

How can I use Python to calculate the residual sum of squares?

Authored by
stats writer

April 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use Python to calculate the residual sum of squares?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=138694>

Python is a high-level programming language that offers a variety of tools and functions for data analysis and manipulation. One of these tools is the ability to calculate the residual sum of squares, which is a measure of the overall difference between a set of data points and a mathematical model. By utilizing built-in functions and libraries, users can easily input their data and model equations into Python to calculate the residual sum of squares. This allows for efficient and accurate analysis of data and can assist in identifying any discrepancies or patterns within the data set.

Calculate Residual Sum of Squares in Python

A is the difference between an observed value and a predicted value in a regression model.

It is calculated as:

Residual = Observed value - Predicted value

One way to understand how well a regression model fits a dataset is to calculate the residual sum of squares, which is calculated as:

Residual sum of squares = $\sum(e_i)^2$

where:

Σ : A Greek symbol that means "sum"

e_i : The i th residual

The lower the value, the better a model fits a dataset.

This tutorial provides a step-by-step example of how to calculate the residual sum of squares for a regression model in Python.

Step 1: Enter the Data

For this example we'll enter data for the number of hours spent studying, total prep exams taken, and exam score received by 14 different students:

```
import pandas as pd
```

```
#create DataFrame
```

```
df = pd.DataFrame({'hours': ,  
'exams': ,  
'score': })
```

Step 2: Fit the Regression Model

Next, we'll use the OLS() function from the statsmodels library to perform ordinary least squares regression, using "hours" and "exams" as the predictor variables and "score" as the response variable:

```
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()
```

```
#view model summary
```

```
print(model.summary())
```

```
OLS Regression Results
```

```
=====
```

```
Dep. Variable: score R-squared: 0.722
```

```
Model: OLS Adj. R-squared: 0.671
```

```
Method: Least Squares F-statistic: 14.27
```

```
Date: Sat, 02 Jan 2021 Prob (F-statistic): 0.000878
```

```
Time: 15:58:35 Log-Likelihood: -41.159
```

```
No. Observations: 14 AIC: 88.32
```

```
Df Residuals: 11 BIC: 90.24
```

```
Df Model: 2
```

Covariance Type: nonrobust

```
=====
=====
coef std err t P>|t|
-----
const 71.8144 3.680 19.517 0.000 63.716 79.913
hours 5.0318 0.942 5.339 0.000 2.958 7.106
exams -1.3186 1.063 -1.240 0.241 -3.658 1.021
=====
=====
Omnibus: 0.976 Durbin-Watson: 1.270
Prob(Omnibus): 0.614 Jarque-Bera (JB): 0.757
Skew: -0.245 Prob(JB): 0.685
Kurtosis: 1.971 Cond. No. 12.1
=====
=====
```

Step 3: Calculate the Residual Sum of Squares

We can use the following code to calculate the residual sum of squares for the model:

```
print(model.ssr)
```

```
293.25612951525414
```

The residual sum of squares turns out to be 293.256.

ARABPSYCHOLOGY.COM