NAME- AKASH DAS
ROLL NUMBER- 401
SUPERVISOR – PROF. DURBA BHATTACHARYA

TITLE- APPLICATION OF STATISTICS IN CRICKET DATA ANALYTICS

Session: 2019-2022

I affirm that I have identified all my sources and that no part of my dissertation paper uses unacknowledged materials.

TABLE OF CONTENTS

Introduction:

In this era, while many sports are played in different countries, a few have been liked and encouraged a little more than others. Similarly, cricket is one of the most picked and played sports of this modern era. Cricket was introduced in England in the sixteenth century. Initially, cricket was introduced and played in a test format only. After some time, and due to some conditions and policies, the international cricket council introduced additional formats such as the T20 and ODI format. There are three official formats in which cricket is played internationally with varying durations and standards. The one-day international (ODI) cricket is one of the most played and liked structures by everyone. In this format, 100 overs of play are designed. Given that there are 100 overs in a one-day game, each team plays 50 overs, with the aim to fight and win. The datasets are available on different electronic databases with the maximum available information.

The Role of Statistics in Cricket:

STATISTICS means the practice or science of collecting and analysing numerical data in large quantities .It is a branch of mathematics dealing with the collection, analysis, interpretation, and presentation of masses of numerical data.

Statistics have always had a significant role in sports. Sports analytics is on the rise and will continue to play a significant role in how teams operate, pick their players, how they play the game, etc.

Cricket is no different. The runs scored by a batsman, the wickets taken by a bowler, or the matches won by a cricket team – these are all examples of the most important numbers in the game of cricket. Maintaining a record of all such statistics has multiple benefits. The teams and the individual players can dig deep into this data and find areas of improvement. It can also be used to assess an opponent’s strengths and weaknesses.

This record gives us quite a lot of useful information, such as:

- The batting average in different formats
- Number of times the batsman scored >= 100 runs
- Highest scores of the batsman
- Number of matches played, etc.

An extensive exploratory analysis of this well-structured data can be helpful in comparing multiple players.

Think about it – we can try to answer the age-old questions like who's the all-time best batsman? Who is better batsman the legend SACHIN TENDULKAR or the modern great VIRAT KOHLI? And so on.

This analysis would be based on their individual records. Imagine the amount of fun we can have using this data!

However, there are many crucial insights that are difficult and cumbersome to obtain by using only traditional data analysis techniques. Since cricket is a team game, it involves interaction among the players within a team as well as with those in the opposition team.

It is quite difficult to win matches if individuals focus on their respective performances only. They have to support their fellow players as well.

In this study I will answer 3 question regarding cricket data analytics with the help of the statistics and using python programming language for calculation purpose. The questions are-

1. *SACHIN TENDULKAR vs. VIRAT KOHLI (IN ODI): Statistical comparison(batsman vs. batsman)*

2. *MS DHONI vs. DALE STEYN (IN IPL): Statistical comparison(batsman vs. bowler match-up)*
3. *Whether MS DHONI is still fit to bat or not? (analyzing the performance of a player)*

LET'S MOVE FORWARD TO THE 1ST QUESTION

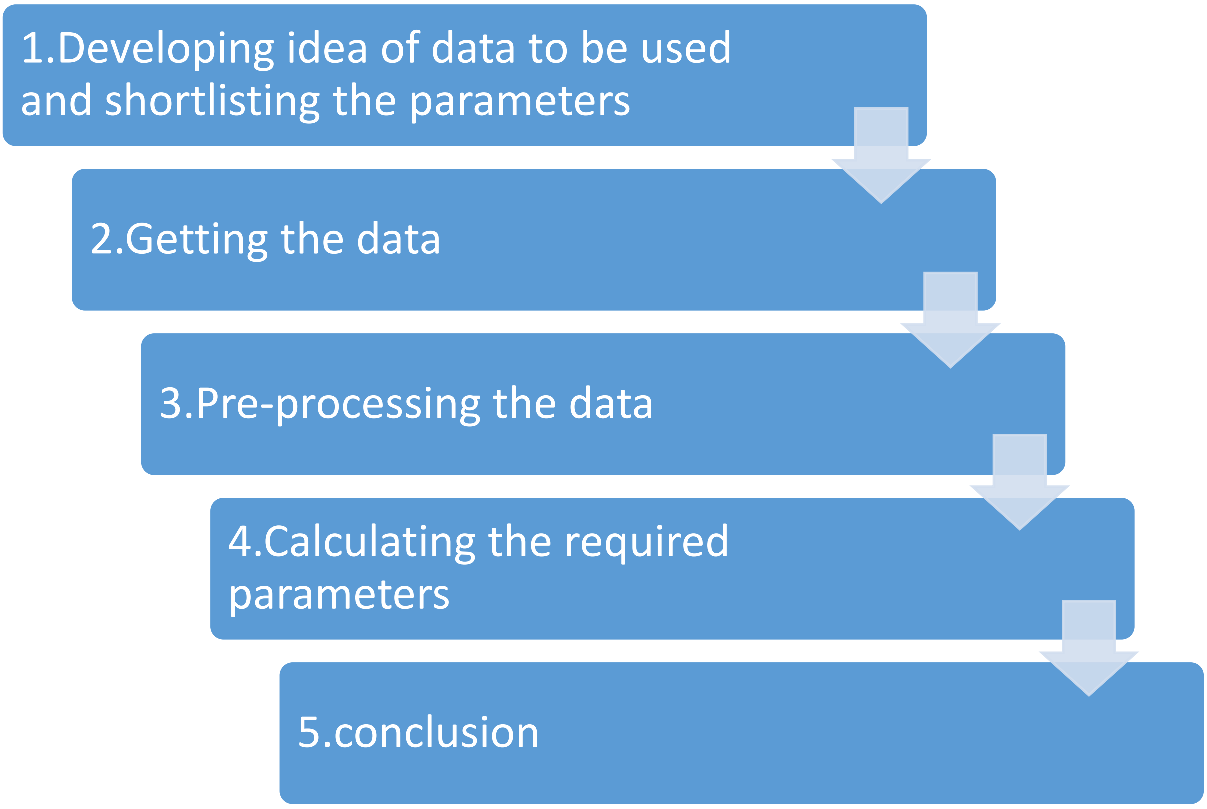
1. SACHIN TENDULKAR vs. VIRAT KOHLI (IN ODI): Statistical comparison(batsman vs. batsman)

It's clearly a batsman vs. batsman comparison of a same country (team). Of course, they from different generations. Sachin Tendulkar made his international debut on November 15, 1989 against Pakistan in Karachi and Virat Kohli made his international debut on 18 August, 2008 against Sri Lanka in Dambulla.

Master Blaster Sachin Tendulkar remains the greatest batter of all time in international cricket. Known as the 'God of Cricket', Tendulkar has plenty of monumental records to his name. Even matching them seems far-fetched. Among active cricketers, former Indian captain Virat Kohli is deemed the successor of Tendulkar.

The debate SACHIN vs. VIRAT happens over a long period of time. The current generation has seen more Virat than Sachin so they think Virat is better on the other hand many people think Sachin is better than Virat. Let compare these two great batsman analytically using the data. Let us see how we can compare 2 batsman of 2 different generations.

APPROACH WE SHOULD TAKE TO COMPARE TWO BATSMAN OF DIFFERENT GENERATIONS:



1.Developing idea of data to be used and shortlisting the parameters

2.Getting the data

3.Pre-processing the data

4.Calculating the required parameters

5.conclusion

1.Developing idea of data to be used and shortlisting the parameters

SACHIN TENDULKAR

PLAYING YEARS: 1989-2012

FIRST CENTURY IN 1994

VIRAT KOHLI

PLAYING YEARS: 2008-PRESENT

FIRST CENTURY IN 2009

We will consider their data from the year they are getting their 1st ODI century. We will compare them w.r.t 10 years data. It means Sachin's data from 1994 to 2004 and Virat's data from 2009 to 2019. Observe that Virat's debut 2008 so his records is not involved in Sachin's data. Again we take Sachin's data till 2004 so it not involve in Virat's data, because it is from the year 2009.

SACHIN'S DATA – 1994 TO 2004 (10 YEARS)

VIRAT'S DATA -- 2009 TO 2019 (10 YEARS)

PARAMETERS WE WILL USE FOR COMPARISON:

- BATTING RUNS PER INNINGS (RPI)

It's get the idea about the number of runs scored by the batsman per match (innings) on an average.

# Runs per innings = Total Runs/Total Innings

- STRIKE RATE

It's give the idea about the number of runs scored by the batsman per 100 balls on an average. Percentage of strike rate indicates if the batsman 100 then how much he can score.

# SR = 100*(Total Runs/Total Balls)

- 100'S PER MATCH

It indicates the no. of match required to the batsman to score a century or 100. Note that if the value of 100'S PER MATCH of one batsman is less than the other then he is better.

#100's per match= total number of (100's) / Total Innings

- 50’S PER MATCH

It indicates the no. of match required to the batsman to score a half-century or 50. Note that if the value of 50’S PER MATCH of one batsman is less than the other then he is better.

#50’s per match= total number of (50's) / Total Innings

- TEAM CONTRIBUTION

It indicates how much the batsman contribute for his team

# Team contribution = Player Runs/Team Runs without the player

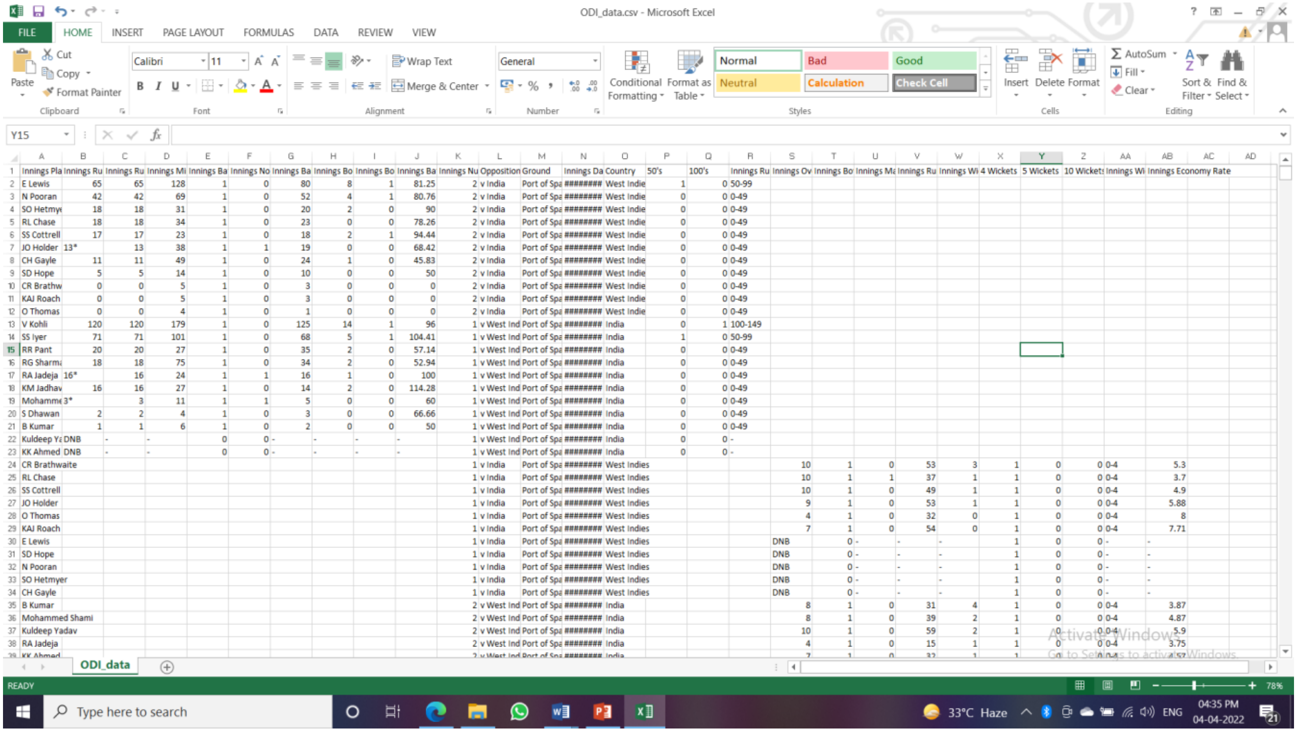
The percentage of team contribution shows that if the team scores total 100 runs then how much the batsman contributed.

[ex: suppose player “x” scored 50 runs in a match and his team scored 200 runs , then his team contribution for that match is = runs scored by “x” / total runs scored by his team without “x”

i.e. {50/ (200-50)}*100 = 33.33%]

2. Getting the data

For this study I collected the raw data from Statsguru, Cricinfo's searchable cricket statistics database (https://stats.espncricinfo.com/ci/engine/stats/index.html). A small glimpse of the raw data is given below.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	Innings Pla	Innings Ru	Innings Ru	Innings Mi	Innings Ba	Innings No	Innings Ba	Innings Bo	Innings Bo	Innings Ba	Innings Nu	Opposition	Ground	Innings Da	Country	50's	100's	Innings Ru	Innings Ov	Innings Bo	Innings Ma	Innings Ru	Innings Wi	4 Wickets	5 Wickets	10 Wickets	Innings Wi	Innings Economy Rate
2	E Lewis	65	65	128	1	0	80	8	1	81.25	2	v India	Port of Spa	########	West Indie	1	0	50-99										
3	N Pooran	42	42	69	1	0	52	4	1	80.76	2	v India	Port of Spa	########	West Indie	0	0	0-49										
4	SO Hetmye	18	18	31	1	0	20	2	0	90	2	v India	Port of Spa	########	West Indie	0	0	0-49										
5	RL Chase	18	18	34	1	0	23	0	0	78.26	2	v India	Port of Spa	########	West Indie	0	0	0-49										
6	SS Cottrell	17	17	23	1	0	18	2	1	94.44	2	v India	Port of Spa	########	West Indie	0	0	0-49										
7	JO Holder	13*	13	38	1	1	19	0	0	68.42	2	v India	Port of Spa	########	West Indie	0	0	0-49										
8	CH Gayle	11	11	49	1	0	24	1	0	45.83	2	v India	Port of Spa	########	West Indie	0	0	0-49										
9	SD Hope	5	5	14	1	0	10	0	0	50	2	v India	Port of Spa	########	West Indie	0	0	0-49										
10	CR Brathw	0	0	5	1	0	3	0	0	0	2	v India	Port of Spa	########	West Indie	0	0	0-49										
11	KAJ Roach	0	0	5	1	0	3	0	0	0	2	v India	Port of Spa	########	West Indie	0	0	0-49										
12	O Thomas	0	0	4	1	0	1	0	0	0	2	v India	Port of Spa	########	West Indie	0	0	0-49										
13	V Kohli	120	120	179	1	0	125	14	1	96	1	v West Ind	Port of Spa	########	India	0	1	100-149										
14	SS Iyer	71	71	101	1	0	68	5	1	104.41	1	v West Ind	Port of Spa	########	India	1	0	50-99										
15	RR Pant	20	20	27	1	0	35	2	0	57.14	1	v West Ind	Port of Spa	########	India	0	0	0-49										
16	RG Sharma	18	18	75	1	0	34	2	0	52.94	1	v West Ind	Port of Spa	########	India	0	0	0-49										
17	RA Jadeja	16*	16	24	1	1	16	1	0	100	1	v West Ind	Port of Spa	########	India	0	0	0-49										
18	KM Jadhav	16	16	27	1	0	14	2	0	114.28	1	v West Ind	Port of Spa	########	India	0	0	0-49										
19	Mohamme	3*	3	11	1	1	5	0	0	60	1	v West Ind	Port of Spa	########	India	0	0	0-49										
20	S Dhawan	2	2	4	1	0	3	0	0	66.66	1	v West Ind	Port of Spa	########	India	0	0	0-49										
21	B Kumar	1	1	6	1	0	2	0	0	50	1	v West Ind	Port of Spa	########	India	0	0	0-49										
22	Kuldeep Ya	DNB	-	-	0	0	-	-	-	-	1	v West Ind	Port of Spa	########	India	0	0	-										
23	KK Ahmed	DNB	-	-	0	0	-	-	-	-	1	v West Ind	Port of Spa	########	India	0	0	-										
24	CR Brathwaite										1	v India	Port of Spa	########	West Indies				10	1	0	53	3	1	0	0	0-4	5.3
25	RL Chase										1	v India	Port of Spa	########	West Indies				10	1	1	37	1	1	0	0	0-4	3.7
26	SS Cottrell										1	v India	Port of Spa	########	West Indies				10	1	0	49	1	1	0	0	0-4	4.9
27	JO Holder										1	v India	Port of Spa	########	West Indies				9	1	0	53	1	1	0	0	0-4	5.88
28	O Thomas										1	v India	Port of Spa	########	West Indies				4	1	0	32	0	1	0	0	0-4	8
29	KAJ Roach										1	v India	Port of Spa	########	West Indies				7	1	0	54	0	1	0	0	0-4	7.71
30	E Lewis										1	v India	Port of Spa	########	West Indies				DNB	0	-	-	-	1	0	0	-	-
31	SD Hope										1	v India	Port of Spa	########	West Indies				DNB	0	-	-	-	1	0	0	-	-
32	N Pooran										1	v India	Port of Spa	########	West Indies				DNB	0	-	-	-	1	0	0	-	-
33	SO Hetmyer										1	v India	Port of Spa	########	West Indies				DNB	0	-	-	-	1	0	0	-	-
34	CH Gayle										1	v India	Port of Spa	########	West Indies				DNB	0	-	-	-	1	0	0	-	-
35	B Kumar										2	v West Ind	Port of Spa	########	India				8	1	0	31	4	1	0	0	0-4	3.87
36	Mohammed Shami										2	v West Ind	Port of Spa	########	India				8	1	0	39	2	1	0	0	0-4	4.87
37	Kuldeep Yadav										2	v West Ind	Port of Spa	########	India				10	1	0	59	2	1	0	0	0-4	5.9
38	RA Jadeja										2	v West Ind	Port of Spa	########	India				4	1	0	15	1	1	0	0	0-4	3.75
39	KK Ahmed										2	v West Ind	Port of Spa	########	India				7	1	0	32	1	1	0	0	0-4	4.57

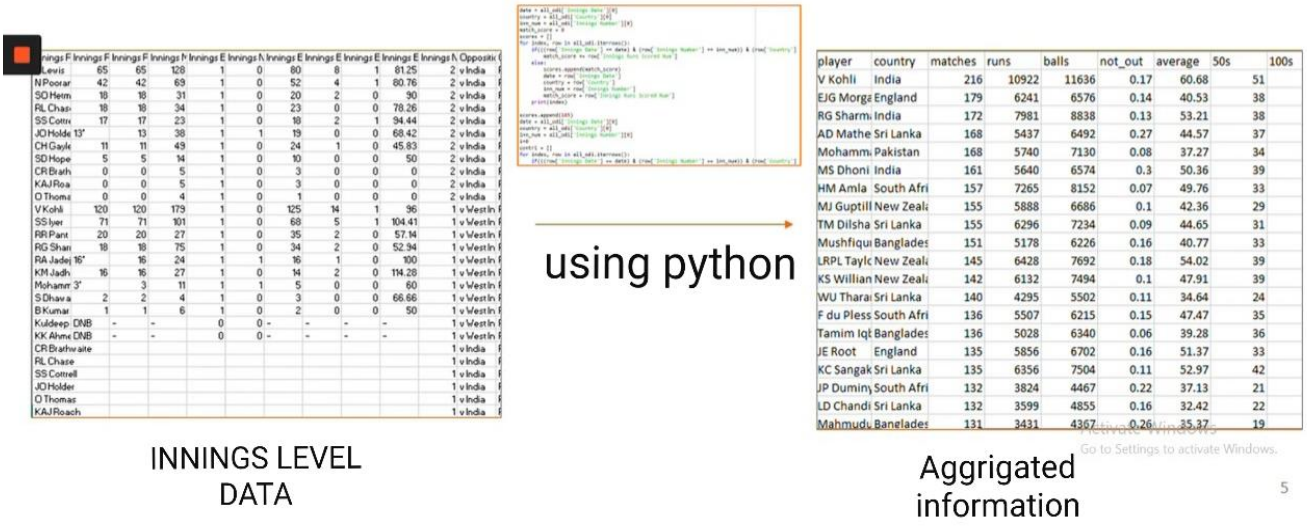
RAW DATA

3. Pre-processing the data

Now I am cleaning and pre-processing the raw data and extracting the required rows and columns. I use python to pre-processing the data.

Observe that for this study it is only required the data only for the time lines 1994 to 2004 and 2009 to 2019.

Then there are some missing values marked as "NAN" these are not required so, I remove them and the corresponding rows. Now removing the other unwanted data makes the innings level data to aggregated data.



nings F	Innings F	Innings F	Innings N	Innings E	Innings N	Innings E	Innings E	Innings E	Innings E	Innings N	Oppositi	(
Lewis	65	65	128	1	0	80	8	1	81.25	2	v India	F
N Poorar	42	42	69	1	0	52	4	1	80.76	2	v India	F
SO Hetm	18	18	31	1	0	20	2	0	90	2	v India	F
RL Chas	18	18	34	1	0	23	0	0	78.26	2	v India	F
SS Cottr	17	17	23	1	0	18	2	1	94.44	2	v India	F
JO Holde	13*		13	38	1	1	19	0	0	68.42	2	v India	F
CH Gayle	11	11	49	1	0	24	1	0	45.83	2	v India	F
SD Hope	5	5	14	1	0	10	0	0	50	2	v India	F
CR Brath	0	0	5	1	0	3	0	0	0	2	v India	F
KAJ Roa	0	0	5	1	0	3	0	0	0	2	v India	F
O Thoma	0	0	4	1	0	1	0	0	0	2	v India	F
V Kohli	120	120	179	1	0	125	14	1	96	1	v West In	F
SS Iyer	71	71	101	1	0	68	5	1	104.41	1	v West In	F
RR Pant	20	20	27	1	0	35	2	0	57.14	1	v West In	F
RG Shar	18	18	75	1	0	34	2	0	52.94	1	v West In	F
RA Jadej	16*		16	24	1	1	16	1	0	100	1	v West In	F
KM Jadh	16	16	27	1	0	14	2	0	114.28	1	v West In	F
Mohamm	3*		3	11	1	1	5	0	0	60	1	v West In	F
S Dhawa	2	2	4	1	0	3	0	0	66.66	1	v West In	F
B Kumar	1	1	6	1	0	2	0	0	50	1	v West In	F
Kuldeep	DNB	-	-	0	0	-	-	-	-	1	v West In	F
KK Ahme	DNB	-	-	0	0	-	-	-	-	1	v West In	F
CR Brathwaite										1	v India	F
RL Chase										1	v India	F
SS Cottrell										1	v India	F
JO Holder										1	v India	F
O Thomas										1	v India	F
KAJ Roach										1	v India	F

INNINGS LEVEL DATA

```
date = all_odi['Innings Date'][0]
country = all_odi['Country'][0]
inn_num = all_odi['Innings Number'][0]
match_score = 0
scores = []
for index, row in all_odi.iterrows():
    if(((row['Innings Date'] == date) & (row['Innings Number'] == inn_num)) & (row['Country']
        match_score += row['Innings Runs Scored Num']
    else:
        scores.append(match_score)
        date = row['Innings Date']
        country = row['Country']
        inn_num = row['Innings Number']
        match_score = row['Innings Runs Scored Num']
    print(index)

scores.append(105)
date = all_odi['Innings Date'][0]
country = all_odi['Country'][0]
inn_num = all_odi['Innings Number'][0]
i=0
contri = []
for index, row in all_odi.iterrows():
    if(((row['Innings Date'] == date) & (row['Innings Number'] == inn_num)) & (row['Country']
```

using python

player	country	matches	runs	balls	not_out	average	50s	100s
V Kohli	India	216	10922	11636	0.17	60.68	51	
EJG Morga	England	179	6241	6576	0.14	40.53	38	
RG Sharm	India	172	7981	8838	0.13	53.21	38	
AD Mathe	Sri Lanka	168	5437	6492	0.27	44.57	37	
Mohamm	Pakistan	168	5740	7130	0.08	37.27	34	
MS Dhoni	India	161	5640	6574	0.3	50.36	39	
HM Amla	South Afri	157	7265	8152	0.07	49.76	33	
MJ Guptill	New Zeal	155	5888	6686	0.1	42.36	29	
TM Dilsha	Sri Lanka	155	6296	7234	0.09	44.65	31	
Mushfiqu	Banglades	151	5178	6226	0.16	40.77	33	
LRPL Taylo	New Zeal	145	6428	7692	0.18	54.02	39	
KS Willian	New Zeal	142	6132	7494	0.1	47.91	39	
WU Thara	Sri Lanka	140	4295	5502	0.11	34.64	24	
F du Pless	South Afri	136	5507	6215	0.15	47.47	35	
Tamim Iqb	Banglades	136	5028	6340	0.06	39.28	36	
JE Root	England	135	5856	6702	0.16	51.37	33	
KC Sangak	Sri Lanka	135	6356	7504	0.11	52.97	42	
JP Duminy	South Afri	132	3824	4467	0.22	37.13	21	
LD Chandi	Sri Lanka	132	3599	4855	0.16	32.42	22	
Mahmudu	Banglades	131	3431	4367	0.26	35.37	19	

Aggrigated information

5

4 .Calculating the required parameter

PARAMETERS	SACHIN (1994-2004)	VIRAT (2009-2019)
RUNS PER INNINGS	43.6	50.2
STRIKE RATE	88.21	93.56
100'S PER MATCH	0.13	0.18
50'S PER MATCH	0.21	0.23
TEAM CONTRIBUTION	16.9%	17.6%

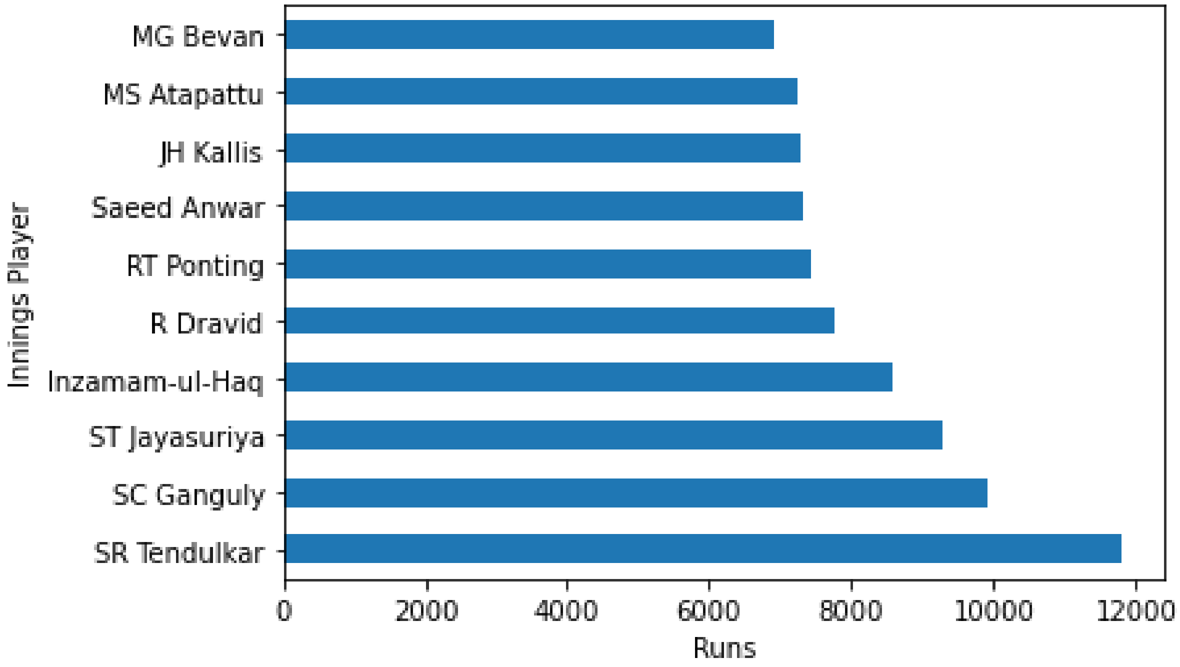
From the above table we can say that Virat scores almost 43.6 runs per match while Sachin scores 50.2 runs per match. If Virat plays 100 balls then he will score 93.56 runs on an average and Sachin will score 88.21 runs on an average. Sachin needs less no. of matches to score a 100 and 50 compare to Virat. If the team scores total100 runs then Virat contributes 17.6 runs and Sachin contributes 16.9 runs to the team.

Now, we know that these two batsman are of different generations, so it’s not fair if we only consider the above parameters to compare them. So we should consider the performance of top batsmen of their timeline and normalise the above parameters them on the basis of that.

Now, the top 10 run scorers in Sachin’s timeline (1994-2004) are------

Innings Player	**runs scored**
1. SR Tendulkar	11818
2. SC Ganguly	9911
3. ST Jayasuriya	9297
4. Inzamam-ul-Haq	8561
5. R Dravid	7751
6. RT Ponting	7422
7. Saeed Anwar	7320
8. JH Kallis	7267
9. MS Atapattu	7253
10. MG Bevan	6912

Visualisation

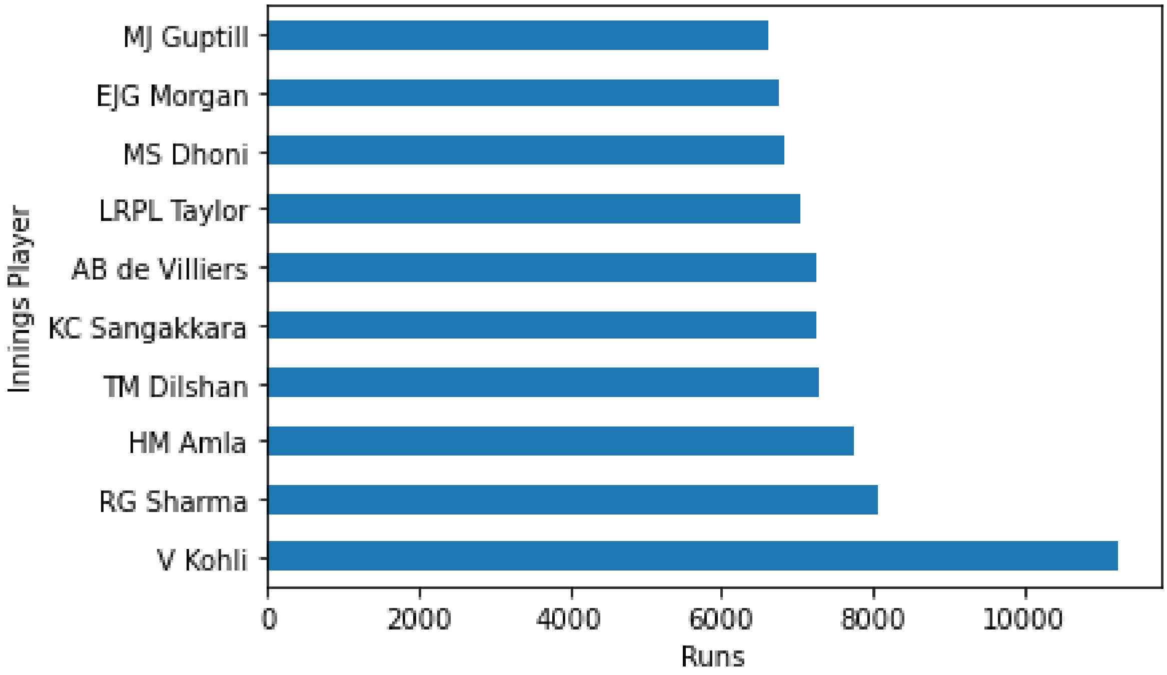


It shows that Sachin is the highest run scorer of his time period and almost 2000 runs ahead than the 2nd highest run scorer SOURAV GANGULY.

Now, the top 10 run scorers in Virat's timeline (1994-2004) are------

Innings Player	**runs scored**
1. V Kohli	11247
2. RG Sharma	8083
3. HM Amla	7745
4. TM Dilshan	7296
5. KC Sangakkara	7275
6. AB de Villiers	7247
7. LRPL Taylor	7059
8. MS Dhoni	6838
9. EJG Morgan	6748
10. MJ Guptill	6626

VISUALISATION:



It highlights that Virat completely dominates the other top scorers of his timeline and he almost 4000 runs ahead than the 2nd highest run scorer ROHIT SHARMA.

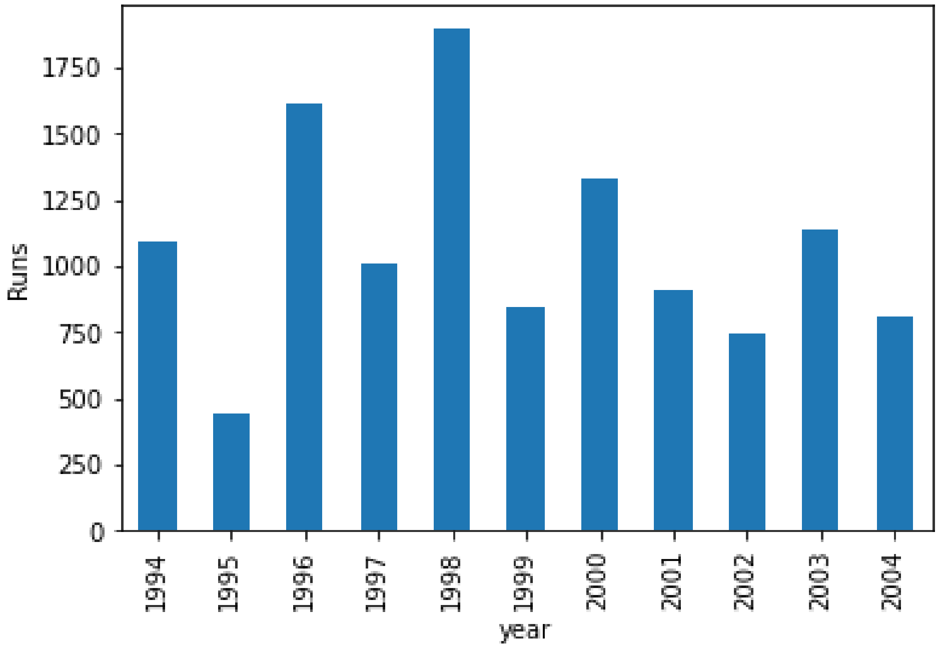
Another observation-

Runs scored by Sachin=11818	**Avg. runs scored by other 9 top scorers=7966**	**Difference = 3852**
Runs scored by Virat =11247	Avg. runs scored by other 9 top scorers=7213	Difference = 4034

Observe that both Sachin and Virat dominate their competitors of their timeline but Virat is more dominating than Sachin.

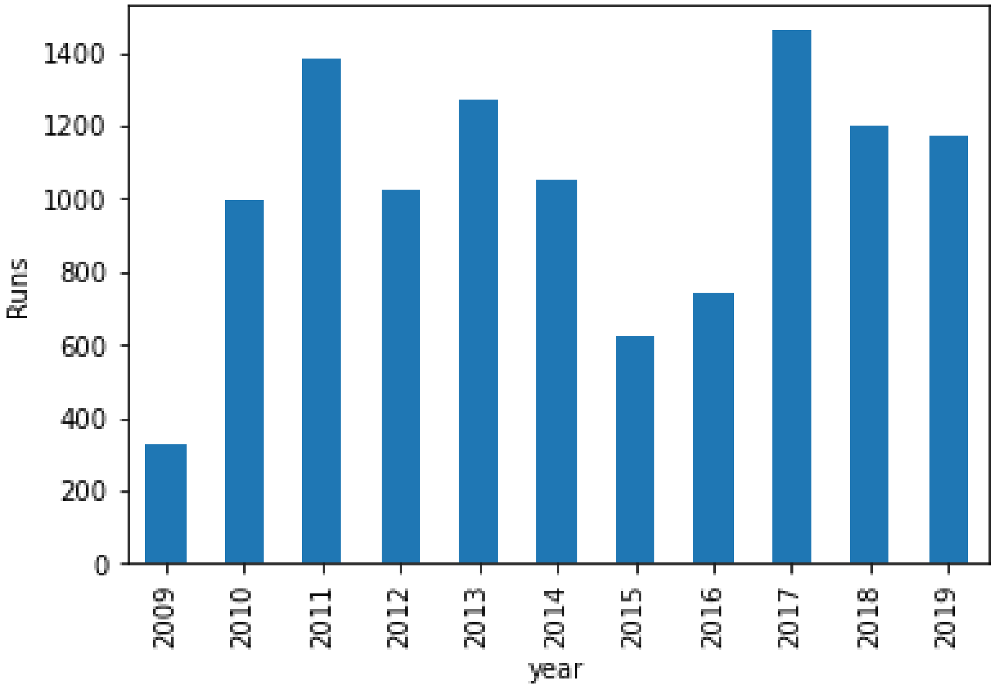
Now compare the both players performance year wise and let’s see which year is best for them and which is worst.

Sachin's scored run year wise-



It's clear that 1988 is his best year and in 1995 he didn't get much runs.

Now, Virat's scored run year wise-



It's clear that 2017 is his best year and in 2009 he didn't get much runs.

NORMALIZATION:

Normalisation is a technique to change the values of numeric columns in the dataset to use a common scale. In the method of normalization we take ratio of the values of parameters of one player and his competitors of his timeline. For Sachin we take the values his parameters and divide them with the average of exactly those parameters of his competitors in the years 1994 – 2004.And same for Virat .

Normalized parameter = $\dfrac{original\ parameter}{avarage\ of\ competitors\ parameter}$

Now we compare the normalised parameter of Virat and Sachin for best comparisons. The normalized parameters of both batsmen are given in the following table.

NORMALIZED PARAMETERS	SACHIN (1994-2004)	VIRAT (2009-2019)	WINNER
RUNS PER INNINGS	1.94	2.03	VIRAT
STRIKE RATE	1.22	1.12	SACHIN
100'S PER MATCH	0.15	0.18	SACHIN
50'S PER MATCH	0.57	0.55	VIRAT
TEAM CONTRIBUTION	16.9%	17.6%	VIRAT

5. Conclusion

On the light of the given data we can conclude that SACHIN wins the battle w.r.t 2 parameters “strike rate” and “100’s per match”. On the other hand VIRAT wins the battle w.r.t 3 parameters “RUNS PER INNINGS”, “50’S PER MATCH” and “TEAM CONTRIBUTION”.

Clearly VIRAT is the ultimate winner w.r.t to this analytical comparison. Although we take into account both kind of innings 1st or 2nd in these study, the outcome may be differ if we only consider 1st innings or 2nd innings one at a time.

Let's move to our 2nd question-

2. *MS DHONI vs. DALE STEYN (IN IPL): Statistical comparison(batsman vs. bowler match-up)*

It's clearly a batsman vs. bowler comparison, this type of comparison known as match-up.

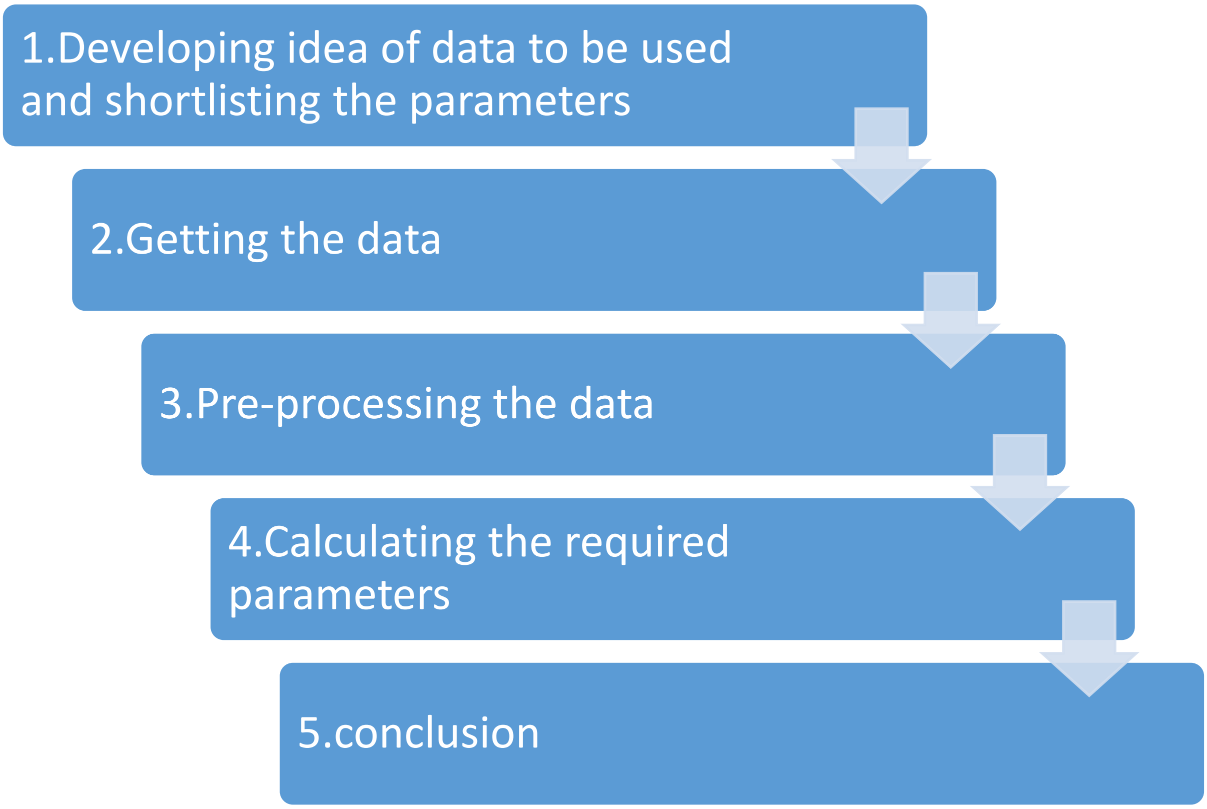
"Matchups": pitting bowlers against the optimum batsmen to stifle run scoring and take cheap wickets. Match ups are very useful to visualise which bowler is more deadly to a specific batsman or vice versa. Match ups help to make decision such as who will bowl the next over or who will going to bat next etc.

The South African pacer DALE STEYN is one of the most valuable bowler in the death overs in IPL. On the other hand the ex-Indian captain MAHENDRA SINGH DHONI is one of the greatest finisher of all time.

In IPL many times these two legends face off each other. Let us see who is more dominated.

Let compare these two players analytically using the ball by ball IPL data.

<u>APPROACH WE SHOULD TAKE TO OBTAIN MATCH UPS (BATSMAN VS. BOWLER COMPARISON)</u>



1.Developing idea of data to be used and shortlisting the parameters

2.Getting the data

3.Pre-processing the data

4.Calculating the required parameters

5.conclusion

1. **<u>Developing idea of data to be used and shortlisting the parameters</u>**

<u>DALE STEYN</u>

DEBUT SEASON 2008

IPL TEAMS: RCB, DC, SRH, GL

<u>MS DHONI</u>

DEBUT SEASON 2008

IPL TEAMS: CSK, RPS

PARAMETERS WE WILL USE FOR COMPARISON:

- NUMBER OF RUNS SCORED
- NUMBER OF BALLS FACED
- NUMBER OF TIMES DISMISSED
- STRIKE RATE

It's give the idea about the number of runs scored by the batsman per 100 balls on an average. Percentage of strike rate indicates if the batsman 100 then how much he can score.

# SR = 100*(Total Runs/Total Balls)

#METHODOLOGY:

We first need to find what are the numbers when MS DHONI faces DALE STEYN in IPL? Then following the steps-

Step 1: Filter by player names

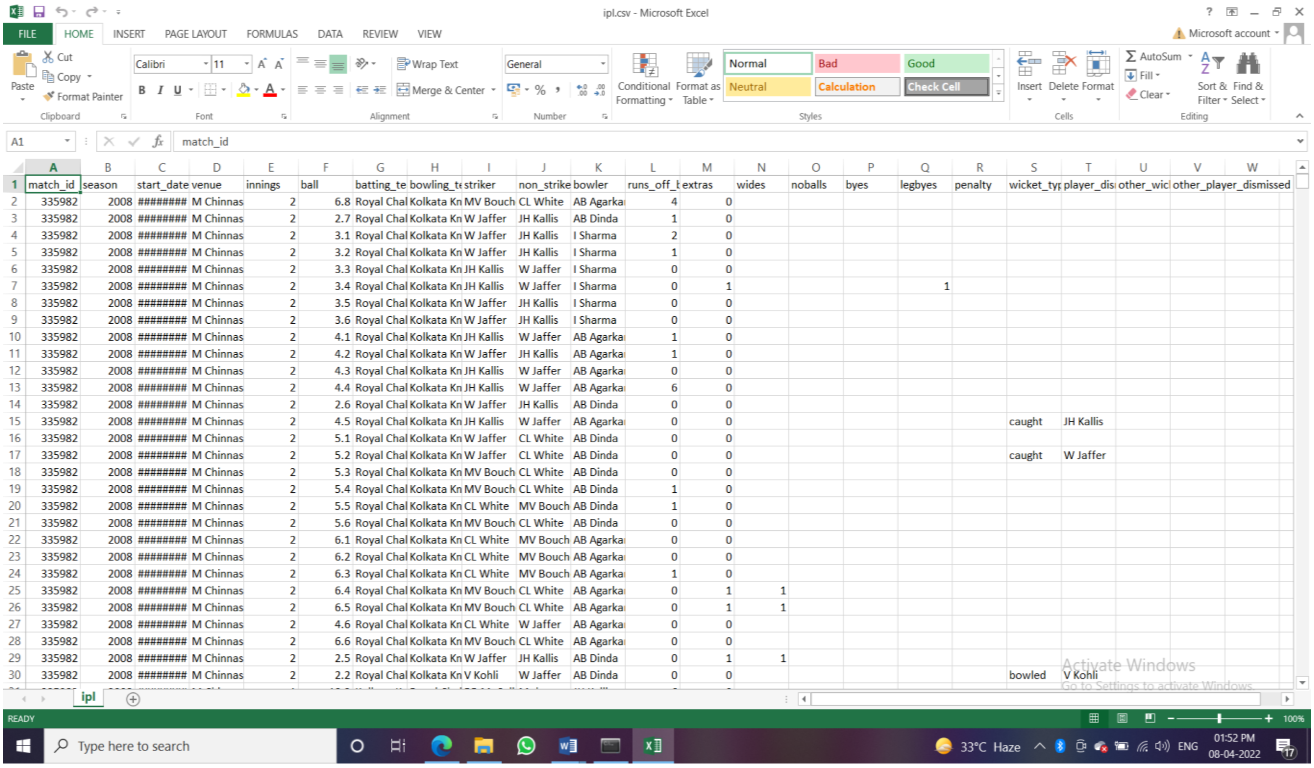
Step 2: Use these names & assign it to striker & bowler

Step 3: Get the required columns

Step 4: conclusion

2. Getting the data

For this study I collected the raw data from cricksheet (Match data - Cricsheet) IPL ball-by-ball data. A small glimpse of the raw data is given below.


ipl.csv - Microsoft Excel
FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW
Microsoft account
Paste Cut Copy Format Painter
Clipboard
Calibri 11
Font
Wrap Text Merge & Center
Alignment
General
Number
Conditional Formatting Format as Table
Normal Bad Good Neutral Calculation Check Cell
Styles
Insert Delete Format
Cells
AutoSum Fill Clear Sort & Filter Find & Select
Editing
A1
match_id
ipl
READY
Type here to search
Activate Windows
Go to Settings to activate Windows.
33°C Haze
ENG
01:52 PM
08-04-2022
100%

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	match_id	season	start_date	venue	innings	ball	batting_te	bowling_te	striker	non_strike	bowler	runs_off_b	extras	wides	noballs	byes	legbyes	penalty	wicket_typ	player_dis	other_wic	other_player_dismissed	
2	335982	2008	########	M Chinnas	2	6.8	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Agarka	4	0										
3	335982	2008	########	M Chinnas	2	2.7	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	AB Dinda	1	0										
4	335982	2008	########	M Chinnas	2	3.1	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	I Sharma	2	0										
5	335982	2008	########	M Chinnas	2	3.2	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	I Sharma	1	0										
6	335982	2008	########	M Chinnas	2	3.3	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	I Sharma	0	0										
7	335982	2008	########	M Chinnas	2	3.4	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	I Sharma	0	1				1						
8	335982	2008	########	M Chinnas	2	3.5	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	I Sharma	0	0										
9	335982	2008	########	M Chinnas	2	3.6	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	I Sharma	0	0										
10	335982	2008	########	M Chinnas	2	4.1	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	AB Agarka	1	0										
11	335982	2008	########	M Chinnas	2	4.2	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	AB Agarka	1	0										
12	335982	2008	########	M Chinnas	2	4.3	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	AB Agarka	0	0										
13	335982	2008	########	M Chinnas	2	4.4	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	AB Agarka	6	0										
14	335982	2008	########	M Chinnas	2	2.6	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	AB Dinda	0	0										
15	335982	2008	########	M Chinnas	2	4.5	Royal Chal	Kolkata Kn	JH Kallis	W Jaffer	AB Agarka	0	0						caught	JH Kallis			
16	335982	2008	########	M Chinnas	2	5.1	Royal Chal	Kolkata Kn	W Jaffer	CL White	AB Dinda	0	0										
17	335982	2008	########	M Chinnas	2	5.2	Royal Chal	Kolkata Kn	W Jaffer	CL White	AB Dinda	0	0						caught	W Jaffer			
18	335982	2008	########	M Chinnas	2	5.3	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Dinda	0	0										
19	335982	2008	########	M Chinnas	2	5.4	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Dinda	1	0										
20	335982	2008	########	M Chinnas	2	5.5	Royal Chal	Kolkata Kn	CL White	MV Bouch	AB Dinda	1	0										
21	335982	2008	########	M Chinnas	2	5.6	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Dinda	0	0										
22	335982	2008	########	M Chinnas	2	6.1	Royal Chal	Kolkata Kn	CL White	MV Bouch	AB Agarka	0	0										
23	335982	2008	########	M Chinnas	2	6.2	Royal Chal	Kolkata Kn	CL White	MV Bouch	AB Agarka	0	0										
24	335982	2008	########	M Chinnas	2	6.3	Royal Chal	Kolkata Kn	CL White	MV Bouch	AB Agarka	1	0										
25	335982	2008	########	M Chinnas	2	6.4	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Agarka	0	1	1									
26	335982	2008	########	M Chinnas	2	6.5	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Agarka	0	1	1									
27	335982	2008	########	M Chinnas	2	4.6	Royal Chal	Kolkata Kn	CL White	W Jaffer	AB Agarka	0	0										
28	335982	2008	########	M Chinnas	2	6.6	Royal Chal	Kolkata Kn	MV Bouch	CL White	AB Agarka	0	0										
29	335982	2008	########	M Chinnas	2	2.5	Royal Chal	Kolkata Kn	W Jaffer	JH Kallis	AB Dinda	0	1	1									
30	335982	2008	########	M Chinnas	2	2.2	Royal Chal	Kolkata Kn	V Kohli	W Jaffer	AB Dinda	0	0						bowled	V Kohli			

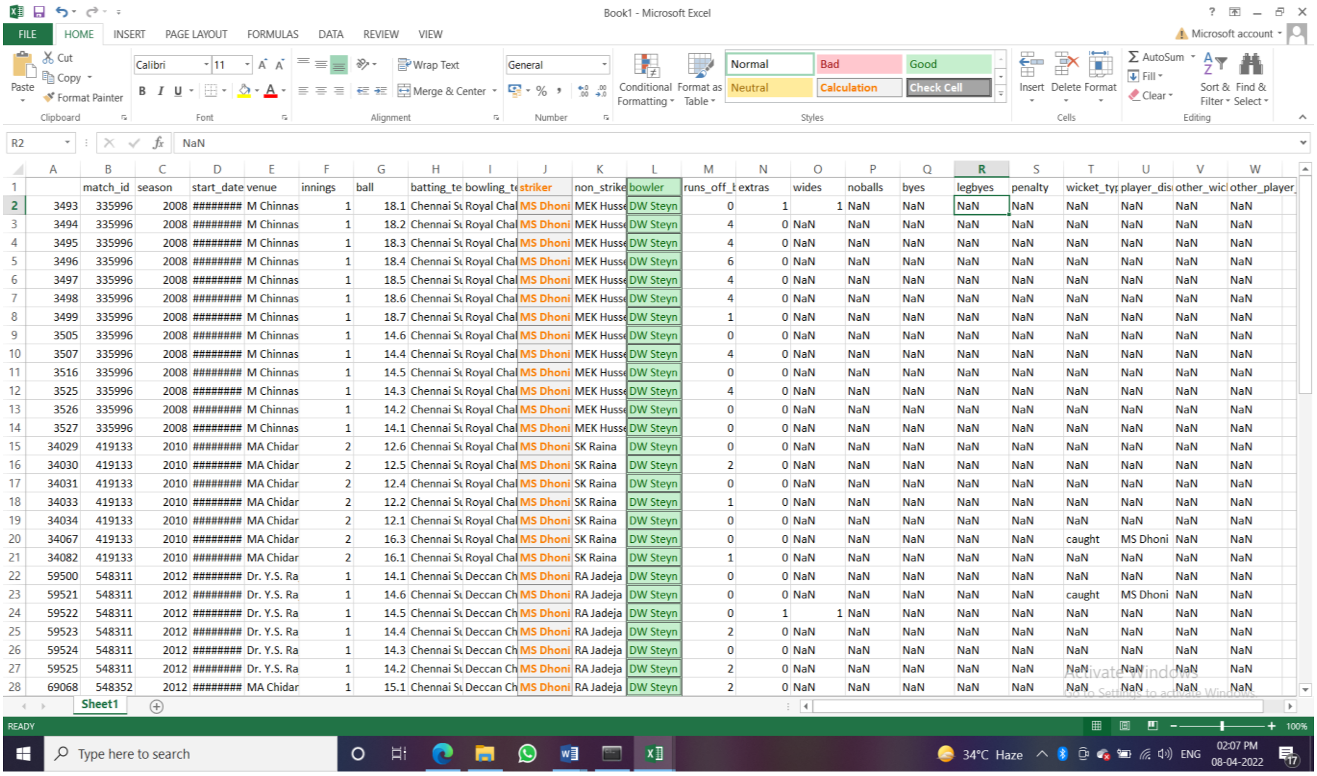
<u>IPL BALL BY BALL RAW DATA</u>

3. Pre-processing the data

Now I am cleaning and pre-processing the raw data and extracting the required rows and columns. I use python to pre-processing the data.

Observe that for this study it is only required the data only for MS Dhoni as a batsman and Dale Steyn as a bowler. That means when the striker name is MS Dhoni and the bowler name is DW Steyn and sort the required data.

Then there are some missing values marked as "NAN" these are not required so, I remove them and the corresponding rows. A screenshot of the processed data is given below.



R2 NaN

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1		match_id	season	start_date	venue	innings	ball	batting_te	bowling_te	striker	non_strike	bowler	runs_off_b	extras	wides	noballs	byes	legbyes	penalty	wicket_typ	player_dis	other_wicl	other_player_
2	3493	335996	2008	########	M Chinnas	1	18.1	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	0	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	3494	335996	2008	########	M Chinnas	1	18.2	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	3495	335996	2008	########	M Chinnas	1	18.3	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5	3496	335996	2008	########	M Chinnas	1	18.4	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	6	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	3497	335996	2008	########	M Chinnas	1	18.5	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	3498	335996	2008	########	M Chinnas	1	18.6	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	3499	335996	2008	########	M Chinnas	1	18.7	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	3505	335996	2008	########	M Chinnas	1	14.6	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	3507	335996	2008	########	M Chinnas	1	14.4	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	3516	335996	2008	########	M Chinnas	1	14.5	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12	3525	335996	2008	########	M Chinnas	1	14.3	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	4	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13	3526	335996	2008	########	M Chinnas	1	14.2	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	3527	335996	2008	########	M Chinnas	1	14.1	Chennai Su	Royal Chal	MS Dhoni	MEK Husse	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	34029	419133	2010	########	MA Chidar	2	12.6	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	34030	419133	2010	########	MA Chidar	2	12.5	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
17	34031	419133	2010	########	MA Chidar	2	12.4	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18	34033	419133	2010	########	MA Chidar	2	12.2	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19	34034	419133	2010	########	MA Chidar	2	12.1	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
20	34067	419133	2010	########	MA Chidar	2	16.3	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	caught	MS Dhoni	NaN	NaN
21	34082	419133	2010	########	MA Chidar	2	16.1	Chennai Su	Royal Chal	MS Dhoni	SK Raina	DW Steyn	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22	59500	548311	2012	########	Dr. Y.S. Ra	1	14.1	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	59521	548311	2012	########	Dr. Y.S. Ra	1	14.6	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	caught	MS Dhoni	NaN	NaN
24	59522	548311	2012	########	Dr. Y.S. Ra	1	14.5	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	0	1	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25	59523	548311	2012	########	Dr. Y.S. Ra	1	14.4	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
26	59524	548311	2012	########	Dr. Y.S. Ra	1	14.3	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
27	59525	548311	2012	########	Dr. Y.S. Ra	1	14.2	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
28	69068	548352	2012	########	MA Chidar	1	15.1	Chennai Su	Deccan Ch	MS Dhoni	RA Jadeja	DW Steyn	2	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

PROCESSED DATA (WHERE STRIKER = MS DHONI AND BOWLER = D.STEYN)

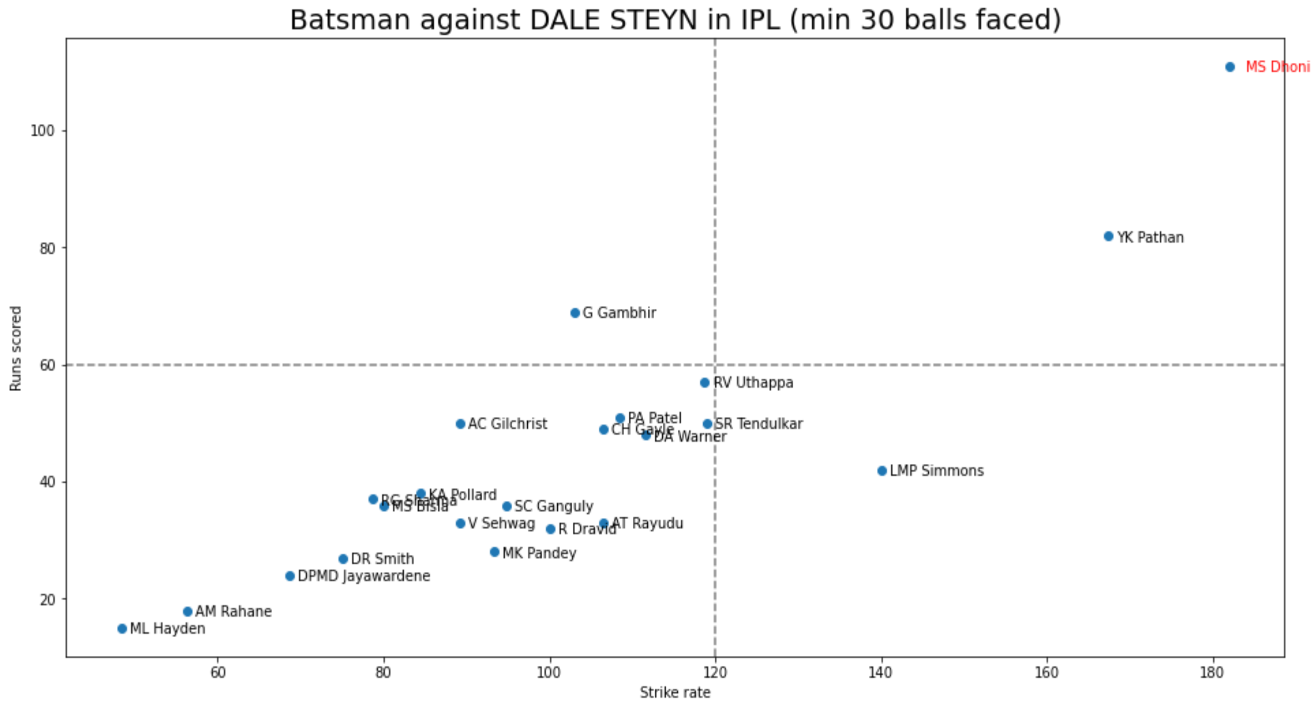
4 .Calculating the required parameter

RUNS SCORED	**111**
BALLS FACED	61
NUMBER OF TIMES DISMISSED	2
STRIKE RATE	181.96

In IPL, MS Dhoni has managed to score 111 runs off 61 balls by Dale Steyn with the strike rate of 181.96, while Dale Steyn has dismissed him 2 times. Now this above information not enough to compare them. We should check the performance of STEYN against all other batsman and similarly performance of DHONI against all other bowlers and then we can conclude who win this match up.

Visualisation

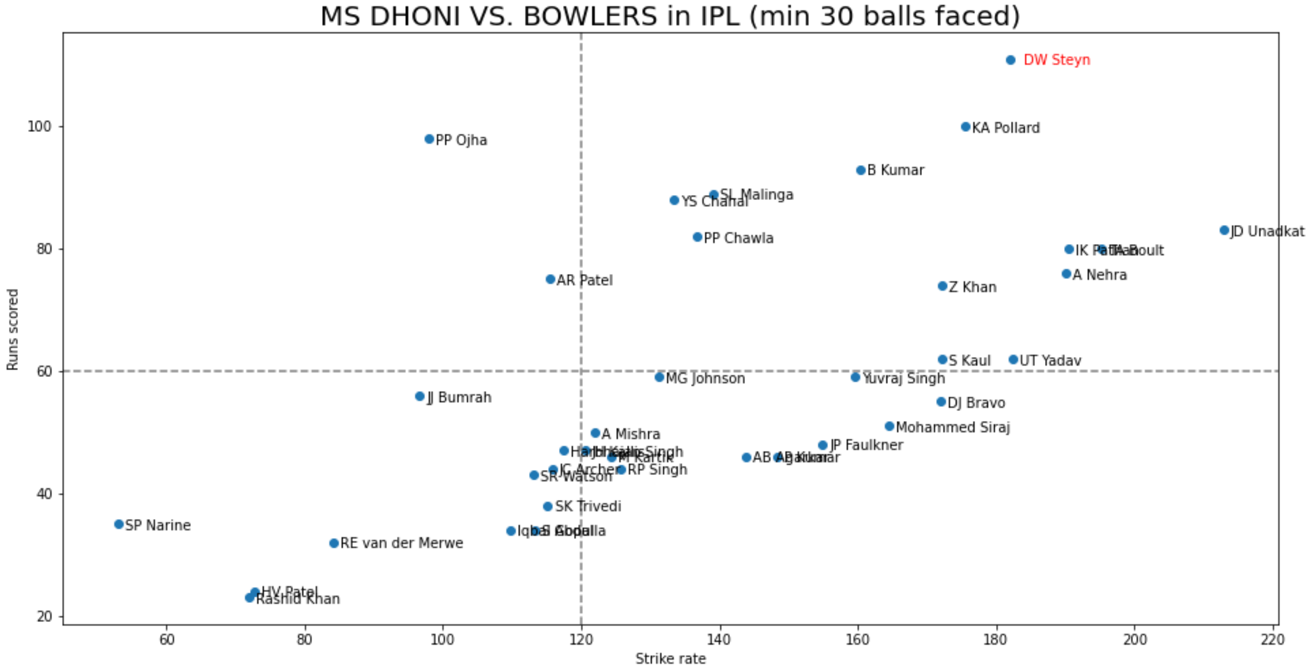
I use scatter plot to visualise the performance of batsmen against DALE STEYN, the x-axis represents the strike rate and y-axis represent the number of runs scored. I include only those batsmen who have played minimum 30 balls against STEYN for proper comparison.



Observe that batsman in 3rd quadrant are the favourite opponent of STEYN, they can't score much runs against him also with poor strike rate. The batsmen is 1st quadrant are least favourite opponent of STEYN, they scored so many runs with healthy strike rate against him.

The scatterplot clearly highlights that MS DHONI (marked in red colour) scored most number of runs as well as with highest strike rate against STEYN.

Similarly, I use scatter plot to visualise the performance of bowlers against MS DHONI, the x-axis represents the strike rate and y-axis represent the number of runs scored. I include only those bowlers who have bowled minimum 30 balls against DHONI for proper comparison.



Observe that the bowlers in the 1st quadrant are favourite bowlers of DHONI, he has scored most number of runs against them with high strike rate. DALE STEYN (marked in red colour) is in the 1st quadrant which indicates he is one of the favourite bowler of MS DHONI. That means DHONI scored many runs against STEYN with a healthy strike rate.

5. Conclusion

From the above calculated parameters we can see that MS DHONI has scored 111 runs out off 61 balls with a strike rate of 181.96 against DALE STEYN and only 2 times he get dismissed.

Also from the above scatter plots it's also clear that MS DHONI plays STEYN really well and DHONI is the most run scorer against DALE STEYN with the highest strike rate. So, we can conclude that MS DHONI wins this match up against DALE STEYN.

Let's move to the 3rd question---

3 *Whether MS DHONI is still fit to bat or not? (analyzing the performance of a player)*

Mahendra Singh Dhoni, as known to millions of cricket fans worldwide by MSD, has been a stalwart and a true heavyweight in the game of cricket for so many years. A genuine gentleman & a man of integrity, MSD has been a great brand ambassador for cricket ever since the day he stepped foot inside the ropes. While no one can question his abilities as a skipper & his skills behind the wickets, doubts have started surfacing whether he can continue his mesmerising destruction with the willow in hand like he has been doing for so many years. We try to quantify his efforts and contribution with the bat in the Indian Premier League and analyse whether anyone should pay heed to the ever-growing and louder-by-the-day murmurs about his supposed-drop in performance and whether he can still deliver with the bat. Since he is a finisher we are also interested in his performance in death overs.

KEY PERFORMANCE INDICATORS

- Dot ball percentage
- Is CSK better without MSD the batter?
- Strike rate
- Number of boundary smashed in the last overs

MSD'S STATS OVER THE YEARS

MSD has been playing for the Chennai Super Kings and Rising Pune Supergiants franchise ever since the dawn of the league called IPL. His numbers in the IPL, as shown below, are amongst the greatest of the greats.

Timeline	Innings	Runs	Balls Played	Strike Rate
2008-2021	220	4746	3604	131.69

However, if we delve deep into his stats, there's something that needs attention.

Timeline	Innings	Runs	Balls Played	Strike Rate
2008-2017	159	3561	2680	131.68
2018-2021	61	1185	924	131.68

So, clearly, there's a dip in his numbers over the last three years. This demands some further inspection and we proceed to do so.

1. DOT BALL PERCENTAGE

Dot ball percentage is defined as the percentage of dot balls played by a batter, defined over a certain timeframe. Here, we compare the average Dot ball percentage in IPL vs the same for MSD, over the two timeframes.

SEASONS	NO. OF BALLS PLAYED	NO. OF DOT BALLS	DOT BALL PERCENTAGE
2008-2017	2680	925	34.51%
2018-2021	924	353	38.2%

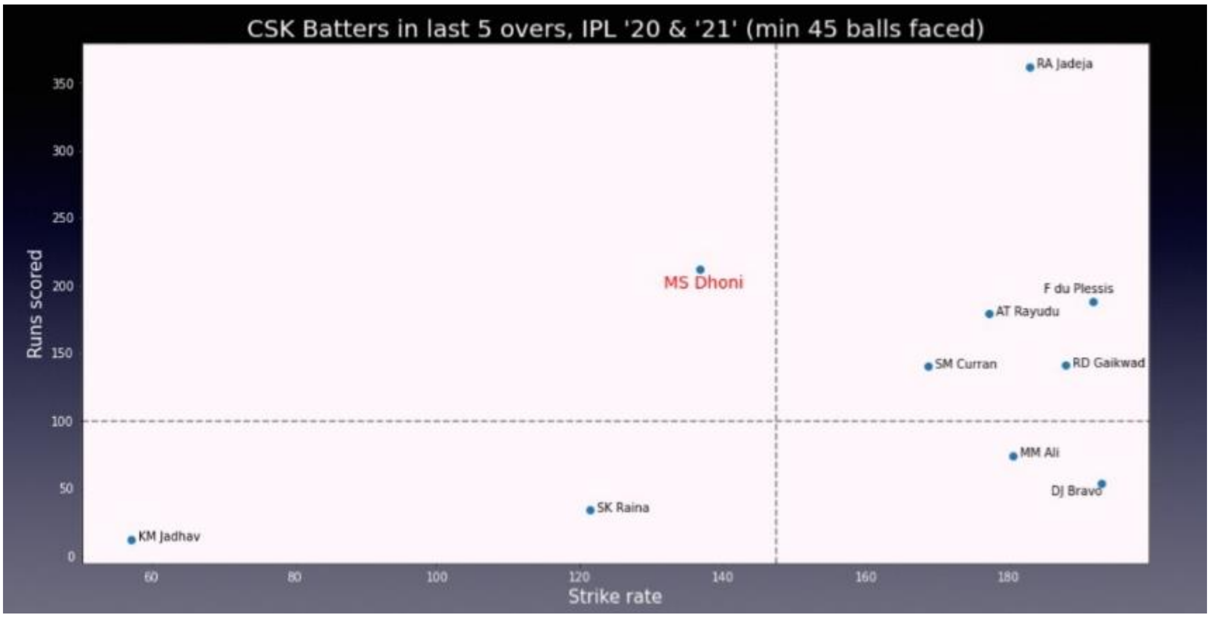
The above table clearly indicates that Dhoni played more dot balls in last 3 years compare to previous years. In 2008-2017 if he played 100 balls then he scored none or 0 in 34 balls on an average. Again in 2018-2021 if he played 100 balls then he scored none or 0 in 38 balls on an average. So we can say that he plays more dot balls in last 3 years.

2. Is CSK better without MSD the batter?

So we need to check team contribution of DHONI. Since he is a finisher we are also interested in his performance in death overs.

Season	Team contribution(all over)	Team contribution in last 5 overs
2008-2017	19.2%	49%
2018-2021	13%	36%

The above table shows that DHONI contributed much more runs in last 5 overs for his team as he played the role of a finisher. But observe that in 2008-2017 if he contributed 49% runs in last 5 overs whereas in last 3 years it reduced down 36%. But 36% runs in last 5 overs not a bad contribution at all. Though the overall team contribution also reduced in last 3 years.



The above plot shows runs vs. strike rate of CSK players (min 45 balls played) in last 5 overs of last 2 seasons. Here we can see that MSD (marked in red)get moderate runs in last 5 overs but strike rate is not very healthy.

3. Strike rate

Let's check MSD'S overall strike rate and that of last 3 years.

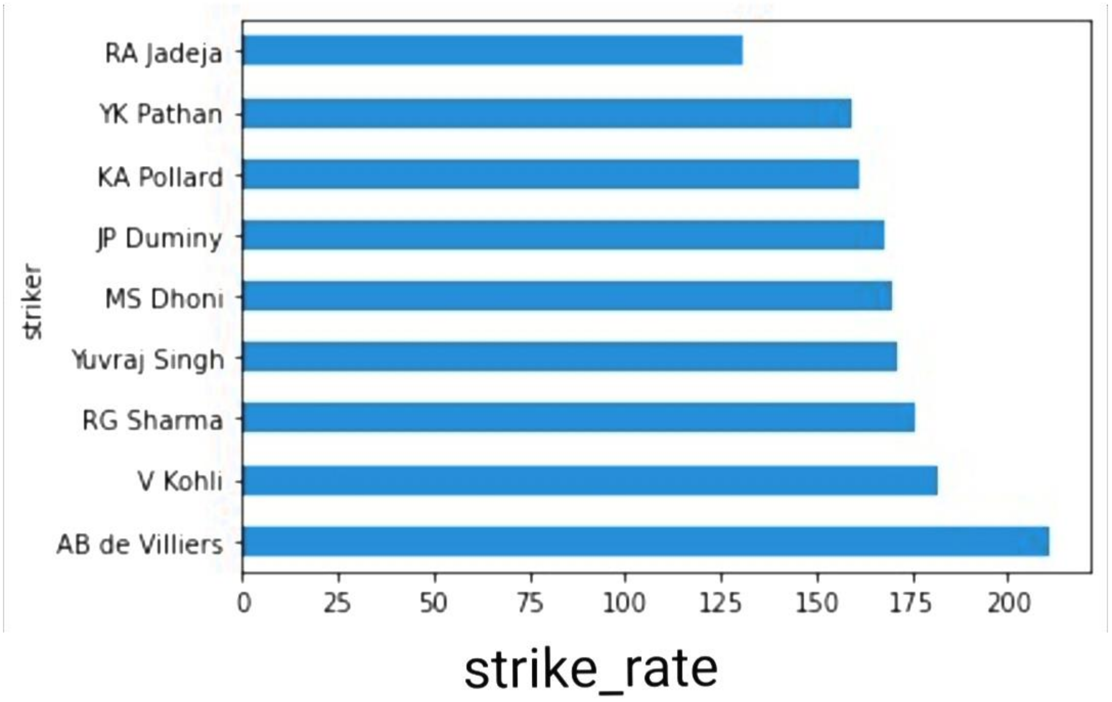
Season	Strike rate (all over)	Strike rate in last 5 overs
2008-2017	131.68%	169.6%
2018-2021	131.68%	162.2%

The table shows his strike rate is more all less same in both the time line. For better understanding compare his strike rate with other players. As we know MSD is the finisher, batted in death overs so we compare strike rates in last 5 overs.

TOP 9 BATSMAN WITH HIGHEST STRIKE RATE IN LAST 5 OVERS IN 2008-2017(minimum 500 balls played):-

Striker	Strike rate
1. AB de Villiers	211.052632
2. V Kohli	181.868132
3. RG Sharma	175.668449
4. Yuvraj Singh	171.124031
5. MS Dhoni	169.607843
6. JP Duminy	167.760618
7. KA Pollard	161.217184
8. YK Pathan	159.246575
9. RA Jadeja	130.729167

VISUALISATION:

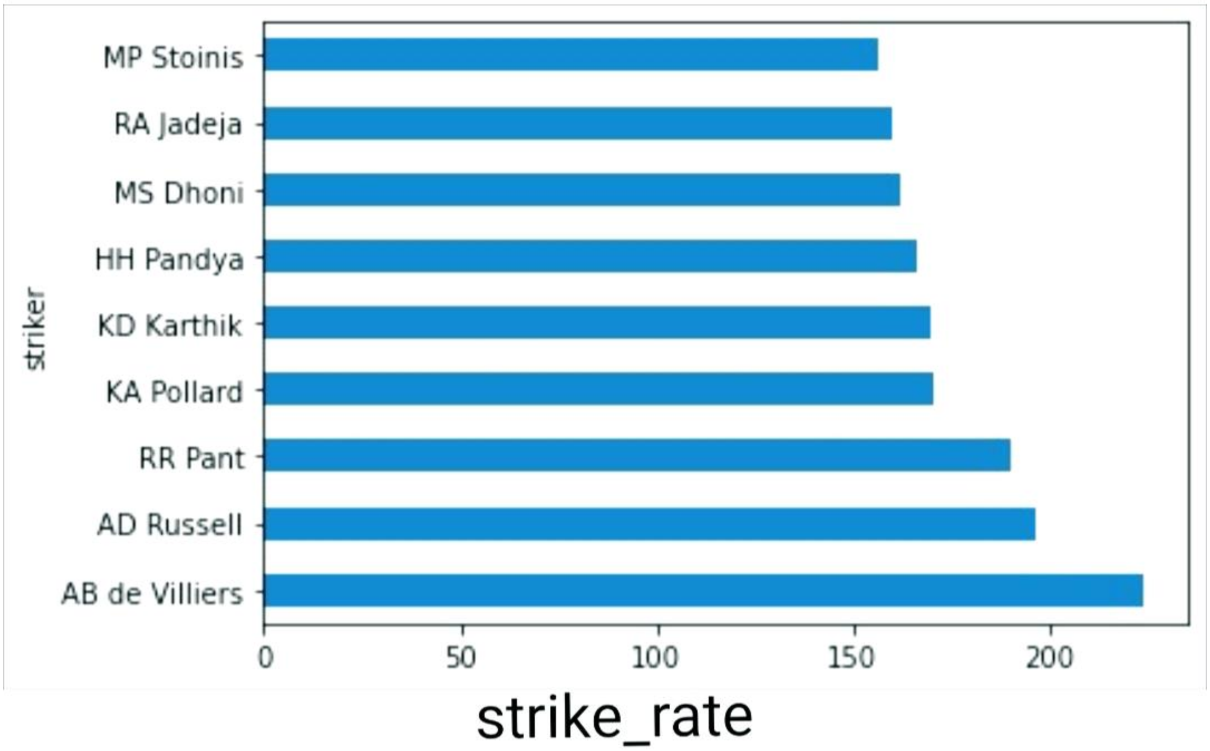


Clearly MS DHONI ranked no. 5 in terms of highest strike rate in death overs in 2008-2017 seasons. It shows how consistently he scored runs in death overs for his teams. Also if we consider more experience as more numbers of balls played (let minimum 1000 balls played) then MSD comes out to be in rank 1. So he is too good as a finisher in these years.

TOP 9 BATSMAN WITH HIGHEST STRIKE RATE IN LAST 5 OVERS IN 2018-2021(minimum 250 balls played):-

	Striker	**Strike rate**
1.	AB de Villiers	223.905724
2.	AD Russell	196.060606
3.	RR Pant	190.332326
4.	KA Pollard	170.381232
5.	KD Karthik	169.428571
6.	HH Pandya	165.848214
7.	MS Dhoni	162.212944
8.	RA Jadeja	160.182371
9.	MP Stoinis	156.299213

VISUALISATION:



Clearly MS DHONI ranked no. 7 in terms of highest strike rate in death overs in 2018-2021 seasons. It shows his performance lacked little bit in last 3 years. So he is still a good finisher but not as effective as vintage MSD.

4. **Number of boundary smashed in the last overs**

No. of boundaries smashed by DHONI in last 5 overs according to the seasons.

SEASON	NO. OF SIXES	NO. OF FOURS	TOTAL NO. OF BOUNDARIES	TOTAL NO. OF BALLS FACED
2008-2017	114	155	269	1703
2018-2021	50	52	102	777
Total	164	207	371	2480

In 2008-2017 MSD scored total 269 boundaries out of 1703 balls in last 5 overs , in this time period his boundaries per ball percentage was 15.8%. In 2018-2021 MSD scored total 102 boundaries out of 777 balls in last 5 overs , in this time period his boundaries per ball percentage was 13.1%. Which indicates that his boundaries hitting capability is also lacked in last 3 years compare to other seasons.

Let's compare MSD's boundary hitting capability with other players.

SEASONS	TOP 10 SIX HITTERS IN LAST 5 OVERS	TOP 10 FOUR HITTERS IN LAST 5 OVERS
2008-2017	1. 1.MS Dhoni 114 2. KA Pollard 92 3. AB de Villiers 86 4. RG Sharma 81 5. Yuvraj Singh 66 6. V Kohli 62 7. DA Miller 50 8. JP Duminy 47 9. YK Pathan 45 10. JA Morkel 39	1. MS Dhoni 155 2. RG Sharma 107 3. AB de Villiers 93 4. KA Pollard 90 5. YK Pathan 81 6. V Kohli 77 7. KD Karthik 63 8. Yuvraj Singh 61 9. IK Pathan 60 10. Harbhajan Singh 57
2018-2021	1. AD Russell 65 2. AB de Villiers 54 3. HH Pandya 51 4. MS Dhoni 50 5. KA Pollard 45 6. RR Pant 41 7. RA Jadeja 27 8. KD Karthik 27 9. KL Rahul 23 10. SV Samson 22	1. KD Karthik 59 2. RR Pant 58 3. HH Pandya 54 4. MS Dhoni 52 5. RA Jadeja 42 6. AB de Villiers 42 7. AD Russell 41 8. MP Stoinis 39 9. KL Rahul 36 10. KA Pollard 35
ALL SEASONS	1. MS Dhoni 164 2. AB de Villiers 140 3. KA Pollard 137 4. RG Sharma 90 5. AD Russell 86 6. V Kohli 78 7. HH Pandya 74 8. Yuvraj Singh 68 9. AT Rayudu 58 10. YK Pathan 56	1. MS Dhoni 207 2. AB de Villiers 135 3. KA Pollard 125 4. RG Sharma 123 5. KD Karthik 122 6. RA Jadeja 97 7. YK Pathan 93 8. V Kohli 92 9. AT Rayudu 80 10. HH Pandya 71

The above table shows that MS Dhoni is the most boundary hitter of all time in last 5 overs of IPL. From season 2008 to season 2017 he ranked no.1 in both highest number of 6 hitting and most number of 4 hitting in death overs. But in last 3 years he ranked no.4 in both highest number of 6 hitting and most number of 4 hitting in death overs.

CONCLUSION:

By comparing above parameters we can conclude that MS Dhoni’s performance as a finisher lacked in last 3 years. But he is still good than many other players. As he carried his team in last overs for a very long time his finishing skill slightly lacked than his vintage version. But still he contributed almost 36% runs in crucial death overs. Definitely he is an asset for his team in death overs. So we conclude that though his performance lacked in last few years he is still fit to bat. Moreover we take into account both kind of innings 1st or 2nd in these study, the outcome may be differ if we only consider 1st innings or 2nd innings one at a time. By observing matches intuitively we can say that MS Dhoni playing a vital role when his team is chasing runs.

APPENDIX:

I use python programming language for this whole study.

Python code for SACHIN vs. VIRAT comparison:

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt

#to display all rows columns
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', -1)
```

SACHIN VS VIRAT

```
df = pd.read_csv('ODI_data.csv')

#METHODOLOGY:

# step 1: sortlisting the parameters
# step 2: pre-processing the data
# step 3: calculation of required parameters and visualisation
# step 4: conclusion

df.head(2)

len(df), len(df.columns)

(171968, 28)

#variables we are looking for:-

# Runs per innings
# SR
# 100's
# 50's
# Team contribution

df['Innings Runs Scored Num'].unique()

df = df[df['Innings Runs Scored Num'] != '-']

df['Innings Runs Scored Num'].unique()

df = df.dropna(subset = ['Innings Runs Scored Num'])

df['Innings Runs Scored Num'].unique()
```

```
df.head(1)

# convert to datetime
df['Innings Date'] = pd.to_datetime(df['Innings Date'])

df['year'] = df['Innings Date'].dt.year

df.tail(1)

df['Innings Runs Scored Num'] = df['Innings Runs Scored Num'].astype('int'
)

df['Innings Balls Faced'] = df['Innings Balls Faced'].astype('int')

df['Innings Not Out Flag'] = df['Innings Not Out Flag'].astype('int')

# Sachin 1994 - 2004
# Virat 2009 - 2019

sachin_df = df[(df.year >= 1994) & (df.year <= 2004)]

kohli_df = df[(df.year >= 2009) & (df.year <= 2019)]

sachin_df.head(2)

kohli_df.head(2)

  # Runs per innings = Total Runs/Total Innings
# SR = 100*(Total Runs/Total Balls)
# 100's = sum(100's)
# 50's = sum(50's)
# Team contribution = Player Runs/Team Runs (ex: Virat 50/ Team Ind 150 =>
50/150 : 33%)

# df.dtypes

# sachin_df.to_csv('sachin_data.csv')

# what is the total runs scored by sachin in these time frames?

# sachin_df.head(20)

# SR Tendulkar
sdf = sachin_df[sachin_df['Innings Player'] == 'SR Tendulkar']

sdf.head()


sum(sdf['Innings Runs Scored Num'])

11818

kdf = kohli_df[kohli_df['Innings Player'] == 'V Kohli']

# kohli_df['Innings Player'].unique()

kdf.head()
```

```
sum(kdf['Innings Runs Scored Num'])
```

```
11247
```

```
len(kdf), len(sdf)
```

```
(224, 271)
```

```
# RPI - Sachin, Virat
sum(kdf['Innings Runs Scored Num'])/len(kdf), sum(sdf['Innings Runs Scored
Num'])/len(sdf)
```

```
(50.20982142857143, 43.608856088560884)
```

```
# SR
100*sum(kdf['Innings Runs Scored Num'])/sum(kdf['Innings Balls Faced']), 1
00*sum(sdf['Innings Runs Scored Num'])/sum(sdf['Innings Balls Faced'])
```

```
(93.56126778138258, 88.21377920429947)
```

```
# 100's
sum(kdf["100's"]), sum(sdf["100's"])
```

```
(42.0, 37.0)
```

```
# 50's
sum(kdf["50's"]), sum(sdf["50's"])
```

```
(53.0, 57.0)
```

```
# Team Contribution - Runs score by each player, Runs by team
sum(kdf['Innings Runs Scored Num']), sum(sdf['Innings Runs Scored Num'])
```

```
(11247, 11818)
```

```
# 1994 - 2004 = All players
sum(sachin_df[sachin_df.Country == 'India']['Innings Runs Scored Num'])
```

```
69715
```

```
# 2009 - 2019 = All players
sum(kohli_df[kohli_df.Country == 'India']['Innings Runs Scored Num'])
```

```
63867
```

```
100*sum(kdf['Innings Runs Scored Num'])/sum(kohli_df[kohli_df.Country == '
India']['Innings Runs Scored Num'])
```

```
17.610033350556627
```

```
100*sum(sdf['Innings Runs Scored Num'])/sum(sachin_df[sachin_df.Country ==
'India']['Innings Runs Scored Num'])
```

```
16.951875493078965
```

Visualizations:

```
sachin_df.groupby(['Innings Player'])['Innings Runs Scored Num'].sum().sor
t_values(ascending = False).head(10)
```

```
sachin_df.groupby(['Innings Player'])['Innings Runs Scored Num'].sum().sor
t_values(ascending = False).head(10).plot(kind = 'barh')
plt.xlabel('Runs')
plt.show()
```

```
kohli_df.groupby(['Innings Player'])['Innings Runs Scored Num'].sum().sort
_values(ascending = False).head(10)
```

```
kohli_df.groupby(['Innings Player'])['Innings Runs Scored Num'].sum().sort
_values(ascending = False).head(10).plot(kind = 'barh')
plt.xlabel('Runs')
plt.show()
```

```
sdf.head(1)
```

```
sdf.groupby(['year'])['Innings Runs Scored Num'].sum().plot(kind = 'bar')
plt.ylabel('Runs')
```

```
Text(0, 0.5, 'Runs')
```

```
kdf.groupby(['year'])['Innings Runs Scored Num'].sum().plot(kind = 'bar')
plt.ylabel('Runs')
```

```
Text(0, 0.5, 'Runs')
```

Normalization:

```
# RPI - Sachin, Virat
sum(kdf['Innings Runs Scored Num'])/len(kdf), sum(sdf['Innings Runs Scored
Num'])/len(sdf)
```

```
(50.20982142857143, 43.608856088560884)
```

```
# Kohli_df = player runs with Kohli
# player runs excluding Kohli => not_kohli = kohli_df[kohli_df.player_name
!= 'V Kohli']
```

```
# RPI - Sachin, Virat
sum(kohli_df['Innings Runs Scored Num'])/len(kohli_df)
```

```
24.99673202614379
```

```
non_kohli_df = kohli_df[kohli_df['Innings Player'] != 'V Kohli']
```

```
non_sachin_df = sachin_df[sachin_df['Innings Player'] != 'SR Tendulkar']
```

```
# Avg = 25 runs
# Kohli = 50
```

```
(sum(kdf['Innings Runs Scored Num'])/len(kdf))/(sum(non_kohli_df['Innings
Runs Scored Num'])/len(non_kohli_df))
```

```
2.029683688052565
```

```
(sum(sdf['Innings Runs Scored Num'])/len(sdf))/(sum(non_sachin_df['Innings Runs Scored Num'])/len(non_sachin_df))
```

```
1.9437755485945407
```

```
# kohli => other
# SR = V = 93, Other = 80, V/other, S/others
# 100s - Number of matches to score a 100
# 50s - Number of matches to score a 50
# Team contribution - V_cont/O_cont
```

```
200/40, 200/37
```

```
(5.0, 5.405405405405405)
```

Strike Rate:

```
# sr of sachin
sum(sdf['Innings Runs Scored Num'])/sum(sdf['Innings Balls Faced'])
```

```
0.8821377920429947
```

```
# sr of sachin's peers
sum(non_sachin_df['Innings Runs Scored Num'])/sum(non_sachin_df['Innings Balls Faced'])
```

```
0.7233808936558636
```

```
# sr of kohli
sum(kdf['Innings Runs Scored Num'])/sum(kdf['Innings Balls Faced'])
```

```
0.9356126778138258
```

```
# sr of kohli's peers
sum(non_kohli_df['Innings Runs Scored Num'])/sum(non_kohli_df['Innings Balls Faced'])
```

```
0.8342743413330611
```

```
# normalized sachin's value
sachin_sr = sum(sdf['Innings Runs Scored Num'])/sum(sdf['Innings Balls Faced'])
sachin_peer_sr = sum(non_sachin_df['Innings Runs Scored Num'])/sum(non_sachin_df['Innings Balls Faced'])
sachin_sr/sachin_peer_sr
```

```
1.2194651528391862
```

```
# normalized kohli's value
kohli_sr = sum(kdf['Innings Runs Scored Num'])/sum(kdf['Innings Balls Faced'])
kohli_peer_sr = sum(non_kohli_df['Innings Runs Scored Num'])/sum(non_kohli_df['Innings Balls Faced'])
kohli_sr/kohli_peer_sr
```

```
1.121468839996732
```

100's: Number of matches to score a 100

```
# sachin matches per 100
len(sdf)/sum(sdf["100's"])
```

```
7.324324324324325
```

```
# sachin peers - matches per 100
len(non_sachin_df)/sum(non_sachin_df["100's"])
```

```
47.377969762419006
```

```
# kohli matches per 100
len(kdf)/sum(kdf["100's"])
```

```
5.333333333333333
```

```
# kohli peers - matches per 100
len(non_kohli_df)/sum(non_kohli_df["100's"])
```

```
29.311827956989248
```

```
# normalized sachin value
sachin_mper_100 = len(sdf)/sum(sdf["100's"])
sachin_peers_mper_100 = len(non_sachin_df)/sum(non_sachin_df["100's"])
sachin_mper_100/sachin_peers_mper_100
```

```
0.15459346107595562
```

```
# normalized virat value
kohli_mper_100 = len(kdf)/sum(kdf["100's"])
kohli_peers_mper_100 = len(non_kohli_df)/sum(non_kohli_df["100's"])
kohli_mper_100/kohli_peers_mper_100
```

```
0.18195157740278795
```

50's: Number of matches to score a 50

```
# sachin matches per 100
len(sdf)/sum(sdf["50's"])
```

```
4.754385964912281
```

```
# sachin peers - matches per 100
len(non_sachin_df)/sum(non_sachin_df["50's"])
```

```
8.33751425313569
```

```
# kohli matches per 100
len(kdf)/sum(kdf["50's"])
```

```
4.226415094339623
```

```
# kohli peers - matches per 100
len(non_kohli_df)/sum(non_kohli_df["50's"])
```

```
7.673469387755102
```

```
# normalized sachin value
sachin_mper_50 = len(sdf)/sum(sdf["50's"])
```

```
sachin_peers_mper_50 = len(non_sachin_df)/sum(non_sachin_df["50's"])
sachin_mper_50/sachin_peers_mper_50
```

```
0.5702402203539483
```

```
# normalized virat value
kohli_mper_50 = len(kdf)/sum(kdf["50's"])
kohli_peers_mper_50 = len(non_kohli_df)/sum(non_kohli_df["50's"])
kohli_mper_50/kohli_peers_mper_50
```

```
0.5507828181453231
```

Team Contribution: Here we are already comparing with peers, hence no need of a normalization

```
# % of team runs by sachin
100*sum(sdf['Innings Runs Scored Num'])/(sum(non_sachin_df[non_sachin_df.C
ountry == 'India']['Innings Runs Scored Num'])+sum(sdf['Innings Runs Score
d Num']))
```

```
16.951875493078965
```

```
# % of team runs by kohli
100*sum(kdf['Innings Runs Scored Num'])/(sum(non_kohli_df[non_kohli_df.Cou
ntry == 'India']['Innings Runs Scored Num'])+sum(kdf['Innings Runs Scored
Num']))
```

```
17.610033350556627
```

Python code for DHONI vs. STEYN matchup:

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np

import matplotlib.pyplot as plt

#to display all rows columns
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.expand_frame_repr', False)
pd.set_option('max_colwidth', -1)
```

MS DHONI VS. DALE STEYN MATCH-UP

```
# ipl_ball_by_ball_data
```

```
df = pd.read_csv('ipl.csv')
```

```
df.head(1)
```

```
df.innings.unique()
```

```
array([2, 1, 4, 3, 6, 5], dtype=int64)

df = df[(df.innings == 1) | (df.innings == 2)]

df.innings.unique()

array([2, 1], dtype=int64)

#METHODOLOGY:

# What are the numbers when MS DHONI faces DALE STEYN?
# step 1: Filter by player names
# step 2: Use these names & assign it to striker & bowler
# step 3: Get the required columns
# step 4: conclusion

df.bowling_team.unique()

df[df.bowling_team == 'Deccan Chargers']['bowler'].unique()

df[df.batting_team == 'Chennai Super Kings']['striker'].unique()

# player 1:  MS Dhoni
# player 2:  DW Steyn

r_df=df[(df.striker == 'MS Dhoni') & (df.bowler == 'DW Steyn')]

r_df.head(100)

#required variables.

# No of runs scored? strike rate?
# No of balls faced?
# No of times dismissed?

# runs scored
sum(r_df.runs_off_bat)

111

# balls faced
len(r_df)

61

# outs
len(r_df[r_df.player_dismissed == 'MS Dhoni'])

2

# strike rate
100*sum(r_df.runs_off_bat)/len(r_df)

181.9672131147541

# Comparision against all batsman steyn has bowled to
# player 1:    MS Dhoni
# player 2:    DW Steyn

ds_df=df[df.bowler =='DW Steyn']

# runs scored by all batsman vs steyn
bdf1=pd.DataFrame(ds_df.groupby('striker')['runs_off_bat'].sum()).reset_in
dex()

bdf1.head()

bdf2 = pd.DataFrame(ds_df.groupby('striker')['ball'].count()).reset_index(
)

bdf2.head()

bdf3 = bdf1.merge(bdf2, on = 'striker', how = 'left')

bdf3.head(10)

bdf3['strike_rate'] = 100*bdf3['runs_off_bat']/bdf3['ball']

bdf3.head()

# min criteria: 30 balls

bdf3 = bdf3[bdf3.ball >= 30]

bdf3.head()

ms_df=df[df.striker =='MS Dhoni']

# runs scored by this dhoni vs all bowlers
adf1 = pd.DataFrame(ms_df.groupby('bowler')['runs_off_bat'].sum()).reset_i
ndex()

adf1.head()

# balls faced by dhoni
adf2 = pd.DataFrame(ms_df.groupby('bowler')['ball'].count()).reset_index()

adf2.head()

adf3 = adf1.merge(adf2, on = 'bowler', how = 'left')

adf3.head(10)

adf3['strike_rate'] = 100*adf3['runs_off_bat']/adf3['ball']

adf3.head()

# min criteria : 30 balls

adf3 = adf3[adf3.ball >= 30]

# bdf3, adf3

bdf3.reset_index(inplace = True, drop = True)
adf3.reset_index(inplace = True, drop = True)
```

```
bdf3.head()

# adf3.sort_values('strike_rate', ascending = False)

plt.figure(figsize = (16, 8))
plt.scatter(bdf3.strike_rate, bdf3.runs_off_bat)

for i in range(len(bdf3)):
#     plt.text(x, y, text)
    if bdf3.striker[i] == 'MS Dhoni':
        plt.text(bdf3.strike_rate[i] + 2, bdf3.runs_off_bat[i] - 1, bdf3.s
triker[i], color = 'red')
    else:plt.text(bdf3['strike_rate'][i] + 1, bdf3['runs_off_bat'][i] - 1,
bdf3['striker'][i] )

plt.axvline(120, ls = '--', color = 'grey')
plt.axhline(60, ls = '--', color = 'grey')
plt.title('Batsman against DALE STEYN in IPL (min 30 balls faced)', fontsi
ze = 20)
plt.xlabel('Strike rate')
plt.ylabel('Runs scored')
plt.show()



plt.figure(figsize = (16, 8))
plt.scatter(adf3.strike_rate, adf3.runs_off_bat)

for i in range(len(adf3)):
#     plt.text(x, y, text)


    if adf3.bowler[i] == 'DW Steyn':
        plt.text(adf3.strike_rate[i] + 2, adf3.runs_off_bat[i] - 1, adf3.b
owler[i], color = '')
    else:
        plt.text(adf3['strike_rate'][i] + 1, adf3['runs_off_bat'][i] - 1,
adf3['bowler'][i] )
plt.axvline(120, ls = '--', color = 'grey')
plt.axhline(60, ls = '--', color = 'grey')
plt.title('MS DHONI VS. BOWLERS in IPL (min 30 balls faced)', fontsize = 2
0)
plt.xlabel('Strike rate')
plt.ylabel('Runs scored')
plt.show()
```

Python code for DHONI’S performance analysis:

```
df=df[df.season >=2018]

df.head()

msd_df = df[(df.striker == 'MS Dhoni')]

msd_df.head(2)

sum(msd_df.runs_off_bat)

len(msd_df)

sr_msd=(4746/3604)*100

sr_msd

dot=msd_df[msd_df.runs_off_bat==0]

len(dot)

csk_df=df[(df.batting_team == 'Chennai Super Kings')]

#non_dhoni_dataframe of csk
csk_df = csk_df[csk_df['striker'] != 'MS Dhoni']

csk_df.head()

rps_df=df[(df.batting_team == 'Rising Pune Supergiants')]

#non_dhoni_dataframe for rps
rps_df = rps_df[rps_df['striker'] != 'MS Dhoni']

rps_df.head(2)

sum(rps_df.runs_off_bat)+sum(csk_df.runs_off_bat)

#importence of mahi
sum(msd_df.runs_off_bat)

#team contribution of dhoni
(sum(msd_df.runs_off_bat)/(sum(rps_df.runs_off_bat)+sum(csk_df.runs_off_bat)))*100

df.head()

ldf = df[df.ball >= 15.0]

ldf.head()

msd2_df = ldf[(ldf.striker == 'MS Dhoni')]

msd2_df.head()

csk2_df=ldf[(ldf.batting_team == 'Chennai Super Kings')]

#non_dhoni_dataframe for csk in last 5 overs
csk2_df = csk2_df[csk2_df['striker'] != 'MS Dhoni']
```

```
csk2_df.head()

rps2_df=ldf[(ldf.batting_team == 'Rising Pune Supergiants')]

#non_dhoni_dataframe for rps in last 5 overs
rps2_df = rps2_df[rps2_df['striker'] != 'MS Dhoni']

rps2_df.head()

sum(msd2_df.runs_off_bat)

#team contribution of dhoni in last 5 overs

sum(msd2_df.runs_off_bat)/(sum(rps2_df.runs_off_bat)+sum(csk2_df.runs_off_bat))*100

ldf['strike_rate'] = 100*ldf['runs_off_bat']/ldf['ball']

ld=ldf['strike_rate']

ld.sort_values(ascending = False).head()

# balls faced
ldf2 = pd.DataFrame(ldf.groupby('bowler')['ball'].count()).reset_index()

ldf2.head()

ldf.head()

ldf2 = pd.DataFrame(ldf.groupby('striker')['ball'].count()).reset_index()

ldf2.head()

ldf3 = pd.DataFrame(ldf.groupby('striker')['runs_off_bat'].sum()).reset_index()

ldf3.head()

ldf4 = ldf2.merge(ldf3, on = 'striker', how = 'left')

ldf4.head()

ldf4['strike_rate'] = 100*ldf4['runs_off_bat']/ldf4['ball']

ldf4 = ldf4[ldf4.ball >= 250]

ldf4.head(10)

ldf4.groupby(['striker'])['strike_rate'].sum().sort_values(ascending = False).head(10)

ldf4.groupby(['striker'])['strike_rate'].sum().sort_values(ascending = False).head(10).plot(kind = 'barh')
plt.xlabel('Strike rate')
plt.show()

df.head()
```

```
#number of sixes hit by all batsman in last 5 overs
x=6.0

df6 = ldf[ldf.runs_off_bat==x]

df6.head()

df66 = pd.DataFrame(df6.groupby('striker')['runs_off_bat'].sum()).reset_index()

df66.head()

df66['no._of_sixes'] = df66['runs_off_bat']/x

df66['no._of_sixes'] = df66['no._of_sixes'].astype('int')

df66.head()

df66.groupby(['striker'])['no._of_sixes'].sum().sort_values(ascending = False).head(10)

#most no. of sixes in death overs(15 to 20)


#MS Dhoni          164
#AB de Villiers    140
#KA Pollard        137
#RG Sharma         90
#AD Russell        86
#V Kohli           78
```

REFERENCES:

FOR COLLECTING DATA:-

- https://stats.espncricinfo.com/ci/engine/stats/index.html
- Match data - Cricsheet
- https://en.wikipedia.org/

KNOWLEDGE ABOUT CRICKET ANALYSIS

- Cricket 2.0: Inside the T20 Revolution - by Tim Wigmore (Author), Freddie Wilde (Author),Harsha Bhogle (Foreword), Michael Vaughan (Foreword)
- https://www.analyticsvidhya.com/blog/2020/02/network-analysis-ipl-data/

PICTURES OF PLAYERS TAKEN FROM GOOGLE

SOFTWARE USED: PYTHON (JUPYTER NOTEBOOK, GOOGLE COLAB)