第5章 数值数据智能分析技术

【实验目的】

(1) 掌握 NumPy 库的基本功能。

(2) 掌握一维数组和二维数组的创建。

(3) 掌握数据的输入、编辑和修改操作。

(4) 掌握外部数据的导入导出操作。

(5) 掌握数据的引用、公式和函数的使用方法。

(6) 掌握数据的排序、筛选与分类汇总操作。

(7) 初步利用 NumPy 和 Pandas 进行数据存储与预处理。

(8) 掌握数据可视化的使用方法,利用数据图表对数据进行分析。

【实验环境】

(1) Python 3.13.1 及更高版本。

(2) NumPy 库安装包。

(3) Pandas 库安装包。

(4) Matplotlib 库安装包。

【实验内容】

(1) 了解 NumPy 库的基本功能。

(2) 掌握一维数组和二维数组的创建。

(3) 掌握 NumPy 库对数组的操作与运算。

(4) 在 Pandas 中导入导出外部数据。

(5) Pandas 库的基本操作。

(6) 使用排序、筛选和分类汇总。

(7) 了解 Matplotlib 库的基本功能。

(8) 掌握 Matplotlib 库的使用方法。

(9) 利用数据图表对数据进行分析。

实验 5.1 NumPy 库的基础操作

【实验要求】

(1) 安装第三方 NumPy 库。

(2) 掌握 NumPy 库的基本功能。

(3) 掌握一维数组和二维数组的创建。

(4) 掌握 NumPy 库对数组的操作与运算。

【实验步骤】

(1) 创建 0～9 的一维数字数组。

导入 NumPy 库,设置别名为 np。创建从 0～9 的一维数组,代码如下:

```
>>> import numpy as np
>>> arr = np.arange(10)
>>> print(arr)
[0 1 2 3 4 5 6 7 8 9]
```

(2) 显示 NumPy 的版本号。

```
>>> print(np.__version__)
1.18.5
```

(3) 创建一个元素全为 True 的 3×3 数组。

创建一个布尔类型的数组,代码如下:

```
>>> import numpy as np
arr = np.full([3, 3], True, dtype = np.bool)
print(arr)
[[ True  True  True]
[ True  True  True]
[ True  True  True]]
```

(4) 从数组 arr 中提取所有奇数。

从一维数组中提取满足指定条件的元素,有两种实现方法。

方法一:利用索引值访问数据元素。

```
import numpy as np
arr = np.arange(10)
x = arr[arr % 2 == 1]
print(x)
```

方法二:利用下标值访问数据元素。

```
import numpy as np
arr = np.arange(10)
```

```
x = arr[arr % 2 == 1]
print(x)
```

输出结果如下：

```
[1 3 5 7 9]
```

(5) 将数组 arr 中的偶数元素替换为 0。

将数组 arr 中的偶数元素替换为 0，代码如下：

```
import numpy as np
arr = np.arange(10)
index = np.where(arr % 2 == 0)
arr[index] = 0
print(arr)
```

输出结果如下：

```
[0 1 0 3 0 5 0 7 0 9]
```

(6) 将数组 arr 中的所有偶数元素替换为 0，而不改变数组 arr。

在不影响原始数组的情况下替换满足条件的元素项，代码如下：

```
import numpy as np
arr = np.arange(10)
x = np.where(arr % 2 == 0, 0, arr)
print(x)
print(arr)
```

输出结果如下：

```
[0 1 0 3 0 5 0 7 0 9]
[0 1 2 3 4 5 6 7 8 9]
```

(7) 将数组 arr 转换为 2 行的二维数组。

将一维数组转换成二维数组，有两种实现方法。

方法一：利用从左向右的索引值访问数据元素。

```
>>> import numpy as np
>>> arr = np.arange(10)
>>> x = np.reshape(arr, newshape = [2, 5])
>>> print(x)
[[0 1 2 3 4]
[5 6 7 8 9]]
```

方法二：利用从右向左的索引值访问数据元素。

```
>>> import numpy as np
>>> arr = np.arange(10)
>>> x = np.reshape(arr, newshape = [2, -1])
>>> print(x)
[[0 1 2 3 4]
[5 6 7 8 9]]
```

(8) 反转二维数组 arr 的行。

```
>>> import numpy as np
>>> arr = np.arange(9).reshape(3, 3)
>>> print(arr)
[[0 1 2]
[3 4 5]
[6 7 8]]

>>> x = arr[:, ::-1]
>>> print(x)
[[2 1 0]
[5 4 3]
[8 7 6]]
```

(9) 创建一个 5×3 的二维数组，其中包含 5～10 的随机数。

```
>>> import numpy as np
>>> x = np.random.randint(5, 10, [5, 3])
>>> print(x)
[[5 8 8]
[5 6 8]
[8 8 7]
[6 7 9]
[6 5 8]]

>>> x = np.random.uniform(5, 10, [5, 3])
>>> print(x)
[[6.73675226 8.50271284 9.66526032]
[9.42365472 7.56513263 7.86171898]
[9.31718935 5.71579324 9.92067933]
[8.90907128 8.05704153 6.0189007 ]
[8.70753644 7.75056151 5.71714203]]
```

(10) 只打印或显示 NumPy 数组 rand_arr 的小数点后 3 位数。

```
>>> import numpy as np
>>> rand_arr = np.random.random([5, 3])
>>> print(rand_arr)
[[0.33033427 0.05538836 0.05947305]
```

```
[0.36199439 0.48844555 0.26309599]
[0.05361816 0.71539075 0.60645637]
[0.95000384 0.31424729 0.41032467]
[0.36082793 0.50101268 0.6306832 ]]

>>> np.set_printoptions(precision = 3)
>>> print(rand_arr)
[[0.33   0.055   0.059]
[0.362   0.488   0.263]
[0.054   0.715   0.606]
[0.95   0.314   0.41 ]
[0.361   0.501   0.631]]
```

实验 5.2　Pandas 库的基础操作

【实验要求】

(1) 掌握 Pandas 库的基本功能。

(2) 掌握数据的导入与导出。

(3) 掌握数据的统计与分组。

(4) 掌握数据筛选和过滤功能。

【实验步骤】

(1) 导入 Pandas 库。

导入 Pandas 库,设置别名为 pd,代码如下:

```
>>> import pandas as pd
```

(2) 显示 Pandas 的版本号。

```
>>> print(pd.__version__)
3.13.1
```

(3) 导入 staff.xlsx 文件。

```
>>> import pandas as pd
>>> df = pd.read_excel(r'D:\导入导出文件\staff.xlsx', usecols = ['gonghao', 'name','position',
'salary'])
#读取工号,姓名,职务,工资字段,使用默认索引
>>> print(df[:10])                                    #输出前 5 行数据
   gonghao  name   sex   birthday          position   salary
0    102    张蓝     女   1978-03-20        总经理     8000
1    301    李建设   男   1980-10-15          经理     5650
2    402    赵也声   男   1977-08-30          经理     4200
3    404    章曼雅   女   1985-01-12          经理     5650
4    704    杨明     男   1973-11-11        保管员     2100
```

(4) 用 describe()函数实现统计汇总。

```
>>> df.describe()
```

运行结果如图 5.1 所示。

	gonghao	salary
count	10.000000	10.000000
mean	772.700000	3834.000000
std	437.624408	2094.432832
min	102.000000	1860.000000
25%	402.500000	1995.000000
50%	902.500000	3530.000000
75%	1177.250000	5287.500000
max	1205.000000	8000.000000

图 5.1 describe()函数运行结果

(5) 按照性别分组,查询女员工的工资。

```
>>> grouped = df.groupby('sex')                    #按性别分组
>>> print (grouped.get_group('女'))                #输出女员工的工资
   gonghao  name  sex      position  salary
0      102   张蓝    女       总经理    8000
3      404   章曼雅  女         经理    5650
6     1103   张其    女       业务员    1860
8     1203   任德芳  女       业务员    1960
9     1205   刘东珏  女       业务员    1860
```

(6) 查询男女各组的平均工资。

```
>>> grouped = df.groupby('sex')
>>> print(grouped['salary'].agg(np.mean))
sex
女    3866
男    3802
Name: salary, dtype: int64
```

执行上面的代码,可知女员工的平均工资为 3866 元,男员工的平均工资为 3802 元。

(7) 对数据源 staff.csv 中的数据,按照工资的降序进行排序并显示工资最高的 3 位员工。

```
>>> sorted_df = df. sort_values(by = 'staff')      #按列的值排序
>>> sorted_df.head(3)                              #显示工资最高前 3 名
 gonghao    name  sex  position  salary
0   102     张蓝   女   总经理     8000
1   301   李建设   男   经理       5650
3   404   章曼雅   女   经理       5650
```

