

Why Pandas?

Better and more flexible than Excel

Loading Data using Pandas

```
In [2]: In ▶ import pandas as pd
```

```
df = pd.read_csv('pokemon_data.csv')
```

```
print(df.head(5)) #prints first 5 rows
```

```
print(df.tail(5)) #prints last 5 rows
```

```
# df_xlxs = pd.read_excel('pokemon_data.xlsx') // Used for import excel files
```

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

	Sp. Def	Speed	Generation	Legendary
0	65	45	1	False
1	80	60	1	False
2	100	80	1	False
3	120	80	1	False
4	50	65	1	False

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	\
795	719	Diancie	Rock	Fairy	50	100	150	100	
796	719	DiancieMega Diancie	Rock	Fairy	50	160	110	160	
797	720	HoopaHoopa Confined	Psychic	Ghost	80	110	60	150	
798	720	HoopaHoopa Unbound	Psychic	Dark	80	160	60	170	
799	721	Volcanion	Fire	Water	80	110	120	130	

	Sp. Def	Speed	Generation	Legendary
795	150	50	6	True
796	110	110	6	True
797	130	70	6	True
798	130	80	6	True
799	90	70	6	True

Now let's do some data reading...

```
In [3]: In ▶ # Read Headers
```

```
df.columns
```

```
Out[3]: Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk',  
              'Sp. Def', 'Speed', 'Generation', 'Legendary'],  
              dtype='object')
```

In [4]:  # Read Each Column

```
print(df[['Name', 'Type 1', 'HP']]) # Specify Which Columns
```

```
      Name  Type 1  HP
0  Bulbasaur   Grass  45
1   Ivysaur   Grass  60
2   Venusaur   Grass  80
3  VenusaurMega Venusaur   Grass  80
4   Charmander   Fire  39
..      ...      ...  ..
795      Diancie   Rock  50
796  DiancieMega Diancie   Rock  50
797  HoopaHoopa Confined  Psychic  80
798  HoopaHoopa Unbound  Psychic  80
799      Volcanion   Fire  80
```

[800 rows x 3 columns]

In [*]:  # Read Each Row

```
print(df.iloc[0:4])

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

# df.loc[df['Type 1'] == "Grass"] ### OUTPUT TOO LONG but it works!
```

In [6]:  # Read A Specific Location

```
print(df.iloc[2,1])
```

Venusaur

Now lets learn how to sort our data...

In [7]:  df.describe() # Gives us statistical values for each header

Out[7]:

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

```
In [8]: df.sort_values(['Type 1', 'HP'], ascending=[1,0])
```

Out[8]:

	#	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	HeracrossMega 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

```
In [9]: df.sort_values(['Type 1', 'HP'], ascending=False)
```

Out[9]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
351	321	Wailord	Water	NaN	170	90	45	90	45	60	3	False
655	594	Alomomola	Water	NaN	165	75	80	40	45	65	5	False
142	131	Lapras	Water	Ice	130	85	80	85	95	60	1	False
145	134	Vaporeon	Water	NaN	130	65	60	110	95	65	1	False
350	320	Wailmer	Water	NaN	130	70	35	70	35	60	3	False
...
314	290	Nincada	Bug	Ground	31	45	90	30	30	40	3	False
462	415	Combee	Bug	Flying	30	30	42	30	42	70	4	False
603	543	Venipede	Bug	Poison	30	45	59	30	39	57	5	False
230	213	Shuckle	Bug	Rock	20	10	230	10	230	5	2	False
316	292	Shedinja	Bug	Ghost	1	90	45	30	30	40	3	False

800 rows × 12 columns

Manipulating Data...

In [10]: df.head()

Out[10]:

	#	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 [11]: # Combining values to Create a Total Column

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

In [12]: df.head() # VOILA! New Column

Out[12]:

	#	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 [13]: # Dropping Columns

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

In [14]: df.head() # NO MORE TOTAL COLUMN!!!

Out[14]:

	#	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 [15]: # Adding Columns Again

```
df['Total'] = df.iloc[:,4:10].sum(axis = 1) # Axis = 0 would add Verticall... Axis = 1
```

In [16]: `df.head()`

Out[16]:

	#	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 [17]: `# Rearranging Columns`

```
cols = list(df.columns.values) # getting columns
df = df[cols[0:4] + [cols[-1]] + cols[4:12]] # Actually moving the column over
df.head(5)
```

Out[17]:

	#	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

Saving our Data from Python itself!

In [18]: `df.to_csv('modified_csv.csv', index = False) # Saving file without the index numbers to`
`df.to_excel('modified_xlsx.xlsx', index=False) # In excel format`

FILTERING DATA!!!

```
In [19]: new_df = df.loc[(df['Type 1'] == 'Grass') & (df['Type 2'] == "Poison") & (df['HP'] > 70)]
new_df.head() # Only the grass, poison, and HP == 70 types...
```

Out[19]:

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

```
In [20]: new_df2 = df.loc[(df['Type 1'] == 'Grass') | (df['Type 2'] == "Poison") | (df['HP'] > 70)]
new_df2.head() # Only the grass, poison, OR HP == 70 types...
```

Out[20]:

	#	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
6	6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False

```
In [21]: df.head()
```

Out[21]:

	#	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

```
In [22]: new_df.head() # old index stays
```

Out[22]:

	#	Name	Type_1	Type_2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
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
50	45	Vileplume	Grass	Poison	490	75	80	85	110	90	50	1	False
77	71	Victreebel	Grass	Poison	490	80	105	65	100	70	70	1	False
652	591	Amoonguss	Grass	Poison	464	114	85	70	85	80	30	5	False

```
In [23]: new_df2.head() # old index stays
```

Out[23]:

	#	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
6	6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False

```
In [24]: new_df.to_csv('filtered csv.csv', index = False)
```

```
In [25]: # To update index

new_df = new_df.reset_index()
new_df # new updated index
```

Out[25]:

	index	#	Name	Type_1	Type_2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
1	3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
2	50	45	Vileplume	Grass	Poison	490	75	80	85	110	90	50	1	False
3	77	71	Victreebel	Grass	Poison	490	80	105	65	100	70	70	1	False
4	652	591	Amoonguss	Grass	Poison	464	114	85	70	85	80	30	5	False

```
In [26]: new_df2.reset_index(drop = True, inplace=True)
new_df2
```

Out[26]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leg
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	
4	6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	
...
377	717	Yveltal	Dark	Flying	680	126	131	95	131	98	99	6	
378	718	Zygarde50% Forme	Dragon	Ground	600	108	100	121	81	95	95	6	
379	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	
380	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	
381	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	

382 rows × 13 columns



In [27]: # More Adv. Filtering of Data

```
df.loc[df['Name'].str.contains("Mega")] # only columns that contain mega
```

Out[27]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	I
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	
7	6	CharizardMega Charizard X	Fire	Dragon	634	78	130	111	130	85	100	1	
8	6	CharizardMega Charizard Y	Fire	Flying	634	78	104	78	159	115	100	1	
12	9	BlastoiseMega Blastoise	Water	NaN	630	79	103	120	135	115	78	1	
19	15	BeedrillMega Beedrill	Bug	Poison	495	65	150	40	15	80	145	1	
23	18	PidgeotMega Pidgeot	Normal	Flying	579	83	80	80	135	80	121	1	
71	65	AlakazamMega Alakazam	Psychic	NaN	590	55	50	65	175	95	150	1	
87	80	SlowbroMega Slowbro	Water	Psychic	590	95	75	180	130	80	30	1	
102	94	GengarMega Gengar	Ghost	Poison	600	60	65	80	170	95	130	1	
124	115	KangaskhanMega Kangaskhan	Normal	NaN	590	105	125	100	60	100	100	1	
137	127	PinsirMega Pinsir	Bug	Flying	600	65	155	120	65	90	105	1	
141	130	GyaradosMega Gyarados	Water	Dark	640	95	155	109	70	130	81	1	
154	142	AerodactylMega Aerodactyl	Rock	Flying	615	80	135	85	70	95	150	1	
163	150	MewtwoMega Mewtwo X	Psychic	Fighting	780	106	190	100	154	100	130	1	
164	150	MewtwoMega Mewtwo Y	Psychic	NaN	780	106	150	70	194	120	140	1	
168	154	Meganium	Grass	NaN	525	80	82	100	83	100	80	2	
196	181	AmpharosMega Ampharos	Electric	Dragon	610	90	95	105	165	110	45	2	
224	208	SteelixMega Steelix	Steel	Ground	610	75	125	230	55	95	30	2	
229	212	ScizorMega Scizor	Bug	Steel	600	70	150	140	65	100	75	2	
232	214	HeracrossMega Heracross	Bug	Fighting	600	80	185	115	40	105	75	2	
248	229	HoundoomMega Houndoom	Dark	Fire	600	75	90	90	140	90	115	2	
268	248	TyranitarMega Tyranitar	Rock	Dark	700	100	164	150	95	120	71	2	
275	254	SceptileMega Sceptile	Grass	Dragon	630	70	110	75	145	85	145	3	

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	I
279	257	BlazikenMega Blaziken	Fire	Fighting	630	80	160	80	130	80	100	3	
283	260	SwampertMega Swampert	Water	Ground	635	100	150	110	95	110	70	3	
306	282	GardevoirMega Gardevoir	Psychic	Fairy	618	68	85	65	165	135	100	3	
327	302	SableyeMega Sableye	Dark	Ghost	480	50	85	125	85	115	20	3	
329	303	MawileMega Mawile	Steel	Fairy	480	50	105	125	55	95	50	3	
333	306	AggronMega Aggron	Steel	NaN	630	70	140	230	60	80	50	3	
336	308	MedichamMega Medicham	Fighting	Psychic	510	60	100	85	80	85	100	3	
339	310	ManectricMega Manectric	Electric	NaN	575	70	75	80	135	80	135	3	
349	319	SharpedoMega Sharpedo	Water	Dark	560	70	140	70	110	65	105	3	
354	323	CameruptMega Camerupt	Fire	Ground	560	70	120	100	145	105	20	3	
366	334	AltariaMega Altaria	Dragon	Fairy	590	75	110	110	110	105	80	3	
387	354	BanetteMega Banette	Ghost	NaN	555	64	165	75	93	83	75	3	
393	359	AbsolMega Absol	Dark	NaN	565	65	150	60	115	60	115	3	
397	362	GlalieMega Glalie	Ice	NaN	580	80	120	80	120	80	100	3	
409	373	SalamenceMega Salamence	Dragon	Flying	700	95	145	130	120	90	120	3	
413	376	MetagrossMega Metagross	Steel	Psychic	700	80	145	150	105	110	110	3	
418	380	LatiasMega Latias	Dragon	Psychic	700	80	100	120	140	150	110	3	
420	381	LatiosMega Latios	Dragon	Psychic	700	80	130	100	160	120	110	3	
426	384	RayquazaMega Rayquaza	Dragon	Flying	780	105	180	100	180	100	115	3	
476	428	LopunnyMega Lopunny	Normal	Fighting	580	65	136	94	54	96	135	4	
494	445	GarchompMega Garchomp	Dragon	Ground	700	108	170	115	120	95	92	4	
498	448	LucarioMega Lucario	Fighting	Steel	625	70	145	88	140	70	112	4	
511	460	AbomasnowMega Abomasnow	Grass	Ice	594	90	132	105	132	105	30	4	
527	475	GalladeMega Gallade	Psychic	Fighting	618	68	165	95	65	115	110	4	
591	531	AudinoMega Audino	Normal	Fairy	545	103	60	126	80	126	50	5	
796	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	

```
In [28]: df.loc[~df['Name'].str.contains("Mega")] # only columns that dont contain mega
```

Out[28]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leger
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	
5	5	Charmeleon	Fire	NaN	405	58	64	58	80	65	80	1	
...
794	718	Zygarde50% Forme	Dragon	Ground	600	108	100	121	81	95	95	6	
795	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	
797	720	HoopaaHoopaa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	
798	720	HoopaaHoopaa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	

751 rows × 13 columns

```
In [29]: # Regular Expressions

import re

df.loc[df["Type 1"].str.contains('fire|grass', regex=True)]
```

Out[29]:

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
---	------	--------	--------	-------	----	--------	---------	---------	---------	-------	------------	-----------

```
In [30]: df.loc[df["Type 1"].str.contains('fire|grass', flags=re.I, regex=True)] # regex to ignore
```

Out[30]:

	#	Name	Type ₁	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legend
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	I
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	I
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	I
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	I
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	I
...
735	667	Littleo	Fire	Normal	369	62	50	58	73	54	72	6	I
736	668	Pyroar	Fire	Normal	507	86	68	72	109	66	106	6	I
740	672	Skiddo	Grass	NaN	350	66	65	48	62	57	52	6	I
741	673	Gogoat	Grass	NaN	531	123	100	62	97	81	68	6	I
799	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	I

122 rows × 13 columns



In [31]:

df.loc[df["Name"].str.contains('pi[a-z]', flags=re.I, regex=True)] # anything with 'pi'

Out[31]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	L
13	10	Caterpie	Bug	NaN	195	45	30	35	20	20	45	1	
20	16	Pidgey	Normal	Flying	251	40	45	40	35	35	56	1	
21	17	Pidgeotto	Normal	Flying	349	63	60	55	50	50	71	1	
22	18	Pidgeot	Normal	Flying	479	83	80	75	70	70	101	1	
23	18	PidgeotMega Pidgeot	Normal	Flying	579	83	80	80	135	80	121	1	
30	25	Pikachu	Electric	NaN	320	35	55	40	50	50	90	1	
42	37	Vulpix	Fire	NaN	299	38	41	40	50	65	65	1	
76	70	Weepinbell	Grass	Poison	390	65	90	50	85	45	55	1	
84	78	Rapidash	Fire	NaN	500	65	100	70	80	80	105	1	
136	127	Pinsir	Bug	NaN	500	65	125	100	55	70	85	1	
137	127	PinsirMega Pinsir	Bug	Flying	600	65	155	120	65	90	105	1	
181	167	Spinarak	Bug	Poison	250	40	60	40	40	40	30	2	
186	172	Pichu	Electric	NaN	205	20	40	15	35	35	60	2	
202	187	Hoppip	Grass	Flying	250	35	35	40	35	55	50	2	
219	204	Pineco	Bug	NaN	290	50	65	90	35	35	15	2	
239	221	Piloswine	Ice	Ground	450	100	100	80	60	60	50	2	
266	247	Pupitar	Rock	Ground	410	70	84	70	65	70	51	2	
345	316	Gulpin	Poison	NaN	302	70	43	53	43	53	40	3	
357	326	Grumpig	Psychic	NaN	470	80	45	65	90	110	80	3	
358	327	Spinda	Normal	NaN	360	60	60	60	60	60	60	3	
359	328	Trapinch	Ground	NaN	290	45	100	45	45	45	10	3	
390	357	Tropius	Grass	Flying	460	99	68	83	72	87	51	3	
438	393	Piplup	Water	NaN	314	53	51	53	61	56	40	4	
463	416	Vespiquen	Bug	Flying	474	70	80	102	80	102	40	4	
488	440	Happiny	Normal	NaN	220	100	5	5	15	65	30	4	
490	442	Spiritomb	Ghost	Dark	485	50	92	108	92	108	35	4	
502	452	Drapion	Poison	Dark	500	70	90	110	60	75	95	4	
557	498	Tepig	Fire	NaN	308	65	63	45	45	45	45	5	
558	499	Pignite	Fire	Fighting	418	90	93	55	70	55	55	5	
578	519	Pidove	Normal	Flying	264	50	55	50	36	30	43	5	
596	536	Palpitoad	Water	Ground	384	75	65	55	65	55	69	5	
716	648	MeloettaPirouette Forme	Normal	Fighting	600	100	128	90	77	77	128	5	
718	650	Chespin	Grass	NaN	313	56	61	65	48	45	38	6	

```
In [32]: df.loc[df["Name"].str.contains('^pi[a-z]', flags=re.I, regex=True)] # anything that sta
```

Out[32]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legen
20	16	Pidgey	Normal	Flying	251	40	45	40	35	35	56	1	I
21	17	Pidgeotto	Normal	Flying	349	63	60	55	50	50	71	1	I
22	18	Pidgeot	Normal	Flying	479	83	80	75	70	70	101	1	I
23	18	PidgeotMega Pidgeot	Normal	Flying	579	83	80	80	135	80	121	1	I
30	25	Pikachu	Electric	NaN	320	35	55	40	50	50	90	1	I
136	127	Pinsir	Bug	NaN	500	65	125	100	55	70	85	1	I
137	127	PinsirMega Pinsir	Bug	Flying	600	65	155	120	65	90	105	1	I
186	172	Pichu	Electric	NaN	205	20	40	15	35	35	60	2	I
219	204	Pineco	Bug	NaN	290	50	65	90	35	35	15	2	I
239	221	Piloswine	Ice	Ground	450	100	100	80	60	60	50	2	I
438	393	Piplup	Water	NaN	314	53	51	53	61	56	40	4	I
558	499	Pignite	Fire	Fighting	418	90	93	55	70	55	55	5	I
578	519	Pidove	Normal	Flying	264	50	55	50	36	30	43	5	I

Conditonal Changes Breaking the Rule of Pokemon

```
In [39]: df.loc[df['Type 1'] == 'Fire', 'Type 1'] = 'Flamer'
df # All fire types are now flammers... basically if x == y then x = z
```

Out[39]:

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

800 rows × 13 columns



```
In [40]: df.loc[df['Type 1'] == 'Flamer', 'Legendary'] = True
df # all flammers are now also Legendary...
```

Out[40]:

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

800 rows × 13 columns



Grouping Data for Mathematical Values


```
In [41]: df.groupby(['Type 1']).mean() # average value of each of the types of pokemon
```

Out[41]:

	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Ge
Type 1									
Bug	334.492754	378.927536	56.884058	70.971014	70.724638	53.869565	64.797101	61.681159	
Dark	461.354839	445.741935	66.806452	88.387097	70.225806	74.645161	69.516129	76.161290	
Dragon	474.375000	550.531250	83.312500	112.125000	86.375000	96.843750	88.843750	83.031250	
Electric	363.500000	443.409091	59.795455	69.090909	66.295455	90.022727	73.704545	84.500000	
Fairy	449.529412	413.176471	74.117647	61.529412	65.705882	78.529412	84.705882	48.588235	
Fighting	363.851852	416.444444	69.851852	96.777778	65.925926	53.111111	64.703704	66.074074	
Flamer	327.403846	458.076923	69.903846	84.769231	67.769231	88.980769	72.211538	74.442308	
Flying	677.750000	485.000000	70.750000	78.750000	66.250000	94.250000	72.500000	102.500000	
Ghost	486.500000	439.562500	64.437500	73.781250	81.187500	79.343750	76.468750	64.343750	
Grass	344.871429	421.142857	67.271429	73.214286	70.800000	77.500000	70.428571	61.928571	
Ground	356.281250	437.500000	73.781250	95.750000	84.843750	56.468750	62.750000	63.906250	
Ice	423.541667	433.458333	72.000000	72.750000	71.416667	77.541667	76.291667	63.458333	
Normal	319.173469	401.683673	77.275510	73.469388	59.846939	55.816327	63.724490	71.551020	
Poison	251.785714	399.142857	67.250000	74.678571	68.821429	60.428571	64.392857	63.571429	
Psychic	380.807018	475.947368	70.631579	71.456140	67.684211	98.403509	86.280702	81.491228	
Rock	392.727273	453.750000	65.363636	92.863636	100.795455	63.340909	75.477273	55.909091	
Steel	442.851852	487.703704	65.222222	92.703704	126.370370	67.518519	80.629630	55.259259	
Water	303.089286	430.455357	72.062500	74.151786	72.946429	74.812500	70.517857	65.964286	



In [42]:

df.groupby(['Type 1']).median()

Out[42]:

	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
Type 1										
Bug	291.0	395.0	60.0	65.0	60.0	50.0	60.0	60.0	3.0	0.0
Dark	509.0	465.0	65.0	88.0	70.0	65.0	65.0	70.0	5.0	0.0
Dragon	443.5	600.0	80.0	113.5	90.0	105.0	90.0	90.0	4.0	0.0
Electric	403.5	477.5	60.0	65.0	65.0	95.0	79.5	88.0	4.0	0.0
Fairy	669.0	405.0	78.0	52.0	66.0	75.0	79.0	45.0	6.0	0.0
Fighting	308.0	455.0	70.0	100.0	70.0	40.0	63.0	60.0	3.0	0.0
Flamer	289.5	482.0	70.0	84.5	64.0	85.0	67.5	78.5	3.0	1.0
Flying	677.5	557.5	79.0	85.0	75.0	103.5	80.0	116.0	5.5	0.5
Ghost	487.0	464.5	59.5	66.0	72.5	65.0	75.0	60.5	4.0	0.0
Grass	372.0	430.0	65.5	70.0	66.0	75.0	66.0	58.5	3.5	0.0
Ground	363.5	440.0	75.0	85.0	84.5	47.5	62.5	65.0	3.0	0.0
Ice	371.5	467.5	70.0	67.0	75.0	77.5	70.0	62.0	3.0	0.0
Normal	296.5	415.0	70.0	70.5	60.0	50.0	60.5	71.0	3.0	0.0
Poison	139.5	401.5	67.5	74.0	67.0	60.0	60.5	62.5	1.5	0.0
Psychic	386.0	485.0	68.0	57.0	65.0	95.0	90.0	80.0	3.0	0.0
Rock	362.5	467.5	68.5	95.0	100.0	60.0	70.0	50.0	3.0	0.0
Steel	379.0	500.0	60.0	89.0	120.0	55.0	80.0	50.0	3.0	0.0
Water	275.0	455.0	70.0	72.0	70.0	70.0	65.0	65.0	3.0	0.0

In [44]:

df.groupby(['Type 1']).mean().sort_values('Defense',ascending = False)

Out[44]:

	#	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Ge
Type 1									
Steel	442.851852	487.703704	65.222222	92.703704	126.370370	67.518519	80.629630	55.259259	
Rock	392.727273	453.750000	65.363636	92.863636	100.795455	63.340909	75.477273	55.909091	
Dragon	474.375000	550.531250	83.312500	112.125000	86.375000	96.843750	88.843750	83.031250	
Ground	356.281250	437.500000	73.781250	95.750000	84.843750	56.468750	62.750000	63.906250	
Ghost	486.500000	439.562500	64.437500	73.781250	81.187500	79.343750	76.468750	64.343750	
Water	303.089286	430.455357	72.062500	74.151786	72.946429	74.812500	70.517857	65.964286	
Ice	423.541667	433.458333	72.000000	72.750000	71.416667	77.541667	76.291667	63.458333	
Grass	344.871429	421.142857	67.271429	73.214286	70.800000	77.500000	70.428571	61.928571	
Bug	334.492754	378.927536	56.884058	70.971014	70.724638	53.869565	64.797101	61.681159	
Dark	461.354839	445.741935	66.806452	88.387097	70.225806	74.645161	69.516129	76.161290	
Poison	251.785714	399.142857	67.250000	74.678571	68.821429	60.428571	64.392857	63.571429	
Flamer	327.403846	458.076923	69.903846	84.769231	67.769231	88.980769	72.211538	74.442308	
Psychic	380.807018	475.947368	70.631579	71.456140	67.684211	98.403509	86.280702	81.491228	
Electric	363.500000	443.409091	59.795455	69.090909	66.295455	90.022727	73.704545	84.500000	
Flying	677.750000	485.000000	70.750000	78.750000	66.250000	94.250000	72.500000	102.500000	
Fighting	363.851852	416.444444	69.851852	96.777778	65.925926	53.111111	64.703704	66.074074	
Fairy	449.529412	413.176471	74.117647	61.529412	65.705882	78.529412	84.705882	48.588235	
Normal	319.173469	401.683673	77.275510	73.469388	59.846939	55.816327	63.724490	71.551020	



In []: