

Project 3 - Pokémon Data Analysis

([Google Slides](#) | [Project Repository](#))

Caden Harris & Nate Taylor

Introduction

The dataset we are looking at contains Pokemon and their stats, types, names, generations, moves, and adjacent information. Analyzing Pokemon can be useful for players when building their teams as well as inform non-players about interesting statistics within the creatures. Our analysis involves means, Pearson correlation tests, bar charts, scatterplots, **[fill in]**. We found in our results the fastest Pokemon types, the inflation of stats over time, and

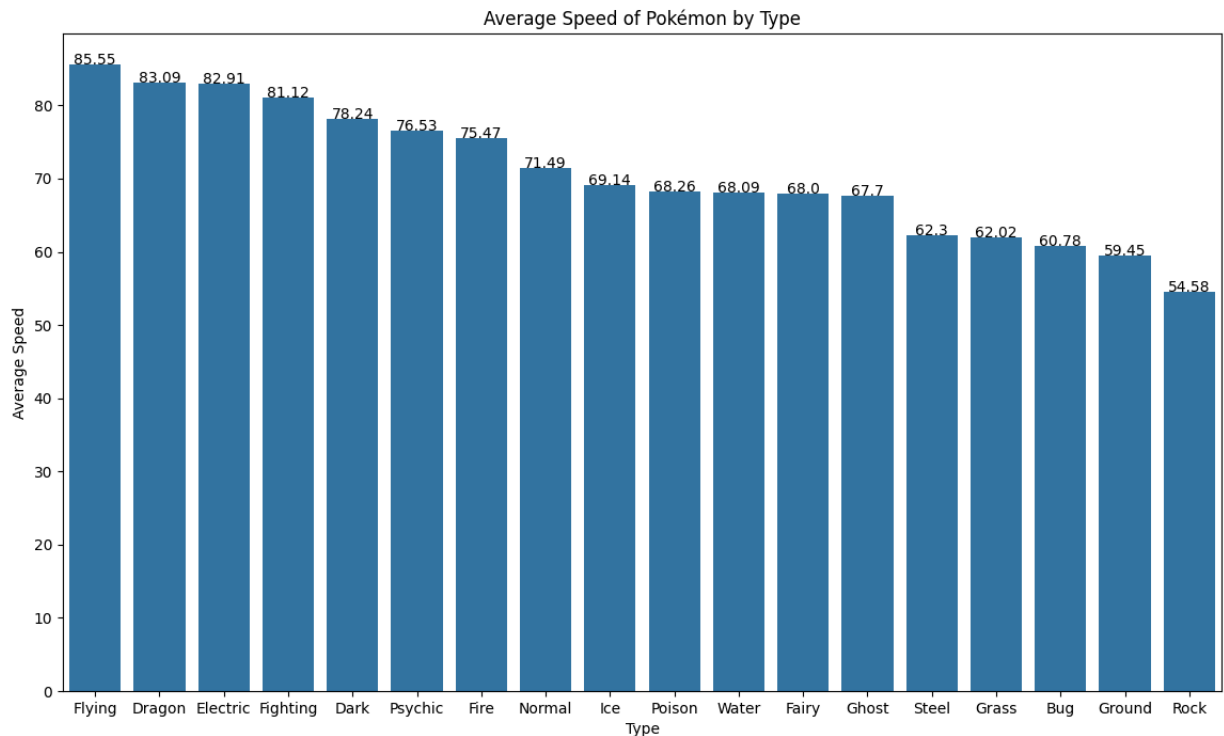
Dataset

The Pokebase python wrapper for the PokeAPI has a vast amount of data on all Pokemon and adjacent data present in the games. The compiled dataset we used has some of the same data but placed into a CSV to make it faster to pull many pokemon and their stats at once. These are suitable because they provide accurate up to date information on every Pokemon and the data values within them.

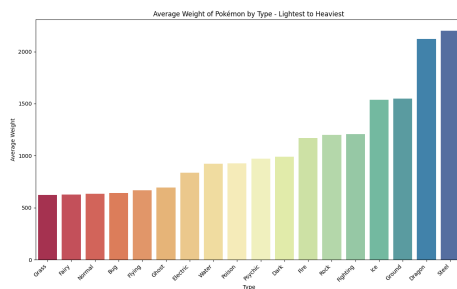
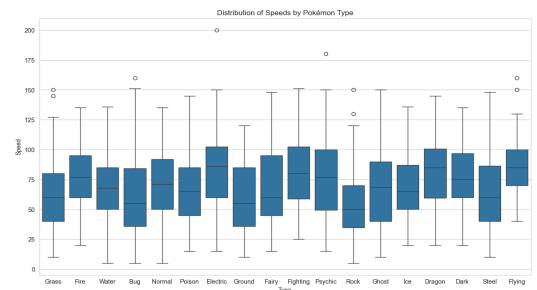
Analysis technique - *describes what technique was used and why it was used. You do not need to describe the technique in detail, but you need to describe why you think it is suitable for the data and/or the domain. 1 paragraph.*

Results

For one of our analyses we wanted to look at a pokemon stat by type. We picked speed because it makes intuitive sense for certain types to be “faster”. The results showed that Flying types have the highest average Speed stat and Rock types have the lowest.

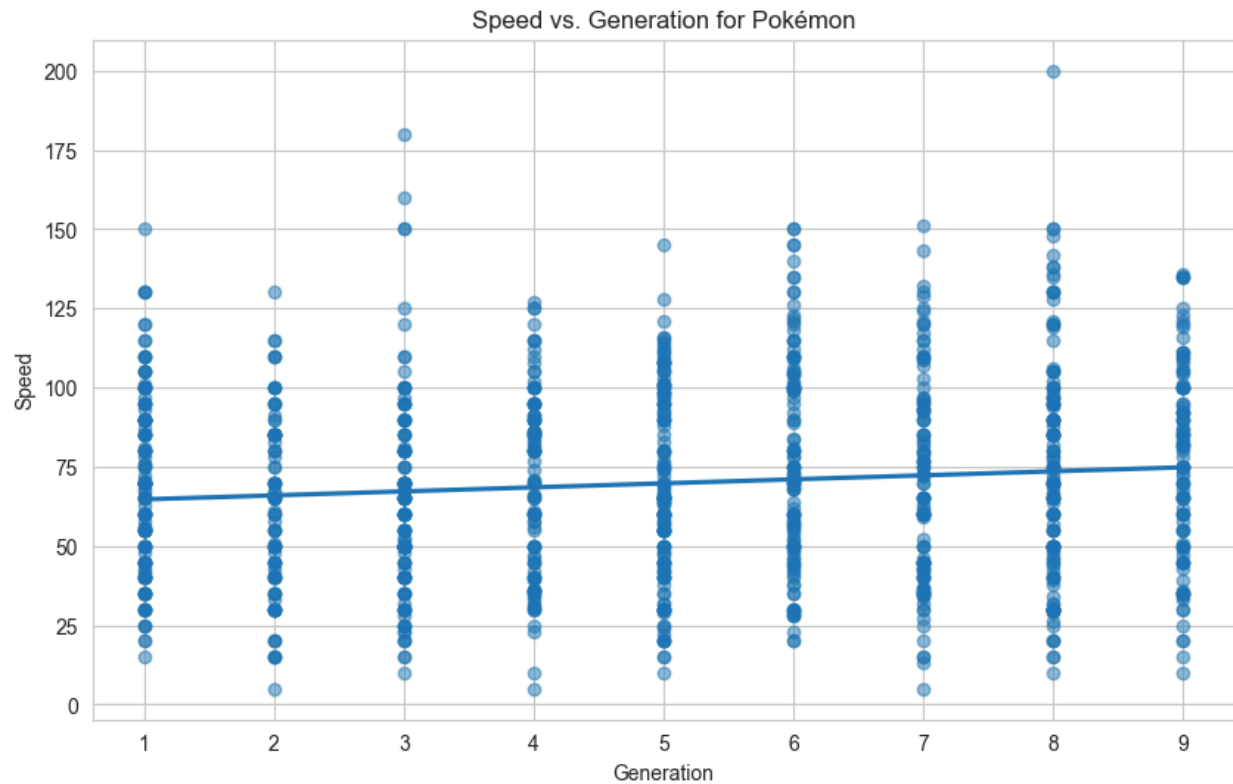


There is a clear difference between the types and putting them in descending order shows us the trend of fastest to slowest types. Interestingly, the fastest singular Pokemon is an Electric type, Regieleki with a value of 200. The slowest type is Rock which makes intuitive sense since rocks are slow and heavy, and flying types are going to be lighter and swifter. This is useful for showing players that if they want to value speed, flying types are going to be more desirable.



Just to look at it, we also calculated the average weights of Pokemon by type. Flying and Rock are on opposite ends of the spectrum, but not on the polar ends as one might assume based on the average weight stats. This demonstrates that stats are more complex and have many factors involved for the composition of the Pokemon.

We also looked at the correlation between speed and generation (and other stats) to see if there is any connection:



We get a correlation of ~ 0.086 with a p-value of ~ 0.003 . This suggests a slight upward increase of the stat over the generations which means there is stat inflation over time as more Pokémon are introduced. We did the same analysis for every stat in the compiled data and it yielded near-identical lines for each of them, emphasizing there is stat inflation.

Technical - *This section should be written for the instructor. It should be between 1 and 2 paragraphs (you do not need to have section header text for the subsections).*

Data Preparation - Any formatting, cleaning, and/or wrangling you had to do to prepare the dataset for analysis.

Analysis - Why the analysis technique is suitable for your dataset and goals. For example, if you use a Support Vector Machine to predict the sepal length of an Iris, you should describe why an SVM is a good choice. Describe any novel techniques that you are using. This subsection is required even if you are required to use a particular analysis technique for this project.

Analysis process - This subsection describes the process you went through to find your results. You should include any failed attempts and adjustments you might have made. Also describe any alternative approaches that should/could have been taken. You may have a lot of text for this or very little.

Part of our data preparation involved figuring out the PokeAPI which could be slow when pulling lots of data. To look at Speed and Types, we decided to pull in the pre-compiled dataset. To clear it and format it to look at stats by Type, we had to consolidate types for Pokemon with two types. This involved clearing the empty 2nd type column for some and considering each type as its own statistic.