Wide and long data formats

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

You will learn

- Wide and long formats
- Long to wide transformation
- Wide to long transformation
- Stacking and unstacking columns
- Reshaping and handling complex data, such as string columns or JSON data

Why it is important

- Tidy datasets
- Data is not in the appropriate format for analysis:
  - Human readable vs. statistical analysis
- Nested data in DataFrames is complex to handle
- Get summary statistics for multi-level index DataFrames

Shape of data

- The way in which a dataset is organized in rows and columns

```
fifa_players = pd.read_csv("fifa_players.csv")
fifa_players
```

```
              name    age   nationality          club
0       Lionel Messi     32     Argentina     Barcelona
1  Cristiano Ronaldo     34      Portugal      Juventus
2    Neymar da Silva     27        Brazil  Saint-Germain
```

```
fifa_players.shape
```

```
(3, 4)
```

Wide format

```
fifa_players
```

```
                name    age   nationality           club
0       Lionel Messi     32     Argentina      Barcelona
1  Cristiano Ronaldo     34      Portugal       Juventus
2    Neymar da Silva     27        Brazil  Saint-Germain
```

Wide format

```
fifa_players
```

```
                      vv
                name   | age | nationality          club
0       Lionel Messi   | 32  |  Argentina      Barcelona
1  Cristiano Ronaldo   | 34  |   Portugal       Juventus
2    Neymar da Silva   | 27  |     Brazil  Saint-Germain
                       ^^
```

- Each feature is in a separate column

Wide format

```
fifa_players
```

```
                name    age   nationality           club
0       Lionel Messi     32     Argentina      Barcelona <--
1  Cristiano Ronaldo     34      Portugal       Juventus <--
2    Neymar da Silva     27        Brazil  Saint-Germain <--
```

- Each feature is in a separate column
- Each rows contains many features of the same player

Wide format

```
fifa_players
```

```
                name    age   nationality           club
0       Lionel Messi     32     Argentina      Barcelona
 ------------------------------------------------------
1    Cristiano Ronaldo    NaN  <-    Portugal       Juventus
 ------------------------------------------------------
2    Neymar da Silva     27        Brazil  Saint-Germain
```

- Each feature is in a separate column
- Each rows contains many features of the same player
- No repetition but large number of missing values
- Simple statistics and imputation

Long format

```
fifa_players_long.head()
```

```
                name     variable     value
0 Cristiano Ronaldo  nationality   Portugal
1 Cristiano Ronaldo         club   Juventus
2      Lionel Messi          age         32
3      Lionel Messi  nationality  Argentina
4      Lionel Messi         club  Barcelona
```

Long format

```
fifa_players_long.head()
```

```
                name     variable     value
0   Cristiano Ronaldo  nationality   Portugal <--
1 Cristiano Ronaldo         club  Juventus
2      Lionel Messi          age        32
3        Lionel Messi  nationality  Argentina <--
4      Lionel Messi         club  Barcelona
```

- Each row represents one feature

Long format

```
fifa_players_long.head()
```

```
                name     variable     value
0   Cristiano Ronaldo  nationality   Portugal <--
1   Cristiano Ronaldo         club   Juventus <--
2      Lionel Messi           age          32
3      Lionel Messi  nationality  Argentina
4      Lionel Messi         club  Barcelona
```

- Each row represents one feature
- Multiple rows for each player

Long format

```
fifa_players_long.head()
```

```
   |              name |    variable      value
0  | Cristiano Ronaldo | nationality   Portugal
1  | Cristiano Ronaldo |        club   Juventus
2  |      Lionel Messi |         age         32
3  |      Lionel Messi | nationality  Argentina
4  |      Lionel Messi |        club  Barcelona
         ^^^^^^^^^^^
```

- Each row represents one feature
- Multiple rows for each player
- A column (`name`) to identify same player

