用 Python 做统计分析

作者：陈震

组织：西南大学

时间：August 25, 2019

版本：1.00

ElegantLaTeX Program

Victory won't come to us unless we go to it. — M. Moore

目　录

第 1 章 Python 概述与安装

1.1 Python 的发展历史与应用

1.2 Anaconda + Spyder 安装

1.3 Pycharm + Python 安装

第 2 章 Python 的操作基础

2.1 Python 的常用数据类型

2.2 Python 的常用数据结构

2.3 Python 中定义函数

2.4 Python 读取数据

第 3 章 Python 中的 Numpy 与 Pandas

3.1 Numpy 的常用操作

3.2 Pandas 的常用操作

第 4 章 统计数据的图形描述

使用图形描述统计结果是应用统计的基本技能之一。人类在接受信息时，大脑皮层总是优先提取可视化的图形，其次才是具体的文字内容。一些漂亮的图形会使得读者或观众迅速得知我们想要表达的信息。因此不论是写论文，还是汇报展示，做到图文并茂非常重要。

由于 Python 具有良好的可扩展性，它在画图方面具有大量的绘图包，包括模仿 Matlab 绘图风格的 matplotlib, 画地理图形的 geoplotlib，能在浏览器中生成优美图形的 Pyechart 等。本章将展示如何用 Python 的 matplotlib 宏包来画出各种各样的统计图形。

4.1 plot 函数的基本用法

使用 matplotlib 包画图时，我们一般加载里面的 pyplot，并命名为 plt，然后使用 plot 函数画图。

```
# 导入 matplotlib 中的 plot， 并命名为常用名 plt
import matplotlib.pyplot as plt
```

例如，下面的代码画出正弦函数 $y = sin(x)$ 的图形。

```
# 导入宏包
import matplotlib.pyplot as plt
import numpy as np

# 生成数据
x = np.arange(0, 10, 0.1) # 横坐标数据为从0到10之间，步长为0.1的等差数组
y = np.sin(x) # 纵坐标数据为 x 对应的 sin(x) 值

# 生成图形
plt.plot(x, y)

# 显示图形
plt.show()
```

生成的图形如图 4.1 所示。利用 plot 函数，我们可以对图形进行更多精细的设置，官方的详细文档可以参看：https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html。Plot 函数的基本语法是：

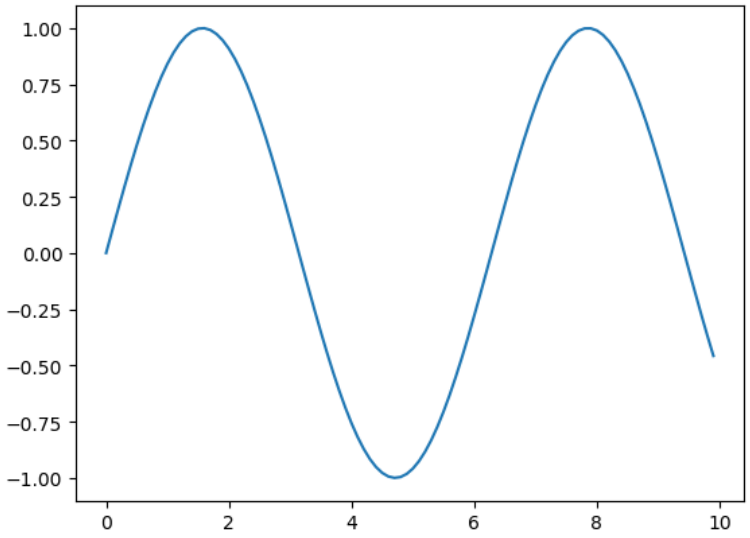

图 4.1: $y = sin(x)$ 的 Python 图像

plot 函数的语法

plot([x], y, [fmt], **kwargs)

[x]	可选参数，横坐标轴数据
y	纵坐标轴数据
[fmt]	可选参数，字符串，定义图形的基本样式：颜色，点形，线形
****kwargs**	不定长的关键字参数，用字典形式设置图形的其他属性

[fmt] 的常用代码（包括颜色代码、点形代码、线形代码），由表 4.1 所示。

表 4.1: Plot 函数中的颜色、点形、线性代码

(a) Plot 函数的颜色代码

颜色代码	颜色
'b'	蓝色
'r'	红色
'g'	绿色
'k'	黑色
'w'	白色
'y'	黄色

(b) Plot 函数的点形代码

线形代码	线形
'o'	实心圆形
'.'	点形
'+'	十字形
'*'	星号
'+'	加号
'x'	叉号

(c) Plot 函数的线形代码

线形代码	线形
'-'	实线
'--'	虚线
'-.'	折线
':'	点线

kwargs 的常用设置包括线条的粗细 linewidth，图像标签 label 等。下面一些 plot 函数的代码展示了 [x]，[fmt]，Kwargs 的一些可选用法。

```
>>> plot(x, y)       # 根据横坐标数据 x 与纵坐标数据 y 画图，采用默认的颜色、点形与线性
>>> plot(y)          # 据纵坐标数据 y 画图，横坐标数据默认为从 0 到 N-1，步长为 1 的等差数组
>>> plot(x, y, 'bo') # 颜色为蓝色('b')、点形为圆('o')
>>> plot(y, 'g-.')   # 颜色为绿色('g')，线形为折线('-.')
>>> plt.plot(x, y, 'yo:', label='y=sin(x)', linewidth=2) # 颜色为黄色('y')，点形为圆形('o')，线形为虚线(':')，lable 内容为 'y=sin(x)'，线条宽度为 2
```

如果我们想自定义坐标轴的标题，坐标轴的刻度，坐标轴刻度的范围，设置图形标题，添加图例时，可以通过设置 pyplot 函数中的 xlable（横坐标轴标题）, ylabel（纵坐标轴标题）, xticks（横坐标轴刻度），yticks（纵坐标轴刻度），title（图形标题）, grid（显示网格），legend（显示图例）等属性来实现。经过自定义设置，对图 4.1 的代码进行一下修改，生成图 4.2。

```
# 导入宏包
import matplotlib.pyplot as plt
import numpy as np

# 这两行代码使得 pyplot 画出的图形中可以显示中文
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 生成数据
x = np.arange(0, 10, 0.5)
y = np.sin(x)

# 生成图形
plt.plot(x, y, 'go:', label='y=sin(x)', linewidth=2) # 颜色绿色，点形圆形，线性虚线，设置图例显示内容，线条宽度为2

plt.ylabel('y') # 横坐标轴的标题
plt.xlabel('x') # 纵坐标轴的标题
plt.xticks(np.arange(0, 11, 1)) # 设置横坐标轴的刻度为 0 到 10 的数组
plt.ylim([-2, 2]) # 设置纵坐标轴范围为 -2 到 2
plt.legend() # 显示图例，图例中内容由 label 定义
plt.grid() # 显示网格
plt.title('我的第一个 Python 图形') # 图形的标题

# 显示图形
plt.show()
```

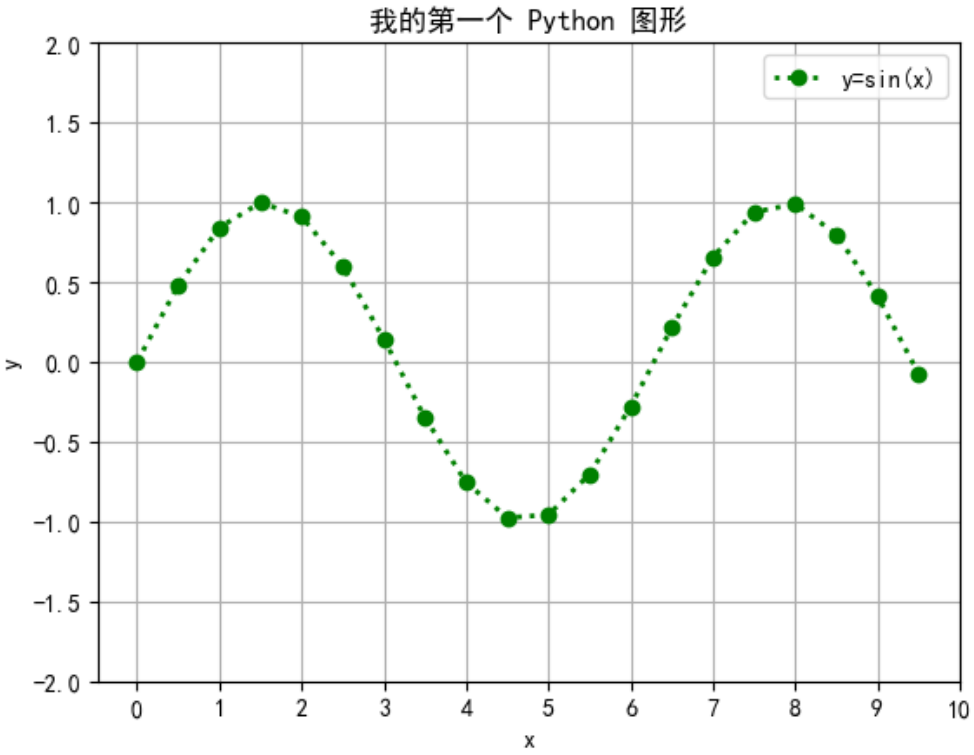

图 4.2: 自定义坐标轴标题、刻度、范围，显示图例、网格与图形标题

4.2 线图

在统计学中，线图一般用来表示时间序列数据，线图称为趋势图（run chart），因为从线图中可以清晰地看出数据虽时间变化的趋势。表 4.2 是我国近 10 年的 GDP 增长率，以及三大产业在近 10 年的增长率。

表 4.2: 我国近 10 年的经济增长率，数据来源：国家统计局

时间	GDP 增长率	第一产业增长率	第二产业增长率	第三产业增长率
2009 年	9.40	4.00	10.30	9.60
2010 年	10.60	4.30	12.70	9.70
2011 年	9.60	4.20	10.70	9.50
2012 年	7.90	4.50	8.40	8.00
2013 年	7.80	3.80	8.00	8.30
2014 年	7.30	4.10	7.40	7.80
2015 年	6.90	3.90	6.20	8.20
2016 年	6.70	3.30	6.30	7.70
2017 年	6.80	4.00	5.90	7.90
2018 年	6.60	3.50	5.80	7.60

在画图时，横坐标轴数据为年份，纵坐标轴数据分别为 GDP 增长率，第一产业增长率，第二产业增长率，第三产业增长率。为了将四个纵坐标轴数据显示在一个图形上，可

以用四个 plot 函数进行划线。Python 画图的代码为：

```
import matplotlib.pyplot as plt

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 输入纵坐标轴数据与横坐标轴数据
gdp_rate = [9.4, 10.6, 9.6, 7.9, 7.8, 7.3, 6.9, 6.7, 6.8, 6.6]
first_industry_rate = [4.0, 4.3, 4.2, 4.50, 3.8, 4.1, 3.9, 3.3, 4.0, 3.5]
second_industry_rate = [10.3, 12.7, 10.7, 8.4, 8.0, 7.4, 6.2, 6.3, 5.9, 5.8]
third_industry_rate = [9.6, 9.7, 9.5, 8.0, 8.3, 7.8, 8.2, 7.7, 7.9, 7.6]
years = [2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018]

# 4 个 plot 函数画出 4 条线，线形为折线，每条线对应各自的标签 label
plt.plot(years, gdp_rate, '.-', label='GDP增长率')
plt.plot(years, first_industry_rate, '.-', label='第一产业增长率')
plt.plot(years, second_industry_rate, '.-', label='第二产业增长率')
plt.plot(years, third_industry_rate, '.-', label='第三产业增长率')

plt.xticks(years) # 设置横坐标刻度为给定的年份
plt.xlabel('年份') # 设置横坐标轴标题
plt.legend() # 显示图例，即每条线对应 label 中的内容
plt.show() # 显示图形
```

显示效果如图 4.3 所示。

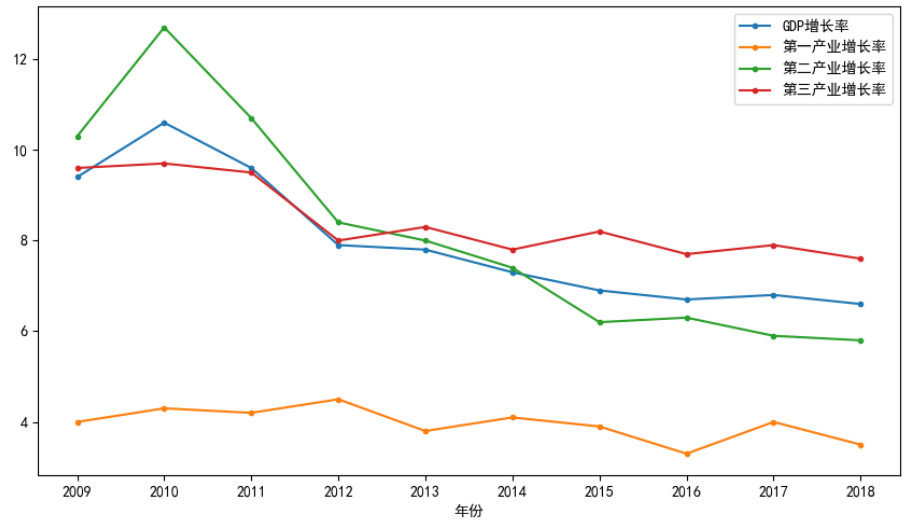

图 4.3: 近十年我国的 GDP 以及各产业增长率

4.3 散点图

散点图是数据点在直角坐标系平面上的分布图，在统计学的回归分析与预测中经常用到。用横轴代表变量 x，纵轴代表变量 y，每组数据 (x_i, y_i) 在坐标系中用一个点表示。做散点图要用到 pyplot 中的 scatter 函数，该函数的基本语法为[1]：

scatter 函数的语法

scatter(x, y, [s], [c], **kwargs)

x	横坐标轴数据
y	纵坐标轴数据
[s]	可选参数，一个数或一个数组，设置每个散点的大小
[c]	可选参数，一个数或一个数组，设置每个散点的颜色
****kwargs**	不定长的关键字参数，用字典形式设置图形的其他属性

**kwargs 中常设置的是不透明度属性 alpha，其大小为 0 到 1 之间的浮点数。假设某个农产品的产量与温度和降雨量的关系如表 4.3 所示。

表 4.3: 某农产品的产量与温度和降雨量的关系

产量	温度	降雨量
1125	6	25
1725	8	40
2250	10	58
2875	13	68
2900	14	110
3750	16	98
4125	21	120

作出产量与温度的散点图的 Python 代码与图形如下：

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

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 输入产量与温度数据
production = [1125, 1725, 2250, 2875, 2900, 3750, 4125]
tem = [6, 8, 10, 13, 14, 16, 21]
```

[1]在 matplotlib 宏包中，0 到 1 之间的浮点数产生相应的 RGB 颜色

```
colors = np.random.rand(len(tem)) # 颜色数组
plt.scatter(tem, production, s=200, c=colors) # 画散点图，大小为 200
plt.xlabel('温度') # 横坐标轴标题
plt.ylabel('产量') # 纵坐标轴标题
plt.show()
```

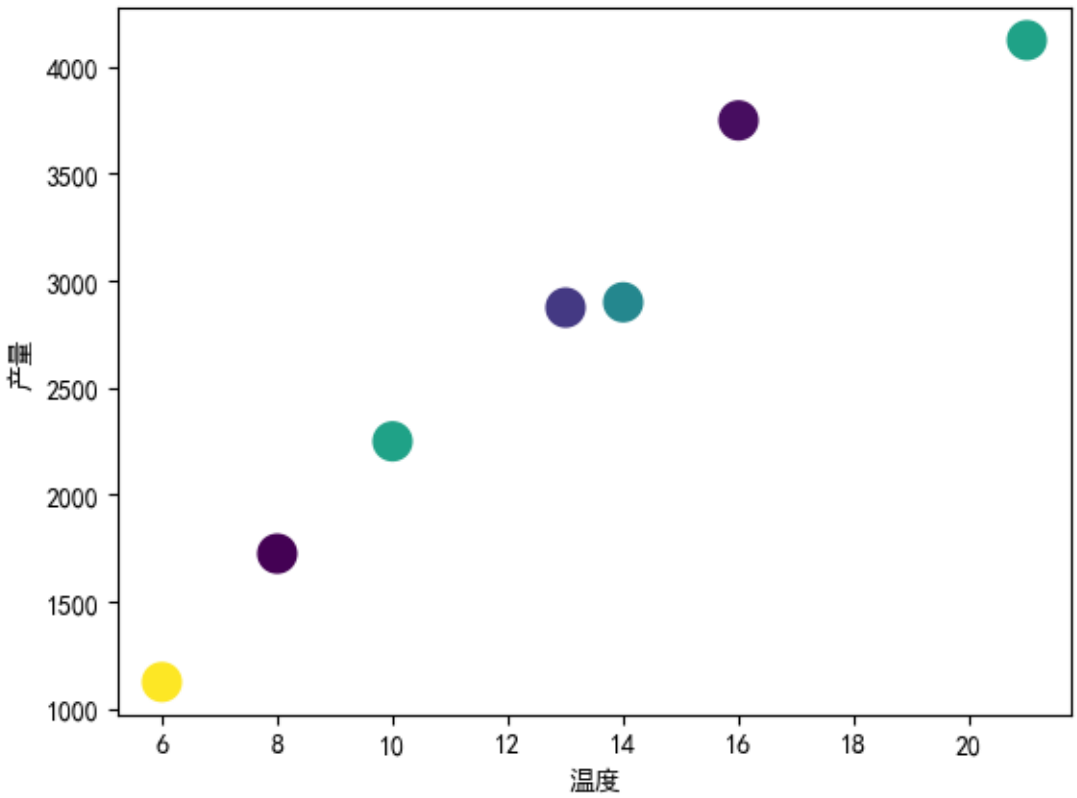

图 4.4: 产量与温度的散点图

若将散点大小的数据换为第三个变量的数值，则可以作出反映三个变量关系的气泡图。下面的代码和图形做出了一个气泡图。图 4.5 反映了产量与温度、降雨量的关系：温度数值在横坐标轴，降雨量数值在纵坐标轴，降雨量的大小用气泡的大小表示。

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

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 输入产量与温度数据
production = [1125, 1725, 2250, 2875, 2900, 3750, 4125]
```

```
tem = [6, 8, 10, 13, 14, 16, 21]
rain = [25, 40, 58, 68, 110, 98, 120]

colors = np.random.rand(len(tem)) # 颜色数组
size = production
plt.scatter(tem, rain, s=size, c=colors, alpha=0.6) # 画散点图，alpha=0.6 表示不
    透明度为 0.6
plt.ylim([0, 150]) # 纵坐标轴范围
plt.xlim([0, 30]) # 横坐标轴范围
plt.xlabel('温度') # 横坐标轴标题
plt.ylabel('降雨量') # 纵坐标轴标题
plt.show()
```

