

# Crosstab Tutorial

```
from google.colab import drive
drive.mount('/content/drive')
```

➞ Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6qk8qdgf4n4g3pfee6491f](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491f)

Enter your authorization code:

.....

Mounted at /content/drive

```
import pandas as pd
df = pd.read_excel("/content/drive/My Drive/pandas/pandas/13_crosstab/survey.xls")
df
```

➞

	Name	Nationality	Sex	Age	Handedness
0	Kathy	USA	Female	23	Right
1	Michael	USA	Female	40	Right

```
pd.crosstab(df.Nationality,df.Handedness) # rows and columns
```

```
↳ Handedness Left Right
```

Nationality

<b>Bangladesh</b>	2	0
<b>China</b>	2	1
<b>India</b>	2	1
<b>USA</b>	1	3

```
9 Yan China Female 52 Right
```

```
pd.crosstab(df.Handedness,df.Nationality)
```

```
↳ Nationality Bangladesh China India USA
```

Handedness

<b>Left</b>	2	2	2	1
<b>Right</b>	0	1	1	3

```
pd.crosstab(df.Sex,df.Handedness)
```

```
↳ Handedness Left Right
```

Sex

<b>Female</b>	2	3
<b>Male</b>	5	2

## Margins

```
pd.crosstab(df.Sex,df.Handedness, margins=True)
```

↳

Handedness	Left	Right	All
Sex			
Female	2	3	5
Male	5	2	7
All	7	5	12

## Multi Index Column and Rows

```
pd.crosstab(df.Sex, [df.Handedness,df.Nationality], margins=True)
```

↳

Sex	Handedness	Left				Right			All
	Nationality	Bangadesh	China	India	USA	China	India	USA	
Female		1	1	0	0	1	0	2	5
Male		1	1	2	1	0	1	1	7
All		2	2	2	1	1	1	3	12

```
pd.crosstab([df.Nationality, df.Sex], [df.Handedness], margins=True)
```

↳

	Handedness	Left	Right	All
Nationality	Sex			
Bangladesh	Female	1	0	1
	Male	1	0	1
China	Female	1	1	2
	Male	1	0	1
India	Male	2	1	3
USA	Female	0	2	2
	Male	1	1	2

## Normalize

```
pd.crosstab(df.Sex, df.Handedness, normalize='index')
```

↳

Handedness	Left	Right
Sex		
Female	0.400000	0.600000
Male	0.714286	0.285714

## Aggfunc and Values

```
import numpy as np
pd.crosstab(df.Sex, df.Handedness, values=df.Age, aggfunc=np.average)
```

↳

Handedness	Left	Right
Sex		
Female	44.5	31.0

```
import pandas as pd
import seaborn as sns
```

```
↳ /usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprecate
import pandas.util.testing as tm
```

```
# Define the headers since the data does not have any
headers = ["symboling", "normalized_losses", "make", "fuel_type", "aspiration",
           "num_doors", "body_style", "drive_wheels", "engine_location",
           "wheel_base", "length", "width", "height", "curb_weight",
           "engine_type", "num_cylinders", "engine_size", "fuel_system",
           "bore", "stroke", "compression_ratio", "horsepower", "peak_rpm",
           "city_mpg", "highway_mpg", "price"]
```

```
# Read in the CSV file and convert "?" to NaN
df_raw = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/autos/i
                    header=None, names=headers, na_values="?" )
```

```
# Define a list of models that we want to review
models = ["toyota", "nissan", "mazda", "honda", "mitsubishi", "subaru", "volkswagen", "vo
```

```
# Create a copy of the data with only the top 8 manufacturers
df = df_raw[df_raw.make.isin(models)].copy()
```

```
pd.crosstab(df.make, df.body_style)
```

↳

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
<b>honda</b>	0	0	7	5	1
<b>mazda</b>	0	0	10	7	0
<b>mitsubishi</b>	0	0	9	4	0
<b>nissan</b>	0	1	5	9	3
<b>subaru</b>	0	0	3	5	4
<b>toyota</b>	1	3	14	10	4
<b>volkswagen</b>	1	0	1	9	1
<b>volvo</b>	0	0	0	8	3

```
df.groupby(['make', 'body_style'])['body_style'].count().unstack().fillna(0)
```

↳

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
<b>honda</b>	0.0	0.0	7.0	5.0	1.0
<b>mazda</b>	0.0	0.0	10.0	7.0	0.0
<b>mitsubishi</b>	0.0	0.0	9.0	4.0	0.0
<b>nissan</b>	0.0	1.0	5.0	9.0	3.0
<b>subaru</b>	0.0	0.0	3.0	5.0	4.0
<b>toyota</b>	1.0	3.0	14.0	10.0	4.0
<b>volkswagen</b>	1.0	0.0	1.0	9.0	1.0
<b>volvo</b>	0.0	0.0	0.0	8.0	3.0

```
df.pivot_table(index='make', columns='body_style', aggfunc='count', fill_val=0)
```

```
df.pivot_table(index= make , columns= body_style , aggfunc={ body_style :len}, fill_val
```



body\_style

body\_style   convertible   hardtop   hatchback   sedan   wagon

make

<b>honda</b>	0	0	7	5	1
<b>mazda</b>	0	0	10	7	0
<b>mitsubishi</b>	0	0	9	4	0
<b>nissan</b>	0	1	5	9	3
<b>subaru</b>	0	0	3	5	4
<b>toyota</b>	1	3	14	10	4
<b>volkswagen</b>	1	0	1	9	1
<b>volvo</b>	0	0	0	8	3

```
pd.crosstab(df.make, df.num_doors, margins=True, margins_name="Total")
```



```
num_doors  four  two  Total
```

```
pd.crosstab(df.make, df.body_style, values=df.curb_weight, aggfunc='mean').round(0)
```

↗

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
<b>honda</b>	NaN	NaN	1970.0	2289.0	2024.0
<b>mazda</b>	NaN	NaN	2254.0	2361.0	NaN
<b>mitsubishi</b>	NaN	NaN	2377.0	2394.0	NaN
<b>nissan</b>	NaN	2008.0	2740.0	2238.0	2452.0
<b>subaru</b>	NaN	NaN	2137.0	2314.0	2454.0
<b>toyota</b>	2975.0	2585.0	2370.0	2338.0	2708.0
<b>volkswagen</b>	2254.0	NaN	2221.0	2342.0	2563.0
<b>volvo</b>	NaN	NaN	NaN	3023.0	3078.0

```
pd.crosstab(df.make, df.body_style, normalize=True)
```

↗



```

    body_style  convertible  hardtop  hatchback    sedan    wagon
    make
pd.crosstab(df.make, df.body_style, normalize='columns')

```

↗

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
<b>honda</b>	0.0	0.00	0.142857	0.087719	0.0625
<b>mazda</b>	0.0	0.00	0.204082	0.122807	0.0000
<b>mitsubishi</b>	0.0	0.00	0.183673	0.070175	0.0000
<b>nissan</b>	0.0	0.25	0.102041	0.157895	0.1875
<b>subaru</b>	0.0	0.00	0.061224	0.087719	0.2500
<b>toyota</b>	0.5	0.75	0.285714	0.175439	0.2500
<b>volkswagen</b>	0.5	0.00	0.020408	0.157895	0.0625
<b>volvo</b>	0.0	0.00	0.000000	0.140351	0.1875

```
pd.crosstab(df.make, df.body_style, normalize='index')
```

↗

```
body_style convertible hardtop hatchback sedan wagon
make
```

```
pd.crosstab(df.make, [df.body_style, df.drive_wheels])
```

↗

body_style	convertible		hardtop		hatchback			sedan			wagon			
drive_wheels	fwd	rwd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	
make														
honda	0	0	0	0	0	7	0	0	5	0	0	1	0	
mazda	0	0	0	0	0	6	4	0	5	2	0	0	0	
mitsubishi	0	0	0	0	0	9	0	0	4	0	0	0	0	
nissan	0	0	1	0	0	2	3	0	9	0	0	3	0	
subaru	0	0	0	0	1	2	0	2	3	0	2	2	0	
toyota	0	1	0	3	0	8	6	0	7	3	2	1	1	
volkswagen	1	0	0	0	0	1	0	0	9	0	0	1	0	
volvo	0	0	0	0	0	0	0	0	0	8	0	0	3	

```
pd.crosstab([df.make, df.num_doors], [df.body_style, df.drive_wheels],
            rownames=['Auto Manufacturer', "Doors"],
            colnames=['Body Style', "Drive Type"],
            dropna=False)
```

↗

```
pd.crosstab(df.make, [df.body_style, df.drive_wheels], values=df.curb_weight, aggfunc='mean').fillna('-')
```

body_style	convertible		hardtop		hatchback		sedan			wagon			
drive_wheels	fwd	rwd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd
make													
honda	-	-	-	-	-	1970	-	-	2288.8	-	-	2024	-
mazda	-	-	-	-	-	2148.33	2411.25	-	2231.6	2685	-	-	-
mitsubishi	-	-	-	-	-	2376.56	-	-	2394	-	-	-	-
nissan	-	-	2008	-	-	2176	3116.33	-	2237.89	-	-	2452.33	-
subaru	-	-	-	-	2240	2085	-	2447.5	2225	-	2535	2372.5	-
toyota	-	2975	-	2585	-	2177.25	2626.83	-	2258.57	2521.67	2700	2280	2151
volkswagen	2254	-	-	-	-	2221	-	-	2342.22	-	-	2563	-
volvo	-	-	-	-	-	-	-	-	-	3023	-	-	3077.67

Average curb weight for all fwd toyota wagons

### Normalize All Values

```
pd.crosstab(df.make, df.body_style, normalize=True)
```

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
honda	0.000000	0.000000	0.054688	0.039062	0.007812
mazda	0.000000	0.000000	0.078125	0.054688	0.000000
mitsubishi	0.000000	0.000000	0.070312	0.031250	0.000000
nissan	0.000000	0.007812	0.039062	0.070312	0.023438
subaru	0.000000	0.000000	0.023438	0.039062	0.031250
toyota	0.007812	0.023438	0.109375	0.078125	0.031250
volkswagen	0.007812	0.000000	0.007812	0.070312	0.007812
volvo	0.000000	0.000000	0.000000	0.062500	0.023438

### Normalize Rows

```
pd.crosstab(df.make, df.body_style, normalize='index')
```

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
honda	0.000000	0.000000	0.538462	0.384615	0.076923
mazda	0.000000	0.000000	0.588235	0.411765	0.000000
mitsubishi	0.000000	0.000000	0.692308	0.307692	0.000000
nissan	0.000000	0.055556	0.277778	0.500000	0.166667
subaru	0.000000	0.000000	0.250000	0.416667	0.333333
toyota	0.031250	0.093750	0.437500	0.312500	0.125000
volkswagen	0.083333	0.000000	0.083333	0.750000	0.083333
volvo	0.000000	0.000000	0.000000	0.727273	0.272727

### Normalize Columns

```
pd.crosstab(df.make, df.body_style, normalize='columns')
```

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
honda	0.0	0.00	0.142857	0.087719	0.0625
mazda	0.0	0.00	0.204082	0.122807	0.0000
mitsubishi	0.0	0.00	0.183673	0.070175	0.0000
nissan	0.0	0.25	0.102041	0.157895	0.1875
subaru	0.0	0.00	0.061224	0.087719	0.2500
toyota	0.5	0.75	0.285714	0.175439	0.2500
volkswagen	0.5	0.00	0.020408	0.157895	0.0625
volvo	0.0	0.00	0.000000	0.140351	0.1875

Practical Business Python [pbpython.com/pandas-crosstab.html](http://pbpython.com/pandas-crosstab.html)



		Body Style	convertible			hardtop			hatchback			sedan			wagon		
		Drive Type	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd
Auto	Manufacturer	Doors															
	honda	four	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0
		two	0	0	0	0	0	0	0	7	0	0	1	0	0	0	0
	mazda	four	0	0	0	0	0	0	0	1	0	0	4	2	0	0	0
		two	0	0	0	0	0	0	0	5	4	0	0	0	0	0	0
	mitsubishi	four	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
		two	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0
	nissan	four	0	0	0	0	0	0	0	1	0	0	5	0	0	3	0
		two	0	0	0	0	1	0	0	1	3	0	4	0	0	0	0
	subaru	four	0	0	0	0	0	0	0	0	0	2	3	0	2	2	0
		two	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0
	toyota	four	0	0	0	0	0	0	0	6	0	0	7	1	2	1	1

```
pd.crosstab(df.make, [df.body_style, df.drive_wheels], values=df.curb_weight, aggfunc='me
```



body_style	convertible		hardtop		hatchback			sedan			wagon		
drive_wheels	fwd	rwd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd	4wd	fwd	rwd
make													
honda	-	-	-	-	-	1970	-	-	2288 8	-	-	2024	-

## Pandas crosstab explained

Source dataframe

make	body_style	drive_wheels	num_doors	curb_weight
mazda	hatchback	fwd	two	2385
volvo	sedan	rwd	four	2912
toyota	sedan	fwd	four	2140
mitsubishi	hatchback	fwd	two	1944
volkswagen	sedan	fwd	two	2261
mazda	sedan	fwd	four	2410
mazda	hatchback	rwd	two	2380
peugot	sedan	rwd	four	3197
mazda	sedan	fwd	four	1945
nissan	sedan	fwd	two	1918

Basic Usage

```
pd.crosstab(df.make, df.body_style)
```

body_style	convertible	hardtop	hatchback	sedan	wagon
make					
honda	0	0	7	5	1
mazda	0	0	10	7	0
mitsubishi	0	0	9	4	0
nissan	0	1	5	9	3
subaru	0	0	3	5	4
toyota	1	3	14	10	4
volkswagen	1	0	1	9	1
volvo	0	0	0	8	3

Margin Totals

```
pd.crosstab(df.make, df.num_doors,
            margins=True,
            margins_name="Total")
```

num_doors	four	two	Total
make			
honda	5	8	13
mazda	7	9	16
mitsubishi	4	9	13
nissan	9	9	18
subaru	9	3	12
toyota	18	14	32
volkswagen	8	4	12
volvo	11	0	11
Total	71	56	127

Label Rows and Columns

```
pd.crosstab(df.make, df.body_style,
            rownames=['Auto Manufacturer'],
            colnames=['Body Style'])
```

Body Style	convertible	hardtop	hatchback	sedan	wagon
Auto Manufacturer					
honda	0	0	7	5	1
mazda	0	0	10	7	0
mitsubishi	0	0	9	4	0
nissan	0	1	5	9	3
subaru	0	0	3	5	4
toyota	1	3	14	10	4
volkswagen	1	0	1	9	1
volvo	0	0	0	8	3

Grouping and Aggregating Values