```
              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
```

```
1 57   Treated Male  27     Some
2 46   Treated Male  29     None
3 77   Treated Male  30     None
4 17   Treated Male  32   Marked
5 36   Treated Male  46   Marked
6 23   Treated Male  58   Marked
```

治疗情况（安慰剂治疗、用药治疗）、性别（男性、女性）和改善情况（无改善、一定程度的改善、显著改善）均为类别型因子[①]。下一节中，我们将使用此数据创建频数表和列联表（交叉的分类）。

7.2.1 生成频数表

R中提供了用于创建频数表和列联表的若干种方法。其中最重要的函数已列于表7-1中。

表7-1 用于创建和处理列联表的函数

函 数	描 述
table(*var1*, *var2*, …, *varN*)	使用N个类别型变量（因子）创建一个N维列联表
xtabs(*formula*, *data*)	根据一个公式和一个矩阵或数据框创建一个N维列联表
prop.table(*table*, *margins*)	依*margins*定义的边际列表将表中条目表示为分数形式
margin.table(*table*, *margins*)	依*margins*定义的边际列表计算表中条目的和
addmargins(*table*, *margins*)	将概述边*margins*（默认是求和结果）放入表中
ftable(*table*)	创建一个紧凑的“平铺”式列联表

接下来，我们将逐个使用以上函数来探索类别型变量。我们首先考察简单的频率表，接下来是二维列联表，最后是多维列联表。第一步是使用table()或xtabs()函数创建一个表，然后使用其他函数处理它。

1. 一维列联表

可以使用table()函数生成简单的频数统计表。示例如下：

```
> mytable <- with(Arthritis, table(Improved))
> mytable
Improved
  None   Some Marked
    42     14     28
```

可以用prop.table()将这些频数转化为比例值：

```
> prop.table(mytable)
Improved
  None   Some Marked
 0.500  0.167  0.333
```

或使用prop.table()*100转化为百分比：

① 分别对应数据中的变量*Treatment*（*Placebo*、*Treated*）、*Sex*（*Male*、*Female*）和*Improved*（*None*、*Some*、*Marked*）。

——译者注

```
> prop.table(mytable)*100
Improved
  None   Some Marked
  50.0   16.7   33.3
```

这里可以看到，有50%的研究参与者获得了一定程度或者显著的改善（16.7 + 33.3）。

2. 二维列联表

对于二维列联表，table()函数的使用格式为：

```
mytable <- table(A, B)
```

其中的A是行变量，B是列变量。除此之外，xtabs()函数还可使用公式风格的输入创建列联表，格式为：

```
mytable <- xtabs(~ A + B, data=mydata)
```

其中的mydata是一个矩阵或数据框。总的来说，要进行交叉分类的变量应出现在公式的右侧（即~符号的右方），以+作为分隔符。若某个变量写在公式的左侧，则其为一个频数向量（在数据已经被表格化时很有用）。

对于Arthritis数据，有：

```
> mytable <- xtabs(~ Treatment+Improved, data=Arthritis)
> mytable
         Improved
Treatment None Some Marked
  Placebo   29    7      7
  Treated   13    7     21
```

你可以使用margin.table()和prop.table()函数分别生成边际频数和比例。行和与行比例可以这样计算：

```
> margin.table(mytable, 1)
Treatment
Placebo Treated
     43      41
> prop.table(mytable, 1)
         Improved
Treatment  None  Some Marked
  Placebo 0.674 0.163  0.163
  Treated 0.317 0.171  0.512
```

下标1指代table()语句中的第一个变量。观察表格可以发现，与接受安慰剂的个体中有显著改善的16%相比，接受治疗的个体中的51%的个体病情有了显著的改善。

列和与列比例可以这样计算：

```
> margin.table(mytable, 2)
Improved
  None   Some Marked
    42     14     28
> prop.table(mytable, 2)
         Improved
Treatment  None  Some Marked
Placebo 0.690 0.500  0.250
Treated 0.310 0.500  0.750
```

这里的下标2指代table()语句中的第二个变量。

各单元格所占比例可用如下语句获取：

```
> prop.table(mytable)
          Improved
Treatment    None    Some Marked
  Placebo 0.3452 0.0833 0.0833
  Treated 0.1548 0.0833 0.2500
```

你可以使用addmargins()函数为这些表格添加边际和。例如，以下代码添加了各行的和与各列的和：

```
> addmargins(mytable)
          Improved
Treatment None Some Marked Sum
  Placebo   29    7      7  43
  Treated   13    7     21  41
  Sum       42   14     28  84
> addmargins(prop.table(mytable))
          Improved
Treatment    None    Some Marked     Sum
  Placebo 0.3452 0.0833 0.0833 0.5119
  Treated 0.1548 0.0833 0.2500 0.4881
  Sum     0.5000 0.1667 0.3333 1.0000
```

在使用addmargins()时，默认行为是为表中所有的变量创建边际和。作为对照：

```
> addmargins(prop.table(mytable, 1), 2)
          Improved
Treatment  None  Some Marked   Sum
  Placebo 0.674 0.163  0.163 1.000
  Treated 0.317 0.171  0.512 1.000
```

仅添加了各行的和。类似地，

```
> addmargins(prop.table(mytable, 2), 1)
          Improved
Treatment  None  Some Marked
  Placebo 0.690 0.500  0.250
  Treated 0.310 0.500  0.750
  Sum     1.000 1.000  1.000
```

添加了各列的和。在表中可以看到，有显著改善患者中的25%是接受安慰剂治疗的。

注意 table()函数默认忽略缺失值（NA）。要在频数统计中将NA视为一个有效的类别，请设定参数useNA="ifany"。

使用gmodels包中的CrossTable()函数是创建二维列联表的第三种方法。CrossTable()函数仿照SAS中PROC FREQ或SPSS中CROSSTABS的形式生成二维列联表。示例参阅代码清单7-11。

代码清单7-11　使用CrossTable生成二维列联表

```
> library(gmodels)
> CrossTable(Arthritis$Treatment, Arthritis$Improved)

   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  84

                    | Arthritis$Improved
Arthritis$Treatment |       None |       Some |     Marked | Row Total |
--------------------|------------|------------|------------|-----------|
            Placebo |         29 |          7 |          7 |        43 |
                    |      2.616 |      0.004 |      3.752 |           |
                    |      0.674 |      0.163 |      0.163 |     0.512 |
                    |      0.690 |      0.500 |      0.250 |           |
                    |      0.345 |      0.083 |      0.083 |           |
--------------------|------------|------------|------------|-----------|
            Treated |         13 |          7 |         21 |        41 |
                    |      2.744 |      0.004 |      3.935 |           |
                    |      0.317 |      0.171 |      0.512 |     0.488 |
                    |      0.310 |      0.500 |      0.750 |           |
                    |      0.155 |      0.083 |      0.250 |           |
--------------------|------------|------------|------------|-----------|
       Column Total |         42 |         14 |         28 |        84 |
                    |      0.500 |      0.167 |      0.333 |           |
--------------------|------------|------------|------------|-----------|
```

CrossTable()函数有很多选项，可以做许多事情：计算（行、列、单元格）的百分比；指定小数位数；进行卡方、Fisher和McNemar独立性检验；计算期望和（皮尔逊、标准化、调整的标准化）残差；将缺失值作为一种有效值；进行行和列标题的标注；生成SAS或SPSS风格的输出。参阅help(CrossTable)以了解详情。

如果有两个以上的类别型变量，那么你就是在处理多维列联表。我们将在下面考虑这种情况。

3. 多维列联表

table()和xtabs()都可以基于三个或更多的类别型变量生成多维列联表。margin.table()、prop.table()和addmargins()函数可以自然地推广到高于二维的情况。另外，ftable()函数可以以一种紧凑而吸引人的方式输出多维列联表。代码清单7-12中给出了一个示例。

代码清单7-12 三维列联表

```
> mytable <- xtabs(~ Treatment+Sex+Improved, data=Arthritis)
> mytable
, , Improved = None

         Sex
Treatment Female Male
  Placebo     19   10
  Treated      6    7

, , Improved = Some

         Sex
Treatment Female Male
  Placebo      7    0
  Treated      5    2

, , Improved = Marked

         Sex
Treatment Female Male
  Placebo      6    1
  Treated     16    5

> ftable(mytable)
                  Sex Female Male
Treatment Improved
Placebo   None             19   10
          Some              7    0
          Marked            6    1
Treated   None              6    7
          Some              5    2
          Marked           16    5

> margin.table(mytable, 1)
Treatment
Placebo Treated
     43      41
> margin.table(mytable, 2)
Sex
Female   Male
    59     25
> margin.table(mytable, 3)
Improved
  None   Some Marked
    42     14     28
> margin.table(mytable, c(1, 3))
         Improved
Treatment None Some Marked
  Placebo   29    7      7
  Treated   13    7     21
> ftable(prop.table(mytable, c(1, 2)))
                 Improved  None  Some Marked
Treatment Sex
```

❶ 各单元格的频数

❷ 边际频数

❸ 治疗情况（**Treatment**）× 改善情况（**Improved**）的边际频数

❹ 治疗情况（**Treatment**）× 性别（**Sex**）的各类改善情况比例

```
Placebo   Female          0.594 0.219  0.188
          Male            0.909 0.000  0.091
Treated   Female          0.222 0.185  0.593
          Male            0.500 0.143  0.357

> ftable(addmargins(prop.table(mytable, c(1, 2)), 3))
                  Improved  None  Some Marked   Sum
Treatment Sex
Placebo   Female          0.594 0.219  0.188 1.000
          Male            0.909 0.000  0.091 1.000
Treated   Female          0.222 0.185  0.593 1.000
          Male            0.500 0.143  0.357 1.000
```

第❶部分代码生成了三维分组各单元格的频数。这段代码同时演示了如何使用ftable()函数输出更为紧凑和吸引人的表格。

第❷部分代码为治疗情况（Treatment）、性别（Sex）和改善情况（Improved）生成了边际频数。由于使用公式~Treatement+Sex+Improve创建了这个表，所以Treatment需要通过下标1来引用，Sex通过下标2来引用，Improve通过下标3来引用。

第❸部分代码为治疗情况（Treatment） × 改善情况（Improved）分组的边际频数，由不同性别（Sex）的单元加和而成。每个Treatment × Sex组合中改善情况为None、Some和Marked患者的比例由❹给出。在这里可以看到治疗组的男性中有36%有了显著改善，女性为59%。总而言之，比例将被添加到不在prop.table()调用中的下标上（本例中是第三个下标，或称Improve）。在最后一个例子中可以看到这一点，你在那里为第三个下标添加了边际和。

如果想得到百分比而不是比例，可以将结果表格乘以100。例如：

```
ftable(addmargins(prop.table(mytable, c(1, 2)), 3)) * 100
```

将生成下表：

```
                     Sex Female  Male   Sum
Treatment Improved
Placebo   None            65.5  34.5 100.0
          Some           100.0   0.0 100.0
          Marked          85.7  14.3 100.0
Treated   None            46.2  53.8 100.0
          Some            71.4  28.6 100.0
          Marked          76.2  23.8 100.0
```

列联表可以告诉你组成表格的各种变量组合的频数或比例，不过你可能还会对列联表中的变量是否相关或独立感兴趣。下一节我们会讲解独立性的检验。

7.2.2　独立性检验

R提供了多种检验类别型变量独立性的方法。本节中描述的三种检验分别为卡方独立性检验、Fisher精确检验和Cochran-Mantel－Haenszel检验。

1. 卡方独立性检验

你可以使用chisq.test()函数对二维表的行变量和列变量进行卡方独立性检验。示例参见代码清单7-13。

代码清单7-13 卡方独立性检验

```
> library(vcd)
> mytable <- xtabs(~Treatment+Improved, data=Arthritis)
> chisq.test(mytable)

        Pearson's Chi-squared test

data:  mytable
X-squared = 13.1, df = 2, p-value = 0.001463

> mytable <- xtabs(~Improved+Sex, data=Arthritis)
> chisq.test(mytable)

        Pearson's Chi-squared test

data:  mytable
X-squared = 4.84, df = 2, p-value = 0.0889

Warning message:
In chisq.test(mytable) : Chi-squared approximation may be incorrect
```

❶ 治疗情况和改善情况不独立

❷ 性别和改善情况独立

在结果❶中，患者接受的治疗和改善的水平看上去存在着某种关系（$p < 0.01$）。而患者性别和改善情况之间却不存在关系（$p > 0.05$）❷。这里的p值表示从总体中抽取的样本行变量与列变量是相互独立的概率。由于❶的概率值很小，所以你拒绝了治疗类型和治疗结果相互独立的原假设。由于❷的概率不够小，故没有足够的理由说明治疗结果和性别之间是不独立的。代码清单7-13中产生警告信息的原因是，表中的6个单元格之一（男性 - 一定程度上的改善）有一个小于5的值，这可能会使卡方近似无效。

2. Fisher精确检验

可以使用`fisher.test()`函数进行Fisher精确检验。Fisher精确检验的原假设是：边界固定的列联表中行和列是相互独立的。其调用格式为`fisher.test(`*`mytable`*`)`，其中的*`mytable`*是一个二维列联表。示例如下：

```
> mytable <- xtabs(~Treatment+Improved, data=Arthritis)
> fisher.test(mytable)
        Fisher's Exact Test for Count Data

data:  mytable
p-value = 0.001393
alternative hypothesis: two.sided
```

与许多统计软件不同的是，这里的`fisher.test()`函数可以在任意行列数大于等于2的二维列联表上使用，但不能用于2×2的列联表。

3. Cochran-Mantel–Haenszel检验

`mantelhaen.test()`函数可用来进行Cochran–Mantel–Haenszel卡方检验，其原假设是，两个名义变量在第三个变量的每一层中都是条件独立的。下列代码可以检验治疗情况和改善情况在性别的每一水平下是否独立。此检验假设不存在三阶交互作用（治疗情况 × 改善情况 × 性别）。

```
> mytable <- xtabs(~Treatment+Improved+Sex, data=Arthritis)
> mantelhaen.test(mytable)
```

```
        Cochran-Mantel-Haenszel test

data:  mytable
Cochran-Mantel-Haenszel M^2 = 14.6, df = 2, p-value = 0.0006647
```

结果表明，患者接受的治疗与得到的改善在性别的每一水平下并不独立（即，分性别来看，用药治疗的患者较接受安慰剂的患者有了更多的改善）。

7.2.3　相关性的度量

上一节中的显著性检验评估了是否存在充分的证据以拒绝变量间相互独立的原假设。如果可以拒绝原假设，那么你的兴趣就会自然而然地转向用以衡量相关性强弱的相关性度量。vcd包中的assocstats()函数可以用来计算二维列联表的phi系数、列联系数和Cramer's V系数。代码清单7-14给出了一个示例。

代码清单7-14　二维列联表的相关性度量

```
> library(vcd)
> mytable <- xtabs(~Treatment+Improved, data=Arthritis)
> assocstats(mytable)

                    X^2 df  P(> X^2)
Likelihood Ratio 13.530  2 0.0011536
Pearson          13.055  2 0.0014626

Phi-Coefficient   : 0.394
Contingency Coeff.: 0.367
Cramer's V        : 0.394
```

总体来说，较大的值意味着较强的相关性。vcd包也提供了一个kappa()函数，可以计算混淆矩阵的Cohen's kappa值以及加权的kappa值。（举例来说，混淆矩阵可以表示两位评判者对于一系列对象进行分类所得结果的一致程度。）

7.2.4　结果的可视化

R中拥有远远超出其他多数统计软件的、可视地探索类别型变量间关系的方法。通常，我们会使用条形图进行一维频数的可视化（参见6.1节）。vcd包中拥有优秀的、用于可视化多维数据集中类别型变量间关系的函数，可以绘制马赛克图和关联图（参见11.4节）。最后，ca包中的对应分析函数允许使用多种几何表示（Nenadic & Greenacre，2007）可视地探索列联表中行和列之间的关系。

7.2.5　将表转换为扁平格式

我们将以一个在其他R书籍中极少涵盖但又非常实用的话题结束本节。在你已经拥有一个列联表却需要原始的数据时怎么办？举例来说，假设有以下列联表：

```
                    Sex Female Male
Treatment Improved
Placebo   None             19   10
          Some              7    0
          Marked            6    1
Treated   None              6    7
          Some              5    2
          Marked           16    5
```

但需要的是这种格式：

```
  ID Treatment  Sex Age Improved
1 57   Treated Male  27     Some
2 46   Treated Male  29     None
3 77   Treated Male  30     None
4 17   Treated Male  32   Marked
5 36   Treated Male  46   Marked
6 23   Treated Male  58   Marked
[78 more rows go here]
```

R中的许多统计函数接受的是后一种格式而不是前一种格式。你可以使用代码清单7-15中提供的函数将R中的表转换回扁平的数据格式。

代码清单7-15 通过table2flat将表转换为扁平格式

```
table2flat <- function(mytable) {
  df <- as.data.frame(mytable)
  rows  <- dim(df)[1]
  cols  <- dim(df)[2]
  x <- NULL
  for (i in 1:rows){
    for (j in 1:df$Freq[i]){
      row <- df[i,c(1:(cols-1))]
      x <- rbind(x,row)
    }
  }
  row.names(x)<-c(1:dim(x)[1])
  return(x)
}
```

此函数可以接受一个R中的表格（行列数任意）并返回一个扁平格式的数据框。你也可以使用这个函数来输入已发表的研究中的表格。举例来说，假设你在某期刊上看到了表7-2，并想以扁平格式将其保存到R中。

表7-2 Arthritis数据集中治疗情况和改善情况的列联表

治疗情况	改善情况		
	无改善	一定程度的改善	显著改善
安慰剂治疗	29	7	7
用药治疗	13	17	21

代码清单7-16描述了一种可以解决这个问题的方法。

代码清单7-16　使用table2flat()函数转换已发表的数据

```
> treatment <- rep(c("Placebo", "Treated"), times=3)
> improved <- rep(c("None", "Some", "Marked"), each=2)
> Freq <- c(29,13,7,17,7,21)
> mytable <- as.data.frame(cbind(treatment, improved, Freq))
> mydata <- table2flat(mytable)
> head(mydata)
  treatment improved
1   Placebo     None
2   Placebo     None
3   Placebo     None
4   Treated     None
5   Placebo     Some
6   Placebo     Some
[12 more rows go here]
```

对列联表的讨论暂时到此为止，我们将在第11章和第15章中探讨更多高级话题。下面，我们将开始关注各种类型的相关系数。

7.3　相关

相关系数可以用来描述定量变量之间的关系。相关系数的符号（±）表明关系的方向（正相关或负相关），其值的大小表示关系的强弱程度（完全不相关时为0，完全相关时为1）。

本节中，我们将关注多种相关系数和相关性的显著性检验。我们将使用R基础安装中的state.x77数据集，它提供了美国50个州在1977年的人口、收入、文盲率、预期寿命、谋杀率和高中毕业率数据。数据集中还收录了气温和土地面积数据，但为了节约空间，这里将其丢弃。你可以使用help(state.x77)了解数据集的更多信息。除了基础安装以外，我们还将使用psych和ggm包。

7.3.1　相关的类型

R可以计算多种相关系数，包括Pearson相关系数、Spearman相关系数、Kendall相关系数、偏相关系数、多分格（polychoric）相关系数和多系列（polyserial）相关系数。下面让我们依次理解这些相关系数。

1. Pearson、Spearman和Kendall相关

Pearson积差相关系数衡量了两个定量变量之间的线性相关程度。Spearman等级相关系数则衡量分级定序变量之间的相关程度。Kendall's Tau相关系数也是一种非参数的等级相关度量。

cor()函数可以计算这三种相关系数，而cov()函数可用来计算协方差。两个函数的参数有很多，其中与相关系数的计算有关的参数可以简化为：

```
cor(x, use= , method= )
```

这些参数详述于表7-3中。