

<https://www.youtube.com/watch?v=vmEHCJofslg>  
[\(https://www.youtube.com/watch?v=vmEHCJofslg\)](https://www.youtube.com/watch?v=vmEHCJofslg)

In [1]: `import pandas as pd`

In [2]: `##upload csv file`  
`df= pd.read_csv('pokemon_data.csv')`

```
In [3]: print(df)
```

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

[800 rows x 12 columns]

```
In [4]: df.head(10)
```

```
Out[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
4 4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False
5 5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False
6 6	Charizard	Fire	Flying	78	84	78	109	85	100	1	False
7 6	CharizardMega Charizard X	Fire	Dragon	78	130	111	130	85	100	1	False
8 6	CharizardMega Charizard Y	Fire	Flying	78	104	78	159	115	100	1	False
9 7	Squirtle	Water	NaN	44	48	65	50	64	43	1	False

```
In [5]: # read data in pandas
```

```
In [63]: df.columns
```

```
Out[63]: Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk',  
               'Sp. Def', 'Speed', 'Generation', 'Legendary', 'Total'],  
               dtype='object')
```

```
In [66]: df.index
```

```
Out[66]: RangeIndex(start=0, stop=800, step=1)
```

```
In [7]: df['Name'] #Print a column
```

```
Out[7]: 0          Bulbasaur  
1          Ivysaur  
2          Venusaur  
3  VenusaurMega Venusaur  
4          Charmander  
...  
795          Diancie  
796  DiancieMega Diancie  
797  HoopaHoopa Confined  
798  HoopaHoopa Unbound  
799          Volcanion  
Name: Name, Length: 800, dtype: object
```

```
In [ ]:
```

```
In [12]: df['Name'][0:-80] #slicing
```

```
Out[12]: 0          Bulbasaur  
1          Ivysaur  
2          Venusaur  
3  VenusaurMega Venusaur  
4          Charmander  
...  
715  MeloettaAria Forme  
716  MeloettaPirouette Forme  
717          Genesect  
718          Chespin  
719          Quilladin  
Name: Name, Length: 720, dtype: object
```

```
In [16]: df[['Name', 'Attack', 'Defense']] #print multiple columns
```

Out[16]:

	Name	Attack	Defense
0	Bulbasaur	49	49
1	Ivysaur	62	63
2	Venusaur	82	83
3	VenusaurMega Venusaur	100	123
4	Charmander	52	43
...	...	...	...
795	Diancie	100	150
796	DiancieMega Diancie	160	110
797	HoopaHoopa Confined	110	60
798	HoopaHoopa Unbound	160	60
799	Volcanion	110	120

800 rows × 3 columns

```
In [17]: #Print a particular row or mulitple rows and column
```

```
In [62]: df.iloc[1:4,0:5]
```

Out[62]:

#	Name	Type 1	Type 2	HP
1	Ivysaur	Grass	Poison	60
2	Venusaur	Grass	Poison	80
3	VenusaurMega Venusaur	Grass	Poison	80

```
In [21]: # read a specific position
```

```
In [22]: df.iloc[4,1]
```

```
Out[22]: 'Charmander'
```

```
In [23]: #iterate through row and print a specific column
```

```
In [25]: for index, row in df.iterrows():
    print(index, row['Name'])
```

```
0 Bulbasaur
1 Ivysaur
2 Venusaur
3 VenusaurMega Venusaur
4 Charmander
5 Charmeleon
6 Charizard
7 CharizardMega Charizard X
8 CharizardMega Charizard Y
9 Squirtle
10 Wartortle
11 Blastoise
12 BlastoiseMega Blastoise
13 Caterpie
14 Metapod
15 Butterfree
16 Weedle
17 Kakuna
18 Beedrill
19 BeedrillMega Beedrill
```

In [27]: `df.loc[df['Type 1']=="Fire"]` *#Looking for special values in a column*

Out[27]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False
5	5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False
6	6	Charizard	Fire	Flying	78	84	78	109	85	100	1	False
7	6	CharizardMega Charizard X	Fire	Dragon	78	130	111	130	85	100	1	False
8	6	CharizardMega Charizard Y	Fire	Flying	78	104	78	159	115	100	1	False
42	37	Vulpix	Fire	NaN	38	41	40	50	65	65	1	False
43	38	Ninetales	Fire	NaN	73	76	75	81	100	100	1	False
63	58	Growlithe	Fire	NaN	55	70	45	70	50	60	1	False
64	59	Arcanine	Fire	NaN	90	110	80	100	80	95	1	False
83	77	Ponyta	Fire	NaN	50	85	55	65	65	90	1	False
84	78	Rapidash	Fire	NaN	65	100	70	80	80	105	1	False
135	126	Magmar	Fire	NaN	65	95	57	100	85	93	1	False
147	136	Flareon	Fire	NaN	65	130	60	95	110	65	1	False
158	146	Moltres	Fire	Flying	90	100	90	125	85	90	1	True
169	155	Cyndaquil	Fire	NaN	39	52	43	60	50	65	2	False
170	156	Quilava	Fire	NaN	58	64	58	80	65	80	2	False
171	157	Typhlosion	Fire	NaN	78	84	78	109	85	100	2	False
236	218	Slugma	Fire	NaN	40	40	40	70	40	20	2	False
237	219	Magcargo	Fire	Rock	50	50	120	80	80	30	2	False
259	240	Magby	Fire	NaN	45	75	37	70	55	83	2	False
263	244	Entei	Fire	NaN	115	115	85	90	75	100	2	True
270	250	Ho-oh	Fire	Flying	106	130	90	110	154	90	2	True
276	255	Torchic	Fire	NaN	45	60	40	70	50	45	3	False
277	256	Combusken	Fire	Fighting	60	85	60	85	60	55	3	False
278	257	Blaziken	Fire	Fighting	80	120	70	110	70	80	3	False
279	257	BlazikenMega Blaziken	Fire	Fighting	80	160	80	130	80	100	3	False

#		Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	
352	322	Numel	Fire	Ground	60	60	40	65	45	35	3	False	
353	323	Camerupt	Fire	Ground	70	100	70	105	75	40	3	False	
354	323	Camerupt	Mega Camerupt	Fire	Ground	70	120	100	145	105	20	3	False
355	324	Torkoal	Fire	NaN	70	85	140	85	70	20	3	False	
435	390	Chimchar	Fire	NaN	44	58	44	58	44	61	4	False	
436	391	Monferno	Fire	Fighting	64	78	52	78	52	81	4	False	
437	392	Infernape	Fire	Fighting	76	104	71	104	71	108	4	False	
518	467	Magmortar	Fire	NaN	75	95	67	125	95	83	4	False	
542	485	Heatran	Fire	Steel	91	90	106	130	106	77	4	True	
557	498	Tepig	Fire	NaN	65	63	45	45	45	45	5	False	
558	499	Pignite	Fire	Fighting	90	93	55	70	55	55	5	False	
559	500	Emboar	Fire	Fighting	110	123	65	100	65	65	5	False	
572	513	Panshear	Fire	NaN	50	53	48	53	48	64	5	False	
573	514	Simisear	Fire	NaN	75	98	63	98	63	101	5	False	
614	554	Darumaka	Fire	NaN	70	90	45	15	45	50	5	False	
615	555	Darmanitan	Standard Mode	Fire	NaN	105	140	55	30	55	95	5	False
616	555	Darmanitan	Zen Mode	Fire	Psychic	105	30	105	140	105	55	5	False
692	631	Heatmor	Fire	NaN	85	97	66	105	66	65	5	False	
721	653	Fennekin	Fire	NaN	40	45	40	62	60	60	6	False	
722	654	Braixen	Fire	NaN	59	59	58	90	70	73	6	False	
723	655	Delphox	Fire	Psychic	75	69	72	114	100	104	6	False	
730	662	Fletchinder	Fire	Flying	62	73	55	56	52	84	6	False	
731	663	Talonflame	Fire	Flying	78	81	71	74	69	126	6	False	
735	667	Litleo	Fire	Normal	62	50	58	73	54	72	6	False	
736	668	Pyroar	Fire	Normal	86	68	72	109	66	106	6	False	
799	721	Volcanion	Fire	Water	80	110	120	130	90	70	6	True	

In [28]: #find all the statistics

```
df.describe()
```

Out[28]:

	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
<b>count</b>	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000
<b>mean</b>	362.813750	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500	3.32375
<b>std</b>	208.343798	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474	1.66129
<b>min</b>	1.000000	1.000000	5.000000	5.000000	10.000000	20.000000	5.000000	1.00000
<b>25%</b>	184.750000	50.000000	55.000000	50.000000	49.750000	50.000000	45.000000	2.00000
<b>50%</b>	364.500000	65.000000	75.000000	70.000000	65.000000	70.000000	65.000000	3.00000
<b>75%</b>	539.250000	80.000000	100.000000	90.000000	95.000000	90.000000	90.000000	5.00000
<b>max</b>	721.000000	255.000000	190.000000	230.000000	194.000000	230.000000	180.000000	6.00000

