第 3 章

认 识 数 据

数据挖掘，先有数据，然后有挖掘，认识数据是数据挖掘的前提。在现实世界中，数据一般都是有缺失的、异构的、有量纲的。认识数据，不仅要了解数据的属性（维）、类型和量纲，还要了解数据的分布特性。洞察数据的特征，检验数据的质量，有助于后续的挖掘工作，否则，没有高质量的数据，数据挖掘的结果将是空中楼阁。

3.1 数据类型

3.1.1 属性类型

数据集是数据对象的集合，一个数据对象代表一个实体。例如，在销售数据集中，数据对象可以是购买商品的顾客、销售的一个个商品或一次交易记录。在医疗数据集中，数据对象可以是医生、患者或药品。通常，数据对象用属性来描述，属性用来表示数据对象的一个特征。数据对象又称为样本、实例或数据点。在数据库中，一个数据对象对应数据库的一行，即一条记录、一个数据元组，是字段的集合。

通常，属性、字段、维、特征和变量可以互换使用。在不同的应用场景中，对属性有不同的称呼，如在数据仓库中，称为维；在机器学习中，称为特征；在统计学中，称为变量。在数据挖掘中，更偏向于使用属性来描述一个对象。例如，描述学生对象的属性可能包括姓名、性别、年龄和家庭地址。用来描述一个给定对象的属性组（集合）称为属性向量（特征向量）。一个属性的类型由该属性可能具有的值的集合决定，属性可分为标称属性、二元属性、序数属性、数值属性、离散属性与连续属性。图 3-1 为是否打羽毛球的样本数据集。

属性

序号	天气状况	气温	湿度	风力	是否玩
1	晴天	热	高	低	否
2	晴天	热	高	高	否
3	阴天	热	高	低	是
4	下雨	适宜	高	低	是
5	下雨	冷	正常	低	是

对象

图 3-1　是否打羽毛球的样本数据集

下面给出数据对象属性的常见类型。

1. 标称属性

标称属性的值是事物的标号或事物的名称,每个值表示类别、编码、状态,因此标称属性又被称为分类。标称属性的值没有次序信息,在计算机领域,也可以称为枚举型。假设头发颜色是描述人的一个属性,在现实生活中头发颜色的可能取值为黑色、棕色、淡黄色、红色、灰色、白色等类别。头发颜色就是一个标称属性。

尽管我们说标称属性的值是一些符号或“事物的名称”,但是可以用数字表示这些符号或名称。例如对于头发颜色,可以指定数字 0 表示黑色、1 表示棕色、2 表示灰色,等等。与一个商品的价格减去另一个商品的价格(得到两个商品的差价)不同,用表示灰色的数字 2 减去表示棕色的 1 是毫无意义的。

尽管一个标称属性可以取整数值,但标称属性值并不具有有意义的序,获取这种属性的均值(平均值)或中位数(中值)没有任何意义。

2. 二元属性

二元属性是一种标称属性,只有两个类别或状态:0 或 1,其中 0 通常表示该属性不出现,而 1 表示该属性出现。出现、不出现发生的概率可能相同,也可能不相同,因而二元属性分为对称二元属性和非对称二元属性。

一个二元属性是对称的,如果它的两种状态具有同等价值并且携带相同的权重;关于哪个结果应该用 0 或 1 编码并无偏好。这样的例子如抛硬币数字是否向上的属性。

一个二元属性是非对称的,如果其状态的结果不是同样重要的,即出现、不出现发生概率不相同,如卫星发射的成功和不成功结果。

3. 序数属性

序数属性对应的可能的值之间具有有意义的序(也就是对应的值有先后次序),但是相继值之间的差是未知的。

通常是通过把数值量的值域离散化成有限个有序类别的方式得到序数属性,如服务满意度问卷调查中,顾客的满意度可以是:0 代表非常不满意,1 代表不满意,2 代表中性,3 代表满意,4 代表很满意。序数属性被用来衡量无法客观度量的属性。

4. 数值属性

数值属性是定量的,是可测量的数值,为整数或实数,如一个人的身高、体重、收入等属性就是数值属性。

5. 离散属性与连续属性

离散属性具有有限个数的取值或无限可数的取值,如描述人的年龄、工号属性就是两个离散属性,二元属性可看作是一种特殊的离散属性。

如果属性不是离散的,则它是连续的。连续属性为取值为实数的属性,如描述人的温度、重量和高度的属性就是三个连续属性。

3.1.2 数据集的类型

数据集的类型是从集合整体上分析数据对象的类型。本书从数据对象之间的结构关系角度进行分类,比较常见的有记录数据、有序数据和图形数据。

1. 记录数据

许多数据挖掘任务都是假定数据集是记录(数据对象)的汇集,每个记录包含固定的数据字段(属性)集。记录数据最常见的类型是事务数据(transaction data)。

事务数据是一种特殊类型的记录数据,其中每个记录(数据)涉及一系列的项。考虑顾客一次购物所买的商品集合构成一个事务,而所有购买的商品作为项。

2. 有序数据

数据对象之间存在时间或空间上的顺序关系,称为有序数据。

1) 时序数据

时序数据也称时间数据,可以看作记录数据的扩充,其中每一个记录包含一个与之相关联的时间。时间也可以与每个属性相关,如每个记录可以是一位顾客的购物历史,包含不同时间购买的商品列表。使用这些信息,可能发现:买了苹果手机的人,是不会再关注那些低端的 Android 手机的。

时序数据一般是由硬件设备或系统监控软件连续采集形成的数据。例如股票价格波动信息,医疗仪器监视病人的心跳、血压、呼吸数值,环境传感器连续记录的温度、湿度数值等。

2) 序列数据

序列数据是一个数据集合,如词或字母的序列、基因组序列等。

3) 时间序列数据

时间序列数据是一种特殊的时序数据,其中每个记录都是一个时间序列,即一段时间以来的测量序列。需要注意的是:在分析时间序列数据时,需要考虑时间自相关,即如果两个测量的时间很近,则这些测量的值通常非常相似。

4) 空间有序数据

某些数据也许还会拥有空间属性,如位置或区域。空间有序数据的例子有很多,例如从不同地方收集气象数据。空间有序数据的一个重要的特点就是空间自相关性,即物理上靠近的对象趋向于其他方面也相似。

3. 图形数据

如果数据对象之间有一定的依赖关系,这些依赖关系构成图形或网状结构,把这种数据集称为图形数据。

考虑互联网中相互链接的网页,可以把网页看作图中的结点,把它们之间的连接看作成图中的边,搜索引擎就是不断沿着网页中的超链接进行搜索的。类似的还有社交网络图形数据,社交用户可以看作结点,用户之间的联系看作边。

3.2　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环节,是数据挖掘(也称数据分析,本书将两个概念等同)结论有效性和准确性的基础,没有高质量数据,数据挖掘构建的模型将是空中楼阁。数据质量通常是指数据值的质量,包括准确性、完整性和一致性。数据的准确性是指数据不包含错误或异常值,数据的完整性是指数据不包含缺失值,数据的一致性是数据在各个数据源中都是相同的。

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,脏数据会降低数据的质

量,影响数据挖掘的结果。脏数据一般是指不符合要求,以及不能直接进行数据挖掘的数据,具体包括缺失值、异常值、不一致的值、重复数据及含有特殊符号(如#、¥、*)的数据。在进行数据挖掘之前,需要对脏数据进行处理,以提高数据的质量。

3.2.1 缺失值分析

数据的缺失主要包括记录的缺失和记录中某个字段信息的缺失,两者都会造成分析结果的不准确。造成数据缺失的原因是多方面的,主要有以下几种。

(1) 有些信息暂时无法获取。例如在医疗数据库中,并非所有病人的所有临床检验结果都能在给定的时间内得到,就致使一部分属性值空缺出来。

(2) 有些信息是被遗漏的。可能是因为输入时认为不重要、忘记填写了或对数据理解错误而遗漏,也可能是由于数据采集设备的故障、存储介质的故障、传输介质的故障、一些人为因素等原因而丢失了。

(3) 有些对象的某个或某些属性是不可用的。例如未婚者的配偶姓名、儿童的固定收入状况等。

(4) 有些信息(被认为)是不重要的,如数据库的设计者并不在乎某个属性的取值。

(5) 获取这些信息的代价太大。

缺失值的存在,对数据挖掘主要造成了三方面的影响:数据挖掘建模将丢失大量的有用信息;数据挖掘模型所表现出的不确定性更加显著,模型中蕴涵的规律更难把握;包含空值的数据会使数据挖掘建模过程陷入混乱,导致不可靠的输出。对缺失值的处理,要具体问题具体分析,因为属性缺失有时并不意味着数据缺失,缺失本身是包含信息的,所以需要根据不同应用场景下缺失值可能包含的信息,对缺失值进行合理处置。

缺失值分析包括查看含有缺失值的记录和属性,以及包含缺失值的观测总数和缺失率。

```
>>> import numpy as np
>>> import pandas as pd
>>> df = pd.DataFrame([[89, 78, 92, np.nan],[70, 86, 97, np.nan],[80, 90, 85, np.
nan],[92,95,89,np.nan],[np.nan,np.nan,np.nan,np.nan]],columns=list('ABCD'))
>>> df
      A     B     C    D
0  89.0  78.0  92.0  NaN
1  70.0  86.0  97.0  NaN
2  80.0  90.0  85.0  NaN
3  92.0  95.0  89.0  NaN
4   NaN   NaN   NaN  NaN
>>> df.isnull().sum()                    #统计空值情况,可以看出每列都存在缺失值
A    1
B    1
C    1
D    5
dtype: int64
>>> df.isnull().any()                    #查找存在缺失值的列,可以看出每列中都存在缺失值
A    True
```

```
B    True
C    True
D    True
dtype: bool
>>> df.isnull().all()                    #查找均为缺失值的列,发现 D 列全为空
A    False
B    False
C    False
D    True
dtype: bool
>>> nan_lines = df.isnull().any(1)       #查找存在缺失值的行
>>> nan_lines.sum()                      #统计有多少行存在缺失值
5
>>> df[nan_lines]                        #查看有缺失值的行信息
      A     B     C    D
0  89.0  78.0  92.0  NaN
1  70.0  86.0  97.0  NaN
2  80.0  90.0  85.0  NaN
3  92.0  95.0  89.0  NaN
4   NaN   NaN   NaN  NaN
```

3.2.2　异常值分析


异常值分析

异常值分析是检验数据是否有录入错误以及含有不合常理的数据。异常值是指样本中的明显偏离其余观测值的个别值,异常值也称为离群点。忽视异常值的存在是十分危险的,不加剔除地把异常值包括进数据的计算分析过程中,对结果会产生不良影响。重视异常值的出现,分析其产生的原因,常常成为发现问题进而改进决策的契机。异常值的分析也称为离群点分析。

从使用的主要技术路线的不同,可将异常值检测方法分为:基于统计的方法、基于距离的方法、基于偏差的方法、基于箱形图分析的方法、基于密度的方法、基于聚类的方法等。下面介绍基于统计的方法、基于偏差的方法和基于箱形图分析的方法。

1. 基于统计的方法

基于统计的方法是通过为数据创建一个模型,根据数据拟合模型的情况来评估它们,查看哪些数据是不合理的。最常用的统计模型是最大值和最小值模型,用来判断数据集的最大值的最小值的取值是否超出了合理的范围。如人的身高的最大值为 3.3m,则该变量的取值存在异常。再如客户年龄的最大值为 199 岁,则该变量的取值存在异常。

2. 基于偏差的方法

基于偏差的方法的基本思想是通过检查一组数据的主要特性来确定数据是否异常,如果一个数据的特性与给定的描述过分地偏离,则该数据被认为是异常数据。基于偏差的异常值检测方法最常采用的技术是序列异常技术。序列异常技术的核心是要构建一个相异度函数,如果样本数据间的相似度较高,相异度函数的值就比较小;反之,如果样本数据间的相异度越大,相异度函数的值就越大(例如方差就是满足这种要求的函数)。这种方法,多是该

数据服从正态分布，异常值被定义为一组测定值中与平均值的偏差超过三倍标准差的值。在正态分布下，距离平均值 3σ 之外的值出现的概率为 $P(|x-\mu|>3\sigma)\leqslant 0.003$，属于极个别的小概率事件。如果数据不服从正态分布，也可以用远离平均值的多少倍标准差来描述。

3. 基于箱形图分析的方法

箱形图，又称为盒式图、盒状图或箱线图，是一种用作显示一组数据分散情况的统计图，因形状如箱子而得名。一个箱形图举例如图 3-2 所示，其中应用到了分位数的概念。箱形图的绘制方法是：先找出一组数据的中位数、上四分位数、下四分位数、上限、下限；然后，连接两个四分位数画出箱子；中位数在箱子中间。

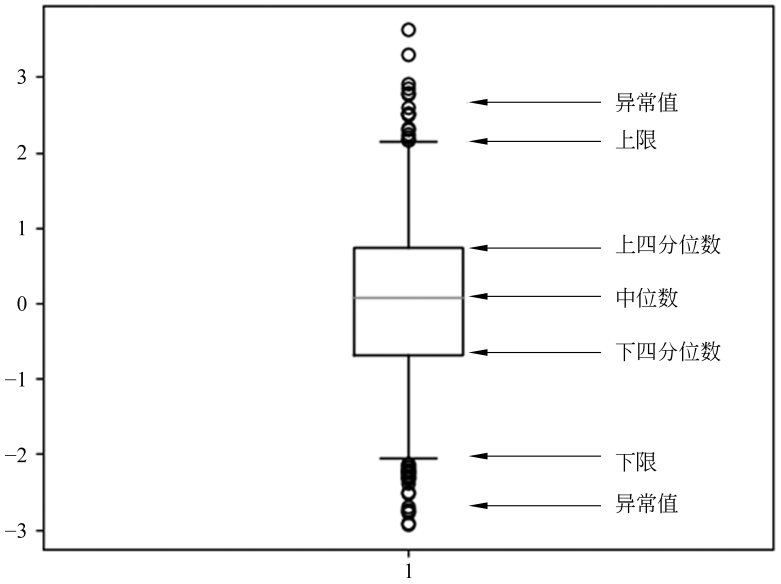


图 3-2　箱形图举例

箱形图提供了识别异常值的一个标准：异常值通常被定义为小于 QL−1.5IQR 或大于 QU+1.5IQR 的值。QL 称为下四分位数，表示全部观察值中有四分之一的数据取值比它小；QU 称为上四分位数，表示全部观察值中有四分之一的数据取值比它大；IQR 称为四分位数间距，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。上限是非异常范围内的最大值(如定义为 QU+1.5IQR)，下限是非异常范围内的最小值(如定义为 QL−1.5IQR)。中位数，即二分之一分位数，计算的方法就是将一组数据按从小到大的顺序，取中间这个数，中位数在箱子中间。

箱形图依据实际数据绘制，没有对数据作任何限制性要求(如服从某种特定的分布形式)，它只是真实直观地表现数据分布的本来面貌；另一方面，箱形图判断异常值的标准以四分位数和四分位距为基础，四分位数具有一定的鲁棒性，多达 25%的数据可以变得任意远而不会很大地扰动四分位数，所以异常值不会影响箱形图的数据形状。由此可见，箱形图识别异常值的结果比较客观，在识别异常值方面有一定的优越性。

```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
'''生成一组正态分布的随机数，数量为 1000，loc 为概率分布的均值，对应着整个分布的中心
```

```
centre;scale 为概率分布的标准差,对应于分布的宽度,scale 越大,分布曲线越矮胖,scale 越小,分布曲线越瘦高;size 为生成的随机数的数量'''
>>> data = np.random.normal(size = (1000, ), loc = 0, scale = 1)
'''whis 默认是 1.5,通过调整它的数值来设置异常值显示的数量,如果想显示尽可能多的异常值,whis 设置为较小的值,否则设置为较大的值'''
>>> plt.boxplot(data, sym ="o", whis = 1)   #绘制箱形图,sym 设置异常值点的形状
>>> plt.show()                              #显示绘制的箱形图,如图 3-3 所示
```

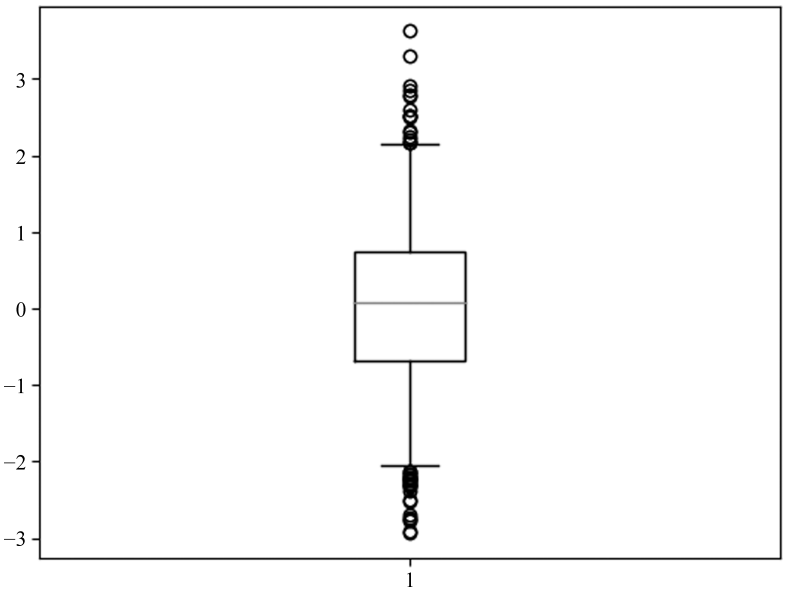


图 3-3　绘制的箱形图

在餐饮系统中的销量额数据可能出现缺失值和异常值,如果数据记录和属性较多,就需要编写程序来检测出缺失值和异常值。在 pandas 中,只需要读入数据,然后使用 describe()函数就可以查看数据的基本情况。

```
>>> import pandas as pd
>>> import matplotlib.pyplot as plt
>>> import matplotlib
>>> matplotlib.rcParams['font.family'] = 'FangSong'          #FangSong 是中文仿宋
>>> matplotlib.rcParams['font.size'] = 15                    #设置字体的大小
#读取数据,指定“日期”列为索引列
>>> data = pd.read_excel('catering_sale.xls',index_col='日期')  #读取餐饮数据
>>> data
日期            销量
2018-03-01    51.0
2018-02-28  2618.2
...            ...
2017-08-31  3494.7
2017-08-30  3691.9
>>> data.boxplot()
```

```
<matplotlib.axes._subplots.AxesSubplot object at 0x000000000E561A58>
>>> plt.show()    #餐饮数据的箱形图,如图 3-4 所示
```

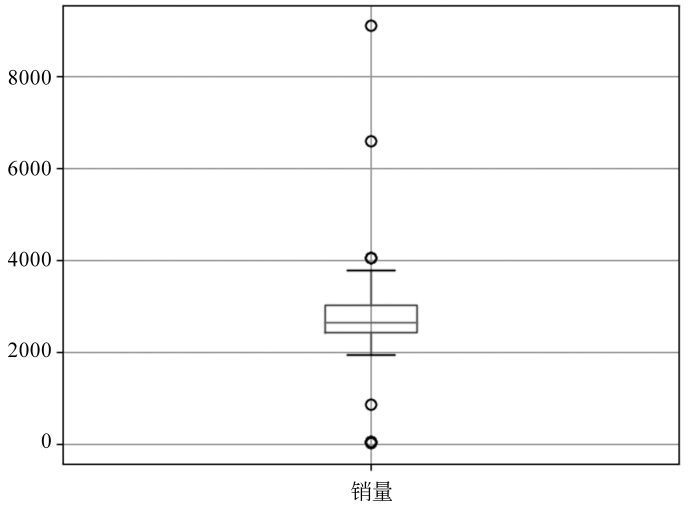


图 3-4　餐饮数据的箱形图

3.2.3　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。

在数据有多份副本的情况下,如果网络、服务器或者软件出现故障,会导致部分副本写入成功,部分副本写入失败。这就造成各个副本之间的数据不一致,数据内容冲突。在数据挖掘过程中,数据不一致主要发生在数据集成的过程中,可能是由于数据来自于不同的数据源、对于重复存放的数据未能进行一致性更新造成的。在关系数据库中,不一致性可能存在于单个元组中、同一关系(表)的不同元组之间、不同关系(表)的元组之间。

例如,两张表中都存储了用户的家庭地址,但在用户的家庭地址发生改变时只更新了一张表中的数据,那么这两张表中就有了不一致的数据。

3.3　数据特征分析

对数据进行质量分析以后,接下来可通过绘制图表、计算某些特征量等手段进行数据的特征分析。主要从分布分析、统计量分析、周期性分析、相关性分析等角度进行数据的特征分析。

3.3.1　分布特征

分布分析用来揭示数据的分布特征和分布类型,显示其分布情况。分布分析主要分为两种:对定量数据的分布分析和对定性数据的分布分析。

1. 定量数据的分布分析

面对大量的数据,人们通常希望知道数据的大致分布情况,这里可使用直方图图像来描

述数据的分布情况。直方图图像由一批长方形构成，通过长方形的面积或高度来代表对应组的数据所占的比例。

直方图有两种类型：当用长方形的面积代表对应组的频数与组距的比时，称为频率分布直方图；当用长方形的高代表对应组的频数时，则称为频数分布直方图。

绘制定量数据的直方图进行分布分析通常按照以下步骤执行。

(1) 求极差。

(2) 决定组距与组数。

(3) 决定分点。

(4) 得到频率分布表。

(5) 绘制频率分布直方图。

绘制定量数据的直方图应遵循的原则有 3 个。

(1) 所有分组必须将所有数据包含在内。

(2) 各组的组宽最好相等。

(3) 各组相斥。

2. 定性数据的分布分析

对于定性数据，通常根据数据的分类类型来分组，可以采用饼图和条形图来描述定性数据的分布。

3.3.2　统计量特征

基本统计量可以用来识别数据的性质，凸显哪些数据应该视为噪声或离群点。基本统计量包括描述数据集中趋势的统计值(平均数、中位数和众数)、描述数据离散程度的统计量(极差、四分位数、平均差、方差、标准差和离散系数)和描述数据分布状况的统计量(偏态系数)。有了这些基本统计量，数据挖掘人员就掌握了数据的基本特征。通过这些基本统计量对数据进行统计分析后，可以基本确定对数据做进一步挖掘的方向。

1. 数据集中趋势的统计值

1) 平均数

平均数又称均值，能反映一组数据的集中趋势，可作为数据代表与另一组数据相比较，以明确两组数据之间的差异状况。平均数的类型包括：算数平均数、加权平均数、几何平均数、调和平均数。统计分析中，算数平均数应用最普遍。

(1) 算术平均数。

全部数据的总和除以数据总个数所得的商称为算术平均数，简称均数(mean)。

利用下式可直接计算算术平均数：

$$\bar{x}=\frac{x_1+x_2+\cdots+x_n}{n}=\frac{1}{n}\sum_{i=1}^{n}x_i$$

上式适用于不分组的小样本数据。下面给出利用 Python 求算术平均数的具体过程。

```
>>> import numpy as np                          #如不特别说明,全文中的 np 均是这个含义
>>> scores = [91, 95, 97, 99, 92, 93, 96, 98]
>>> np.mean(scores)                             #求 scores 的均值
95.125
```

可利用频数分布表计算法求算术平均值的近似值：

$$\bar{x}=\frac{f_1X_1+f_2X_2+\cdots+f_kX_k}{f_1+f_2+\cdots+f_k}=\frac{1}{N}\sum \boldsymbol{fX}$$

其中，f 表示各组频数，X 表示各组组中值，k 表示组数，N 表示总频数。

上式适用于已经编制成频数分布表的分组数据。

(2) 加权平均数。

加权平均数是指具有不同权重数据的平均数。

① 已知权重的加权平均数计算公式。

$$\overline{X}_w=\frac{W_1X_1+W_2X_2+\cdots+W_NX_N}{W_1+W_2+\cdots+W_N}=\frac{\sum \boldsymbol{WX}}{\sum \boldsymbol{W}}$$

其中 W_N 表示各观察值的权重，X_N 表示具有不同权重的观察值。

已知权重的加权平均数计算公式举例：学生期末成绩，其中期中考试占 30%、期末考试占 50%、作业占 20%，假如某人期中考试得了 80、期末 90、作业分 94，如果是算数平均，那么就是：

$$\frac{80+90+94}{3}=88$$

加权平均数就是：

$$\frac{80\times30+90\times50+94\times20}{30+50+20}=87.8$$

② 未知权重的加权平均数计算公式。

$$\overline{X}_t=\frac{N_1\overline{X}_1+N_2\overline{X}_2+\cdots+N_K\overline{X}_K}{N_1+N_2+\cdots+N_K}=\frac{\sum N\overline{X}}{\sum N}$$

其中 N_K 表示各组数据的频数，$\overline{X}_K$ 表示各组数据的平均值。

未知权重的加权平均数计算公式举例如下。

学校期末考试的两个班的“数据挖掘”课程考试情况：一班 50 人，平均 83；二班 60 人，平均 85，加权平均数就是：

$$\frac{50\times83+60\times85}{50+60}=84.1$$

(3) 几何平均数。

几何平均数是指 N 个数据连乘积的 N 次方根，符号为 $\overline{X}_g$。

$$\overline{X}_g=\sqrt[N]{X_1\times X_2\times\cdots\times X_N}=\sqrt[N]{\prod_{i=1}^{N}X_i}$$

几何平均数可用于计算入学人数增加率、学校经费增加率、阅读能力提高率等。

(4) 调和平均数。

调和平均数是指一组数据中每个数据的倒数的算数平均数的倒数，符号为 $\overline{X}_H$。

$$\overline{X}_H=\frac{1}{\frac{1}{N}\left(\frac{1}{X_1}+\frac{1}{X_2}+\cdots+\frac{1}{X_N}\right)}=\frac{N}{\sum_{i=1}^{N}\left(\frac{1}{X_i}\right)}$$

调和平均数可用于计算平均学习速度，如阅读速度、解题速度、识字速度等。

2）中位数

中位数（又称中值，median）代表一个样本、种群或概率分布中的一个数值，其可将数值集合划分为相等的上下两部分。对于有限的数集，可以通过把所有观察值按从高到低排序后找出正中间的一个作为中位数。如果观察值有偶数个，则中位数不唯一，通常取最中间的两个数值的平均数作为中位数。

```
>>> np.median(scores)                    #求 scores 的中位数
95.5
>>> np.median([1,3,4])                   #求 1,3,4 的中位数
3.0
```

3）众数

众数是一组数据中出现次数最多的数值，如数据 1、2、3、3、4 的众数是 3。有时众数在一组数中有好几个，如数据 2、3、－1、2、1、3 中，2、3 都出现了两次，它们都是这组数据中的众数。

用 NumPy 中建立元素出现次数的索引的方法求众数。

```
>>> c=np.bincount([2,3,2,5,1,3,1,1,9])
>>> np.argmax(c)
1
```

2. 数据离散程度的统计量

数据的离散程度即衡量一组数据的分散程度如何，其衡量的标准和方式有很多，包括极差、分位数、四分位数、百分位数和四分位差。五种度量值可以用箱形图统一表示，它对于识别离群点是有用的。方差和标准差也可以标识数据分布的散布程度。此外，离散系数（变异系数）也用来衡量数据离散程度。而具体选择哪一种方式则需要依据实际的数据要求进行选择。

1）极差

极差为数据样本中的最大值与最小值的差值，反映了数据样本的数值范围，是最基本的衡量数据离散程度的方式。如在数学考试中，一个班学生得分的极差为 50，反映了学习最好的学生与学习最差的学生得分差距为 50。

```
>>> import numpy as np
>>> scores1 = [42,45,67,75,38,87,78,89,76, 67,82,89,95,67,92]
>>> print('极差:',np.max(scores1)-np.min(scores1))
极差: 57
```

2）四分位数

把所有数据由小到大排列并分成四等份，处于三个分割点位置的数值就是四分位数。

处于第一个分割点位置的数据称为第一四分位数（Q_1），又称“下四分位数”。

处于第二个分割点位置的数据称为第二四分位数（Q_2），又称“中位数”。

处于第三个分割点位置的数据称为第三四分位数（Q_3），又称“上四分位数”。

四分位差（QD）$=Q_3-Q_1$，即数据样本的上四分位数和下四分位数的差值，四分位差反

映了数据中间部分(占总数据的 50%)数据的离散程度,其数值越小,说明中间的数据越集中;其数值越大,说明中间的数据越分散。四分位差的大小在一定程度上可反映中位数对一组数据的代表程度。下面给出用 Python 扩展库 stats 来实现四分位差的求解。

```
>>> import stats as sts
>>> print('下四分位数:',sts.quantile(scores1,p=0.25))            #求下四分位数
下四分位数: 67
>>> print('上四分位数:',sts.quantile(scores1,p=0.75))            #求上四分位数
上四分位数: 89
>>> sts.quantile(scores1,p=0.75)-sts.quantile(scores1,p=0.25)  #求四分位差
22
```

3) 平均差

平均差是样本所有数据与其算术平均数的差绝对值的算术平均数。平均差反映各个数据与算术平均数之间的平均差异。平均差越大,表明各个数据与算术平均数的差异程度越大,该算术平均数的代表性就越小;平均差越小,表明各个数据与算术平均数的差异程度越小,该算术平均数的代表性就越大。

平均差的计算公式为:

$$\mathrm{MD}=\frac{1}{N}\sum_{i=1}^{N}\mid X_i-\overline{X}\mid$$

其中,X_i 为变量,$\overline{X}$ 为算术平均数,N 为变量值的个数。

4) 方差(variance)与标准差(standard deviation)

方差与标准差都是数据散布度量,它们指出数据分布的散布程度。标准差是方差的算术平方根,低标准差意味数据观测趋向于非常靠近均值,而高标准差表示数据散布在一个大的值域中。平均数相同的两组数据,标准差未必相同。

数值属性 X 的 N 个观测值 $x_1,x_2,\cdots,x_N$ 的方差(variance)是:

$$\sigma^2=\frac{\sum_{i=1}^{N}(X_i-\overline{X})^2}{N}$$

其中,σ^2 为方差,X_i 为变量,$\overline{X}$ 为总体均值,N 为样本总数。

统计中的方差是每个样本值与全体样本值的平均数之差的平方值的平均数。在统计描述中,方差用来计算每一个变量与总体平均数之间的差异,统计中的方差计算公式为:

$$\sigma^2=\frac{\sum_{i=1}^{N}(X_i-\overline{X})^2}{N}$$

其中,σ^2 为统计方差,X_i 为变量,$\overline{X}$ 为总体均值,N 为样本总数。

```
>>> print('方差:',np.var(scores1))
方差: 316.5066666666667
>>> print('标准差:',np.std(scores1))
标准差: 17.79063424014632
```

5）离散系数(coefficient of variation)

针对不同数据样本的标准差和方差，因数据衡量单位不同其结果自然无法直接进行对比，为出具一个相同的衡量指标，则进行了离散系数的计算。离散系数为一组数据的标准差与平均数之比：

$$cv = \frac{\sigma}{\overline{X}}$$

离散系数的用途有：比较单位不同的数据的差异程度；比较单位相同但平均数差异很大的两组数据的差异程度；判断特殊差异情况，cv 值通常为 5%～35%，如果 cv 值大于 35%，可怀疑所求平均数是否失去意义；如果 cv 值小于 5%，可怀疑平均数与标准差是否计算错误。

```
>>> print('离散系数:',np.std(scores1)/np.mean(scores1))
离散系数: 0.2450500584042193
```

3. 数据分布状况的统计量

可以使用 pandas 库的 describe()方法来生成数据集的多个汇总统计。

```
>>> import pandas as pd
>>> data={'score':[86,88,92,94,84], 'name':['WangLi','LiHua','LiuTao','WangFei',
'LiMing']}
>>> df=pd.DataFrame(data, columns=['name','score'], index=['101','102','103',
'104', '105'])
>>> df
        name  score
101   WangLi     86
102    LiHua     88
103   LiuTao     92
104  WangFei     94
105   LiMing     84
>>> df.describe()                          #一次性产生多个汇总统计
           score
count   5.000000
mean   88.800000
std     4.147288
min    84.000000
25%    86.000000
50%    88.000000
75%    92.000000
max    94.000000
```

describe()函数的返回结果包括 count、mean、std、min、max 以及百分位数。默认情况下，百分位数分三档：25%、50%、75%，其中 50%就是中位数。

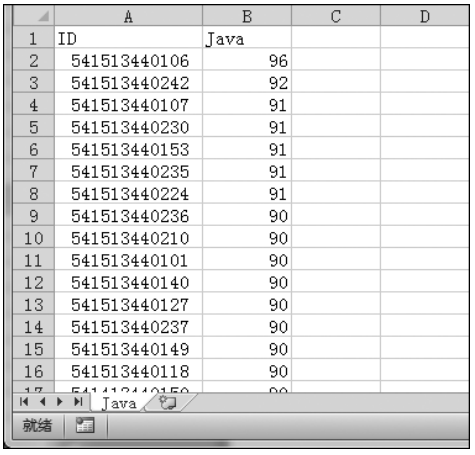
数据分布的不对称性称作偏态。偏态系数(Cs)以平均值与中位数之差对标准差之比率来衡量偏斜的程度。

偏态系数的取值为 0 时，表示数据为完全的对称分布。

偏态系数的取值为正数时，表示数据为正偏态或右偏态。

偏态系数的取值为负数时,表示数据为负偏态或左偏态。

Java 考试成绩如图 3-5 所示。



	A	B	C	D
1	ID	Java		
2	541513440106	96		
3	541513440242	92		
4	541513440107	91		
5	541513440230	91		
6	541513440153	91		
7	541513440235	91		
8	541513440224	91		
9	541513440236	90		
10	541513440210	90		
11	541513440101	90		
12	541513440140	90		
13	541513440127	90		
14	541513440237	90		
15	541513440149	90		
16	541513440118	90		

Java

就绪

图 3-5 Java 考试成绩

```
import pandas as pd
df = pd.read_csv('Java.csv')
data=df['Java']
statistics = data.describe()                                    #保存基本统计量
statistics['range']=statistics['max']-statistics['min']  #增加极差
statistics['cv']=statistics['std']/statistics['mean']     #增加变异系数
statistics['QD']=statistics['75%']-statistics['25%']      #增加四分位差
#统计结果中增加偏态系数'Cs'
statistics['Cs']=(statistics['mean']-data.median())/statistics['std']
print(statistics)
```

运行上面的程序,可以得到下面的结果,此结果为 Java 考试成绩的统计量情况。

```
count  105.000000
mean    80.780952
std      7.617012
min     61.000000
25%     76.000000
50%     81.000000
75%     87.000000
max     96.000000
range   35.000000
cv       0.094292
QD      11.000000
Cs      -0.028758
```

3.3.3 周期性特征

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。根据周期

时间的长短不同分为年度周期性趋势、季节性周期趋势、月度周期性趋势、周周期性趋势、天周期性趋势、小时周期性趋势等。

例如，要对航空旅客数量进行预测，可以先分析旅客数量的时序图来直观地估计旅客数量的变化趋势。

图 3-6 是 1949 年 1 月至 1960 年 12 月某航空公司运载的旅客数量的时序图，总体来看，每年内运送的旅客数量呈周期性变化，各年运送的总旅客数量呈递增趋势。

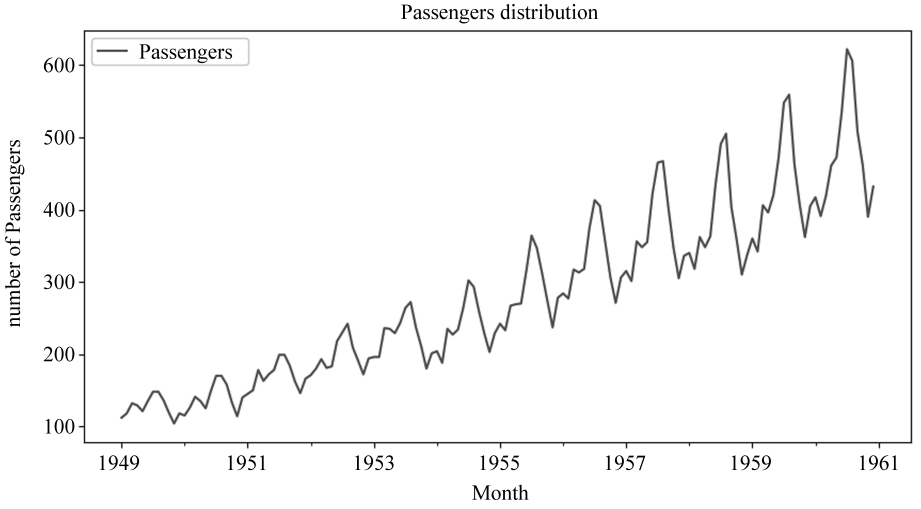


图 3-6　1949 年 1 月至 1960 年 12 月某航空公司运载的旅客数量的时序图

图 3-6 的实现代码如下：

```
>>> import pandas as pd
>>> import matplotlib.pylab as plt
>>>                                          #导入航空旅客数据
>>> data = pd.read_csv("D:\\mypython\\AirPassengers.csv")
>>> data.head(3)                            #显示前 3 条记录
     Month  Passengers
0  1949-01         112
1  1949-02         118
2  1949-03         132
#将 Month 的数据格式转为'%Y-%m-%d'格式
>>> data['Month'] = pd.to_datetime(data['Month'],format='%Y-%m-%d')
>>> df = data.set_index(['Month'])          #将 data 中的列 Month 设置成索引 index
>>> df.head(3)
Month      Passengers
1949-01-01         112
1949-02-01         118
1949-03-01         132
>>> df['Passengers'].plot(kind='line',figsize=[10,5],legend=True,
title='Passengers distribution')           #绘制线型图
>>> plt.ylabel('number of Passengers')      #为 y 轴添加标签
```

```
>>> plt.show()                                    #显示绘制的图形
```


相关性特征

3.3.4 相关性特征

相关分析是研究两个或两个以上的变量间的相关关系的统计分析方法。例如,人的身高和体重之间、空气中的相对湿度与降雨量之间的相关关系。在一段时期内商品房价格随经济水平上升而上升,这说明两指标间是正相关关系;而在另一时期,随着经济水平的进一步发展,出现商品房价格下降的现象,两指标间就是负相关关系。

为了确定相关变量之间的关系,首先应该收集一些数据,这些数据应该是成对的。例如,每人的身高和体重。然后在直角坐标系上描述这些点,这一组点集称为“散点图”。如果这些数据在二维坐标轴中构成的数据点分布在一条直线的周围,那么就说明变量间存在线性相关关系。

相关系数是变量间关联程度的最基本测度之一,如果我们想知道两个变量之间的相关性,那么就可以计算相关系数来进行判定。相关系数 r 的取值为$-1\sim1$。正相关时,r 值为$0\sim1$,散点图是斜向上的,这时一个变量增加,另一个变量也增加;负相关时,r 值为$-1\sim0$,散点图是斜向下的,此时一个变量增加,另一个变量将减少。r 的绝对值越接近 1,两变量的关联程度越强;r 的绝对值越接近 0,两变量的关联程度越弱。

相关系数的值则是由协方差除以两个变量的标准差而得,相关系数的取值为$-1\sim1$,-1 表示完全负相关,1 表示完全相关。相关系数 r 的计算公式表示如下:

$$r=\frac{\operatorname{cov}(X,Y)}{\sqrt{\operatorname{var}(X)\operatorname{var}(Y)}}=\frac{E\{[X-E(X)][Y-E(Y)]\}}{\sigma_x\sigma_y}$$

其中,$\operatorname{cov}(X,Y)$为 X 与 Y 的协方差,$\operatorname{var}(X)$为 X 的方差,$\operatorname{var}(Y)$为 Y 的方差。

一家软件公司在全国有许多代理商,为研究它的财务软件产品的广告投入与销售额的关系,统计人员随机选择 10 家代理商进行观察,搜集到年广告费投入与月均销售额的数据,并编制成相关表,如表 3-1 所示。

表 3-1 年广告费投入与月均销售额

年广告费投入/万元	月均销售额/万元
12.5	21.2
15.3	23.9
23.2	32.9
26.4	34.1
33.5	42.5
34.4	43.2
39.4	49.0
45.2	52.8
55.4	59.4
60.9	63.5

```
>>> import pandas as pd
>>> import matplotlib.pylab as plt
>>> import matplotlib
>>> matplotlib.rcParams['font.family'] = 'FangSong'  #设置字体显示格式
>>> matplotlib.rcParams['font.size'] = '15'          #设置字体大小
>>> data = pd.read_csv("D:\\mypython\\ad-sales.csv") #读取代理商数据
>>> print(data.corr())                               #输出相关系数矩阵
             年广告费投入   月均销售额
年广告费投入      1.000000   0.994198
月均销售额        0.994198   1.000000
#绘制代理商数据的散点图
>>> data.plot(kind='scatter', x='年广告费投入', y='月均销售额', figsize=[5,5],
title='相关性分析')
<matplotlib.axes._subplots.AxesSubplot object at 0x0000000010BB69E8>
>>> plt.show()  #显示散点图,如图 3-7 所示
```

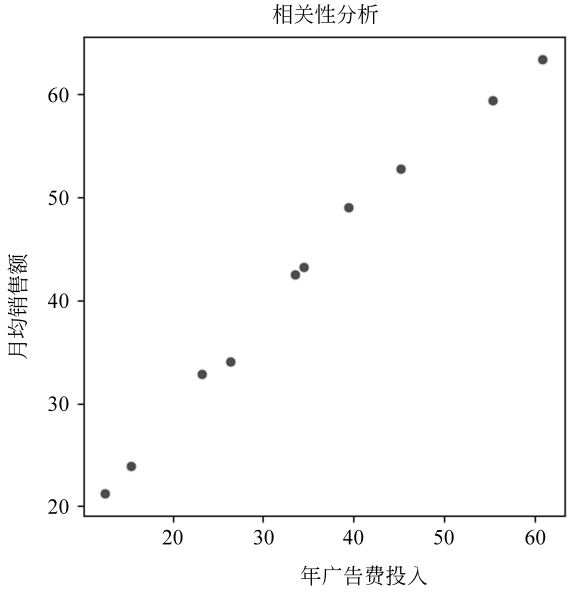


图 3-7　散点图

年广告费投入与月均销售额的相关系数为 0.994198，说明年广告费投入与月均销售额之间有高度的线性正相关关系。

3.4　本章小结

本章主要讲解数据的认识。先讲解了数据类型，具体包括属性类型和数据对象的类型。然后讲解了数据质量分析，具体包括缺失值分析、异常值分析、一致性分析。最后，讲解了数据特征分析，具体包括分布特征、统计量特征、周期性特征和相关性特征。

第 4 章

数据预处理

数据挖掘工作始终是以数据为中心开展的，分类、聚类、回归、关联分析以及可视化等工作的顺利进行完全建立在良好的输入数据的基础之上。采集到的原始数据通常来自多个异构数据源，数据在准确性、完整性和一致性等方面存着多种多样的问题。在数据挖掘之前，首先要做的就是对数据进行预处理，处理数据中的“脏数据”，从而提高数据分析的准确性和有效性。数据预处理通常包括数据清洗、数据集成、数据规范化、数据离散化、数据归约等过程。

4.1 数据清洗

人工输入错误、仪器设备测量精度以及数据收集过程机制缺陷等都会影响采集的数据质量，具体包括测量误差、数据收集错误、噪声、离群点、缺失值、不一致值、重复数据等。数据清洗阶段的主要任务就是填写缺失值、光滑噪声数据、删除离群点和解决属性的不一致性。


处理缺失值

4.1.1 处理缺失值

数据的收集过程很难做到数据全部完整，如数据库表格中的列值不会全部强制性不为空、问卷调查对象不想回答某些选项、数据采集设备异常、数据录入遗漏等。处理缺失值的方法主要有三类：删除元组、数据补齐和不处理。

1. 删除元组

删除元组就是将存在遗漏属性值的对象（也称元组、记录）删除，从而得到一个完备的信息表。这种方法简单易行，在对象有多个属性缺失值、被删除的含缺失值的对象与信息表中的数据量相比非常小的情况下是非常有效的。然而，这种方法有很大的局限性，它是以减少历史数据来换取信息的完备，会造成数据资源的浪费，丢弃了大量隐藏在这些对象中的信息。在信息表中本来包含的对象很少的情况下，删除少量对象就足以严重影响信息表信息的客观性和结果的正确性。因此，当遗漏数据所占比例较大，特别当遗漏数据非随机分布时，这种方法可能导致数据发生偏离，从而得出错误的结论。

