Tennis

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

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)

1. Number of Matches Per Year
2. import pandas as pd  
   import matplotlib.pyplot as plt  
     
   # Load the data  
   df = pd.read\_csv('atp-tennis-2000-2023daily-pull.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()
3. Total Points Scored by Each Player

# 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']  
total\_points = total\_points.sort\_values(by='TotalPoints', ascending=False)  
  
plt.figure(figsize=(14, 8))  
total\_points['TotalPoints'].plot(kind='bar')  
plt.title('Total Points Scored by Each Player')  
plt.xlabel('Player')  
plt.ylabel('Total Points')  
plt.xticks(rotation=90)  
plt.grid(True)  
plt.show()

1. Win Percentage of Each Player
2. # Count wins for each player  
   player1\_wins = df[df['Winner'] == df['Player\_1']].groupby('Player\_1').size()  
   player2\_wins = df[df['Winner'] == df['Player\_2']].groupby('Player\_2').size()  
     
   # Count total matches played by each player  
   player1\_matches = df.groupby('Player\_1').size()  
   player2\_matches = df.groupby('Player\_2').size()  
   total\_matches = pd.concat([player1\_matches, player2\_matches], axis=1).fillna(0)  
   total\_matches.columns = ['Player1Matches', 'Player2Matches']  
   total\_matches['TotalMatches'] = total\_matches['Player1Matches'] + total\_matches['Player2Matches']  
     
   # Calculate win percentages  
   player\_wins = pd.concat([player1\_wins, player2\_wins], axis=1).fillna(0)  
   player\_wins.columns = ['PlayerWins', 'PlayerWins']  
   player\_wins['TotalWins'] = player\_wins['PlayerWins'] + player\_wins['PlayerWins']  
   player\_wins['WinPercentage'] = (player\_wins['TotalWins'] / total\_matches['TotalMatches']) \* 100  
   player\_wins = player\_wins.sort\_values(by='WinPercentage', ascending=False)  
     
   plt.figure(figsize=(14, 8))  
   player\_wins['WinPercentage'].plot(kind='bar')  
   plt.title('Win Percentage of Each Player')  
   plt.xlabel('Player')  
   plt.ylabel('Win Percentage')  
   plt.xticks(rotation=90)  
   plt.grid(True)  
   plt.show()
3. Average Odds for Winners and Losers

# Calculate average odds for winners and losers  
average\_odds = df.groupby('Winner').agg({'Odd\_1': 'mean', 'Odd\_2': 'mean'})  
average\_odds.columns = ['AverageOdd1', 'AverageOdd2']  
  
plt.figure(figsize=(12, 6))  
average\_odds[['AverageOdd1', 'AverageOdd2']].plot(kind='bar', stacked=True)  
plt.title('Average Odds for Winners and Losers')  
plt.xlabel('Winner')  
plt.ylabel('Average Odds')  
plt.xticks(rotation=90)  
plt.grid(True)  
plt.show()

1. Number of Matches by Surface Type
2. # Count number of matches by surface type  
   matches\_by\_surface = df['Surface'].value\_counts()  
     
   plt.figure(figsize=(12, 6))  
   matches\_by\_surface.plot(kind='bar')  
   plt.title('Number of Matches by Surface Type')  
   plt.xlabel('Surface Type')  
   plt.ylabel('Number of Matches')  
   plt.grid(True)  
   plt.show()
3. Tournament Win Counts

# Count number of wins for each tournament  
tournament\_wins = df['Tournament'].value\_counts()  
  
plt.figure(figsize=(12, 6))  
tournament\_wins.plot(kind='bar')  
plt.title('Number of Wins by Tournament')  
plt.xlabel('Tournament')  
plt.ylabel('Number of Wins')  
plt.xticks(rotation=90)  
plt.grid(True)  
plt.show()

1. Distribution of Match Scoress
2. # Split the scores and convert to numeric  
   score\_split = df['Score'].str.split('-', expand=True)  
   df['Score1'] = pd.to\_numeric(score\_split[0], errors='coerce')  
   df['Score2'] = pd.to\_numeric(score\_split[1], errors='coerce')  
     
   plt.figure(figsize=(14, 8))  
   plt.hist(df['Score1'].dropna(), bins=20, alpha=0.5, label='Player 1 Score')  
   plt.hist(df['Score2'].dropna(), bins=20, alpha=0.5, label='Player 2 Score')  
   plt.title('Distribution of Match Scores')  
   plt.xlabel('Score')  
   plt.ylabel('Frequency')  
   plt.legend()  
   plt.grid(True)  
   plt.show()