* NAME :- PRANAV MURKAR
* PRN:-202401100066
* ROLL NO:-CS5-06

1. **Find the Top 5 Players by Overall Rating**

Solution:- import pandas as pd

df.sort\_values(by='Overall', ascending=False)[['Name', 'Overall']].head(5)

1. **Calculate the Average Age of Players**

Solution:- df['Age'].mean()

1. **Find the Player with the Highest Potential**

Solution:- df.loc[df['Potential'].idxmax(), ['Name', 'Potential']]

1. **Count the Number of Players by Nationality**

Solution:- df['Nationality'].value\_counts()

1. **Find Clubs with the Most Players Represented**

Solution:- df['Club'].value\_counts().head(10)

1. **Average Wage of Players by Club**

Solution:- df.groupby('Club')['Wage'].mean().sort\_values(ascending=False)

1. **Filter Players Older than 35**

Solution:- df[df['Age'] > 35][['Name', 'Age']]

1. **Find the Player with the Highest Release Clause**

Solution:- df.loc[df['Release Clause'].idxmax(), ['Name', 'Release Clause']]

1. **Plot a Histogram of Player Ages**

Solution:- import matplotlib.pyplot as plt

df['Age'].hist(bins=20)

plt.xlabel('Age')

plt.ylabel('Number of Players')

plt.title('Distribution of Player Ages')

plt.show()

1. **Identify the Youngest Player with an Overall > 85**

Solution:- top\_players = df[df['Overall'] > 85]

top\_players.loc[top\_players['Age'].idxmin()]

1. **Find Players with Same Overall and Potential**

Solution:- df[df['Overall'] == df['Potential']][['Name', 'Overall', 'Potential']]

1. **Players with Value over 100M**

Solution:- df[df['Value'] > 100\_000\_000][['Name', 'Value']]

1. **Most Popular Position**

Solution:- df['Position'].value\_counts().idxmax()

1. **Top 10 Goalkeepers by Overall Rating**

Solution:- gk\_df = df[df['Position'] == 'GK']

gk\_df.sort\_values(by='Overall', ascending=False)[['Name', 'Overall']].head(10)

1. **Average Strength Attribute by Position**

Solution:- df.groupby('Position')['Strength'].mean().sort\_values(ascending=False)

1. **Find Left-footed Players with High Potential**

Solution:- df[df['Preferred Foot'] == 'Left'].sort\_values('Potential', ascending=False)[['Name', 'Potential']].head(10)

1. **Calculate Correlation between Age and Overall Rating**

**Solution:-** **df[['Age', 'Overall']].corr()**

1. **Top Nationalities by Average Player Value**

**Solution:-** **df.groupby('Nationality')['Value'].mean().sort\_values(ascending=False).head(10)**

1. **Identify Players Without a Club**

**Solution:-** **df[df['Club'].isna()][['Name']]**

1. **Find Players Whose Names Start with 'Cristiano'**

**Solution:-** **df[df['Name'].str.startswith('Cristiano')]**