西安交通大学实验报告

课程 **Python数据处理** 实验名称 **python函数结构控制**  共 **17** 页

系 别 **电信学部** 实 验 日 期 **2023**年**3**月**29**日

专业班级 实 验 报 告 日 期 **2023**年**3**月**29**日

姓 名 学号 报 告 退 发( 订正 、 重做 )

同 组 人\_\_\_\_\_\_\_\_\_\_**无**\_\_\_\_\_\_\_\_\_ 教 师 审 批 签 字

*本次实验合计约使用****2****个小时*

# **实验目的**

1. **掌握pyplot常用的绘图参数的调节方式**
2. **掌握绘制图形的保存与展示方法**
3. **掌握散点图和折线图的作用与绘制方法**
4. **掌握直方图、饼图和箱线图的作用与绘制方法**

## 实验题目一：

**使用Numpy和Matploitlib完成以下实验题目，从附件中下载数据集为populations.npz。**

**1）使用Numpy取出人口数据（包括6个特征，即年末总人口、男性人口、女性人口、城镇人口、乡村人口和年份）**

**2）创建画布，添加4个子图**

**3）在编号为1的子图上绘制散点图，通过该子图可以了解男女人口随时间发生的变化**

**4）在编号为2的子图上绘制散点图，通过该子图可以了解城乡人口随时间发生的变化**

**5）在编号为3的子图上绘制折线图，通过该子图可以了解男女人口随时间发生的变化**

**6）在编号为4的子图上绘制折线图，通过该子图可以了解城乡人口随时间发生的变化**

**7）保存并显示图形**

**提示：读取数据集程序段如下：**

**import numpy as np**

**import matplotlib.pyplot as plt**

**plt.rcParams['font.sans-serif'] = 'SimHei' ## 设置中文显示**

**plt.rcParams['axes.unicode\_minus'] = False**

**ldata = np.load('d:/2021数据/python程序设计\_智慧金融001-005/正式教学PPT顾/populations\_g.npz',allow\_pickle=True)**

**fl=ldata.files #查看导入进来的数组名称有哪些**

**print(fl)**

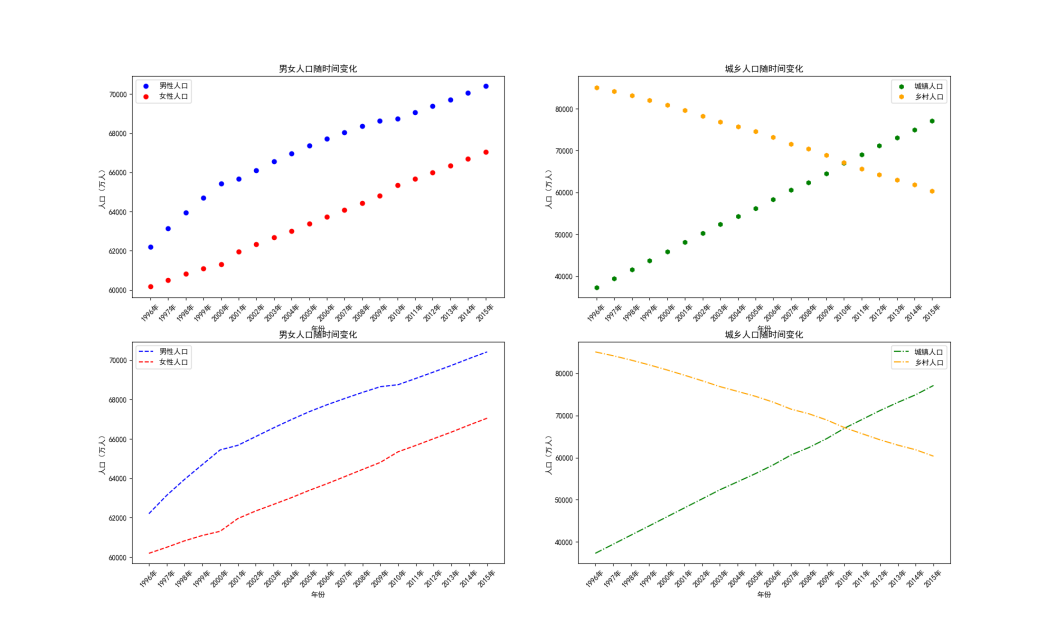
**假设数组名为“gu”，导入数组语句如下：**

**Name=ldata[‘gu’]**

#### 程序源代码：

**import numpy as np  
import matplotlib.pyplot as plt  
# 设置中文显示  
  
plt.rcParams['font.sans-serif'] = 'SimHei'  
plt.rcParams['axes.unicode\_minus'] = False  
# 导入数据  
  
data = np.load('populations.npz', allow\_pickle=True)  
feature\_names = data['feature\_names']  
populations = data['data'][19::-1]  
# 创建画布和子图  
  
fig = plt.figure(figsize=(20,12))  
ax1 = fig.add\_subplot(2, 2, 1)  
ax2 = fig.add\_subplot(2, 2, 2)  
ax3 = fig.add\_subplot(2, 2, 3)  
ax4 = fig.add\_subplot(2, 2, 4)  
# 绘制散点图：男女人口随时间变化  
  
ax1.scatter(populations[:, 0], populations[:, 2], marker='o', c='b')  
ax1.scatter(populations[:, 0], populations[:, 3], marker='o', c='r')  
ax1.set\_xlabel('年份')  
ax1.set\_ylabel('人口（万人）')  
ax1.set\_title('男女人口随时间变化')  
ax1.legend(['男性人口', '女性人口'])  
# 绘制散点图：城乡人口随时间变化  
  
ax2.scatter(populations[:, 0], populations[:, 4], marker='h', c='green')  
ax2.scatter(populations[:, 0], populations[:, 5], marker='h', c='orange')  
ax2.set\_xlabel('年份')  
ax2.set\_ylabel('人口（万人）')  
ax2.set\_title('城乡人口随时间变化')  
ax2.legend(['城镇人口', '乡村人口'])  
# 绘制折线图：男女人口随时间变化  
  
ax3.plot(populations[:, 0], populations[:, 2], color='b', linestyle='--')  
ax3.plot(populations[:, 0], populations[:, 3], color='r', linestyle='--')  
ax3.set\_xlabel('年份')  
ax3.set\_ylabel('人口（万人）')  
ax3.set\_title('男女人口随时间变化')  
ax3.legend(['男性人口', '女性人口'])  
# 绘制折线图：城乡人口随时间变化  
  
ax4.plot(populations[:, 0], populations[:, 4], color='green', linestyle='-.')  
ax4.plot(populations[:, 0], populations[:, 5], color='orange', linestyle='-.')  
ax4.set\_xlabel('年份')  
ax4.set\_ylabel('人口（万人）')  
ax4.set\_title('城乡人口随时间变化')  
ax4.legend(['城镇人口', '乡村人口'])  
  
for ax in [ax1, ax2, ax3, ax4]:  
 ax.tick\_params(axis='x', rotation=45)  
  
# 保存并显示图形  
  
plt.savefig('population.png')  
plt.show()**

#### 运行结果：

****

## 实验题目二：

**使用Numpy和Matploitlib完成以下实验题目，从附件中下载数据集为populations.npz。**

**1）使用Numpy取出人口数据（包括6个t特征，即年末总人口、男性人口、女性人口、城镇人口、乡村人口和年份）**

**2）创建画布，添加6个子图**

**3）在编号为1的子图上绘制直方图， 通过该子图可以了解男女人口随时间发生的变化**

**4）在编号为2的子图上绘制直方图， 通过该子图可以了解城乡人口随时间发生的变化**

**5）在编号为3的子图上绘制饼图， 通过该子图可以了解2015年男女人口比例**

**6）在编号为4的子图上绘制饼图， 通过该子图可以了解2015年城乡人口比例**

**7）在编号为5的子图上绘制箱线图， 通过该子图可以了解男女人口随时间发生的变化**

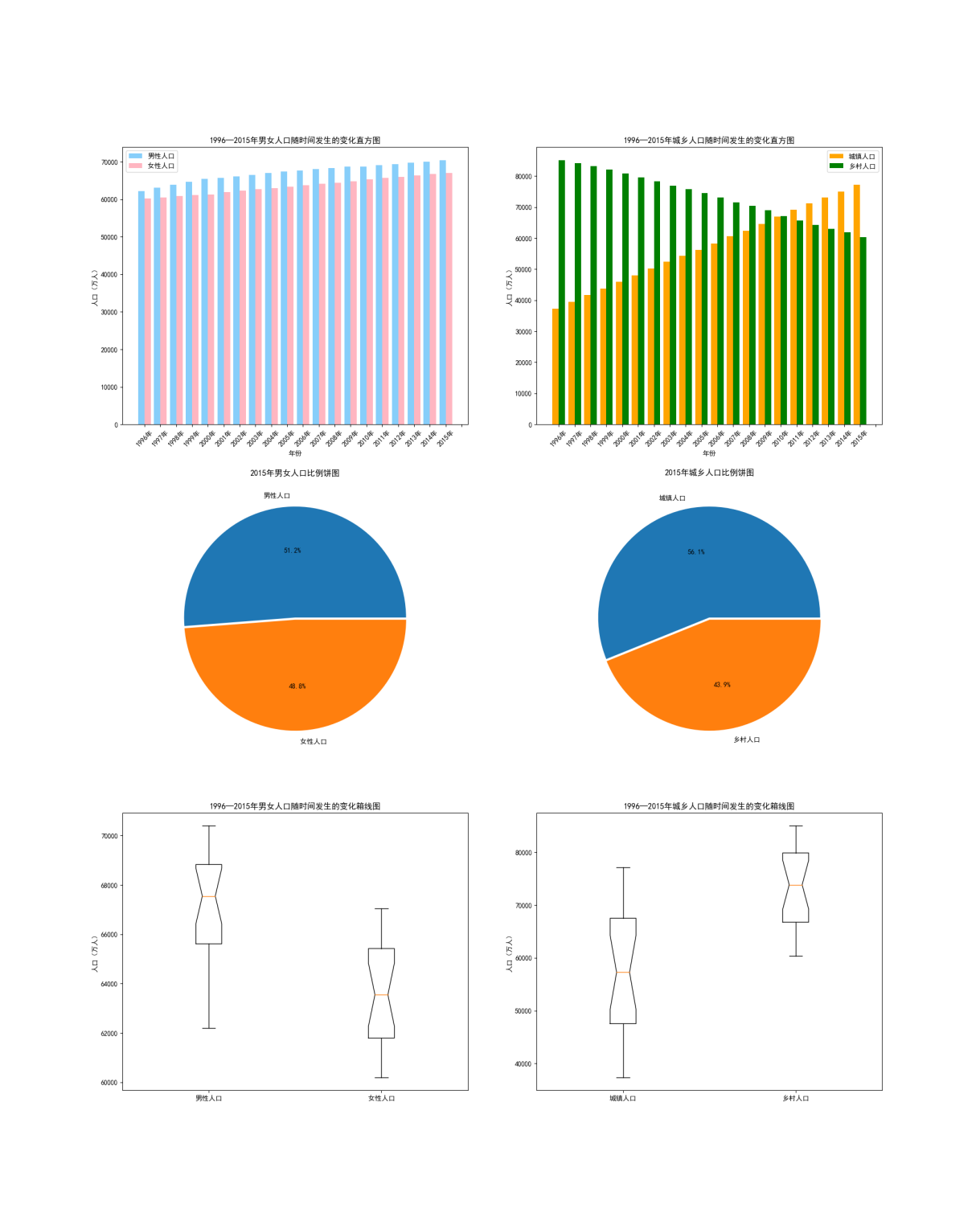
**8）在编号为6的子图上绘制箱线图， 通过该子图可以了解城乡人口随时间发生的变化**

**9）保存并显示图形**

#### 程序源代码

**import numpy as np  
import matplotlib.pyplot as plt  
  
# 设置中文显示  
plt.rcParams['font.sans-serif'] = 'SimHei'  
# 解决负号显示问题  
plt.rcParams['axes.unicode\_minus'] = False  
  
# 导入数据  
data = np.load('populations.npz', allow\_pickle=True)  
fl = data.files  
print(fl) # 查看导入进来的数组名称有哪些  
  
name = data['feature\_names']  
values = data['data']  
print(name)  
print(values)  
  
# 将数据矩阵倒置  
values = np.flip(values, axis=0)  
print(values) # 输出倒置后的矩阵  
  
# 设置画布  
p1 = plt.figure(figsize=(20, 25))  
  
# 绘制第一幅  
ax1 = p1.add\_subplot(3, 2, 1)  
Y1 = values[2:, 2]  
Y2 = values[2:, 3]  
X = np.arange(len(values[2:, 0]))  
ax1.bar(X-0.2, Y1, width=0.4, facecolor='lightskyblue')  
ax1.bar(X+0.2, Y2, width=0.4, facecolor='lightpink')  
plt.xlabel('年份')  
plt.ylabel('人口（万人）')  
label = values[2:, 0]  
plt.xticks(range(21), label, rotation=45)  
plt.title('1996—2015年男女人口随时间发生的变化直方图')  
plt.legend(['男性人口', '女性人口'])  
  
ax2 = p1.add\_subplot(3, 2, 2)  
Y1 = values[2:, 4]  
Y2 = values[2:, 5]  
X = np.arange(len(values[2:, 0]))  
ax2.bar(X-0.2, Y1, width=0.4, facecolor='orange')  
ax2.bar(X+0.2, Y2, width=0.4, facecolor='green')  
plt.xlabel('年份')  
plt.ylabel('人口（万人）')  
label = values[2:, 0]  
plt.xticks(range(21), label, rotation=45)  
plt.title('1996—2015年城乡人口随时间发生的变化直方图')  
plt.legend(['城镇人口', '乡村人口'])  
  
ax3 = p1.add\_subplot(3, 2, 3)  
plt.axis('equal')  
label = ['男性人口', '女性人口']  
explode = [0.01, 0.01]  
ax3.pie(values[-1, 2:4], explode=explode, labels=label, autopct='%1.1f%%')  
plt.title('2015年男女人口比例饼图')  
  
ax4 = p1.add\_subplot(3, 2, 4)  
plt.axis('equal')  
label2 = ['城镇人口', '乡村人口']  
explode = [0.01, 0.01]  
ax4.pie(values[-1, 4:6], explode=explode, labels=label2, autopct='%1.1f%%')  
plt.title('2015年城乡人口比例饼图')  
  
ax5 = p1.add\_subplot(3, 2, 5)  
label1 = ['男性人口', '女性人口']  
number1 = (list(values[2:, 2]), list(values[2:, 3]))  
plt.boxplot(number1, notch=True, labels=label1, meanline=True)  
plt.ylabel('人口（万人）')  
plt.title('1996—2015年男女人口随时间发生的变化箱线图')  
  
ax6 = p1.add\_subplot(3, 2, 6)  
label2 = ['城镇人口', '乡村人口']  
number2 = (list(values[2:, 4]), list(values[2:, 5]))  
plt.boxplot(number2, notch=True, labels=label2, meanline=True)  
plt.ylabel('人口（万人）')  
plt.title('1996—2015年城乡人口随时间发生的变化箱线图')  
  
plt.savefig('population2.png')  
  
plt.show()**

#### 运行结果：



## 实验题目三：

**使用Numpy和Matploitlib完成以下实验题目，从附件中下载数据集score.npz。**

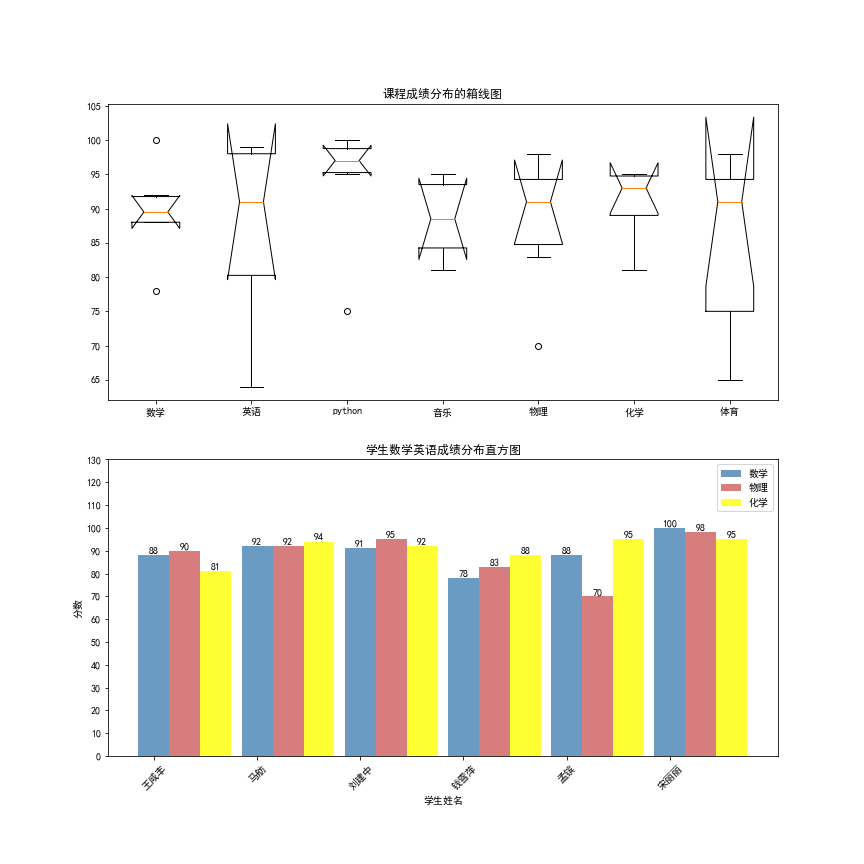
**1）使用Numpy取出score.npz数据集中数据，数据集共有三个数组，分别是name、class\_name、scores。Scores数据集是二维数组，6行7列分别表示6名学生、7门课程，即一行为某学生取得的分数，一列为某课程的分数。**

**2）创建画布，绘制两幅子图：箱线图和直方图**

**3）在编号为1的子图上绘制箱线图，根据每门课程成绩数据分布，一门课程一个箱线图，子图1中绘制7门课程的箱线图。**

**4）在编号为2的子图上绘制直方图，根据每位学生成绩数据分布，绘制数理化三门成绩的直方图，将6位学生姓名显示在X轴。**

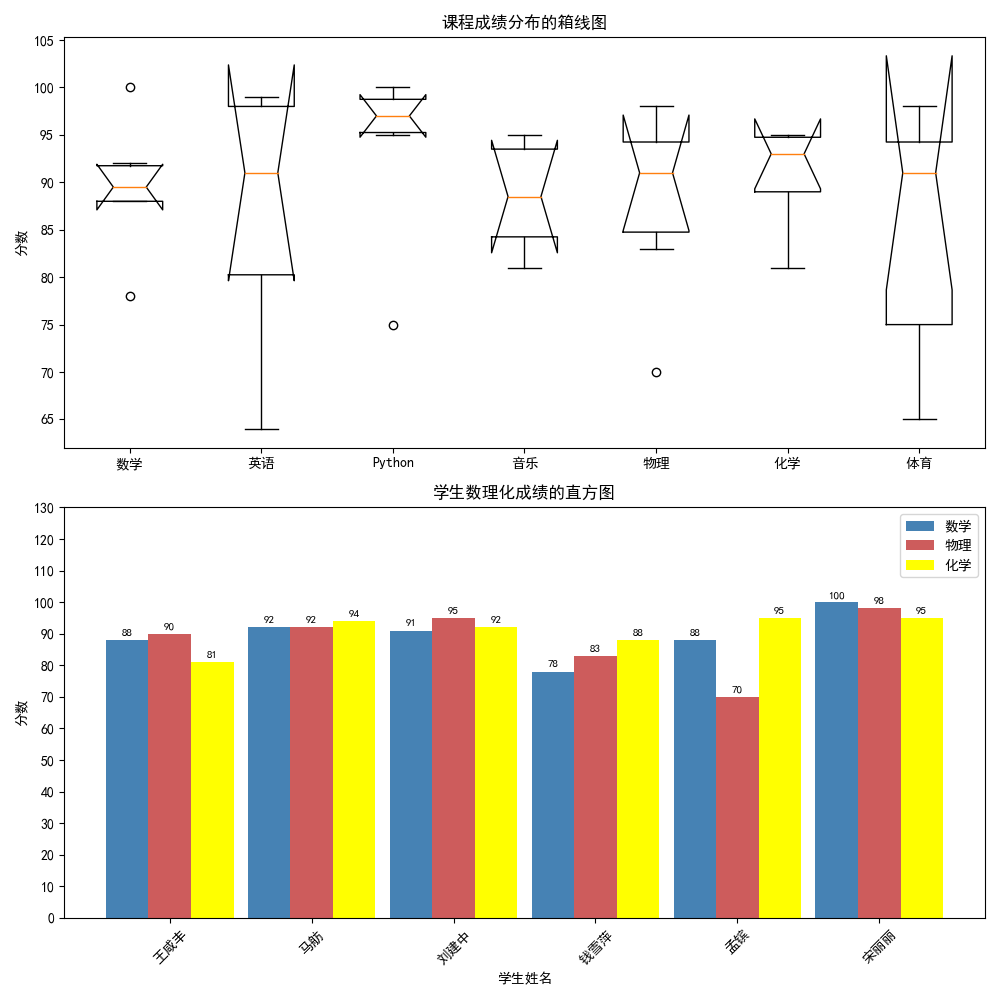
**5）保存并显示图形，参考图形如下：**



#### 程序源代码：

**import numpy as np  
import matplotlib.pyplot as plt  
  
# 设置中文显示和正常显示  
plt.rcParams['font.sans-serif'] = 'SimHei'  
plt.rcParams['axes.unicode\_minus'] = False  
  
# 导入数据  
data = np.load('score.npz', allow\_pickle=True)  
name = data['name']  
class\_name = data['class\_name']  
scores = data['scores']  
  
# 创建画布，绘制两幅子图  
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 10))  
  
# 第一幅子图：箱线图  
labels = ['数学', '英语', 'Python', '音乐', '物理', '化学', '体育']  
box\_data = ([list(scores[:, i]) for i in range(7)])  
ax1.boxplot(box\_data, notch=True, labels=labels, meanline=True)  
ax1.set\_ylabel('分数')  
ax1.set\_title('课程成绩分布的箱线图')  
  
# 第二幅子图：直方图  
math\_scores = scores[:, 0]  
physics\_scores = scores[:, -3]  
chemistry\_scores = scores[:, -2]  
positions = np.arange(len(scores))  
width = 0.3  
ax2.bar(positions, math\_scores, width=width, facecolor='steelblue')  
ax2.bar(positions + width, physics\_scores, width=width, facecolor='indianred')  
ax2.bar(positions + 2 \* width, chemistry\_scores, width=width, facecolor='yellow')  
ax2.set\_xticks(positions + width)  
ax2.set\_xticklabels(name, rotation=45)  
ax2.set\_xlabel('学生姓名')  
ax2.set\_ylabel('分数')  
ax2.set\_yticks(np.arange(0, 131, 10))  
ax2.set\_title('学生数理化成绩的直方图')  
ax2.legend(['数学', '物理', '化学'])  
  
# 在柱状图上标注数值  
for x, y in zip(positions, math\_scores):  
 ax2.text(x, y + 1, '%d' % y, ha='center', va='bottom', fontsize=8)  
for x, y in zip(positions, physics\_scores):  
 ax2.text(x + width, y + 1, '%d' % y, ha='center', va='bottom', fontsize=8)  
for x, y in zip(positions, chemistry\_scores):  
 ax2.text(x + 2 \* width, y + 1, '%d' % y, ha='center', va='bottom', fontsize=8)  
  
# 保存并显示图形  
plt.tight\_layout()  
plt.savefig('score.png')  
plt.show()**

#### 运行结果：

****

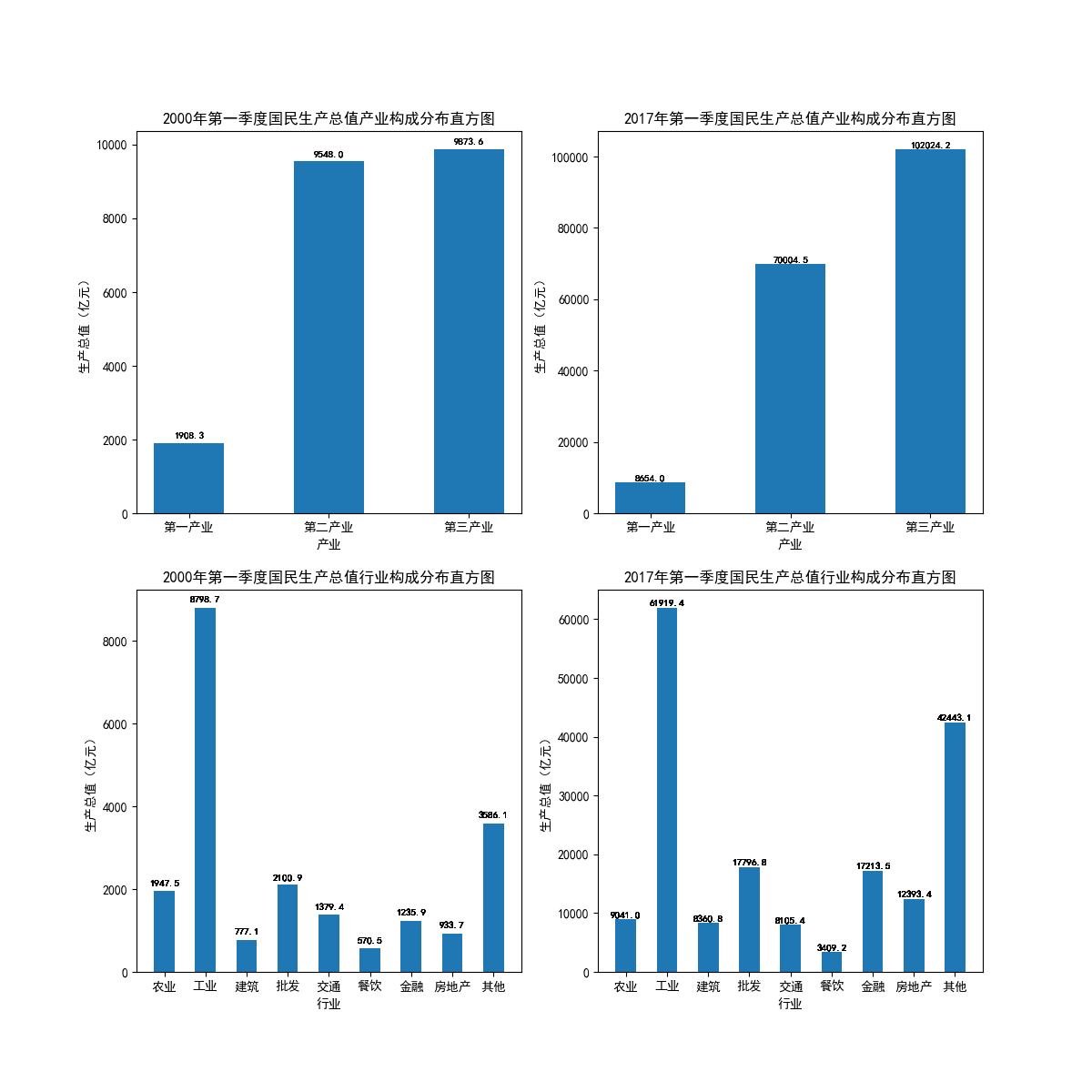
## 实验题目四：

**将第3章ppt中61页绘制直方图程序修改，使每个直方图柱顶都显示相关的值。本题数据集下载附件”国民经济核算季度数据.npz”。**

#### 程序源代码：

**import numpy as np  
import matplotlib.pyplot as plt  
  
data = np.load('国民经济核算季度数据.npz', allow\_pickle=True)  
name = data['columns'] # 提取其中的columns数组，视为数据的标签  
values = data['values'] # 提取其中的values数组，数据的存在位置  
  
plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文显示  
plt.rcParams['axes.unicode\_minus'] = False  
  
label1 = ['第一产业', '第二产业', '第三产业'] # 刻度标签1  
label2 = ['农业', '工业', '建筑', '批发', '交通', '餐饮', '金融', '房地产', '其他'] # 刻度标签2  
  
p = plt.figure(figsize=(12, 12))  
  
  
def autolabel(rects): # 添加标签的自定义函数  
 for rect in rects:  
 height = rect.get\_height()  
 plt.text(rect.get\_x() + rect.get\_width() / 2., # x坐标  
 height + 100, # y坐标  
 '{:.1f}'.format(height), # 展示的文本内容  
 ha='center', # 水平对齐方式  
 va='bottom', # 垂直对齐方式  
 fontsize=8) # 字体大小  
  
  
# 子图1  
ax1 = p.add\_subplot(2, 2, 1)  
bar1 = plt.bar(range(3), values[0, 3:6], width=0.5) # 绘制直方图,values[0,3:6]表示第一行第4,5,6列  
plt.xlabel('产业') # 添加横轴标签  
plt.ylabel('生产总值（亿元）') # 添加y轴名称  
plt.xticks(range(3), label1)  
plt.title('2000年第一季度国民生产总值产业构成分布直方图')  
for x, y in zip(range(3), values[0, 3:6]):  
 plt.text(x, y + 100, '{:.1f}'.format(y), ha='center', va='bottom', fontsize=8)  
 # 在柱状图上方加上数量  
 autolabel(bar1) # 调用autolabel函数添加标签  
  
# 子图2  
ax2 = p.add\_subplot(2, 2, 2)  
bar2 = plt.bar(range(3), values[-1, 3:6], width=0.5) # 绘制直方图，values[-1,3:6]表示最后一行第4,5，6列  
plt.xlabel('产业') # 添加横轴标签  
plt.ylabel('生产总值（亿元）') # 添加y轴名称  
plt.xticks(range(3), label1)  
plt.title('2017年第一季度国民生产总值产业构成分布直方图')  
for x, y in zip(range(3), values[-1, 3:6]):  
 plt.text(x, y + 100, '{:.1f}'.format(y), ha='center', va='bottom', fontsize=8)  
 # 在柱状图上方加上数量  
 autolabel(bar2) # 调用autolabel函数添加标签  
  
# 子图3  
ax3 = p.add\_subplot(2, 2, 3)  
bar3 = plt.bar(range(9), values[0, 6:], width=0.5) # 绘制直方图,values[0,6:]表示第一行从第7列开始到最后  
plt.xlabel('行业') # 添加横轴标签  
plt.ylabel('生产总值（亿元）') # 添加y轴名称  
plt.xticks(range(9), label2)  
plt.title('2000年第一季度国民生产总值行业构成分布直方图') # 添加图表标题  
for x, y in zip(range(9), values[0, 6:]):  
 plt.text(x, y + 100, '{:.1f}'.format(y), ha='center', va='bottom', fontsize=8)  
 # 在柱状图上方加上数量  
 autolabel(bar3) # 调用autolabel函数添加标签  
  
# 子图4  
ax4 = p.add\_subplot(2, 2, 4)  
bar4 = plt.bar(range(9), values[-1, 6:], width=0.5) # 绘制直方图,values[-1,6:]表示最后一行从第7列开始直到最后  
plt.xlabel('行业') # 添加横轴标签  
plt.ylabel('生产总值（亿元）') # 添加y轴名称  
plt.xticks(range(9), label2)  
plt.title('2017年第一季度国民生产总值行业构成分布直方图') # 添加图表标题  
for x, y in zip(range(9), values[-1, 6:]):  
 plt.text(x, y + 100, '{:.1f}'.format(y), ha='center', va='bottom', fontsize=8)  
 # 在柱状图上方加上数量  
 autolabel(bar4) # 调用autolabel函数添加标签  
  
# 保存并显示图形  
plt.savefig('gdp.png')  
plt.show()**

#### 运行结果：



## 实验题目五：

**使用Numpy和Matploitlib实现下列功能：从附件中下载mndata.npz。导入数据集mndata.npz，该数据集共有三个数组，分别是name、class\_name、scores。name数据集是一维数组，存放15位学生姓名。class\_ name数据集是一维数组，存放7门课程名称。 scores数据集是二维数组，15行7列分别表示15位学生、7门课程，即一行为某学生取得的分数，一列为某课程的分数。**

**1）创建画布，绘制三幅子图：直方图、箱线图和饼图**

**2）在编号为1的子图上绘制直方图，根据学生成绩数据分布，绘制数理化三门成绩的直方图，将五位学生姓名（下标5、6、7、8、9）显示在X轴，取下标0、2、4的学生数据绘制直方图。**

**3）在编号为2的子图上绘制箱线图，根据课程成绩数据分布，一门课程一个箱线图，子图1中绘制下标0、2、4课程的箱线图。**

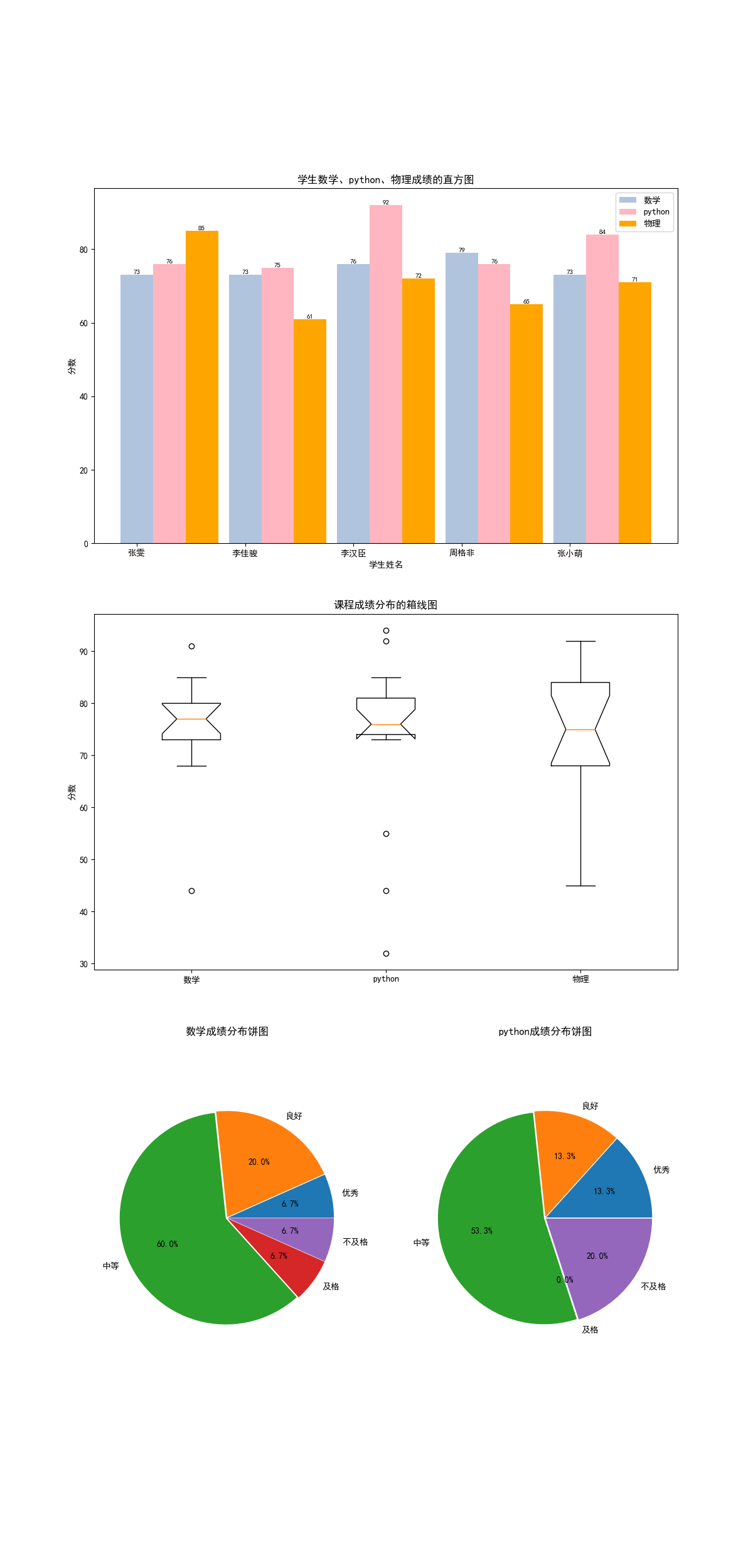
**4）在编号为3的子图上绘制饼图，根据学生成绩数据，绘制下标0、2课程成绩饼图。将课程成绩分为五个档次：优秀（90~100）、良好(80~89)、中等(70~79)、及格(60~69)、不及格(0~59)。**

**5）保存并显示图形。**

#### 程序源代码：

**import numpy as np  
import matplotlib.pyplot as plt  
  
# 设置中文显示  
plt.rcParams['font.sans-serif'] = 'SimHei'  
plt.rcParams['axes.unicode\_minus'] = False  
  
# 导入数据集  
data = np.load('mndata.npz', allow\_pickle=True)  
class\_name = data['class\_n']  
student\_name = data['student\_n']  
scores = data['score']  
  
# 创建画布  
fig = plt.figure(figsize=(12, 25))  
  
# 绘制子图1：直方图  
ax1 = fig.add\_subplot(3, 1, 1)  
width = 0.3  
X = np.arange(len(student\_name[5:10]))  
ax1.bar(X, scores[5:10, 0], width=width, facecolor='lightsteelblue')  
ax1.bar(X + 0.3, scores[5:10, 2], width=width, facecolor='lightpink')  
ax1.bar(X + 0.6, scores[5:10, 4], width=width, facecolor='orange')  
plt.xlabel('学生姓名')  
plt.ylabel('分数')  
label = student\_name[5:10]  
plt.xticks(range(5), label)  
plt.title('学生数学、python、物理成绩的直方图')  
plt.legend(['数学', 'python', '物理'])  
  
# 在每个柱子上标上数字  
math = scores[5:10, 0]  
physics = scores[5:10, 4]  
python = scores[5:10, 2]  
for x, y in zip(X, math):  
 plt.text(x, y, '%d' % y, ha='center', va='bottom', fontsize=8)  
for x, y in zip(X, python):  
 plt.text(x + width, y, '%d' % y, ha='center', va='bottom', fontsize=8)  
for x, y in zip(X, physics):  
 plt.text(x + width \* 2, y, '%d' % y, ha='center', va='bottom', fontsize=8)  
  
# 绘制子图2：箱线图  
ax2 = fig.add\_subplot(3, 1, 2)  
label1 = ['数学', 'python', '物理']  
number1 = [list(scores[:, i]) for i in range(0, 5, 2)]  
ax2.boxplot(number1, notch=True, labels=label1, meanline=True)  
plt.ylabel('分数')  
plt.title('课程成绩分布的箱线图')  
  
# 绘制子图3：饼图  
ax3 = fig.add\_subplot(3, 2, 5)  
plt.axis('equal')  
math1 = scores[:, 0]  
python1 = scores[:, 2]  
math1.sort()  
python1.sort()  
math\_grade = np.array([1, 3, 9, 1, 1])  
python\_grade = np.array([2, 2, 8, 0, 3])  
label2 = ['优秀', '良好', '中等', '及格', '不及格']  
explode = [0.01] \* 5  
ax3.pie(math\_grade, explode=explode, labels=label2, autopct='%1.1f%%')  
plt.title('数学成绩分布饼图')  
  
# 绘制子图4：饼图  
ax4 = fig.add\_subplot(3, 2, 6)  
plt.axis('equal')  
ax4.pie(python\_grade, explode=explode, labels=label2, autopct='%1.1f%%')  
plt.title('python成绩分布饼图')  
  
# 保存并显示图形  
plt.savefig('grades.png')  
plt.show()**

#### 运行结果：

****

# **实验小结**

**在本次matplotlib绘图实验中，我学会了使用Python通过这个强大的可视化库生成各种图形，包括散点图、折线图、柱状图、饼图等。同时，我也学会了对各种图形进行定制化，如添加标题、标签、轴标尺、颜色、图例等，使得图表更加直观、美观。**

**此外，我还掌握了自定义函数的方法，用于实现某种特定的需求，如在直方图上添加标签等。通过这次实验，我对于matplotlib绘图库的应用有了更深入的了解和认识，这对我今后在数据处理中的应用会有很大的帮助。**