How to Perform a Granger-Causality Test in Python: A Step-by-Step Guide

Authored by

stats writer

December 2, 2025

Granger-Causality Test is a statistical test used to determine if one time series is useful in predicting another. It assesses whether one time series can be used to predict the future values of another time series, and is typically applied when two time series are believed to be related. The test can be implemented in Python using statistical libraries such as StatsModels.

The **Granger Causality test** is used to determine whether or not one time series is useful for forecasting another.

This test uses the following null and alternative hypotheses:

Null Hypothesis (H0): Time series *x* does not Granger-cause time series *y*

Alternative Hypothesis (HA): Time series *x* Granger-causes time series *y*

The term "Granger-causes" means that knowing the value of time series *x* at a certain lag is useful for predicting the value of time series *y* at a later time period.

This test produces an F test statistic with a corresponding . If the p-value is less than a certain significance level (i.e. $\alpha = .05$), then we can reject the null hypothesis and conclude that we have sufficient evidence to say that time series *x* Granger-causes time series y.

We can use the function from the statsmodels package to perform a Granger-Causality test in Python:

from statsmodels.tsa.stattools import grangercausalitytests

#perform Granger-Causality test
grangercausalitytests(df], maxlag=)

Note that **maxlag** indicates the number of lags to use in the first time series.

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

Step 1: Load the Data

For this example, we'll use a dataset that contains values for the number of eggs manufactured along with the number of chickens in the U.S. from 1930 to 1983:

import pandas as pd

#define URL where dataset is located
url = "https://raw.githubusercontent.com/arabpsychology/Miscellaneous/main/chicken_egg.txt"

#read in dataset as pandas DataFrame
df = pd.read_csv(url, sep=" ")

#view first five rows of DataFrame
df.head()

year chicken egg
0 1930 468491 3581
1 1931 449743 3532
2 1932 436815 3327
3 1933 444523 3255
4 1934 433937 3156

Step 2: Perform the Granger-Causality Test

Next, we'll use the **grangercausalitytests()** function to perform a Granger-Causality test to see if the number of eggs manufactured is predictive of the future number of chickens. We'll run the test using three lags:

from statsmodels.tsa.stattools import grangercausalitytests

#perform Granger-Causality test
grangercausalitytests(df], maxlag=)

Granger Causality
number of lags (no zero) 3
ssr based F test: F=5.4050 , p=0.0030 , df_denom=44, df_num=3
ssr based chi2 test: chi2=18.7946 , p=0.0003 , df=3
likelihood ratio test: chi2=16.0003 , p=0.0011 , df=3
parameter F test: F=5.4050 , p=0.0030 , df_denom=44, df_num=3

Since the p-value is less than .05, we can reject the null hypothesis of the test and conclude that knowing the number of eggs is useful for predicting the future number of chickens.

Step 3: Perform the Granger-Causality Test in Reverse

Although we rejected the null hypothesis of the test, it's actually possible that there is a case of reverse causation happening. That is, it's possible that the number of chickens is causing the number of eggs to change.

To rule out this possibility, we need to perform the Granger-Causality test in reverse, using chickens as the predictor variable and eggs as the :

from statsmodels.tsa.stattools import grangercausalitytests

#perform Granger-Causality test
grangercausalitytests(df], maxlag=)

Granger Causality
number of lags (no zero) 3
ssr based F test: F=0.5916 , p=0.6238 , df_denom=44, df_num=3
ssr based chi2 test: chi2=2.0572 , p=0.5606 , df=3
likelihood ratio test: chi2=2.0168 , p=0.5689 , df=3
parameter F test: F=0.5916 , p=0.6238 , df_denom=44, df_num=3

The F test statistic turns out to be **0.5916** and the corresponding p-value is **0.6238**.

Since the p-value isn't less than .05, we can't reject the null hypothesis. That is, the number of chickens isn't predictive of the future number of eggs.

Thus, we can conclude that knowing the number of eggs is useful for predicting the future number of chickens.

The following tutorials explain how to perform other common tasks with time series in Python: