

人工智能程序设计

python



```
import turtle
turtle.setup(650,350,200,200)
turtle.penup()
turtle.fd(-250)
turtle.pendown()
turtle.pensize(25)
turtle.color("purple")
for i in range(4):
    turtle.circle(40, 80)
    turtle.circle(-40, 80)
    turtle.circle(40, 80/2)
    turtle.fd(40)
    turtle.circle(16, 180)
    turtle.fd(40 * 2/3)
```



人工智能程序设计

10.3 数据可视化: MATPLOTLIB

北京石油化工学院 人工智能研究院

刘 强

Matplotlib 简介

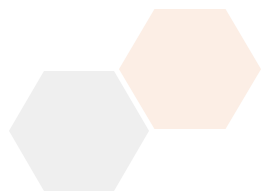
Matplotlib 是 Python 最主要的绘图库：

- 提供类似 **MATLAB** 的绘图接口
- 是 Python 数据可视化的基础
- 几乎所有其他可视化库都建立在 **Matplotlib** 之上
- 可以创建静态、动态和交互式图表



10.3.1 Matplotlib基础与环境配置

- **功能全面**：支持各种类型的图表，从简单的折线图到复杂的 3D 图形
- **高度可定制**：提供精细的样式控制，可创建出版级质量的图表
- **多平台支持**：支持多种输出格式（PNG、PDF、SVG 等）
- **与NumPy和Pandas集成**：与科学计算库无缝集成
- **面向对象接口**：提供灵活的编程接口



环境搭建

安装 **Matplotlib** 库:

```
## 使用pip安装
pip install matplotlib

## 验证安装
python -c "import matplotlib; print(matplotlib.__version__)"
```

导入 **Matplotlib** 的标准方式, 通常使用 **plt** 作为 **pyplot** 的别名:

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

## 设置中文字体支持 (可选)
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False
```

第一个Matplotlib图表

从一个简单的折线图开始，了解 **Matplotlib** 的基本绘图函数：

```
## 使用np.linspace创建等间距的数值序列
x = np.linspace(0, 10, 100)
y = np.sin(x)

## 创建图形，figsize参数设置图表大小（宽，高）
plt.figure(figsize=(8, 6))
plt.plot(x, y)
plt.title('正弦函数图像')
plt.xlabel('x轴')
plt.ylabel('y轴')
plt.grid(True)
plt.show()
```



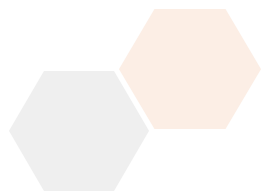
Matplotlib的两种接口

使用 **pyplot** 的函数式接口，直接调用 **plt** 模块的函数：

```
plt.figure()  
plt.plot([1, 2, 3, 4], [1, 4, 2, 3])  
plt.show()
```

使用 **subplots** 函数创建图形对象 (**fig**) 和坐标轴对象 (**ax**):

```
fig, ax = plt.subplots()  
ax.plot([1, 2, 3, 4], [1, 4, 2, 3])  
plt.show()
```



10.3.2 基础图表绘制

折线图适合展示数据的趋势变化，使用 **marker**、**label**、**linewidth** 等参数美化：

```
months = ['1月', '2月', '3月', '4月', '5月', '6月']
sales = [120, 135, 158, 142, 168, 195]
profit = [20, 25, 35, 28, 40, 55]

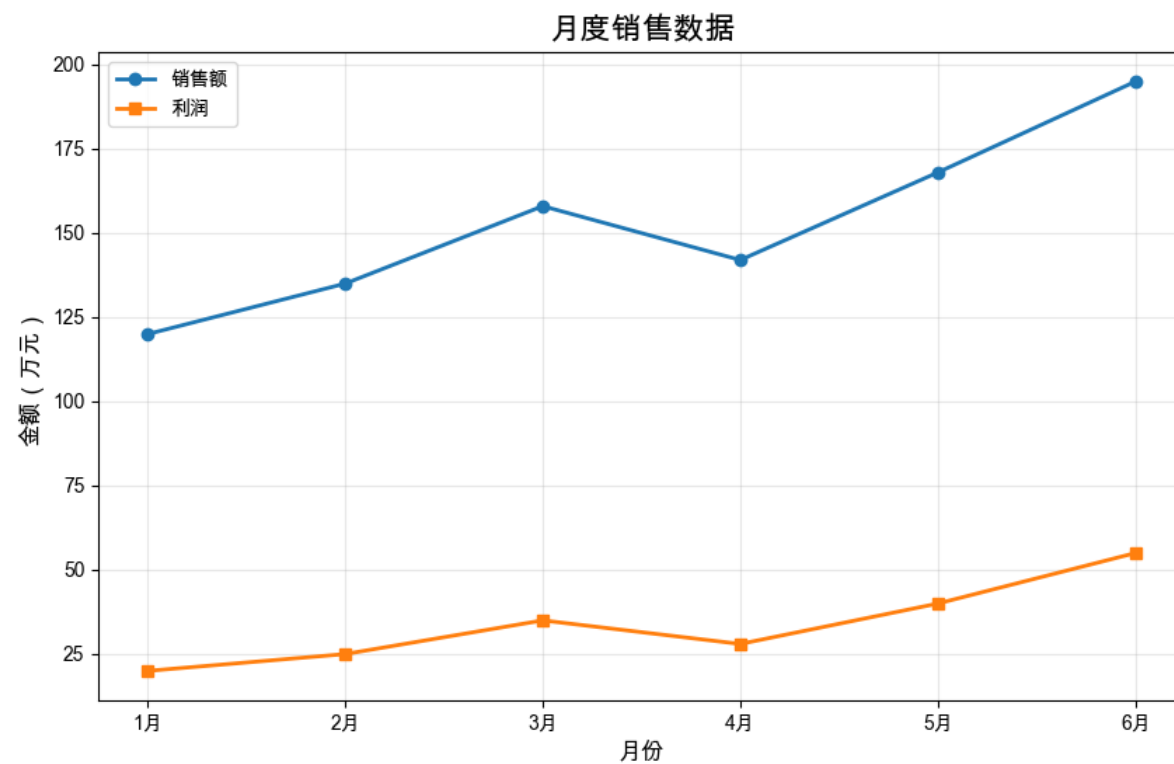
plt.figure(figsize=(10, 6))
plt.plot(months, sales, marker='o', label='销售额', linewidth=2)
plt.plot(months, profit, marker='s', label='利润', linewidth=2)

plt.title('月度销售数据', fontsize=16)
plt.xlabel('月份', fontsize=12)
plt.ylabel('金额（万元）', fontsize=12)
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



折线图效果

图 10.3.1 折线图



柱状图

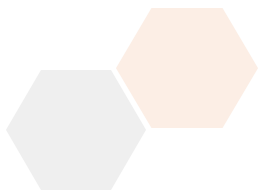
柱状图适合比较不同类别的数据，使用 `bar` 函数绘制：

```
categories = ['产品A', '产品B', '产品C', '产品D']
values = [23, 45, 56, 78]

plt.figure(figsize=(8, 6))
bars = plt.bar(categories, values, color=['red', 'blue', 'green', 'orange'])
plt.title('产品销量对比')
plt.ylabel('销量')

## 在柱子上显示数值
for bar, value in zip(bars, values):
    plt.text(bar.get_x() + bar.get_width()/2, bar.get_height() + 1,
             str(value), ha='center', va='bottom')

plt.show()
```



水平柱状图

使用 `barh` 函数绘制水平柱状图：

```
plt.figure(figsize=(8, 6))  
plt.barh(categories, values, color='skyblue')  
plt.title('产品销量对比（水平）')  
plt.xlabel('销量')  
plt.show()
```

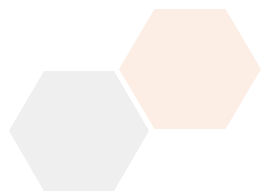


散点图

散点图用于显示两个变量之间的关系，使用 `scatter` 函数绘制：

```
## 设置随机种子确保结果可重现
np.random.seed(42)
x = np.random.randn(100)
y = 2 * x + np.random.randn(100)

plt.figure(figsize=(8, 6))
## alpha参数设置透明度，c参数设置颜色，s参数设置点的大小
plt.scatter(x, y, alpha=0.6, c='blue', s=50)
plt.title('散点图示例')
plt.xlabel('X变量')
plt.ylabel('Y变量')
plt.grid(True, alpha=0.3)
plt.show()
```

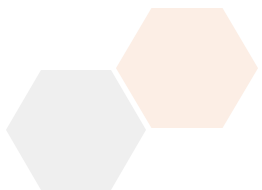


饼图

饼图用于显示各部分占总体的比例，使用 **pie** 函数绘制：

```
labels = ['苹果', '三星', '华为', '小米', '其他']
sizes = [25, 20, 18, 12, 25]
colors = ['gold', 'lightcoral', 'lightskyblue', 'lightgreen', 'plum']

plt.figure(figsize=(8, 8))
## autopct参数设置百分比显示格式, startangle设置起始角度
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%',
        startangle=90, shadow=True)
plt.title('手机市场份额')
plt.axis('equal') # 确保饼图显示为正圆形
plt.show()
```



直方图

直方图用于显示数据的分布，使用 `hist` 函数绘制：

```
## 生成正态分布数据：均值=100，标准差=15，数据量=1000
data = np.random.normal(100, 15, 1000)

plt.figure(figsize=(8, 6))
## bins参数设置柱子数量，edgecolor设置边框颜色
plt.hist(data, bins=30, color='skyblue', alpha=0.7, edgecolor='black')
plt.title('数据分布直方图')
plt.xlabel('数值')
plt.ylabel('频次')
plt.grid(True, alpha=0.3)
plt.show()
```



10.3.3 Ask AI: 深入学习数据可视化

掌握了 Matplotlib 基础后，可以向 AI 助手询问更多高级可视化技术：

- "如何使用 Seaborn 简化统计图表的绘制？"
- "如何使用 Plotly 创建交互式图表？"
- "如何选择合适的 Python 可视化库？"
- "如何为商业报告制作专业的图表？"
- "如何创建自定义的图表样式和主题？"



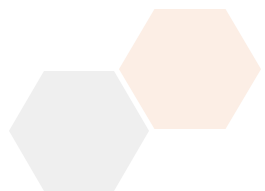
实践练习

练习 10.3.1：基础图表制作

使用以下销售数据创建多种图表：

1. 创建折线图比较线上线下销售趋势
2. 制作柱状图对比各月销售额
3. 绘制饼图显示总销售额的线上线下占比

```
data = {  
    '月份': ['1月', '2月', '3月', '4月', '5月', '6月'],  
    '线上销售': [120, 135, 158, 142, 168, 195],  
    '线下销售': [80, 95, 88, 106, 125, 140]  
}
```

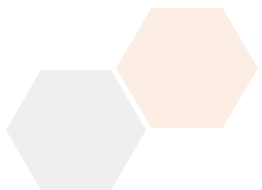


实践练习

练习 10.3.2：网站访问统计可视化

生成一组模拟的网站访问数据：

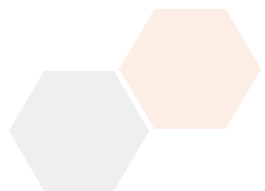
1. 创建直方图显示每日访问量分布
2. 制作散点图分析广告投入与访问量的关系
3. 分别创建两个独立的图表进行对比分析



实践练习

练习 10.3.3：图表美化与定制

1. 选择合适的颜色方案和样式
2. 添加详细的标题、标签和图例
3. 调整图表大小和布局，使其适合报告展示



本节小结

- **Matplotlib** 是 Python 数据可视化的基础库
- 基础图表：折线图、柱状图、散点图、饼图、直方图
- 图表美化：标题、标签、图例、颜色、样式
- 两种接口：pyplot 函数式接口和面向对象接口

