

数据可视化作业

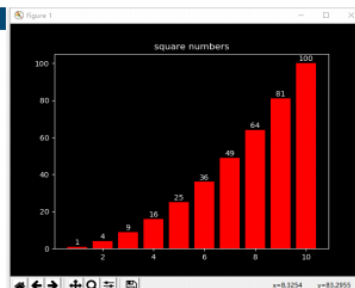
作业一

1. 作业要求

14.1.4 直方图

一、对右图进行修改：

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



2. 解决方案

1. 更换图形的风格

```
plt.style.use('seaborn-bright')
```

更换为 seaborn-bright 样式

2. 将 x 轴的数据改为-10 到 10

```
x = np.arange(-10, 11, 1)
ax.set_xlim(-11, 11)
ax.set_xticks(x)
```

x 轴数据从-10 到 10(包括), 以 1 位跨度

3. 自行构造 y 值函数

```
y = x * x + x
```

$y = x^2 + x$

4. 将直方图上的数字位置改到柱形图的内部垂直居中

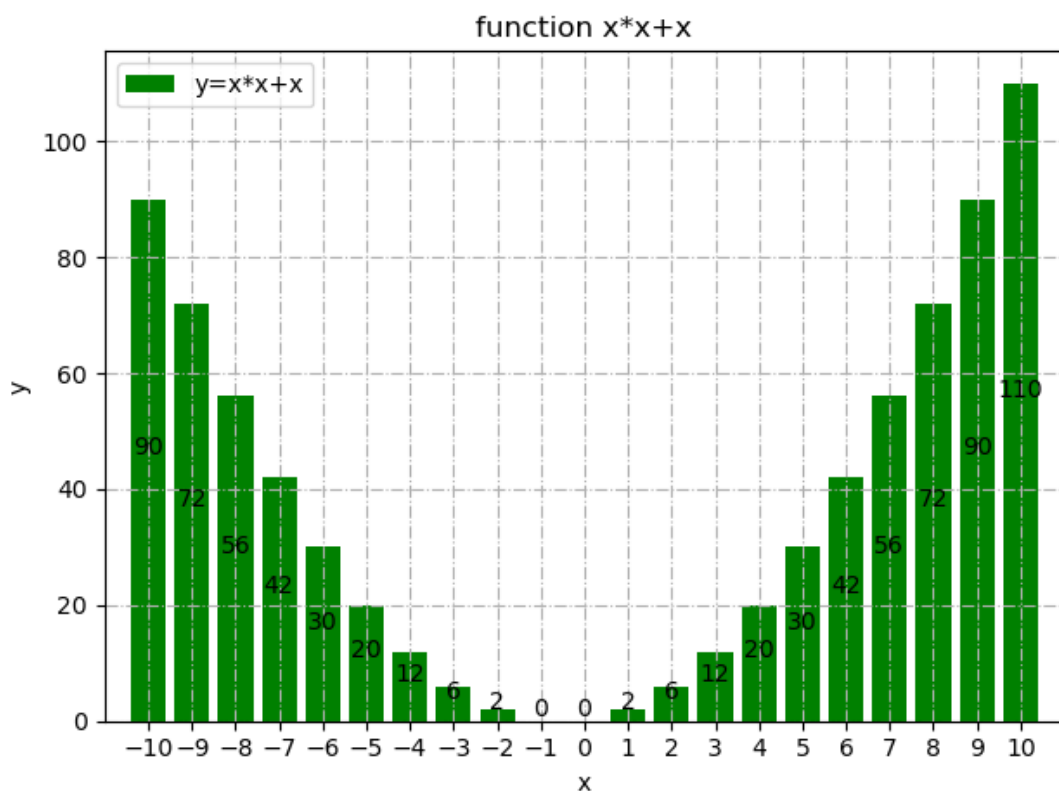
```
for a, b in zip(x, y):
    plt.text(a, b / 2, '%d' % b, ha='center', va='bottom', fontsize=10)
```

