Question 6 - Using the data from Question 3, write code to analyze the data and answer the following questions Note 1. Draw plots to demonstrate the analysis for the following questions for better visualizations.

2. Write code comments wherever required for code understanding

Insights to be drawn -

● Get all Pokemons whose spawn rate is less than 5%

● Get all Pokemons that have less than 4 weaknesses

● Get all Pokemons that have no multipliers at all

● Get all Pokemons that do not have more than 2 evolutions

● Get all Pokemons whose spawn time is less than 300 seconds.

Note - spawn time format is "05:32", so assume "minute: second" format and perform the analysis.

● Get all Pokemon who have more than two types of capabilities

Ans:

In [10]:
```
import pandas as pd
import numpy as np
import requests
```

In [11]:
```
# Retrieve the raw JSON data

def json_to_csv(link):
    response = requests.get(link)
    data = response.json()["pokemon"]

    # Convert JSON data to DataFrame
    df = pd.DataFrame(data)

    df.to_csv("Output.csv", index=False)
```

In [12]:
```
link = "https://raw.githubusercontent.com/Biuni/PokemonGO-Pokedex/master/pokedex.json"
json_to_csv(link)
```

In [13]:
```
df1 = pd.read_csv("Output.csv")
```

In [14]:
```
df1
```

Out[14]:

	id	num	name	img	type	height	weight	candy	candy_count	egg	spaw
0	1	1	Bulbasaur	http://www.serebii.net/pokemongo/pokemon/001.png	['Grass', 'Poison']	0.71 m	6.9 kg	Bulbasaur Candy	25.0	2 km	
1	2	2	Ivysaur	http://www.serebii.net/pokemongo/pokemon/002.png	['Grass', 'Poison']	0.99 m	13.0 kg	Bulbasaur Candy	100.0	Not in Eggs	
2	3	3	Venusaur	http://www.serebii.net/pokemongo/pokemon/003.png	['Grass', 'Poison']	2.01 m	100.0 kg	Bulbasaur Candy	NaN	Not in Eggs	
3	4	4	Charmander	http://www.serebii.net/pokemongo/pokemon/004.png	['Fire']	0.61 m	8.5 kg	Charmander Candy	25.0	2 km	
4	5	5	Charmeleon	http://www.serebii.net/pokemongo/pokemon/005.png	['Fire']	1.09 m	19.0 kg	Charmander Candy	100.0	Not in Eggs	
...	...	...	...	...	...	...	...	...	...	...	
146	147	147	Dratini	http://www.serebii.net/pokemongo/pokemon/147.png	['Dragon']	1.80 m	3.3 kg	Dratini Candy	25.0	10 km	
147	148	148	Dragonair	http://www.serebii.net/pokemongo/pokemon/148.png	['Dragon']	3.99 m	16.5 kg	Dratini Candy	100.0	Not in Eggs	
148	149	149	Dragonite	http://www.serebii.net/pokemongo/pokemon/149.png	['Dragon', 'Flying']	2.21 m	210.0 kg	Dratini Candy	NaN	Not in Eggs	
149	150	150	Mewtwo	http://www.serebii.net/pokemongo/pokemon/150.png	['Psychic']	2.01 m	122.0 kg	None	NaN	Not in Eggs	
150	151	151	Mew	http://www.serebii.net/pokemongo/pokemon/151.png	['Psychic']	0.41 m	4.0 kg	None	NaN	Not in Eggs	

151 rows × 17 columns

In [15]:
```python
# Get all Pokemons whose spawn rate is less than 5%
five_perc = df1["spawn_chance"].quantile(0.05)
df1[df1["spawn_chance"] < five_perc]["name"]
```

Out[15]:
```
131         Ditto
143      Articuno
144        Zapdos
145       Moltres
148     Dragonite
149        Mewtwo
150           Mew
Name: name, dtype: object
```

In [16]:
```python
import ast
```

In [17]:
```python
# Get all Pokemons that have less than 4 weaknesses

df1["weaknesses"] = df1["weaknesses"].apply(lambda x : ast.literal_eval(x))
df1[df1["weaknesses"].apply(lambda x : len(x)) < 4]["name"]
```

Out[17]:
```
3       Charmander
4       Charmeleon
5        Charizard
6         Squirtle
7        Wartortle
           ...
145        Moltres
146        Dratini
147      Dragonair
149         Mewtwo
150            Mew
Name: name, Length: 102, dtype: object
```

In [18]:
```python
# Get all Pokemons that have no multipliers at all
df1[df1["multipliers"].isna()]["name"]
```

```
Out[18]: 2         Venusaur
         5        Charizard
         8        Blastoise
         11      Butterfree
         14        Beedrill
                    ...
         144         Zapdos
         145        Moltres
         148      Dragonite
         149         Mewtwo
         150            Mew
         Name: name, Length: 81, dtype: object
```

```
In [19]: df1
```

Out[19]:

	id	num	name	img	type	height	weight	candy	candy_count	egg	spaw
0	1	1	Bulbasaur	http://www.serebii.net/pokemongo/pokemon/001.png	['Grass', 'Poison']	0.71 m	6.9 kg	Bulbasaur Candy	25.0	2 km	
1	2	2	Ivysaur	http://www.serebii.net/pokemongo/pokemon/002.png	['Grass', 'Poison']	0.99 m	13.0 kg	Bulbasaur Candy	100.0	Not in Eggs	
2	3	3	Venusaur	http://www.serebii.net/pokemongo/pokemon/003.png	['Grass', 'Poison']	2.01 m	100.0 kg	Bulbasaur Candy	NaN	Not in Eggs	
3	4	4	Charmander	http://www.serebii.net/pokemongo/pokemon/004.png	['Fire']	0.61 m	8.5 kg	Charmander Candy	25.0	2 km	
4	5	5	Charmeleon	http://www.serebii.net/pokemongo/pokemon/005.png	['Fire']	1.09 m	19.0 kg	Charmander Candy	100.0	Not in Eggs	
...	...	...	...	...	...	...	...	...	...	...	
146	147	147	Dratini	http://www.serebii.net/pokemongo/pokemon/147.png	['Dragon']	1.80 m	3.3 kg	Dratini Candy	25.0	10 km	
147	148	148	Dragonair	http://www.serebii.net/pokemongo/pokemon/148.png	['Dragon']	3.99 m	16.5 kg	Dratini Candy	100.0	Not in Eggs	
148	149	149	Dragonite	http://www.serebii.net/pokemongo/pokemon/149.png	['Dragon', 'Flying']	2.21 m	210.0 kg	Dratini Candy	NaN	Not in Eggs	
149	150	150	Mewtwo	http://www.serebii.net/pokemongo/pokemon/150.png	['Psychic']	2.01 m	122.0 kg	None	NaN	Not in Eggs	
150	151	151	Mew	http://www.serebii.net/pokemongo/pokemon/151.png	['Psychic']	0.41 m	4.0 kg	None	NaN	Not in Eggs	

151 rows × 17 columns

```
In [20]: # Get all Pokemons that do not have more than 2 evolutions

         df = df1.copy()
```

```
In [21]: df["next_evolution"].dropna(inplace=True)
```

```
In [22]: df["Evloutions_2"] = df["next_evolution"].dropna().apply(lambda x : ast.literal_eval(x)).apply(lambda x : len(x
```

```
In [23]: df.loc[df["Evloutions_2"] == True]["name"]
```

```
Out[23]: 1          Ivysaur
         4       Charmeleon
         7        Wartortle
         10         Metapod
         13          Kakuna
         16       Pidgeotto
         18         Rattata
         20         Spearow
         22           Ekans
         24         Pikachu
         26       Sandshrew
         29        Nidorina
         32        Nidorino
         34        Clefairy
         36          Vulpix
         38      Jigglypuff
         40           Zubat
         43           Gloom
         45           Paras
         47         Venonat
         49         Diglett
         51          Meowth
         53         Psyduck
         55          Mankey
         57       Growlithe
         60       Poliwhirl
         63         Kadabra
         66         Machoke
         69      Weepinbell
         71       Tentacool
         74        Graveler
         76          Ponyta
         78        Slowpoke
         80       Magnemite
         83           Doduo
         85            Seel
         87          Grimer
         89        Shellder
         92         Haunter
         95         Drowzee
         97          Krabby
         99         Voltorb
         101      Exeggcute
         103         Cubone
         108        Koffing
         110        Rhyhorn
         115         Horsea
         117        Goldeen
         119         Staryu
         128       Magikarp
         137        Omanyte
         139         Kabuto
         147      Dragonair
         Name: name, dtype: object
```

```
In [24]: df1
```

Out[24]:

	id	num	name	img	type	height	weight	candy	candy_count	egg	spaw
0	1	1	Bulbasaur	http://www.serebii.net/pokemongo/pokemon/001.png	['Grass', 'Poison']	0.71 m	6.9 kg	Bulbasaur Candy	25.0	2 km	
1	2	2	Ivysaur	http://www.serebii.net/pokemongo/pokemon/002.png	['Grass', 'Poison']	0.99 m	13.0 kg	Bulbasaur Candy	100.0	Not in Eggs	
2	3	3	Venusaur	http://www.serebii.net/pokemongo/pokemon/003.png	['Grass', 'Poison']	2.01 m	100.0 kg	Bulbasaur Candy	NaN	Not in Eggs	
3	4	4	Charmander	http://www.serebii.net/pokemongo/pokemon/004.png	['Fire']	0.61 m	8.5 kg	Charmander Candy	25.0	2 km	
4	5	5	Charmeleon	http://www.serebii.net/pokemongo/pokemon/005.png	['Fire']	1.09 m	19.0 kg	Charmander Candy	100.0	Not in Eggs	
...	...	...	...	...	...	...	...	...	...	...	
146	147	147	Dratini	http://www.serebii.net/pokemongo/pokemon/147.png	['Dragon']	1.80 m	3.3 kg	Dratini Candy	25.0	10 km	
147	148	148	Dragonair	http://www.serebii.net/pokemongo/pokemon/148.png	['Dragon']	3.99 m	16.5 kg	Dratini Candy	100.0	Not in Eggs	
148	149	149	Dragonite	http://www.serebii.net/pokemongo/pokemon/149.png	['Dragon', 'Flying']	2.21 m	210.0 kg	Dratini Candy	NaN	Not in Eggs	
149	150	150	Mewtwo	http://www.serebii.net/pokemongo/pokemon/150.png	['Psychic']	2.01 m	122.0 kg	None	NaN	Not in Eggs	
150	151	151	Mew	http://www.serebii.net/pokemongo/pokemon/151.png	['Psychic']	0.41 m	4.0 kg	None	NaN	Not in Eggs	

151 rows × 17 columns

In [25]:
```
# Get all Pokemons whose spawn time is less than 300 seconds
df = df1.copy()
```

In [26]:
```
df["spawn_time_less_then_300"] = df["spawn_time"].dropna().apply(lambda x: (int(x.split(":")[0]) * 60) +(int(x.
```

In [27]:
```
df.loc[df["spawn_time_less_then_300"] == True]["name"]
```

Out[27]:
```
6        Squirtle
8       Blastoise
10        Metapod
12         Weedle
13         Kakuna
          ...
127        Tauros
129      Gyarados
134       Jolteon
136       Porygon
139        Kabuto
Name: name, Length: 75, dtype: object
```

In [28]:
```
# Get all Pokemon who have more than two types of capabilities
df = df1.copy()
```

In [29]:
```
df["type"] = df["type"].apply(lambda x : ast.literal_eval(x))
```

In [30]:
```
df[df["type"].apply(lambda x : len(x) > 2)]["name"]
#There are no pokemon more than two types of capabilities
```

Out[30]:
```
Series([], Name: name, dtype: object)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js