PROCESS BOOK
IPL 2022 VISUALIZATION

1. **Basic Information**

Project title: IPL 2022 Visualization
Names: Harsh Mahajan, Kunal Manjare, Dhruvil Shah
Email IDs: u1413898@utah.edu, u1419704@utah.edu, u1420007@utah.edu
uID: u1413898, u1419704, u1420007
Link to repository: https://github.com/kunal911/Data-Visualization-Final-Project-

2. **Overview and Motivation**

Cricket is the second most played sport with over 2.6 billion viewers across the world and the Indian Premier League (IPL) is considered to be the pinnacle of franchise cricket contributing to over 220 million viewers. Before we dive into our motivation for this project, we would like to talk a bit about cricket and IPL.

Cricket

There are primarily three types of international cricket matches-- test matches, one day internationals, and Twenty20. Among these, one day internationals (ODIs) are the most commonly played type of matches in international tournaments including the ICC Cricket World Cup and the ICC Champions Trophy. Matches are played between two teams each consisting of 11 players.

1. Each One Day International(ODI) match consists of two innings requiring both teams to bat and field once. Each inning is composed of 50 overs and each over consists of six deliveries bowled by the one player from the fielding team. The goal of the batting team is to score as many runs as possible and avoid their player's being dismissed by the fielding team while doing so. At the end of the match, the team with a higher score wins.
2. Twenty20 cricket, also called T20, truncated the form of cricket that revolutionized the game when it was introduced in 2003.The basic rules are the same as for the longer versions, but innings are limited to 20 overs a side, with a maximum of four overs for each bowler and restrictions on the placement of fielders designed to encourage big hitting by the batsmen and high scores.
3. A Test match is the longest form of cricket and the format in which the real essence of the game lies. It is a true test of the ability and temperament of a cricketer and the most intriguing of all formats. A test match would go on for a duration of five days with two innings per side. A standard day in test cricket consists of three sessions with 30 overs in each session. The breaks between sessions are 40 minutes for lunch and 20 minutes for tea.

IPL

The 2008-founded Indian Premier League (IPL) is a professional Twenty20 (T20) cricket league in India. Major Indian cities have teams in the league, which uses a round-robin group and knockout system. The IPL, which was created by the Board of Control for Cricket in India (BCCI), has grown to be the most lucrative and well-liked cricket league in the world. Matches typically start in the late afternoon or early evening so that at least some of the games can be played at night under floodlights for international television broadcasts.

With each game more data (like balls faced, runs, wickets etc.) is being generated and with each chunk of data being produced, it becomes harder to keep a track of everything. The struggle of going through multiple files everytime motivated us to work on a solution.

3. **Project Objectives**

   The objective of this project is to combine our skills at creating beautiful visualizations and our knowledge & passion for cricket and turn it into something meaningful. This project is meant for people who would like to dive deep into the archives of IPL and use the data to create their own hypothesis. We are attempting to do this by creating a comprehensive dashboard while following the design guidelines taught to us.

4. **Related work**

   https://cj-mayes.com/2022/08/18/building-a-premier-league-dashboard-part-2/
   https://www.t20worldcup.com/home-page
   https://knoema.com/zaxeuv/indian-premier-league-2008-2014

5. **Questions**

   There were a few questions that we tried to answer at the start. They were:

   1. How did the teams perform in each match?
   2. Who won the match and by what margin?
   3. Who was the best player?

   As we moved ahead with the project, more questions rose up:

   1. Were the teams consistent through the years?
   2. Which areas on the pitch can be considered "strong" for the bowler and the batsman?
   3. Which pairing of batsmen is the strongest?

6. **Data**

   We were able to find a basic dataset from online sources (data). We used both the CSV files in this link for our project and then performed preprocessing as mentioned below. Columns like "extra type", "non-boundary"," kind" and "fielder involved" were not useful for us.

7. **Data Processing**

The data we have collected from the above-mentioned sources has provided us with the majority of the information needed for our visualizations. However, different columns are redundant for different visualizations and to remove those columns we perform data preprocessing.

1. **Partnership:** Our partnership graph required us to calculate the total runs scored by a batsman while there were different batsmen on the non-striker's end. To do this we had to perform grouping of our data based on overs and innings and then perform summation of all the rows where the same pair of batsmen exists.

2. **Runs Per Over:** To create a runs per over graph, we need to use our ball-by-ball dataset and then use the "total_run" column to find the runs scored in each over. There were two ways to go about this.
    - Use a loop and perform a summation operation after every 6 balls(each over has 6 balls in cricket).
    - Group our dataset by the "over" and "innings" columns to ensure the overs of each unique ID are not summed with the overs of another ID. Once the grouping is performed we perform addition on the "total_runs" column to find the runs scored in each over.

    We chose the second method two to prevent exceptions where the number of Overs are not exactly equal to 6 due to various factors like extras and reaching the target

3. **Wagon Wheel and Heat Map:** We have used python libraries like pandas and urllib3 for this particular dataset. Data which contained information regarding the location of the ball on the pitch and the sector in which the shot was hit is not available online. To find this information we performed Web Scraping on a website which contained live commentary of each match. The first step is to define keywords for both our requirements. For bowling length we have the keywords: ["Full Toss","Yorker","Full Length","full","fuller","Good length","length","short","good length] and for the wagon wheel we use the keywords:
    "cover","point","punch","punches","third","fine-leg","leg","square","square-leg","midwicket","midwicket","hook","pull","mid-on","mid-off","long-on","long-off","fine","edge","straight","sweep","down the ground","over the bowler's head","upper-cut".
    The next step was to use the url template and go through the commentary for each match defined by its unique ID. Once the commentary is accessed we then look for these predefined keywords and store the appropriate information in a dataframe.

8. Exploratory Data Analysis

Using our previous knowledge of the sport and looking at the dataset, we have selected certain parameters which are crucial to communicate enough information that are required in an archive. They are:

1. Playing 11

2. Ball-by ball score
3. Best batsman of each team
4. Best bowler of each team
5. Partnerships created by each batting pairing in both innings
6. Runs per over
7. Ranking history of both teams from 2008-2022 (Lucknow Supergiants don't have a ranking history as they joined the competition in 2022).

Once the parameters were set, we went through our datasets and selected the best visualizations for each of them.

1. **Playing 11**: There were two ways we could have displayed this information. One was with a simple table containing the names of each player and the other was creating a scaled down representation of a cricket stadium and placing each player along with their photograph with their position being defined on their role in the team

   We chose to use the 1st representation as we did not want to leave any ambiguity in the interpretation of our data, which was a major risk with the second visualization.

2. **Ball-by-by score**: This was one of the most important features of our project as there isn't a lot of previous work done on this aspect of the match. There were multiple ways to display this as well. The first being, making a curved rectangle for each ball and grouping all the balls into a single bar based on the over number for both innings. This visualization would have provided the user with in depth information but would make it very difficult for the user to compare the way both innings progressed. The next approach was to make two line graphs which would show the change in the score with each ball. The color of each path was decided by the team colors.

   We went with the second approach as it resolved the issue of comparison while making sure the data is communicated properly without risking misinformation.

3. **Partnership:** Partnership is the total of the runs scored by each batting partnership, our preprocessing provides us with the total number of runs scored by each batsman along with another batsman. To depict this we made a selection between a node graph, bi-directional bar graph or a multi-variable venn diagram.

   The first two options were the most valid options for us as they are the best ways to show how an attribute is shared between a pair and also helps us compare each participant individually. A node graph is the second-best approach for this particular task as the number of nodes will be too high and varying which would cause a drastic drop in the readability of the graph. On the other hand, creating different bi-direcitonal graphs for each innings gives a much clearer idea on the

partnerships created during the match and also help us with finding the best batsman in the match by looking at the length of the bars.

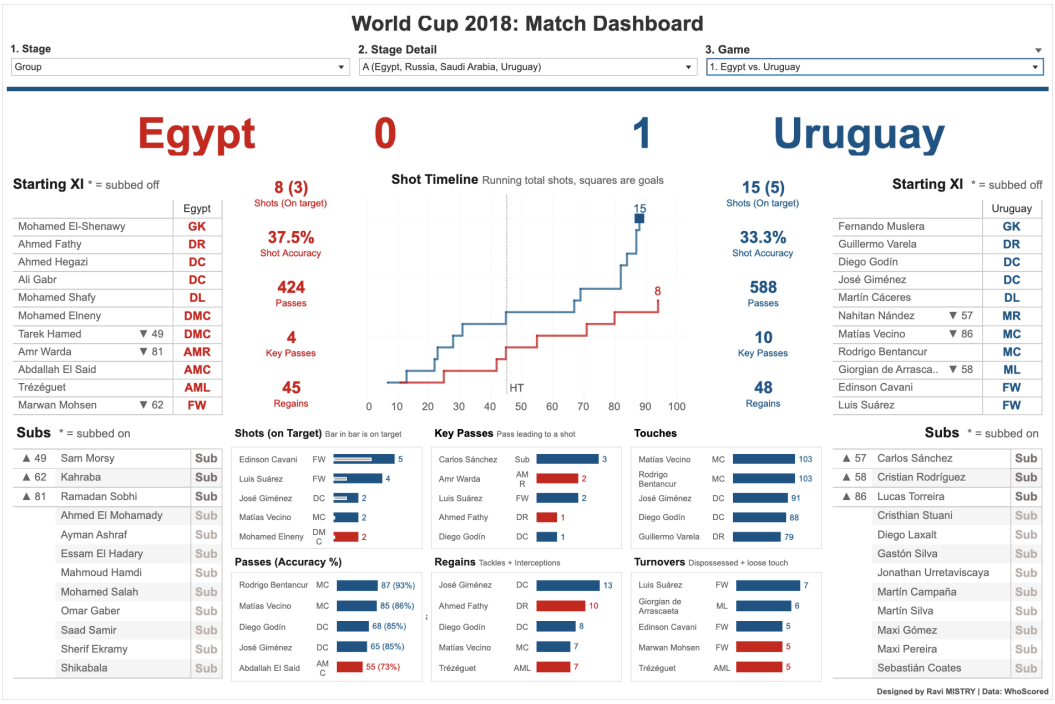
4. **Runs per over:** This parameter is used to understand how the team performed throughout the match, to do this we used a bar graph where the height of each bar depicts the runs scored by the team per over. This could also have been represented by using an area graph but that would compromise the accuracy of the reading.

5. **Ranking History:** We had the option to use a number line for each year and then highlight the teams or we could use a line graph to show the changes in the ranks held by each team over the years. The line graph visualization would have proved to affect the user experience negatively as they would have to scroll through all the years to reach the end, which is why we made a line graph.

