# UNIVERSITY OF UTAH

# DATA VISUALIZATION PROJECT FALL 2018
# (CS-5630 / CS-6630)

## INDIAN PREMIER LEAGUE SEASON DATA

BY:
BRETT REEDER (U0921074),
SRIVATHSAN GOMADAM RAMESH (U1208099),
ABHINANDAN BAIREDDY (U1210208)

# Table of Contents

**Project Proposal 10/28/2018**

1) **Basic Info**

Indian Premier League Stats Visualization:

**Team Member 1**

**Name**: Brett Reeder

**uID:** u0921074

**Email**: BrettReeder63@hotmail.com

**Team Member 2**

**Name**: Srivathsan Gomadam Ramesh

**uID**: u1208099

**Email:** srivathsangr16@gmail.com

**Team Member 3**

**Name**: Abhinandan Baireddy

**uID**: u1210208

**Email**: abhi.baireddy@utah.edu

**Project Repository:** https://github.com/breeder88/visproject

2) **Background and Motivation**

The IPL (Indian Premier League) is the world's largest cricket league in terms of attendance, and is the sixth most attended sports league in the world, with a value of $6.3 billion. In IPL, 11 Indian teams compete against each other in a group stage, earning 2 points per win. The top 4 teams compete for final bracket. While many visualizations exist for sports such as basketball (march madness especially) and football. Most of the statistics of IPL cricket games are simply located in tables with queries. While this is useful for finding specific stats, these formats are not easily digestible, nor do they offer the ability to directly compare teams or players.

The present project aims to analyse IPL match data from the beginning of the IPL (2008) to 2016. Both teams and key player stats will be visualized over the seasons, offering team comparisons and visualizations for performance over multiple seasons. Current visualizations don't offer this format for viewing the data, so visualization of statistics over time, as well as comparisons, would be welcome.

3) **Project Objectives**

This visualization project aims to analyse the patterns in team and player performance within a season seasons. Both batsmen top bowlers will be listed for teams that play against each other, allowing for the user to see which players consistently perform well in each team.

4) **Data**

Player and match data from 2008 to 2016 was retrieved from github: https://gist.githubusercontent.com/masterofpun/b66ce9311cc27946d0597dbe1bc8833a/raw/ipl_stats_2016.json

More match data was retrieved from Kaggle: https://www.kaggle.com/manasgarg/ipl#matches.csv

The match data is in the form of a json, which contains an array of games. Each game has player data for each round, from which the relevant stats can be derived. Season long stats for wickets and runs should be tracked for all players, and ranked for each game, where the top players may be shown for every game.

The additional match data shows the winning teams and their margins of victory and loss. The data is in CSV format and needs to be edited so that it contains consistent IDs for each team.

5) **Data processing**

Player stats must be calculated from game stats over all games for season stats, aggregating player information for visualization. Season-long stats for wickets and runs should be tracked for all players, and ranked for each game, where the top players may be shown for every game. In addition, teams should be linked to states, which may be shown on a topojson map of india. Game data will also be compiled into season-long data for ranking teams across seasons.

Since the data is relatively short, season and state data can be obtained from the IPL website. Game and player data may be added to the json file using javascript to process and alter the base json file.

6) **Visualization design**

The data will be displayed in various standard charts and tables, as tables are very good for ranking teams, can be sorted, and have a clear hierarchy. A specialized chart will be developed to display game stats for a selected team, having a central team with each game arrayed around it like spokes on a wheel or marks on a clock.

Bar charts and line charts can be used to compare teams and show players' performance over time. Comparing bar sizes is a good way of encoding information, and numbers may be added to the bars to further enhance readability and accuracy. Line charts are excellent at showing changes over time, thus will be used to show how top players perform between seasons. A map of India with highlightable states would help to encode teams geographically and creates a compelling center piece for the visualization webpage.

7) **Must have features**

Team stats must be displayed with sorting, as these are crucial to understanding performance during a season. A season timeline is also essential to the visualization, providing an easy way to select seasons. Finally, flashier visualizations are essential, and a custom team selector should be implemented, as well as a map of India to select teams.

8) **Optional features**

Additional useful features may be added, if time permits, including game by game data, player season statistics, and a final 4 bracket. Adding team logos would also make icons more recognizable and compelling and should also be considered for addition. Finally, adding more in-depth player stats that allow for player by player comparison for the top 50 or 100 players would improve the visualization's usefulness by offering more information on the players, not just the teams.

9) **Project Schedule**

**Oct 28 - Nov 3:** Derive all stats and create any additional data files needed. Implement india map with states colored correctly for each season. Implement season timeline. Create table dataset, derive winners, top batters, top bowlers, and losers from the games dataset.

**Nov 4 - Nov 9 (milestone 1):** Implement the sortable table, implement the team game view, and winner/loser table. Implement team comparison graphs and player stats over time.

**Nov 10 - Nov 17:** Add additional comparison features and consider the optional objectives. Flesh out the process book.

**Nov 18 - Nov 24:** Finish adding any optional features, clean code and write documentation (READMEs, ect).

**Nov 25 - Nov 30:** Create screencast, finish process book with insights, and develop a story from the data.

**Feedback Session 10/30/18**

Aaron Young

Something to explain the rules of cricket to a user who might not be familiar.

**Add link to cricket basic rules.**

What's the focus of the visualization? Teams or players?

**Focus on teams, people mainly follow their state team more than players.**

Maybe add an arrow or dial to show time progression in the custom visualization.

**Redesign the custom visualization (Fig 1.)**

How will you interact with data? Just options? or would there be anything more advanced? (brushing can be implemented in the over time data)

**Implement brushing for line graphs so the user can view subsets of games.**

What views will be linked?

**Season timeline will link to the map and all game data. The table will be linked to the custom visualization, and the custom visualization will be linked to the comparison view and season view.**

Can you show a performance over time plot?

**This will be accomplished by the season view, which will show performance over time for the team of interest game by game.**

Maybe you should do visualization only by a single time scale (season, team, ect). It seems difficult to do all these different things.

**We can change to focus mainly on season data, which will have a time scale based off of games.**

Make the dynamic drawing a must have.

**Dynamic drawing is easy to implement, can be used to draw the line charts.**

Kevin Le

The custom visualiztion doesn't make it clear that time is something that it is sorted by. Maybe a vertical linear arrangement would be better. It should be rearranged to show the progression of the season better.

**Redesign the custom visualization (Fig 1.)**

Abhialash Tiruveedhula

Why not rank teams with a stacked bar chart instead of side by side bars? it shows differences better.

**Stacked bar charts would be a better way of showing team by team differences. Numbers can be overlaid atop the bars to show exact numbers.**

Make a line chart showing season ranking over time.

**This can be a stretch goal, it may clutter the view, so its position should be considered carefully.**

**Addressing feedback**

The feedback that we received was especially helpful in narrowing down the scope of our project, focusing on one aspect of the data (teams or players), along with what time intervals would be of interest (season data, all time data, or game by game data). The peer feedback session also made us realize that the custom season game visualization also needs some rethinking to better show the progression of time over a season. The clock-like setting doesn't make the chronology very clear, but a timeline would make the game order very clear.

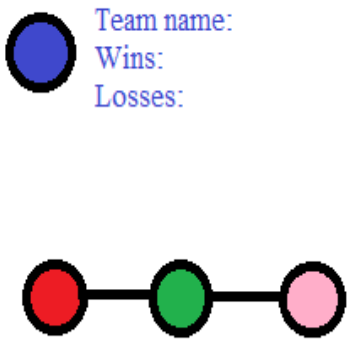



Fig 1. Timeline design, which shows clear linear progression between games, which are represented by the connected and colored circles.

**TA Meeting and Revisions 11/8/2018**

The TA meeting suggested to us that we should expand the scope of our visualization and rethink a couple of the suggestions from the peer feedback session. Firstly, rather than narrowing down the visualization to only give team information by season, we decided to expand it, but keep each visualization distinct by using multiple different tabs, as seen in Figure 2. Team comparison data has been shifted to a new tab to make the season long visualization more coherent. The player statistics have also been shifted to a new tab for similar reasons. The season timeline also got a redesign, visually encoding wins and losses by ups and downs.

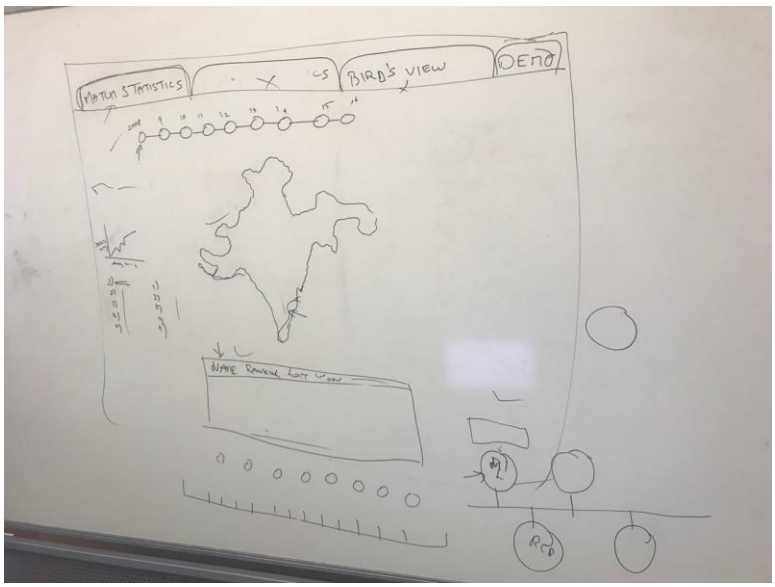


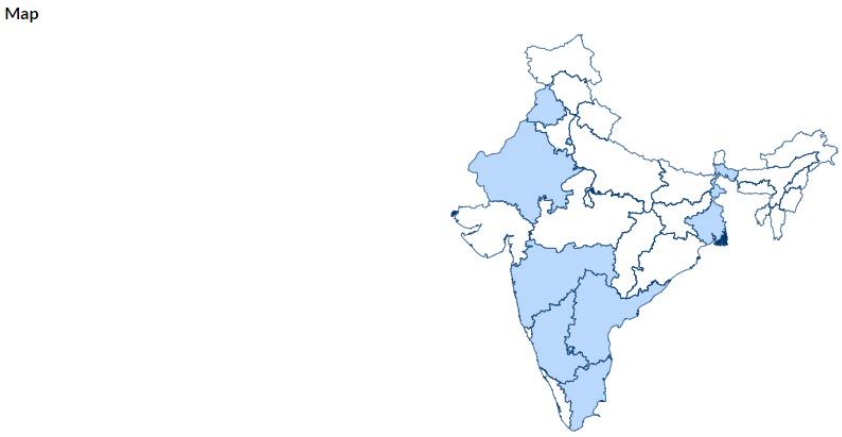

Fig 2. Redesigned visualization, separating categories by tabs and adding an improved game timeline.

**Milestone 11/9/2018**

Data has been processed, adding attributes that cover the winner of each game, the city in which the game was played, and whether or not the game was a home game or an away game. These attributes may be used to determine homefield support, and assign game winner statistics more easily. In addition, a season timeline prototype has been implemented, depicted in Figure 3. Team logos are used to provide a very easy to distinguish visual way of determining the winner of a season.


Fig 3. Preliminary timeline design, showing season champions for each year, using their logos.

A map has been implemented that shows the states that participated in the league in the season selected from the season timeline with solid fill. The map is same for all the seasons except two seasons where new teams were added to the league.




A table that shows the season standings of all teams has been implemented.

**Season Team Rankings**

| Team | Played | Won | Lost | Net Run Rate | Points |
|---|---|---|---|---|---|
| Rajasthan Royals | 14 | 11 | 3 | 0.632 | 22 |
| Kings XI Punjab | 14 | 10 | 4 | 0.509 | 20 |
| Chennai Super Kings | 14 | 8 | 6 | -0.192 | 16 |
| Delhi Daredevils | 14 | 7 | 6 | 0.342 | 15 |
| Mumbai Indians | 14 | 7 | 7 | 0.57 | 14 |
| Kolkatta Knight Riders | 14 | 6 | 7 | -0.147 | 13 |
| Royal Challengers Bangalore | 14 | 4 | 10 | -1.16 | 8 |
| Deccan Chargers | 14 | 2 | 12 | -0.467 | 4 |

The table and the map have the additional functionality to select a team. This functionality is not implemented yet. The selection would alter the team season timeline implemented later.

**Added Timeline for Games with a Selected Team 11/11/2018**

A prototype timeline for every game that a selected team played in a season was implemented. The opposing team's logo is shown upfront, but their size may need tweaking, as they can be very compressed. Games that the selected team wins are shown above the timeline, while games that the selected team lost are below the timeline. Everything is chronologically ordered. The view still needs work, highlighting, tooltips, the currently selected team, and the team selection portion of the view still need implementation.

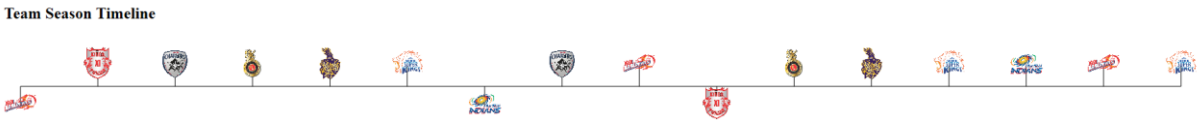



Fig 4. Season games timeline Prototype

**Added linking between all views, reset button 11/13/2018**

Linking between the season timeline, game timeline, and team selector was implemented. Transitions will be added to smooth the process of switching between views. Tooltips need implementation, however, and linking with a team comparison view needs to be completed as well. It may also be useful to list the date of the game on the opposite side of the timeline to make the chronology even more clear. Fading in the teams along the timeline in chronological order may also be another way of demonstrating the progression of games over a season.


Fig 5. Season games timeline with reset button implemented

**Post Milestone Team Meeting 11/17/2018**

Various optional features were discussed the "bird's view" feature was finalized. The bird's view feature would show the statistics of teams or players across all the seasons. It consisted of the following visualizations.

1. A stacked bar chart showing the influence of home ground on a team's performance

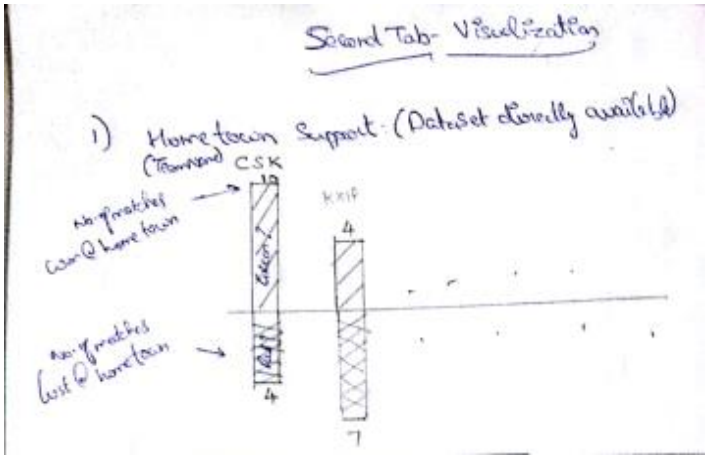



Fig 6. A stacked bar chart showing the influence of home ground on a team's performance

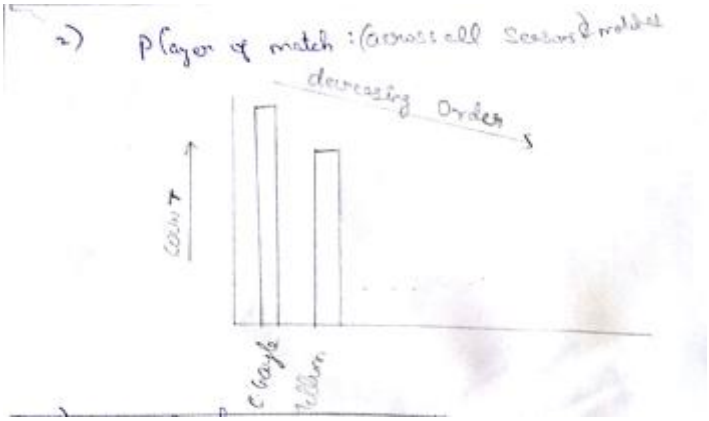



Fig 7. A bar chart showing a player's performance across all the seasons

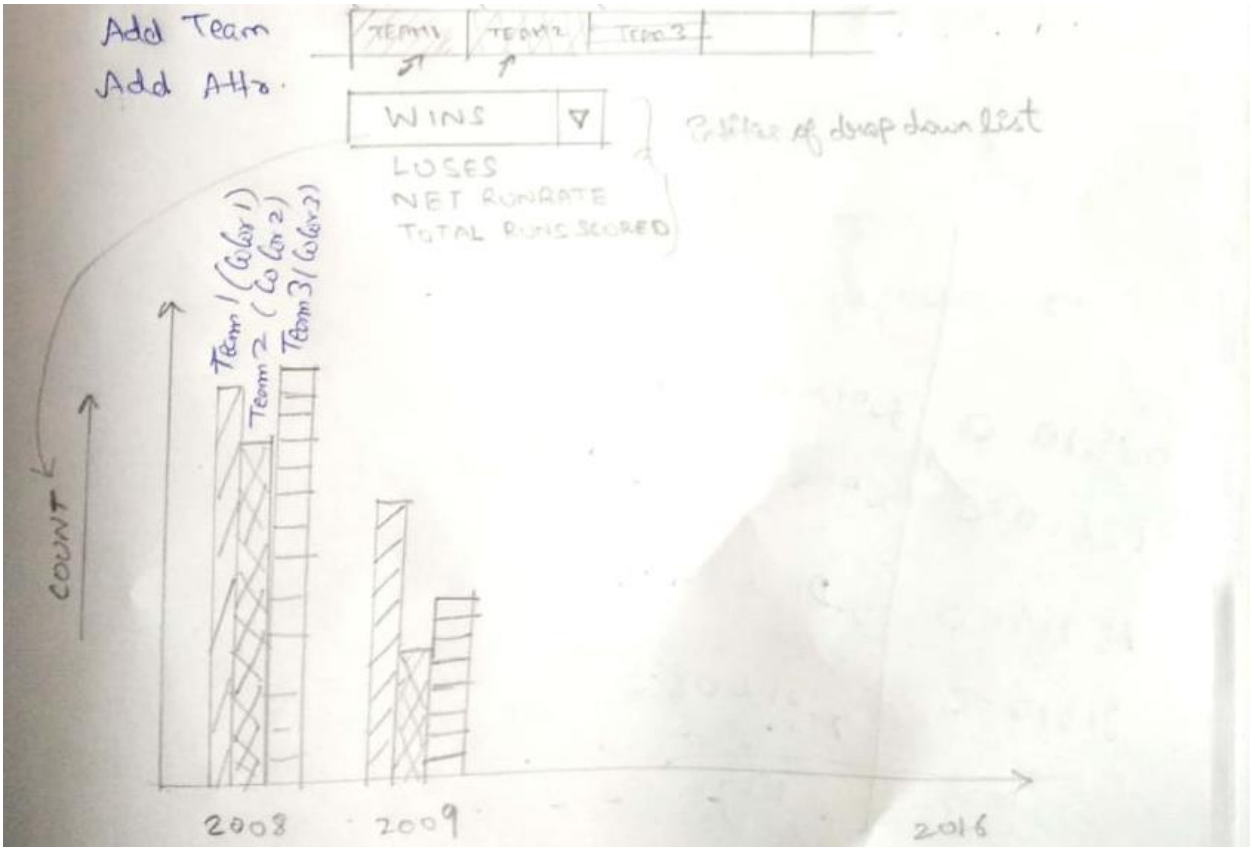



Fig 8. A comparison chart that compares teams. The user can select the teams to compare and attributes over which the comparison is done.

**Game selection from game timeline 11/18/2018**

A game statistics view has been implemented, appearing when a specific game is clicked (if data is available). The teams are compared and color coded to act as a legend for the game-based plots. Data shown in this game selection view include the date which the game occurred, the stadium in which it was hosted, and the coin toss winner's decision. These plots will feature batting and bowling statistics for the game. Batters and bowlers are treated as data points. A line plot will show batting data for the game, while a parallel coordinate plot will show the bowling data for the game.


Fig 9. The game selection view, which acts as a legend for the subsequent plots.

**Game batting data view 11/20/2018**

A batting data view has been implemented for the selected game, showing runs vs. player number (first batsman, second batsman, etc.). A tooltip shows more in-depth data about the player, such as their strike rate, team name, and player name. Sometimes, two datapoints coincide; in this case, the size of one of those datapoints is increased to allow the user to mouseover both the datapoints, thus preserving information. The game selection view acts as a legend, where the red team is the first team to bat, and the blue team is the second team to bat.

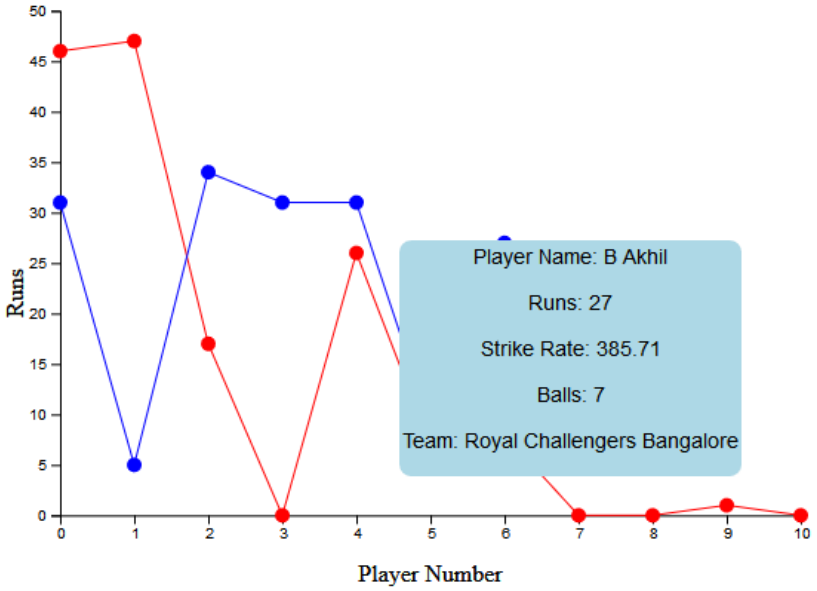



Fig 10. Batting plot showing runs over the course of the game for the batsmen.

**TA Meeting and Revisions 11/21/2018**

The TA reviewed our work so far and suggested a few design and styling changes.

1. It is not obvious that in the season timeline, the logos correspond to the winners of that season. Use circles for the season timeline and keep all team logos below the timeline. Highlight the logos according to the selected season.

2. Figure out a less ambiguous way to colour the states of the map as some teams have the same logos with the same colours.

3. Figure out a way to show the relationship between areas of the map and the teams that represent those areas.

4. There is a lot of white space next to the table. If it remains so even after finishing the styling part, figure out a way to minimize the white space by adding in new details.

**Addressing TA Feedback**

1: Changed the heading to “Season Winner Timeline”

2 & 3: Implemented synchronized highlighting between map and table. Clicking on an active state of the map will highlight the corresponding team in the table. Clicking on a row of the table will highlight the state on the map that is represented by the selected team.

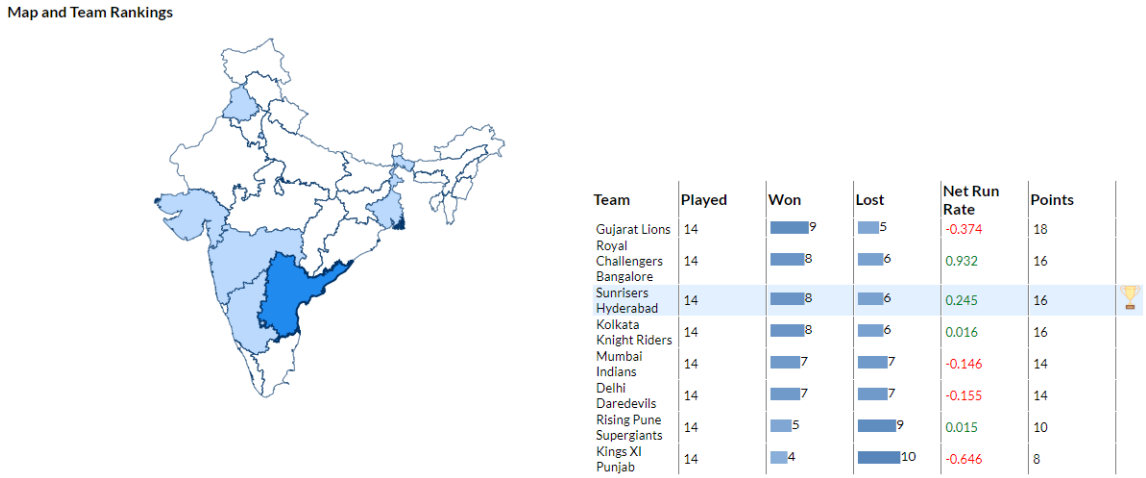



Fig 11. Map and table in the side-by side arrangement.

4: Placed the table beside the map.

**Game bowling view 11/25/2018**

A bowling data prototype has been constructed to show the many dimension of the bowling portion of the dataset. Economy, runs conceded, wickets taken, and overs bowled are shown. The players’ names are shown on the left-hand side and moving the mouse pointer over them highlights a player’s line, helping the user see through the complexity of the graph for any player. Team 1 is shown from the top down, while team 2 is shown from the bottom up. Again, the team is distinguished by the same colour scheme as that of other corresponding game view visualizations to make it intuitive to understand, ensuring consistency throughout all the visualizations.

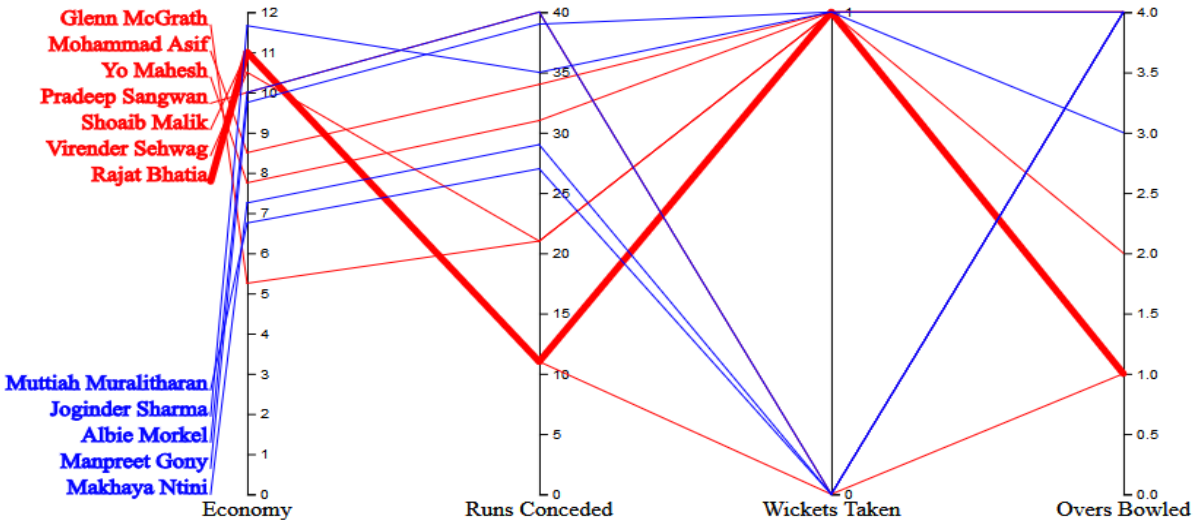



Fig 12. Parallel coordinates plot for multidimentional bowler statistics.

**Home team support graph 11/30/2018**

A graph to show home team support over all seasons using stacked bars was implemented. The data was filtered so that only the relevant games, those played at the team's home stadium, where considered. Only the top 8 most prolific teams were shown. Some teams only played one or two seasons and were omitted to make the visualization more even. A tooltip shows exact stats for wins and losses, while the colours distinguish the two categories. Red's negative connotations made it an obvious choice for losses, while blue provided contrast for wins.

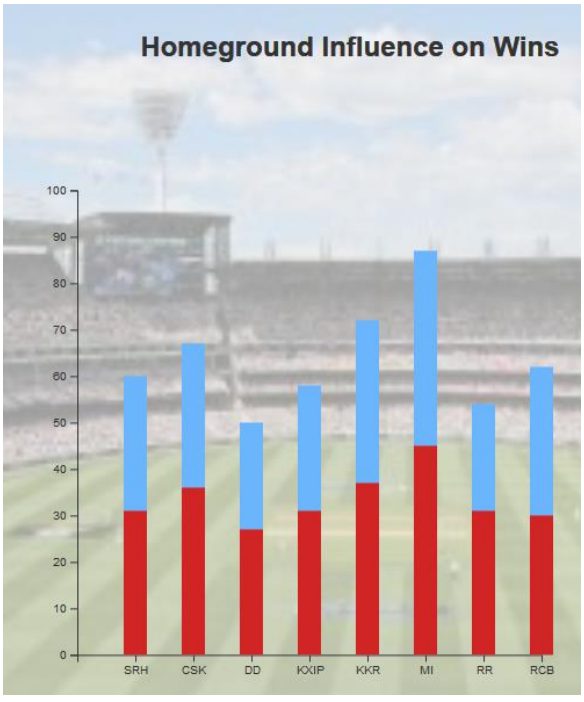



Fig 13. Hometown support plot over all seasons.

**Conclusions**

As the project progressed, it became clear that the focus would need to be narrowed in on a specific aspect of the data. This could have been team performance, all-time team performance, and player performance. We decided to focus on team performance over a season, as our data was best suited to team data. The season-long time scale was chosen to reflect additions and removals of specific teams with each season. While this season scale was suitable, we still wanted to be able to see some all-time performance statistics, as well as tell who played on each team. This resulted in our game view and "panorama" view, which focus on player data for any given game and all-time home wins, respectively.

We visualized the data as we went, not doing any particularly in-depth exploratory data analysis beyond basic counting in scripts. The counting was used to determine if any special cases needed to be accounted for in a visualization (overlapping datapoints, no data games, ect.). This prevented us from focusing too narrowly on our exploratory visualizations and kept development flexible.

This project is intended to help fans of cricket learn about the IPL, which they may not be too familiar with, as the game has many different leagues across the globe. It focuses on implementing an easily explorable interface that starts large, with season data (fig 14.), but allows for the user to drill down into details like game and player stats (fig 15.).

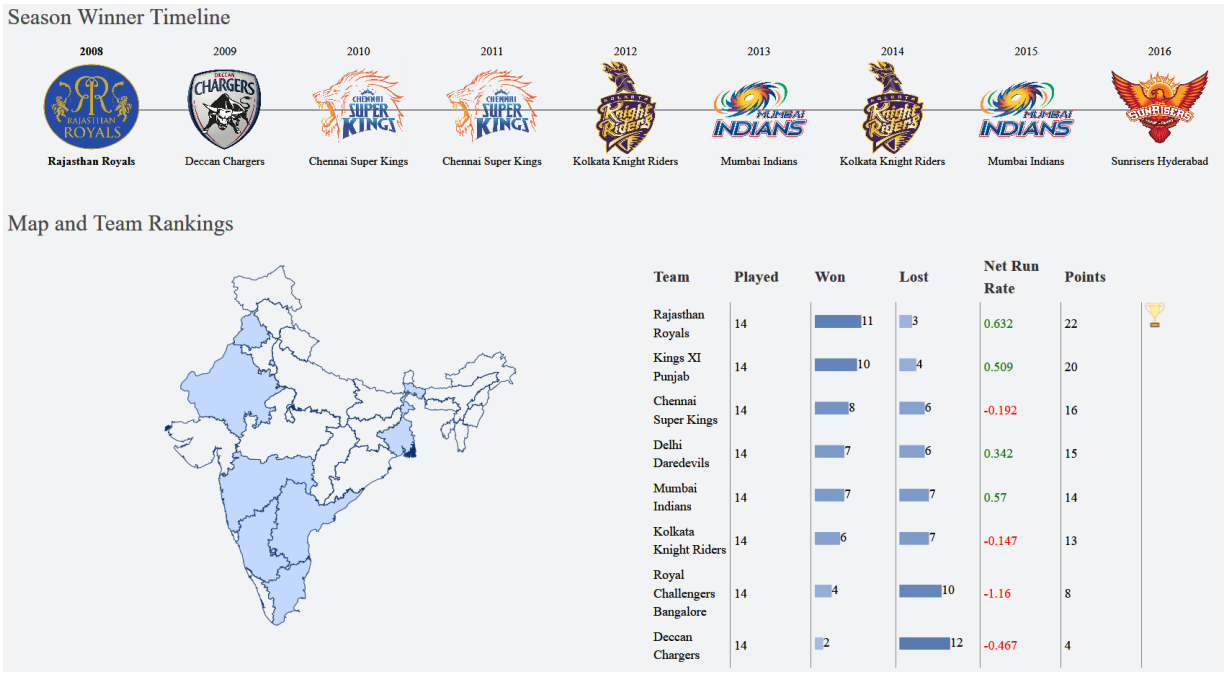



Fig 14. High level overview of season long statistics

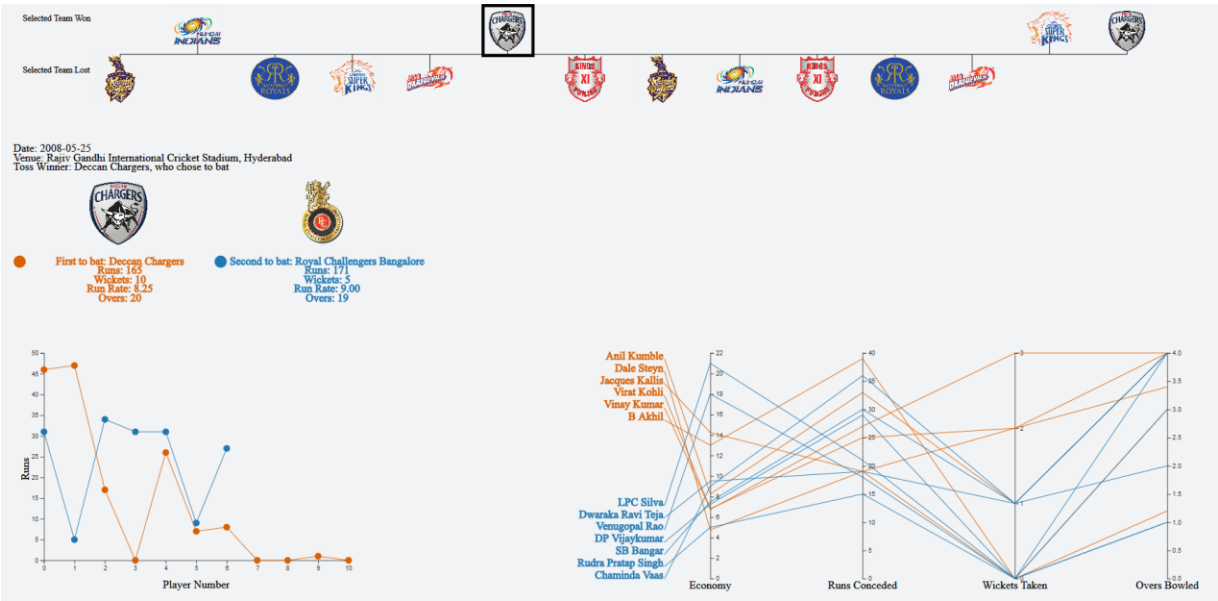



Fig. 15 More in depth statistics which show game details, including player statistics over the course of a game.

