

How to Create Regression Tables in R with the Stargazer Package

Authored by
stats writer

February 1, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Create Regression Tables in R with the Stargazer Package*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129063>

Using the Stargazer Package in R: A Comprehensive Guide to Professional Table Generation

The Stargazer package in R is an essential utility for statisticians and researchers who need to transform raw statistical output into publication-ready formats. This powerful tool excels at creating highly professional, well-formatted tables containing crucial regression results, summary statistics, and descriptive summaries. By automating the process of table generation, Stargazer ensures consistency and dramatically reduces the manual effort required to prepare results for academic papers, reports, or presentations.

One of the primary benefits of using Stargazer is its capacity to combine and present results from multiple statistical models side-by-side in a concise and organized manner. This capability is vital when comparing different model specifications or testing the robustness of findings across various datasets. To leverage this functionality, you must first ensure the package is loaded into your R session using the standard `library()` function. Following installation and loading, the core `stargazer()` function becomes the central command for defining your desired table output.

When utilizing `stargazer()`, you input your pre-calculated regression models (often generated using functions like `lm()` or `glm()`) and specify a range of formatting options tailored to your needs. These parameters allow granular control over the final presentation, including the table title, descriptive column labels, and the specific coefficients you wish to display. Below is a foundational example demonstrating how to simultaneously summarize two linear models, `lm1` and `lm2`, specifying the exact variables to be included in the output:

```
library(stargazer)
stargazer(lm1, lm2, title = "Regression Results", column.labels = c("Model 1", "Model 2"), coef = list("Intercept", "Variable 1", "Variable 2"))
```

Executing this command instantly generates a comprehensive table summarizing the estimated parameters of both statistical models. The output explicitly includes the intercept and the two specified variables, formatted clearly with an appropriate title and labeled columns, thereby simplifying complex model interpretation for the audience. The elegance and simplicity of the Stargazer package streamline the entire workflow for presenting rigorous regression analysis findings in R.

The Indispensable Role of Stargazer in Statistical Reporting

The move from raw statistical output--often dense and technical console printouts--to polished, publication-ready tables is a critical step in any data analysis pipeline. Historically, researchers

spent countless hours manually transcribing or reformatting output, a process that was not only tedious but also prone to transcription errors. The advent of specialized packages like Stargazer revolutionized this process, providing a robust, repeatable, and automated solution that adheres to the strict formatting standards required by academic journals and professional institutions.

Stargazer's utility extends far beyond mere aesthetic improvement; it enhances the clarity and accessibility of complex statistical findings. By standardizing the display of key information--such as standard errors, p-values, R-squared metrics, and observation counts--it allows readers to quickly grasp the essence of the analysis without getting lost in verbose descriptions. Furthermore, the package is designed to handle various types of model outputs, including linear, generalized linear, and robust regression models, offering a unified reporting framework across diverse statistical methodologies.

This package empowers users to generate high-quality tables directly from their R environment that are instantly usable in sophisticated document creation systems. While the examples here focus on text output, Stargazer is renowned for its seamless integration with LaTeX and HTML, making it an indispensable tool for researchers who rely on these standards for producing academic papers, dissertations, or web-based reports. The ease with which it handles formatting complexities--like aligning decimal points and applying significance stars--is a cornerstone of its popularity.

Getting Started: Installation and Setup of the Stargazer Package

Before any package can be utilized effectively within the R environment, it must be properly installed and loaded. Installation is a straightforward, one-time procedure typically handled via the `install.packages()` function, connecting directly to the Comprehensive R Archive Network (CRAN). Once the package resides locally on your system, it must be loaded into the current session using `library()` every time you start a new analysis session where you intend to use the package's functions. This ensures all necessary functions, including `stargazer()`, are accessible.

The following code snippet demonstrates the necessary commands to both install and then load the Stargazer package. It is crucial to run the installation step only once, but the loading step must be repeated for every new session. This process is fundamental to getting started with generating professional tables in R, setting the stage for subsequent data exploration and model summarization activities.

```
#install stargazer package  
install.packages('stargazer')
```

```
#load stargazer package  
library(stargazer)
```

Upon successful execution of these commands, the full functionality of the package becomes available. We can then proceed to utilize the core `stargazer()` function to produce the high-quality statistical summaries and tables required for our reporting needs. For the following examples, we will utilize the widely known built-in R dataset, `mtcars`, which provides data on 32 automobiles, offering a practical context for demonstrating the package's capabilities in generating both descriptive statistics and regression summaries.

