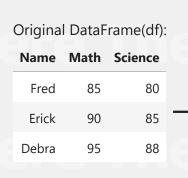
pandas.melt

The **pd.melt()** function reshapes a DataFrame from wide to long format by unpivoting selected columns into rows. This is useful for data normalization and preparing data for analysis.



Name	Subject	Score	
Fred	Math	85	Code Snippet:
Erick	Math	90	pd.melt(
Debra	Math	95	df, id_vars=['Name'],
Fred	Science	80	value_vars=['Math', 'Science'],
Erick	Science	85	var_name='Subject', value name='Score'
Debra	Science	88)

Melted DataFrame:

Contents

What Makes This Ebook Unique

Tips for Effective Use of This Ebook

Introduction

The "frame" Parameter

The "id_vars" Parameter

The "value_vars" Parameter

The "var_name" Parameter

The "value_name" Parameter

The "col_level" Parameter

The "ignore_index" Parameter

Author Biography

Other Ebooks

Contacts and Social Media

Sources & References

Thank You!

1st Edition

Why This E-book?

"The aim of this ebook is to give you the 'aha' moment right away at the start of learning a new concept."

- Practical step By Step Guide With Simple Examples
- Visual Illustrations and Interactive
- Simple Datasets
- Comprehensive Coverage (pandas Documentation used as reference)

Tips for Effective Use of This Ebook

Use Examples First: If you don't understand the text, go straight to the examples—they're self-explanatory. After reviewing them, return to the text to grasp the practical concepts.

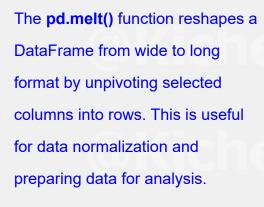
Practice with Datasets: Copy the datasets provided and practice using them to reinforce your understanding.

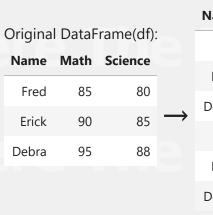
Click on the blue links to go directly to the section you want to learn about.

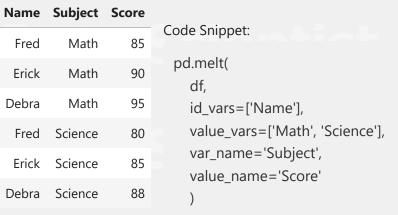
Introduction

```
#import libraries
import numpy as np
import pandas as pd
```

pandas.melt







The **frame** parameter

The DataFrame to be melted.

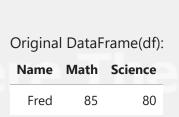
The id_vars parameter

It specifies which columns will remain fixed (identifier variables) while others are "melted" or unpivoted into rows.

It takes a scalar, tuple, list, or ndarray and It's optional.

Debra

When id_vars is a scalar: Here, "Name" is a scalar (single column) used as an identifier, and it remains constant as other



Code Snippe	Melted DataFrame:				
	Score	Subject	Name		
pd.melt(85	Math	Fred		
df,	90	Math	Erick		
,					

Math

columns are unpivoted. Each row in the melted DataFrame will include the "Name", the subject, and the corresponding score.

Name	Math	Science		Name	Subject	Score
Erick	90	85	\rightarrow	Fred	Science	80
Debra	95	88		Erick	Science	85
				Debra	Science	88

id_vars='Name',
var_name='Subject',
value_name='Score'
)

Code Snippet:

df

df = pd.DataFrame({
 'Name': ['Fred', 'Erick', 'Debra'],
 'Math': [85, 90, 95],
 'Science': [80, 85, 88]



df:

10	Name	Math	Science
0	Fred	85	80
1	Erick	90	85
2	Debra	95	88

When id_vars is a tuple:
Here, ("Name", "Year") is a
tuple used as identifiers.
Both columns remain

constant as the other columns are melted.

Each row in the melted

DataFrame will include the

Original DataFrame(df):

9		`	,	
Name	Year	Math	Science	
Fred	2021	85	80	
Erick	2021	90	85	-
Debra	2022	95	88	
	Fred Erick	Name Year Fred 2021 Erick 2021 Debra 2022	Fred 2021 85 Erick 2021 90	Erick 2021 90 85

Name	Year	Subject	Score	Code Snippet:
Fred	2021	Math	85	
Erick	2021	Math	90	pd.melt(df.
Debra	2022	Math	95	id_vars=('Name', 'Year'
Fred	2021	Science	80	var_name='Subject',

"Name", "Year", the subject, and the corresponding score.

Name	Year	Subject	Score
Erick	2021	Science	85
Debra	2022	Science	88

value_name='Score'
)

Code Snippet:

df = pd.DataFrame({

'Name': ['Fred', 'Erick', 'Debra'],

'Year': [2021, 2021, 2022],

'Math': [85, 90, 95],

'Science': [80, 85, 88]

})

df

df:

	Name	Year	Math	Science
0	Fred	2021	85	80
1	Erick	2021	90	85
2	Debra	2022	95	88

When id_vars is a list:

Here, ["Name", "Year"] is a list used as identifiers. All specified columns remain constant while the remaining columns are melted.

Each row in the melted
DataFrame will include the
"Name", "Year", the subject,

Original DataFrame(df):

_		-	•
Name	Year	Math	Science
Fred	2021	85	80
Erick	2021	90	85
Debra	2022	95	88

Name	Year	Subject	Score	Code Snippet:
Fred	2021	Math	85	
Erick	2021	Math	90	pd.melt(df,
Debra	2022	Math	95	id_vars=['Name', 'Year
Fred	2021	Science	80	var_name='Subject',
Erick	2021	Science	85	value_name='Score'
Debra	2022	Science	88)

and the corresponding score.

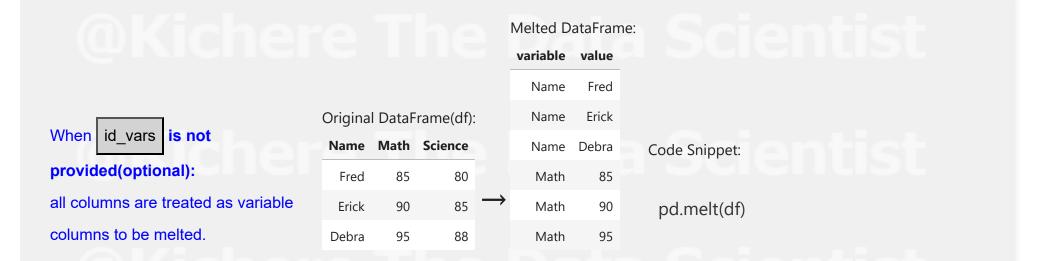
When id_vars is an ndarray:
Here, np.array(["Name", "Year"]) is an ndarray used as identifiers. Both columns remain constant while the other columns are melted.
Each row in the melted
DataFrame will include the "Name", "Year", the subject, and the corresponding

score.

Original DataFrame(df):

Name	Year	Math	Science
Fred	2021	85	80
Erick	2021	90	85
Debra	2022	95	88

Code Snippet: Name Year Subject Score Fred 2021 Math pd.melt(df, Erick 2021 90 Math id_vars=np.array(Debra 2022 Math ['Name', 'Year']), Fred 2021 Science 80 var_name='Subject', 2021 Science value name='Score' Debra 2022 Science 88



variable	value
Science	80
Science	85
Science	88

The value vars parameter

It specifies which column(s) to unpivot

It takes a scalar, tuple, list, or ndarray and It's optional.

When value_vars is a scalar: Here, "Math" is a scalar (single column) used as the variable to melt.

The melted DataFrame will include "Name" as the identifier and the "Math" scores as the variable values.

Original DataFrame(df): Name Math Science

Fred	85	80	
Erick	90	85	-
Debra	95	88	

Melted DataFrame:

	Name	Subject	Score
	Fred	Math	85
\rightarrow	Erick	Math	90
	Debra	Math	95

Code Snippet:

```
pd.melt(
  df,
  id_vars='Name',
  value_vars='Math',
  var_name='Subject',
  value_name='Score'
```



Fred 2021

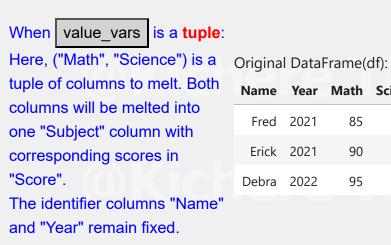
Erick 2021

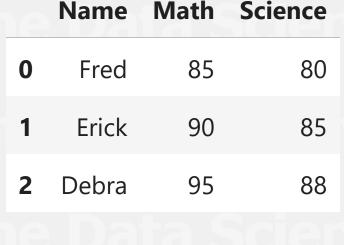
85

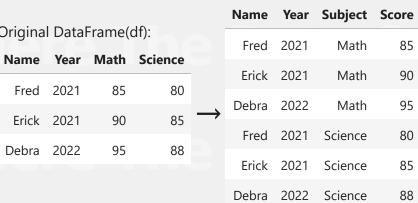
90

95

df:







Melted DataFrame:

Code Snippet: pd.melt(df, id_vars=('Name', 'Year'), value_vars=('Math', 'Science'), var_name='Subject', value name='Score'

```
Code Snippet:
df = pd.DataFrame({
   'Name': ['Fred', 'Erick', 'Debra'],
   'Year': [2021, 2021, 2022],
```

Year Math Science Name 85 80 2021 Fred

'Math': [85, 90, 95], 'Science': [80, 85, 88] }) df

	Name	Year	Math	Science
1	Erick	2021	90	85
2	Debra	2022	95	88



list:

Here, ["Math", "Science"] is used to specify the columns Original DataFrame(df): that will be melted. Both "Math" and "Science" columns are unpivoted into the new "Subject" column. Each row in the melted DataFrame will include "Name", "Year", the subject (Math or Science), and the corresponding score.

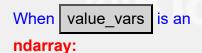
9		•	
Name	Year	Math	Science
Fred	2021	85	80
Erick	2021	90	85
Debra	2022	95	88

Melted DataFrame:

	Name	Year	Subject	Score
	Fred	2021	Math	85
	Erick	2021	Math	90
•	Debra	2022	Math	95
	Fred	2021	Science	80
	Erick	2021	Science	85
	Debra	2022	Science	88

Code Snippet:

```
pd.melt(
  df,
  id_vars=['Name', 'Year'],
  value_vars=['Math', 'Science'],
  var_name='Subject',
  value_name='Score'
```



Here, np.array(["Math", "Science"]) is an ndarray used for value variables.

Original DataFrame(df):



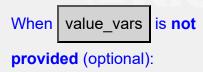
Code Snippet:

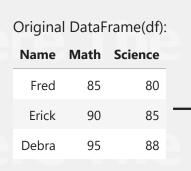
	Score	Subject	Year	Name
1.6	85	Math	2021	Fred
dt, id vars=['Name', 'Year	90	Math	2021	Erick

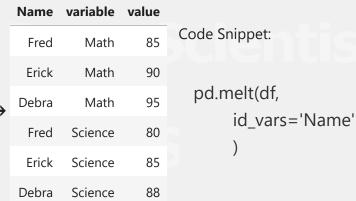
Only the specified columns
are melted.
Each row in the melted
DataFrame will include the
"Name", "Year", the subject,
and the corresponding
score.

Name	Year	Math	Science		Name	Year	Subject	Score
Fred	2021	85	80	\rightarrow	Debra	2022	Math	95
Erick	2021	90	85		Fred	2021	Science	80
Debra	2022	95	88		Erick	2021	Science	85
					Debra	2022	Science	88

value_vars=np.array(
['Math', 'Science']),
var_name='Subject',
value_name='Score'
)







Melted DataFrame:

The var_name parameter

It specifies the name to use for the 'variable' column.

It takes a scalar and defaults to None



When var name is **provided**:

Original DataFrame(df):

Original DataFrame(di).				
Name	Math	Science		
Fred	85	80		
Erick	90	85	-	
Debra	95	88		

Melted DataFrame:

	Name	Subject	value
>	Fred	Math	85
	Erick	Math	90
	Debra	Math	95
	Fred	Science	80
	Erick	Science	85
	Debra	Science	88

Code Snippet:

pd.melt(
 df,
 id_vars='Name',
 var_name='Subject',
)

Willelie!

When var_name is not provided:

Default var_name: "variable"

Original DataFrame(df):

Name	Math	Science	
Fred	85	80	
Erick	90	85	_
Debra	95	88	

Melted DataFrame:

	Name	variable	value	Code Snippet:
→	Fred	Math	85	Scienti
	Erick	Math	90	pd.melt(
	Debra	Math	95	df,
	Fred	Science	80	id_vars='Name',
	Erick	Science	85)
	Debra	Science	88	

df: Code Snippet:

```
df = pd.DataFrame({
    'Name': ['Fred', 'Erick', 'Debra'],
    'Math': [85, 90, 95],
    'Science': [80, 85, 88]
})
df

2
```

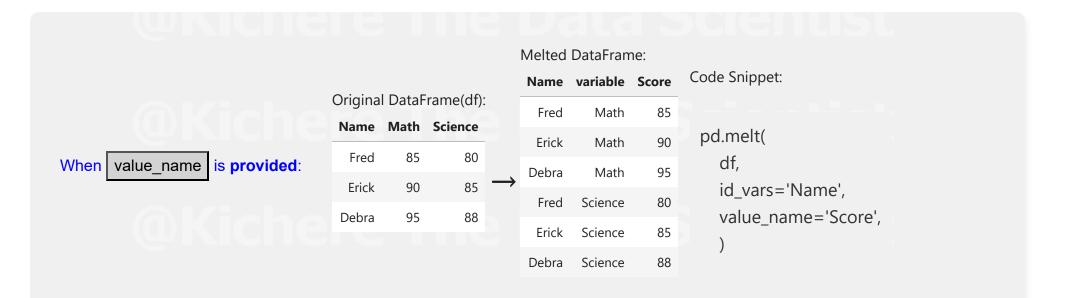
	Name	Math	Science
0	Fred	85	80
1	Erick	90	85
2	Debra	95	88

The value_name parameter

It specifies the name to use for the 'value' column,

It can't be an existing column label.

It takes a scalar and defaults to None



@Kichere The Data Scientist

Melted DataFrame:

When value_name is not provided:

Default value_name: "value"

Original DataFrame(df):					
Name	Math	Science			
Fred	85	80			
Erick	90	85			
Debra	95	88			

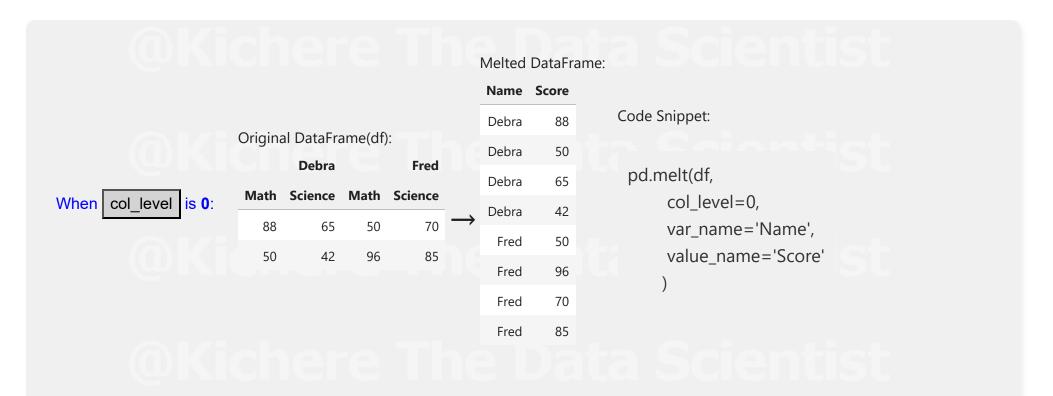
	Name	variable	value	Code Snippet:
	Fred	Math	85	scienti
	Erick	Math	90	pd.melt(
>	Debra	Math	95	df,
	Fred	Science	80	id_vars='Name',
	Erick	Science	85)
	Debra	Science	88	

df: Code Snippet: Name Math Science df = pd.DataFrame({ 'Name': ['Fred', 'Erick', 'Debra'], Fred 85 80 0 'Math': [85, 90, 95], 'Science': [80, 85, 88] Erick 1 90 85 }) Debra 95 88 df

The col_level parameter

It used to melt when columns are a MultiIndex.

It takes a scalar and it is optional.





	Fred		Debra	De		
	Science	Math	Science	Math		
-	70	50	65	88		
	85	96	42	50		

	•	
	Math	88
	Math	50
	Science	65
\rightarrow	Science	42
	Math	50

Melted DataFrame:

Subject Score

Code Snippet:

pd.melt(df, col_level=1, var_name='Subject',

Subject	Score	value_name='Scor
Math	96)
Science	70	
Science	85	

```
Code Snippet:
                                         df:
 df = pd.DataFrame({
                                                        Debra
                                                                             Fred
   ('Debra', 'Math'): [88, 50],
                                              Math Science Math Science
   ('Debra', 'Science'): [65, 42],
   ('Fred', 'Math'): [50, 96],
                                          0
                                                 88
                                                            65
                                                                    50
                                                                               70
   ('Fred', 'Science'): [70, 85]
 })
                                                 50
                                                            42
                                                                    96
                                                                               85
 df
```

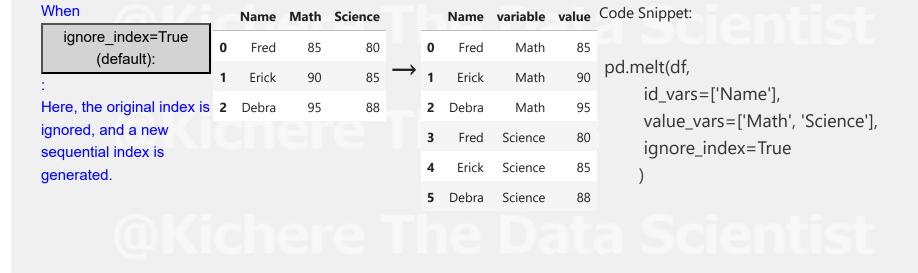
The <u>ignore_index</u> parameter

It used to ignore or retain original index.

It is boolean and defaults to True.

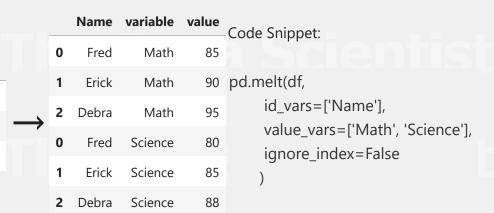
Melted DataFrame:

Original DataFrame(df):



Melted DataFrame:

Original DataFrame(df): When Name Math Science ignore index=False 80 Fred 85 The original index is 85 Erick 90 preserved and repeated for each melted row. **2** Debra 88 95



@Kichere Thata Scientist

Code Snippet:

df = pd.DataFrame({
 'Name': ['Fred', 'Erick', 'Debra'],

0 Fred 85 80

Name Math Science

'Math': [85, 90, 95], 'Science': [80, 85, 88]	\rightarrow		Name	Math	Science
}) df		1	Erick	90	85
Mkichere		2	Debra	95	88

Sources & References

pandas.melt Documentation
(https://pandas.pydata.org/docs/reference/api/pandas.melt.html)

Other E-books on pandas

Download (https://github.com/kicherethedatascientist/My_e-Books)

Author Biography

Kichere Magubu is a data enthusiast and content creator.

Contacts and Social Media

Kichere Magubu

Dar es salaam, Tanzania Kichere Magubu Kichere The Data Scientist KichereTheDataScientist ★ KichereTheDataScientist kicherethedatascientist@gmail.com kicherethedatascientist +255 654 729 851 Powered by Eastern Africa Statistical Training Centre

Thank You!

Thank you for reading this e-book! If you found it valuable, please consider leaving an honest review. Your feedback and support mean a lot to me!

```
#print("The cell to convert jupyter notebook to html")
!jupyter nbconvert --to hide_code_html "melt.ipynb"
```