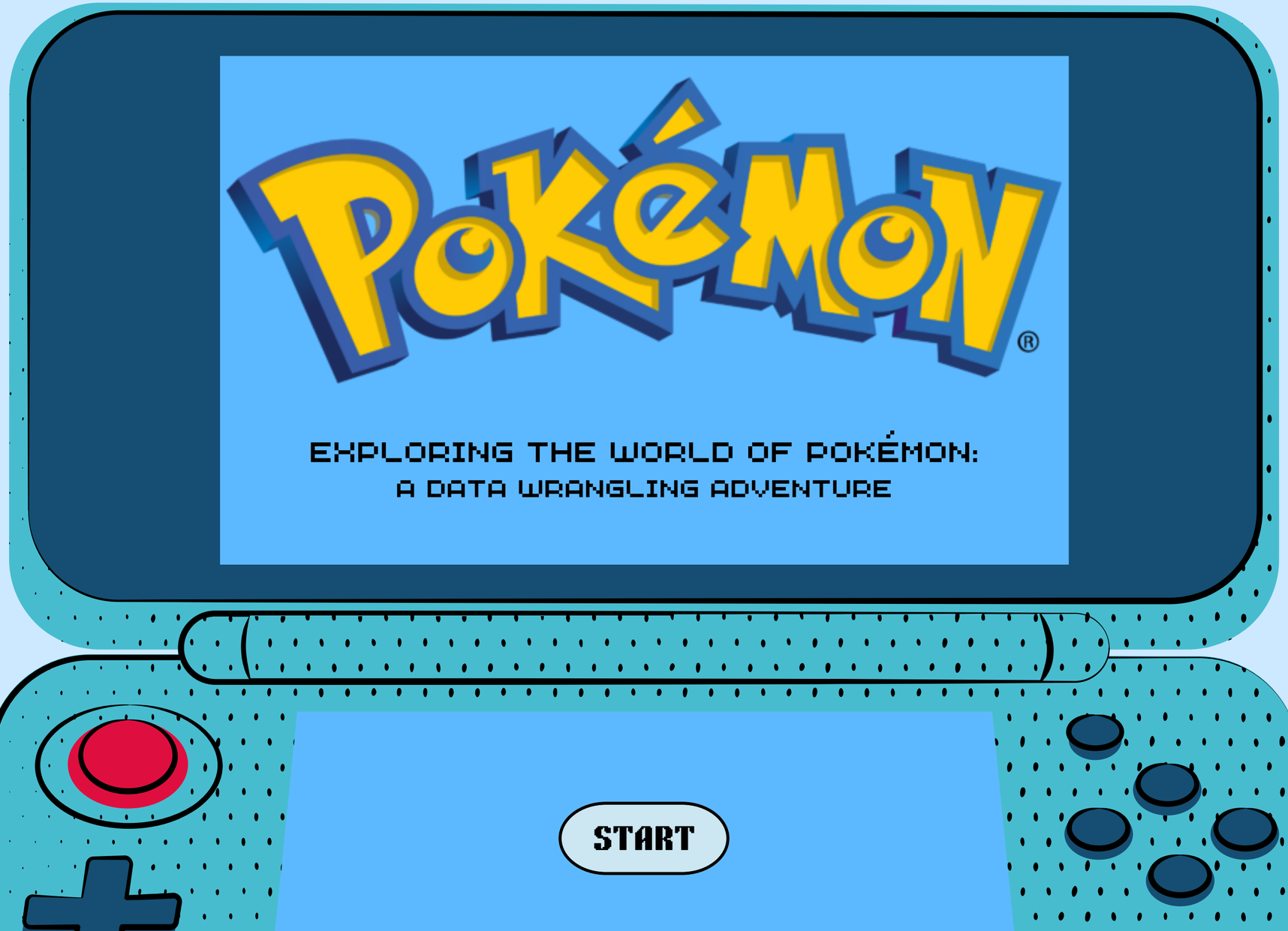

Pokémon
EXPLORING THE WORLD OF POKÉMON:
A DATA WRANGLING ADVENTURE
START

OUR TEAM

ALEX

CAMILO

STEPH

AL

ARYAN

AGENDA

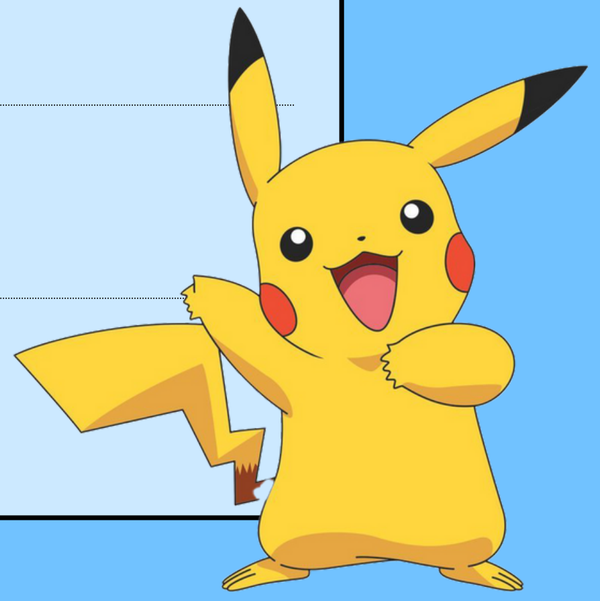
01 Introduction

02 The Data

03 Preparing and Cleaning the Data

04 Data Analysis

05 Conclusions



01

INTRODUCTION

Welcome to our Pokémon data adventure!

Pokémon, short for "Pocket Monsters," is a Japanese multimedia franchise first introduced by Nintendo in 1996.

The franchise encompasses video games, trading card games, animated television series, movies, toys, and more, making it one of the most successful and recognizable worldwide.

Overall, Pokémon has left an indelible mark on popular culture, inspiring generations of fans with its immersive world, memorable characters, and engaging gameplay experiences.

The project:

- Objective: Gain valuable insights into Pokémon characteristics, behaviors, and trends.
- Methodology: Utilizing Python programming and data wrangling techniques to uncover insights.

Key Points:

- Harnessing the power of Python for data analysis.
- Unleashing the potential of exploratory data analysis (EDA) techniques.
- Enhancing our understanding of Pokémon through data-driven insights.

02

THE DATA

DATA

- Name: Pokedex - df1
- Format: CSV
- Source: Kaggle / Master Pokemon Dataset and Corpus

1045 Rows

55 Columns

Data Types: float64(25), int64(12), object(18)

Memory usage: 457.2 KB

Contains information for Pokémon 1-1045 in the National Pokédex

- General Info: Name, Japanese name, Generation, Status, Species, Type, Height, Weight.
- Stats: The six primary factors determining how a Pokémon will perform in battle. They are HP, Attack, Defense, Special Attack, Special Defense, and Speed.
- Battle performance: How strong or weak the Pokémon performance is against different types.

DATA

- Name: Poke_corpus - df2
- Format: CSV
- Source: Kaggle / Master Pokemon Dataset and Corpus

1045 Rows

1 Column

Data Types: Object(1)

Memory usage: 16.3 KB

It contains a text corpus for each Pokémon created from all the information present in the Pokédex file. This is a detailed description of all the pokémon features.

DATA

- Name: Pokemon -df3
- Format: CSV
- Source: Kaggle / Pokemon

800 Rows

1 2 Columns

Data Types: bool(1), int64(8), object(3)

Memory usage: 75.8 KB

This dataframe contains information on 800 Pokemon from Seven Generations of Pokemon. The information contained in this dataset includes similar information to the Pokédex df but we are interested in one particular column:

- Legendary: True if Legendary Pokémon, False if not.

DATA

- Name: Pokemon_gender - gender_df
- Format: CSV obtained by web scrapping

- Source: ListFist.com / List of Pokémon by Gender

1075 Rows

3 Columns

Data Types: object (3)

Memory usage: 25.2 KB

The data represents all the Pokémon and the probability of being of each gender. Web scraping:

```
url = "https://listfist.com/list-of-pokemon-by-gender"
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')
table = soup.find('table')
```

```
rows = table.find_all('tr')
for row in rows[1:]:
    cells = row.find_all('td')
    pokemon_name = cells[1].text.strip()
    male_percentage = cells[2].text.strip()
    female_percentage = cells[3].text.strip()
    pokemon_names.append(pokemon_name)
    male_percentages.append(male_percentage)
    female_percentages.append(female_percentage)
```

```
poke_gender = pd.DataFrame({'Name': pokemon_names,
                            'Male Percentage': male_percentages,
                            'Female Percentage': female_percentages})
poke_gender.to_csv('pokemon_gender.csv', index=False)
```

DATA

- Name: Pokemon Images
- Format: png
- Source: Kaggle / Pokémon Image Dataset

816 Images

Size: 25.3 MB

Images of all Pokemon from generation 1 to generation 7

03

PREPARING AND CLEANING THE DATA

1. We drop the columns we are not going to use from Pokédex (df1)

```
columns_to_drop = ['japanese_name', 'base_friendship', 'base_experience',
                   'growth_rate', 'egg_type_number', 'egg_type_1', 'egg_type_2',
                   'percentage_male', 'egg_cycles', 'smogon_description', 'bulba_description',
                   'moves', 'ability_1_description', 'ability_2_description', 'ability_hidden_description',
                   'against_fire', 'against_water', 'against_electric', 'against_grass',
                   'against_ice', 'against_fight', 'against_poison', 'against_ground',
                   'against_flying', 'against_psychic', 'against_bug', 'against_rock',
                   'against_ghost', 'against_dragon', 'against_dark', 'against_steel',
                   'against_fairy']


df1=df1.drop(columns=columns_to_drop)
```

2. We concatenate Pokédex (df1) and Poke_Corpus (df2)

```
df_concat = pd.concat([df1, df2], axis=1)
```

3. We merge the concatenated_df with Pokemon (df3) to add the "Legendary" column

```
df4 = pd.merge(df_concat, df3[['Name', 'Legendary']], left_index=True, right_index=True, how='left')
```

4. We merge the gender_df f to add the female and male percentages

```
df = df4.merge(gender_df, left_on='name', right_index=True, how='left')
```

5. We create the new column "Image" where we add the image for each Pokémon

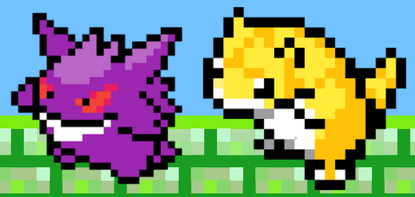
```python
def get_pokemon_image(row):
    image_path = image_folder / (row['name'].lower() + '.png')
    if image_path.is_file():
        with open(image_path, 'rb') as f:
            return f.read()
    else:
        return None

df['Image'] = df.apply(get_pokemon_image, axis=1)
```

```python
display(Image(data=df.at[1, 'Image']))
```

Resulting df: 1045 rows x 28 columns

#	Column	Non-Null Count	Dtype
0	pokedex_number	1045 non-null	int64
1	name	1045 non-null	object
2	generation	1045 non-null	int64
3	status	1045 non-null	object
4	species	1045 non-null	object
5	type_number	1045 non-null	int64
6	type_1	1045 non-null	object
7	type_2	553 non-null	object
8	height_m	1045 non-null	float64
9	weight_kg	1044 non-null	float64
10	abilities_number	1045 non-null	int64
11	total_points	1045 non-null	int64
12	hp	1045 non-null	int64
13	attack	1045 non-null	int64
14	defense	1045 non-null	int64
15	sp_attack	1045 non-null	int64
16	sp_defense	1045 non-null	int64
17	speed	1045 non-null	int64
18	catch_rate	1027 non-null	float64
19	against_normal	1045 non-null	float64
20	ability_1	1044 non-null	object
21	ability_2	939 non-null	object
22	ability_hidden	652 non-null	object
23	pokemon_info	1045 non-null	object
24	Legendary	800 non-null	object
25	Male Percentage	868 non-null	object
26	Female Percentage	868 non-null	object
27	Image	783 non-null	object



Cleaning the data, checking for missing values:

```
df.isna().sum()
```

pokedex_number	0
name	0
generation	0
status	0
species	0
type_number	0
type_1	0
type_2	492
height_m	0
weight_kg	1
abilities_number	0
total_points	0
hp	0
attack	0
defense	0
sp_attack	0
sp_defense	0
speed	0
catch_rate	18
against_normal	0
ability_1	1
ability_2	106
ability_hidden	393
pokemon_info	0
Legendary	245
Male Percentage	177
Female Percentage	177
Image	262

In our dataset, some columns contain null values, but these do not indicate missing information in all cases:

- 'type_2' column: 492 Pokémon have only one type.
- 'ability_2' column: 106 Pokémon have only one ability.
- 'ability_hidden' column: 393 Pokémon don't have a hidden ability.
- 'Image' column: 262 Pokémon lack images, which are not essential for our analysis.

Regarding the columns 'ability_1' and 'weight_kg', it's possible that null values indicate missing information. To address this:

1. We found the rows for the missing data:

```
df[df['ability_1'].isna()]
```

	pokedex_number	name
1039	895	Regidrago

```
df[df['weight_kg'].isna()]
```

	pokedex_number	name
1033	890	Eternatus Eternamax

2. We used one of official Pokémon websites to find the missing info and fill it:

```
df.loc[df['ability_1'].isna(), 'ability_1'] = "Dragon's Maw"
df.loc[df['weight_kg'].isna(), 'weight_kg'] = 950
```

To address the missing “Legendary” column classification for some Pokémon, we can utilize the 'status' column to fill in this information:

```
df['Legendary']=df.apply(lambda row: True if row['status']=='Legendary'else False
                         if pd.isnull(row['Legendary']) else row['Legendary'], axis=1)
```

If both the 'Male Percentage' and 'Female Percentage' columns contain null values, it implies an equal gender distribution, where the proportion is assumed to be 50% male and 50% female. In such cases, we can confidently fill the NaN values with 50% for both genders:

```
df['Male Percentage'].fillna('50%', inplace=True)
df['Female Percentage'].fillna('50%', inplace=True)
```

Check again for null values:

```
df.isna().sum()
```

pokedex_number	0
name	0
generation	0
status	0
species	0
type_number	0
type_1	0
type_2	492
height_m	0
weight_kg	0
abilities_number	0
total_points	0
hp	0
attack	0
defense	0
sp_attack	0
sp_defense	0
speed	0
catch_rate	18
against_normal	0
ability_1	0
ability_2	106
ability_hidden	393
pokemon_info	0
Legendary	0
Male Percentage	0
Female Percentage	0
Image	262

We handled correctly the missing data!

Looking at the numbers of Pokémon, we can see that Mega Evolutions and different forms are included in the list. Therefore, if we want to obtain the true number of Pokémon for each generation, we need to drop the rows where the 'pokedex_number' is repeated:

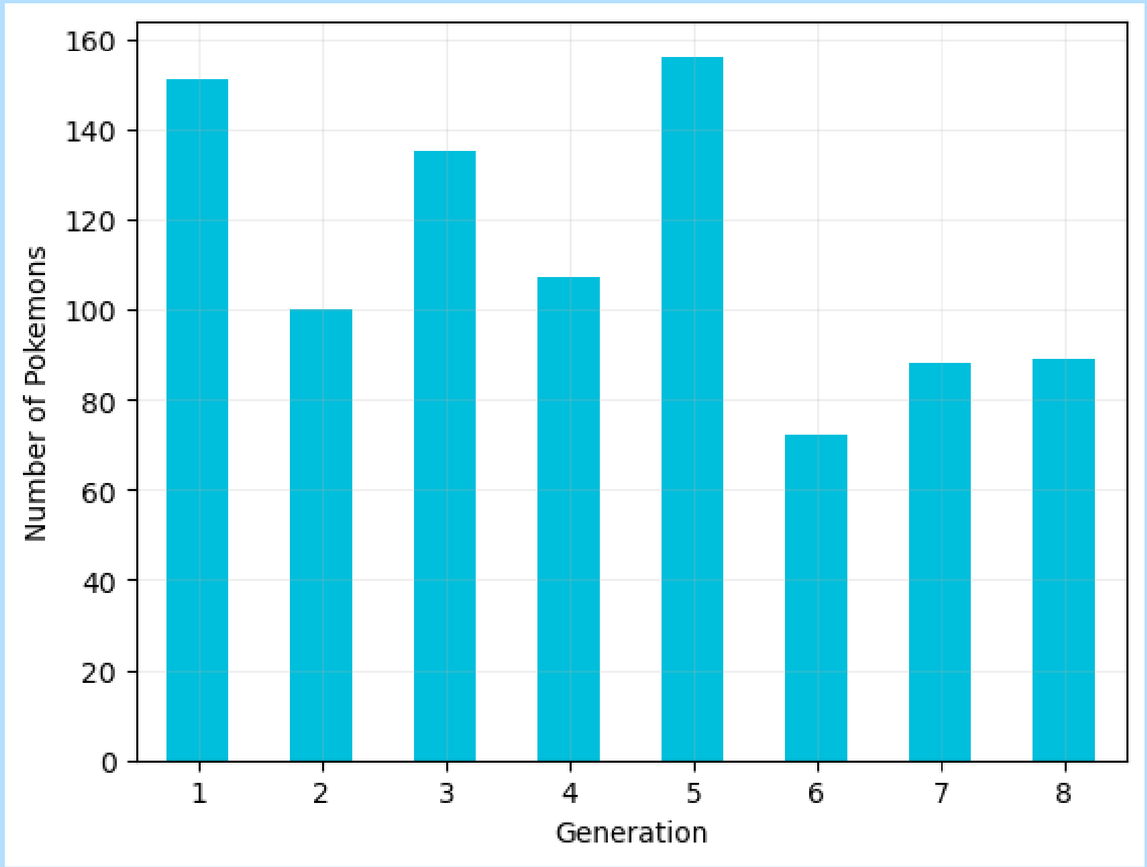
```
df_nomega = df.drop_duplicates(subset=['pokedex_number'])
poke_by_gen = df_nomega['generation'].value_counts().sort_index()
poke_by_gen
```

```
1    151
2    100
3    135
4    107
5    156
6     72
7     88
8     89
```

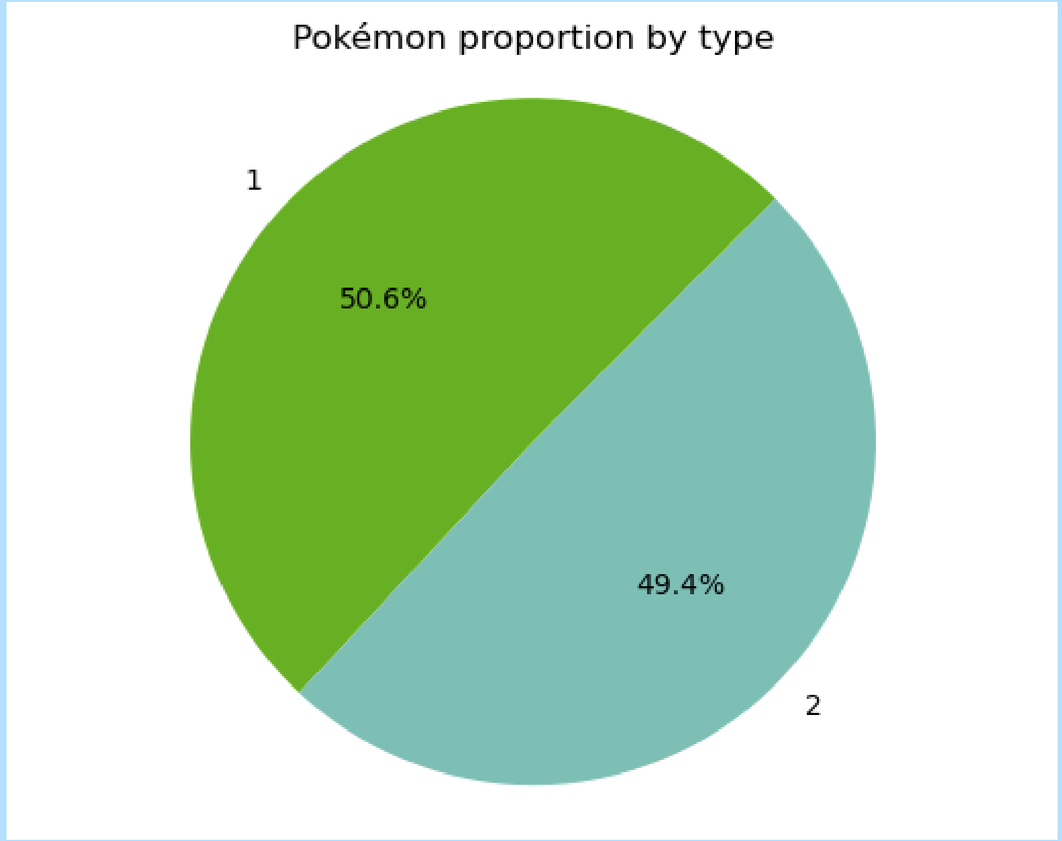
04

DATA ANALYSIS

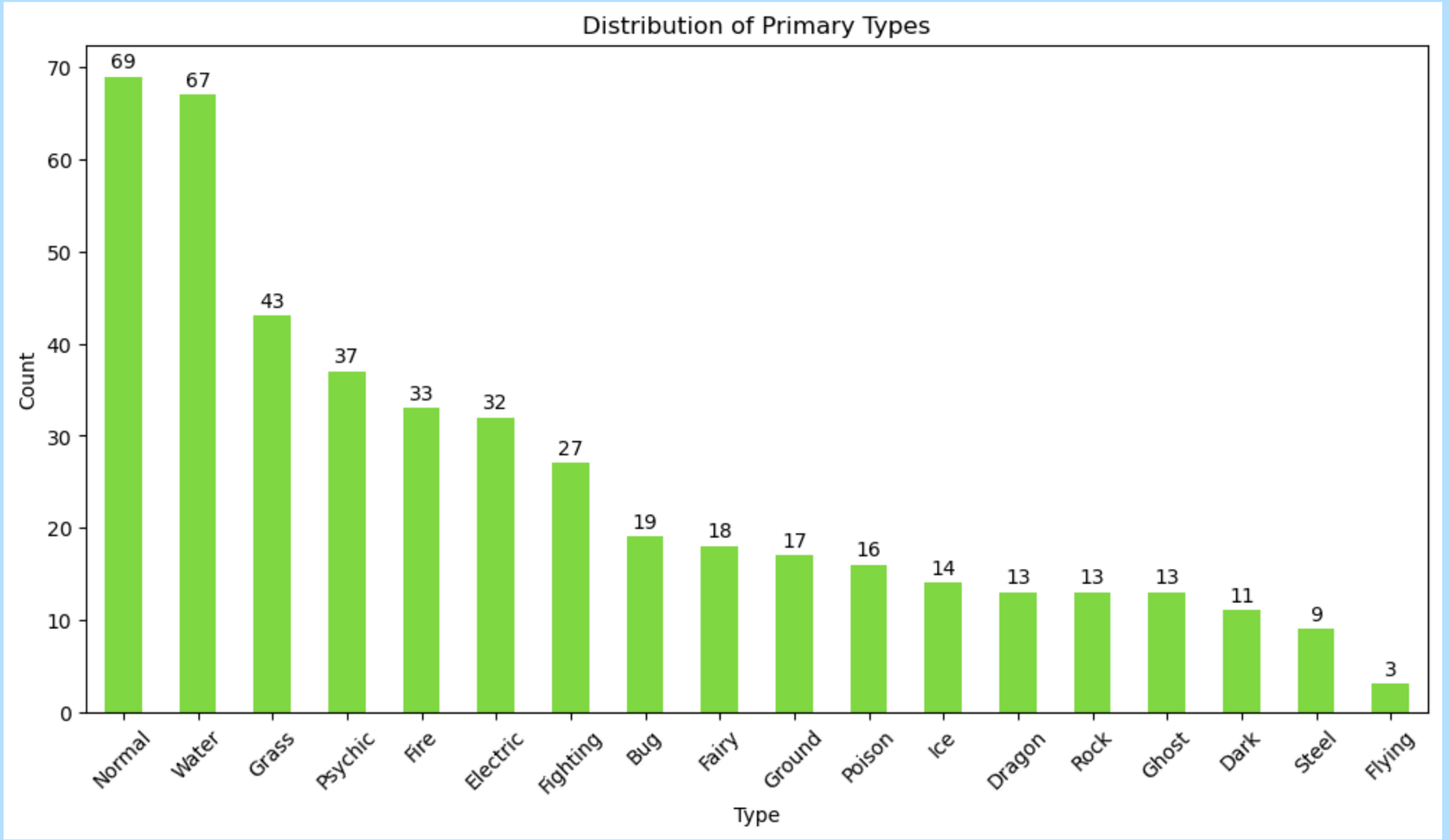
POKÉMON BY GENERATION



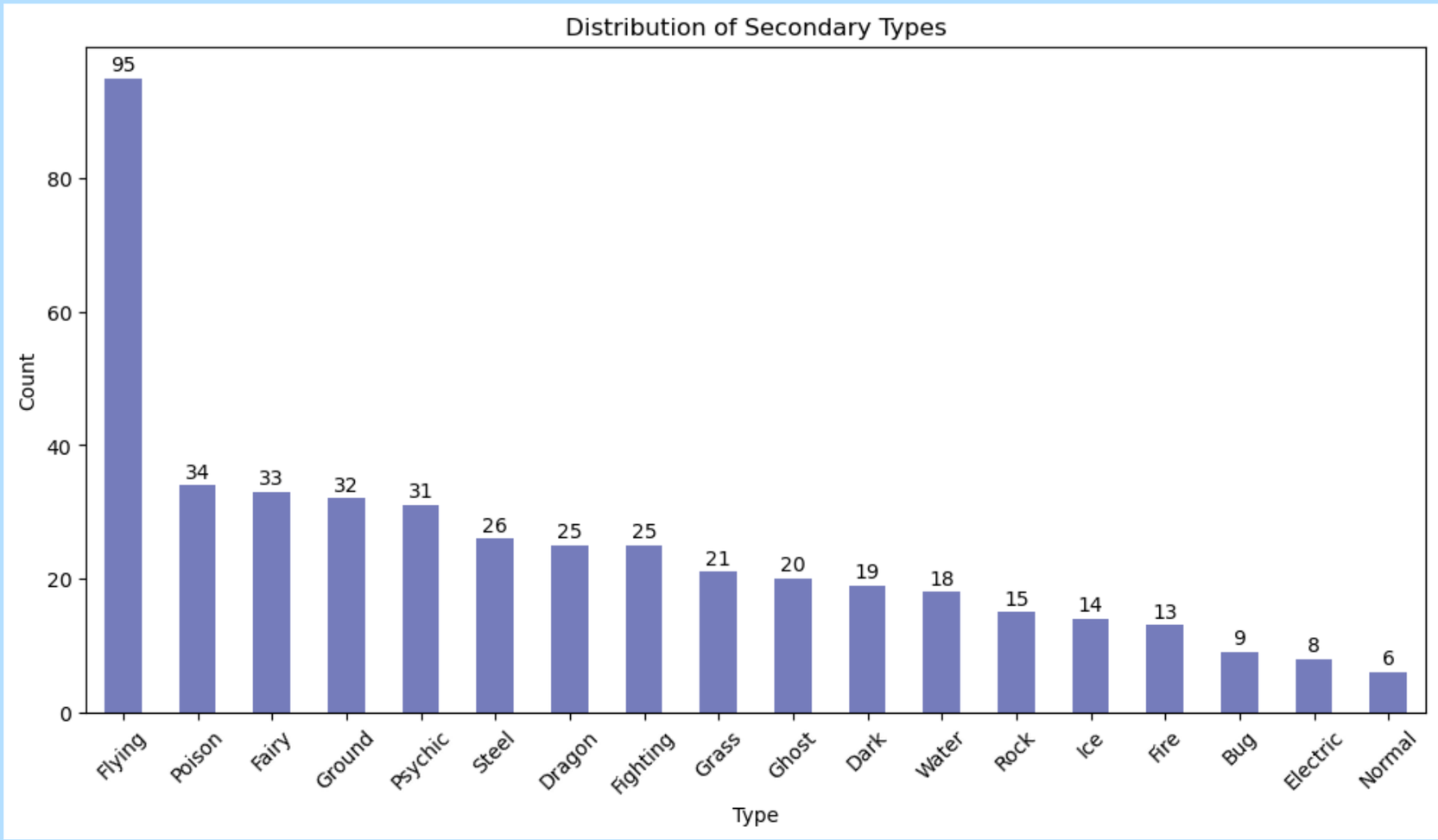
POKÉMON TYPE PROPORTION



TYPE 1 DISTRIBUTION



TYPE 2 DISTRIBUTION



GEN 1 VS GEN 2 TYPES DISTRIBUTION

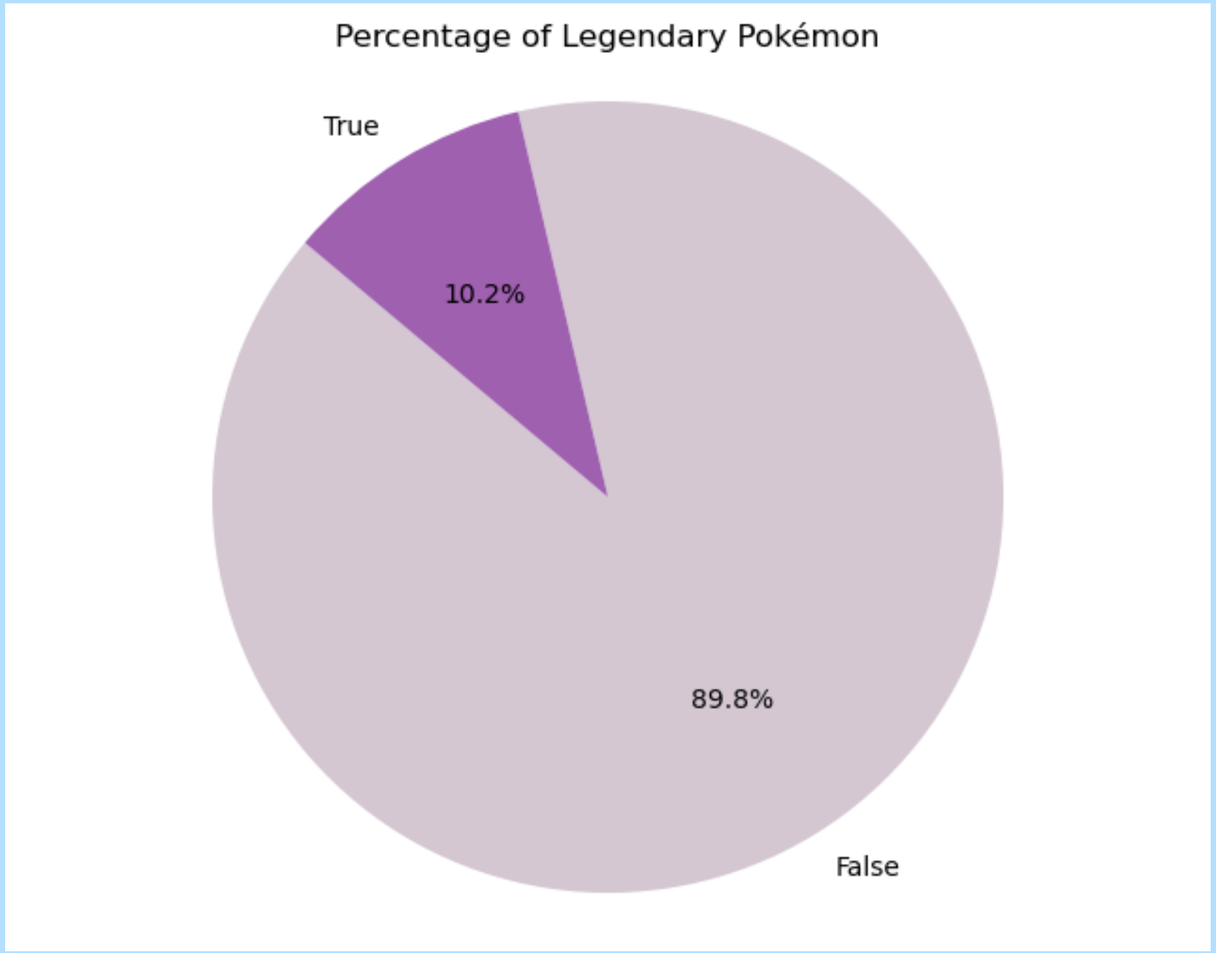
Null Hypothesis (H0): There is no significant difference in the distribution of Pokémon types between Generation 1 and Generation 2.
Alternative Hypothesis (Ha): There is a significant difference in the distribution of Pokémon types between Generation 1 and Generation 2.

```
Chi-square statistic: 203.4603768821582, p-value: 2.2842580105164193e-06
```

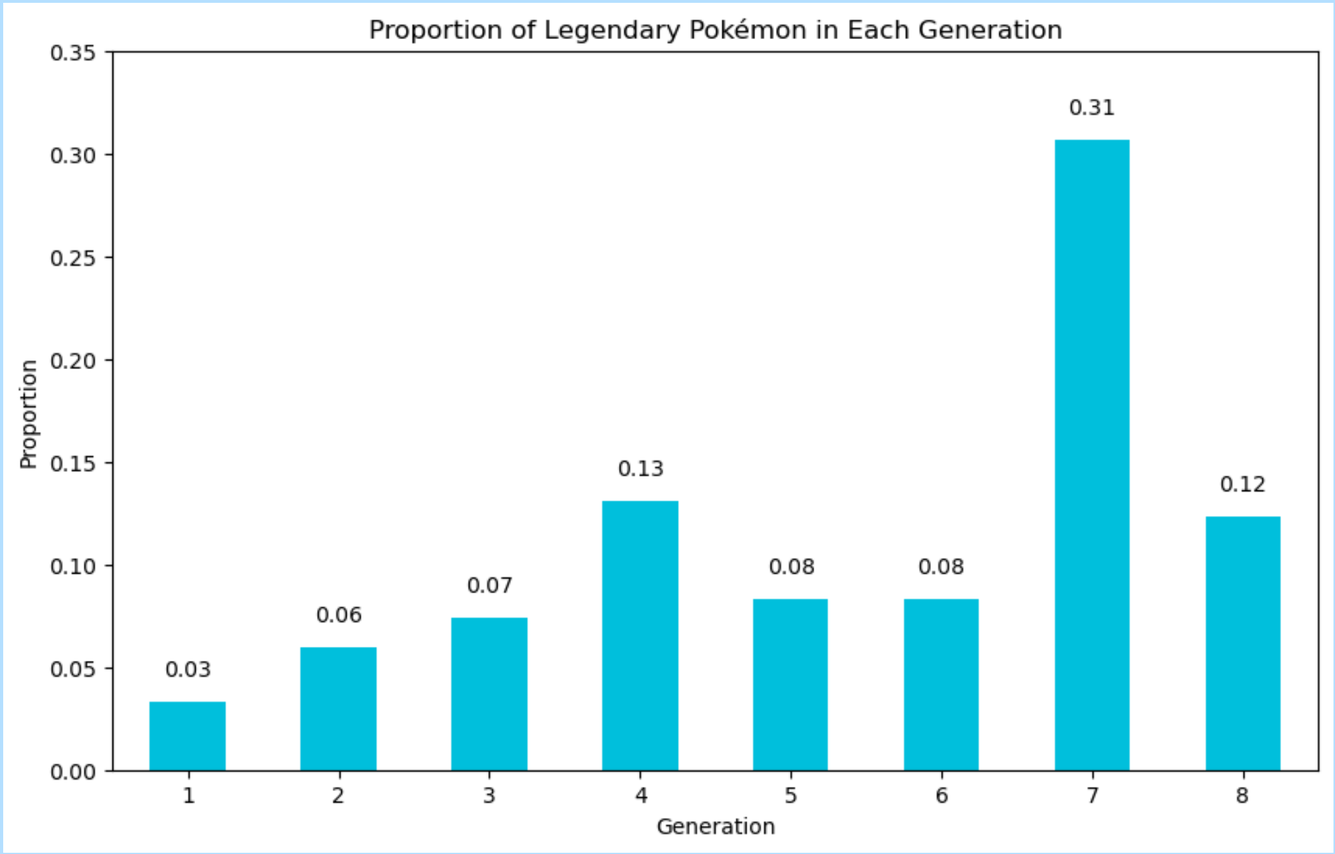
Since the p-value (2.28e-06) is much smaller than α = 0.05, we reject the null hypothesis. We conclude that there is a significant difference in the distribution of Pokémon types between Generation 1 and Generation 2. This suggests that the proportions of Pokémon types differ significantly between the two generations.

There is a relationship between the Pokémon types and the generation they belong to.

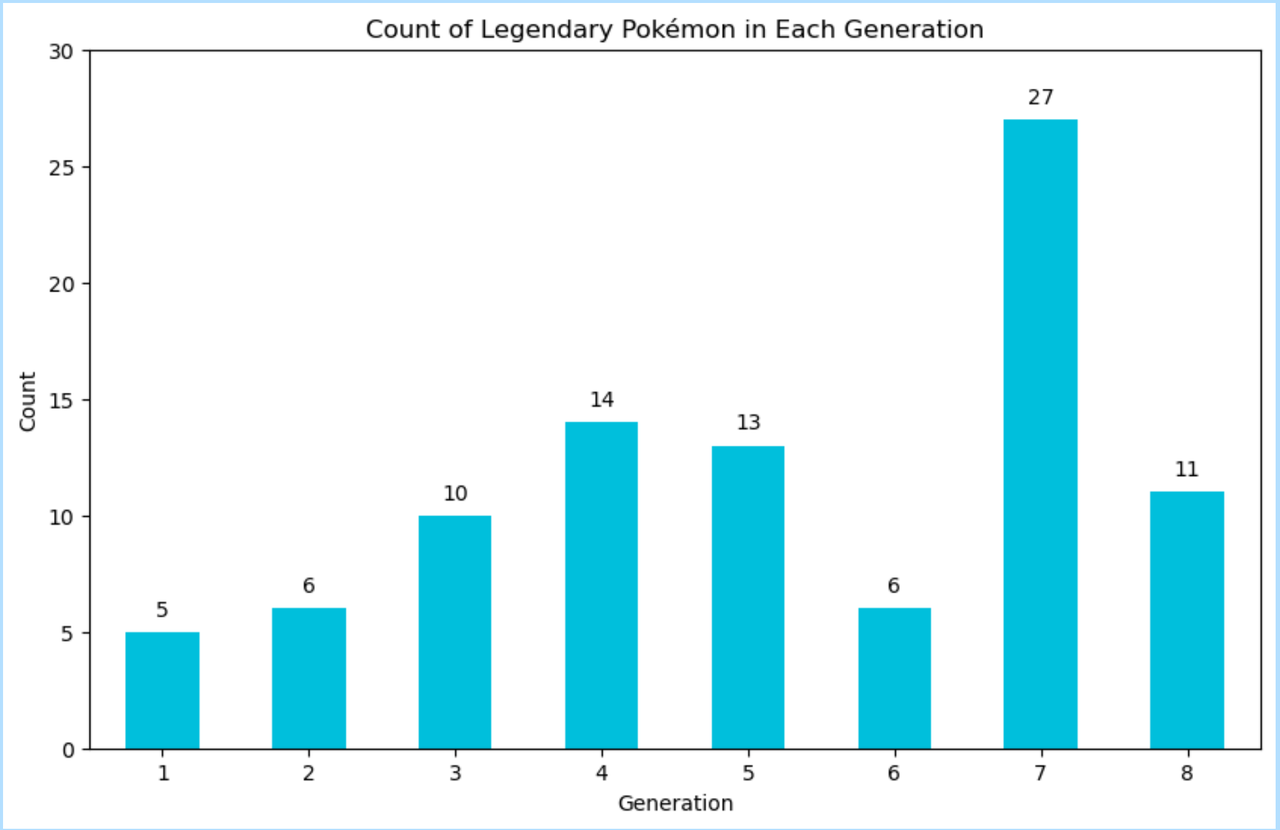
LEGENDARY POKÉMON PROPORTION



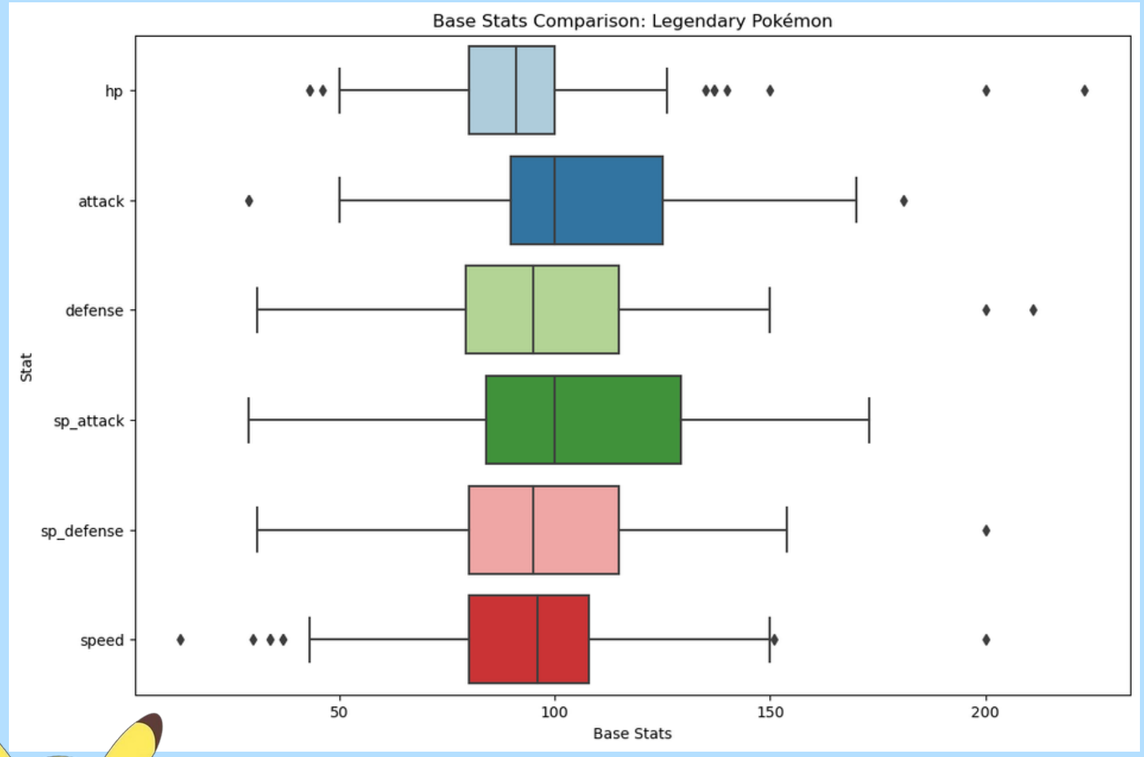
LEGENDARY POKÉMON PROPORTION BY GENERATION

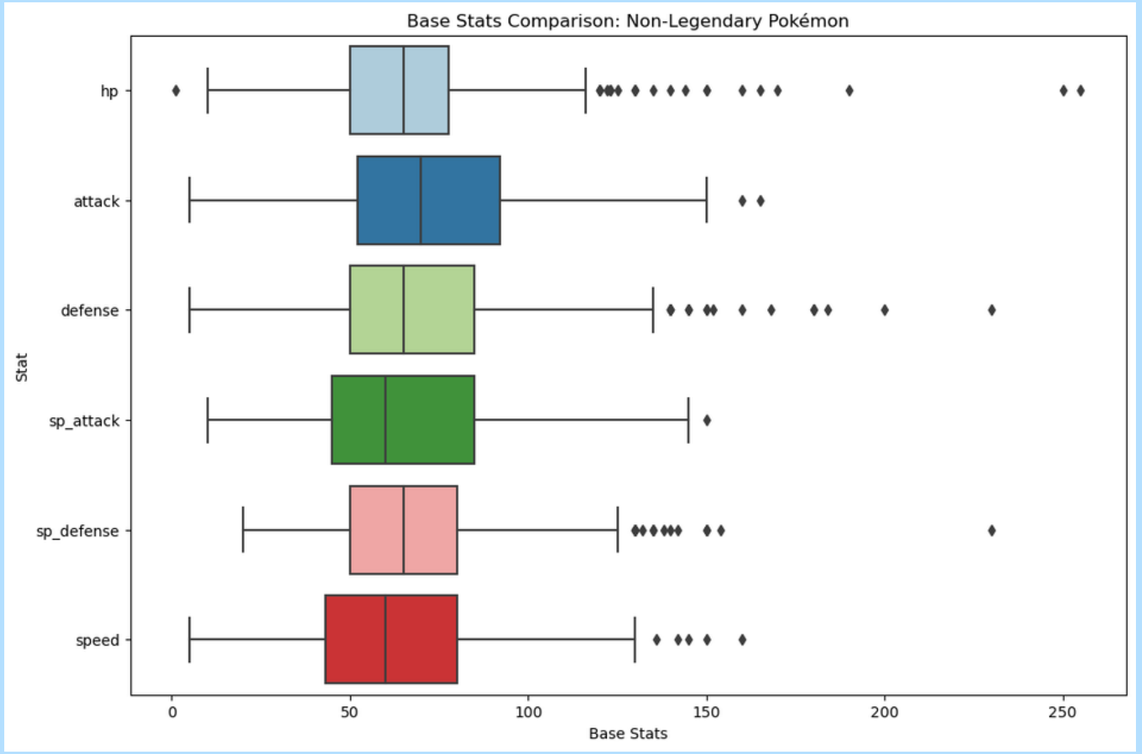


LEGENDARY POKÉMON COUNT BY GENERATION



BASE STATS LEGENDARY VS NON-LEGENDARY





LEGENDARY VS NON-LEGENDARY

Null Hypothesis (H0): There is no significant difference between the means of the attack stat for Legendary and Non-Legendary Pokémon

Alternative Hypothesis (Ha): There is a significant difference between the means of the attack stat for Legendary and Non-Legendary Pokémon

H0: $\mu\_legendary\_attack = \mu\_non\_legendary\_attack$
Ha: $\mu\_legendary\_attack \neq \mu\_non\_legendary\_attack$

```
ANOVA - F-statistic: 66.43527574273688 p-value: 9.871399314347772e-41
```

As the P-Value is extremely low(9.871399314347772e-41),we reject the null hypothesis and conclude that there is a significant difference in the mean base stats between Legendary and Non-Legendary Pokémon

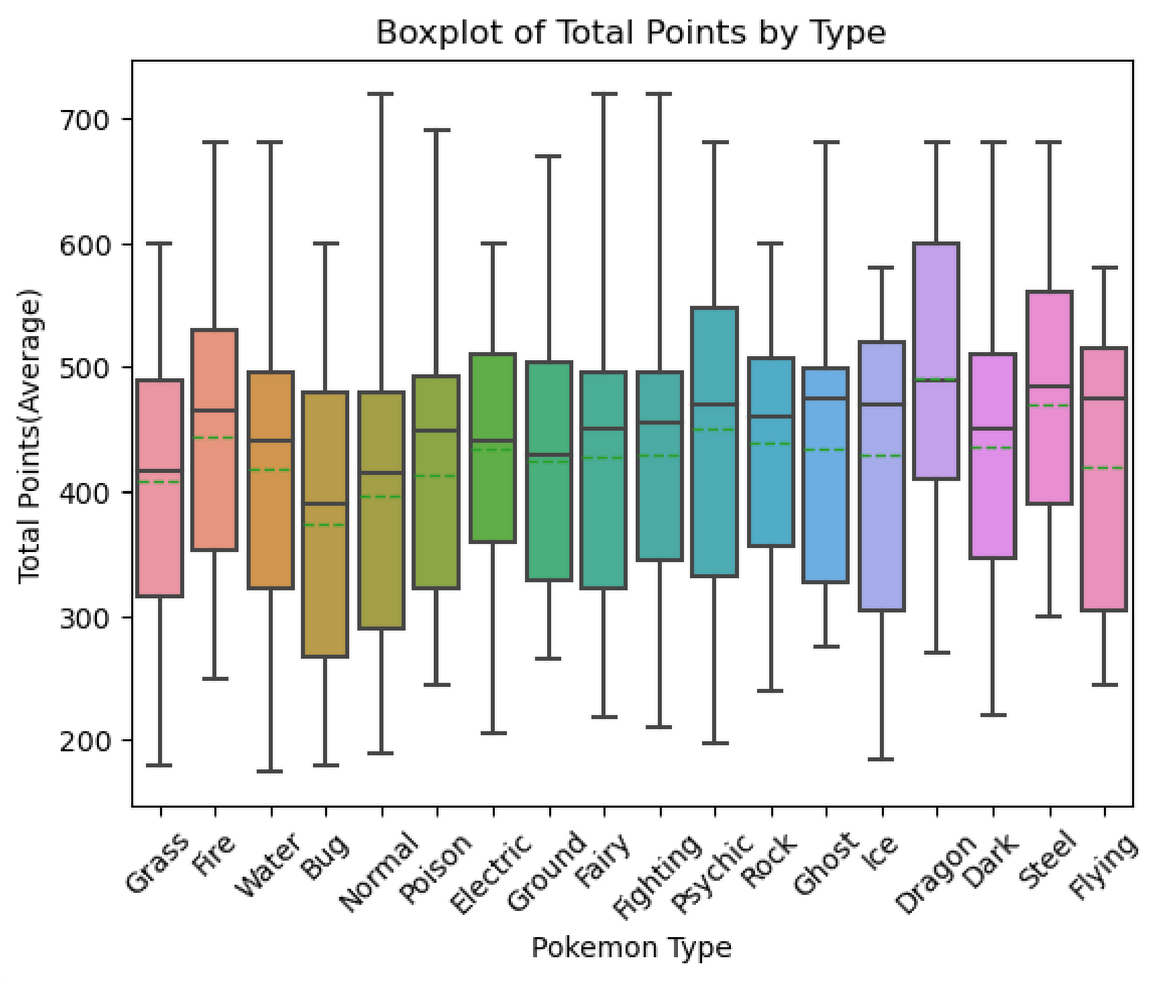
LEGENDARY VS NON-LEGENDARY TYPES DISTRIBUTION

Null Hypothesis (H0): There is no significant difference in the distribution of Pokémon types between Legendary and non-Legendary Pokémon.
Alternative Hypothesis (HA): There is a significant difference in the distribution of Pokémon types between Legendary and non-Legendary Pokémon.

```
Chi-square statistic: 40.675986050986054, p-value: 0.19998986685639075
```

Since the p-value (0.19999) is greater than α = 0.05, we fail to reject the null hypothesis. We do not have sufficient evidence to conclude that there is a significant difference in the distribution of Pokémon types between Legendary and non-Legendary Pokémon. This suggests that the proportions of Pokémon types among Legendary and non-Legendary Pokémon are similar

MOST POWERFUL POKEMON BY TYPE



GRASS TYPE HP

Null Hypothesis (H0): The mean HP of Grass-type Pokémon is equal to the overall mean HP of all Pokémon.
Alternative Hypothesis (HA): The mean HP of Grass-type Pokémon is not equal to the overall mean HP of all Pokémon.

H0: $\mu_{grass} = \mu_{all}$
HA: $\mu_{grass} \neq \mu_{all}$

```
t-statistic: -1.6659106676139124, p-value: 0.09941264119145597
```

With a p-value of 0.099, we fail to reject the null hypothesis at a significance level of 0.05. This means that we do not have enough evidence to conclude that the mean HP of Grass-type Pokémon is significantly different from the overall mean HP of all Pokémon.

GRASS TYPE VS FIRE TYPE HP

Null Hypothesis (H0): The mean HP of Grass-type Pokémon is equal to the mean HP of Fire-type Pokémon.
Alternative Hypothesis (HA): The mean HP of Grass-type Pokémon is not equal to the mean HP of Fire-type Pokémon.

H0: $\mu_{grass} = \mu_{fire}$
HA: $\mu_{grass} \neq \mu_{fire}$

```
t-statistic: -0.894325473458671, p-value: 0.3726610980626749
```

We fail to reject the null hypothesis. This means that we do not have enough evidence to conclude that there is a significant difference in the mean HP between Grass-type and Fire-type Pokémon

MEAN ATTACK ACROSS POKEMON TYPES

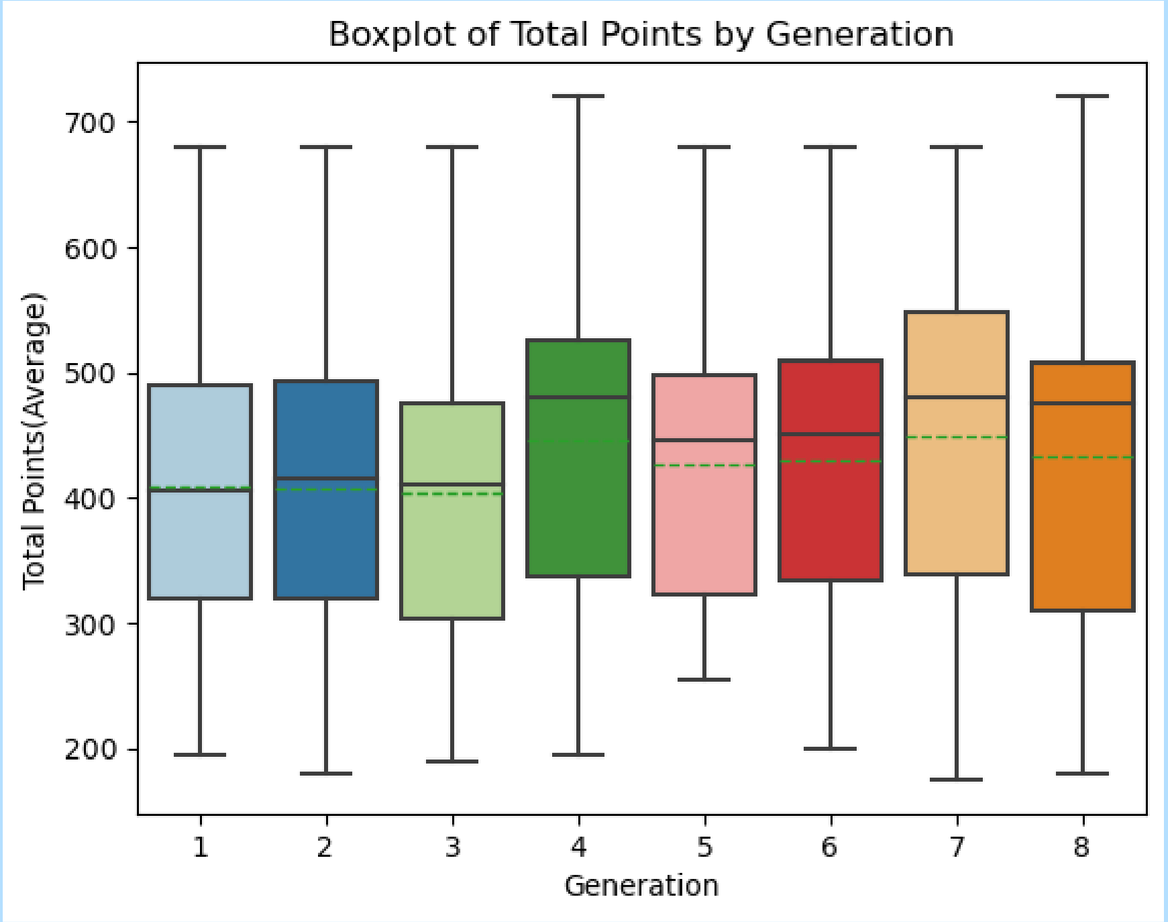
Null Hypothesis (H0): There is no significant difference in the mean Attack stat across different Pokémon types.
Alternative Hypothesis (Ha): There is a significant difference in the mean Attack stat across different Pokémon types.

H0: $\mu\_types = \mu$
HA: $\mu\_types \neq \mu$

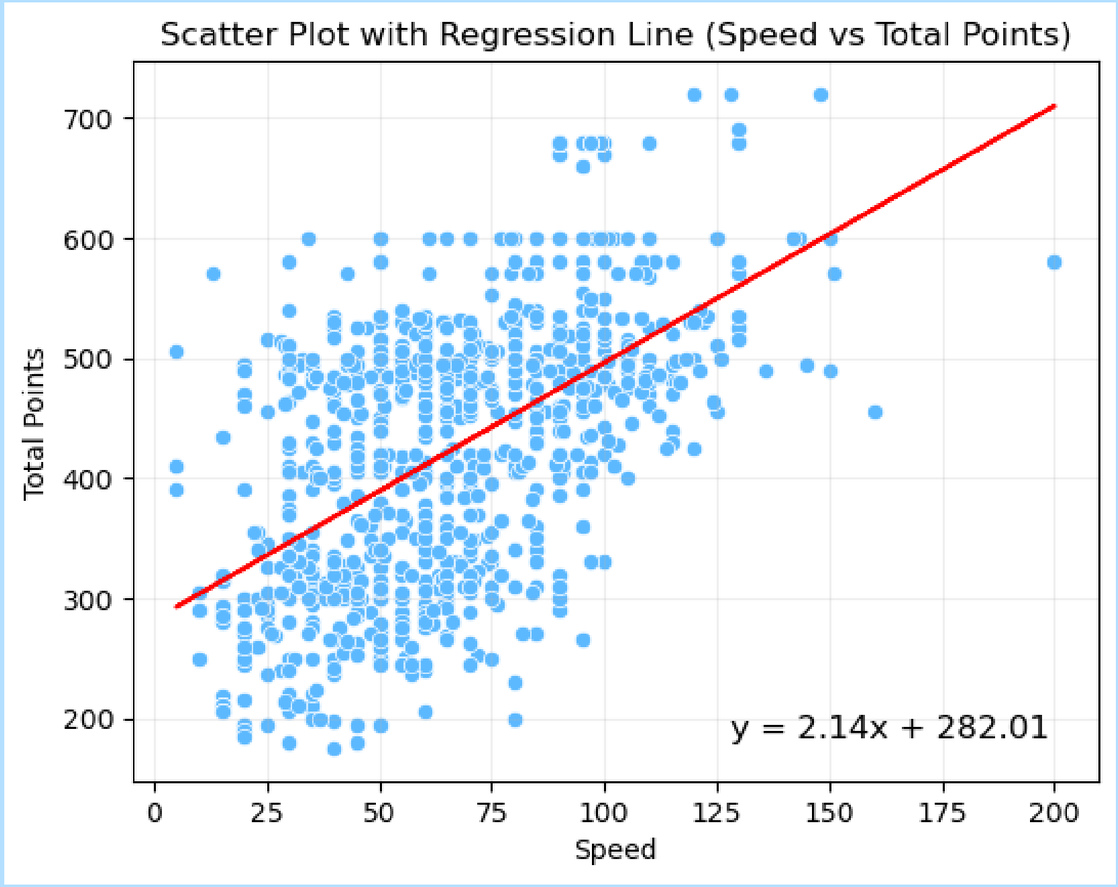
```
F-statistic: 6.608051717439042, p-value: 5.21285145439182 4e-15
```

Since the p-value (5.21e-15) is much smaller than the typical significance level ($\alpha = 0.05$), we reject the null hypothesis. We conclude that there is a significant difference in the Attack stat across different Pokémon types.

POKÉMON POWER BY GENERATION



CORRELATION BETWEEN POKÉMON SPEED AND THEIR POWER



POKÉMON WITH HIGHEST STATS PER GENERATION

SPECIAL ATTACK

Name: Mewtwo
154

Name: Espeon
130

Name: Kyogre
150

Name: Dialga
150

Name: Reshiram
150

Name: Aegislash Blade Forme
150

Name: Xurkitree
173

Name: Cursola
145

POKÉMON WITH HIGHEST STATS PER GENERATION

ATTACK

Name: Dragonite
134

Name: Tyranitar
134

Name: Slaking
160

Name: Rampardos
165

Name: Zekrom
150

Name: Aegislash Blade Forme
150

Name: Kartana
181

Name: Zacian Crowned Sword
170

POKÉMON WITH HIGHEST STATS PER GENERATION

DEFENSE

Name: Cloyster
180

Name: Shuckle
230

Name: Regirock
200

Name: Bastiodon
168

Name: Cofagrigus
145

Name: Avalugg
184

Name: Stakataka
211

Name: Runerigus
145

POKÉMON WITH HIGHEST STATS PER GENERATION

SPECIAL DEFENSE

Name: Articuno
125

Name: Shuckle
230

Name: Regice
200

Name: Probopass
150

Name: Cryogonal
135

Name: Florges
154

Name: Toxapex
142

Name: Zamazenta Crowned Shield
145

POKÉMON WITH HIGHEST STATS PER GENERATION

SPEED

Name: Electrode
150

Name: Crobat
130

Name: Ninjask
160

Name: Weavile
125

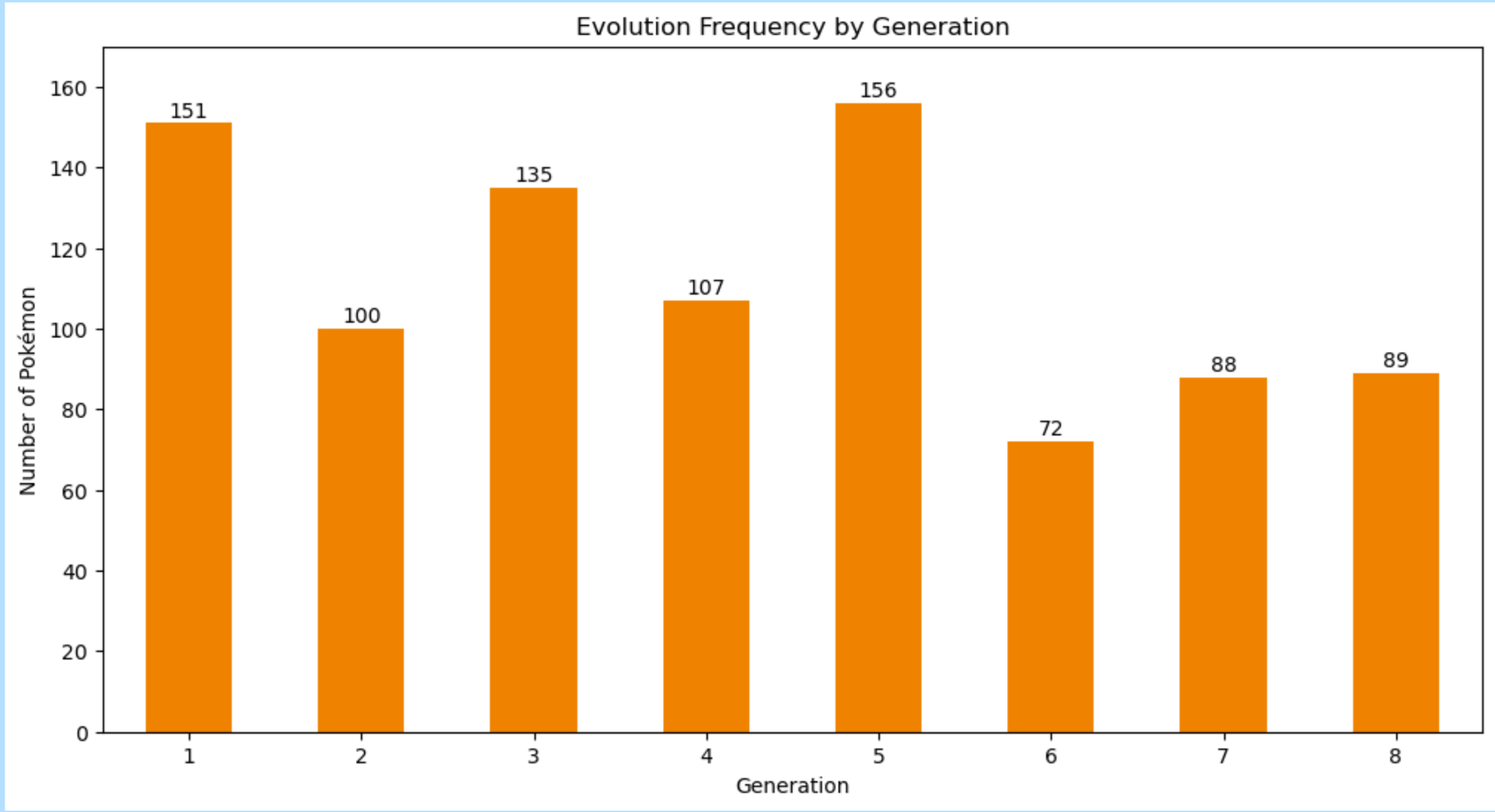
Name: Accelgor
145

Name: Talonflame
126

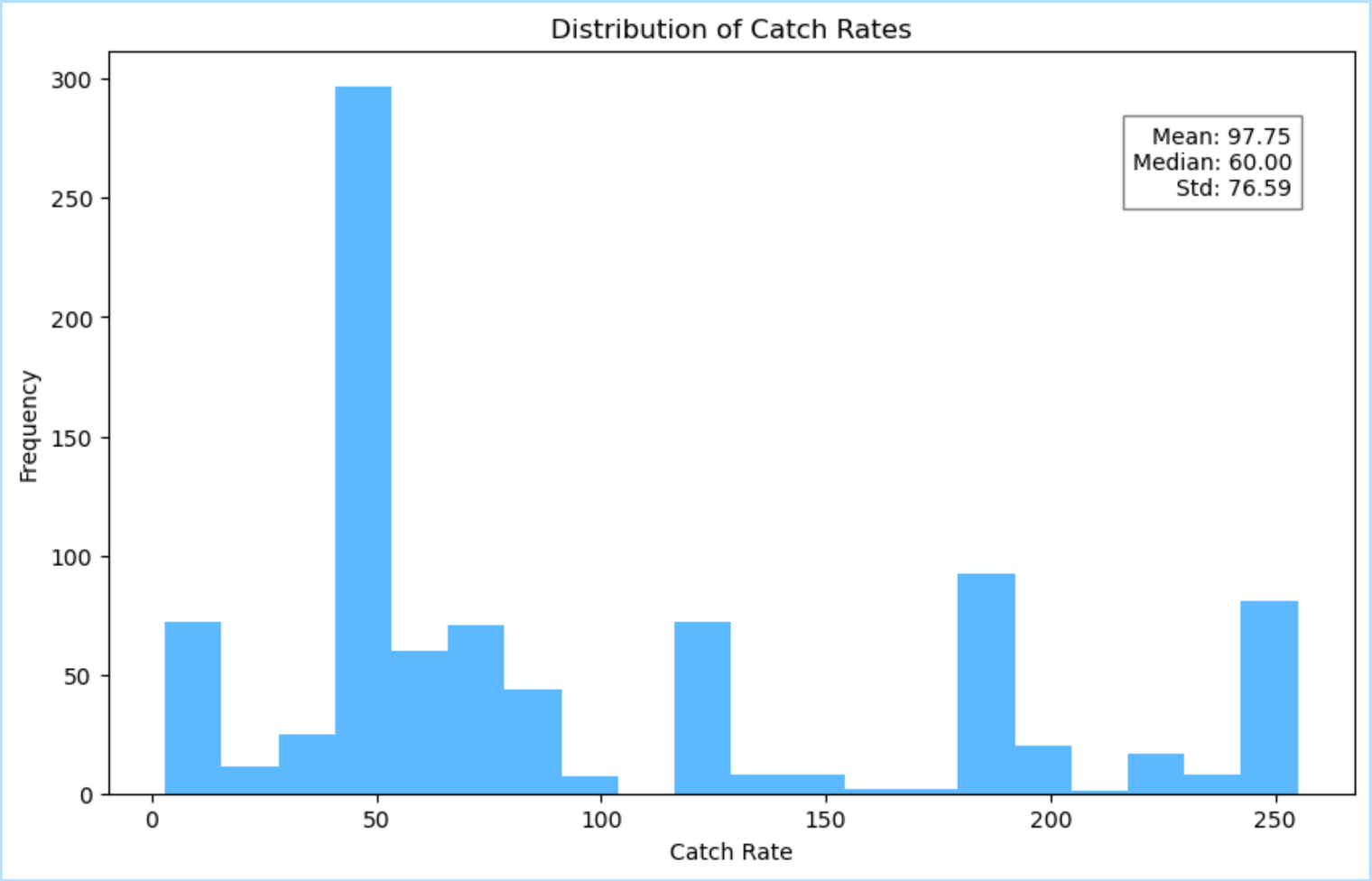
Name: Pheromosa
151

Name: Regieleki
200

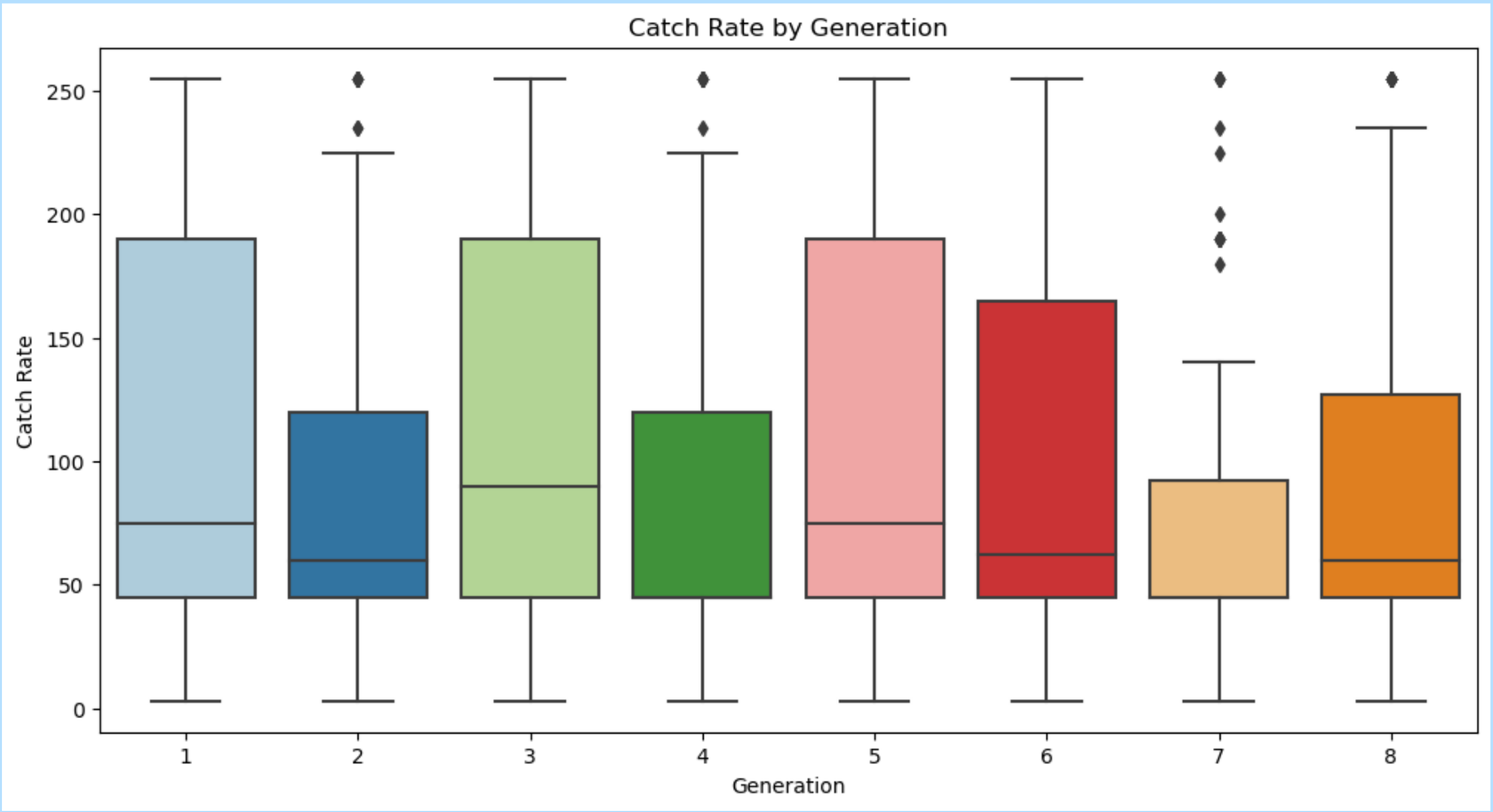
EVOLUTION FREQUENCY BY GENERATION



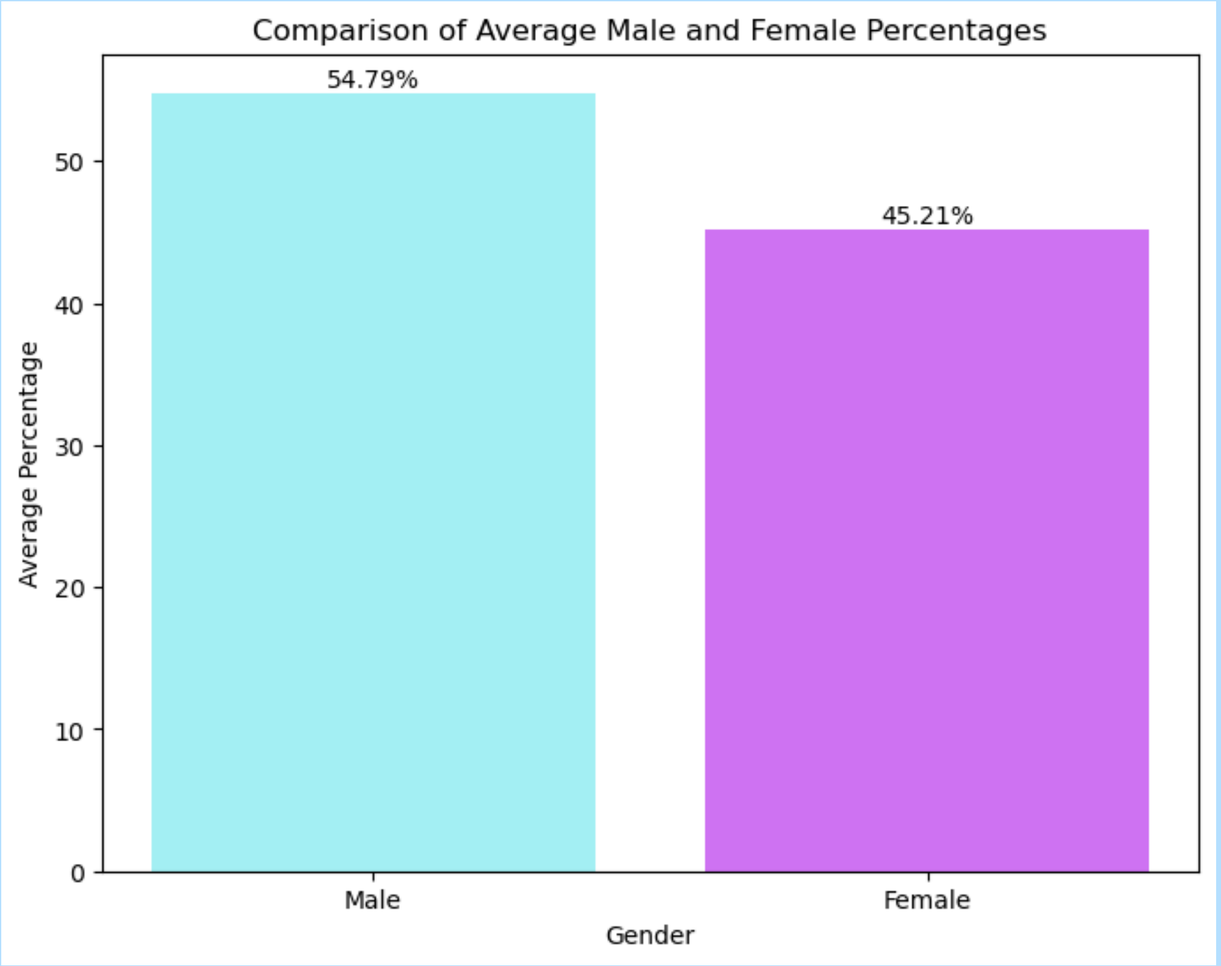
CATCH RATE DISTRIBUTION



CATCH RATE BY GENERATION



GENDER DISTRIBUTION



05

CONCLUSIONS

1. Pokémon Types and Power: Certain types of Pokémon tend to have higher average total points, indicating greater power. Dragon, Psychic, and Steel types emerged as among the most powerful based on average total points.

2. Generational Strength: Contrary to expectations, the analysis revealed that the seventh generation hosts the most powerful Pokémon, followed by the fourth and eighth generations. This suggests that higher generation numbers don't always correlate with increased power.

3. Correlation Analysis: A moderate positive linear correlation exists between Pokémon speed and their total points, suggesting that faster Pokémon tend to have higher total points, indicating higher overall power.

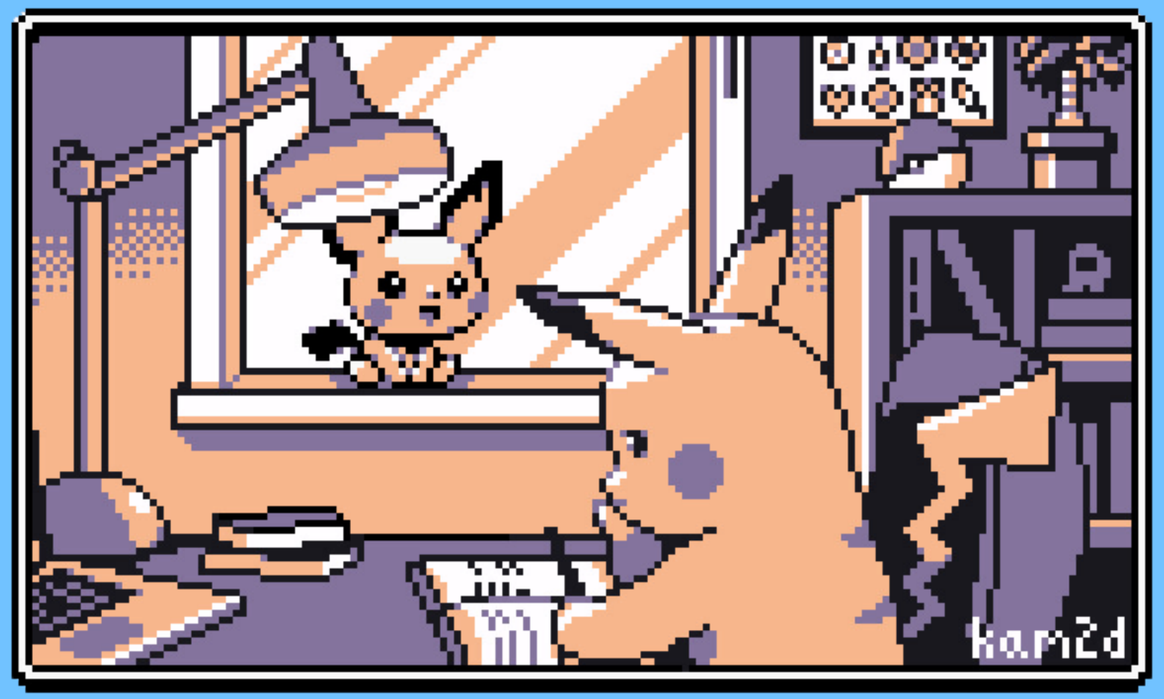
4. Comparing Legendary and Non-Legendary Pokémon: Legendary Pokémon generally exhibit higher average base stats compared to non-legendary Pokémon, though there is overlap in the ranges. Statistical tests confirmed significant differences in various base stats between the two groups.

5. Type Distribution: Exploration of the distribution of Pokémon types across generations highlighted the most common and rare type combinations. Normal and Water types emerged as the most common, while certain combinations like Fire-Dark and Dragon-Ice were among the rarest.

7. Importance of Data Processing: Thorough data gathering, cleaning, and processing were essential in extracting meaningful insights from complex datasets, emphasizing their crucial role in data analysis.

Overall, this analysis provides valuable insights for Pokémon enthusiasts, researchers, and game developers, shedding light on the diverse characteristics and dynamics within the Pokémon world.

THANK YOU!



STUDYING FOR THE FINAL EXAM