How can I find the p-value of a correlation coefficient in Pandas?

Authored by

stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I find the p-value of a correlation coefficient in Pandas?*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=151352

ARABPSYCHOLOGY.COM

To find the p-value of a correlation coefficient in Pandas, one can use the "corr" function in conjunction with the "corrcoef" method. This will calculate the correlation coefficient and its corresponding p-value, which measures the significance of the relationship between two variables. The p-value can also be interpreted as the probability of obtaining a correlation coefficient at least as extreme as the observed value, assuming the null hypothesis is true. By utilizing these functions in Pandas, one can easily determine the strength and significance of a correlation between two variables.

Find P-value of Correlation Coefficient in Pandas

The can be used to measure the linear association between two variables.

This correlation coefficient always takes on a value between -1 and 1 where:

-1: Perfectly negative linear correlation between two variables.0: No linear correlation between two variables.1: Perfectly positive linear correlation between two variables.

To determine if a correlation coefficient is statistically significant, you can calculate the corresponding t-score and p-value.

The formula to calculate the t-score of a correlation coefficient (r) is:

ARABPSYCHOLOGY.COM

t = r√n-2 / √1-r2

The p-value is calculated as the corresponding two-sided p-value for the t-distribution with n-2 degrees of freedom.

To calculate the p-value for a Pearson correlation coefficient in pandas, you can use the pearsonr() function from the SciPy library:

```
from scipy.statsimport pearsonr

pearsonr(df, df)
```

This function will return the Pearson correlation coefficient between columns column1 and column2 along with the corresponding p-value that tells us whether or not the correlation coefficient is statistically significant.

If you would like to calculate the p-value for the Pearson correlation coefficient of each possible pairwise combination of columns in a DataFrame, you can use the following custom function to do so:

```
def r_pvalues(df):
```

ARABPSYCHOLOGY.COM

```
cols = pd.DataFrame(columns=df.columns)
p = cols.transpose().join(cols, how='outer')
for r in df.columns:
for c in df.columns:
tmp = df.notnull() & df.notnull()]
p = round(pearsonr(tmp, tmp), 4)
return p
```

The following examples show how to calculate p-values for correlation coefficients in practice with the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'x': ,
'y': ,
'z': })

#view DataFrame
print(df)
```

x y z

0 4 10.0 20

1 5 12.0 24

ARABPSYCHOLOGY.COM

2 5 14.0 24

3 7 18.0 23

4 8 NaN 19

5 10 19.0 15

6 12 13.0 18

7 13 20.0 14

8 14 14.0 10

9 15 NaN 12

Example 1: Calculate P-Value for Correlation Coefficient Between Two Columns in Pandas

The following code shows how to calculate the Pearson correlation coefficient and corresponding p-value for the x and y columns in the DataFrame:

```
from scipy.statsimport pearsonr

#drop all rows with NaN values
df_new = df.dropna()

#calculation correlation coefficient and p-value between x and y
pearsonr(df_new, df_new)

PearsonRResult(statistic=0.479162198588383,
```

ARABPSYCHOLOGY.COM

pvalue=0.22961622926360523)

The Pearson correlation coefficient is 0.4792.The corresponding p-value is 0.2296.

Since the correlation coefficient is positive, it indicates that there is a positive linear relationship between the two variables.

However, since the p-value of the correlation coefficient is not less than 0.05, the correlation is not statistically significant.

Note that we can also use the following syntax to extract the p-value for the correlation coefficient:

#extract p-value of correlation coefficient
pearsonr(df_new, df_new)

0.22961622926360523

The p-value for the correlation coefficient is 0.2296.

This matches the p-value from the previous output.

ARABPSYCHOLOGY.COM

Example 2: Calculate P-Value for Correlation Coefficient Between All Columns in Pandas

The following code shows how to calculate the Pearson correlation coefficient and corresponding p-value for each pairwise combination of columns in the pandas DataFrame:

```
#create function to calculate p-values for each pairwise correlation coefficient
def r_pvalues(df):
cols = pd.DataFrame(columns=df.columns)
p = cols.transpose().join(cols, how='outer')
for r in df.columns:
for c in df.columns:
tmp = df.notnull() & df.notnull()]
p = round(pearsonr(tmp, tmp), 4)
return p

#use custom function to calculate p-values
r_pvalues(df)

x y z
x 0.0 0.2296 0.0005
y 0.2296 0.0 0.4238
```

ARABPSYCHOLOGY.COM

z 0.0005 0.4238 0.0

From the output we can see:

The p-value for the correlation coefficient between x and y is 0.2296.The p-value for the correlation coefficient between x and z is 0.0005.The p-value for the correlation coefficient between y and z is 0.4238.

Note that we rounded the p-values to four decimal places in our custom function.

Feel free to change the 4 in the last line of the function to a different number to round to a different number of decimal places.

Note: You can find the complete documentation for the SciPy pearsonr() function .

The following tutorials explain how to perform other common tasks in pandas:

ARABPSYCHOLOGY.COM