Learning Descriptive Statistics with the `describe()` Function in R

Authored by

Mohammed looti

November 13, 2025

The Essential Role of Comprehensive Descriptive Statistics in R

In the early stages of any quantitative analysis project, the calculation of descriptive statistics is the indispensable foundation for understanding the characteristics, structure, and underlying distribution of a dataset. Data analysts routinely need to compute crucial metrics--such as the mean, median, range, and various measures of variability--for dozens or even hundreds of columns within an R data frame simultaneously. While base R provides individual functions like `mean()` or `sd()`, attempting to apply these operations efficiently across numerous variables often results in repetitive, cumbersome, and error-prone code.

The traditional approach to generating a comprehensive summary often requires writing complex functions, iterating through variables using loops, or chaining multiple commands together. This complexity detracts from the speed and fluidity required during the crucial exploratory data analysis (EDA) phase. The primary objective for efficient data exploration is to obtain a single, concise, and informative summary table that instantly illuminates the central tendency, dispersion, and shape of the data distribution for all relevant numeric variables.

This expert guide introduces the highly streamlined approach offered by the **describe()** function. This powerful tool is specifically engineered to bypass the necessity of writing extensive iterative code or complex base R command sequences, thereby significantly accelerating data exploration, quality assessment, and initial reporting processes for analysts working with the R environment.

Introducing the psych Package and the Power of describe()

The most effective and widely adopted methodology for generating a complete set of descriptive statistics in R relies on the **describe()** function, which is a core component of the highly respected psych package. Although the **psych** package was originally designed to support personality, psychometric, and psychological research, its robust statistical capabilities have established it as an essential utility for general-purpose data analysis across many disciplines.

In contrast to native R functions that demand iterative application to construct a full suite of metrics, **describe()** is expertly designed to return a vast array of statistics--encompassing measures of central tendency, dispersion, and distribution shape (including skewness and kurtosis)--for every column contained in the input object (whether it be a vector, matrix, or data frame) using only a single command. This centralized efficiency dramatically improves the speed and thoroughness of the initial data screening process.

The output produced by **describe()** is exceptionally comprehensive. It extends beyond the rudimentary mean and standard deviation, incorporating valuable yet often less common metrics such as the trimmed mean, the median absolute deviation (MAD), and the standard error (se). This high level of detail empowers analysts to quickly assess data quality, identify potential outliers, and

gain immediate insight into the symmetry and peakedness of the variable distributions without needing multiple supplementary calculations.

Syntax and Key Arguments of the describe() Function

To effectively leverage the comprehensive capabilities of this tool, it is paramount to understand its core syntax and the various optional arguments available for customizing the resulting statistical output. The fundamental structure of the **describe()** function is designed to be flexible, allowing it to easily adapt to different analytical requirements, particularly regarding the handling of missing data and the inclusion of specific distribution characteristics.

The general syntax for invoking the **describe()** function is structured as follows:

describe(x, na.rm=TRUE, interp=FALSE, skew = TRUE, ranges = TRUE, ...)

The function accepts several key arguments that meticulously control how the statistics are calculated and what specific information is incorporated into the final summary table. These parameters allow the user to precisely tailor the descriptive output to the exact requirements of their data analysis project:

x: This is the required primary argument, specifying the name of the vector, matrix, or data frame for which the descriptive statistics must be calculated.
na.rm: A logical argument, which defaults to **TRUE**. When set to **TRUE**, all Not Available (NA) values are systematically removed before the summary statistics are computed, ensuring that missing data does not distort the resulting calculations.
interp: A logical argument, which defaults to **FALSE**. This parameter dictates whether the median value should be calculated using the standard rank method or an interpolated method, an option sometimes preferred for ordinal data or specific statistical models.
skew: A logical argument, defaulting to **TRUE**. If active, the function will calculate and include measures of skewness and kurtosis, providing critical quantifiable information regarding the distribution's shape.
ranges: A logical argument, defaulting to **TRUE**. When set to **TRUE**, the function calculates and reports the total range of values for each variable (the difference between the maximum and minimum observations).

Setting Up the R Environment: Installation and Loading

Before the powerful **describe()** function can be executed, the necessary package must first be installed and then actively loaded into the current R session. Since **describe()** belongs to the psych package and is not included in base R, this preparation step is absolutely mandatory. Failure to successfully install and load the package will inevitably result in an error when the user attempts

to call the function.

To ensure a smooth analytical workflow, the initial prerequisite is the installation of the **psych** package, retrieved directly from the Comprehensive R Archive Network (CRAN). This installation is achieved using the standard command executed in the R console:

install.packages('psych')

Once the installation phase is complete, the package must be loaded into the memory of the active R session for its functions to become accessible for use. This loading step is crucial and must be repeated every time a new R session is initiated where functions from the package, such as **describe()**, are required. The `library()` function is responsible for handling this loading process. Following these two successful steps, the user is fully prepared to proceed with using **describe()** to analyze their complex datasets.

Practical Application: Generating Statistics for a Sample Data Frame

To demonstrate the remarkable power and simplicity inherent in the **describe()** function, we will walk through a concise practical example. We will use a simulated dataset representing performance metrics for a small group of hypothetical basketball players. This example will clearly showcase how quickly a complete statistical summary can be generated for multiple variables simultaneously.

We begin by constructing a sample data frame in R. This frame intentionally incorporates both categorical data (the `team` variable) and several quantitative variables (`points`, `assists`, and `rebounds`), simulating a typical mixed-variable dataset encountered in real-world analysis:

 # Create the sample data frame
df <- data.frame(team=c('A', 'A', 'A', 'A', 'B', 'B', 'B', 'B'),
points=c(99, 68, 86, 88, 95, 74, 78, 93),
assists=c(22, 28, 31, 35, 34, 45, 28, 31),
rebounds=c(30, 28, 24, 24, 30, 36, 30, 29))

View the data frame structure
df

team points assists rebounds
1 A 99 22 30
2 A 68 28 28
3 A 86 31 24
4 A 88 35 24

5 B 95 34 30
6 B 74 45 36
7 B 78 28 30
8 B 93 31 29

The created data frame, named `df`, provides information on eight individual players, including their team assignment and three core performance statistics: total points scored, total assists, and total rebounds accumulated. Our immediate goal is to compute a full suite of descriptive statistics--including mean, median, range, and measures of distribution shape--for these quantitative variables with maximum efficiency.

To achieve this, we first load the **psych** package using the `library()` command, followed by applying the **describe()** function directly to our data frame, `df`:

library(psych)

# Calculate comprehensive descriptive statistics for the data frame
describe(df)

vars n mean sd median trimmed mad min max range skew kurtosis
team* 1 8 1.50 0.53 1.5 1.50 0.74 1 2 1 0.00 -2.23
points 2 8 85.12 10.88 87.0 85.12 12.60 68 99 31 -0.25 -1.62
assists 3 8 31.75 6.71 31.0 31.75 4.45 22 45 23 0.55 -0.51
rebounds 4 8 28.88 3.83 29.5 28.88 1.48 24 36 12 0.30 -0.85
se
team* 0.19
points 3.85
assists 2.37
rebounds 1.36

Deconstructing the Comprehensive describe() Output

The summary table produced by the **describe()** function is highly dense and informative, presenting a dedicated row for each variable in the data frame and columns detailing a wide spectrum of statistical measures. The ability to correctly interpret this output is crucial for deriving actionable insights regarding the distribution and quality of the raw data.

It is essential to recognize that **describe()**, by default, attempts to calculate statistics for all variables, regardless of whether they are inherently numeric or categorical. In our working example, the `team` column is a character (non-numeric) variable. The function accommodates this

by internally coercing the character data into numeric factors, which generates the `team*` row in the output table. Because standard measures like the mean and standard deviation are statistically meaningless for nominal categorical data, the values reported in the `team*` row should generally be disregarded during the analytical interpretation phase.

For the truly quantitative variables (`points`, `assists`, `rebounds`), the output delivers a robust statistical profile. For instance, the `points` variable shows a mean of 85.12 and a standard deviation (sd) of 10.88, which quantifies the average score and the typical variation around that central point. The reported skewness value of -0.25 indicates a minor left (negative) skew, while the kurtosis of -1.62 suggests the distribution is somewhat platykurtic (flatter or less peaked than a standard normal distribution).

The following list provides a detailed explanation for each column returned by the function, serving as a critical dictionary for accurate result interpretation:

vars: The sequential index number of the variable within the input data frame.
n: The total count of valid observations (non-missing values) used for calculations for that specific variable.
mean: The calculated arithmetic average value of the variable.
sd: The standard deviation, the most common measure of the amount of variation or dispersion in the data set.
median: The middle value of the data when the observations are arranged in numerical order.
trimmed: The trimmed mean, calculated after removing a certain percentage (typically 10%) of the smallest and largest values, making it a measure of central tendency highly robust against outliers.
mad: The median absolute deviation, another highly robust measure of statistical dispersion, often preferred over the standard deviation when outliers are present.
min: The absolute minimum observed value for the variable.
max: The absolute maximum observed value for the variable.
range: The total spread of the data, calculated as the difference between the maximum and minimum values (max - min).
skew: The skewness value, which quantifies the degree and direction of asymmetry of the variable's probability distribution.
kurtosis: The kurtosis value, describing the "tailedness" and peakedness of the distribution in comparison to a normal distribution.
se: The standard error of the mean, an estimate of how closely the sample mean is likely to approximate the true population mean.

Conclusion and Next Steps in R Data Analysis

By systematically employing the **describe()** function provided by the psych package, data analysts

can significantly compress the time required to transition from raw data ingestion to generating a comprehensive statistical overview. This function stands as an indispensable tool for conducting efficient data auditing, rapid quality checking, and exploratory analysis, greatly accelerating the initial phases of any data project.

To further advance your proficiency in R, particularly in data manipulation and advanced summary generation, it is highly recommended to explore related functions concerning data aggregation, transformation, and statistical visualization. Mastering these powerful tools will unlock the capacity for generating deeper insights into complex data distributions and multivariate relationships.

The following resources provide further guidance on performing other common statistical and data management tasks efficiently in R:

<!--

Featured Posts

-->