Histograms

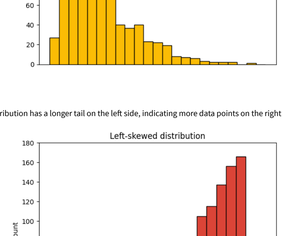
As you've been learning, the purpose of exploratory data analysis (EDA) is just what its name says: explore and analyze the data. As a data professional, you'll almost always begin with a guiding question or objective, such as, "Where are the highest emitters of carbon dioxide located?" or "Determine the characteristics of people most likely to buy product X." Reflecting on this often throughout your process creates a driving force that keeps you on track.

One of the most important tools at your disposal when exploring data is the **histogram**. A histogram is a graphical representation of a frequency distribution, which shows how frequently each value in a dataset or variable occurs. It's essential for data professionals to understand the distributions of their data, because this knowledge drives many downstream decisions around experiment design, modeling, and further analysis. In this reading, you'll learn about histograms, what they are, how to make them, and how to interpret them.

Introduction to histograms

Histograms are commonly used to illustrate the shape of a distribution, including the presence of any outliers, the center of the distribution, and the spread of the data. Histograms are typically represented by a series of bars, where each bar represents a range of values. Bar height represents the frequency or count of the data points within that range.

The following example is a histogram of the number of seconds between eruptions of the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.



The x-axis represents the number of seconds between eruptions. The y-axis represents the eruption count. So, as indicated by the second bar in the graph, there are 20 eruptions that occurred after a wait time of 45-49 seconds.

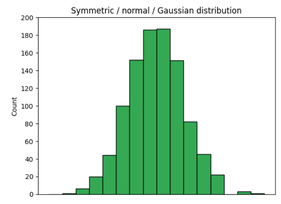
The importance of histograms

Histograms are an essential tool for understanding the characteristics of a dataset. They provide a visual representation of the data's distribution and enable data professionals to identify patterns, trends, or outliers within the data. Histograms can also help data professionals choose appropriate statistical tests and models for the data and determine whether the data meets any assumptions required for the analysis. Histograms are widely used in any field and any situation that requires any kind of data analysis, including finance, health care, engineering, and social sciences.

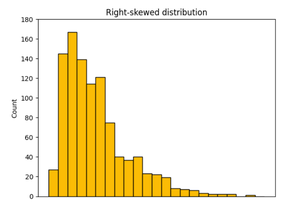
How to interpret histograms

Interpreting histograms involves understanding the shape, center, and spread of the distribution. There are several common shapes of histograms, including:

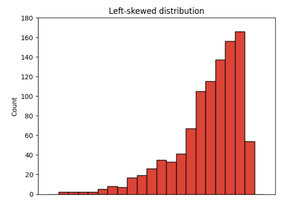
1. **Symmetric:** A symmetric histogram has a bell-shaped curve with a peak in the middle, indicating that the data is evenly distributed around the mean. This is also known as a normal, or Gaussian, distribution.



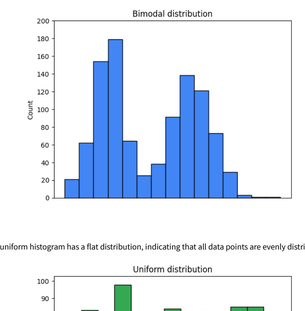
2. **Skewed:** A skewed histogram has a longer tail on one side than the other. A **right-skewed** histogram has a longer tail on the right side, indicating that there are more data points on the left side of the histogram.



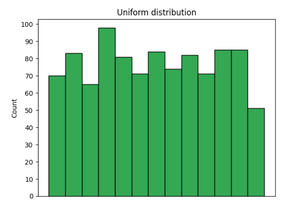
A **left-skewed** distribution has a longer tail on the left side, indicating more data points on the right side.



3. **Bimodal:** A bimodal histogram has two distinct peaks, indicating that the data has two modes.



4. **Uniform:** A uniform histogram has a flat distribution, indicating that all data points are evenly distributed.



The examples provided are not the only distributions you'll encounter, but they are some of the most common. Soon, you will learn more about distributions.

Now, return to the Old Faithful geyser histogram at the beginning of this reading. Ask yourself: what type of distribution is represented by that graph? In addition to the shape, it's important to understand the center and spread. The center of the distribution is typically represented by the mean or median, while the spread is represented by the standard deviation or range of the data. The center and spread can provide insights into data concentration and variability.

How to create histograms

Python's seaborn and matplotlib libraries provide simple and powerful options to create histograms.

plt.hist(x, bins=10, ...)

To generate a histogram in matplotlib, use the `hist()` function in the pyplot module. The function can take many different arguments, but the primary ones are:

- **x:** A sequence of values representing the data you want to plot. It can be a list, tuple, NumPy array, pandas series, and so on.
- **bins:** The number of bins you want to sort your data into. The default value is 10, but this parameter can be an int, sequence, or string. If you use a sequence, it defines the bin edges, including the left edge of the first bin and the right edge of the last bin. In other words, if bins = [1, 3, 5, 7], then the first bin is [1–3) (including 1, but excluding 3) and the second [3–5). The last bin, however, is [5–7], which includes 7. A string refers to a predefined binning strategy supported by numpy. Refer to the documentation for more information.

The following example demonstrates how to generate the Old Faithful geyser histogram from the beginning of this reading using the `plt.hist()` function.

```
# Plot histogram with matplotlib pyplot
plt.hist(df['seconds'], bins=range(40, 101, 5))
plt.xticks(range(35, 101, 5))
plt.yticks(range(0, 61, 10))
plt.xlabel('seconds')
plt.ylabel('count')
plt.title('Old Faithful geyser - time between eruptions')
plt.show();
```

Run
Reset

In this case, the data being plotted is the seconds column of the dataframe. The bins begin at 40 seconds and go to 100 seconds in steps of five, for a total of 12 bins.

sns.histplot(x, bins, binrange, binwidth ...)

One way to generate a histogram in seaborn is to use the `sns.histplot()` function. Like the matplotlib function, `sns.histplot()` can take many arguments. Here are some important ones:

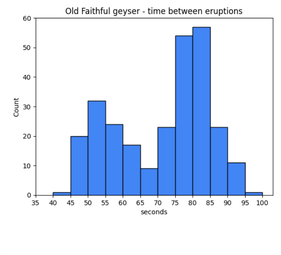
- **x:** The data sequence. Same as `plt.hist()`
- **bins:** Same as `plt.hist()`
- **binrange:** Lowest and highest value for bin edges; can be used either with `bins` or `binwidth`; defaults to data extremes
- **binwidth:** Width of each bin, overrides `bins` but can be used with `binrange`

The following example is the code used to generate the Old Faithful geyser histogram using the seaborn `histplot()` function. It uses all of the previously mentioned parameters. Run this code block to generate a histogram.

Notice in this case that `binrange` is defined from 40 to 100 and `binwidth` is set to 5. This produces the same results as setting `bins=range(40, 101, 5)`. This example also makes use of a couple of style parameters by specifying a particular color using hex code notation and setting the color saturation level to 100%, as indicated by the `alpha` parameter.

Note: The following code block is not interactive.

```
# Plot histogram with seaborn
ax = sns.histplot(df['seconds'], binrange=(40, 100), binwidth=5, color='#4285F4', alpha=1)
ax.set_xticks(range(35, 101, 5))
ax.set_yticks(range(0, 61, 10))
plt.title('Old Faithful geyser - time between eruptions')
plt.show();
```



Key takeaways

Histograms help data professionals understand the frequency distributions of their dataset and variables. Knowledge of the shape and type of data distribution will affect important downstream decisions, such as statistical tests and model architecture selection. Additionally, knowing the shape of your data gives valuable insights into the story that your data is telling you by helping you understand its distributional trends.

Mark as completed

Like Dislike Report an issue