

How can I plot histograms by group in Pandas?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I plot histograms by group in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=154393>

Pandas is a popular data analysis library in Python that offers various tools for visualizing data. One of the commonly used visualizations is histograms, which display the distribution of a numerical variable. In Pandas, it is possible to plot histograms by group, where the data is grouped by a categorical variable and a separate histogram is plotted for each group. This can be achieved by using the "groupby" function to group the data and then using the "hist" function to plot the histograms. This allows for easy comparison of the distribution of the numerical variable across different groups. Additionally, various customization options are available in Pandas to enhance the visual appearance of the histograms.

Plot Histograms by Group in Pandas

You can use the following methods to plot histograms by group in a pandas DataFrame:

Method 1: Plot Histograms by Group Using Multiple Plots

```
df.hist(by=df)
```

Method 2: Plot Histograms by Group Using One Plot

```
plt.hist(group1, alpha=0.5,  
label='group1')plt.hist(group2, alpha=0.5,  
label='group2')plt.hist(group3, alpha=0.5,  
label='group3')
```

The following examples show how to use each method in practice with the following pandas DataFrame that

shows the points scored by basketball players on three different teams:

```
import pandas as pd
```

```
import numpy as np
```

```
#make this example reproducible
```

```
np.random.seed(1)
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': np.repeat(, 100),
```

```
'points': np.random.normal(loc=20, scale=2, size=300)}))
```

```
#view head of DataFrame
```

```
print(df.head())
```

```
team points
```

```
0 A 23.248691
```

```
1 A 18.776487
```

```
2 A 18.943656
```

```
3 A 17.854063
```

```
4 A 21.730815
```

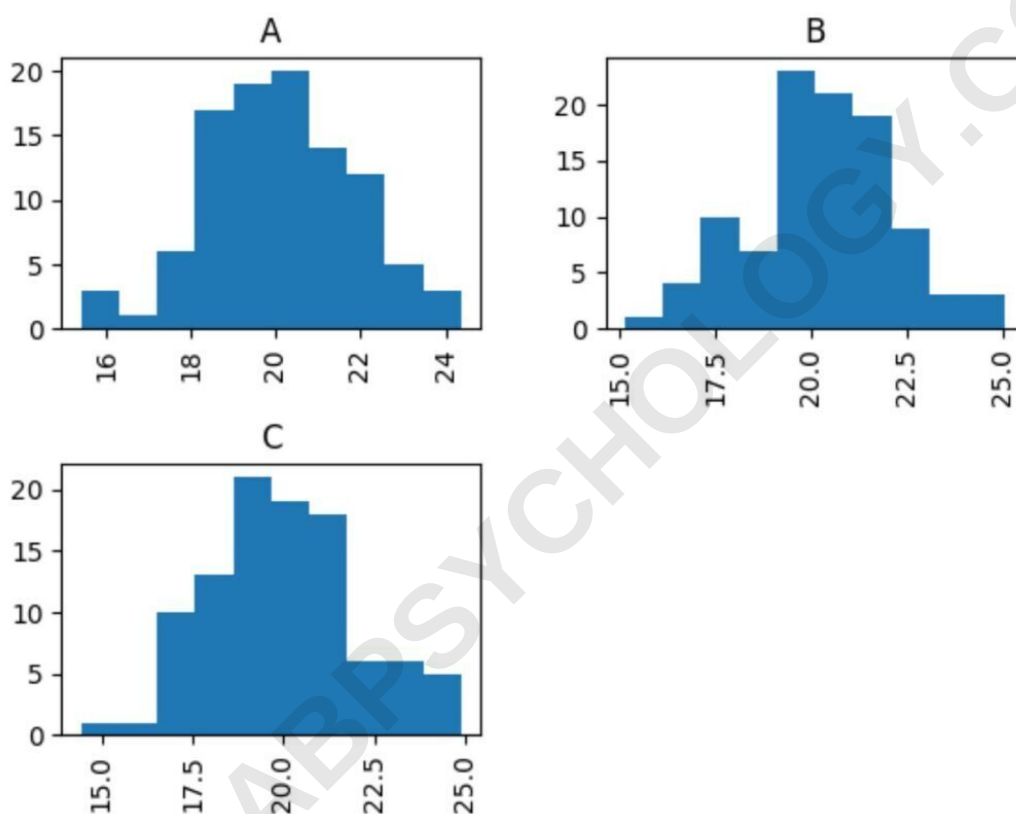
Example 1: Plot Histograms by Group Using Multiple Plots

The following code shows how to create three

histograms that display the distribution of points scored by players on each of the three teams:

#create histograms of points by team

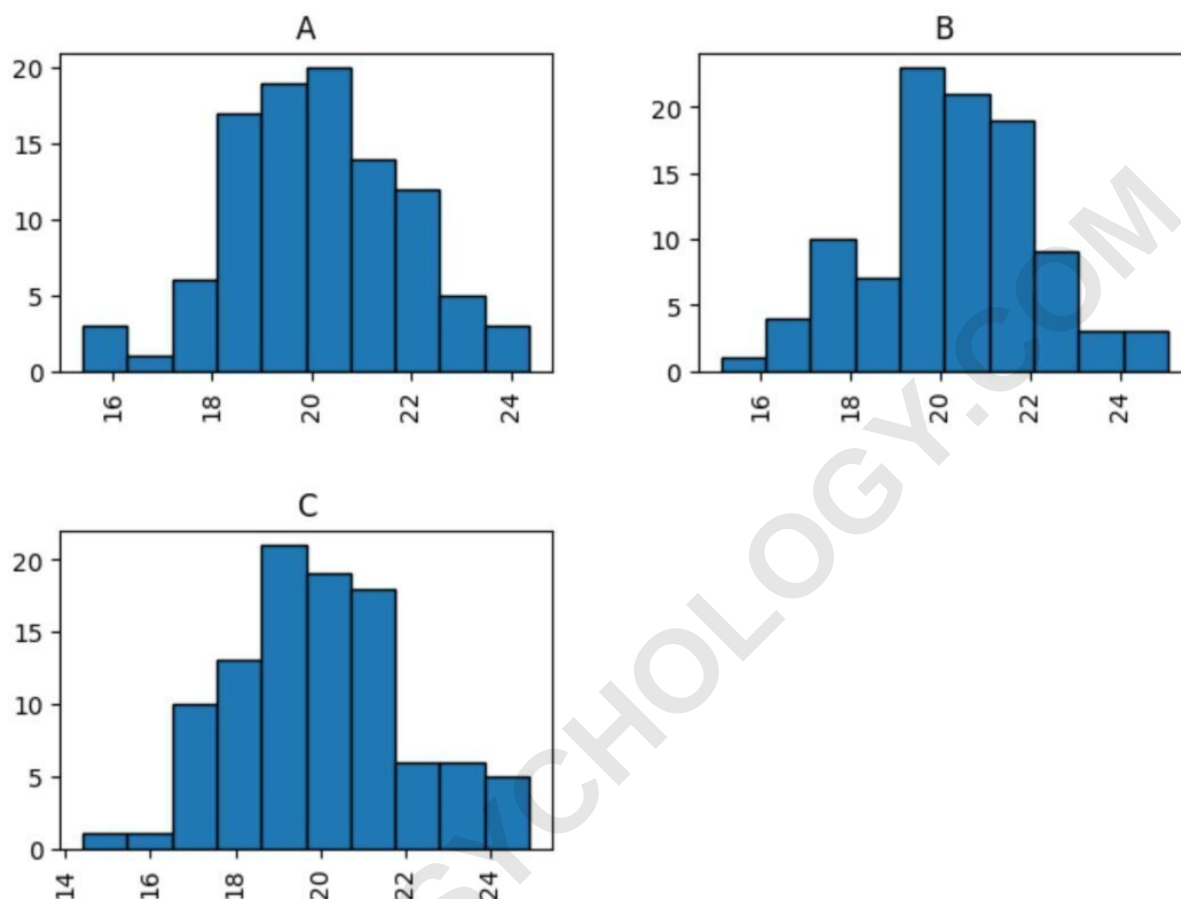
df.hist(by=df)



We can also use the `edgecolor` argument to add edge lines to each histogram and the `figsize` argument to increase the size of each histogram to make them easier to view:

#create histograms of points by team

```
df.hist(by=df, edgecolor='black', figsize = (8,6))
```



Example 2: Plot Histograms by Group Using One Plot

The following code shows how to create three histograms and place them all on the same plot:

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

```
#define points values by group
```

```
A = df.loc == 'A', 'points']
```

```
B = df.loc == 'B', 'points']
```

```
C = df.loc == 'C', 'points']
```

```
#add three histograms to one plot
```

```
plt.hist(A, alpha=0.5, label='A')
```

```
plt.hist(B, alpha=0.5, label='B')
```

```
plt.hist(C, alpha=0.5, label='C')
```

```
#add plot title and axis labels
```

```
plt.title('Points Distribution by Team')
```

```
plt.xlabel('Points')
```

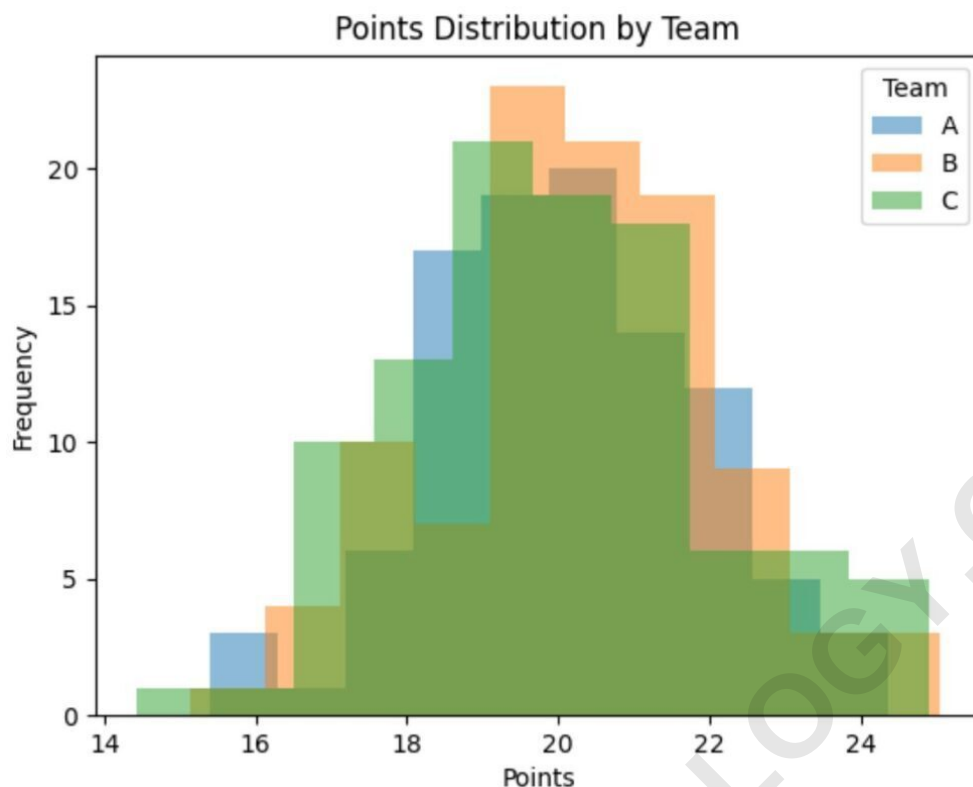
```
plt.ylabel('Frequency')
```

```
#add legend
```

```
plt.legend(title='Team')
```

```
#display plot
```

```
plt.show()
```



The end result is one plot that displays three overlaid histograms.

Note: The alpha argument specifies the transparency of each histogram. This value can range from 0 to 1. By setting this value equal to 0.5, we're able to better view each overlaid histogram.

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