修改 b 的值为 1/2

5. 设置图例、标题等

```
plt.bar(x, y, color='g', label='y=x*x+x')
plt.legend(loc='upper left')
plt.xlabel("x")
plt.ylabel("y")
plt.grid(True, linestyle='-.', alpha=1)
```

3. 最终图形



作业二

1. 作业要求

二、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。

2. 解决方案

1. 读入文件，每次读取一行，将每一行数据保存到一个数组 scores 中

```
filename = "data1402.csv"
scores = []
with open(filename, 'r') as csvfile:
    f_csv = csv.reader(csvfile)
    for row in f_csv:
        scores.append(float(row[0]))
```

2. 创建一个字典对象 scorescount，用于记录每个整数分数对应的人数。对 scores 中的每一个元素(小数)，先转换为对应的整数。这个整数 即是 scorescount 中的键值，数值是在原有的基础上加 1

```
def count_elements(scores):
    scorescount = {}
    for i in scores:
        scorescount[int(i)] = scorescount.get(int(i), 0) + 1
```

3. 创建新字典_scorecount，用于记录每个分数段的人数，其键值是各个分数段，数值是每个分数段的人数

```
_scorescount = {}
i = 60
while (i < 95):
    for j in range(5):
        _scorescount[str(i) + "-" + str(i + 4)] = _scorescount.get(str(i) + "-" + str(i + 4),
                                                                    0) + scorescount.get(i + j, 0)
    i += 5
return _scorescount
```

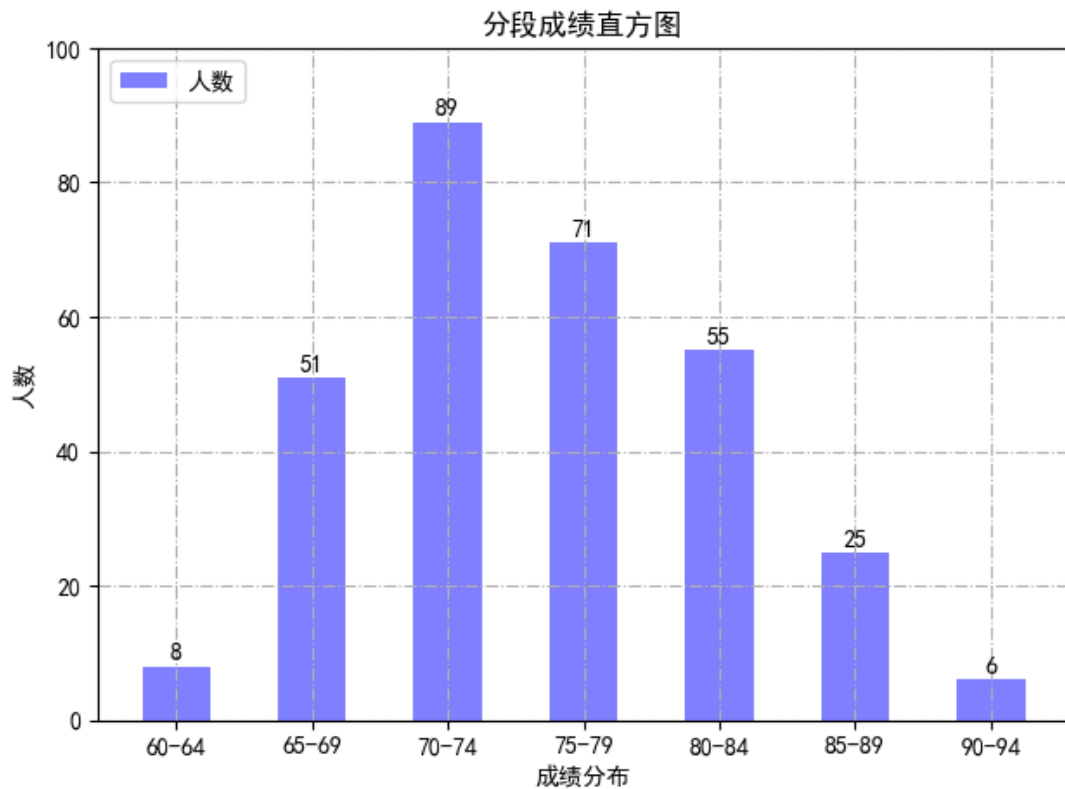
4. 根据字典_scorecount，设置图例、标签等、制图

```
counted = count_elements(scores)
plt.ylim(0, 100)
plt.bar(counted.keys(), counted.values(), 0.5, alpha=0.5, color='b')
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.xlabel("成绩分布")
plt.ylabel("人数")
plt.title("分段成绩直方图")
plt.legend(["人数"], loc='upper left')
plt.grid(True, linestyle='-.', alpha=1)
```

5. 设置直方图上数字显示，在每个柱形图上方

```
for a, b in zip(counted.keys(), counted.values()):
    plt.text(a, b + 0.3, '%d' % b, ha='center', va='bottom', fontsize=10)
plt.show()
```

3. 最终图形



作业三

1. 作业要求

三、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。

2. 解决方案

1. 随机生成一个数字范围在 1-200 的 10×3 的二维数组，每一行数据按顺序分别代表第一学期、第二学期、第三学期的排名

```
# 随机生成10x3数组(范围1-200)
```

```
rank = np.random.randint(1, 201, size=(10, 3))
```

```
print(rank)
```

2. 将每一行的三个数据分别保存到数组 Semester1、Semester2、Semester3 中

```
Semester1 = []
Semester2 = []
Semester3 = []
for row in rank:
    Semester1.append(int(row[0]))
    Semester2.append(int(row[1]))
    Semester3.append(int(row[2]))
```

3. 设置 x 坐标，为 1 到 10, 代表 10 个学生，生成三个学期直方图，从左到右按第一、第二、第三顺序排列，每个直方图宽度为 0.3，依次向右排列

```
plt.bar(x-0.3, Semester1, 0.3, alpha=0.5, color='g')
plt.bar(x, Semester2, 0.3, alpha=0.5, color='b')
plt.bar(x+0.3, Semester3, 0.3, alpha=0.5, color='r')
```

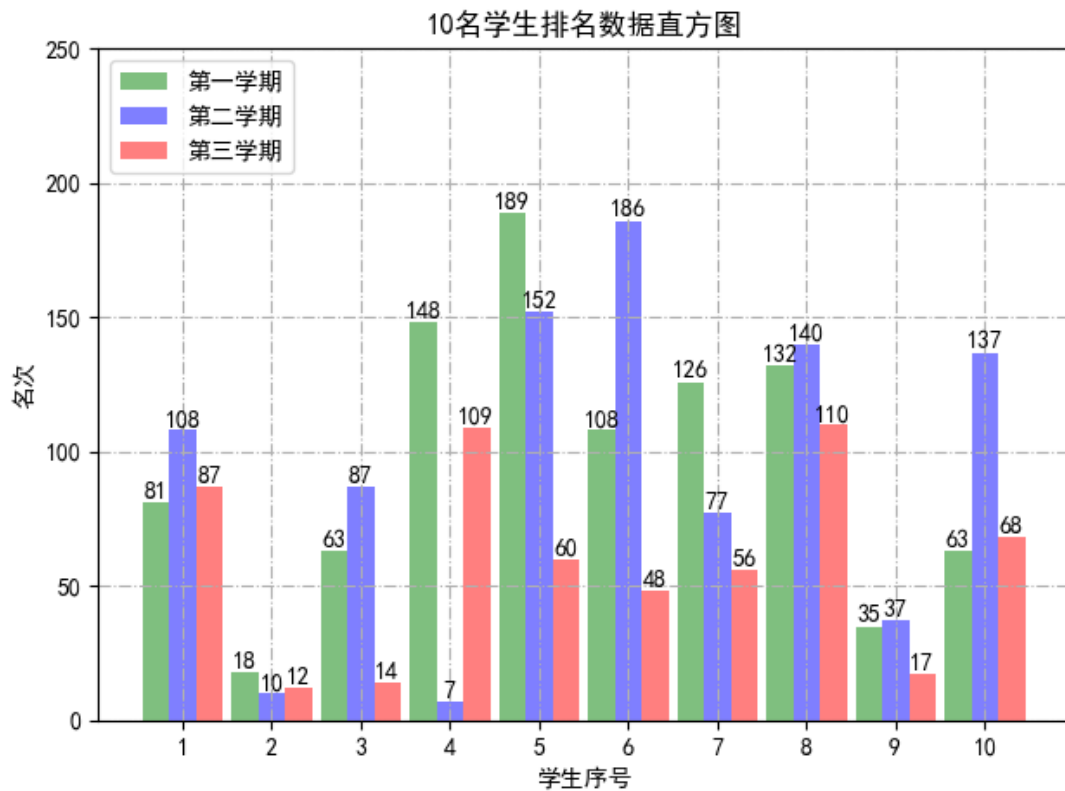
4. 设置直方图上方数字显示

```
for a, b in zip(x, Semester1):
    plt.text(a - 0.3, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
for a, b in zip(x, Semester2):
    plt.text(a, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
for a, b in zip(x, Semester3):
    plt.text(a + 0.3, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
```

5. 设置 x 和 y 坐标范围，用于最直观展示，设置图例、标题等

```
ax.set_xticks(x)
plt.ylim(0, 250)
plt.legend(['第一学期', '第二学期', '第三学期'], loc='upper left')
plt.title('10名学生排名数据直方图')
plt.xlabel('学生序号')
plt.ylabel('名次')
plt.grid(True, linestyle='-.', alpha=1)
plt.show()
```

3. 最终图形



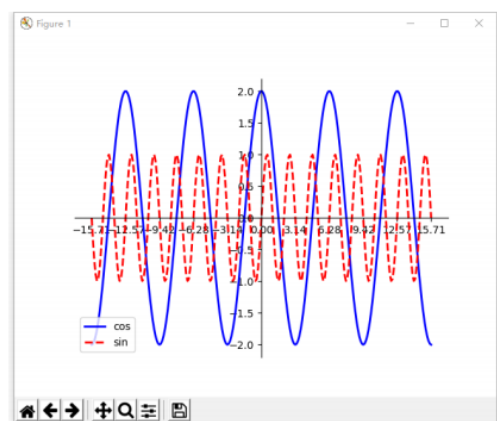
作业四

1. 作业要求

14.1.4 直方图

四、线图

- 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



2. 解决方案

- 将 x 范围调整为 $-5\pi \sim 5\pi$ ，在此范围内生成 640 个数据

```
fig,ax=plt.subplots()
x=np.linspace(-5*np.pi,5*np.pi,640)
```

2. 调到 $\cos(x)$ 的幅度和 $\sin(x)$ 的频率，将 $\cos(x)$ 振幅变为 2 倍， $\sin(x)$ 频率增大 3 倍

```
cos,sin=2*np.cos(x),np.sin(x*3)
```

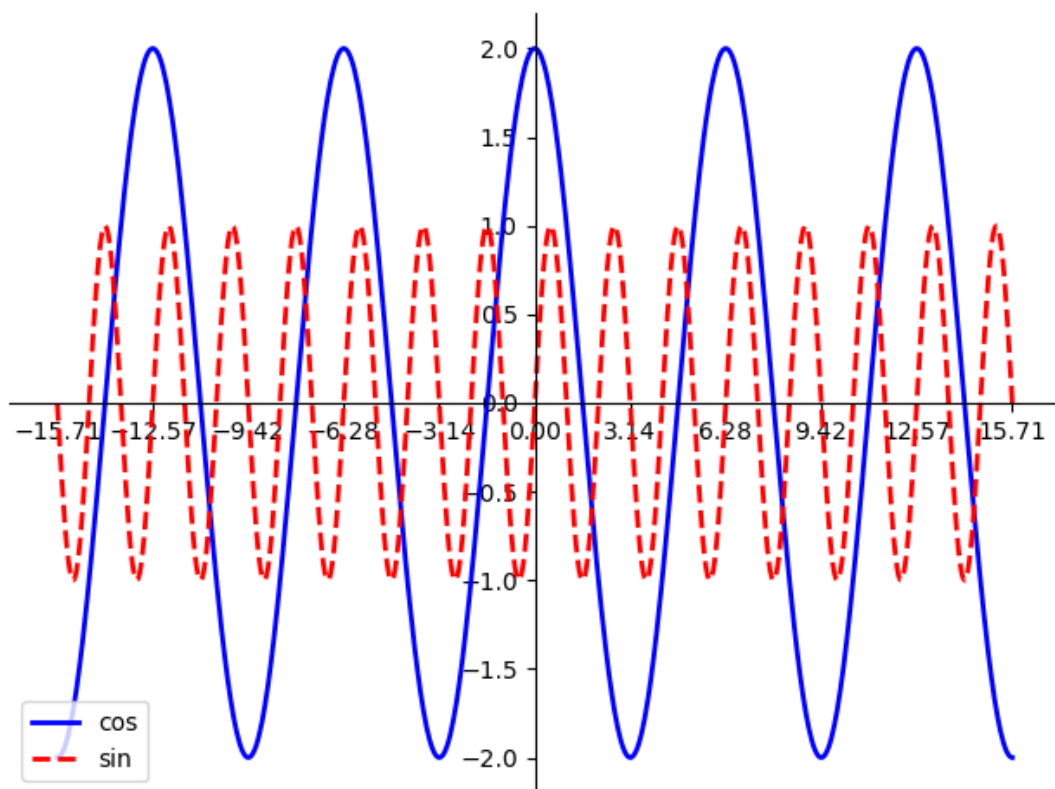
3. 画出十字型的坐标轴

```
ax.spines["right"].set_visible(False)
ax.spines["top"].set_visible(False)
ax.spines['bottom'].set_position(('data', 0))
ax.xaxis.set_ticks_position('bottom')
ax.spines['left'].set_position(('data', 0))
ax.yaxis.set_ticks_position('left')
```

4. 设置标题、图例等，制图

```
ax.set_xticks([i * np.pi for i in range(-5, 6)])
plt.plot(x, cos, color="blue", linewidth=2, linestyle="-", label="cos")
plt.plot(x, sin, color="red", linewidth=2, linestyle="--", label="sin")
plt.legend(loc='lower left')
```

3. 最终图形



作业五

1. 作业要求

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. 解决方案

1. 计算得到 10-15 年 PM 指数月平均数据

```
fileNameStr = 'BeijingPM20100101_20151231.csv'
df = pd.read_csv(fileNameStr, encoding='utf-8')
df.drop(df.columns[[range(10, 18)]], axis=1, inplace=True)
# 去掉三列PM数据全部为空的行
df.dropna(axis=0, how='all', subset=['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US Post'], inplace=True)
df['sum'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US Post']].sum(axis=1)
df['count'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US Post']].count(axis=1)
df['ave'] = round(df['sum'] / df['count'], 2)
bj2 = df.groupby(['year', 'month'])['ave'].mean()#按月份计算平均值
am = pd.DataFrame(bj2).reset_index()
```

2. 将 10-15 年的月平均数据分别存入数组 year_2010、year_2011、year_2012、year_2013、year_2014、year_2015

```
year_2010 = []
year_2011 = []
year_2012 = []
year_2013 = []
year_2014 = []
year_2015 = []
for i in range(0, 12):
    print(am.iloc[i]['ave'])
    year_2010.append(am.iloc[i]['ave'])
    year_2011.append(am.iloc[i + 12]['ave'])
    year_2012.append(am.iloc[i + 24]['ave'])
    year_2013.append(am.iloc[i + 36]['ave'])
    year_2014.append(am.iloc[i + 48]['ave'])
    year_2015.append(am.iloc[i + 60]['ave'])
```

