Inflated Statistics in Test Cricket

MTH208A Group 8 Project

Parmar Soham Muljibhai (210703) Samarth Kumar (210911)
Krati Jain (241080075) Shreyasi Mondal (241080097)

2024-11-09

Aim

In any sport, we often hear the word 'average' which is meant to show a a player's greatness, consistency and skillset in that particular sport. So in cricket, analysts very often look at batting and bowling averages, defined in later sections of this report, to justify how good a cricketer is. As emerging data scientists (and physicists in the group), we try to demystify the fact that a good average implies a good player. Here we will show how averages can be inflated to such an amount that a player (definitely not defaming them) who is just decent, finds their name in the debates of GOAT (Greatest of All Time) despite having lesser influence on the game.

Data

The datasets contain the statistics of top 10 test batters, bowlers and all-rounders opposition wise and the host countries wise, scraped from ESPN ("ESPN Cricinfo Website," n.d.) until November 13th 2024. The reader is advised to keep in mind a few terms. (i) Opposition-wise statistics have all the statistics of a certain player when they have played against a certain country and (ii) host-wise is when a player has played the game in a certain country. These two terms (opposition-wise and host-wise) are used throughout this report.

Variables

For each type of cricketers we have two datasets, one for host wise data and one for opposition wise data. For batters, we have the data for their number of innings batted, number of not-outs, and average against each team along with their overall career batting average. For bowlers, we have the runs conceded by them, number of wickets taken, and average against each team

along with their overall career bowling average. Similarly for all-rounders, we have both their bowling data and batting data.

Apart from these numerical data, we have also stored images of all the 30 players (10 batters, 10 bowlers and 10 all rounders).

Here's an example:

This is the structure of the RData file of batting stats of all-rounders.

i. 'Name' contains the name of top 10 all-rounders.
ii. 'Against' is a list of their innings batted, no. of not-outs and average against each team that they have batted against. It'll show NA for the teams against whom they have not played.
iii. 'Host' is a list of their innings batted, no. of not-outs and average in each country that they have batted in. It'll show NA if they haven't batted in that country.
iv. 'OA' is a list of their overall career batting average in Test cricket.
v. 'Picture' contains urls of pictures of all the top 10 all-rounders.

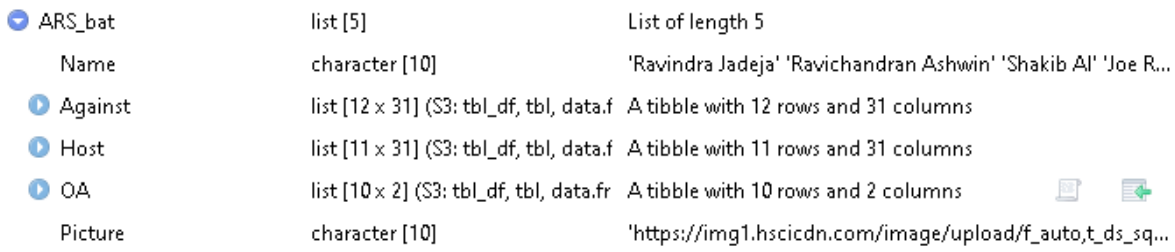

Figure 1: Raw data structure scrapped from the webpage (a list of tibble)

Obtaining the Data

The entire data was scrapped from the ESPN website ("ESPN Cricinfo Website," n.d.). All the statistics and pictures of every player were available there. The major difficulty faced was due to the dynamic nature of the website. Interactions in the website changed the webpage without updating the url, hence using direct scarping libraries in R very difficult.

To solve this issue, we used python to simulate a web browser using the Selenium library ("Web Testing Library for Robot Framework," n.d.). This allowed us to interact with the browser and have access to the dynamic source. We scraped the required data and then processed it into the required form in R.

Biases in the Data

Our main aim itself was to find out the biases in the data. In every team sport, it very often happens that a player gets highly rated for their good overall performances. But in reality, that player might have scored well only against poor opponents and in easy conditions. Whereas, some players might have played well against strong opposition and in difficult conditions.

We had impressions of such cricketers being in top 10 test rankings. So we have done a thorough analysis of their stats and tried to find if someone is overrated, i.e., that player has inflated statistics and because of that he is among the top ranked cricketers.

Talking about the actual bias, here's an example of host-wise stats of Prabath Jayasuriya (Sri Lankan player):

```
> bowl.host(7)
Stats of Prabath Jayasuriya host-team-wise
# A tibble: 11 × 4
   Country Prabath Jayasuriya R…¹ Prabath Jayasuriya W…² Prabath Jayasuriya A…³
   <chr>                    <dbl>                  <int>                  <dbl>
 1 Austra…                     NA                     NA                     NA
 2 Bangla…                    234                      4                   58.5
 3 England                    384                      8                   48
 4 India                       NA                     NA                     NA
 5 New Ze…                    224                      4                   56
 6 Pakist…                     NA                     NA                     NA
 7 South …                     NA                     NA                     NA
 8 Sri La…                   1915                     81                   23.6
 9 U.A.E.                      NA                     NA                     NA
10 West I…                     NA                     NA                     NA
11 Zimbab…                     NA                     NA                     NA
# i abbreviated names: ¹`Prabath Jayasuriya Runs`, ²`Prabath Jayasuriya Wkts`,
#   ³`Prabath Jayasuriya Avg`
Avg of Prabath Jayasuriya in Test matches hosted in England is 48
Avg of Prabath Jayasuriya in Test matches hosted in New Zealand is 56
Overall average of Prabath Jayasuriya in tests hosted by strong teams is 50.66667

Avg of Prabath Jayasuriya in Test matches hosted in Bangladesh is 58.5
Overall average of Prabath Jayasuriya in tests hosted by weak teams is 58.5
>
```

Figure 2: Host-wise Statistics of Prabath Jayasuriya

He has mainly played in his home-conditions only, which can be seen from the fact that out of his 97 test wickets, 81 were in Sri-Lanka.

Data Analysis

We had the career statistics of top-10 ranked Test players from the ESPN website and after scrapping it, we saved it to RData files. Then we did the required analysis for every player type (batters, bowlers and all-rounders) both opposition-wise and host-wise.

Generic Functions

This was done using generic functions for every player type. One such generic function is show below:

```
## allrounder batting opposition wise
# Please load Allrounders_bat.RData file by running the below commented code
# load("Allrounders_bat.RData")
# You can download this file from the Main sub-folder of Data Folder

allround.bat.against <- function(number){
  # rank is a vector containing column indexes that we need for analysis for nth ranked batte
  rank <- c(1, 3*number-1, 3*number, 3*number+1)

  # ARS_bat is the list of tibbles (available in Main sub-folder of Data folder)
  allround.bat.opposition <- ARS_bat[2]$Against[rank] # opposition teams

  cat("Stats of", ARS_bat$Name[number], "opposition-wise \n")
  print(allround.bat.opposition)

  strong.teams <- c("Australia", "India", "New Zealand", "South Africa", "England")
  weak.teams <- c("Ireland", "Bangladesh", "Afghanistan", "Zimbabwe")

  strong.runs <- 0
  weak.runs <- 0
  strong.inns <- 0
  weak.inns <- 0

  # Loop through opposition teams and calculate stats
  for(i in 1:nrow(allround.bat.opposition)){
    # Convert columns to numeric, safely handling NAs and non-numeric values
    # We have to do this because of Harry Brooks (a batter), who never got out against weak t
    avg_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 4]]))
    inns_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 2]]))
    no_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 3]]))
```

```
  if(allround.bat.opposition[[i, 1]] %in% strong.teams){
    # Only proceed if avg is not NA
    if(!is.na(avg_value)){
      cat("Avg of", ARS_bat$Name[number], "against", allround.bat.opposition[[i, 1]], "is"

      # If all values are numeric, calculate strong runs and innings
      if(!is.na(inns_value) & !is.na(no_value) & !is.na(avg_value)){
        strong.runs <- strong.runs + avg_value * (inns_value - no_value)
      }
      if(!is.na(inns_value) & !is.na(no_value)){
        strong.inns <- strong.inns + (inns_value - no_value)
      }
    }
  }
}

# Avoid division by zero and handle NA
if(!is.na(strong.inns) && strong.inns > 0){
  strong.avg <- strong.runs / strong.inns
  cat("Overall average of", ARS_bat$Name[number], "against strong teams is", strong.avg, "
} else {
  cat("Has not batted or insufficient data against strong teams\n")
  strong.avg = NA
}

cat("\n")

for(i in 1:nrow(allround.bat.opposition)){
  avg_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 4]]))
  inns_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 2]]))
  no_value <- suppressWarnings(as.numeric(allround.bat.opposition[[i, 3]]))

  if(allround.bat.opposition[[i, 1]] %in% weak.teams){
    if(!is.na(avg_value)){
      cat("Avg of", ARS_bat$Name[number], "against", allround.bat.opposition[[i, 1]], "is"

      # If all values are numeric, calculate weak runs and innings
      if(!is.na(inns_value) & !is.na(no_value) & !is.na(avg_value)){
        weak.runs <- weak.runs + avg_value * (inns_value - no_value)
      }
      if(!is.na(inns_value) & !is.na(no_value)){
        weak.inns <- weak.inns + (inns_value - no_value)
```

```
        }
      }
    }
  }

  # Avoid division by zero and handle NA
  if(!is.na(weak.inns) && weak.inns > 0){
    weak.avg <- weak.runs / weak.inns
    cat("Overall average of", ARS_bat$Name[number], "against weak teams is", weak.avg, "\n")
  } else {
    cat(ARS_bat$Name[number], "has not batted or insufficient data against weak teams\n")
    weak.avg = NA
  }
  return(c(strong.runs, weak.runs,strong.inns, weak.inns,strong.avg, weak.avg))
}
```

This function does the analysis of batting statistics of top 10 test all-rounders opposition wise, so we named it 'allround.bat.against'. Two variables of our main interest that this function calculates are 'strong.avg' and 'weak.avg'. strong.avg is the batting average of all-rounders against strong oppositions and weak.avg is the batting average of all-rounders against weak oppositions. Similar analysis of top 10 all-rounders' batting statistics host wise is also done. Same for their bowling statistics. This way, we have four generic functions for all-rounders: opposition-wise batting, host-wise batting, opposition-wise bowling, host-wise bowling. Similarly, we have two functions for batters and two functions for bowlers. This in total accounts for 8 generic functions.

Major hurdles in writing these generic functions were (i) NA values and (ii)'-'. (i) Say we have an Indian player, Virat Kohli. Then we will have NA everywhere wherever the opposition is India because an Indian player can't play against India. Also, incase of never having played against some opposition or in a particular country, we will have NA there. (ii) '-' appears incase of infinites or non-defined values. Suppose a batter has batted against a team very few times and that team never managed to get that batter out. This means that the batter will have an infinite average against that team since

$$batting\ average = \frac{runs\ scored}{(no.\ of\ innings\ batted)\ -\ (no.\ of\ Not\ Outs)}$$

Similarly, for a bowler, if he has bowled during a game but failed to grab a wicket, then that bowler will have an infinite average against that team since

$$bowling\ average = \frac{runs\ conceded}{wickets\ taken}$$

Classifying teams

As you must have noticed that we have defined strong and weak teams in the function above. This classification is based on the team performances in 2010-2019 decade, which again we found from ESPN website:

