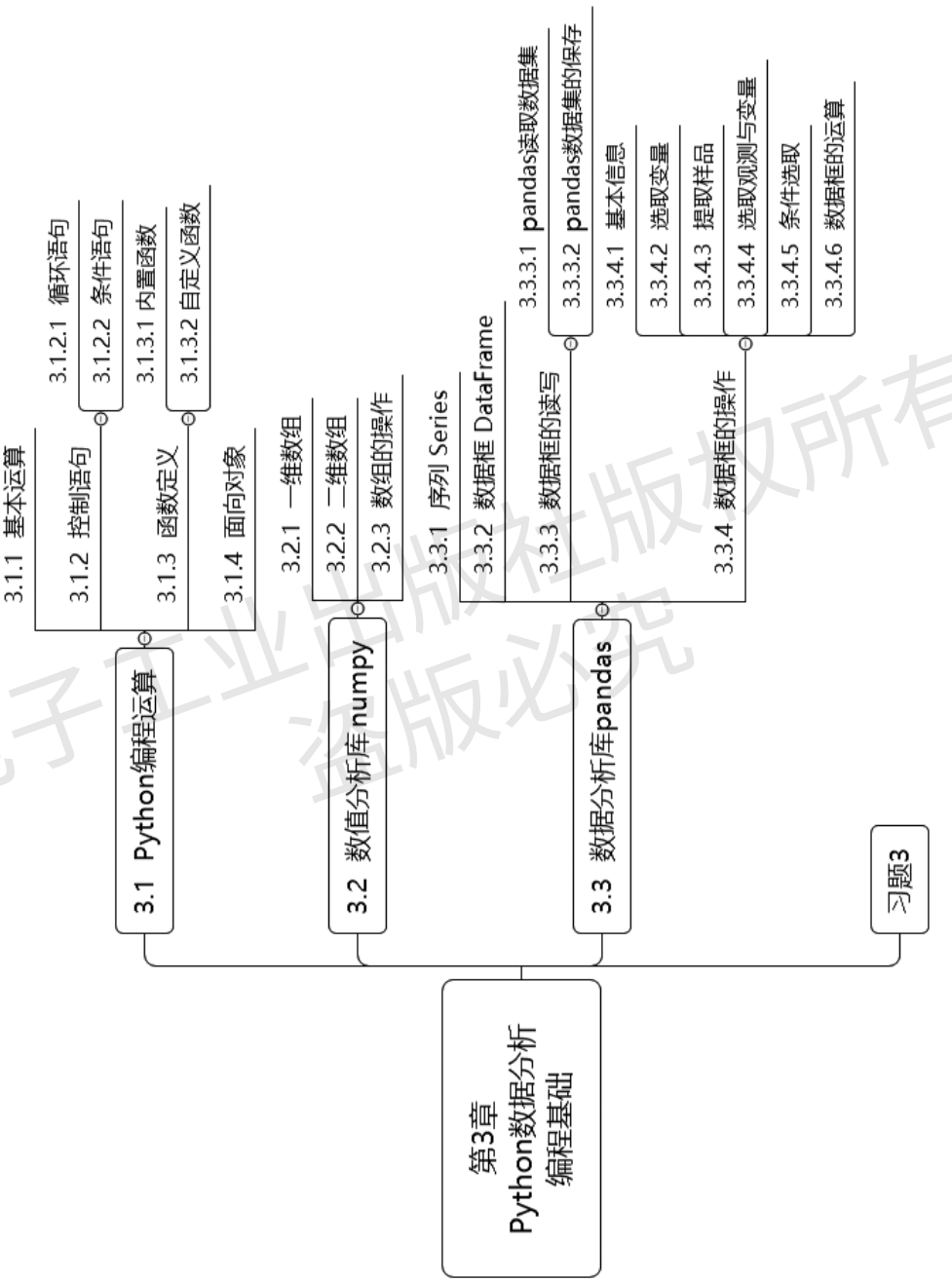
第 3 章　Python 数据分析编程基础



第 3 章思维导图

网上有大量的 Python 编程基础知识介绍，如

```
http://www.runoob.com/Python/Python-dictionary.html
```

