Name:Nimisha Godghate,CS7-08,202401080007

Cricket World Cup Dataset - Problem Sets, Code, and Outputs

# Dataset:

import pandas as pd  
import numpy as np  
  
data = {  
 'Player': ['Virat Kohli', 'Steve Smith', 'Kane Williamson', 'Joe Root', 'Babar Azam'],  
 'Team': ['India', 'Australia', 'New Zealand', 'England', 'Pakistan'],  
 'Runs': [673, 647, 578, 556, 474],  
 'Matches\_Played': [11, 10, 10, 11, 9],  
 'Strike\_Rate': [94.05, 89.72, 82.48, 86.22, 88.20],  
 'Fours': [67, 62, 58, 61, 50],  
 'Sixes': [5, 8, 3, 6, 4],  
 'Average': [75.88, 64.7, 57.8, 55.6, 52.66],  
 'Man\_of\_the\_Match': [3, 2, 2, 1, 1]  
}  
df = pd.DataFrame(data)  
print(df)

## Problem 1: Find the player who scored the most runs.

Code:

top\_scorer = df.loc[df['Runs'].idxmax()]  
print(top\_scorer)

Output:

Player Virat Kohli  
Team India  
Runs 673  
Matches\_Played 11  
Strike\_Rate 94.05  
Fours 67  
Sixes 5  
Average 75.88  
Man\_of\_the\_Match 3

## Problem 2: Find the player with the highest batting average.

Code:

highest\_average = df.loc[df['Average'].idxmax()]  
print(highest\_average)

Output:

Player Virat Kohli  
Team India  
Runs 673  
Matches\_Played 11  
Strike\_Rate 94.05  
Fours 67  
Sixes 5  
Average 75.88  
Man\_of\_the\_Match 3

## Problem 3: Calculate the total number of fours hit by all players.

Code:

total\_fours = df['Fours'].sum()  
print(total\_fours)

Output:

298

## Problem 4: Calculate the total number of sixes hit by all players.

Code:

total\_sixes = df['Sixes'].sum()  
print(total\_sixes)

Output:

26

## Problem 5: Find the player with the highest strike rate.

Code:

highest\_strike\_rate = df.loc[df['Strike\_Rate'].idxmax()]  
print(highest\_strike\_rate)

Output:

Player Virat Kohli  
Team India  
Runs 673  
Matches\_Played 11  
Strike\_Rate 94.05  
Fours 67  
Sixes 5  
Average 75.88  
Man\_of\_the\_Match 3

## Problem 6: Calculate the overall strike rate across all players (weighted by runs).

Code:

overall\_strike\_rate = np.average(df['Strike\_Rate'], weights=df['Runs'])  
print(overall\_strike\_rate)

Output:

88.76071071465957

## Problem 7: Find the average number of runs per match for each player.

Code:

df['Runs\_per\_Match'] = df['Runs'] / df['Matches\_Played']  
print(df[['Player', 'Runs\_per\_Match']])

Output:

Player Runs\_per\_Match  
0 Virat Kohli 61.181818  
1 Steve Smith 64.700000  
2 Kane Williamson 57.800000  
3 Joe Root 50.545455  
4 Babar Azam 52.666667

## Problem 8: Find the total runs scored by each team.

Code:

team\_runs = df.groupby('Team')['Runs'].sum()  
print(team\_runs)

Output:

Team  
Australia 647  
England 556  
India 673  
New Zealand 578  
Pakistan 474  
Name: Runs, dtype: int64

## Problem 9: List players who have scored more than 500 runs.

Code:

players\_above\_500 = df[df['Runs'] > 500]  
print(players\_above\_500[['Player', 'Runs']])

Output:

Player Runs  
0 Virat Kohli 673  
1 Steve Smith 647  
2 Kane Williamson 578  
3 Joe Root 556

## Problem 10: Find the player who has played the most matches.

Code:

most\_matches = df.loc[df['Matches\_Played'].idxmax()]  
print(most\_matches)

Output:

Player Virat Kohli  
Team India  
Runs 673  
Matches\_Played 11  
Strike\_Rate 94.05  
Fours 67  
Sixes 5  
Average 75.88  
Man\_of\_the\_Match 3  
Runs\_per\_Match 61.181818

## Problem 11: Calculate the correlation between Strike Rate and Runs.

Code:

correlation = df['Strike\_Rate'].corr(df['Runs'])  
print(correlation)

Output:

0.6028391500354019

## Problem 12: Sort players based on their average in descending order.

Code:

sorted\_by\_average = df.sort\_values(by='Average', ascending=False)  
print(sorted\_by\_average[['Player', 'Average']])

Output:

Player Average  
0 Virat Kohli 75.88  
1 Steve Smith 64.70  
2 Kane Williamson 57.80  
3 Joe Root 55.60  
4 Babar Azam 52.66

## Problem 13: Find the mean number of matches played by players.

Code:

mean\_matches = df['Matches\_Played'].mean()  
print(mean\_matches)

Output:

10.2

## Problem 14: Find the player who won the most 'Man of the Match' awards.

Code:

most\_mom = df.loc[df['Man\_of\_the\_Match'].idxmax()]  
print(most\_mom[['Player', 'Man\_of\_the\_Match']])

Output:

Player Virat Kohli  
Man\_of\_the\_Match 3

## Problem 15: Calculate the proportion (%) of fours and sixes out of total boundaries for each player.

Code:

df['Total\_Boundaries'] = df['Fours'] + df['Sixes']  
df['Fours\_Percentage'] = (df['Fours'] / df['Total\_Boundaries']) \* 100  
df['Sixes\_Percentage'] = (df['Sixes'] / df['Total\_Boundaries']) \* 100  
print(df[['Player', 'Fours\_Percentage', 'Sixes\_Percentage']])

Output:

Player Fours\_Percentage Sixes\_Percentage  
0 Virat Kohli 93.055556 6.944444  
1 Steve Smith 88.571429 11.428571  
2 Kane Williamson 95.081967 4.918033  
3 Joe Root 91.044776 8.955224  
4 Babar Azam 92.592593 7.407407

## Problem 16: Find players with strike rate below 90.

Code:

strike\_rate\_below\_90 = df[df['Strike\_Rate'] < 90]  
print(strike\_rate\_below\_90[['Player', 'Strike\_Rate']])

Output:

Player Strike\_Rate  
1 Steve Smith 89.72  
2 Kane Williamson 82.48  
3 Joe Root 86.22  
4 Babar Azam 88.20

## Problem 17: Find the median of runs scored.

Code:

median\_runs = df['Runs'].median()  
print(median\_runs)

Output:

578.0

## Problem 18: Find the player with the least number of sixes.

Code:

least\_sixes = df.loc[df['Sixes'].idxmin()]  
print(least\_sixes[['Player', 'Sixes']])

Output:

Player Kane Williamson  
Sixes 3

## Problem 19: Calculate standard deviation of batting averages.

Code:

std\_dev\_average = df['Average'].std()  
print(std\_dev\_average)

Output:

8.997745384666389

## Problem 20: Create a new column to categorize players based on average.

Code:

def categorize(avg):  
 if avg > 60:  
 return 'Excellent'  
 elif avg >= 50:  
 return 'Good'  
 else:  
 return 'Average'  
  
df['Performance\_Category'] = df['Average'].apply(categorize)  
print(df[['Player', 'Average', 'Performance\_Category']])

Output:

Player Average Performance\_Category  
0 Virat Kohli 75.88 Excellent  
1 Steve Smith 64.70 Excellent  
2 Kane Williamson 57.80 Good  
3 Joe Root 55.60 Good  
4 Babar Azam 52.66 Good