第 3 章　数据的描述统计量

一组样本数据分布的数值特征可以从三个方面进行描述：一是数据的水平（也称为集中趋势或位置度量），反映全部数据的数值大小；二是数据的差异，反映各数据间的离散程度；三是分布的形状，反映数据分布的偏度和峰度。本章主要介绍描述数据的各统计量的计算方法及其 python 实现。

3.1　描述水平的统计量

数据的水平是指其数值的大小。描述数据水平的概括性统计量主要有平均数、分位数和众数等。

3.1.1　平均数

平均数（mean）是一组数据相加后除以数据的个数得到的结果，也称均值。设一组样本数据为 x_1，x_2，…，x_n，样本量（样本数据的个数）为 n，样本平均数用 $\bar{x}$（读作 x-bar）表示，计算公式为[①]：

$$\bar{x}=\frac{x_1+x_2+\cdots+x_n}{n}=\frac{\sum_{i=1}^{n}x_i}{n} \tag{3.1}$$

式（3.1）也称为**简单平均数**（simple mean）。

如果样本数据被分成 k 组，各组的组中值（组的中间值，是一个组的下限值与上限值的平均数）分别用 m_1，m_2，…，m_k 表示，各组的频数分别用 f_1，f_2，…，f_k 表示，则样本平均数的计算公式为：

$$\bar{x}=\frac{m_1f_1+m_2f_2+\cdots+m_kf_k}{f_1+f_2+\cdots+f_k}=\frac{\sum_{i=1}^{k}m_if_i}{n} \tag{3.2}$$

---

① 如果有总体的全部数据 x_1，x_2，…，x_N，总体平均数用 μ 表示，计算公式为：$\mu=\frac{\sum_{i=1}^{N}x_i}{N}$。实际中，总体平均数往往是不知道的，都是根据样本平均数来推断的。

式（3.2）也称为**加权平均数**（weighted mean）[①]

【例 3-1】（数据：example3_1.csv）在某年级中随机抽取 30 名学生，得到每名学生的数学考试分数如表 3-1 所示。计算 30 名学生考试分数的平均数。

表 3-1　30 名学生的数学考试分数　　单位：分

85	97	83	61	67	86
55	92	70	86	81	75
91	55	96	86	89	91
66	87	72	92	50	82
79	90	90	85	95	66

解： 使用 pandas 模块中 DataFrame 的内置方法、numpy 模块中的 average 函数等均可以计算平均数，代码和结果如代码框 3-1 所示。

代码框 3-1　计算 30 名学生考试分数的平均数

```
# 使用 pandas 模块中 DataFrame 的内置方法
import pandas as pd
example3_1 = pd.read_csv('C:/pydata/chap03/example3_1.csv', encoding='gbk')
pd.DataFrame.mean(example3_1)              # 或写成 example3_1['分数'].mean()
```

```
分数     80.0
dtype: float64
```

注：dtype 表示输出结果的数据类型；float64 表示浮点值（带小数点的值）64 位。

【例 3-2】（数据：example3_2. csv）沿用例 3-1。假定将 30 名学生的数学考试分数分组后的结果如表 3-2 所示。计算考试分数的平均数。

表 3-2　30 名学生数学考试分数的分组数据

分组	组中值（m）	人数（f）
60 以下	55	3
60～70	65	4
70～80	75	4

① 如果总体的 N 个数据被分成 k 组，各组的组中值分别用 M_1，M_2，…，M_k 表示，各组数据出现的频数分别用 F_1，F_2，…，F_k 表示，则总体加权平均数的计算公式为：$\bar{x}=\dfrac{M_1F_1+M_2F_2+\cdots+M_kF_k}{f_1+f_2+\cdots+f_k}=\dfrac{\sum_{i=1}^{k}M_iF_i}{N}$。

续表

分组	组中值（m）	人数（f）
80～90	85	10
90～100	95	9
合计	—	30

解：计算加权平均数的代码和结果如代码框 3-2 所示。

代码框 3-2　计算 30 名学生考试分数的加权平均数

```
# 使用 numpy 中的 average 函数
import pandas as pd
import numpy as np
example3_2 = pd.read_csv("C:/pydata/chap03/example3_2.csv",encoding='gbk')
m=example3_2['组中值']
f=example3_2['人数']
print('加权平均 :',np.average(a=m,weights=f))    # 计算加权平均数并输出结果
```

```
加权平均：81.0
```

对于同一组数据，根据原始数据计算的平均数与根据分组后数据计算的平均数结果是有差异的。因为分组后是用组中值代表该组数据，实际上假定该组数据在组中值两侧呈对称分布，如果实际情况不是对称分布，根据分组后数据计算的平均数就会有偏差。因此，在有原始数据的情况下，应根据原始数据计算平均数。加权平均数只在所得到的数据本身就是分组数据的情况下使用。

3.1.2　分位数

一组数据按从小到大排序后，可以找出排在某个位置上的数值，该数值可以代表数据水平的高低。这些位置上的数值就是相应的**分位数**（quantile）。常用的分位数有中位数、四分位数、百分位数等。

1. 中位数

中位数（median）是一组数据排序后处于中间位置上的数值，用 M_e 表示。中位数是用一个点将全部数据等分成两部分，每部分包含 50% 的数据，一部分数据比中位数大，另一部分比中位数小。中位数是用中间位置上的值代表数据的水平，其特点是不受极端值的影响，在研究收入分配时很有用。

计算中位数时，要先对 n 个数据从小到大进行排序，然后确定中位数的位置，最后计算中位数的具体数值。

设一组数据 x_1，x_2，…，x_n 按从小到大排序后为 $x_{(1)}$，$x_{(2)}$，…，$x_{(n)}$，则中位数就是 $(n+1)/2$ 位置上的值。计算公式为：

$$M_e=\begin{cases} x_{\left(\frac{n+1}{2}\right)} & n\text{为奇数} \\ \frac{1}{2}\left\{x_{\left(\frac{n}{2}\right)}+x_{\left(\frac{n}{2}+1\right)}\right\} & n\text{为偶数} \end{cases} \tag{3.3}$$

【例 3-3】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生数学考试分数的中位数。

解：代码和结果如代码框 3-3 所示。

代码框 3-3 计算 30 名学生考试分数的中位数

```
import pandas as pd
example3_1['中位数'].median()          #  计算中位数
```

```
中位数     85.0
dtype: float64
```

2. 四分位数

四分位数（quartile）是一组数据排序后处在 25% 和 75% 位置上的数值。它是用 3 个点将全部数据等分为 4 部分，其中每部分包含 25% 的数据。显然，中间的四分位数就是中位数，因此通常所说的四分位数是指处在 25% 位置上和 75% 位置上的两个数值。

与中位数的计算方法类似，计算四分位数时，首先对数据进行排序，然后确定四分位数所在的位置，该位置上的数值就是四分位数。与中位数不同的是，四分位数位置的确定方法有多种①，每种方法得到的结果可能会有一定差异，但差异不会很大（一般相差不会超过一个位次）。由于不同软件使用的计算方法可能不一样，对同一组数据用不同软件得到的四分位数结果也可能会有差异，但不会影响分析的结论。

设 25% 位置上的四分位数为 $Q_{25\%}$，75% 位置上的四分位数为 $Q_{75\%}$，Python 默认的计算四分位数位置的公式为：

$$Q_{25\%}\text{位置}=\frac{n+3}{4},\quad Q_{75\%}\text{位置}=\frac{3n+1}{4} \tag{3.4}$$

如果位置是整数，四分位数就是该位置对应的数值；如果是在整数加 0.5 的位置上，则取该位置两侧数值的平均数；如果是在整数加 0.25 或 0.75 的位置上，则四分位数等于该位置前面的数值加上按比例分摊的位置两侧数值的差值。

【例 3-4】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的四分位数。

解：首先，对 n 个数据从小到大进行排序，得到的结果如下：

50 55 55 61 66 66 67 70 72 75 79 81 82 83 85

85 86 86 86 87 89 90 90 91 91 92 92 95 96 97

① 式（3.4）是 Python、R 和 Excel 等默认使用的位置确定公式。Minitab 和 SPSS 软件使用的四分位数位置的计算公式为：$Q_{25\%}=\frac{n+1}{4}$，$Q_{75\%}=\frac{3(n+1)}{4}$。

其次，计算出四分位数的位置和相应的数值。

$Q_{25\%}$位置$=\dfrac{n+3}{4}=\dfrac{30+3}{4}=8.25$，即 $Q_{25\%}$ 在第 8 个数值（70）和第 9 个数值（72）之间 0.25 的位置上，因此，$Q_{25\%}=70+0.25\times(72-70)=70.5$。

$Q_{75\%}$位置$=\dfrac{3n+1}{4}=\dfrac{3\times30+1}{4}=22.75$，即 $Q_{75\%}$ 在第 22 个数值（90）和第 23 个数值（90）之间 0.75 的位置上，因此，$Q_{75\%}=90+0.75\times(90-90)=90$　。

计算四分位数的代码和结果如代码框 3-4 所示。

代码框 3-4　计算 30 名学生考试分数的四分位数

```
import pandas as pd
pd.DataFrame.quantile(example3_1,q=[0.25,0.75],interpolation='linear')
                                    # 计算四分位数，默认采用线性插值
```

	分数
0.25	70.5
0.75	90.0

注：DataFrame.quantile 函数提供了 5 种四分位数的算法，默认算法是采用线性插值，即按式（3.4）计算四分位数的位置。该算法与 R 默认的算法及 Excel 算法相同。

在 $Q_{25\%}$ 和 $Q_{75\%}$ 之间大约包含了 50% 的数据。就上面 30 个学生的考试分数而言，可以说大约有一半学生的考试分数在 70.5～90 分之间。

3. 百分位数

百分位数（percentile）是用 99 个点将数据 100 等分，处在各分位点上的数值就是百分位数。百分位数提供了各项数据在最小值和最大值之间分布的信息。

与四分位数类似，百分位数也有多种算法[①]，每种算法的结果不尽相同，但差异不会很大。设 $P_{i\%}$ 为第 i 个百分位数，Python 默认的第 i 个百分位数的位置计算公式为：

$$P_{i\%}\text{位置}=\frac{i}{100}\times(n-1) \tag{3.5}$$

由于 Python 是一种索引从 0 开始计数的语言，因此，排序后的第 1 个数的索引为 0，第 2 个数为 1，第 3 个数为 2，依此类推。

如果位置是整数，百分位数就是该位置对应的数值；如果位置不是整数，百分位数等于该位置前面的数值加上按比例分摊的位置两侧数值的差值。显然，中位数就是第 50 个百分位数 $P_{50\%}$，$Q_{25\%}$ 和 $Q_{75\%}$ 就是第 25 个百分位数 $P_{25\%}$ 和第 75 个百分位数 $P_{75\%}$。

---

① 式（3.5）是 Python、R 和 Excel 等使用的位置确定公式。Minitab 和 SPSS 软件使用的第 i 个百分位数的位置计算公式为：$P_{i\%}\text{位置}=\dfrac{i}{100}\times(n+1)$。

【例 3-5】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的第 10、25、50、75 和 90 个百分位数。

解：根据式（3.5）有：$P_{10\%}$位置$=\frac{10}{100}\times(30-1)=2.9$。该百分位数在第 3 个值（55）和第 4 个值（61）之间 0.9 的位置上，因此$P_{10\%}=55+0.9\times(61-55)=60.4$。其他百分位数的算法类似。

计算百分位数的代码和结果如代码框 3-5 所示。

代码框 3-5 计算 30 名学生考试分数的百分位数

```
import pandas as pd
pd.DataFrame.quantile(example3_1,q=[0.1,0.25,0.5,0.75,0.9],interpola-
tion='linear')                    # 计算百分位数，默认采用线性插值
```

	分数
0.10	60.4
0.25	70.5
0.50	85.0
0.75	90.0
0.90	92.3

3.1.3 众数

众数（mode）是一组数据中出现频数最多的数值，用 M_0 表示。一般情形下，只有在数据量较大时众数才有意义。从分布的角度看，众数是一组数据分布的峰值点所对应的数值。如果数据的分布没有明显的峰值，众数也可能不存在；如果有两个或多个峰值，也可以有两个或多个众数。

【例 3-6】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的众数。

解：计算众数的代码和结果如代码框 3-6 所示。

代码框 3-6 计算 30 名学生考试分数的众数

```
import pandas as pd
mode=example3_1['分数'].mode()  # 或写成 pd.DataFrame.mode(example3_1)
print(mode)
```

```
0    86
dtype: int64
```

本节介绍了描述数据水平的几个主要统计量。实际应用时选择哪个统计量来代表一组数据的水平，取决于数据的分布形态。当数据分布对称或接近对称时，建议使用平均数；当数据分布明显偏斜时，可考虑使用分位数或众数。

3.2 描述差异的统计量

数据之间的差异反映了数据的离散程度。各水平统计量对该组数据的代表程度取决于数据的离散程度，离散程度越大，各水平统计量对该组数据的代表性就越差；离散程度越小，其代表性就越好。描述样本数据离散程度的统计量主要有极差、四分位差、方差和标准差以及测度相对离散程度的变异系数等。

3.2.1 极差和四分位差

极差（range）是一组数据的最大值与最小值之差，也称**全距**，用 R 表示。计算公式为：

$$R = \max(x) - \min(x) \tag{3.6}$$

比如，根据例 3-1 中的数据，计算 30 名学生考试分数的极差为：$R = 97 - 50 = 47$。由于极差只利用了一组数据两端的信息，因此容易受极端值的影响，不能全面反映差异状况。虽然极差在实际中很少单独使用，但它总是作为分析数据离散程度的一个参考值。

四分位差也称**四分位距**（interquartile range），是一组数据 75% 位置上的四分位数与 25% 位置上的四分位数之差，用 IQR 表示，计算公式为：

$$\mathrm{IQR} = Q_{75\%} - Q_{25\%} \tag{3.7}$$

四分位差反映了中间 50% 数据的离散程度，其数值越小，说明中间的数据越集中；数值越大，说明中间的数据越分散。四分位差不受极值的影响。此外，由于中位数处于数据的中间位置，因此，四分位差的大小在一定程度上也说明了中位数对一组数据的代表程度。

【例 3-7】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的极差和四分位差。

解：计算极差和四分位差的代码和结果如代码框 3-7 所示。

代码框 3-7 计算 30 名学生考试分数的极差和四分位差

```
# 计算极差
import pandas as pd
R=example3_1['分数'].max() - example3_1['分数'].min()
print('极差 :',R)
```

```
极差：47
```

```
# 计算四分位差
import numpy as np
IQR=np.quantile(example3_1['分数'], q=0.75) - np.quantile(example3_1['分数'],
q=0.25)
print('IQR =',IQR)
```

```
IQR = 19.5
```

3.2.2 方差和标准差

如果考虑每个数据 x_i 与其平均数 $\bar{x}$ 之间的差异，以此作为一组数据离散程度的度量，结果要比极差和四分位差更为全面、准确。这就需要计算每个数据 x_i 与其平均数 $\bar{x}$ 离差的平均数。但由于 $\sum_{i=1}^{n}(x_i-\bar{x})$ 等于 0，因此需要进行一定的处理。一种方法是将离差取绝对值，求和后再平均，这一结果称为**平均离差**（mean deviation）或**平均绝对离差**（mean absolute deviation）；另一种方法是将离差平方后再求平均数，这一结果称为**方差**（variance）。方差开平方根后的结果称为**标准差**（standard deviation），它是一组数据与其平均数相比平均相差的数值。方差（或标准差）是实际中应用最广泛的度量数据离散程度的统计量。

设样本方差为 s^2，计算公式为：

$$s^2=\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1} \tag{3.8}$$

样本标准差的计算公式为：

$$s=\sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n-1}} \tag{3.9}$$

与方差不同的是，标准差与原始数据的计量单位相同，其实际意义要比方差清楚。因此，在对实际问题进行分析时通常使用标准差。

【例 3-8】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的方差和标准差。

解：计算方差和标准差的代码和结果如代码框 3-8 所示。

代码框 3-8　计算 30 名学生考试分数的方差和标准差

```
# 计算方差
import pandas as pd
var = example3_1['分数'].var(ddof=1)        # 计算方差
print('方差：',var)
```

```
方差 : 174.6206896551724
```

```
# 计算标准差
sd=example3_1['分数'].std()
print('标准差 :',round(sd,2))                # 结果保留 2 位小数
```

```
标准差：13.21
```

注：方差和标准差的自由度由函数的参数 ddof（自由度）设置，不同函数的默认设置不同。numpy 中的函数默认 ddof=0，即分母是 n 而非 n-1，pandas 中的函数默认 ddof=1，即自由度为 n-1。

3.2.3 变异系数

标准差是反映数据离散程度的绝对值，其数值的大小受原始数据大小的影响，数据的观测

值越大，标准差的值通常也就越大。此外，标准差与原始数据的计量单位相同，采用不同计量单位计量的数据，其标准差的值也就不同。因此，对于不同样本的数据，如果原始数据的观测值相差较大或计量单位不同，就不能用标准差直接比较其离散程度，这时需要计算变异系数。

变异系数（coefficient of variation，CV）也称**离散系数**，它是一组数据的标准差与其相应的平均数之比。由于变异系数消除了数值大小和计量单位对标准差的影响，因而可以反映一组数据的相对离散程度。其计算公式为：

$$CV = \frac{s}{\bar{x}} \tag{3.10}$$

变异系数主要用于比较不同样本数据的离散程度。其数值越大，说明数据的相对离散程度就越大，数值越小，说明数据的相对离散程度就越小[①]。

【例 3-9】（数据：example2_3. csv）沿用第 2 章的例 2-3。计算 6 个城市 AQI 的平均数、标准差和变异系数，比较 AQI 离散程度的大小。

解：如果各城市 AQI 的平均数差异不大，可以直接比较标准差的大小，否则需要用变异系数比较其离散程度。代码和结果如代码框 3-9 所示。

代码框 3-9　计算 6 个城市 AQI 的平均数、标准差和变异系数

```
import pandas as pd
import numpy as np
example2_3 = pd.read_csv('C:/pydata/chap02/example2_3.csv', encoding='gbk')

s_mean = example2_3.mean()      # 计算平均数
s_sd = example2_3.std()         # 计算标准差
s_cv = s_sd / s_mean            # 计算变异系数
df = pd.DataFrame({"平均数": s_mean, "标准差": s_sd, "变异系数": s_cv})
                                # 生成结果数据框
np.round(df, 4)                 # 结果保留 4 位小数
```

	平均数	标准差	变异系数
北京	78.445 4	41.627 8	0.530 7
上海	67.939 9	28.977 2	0.426 5
郑州	98.907 1	42.720 7	0.431 9
武汉	72.153 0	30.354 9	0.420 7
西安	91.789 6	45.708 7	0.498 0
沈阳	78.571 0	40.669 8	0.517 6

比较代码框 3-9 的变异系数可知，AQI 离散程度最大的是北京，最小的是武汉。

① 当平均数接近 0 时，变异系数的值趋于无穷大，此时需要慎重解释。

3.2.4 标准分数

标准分数（standard score）也称 z 分数或标准化值。它可以用于度量每个数值在该组数据中的相对位置，并判断一组数据是否有离群点。比如，全班的平均考试分数为 80 分，标准差为 10 分，如果一个学生的考试分数是 90 分，表示距离平均分数有 1 个标准差的距离。这里的 1 就是这个学生考试成绩的标准分数。标准分数描述的是某个数据与平均数相比相差多少个标准差，它是某个数据与其平均数的差除以标准差后的数值。设标准分数为 z，计算公式为：

$$z_i = \frac{x_i - \bar{x}}{s} \tag{3.11}$$

式（3.11）也就是统计上常用的标准化公式。在对多个具有不同计量单位的变量进行分析时，常常需要对各变量做标准化处理，也就是把一组数据转化成平均数为 0、标准差为 1 的新数据。实际上，标准分数只是将原始数据做了线性变换，它并没有改变某个数值在该组数据中的位置，也没有改变该组数据分布的形状。

【例 3-10】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的标准分数。

解：计算标准分数的代码和结果如代码框 3-10 所示。

代码框 3-10　计算 30 名学生考试分数的标准分数

```
import pandas as pd
import numpy as np
from scipy import stats        # scipy 是基于 numpy 的科学计算包
example3_1 = pd.read_csv('C:/pydata/chap03/example3_1.csv', encoding='gbk')
z = stats.zscore(example3_1['分数'], ddof=1)  # ddof 是自由度
z=np.round(z, 4)
print('标准分数：','\n',z)
```

```
标准分数：
[0.3784  -1.8919   0.8324  -1.0594  -0.0757   1.2865   0.9081  -1.8919
 0.5297   0.7567   0.227   -0.7567   1.2108  -0.6054   0.7567  -1.4378
 0.454    0.454    0.9081   0.3784  -0.9838   0.0757   0.6811  -2.2702
 1.1351   0.454   -0.3784   0.8324   0.1513  -1.0594]
```

第一个学生的标准分数为 0.378 4，表示这个学生的考试分数与平均分数（80 分）相比高出 0.378 4 个标准差；第二个学生的标准分数为 -1.891 9，表示考试分数与平均分数相比低 1.891 9 个标准差。其余的含义类似。

根据标准分数可以判断一组数据中是否存在离群点。经验表明：当一组数据对称分布时，约有 68% 的数据在平均数加减 1 个标准差的范围之内，约有 95% 的数据在平均数加减 2 个标准差的范围之内，约有 99% 的数据在平均数加减 3 个标准差的范围之内。可以想象，一组数据中低于或高于平均数 3 倍标准差的数值是很少的，因此，通常将 3 个标准差之外的数据确定

为离群点。

3.3 描述分布形状的统计量

通过直方图或核密度图可以大致看出数据的分布是否对称。对于不对称的分布，要想知道不对称程度则需要计算相应的描述统计量。偏度系数和峰度系数就是对分布不对称程度和峰值高低的一种度量。

3.3.1 偏度系数

偏度（skewness）是指数据分布的不对称性，这一概念由统计学家 K. Pearson 于 1895 年首次提出。测度数据分布不对称性的统计量称为偏度系数（coefficient of skewness），记为 SK。

设$m_r=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^r$为样本的 r 阶中心距，偏度系数有以下 3 种算法。

算法 1：对应于 R 的 e1071 包中 skewness 函数的 type=1，该方法也是很多比较传统教材中的定义。计算公式为：

$$\mathrm{SK}_1=\frac{m_3}{m_2^{3/2}}=\frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]^{\frac{3}{2}}} \tag{3.12}$$

式中，m_3 为样本的三阶中心矩；m_2 为样本的二阶中心矩。

算法 2：该算法对应的实现软件有：R 的 e1071 包中 skewness 函数的 type=2，Python 的 pandas 模块中 DataFrame.skew 函数、SPSS、SAS、Excel 软件中的默认算法。计算公式为：

$$\mathrm{SK}_2=\mathrm{SK}_1\times\frac{\sqrt{n(n-1)}}{n-2}=\frac{m_3}{m_2^{3/2}}\times\frac{\sqrt{n(n-1)}}{n-2} \tag{3.13}$$

算法 3：该算法对应的实现软件有：R 的 e1071 包中 skewness 函数的 type=3（函数默认算法）和 Minitab（默认算法）。计算公式为：

$$\mathrm{SK}_3=\frac{m_3}{s^3}=\mathrm{SK}_1\left(\frac{n-1}{n}\right)^{3/2}=\frac{m_3}{m_2^{3/2}}\left(\frac{n-1}{n}\right)^{3/2} \tag{3.14}$$

式中，s 是样本标准差。

当数据对称分布时，偏度系数等于 0。偏度系数越接近 0，偏斜程度就越低，也就越接近对称分布。如果偏度系数明显不同于 0，表示分布是不对称的。若偏度系数大于 1 或小于 −1，视为严重偏斜分布；若偏度系数在 0.5～1 或 −1～−0.5 之间，视为中等偏斜分布；若偏度系数在 −0.5～0.5 之间，视为轻微偏斜。其中负值表示左偏分布（在分布的左侧有长尾），正值则表示右偏分布（在分布的右侧有长尾）。

3.3.2 峰度系数

峰度（kurtosis）是指数据分布峰值的高低，这一概念由统计学家 K.Pearson 于 1905 年首次提出。测度一组数据分布峰值高低的统计量是**峰度系数**（coefficient of kurtosis），记作 K。

设$m_r=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^r$为样本的 r 阶中心距，峰度系数有以下 3 种算法。

算法 1：对应于 R 的 e1071 包中 kurtosis 函数的 type=1，该方法也是很多传统教材中的定义。计算公式为：

$$K_1=\frac{m_4}{(m_2)^2}-3 \tag{3.15}$$

算法 2：该算法对应的实现软件有：对应于 R 的 e1071 包中 kurtosis 函数的 type=2，Python 的 pandas 模块中 DataFrame. kurt 函数、SPSS、SAS、Excel 软件中的默认算法。计算公式为：

$$K_2=[(n+1)K_1+6]\times\frac{n-1}{(n-2)(n-3)} \tag{3.16}$$

算法 3：该算法对应的实现软件有：R 的 e1071 包中 kurtosis 函数的 type=3（函数默认算法）和 Minitab（默认算法）。计算公式为：

$$K_3=\frac{m_4}{s^4}-3=(K_1+3)\left(1-\frac{1}{n}\right)^2-3 \tag{3.17}$$

式中，s 是样本标准差。

峰度通常是与标准正态分布相比较而言的。由于标准正态分布的峰度系数为 0，当 $K>0$ 时，为尖峰分布，数据分布的峰值比标准正态分布高，数据相对集中；当 $K<0$ 时，为扁平分布，数据分布的峰值比标准正态分布低，数据相对分散。

图 3-1 模拟了不同分布形状对应的偏度系数和峰度系数。

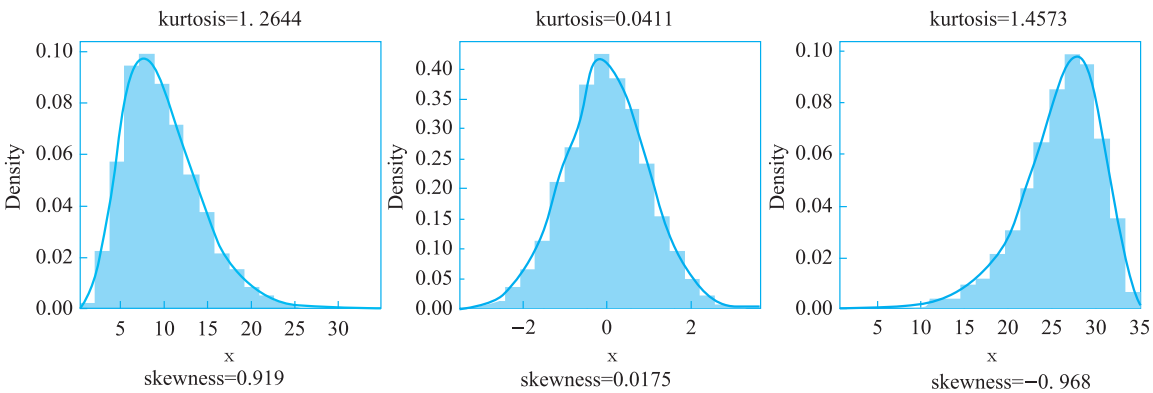

图 3-1 不同分布形状对应的偏度系数和峰度系数

【例 3-11】（数据：example3_1. csv）沿用例 3-1。计算 30 名学生考试分数的偏度系数和峰度系数。

解：计算偏度系数和峰度系数的代码和结果如代码框 3-11 所示。

代码框 3-11 计算 30 名学生考试分数的偏度系数和峰度系数

```
# 计算偏度系数
import pandas as pd
example3_1 = pd.read_csv('C:/pydata/chap03/example3_1.csv', encoding='gbk')
skew = example3_1['分数'].skew()

# 计算峰度系数
kurt = example3_1['分数'].kurt()
print("偏度系数：", round(skew, 4),'\n'"峰度系数：", round(kurt, 4))
```

```
偏度系数： -0.831 4
峰度系数： -0.351 5
```

代码框 3-11 中的结果显示，30 名学生考试分数的偏度系数为 -0.831 4，表示考试分数的分布为中等程度的左偏分布。峰度系数为 -0.351 5，表示考试分数分布的峰值比标准正态分布的峰值要略低一些。

3.4 一个综合描述的例子

以上介绍了各统计量的算法。这些统计量可以使用 Python 不同模块中的函数进行计算。在分析实际数据时，往往需要一次输出多个统计量对数据进行综合性描述分析。比如，对于例 3-9 的数据，要想一次输出 6 个城市 AQI 的一些主要描述统计量，可以使用 pandas 中的 DataFrame.describe 函数输出常用的描述统计量，代码和结果如代码框 3-12 所示。

代码框 3-12 汇总输出例 3-9 的主要描述统计量

```
import pandas as pd
df = pd.read_csv("C:/pydata/chap02/example2_3.csv",encoding='gbk')
np.round(df.describe(),2)
```

	北京	上海	郑州	武汉	西安	沈阳
count	366.00	366.00	366.00	366.00	366.00	366.00
mean	78.45	67.94	98.91	72.15	91.79	78.57
std	41.63	28.98	42.72	30.35	45.71	40.67
min	22.00	27.00	32.00	20.00	30.00	22.00

续表

	北京	上海	郑州	武汉	西安	沈阳
25%	47.00	47.00	72.00	49.00	61.00	52.00
50%	69.00	60.00	90.00	69.00	81.00	67.00
75%	98.75	81.75	120.00	88.00	111.75	94.75
max	257.00	206.00	298.00	223.00	278.00	315.00

输出的主要描述统计量有样本量（count）、平均数（mean）、标准差（std）、最小值（min）、中位数（50%）和两个四分位数（25% 和 75%）、最大值（max）。

在实际分析中，通常要对数据从图表和统计量两个方面同时进行描述。下面结合一个例子说明对数据进行综合描述的基本思路。

【例 3-12】（数据：example3_12. csv）在某大学随机抽取 60 名大学生，调查得到他们的性别、家庭所在地和月生活费支出（单位：元）数据如表 3-3 所示。对调查数据进行综合分析。

表 3-3　60 名大学生的调查数据

性别	家庭所在地	月生活费支出（元）	性别	家庭所在地	月生活费支出（元）
女	中小城市	1 500	女	乡镇地区	1 850
男	大型城市	2 000	女	乡镇地区	2 000
男	大型城市	1 800	女	中小城市	1 700
女	中小城市	1 600	女	大型城市	1 800
女	中小城市	2 000	男	中小城市	1 860
女	大型城市	2 100	男	乡镇地区	1 950
男	大型城市	1 100	女	中小城市	1 900
男	大型城市	1 780	男	中小城市	2 000
女	中小城市	1 550	女	乡镇地区	1 870
女	乡镇地区	1 300	女	中小城市	1 900
男	大型城市	2 000	女	大型城市	2 400
男	大型城市	1 700	女	大型城市	2 000
女	中小城市	1 400	女	中小城市	2 360
女	大型城市	1 500	女	中小城市	2 050
男	大型城市	1 400	女	大型城市	2 200
男	大型城市	1 480	男	大型城市	2 000
女	中小城市	2 350	女	中小城市	1 750
女	中小城市	1 450	女	中小城市	2 250

续表

性别	家庭所在地	月生活费支出（元）	性别	家庭所在地	月生活费支出（元）
男	大型城市	1 500	女	大型城市	2 800
男	大型城市	1 760	女	大型城市	1 900
男	中小城市	1 300	男	大型城市	2 000
男	中小城市	1 600	男	乡镇地区	1 900
女	中小城市	1 680	女	中小城市	2 200
男	中小城市	1 850	女	乡镇地区	1 800
男	乡镇地区	1 500	女	乡镇地区	1 900
女	中小城市	1 600	男	乡镇地区	1 500
男	大型城市	1 300	男	大型城市	2 000
女	大型城市	1 800	男	大型城市	1 900
女	大型城市	1 550	女	大型城市	2 300
男	中小城市	1 350	女	中小城市	1 900

解：这里涉及两个类别变量和一个数值变量。对于性别和家庭所在地两个类别变量，可以对其频数进行计数，计算百分比，并画出条形图和饼图等进行描述。对于月生活费支出变量，可以绘制直方图、核密度图、箱线图等来观察其分布特征，还可绘制轮廓图、雷达图等比较不同性别和不同家庭所在地月生活费支出的相似性，并计算均值和标准差等统计量进行分析。下面给出一些主要的分析思路供读者参考。

首先，绘制图形描述月生活费支出的分布特征。为反映全部学生的月生活费支出的整体分布特征，可以绘制出 60 名学生月生活费支出的直方图、核密度图、箱线图等；为观察按性别和家庭所在地分组的月生活费支出的分布特征，可以绘制分组箱线图和分组核密度图等。代码和结果如代码框 3-13 所示。

代码框 3-13　60 名大学生月生活费支出的图形描述

```
# 月生活费支出的直方图和箱线图（图 3-2）
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
plt.rcParams['font.sans-serif'] = ['SimHei']

example3_12 = pd.read_csv('C:/pydata/chap03/example3_12.csv', encoding='gbk')
x = example3_12 ['月生活费支出']
fig = plt.figure(figsize=(6,5))
spec = fig.add_gridspec(nrows=2, ncols=1, height_ratios=[1,3])
                                # 设置图形的行数、列数和高度比
```

```
ax1 = fig.add_subplot(spec[0,:])
x.plot(kind='box', vert=False)
ax1.spines['right'].set_color('none')
ax1.spines['top'].set_color('none')
ax1.spines['left'].set_color('none')
ax1.spines['bottom'].set_color('none')
ax1.set_xticks([])

ax2 = fig.add_subplot(spec[1,:])
x.plot(bins=10, kind='hist', ax=ax2, density=True, legend=False)
x.plot(kind='density', ax=ax2, linewidth=3)
ax2.set_xlabel('月生活费支出')
ax2.set_ylabel('频率')
ax2.set_xlim(x.min(), x.max())
ax2.spines['right'].set_color('none')
ax2.spines['top'].set_color('none')
plt.tight_layout()
plt.show()
```

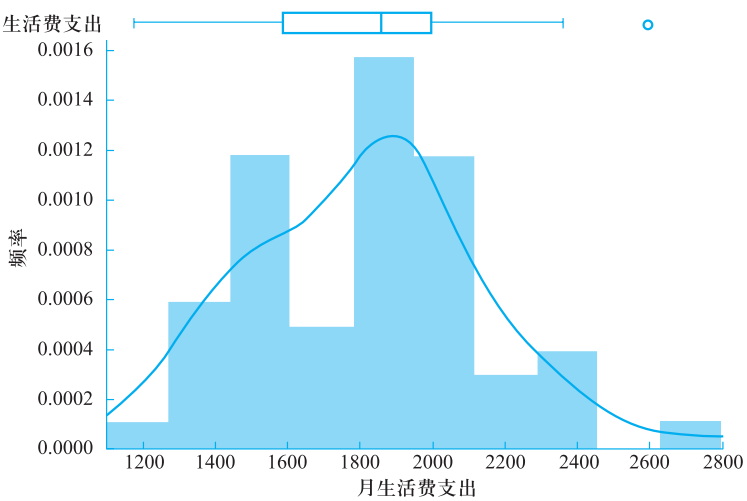

图 3-2 带有箱线图、核密度估计曲线的月生活费支出的直方图

```
# 按性别和家庭所在地分类绘制核密度图（图 3-3）
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
example3_12 = pd.read_csv('C:/pydata/chap03/example3_12.csv', encoding='gbk')
plt.rcParams['font.sans-serif'] = ['SimHei']
```

```
plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
sns.kdeplot('月生活费支出',hue='性别',shade=True,data=example3_12)
plt.title('(a) 按性别分组')
plt.subplot(1,2,2)
sns.kdeplot('月生活费支出',hue='家庭所在地',shade=True,data=example3_12)
plt.title('(b) 按家庭所在地分组')
```

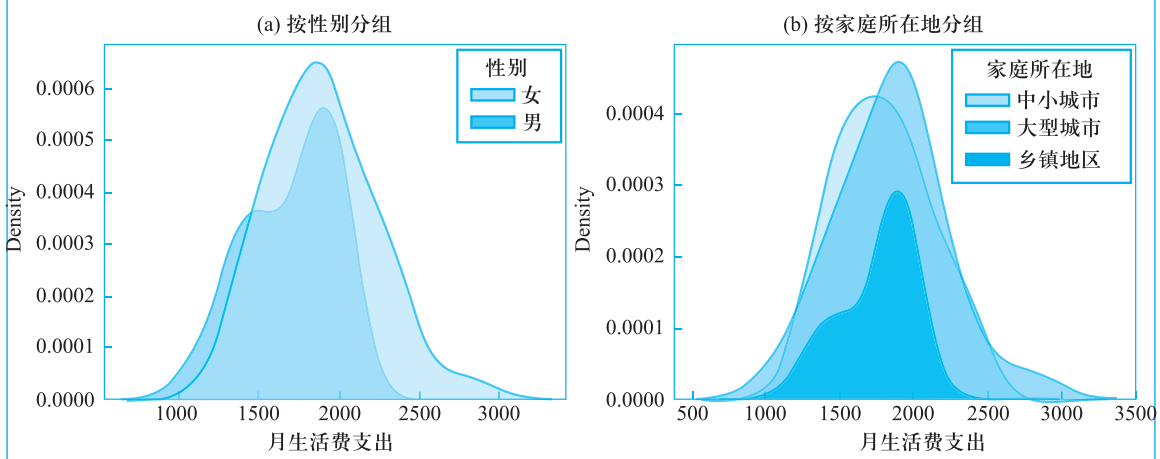

图 3-3 按性别和家庭所在地分组的核密度图

```
# 按性别和家庭所在地分组绘制箱线图（图 3-4）
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.figure(figsize=(9,6))
sns.boxplot(x='性别', y="月生活费支出", hue='家庭所在地', data=example3_12)
plt.show()
```

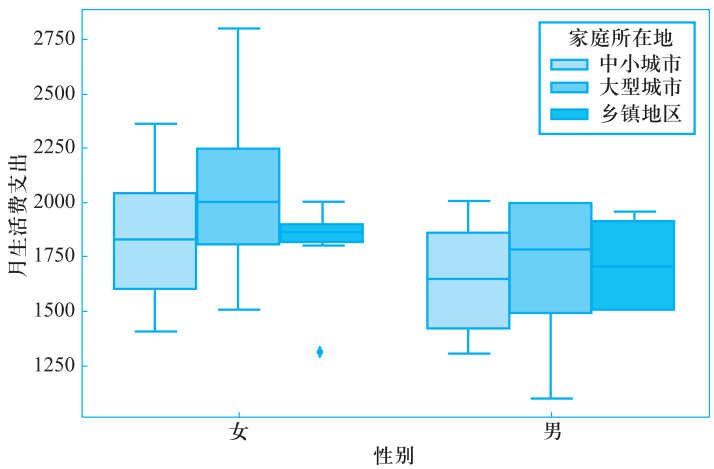

图 3-4 按性别和家庭所在地分组的箱线图

```
# 绘制按性别和家庭所在地分组的点图（图 3-5）
plt.subplots(1,2,figsize=(8, 4))
plt.subplot(121)
sns.stripplot(x='性别',y='月生活费支出',
              jitter=True,                # 扰动数据
              size=5,                     # 设置点的大小
              data=example3_12)
plt.title('(a) 按性别分组')

plt.subplot(122)
sns.stripplot(x='家庭所在地',y='月生活费支出',size=5,data=example3_12)
plt.title('(b) 按家庭所在地分组')
```

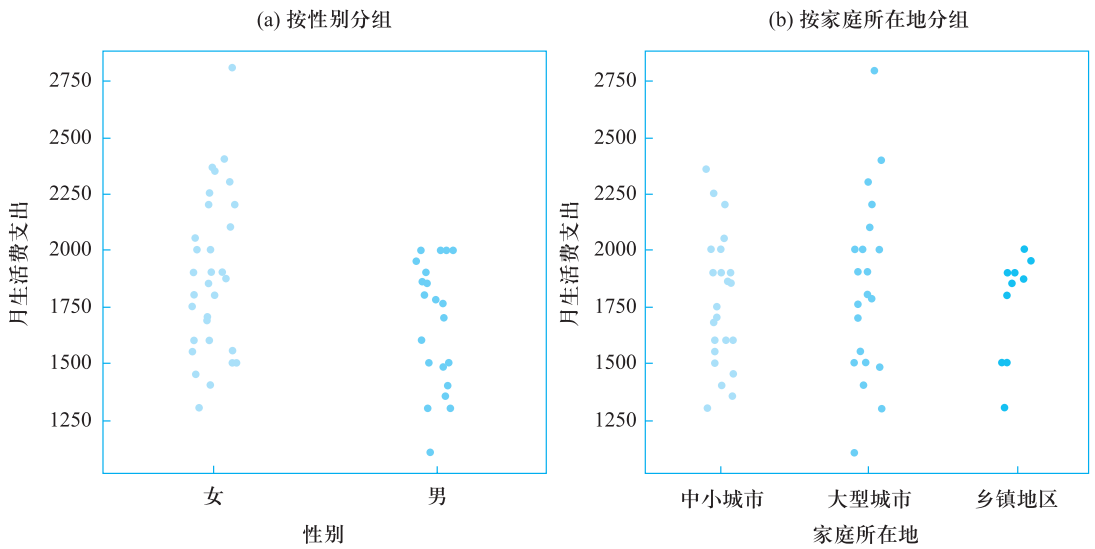

图 3-5 按性别和家庭所在地分组的点图

图 3-2 显示，大学生月生活费支出的分布基本上是对称的，也就是以均值为中心，两侧依次减少，这基本上符合大学生生活费支出的特点。图 3-3 分别展示了按性别和家庭所在地分组的核密度图，结果显示，女生支出的平均水平明显高于男生；大型城市和中小城市的平均支出水平差异不大，乡镇地区的平均支出水平偏低。大型城市女生支出的平均水平明显高于中小城市和乡镇地区。图 3-4 展示了按性别和按家庭所在地交叉分组的学生月生活费支出分布的特点。图 3-5 的分类点图显示了月生活费支出各点的分布和离散状况。

然后，对学生性别和家庭所在地进行频数分析，观察各类学生人数的分布；对全部学生的月生活费支出计算描述统计量并进行分析；按性别和家庭所在地分组计算描述统计量等。代码和结果如代码框 3-14 所示。

代码框 3-14 对性别和家庭所在地计数、对月生活费支出计算描述统计量

```
# 对性别和家庭所在地进行频数分析
import pandas as pd;import numpy as np
df = pd.read_csv("C:/pydata/chap03/example3_12.csv",encoding='gbk')
tab = pd.pivot_table(df,index=['性别'],columns=['家庭所在地'],
                                                    # 设置行和列变量
                margins=True, margins_name='合计',  # 添加边际和并重命名为合计
                aggfunc=len)                        # 使用长度聚集函数
tab
```

性别 \ 家庭所在地	中小城市	乡镇地区	大型城市	合计
女	18	6	11	35
男	6	4	15	25
合计	24	10	26	60

```
# 对全部学生的月生活费支出进行概括性描述
desc=example3_12.describe(include=None)             # 不包括类别变量
print('全部学生支出的概括性描述')
round(desc,4)
```

全部学生支出的概括性描述

	月生活费支出
count	60.0000
mean	1812.3333
std	320.9962
min	1100.0000
25%	1550.0000
50%	1850.0000
75%	2000.0000
max	2800.0000

```
# 按性别和家庭所在地分组计算描述统计量（使用 pd 中的 pivot_table 函数）
import pandas as pd;import numpy as np
tab = pd.pivot_table(example3_12,index=['性别','家庭所在地'],values=['月生活费支出']
                margins=True, margins_name='合计',
                aggfunc=[sum,min,max,np.median,np.mean,np.std])
                                # 分类计算总和、最小值和最大值、中位数、平均数和标准差
tab
```

性别	家庭所在地	sum 月生活费支出	min 月生活费支出	max 月生活费支出	median 月生活费支出	mean 月生活费支出	std 月生活费支出
女	中小城市	33140	1400	2360	1825	1841.111111	308.942959
	乡镇地区	10720	1300	2000	1860	1786.666667	247.521043
	大型城市	22350	1500	2800	2000	2031.818182	383.583581
男	中小城市	9960	1300	2000	1725	1660.000000	290.172363
	乡镇地区	6850	1500	1950	1700	1712.500000	246.221445
	大型城市	25720	1100	2000	1780	1714.666667	293.400425
合计		108740	1100	2800	1850	1812.333333	318.309947

注：如果函数提供的统计量不能满足需要，可以自己编写函数计算所需的描述统计量。

```
# 按性别和家庭所在地分组计算描述统计量（使用自编函数）
def my_summary(df, col=['性别']):
    df_res = pd.DataFrame()
    df_res['n'] = df.groupby(col)['月生活费支出'].count()
    df_res['平均数'] = df.groupby(col)['月生活费支出'].mean().round(3)
    df_res['中位数'] = df.groupby(col)['月生活费支出'].median()
    df_res['标准差'] = df.groupby(col)['月生活费支出'].std().round(4)
    df_res['全距'] = df.groupby(col)['月生活费支出'].apply(lambda x: x.max()-x.min())
     df_res['变异系数'] = df.groupby(col)['月生活费支出'].apply(lambda x: x.std()/x.mean())
    df_res['偏度系数'] = df.groupby(col)['月生活费支出'].skew()
    return df_res
df1=my_summary(example3_12, ['性别'])
print('按性别分组');df1
```

按性别分组

性别	n	平均数	中位数	标准差	全距	变异系数	偏度系数
女	35	1891.714	1900	331.1521	1500	0.175054	0.502824
男	25	1701.200	1780	275.4893	900	0.161938	-0.548589

```
df2=my_summary(example3_12, ['家庭所在地'])
print('按家庭所在地分组');df2
```

按家庭所在地分组

家庭所在地	n	平均数	中位数	标准差	全距	变异系数	偏度系数
中小城市	24	1795.833	1800	308.6565	1060	0.171874	0.269364
乡镇地区	10	1757.000	1860	236.0344	700	0.134339	-1.052810
大型城市	26	1848.846	1850	364.1354	1700	0.196953	0.321079

```
df3=my_summary(example3_12, ['性别', '家庭所在地'])
print('同时按性别和家庭所在地分组');df3
```

同时按性别和家庭所在地分组

性别	家庭所在地	n	平均数	中位数	标准差	全距	变异系数	偏度系数
女	中小城市	18	1841.111	1825	308.9430	960	0.167802	0.345656
	乡镇地区	6	1786.667	1860	247.5210	700	0.138538	-2.042846
	大型城市	11	2031.818	2000	383.5836	1300	0.188788	0.515560
男	中小城市	6	1660.000	1725	290.1724	700	0.174803	-0.276271
	乡镇地区	4	1712.500	1700	246.2214	450	0.143779	0.035589
	大型城市	15	1714.667	1780	293.4004	900	0.171112	-0.763562

注：使用 groupby 函数也可以对 pandas 的数据框进行分组统计。比如，按性别分组计算月生活费的平均数，代码为 example3_12.groupby('性别').mean()；按性别和家庭所在地分组计算标准差，代码为 example3_12.groupby(['性别','家庭所在地']).std()。

代码框 3-14 的结果显示，女生月生活费支出的平均数和中位数均高于男生，同时，女生月生活费支出的标准差和全距也都大于男生，相应的变异系数，$CV_{女}$=0.175 054＞$CV_{男}$=0.161 938，说明女生月生活费支出的离散程度大于男生。从分布形态看，女生月生活费支出的偏度系数是 0.502 824，为右偏分布，而男生的月生活费支出偏度系数是 -0.548 589，则为左偏分布。

从按家庭所在地分类描述的结果看，来自不同家庭的学生月生活费支出也有差异。大型城市的学生月生活费支出均值高于中小城市和乡镇地区，但中位数则是乡镇地区最高。从标准差看，乡镇地区的标准差最小。从变异系数看，$CV_{大型城市}$=0.196 953＞$CV_{中小城市}$=0.171 874＞

$CV_{乡镇地区}$=0.134 339，乡镇地区的学生月生活费支出的离散程度最小，而大型城市则最大。从分布形态看，乡镇地区学生月生活费支出的偏度系数为 −1.052 810，呈现严重的左偏分布，而大型城市和中小城市学生的月生活费支出则呈现轻微的右偏分布。

本节通过一个例子展示了数据的综合描述方法，从可视化和统计量两个角度入手，使读者了解描述性分析的思路，实际应用时读者可根据自身的需要从不同的角度进行分析。

习题

3.1　随机抽取 50 个网络购物的消费者，调查他们某月的网购金额，结果如下表所示。

1 004	602	1 540	522	878	916	1 166	1 062	344	1 200
921	990	1 309	528	838	1 492	928	1 299	1 107	981
928	1 135	789	1 018	905	935	939	729	1 802	645
1 148	877	2 270	957	840	576	1 110	570	1 253	1 133
1 416	1 380	513	1 423	1 224	289	1 247	657	1 816	1 481

（1）计算平均数、标准差、极差和四分位差。

（2）计算 10%、25%、50%、75%、90% 的分位数。

（3）计算标准分数，检测数据的离群点。

（4）计算偏度系数和峰度系数，分析网购金额的分布特征。

3.2　一种产品需要人工组装，现有 3 种可供选择的组装方法。为检验哪种方法更好，随机抽取 15 个工人，让他们分别用 3 种方法组装。下面是 15 个工人分别用 3 种方法在相同的时间内组装的产品数量（单位：个）：

方法 A 的产品数量	方法 B 的产品数量	方法 C 的产品数量
164	129	125
167	130	126
168	129	126
165	130	127
170	131	126
165	130	128
164	129	127
168	127	126
164	128	127
162	128	127

续表

方法 A 的产品数量	方法 B 的产品数量	方法 C 的产品数量
163	127	125
166	128	126
167	128	116
166	125	126
165	132	125

选择适当的图形和统计量比较 3 种方法组装产品数量的分布特点。

3.3 在大学生中随机抽取 50 名男生和 50 名女生，测得身高数据如下（单位：cm）。

男生身高		女生身高	
182.1	171.2	175.4	164.0
183.3	175.1	162.7	172.5
176.6	181.1	174.6	162.1
177.3	182.3	164.8	169.8
174.3	174.4	157.0	168.0
178.9	188.6	157.3	165.6
177.5	181.0	164.3	163.2
179.5	166.7	167.6	164.5
168.3	176.6	167.8	166.5
173.3	184.2	168.7	179.1
170.3	176.1	170.7	172.7
177.9	175.9	168.2	163.5
174.4	186.6	171.7	167.8
196.0	188.5	166.3	175.3
178.4	166.5	172.3	174.6
188.5	174.9	167.3	168.5
174.2	177.5	179.2	167.9
191.4	180.2	174.9	167.2
174.4	188.2	173.8	168.3
176.8	178.3	167.4	165.2
177.2	181.9	163.7	164.0

续表

男生身高		女生身高	
167.8	189.1	165.2	166.2
183.4	178.0	158.9	168.9
174.7	167.8	169.9	162.0
174.7	186.1	168.5	167.2

利用图形和描述统计量比较分析男女学生身高的特点和差异。

本章图片