请大家自行学习。由于本书重点为介绍 Python 的数据分析，所以对 Python 编程的基础知识将不展开讨论。

3.1 Python 编程运算

3.1.1 基本运算

与 Basic、VB、C、C++和 Java 等一样，Python 具有编程功能，但 Python 是新时期的编程语言，具有面向对象的功能，同时 Python 还是面向函数的语言。既然 Python 是一种编程语言，它就具有常规语言的算术运算符和逻辑运算符(见表 3-1)，以及控制语句、自定义函数等功能。下面对 Python 的编程特点做简单介绍。

表 3-1 Python 中常用的算术运算符和逻辑运算符

算术运算符	含　义	逻辑运算符	含　义
+	加	<(<=)	小于(小于等于)
–	减	>(>=)	大于(大于等于)
*	乘	==	等于
/	除	!=	不等于
**	幂	not x	非 x
%	取模	or	或
//	整除	and	与

3.1.2 控制语句

编程离不开对程序的控制，下面介绍几个最常用的控制语句，其他控制语句见 Python 手册。

3.1.2.1 循环语句

Python 的 for 循环可以遍历任何序列的项目，如一个列表或一个字符串。for 循环允许循环使用向量或数列的每个值，在编程中非常有用。

for 循环的语法格式如下：

```
for iterating_var in sequence:
    statements(s)
```

Python 的 for 循环功能比其他语言更为强大，例如：

In	for i in range(1,5):　　#range(1,n)表示 1 到 n–1 的列表 　print(i)

Out	1 2 3 4
In	fruits = ['banana', 'apple', 'mango'] for fruit in fruits: 　　print('当前水果:', fruit)
Out	当前水果: banana 当前水果: apple 当前水果: mango

下面是 for 循环的简洁写法，输出结果仍为列表，非常有用。

In	[i for i in range(1,5)]　#循环的简洁写法
Out	[1, 2, 3, 4]

3.1.2.2　条件语句

if/else 语句是分支语句中的主要语句，其格式如下：

In	a = –100
	if a < 100:
	　　print("数值小于 100")
	else:
	　　print("数值大于 100")

Out	数值小于 100

Python 中有更简洁的形式来表达 if/else 语句。

In	–a if a<0 else a　　#if/else 的简洁语法
Out	100

注意：在循环和条件等语句及下面的函数中要输出结果，须用 print 命令，这时只用变量名等对象是无法显示结果的。

3.1.3　函数定义

3.1.3.1　内置函数

在较复杂的计算问题中，有时一个任务可能需要重复多次，这时不妨自定义函数，这么做的好处是，函数内的变量是局部的，即函数运行结束后它们不再保存到当前的工作空间，这就可以避免许多不必要的混淆和内存空间的占用。

要学好 Python 数据分析，就必须掌握 Python 中的函数及其定义方法。表 3-2 所示是 Python 中常用的数学函数和数组函数。

表 3-2　Python 中常用的数学函数和数组函数

math 包的数学函数	含义（针对数值 x）	numpy 包的数学函数	含义（针对数组 X）
abs（x）	数值的绝对值	len（X）	数组中元素个数
sqrt（x）	数值的平方根	sum（X）	数组中元素求和
log（x）	数值的对数	prod（X）	数组中元素求积
exp（x）	数值的指数	min（X）	数组中元素最小值
round（x,n）	有效位数 n	max（X）	数组中元素最大值
sin（x）,cos（x）,…	三角函数	sort（X）	数组中元素排序
		rank（X）	数组中元素秩次

Python 与其他统计软件最大的区别之一是，可以随时随地自定义函数，而且可以像使用 Python 的内置函数一样使用自定义函数。

3.1.3.2　自定义函数

不同于 SAS、SPSS 等基于过程的统计软件，Python 进行数据分析是基于函数和面向对象进行的，所有 Python 的命令都是以函数形式出现的。由于 Python 是开源的，故所有函数使用者都可以查看其源代码，而且所有人都可以随时定义自己的数据分析函数。下面简单介绍 Python 的函数定义方法。定义函数的句法：

```
def 函数名(参数 1, 参数 2, …):
    函数体
  return
```

函数名可以是任意字符，但之前定义过的要小心使用，后定义的函数会覆盖先定义的函数。

注意：如果函数只用来计算，不需要返回结果，则可用 print 函数，这时只用变量名是无法显示结果的。

一旦定义了函数名，就可以像 Python 的其他函数一样使用，比如，要定义一个用来求一组数据均值的函数，可以用与 C、C++、VB 等语言相同的方式定义，但方便得多。

例如，自定义计算向量 $X=(x_1, x_2,\cdots,x_n)$ 均值的函数 $\bar{x}=\dfrac{\sum_{i=1}^{n} x_i}{n}$，代码如下：

In	def xbar（x）: 　　n=len（x） 　　S=sum（[i for i in x]） 　　xbar=S/n 　　return（xbar）
In	X=[1,3,6,4,9,7,5,8,2]; X xbar（X）
Out	5.0

注意，上述函数中的 x 称为形参(形式参数)，而 X 称为实参(实际参数)。

当然，Python 已内置求列表和数组的函数，可直接使用，如下。其他统计函数计算见第 4 章。

In	import numpy as np np.mean(X)
Out	5.0

要了解任何一个 Python 函数，使用 help()函数即可，例如，命令 help(sum)或?sum 将显示 sum 函数的使用帮助。

3.1.4 面向对象

Python 是一种面向对象的语言(一般使用者可暂不了解)。

前面介绍的 Python 基本数据类型和标准类型都是 Python 的数据对象，各种 Python 函数也是对象。由于 Python 函数的许多计算结果都放在对象中，这使得 Python 的结果通常比 SAS、SPSS 和 Stata 等数据分析软件的结果简洁，需要时才调用，这为进一步分析提供了方便。

下面通过编写一个函数的过程来简单介绍 Python 面向对象函数的编写技术。

例如，计算向量 $X=(x_1, x_2,\cdots, x_n)$ 的离均差平方和函数

$$SS=\sum_{i=1}^{n}(x_i-\bar{x})^2=\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\Big/n$$

有了离均差平方和函数，就可做许多统计计算，如计算方差、标准差，进行方差分析和相关与回归分析等。

In	def SS1(x):
	  n=len(x)
	  S1=sum([i for i in x])
	  S2x=sum([i*i for i in x])
	  Sx2=sum([i for i in x])**2
	  SS=S2x−Sx2/n
	  return(SS)
		#计算离均差平方和函数
		
		#计算列表的和
		#计算列表的平方和
		#计算列表和的平方

In	X=[1,3,6,4,9,7,5,8,2]; SS(X)	
Out	60.0	

Python 一次可以返回多个数据对象，比如，可返回数据的均值、平方和、离均差平方和、方差、标准差，但一般要用到列表类型。这里的列表类型是比数据框更高级的数据对象，相当于非结构化数据类型，有了列表类型，也为大数据分析提供了便利，其原因是大数据中很多数据都呈非结构化特点。下面简单介绍 Python 列表类型的用法，初学者可暂不学习。

In	def SS2(x):　　#返回多个值函数 　　n=len(x) 　　S1=sum([i for i in x])　　#计算列表的和 　　xbar=S1/n 　　S2x=sum([i*i for i in x])　　#计算列表的平方和 　　Sx2=sum([i for i in x])**2　　#计算列表和的平方 　　SS=S2x−Sx2/n 　　return[n,S1,xbar,S2x,Sx2,SS] #返回例数、均值、平方和、和的平方、离均差平方和的列表
In	SS2(X)
Out	[9, 285, 2025, 60.0]

如果一个数据对象需要包含不同类型的数据对象，可以采用列表的形式。

列表中对象的成分访问方式与变量和数据基本一样，可以用下标获取，但不完全一样，在此不详述。

In	SS2(X)[0]　#取第 1 个对象 SS2(X)[1]　#取第 2 个对象 SS2(X)[2]　#取第 3 个对象 SS2(X)[3]　#取第 4 个对象 SS2(X)[4]　#取第 5 个对象 SS2(X)[5]　#取第 6 个对象
Out	9 45 5.0 285 2025 60.0

可以使用 type 函数来查看数据或对象的类型。

In	type(SS2(X))
Out	list
In	type(SS2(X)[3])
Out	float

3.2　数值分析库 numpy

numpy 是使用 Python 进行科学计算的基础软件包，具有 MATLAB 和 R 的大多数数值运算功能。除常用的向量和矩阵运算外，还包括

- 功能强大的多维数组对象
- 精密广播功能函数(简化数组的循环)

- 集成 C/C+和 Fortran 代码的工具
- 强大的线性代数、傅里叶变换和随机数功能

在使用 numpy 库前，须加载其到内存中，语句为 import numpy，通常将其简化为

```
import numpy as np
```

3.2.1 一维数组

一维数组即我们常说的向量。

In	import numpy as np np.array([1,2,3,4,5])	#加载数组包 #一维数组
Out	array([1, 2, 3, 4, 5])	
In	np.array([1,2,3,np.nan,5])	#包含缺失值的数组
Out	array([1., 2., 3., nan, 5.])	
In	np.arange(9) np.arange(1,9,0.5) np.linspace(1,9,5)	#数组序列 #等差数列 #等距数列
Out	array([0, 1, 2, 3, 4, 5, 6, 7, 8]) array([1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7.,7.5,8.,8.5]) array([1., 3., 5., 7., 9.])	

3.2.2 二维数组

二维数组即我们常说的矩阵，但数组可以推广到多维情形。

In	np.array([[1,2],[3,4],[5,6]])	#二维数组
Out	array([[1, 2], [3, 4], [5, 6]])	
In	A=np.arange(9).reshape((3,3));A	#形成 3×3 矩阵
Out	array([[0, 1, 2], [3, 4, 5], [6, 7, 8]])	

3.2.3 数组的操作

(1) 数组的维度

In	A.shape	
Out	(3, 3)	#元组类型

(2) 对角阵

In	np.diag(A)	#对角阵
Out	array([0, 4, 8])	

(3) 零数组

In	np.zeros((3,3))	#零矩阵
Out	array([[0., 0., 0.], [0., 0., 0.], [0., 0., 0.]])	

(4) 1 数组

In	np.ones((3,3))	#1 矩阵
Out	array([[1., 1., 1.], [1., 1., 1.], [1., 1., 1.]])	

(5) 单位阵

In	np.eye(3)	#单位阵
Out	array([[1., 0., 0.], [0., 1., 0.], [0., 0., 1.]])	

3.3 数据分析库 pandas

在数据分析中，数据通常以向量或变量(一维数组，Python 中用序列表示)和矩阵(二维数组，Python 中用数据框表示)的形式出现，下面结合 Python 介绍 pandas 的基本数据操作，详见 https://pandas.pydata.org/pandas-docs/stable/user_guide/index.html#user-guide。

注意：在 Python 编程中，变量通常以列表(一组数据)形式出现，而不是一般编程语言的标量(一个数据)。

3.3.1 序列 Series

(1) 创建序列(向量、一维数组)

假如要创建一个含有 n 个数值的向量 $X=(x_1,x_2,\cdots,x_n)$，Python 中可由列表创建序列，这些列表可以数字的，也可以是字符串的，还可以是混合的。

特别**说明**：Python 中显示数据或对象内容直接用其名称，见下。

(2) 生成序列

In	import pandas as pd pd.Series()	#加载数据分析包 #生成空序列
Out	Series([], dtype: float64)	

(3) 根据列表构建序列

In	X=[1,3,6,4,9];

	weight=[67,66,83,68,70]; sex=['女','男','男','女','男']; S1=pd.Series(X);S1 S2=pd.Series(weight);S2 S3=pd.Series(sex);S3
Out	0 1 1 3 2 6 3 4 4 9 dtype: int64 0 67 1 66 2 83 3 68 4 70 dtype: int64 0 女 1 男 2 男 3 女 4 男 dtype: object

(4) 序列合并

In	pd.concat([S2,S3],axis=0) #按行合并序列
Out	0 67 1 66 2 83 3 68 4 70 0 女 1 男 2 男 3 女 4 男
In	pd.concat([S2,S3],axis=1) #按列合并序列
Out	0 1 0 67 女 1 66 男 2 83 男 3 68 女 4 70 男

(5) 序列切片

In	S1[2] S3[1:4]
Out	6 1　男 2　男 3　女

3.3.2 数据框 DataFrame

pandas 中用函数 DataFrame() 生成数据框。DataFrame() 命令可用序列构成一个数据框，如下面的 df1 和 df2。数据框相当于关系数据库中的结构化数据类型，传统的数据大都以结构化数据存储于关系数据库中，因而传统的数据分析是以数据框为基础的。Python 中的数据分析大都是基于数据框进行的，所以本书的分析也是以数据框形式的数据分析为主，向量和矩阵都可以看成数据框的一个特例。

(1) 生成数据框

In	pd.DataFrame()　　#生成空数据框
Out	–

(2) 根据列表创建数据框

In	pd.DataFrame(X)
Out	0 0　1 1　3 2　6 3　4 4　9
	pd.DataFrame(weight,columns=['weight'], index=['A','B','C','D','E'])
	weight A　67 B　66 C　83 D　68 E　70

(3) 根据字典创建数据框

In	df1=pd.DataFrame({'S1':S1,'S2':S2,'S3':S3}); df1
Out	S1　S2　S3 0　1.0　67　女

	1 3.0 66 男 2 6.0 83 男 3 4.0 68 女 4 9.0 70 男
In	df2=pd.DataFrame({'sex':sex,'weight':weight},index=X);df2
Out	sex weight 1 女 67 3 男 66 6 男 83 4 女 68 9 男 70

(4) 增加数据框列

In	df2['weight2']=df2['weight']**2; df2 #生成新列
Out	sex weight weight2 1 女 67 4489 3 男 66 4356 6 男 83 6889 4 女 68 4624 9 男 70 4900

(5) 删除数据框列

In	del df2['weight2']; df2 #删除数据列
Out	sex weight 1 女 67 3 男 66 6 男 83 4 女 68 9 男 70

(6) 缺失值处理

In	df3=pd.DataFrame({'S2':S2,'S3':S3},index=S1);df3
Out	S2 S3 1 66.0 男 3 68.0 女 6 NaN NaN 4 70.0 男 9 NaN NaN
In	df3.isnull() #若是缺失值则返回 True，否则返回 False
Out	S2 S3 1 False False

	3 False False 6 True True 4 False False 9 True True	
In	df3.isnull().sum()	#返回每列包含的缺失值的个数
Out	S2 2 S3 2	
In	df3.dropna()	#直接删除含有缺失值的行，多变量谨慎使用
Out	S2 S3 1 66.0 男 3 68.0 女 4 70.0 男	

(7) 数据框排序

In	df3.sort_index()	#按 index 排序
Out	S2 S3 1 66.0 男 3 68.0 女 4 70.0 男 6 NaN NaN 9 NaN NaN	
In	df3.sort_values(by='S3')	#按 S3 列值排序
Out	S2 S3 3 68.0 女 1 66.0 男 4 70.0 男 6 NaN NaN 9 NaN NaN	

3.3.3 数据框的读写

3.3.3.1 pandas 读取数据集

大量的数据常常是从外部文件读入，而不是在 Python 中直接输入的。外部的数据源有很多，可以是电子表格、数据库、文本文件等形式。Python 的导入工具非常简单，但是对导入文件有一些比较严格的限制。

前面我们讲到，电子表格是目前进行数据管理和编辑最为方便的工具，所以可以考虑用电子表格管理数据，用 Python 分析数据(适用于全书)，电子表格与 Python 之间的数据交换过程非常简单。

本书使用的是 pandas 包读取数据的方式，须事先调用 pandas 包，即

```
import pandas as pd
```

(1) 从剪切板上读取

先在 DaPy_data.xls 数据文件的【Bsdata】表中选取 A1:H5，复制，然后在 Python 中读取数据。使用剪切板命令(clipboard)可复制任何数据。

In
```
import pandas as pd
Bsdata=pd.read_clipboard();    #从剪切板上复制数据
Bsdata
```

Out

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python

这里，Bsdata 为读入 Python 中的数据框名，clipboard 为剪切板。

(2) 读取 csv 格式数据

虽然 Python 可以直接复制表格数据，但也可读取电子表格工作簿中的一个表格(例如，在 Excel 中将数据 DaPy_data.xlsx 的表单[Bsdata]另存为 DaPy_BS.csv，csv 格式数据本质上也是文本文件，是以逗号分隔的文本数据，既可用记事本打开，也可用电子表格软件打开，是最通用的数据格式)，其读取命令也最为简单，如下所示。

