

How can a Durbin-Watson test be performed in Python?

Authored by
stats writer

April 17, 2024

RECOMMENDED CITATION

stats writer (2024). *How can a Durbin-Watson test be performed in Python?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136242>

The Durbin-Watson test is a statistical test used to detect autocorrelation in a dataset. It is commonly performed in Python by using the built-in function available in the Statsmodels library. The process involves importing the necessary libraries, loading the dataset, and fitting a regression model. Then, the Durbin-Watson statistic can be calculated using the appropriate function from Statsmodels. The output of the test can be interpreted to determine the presence and strength of autocorrelation in the data. This test is a useful tool for analyzing time series data or any dataset where the observations may be correlated.

Perform a Durbin-Watson Test in Python

One of the is that there is no correlation between the residuals. In other words, the residuals are assumed to be independent.

One way to determine if this assumption is met is to perform a , which is used to detect the presence of autocorrelation in the residuals of a . This test uses the following hypotheses:

H0 (null hypothesis): There is no correlation among the residuals.

HA (alternative hypothesis): The residuals are autocorrelated.

The test statistic is approximately equal to $2*(1-r)$ where r is the sample autocorrelation of the residuals. Thus, the test statistic will always be between 0 and 4 with the

following interpretation:

A test statistic of 2 indicates no serial correlation. The closer the test statistics is to 0, the more evidence of positive serial correlation. The closer the test statistics is to 4, the more evidence of negative serial correlation.

As a rule of thumb, test statistic values between the range of 1.5 and 2.5 are considered normal. However, values outside of this range could indicate that autocorrelation is a problem.

This tutorial explains how to perform a Durbin-Watson test in Python.

Example: Durbin-Watson Test in Python

Suppose we have the following dataset that describes the attributes of 10 basketball players:

```
import numpy as np  
import pandas as pd  
  
#create dataset  
df = pd.DataFrame({'rating': ,  
'points': ,
```

```
'assists': ,  
'rebounds': })
```

```
#view dataset
```

```
df
```

```
rating points assists rebounds
```

```
0 90 25 5 11
```

```
1 85 20 7 8
```

```
2 82 14 7 10
```

```
3 88 16 8 6
```

```
4 94 27 5 6
```

```
5 90 20 7 9
```

```
6 76 12 6 6
```

```
7 75 15 9 10
```

```
8 87 14 9 10
```

```
9 86 19 5 7
```

Suppose we fit a multiple linear regression model using *rating* as the response variable and the other three variables as the predictor variables:

```
from statsmodels.formula.api import ols
```

```
#fit multiple linear regression model
```

```
model = ols('rating ~ points + assists + rebounds',  
data=df).fit()
```

```
#view model summary  
print(model.summary())
```

We can perform a Durbin Watson using the `statsmodels` library to determine if the residuals of the regression model are autocorrelated:

```
from statsmodels.stats.stattools import durbin_watson
```

```
#perform Durbin-Watson test  
durbin_watson(model.resid)
```

2.392

The test statistic is 2.392. Since this is within the range of 1.5 and 2.5, we would consider autocorrelation not to be problematic in this regression model.

How to Handle Autocorrelation

1. For positive serial correlation, consider adding lags of the dependent and/or independent variable to the model.

2. For negative serial correlation, check to make sure that none of your variables are *overdifferenced*.

3. For seasonal correlation, consider adding seasonal dummy variables to the model.

ARABPSYCHOLOGY.COM