

How can I create a relative frequency histogram in Matplotlib?

Authored by
stats writer

June 29, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I create a relative frequency histogram in Matplotlib?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=159263>

A relative frequency histogram is a graphical representation of data that shows the proportion of observations falling within certain categories. To create a relative frequency histogram in Matplotlib, one must first import the necessary library and data. Then, the data can be grouped into bins and the frequency of each bin can be calculated. Finally, using Matplotlib's histogram function, the data can be plotted with the frequency on the y-axis and the bins on the x-axis. By dividing the frequency by the total number of observations, the relative frequency can be obtained, giving a visual representation of the distribution of the data.

Create a Relative Frequency Histogram in Matplotlib

A is a graph that displays the relative frequencies of values in a dataset.

You can use the following syntax to create a relative frequency histogram in Matplotlib in Python:

```
import matplotlib.pyplot as plt
import numpy as np
#define plotting area
fig = plt.figure()
ax = fig.add_subplot(111)

#create relative frequency histogram
ax.hist(data,          edgecolor='black',
weights=np.ones_like(data) / len(data))
```

The following example shows how to use this syntax in practice.

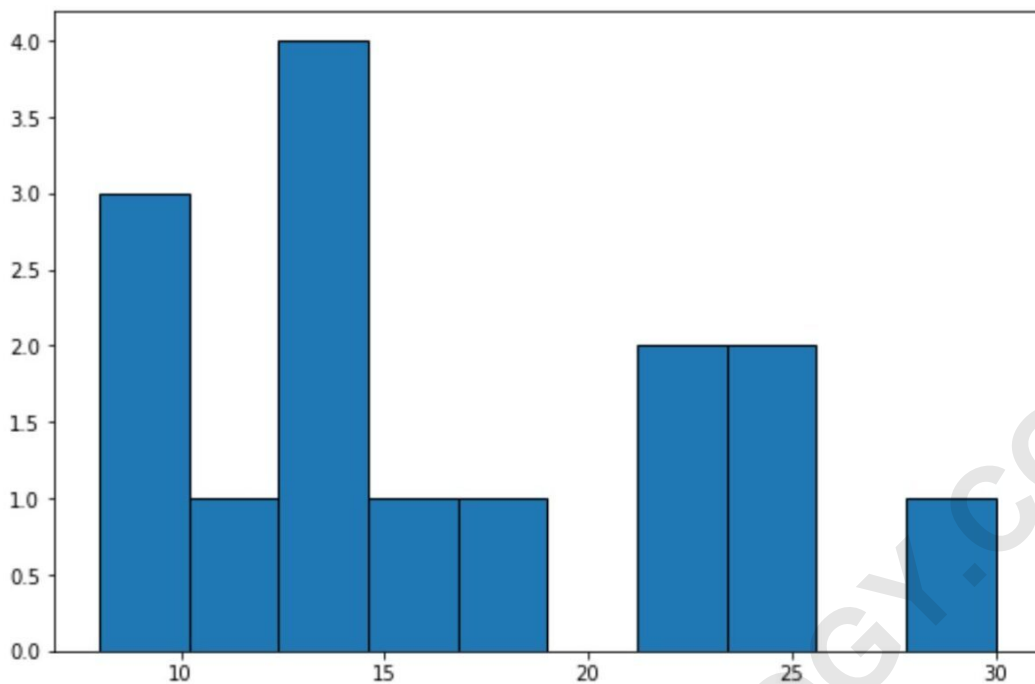
Example: Create Relative Frequency Histogram in Matplotlib

The following code shows how to create a regular frequency histogram in Matplotlib:

```
import numpy as np
import matplotlib.pyplot as plt

#define data values
data =

#create frequency histogram
fig = plt.figure()
ax = fig.add_subplot(111)
ax.hist(data, edgecolor='black')
```



The x-axis shows the bins for the data values and the y-axis shows the frequency for each bin.

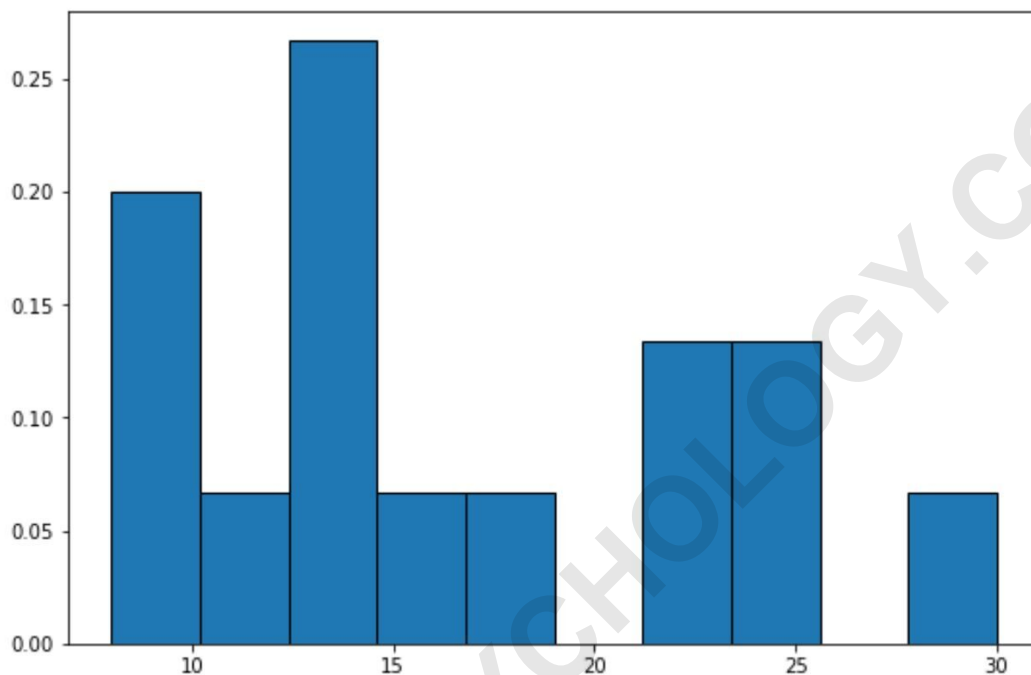
We can use the following code to instead display relative frequencies on the y-axis:

```
import numpy as np
import matplotlib.pyplot as plt

#define data values
data =

#create relative frequency histogram
fig = plt.figure()
```

```
ax = fig.add_subplot(111)
ax.hist(data, edgecolor='black',
weights=np.ones_like(data) / len(data))
```



The y-axis now displays relative frequencies.

For example, there are 15 total values in the dataset.

So instead of showing a frequency of 4 on the y-axis for the tallest bar in the graph, the y-axis now shows $4/15 = 0.2667$.

We can also use the `PercentFormatter()` function from Matplotlib to display the values on the y-axis as

percentages:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from matplotlib.ticker import PercentFormatter
```

```
#define data values
```

```
data =
```

```
#create relative frequency histogram with percentages  
on y-axis
```

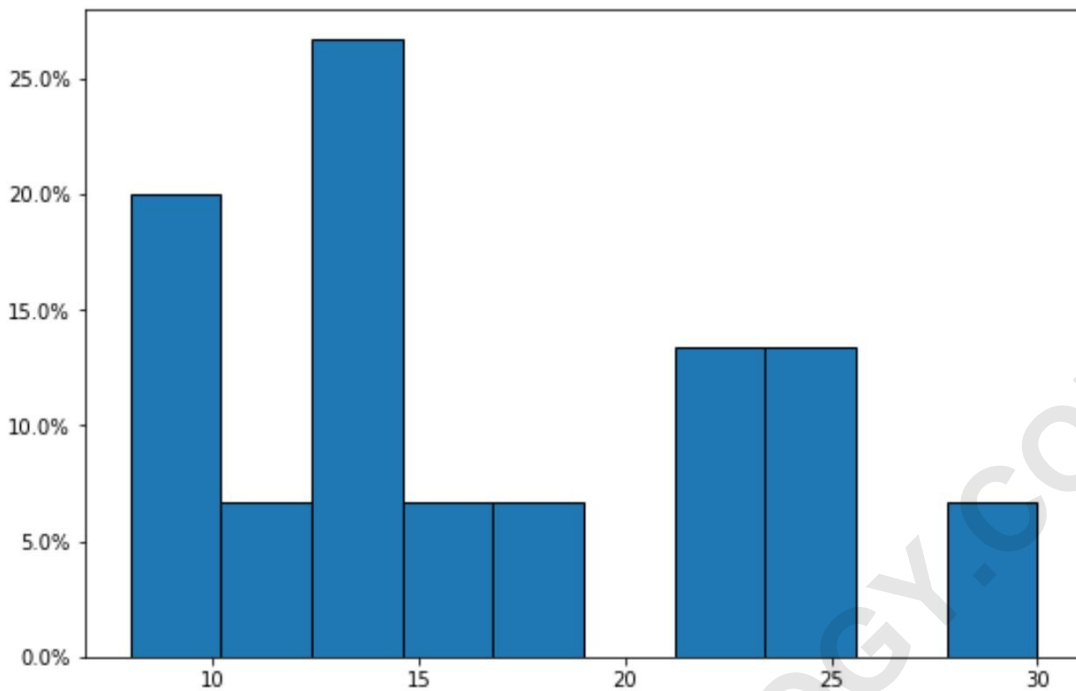
```
fig = plt.figure()
```

```
ax = fig.add_subplot(111)
```

```
ax.hist(data, edgecolor='black',
```

```
weights=np.ones_like(data)*100 / len(data))
```

```
ax.yaxis.set_major_formatter(PercentFormatter())
```



Notice that the y-axis now displays the relative frequencies as percentages.

Additional Resources

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