

[Open in app](#)

Published in CodeX



Sharone Li

Feb 23 · 3 min read ★ · Listen



How to Batch Rename Columns in Pandas Based on Patterns

Share a Piece of Simple Code That Does the Trick and Hope You Will Find it Helpful!

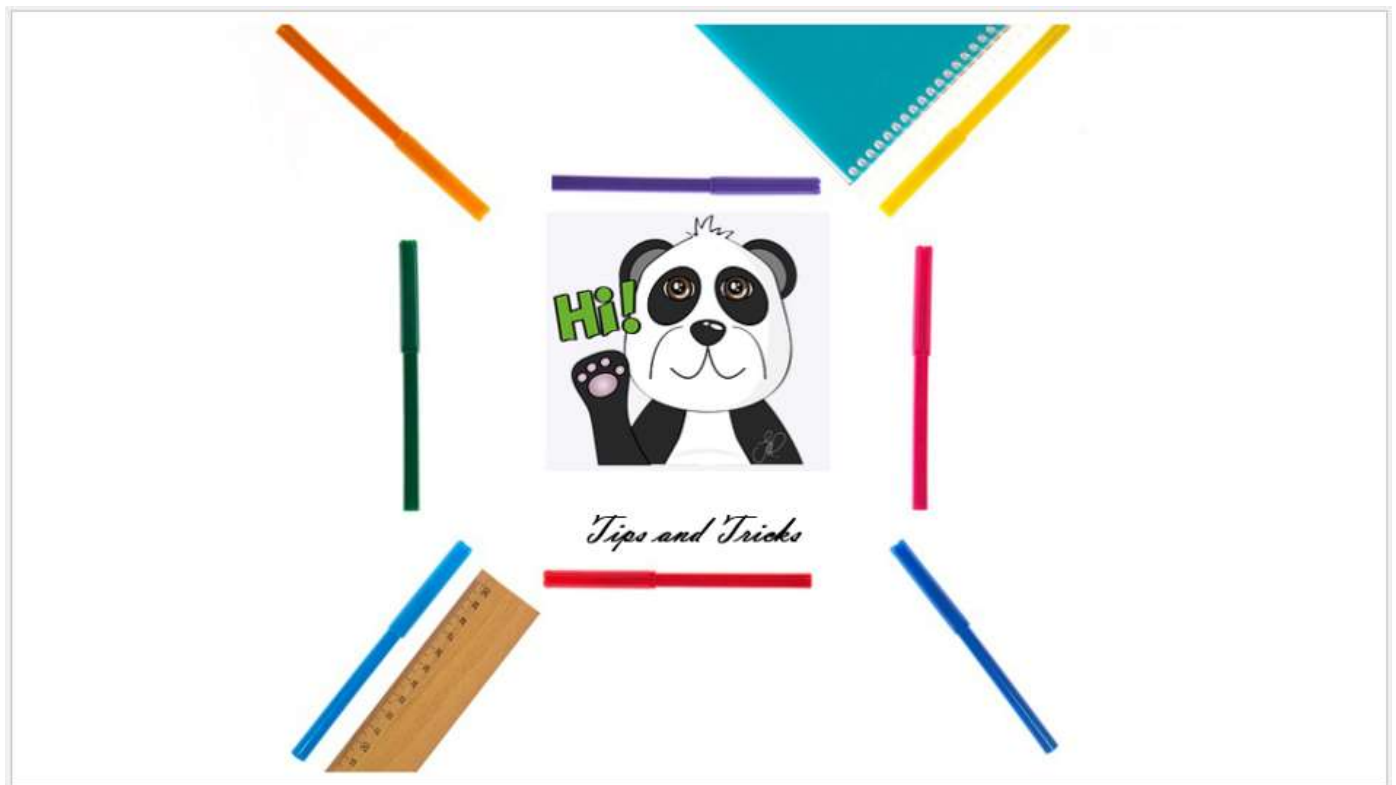


Image by [Pixabay](#) (Modified by Author)

If you have been following my Medium blog for some time, you may notice that I usually like to share complete data science projects I have done in my spare time, such as data visualization, streamlit apps, machine learning, etc. While doing those projects, there are




[Open in app](#)

In our daily data science work, we often need to massage and clean the data before doing any analysis on it. Renaming a column or multiple columns in a Pandas dataframe is a very common task during that process and is quite straightforward to do using Pandas' `rename()` method. Below is a simple example to rename two columns, 'A' and 'B', to 'a' and 'b' respectively.

```
data_frame.rename(columns={"A": "a", "B": "c"})
```

Now, take a look at the following example (you can download the data from the [World Bank's website](#)):

```
df.head()
```

	Series Name	Series Code	Country Name	Country Code	1960 [YR1960]	1961 [YR1961]	1962 [YR1962]	1963 [YR1963]	
26	GDP (current US\$)	NY.GDP.MKTP.CD	Brazil	BRA	..	17275940449.3837	19231747851.5332	23287712878.2002	20
35	GDP (current US\$)	NY.GDP.MKTP.CD	Canada	CAN	40461721692.6468	40934952063.9468	42227447631.9159	45029988561.2124	4
41	GDP (current US\$)	NY.GDP.MKTP.CD	China	CHN	59716467625.3148	50056868957.6732	47209359005.6057	50706799902.5104	59
68	GDP (current US\$)	NY.GDP.MKTP.CD	France	FRA	62225478000.8822	67461644222.0352	75607529809.9288	84759195105.8693	94
73	GDP (current US\$)	NY.GDP.MKTP.CD	Germany	DEU

5 rows × 65 columns

Image by Author

The dataframe has a total of 65 columns, 61 of which are time-related and exhibit a clear pattern in their column names (1960[YR1960] — 2020[YR2020]). We want to rename



[Open in app](#)

To batch rename the 61 columns in the dataframe, we need to first get the list of the old column names using `list(df.iloc[:, 4:].columns)`. Then we use the `for` loop to rename each column name in the list by only taking a substring `col[6:12]` of the original column name.

```
new_columns=[col[6:12] for col in list(df.iloc[:, 4:].columns)]
```

new_columns

```
['YR1960',  
'YR1961',  
'YR1962',  
'YR1963',  
'YR1964',  
'YR1965',  
'YR1966',  
'YR1967',  
'YR1968',  
'YR1969',  
'YR1970',  
'YR1971',  
'YR1972',  
...]
```

Image by Author

You can also do this:

```
new_columns=[col[6:12].replace('YR','YEAR_') for col in  
list(df.iloc[:, 4:].columns)]
```



[Open in app](#)

new_columns

```
['YEAR_1960',  
 'YEAR_1961',  
 'YEAR_1962',  
 'YEAR_1963',  
 'YEAR_1964',  
 'YEAR_1965',  
 'YEAR_1966',  
 'YEAR_1967',  
 'YEAR_1968',  
 'YEAR_1969',  
 'YEAR_1970',
```

Image by Author

```
new_columns=[col[6:12].replace('YR','YEAR_').lower() for col in  
list(df.iloc[:, 4:].columns)]
```

new_columns

```
['year_1960',  
 'year_1961',  
 'year_1962',  
 'year_1963',  
 'year_1964',  
 'year_1965',  
 'year_1966',  
 'year_1967',  
 'year_1968',  
 'year_1969',  
 'year_1970',
```

We can then assign these two column names to the dataframe:




[Open in app](#)

```
df_new.head()
```

	year_1960	year_1961	year_1962	year_1963	year_1964	year_1965	year_1966
26	..	17275940449.3837	19231747851.5332	23287712878.2002	20963733694.9749	22465522884.0988	2828332373
35	40461721692.6468	40934952063.9468	42227447631.9159	45029988561.2124	49377522896.703	54515179580.7148	6108838403
41	59716467625.3148	50056868957.6732	47209359005.6057	50706799902.5104	59708343488.5043	70436266146.7219	7672028596
68	62225478000.8822	67461644222.0352	75607529809.9288	84759195105.8693	94007851047.3678	101537248148.427	1100458521
73

Image by Author

Combine the renamed columns with the other unchanged columns:

```
df2=df.iloc[:, 0:4].merge(df_new, left_index=True, right_index=True)
df2.head()
```

	Series Name	Series Code	Country Name	Country Code	year_1960	year_1961	year_1962	year_1963	year_1964	year_1965	year_1966
26	GDP (current US\$)	NY.GDP.MKTP.CD	Brazil	BRA	..	17275940449.3837	19231747851.5332	23287712878.2002	20963733694.9749	22465522884.0988	2828332373
35	GDP (current US\$)	NY.GDP.MKTP.CD	Canada	CAN	40461721692.6468	40934952063.9468	42227447631.9159	45029988561.2124	49377522896.703	54515179580.7148	6108838403
41	GDP (current US\$)	NY.GDP.MKTP.CD	China	CHN	59716467625.3148	50056868957.6732	47209359005.6057	50706799902.5104	59708343488.5043	70436266146.7219	7672028596
68	GDP (current US\$)	NY.GDP.MKTP.CD	France	FRA	62225478000.8822	67461644222.0352	75607529809.9288	84759195105.8693	94007851047.3678	101537248148.427	1100458521
73	GDP (current US\$)	NY.GDP.MKTP.CD	Germany	DEU

Image by Author

There you go! We have batch-changed column names based on certain patterns observed in the original column names. Thanks for reading this super short tutorial. :) I hope you



[Open in app](#)

The dataset used in demoing the app is an open dataset (no license required) downloaded from The World Bank's website:

https://data.worldbank.org/indicator/NY.GDP.MKTP.CD?name_desc=true

You can unlock full access to my writing and the rest of Medium by signing up for Medium membership (\$5 per month) through this [referral link](#). By signing up through this link, I will receive a portion of your membership fee at no additional cost to you. Thank you!

Join Medium with my referral link - Sharone Li

As a Medium member, a portion of your membership fee goes to writers you read, and you get full access to every story...

medium.com

Sign up for CrunchX

By CodeX

A weekly newsletter on what's going on around the tech and programming space [Take a look.](#)

Get this newsletter

Emails will be sent to insightsbees@gmail.com.
[Not you?](#)





Open in app

