“Comprehensive Data Analysis and Visualization of the Pokémon Dataset”

A report generated through the analysis of Pokémon data, showcasing key insights and trends across various attributes such as types, stats, and generations, complemented by graphical visualizations created using Python libraries like Matplotlib and Pandas.

In
Third Semester
By

USN	Name of the Student
1MS23IS097	Prince Khatri
1MS23IS091	Prakhar Srivastava

Under the guidance of
Shivananda S
Professor
Dept. of ISE, RIT

RAMAIAH Institute of Technology

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY
(AUTONOMOUS INSTITUTE AFFILIATED TO VTU)

M. S. RAMAIAH NAGAR, M. S. R. I. T. POST, BANGALORE – 560054

2024-2025

RAMAIAH INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled “Comprehensive Data Analysis and Visualization of the Pokémon Dataset” is a bonafide work carried out by Prince Khatri and Prakhar Srivastava bearing USN: 1MS23IS097 and 1MS23IS091 submitter report for Project Based Learning in partial fulfillment of requirements of Continuous Internal Evaluation of the course “Numerical Analysis with Python (ISAEC393) of Third Semester B.E. It is certified that all corrections/suggestions indicated for internal assessment has been incorporated in the report. The project has been approved as it satisfies the academic requirements in respect of project work prescribed by the above said course.

____

Signature of the Guide
Mr. Shivananda S
Professor
Dept. of ISE, RIT
Bangalore 54

Acknowledgement

We would like to take this opportunity to thank everyone who whole heartedly helped us in the project. We are very much thankful to our principal Dr. N V R Naidu for providing us the facilities to carry out our project work. We are grateful to our Head of Department, Dr. Anita Kanavalli, Professor and Head, Department of Information Science Engineering for providing us with this opportunity. We also wish to express our gratitude to our mini-project guide, Mr. Shivananda S, Professor, ISE, who was abundantly helpful and offered invaluable assistance, exemplary guidance and constant encouragement. We would also like to thank all teaching and non-teaching staff of our department for their kind cooperation in the successful completion of this project.

Contents

Introduction

The Pokémon dataset provides detailed information on 800 unique Pokémon, including their attributes, elemental types, base stats, generation, and legendary status. This comprehensive dataset serves as an excellent foundation for exploring various trends and patterns within the Pokémon universe.

Objectives

The primary goal of this project is to perform an in-depth analysis of the dataset to uncover meaningful trends, correlations, and insights. By leveraging statistical methods and data visualization techniques, the analysis will focus on exploring relationships between key attributes, such as the impact of type combinations on stats, the distribution of stats across generations, and the distinguishing features of legendary Pokémon.

The report will include:

1. **Data Exploration and Cleaning**:
   - An initial overview of the dataset to understand its structure and identify missing or inconsistent data.
   - Cleaning and preprocessing steps to ensure the data is ready for analysis.
2. **Descriptive Analysis**:
   - Summary statistics to understand the central tendencies and variability of key attributes like HP, Attack, Defense, and Speed.
   - Distribution analysis of Pokémon types and generations.
3. **Visualization of Insights**:
   - Graphical representations using Python libraries like Matplotlib and Seaborn to visualize trends, such as:
     - The frequency distribution of Pokémon types and their combinations.

        - The evolution of stats across generations.
        - Comparison of base stats between legendary and non-legendary Pokémon.
    - Heatmaps and scatter plots to explore correlations between attributes.

4. **Advanced Analysis**:
    - Identifying outliers or unique Pokémon based on their stats.
    - Examining the role of dual types in boosting performance.
    - Clustering Pokémon based on their attributes for deeper insights.
5. **Reproducibility**:
    - A detailed step-by-step guide for replicating the analysis, including Python commands and explanations of the logic behind each step.
    - Use of libraries such as Pandas for data manipulation, Matplotlib and Seaborn for visualization, and NumPy for statistical operations.

This project not only aims to uncover fascinating insights about Pokémon but also serves as a learning resource for data analysis enthusiasts. By following this report, readers will gain a deeper understanding of the Pokémon dataset and practical experience in using Python for data analysis and visualization.

Dataset Overview

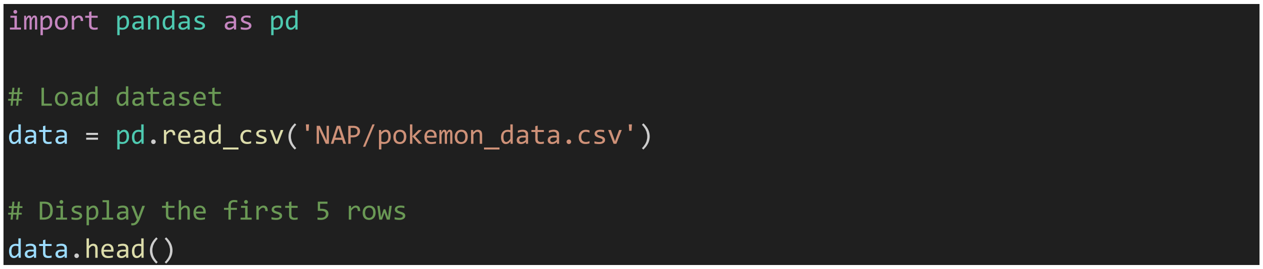
The dataset includes the following fields:

- **ID**: A unique identifier assigned to each Pokémon.
- **Name**: The name of the Pokémon.
- **Type 1**: The primary elemental type of the Pokémon (e.g., Fire, Water, Grass).
- **Type 2**: The secondary elemental type of the Pokémon, if applicable.
- **HP**: Health Points, representing the Pokémon's ability to endure damage.
- **Attack**: The Pokémon's physical attack strength.
- **Defense**: The Pokémon's physical defense strength.
- **Sp. Atk**: Special attack strength, used for non-physical moves.
- **Sp. Def**: Special defense strength, used to resist non-physical moves.
- **Speed**: The Pokémon's speed, determining the order of moves in battle.
- **Generation**: The Pokémon's debut generation in the series.
- **Legendary**: A Boolean field indicating whether the Pokémon is classified as legendary.

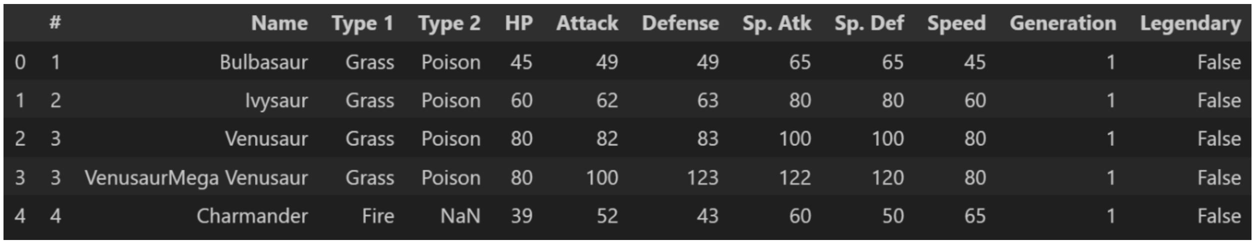
This structured dataset provides comprehensive information for analysis and visualization of Pokémon attributes, allowing for in-depth insights into their characteristics and performance.

Step-by-step Analysis

Loading the Dataset

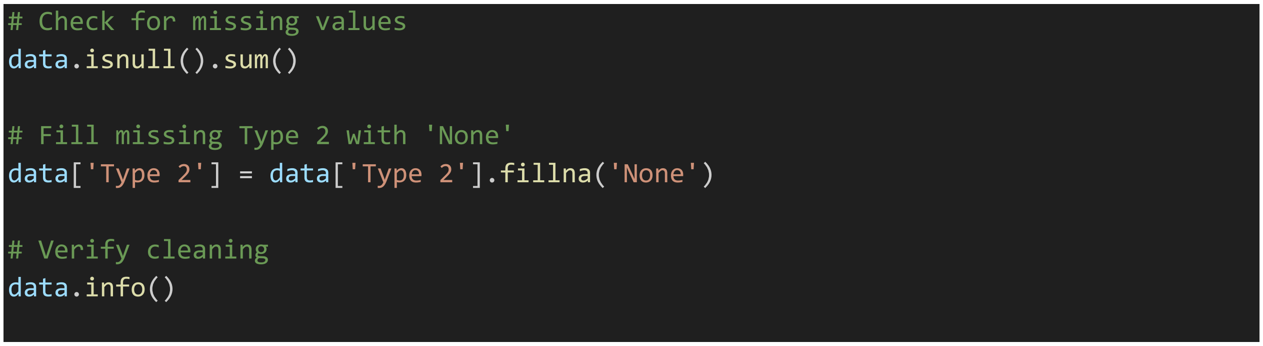


```
import pandas as pd

# Load dataset
data = pd.read_csv('NAP/pokemon_data.csv')

# Display the first 5 rows
data.head()
```



	#	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

Data Cleaning



```
# Check for missing values
data.isnull().sum()

# Fill missing Type 2 with 'None'
data['Type 2'] = data['Type 2'].fillna('None')

# Verify cleaning
data.info()
```

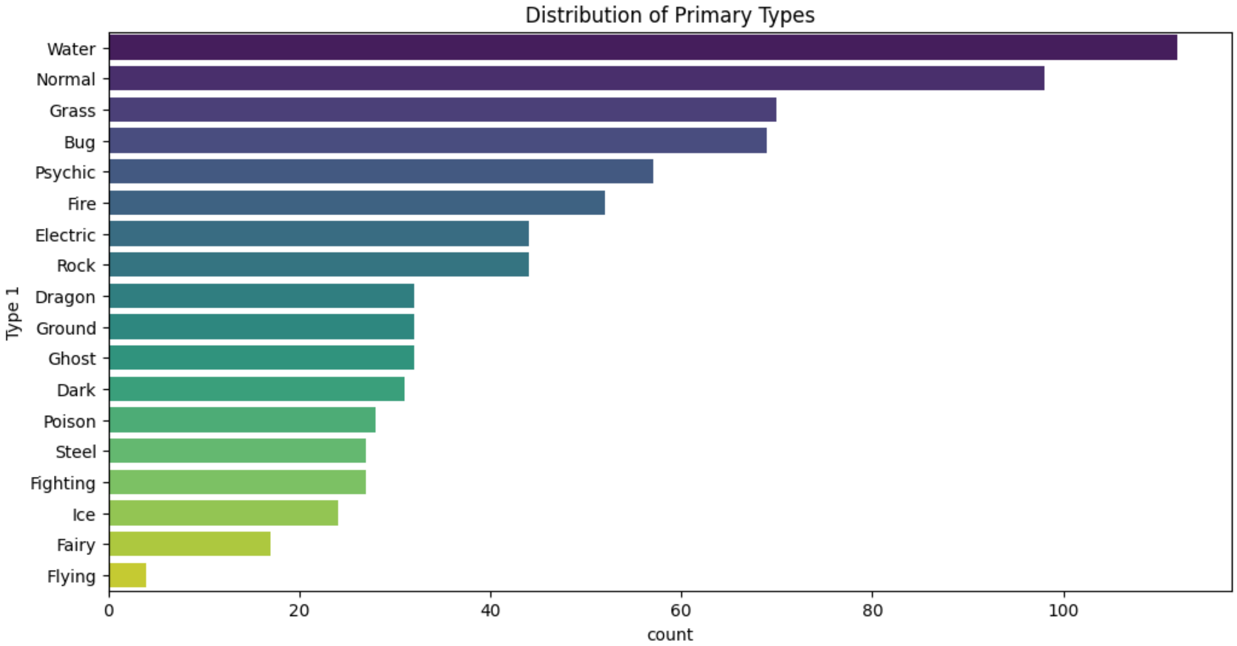
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   #           800 non-null    int64 
 1   Name        800 non-null    object
 2   Type 1      800 non-null    object
 3   Type 2      800 non-null    object
 4   HP          800 non-null    int64 
 5   Attack      800 non-null    int64
 6   Defense     800 non-null    int64
 7   Sp. Atk     800 non-null    int64
 8   Sp. Def     800 non-null    int64
 9   Speed       800 non-null    int64
 10  Generation  800 non-null    int64
 11  Legendary   800 non-null    bool
dtypes: bool(1), int64(8), object(3)
memory usage: 69.7+ KB
```

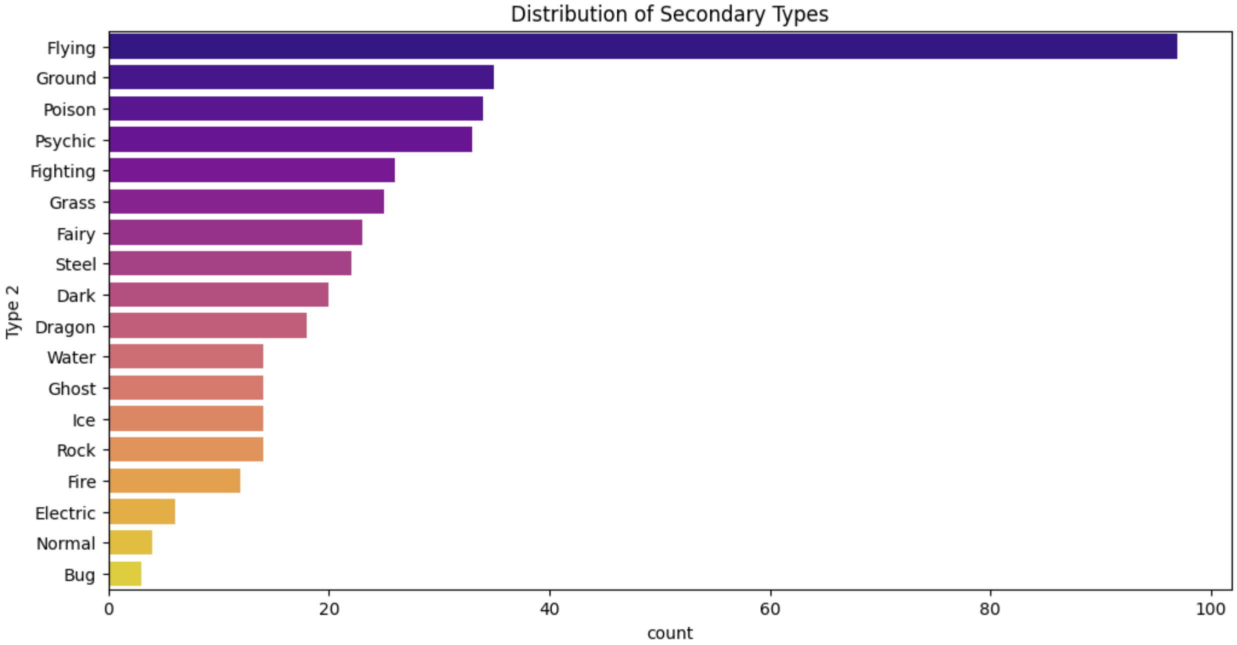
Explanatory Data Analysis

A. Distribution of Pokémon Types

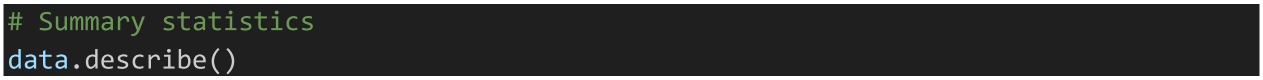
To visualize the frequency of primary and secondary types of Pokémon.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Countplot for Type 1
plt.figure(figsize=(12, 6))
sns.countplot(y='Type 1', data=data, order=data['Type 1'].value_counts().index,
palette='viridis')
plt.title('Distribution of Primary Types')
plt.show()

# Countplot for Type 2
plt.figure(figsize=(12, 6))
sns.countplot(y='Type 2', data=data, order=data['Type 2'].value_counts().index,
palette='plasma')
plt.title('Distribution of Secondary Types')
plt.show()
```

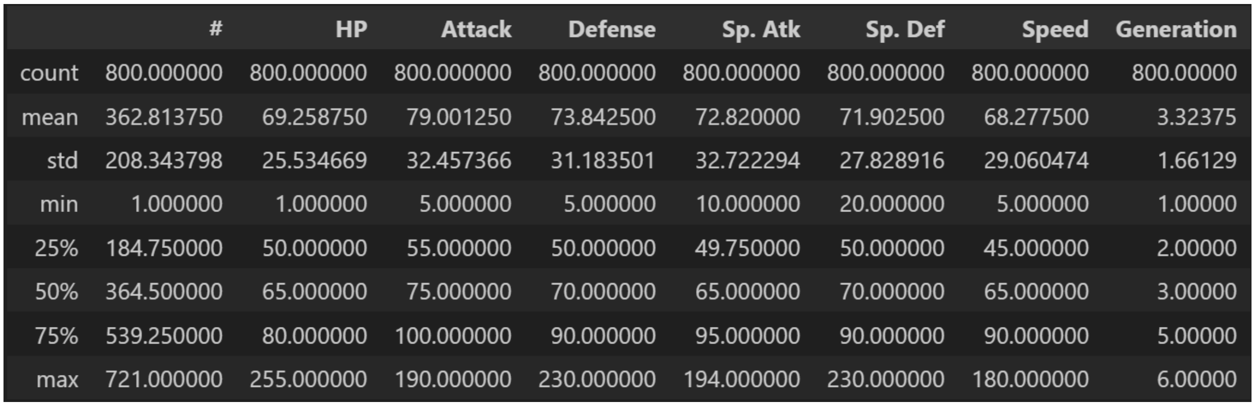

Distribution of Primary Types
Water
Normal
Grass
Bug
Psychic
Fire
Electric
Rock
Dragon
Ground
Ghost
Dark
Poison
Steel
Fighting
Ice
Fairy
Flying
Type 1
0
20
40
60
80
100
count


Distribution of Secondary Types
Flying
Ground
Poison
Psychic
Fighting
Grass
Fairy
Steel
Dark
Dragon
Water
Ghost
Ice
Rock
Fire
Electric
Normal
Bug
Type 2
0
20
40
60
80
100
count

B. Statistical Summary



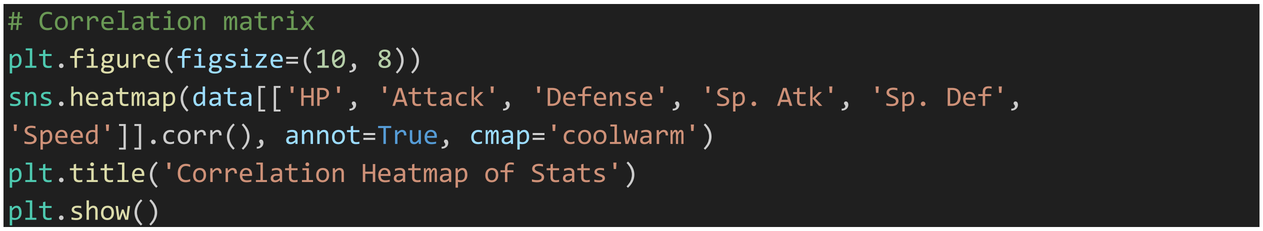
```
# Summary statistics
data.describe()
```



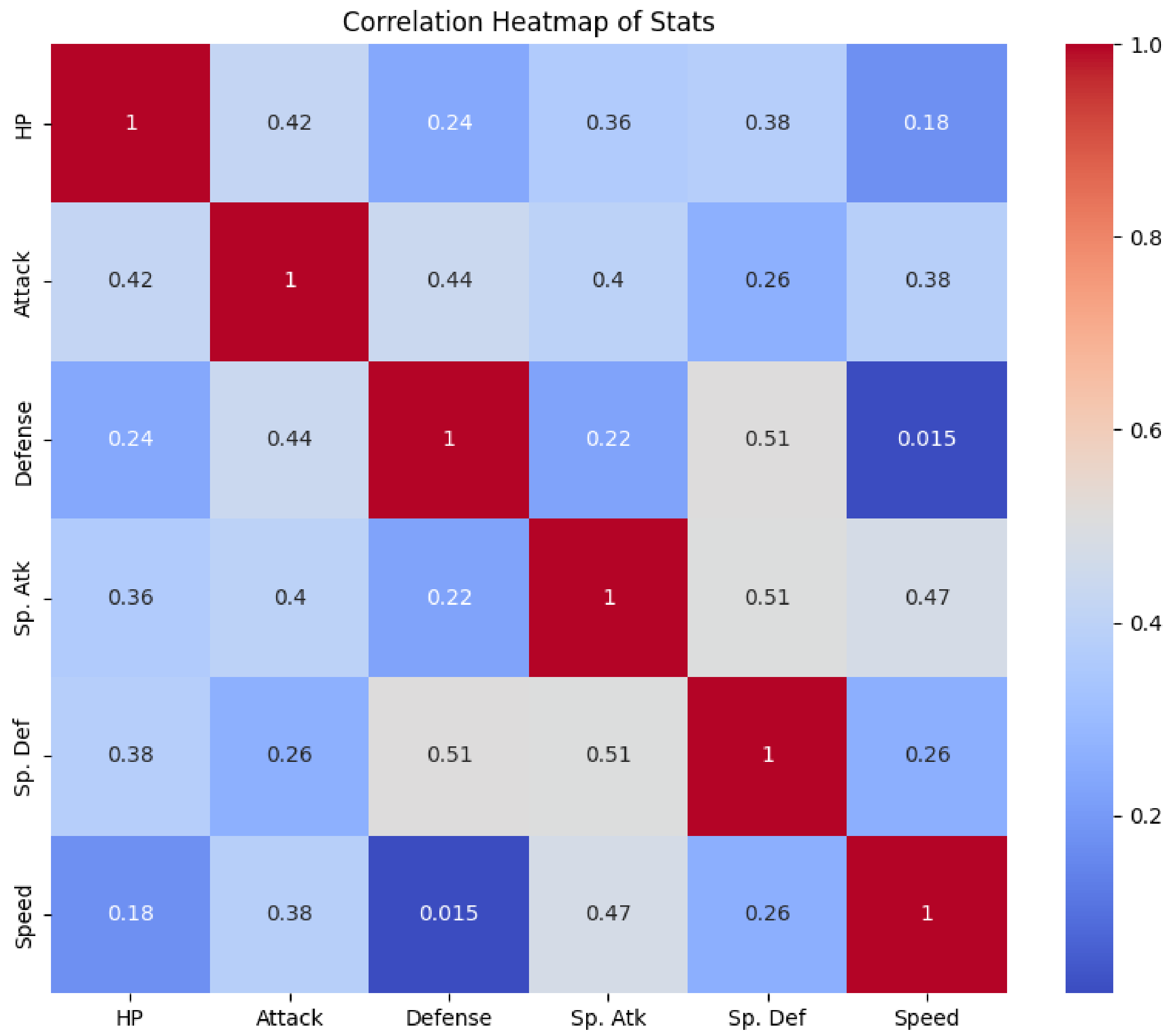
	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
count	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.00000
mean	362.813750	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500	3.32375
std	208.343798	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474	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

C. Correlation Heatmaps

To identify correlation between stats.

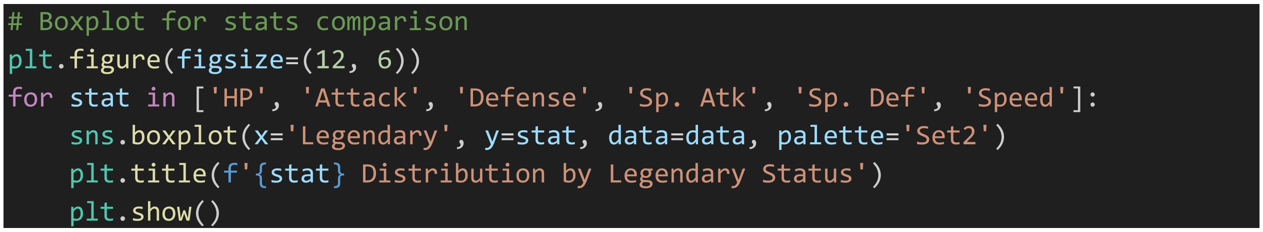


```
# Correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(data[['HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def',
'Speed']].corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Stats')
plt.show()
```

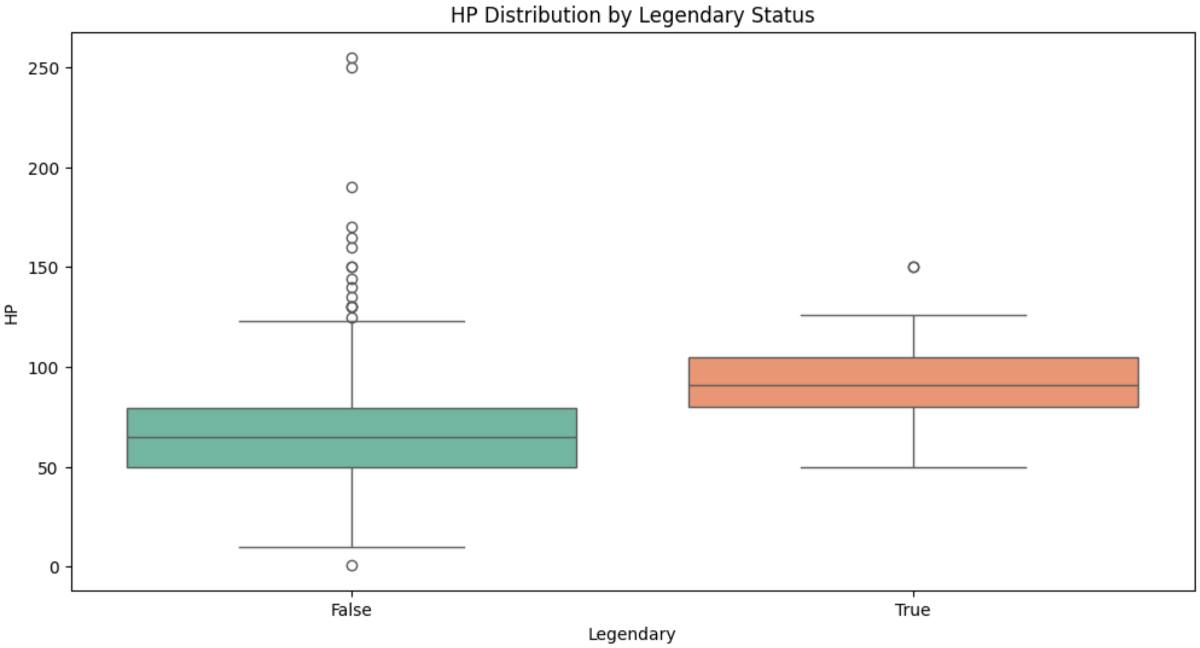


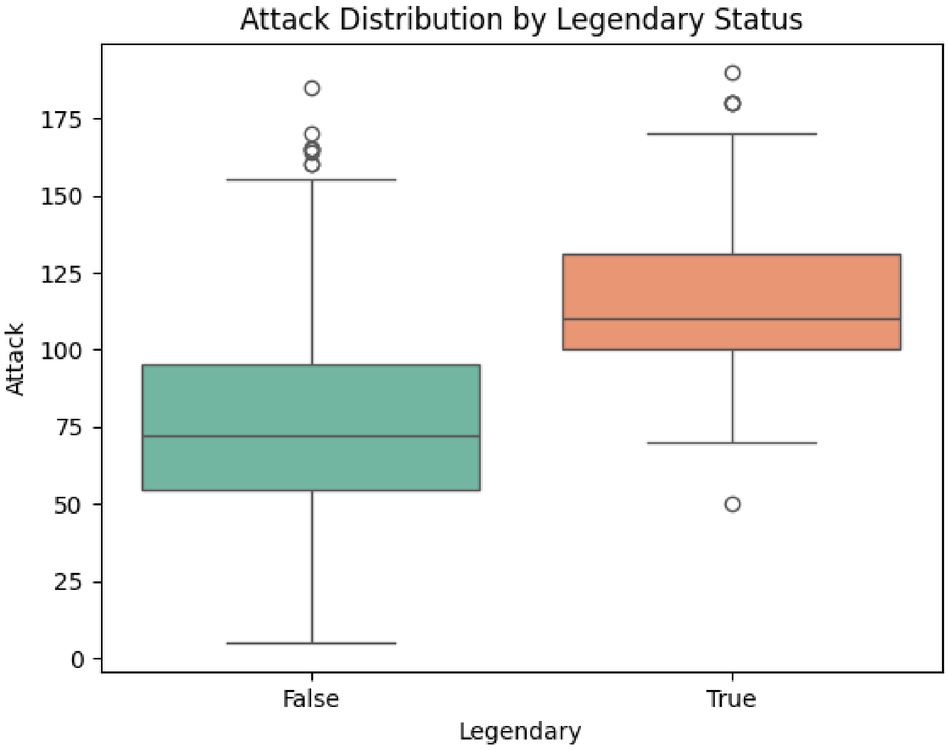
Legendary vs Non-Legendary Comparison

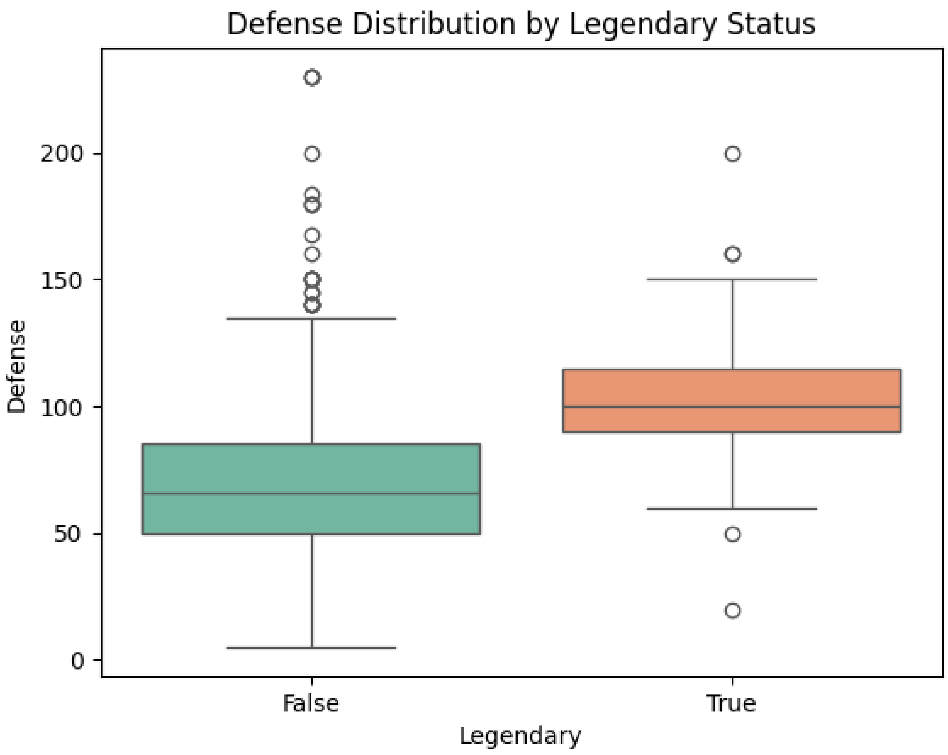
Analyze differences in stats between Legendary and non-Legendary Pokémon.

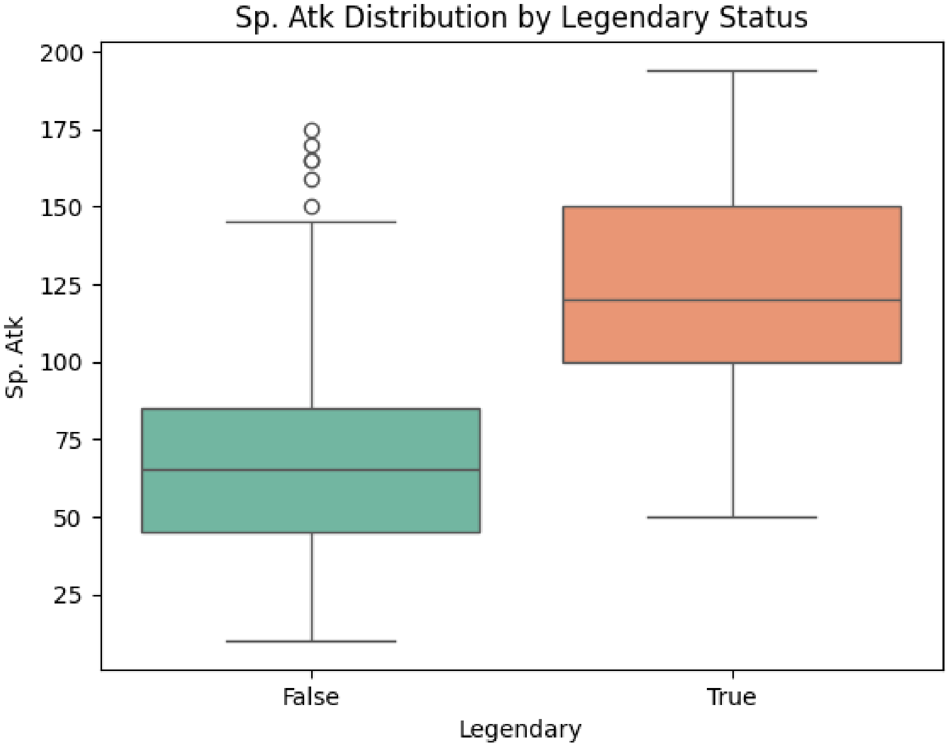


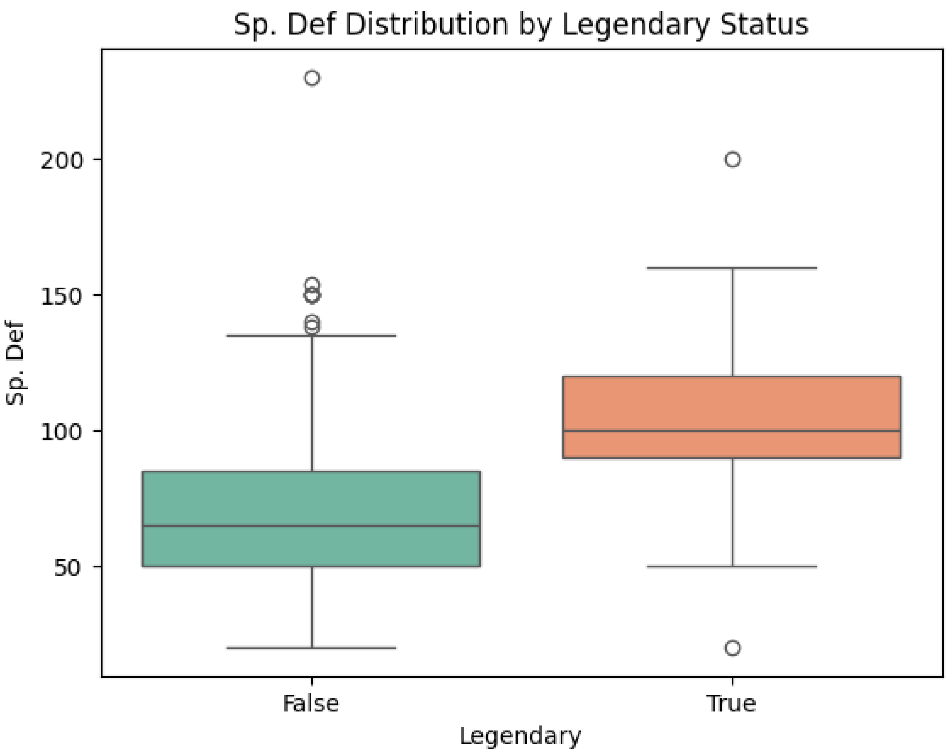
```
# Boxplot for stats comparison
plt.figure(figsize=(12, 6))
for stat in ['HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def', 'Speed']:
    sns.boxplot(x='Legendary', y=stat, data=data, palette='Set2')
    plt.title(f'{stat} Distribution by Legendary Status')
    plt.show()
```


HP Distribution by Legendary Status
HP
250
200
150
100
50
0
False
True
Legendary

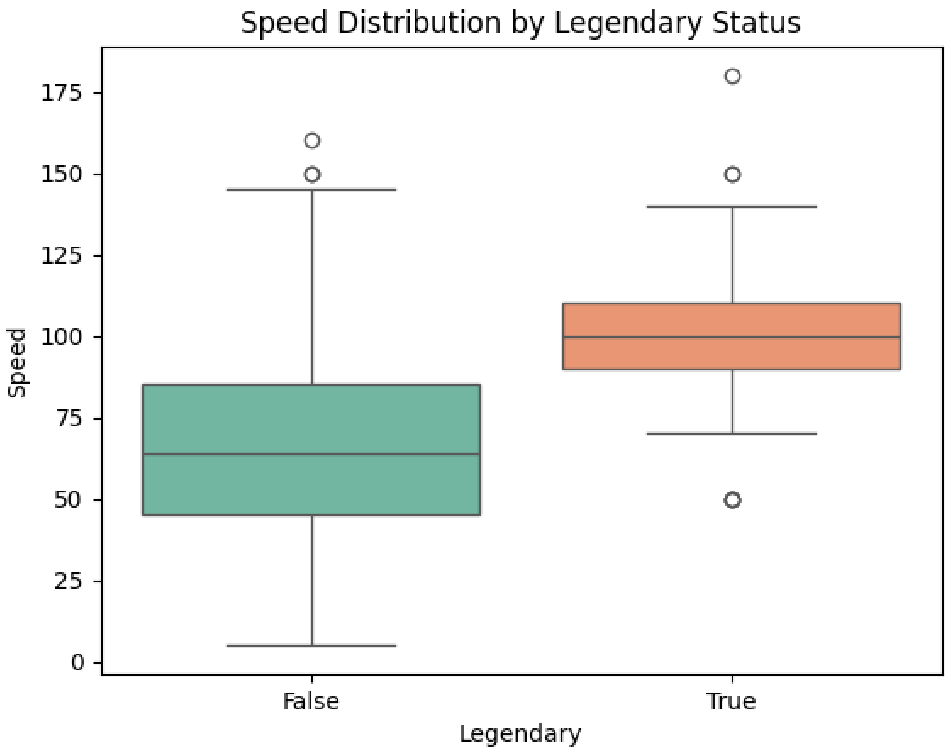

Attack Distribution by Legendary Status
Attack
175
150
125
100
75
50
25
0
False
True
Legendary


Defense Distribution by Legendary Status
Defense
0
50
100
150
200
False
True
Legendary


Sp. Atk Distribution by Legendary Status
Sp. Atk
25
50
75
100
125
150
175
200
False
True
Legendary



Sp. Def Distribution by Legendary Status

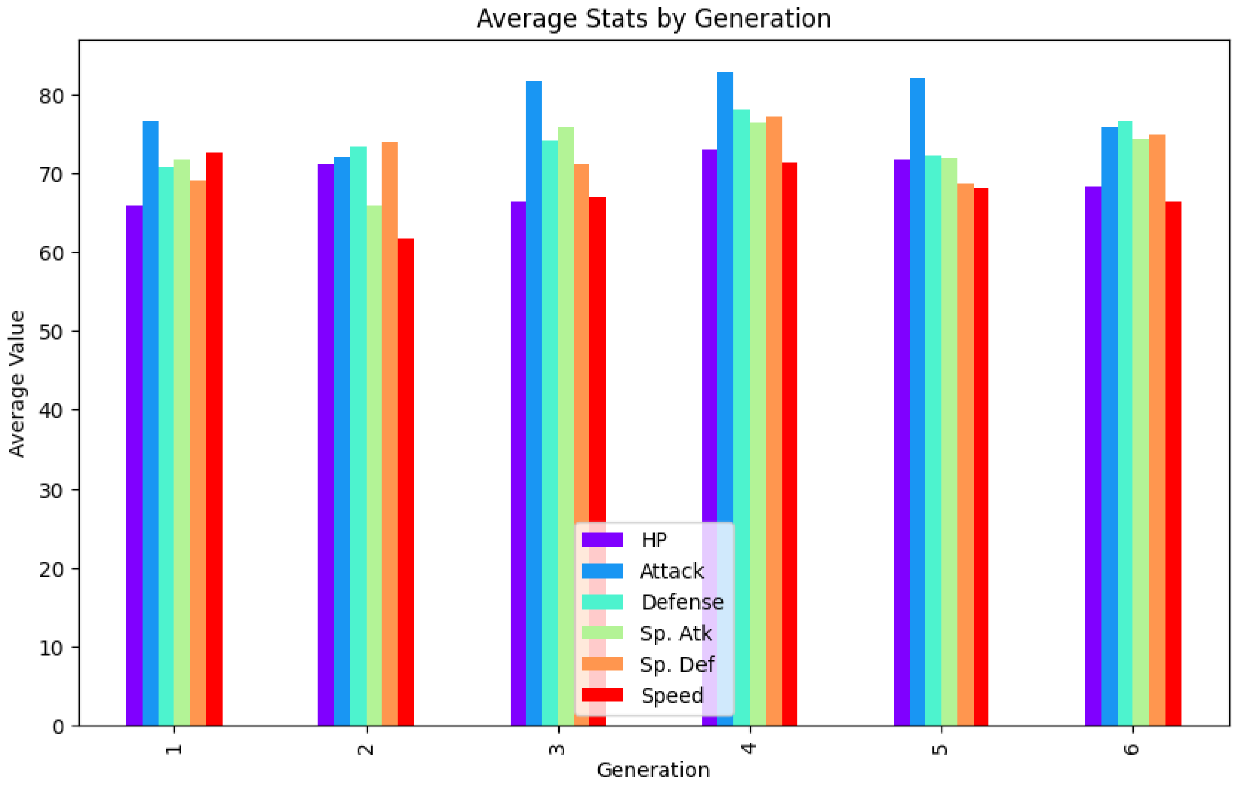


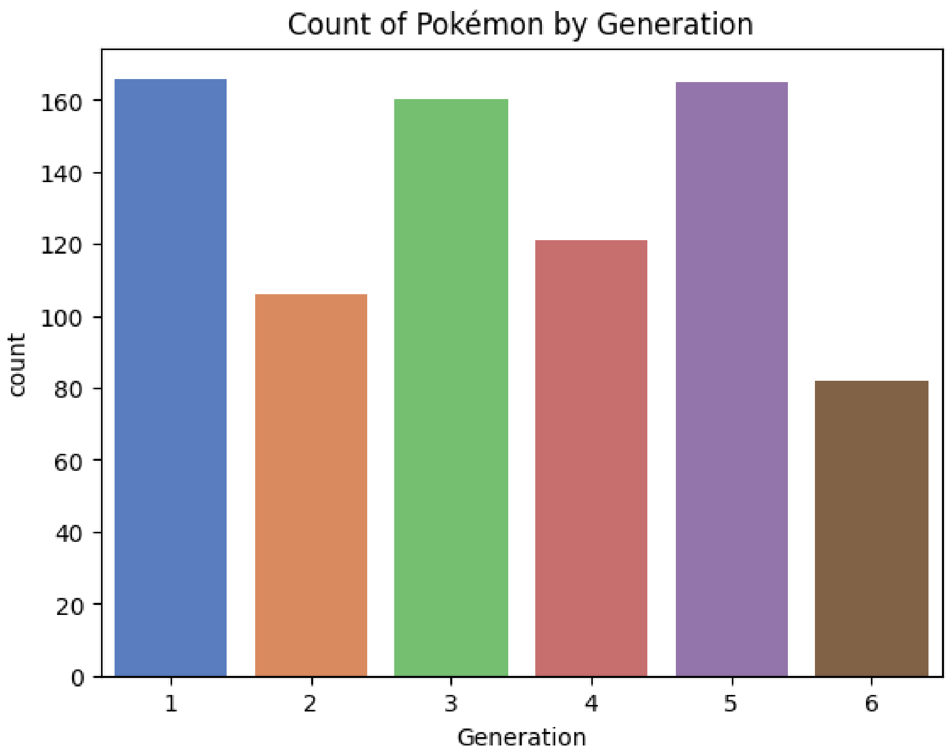
Speed Distribution by Legendary Status

Generation Trends

To visualize how stats and counts vary across generations:

```
# Average stats by generation
stats_by_gen = data.groupby('Generation')[['HP', 'Attack', 'Defense', 'Sp. Atk',
'Sp. Def', 'Speed']].mean()
stats_by_gen.plot(kind='bar', figsize=(10, 6), colormap='rainbow')
plt.title('Average Stats by Generation')
plt.ylabel('Average Value')
plt.show()

# Count of Pokémon by generation
sns.countplot(x='Generation', data=data, palette='muted')
plt.title('Count of Pokémon by Generation')
plt.show()
```





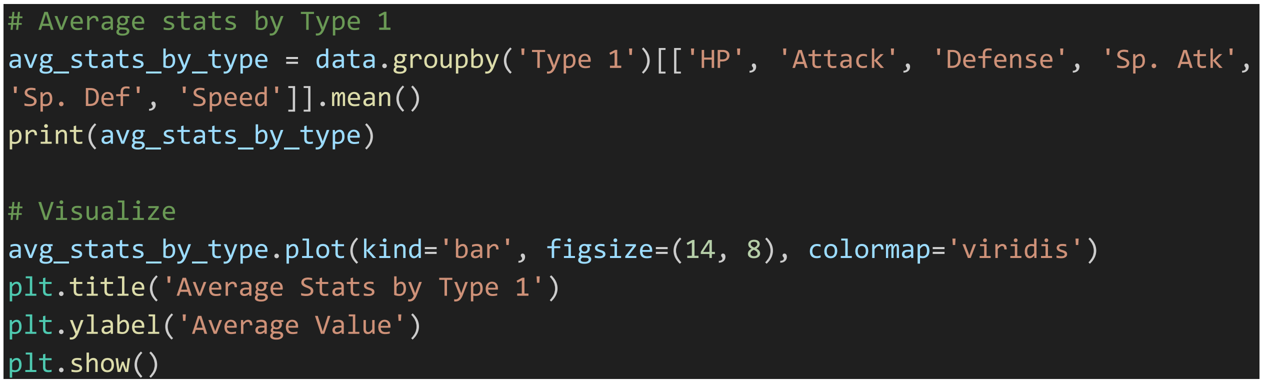
Top Performers

To identify the top 10 Pokémon from the analyzed data.

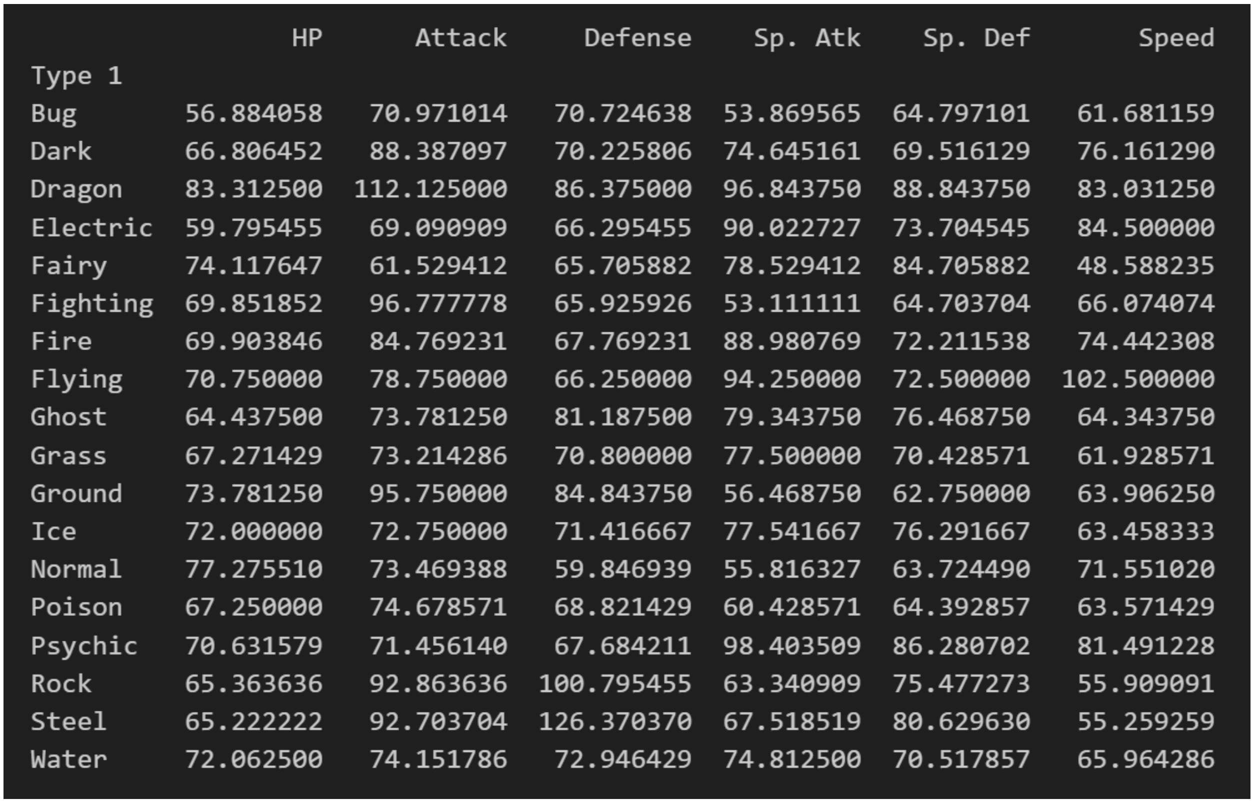
```
# Add a Total column
data['Total'] = data[['HP', 'Attack', 'Defense', 'Sp. Atk', 'Sp. Def',
'Speed']].sum(axis=1)

# Top 10 Pokémon by Total stats
top_10 = data.nlargest(10, 'Total')[['Name', 'Type 1', 'Type 2', 'Total']]
print(top_10)
```

```
#                        Name   Type 1    Type 2  Total
163       MewtwoMega Mewtwo X  Psychic  Fighting    780
164       MewtwoMega Mewtwo Y  Psychic       NaN    780
426     RayquazaMega Rayquaza   Dragon    Flying    780
422       KyogrePrimal Kyogre    Water       NaN    770
424     GroudonPrimal Groudon   Ground      Fire    770
552                    Arceus   Normal       NaN    720
268   TyranitarMega Tyranitar     Rock      Dark    700
409   SalamenceMega Salamence   Dragon    Flying    700
413   MetagrossMega Metagross    Steel   Psychic    700
418         LatiasMega Latias   Dragon   Psychic    700
```

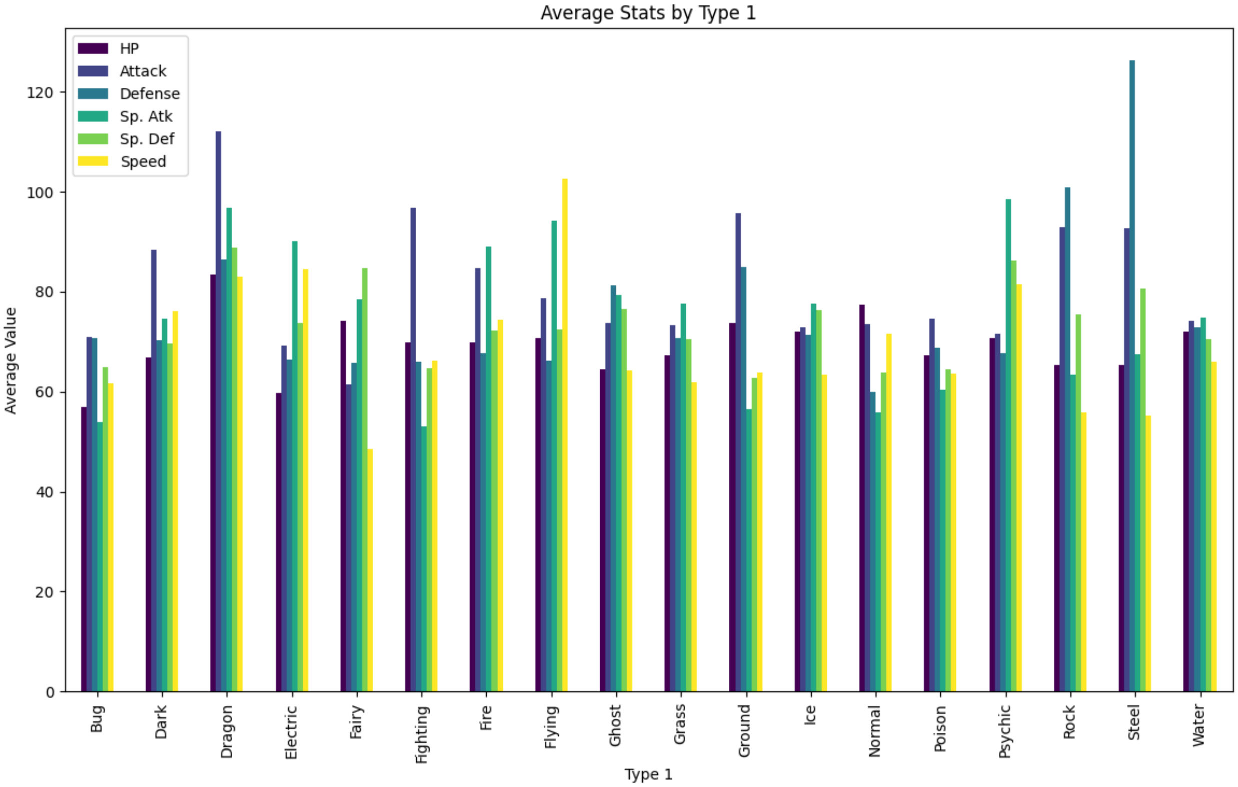
Type Effectiveness



```python
# Average stats by Type 1
avg_stats_by_type = data.groupby('Type 1')[['HP', 'Attack', 'Defense', 'Sp. Atk',
'Sp. Def', 'Speed']].mean()
print(avg_stats_by_type)

# Visualize
avg_stats_by_type.plot(kind='bar', figsize=(14, 8), colormap='viridis')
plt.title('Average Stats by Type 1')
plt.ylabel('Average Value')
plt.show()
```



```
                 HP      Attack     Defense    Sp. Atk    Sp. Def       Speed
Type 1
Bug       56.884058   70.971014   70.724638  53.869565  64.797101   61.681159
Dark      66.806452   88.387097   70.225806  74.645161  69.516129   76.161290
Dragon    83.312500  112.125000   86.375000  96.843750  88.843750   83.031250
Electric  59.795455   69.090909   66.295455  90.022727  73.704545   84.500000
Fairy     74.117647   61.529412   65.705882  78.529412  84.705882   48.588235
Fighting  69.851852   96.777778   65.925926  53.111111  64.703704   66.074074
Fire      69.903846   84.769231   67.769231  88.980769  72.211538   74.442308
Flying    70.750000   78.750000   66.250000  94.250000  72.500000  102.500000
Ghost     64.437500   73.781250   81.187500  79.343750  76.468750   64.343750
Grass     67.271429   73.214286   70.800000  77.500000  70.428571   61.928571
Ground    73.781250   95.750000   84.843750  56.468750  62.750000   63.906250
Ice       72.000000   72.750000   71.416667  77.541667  76.291667   63.458333
Normal    77.275510   73.469388   59.846939  55.816327  63.724490   71.551020
Poison    67.250000   74.678571   68.821429  60.428571  64.392857   63.571429
Psychic   70.631579   71.456140   67.684211  98.403509  86.280702   81.491228
Rock      65.363636   92.863636  100.795455  63.340909  75.477273   55.909091
Steel     65.222222   92.703704  126.370370  67.518519  80.629630   55.259259
Water     72.062500   74.151786   72.946429  74.812500  70.517857   65.964286
```


Average Stats by Type 1
HP
Attack
Defense
Sp. Atk
Sp. Def
Speed
Average Value
0
20
40
60
80
100
120
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
Type 1

Conclusion

This project demonstrates the power of data analysis and visualization in uncovering meaningful insights from the Pokémon dataset. By systematically cleaning the data, visualizing distributions, and exploring correlations, we gained a deeper understanding of the relationships between Pokémon attributes, types, and generations. The analysis also highlighted the unique characteristics of Legendary Pokémon and identified trends that distinguish top-performing Pokémon.

These findings not only enhance our understanding of Pokémon characteristics but also serve as a foundation for further exploration. The reproducible Python commands included in this project provide an accessible framework for extending the analysis, allowing others to customize and build upon these insights.

Overall, this project showcases the potential of data-driven approaches to analyze structured datasets, offering valuable insights and practical experience in leveraging Python for data analysis and visualization.

References

Dataset

- Pokémon Dataset: https://gist.github.com/armgilles/194bcff35001e7eb53a2a8b441e8b2c6

  Credits: https://github.com/armgilles

Libraries and Dependencies

1. **Pandas**:
   - Official Documentation: https://pandas.pydata.org/
2. **NumPy**:
   - Official Documentation: https://numpy.org/
3. **Matplotlib**:
   - Official Documentation: https://matplotlib.org/
4. **Seaborn**:
   - Official Documentation: https://seaborn.pydata.org/