pandas-lecture-5-dec-batch

June 13, 2023

0.1 Pandas-Lecture - 5

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
!gdown 173A59xh2mnpmljCCB9bhC4C5eP2IS6qZ
```

```
Downloading…
From: https://drive.google.com/uc?id=173A59xh2mnpmljCCB9bhC4C5eP2IS6qZ
To: /Users/satish/Desktop/scaler/Dec Tue Batch - DAV-1/Pfizer_1.csv
100%|                    | 1.51k/1.51k [00:00<00:00, 1.86MB/s]
```

[3]:
```
data = pd.read_csv('Pfizer_1.csv')
data
```

[3]:

	Date	Drug_Name	Parameter	1:30:00	2:30:00	3:30:00	4:30:00	5:30:00	6:30:00	7:30:00	8:30:00	9:30:00	10:30:00
0	15-10-2020	diltiazem hydrochloride	Temperature	23.0	22.0	NaN	21.0	21.0	22	23.0	21.0	22.0	20
1	15-10-2020	diltiazem hydrochloride	Pressure	12.0	13.0	NaN	11.0	13.0	14	16.0	16.0	24.0	18
2	15-10-2020	docetaxel injection	Temperature	NaN	17.0								
3	15-10-2020	docetaxel injection	Pressure	NaN	22.0								
4	15-10-2020	ketamine hydrochloride	Temperature	24.0	NaN								
5	15-10-2020	ketamine hydrochloride	Pressure	8.0	NaN								
6	16-10-2020	diltiazem hydrochloride	Temperature	34.0	35.0								
7	16-10-2020	diltiazem hydrochloride	Pressure	18.0	19.0								
8	16-10-2020	docetaxel injection	Temperature	46.0	47.0								
9	16-10-2020	docetaxel injection	Pressure	23.0	24.0								
10	16-10-2020	ketamine hydrochloride	Temperature	8.0	9.0								
11	16-10-2020	ketamine hydrochloride	Pressure	12.0	12.0								
12	17-10-2020	diltiazem hydrochloride	Temperature	20.0	19.0								
13	17-10-2020	diltiazem hydrochloride	Pressure	3.0	4.0								
14	17-10-2020	docetaxel injection	Temperature	12.0	13.0								
15	17-10-2020	docetaxel injection	Pressure	20.0	22.0								
16	17-10-2020	ketamine hydrochloride	Temperature	13.0	14.0								
17	17-10-2020	ketamine hydrochloride	Pressure	8.0	9.0								

```
2      18.0      NaN     17.0       18      NaN      NaN     23.0        23
3      22.0      NaN     22.0       23      NaN      NaN     27.0        26
4       NaN     27.0      NaN       26     25.0     24.0     23.0        22
5       NaN      7.0      NaN        9     10.0     11.0     10.0         9
6      36.0     36.0     37.0       38     37.0     38.0     39.0        40
7      20.0     21.0     22.0       23     24.0     25.0     25.0        24
8       NaN     48.0     48.0       49     50.0     52.0     55.0        56
9       NaN     25.0     26.0       27     28.0     29.0     28.0        28
10     10.0      NaN     11.0       12     12.0     11.0      NaN        13
11     13.0      NaN     15.0       15     15.0     15.0      NaN        16
12     19.0     18.0     17.0       16     15.0      NaN     13.0        14
13      4.0      4.0      6.0        8      9.0      NaN      9.0        11
14     14.0     15.0     16.0       17     18.0     19.0     20.0        21
15     22.0     22.0     22.0       23     25.0     26.0     27.0        28
16     15.0     16.0     17.0       18     19.0     20.0     21.0        22
17     10.0     11.0     11.0       12     12.0     11.0     12.0        13

    11:30:00  12:30:00
0       20.0        21
1       19.0        20
2       25.0        25
3       29.0        28
4       21.0        20
5        9.0        11
6        NaN        42
7        NaN        27
8       57.0        58
9       29.0        30
10      14.0        15
11      17.0        18
12      11.0        10
13      13.0        14
14      22.0        23
15      29.0        28
16      23.0        24
17      14.0        15
```

[4]:
```
data.shape
```

[4]:
```
(18, 15)
```

[5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 15 columns):
 #   Column     Non-Null Count  Dtype
```

```
---  ------      --------------  -----
 0   Date        18 non-null     object
 1   Drug_Name   18 non-null     object
 2   Parameter   18 non-null     object
 3   1:30:00     16 non-null     float64
 4   2:30:00     16 non-null     float64
 5   3:30:00     12 non-null     float64
 6   4:30:00     14 non-null     float64
 7   5:30:00     16 non-null     float64
 8   6:30:00     18 non-null     int64
 9   7:30:00     16 non-null     float64
 10  8:30:00     14 non-null     float64
 11  9:30:00     16 non-null     float64
 12  10:30:00    18 non-null     int64
 13  11:30:00    16 non-null     float64
 14  12:30:00    18 non-null     int64
dtypes: float64(9), int64(3), object(3)
memory usage: 2.2+ KB
```

[7]:
```
pd.melt(data, id_vars=['Date', 'Drug_Name', 'Parameter'])
```

[7]:

	Date	Drug_Name	Parameter	variable	value
0	15-10-2020	diltiazem hydrochloride	Temperature	1:30:00	23.0
1	15-10-2020	diltiazem hydrochloride	Pressure	1:30:00	12.0
2	15-10-2020	docetaxel injection	Temperature	1:30:00	NaN
3	15-10-2020	docetaxel injection	Pressure	1:30:00	NaN
4	15-10-2020	ketamine hydrochloride	Temperature	1:30:00	24.0
..	…	…	…	…	…
211	17-10-2020	diltiazem hydrochloride	Pressure	12:30:00	14.0
212	17-10-2020	docetaxel injection	Temperature	12:30:00	23.0
213	17-10-2020	docetaxel injection	Pressure	12:30:00	28.0
214	17-10-2020	ketamine hydrochloride	Temperature	12:30:00	24.0
215	17-10-2020	ketamine hydrochloride	Pressure	12:30:00	15.0

[216 rows x 5 columns]

[8]:
```
data_melt = pd.melt(data, id_vars=['Date', 'Drug_Name', 'Parameter'],
       var_name='time',
       value_name='reading')
data_melt
```

[8]:

	Date	Drug_Name	Parameter	time	reading
0	15-10-2020	diltiazem hydrochloride	Temperature	1:30:00	23.0
1	15-10-2020	diltiazem hydrochloride	Pressure	1:30:00	12.0
2	15-10-2020	docetaxel injection	Temperature	1:30:00	NaN
3	15-10-2020	docetaxel injection	Pressure	1:30:00	NaN
4	15-10-2020	ketamine hydrochloride	Temperature	1:30:00	24.0

```
..          …                        …           …       …       …
211  17-10-2020  diltiazem hydrochloride     Pressure  12:30:00     14.0
212  17-10-2020      docetaxel injection  Temperature  12:30:00     23.0
213  17-10-2020      docetaxel injection     Pressure  12:30:00     28.0
214  17-10-2020   ketamine hydrochloride  Temperature  12:30:00     24.0
215  17-10-2020   ketamine hydrochloride     Pressure  12:30:00     15.0

[216 rows x 5 columns]
```

[9]:
```
pd.melt?
```

```
Signature:
pd.melt(
    frame: 'DataFrame',
    id_vars=None,
    value_vars=None,
    var_name=None,
    value_name='value',
    col_level=None,
    ignore_index: 'bool' = True,
) -> 'DataFrame'
Docstring:
Unpivot a DataFrame from wide to long format, optionally leaving identifiers set.

This function is useful to massage a DataFrame into a format where one
or more columns are identifier variables (`id_vars`), while all other
columns, considered measured variables (`value_vars`), are "unpivoted" to
the row axis, leaving just two non-identifier columns, 'variable' and
'value'.

Parameters
----------
id_vars : tuple, list, or ndarray, optional
    Column(s) to use as identifier variables.
value_vars : tuple, list, or ndarray, optional
    Column(s) to unpivot. If not specified, uses all columns that
    are not set as `id_vars`.
var_name : scalar
    Name to use for the 'variable' column. If None it uses
    ``frame.columns.name`` or 'variable'.
value_name : scalar, default 'value'
    Name to use for the 'value' column.
col_level : int or str, optional
    If columns are a MultiIndex then use this level to melt.
ignore_index : bool, default True
    If True, original index is ignored. If False, the original index is retained.
    Index labels will be repeated as necessary.
```

```
    .. versionadded:: 1.1.0

Returns
-------
DataFrame
    Unpivoted DataFrame.

See Also
--------
DataFrame.melt : Identical method.
pivot_table : Create a spreadsheet-style pivot table as a DataFrame.
DataFrame.pivot : Return reshaped DataFrame organized
    by given index / column values.
DataFrame.explode : Explode a DataFrame from list-like
        columns to long format.

Notes
-----
Reference :ref:`the user guide <reshaping.melt>` for more examples.

Examples
--------
>>> df = pd.DataFrame({'A': {0: 'a', 1: 'b', 2: 'c'},
…                    'B': {0: 1, 1: 3, 2: 5},
…                    'C': {0: 2, 1: 4, 2: 6}})
>>> df
   A  B  C
0  a  1  2
1  b  3  4
2  c  5  6

>>> pd.melt(df, id_vars=['A'], value_vars=['B'])
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5

>>> pd.melt(df, id_vars=['A'], value_vars=['B', 'C'])
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5
3  a        C      2
4  b        C      4
5  c        C      6

The names of 'variable' and 'value' columns can be customized:
```

```
>>> pd.melt(df, id_vars=['A'], value_vars=['B'],
…         var_name='myVarname', value_name='myValname')
   A myVarname  myValname
0  a         B          1
1  b         B          3
2  c         B          5

Original index values can be kept around:

>>> pd.melt(df, id_vars=['A'], value_vars=['B', 'C'], ignore_index=False)
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5
0  a        C      2
1  b        C      4
2  c        C      6

If you have multi-index columns:

>>> df.columns = [list('ABC'), list('DEF')]
>>> df
   A  B  C
   D  E  F
0  a  1  2
1  b  3  4
2  c  5  6

>>> pd.melt(df, col_level=0, id_vars=['A'], value_vars=['B'])
   A variable  value
0  a        B      1
1  b        B      3
2  c        B      5

>>> pd.melt(df, id_vars=[('A', 'D')], value_vars=[('B', 'E')])
  (A, D) variable_0 variable_1  value
0      a          B          E      1
1      b          B          E      3
2      c          B          E      5
File:      /usr/local/lib/python3.9/site-packages/pandas/core/reshape/melt.py
Type:      function
```

[10]:
```
data_melt.shape
```

[10]:
```
(216, 5)
```

[12]:
```
data_melt
```

[12]:

	Date	Drug_Name	Parameter	time	reading
0	15-10-2020	diltiazem hydrochloride	Temperature	1:30:00	23.0
1	15-10-2020	diltiazem hydrochloride	Pressure	1:30:00	12.0
2	15-10-2020	docetaxel injection	Temperature	1:30:00	NaN
3	15-10-2020	docetaxel injection	Pressure	1:30:00	NaN
4	15-10-2020	ketamine hydrochloride	Temperature	1:30:00	24.0
..	…	…	…	…	…
211	17-10-2020	diltiazem hydrochloride	Pressure	12:30:00	14.0
212	17-10-2020	docetaxel injection	Temperature	12:30:00	23.0
213	17-10-2020	docetaxel injection	Pressure	12:30:00	28.0
214	17-10-2020	ketamine hydrochloride	Temperature	12:30:00	24.0
215	17-10-2020	ketamine hydrochloride	Pressure	12:30:00	15.0

[216 rows x 5 columns]

[ ]:
```

```

[15]:
```
data_melt.pivot(index=['Date', 'Drug_Name', 'Parameter'],
                columns='time',
                values='reading').reset_index()
```

[15]:

time	Date	Drug_Name	Parameter	10:30:00	11:30:00	\
0	15-10-2020	diltiazem hydrochloride	Pressure	18.0	19.0	
1	15-10-2020	diltiazem hydrochloride	Temperature	20.0	20.0	
2	15-10-2020	docetaxel injection	Pressure	26.0	29.0	
3	15-10-2020	docetaxel injection	Temperature	23.0	25.0	
4	15-10-2020	ketamine hydrochloride	Pressure	9.0	9.0	
5	15-10-2020	ketamine hydrochloride	Temperature	22.0	21.0	
6	16-10-2020	diltiazem hydrochloride	Pressure	24.0	NaN	
7	16-10-2020	diltiazem hydrochloride	Temperature	40.0	NaN	
8	16-10-2020	docetaxel injection	Pressure	28.0	29.0	
9	16-10-2020	docetaxel injection	Temperature	56.0	57.0	
10	16-10-2020	ketamine hydrochloride	Pressure	16.0	17.0	
11	16-10-2020	ketamine hydrochloride	Temperature	13.0	14.0	
12	17-10-2020	diltiazem hydrochloride	Pressure	11.0	13.0	
13	17-10-2020	diltiazem hydrochloride	Temperature	14.0	11.0	
14	17-10-2020	docetaxel injection	Pressure	28.0	29.0	
15	17-10-2020	docetaxel injection	Temperature	21.0	22.0	
16	17-10-2020	ketamine hydrochloride	Pressure	13.0	14.0	
17	17-10-2020	ketamine hydrochloride	Temperature	22.0	23.0	

time	12:30:00	1:30:00	2:30:00	3:30:00	4:30:00	5:30:00	6:30:00	7:30:00	\
0	20.0	12.0	13.0	NaN	11.0	13.0	14.0	16.0	
1	21.0	23.0	22.0	NaN	21.0	21.0	22.0	23.0	
2	28.0	NaN	22.0	22.0	NaN	22.0	23.0	NaN	

3	25.0	NaN	17.0	18.0	NaN	17.0	18.0	NaN
4	11.0	8.0	NaN	NaN	7.0	NaN	9.0	10.0
5	20.0	24.0	NaN	NaN	27.0	NaN	26.0	25.0
6	27.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0
7	42.0	34.0	35.0	36.0	36.0	37.0	38.0	37.0
8	30.0	23.0	24.0	NaN	25.0	26.0	27.0	28.0
9	58.0	46.0	47.0	NaN	48.0	48.0	49.0	50.0
10	18.0	12.0	12.0	13.0	NaN	15.0	15.0	15.0
11	15.0	8.0	9.0	10.0	NaN	11.0	12.0	12.0
12	14.0	3.0	4.0	4.0	4.0	6.0	8.0	9.0
13	10.0	20.0	19.0	19.0	18.0	17.0	16.0	15.0
14	28.0	20.0	22.0	22.0	22.0	22.0	23.0	25.0
15	23.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0
16	15.0	8.0	9.0	10.0	11.0	11.0	12.0	12.0
17	24.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0

time	8:30:00	9:30:00
0	16.0	24.0
1	21.0	22.0
2	NaN	27.0
3	NaN	23.0
4	11.0	10.0
5	24.0	23.0
6	25.0	25.0
7	38.0	39.0
8	29.0	28.0
9	52.0	55.0
10	15.0	NaN
11	11.0	NaN
12	NaN	9.0
13	NaN	13.0
14	26.0	27.0
15	19.0	20.0
16	11.0	12.0
17	20.0	21.0

[ ]:

[14]:
```
data_melt.pivot?
```

```
Signature: data_melt.pivot(index=None, columns=None, values=None) -> 'DataFrame'
Docstring:
Return reshaped DataFrame organized by given index / column values.

Reshape data (produce a "pivot" table) based on column values. Uses
unique values from specified `index` / `columns` to form axes of the
resulting DataFrame. This function does not support data
```

```
aggregation, multiple values will result in a MultiIndex in the
columns. See the :ref:`User Guide <reshaping>` for more on reshaping.

Parameters
----------
index : str or object or a list of str, optional
    Column to use to make new frame's index. If None, uses
    existing index.

    .. versionchanged:: 1.1.0
       Also accept list of index names.

columns : str or object or a list of str
    Column to use to make new frame's columns.

    .. versionchanged:: 1.1.0
       Also accept list of columns names.

values : str, object or a list of the previous, optional
    Column(s) to use for populating new frame's values. If not
    specified, all remaining columns will be used and the result will
    have hierarchically indexed columns.

Returns
-------
DataFrame
    Returns reshaped DataFrame.

Raises
------
ValueError:
    When there are any `index`, `columns` combinations with multiple
    values. `DataFrame.pivot_table` when you need to aggregate.

See Also
--------
DataFrame.pivot_table : Generalization of pivot that can handle
    duplicate values for one index/column pair.
DataFrame.unstack : Pivot based on the index values instead of a
    column.
wide_to_long : Wide panel to long format. Less flexible but more
    user-friendly than melt.

Notes
-----
For finer-tuned control, see hierarchical indexing documentation along
with the related stack/unstack methods.
```

```
Reference :ref:`the user guide <reshaping.pivot>` for more examples.

Examples
--------
>>> df = pd.DataFrame({'foo': ['one', 'one', 'one', 'two', 'two',
…                           'two'],
…                    'bar': ['A', 'B', 'C', 'A', 'B', 'C'],
…                    'baz': [1, 2, 3, 4, 5, 6],
…                    'zoo': ['x', 'y', 'z', 'q', 'w', 't']})
>>> df
    foo   bar  baz  zoo
0   one   A    1    x
1   one   B    2    y
2   one   C    3    z
3   two   A    4    q
4   two   B    5    w
5   two   C    6    t

>>> df.pivot(index='foo', columns='bar', values='baz')
bar  A   B   C
foo
one  1   2   3
two  4   5   6

>>> df.pivot(index='foo', columns='bar')['baz']
bar  A   B   C
foo
one  1   2   3
two  4   5   6

>>> df.pivot(index='foo', columns='bar', values=['baz', 'zoo'])
      baz       zoo
bar   A  B  C   A  B  C
foo
one   1  2  3   x  y  z
two   4  5  6   q  w  t

You could also assign a list of column names or a list of index names.

>>> df = pd.DataFrame({
…        "lev1": [1, 1, 1, 2, 2, 2],
…        "lev2": [1, 1, 2, 1, 1, 2],
…        "lev3": [1, 2, 1, 2, 1, 2],
…        "lev4": [1, 2, 3, 4, 5, 6],
…        "values": [0, 1, 2, 3, 4, 5]})
>>> df
    lev1 lev2 lev3 lev4 values
0   1    1    1    1    0
```

```
1   1    1    2    2    1
2   1    2    1    3    2
3   2    1    2    4    3
4   2    1    1    5    4
5   2    2    2    6    5

>>> df.pivot(index="lev1", columns=["lev2", "lev3"],values="values")
lev2    1         2
lev3    1    2    1    2
lev1
1     0.0  1.0  2.0  NaN
2     4.0  3.0  NaN  5.0

>>> df.pivot(index=["lev1", "lev2"], columns=["lev3"],values="values")
      lev3    1    2
lev1  lev2
   1     1  0.0  1.0
         2  2.0  NaN
   2     1  4.0  3.0
         2  NaN  5.0

A ValueError is raised if there are any duplicates.

>>> df = pd.DataFrame({"foo": ['one', 'one', 'two', 'two'],
…                    "bar": ['A', 'A', 'B', 'C'],
…                    "baz": [1, 2, 3, 4]})
>>> df
   foo bar  baz
0  one   A    1
1  one   A    2
2  two   B    3
3  two   C    4

Notice that the first two rows are the same for our `index`
and `columns` arguments.

>>> df.pivot(index='foo', columns='bar', values='baz')
Traceback (most recent call last):
   …
ValueError: Index contains duplicate entries, cannot reshape
File:      /usr/local/lib/python3.9/site-packages/pandas/core/frame.py
Type:      method
```

[17]:
```
data_tidy = data_melt.pivot(index=['Date', 'Drug_Name', 'time'],
                columns='Parameter',
                values='reading').reset_index()
```

```
data_tidy
```

[17]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0
..                ...                      ...       ...       ...
103        17-10-2020   ketamine hydrochloride   5:30:00      11.0
104        17-10-2020   ketamine hydrochloride   6:30:00      12.0
105        17-10-2020   ketamine hydrochloride   7:30:00      12.0
106        17-10-2020   ketamine hydrochloride   8:30:00      11.0
107        17-10-2020   ketamine hydrochloride   9:30:00      12.0

Parameter  Temperature
0                 20.0
1                 20.0
2                 21.0
3                 23.0
4                 22.0
..                 ...
103               17.0
104               18.0
105               19.0
106               20.0
107               21.0

[108 rows x 5 columns]
```

[18]:
```
data.head()
```

[18]:
```
         Date                Drug_Name    Parameter  1:30:00  2:30:00  \
0  15-10-2020  diltiazem hydrochloride  Temperature     23.0     22.0
1  15-10-2020  diltiazem hydrochloride     Pressure     12.0     13.0
2  15-10-2020      docetaxel injection  Temperature      NaN     17.0
3  15-10-2020      docetaxel injection     Pressure      NaN     22.0
4  15-10-2020   ketamine hydrochloride  Temperature     24.0      NaN

   3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
0      NaN     21.0     21.0       22     23.0     21.0     22.0        20
1      NaN     11.0     13.0       14     16.0     16.0     24.0        18
2     18.0      NaN     17.0       18      NaN      NaN     23.0        23
3     22.0      NaN     22.0       23      NaN      NaN     27.0        26
4      NaN     27.0      NaN       26     25.0     24.0     23.0        22

   11:30:00  12:30:00
```

```
0      20.0        21
1      19.0        20
2      25.0        25
3      29.0        28
4      21.0        20
```

[19]:
```
data_melt.head()
```

[19]:
```
         Date                Drug_Name    Parameter     time  reading
0  15-10-2020  diltiazem hydrochloride  Temperature  1:30:00     23.0
1  15-10-2020  diltiazem hydrochloride     Pressure  1:30:00     12.0
2  15-10-2020      docetaxel injection  Temperature  1:30:00      NaN
3  15-10-2020      docetaxel injection     Pressure  1:30:00      NaN
4  15-10-2020   ketamine hydrochloride  Temperature  1:30:00     24.0
```

[20]:
```
data_tidy.head()
```

[20]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0

Parameter  Temperature
0                 20.0
1                 20.0
2                 21.0
3                 23.0
4                 22.0
```

0.1.1 Handling Missing Values

[21]:
```
data.head()
```

[21]:
```
         Date                Drug_Name    Parameter  1:30:00  2:30:00  \
0  15-10-2020  diltiazem hydrochloride  Temperature     23.0     22.0
1  15-10-2020  diltiazem hydrochloride     Pressure     12.0     13.0
2  15-10-2020      docetaxel injection  Temperature      NaN     17.0
3  15-10-2020      docetaxel injection     Pressure      NaN     22.0
4  15-10-2020   ketamine hydrochloride  Temperature     24.0      NaN

   3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
0      NaN     21.0     21.0       22     23.0     21.0     22.0        20
1      NaN     11.0     13.0       14     16.0     16.0     24.0        18
2     18.0      NaN     17.0       18      NaN      NaN     23.0        23
3     22.0      NaN     22.0       23      NaN      NaN     27.0        26
```

```
4      NaN     27.0      NaN       26     25.0     24.0     23.0         22

   11:30:00  12:30:00
0      20.0        21
1      19.0        20
2      25.0        25
3      29.0        28
4      21.0        20
```

[22]:
```
type(None)
```

[22]:
```
NoneType
```

[23]:
```
type(np.nan)
```

[23]:
```
float
```

[24]:
```
pd.Series([1, np.nan, 3])
```

[24]:
```
0    1.0
1    NaN
2    3.0
dtype: float64
```

[25]:
```
pd.Series([1, np.nan, 3, None])
```

[25]:
```
0    1.0
1    NaN
2    3.0
3    NaN
dtype: float64
```

[26]:
```
pd.Series(['1', 'np.nan', '3', None])
```

[26]:
```
0         1
1    np.nan
2         3
3      None
dtype: object
```

[27]:
```
pd.Series(['1', 'np.nan', '3', np.nan])
```

[27]:
```
0         1
1    np.nan
2         3
3       NaN
dtype: object
```

How to know number of missing values/data in rows/colums?

[32]:
```
data.isnull().sum()
```

[32]:
```
Date          0
Drug_Name     0
Parameter     0
1:30:00       2
2:30:00       2
3:30:00       6
4:30:00       4
5:30:00       2
6:30:00       0
7:30:00       2
8:30:00       4
9:30:00       2
10:30:00      0
11:30:00      2
12:30:00      0
dtype: int64
```

[33]:
```
data.isnull().sum(axis=1)
```

[33]:
```
0     1
1     1
2     4
3     4
4     3
5     3
6     1
7     1
8     1
9     1
10    2
11    2
12    1
13    1
14    0
15    0
16    0
17    0
dtype: int64
```

[37]:
```
data.dropna()
```

[37]:

	Date	Drug_Name	Parameter	1:30:00	2:30:00	\
14	17-10-2020	docetaxel injection	Temperature	12.0	13.0	

```
15  17-10-2020     docetaxel injection     Pressure     20.0     22.0
16  17-10-2020  ketamine hydrochloride  Temperature     13.0     14.0
17  17-10-2020  ketamine hydrochloride     Pressure      8.0      9.0

    3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
14     14.0     15.0     16.0       17     18.0     19.0     20.0        21
15     22.0     22.0     22.0       23     25.0     26.0     27.0        28
16     15.0     16.0     17.0       18     19.0     20.0     21.0        22
17     10.0     11.0     11.0       12     12.0     11.0     12.0        13

    11:30:00  12:30:00
14      22.0        23
15      29.0        28
16      23.0        24
17      14.0        15
```

[38]:
```
data.fillna(0)
```

[38]:
```
           Date                Drug_Name    Parameter  1:30:00  2:30:00  \
0   15-10-2020  diltiazem hydrochloride  Temperature     23.0     22.0
1   15-10-2020  diltiazem hydrochloride     Pressure     12.0     13.0
2   15-10-2020     docetaxel injection  Temperature      0.0     17.0
3   15-10-2020     docetaxel injection     Pressure      0.0     22.0
4   15-10-2020   ketamine hydrochloride  Temperature     24.0      0.0
5   15-10-2020   ketamine hydrochloride     Pressure      8.0      0.0
6   16-10-2020  diltiazem hydrochloride  Temperature     34.0     35.0
7   16-10-2020  diltiazem hydrochloride     Pressure     18.0     19.0
8   16-10-2020     docetaxel injection  Temperature     46.0     47.0
9   16-10-2020     docetaxel injection     Pressure     23.0     24.0
10  16-10-2020   ketamine hydrochloride  Temperature      8.0      9.0
11  16-10-2020   ketamine hydrochloride     Pressure     12.0     12.0
12  17-10-2020  diltiazem hydrochloride  Temperature     20.0     19.0
13  17-10-2020  diltiazem hydrochloride     Pressure      3.0      4.0
14  17-10-2020     docetaxel injection  Temperature     12.0     13.0
15  17-10-2020     docetaxel injection     Pressure     20.0     22.0
16  17-10-2020   ketamine hydrochloride  Temperature     13.0     14.0
17  17-10-2020   ketamine hydrochloride     Pressure      8.0      9.0

    3:30:00  4:30:00  5:30:00  6:30:00  7:30:00  8:30:00  9:30:00  10:30:00  \
0       0.0     21.0     21.0       22     23.0     21.0     22.0        20
1       0.0     11.0     13.0       14     16.0     16.0     24.0        18
2      18.0      0.0     17.0       18      0.0      0.0     23.0        23
3      22.0      0.0     22.0       23      0.0      0.0     27.0        26
4       0.0     27.0      0.0       26     25.0     24.0     23.0        22
5       0.0      7.0      0.0        9     10.0     11.0     10.0         9
6      36.0     36.0     37.0       38     37.0     38.0     39.0        40
7      20.0     21.0     22.0       23     24.0     25.0     25.0        24
```

```
8       0.0     48.0     48.0       49     50.0     52.0     55.0        56
9       0.0     25.0     26.0       27     28.0     29.0     28.0        28
10     10.0      0.0     11.0       12     12.0     11.0      0.0        13
11     13.0      0.0     15.0       15     15.0     15.0      0.0        16
12     19.0     18.0     17.0       16     15.0      0.0     13.0        14
13      4.0      4.0      6.0        8      9.0      0.0      9.0        11
14     14.0     15.0     16.0       17     18.0     19.0     20.0        21
15     22.0     22.0     22.0       23     25.0     26.0     27.0        28
16     15.0     16.0     17.0       18     19.0     20.0     21.0        22
17     10.0     11.0     11.0       12     12.0     11.0     12.0        13

    11:30:00  12:30:00
0       20.0        21
1       19.0        20
2       25.0        25
3       29.0        28
4       21.0        20
5        9.0        11
6        0.0        42
7        0.0        27
8       57.0        58
9       29.0        30
10      14.0        15
11      17.0        18
12      11.0        10
13      13.0        14
14      22.0        23
15      29.0        28
16      23.0        24
17      14.0        15
```

[41]:
```
data['2:30:00'].fillna(data['2:30:00'].mean())
```

[41]:
```
0     22.0000
1     13.0000
2     17.0000
3     22.0000
4     18.8125
5     18.8125
6     35.0000
7     19.0000
8     47.0000
9     24.0000
10     9.0000
11    12.0000
12    19.0000
13     4.0000
```

```
14    13.0000
15    22.0000
16    14.0000
17     9.0000
Name: 2:30:00, dtype: float64
```

[42]:
```
data_tidy
```

[42]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0
..                ...                      ...       ...       ...
103        17-10-2020   ketamine hydrochloride   5:30:00      11.0
104        17-10-2020   ketamine hydrochloride   6:30:00      12.0
105        17-10-2020   ketamine hydrochloride   7:30:00      12.0
106        17-10-2020   ketamine hydrochloride   8:30:00      11.0
107        17-10-2020   ketamine hydrochloride   9:30:00      12.0

Parameter  Temperature
0                20.0
1                20.0
2                21.0
3                23.0
4                22.0
..                ...
103              17.0
104              18.0
105              19.0
106              20.0
107              21.0

[108 rows x 5 columns]
```

[43]:
```
def temp_mean(x):
    x['temp_avg'] = x['Temperature'].mean()
    return x

data_tidy = data_tidy.groupby('Drug_Name').apply(temp_mean)
data_tidy
```

[43]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
```

```
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0
..                ...                      ...       ...       ...
103        17-10-2020   ketamine hydrochloride   5:30:00      11.0
104        17-10-2020   ketamine hydrochloride   6:30:00      12.0
105        17-10-2020   ketamine hydrochloride   7:30:00      12.0
106        17-10-2020   ketamine hydrochloride   8:30:00      11.0
107        17-10-2020   ketamine hydrochloride   9:30:00      12.0

Parameter  Temperature   temp_avg
0                 20.0  24.848485
1                 20.0  24.848485
2                 21.0  24.848485
3                 23.0  24.848485
4                 22.0  24.848485
..                 ...        ...
103               17.0  17.709677
104               18.0  17.709677
105               19.0  17.709677
106               20.0  17.709677
107               21.0  17.709677

[108 rows x 6 columns]
```

[46]:
```
data_tidy['Temperature'].fillna(data_tidy['temp_avg'], inplace=True)
data_tidy[:20]
```

[46]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0
5          15-10-2020  diltiazem hydrochloride   3:30:00       NaN
6          15-10-2020  diltiazem hydrochloride   4:30:00      11.0
7          15-10-2020  diltiazem hydrochloride   5:30:00      13.0
8          15-10-2020  diltiazem hydrochloride   6:30:00      14.0
9          15-10-2020  diltiazem hydrochloride   7:30:00      16.0
10         15-10-2020  diltiazem hydrochloride   8:30:00      16.0
11         15-10-2020  diltiazem hydrochloride   9:30:00      24.0
12         15-10-2020      docetaxel injection  10:30:00      26.0
13         15-10-2020      docetaxel injection  11:30:00      29.0
14         15-10-2020      docetaxel injection  12:30:00      28.0
15         15-10-2020      docetaxel injection   1:30:00       NaN
16         15-10-2020      docetaxel injection   2:30:00      22.0
17         15-10-2020      docetaxel injection   3:30:00      22.0
18         15-10-2020      docetaxel injection   4:30:00       NaN
```

```
19          15-10-2020      docetaxel injection   5:30:00      22.0

Parameter  Temperature   temp_avg
0            20.000000  24.848485
1            20.000000  24.848485
2            21.000000  24.848485
3            23.000000  24.848485
4            22.000000  24.848485
5            24.848485  24.848485
6            21.000000  24.848485
7            21.000000  24.848485
8            22.000000  24.848485
9            23.000000  24.848485
10           21.000000  24.848485
11           22.000000  24.848485
12           23.000000  30.387097
13           25.000000  30.387097
14           25.000000  30.387097
15           30.387097  30.387097
16           17.000000  30.387097
17           18.000000  30.387097
18           30.387097  30.387097
19           17.000000  30.387097
```

[47]:
```
def pressure_mean(x):
    x['pressure_avg'] = x['Pressure'].mean()
    return x

data_tidy = data_tidy.groupby('Drug_Name').apply(pressure_mean)
data_tidy
```

[47]:
```
Parameter        Date                Drug_Name      time  Pressure  \
0          15-10-2020  diltiazem hydrochloride  10:30:00      18.0
1          15-10-2020  diltiazem hydrochloride  11:30:00      19.0
2          15-10-2020  diltiazem hydrochloride  12:30:00      20.0
3          15-10-2020  diltiazem hydrochloride   1:30:00      12.0
4          15-10-2020  diltiazem hydrochloride   2:30:00      13.0
..                ...                      ...       ...       ...
103        17-10-2020   ketamine hydrochloride   5:30:00      11.0
104        17-10-2020   ketamine hydrochloride   6:30:00      12.0
105        17-10-2020   ketamine hydrochloride   7:30:00      12.0
106        17-10-2020   ketamine hydrochloride   8:30:00      11.0
107        17-10-2020   ketamine hydrochloride   9:30:00      12.0

Parameter  Temperature   temp_avg  pressure_avg
0                 20.0  24.848485     15.424242
1                 20.0  24.848485     15.424242
```

```
2                 21.0  24.848485     15.424242
3                 23.0  24.848485     15.424242
4                 22.0  24.848485     15.424242
..                 …        …            …
103               17.0  17.709677     11.935484
104               18.0  17.709677     11.935484
105               19.0  17.709677     11.935484
106               20.0  17.709677     11.935484
107               21.0  17.709677     11.935484

[108 rows x 7 columns]
```