

IPL ANALYSIS

“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.

- I have analyze IPL data from 2021 to 2023 and predicted data for 2024.
- I've find top 10 batsmen, top 10 bowlers
- Winners and runner ups.
- Strike rate
- Most of the Wickets taken by bowlers etc.

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

```
4  -- Top 10 batsmen based on past 3 years total runs scored.
5  • select b.batsmanName,sum(runs) as total_runs
6  from fact_bating_summary b
7  join dim_match_summary m
8  on b.match_id=m.match_id
9  where YEAR(STR_TO_DATE(matchDate, '%b %d, %Y'))>=year(current_date())-3
0  group by b.batsmanname
1  order by total_runs
2  desc
3  limit 10;
```

	batsmanName	total_runs
▶	ShubmanGill	1334
	JosButtler	1255
	FafduPlessis	1198
	ViratKohli	980
	DavidWarner	948
	RuturajGaikwad	932
	SuryakumarYadav	908
	KLRahul	890
	YashasviJaiswal	883
	DevonConway	877

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

```
1 • SELECT
2     b.batsmanName,
3     SUM(runs) AS total_runs,
4     SUM(CASE WHEN `out/not_out` = 'out' THEN 1 ELSE 0 END) AS total_dismissal,
5     SUM(balls) AS total_balls,
6     ROUND(
7         SUM(runs) / NULLIF(SUM(CASE WHEN `out/not_out` = 'out' THEN 1 ELSE 0 END), 0),
8         2
9     ) AS batting_avg
10  FROM fact_bating_summary b
11  JOIN dim_match_summary m ON b.match_id = m.match_id
12  WHERE YEAR(STR_TO_DATE(matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3
13  GROUP BY b.batsmanName
14  HAVING total_balls > 180
15  ORDER BY batting_avg DESC
16  LIMIT 10;
```

batsmanName	total_runs	total_dismissal	total_balls	batting_avg
RajatPatidar	333	6	218	55.50
CameronGreen	452	9	282	50.22
RinkuSingh	648	13	434	49.85
HeinrichKlaasen	448	9	253	49.78
DavidMiller	740	15	515	49.33
DevonConway	877	18	629	48.72
ShubmanGill	1334	28	909	47.64
KLRahul	890	19	697	46.84
JosButtler	1255	29	861	43.28
SuryakumarYadav	908	21	542	43.24

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

```
34 • SELECT
35     b.batsmanName, sum(balls) as total_balls,
36     ROUND(AVG(b.SR), 2) AS avg_strike_rate
37 FROM fact_bating_summary b
38 JOIN dim_match_summary m ON b.match_id = m.match_id
39 WHERE
40     STR_TO_DATE(m.matchDate, '%b %d, %Y') IS NOT NULL AND
41     YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3 and balls>60
42 GROUP BY b.batsmanName
43 ORDER BY avg_strike_rate DESC
44 LIMIT 10;
```

batsmanName	total_balls	avg_strike_rate
QuintondeKock	70	200
YashasviJaiswal	62	200
JosButtler	194	164.79
ViratKohli	124	162.15
PrabhsimranSingh	65	158.46
FafduPlessis	64	150
ShikharDhawan	66	150
MitchellMarsh	62	143.54
KLRahul	123	138.8

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

```
1  -- Top 10 bowlers based on past 3 years total wickets taken.
2  • select * from fact_bowling_summary;
3  • select n.bowlerName,sum(n.wickets) as total_wickets
4  from fact_bowling_summary n
5  join dim_match_summary m
6  on
7  n.match_id=m.match_id
8  where STR_TO_DATE(m.matchDate, '%b %d, %Y') IS NOT NULL AND
9  YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3
10 group by bowlerName,YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y'))
11 order by total_wickets
12 desc
13 limit 10;
```

	bowlerName	total_wickets
▶	MohammedShami	28
	YuzvendraChahal	27
	RashidKhan	27
	WaninduHasaranga	26
	MohitSharma	24
	KagisoRabada	23
	UmransMalik	22
	PiyushChawla	22
	YuzvendraChahal	21
	TusharDeshpande	21

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

```
select b.bowlerName,sum(b.runs)as total_runs,
sum(wickets) as total_wickets,
sum(b.overs)*6 as total_balls
from fact_bowling_summary b
join dim_match_summary m on b.match_id=m.match_id
where STR_TO_DATE(m.matchDate, '%b %d, %Y') IS NOT NULL
      AND YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3
      group by bowlerName, YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y'))
      having total_balls>=60 ),
qualified_bowlers as
(
select bowlerName,
round(sum(total_runs)/nullif(sum(total_wickets),0),2) as bowling_average
from filter_data
group by bowlerName
)
select * from qualified_bowlers
      ORDER BY bowling_average ASC
LIMIT 10;
```

	bowlerName	bowling_average
▶	MarkWood	11.82
	MohitSharma	13.54
	AkashMadhwal	15.64
	MichaelBracewell	15.83
	MitchellMarsh	17.00
	MohsinKhan	17.88
	DeepakChahar	18.17
	DwayneBravo	18.69
	AndreRussell	18.75
	JoshHazlewood	18.85

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

```
WITH filter_data AS (  
    SELECT  
        b.bowlerName,  
        SUM(b.overs) * 6 AS total_balls,  
        SUM(b.runs) AS total_runs  
    FROM fact_bowling_summary b  
    JOIN dim_match_summary m  
        ON b.match_id = m.match_id  
    WHERE STR_TO_DATE(m.matchDate, '%b %d, %Y') IS NOT NULL  
        AND YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3  
    GROUP BY b.bowlerName, YEAR(STR_TO_DATE(m.matchDate, '%b %d, %Y'))  
    HAVING total_balls >= 60  
)  
  
qualified_bowlers AS (  
    SELECT  
        bowlerName,  
        ROUND(SUM(total_runs) / (SUM(total_balls) / 6), 2) AS economy_rate  
    FROM filter_data  
    GROUP BY bowlerName  
)  
  
SELECT *  
FROM qualified_bowlers  
ORDER BY economy_rate ASC  
LIMIT 10;
```

bowlerName	economy_rate
SunilNarine	6.68
DavidWilley	6.81
MitchellSantner	6.81
GlennMaxwell	6.88
MohsinKhan	7.07
JaspriBumrah	7.2
KrunalPandya	7.22
RashidKhan	7.29
AxarPatel	7.32
MoeenAli	7.47

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

```
with filt_data as
(
select b.batsmanName,sum(b.balls) as total_balls,sum(b.4s) as total_4s,
sum(b.6s) as total_6s
from fact_bating_summary b
join dim_match_summary m
on b.match_id=m.match_id
where year(str_to_date(matchdate,'%b %d, %Y'))>=year(curdate())-3
group by b.batsmanName,year(str_to_date(matchdate,'%b %d, %Y'))
having total_balls>60),

quali_batsmen as (
select batsmanName,round((sum(total_4s)+sum(total_6s))/sum(total_balls)*100,2) as boundry_percentage
from filt_data
group by batsmanname)

select * from quali_batsmen
order by boundry_percentage desc
limit 5;
```

	batsmanName	boundry_percentage
▶	GlennMaxwell	26.77
	PhilSalt	25.56
	JasonRoy	25.53
	YashasviJaiswal	25.35
	LiamLivingstone	24.82

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

```
WITH filtered_data AS
(
    SELECT
        b.bowlerName,
        SUM(b.`0s`) AS total_dot_balls,
        SUM(b.overs) * 6 AS total_balls
    FROM fact_bowling_summary b
    JOIN dim_match_summary m ON b.match_id = m.match_id
    WHERE YEAR(STR_TO_DATE(matchDate,'%b %d, %Y')) >= YEAR(CURDATE()) - 3
    GROUP BY b.bowlerName
),
bowlers AS
(
    SELECT
        bowlerName,
        ROUND(SUM(total_dot_balls) / SUM(total_balls) * 100, 2) AS dot_ball_percentage
    FROM filtered_data
    GROUP BY bowlerName
)
SELECT *
FROM bowlers
ORDER BY dot_ball_percentage DESC
LIMIT 5;
```

	bowlerName	dot_ball_percentage
▶	ShreyasIyer	66.67
	AdamMilne	65.22
	ReeceTopley	58.33
	DewaldBrevis	55.56
	SimarjeetSingh	52.78

Top 4 teams based on past 3 years winning %.

```
WITH recent_matches AS (  
    SELECT match_id, matchDate, team1, team2, winner  
    FROM dim_match_summary  
    WHERE YEAR(STR_TO_DATE(matchDate, '%b %d, %Y')) >= YEAR(CURDATE()) - 3  
) ,  
all_teams AS (  
    SELECT team1 AS team FROM recent_matches  
    UNION ALL  
    SELECT team2 AS team FROM recent_matches  
) ,  
matches_played AS (  
    SELECT team, COUNT(*) AS total_matches  
    FROM all_teams  
    GROUP BY team  
) ,  
wins AS (  
    SELECT winner AS team, COUNT(*) AS total_wins  
    FROM recent_matches  
    WHERE winner IS NOT NULL  
    GROUP BY winner  
) ,  
win_percentage_calc AS (  
    SELECT  
        m.team,  
        m.total_matches,  
        COALESCE(w.total_wins, 0) AS total_wins,  
        ROUND(COALESCE(w.total_wins, 0) / m.total_matches * 100, 2) AS win_percentage  
    FROM matches_played m  
    LEFT JOIN wins w ON m.team = w.team  
)
```

Top 4 teams based on past 3 years winning %.

```
SELECT *  
FROM win_percentage_calc  
ORDER BY win_percentage DESC  
LIMIT 4;
```

	team	total_matches	total_wins	win_percentage
▶	Titans	32	23	71.88
	Super Giants	29	17	58.62
	Royals	31	17	54.84
	RCB	30	16	53.33

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

- ```
SELECT
 team2 AS chasing_team,
 COUNT(*) AS chasing_wins
FROM dim_match_summary
WHERE
 STR_TO_DATE(matchDate, '%b %d, %Y') >= DATE_SUB(CURDATE(), INTERVAL 3 YEAR)
 AND team2 = winner
GROUP BY team2
ORDER BY chasing_wins DESC
LIMIT 2;
```

|   | chasing_team | chasing_wins |
|---|--------------|--------------|
| ▶ | Titans       | 12           |
|   | Mumbai       | 9            |

## Top 4 qualifying teams

```
1 • SELECT
2 winner AS team_name,
3 COUNT(*) AS total_wins
4 FROM
5 dim_match_summary
6 WHERE
7 YEAR(STR_TO_DATE(matchDate, '%b %d, %Y')) = 2023
8 AND winner IS NOT NULL
9 GROUP BY
10 winner
11 ORDER BY
12 total_wins DESC
13 LIMIT 4;
```

|   | team_name    | total_wins |
|---|--------------|------------|
| ► | Titans       | 11         |
|   | Mumbai       | 9          |
|   | Super Kings  | 9          |
|   | Super Giants | 8          |

Based on consistent and strong performances from 2021 - 2023, the following four teams have emerged as top contenders.

## Winner and runner-up

```
1 • SELECT
2 matchDate,
3 winner,
4 CASE
5 WHEN team1 = winner THEN team2
6 ELSE team1
7 END AS runner_up,
8 YEAR(STR_TO_DATE(matchDate, '%b %d, %Y')) AS year
9 FROM dim_match_summary
10 WHERE YEAR(STR_TO_DATE(matchDate, '%b %d, %Y'))
11 ORDER BY STR_TO_DATE(matchDate, '%b %d, %Y') desc
12 LIMIT 1;
```

| matchDate    | winner | runner_up | year |
|--------------|--------|-----------|------|
| May 26, 2023 | Titans | Mumbai    | 2023 |