

How can I test for normality in Python using four different methods?

Authored by
stats writer

June 28, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I test for normality in Python using four different methods?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=155881>

Testing for normality is a crucial step in statistical analysis to determine whether a dataset follows a normal distribution. In Python, there are four commonly used methods for testing normality: Shapiro-Wilk test, D'Agostino and Pearson's test, Anderson-Darling test, and Kolmogorov-Smirnov test. Each method has its own assumptions and limitations, but they all aim to assess the data's similarity to a normal distribution. By using these four methods, researchers can have a more comprehensive understanding of the normality of their dataset and make informed decisions for further analysis.

Test for Normality in Python (4 Methods)

Many statistical tests make the that datasets are normally distributed.

There are four common ways to check this assumption in Python:

1. (Visual Method) Create a histogram.

If the histogram is roughly "bell-shaped", then the data is assumed to be normally distributed.

2. (Visual Method) Create a Q-Q plot.

If the points in the plot roughly fall along a straight diagonal line, then the data is assumed to be normally distributed.

3. (Formal Statistical Test) Perform a Shapiro-Wilk Test.

If the p-value of the test is greater than $\alpha = .05$, then the data is assumed to be normally distributed.

4. (Formal Statistical Test) Perform a Kolmogorov-Smirnov Test.

If the p-value of the test is greater than $\alpha = .05$, then the data is assumed to be normally distributed.

The following examples show how to use each of these methods in practice.

Method 1: Create a Histogram

The following code shows how to create a histogram for a dataset that follows a :

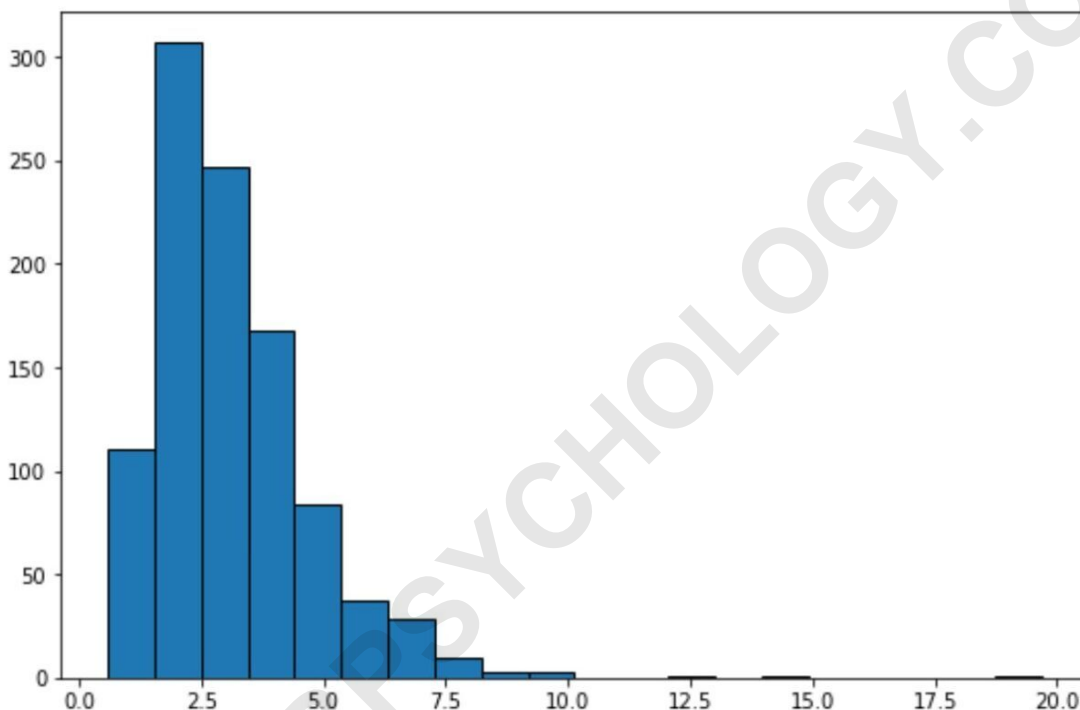
```
import math
import numpy as np
from scipy.stats import lognorm
import matplotlib.pyplot as plt

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal
distributed values
```

```
lognorm_dataset = lognorm.rvs(s=.5,  
scale=math.exp(1), size=1000)
```

```
#create histogram to visualize values in dataset  
plt.hist(lognorm_dataset, edgecolor='black', bins=20)
```



By simply looking at this histogram, we can tell the dataset does not exhibit a "bell-shape" and is not normally distributed.

Method 2: Create a Q-Q plot

```
import math  
import numpy as np
```

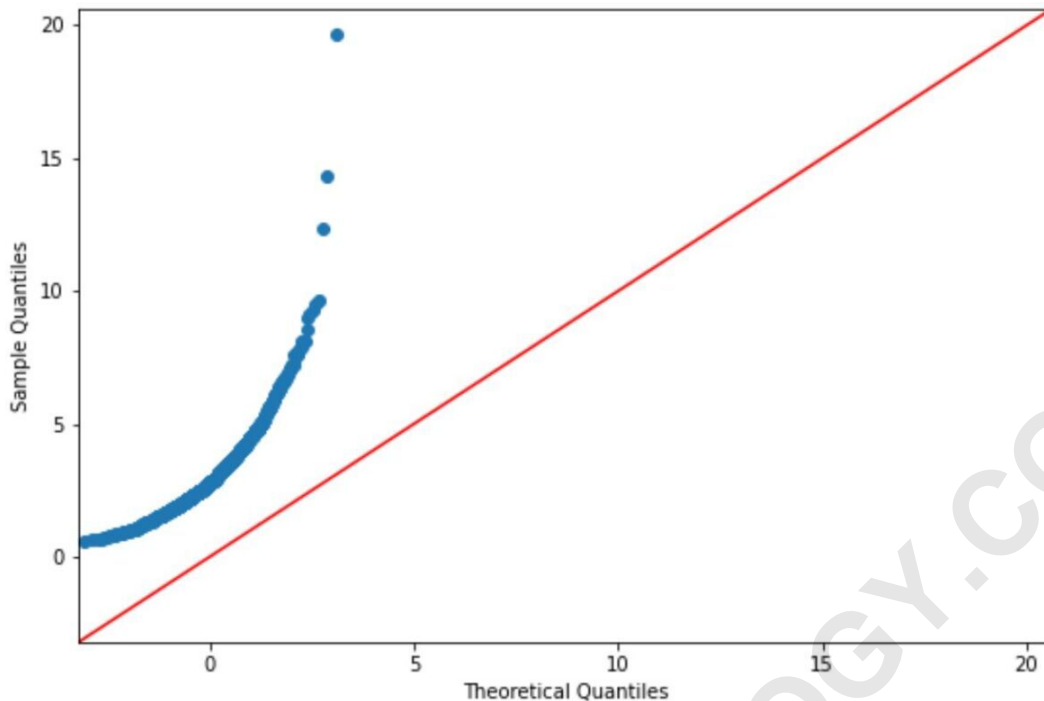
```
from scipy.stats import lognorm
import statsmodels.api as sm
import matplotlib.pyplot as plt

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal
distributed values
lognorm_dataset = lognorm.rvs(s=.5,
scale=math.exp(1), size=1000)

#create Q-Q plot with 45-degree line added to plot
fig = sm.qqplot(lognorm_dataset, line='45')

plt.show()
```



If the points on the plot fall roughly along a straight diagonal line, then we typically assume a dataset is normally distributed.

However, the points on this plot clearly don't fall along the red line, so we would not assume that this dataset is normally distributed.

This should make sense considering we generated the data using a log-normal distribution function.

Method 3: Perform a Shapiro-Wilk Test

The following code shows how to perform a Shapiro-

Wilk for a dataset that follows a log-normal distribution:

```
import math
import numpy as np
from scipy.stats import shapiro
from scipy.stats import lognorm

#make this example reproducible
np.random.seed(1)

#generate dataset that contains 1000 log-normal
distributed values
lognorm_dataset = lognorm.rvs(s=.5,
scale=math.exp(1), size=1000)

#perform Shapiro-Wilk test for normality
shapiro(lognorm_dataset)

ShapiroResult(statistic=0.8573324680328369,
pvalue=3.880663073872444e-29)
```

From the output we can see that the test statistic is 0.857 and the corresponding p-value is 3.88e-29 (extremely close to zero).

Since the p-value is less than .05, we reject the null

hypothesis of the Shapiro-Wilk test.

This means we have sufficient evidence to say that the sample data does not come from a normal distribution.

Method 4: Perform a Kolmogorov-Smirnov Test

The following code shows how to perform a Kolmogorov-Smirnov test for a dataset that follows a log-normal distribution:

```
import math  
import numpy as np  
from scipy.stats import kstest  
from scipy.stats import lognorm  
  
#make this example reproducible  
np.random.seed(1)  
  
#generate dataset that contains 1000 log-normal  
distributed values  
lognorm_dataset = lognorm.rvs(s=.5,  
scale=math.exp(1), size=1000)  
  
#perform Kolmogorov-Smirnov test for normality  
kstest(lognorm_dataset, 'norm')
```

KstestResult(statistic=0.84125708308077, pvalue=0.0)

From the output we can see that the test statistic is 0.841 and the corresponding p-value is 0.0.

Since the p-value is less than .05, we reject the null hypothesis of the Kolmogorov-Smirnov test.

This means we have sufficient evidence to say that the sample data does not come from a normal distribution.

How to Handle Non-Normal Data

If a given dataset is *not* normally distributed, we can often perform one of the following transformations to make it more normally distributed:

1. **Log Transformation:** Transform the values from x to $\log(x)$.
2. **Square Root Transformation:** Transform the values from x to \sqrt{x} .
3. **Cube Root Transformation:** Transform the values from x to $x^{1/3}$.

By performing these transformations, the dataset

typically becomes more normally distributed.

Read [this tutorial](#) to see how to perform these transformations in Python.

ARABPSYCHOLOGY.COM