Term Project

Author: Jordan,Andrew

Date: 08/20/2022

In [39]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sqlite3

%pwd
```

Out[39]:

```
'C:\\Users\\Andrew\\Documents\\Grad School\\DSC 540 - Data Preparation\\Project'
```

In [40]:

```
#Load data
pokemon1 = pd.read_csv("data/pokemon1.csv") #Flat file
pokemon2 = pd.read_csv("data/pokemon2.csv") #Website
pokemon3 = pd.read_csv("data/pokemon3.csv") #Api
```

In [41]:

```
#Matching column names between datasets
pokemon1 = pokemon1.rename(columns={'pokedex_number':'pokedex'})
pokemon2.columns= pokemon2.columns.str.lower()
pokemon3.columns= pokemon3.columns.str.lower()
```

In [42]:

```
print("Flat file column names:")
print(pokemon1.columns)

print("Website column names:")
print(pokemon2.columns)

print("Api column names:")
print(pokemon3.columns)
```

```
Flat file column names:
Index(['pokedex', 'name', 'generation', 'type1', 'type2', 'ability_1',
       'ability_2', 'ability_3', 'base_total', 'hp', 'attack', 'defense',
       'sp_attack', 'sp_defense', 'speed', 'against_bug', 'against_dark',
       'against_dragon', 'against_electric', 'against_fairy',
       'against_fighting', 'against_fire', 'against_flying', 'against_ghost',
       'against_grass', 'against_ground', 'against_ice', 'against_normal',
       'against_poison', 'against_psychic', 'against_rock', 'against_steel',
       'against_water', 'capture_rate', 'base_egg_steps', 'base_happiness',
       'is_legendary', 'is_mythical', 'is_mega'],
      dtype='object')
Website column names:
Index(['pokedex', 'name', 'total_stats', 'hp', 'attack', 'defense', 'spatk',
       'spdef', 'speed', 'type1', 'type2'],
      dtype='object')
Api column names:
Index(['pokedex', 'name', 'height', 'weight', 'base_experience'], dtype='object')
```

In [43]:

```
#Dropping specific columns from each db
pokemon1 = pokemon1.drop(['generation','base_total', 'hp', 'attack', 'defense',
       'sp_attack', 'sp_defense', 'speed','capture_rate', 'base_egg_steps', 'base_happiness',
       'is_legendary', 'is_mythical', 'is_mega'], axis=1)
pokemon2 = pokemon2.drop(['name','type1', 'type2'], axis=1)
pokemon3 = pokemon3.drop(['name'], axis=1)
```

In [44]:

```
print("Flat file column names:")
print(pokemon1.columns)

print("Website column names:")
print(pokemon2.columns)

print("Api column names:")
print(pokemon3.columns)
```

```
Flat file column names:
Index(['pokedex', 'name', 'type1', 'type2', 'ability_1', 'ability_2',
       'ability_3', 'against_bug', 'against_dark', 'against_dragon',
       'against_electric', 'against_fairy', 'against_fighting', 'against_fire',
       'against_flying', 'against_ghost', 'against_grass', 'against_ground',
       'against_ice', 'against_normal', 'against_poison', 'against_psychic',
       'against_rock', 'against_steel', 'against_water'],
      dtype='object')
Website column names:
Index(['pokedex', 'total_stats', 'hp', 'attack', 'defense', 'spatk', 'spdef',
       'speed'],
      dtype='object')
Api column names:
Index(['pokedex', 'height', 'weight', 'base_experience'], dtype='object')
```

In [45]:

```
#Create connection
conn = sqlite3.connect('termproject.sqlite')
#Create cursor
cur = conn.cursor()
```

In [46]:

```
#Convert dataframes to lists
flat_list = pokemon1.values.tolist()
web_list = pokemon2.values.tolist()
api_list = pokemon3.values.tolist()
```

In [47]:

```
#Create table1
table1 = """ CREATE TABLE myflatfile_db2
            (pokedex INTEGER,
            name CHAR(50),
            type1 CHAR(50),
            type2 CHAR(50),
            ability_1 CHAR(50),
            ability_2 CHAR(50),
            ability_3 CHAR(50),
            'against_bug' FLOAT,
            'against_dark' FLOAT,
           'against_dragon' FLOAT,
           'against_electric' FLOAT,
           'against_fairy' FLOAT,
           'against_fighting' FLOAT,
           'against_fire' FLOAT,
           'against_flying' FLOAT,
           'against_ghost' FLOAT,
           'against_grass' FLOAT,
           'against_ground' FLOAT,
           'against_ice' FLOAT,
           'against_normal' FLOAT,
           'against_poison' FLOAT,
           'against_psychic' FLOAT,
           'against_rock' FLOAT,
           'against_steel' FLOAT,
           'against_water' FLOAT
            ); """
#Add table1 to database
conn.execute(table1)
conn.commit()
```

In [48]:

```
#Add data to sql table
stmt1 = "INSERT INTO myflatfile_db2 VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
conn.executemany(stmt1, flat_list)
conn.commit()
```

In [49]:

```
#Create table2
table2 = """ CREATE TABLE mywebsite_db2
            (pokedex INTEGER,
            total_stats INTEGER,
            hp INTEGER,
            attack INTEGER,
            defense INTEGER,
            spatk INTEGER,
            spdef INTEGER,
            speed INTEGER
            ); """
#Add table2 to database
conn.execute(table2)
conn.commit()
```

In [50]:

```
#Add data to sql table
stmt2 = "INSERT INTO mywebsite_db2 VALUES(?, ?, ?, ?, ?, ?, ?, ?)"
conn.executemany(stmt2, web_list)
conn.commit()
```

In [51]:

```
#Create table3
table3 = """ CREATE TABLE myapi_db2
            (pokedex INTEGER,
            height INTEGER,
            weight INTEGER,
            base_experience INTEGER
            ); """
#Add table3 to database
conn.execute(table3)
conn.commit()
```

In [52]:

```
#Add data to sql table
stmt3 = "INSERT INTO myapi_db2 VALUES(?, ?, ?, ?)"
conn.executemany(stmt3, api_list)
conn.commit()
```

In [53]:

```
#Queries
pokemon1_query = pd.read_sql_query ('''Select * From myflatfile_db2 ''', conn)
pokemon2_query = pd.read_sql_query ('''Select * From mywebsite_db2 ''', conn)
pokemon3_query = pd.read_sql_query ('''Select * From myapi_db2 ''', conn)
```

In [60]:

```
#Load tables
pokemon1_df = pd.DataFrame(pokemon1_query, columns = ['pokedex', 'name', 'type1', 'type2', 'ability_1', 'ability_2',
       'ability_3', 'against_bug', 'against_dark', 'against_dragon',
       'against_electric', 'against_fairy', 'against_fighting', 'against_fire',
       'against_flying', 'against_ghost', 'against_grass', 'against_ground',
       'against_ice', 'against_normal', 'against_poison', 'against_psychic',
       'against_rock', 'against_steel', 'against_water'])
pokemon2_df = pd.DataFrame(pokemon2_query, columns = ['pokedex','total_stats', 'hp', 'attack', 'defense', 'spatk', 's
pokemon3_df = pd.DataFrame(pokemon3_query, columns = ['pokedex','height','weight','base_experience'])
```

In [55]:

```
#Close connection
conn.close()
```

In [73]:

```
#Join dataframes
flat_web_df = pokemon1_df.join(pokemon2_df.set_index('pokedex'), on='pokedex', lsuffix='_flat',rsuffix='_web')
```

In [74]:

```
#Check for success
flat_web_df.head()
```

Out[74]:

	pokedex	name	type1	type2	ability_1	ability_2	ability_3	against_bug	against_dark	against_dragon	...	against_rock	agai
0	1	Bulbasaur	Grass	Poison	Overgrow	Chlorophyll	None	1.0	1.0	1.0	...	1.0	
1	2	Ivysaur	Grass	Poison	Overgrow	Chlorophyll	None	1.0	1.0	1.0	...	1.0	
2	3	Venusaur	Grass	Poison	Overgrow	Chlorophyll	None	1.0	1.0	1.0	...	1.0	
3	4	Charmander	Fire	None	Blaze	Solar Power	None	0.5	1.0	1.0	...	2.0	
4	5	Charmeleon	Fire	None	Blaze	Solar Power	None	0.5	1.0	1.0	...	2.0	

5 rows × 32 columns

In [78]:

```
#Join remaining dataframe
complete_pokemon_df = flat_web_df.join(pokemon3_df.set_index('pokedex'), on='pokedex')
```

In [79]:

```
#Check for success
complete_pokemon_df
```

Out[79]:

	pokedex	name	type1	type2	ability_1	ability_2	ability_3	against_bug	against_dark	against_dragon	...	total_stats
0	1	Bulbasaur	Grass	Poison	Overgrow	Chlorophyll	None	1.00	1.0	1.0	...	318
1	2	Ivysaur	Grass	Poison	Overgrow	Chlorophyll	None	1.00	1.0	1.0	...	405
2	3	Venusaur	Grass	Poison	Overgrow	Chlorophyll	None	1.00	1.0	1.0	...	525
3	4	Charmander	Fire	None	Blaze	Solar Power	None	0.50	1.0	1.0	...	309
4	5	Charmeleon	Fire	None	Blaze	Solar Power	None	0.50	1.0	1.0	...	405
...	...	...	...	...	...	...	...	...	...	...	...	...
900	901	Ursaluna	Normal	Ground	Guts	Bulletproof	Unnerve	1.00	1.0	1.0	...	550
901	902	Basculegion male	Water	Ghost	Rattled	Adaptability	Mold Breaker	0.50	2.0	1.0	...	530
902	903	Sneasler	Poison	Fighting	Pressure	Poison Touch	None	0.25	0.5	1.0	...	510
903	904	Overqwil	Dark	Poison	Poison Point	Swift Swim	Intimidate	1.00	0.5	1.0	...	510
904	905	Enamorus Incarnate Forme	Fairy	Flying	Healer	Contrary	None	0.25	0.5	0.0	...	580

905 rows × 35 columns

In [88]:

```
complete_pokemon_df.to_csv('C:\\Users\\Andrew\\Documents\\Grad School\\DSC 540 - Data Preparation\\Project/final_poke
```

SUCCESS!