How do you perform a Breusch-Godfrey test in R?

Authored by

stats writer

April 28, 2024

The Breusch-Godfrey test is a statistical test used to check for the presence of autocorrelation in a regression model. In order to perform this test in R, you can follow these steps:

1. Load the "lmtest" package in R by using the command: library(lmtest)

2. Use the "lm" function to fit your regression model and save it in an object (let's say "model1").

3. Use the "bgtest" function from the "lmtest" package to perform the Breusch-Godfrey test on the residuals of your model. This can be done by using the command: bgtest(model1)

4. The output of this command will include the test statistic, degrees of freedom, and the p-value. The null hypothesis of this test is that there is no autocorrelation present in the residuals of your model. If the p-value is less than the chosen significance level (usually 0.05), then you can reject the null hypothesis and conclude that there is significant autocorrelation present in your model.

5. You can also visualize the results of the Breusch-Godfrey test by plotting the residuals of your model against the lagged residuals using the "plot" function. This can help you identify any patterns or trends in the residuals that indicate the presence of autocorrelation.

In summary, the Breusch-Godfrey test in R can be performed by loading the "lmtest" package, fitting a regression model, and using the "bgtest" function to test for autocorrelation in the residuals.

Perform a Breusch-Godfrey Test in R

One of the is that there is no correlation between the residuals, e.g. the residuals are independent.

To test for first-order autocorrelation, we can perform a . However, if we'd like to test for autocorrelation at higher orders then we need to perform a Breusch-Godfrey test.

This test uses the following :

H0 (null hypothesis): There is no autocorrelation at any order less than or equal to *p*.

HA (alternative hypothesis): There exists autocorrelation at some order less than or equal to *p*.

The test statistic follows a Chi-Square distribution with *p* degrees of freedom.

If the that corresponds to this test statistic is less than a certain significance level (e.g. 0.05) then we can reject the null hypothesis and conclude that autocorrelation exists among the residuals at some order less than or equal to *p*.

To perform a Breusch-Godfrey test in R, we can use the bgtest(y ~ x, order = p) function from the lmtest library.

This tutorial provides an example of how to use this syntax in R.

Example: Breusch-Godfrey Test in R

First, let's create a fake dataset that contains two predictor variables (x1 and x2) and one response variable (y).

```
#create dataset
df <- data.frame(x1=c(3, 4, 4, 5, 8, 9, 11, 13, 14, 16, 17, 20),
x2=c(7, 7, 8, 8, 12, 4, 5, 15, 9, 17, 19, 19),
y=c(24, 25, 25, 27, 29, 31, 34, 34, 39, 30, 40, 49))

#view first six rows of dataset
head(df)

x1 x2 y
1 3 7 24
2 4 7 25
3 4 8 25
4 5 8 27
5 8 12 29
6 9 4 31
```


Next, we can perform a Breusch-Godfrey test using the bgtest() function from the lmtest package.

For this example, we'll test for autocorrelation among the residuals at order p =3:

```
#load lmtest package
library(lmtest)
```

```
#perform Breusch-Godfrey test
bgtest(y ~ x1 + x2, order=3, data=df)

Breusch-Godfrey test for serial correlation of order up to 3

data: y ~ x1 + x2
LM test = 8.7031, df = 3, p-value = 0.03351
```

From the output we can see that the test statistic is X2 = 8.7031 with 3 degrees of freedom. The corresponding p-value is 0.03351.

Since this p-value is less than 0.05, we can reject the null hypothesis and conclude that autocorrelation exists among the residuals at some order less than or equal to 3.

How to Handle Autocorrelation

If you reject the null hypothesis and conclude that autocorrelation is present in the residuals, then you have a few different options to correct this problem if you deem it to be serious enough:

For positive serial correlation, consider adding lags of

the dependent and/or independent variable to the model.For negative serial correlation, check to make sure that none of your variables are *overdifferenced*.For seasonal correlation, consider adding seasonal dummy variables to the model.