Open in app

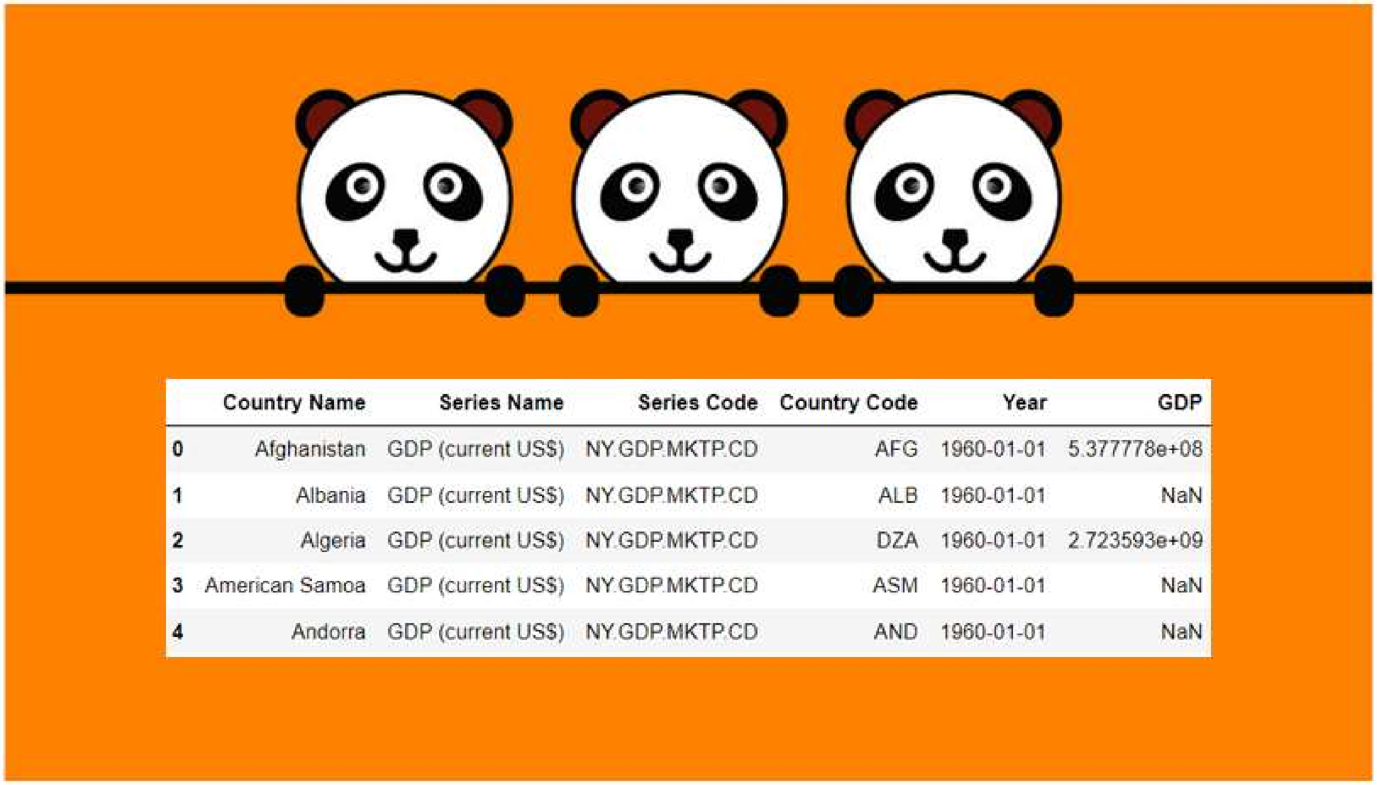
Published in Towards Data Science

Sharone Li

Feb 24 · 4 min read ★ · Listen

Why and How to Reshape a Pandas Dataframe from Wide to Long

A Simple Piece of Code Does the Trick

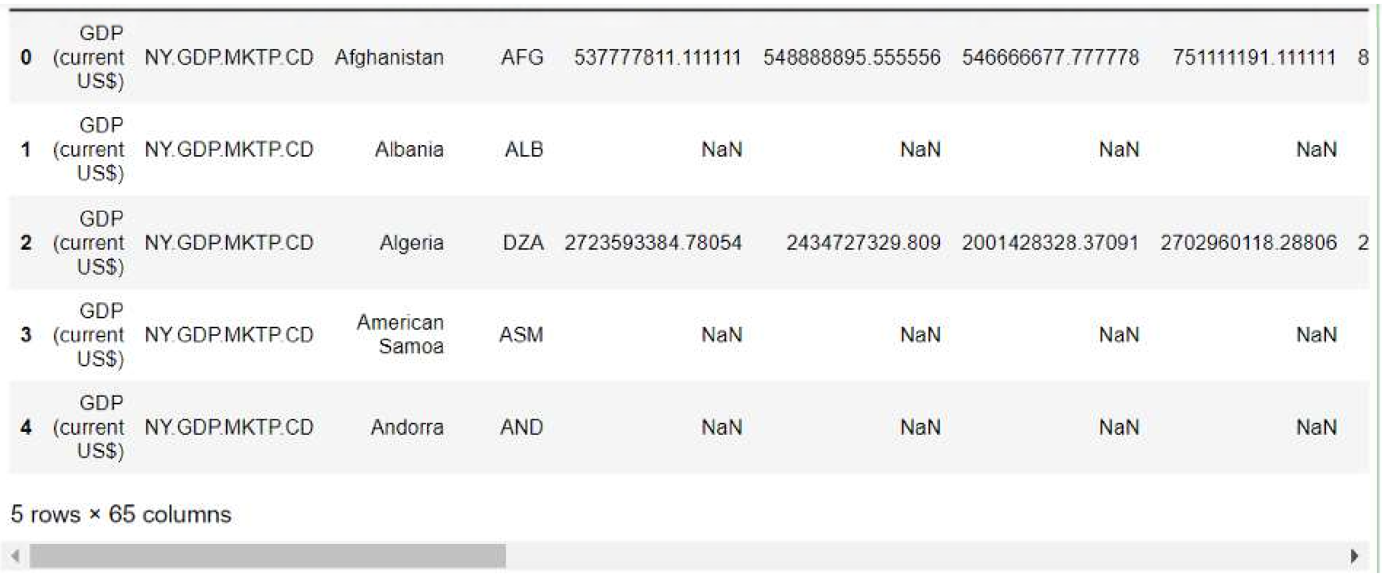


	Country Name	Series Name	Series Code	Country Code	Year	GDP
0	Afghanistan	GDP (current US$)	NY.GDP.MKTP.CD	AFG	1960-01-01	5.377778e+08
1	Albania	GDP (current US$)	NY.GDP.MKTP.CD	ALB	1960-01-01	NaN
2	Algeria	GDP (current US$)	NY.GDP.MKTP.CD	DZA	1960-01-01	2.723593e+09
3	American Samoa	GDP (current US$)	NY.GDP.MKTP.CD	ASM	1960-01-01	NaN
4	Andorra	GDP (current US$)	NY.GDP.MKTP.CD	AND	1960-01-01	NaN

Image by Pixabay (Modified by Author)

As data scientists, we know that data does not always come to us with the most desirable format that's ready for analysis or visualization. Reshaping a table or pandas dataframe from wide to long format falls into one of those scenarios where we need to apply some 'tricks' to the data to proceed with the analysis.

the World Bank's website.



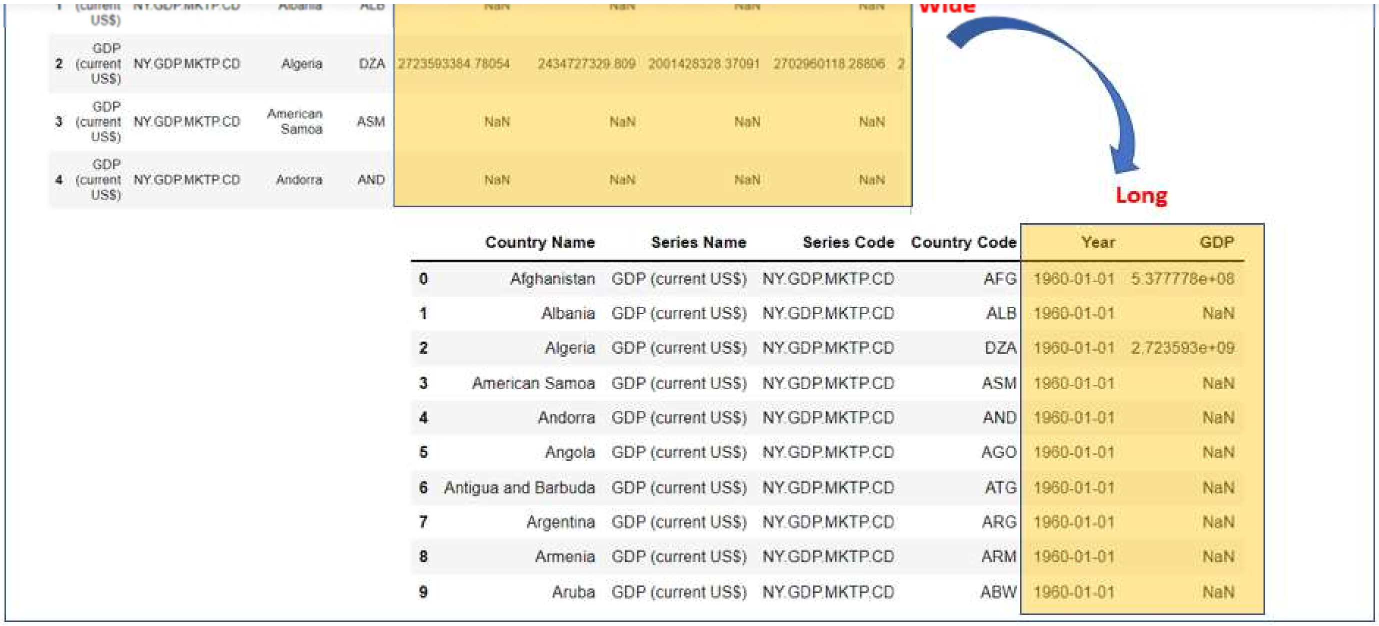
0	GDP (current US$)	NY.GDP.MKTP.CD	Afghanistan	AFG	537777811.111111	548888895.555556	546666677.777778	751111191.111111	8
1	GDP (current US$)	NY.GDP.MKTP.CD	Albania	ALB	NaN	NaN	NaN	NaN	
2	GDP (current US$)	NY.GDP.MKTP.CD	Algeria	DZA	2723593384.78054	2434727329.809	2001428328.37091	2702960118.28806	2
3	GDP (current US$)	NY.GDP.MKTP.CD	American Samoa	ASM	NaN	NaN	NaN	NaN	
4	GDP (current US$)	NY.GDP.MKTP.CD	Andorra	AND	NaN	NaN	NaN	NaN	

5 rows × 65 columns

Image by Author

This is a typical example of a ‘wide-format’ table. For each country, we have its GDP in each year in a separate column, with a total of 61 columns/years of GDP data. Basically, the table extends by year horizontally, making it very wide, thus the name ‘wide-format’.

While a ‘wide-format’ table seems easy to read and is quite common in reality, it is very difficult to plot in Python. We need to ‘transpose’ the table/dataframe so that we have one variable that represents the time column (in our case, the ‘Year’) and another column that has the value (in our case, the GDP) for each year.



1	(current US$)	NY.GDP.MKTP.CD	Albania	ALB	NaN	NaN	NaN	NaN	
2	GDP (current US$)	NY.GDP.MKTP.CD	Algeria	DZA	2723593384.78054	2434727329.809	2001428328.37091	2702960118.28806	2
3	GDP (current US$)	NY.GDP.MKTP.CD	American Samoa	ASM	NaN	NaN	NaN	NaN	
4	GDP (current US$)	NY.GDP.MKTP.CD	Andorra	AND	NaN	NaN	NaN	NaN	

	Country Name	Series Name	Series Code	Country Code	Year	GDP
0	Afghanistan	GDP (current US$)	NY.GDP.MKTP.CD	AFG	1960-01-01	5.377778e+08
1	Albania	GDP (current US$)	NY.GDP.MKTP.CD	ALB	1960-01-01	NaN
2	Algeria	GDP (current US$)	NY.GDP.MKTP.CD	DZA	1960-01-01	2.723593e+09
3	American Samoa	GDP (current US$)	NY.GDP.MKTP.CD	ASM	1960-01-01	NaN
4	Andorra	GDP (current US$)	NY.GDP.MKTP.CD	AND	1960-01-01	NaN
5	Angola	GDP (current US$)	NY.GDP.MKTP.CD	AGO	1960-01-01	NaN
6	Antigua and Barbuda	GDP (current US$)	NY.GDP.MKTP.CD	ATG	1960-01-01	NaN
7	Argentina	GDP (current US$)	NY.GDP.MKTP.CD	ARG	1960-01-01	NaN
8	Armenia	GDP (current US$)	NY.GDP.MKTP.CD	ARM	1960-01-01	NaN
9	Aruba	GDP (current US$)	NY.GDP.MKTP.CD	ABW	1960-01-01	NaN

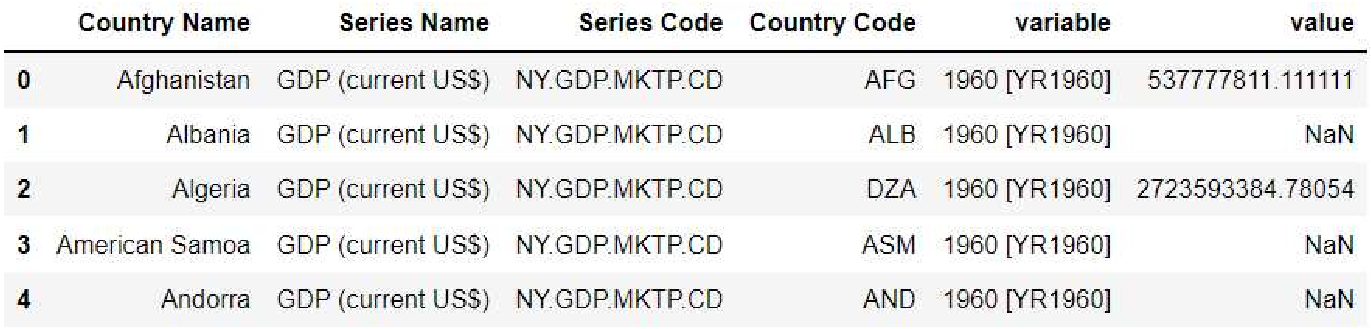
Image by Author

So how to do it in Python? In order to reshape the dataframe from wide to long, we will need to use a pandas function called `pd.melt()` . In the code below:

Line 2: we get the list of the 'Year' columns from the pandas dataframe. These are the columns that are to be transposed.

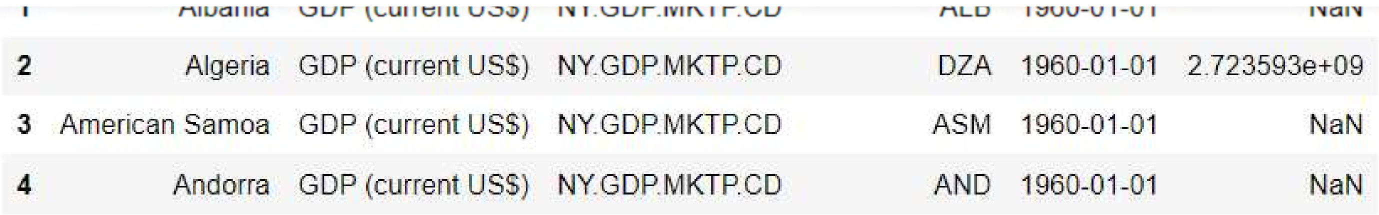
Line 3: we specify which variables in the dataframe are 'id_var' — which basically are the columns that are not to be changed/transposed (i.e., the first four columns in our dataframe). We will also need to specify which variables in the dataframe are to be transposed — in our case, all the year variables (1960[YR1960], 1961[YR1961], 1962[YR1962], etc.)

The reshaped long-format dataframe will have a 'variable' column that represents the year, and a 'value' column that represents the value of each year, i.e., GDP. The column names 'variable' and 'value' are both default names given by `pd.melt()` .



	Country Name	Series Name	Series Code	Country Code	variable	value
0	Afghanistan	GDP (current US$)	NY.GDP.MKTP.CD	AFG	1960 [YR1960]	537777811.111111
1	Albania	GDP (current US$)	NY.GDP.MKTP.CD	ALB	1960 [YR1960]	NaN
2	Algeria	GDP (current US$)	NY.GDP.MKTP.CD	DZA	1960 [YR1960]	2723593384.78054
3	American Samoa	GDP (current US$)	NY.GDP.MKTP.CD	ASM	1960 [YR1960]	NaN
4	Andorra	GDP (current US$)	NY.GDP.MKTP.CD	AND	1960 [YR1960]	NaN

Image by Author



1	Albania	GDP (current US$)	NY.GDP.MKTP.CD	ALB	1960-01-01	NaN
2	Algeria	GDP (current US$)	NY.GDP.MKTP.CD	DZA	1960-01-01	2.723593e+09
3	American Samoa	GDP (current US$)	NY.GDP.MKTP.CD	ASM	1960-01-01	NaN
4	Andorra	GDP (current US$)	NY.GDP.MKTP.CD	AND	1960-01-01	NaN

Image by Author

Bonus: Create a Bar Chart Race Using the Reshaped Dataframe

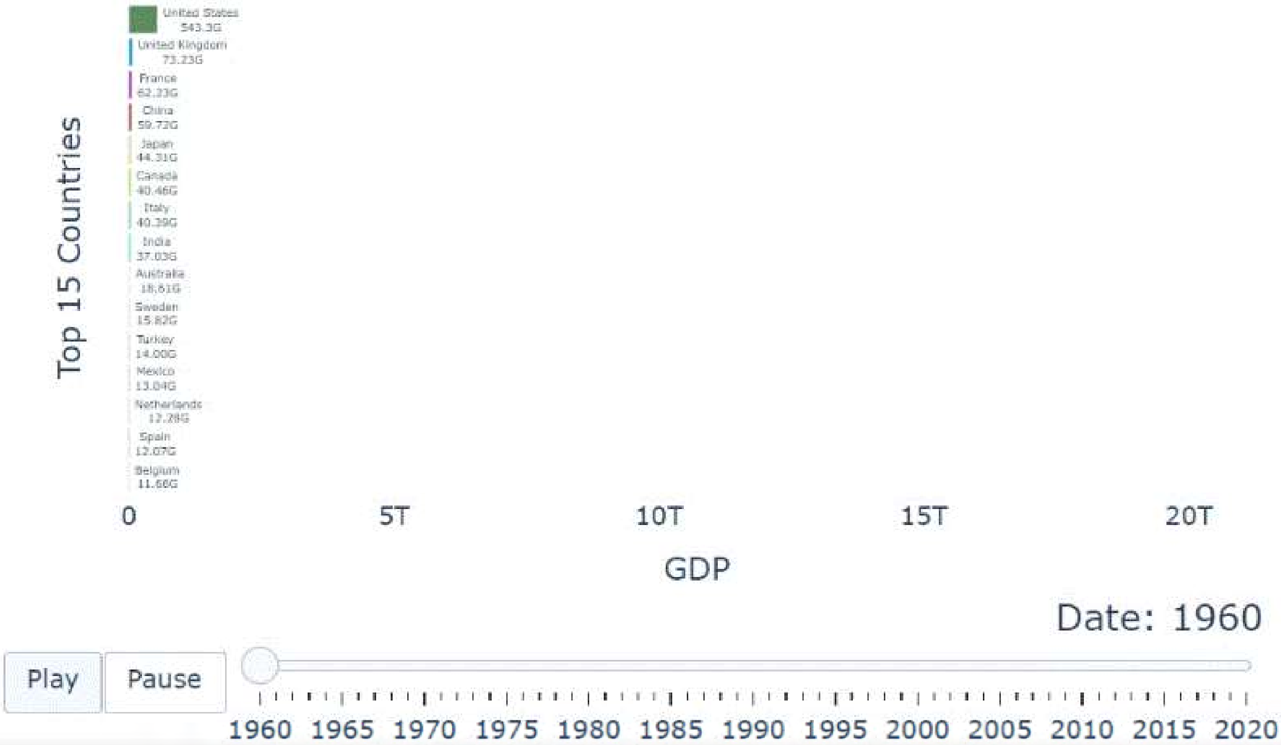
Now we can easily create a bar chart race animation from the **reshaped** dataframe using `raceplotly`. If you don't have `raceplotly`, you can install it by running the following command line:

```
pip install raceplotly
```

The most important and required arguments in `raceplotly` are:

- `item_column`: name of the column representing the items/bars to be ranked ('Country_Name')
- `value_column`: name of the column representing the value/metric to be used for ranking ('GDP')
- `time_column`: name of the column representing the time variable ('Year')

Open in app


Top 15 Countries
United States
543.3G
United Kingdom
73.23G
France
62.23G
China
59.72G
Japan
44.31G
Canada
40.46G
Italy
40.39G
India
37.03G
Australia
18.61G
Sweden
15.82G
Turkey
14.00G
Mexico
13.04G
Netherlands
12.28G
Spain
12.07G
Belgium
11.66G
0
5T
10T
15T
20T
GDP
Date: 1960
Play
Pause
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020

achieve the desired results. Thanks for reading. I hope you enjoyed this quick Python tutorial!

Data Source:

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