Result summary in 2010s in Tests

Result Summary

Team	Span	Mat	Won	Lost	Draw	Tied	Tie+W	Tie+L	NR	W/L	%W	%L	%D	%
Afghanistan	2018-2019	4	2	2	0	0	0	0	0	1	50.00	50.00	0.00	50.00
Australia	2010-2019	112	57	38	17	0	0	0	0	1.5	50.89	33.92	15.17	60.00
Bangladesh	2010-2019	56	10	36	10	0	0	0	0	0.277	17.85	64.28	17.85	21.73
England	2010-2019	126	57	46	23	0	0	0	0	1.239	45.23	36.50	18.25	55.33
India	2010-2019	107	56	29	22	0	0	0	0	1.931	52.33	27.10	20.56	65.88
Ireland	2018-2019	3	0	3	0	0	0	0	0	-	0.00	100.00	0.00	0.00
New Zealand	2010-2019	83	32	31	20	0	0	0	0	1.032	38.55	37.34	24.09	50.79
Pakistan	2010-2019	83	33	37	13	0	0	0	0	0.891	39.75	44.57	15.66	47.14
South Africa	2010-2019	90	45	25	20	0	0	0	0	1.8	50.00	27.77	22.22	64.28
Sri Lanka	2010-2019	95	31	40	24	0	0	0	0	0.775	32.63	42.10	25.26	43.66
West Indies	2010-2019	83	22	43	18	0	0	0	0	0.511	26.50	51.80	21.68	33.84
Zimbabwe	2011-2018	24	4	19	1	0	0	0	0	0.21	16.66	79.16	4.16	17.39

Figure 3: Team-wise test performances of the past decade as obtained from ESPN ("ESPN Cricinfo Website," n.d.)

Since we are considering only 'current' top 10 test ranked cricketers, we only considered team performances of the past decade (2010-2019), as most of these current top players have played

during the past decade, to decide upon which teams are strong and which teams are weak. We ranked the teams based on their non-loss %, i.e., the % of matches they managed to win and draw. That's why, Australia, India, England, South Africa and New Zealand are considered as strong teams whereas Ireland, Bangladesh, Zimbabwe are considered as weak teams. The reason for doing so is that a team can have suitable home conditions and easily win matches, like we see in case of Pakistan who have just slightly more win % than New Zealand but they have lost more matches than what they have won. And so their loss % is way higher than that of New Zealand. Note that Afghanistan was also included in the category of weak teams because they have never really played test matches against stronger oppositions (except India) so their non-loss % is 50%.

Thus, the classification part was manually done based on past decade performance and cricket knowledge of the makers of this app.

On a clarification note, the performance of teams in earlier decades doesn't matter here since we are analysing stats of current top 10 test ranked players. So, a team being strong in 80s and 90s decade has no effect on our analysis as players under our consideration have not played against strong teams of older times. That's why West Indies, Pakistan and Sri Lanka are neither considered among strong nor weak teams. Although, we do agree that we could have done the same analysis for 'moderately ranked teams' but we realised that late. This can also inflate statistics and non-strong teams should be given attention in such cases.

Results Drawn from the Data

The data that we have scrapped and analysed can answer the following questions:

i. Does a player only plays well against weaker opposition or against any kind of opposition?

ii. Are there players who only play good against strong opposition (surprising answer for non-viewers of the sport)?

iii. Are the overall averages really accurate?

Shiny Web App

Our web app is divided into three panes (with the help of "Bslib UI Toolkit," n.d.). All of which are explained below:

I. Player Statistics

The first pane of our app is focussed on displaying the selected player's statistics along with their image. This essentially show cases all our RData files and all the data that we scrapped.

II. Match Simulation

In our second pane, we silmulate a match between our fantasy teams. We get to choose the name of our team, select 5 batters and 5 bowlers. The user is free to make their own team by choosing 5 batters and bowlers from the top-10 test rankings and the rest 5 batters and bowlers will go to the opposition team. Since this add up to 10 players in each team and a playing team normally consists of 11 players, the other two players for each team will be selected from our group members. Say if Samarth and Shreyasi are selected, Samarth will play and Shreyasi will be the 12th player (sitting out, substitute fielder). The actual simulation is simply based on normal distribution for each player's score where mean is that player's overall career average and standard deviation was assumed to be 10 without loss of generality.

One may get confused we simulated for only two innings (or why did we choose to write two innings) when a test match is consists of four innings. It's just the nomenclature that we followed was match-based instead of team-based. So, when talking about the match, we only say two innings (1st innings of both teams, 2nd innings of both teams). Below is an example of scorecard of the test matches which are match-based:

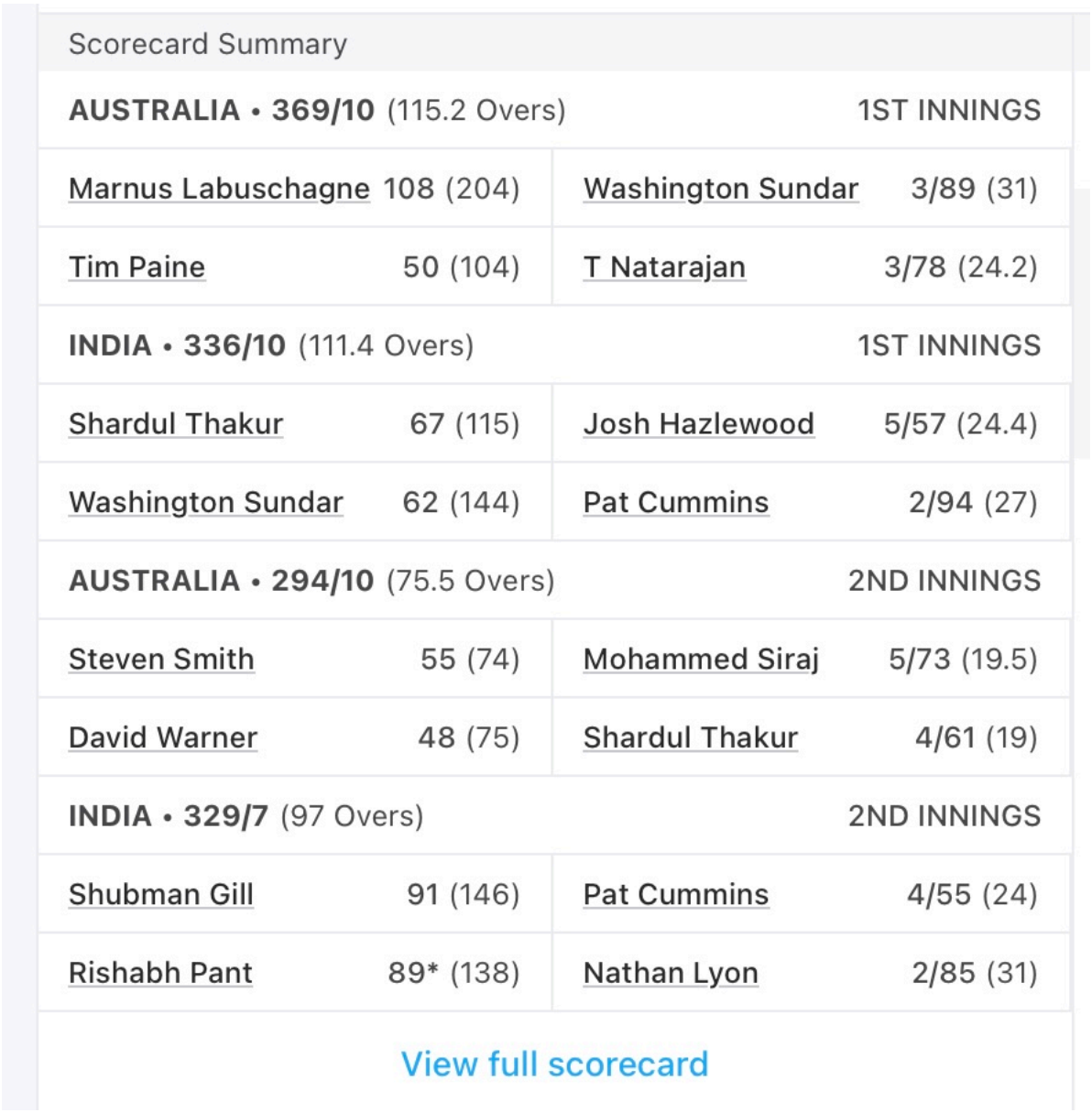

Figure 4: Scorecard of India's historic Gabba victory which ended the 36-years long unbeaten run of Australia at Brisbane

Clearly one can notice the innings distribution here as 1st innings of Australia, then India and 2nd innings of Australia, then India. Had it been a team-based simulation, then it would have been 1st and 3rd innings, or 2nd and 4th innings.

III. Analysis

In the the last pane we have plotted graphs using ggplot which are generated from the outputs of our generic function. We have stored those outputs into a data frame which are used here as input parameters to ggplot function. This serves as the visualization portion.

Important Visualizations

Here we show graphs of how well a player has performed against strong and weak teams, as well as in Test matches hosted by strong test playing nations and weak test playing nations. On the x-axis, we have name of the players based on the player type that once selects and on the y-axis, we have their avergaes. The closer the overall average dot is to the strong_avg than to the weak_avg, we can conclude that such players have more tendency to perform against stronger teams than against weaker teams. In all the graphs, strong_avg_Opp and weak_avg_Opp stands for averages of player against strong and weak oppositions, respectively. Whereas, strong_avg_Host and weak_avg_Host stands for averages of player in Test Matches hosted by strong and weak test playing nations, respectively. In case one only wants this comparison of overall average with opposition-wise averages or with host-wise averages, they can do so by selecting the respective tabs as shown in the images below.

Batters

Graph for top-10 test batters is shown below:

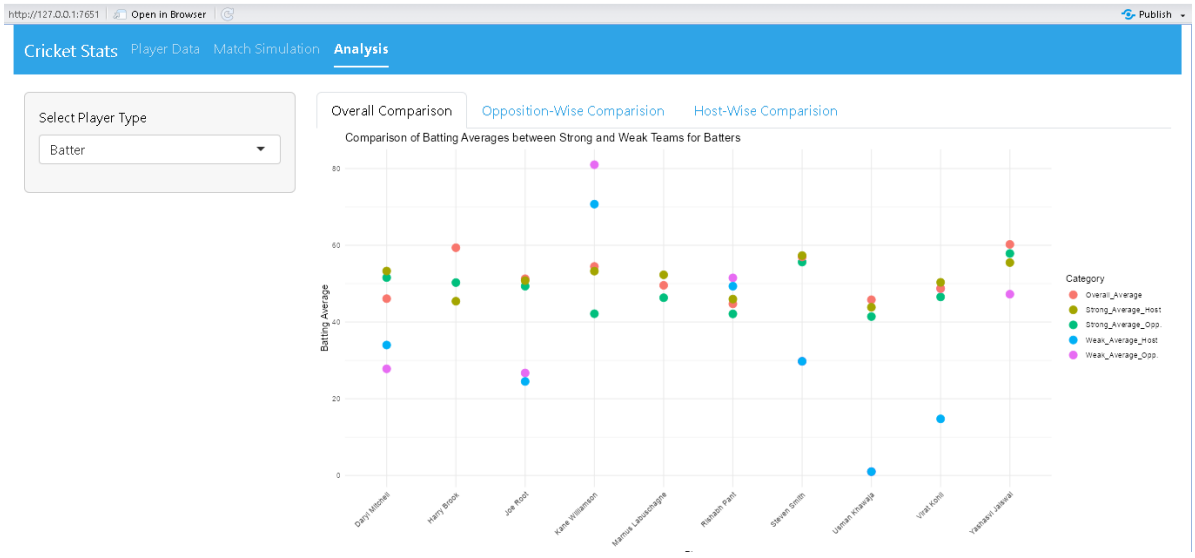

Figure 5: Analysis of batters' performance

As you can see that the overall average of Kane Williamson is around 55. So one expects him amongst the all time best batters in the Test format. But, if you look at his weak_avg_Host (batting average in test matches hosted by weaker teams) and weak_avg_Opp (batting average against weaker oppositions), they are around 70 and 80 respectively. This means that his overall stats are inflated. On the other hand, if you look at the stats of players like Steve Smith or Virat Kohli, their overall average is same as their strong averages (both opposition-wise and host-wise). This shows that their overall stats are not inflated as opposed to Kane Williamson.

A similar observation can be made for other player types as well.

Bowlers

Graph for top-10 test bowlers is shown below:

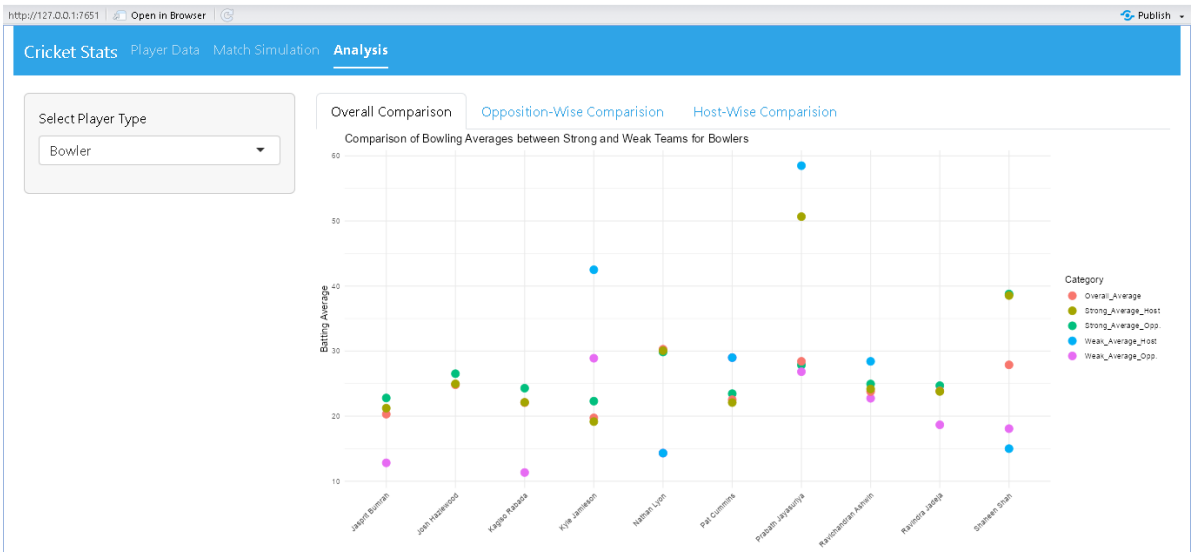

Figure 6: Analysis of bowlers' performance

All-rounders

Graph for batting stats of top-10 test all-rounders is shown below:

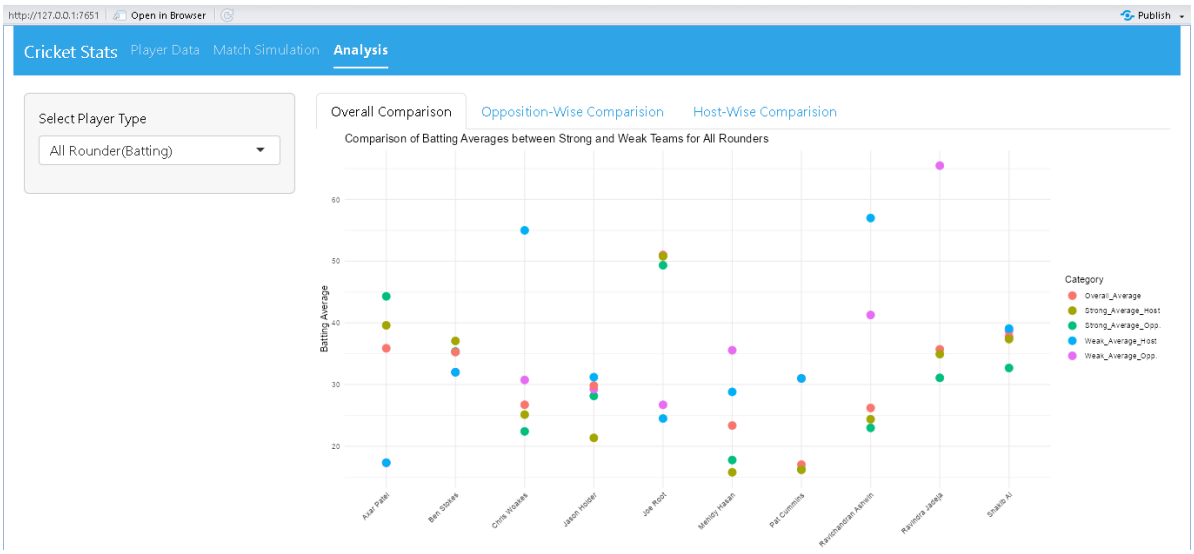

Figure 7: Analysis of allrounders' batting performance

Graph for bowling stats of top-10 test all-rounders is shown below:

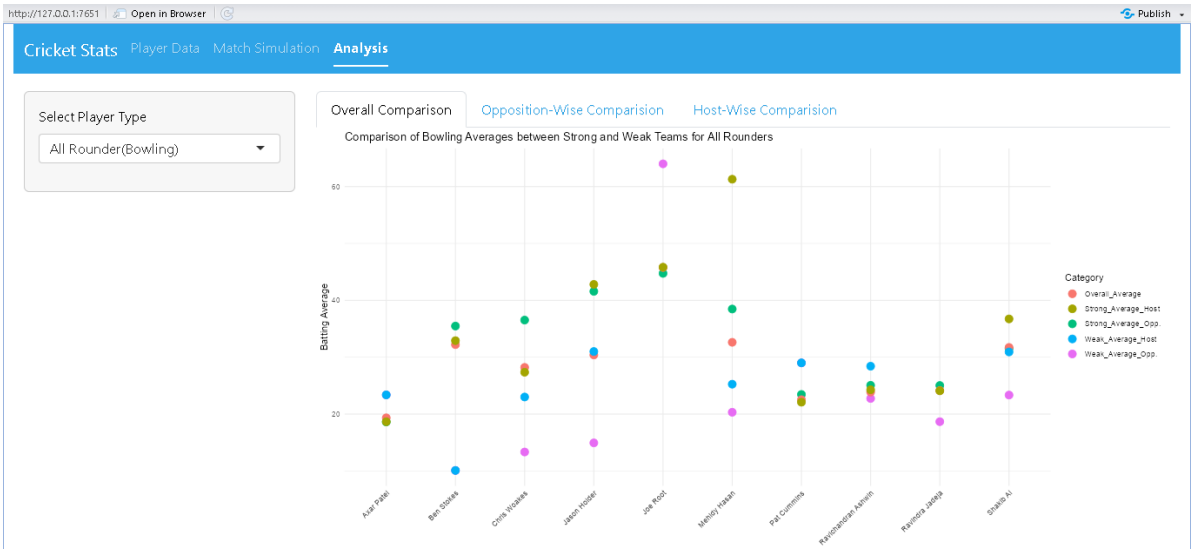

Figure 8: Analysis of allrounders' bowling performance

Conclusions

i. Cricketers certainly like playing against some specific opposition team

ii. Conditions matter a lot for cricketers' ability to perform at their best
iii. Rankings don't always justify the actual playing ability

Improvement prospects

Some more conditions that we could have explored: i. Peak performance of the player during which season (time period in years) ii. Performance of the player the player during ICC Tournaments (if done for limited overs cricket) iii. What affects players' performances? There once aired a very interesting stat comparison of some players during a live-match: Stats of players before and after marriage iv. Rough phases, and how well did the player perform on coming back to form.

Acknowledgements

We would like to thank Prof. Dootika Vats for sharing her valuable knowledge throughout this course and giving us a chance to showcase our web app. If not for the chance of presenting infront of an audience, it wouldn't have been this much fun. We are also grateful to some of the internet sources that helped us in making the app as well as this report ("Distill Format," n.d. ; "R Markdown Editor," n.d.).

"Bslib UI Toolkit." n.d. https://rstudio.github.io/bslib/.
"Distill Format." n.d. https://rstudio.github.io/distill/basics.html.
"ESPN Cricinfo Website." n.d. https://www.espncricinfo.com.
"R Markdown Editor." n.d. https://posit.co/blog/exploring-rstudio-visual-markdown-editor/.
"Web Testing Library for Robot Framework." n.d. https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html.