How can the confint() function be used in R?

Authored by

stats writer

June 26, 2024

The confint() function in R is a useful tool for obtaining confidence intervals for model parameters in statistical analysis. It can be used to calculate the upper and lower bounds of a confidence interval for a given set of data, with a specified level of confidence. This function is particularly useful for evaluating the precision and accuracy of model estimates and determining the range in which the true value of a parameter is likely to fall. By providing a measure of uncertainty, the confint() function allows researchers to make more informed decisions based on their data and can help improve the reliability of statistical analyses. It is a versatile function that can be applied to a variety of models and data types, making it an essential tool for data analysis in R.

Use the confint() Function in R

You can use the confint() function in R to calculate a confidence interval for one or more parameters in a fitted regression model.

This function uses the following basic syntax:

confint(object, parm, level=0.95)

where:

object: Name of the fitted regression modelparm: Parameters to calculate confidence interval for (default is all)level: Confidence level to use (default is 0.95)

The following example shows how to use this function in practice.

Example: How to Use confint() Function in R

Suppose we have the following data frame in R that shows the number of hours spent studying, number of practice exams taken, and final exam score for 10 students in some class:

```
#create data frame
df <- data.frame(score=c(77, 79, 84, 85, 88, 99, 95, 90, 92, 94),
hours=c(1, 1, 2, 3, 2, 4, 4, 2, 3, 3),
prac_exams=c(2, 3, 3, 2, 4, 5, 4, 3, 5, 4))

#view data frame
df

score hours prac_exams
1 77 1 2
2 79 1 3
3 84 2 3
4 85 3 2
5 88 2 4
6 99 4 5
7 95 4 4
8 90 2 3
```

9 92 3 5

10 94 3 4

Now suppose we would like to fit the following multiple linear regression model in R:

Exam score = β0 + β1(hours) + β2(practice exams)

We can use the function to fit this model:

```
#fit multiple linear regression model
fit <- lm(score ~ hours + prac_exams, data=df)

#view summary of model
summary(fit)

Call:
lm(formula = score ~ hours + prac_exams, data = df)

Residuals:
Min 1Q Median 3Q Max
-2.4324 -1.2632 -0.8956 0.4316 5.1412

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 68.4029 2.8723 23.815 5.85e-08 ***
```

hours 4.1912 0.9961 4.207 0.0040 **

prac_exams 2.6912 0.9961 2.702 0.0306 *

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

Residual standard error: 2.535 on 7 degrees of freedom

Multiple R-squared: 0.9005, Adjusted R-squared: 0.8721

F-statistic: 31.68 on 2 and 7 DF, p-value: 0.0003107

Notice that the model summary displays the fitted regression coefficients:

Intercept = 68.4029hours = 4.1912prac_exams = 2.6912

To obtain a 95% confidence interval for each of these coefficients, we can use the confint() function:

#calculate 95% confidence interval for each coefficient in model

confint(fit)

2.5 % 97.5 %

(Intercept) 61.6111102 75.194772

hours 1.8357237 6.546629

prac_exams 0.3357237 5.046629

95% C.I. for Intercept = 95% C.I. for hours = 95% C.I. for prac_exams =

To instead calculate a 99% confidence interval, simply change the value for the level argument:

```
#calculate 99% confidence interval for each coefficient in model
confint(fit, level=0.99)
```

```
0.5 % 99.5 %
(Intercept) 58.3514926 78.454390
hours 0.7052664 7.677087
prac_exams -0.7947336 6.177087
```

And to only calculate a confidence interval for a specific parameter, simply specify the coefficient using the parm argument:

```
#calculate 99% confidence interval for hours
confint(fit, parm='hours', level=0.99)
```

```
0.5 % 99.5 %
hours 0.7052664 7.677087
```

Notice that the 99% confidence interval is shown for the hours variable only.

The following tutorials provide additional information about linear regression in R: