

Learn How to Create Cross-Tabulation Tables in R with the CrossTable() Function

Authored by
Mohammed looti

November 13, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Create Cross-Tabulation Tables in R with the CrossTable() Function*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=24049>

Introduction to Cross-Tabulation in R

Calculating a [cross-tabulation](#), often referred to as a [contingency table](#), is a core method in statistical analysis used to summarize the relationship between two or more [categorical variables](#). This powerful technique involves systematically grouping raw data based on defined categories and then tallying the frequency of observations for every possible combination of those categories.

While the base [R programming language](#) provides simple functions, such as `table()`, for generating raw frequency counts, analysts often require immediate access to associated statistical measures. These measures typically include various proportions (row, column, and total percentages) and metrics crucial for preliminary hypothesis testing and advanced [descriptive statistics](#).

The solution for generating rich, comprehensive cross-tabs in R is the specialized function **CrossTable()**. This function is conveniently housed within the robust [gmodels](#) package. **CrossTable()** is specifically engineered to produce highly readable output that includes not only the absolute counts but also the necessary proportional statistics needed for insightful data interpretation.

Understanding the CrossTable() Function and Syntax

The **CrossTable()** function is exceptionally versatile, offering a comprehensive summary that is ideal for both initial exploratory data analysis and formal [hypothesis testing](#). It serves as the preferred alternative to the standard `table()` function whenever a user needs more immediate statistical detail beyond simple counts.

The function employs a highly intuitive syntax, enabling users to rapidly define the two primary variables they intend to cross-reference. To effectively leverage its capabilities, a solid understanding of the core arguments is necessary for customizing the resulting table and ensuring the output meets specific analytical requirements.

The basic syntax structure required to execute **CrossTable()** is presented below:

CrossTable(x, y, digits=3, ...)

The primary arguments govern the structure and precision of the output:

x: Specifies the name of the first vector or variable. Conventionally, this variable dictates the categories that will form the **rows** of the resulting contingency table.

y: Specifies the name of the second vector or variable. This variable dictates the categories that will form the **columns** of the resulting contingency table.

digits: This is an optional argument controlling the precision of the numerical output. It sets the

maximum number of decimal places displayed for the calculated cell proportions and other statistical metrics. The default setting is typically 3.

...: This crucial placeholder allows for the inclusion of numerous additional statistical arguments. These can include options to suppress certain proportions, include [Chi-square statistics](#), calculate expected values, or apply Fisher's exact test, depending on the specific needs of the statistical test.

Before we proceed to the practical application, we must ensure our R environment is properly configured by installing and loading the required package.

Prerequisites: Installing and Loading the `gmodels` Package

To fully utilize the power and functionality of **`CrossTable()`**, the containing package, [gmodels](#), must be both installed on your system and loaded into your current R session. If this is your initial time accessing this package, the installation step is mandatory. However, once installed, you only need to execute the loading command in all subsequent R sessions.

To acquire the package and make it available locally, execute the following command directly within your R console or script editor:

```
install.packages('gmodels')
```

Following a successful installation, the package must be activated using the standard `library()` function. This essential step makes every function within the package, including **`CrossTable()`**, fully accessible and executable within your ongoing analysis environment.

This prerequisite loading step is crucial. Attempting to call **`CrossTable()`** without first loading the library will invariably result in an error message indicating that the function could not be found. Always confirm successful library loading before proceeding to the data structuring and analysis phase.

Practical Example: Structuring Data for Cross-Tabulation

To practically demonstrate the utility and robust output of the **`CrossTable()`** function, we will construct a common scenario encountered in social science research: investigating the potential relationship or association between two variables, specifically gender and political party preference. Analyzing this joint frequency is the archetypal use case for cross-tabulation.

In this constructed scenario, we hypothesize that political preference might exhibit a notable difference when stratified by gender. To explore this, we simulate the results of a small, simplified random sample survey, collecting data from 20 hypothetical registered voters on two key attributes:

their self-identified gender and their primary political affiliation.

We organize these synthetic survey responses into a standard R [data frame](#), which we name `df`. This structure includes two primary [categorical variables](#): `gen` (Gender, coded as M or F) and `pol` (Political Preference, coded as D for Democrat, I for Independent, or R for Republican).

#create data frame

```
df <- data.frame(gen=rep(c('M', 'F'), each=10),  
pol=c('D', 'D', 'D', 'D', 'R', 'R', 'I', 'I', 'I', 'D',  
'I', 'R', 'R', 'D', 'D', 'D', 'D', 'D', 'R', 'I'))
```

#view data frame

```
df
```

```
gen pol
```

```
1 M D
```

```
2 M D
```

```
3 M D
```

```
4 M D
```

```
5 M R
```

```
6 M R
```

```
7 M I
```

```
8 M I
```

```
9 M I
```

```
10 M D
```

```
11 F I
```

```
12 F R
```

```
13 F R
```

```
14 F D
```

```
15 F D
```

```
16 F D
```

```
17 F D
```

```
18 F D
```

```
19 F R
```

```
20 F I
```

The resulting [data frame](#) clearly defines the two variables of interest for our analysis. The `gen` column captures the respondent's gender (M=Male, F=Female), while the `pol` column records their political preference. The immediate objective is now to summarize the joint frequencies of these two attributes using the specialized function.

Executing the Cross-Tabulation and Reviewing Raw Output

With our structured [data frame](#) prepared and the necessary [gmodels](#) package loaded, the subsequent step is to execute the **CrossTable()** function call. We assign the gender variable (`df$gen`) as the row variable (`x`) and the political preference variable (`df$pol`) as the column variable (`y`). This arrangement ensures that the table is structured for clear interpretation of how preferences vary across gender categories.

Executing this function generates a highly detailed output table that includes the raw counts (N) and a comprehensive set of proportional measures. This structure provides a complete statistical portrait of the relationship between gender and political preference within our sampled population.

The code snippet below shows the mandatory library loading followed by the function execution:

```
library(gmodels)
```

```
#perform cross-tabulation of gender and political preference
```

```
CrossTable(x=df$gen, y=df$pol)
```

Cell Contents

```
|-----|
| N |
| Chi-square contribution |
| N / Row Total |
| N / Col Total |
| N / Table Total |
|-----|
```

Total Observations in Table: 20

```
| df$pol
df$gen | D | I | R | Row Total |
-----|-----|-----|-----|-----|
F | 5 | 2 | 3 | 10 |
| 0.000 | 0.100 | 0.100 | |
| 0.500 | 0.200 | 0.300 | 0.500 |
| 0.500 | 0.400 | 0.600 | |
| 0.250 | 0.100 | 0.150 | |
-----|-----|-----|-----|-----|
M | 5 | 3 | 2 | 10 |
| 0.000 | 0.100 | 0.100 | |
```

```
| 0.500 | 0.300 | 0.200 | 0.500 |
| 0.500 | 0.600 | 0.400 | |
| 0.250 | 0.150 | 0.100 | |
-----|-----|-----|-----|
Column Total | 10 | 5 | 5 | 20 |
| 0.500 | 0.250 | 0.250 | |
-----|-----|-----|-----|
```

The resulting output is a densely packed [contingency table](#). Before attempting to draw any statistical conclusions from the numbers, it is absolutely essential to thoroughly review and understand the "Cell Contents" key provided at the top, as this key defines precisely what each line of numerical data within every cell represents.

Detailed Interpretation of the `CrossTable` Output Metrics

The primary advantage of using `CrossTable()` lies in its ability to simultaneously present five distinct statistical metrics within every intersectional cell. Interpreting these five lines correctly is the most critical stage in transforming raw data frequencies into meaningful analytical insights regarding the joint distribution of the variables.

The five core components displayed in each cell are defined as follows:

N (Absolute Count): This represents the raw frequency, or the total number of observations, that fall into the specific combination of the row and column categories.

Chi-square contribution: This value quantifies the cell's individual contribution to the overall [Chi-square statistic](#). Crucially, higher values here suggest that the observed frequency in that specific cell deviates significantly from the expected frequency, assuming there is no association (independence) between the two variables.

N / Row Total (Row Proportion): This is the proportion of observations in that cell relative to the total number of observations in its corresponding row. This metric is invaluable for analyzing conditional probabilities based on the row variable (e.g., "What percentage of Females are Democrats?").

N / Col Total (Column Proportion): This is the proportion of observations in that cell relative to the total number of observations in its corresponding column. This metric helps analyze conditional probabilities based on the column variable (e.g., "What percentage of Democrats are Female?").

N / Table Total (Total Proportion): This is the proportion of observations in that cell relative to the grand total of all observations across the entire table. This provides the overall joint probability.

To solidify this understanding, let us analyze the first cell of the output, located at the intersection of Female (F) and Democrat (D):

The absolute count (N) is **5**. This means 5 individuals in the sample are Female Democrats.

The contribution to the **Chi-Square statistic** is **0.000**. This suggests that the observed count of 5 aligns perfectly with the expected count under the null hypothesis of independence.

The N / Row Total is **0.500** (or **50%**). This indicates that half of all sampled females prefer the Democratic party.

The N / Col Total is **0.500** (or **50%**). This indicates that among all sampled Democrats, exactly half are female.

The N / Table Total is **0.250** (or **25%**). This shows that Female Democrats constitute one-quarter of the entire group surveyed.

By systematically interpreting the marginal distributions (Row and Column Totals) alongside these five proportional values within the cells, analysts can rapidly identify patterns, test for significant differences, and quantify potential associations between the [categorical variables](#) under investigation. In our example, while overall counts are balanced, the proportional distribution across Independent and Republican preferences shows subtle differences that warrant further statistical exploration.

Conclusion and Next Steps in R Analysis

The **CrossTable()** function, provided by the essential [gmodels](#) package, offers an exceptionally detailed and efficient methodology for generating [cross-tabulation](#) tables within the R environment. It moves significantly beyond the limitations of simple frequency counts by instantly calculating critical statistical proportions and contributions required for association tests, establishing it as an indispensable asset for comprehensive [descriptive statistics](#).

True mastery of this function relies on accurately interpreting the five specific metrics presented within each cell. This skill allows researchers to precisely quantify observed relationships between variables, which then provides a solid foundation for subsequent, more complex inferential statistical procedures. Due to its robustness and detailed output, **CrossTable()** is highly recommended as a staple tool in any R user's statistical workflow.

To continue enhancing your proficiency in R for data management and advanced statistical analysis, explore related tutorials that cover other common data manipulation and inferential techniques.