Understanding the Core Syntax and Essential Arguments

The power of the `stargazer()` function lies in its simple yet flexible syntax, which allows users to direct the function to analyze a list of models or a data frame, specifying exactly how the output should be structured and exported. Mastering the basic arguments is key to unlocking the package's potential for generating tailored reports. The function utilizes a fundamental structure that includes defining the object to be summarized (be it a model list or a data frame), specifying the output format, assigning a title, and defining an export path.

The following schematic represents the core syntax of the `stargazer()` function, highlighting the necessary components for standard use:

```
stargazer(df, type='text', title='my_title', out='my_data.txt', ...)
```

The primary arguments within this function call determine the nature of the output generated:

df: This argument represents the name of the statistical object or data frame that the user intends to summarize. This could be a single data frame for descriptive statistics or a list of fitted regression models.

type: This is a crucial parameter dictating the format of the output. Common options include `'text'` (for console display or plain text files), `'html'` (for web embedding), and `'latex'` (for academic papers and documents).

title: A string used to provide a descriptive title that will appear at the top of the generated table, enhancing clarity and organization.

out: Specifies the file path and name to be used when exporting the resulting table. If this argument is omitted, the output is typically displayed directly in the R console or viewer, depending on the `type` specified.

It is important to note the versatility of the `out` argument concerning file formats. While we utilize `.txt` extensions in the examples provided here to export plain text files, users requiring integration into web documents should adjust this to `.html` for exporting HTML files. Similarly, users employing LaTeX for document typesetting would specify a `.tex` extension and set the `type` argument accordingly, ensuring seamless compatibility with high-fidelity document workflows.

Generating High-Quality Summary Statistics Tables

The `Stargazer` package is not solely focused on regression results; it is also highly effective at producing comprehensive tables of summary statistics. Descriptive statistics are often the first step in any robust analysis, providing foundational insights into the central tendency, dispersion, and distribution of variables within a data frame. Using `stargazer()` for this purpose allows analysts to quickly review critical metrics for quality control and initial exploration, presented in a clean, standardized format.

The function is designed to iterate through all variables in the supplied data frame, calculating and displaying a standardized set of metrics. These typically include the count of observations (N), the mean, standard deviation, minimum, maximum, and various percentiles (25th and 75th). This level of detail ensures that the reader gains a complete understanding of the data's characteristics before diving into complex modeling. We can demonstrate this capability using the built-in `mtcars` dataset, which is a classic example in R documentation.

The following code creates a descriptive table summarizing all numerical variables within the `mtcars` data frame. We specify `type='text'` to display the output neatly in the console and direct the output to an external file named `mtcars_data.txt` for easy archival and sharing:

```
#create table that provide summary statistics of each variable in dataset
stargazer(mtcars, type='text', title='Summary Statistics', out='mtcars_data.txt')
```

Summary Statistics

```
=====
Statistic N Mean St. Dev. Min Pctl(25) Pctl(75) Max
-----
mpg 32 20.091 6.027 10 15.4 22.8 34
cyl 32 6.188 1.786 4 4 8 8
disp 32 230.722 123.939 71 120.8 326 472
hp 32 146.688 68.563 52 96.5 180 335
drat 32 3.597 0.535 2.760 3.080 3.920 4.930
wt 32 3.217 0.978 1.513 2.581 3.610 5.424
qsec 32 17.849 1.787 14.500 16.892 18.900 22.900
vs 32 0.438 0.504 0 0 1 1
am 32 0.406 0.499 0 0 1 1
gear 32 3.688 0.738 3 3 4 5
carb 32 2.812 1.615 1 2 4 8
-----
```

Detailed Examination of the Summary Statistics Output

The resulting table demonstrates how effectively Stargazer condenses voluminous data information into an easily digestible structure. Crucially, a table of summary statistics has been accurately produced for every variable present in the mtcars data frame. This includes variables like 'mpg' (Miles per Gallon), 'hp' (Horsepower), and 'wt' (Weight), each analyzed across all 32 observations (N=32).

Reviewing this output provides immediate insights; for example, we see that the mean MPG is approximately 20.091, with a standard deviation of 6.027, indicating a moderate spread in fuel efficiency. The inclusion of percentiles (Pctl(25) and Pctl(75)) helps identify the interquartile range, providing a measure of variability less sensitive to outliers than the standard deviation. This detailed presentation ensures that the statistical assumptions and data properties underlying any subsequent regression modeling are fully transparent.

If the `out` argument was used, as in this example (`out='mtcars_data.txt'`), the user can subsequently navigate to the specified export location to view the plain text file containing these summary statistics. This exported file is a perfect, ready-to-use component for integration into formal reports or appendices, demonstrating the seamless transition from analysis in R to formal documentation.

