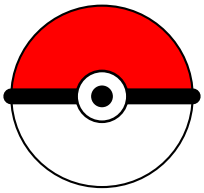

Pokémon Analysis

Data Science · Mr.Merrick · February 11, 2026

Dr. Oak has hired you as a junior data scientist to help him understand patterns and trends in Pokémon data. Your mission: ask questions, explore creatively, and use R to find insights worth sharing.

Overview

Using the Pokémon dataset, your task is to conduct open-ended exploratory research. You will use R to clean, visualize, and describe one-variable and simple two-variable patterns that help answer questions about Pokémon. A detailed description of the dataset can be found on page 2.

Guiding Research Questions (Choose, Combine, or Extend)

Dr. Oak encourages creativity. Use these as inspiration—but feel free to pose your own related questions. There are no "right answers," only thoughtful, data-driven ones.

Q1. Summarize the Dataset.
How many observations? How many variables?

Q2. What does it mean for a Pokémon to be "strong"?
Define "strength" however you wish (e.g., Combat Power, Attack, or a combination). Which Pokémon best embody that strength? How common or rare are they?

Q3. How do different Pokémon *types* vary in their attributes? Do certain types tend to be heavier, faster, or have higher CP? Are there clear patterns when you look across type distributions?

Q4. Are Legendary Pokémon noticeably stronger than non-Legendaries? How do their stat distributions compare? Are they uniformly dominant, or do some normal Pokémon rival them?

Q5. Is there an association between a Pokémon's weight and height? Does size predict other characteristics like CP or speed? Are certain types (e.g., Flying, Water) exceptions to general trends?

Q6. Have newer generations of Pokémon become stronger or weaker? Compare distributions of total strength or CP across generations. What trends emerge as new Pokémon are introduced?

Your report should explore **at least three** of these questions in depth and be submitted as a pdf.

Dataset Description and Context

The **Complete Pokémon Dataset** compiled by Rounak Banik (2018) is published on Kaggle. This dataset consolidates information from the official Pokémon main-series video games (*Pokémon Red/Blue* through *Pokémon Sun/Moon*, Generations I–VII). It includes one row per Pokémon species or form, with variables describing type, base statistics, generation, and legendary status.

- **Citation:** Banik, R. (2018). *The Complete Pokémon Dataset*. Kaggle. Retrieved from https://www.kaggle.com/datasets/rounakbanik/pokemon

These base statistics are drawn from Game Freak's canonical data tables and are consistent with the in-game Pokédex entries from the core RPG series. No player-specific modifications (e.g., individual values or effort values) are included—these are species-level attributes.

Relation to Pokémon GO and the Trading Card Game

While all three media (the main games, Pokémon GO, and the Trading Card Game) share the same Pokémon species, they use different numerical systems for attributes:

System	Stats Used	Range / Scale	Notes
Main Games	HP, Attack, Defense, Special Attack, Special Defense, Speed	1–255	Turn-based RPG; 6 attributes per Pokémon.
Pokémon GO	Attack, Defense, Stamina	∼10–300	Simplified for mobile play; 3 attributes + CP.
Trading Card Game (TCG)	HP, Attack Power (per move)	30–340 HP typical	Designed for card game balance.

Table 1: Comparison of stat systems across Pokémon media.

In the main series, base stats form the foundation of a deeper system involving:

- **Individual Values (IVs):** Random 0–31 bonuses unique to each Pokémon
- **Effort Values (EVs):** Training-based stat adjustments
- **Natures:** Personality modifiers that boost or reduce specific stats

Together, these create large individual variation between Pokémon of the same species—an important concept in probability and variation modeling.

By contrast, **Pokémon GO** collapses these six core stats into three (Attack, Defense, Stamina) to suit a real-time mobile environment. Individual variation is introduced via simplified IVs ranging from 0–15 per stat. The game's "Combat Power (CP)" is then derived from these stats through a multiplicative formula.

The **Trading Card Game (TCG)** discards these mechanics entirely, using abstract HP and attack values chosen for card game balance rather than biological realism or direct translation from the video games.

Pokémon Data Science Tutorial — Group Tutorial (R)

Setup and Data Import

We'll load a few tidy packages, import the CSV, and clean column names.

```r
# --- Libraries ---
library(readr)
library(dplyr)
library(ggplot2)
library(janitor)

# --- Load and clean dataset ---
pk <- read_csv("pokemon.csv") # ensure this file is in your working directory
pk <- clean_names(pk) # e.g., "Sp. Atk" -> "sp_atk"

# --- Preview the data ---
glimpse(pk)
summary(pk)
```

Q1. Summarize the Dataset

Prompt: How many observations (rows)? How many variables (columns)? What are the basic characteristics of the data?

```r
# Rows and columns
dim(pk)

# Variable names
names(pk)

# --- Create a total stat variable (sum of base stats) ---
# NOTE: Depending on your CSV, the columns are typically: hp, attack, defense, sp_atk, sp_def,
    speed.
# If your file uses "sp_attack"/"sp_defense", rename them or adjust the line below.
pk <- pk %>%
  mutate(total = hp + attack + defense + sp_atk + sp_def + speed)

# Simple summary statistics
pk %>%
  summarise(
    n_pokemon = n(),
    n_variables = ncol(pk),
    mean_total = mean(total, na.rm = TRUE),
    median_total= median(total, na.rm = TRUE)
  )

# Average total stats by Generation
pk %>%
  group_by(generation) %>%
  summarise(avg_total = mean(total, na.rm = TRUE))
```

Q2. What Does it Mean for a Pokémon to be "Strong"?

