

## IMPORT LIBRARIES

```
In [1]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

## READ THE CSV FILE

```
In [9]: df = pd.read_csv('D:/python/pandas/project/rcb_vs_kkr.csv')  
print(df)
```

	match_no	date	stage	venue	batting_team \
0	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
1	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
2	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
3	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
4	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
..	...	...	...	...	...
218	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
219	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
220	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
221	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
222	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB

	bowling_team	innings	over	striker	bowler	runs_of_bat \
0	RCB	1	0.1	de Kock	Hazlewood	0
1	RCB	1	0.2	de Kock	Hazlewood	4
2	RCB	1	0.3	de Kock	Hazlewood	0
3	RCB	1	0.4	de Kock	Hazlewood	0
4	RCB	1	0.5	de Kock	Hazlewood	0
..	...	...	...	...	...	...
218	KKR	2	15.4	Livingstone	Vaibhav Arora	4
219	KKR	2	15.5	Livingstone	Vaibhav Arora	0
220	KKR	2	15.6	Livingstone	Vaibhav Arora	1
221	KKR	2	16.1	Livingstone	Spencer Johnson	6
222	KKR	2	16.2	Livingstone	Spencer Johnson	4

	extras	wide	legbyes	byes	noballs	wicket_type	player_dismissed \
0	0	0	0	0	0	NaN	NaN
1	0	0	0	0	0	NaN	NaN
2	0	0	0	0	0	NaN	NaN
3	0	0	0	0	0	NaN	NaN
4	0	0	0	0	0	caught	de Kock
..	...	...	...	...	...	...	...
218	0	0	0	0	0	NaN	NaN
219	0	0	0	0	0	NaN	NaN
220	0	0	0	0	0	NaN	NaN
221	0	0	0	0	0	NaN	NaN
222	0	0	0	0	0	NaN	NaN

	fielder
0	NaN
1	NaN
2	NaN

```

3          NaN
4    Jitesh Sharma
..          ...
218         NaN
219         NaN
220         NaN
221         NaN
222         NaN

```

[223 rows x 19 columns]

In [11]: `df.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 223 entries, 0 to 222
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   match_no              223 non-null   int64
1   date                  223 non-null   object
2   stage                 223 non-null   object
3   venue                 223 non-null   object
4   batting_team          223 non-null   object
5   bowling_team          223 non-null   object
6   innings               223 non-null   int64
7   over                  223 non-null   float64
8   striker               223 non-null   object
9   bowler                223 non-null   object
10  runs_of_bat           223 non-null   int64
11  extras                223 non-null   int64
12  wide                  223 non-null   int64
13  legbyes               223 non-null   int64
14  byes                  223 non-null   int64
15  noballs               223 non-null   int64
16  wicket_type           11 non-null    object
17  player_dismissed      11 non-null    object
18  fielder                8 non-null     object
dtypes: float64(1), int64(8), object(10)
memory usage: 33.2+ KB

```

In [13]: `df.head()`

Out[13]:

	match_no	date	stage	venue	batting_team	bowling_team	innings	over	striker	bowler	runs_of_bat	extras	wide	legbyes	byes	no
0	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR	RCB	1	0.1	de Kock	Hazlewood	0	0	0	0	0	
1	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR	RCB	1	0.2	de Kock	Hazlewood	4	0	0	0	0	
2	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR	RCB	1	0.3	de Kock	Hazlewood	0	0	0	0	0	
3	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR	RCB	1	0.4	de Kock	Hazlewood	0	0	0	0	0	
4	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR	RCB	1	0.5	de Kock	Hazlewood	0	0	0	0	0	

In [15]:

df.tail()

Out[15]:

	match_no	date	stage	venue	batting_team	bowling_team	innings	over	striker	bowler	runs_of_bat	extras	wide	legbyes	byes
218	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB	KKR	2	15.4	Livingstone	Vaibhav Arora	4	0	0	0	0
219	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB	KKR	2	15.5	Livingstone	Vaibhav Arora	0	0	0	0	0
220	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB	KKR	2	15.6	Livingstone	Vaibhav Arora	1	0	0	0	0
221	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB	KKR	2	16.1	Livingstone	Spencer Johnson	6	0	0	0	0
222	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB	KKR	2	16.2	Livingstone	Spencer Johnson	4	0	0	0	0

In [17]:

df.duplicated()

Out[17]:

0 False  
1 False  
2 False  
3 False  
4 False  
...  
218 False  
219 False  
220 False  
221 False  
222 False  
Length: 223, dtype: bool

In [19]:

df.duplicated().sum()

Out[19]:

0

```
In [25]: df = df.drop_duplicates()  
print("/n Data after removing duplicates",df)
```

/n	Data after removing duplicates	match_no	date	stage	venue	batting_team \
0	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR		
1	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR		
2	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR		
3	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR		
4	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR		
..	...	...	...	...		
218	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB		
219	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB		
220	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB		
221	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB		
222	1 Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB		

	bowling_team	innings	over	striker	bowler	runs_of_bat \
0	RCB	1	0.1	de Kock	Hazlewood	0
1	RCB	1	0.2	de Kock	Hazlewood	4
2	RCB	1	0.3	de Kock	Hazlewood	0
3	RCB	1	0.4	de Kock	Hazlewood	0
4	RCB	1	0.5	de Kock	Hazlewood	0
..	...	...	...	...	...	...
218	KKR	2	15.4	Livingstone	Vaibhav Arora	4
219	KKR	2	15.5	Livingstone	Vaibhav Arora	0
220	KKR	2	15.6	Livingstone	Vaibhav Arora	1
221	KKR	2	16.1	Livingstone	Spencer Johnson	6
222	KKR	2	16.2	Livingstone	Spencer Johnson	4

	extras	wide	legbyes	byes	noballs	wicket_type	player_dismissed \
0	0	0	0	0	0	unknown	unknown
1	0	0	0	0	0	unknown	unknown
2	0	0	0	0	0	unknown	unknown
3	0	0	0	0	0	unknown	unknown
4	0	0	0	0	0	caught	de Kock
..	...	...	...	...	...	...	...
218	0	0	0	0	0	unknown	unknown
219	0	0	0	0	0	unknown	unknown
220	0	0	0	0	0	unknown	unknown
221	0	0	0	0	0	unknown	unknown
222	0	0	0	0	0	unknown	unknown

	fielder
0	unknown
1	unknown
2	unknown

```

3      unknown
4    Jitesh Sharma
..      ...
218    unknown
219    unknown
220    unknown
221    unknown
222    unknown

```

[223 rows x 19 columns]

In [21]: `df.describe()`

Out[21]:

	match_no	innings	over	runs_of_bat	extras	wide	legbyes	byes	noballs
<b>count</b>	223.0	223.000000	223.000000	223.000000	223.000000	223.000000	223.000000	223.0	223.0
<b>mean</b>	1.0	1.443946	8.981614	1.533632	0.040359	0.022422	0.017937	0.0	0.0
<b>std</b>	0.0	0.497966	5.388424	1.842318	0.197242	0.148383	0.133022	0.0	0.0
<b>min</b>	1.0	1.000000	0.100000	0.000000	0.000000	0.000000	0.000000	0.0	0.0
<b>25%</b>	1.0	1.000000	4.400000	0.000000	0.000000	0.000000	0.000000	0.0	0.0
<b>50%</b>	1.0	1.000000	8.600000	1.000000	0.000000	0.000000	0.000000	0.0	0.0
<b>75%</b>	1.0	2.000000	13.350000	2.000000	0.000000	0.000000	0.000000	0.0	0.0
<b>max</b>	1.0	2.000000	19.600000	6.000000	1.000000	1.000000	1.000000	0.0	0.0

In [23]:

```

df['wicket_type'] = df['wicket_type'].fillna('unknown')
df['player_dismissed'] = df['player_dismissed'].fillna('unknown')
df['fielder'] = df['fielder'].fillna('unknown')
print("\n Data after updating")
print(df)

```



/n Data after updating

	match_no	date	stage	venue	batting_team \
0	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
1	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
2	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
3	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
4	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	KKR
..	...	...	...	...	...
218	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
219	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
220	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
221	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB
222	1	Mar 22, 2025	League stage	Eden Gardens, Kolkata	RCB

	bowling_team	innings	over	striker	bowler	runs_of_bat \
0	RCB	1	0.1	de Kock	Hazlewood	0
1	RCB	1	0.2	de Kock	Hazlewood	4
2	RCB	1	0.3	de Kock	Hazlewood	0
3	RCB	1	0.4	de Kock	Hazlewood	0
4	RCB	1	0.5	de Kock	Hazlewood	0
..	...	...	...	...	...	...
218	KKR	2	15.4	Livingstone	Vaibhav Arora	4
219	KKR	2	15.5	Livingstone	Vaibhav Arora	0
220	KKR	2	15.6	Livingstone	Vaibhav Arora	1
221	KKR	2	16.1	Livingstone	Spencer Johnson	6
222	KKR	2	16.2	Livingstone	Spencer Johnson	4

	extras	wide	legbyes	byes	noballs	wicket_type	player_dismissed \
0	0	0	0	0	0	unknown	unknown
1	0	0	0	0	0	unknown	unknown
2	0	0	0	0	0	unknown	unknown
3	0	0	0	0	0	unknown	unknown
4	0	0	0	0	0	caught	de Kock
..	...	...	...	...	...	...	...
218	0	0	0	0	0	unknown	unknown
219	0	0	0	0	0	unknown	unknown
220	0	0	0	0	0	unknown	unknown
221	0	0	0	0	0	unknown	unknown
222	0	0	0	0	0	unknown	unknown

	fielder
0	unknown
1	unknown

```

2         unknown
3         unknown
4    Jitesh Sharma
..
218        unknown
219        unknown
220        unknown
221        unknown
222        unknown

