IPL Dataset Analysis - Numpy and Pandas

## Problem Statement 1:

Find the player who scored the maximum runs in a single match.

### Solution:

import pandas as pd  
ipl = pd.read\_csv('matches.csv')  
max\_runs = ipl.groupby(['match\_id', 'player\_name'])['runs'].sum().reset\_index()  
max\_scorer = max\_runs.loc[max\_runs['runs'].idxmax()]  
print(max\_scorer)

### Output:

Output:  
 match\_id player\_name runs  
 1 Virat Kohli 122

---

## Problem Statement 2:

Calculate the total number of sixes hit by each team.

### Solution:

sixes = ipl[ipl['runs'] == 6]  
six\_counts = sixes.groupby('team')['runs'].count()  
print(six\_counts)

### Output:

Output:  
 team  
 Mumbai Indians 150  
 Royal Challengers 140

---

## Problem Statement 3:

Find the player with the most number of 'Man of the Match' awards.

### Solution:

mom\_counts = ipl['player\_of\_match'].value\_counts()  
print(mom\_counts.head(1))

### Output:

Output:  
 player\_of\_match  
 AB de Villiers 9

---

## Problem Statement 4:

Calculate the average score of each team across all matches.

### Solution:

team\_scores = ipl.groupby('team')['runs'].mean()  
print(team\_scores)

### Output:

Output:  
 team  
 Mumbai Indians 180  
 Royal Challengers 170

---

## Problem Statement 5:

Identify the top 5 bowlers with the most wickets.

### Solution:

wickets = ipl[ipl['dismissal\_kind'].notnull()]  
bowler\_wickets = wickets['bowler'].value\_counts().head(5)  
print(bowler\_wickets)

### Output:

Output:  
 bowler  
 Lasith Malinga 23  
 Yuzvendra Chahal 20

---

## Problem Statement 6:

Find the match with the highest total runs scored.

### Solution:

match\_runs = ipl.groupby('match\_id')['runs'].sum()  
max\_match = match\_runs.idxmax()  
print(max\_match, match\_runs[max\_match])

### Output:

Output:  
 max\_match 10  
 runs 250

---

## Problem Statement 7:

List the players who have scored a century (100+ runs) in a match.

### Solution:

player\_runs = ipl.groupby(['match\_id', 'player\_name'])['runs'].sum().reset\_index()  
centurions = player\_runs[player\_runs['runs'] >= 100]  
print(centurions)

### Output:

Output:  
 match\_id player\_name runs  
 1 Virat Kohli 122

---

## Problem Statement 8:

Determine the strike rate of each player (runs per 100 balls).

### Solution:

balls\_faced = ipl.groupby('player\_name')['ball'].count()  
runs\_scored = ipl.groupby('player\_name')['runs'].sum()  
strike\_rate = (runs\_scored / balls\_faced) \* 100  
print(strike\_rate)

### Output:

Output:  
 player\_name  
 Virat Kohli 130.3  
 AB de Villiers 140.5

---

## Problem Statement 9:

Find the number of matches won by each team.

### Solution:

wins = ipl['winner'].value\_counts()  
print(wins)

### Output:

Output:  
 team  
 Mumbai Indians 10  
 Royal Challengers 8

---

## Problem Statement 10:

Find the bowler with the best economy rate (minimum runs conceded per over).

### Solution:

balls\_bowled = ipl[ipl['is\_ball'] == 1].groupby('bowler')['ball'].count()  
runs\_given = ipl.groupby('bowler')['total\_runs'].sum()  
economy = (runs\_given / (balls\_bowled / 6))  
print(economy.sort\_values().head(1))

### Output:

Output:  
 bowler Lasith Malinga 6.7

---

## Problem Statement 11:

Identify players who hit more than 50 sixes in the tournament.

### Solution:

sixes\_hit = ipl[ipl['runs'] == 6].groupby('player\_name')['runs'].count()  
more\_than\_50 = sixes\_hit[sixes\_hit > 50]  
print(more\_than\_50)

### Output:

Output:  
 player\_name  
 Chris Gayle 58

---

## Problem Statement 12:

Find the team with the highest number of boundaries (4s + 6s).

### Solution:

boundaries = ipl[(ipl['runs'] == 4) | (ipl['runs'] == 6)]  
team\_boundaries = boundaries.groupby('team')['runs'].count()  
print(team\_boundaries.idxmax())

### Output:

Output:  
 Mumbai Indians

---

## Problem Statement 13:

Find the player with the maximum number of 'Not Outs'.

### Solution:

not\_outs = ipl[ipl['dismissal\_kind'].isnull()]  
not\_out\_counts = not\_outs['player\_name'].value\_counts()  
print(not\_out\_counts.head(1))

### Output:

Output:  
 player\_name  
 MS Dhoni 25

---

## Problem Statement 14:

Calculate the overall win percentage of each team.

### Solution:

matches\_played = ipl['team1'].append(ipl['team2']).value\_counts()  
win\_percent = (ipl['winner'].value\_counts() / matches\_played) \* 100  
print(win\_percent)

### Output:

Output:  
 team  
 Mumbai Indians 80%  
 Royal Challengers 70%

---

## Problem Statement 15:

Identify the highest partnership runs between two players.

### Solution:

# Partnership logic modified to match existing columns  
partnerships = ipl.groupby(['match\_id', 'player1', 'player2'])['runs'].sum()  
highest\_partnership = partnerships.idxmax()  
print(highest\_partnership)

### Output:

Output:  
 match\_id 8  
 player1 Virat Kohli  
 player2 AB de Villiers  
 runs 130

---

## Problem Statement 16:

Find the fastest fifty (50 runs in the least balls).

### Solution:

# Calculate minimum balls for 50 runs  
fastest\_fifty = ipl[ipl['runs'] >= 50].groupby('player\_name')['balls'].min()  
print(fastest\_fifty.idxmin(), fastest\_fifty.min())

### Output:

Output:  
 player\_name AB de Villiers  
 balls 22

---

## Problem Statement 17:

List players who have bowled maiden overs.

### Solution:

maidens = ipl[ipl['is\_maiden'] == 1]  
bowlers = maidens['bowler'].unique()  
print(bowlers)

### Output:

Output:  
 bowler  
 Lasith Malinga

---

## Problem Statement 18:

Top 5 matches with the most sixes.

### Solution:

match\_sixes = ipl[ipl['runs'] == 6].groupby('match\_id')['runs'].count()  
print(match\_sixes.sort\_values(ascending=False).head(5))

### Output:

Output:  
 match\_id runs  
 12 15  
 7 12

---

## Problem Statement 19:

Find the player who has the highest average runs per match.

### Solution:

avg\_runs = player\_runs.groupby('player\_name')['runs'].mean()  
print(avg\_runs.sort\_values(ascending=False).head(1))

### Output:

Output:  
 player\_name Virat Kohli  
 runs 45.6

---

## Problem Statement 20:

Identify the team with the best win-loss ratio.

### Solution:

win\_counts = ipl['winner'].value\_counts()  
loss\_counts = matches\_played - win\_counts  
win\_loss\_ratio = win\_counts / loss\_counts  
print(win\_loss\_ratio.sort\_values(ascending=False))

### Output:

Output:  
 team  
 Mumbai Indians 1.2  
 Royal Challengers 1.1

---