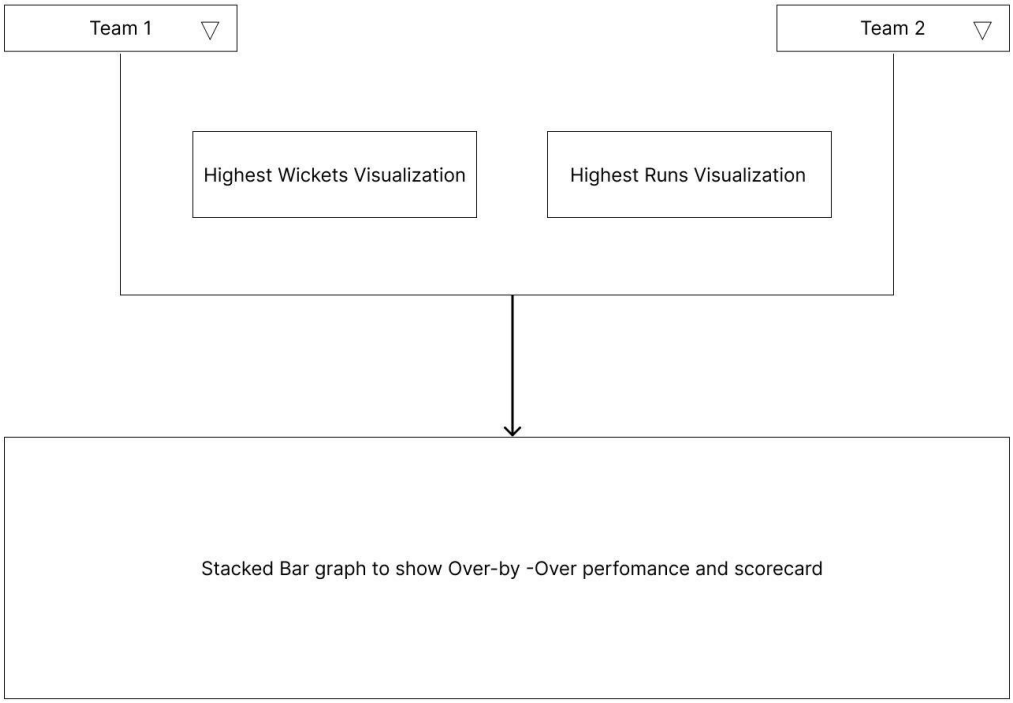
   The only design options that we could think about for the **Best bowler** and the **Best batsman** were the heatmap and the wagon wheel respectively. The heatmap depicts a cricket pitch and each section depicts a different section of the pitch.

9. Design Evolution

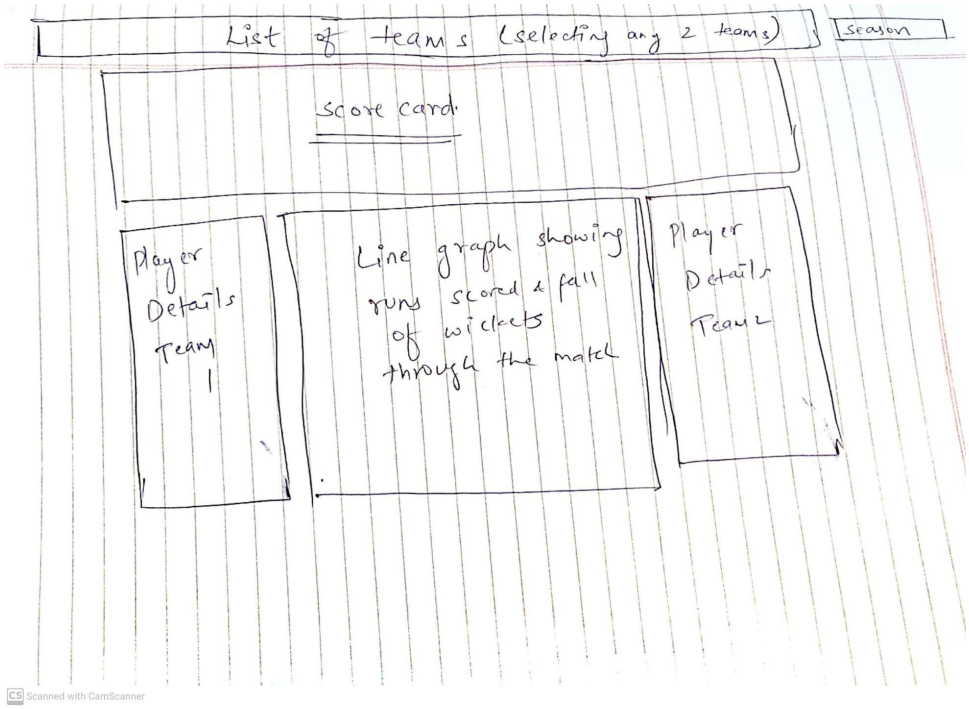
Reference Visualization:



