

实验三 Matplotlib 数据可视化

1.实验类型

验证型实验

2.实验目的和要求

- (1) 掌握 pyplot 基础语法;
- (2) 掌握散点图、折线图的绘制方法;
- (3) 掌握直方图的绘制方法;
- (4) 掌握饼图的绘制方法;
- (5) 掌握箱线图的绘制方法;

3.实验内容

题目一：分析 1996-2015 年人口数据特征间的关系

(1)训练要点

- 掌握 pyplot 基础语法
- 掌握子图的绘制方法
- 掌握散点图、折线图的绘制方法

(2)需求说明

人口数据总共拥有 6 个特征，分别为年末总人口、男性人口、女性人口、城镇人口、乡村人口和年份。查看各个特征随着时间推移发生的变化情况可以分析出未来男女人口比例、城乡人口变化的方向。

(3)实现步骤

- 使用 numpy 库读取人口数据。
- 创建画布，并添加子图。
- 在两个子图上分别绘制散点图和折线图。
- 使用不同的颜色、不同形状的点，绘制散点图和折线图。
- 保存，显示图片。
- 分析未来人口变化趋势。

题目二：分析 1996-2015 年人口数据各个特征的分布与分散状况

(1)训练要点

- 掌握直方图的绘制
- 掌握饼图的绘制方法

- 掌握箱线图的绘制方法

(2)需求说明

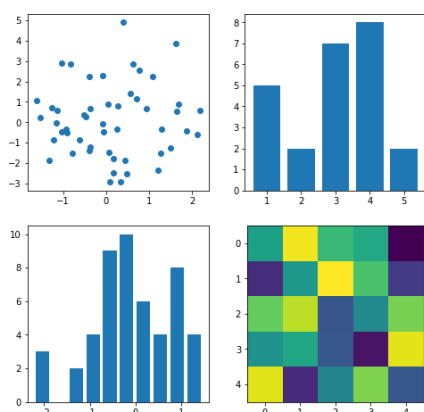
通过绘制各年份男女人口数目及城乡人口数目的直方图，男女人口比例及城乡人口比例的饼图可以发现人口结构的变化。而绘制每个特征的箱线图则可以发现不同特征增长或者减少的速率是否变得缓慢。

(3)实现步骤

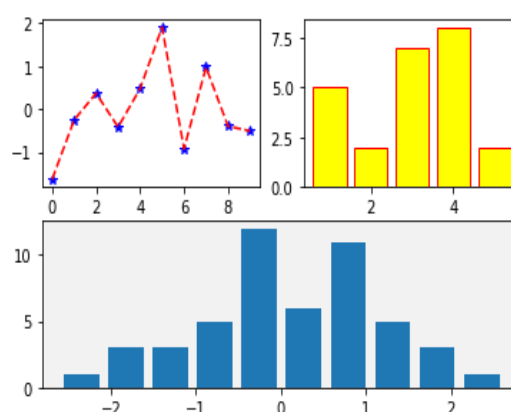
- 创建 3 幅画布并添加对应数目的子图。
- 在每一幅子图上绘制对应的图形。
- 保存和显示图形。
- 根据图形，分析我国人口结构变化情况以及变化速率的增减状况。

题目三：DIY 可视化

绘制如下的图形之一，只要求布局一样，显示的数据内容和可视化方式自己决定，可以是学习中、娱乐中、生活中的任意数据。图形二选一，选择的结果见附件。



图一

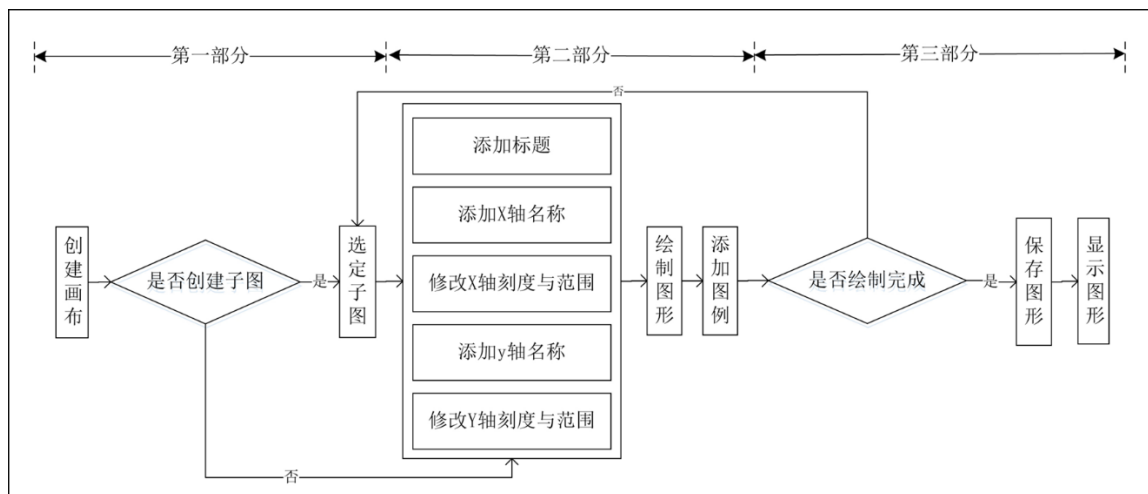


图二

4.实验背景知识

Matplotlib 是一个 Python 2D 绘图库，可以生成各种硬拷贝格式和跨平台交互式环境的出版物质量数据。Matplotlib 可用于 Python 脚本，Python 和 IPython shell，Jupyter 笔记本，Web 应用程序服务器和四个图形用户界面工具包。对于简单的绘图，pyplot 模块提供类似 MATLAB 的接口，它可与 NumPy 一起使用，提供了一种有效的 MatLab 开源替代方案。

(1) 基本绘图流程



（2）创建画布与创建子图

函数名称	函数作用
<code>plt.figure</code>	创建一个空白画布，可以指定画布大小，像素。
<code>figure.add_subplot</code>	创建并选中子图，可以指定子图的行数，列数，与选中图片编号。

（3）画布内容

函数名称	函数作用
<code>plt.title</code>	在当前图形中添加标题，可以指定标题的名称、位置、颜色、字体大小等参数。
<code>plt.xlabel</code>	在当前图形中添加x轴名称，可以指定位置、颜色、字体大小等参数。
<code>plt.ylabel</code>	在当前图形中添加y轴名称，可以指定位置、颜色、字体大小等参数。
<code>plt.xlim</code>	指定当前图形x轴的范围，只能确定一个数值区间，而无法使用字符串标识。
<code>plt.ylim</code>	指定当前图形y轴的范围，只能确定一个数值区间，而无法使用字符串标识。
<code>plt.xticks</code>	指定x轴刻度的数目与取值。
<code>plt.yticks</code>	指定y轴刻度的数目与取值。
<code>plt.legend</code>	指定当前图形的图例，可以指定图例的大小、位置、标签。

（4）保存与展示图形

函数名称	函数作用
<code>plt.savefig</code>	保存绘制的图片，可以指定图片的分辨率、边缘的颜色等参数。
<code>plt.show</code>	在本机显示图形。

（5）绘制散点图

`matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None, alpha=None, **kwargs)`

主要参数说明：

参数名称	说明
x , y	接收array。表示x轴和y轴对应的数据。无默认。
s	接收数值或者一维的array。指定点的大小，若传入一维array则表示每个点的大小。默认为None。
c	接收颜色或者一维的array。指定点的颜色，若传入一维array则表示每个点的颜色。默认为None
marker	接收特定string。表示绘制的点的类型。默认为None。
alpha	接收0-1的小数。表示点的透明度。默认为None。

（6）绘制折线图

*matplotlib.pyplot.plot(*args, **kwargs)*

plot 函数在官方文档的语法中只要求填入不定长参数，实际可以填入的主要参数主要如下：

参数名称	说明
x , y	接收array。表示x轴和y轴对应的数据。无默认。
color	接收特定string。指定线条的颜色。默认为None。
linestyle	接收特定string。指定线条类型。默认为 “-” 。
marker	接收特定string。表示绘制的点的类型。默认为None。
alpha	接收0-1的小数。表示点的透明度。默认为None。

（7）绘制条形图、直方图

条形图：

*matplotlib.pyplot.bar(left, height, width = 0.8, bottom = None, hold = None, data = None, **kwargs)*

常用参数如下：

参数名称	说明
left	接收array。表示x轴数据。无默认。
height	接收array。表示x轴所代表数据的数量。无默认。
width	接收0-1之间的float。指定直方图宽度。默认为0.8。
color	接收特定string或者包含颜色字符串的array。表示直方图颜色。默认为None。

直方图：

matplotlib.pyplot.hist(x, bins, rwidth ,color)

常用参数如下：

参数名称	说明
x	接收array。指定要绘制的数据。
bins	接收array。直方图条形的个数。
rwidth	设置直方图条形的宽度。
color	接收特定string或者包含颜色字符串的array。表示直方图颜色。默认为None。

(8) 绘制饼图

`matplotlib.pyplot.pie(x, explode=None, labels=None, colors=None, autopct=None, pctdistance=0.6, shadow=False, labeldistance=1.1, startangle=None, radius=None, ...)`

常用参数如下：

参数名称	说明	参数名称	说明
x	接收array。表示用于绘制撇的数据。无默认。	autopct	接收特定string。指定数值的显示方式。默认为None。
explode	接收array。表示指定项离饼图圆心为n个半径。默认为None。	pctdistance	接收float。指定每一项的比例和距离饼图圆心n个半径。默认为0.6。
labels	接收array。指定每一项的名称。默认为None。	labeldistance	接收float。指定每一项的名称和距离饼图圆心多少个半径。默认为1.1。
color	接收特定string或者包含颜色字符串的array。表示饼图颜色。默认为None。	radius	接收float。表示饼图的半径。默认为1。

(9) 绘制箱线图

`matplotlib.pyplot.boxplot(x, notch=None, sym=None, vert=None, whis=None, positions=None, widths=None, patch_artist=None, meanline=None, labels=None, ...)`

常用参数如下：

参数名称	说明	参数名称	说明
x	接收array。表示用于绘制箱线图的数据。无默认。	positions	接收array。表示图形位置。默认为None。
notch	接收boolean。表示中间箱体是否有缺口。默认为None。	widths	接收scalar或者array。表示每个箱体的宽度。默认为None。
sym	接收特定sting。指定异常点形状。默认为None。	labels	接收array。指定每一个箱线图的标签。默认为None。
vert	接收boolean。表示图形是横向纵向或者横向。默认为None。	meanline	接收boolean。表示是否显示均值线。默认为False。

(10) 部分代码参考：

代码 1 1996-2015 年人口数据散点图与折线图

```
In[1]: import os
import numpy as np
import matplotlib.pyplot as plt
os.chdir('C:/Training/数据')
plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文显示
feature = np.load('./第 3 章/data/populations.npz')['feature_names'] # 加载列名
amount = np.load('./第 3 章/data/populations.npz')['data'] # 加载数据

p1 = plt.figure(figsize=(9,10)) #设置画布
ax1 = p1.add_subplot(2,1,1) #创建第一幅子图
plt.title('1996-2015 年人口数据散点图与折线图')
plt.ylabel('人口数量')
plt.xticks(rotation = 45)
for i in range(1,6): #利用 for 循环画图
    plt.scatter(amount[:,0][0:20],amount[:,i][0:20])
```

```

ax1.legend(feature[1:],bbox_to_anchor=(1, 1),ncol=1) #图例放图外，以免遮挡信息

ax2 = p1.add_subplot(2,1,2)      #创建第二幅子图
plt.ylabel('人口数量')
plt.xticks(rotation = 45)
for i in range(1,6):
    plt.plot(amount[:,0][0:20],amount[:,i][0:20])
ax2.legend(feature[1:],bbox_to_anchor=(1, 1),ncol=1)

```

代码 2 1996-2015 年男女人口及乡镇人口数目直方图

```

In[1]: import os
import numpy as np
import matplotlib.pyplot as plt
os.chdir('C:/Training/数据')
plt.rcParams['font.sans-serif'] = ['SimHei'] # 设置中文显示
feature = np.load('./第 3 章/data/populations.npz')['feature_names'] # 加载列名
amount = np.load('./第 3 章/data/populations.npz')['data'] # 加载数据
#第一幅画布
p1 = plt.figure(figsize=(9,9)) #设置画布
ax1 = p1.add_subplot(2,1,1)      #创建第一幅子图
plt.title('1996-2015 年男女人口及乡镇人口数目直方图')
plt.ylabel('人口数量')
#plt.bar(amount[:,0],amount[:,2],width=0.5)
#plt.bar(amount[:,0],amount[:,3],width=0.5)
plt.bar(np.arange(22),amount[:,2],width=0.3)
plt.bar(np.arange(22)+0.3,amount[:,3],width=0.3)
plt.xticks(np.arange(22),amount[:,0],rotation = 45)
ax1.legend(feature[2:4],bbox_to_anchor=(1, 1),ncol=1) #图例放图外，以免遮挡信息

ax2 = p1.add_subplot(2,1,2)      #创建第二幅子图
plt.ylabel('人口数量')
plt.bar(np.arange(22),amount[:,4],width=0.3)
plt.bar(np.arange(22)+0.3,amount[:,5],width=0.3)
plt.xticks(range(22),amount[:,0],rotation = 45)
ax2.legend(feature[4:6],bbox_to_anchor=(1, 1),ncol=1)

plt.savefig('./第 3 章/tmp/1996-2015 年男女人口及乡镇人口数目直方图.png')
plt.show

```

代码 3 1996-2015 年男女人口及乡镇人口比例饼图

```

In[1]: i=0
l=1
p2 = plt.figure(figsize=(4,44))
while i <20:
    pie1 = p2.add_subplot(20,2,l)
    label = ['男','女']
    explode = [0.01,0.01]
    plt.pie(amount[19-i,2:4],explode=explode,labels=label,autopct='%1.1f%%')
    plt.title('%d 年男女与城乡人口比例' %(1996+i))
    i +=1
    l +=2
j=0
k=2
while j <20:
    pie1 = p2.add_subplot(20,2,k)
    label = ['城镇人口','乡村人口']
    explode = [0.01,0.01]
    plt.pie(amount[19-j,4:6],explode=explode,labels=label,autopct='%1.1f%%')
    j +=1
    k +=2

plt.savefig('./第 3 章/tmp/1996-2015 年男女人口及乡镇人口比例饼图.png')
plt.show

```

代码 4 1996-2015 年人口变化箱线图

```

In[1]: #第三幅画布
plt.figure(figsize=(9,9))
label=['年末总人口','男性人口','女性人口','城镇人口','乡村人口']
population=(amount[:,1],amount[:,2],amount[:,3],amount[:,4],amount[:,5])
plt.boxplot(population,notch=True,labels=label,meanline=True)
plt.title('1996~2015 年人口变化箱线图')
plt.ylabel('人口数量(万人)')

plt.savefig('./第 3 章/tmp/1996~2015 年人口变化箱线图.png')
plt.show

```

5.实验思考

- (1) 散点图，折线图的主要功能是什么，有什么异同点？
- (2) 饼图，直方图的主要功能是什么，有什么异同点？
- (3) 箱型图的主要功能是什么？