Project: Milestone 4

Author: Jordan,Andrew

Date: 08/12/2022

In [17]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import json
import requests
import urllib
import pokepy #api wrapper
import pokebase as pb #api wrapper
import random
from typing import Dict, Iterable, List, KeysView, ItemsView, ValuesView
```

1.Create list with pokemon api data

In [18]:
```
#Loop call to list
x=0
result=[]
while x < 500:
    x=x +1
    url_stat = "https://pokeapi.co/api/v2/pokemon/" + str (x) + "/" #Set url

    response_stat = requests.get(url_stat)
    pokelist= pd.json_normalize(json.loads(response_stat.text))[['id','name','height','weight','base_experience']].va
    result.append(pokelist)
```

2.Create dataframe

In [19]:
```
#Create dataframe from list
api_df=pd.DataFrame(result,columns=['id','name','height','weight','base_experience'])
api_df
```

Out[19]:

	id	name	height	weight	base_experience
0	1	bulbasaur	7	69	64
1	2	ivysaur	10	130	142
2	3	venusaur	20	1000	263
3	4	charmander	6	85	62
4	5	charmeleon	11	190	142
...	...	...	...	...	...
495	496	servine	8	160	145
496	497	serperior	33	630	238
497	498	tepig	5	99	62
498	499	pignite	10	555	146
499	500	emboar	16	1500	238

500 rows × 5 columns

3.Fix Casing

In [20]:
```
#capitalize first letter of each name
api_df['name'] = api_df['name'].apply(lambda x: x.capitalize())
api_df
```

Out[20]:

	id	name	height	weight	base_experience
0	1	Bulbasaur	7	69	64
1	2	Ivysaur	10	130	142
2	3	Venusaur	20	1000	263
3	4	Charmander	6	85	62
4	5	Charmeleon	11	190	142
...	...	...	...	...	...
495	496	Servine	8	160	145
496	497	Serperior	33	630	238
497	498	Tepig	5	99	62
498	499	Pignite	10	555	146
499	500	Emboar	16	1500	238

500 rows × 5 columns

Following modifications are to practice, not actual modifications that will be made in final project

4.Remove pokemon with base experience above 100

In [21]:
```
api_df_practice = api_df[api_df['base_experience']<100]
api_df_practice
```

Out[21]:

	id	name	height	weight	base_experience
0	1	Bulbasaur	7	69	64
3	4	Charmander	6	85	62
6	7	Squirtle	5	90	63
9	10	Caterpie	3	29	39
10	11	Metapod	7	99	72
...	...	...	...	...	...
455	456	Finneon	4	70	66
457	458	Mantyke	10	650	69
458	459	Snover	10	505	67
494	495	Snivy	6	81	62
497	498	Tepig	5	99	62

180 rows × 5 columns

5.Create columns for abilities, fill with placeholder NaN values

In [23]:
```
api_df_practice = api_df_practice.assign(Ability1='',Ability2='',Ability3='')
api_df_practice
```

Out[23]:

	id	name	height	weight	base_experience	Ability1	Ability2	Ability3
0	1	Bulbasaur	7	69	64			
3	4	Charmander	6	85	62			
6	7	Squirtle	5	90	63			
9	10	Caterpie	3	29	39			
10	11	Metapod	7	99	72			
...	...	...	...	...	...	...	...	...
455	456	Finneon	4	70	66			
457	458	Mantyke	10	650	69			
458	459	Snover	10	505	67			
494	495	Snivy	6	81	62			
497	498	Tepig	5	99	62			

180 rows × 8 columns

In [30]:
```
api_df_practice = api_df_practice.replace(r'^\s*$', np.nan, regex=True)
api_df_practice
```

Out[30]:

	id	name	height	weight	base_experience	Ability1	Ability2	Ability3
0	1	Bulbasaur	7	69	64	NaN	NaN	NaN
3	4	Charmander	6	85	62	NaN	NaN	NaN
6	7	Squirtle	5	90	63	NaN	NaN	NaN
9	10	Caterpie	3	29	39	NaN	NaN	NaN
10	11	Metapod	7	99	72	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
455	456	Finneon	4	70	66	NaN	NaN	NaN
457	458	Mantyke	10	650	69	NaN	NaN	NaN
458	459	Snover	10	505	67	NaN	NaN	NaN
494	495	Snivy	6	81	62	NaN	NaN	NaN
497	498	Tepig	5	99	62	NaN	NaN	NaN

180 rows × 8 columns