How can a table be created using Matplotlib?

Authored by

stats writer

April 22, 2024

A table can be created using Matplotlib by first importing the necessary library, then defining the data to be included in the table. Next, the table can be created by calling the .table() function and passing in the data as a list or array. Additional formatting options such as row and column labels, cell colors, and borders can be applied using various parameters. Finally, the table can be displayed using the .show() function. This process allows for the creation of customizable and visually appealing tables within a Matplotlib figure.

Create a Table with Matplotlib

You can use one of the two following methods to create tables in Python using Matplotlib:

Method 1: Create Table from pandas DataFrame

#create pandas DataFrame
df = pd.DataFrame(np.random.randn(20, 2), columns=)

#create table
table = ax.table(cellText=df.values, colLabels=df.columns, loc='center')

Method 2: Create Table from Custom Values

#create values for table
table_data=,
,
,

,

]

#create table
table = ax.table(cellText=table_data, loc='center')

This tutorial provides examples of how to use these methods in practice.

Example 1: Create Table from pandas DataFrame

The following code shows how to create a table in Matplotlib that contains the values in a pandas DataFrame:

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#make this example reproducible
np.random.seed(0)

#define figure and axes
fig, ax = plt.subplots()

#hide the axes

```
fig.patch.set_visible(False)
ax.axis('off')
ax.axis('tight')

#create data
df = pd.DataFrame(np.random.randn(20, 2), columns=)

#create table
table = ax.table(cellText=df.values,
colLabels=df.columns, loc='center')

#display table
fig.tight_layout()
plt.show()
```

First	Second
1.764052345967664	0.4001572083672233
0.9787379841057392	2.240893199201458
1.8675579901499675	-0.977277879876411
0.9500884175255894	-0.1513572082976979
-0.10321885179355784	0.41059850193837233
0.144043571160878	1.454273506962975
0.7610377251469934	0.12167501649282841
0.44386323274542566	0.33367432737426683
1.4940790731576061	-0.20515826376580087
0.31306770165090136	-0.8540957393017248
-2.5529898158340787	0.6536185954403606
0.8644361988595057	-0.7421650204064419
2.2697546239876076	-1.4543656745987648
0.04575851730144607	-0.1871838500258336
1.5327792143584575	1.469358769900285
0.1549474256969163	0.37816251960217356
-0.8877857476301128	-1.980796468223927
-0.3479121493261526	0.15634896910398005
1.2302906807277207	1.2023798487844113
-0.3873268174079523	-0.30230275057533557

Example 2: Create Table from Custom Values

The following code shows how to create a table in Matplotlib that contains custom values:

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

#define figure and axes
fig, ax = plt.subplots()

#create values for table
table_data=,
,
,
,
]

#create table
table = ax.table(cellText=table_data, loc='center')

#modify table
table.set_fontsize(14)
table.scale(1,4)
```

```
ax.axis('off')

#display table
plt.show()
```

Player 1	30
Player 2	20
Player 3	33
Player 4	25
Player 5	12

Note that the table.scale(width, length) modifies the width and length of the table. For example, we could make the table even longer by modifying the length:

```
table.scale(1,10)
```

Player 1	30
Player 2	20
Player 3	33
Player 4	25
Player 5	12

How to Add Text to Matplotlib Plots

How to Set the Aspect Ratio in Matplotlib

How to Change Legend Font Size in Matplotlib