

RA27246_aula6_ex6-5_cap8

April 26, 2024

0.1 IA376I – Tópicos em Engenharia de Computação VII

0.1.1 Tópico: Análise de Dados Visual (Visual Analytics)

Professora: Wu, Shin - Ting Aluno: Luiz Roberto Albano Junior RA: 272746

Aula 06 - 19/04/2024

0.1.2 Exercícios 6.5

Exercício 2: Reproduza os exemplos fornecidos nos Capítulos 6, 7 e 8 em [89] (Python) ou nos Capítulos 9 a 16 em [90] (R). Em ambas as referências, são abordadas diversas funções adicionais de manipulação dos dados, além das apresentadas neste capítulo, proporcionando uma visão mais abrangente das capacidades das respectivas linguagens de programação.

Reprodução dos passos:

Capítulo 8: organização de dados

Indexação hierárquica

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

#Base da URL do repositório com os arquivos de exemplo utilizados no livro
base_url = "https://raw.githubusercontent.com/wesm/pydata-book/3rd-edition/"
```

```
[ ]: data = pd.Series(np.random.uniform(size=9),
                      index=[["a", "a", "a", "b", "b", "c", "c", "d", "d"],
                             [1, 2, 3, 1, 3, 1, 2, 2, 3]])

data
```

```
[ ]: a 1    0.562020
     2    0.242045
     3    0.990003
     b 1    0.301238
     3    0.325411
```

```
c 1    0.785504
   2    0.986494
d 2    0.741712
   3    0.435975
dtype: float64
```

```
[ ]: data.index
```

```
[ ]: MultiIndex([('a', 1),
                 ('a', 2),
                 ('a', 3),
                 ('b', 1),
                 ('b', 3),
                 ('c', 1),
                 ('c', 2),
                 ('d', 2),
                 ('d', 3)],
                )
```

```
[ ]: data["b"]
```

```
[ ]: 1    0.301238
     3    0.325411
dtype: float64
```

```
[ ]: data["b":"c"]
```

```
[ ]: b 1    0.301238
     3    0.325411
     c 1    0.785504
       2    0.986494
dtype: float64
```

```
[ ]: data.loc[["b", "d"]]
```

```
[ ]: b 1    0.301238
     3    0.325411
     d 2    0.741712
       3    0.435975
dtype: float64
```

```
[ ]: data.loc[:, 2]
```

```
[ ]: a    0.846483
     c    0.781111
     d    0.594293
dtype: float64
```

```
[ ]: data.unstack()
```

```
[ ]:      1      2      3
a  0.447380  0.846483  0.665585
b  0.420101      NaN  0.554560
c  0.389662  0.781111      NaN
d      NaN  0.594293  0.515823
```

```
[ ]: data.unstack().stack()
```

```
[ ]: a  1    0.447380
      2    0.846483
      3    0.665585
b  1    0.420101
      3    0.554560
c  1    0.389662
      2    0.781111
d  2    0.594293
      3    0.515823
dtype: float64
```

```
[ ]: frame = pd.DataFrame(np.arange(12).reshape((4, 3)),
                           index=[["a", "a", "b", "b"], [1, 2, 1, 2]],
                           columns=[["Ohio", "Ohio", "Colorado",
                                     "Green", "Red", "Green"]])
frame
```

```
[ ]:      Ohio      Colorado
      Green Red      Green
a 1      0      1          2
   2      3      4          5
b 1      6      7          8
   2      9     10         11
```

```
[ ]: frame.index.names = ["key1", "key2"]
frame.columns.names = ["state", "color"]
frame
```

```
[ ]: state      Ohio      Colorado
color      Green Red      Green
key1 key2
a      1          0      1          2
      2          3      4          5
b      1          6      7          8
      2          9     10         11
```

```
[ ]: frame.index.nlevels
```

```
[ ]: 2
```

```
[ ]: frame["Ohio"]
```

```
[ ]: color      Green  Red
      key1 key2
a      1         0    1
      2         3    4
b      1         6    7
      2         9   10
```

```
[ ]: pd.MultiIndex.from_arrays([
      ["Ohio", "Ohio", "Colorado"],
      ["Green", "Red", "Green"]
    ],
    names=["state", "color"]
  )
```

```
[ ]: MultiIndex([(    'Ohio', 'Green'),
                  (    'Ohio', 'Red'),
                  ('Colorado', 'Green')],
                names=['state', 'color'])
```

```
[ ]: frame.swaplevel("key1", "key2")
```

```
[ ]: state      Ohio      Colorado
      color      Green Red      Green
      key2 key1
1      a         0    1         2
2      a         3    4         5
1      b         6    7         8
2      b         9   10        11
```

```
[ ]: frame.sort_index(level=1)
```

```
[ ]: state      Ohio      Colorado
      color      Green Red      Green
      key1 key2
a      1         0    1         2
b      1         6    7         8
a      2         3    4         5
b      2         9   10        11
```

```
[ ]: frame.swaplevel(0, 1).sort_index(level=0)
```

```
[ ]: state      Ohio      Colorado
      color      Green Red      Green
```

	key2	key1		
1	a	0	1	2
	b	6	7	8
2	a	3	4	5
	b	9	10	11

```
[ ]: frame.groupby(level="key2").sum()
```

```
[ ]: state  Ohio      Colorado
color Green Red      Green
key2
1          6  8          10
2         12 14          16
```

```
[ ]: frame.groupby(level="color", axis="columns").sum()
```

```
/tmp/ipykernel_40681/775557097.py:1: FutureWarning: DataFrame.groupby with
axis=1 is deprecated. Do `frame.T.groupby(...)` without axis instead.
  frame.groupby(level="color", axis="columns").sum()
```

```
[ ]: color      Green  Red
key1 key2
a     1          2    1
      2          8    4
b     1         14    7
      2         20   10
```

```
[ ]: frame = pd.DataFrame({"a": range(7), "b": range(7, 0, -1),
                           "c": ["one", "one", "one", "two", "two",
                                  "two", "two"],
                           "d": [0, 1, 2, 0, 1, 2, 3]})

frame
```

```
[ ]:   a  b  c  d
0  0  7 one  0
1  1  6 one  1
2  2  5 one  2
3  3  4 two  0
4  4  3 two  1
5  5  2 two  2
6  6  1 two  3
```

```
[ ]: frame2 = frame.set_index(["c", "d"])
frame2
```

```
[ ]:   a  b
c  d
```

```

one 0 0 7
    1 1 6
    2 2 5
two 0 3 4
    1 4 3
    2 5 2
    3 6 1

```

```
[ ]: frame.set_index(["c", "d"], drop=False)
```

```

[ ]:      a  b    c  d
c  d
one 0 0 7 one 0
    1 1 6 one 1
    2 2 5 one 2
two 0 3 4 two 0
    1 4 3 two 1
    2 5 2 two 2
    3 6 1 two 3

```

```
[ ]: frame2.reset_index()
```

```

[ ]:      c  d  a  b
0 one 0 0 7
1 one 1 1 6
2 one 2 2 5
3 two 0 3 4
4 two 1 4 3
5 two 2 5 2
6 two 3 6 1

```

```

[ ]: df1 = pd.DataFrame({"key": ["b", "b", "a", "c", "a", "a", "b"],
                        "data1": pd.Series(range(7), dtype="Int64")})
df2 = pd.DataFrame({"key": ["a", "b", "d"],
                    "data2": pd.Series(range(3), dtype="Int64")})
df1

```

```

[ ]:   key  data1
0    b      0
1    b      1
2    a      2
3    c      3
4    a      4
5    a      5
6    b      6

```

```
[ ]: df2
```

```
[ ]:   key  data2
      0    a      0
      1    b      1
      2    d      2
```

```
[ ]: pd.merge(df1, df2)
```

```
[ ]:   key  data1  data2
      0    b      0      1
      1    b      1      1
      2    a      2      0
      3    a      4      0
      4    a      5      0
      5    b      6      1
```

```
[ ]: pd.merge(df1, df2, on="key")
```

```
[ ]:   key  data1  data2
      0    b      0      1
      1    b      1      1
      2    a      2      0
      3    a      4      0
      4    a      5      0
      5    b      6      1
```

```
[ ]: df3 = pd.DataFrame({"lkey": ["b", "b", "a", "c", "a", "a", "b"],
                        "data1": pd.Series(range(7), dtype="Int64")})
df3
```

```
[ ]:   lkey  data1
      0    b      0
      1    b      1
      2    a      2
      3    c      3
      4    a      4
      5    a      5
      6    b      6
```

```
[ ]: df4 = pd.DataFrame({"rkey": ["a", "b", "d"],
                        "data2": pd.Series(range(3), dtype="Int64")})
df4
```

```
[ ]:   rkey  data2
      0    a      0
      1    b      1
      2    d      2
```

