
北京邮电大学

Beijing University of Posts and Telecommunications

实 验 报 告

题目：数据可视化-作业 1

课 程 名 称：Python 程序设计

姓 名（学号）：凌国瀚 (2018213344)

学 院：计算机学院

专 业：网络工程

指 导 教 师：杨亚老师

二〇二〇年 十二月

目录

实 验 报 告	1
1. 题目要求.....	3
1.1 实验内容一：修改图片.....	3
1.2 实验内容二：分段统计数据.....	3
1.3 实验内容三：对比排名数据.....	3
1.4 实验内容四：调整线图.....	4
1.5 实验内容五：用线图展示北京空气质量数据.....	4
2. 实验内容一.....	4
3. 实验内容二.....	5
4. 实验内容三.....	6
5. 实验内容四.....	6
6. 实验内容五.....	7
7. 源代码.....	8
7.1 第一部分.....	8
7.2 第二部分.....	9
7.3 第三部分.....	10
7.4 第四部分.....	11
7.5 第五部分.....	12

1. 题目要求

1.1 实验内容一：修改图片

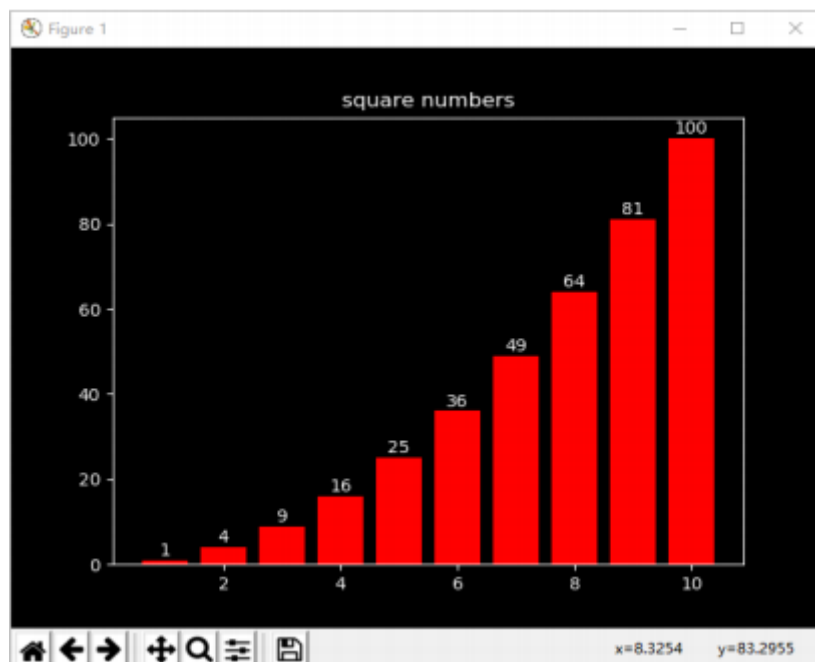


图 1 示例图片 1

1. 请更换图形的风格
2. 请将 x 轴的数据改为-10 到 10
3. 请自行构造一个 y 值的函数
4. 请将直方图上的数字，位置改到柱形图的内部垂直居中的位置