In
```
Bsdata=pd.read_csv("DaPy_BS.csv",encoding='utf-8')  #有时需用 GBK 格式
Bsdata
```

Out

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python
⋮								

(3) 读取 Excel 格式数据

使用 pandas 包中的 read_excel 可直接读取 Excel 文档中的任意表单数据，其读取命令也比较简单(建议使用)，例如，要读取 DaPy_data.xlsx 表单的[Bsdata]，可用以下命令。

In
```
Bsdata=pd.read_excel('DaPy_data.xlsx','Bsdata'); Bsdata
```

Out

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python
⋮								

(4) 读取其他统计软件的数据

要调用 SAS、SPSS、Stata 等统计软件的数据集，须先用相应的包，详见 Python 手册。

3.3.3.2 pandas 数据集的保存

Python 读取和保存数据集的最好方式是 csv 和 xlsx 文件格式，pandas 保存数据的命令也很简单，如下所示。

In	#将数据框 Bsdata 保存到 Bsdata.csv 中
	Bsdata.to_csv('Bsdata.csv')

In	#将数据框 Bsdata 保存到 Bsdata.xlsx 中 Bsdata.to_excel('Bsdata.xlsx',index=False)　　#index=False 表示不保存行标签

3.3.4 数据框的操作

3.3.4.1 基本信息

(1) 数据框显示

有三种显示数据框内容的函数，即 info(显示数据结构)、head(默认显示数据框前 5 行)、tail(默认显示数据框后 5 行)。

In	Bsdata.info()　　#数据框信息
Out	<class 'pandas.core.frame.DataFrame'> RangeIndex: 52 entries, 0 to 51 Data columns (total 8 columns): # Column Non-Null Count Dtype --- ------ -------------- ----- 0 学号 52 non-null int64 1 性别 52 non-null object 2 身高 52 non-null int64 3 体重 52 non-null int64 4 支出 52 non-null float64 5 开设 52 non-null object 6 课程 52 non-null object 7 软件 52 non-null object dtypes: float64(1), int64(3), object(4) memory usage: 3.4+ KB
In	Bsdata.head()　　#显示前 5 行

Out

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel

	4 1512331015 男 154 55 25.9 有必要 都学习过 Python
In	Bsdata.tail() #显示后 5 行
Out	学号 性别 身高 体重 支出 开设 课程 软件 47 1538319004 男 175 68 44.4 不清楚 统计方法 SAS 48 1538254010 女 166 65 5.3 不清楚 编程技术 Python 49 1540294017 女 159 58 71.4 不清楚 都学习过 SPSS 50 1540365026 女 169 73 5.5 有必要 统计方法 Excel 51 1540388036 女 165 67 56.8 不必要 概率统计 SAS

(2) 数据框列名(变量名)

In	Bsdata.columns #查看列名称
Out	Index(['学号', '性别', '身高', '体重', '支出', '开设', '课程', '软件'], dtype='object')

(3) 数据框行名(样品名)

In	Bsdata.index #数据框行名
Out	RangeIndex(start=0, stop=52, step=1)

(4) 数据框维度

In	Bsdata.shape #显示数据框的行数和列数 Bsdata.shape[0] #数据框行数 Bsdata.shape[1] #数据框列数
Out	(52, 8) 52 8

(5) 数据框值(数组)

In	Bsdata.values[:5] #数据框值数组
Out	array([[1510248008, '女', 167, 71, 46.0, '不清楚', '都未学过', 'No'], [1510229019, '男', 171, 68, 10.4, '有必要', '概率统计', 'Matlab'], [1512108019, '女', 175, 73, 21.0, '有必要', '统计方法', 'SPSS'], [1512332010, '男', 169, 74, 4.9, '有必要', '编程技术', 'Excel'], [1512331015, '男', 154, 55, 25.9, '有必要', '都学习过', 'Python']], dtype=object)

3.3.4.2 选取变量

选取数据框中变量的方法主要有以下几种。

(1)[' ']或“.”法：这是 Python 中最直观的选取变量的方法，比如，要选取数据框 Bsdata 中的“身高”和“体重”变量，直接用“Bsdata.身高”与“Bsdata.体重”即可，也可用 Bsdata['身高']与 Bsdata['体重']，该方法书写比“.”法烦琐，却是不容易出错且直观的一种方法，可推广到多个变量的情形，推荐使用。

In	Bsdata['身高'] #选取一列数据，一列时也可用“Bsdata.身高”

Out	0　　167 1　　171 2　　175 3　　169 4　　154 ⋮
In	Bsdata[['身高','体重']]　#选取两列数据
Out	身高　体重 0　167　71 1　171　68 2　175　73 3　169　74 4　154　55 ⋮

(2)下标法：由于数据框是二维数组(矩阵)的扩展，所以也可以用矩阵的列下标来选取变量数据，用这种方法进行矩阵(数据框)运算比较方便。比如，dat.iloc[i,j]表示数据框(矩阵)的第 i 行、第 j 列数据，dat.iloc[i,]表示 dat 的第 i 行数据向量，而 dat.iloc[,j]表示 dat 的第 j 列数据向量(变量)。再如，“身高”和“体重”变量在数据框 Bsdata 的第 3、4 两列。但要注意，Python 的下标是从 0 开始的。

In	Bsdata.iloc[:,2]　　#选取第 1 列
Out	0　　167 1　　171 2　　175 3　　169 4　　154 ⋮
In	Bsdata.iloc[:,2:4]　　#选取第 3、4 列
Out	身高　体重 0　167　71 1　171　68 2　175　73 3　169　74 4　154　55 ⋮

3.3.4.3　提取样品

In	Bsdata.loc[3]　　#提取第 4 行
Out	学号　　1512332010 性别　　男 身高　　169

	体重 74 支出 4.9 开设 有必要 课程 编程技术 软件 Excel
In	Bsdata.loc[3:5] #提取第 3 至 5 行
Out	学号 性别 身高 体重 支出 开设 课程 软件 3 1512332010 男 169 74 4.9 有必要 编程技术 Excel 4 1512331015 男 154 55 25.9 有必要 都学习过 Python 5 1516248014 男 183 76 85.6 不必要 编程技术 Excel

3.3.4.4 选取观测与变量

同时选取观测与变量数据的方法就是将选取变量和提取样品方法结合使用。例如，我们要选取数据框中男生的部分数据，可用以下语句。

In	Bsdata.loc[:3,['身高','体重']]
Out	身高 体重 0 167 71 1 171 68 2 175 73 3 169 74
	Bsdata.iloc[:3,:5] #选取第 0 至 2 行和 1 至 5 列数据
	学号 性别 身高 体重 支出 0 1510248008 女 167 71 46.0 1 1510229019 男 171 68 10.4 2 1512108019 女 175 73 21.0

3.3.4.5 条件选取

例如，选取身高超过 180cm 的男生的数据以及身高超过 180cm 且体重小于 80kg 的男生的数据，可用以下语句。

In	Bsdata[Bsdata['身高']>180]
Out	学号 性别 身高 体重 支出 开设 课程 软件 5 1516248014 男 183 76 85.6 不必要 编程技术 Excel 10 1520100029 男 184 82 10.3 有必要 都学习过 SAS 21 1525352033 男 185 83 5.1 有必要 都学习过 SPSS 32 1530243029 男 186 87 9.5 不必要 都未学过 No
In	Bsdata[(Bsdata['身高']>180) & (Bsdata['体重']<80)]
Out	学号 性别 身高 体重 支出 开设 课程 软件 5 1516248014 男 183 76 85.6 不必要 编程技术 Excel

3.3.4.6 数据框的运算

(1) 生成新的数据框

可以通过选择变量名来形成新的数据框。

In
```
Bsdata['体重指数']=Bsdata['体重']/(Bsdata['身高']/100)**2
round(Bsdata[:5],2)
```

Out

	学号	性别	身高	体重	支出	开设	课程	软件	体重指数
0	1510248008	女	167	71	46.0	不清楚	都未学过	No	25.46
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab	23.26
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS	23.84
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel	25.91
4	1512331015	男	154	55	25.9	有必要	都学习过	Python	23.19