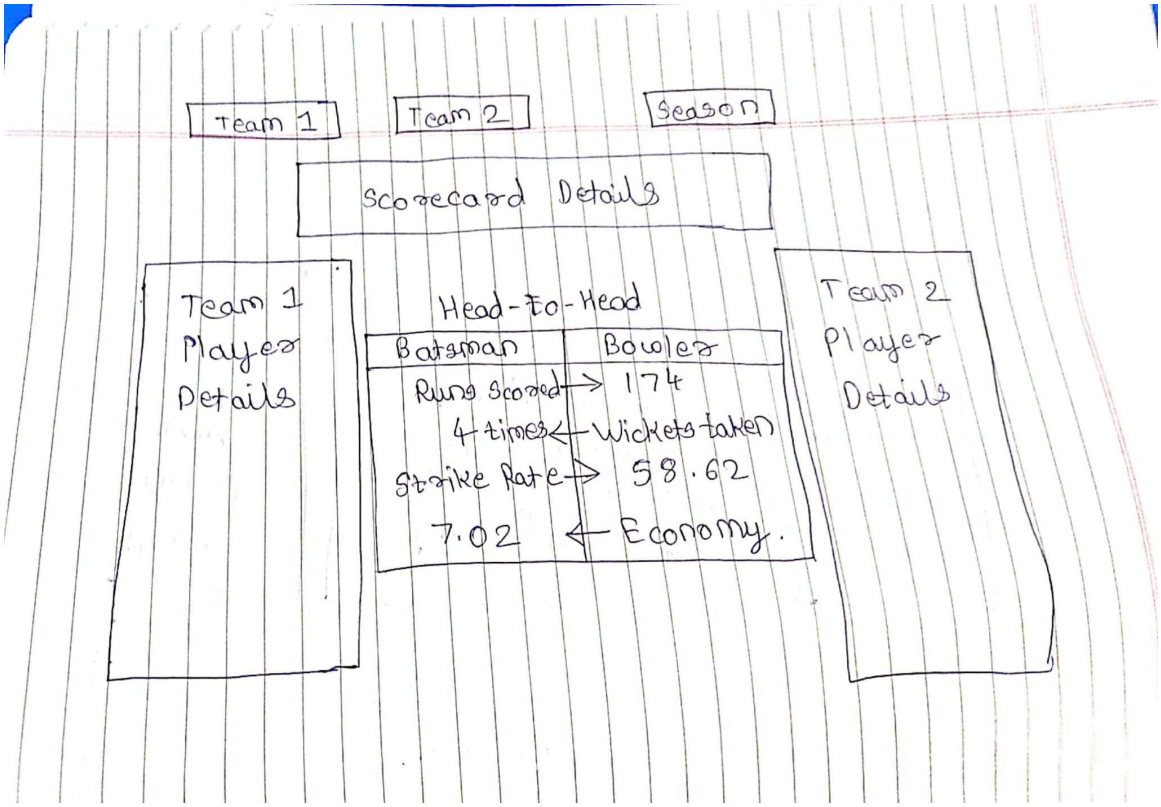
Design 1:



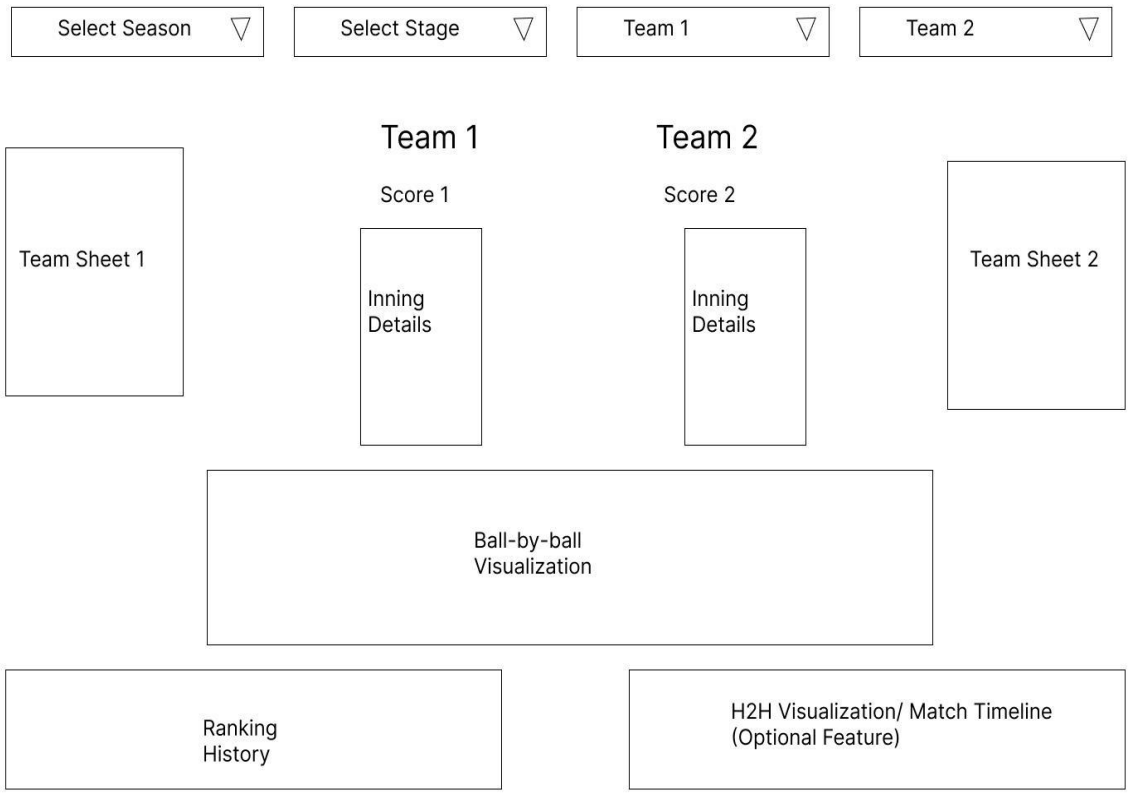
Design 2:


List of teams (selecting any 2 teams)
season
score card
Player Details Team 1
Line graph showing runs scored & fall of wickets through the match
Player Details Team 2

Design 3:


Team 1
Team 2
Season
Scorecard Details
Team 1 Player Details
Head-To-Head
Batsman
Bowler
Runs Scored → 174
4 times ← Wickets taken
Strike Rate → 58.62
7.02 ← Economy.
Team 2 Player Details

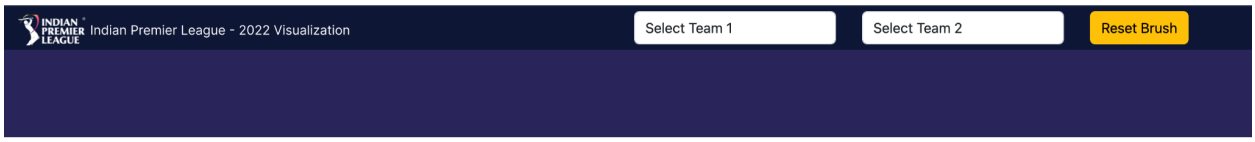
Design:



10. Implementation

- **Header -**

The header is the first thing that shows up when the dashboard is loaded. It asks the user to select the two teams they are interested in.

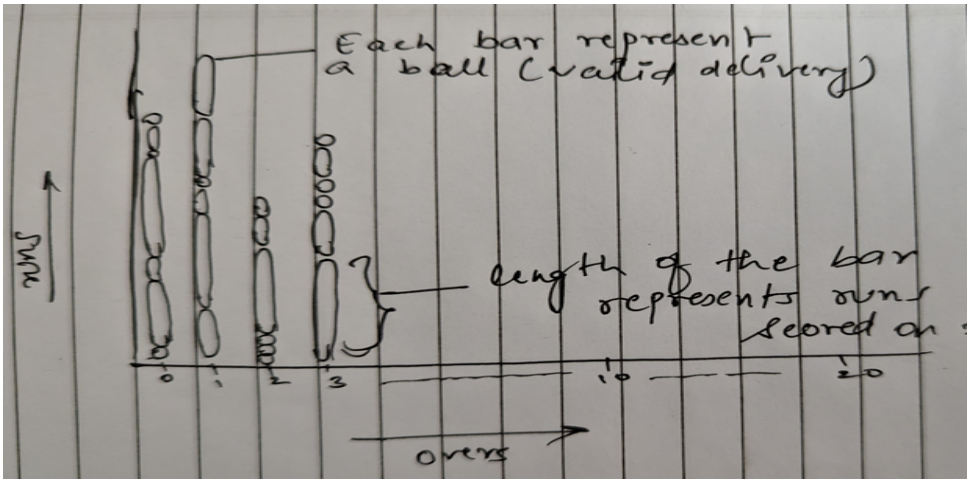


As soon as the user selects the teams, our dashboards look through the dataset for the same pairing of teams along with the match ID assigned to it. This match ID is extremely important as all the visualizations can only identify the data they need to use through the corresponding match ID.

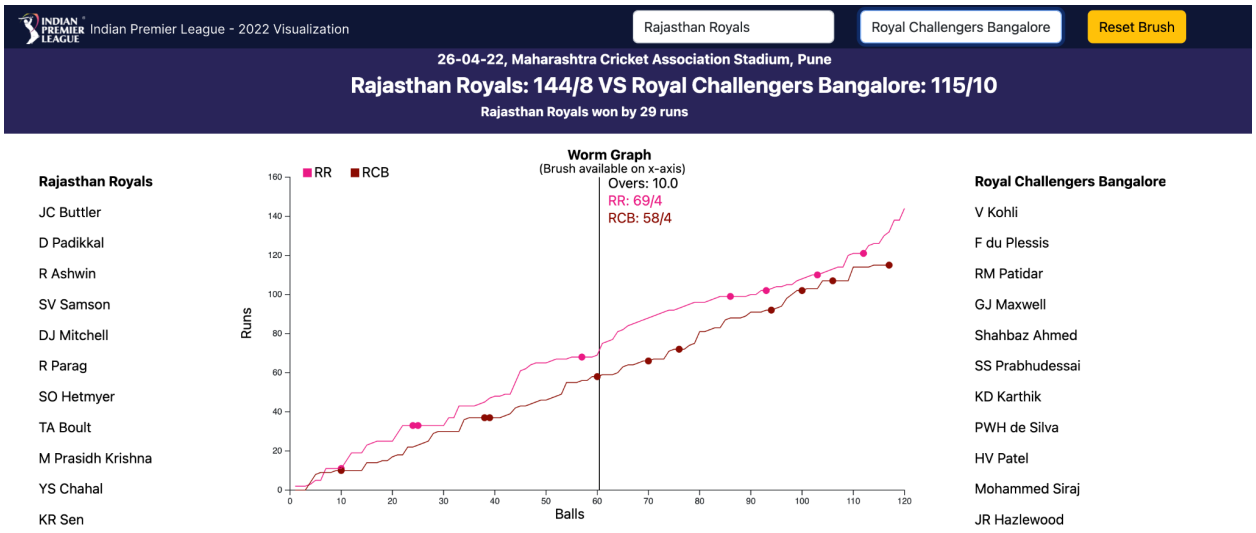
Once the teams are selected by the user (in this case, Rajasthan Royals and Royal Challengers Bangalore), we display an info tab. This tab contains information like the date, venue of the match, the scores of both teams and finally the result of the match. This data is obtained for the "IPL_Matches_2022.csv" file using the columns "WonBy", "Margin" and "method." This is done so that the user can take a quick glance at it to factor in various parameters like pitch type, weather etc.

