Python Project

| Hussam AlNees | 0172310 |
| --- | --- |
| Husam AlKhateeb | 0193431 |



Dataframe Description

We used the three dataframes that were given to us(RecentProfessionalPlayers.csv, WorldCupMatches.csv, Players\_WorldCup\_Wins.csv)

We used the shape function to know the number of columns and rows in the dataframes:



RecentProfessionalPlayers.csv: 17588 rows and 14 columns

WorldCupMatches.csv: 4572 rows and 20 columns

Players\_WorldCup\_Wins.csv: 445 rows and 5 columns

The RecentProfessionalPlayers dataframe describes players, their clubs, their ratings and different stats and traits about them

WorldCupMatches dataframe describes world cup matches and their stage the goals of the home teams and away teams of half time and full time and the attendance in these matches and other traits of the matches

PlayerWorldCupWins dataframe describes the players who won the world cup and which years they won it in and the other appearances of the players who won the world cup.

Team Roles

Timeline:

1. We came up with the ideas about the graphs and plots on the 5th of January
2. We started working on the 6th of January
3. We finished working on the project on the 9th of January

Husam AlKhateeb made the first, third, fourth and fifth graphs.

Husam AlNees made the second, sixth and seventh graphs.

First Graph (Prefered Foot and Position Relation)

This graph describes the relation between the position of the player and the preferred foot used for the player.

1- We firstly determined the players who play in left positions and the right positions

LFP = players[(players.Club\_Position == 'LW') | (players.Club\_Position == 'LB') | (players.Club\_Position == 'LM') | (players.Club\_Position == 'LCB') | (players.Club\_Position == 'LS')]

RFP = players[(players.Club\_Position == 'RW') | (players.Club\_Position == 'RB') | (players.Club\_Position == 'RM') | (players.Club\_Position == 'RCB') | (players.Club\_Position == 'RS')]

2- Then we determined the number of players who play with the left and right foot in the left and right positions

RP\_RF = RFP[(RFP.Preffered\_Foot == 'Right')]

RP\_LF = RFP[(RFP.Preffered\_Foot == 'Left')]

RP\_RF\_LMAO = RP\_RF.Preffered\_Foot.to\_numpy()

RP\_LF\_LMAO = RP\_LF.Preffered\_Foot.to\_numpy()

lengthOfL1 = RP\_RF\_LMAO.size

lengthOfR1 = RP\_LF\_LMAO.size

all\_RP = lengthOfL+lengthOfR

LP\_RF = LFP[(LFP.Preffered\_Foot == 'Right')]

LP\_LF = LFP[(LFP.Preffered\_Foot == 'Left')]

LP\_RF\_LMAO = LP\_RF.Preffered\_Foot.to\_numpy()

LP\_LF\_LMAO = LP\_LF.Preffered\_Foot.to\_numpy()

lengthOfL = LP\_RF\_LMAO.size

lengthOfR = LP\_LF\_LMAO.size

all\_LP = lengthOfL+lengthOfR

allPos = all\_LP+all\_RP

3- then we got each percentage for each foot and position

perLPRF = (lengthOfL/allPos) \* 100

perLPLF = (lengthOfR/allPos) \* 100

perRPRF = (lengthOfL1/allPos) \* 100

perRPLF = (lengthOfR1/allPos) \* 100

width = 0.10 # the width of the bars: can also be len(x) sequence

labels = ['Left Footed Players', 'Right Footed Players']

4- We then used the plot commands to draw the graph

fig, ax = plt.subplots()

names = []

ax.bar(labels[0], perLPLF, color='blue',edgecolor='white', label = 'Left Position, Left Foot')

ax.bar(labels[0], perLPRF, bottom=perLPLF, color='deepskyblue',edgecolor='white', label = 'Left Position, Right Foot')

ax.bar(labels[1], perRPRF, color='red',edgecolor='white', label = 'Right Position, Right Foot')

ax.bar(labels[1], perRPLF, bottom=perRPRF, color='indianred',edgecolor='white', label = 'Right Position, Left Foot')

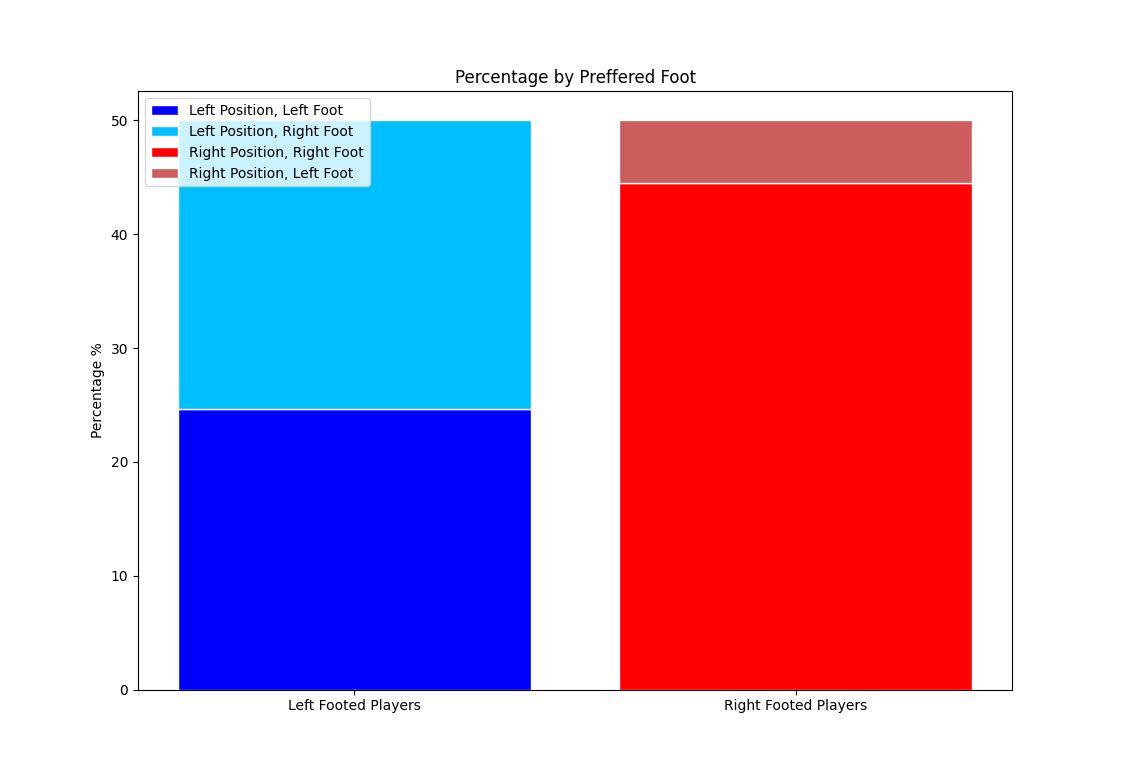
ax.set\_ylabel('Percentage %')

ax.set\_title('Percentage by Preffered Foot')

ax.legend()

plt.show()

**The Graph**



As we can see from the graph the left position players are almost equal in percentage of preferred foot, on the contrary a very small percentage of right position players use the left foot.

Second Graph (Percentage of World Cup Wins per continent)

This graph describes the percentage of World Cup wins for each continent.

1- We firstly determined the years for each world cup win for each country(this was done for each country)

Brazil = worldCupWins[(worldCupWins.Team == 'Brazil')]

years = Brazil.Years.to\_numpy()

ndYears = np.unique(years)

newYears = []

for i in ndYears:

newYears.append(i.replace(","," ").lower().split())

newnewYears = np.concatenate(newYears)

2- Then we determined the percentage for each country of World Cup Wins.

bnum = newnewYears

unum = newnewYearsU

anum = newnewYearsA

fnum = newnewYearsF

inum = newnewYearsI

snum = newnewYearsS

gnum = newnewYearsG

enum = newnewYearsE

numOfAll=[]

numOfAll = np.concatenate((bnum, unum, anum, fnum, inum, snum, gnum, enum))

allWorldCups = numOfAll.size

bavg = newnewYears.size/allWorldCups

uavg = newnewYearsU.size/allWorldCups

aavg = newnewYearsA.size/allWorldCups

favg = newnewYearsF.size/allWorldCups

iavg = newnewYearsI.size/allWorldCups

savg = newnewYearsS.size/allWorldCups

gavg = newnewYearsG.size/allWorldCups

eavg = newnewYearsE.size/allWorldCups

3- We then used the plot commands to draw the graph

width = 0.10

fig, ax = plt.subplots()

names = []

ax.bar(labels[0], brazil, color='gold',edgecolor='white', label = 'Brazil')

ax.bar(labels[0], uruguay, bottom=brazil, color='cornsilk',edgecolor='white', label = 'Uruguay')

ax.bar(labels[0], argentina, bottom=brazil+uruguay, color='lightskyblue',edgecolor='white', label = 'Argentina')

ax.bar(labels[1], italy, color='slateblue',edgecolor='white', label = 'Italy')

ax.bar(labels[1], germany, bottom=italy, color='palevioletred',edgecolor='white', label = 'Germany')

ax.bar(labels[1], france, bottom=germany+italy, color='b',edgecolor='white', label = 'France')

ax.bar(labels[1], spain, bottom=germany+italy+france, color='yellow',edgecolor='white', label = 'Spain')

ax.bar(labels[1], england, bottom=germany+italy+france+spain, color='grey',edgecolor='white', label = 'England')

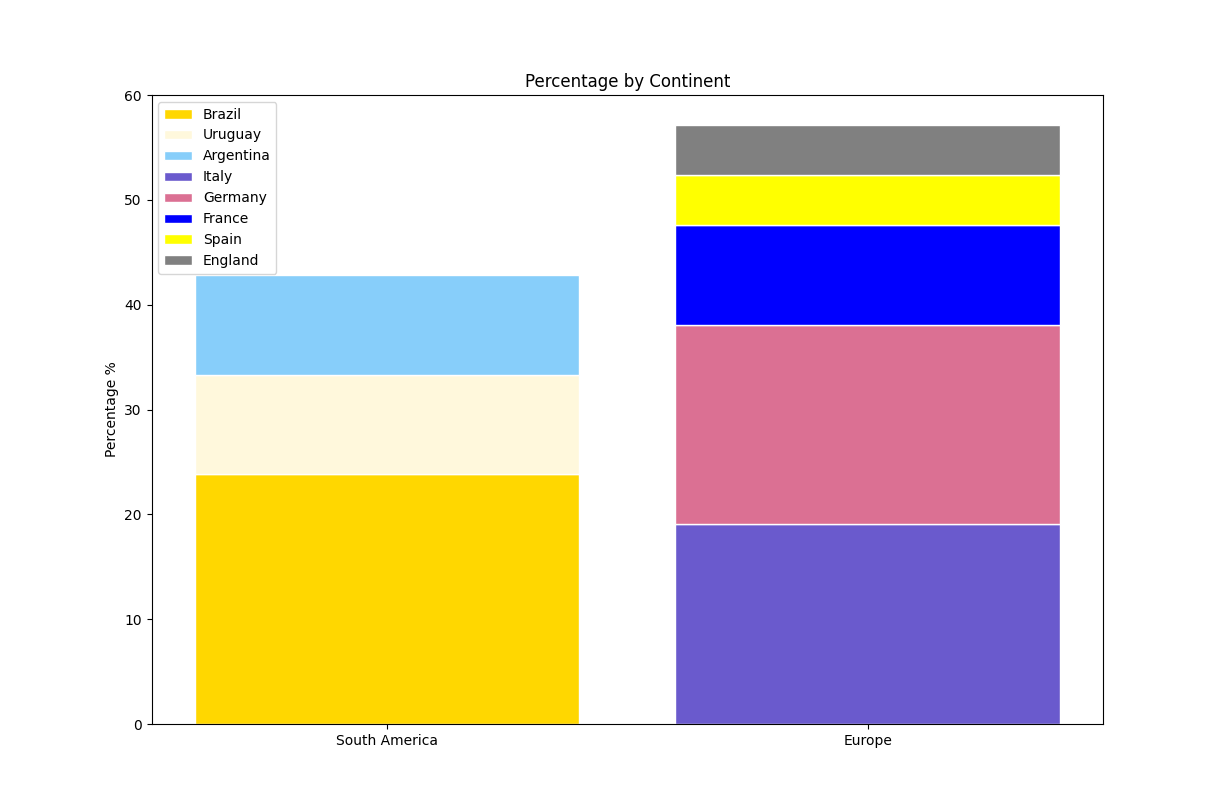
ax.set\_ylabel('Percentage %')

ax.set\_title('Percentage by Continent')

ax.legend()

plt.show()

**The Graph**



As we can see from the graph the percentage for each continent of world cup wins and it also shows the percentage for each country of world cup wins.

Third and Fourth Graph (Percentage of Games Won With a Comeback)

These 2 graphs describe the percentage of games won with a comeback(losing at half time and winning at full time)

1- We Firstly calculated the Half time scores by subtracting the half time Away goals from Home half time goals, we did the same for the full time Goals

halfTimeScore = (matches["Half\_time\_Home\_Goals"][0:852] - matches["Half\_time\_Away\_Goals"][0:852]).to\_numpy()

fullTimeScore = (matches["Home\_Team\_Goals"][0:852] - matches["Away\_Team\_Goals"][0:852]).to\_numpy()

n = np.subtract(fullTimeScore, halfTimeScore)

2- We then calculated the number of wins by comeback and the wins through the whole game, and the number of losses by home teams.

draw = 0

comeBackWin = 0

win = 0

lose = 0

hTLose = 0

for i in range(0, fullTimeScore.size):

if (halfTimeScore[i] < 0 and fullTimeScore[i] > 0):

comeBackWin += 1

hTLose += 1

elif fullTimeScore[i] == 0:

draw += 1

elif (fullTimeScore[i] > 0 and halfTimeScore[i] > 0):

win += 1

elif (fullTimeScore[i] < 0):

lose += 1

3- Then we calculated the win percentage and comeback wins from the total of games won, and the loss percentage from half time losses and comeback win percentage from half time losses

totalGamesWon = win + comeBackWin

winPer = (win / totalGamesWon) \* 100

cbWinPer = (comeBackWin / totalGamesWon) \* 100

losePer = (lose / hTLose) \* 100

cbWinPer1 = (cbWinPer/hTLose) \* 100

4- We then used the plot commands to draw the graph

values = [winPer, cbWinPer]

Ans= ['Games Won The Whole Game','Games Won With a Comeback']

plt.pie(values,labels = Ans,autopct = '%1.1f%%',startangle=90,explode =(0,.1))

plt.legend()

plt.title('Percentage of Games Won with a Comeback Relative to All the Games Won')

plt.show()

values1 = [losePer, cbWinPer1]

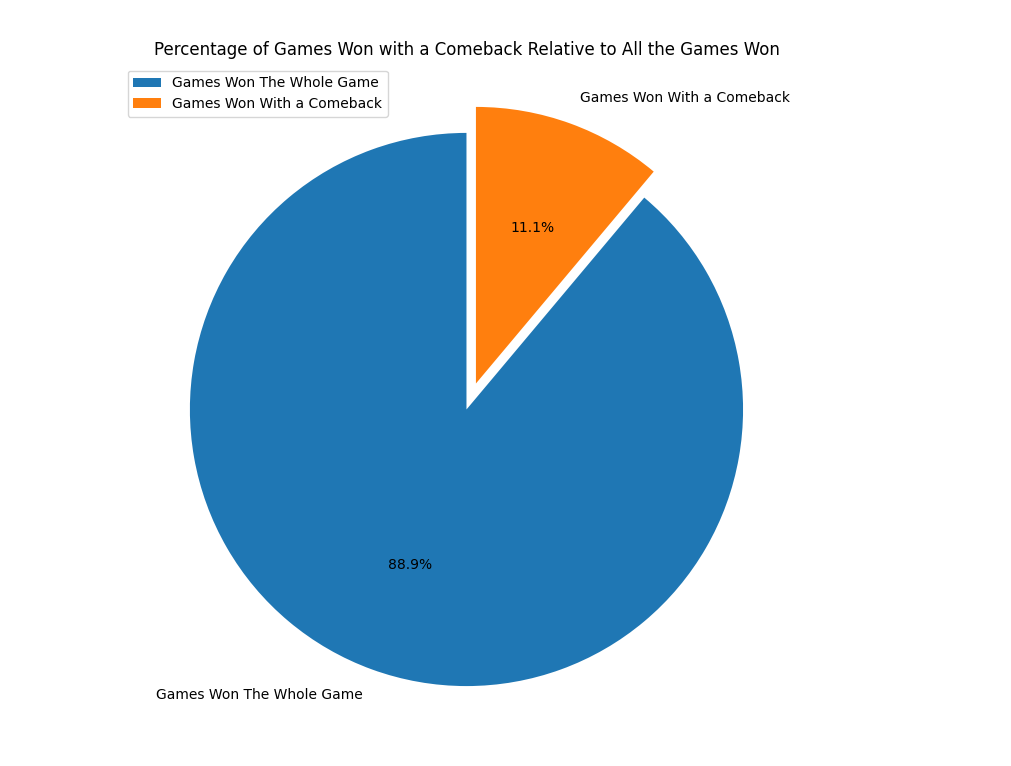
Ans1= ['Percentage of Games Lost Relative to Games Lost After Half Time','Percentage of Games Won With a Comeback Relative to Games Lost After Half Time']

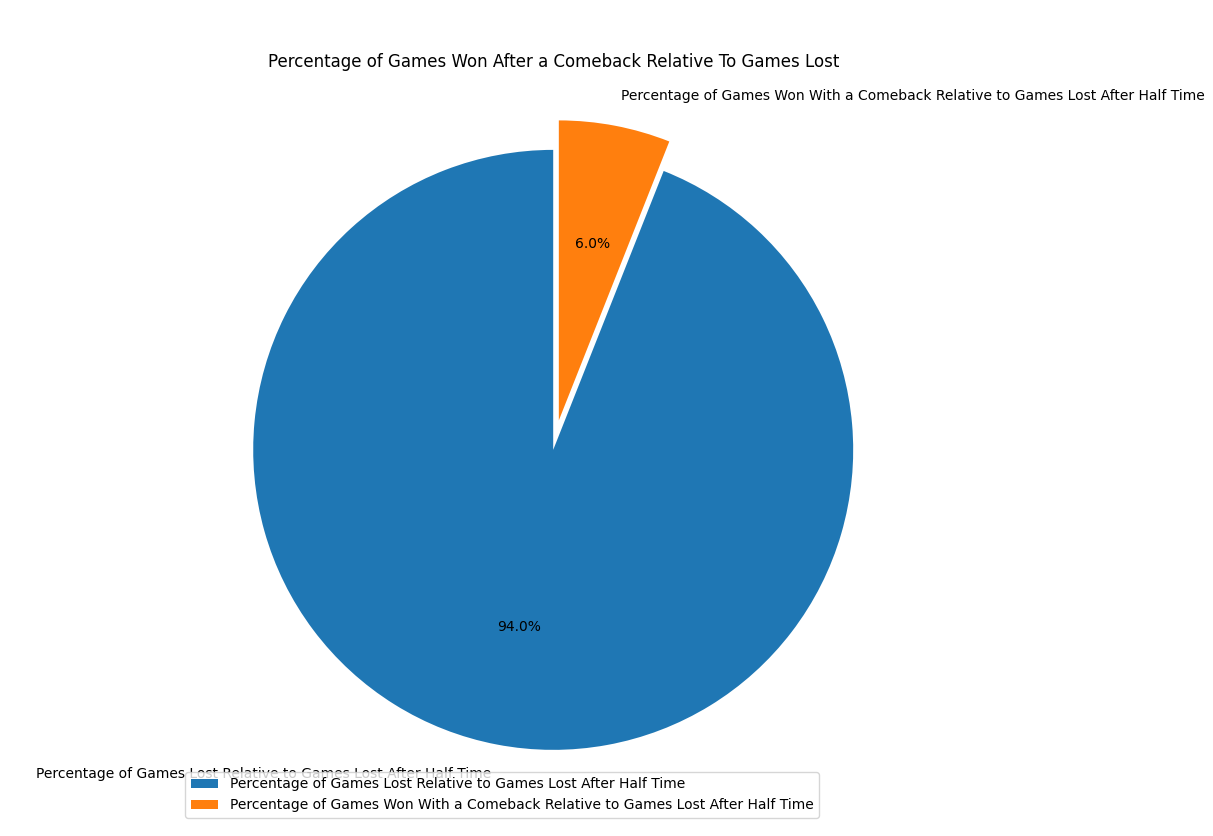
plt.pie(values1,labels = Ans1,autopct = '%1.1f%%',startangle=90,explode =(0,.1))

plt.legend()

plt.title('Percentage of Games Won After a Comeback Relative To Games Lost')

plt.show()





1. As we can see in the first pie chart the percentage of home comeback wins relative to wins from the start of the game
2. In the second pie chart it shows the percent of won games with comeback relative to the games lost after also losing in the half time

Fifth Graph (Scatter plot for clubs, stamina and strength)

This graph shows the average for clubs stamina and strength for the players with a rating higher than 86

This code describes the way we made the scatter plot graph.

1- we firstly determined the players with a higher rating than 86, then we got the average for each club’s stamina and strength

2- We then used the plot commands to draw the graph

topPlayers = players[(players.Rating >= 86)]

avgOfAll = topPlayers.groupby((['Club'])).mean()

avgOfAll = avgOfAll.reset\_index()

avg = avgOfAll.drop(columns=['Contract\_Expiry'])

avg['Club'].str.split()

avg['Club'] = avg['Club'].str.split().str[0]

plt.scatter('Club', 'Stamina', c='Strength', data=avg, cmap='viridis')

plt.xlabel('Clubs')

plt.ylabel('Stamina')

plt.colorbar(label = 'Strength Color Bar');

plt.show()

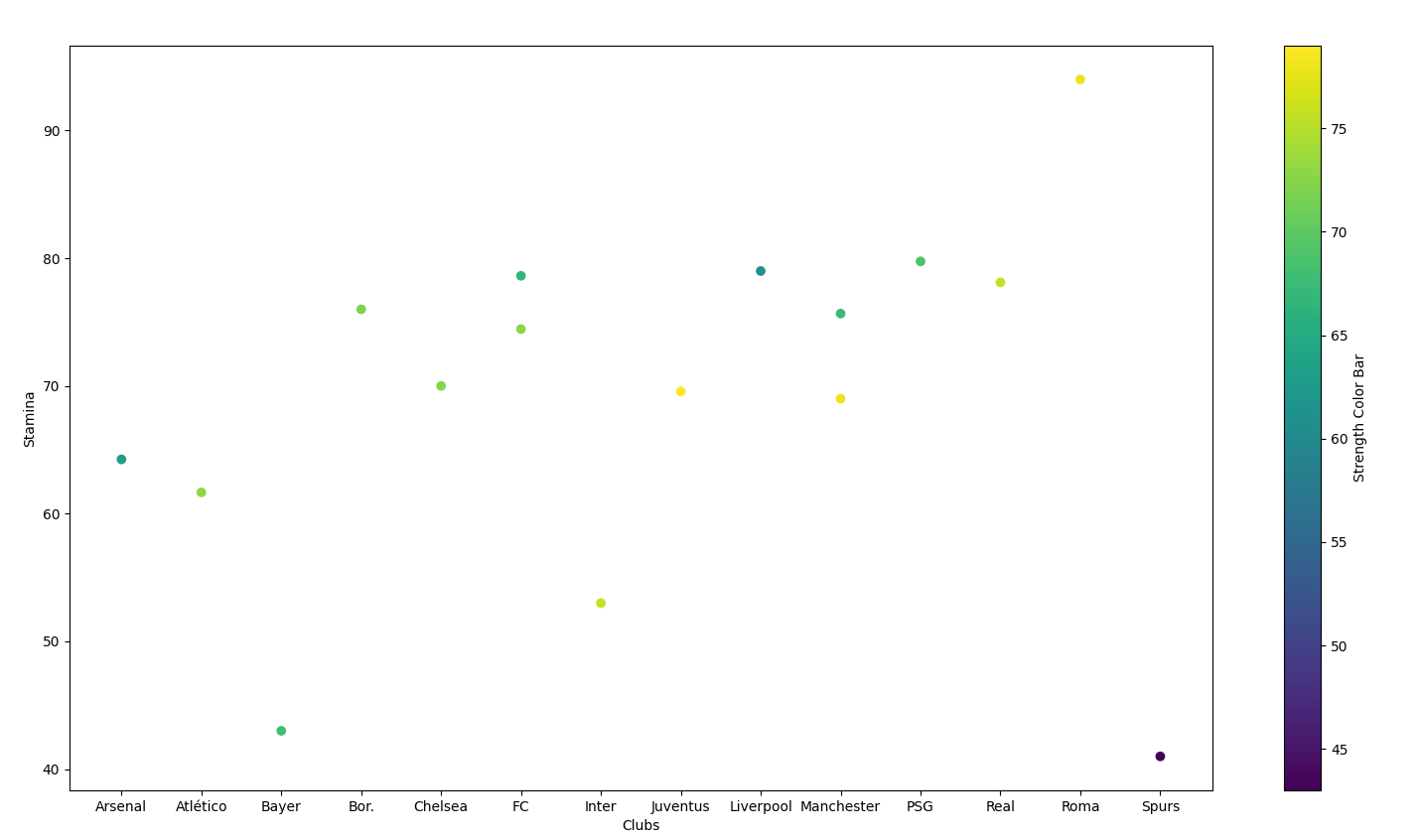
**The Graph**

The graphs shows the relationship between the club's average stamina and average strength

X axis is the clubs names

Y axis is the stamina average for each club

The color bar is the average strength for each club



Sixth Graph (Top Attendance For Each Team)

1- We sorted the dataframe by attendance

sorted\_attendance = matches.sort\_values(by=['Attendance'], ascending=False)

2- Then we got the top attendances for the matches and got the teams that played in the matches initials and concatenated the initials for the home and away team

top\_attendances = sorted\_attendance.head(10)

top\_home = top\_attendances.loc[:,"Home Team Initials"]

top\_away = top\_attendances.loc[:,"Away Team Initials"]

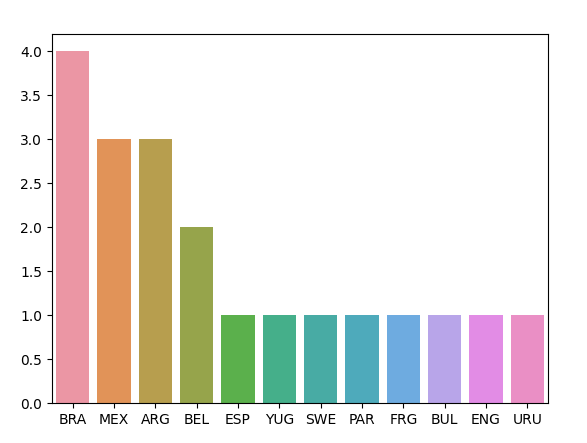
top\_all = pd.concat([top\_away,top\_home])

3- We then used the plot commands to draw the graph

x = top\_all.value\_counts()

sns.barplot(x= x.index,y = x.values)

plt.show()



A bar plot for the number of the most attended matches played by each national team. We took a list of the 10 highest matches in terms of attendance, then added up each appearance of the national teams in the aforementioned list and showed it in a bar plot.

Seventh Graph (Heights of Professional Players Frequency)

1- We determined a dataframe for the heights of the players

height = pd.DataFrame(players["Height"])

2- then we removed the (cm) from the heights cells and converted them into integers

height.dropna()

height['Height'].str.split()

height['Height'] = height['Height'].str.split().str[0]

height["Height"] = height["Height"].astype(str).astype(int)

3- then we determined the Mean and STD for the heights by using the describe function

mean = height.describe().mean()

std = height.describe().std()

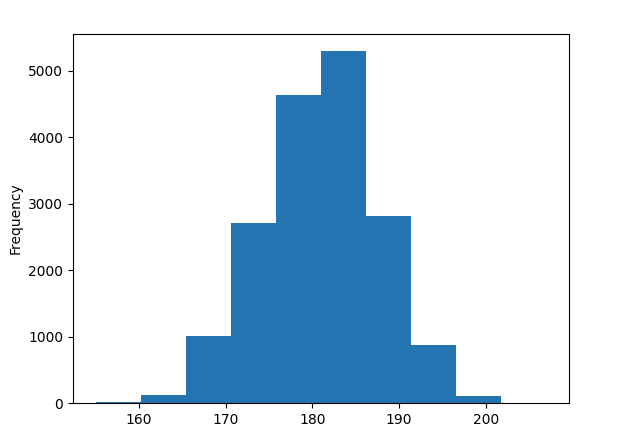
x\_values = np.arange(160, 200, 1)

y\_values = norm(mean, std)

4- We then used the plot commands to draw the graph

height['Height'].plot(kind='hist')

plt.show()



a histogram plot for the heights of professional players that by looking at, we can know what is nearly the median of their heights, they are mostly concentrated in the 180 to 190 cm region.

***THE END***