Python 程序设计: 数据可视化作业
作业报告

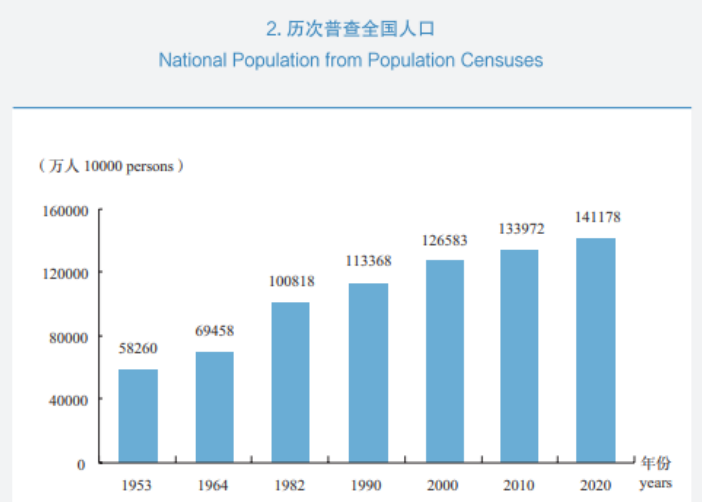
吴镇均
2020211448
北京邮电大学　计算机科学与技术

2022 年 12 月 25 日

1　作业 1

1.1　题目要求

依据图标信息绘制历次人口普查全国人口数量柱状图。



单位：万人　(10000 persons)

普查年份 Census Years	1953	1964	1982	1990	2000	2010	2020
全国人口 National Population	58260	69458	100818	113368	126583	133972	141178

图 1: 作业 1 图

1.2　核心代码

```
1 import matplotlib.pyplot as plt
```

```
2
3  year = ['1953年', '1964年', '1982年', '1990年', '2000年', '2010年', "2020年"]
4  population = [58260, 69458, 100818, 113368, 126583, 133972, 141178]
5  plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示中文
6  plt.grid(axis='y', which='major')
7  plt.xlabel('年份 years')
8  plt.ylabel(' (万人 10000 persons) ')
9  plt.title('历次普查全国人口')
10 plt.ticklabel_format(style='plain')
11 plt.subplots_adjust(left=0.15)
12 plt.bar(year, population, width=0.5, align='center', color='slateblue', bottom=0.8)
13 for x, y in zip(year, population):
14     plt.text(x, y, format(y, ','), ha='center', fontsize=9)
15 plt.legend(['人口 (万) '])
16 plt.savefig('visual_1.png')
17 plt.show()
```

1.3 生成图表

生成图表如下:

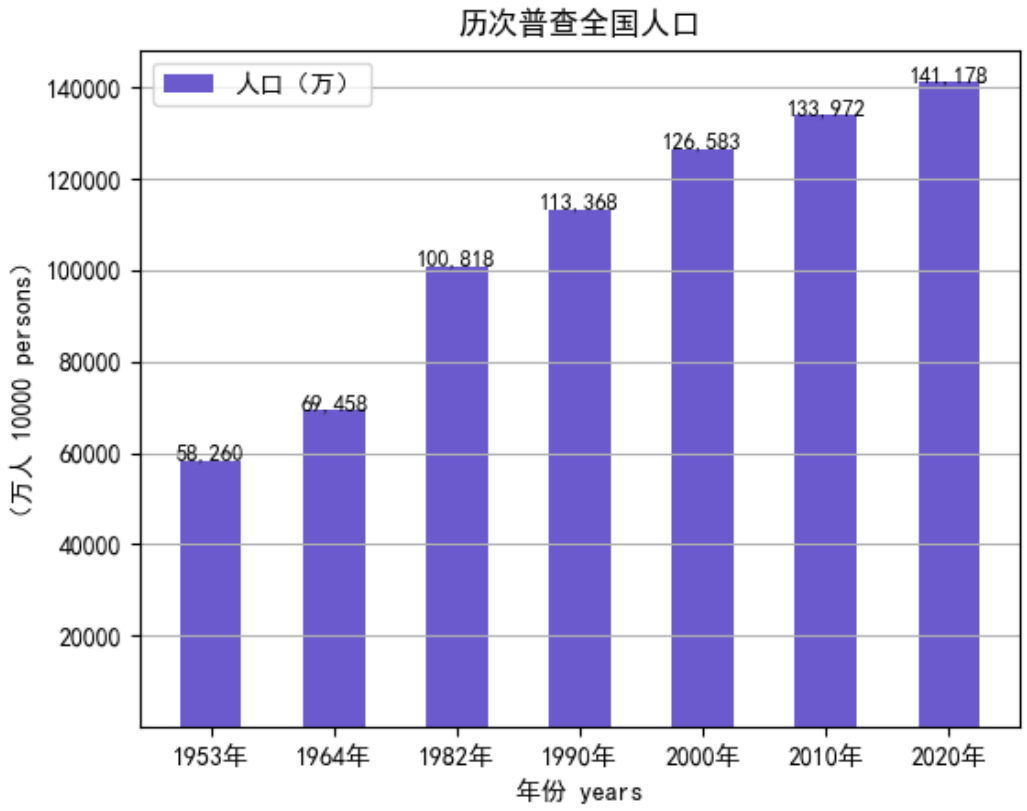


图 2: 生成图表

2 作业 2

2.1 题目要求

依据图信息绘制某人 2020 年支付宝年支出情况饼图，图中应展示各类型支出占总支出的比例。

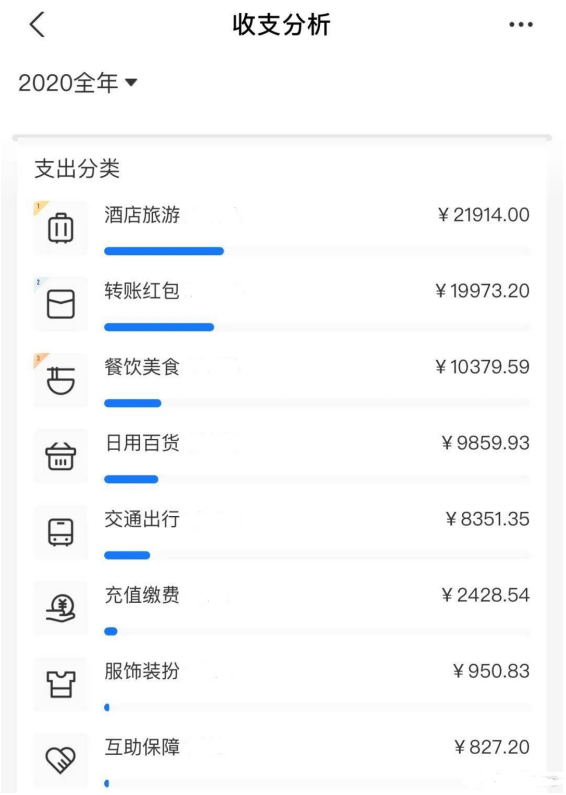


图 3: 作业 2 图

2.2 核心代码

```
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示中文
plt.figure(figsize=(8, 6.5))
items = ['酒店旅游', '转账红包', '餐饮美食', '日用百货', '交通出行', '充值缴费',
    '服饰装扮', '互助保障']
spendings = [21914.00, 19973.20, 10379.59, 9859.93, 8351.35, 2428.54, 950.83, 827.20]
colors = ['r', 'y', 'slateblue', 'g', 'm', 'cyan', 'darkorange', 'lawngreen']
plt.pie(spendings, labels=items, colors=colors, labeldistance=1.05,
    autopct='%.2f%%', textprops={'fontsize': 9, 'color': 'k'})
plt.title('2020年支付宝年支出情况')
plt.savefig('visual_2.png')
plt.show()
```

2.3 生成图表

生成图表如下：

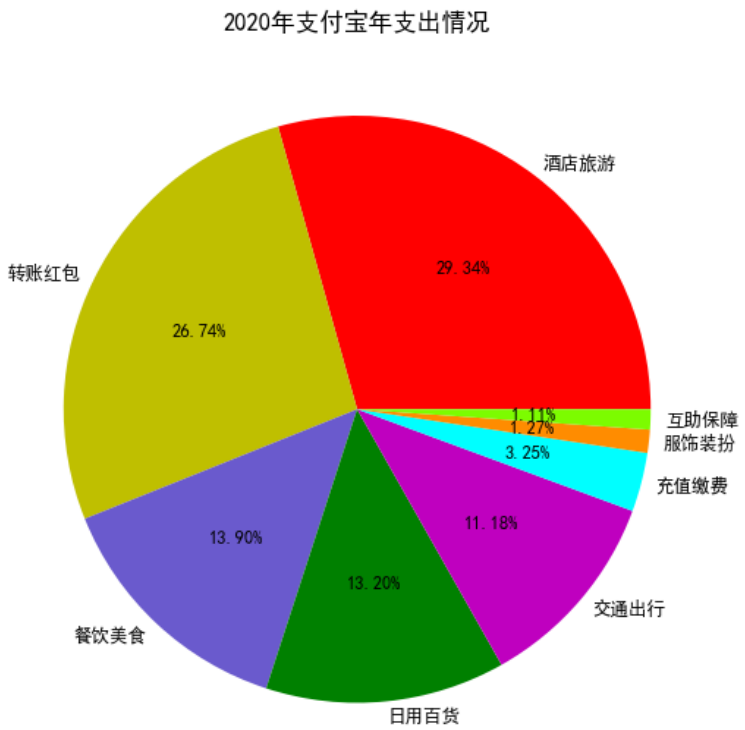


图 4: 生成图表

3 作业 3

3.1 题目要求

使用 IRIS 数据集，在一个 figure 中绘制出右侧的 16 个子图。分别使用花瓣长度、花瓣宽度、花萼长度和花萼宽度这四种数据，两两组合，绘制散点图。

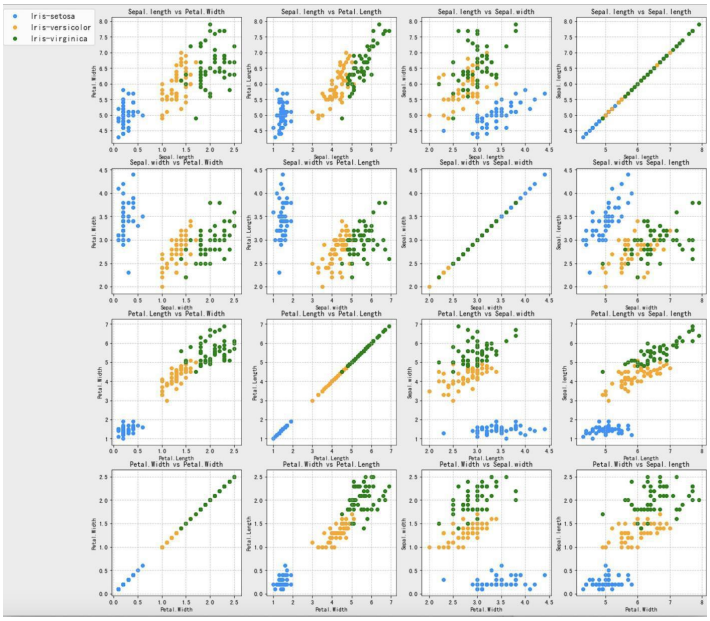


图 5: 作业 3 图

3.2 核心代码

首先采用了unique方法来来对种类进行去重处理，方便后续对不同种类的点进行上色及区分。同时也采用了subplots_adjust方法来调整子图之间的间距，避免内容重叠。

之后采用了遍历数组的方式来避免重复的代码编写。其中使用了xnames和ynames变量来存储题目中要求的属性。通过嵌套循环的形式达到两两配对的效果。并对每一种配对都生成了相应的子图。

最后采用了figlegend方法来生成全局图例。

```
1  import matplotlib.pyplot as plt
2  import pandas as pd
3
4  iris = pd.read_csv('iris.csv')
5  xnames = ['Sepal.Length', 'Sepal.Width', 'Petal.Length', 'Petal.Width']
6  ynames = ['Petal.Width', 'Petal.Length', 'Sepal.Width', 'Sepal.Length']
7
8  Species = iris['Species'].unique()
9  Colors = ['dodgerblue', 'orange', 'g']
10
11 fig, ax = plt.subplots(4, 4, figsize=(16, 16))
12
13 plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=0.5,
       hspace=0.5)
14
15 for x in range(len(xnames)):
16     for y in range(len(ynames)):
17         for i in range(len(Species)):
18             ax[x][y].scatter(iris.loc[iris['Species'] == Species[i], xnames[x]],
                   iris.loc[iris['Species'] == Species[i], ynames[y]], s=7, c=Colors[i],
```

```
                label=Species[i])
            ax[x][y].set_title(xnames[x] + ' vs ' + ynames[y])
            ax[x][y].grid(True)
            ax[x][y].set_xlabel(xnames[x])
            ax[x][y].set_ylabel(ynames[y])
labelname = []
for i in range(len(Species)):
    labelname.append('lris-' + Species[i])
plt.figlegend(labels=labelname, loc='upper left', bbox_to_anchor=(-0., 0.4, 0.5,
    0.5))
plt.savefig('visual_3.png')
plt.show()
```

3.3 生成图表

生成图表如下：

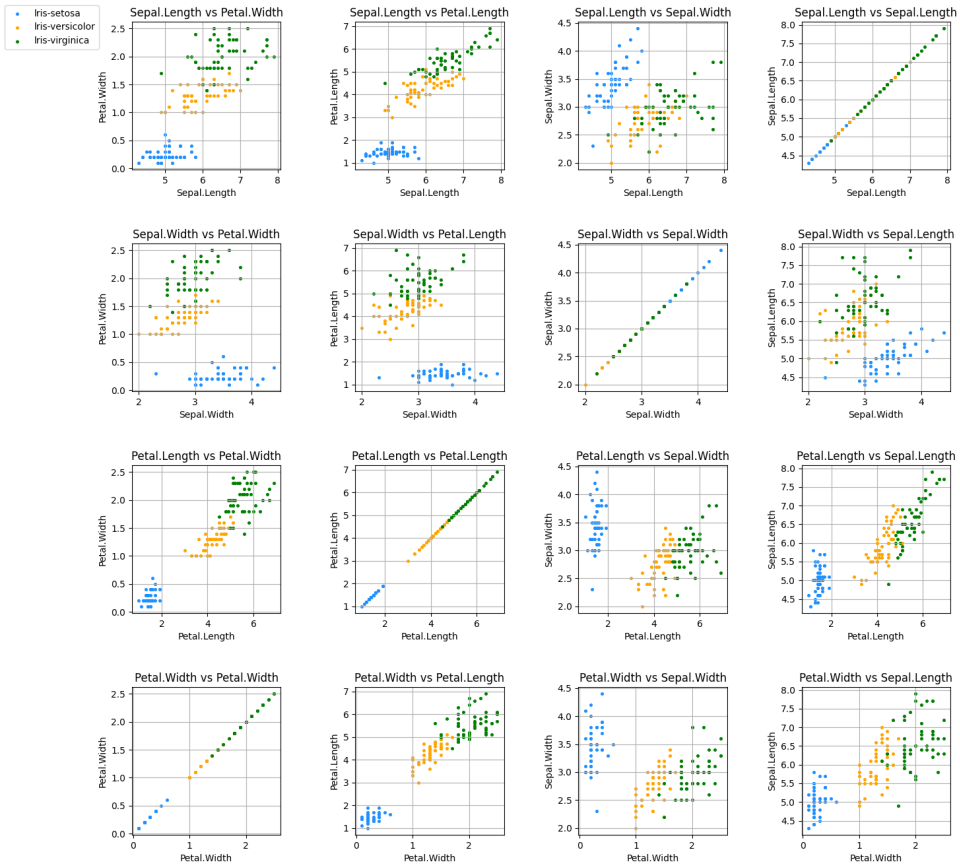


图 6: 生成图表

4 作业 4

4.1 题目要求

使用给出的“八年级期末考试成绩表.xlsx”，在一个 figure 中绘制六个子图，分别绘制六门课程的成绩分段统计情况直方图，每 10 分一个分段。

4.2 核心代码

这里选择了地理、历史、政治、生物、物理、英语六门课程作为子图的生成数据。

其中，为了贴合现实情况，将前五门课程的区间设置为了 0~100，将英语课程成绩的区间设置为了 0~120 。

在核心代码中，首先使用了`subplots_adjust`方法来调整子图之间的间距，避免子图内容重叠，影响阅读。

之后采用了与作业 3 代码中类似的方法，通过遍历 `courses`数组来分别生成对应子图，并标上相应标签和标题等。

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

df = pd.read_excel('八年级期末考试成绩表.xlsx')
plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示中文
courses = ['地理分数', '历史分数', '政治分数', '生物分数', '物理分数', '英语分数']

