Exploring Univariate Data

Using Super Hero data to get started with univariate EDA in Python

Tara Boyle Follow
Mar 12, 2019 · 4 min read ★

Photo by Limor Zellermayer on Unsplash

Wikipedia states that “univariate analysis is perhaps the simplest form of statistical analysis. . . The key fact is that only one variable is involved."

Because univariate analysis is so simple, it's a good place to start in an exploratory analysis.

Some questions to consider when getting started can include:

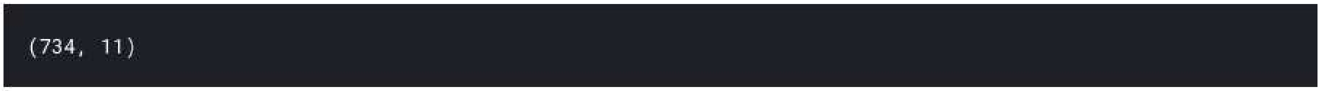
- How many variables do I have?
- Do I have missing data?
- What types of variables do I have?

We will explore the Super Heroes Dataset from Kaggle to begin answering these questions. The data includes two csv files. The first which we'll use here, contains characteristics of each Super Hero. The second lists what superpowers each hero has. The full notebook can be found here.

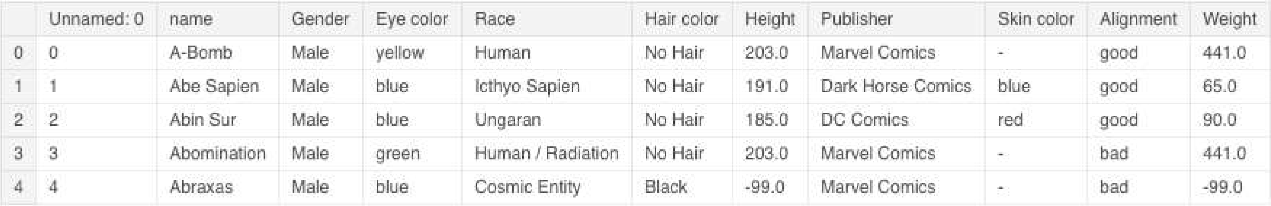
How many variables do I have?

I like to get started by viewing the first few rows of the dataframe and printing out the shape:

```
print(info_df.shape)
info_df.head()
```



```
(734, 11)
```



	Unnamed: 0	name	Gender	Eye color	Race	Hair color	Height	Publisher	Skin color	Alignment	Weight
0	0	A-Bomb	Male	yellow	Human	No Hair	203.0	Marvel Comics	-	good	441.0
1	1	Abe Sapien	Male	blue	Icthyo Sapien	No Hair	191.0	Dark Horse Comics	blue	good	65.0
2	2	Abin Sur	Male	blue	Ungaran	No Hair	185.0	DC Comics	red	good	90.0
3	3	Abomination	Male	green	Human / Radiation	No Hair	203.0	Marvel Comics	-	bad	441.0
4	4	Abraxas	Male	blue	Cosmic Entity	Black	-99.0	Marvel Comics	-	bad	-99.0

Right away we can see we have the column 'Unnamed: 0' that we can most likely safely drop. That leaves us with 10 total variables.

We can also see in the Skin Color column that we are missing some values, bringing us to our next question.

Do I have missing data?

We already saw we have some missing data, but let's check for the sum of null values for each variable.

```
#check for null values
info_df.isnull().sum()

name           0
Gender         0
Eye color      0
Race           0
Hair color     0
Height         0
Publisher     15
Skin color     0
Alignment      0
Weight         2
dtype: int64
```

While this shows there is missing data, it may be somewhat misleading.

Above we saw the skin color column contains dash values, which Python technically does not interpret as null values. This shows that visual inspection of the data is important. We can clean up the dashes and replace with NaN:

```
info_df['Skin color'].replace('-', np.nan, inplace=True)
```

After cleaning up our data we can move on to the next question.

What types of variables do I have?

Variables can be one of two types: categorical or numerical.

Categorical Data

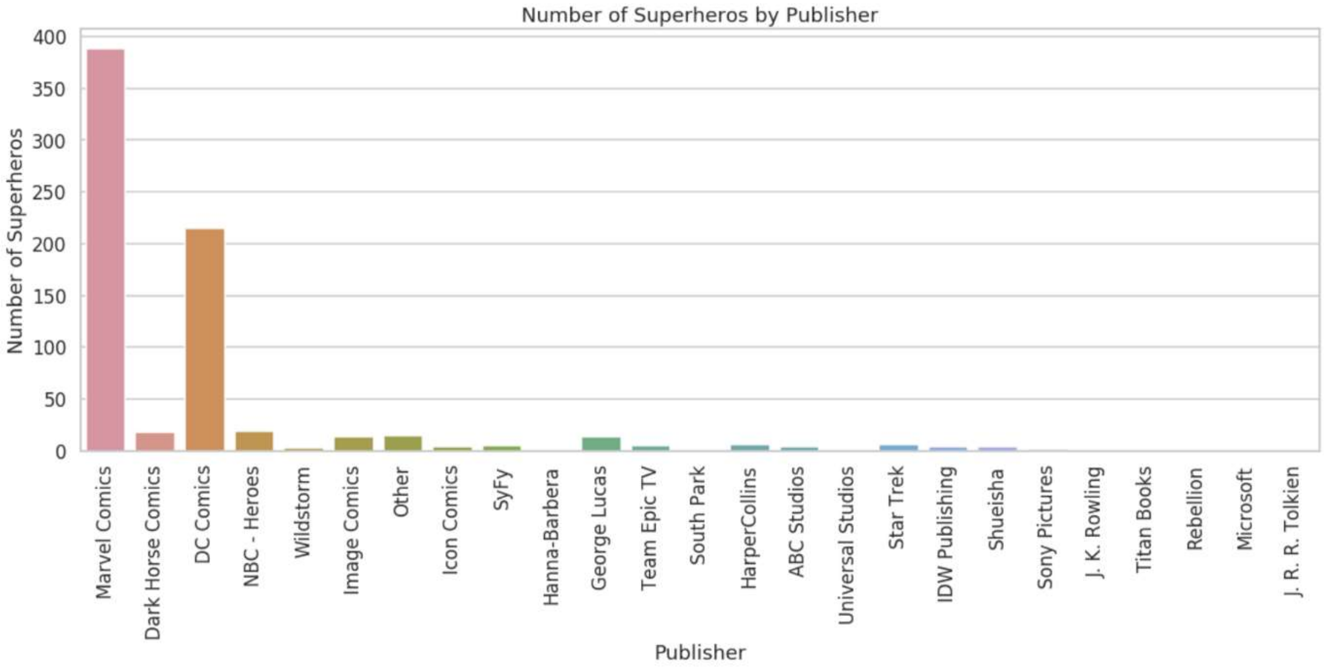
Categorical data classify items into groups. This type of data can be further broken down into nominal, ordinal, and binary values.

- Ordinal values have a set order. An example here could be a ranking of low to high.
- Nominal values have no set order. Examples include the Super Hero's gender and alignment.
- Binary data has only two values. This could be represented as True/False or 1/0.

A common way to summarize categorical variables is with a frequency table. To visualize we will use a bar chart.

```
sns.countplot(x='Publisher', data='info_df')

plt.title('Number of Superheros by Publisher')
plt.ylabel('Number of Superheros')
plt.xlabel('Publisher')
plt.xticks(rotation = 90)
plt.show();
```



Barchart visualizing the number of Super Heroes by publisher.

Here we can see that Marvel Comics has the greatest number of Super Heroes followed by DC Comics.

Numerical Data

Numerical data are values that we can perform mathematical operations on. They are further broken down into continuous and discrete data types.

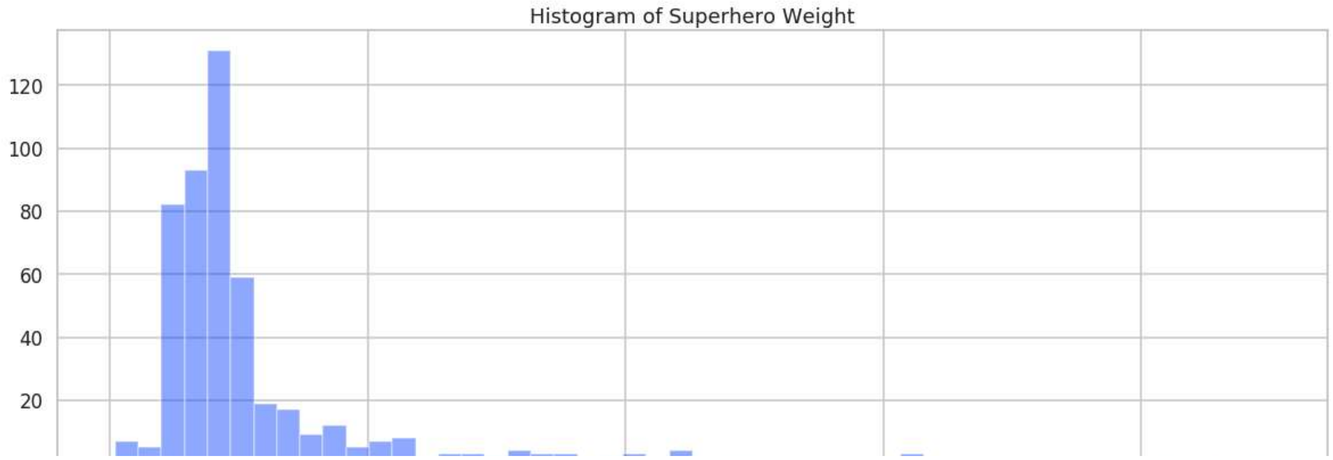
- Discreet variables have to be an integer. An example is number of Super Heroes.
- Continuous can be any value. Examples here include height and weight.

Numerical data can be visualized with a histogram. Histograms are a great first analysis of continuous data. Four main aspects to consider here are shape, center, spread, and outliers.

- Shape is the overall appearance of the histogram. It can be symmetric, skewed, uniform, or have multiple peaks.
- Center refers to the mean or median.
- Spread refers to the range or how far the data reaches.
- Outliers are data points that fall far from the bulk of the data.

```
sns.distplot(info_2.Weight, kde=False)

plt.title('Histogram of Superhero Weight')
plt.show();
```



Histogram of superhero weight.

From the histogram we can see that most Super Heroes have a weight between about 50 and 150 pounds. We have one peak at about 100 pounds and outliers with weights above 800. We can confirm this by printing a numerical summary with the describe function:

```
info_2.Weight.describe()

count    490.000000
mean     112.179592
std      104.422653
min        4.000000
25%       61.000000
50%       81.000000
75%      106.000000
max      900.000000
Name: Weight, dtype: float64
```

Describe shows us key statistics including mean, standard deviation, and max value. Using summary statistics in addition to our histogram above can begin to give us a good idea of what our data looks like.

Univariate Conclusions

The dataset contains 10 variables and some missing data. We saw that DC Comics has the most Super Heroes and that the weight variable has some outliers.

This is by no means a complete exploratory analysis. We can continue to explore the remaining variables and move on to bivariate analysis.

Something interesting to explore further could be to compare Marvel and DC Comics. Can we use data science to determine the superior universe?

Data Science   Eda   Seaborn   Univariate   Data Analysis