Python 数据分析与可视化习题答案

本书还为教师提供了额外的习题库，网站：www.qingline.net　教师可以联系微信：15620828006

第 1 章 数据分析与可视化概述

1.8 本章习题

一、选择题

（D ）

（ C ）

（ C ）

二、简答题

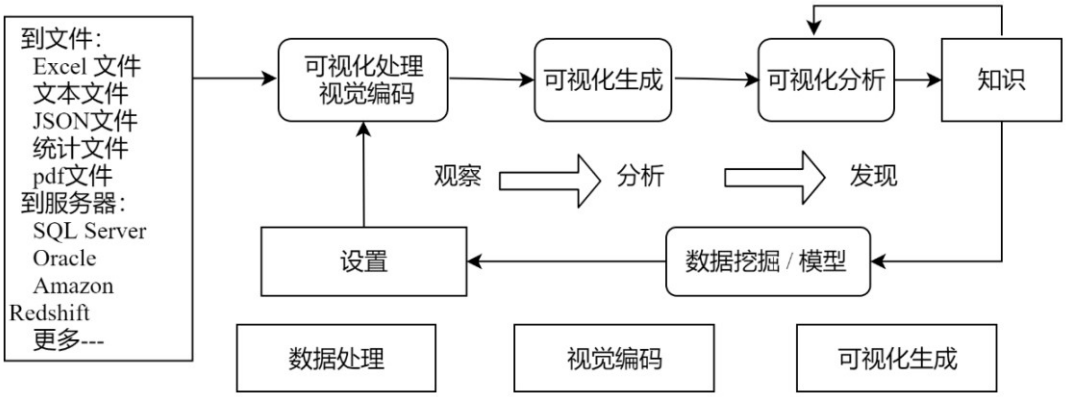
1.

数据分析与数据挖掘之间的区别

差异类型	数据分析	数据挖掘
定义	描述和探索性分析，评估现状和修正不足	技术性的“采矿”过程，发现未知的模式和规律
侧重点	实际的业务知识	实际的业务知识
技能	统计学、数据库、Excel、可视化等	过硬的数学功底和编程技术
结果	需结合业务知识解读统计结果	模型或规则

2.

数据可视化分析过程如图 1-2 所示，包括数据处理、视觉编码和可视化生成。数据处理聚焦于数据的采集、清理、预处理、分析和挖掘；视觉编码聚焦于解决对光学图像进行接收、提取信息、加工变换、模式识别及存储显示；可视化生成则聚焦于解决将数据转换成图形，并进行交互处理。数据可视化分析通过对数据不断地观察、分析从而发现有用的信息模式。



3.

Python 第三方包的安装方式较多，本书建议采用以下方式进行安装和管理。

（1）在 CMD 命令窗口中，使用 conda 命令进行自动下载安装，用法如下：

conda install <包名称列表>　　#安装包

conda remove <包名称列表>　#卸载包

conda search<搜索项>　　#搜索包

conda list　　#查看所有包

conda update<包名称>　　#升级包

（2）在 CMD 命令窗口中使用 pip 命令，用法如下：

pip install <包名>　　#安装包

pip install--upgrade〈包名〉　　#更新包

pip uninstall <包名>　　#删除包

也可以在 Jupyter notebook 的 cell 中运行 pip 命令执行相应的命令，只需在命令前加“！”，如执行 ！pip install 包名 进行包的安装。

4.

按 esc 键切换为命令模式，按 enter 键进入编辑模式

第 2 章 Python 编程基础

2.6 本章习题

一、判断对错

（ × ）

（ ✓ ）

（ ✓ ）

（ ✓ ）

（ ✓ ）

（ × ）

（ ✓ ）

（ × ）

（ × ）

（ × ）

二、填空题

1.不是

2.__None__。

3.__r__。

4.__97__。

三、编程题

编程题答案略

第 3 章 Numpy 数值计算基础

3.7 本章习题

一、选择题

（ B ）

（ C ）

（ D ）

二、填空题

1. 1 和 7 ； 4 和 7 ； 2 。

2.元素是 1 和 9 。

3. array([[0, 1],

[4, 5]]), array([[2, 3], [6, 7]])] 。

4. transpose

5. np.linspace(0,1,12) 。

6. 直接排序 、 将 x 中的元素从小到大排列，提取其对应的 index(索引) 和 对数组或列表按照某一行或列进行排序 。

7.

```
a= np.zeros((10,10), dtype =int)
a[0,:]=1
a[:,9]=1
a[:,0]=1
a[9,:]=1
```

8. arr[arr%2!=0]=-1 。

第 4 章部分 Pandas 统计分析基础

4.9 本章习题

一、填空题

1. reset_index 。

2. 索引

3. 位置

4. 数据元素 和 维度 。

5. 前向填充 和 后向填充 。

6. 抽取比例 。

7. 是否在原数据上修改 。

8. NAN 值 。

9. 将函数套用到数据的的行与列上 。

10. kind 。

二、判断对错

1. （ × ）

2. （ ✓ ）

3. （ × ）

4. （ ✓ ）

5. （ ✓ ）

6. （ × ）

第 5 章 Pandas 数据载入与预处理

5.7 本章习题

一、单项选择题

1. （ C ）

2.（ D ）

二、判断对错题

1. （ ✓ ）

2. （ × ）

3.（×）

4. （× ）

5.（ ✓ ）

三、简答题

1.

dropna 中的参数 thresh 当传入 thresh = N 时，表示要求一行至少具有 N 个非 NaN 才能存活。

2.

a.散点图方法观察 b. 箱线图分析 c. 3σ 法则，原理略，见课本。

3.

不同特征之间往往具有不同的量纲，由此造成数值间的差异很大。因此为了消除特征之间量纲和取值范围的差异可能会造成的影响，需要对数据进行标准化处理。

4.

将数据的值域划分成具有相同宽度的区间，区间个数由数据本身的特点决定或由用户指定。Pandas 提供了 cut 函数，可以进行连续型数据的等宽离散化。cut 函数的基础语法格式为：

pandas.cut(x,bins,right=True,labels=None,retbins=False,precision=3)

第 6 章 Matplotlib 数据可视化基础

6.7 本章习题

一、单项选择题

1. （ A ）

2. （ B ）

3.（ C ）

4.（ C ）

二、简答题

1.

有时空数据可视化，层次与网络结构数据可视化，文本和跨媒体数据可视化以及多变量数据可视化。

2.

查找到当前用户的配置文件目录，然后用编辑器打开，修改 matplotlibrc 文件，即可修改配置参数。

3.

text 函数在指定位置加入文本注释，annotate 函数在图像实现带有指向型的文本注释。

三、绘图题

1. 根据如下绘制写出相应的代码。

```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
# 创建 x 轴数据，从-pi 到 pi 平均取 256 个点；
```

```
x = np.linspace(-np.pi,np.pi,256,endpoint=True)
# 创建 y 轴数据，根据 X 的值，求正弦和余弦函数；
sin,cos = np.sin(x),np.cos(x)
# 设置正弦函数曲线的颜色为蓝色(blue)，线型为实线，线宽为 2.5mm；
#余弦函数曲线的颜色为红色(red)，线型为实线，线宽为 2.5mm。
plt.plot(x,sin,"b-",lw=2.5,label="正弦 Sin()")
plt.plot(x,cos,"r-",lw=2.5,label="余弦 Cos()")
# 设置坐标轴的范围，将 x 轴、y 轴同时拉伸 1.5 倍，
plt.xlim(x.min()*1.5,x.max()*1.5)
plt.ylim(cos.min()*1.5,cos.max()*1.5)
# 设置 x 轴、y 轴的坐标刻度，
plt.xticks([-np.pi,-np.pi/2,0,np.pi/2,np.pi],[r'$-\pi$',r'$-
\pi/2$',r'$0$',r'$\pi/2$',r'$\pi$'])
plt.yticks([-1,0,1])
# 为图表添加标题“绘图实例之 COS()&SIN()”，字体大小设置为 16，字体颜色设置为绿
色；
plt.title("绘图实例之 COS()&SIN()",fontsize=16,color="green")
# 在图表右下角位置文本为“python-matplotlib”，文本大小为 16，文本颜色为紫色
plt.text(+2.1,-1.4,"python_matplotlib",fontsize=16,color="purple")
# 获取 Axes 对象，并隐藏右边界和上边界；
ax=plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
#  将 x 轴的坐标刻度设置在坐标轴下侧，坐标轴平移至经过零点（0,0）的位置，
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
# 将 y 轴的坐标刻度设置在坐标轴左侧，坐标轴平移至经过零点（0,0）的位置，
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
# 添加图例，图例位置为左上角，图例文字大小为 12，
plt.legend(loc="upper left",fontsize=12)
```

```
# 用绘制散点图的方法在正弦，余弦函数上标注这两个点的位置，设置点大小为 50，设置相应的点颜色；
t1=-np.pi
t2=2*np.pi/3
plt.scatter([t1,],[np.cos(t1),], 50, color ='b')
plt.scatter([t2,],[np.sin(t2),], 50, color ='r')
# 为图表添加带箭头的注释；
plt.annotate(r'$\sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
             xy=(t2,np.sin(t2)),     #点的位置
             xycoords='data',        #注释文字的偏移量
             xytext=(+10,+30),       #文字离点的横纵距离
             textcoords='offset points',
             fontsize=14,        #注释的大小
             arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"))
plt.annotate(r'$\cos(-\pi)=-1$',
             xy=(t1,np.cos(t1)),    #点的位置
             xycoords='data',        #注释文字的偏移量
             xytext=(0,-40),         #文字离点的横纵距离
             textcoords='offset points',
             fontsize=14,     #注释的大小
             arrowprops=dict(arrowstyle="->",connectionstyle="arc3,rad=.2"))
# 获取 x,y 轴的刻度，并设置字体；
for label in ax.get_xticklabels()+ax.get_yticklabels():
    label.set_fontsize(18)
# 绘制填充区域；#设置正弦函数的填充区域，颜色为绿色（green），余弦函数的填充区域，颜色为紫色（purple）
plt.fill_between(x, np.abs(x)<0.5, sin, sin>0.5, color='g', alpha=0.8)
plt.fill_between(x, cos, where=(-2.5<x)&(x<-0.5), color='purple')
# 绘制网格线
plt.grid()
plt.show()
```

第 7 章 Seaborn 可视化

7.5 本章习题

一、判断对错题

（ × ）

（ × ）

（ ✓ ）

（ ✓ ）

二、简答题

1.

Seaborn 通过 set 函数实现风格设置。

seaborn.set(context='notebook', style='darkgrid', palette='deep', font='sans-serif', font_scale=1, color_codes=True, rc=None)

2.

Seaborn 将 matplotlib 的参数划分为两个独立的组合。第一组是设置绘图的外观风格的，第二组主要将绘图的各种元素按比例缩放的，以至可以嵌入到不同的背景环境中。控制这些参数的接口主要有两对方法：

- 控制风格：axes_style(), set_style()
- 缩放绘图：plotting_context(), set_context()

每对方法中的第一个方法（axes_style(), plotting_context()）会返回一组字典参数，而第二个方法（set_style(), set_context()）会设置 matplotlib 的默认参数。

利用 set_style( )设置主题，Seaborn 有五个预设的主题： darkgrid、 whitegrid、dark、white 和 ticks，默认为 darkgrid。

3.

利用 with 语句

```
with sns.axes_style("darkgrid"):
    plt.subplot(2,1,1)
    sinplot( )
```

第 8 章 pyechaerts 可视化

8.5 本章习题

一、简答题

1.

图形绘制过程，基本上所有的图表类型都是这样绘制的：

```
chart_name = Type()          #初始化具体类型图表
chart_name .add()            #添加数据及配置项
chart_name .render()         #生成本地文件（html/svg/jpeg/png/pdf/gif）
chart_name .render_notebook  #在 jupyter notebook 中显示
```

2.

V1 版本开始支持链式调用

```
bar = ( Bar()
        .add_xaxis(["衬衫", "毛衣", "领带", "裤子", "风衣",
"高跟鞋", "袜子"])
        .add_yaxis("商家 A", [114, 55, 27, 101, 125, 27, 105])
        .set_global_opts(title_opts = opts.TitleOpts(title = "某商场销售
情况")) )
bar.render_notebook()
```

3.

桑基图(Sankey diagram)即桑基能量分流图，也叫桑基能量平衡图。它是一种特定类型的流程图，图中延伸的分支的宽度对应数据流量的大小，通常应用于能源、材料成分、金融等数据的可视化分析。Pyecharts 中利用 Sankey 绘制桑基图。

平行坐标图（Parallel Coordinates Plot）是对于具有多个属性问题的一种可视化方法。在平行坐标图中，数据集的一行数据在平行坐标图中用一条折线表示，纵向是属性值，横向是属性类别（用索引表示）。

4.

略

第 9 章 时间序列数据分析

9.8 本章习题

一、填空题

1.

data_range

2.

now( )

3. **时间类型**。

4.下采样 。

二、简答题

1.

在多个时间点观测或测量的数据形成了时间序列。时间序列数据是一种重要的结构化数据类型，如金融、经济、生态学、神经科学和物理学等领域。

2.

用 Python 来进行平稳性检验主要有时序图检验、自相关图检验以及构造统计量进行检验 3 种方法。

（1）时序图

时序图就是普通的时间序列图，即以时间为横轴，观察值为纵轴进行检验。利用时序图可以粗略观察序列的平稳性。

（2）自相关图检验

平稳序列通常具有短期相关性，即随着延迟期数 k 的增加，平稳序列的自相关系数会很快地衰减向零，而非平稳序列的自相关系数的衰减速度会比较慢。画自相关图用到的是 statsmodels 中的 plot_acf 方法。自相关图中横轴表示延迟期数，纵轴表示自相关系数。

（3）构造统计量

利用绘图判断序列的平稳性比较直观，但不够精确，ADF（Augmented Dickey-Fuller）法直接通过假设检验的方式来验证平稳性。ADF 的原假设（H0）和备择假设（H1）如下：

H0：具有单位根，属于非平稳序列；

H1：没有单位根，属于平稳序列。

Python 中可以使用 statsmodels 中的 adfuller 方法进行 ADF 检验，直接输入数据，即可返回 7 个数值。其中的第一个返回值 adf 就是 ADF 方法的检验结果，这个值理论上越负越能拒绝原假设；第二个返回值 pvalue 以常用的判断标准值 0.05 作为参考，若其值大于 0.05，说明支持原假设，反之拒绝原假设，表明该序列是一个平稳序列。

第 10 章 SciPy 科学计算

10.8 本章习题

简答题

1.

Scipy 是一款用于数学、科学和工程领域的 Python 工具包，可以处理插值、积分、优化、图像处理、常微分方程数值解的求解、信号处理等问题。

2.

见课本 P214

3.

SciPy.optimize 包提供了几种常用的优化算法，包括用来求有/无约束的多元标量函数最小值算法，最小二乘法，求有/无约束的单变量函数最小值算法，还有解各种复杂方程的算法。

4.

见课本 P218

5.

见课本 P219

第 11 章　统计与机器学习部分

11.7 本章习题

一、填空题

1. 分类

2. 回归

3. PCA （或主成分分析）

4. 均值

5. 超平面

6. 凝聚 和 分裂

二、简答题

1.

