INDIAN PREMIER LEAGUE ®

IPL – Auction

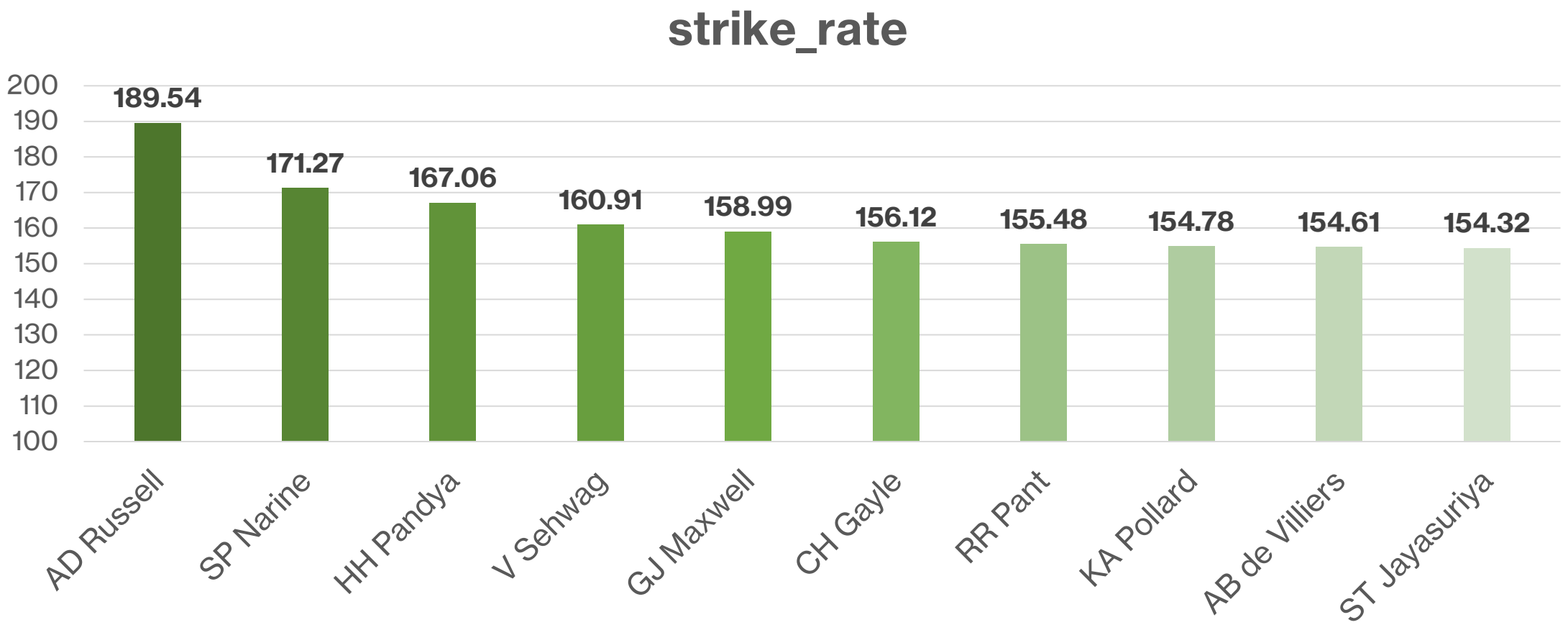
Building auction strategy for new IPL team by analyzing previous IPL data to produce a strong and balanced squad.

Aggressive Batsmen

```sql
SELECT  batsman AS "Aggressive batsmen",
        total_runs,
        balls_faced,
        ROUND(total_runs * 100.0
              / NULLIF(balls_faced, 0), 2) AS strike_rate
FROM ( SELECT
        batsman,
        SUM(CASE WHEN extras_type != 'wides'
            THEN batsman_runs ELSE total_runs END) AS total_runs,
        COUNT(CASE WHEN extras_type != 'wides'
              THEN ball ELSE NULL END) AS balls_faced
       FROM IPL_Ball
       GROUP BY batsman
       HAVING  COUNT(CASE WHEN extras_type != 'wides'
                     THEN ball ELSE NULL END) >= 500)
ORDER BY strike_rate DESC
LIMIT 10;
```

Aggressive batsmen	total_runs	balls_faced	strike_rate
AD Russell	1577	832	189.54
SP Narine	930	543	171.27
HH Pandya	1415	847	167.06
V Sehwag	2824	1755	160.91
GJ Maxwell	1547	973	158.99
CH Gayle	4963	3179	156.12
RR Pant	2127	1368	155.48
KA Pollard	3122	2017	154.78
AB de Villiers	4935	3192	154.61
ST Jayasuriya	821	532	154.32

Aggressive Batsmen

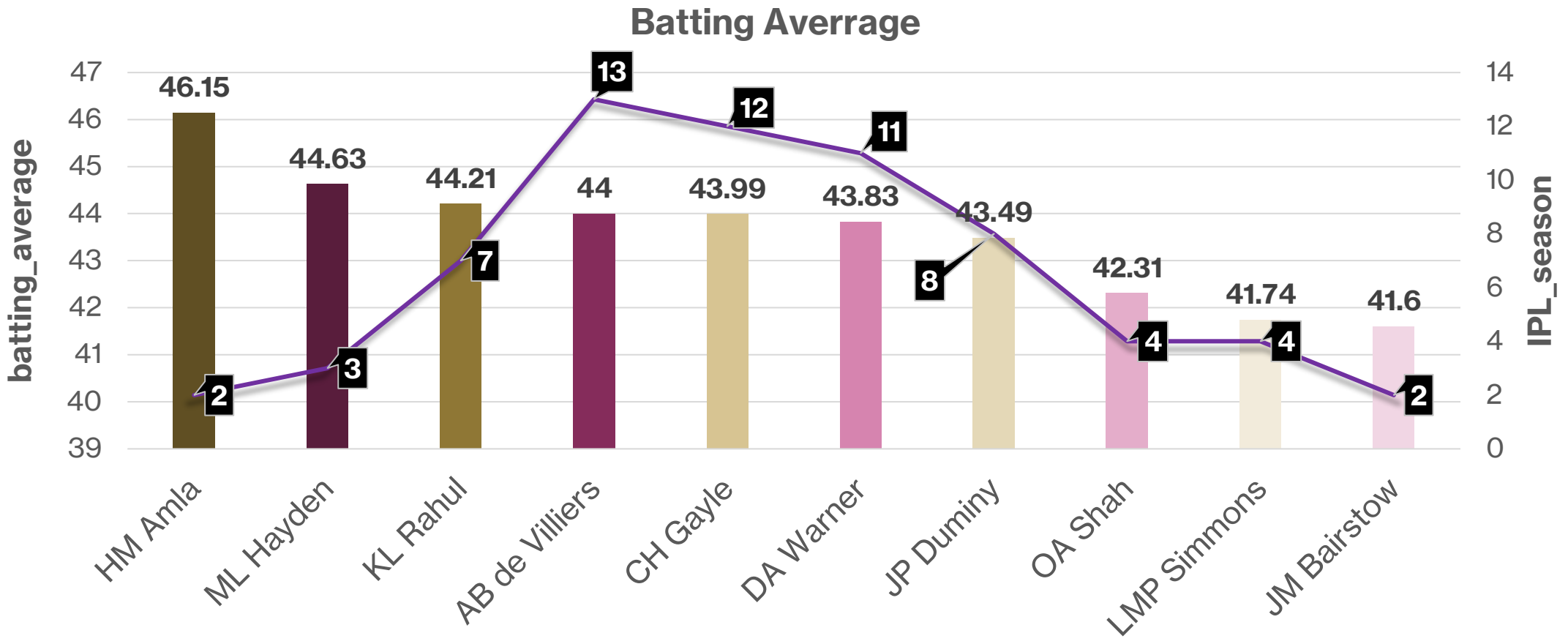


Anchor Batsmen

```
SELECT *,
       CASE WHEN dismissals > 0
       THEN ROUND(CAST(total_runs AS DECIMAL) / dismissals, 2)
       ELSE 0 END AS batting_average
FROM (SELECT batsman AS "Anchor Batsmen",
       COUNT(IPL_year) AS "IPL_season_played",
       SUM("total runs") AS total_runs,
       SUM(dismissals) AS dismissals
      FROM (SELECT ball.batsman,
                   EXTRACT(year FROM match.date) AS IPL_year,
                   SUM(ball.total_runs) AS "total runs",
                   SUM(CASE WHEN ball.is_wicket=1
                       THEN 1 ELSE 0 END) AS dismissals
            FROM ipl_ball AS ball
            LEFT JOIN ipl_matches AS match ON ball.id = match.id
            GROUP BY ball.batsman, EXTRACT(year FROM match.date)
            HAVING SUM(CASE WHEN ball.is_wicket=1
                       THEN 1 ELSE 0 END) > 0) AS subquery
    GROUP BY batsman
    HAVING COUNT(IPL_year) >= 2) AS subquery
ORDER BY batting_average DESC
LIMIT 10;
```

Anchor Batsmen	IPL_season_played	total_runs	dismissals	batting_avg
HM Amla	2	600	13	46.15
ML Hayden	3	1205	27	44.63
KL Rahul	7	2741	62	44.21
AB de Villiers	13	5016	114	44
CH Gayle	12	5103	116	43.99
DA Warner	11	5522	126	43.83
JP Duminy	8	2131	49	43.49
OA Shah	4	550	13	42.31
LMP Simmons	4	1127	27	41.74
JM Bairstow	2	832	20	41.6

Anchor Batsmen



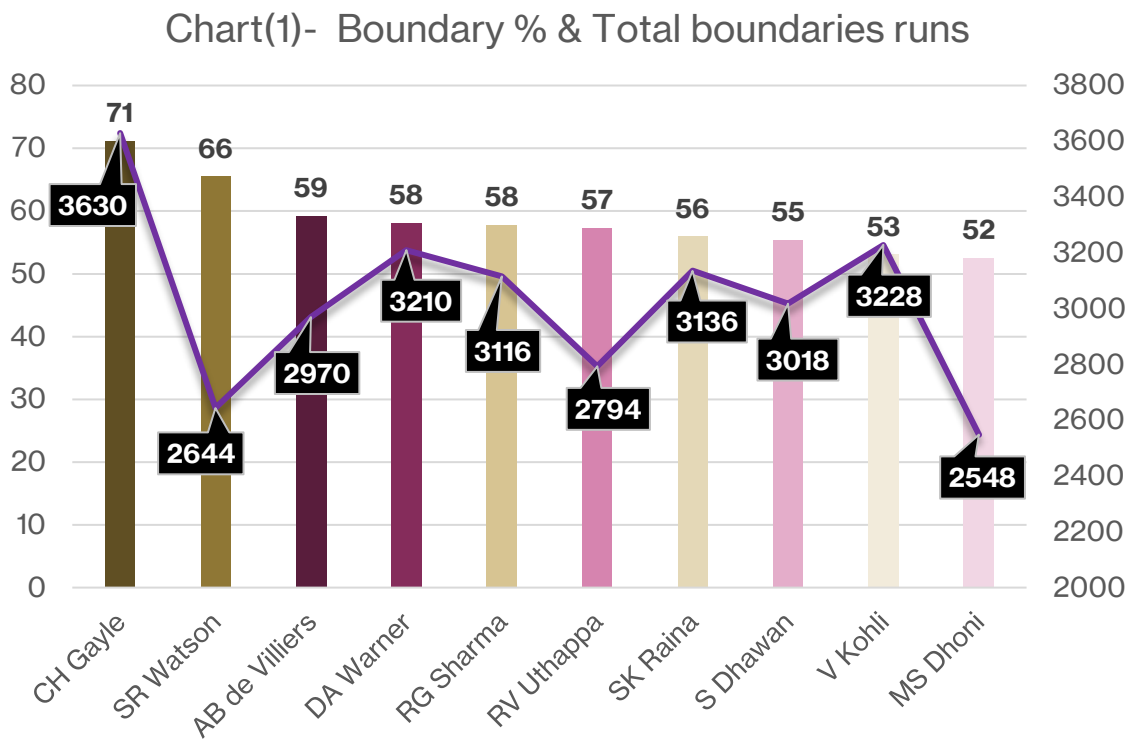
Hard-hitting Player

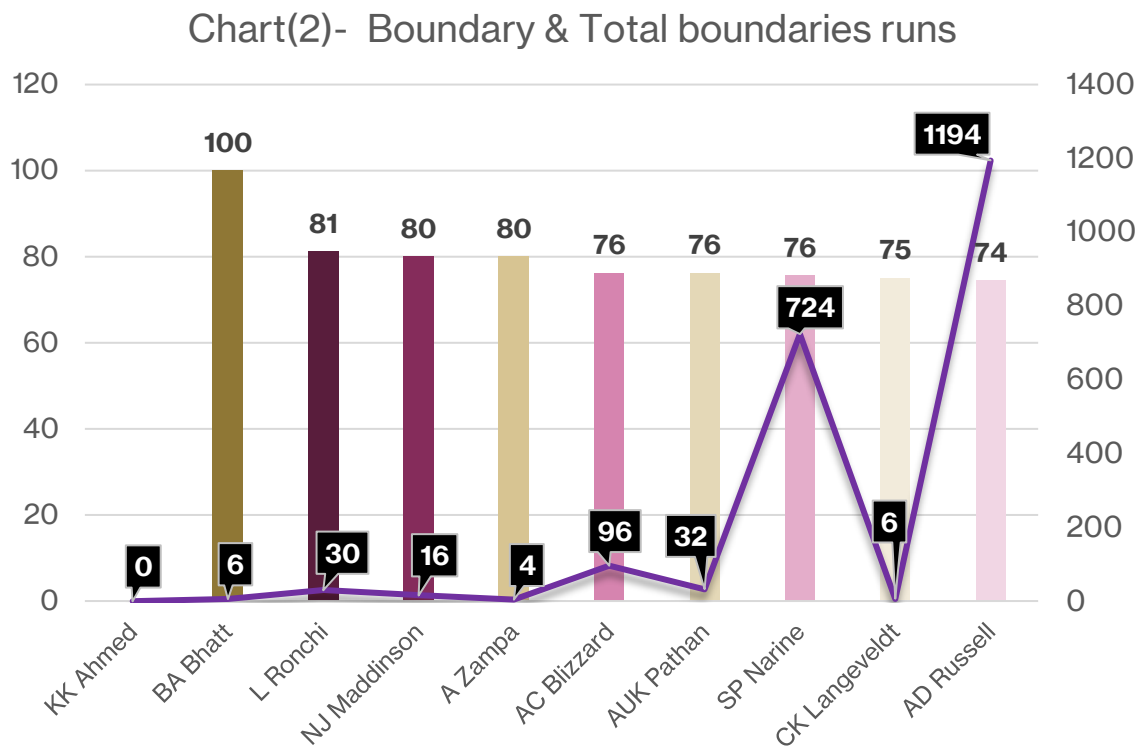
```sql
SELECT batsman AS "Hard-hitting player",
       COUNT(DISTINCT EXTRACT(year FROM match.date))
                AS "IPL seasons played",
       SUM(CASE WHEN ball.batsman_runs IN (4, 6)
           THEN ball.batsman_runs ELSE 0 END)
                AS "Total boundaries runs",
       SUM(ball.total_runs) AS "Total runs",
       ROUND(SUM(CASE WHEN ball.batsman_runs IN (4, 6)
                 THEN ball.batsman_runs ELSE 0 END) * 100.0
             / NULLIF(SUM(ball.total_runs), 0), 2)
                AS "Boundary percentage"
FROM ipl_ball AS ball
LEFT JOIN ipl_matches AS match ON ball.id = match.id
GROUP BY batsman
HAVING COUNT(DISTINCT EXTRACT(year FROM match.date)) >= 2
ORDER BY "Total boundaries runs" DESC
LIMIT 10;
```

Note: To prioritize Hard-hitting players based on the total number of boundary runs rather than the boundary percentage, we'll sort the results by "Total boundaries runs" instead of "Boundary percentage."

Hard-hitting player	seasons played	Total boundaries runs	Total runs	Boundary %
CH Gayle	12	3630	5103	71.13
SR Watson	12	2644	4036	65.51
AB de Villiers	13	2970	5016	59.21
DA Warner	11	3210	5522	58.13
RG Sharma	13	3116	5394	57.77
RV Uthappa	13	2794	4878	57.28
SK Raina	12	3136	5604	55.96
S Dhawan	13	3018	5452	55.36
V Kohli	13	3228	6081	53.08
MS Dhoni	13	2548	4855	52.48

Hard-hitting Player





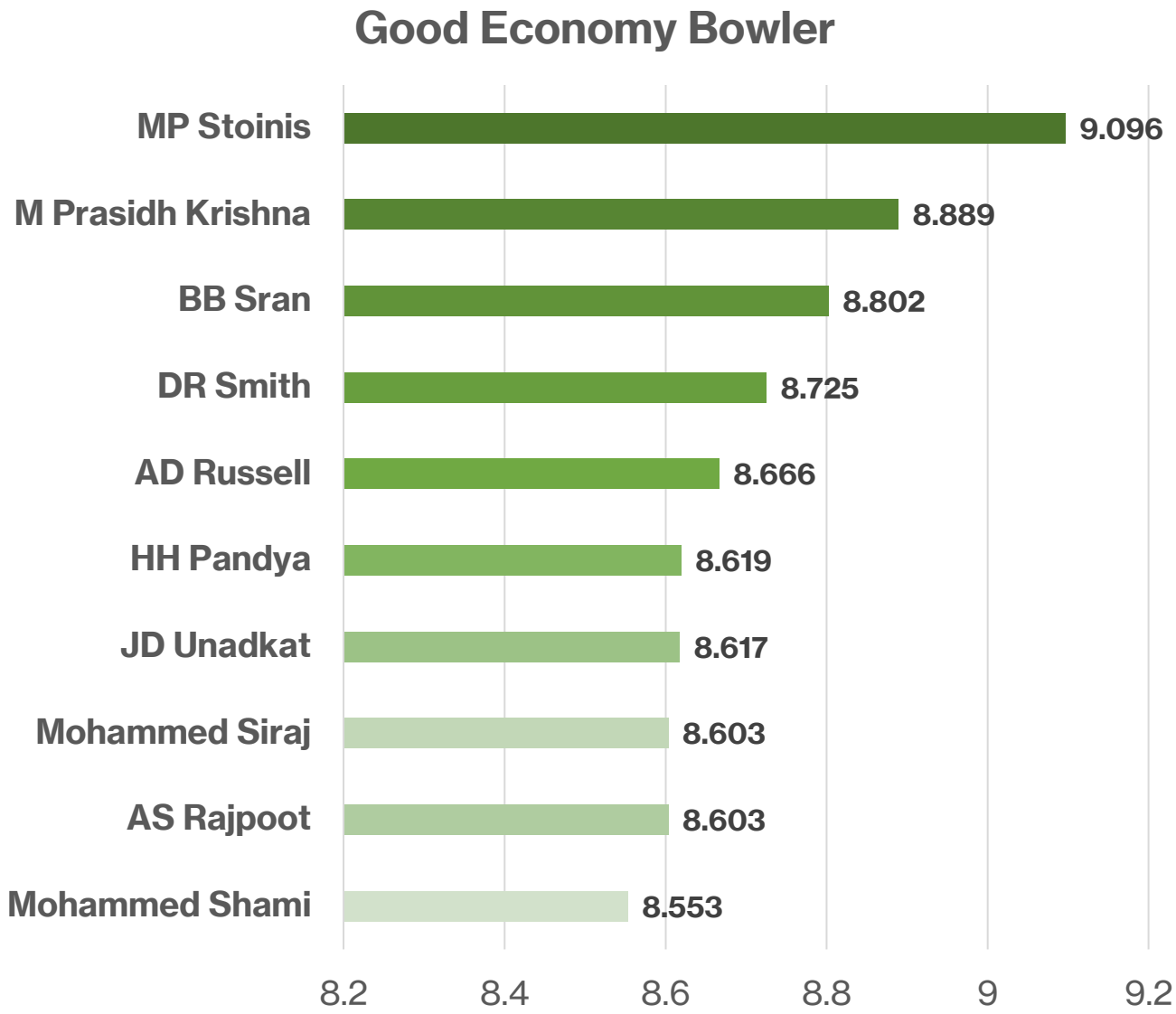
Note: Sort players by "Total boundaries runs" instead of "Boundary percentage" to prioritize hard-hitting players. Total boundary runs reflect a player's ability to make significant contributions to the team's total score. In Chart(2), BA Bhatt has a total of 6 boundary runs out of 6 total runs, his boundary percentage would be 100%. However, this may not accurately represent their hard-hitting ability. By sorting based on total boundary runs, we prioritize players who have scored the most runs in boundaries. In Chart(1), CH Gayle has scored 3630 boundary runs out of 5103 total runs, he would be considered a hard-hitting player. This approach ensures that we select players who have consistently made substantial contributions to their team's score through aggressive batting.

Bowlers with Good Economy

```
SELECT bowler AS "Good Economy Bowler",
       SUM(CASE WHEN extras_type IN ('byes', 'legbyes')
           THEN batsman_runs ELSE total_runs END)
                AS "Total Runs Conceded",
       CEIL(COUNT(bowler) / 6.0) AS "Total Overs Bowled",
       ROUND(SUM(CASE WHEN extras_type IN ('byes', 'legbyes')
                 THEN batsman_runs ELSE total_runs END) /
           (COUNT(bowler) / 6.0),3)  AS "Economy"
FROM ipl_ball
GROUP BY bowler
HAVING COUNT(bowler) >= 500
ORDER BY "Economy" DESC
LIMIT 10;
```

Good Economy Bowler	Total Runs Conceded	Total Overs Bowled	Economy
MP Stoinis	852	94	9.096
M Prasidh Krishna	800	90	8.889
BB Sran	757	86	8.802
DR Smith	810	93	8.725
AD Russell	1713	198	8.666
HH Pandya	1313	153	8.619
JD Unadkat	2420	281	8.617
AS Rajpoot	813	95	8.603
Mohammed Siraj	1084	126	8.603
Mohammed Shami	2007	235	8.553

Bowlers with Good Economy

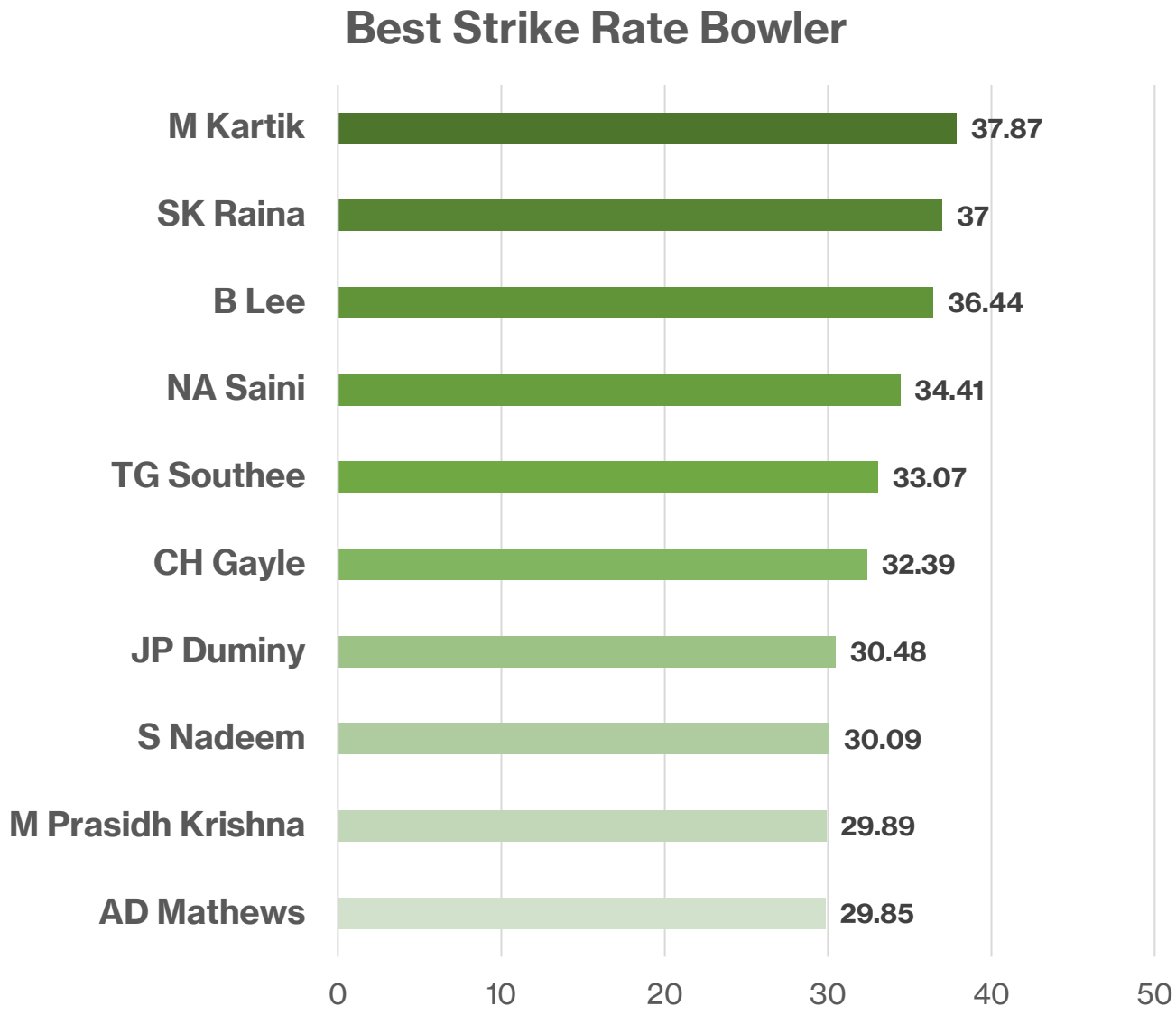


Bowlers with The Best Strike Rate

```
SELECT bowler AS "Best Strike Rate Bowler",
       COUNT(bowler) AS "number of balls bowled",
       SUM(is_wicket) AS "Total Wickets Taken",
       ROUND(COUNT(bowler)*1.00
             /NULLIF(SUM(is_wicket),0),2) AS "Strike Rate"
    FROM ipl_ball
    WHERE NOT dismissal_kind
          IN('run out','retired hurt','obstucting the field')
    GROUP BY bowler
HAVING COUNT(bowler) > 500
ORDER BY "Strike Rate" DESC
LIMIT 10;
```

Best Strike Rate Bowler	number of balls bowled	Total Wickets Taken	Strike Rate
M Kartik	1174	31	37.87
SK Raina	925	25	37
B Lee	911	25	36.44
NA Saini	585	17	34.41
TG Southee	926	28	33.07
CH Gayle	583	18	32.39
JP Duminy	701	23	30.48
S Nadeem	1414	47	30.09
M Prasidh Krishna	538	18	29.89
AD Mathews	806	27	29.85

Bowlers with The Best Strike Rate

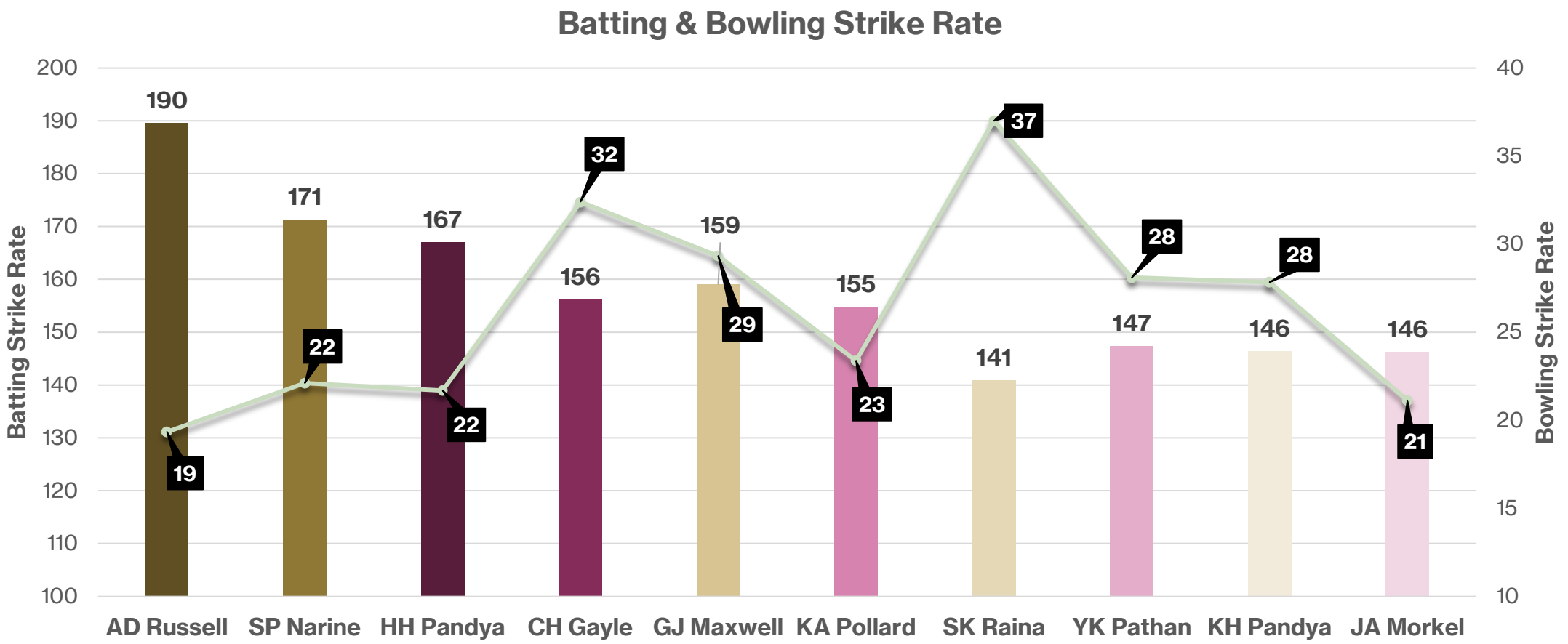


ALL Rounder

```sql
SELECT BattingStats.player AS "All-Rounder",
       BattingStats.balls_faced AS "Balls Faced",
       BattingStats.total_runs AS "Total Runs Scored",
       BattingStats.batting_strike_rate AS "Batting Strike Rate",
       BowlingStats.balls_bowled AS "Balls Bowled",
       BowlingStats.total_wickets AS "Total Wickets Taken",
       BowlingStats.bowling_strike_rate AS "Bowling Strike Rate"
FROM ( SELECT batsman AS player,
              COUNT(CASE WHEN extras_type != 'wides' THEN ball ELSE NULL END) AS balls_faced,
              SUM(CASE WHEN extras_type != 'wides' THEN batsman_runs ELSE total_runs END) AS total_runs,
              ROUND(SUM(CASE WHEN extras_type != 'wides' THEN batsman_runs ELSE total_runs END)* 100.0
                    / NULLIF(COUNT(CASE WHEN extras_type != 'wides' THEN ball ELSE NULL END), 0), 2)
                    AS batting_strike_rate
       FROM ipl_ball
       GROUP BY batsman
       HAVING COUNT(CASE WHEN extras_type != 'wides' THEN ball ELSE NULL END)>500) AS BattingStats
JOIN ( SELECT bowler AS player,
              COUNT(bowler) AS balls_bowled,
              SUM(is_wicket) AS total_wickets,
              ROUND(COUNT(bowler)*1.00/NULLIF(SUM(is_wicket),0),2) AS bowling_strike_rate
       FROM ipl_ball
       WHERE NOT dismissal_kind IN('run out','retired hurt','obstucting the field')
       GROUP BY bowler
       HAVING COUNT(bowler) > 500) AS BowlingStats
ON BattingStats.player = BowlingStats.player
-- Ordering results by the average of batting and bowling strike rates in descending order
ORDER BY (BattingStats.batting_strike_rate + BowlingStats.bowling_strike_rate) / 2 DESC
LIMIT 10;
```

All-Rounder	Balls Faced	Total Runs	Batting SR	Balls Bowled	Wickets Taken	Bowling SR
AD Russell	832	1577	189.54	1180	61	19.34
SP Narine	543	930	171.27	2808	127	22.11
HH Pandya	847	1415	167.06	911	42	21.69
CH Gayle	3179	4963	156.12	583	18	32.39
GJ Maxwell	973	1547	158.99	557	19	29.32
KA Pollard	2017	3122	154.78	1403	60	23.38
SK Raina	3914	5517	140.96	925	25	37
YK Pathan	2241	3302	147.34	1180	42	28.1
KH Pandya	702	1028	146.44	1280	46	27.83
JA Morkel	686	1003	146.21	1796	85	21.13

ALL Rounder


Batting & Bowling Strike Rate
Batting Strike Rate
Bowling Strike Rate
200
190
180
170
160
150
140
130
120
110
100
40
35
30
25
20
15
10
190
171
167
156
159
155
141
147
146
146
19
22
22
32
29
23
37
28
28
21
AD Russell
SP Narine
HH Pandya
CH Gayle
GJ Maxwell
KA Pollard
SK Raina
YK Pathan
KH Pandya
JA Morkel

Wicketkeeper

Based on these **5 criteria**, we can analyze the performance of wicketkeepers across IPL seasons and select the most suitable candidate for the wicketkeeper position in the team.

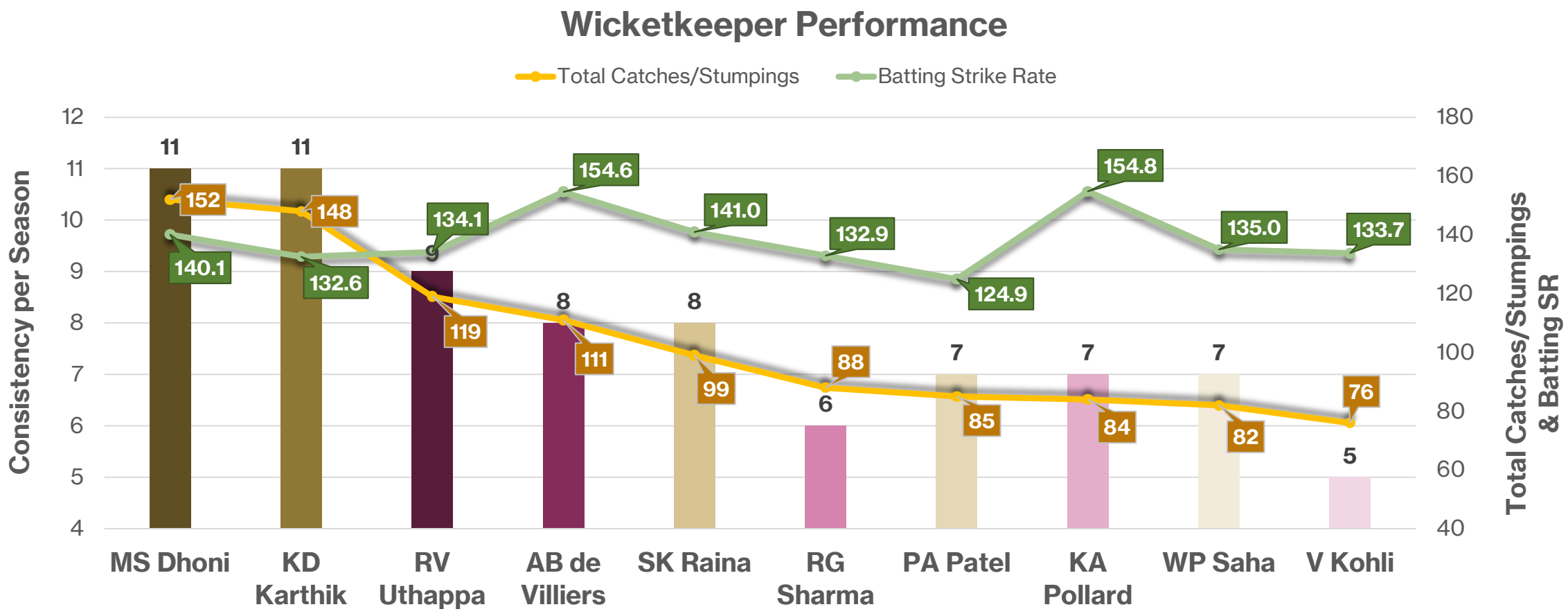
1. **Experience:** The wicketkeeper should have participated in multiple IPL seasons (More than 10) to bring experience to the team
2. **Catches/Stumpings:** The wicketkeeper should have a significant number of catches and stumpings, indicating their agility and proficiency behind the stumps. in fielder column player name is written who took the catch
3. **Runs Scored:** While wicketkeeping, the player should also contribute with the bat, so we can consider their total runs scored in the IPL matches.
4. **Batting Strike Rate:** A good batting strike rate suggests the ability to score quick runs, which is crucial in T20 cricket
5. **Consistency:** Consistency in performance is important, so we can consider the average number of dismissals per Season in dismissal_kind column

Wicketkeeper

```sql
SELECT FieldingStats.player AS "Wicketkeeper",
       FieldingStats.experience AS "Experience",
       FieldingStats.catches_stumpings AS "Total Catches/Stumpings",
       BattingStats.total_runs AS "Total Runs Scored",
       BattingStats.batting_strike_rate AS "Batting Strike Rate",
       ROUND(FieldingStats.catches_stumpings
             / NULLIF(FieldingStats.experience, 0), 2) AS "Consistency per Season"
FROM (
    SELECT B.fielder AS player,
           COUNT(DISTINCT EXTRACT(year FROM m.date)) AS experience,
           COUNT(CASE WHEN B.dismissal_kind IN ('caught', 'stumped') THEN 1 END) AS catches_stumpings
    FROM IPL_Ball b
    JOIN IPL_matches m ON b.id = m.id
    WHERE NOT fielder = 'NA'
    GROUP BY fielder
    HAVING COUNT(DISTINCT EXTRACT(year FROM m.date)) > 10) AS FieldingStats
LEFT JOIN (
    SELECT batsman AS player,
           SUM(CASE WHEN extras_type != 'wides' THEN batsman_runs ELSE total_runs END) AS total_runs,
           ROUND(SUM(CASE WHEN extras_type != 'wides' THEN batsman_runs ELSE total_runs END) * 100.0
                 / NULLIF(COUNT(CASE WHEN extras_type != 'wides'
                                THEN ball ELSE NULL END), 0), 2) AS batting_strike_rate
    FROM IPL_Ball
    GROUP BY batsman
    HAVING COUNT(CASE WHEN extras_type != 'wides' THEN ball ELSE NULL END) > 500) AS BattingStats
ON FieldingStats.player = BattingStats.player
ORDER BY "Total Catches/Stumpings" DESC, "Total Runs Scored" DESC, "Batting Strike Rate" DESC
LIMIT 10;
```

Wicketkeeper	Experience	Catches/ Stumpings	Total Runs Scored	Batting SR	Consistency / Season
MS Dhoni	13	152	4746	140.12	11
KD Karthik	13	148	3909	132.55	11
RV Uthappa	13	119	4753	134.11	9
AB de Villiers	13	111	4935	154.61	8
SK Raina	12	99	5517	140.96	8
RG Sharma	13	88	5321	132.89	6
PA Patel	12	85	2946	124.94	7
KA Pollard	11	84	3122	154.78	7
WP Saha	11	82	2024	135.02	7
V Kohli	13	76	6013	133.74	5

Wicketkeeper



Additional Questions 1

Get the count of cities that have hosted an IPL match

```
SELECT COUNT ( DISTINCT city )
                AS "Count of cities
that have hosted an IPL match"
FROM IPL_matches;
```

Count of cities that have hosted an IPL match
33

Additional Questions 2

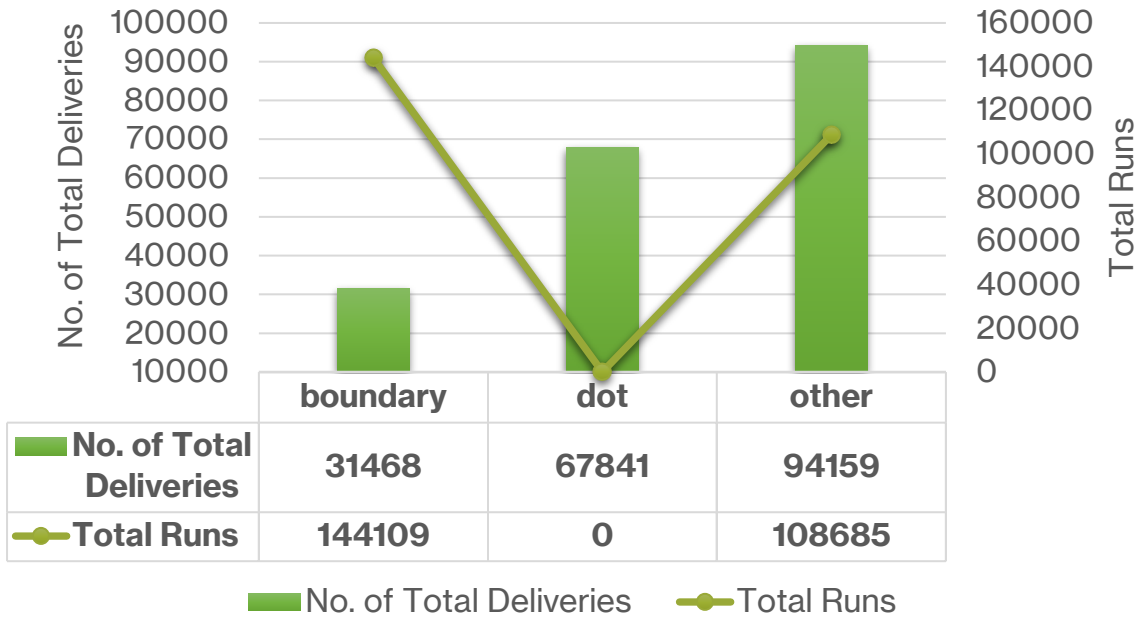
Create table deliveries_v02 with all the columns of the table ‘deliveries’ and an additional column ball_result containing values boundary, dot or other depending on the total_run

```
CREATE TABLE deliveries_v02 AS
SELECT *,
       CASE
           WHEN total_runs >= 4
           THEN 'boundary'
           WHEN total_runs = 0
           THEN 'dot'
           ELSE 'other'
       END AS ball_result
FROM IPL_Ball;
```

Additional Questions 3

Write a query to fetch the total number of boundaries and dot balls from the deliveries_v02 table

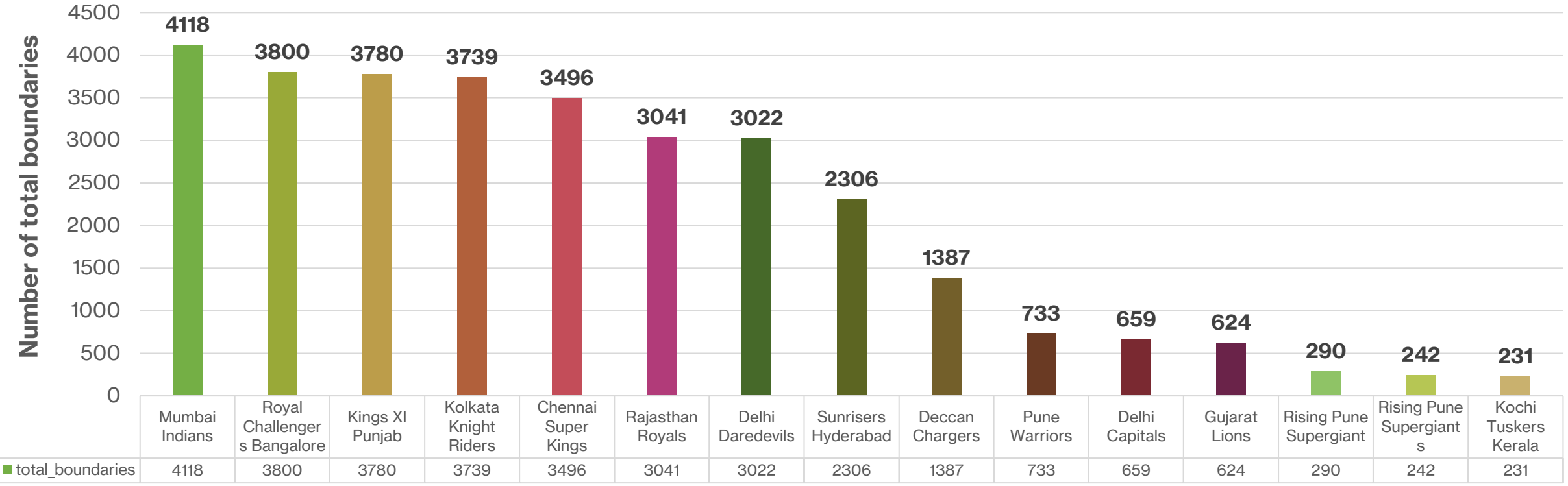
```
SELECT   ball_result,
         COUNT (*)
         AS "No. of Total Deliveries",
         SUM (total_runs) AS "Total Runs"
FROM deliveries_v02
GROUP BY  ball_result;
```



Additional Questions 4

Write a query to fetch the total number of boundaries scored by each team from the deliveries_v02 table and order it in descending order of the number of boundaries scored

```sql
SELECT batting_team,
       COUNT(CASE WHEN ball_result = 'boundary' THEN 1 END)
         AS "No of total_boundaries"
FROM deliveries_v02
GROUP BY batting_team
ORDER BY "No of total_boundaries" DESC;
```

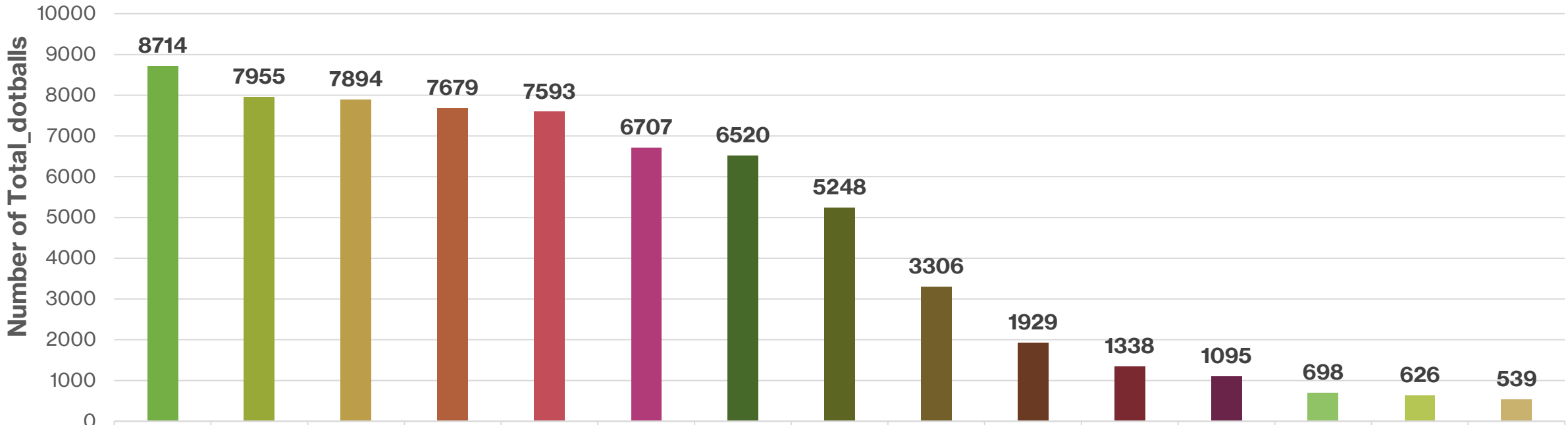


	Mumbai Indians	Royal Challengers Bangalore	Kings XI Punjab	Kolkata Knight Riders	Chennai Super Kings	Rajasthan Royals	Delhi Daredevils	Sunrisers Hyderabad	Deccan Chargers	Pune Warriors	Delhi Capitals	Gujarat Lions	Rising Pune Supergiant	Rising Pune Supergiants	Kochi Tuskers Kerala
total_boundaries	4118	3800	3780	3739	3496	3041	3022	2306	1387	733	659	624	290	242	231

Additional Questions 5

Write a query to fetch the total number of dot balls bowled by each team and order it in
descending order of the total number of dot balls bowled

```
SELECT bowling_team,
       COUNT(CASE WHEN ball_result = 'dot'
                  THEN 1 END) AS total_dotballs
FROM deliveries_v02
GROUP BY bowling_team
ORDER BY total_dotballs DESC;
```

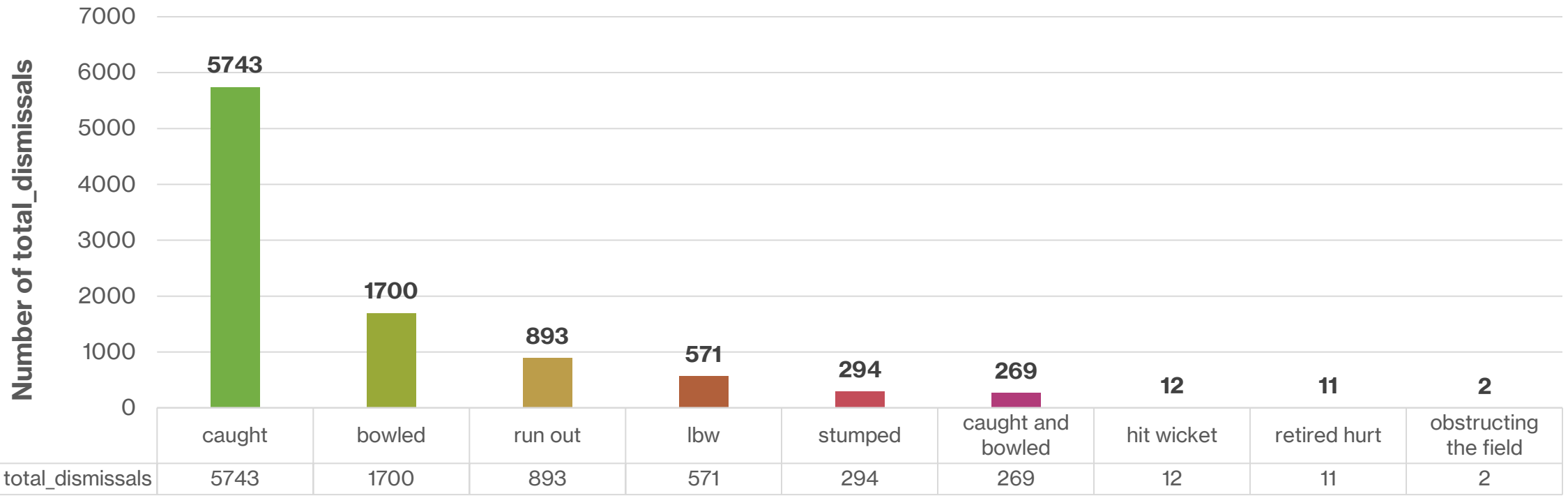


	Mumbai Indians	Royal Challengers Bangalore	Kolkata Knight Riders	Kings XI Punjab	Chennai Super Kings	Rajasthan Royals	Delhi Daredevils	Sunrisers Hyderabad	Deccan Chargers	Pune Warriors	Delhi Capitals	Gujarat Lions	Rising Pune Supergiant	Kochi Tuskers Kerala	Rising Pune Supergiants
total_dotballs	8714	7955	7894	7679	7593	6707	6520	5248	3306	1929	1338	1095	698	626	539

Additional Questions 6

Write a query to fetch the total number of dismissals by dismissal kinds, where dismissal kind is not NA

```
SELECT dismissal_kind,
        COUNT(*) AS total_dismissals
FROM IPL_Ball
WHERE dismissal_kind != 'NA'
GROUP BY dismissal_kind
ORDER BY total_dismissals DESC;
```

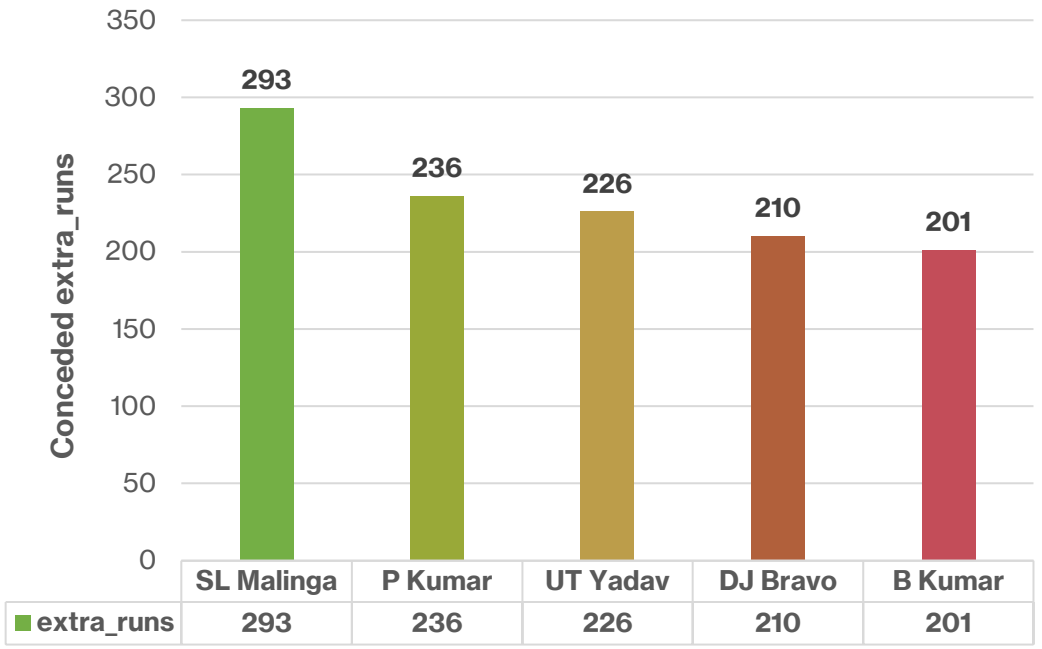


	caught	bowled	run out	lbw	stumped	caught and bowled	hit wicket	retired hurt	obstructing the field
total_dismissals	5743	1700	893	571	294	269	12	11	2

Additional Questions 7

Write a query to get the top 5 bowlers who conceded maximum extra runs from the deliveries table

```
SELECT bowler,
        SUM(extra_runs) AS "Conceded
extra_runs"
        FROM IPL_Ball
        GROUP BY bowler
        ORDER BY SUM(extra_runs) DESC
        LIMIT 5;
```



Additional Questions 8

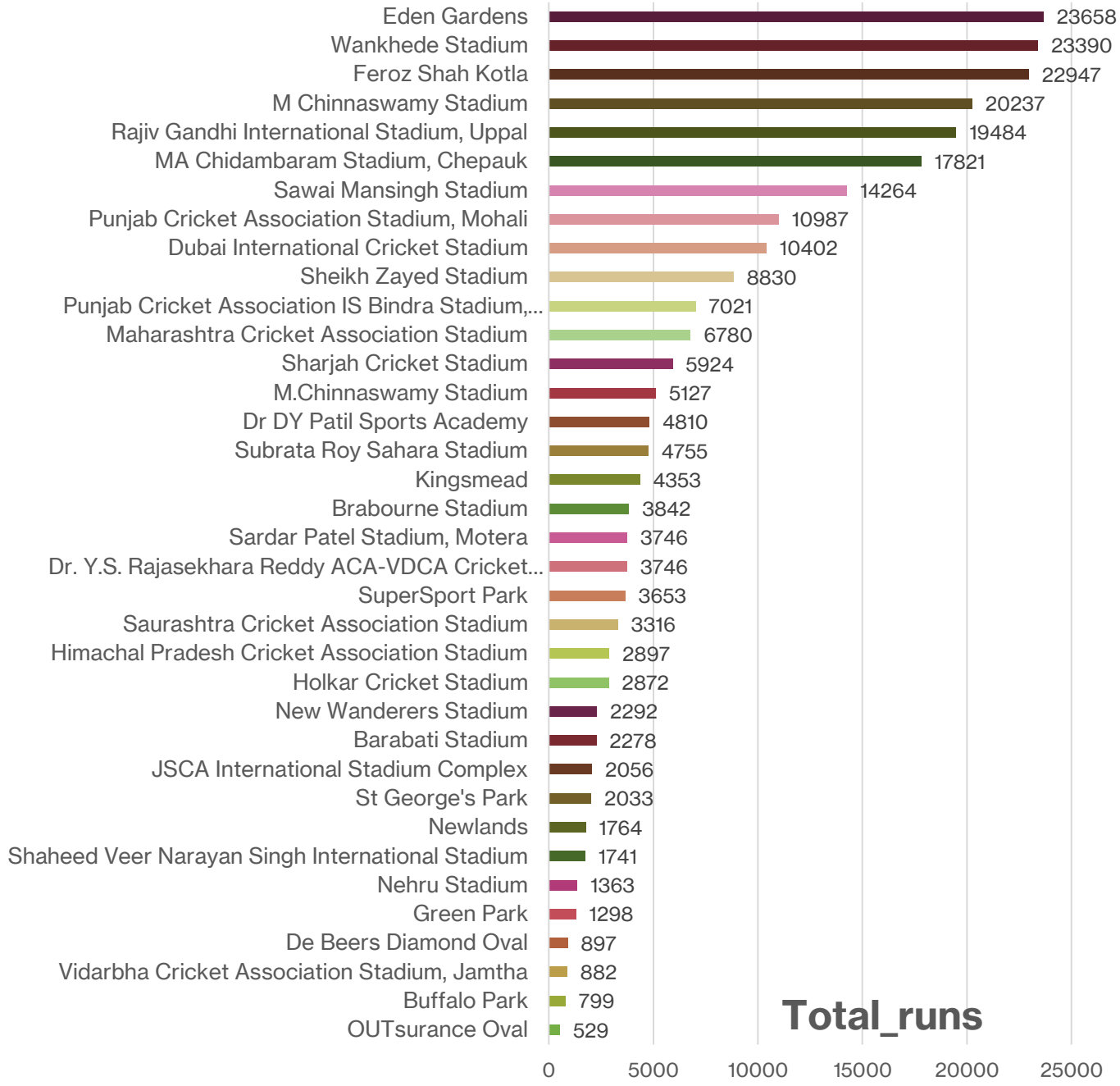
Write a query to create a table named deliveries_v03 with all the columns of deliveries_v02 table and two additional column (named venue and match_date) of venue and date from table matches

```
CREATE TABLE deliveries_v03 AS
SELECT deliveries.*,
        match.venue AS "Venue of the
Match",
        match.date AS "Date of the Match"
        FROM deliveries_v02 AS deliveries
        LEFT JOIN ipl_matches AS match
        ON deliveries.id = match.id;
```

Additional Questions 9

Write a query to fetch the total runs scored for each venue and order it in the descending order of total runs scored.

```
SELECT m.venue AS "Venue of the Match",
       SUM(b.total_runs) AS total_runs
FROM IPL_Ball b
JOIN IPL_matches m ON b.id = m.id
GROUP BY m.venue
ORDER BY total_runs DESC;
```



Additional Questions 10

Write a query to fetch the year-wise total runs scored at Eden Gardens and order it in the descending order of total runs scored.

```
SELECT EXTRACT(year FROM m.date) AS year,
       SUM(b.total_runs) AS total_runs
FROM IPL_Ball b
JOIN IPL_matches m ON b.id = m.id
WHERE m.venue = 'Eden Gardens'
GROUP BY year
ORDER BY total_runs DESC;
```