Mlb_At_Bat_Simulator

October 18, 2024

1 Simulating 2024 Aaron Judge vs 2024 Shohei Ohtani by: Jordan Wolfe

1.0.1

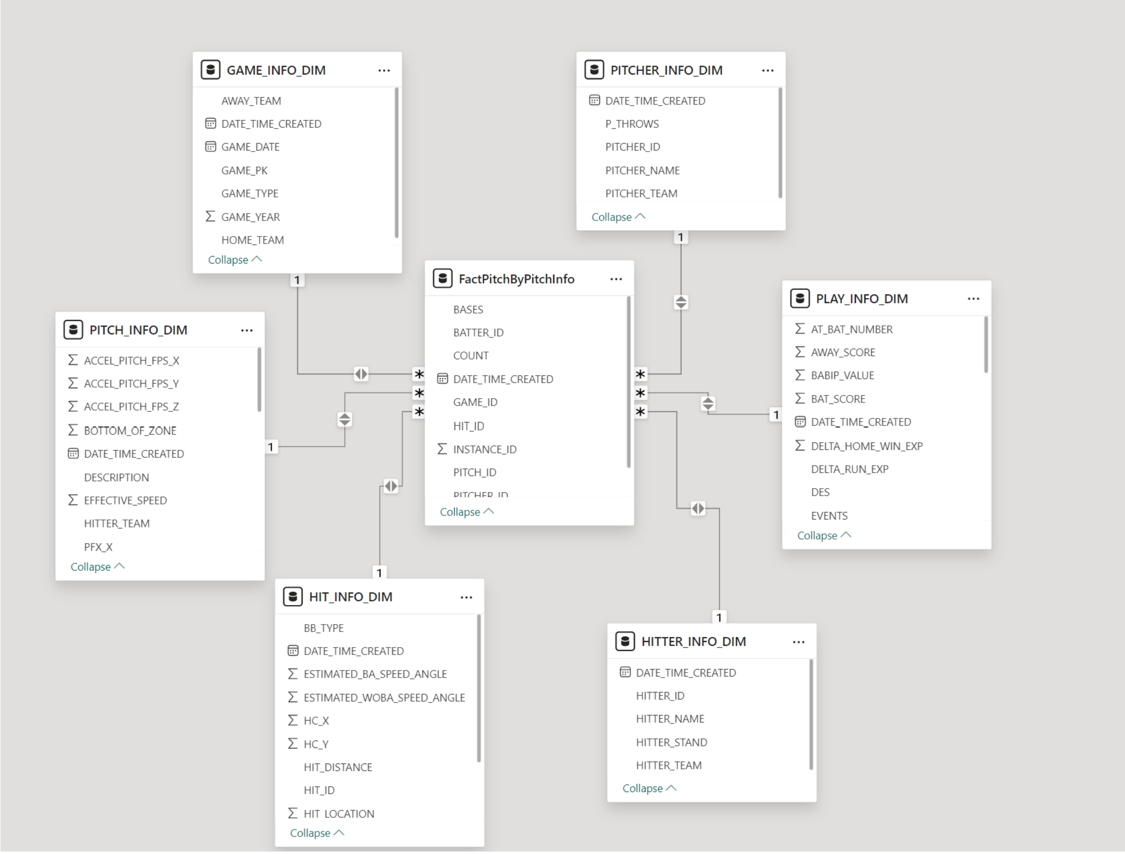
Aaron Judge and Shohei Ohtani have had great seasons so far in their 2024 campaigns and are currently in ALCS and NLCS competing to face off in the World Series. That made me think of the question: Which player would have better performance if they

were to swap teams. I will use a Markovian Style Simulation based on their 2023 and 2024 Play-By-Play along with the One-Way Anova Test to try to solve this (*Note Baserunning is ignored in this simplified model so sorry Ohtani :))

[21]:
```
# All imports
library(DBI)
library(odbc)
library(tidyverse)
library(baseballr)
library(stringr)
```