1.2 实验内容二：分段统计数据

对成绩数据 data1402.csv 进行分段统计：每 5 分作为一个分数段，展示出每个分数段的人数直方图。

1.3 实验内容三：对比排名数据

自行创建出 10 个学生的 3 个学期排名数据，并通过直方图进行对比展示。

1.4 实验内容四：调整线图

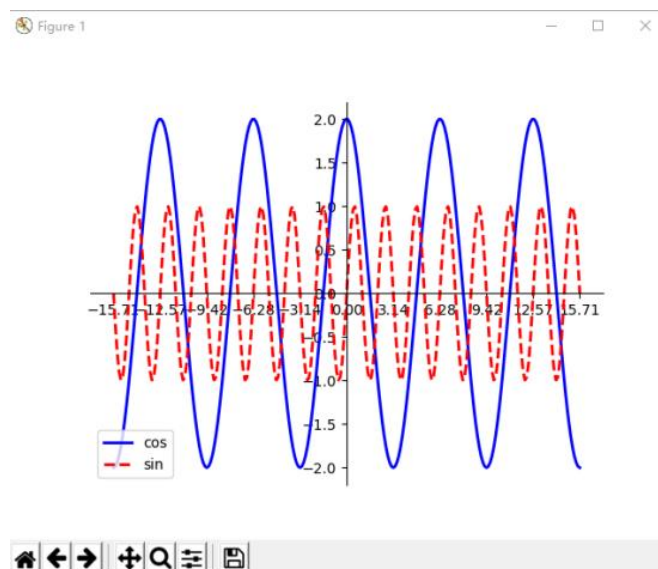


图 2 实例图片 2

- 1.把这个图像做一些调整，要求出现 5 个完整的波峰。
- 2.调大 cos 波形的幅度
- 3.调大 sin 波形的频率

1.5 实验内容五：用线图展示北京空气质量数据

展示 10-15 年 PM 指数月平均数据的变化情况，一幅图中有 6 条曲线，每年 1 条曲线。

2.实验内容一

更换图像风格为 **bmh**，x 轴数据范围变为 $[-10, 10]$ ，y 值对应函数为 $y = x^3$ ，直方图数字改为图像柱体垂直居中位置。最终得到图像如下：

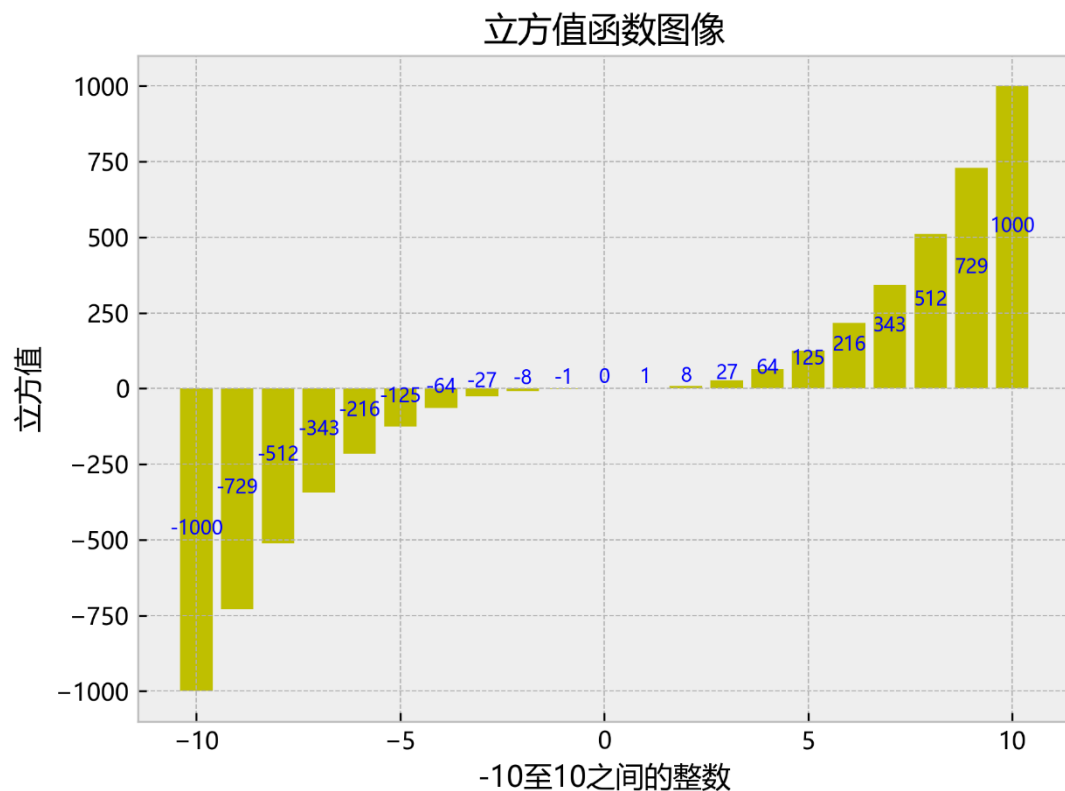


图 3 直方图修改结果

3.实验内容二

以 5 分为分段长度，将 100 分的分数范围分为 20 段，并删去其中没有数据的分数段，得到如下图像：

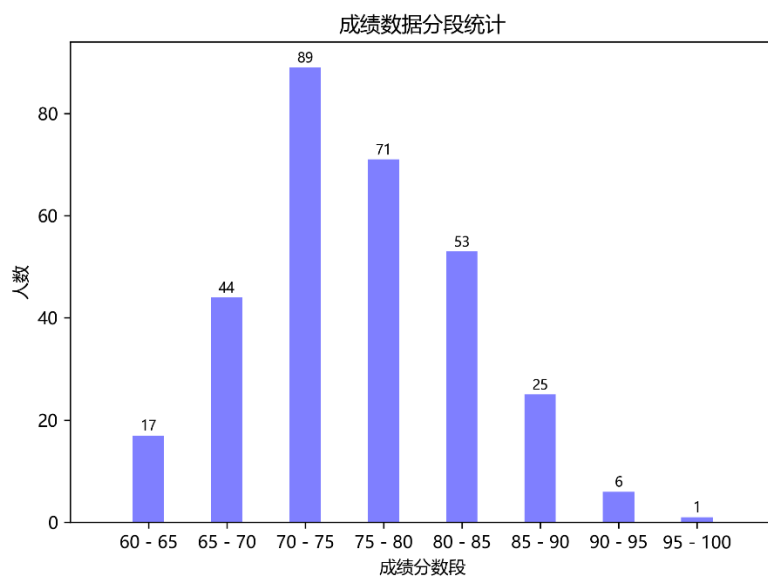


图 4 成绩数据分段统计图像

4.实验内容三

通过 random 库将排名随机，生成三个学期每个学生的排名，通过 matplotlib 的子图功能做出排名对比图如下：

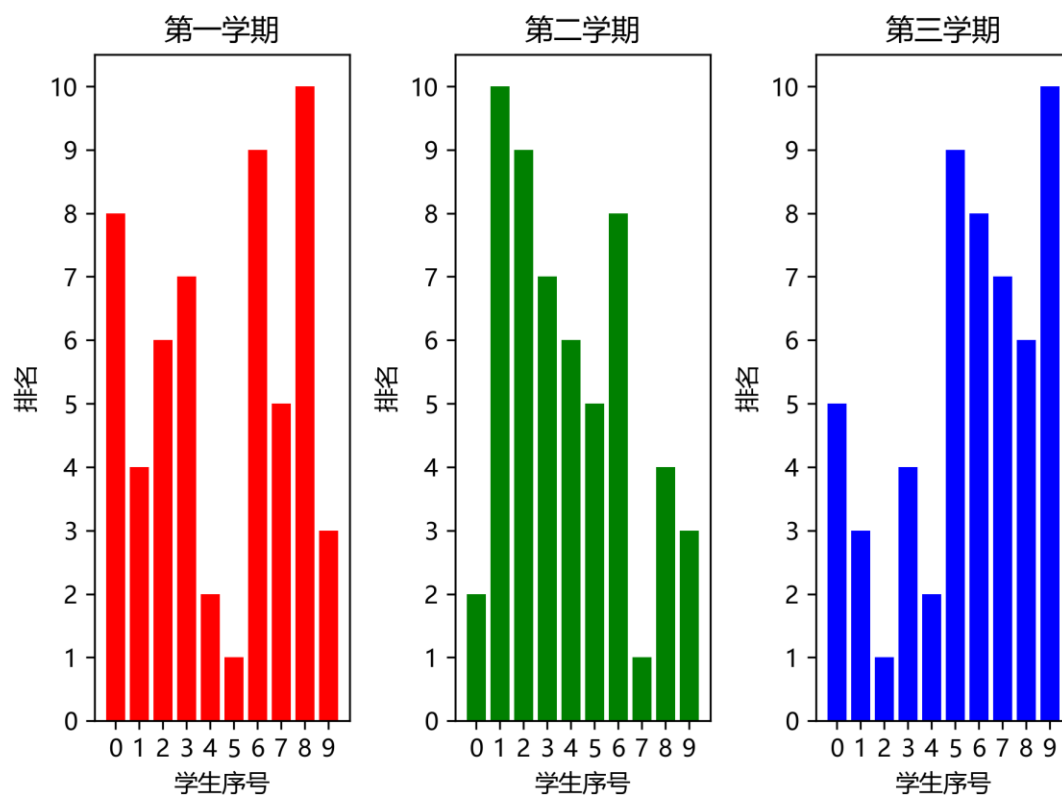


图 5 三学期排名对比图

5.实验内容四

通过调整 x 轴坐标范围，改变数据生成时横纵坐标数据的倍数，即可实现要求，得到的结果如下图：

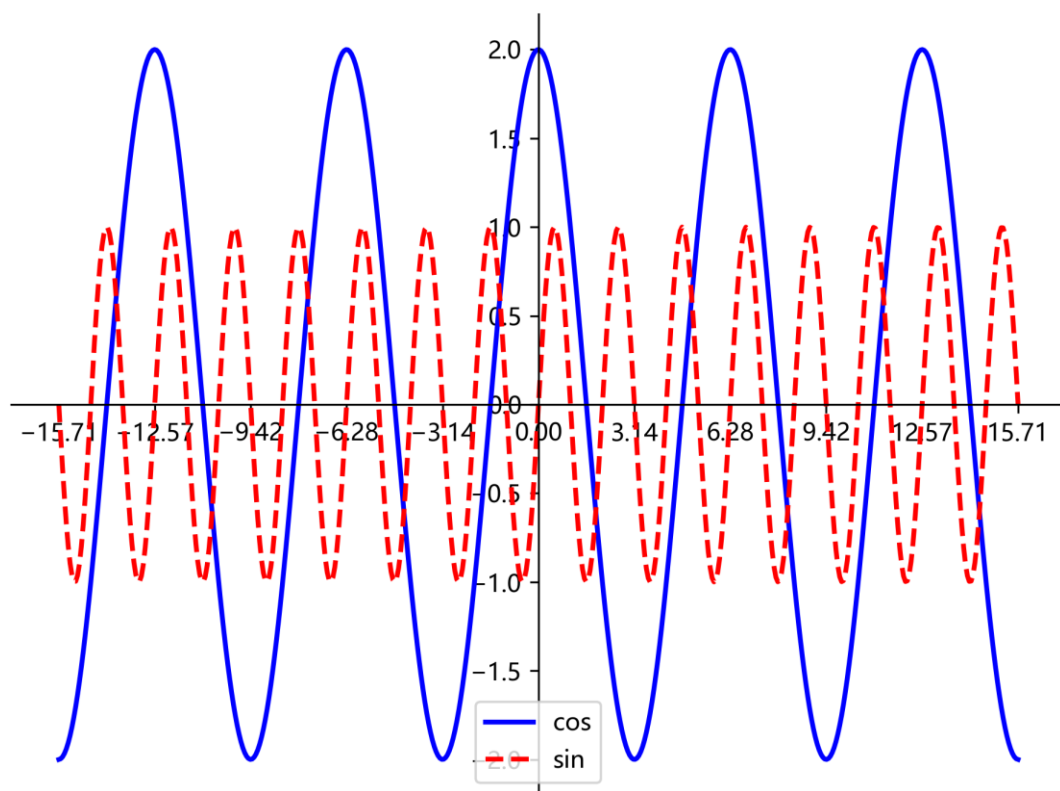


图 6 修改后的三角函数线图

6.实验内容五

利用 Pandas 处理数据（数据 csv 已经经过简化，删去了不必要的列），分别用不同颜色画出各个年份的线图，得到结果如下：

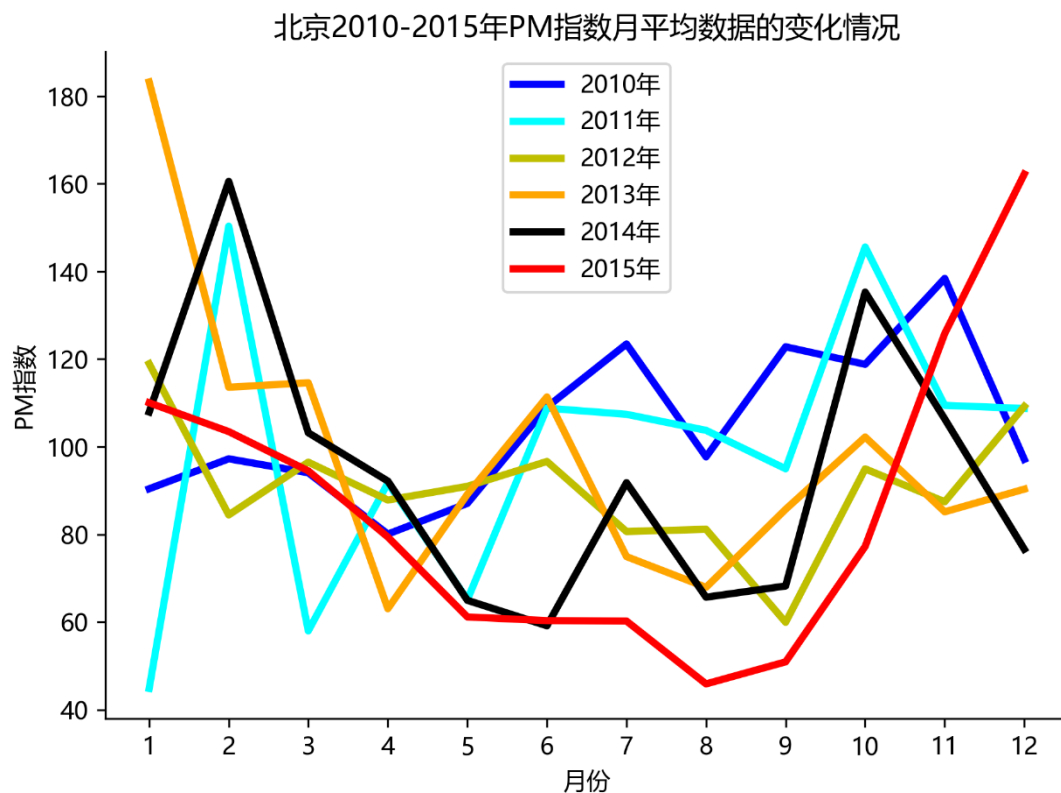


图 7PM 指数月平均数据变化图

7.源代码

7.1 第一部分

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

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 用来正常显示中文
标签
plt.rcParams['savefig.dpi'] = 300 # 图片像素
plt.rcParams['figure.dpi'] = 300 # 分辨率
plt.style.use('bmh')

fig, ax = plt.subplots()
ax.set_title("立方值函数图像")
x_list = [i for i in range(-10, 11)]
x = np.array(x_list) # 创建一个 numpy 数组 x
y = x ** 3 # 创建一个 numpy 数组 y, 内容为 x 中数据的立方值
plt.bar(x, y, color='y') # bar 的颜色改为红色
```



```

for a, b in zip(x, y): # 在直方图上显示数字
    plt.text(a, b / 2, '%d' % b, ha='center', va='bottom', fontsize=8,
             color='blue')
plt.xlabel('-10 至 10 之间的整数')
plt.ylabel('立方值')
plt.tight_layout()
plt.savefig('part1-1.png')
plt.show()

```

7.2 第二部分

```

import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('data1402.csv', encoding='utf-8', header=None, names=['Score'])
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 用来正常显示中文
标签
plt.rcParams['savefig.dpi'] = 300 # 图片像素
plt.rcParams['figure.dpi'] = 300 # 分辨率

sections = [i for i in range(101) if i % 5 == 0]
label = [sections[i + 1] for i in range(len(sections) - 1)]

result = pd.cut(df['Score'], sections, labels=label)
student_count = pd.value_counts(result, sort=False) # 得到各部分的数量

student_count_drop_zero = student_count.drop(student_count[student_count == 0].keys())
print(student_count_drop_zero)
print(student_count_drop_zero.describe())
print(student_count_drop_zero.values)
plt.axis([50, 100, 0, 30]) # 设置 x 轴和 y 轴的最小和最大值
plt.bar(student_count_drop_zero.keys(), student_count_drop_zero, 2, alp
ha=0.5, color='b')
plt.xlim(student_count_drop_zero.keys()[0] - 5, student_count_drop_zero
.keys()[-1] + 5)
plt.ylim(0, student_count_drop_zero.max() + 5)
plt.xticks(student_count_drop_zero.keys(),
            [str(item - 5) + ' - ' + str(item) for item in student_count
_drop_zero.keys()])
for a, b in zip(student_count_drop_zero.keys(), student_count_drop_zero
): # 在直方图上显示数字

```

```

plt.text(a, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=8,
, color='black')

plt.title('成绩数据分段统计')
plt.xlabel("成绩分数段")
plt.ylabel("人数")
# 绘制直方图，第三个参数表示直方图的宽度，alpha 为透明度,color 为颜色
plt.tight_layout()
plt.savefig('part1-2.png')
plt.show()

```

7.3 第三部分

```

import random

import matplotlib.pyplot as plt
import numpy as np
from matplotlib.ticker import MultipleLocator
from pandas import DataFrame

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 用来正常显示中文
标签
plt.rcParams['savefig.dpi'] = 300 # 图片像素
plt.rcParams['figure.dpi'] = 300 # 分辨率

base_list = [j for j in range(1, 11)]
list_data = []
for i in range(3):
    random.shuffle(base_list)
    list_data.append(base_list.copy())

# print(np.array(list_data).T)
df = DataFrame(np.array(list_data).T, columns=['第一学期', '第二学期', '
第三学期'])

print(df)
print(df.info())

plt.subplot(1, 3, 1)
plt.bar(df['第一学期'].keys(), df['第一学期'], color='r')
ax = plt.gca()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_major_locator(MultipleLocator(1))

```

```
plt.xlabel('学生序号')
plt.ylabel('排名')
plt.title('第一学期')

plt.subplot(1, 3, 2)
plt.bar(df['第二学期'].keys(), df['第二学期'], color='g')
ax = plt.gca()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_major_locator(MultipleLocator(1))
plt.xlabel('学生序号')
plt.ylabel('排名')
plt.title('第二学期')

plt.subplot(1, 3, 3)
plt.bar(df['第三学期'].keys(), df['第三学期'], color='b')
ax = plt.gca()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.yaxis.set_major_locator(MultipleLocator(1))
plt.xlabel('学生序号')
plt.ylabel('排名')
plt.title('第三学期')

plt.tight_layout()
plt.savefig('part1-3.png')
plt.show()
```

7.4 第四部分

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

plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 用来正常显示中文
标签
plt.rcParams['savefig.dpi'] = 300 # 图片像素
plt.rcParams['figure.dpi'] = 300 # 分辨率

fig, ax = plt.subplots()
x = np.linspace(-5 * np.pi, 5 * np.pi, 512) # 生成从-2π到2π的256个数据
cos, sin = np.cos(x), np.sin(3 * x) # 分别计算x的cos和sin函数值
ax.set_xticks([i * np.pi for i in range(-5, 6)]) # 设置x轴的刻度
plt.plot(x, 2 * cos, color="blue", linewidth=2, linestyle="-",
        label="cos") # 画出cos曲线，颜色为蓝色，线宽为2，连续线
```

```
plt.plot(x, sin, color="red", linewidth=2, linestyle="--", label="sin") # 画出 sin 曲线，颜色为红色，线宽为 2，间断线
# 画出十字形的坐标轴
ax.spines["right"].set_visible(False) # 隐藏右边框
ax.spines["top"].set_visible(False) # 隐藏上边框
ax.spines['bottom'].set_position(('data', 0)) # 设置下边框到 y 轴 0 的位置
ax.xaxis.set_ticks_position('bottom') # 刻度值设置在下方
ax.spines['left'].set_position(('data', 0)) # 设置左边框到 x 轴 0 的位置
ax.yaxis.set_ticks_position('left') # 刻度值设置在左侧
plt.legend(loc='best')
plt.tight_layout()
plt.savefig('part1-4.png')
plt.show()
```

7.5 第五部分

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

df = pd.read_csv('Beijing_reduced_data.csv', encoding='utf-8')
plt.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 用来正常显示中文
标签
plt.rcParams['savefig.dpi'] = 300 # 图片像素
plt.rcParams['figure.dpi'] = 300 # 分辨率

df_every_year_month = DataFrame()

fig, ax = plt.subplots()

for i in range(2010, 2016):
    df_temp = df[df['year'] == i]
    df_every_year_month[str(i)] = df_temp.groupby('month').mean()['average']

print(df_every_year_month.head())
print(df_every_year_month.info())

ax.set_xticks([i for i in range(1, 13)]) # 设置 x 轴的刻度
plt.plot(df_every_year_month['2010'].keys(), df_every_year_month['2010'], color="blue", linewidth=3, label="2010 年")
```

```
plt.plot(df_every_year_month['2011'].keys(), df_every_year_month['2011'],
         color="cyan", linewidth=3,
         label="2011 年")
plt.plot(df_every_year_month['2012'].keys(), df_every_year_month['2012'],
         color="y", linewidth=3,
         label="2012 年")
plt.plot(df_every_year_month['2013'].keys(), df_every_year_month['2013'],
         color="orange", linewidth=3,
         label="2013 年")
plt.plot(df_every_year_month['2014'].keys(), df_every_year_month['2014'],
         color="black", linewidth=3,
         label="2014 年")
plt.plot(df_every_year_month['2015'].keys(), df_every_year_month['2015'],
         color="r", linewidth=3,
         label="2015 年")

plt.title('北京 2010-2015 年 PM 指数月平均数据的变化情况')
plt.xlabel('月份')
plt.ylabel('PM 指数')

ax.spines["right"].set_visible(False) # 隐藏右边框
ax.spines["top"].set_visible(False) # 隐藏上边框
ax.xaxis.set_ticks_position('bottom') # 刻度值设置在下方
ax.yaxis.set_ticks_position('left') # 刻度值设置在左侧
plt.legend(loc='best')
plt.tight_layout()
plt.savefig('part1-5.png')
plt.show()
```