```
In [29]: #sort values by name  
df.sort_values('Name')
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
510	460	Abomasnow	Grass	Ice	90	92	75	92	85	60	4	False
511	460	AbomasnowMega Abomasnow	Grass	Ice	90	132	105	132	105	30	4	False
68	63	Abra	Psychic	NaN	25	20	15	105	55	90	1	False
392	359	Absol	Dark	NaN	65	130	60	75	60	75	3	False
393	359	AbsolMega Absol	Dark	NaN	65	150	60	115	60	115	3	False
...	...	...	...	...	...	...	...	...	...	...	...	...
632	571	Zoroark	Dark	NaN	60	105	60	120	60	105	5	False
631	570	Zorua	Dark	NaN	40	65	40	80	40	65	5	False
46	41	Zubat	Poison	Flying	40	45	35	30	40	55	1	False
695	634	Zweilous	Dark	Dragon	72	85	70	65	70	58	5	False
794	718	Zygarde50% Forme	Dragon	Ground	108	100	121	81	95	95	6	True

800 rows × 12 columns

```
In [30]: #sort values by name by descending  
df.sort_values('Name', ascending=False)
```

```
Out[30]:
```

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
794	Zygarde50% Forme	Dragon	Ground	108	100	121	81	95	95	6	True
695	Zweilous	Dark	Dragon	72	85	70	65	70	58	5	False
46	Zubat	Poison	Flying	40	45	35	30	40	55	1	False
631	Zorua	Dark	NaN	40	65	40	80	40	65	5	False
632	Zoroark	Dark	NaN	60	105	60	120	60	105	5	False
...	...	...	...	...	...	...	...	...	...	...	...
393	AbsolMega Absol	Dark	NaN	65	150	60	115	60	115	3	False
392	Absol	Dark	NaN	65	130	60	75	60	75	3	False
68	AbomasnowMega Abomasnow	Grass	Ice	90	132	105	132	105	30	4	False
510	Abomasnow	Grass	Ice	90	92	75	92	85	60	4	False

800 rows × 12 columns

```
In [31]: #sort multiple rows
```

```
In [32]: df.sort_values(['Type 1', 'HP'], ascending=[1,0])
```

Out[32]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	
520	469	Yanmega	Bug	Flying	86	76	86	116	56	95	4	False
698	637	Volcarona	Bug	Fire	85	60	65	135	105	100	5	False
231	214	Heracross	Bug	Fighting	80	125	75	40	95	85	2	False
232	214	Mega Heracross	Bug	Fighting	80	185	115	40	105	75	2	False
678	617	Accelgor	Bug	NaN	80	70	40	100	60	145	5	False
...	...	...	...	...	...	...	...	...	...	...	...	...
106	98	Krabby	Water	NaN	30	105	90	25	25	50	1	False
125	116	Horsea	Water	NaN	30	40	70	70	25	60	1	False
129	120	Staryu	Water	NaN	30	45	55	70	55	85	1	False
139	129	Magikarp	Water	NaN	20	10	55	15	20	80	1	False
381	349	Feebas	Water	NaN	20	15	20	10	55	80	3	False

800 rows × 12 columns

## Making changes to the data

```
In [33]: df.head(6)
```

Out[33]:

#	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	Mega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False
5	5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False

```
In [34]: df['Total']= df['HP']+df['Attack']+df['Defense']+df['Sp. Atk']+df['Sp. Def']+df['Speed']
```

```
In [72]: #drop a column  
df.drop(columns=['Generation'])
```

Out[72]:

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

800 rows × 12 columns

```
In [74]: #drop columns  
df.drop(columns='Total')
```

Out[74]:

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

800 rows × 12 columns

```
In [69]: df.head(5)
```

Out[69]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
0	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False	318
1	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309

```
In [70]: # drop row  
df.drop(index=4)
```

```
Out[70]:
```

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total	
0	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False	318	
1	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405	
2	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525	
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625	
5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	False	405	
...	...	...	...	...	...	...	...	...	...	...	...	...	
795	Diancie	Rock	Fairy	50	100	150	100	150	50	6	True	600	
796	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700	
797	Hoopo	Hoopa Confined	Psychic	Ghost	80	110	60	150	130	70	6	True	600
798	Hoopo	Hoopa Unbound	Psychic	Dark	80	160	60	170	130	80	6	True	680
799	Volcanion	Fire	Water	80	110	120	130	90	70	6	True	600	

799 rows × 13 columns

```
In [43]:
```

```
In [44]: df = df.drop(columns='Total')
```

```
In [71]: df.head(4)
```

```
Out[71]:
```

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
0	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	False	318
1	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625

```
In [46]: #use .sum function for total
```

```
In [49]: df['Total']=df.iloc[:,4:10].sum(axis=1)
```

```
In [50]: df.head(5)
```

Out[50]:

#	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	318
1 2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2 3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3 3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4 4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309

```
In [52]: df.columns
```

Out[52]: Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk',  
'Sp. Def', 'Speed', 'Generation', 'Legendary', 'Total'],  
dtype='object')

```
In [53]: # rearranging columns ????????????
```

```
In [54]: #saving_CSV  
df.to_csv('modified_poke.csv',index=False)
```

## Filtering data

```
In [77]: df.head()
```

```
Out[77]:
```

#	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	318
1 2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2 3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3 3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4 4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309

```
In [88]: new_df=df.loc[(df['Type 1']=='Grass') & (df['Type 2']=='Poison') & (df['HP']>70)]  
#() are very important
```

```
In [89]: new_df
```

```
Out[89]:
```

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
2 3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3 3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
50 45	Vileplume	Grass	Poison	75	80	85	110	90	50	1	False	490
77 71	Victreebel	Grass	Poison	80	105	65	100	70	70	1	False	490
652 591	Amoonguss	Grass	Poison	114	85	70	85	80	30	5	False	464

```
In [90]: #reset index
```

```
In [93]: new_df.reset_index(drop=True,inplace=True)
```

In [94]: new\_df

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
0	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
1	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
2	45	Vileplume	Grass	Poison	75	80	85	110	90	50	1	False	490
3	71	Victreebel	Grass	Poison	80	105	65	100	70	70	1	False	490
4	591	Amoonguss	Grass	Poison	114	85	70	85	80	30	5	False	464

In [95]: df

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	50	100	150	100	150	50	6	True	600
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700
797	720	Hoopaa Hoopa Confined	Psychic	Ghost	80	110	60	150	130	70	6	True	600
798	720	Hoopaa Hoopa Unbound	Psychic	Dark	80	160	60	170	130	80	6	True	680
799	721	Volcanion	Fire	Water	80	110	120	130	90	70	6	True	600

800 rows × 13 columns

In [96]: *#filtering out names containing a particular string (Mega)*  
df.loc[df['Name'].str.contains('Mega')]

Out[96]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
7	CharizardMega Charizard X	Fire	Dragon	78	130	111	130	85	100	1	False	634
8	CharizardMega Charizard Y	Fire	Flying	78	104	78	159	115	100	1	False	634
12	BlastoiseMega Blastoise	Water	NaN	79	103	120	135	115	78	1	False	630
19	BeedrillMega Beedrill	Bug	Poison	65	150	40	15	80	145	1	False	495
23	PidgeotMega Pidgeot	Normal	Flying	83	80	80	135	80	121	1	False	579
71	AlakazamMega Alakazam	Psychic	NaN	55	50	65	175	95	150	1	False	590
87	SlowbroMega Slowbro	Water	Psychic	95	75	180	130	80	30	1	False	590
102	GengarMega Gengar	Ghost	Poison	60	65	80	170	95	130	1	False	600
124	KangaskhanMega Kangaskhan	Normal	NaN	105	125	100	60	100	100	1	False	590
137	PinsirMega Pinsir	Bug	Flying	65	155	120	65	90	105	1	False	600
141	GyaradosMega Gyarados	Water	Dark	95	155	109	70	130	81	1	False	640
154	AerodactylMega Aerodactyl	Rock	Flying	80	135	85	70	95	150	1	False	615
163	MewtwoMega Mewtwo X	Psychic	Fighting	106	190	100	154	100	130	1	True	780
164	MewtwoMega Mewtwo Y	Psychic	NaN	106	150	70	194	120	140	1	True	780
168	Meganium	Grass	NaN	80	82	100	83	100	80	2	False	525
196	AmpharosMega Ampharos	Electric	Dragon	90	95	105	165	110	45	2	False	610
224	SteelixMega Steelix	Steel	Ground	75	125	230	55	95	30	2	False	610
229	ScizorMega Scizor	Bug	Steel	70	150	140	65	100	75	2	False	600
232	HeracrossMega Heracross	Bug	Fighting	80	185	115	40	105	75	2	False	600
248	HoundoomMega Houndoom	Dark	Fire	75	90	90	140	90	115	2	False	600
268	TyranitarMega Tyranitar	Rock	Dark	100	164	150	95	120	71	2	False	700
275	SceptileMega Sceptile	Grass	Dragon	70	110	75	145	85	145	3	False	630
279	BlazikenMega Blaziken	Fire	Fighting	80	160	80	130	80	100	3	False	630
283	SwampertMega Swampert	Water	Ground	100	150	110	95	110	70	3	False	635
306	GardevoirMega Gardevoir	Psychic	Fairy	68	85	65	165	135	100	3	False	618

#		Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total
327	302	SableyeMega Sableye	Dark	Ghost	50	85	125	85	115	20	3	False	480
329	303	MawileMega Mawile	Steel	Fairy	50	105	125	55	95	50	3	False	480
333	306	AggronMega Aggron	Steel	NaN	70	140	230	60	80	50	3	False	630
336	308	MedichamMega Medicham	Fighting	Psychic	60	100	85	80	85	100	3	False	510
339	310	ManectricMega Manectric	Electric	NaN	70	75	80	135	80	135	3	False	575
349	319	SharpedoMega Sharpedo	Water	Dark	70	140	70	110	65	105	3	False	560
354	323	CameruptMega Camerupt	Fire	Ground	70	120	100	145	105	20	3	False	560
366	334	AltariaMega Altaria	Dragon	Fairy	75	110	110	110	105	80	3	False	590
387	354	BanetteMega Banette	Ghost	NaN	64	165	75	93	83	75	3	False	555
393	359	AbsolMega Absol	Dark	NaN	65	150	60	115	60	115	3	False	565
397	362	GligarMega Gligar	Ice	NaN	80	120	80	120	80	100	3	False	580
409	373	SalamenceMega Salamence	Dragon	Flying	95	145	130	120	90	120	3	False	700
413	376	MetagrossMega Metagross	Steel	Psychic	80	145	150	105	110	110	3	False	700
418	380	LatiasMega Latias	Dragon	Psychic	80	100	120	140	150	110	3	True	700
420	381	LatiosMega Latios	Dragon	Psychic	80	130	100	160	120	110	3	True	700
426	384	RayquazaMega Rayquaza	Dragon	Flying	105	180	100	180	100	115	3	True	780
476	428	LopunnyMega Lopunny	Normal	Fighting	65	136	94	54	96	135	4	False	580
494	445	GarchompMega Garchomp	Dragon	Ground	108	170	115	120	95	92	4	False	700
498	448	LucarioMega Lucario	Fighting	Steel	70	145	88	140	70	112	4	False	625
511	460	AbomasnowMega Abomasnow	Grass	Ice	90	132	105	132	105	30	4	False	594
527	475	GalladeMega Gallade	Psychic	Fighting	68	165	95	65	115	110	4	False	618
591	531	AudinoMega Audino	Normal	Fairy	103	60	126	80	126	50	5	False	545
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700

```
In [97]: #To do the opposite you can use the ~ before your methods
```

```
In [98]: df.loc[~df['Name'].str.contains('Mega')]
```

```
In [99]: #regex filtering are super powerful filtering based on patterns
```

```
In [101]: import re  
df.loc[df['Type 1'].str.contains('Fire|Grass', regex=True)]
```

Out[101]:

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	False	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
735	667	Litleo	Fire	Normal	62	50	58	73	54	72	6	False	369
736	668	Pyroar	Fire	Normal	86	68	72	109	66	106	6	False	507
740	672	Skiddo	Grass	NaN	66	65	48	62	57	52	6	False	350
741	673	Gogoat	Grass	NaN	123	100	62	97	81	68	6	False	531
799	721	Volcanion	Fire	Water	80	110	120	130	90	70	6	True	600

122 rows × 13 columns

```
In [102]: df.loc[df['Name'].str.contains('^Pi[a-z]*', regex=True)] #^ makes sure that it starts with pi
```

Out[102]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	Total	
20	Pidgey	Normal	Flying	40	45	40	35	35	56	1	False	251	
21	Pidgeotto	Normal	Flying	63	60	55	50	50	71	1	False	349	
22	Pidgeot	Normal	Flying	83	80	75	70	70	101	1	False	479	
23	PidgeotMega Pidgeot	Normal	Flying	83	80	80	135	80	121	1	False	579	
30	Pikachu	Electric		35	55	40	50	50	90	1	False	320	
136	Pinsir	Bug		65	125	100	55	70	85	1	False	500	
137	PinsirMega Pinsir	Bug	Flying	65	155	120	65	90	105	1	False	600	
186	Pichu	Electric		NaN	20	40	15	35	35	60	2	False	205
219	Pineco	Bug		50	65	90	35	35	15	2	False	290	
239	Piloswine	Ice	Ground	100	100	80	60	60	50	2	False	450	
438	Piplup	Water		53	51	53	61	56	40	4	False	314	
558	Pignite	Fire	Fighting	90	93	55	70	55	55	5	False	418	
578	Pidove	Normal	Flying	50	55	50	36	30	43	5	False	264	

## Conditional changes in columns

```
In [104]: df.loc[df['Type 1']=='Fire', 'Type 1'] = 'Flamer'
```

In [105]: df

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Flamer	NaN	39	52	43	60	50	65	1	False	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	50	100	150	100	150	50	6	True	600
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700
797	720	HoopaHoopa Confined	Psychic	Ghost	80	110	60	150	130	70	6	True	600
798	720	HoopaHoopa Unbound	Psychic	Dark	80	160	60	170	130	80	6	True	680
799	721	Volcanion	Flamer	Water	80	110	120	130	90	70	6	True	600

800 rows × 13 columns

```
In [106]: df
```

Out[106]:

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Flamer	NaN	39	52	43	60	50	65	1	False	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	50	100	150	100	150	50	6	True	600
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700
797	720	HoopoHoopo Confined	Psychic	Ghost	80	110	60	150	130	70	6	True	600
798	720	HoopoHoopo Unbound	Psychic	Dark	80	160	60	170	130	80	6	True	680
799	721	Volcanion	Flamer	Water	80	110	120	130	90	70	6	True	600

800 rows × 13 columns

```
In [111]: df.loc[df['Type 1']=='Flamer','Legendary'] = True #all flamer pokemons are now Legendary
```

```
In [112]: df
```

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	False	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	False	625
4	4	Charmander	Flamer	NaN	39	52	43	60	50	65	1	True	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	50	100	150	100	150	50	6	True	600
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	6	True	700
797	720	HoopoHoopo Confined	Psychic	Ghost	80	110	60	150	130	70	6	True	600
798	720	HoopoHoopo Unbound	Psychic	Dark	80	160	60	170	130	80	6	True	680
799	721	Volcanion	Flamer	Water	80	110	120	130	90	70	6	True	600

800 rows × 13 columns

```
In [113]: #for changes in multiple frame we need to send a list instead of a string
```

```
In [118]: df.loc[df['Total']>500,['Generation','Legendary']]=['MODIFIED1','MODIFIED2']
```

In [119]: df

	#	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	318
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	False	405
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	MODIFIED1	MODIFIED2	525
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	MODIFIED1	MODIFIED2	625
4	4	Charmander	Flamer	NaN	39	52	43	60	50	65	1	True	309
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	719	Diancie	Rock	Fairy	50	100	150	100	150	50	MODIFIED1	MODIFIED2	600
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	110	110	MODIFIED1	MODIFIED2	700
797	720	HoopoHoopo Confined	Psychic	Ghost	80	110	60	150	130	70	MODIFIED1	MODIFIED2	600
798	720	HoopoHoopo Unbound	Psychic	Dark	80	160	60	170	130	80	MODIFIED1	MODIFIED2	680
799	721	Volcanion	Flamer	Water	80	110	120	130	90	70	MODIFIED1	MODIFIED2	600

800 rows × 13 columns

## Aggregate statistics (Groupby)

```
In [120]: df.groupby(['Type 1']).mean()          #error because of Panda version

-----
NotImplementedError                                Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\groupby\groupby.py:1490, in GroupBy._cython_agg_general.array_func(values)
1489     try:
1490         result = self.grouper._cython_operation(
1491             "aggregate",
1492             values,
1493             how,
1494             axis=data.ndim - 1,
1495             min_count=min_count,
1496             **kwargs,
1497         )
1498     except NotImplementedError:
1499         # generally if we have numeric_only=False
1500         # and non-applicable functions
1501         # try to python agg
1502         # TODO: shouldn't min_count matter?
```

```
In [121]: df= pd.read_csv('pokemon_data.csv')
```

```
In [123]: df.groupby(['Type 1']).mean(numeric_only=True)
```

Out[123]:

	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
<b>Type 1</b>									
<b>Bug</b>	334.492754	56.884058	70.971014	70.724638	53.869565	64.797101	61.681159	3.217391	0.000000
<b>Dark</b>	461.354839	66.806452	88.387097	70.225806	74.645161	69.516129	76.161290	4.032258	0.064516
<b>Dragon</b>	474.375000	83.312500	112.125000	86.375000	96.843750	88.843750	83.031250	3.875000	0.375000
<b>Electric</b>	363.500000	59.795455	69.090909	66.295455	90.022727	73.704545	84.500000	3.272727	0.090909
<b>Fairy</b>	449.529412	74.117647	61.529412	65.705882	78.529412	84.705882	48.588235	4.117647	0.058824
<b>Fighting</b>	363.851852	69.851852	96.777778	65.925926	53.111111	64.703704	66.074074	3.370370	0.000000
<b>Fire</b>	327.403846	69.903846	84.769231	67.769231	88.980769	72.211538	74.442308	3.211538	0.096154
<b>Flying</b>	677.750000	70.750000	78.750000	66.250000	94.250000	72.500000	102.500000	5.500000	0.500000
<b>Ghost</b>	486.500000	64.437500	73.781250	81.187500	79.343750	76.468750	64.343750	4.187500	0.062500
<b>Grass</b>	344.871429	67.271429	73.214286	70.800000	77.500000	70.428571	61.928571	3.357143	0.042857
<b>Ground</b>	356.281250	73.781250	95.750000	84.843750	56.468750	62.750000	63.906250	3.156250	0.125000
<b>Ice</b>	423.541667	72.000000	72.750000	71.416667	77.541667	76.291667	63.458333	3.541667	0.083333
<b>Normal</b>	319.173469	77.275510	73.469388	59.846939	55.816327	63.724490	71.551020	3.051020	0.020408
<b>Poison</b>	251.785714	67.250000	74.678571	68.821429	60.428571	64.392857	63.571429	2.535714	0.000000
<b>Psychic</b>	380.807018	70.631579	71.456140	67.684211	98.403509	86.280702	81.491228	3.385965	0.245614
<b>Rock</b>	392.727273	65.363636	92.863636	100.795455	63.340909	75.477273	55.909091	3.454545	0.090909
<b>Steel</b>	442.851852	65.222222	92.703704	126.370370	67.518519	80.629630	55.259259	3.851852	0.148148
<b>Water</b>	303.089286	72.062500	74.151786	72.946429	74.812500	70.517857	65.964286	2.857143	0.035714

```
In [125]: df.groupby(['Type 1']).mean(numeric_only=True).sort_values('HP', ascending=False)
```

```
Out[125]:
```

Type 1	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
<b>Dragon</b>	474.375000	83.312500	112.125000	86.375000	96.843750	88.843750	83.031250	3.875000	0.375000
<b>Normal</b>	319.173469	77.275510	73.469388	59.846939	55.816327	63.724490	71.551020	3.051020	0.020408
<b>Fairy</b>	449.529412	74.117647	61.529412	65.705882	78.529412	84.705882	48.588235	4.117647	0.058824
<b>Ground</b>	356.281250	73.781250	95.750000	84.843750	56.468750	62.750000	63.906250	3.156250	0.125000
<b>Water</b>	303.089286	72.062500	74.151786	72.946429	74.812500	70.517857	65.964286	2.857143	0.035714
<b>Ice</b>	423.541667	72.000000	72.750000	71.416667	77.541667	76.291667	63.458333	3.541667	0.083333
<b>Flying</b>	677.750000	70.750000	78.750000	66.250000	94.250000	72.500000	102.500000	5.500000	0.500000
<b>Psychic</b>	380.807018	70.631579	71.456140	67.684211	98.403509	86.280702	81.491228	3.385965	0.245614
<b>Fire</b>	327.403846	69.903846	84.769231	67.769231	88.980769	72.211538	74.442308	3.211538	0.096154
<b>Fighting</b>	363.851852	69.851852	96.777778	65.925926	53.111111	64.703704	66.074074	3.370370	0.000000
<b>Grass</b>	344.871429	67.271429	73.214286	70.800000	77.500000	70.428571	61.928571	3.357143	0.042857
<b>Poison</b>	251.785714	67.250000	74.678571	68.821429	60.428571	64.392857	63.571429	2.535714	0.000000
<b>Dark</b>	461.354839	66.806452	88.387097	70.225806	74.645161	69.516129	76.161290	4.032258	0.064516
<b>Rock</b>	392.727273	65.363636	92.863636	100.795455	63.340909	75.477273	55.909091	3.454545	0.090909
<b>Steel</b>	442.851852	65.222222	92.703704	126.370370	67.518519	80.629630	55.259259	3.851852	0.148148
<b>Ghost</b>	486.500000	64.437500	73.781250	81.187500	79.343750	76.468750	64.343750	4.187500	0.062500
<b>Electric</b>	363.500000	59.795455	69.090909	66.295455	90.022727	73.704545	84.500000	3.272727	0.090909
<b>Bug</b>	334.492754	56.884058	70.971014	70.724638	53.869565	64.797101	61.681159	3.217391	0.000000

```
In [127]: df.groupby(['Type 1']).sum(numeric_only=True)
```

```
Out[127]:
```

