

[HOME](#) [PYTHON](#) [R](#) [LINUX](#) [CONFERENCES](#) [ABOUT](#) [RESOURCES](#)

Python, R, and Linux Tips

To Learn Data Science

You are here: [Home](#) / [Python](#) / [Pandas DataFrame](#) / [change column names in Pandas](#) / How To Change Column Names and Row Indexes in Pandas?

How To Change Column Names and Row Indexes in Pandas?

March 20, 2018 by [cmdline](#)

One of the most common operations one might do while cleaning the data or doing exploratory data analysis in doing data science is manipulating/fixing the column names or row names.

In this post, we will see

1. How to rename columns of pandas dataframe?
2. How to change row names or row indexes of a pandas dataframe?

Let us first load pandas .

```
1 | # import pandas
2 | >import pandas as pd
```

Let us use gapminder data from software carpentry website.

```
1 | # link to gapminder data
2 | data_url = 'http://bit.ly/2cLzoxH'
3 | # read data from url as pandas dataframe
4 | >gapminder = pd.read_csv(data_url)
```

let us check the names of the columns of the dataframe, the first three rows of the data, using head function.

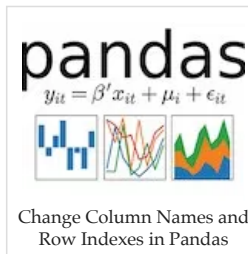
```
1 | >print(gapminder.head(3))
2 |   country  year  pop continent  lifeExp  gdpPercap
3 | 0 Afghanistan 1952  8425333  Asia  28.801  779.445314
4 | 1 Afghanistan 1957  9240934  Asia  30.332  820.853030
5 | 2 Afghanistan 1962 10267083  Asia  31.997  853.100710
```

We can also use columns function to get the column names.

```
1 | >gapminder.columns
2 | Index(['country', 'year', 'pop', 'continent', 'lifeExp', 'gdpPercap'])
```

1. How to Rename Columns in Pandas?

One can change the column names of a pandas dataframe in at least two ways. One way to rename columns in Pandas is to use `df.columns` from Pandas and assign new names directly.



Subscribe to Blog via Email

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Follow me on Twitter

Tweets by [@cmdline_tips](#)

cmdline_tips Retweeted



Adam Pegler

@pegleraj

my favourite new R function is `dplyr::fill()`, which fills a vector (column) with empty elements (cells) with the value in the last non-empty cell. Magic. [#rstats](#)
tidyr.tidyverse.org/reference/fill...

Fill in missing values. —...

Fills missing values in sele...
tidyr.tidyverse.org

Jun 6, 2019

cmdline_tips Retweeted



Probability Fact

@ProbFact

Probability that a pair of cards stay together after a shuffle
johnndcook.com/blog/2017/01/2...

Jun 18, 2019

cmdline_tips Retweeted



Dan Bader

@dbader_org

[PyCoder's Weekly](#) Issue #373 (June 18, 2019) is out: pycoders.com/issues/373

PyCoder's Weekly I Issue #373

Issue #373 of the PyCoder's Weekly ne...
pycoders.com

For example, if you have the names of columns in a list, you can assign the list to column names directly.

Jun 18, 2019

To change the columns of gapminder dataframe, we can assign the list of new column names to gapminder.columns as

```
1 >gapminder.columns = ['country','year','population',
2                       'continent','life_exp','gdp_per_cap']
```

This will assign the names in the list as column names for the data frame “gapminder”. We can check the dataframe to see that if it has new column names using head() function.

```
1 >gapminder.head(3)
2      country  year  population  continent  life_exp  gdp_per_cap
3 0 Afghanistan 1952   8425333      Asia    28.801   779.445314
4 1 Afghanistan 1957   9240934      Asia    30.332   820.853030
5 2 Afghanistan 1962  10267083      Asia    31.997   853.100710
```

A problem with this approach to change column names is that one has to change names of all the columns in the data frame. This approach would not work, if we want to change just change the name of one column.

2. Pandas rename function to Rename Columns

Another way to change column names in pandas is to use **rename** function. Using **rename** to change column names is a much better way than before. One can change names of specific column easily. And not all the column names need to be changed.

To change column names using **rename** function in Pandas, one needs to specify a mapper, a dictionary with old name as keys and new name as values. Here is an example to change many column names using a dictionary. We will also use **inplace=True** to change column names in place.

```
1 >gapminder.rename(columns={'pop':'population',
2                           'lifeExp':'life_exp',
3                           'gdpPercap':'gdp_per_cap'},
4                   inplace=True)
5
6 >print(gapminder.columns)
7
8 Index([u'country', u'year', u'population', u'continent', u'life_
9       u'gdp_per_cap'],
10      dtype='object')
11
12 >gapminder.head(3)
13
14      country  year  population  continent  life_exp  gdp_per_cap
15 0 Afghanistan 1952   8425333      Asia    28.801   779.445314
16 1 Afghanistan 1957   9240934      Asia    30.332   820.853030
17 2 Afghanistan 1962  10267083      Asia    31.997   853.100710
```

One of the biggest advantages of using **rename** function is that we can use rename to change as many column names as we want.

Let us change the name of a single column.

```
1 >gapminder.rename(columns={'pop':'population'}, inplace=True)
2
3 >print(gapminder.columns)
4 Index([u'country', u'year', u'population', u'continent', u'life_
5       u'gdpPercap'],
6      dtype='object')
7
8 >gapminder.head(3)
9
10      country  year  population  continent  lifeExp  gdpPercap
11 0 Afghanistan 1952   8425333      Asia    28.801   779.445314
```

cmdline_tips Retweeted

**Mara Averick**

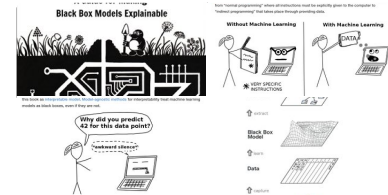
@dataandme

ICYMI, @ChristophMolnar's free book is worth a look!

"Interpretable Machine Learning: A Guide for Making Black Box Models Explainable"

buff.ly/2JqIt08 #MachineLearning

#DataScience



Jun 18, 2019

cmdline_tips Retweeted

**Data Science Renee**

@BecomingDataSci

Does anyone have favorite articles/resources on ML infrastructure, pipeline engineering, and production model maintenance?

Jun 18, 2019

Embed

View on Twitter

Tags

[Basic NumPy](#) [Book Review](#) [Count Lines](#) [Create Directory](#) [Create Soft Link](#) [Data Science](#) [Data Science Books](#) [Data Science Resources](#) [Data Visualization](#) [Dropbox](#) [Dropbox Free Space](#) [Dropbox Tips](#) [Emacs](#) [Emacs Tips](#) [File Size](#) [ggplot2](#) [Linux Commands](#) [Linux Tips](#) [Mac Os X Tips](#) [Maximum Likelihood Estimation in R](#) [MLE in R](#) [NumPy](#) [Pandas Dataframe](#) [Pandas Data Frame](#) [pandas groupby\(\)](#) [Pandas select columns](#) [Python](#) [Python 3](#) [Python Boxplot](#) [Python Tips](#) [R](#) [rstats](#) [R Tips](#) [Seaborn Boxplot](#) [Shell Scripting](#) [Soft Link to Directory](#) [Sort by File Size](#) [Sparse Matrix in Python](#) [tidy evaluation](#) [tidyverse](#) [Vim](#) [Vim Color Schemes](#) [Vim Color Scheme Tips](#) [Vim Color Syntax](#) [Vim Tips](#)

```

11 | 1 Afghanistan 1957 9240934 Asia 30.332 820.853030
12 | 2 Afghanistan 1962 10267083 Asia 31.997 853.100710

```

Pandas **rename** function can also take a function as input instead of a dictionary. For example, we can write a lambda function to take the current column names and consider only the first three characters for the new column names.

```

1 | >gapminder.rename(columns=lambda x: x[0:3], inplace=True)
2 |
3 | >gapminder.head(3)
4 |
5 |      coun year      pop cont  life      gdpP
6 | 0 Afghanistan 1952  8425333 Asia 28.801 779.445314
7 | 1 Afghanistan 1957  9240934 Asia 30.332 820.853030
8 | 2 Afghanistan 1962 10267083 Asia 31.997 853.100710

```

How To Change and Row Names/Indexes in Pandas?

Another good thing about pandas **rename** function is that, we can also use it to change row indexes or row names.

We just need to use **index** argument and specify, we want to change index not columns.

For example, to change row names 0 and 1 to 'zero' and 'one' in our gapminder dataframe, we will construct a dictionary with old row index names as keys and new row index as values.

```

1 | >gapminder.rename(index={0:'zero',1:'one'}, inplace=True)
2 | >print(gapminder.head(4))
3 |
4 |      country year      pop continent  lifeExp  gdpPerCap
5 | zero Afghanistan 1952  8425333 Asia 28.801 779.445314
6 | one Afghanistan 1957  9240934 Asia 30.332 820.853030
7 | 2 Afghanistan 1962 10267083 Asia 31.997 853.100710
8 | 3 Afghanistan 1967 11537966 Asia 34.020 836.197138

```

We can see that just first two rows have new names as we intended.

How To Change Column Names and Row Indexes Simultaneously in Pandas?

With pandas' **rename** function, one can also change both column names and row names simultaneously by using both **column** and **index** arguments to rename function with corresponding mapper dictionaries.

Let us change the column name "lifeExp" to "life_exp" and also row indices "0 & 1" to "zero and one".

```

1 | >gapminder.rename(columns={'lifeExp':'life_exp'},
2 |                  index={0:'zero',1:'one'},
3 |                  inplace=True)
4 | >print(gapminder.head(4))
5 |
6 |      country year      pop continent  life_exp  gdpPerCap
7 | zero Afghanistan 1952  8425333 Asia 28.801 779.445314
8 | one Afghanistan 1957  9240934 Asia 30.332 820.853030
9 | 2 Afghanistan 1962 10267083 Asia 31.997 853.100710
10 | 3 Afghanistan 1967 11537966 Asia 34.020 836.197138

```

Share this:



Related posts:

1. [How To Select One or More Columns in Pandas?](#)
2. [How To Create a Column Using Condition on Another Column in Pandas?](#)
3. [How To Convert a Column to Row Name/Index in Pandas?](#)
4. [How To Split A Column or Column Names in Pandas and Get Part of it?](#)

Filed Under: [change column names in Pandas](#), [change row names in Pandas](#), [Python Tips](#), [rename column names in Pandas](#) Tagged With: [change column names in Pandas](#), [change row names in Pandas](#), [Pandas rename](#), [Python Tips](#)

[RETURN TO TOP OF PAGE](#)

[COPYRIGHT © 2019 ON GENESIS FRAMEWORK · WORDPRESS · LOG IN](#)

u