Reshaping with melt

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

Wide to long transformation

- Perform analytics
- Plot different variables in the same graph

Wide to long transformation

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	variable	value
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155
6	John	Wick	weight	70
7	Mary	Shelley	weight	60
8	Alice	Liddell	weight	58

```
df.melt(                    )
```

Melt

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	variable	value
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155
6	John	Wick	weight	70
7	Mary	Shelley	weight	60
8	Alice	Liddell	weight	58

```
df.melt(id_vars=                    )
```

Melt

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	variable	value
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155
6	John	Wick	weight	70
7	Mary	Shelley	weight	60
8	Alice	Liddell	weight	58

```
df.melt(id_vars=["first", "last"])
```

Melt

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	variable	value
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155
6	John	Wick	weight	70
7	Mary	Shelley	weight	60
8	Alice	Liddell	weight	58

```
df.melt(id_vars=["first", "last"])
```

Melt

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	variable	value
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155
6	John	Wick	weight	70
7	Mary	Shelley	weight	60
8	Alice	Liddell	weight	58

```
df.melt(id_vars=["first", "last"])
```

Melting data

```
books
```

```
                          title isbn language pages
0                 Mostly Harmless  074      eng   260
1           The Hitchhiker's Guide  072      eng   215
2 El restaurante del fin del mundo  071      spa   250
```

```
books.melt(id_vars='title')
```

```
                                title  variable   value
0                      Mostly Harmless      isbn     074
1                The Hitchhiker's Guide      isbn     072
2  El restaurante del fin del mundo      isbn     071
3                      Mostly Harmless  language     eng
4                The Hitchhiker's Guide  language     eng
5  El restaurante del fin del mundo  language     spa
6                      Mostly Harmless     pages     260
7                The Hitchhiker's Guide     pages     215
8  El restaurante del fin del mundo     pages     250
```

Values and variables

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	feature	amount
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155

```
df.melt(id_vars=["first","last"], value_vars=          , var_name=          , value_name=          )
```

Values and variables

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	feature	amount
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155

```
df.melt(id_vars=["first","last"], value_vars=["age","height"], var_name=          , value_name=          )
```

Values and variables

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	feature	amount
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155

```
df.melt(id_vars=["first", "last"], value_vars=["age", "height"], var_name="feature", value_name="amount")
```

Values and variables

	first	last	age	height	weight
0	John	Wick	50	185	70
1	Mary	Shelley	25	164	60
2	Alice	Liddell	16	155	58

	first	last	feature	amount
0	John	Wick	age	50
1	Mary	Shelley	age	25
2	Alice	Liddell	age	16
3	John	Wick	height	185
4	Mary	Shelley	height	164
5	Alice	Liddell	height	155

```
df.melt(id_vars=["first", "last"], value_vars=["age", "height"], var_name="feature", value_name="amount")
```

Specifying values to melt

```
books.melt(id_vars='title', value_vars=['language_code', 'num_pages'])
```

```
                              title  variable   value
0                    Mostly Harmless  language     eng
1             The Hitchhiker's Guide  language     eng
2  El restaurante del fin del mundo  language     spa
3                    Mostly Harmless     pages     260
4             The Hitchhiker's Guide     pages     215
5  El restaurante del fin del mundo     pages     250
```

Naming values and variables

```
books.melt(id_vars='title', value_vars=['language_code', 'isbn'], var_name='feature', value_name='code')
```

```
                            title   feature    code
0                   Mostly Harmless      isbn     074
1            The Hitchhiker's Guide      isbn     072
2  El restaurante del fin del mundo      isbn     071
3                   Mostly Harmless  language     eng
4            The Hitchhiker's Guide  language     eng
5  El restaurante del fin del mundo  language     spa
```

Let's practice!

RESHAPING DATA WITH PANDAS

Wide to long function

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

Wide to long transformation

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

Wide to long transformation

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

		age	weight
name	year		
John Wick	2019	50	70
Mary Shelley	2019	25	60
Alice Liddell	2019	16	58
John Wick	2020	51	72
Mary Shelley	2020	26	61
Alice Liddell	2020	17	57

```
pd.wide_to_long(                                                    )
```

Wide to long function

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

		age	weight
name	year		
John Wick	2019	50	70
Mary Shelley	2019	25	60
Alice Liddell	2019	16	58
John Wick	2020	51	72
Mary Shelley	2020	26	61
Alice Liddell	2020	17	57

```
pd.wide_to_long(df , stubnames =          , i =          , j =          )
```

Wide to long function

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

		age	weight
name	year		
John Wick	2019	50	70
Mary Shelley	2019	25	60
Alice Liddell	2019	16	58
John Wick	2020	51	72
Mary Shelley	2020	26	61
Alice Liddell	2020	17	57

```
pd.wide_to_long(df , stubnames = ["age", "weight"] , i =          , j =          )
```

Wide to long function

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

		age	weight
name	year		
John Wick	2019	50	70
Mary Shelley	2019	25	60
Alice Liddell	2019	16	58
John Wick	2020	51	72
Mary Shelley	2020	26	61

```
pd.wide_to_long(df , stubnames = ["age", "weight"] , i =          , j = "year" )
```

Wide to long function

	name	age2019	weight2019	age2020	weight2020
0	John Wick	50	70	51	72
1	Mary Shelley	25	60	26	61
2	Alice Liddell	16	58	17	57

		age	weight
name	year		
John Wick	2019	50	70
Mary Shelley	2019	25	60
Alice Liddell	2019	16	58
John Wick	2020	51	72
Mary Shelley	2020	26	61
Alice Liddell	2020	17	57

```
pd.wide_to_long(df , stubnames = ["age", "weight"] , i = "name" , j = "year" )
```

Reshaping data

```
books
```

```
                          title  ratings2019  sold2019  ratings2020  sold2020
0                 Mostly Harmless          4.2       456          4.3       436
1           The Hitchhiker's Guide          4.8       980          4.9       998
2 El restaurante del fin del mundo          4.5       678          4.6       638
```

Reshaping data

```
pd.wide_to_long(books,                                                    )
```

Reshaping data

```
pd.wide_to_long(books, stubnames=['ratings', 'sold']                    )
```

Reshaping data

```
pd.wide_to_long(books, stubnames=['ratings', 'sold'],          , j='year')
```

Reshaping data

```
pd.wide_to_long(books, stubnames=['ratings', 'sold'], i='title', j='year')
```

```
                                    ratings    sold
                          title  year
0                  Mostly Harmless  2019     4.2     456
1            The Hitchhiker's Guide  2019     4.8     980
2 El restaurante del fin del mundo  2019     4.5     678
3                  Mostly Harmless  2020     4.4     436
4            The Hitchhiker's Guide  2020     4.9     998
5 El restaurante del fin del mundo  2020     4.6     638
```

DataFrame with index

```
books_with_index
```

```
                                author  ratings2019 sold2019
                   title
0   To Kill a Mockingbird       Harper Lee          4.7      456
1  The Hitchhiker's Guide    Douglas Adams          4.8      980
2           The Black Cat   Edgar Alan Poe          4.5      678
```

```
pd.wide_to_long(books_with_index, stubnames=['ratings', 'sold'], i='author', j='year')
```

```
                      ratings    sold
          author  year
0     Harper Lee  2019      4.2      456
1  Douglas Adams  2019      4.8      980
2 Edgar Alan Poe  2019      4.5      678
```

DataFrame with index

```
books_with_index.reset_index(drop=False, inplace=True)
pd.wide_to_long(books_with_index, stubnames=['ratings', 'sold'], i=['author', 'title'], j='year')
```

```
                                          ratings  sold
                  title           author    year
0   To Kill a Mockingbird       Harper Lee    2019       4.7   456
1  The Hitchhiker's Guide    Douglas Adams    2019       4.8   980
2           The Black Cat   Edgar Alan Poe    2019       4.5   678
```

sep argument

```
new_books
```

	title	author	ratings_2019	sold_2019	ratings_2020	sold_2020
0	A Murder Is Announced	Agatha Christie	4.4	796	4.8	856
1	Sherlock Holmes	Sir A. Conan Doyle	4.5	780	4.8	818
2	The Sparrow	Mary Doria Russell	4.2	178	4.1	238

sep argument

```
pd.wide_to_long(new_books, stubnames=['ratings', 'sold'], i=['title', 'author'], j='year')
```

```
                    sold_2020 ratings_2020 ratings_2019 sold_2019  ratings sold
title  author year
```

sep argument

```
pd.wide_to_long(new_books, stubnames=['ratings', 'sold'], i=['title', 'author'], j='year', sep='_')
```

				ratings	sold
	title	author	year		
0	A Murder Is Announced	Agatha Christie	2019	4.4	796
1	Sherlock Holmes	Sir A. Conan Doyle	2019	4.5	780
2	The Sparrow	Mary Doria Russell	2019	4.2	178
3	A Murder Is Announced	Agatha Christie	2020	4.8	856
4	Sherlock Holmes	Sir A. Conan Doyle	2020	4.8	818
5	The Sparrow	Mary Doria Russell	2020	4.1	238

suffix argument

```
another_books
```

	title	ratings_one	sold_one	ratings_two	sold_two
0	A Murder Is Announced	4.4	796	4.8	856
1	Sherlock Holmes	4.5	780	4.8	818
2	The Sparrow	4.2	178	4.1	238

suffix argument

```
pd.wide_to_long(another_books, stubnames=['ratings', 'sold'], i='title', j='edition', sep='_')
```

```
            sold_one ratings_one ratings_two sold_two  ratings sold
title  year
```

suffix argument

```
pd.wide_to_long(another_books, stubnames=['ratings', 'sold'], i='title', j='edition', sep='_', suffix='\w+')
```

			ratings	sold
	title	edition		
0	A Murder Is Announced	one	4.4	796
1	Sherlock Holmes	one	4.5	780
2	The Sparrow	one	4.2	178
3	A Murder Is Announced	two	4.8	856
4	Sherlock Holmes	two	4.8	818
5	The Sparrow	two	4.1	238

Let's practice!

RESHAPING DATA WITH PANDAS

Working with string columns

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

Columns with strings

```
books
```

```
                  title  raitings_2015 sold_2015 raitings_2016 sold_2016
0  The Civil War:Vol. 1            4.3       234            4.2       254
1  The Civil War:Vol. 2            4.5       525            4.3       515
2  The Civil War:Vol. 3            4.1       242            4.2       251
```

```
books['title'].dtypes
```

```
dtype('O')
```

String methods

- `pandas` Series string processing methods
- Access easily by `str` attribute

Splitting into two columns

```
books
```

```
                  title  raitings_2015 sold_2015 raitings_2016 sold_2016
0  The Civil War:Vol. 1            4.3       234            4.2       254
1  The Civil War:Vol. 2            4.5       525            4.3       515
2  The Civil War:Vol. 3            4.1       242            4.2       251
```

```
books['title']
```

Splitting into two columns

```
books
```

```
                  title  raitings_2015 sold_2015 raitings_2016 sold_2016
0  The Civil War:Vol. 1            4.3       234            4.2       254
1  The Civil War:Vol. 2            4.5       525            4.3       515
2  The Civil War:Vol. 3            4.1       242            4.2       251
```

```
books['title'].str.split(':')
```

```
0    [The Civil War,  Vol. 1]
1    [The Civil War,  Vol. 2]
2    [The Civil War,  Vol. 3]
```

Splitting into two columns

```
books
```

```
                   title  raitings_2015 sold_2015 raitings_2016 sold_2016
0  The Civil War:Vol. 1             4.3       234            4.2       254
1  The Civil War:Vol. 2             4.5       525            4.3       515
2  The Civil War:Vol. 3             4.1       242            4.2       251
```

```
books['title'].str.split(":").str.get(0)
```

```
0    The Civil War
1    The Civil War
2    The Civil War
```

Splitting into two columns

```
books
```

```
                  title  raitings_2015 sold_2015 raitings_2016 sold_2016
0  The Civil War:Vol. 1            4.3       234            4.2       254
1  The Civil War:Vol. 2            4.5       525            4.3       515
2  The Civil War:Vol. 3            4.1       242            4.2       251
```

```
books['title'].str.split(":", expand=True)
```

```
              0        1
0  The Civil War   Vol. 1
1  The Civil War   Vol. 2
2  The Civil War   Vol. 3
```

Splitting into two columns

```
books[['main_title', 'subtitle']] = books['title'].str.split(":", expand=True)
```

```
books.drop('title', axis=1, inplace=True)
pd.wide_to_long(books , stubnames=['ratings', 'sold'], i=['main_title', 'subtitle'], j='year')
```

```
                          ratings sold
   main_title  subtitle  year
The Civil War    Vol. 1  2015      4.3  234
                         2016      4.2  254
                 Vol. 2  2015      4.5  525
                         2016      4.3  515
                 Vol. 3  2015      4.1  242
                         2016      4.2  251
```

Concatenate two columns

```
books_new
```

```
   name_author  lastname_author  nationality  number_books
0    Virginia             Wolf      British            50
1    Margaret           Atwood     Canadian            40
2      Harper              Lee     American             2
```

Concatenate two columns

```
books_new
```

```
   name_author  lastname_author  nationality  number_books
0    Virginia             Wolf      British            50
1    Margaret           Atwood     Canadian            40
2      Harper              Lee     American             2
```

```
books_new['name_author'].str.cat(books_new['lastname_author'], sep=' ')
```

```
0      Virginia Wolf
1    Margaret Atwood
2         Harper Lee
```

Concatenate two columns

```
books_new
```

```
   name_author  lastname_author  nationality  number_books
0    Virginia             Wolf      British            50
1    Margaret           Atwood     Canadian            40
2      Harper              Lee     American             2
```

```
books_new['author'] = books_new['name_author'].str.cat(books_new['lastname_author'], sep=' ')
books_new
```

```
  name_author lastname_author nationality  number_books           author
0    Virginia            Wolf     British            50    Virginia Wolf
1    Margaret          Atwood    Canadian            40  Margaret Atwood
2      Harper             Lee    American             2       Harper Lee
```

Concatenate two columns

```
books_new
```

```
   name_author  lastname_author  nationality  number_books
0    Virginia             Wolf      British            50
1    Margaret           Atwood     Canadian            40
2      Harper              Lee     American             2
```

```
books_new.melt(id_vars='author', value_vars=['nationality', 'number_books'], var_name='feature', value_name='value')
```

```
            author       feature     value
0    Virginia Wolf   nationality   British
1  Margaret Atwood   nationality  Canadian
2       Harper Lee   nationality  American
3    Virginia Wolf  number_books        50
4  Margaret Atwood  number_books        40
5       Harper Lee  number_books         2
```

Concatenate index

```
comics_marvel
```

```
           subtitle  year  ratings  sold
main_title
Avengers       Next  1992      4.5   234
Avengers    Forever  1998      4.6   224
Avengers       2099  1999      4.8   141
```

Concatenate index

```
comics_marvel.head(2)
```

```
           subtitle  year  ratings  sold
main_title
Avengers       Next  1992      4.5   234
Avengers    Forever  1998      4.6   224
```

```
comics_marvel.index = comics_marvel.index.str.cat(comics_marvel['subtitle'], sep='-')
books
```

```
                subtitle  year  ratings  sold
main_title
Avengers-Next       Next  1992      4.5   234
Avengers-Forever  Forever  1998      4.6   224
Avengers-2099       2099  1999      4.8   141
```

Split index

```
comics_marvel.index = comics_marvel.index.str.split('-', expand=True)
comics_marvel
```

```
                subtitle  year  ratings  sold
Avengers Next       Next  1992      4.5   234
         Forever  Forever  1998      4.6   224
         2099       2099  1999      4.8   141
```

Concatenate Series

```
books_new['name_author']
```

```
0    Virginia
1    Margaret
2      Harper
```

```
new_list = ['Wolf', 'Atwood', 'Lee']
books_new['name_author'].str.cat(new_list, sep=' ')
```

```
0      Virginia Wolf
1    Margaret Atwood
2         Harper Lee
```

Let's practice!

RESHAPING DATA WITH PANDAS