Pokémon Data Analysis Using SQL

Exploring Stats, Types, and Generations in the Pokémon Dataset
David Praseuth
April 2025
Tools Used: MySQL Workbench, SQL, Google Docs, GitHub

1. Introduction

This project explores a Pokémon dataset using SQL, aiming to uncover patterns in stats, types, and legendary distributions across generations. I chose this dataset not just because it's rich in information, but because Pokémon has always been a shared interest between me and my young niece and nephew. As I improve my data skills, connecting with something they love made this project more meaningful and fun.

My goal was to write clean, readable SQL to answer real questions about the dataset and translate those findings into clear insights. This also serves as one of my first steps in building a data analyst portfolio.

2. Dataset Description

This project uses a Pokémon dataset originally sourced from Kaggle and compiled using PokeAPI as of March 2025. The dataset contains 1,025 Pokémon entries, including alternate forms such as regional variants, Mega Evolutions, and other stat- or type-changing forms.

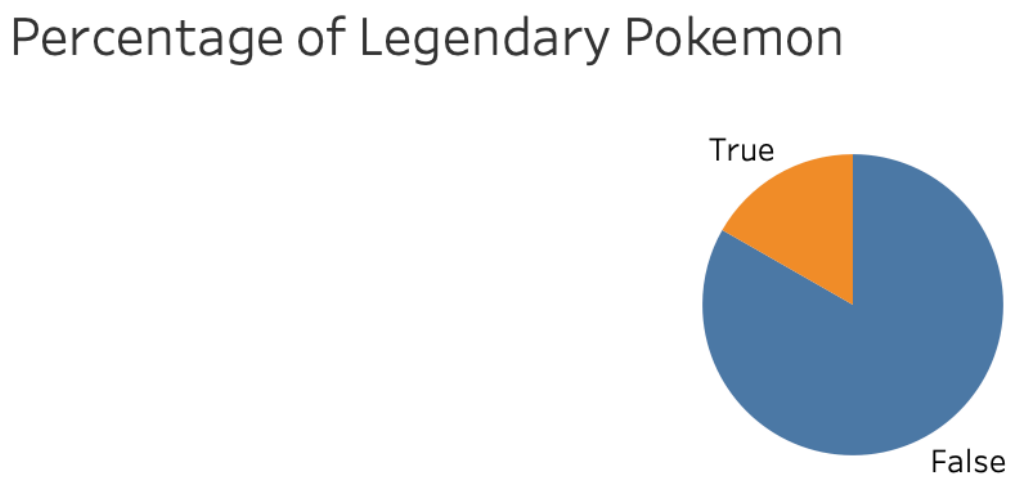
Each row represents a Pokémon or one of its alternate forms, and includes the following key attributes:

- Identification: Name, National Dex number, Generation, Evolution stage
- Typing: Primary type, secondary type (with secondary type flag), and most common type combinations
- Stats: Base stat total, HP, Attack, Defense, Special Attack, Special Defense, Speed
- Classification: Legendary status, alternative form type and flag
- Catch and Size Data: Catch rate, height (in dm and inches), weight (in hg and pounds)
- Color ID: For visual classification and sorting

3. Analysis Questions

1. What % of Pokemon are legendary?

```
SELECT COUNT(*) AS Total_Pokemon,
SUM(CASE WHEN `legendary status` = "True" THEN 1 ELSE 0 END) AS
Total_Legendary,
ROUND(
(SUM(CASE WHEN `legendary status` = "True" THEN 1 ELSE 0 END) * 100.0) /
COUNT(*),
2) AS Percent_Legendary
FROM all_pokemon_data;
```



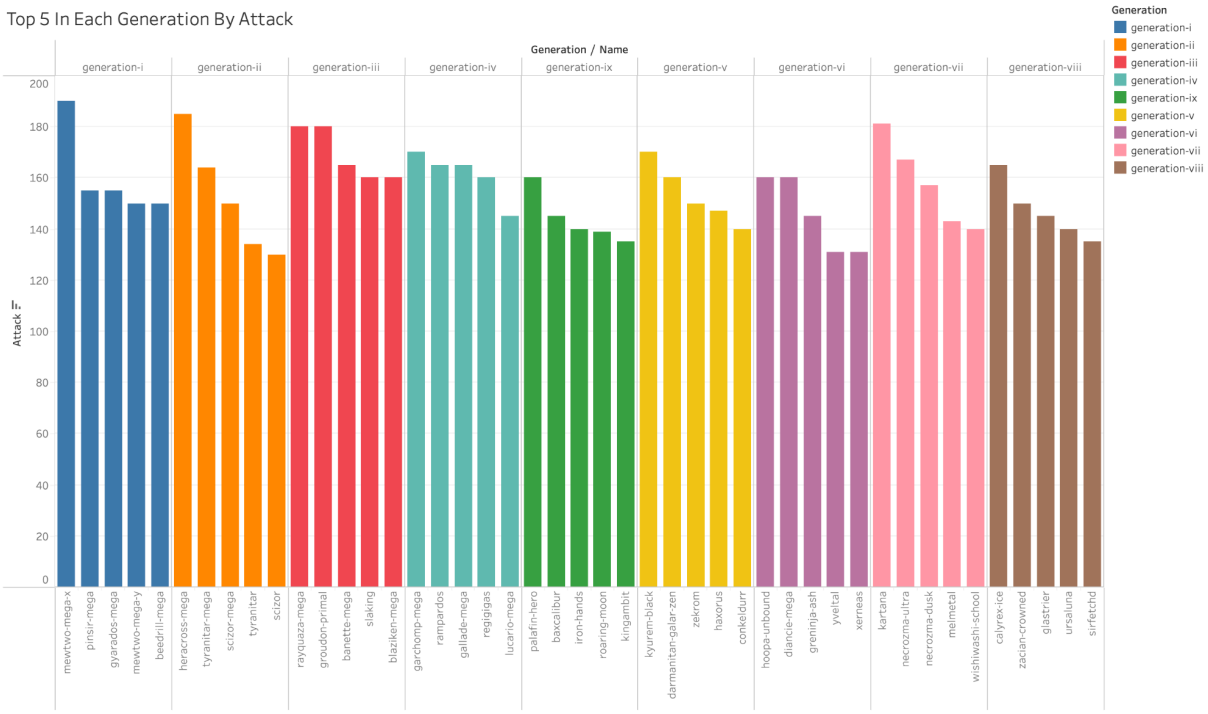
Insight: Approximately 16.72% of all Pokémon are considered legendary, highlighting their rarity and special status within the franchise.

2. What are the top 5 Pokemon in each generation by Attack?

```
WITH pokemon_rank AS (
SELECT *, ROW_NUMBER() OVER (PARTITION BY generation ORDER BY attack
DESC) AS atk_rank
FROM all_pokemon_data)

SELECT name, attack, generation, atk_rank
```

```
FROM pokemon_rank
WHERE atk_rank <= 5
ORDER BY generation, atk_rank;
```



Insight: Each generation features at least one standout attacker, but Generations I and III dominate in raw offensive power. Legendary or pseudo-legendary Pokémon are often among the top.

3. *What are the top 5 Pokemon in each type?*

```
WITH ranked_pokemon AS (
    SELECT
        name,
        `primary typing`,
        `base stat total`,
        ROW_NUMBER() OVER (
        PARTITION BY `primary typing`
        ORDER BY `base stat total` DESC
        ) AS ranking

    FROM all_pokemon_data
```

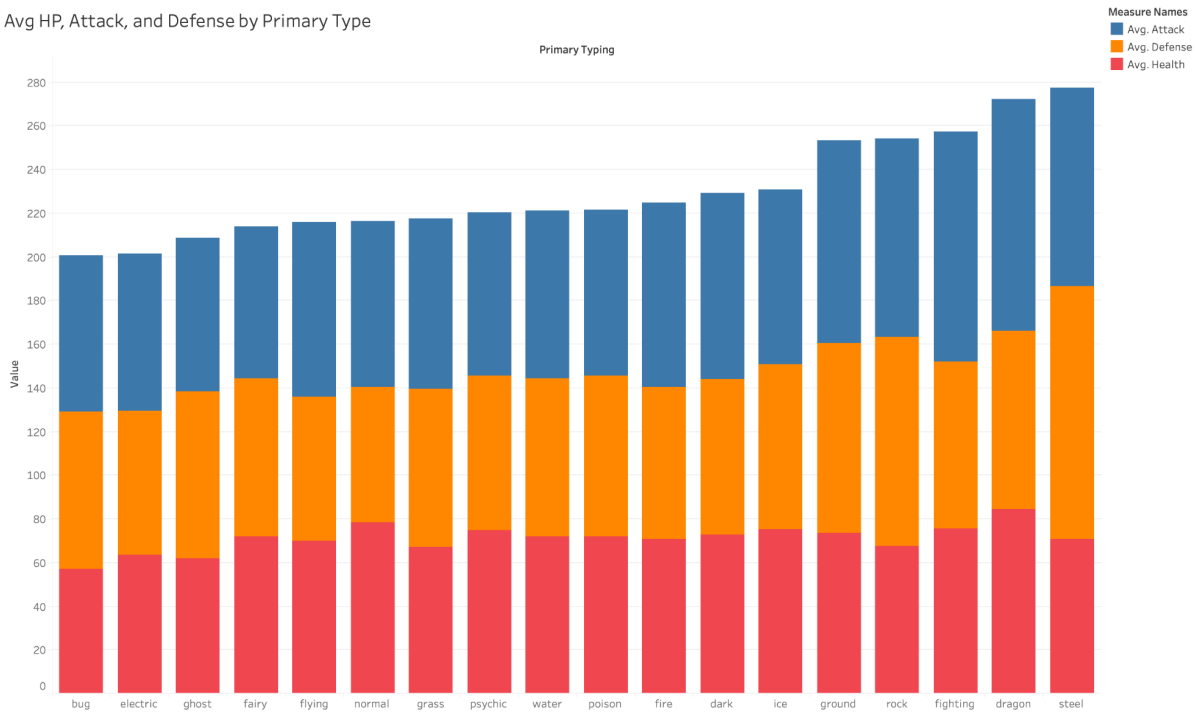
)

```
SELECT *
FROM ranked_pokemon
WHERE ranking <= 5
ORDER BY `primary typing`, ranking;
```

Insight: Legendaries dominate the top of most types, but a few non-legendaries, such as Scizor show that strong Pokémon can exist outside of legendary status.

4. *What is the average HP, Attack, and Defense by primary type?*

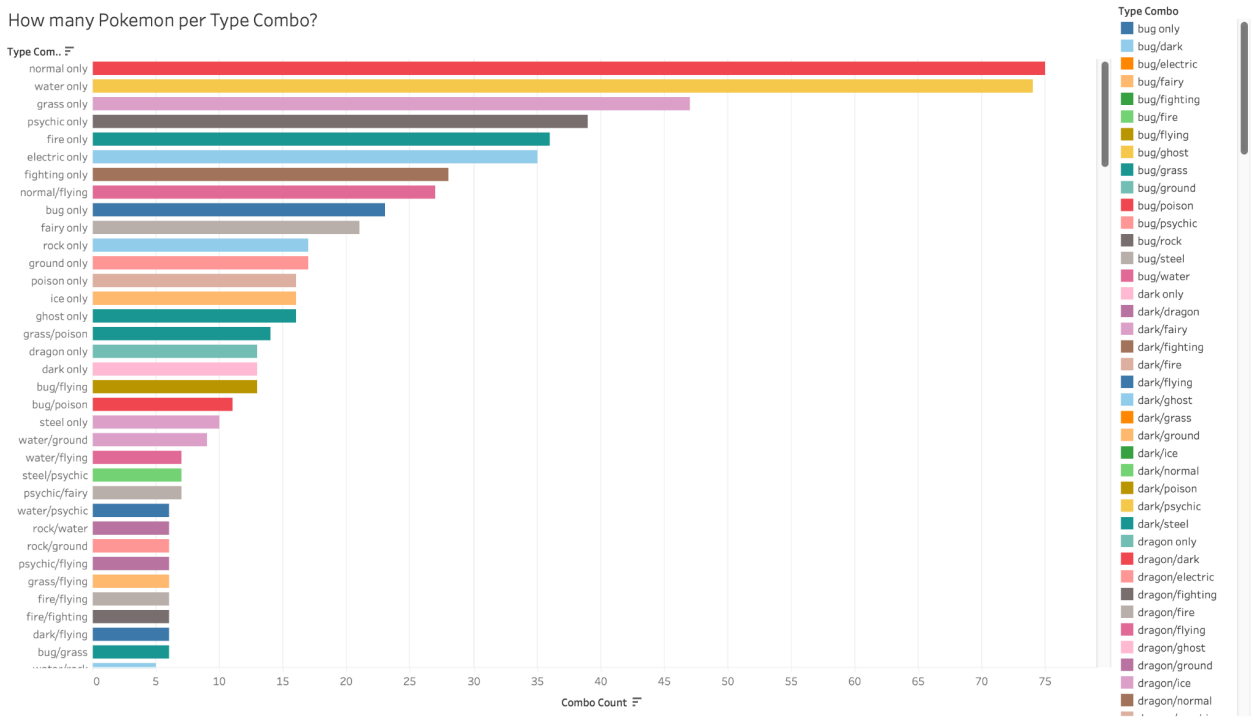
```
SELECT `primary typing`, avg(health) AS avg_HP, avg(attack) AS avg_Atk,
avg(defense) AS avg_Def
FROM all_pokemon_data
GROUP BY `primary typing`;
```



Insight: Rock and Steel types tend to have higher Defense averages, while Fighting and Dragon types lead in Attack. HP is highest in Normal and Water types, often due to bulky Pokémon like Snorlax.

5. *What is the most common type combo?*

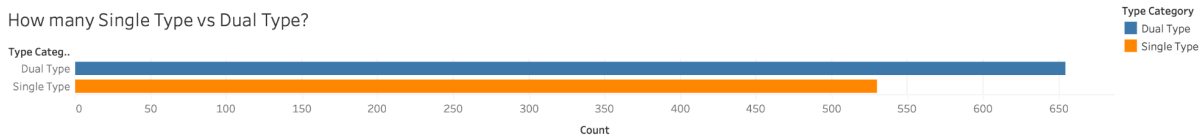
```
SELECT
        CASE
                WHEN `secondary typing` = '' OR `secondary typing` IS NULL THEN
CONCAT(`primary typing`, ' only')
      ELSE CONCAT(`primary typing`, '/', `secondary typing`)
        END AS type_combo,
   COUNT(*) AS combo_count
FROM all_pokemon_data
WHERE form = "base"
GROUP BY type_combo
ORDER BY combo_count DESC;
```



Insight: While there are a wide variety of type combinations in the Pokémon universe, the most common is actually no combination at all – a pure Normal type. There are 75 Normal only Pokémon, highlighting how often this type is used as a base or default form, particularly in early-game species.

6. ***How many dual vs single type?***

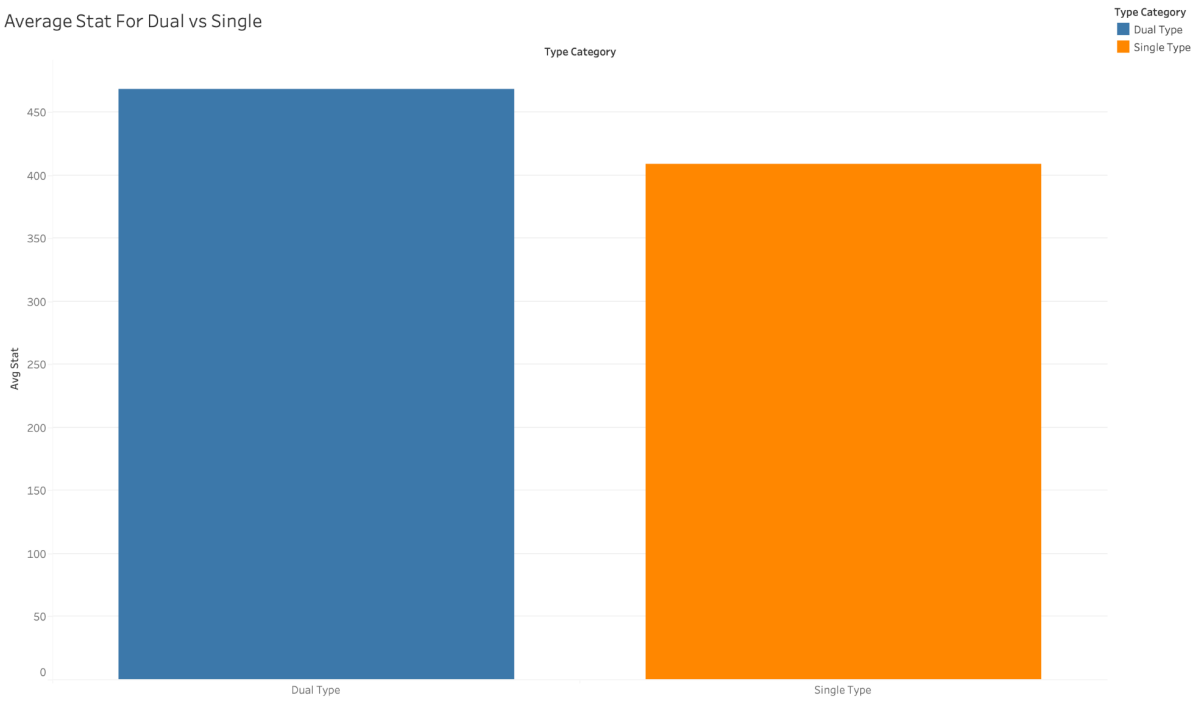
```
SELECT
    CASE
        WHEN `secondary typing` IS NULL OR `secondary typing` = '' THEN
'Single Type'
        ELSE 'Dual Type'
    END AS type_category,
  COUNT(*) AS count
FROM all_pokemon_data
GROUP BY type_category;
```



Insight: Around 55% of all Pokémon are dual-typed, showcasing how common type combinations are in the series. This provides more strategic variety for players.

7. ***What is the average base stat for single vs dual type?***

```
SELECT
    CASE
        WHEN `secondary typing` IS NULL OR `secondary typing` = '' THEN
'Single Type'
        ELSE 'Dual Type'
    END AS type_category,
  avg(`base stat total`) AS avg_stat
FROM all_pokemon_data
GROUP BY type_category;
```



Insight: With dual-types offering more tactical diversity, it's no surprise they carry slightly higher average stats — a nod to the complexity they bring to team building.

4. Conclusion

Through this project, I practiced SQL skills including window functions, CTEs, CASE statements, and aggregations.
The Pokémon dataset provided a fun way to uncover patterns in base stats, type advantages, and legendary Pokémon distribution.
In future projects, I would also like to explore battle outcomes or deeper evolution analysis.