fig, ax = plt.subplots(2, 3, figsize=(20, 10))
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=0.5,
    hspace=0.45)
for i in range(len(courses)):
    x = int(i / 3)
    y = int(i % 3)
    ax[x][y].set_xlabel('分数', fontsize=12)
    ax[x][y].set_ylabel('学生数量', fontsize=12)
    ax[x][y].set_title('八年级期末考试' + courses[i] + '成绩分布')
    if courses[i] == '英语分数':
        ax[x][y].hist(df[courses[i]], 12, (0, 120), facecolor='blue',
            edgecolor='black')
    else:
        ax[x][y].hist(df[courses[i]], 10, (0, 100), facecolor='blue',
            edgecolor='black')

plt.savefig('visual_4.png')
plt.show()
```

4.3 生成图表

生成图表如下：

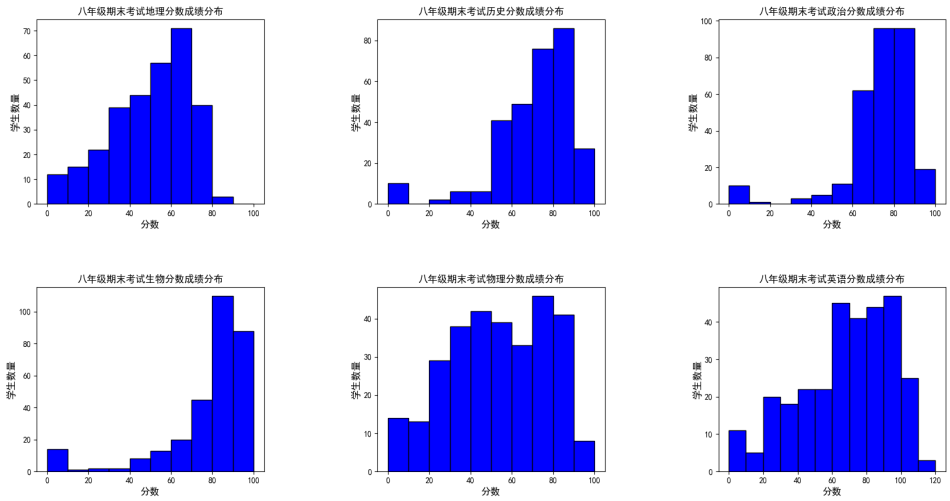


图 7: 生成图表

5 作业 5

5.1 题目要求

使用 BeijingPM20100101_20151231.csv 数据集，展示北京市 2010 2015 年 PM2.5 指数月平均数据的变化情况，在同一幅图中绘制六条折线，每年一条折线

5.2 核心代码

本作业代码主要分为两个部分，一是对数据进行预处理获取每年每月的平均数，二是利用获取到的数据生成对应的折线图像。

数据预处理 在该部分，首先使用了`numpy`包中提供的方法生成了二维数组用于存储每年每月对应的平均数。之后根据年份和月份遍历读取到的数据。

其中，**会将 PM 值为空值的部分默认为当年当月其他数据的平均值**。采用这种方式来减小空值对最后数据的影响。具体做法则是直接调用`pd.mean`方法，该方法会自动忽略为空的项，并对非空的项取均值，也就是默认为空项的值为非空项的均值。

而如果某年某月中某地点的 PM 记录值均为空，那么会用当年当月其他地点的平均值来作为该地的平均值。

采用这种空值处理方法后，会求出不同地点的当年当月的 PM 值均值，然后对这些值求总平均，并将该值作为此年此月的 PM 平均值，存储到二维数组中。

图像生成 依次生成 2010-2015 年相应的图像，并尽可能地对每一年的折线图采用不同的颜色和图例，以达到一定的区分效果。