- **Ball by Ball comparison –**

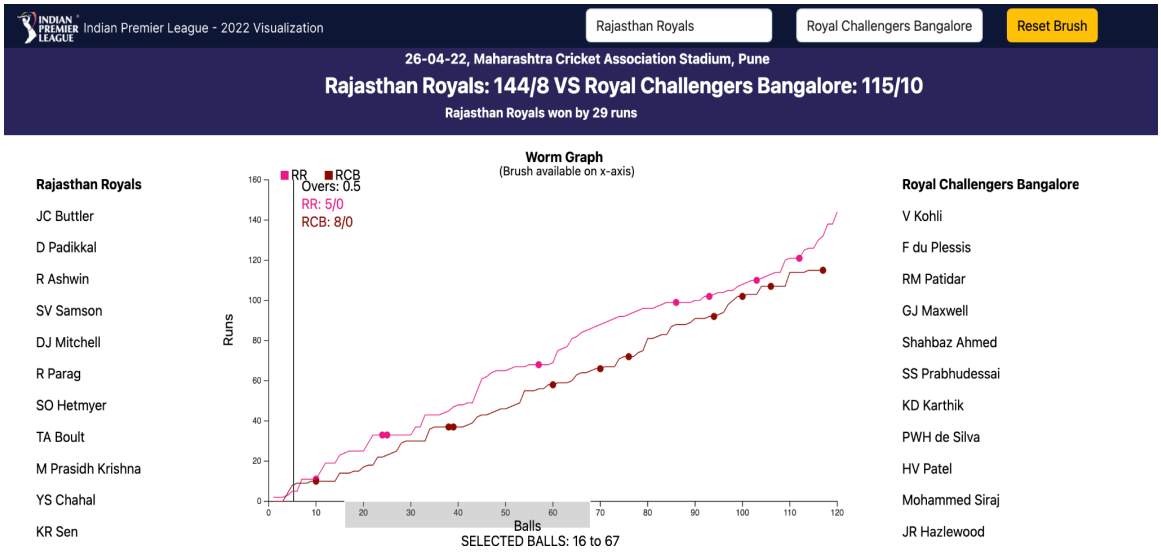
  Initially, we thought of the following design for a ball-by-ball comparison.



In this design, each bar represents a valid delivery, and the size of the bar means the runs scored on that delivery. We would have displayed two such graphs, one for each inning. But this graph wasn't intuitive as it would have been hard for the user to compare the innings on separate charts. It's easy to compare the innings when they are portrayed on the same chart. Hence, we decided to go ahead with a line chart (worm graph).

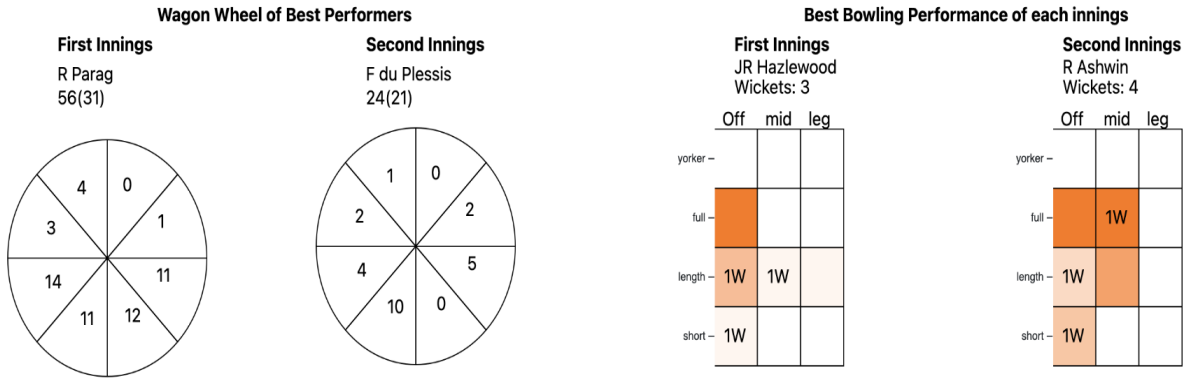


Here we display a worm graph that shows the ball-by-ball performance of both teams. The chart provides a high-level view and allows users to dive deep for further analysis. The user can hover along the worm graph to look at the precise scores of both teams at a particular point. Even if the user does not choose to hover over the graph, they can use it to compare the performance of both teams. The circles along each line represent a fall of a wicket. This graph helps compare the performances of both teams throughout the match. With the hover functionality, the user can determine whether a team dominates certain match stages.
Along with the worm graph, we also display the playing 11 of both teams.



We thought of providing more power to the user to analyze a particular game period by selecting the overs during which they want to analyze the game. Hence, we have added a brush to the worm graph to allow the user to select certain stages of the match, affecting the graphs displayed later. There is a "reset brush" button in the header, which is also used to send the graph back to its default selection.

- **Best Player Analysis -**

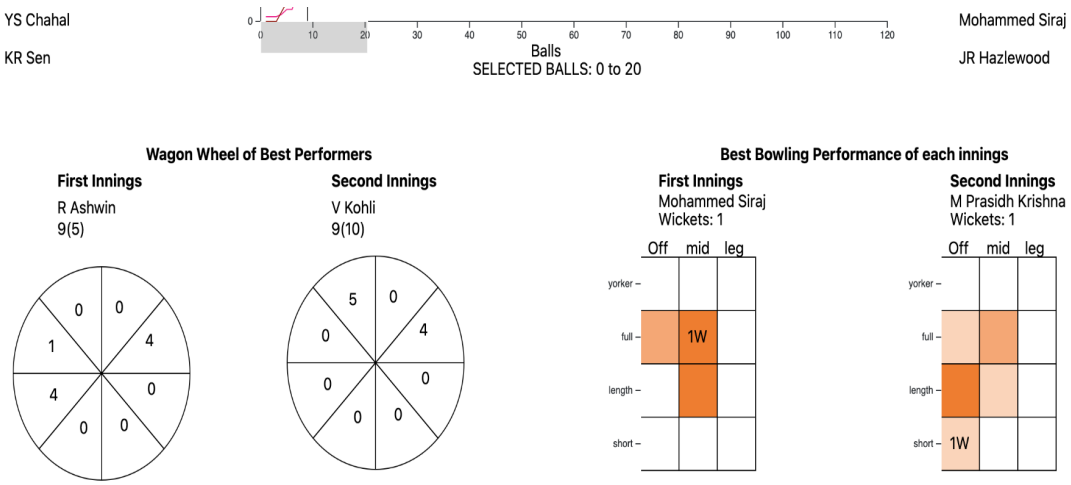


The next two graphs that we look at are the Wagon wheel and the pitch data. We obtained the dataset for both these visualizations by performing web scraping on a website that provides live commentary of each match (as mentioned in the "Data Processing" section).

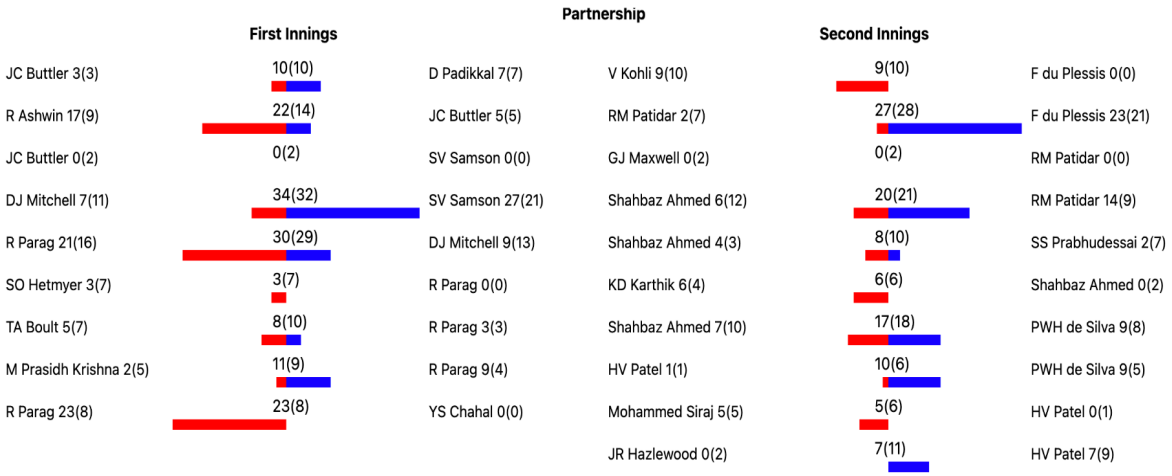
Not every player can score runs all around the ground. Wouldn't it be best for the user to analyze the area where the highest scorer of that match has scored runs? For ex, in the above screenshot, a player (i.e. Riyan Parag) scored 14 runs in the covers region. This feature is also helpful for the opponent team because they now know the opponent player's strongest area. This feature allows users to analyze better and enable the opponent to better strategies. The wagon wheel depicts a cricket stadium, and each sector represents a different part of the field, e.g., the bottom two sectors and long on and long off and so on. Each wagon wheel shows the best-performing batsman in each inning, and the number inside the sector shows the runs scored by the batsman by playing shots in a particular part of the field.

Understanding the bowling performance is one of the most crucial aspects of the game. It's essential to analyze in which area a particular bowler bowls and what's his wicket-taking area. It helps the batsman better prepare for that bowler. The heatmap shows us the cricket pitch, and each rectangle shows different parts of the pitch, as shown by the axes. We are assuming that the stumps are placed at mid. A darker color shows that the bowler pitched more balls in that area. The number inside the boxes is the number of wickets the bowler takes when the ball is pitched in that specific section.

Both these graphs help us identify the best-performing batsman and bowler during both innings. If the user has selected a particular number of balls in the worm graph, both graphs will get updated accordingly and show us the best performers within those balls, as seen in the screenshot below. This feature is essential because players play differently in different game phases. The batsman plays differently in the first four overs(24 balls) as compared to the middle phase. Similarly, bowlers have separate line and lengths according to the match phase. This feature empowers the user to analyze the different match phases and to get information about the best-performing players during that phase.

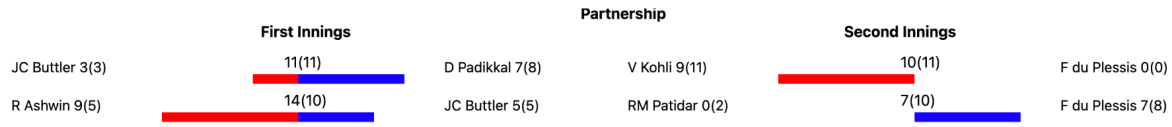


- **Partnership -**



It is said in the game of cricket that partnerships can win you matches. Therefore, we now shift our focus to partnerships. Here we use a bi-directional bar graph to depict the number of runs by a batsman along with his non-striker. The number on top of the chart shows us the total runs scored in the partnership and the number of balls taken to reach that score. The number next to the player's name indicates the number of runs scored by that particular player and the number of balls taken to do so. We did not have this information readily available in our dataset and had to use a grouping function over the "batter," "innings," and "ID" columns. Once this was done, we used an aggregation function to sum the "total_run" column to find the total number of runs scored by each player.

Users can also analyze which two players build partnerships during a game phase. This graph gets updated along with the selection of balls in the worm graph; the updated graph shows us the runs scored by the players batting at that particular stage, as shown in the screenshot below.

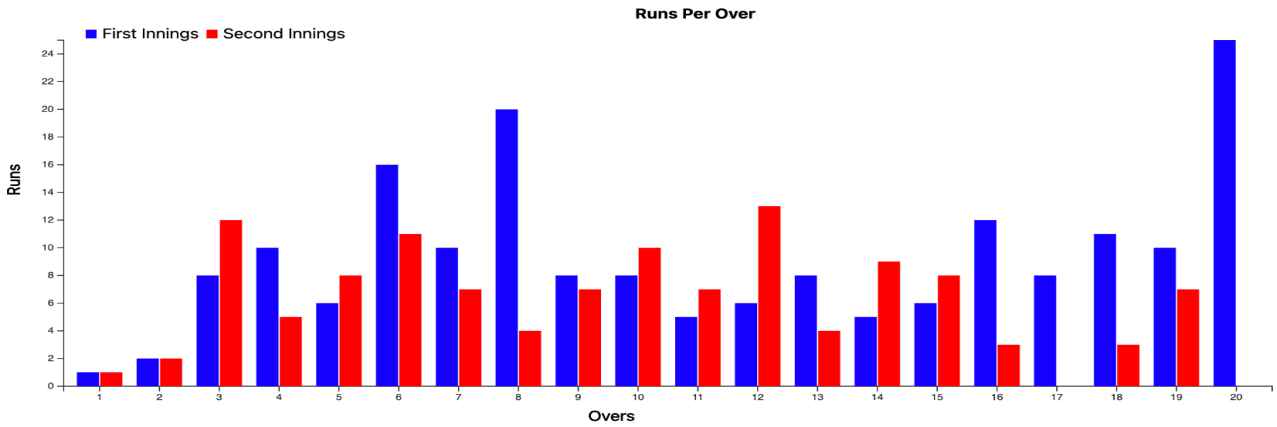


- **Runs per over graph -**

The next graph we have is the Runs Per Over graph. This graph can be used to understand better the rate at which the teams are scoring runs and better understand

players' performance at different stages of the game. The graph's height shows the number of runs scored by the team in that particular over.

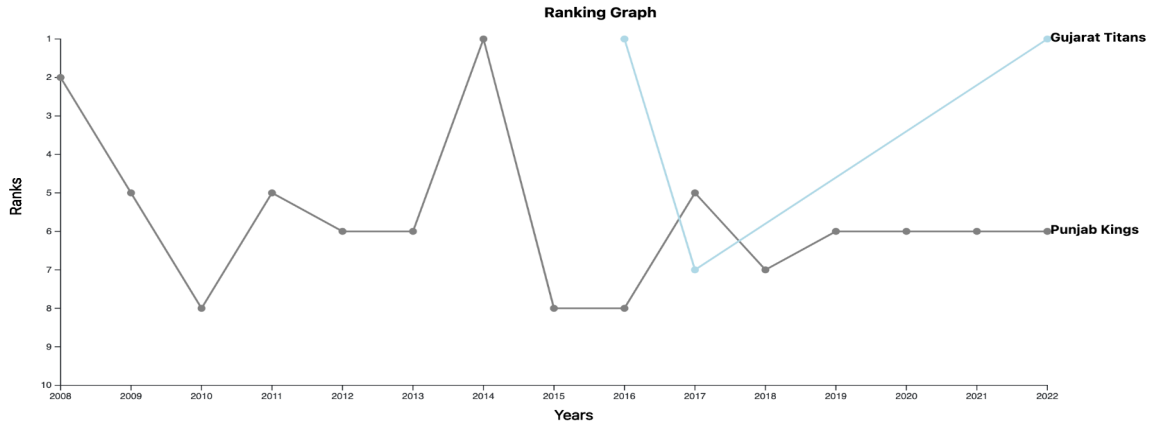
The data for this graph was not available to us, and we had to perform some grouping and aggregation functions over the dataset. In Particular, we used the "ID," "innings," and "over" columns for grouping and used the "total_runs" columns for the aggregation function.



We decided not to update this graph along with the selection of balls in the worm graph as we are already dividing the graph readings by overs.

- **Ranking Graph -**

This brings us to the end of all the graphs affected by the match ID. The final graph is a ranking graph showing the selected teams' historical performance. This graph helps us understand the consistency of a team over the years and lets us know whether any of the selected teams have won the tournament, if yes, in which year. There are a few exceptions, namely, Lucknow Supergiants and Gujarat Titans, who have participated just once and twice, respectively, over the 14 years of IPL.



11. Evaluation

The original dataset had a lot of data which would be very difficult to understand for someone who does not understand the sport. An analyst would also find it very difficult and tedious to go through the vast dataset to look for the required information. This dashboard makes it easy for everyone to access the information and helps make decisions regarding the stronger team/player.

The visualizations worked as we expected and provided us with appropriate information. We tested the dashboard with someone who had no prior knowledge of the sport and could understand the visualizations displayed, proving the project was a success.

The visualization was also able to answer some of the questions, such as:

1. Who is the best batsman of the match?
2. In which region the player scored the most runs?
3. Who is the best bowler in the game?
4. Which length did the bowler bowl to get the most wickets?
5. What is the highest partnership?
6. What are the rankings of the teams from the past 14 seasons?

There are a few further improvements that we could make to it:

1. We could make the same dashboard from 2008-2022 if we have the dataset
2. The design of the current dashboard is fundamental. We could make it even more user-friendly with enough time and correct input.