7.1　描述性统计分析

本节中，我们将关注分析连续型变量的中心趋势、变化性和分布形状的方法。为了便于说明，我们将使用第1章中*Motor Trend* 杂志的车辆路试（mtcars）数据集。我们的关注焦点是每加仑汽油行驶英里数（mpg）、马力（hp）和车重（wt）。

```
> vars <- c("mpg", "hp", "wt")
> head(mtcars[vars])
                   mpg  hp   wt
Mazda RX4         21.0 110 2.62
Mazda RX4 Wag     21.0 110 2.88
Datsun 710        22.8  93 2.32
Hornet 4 Drive    21.4 110 3.21
Hornet Sportabout 18.7 175 3.44
Valiant           18.1 105 3.46
```

我们将首先查看所有32种车型的描述性统计量，然后按照变速箱类型（am）和汽缸数（cyl）考察描述性统计量。变速箱类型是一个以0表示自动挡、1表示手动挡来编码的二分变量，而汽缸数可为4、5或6。

7.1.1　方法云集

在描述性统计量的计算方面，R中的选择多得让人尴尬。让我们从基础安装中包含的函数开始，然后查看那些用户贡献包中的扩展函数。

对于基础安装，你可以使用summary()函数来获取描述性统计量。代码清单7-1展示了一个示例。

代码清单7-1　通过summary()计算描述性统计量

```
> summary(mtcars[vars])
      mpg              hp              wt
 Min.   :10.4   Min.   : 52.0   Min.   :1.51
 1st Qu.:15.4   1st Qu.: 96.5   1st Qu.:2.58
 Median :19.2   Median :123.0   Median :3.33
 Mean   :20.1   Mean   :146.7   Mean   :3.22
 3rd Qu.:22.8   3rd Qu.:180.0   3rd Qu.:3.61
 Max.   :33.9   Max.   :335.0   Max.   :5.42
```

summary()函数提供了最小值、最大值、四分位数和数值型变量的均值，以及因子向量和逻辑型向量的频数统计。你可以使用第5章中的apply()函数或sapply()函数计算所选择的任意描述性统计量。对于sapply()函数，其使用格式为：

```
sapply(x, FUN, options)
```

其中的*x*是你的数据框（或矩阵），*FUN*为一个任意的函数。如果指定了*options*，它们将被传递给*FUN*。你可以在这里插入的典型函数有mean、sd、var、min、max、median、length、range和quantile。函数fivenum()可返回图基五数总括（Tukey's five-number summary，即最小值、下四分位数、中位数、上四分位数和最大值）。

令人惊讶的是，基础安装并没有提供偏度和峰度的计算函数，不过你可以自行添加。代码清单7-2中的示例计算了若干描述性统计量，其中包括偏度和峰度。

代码清单7-2 通过sapply()计算描述性统计量

```
> mystats <- function(x, na.omit=FALSE){
                if (na.omit)
                    x <- x[!is.na(x)]
                m <- mean(x)
                n <- length(x)
                s <- sd(x)
                skew <- sum((x-m)^3/s^3)/n
                kurt <- sum((x-m)^4/s^4)/n - 3
                return(c(n=n, mean=m, stdev=s, skew=skew, kurtosis=kurt))

              }

> sapply(mtcars[vars], mystats)
             mpg       hp       wt
n         32.000  32.000 32.0000
mean      20.091 146.688  3.2172
stdev      6.027  68.563  0.9785
skew       0.611   0.726  0.4231
kurtosis -0.373  -0.136 -0.0227
```

对于样本中的车型，每加仑汽油行驶英里数的平均值为20.1，标准差为6.0。分布呈现右偏（偏度+0.61），并且较正态分布稍平（峰度–0.37）。如果你对数据绘图，这些特征最显而易见。请注意，如果你只希望单纯地忽略缺失值，那么应当使用sapply(mtcars[vars], mystats, na.omit=TRUE)。

扩展

若干用户贡献包都提供了计算描述性统计量的函数，其中包括Hmisc、pastecs和psych。由于这些包并未包括在基础安装中，所以需要在首次使用之前先进行安装（参考1.4节）。

Hmisc包中的describe()函数可返回变量和观测的数量、缺失值和唯一值的数目、平均值、分位数，以及五个最大的值和五个最小的值。代码清单7-3提供了一个示例。

代码清单7-3 通过Hmisc包中的describe()函数计算描述性统计量

```
> library(Hmisc)
> describe(mtcars[vars])

 3  Variables      32  Observations
---------------------------------------------------------------------------
mpg
n missing  unique  Mean    .05   .10     .25   .50    .75    .90    .95
32       0    25   20.09 12.00  14.34  15.43  19.20  22.80  30.09  31.30

lowest : 10.4 13.3 14.3 14.7 15.0, highest: 26.0 27.3 30.4 32.4 33.9
---------------------------------------------------------------------------
hp
n missing  unique    Mean    .05     .10   .2     .50   .75   .90     .95
32       0     22   146.7  63.65  66.00 96.50 123.00 180.00 243.50 253.55
```

```
lowest :  52  62  65  66  91, highest: 215 230 245 264 335
------------------------------------------------------------------------
wt
n missing  unique    Mean     .05     .10     .25     .50     .75     .90     .95
32       0      29   3.217   1.736   1.956   2.581   3.325   3.610   4.048 5.293

lowest : 1.513 1.615 1.835 1.935 2.140, highest: 3.845 4.070 5.250 5.345
5.424
------------------------------------------------------------------------
```

pastecs包中有一个名为stat.desc()的函数，它可以计算种类繁多的描述性统计量。使用格式为：

```
stat.desc(x, basic=TRUE, desc=TRUE, norm=FALSE, p=0.95)
```

其中的x是一个数据框或时间序列。若basic=TRUE（默认值），则计算其中所有值、空值、缺失值的数量，以及最小值、最大值、值域，还有总和。若desc=TRUE（同样也是默认值），则计算中位数、平均数、平均数的标准误、平均数置信度为95%的置信区间、方差、标准差以及变异系数。最后，若norm=TRUE（不是默认的），则返回正态分布统计量，包括偏度和峰度（以及它们的统计显著程度）和Shapiro－Wilk正态检验结果。这里使用了p值来计算平均数的置信区间（默认置信度为0.95）。代码清单7-4给出了一个示例。

代码清单7-4　通过pastecs包中的stat.desc()函数计算描述性统计量

```
> library(pastecs)
> stat.desc(mtcars[vars])
                 mpg       hp      wt
nbr.val        32.00   32.000  32.000
nbr.null        0.00    0.000   0.000
nbr.na          0.00    0.000   0.000
min            10.40   52.000   1.513
max            33.90  335.000   5.424
range          23.50  283.000   3.911
sum           642.90 4694.000 102.952
median         19.20  123.000   3.325
mean           20.09  146.688   3.217
SE.mean         1.07   12.120   0.173
CI.mean.0.95    2.17   24.720   0.353
var            36.32 4700.867   0.957
std.dev         6.03   68.563   0.978
coef.var        0.30    0.467   0.304
```

似乎这还不够，psych包也拥有一个名为describe()的函数，它可以计算非缺失值的数量、平均数、标准差、中位数、截尾均值、绝对中位差、最小值、最大值、值域、偏度、峰度和平均值的标准误。代码清单7-5中有一个示例。

代码清单7-5　通过psych包中的describe()计算描述性统计量

```
> library(psych)

Attaching package: 'psych'
```

```
        The following object(s) are masked from package:Hmisc :
         describe

> describe(mtcars[vars])
     var  n   mean     sd median trimmed   mad   min    max
mpg    1 32  20.09   6.03  19.20   19.70  5.41 10.40  33.90
hp     2 32 146.69 68.56 123.00  141.19 77.10 52.00 335.00
wt     3 32   3.22   0.98   3.33    3.15  0.77  1.51   5.42
      range skew kurtosis    se
mpg  23.50 0.61    -0.37  1.07
hp  283.00 0.73    -0.14 12.12
wt    3.91 0.42    -0.02  0.17
```

一语中的，选择多得简直让人尴尬！

注意 在前面的示例中，psych包和Hmisc包均提供了名为describe()的函数。R如何知道该使用哪个呢？简言之，如代码清单7-5所示，最后载入的程序包优先。在这里，psych在Hmisc之后被载入，然后显示了一条信息，提示Hmisc包中的describe()函数被psych包中的同名函数所屏蔽（masked）。键入describe()后，R在搜索这个函数时将首先找到psych包中的函数并执行它。如果你想改而使用Hmisc包中的版本，可以键入Hmisc::describe(mt)。这个函数仍然在那里。你只是需要给予R更多信息以找到它。

你已经了解了如何为整体的数据计算描述性统计量，现在让我们看看如何获取数据中各组的统计量。

7.1.2 分组计算描述性统计量

在比较多组个体或观测时，关注的焦点经常是各组的描述性统计信息，而不是样本整体的描述性统计信息。同样地，在R中完成这个任务有若干种方法。我们将以获取变速箱类型各水平的描述性统计量开始。

在第5章中，我们讨论了整合数据的方法。你可以使用aggregate()函数（5.6.2节）来分组获取描述性统计量，如代码清单7-6所示。

代码清单7-6 使用aggregate()分组获取描述性统计量

```
> aggregate(mtcars[vars], by=list(am=mtcars$am), mean)
  am  mpg  hp   wt
1  0 17.1 160 3.77
2  1 24.4 127 2.41
> aggregate(mtcars[vars], by=list(am=mtcars$am), sd)
  am  mpg   hp    wt
1  0 3.83 53.9 0.777
2  1 6.17 84.1 0.617
```

注意list(am=mtcars$am)的使用。如果使用的是list(mtcars$am)，则am列将被标注为Group.1而不是am。你使用这个赋值指定了一个更有帮助的列标签。如果有多个分组变量，可以

使用by=list(*name1=groupvar1, name2=groupvar2, ... , groupvarN*)这样的语句。

遗憾的是，aggregate()仅允许在每次调用中使用平均数、标准差这样的单返回值函数。它无法一次返回若干个统计量。要完成这项任务，可以使用by()函数。格式为：

```
by(data, INDICES, FUN)
```

其中*data*是一个数据框或矩阵，*INDICES*是一个因子或因子组成的列表，定义了分组，*FUN*是任意函数。代码清单7-7提供了一个示例。

代码清单7-7　使用by()分组计算描述性统计量

```
> dstats <- function(x)(c(mean=mean(x), sd=sd(x)))
> by(mtcars[vars], mtcars$am, dstats)

mtcars$am: 0
mean.mpg  mean.hp  mean.wt   sd.mpg    sd.hp    sd.wt
  17.147  160.263    3.769    3.834   53.908    0.777
------------------------------------------------
mtcars$am: 1
mean.mpg  mean.hp  mean.wt   sd.mpg    sd.hp    sd.wt
  24.392  126.846    2.411    6.167   84.062    0.617
```

扩展

doBy包和psych包也提供了分组计算描述性统计量的函数。同样地，它们未随基本安装发布，必须在首次使用前进行安装。doBy包中summaryBy()函数的使用格式为：

```
summaryBy(formula, data=dataframe, FUN=function)
```

其中的*formula*接受以下的格式：

```
var1 + var2 + var3 + … + varN ~ groupvar1 + groupvar2 + … + groupvarN
```

在~左侧的变量是需要分析的数值型变量，而右侧的变量是类别型的分组变量。*function*可为任何内建或用户自编的R函数。使用7.2.1节中创建的mystats()函数的一个示例如代码清单7-8所示。

代码清单7-8　使用doBy包中的summaryBy()分组计算概述统计量

```
> library(doBy)
> summaryBy(mpg+hp+wt~am, data=mtcars, FUN=mystats)
  am mpg.n mpg.mean mpg.stdev mpg.skew mpg.kurtosis hp.n hp.mean hp.stdev
1  0    19     17.1      3.83   0.0140       -0.803   19     160      53.9
2  1    13     24.4      6.17   0.0526       -1.455   13     127      84.1
  hp.skew hp.kurtosis wt.n wt.mean wt.stdev wt.skew wt.kurtosis
1 -0.0142      -1.210   19    3.77    0.777   0.976       0.142
2  1.3599       0.563   13    2.41    0.617   0.210      -1.174
```

psych包中的describe.by()函数可计算和describe相同的描述性统计量，只是按照一个或多个分组变量分层，如代码清单7-9所示。

代码清单7-9 使用psych包中的describe.by()分组计算概述统计量

```
> library(psych)
> describe.by(mtcars[vars], mtcars$am)
group: 0
    var  n   mean    sd median trimmed   mad   min    max
mpg   1 19  17.15  3.83  17.30   17.12  3.11 10.40  24.40
hp    2 19 160.26 53.91 175.00  161.06 77.10 62.00 245.00
wt    3 19   3.77  0.78   3.52    3.75  0.45  2.46   5.42
     range  skew kurtosis    se
mpg  14.00  0.01    -0.80  0.88
hp  183.00  0.01     1.21 12.37
wt    2.96  0.98     0.14  0.18
------------------------------------------------

group: 1
    var  n   mean    sd median trimmed   mad   min    max
mpg   1 13  24.39  6.17  22.80   24.38  6.67 15.00  33.90
hp    2 13 126.85 84.06 109.00  114.73 63.75 52.00 335.00
wt    3 13   2.41  0.62   2.32    2.39  0.68  1.51   3.57
     range skew kurtosis    se
mpg  18.90 0.05    -1.46  1.71
hp  283.00 1.36     0.56 23.31
wt    2.06 0.21    -1.17  0.17
```

与前面的示例不同，describe.by()函数不允许指定任意函数，所以它的普适性较低。若存在一个以上的分组变量，你可以使用list(*groupvar1*, *groupvar2*, ... , *groupvarN*)来表示它们。但这仅在分组变量交叉后不出现空白单元时有效。

最后，可以使用5.6.3节中描述的reshape包灵活地按组导出描述性统计量。（如果你尚未阅读那一节，建议在继续往下读之前先看看它。）首先，使用：

```
dfm <- melt(dataframe, measure.vars=y, id.vars=g)
```

融合数据框。其中的*dataframe*包含着数据，*y*是一个向量，指明了要进行概述的数值型变量（默认使用所有变量），而*g*是由一个或多个分组变量组成的向量。然后使用：

```
cast(dfm, groupvar1 + groupvar2 + … + variable ~ ., FUN)
```

重铸数据。分组变量以+号分隔，这里的variable只取其字面含义[①]，而*FUN*是一个任意函数。

在本节的最后一个例子中，我们将运用数据重塑的方法来取得由变速箱类型与汽缸数形成的每个亚组的描述性统计量。我们要获取的描述性统计量是样本大小、平均数和标准差。代码和结果如代码清单7-10所示。

代码清单7-10 通过reshape包分组计算概述统计量

```
> library(reshape)
> dstats <- function(x)(c(n=length(x), mean=mean(x), sd=sd(x)))
> dfm <- melt(mtcars, measure.vars=c("mpg", "hp", "wt"),
```

① 即仅表示重铸后数据框中的变量variable。——译者注

```
              id.vars=c("am", "cyl"))
> cast(dfm, am + cyl + variable ~ ., dstats)

   am cyl variable  n   mean     sd
1   0   4      mpg  3  22.90  1.453
2   0   4       hp  3  84.67 19.655
3   0   4       wt  3   2.94  0.408
4   0   6      mpg  4  19.12  1.632
5   0   6       hp  4 115.25  9.179
6   0   6       wt  4   3.39  0.116
7   0   8      mpg 12  15.05  2.774
8   0   8       hp 12 194.17 33.360
9   0   8       wt 12   4.10  0.768
10  1   4      mpg  8  28.07  4.484
11  1   4       hp  8  81.88 22.655
12  1   4       wt  8   2.04  0.409
13  1   6      mpg  3  20.57  0.751
14  1   6       hp  3 131.67 37.528
15  1   6       wt  3   2.75  0.128
16  1   8      mpg  2  15.40  0.566
17  1   8       hp  2 299.50 50.205
18  1   8       wt  2   3.37  0.283
```

我个人认为这种方式最为简洁动人。数据分析人员对于展示哪些描述性统计量以及结果采用什么格式都有着自己的偏好，这也许就是有如此多不同方法的原因。你可以选择最适合的方式，或是创造属于自己的方法！

7.1.3　结果的可视化

分布特征的数值刻画的确很重要，但是这并不能代替视觉呈现。对于定量变量，我们有直方图（6.3节）、密度图（6.4节）、箱线图（6.5节）和点图（6.6节）。它们都可以让我们洞悉那些依赖于观察一小部分描述性统计量时忽略的细节。

目前我们考虑的函数都是为定量变量提供概述的。下一节中的函数则允许考察类别型变量的分布。

7.2　频数表和列联表

在本节中，我们将着眼于类别型变量的频数表和列联表，以及相应的独立性检验、相关性的度量、图形化展示结果的方法。我们除了使用基础安装中的函数，还将连带使用vcd包和gmodels包中的函数。下面的示例中，假设A、B和C代表类别型变量。

本节中的数据来自vcd包中的Arthritis数据集。这份数据来自Kock & Edward（1988），表示了一项风湿性关节炎新疗法的双盲临床实验的结果。前几个观测是这样的：

```
> library(vcd)
> head(Arthritis)
  ID Treatment  Sex Age Improved
```