SUMMER ANALYTICS 2022

Week-1 Assignment

Gotta catch 'em all !

Welcome to your first assignment of Summer Analytics 2022! We hope you are excited to implement and test everything you have learnt up until now. The dataset which you'll use includes information about Pokemons.

We've got an interesting set of questions for you to get a basic understanding of pandas and data visualization libraries. GOOD LUCK!

Let's get started with importing numpy, pandas, seaborn and matplotlib!

Note - matplotlib should be imported with the command :

```
import matplotlib.pyplot as plt
```

1) Start by importing all important libraries

For eg, "import numpy as np"

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2) Read the csv file and assign it to a variable .

In [4]:
```
pmon=pd.read_csv('C:\\Users\\avish\\pokemon.csv',encoding='latin1')
```

Out[4]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	True
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	True
797	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	True
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	True

800 rows × 13 columns

3) Display shape of dataframe

Expected Output - (800, 13)

In [5]:
```
pmon.shape
```

Out[5]:
```
(800, 13)
```

4) Print all columns of dataframe

Return an array containing names of all the columns.

In [6]:
```
pmon.columns
```

Out[6]:
```
Index(['#', 'Name', 'Type 1', 'Type 2', 'Total', 'HP', 'Attack', 'Defense',
       'Sp. Atk', 'Sp. Def', 'Speed', 'Generation', 'Legendary'],
      dtype='object')
```

5) Remove the column '#' and update the dataframe.

In [8]:
```
pmon.drop('#',axis=1,inplace=True)
```

6) Set the 'Name' column as the index of dataframe

In [10]:
```
pmon.set_index('Name',inplace=True)
```

7) Print a list of all the unique Type-1 powers

In [16]:
```
pmon['Type 1'].unique()
```

Out[16]:
```
array(['Grass', 'Fire', 'Water', 'Bug', 'Normal', 'Poison', 'Electric',
       'Ground', 'Fairy', 'Fighting', 'Psychic', 'Rock', 'Ghost', 'Ice',
       'Dragon', 'Dark', 'Steel', 'Flying'], dtype=object)
```

8) Create a column which contains the Type 1 and Type 2 abilities of pokemons, seperated with a '+" sign. Also, display the no. of pokemons that have type-1 power as 'Psychic' and type 2 power as 'Flying' using this new column.

In [29]:
```
pmon['type1_type2']=pmon['Type 1']+'+'+pmon['Type 2']
pmon[pmon.type1_type2=='Psychic+Flying'].count()
```

Out[29]:
```
Type 1         6
Type 2         6
Total          6
HP             6
Attack         6
Defense        6
Sp. Atk        6
Sp. Def        6
Speed          6
Generation     6
Legendary      6
type1_type2    6
dtype: int64
```

dtype: int64

GRADED Questions (To be answered in the quiz)

Try to retrieve some information from the data and answer the questions below . BEST OF LUCK !!

1. How many pokemons have 'Mega' in their name?

In [37]:
```
pmon.Name[pmon.Name.str.contains('Mega')].count()
```

Out[37]: 49

2. What is the standard deviation of Sp. Def. in the dataset ?

In [74]:
```
round(pmon['Sp. Def'].std(),3)
```

Out[74]: 27.829

3. What percentage (upto 3 decimal places) of pokemons are legendary ?

In [54]:
```
pmon['Legendary']=pmon.Legendary.astype(int)
a=pmon.Legendary[pmon.Legendary==1].count()
b=pmon.Legendary.count()
percentage=(a/b)*100
percentage
```

Out[54]: 8.125

Out[54]: 8.125

4. Name the pokemon(s) with Maximum Defense.

In [61]:
```
a=pmon.Defense.max()
pmon.Name[pmon.Defense==a]
```

Out[61]:
```
224    SteelixMega Steelix
230                Shuckle
333      AggronMega Aggron
Name: Name, dtype: object
```

5. Which poison pokemon has the strongest attack ?

In [71]:
```
a=pmon[pmon['Type 1']=='Poison']
b=a.Attack.max()
a[a.Attack==b]
```

Out[71]:

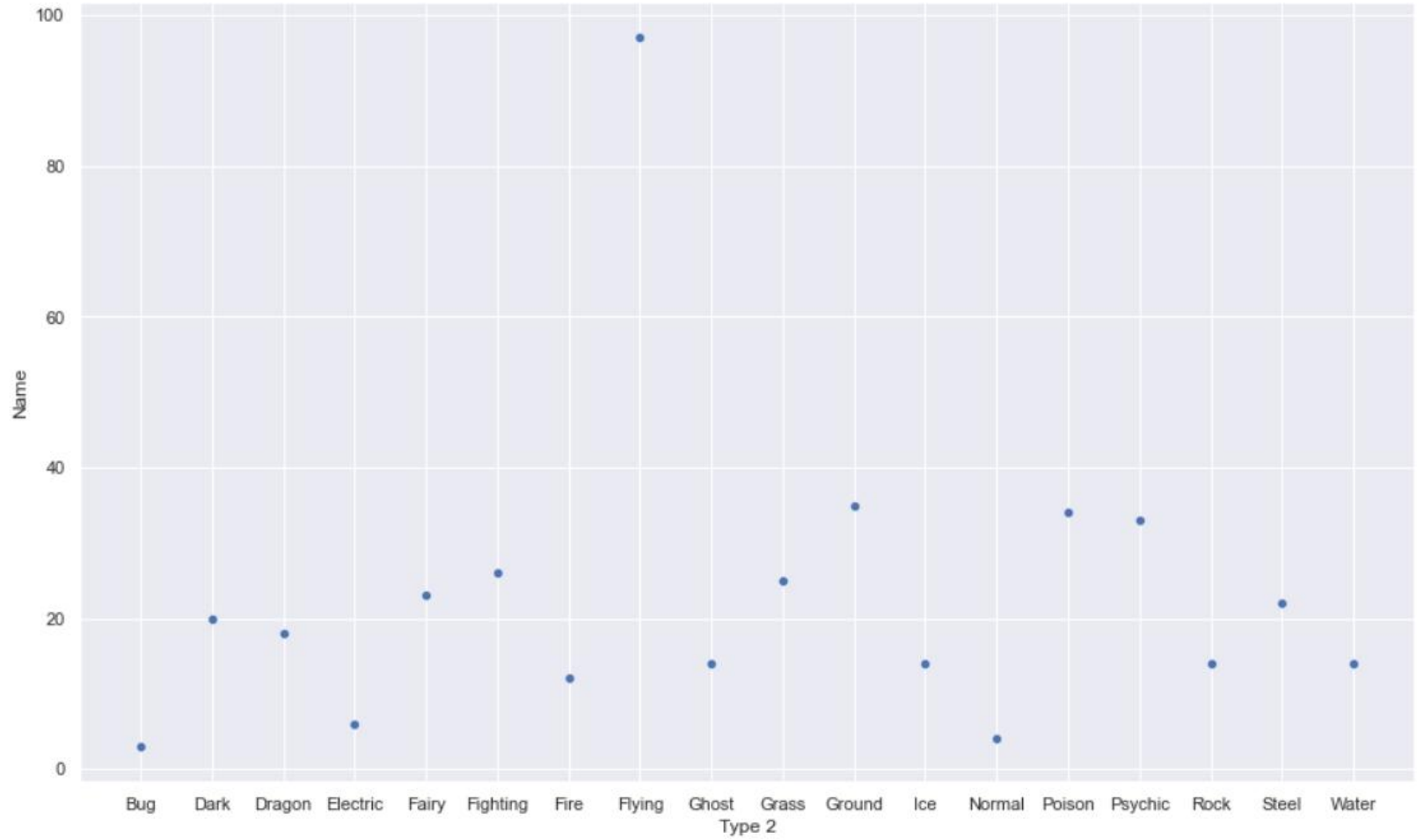
	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	type1_type2
504	Toxicroak	Poison	Fighting	490	83	106	65	86	65	85	4	0	Poison+Fighting

6. Using seaborn make different types of plots, observe the trend and answer the questions given in the form.

In [94]:
```
a=pmon.groupby('Type 2').count()
c=a['Name']
sns.set(rc = {'figure.figsize':(15,9)})
sns.scatterplot(data=c)
```

Out[94]:
```
<AxesSubplot:xlabel='Type 2', ylabel='Name'>
```

Out[94]: <AxesSubplot:xlabel='Type 2', ylabel='Name'>



In [177]: a=pmon.groupby('Type 1').count().sort_values(by='Legendary')

jupyter SA2022_W1 Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Markdown

In [177]:
```
a=pmon.groupby('Type 1').count().sort_values(by='Legendary')
a
```

Out[177]:

Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	type1_type2
Flying	2	4	4	4	4	4	4	4	4	4	2
Fairy	2	17	17	17	17	17	17	17	17	17	2
Ice	11	24	24	24	24	24	24	24	24	24	11
Fighting	7	27	27	27	27	27	27	27	27	27	7
Steel	22	27	27	27	27	27	27	27	27	27	22
Poison	13	28	28	28	28	28	28	28	28	28	13
Dark	21	31	31	31	31	31	31	31	31	31	21
Ghost	22	32	32	32	32	32	32	32	32	32	22
Dragon	21	32	32	32	32	32	32	32	32	32	21
Ground	19	32	32	32	32	32	32	32	32	32	19
Electric	17	44	44	44	44	44	44	44	44	44	17
Rock	35	44	44	44	44	44	44	44	44	44	35
Fire	24	52	52	52	52	52	52	52	52	52	24
Psychic	19	57	57	57	57	57	57	57	57	57	19
Bug	52	69	69	69	69	69	69	69	69	69	52
Grass	37	70	70	70	70	70	70	70	70	70	37
Normal	37	98	98	98	98	98	98	98	98	98	37
Water	53	112	112	112	112	112	112	112	112	112	53

jupyter SA2022_W1 Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

In [166]:
```
a
```

Out[166]:
```
Legendary  Generation
1          2              5
           1              6
           6              8
           4             13
           5             15
           3             18
0          6             74
           2            101
           4            108
           3            142
           5            150
           1            160
Name: Generation, dtype: int64
```

In [ ]:
```
#your code here
```

7. Which is the second fastest non-legendary 'Ghost' type pokemon from 4th generation ?

In [144]:
```
a=pmon[(pmon.Legendary==0)&(pmon['Type 2']=='Ghost')&(pmon.Generation==4)]
a
```

Out[144]:

Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	type1_type2
Froslass	Ice	Ghost	480	70	80	70	80	70	110	4	0	Ice+Ghost
Rotom	Electric	Ghost	440	50	50	77	95	77	91	4	0	Electric+Ghost

8. How many non-legendary pokemons have stronger defence but weaker attack than Charizard?

7. Which is the second fastest non-legendary 'Ghost' type pokemon from 4th generation ?

In [144]:
```
a=pmon[(pmon.Legendary==0)&(pmon['Type 2']=='Ghost')&(pmon.Generation==4)]
a
```

Out[144]:

Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	type1_type2
Froslass	Ice	Ghost	480	70	80	70	80	70	110	4	0	Ice+Ghost
Rotom	Electric	Ghost	440	50	50	77	95	77	91	4	0	Electric+Ghost

8. How many non-legendary pokemons have stronger defence but weaker attack than Charizard?

In [157]:
```
at=pmon.loc['Charizard',:].Attack
df=pmon.loc['Charizard',:].Defense
a=pmon[(pmon.Legendary==0)&(pmon.Defense>df)&(pmon.Attack<at)]
a.shape
```

Out[157]: (116, 12)

Congratulations on coming this far! Since we were having so much fun playing with this