

How to Easily Perform a Breusch-Godfrey Test in Python

Authored by
stats writer

December 5, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Perform a Breusch-Godfrey Test in Python*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106037>

A fundamental requirement for establishing the validity of a standard Regression Model, particularly when employing Ordinary Least Squares (OLS) estimation, is that the residuals (the errors) must be independent. This means there should be no systematic relationship or correlation between the error terms across different time periods or observations. When this assumption is violated, our model's coefficients may be estimated correctly, but the standard errors--and thus, our derived confidence intervals and hypothesis tests--become unreliable and biased, often leading to misleading conclusions about the significance of predictor variables.

The presence of serial dependence in the error terms is known as Autocorrelation. To detect if this critical assumption has been violated, statistical tests are necessary. While the Durbin-Watson test is commonly used to check for first-order Autocorrelation, it is limited in its scope. When analysts need to test for serial correlation at higher orders ($p > 1$), or when lagged dependent variables are included in the model, the powerful and versatile Breusch-Godfrey Test becomes the preferred tool.

Understanding the Assumption of Non-Autocorrelation in Regression

In the realm of time series analysis and general linear modeling, the assumption that the error terms are independently and identically distributed (i.i.d.) is paramount. If a model exhibits significant Autocorrelation--meaning the residual error from one period is correlated with the error from a previous period--it suggests that relevant information has been omitted from the model specification, or that the dynamics of the time series are not being captured correctly.

Ignoring the presence of serial correlation can have severe statistical consequences. Although OLS estimators remain unbiased and consistent even with autocorrelation, they lose their efficiency. More critically, the variance of the error term becomes underestimated, resulting in standard errors that are too small. This inflation of precision leads to inflated t-statistics, increasing the likelihood of incorrectly rejecting the null hypothesis--a Type I error. Therefore, diagnosing and addressing autocorrelation is an essential step in robust model building.

The Breusch-Godfrey Test: Theory and Hypotheses

The Breusch-Godfrey Test, also known as the LM (Lagrange Multiplier) test for serial correlation, is designed to identify autocorrelation of any order up to a specified lag (p). Unlike the Durbin-Watson test, the Breusch-Godfrey test is robust even in models that include lagged values of the dependent variable as predictors, making it highly flexible for dynamic models.

The test is based on an auxiliary regression, where the residuals of the original model are regressed on the original predictor variables and the lagged values of the residuals themselves (up to lag p). The resulting R-squared value from this auxiliary regression is used to construct the test statistic. This test uses the following formal hypotheses:

H0 (Null Hypothesis): There is no Autocorrelation in the residuals up to order p . Statistically, this implies that the coefficients for all lagged residuals in the auxiliary regression are simultaneously zero.

HA (Alternative Hypothesis): There exists Autocorrelation at some order less than or equal to p . This suggests at least one of the lagged residual coefficients is non-zero.

The test statistic follows a Chi-Square distribution with p degrees of freedom, where p is the maximum lag order being tested. If the P-value corresponding to this test statistic is less than the predetermined significance level (commonly $\alpha = 0.05$), we reject the null hypothesis, concluding that significant serial correlation exists.

We can efficiently perform the Breusch-Godfrey Test in Python by utilizing the robust functionality provided by the statsmodels library, specifically through the `acorr_breusch_godfrey()` function. The following example details the implementation process.

Step 1: Creating the Sample Dataset

Before fitting any model or performing diagnostic tests, we must prepare our data. For this demonstration, we will create a simple dataset suitable for a Linear Regression Model. This dataset will include two predictor variables, x_1 and x_2 , and a single response variable, y . We will use the **pandas** library in Python to structure this data efficiently as a DataFrame.

The data points are intentionally sequenced to mimic a time series structure, although the test is applicable to any ordered data where the sequence matters (e.g., cross-sectional data ordered by a specific criterion). This step ensures we have a clear, manageable dataset upon which to build our statistical model and subsequently test its underlying assumptions.

```
import pandas as pd
```

```
#create dataset
df = pd.DataFrame({'x1': ,
                  'x2': ,
                  'y': })
```

```
#view first five rows of dataset
df.head()
```

```
x1 x2 y
0 3 7 24
1 4 7 25
2 4 8 25
```

3 5 8 27

4 8 12 29

Step 2: Fitting the Linear Regression Model

The Breusch-Godfrey Test is applied to the residuals derived from an existing statistical model. Therefore, the second crucial step involves fitting a standard Regression Model using our defined predictor variables (x_1 and x_2) and the response variable (y). We will leverage the Ordinary Least Squares (OLS) function provided by the statsmodels library, which is a powerful tool for statistical modeling in Python.

It is essential to correctly define the dependent and independent variables and include an intercept term (constant) in the model, which is a standard procedure for linear regression. The `sm.add_constant(x)` function handles the addition of this intercept, ensuring the model is correctly specified. Once the model is fitted using `sm.OLS(y, x).fit()`, the resulting model object contains all necessary statistics, including the residuals required for the subsequent diagnostic test.

```
import statsmodels.api as sm
```

```
#define response variable
```

```
y = df
```

```
#define predictor variables
```

```
x = df]
```

```
#add constant to predictor variables
```

```
x = sm.add_constant(x)
```

```
#fit linear regression model
```

```
model = sm.OLS(y, x).fit()
```

Step 3: Executing the Breusch-Godfrey Test

With the OLS model fitted, we can now proceed to perform the Breusch-Godfrey Test. We must specify the maximum order of autocorrelation we wish to test for, denoted by p or `nlags`. Choosing the appropriate lag order is a critical decision; generally, it should be informed by the context of the data (e.g., seasonality or known lag effects). For this pedagogical example, we will select $p = 3$, meaning we are testing for autocorrelation up to the third lag.

We access the testing function, `acorr_breusch_godfrey()`, through the `diagnostics` module of the

`statsmodels` library. This function requires the fitted model object and the specified number of lags (`nlags`) as input. The output provides several key statistical metrics necessary for hypothesis testing.

```
import statsmodels.stats.diagnostic as dg
```

```
#perform Breusch-Godfrey test at order  $p = 3$ 
```

```
print(dg.acorr_breusch_godfrey(model, nlags=3))
```

```
(8.70314827, 0.0335094873, 5.27967224, 0.0403980576)
```

Interpreting the Results and Making Decisions

The output of the `acorr_breusch_godfrey()` function returns a tuple containing four values. These values correspond to different forms of the test statistic and their associated p-values, based on two different distributions (Chi-Square and F-distribution). For general interpretation using the Lagrange Multiplier approach, we focus on the first two values: the test statistic and its corresponding P-value from the Chi-Square distribution.

Specifically, the output shows the following key results:

Test statistic X2 (Lagrange Multiplier Statistic) = **8.7031**

P-value (Chi-Square) = **0.0335**

To reach a conclusion, we compare the calculated P-value (0.0335) against our chosen significance level (α). Assuming a conventional significance level of $\alpha = 0.05$, we observe that the p-value is less than 0.05 ($0.0335 < 0.05$). This result provides statistically significant evidence to reject the null hypothesis (H_0). Consequently, we conclude that significant Autocorrelation exists among the residuals of the fitted model at some order less than or equal to 3. This indicates that the standard errors derived from our OLS model are likely unreliable.

Strategies for Addressing Autocorrelation

When the Breusch-Godfrey Test confirms the presence of serial correlation, analysts must take remedial actions to restore the validity of their statistical inferences. The approach taken depends heavily on whether the correlation is positive, negative, or seasonal in nature, and whether it is intrinsic to the data generation process or a symptom of model misspecification.

Several techniques can be employed to correct or mitigate the effects of autocorrelation, ranging from model adjustments to using more robust estimation methods. Choosing the best strategy requires careful consideration of the underlying economic or physical theory driving the time series.

Model Re-specification: For positive serial correlation, often the best approach is to improve the model structure itself. This involves adding lagged values of the dependent variable (e.g., Y_{t-1}) and/or lagged values of the independent variables (e.g., X_{t-1}) to the Regression Model. This adjustment often captures the dynamic processes that were previously relegated to the error term.

Addressing Negative Correlation: If negative serial correlation is detected, the analyst should investigate whether any variables have been *overdifferenced*. Differencing is a common technique to achieve stationarity, but over-differencing can introduce an artificial negative correlation structure into the data.

Seasonal Effects: For data exhibiting clear seasonal patterns, serial correlation might manifest at specific seasonal lags (e.g., lag 4 for quarterly data or lag 12 for monthly data). In this case, incorporating seasonal dummy variables or applying seasonal differencing can effectively remove the structure from the residuals.

Using Robust Standard Errors: If model re-specification proves challenging or undesirable, an alternative is to use estimation methods that provide standard errors robust to autocorrelation. Techniques like Newey-West standard errors adjust the standard error estimates to account for serial correlation without changing the OLS coefficient estimates themselves, thus ensuring valid hypothesis testing.