

# How can I create frequency tables in Python?

Authored by  
**stats writer**

April 17, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I create frequency tables in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=136188>

Creating frequency tables in Python involves using the built-in functions and libraries available in the programming language. First, the data must be imported into Python using the appropriate function or library, such as pandas or numpy. Then, the data can be organized and sorted into categories using functions like groupby or value\_counts. Finally, the frequency table can be created by counting the number of occurrences of each category and displaying the results in a table format. Additional steps may be necessary depending on the specific data and desired output.

## Create Frequency Tables in Python

**A frequency table is a table that displays the frequencies of different categories. This type of table is particularly useful for understanding the distribution of values in a dataset.**

**This tutorial explains how to create frequency tables in Python.**

### One-Way Frequency Table for a Series

**To find the frequencies of individual values in a pandas Series, you can use the value\_counts() function:**

```
import pandas as pd
```

```
#define Series
```

```
data = pd.Series()
```

```
#find frequencies of each value
```

## `data.value_counts()`

3 4

1 3

4 2

5 1

2 1

You can add the argument `sort=False` if you don't want the data values sorted by frequency:

## `data.value_counts(sort=False)`

1 3

2 1

3 4

4 2

5 1

The way to interpret the output is as follows:

The value "1" occurs 3 times in the Series. The value "2" occurs 1 time in the Series. The value "3" occurs 4 times in the Series.

**And so on.**

**One-Way Frequency Table for a DataFrame**

**To find frequencies of a pandas DataFrame you can use the `crosstab()` function, which uses the following syntax:**

**`crosstab(index, columns)`**

**where:**

**index: name of column to group by  
columns: name to give to frequency column**

**For example, suppose we have a DataFrame with information about the letter grade, age, and gender of 10 different students in a class. Here's how to find the frequency for each letter grade:**

```
#create data  
df = pd.DataFrame({'Grade': ,  
'Age': ,  
'Gender': })
```

```
#view data
```

```
df
```

## Grade Age Gender

0 A 18 M

1 A 18 M

2 A 18 F

3 B 19 F

4 B 19 F

5 B 20 M

6 B 18 M

7 C 18 F

8 D 19 M

9 D 19 F

#find frequency of each letter grade

```
pd.crosstab(index=df, columns='count')
```

```
col_0 count
```

```
Grade
```

```
A 3
```

```
B 4
```

```
C 1
```

```
D 2
```

3 students received an 'A' in the class.4 students received a 'B' in the class.1 student received a 'C' in the

**class.2 students received a 'D' in the class.**

**We can use a similar syntax to find the frequency counts for other columns. For example, here's how to find frequency by age:**

```
pd.crosstab(index=df, columns='count')
```

```
col_0 count
```

```
Age
```

```
18 5
```

```
19 4
```

```
20 1
```

**The way to interpret this is as follows:**

**5 students are 18 years old.4 students are 19 years old.1 student is 20 years old.**

**You can also easily display the frequencies as proportions of the entire dataset by dividing by the sum:**

```
#define crosstab
```

```
tab = pd.crosstab(index=df, columns='count')
```

```
#find proportions
```

```
tab/tab.sum()
```

```
col_0 count
```

```
Age
```

```
18 0.5
```

```
19 0.4
```

```
20 0.1
```

The way to interpret this is as follows:

50% of students are 18 years old.40% of students are 19 years old.10% of students are 20 years old.

Two-Way Frequency Tables for a DataFrame

You can also create a two-way frequency table to display the frequencies for two different variables in the dataset. For example, here's how to create a two-way frequency table for the variables Age and Grade:

```
pd.crosstab(index=df, columns=df)
```

```
Grade A B C D
```

```
Age
```

```
18 3 1 1 0
```

**19 0 2 0 2**

**20 0 1 0 0**

**The way to interpret this is as follows:**

**There are 3 students who are 18 years old and received an 'A' in the class. There is 1 student who is 18 years old and received a 'B' in the class. There is 1 student who is 18 years old and received a 'C' in the class. There are students who are 18 years old and received a 'D' in the class.**

**And so on.**

***You can find the complete documentation for the `crosstab()` function .***