# Sports Basics Special Edition Magazine on (IPL) 2024 Presentation

**Presented By : Vineet Patyal**

Codebasics Resume Project Challenge #10

# CONTENT

# Problem Statement

"Sports Basics" is a sports blog company that entered space recently. They wanted to get more traffic to their website by releasing a special edition magazine on IPL 2024. This magazine aims to provide interesting insights and facts for fans, analysts and teams based on the last 3 years' data. The chief editor Tony Sharma oversees this publication, and he believes in data analytics. He reached out to Peter Pandey, a journalist in his team who is a data savvy cricket enthusiast.


# Introduction

The 17th IPL season (IPL 2024) runs from March 22nd to May 26th. Ten teams battle it out in Twenty20 matches across India. Chennai Super Kings are the defending champions. Playoffs begin May 21st with the final in Chennai on May 26th.

INDIAN PREMIER LEAGUE

# Team Playing

- Chennai Super Kings
- Delhi Capitals
- Lucknow Super Giants
- Mumbai Indians
- Sunrisers Hyderabad
- Gujarat Titans
- Sunrisers Hyderabad
- Punjab Kings
- Rajasthan Royals
- Royal Challengers Bengaluru

INDIAN PREMIER LEAGUE

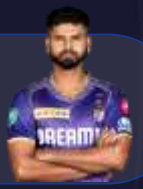
# Team Captions

Ruturaj Gaikwad

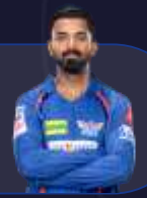
Shubman Gill

Rishabh Pant

Shreyas Iyer

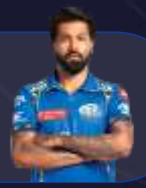


KL Rahul



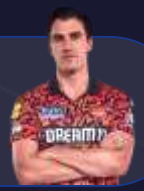
Shikhar Dhawan

Hardik Pandya



Sanju Samson

Pat Cummins



Faf du Plessis

# OBJECTIVE

- Increase traffic to the Sports Basics website through a special edition magazine focusing on IPL 2024.
- Objective is to provide insights and facts for fans, analysts, and teams based on data analytics of the last 3 years' IPL matches.
- Utilize data analytics to extract meaningful insights and trends from the IPL match data spanning the past three years.
- Utilizing soft tools such as Excel & Power Query For Data Transformation, Power BI for Visuals , SQL for Primary Insights & Power point for presentation.
- Using 3 years provided data for Predictions.
- Providing valuable primary Insights such as:

1. Top 10 batsmen based on past 3 years total runs scored.
2. Top 10 batsmen based on past 3 years batting average. (min 60 balls faced in each season)
3. Top 10 batsmen based on past 3 years strike rate (min 60 balls faced in each season)
4. Top 10 bowlers based on past 3 years total wickets taken.
5. Top 10 bowlers based on past 3 years bowling average. (min 60 balls bowled in each season)
6. Top 10 bowlers based on past 3 years economy rate. (min 60 balls bowled in each season)
7. Top 5 batsmen based on past 3 years boundary % (fours and sixes).
8. Top 5 bowlers based on past 3 years dot ball %.

# TOOLS USED

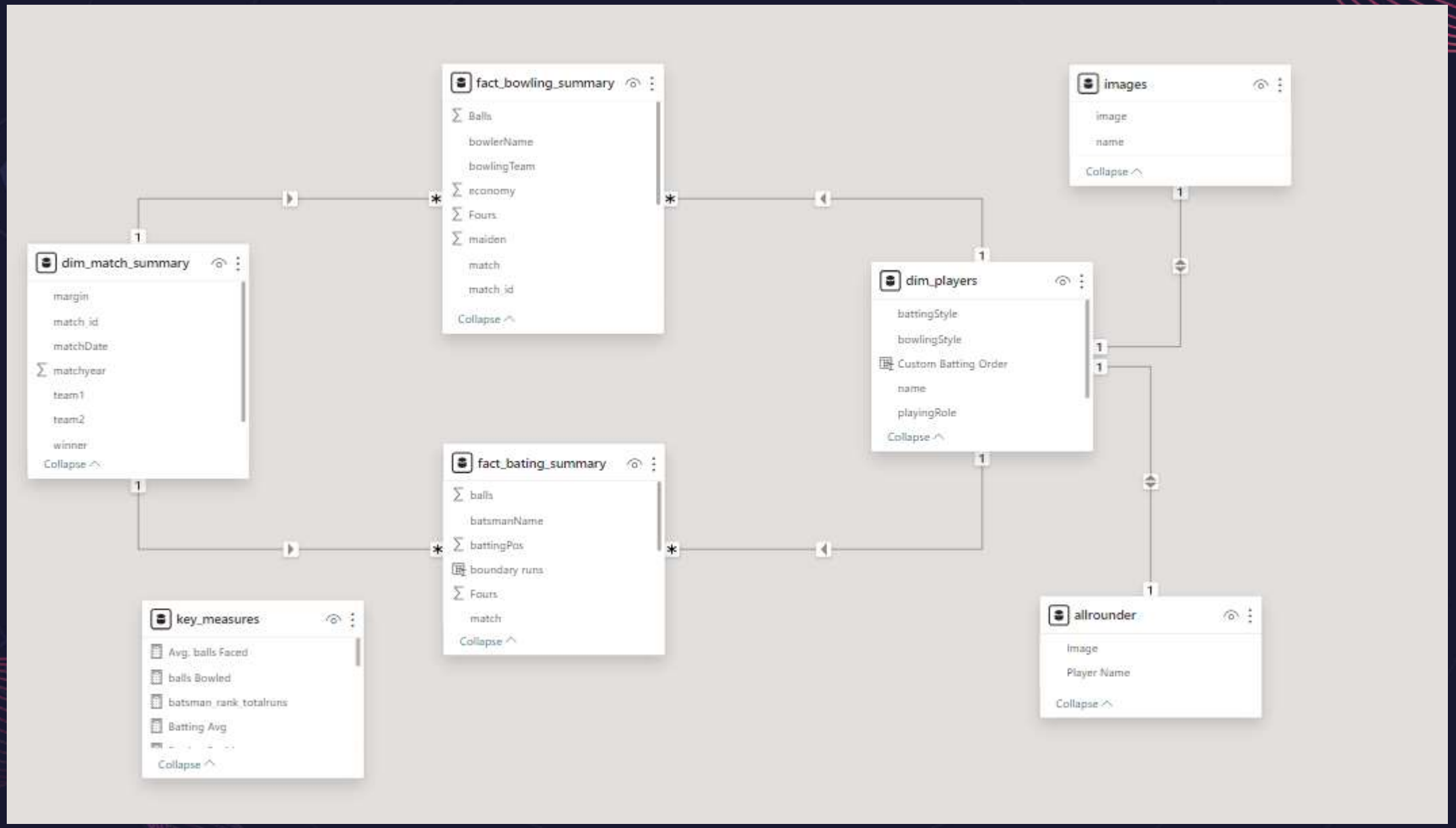
- EXCEL and Power Query used for Data Cleaning and Data Transformation

- Power BI is used for visuals and various predictions mentioned in the presentation


- MySQL language is used to derive primary Insights

- Power Point is used for visual presentation

# ENTITY RELATIONSHIP DIAGRAM



# PRIMARY INSIGHTS

1. Top 10 batsmen based on past 3 years total runs scored.

**SQL Query**

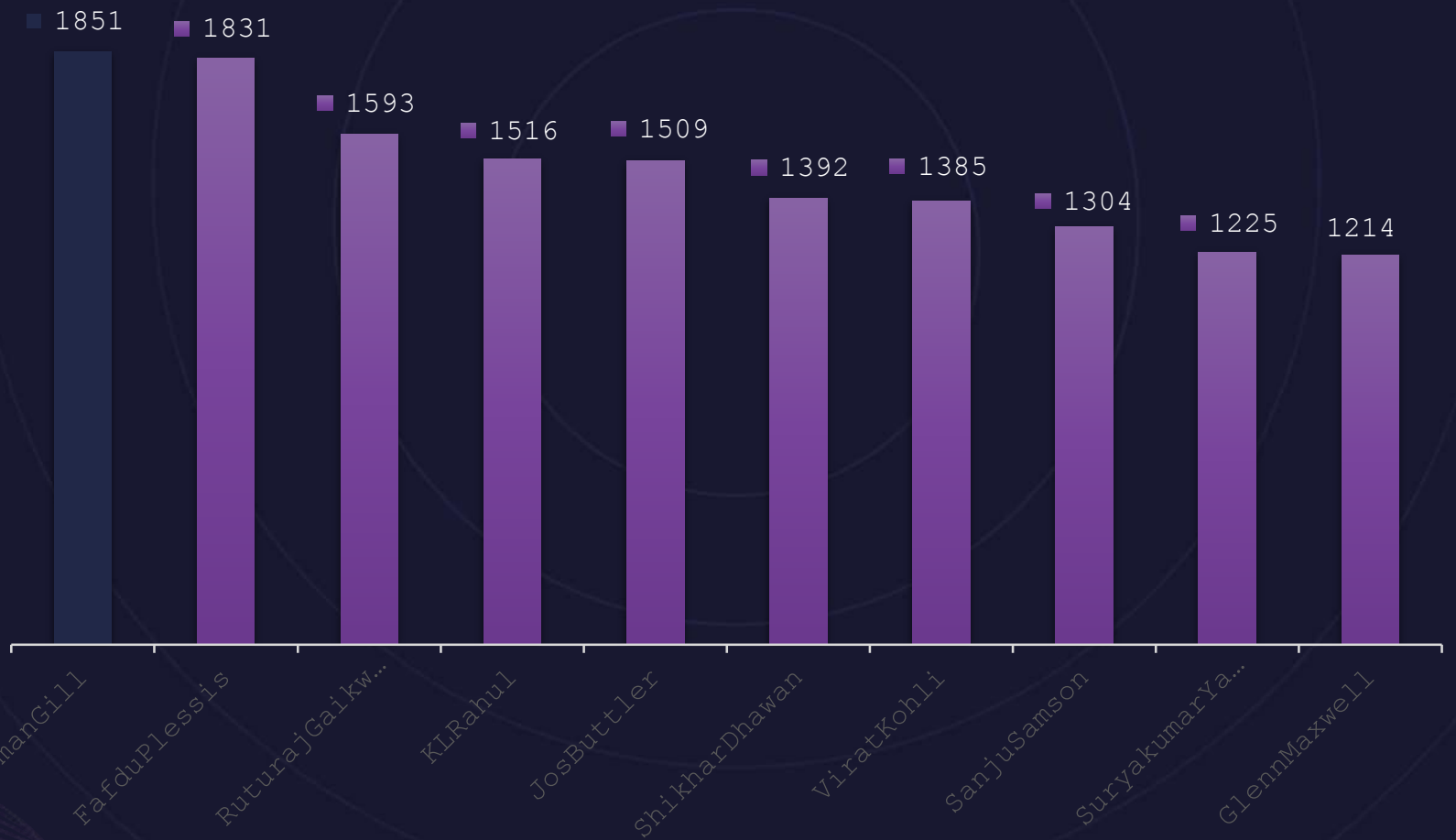
```
SELECT
      batsmanName,
      sum(runs) as
total_runs_scored
FROM
ipl_2024.fact_bating_summ
ary
group by
      batsmanName
order by
      total_runs_scored
desc
      limit 10;
```

**OUTPUT**

| batsmanName | total_runs_scored |
|---|---|
| ShubmanGill | 1851 |
| FafduPlessis | 1831 |
| RuturajGaikwad | 1593 |
| KLRahul | 1516 |
| JosButtler | 1509 |
| ShikharDhawan | 1392 |
| ViratKohli | 1385 |
| SanjuSamson | 1304 |
| SuryakumarYadav | 1225 |
| GlennMaxwell | 1214 |

# Top 10 batsmen based on past 3 years total runs scored.

## VISUALS



## 2. Top 10 batsmen based on past 3 years batting average. (min 60 balls faced in each season)

### SQL Query

```
WITH yearly_stats AS (
SELECT
fs.batsmanName,  dm.matchyear,
SUM(fs.runs) AS total_runs,
SUM(fs.balls) AS total_balls
FROM
fact_bating_summary fs    JOIN   dim_match_summary dm
ON
fs.match_id = dm.match_id
WHERE
dm.matchyear BETWEEN 2021 AND 2023
GROUP BY
fs.batsmanName, dm.matchyear
),
yearly_ball_counts AS (
SELECT   batsmanName,
SUM(CASE WHEN matchyear = 2021 THEN total_balls END)
AS balls_2021,
SUM(CASE WHEN matchyear = 2022 THEN total_balls END)
AS balls_2022,
SUM(CASE WHEN matchyear = 2023 THEN total_balls END)
AS balls_2023
FROM
yearly_stats
GROUP BY
batsmanName
),
qualified_batsmen AS (
SELECT  batsmanName
FROM    yearly_ball_counts
WHERE
balls_2021 >= 60 AND balls_2022 >= 60 AND balls_2023
overall_stats AS (
SELECT   fs.batsmanName,
SUM(fs.runs) AS overall_runs,
SUM(fs.out) AS overall_outs
FROM   fact_bating_summary fs
JOIN   dim_match_summary dm ON fs.match_id =
dm.match_id    WHERE
dm.matchyear BETWEEN 2021 AND 2023   AND
fs.batsmanName IN (SELECT batsmanName FROM
qualified_batsmen)
GROUP BY
fs.batsmanName
),
final_stats AS (
SELECT
os.batsmanName,   os.overall_runs,
os.overall_outs,        round((os.overall_runs /
os.overall_outs),2) AS batting_average    FROM
overall_stats os
)

SELECT
batsmanName,
overall_runs,
overall_outs,
batting_average

FROM
final_stats

ORDER BY batting_average DESCLIMIT 10;
```

CTE1 → •Yearly_stats

CTE2 → •Yearly_ball_counts

CTE3 → •qualified_batsmen

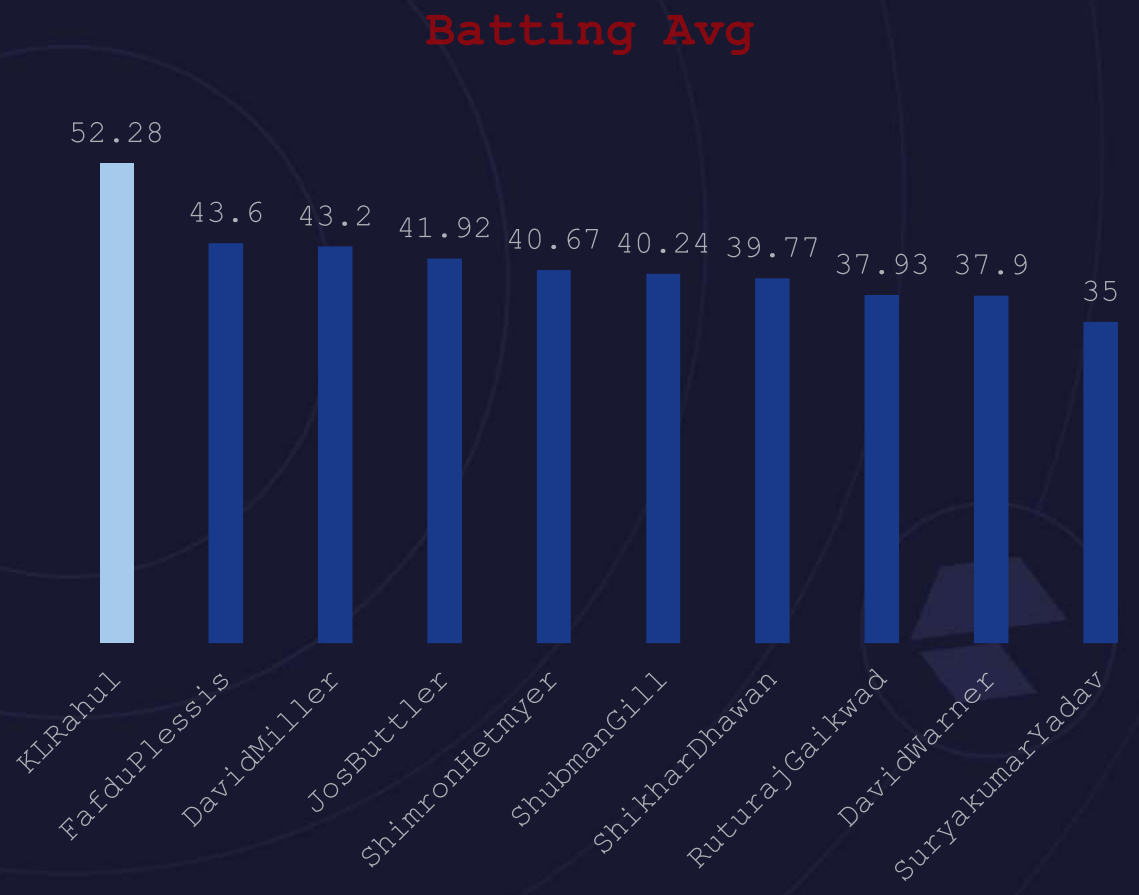
CTE4 → •overall_stats

CTE5 → •final_stats

Common Table Expressions ( CTE )

# VISUAL OUTPUT

| batsmanName | overall_runs | overall_outs | batting_average |
|---|---|---|---|
| KLRahul | 1516 | 29 | 52.28 |
| FafduPlessis | 1831 | 42 | 43.60 |
| DavidMiller | 864 | 20 | 43.20 |
| JosButtler | 1509 | 36 | 41.92 |
| ShimronHetmyer | 854 | 21 | 40.67 |
| ShubmanGill | 1851 | 46 | 40.24 |
| ShikharDhawan | 1392 | 35 | 39.77 |
| RuturajGaikwad | 1593 | 42 | 37.93 |
| DavidWarner | 1137 | 30 | 37.90 |
| SuryakumarYadav | 1225 | 35 | 35.00 |



## 3. Top 10 batsmen based on past 3 years strike rate (min 60 balls faced in each season)

### SQL Query

```
WITH yearly_stats AS (
    SELECT
        fs.batsmanName, dm.matchyear,
        SUM(fs.runs) AS total_runs,
        SUM(fs.balls) AS total_balls
    FROM
        fact_bating_summary fs JOIN dim_match_summary dm
       ON fs.match_id = dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
    GROUP BY
        fs.batsmanName,
        dm.matchyear
),
yearly_ball_counts AS (
    SELECT  batsmanName,
        SUM(CASE WHEN matchyear = 2021 THEN total_balls
END) AS balls_2021,
        SUM(CASE WHEN matchyear = 2022 THEN total_balls
END) AS balls_2022,
        SUM(CASE WHEN matchyear = 2023 THEN total_balls
END) AS balls_2023
    FROM
        yearly_stats
    GROUP BY
        batsmanName
),
qualified_batsmen AS (
SELECT  batsmanName
FROM    yearly_ball_counts
WHERE
balls_2021 >= 60 AND balls_2022 >= 60 AND balls_2023 >=
overall_stats AS (
    SELECT
        fs.batsmanName,
        SUM(fs.runs) AS overall_runs,
        SUM(fs.balls) AS overall_balls
    FROM
        fact_bating_summary fs
    JOIN  dim_match_summary dm ON fs.match_id = dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
        AND fs.batsmanName IN (SELECT batsmanName FROM
qualified_batsmen)
    GROUP BY
        fs.batsmanName
),
final_stats AS (
    SELECT
        os.batsmanName, os.overall_runs, os.overall_balls,
        round((os.overall_runs / os.overall_balls) * 100,2)
AS strike_rate
    FROM
        overall_stats os
)
SELECT
    batsmanName, overall_runs,
    overall_balls, strike_rate
FROM
    final_stats
ORDER BY
    strike_rate DESC
LIMIT 10;
```

CTE1 { •Yearly_stats

CTE2 { •Yearly_ball_counts

CTE3 { •qualified_batsmen

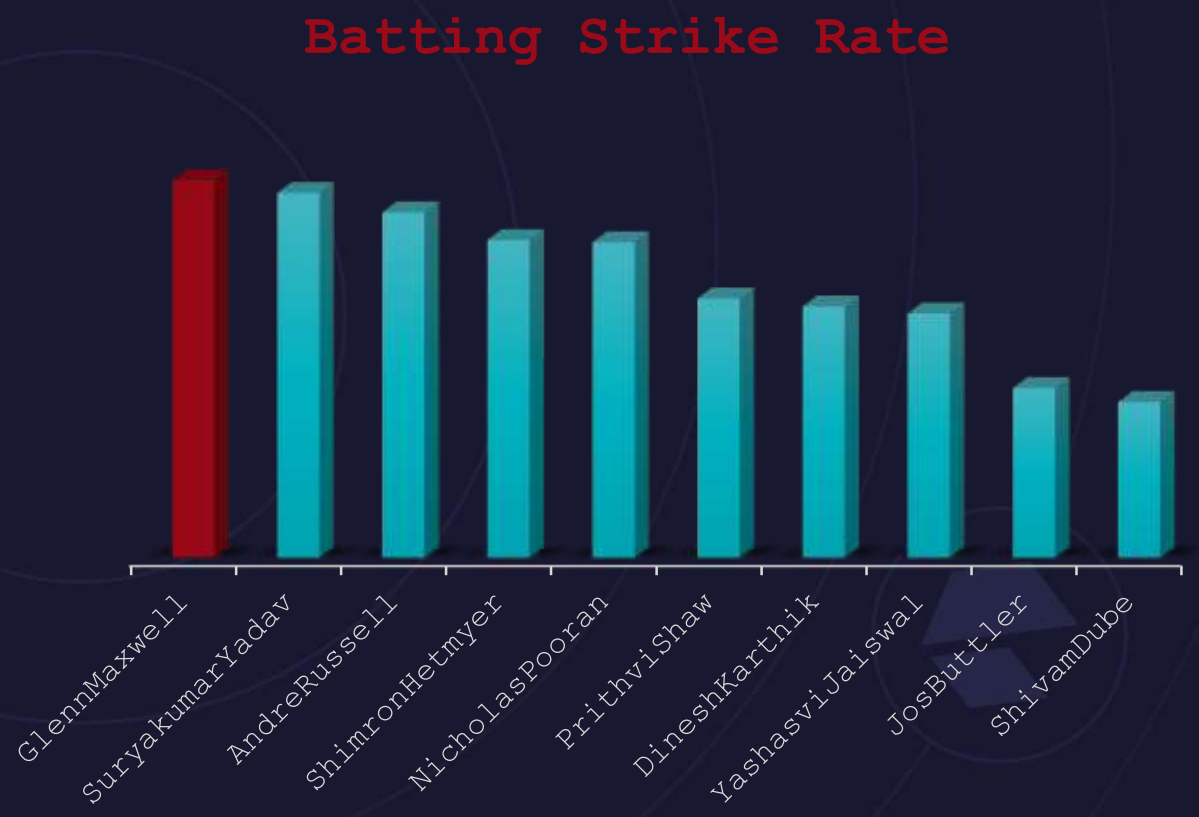
CTE4 { •overall_stats

CTE5 { •final_stats

Common Table Expressions ( CTE )

# VISUAL OUTPUT

| batsmanName | overall_runs | overall_balls | strike_rate |
|---|---|---|---|
| GlennMaxwell | 1214 | 752 | 161.44 |
| SuryakumarYadav | 1225 | 763 | 160.55 |
| AndreRussell | 745 | 468 | 159.19 |
| ShimronHetmyer | 854 | 543 | 157.27 |
| NicholasPooran | 729 | 464 | 157.11 |
| PrithviShaw | 815 | 532 | 153.20 |
| DineshKarthik | 693 | 454 | 152.64 |
| YashasviJaiswal | 1132 | 744 | 152.15 |
| JosButtler | 1509 | 1027 | 146.93 |
| ShivamDube | 937 | 642 | 145.95 |



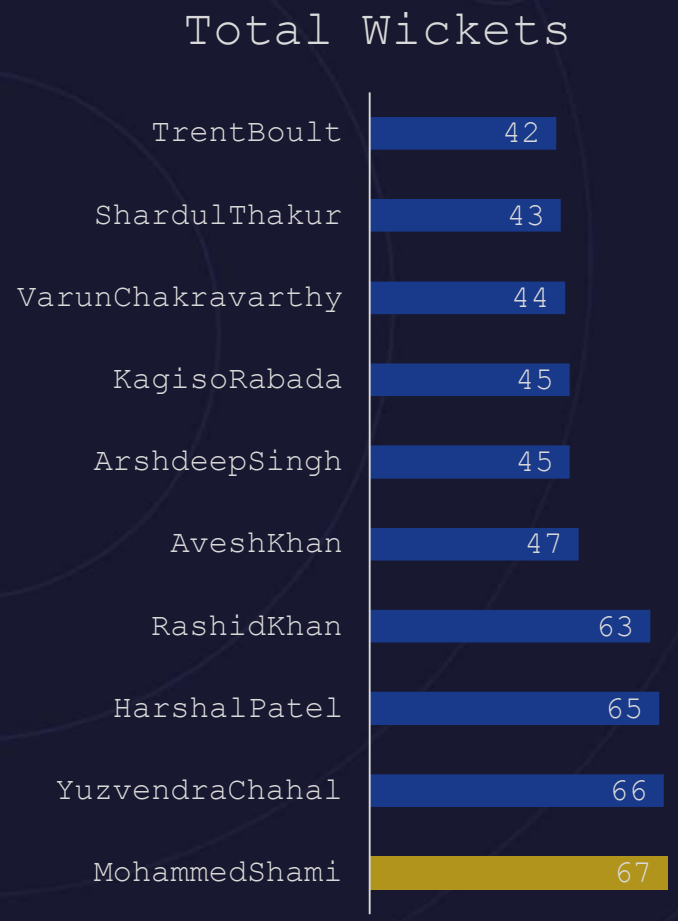
## 4. Top 10 bowlers based on past 3 years total wickets taken.

### SQL Query

```
Select
        bowlername,
        SUM(wickets) as
total_wickets
from

fact_bowling_summary
group by
        bowlerName
order by
       total_wickets
       desc
       Limit 10;
```

### Visual Output



## 5. Top 10 bowlers based on past 3 years bowling average. (min 60 balls bowled in each season)

### Query

```
WITH yearly_stats AS (
    SELECT
        fbs.bowlerName, dm.matchyear,
        SUM(fbs.runs) AS total_runs_conceded, SUM(fbs.balls) AS
total_balls_bowled
    FROM
        fact_bowling_summary fbs
    JOIN
        dim_match_summary dm ON fbs.match_id = dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
    GROUP BY
        fbs.bowlerName, dm.matchyear
),
yearly_balls_bowled AS (
    SELECT
        bowlerName,
        SUM(CASE WHEN matchyear = 2021 THEN total_balls_bowled END)
AS balls_bowled_2021,
        SUM(CASE WHEN matchyear = 2022 THEN total_balls_bowled END)
AS balls_bowled_2022,
        SUM(CASE WHEN matchyear = 2023 THEN total_balls_bowled END)
AS balls_bowled_2023
    FROM
        yearly_stats
    GROUP BY
        bowlerName
),
qualified_bowlers AS (
    SELECT bowlerName
    FROM  yearly_balls_bowled
    WHERE
```

CTE1 { •Yearly_stats

CTE2 { •Yearly_balls_bowled

CTE3 { •qualified_bowlers

```
overall_stats AS (
    SELECT
        fbs.bowlerName,
        SUM(fbs.runs) AS overall_runs_conceded,
        SUM(fbs.wickets) AS overall_wickets_taken
    FROM
        fact_bowling_summary fbs
    JOIN
        dim_match_summary dm ON fbs.match_id =
dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
        AND fbs.bowlerName IN (SELECT bowlerName
    FROM qualified_bowlers)
    GROUP BY fbs.bowlerName
),
final_stats AS (
    SELECT
        os.bowlerName, os.overall_runs_conceded,
os.overall_wickets_taken,
        round(SUM(os.overall_runs_conceded)/
        SUM(os.overall_wickets_taken),2) AS bowling_avg
    FROM
        overall_stats os
    group by
        os.bowlerName
)
SELECT
    bowlerName, overall_runs_conceded,
    overall_wickets_taken, bowling_avg
FROM final_stats
ORDER BY  bowling_avg ASC
```

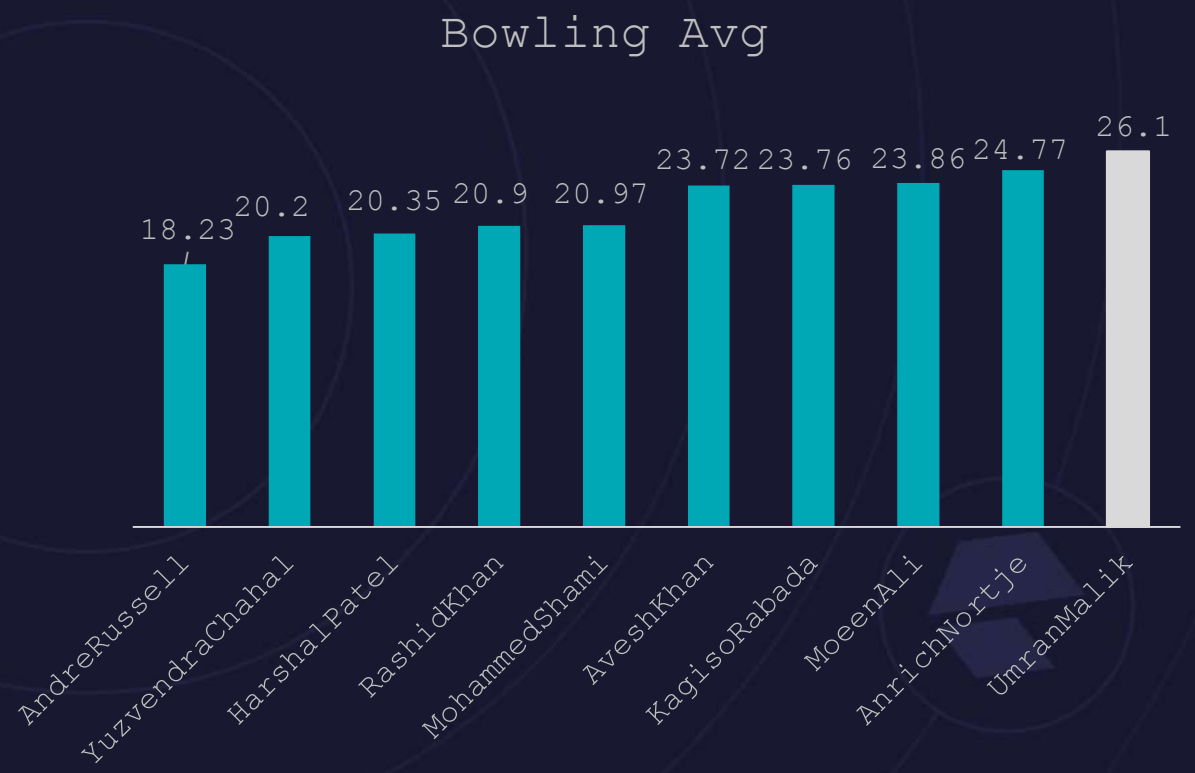
CTE4 { •overall_stats

CTE5 { •final_stats

Common Table Expressions ( CTE )

# VISUAL OUTPUT

| bowlerName | overall_runs_conceded | overall_wickets_taken | bowling_avg |
|---|---|---|---|
| AndreRussell | 638 | 35 | 18.23 |
| YuzvendraChahal | 1333 | 66 | 20.20 |
| HarshalPatel | 1323 | 65 | 20.35 |
| RashidKhan | 1317 | 63 | 20.90 |
| MohammedShami | 1405 | 67 | 20.97 |
| AveshKhan | 1115 | 47 | 23.72 |
| KagisoRabada | 1069 | 45 | 23.76 |
| MoeenAli | 501 | 21 | 23.86 |
| AnrichNortje | 768 | 31 | 24.77 |
| UmranMalik | 757 | 29 | 26.10 |



## 6. Top 10 bowlers based on past 3 years economy rate. (min 60 balls bowled in each season)

### Query

```
WITH yearly_stats AS (
    SELECT
        fbs.bowlerName, dm.matchyear,
        SUM(fbs.runs) AS total_runs_conceded,  SUM(fbs.balls) AS
total_balls_bowled
    FROM
        fact_bowling_summary fbs
    JOIN
        dim_match_summary dm ON fbs.match_id = dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
    GROUP BY
        fbs.bowlerName, dm.matchyear
),
yearly_balls_bowled AS (
    SELECT
        bowlerName,
        SUM(CASE WHEN matchyear = 2021 THEN total_balls_bowled END)
AS balls_bowled_2021,
        SUM(CASE WHEN matchyear = 2022 THEN total_balls_bowled END)
AS balls_bowled_2022,
        SUM(CASE WHEN matchyear = 2023 THEN total_balls_bowled END)
AS balls_bowled_2023
    FROM
        yearly_stats
    GROUP BY
        bowlerName
),
qualified_bowlers AS (
    SELECT  bowlerName
    FROM
        yearly_balls_bowled
overall_stats AS (
    SELECT
        fbs.bowlerName, SUM(fbs.runs) AS
overall_runs_conceded,
        SUM(fbs.balls) AS overall_balls_bowled
    FROM
        fact_bowling_summary fbs
    JOIN
        dim_match_summary dm ON fbs.match_id =
dm.match_id
    WHERE
        dm.matchyear BETWEEN 2021 AND 2023
        AND fbs.bowlerName IN (SELECT bowlerName FROM
qualified_bowlers)
    GROUP BY
        fbs.bowlerName
),
final_stats AS (
    SELECT
        os.bowlerName, os.overall_runs_conceded,
os.overall_balls_bowled,
        round(SUM(os.overall_runs_conceded)/
        SUM(os.overall_balls_bowled/6),2) AS
economy_rate
    FROM   overall_stats os
    Group by    os.bowlerName
)
SELECT
    bowlerName, overall_runs_conceded,
overall_balls_bowled,
    round(sum(overall_balls_bowled/6)) as overs_bowled,
economy_rate
```

CTE1 { •Yearly_stats

CTE2 { •Yearly_balls_bowled

CTE3 { •qualified_bowlers

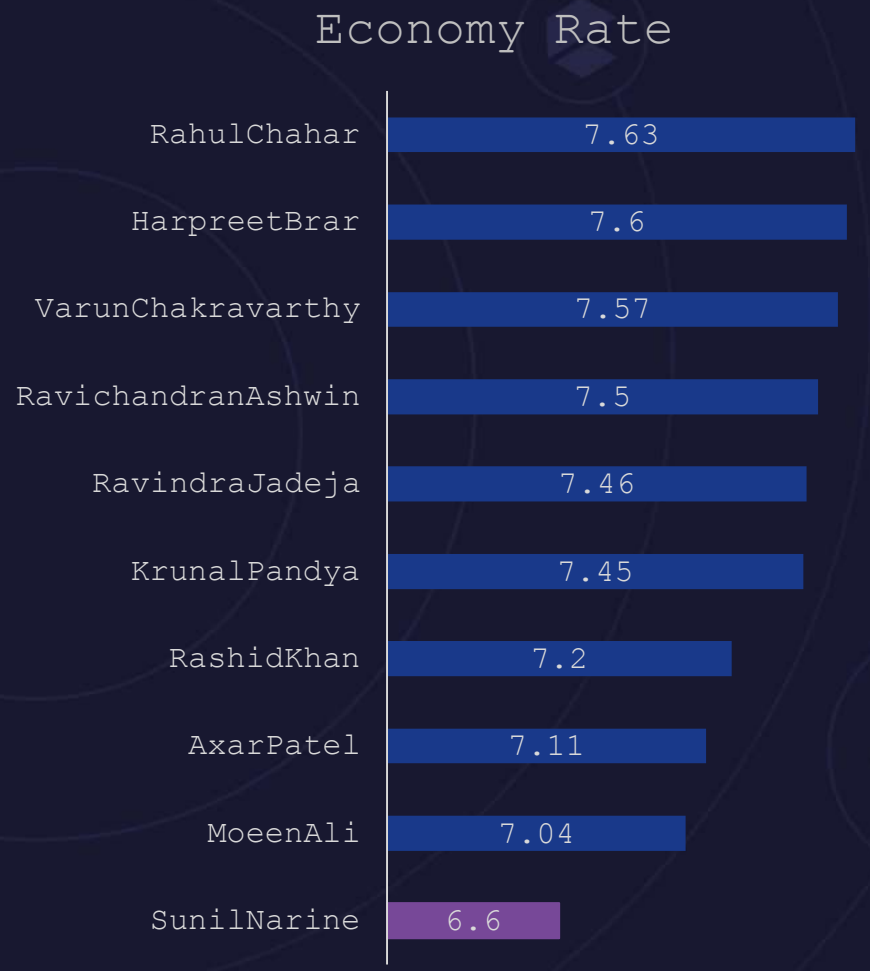
CTE4 { •overall_stats

CTE5 { •final_stats

Common Table Expressions ( CTE )

# Visual Outputs

| bowlerName | overall_runs_conceded | overall_balls_bowled | overs_bowled | economy_rate |
|---|---|---|---|---|
| SunilNarine | 1056 | 960 | 160 | 6.60 |
| MoeenAli | 501 | 427 | 71 | 7.04 |
| AxarPatel | 939 | 792 | 132 | 7.11 |
| RashidKhan | 1317 | 1097 | 183 | 7.20 |
| KrunalPandya | 843 | 679 | 113 | 7.45 |
| RavindraJadeja | 1014 | 816 | 136 | 7.46 |
| RavichandranAshwin | 1175 | 940 | 157 | 7.50 |
| VarunChakravarthy | 1209 | 958 | 160 | 7.57 |
| HarpreetBrar | 503 | 397 | 66 | 7.60 |
| RahulChahar | 1063 | 836 | 139 | 7.63 |



# 7. Top 5 batsmen based on past 3 years boundary % (fours and sixes).

## SQL Query

```
WITH boundary_stats AS (
    SELECT
        fbs.batsmanName,
        SUM(fbs.Fours * 4 + fbs.Sixes * 6) AS boundary_runs, --
Total boundary runs
        SUM(fbs.runs) AS total_runs -- Total runs scored
    FROM
        fact_bating_summary fbs
    INNER JOIN
        dim_match_summary dms ON fbs.match_id = dms.match_id
    WHERE
        dms.matchyear BETWEEN YEAR(CURDATE()) - 3 AND
YEAR(CURDATE()) - 1
    GROUP BY
        fbs.batsmanName
),
boundary_percentage AS (
    SELECT
        batsmanName,
        boundary_runs,
        total_runs,
        round((sum(boundary_runs) / sum(total_runs) * 100),2) AS
boundary_percent -- Calculating boundary percentage
    FROM
        boundary_stats
    WHERE
        total_runs > 500
    Group by
        batsmanName
)
SELECT
    batsmanName,
    boundary_runs,
    total_runs,
    boundary_percent
FROM
    boundary_percentage
ORDER BY
    boundary_percent DESC
LIMIT 5;
```

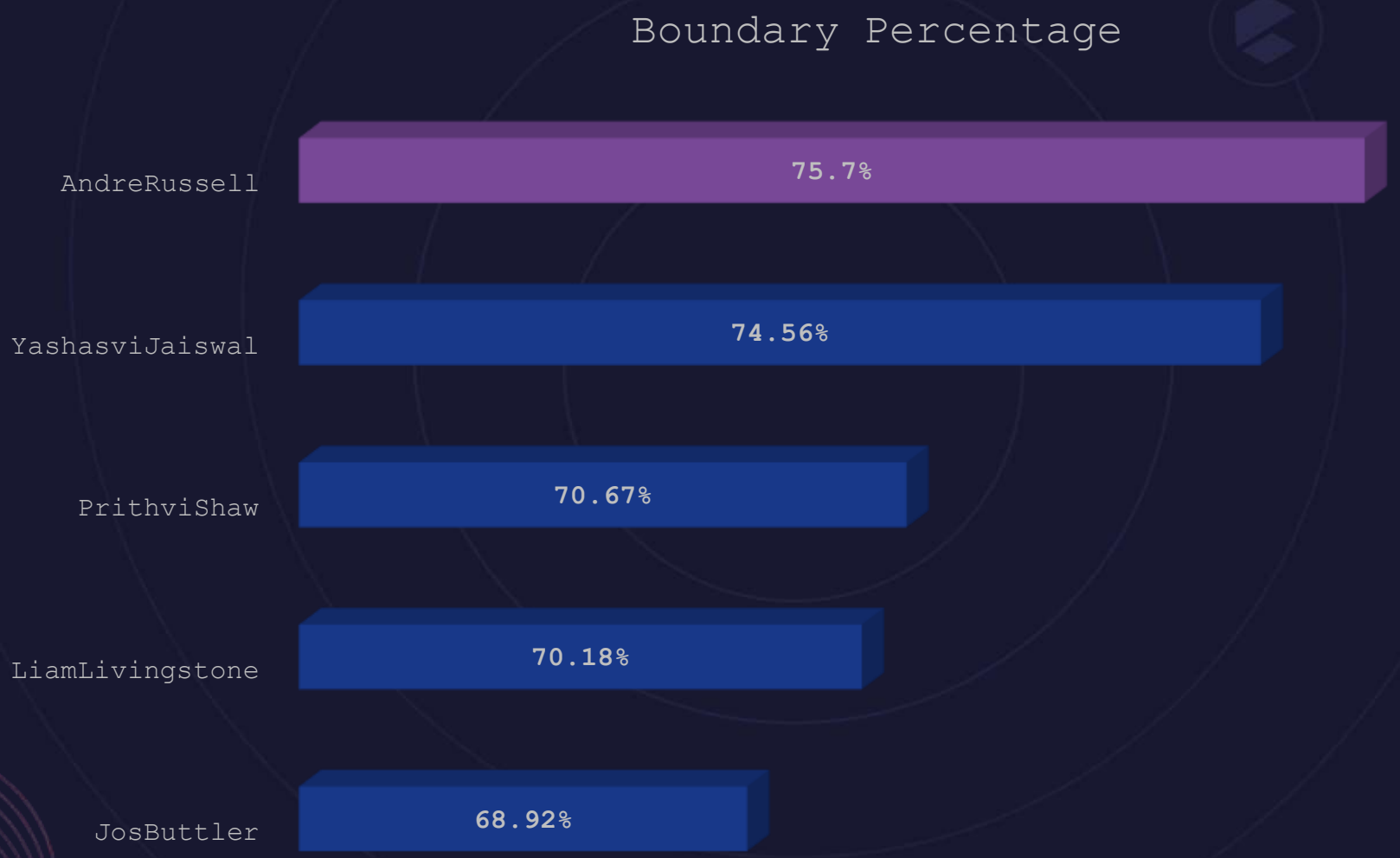
CTE1 { •boundary_stats

CTE2 { •Boundary_percentage

## Query Output

| batsmanName | boundary_runs | total_runs | boundary_percent |
|---|---|---|---|
| AndreRussell | 564 | 745 | 75.70 |
| YashasviJaiswal | 844 | 1132 | 74.56 |
| PrithviShaw | 576 | 815 | 70.67 |
| LiamLivingstone | 532 | 758 | 70.18 |
| JosButtler | 1040 | 1509 | 68.92 |

# Visual Output



# 8. Top 5 bowlers based on past 3 years dot ball %.

## SQL Query

```
WITH yearly_performance AS (
    SELECT
        dms.matchyear,fbs.bowlerName,
        SUM(fbs.Zeros) AS total_dot_balls, SUM(fbs.balls) AS
total_balls
    FROM
        fact_bowling_summary fbs
    JOIN  dim_match_summary dms ON fbs.match_id = dms.match_id
    WHERE
        dms.matchyear IN (2021, 2022, 2023)
    GROUP BY
        dms.matchyear, fbs.bowlerName
),
bowlers_active_all_years AS (
    SELECT
        bowlerName
    FROM
        yearly_performance
    GROUP BY  bowlerName
    HAVING
        COUNT(DISTINCT matchyear) = 3
),
overall_performance AS (
    SELECT
        yp.bowlerName, SUM(yp.total_dot_balls) AS total_dot_balls,
        SUM(yp.total_balls) AS total_balls
    FROM
        yearly_performance yp
    WHERE
        yp.bowlerName IN (SELECT bowlerName FROM
bowlers_active_all_years)
    GROUP BY yp.bowlerName
),
dot_ball_percentage AS (
    SELECT
        op.bowlerName, op.total_dot_balls,
op.total_balls,
        round((op.total_dot_balls / op.total_balls)*
100,2) AS dotball_percent
    FROM
        overall_performance op
)
SELECT
    dbp.bowlerName, dbp.total_dot_balls,
    dbp.total_balls, dbp.dotball_percent
FROM
    dot_ball_percentage dbp
ORDER BY
    dbp.dotball_percent DESC
LIMIT 5;
```

CTE1 { •Yearly_performance

CTE2 { •Bowlers_active_all_years

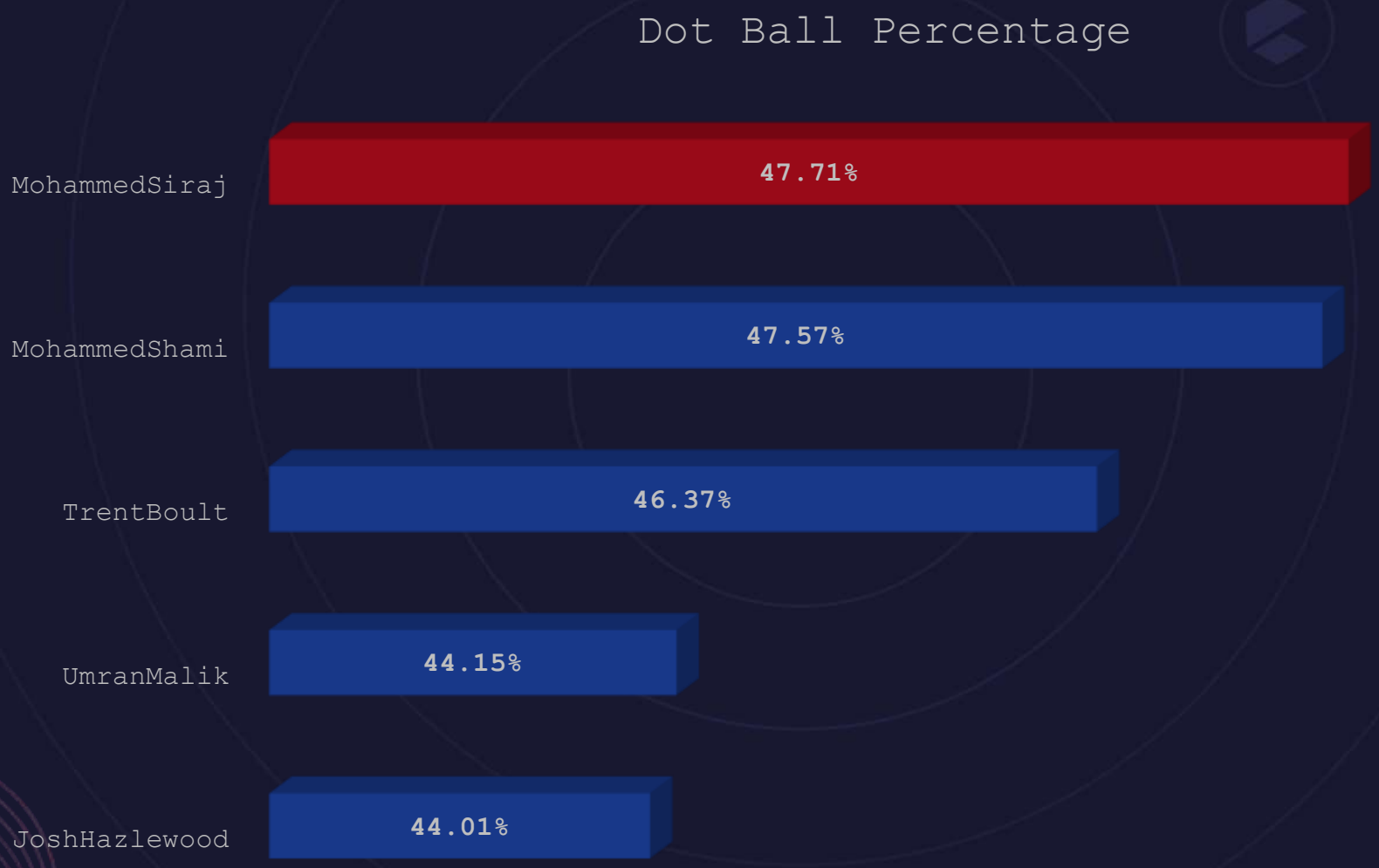
CTE2 { •Overall_performance

CTE2 { •dot_ball_percentage

## Query Output

| bowlerName | total_dot_balls | total_balls | dotball_percent |
|---|---|---|---|
| MohammedSiraj | 438 | 918 | 47.71 |
| MohammedShami | 510 | 1072 | 47.57 |
| TrentBoult | 421 | 908 | 46.37 |
| UmranMalik | 215 | 487 | 44.15 |
| JoshHazlewood | 239 | 543 | 44.01 |

# Visual Output



# 9. Top 4 teams based on past 3 years winning %.

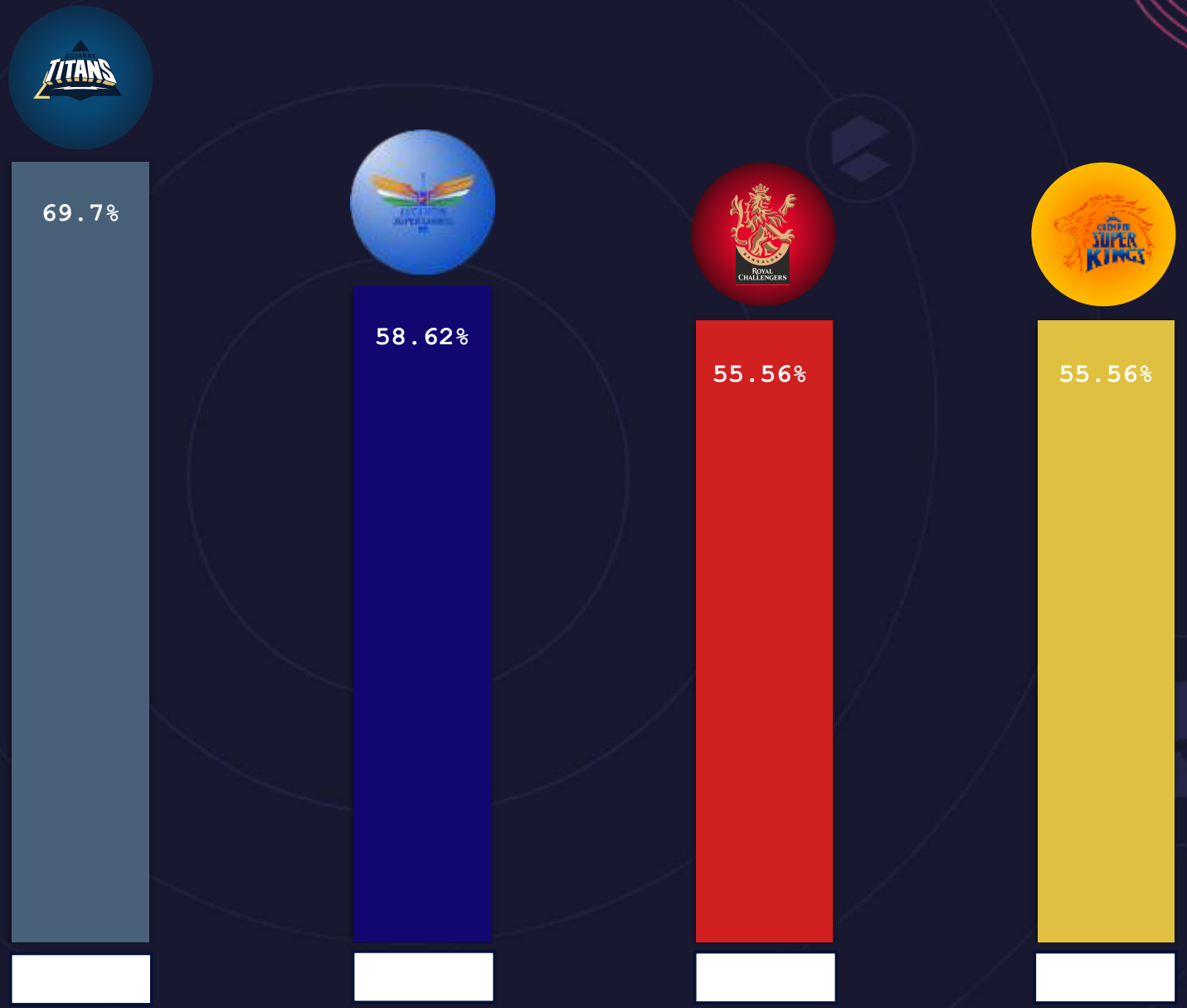
## SQL Query

```
WITH team_wins AS (
    SELECT
        team1 AS team, matchyear,
        COUNT(*) AS wins
    FROM
        dim_match_summary
    WHERE
        winner = team1
        AND matchyear BETWEEN 2021 AND 2023
    GROUP BY
        team1, matchyear
    UNION ALL
    SELECT
        team2 AS team, matchyear,
        COUNT(*) AS wins
    FROM
        dim_match_summary
    WHERE
        winner = team2
        AND matchyear BETWEEN 2021 AND 2023
    GROUP BY
        team2, matchyear
),
team_matches AS (
    SELECT
        team1 AS team,
        matchyear,
        COUNT(*) AS matches
    FROM
        dim_match_summary
WHERE
        matchyear BETWEEN 2021 AND 2023
    GROUP BY
        team1, matchyear
    UNION ALL
    SELECT
        team2 AS team, matchyear,
        COUNT(*) AS matches
    FROM
        dim_match_summary
    WHERE
        matchyear BETWEEN 2021 AND 2023
    GROUP BY
        team2, matchyear
)
SELECT
    tw.team,
    SUM(tw.wins) AS total_wins,
    SUM(tm.matches) AS total_matches,
    round((SUM(tw.wins) / SUM(tm.matches)) * 100, 2) AS
winning_percentage
FROM
    team_wins tw
JOIN
    team_matches tm ON tw.team = tm.team AND
tw.matchyear = tm.matchyear
GROUP BY
    tw.team
ORDER BY
    winning_percentage DESC
LIMIT 4;
```

- CTE1: team_wins ( total wins obtained by teams in both team1 & team2 category )
- UNION ALL: Merge between Team1 & Team2 for total wins of all teams
- CTE2: team_matches ( total matches played by teams in team1 & team2 category )
- UNION ALL: Merge between Team1 & Team2 for total matches played by all teams

# Top 4 Qualifying Teams

Winning percentage over the last 3 years of IPL seasons :
Gujarat Titans (69.7%), Lucknow Super Giants (58.62%),
Royal Challengers Bangalore (55.56%), and Chennai Super Kings (55.56%).



## 10. Top 2 teams with the highest number of wins achieved by chasing targets over the past 3 years.

**SQL Query**

CTE: Chasing_wins

```
With chasing_wins as (
    Select
          team2 as chasing_team,
          matchyear,
          count(*) as wins
          from
        dim_match_summary
          Where
         winner = team2 and
         matchyear between 2021 and 2023
          group by
         team2, matchyear
)
    select
         cw.chasing_team,
         sum(cw.wins) as total_wins
    from
         chasing_wins cw
    group by
         cw.chasing_team
    order by
         total_wins desc
          limit 2;
```

**Query Output**

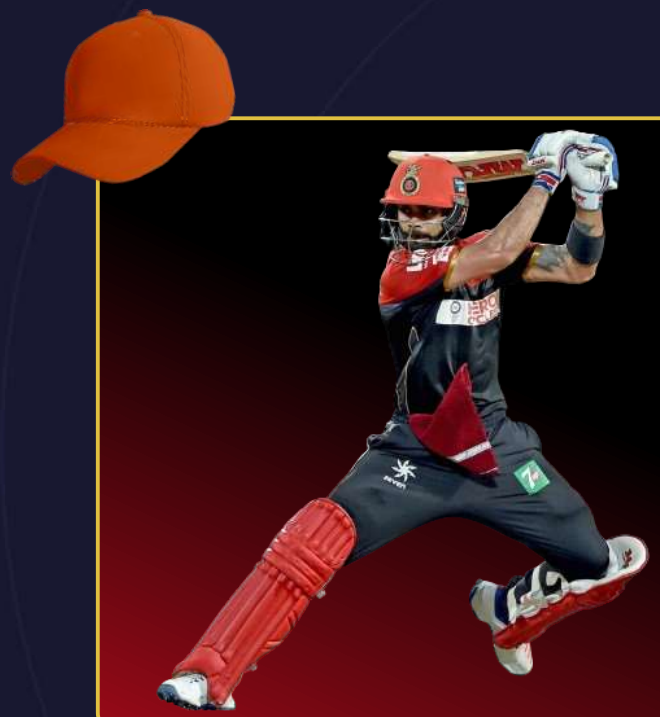
| | chasing_team | total_wins |
|---|---|---|
| ▶ | KKR | 14 |
| | Capitals | 14 |

# Top 2 Teams with Highest wins in past 3 years




**Kolkata Knight Riders and Delhi Capitals both achieved the highest number of victories, totalling 14 wins each.**

# ORANGE CAP PREDICTION



## Overall Performance

**Name** - VIRAT KOHLI
**Team** - ROYAL CHALLENGERS BENGLURU
**Total runs** - 1385
**Innings** - 45
**Strike Rate** - 127.06
**Batting Avg** - 33.78
**Boundary%** - 54.73 %

## 2021

**Total runs** - 405
Batting Avg - 28.93
**Innings** - 15
**Strike Rate** - 119.47
**Boundary%** - 55.80 %

## 2022

**Total runs** - 341
Batting Avg - 22.73
**Innings** - 16
**Strike Rate** - 115.99
**Boundary%** - 51.61 %

## 2023

**Total runs** - 639
Batting Avg - 53.25
**Innings** - 14
**Strike Rate** - 139.82
**Boundary%** - 55.71 %

# PURPLE CAP PREDICTION

## Overall Performance

**Name** - **YUZVENDRA CHAHAL**

**Team** - RAJATHAN ROYALS

| | | |
|---|---|---|
| **Total Wickets** | - | 66 |
| **Innings** | - | 46 |
| **Strike Rate** | - | 15.80 |
| Bowling Eco | - | 7.67 |
| **Dot Ball %** | - | 34.61 % |

## 2021

| | | |
|---|---|---|
| **Total Wickets** | - | 18 |
| Bowling Eco | - | 7.06 |
| **Innings** | - | 18 |
| **Strike Rate** | - | 17.67 |
| **Dot Ball %** | - | 39.94% |

**RANK 4**

## 2022

| | | |
|---|---|---|
| **Total Wickets** | - | 27 |
| Bowling Eco | - | 7.75 |
| **Innings** | - | 17 |
| **Strike Rate** | - | 15.11 |
| **Dot Ball %** | - | 33.58% |

**RANK 1**

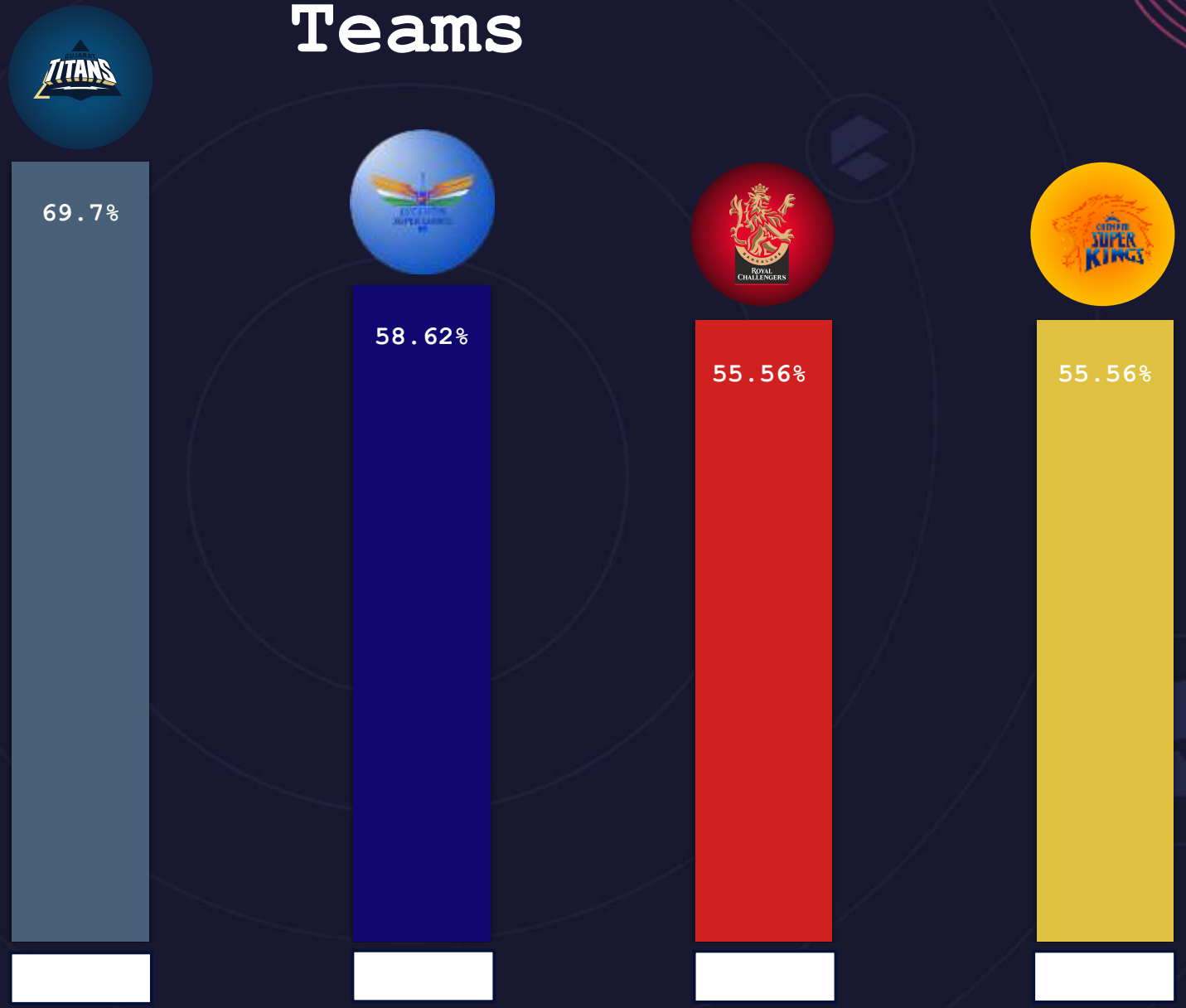
## 2023

| | | |
|---|---|---|
| **Total Wickets** | - | 21 |
| Bowling Eco | - | 8.18 |
| **Innings** | - | 14 |
| **Strike Rate** | - | 15.10 |
| **Dot Ball %** | - | 30.60% |

**RANK 4**

# Top 4 Qualifying Teams

Prediction of Top 4 Qualifying teams based on winning percentage over the last 3 years of IPL seasons : Gujarat Titans (69.7%), Lucknow Super Giants (58.62%), Royal Challengers Bangalore (55.56%), and Chennai Super Kings (55.56%).



INDIAN PREMIER LEAGUE

# Winner and Runner up?


INDIAN PREMIER LEAGUE


Winner


Runner up

- **Consistent Performances**: Lucknow Super Giants is known for it’s consistent performances in previous season, with key players delivering consistently strong performances in both batting and bowling departments.
- **Balanced Squad**: Lucknow Super Giants might have had a well-balanced squad comprising both experienced players and emerging talents.
- KL Rahul is really good at staying cool and making smart choices during tough moments in games. This can help the team stay focused and play better when things get intense.

# BEST 11

| Player Name | Image | Team | Batting Style | Playing Role | Bowling Style | Batting Avg | Batting S/R | Economy | Bowling Avg | Bowling S/R | Batting Order |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RohitSharma |  | Mumbai | Right hand Bat | Top order Batter | Right arm Offbreak | 22.81 | 127.07 | 11.14 | 0.00 | 0.00 | 1 |
| FafduPlessis |  | Super Kings | Right hand Bat | Middle order Batter | Legbreak | 43.60 | 140.85 | | | | 2 |
| ViratKohli |  | RCB | Right hand Bat | Top order Batter | Right arm Medium | 33.78 | 127.06 | | | | 3 |
| SuryakumarYadav |  | Mumbai | Right hand Bat | Batter | Right arm Medium, Right arm Offbreak | 35.00 | 160.55 | | | | 4 |
| CameronGreen |  | Mumbai | Right hand Bat | Batting Allrounder | Right arm Fast medium | 50.22 | 160.28 | 9.50 | 60.17 | 38.00 | 5 |

INDIAN PREMIER LEAGUE

# BEST 11

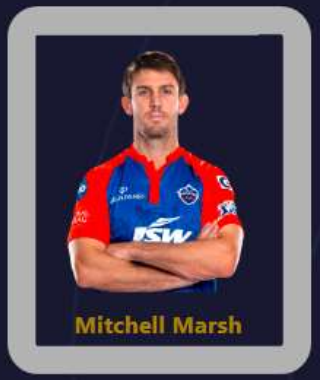
| Player Name | Image | Team | Batting Style | Playing Role | Bowling Style | Batting Avg | Batting S/R | Economy | Bowling Avg | Bowling S/R | Batting Order |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HardikPandya |  | Mumbai | Right hand Bat | Allrounder | Right arm Medium fast | 30.97 | 130.43 | 8.11 | 40.91 | 30.27 | 6 |
| AndreRussell |  | KKR | Right hand Bat | Allrounder | Right arm Fast | 27.59 | 159.19 | 10.24 | 18.23 | 10.69 | 7 |
| RashidKhan |  | Sunrisers | Right hand Bat | Bowling Allrounder | Legbreak Googly | 20.27 | 176.74 | 7.20 | 20.90 | 17.41 | 8 |
| PatCummins |  | KKR | Right hand Bat | Bowler | Right arm Fast | 22.29 | 195.00 | 9.62 | 28.06 | 17.50 | 9 |
| YuzvendraChahal |  | RCB | Right hand Bat | Bowler | Legbreak Googly | 15.00 | 42.86 | 7.67 | 20.20 | 15.80 | 10 |
| JaspritBumrah |  | Mumbai | Right hand Bat | Bowler | Right arm Fast | 4.00 | 64.00 | 7.32 | 22.03 | 18.06 | 11 |

# TOP 3 ALL ROUNDERS

Ravindra Jadeja

| Total Runs | Batting Avg | Batting Strike rate | Boundary % |
|---|---|---|---|
| 533 | 31.35 | 137.73 | 52.91% |
| 227, 116, 190 | 76, 19, 24 | 146, 118, 143 | 57.27%, 46.55%, 51.58% |

| wickets | Bowling_Economy | Bowling Strike Rate | Dot ball % |
|---|---|---|---|
| 37 | 7.46 | 22.05 | 32.84% |
| 13, 5, 19 | 7.52, 7.68, 7.46 | 57.27%, 46.55%, 51.58% | 57.27%, 46.55%, 51.58% |



Mitchell Marsh

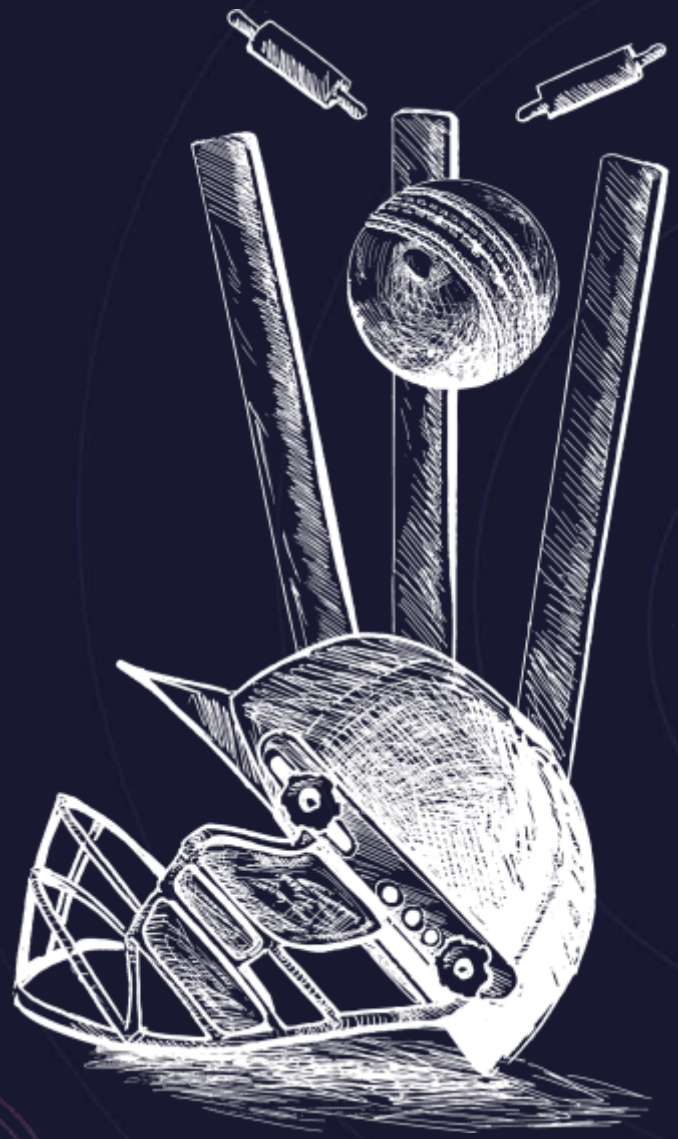
| Total Runs | Batting Avg | Batting Strike rate | Boundary % |
|---|---|---|---|
| 379 | 22.29 | 132.52 | 64.91% |
| 251, 128 | 31, 14 | 132.8, 132.0 | 67.19%, 63.75% |

| wickets | Bowling_Economy | Bowling Strike Rate | Dot ball % |
|---|---|---|---|
| 16 | 8.46 | 12.06 | 31.61% |
| 4, 12 | 8.00, 8.90 | 67.19%, 63.75% | 67.19%, 63.75% |

Andre Russell

| Total Runs | Batting Avg | Batting Strike rate | Boundary % |
|---|---|---|---|
| 745 | 27.59 | 159.19 | 75.70% |
| 183, 335, 227 | 26, 37, 21 | 153, 174, 146 | 78.81%, 76.50%, 70.48% |

| wickets | Bowling_Economy | Bowling Strike Rate | Dot ball % |
|---|---|---|---|
| 35 | 10.24 | 10.69 | 34.49% |
| 11, 17, 7 | 9.9, 9.9 | 78.81%, 76.50%, 70.48% | 78.81%, 76.50%, 70.48% |



# Conclusion

In conclusion, IPL is a huge hit among cricket fans. At the end of this, we will see which team will win the trophy in the final match. So keep watching and keep supporting your Team.

INDIAN PREMIER LEAGUE

# Thank You!