《python数据分析与挖掘实战》笔记第3章

**文章目录**

* + [第3章：数据探索](https://cloud.tencent.com/developer?from=10680)
    - [3.1、数据质量分析](https://cloud.tencent.com/developer?from=10680)
    - [3.2、数据特征分析](https://cloud.tencent.com/developer?from=10680)
      * [3.2.1、分布分析](https://cloud.tencent.com/developer?from=10680)
      * [3.2.2、对比分析](https://cloud.tencent.com/developer?from=10680)
      * [3.2.3、统计量分析](https://cloud.tencent.com/developer?from=10680)
        + [1.集中趋势度量](https://cloud.tencent.com/developer?from=10680)
        + [2.离中趋势度量](https://cloud.tencent.com/developer?from=10680)
      * [3.2.4、周期性分析](https://cloud.tencent.com/developer?from=10680)
      * [3.2.5、贡献度分析](https://cloud.tencent.com/developer?from=10680)
      * [3.2.6、相关性分析](https://cloud.tencent.com/developer?from=10680)
        + [1. 直接绘制散点图](https://cloud.tencent.com/developer?from=10680)
        + [2. 绘制散点图矩阵](https://cloud.tencent.com/developer?from=10680)
        + [3. 计算相关系数](https://cloud.tencent.com/developer?from=10680)
    - [3.3、python主要数据探索函数](https://cloud.tencent.com/developer?from=10680)
      * [3.3.1、基本统计特征函数](https://cloud.tencent.com/developer?from=10680)
        + [corr()](https://cloud.tencent.com/developer?from=10680)
        + [cov()](https://cloud.tencent.com/developer?from=10680)
        + [skew/kurt](https://cloud.tencent.com/developer?from=10680)
      * [3.3.2、拓展统计特征函数](https://cloud.tencent.com/developer?from=10680)
      * [3.3.3、统计作图函数](https://cloud.tencent.com/developer?from=10680)
        + [(1) plot](https://cloud.tencent.com/developer?from=10680)
        + [(2) pie](https://cloud.tencent.com/developer?from=10680)
        + [(3) hist](https://cloud.tencent.com/developer?from=10680)
        + [(4) boxplot](https://cloud.tencent.com/developer?from=10680)
        + [(5) plot(logx = True) / plot(logy = True)](https://cloud.tencent.com/developer?from=10680)
        + [(6)plot(yerr = error)](https://cloud.tencent.com/developer?from=10680)
    - [3.4、小结](https://cloud.tencent.com/developer?from=10680)

**第3章：数据探索**

根据观测、调查收集到初步的样本数据集后，接下来要考虑的问题是：

* 样本数据集的数量和质量是否满足模型构建的要求？
* 是否出现从未设想过的数据状态？
* 其中有没有什么明显的规律和趋势？
* 各因素之间有什么样的关联性？

**3.1、数据质量分析**

**数据质量分析**是数据挖掘中数据准备过程的重要一环，是数据预处理的前提，也是数据挖掘分析结论有效性和准确性的基础，没有可信的数据，数据挖掘构建的模型将是空中楼阁。

**数据质量分析的主要任务**是检查原始数据中是否存在脏数据，脏数据一般是指不符合要求，以及不能直接进行相应分析的数据。

在常见的数据挖掘工作中，脏数据包括如下内容：

* 缺失值
* 异常值
* 不一致的值
* 重复数据及含有特殊符号（如#、￥、\*）的数据

缺失值的处理分为删除存在缺失值的记录、对可能值进行插补和不处理。

异常值是指样本中的个别值，其数值明显偏离其余的观测值。异常值也称为离群点，异常值的分析也称为离群点的分析。

代码清单3-1，餐饮销售额数据异常值检测代码

#-\*- coding: utf-8 -\*-

import pandas as pd

catering\_sale = '../data/catering\_sale.xls' #餐饮数据

data = pd.read\_excel(catering\_sale, index\_col = u'日期') #读取数据，指定“日期”列为索引列

print(data.describe()) # 查看数据的基本情况

import matplotlib.pyplot as plt #导入图像库

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

plt.figure() #建立图像

p = data.boxplot(return\_type='dict') #画箱线图，直接使用DataFrame的方法

x = p['fliers'][0].get\_xdata() # 'flies'即为异常值的标签

y = p['fliers'][0].get\_ydata()

y.sort() #从小到大排序，该方法直接改变原对象

#用annotate添加注释

#其中有些相近的点，注解会出现重叠，难以看清，需要一些技巧来控制。

#以下参数都是经过调试的，需要具体问题具体调试。

for i in range(len(x)):

if i>0:

plt.annotate(y[i], xy = (x[i],y[i]), xytext=(x[i]+0.05 -0.8/(y[i]-y[i-1]),y[i]))

else:

plt.annotate(y[i], xy = (x[i],y[i]), xytext=(x[i]+0.08,y[i]))

plt.show() #展示箱线图

复制

describe()函数可以查看数据的基本情况

销量

count 200.000000

mean 2755.214700

std 751.029772

min 22.000000

25% 2451.975000

50% 2655.850000

75% 3026.125000

max 9106.440000

复制

其中count是非空值数，通过len(data)可以知道数据的记录为201条，因此缺失值数为1。另外提供的基本参数还有平均值（mean）、标准差（std）、最小值（min）、最大值（max）以及1/4、1/2、3/4分位数（25%、50%、75%）。

* 异常值检测箱型图
* 一致性分析 数据不一致性是指数据的矛盾性、不相容性。直接对不一致的数据进行挖掘，可能会产生与实际相违背的挖掘结果。

**3.2、数据特征分析**

**3.2.1、分布分析**

分布分析能揭示数据的分布特征和分布类型。对于**定量数据**，欲了解其分布形式是对称的还是非对称的，发现某些特大或特小的可疑值，**可通过绘制频率分布表、绘制频率分布直方 图、绘制茎叶图进行直观地分析**；对于**定性分类数据，可用饼图和条形图直观地显示分布情况。**

1. 定量数据的分布分析

对于定量变量而言，选择“组数”和“组宽”是做频率分布分析时最主要的问题，一般 按照以下步骤进行。

1）求极差。 2）决定组距与组数。 3）决定分点。 4）列出频率分布表。 5）绘制频率分布直方图。

遵循的主要原则如下。 1） 各组之间必须是相互排斥的。 2） 各组必须将所有的数据包含在内。 3） 各组的组宽最好相等。

1. 定性数据的分布分析

对于定性变量，**常常根据变量的分类类型来分组**，可以釆用饼图和条形图来描述定性变量的分布。

**3.2.2、对比分析**

对比分析是指把两个相互联系的指标进行比较，从数量上展示和说明研究对象规模的大 小，水平的高低，速度的快慢，以及各种关系是否协调。特别适用于指标间的横纵向比较、 时间序列的比较分析。在对比分析中，选择合适的对比标准是十分关键的步骤，只有选择合 适，才能做出客观的评价，选择不合适，评价可能得出错误的结论。

对比分析主要有以下两种形式。

（1）**绝对数比较** 绝对数比较是利用绝对数进行对比，从而寻找差异的一种方法。

（2） **相对数比较** 相对数比较是由两个有联系的指标对比计算的，用以反映客观现象之间数量联系程度 的综合指标，其数值表现为相对数。由于研究目的和对比基础不同，相对数可以分为以下 几种。

1） **结构相对数**：将同一总体内的部分数值与全部数值对比求得比重，用以说明事物的 性质、结构或质量。如居民食品支出额占消费支出总额比重、产品合格率等。 2） **比例相对数**：将同一总体内不同部分的数值进行对比，表明总体内各部分的比例关 系。如人口性别比例、投资与消费比例等。 3） **比较相对数**：将同一时期两个性质相同的指标数值进行对比，说明同类现象在不同 空间条件下的数量对比关系。如不同地区商品价格对比，不同行业、不同企业间某项指标对 比等。 4） **强度相对数**：将两个性质不同但有一定联系的总量指标进行对比，用以说明现象的 强度、密度和普遍程度。如人均国内生产总值用“元/人”表示，人口密度用“人/平方公 里”表示，也有用百分数或千分数表示的，如人口出生率用％。表示。 5） **计划完成程度相对数**：是某一时期实际完成数与计划数的对比，用以说明计划完成 程度。 6） **动态相对数**：将同一现象在不同时期的指标数值进行对比，用以说明发展方向和变 化的速度。如发展速度、增长速度等。

**3.2.3、统计量分析**

用统计指标对定量数据进行统计描述，常从**集中趋势和离中趋势**两个方面进行分析。

**平均水平的指标**是对个体集中趋势的度量，使用最广泛的是均值和中位数； **反映变异程度的指标**则是对个体离开平均水平的度量，使用较广泛的是标准差（方差）、四分位间距。

**1.集中趋势度量**

（1）均值 均值是所有数据的平均值。

作为一个统计量，均值的主要问题是对极端值很敏感。如果数据中存在极端值或者数据 是偏态分布的，那么均值就不能很好地度量数据的集中趋势。为了消除少数极端值的影响， 可以使用截断均值或者中位数来度量数据的集中趋势。截断均值是去掉高、低极端值之后的平均数。

（2）中位数 中位数是将一组观察值按从小到大的顺序排列，位于中间的那个数。即在全部数据中， 小于和大于中位数的数据个数相等。

⑶众数 众数是指数据集中出现最频繁的值。众数并不经常用来度量定性变量的中心位置，更适 用于定性变量。众数不具有唯一性。当然，众数一般用于离散型变量而非连续型变量。

**2.离中趋势度量**

（1）极差 极差=最大值一最小值

极差对数据集的极端值非常敏感，并且忽略了位于最大值与最小值之间的数据的分布 情况。

(2)标准差 标准差度量数据偏离均值的程度

(3) 变异系数 变异系数度量标准差相对于均值的离中趋势

变异系数主要用来比较两个或多个具有不同单位或不同波动幅度的数据集的离中趋势。

(4 )四分位数间距 四分位数包括上四分位数和下四分位数。将所有数值由小到大排列并分成四等份，处于 第一个分割点位置的数值是下四分位数，处于第二个分割点位置(中间位置)的数值是中位 数，处于第三个分割点位置的数值是上四分位数。

四分位数间距，是上四分位数QU,与下四分位数QL之差，其间包含了全部观察值的一 半。其值越大，说明数据的变异程度越大；反之，说明变异程度越小。

代码清单3-2，餐饮销量数据统计量分析代码

#-\*- coding: utf-8 -\*-

#餐饮销量数据统计量分析

from \_\_future\_\_ import print\_function

import pandas as pd

catering\_sale = '../data/catering\_sale.xls' #餐饮数据

data = pd.read\_excel(catering\_sale, index\_col = u'日期') #读取数据，指定“日期”列为索引列

data = data[(data[u'销量'] > 400)&(data[u'销量'] < 5000)] #过滤异常数据

statistics = data.describe() #保存基本统计量

statistics.loc['range'] = statistics.loc['max']-statistics.loc['min'] #极差

statistics.loc['var'] = statistics.loc['std']/statistics.loc['mean'] #变异系数

statistics.loc['dis'] = statistics.loc['75%']-statistics.loc['25%'] #四分位数间距

print(statistics)

复制

结果

销量

count 195.000000

mean 2744.595385

std 424.739407

min 865.000000

25% 2460.600000

50% 2655.900000

75% 3023.200000

max 4065.200000

range 3200.200000

var 0.154755

dis 562.600000

复制

**3.2.4、周期性分析**

**周期性分析**是探索某个变量是否随着时间变化而呈现出某种周期变化趋势。时间尺度相对较长的周期性趋势有年度周期性趋势、季节性周期趋势，相对较短的有月度周期性趋势、 周度周期性趋势，甚至更短的天、小时周期性趋势。

**3.2.5、贡献度分析**

贡献度分析又称帕累托分析，它的原理是帕累托法则，又称20/80定律。同样的投入放 在不同的地方会产生不同的效益。例如，对一个公司来讲，80%的利润常常来自于20%最畅 销的产品，而其他80%的产品只产生了 20%的利润。

代码清单3-3，菜品盈利帕累托图代码

#-\*- coding: utf-8 -\*-

#菜品盈利数据 帕累托图

from \_\_future\_\_ import print\_function

import pandas as pd

#初始化参数

dish\_profit = '../data/catering\_dish\_profit.xls' #餐饮菜品盈利数据

data = pd.read\_excel(dish\_profit, index\_col = u'菜品名')

data = data[u'盈利'].copy()

data.sort\_index(ascending = False)

import matplotlib.pyplot as plt #导入图像库

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

plt.figure()

data.plot(kind='bar')

plt.ylabel(u'盈利（元）')

p = 1.0\*data.cumsum()/data.sum()

p.plot(color = 'r', secondary\_y = True, style = '-o',linewidth = 2)

plt.annotate(format(p[6], '.4%'), xy = (6, p[6]), xytext=(6\*0.9, p[6]\*0.9), arrowprops=dict(arrowstyle="->", connectionstyle="arc3,rad=.2")) #添加注释，即85%处的标记。这里包括了指定箭头样式。

plt.ylabel(u'盈利（比例）')

plt.show()

复制

* 效果图

**3.2.6、相关性分析**

分析连续变量之间线性相关程度的强弱，并用适当的统计指标表示出来的过程称为**相关分析**。

**1. 直接绘制散点图**

判断两个变量是否具有线性相关关系的最直观的方法是直接绘制散点图，如图3-11所示。

**2. 绘制散点图矩阵**

需要同时考察多个变量间的相关关系时，一一绘制它们间的简单散点图是十分麻烦的。 此时可利用散点图矩阵同时绘制各变量间的散点图，从而快速发现多个变量间的主要相关 性，这在进行多元线性回归时显得尤为重要。

散点图矩阵如图3-12所示。

**3. 计算相关系数**

为了更加准确地描述变量之间的线性相关程度，可以通过计算相关系数来进行相关分析。在二元变量的相关分析过程中比较常用的有Pearson相关系数、Spearman秩相关系数和判定系数。

(1 ) Pearson相关系数

一般用于分析两个连续性变量之间的关系，其计算公式如下。

(2 ) Spearman秩相关系数 Pearson线性相关系数要求连续变量的取值服从正态分布。不服从正态分布的变量、分类或等级变量之间的关联性可采用Spearman秩相关系数，也称等级相关系数来描述。

因为一个变量的相同的取值必须有相同的秩次，所以在计算中采用的秩次是排序后所在位置的平均值。

只要两个变量具有严格单调的函数关系，那么它们就是完全Spearman相关的，这与Pearson 相关不同，Pearson相关只有在变量具有线性关系时才是完全相关的。

(3 )判定系数 判定系数是相关系数的平方，用r的平方表示；用来衡量回归方程对y的解释程度。判定系数取值范围：0=<r的平方<=1。r的平方越接近于1,表明x与y之间的相关性越强；r的平方越接近于0,表明两个变量之间几乎没有直线相关关系。

代码清单3-4，餐饮销量数据相关性分析

#-\*- coding: utf-8 -\*-

#餐饮销量数据相关性分析

from \_\_future\_\_ import print\_function

import pandas as pd

catering\_sale = '../data/catering\_sale\_all.xls' #餐饮数据，含有其他属性

data = pd.read\_excel(catering\_sale, index\_col = u'日期') #读取数据，指定“日期”列为索引列

data.corr() #相关系数矩阵，即给出了任意两款菜式之间的相关系数

result1=data.corr()[u'百合酱蒸凤爪'] #只显示“百合酱蒸凤爪”与其他菜式的相关系数

print(result1)

print('-'\*50)

result2=data[u'百合酱蒸凤爪'].corr(data[u'翡翠蒸香茜饺']) #计算“百合酱蒸凤爪”与“翡翠蒸香茜饺”的相关系数

print(result2)

复制

* 结果

百合酱蒸凤爪 1.000000

翡翠蒸香茜饺 0.009206

金银蒜汁蒸排骨 0.016799

乐膳真味鸡 0.455638

蜜汁焗餐包 0.098085

生炒菜心 0.308496

铁板酸菜豆腐 0.204898

香煎韭菜饺 0.127448

香煎罗卜糕 -0.090276

原汁原味菜心 0.428316

Name: 百合酱蒸凤爪, dtype: float64

--------------------------------------------------

0.009205803051836475

复制

**3.3、python主要数据探索函数**

Python中用于数据探索的库主要是Pandas （数据分析）和Matplotlib （[数据可视化](https://cloud.tencent.com/product/yuntu?from=10680)）。其 中，Pandas提供了大量的与数据探索相关的函数，这些数据探索函数可大致分为**统计特征函数与统计作图函数**，而作图函数依赖于Matplotlib，所以往往又会跟Matplotlib结合在一起使用。

**3.3.1、基本统计特征函数**

统计特征函数用于计算数据的**均值、方差、标准差、分位数、相关系数和协方差等**，这些统计特征能反映出数据的整体分布。本小节所介绍的统计特征函数如表3-8所示，它们主要作为Pandas的对象DataFrame或Series的方法岀现。

* 表3-8 Pandas主要统计特征函数

| 方法名 | 函数功能 | 所属库 |
| --- | --- | --- |
| sum() | 计算数据样本的总和(按列计算) | Pandas |
| mean() | 计算数据样本的算术平均数 | Pandas |
| var() | 计算数据样本的方差 | Pandas |
| std() | 计算数据样本的标准差 | Pandas |
| corr() | 计算数据样本的Spearman (Pearson)相关系数矩阵 | Pandas |
| cov() | 计算数据样本的协方差矩阵 | Pandas |
| skew() | 样本值的偏度(三阶矩) | Pandas |
| kurt() | 样本值的峰度(四阶矩) | Pandas |
| describe() | 给出样本的基本描述(基本统计量如均值、标准差等) | Pandas |

**corr()**

功能：计算数据样本的Spearman (Pearson)相关系数矩阵。 使用格式：D.corr(method-pearson')

样本D可为DataFrame,返回相关系数矩阵，method参数为计算方法，支持pearson (皮尔森相关系数，默认选项)、kendall (肯德尔系数)、spearman (斯皮尔曼系数)；

Sl.corr(S2, method-pearson')， SI、S2 均为 Series,这种格式指定计算两个 Series 之间的相关系数。

实例：计算两个列向量的相关系数，釆用Spearman方法。

代码清单3-5，计算两个列向量的相关系数

# -\*- coding:utf-8 -\*-

# 釆用Spearman方法计算两个列向量的相关系数

import pandas as pd

D = pd.DataFrame ([range (1, 8) , range (2, 9) ] ) #生成样本D, —行为 1~7, —行为2~8

D.corr (method= 'pearson') #计算相关系数矩阵

S1 = D.loc[0] # 提取第一行

S2 = D.loc[1] # 提取第二行

result=S1.corr (S2, method= 'pearson') #计算SI、S2的相关系数

print(result)

复制

**cov()**

功能：计算数据样本的协方差矩阵。 使用格式：D.cov()

样本D可为DataFrame,返回协方差矩阵； Sl.cov(S2)， SI、S2均为Series，这种格式指定计算两个Series之间的协方差。

实例：计算6x5随机矩阵的协方差矩阵。

代码清单3-6，计算6x5随机矩阵的协方差矩阵

# -\*- coding:utf-8 -\*-

# 计算6x5随机矩阵的协方差矩阵

import pandas as pd

import numpy as np

D = pd.DataFrame (np.random.randn(6, 5)) #产生6X5随机矩阵

D.cov() #计算协方差矩阵

result=D[0].cov(D[1]) #计算第一列和第二列的协方差

print(result)

复制

**skew/kurt**

功能：计算数据样本的偏度(三阶矩)/峰度(四阶矩)。 使用格式：D.skew() / D.kurt()

计算样本D的偏度(三阶矩)/峰度(四阶矩)。样本D可为DataFrame或Series。

实例：计算6x5随机矩阵的偏度(三阶矩)/峰度(四阶矩)。

代码清单3-7，计算6x5随机矩阵的偏度(三阶矩)/峰度(四阶矩)

# -\*- coding:utf-8 -\*-

# 计算6x5随机矩阵的偏度(三阶矩)/峰度(四阶矩)

import pandas as pd

import numpy as np

D = pd.DataFrame (np.random.randn(6,5)) #产生6x5随机矩阵

result1=D.skew()

print(result1)

print('-'\*50)

result2=D.kurt()

print(result2)

复制

* 结果

0 -0.108415

1 -0.607820

2 0.280084

3 2.217468

4 -0.629157

dtype: float64

--------------------------------------------------

0 -0.084542

1 -1.142215

2 0.906740

3 5.187245

4 0.397617

dtype: float64

复制

**3.3.2、拓展统计特征函数**

除了上述基本的统计特征外，Pandas还提供了一些非常方便实用的**计算统计特征的函数, 主要有累积计算(cum)和滚动计算(pd.rolling\_)**,见表3-8和表3-9。

**表3-9 Pandas累积统计特征函数**

| 方法名 | 函数功能 | 所属库 |
| --- | --- | --- |
| cumsum() | 依次给出前1、2、…、n个数的和 | Pandas |
| cumprod() | 依次给出前1、2、…、n个数的积 | Pandas |
| cummax() | 依次给出前1、2、…、n个数的最大值 | Pandas |
| cummin() | 依次给出前1、2、…、n个数的最小值 | Pandas |

**表3-10 Pandas累积统计特征函数**

| 方法名 | 函数功能 | 所属库 |
| --- | --- | --- |
| rolling\_sum() | 计算数据样本的总和（按列计算） | Pandas |
| rolling\_mean() | 数据样本的算术平均数 | Pandas |
| rolling\_var() | 计算数据样本的方差 | Pandas |
| rolling\_std() | 计算数据样本的标准差 | Pandas |
| rolling\_corr() | 计算数据样本的Spearman （Pearson）相关系数矩阵 | Pandas |
| rolling\_cov() | 计算数据样本的协方差矩阵 | Pandas |
| rolling\_skew() | 样本值的偏度（三阶矩） | Pandas |
| rolling\_kurt() | 样本值的峰度（四阶矩） | Pandas |

其中，cum系列函数是作为DataFrame或’Series对象的方法而出现的，因此命令格式为 D.cumsum(),而rolling\_系列是pandas的函数，不是DataFrame或Series对象的方法，因此，它们的使用格式为pd.rolling\_mean(D, k),意思是每k列计算一次均值，滚动计算。

实例：

D=pd.Series (range (0, 20) ) #构造Series,内容为0~19共20个整数

D.cumsum () #给出前n项和

pd.rolling\_sum(D, 2) #依次对相邻两项求和

复制

**3.3.3、统计作图函数**

通过统计作图函数绘制的图表可以直观地反映出数据及统计量的性质及其内在规律，如 盒图可以表示多个样本的均值，误差条形图能同时显示下限误差和上限误差，最小二乘拟合曲线图能分析两变量间的关系。 **表3-11 Python主要统计作图函数**

| 作图函数名 | 作图函数功能 | 所属工具箱 |
| --- | --- | --- |
| plot() | 绘制线性二维图，折线图 | Matplotlib/Pandas |
| pie() | 绘制饼型图 | Matplotlib/Pandas |
| hist() | 绘制二维条形直方图，可显示数据的分配情形 | Matplotlib/Pandas |
| boxplot() | 绘制样本数据的箱形图 | Pandas |
| plot(logy = True) | 绘制y轴的对数图形 | Pandas |
| plot(yen = error) | 绘制误差条形图 | Pandas |

在作图之前，通常要加载以下代码。

import matplotlib.pyplot as plt #导入作图库

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

plt.figure(figsize = (7, 5)) #创建图像区域，指定比例

复制

作图完成后，一般通过plt.show()来显示作图结果。

**(1) plot**

功能：绘制线性二维图、折线图。 使用格式：

* plt.plot(x, y, S)

这是Matplotlib通用的绘图方式，绘制对于x (即以x为横轴的二维图形)，字符串参量S指定绘制时图形的类型、样式和颜色，常用的选项有：'b’为蓝色、'r’为红色、'g’为绿色、‘o’为圆圈、’+‘为加号标记、’-‘为实线、’–'为虚线。当x、y均为实数同维向量时， 则描出点(x(i),y(f)),然后用直线依次相连。

* D.plot(kind = ‘box’)

这里使用的是DataFrame或Series对象内置的方法作图，默认以Index为横坐标，每列数据为纵坐标自动作图，通过kind参数指定作图类型，支持line(线)、bar(条形)、barh、hist(直方图)、box (箱线图)、kde (密度图)和area、pie (饼图)等，同时也能够接受plt.plot()中接受的参数。因此，如果数据已经被加载为Pandas中的对象，那么以这种方式作图是比较简 洁的。

实例：在区间(0=<x<=2π)绘制一条蓝色的正弦虚线，并在每个坐标点标上五角星。

代码清单3-8，绘制一条蓝色的正弦虚线

# -\*- coding:utf-8 -\*-

# 在区间(0=<x<=2π)绘制一条蓝色的正弦虚线，并在每个坐标点标上五角星

import numpy as np

import matplotlib.pyplot as plt #导入作图库

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

plt.figure(figsize = (7, 5)) #创建图像区域，指定比例

x = np.linspace(0,2\*np.pi, 50) #x坐标输入

y = np.sin(x) #计算对应x的正弦值

plt.plot (x, y,'bp--') #控制图形格式为蓝色带星虚线,，显示正弦曲线

plt.show()

复制

绘制图形如图3-13所示。

**(2) pie**

功能：绘制饼型图。 使用格式：plt.pie(size) 使用Matplotlib绘制饼图，其中size是一个列表，记录各个扇形的比例。pie有丰富的参 数，详情请参考下面的实例。

实例：通过向量［15,30,45,10］画饼图，注上标签，并将第2部分分离出来。绘制结果

代码清单3-9，画饼图

# -\*- coding:utf-8 -\*-

# 通过向量［15,30,45,10］画饼图，注上标签，并将第2部分分离出来。

import matplotlib.pyplot as pit

# The slices will be ordered and plotted counter-clockwise.

labels = 'Frogs', 'Hogs', 'Dogs', 'Logs' #定义标签

sizes = [15, 30, 45, 10] # 每一块的比例

colors = ['yellowgreen','gold','lightskyblue','lightcoral'] #每一块的颜色

explode = (0, 0.1, 0, 0) #突出显示，这里仅仅突出显示第二块(即'Hogs')

pit.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%',shadow=True, startangle=90)

pit.axis ('equal') #显示为圆(避免比例压缩为椭圆)

pit.show()

复制

**(3) hist**

功能：绘制二维条形直方图，可显示数据的分布情形。  使用格式：Plt.hist(x, y) 其中，x是待绘制直方图的一维数组，y可以是整数，表示均匀分为n组；也可以是列表, 列表各个数字为分组的边界点(即手动指定分界点)。

实例：绘制二维条形直方图，随机生成有1000个元素的服从正态分布的数组，分成10组绘制直方图。绘制结果如图3-15所示。

代码清单3-10，二维条形直方图

# -\*- coding:utf-8 -\*-

import matplotlib.pyplot as plt

import numpy as np

x = np.random.randn(1000) #1000个服从正态分布的随机数

plt.hist (x, 10) #分成10组进行绘制直方图

plt.show()

复制

**(4) boxplot**

功能：绘制样本数据的箱形图。 使用格式：D.boxplot() / D.plot(kind = ‘box’)

有两种比较简单的方式绘制D的箱形图，其中一种是直接调用DataFrame的boxplot() 方法；另外一种是调用Series或者DataFrame的plot()方法，并用kind参数指定箱形图 (box)。其中，盒子的上、下四分位数和中值处有一条线段。箱形末端延伸出去的直线称为**须**, 表示盒外数据的长度。如果在须外没有数据，则在须的底部有一点，点的颜色与须的颜色相同。

实例：绘制样本数据的箱形图，样本由两组正态分布的随机数据组成。其中，一组数据均值为0,标准差为1,另一组数据均值为1,标准差为1。绘制结果如图3-16所示。

代码清单3-11，箱形图

# -\*- coding:utf-8 -\*-

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

x = np.random.randn(1000) #1000个服从正态分布的随机数

D = pd.DataFrame([x, x+1]).T #构造两列的DataFrame

D.plot(kind = 'box') #调用Series内置的作图方法画图，用kind参数指定箱形图box

plt.show()

复制

**(5) plot(logx = True) / plot(logy = True)**

功能：绘制x或y轴的对数图形。 使用格式：D.plot(logx = True) / D.plot(logy = True)

对x轴(y轴)使用对数刻度(以10为底)，y轴(X轴)使用线性刻度，进行plot函数绘图，D 为 Pandas 的 DataFrame 或者 Series。

实例：构造指数函数数据使用plot(logy = True)函数进行绘图，绘制结果如图3-17 所示。

代码清单3-12，指数函数plot图

# -\*- coding:utf-8 -\*-

import matplotlib.pyplot as plt #导入作图库

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

import numpy as np

import pandas as pd

x = pd.Series(np.exp(np.arange(20))) #原始数据

x.plot(label = '原始数据图',legend = True)

plt.show()

x.plot (logy = True, label = '对数数据图',legend = True)

plt.show()

复制

**(6)plot(yerr = error)**

功能：绘制误差条形图。 使用格式：D.plot(yerr = error)

绘制误差条形图。D为Pandas的DataFrame或Series,代表着均值数据列，而error则 是误差列，此命令在y轴方向画出误差棒图；类似地，如果设置参数xerr = error,则在x轴 方向画出误差棒图。

实例：绘制误差棒图。绘制结果如图3-18所示。

代码清单3-13，绘制误差棒图

# -\*- coding:utf-8 -\*-

import matplotlib.pyplot as plt #导入作图库

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

import numpy as np

import pandas as pd

error = np.random.randn (10) #定义误差列

y = pd.Series(np.sin(np.arange(10))) #均值数据列

y.plot(yerr = error) #绘制误差图

plt.show()

复制

**3.4、小结**

本章从应用的角度出发，从数据质量分析和数据特征分析两个方面对数据进行探索分析，最后介绍了 Python常用的数据探索函数及用例。数据质量分析要求我们拿到数据后先检测是否存在缺失值和异常值；数据特征分析要求我们在数据挖掘建模前，通过频率分布分析、 对比分析、帕累托分析、周期性分析、相关性分析等方法，对采集的样本数据的特征规律进 行分析，以了解数据的规律和趋势，为数据挖掘的后续环节提供支持。

本文参与[腾讯云自媒体分享计划](https://cloud.tencent.com/developer/support-plan) ，欢迎热爱写作的你一起参与！

本文分享自作者个人站点/博客：https://blog.csdn.net/xc\_zhou/[复制](javascript:void(0);)

如有侵权，请联系 cloudcommunity@tencent.com 删除。

[展开阅读全文](javascript:;)

[编程算法](https://cloud.tencent.com/developer/tag/10663?entry=article)[数据挖掘](https://cloud.tencent.com/developer/tag/10148?entry=article)[Python](https://cloud.tencent.com/developer/tag/10169?entry=article)[MATLAB](https://cloud.tencent.com/developer/tag/10294?entry=article)

[举报](javascript:;)

点赞 2分享

相关文章

* 《python数据分析与挖掘实战》笔记第5章

分类和预测是预测问题的两种主要类型，分类主要是预测分类标号（离散属性），而预测 主要是建立连续值函数模型，预测给定自变量对应的因变量的值。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第4章

数据预处理一方面是要提高数据的质量，另一方面是要让 数据更好地适应特定的挖掘技术或工具。统计发现，在数据挖掘的过程中，数据预处理工作量占到了整个过程的60%。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第2章

python有4个内建的数据结构–list（列表）、tuple（元组）、dictionary（字典）以及set（集合），它们可以统称为容器。

[周小董](https://cloud.tencent.com/developer/user/4908043)

* 《python数据分析与挖掘实战》笔记第1章

数据挖掘是从大量数据（包括文本）中挖掘出隐含的、先前未知的、对决策有潜在价值的关系、模式和趋势，并用这些知识和规则建立用于决策支持的模型，提供预测性决策支持的方...

[周小董](https://cloud.tencent.com/developer/user/4908043)

* Python数据分析与实战挖掘

基础篇 书推荐：《用python做科学计算》 ? 扩展库 简介 Numpy数组支持，以及相应的高效处理函数 Scipy矩阵支持，以及相应的矩阵数值计算模块 Ma...

[企鹅号小编](https://cloud.tencent.com/developer/user/1008345)

* Python数据分析与挖掘实战.pdf

[学一学大数据](https://cloud.tencent.com/developer/user/1685822)

* 关于《Python数据挖掘入门与实战》读书笔记二（亲和性分析）

首先确保最新的版本吧，python最让人头痛和无解的就是版本兼容性问题，不过既然选择了就不要怕麻烦，不断的纠错不断的尝试，在频繁的错误中提高自己。

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* 关于《Python数据挖掘入门与实战》读书笔记七（主成分分析二）

主成分分析算法（Principal Component Analysis，PCA）的目的是找到能用较少信息描述数据集的特征组合。它意在发现彼此之间没有相关性、能...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* Python业务分析实战｜共享单车数据挖掘

自行车共享系统是传统自行车租赁的新一代，从注册会员、租赁到归还的整个过程都是自动化的。通过这些系统，用户可以很容易地从一个特定的位置租用自行车，并在另一个位置归...

[数据STUDIO](https://cloud.tencent.com/developer/user/8756457)

* 关于《Python数据挖掘入门与实战》读书笔记六（主成分分析一）

特征抽取是数据挖掘任务最为重要的一个环节，一般而言，它对最终结果的影响要高过数据挖掘算法本身。但怎样选取好的特征，还没有严格、快捷的规则可循，这也是数据挖掘科学...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* Python数据分析与挖掘实战 (大数据技术丛书) - 张良均 等著

10余位数据挖掘领域资深专家和科研人员，10余年大数据挖掘咨询与实施经验结晶。从数据挖掘的应用出发，以电力、航空、医疗、互联网、生产制造以及公共服务等行业真实案...

[用户3157710](https://cloud.tencent.com/developer/user/3157710)

* Python 爬取淘宝商品数据挖掘分析实战

? 作者 孙方辉 本文为CDA志愿者投稿作品，转载需授权 项目内容 本案例选择>> 商品类目：沙发； 数量：共100页 4400个商品； 筛选条件：天猫...

[CDA数据分析师](https://cloud.tencent.com/developer/user/1283928)

* 关于《Python数据挖掘入门与实战》读书笔记一（数据挖掘、机器学习、深度学习、人工智能）

Python数据挖掘、Python机器学习、Python深度学习的书籍买了不少本了，但真正读下来的却很少，为何？涉及高等数学，图论、离散数学等等，对我来说和天文...

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)

* 慕课网数据分析与挖掘实战（一）-数据获取

其中，urllib，urllib2，requests，scrapy主要用于数据抓取 xpath主要用于分析提取有价值的内容 phantomjs，beauti...

[Meet相识](https://cloud.tencent.com/developer/user/3090887)

* 《数据挖掘与数据化运营实战 思路、方法、技巧与应用》—— 读书笔记

[用户1154259](https://cloud.tencent.com/developer/user/1154259)

* 关于《Python数据挖掘入门与实战》读书笔记三（估计器和近邻算法）

Scikit-learn库，实现了一系列数据挖掘算法，提供通用编程接口、标准化的测试和调参工具，便于用户尝试不同算法对其进行充分测试和查找优参数值。

[python与大数据分析](https://cloud.tencent.com/developer/user/7190337)