Bokeh

点击一键运行代码→ 获取更多速查表→ 使用说明书[视频 & 图文]→ bilibili Show Me AI

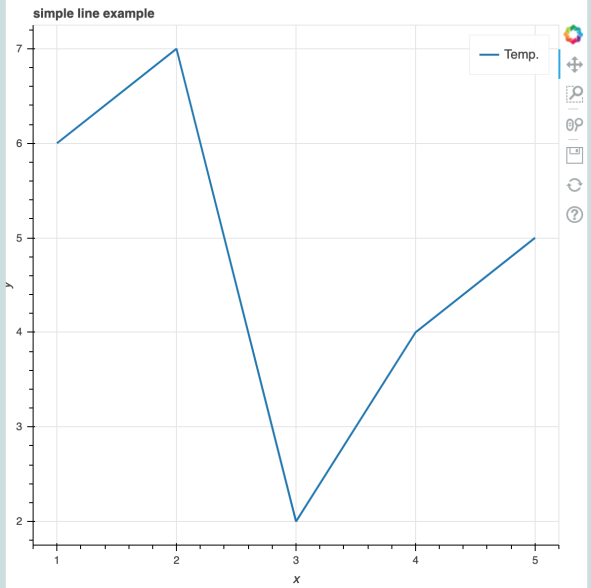
Bokeh 是一款交互式可视化库，可以通过 Python，快速便捷地为大型流数据集提供优雅简洁的高性能交互式图表，在浏览器中呈现。

作者 | 韩信子 @ShowMeAI
设计 | 南 乔 @ShowMeAI
参考 | datacamp cheatsheet

bokeh.plotting 界面绘图基本步骤与示例

Step 1 准备数据（Python 列表、Numpy 数组、Pandas 数据框或其它序列值） Step 2 创建图形
Step 3 为数据添加渲染器，自定义可视化图 Step 4 指定生成的输出类型 Step 5 显示视图或保存结果

```
> from bokeh.plotting import figure
> from bokeh.io import output_file, show
> x = [1, 2, 3, 4, 5]                                    #Step 1
> y = [6, 7, 2, 4, 5]
> p = figure(title="simple line example", x_axis_label='x', y_axis_label='y')   #Step 2
> p.line(x, y, legend_label="Temp.", line_width=2)       #Step 3
> output_file("lines.html")                              #Step 4
> show(p)                                                #Step 5
```



1. 数据准备

通常，Bokeh 在后台把数据转换为列数据源，不过也可手动转换：

```
> import numpy as np
> import pandas as pd
> df = pd.DataFrame(np.array([[33.9,4,65, 'US'],[32.4,4,66, 'Asia'], [21.4,4,109, 'Europe']]),
                    columns=['mpg','cyl', 'hp', 'origin'],
                    index=['Toyota', 'Fiat', 'Volvo'])
> from bokeh.models import ColumnDataSource
> cds_df = ColumnDataSource(df)
```

2. 绘图

figure 函数

```
> from bokeh.plotting import figure
> p1 = figure(plot_width=300, tools='pan,box_zoom')
> p2 = figure(plot_width=300, plot_height=300, x_range=(0, 8), y_range=(0, 8))
> p3 = figure()
```

3. 渲染器与自定义可视化

图示符

散点标记

```
> p1.circle(np.array([1,2,3]), np.array([3,2,1]), fill_color='white')
> p2.square(np.array([1.5,3.5,5.5]), [1,4,3],  color= 'blue' , size=1)
```

线型图示符

```
> p1.line([1,2,3,4], [3,4,5,6], line_width=2)
> p2.multi_line(pd.DataFrame([[1,2,3],[5,6,7]]),  pd.DataFrame([[3,4,5],[3,2,1]]), color="blue")
```

自定义图示符

图示符选择与反选

```
> p = figure(tools='box_select')
> p.circle('mpg', 'cyl', source=cds_df, selection_color='red', nonselection_alpha=0.1)
```

绘图区内部

```
> from bokeh.models import CategoricalColorMapper
> color_mapper = CategoricalColorMapper(factors=['US', 'Asia', 'Europe'], palette=['blue', 'red', 'green'])
> p3.circle('mpg', 'cyl', source=cds_df,color=dict(field='origin', transform=color_mapper), legend_label='Origin')
```

3. 渲染器与自定义可视化

图例位置

绘图区内部

```
> p.legend.location = 'bottom_left'
```

绘图区外部

```
> from bokeh.models import Legend
> r1 = p2.asterisk(np.array([1,2,3]), np.array([3,2,1]))
> r2 = p2.line([1,2,3,4], [3,4,5,6])
> legend = Legend(items=[("One" ,[ r1]),("Two",[r2])], location=(0, -30))
> p.add_layout(legend, 'right')
```

图例方向

```
> p.legend.orientation = "horizontal"
> p.legend.orientation = "vertical"
```

图例背景与边框

```
> p.legend.border_line_color = "navy"
> p.legend.background_fill_color = "white"
```

栅格布局

```
> from bokeh.layouts import gridplot
> row1 = [p1,p2]
> row2 = [p3]
> layout = gridplot([[p1,p2],[p3]])
```

行列布局

行

```
> from bokeh.layouts import row
> layout = row(p1,p2,p3)
```

列

```
> from bokeh.layouts import column
> layout = column(p1,p2,p3)
```

行列嵌套

```
> layout = row(column(p1,p2), p3)
```

标签布局

```
> from bokeh.models.widgets import Panel, Tabs
> tab1 = Panel(child=p1, title="tab1")
> tab2 = Panel(child=p2, title="tab2")
> layout = Tabs(tabs=[tab1, tab2])
```

链接图

链接坐标轴

```
> p2.x_range = p1.x_range
> p2.y_range = p1.y_range
```

链接刷

```
> p4 = figure(plot_width = 100, tools='box_select,lasso_select')
> p4.circle('mpg', 'cyl', source=cds_df)
> p5 = figure(plot_width = 200, tools='box_select,lasso_select')
> p5.circle('mpg', 'hp', source=cds_df)
> layout = row(p4,p5)
```

4. 输出与导出

Notebook

```
> from bokeh.io import output_notebook, show
> output_notebook()
```

HTML

本地 HTML

```
> from bokeh.embed import file_html
> from bokeh.resources import CDN
> html = file_html(p, CDN, "my_plot")

> from bokeh.io import output_file, show
> output_file('my_bar_chart.html', mode='cdn')
```

组件

```
> from bokeh.embed import components
> script, div = components(p)
```

PNG

```
> from bokeh.io import export_png
> export_png(p, filename="plot.png")
```

SVG

```
> from bokeh.io import export_svgs
> p.output_backend = "svg"
> export_svgs(p, filename="plot.svg")
```

5. 显示或保存图形

show 与 save 函数

```
> show(p1)
> save(p1)

> show(layout)
> save(layout)
```