2. 数据补齐

数据补齐是使用一定的值对缺失属性值进行填充补齐，从而使信息表完备化。在数据挖掘中，面对的通常是大型的数据库，它的属性有几十个甚至上百个，因为一个属性值的缺

失而放弃大量的其他属性值，这种删除是对信息的极大浪费，因此产生了以可能值对缺失值进行补齐的思想与方法，常用的有如下几种方法。

1）均值补齐

数据的属性分为数值型和非数值型。如果缺失值是数值型的，就以该属性存在值的平均值来补齐缺失的值；如果缺失值是非数值型的，就根据统计学中的众数原理，用该属性的众数（即出现频率最高的值）来补齐缺失的值。

2）利用同类均值补齐

同类均值补齐首先利用聚类方法预测缺失记录所属种类，然后使用与存在缺失值的记录属于同一类的其他记录的平均值来填充空缺值。

3）就近补齐

对于一个包含空值的对象，就近补齐法是在完整数据中找到一个与它最相似的对象，然后用这个相似对象的值来进行填充。不同的问题可能会选用不同的标准来对相似进行判定。该方法的缺点在于难以定义相似标准，主观因素较多。

4）拟合补齐

基于完整的数据集，建立拟合曲线。对于包含空值的对象，将已知属性值代入拟合曲线来估计未知属性值，以此估计值来进行填充。

（1）多项式曲线拟合。

曲线拟合是指用连续曲线近似地刻画或比拟平面上一组离散点所表示的坐标之间的函数关系，是一种用解析表达式逼近离散数据的方法。数据分析中经常会遇到某些记录的个别字段出现缺失值的情况，若能根据记录数据集在该字段的已经存在的值找到一个连续的函数（也就是曲线），使得该字段的已经存在的这些值与曲线能够在最大程度上近似吻合，然后就可以根据拟合曲线函数估算缺失值。

polyfit()函数是 NumPy 中用于最小二乘多项式拟合的一个函数。最小二乘多项式拟合的主要思想是：假设有一组实验数据(x_i, y_i)，事先知道它们之间应该满足某函数关系$y_i = f(x_i)$，通过这些已知信息来确定函数 f 的一些参数。例如，如果函数 f 是线性函数$f(x) = kx + b$，那么参数 k 和 b 就是需要确定的值。如果用 p 表示函数中需要确定的参数，那么目标就是找到一组 p，使得下面关于 p 的 s 函数的函数值最小：

$$s(p) = \sum_{i=1}^{m} [y_i - f(x_i, p)]^2$$

numpy.polyfit()函数的语法格式如下：

```
numpy.polyfit(x, y, deg)
```

作用：numpy.polyfit()使用拟合度为 deg 的多项式 $p(x) = p[0] * x**\text{deg} + \cdots + p[\text{deg}]$拟合一组离散点$(x, y)$，返回具有最小平方误差的拟合函数 $p(x)$的系数向量（含有 deg+1 个数），第一个数是最高次幂的系数。实际上是根据已知点的坐标确定拟合度为 deg 的多项式的系数。

参数说明如下。

x：数据点对应的横坐标，可为行向量、矩阵。

y：数据点对应的纵坐标，可为行向量、矩阵。

deg：要拟合的多项式的阶数，一阶为直线拟合，二阶为抛物线拟合，并非阶数越高越好，看拟合情况而定。一般而言，拟合阶数越大，误差越小，但往往会增大表达式的复杂程度，还有可能出现过拟合，所以要在误差和表达式简洁程度方面综合考虑，确定最佳阶数，一般为3～5最佳。

numpy.polyval(p，x)函数用来计算p所对应的多项式在x处的函数值。

若 p 是长度为 N 的向量，numpy.polyval(p，x)函数返回的值是：

$p[0]*x**(N-1)+p[1]*x**(N-2)+\cdots+p[N-2]*x+p[N-1]$

下面给出进行3阶多项式拟合的程序代码。

```
>>> import matplotlib.pyplot as plt
>>> import numpy as np
>>> x = np.linspace(100,200,30)              #返回 30 个[100,200]上均匀间隔的数字序列
#random_integers(5,20,30)生成[5,20]上离散均匀分布的 30 个整数值
>>> y = x + np.random.random_integers(5,20,30)
>>> p2 = np.polyfit(x,y,deg=3)               #3 阶多项式拟合
>>> q2 = np.polyval(p2, x)                   #计算 p2 所指定的三次多项式在 x 处的函数值
>>> plt.plot(x, y, 'o')
[<matplotlib.lines.Line2D object at 0x000000000EFE1B38>]
>>> plt.plot(x, q2,'k' )
[<matplotlib.lines.Line2D object at 0x000000000E8C9B00>]
>>> plt.show()  #显示 3 阶多项式拟合的绘图结果，如图 4-1 所示
```

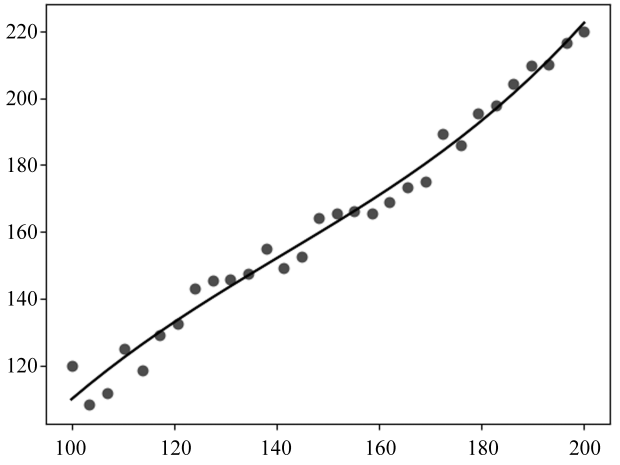


图 4-1　3阶多项式拟合的绘图结果

(2) 各种函数的拟合。

scipy的optimize模块提供了函数最小值(标量或多维)、曲线拟合和寻找等式的根的函数。optimize模块的curve_fit()函数用来将设定的函数f拟合已知的数据集。curve_fit()函数的语法格式如下：

```
scipy.optimize.curve_fit(f, xdata, ydata)
```

作用：使用非线性最小二乘法将函数f拟合到数据。