Prompt: We'll define strength as the sum of the six base stats (`total`). Find the strongest Pokémon overall and visualize the distribution.

```
# Top 10 Pokemon by total stats
pk %>%
  arrange(desc(total)) %>%
  select(name, type_1, type_2, total) %>%
  head(10)

# Simple histogram of total strength
ggplot(pk, aes(x = total)) +
  geom_histogram(bins = 25, fill = "skyblue", color = "white") +
  labs(title = "Distribution of Total Pokemon Strength",
       x = "Total Base Stats", y = "Count")
```

Q3. How Do Different Types Vary?

Prompt: Compare average strength across primary types and show a type comparison plot.

```
# Average total stats by primary type
pk %>%
  group_by(type_1) %>%
  summarise(avg_total = mean(total, na.rm = TRUE)) %>%
  arrange(desc(avg_total))

# Boxplot comparing types
ggplot(pk, aes(x = type_1, y = total)) +
  geom_boxplot(fill = "lightgreen") +
  coord_flip() +
  labs(title = "Total Stats by Primary Type", x = "Type", y = "Total Stats")
```

Q4. Are Legendary Pokémon Stronger?

Prompt: Compare totals for Legendary vs. non-Legendary. Convert the 0/1 indicator to labeled categories for plotting.

```
# Summary table by legendary indicator (0/1)
pk %>%
  group_by(is_legendary) %>%
  summarise(avg_total = mean(total, na.rm = TRUE),
            count = n())

# Make a labeled factor for plotting
pk <- pk %>%
  mutate(is_legendary_factored = factor(is_legendary,
                                        levels = c(0, 1),
                                        labels = c("No", "Yes")))

# Boxplot: Legendary vs Non-Legendary
ggplot(pk, aes(x = is_legendary_factored, y = total, fill = is_legendary_factored)) +
  geom_boxplot() +
  labs(title = "Legendary vs Non-Legendary Pokemon",
       x = "Legendary?",
       y = "Total Stats") +
```

```
  scale_fill_manual(values = c("No" = "skyblue", "Yes" = "orange"))

# Faceted histograms (same y-scale)
ggplot(pk, aes(x = total)) +
  geom_histogram(bins = 25, color = "white", fill = "skyblue") +
  facet_wrap(~ is_legendary_factored, ncol = 2) +
  labs(title = "Distribution of Total Base Stats",
       x = "Total (BST)",
       y = "Count")

# Faceted histograms (free y-axis to handle different group sizes)
ggplot(pk, aes(x = total)) +
  geom_histogram(bins = 25, color = "white", fill = "skyblue") +
  facet_wrap(~ is_legendary_factored, ncol = 2, scales = "free_y") +
  labs(title = "Distribution of Total Base Stats",
       x = "Total (BST)",
       y = "Count")
```

Q5. Is There an Association Between Height and Weight?

Prompt: Make a simple scatter plot and (optionally) compute correlation.

```
ggplot(pk, aes(x = height_m, y = weight_kg)) +
  geom_point(alpha = 0.7, color = "darkblue") +
  labs(title = "Height vs Weight of Pokemon", x = "Height (m)", y = "Weight (kg)")

# Optional: correlation
cor(pk$height_m, pk$weight_kg, use = "complete.obs")
```

Q6. Have Newer Generations Become Stronger or Weaker?

Prompt: Compare total strength across generations.

```
ggplot(pk, aes(x = factor(generation), y = total, fill = factor(generation))) +
  geom_boxplot() +
  labs(title = "Pokemon Strength Across Generations",
       x = "Generation", y = "Total Stats") +
  scale_fill_brewer(palette = "Set3")

pk %>%
  group_by(generation) %>%
  summarise(avg_total = mean(total, na.rm = TRUE),
            median_total = median(total, na.rm = TRUE))
```

Wrap-Up

We used simple summaries and visuals to explore patterns: overall strength, type differences, legendary comparisons, size relationships, and generation trends. In future assignments, you'll extend these ideas and write short interpretations for each figure or table.

Python Supplement (pandas + seaborn)

Setup & Data Import

```
# If needed:
# !pip install pandas seaborn matplotlib

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_theme(context="notebook", style="whitegrid")

# Load CSV (same directory as your notebook / script)
pk = pd.read_csv("pokemon.csv")

print(pk.shape)
print(pk.columns.tolist())
```

Q1. Summarize the Dataset

```
# Create 'total' (Base Stat Total) if you have these columns
stat_cols = ["hp", "attack", "defense", "sp_atk", "sp_def", "speed"]
missing = [c for c in stat_cols if c not in pk.columns]
if missing:
    print("Missing:", missing)
else:
    pk["total"] = pk[stat_cols].sum(axis=1)

# Rows, columns
n_rows, n_cols = pk.shape
print("observations:", n_rows, "| variables:", n_cols)

# Simple summary
if "total" in pk.columns:
    print(pk["total"].describe()[["mean","50%"]]) # mean, median

# Average total by Generation
if {"generation","total"}.issubset(pk.columns):
    print(pk.groupby("generation")["total"].mean())
```

Q2. What Does it Mean to be "Strong"?

```
# Define strength as 'total' and list strongest Pok\'emon
cols = [c for c in ["name", "type_1", "type_2", "total"] if c in pk.columns]
if "total" in pk.columns and cols:
    print(pk.sort_values("total", ascending=False)[cols].head(10))

# Histogram of total
if "total" in pk.columns:
    plt.figure()
    sns.histplot(pk["total"], bins=25)
    plt.title("Distribution of Total Pok\'emon Strength")
```

```
    plt.xlabel("Total Base Stats"); plt.ylabel("Count")
    plt.tight_layout(); plt.show()
```

Q3. How Do Types Vary?

```
# Average total by primary type
if {"type_1","total"}.issubset(pk.columns):
    avg_by_type = (pk.groupby("type_1")["total"]
                   .mean().reset_index(name="avg_total")
                   .sort_values("avg_total", ascending=False))
    print(avg_by_type)

# Boxplot by type
if {"type_1","total"}.issubset(pk.columns):
    plt.figure(figsize=(8,6))
    sns.boxplot(data=pk, x="type_1", y="total")
    plt.title("Total Stats by Primary Type")
    plt.xlabel("Type"); plt.ylabel("Total Stats")
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout(); plt.show()
```

Q4. Are Legendary Pokémon Stronger?

```
# Summaries by legendary status
if {"is_legendary","total"}.issubset(pk.columns):
    print(pk.groupby("is_legendary")["total"]
           .agg(avg_total="mean", count="size")
           .reset_index())

    # Make a labeled flag for plotting
    if pk["is_legendary"].dtype != "O":
        pk["is_legendary_flag"] = pk["is_legendary"].astype(int)
    else:
        pk["is_legendary_flag"] = pk["is_legendary"].astype(str).str.lower().map({
            "0":"0","1":"1","false":"0","true":"1","no":"0","yes":"1"
        }).fillna("0").astype(int)
    pk["is_legendary_label"] = pk["is_legendary_flag"].map({0:"No", 1:"Yes"})

    # Boxplot
    plt.figure()
    sns.boxplot(data=pk, x="is_legendary_label", y="total")
    plt.title("Legendary vs Non-Legendary (Total Stats)")
    plt.xlabel("Legendary?"); plt.ylabel("Total Stats")
    plt.tight_layout(); plt.show()

    # Faceted histograms (same y)
    g = sns.FacetGrid(pk, col="is_legendary_label", sharey=True)
    g.map_dataframe(sns.histplot, x="total", bins=25)
    g.set_axis_labels("Total (BST)", "Count")
    g.fig.subplots_adjust(top=0.85)
    g.fig.suptitle("Distribution of Total Base Stats")
    plt.show()

    # Faceted histograms (free y)
```

```
    g2 = sns.FacetGrid(pk, col="is_legendary_label", sharey=False)
    g2.map_dataframe(sns.histplot, x="total", bins=25)
    g2.set_axis_labels("Total (BST)", "Count")
    g2.fig.subplots_adjust(top=0.85)
    g2.fig.suptitle("Distribution of Total Base Stats (free y)")
    plt.show()
```

Q5. Height vs Weight

```
if {"height_m","weight_kg"}.issubset(pk.columns):
    plt.figure()
    sns.scatterplot(data=pk, x="height_m", y="weight_kg", alpha=0.7)
    plt.title("Height vs Weight of Pok\'emon")
    plt.xlabel("Height (m)"); plt.ylabel("Weight (kg)")
    plt.tight_layout(); plt.show()

    print("Correlation:",
          pk[["height_m","weight_kg"]].corr().iloc[0,1])
```

Q6. Generations: Stronger or Weaker?

```
if {"generation","total"}.issubset(pk.columns):
    plt.figure()
    sns.boxplot(data=pk, x=pk["generation"].astype("category"), y="total")
    plt.title("Pok\'emon Strength Across Generations")
    plt.xlabel("Generation"); plt.ylabel("Total Stats")
    plt.tight_layout(); plt.show()

    print(pk.groupby("generation")["total"]
            .agg(avg_total="mean", median_total="median")
            .reset_index())
```