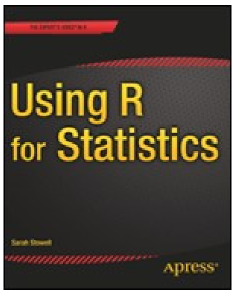

Using R for Statistics

by Sarah Stowell

Chapter 6: Tabular Data

Overview

This chapter explains how to summarize and calculate statistics for categorical variables. Categorical data is normally summarized in a frequency table (for one variable) or contingency table (for two or more variables), which is a table showing the number of values in each category or in each combination of categories. Data summarized in this form is known as tabular data.

As well as summarizing your categorical data in a table, you may need to compare it with a hypothesized distribution using the chi-square goodness-of-fit test. You may also want to determine whether there is any association between two or more categorical variables. The chi-square test of association and the Fisher's exact test are two methods that can help you with this.

You will learn how to:

- create frequency and contingency tables to summarise categorical data, and how to store the results as a table object
- display your table with percentages or marginal sums
- perform a chi-square goodness-of-fit test
- perform a chi-square test of association to identify a relationship between two or more categorical variables
- perform Fisher's exact test of association for a two-by-two contingency table
- perform a test of proportions

This chapter uses the `warpbreaks` and `esoph` datasets and the `Titanic` table object, which are included with R. You can view more information about them by entering `help(`*`datasetname`*`)`. It also uses the `people2` dataset (which is a modified version of the `people` dataset introduced in Chapter 3), and the `apartments` dataset. These are both available with the downloads for this book and described in Appendix C.

Frequency Tables

Frequency tables summarize a categorical variable by displaying the number of observations belonging to each category. Frequency tables for two or more categorical variables (known as contingency tables or cross tabs) summarize the relationship between two or more categorical variables by displaying the number of observations that fall into each combination of categories.

In R, a table is also a type of object that holds tabulated data. There are some example table objects included with R, such as `HairEyeColor` and `Titanic`.

Creating Tables

To create a one-dimensional frequency table showing the number of observations for each level of a factor variable, use the `table` function:

```
> table(people2$Eye.Color)
```

```
Blue Brown Green
   7     6     3
```

Whereas R allows you to create tables from any type of variable, they are only really meaningful for factor variables with a relatively small number of values. If you want to include a continuous variable, first divide it into categories with the `cut` function, as explained in Chapter 3 under "Dividing a Continuous Variable into Categories." To add an additional column to the table showing the numbers of missing values (if there are any), set the `useNA` argument:

```
> table(dataset$factor1, useNA="ifany")
```

To create a two-dimensional contingency table, give two variables as input:

```
> table(people2$Eye.Color, people2$Sex)
```

```
      Male Female
Blue     4      3
Brown    5      1
Green    1      2
```

Similarly, you can create contingency tables for three or more variables:

```
> table(people2$Eye.Color, people2$Height.Cat, people2$Sex)
```

```
, , = Male
```

```
       Medium Short Tall
Blue        2     0    2
Brown       4     0    1
Green       0     0    1

, , = Female


       Medium Short Tall
Blue        3     0    0
Brown       0     0    0
Green       0     2    0
```

To save a table as a table object, assign the output of the `table` function to a new object name:

```
> sexeyetable<-table(people2$Eye.Color, people2$Sex)
```

Once you have created a table object, you can apply further functions to the object to create output that is relevant to tabular data. For example, you could use the `pie` function to create a pie chart showing the proportion of people with each eye color, or use the `summary` function to test for association between eye color and sex. You will learn about some of these functions later in this chapter. Save the `sexeyetable` object, as it will be useful later on.

Displaying Tables

There are a few functions that allow you to present table objects in different ways. These include `ftable`, `prop.table`, and `addmargins`.

The `ftable` function displays your table in a more compact way, which is useful for tables with three or more dimensions:

```
> ftable(Titanic)
```

```
                   Survived  No Yes
Class Sex    Age
1st   Male   Child            0   5
             Adult          118  57
      Female Child            0   1
             Adult            4 140
2nd   Male   Child            0  11
             Adult          154  14
      Female Child            0  13
             Adult           13  80
3rd   Male   Child           35  13
             Adult          387  75
      Female Child           17  14
             Adult           89  76
Crew  Male   Child            0   0
             Adult          670 192
      Female Child            0   0
             Adult            3  20
```

The `prop.table` function displays the table with each cell count expressed as a proportion of the total count:

```
> prop.table(sexeyetable)
```

```
        Male Female
Blue  0.2500 0.1875
Brown 0.3125 0.0625
Green 0.0625 0.1250
```

To display the cell counts expressed as a proportion of the row or column totals instead of the grand total, set the `margin` argument to 1 for rows, 2 for columns, and 3+ for higher dimensions:

```
> prop.table(sexeyetable, margin=2)
```

```
           Male    Female
Blue  0.4000000 0.5000000
Brown 0.5000000 0.1666667
Green 0.1000000 0.3333333
```

To display percentages, multiply the whole table by 100. You can also use the `round` function to round all of the numbers in the table:

```
> round(prop.table(sexeyetable)*100)
```

```
      Male Female
Blue    25     19
Brown   31      6
Green    6     12
```

The `addmargins` function displays your table with row and column totals:

```
> addmargins(sexeyetable)
```

```
      Male Female Sum
Blue     4      3   7
Brown    5      1   6
Green    1      2   3
Sum     10      6  16
```

To add margins to just one dimension of the table, set the `margin` argument to 1 for rows, 2 for columns, and 3+ for higher dimensions:

```
> addmargins(sexeyetable, margin=1)
```

```
      Male Female
Blue     4      3
Brown    5      1
Green    1      2
Sum     10      6
```

Creating Tables from Count Data

So far, you have created tables by counting each row in the dataset as one observation. However, occasionally you may have a dataset in which count data has already been aggregated, such as the `warpbreaks` dataset (included with R).

To create a table from a data frame containing count data, use the `xtabs` function:

```
> xtabs(breaks~wool+tension, warpbreaks)
```

```
    tension
wool   L   M   H
   A 401 216 221
   B 254 259 169
```

If the data frame has more than one column of count data, put them inside the `list` function:

```
> xtabs(list(counts1, counts2)~factor1, dataset)
```

You can create a table object by assigning the output of the `xtabs` function to a new object name:

```
> warpbreakstable<-xtabs(breaks~wool+tension, warpbreaks)
```

To create a data frame of count data from a table object, use the `as.data.frame` function:

```
> as.data.frame(Titanic)
```

```
   Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
4   Crew   Male Child       No    0
5    1st Female Child       No    0
6    2nd Female Child       No    0
7    3rd Female Child       No   17
8   Crew Female Child       No    0
9    1st   Male Adult       No  118
10   2nd   Male Adult       No  154
11   3rd   Male Adult       No  387
12  Crew   Male Adult       No  670
```

```
13   1st Female Adult       No    4
14   2nd Female Adult       No   13
15   3rd Female Adult       No   89
16  Crew Female Adult       No    3
17   1st   Male Child      Yes    5
18   2nd   Male Child      Yes   11
19   3rd   Male Child      Yes   13
20  Crew   Male Child      Yes    0
21   1st Female Child      Yes    1
22   2nd Female Child      Yes   13
23   3rd Female Child      Yes   14
24  Crew Female Child      Yes    0
25   1st   Male Adult      Yes   57
26   2nd   Male Adult      Yes   14
27   3rd   Male Adult      Yes   75
28  Crew   Male Adult      Yes  192
29   1st Female Adult      Yes  140
30   2nd Female Adult      Yes   80
31   3rd Female Adult      Yes   76
32  Crew Female Adult      Yes   20
```

Creating a Table Directly

Sometimes you may not have a dataset at all, only a table of counts. In order to be able to perform analysis such as the chi-square test (covered later in this chapter), you will need to enter your data into R as a table object.

You can create a one-dimensional table object with the `as.table` function. Enter the counts for each of the categories as shown here:

```
> table1D<-as.table(c(5, 21, 17, 3, 1))
```

When you view the table, you can see that R has given the categories default names of A, B, and so on.

```
> table1D
```

```
A  B  C D E
5 21 17 3 1
```

To overwrite the default names with the correct category names, use the `row.names` function:

```
> row.names(table1D)<-c("Category 1", "Category 2", "Category 3", "Category 4", "Category 5")
```

You can also enter a two-dimensional table into R in a similar way. Suppose that you want to enter the data shown in Table 6-1.

Table 6-1: Number of subjects with and without a disease after treatment with either an active treatment or a control

		Disease Status	
		Resolved	*Unresolved*
Group	*Active Treatment Group*	15	11
	Control Group	10	15

To create a two-dimensional table, you will first have to create a *matrix* object and then convert it to a table object. A matrix is a type of object that holds a rectangular grid of data, where all the data is of the same type (either character strings or numbers). It is a bit like a vector, except with two dimensions. To create a matrix, use the `matrix` function:

```
> matrix1<-matrix(c(15, 10, 11, 15), nrow=2)
```

This creates the matrix shown here:

```
> matrix1
```

```
     [,1] [,2]
[1,]   15   11
[2,]   10   15
```

As you can see, R has arranged the list of values into a two-by-two matrix. R begins by filling the first column of the matrix from top to bottom, before moving onto the second column and so on. The `nrow` argument tells R how many rows the matrix should have, and the number of columns required is calculated automatically. Alternatively, you can use the `ncol` argument to specify the number of columns.

To give proper labels to the table dimensions and levels, use the `dimnames` argument:

```
> matrix1<- matrix(c(15, 9, 11, 17), nrow=2, dimnames=list(Group=c("Active", "Control"),
    Disease=c("Resolved", "Unresolved")))
```

Once you have created the matrix, use the `as.table` function to convert it to a table object:

```
> treattable<-as.table(matrix1)
```

Alternatively, you can create the table in one step by nesting the matrix function inside the `as.table` function:

```
> treattable<-as.table(matrix(c(15, 9, 11, 16), nrow=2, dimnames=list(Group=c("Active", "Control"),
    Disease=c("Resolved", "Unresolved"))))
```

Chi-Square Goodness-of-Fit Test

The *chi-square goodness-of-fit* test (also known as the Pearson's chi-squared test or χ^2 test) allows you to compare categorical data with a theoretical distribution. It has the null hypothesis that the data follows the specified distribution, and the alternative hypothesis that it does not. The test is only suitable if sufficient data is available, which is commonly defined as each category having an expected frequency (under the null hypothesis) of at least five.

The test should not be confused with the *chi-square test of association* (see the next section in this chapter), which helps to determine whether two or more categorical variables are associated. The chi-square goodness-of-fit test and the chi-square test of association are both forms of the Pearson chi-square test, but they answer different questions about the data. To illustrate the difference, if you had recorded the results of a series of six-sided die rolls and wanted to use this data to determine whether your die was fair, you would use the chi-square goodness-of-fit test. The null hypothesis for the test would be that each of the six sides is equally likely to be rolled, with probability 1/6. However, if you had recorded the name of the person rolling the die as well as the result of the die roll and you wanted to determine whether there was any relationship between the result of the roll and the person that rolled the die, you would use the chi-square test of association.

Note For more details about hypothesis testing, see Chapter 10.

You can perform a chi-square goodness-of-fit test with the `chisq.test` function. If you have a one-dimensional table object, you can test it against the uniform distribution (i.e., against the null hypothesis that all categories are equally likely to occur), as shown here:

```
> chisq.test(tableobject)
```

If you are using raw data, nest the `table` function inside the `chisq.test` function:

```
> chisq.test(table(dataset$factor1))
```

To test the data against a different theoretical distribution, use the `p` argument to give a list of expected relative frequencies under the null hypothesis. You must give the same number of relative frequencies as there are levels in your table, and they must sum to one.

For example, to test the hypothesis that 10 percent of the population belong to the first category, 40 percent to the second category, 40 percent to the third category, and 10 percent to the fourth category, use the command:

```
> chisq.test(tableobject, p=c(0.1, 0.4, 0.4, 0.1))
```

Example 6-1: Chi-Square Goodness-of-Fit Test Using the Apartments Data

Consider the `apartments` dataset (available with the downloads for this book and described in Appendix C), which give details of thirty-nine one-bedroom apartments advertised for rent in a particular area of the united Kingdom in October 2012. The `Price.Cat` variable gives the rental price category for the apartment.

To create a table showing the number of apartments in each price category, use the command:

```
> table(apartments$Price.Cat)
```

```
£500-550 £551-600 £601-650    £651+
       5       13        8        6
```

It is believed that 20 percent of the one-bedroom apartments in this area have a rental price less than £550, 30 percent have a price between £551 and £600, 30 percent have a price between £601 and £650, and 20 percent have a rental price greater than £650.

Suppose that you want to use a chi-square goodness-of-fit test to determine whether the data is consistent with the hypothesized price distribution. The test has the null hypothesis that the described price distribution is correct, and the alternative hypothesis that it is not. A significance level of 0.05 is used.

To perform the test, use the command:

```
> chisq.test(table(apartments$Price.Cat), p=c(0.2, 0.3, 0.3, 0.2))
```

This gives the output:

```
        Chi-squared test for given probabilities

data: table(apartments$Price.Cat)
X-squared = 1.8021, df = 3, p-value = 0.6145
```

The p-value of 0.6145 is not less than the significance level of 0.05, so we cannot reject the null hypothesis. This means that the data is consistent with the hypothesised price distribution.

Tests of Association Between Categorical Variables

In the same way that we can look at the association between continuous variables using statistics such as covariance and correlation, there are methods available to help you to determine whether there is an association between categorical variables. Two of these are the chi-square test of association and Fisher's exact test. These tests answer the same question but are suitable in different situations.

The chi-square test of association is used to test for association between two or more variables, and can be used regardless of how many levels there are in each variable. However, the test is only suitable when there is plenty of data available.

By contrast, Fisher's exact test can only be used to look for association between two categorical variables which each have two levels. However, it is suitable even when very little data is available.

The following sections explain how to perform these tests in R.

Chi-Square Test of Association

The *chi-square test of association* (sometimes called the chi-square test of independence) helps to determine whether two or more categorical variables are associated. The test has the null hypothesis that the variables are independent and the alternative hypothesis that they are not independent (i.e., at least two of the variables are associated) The test is only suitable if there is sufficient data, which is commonly defined as all table cells having expected counts (under the null hypothesis) of at least five.

The `summary` function performs a chi-square test of association when given a table object as input. If you have already created a table object, use the `summary` function directly:

```
> summary(tableobject)
```

If you have raw data, nest the `table` function inside the `summary` function:

```
> summary(table(dataset$var1, dataset$var2, dataset$var3))
```

Example 6-2: Chi-Square Test Of Association Using People2 Data

Suppose that you want to use a chi-square test of association to determine whether sex and eye colour are associated, using the `people2` dataset.

If you still have the `sexeyetable` object created earlier in this chapter in the "Frequency Tables" section, use the command:

```
> summary(sexeyetable)
```

Alternatively, you can perform the test using the raw data:

```
> summary(table(people2$Sex, people2$Eye.Colour))
```

```
Number of cases in table: 16
Number of factors: 2
Test for independence of all factors:
        Chisq = 2.2857, df = 2, p-value = 0.3189
        Chi-squared approximation may be incorrect
```

As the p-value of 0.3189 is not less than the significance level of 0.05, we cannot reject the null hypothesis that sex and eye color are independent. This means that there is no evidence of an association between the two.

The warning `Chi-squared approximation may be incorrect` tells us that as some cells have expected counts less than five. This means that the results may be unreliable and should be interpreted with caution. A discussion of chi-square tests with small expected counts can be found on page 39 of *The Analysis of Contingency Tables*, Second Edition, by B. S. Everitt (Chapman & Hall/CRC, 1992).

Example 6-3: Chi-Square Test of Association Using the Esoph Data

Consider the `esoph` dataset, which is included with R. The dataset gives the results of a case-control study of oesophageal cancer. You can view more details by entering `help(esoph)`. The `agegp`, `alcgp` and `tobgp` variables list categories for the subjects' age, alcohol consumption, and smoking habits. The variables `ncases` and `ncontrols` give the number of subjects with and without oesophageal cancer that fall into each of these categories.

Suppose that you want to use a chi-square test of association to determine where there is any association between smoking habits and oesophageal cancer. The test has the null hypothesis that smoking habits and oesophageal cancer are independent, and the alternative hypothesis that they are associated. A significance level of 0.05 will be used.

As the dataset contains data that is already expressed as counts, you can use the `xtabs` function to create a table giving the number of subjects with and without oesophageal cancer that fall into each category of smoking habits:

```
> tobacco<-xtabs(cbind(ncases, ncontrols)~tobgp, esoph)
> tobacco
```

```
tobgp       ncases ncontrols
  0-9g/day      78       525
  10-19         58       236
  20-29         33       132
  30+           31        82
```

perform the test, use the command:

```
> summary(tobacco)
```

```
Call: xtabs(formula = cbind(ncases, ncontrols) ~ tobgp, data = esoph)
Number of cases in table: 1175
Number of factors: 2
Test for independence of all factors:
        Chisq = 18.363, df = 3, p-value = 0.0003702
```

As the p-value of 0.0003702 is less than the significance level of 0.05, we can reject the null hypothesis that smoking category and oesophageal cancer are independent, in favour of the alternative hypothesis that the two are associated. This means that there is evidence of a relationship between smoking habits and oesophageal cancer.

Fisher's Exact Test

The Fisher's exact test is used to test for association between two categorical variables that each have two levels. Unlike the chi-square test of association, it can be used even when very little data is available. The test has the null hypothesis that the two variables are independent and the alternative hypothesis that they are not independent.

You can perform a Fisher's exact test with the `fisher.test` function. You can use the function with a two-by-two table object:

```
> fisher.test(tableobject)
```

You can also use raw data:

```
> fisher.test(dataset$var1, dataset$var2)
```

The test results are accompanied by a 95 percent confidence interval for the odds ratio. You can change the size of the interval with the `conf.level` argument:

```
> fisher.test(dataset$var1, dataset$var2, conf.level=0.99)
```

Example 6-4: Fisher's Exact Test Using People2 Data

Using the `table` function, we can see there are more left-handed women than left-handed men in the `people2` dataset:

```
> table(people2$Sex, people2$Handedness)
```

```
       Left Right
Male      0     9
Female    2     2
```

Suppose you want to perform a Fisher's exact test to determine whether there is any statistically significant relationship between sex and handedness. The test has the null hypothesis that sex and handedness are independent, and the alternative hypothesis that they are associated. A significance level of 0.05 is used.

To perform the test, use the command:

```
> fisher.test(people2$Sex, people2$Handedness)
```

This gives the output:

```
        Fisher's Exact Test for Count Data

data:  people2$Sex and people2$Handedness
p-value = 0.07692
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.000000 2.098087
sample estimates:
odds ratio
         0
```

Because the p-value of 0.07692 is not less than the significance level of 0.05, we cannot reject the null hypothesis that sex and handedness are independent. This means that there is no evidence of a relationship between sex and handedness.

Proportions Test

A test of proportions allows you to compare the proportion of observations with a given outcome or attribute (referred to as a *success*) across two or more groups of observations to determine whether they are significantly different. The null hypothesis for the test is that the probability of a success is the same for all of the groups, and the alternative hypothesis is that the probability of success is not the same for all of the groups.

The test can be applied to an n-by-two contingency table, where the two columns give the number of successes and failures, and the n rows give the number that fall into each group.

You can perform a test of proportions with the prop.test function. If your data is already saved in a table object, you can use the command:

```
> prop.test(tableobject)
```

For the command to work, your table must have exactly two columns, and the rows of the table should show the different groups that you want to compare. If your table is the wrong way around, so that the columns give groups and the rows give successes and failures, you can use the t function to transpose the table:

```
> prop.test(t(tableobject))
```

If you don't have a table object, you can perform the test using raw data by nesting the `table` function inside the `prop.test` function:

```
>prop.test(table(dataset$groups, dataset$outcome))
```

Alternatively, you can perform the test using raw count data:

```
> prop.test(c(success1, success2), c(n1, n2))
```

where `success1` and `success2` are the number of successes in Group 1 and Group 2, respectively, and `n1` and `n2` are the total number of observations in Group 1 and Group 2, respectively.

Example 6-5: Proportions Test

Consider the data shown in Table 6-1 (under "Creating Tables Directly"), which gives the number of patients with resolved and unresolved disease after treatment with either an active treatment or a control. If you still have the `treattable` table object, you can view the contents:

```
> treattable
```

```
         Disease
Group     Resolved  Unresolved
  Active        15          11
  Control        8          17
```

To see the proportion with resolved and unresolved disease for each treatment group, use the command:

```
> prop.table(treattable, margin=1)
```

```
        Disease
Group     Resolved  Unresolved
  Active  0.5769231  0.4230769
  Control 0.3200000  0.6800000
```

From the output, you can see that the proportion of subjects with resolved disease is 58 percent in the treatment group and 32 percent in the control group. Suppose that you want to use a test of proportions to help determine whether this difference is significant or whether it is just the result of random variation. A significance level of 0.05 will be used.

To perform the test, enter the command:

```
> prop.test(treattable)
```

Alternatively, you can perform the test using the raw data:

```
> prop.test(c(15,8), c(26,25))
```

This produces the following output:

```
        2-sample test for equality of proportions with continuity correction

data:  treattable
X-squared = 1.6153, df = 1, p-value = 0.2038
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.08963846 0.52348461
sample estimates:
   prop 1    prop 2
0.5769231 0.3600000
```

From the output, you can see that the p-value of 0.2038 is not less than the significance level of 0.05, so we cannot reject the null hypothesis that the proportion of patients with resolved disease is the same in both treatment groups. This means that there is no evidence of a difference in the probability of disease resolution for the active treatment and control.

Summary

You should be able to create frequency and contingency tables to summarize categorical data and be able to present them with marginal sums or as percentages if required. You should be able to compare your categorical data to a hypothesised distribution using the chi-square goodness-of-fit test. You should also be able to use the chi-square test of association or Fisher’s exact test to look for association between two categorical variables. Finally, you should be able to use a test to compare two or more proportions.

This table summarizes the most important commands covered.

Task	Command
Contingency table	`table`(*`dataset`*`$`*`factor1`*, *`dataset`*`$`*`factor2`*)
Compact table	`ftable`(*`tableobject`*)
Proportions table	`prop.table`(*`tableobject`*)
Table with margins	`addmargins`(*`tableobject`*)
Chi-square goodness-of-fit test	`chisq.test`(*`tableobject`*, `p=c(p1, p2, pN))`
	`chisq.test`(`table`(*`dataset`*`$`*`factor1`*), `p=c(p1, p2, pN))`
Chi-square test of association	`summary`(*`tableobject`*)
	`summary`(`table`(*`dataset`*`$`*`factor1`*, *`dataset`*`$`*`factor2`*))
Fisher's exact test	`fisher.test`(*`tableobject`*)
	`fisher.test`(*`dataset`*`$`*`factor1`*, *`dataset`*`$`*`factor2`*)
Proportions test	`prop.test`(*`tableobject`*)
	`prop.test`(`table`(*`dataset`*`$`*`groups`*,*`dataset`*`$`*`outcome`*))

Now that you have learned how to summarize continuous and categorical variables, we can move on to the next chapter, in which you will learn about probability distributions.