(8) 筛选工资低于 3000 元的员工。

```
>>> df_salary = df [(df.salary < 3000)]
>>> print(df_salary)
   gonghao  name   sex   position   salary
4      704   杨明     男   保管员   2100
6     1103   张其     女   业务员   1860
7     1202   石破天   男   业务员   2860
8     1203   任德芳   女   业务员   1960
9     1205   刘东珏   女   业务员   1860
```

(9) 计算男员工的最高工资和最低工资。

```
>>> df_max =  df[(df.sex == '男')].salary.max()          #计算男员工的最高工资
>>> print(df_max)
5650
>>> df_min =  df[(df.sex == '男')].salary.min()          #计算男员工的最低工资
>>> print(df_min)
2100
```

(10) 筛选 name、sex、position 三列数据。

```
>>> df_filter = df.filter(items = ['name','sex',  'position'])      #筛选需要的列
>>> print(df_filter)
    name     sex      position
0   张蓝      女       总经理
1   李建设    男         经理
2   赵也声    男         经理
3   章曼雅    女         经理
4   杨明      男       保管员
5   王宜淳    男         经理
6   张其      女       业务员
7   石破天    男       业务员
8   任德芳    女       业务员
9   刘东珏    女       业务员
```

实验 5.3　Matplotlib 库的画图基础

【实验要求】

(1) 掌握 Matplotlib 库的画图基本功能。

(2) 掌握折线图的绘制方法。

(3) 掌握散点图的绘制方法。

(4) 掌握柱形图的绘制方法。

(5) 掌握饼图的绘制方法。

【实验步骤】

(1) 使用 pip 安装 Matplotlib。

```
pip install matplotlib
```

(2) 导入模块并起别名为 plt。

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

(3) 简单绘图。

简单绘图命令格式如下：

```
plt.plot(x, y)                                    #绘制 y = f(x) 的图像
plt.show()                                        #显示图像
```

下面绘制 10 个随机数构成的折线图,代码如下：

```
import numPy as np
import pandas as pd
import matplotlib .pyplot as plt
plt.show()                                        #图表窗口
plt.plot(np.random.rand(10))                      #基础绘图
```

输出结果如图 5.2 所示。

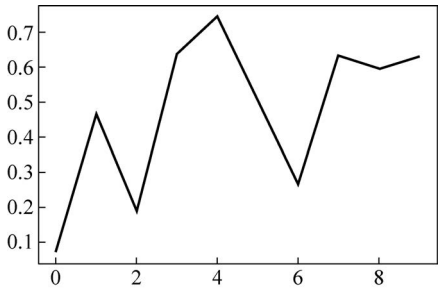

图 5.2　可视化视图

(4) 在 Matplotlib 中使用中文。

```
#在 Matplotlib 中使用中文
from pylab import mpl
mpl.rcParams['font.sans - serif'] = ['FangSong']   #指定默认字体为仿宋体
mpl.rcParams['axes.unicode_minus'] = False         #解决保存图像是负号'-'显示为方块的问题
```

(5) 绘制一个简单折线图。

折线图是以折线的上升或下降来表示统计数量的增减变化的统计图。

特点：能够显示数据的变化趋势,反映事物的变化情况。

绘制折线图需要调用 plot()函数,格式如下：

```
plt.plot(x, y)
```

下面实现由一个随机数构成的简单折线图，代码如下：

```
#简单折线图
import matplotlib.pyplot as plt
import pandas as pd
import numPy as np
#plt.rcParams['font.sans-serif'] = ['SimHei']                    #用来正常显示中文标签
#plt.rcParams['axes.unicode_minus'] = False                      #用来正常显示负号
df = pd.DataFrame(np.random.rand(10,2),columns=['A','B'])
fig = df.plot(figsize=(8,4))                                     #创建图表窗口,设置窗口大小
plt.title('Title')                                               #图名
plt.xlabel('x轴')                                                #x轴标签
plt.ylabel('y轴')                                                #y轴标签
plt.legend(loc = 'upper right')                                  #显示图例,loc表示位置
plt.show()
```

这样一个简单的绘图就出来了，这里面有两条折线图，位于一块画布上，输出结果如图 5.3 所示。

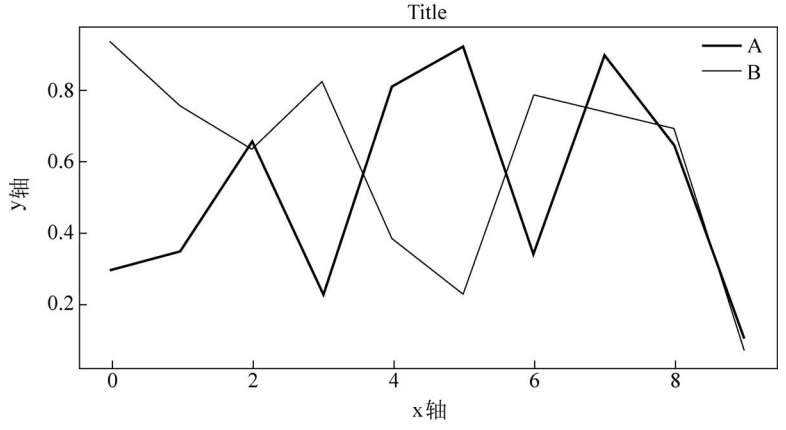

图 5.3　简单折线图

(6) 绘制散点图。

散点图主要是表示 x 和 y 之间的关系。采用两组数据构成多个坐标点，考查坐标点的分布，判断两变量之间是否存在某种关联或总结坐标点的分布模式。

特点：判断变量之间是否存在数量关联趋势，展示离群点(分布规律)。

绘制散点图需要调用 scatter()函数，格式如下：

```
plt.scatter(x, y)
```

使用 NumPy.random 模块中的 randn()函数生成两组数据，使用 scatter()函数绘制散点图，代码如下：

```
#绘制简单的散点图
x = np.random.randn(1000)
y = np.random.randn(1000)
plt.scatter(x,y)
```

```
plt.title('scatter')
plt.show()
```

输出结果如图 5.4 所示。

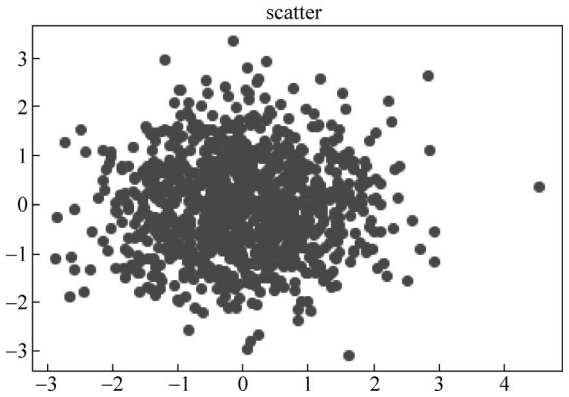

图 5.4　散点图

(7) 绘制简单的柱形图。

柱形图是将排列在工作表的列或行中的数据绘制到柱状图中。

特点：绘制离散的数据，能够一眼看出各个数据的大小，比较数据之间的差别。

绘制折线图需要调用 plot()函数，格式如下：

```
plt.bar(x, width, align = 'center', ** kwargs)
```

实现绘制一个简单柱形图的代码如下：

```
#简单柱形图
import matplotlib.pyplot  as  plt
data = [5, 20, 15, 25, 10]
plt.bar(range(len(data)), data)
plt.show()
```

输出结果如图 5.5 所示。

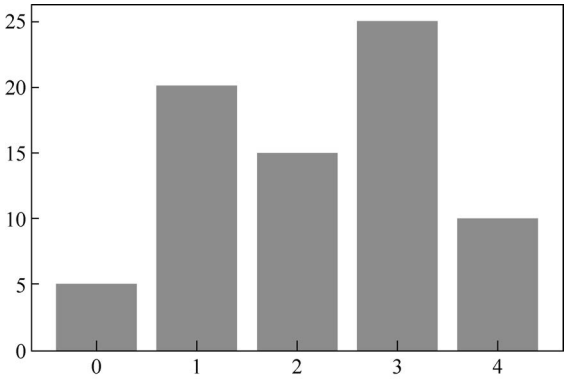

图 5.5　柱形图

(8) 统计各部门员工人数,绘制柱形图。

首先,导入“部门员工.csv”文件,显示部门员工表。

```
import pandas as pd
data = pd.read_csv(r'D:\导入导出文件\部门员工.csv')
data.head()
```

输出如图 5.6 所示的部门员工数据表。

	Unnamed: 0	gonghao	name	sex	birthday	department	position	salary
0	0	102	张蓝	女	1978-03-20	总经理室	总经理	8000
1	1	301	李建设	男	1980-10-15	人事部	经理	5650
2	2	402	赵也声	男	1977-08-30	财务部	经理	4200
3	3	404	章曼雅	女	1985-01-12	财务部	经理	5650
4	4	704	杨明	男	1973-11-11	书库	保管员	2100

图 5.6　部门员工数据表

然后,计算每个部门的人数,这里需要用到循环语句实现在每个柱上显示标签。柱形图参数如下所示。

- get_x(): 表示每个柱形的 x 轴位置。
- i.get_width(): 表示每个柱形的宽度。
- i.get_height(): 表示每个柱形的高度。

绘制柱形图的代码如下:

```
#绘制柱形图
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']          #用来正常显示中文标签
dep = data['department'].value_counts()
print(dep)
b = plt.bar(dep.index,dep.values)                     #柱状图
plt.yticks(range(6))                                  #设置 y 轴显示 1～5 的数
for i in b:                                           #循环每个柱上显示人数的标签
   plt.text(i.get_x() + i.get_width() / 2,i.get_height() + 0.1,str(i.get_height()))
```

输出结果如图 5.7 所示。

(9) 绘制饼图。

接(8)中的实例,绘制饼图,格式如下:

```
plt.pie()
```

参数如下所示。

- startangle: 设置饼图的起始角度。
- explode: 每块的顶点距离圆心的长度。
- autopct: 设置比例值,小数位数保留几位。
- labeldistance: 设置标签到圆心的距离。

绘制饼图的代码如下:

```
plt.pie(dep,labels = dep.index,autopct = '%.2f%%')
```

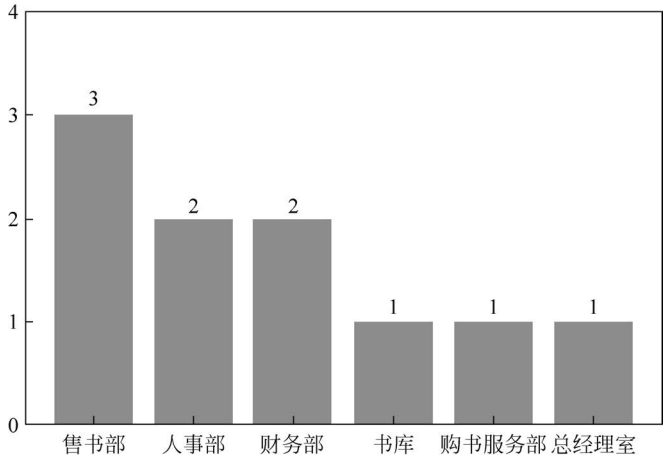

图 5.7　部门员工人数柱形图

输出结果如下：

```
([<Matplotlib.patches.Wedge at 0x20c92d546d0>,
  <Matplotlib.patches.Wedge at 0x20c92d54b80>,
  <Matplotlib.patches.Wedge at 0x20c92d63250>,
  <Matplotlib.patches.Wedge at 0x20c92d63880>,
  <Matplotlib.patches.Wedge at 0x20c92d63f10>,
  <Matplotlib.patches.Wedge at 0x20c92d6e5e0>],
[Text(0.6465637441936395, 0.8899187180267095, '售书部'),
 Text(-0.8899187482945419, 0.6465637025335369, '人事部'),
 Text(-0.8899186272232008, -0.6465638691739386, '财务部'),
 Text(1.2873679044788556e-07, -1.0999999999999925, '书库'),
 Text(0.646563890003987, -0.8899186120892812, '购书服务部'),
 Text(1.0461622140716127, -0.3399185517867209, '总经理室')],
[Text(0.3526711331965306, 0.48541020983275057, '30.00%'),
 Text(-0.4854102263424773, 0.3526711104728383, '20.00%'),
 Text(-0.48541016030356404, -0.3526712013676028, '20.00%'),
 Text(7.022006751702848e-08, -0.5999999999999959, '10.00%'),
 Text(0.3526712127294474, -0.48541015204869875, '10.00%'),
 Text(0.5706339349481523, -0.18541011915639322, '10.00%')])
```

绘制的部门员工人数比例饼图如图 5.8 所示。

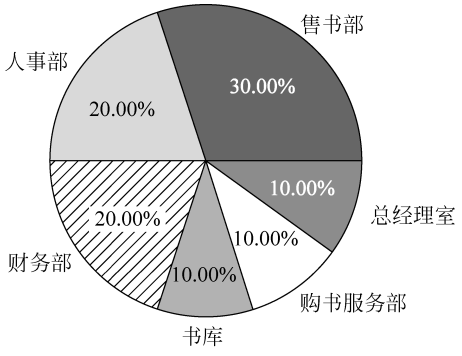

图 5.8　部门员工人数比例饼图