

How can I perform robust regression in R using a step-by-step approach?

Authored by
stats writer

April 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I perform robust regression in R using a step-by-step approach?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=140467>

Robust regression is a statistical method used to analyze data that contains outliers or influential observations. In R, this can be performed using a step-by-step approach to ensure accurate and reliable results. The first step is to load the necessary packages for robust regression, such as the "robustbase" package. Then, the data should be preprocessed by identifying and handling any outliers or influential observations. Next, the robust regression model can be built using the "lmrob" or "rlm" function, depending on the specific type of robust regression needed. This step involves specifying the dependent and independent variables, as well as any additional parameters. The model can then be evaluated by examining the robustness of the coefficients and residuals. Finally, any necessary adjustments or transformations can be made to improve the model's performance. By following this step-by-step approach, one can effectively perform robust regression in R and obtain reliable and accurate results for their data analysis.

Perform Robust Regression in R (Step-by-Step)

Robust regression is a method we can use as an alternative to ordinary least squares regression when there are outliers or in the dataset we're working with.

To perform robust regression in R, we can use the rlm() function from the MASS package, which uses the following syntax:

The following step-by-step example shows how to perform robust regression in R for a given dataset.

Step 1: Create the Data

First, let's create a fake dataset to work with:

```
#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
```

Step 2: Perform Ordinary Least Squares Regression

Next, let's fit an ordinary least squares regression model and create a plot of the .

In practice, we often consider any standardized residual with an absolute value greater than 3 to be an outlier.

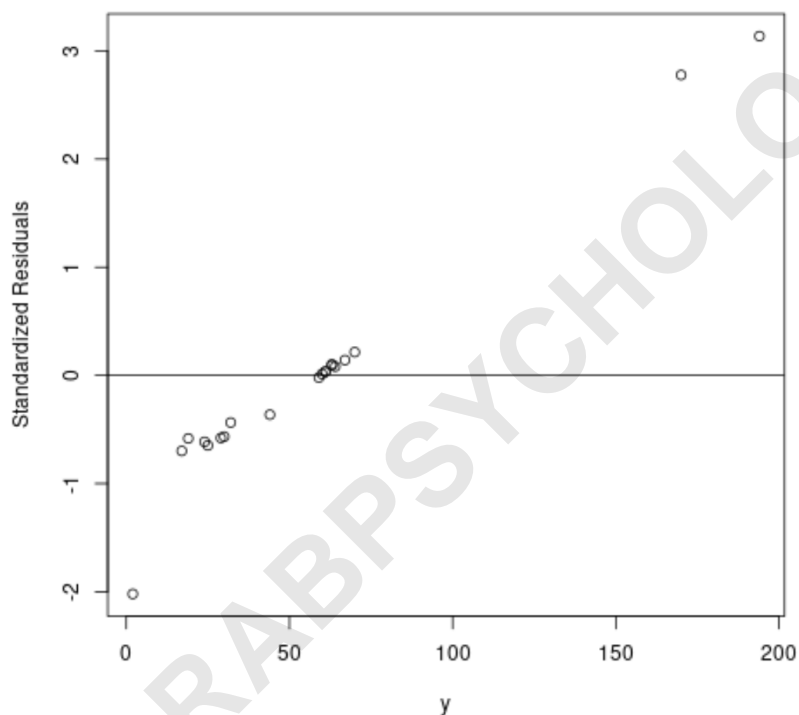
```
#fit ordinary least squares regression model
```

```
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)
```



From the plot we can see that there are two observations with standardized residuals around 3.

This is an indication that there are two potential outliers in the dataset and thus we may benefit from performing

robust regression instead.

Step 3: Perform Robust Regression

Next, let's use the `rlm()` function to fit a robust regression model:

```
library(MASS)
```

```
#fit robust regression model
```

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

To determine if this robust regression model offers a better fit to the data compared to the OLS model, we can calculate the residual standard error of each model.

The following code shows how to calculate the RSE for each model:

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

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

```
49.41848
```

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

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

9.369349

We can see that the RSE for the robust regression model is much lower than the ordinary least squares regression model, which tells us that the robust regression model offers a better fit to the data.

ARABPSYCHOLOGY.COM