Sports analysis case study – answer

### Step-by-Step Instructions

1. Setup the Environment:
   * Ensure you have Python installed on your computer.
   * Install the necessary libraries: pandas, matplotlib, and Jupyter Notebook (if you prefer an interactive environment).

sh

Copy code

pip install pandas matplotlib jupyter

1. Prepare the Data:
   * Download the IPL datasets (IPL Ball-by-Ball 2008-2020.csv and IPL Matches 2008-2020.csv).
   * Place the files in a directory where you'll run your Python script or Jupyter Notebook.
2. Load the Data:
   * Start by importing the necessary libraries and loading the datasets into pandas DataFrames.

import pandas as pd

# Load the datasets

ball\_by\_ball\_df = pd.read\_csv('IPL Ball-by-Ball 2008-2020.csv')

matches\_df = pd.read\_csv('IPL Matches 2008-2020.csv')

# Display the first few rows of each dataframe to understand their structure

print(ball\_by\_ball\_df.head())

print(matches\_df.head())

1. Basic Analysis of IPL Data:
   * Generate a summary of the matches dataset.

matches\_summary = matches\_df.describe(include='all')

print(matches\_summary)

1. In-depth Analysis of Batsman Performance:
   * Calculate the total runs and balls faced by each batsman.

batsman\_performance = ball\_by\_ball\_df.groupby('batsman').agg({

'batsman\_runs': 'sum',

'id': 'count'

}).rename(columns={'id': 'balls\_faced'}).reset\_index()

# Top 10 Batsmen by runs

top\_batsmen = batsman\_performance.sort\_values(by='batsman\_runs', ascending=False).head(10)

print(top\_batsmen)

1. Analysis of Toss Decisions Across Seasons:
   * Extract the season from the date column and analyze the toss decisions.

matches\_df['season'] = pd.to\_datetime(matches\_df['date']).dt.year

toss\_decisions = matches\_df.groupby(['season', 'toss\_decision']).size().unstack().fillna(0)

print(toss\_decisions)

1. Analysis of Toss Winning and Match Winning:
   * Determine the ratio of matches where the team that won the toss also won the match.

toss\_and\_match\_wins = matches\_df[matches\_df['toss\_winner'] == matches\_df['winner']].shape[0]

total\_matches = matches\_df.shape[0]

toss\_win\_match\_win\_ratio = toss\_and\_match\_wins / total\_matches

print(f"Toss Win to Match Win Ratio: {toss\_win\_match\_win\_ratio:.2f}")

1. Analysis of Tournament Wins by Team:
   * Count the number of matches won by each team.

tournament\_wins = matches\_df.groupby('winner').size().reset\_index(name='count').sort\_values(by='count', ascending=False)

print(tournament\_wins)

1. Comparative Analysis of Teams:
   * Perform a comparative analysis of teams based on the number of matches played and the average winning margin.

teams\_comparative\_analysis = matches\_df.groupby('team1').agg({

'id': 'count',

'result\_margin': 'mean'

}).rename(columns={'id': 'matches\_played', 'result\_margin': 'average\_margin'}).reset\_index()

print(teams\_comparative\_analysis)

1. Visualize the Results (Optional):
   * Use matplotlib or another visualization library to create charts for better insights.

import matplotlib.pyplot as plt

# Example: Bar chart for top batsmen

top\_batsmen.plot(kind='bar', x='batsman', y='batsman\_runs', title='Top 10 Batsmen by Runs')

plt.show()

### Additional Tips

* Encourage the student to explore the data further and ask their own questions.
* They can save their results to CSV files or create a report with their findings.
* Ensure they understand each step and the reason behind each analysis.

This should provide a comprehensive guide for an internship student to perform the IPL data analysis activity.