

```
In [1]: import pandas as pd
import numpy as np

In [2]: df=pd.read_csv(r"C:\Users\91983\Downloads\pokemon_data.csv")

In [3]: df.head()
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False

```
In [5]: df.iloc[0:4]
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False

```
In [76]: Name = df["Type 1"] ## to look at the specefic columns name, only for string
```

```
In [78]: Name
```

```
Out[78]: 0      Grass
1      Grass
2      Grass
3      Grass
4      Fire
...
795    Rock
796    Rock
797    Psychic
798    Psychic
799    Fire
Name: Type 1, Length: 800, dtype: object
```

```
In [18]: df.loc[df["Name"] == "Charmeleon"] ### to look ate the specefic value ina specific columns name
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
5	5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False

```
In [31]: ### to combined some columns and create a new columns

df["Total"] = df["Defense"]+df["HP"]+df["Attack"]+df["Defense"]+df["Sp. Atk"]+df["Sp. Atk"]+df["Speed"]

In [32]: df.head()
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False	367
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	468
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	608
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	750
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	362

```
In [34]: df.head() ## to see top rows of data
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False

```
In [46]: d= "hcum yrev mih evol i dna nos ym si hsnardur "[::-1] ##### to reverse the string
```

```
In [47]: d
```

```
Out[47]: ' rudransh is my son and i love him very much'
```

```
In [52]: df.dropna() ## drop row when aleats one element is missing
```

	#	Name	Type 1	Type 2	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	123	122	120	80	1	False
6	6	Charizard	Fire	Flying	78	109	85	100	1	False
...
795	719	Diancie	Rock	Fairy	150	100	150	50	6	True
796	719	DiancieMega Diancie	Rock	Fairy	110	160	110	110	6	True
797	720	HoopaHoopa Confined	Psychic	Ghost	60	150	130	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	60	170	130	80	6	True
799	721	Volcanion	Fire	Water	120	130	90	70	6	True

414 rows × 10 columns

```
In [63]: df.dropna(axis= 1) ## drop columns when aleats one element is missing
```

	#	Name	Type 1	Type 2	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	83	100	100	80	1	False
3	3	VenusaurMega Venusaur	Grass	Poison	123	122	120	80	1	False
6	6	Charizard	Fire	Flying	78	109	85	100	1	False
...
795	719	Diancie	Rock	Fairy	150	100	150	50	6	True
796	719	DiancieMega Diancie	Rock	Fairy	110	160	110	110	6	True
797	720	HoopaHoopa Confined	Psychic	Ghost	60	150	130	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	60	170	130	80	6	True
799	721	Volcanion	Fire	Water	120	130	90	70	6	True

414 rows × 10 columns

```
In [53]: df.dropna(how="all") ## drop row where all elements are missiong
```

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

800 rows × 10 columns

```
In [65]: df.take([2, 3],axis= 1) ##( axis = 0 for row, axis = 1 for columns ) [2,3] means columns or row indexing ]
```

	Type 1	Type 2
0	Grass	Poison
1	Grass	Poison
2	Grass	Poison
3	Grass	Poison
4	Fire	NaN
...
795	Rock	Fairy
796	Rock	Fairy
797	Psychic	Ghost
798	Psychic	Dark
799	Fire	Water

800 rows × 2 columns

```
In [55]: df.take([-1, -2])
```

	#	Name	Type 1	Type 2	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
799	721	Volcanion	Fire	Water	120	130	90	70	6	True
798	720	HoopaHoopa Unbound	Psychic	Dark	60	170	130	80	6	True

```
In [83]: e=df[["Name","Type 1"]] ### to combine two string columns and make a new columns
```

```
In [85]: e.head()
```

	Name	Type 1
0	Bulbasaur	Grass
1	Ivysaur	Grass
2	Venusaur	Grass
3	VenusaurMega Venusaur	Grass
4	Charmander	Fire

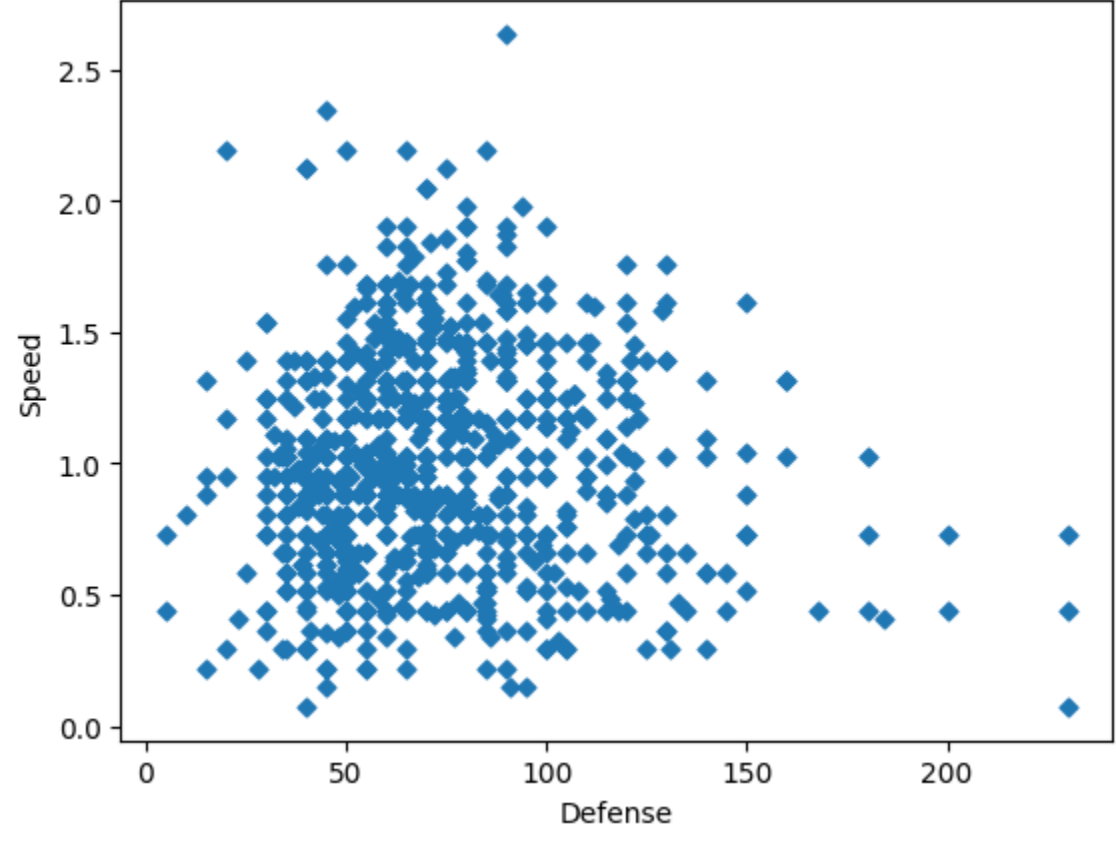
```
In [86]: import matplotlib.pyplot as plt
```

```
In [105]: xpoints=df.("Defense")

