

How can robust standard errors be calculated in R?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can robust standard errors be calculated in R?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153140>

Robust standard errors are a statistical method used to improve the accuracy of statistical analysis in the presence of outliers or heteroskedasticity (unequal variances). In R, robust standard errors can be calculated using the "lmRob" function from the "robust" package. This function utilizes robust regression methods such as Huber and biweight estimators to estimate the standard errors of the coefficients in a linear regression model. These robust standard errors provide more reliable and accurate results compared to traditional standard errors, making them a useful tool for researchers and analysts in various fields.

Calculate Robust Standard Errors in R

One of the is that the of the model are equally scattered at each level of the predictor variable.

When this assumption is violated, we say that is present in a regression model.

When this occurs, the standard errors for the regression coefficients in the model become untrustworthy.

To account for this, we can calculate robust standard errors, which are "robust" against heteroscedasticity and can give us a better idea of the true standard error values for the regression coefficients.

The following example shows how to calculate robust standard errors for a regression model in R.

Example: Calculating Robust Standard Errors in R

Suppose we have the following data frame in R that contains information on the hours studied and exam score received by 20 students in some class:

#create data frame

```
df <- data.frame(hours=c(1, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4,
4, 5, 5, 5, 6, 6, 7, 7, 8),
score=c(67, 68, 74, 70, 71, 75, 80, 70, 84, 72,
88, 75, 95, 75, 99, 78, 99, 65, 96, 70))
```

#view head of data frame

```
head(df)
```

```
hours score
```

```
1 1 67
```

```
2 1 68
```

```
3 1 74
```

```
4 1 70
```

```
5 2 71
```

```
6 2 75
```

We can use the function to fit a regression model in R that uses hours as the predictor variable and score as

the response variable:

#fit regression model

```
fit <- lm(score ~ hours, data=df)
```

#view summary of model

```
summary(fit)
```

Call:

```
lm(formula = score ~ hours, data = df)
```

Residuals:

```
Min 1Q Median 3Q Max
```

```
-19.775 -5.298 -3.521 7.520 18.116
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 71.158 4.708 15.11 1.14e-11 ***
```

```
hours 1.945 1.075 1.81 0.087 .
```

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 10.48 on 18 degrees of freedom

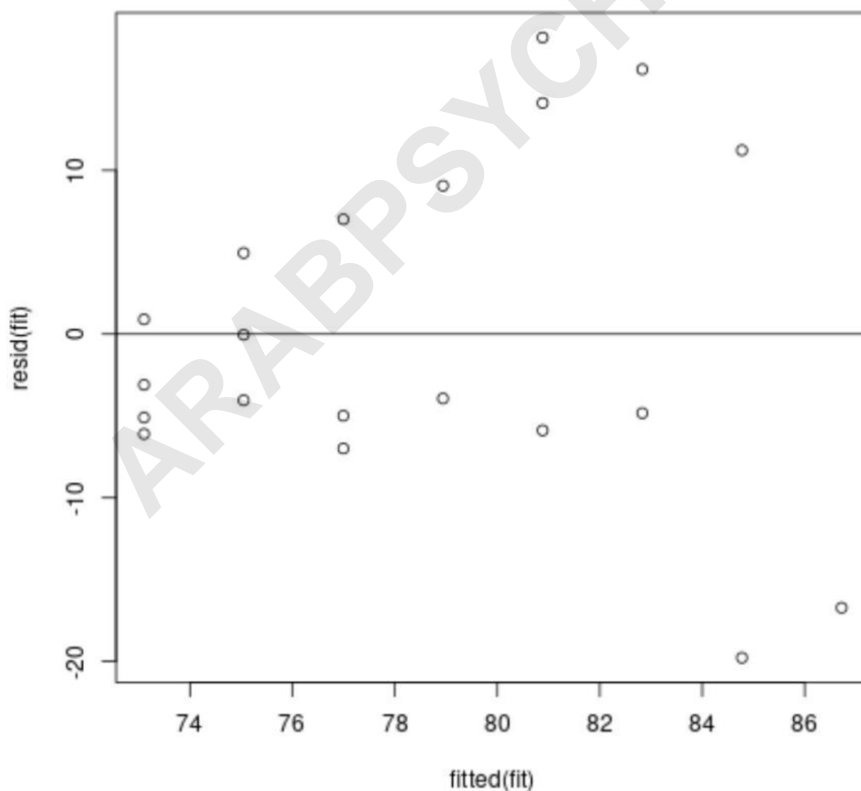
Multiple R-squared: 0.154, Adjusted R-squared: 0.107

F-statistic: 3.278 on 1 and 18 DF, p-value: 0.08696

The easiest way to visually check if heteroscedasticity is a problem in the regression model is to create a residual plot:

```
#create residual vs. fitted plot  
plot(fitted(fit), resid(fit))
```

```
#add a horizontal line at y=0  
abline(0,0)
```



The x-axis shows the fitted values of the response variable and the y-axis shows the corresponding residuals.

From the plot we can see that the variance in the residuals increases as the fitted values increase.

This is an indication that heteroscedasticity is likely a problem in the regression model and the standard errors from the model summary are untrustworthy.

To calculate robust standard errors, we can use the `coeftest()` function from the `lmtest` package and the `vcovHC()` function from the `sandwich` package as follows:

```
library(lmtest)
library(sandwich)
```

```
#calculate robust standard errors for model coefficients
coeftest(fit, vcov = vcovHC(fit, type = 'HC0'))
```

t test of coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.1576 3.3072 21.5160 2.719e-14 ***
```

hours 1.9454 1.2072 1.6115 0.1245

Signif. codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

Notice that the standard error for the hours predictor variable increased from 1.075 in the previous model summary to 1.2072 in this model summary.

Since heteroscedasticity is present in the original regression model, this estimate for the standard error is more trustworthy and should be used when calculating a confidence interval for the hours predictor variable.

Note: The most common type of estimate to calculate in the `vcovHC()` function is 'HC0', but you can refer to the `?vcovHC` help page to find other estimate types.