The visualization tools that we created were especially helpful teaching us how games tend to progress, as well as how teams can consistently win. There are a few major teams in the IPL which consistently win the title, a trend that is typical of most sports leagues. 4 out of the 5 teams that won championships are repeat winners (the 2009 and 2016 champions are the same team, which was renamed). The top scoring teams didn't always take the trophy either, and the winner could even be a fourth ranked team! Teams tend to start off their games with their best batsmen, scoring many of their runs in their first few players. Later on in the game, less and less competent batsmen get to bat, typically resulting in much fewer runs for the batting team near the end of the half.

There is always room for improvement with visualization, as user feedback can be put to use making a better product. Further feedback sessions would greatly help in improving the usability and feature set of our visualization, allowing us to better address what the user wants to see. One of the more requested features was comparing teams. While our visualization does allow the user to compare teams that competed in a game, as well as their home support, it doesn't allow for two disparate teams to be compared directly. In addition, transitions are a great way to increase the appeal of interactivity in a visualization, and they are certainly worth adding in future iterations.

## APPENDIX

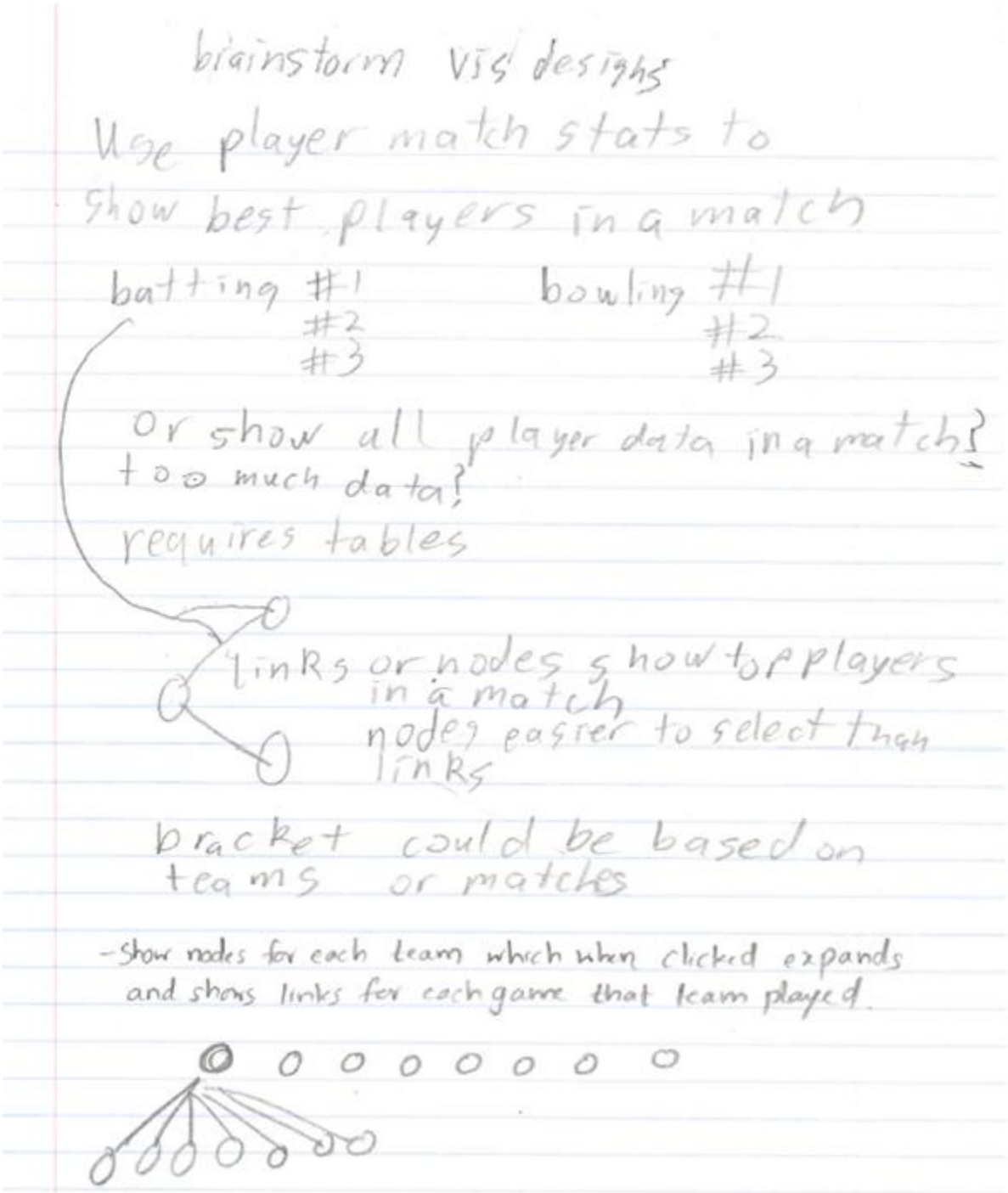



Figure 1. Brainstorm of individual visualization designs.

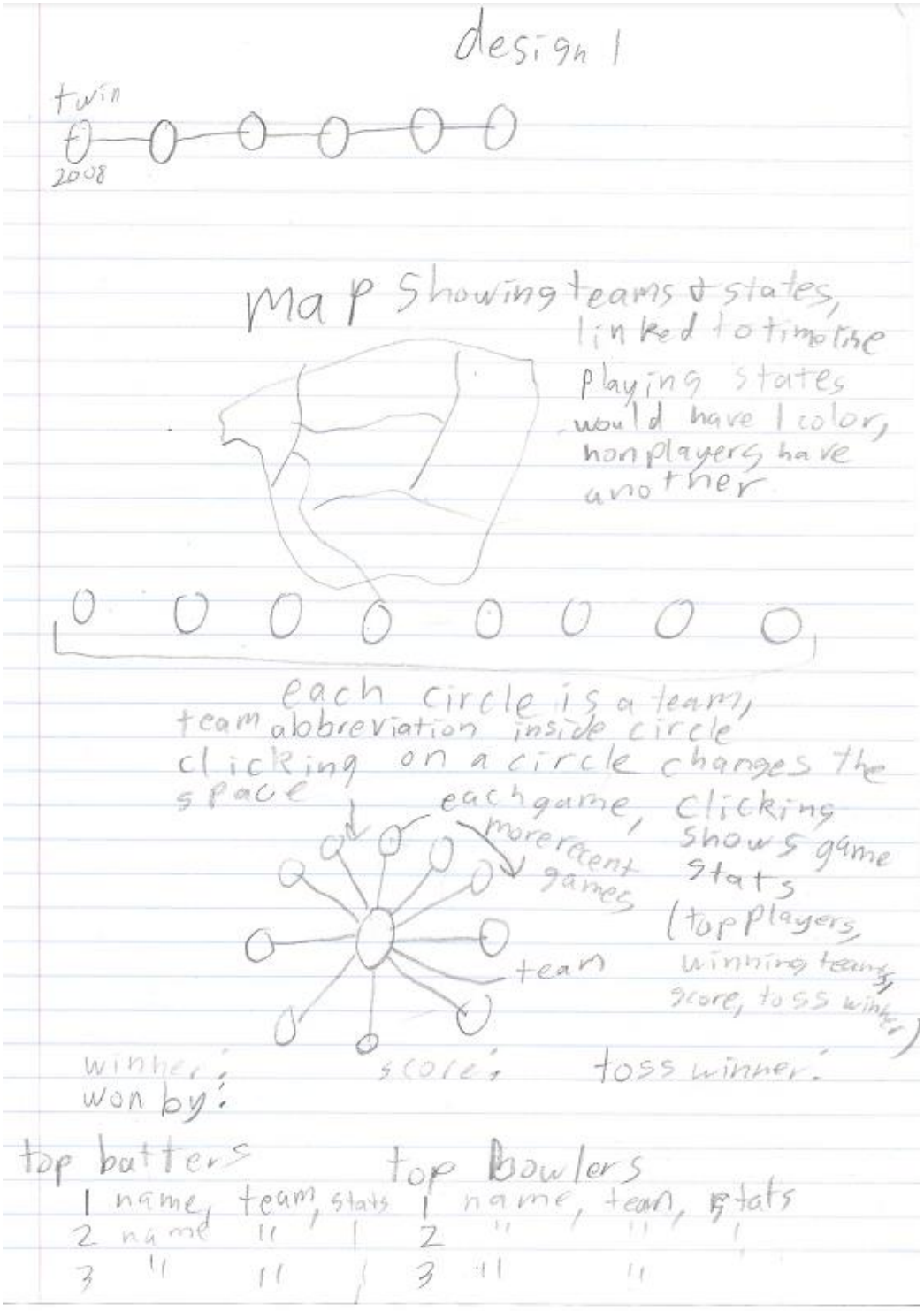



Figure 2. Potential design 1, page 1.

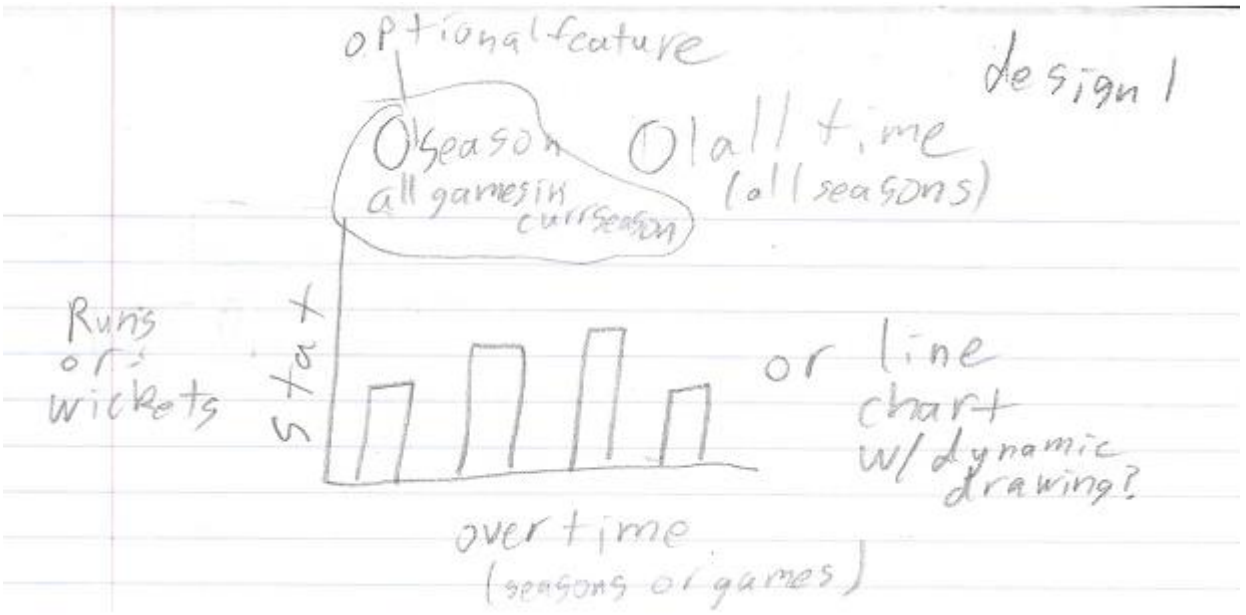



Figure 3. Potential design 1, page 2.

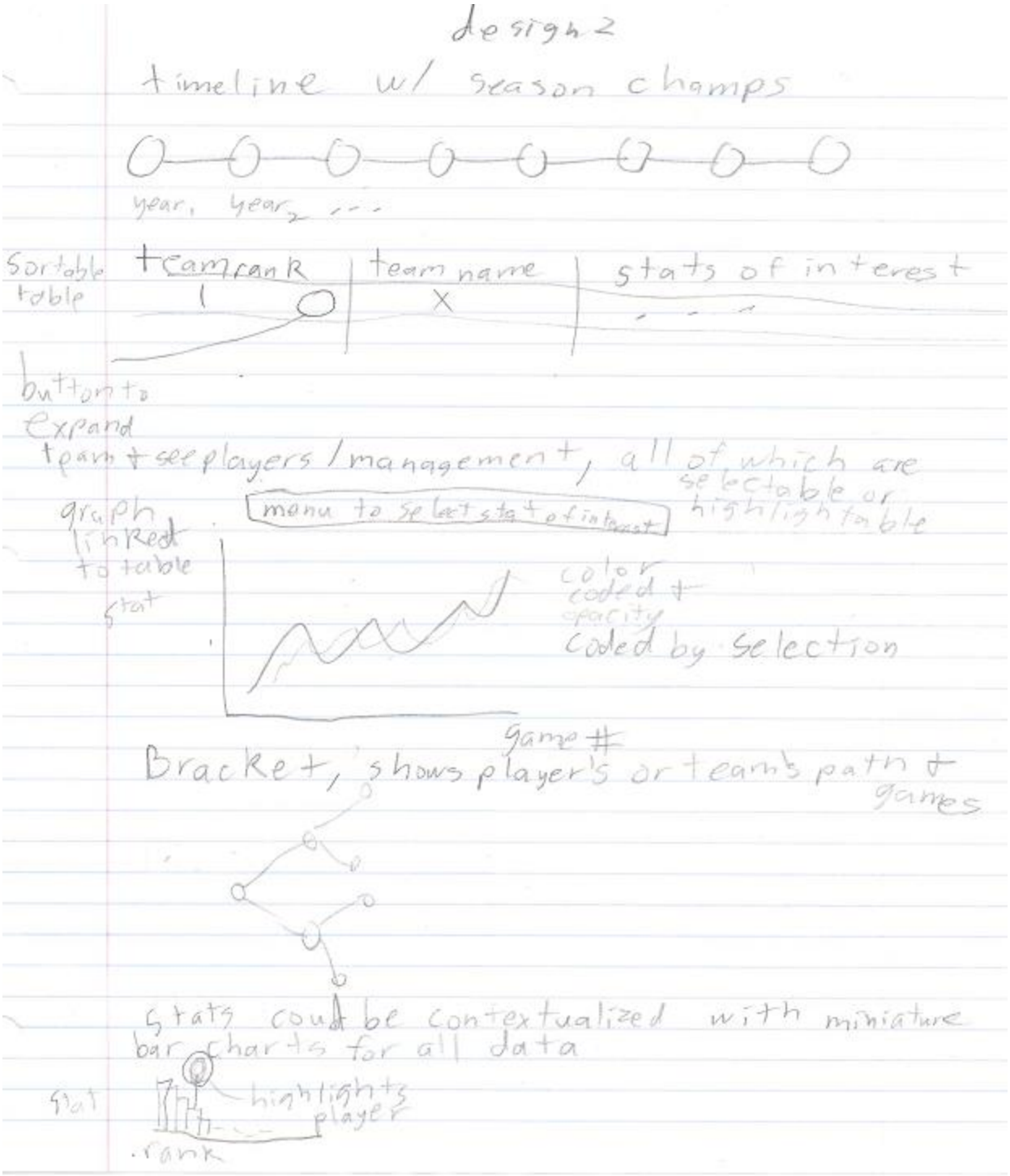



Figure 4. Potential design 2.

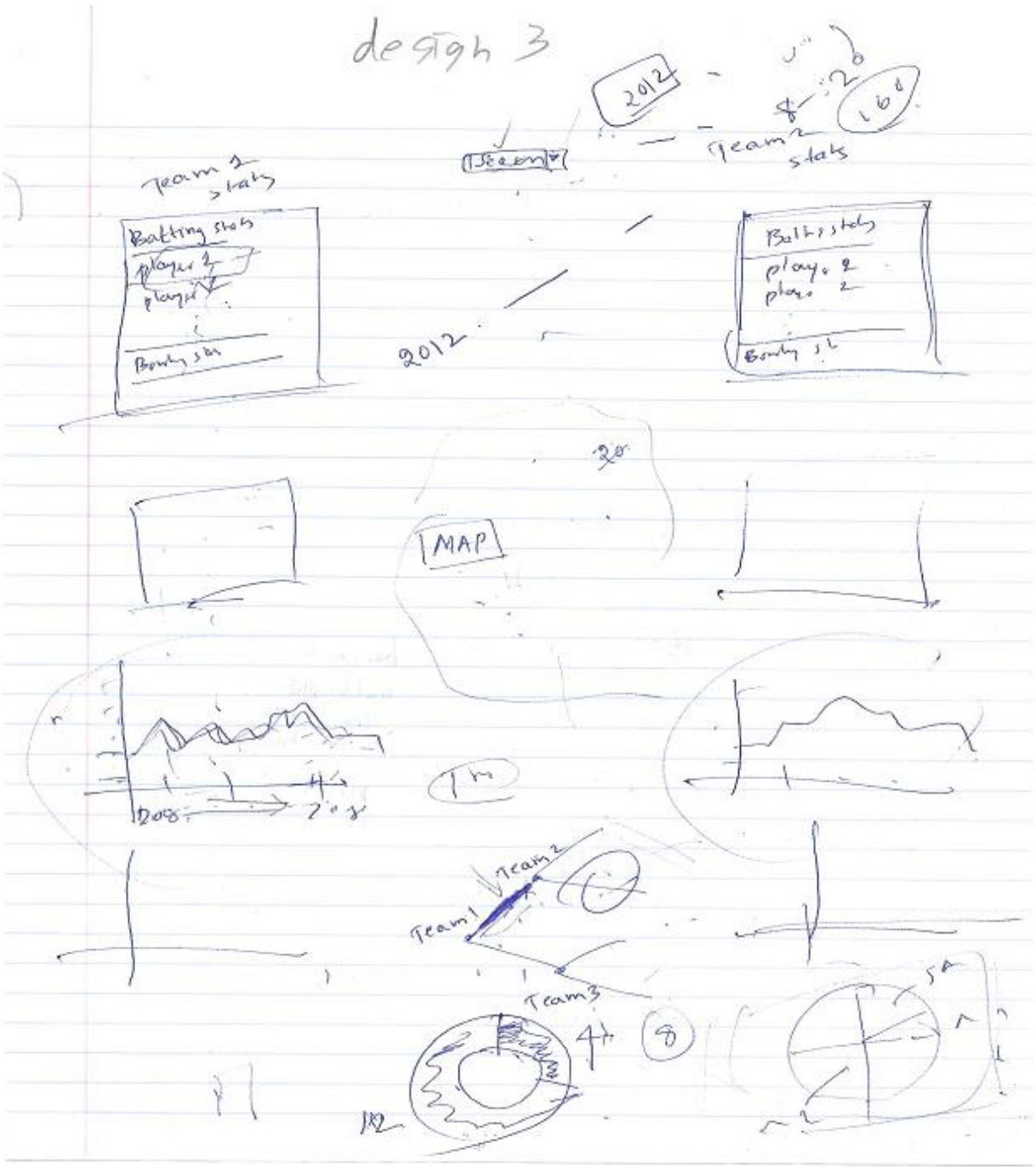



Figure 5 Potential design 3.

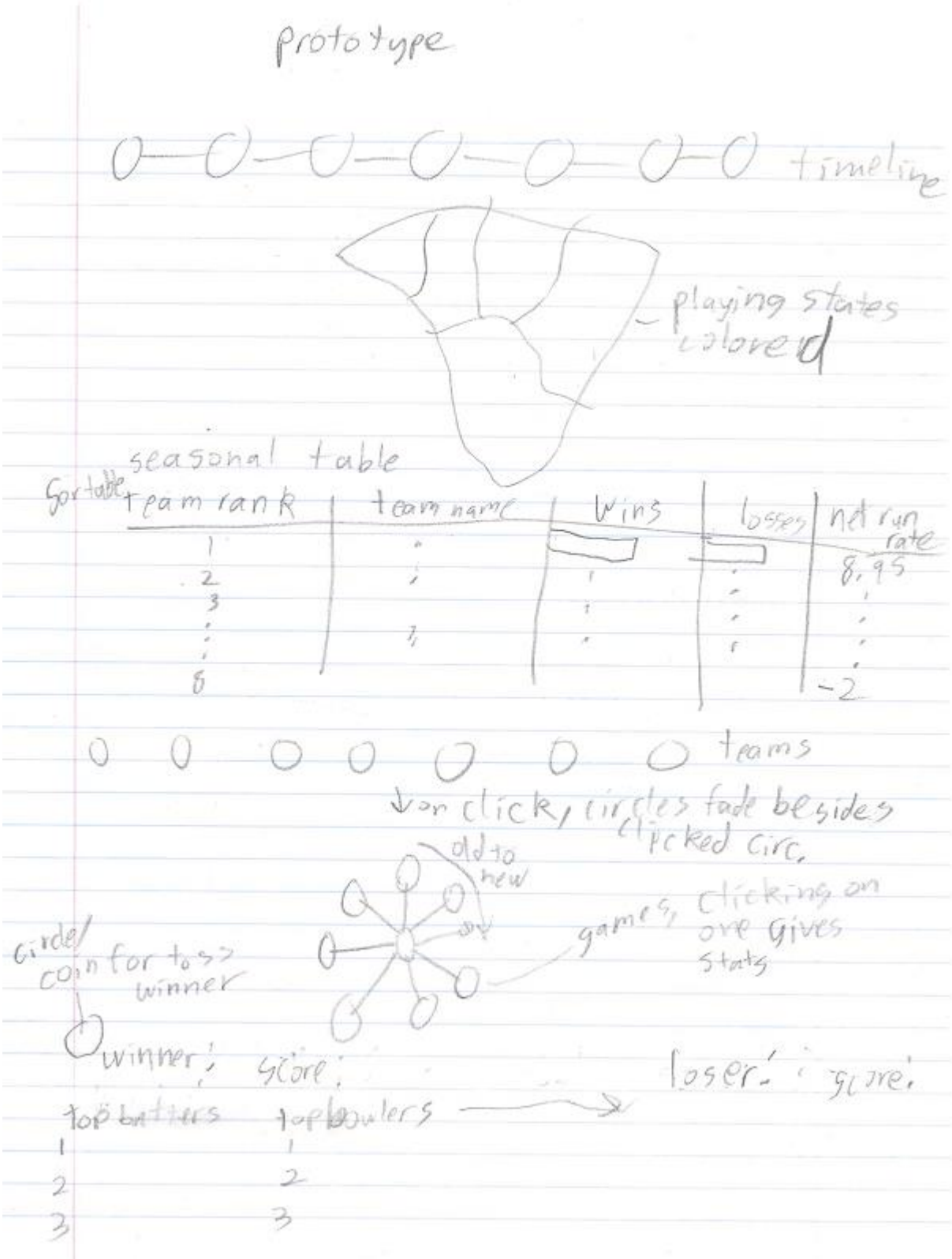



Figure 6. Prototype design, page 1.

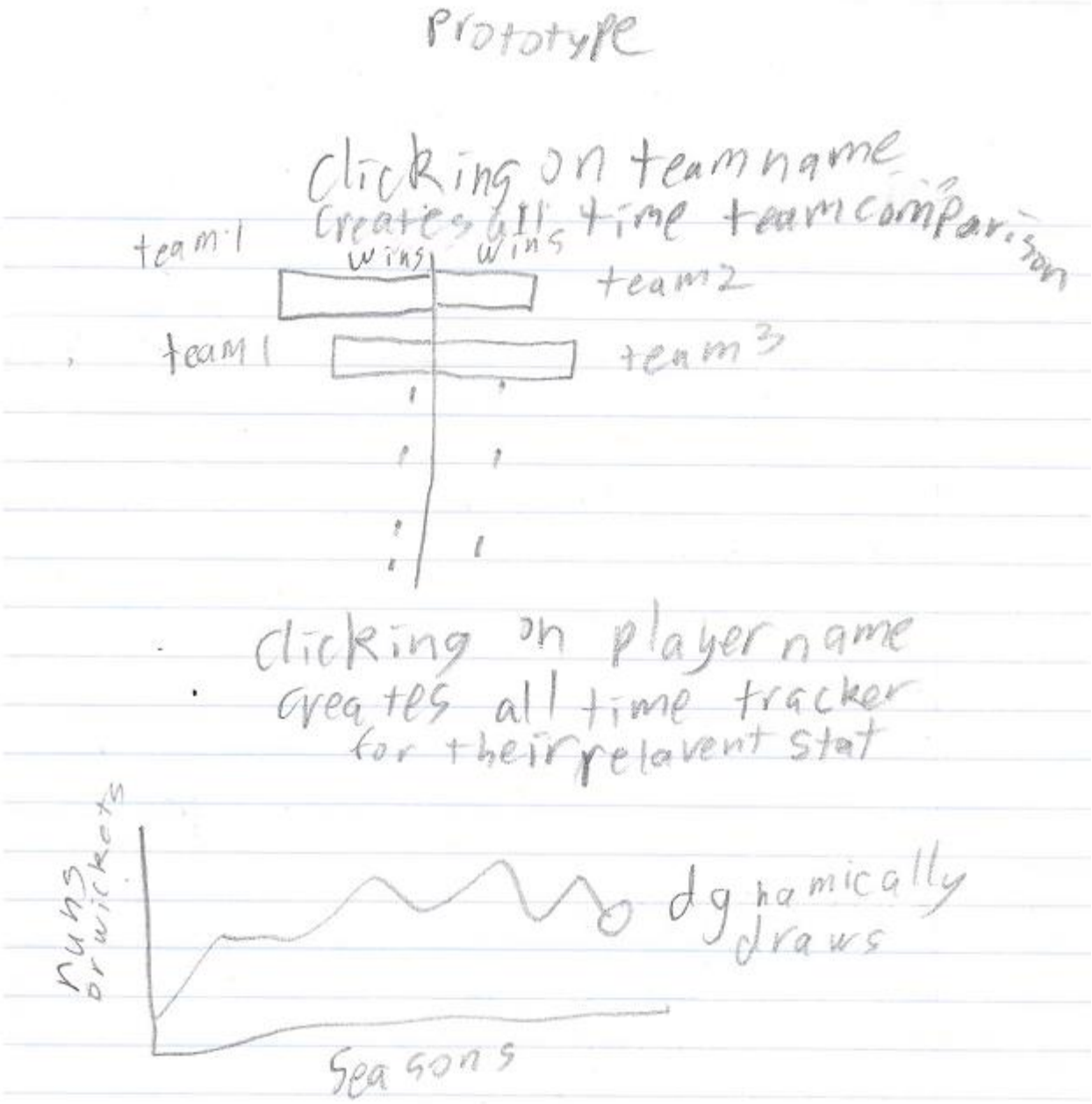



Figure 7. Prototype design, page 2.


Figure 8. Final timeline implementation.

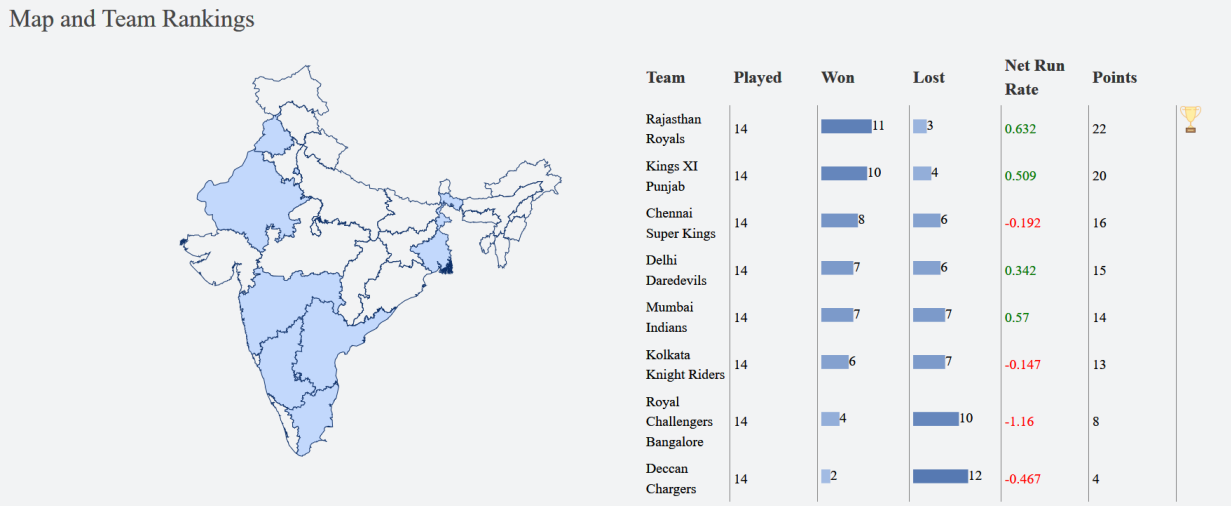



Figure 9. Final table and map implementation.


Figure 10. Final team selector implementation.

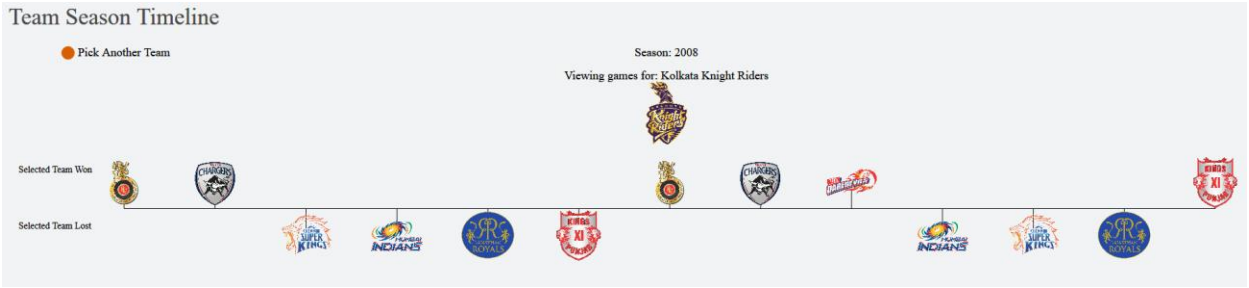



Figure 11. Final team season timeline selection.

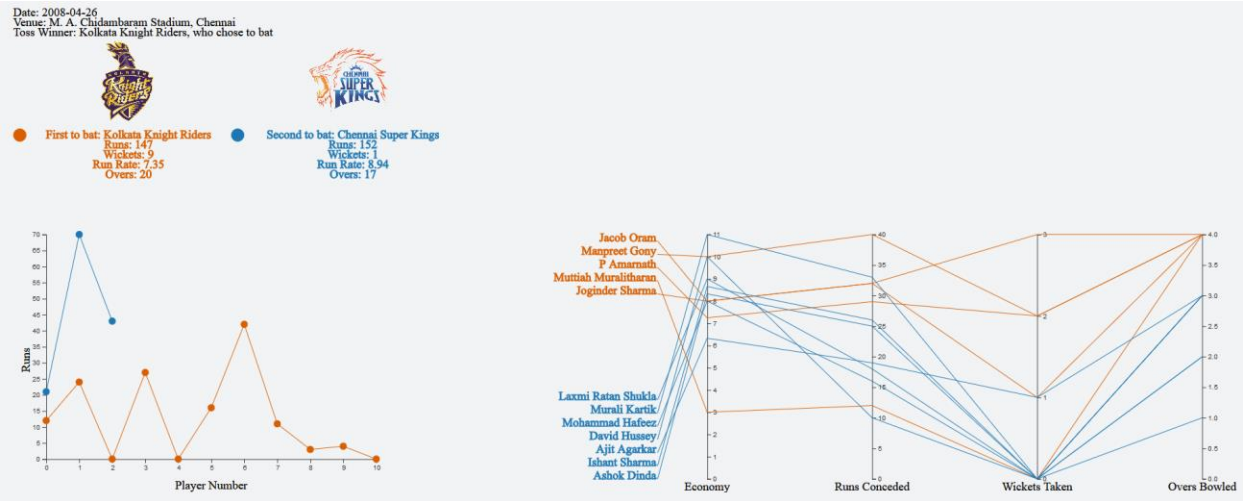



Figure 12. Final game view.

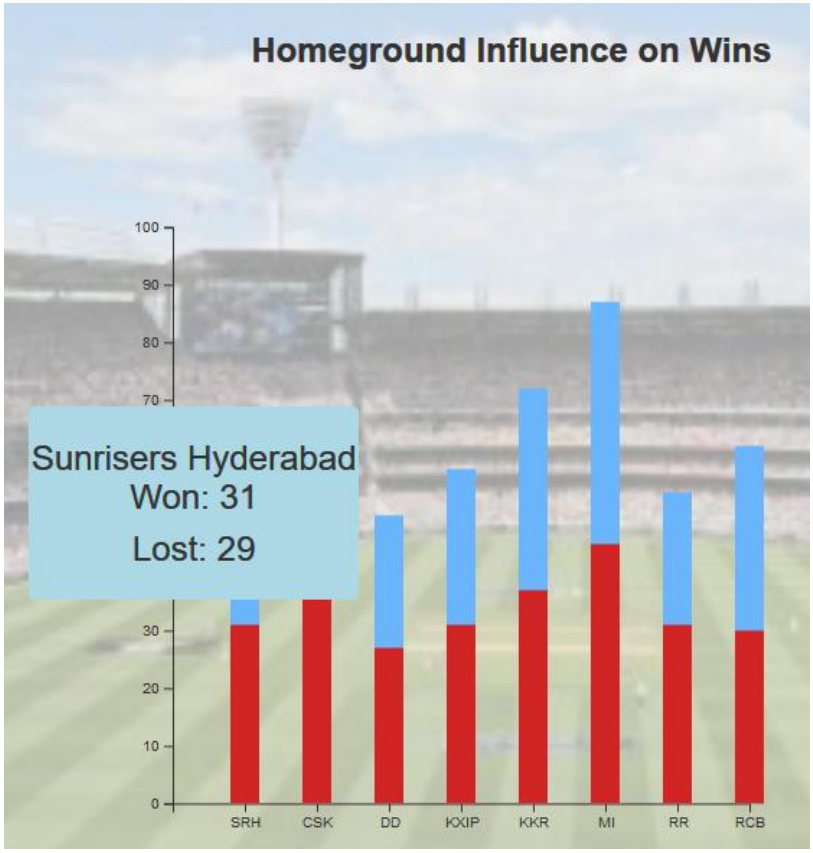



Figure 13. Final home support view demonstrating the included tooltip.

*THANK YOU !*