

Python 数据可视化作业 1

班级：2018211313 班

学号：2018211366

姓名：蒋潇逸

版本：1.0

更新：December 6, 2020

本文档是 Python 数据可视化作业 1 报告。

目录

1	要求 1	2
2	要求 2	3
3	要求 3	5
4	要求 4	7
5	要求 5	9

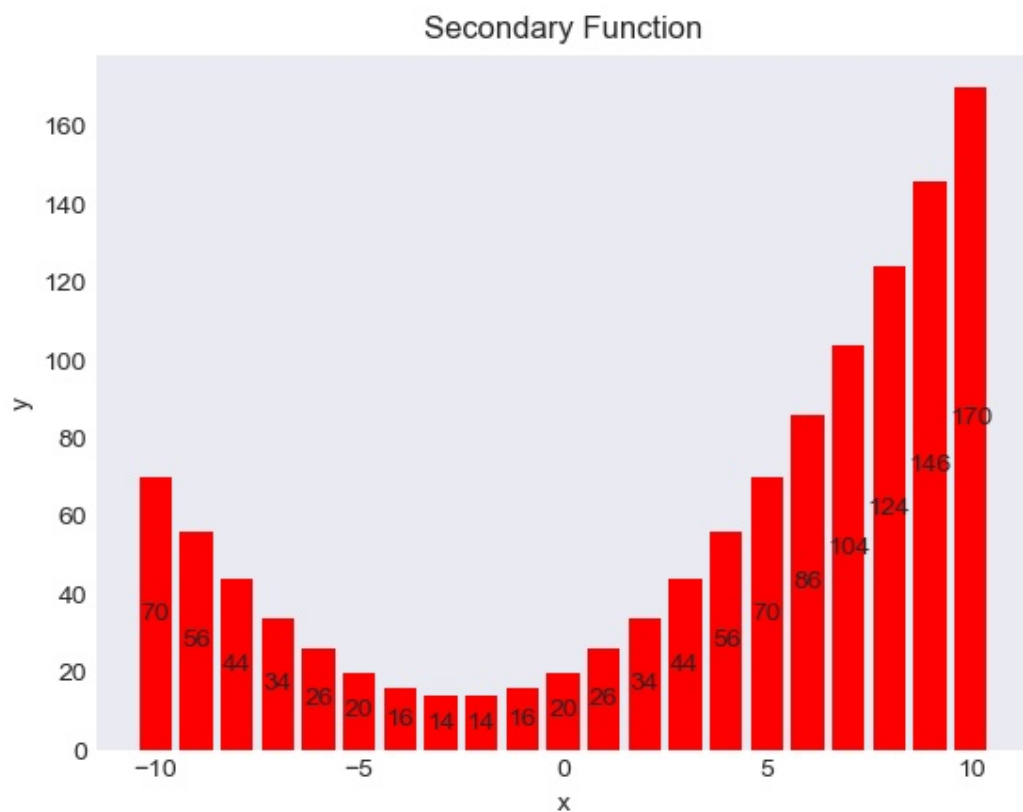
1 要求 1

对图片进行修改：

- 请更换图形的风格
- 请将 x 轴的数据改为-10 到 10
- 请自行构造一个 y 值的函数
- 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

代码如下

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 plt.style.use('seaborn-dark') # 设置图像风格
5 fig, ax = plt.subplots()
6 ax.set_title("Secondary Function")
7 # 设置坐标轴名称
8 plt.xlabel('x')
9 plt.ylabel('y')
10 x = []
11 for i in range(-10, 11): # 将x轴的数据改为-10到10
12     x.append(i)
13 x = np.array(x) # 创建一个numpy数组x
14 y = x * x + 5 * x + 20 # 自行构造一个y值的函数
15 plt.bar(x, y, color='r') # bar的颜色改为红色
16 for a, b in zip(x, y): # 在直方图上显示数字
17     # 位置改到柱形图的内部垂直居中的位置
18     plt.text(a, b / 2, '%d' % b, ha='center', va='center', fontsize
19              =10)
20 plt.savefig("C:\\Users\\Lenovo\\Desktop\\1.jpg")
21 plt.show()
```



2 要求 2

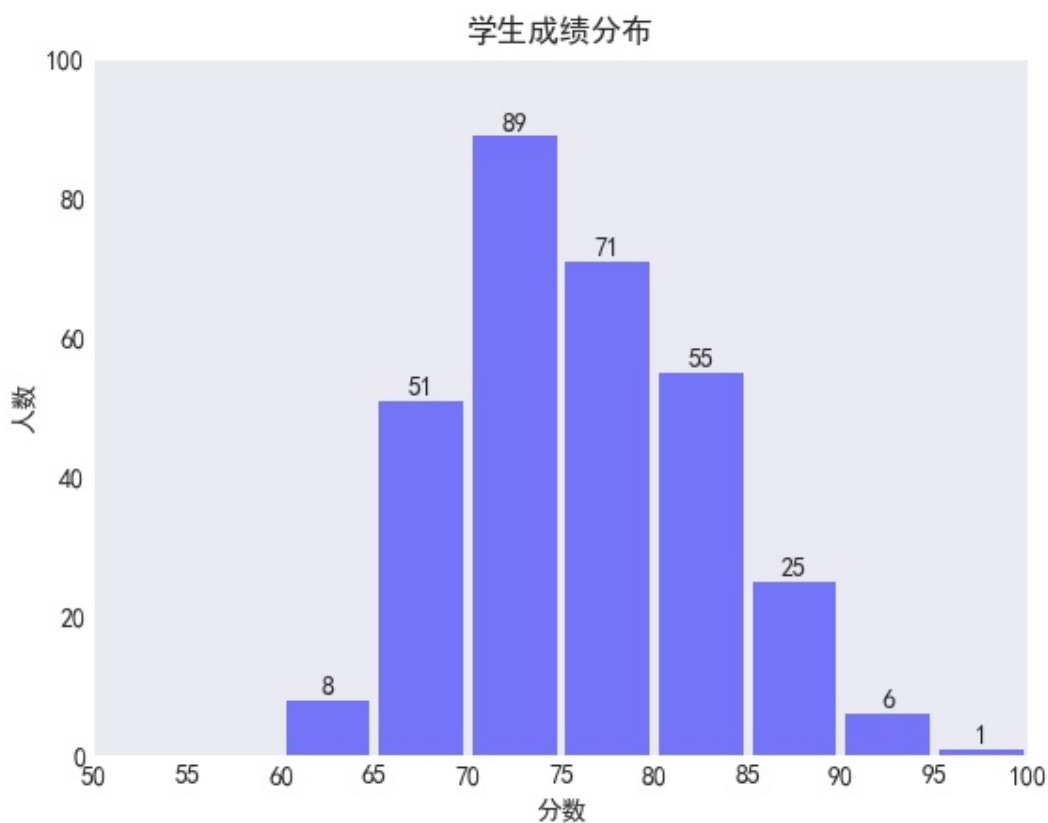
对成绩数据 `data1402.csv` 进行分段统计：每 5 分作为一个分数段，展示出每个分数段的人数直方图。

```
1 import csv
2 import matplotlib.pyplot as plt
3
4
5 def count_elements(scores):
6     scoreCnt = {}
7     for emp in scores:
8         section = emp // 5 * 5 # 判断所在区间
9         scoreCnt[section] = scoreCnt.get(section, 0) + 1
10    return scoreCnt
11
12
```

```

13 scores = [] # 列表对象
14 with open('data1402.csv', 'r') as csvFile:
15     f_csv = csv.reader(csvFile) # 读取文件
16     for row in f_csv:
17         scores.append(int(float(row[0])))
18 counted = count_elements(scores) # 字典 = {分数 : 人数}
19 plt.style.use('seaborn-dark') # 图像风格
20 fig, ax = plt.subplots()
21 plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
22 ax.set_title("学生成绩分布")
23 # 设置坐标轴名称
24 plt.xlabel('分数')
25 plt.ylabel('人数')
26 plt.axis([50, 100, 0, 100])
27 # 设置x轴区间, 以5为步长
28 plt.xticks([x for x in range(50, 101) if x % 5 == 0]) # x标记step设置为5
29
30 # 绘制频次直方图, 以5为步长
31 step = 5
32 bins = [x for x in range(50, 101, step)]
33 plt.hist(scores, bins, alpha=0.5, histtype='bar', rwidth=0.9, color='
    b')
34 for key in counted:
35     # 在直方图上显示数字
36     plt.text(key + 2.5, counted.get(key, 0) + 0.2, '%d' % counted.get
        (key, 0), ha='center', va='bottom', fontsize=10)
37 plt.savefig("C:\\Users\\Lenovo\\Desktop\\2.jpg")
38 plt.show()

```



3 要求 3

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。

```

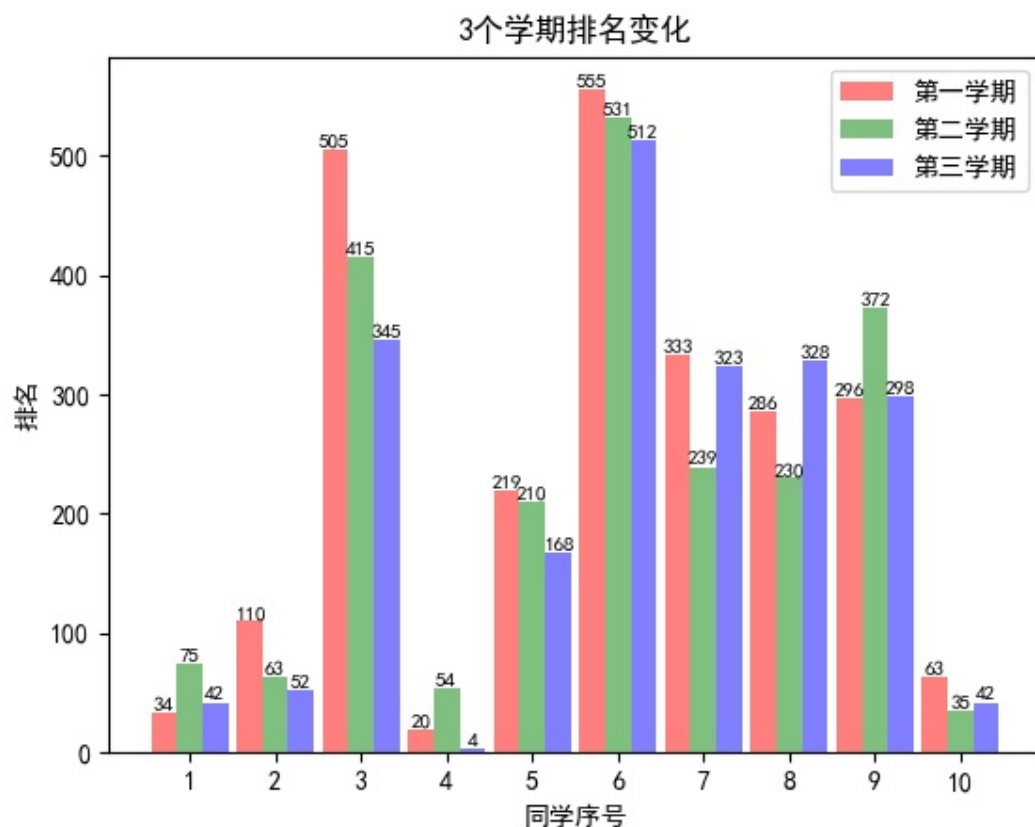
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.pyplot import MultipleLocator
4 from random import randint
5
6 ax = plt.axes() # 创建坐标对象
7 ax.set_title("3个学期排名变化") # 直方图标题
8 ax.set_ylabel('排名')
9 ax.set_xlabel('同学序号')
10 plt.rcParams['font.sans-serif'] = ['SimHei'] # 添加对中文字体的支持
11 semester1 = [] # 保存第一学期的排名
12 semester2 = [] # 保存第二学期的排名
13 semester3 = [] # 保存第三学期的排名

```

```

14 for i in range(10):
15     # 生成10名同学三个学期的排名,排名浮动100名, 且在[1,570]内
16     s1 = randint(1, 570)
17     s2 = s1 + randint(-100, 100)
18     while s2 <= 0 or s2 >= 570:
19         s2 = s1 + randint(-100, 100)
20     s3 = s2 + randint(-100, 100)
21     while s3 <= 0 or s3 >= 570:
22         s3 = s2 + randint(-100, 100)
23     semester1.append(s1)
24     semester2.append(s2)
25     semester3.append(s3)
26
27 # 创建三个直方图, 注意它们在x轴的位置, 不能重合
28 x = np.arange(1, 11) # 生成横轴数据
29 x_major_locator = MultipleLocator(1) # 显示刻度
30 ax.xaxis.set_major_locator(x_major_locator)
31 # 生成第一学期的排名直方图, 宽度为0.3
32 plt.bar(x - 0.3, semester1, 0.3, alpha=0.5, color='r', label='第一学
    期')
33 # 生成第二学期的排名直方图, 上一个直方图向右0.3宽度
34 plt.bar(x, semester2, 0.3, alpha=0.5, color='g', label='第二学期')
35 # 生成第三学期的排名直方图, 向右0.3宽度
36 plt.bar(x + 0.3, semester3, 0.3, alpha=0.5, color='b', label='第三学
    期')
37 # 在直方图上显示数字
38 for a, b in zip(x, semester1):
39     plt.text(a - 0.3, b + 0.2, b, ha='center', va='bottom', fontsize
        =8)
40 for a, b in zip(x, semester2):
41     plt.text(a, b + 0.2, b, ha='center', va='bottom', fontsize=8)
42 for a, b in zip(x, semester3):
43     plt.text(a + 0.3, b + 0.2, b, ha='center', va='bottom', fontsize
        =8)
44 # 显示图例
45 plt.legend(loc='best')
46 plt.savefig("C:\\Users\\Lenovo\\Desktop\\3.jpg")
47 plt.show()

```



4 要求 4

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

```

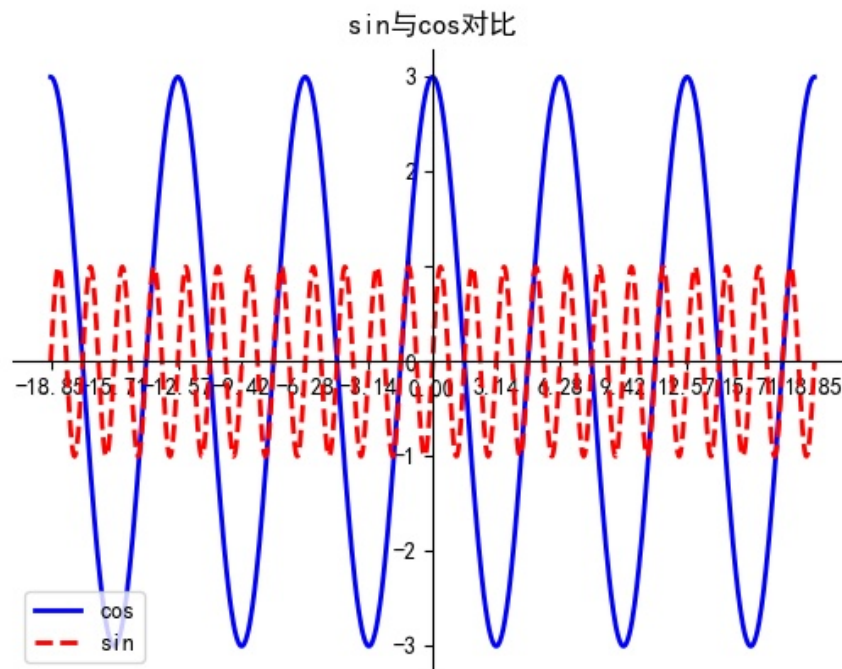
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 ax = plt.axes() # 创建坐标对象
5 plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
6 plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
7 ax.set_title("sin与cos对比") # 线图标题
8
9 x = np.linspace(-6 * np.pi, 6 * np.pi, 1024) # 1024个数据
10 # 调大振幅和频率
11 cos, sin = 3 * np.cos(x), np.sin(4 * x)

```

```

12 ax.set_xticks([i * np.pi for i in range(-6, 7)]) # 设置x轴刻度
13 # cos曲线, 颜色为蓝色, 线宽为2, 连续线
14 plt.plot(x, cos, color='blue', linewidth=2, alpha=1, linestyle='-',
15          label='cos')
16 # sin曲线, 颜色为红色, 线宽为2, 间断线
17 plt.plot(x, sin, color='red', linewidth=2, alpha=1, linestyle='--',
18          label='sin')
19
20 # 画出十字形的坐标轴
21 ax.spines['right'].set_visible(False) # 隐藏右边框
22 ax.spines['top'].set_visible(False) # 隐藏上边框
23 ax.spines['bottom'].set_position(('data', 0)) # 设置下边框到y轴0的位置
24 ax.xaxis.set_ticks_position('bottom') # 刻度值设在下方
25 ax.spines['left'].set_position(('data', 0)) # 设置左边框到x轴0的位置
26 ax.yaxis.set_ticks_position('left') # 刻度值设在左侧
27
28 ax.legend(loc='lower left') # 在左下角显示图例
29 plt.savefig("C:\\Users\\Lenovo\\Desktop\\4.jpg")
30 plt.show()

```



5 要求 5

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。

```
1 import csv
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from matplotlib.ticker import MultipleLocator
5
6 PM_2010 = [] # 列表对象
7 PM_2011 = []
8 PM_2012 = []
9 PM_2013 = []
10 PM_2014 = []
11 PM_2015 = []
12 with open('ave_by_month.csv', 'r') as csvFile:
13     f_csv = csv.reader(csvFile) # 读取文件
14     temp = 0
15     for row in f_csv:
16         if temp == 0:
17             temp = temp + 1
18         elif len(PM_2010) != 12:
19             PM_2010.append(round(float(row[2]), 2))
20         elif len(PM_2011) != 12:
21             PM_2011.append(round(float(row[2]), 2))
22         elif len(PM_2012) != 12:
23             PM_2012.append(round(float(row[2]), 2))
24         elif len(PM_2013) != 12:
25             PM_2013.append(round(float(row[2]), 2))
26         elif len(PM_2014) != 12:
27             PM_2014.append(round(float(row[2]), 2))
28         elif len(PM_2015) != 12:
29             PM_2015.append(round(float(row[2]), 2))
30 ax = plt.axes() # 创建坐标对象
31 plt.style.use('bmh') # 设置图像风格
32 plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
33 plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
34 ax.set_title("北京市10-15年PM指数月平均数据的变化情况") # 直方图标题
```

```

35 ax.set_ylabel('PM2.5指数')
36 ax.set_xlabel('月份')
37 x = np.arange(1, 13) # 生成横轴数据
38 x_major_locator = MultipleLocator(1)
39 ax.xaxis.set_major_locator(x_major_locator)
40 plt.plot(x, PM_2010, color="blue", linewidth=2, linestyle="--", label=
    "2010")
41 plt.plot(x, PM_2011, color="red", linewidth=2, linestyle="--", label="
    2011")
42 plt.plot(x, PM_2012, color="orange", linewidth=2, linestyle="--",
    label="2012")
43 plt.plot(x, PM_2013, color="black", linewidth=2, linestyle="--", label
    ="2013")
44 plt.plot(x, PM_2014, color="green", linewidth=2, linestyle="--", label
    ="2014")
45 plt.plot(x, PM_2015, color="pink", linewidth=2, linestyle="--", label=
    "2015")
46 ax.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0)
47 plt.savefig("C:\\Users\\Lenovo\\Desktop\\5.jpg")
48 plt.show()

```

北京市10-15年PM指数月平均数据的变化情况

