

# How to Create Publication-Ready Tables from R Regression Output with Stargazer

Authored by  
**stats writer**

January 17, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Create Publication-Ready Tables from R Regression Output with Stargazer*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126520>

## Introduction to the Stargazer Package

The stargazer package in the R programming language is an essential utility for researchers and analysts. Its primary function is to transform complex statistical output--specifically summary statistics and regression model results--into publication-ready tables. These tables are renowned for their clean aesthetics and adherence to academic formatting standards, making them ideal for integration into research papers, theses, or professional presentations. Utilizing **stargazer** significantly streamlines the process of communicating complex statistical findings effectively and clearly to a broad audience, replacing tedious manual formatting with automated, high-fidelity output generation.

To begin leveraging the power of this package, users must first ensure it is installed and loaded within their current R session. Once operational, the core `stargazer()` function acts as a versatile engine capable of handling various input types, whether you are summarizing variables within a large data frame or presenting the results of sophisticated econometric models. For instance, if a researcher is studying housing prices, **stargazer** can quickly generate a comprehensive table detailing descriptive statistics (like means, standard deviations, and quartiles) for prices across different geographic regions, ensuring the data is presented in a consistent and professional manner.

You can utilize the powerful stargazer package within R to produce exceptional quality tables suitable for immediate use in academic publications and reports. The following detailed example walks through the necessary steps, starting with installation and moving toward generating both descriptive and inferential statistical output using a standard, built-in R dataset.

## Prerequisites and Installation

Before utilizing any package in R, it must be installed on your system and subsequently loaded into the active environment. The stargazer package is available on CRAN (Comprehensive R Archive Network), making the installation process straightforward using the `install.packages()` command. Following installation, the `library()` function ensures that the package's functions are available for immediate use in your script or console session.

We begin our practical example by executing the required code block to set up the environment:

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

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

## Core Syntax of the stargazer Function

Once the package is successfully loaded, the primary function, `stargazer()`, is ready for use to produce high quality statistical tables. This function is highly customizable, allowing users to control the input data, the output format, the title of the table, and the file destination for exporting the results. Understanding the basic syntax and its key arguments is crucial for generating tailored output.

The generalized structure of the function call is as follows:

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

The key arguments govern how the function processes the data and formats the resulting table:

**df:** This argument specifies the name of the statistical object being summarized. This could be a data frame containing raw data for descriptive summary statistics, or a fitted statistical object, such as a linear model (`lm`) or generalized linear model (`glm`), for regression model summaries.

**type:** This parameter determines the output format. Common options include `'text'` (for console display or plain text files), `'html'` (for web display or easy integration into word processors), and `'latex'` (standard for academic journals).

**title:** A string defining the descriptive title that will appear at the top of the generated table.

**out:** Specifies the file path and name to which the generated table will be saved.

It is important to note the flexibility of the `out` parameter regarding file type. For the examples presented here, we will utilize the `.txt` extension to export simple text files suitable for console viewing. However, researchers frequently use the `.html` extension to export tables as high-fidelity HTML files, or the `.tex` extension for LaTeX integration, depending on the publication requirements.

## Two Primary Uses of stargazer

The `stargazer()` function is predominantly utilized to generate two distinct categories of statistical tables, each serving a critical purpose in data analysis and reporting:

Generation of comprehensive descriptive summary statistics for all relevant variables contained within a data frame.

Production of highly standardized and easily interpretable summary tables for the results of a fitted regression model, detailing coefficients, statistical significance, and overall model fit statistics.

## Generating Descriptive Summary Statistics

The first common application involves producing a table of descriptive statistics. We will

demonstrate this using the well-known **mtcars** dataset, which is built into the base R programming language environment. This dataset provides information on 32 automobiles regarding 11 variables, including fuel consumption (mpg), cylinder count (cyl), and horsepower (hp). By passing the `mtcars` object directly to `stargazer()`, we instruct the package to calculate and format key descriptive metrics for every column.

The following code snippet creates a descriptive table, outputs the result to the console (by setting `type='text'`), and simultaneously saves the output to a text file named `mtcars_data.txt`.

```
#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

```
-----
```

## Interpreting Descriptive Output

The resulting output is a meticulously structured table that provides essential summary statistics for every quantitative variable within the **mtcars** data frame. Each row corresponds to a variable (e.g., `mpg`, `hp`), and columns provide statistical measures such as the sample size (N), the Arithmetic Mean (Mean), the Standard Deviation (St. Dev.), the minimum and maximum values (Min and Max), and critical percentiles (Pctl(25) and Pctl(75)). This display is significantly more organized and professional than the default output provided by R's base functions.

For instance, examining the row for `mpg` (miles per gallon), we immediately see that the average MPG is 20.091 with a standard deviation of 6.027, ranging from a minimum of 10 to a maximum of

34. This succinct presentation allows readers to quickly grasp the distribution and central tendency of the variables being analyzed. Furthermore, because we used the `out` parameter, the complete summary statistics table has been exported to the specified text file (`mtcars_data.txt`) in the working directory, facilitating easy integration into external documents.

## Generating Regression Model Summaries

The most frequent and powerful application of the `stargazer` package is the summarization of fitted statistical objects. Unlike descriptive statistics, which summarize the raw data, regression summaries report the results of inference--that is, the relationship between dependent and independent variables. To demonstrate this, we will first fit a simple linear model (`lm`) predicting miles per gallon (`mpg`) based on displacement (`disp`) and horsepower (`hp`) using the `mtcars` data.

Once the model (`fit`) is created, we pass this fitted object directly to the `stargazer()` function. The package automatically recognizes that the input is a statistical model and structures the output accordingly, focusing on coefficients, statistical significance, and overall model diagnostics.

```
#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 Regression Output

The resulting regression model summary is highly informative. The upper section of the table displays the estimated coefficients for each predictor variable (`disp` and `hp`) and the intercept (Constant). Beneath each coefficient estimate, the standard error is provided in parentheses. The asterisks (e.g., \*\*\*, \*) indicate the level of statistical significance, based on the p-value thresholds defined at the bottom of the table.

The output displays the coefficients for each term in the regression model along with various summary statistics near the bottom of the table. In our example, the coefficient for `disp` (displacement) is -0.030 and is highly significant (\*\*\*,  $p < 0.01$ ). The lower section of the table provides crucial model diagnostics, including the R-squared (0.748), which indicates that approximately 74.8% of the variance in MPG is explained by displacement and horsepower.

## Advanced Output Formatting: HTML and LaTeX

While the examples above utilized the `type='text'` setting for simple console display, the true power of **stargazer** lies in its ability to generate output in formats optimized for publication quality. By changing the `type` argument to `'html'` or `'latex'`, the package generates code that can be directly embedded into web pages or academic manuscripts compiled using LaTeX, respectively.

For instance, setting `type='html'` and specifying an output file name ending in `.html` will produce a complete HTML table definition, complete with CSS classes and inline styling, ensuring perfect visual fidelity regardless of the destination platform. This capability is vital for researchers aiming for maximum efficiency and professionalism in their statistical reporting.

**Note:** Comprehensive documentation detailing all customization options, including how to select specific statistics, rename variables, or combine multiple models into a single comparative table, can be found on the official documentation page for the stargazer package.

## Further Resources for R Data Manipulation

To enhance your statistical workflow within R, consider exploring related topics on data structure and manipulation:

[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