

How can VIF be calculated in Python?

Authored by
stats writer

April 17, 2024

RECOMMENDED CITATION

stats writer (2024). *How can VIF be calculated in Python?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=136239>

The Variance Inflation Factor (VIF) is a statistical measure used to assess the severity of multicollinearity in regression analysis. In simple terms, it measures how much the variance of a particular variable is inflated due to its correlation with other independent variables in the model. In Python, the VIF can be calculated using the statsmodels library, which provides a function called "variance_inflation_factor". This function takes in a matrix of independent variables and returns a list of VIF values for each variable. By comparing the VIF values against a certain threshold, one can determine if there is significant multicollinearity present in the model. Therefore, using the variance_inflation_factor function in Python can help researchers and data analysts identify and address any issues related to multicollinearity in their regression models.

Calculate VIF in Python

in regression analysis occurs when two or more explanatory variables are highly correlated with each other, such that they do not provide unique or independent information in the regression model.

If the degree of correlation is high enough between variables, it can cause problems when fitting and interpreting the regression model.

One way to detect multicollinearity is by using a metric known as the variance inflation factor (VIF), which measures the correlation and strength of correlation between the explanatory variables in a .

This tutorial explains how to calculate VIF in Python.

Example: Calculating VIF in Python

For this example we'll use a 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 would like to fit a multiple linear regression model using rating as the response variable and points, assists, and rebounds as the explanatory variables.

To calculate the VIF for each explanatory variable in the model, we can use the from the statsmodels library:

```
from patsy import dmatrices
from statsmodels.stats.outliers_influence import
variance_inflation_factor
```

```
#find design matrix for linear regression model using
'rating' as response variable
```

```
y, X = dmatrices('rating ~ points+assists+rebounds',
data=df, return_type='dataframe')
```

```
#calculate VIF for each explanatory variable
```

```
vif = pd.DataFrame()
```

```
vif = )]
```

```
vif = X.columns
```

#view VIF for each explanatory variable
vif

VIF variable

0 101.258171 Intercept

1 1.763977 points

2 1.959104 assists

3 1.175030 rebounds

We can observe the VIF values for each of the explanatory variables:

points: 1.76assists: 1.96rebounds: 1.18

Note: Ignore the VIF for the "Intercept" in the model since this value is irrelevant.

How to Interpret VIF Values

The value for VIF starts at 1 and has no upper limit. A general rule of thumb for interpreting VIFs is as follows:

A value of 1 indicates there is no correlation between a given explanatory variable and any other explanatory variables in the model. A value between 1 and 5 indicates moderate correlation between a given

explanatory variable and other explanatory variables in the model, but this is often not severe enough to require attention. A value greater than 5 indicates potentially severe correlation between a given explanatory variable and other explanatory variables in the model. In this case, the coefficient estimates and p-values in the regression output are likely unreliable.

Given that each of the VIF values for the explanatory variables in our regression model are close to 1, multicollinearity is not a problem in our example.