

How can I perform a Ljung-Box Test using Python?

Authored by
stats writer

April 20, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I perform a Ljung-Box Test using Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=137513>

The Ljung-Box Test is a statistical method used to test for the presence of autocorrelation in a dataset. This test is commonly used in time series analysis to determine if a series of data points are related to each other. To perform a Ljung-Box Test using Python, one can use the statsmodels library, which includes a function for conducting the test. This function takes in the dataset as an input and returns the test statistic and corresponding p-value. By comparing the p-value to a chosen significance level, one can determine if there is evidence of autocorrelation in the data. The Ljung-Box Test in Python provides a simple and efficient way to assess the presence of autocorrelation, thus aiding in the accurate analysis of time series data.

Perform a Ljung-Box Test in Python

The Ljung-Box test is a statistical test that checks if autocorrelation exists in a time series.

It uses the following hypotheses:

H0: The residuals are independently distributed.

HA: The residuals are not independently distributed; they exhibit serial correlation.

Ideally, we would like to fail to reject the null hypothesis. That is, we would like to see the p-value of the test be greater than 0.05 because this means the residuals for our time series model are independent, which is often an assumption we make when creating a model.

This tutorial explains how to perform a Ljung-Box test

in Python.

Example: Ljung-Box Test in Python

To perform the Ljung-Box test on a data series in Python, we can use the `acorr_ljungbox()` function from the statsmodels library which uses the following syntax:

```
acorr_ljungbox(x, lags=None)
```

where:

x: The data series
lags: Number of lags to test

This function returns a test statistic and a corresponding p-value. If the p-value is less than some threshold (e.g. $\alpha = .05$), you can reject the null hypothesis and conclude that the residuals are not independently distributed.

The following code shows how to use this function to perform the Ljung-Box test on the built-in statsmodels dataset called "SUNACTIVITY":

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

```
#load data series
```

```
data = sm.datasets.sunspots.load_pandas().data
```

```
#view first ten rows of data series
```

```
data
```

```
YEAR SUNACTIVITY
```

```
0 1700.0 5.0
```

```
1 1701.0 11.0
```

```
2 1702.0 16.0
```

```
3 1703.0 23.0
```

```
4 1704.0 36.0
```

```
#fit ARMA model to dataset
```

```
res = sm.tsa.ARMA(data, (1,1)).fit(dispatch=-1)
```

```
#perform Ljung-Box test on residuals with lag=5
```

```
sm.stats.acorr_ljungbox(res.resid, lags=, return_df=True)
```

```
lb_stat lb_pvalue
```

```
5 107.86488 1.157710e-21
```

The test statistic of the test is 107.86488 and the p-value of the test is 1.157710e-21, which is much less than

0.05. Thus, we reject the null hypothesis of the test and conclude that the residuals are not independent.

Note that we chose to use a lag value of 5 in this example, but you can choose any value that you would like to use for the lag. For example, we could instead use a value of 20:

```
#perform Ljung-Box test on residuals with lag=20  
sm.stats.acorr_ljungbox(res.resid, lags=,  
return_df=True)
```

```
lb_stat lb_pvalue  
20 343.634016 9.117477e-61
```

Depending on your particular situation you may choose a lower or higher value to use for the lag.