None

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.

Original DataFrame(df):

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

→

Melted DataFrame:

Name	Subject	Score
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'],
    value_vars=['Math', 'Science'],
    var_name='Subject',
    value_name='Score'
    )
```


Contents

****

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

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

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.

Original DataFrame(df):

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

→

Melted DataFrame:

Name	Subject	Score
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'],
    value_vars=['Math', 'Science'],
    var_name='Subject',
    value_name='Score'
    )
```


Syntax

```
pd.melt(frame,
        id_vars=None,
        value_vars=None,
        var_name=None,
        value_name='value',
        col_level=None,
        ignore_index=True
        )
```

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.

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

Original DataFrame(df):

Name	Math	Science
Fred	85	80

Melted DataFrame:

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

Code Snippet:

```
pd.melt(
    df,
```

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

Name	Math	Science
Erick	90	85
Debra	95	88

→

Name	Subject	Score
Fred	Science	80
Erick	Science	85
Debra	Science	88

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


Code Snippet:

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

→

df:

	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):

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

Code Snippet:

```
pd.melt(
    df,
    id_vars=('Name', 'Year'),
    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

→

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'],
    var_name='Subject',
    value_name='Score'
    )
```

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

→

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=np.array(
        ['Name', 'Year']),
    var_name='Subject',
    value_name='Score'
    )
```

When `id_vars` **is not provided(optional):**

all columns are treated as variable columns to be melted.

Original DataFrame(df):

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

→

Melted DataFrame:

variable	value
Name	Fred
Name	Erick
Name	Debra
Math	85
Math	90
Math	95

Code Snippet:

```
pd.melt(df)
```

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
Erick	Math	90
Debra	Math	95

Code Snippet:

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

df:

Code Snippet:

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

→

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

When `value_vars` is a **tuple**:
Here, ("Math", "Science") is a tuple of columns to melt. Both columns will be melted into one "Subject" column with corresponding scores in "Score".
The identifier columns "Name" and "Year" remain fixed.

Original DataFrame(df):

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'
    )
```

Code Snippet:

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

df:

	Name	Year	Math	Science
0	Fred	2021	85	80

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

→

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

When `value_vars` is a **list**:
Here, ["Math", "Science"] is used to specify the columns 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.

Original DataFrame(df):

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'
    )
```

When `value_vars` is an **ndarray:**
Here, np.array(["Math", "Science"]) is an ndarray used for value variables.

Original DataFrame(df):

Melted DataFrame:

Name	Year	Subject	Score
Fred	2021	Math	85
Erick	2021	Math	90

Code Snippet:

```
pd.melt(
    df,
    id_vars=['Name', 'Year'],
```

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
Fred	2021	85	80
Erick	2021	90	85
Debra	2022	95	88

→

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

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

When `value_vars` is **not provided** (optional):

Original DataFrame(df):

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

→

Melted DataFrame:

Name	variable	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'
        )
```

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):

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',
    )
```

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
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',
    )
```

df:

Code Snippet:

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

→

	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

When `value_name` is **provided**:

Original DataFrame(df):

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

→

Melted DataFrame:

Name	variable	Score
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',
    value_name='Score',
    )
```

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

→

Melted DataFrame:

Name	variable	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',
    )
```

Code Snippet:

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

→

df:

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

The col_level parameter

It used to melt when columns are a MultiIndex.

It takes a scalar and it is optional.

When col_level is **0**:

Original DataFrame(df):

	Debra		Fred	
	Math	**Science**	**Math**	**Science**
	88	65	50	70
	50	42	96	85

→

Melted DataFrame:

Name	**Score**
Debra	88
Debra	50
Debra	65
Debra	42
Fred	50
Fred	96
Fred	70
Fred	85

Code Snippet:

```
pd.melt(df,
        col_level=0,
        var_name='Name',
        value_name='Score'
       )
```

When col_level is **1**:

Original DataFrame(df):

	Debra		Fred	
	Math	**Science**	**Math**	**Science**
	88	65	50	70
	50	42	96	85

→

Melted DataFrame:

Subject	**Score**
Math	88
Math	50
Science	65
Science	42
Math	50

Code Snippet:

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

Subject	Score
Math	96
Science	70
Science	85

```
    value_name='Score'
)
```

Code Snippet:

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

→

df:

	Debra		Fred	
	Math	**Science**	**Math**	**Science**
0	88	65	50	70
1	50	42	96	85

The ignore_index parameter

It used to ignore or retain original index.

It is boolean and defaults to True.

Melted DataFrame:

Original DataFrame(df):

When `ignore_index=True` (default):

:

Here, the original index is ignored, and a new sequential index is generated.

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

→

	Name	variable	value
0	Fred	Math	85
1	Erick	Math	90
2	Debra	Math	95
3	Fred	Science	80
4	Erick	Science	85
5	Debra	Science	88

Code Snippet:

```
pd.melt(df,
        id_vars=['Name'],
        value_vars=['Math', 'Science'],
        ignore_index=True
       )
```

When `ignore_index=False`:

The original index is preserved and repeated for each melted row.

Original DataFrame(df):

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

→

Melted DataFrame:

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

Code Snippet:

```
pd.melt(df,
        id_vars=['Name'],
        value_vars=['Math', 'Science'],
        ignore_index=False
       )
```

Code Snippet:

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

df:

	Name	Math	Science
0	Fred	85	80

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

→

	Name	Math	Science
1	Erick	90	85
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"
```