**Business Task -** (ASK)

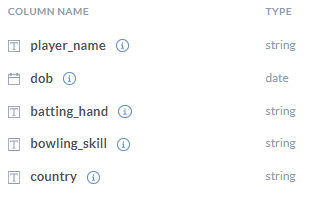
To analyse players and matches data of “IPL (Indian Premiere League) – Cricket Tournament conducted annually by BCCI” to choose the best 11 players for the dream team of all time.

**Dataset -** (PREPARE)

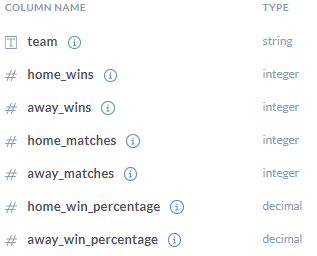
Data for this analysis has been taken from kaggle. <https://www.kaggle.com/ramjidoolla/ipl-data-set>.

Files in the dataset (**IPL Capstone**) –

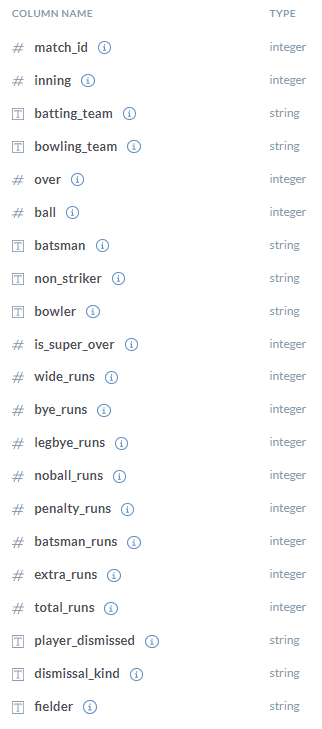
* Players.xlsx



* teamwise\_home\_and\_away.csv



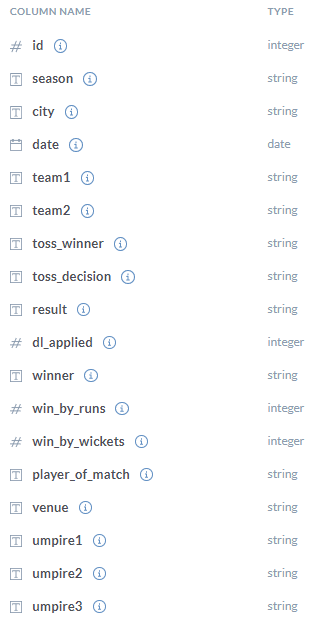
* deliveries.csv



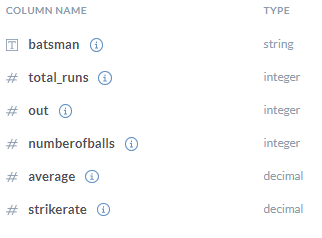
* Teams.csv



* matches.csv



* most\_runs\_average\_strikerate.csv



I sorted the data in **matches.csv** file with respect to **date** column in **ascending** order using Google Sheets.

**Data Cleaning** (PROCESS)

**Deliveries.csv**

Deliveries.csv file has **23** duplicate rows which were removed through Google Sheets.

Select the data in the sheet.

Go to Data in the menu bar and then select Remove Duplicates option.

There were **96** observations for which value of **inning** column was greater than 2. T20 cricket match normally consists of 2 innings. More than 2 innings means that result after two innings resulted in a tie between the two teams, so a super over is bowled by each side to decide the winner. Since, these super over stats doesn’t gets added to player career stats, so I decided to filter out all those observations where value of **innings** columns was greater than 2.

Since I filtered out all rows which were a part of superover, I also removed **superover** column as well.

In the **batting\_team** and **bowling\_team** column, there were two values(Rising Pune Supergiant and Rising Pune Supergiants) for the same team – **Rising Pune Supergiants**.

So with the help of Find and Replace, I changed the value in both columns from Rising Pune Supergiant to Rising Pune Supergiants.

**Note** – In all the remaining 5 csv sheets, we will do the same process of changing the value of Rising Pune Supergiant to Rising Pune Supergiants.

**Matches.csv**

There were 6 observations in which data was missing in **city** column. I entered the respective city names for those observations by looking at the **venue** column in the same dataset.

Changed the value Rising Pune Supergiant to Rising Pune Supergiants in columns **team\_1**, **team\_2**, **toss\_winner**, **winner.**

**venue** column contained slightly different values for the same stadium names. So, I changed them respectively.

* M. Chinnaswamy Stadium - M Chinnaswamy Stadium
* M.A. Chidambaram Stadium - MA Chidambaram Stadium, Chepauk
* Feroz Shah Kotla Ground - Feroz Shah Kotla
* Punjab Cricket Association IS Bindra

Stadium, Mohali - Punjab Cricket Association Stadium, Mohali

* Rajiv Gandhi International

Stadium, Uppal - Rajiv Gandhi Intl. Cricket Stadium

* Dr. Y.S. Rajasekhara Reddy

ACA-VDCA Cricket Stadium - ACA-VDCA Stadium

**most\_runs\_average\_strikerate.csv**

There were **34** batsmen whose **average** was missing. This data is not just missing at random. Infact, these batsmen didn’t got out for even once andBatting **average** of a batsmen is calculated only **after** he **gets out at least once**. If he is debuting and is unbeaten his batting **average will** be undefined. So, I removed these 34 observations from the dataset.

I also round off the columns **average** and **strikerate** to two decimal points.

**Players.csv**

I filled **95** missing entries in the **Country** column .

Zimbabwe was misspelt in two rows in Country column so I corrected it.

**teams.csv**

Deleted the row in which **team1** column value was Rising Pune Supergiant.

**teamwise\_home\_and\_away.csv**

Changed the value in **team** column from Rising Pune Supergiant to Rising Pune Supergiants.

I also changed the format of **home\_win\_percentage** and **away\_win\_percentage** columns from **Number** to **Percent**

**ANALYSE**

**deliveries.csv**

Created a column **runs\_conceded\_by\_bowler**

**runs\_conceded\_by\_bowler** = **total\_runs** – **bye\_runs** – **legbye\_runs**

(Runs scored in a bye or legbye delivery doesn’t gets added to bowler stats)

Created a column **is\_legal\_delivery**

**is\_legal\_delivery** = **IF(J2+M2=0,1,0)** …………..

(Logic – if wide\_runs(J2) and noball\_runs(M2) scored are 0 in a delivery, then it is a legal delivery, otherwise it isn’t.)

This column is created because I wanted to find no of overs bowled by each bowler through **total\_deliveries**/ 6(no of balls in a over). If I had calculated no balls and wide balls in the total\_deliveries, then my answer would have been incorrect for the **total\_overs\_bowled.**

**Note – The above mentioned two columns(total\_deliveries, total\_overs\_bowled) are created in new csv file Bowling\_Stats.csv.**

Created a column **bowler\_wickets**. Assigned the value of **0** if **dismissal\_kind** was any of the following : [**obstructing\_the\_field**, **retired\_hurt**, **run\_out**] or if the dismissal\_kind was **blank**(no wicket was taken on those deliveries). In rest of the cases, I assigned the value **1**.

Note – If a player gets out due to obstructing the field, retired hurt or run\_out, then these wickets are not considered in the bowler stats.

**Bowling\_Stats.csv**

Created a column **bowler** by copying the column **Player\_Name** from **Players.csv** file.

Created a **column runs\_conceded** to calculate the no of runs conceded by each bowler over his IPL career (Data till IPL 2019).

For example – runs conceded by A Ashish Reddy (A3) can be calculated as

= SUMIF(deliveries!I2:I178960,A3,deliveries!R2:R178960)

Created a column **total\_deliveries** to calculate the no of legal deliveries bowled by each bowler over his IPL career (Data till IPL 2019).

For example – total deliveries bowled by A Ashish Reddy (A3) can be calculated as

= SUMIF(deliveries!I2:I178960,A3,deliveries!S2:S178960)

Created a column **total\_overs\_bowled** to calculate the total no of overs bowled by each bowler over his IPL career (Data till IPL 2019).

**total\_overs\_bowled** =  **total\_deliveries** / **6** (Since, each over consists of 6 deliveries)

I also round off the values to the nearest integer.

I filtered out all those observations(**161** rows) from the data where value of **total\_deliveries** was **0**. This means that they were proper batsmen and have not bowled even a single ball in their IPL career, so it was better to filter out those rows.

Created column **economy(runs\_per\_over )** to calculate bowler’s economy.

**economy(runs\_per\_over ) = runs\_conceded** / **total\_overs\_bowled**

I also round off the value to nearest two decimal places.