Type 1	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
Bug	23080	3925	4897	4880	3717	4471	4256	222	0
Dark	14302	2071	2740	2177	2314	2155	2361	125	2
Dragon	15180	2666	3588	2764	3099	2843	2657	124	12
Electric	15994	2631	3040	2917	3961	3243	3718	144	4
Fairy	7642	1260	1046	1117	1335	1440	826	70	1
Fighting	9824	1886	2613	1780	1434	1747	1784	91	0
Fire	17025	3635	4408	3524	4627	3755	3871	167	5
Flying	2711	283	315	265	377	290	410	22	2
Ghost	15568	2062	2361	2598	2539	2447	2059	134	2
Grass	24141	4709	5125	4956	5425	4930	4335	235	3
Ground	11401	2361	3064	2715	1807	2008	2045	101	4
Ice	10165	1728	1746	1714	1861	1831	1523	85	2
Normal	31279	7573	7200	5865	5470	6245	7012	299	2
Poison	7050	1883	2091	1927	1692	1803	1780	71	0
Psychic	21706	4026	4073	3858	5609	4918	4645	193	14
Rock	17280	2876	4086	4435	2787	3321	2460	152	4
Steel	11957	1761	2503	3412	1823	2177	1492	104	4
Water	33946	8071	8305	8170	8379	7898	7388	320	4

```
In [132]: df.groupby(['Type 1']).count()
```

```
Out[132]: #  Name  Type 2  HP  Attack  Defense  Sp. Atk  Sp. Def  Speed  Generation  Legendary
```

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

```
In [133]: # count only one column ??????
```

```
In [134]: df['count']=1
```

In [135]: df

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

800 rows × 13 columns

```
In [136]: df.groupby(['Type 1']).count()['count']
```

```
Out[136]: Type 1  
Bug           69  
Dark          31  
Dragon         32  
Electric       44  
Fairy          17  
Fighting       27  
Fire           52  
Flying          4  
Ghost          32  
Grass          70  
Ground         32  
Ice            24  
Normal         98  
Poison         28  
Psychic        57  
Rock           44  
Steel           27  
Water          112  
Name: count, dtype: int64
```

```
In [137]: df.groupby(['Type 1', 'Type 2']).count()['count']      # count with type 1 and 2
```

```
Out[137]: Type 1  Type 2  
Bug    Electric     2  
        Fighting     2  
        Fire         2  
        Flying        14  
        Ghost         1  
        ..  
Water   Ice          3  
        Poison        3  
        Psychic       5  
        Rock          4  
        Steel          1  
Name: count, Length: 136, dtype: int64
```

# Working with large amount data ????????

```
In [146]: new_df_2=pd.DataFrame(columns=df.columns)
for df2 in pd.read_csv('modified_poke.csv',chunksize=4):
    results=df2.groupby(['Type 1']).count()
    new_df_2=pd.concat([new_df_2,results])
```

```
In [148]: new_df_2
```

Out[148]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	count	Total
	Grass	4	4	NaN	4	4	4	4	4	4	4	4	4	4.0
	Fire	4	4	NaN	2	4	4	4	4	4	4	4	4	4.0
	Fire	1	1	NaN	1	1	1	1	1	1	1	1	1	1.0
	Water	3	3	NaN	0	3	3	3	3	3	3	3	3	3.0
	Bug	3	3	NaN	1	3	3	3	3	3	3	3	3	3.0
	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	Fairy	1	1	NaN	0	1	1	1	1	1	1	1	1	1.0
	Rock	1	1	NaN	1	1	1	1	1	1	1	1	1	1.0
	Fire	1	1	NaN	1	1	1	1	1	1	1	1	1	1.0
	Psychic	2	2	NaN	2	2	2	2	2	2	2	2	2	2.0
	Rock	1	1	NaN	1	1	1	1	1	1	1	1	1	1.0

460 rows × 14 columns

## removing duplicates

In [ ]:

```
# df.duplicated()  
# df.drop_duplicates()
```