(2) 数据框的合并 concat()

可以用 pd.concat()将两个或两个以上向量、矩阵或数据框合并起来，参数 axis=0 表示按行合并，axis=1 表示按列合并。

In
```
pd.concat([Bsdata.身高, Bsdata.体重],axis=0)  #按行合并
```

Out
```
0    167
1    171
2    175
3    169
4    154
⋮
```

In
```
pd.concat([Bsdata.身高, Bsdata.体重],axis=1)  #按列合并
```

Out

	身高	体重
0	167	71
1	171	68
2	175	73
3	169	74
4	154	55
⋮		

(3) 数据框转置.T

In
```
Bsdata.iloc[:3,:5].T
```

Out

	0	1	2
学号	1510248008	1510229019	1512108019
性别	女	男	女
身高	167	171	175
体重	71	68	73
支出	46	10.4	21

习题 3

一、选择题

1．以下哪个选项可以创建一个 3×3 的单位矩阵？_____

A．np.range(3,3)　B．np.zeros(3)　　C．np.eye(3)　　D．np.eye[3]

2．以下哪个选项可以显示数据框 Bsdata 的数据结构？_____

A．Bsdata.info()　B．Bsdata.head()　　C．Bsdata.tail()　D．Bsdata.index()

3．关于 pandas 库的 DataFrame 对象，哪个说法是正确的？_____

A．DataFrame 是二维带索引的数组，索引可自定义

B．DataFrame 与二维 ndarray 类型在数据运算上方法一致

C．DataFrame 只能表示二维数据

D．DataFrame 由两个 Series 组成

4．有如下代码：

```
import pandas as pd
a = pd.Series([9, 8, 7, 6], index=['a','b','c','d'])
```

哪个是 print(a.index)的结果？_____

A．[9, 8, 7, 6]　　B．['a','b','c','d']

C．('a','b','c','d')　　D．Index(['a','b','c','d'])

5．下面两段代码，哪个说法不正确？_____

```
import numpy as np
a = np.array([0, 1, 2, 3, 4])
import pandas as pd
b = pd.Series([0, 1, 2, 3, 4])
```

A．a 和 b 是不同的数据类型，不能直接运算

C．a 和 b 都是一维数据

B．a 和 b 表达同样的数据内容

D．a 参与运算的执行速度比 b 快

6．以下哪一个步骤不属于数据清洗？_____

A．去重　　B．删除缺失值　　C．数据合并　　D．异常值检测

7．下面关于 Series 和 DataFrame 的理解，哪个是不正确的？_____

A．DataFrame 表示带索引的二维数据

B．Series 和 DataFrame 之间不能进行运算

C．Series 表示带索引的一维数据

D．可以像对待单一数据一样对待 Series 和 DataFrame 对象

8．阅读如下代码：

```
import pandas as pd
dt = {'one': [', 8, 7, 6], 'wo': [', 2, 1, 0]}
a = pd.DataFrame(dt)
```

希望获得['one','t'o']，应使用如下哪个语句？_____

A．a.index　　B．a.row　　C．a.values　　D．a.columns

二、分析题

1．请创建下列 Python 数组，并计算。

(1) 创建一个 2×2 的数组，计算对角线上元素的和。

(2) 创建一个长度为 9 的一维数据，数组元素为 0～8，并将它重新变为 3×3 的二维数组。

(3) 创建两个 3×3 的数组，分别将它们合并为 3×6、6×3 的数组后，拆分为 3 个数组。

2．调查数据。某公司对财务部门人员的抽烟状态进行调查，结果为：否，否，否，是，是，否，否，是，否，是，否，否，是，是，否，是，否，否，是，是。

(1) 请用列表录入该数据。

(2) 请将这组数据输入电子表格，并将其读入 Python。

3．对第 2 章建立的电子表格 mydata1.xlsx 数据，完成如下任务。

(1) 将上面的分析题 1 和 2 的数据写入其中的表单。

(2) 分别用 Python 的 read_csv 和 read_excel 函数读取。

(3) 对其中的学生数据，用 Python 方法获取性别、数学成绩和统计学成绩变量，并筛选不同性别学生的成绩。