_prob_data <- function(year) {
  # Load in player IDs
  playerIDs <- read_csv(paste0("../data/of_playerIDs_", year, ".csv")) %>%
    select(player_id)

  # Scrape play-by-play data
  data <- lapply(1:nrow(playerIDs),
    function(j) {
      # Scrape data
      rawdata <- fromJSON(
        file = paste0("https://baseballsavant.mlb.com/player-services/range?playerId=",
          playerIDs[j,], "&season=", year,
          "&playerType=outfielder"), simplify = TRUE
      )

      # If the URL exists:
      if (length(rawdata) > 0) {

        # Change any null columns (e.g., sprint_speed) to NA
        for (k in 1:length(rawdata)) {
          rawdata[[k]][sapply(rawdata[[k]], is.null)] <- NA
        }

        # Convert raw data to tibble
        tibble(data.frame(matrix(unlist(rawdata),
                                nrow = length(rawdata),
                                byrow = TRUE,
                                dimnames = list(1:length(rawdata),
                                                  names(rawdata[[1]])))) %>%
          mutate(across(c(game_pk:name_display_first_last, pos),
                        as.factor),
                 across(c(stars:distance, hang_time, out:sprint_speed),
                        as.numeric))
      )
    }
  }
```

```

    }
  )

  return(tibble(do.call(rbind.data.frame, data)))
}

clean_of_catch_prob_data <- function(year) {
  of_catch_prob <- read_csv(paste0("../data/of_catch_prob_",
    year, "_orig.csv")) %>%
    filter(!is.na(sprint_speed)) %>%
    mutate(game_pk = as.factor(game_pk),
           play_id = as.factor(play_id),
           player = as.factor(name_display_first_last),
           pos = as.factor(case_when(
             pos == 7 ~ "LF",
             pos == 8 ~ "CF",
             pos == 9 ~ "RF"
           ))) %>%
    select(-name_display_first_last) %>%
    mutate(a1 = -start_pos_x, a2 = -start_pos_y,
           b1 = landing_pos_x - start_pos_x,
           b2 = landing_pos_y - start_pos_y) %>%
    mutate(eta = atan2(a1*b2 - a2*b1, a1*b1 + a2*b2) * 180/pi) %>%
    mutate(eta_adjusted = eta * (eta >= 0) + (360 + eta) * (eta < 0)) %>%
    mutate(route_angle = (270 + eta_adjusted) %% 360) %>%
    mutate(route_direction = as.factor(case_when(
      route_angle >= 0 & route_angle < 60 ~ "back_right",
      route_angle >= 60 & route_angle < 120 ~ "back_middle",
      route_angle >= 120 & route_angle < 180 ~ "back_left",
      route_angle >= 180 & route_angle < 240 ~ "in_left",
      route_angle >= 240 & route_angle < 300 ~ "in_middle",
      route_angle >= 300 & route_angle < 360 ~ "in_right",
    ))) %>%
    mutate(ball_distance = sqrt(landing_pos_x^2 + landing_pos_y^2)) %>%
    mutate(ball_angle = -atan2(landing_pos_y, landing_pos_x) * 180 / pi + 90) %>%
    mutate(ball_section = as.factor(case_when(
      ball_angle <= -33.75 ~ "LF",
      ball_angle <= -11.25 ~ "LC",
      ball_angle <= 11.25 ~ "CF",
      ball_angle <= 33.75 ~ "RC",
      ball_angle > 33.75 ~ "RF"
    )))

  teams <- fg_batter_leaders(startseason = year, endseason = year) %>%
    mutate(team_name = as.factor(ifelse(team_name == "- - -",
      "MUL", team_name))) %>%
    select(player = PlayerName, team = team_name)
  of_catch_prob_teams <- teams %>% right_join(of_catch_prob, by = "player")

  stadiums <- mlb_schedule(season = year) %>%
    mutate(game_pk = as.factor(game_pk),
           stadium = as.factor(venue_name)) %>%
    select(game_pk, stadium)

```

```

of_catch_prob_teams_stadiums <- of_catch_prob_teams %>%
  left_join(stadiums, by = "game_pk")

dimensions_table <- read_html("http://www.andrewclem.com/Baseball/Stadium_statistics.html") %>%
  html_elements("table") %>%
  html_table()

dimensions_colnames <- dimensions_table[[1]] %>% janitor::row_to_names(row_number=1) %>%
  select(stadium = "Stadium (see notes)",
         wall_height_lf = "LF", wall_height_cf = "CF",
         wall_height_rf = "RF", wall_dist_lf = "Left field",
         wall_dist_lc = "Left-center", wall_dist_cf = "Center field",
         wall_dist_rc = "Right-center", wall_dist_rf = "Right field") %>%
  colnames()

dimensions_data <- dimensions_table[[2]] %>%
  select(X1, X12, X13, X14, X17, X18, X19, X20, X21) %>%
  rename_with(~dimensions_colnames) %>%
  mutate(stadium = as.factor(stadium)) %>%
  mutate(stadium = str_remove_all(stadium, " \\*")) %>%
  mutate(across(contains("wall_"),
                function(col) {str_remove_all(col, "[()\\[\\]]")})) %>%
  mutate(across(contains("wall_"), as.numeric)) %>%
  mutate(stadium = fct_recode(stadium,
                              "Old Yankee Stadium" = "Yankee Stadium",
                              "Yankee Stadium" = "Yankee Stadium II",
                              "American Family Field" = "Miller Park",
                              "Busch Stadium" = "Busch Stadium III",
                              "Great American Ball Park" = "Great American Ballpark",
                              "loanDepot park" = "Marlins Park",
                              "Petco Park" = "PETCO Park",
                              "T-Mobile Park" = "Safeco Field",
                              "Truist Park" = "Truist (ex-SunTrust) Park"))

of_catch_prob_teams_stadiums_dims <- of_catch_prob_teams_stadiums %>%
  left_join(dimensions_data, by = "stadium") %>%
  mutate(wall_dist_lflc = 0.5 * (wall_dist_lf + wall_dist_lc),
         wall_dist_lccf = 0.5 * (wall_dist_lc + wall_dist_cf),
         wall_dist_cfrf = 0.5 * (wall_dist_cf + wall_dist_rc),
         wall_dist_rcrf = 0.5 * (wall_dist_rc + wall_dist_rf)) %>%
  mutate(wall_dist_lf_x = wall_dist_lf * cos((90 + 45) * pi / 180),
         wall_dist_lf_y = wall_dist_lf * sin((90 + 45) * pi / 180),
         wall_dist_lflc_x = wall_dist_lflc * cos((90 + 33.75) * pi / 180),
         wall_dist_lflc_y = wall_dist_lflc * sin((90 + 33.75) * pi / 180),
         wall_dist_lc_x = wall_dist_lc * cos((90 + 22.5) * pi / 180),
         wall_dist_lc_y = wall_dist_lc * sin((90 + 22.5) * pi / 180),
         wall_dist_lccf_x = wall_dist_lccf * cos((90 + 11.25) * pi / 180),
         wall_dist_lccf_y = wall_dist_lccf * sin((90 + 11.25) * pi / 180),
         wall_dist_cf_x = wall_dist_cf * cos((90 + 0) * pi / 180),
         wall_dist_cf_y = wall_dist_cf * sin((90 + 0) * pi / 180),
         wall_dist_cfrf_x = wall_dist_cfrf * cos((90 - 11.25) * pi / 180),
         wall_dist_cfrf_y = wall_dist_cfrf * sin((90 - 11.25) * pi / 180),
         wall_dist_rc_x = wall_dist_rc * cos((90 - 22.5) * pi / 180),
         wall_dist_rc_y = wall_dist_rc * sin((90 - 22.5) * pi / 180),
         wall_dist_rcrf_x = wall_dist_rcrf * cos((90 - 33.75) * pi / 180),
         wall_dist_rcrf_y = wall_dist_rcrf * sin((90 - 33.75) * pi / 180))

```

```

        wall_dist_rf_x = wall_dist_rf * cos((90 - 45) * pi / 180)
        wall_dist_rf_y = wall_dist_rf * sin((90 - 45) * pi / 180)
    mutate(dist_to_lf = sqrt(
        (landing_pos_x - wall_dist_lf_x)^2 + (landing_pos_y - wall_dist_lf_y)^2
    ),
    dist_to_lflc = sqrt(
        (landing_pos_x - wall_dist_lflc_x)^2 + (landing_pos_y - wall_dist_lflc_y)^2
    ),
    dist_to_lc = sqrt(
        (landing_pos_x - wall_dist_lc_x)^2 + (landing_pos_y - wall_dist_lc_y)^2
    ),
    dist_to_lccf = sqrt(
        (landing_pos_x - wall_dist_lccf_x)^2 + (landing_pos_y - wall_dist_lccf_y)^2
    ),
    dist_to_cf = sqrt(
        (landing_pos_x - wall_dist_cf_x)^2 + (landing_pos_y - wall_dist_cf_y)^2
    ),
    dist_to_cfrf = sqrt(
        (landing_pos_x - wall_dist_cfrf_x)^2 + (landing_pos_y - wall_dist_cfrf_y)^2
    ),
    dist_to_rc = sqrt(
        (landing_pos_x - wall_dist_rc_x)^2 + (landing_pos_y - wall_dist_rc_y)^2
    ),
    dist_to_rcrf = sqrt(
        (landing_pos_x - wall_dist_rcrf_x)^2 + (landing_pos_y - wall_dist_rcrf_y)^2
    ),
    dist_to_rf = sqrt(
        (landing_pos_x - wall_dist_rf_x)^2 + (landing_pos_y - wall_dist_rf_y)^2
    ),
    mutate(closest = pmin(dist_to_lf, dist_to_lflc, dist_to_lc, dist_to_lccf,
        dist_to_cf, dist_to_cfrf, dist_to_rc, dist_to_rcrf, dist_to_rf),
    mutate(relevant_wall_distance = case_when(
        dist_to_lf == closest ~ wall_dist_lf,
        dist_to_lflc == closest ~ wall_dist_lflc,
        dist_to_lc == closest ~ wall_dist_lc,
        dist_to_lccf == closest ~ wall_dist_lccf,
        dist_to_cf == closest ~ wall_dist_cf,
        dist_to_cfrf == closest ~ wall_dist_cfrf,
        dist_to_rc == closest ~ wall_dist_rc,
        dist_to_rcrf == closest ~ wall_dist_rcrf,
        dist_to_rf == closest ~ wall_dist_rf
    )) %>%
    mutate(closest_wall = as.factor(case_when(
        dist_to_lf == closest ~ "LF",
        dist_to_lflc == closest ~ "LFLC",
        dist_to_lc == closest ~ "LC",
        dist_to_lccf == closest ~ "LCCF",
        dist_to_cf == closest ~ "CF",
        dist_to_cfrf == closest ~ "CFRC",
        dist_to_rc == closest ~ "RC",
        dist_to_rcrf == closest ~ "RCRF",
        dist_to_rf == closest ~ "RF"
    ))) %>%
    select(-contains("norm_start_pos"),
        -a1, -a2, -b1, -b2, -eta, -eta_adjusted,
        -contains("dist_to_"), -closest)

    return(of_catch_prob_teams_stadiums_dims)
}

```

```

of_catch_prob_2024_orig <- scrape_of_catch_prob_data("2024")
of_catch_prob_2024_orig %>% write_csv("../data/of_catch_prob_2024_orig.csv")

```

```
of_catch_prob_2024 <- clean_of_catch_prob_data("2024")
of_catch_prob_2024 %>% write_csv("../data/of_catch_prob_2024.csv")

of_catch_prob_2023_orig <- scrape_of_catch_prob_data("2023")
of_catch_prob_2023_orig %>% write_csv("../data/of_catch_prob_2023_orig.csv")
of_catch_prob_2023 <- clean_of_catch_prob_data("2023")
of_catch_prob_2023 %>% write_csv("../data/of_catch_prob_2023.csv")
```