**Tennis Dataset:**

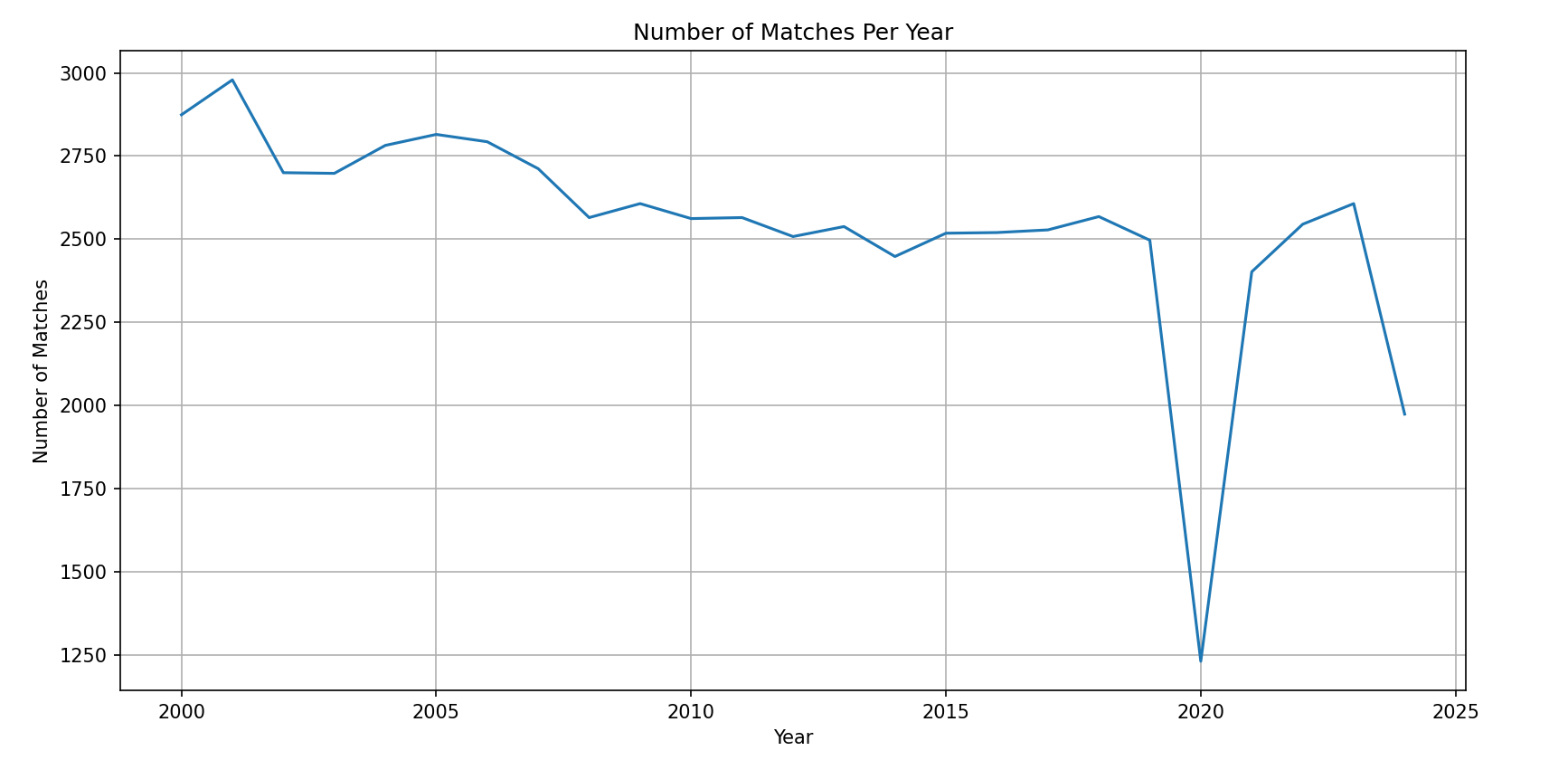
Dataset Link:

<https://www.kaggle.com/datasets/dissfya/atp-tennis-2000-2023daily-pull>

1. **Number of Matches Per Year**

This code calculates and visualizes the number of tennis matches per year. It first converts the 'Date' column to datetime format and extracts the year. It then counts the matches for each year and plots the data using a line chart to show trends over time.

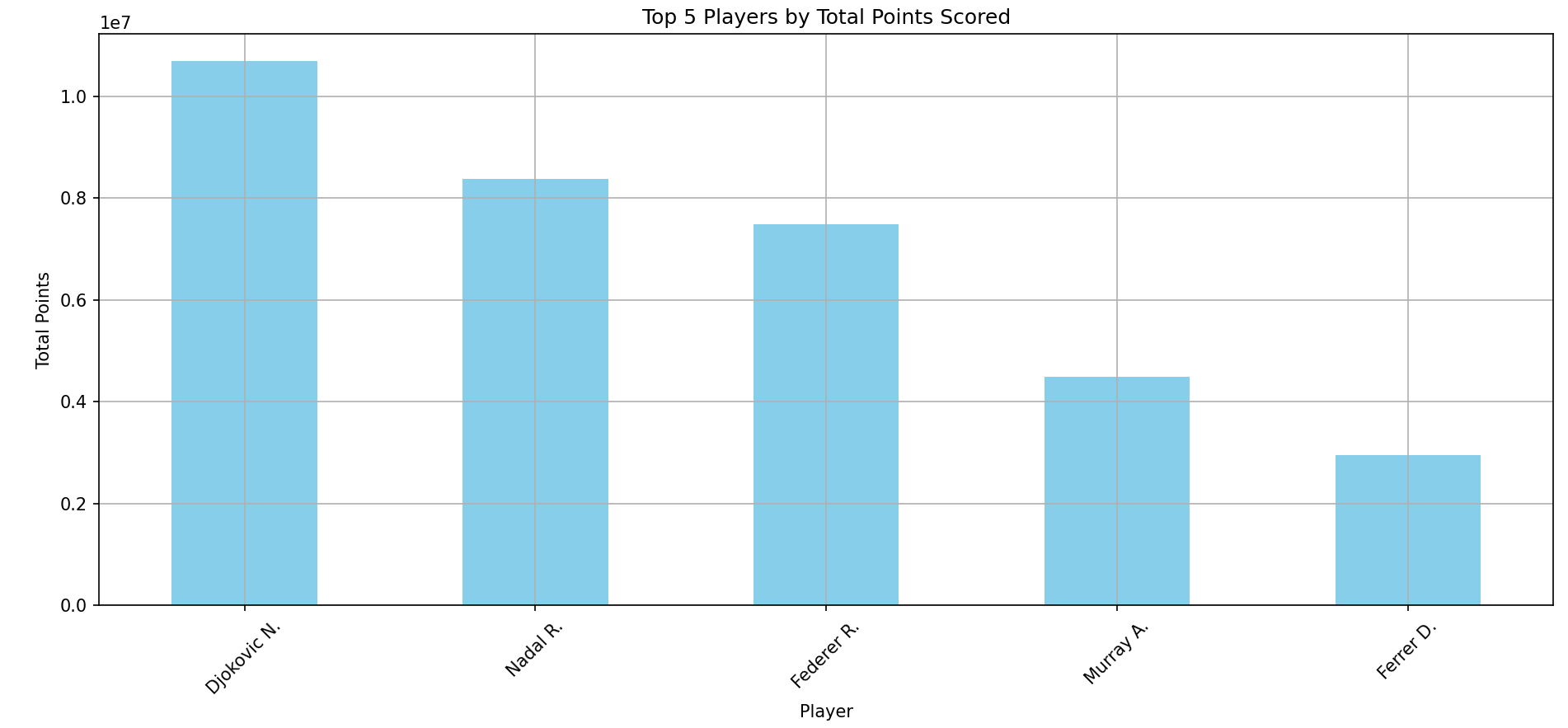
import pandas as pd  
import matplotlib.pyplot as plt  
  
# Load the data  
df = pd.read\_csv('atp\_tennis.csv')  
  
# Convert 'Date' to datetime  
df['Date'] = pd.to\_datetime(df['Date'])  
df['Year'] = df['Date'].dt.year  
  
# Count number of matches per year  
matches\_per\_year = df.groupby('Year').size()  
  
plt.figure(figsize=(12, 6))  
matches\_per\_year.plot(kind='line')  
plt.title('Number of Matches Per Year')  
plt.xlabel('Year')  
plt.ylabel('Number of Matches')  
plt.grid(True)  
plt.show()

**Output:**

1. **Total Points Scored by Each Player**

This code aggregates total points scored by each player from both Player\_1 and Player\_2 columns, combines and sums these points, then identifies the top 5 players with the highest total points.

# Aggregate points scored by each player  
player1\_points = df.groupby('Player\_1')['Pts\_1'].sum()  
player2\_points = df.groupby('Player\_2')['Pts\_2'].sum()  
  
# Combine points  
total\_points = pd.concat([player1\_points, player2\_points], axis=1).fillna(0)  
total\_points.columns = ['Player1Points', 'Player2Points']  
total\_points['TotalPoints'] = total\_points['Player1Points'] + total\_points['Player2Points']  
  
# Sort values and pick top 5 players  
top\_5\_players = total\_points.sort\_values(by='TotalPoints', ascending=False).head(5)  
  
# Plotting  
plt.figure(figsize=(10, 6))  
top\_5\_players['TotalPoints'].plot(kind='bar', color='skyblue')  
plt.title('Top 5 Players by Total Points Scored')  
plt.xlabel('Player')  
plt.ylabel('Total Points')  
plt.xticks(rotation=45)  
plt.grid(True)  
  
# Show plot  
plt.tight\_layout()  
plt.show()

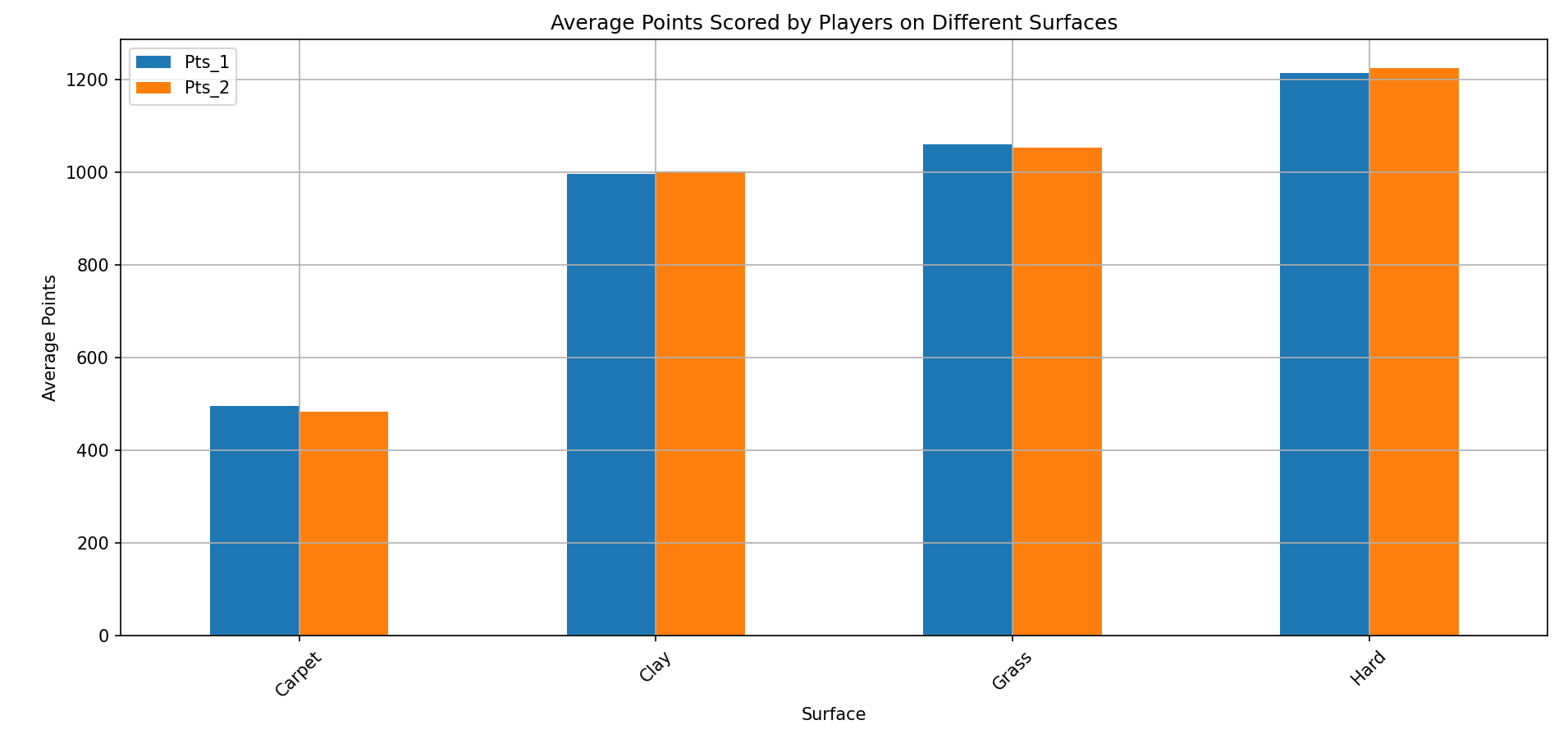
**Output:** 

1. **Average Points by Court Surface**

This visualization will show the average points scored by players on different court surfaces, broken down by Player 1 and Player 2.

# Group by surface and calculate mean points for both players  
avg\_points\_surface = df.groupby('Surface')[['Pts\_1', 'Pts\_2']].mean()  
  
# Plot  
plt.figure(figsize=(10, 6))  
avg\_points\_surface.plot(kind='bar', figsize=(10, 6), color=['#1f77b4', '#ff7f0e'])  
plt.title('Average Points Scored by Players on Different Surfaces')  
plt.xlabel('Surface')  
plt.ylabel('Average Points')  
plt.xticks(rotation=45)  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

**Output:**

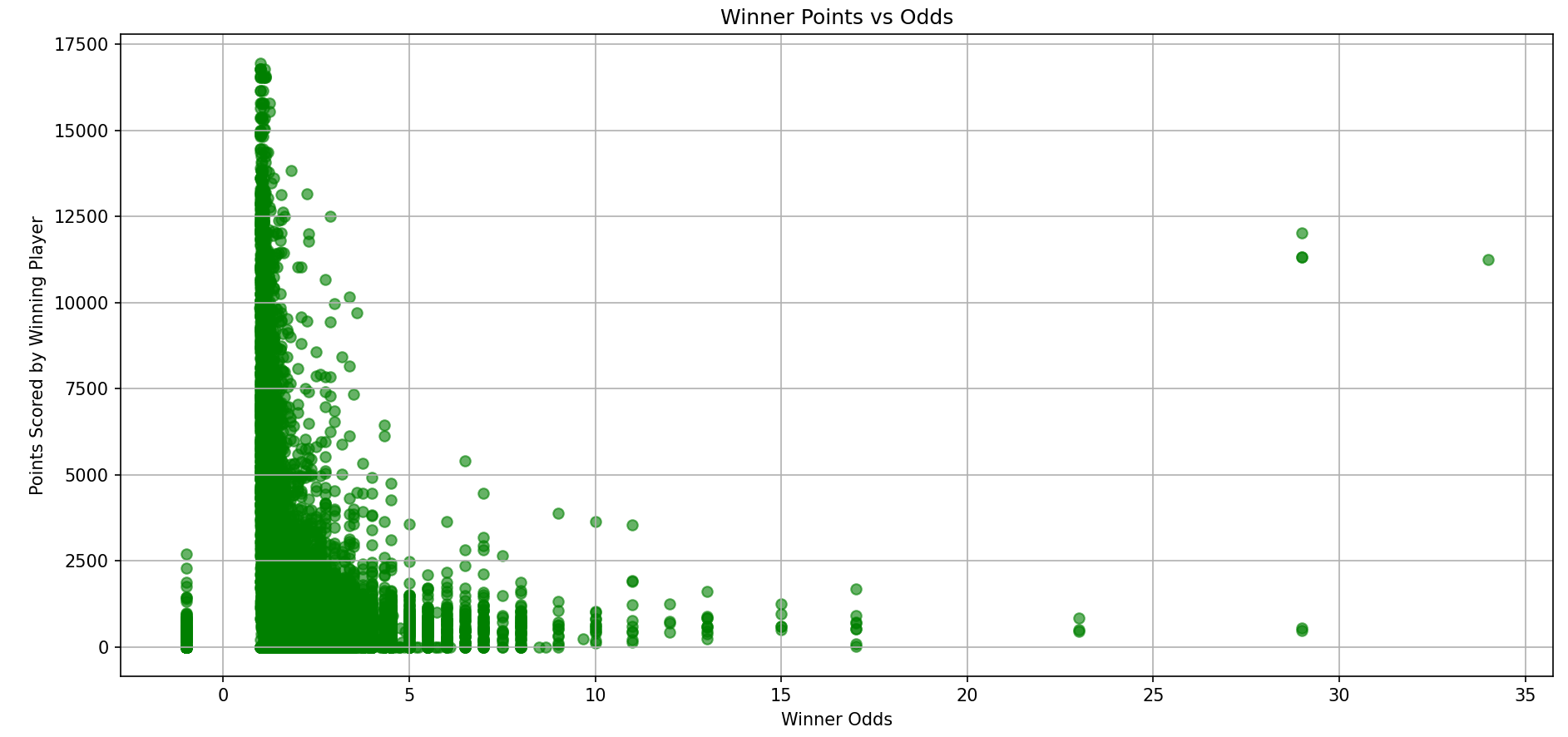


### ****Winning Player Points vs Odds****

### This plot compares how many points were scored by winning players based on their odds. It visualizes the relationship between points and betting odds.

# Filter data to only include the winner's points and odds  
winner\_data = df[df['Winner'] == df['Player\_1']][['Pts\_1', 'Odd\_1']].copy()  
winner\_data.rename(columns={'Pts\_1': 'WinnerPoints', 'Odd\_1': 'WinnerOdds'}, inplace=True)  
  
# Scatter plot of points vs odds  
plt.figure(figsize=(10, 6))  
plt.scatter(winner\_data['WinnerOdds'], winner\_data['WinnerPoints'], alpha=0.6, color='green')  
plt.title('Winner Points vs Odds')  
plt.xlabel('Winner Odds')  
plt.ylabel('Points Scored by Winning Player')  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

**Output:**

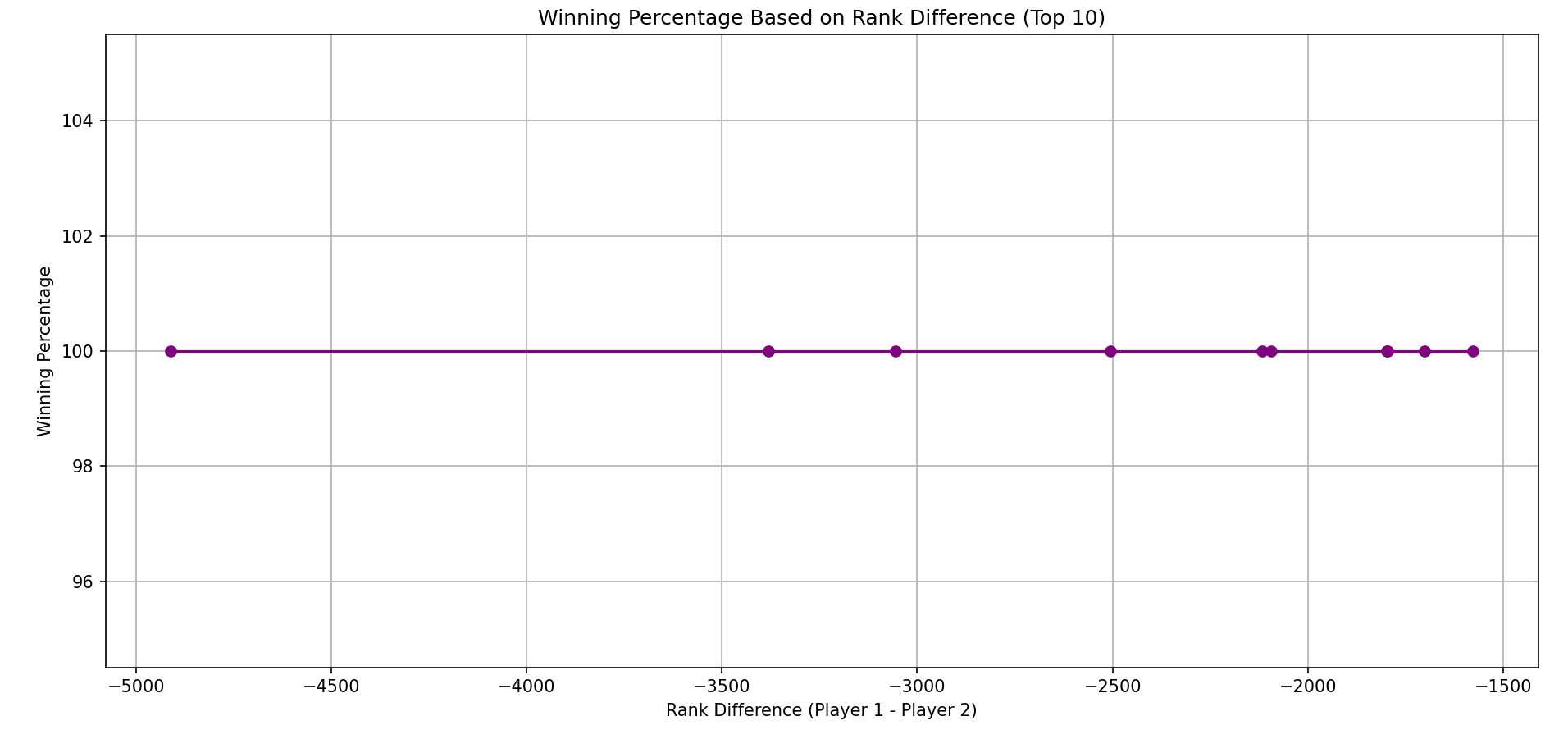


### ****Rank Difference and Winning Percentage****

### This visualization explores how rank difference between two players affects the percentage of times the higher-ranked player wins.

# Create a column for rank difference  
df['RankDiff'] = df['Rank\_1'] - df['Rank\_2']  
  
# Calculate winning percentage based on rank difference  
win\_by\_rank\_diff = df.groupby('RankDiff').apply(lambda group: (group['Winner'] == group['Player\_1']).mean() \* 100)  
  
# Select only the first 10 rank differences  
win\_by\_rank\_diff = win\_by\_rank\_diff.head(10)  
  
# Plot  
plt.figure(figsize=(10, 6))  
win\_by\_rank\_diff.plot(kind='line', marker='o', color='purple')  
plt.title('Winning Percentage Based on Rank Difference (Top 10)')  
plt.xlabel('Rank Difference (Player 1 - Player 2)')  
plt.ylabel('Winning Percentage')  
plt.grid(True)  
plt.tight\_layout()  
plt.show()

**Output:**

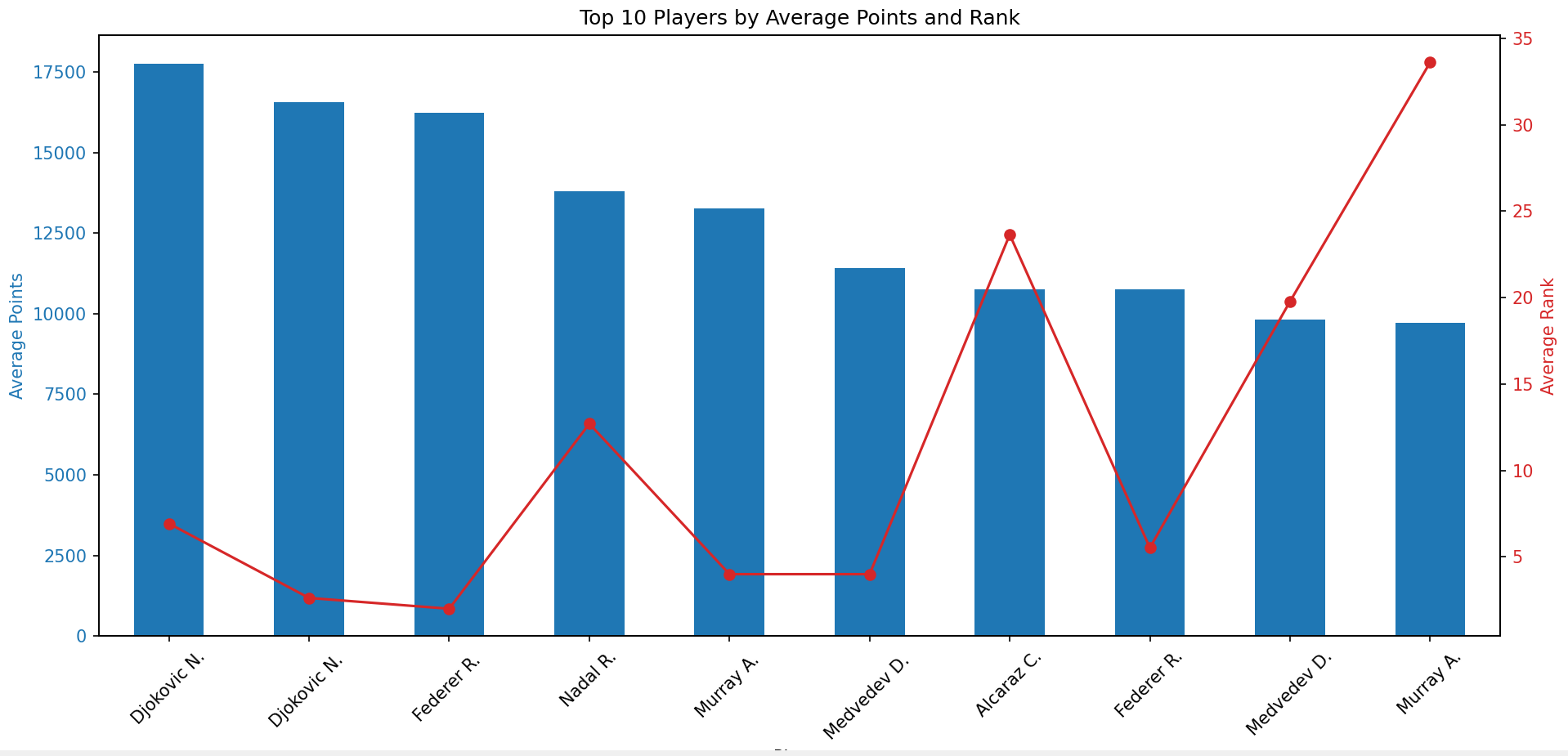


### Top 10 Players by Average Points Scored and Rank

### Visualize the top 10 players based on their average points scored, alongside their average rank.

# Calculate average points and rank for each player  
player\_avg\_stats = pd.concat([  
 df.groupby('Player\_1').agg({'Pts\_1': 'mean', 'Rank\_1': 'mean'}),  
 df.groupby('Player\_2').agg({'Pts\_2': 'mean', 'Rank\_2': 'mean'})  
], axis=1).fillna(0)  
  
player\_avg\_stats.columns = ['Avg\_Pts\_1', 'Avg\_Rank\_1', 'Avg\_Pts\_2', 'Avg\_Rank\_2']  
player\_avg\_stats['Avg\_Pts'] = player\_avg\_stats[['Avg\_Pts\_1', 'Avg\_Pts\_2']].sum(axis=1)  
player\_avg\_stats['Avg\_Rank'] = player\_avg\_stats[['Avg\_Rank\_1', 'Avg\_Rank\_2']].mean(axis=1)  
  
top\_10\_players = player\_avg\_stats.sort\_values(by='Avg\_Pts', ascending=False).head(10)  
  
# Plotting  
fig, ax1 = plt.subplots(figsize=(14, 8))  
  
color = 'tab:blue'  
ax1.set\_xlabel('Player')  
ax1.set\_ylabel('Average Points', color=color)  
top\_10\_players['Avg\_Pts'].plot(kind='bar', color=color, ax=ax1)  
ax1.tick\_params(axis='y', labelcolor=color)  
ax1.set\_xticklabels(top\_10\_players.index, rotation=45)  
  
ax2 = ax1.twinx()  
color = 'tab:red'  
ax2.set\_ylabel('Average Rank', color=color)  
top\_10\_players['Avg\_Rank'].plot(kind='line', color=color, marker='o', ax=ax2)  
ax2.tick\_params(axis='y', labelcolor=color)  
  
plt.title('Top 10 Players by Average Points and Rank')  
fig.tight\_layout()  
plt.show()

**Output:**

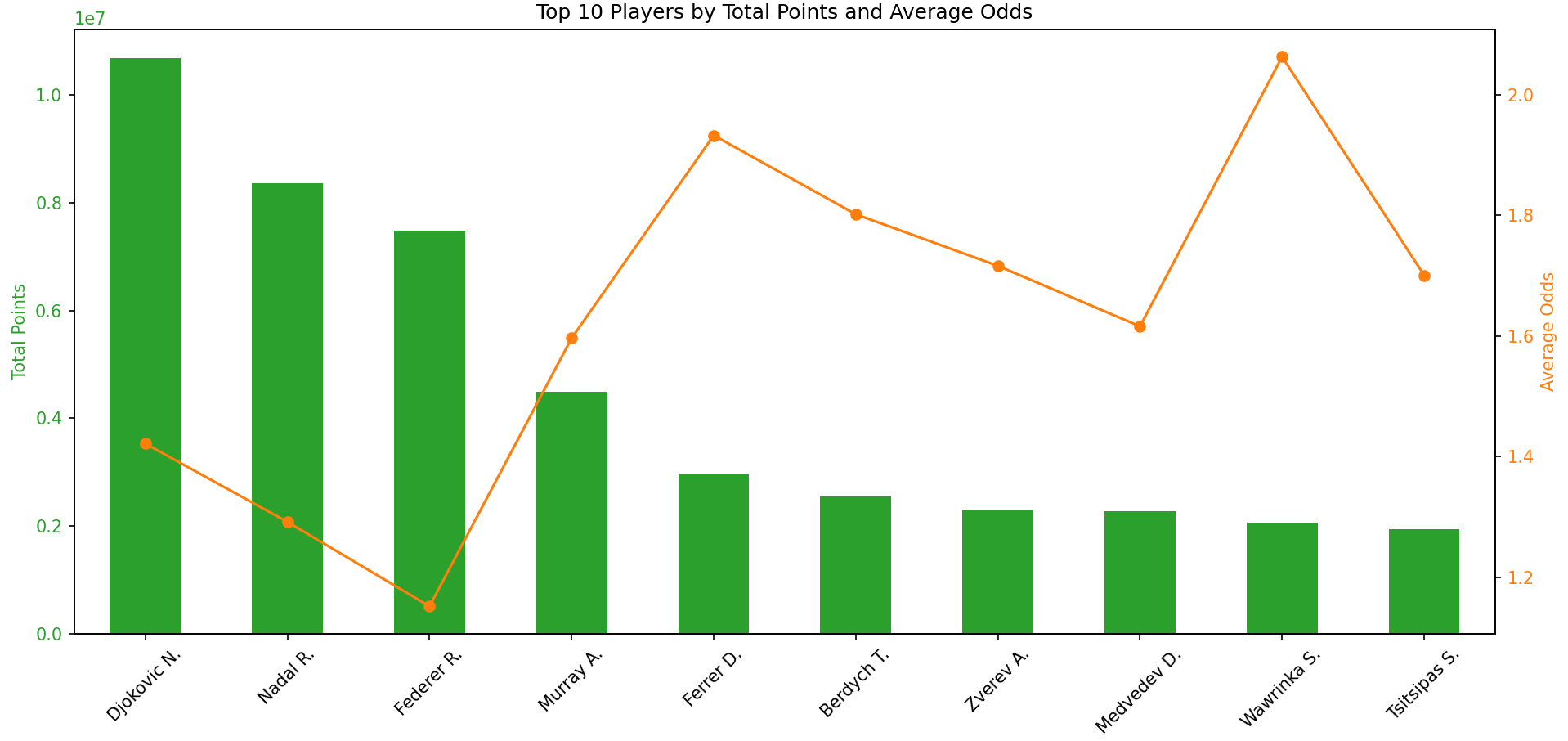


### Points Distribution by Surface and Court

Visualize the total points scored on different surfaces and courts.

# Calculate total points and average odds  
player\_stats = pd.concat([  
 df.groupby('Player\_1').agg({'Pts\_1': 'sum', 'Odd\_1': 'mean'}),  
 df.groupby('Player\_2').agg({'Pts\_2': 'sum', 'Odd\_2': 'mean'})  
], axis=1).fillna(0)  
  
player\_stats.columns = ['Total\_Points\_1', 'Avg\_Odd\_1', 'Total\_Points\_2', 'Avg\_Odd\_2']  
player\_stats['Total\_Points'] = player\_stats[['Total\_Points\_1', 'Total\_Points\_2']].sum(axis=1)  
player\_stats['Avg\_Odd'] = player\_stats[['Avg\_Odd\_1', 'Avg\_Odd\_2']].mean(axis=1)  
  
top\_10\_players = player\_stats.sort\_values(by='Total\_Points', ascending=False).head(10)  
  
# Plotting  
fig, ax1 = plt.subplots(figsize=(14, 8))  
  
color = 'tab:green'  
ax1.set\_xlabel('Player')  
ax1.set\_ylabel('Total Points', color=color)  
top\_10\_players['Total\_Points'].plot(kind='bar', color=color, ax=ax1)  
ax1.tick\_params(axis='y', labelcolor=color)  
ax1.set\_xticklabels(top\_10\_players.index, rotation=45)  
  
ax2 = ax1.twinx()  
color = 'tab:orange'  
ax2.set\_ylabel('Average Odds', color=color)  
top\_10\_players['Avg\_Odd'].plot(kind='line', color=color, marker='o', ax=ax2)  
ax2.tick\_params(axis='y', labelcolor=color)  
  
plt.title('Top 10 Players by Total Points and Average Odds')  
fig.tight\_layout()  
plt.show()

**Output:**

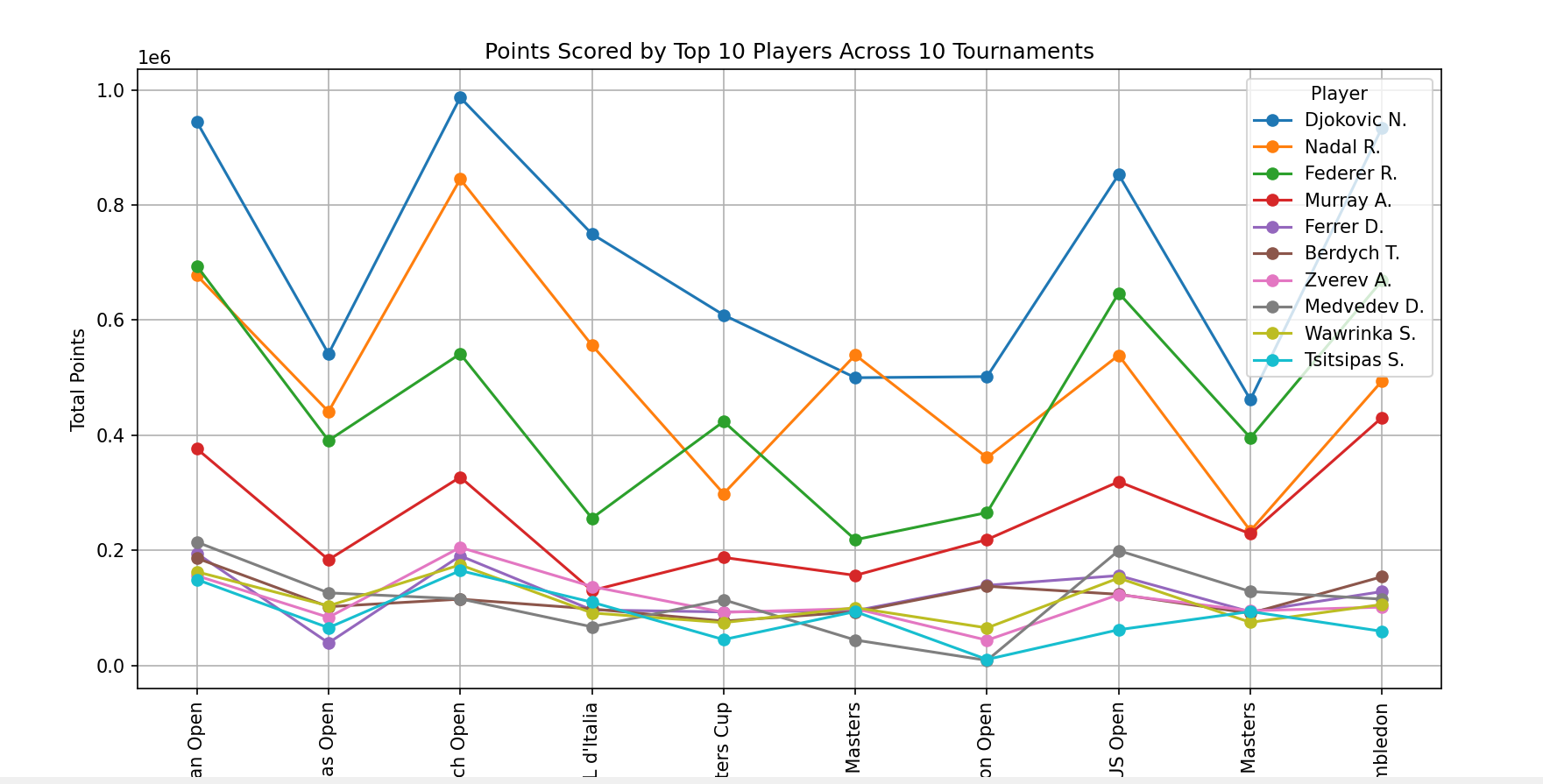
****

### Comparison of Points Scored by Top 10 Players Across 10 Tournaments

### Visualize the total points scored by the top 10 players across different tournaments.

import matplotlib.pyplot as plt  
import pandas as pd  
  
# Calculate total points by player and tournament  
tournament\_player\_points = pd.concat([  
 df.groupby(['Tournament', 'Player\_1'])[['Pts\_1']].sum().reset\_index(),  
 df.groupby(['Tournament', 'Player\_2'])[['Pts\_2']].sum().reset\_index().rename(columns={'Pts\_2': 'Pts\_1', 'Player\_2': 'Player\_1'})  
], axis=0).fillna(0)  
  
tournament\_player\_points = tournament\_player\_points.groupby(['Tournament', 'Player\_1']).sum().reset\_index()  
  
# Get the top 10 players by total points  
top\_10\_players = tournament\_player\_points.groupby('Player\_1')['Pts\_1'].sum().nlargest(10).index  
  
# Filter for top 10 players  
top\_10\_data = tournament\_player\_points[tournament\_player\_points['Player\_1'].isin(top\_10\_players)]  
  
# Get the top 10 tournaments by total points scored  
top\_10\_tournaments = top\_10\_data.groupby('Tournament')['Pts\_1'].sum().nlargest(10).index  
  
# Filter for top 10 tournaments  
top\_10\_data = top\_10\_data[top\_10\_data['Tournament'].isin(top\_10\_tournaments)]  
  
# Plotting  
plt.figure(figsize=(14, 8))  
for player in top\_10\_players:  
 subset = top\_10\_data[top\_10\_data['Player\_1'] == player]  
 plt.plot(subset['Tournament'], subset['Pts\_1'], marker='o', label=player)  
  
plt.title('Points Scored by Top 10 Players Across 10 Tournaments')  
plt.xlabel('Tournament')  
plt.ylabel('Total Points')  
plt.xticks(rotation=90)  
plt.legend(title='Player')  
plt.grid(True)  
plt.show()

**Output:**

****

For giving path:

import pandas as pd

df = pd.read\_csv('abfss://aamna@youexcel.dfs.core.windows.net/football/goalscorers.csv',

storage\_options = {'account\_key':'/dhYW+OiOeruQ2q2lRlOVQHX+VYV50A1MDZYASkOS6BP6NqNMeL12c2uqhhUzDp92mNhSurH6uI++AStKYyqpQ=='})

display(df)

For converting csv to sql

(FOR TRANSFERING DATA FROM CSV TO SQL)

CREATE TABLE [dbo].[department]

(

Name nvarchar(218),

Age nvarchar(218),

City nvarchar(218)

)

WITH

(

DISTRIBUTION – ROUND\_ROBIN,

CLUSTERED COLUMNSTORE INDEX

--HEAP

)

GO

COPY INTO [dbo].[dept]

(Name 1, Age 2, City 3)

From 'https://youexcel.dfs.core.windows.net/aamna/mid.csv'

WITH

(

    FILE\_TYPE = 'CSV',

    MAXERRORS = 0,

    IDENTITY\_INSERT = 'OFF'

)

SELECT \* FROM [dbo].[dept]