

实验 8 基于 Matplotlib 多类型图形绘制实践实习指导书

一、实验基本信息

项目	内容
实验名称	基于 Matplotlib 多类型图形绘制实践
授课课程	大数据分析
授课班级	工业工程 2241 班
实验学时	2 学时（90 分钟）
实验类型	综合技能型实验
实验目标	1. 掌握 Matplotlib 核心绘图流程，独立完成折线图、柱状图、散点图、饼图、雷达图的绘制；2. 熟练设置图形标题、轴标签、图例、颜色等样式元素，确保中文正常显示；3. 学会多图形子图组合布局，培养代码逐行校验、参数规范设置、结果可视化验证的科研品质。

二、实验原理

Matplotlib 是 Python 主流可视化库，核心通过 `pyplot` 模块实现“画布 - 图形 - 样式”三层绘图逻辑：

- 1. 画布 (Figure)：用 plt.figure() 创建，是所有图形的载体，可设置尺寸（如 figsize=(8,5)）和背景色；
- 2. 图形绘制：不同图形对应专用函数（如 plt.plot() 画折线图、plt.bar() 画柱状图、plt.polar() 画雷达图）；
- 3. 样式设置：通过 color（颜色）、label（图例标签）、linestyle（线型）等参数调整外观，用 plt.legend() 显示图例；
- 4. 结果输出：plt.savefig() 保存图片（需在 plt.show() 前执行），plt.show() 展示图形。

图形类型	核心函数 / 方法	适用场景	关键参数示例
折线图	plt.plot()	趋势变化（如销量走势）	color='blue'、marker='o'
柱状图	plt.bar()	类别对比（如产品销量）	width=0.6、edgecolor='black'
散点图	plt.scatter()	变量相关性（如温度与销量）	s=100、alpha=0.7
饼图	plt.pie()	占比展示（如成绩分布）	autopct='%1.1f%%'、explode=[0.1,0,0,0]
雷达图	plt.polar()	多维度对比（如多科成绩）	angles（极角）、fill=True

三、实验环境准备

1. 软件要求

- 5. 操作系统：Windows 10/11（64 位）；
- 6. 开发工具：Spyder（Anaconda 自带，确保 IPython 控制台可正常运行代码）；
- 7. 依赖库：
 - 1. matplotlib（≥3.5.0，绘图核心）；
 - 2. numpy（≥1.20.0，生成模拟数据）。

2. 环境检查与安装

打开 Spyder，在 IPython 控制台执行以下代码，验证环境：

```
import matplotlib.pyplot as plt
import numpy as np
print("Matplotlib 版本：", plt.matplotlib.__version__)
print("NumPy 版本：", np.__version__)
print("环境正常！")
```

若提示“模块不存在”，打开 Anaconda Prompt 执行：

```
conda install matplotlib numpy
```

四、实验内容与操作步骤

模块 1：中文显示配置与基础折线图（20 分钟）

目标

解决中文乱码问题，掌握折线图绘制与基础标签设置。

操作步骤

1. 新建代码文件 `Matplotlib_Exp8.py`，输入以下代码：

```
# 模块 1：中文配置与折线图绘制
import matplotlib.pyplot as plt
import numpy as np

# 关键：配置中文显示（必须放在绘图前）
plt.rcParams["font.family"] = ["SimHei", "WenQuanYi Micro Hei", "Heiti TC"]
plt.rcParams["axes.unicode_minus"] = False # 确保负号正常显示

# 生成模拟数据
```

```

x = np.array([1, 2, 3, 4, 5]) # 天数
y = np.array([20, 35, 30, 45, 50]) # 销量

# 1. 创建画布
plt.figure(figsize=(8, 5), facecolor='white')

# 2. 绘制折线图
plt.plot(x, y, color='blue', marker='o', label='每日销量')

# 3. 添加标签与图例
plt.title('5 天销量变化折线图', fontsize=12)
plt.xlabel('天数', fontsize=10)
plt.ylabel('销量（件）', fontsize=10)
plt.legend()

# 4. 保存并展示图形
plt.savefig('折线图.png', dpi=300) # 先保存再展示
plt.show()

# 细节检查
print("折线图数据检查： x={}, y={}".format(x, y))
print("若中文乱码，检查 plt.rcParams 配置是否正确！")

```

1. 运行后检查：

1. 工作目录生成折线图.png，中文标题、标签无乱码；
2. 图形含蓝色折线、圆形标记，图例显示“每日销量”。

模块 2：柱状图与散点图绘制（25 分钟）

目标

掌握柱状图数值标签添加、散点图大小与透明度设置。

操作步骤

继续添加代码：

```
# 模块 2：柱状图与散点图绘制
```

```

# ----- 柱状图 -----
products = ['产品 A', '产品 B', '产品 C']
sales = [120, 85, 150]
plt.figure(figsize=(8, 5))
bars = plt.bar(products, sales, color=['green', 'orange', 'purple'], width=0.6)
# 给柱子添加数值标签
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 5,
             str(height), ha='center')
plt.title('3 种产品季度销量柱状图', fontsize=12)
plt.xlabel('产品类型', fontsize=10)
plt.ylabel('销量（件）', fontsize=10)
plt.savefig('柱状图.png', dpi=300)
plt.show()

# ----- 散点图 -----
temp = np.array([18, 20, 22, 25, 28, 30]) # 温度
ice_sales = np.array([50, 65, 80, 120, 150, 180]) # 冰淇淋销量
plt.figure(figsize=(8, 5))
plt.scatter(temp, ice_sales, s=100, c='red', alpha=0.7)
plt.title('温度与冰淇淋销量散点图', fontsize=12)
plt.xlabel('温度（℃）', fontsize=10)
plt.ylabel('销量（个）', fontsize=10)
plt.savefig('散点图.png', dpi=300)
plt.show()

# 细节检查
print("柱状图产品数量： {}（预期 3 个）".format(len(products)))
print("散点图数据点数量： {}（预期 6 个）".format(len(temp)))

```

1. 运行后检查：

1. 柱状图每个柱子顶部有销量数值，颜色与代码设置一致；

2. 散点图有 6 个红色点，点大小适中、无重叠。

模块 3：饼图与雷达图绘制（30 分钟）

目标

掌握饼图百分比显示与雷达图多维度对比逻辑。

操作步骤

继续添加代码：

```
# 模块 3：饼图与雷达图绘制

# ----- 饼图 -----

grades = ['优秀', '良好', '及格', '不及格']
ratios = [15, 30, 40, 15]
colors = ['#FFA07A', '#98FB98', '#87CEFA', '#DDA0DD']
plt.figure(figsize=(8, 5))
plt.pie(ratios, labels=grades, colors=colors,
        autopct='%1.1f%%', explode=[0.1, 0, 0, 0])
plt.title('班级成绩等级占比饼图', fontsize=12)
plt.savefig('饼图.png', dpi=300)
plt.show()

# ----- 雷达图 -----

subjects = ['数学', '语文', '英语', '物理', '化学']
student1 = [90, 85, 88, 92, 78]
student2 = [82, 95, 80, 88, 90]

# 计算极角并闭合数据
angles = np.linspace(0, 2 * np.pi, len(subjects), endpoint=False)
student1 = np.concatenate((student1, [student1[0]]))
student2 = np.concatenate((student2, [student2[0]]))
subjects = np.concatenate((subjects, [subjects[0]]))
angles = np.concatenate((angles, [angles[0]]))
```

```
plt.figure(figsize=(8, 8))
ax = plt.subplot(111, polar=True) # 极坐标投影
ax.plot(angles, student1, 'o-', linewidth=2, label='学生 1')
ax.fill(angles, student1, alpha=0.25)
ax.plot(angles, student2, 's-', linewidth=2, label='学生 2')
ax.fill(angles, student2, alpha=0.25)
ax.set_thetagrids(angles * 180 / np.pi, subjects)
ax.set_ylim(0, 100)
plt.title('学生成绩雷达图（5 科对比）', fontsize=12)
plt.legend(loc='upper right')
plt.savefig('雷达图.png', dpi=300)
plt.show()
# 细节检查
print("饼图占比总和: {}% (预期 100%) ".format(sum(ratios)))
print("雷达图是否闭合: {}".format(student1[0] == student1[-1]))
```

1. 运行后检查：

1. 饼图显示百分比，“优秀”扇形轻微突出；
2. 雷达图为闭合多边形，2 条曲线分别对应 2 名学生，科目标签无重叠。

模块 4：多图形子图组合（10 分钟）

目标

掌握 2×2 子图布局，实现多类型图形共存。

操作步骤

继续添加代码：

```
# 模块 4：多图形子图组合
plt.figure(figsize=(12, 10))
# 子图 1：折线图（第 1 行第 1 列）
```

```

plt.subplot(2, 2, 1)
plt.plot(x, y, color='blue', marker='o')
plt.title('折线图子图', fontsize=10)
# 子图 2: 柱状图 (第 1 行第 2 列)
plt.subplot(2, 2, 2)
plt.bar(products, sales, color=['green', 'orange', 'purple'])
plt.title('柱状图子图', fontsize=10)
# 子图 3: 散点图 (第 2 行第 1 列)
plt.subplot(2, 2, 3)
plt.scatter(temp, ice_sales, c='red')
plt.title('散点图子图', fontsize=10)
# 子图 4: 雷达图 (第 2 行第 2 列, 极坐标)
plt.subplot(2, 2, 4, polar=True)
plt.plot(angles[:-1], student1[:-1], 'o-', linewidth=2) # 去除闭合的最后一个点
plt.title('雷达图子图', fontsize=10)
plt.tight_layout() # 自动调整子图间距
plt.savefig('子图组合.png', dpi=300)
plt.show()
print("子图数量: 4 个 (2 行 2 列) ")

```

1. 运行后检查:

1. 4 个子图无标签重叠, 布局整齐;
2. 每个子图对应一种图形类型, 雷达图在极坐标下正常显示。

模块 5: 实验结果验证与整理 (5 分钟)

目标

确保所有图形生成, 培养结果自查习惯。

操作步骤

添加代码:

```
# 模块 5：结果验证与整理

import os

required_files = ['折线图.png', '柱状图.png', '散点图.png',
                  '饼图.png', '雷达图.png', '子图组合.png']

print("\n=== 实验结果检查 ===")

for file in required_files:
    if os.path.exists(file):
        print(f"✅ 成功生成：{file}")
    else:
        print(f"❌ 缺失文件：{file}（请重新运行对应模块）")

print("\n=== 操作规范自查 ===")

print("1. 中文是否全部正常显示？")
print("2. 图形是否都保存为.png 且清晰？")
print("3. 子图是否通过 plt.tight_layout()避免重叠？")
print("4. 代码是否添加了关键注释？")
```

五、实验注意事项

1. 中文显示必配：绘图前必须添加 `plt.rcParams` 配置，否则中文标题 / 标签会显示为乱码。
2. 保存与展示顺序：`plt.savefig()` 必须在 `plt.show()` 前执行，否则保存空白图片。
3. 数据一致性：折线图、散点图的 x 和 y 数据长度必须一致，否则报“维度不匹配”错误。
4. 雷达图闭合关键：通过 `np.concatenate()` 在数据、角度、标签末尾添加首个元素，确保图形闭合。
5. 子图布局优化：多子图组合时用 `plt.tight_layout()` 自动调整间距，避免标签重叠。

六、实验报告要求

1. 结构要求：包含“实验目的”“实验原理”“实验步骤（附各模块代码截图）”“实验结果（附 6 张图形截图，标注图名）”“问题与解决方法”“总结”。
2. 关键记录：

1. 至少记录 2 个错误及解决过程（如“雷达图不闭合：添加 `np.concatenate()` 后正常”）；
 2. 对比各图形适用场景（如“雷达图适合多维度对比，饼图适合占比展示”）。
1. **截图规范：**图形截图需完整显示标题、标签、图例，代码截图需包含注释，证明为本人操作。

七、实验考核标准

考核维度	考核要点（总分 100 分）	分值
代码正确性	6 个图形全部生成，无语法错误，数据匹配正确（如折线图 x 与 y 长度一致）	40
细节规范性	中文显示正常，标签完整，参数设置合理（如饼图有百分比、雷达图闭合）	30
报告完整性	结构完整，截图清晰且标注规范，“问题与解决”真实可信	20
科研品质	代码有注释，主动检查数据一致性，按提示修正错误	10

本指导书严格遵循“操作 - 校验 - 总结”的科研训练逻辑，通过逐模块细节要求，培养学生从代码编写到结果验证的全流程严谨性。