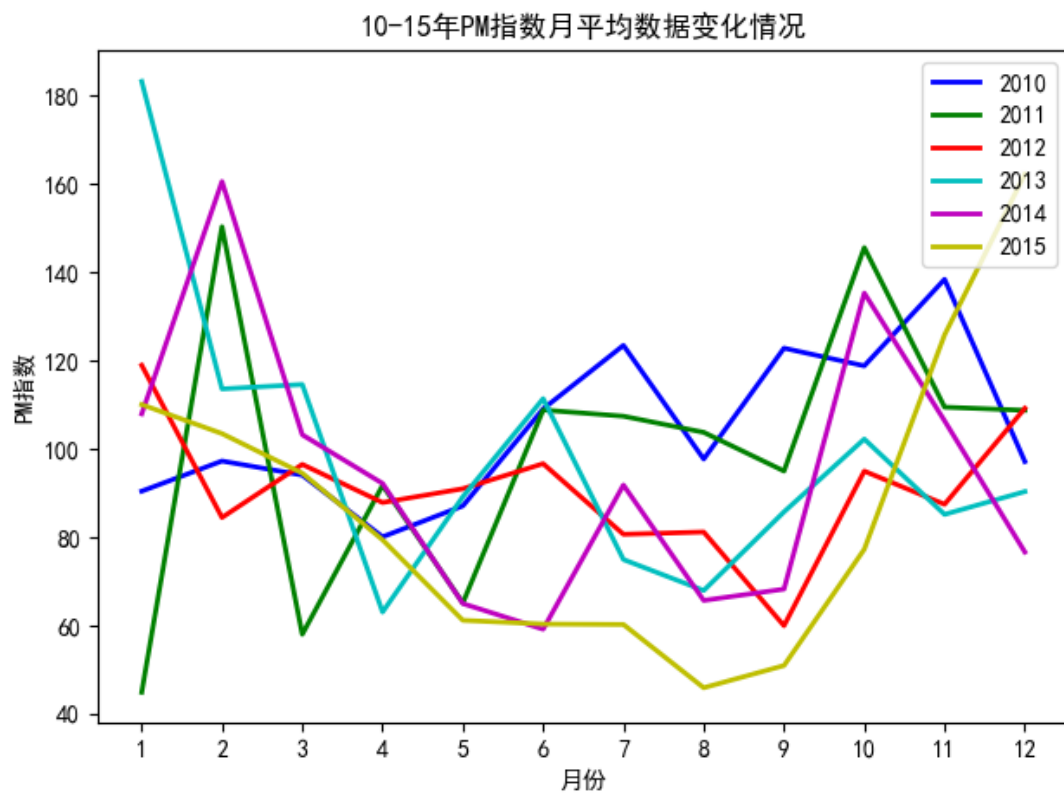
3. 设置中文字体和 x 轴范围，x 轴、y 轴标签

```
plt.rcParams['font.sans-serif'] = ['SimHei']
x = np.arange(1, 13)
plt.xticks(range(1, 13))
plt.title('10-15年PM指数月平均数据变化情况')
plt.xlabel('月份')
plt.ylabel('PM指数')
```


4. 分别对 6 年数据进行绘制折线图，并表以不同的颜色

```
plt.plot(x, year_2010, color="b", linewidth=2, linestyle="-", label="2010")
plt.plot(x, year_2011, color="g", linewidth=2, linestyle="-", label="2011")
plt.plot(x, year_2012, color="r", linewidth=2, linestyle="-", label="2012")
plt.plot(x, year_2013, color="c", linewidth=2, linestyle="-", label="2013")
plt.plot(x, year_2014, color="m", linewidth=2, linestyle="-", label="2014")
plt.plot(x, year_2015, color="y", linewidth=2, linestyle="-", label="2015")
plt.legend(loc='upper right')
plt.show()
```

3. 最终图形



附录: dataVisualization.py

```
1. import matplotlib
2. import matplotlib.pyplot as plt
3. import numpy as np
4. import csv
5. import pandas as pd
6. #作业一
7. def job1():
8.     plt.style.use('seaborn-bright')
9.     fig, ax = plt.subplots()
10.    ax.set_title("function  $x*x+x$ ")
11.    x = np.arange(-10, 11, 1)
12.    ax.set_xlim(-11, 11)
13.    ax.set_xticks(x)
14.    y = x * x + x
15.    plt.bar(x, y, color='g', label='y=x*x+x')
16.    plt.legend(loc='upper left')
17.    plt.xlabel("x")
18.    plt.ylabel("y")
19.    plt.grid(True, linestyle='-.', alpha=1)
20.    for a, b in zip(x, y):
21.        plt.text(a, b / 2, '%d' % b, ha='center', va='bottom', fontsize=10)
22.    plt.show()
23.
24. #作业二
25. def job2():
26.     def count_elements(scores):
27.         scorescount = {}
28.         for i in scores:
29.             scorescount[int(i)] = scorescount.get(int(i), 0) + 1
30.         _scorescount = {}
31.         i = 60
32.         while (i < 95):
33.             for j in range(5):
34.                 _scorescount[str(i) + "-"
35.                    + str(i + 4)] = _scorescount.get(str(i) + "-" + str(i + 4),
36.                    ) + scorescount.get(i + j, 0)
37.                 i += 5
38.             return _scorescount
39.     filename = "data1402.csv"
```

```

40.     scores = []
41.     with open(filename, 'r') as csvfile:
42.         f_csv = csv.reader(csvfile)
43.         for row in f_csv:
44.             scores.append(float(row[0]))
45.
46.     counted = count_elements(scores)
47.     plt.ylim(0, 100)
48.     plt.bar(counted.keys(), counted.values(), 0.5, alpha=0.5, color='b')
49.     plt.rcParams['font.sans-serif'] = ['SimHei']
50.     plt.xlabel("成绩分布")
51.     plt.ylabel("人数")
52.     plt.title("分段成绩直方图")
53.     plt.legend(["人数"], loc='upper left')
54.     plt.grid(True, linestyle='-.', alpha=1)
55.     for a, b in zip(counted.keys(), counted.values()):
56.         plt.text(a, b + 0.3, '%d' % b, ha='center', va='bottom', fontsize=10
57.         )
58.     plt.show()
59. #作业三
60. def job3():
61.     # 随机生成 10x3 数组(范围 1-200)
62.     rank = np.random.randint(1, 201, size=(10, 3))
63.     print(rank)
64.     fig, ax = plt.subplots()
65.     plt.rcParams['font.sans-serif'] = ['SimHei']
66.     Semester1 = []
67.     Semester2 = []
68.     Semester3 = []
69.     for row in rank:
70.         Semester1.append(int(row[0]))
71.         Semester2.append(int(row[1]))
72.         Semester3.append(int(row[2]))
73.     x = np.arange(1, 11)
74.     plt.bar(x - 0.3, Semester1, 0.3, alpha=0.5, color='g')
75.     plt.bar(x, Semester2, 0.3, alpha=0.5, color='b')
76.     plt.bar(x + 0.3, Semester3, 0.3, alpha=0.5, color='r')
77.     for a, b in zip(x, Semester1):
78.         plt.text(a - 0.3, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10)
79.     for a, b in zip(x, Semester2):
80.         plt.text(a, b + 0.2, '%d' % b, ha='center', va='bottom', fontsize=10
81.         )

```

```

81.     for a, b in zip(x, Semester3):
82.         plt.text(a + 0.3, b + 0.2, '%d' % b, ha='center', va='bottom', fonts
            ize=10)
83.     ax.set_xticks(x)
84.     plt.ylim(0, 250)
85.     plt.legend(['第一学期', '第二学期', '第三学期'], loc='upper left')
86.     plt.title('10 名学生排名数据直方图')
87.     plt.xlabel('学生序号')
88.     plt.ylabel('名次')
89.     plt.grid(True, linestyle='-.', alpha=1)
90.     plt.show()
91.
92. def job4():
93.     fig,ax=plt.subplots()
94.     x=np.linspace(-5*np.pi,5*np.pi,640)
95.     cos,sin=2*np.cos(x),np.sin(x*3)
96.     ax.set_xticks([i * np.pi for i in range(-5, 6)])
97.     plt.plot(x, cos, color="blue", linewidth=2, linestyle="--", label="cos")
98.     plt.plot(x, sin, color="red", linewidth=2, linestyle="--", label="sin")
99.     plt.legend(loc='lower left')
100.    ax.spines["right"].set_visible(False)
101.    ax.spines["top"].set_visible(False)
102.    ax.spines['bottom'].set_position(('data', 0))
103.    ax.xaxis.set_ticks_position('bottom')
104.    ax.spines['left'].set_position(('data', 0))
105.    ax.yaxis.set_ticks_position('left')
106.
107.    plt.show()
108.
109. def job5():
110.     fileNameStr = 'BeijingPM20100101_20151231.csv'
111.     df = pd.read_csv(fileNameStr, encoding='utf-8')
112.     df.drop(df.columns[[range(10, 18)]], axis=1, inplace=True)
113.     # 去掉三列 PM 数据全部为空的行
114.     df.dropna(axis=0, how='all', subset=['PM_Dongsi', 'PM_Dongsihuan', 'PM_
        Nongzhanguan', 'PM_US Post'], inplace=True)
115.     df['sum'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_US
        Post']].sum(axis=1)
116.     df['count'] = df[['PM_Dongsi', 'PM_Dongsihuan', 'PM_Nongzhanguan', 'PM_
        US Post']].count(axis=1)
117.     df['ave'] = round(df['sum'] / df['count'], 2)
118.     bj2 = df.groupby(['year', 'month'])['ave'].mean()#按月份计算平均值

```

```
119.     am =pd.DataFrame(bj2).reset_index()
120.     year_2010 = []
121.     year_2011 = []
122.     year_2012 = []
123.     year_2013 = []
124.     year_2014 = []
125.     year_2015 = []
126.     for i in range(0, 12):
127.         print(am.iloc[i]['ave'])
128.         year_2010.append(am.iloc[i]['ave'])
129.         year_2011.append(am.iloc[i + 12]['ave'])
130.         year_2012.append(am.iloc[i + 24]['ave'])
131.         year_2013.append(am.iloc[i + 36]['ave'])
132.         year_2014.append(am.iloc[i + 48]['ave'])
133.         year_2015.append(am.iloc[i + 60]['ave'])
134.
135.     plt.rcParams['font.sans-serif'] = ['SimHei']
136.     x = np.arange(1, 13)
137.     plt.xticks(range(1, 13))
138.     plt.title('10-15 年 PM 指数月平均数据变化情况')
139.     plt.xlabel('月份')
140.     plt.ylabel('PM 指数')
141.     plt.plot(x, year_2010, color="b", linewidth=2, linestyle="-",
142.             label="2010")
143.     plt.plot(x, year_2011, color="g", linewidth=2, linestyle="-",
144.             label="2011")
145.     plt.plot(x, year_2012, color="r", linewidth=2, linestyle="-",
146.             label="2012")
147.     plt.plot(x, year_2013, color="c", linewidth=2, linestyle="-",
148.             label="2013")
149.     plt.plot(x, year_2014, color="m", linewidth=2, linestyle="-",
150.             label="2014")
151.     plt.plot(x, year_2015, color="y", linewidth=2, linestyle="-",
152.             label="2015")
153.     plt.legend(loc='upper right')
154.     plt.show()
155.
156. if __name__ == '__main__':
157.     job1()
158.     job2()
159.     job3()
160.     job4()
161.     job5()
```