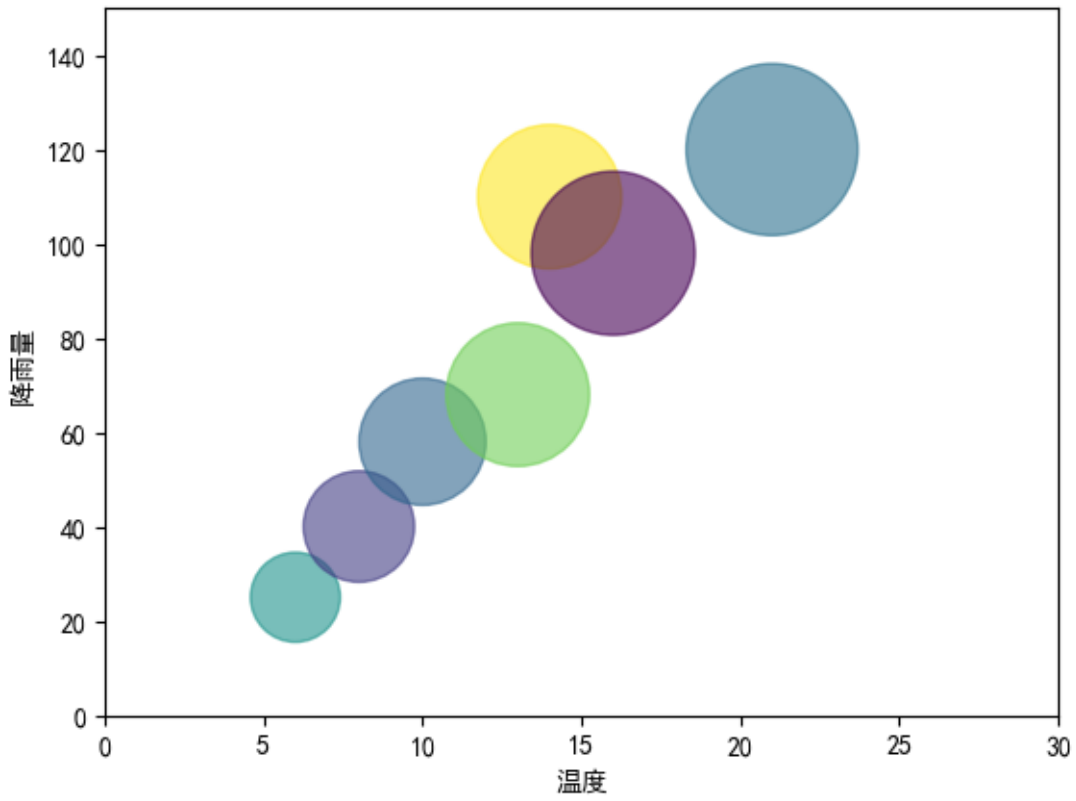

图 4.5: 产量与温度、降雨量关系的气泡图

4.4 条形图

条形图（bar chart），也称为柱状图，是一种以长方形的长度为变量的统计图表，长方形的长度与它所对应的变量数值呈一定比例。画条形图要用到 pyplot 中的 bar 函数，该函数的基本语法为：

bar 函数的语法

bar(x, height, [width], **kwargs)

x	数组，每个条形的横坐标
height	一个数或一个数组，条形的高度
[width]	可选参数，一个数或一个数组，条形的宽度，默认为 0.8
****kwargs**	不定长的关键字参数，用字典形式设置条形图的其他属性

**kwargs 中常设置的参数包括图形标签 label，颜色标签 color，不透明度 alpha 等。假设某项针对男女大学生购买饮用水爱好的调查结果如下表：

表 4.4: 男女大学生购买饮用水爱好调查表

买水选择	男	女
碳酸饮料	6	9
绿茶	7	4
矿泉水	6	4
果汁	1	5
其他	2	6
总计	22	28

画出男生饮用水情况的直方图，代码如下：

```
import matplotlib.pyplot as plt

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

waters = ('碳酸饮料', '绿茶', '矿泉水', '果汁', '其他')
buy_number = [6, 7, 6, 1, 2]

plt.bar(waters, buy_number)
plt.title('男性购买饮用水情况的调查结果')

plt.show()
```

显示的图形：

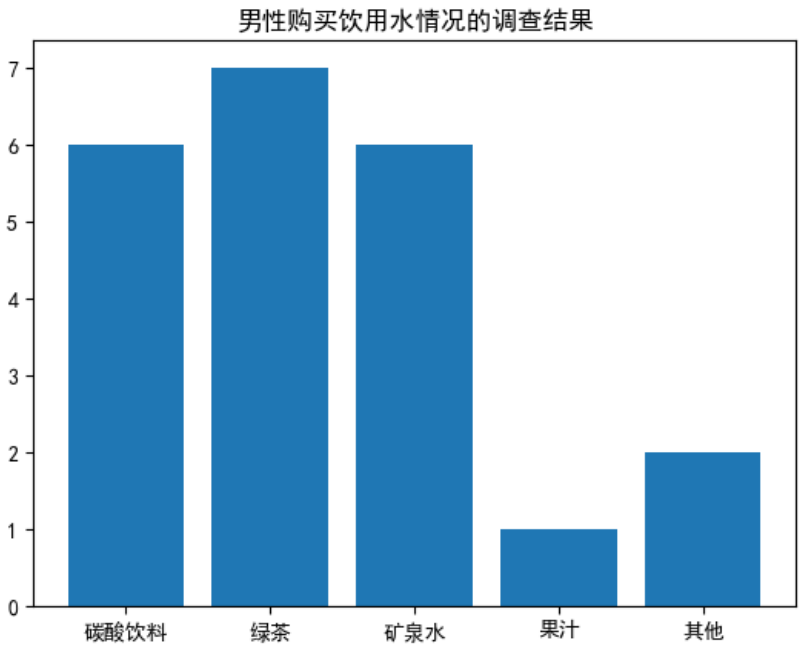

图 4.6: 男性购买饮用水条形图

若要生成横的条形图，则可以使用 barh 函数，其语法与 bar 函数非常类似。

barh 函数的语法

bar(x, width, [height], **kwargs)

y	数组，每个条形的纵坐标
width	一个数或一个数组，条形的宽度
[height]	可选参数，一个数或一个数组，条形的高度，默认为 0.8
kwargs	不定长的关键字参数，用字典形式设置条形图的其他属性

对应的代码与图形为：

```
import matplotlib.pyplot as plt

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

waters = ('碳酸饮料', '绿茶', '矿泉水', '果汁', '其他')
buy_number = [6, 7, 6, 1, 2]

plt.barh(waters, buy_number) # 横放条形图函数 barh
```

```
plt.title('男性购买饮用水情况的调查结果')

plt.show()
```

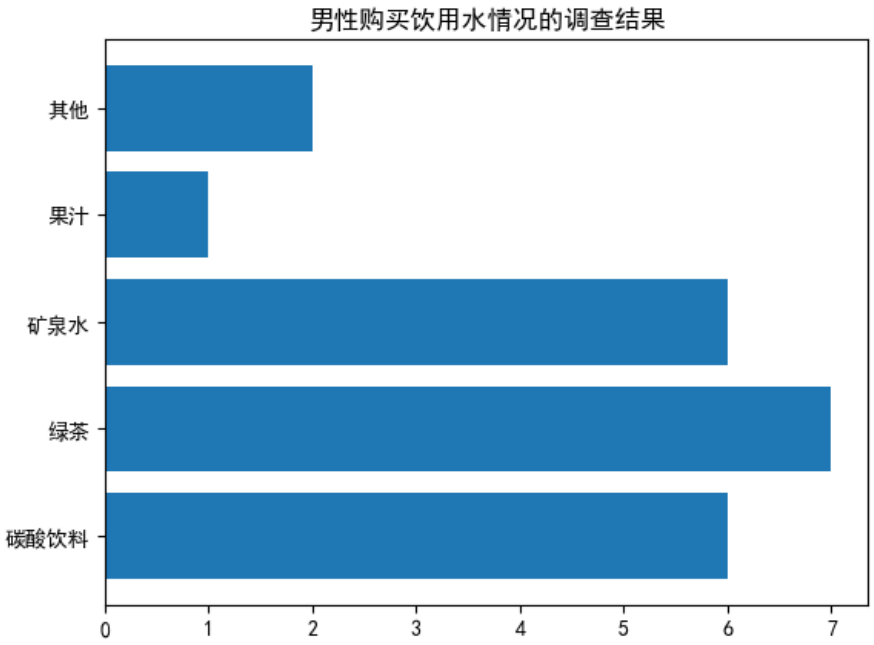

图 4.7: 男性购买饮用水条形图（条形横放）

若要将男生与女生的调查情况画出两个条形图一块显示，则可以使用 bar 或 barh 函数两次，并调整 bar 或 barh 函数的条形图位置坐标以及相应刻度，使得两组条形图能够并排显示。

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

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 输入统计数据
waters = ('碳酸饮料', '绿茶', '矿泉水', '果汁', '其他')
buy_number_male = [6, 7, 6, 1, 2]
buy_number_female = [9, 4, 4, 5, 6]

bar_width = 0.3 # 条形宽度
index_male = np.arange(len(waters)) # 男生条形图的横坐标
index_female = index_male + bar_width # 女生条形图的横坐标
```

```
# 使用两次 bar 函数画出两组条形图
plt.bar(index_male, height=buy_number_male, width=bar_width, color='b', label='
    男性')
plt.bar(index_female, height=buy_number_female, width=bar_width, color='g',
    label='女性')

plt.legend() # 显示图例
plt.xticks(index_male + bar_width/2, waters) # 让横坐标轴刻度显示 waters 里的饮
    用水， index_male + bar_width/2 为横坐标轴刻度的位置
plt.ylabel('购买量') # 纵坐标轴标题
plt.title('购买饮用水情况的调查结果') # 图形标题

plt.show()
```

生成图 4.8。

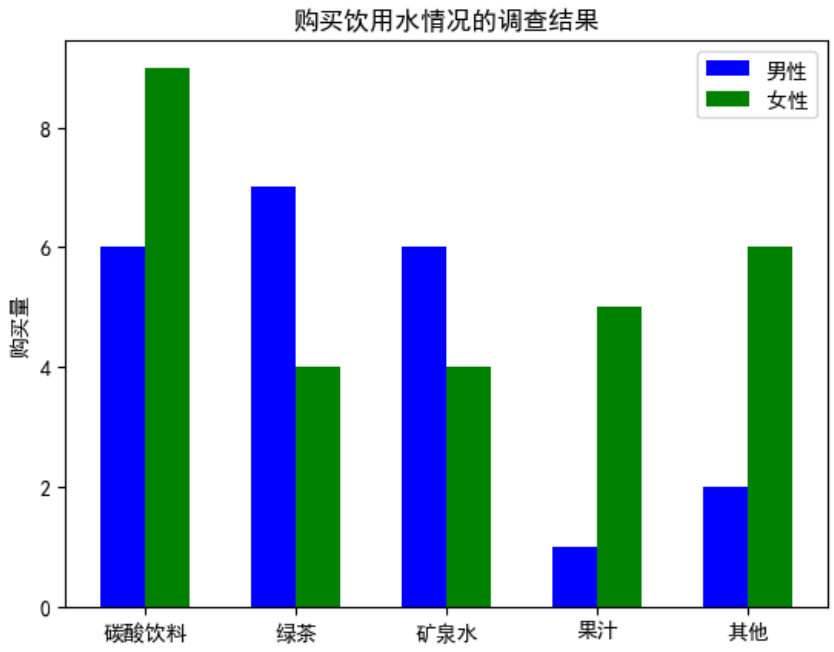

图 4.8: 男女购买饮用水条形图

4.5 直方图

直方图（histogram）虽然在样式上类似条形图，但它们的作用不一样。直方图用不同的矩形表示频数，常用来观察一组数据的概率分布。在直角坐标中，用横轴表示数据分组，纵轴表示频数或频率，各组与相应的频数就形成了一个个矩形，即直方图。

画直方图用到 pyplot 中的 hist 函数，它的基本语法为：

hist 函数的语法

[n, bins, patches] = hist(x, [bins], **kwargs)

输入值:

x	数组，需要绘制直方图的数值
[bins]	可选参数，数据的组数，若不指定则 hist 函数默认计算一个组数
****kwargs**	不定长的关键字参数，用字典形式设置条形图的其他属性

返回值:

n	一个数组，每组直方图的频数（或概率密度）
bins	一个数组，直方图的边界数值
patches	一个对象组，每个对象代表直方图的矩形

**kwargs 中常用来设置的属性包括直方的边界线颜色 edgecolor，不透明度 alpha 等。需要注意的是，hist 函数的三个返回值一般用来继续对数据进行分析，并不是必须要写或必须要使用的。下面举例说明该函数，例如某个饭店每天接待的顾客数有以下记录值：

表 4.5: 某饭店每天接待的顾客数的记录值

141	159	166	172	177	182	188	196	203	214
143	160	167	173	177	183	189	196	203	215
144	160	168	173	178	184	189	196	205	218
149	161	168	174	178	185	189	196	206	223
150	161	168	174	178	186	190	196	207	225
152	162	170	174	179	186	190	197	208	226
153	163	171	175	179	187	191	197	209	228
153	163	171	175	179	187	192	198	210	233
154	164	172	175	180	187	194	198	210	233
155	165	172	175	180	187	194	200	211	234
156	165	172	176	181	188	195	201	211	234
158	165	172	176	182	188	195	202	213	237

在 Phthon 中编写代码，并画出对应的直方图。