Cell In [105], line 1
xpoints=df.("Defense")
^
SyntaxError: invalid syntax
```

```
In [112]: df.plot(kind = 'scatter', x = 'Defense', y = 'Speed',marker="D")
```

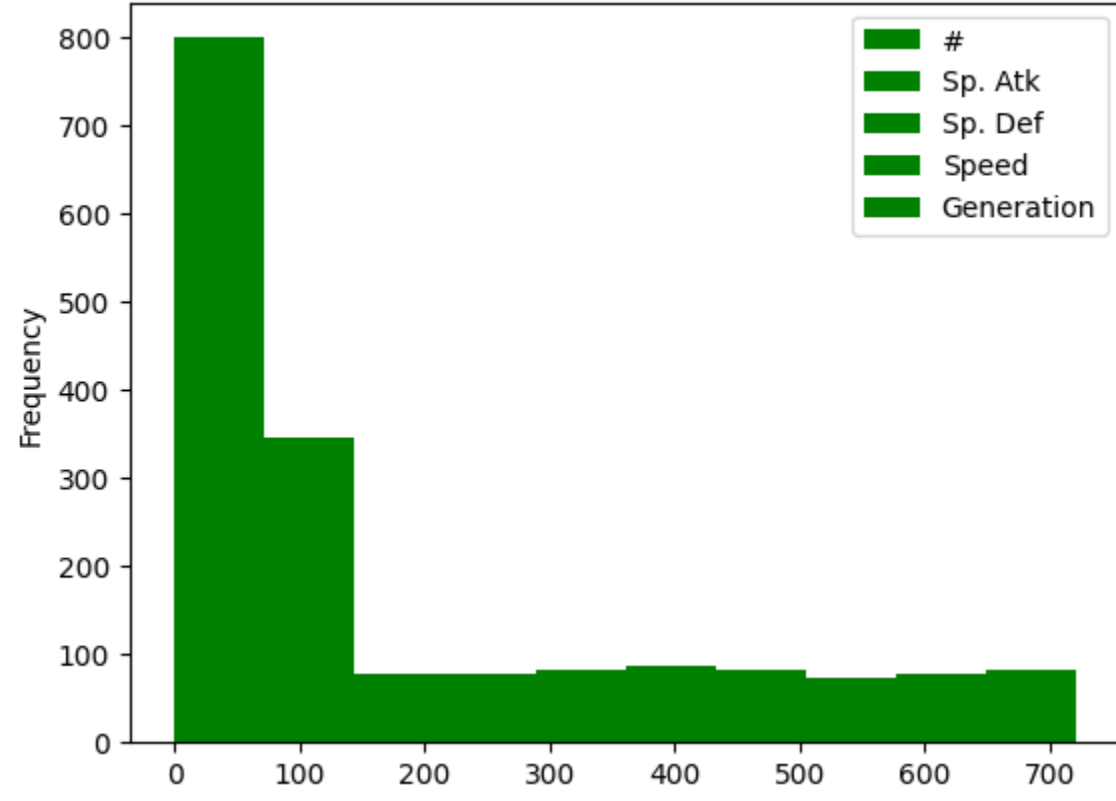
Out[112]: <AxesSubplot: xlabel='Defense', ylabel='Speed'>



```
In [122]: #####to compare the data of two columns ,here alpha value may be 1 for less tranperant and 0.5 for more tranperant

df.plot(kind="hist",x="Defense",color="g",)
```

Out[122]: <AxesSubplot: ylabel='Frequency'>



```
In [103]: df["Speed"]=df["Speed"]/df["Speed"].mean()
```

```
In [104]: df["Speed"]
```

```
Out[104]: 0      0.659075
1      0.878767
2      1.171689
3      1.171689
4      0.951997
...
795    0.732306
796    1.611072
797    1.025228
798    1.171689
799    1.025228
Name: Speed, Length: 800, dtype: float64
```

```
In [123]: from sklearn import linearRegression

-----
ImportError                                Traceback (most recent call last)
Cell In [123], line 1
----> 1 from sklearn import linearRegression

ImportError: cannot import name 'linearRegression' from 'sklearn' (C:\Users\91983\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\__init__.py)
```

```
In [125]: import scikitlearn as sk

-----
ModuleNotFoundError                        Traceback (most recent call last)
Cell In [125], line 1
----> 1 import scikitlearn as sk

ModuleNotFoundError: No module named 'scikitlearn'
```

```
In [126]: from sklearn.linear_model import LinearRegression
```

```
In [128]: x=df[["Defense"]]
```

```
In [130]: y=df[["Speed"]]
```

```
In [134]: lm=LinearRegression()
```

```
In [136]: lm=Fit(x,y)

-----
NameError                                Traceback (most recent call last)
Cell In [136], line 1
----> 1 lm=Fit(x,y)

NameError: name 'Fit' is not defined
```

```
In [ ]:
```