Long format

```
fifa_players_long.head()
```

```
                name     variable     value
0 Cristiano Ronaldo  nationality   Portugal
1 Cristiano Ronaldo         club   Juventus
2      Lionel Messi          age         32
3      Lionel Messi  nationality  Argentina
4      Lionel Messi         club  Barcelona
```

- Each row represents one feature
- Multiple rows for each player
- A column (`name`) to identify same player
- Tidy data:
  - Better to summarize data
  - Key-value pairs
  - Preferred for analysis and graphing

Reshaping data

- Transforming a DataFrame or Series structure to adjust it for analysis
  - Transposing a DataFrame

```
fifa_players.set_index('club')
```

```
                          name  age  nationality
        club
    Barcelona        Lionel Messi   32    Argentina
     Juventus   Cristiano Ronaldo  NaN     Portugal
Saint-Germain     Neymar da Silva   27       Brazil
```

Reshaping data

- Transforming a DataFrame or Series structure to adjust it for analysis
  - Transposing a DataFrame

```
fifa_players.set_index('club')[['name', 'nationality']]
```

```
                          name   nationality
        club
    Barcelona        Lionel Messi       Argentina
     Juventus   Cristiano Ronaldo        Portugal
Saint-Germain     Neymar da Silva          Brazil
```

Reshaping data

- Transforming a DataFrame or Series structure to adjust it for analysis
  - Transposing a DataFrame

```
fifa_players.set_index('club')[['name', 'nationality']].transpose()
```

```
       club         Barcelona           Juventus     Saint-Germain
       name      Lionel Messi  Cristiano Ronaldo   Neymar da Silva
nationality         Argentina           Portugal            Brazil
```

Reshaping data

- Converting data from wide to long format and vice versa
- Unit of analysis:
  - Long format -> characteristic of a player
  - Wide format -> each player

Wide to long transformation

- Performed using `pandas` functions, such as:
  - `.melt()`
  - `.wide_to_long()`

Long to wide format

- Transform data using `pandas` methods, for example:
  - `.pivot()`
  - `.pivot_table()`

Let's practice!

RESHAPING DATA WITH PANDAS

Reshaping using pivot method

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

From long to wide

- Demonstrate relationship between two columns
- Time series operations with the variables
- Operation that requires columns to be the unique variable

[1] https://pandas.pydata.org/docs/user_guide/reshaping.html

From long to wide

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

df.pivot(, ,)

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

```
df.pivot(index=        , columns=        , values=        )
```

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

```
df.pivot(index="Year", columns=          , values=          )
```

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

```
df.pivot(index="Year", columns="Name", values= )
```

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

```
df.pivot(index="Year", columns="Name", values="Weight")
```

Pivot method

	Name	Year	Weight
0	John	2013	80
1	Mary	2013	65
2	Mary	2014	68
3	John	2014	83
4	Laura	2014	71

Name	John	Mary	Laura
Year			
2013	80	65	NaN
2014	83	68	71

```
df.pivot(index="Year", columns="Name", values="Weight")
```

Pivoting a dataset

```
fifa = pd.read_csv('fifa_players.csv')
fifa.head()
```

	name	variable	metric_system	imperial_system
0	Cristiano Ronaldo	weight	83	183.00
1	J. Oblak	weight	87	191.00
2	Cristiano Ronaldo	height	187	6.13
3	J. Oblak	height	188	6.16

Pivoting a dataset

```
fifa.pivot(index='name'                                                    )
```

Pivoting a dataset

```
fifa.pivot(index='name', columns='variable'                    )
```

Pivoting a dataset

```
fifa.pivot(index='name', columns='variable', values='metric_system')
```

```
variable  height   weight
name
Cristiano Ronaldo     187       83
J. Oblak              188       87
```

Pivoting multiple columns

```
fifa.pivot(index='name', columns='variable', values=['metric_system', 'imperial_system'])
```

	metric_system		imperial_system	
variable	height	weight	height	weight
name				
Cristiano Ronaldo	187	83	6.13	183.0
J. Oblak	188	87	6.16	191.0

Pivoting multiple columns

	Name	Year	Weight	Age
0	John	2013	80	30
1	Mary	2013	65	28
2	Mary	2014	68	29
3	John	2014	83	31
4	Laura	2014	71	34

	Weight			Age		
Name	John	Mary	Laura	John	Mary	Laura
Year						
2013	80	65	NaN	30	28	NaN
2014	83	68	71	31	29	34

```
df.pivot(index="Year", columns="Name")
```

Pivoting multiple columns

```
fifa.pivot(index="name", columns="variable")
```

	metric_system		imperial_system	
variable	height	weight	height	weight
name				
Cristiano Ronaldo	187	83	6.13	183.0
J. Oblak	188	87	6.16	191.0

Duplicate entries error

```
another_fifa.head()
```

```
                 name    variable  metric_system  imperial_system
0   Cristiano Ronaldo      weight             83           183.00
1            J. Oblak      weight             87           191.00
2   Cristiano Ronaldo      height            187             6.13
3            J. Oblak      height            188             6.16
4   Cristiano Ronaldo      height            187             6.14
```

Duplicate entries error

```
another_fifa.head()
```

```
                name    variable  metric_system  imperial_system
0   Cristiano Ronaldo      weight             83           183.00
1            J. Oblak      weight             87           191.00
  2   Cristiano Ronaldo      height            187             6.13 <--
3            J. Oblak      height           188              6.16
  4   Cristiano Ronaldo      height            187             6.14 <--
```

Duplicate entries error

```
another_fifa.pivot(index="name", columns="variable")
```

```
ValueError: Index contains duplicate entries, cannot reshape
```

```
another_fifa = another_fifa.drop(4, axis=0)
another_fifa.pivot(index="name", columns="variable")
```

```
                  metric_system     imperial_system
variable             height  weight          height   weight
name
Cristiano Ronaldo       187      83            6.13    183.0
J. Oblak                188      87            6.16    191.0
```

Let's practice!

RESHAPING DATA WITH PANDAS

Pivot tables

RESHAPING DATA WITH PANDAS

Maria Eugenia Inzaugarat
Data Scientist

datacamp

Pivot method limitations

```
another_fifa.head()
```

```
                name    variable  metric_system  imperial_system
0   Cristiano Ronaldo      weight             83           183.00
1            J. Oblak      weight             87           191.00
2   Cristiano Ronaldo      height            187             6.13
3     J. Oblak             height            188             6.16
4   Cristiano Ronaldo      height            187             6.14
```

```
another_fifa.pivot(index="name", columns="variable")
```

```
Traceback (most recent call last):
   ValueError: Index contains duplicate entries, cannot reshape
```

Pivot method limitations

- General purpose pivoting
- Index/column pair must be unique
- Cannot aggregate values

Pivot table

- A DataFrame containing statistics that summarizes the data of a larger DataFrame

Name	John	Mary
Year		
2013	80.5	66.5
2014	83	68

Pivot table

	Name	Year	Weight
0	John	2013	80
1	John	2013	81
2	Mary	2013	67
3	Mary	2013	66
4	John	2014	82
5	John	2014	84
6	Mary	2014	69
7	Mary	2014	67

Pivot table

	Name	Year	Weight
0	John	2013	80
1	John	2013	81
2	Mary	2013	67
3	Mary	2013	66
4	John	2014	82
5	John	2014	84
6	Mary	2014	69
7	Mary	2014	67

Name	John	Mary
Year		
2013	80.5	66.5
2014	83	68

df.pivot_table(, , ,)

Pivot table

	Name	Year	Weight
0	John	2013	80
1	John	2013	81
2	Mary	2013	67
3	Mary	2013	66
4	John	2014	82
5	John	2014	84
6	Mary	2014	69
7	Mary	2014	67

Name	John	Mary
Year		
2013	80.5	66.5
2014	83	68

```
df.pivot_table(index="Year", columns="Name",                    )
                                              ,
```

Pivot table

	Name	Year	Weight
0	John	2013	80
1	John	2013	81
2	Mary	2013	67
3	Mary	2013	66
4	John	2014	82
5	John	2014	84
6	Mary	2014	69
7	Mary	2014	67

Name	John	Mary
Year		
2013	80.5	66.5
2014	83	68

```
df.pivot_table(index="Year", columns="Name", values="Weight", aggfunc="mean")
```

Pivot table

```
another_fifa.pivot_table(index="name", columns="variable", aggfunc="mean")
```

```
                  metric_system     imperial_system
variable             height  weight          height   weight
name
Cristiano Ronaldo       187      83           6.135    183.0
J. Oblak                188      87           6.160    191.0
```

Hierarchical indexes

```
fifa_players.head(6)
```

```
      first     last   movement   overall  attacking
0    Lionel    Messi   shooting        92         70
1 Cristiano  Ronaldo   shooting        93         89
2    Lionel    Messi    passing        92         92
3 Cristiano  Ronaldo    passing        82         83
4    Lionel    Messi    passing        96         88
5 Cristiano  Ronaldo    passing        89         84
```

Hierarchical indexes

```
fifa_players.head(6)
```

```
      first     last   movement   overall  attacking
0    Lionel    Messi   shooting        92         70
1 Cristiano  Ronaldo   shooting        93         89
2    Lionel    Messi    passing        92         92
3 Cristiano  Ronaldo    passing        82         83
4    Lionel    Messi    passing        96         88
5 Cristiano  Ronaldo    passing        89         84
```

```
fifa_players.pivot_table(index=                , columns="movement", values=                      , aggfunc=     )
```

Hierarchical indexes

```
fifa_players.head(6)
```

```
      first     last   movement   overall  attacking
0    Lionel    Messi   shooting        92         70
1 Cristiano  Ronaldo   shooting        93         89
2    Lionel    Messi    passing        92         92
3 Cristiano  Ronaldo    passing        82         83
4    Lionel    Messi    passing        96         88
5 Cristiano  Ronaldo    passing        89         84
```

```
fifa_players.pivot_table(index=["first", "last"], columns="movement", values=                    , aggfunc=     )
```

Hierarchical indexes

```
fifa_players.head(6)
```

```
      first     last   movement   overall  attacking
0    Lionel    Messi   shooting        92         70
1 Cristiano  Ronaldo   shooting        93         89
2    Lionel    Messi    passing        92         92
3 Cristiano  Ronaldo    passing        82         83
4    Lionel    Messi    passing        96         88
5 Cristiano  Ronaldo    passing        89         84
```

```
fifa_players.pivot_table(index=["first", "last"], columns="movement", values=["overall", "attacking"], aggfunc="max")
```

```
                       attacking           overall
          movement  passing shooting  passing shooting
    first     last
Cristiano  Ronaldo       84       89       89       93
   Lionel    Messi       92       70       96       92
```

Margins

```
fifa_players.pivot_table(index=["first", "last"], columns="movement", aggfunc="count",              )
```

Margins

```
fifa_players.pivot_table(index=["first", "last"], columns="movement", aggfunc="count", margins=True)
```

		attacking			overall		
	movement	passing	shooting	All	passing	shooting	All
First	Last						
Cristiano	Ronaldo	2	1	3	2	1	3
Lionel	Messi	2	1	3	2	1	3
All		4	2	6	4	2	6

Pivot or pivot table?

Does the DataFrame have more than one value for each index/column pair?

Do you need to have a multi-index in your resulting pivoted DataFrame?

Do you need summary statistics of your large DataFrame?

Yes! Use `.pivot_table()`

Let's practice!

RESHAPING DATA WITH PANDAS