Created a column **total\_wickets** to calculate total wickets taken by a bowler over his IPL career (Data till IPL 2019)

For example – total wickets taken by A Ashish Reddy (A3) can be calculated as

= SUMIF(deliveries!I2:I178960,A3,deliveries!V2:V178960)

Filtered out all rows where **total\_deliveries** bowled were **less than 250**.

Created column **average\_runs\_conceded\_per\_wicket**

**average\_runs\_conceded\_per\_wicket** **= runs\_conceded** / **total\_wickets**

Created column **average\_balls\_bowled\_per\_wicket**

**average\_balls\_bowled\_per\_wicket** = **total\_deliveries** / **total\_wickets**

Rounded off columns average\_runs\_conceded\_per\_wicket & average\_balls\_bowled\_per\_wicket to nearest two decimal places

Created column **combined\_bowling\_rate** to get a combined score of economy(**E**), average\_runs\_conceded\_per\_wicket(**G**) and average\_balls\_bowled\_per\_wicket(**H**)

**combined\_bowling\_rate** = **3 / ( (1/E) + (1/G) + (1/H) )**

Rounded off the combined\_bowling\_rate column to nearest two decimal places.

Sorted the dataset in increasing order of combined\_bowling\_rate

**most\_runs\_average\_strikerate.csv**

I filtered out all the rows where **total\_runs** scored by batsmen were **less than 1000**.

I created a column **BASRA** (Batting Average & Strike Rate) –

**BASRA** = **average** + **strikerate**

Then I sorted the data in decreasing order of BASRA

**matches.csv**

Created column **bat\_first\_win** by assigning value 0 if **win\_by\_runs** column value is 0, else 1.

Summed this column to calculate the total no of wins by teams batting first.

Calculated **Bat\_First\_Win\_Percent**

Created column **chase\_win** by assigning value 0 if **win\_by\_wickets** column value is 0, else 1.

Summed this column to calculate the total no of wins by teams who chased the target.

Calculated **Chase\_Win\_Percent**

**Players.csv**

Created a column **Overseas\_Player** and assigned the value “**No**” if value in Country column is India and “**Yes**” for rest cases.

**Reason** – We wanted to create an all time best IPL team of 11 players. Since, each team cannot name more than 4 overseas players in their starting 11, this column is needed.

**IPL Rule** - [IPL Laws - law-1-the-players](https://www.iplt20.com/about/match-playing-conditions/law-1-the-players)

**teamwise\_home\_and\_away.csv**

Created column **overall\_win\_percentage** by taking average of **home\_win\_percentage** and **away\_win\_percentage** columns.

Rounded off to nearest two decimal places.

**SQL**

**Query to get no of wins by teams batting first (venue wise)**

SELECT   venue, COUNT(\*)  no\_of\_wins

FROM `skilful-deck-318502.ipl.matches`

WHERE  toss\_decision  =  'bat'

     AND  toss\_winner  =  winner

GROUP  BY  venue;

I saved the output of the above query into a local file **venue\_wise\_batting\_first\_win.csv**

**Query to get no of wins by teams fielding first (venue wise)**

SELECT  venue, COUNT(\*)  no\_of\_wins

FROM  `skilful-deck-318502.ipl.matches`

WHERE  toss\_decision  =  'field'

     AND  toss\_winner  =  winner

GROUP  BY  venue;

I saved the output of the above query into a local file **venue\_wise\_fielding\_first\_win.csv**

**Query to get total no of matches where toss decision was batting (venue wise)**

SELECT  venue, COUNT(\*)  total\_matches

FROM  `skilful-deck-318502.ipl.matches`

WHERE toss\_decision = 'bat'

AND winner  IS  NOT  NULL

GROUP  BY  venue;

I saved the output of the above query into a local file **venue\_wise\_bat\_first\_total\_matches.csv**

**Query to get total no of matches where toss decision was fielding (venue wise)**

SELECT  venue, COUNT(\*)  total\_matches

FROM  `skilful-deck-318502.ipl.matches`

WHERE toss\_decision = 'field'

AND winner  IS  NOT  NULL

GROUP  BY  venue;

I saved the output of the above query into a local file **venue\_wise\_field\_first\_total\_matches.csv**

Uploaded all the above four files into IPL Capstone(Google Sheet where I was working so far)

**venue\_wise\_bat\_first\_total\_matches.csv**

Created column **matches\_won** to calculate no of matches won when toss decision was batting

For Example – We want to calculate matches won in ACA-VDCA Stadium (A2)

=VLOOKUP(A2,venue\_wise\_batting\_first\_win!A2:B34,2,FALSE)

After applying the formua, I pasted only the value into the cells using **Paste Special**

Sorted the data in ascending order of venue

**venue\_wise\_field\_first\_total\_matches.csv**

Created column **matches\_won** to calculate no of matches won when toss decision was fielding

For Example – We want to calculate matches won in ACA-VDCA Stadium (A2)

=VLOOKUP(A2,venue\_wise\_fielding\_first\_win!A2:B34,2,FALSE)

After applying the formua, I pasted only the value into the cells using **Paste Special**

Sorted the data in ascending order of venue

**venue\_wise\_total\_matches.csv**

Copied sheet **venue\_wise\_bat\_first\_total\_matches.csv** into **venue\_wise\_total\_matches.csv**.

Renamed columns **total\_matches** and **matches\_won** to **total\_matches\_bat\_first** & **matches\_won\_bat\_first** respectively**.**

Copied columns **total\_matches** and **matches\_won** from **venue\_wise\_total\_matches.csv** to **venue\_wise\_total\_matches.csv** and renamed these columns to **total\_matches\_field\_first** & **matches\_won\_field\_first** respectively.

Created column **total\_matches** by adding total\_matches\_bat\_first & total\_matches\_field\_first

Created column **matches\_won** by adding matches\_won\_bat\_first & matches\_won\_field\_first

Filtered out the data where **total\_matches were less than 10**.

Created **column Correct\_toss\_decision\_percent** to calculate what percentage of total matches played in each venue resulted in toss\_winner being the match\_winner team. Converted datatype to percent.

Sorted the data in descending order or Correct\_toss\_decision\_percent.

**Top\_20\_Batsmen.csv**

Copied the first 21 rows(including header) from **most\_runs\_average\_strikerate.csv**

Created a column **overseas\_player** through **Players.csv** using **Find and Replace** functionality.

**Top\_20\_Bowlers.csv**

Copied the first 21 rows(including header) from **Bowling\_Stats.csv**

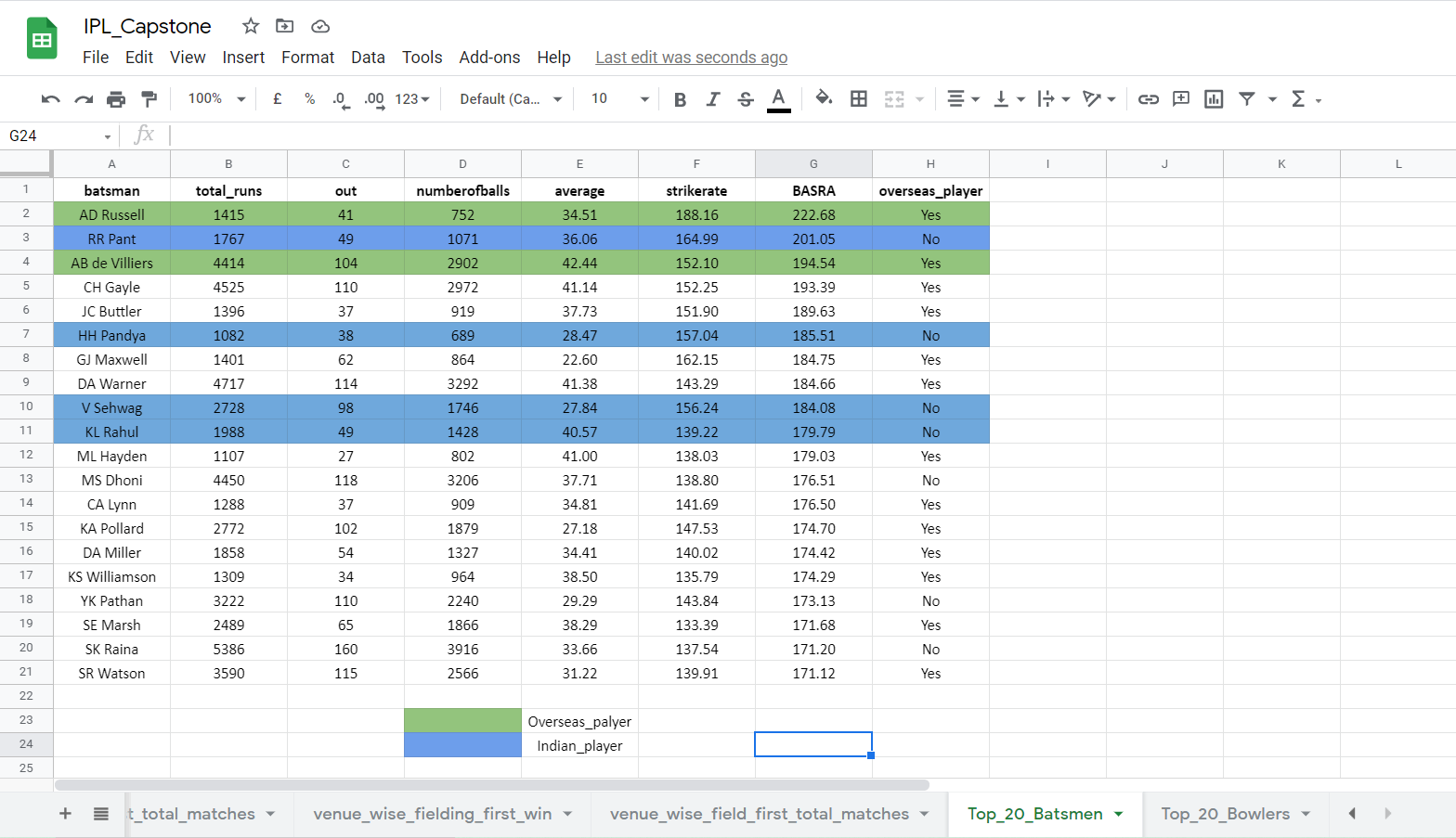
Created a column **overseas\_player** through **Players.csv** using **Find and Replace** functionality.

**Deciding Best 11**

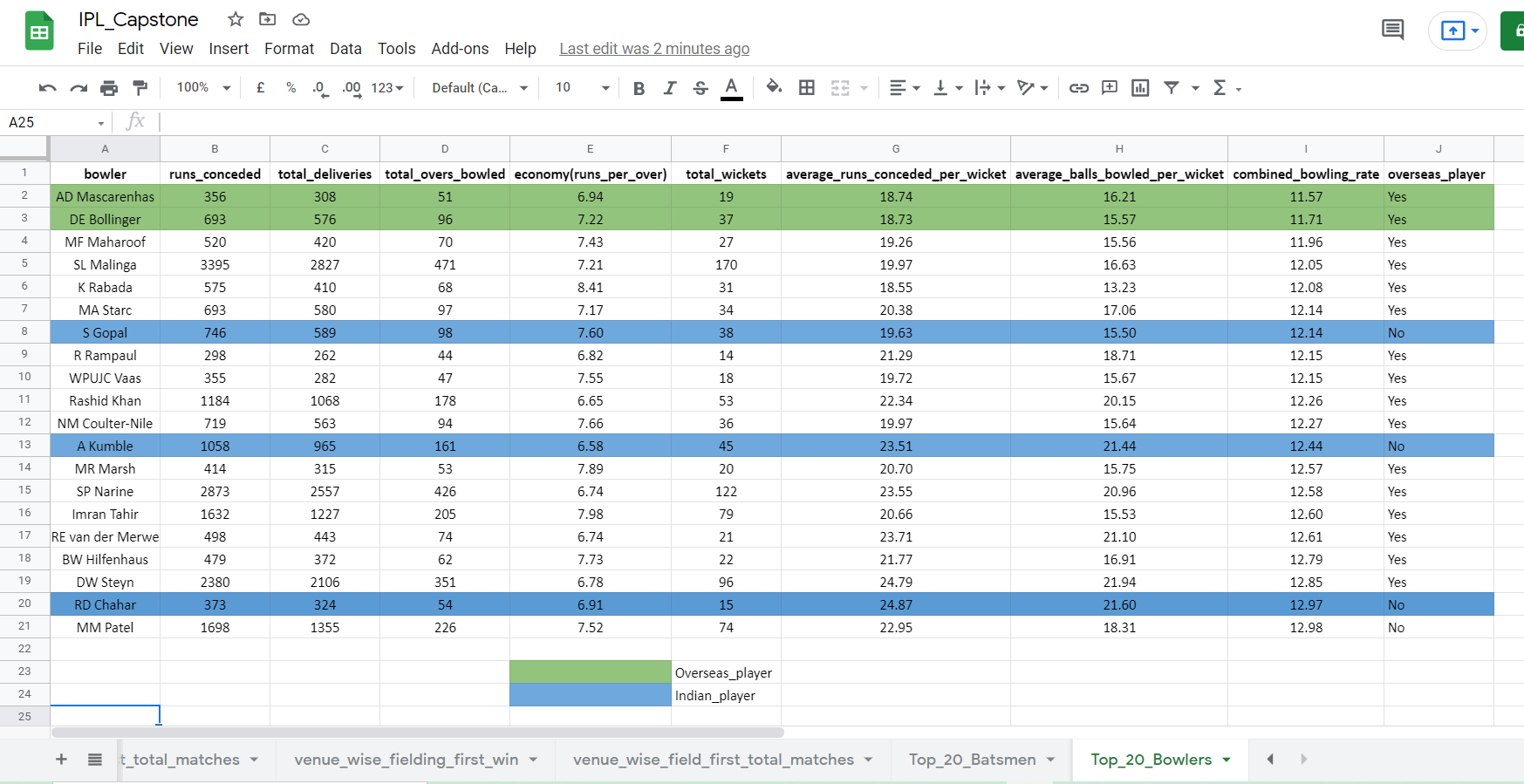
Since we now have separate list of top 20 batsmen and top 20 bowlers according to the metrics defined by us, we can choose the best 11 players of all time in IPL ( Data till IPL 2019).

I decided to choose **6 batsmen** and **5 bowlers**. We also have to keep in mind that a maximum of **4** **foreign players** can play in an IPL team. So, I decided to choose **top 2** foreign players from **Top\_20\_Batsmen** sheet and **top 2** foreign players from **Top\_20\_Bowlers** sheet.

We now have to choose **top 4** (6-2) **Indian batsmen** from Top\_20\_Batsmen sheet and **top 3** (5-2) **Indian bowlers** from Top\_20\_Bowlers sheet.



**6 players selected from Top\_20\_Batsmen**



**5 players selected from Top\_20\_Bowlers**

**SHARE**

I created a dashboard in Tableau Desktop for the All time Best 11 team of IPL (Data till IPL 2019).

Here is a link to my dashboard - [IPL\_Best\_11 - Tableau Dashboard](https://public.tableau.com/views/IPL_Best_11/Dashboard1?:language=en-US&:display_count=n&:origin=viz_share_link)

**Conclusion**

* **Rising Pune Supergiants** has the highest overall win percentage followed by Chennai Super Kings. ( Reference - teamwise\_home\_and\_away)

**Note -** This does not imply that these two teams were the most successful in IPL (Since, the success in IPL is normally described as teams having maximum IPL winner trophies) .There may be a case that these teams have won initial tournament matches more than other teams, but have performed not that well in the knockout matches leading to their elimination

* There were in total **756** matches played during the 12 years of IPL (IPL 2008 to IPL 2019). **44.58%** of those matches were won by teams batting first whereas teams chasing the target were slightly more successful with a figure of **53.70%.** ( Reference - matches)
* **Nehru Stadium** and **SuperSport Park** venues were the most successful venues **(66.67%)** in terms of toss winning teams being the match winning teams. (Lets say there were 100 matches played in Nehru Stadium, then 66.67 ~ 67 matches resulted in toss wining team to be the match winning team. Similar was the case with SuperSport Park) (Reference – venue\_wise\_total\_matches.csv)
* Overseas players attained the majority of top rankings in both batting and bowling. (In the top 10 batsmen, there were only 3 Indian players. In the top 10 bowlers, there were 9 overseas players as compared to only 1 Indian player)

Link to my Tableau dashboard - [IPL\_Best\_11 - Tableau Dashboard](https://public.tableau.com/views/IPL_Best_11/Dashboard1?:language=en-US&:display_count=n&:origin=viz_share_link)

**NOTE** - As per IPL rules of making a team, I have to exclude some top overseas players ( both batsmen and bowlers) as a maximum of 4 overseas players can be in the IPL team. Refer to section 1.2.5 in Law 1 of IPL ([Law-1-the-players](https://www.iplt20.com/about/match-playing-conditions/law-1-the-players))

**References**

**Criteria used for determining best batsmen** [best-batsman-ipl](https://cricket.yahoo.net/news/best-middle-order-batsman-ipl-163429366)

**Criteria used for determining best bowler** [measure\_of\_bowling\_performance](https://www.researchgate.net/publication/272471255_The_combined_bowling_rate_as_a_measure_of_bowling_performance_in_cricket)