How can I create a distribution plot in Matplotlib?

Authored by

stats writer

June 25, 2024

Creating a distribution plot in Matplotlib is a simple process that allows you to visualize the distribution of data in a graphical form. To create a distribution plot, you first need to import the Matplotlib library into your Python code. Then, you can use the "hist" function to plot a histogram of the data and the "show" function to display the plot. You can also customize the plot by adding labels, titles, and changing the color and style of the plot. Overall, creating a distribution plot in Matplotlib is an effective way to visualize and analyze your data distribution.

Create a Distribution Plot in Matplotlib

There are two common ways to create a distribution plot in Python:

Method 1: Create Histogram Using Matplotlib

import matplotlib.pyplotas plt

plt.hist(data, color='lightgreen', ec='black', bins=15)

Note that color controls the fill color of the bars, ec controls the edge color of the bars and bins controls the number of bins in the histogram.

Method 2: Create Histogram with Density Curve Using Seaborn

import seaborn as sns
sns.displot(data, kde=True, bins=15)

Note that kde=True specifies that a density curve should be overlaid on the histogram.

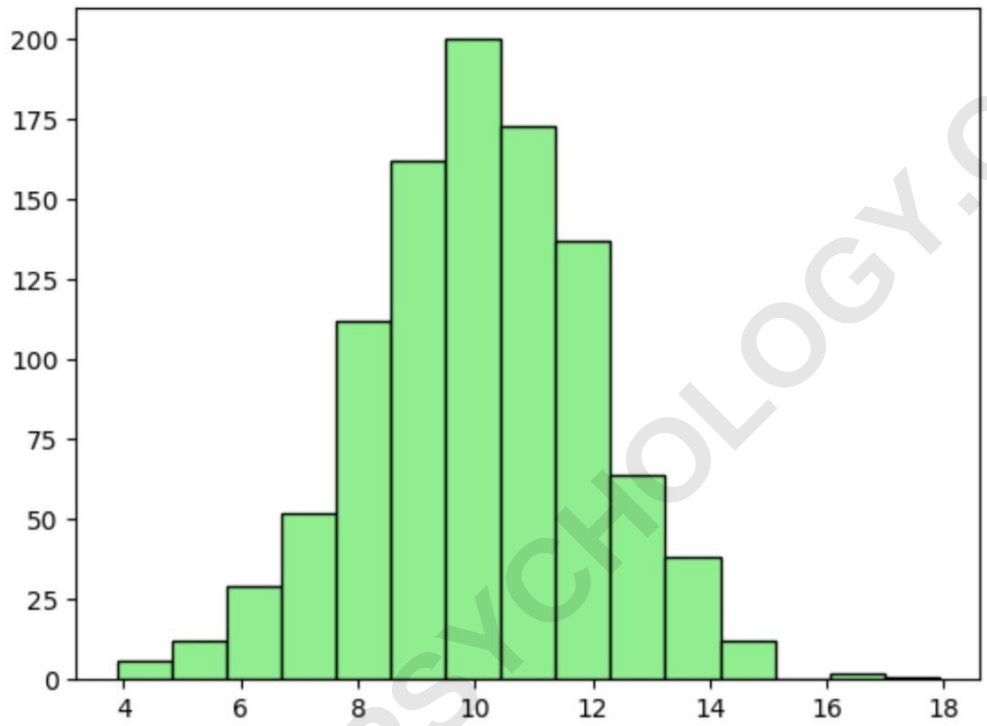
The following examples show how to use each method in practice to visualize the distribution of values in the following NumPy array:

```
import numpy as np

#make this example reproducible.
np.random.seed(1)

#create numpy array with 1000 values that follow normal dist with mean=10 and sd=2
data = np.random.normal(size=1000, loc=10, scale=2)

#view first five values
data

array()
```

Example 1: Create Histogram Using Matplotlib

We can use the following code to create a histogram in Matplotlib to visualize the distribution of values in the NumPy array:

```
import matplotlib.pyplotas plt

#create histogram
plt.hist(data, color='lightgreen', ec='black', bins=15)
```



The x-axis displays the values from the NumPy array and the y-axis displays the frequency of those values.

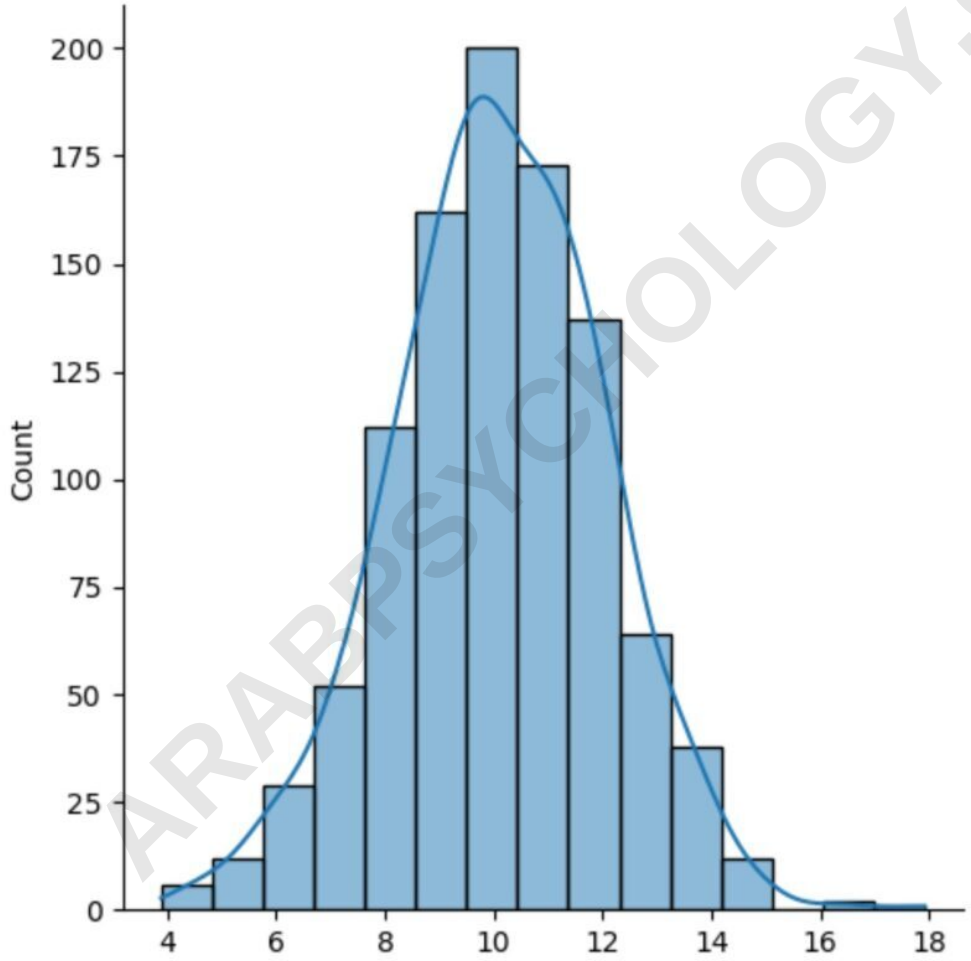
Note that the larger the value you use for the bins argument, the more bars there will be in the histogram.

Example 2: Create Histogram with Density Curve Using Seaborn

We can use the following code to create a histogram

with a overlaid on it using the seaborn data visualization library:

```
import seaborn as sns
#create histogram with density curve overlaid
sns.displot(data, kde=True, bins=15)
```



The result is a histogram with a density curve overlaid on it.

The benefit of using a density curve is that it summarizes the shape of the distribution using a single continuous curve.

Note: You can find the complete documentation for the seaborn displot() function .

The following tutorials explain how to create other common charts in Python: