

How to Perform Robust Regression in R: A Simple Step-by-Step Guide

Authored by
stats writer

December 6, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Perform Robust Regression in R: A Simple Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106114>

Robust regression is a fundamental statistical technique designed to mitigate the influence of extreme observations or outliers within a dataset. Unlike traditional methods like ordinary least squares regression (OLS), robust methods provide stable estimates of regression coefficients even when data points deviate significantly from the primary trend. This guide provides an expert, step-by-step walkthrough on how to perform powerful robust regression analysis using the statistical programming language R.

The process of implementing robust analysis in R typically relies on specialized packages such as `robustbase` or the highly utilized `MASS` package, which contains the key function for this procedure. Achieving robust results involves several crucial steps: initial package installation, model specification, careful parameter setting (such as tuning constants or iteration limits), and finally, the rigorous analysis and interpretation of the resultant coefficients and error metrics.

This tutorial will focus specifically on using the popular `r1m()` function from the **MASS** package, demonstrating how to effectively identify influential points and derive more reliable model parameters compared to standard OLS approaches, thereby ensuring the stability and trustworthiness of your statistical findings.

Understanding Robust Regression: Why We Need Alternatives

Traditional regression models, specifically ordinary least squares regression (OLS), operate under the critical assumption that the error distribution is normal and that there are no significantly influential observations. OLS minimizes the sum of squared residuals, meaning that data points far from the fitted line--the outliers--have a disproportionately large impact on the final coefficient estimates. A single severe outlier can dramatically shift the regression line, leading to biased predictions and misleading conclusions about the relationship between variables.

This is where robust regression techniques provide a necessary remedy. Robust methods, such as M-estimation (which the R function `r1m()` utilizes), seek to minimize a less rapidly increasing function of the residuals than squaring. Instead of minimizing the sum of squared errors, M-estimators minimize a defined function of the residuals, which often uses a tuning constant to determine the threshold beyond which residuals are downweighted. This mathematical approach effectively reduces the leverage of extreme values without requiring their complete removal from the dataset, preserving data integrity while ensuring model stability.

The primary benefit of employing robust techniques is the enhanced reliability of the parameter estimates. In real-world datasets--whether dealing with financial data, environmental measurements, or experimental psychology--the presence of measurement errors, data entry mistakes, or genuinely rare events is common. Relying solely on OLS in such contexts risks drawing flawed conclusions. Robust regression provides a methodology that is less sensitive to these minor violations of standard OLS assumptions, yielding regression coefficients that are more

reflective of the underlying central tendency of the data relationship.

The Limitations of Ordinary Least Squares (OLS)

To fully appreciate the need for robust methods, it is essential to understand the inherent weaknesses of OLS when faced with contaminated data. OLS is optimal under ideal conditions, but its high sensitivity to outliers stems from its reliance on minimizing the squared deviations. Because the influence of a residual is squared, doubling the distance of a point from the regression line quadruples its effect on the calculation of the slope and intercept. This multiplicative effect means that a few extreme points can completely skew the resulting model parameters.

A key diagnostic tool in regression analysis is the examination of standardized residuals. When plotting these residuals against predicted or observed values, OLS assumes that they should be randomly scattered around zero. Observations with standardized residuals exceeding an absolute value greater than 3 are generally flagged as potentially influential outliers. The presence of such points signals that the OLS fit is being stretched to accommodate data that may not belong to the primary data generating process, severely undermining the model's ability to generalize to new, uncontaminated data.

Furthermore, OLS lacks resistance. Resistance in statistics refers to how little an estimate changes when a small fraction of the data is altered or replaced by large values. Estimates derived from OLS have poor resistance, meaning they are easily pulled toward the outliers. Ordinary least squares regression effectively treats every data point as equally important in determining the line of best fit. Robust methods, conversely, apply differential weighting, giving less weight to points that exhibit large deviations, thereby providing a more resistant and trustworthy estimation procedure.

Prerequisites and Setup in R

To begin performing robust regression in R, the primary tool we will utilize is the `r1m()` function, which is contained within the powerful MASS package. The **MASS** package--standing for Modern Applied Statistics with S--is widely used for various statistical tasks, including generalized linear models, multivariate analysis, and robust estimation. If you do not already have this package installed, you must first install it from the Comprehensive R Archive Network (CRAN) and then load it into your current R session.

The standard installation command in the R console is `install.packages("MASS")`. Once installed, the library must be explicitly loaded using the `library(MASS)` command before the `r1m()` function can be called. The syntax for `r1m()` closely mimics that of the standard OLS function, `lm()`, making the transition seamless for analysts already familiar with R's modeling framework.

The general syntax for implementing the robust model is structured as follows:

```
robust_model <- rlm(formula, data=data_frame, method="M", psi=psi.huber)
```

While `rlm()` offers various methods, it defaults to M-estimation using the Huber weighting function (or Tukey's biweight, if specified). The Huber method is generally robust against outliers in the dependent variable (y-space). Understanding these underlying parameters is key to advanced robust modeling, although for this step-by-step tutorial, we will rely on the default settings which provide substantial improvement over OLS.

Step 4: Data Creation and Examination

Our first concrete step is the creation of a working dataset that intentionally includes points designed to act as influential outliers. By constructing a dataset where the standard OLS assumptions are clearly violated, we can effectively demonstrate the advantages of using robust regression. We will define two predictor variables, x_1 and x_2 , and a response variable, y . Note specifically the values in the y variable around the second and fourth observations (170 and 194), which are artificially inflated to skew the OLS fit significantly.

Below is the R code used to generate this simulated data frame and inspect its initial structure:

```
#create data  
df <- data.frame(x1=c(1, 3, 3, 4, 4, 6, 6, 8, 9, 3,  
11, 16, 16, 18, 19, 20, 23, 23, 24, 25),  
x2=c(7, 7, 4, 29, 13, 34, 17, 19, 20, 12,  
25, 26, 26, 26, 27, 29, 30, 31, 31, 32),  
y=c(17, 170, 19, 194, 24, 2, 25, 29, 30, 32,  
44, 60, 61, 63, 63, 64, 61, 67, 59, 70))  
  
#view first six rows of data  
head(df)  
  
x1 x2 y  
1 1 7 17  
2 3 7 170  
3 3 4 19  
4 4 29 194  
5 4 13 24  
6 6 34 2
```

The initial examination of the data confirms the presence of these extreme values. When the response variable y varies widely (from 2 up to 194), while most values cluster below 70, it strongly suggests that the dataset is contaminated. This preliminary inspection mandates a cautious approach to modeling, validating our decision to pursue a robust methodology over the standard ordinary least squares regression.

Step 5: Baseline Comparison using OLS Regression

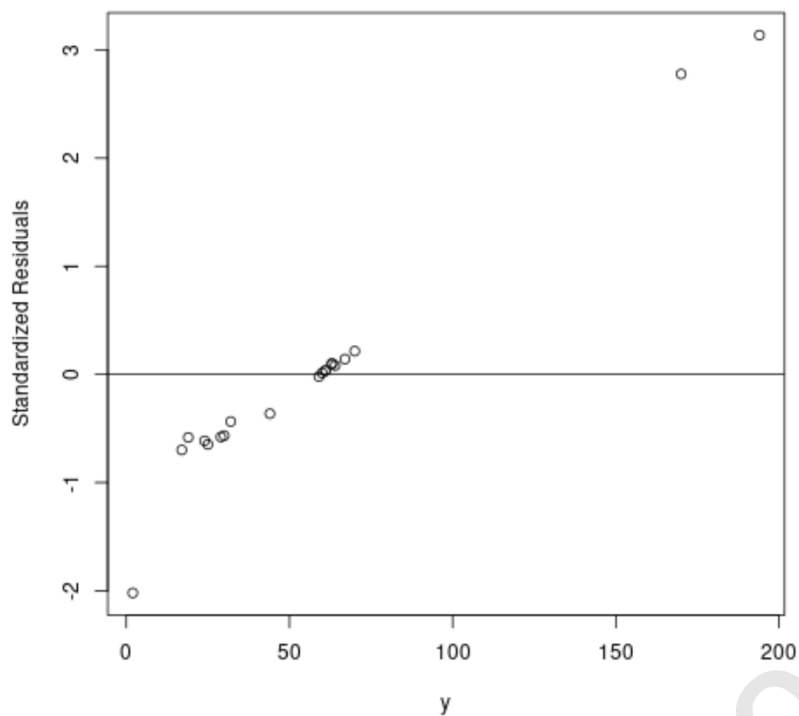
Before fitting the robust model, it is crucial to establish a baseline using OLS. This step serves two purposes: first, it confirms that the OLS model is indeed poorly fitted due to the outliers; and second, it provides a comparative metric (the standardized residuals) to visualize the impact of the influential points. We use the standard `lm()` function in R to fit the OLS model where y is predicted by x_1 and x_2 .

A fundamental diagnostic technique involves plotting the standardized residuals. As noted previously, observations having an absolute standardized residual greater than 3 are typically considered problematic outliers that warrant further investigation or specialized handling. If our OLS model is heavily influenced by these points, we expect to see them clearly separated from the bulk of the data on the residual plot.

The following R code executes the OLS fit and generates the diagnostic plot:

```
#fit ordinary least squares regression model  
ols <- lm(y~x1+x2, data=df)  
  
#create plot of y-values vs. standardized residuals  
plot(df$y, rstandard(ols), ylab='Standardized Residuals', xlab='y')  
abline(h=0)
```

Upon reviewing the resulting plot (shown below), the influence of the extreme observations becomes visually undeniable. We observe two specific data points whose standardized residuals approach or exceed the crucial threshold of 3. These points exert a substantial gravitational pull on the OLS regression line, suggesting that the model coefficients derived from this fit are likely unstable and unreliable for prediction or inference.



The graphical evidence strongly suggests that using OLS is inappropriate for this dataset. The presence of these influential points indicates that we will derive significant benefit from employing a methodology, like robust regression, that actively downweights the influence of these extreme residual values, leading us directly to the next step.

Step 6: Implementing Robust Regression using `r1m()`

Having confirmed the detrimental effect of outliers on the OLS model, we now proceed to fit the robust regression model using the `r1m()` function from the **MASS** package. The `r1m()` function performs M-estimation, which is designed to be highly resistant to data contamination. By invoking `r1m(y~x1+x2, data=df)`, we instruct R to iteratively reweight the observations based on their residuals, minimizing the influence of large deviations.

The iterative process of `r1m()` works by calculating an initial regression fit, determining the residuals, assigning a weight to each observation based on its residual size (smaller residuals get higher weights), and then refitting the model using these new weights. This process repeats until the coefficient estimates stabilize, effectively resulting in a regression line that is primarily driven by the majority of the data, rather than being distorted by the extreme points.

We execute the robust fit as follows:

```
library(MASS)
```

```
#fit robust regression model
robust <- rlm(y~x1+x2, data=df)
```

The resulting `robust` object now contains the resistant parameter estimates. While inspecting the raw coefficients is valuable, the most immediate and objective way to determine if this robust model provides a better fit than the OLS model is by quantitatively comparing their precision metrics, particularly the Residual Standard Error (RSE).

Step 7: Interpreting Results: Comparing Model Performance (RSE)

A critical metric for evaluating the overall fit and precision of a regression model is the Residual Standard Error (RSE), often denoted as `sigma` in R summaries. The RSE estimates the standard deviation of the error term, essentially quantifying the typical distance between the observed data points and the fitted regression line. In simple terms, a lower RSE indicates a more precise and tighter fit of the model to the data.

When comparing two models fitted to the same dataset, the model with a significantly lower RSE is generally considered superior, as it minimizes the unexplained variance. Given that robust regression is designed to ignore or downweight the largest residuals, we anticipate that the RSE of the robust model will be substantially smaller than that of the OLS model, which includes the large squared errors introduced by the outliers.

We use the `summary()$sigma` command to extract and compare the RSE for both the OLS and the robust regression models:

```
#find residual standard error of ols model
```

```
summary(ols)$sigma
```

```
49.41848
```

```
#find residual standard error of robust model
```

```
summary(robust)$sigma
```

```
9.369349
```

The results provide a striking quantitative confirmation of the effectiveness of robust methods. The OLS model yields an RSE of approximately 49.42, indicating a very large typical prediction error, primarily inflated by the two extreme outliers. In stark contrast, the robust regression model achieves an RSE of only 9.37. This five-fold reduction demonstrates conclusively that the robust model provides a far more stable and accurate representation of the underlying linear relationship between the predictors (`x1` and `x2`) and the response (`y`), based on the majority of the

observations.

In conclusion, when dealing with real-world data prone to contamination or standardized residuals exceeding typical thresholds, switching from standard OLS to robust regression techniques ensures that the resulting statistical inferences are not heavily influenced by noise or errors. The `rlm()` function in R offers a straightforward yet powerful solution for achieving stable and reliable coefficient estimates in the presence of challenging data structures.

ARABPSYCHOLOGY.COM