Mlb_At_Bat_Simulator

October 22, 2024

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

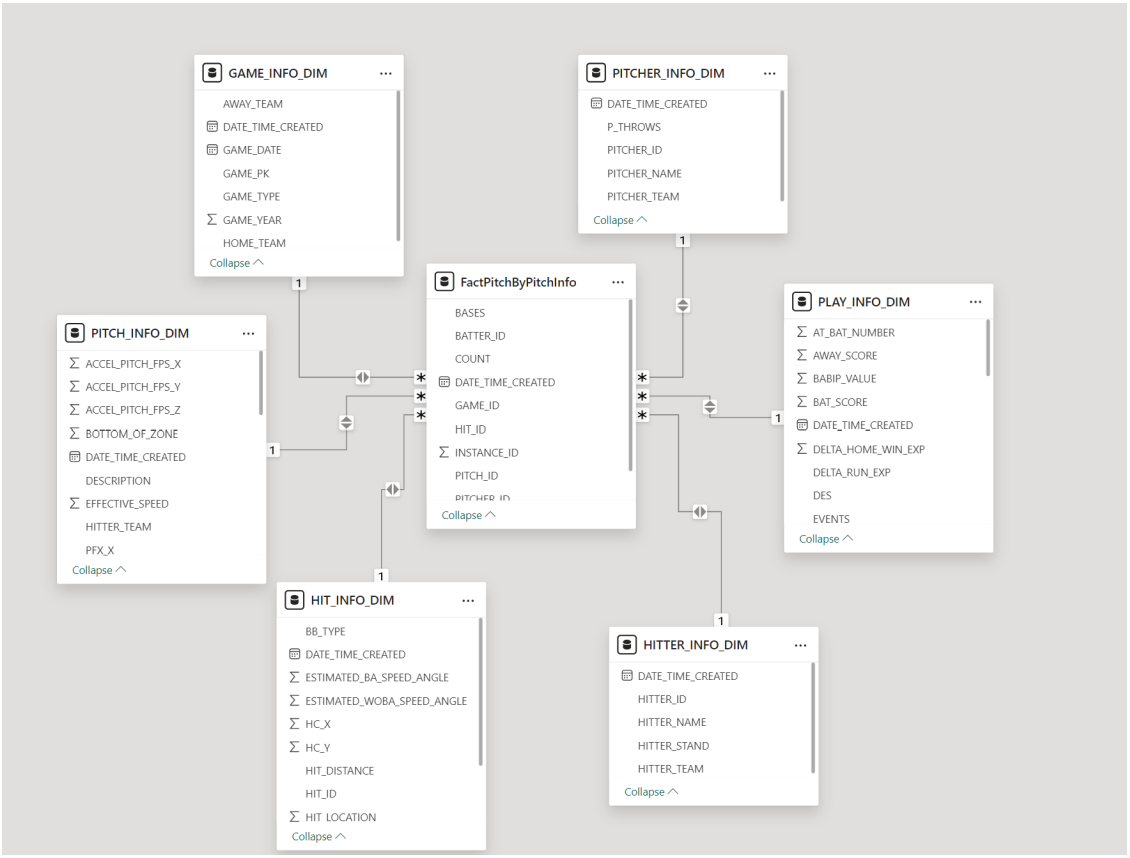
1.0.1

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 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 :))

[2]:
```
# 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

[10]:
```
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 = 2024")
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

[11]:
```
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.

[12]:
```
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)
}
```

[13]:
```
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

[14]:
```
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 field out on event not found for batter in scenario (chose␣
↪field out since it is most common event)
        event <- list(
          event = "field_out",
          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

[15]:
```
# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 20

# 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),
  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: 11.8"
# 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    35      98    12    35    16 0.357 0.427 0.898 1.33
2 Aaron Judge    27      87    23    22     9 0.310 0.455 0.701 1.16
3 Aaron Judge    21      88    22    16     3 0.239 0.391 0.386 0.777
4 Aaron Judge    24      81    23    33    10 0.296 0.452 0.716 1.17
5 Aaron Judge    30      86    17    38     9 0.349 0.456 0.744 1.20
6 Aaron Judge    24      79    24    20     7 0.304 0.466 0.646 1.11
7 Aaron Judge    21      79    22    18     6 0.266 0.426 0.544 0.970
8 Aaron Judge    24      81    16    29     8 0.296 0.412 0.667 1.08
9 Aaron Judge    24      79    15    25     7 0.304 0.415 0.620 1.04
```

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

[16]:
```r
# 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: 5.03703703703704"
# 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         175     681    84    59    15 0.257 0.339 0.372
0.710
2 J.Soto           172     617   134   130    52 0.279 0.407 0.588
0.996
3 A.Judge          203     594   147   170    64 0.342 0.472 0.736
1.21
4 A.Wells          140     656    73   112    27 0.213 0.292 0.387
0.679
5 G.Stanton        140     647    63   114    43 0.216 0.286 0.459
0.745
6 J.Chisholm Jr.   130     633    56    66    24 0.205 0.270 0.359
```

```
0.629
7 A.Volpe          143     615    54    54    14 0.233 0.294 0.363
0.657
8 A.Rizzo          136     585    72    59    14 0.232 0.317 0.354
0.670
9 A.Verdugo        130     572    47    52    12 0.227 0.286 0.334
0.620
```

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

[17]:
```r
# Run simulation for multiple games and accumulate box scores
total_runs <- 0
num_games <- 20

#
# 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: 10.5"
# 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       38      98    11    29    14 0.388 0.450 0.959 1.41
2 S.Ohtani       32      98     9    22     7 0.327 0.383 0.643 1.03
3 S.Ohtani       29      87    17    27     8 0.333 0.442 0.678 1.12
4 S.Ohtani       30      88    16    32    10 0.341 0.442 0.795 1.24
5 S.Ohtani       31      90    11    30    10 0.344 0.416 0.7   1.12
6 S.Ohtani       27      89     7    23     8 0.303 0.354 0.629 0.983
7 S.Ohtani       21      79    13    17     4 0.266 0.370 0.456 0.825
8 S.Ohtani       24      81    11    15     4 0.296 0.380 0.481 0.862
9 S.Ohtani       23      81    10    15     7 0.284 0.363 0.654 1.02
```

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

[18]:
```
# 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
```

```r
total_runs <- 0
num_games <- 162

#
# 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: 5.45679012345679"
# 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      197     694    93   113    55 0.284 0.368 0.592 0.961
2 M.Betts       187     673    89   131    28 0.278 0.362 0.461 0.823
3 F.Freeman     174     673    77    91    24 0.259 0.335 0.431 0.766
4 T.Hernandez   199     678    60   121    40 0.294 0.351 0.522 0.873
5 W.Smith       160     661    57    83    22 0.242 0.302 0.401 0.703
6 T.Edman       196     651    32   126    27 0.301 0.334 0.501 0.835
7 M.Muncy       123     550   125    72    31 0.224 0.367 0.464 0.831
8 K.Hernandez   144     601    51    68    20 0.240 0.299 0.369 0.668
9 A.Pages       154     597    40    79    23 0.258 0.305 0.430 0.735
```

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)

[19]:
```
# 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)
```

[20]:
```
# 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
```

```r
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)
```

[21]:
```
# 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)
```

[22]:
```
# 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

[23]:
```
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 0.2011 0.06704    18.2 2.28e-08 ***
Residuals    56 0.2062 0.00368
---
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

[24]:
```
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     0.108900390  5.022478e-02  0.167576001 0.0000473
Ohtani_LAD-Judge_LAD   -0.050771144 -1.094468e-01  0.007904466 0.1123501
Ohtani_NYY-Judge_LAD    0.007949233 -5.072638e-02  0.066624844 0.9840259
Ohtani_LAD-Judge_NYY   -0.159671535 -2.183471e-01 -0.100995924 0.0000000
```

```
Ohtani_NYY-Judge_NYY  -0.100951157 -1.596268e-01 -0.042275546 0.0001650
Ohtani_NYY-Ohtani_LAD  0.058720378  4.476717e-05  0.117395988 0.0497566
```

From looking at these values from the turkey test some interesting values to look at are the Ohtani_LAD-Judge_LAD value of -.05 and the Ohtani_NYY-Judge_NYY value of -.10 which point to Aaron Judge outperforming Ohtani in the simulations, so overall there isn't much dropoff in OPS in the swap

[25]:
```
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  53294   17765   80.47 <2e-16 ***
Residuals   56  12362     221
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

[26]:
```
#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    70.266667  55.901062  84.632271 0.0000000
Ohtani_LAD-Judge_LAD    5.466667  -8.898938  19.832271 0.7456296
Ohtani_NYY-Judge_LAD   51.000000  36.634395  65.365605 0.0000000
Ohtani_LAD-Judge_NYY  -64.800000 -79.165605 -50.434395 0.0000000
Ohtani_NYY-Judge_NYY  -19.266667 -33.632271  -4.901062 0.0042538
Ohtani_NYY-Ohtani_LAD  45.533333  31.167729  59.898938 0.0000000
```

From looking at the same measures here swapping Ohtani for Judge would net the Yankees on average 19 less RBIs from that spot in the batting lineup in a season with the Dodgers also losing on average 5 RBIs in a season for the swap

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. While also considering the different roles they play for their respective teams in Ohtani leading off and Judge hitting 3rd before cleanup, I belive that they both suit their roles greatly for their teams, which also contributes to the higher number of difference in RBIs