How can I create a histogram for each column in a Pandas DataFrame?

Authored by

stats writer

June 25, 2024

To create a histogram for each column in a Pandas DataFrame, one can use the built-in function "hist()" which takes in a DataFrame and plots a histogram for each numerical column. This function can be further customized by specifying the number of bins and the range of values to be included in the histogram. Additionally, one can also use the "plot(kind='hist')" method to create a histogram for each column in a DataFrame. This method allows for more customization options such as adding labels, titles, and changing the color and style of the histogram. Overall, creating a histogram for each column in a Pandas DataFrame is a simple and effective way to visualize the distribution of data within a dataset.

Pandas: Create Histogram for Each Column in DataFrame

You can use the following basic syntax to create a histogram for each column in a pandas DataFrame:

```
import pandas as pd
import matplotlib.pyplotas plt

#define number of subplots
fig, axis = plt.subplots(1, 3)#create histogram for each column in DataFrame
df.hist(ax=axis)
```

This particular example uses the subplots() function to specify that there are 3 columns in the DataFrame and then creates a histogram for each column.

The following example shows how to use this syntax in

practice.

Example: Create Histogram for Each Column in Pandas Histogram

Suppose we have the following pandas DataFrame that contains three columns:

```
import pandas as pd
import numpy as np

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

#create DataFrame
df = pd.DataFrame({'points': np.random.normal(loc=20, scale=2, size=300),
'assists': np.random.normal(loc=14, scale=3, size=300),
'rebounds': np.random.normal(loc=12, scale=1, size=300)})

#view head of DataFrame
print(df.head())

points assists rebounds
0 23.248691 20.197350 10.927036
1 18.776487 9.586529 12.495159
```

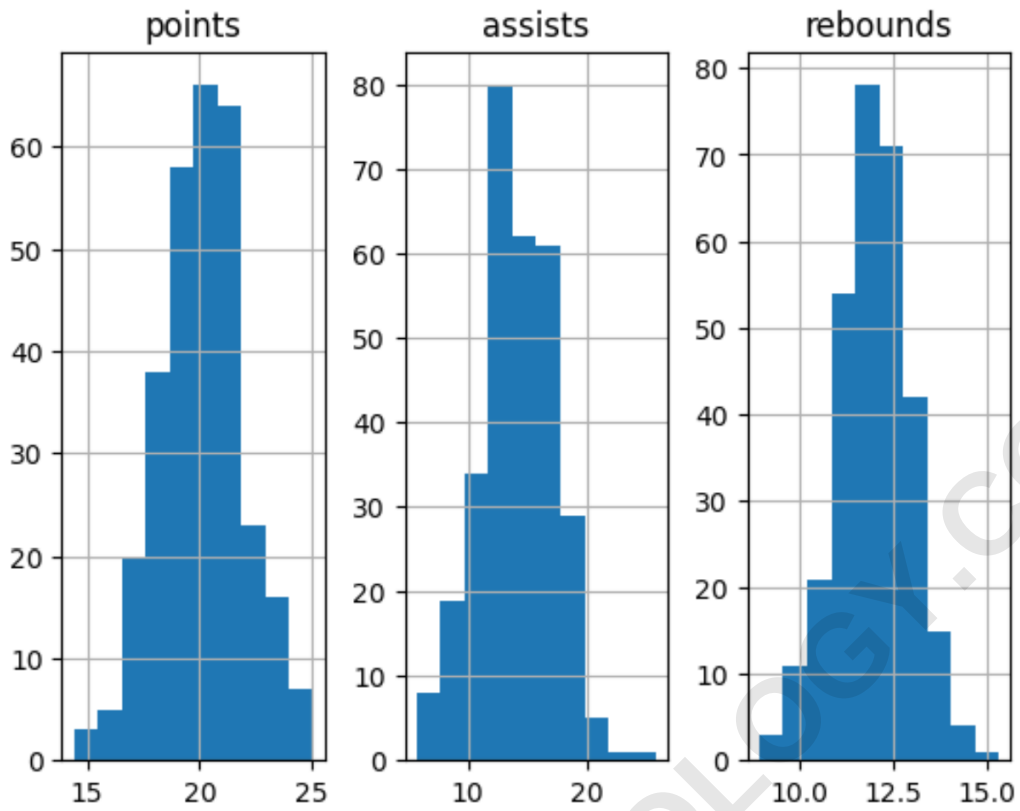
2 18.943656 11.509484 11.047938
3 17.854063 11.358267 11.481854
4 21.730815 13.162707 10.538596

We can use the following syntax to create a histogram for each of the three columns in the DataFrame:

```
import matplotlib.pyplot as plt

#define format for subplots (1 row and 3 columns)
fig, axis = plt.subplots(1, 3)

#create histogram for each column in DataFrame
df.hist(ax=axis)
```



The result is a grid with one row and three columns that displays a histogram for each column in the DataFrame.

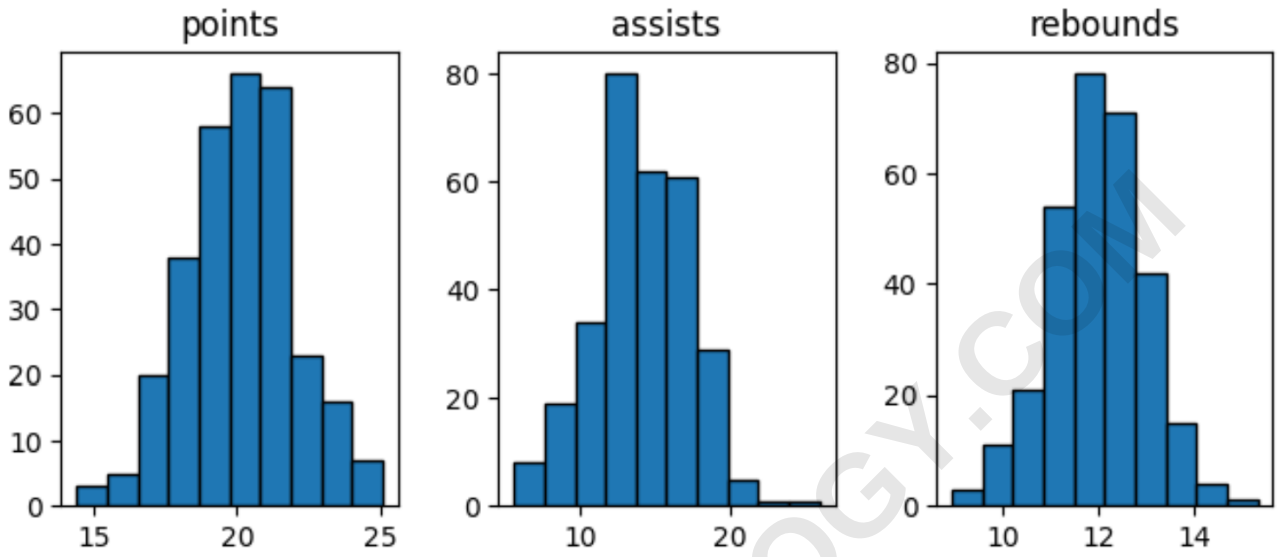
If you'd like, you can use the figsize argument to modify the size of the histograms along with the edgecolor and grid arguments to make the histograms look better:

```
import matplotlib.pyplot as plt

#define format for subplots
fig, axis = plt.subplots(1, 3, figsize=(8,3))

#create histogram for each column in DataFrame
```

df.hist(ax=axis, edgecolor='black', grid=False)



Feel free to play around with the arguments in the subplots() function to define the exact format and size of the histograms.

The following tutorials explain how to perform other common tasks in pandas: