

基本使用

以下是 Matplotlib 的常用模块和基本功能:

1. 导入 Matplotlib

```
python
复制编辑
import matplotlib.pyplot as plt
```

2. 绘制简单折线图

```
python
复制编辑
import matplotlib.pyplot as plt
# 数据
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]
# 创建图形
plt.plot(x, y, label="Prime Numbers", color="blue", linestyle="--",
marker="o")
plt.title("Simple Line Plot") # 标题
plt.xlabel("X Axis")         # X 轴标签
plt.ylabel("Y Axis")         # Y 轴标签
plt.legend()                  # 图例
plt.grid(True)                # 网格线
plt.show()                    # 显示图形
```

常用绘图类型

1. 散点图

```
python
复制编辑
import matplotlib.pyplot as plt
# 数据
x = [1, 2, 3, 4, 5]
y = [2, 4, 1, 3, 6]
# 绘制散点图
plt.scatter(x, y, color="red", label="Data Points")
plt.title("Scatter Plot")
plt.xlabel("X Axis")
```

```
plt.ylabel("Y Axis")
plt.legend()
plt.show()
```

2. 柱状图

```
python
复制编辑
import matplotlib.pyplot as plt
# 数据
categories = ["A", "B", "C", "D"]
values = [3, 7, 5, 9]
# 绘制柱状图
plt.bar(categories, values, color="purple", alpha=0.7)
plt.title("Bar Chart")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```

3. 饼图

```
python
复制编辑
import matplotlib.pyplot as plt
# 数据
labels = ["Apple", "Banana", "Cherry", "Date"]
sizes = [30, 25, 20, 25]
colors = ["red", "yellow", "pink", "brown"]
# 绘制饼图
plt.pie(sizes, labels=labels, colors=colors, autopct="%1.1f%%")
plt.title("Pie Chart")
plt.show()
```

4. 直方图

```
python
复制编辑
import matplotlib.pyplot as pltimport numpy as np
# 随机数据
data = np.random.randn(1000)
# 绘制直方图
plt.hist(data, bins=30, color="green", alpha=0.8)
plt.title("Histogram")
plt.xlabel("Value")
plt.ylabel("Frequency")
```

```
plt.show()
```

自定义图形

Matplotlib 支持丰富的自定义选项：

1. 设置图形大小

```
python  
复制编辑  
plt.figure(figsize=(8, 6))
```

2. 设置颜色、样式、线宽

```
python  
复制编辑  
plt.plot(x, y, color="orange", linestyle="-.", linewidth=2)
```

3. 添加注释

```
python  
复制编辑  
plt.annotate("Max Point", xy=(3, 5), xytext=(4, 6),  
             arrowprops=dict(facecolor="black", arrowstyle="->"))
```

4. 显示网格

```
python  
复制编辑  
plt.grid(True, linestyle="--", alpha=0.7)
```

动画效果

Matplotlib 提供 FuncAnimation 模块实现动画：

```
python  
复制编辑  
from matplotlib.animation import FuncAnimation  
import numpy as np  
import matplotlib.pyplot as plt  
# 数据  
x = np.linspace(0, 2 * np.pi, 100)  
# 初始化绘图
```

```
fig, ax = plt.subplots()
line, = ax.plot([], [], color="blue")
def init():
    ax.set_xlim(0, 2 * np.pi)
    ax.set_ylim(-1, 1)
    return line,
def update(frame):
    y = np.sin(x + frame / 10.0) # 动态更新
    line.set_data(x, y)
    return line,
# 动画
ani = FuncAnimation(fig, update, frames=100, init_func=init, interval=50,
blit=True)
plt.show()
```

Matplotlib 中常用函数及其使用方法

Matplotlib 提供了许多函数，用于数据可视化的各种需求。以下是一些常用的函数及其使用示例：

1. 绘图函数

1.1 plot: 绘制折线图

```
python
复制编辑
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y = [10, 20, 25, 30]

plt.plot(x, y, color="blue", linestyle="--", marker="o", label="Line")
plt.title("Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.show()
```

1.2 scatter: 绘制散点图

python

复制编辑

```
x = [1, 2, 3, 4, 5]
y = [10, 20, 15, 25, 30]
```

```
plt.scatter(x, y, color="red", label="Scatter Points")
plt.title("Scatter Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.legend()
plt.show()
```

1.3 bar: 绘制柱状图

python

复制编辑

```
categories = ["A", "B", "C", "D"]
values = [5, 7, 3, 8]
```

```
plt.bar(categories, values, color="green", alpha=0.7)
plt.title("Bar Chart")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```

1.4 hist: 绘制直方图

python

复制编辑

```
import numpy as np
```

```
data = np.random.randn(1000)
```

```
plt.hist(data, bins=30, color="purple", alpha=0.8)
plt.title("Histogram")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```

1.5 pie: 绘制饼图

```
python
复制编辑
labels = ["Python", "C++", "Java", "Ruby"]
sizes = [40, 30, 20, 10]
colors = ["gold", "yellowgreen", "lightcoral", "lightskyblue"]

plt.pie(sizes, labels=labels, colors=colors, autopct="%1.1f%%",
startangle=140)
plt.title("Pie Chart")
plt.axis("equal") # 确保饼图是圆的
plt.show()
```

1.6 `fill_between`: 填充图形

```
python
复制编辑
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)

plt.fill_between(x, y1, y2, color="lightblue", alpha=0.5)
plt.plot(x, y1, label="sin(x)")
plt.plot(x, y2, label="cos(x)")
plt.legend()
plt.show()
```

2. 自定义功能函数

2.1 `title`: 设置标题

```
python
复制编辑
plt.title("My Plot Title", fontsize=14, color="blue")
```

2.2 `xlabel` 和 `ylabel`: 设置轴标签

```
python
复制编辑
plt.xlabel("X-axis Label", fontsize=12)
plt.ylabel("Y-axis Label", fontsize=12)
```

2.3 `legend`: 添加图例

```
python
复制编辑
plt.legend(loc="upper left", fontsize=10)
```

2.4 grid: 添加网格线

```
python
复制编辑
plt.grid(color="gray", linestyle="--", linewidth=0.5, alpha=0.7)
```

2.5 xlim 和 ylim: 设置坐标范围

```
python
复制编辑
plt.xlim(0, 10)
plt.ylim(-1, 1)
```

2.6 annotate: 添加注释

```
python
复制编辑
x, y = 5, 0.5
plt.scatter(x, y, color="red")
plt.annotate("This is a point", xy=(x, y), xytext=(x+1, y+1),
            arrowprops=dict(facecolor="black", arrowstyle="->"))
```

3. 图形控制

3.1 figure: 创建一个新图形窗口

```
python
复制编辑
plt.figure(figsize=(8, 6), dpi=100)
```

3.2 subplot: 绘制多个子图

```
python
复制编辑
plt.subplot(2, 1, 1) # 行数, 列数, 子图索引
plt.plot([1, 2, 3], [4, 5, 6])
plt.title("Subplot 1")

plt.subplot(2, 1, 2)
```

5. 保存图形

5.1 `savefig`: 将图保存为文件

python

复制编辑

```
plt.savefig("my_plot.png", dpi=300, bbox_inches="tight")
```