分类是一种重要的数据分析形式，它提取刻画重要数据类的模型。数据分类也被称为监督学习，用来训练分类模型的数据需要有已标注的标签，包括学习阶段（构建分类模型）和分类阶段（使用模型预测给定数据的类标号）两个阶段。

将物理或抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类。由聚类所生成的簇是一组数据对象的集合，这些对象与同一个簇中的对象彼此相似，与其他簇中的对

象相异。聚类不需要有事先标注的标签。

2.。

回归分析可以简单理解为数据分析与预测，通过对数据进行分析实现预测，也就是适当扩大已有自变量的取值范围，并承认该回归方程在扩大的定义域内成立。一般来说，回归分析的主要过程和步骤如下：

（1）收集一组包含因变量和自变量的数据；

（2）根据因变量和自变量之间的关系，初步设定回归模型；

（3）求解合理的回归系数；

（4）进行相关性检验，确定相关系数；

（5）利用模型对因变量作出预测或解释，并计算预测值的置信区间。

3.

KNN(k-Nearest Neighbor Classification)算法根据距离函数计算待分类样本 X 和每个训练样本的距离（作为相似度），选择与待分类样本举例最小的 K 个样本作为 X 的 K 个最近邻，最后以 X 的 K 个最近邻中的大多数样本所属的类别作为 X 的类别。

4.

给定一个分类标签 y 和自由特征变量 $x_1, x_2, \mathrm{L}\ , x_n$，$x_i = 1$ 表示样本具有特征 i，而 $x_i = 0$ 表示样本不具有特征 i。如果要知道具有特征 1 到 n 的向量是否属于分类标签 $\mathrm{y_k}$，可以利用贝叶斯公式如 11.1。

$$P(y_{\mathrm{k}}|x_1, \mathrm{L}\ , x_n) = P(y_{\mathrm{k}})\mathrm{P}(x_1, \mathrm{L}\ , x_n|y_{\mathrm{k}}) \Big/ P(x_1, \mathrm{L}\ , x_n) \qquad (11.1)$$

假定一个属性值在给定类上的影响独立于于其他属性的值“类条件独立性”，则

$$P(x_1, \mathrm{L}\ , x_n|y_{\mathrm{k}}) = \prod_{i=1}^{n} P(x_i \mid y_k) \qquad (11.2)$$

由于 $P(x_1, \mathrm{L}\ , x_n)$ 已定，因此比较 $P(\mathrm{y}_1|x_1, \mathrm{L}\ , x_n)$ 和 $P(\mathrm{y}_2|x_1, \mathrm{L}\ , x_n)$ 时与比较 $P(y_1)\mathrm{P}(x_1, \mathrm{L}\ , x_n|y_1)$ 和 $P(y_2)\mathrm{P}(x_1, \mathrm{L}\ , x_n|y_2)$ 等价。假设共有 m 中标签，只需计算 $P(y_{\mathrm{k}})\mathrm{P}(x_1, \mathrm{L}\ , x_n|y_{\mathrm{k}})$，$k = 1, 2, \mathrm{L}\ , m$，

取最大值作为预测的分类标签，即

$$\hat{y} = \arg\max_{k} P(y)\prod_{i=1}^{n} P(x_i \mid y_k) \qquad (11.3)$$

第 12 章 图像数据分析部分

12.7 本章习题

一、简答题

1.

在计算机视觉项目的开发中，OpenCV 作为较大众的开源库，拥有了丰富的常用图像处理函数库，采用 C/C++语言编写，可以运行在 Linux/Windows/Mac 等操作系统上，能够快速的实现一些图像处理和识别的任务。此外，OpenCV 还提供了 Java、Python、cuda 等的使用接口、机器学习的基础算法调用，从而使得图像处理和图像分析变得更加易于上手，让开发人员更多的精力花在算法的设计上。OpenCV 的主要应用领域有计算机视觉领域方向、物体识别、图像分割、人脸识别、动作识别、运动跟踪等。

2.

Matplotlib 显示图像时是按照 rgb 模式，而通过 OpenCV 使用 cv2.imread()命令读取的彩色图像是 BGR 格式。如果有必要的话可以将其从 BGR 格式转换为 RGB 格式。下面语句使用 cv2.cvtColor()命令实现 BGR 格式到 RGB 或灰度图像的转换。

```
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

3.

略