Learning to Create Summary Tables in R with the psych Package

Authored by

Mohammed looti

November 4, 2025

Generating robust summary tables is an indispensable step in any rigorous R data analysis workflow. While native base R functions can provide basic statistics, the most efficient and comprehensive solution for obtaining detailed descriptive metrics is through the **psych library**. Specifically, the **describe()** and **describeBy()** functions offer a powerful, single-command method to generate a full suite of statistics, thereby streamlining the initial data exploration phase and enhancing the quality of your statistical summaries.

library(psych)

# Create a basic summary table for all variables in a data frame
describe(df)

# Create a summary table, grouping statistics by a specific factor variable
describeBy(df, group=df$var_name)

The following detailed examples illustrate how to implement these two cornerstone functions effectively. We will first demonstrate the use of **describe()** for a holistic data view, followed by **describeBy()** for crucial comparative analysis across different subsets of your data.

Example 1: Generating a Basic Summary Table with describe()

To showcase the utility of the **describe()** function, we will first define a sample data frame. This synthetic dataset represents hypothetical sports performance statistics, including measures like **points**, **rebounds**, and **steals**, recorded for players across three distinct teams (A, B, and C). This structure allows us to observe how the function handles both quantitative measures and qualitative, grouping data types simultaneously.

The creation and structure of our sample dataset in R is defined by the following code:

#create data frame
df <- data.frame(team=c('A', 'A', 'B', 'B', 'C', 'C', 'C'),
points=c(15, 22, 29, 41, 30, 11, 19),
rebounds=c(7, 8, 6, 6, 7, 9, 13),
steals=c(1, 1, 2, 3, 5, 7, 5))

#view data frame
df

team points rebounds steals
1 A 15 7 1
2 A 22 8 1

3 B 29 6 2
4 B 41 6 3
5 C 30 7 5
6 C 11 9 7
7 C 19 13 5

Upon applying the **describe()** function from the **psych library** to our newly created data frame, the function automatically processes every variable. It swiftly generates a highly detailed summary table, offering immediate insights into central tendency, dispersion, and distribution shape for each quantitative measure in the dataset, which is invaluable during the exploratory data analysis phase.

library(psych)

#create summary table
describe(df)

vars n mean sd median trimmed mad min max range skew kurtosis
team* 1 7 2.14 0.90 2 2.14 1.48 1 3 2 -0.22 -1.90
points 2 7 23.86 10.24 22 23.86 10.38 11 41 30 0.33 -1.41
rebounds 3 7 8.00 2.45 7 8.00 1.48 6 13 7 1.05 -0.38
steals 4 7 3.43 2.30 3 3.43 2.97 1 7 6 0.25 -1.73
se
team* 0.34
points 3.87
rebounds 0.93
steals 0.87

Interpreting the Detailed Statistical Output

The primary advantage of using **describe()** over simpler summary functions is the extensive list of statistical metrics it provides. For effective data analysis, it is essential to understand the meaning and implications of each column. This output moves beyond simple averages, giving analysts tools to assess variability, symmetry, and the presence of potential outliers.

The following definitions explain the comprehensive statistics generated by the function:

vars: The numerical index representing the column order of the variable in the input data frame.

n: The total count of valid (non-missing) observations included in the calculation for that specific variable.

mean: The arithmetic average, serving as the primary measure of central tendency.

median: The middle value when the data is sorted. This metric is preferred over the mean when the distribution is highly skewed or contains influential outliers.

trimmed: The trimmed mean, a robust measure calculated after a small percentage (typically 10%) of observations are removed from both tails of the distribution to mitigate the effect of extreme values.

mad: The **median absolute deviation**, which quantifies the variability of the data based on the median rather than the mean, making it highly robust to outliers.

min, **max**, **range**: These report the minimum and maximum observed values, respectively, with **range** being the difference between these two extremes (max - min).

skew: A measure of the asymmetry of the data's probability distribution. A positive skewness value indicates a tail extending further to the right (positive direction).

kurtosis: A measure of the 'tailedness' or peakedness of the distribution. High kurtosis often suggests the presence of significant outliers or unusually heavy tails.

se: The standard error of the mean, which estimates how much the sample mean is likely to vary from the true population mean.

It is critical to note how **describe()** handles non-numeric variables. Any variable output marked with an asterisk (*****), such as 'team*', indicates that the function has automatically coerced the **categorical variable** (factor or character type) into a numerical format based on its underlying factor level ordering. Therefore, the resulting summary statistics (mean, median, skewness, etc.) calculated for these specific converted variables should generally be disregarded for statistical interpretation.

Optimizing Output: Using Subset Selection and the fast=TRUE Argument

While the full output of **describe()** is comprehensive, analysts frequently require a more concise view, especially when dealing with production reports or extremely large datasets. The function provides built-in mechanisms to streamline the output, ensuring only the most pertinent statistics are calculated and displayed.

The simplest way to achieve a focused summary is by setting the **fast=TRUE** argument. When activated, the function restricts its calculations to the most essential summary statistics--specifically, *n*, *mean*, *sd*, *min*, *max*, *range*, and *se*--eliminating the calculation of more complex metrics like trimmed mean, skewness, and kurtosis. This results in a cleaner, faster, and more

easily digestible table, as demonstrated below:

#create smaller summary table
describe(df, fast=TRUE)

vars n mean sd min max range se
team 1 7 NaN NA Inf -Inf -Inf NA
points 2 7 23.86 10.24 11 41 30 3.87
rebounds 3 7 8.00 2.45 6 13 7 0.93
steals 4 7 3.43 2.30 1 7 6 0.87

Furthermore, precision is often needed in variable selection. Instead of summarizing the entire data frame, analysts can specify exactly which columns they wish to process. This is accomplished by using R's standard subsetting notation, passing a vector of desired column names (e.g., `df`) directly to the **describe()** function. This method ensures efficient processing and highly relevant reporting, avoiding unnecessary calculations on irrelevant variables.

#create summary table for just 'points' and 'rebounds' columns
describe(df, fast=TRUE)

vars n mean sd min max range se
points 1 7 23.86 10.24 11 41 30 3.87
rebounds 2 7 8.00 2.45 6 13 7 0.93

Example 2: Analyzing Grouped Data with describeBy()

For comparative statistical reporting--a requirement in nearly every analytical study--it is necessary to calculate descriptive statistics broken down by the levels of a specific grouping variable. The **describeBy()** function is specifically engineered to handle this scenario, executing the summary calculations for multiple subsets of the data simultaneously.

Using our sample sports dataset, we employ **describeBy()** to calculate performance statistics (for 'points', 'rebounds', and 'steals') grouped by the **df$team** variable. By retaining the **fast=TRUE** argument, we ensure the grouped output remains concise and focused, allowing for easy comparison between Team A, Team B, and Team C regarding their performance metrics.

#create summary table, grouped by 'team' variable
describeBy(df, group=df$team, fast=TRUE)

Descriptive statistics by group
group: A

vars n mean sd min max range se
team 1 2 NaN NA Inf -Inf -Inf NA
points 2 2 18.5 4.95 15 22 7 3.5
rebounds 3 2 7.5 0.71 7 8 1 0.5
steals 4 2 1.0 0.00 1 1 0 0.0
---
group: B
vars n mean sd min max range se
team 1 2 NaN NA Inf -Inf -Inf NA
points 2 2 35.0 8.49 29 41 12 6.0
rebounds 3 2 6.0 0.00 6 6 0 0.0
steals 4 2 2.5 0.71 2 3 1 0.5
---
group: C
vars n mean sd min max range se
team 1 3 20.00 9.54 11 30 19 5.51
rebounds 3 3 9.67 3.06 7 13 6 1.76
steals 4 3 5.67 1.15 5 7 2 0.67

The resulting output clearly shows the effectiveness of **describeBy()**. It systematically partitions the analysis, providing distinct summary statistics--including mean, standard error, and range--for each unique group level defined by the 'team' variable. This capability is essential for generating rapid, high-quality comparative statistical reports without requiring complex iterative code or manual subsetting.

Further Resources for Statistical Analysis in R

To deepen your understanding of statistical summaries and advanced data manipulation techniques in R, exploring additional resources is highly recommended. Mastering these foundations is key to efficient data science practice and producing reliable results.

How to Calculate Five Number Summary in R