```
import matplotlib.pyplot as plt

x = [141, 159, 166, 172, 177, 182, 188, 196, 203, 214,
     143, 160, 167, 173, 177, 183, 189, 196, 203, 215,
     144, 160, 168, 173, 178, 184, 189, 196, 205, 218,
     149, 161, 168, 174, 178, 185, 189, 196, 206, 223,
     150, 161, 168, 174, 178, 186, 190, 196, 207, 225,
     152, 162, 170, 174, 179, 186, 190, 197, 208, 226,
     153, 163, 171, 175, 179, 187, 191, 197, 209, 228,
     153, 163, 171, 175, 179, 187, 192, 198, 210, 233,
     154, 164, 172, 175, 180, 187, 194, 198, 210, 233,
     155, 165, 172, 175, 180, 187, 194, 200, 211, 234,
     156, 165, 172, 176, 181, 188, 195, 201, 211, 234,
     158, 165, 172, 176, 182, 188, 195, 202, 213, 237]

plt.hist(x, edgecolor='k', alpha=0.35) # 设置直方边线颜色为黑色，不透明度为 0.35
plt.show()
```

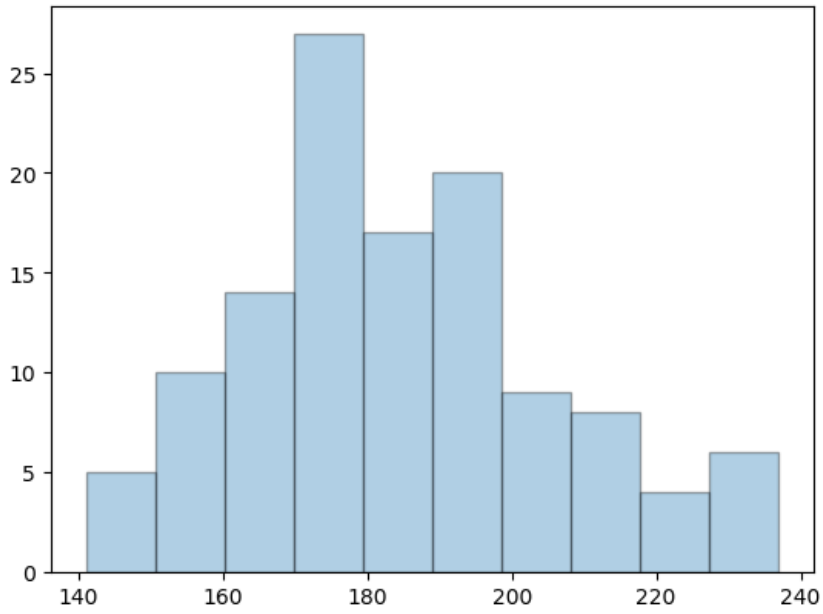

图 4.9: 每天接待顾客数分布的直方图

从图 4.9 可以看出，每天的顾客数大致符合一个正态分布。

4.6 饼图

饼图（pie char）是一个划分为几个扇形的圆形统计图表，一般用于描述频率或百分比之间的相对关系。在饼图中，每个扇区的弧长（以及圆心角和面积）的大小与其所表示的数量呈固定比例。

画饼图一般使用 pyplot 中的 pie 函数，它的基本语法如下：

pie 函数的语法

bar(x, [expode], [labels], [autopic], **kwargs)

x	数组，每个扇区的比例
[expode]	可选参数，数组，每个扇区突出的大小
[labels]	可选参数，字符串数组，每个扇区的标签
[autopct]	可选参数，字符串或函数，每个扇区显示的数字样式
****kwargs**	不定长的关键字参数，用字典形式设置条形图的其他属性

**kwargs 中常用来设置的属性包括是否显示扇区的阴影 shadow，扇区的起始角度 startangle 等。假设有下列调查数据：

表 4.6: 购水类型的调查数据

购水类型	人数
碳酸饮料	15
绿茶	11
矿泉水	10
其他	8
果汁	6

饼图的代码与图形：

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

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

labels = ['果汁', '矿泉水', '绿茶', '其他', '碳酸饮料']
x = [6, 10, 11, 8, 15]
explode = [0, 0.1, 0, 0, 0] # 突出显示第二个扇区

plt.pie(x, explode=explode, labels=labels, autopct='%.2f%%', shadow=True,
    startangle=90)
```

```
plt.legend() # 显示标签
plt.axis('equal') # 让图形和坐标轴相等，这样避免饼图为椭圆形
plt.show()
```

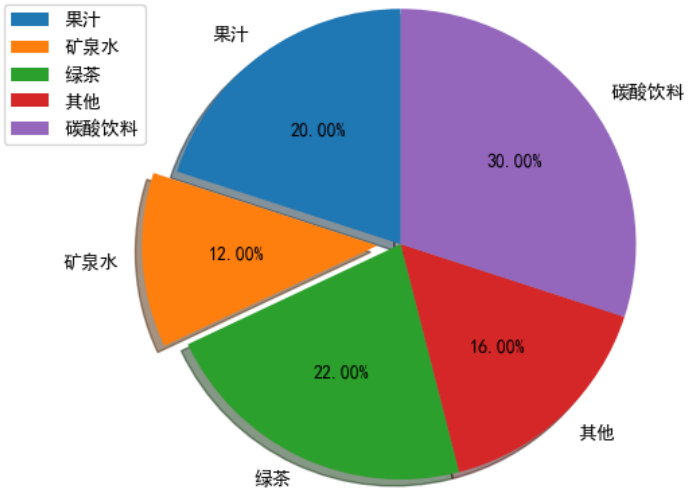

图 4.10: 调查结果的饼图

若在饼图上既要显示比率，又要显示实际调查数字，则可以通过自定义 autopct 函数来实现：

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

# 这两行代码解决 plt 中文显示的问题
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False


# 自定义 autopct 函数
def func(pct, allvals):
    absolute = int(round(pct*np.sum(allvals)/100.0))
    return "{:.1f}%\n({:d})".format(pct, absolute)


labels = ['果汁', '矿泉水', '绿茶', '其他', '碳酸饮料']
x = [6, 10, 11, 8, 15]
explode = [0, 0.1, 0, 0, 0] # 突出显示第二个扇区

plt.pie(x, explode=explode, labels=labels, autopct=lambda pct: func(pct, x), #
```

```
    利用 lambda 定义 pct 函数
        shadow=True, startangle=90)
plt.legend() # 显示标签
plt.axis('equal') # 让图形和坐标轴相等，这样饼图会更好看
plt.show()
```

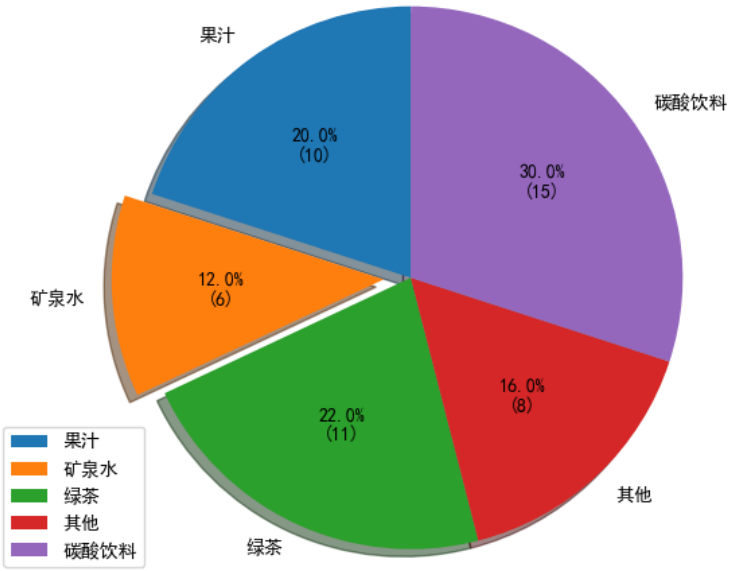

图 4.11: 调查结果的饼图（同事显示比例与原始数据）

4.7 箱线图

箱线图（box plot）在学术论文中经常出现，是一种用来显示一组数据分散情况资料的统计图，由数据的最大值、最小值、中位数、两个四分位数，共五个特征绘制而成。

表 4.7: 学生成绩统计表

	学生编号										
课程名称	1	2	3	4	5	6	7	8	9	10	11
英语	76	90	97	71	70	93	86	83	78	85	81
西方经济学	93	81	76	88	66	79	83	92	78	86	78
市场营销学	74	87	85	69	90	80	77	84	91	74	70
财务管理	68	75	70	84	73	60	76	81	88	68	75
统计学	55	91	68	73	84	81	70	69	94	62	71

第 5 章 概率、概率分布与抽样分布

第 6 章 参数估计与假设检验

第 7 章　方差分析

第 8 章 线性回归

第 9 章 非线性回归

第 10 章 主成分分析

第 11 章 因子分析

第 12 章 时间序列分析和预测