```
1 import matplotlib.pyplot as plt
```

```
import pandas as pd
import numpy as np

months = ['一月', '二月', '三月', '四月', '五月', '六月', '七月', '八月', '九月',
    '十月', '十一月', '十二月']
years = ['2010年', '2011年', '2012年', '2013年', '2014年', '2015年']
df = pd.read_csv('BeijingPM20100101_20151231.csv')

matrix = np.zeros((6, 12))
plt.rcParams['font.sans-serif'] = ['SimHei'] # 显示中文
# 获取每年每月的 PM 平均数
for year in range(2010, 2016):
    for month in range(1, 13):
        x = year - 2010
        y = month - 1
        year_mask = df['year'] == year
        month_mask = df['month'] == month
        mask = year_mask & month_mask
        data = df.loc[mask]
        a = []
        if not np.isnan(data['PM_Dongsi'].mean()):
            a.append(data['PM_Dongsi'].mean())
        if not np.isnan(data['PM_Nongzhanguan'].mean()):
            a.append(data['PM_Nongzhanguan'].mean())
        if not np.isnan(data['PM_Dongsihuan'].mean()):
            a.append(data['PM_Dongsihuan'].mean())
        if not np.isnan(data['PM_US Post'].mean()):
            a.append(data['PM_US Post'].mean())
        ave = np.mean(a)
        matrix[x][y] = ave

plt.xlabel('月份')
plt.ylabel('PM (ug/m^3)')
colors = ['.r-', ',y--', 'oc-.', '^g:', '1m-', 'sb--']
lines = []
for i in range(6):
    p, = plt.plot(months, matrix[i], colors[i])
    lines.append(p)
plt.legend(lines, years, loc='upper right')
plt.grid(linestyle='--')
plt.tick_params(axis='y', direction='in', color='r', grid_color='r')
plt.savefig('visual_5.png')
plt.show()
```

5.3 生成图表

生成图表如下：

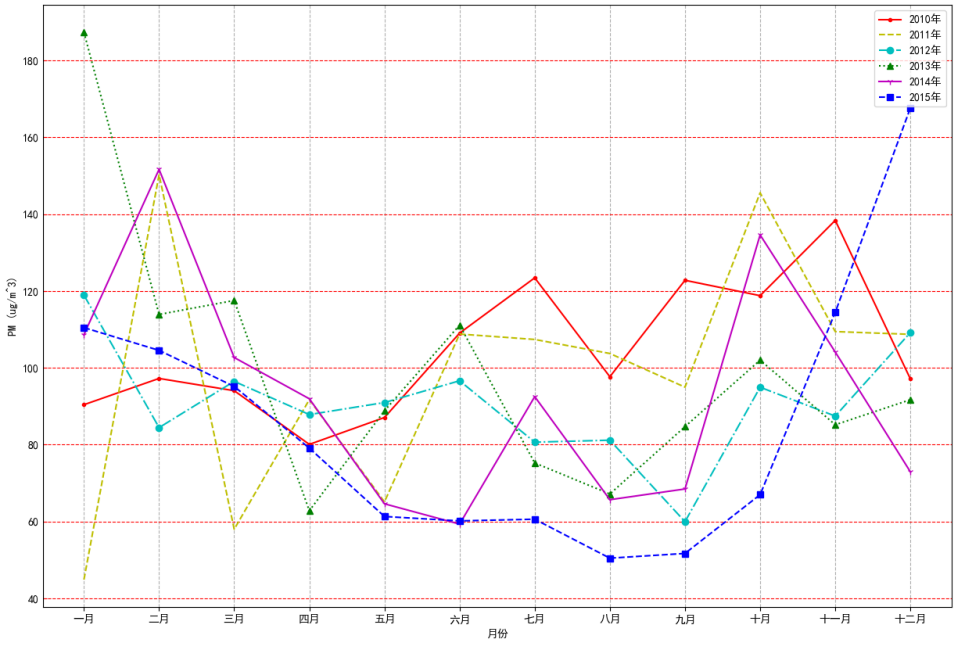

2010年
2011年
2012年
2013年
2014年
2015年
PM（μg/m^3）
180
160
140
120
100
80
60
40
一月
二月
三月
四月
五月
六月
七月
八月
九月
十月
十一月
十二月
月份

图 8: 生成图表