

How to Easily Label Data Points in Your Pandas Scatter Plot

Authored by
stats writer

November 20, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Label Data Points in Your Pandas Scatter Plot*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=98574>

In a Pandas scatter plot, each data point can be labeled by using the label argument in the plot function. This argument takes a list of strings, which will be used to label each point in the plot. When the plot is displayed, each data point will be labeled with its corresponding string from the list. Additionally, the label argument can be combined with the colors argument to apply a different color to each data point based on its label.

You can use the following basic syntax to label the points in a pandas scatter plot:

#create scatter plot of x vs. y

```
ax = df.plot(kind='scatter', x='x_var', y='y_var')
```

#label each point in scatter plot

```
for idx, row in df.iterrows():
```

```
ax.annotate(row, (row, row))
```

This particular example creates a scatter plot of the values in the columns **x_var** and **y_var** and then uses the values in the column called **label_var** as the labels for the points.

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

Example: How to Label Points in Pandas Scatter Plot

Suppose we have the following pandas DataFrame that shows the points and assists for basketball players on various teams:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'team': ,
```

```
'assists': ,
```

```
'points': })
```

```
#view DataFrame
```

```
print(df)
```

```
team assists points
```

```
0 A 3 7
```

```
1 B 4 9
```

```
2 C 4 14
```

```
3 D 5 13
```

```
4 E 5 10
```

5 F 6 11
6 G 7 12
7 H 7 13

We can use the following syntax to create a scatter plot of **assists** vs. **points** and use the values in the **team** column as labels:

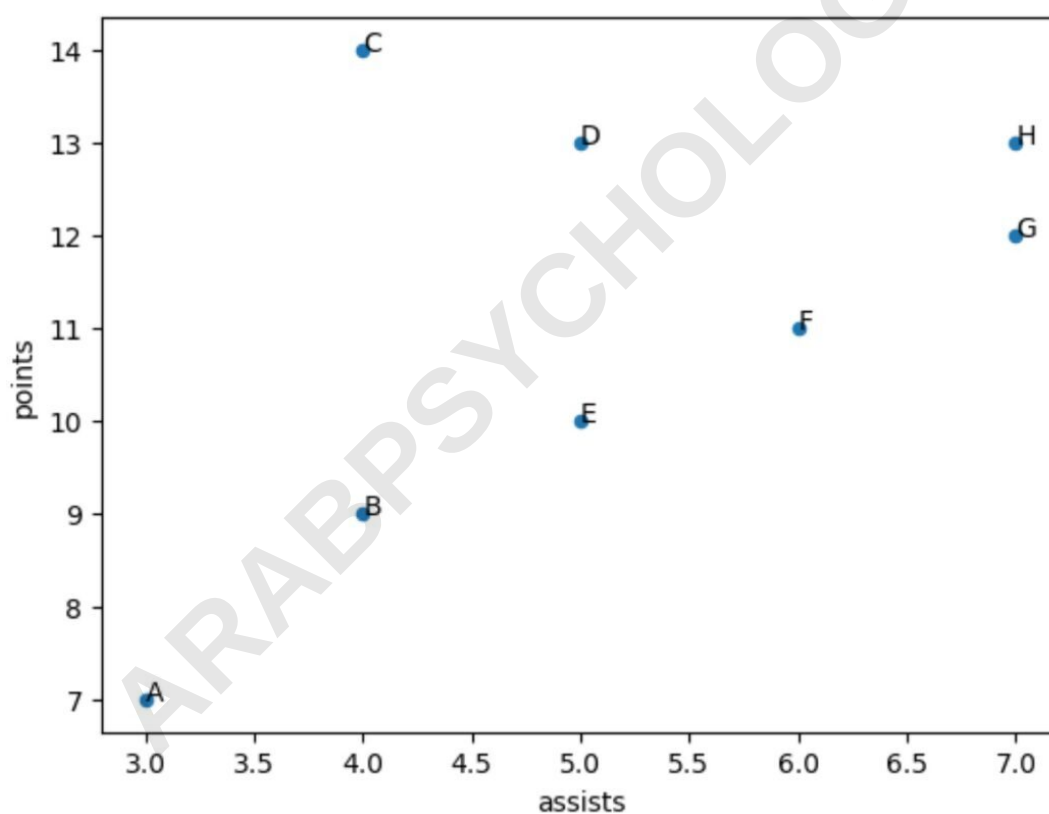
```
#create scatter plot of assists vs. points
```

```
ax = df.plot(kind='scatter', x='assists', y='points')
```

```
#label each point in scatter plot
```

```
for idx, row in df.iterrows():
```

```
ax.annotate(row, (row, row))
```



Notice that each of the points in the plot have a label that tells us which **team** each player is on.

Note that we can also use the following arguments in the **annotate()** function to modify the appearance of the text labels:

xytext: The coordinates to use to offset the text

textcoords: How to offset the text

family: The font family to use for the text

fontsize: The font size to use for the text

The following code shows how to use these arguments in practice to modify the appearance of the text labels:

```
#create scatter plot of assists vs. points
```

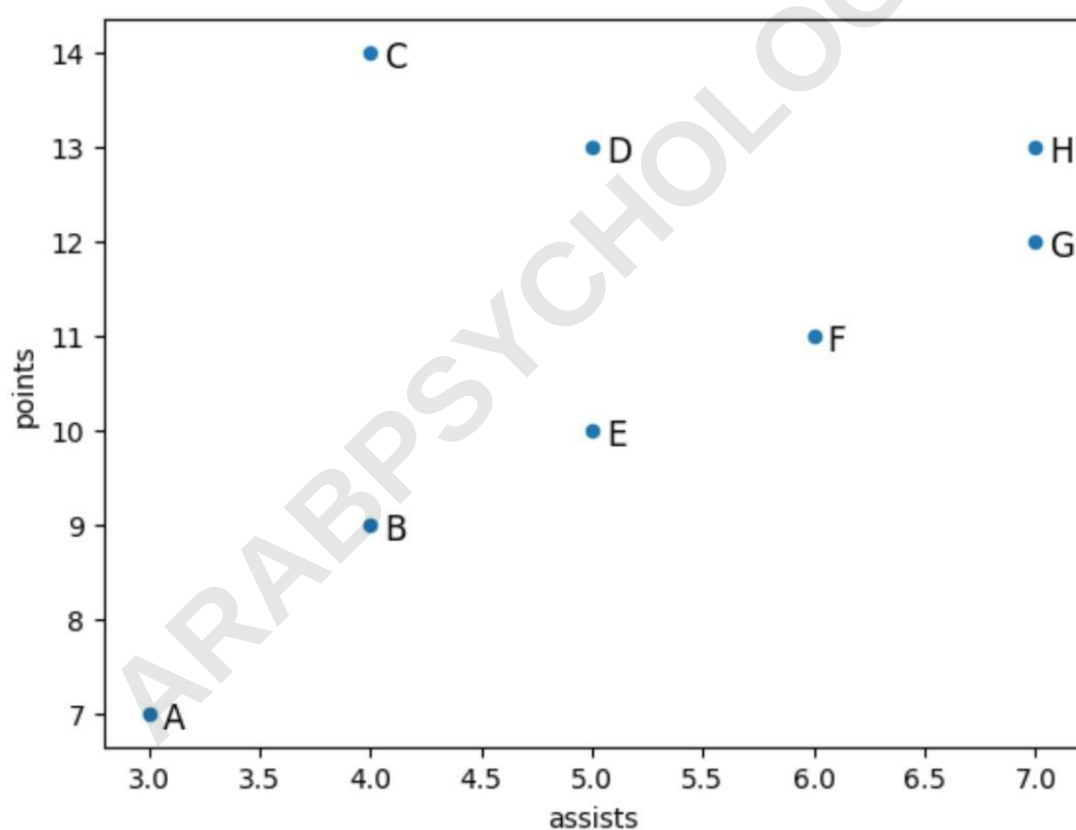
```
ax = df.plot(kind='scatter', x='assists', y='points')
```

```
#add custom label to each point in scatter plot
```

```
for idx, row in df.iterrows():
```

```
ax.annotate(row, (row, row), xytext=(5,-5),
```

```
textcoords='offset points', family='sans-serif', fontsize=12)
```



Notice how the labels are now offset from the points so they're easier to read, they have a larger font size, and use sans-serif as the font family to make them more aesthetically pleasing.