

# How can a contingency table be created in Python?

Authored by  
**stats writer**

April 27, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can a contingency table be created in Python?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=139869>

A contingency table in Python can be created by using the pandas library. This library has a built-in function called "crosstab" which takes in two variables and generates a table showing the frequency distribution of each variable. The first variable is typically the row variable, while the second variable is the column variable. The resulting table displays the count of each combination of the two variables, making it useful for analyzing the relationship between categorical variables. By utilizing the crosstab function, a contingency table can easily be created in Python for data analysis and visualization purposes.

## Create a Contingency Table in Python

**A contingency table is a type of table that summarizes the relationship between two categorical variables.**

**To create a contingency table in Python, we can use the function, which uses the following syntax:**

```
pandas.crosstab(index, columns)
```

**where:**

**index:** name of variable to display in the rows of the contingency table  
**columns:** name of variable to display in the columns of the contingency table

**The following step-by-step example shows how to use this function to create a contingency table in Python.**

**Step 1: Create the Data**

First, let's create a dataset that shows information for 20 different product orders, including the type of product purchased (TV, computer, or radio) along with the country (A, B, or C) that the product was purchased in:

```
import pandas as pd
```

```
#create data
```

```
df = pd.DataFrame({'Order': ,  
'Product': ,  
'Country': })
```

```
#view data
```

```
df
```

```
Order Product Country
```

```
0 1 TV A
```

```
1 2 TV A
```

```
2 3 Comp A
```

```
3 4 TV A
```

```
4 5 TV B
```

```
5 6 Comp B
```

```
6 7 Comp B
```

```
7 8 Comp B
```

**8 9 TV B**

**9 10 Radio B**

**10 11 TV B**

**11 12 Radio B**

**12 13 Radio C**

**13 14 Radio C**

**14 15 Comp C**

**15 16 Comp C**

**16 17 TV C**

**17 18 TV C**

**18 19 Radio C**

**19 20 TV C**

**Step 2: Create the Contingency Table**

**The following code shows how to create a contingency table to count the number of each product ordered by each country:**

**#create contingency table**

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

**Product Comp Radio TV**

**Country**

**A 1 0 3**

**B 3 2 3**

**C 2 3 3**

**Here's how to interpret the table:**

**A total of 1 computer was purchased from country A. A total of 3 computers were purchased from country B. A total of 2 computers were purchased from country C. A total of 0 radios were purchased from country A. A total of 2 radios were purchased from country B. A total of 3 radios were purchased from country C. A total of 3 TV's were purchased from country A. A total of 3 TV's were purchased from country B. A total of 3 TV's were purchased from country C.**

**Step 3: Add Margin Totals to the Contingency Table**

**We can use the argument `margins=True` to add the margin totals to the contingency table:**

```
#add margins to contingency table
pd.crosstab(index=df, columns=df, margins=True)
```

**Product Comp Radio TV All**

**Country**

**A 1 0 3 4**

**B 3 2 3 8**

**C 2 3 3 8**

**All 6 5 9 20**

### **Row Totals:**

**A total of 4 orders were made from country A. A total of 8 orders were made from country B. A total of 8 orders were made from country C.**

### **Column Totals:**

**A total of 6 computers were purchased. A total of 5 radios were purchased. A total of 9 TV's were purchased.**

**The value in the bottom right corner of the table shows that a total of 20 products were ordered from all countries.**