1.1 Step 1: Load the 2023 and 2024 Season Pitch by Pitch Data

1.1.1



1.1.2 This is the database that I have for the pitch by pitch data that I pull from https://baseballsavant.mlb.com/, for this here we are going to need the Fact table, Play Info, Game Info, and hitter info. From those I filter out plays that have the truncated_pa event, walkoff plays (since this would result in Null Bases After which would hurt the simulation program), and I also filter for only 2023 and 2024 games

[22]:
```
con <- dbConnect(odbc(),
                 Driver = "SQL Server",
                 Server = "JORDANS_LAPTOP",
                 Database = "DW_MLB_PITCH_BY_PITCH",
                 Trusted_Connection = "Yes")     # Use Windows Authentication

fact_pitch <- dbGetQuery(con, "SELECT BATTER_ID, COUNT, RS_ON_PLAY,␣
 ↪BASES_AFTER, BASES, PLAY_ID, GAME_ID FROM FactPitchByPitchInfo WHERE␣
 ↪BASES_AFTER IS NOT NULL")

play_info <- dbGetQuery(con, "SELECT PLAY_ID, OUTS_WHEN_UP , EVENTS FROM␣
 ↪PLAY_INFO_DIM WHERE EVENTS != 'truncated_pa'")

game_info <- dbGetQuery(con, "SELECT GAME_PK, GAME_YEAR FROM GAME_INFO_DIM␣
 ↪WHERE GAME_YEAR >= 2023")
hitter_info <- dbGetQuery(con, "SELECT DISTINCT HITTER_ID, HITTER_NAME FROM␣
 ↪HITTER_INFO_DIM;")
# Close the database connection after loading the data
dbDisconnect(con)
```

1.1.3 This joins the tables together to speed up the simulation efforts

[23]:

```
joined_data <- fact_pitch %>%
    inner_join(play_info, by = "PLAY_ID") %>%
    inner_join(game_info, by = c("GAME_ID" = "GAME_PK"))
```

1.2 Step 2: Create a Process to randomly select plays for a batter depending on the base situation and the number of Outs

1.2.1 For this I used a somewhat Markovian Chain-Like Model that will essentially for every at-bat situation take a random event from the batter given the at bat and number of outs from the joined table. It is also important to note that this Markovian-Like model assumes no stolen bases which does have some effect on the true run value.

[24]:

```
get_random_event_for_batter_2023_2024 <- function(batter, bases, outs) {
  # Filter the tables based on the input parameters
  joined_filtered <- joined_data %>%
    filter(BATTER_ID == batter, BASES == bases, OUTS_WHEN_UP == outs)
  if (nrow( joined_filtered ) == 0) return(NULL)  # Return NULL if no records␣
 ↪are found
  # Select a random row
  random_event <-  joined_filtered  %>%
    sample_n(1)  # Randomly pick 1 row
  # Create the output tibble
  output <- tibble(
    outs = outs_map[[random_event$EVENTS]],  # Look up the outs using the events
    batter = batter,
    new_bases = random_event$BASES_AFTER,
    runs_scored = random_event$RS_ON_PLAY,
    event = random_event$EVENTS
  )

  return(output)
}
```

[25]:

```
player_map <- setNames(as.character(hitter_info$HITTER_NAME), as.
  ↪character(hitter_info$HITTER_ID))
player_map <- as.list(player_map)
# Create a named list (dictionary) for play types and their associated outs
outs_map <- list(
  "double" = 0,
  "double_play" = 2,
  "field_out" = 1,
  "fielders_choice" = 0,
  "fielders_choice_out" = 1,
  "force_out" = 1,
  "grounded_into_double_play" = 2,
  "hit_by_pitch" = 0,
  "home_run" = 0,
  "sac_bunt" = 1,
  "sac_bunt_double_play" = 2,
  "sac_fly" = 1,
  "sac_fly_double_play" = 2,
  "single" = 0,
  "strikeout" = 1,
  "strikeout_double_play" = 2,
  "triple" = 0,
  "triple_play" = 3,
  "walk" = 0,
  "catcher_interf"= 0,
  "field_error"= 0

)
```

1.3 Step 3: Create the function to perform a simulation of one entire game

[26]:

```
game_simulator <- function (player_ids, num_games=1, num_innings_per_game=9){
  if (length(player_ids) != 9){
    print('need 9 players')
  }
  else{
   stats <- tibble(
      player_id = player_ids,
      hits = rep(0, 9),
      at_bats = rep(0, 9),
      walks = rep(0, 9),
      rbis = rep(0, 9),
      sf = rep(0,9),
      hrs = rep(0,9),
      doubles=rep(0,9),
      singles=rep(0,9),
      triples=rep(0,9)
    )
  line_score <- tibble(
    "1" = 0,
    "2" = 0,
    "3" =0,
    "4" = 0,
    "5" = 0,
    "6" = 0,
     "7" =  0,
    "8" = 0,
    "9" =0,
    "R" = 0,
    "H" = 0,
    "E" = 0
  )
    current_batter <- 1
    inning_num <- 1
    current_bases <- '0-0-0'


    runs_scored_in_game <- 0
    total_hits <- 0
    while(inning_num <= num_innings_per_game){
      outs <- 0
            inning_runs <- 0
           while (outs < 3){
             event <-␣
  ↪get_random_event_for_batter_2023_2024(player_ids[current_batter],␣
  ↪current_bases, outs)
             if (is.null(event)) {
          # Simulate strikeout
          event <- list(
            event = "strikeout",
            runs_scored = 0,
            new_bases = current_bases,
            outs = 1
          )
        }
              runs_scored_in_game <- runs_scored_in_game + event$runs_scored
             inning_runs <- inning_runs + event$runs_scored
             current_bases <- event$new_bases
                    # Track RBIs (runs scored by teammates from this event)
```

```
             stats$rbis[current_batter] <- stats$rbis[current_batter] +␣
↪event$runs_scored
                      # Track hits and at-bats
               if (event$event %in% c("single", "double", "triple",␣
↪"home_run")) {
                 stats$hits[current_batter] <- stats$hits[current_batter] + 1
                 stats$at_bats[current_batter] <-␣
↪stats$at_bats[current_batter] + 1
                 total_hits <- total_hits+1
                 if(event$event == "home_run"){
                   stats$hrs[current_batter] <- stats$hrs[current_batter] + 1
                 }
                  if(event$event == "double"){
                   stats$doubles[current_batter] <-␣
↪stats$doubles[current_batter] + 1
                 }
                  if(event$event == "triple"){
                   stats$triples[current_batter] <-␣
↪stats$triples[current_batter] + 1
                 }
                  if(event$event == "single"){
                   stats$singles[current_batter] <-␣
↪stats$singles[current_batter] + 1
                 }
               } else if (event$event %in% c("walk", "hit_by_pitch")) {
                 stats$walks[current_batter] <- stats$walks[current_batter] + 1
               } else if (event$event %in% c("sac_bunt",␣
↪"sac_bunt_double_play","sac_fly", "sac_fly_double_play",␣
↪"catcher_interference")){

                 if(event$event == "sac_fly_double_play" || event$event ==␣
↪"sac_fly_double_play" ){
                     stats$sf[current_batter] <- stats$sf[current_batter] +1
                 }

               }
                          else {
                 stats$at_bats[current_batter] <-␣
↪stats$at_bats[current_batter] + 1
               }
         outs <- outs+event$outs
          current_batter <- current_batter+1
         if(current_batter > 9){
           current_batter <- 1
         }
```

```
        }
       line_score[as.character(inning_num)] <- inning_runs
      inning_num <- inning_num+1


    }
    line_score['H'] <- total_hits
    line_score['R'] <- runs_scored_in_game
      box_score <- stats %>%
      mutate(player_name = sapply(as.character(player_id), function(id)␣
 ↪player_map[[id]])) %>%
        mutate(BA = hits/at_bats,
               OBP = (walks+hits)/ (at_bats+walks+ sf) ,
                SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats ))␣
 ↪%>%
              mutate(OPS=OBP+SLG) %>%
      select(player_name, hits, at_bats, walks, rbis, hrs, doubles,␣
 ↪triples,singles, BA, OBP, SLG,OPS)
    return(list(total_runs = line_score['R'], box_score = box_score))




  }
}
```

1.4 Now we can test this simulation with a question: If Aaron Judge and Shohei Ohtani swapped places with eachother in their respective batting lineups, who will perform better for their team in terms of both OPS (OBP + SLG%) and RBIs

[27]:

```
# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162

# Initialize cumulative box score dataframe
cumulative_stats <- tibble(
  player_name = rep('Aaron Judge', 9),  # Assuming same player IDs as before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
```

```r
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(592450, 592450, 592450, 592450, 592450,␣
 ↪592450, 592450, 592450, 592450))

  if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG) %>%
  select(player_name,hits,at_bats,walks,rbis,hrs,BA,OBP,SLG,OPS)


print(paste0('Avg Runs per game: ', total_runs / num_games))
print(cumulative_stats)
```

```
[1] "Avg Runs per game: 11.9691358024691"
# A tibble: 9 × 10
  player_name  hits at_bats walks  rbis   hrs    BA   OBP   SLG   OPS
  <chr>       <dbl>
<dbl> <dbl> <dbl>
```

```
<dbl> <dbl> <dbl>
<dbl> <dbl>
1 Aaron Judge   247     735    196    240     87 0.336 0.476 0.751  1.23
2 Aaron Judge   232     751    170    233     83 0.309 0.436 0.700  1.14
3 Aaron Judge   204     730    173    192     67 0.279 0.417 0.603  1.02
4 Aaron Judge   204     699    175    239     72 0.292 0.434 0.668  1.10
5 Aaron Judge   183     676    180    160     56 0.271 0.424 0.581  1.01
6 Aaron Judge   202     660    180    205     62 0.306 0.455 0.655  1.11
7 Aaron Judge   214     657    153    246     80 0.326 0.453 0.778  1.23
8 Aaron Judge   204     609    190    207     73 0.335 0.493 0.764  1.26
9 Aaron Judge   191     627    162    217     64 0.305 0.447 0.678  1.13
```

1.4.1 This here shows the Avg run scored for a lineup of 9 Aaron Judge's and their season stats of 162 games played

[28]:

```
# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162


# Initialize cumulative box score dataframe
cumulative_stats <- tibble(
    player_name = c('G.Torres', 'J.Soto', 'A.Judge', 'A.Wells', 'G.Stanton', 'J.
 ↪Chisholm Jr.', 'A.Volpe',
                   'A.Rizzo', 'A.Verdugo'),   # Assuming same player IDs as␣
 ↪before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(650402, 665742, 592450, 669224, 519317,␣
 ↪665862, 683011, 519203, 657077))
   if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)  %>%
  select(player_name,hits,at_bats,walks,rbis,hrs,BA,OBP,SLG,OPS)


print(paste0('Avg Runs per game: ', total_runs / num_games))
print(cumulative_stats)
```

```
[1] "Avg Runs per game: 5.51234567901235"
# A tibble: 9 × 10
  player_name      hits at_bats walks  rbis   hrs    BA   OBP   SLG   OPS
  <chr>           <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl>
1 G.Torres          190     681   100    87    20 0.279 0.371 0.421
0.793
2 J.Soto            169     633   135   107    33 0.267 0.396 0.490
0.886
3 A.Judge           188     609   139   184    59 0.309 0.437 0.673
1.11
4 A.Wells           144     650    82   134    30 0.222 0.309 0.422
0.730
5 G.Stanton         154     641    77   118    40 0.240 0.322 0.479
0.801
6 J.Chisholm Jr.    173     663    41    83    39 0.261 0.304 0.489
0.793
```

```
7 A.Volpe          148     621    55     59     15 0.238 0.300 0.362
0.663
8 A.Rizzo          145     607    58     65     12 0.239 0.305 0.339
0.645
9 A.Verdugo        150     585    49     56      6 0.256 0.314 0.362
0.676
```

1.4.2 This here shows the Avg run scored for a lineup of the current New York Yankees and their simulated season stats.

[29]:

```
# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162

#
# Initialize cumulative box score dataframe
cumulative_stats <- tibble(
    player_name = rep("S.Ohtani", 9),   # Assuming same player IDs as before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(660271, 660271, 660271, 660271, 660271,␣
 ↪660271, 660271, 660271, 660271))

 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG) %>%  ␣
 ↪select(player_name,hits,at_bats,walks,rbis,hrs,BA,OBP,SLG,OPS)


print(paste0('Avg Runs per game: ', total_runs / num_games))
print(cumulative_stats)
```

```
[1] "Avg Runs per game: 8.98148148148148"
# A tibble: 9 × 10
  player_name  hits at_bats walks  rbis   hrs    BA   OBP   SLG   OPS
  <chr>       <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl>
1 S.Ohtani      239     756    96   145    58 0.316 0.393 0.623 1.02
2 S.Ohtani      213     726   110   183    70 0.293 0.386 0.663 1.05
3 S.Ohtani      231     729    85   157    60 0.317 0.388 0.660 1.05
4 S.Ohtani      215     693    95   168    61 0.310 0.393 0.648 1.04
5 S.Ohtani      212     679    99   149    62 0.312 0.400 0.652 1.05
6 S.Ohtani      203     669    88   166    54 0.303 0.384 0.614 0.999
7 S.Ohtani      197     662    88   159    51 0.298 0.38  0.595 0.975
8 S.Ohtani      193     640    74   166    60 0.302 0.374 0.672 1.05
9 S.Ohtani      203     619    83   162    50 0.328 0.407 0.666 1.07
```

1.4.3 This here shows the Avg run scored for a lineup of 9 Shohei Ohtani's and their season stats of 162 games played

[30]:

```
# lets get interesting, lets compare if swapping out Ohtani and Judge in the␣
 ↪same batting lineup who performs better and we will use a t test

# Run simulation for multiple games and accumulate box scores
total_runs <- 0
```

```r
num_games <- 162
num_seasons <- 15
ohtani_lad_ops_list <- list()
ohtani_lad_rbi_list <- list()

#
# Initialize cumulative box score dataframe


  cumulative_stats <- tibble(
    player_name = c("S.Ohtani", "M.Betts", "F.Freeman", "T.Hernandez", "W.
 ↪Smith",
                    "T.Edman", "M.Muncy", "K.Hernandez", "A.Pages"),   #␣
 ↪Assuming same player IDs as before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(660271, 605141, 518692, 606192, 669257,␣
 ↪669242, 571970, 571771, 681624))

 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats + walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)   %>% select(player_name,hits,at_bats,walks,rbis,hrs,BA,OBP,SLG,OPS)


print(paste0('Avg Runs per game: ', total_runs / num_games))


#print(ohtani_lad_ops_list)
print(cumulative_stats)
```

```
[1] "Avg Runs per game: 5.46913580246914"
# A tibble: 9 × 10
  player_name  hits at_bats walks  rbis   hrs    BA   OBP   SLG   OPS
  <chr>       <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl> <dbl>
<dbl> <dbl>
1 S.Ohtani      217     680    95   124    59 0.319 0.403 0.657 1.06
2 M.Betts       217     657   107   161    41 0.330 0.424 0.594 1.02
3 F.Freeman     192     664    85   112    28 0.289 0.370 0.485 0.855
4 T.Hernandez   186     680    58   116    31 0.274 0.331 0.469 0.800
5 W.Smith       137     624    80    80    26 0.220 0.308 0.391 0.699
6 T.Edman       154     637    46    96    21 0.242 0.293 0.416 0.709
7 M.Muncy       117     561   120    80    29 0.209 0.348 0.412 0.760
8 K.Hernandez   148     623    40    64    14 0.238 0.284 0.352 0.635
9 A.Pages       124     587    50    53    14 0.211 0.273 0.337 0.610
```

1.5 This here shows a simulation of 162 games for the 2024 Dodgers Lineup as well

1.6 Now to Perform the simulations, for this experiment I will simulate 15 “seasons” each for Ohtani and Aaron Judge for this year and take not of their OPS measures and number of RBIs. I will at the end have 8 lists (Ohtani OPS and RBIs for LADs,Judge OPS and RBIs for NYY, Ohtani OPS and RBIs for NYY after swapping, and Judge OPS and RBI for LADs after swap)

[31]:

```r
# lets get interesting, lets compare if swapping out Ohtani and Judge in the same batting lineup who performs better and we will use a t test

# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162
num_seasons <- 15
ohtani_lad_ops_list <- list()
ohtani_lad_rbi_list <- list()

#
# Initialize cumulative box score dataframe

for(season in 1:num_seasons){

  cumulative_stats <- tibble(
    player_name = c("S.Ohtani", "M.Betts", "F.Freeman", "T.Hernandez", "W.Smith",
                    "T.Edman", "M.Muncy", "K.Hernandez", "A.Pages"),   # Assuming same player IDs as before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(660271, 605141, 518692, 606192, 669257, 669242, 571970, 571771, 681624))

 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)

#print(paste0('Avg Runs per game: ', total_runs / num_games))
  ohtani_lad_ops_list[[season]] <- cumulative_stats %>% filter(player_name ==␣
 ↪"S.Ohtani") %>% pull(OPS)
  ohtani_lad_rbi_list[[season]] <- cumulative_stats %>% filter(player_name ==␣
 ↪"S.Ohtani") %>% pull(rbis)


}
#print(ohtani_lad_ops_list)
#View(cumulative_stats)
```

[32]:

```
# lets get interesting, lets compare if swapping out Ohtani and Judge in the␣
 ↪same batting lineup who performs better and we will use a t test

# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162
num_seasons <- 15
```

```
judge_nyy_ops_list <- list()
judge_nyy_rbi_list <- list()

#
# Initialize cumulative box score dataframe

for(season in 1:num_seasons){


# Initialize cumulative box score dataframe
cumulative_stats <- tibble(
    player_name = c('G.Torres', 'J.Soto', 'A.Judge', 'A.Wells', 'G.Stanton', 'J.
 ↪Chisholm Jr.', 'A.Volpe',
                  'A.Rizzo', 'A.Verdugo'),   # Assuming same player IDs as␣
 ↪before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(650402, 665742, 592450, 669224, 519317,␣
 ↪665862, 683011, 519203, 657077))

 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,
      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)

#print(paste0('Avg Runs per game: ', total_runs / num_games))
  judge_nyy_ops_list[[season]] <- cumulative_stats %>% filter(player_name == "A.
 ↪Judge") %>% pull(OPS)
  judge_nyy_rbi_list[[season]] <- cumulative_stats %>% filter(player_name == "A.
 ↪Judge") %>% pull(rbis)



}
#print(judge_nyy_ops_list)
#View(cumulative_stats)
```

[33]:

```
# swap teams/roles
# lets get interesting, lets compare if swapping out Ohtani and Judge in the␣
 ↪same batting lineup who performs better and we will use a t test

# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162
num_seasons <- 15
ohtani_nyy_ops_list <- list()
ohtani_nyy_rbi_list <- list()

#
# Initialize cumulative box score dataframe

for(season in 1:num_seasons){


# Initialize cumulative box score dataframe
cumulative_stats <- tibble(
    player_name = c('G.Torres', 'J.Soto', 'S.Ohtani', 'A.Wells', 'G.Stanton',␣
 ↪'J.Chisholm Jr.', 'A.Volpe',
                  'A.Rizzo', 'A.Verdugo'),   # Assuming same player IDs as␣
 ↪before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(650402, 665742, 660271, 669224, 519317,␣
 ↪665862, 683011, 519203, 657077))

 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)
```

```
#print(paste0('Avg Runs per game: ', total_runs / num_games))
  ohtani_nyy_ops_list[[season]] <- cumulative_stats %>% filter(player_name ==␣
 ↪"S.Ohtani") %>% pull(OPS)
  ohtani_nyy_rbi_list[[season]] <- cumulative_stats %>% filter(player_name ==␣
 ↪"S.Ohtani") %>% pull(rbis)



}
#print(ohtani_nyy_ops_list)
#View(cumulative_stats)
```

[34]:

```
# lets get interesting, lets compare if swapping out Ohtani and Judge in the␣
 ↪same batting lineup who performs better and we will use a t test

# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 162
num_seasons <- 15
judge_lad_ops_list <- list()
judge_lad_rbi_list <- list()

#
# Initialize cumulative box score dataframe

for(season in 1:num_seasons){

  cumulative_stats <- tibble(
    player_name = c("A.Judge", "M.Betts", "F.Freeman", "T.Hernandez", "W.Smith",
                    "T.Edman", "M.Muncy", "K.Hernandez", "A.Pages"),    #␣
 ↪Assuming same player IDs as before
  hits = rep(0, 9),
  at_bats = rep(0, 9),
  walks = rep(0, 9),
  rbis = rep(0, 9),
  hrs = rep(0,9),
  triples = rep(0,9),
  doubles = rep(0,9),
  singles = rep(0,9),
  BA = rep(0, 9),
  OBP = rep(0, 9)
)

for (i in 1:num_games) {
  game_result <- game_simulator(c(592450, 605141, 518692, 606192, 669257,␣
 ↪669242, 571970, 571771, 681624))
 if (!is.null(game_result$box_score)) {
    total_runs <- total_runs + game_result$total_runs

    # Add game stats to cumulative stats
    cumulative_stats <- cumulative_stats %>%
      mutate(
        hits = hits + game_result$box_score$hits,
        at_bats = at_bats + game_result$box_score$at_bats,
        walks = walks + game_result$box_score$walks,
        rbis = rbis + game_result$box_score$rbis,
        hrs = hrs + game_result$box_score$hrs,
        doubles = doubles + game_result$box_score$doubles,
        triples = triples + game_result$box_score$triples,
        singles = singles + game_result$box_score$singles,


      )
  } else {
    print(paste("Game", i, "failed to generate a valid box score."))
  }
}

# Recalculate BA and OBP for cumulative stats
cumulative_stats <- cumulative_stats %>%
  mutate(BA = ifelse(at_bats > 0, hits / at_bats, 0),
         OBP = ifelse((at_bats + walks) > 0, (walks + hits) / (at_bats +␣
 ↪walks), 0),
          SLG= ( (1*singles + 2*doubles + 3* triples + 4*hrs)/ at_bats )) %>%
              mutate(OPS=OBP+SLG)

#print(paste0('Avg Runs per game: ', total_runs / num_games))
  judge_lad_ops_list[[season]] <- cumulative_stats %>% filter(player_name == "A.
 ↪Judge") %>% pull(OPS)
  judge_lad_rbi_list[[season]]  <- cumulative_stats %>% filter(player_name ==␣
 ↪"A.Judge") %>% pull(rbis)


}
#print(judge_lad_ops_list)
#View(cumulative_stats)
```

1.7 Now time to perform the One-Way Anova to Assess if there is any significant difference between means of the 4 Groups

[35]:

```r
ohtani_ops_lad <- unlist(ohtani_lad_ops_list)
judge_ops_nyy <- unlist(judge_nyy_ops_list)
ohtani_nyy_ops_list <- unlist(ohtani_nyy_ops_list)
judge_lad_ops_list <- unlist(judge_lad_ops_list)


ops_data <- data.frame(
  ops = c(ohtani_ops_lad, judge_ops_nyy, ohtani_nyy_ops_list,␣
 ↪judge_lad_ops_list),
  player = c(rep("Ohtani_LAD", length(ohtani_ops_lad)), rep("Judge_NYY",␣
 ↪length(judge_ops_nyy)), rep("Ohtani_NYY", length(ohtani_nyy_ops_list)),␣
 ↪rep("Judge_LAD", length(judge_lad_ops_list)))


)
anova_result <- aov(ops ~ player, data = ops_data)

# Summary of ANOVA results
summary(anova_result)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)    
player       3 0.2095 0.06983   15.35 2.09e-07 ***
Residuals   56 0.2548 0.00455                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1.8 After performing the One-Way Anova we have a significant P-Value so there is some significant difference in mean between the 4 Groups of OPS

[36]:

```r
tukey_result <- TukeyHSD(anova_result)

# Print the results
print(tukey_result)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = ops ~ player, data = ops_data)

$player
                             diff          lwr          upr     p adj
Judge_NYY-Judge_LAD    0.02132060 -0.043893515  0.086534722 0.8224477
Ohtani_LAD-Judge_LAD  -0.13045359 -0.195667713 -0.065239476 0.0000120
Ohtani_NYY-Judge_LAD  -0.06123535 -0.126449468  0.003978769 0.0730942
Ohtani_LAD-Judge_NYY  -0.15177420 -0.216988316 -0.086560079 0.0000005
```

```
Ohtani_NYY-Judge_NYY  -0.08255595 -0.147770071 -0.017341834 0.0076558
Ohtani_NYY-Ohtani_LAD  0.06921825  0.004004127  0.134432364 0.0334275
```

From looking at these values from the turkey test some interesting values to look at are the Ohtani_LAD-Judge_LAD value of -.13 and the Ohtani_NYY-Judge_NYY value of -.08 which point to Aaron Judge outperforming Ohtani in the simulations

[37]:
```
ohtani_rbi_lad <- unlist(ohtani_lad_rbi_list)
judge_rbi_nyy <- unlist(judge_nyy_rbi_list)
ohtani_nyy_rbi_list <- unlist(ohtani_nyy_rbi_list)
judge_lad_rbi_list <- unlist(judge_lad_rbi_list)


rbi_data <- data.frame(
  ops = c(ohtani_rbi_lad, judge_rbi_nyy, ohtani_nyy_rbi_list,␣
 ↪judge_lad_rbi_list),
  player = c(rep("Ohtani_LAD", length(ohtani_rbi_lad)), rep("Judge_NYY",␣
 ↪length(judge_rbi_nyy)), rep("Ohtani_NYY", length(ohtani_nyy_rbi_list)),␣
 ↪rep("Judge_LAD", length(judge_lad_rbi_list)))


)

anova_result <- aov(ops ~ player, data = rbi_data)

# Summary of ANOVA results
summary(anova_result)
```

```
            Df Sum Sq Mean Sq F value  Pr(>F)
player       3  31349   10450   39.29 8.4e-14 ***
Residuals   56  14895     266
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[38]:
```
#print(t_test_result)
tukey_result <- TukeyHSD(anova_result)

# Print the results
print(tukey_result)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = ops ~ player, data = rbi_data)

$player
                           diff      lwr       upr     p adj
Judge_NYY-Judge_LAD    47.20000  31.43143  62.968566 0.0000000
```

```
Ohtani_LAD-Judge_LAD   -9.80000 -25.56857   5.968566 0.3619711
Ohtani_NYY-Judge_LAD   30.06667  14.29810  45.835233 0.0000293
Ohtani_LAD-Judge_NYY  -57.00000 -72.76857 -41.231434 0.0000000
Ohtani_NYY-Judge_NYY  -17.13333 -32.90190  -1.364767 0.0281854
Ohtani_NYY-Ohtani_LAD  39.86667  24.09810  55.635233 0.0000001
```

From looking at the same measures here swapping Ohtani for Judge would net the Yankees on average 17 less RBIs from that spot in the batting lineup per year with an average increase in 9.8 RBIs per year for the Dodgers in the reverse

2 SUMMARY

2.0.1 Both Ohtani and Judge are great hitters and this was a fun way to use one-way anova to test out a swap, where with baserunning factored in both teams would be fine with either or player