Organizing and Visualizing Data in Python

Becky Lee

2020-05-09

Hello world! Welcome to my very first blog post. This beginner level guide will show you various ways of organizing and visualizing data using Python. Everything is at the beginner level! Let’s get started.

For this tutorial style walkthrough, we’ll be working with the pokemon dataset, renamed ‘poke’ here for convenience. The python packages ‘pandas’ and ‘seaborn’ will come in handy too!

Now that we are somewhat familiar with the dataset, we could start organizing and trying different functions.

One way is to select certain columns, like we do in R with the select() fucntion from dplyr. The format of the code is shown below, where the numbers represent their respective columns. Not that the Name column is ‘0’ and not ‘1’.

The exact same result could be done when using the column names instead of numbers.

Various functions, such as mean(), median(), max(), or count() could be performed on data. The code below takes the mean HP of the dataset defined earlier as ‘poke1’.

We could also use certain criteria to select only observations which meet that condition. By using count() and mean(), we know that only 378 pokemon out of the original 800 have an HP that is above average.

Using the condition above, create another dataset, ‘poke2’ for example. This dataset only includes pokemon which have higher than average HP. Let’s create a new column called ‘Total’, which will be the ‘Attack’ plus ‘Defense’ point number.

Visualizing data is an important part of coding, especially for scientists. Let’s go through a few different types of plots, starting with a general scatterplot.

We could also do faceting, and make separate histograms for each column variable. This includes every pokemon and gives simple distributions which are helpful for eyeballing trends.

To find correlations between every single variable, create a correlation table using corr().

A correlation heatmap puts those numbers above into a visual display with colors.

To see if being Legendary significantly affects a pokemon’s stats, we could facet once again. The example below compares the speed distributions for Legendary vs. non-Legendary pokemons.

That’s the end of the tutorial, but there is plenty more to explore in Python! Good luck.