```

[223 rows x 19 columns]

## TOTAL RUNS SCORED IN THE MATCH

```
In [28]: total_runs = df[['runs_of_bat', 'extras']].sum().sum()
print("Total runs:", total_runs)
```

Total runs: 351

## RUNS SCORED BY TEAMS WITH FALL OF WICKETS

```
In [106... # Convert columns to numeric and handle missing values
df["runs_of_bat"] = pd.to_numeric(df["runs_of_bat"], errors="coerce").fillna(0)
df["extras"] = pd.to_numeric(df["extras"], errors="coerce").fillna(0)

# Calculate total runs per innings
df["total_runs"] = df["runs_of_bat"] + df["extras"]
runs = df.groupby("innings")["total_runs"].sum()

# Count wickets per innings
wickets = df[df["wicket_type"] != "unknown"].groupby("innings")["wicket_type"].count()

# Map innings to team names
teams = {1: "KKR", 2: "RCB"}

# Create match summary
team_names = {1: "KKR", 2: "RCB"}

# Create match summary dictionary
match_summary = {
    team_names[innings]: f"{int(total_runs.loc[innings, 'final_score'])}/{total_wickets.get(innings, 0)}"
    for innings in team_names
}
```

```

}

# Display total socre by each team
print("Total score:")
for team, score in summary.items():
    print(f"{team}: {score}")

# Determine the winner
kkr_runs, kkr_wkts = map(int, summary["KKR"].split("/"))
rcb_runs, rcb_wkts = map(int, summary["RCB"].split("/"))

if rcb_runs > kkr_runs:
    print(f"RCB won by {10 - rcb_wkts} wickets")
elif kkr_runs > rcb_runs:
    print(f"KKR won by {kkr_runs - rcb_runs} runs")
else:
    print("Match tied")

```

Total score:

KKR: 174/8

RCB: 177/3

RCB won by 7 wickets

## LIST OF RUNS SCORED BY BATSMAN FROM EACH TEAM

```

In [91]: import pandas as pd
# Make sure runs are numbers
df["runs_of_bat"] = pd.to_numeric(df["runs_of_bat"], errors="coerce").fillna(0)

# Add up runs for each player on their team
runs_by_player = df.groupby(["batting_team", "striker"])[["runs_of_bat"]].sum()

# For each team, show top 10 run-scorers
for team in runs_by_player.index.get_level_values(0).unique():
    print(f"\nTop scorers for {team}:")
    top_scorers = runs_by_player[team].sort_values(ascending=False).head(10)
    for player, runs in top_scorers.items():
        print(f"{player}: {runs}")

```

Top scorers for KKR:

Rahane: 56

Narine: 44

Angkrish Raghuvanshi: 30

Rinku Singh: 12

Ramandeep Singh: 6

Venkatesh Iyer: 6

Harshit Rana: 5

Russell: 4

de Kock: 4

Spencer Johnson: 1

Top scorers for RCB:

Kohli: 59

Phil Salt: 56

Rajat Patidar: 34

Livingstone: 15

Padikkal: 10

## DETAILED MATCH SUMMARY

In [101...

```
import pandas as pd

# Convert relevant columns to numeric and handle missing values
df["runs_of_bat"] = pd.to_numeric(df["runs_of_bat"], errors="coerce").fillna(0)
df["extras"] = pd.to_numeric(df["extras"], errors="coerce").fillna(0)

# Assume wicket_type already has "unknown" filled in for missing values

# Extract venue info
venue = df["venue"].unique()[0]

# Function to get top 3 batsmen for a given inning
def top_batsmen(df, inning):
    batsmen = df[df["innings"] == inning].groupby("striker")["runs_of_bat"].sum()
    return batsmen.sort_values(ascending=False).head(3)

# Function to get top 3 bowlers for a given inning
def top_bowlers(df, inning):
    valid_bowling = df[(df["innings"] == inning) & (df["wicket_type"] != "unknown")]
    stats = valid_bowling.groupby("bowler")["wicket_type"].count()
    return stats.sort_values(ascending=False).head(3)
```

```

# Calculate total runs and final score per innings
df["total_runs"] = df["runs_of_bat"] + df["extras"]
total_runs = df.groupby("innings")[["runs_of_bat", "extras"]].sum()
total_runs["final_score"] = total_runs["runs_of_bat"] + total_runs["extras"]

# Count total wickets per innings (excluding "unknown")
total_wickets = df[df["wicket_type"] != "unknown"].groupby("innings")["wicket_type"].count()

# Map innings to team names
team_names = {1: "KKR", 2: "RCB"}

# Create match summary dictionary
match_summary = {
    team_names[inning]: f"{int(total_runs.loc[inning, 'final_score'])}/{total_wickets.get(inning, 0)}"
    for inning in team_names
}

# 📊 Display match summary and top performers
print("📋 Match No. 1 Summary")
print(f"📍 Venue: {venue}")

for inning in sorted(df["innings"].unique()):
    team = team_names[inning]
    print(f"\n🏏 Top 3 Batsmen for {team} (Inning {inning}):")
    for batsman, runs in top_batsmen(df, inning).items():
        print(f" - {batsman}: {runs} runs")

    print(f"\n🏏 Top 3 Bowlers against {team}:")
    for bowler, wickets in top_bowlers(df, inning).items():
        print(f" - {bowler}: {wickets} wickets")

# 📋 Display final match score
print("\n📋 Final Match Summary:")
for team, score in match_summary.items():
    print(f" - {team}: {score}")

# 🏆 Determine the winner
kkr_runs = int(total_runs.loc[1, "final_score"])
rcb_runs = int(total_runs.loc[2, "final_score"])
rcb_wickets = total_wickets.get(2, 0)

if rcb_runs > kkr_runs:

```

```

print(f"\n🏏 RCB won the match by {10 - rcb_wickets} wickets!")
elif kkr_runs > rcb_runs:
    print(f"\n🏏 KKR won the match by {kkr_runs - rcb_runs} runs!")
else:
    print("\n🏏 The match was tied!")

```



Match No. 1 Summary



Venue: Eden Gardens, Kolkata



Top 3 Batsmen for KKR (Inning 1):

- Rahane: 56 runs
- Narine: 44 runs
- Angkrish Raghuvanshi: 30 runs



Top 3 Bowlers against KKR:

- Krunal Pandya: 3 wickets
- Hazlewood: 2 wickets
- Rasikh Salam: 1 wickets



Top 3 Batsmen for RCB (Inning 2):

- Kohli: 59 runs
- Phil Salt: 56 runs
- Rajat Patidar: 34 runs



Top 3 Bowlers against RCB:

- Narine: 1 wickets
- Vaibhav Arora: 1 wickets
- Varun Chakaravarthy: 1 wickets



Final Match Summary:

- KKR: 174/8
- RCB: 177/3



RCB won the match by 7 wickets!

## TEAM WISE STATISTICS

In [200...

```

import pandas as pd

# List unique teams
teams = df["batting_team"].unique()

# Dictionary to store summary for each team

```

```

summary = {}

for team in teams:
    team_data = df[df["batting_team"] == team]

    # Grouping by striker to get highest individual score
    highest_score = team_data.groupby("striker")["runs_of_bat"].sum().max()

    summary[team] = {
        "Total Runs": int(team_data["runs_of_bat"].sum() + team_data["extras"].sum()),
        "Total Wickets": team_data[team_data["player_dismissed"] != "unknown"]["player_dismissed"].count(),
        "Total Overs Played": team_data["over"].nunique(),
        "Highest Individual Score": int(highest_score),
        "Fours Hit": (team_data["runs_of_bat"] == 4).sum(),
        "Sixes Hit": (team_data["runs_of_bat"] == 6).sum(),
        "Extras Given": int(team_data["extras"].sum())
    }

# Convert to DataFrame for a clean tabular display
comparison_df = pd.DataFrame(summary).T

# Display the team-wise statistics
print(" 🏏 Team wise statistics:\n")
print(comparison_df)

```

🏏 Team wise statistics:

	Total Runs	Total Wickets	Total Overs Played	Highest Individual Score \
KKR	174	8	20	56
RCB	177	3	17	59

	Fours Hit	Sixes Hit	Extras Given
KKR	18	8	6
RCB	21	7	3

## BAR GRAPH REPRESENTATION OF OVERS vs TOTAL RUNS SCORED WITH WICKETS FALLEN

In [187...

```

import pandas as pd
import matplotlib.pyplot as plt

# Convert over column to int if needed
df["over"] = df["over"].astype(int)

```

```

# Group by innings and over, summing runs
runs_per_over = df.groupby(["innings", "over"])[ "runs_of_bat"].sum()

# Count wickets per over (exclude 'unknown' if present)
wickets_per_over = df[df["wicket_type"].notna() & (df["wicket_type"] != "unknown")].groupby(["innings", "over"])[ "wicket_type"].count()

# Define colors
colors = {1: "purple", 2: "red"}

# Create separate graphs for each inning
fig, axes = plt.subplots(nrows=2, figsize=(10, 8))

for i, inning in enumerate(sorted(df["innings"].unique())):
    inning_data = runs_per_over.xs(inning, level="innings")
    wickets_data = wickets_per_over.xs(inning, level="innings") if inning in wickets_per_over.index.levels[0] else pd.Series()

    axes[i].bar(inning_data.index, inning_data.values, color=colors[inning])

    # Plot wickets markers just above the bar height for those overs
    wicket_x = wickets_data.index
    wicket_y = [inning_data.loc[o] + 0.5 for o in wicket_x] # add small offset to bar height

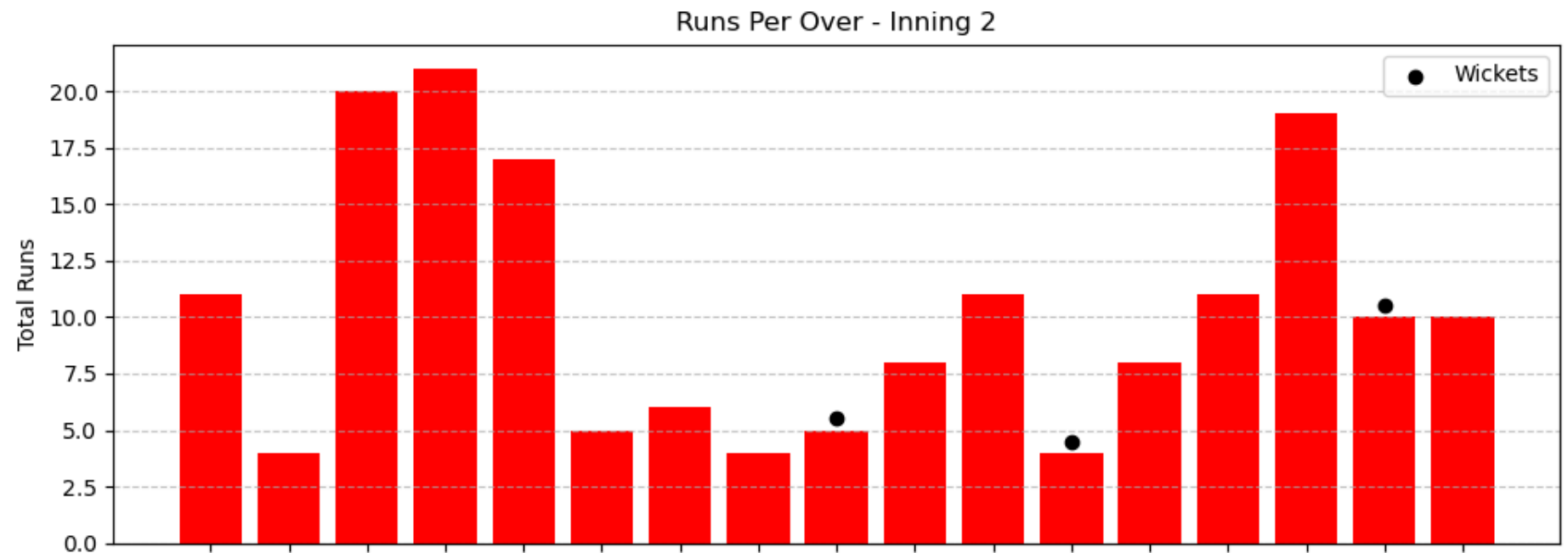
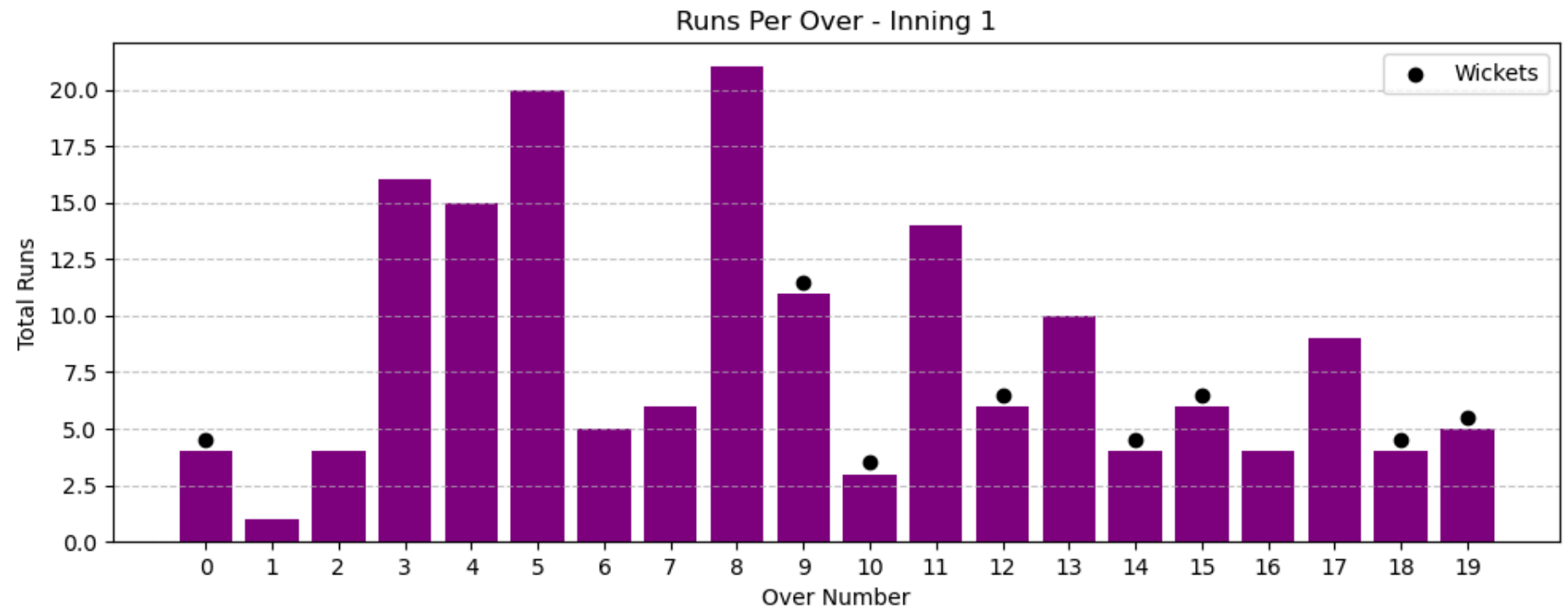
    axes[i].scatter(wicket_x, wicket_y, color="black", marker="o", label="Wickets", zorder=5)

    axes[i].set_xlabel("Over Number")
    axes[i].set_ylabel("Total Runs")
    axes[i].set_title(f"Runs Per Over - Inning {inning}")
    axes[i].set_xticks(inning_data.index)
    axes[i].grid(axis="y", linestyle="--", alpha=0.7)
    axes[i].legend()

plt.tight_layout()
plt.show()

```





0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Over Number

## PIE CHART REPRESENTATION OF MATCH EVENTS

In [134...

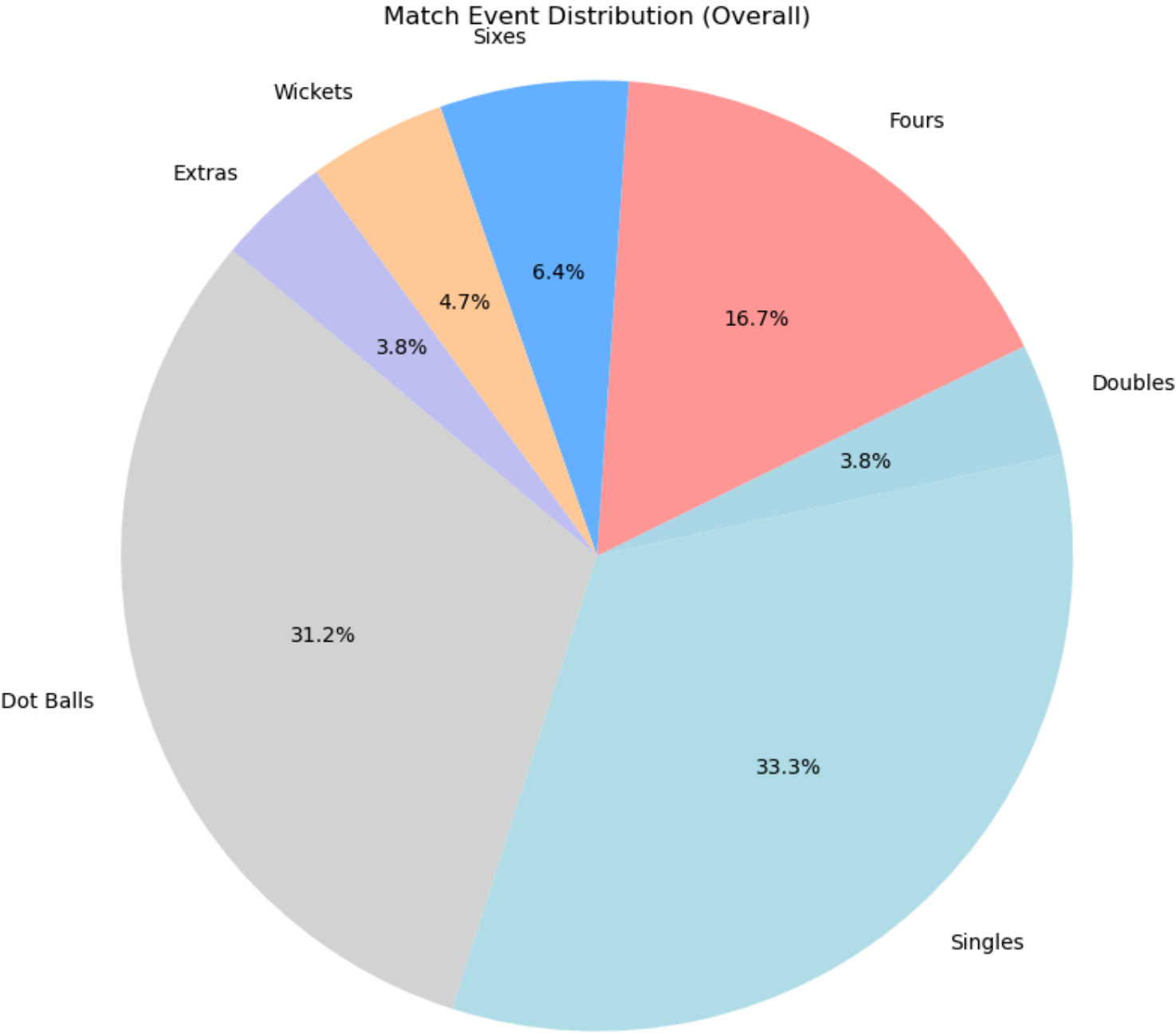
```
import pandas as pd
import matplotlib.pyplot as plt

# Ensure numeric conversion and missing value handling
df["runs_of_bat"] = pd.to_numeric(df["runs_of_bat"], errors="coerce").fillna(0)
df["extras"] = pd.to_numeric(df["extras"], errors="coerce").fillna(0)
df["wicket_type"] = df["wicket_type"].fillna("unknown")

# Count events
dot_balls = ((df["runs_of_bat"] == 0) & (df["extras"] == 0)).sum()
singles = (df["runs_of_bat"] == 1).sum()
doubles = (df["runs_of_bat"] == 2).sum()
fours = (df["runs_of_bat"] == 4).sum()
sixes = (df["runs_of_bat"] == 6).sum()
wickets = (df["wicket_type"] != "unknown").sum()
extras = (df["extras"] > 0).sum()

# Prepare data for pie chart
labels = ['Dot Balls', 'Singles', 'Doubles', 'Fours', 'Sixes', 'Wickets', 'Extras']
sizes = [dot_balls, singles, doubles, fours, sixes, wickets, extras]
colors = ['#d3d3d3', '#b0e0e6', '#add8e6', '#ff9999', '#66b3ff', '#ffcc99', '#c2c2f0']

# Plotting
plt.figure(figsize=(9, 9))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140, colors=colors)
plt.title("Match Event Distribution (Overall)")
plt.axis('equal') # Equal aspect ratio ensures pie is round
plt.show()
```



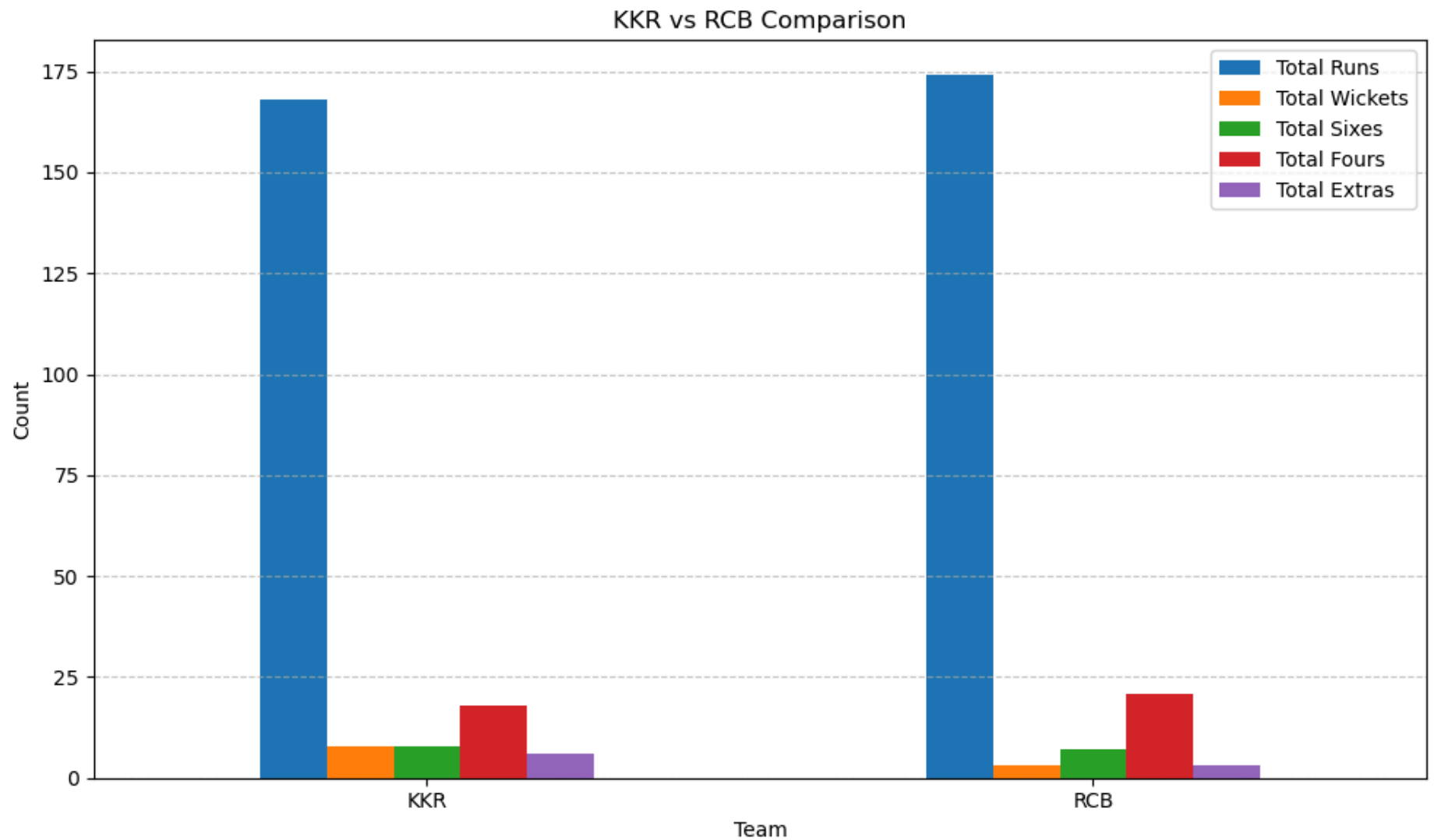
## KKR vs RCB COMPARISON

In [198...

```
import pandas as pd
import matplotlib.pyplot as plt

# Aggregate key statistics per team
team_stats = df.groupby("batting_team").agg(
    total_runs=pd.NamedAgg(column="runs_of_bat", aggfunc="sum"),
    total_wickets=pd.NamedAgg(column="player_dismissed", aggfunc=lambda x: x[x != "unknown"].count()),
    total_sixes=pd.NamedAgg(column="runs_of_bat", aggfunc=lambda x: (x == 6).sum()),
    total_fours=pd.NamedAgg(column="runs_of_bat", aggfunc=lambda x: (x == 4).sum()),
    total_extras=pd.NamedAgg(column="extras", aggfunc="sum")
)

# Plot the comparison chart
team_stats.plot(kind="bar", figsize=(10, 6))
plt.title("KKR vs RCB Comparison")
plt.xlabel("Team")
plt.ylabel("Count")
plt.legend(["Total Runs", "Total Wickets", "Total Sixes", "Total Fours", "Total Extras"])
plt.xticks(rotation=0)
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```



## RUNS SCORED vs OVERS

```
In [181... import pandas as pd
import matplotlib.pyplot as plt

# Ensure runs_of_bat and extras are numeric
```

```
df["runs_of_bat"] = pd.to_numeric(df["runs_of_bat"], errors="coerce").fillna(0)
df["extras"] = pd.to_numeric(df["extras"], errors="coerce").fillna(0)
df["total_runs"] = df["runs_of_bat"] + df["extras"]

# Identify teams
teams = df["batting_team"].unique()
team1, team2 = teams[0], teams[1]

# Split data by team
team1_data = df[df["batting_team"] == team1]
team2_data = df[df["batting_team"] == team2]

# Calculate cumulative runs by over
team1_runs = team1_data.groupby("over")["total_runs"].sum().cumsum()
team2_runs = team2_data.groupby("over")["total_runs"].sum().cumsum()

# Identify fall of wickets by over
team1_wickets = team1_data[team1_data["player_dismissed"] != "unknown"].groupby("over")["player_dismissed"].count()
team2_wickets = team2_data[team2_data["player_dismissed"] != "unknown"].groupby("over")["player_dismissed"].count()

# Plot the graph
plt.figure(figsize=(12, 6))

# Team 1 Line and wicket markers
plt.plot(team1_runs.index, team1_runs.values, color="blue", linestyle="--", linewidth=2, label=f"{team1} Runs")
plt.scatter(team1_wickets.index, team1_runs[team1_wickets.index], color="purple", marker='o', s=100, label=f"{team1} Wickets")

# Team 2 Line and wicket markers
plt.plot(team2_runs.index, team2_runs.values, color="green", linestyle="-. ", linewidth=2, label=f"{team2} Runs")
plt.scatter(team2_wickets.index, team2_runs[team2_wickets.index], color="red", marker='o', s=100, label=f"{team2} Wickets")

# Final plot labels
plt.title("Runs Scored vs Overs with Wicket Markers", fontsize=14)
plt.xlabel("Over", fontsize=12)
plt.ylabel("Cumulative Runs", fontsize=12)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