Creating Professional Regression Model Summaries

The flagship capability of the Stargazer package is generating impeccably formatted summaries of statistical models. Once a model has been fitted--whether it's a simple linear regression (using `lm()`) or a more complex model--`stargazer()` takes the model object as input and structures the results into a standardized table format. This format is designed specifically for easy comparison and publication, focusing on the estimated coefficients, standard errors, and statistical significance levels.

To illustrate this, we will first fit a simple linear regression model predicting Miles per Gallon (mpg) based on displacement (disp) and horsepower (hp) using the mtcars data frame. The resulting model object, stored in the variable `fit`, is then passed to the `stargazer()` function. This process transforms the raw model output into a highly structured table titled 'Regression Summary' and exports it as a text file.

```
#fit regression model
```

```
fit <- lm(mpg ~ disp + hp, data=mtcars)
```

```
#create table that summarizes regression model
```

```
stargazer(fit, type='text', title='Regression Summary', out='mtcars_regression.txt')
```

Regression Summary

Dependent variable:

mpg

disp -0.030***

(0.007)

hp -0.025*

(0.013)

Constant 30.736***

(1.332)

Observations 32

R2 0.748

Adjusted R2 0.731

Residual Std. Error 3.127 (df = 29)

F Statistic 43.095*** (df = 2; 29)

Note: *p<0.1; **p<0.05; ***p<0.01

Interpreting the Regression Output and Key Metrics

The table generated by `stargazer()` is structured to prioritize readability. The first section details the estimated coefficients for each predictor variable, along with the intercept (Constant). Beneath each estimated coefficient, the standard error (in parentheses) is displayed, providing a measure of the precision of the estimate. Furthermore, the use of asterisks (***, **, *) immediately conveys the statistical significance of the findings, based on conventional p-value thresholds defined at the bottom of the table.

In this example, the Stargazer output shows that both displacement (`disp`) and horsepower (`hp`) have negative coefficients, indicating that as these variables increase, Miles per Gallon decreases. Notably, the coefficient for displacement is highly significant (`-0.030***`), while horsepower is significant at the 10% level (`-0.025*`).

Below the coefficient estimates, the table provides crucial model-level summary statistics. These metrics offer an overall assessment of the model's performance and fit. Key metrics include the

total number of observations (32), the R-squared value (0.748), which indicates that approximately 74.8% of the variance in MPG is explained by the model, and the Adjusted R-squared (0.731), which accounts for the number of predictors. The F Statistic and Residual Standard Error provide further quantitative measures of the model's overall statistical validity, completing the professional summary required for rigorous [regression analysis](#) reporting.

Customizing Output Formats: HTML, LaTeX, and Text

A major strength of the [Stargazer package](#) is its ability to output tables optimized for different document types. Researchers rarely publish results exclusively in plain text. Instead, they require seamless integration into complex document production systems like [LaTeX](#) for academic papers or HTML for online content and dynamic reports.

By simply altering the `type` argument in the `stargazer()` function, users can generate native code specific to these environments. Setting `type='latex'` produces high-quality [LaTeX](#) code, complete with necessary commands for table environments, precise column alignment, and formatting of statistical notation. This output can be directly embedded into a `.tex` file, saving significant formatting time and ensuring professional aesthetics.

Alternatively, using `type='html'` generates standard, clean HTML table markup. This format is ideal for integrating statistical results into web pages, dashboards, or reports generated via tools like R Markdown or Quarto. The package ensures that the HTML generated is valid and compatible with modern web standards, making the sharing of results via digital platforms efficient and visually appealing.

Conclusion: Streamlining Your R Workflow

The [Stargazer package](#) stands out as an indispensable tool for anyone conducting statistical analysis in [R](#). By automating the transition from computational output to presentation-quality tables, it drastically enhances productivity and ensures the consistency and professionalism of statistical reporting. Whether the task involves generating detailed [summary statistics](#) for exploratory data analysis or compiling complex side-by-side summaries of multiple [regression results](#), Stargazer provides the definitive solution.

By understanding the basic installation process, core syntax, and the flexibility offered by the `type` argument, analysts can efficiently tailor their output for any required medium--be it text, HTML, or [LaTeX](#). The package's commitment to providing clear, concise, and publication-ready results solidifies its reputation as a cornerstone utility in the modern statistical analyst's toolbox. We highly recommend consulting the official documentation for a comprehensive guide to all advanced customization options available.

Note: You can find the complete documentation for the **stargazer** package [here](#).

[How to Loop Through Column Names in R](#)

[How to Create an Empty Data Frame in R](#)

[How to Append Rows to a Data Frame in R](#)

ARABPSYCHOLOGY.COM