

Learn to Generate Publication-Ready Tables Using the Stargazer Package in R

Authored by
Mohammed loot

November 11, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learn to Generate Publication-Ready Tables Using the Stargazer Package in R*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16909>

As expert [R](#) users transition from routine data exploration to rigorous academic or professional reporting, the capability to generate high-quality, publication-ready tables becomes essential. The **stargazer** package in R is an indispensable utility for data scientists, econometricians, and researchers, specifically engineered to produce aesthetically refined and highly standardized statistical tables. These tables are perfectly suitable for immediate inclusion in journal articles, formal reports, theses, and other professional documentation.

The core strength of the **stargazer** package lies in its ability to streamline the often laborious process of formatting statistical output. It is expertly designed to manage two fundamental reporting requirements: generating comprehensive [summary statistics](#) for various data structures and elegantly presenting the results derived from complex [regression models](#). This detailed guide offers a practical, step-by-step walkthrough demonstrating how to harness the full potential of **stargazer**, utilizing the classic built-in R dataset, `mtcars`, as a practical and accessible foundation for all examples.

By automating the transformation of raw statistical results into polished, formatted tables, **stargazer** significantly enhances the reproducibility and consistency of research output. It eliminates the common pitfalls associated with manual data transfer and reformatting, ensuring that the presentation of quantitative results adheres to the stringent conventions expected in scholarly and industry publications.

The Crucial Role of the stargazer Package in Professional Reporting

In the vast ecosystem of statistical software, raw output--whether from model fitting functions or basic descriptive commands--frequently lacks the necessary polish and standardization required for professional communication. Analysts commonly face the challenge of manually exporting and meticulously reformatting this data into specific document preparation systems, such as word processors or advanced typesetting environments like [LaTeX](#). This manual intervention is not only excessively time-consuming but also introduces potential sources of error and inconsistency, particularly across large-scale projects or collaborative research efforts.

The primary, revolutionary advantage of **stargazer** is its capacity to generate output directly in several widely accepted formats, including plain text, [HTML](#), and LaTeX, often with just a single line of code. By providing this automated transformation capability, researchers can guarantee consistency and ensure the integrity of their statistical results across all published material. This automation drastically reduces the turnaround time between analysis completion and report generation.

Furthermore, **stargazer** is intelligently designed to automatically incorporate critical elements mandated by academic publishing standards. These elements include clearly formatted standard errors, meaningful statistical significance indicators (like asterisks), and essential model fit statistics

(such as R-squared and F-statistics). By adhering rigorously to established conventions, the package ensures that the produced tables are instantly recognizable and understandable to any research audience.

Installation and Preparing the R Environment

Before any of the powerful features of **stargazer** can be leveraged, the package must first be properly installed and then successfully loaded into the current R session. This procedure follows the standard protocol for integrating any third-party package into the R environment, ensuring that all necessary functions, dependencies, and underlying code are readily accessible for execution. As a fundamental practice in reproducible scripting, it is highly recommended to include both the installation command (executed once per environment setup) and the loading command (executed at the start of every session) at the beginning of any analytical script.

To begin our practical application, we execute the following standard R commands to prepare our working environment, confirming that the package is ready for use:

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

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

Once the package is successfully loaded using the `library()` function, we gain immediate access to the core `stargazer()` function. This function acts as the primary gateway, enabling us to produce all the high-quality statistical tables that are the focus of this guide. The next critical step involves gaining a thorough understanding of the fundamental syntax that governs this function's behavior and allows for deep customization of its output.

Deconstructing the Core Function Syntax and Arguments

The `stargazer()` function is characterized by its remarkable flexibility, as it is expertly designed to accept a diverse range of input objects. These inputs can include simple [data frames](#) (for descriptive analysis), single fitted regression models, or even extensive lists of multiple models intended for comparative presentation. While the complete spectrum of customization options available is vast, the basic syntax remains intuitive and relies upon a few crucial arguments to define the input source, the desired output format, and the final destination of the generated file.

The most fundamental structure governing the function call is presented below, illustrating the essential components required for effective use:

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

Each parameter within the function call plays a distinct and crucial role in directing the package's output generation:

df: This is the mandatory initial argument that specifies the primary object intended for summarization. Depending on the context, this input may be the name of a data frame (triggering the summary statistics mode) or a fitted model object (such as the result obtained from an [lm\(\)](#) or [glm\(\)](#) call), initiating the regression summary mode.

type: This argument is arguably the most important output control, as it explicitly defines the format in which the statistical table will be rendered. The most common and useful options include `'text'` (ideal for console display or generating plain text files), `'html'` (suitable for integration into web displays or simple insertion into word processors), and `'latex'` (the professional standard for generating typeset tables in academic publications).

title: This optional but highly recommended argument allows the user to define a clear, descriptive title. This title will be prominently displayed at the top of the generated table, significantly enhancing clarity and organizational structure within the final document.

out: When the user intends to save the table output directly to a file rather than merely displaying it in the R console, the `out` argument must be used. It specifies the complete desired file name and the corresponding file path.

It is best practice to ensure that the file extension specified within the `out` argument is consistent with the chosen `type` argument. For the purposes of demonstrating immediate console display, we utilize `.txt` in our examples. However, users focused on web integration should strictly use `.html`, and those targeting formal academic journals must use the `.tex` extension when `type='latex'` is specified.

Case Study 1: Generating Comprehensive Descriptive Summary Statistics

One of the most valuable and frequently utilized functions of **stargazer** is its ability to rapidly generate a comprehensive table of [summary statistics](#) encompassing all variables within a given dataset. This initial descriptive analysis is a crucial, foundational step for researchers, offering vital insights into the distribution, scale, and central tendency of the data before any inferential analysis is conducted. By employing the built-in `mtcars` [R dataset](#)--which contains key variables related to various vehicle performance metrics--we can quickly and efficiently produce a professional statistical overview.

To initiate the creation of this descriptive summary table, we simply pass the data frame object directly to the `stargazer()` function. We explicitly set `type='text'` to ensure the output is immediately visible within the R console and simultaneously exported to a plain text file, facilitating

both quick review and archival purposes:

```
#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
-----
```

The resulting output is a remarkably clean, professionally structured table that concisely provides essential metrics for every variable present within the input [data frame](#). For each variable (e.g., `mpg`, `hp`, `wt`), the table reports the sample size (N), the average value (Mean), the measure of dispersion (St. Dev.), the minimum and maximum observed values, and the critical 25th and 75th percentiles. This consolidated format is significantly superior to standard, fragmented R output, presenting all information in an organized, easy-to-read matrix. Furthermore, the ability to export this structured data to a file like `mtcars_data.txt` ensures that the results are immediately ready for secure sharing or rigorous archival purposes, maintaining complete integrity of the calculated statistics.

Case Study 2: Summarizing Statistical Regression and Model Results

The most widely celebrated and indispensable feature of **stargazer** is its unmatched capability to format and present the results derived from one or more statistical models, particularly those generated by common R functions such as [lm\(\)](#) for linear regression. This function is absolutely critical for researchers presenting empirical findings, as it efficiently bundles complex data points--including coefficient estimates, corresponding standard errors, indicators of significance levels, and overall model fit statistics--into a single, concise table.

We will fit a straightforward linear [regression model](#) aimed at predicting miles per gallon (`mpg`) based on vehicle displacement (`disp`) and horsepower (`hp`). Once the model is fitted using `lm()`, we pass the resulting model object (named `fit`) directly to the `stargazer()` function. This action automatically triggers the function's specialized model summary capabilities, differentiating its behavior from the descriptive data frame summary mode.

#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

The resulting table is meticulously and logically structured for maximum clarity. The top section explicitly identifies the dependent variable (`mpg`). Below this header, the table lists the independent variables (`disp` and `hp`) along with the intercept term (Constant). For every term in the model, the estimated coefficient is prominently displayed, followed immediately by the corresponding standard error enclosed in parentheses. Crucially, the coefficients are consistently annotated with asterisks

(*, **, ***) to visually denote their statistical significance based on calculated p-values. The bottom portion of the output provides essential overall model diagnostics, including the total number of observations, the R-squared value (which quantifies the proportion of variance explained), the Adjusted R-squared, the Residual Standard Error, and the overall F Statistic. This standardized presentation transforms complex statistical information into a format that is instantly understandable and professionally compliant for any research audience.

Advanced Output Formats: Seamless HTML and LaTeX Integration

While the `type='text'` output is highly valuable for rapid console viewing and simple text file exports, the true, advanced power of **stargazer** is fully realized when the package is utilized to generate outputs for high-fidelity, published documents. The package offers native support for two primary formats that are critical in modern publishing workflows: [HTML](#) and [LaTeX](#).

For users who need to integrate R output into dynamic web pages, various interactive reports, or documents within standard word processors like Microsoft Word, the `type='html'` setting is the ideal choice. This command generates a complete, self-contained HTML table structure, including all necessary formatting and styling tags, which can be effortlessly embedded or pasted into other documents. To export the previous regression summary into a functional HTML file, the command is modified simply by changing the `type` and the `out` arguments:

#Exporting to HTML format for web or Word

```
stargazer(fit, type='html', title='Regression Summary HTML', out='mtcars_regression.html')
```

Conversely, for the demanding world of academic publishing, which predominantly relies on the LaTeX typesetting system, the `type='latex'` option proves absolutely indispensable. The LaTeX output generated by **stargazer** is highly optimized, producing the precise tabular environment code that integrates perfectly into any LaTeX document. This results in perfectly typeset tables that conform automatically to stringent journal standards, a feature that alone saves researchers countless hours of manual, frustrating formatting. Moreover, **stargazer** offers an extensive suite of arguments for minute customization, enabling users to suppress specific statistics, apply variable renaming, insert custom footnote text, or even adjust the significance star thresholds, ensuring the final table meets every specific requirement of the intended publication.

Conclusion and Recommendation for Further Learning

The **stargazer** package distinguishes itself as an absolutely critical tool within the R ecosystem, successfully bridging the often-difficult gap between complex statistical computation and professional, clear communication. By greatly simplifying the process of creating publication-quality tables for both comprehensive [summary statistics](#) and intricate [regression model](#) results, the

package dramatically improves the reproducibility, efficiency, and professional polish of research workflows. Whether the requirement is exporting to plain text for quick internal review, HTML for dynamic report generation, or LaTeX for rigorous academic submission, **stargazer** ensures that statistical output is presented with unparalleled clarity, consistency, and strict adherence to established professional norms.

For researchers seeking to unlock the full potential of this package, including advanced control over features such as robust standard errors, detailed formatting styles, and customized presentation parameters, users are strongly encouraged to consult the official documentation. This resource provides a complete, exhaustive list of all available options, technical specifications, and detailed examples for every advanced usage scenario.

Note: You can find the complete documentation for the **stargazer** package [here](#), which is the authoritative source for all technical specifications.

Additional Resources for R Programming Mastery

To further enhance your expertise in data manipulation and statistical programming within the sophisticated R environment, we recommend exploring the following related tutorials, which address common technical challenges encountered when manipulating and analyzing complex datasets:

[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](#)