1 - import libraries

2 - read the pokemon file

3 - Display a summary of the basic information about this DataFrame and its data

4 - Create a function that get a serie as parameter and return a list of unique values.

Use this function to create a list type1\_list of the unique values of ‘Type 1’ and a list type2\_list of unique values of ‘Type 2’

5 - Add a column named 'Total' that will contain the sum of all the numeric columns (from 'HP' to 'Speed') per row using the sum method

6 -Sort the dataframe by 'Total' and find the mean of the 'Total' column

7 - Create a dataframe with all the rows that have a total greater than its mean

8 - Find the number of 'Attack' and 'Defense' per 'Type 1' and 'Type 2' (including NaN value per 'Type 2')

9 - Compute the mean value for column ‘Total’ for each value of ‘Type 1’

Display the result sorted by ‘Total’ and rounded to 2 digits

10 - Find the names of the 5 pokemons with the highest defense

11 - Find the names of the 5 pokemons with the lowest attack

12 - Who is the best pokemon (with the highest total)

13 -Create a dataframe df with 3 columns: type ,attack ,defense

The rows of the dataframe will be the sum of 'Attack' and the sum of 'Defense' per 'Type 1'

14 -Create a barplot that represents the number of attacks and the number of defense (in the same graph) per type 1 value

##Challenging ….

15 - A DataFrame has a column of groups 'grps' and column of integer values 'vals':

df = pd.DataFrame({'grps': list('aaabbcaabcccbbc'),'vals': [12,345,3,1,45,14,4,52,54,23,235,21,57,3,87]})

For each group, find the sum of the three greatest values. You should end up with the answer as follows:

grps a 409 b 156 c 345

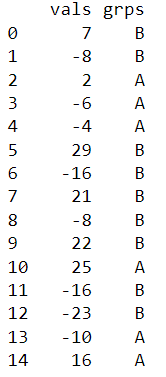
16 -You are given the DataFrame below with a column of group IDs, 'grps', and a column of corresponding integer values, 'vals'.

df = pd.DataFrame({"vals": randint(-30, 30, size=15),

"grps": choice(["A", "B"], 15)})

Create a new column 'new\_val' which contains the same values as 'vals' if ‘vals’>0 and contains the mean of the positive values of ‘vals’ of the group if ‘vals’<0 . For example:

If the dataframe is:



The result will be:

A picture containing text, font, screenshot, number

Description automatically generated

**Answers**

# 1- import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

#2- read the pokemon file

pokemon=pd.read\_csv('pokemon.csv')

pokemon.head(5)

# 3- Display a summary of the basic information about this DataFrame and its data

# 4 - Create a function that get a serie as parameter and return a list of unique values.

# Use this function to create a list type1\_list of the unique values of ‘Type 1’ and a list type2\_list of unique values of ‘Type 2’

def f\_unique(l):

l1=l.unique()

return list(l1)

type1\_list = f\_unique(pokemon['Type 1'])

type2\_list =f\_unique(pokemon['Type 2'])

print(type1\_list)

print(type2\_list)

# 5- Add a column named 'Total' that will contain the sum of all the numeric columns (from 'HP' to 'Speed') per row using the sum method

pokemon['Total']=pokemon.loc[:,'HP':'Speed'].sum(axis=1)

print(pokemon.head(5))

# 6 -Sort the dataframe by 'Total' and find the mean of the 'Total' column

pokemon.sort\_values(['Total'],ascending=False , inplace=True)

print(pokemon.head(10))

total\_mean=pokemon['Total'].mean()

print(total\_mean)print(total\_mean)

# 7- Create a dataframe with all the rows that have a total greater than its mean

pokemon1=pokemon.loc[pokemon['Total']>total\_mean]

print(pokemon1.head(10))

# 8 - Find the number of 'Attack' and 'Defense' per 'Type 1' and 'Type 2' (including NaN value per 'Type 2')

pokemon.groupby(['Type 1','Type 2'],dropna=False)[['Attack','Defense']].count()

# 9- Compute the mean value for column ‘Total’ for each value of ‘Type 1’

# Display the result sorted by ‘Total’ and rounded to 2 digits

print(pokemon.groupby('Type 1')["Total"].mean().round(2).sort\_values())

# 10- Find the names of the 5 pokemons with the highest defense

# 11 -Find the names of the 5 pokemons with the lowest attack

# 12 -Who is the best pokemon (with the highest total)

# 13 -Create a dataframe df with 3 columns: type ,attack ,defense

# The rows of the dataframe will be the sum of 'Attack' and the sum of 'Defense' per 'Type 1'

type\_df=pd.DataFrame(columns=['type','attack','defense'])

i=0

for t in type1\_list:

a=pokemon.loc[pokemon['Type 1']==t]['Attack'].sum()

d=pokemon.loc[pokemon['Type 1']==t]['Defense'].sum()

type\_df.loc[i]=[t,a,d]

i+=1

print(type\_df)

# 14 -Create a barplot that represents the number of attacks and the number of defense (in the same graph) per type 1 value

print(type\_df)

lx=list(type\_df['type'])

print(lx)

ly1=list(type\_df['attack'])

ly2=list(type\_df['defense'])

x\_axis = np.arange(len(lx))

x\_labels=lx

plt.xticks(ticks=np.arange(len(lx)),labels=x\_labels,rotation = 90) # Rotates X-Axis Ticks by 90-degrees

plt.bar(x\_axis-0.2,ly1,color ='red', width = 0.25, label='attack')

plt.bar(x\_axis+0.2,ly2,color ='blue', width = 0.25,label='defense')

plt.legend()

plt.show()

# 15 - A DataFrame has a column of groups 'grps' and column of integer values 'vals':

df = pd.DataFrame({'grps': list('aaabbcaabcccbbc'),'vals': [12,345,3,1,45,14,4,52,54,23,235,21,57,3,87]})

For each group, find the sum of the three greatest values. You should end up with the answer as follows:

grps a 409 b 156 c 345

df = pd.DataFrame({'grps': list('aaabbcaabcccbbc'),'vals': [12,345,3,1,45,14,4,52,54,23,235,21,57,3,87]})

print(df)

for name, group in df.groupby('grps'):

print(name, group['vals'].nlargest(3).sum())

print()

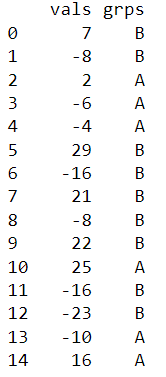
# 16 -You are given the DataFrame below with a column of group IDs, 'grps', and a column of corresponding integer values, 'vals'.

df = pd.DataFrame({"vals": randint(-30, 30, size=15),

"grps": choice(["A", "B"], 15)})

print(df)

# Create a new column 'new\_val' which contains the same values as 'vals' if ‘vals’>0 and contains the mean of the positive values of ‘vals’ of the group if ‘vals’<0 . For example:



A picture containing text, font, screenshot, number

Description automatically generated

def val\_cond(row):

g=row['grps']

if row['vals']<0:

return np.round(df\_1.loc[g]['vals'],2)

return row['vals']

df\_1=df.groupby('grps').apply(lambda x: x[x['vals']>=0].mean())

# contains the mean of the positive values per group

df['new\_val']=df.apply(val\_cond , axis=1)

print(df)