# 第1章习题

1. **填空题**
2. 数据分析 的目的在于将隐藏在一大批看似杂乱无章的数据信息集中提炼出来有用的数据。
3. Anaconda 中包含了conda、Python在内的超过180个科学包及其依赖项。
4. Jupyter Notebook是一个支持 实时 代码、数学方程、可视化和Markdown的Web应用程序。
5. **判断题**
6. 数据分析是一个有目的地收集和整合数据的过程。（× ）数据分析
7. Python是一门胶水语言，可以轻松地操作其它语言编写的库。（√ ）
8. 如果要卸载指定环境中的包，则直接使用remove命令移除即可。（× ）

-Anaconda remove 包名

1. **选择题**
2. 下列选项中，用于搭接数据仓库和保证数据质量的是（B ）。
3. 数据收集
4. 数据处理
5. 数据分析
6. 数据展现
7. 关于Anaconda的说法中，下列描述错误的是（C ）。
8. Anaconda是一个可以对包和环境进行统一管理的发行版本。
9. Anaconda包含了conda、Python在内的超过180个科学包及其依赖项
10. Anaconda是完全开源的、付费的
11. Anaconda避免了单独安装包时需要配置或兼容等各种问题
12. 关于Anaconda的组件中，可以编辑文档且展示数据分析过程的是（D ）。
13. Anaconda Navigator
14. Anaconda Prompt
15. Spyder
16. Jupyter Notebook
17. 下面列出的数据分析库中，用于绘制数组的2D图形的是（C ）。
18. NumPy
19. Pandas
20. Matplotlib
21. NLTK

# 第2章习题

1. **填空题**
2. 在NumPy中，可以使用数组对象\_\_\_\_ndarray\_\_\_\_执行一些科学计算。
3. 如果ndarray.ndim执行的结果为2，则表示创建的是\_\_二\_\_\_维数组。
4. NumPy的数据类型是由一个类型名和元素\_位\_\_字节数\_\_长\_\_\_的数字组成。
5. 如果两个数组的大小（ndarray.shape）不同，则它们进行算术运算时会出现\_\_\_\_广播\_\_\_\_机制。
6. 花式索引是NumPy的一个术语，是指用整数\_\_\_\_数组或列表\_\_\_\_进行索引。
7. **判断题**
8. 通过empty()函数创建的数组，该数组中没有任何的元素。（n ）
9. 如果没有明确地指明数组中元素的类型，则默认为float64。（y ）
10. 数组之间的任何算术运算都会将运算应用到元素级。（y ）
11. 多维数组操作索引时，可以将切片与整数索引混合使用。（y ）
12. 当通过布尔数组索引操作数组时，返回的数据是布尔数组中False对应位置的值。（n ）返回True对应位置的值
13. **选择题**
14. 下列选项中，用来表示数组维度的属性是（A ）。
15. ndim 维度
16. Shape 形状
17. size
18. dtype
19. 下面代码中，创建的是一个3行3列数组的是（D ）。
20. arr = np.array([1, 2, 3])
21. arr = np.array([[1, 2, 3], [4, 5, 6]])
22. arr = np.array([[1, 2], [3, 4]])
23. np.ones((3, 3))
24. 请阅读下面一段程序：

arr\_2d = np.array([[11, 20, 13],[14, 25, 16],[27, 18, 9]])

print(arr\_2d[1, :1])

（arr\_2d[1, :1]取第 2 行第 1 列元素）

执行上述程序后，最终输出的结果为（A ）。

1. [14]
2. [25]
3. [14, 25]
4. [20, 25]
5. 请阅读下面一段程序：

arr = np.arange(6).reshape(1, 2, 3)

print(arr.transpose(2, 0, 1))维度 高度 宽度 三维，一高度，二宽度

执行上述程序后，最终输出的结果为（C ）。

[[[2 5]]

[[0 3]]

[[1 4]]]

[[[1 4]]

[[0 3]]

[[2 5]]]

[[[0 3]]

[[1 4]]

[[2 5]]]

[[[0]

[3]]

[[1]

[4]]

[[2]

[5]]]

1. 下列函数或方法中，用来表示矢量化三元表达式的是（ A）。
2. where()
3. cumsum()
4. sort()
5. unique()

# 第3章习题

1. **填空题**
2. Pandas是一个基于 Numpy 的Python库。
3. Pandas中有两个主要的数据结构分别为 Series 和 DataFrame 。
4. Series结构由 索引 和 数据 组成。
5. 可以使用 列表 和 字典 创建一个Series对象。
6. 数据排序可以分为 按值排列 和 按索引排列 。
7. **判断题**
8. 在DataFrame中每列的数据都可以看做是一个Series对象。（y ）
9. 使用describe()方法会输出多个统计指标。（y ）
10. from\_arrays()方法是将元组列表转换为MultiIndex对象。（n ）
11. read\_csv()函数和read\_table()函数没有区别可以随意替换使用。（n ）
12. Series对象不存在层次化索引。（n ）
13. **选择题**
14. 下列选项中，描述不正确的是（A ）。
15. Pandas中只有Series和DataFrame这两种数据结构
16. Series是一维的数据结构
17. DataFrame是二维的数据结构
18. Series和DataFrame都可以重置索引
19. 下列选项中，描述正确是（B ）。
20. Series是一维数据结构，其索引在右，数据在左
21. DataFrame是二维数据结构，并且该结构具有行索引和列索引
22. Series结构中的数据不可以进行算术运算
23. sort\_values()方法可以将Series或DataFrame中的数据按照索引排序
24. 下列方法中，可以将元组转换为MultiIndex对象的是（A ）。
25. from\_tuples()
26. from\_arrays()
27. from\_product()
28. from\_list()
29. 下列选项中，哪个方法可以一次性输出多个统计指标（A ）。
30. describe()
31. mean()
32. median()
33. sum()
34. 请阅读下面一段程序：

import pandas as pd

ser\_obj = pd.Series(range(1, 6), index=[5, 3, 0, 4, 2])

ser\_obj.sort\_index()

执行上述程序后，最终输出的结果为（B ）。

5 1

3 2

0 3

4 4

2 5

0 3

2 5

3 2

4 4

5 1

5 1

4 4

3 2

2 5

0 3

2 5

4 4

0 3

3 2

5 1

# 第4章习题

1. **填空题**
2. 数据清洗的目的是让数据具有 完整性 、 唯一性 、 权威性 、 合法性 、 一致性 、等特点。
3. 产生缺失值或空值的原因有 人为原因 和 机械原因 。
4. 拉依达原则在检测异常值时必须保证数据遵守 正态分布 。
5. **判断题**
6. rename()方法可以重命名索引名。（y ）
7. drop\_duplicated()方法可以删除重复值。（y ）
8. 在箱形图中超出上界和下界的值称为异常值。（y ）
9. **选择题**
10. 下列选项中，描述不正确的是。（B ）
11. 数据清洗的目的是为了提高数据质量
12. 异常值一定要删除
13. 可使用drop\_duplicates()方法删除重复数据
14. concat()函数可以沿着一条轴将多个对象进行堆叠
15. 请阅读下面一段程序：

from pandas import Series

import pandas as pd

from numpy import NaN

series\_obj = Series([None, 4, NaN])

pd.isnull(series\_obj)

执行上述程序后，最终输出的结果为（A ）。

0 True

1 False

2 True

0 True

1 True

2 False

0 False

1 True

2 True

0 True

1 True

2 True

1. 下列选项中，可以删除缺失值或空值的是。（C ）
2. isnull()
3. notnull()
4. dropna()
5. fillna()
6. 请阅读下面一段程序：

import numpy as np

import pandas as pd

ser\_obj = pd.Series([4, np.nan, 6, 5, -3, 2])

ser\_obj.sort\_values()

执行上述程序后，最终输出的结果为（A ）。

4 -3.0

5 2.0

0 4.0

3 5.0

2 6.0

1 NaN

1 NaN

2 6.0

3 5.0

0 4.0

5 2.0

4 -3.0

5 2.0

0 4.0

3 5.0

2 6.0

4 -3.0

1 NaN

0 4.0

1 NaN

2 6.0

3 5.0

4 -3.0

5 2.0

# 第5章习题

1. **填空题**
2. 分组聚合的流程分为 拆分 、 应用 、 合并 。
3. 分组键的形式可以有 列名 、 列表或数组 、 字典 、 函数 。
4. transform()方法会对产生的标量值进行 广播 操作。
5. 当对一个DataFrame对象进行分组后会返回一个 DataFrameGroupBy 对象。
6. **判断题**
7. 分组聚合的原理一般分为拆分-应用-合并。（y ）
8. 只要使用groupby()方法分组就会产生一个DataFrameGroupby对象。（y）
9. 使用agg()方法进行聚合运算会对产生的标量值进行广播。（n ）
10. 使用transform()方法进行聚合运算，其结果可以保持与原数据形状相同。（y ）
11. apply()方法可以使用广播功能。（ y ）
12. **选择题**
13. 下列选项中，关于groupby()方法说法不正确的是。（A ）
14. 分组键可以是列表或数组，但长度不需要与待分组轴的长度相同
15. 可以通过DataFrame中的列名的值进行分组
16. 可以使用函数进行分组
17. 可使用series或字典分组

# 第6章习题

1. **填空题**
2. 数据可视化常见的统计图表有 柱状图 、 折线图 、 饼图 。（写出3个即可）
3. 在使用Matplotlib绘制图表时，需要导入 matplotlib.pyplot 模块。
4. 在直方图中一般使用横轴表示 数据的类型 ，用纵轴表示 分布情况 。
5. 在使用Matplotlib绘制柱状图时可以使用pyplot模块中的 bar 函数。
6. **判断题**
7. seaborn是基于Matplotlib核心库。（y ）
8. Figure对象可以划分多个绘图区域，每个绘图区域都是一个Axes对象。（y ）
9. 绘制图表时，可以使用subplot()函数创建多个子图。（ n ）
10. Matplotlib默认支持中文显示。（n ）
11. Matplotlib生成的图表可以保存在本地。（y ）
12. **选择题**
13. 下列选项中，关于Matplotlib库说法不正确是。（A ）
14. Matplotlib是一个Python 3D绘图库
15. 可输出PNG、PDF等格式
16. 渐进、交互的方式实现数据可视化
17. 使用简单
18. 下列选项中，不属于Seaborn库的特点是。（D ）
19. Seaborn是基于Matplotlib的可视化库
20. 基于网格绘制出更加复杂的图像集合
21. 多个内置主题及颜色主题
22. 可以处理大量的数据流
23. 下列函数中，可以绘制散点图的函数是。（B ）
24. hist()
25. scatter()
26. bar()
27. pie()
28. 下列选项中，描述不正确的是。（B ）
29. 箱形图可以提供有关数据分散情况的信息，可以很直观地查看数据的四分位分布
30. 折线图是用直线段将数据连接起来而组成的图形，以折线的方式显式数据的变化
31. 饼图显示一个数据序列中的各项的大小与各项总和的比例
32. 条形图是由一系列高度不等的纵向条纹或线段表示数据分布情况

# 第7章习题

1. **填空题**
2. \_\_\_\_时间序列\_\_\_\_是指多个时间点上形成的数值序列。
3. ARIMA模型是一种用于时间序列\_\_\_\_预测\_\_\_\_的常见统计模型。
4. \_\_\_\_重采样\_\_\_\_是指将时间序列从一个频率转换到另一个频率的处理过程。
5. 在Pandas中，使用\_\_\_\_Period\_\_\_\_类表示一个标准的时间段或时期。
6. **判断题**
7. 最基本的时间序列类型是以时间戳为索引的Series对象。（y ）
8. 如果相同频率的两个Period对象进行数学运算，那么计算结果为它们的单位数量。（y ）
9. 任何类型的Pandas对象都可以进行重采样。（n ）
10. DatetimeIndex是一种用来指代一系列时间戳的索引结构。（y ）
11. **选择题**
12. 下列选项中，不可以用做Pandas对象索引的是（A ）。
13. Period
14. DatetimeIndex
15. PeriodIndex
16. MultiIndex
17. 请阅读下面一段程序：

import pandas as pd

period1 = pd.Period('2015/6/1')

print(period1 + 50)

执行上述程序，最终输出的结果为（D ）。

1. 2015-07-18
2. 2015-07-19
3. 2015-07-20
4. 2015-07-21
5. 请阅读下面一段程序：

import pandas as pd

import numpy as np

date\_index = pd.date\_range(start="2018/08/10", periods=5)

date\_se = pd.Series(np.arange(5), index=date\_index)

sorted\_se = date\_se.sort\_index()

print(sorted\_se.truncate(after='2018-8-11'))

执行上述程序，最终输出的结果为（D ）。

2018-08-12 2

2018-08-13 3

2018-08-12 2

2018-08-13 3

2018-08-14 4

2018-08-10 0

2018-08-11 1

2018-08-12 2

2018-08-10 0

2018-08-11 1

1. 下列函数中，用于创建固定频率DatetimeIndex对象的是（B ）。
2. shift()
3. date\_range()
4. period\_range()
5. asfreq()