

PANDAS RENAME

SHARP SIGHT

WHAT YOU'LL LEARN

- How to use the Pandas `rename()` method
- How to rename rows and index values of a DataFrame

PANDAS RENAME OVERVIEW

THE PANDAS RENAME METHOD RENAMES COLUMNS AND INDEX VALUES

```
supercars.rename(columns = {'horsepower': 'hp'})
```

Old name				Output:				New name			
model	make	year	horsepower*		model	make	year	hp*			
911 Turbo	Porsche	2013	513		911 Turbo	Porsche	2013	513			
Z4 3.0i	BMW	2007	215		Z4 3.0i	BMW	2007	215			
Cayman	Porsche	2007	241		Cayman	Porsche	2007	241			
Veyron	Bugatti	2012	1184		Veyron	Bugatti	2012	1184			
550i	BMW	2013	444		550i	BMW	2013	444			

Note: Here, supercars is an "abbreviated" version of the actual data

SYNTAX: RENAME COLUMNS

SYNTAX: RENAME COLUMNS

The name of the DataFrame you want to operate on

The name of the method (called with "dot" syntax)

```
myDataFrame.rename(columns = {'old_var': 'new_var'})
```

The column parameter

An expression that defines the old variable that you want to rename, and the new name you want to give it

SYNTAX: RENAME COLUMNS

```
myDataFrame.rename(columns = {'old_var': 'new_var'})
```



This should be expressed as a dictionary, with the old variable name on the left and the new name on the right

Notice also that the variable names are in quotes ...

RENAME INDEX VALUES


SYNTAX: RENAME INDEX VALUES

The syntax to rename index values is very similar to the syntax to rename columns

```
myDataFrame.rename(index = {'old_val': 'new_val'})
```



The index
parameter



An expression that specifies the old
index value that you want to rename,
and the new name you want to give it

APPLY A FUNCTION TO ALL
COLUMNS OR INDEX VALUES

IT'S ALSO POSSIBLE TO APPLY A FUNCTION TO ALL OF THE COLUMNS / INDEX VALUES

- Supply a function name to the `columns` or `index` parameter
- I'll show you the code to change `columns`
 - There's corresponding syntax to change the index values

SYNTAX: RENAME COLUMNS

The columns
parameter



```
myDataFrame.rename(columns = function)
```



The name of a function that you want
to apply to every column

E.g., you can apply `str.lower` to
change every variable name to lower
case

PARAMETERS OF PANDAS RENAME

THE PARAMETERS OF PANDAS RENAME

Parameter	What it does
<code>columns</code>	<p>Specifies which columns to rename</p> <p>You can also specify the name of a function to apply to every column</p>
<code>index</code>	<p>Specifies which index values to rename</p> <p>You can also specify the name of a function to apply to every index value</p>
<code>inplace</code>	<p>Indicates whether you want to modify the DataFrame directly (inplace) or not ... set to <code>inplace = False</code> by default</p>

THE OUTPUT OF PANDAS RENAME

- By default, `rename()` creates a *new* DataFrame with renamed variables/indexes
 - By default, `inplace` is set to `inplace = False`
 - This leaves the original DataFrame unchanged
- If you set `inplace = True`, `rename()` will directly modify the original DataFrame

RECAP

RECAP OF WHAT WE LEARNED

- You can change the name of columns & index values with `rename()`
- You can also apply a function to all of the columns / index values
- **Next Steps:** Watch the code walkthrough video for step-by-step examples of `rename()`