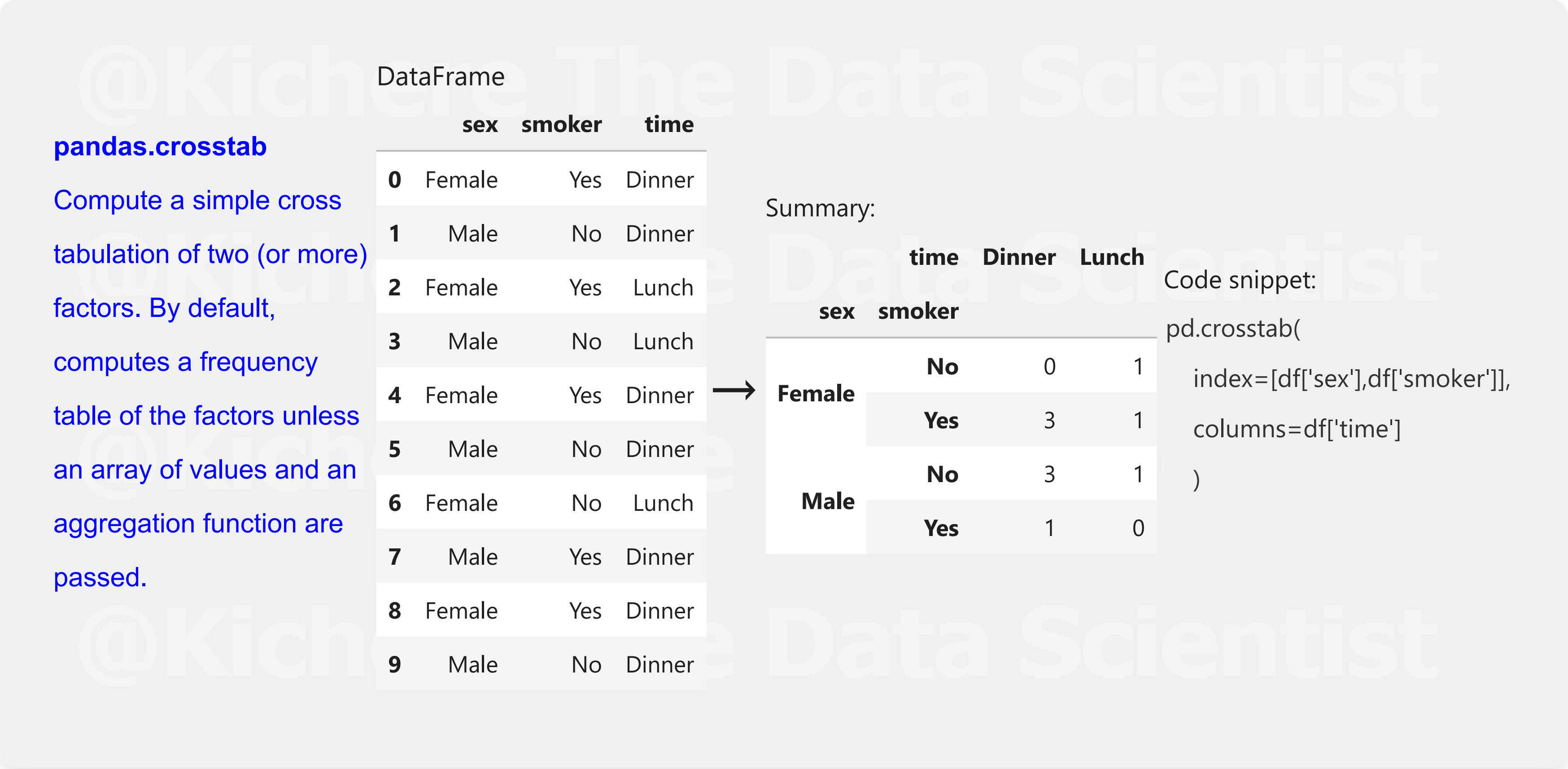
None

`pandas.crosstab Overview`

Count of smokers and non-smokers for each sex



pandas.crosstab

Compute a simple cross tabulation of two (or more) factors. By default, computes a frequency table of the factors unless an array of values and an aggregation function are passed.

DataFrame

	sex	smoker	time
0	Female	Yes	Dinner
1	Male	No	Dinner
2	Female	Yes	Lunch
3	Male	No	Lunch
4	Female	Yes	Dinner
5	Male	No	Dinner
6	Female	No	Lunch
7	Male	Yes	Dinner
8	Female	Yes	Dinner
9	Male	No	Dinner

⟶

Summary:

	time	**Dinner**	**Lunch**
sex	**smoker**		
Female	**No**	0	1
	Yes	3	1
Male	**No**	3	1
	Yes	1	0

Code snippet:

```
pd.crosstab(
    index=[df['sex'],df['smoker']],
    columns=df['time']
    )
```

Preface

****

Contents

****

1st Edition

Why This E-book?

> "The aim of this ebook is to give you the 'aha' moment right away at the start of learning a new concept."

- Practical step By Step Guide With Simple Examples

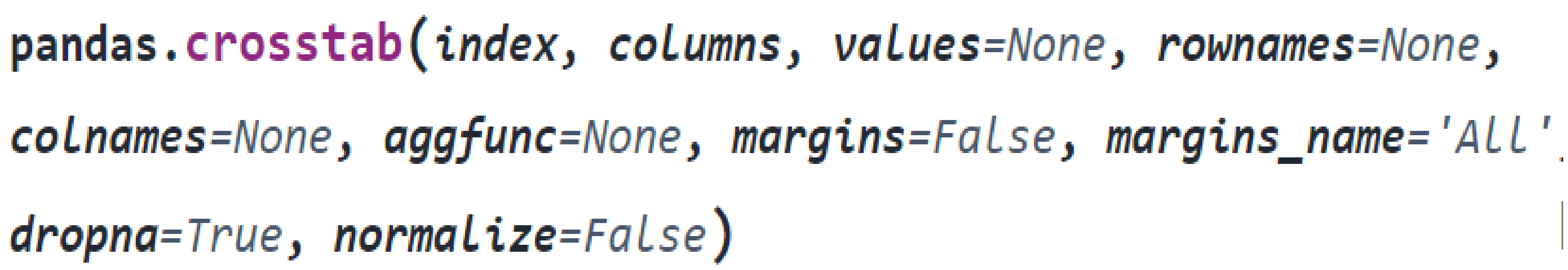
- Visual Illustrations and Interactive
- Simple Datasets
- Comprehensive Coverage(pandas Documentation used as reference)

****

Introduction

pandas.crosstab is a powerful function in the Pandas library used for creating contingency tables, which display the frequency distribution of variables. It allows you to cross-tabulate two or more factors, showing the counts or proportions of each combination of categories. You can easily add margins (totals) and normalize the data to show percentages or fractions, making it a versatile tool for data analysis and summarization.

The "pandas.crosstab" syntax



```
pandas.crosstab(index, columns, values=None, rownames=None,
colnames=None, aggfunc=None, margins=False, margins_name='All',
dropna=True, normalize=False)
```

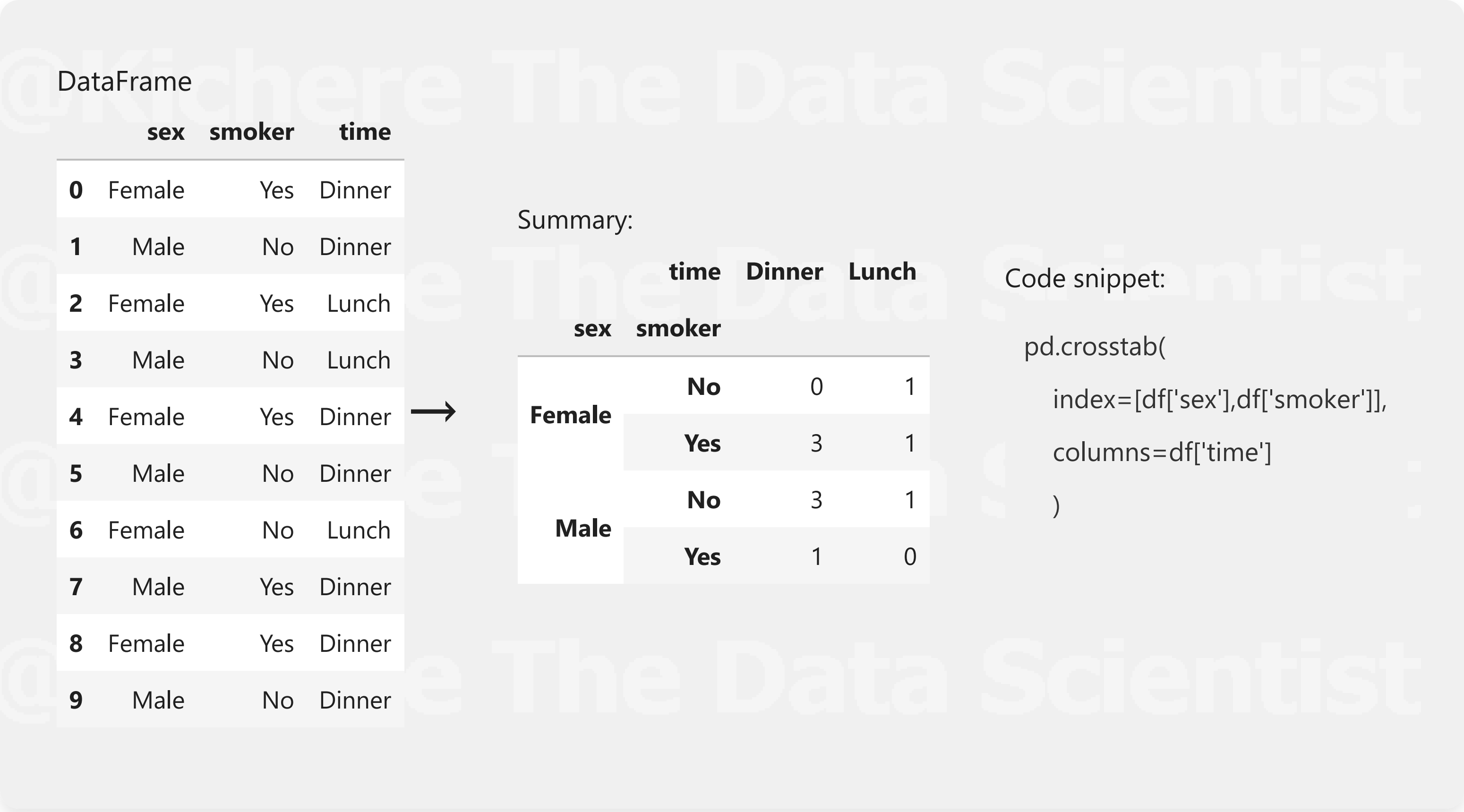
The index parameter

Specificies values to group by in the rows.
This parameter specifies the column(s) whose unique values will be used to form the rows of the resulting cross-tabulation.

The "index" parameter accepts array-like, Series, or list of arrays/Series

```python
import numpy as np
import pandas as pd
```

Count of smokers and non-smokers for each sex



DataFrame

	sex	smoker	time
0	Female	Yes	Dinner
1	Male	No	Dinner
2	Female	Yes	Lunch
3	Male	No	Lunch
4	Female	Yes	Dinner
5	Male	No	Dinner
6	Female	No	Lunch
7	Male	Yes	Dinner
8	Female	Yes	Dinner
9	Male	No	Dinner

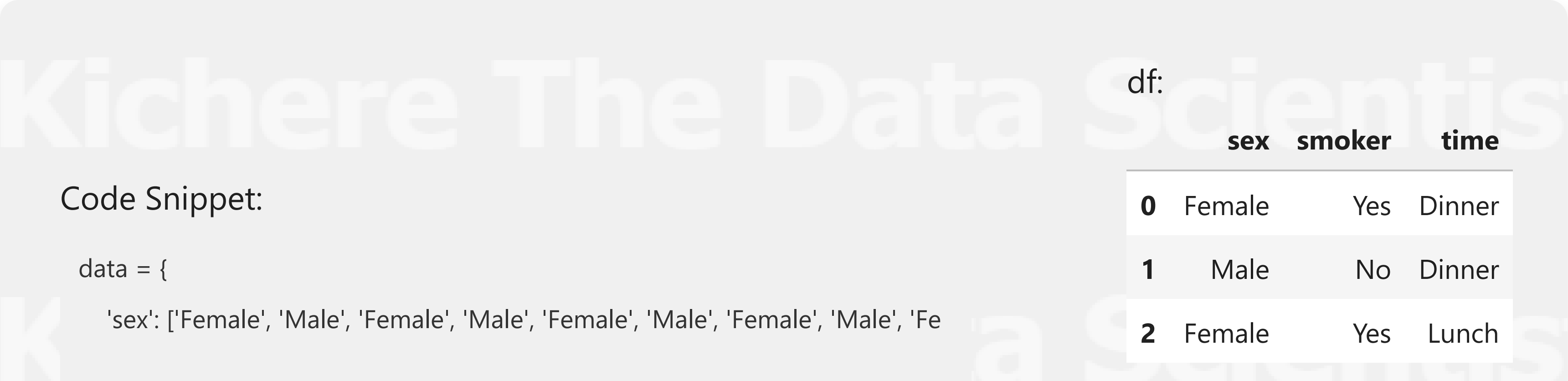
⟶

Summary:

	time	Dinner	Lunch
sex	**smoker**		
Female	**No**	0	1
	Yes	3	1
Male	**No**	3	1
	Yes	1	0

Code snippet:

```
pd.crosstab(
    index=[df['sex'],df['smoker']],
    columns=df['time']
    )
```

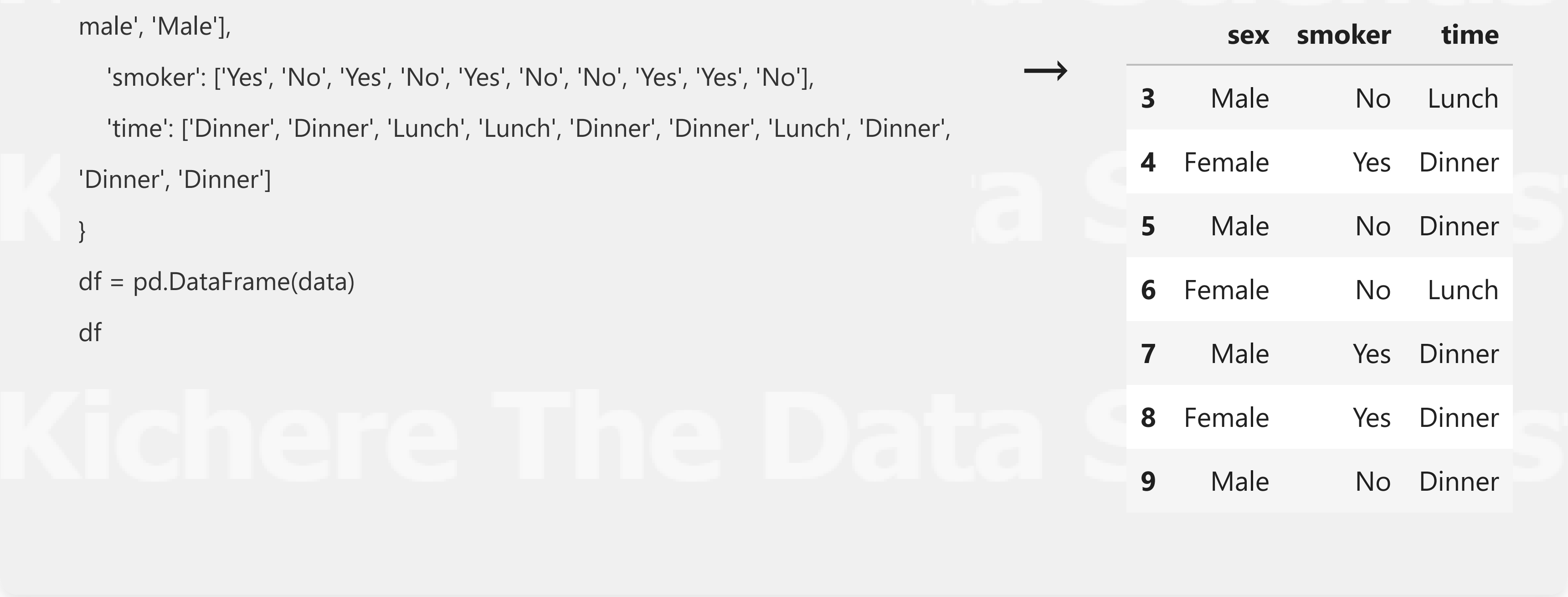


Code Snippet:

```
data = {
    'sex': ['Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Fe
```

df:

	sex	smoker	time
0	Female	Yes	Dinner
1	Male	No	Dinner
2	Female	Yes	Lunch



```
male', 'Male'],
    'smoker': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No'],
    'time': ['Dinner', 'Dinner', 'Lunch', 'Lunch', 'Dinner', 'Dinner', 'Lunch', 'Dinner',
'Dinner', 'Dinner']
}
df = pd.DataFrame(data)
df
```

→

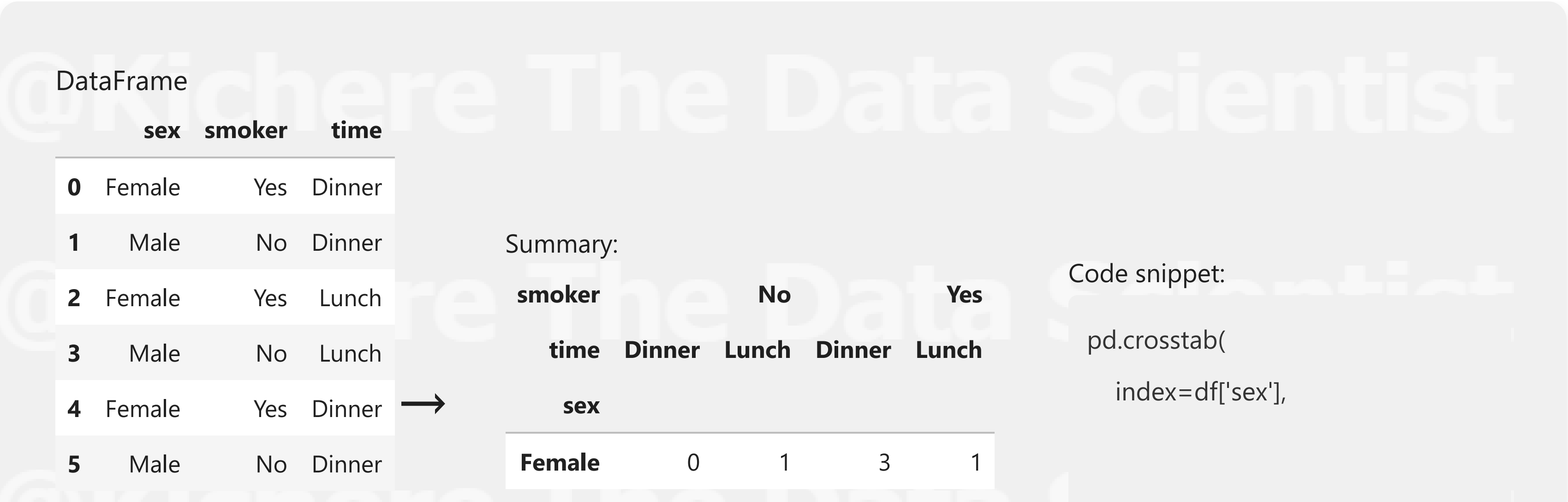
	sex	smoker	time
3	Male	No	Lunch
4	Female	Yes	Dinner
5	Male	No	Dinner
6	Female	No	Lunch
7	Male	Yes	Dinner
8	Female	Yes	Dinner
9	Male	No	Dinner

The columns parameter

Specifies Values to group by in the columns.

The "columns" parameter accepts array-like, Series, or list of arrays/Series

Counts of combinations of smoker and time for each sex category



DataFrame

	sex	smoker	time
0	Female	Yes	Dinner
1	Male	No	Dinner
2	Female	Yes	Lunch
3	Male	No	Lunch
4	Female	Yes	Dinner
5	Male	No	Dinner

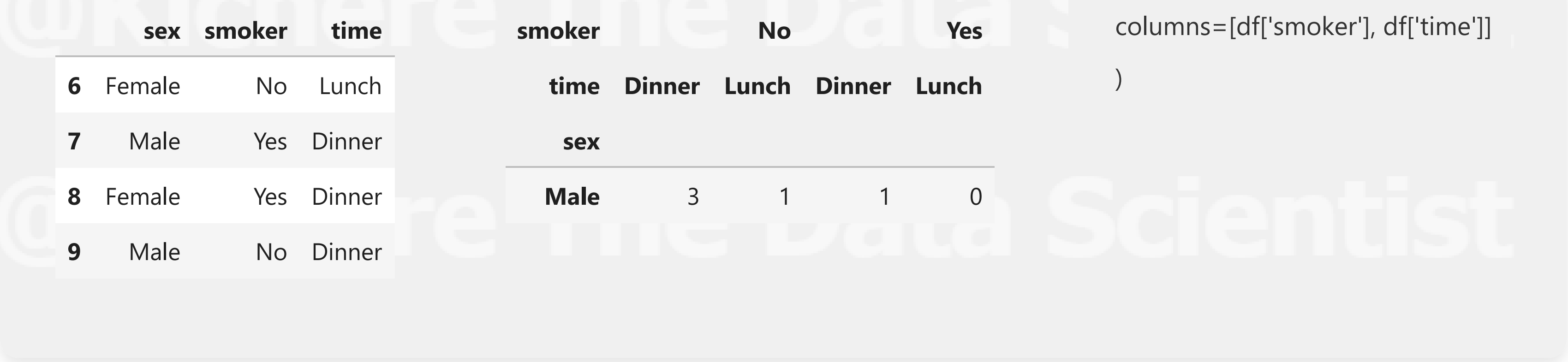
→

Summary:

smoker	No		Yes	
time	**Dinner**	**Lunch**	**Dinner**	**Lunch**
sex				
Female	0	1	3	1

Code snippet:

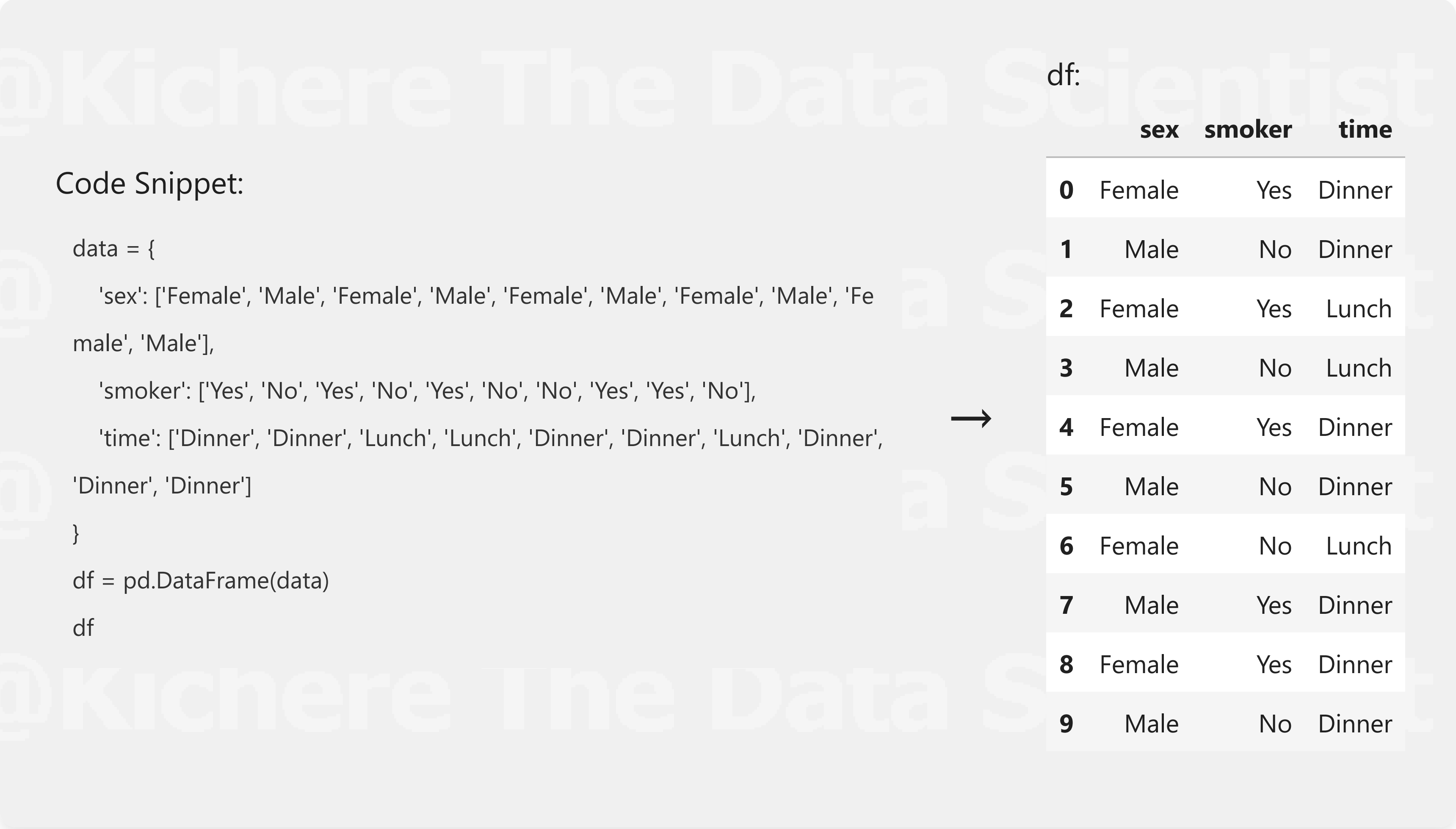
```
pd.crosstab(
    index=df['sex'],
```



	sex	smoker	time
6	Female	No	Lunch
7	Male	Yes	Dinner
8	Female	Yes	Dinner
9	Male	No	Dinner

smoker	No		Yes	
time	Dinner	Lunch	Dinner	Lunch
sex				
Male	3	1	1	0

```
columns=[df['smoker'], df['time']]
)
```



Code Snippet:

```
data = {
    'sex': ['Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Fe
male', 'Male'],
    'smoker': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No'],
    'time': ['Dinner', 'Dinner', 'Lunch', 'Lunch', 'Dinner', 'Dinner', 'Lunch', 'Dinner',
'Dinner', 'Dinner']
}
df = pd.DataFrame(data)
df
```

→

df:

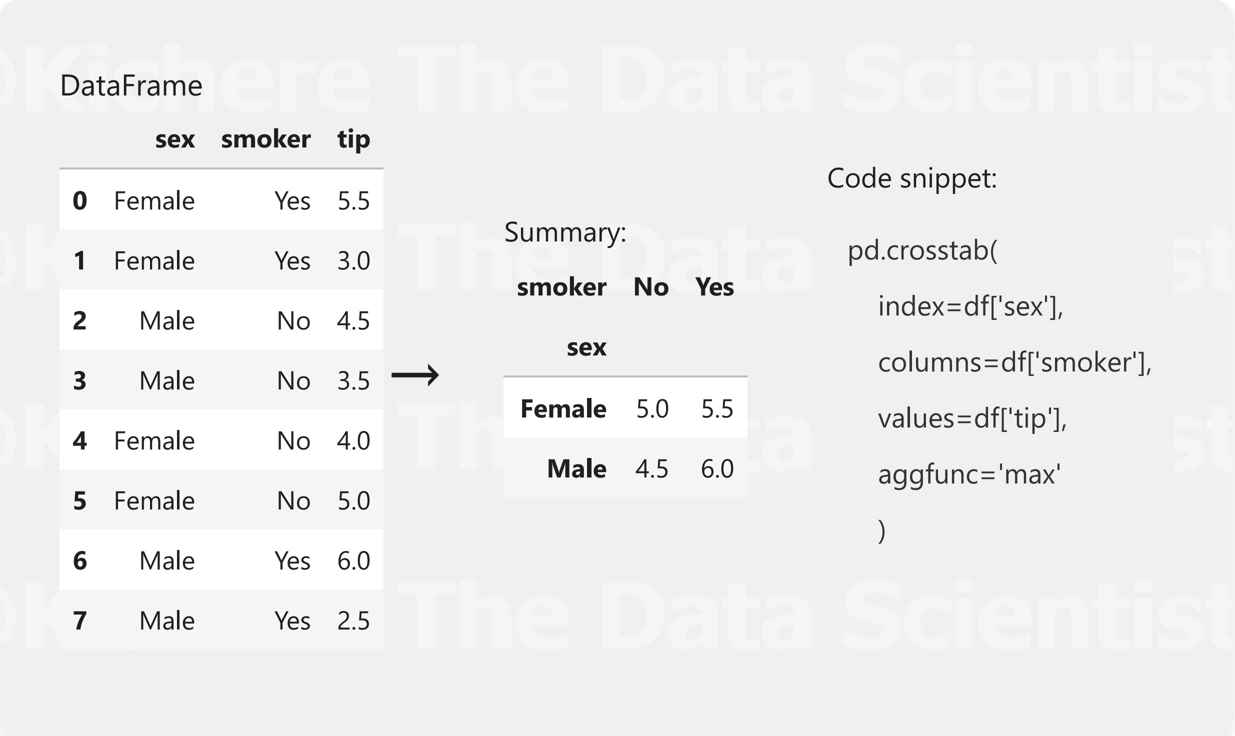
	sex	smoker	time
0	Female	Yes	Dinner
1	Male	No	Dinner
2	Female	Yes	Lunch
3	Male	No	Lunch
4	Female	Yes	Dinner
5	Male	No	Dinner
6	Female	No	Lunch
7	Male	Yes	Dinner
8	Female	Yes	Dinner
9	Male	No	Dinner

The values parameter

Array of values to aggregate according to the factors. Requires aggfunc be specified.

The "values" parameter accepts array-like and is optional.

Maximum tip amount for each combination of 'sex' and 'smoker' status



DataFrame

	sex	smoker	tip
0	Female	Yes	5.5
1	Female	Yes	3.0
2	Male	No	4.5
3	Male	No	3.5
4	Female	No	4.0
5	Female	No	5.0
6	Male	Yes	6.0
7	Male	Yes	2.5

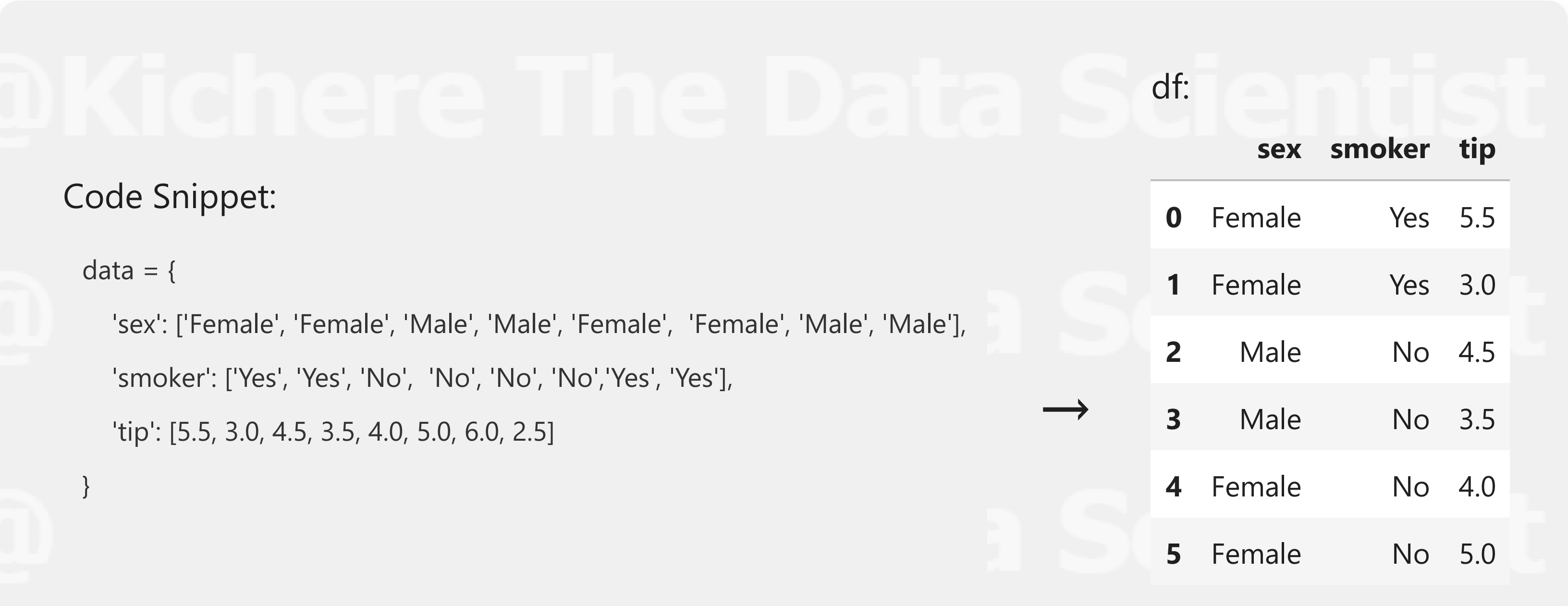
→

Summary:

smoker	No	Yes
sex		
Female	5.0	5.5
Male	4.5	6.0

Code snippet:

```
pd.crosstab(
    index=df['sex'],
    columns=df['smoker'],
    values=df['tip'],
    aggfunc='max'
    )
```



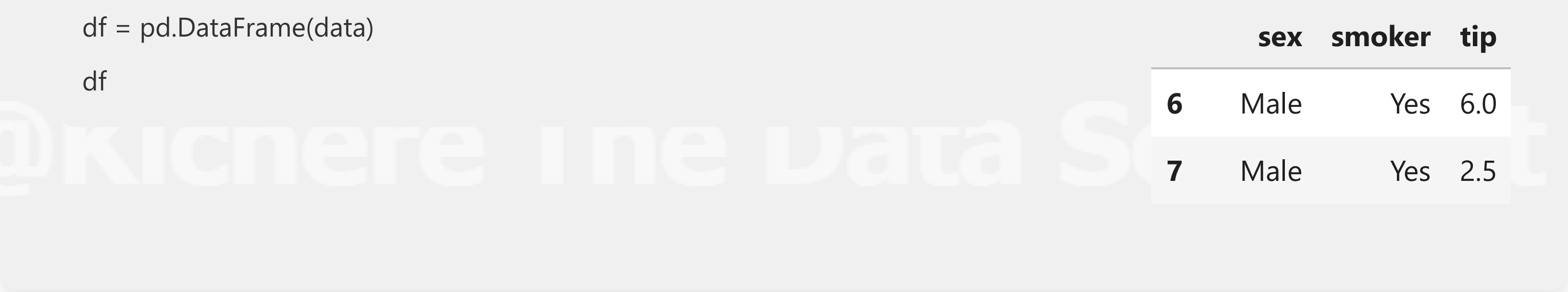
Code Snippet:

```
data = {
    'sex': ['Female', 'Female', 'Male', 'Male', 'Female',  'Female', 'Male', 'Male'],
    'smoker': ['Yes', 'Yes', 'No',  'No', 'No', 'No','Yes', 'Yes'],
    'tip': [5.5, 3.0, 4.5, 3.5, 4.0, 5.0, 6.0, 2.5]
}
```

→

df:

	sex	smoker	tip
0	Female	Yes	5.5
1	Female	Yes	3.0
2	Male	No	4.5
3	Male	No	3.5
4	Female	No	4.0
5	Female	No	5.0



```
df = pd.DataFrame(data)
df
```

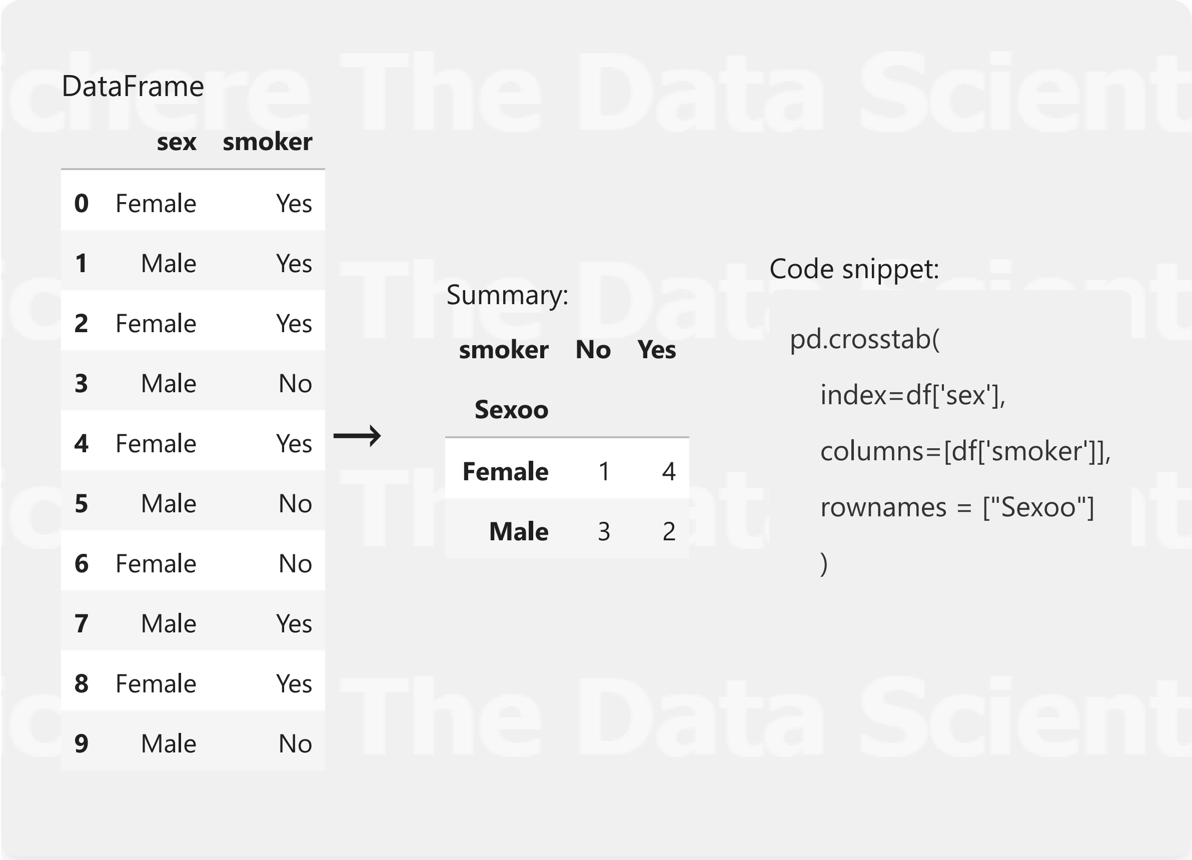
	sex	smoker	tip
6	Male	Yes	6.0
7	Male	Yes	2.5

The `rownames` parameter

"rownames" is used to label the rows in the resulting table.

The "rownames " parameter accepts sequence and it is None by default. If passed, must match number of row arrays passed.

Rename the row index to 'Sexoo'



DataFrame

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

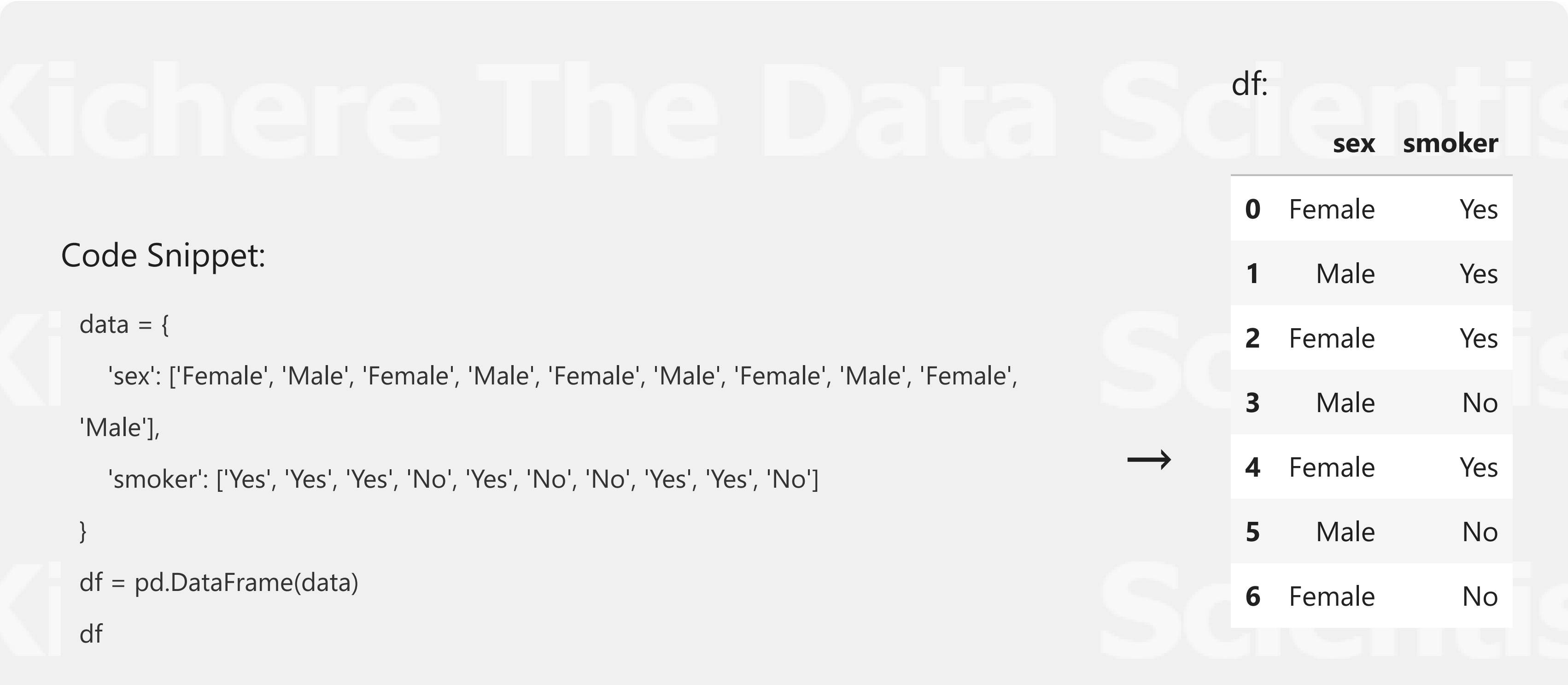
→

Summary:

smoker	**No**	**Yes**
Sexoo		
Female	1	4
Male	3	2

Code snippet:

```
pd.crosstab(
    index=df['sex'],
    columns=[df['smoker']],
    rownames = ["Sexoo"]
    )
```



Code Snippet:

```
data = {
    'sex': ['Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female',
'Male'],
    'smoker': ['Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No']
}
df = pd.DataFrame(data)
df
```

→

df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No

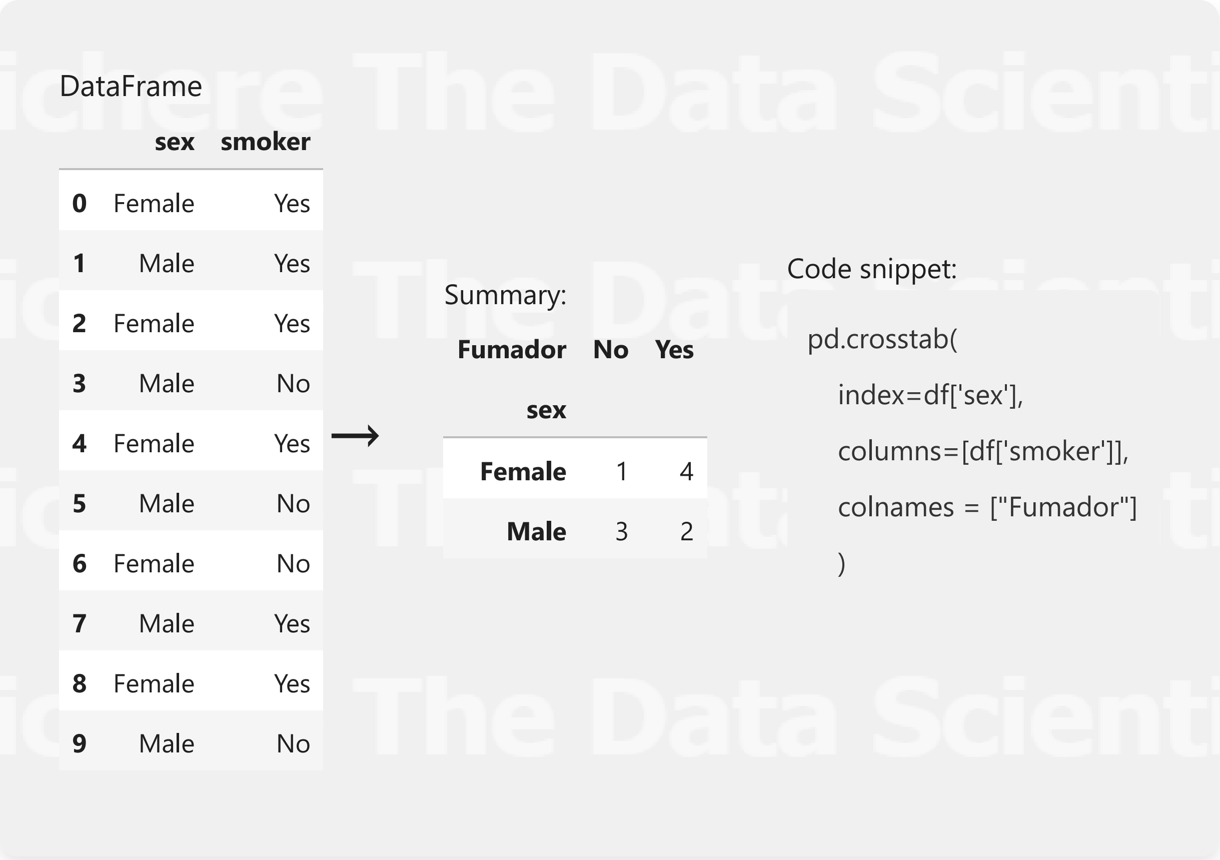
	sex	smoker
7	Male	Yes
8	Female	Yes
9	Male	No

The colnames parameter

"colnames" is used to label the cols in the resulting table.

The "colnames " parameter accepts sequence and it is None by default.
If passed, must match number of col arrays passed.

Rename the column index(smoker) to 'Fumador'



DataFrame

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

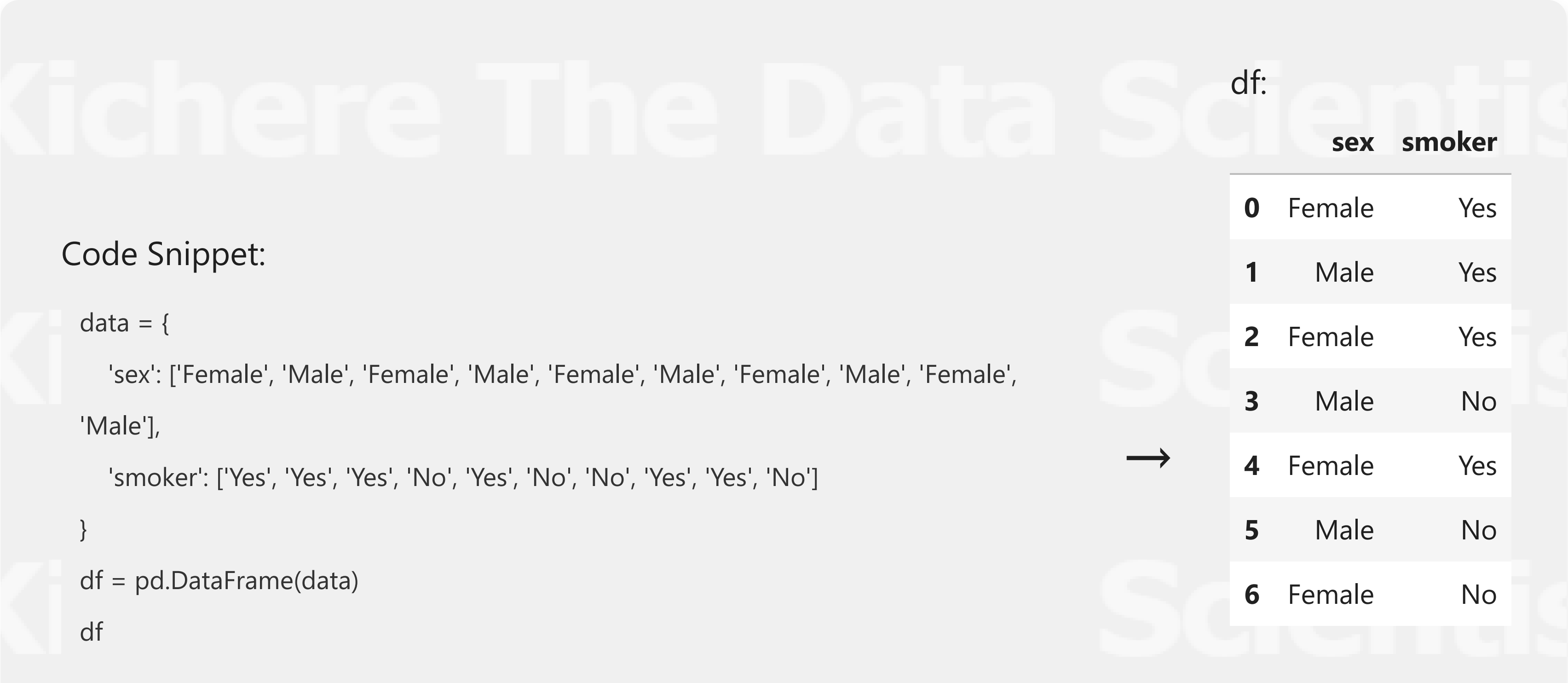
⟶

Summary:

Fumador	No	Yes
sex		
Female	1	4
Male	3	2

Code snippet:

```
pd.crosstab(
    index=df['sex'],
    columns=[df['smoker']],
    colnames = ["Fumador"]
    )
```



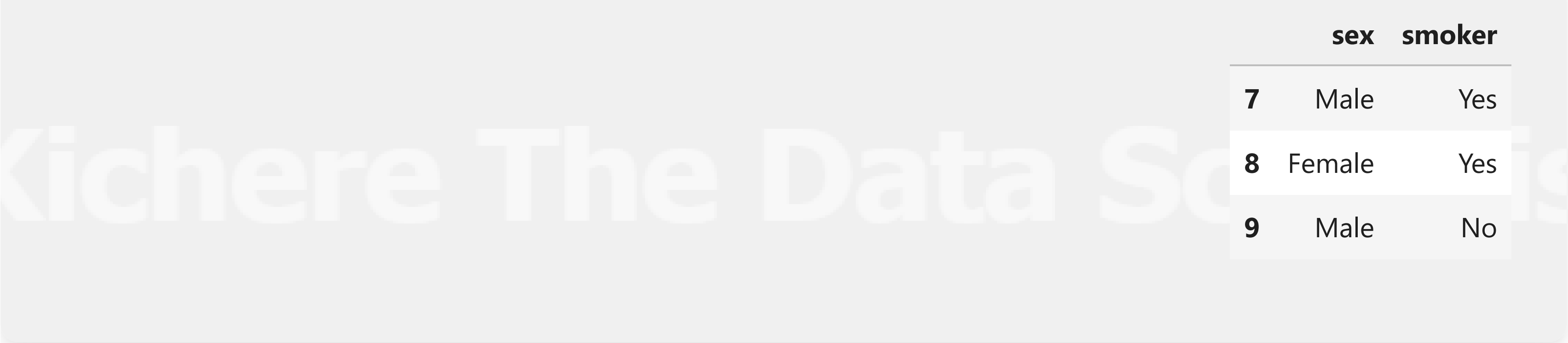
Code Snippet:

```
data = {
    'sex': ['Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female',
'Male'],
    'smoker': ['Yes', 'Yes', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No']
}
df = pd.DataFrame(data)
df
```

⟶

df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No



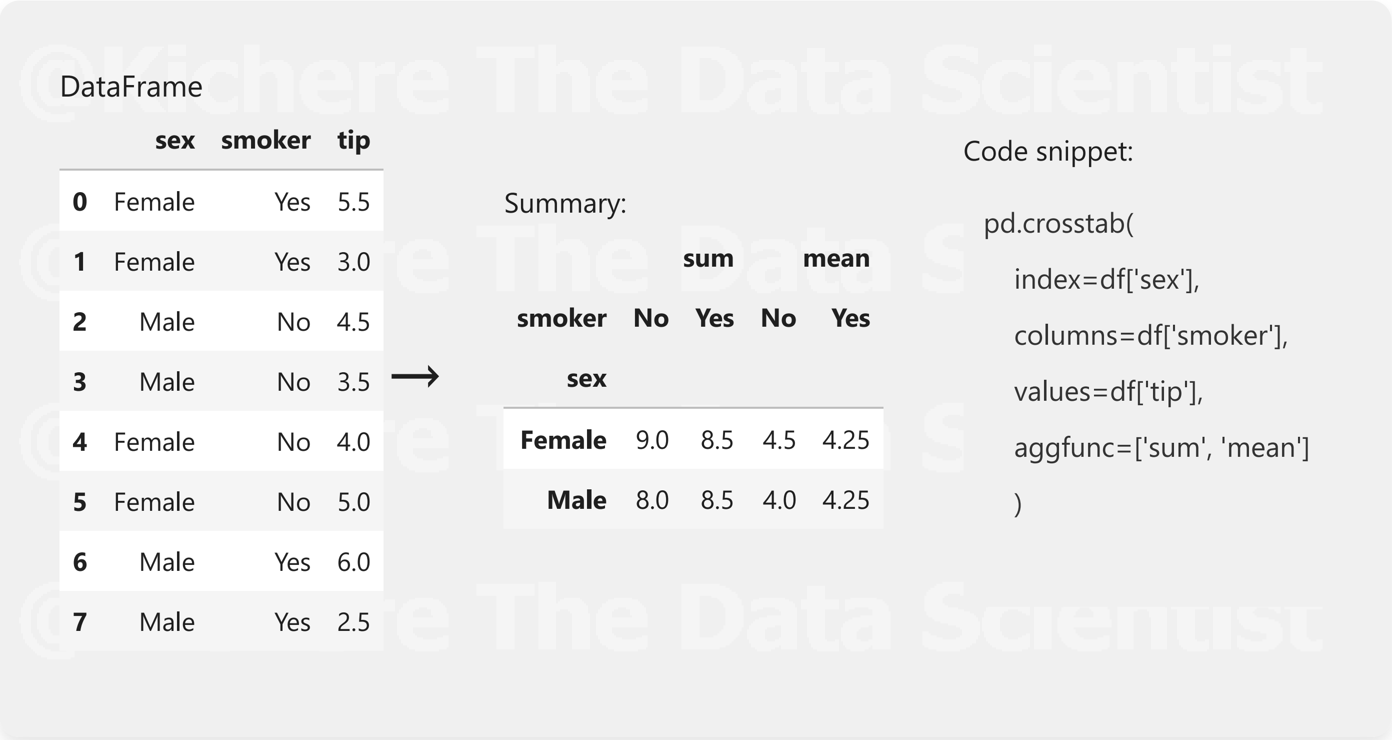
	sex	smoker
7	Male	Yes
8	Female	Yes
9	Male	No

The aggfunc parameter

Accepts a function and it is optional..

If specified, requires values be specified as well.

Sum and the mean of tips for each combination of 'sex' and 'smoker'



DataFrame

	sex	smoker	tip
0	Female	Yes	5.5
1	Female	Yes	3.0
2	Male	No	4.5
3	Male	No	3.5
4	Female	No	4.0
5	Female	No	5.0
6	Male	Yes	6.0
7	Male	Yes	2.5

→

Summary:

	sum		**mean**	
smoker	**No**	**Yes**	**No**	**Yes**
sex				
Female	9.0	8.5	4.5	4.25
Male	8.0	8.5	4.0	4.25

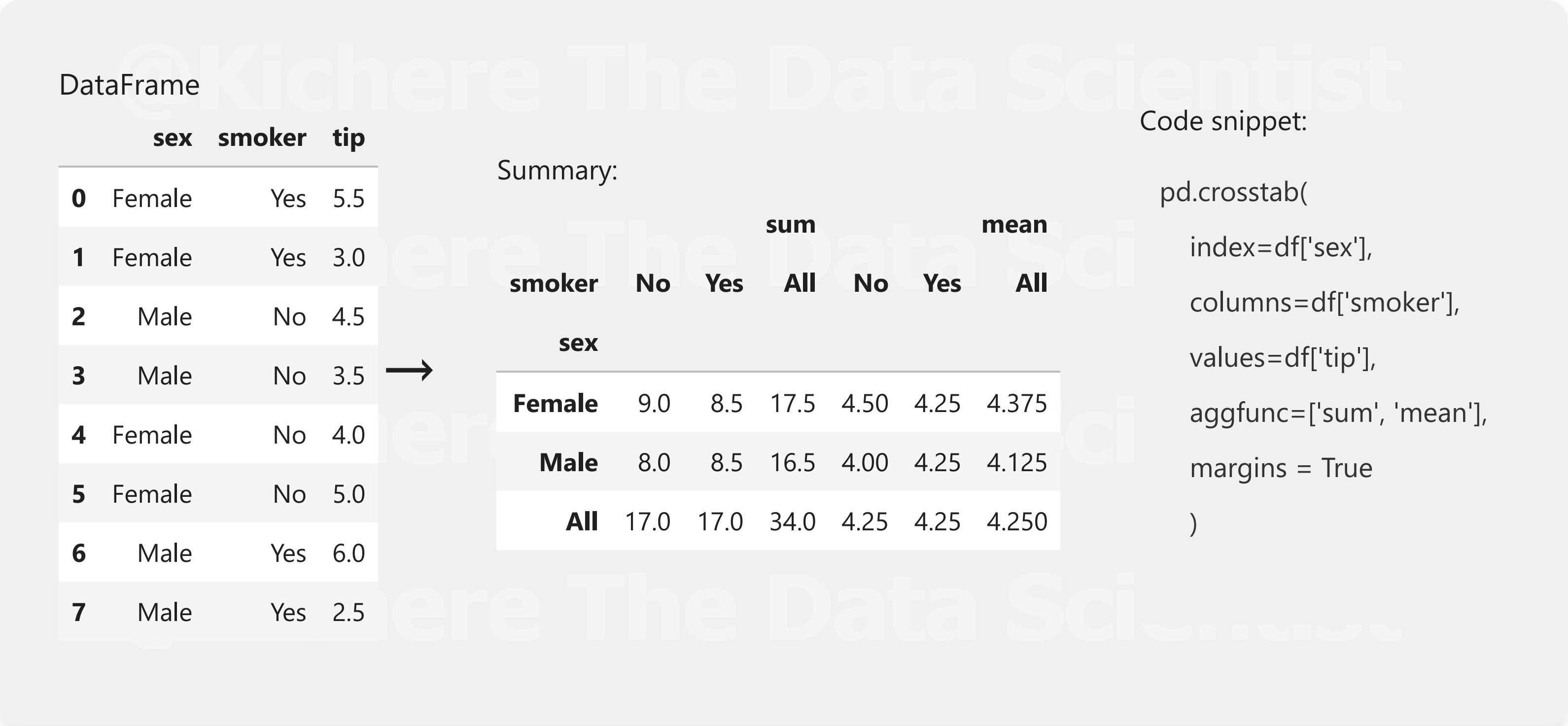
Code snippet:

```
pd.crosstab(
    index=df['sex'],
    columns=df['smoker'],
    values=df['tip'],
    aggfunc=['sum', 'mean']
    )
```

The margins parameter

The "margins" parameter Adds row/column margins (subtotals).
It is boolean and defaults to False.

Sum and the mean of tips for each combination of 'sex' and 'smoker', and also include row and column totals



DataFrame

	sex	smoker	tip
0	Female	Yes	5.5
1	Female	Yes	3.0
2	Male	No	4.5
3	Male	No	3.5
4	Female	No	4.0
5	Female	No	5.0
6	Male	Yes	6.0
7	Male	Yes	2.5

→

Summary:

	sum			mean		
smoker	**No**	**Yes**	**All**	**No**	**Yes**	**All**
sex						
Female	9.0	8.5	17.5	4.50	4.25	4.375
Male	8.0	8.5	16.5	4.00	4.25	4.125
All	17.0	17.0	34.0	4.25	4.25	4.250

Code snippet:

```
pd.crosstab(
    index=df['sex'],
    columns=df['smoker'],
    values=df['tip'],
    aggfunc=['sum', 'mean'],
    margins = True
    )
```

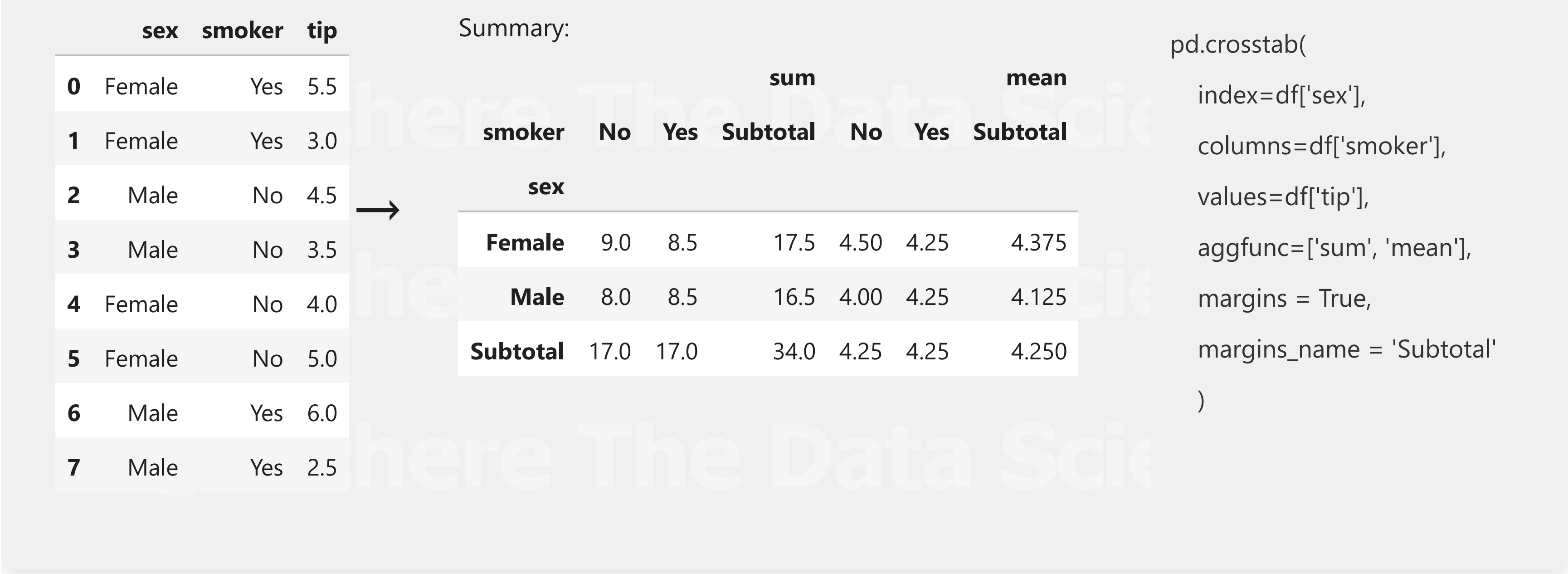
The margins_name parameter

The "margins_name" parameter specifies the name of the row or column that will contain the totals when margins=True."
It is string and By default, the name is set to 'All'.

Sum and the mean of tips for each combination of 'sex' and 'smoker', and also include row and column totals

DataFrame

Code snippet:



	sex	smoker	tip
0	Female	Yes	5.5
1	Female	Yes	3.0
2	Male	No	4.5
3	Male	No	3.5
4	Female	No	4.0
5	Female	No	5.0
6	Male	Yes	6.0
7	Male	Yes	2.5

→

Summary:

	sum			mean		
smoker	**No**	**Yes**	**Subtotal**	**No**	**Yes**	**Subtotal**
sex						
Female	9.0	8.5	17.5	4.50	4.25	4.375
Male	8.0	8.5	16.5	4.00	4.25	4.125
Subtotal	17.0	17.0	34.0	4.25	4.25	4.250

```
pd.crosstab(
    index=df['sex'],
    columns=df['smoker'],
    values=df['tip'],
    aggfunc=['sum', 'mean'],
    margins = True,
    margins_name = 'Subtotal'
    )
```

The dropna parameter

The "dropna" parameter does not include columns whose entries are all NaN. If True, rows with a NaN value in any column will be omitted before computing margins.
It is a boolean and defaults to True.

Show the total sales (sum of 'Price') for each manager across different products.

df:

	Manager	Product	Quantity	Price	Status
0	Debra	CPU	2.0	600.0	None
1	Debra	RAM	NaN	100.0	None
2	Fred	CPU	1.0	300.0	None
3	Fred	RAM	3.0	NaN	None

→

When dropna=False:

Product	**CPU**	**RAM**	**nan**	**All**
Manager				
Debra	600.0	100.0	NaN	700.0
Fred	300.0	0.0	NaN	300.0
NaN	NaN	NaN	0.0	NaN
All	900.0	100.0	NaN	1000.0

```
pd.crosstab(
    index=df['Manager'],
    columns=df['Product'],
```

When dropna=True (default):

Product	**CPU**	**RAM**	**All**
Manager			
Debra	600.0	100.0	700.0
Fred	300.0	0.0	300.0
All	900.0	100.0	1000.0

```
pd.crosstab(
    index=df['Manager'],
    columns=df['Product'],
```

	Manager	Product	Quantity	Price	Status
4	None	None	NaN	NaN	None

```
values=df['Price'],
aggfunc='sum',
margins=True,
dropna = False
)
```

```
values=df['Price'],
aggfunc='sum',
margins=True,
dropna =True
)
```

The normalize parameter

The "normalize" parameter is boolean.
It accepts {'all', 'index', 'columns'}, or {0,1} and defaults to False.
Its purpose is to normalize(Proportion) by dividing all values by the sum of values.

Code Snippet:

```
data = {
    'sex': ['Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Fe
male', 'Male'],
    'smoker': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'No', 'Yes', 'Yes', 'No'],
    'time': ['Dinner', 'Dinner', 'Lunch', 'Lunch', 'Dinner', 'Dinner', 'Lunch', 'Dinner',
'Dinner', 'Dinner']
}
df = pd.DataFrame(data)
df
```

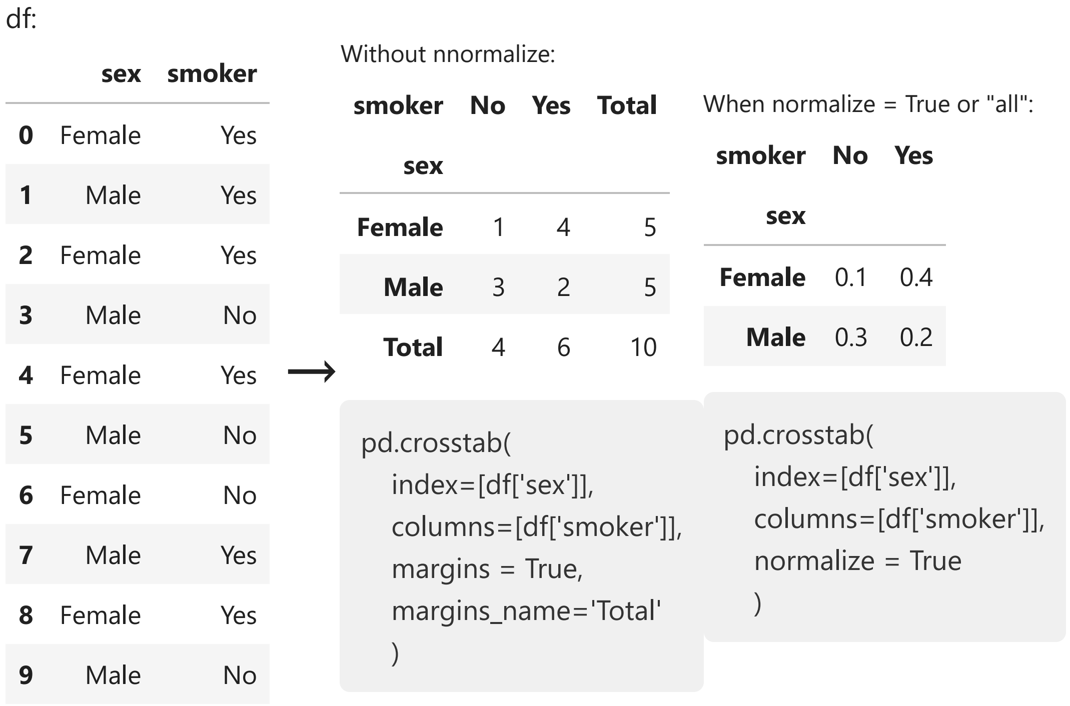
→

df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

If passed ‘all’ or True, will normalize over all values.

Generate a table of sex and smoker status in a DataFrame, showing proportions relative to the entire dataset



df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

⟶

Without nnormalize:

smoker	No	Yes	Total
sex			
Female	1	4	5
Male	3	2	5
Total	4	6	10

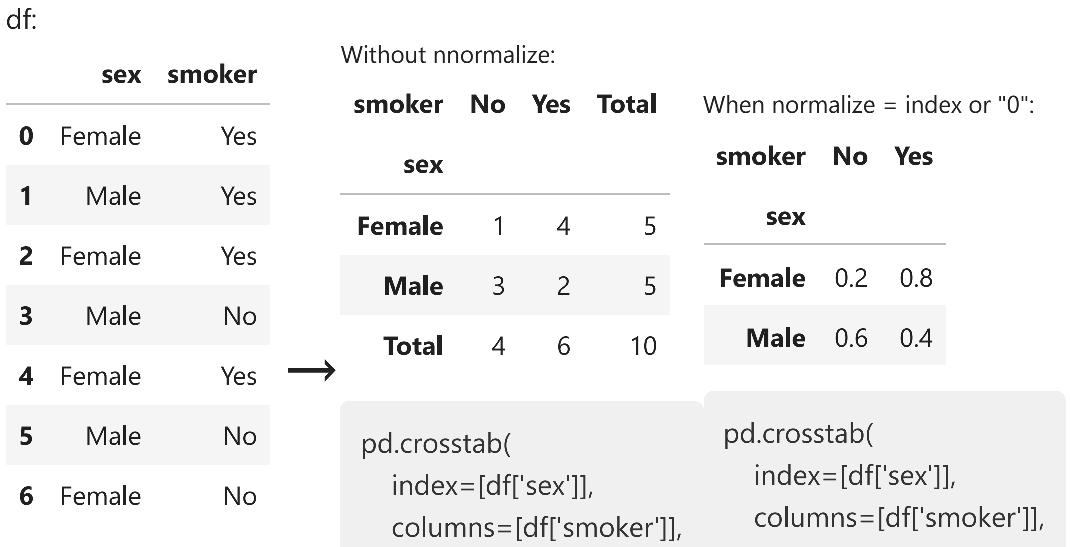
```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    margins = True,
    margins_name='Total'
    )
```

When normalize = True or "all":

smoker	No	Yes
sex		
Female	0.1	0.4
Male	0.3	0.2

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    normalize = True
    )
```

If passed ‘index’ will normalize over each row.



df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No

⟶

Without nnormalize:

smoker	No	Yes	Total
sex			
Female	1	4	5
Male	3	2	5
Total	4	6	10

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
```

When normalize = index or "0":

smoker	No	Yes
sex		
Female	0.2	0.8
Male	0.6	0.4

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
```

	sex	smoker
7	Male	Yes
8	Female	Yes
9	Male	No

```
    margins = True,
    margins_name='Total'
    )
```

```
    normalize = 'index'
    )
```

If passed ‘columns’ will normalize over each column.

df:

	sex	smoker
0	Female	Yes
1	Male	Yes
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

⟶

Without normalize:

smoker	**No**	**Yes**	**Total**
sex			
Female	1	4	5
Male	3	2	5
Total	4	6	10

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    margins = True,
    margins_name='Total'
    )
```

When normalize = 'index' or 1:

smoker	**No**	**Yes**
sex		
Female	0.25	0.666667
Male	0.75	0.333333

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    normalize = 'index'
    )
```

If margins is True, will also normalize margin values.

df:

	sex	smoker
0	Female	Yes
1	Male	Yes

Without nnormalize:

smoker	**No**	**Yes**	**Total**
sex			
Female	1	4	5

When normalize = True or "all":

smoker	**No**	**Yes**	**Total**
sex			
Female	0.1	0.4	0.5

	sex	smoker
2	Female	Yes
3	Male	No
4	Female	Yes
5	Male	No
6	Female	No
7	Male	Yes
8	Female	Yes
9	Male	No

⟶

smoker	**No**	**Yes**	**Total**
sex			
Male	3	2	5
Total	4	6	10

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    margins = True,
    margins_name='Total'
    )
```

smoker	**No**	**Yes**	**Total**
sex			
Male	0.3	0.2	0.5
Total	0.4	0.6	1.0

```
pd.crosstab(
    index=[df['sex']],
    columns=[df['smoker']],
    normalize = True,
    margins = True,
    margins_name='Total'
    )
```

Project(Real Life application)

pandas.crosstab (https://interactivechaos.com/en/python/function/pandascrosstab)

Sources & References

pandas.crosstab Documentation (https://pandas.pydata.org/docs/reference/api/pandas.crosstab.html#pandas.crosstab)

pandas.crosstab (https://interactivechaos.com/en/python/function/pandascrosstab)

Contacts and Social Media

Kichere Magubu

Dar es salaam, Tanzania

Kichere Magubu

▶ Kichere The Data Scientist

KichereTheDataScientist

k KichereTheDataScientist

✉ kicherethedatascientist@gmail.com

kicherethedatascientist

☎ +255 654 729 851

****