```
[ ]: pd.merge(df3, df4, left_on="lkey", right_on="rkey")
```

```
[ ]:   lkey  data1 rkey  data2
0    b      0    b      1
1    b      1    b      1
2    a      2    a      0
3    a      4    a      0
4    a      5    a      0
5    b      6    b      1
```

```
[ ]: pd.merge(df1, df2, how="outer")
```

```
[ ]:   key  data1  data2
0    a      2      0
1    a      4      0
2    a      5      0
3    b      0      1
4    b      1      1
5    b      6      1
6    c      3  <NA>
7    d  <NA>      2
```

```
[ ]: pd.merge(df3, df4, left_on="lkey", right_on="rkey", how="outer")
```

```
[ ]:   lkey  data1 rkey  data2
0    a      2    a      0
1    a      4    a      0
2    a      5    a      0
3    b      0    b      1
4    b      1    b      1
5    b      6    b      1
6    c      3  NaN  <NA>
7  NaN  <NA>    d      2
```

```
[ ]: df1 = pd.DataFrame({"key": ["b", "b", "a", "c", "a", "b"],
                        "data1": pd.Series(range(6), dtype="Int64")})
df1
```

```
[ ]:   key  data1
0    b      0
1    b      1
2    a      2
3    c      3
4    a      4
5    b      5
```



```
[ ]: df2 = pd.DataFrame({"key": ["a", "b", "a", "b", "d"],
                        "data2": pd.Series(range(5), dtype="Int64")})
df2
```

```
[ ]:   key  data2
0    a      0
1    b      1
2    a      2
3    b      3
4    d      4
```

```
[ ]: pd.merge(df1, df2, on="key", how="left")
```

```
[ ]:   key  data1  data2
0    b      0      1
1    b      0      3
2    b      1      1
3    b      1      3
4    a      2      0
5    a      2      2
6    c      3  <NA>
7    a      4      0
8    a      4      2
9    b      5      1
10   b      5      3
```

```
[ ]: pd.merge(df1, df2, how="inner")
```

```
[ ]:   key  data1  data2
0    b      0      1
1    b      0      3
2    b      1      1
3    b      1      3
4    a      2      0
5    a      2      2
6    a      4      0
7    a      4      2
8    b      5      1
9    b      5      3
```

```
[ ]: left = pd.DataFrame({"key1": ["foo", "foo", "bar"],
                          "key2": ["one", "two", "one"],
                          "lval": pd.Series([1, 2, 3], dtype='Int64')})
left
```

```
[ ]:   key1 key2  lval
0   foo  one     1
```

```
1  foo  two    2
2  bar  one    3
```

```
[ ]: right = pd.DataFrame({"key1": ["foo", "foo", "bar", "bar"],
                           "key2": ["one", "one", "one", "two"],
                           "rval": pd.Series([4, 5, 6, 7], dtype='Int64')})
right
```

```
[ ]:   key1 key2  rval
0  foo  one    4
1  foo  one    5
2  bar  one    6
3  bar  two    7
```

```
[ ]: pd.merge(left, right, on=["key1", "key2"], how="outer")
```

```
[ ]:   key1 key2  lval  rval
0  bar  one    3    6
1  bar  two  <NA>    7
2  foo  one    1    4
3  foo  one    1    5
4  foo  two    2  <NA>
```

```
[ ]: pd.merge(left, right, on="key1")
```

```
[ ]:   key1 key2_x  lval key2_y  rval
0  foo    one    1    one    4
1  foo    one    1    one    5
2  foo    two    2    one    4
3  foo    two    2    one    5
4  bar    one    3    one    6
5  bar    one    3    two    7
```

```
[ ]: pd.merge(left, right, on="key1", suffixes=("_left", "_right"))
```

```
[ ]:   key1 key2_left  lval key2_right  rval
0  foo    one    1    one    4
1  foo    one    1    one    5
2  foo    two    2    one    4
3  foo    two    2    one    5
4  bar    one    3    one    6
5  bar    one    3    two    7
```

```
[ ]: left1 = pd.DataFrame({"key": ["a", "b", "a", "a", "b", "c"],
                           "value": pd.Series(range(6), dtype="Int64")})
left1
```

```
[ ]:  key  value
      0    a      0
      1    b      1
      2    a      2
      3    a      3
      4    b      4
      5    c      5
```

```
[ ]: right1 = pd.DataFrame({"group_val": [3.5, 7]}, index=["a", "b"])
      right1
```

```
[ ]:  group_val
a      3.5
b      7.0
```

```
[ ]: pd.merge(left1, right1, left_on="key", right_index=True)
```

```
[ ]:  key  value  group_val
      0    a      0        3.5
      1    b      1        7.0
      2    a      2        3.5
      3    a      3        3.5
      4    b      4        7.0
```

```
[ ]: pd.merge(left1, right1, left_on="key", right_index=True, how="outer")
```

```
[ ]:  key  value  group_val
      0    a      0        3.5
      2    a      2        3.5
      3    a      3        3.5
      1    b      1        7.0
      4    b      4        7.0
      5    c      5         NaN
```

```
[ ]: lefth = pd.DataFrame({"key1": ["Ohio", "Ohio", "Ohio",
                                   "Nevada", "Nevada"],
                           "key2": [2000, 2001, 2002, 2001, 2002],
                           "data": pd.Series(range(5), dtype="Int64")})
      lefth
```

```
[ ]:  key1  key2  data
      0  Ohio  2000     0
      1  Ohio  2001     1
      2  Ohio  2002     2
      3 Nevada  2001     3
      4 Nevada  2002     4
```

```
[ ]: righth_index = pd.MultiIndex.from_arrays(
    [
        ["Nevada", "Nevada", "Ohio", "Ohio", "Ohio", "Ohio"],
        [2001, 2000, 2000, 2000, 2001, 2002]
    ]
)
righth_index
```

```
[ ]: MultiIndex([('Nevada', 2001),
                ('Nevada', 2000),
                ( 'Ohio', 2000),
                ( 'Ohio', 2000),
                ( 'Ohio', 2001),
                ( 'Ohio', 2002)],
                )
```

```
[ ]: righth = pd.DataFrame({"event1": pd.Series([0, 2, 4, 6, 8, 10], dtype="Int64",
                                                index=righth_index),
                           "event2": pd.Series([1, 3, 5, 7, 9, 11], dtype="Int64",
                                                index=righth_index)})
righth
```

```
[ ]:
      event1  event2
Nevada 2001      0      1
      2000      2      3
Ohio    2000      4      5
      2000      6      7
      2001      8      9
      2002     10     11
```

```
[ ]: pd.merge(lefth, righth, left_on=["key1", "key2"], right_index=True)
```

```
[ ]:
      key1  key2  data  event1  event2
0   Ohio  2000      0        4        5
0   Ohio  2000      0        6        7
1   Ohio  2001      1        8        9
2   Ohio  2002      2       10       11
3  Nevada  2001      3         0         1
```

```
[ ]: pd.merge(lefth, righth, left_on=["key1", "key2"],
              right_index=True, how="outer")
```

```
[ ]:
      key1  key2  data  event1  event2
4  Nevada  2000  <NA>        2         3
3  Nevada  2001      3         0         1
4  Nevada  2002      4      <NA>      <NA>
0   Ohio  2000      0         4         5
```

0	Ohio	2000	0	6	7
1	Ohio	2001	1	8	9
2	Ohio	2002	2	10	11

```
[ ]: left2 = pd.DataFrame([[1., 2.], [3., 4.], [5., 6.]],
                          index=["a", "c", "e"],
                          columns=["Ohio", "Nevada"]).astype("Int64")
left2
```

```
[ ]:      Ohio  Nevada
a      1      2
c      3      4
e      5      6
```

```
[ ]: right2 = pd.DataFrame([[7., 8.], [9., 10.], [11., 12.], [13., 14.]],
                           index=["b", "c", "d", "e"],
                           columns=["Missouri", "Alabama"]).astype("Int64")
right2
```

```
[ ]:      Missouri  Alabama
b           7      8
c           9     10
d          11     12
e          13     14
```

```
[ ]: pd.merge(left2, right2, how="outer", left_index=True, right_index=True)
```

```
[ ]:      Ohio  Nevada  Missouri  Alabama
a      1      2      <NA>      <NA>
b  <NA>  <NA>      7      8
c      3      4      9     10
d  <NA>  <NA>     11     12
e      5      6     13     14
```

```
[ ]: left2.join(right2, how="outer")
```

```
[ ]:      Ohio  Nevada  Missouri  Alabama
a      1      2      <NA>      <NA>
b  <NA>  <NA>      7      8
c      3      4      9     10
d  <NA>  <NA>     11     12
e      5      6     13     14
```

```
[ ]: left1.join(right1, on="key")
```

```
[ ]:      key  value  group_val
0    a      0      3.5
```

1	b	1	7.0
2	a	2	3.5
3	a	3	3.5
4	b	4	7.0
5	c	5	NaN

```
[ ]: another = pd.DataFrame([[7., 8.], [9., 10.], [11., 12.], [16., 17.]],
                             index=["a", "c", "e", "f"],
                             columns=["New York", "Oregon"])
another
```

```
[ ]:      New York  Oregon
a         7.0     8.0
c         9.0    10.0
e        11.0    12.0
f        16.0    17.0
```

```
[ ]: left2.join([right2, another])
```

```
[ ]:      Ohio  Nevada  Missouri  Alabama  New York  Oregon
a      1       2      <NA>     <NA>      7.0     8.0
c      3       4       9      10      9.0    10.0
e      5       6      13      14     11.0    12.0
```

```
[ ]: left2.join([right2, another], how="outer")
```

```
[ ]:      Ohio  Nevada  Missouri  Alabama  New York  Oregon
a      1       2      <NA>     <NA>      7.0     8.0
c      3       4       9      10      9.0    10.0
e      5       6      13      14     11.0    12.0
b  <NA>    <NA>       7       8      NaN     NaN
d  <NA>    <NA>      11      12      NaN     NaN
f  <NA>    <NA>    <NA>    <NA>    16.0    17.0
```

```
[ ]: arr = np.arange(12).reshape((3, 4))
arr
```

```
[ ]: array([[ 0,  1,  2,  3],
           [ 4,  5,  6,  7],
           [ 8,  9, 10, 11]])
```

```
[ ]: np.concatenate([arr, arr], axis=1)
```

```
[ ]: array([[ 0,  1,  2,  3,  0,  1,  2,  3],
           [ 4,  5,  6,  7,  4,  5,  6,  7],
           [ 8,  9, 10, 11,  8,  9, 10, 11]])
```

```
[ ]: s1 = pd.Series([0, 1], index=["a", "b"], dtype="Int64")
s1
```

```
[ ]: a    0
     b    1
     dtype: Int64
```

```
[ ]: s2 = pd.Series([2, 3, 4], index=["c", "d", "e"], dtype="Int64")
s2
```

```
[ ]: c    2
     d    3
     e    4
     dtype: Int64
```

```
[ ]: s3 = pd.Series([5, 6], index=["f", "g"], dtype="Int64")
s3
```

```
[ ]: f    5
     g    6
     dtype: Int64
```

```
[ ]: pd.concat([s1, s2, s3])
```

```
[ ]: a    0
     b    1
     c    2
     d    3
     e    4
     f    5
     g    6
     dtype: Int64
```

```
[ ]: pd.concat([s1, s2, s3], axis="columns")
```

```
[ ]:      0    1    2
     a    0 <NA> <NA>
     b    1 <NA> <NA>
     c <NA>    2 <NA>
     d <NA>    3 <NA>
     e <NA>    4 <NA>
     f <NA> <NA>    5
     g <NA> <NA>    6
```

```
[ ]: s4 = pd.concat([s1, s3])
s4
```

```
[ ]: a    0
      b    1
      f    5
      g    6
      dtype: Int64
```

```
[ ]: pd.concat([s1, s4], axis="columns")
```

```
[ ]:      0  1
      a    0  0
      b    1  1
      f <NA> 5
      g <NA> 6
```

```
[ ]: pd.concat([s1, s4], axis="columns", join="inner")
```

```
[ ]:      0  1
      a    0  0
      b    1  1
```

```
[ ]: result = pd.concat([s1, s1, s3], keys=["one", "two", "three"])
      result
```

```
[ ]: one    a    0
      b    1
      two   a    0
      b    1
      three f    5
      g    6
      dtype: Int64
```

```
[ ]: result.unstack()
```

```
[ ]:      a    b    f    g
      one    0    1 <NA> <NA>
      two    0    1 <NA> <NA>
      three <NA> <NA>    5    6
```

```
[ ]: pd.concat([s1, s2, s3], axis="columns", keys=["one", "two", "three"])
```

```
[ ]:      one  two  three
      a    0 <NA> <NA>
      b    1 <NA> <NA>
      c <NA>    2 <NA>
      d <NA>    3 <NA>
      e <NA>    4 <NA>
      f <NA> <NA>    5
```



```
g <NA> <NA>      6
```

```
[ ]: df1 = pd.DataFrame(np.arange(6).reshape(3, 2), index=["a", "b", "c"],
                        columns=["one", "two"])
df1
```

```
[ ]:      one  two
a      0    1
b      2    3
c      4    5
```

```
[ ]: df2 = pd.DataFrame(5 + np.arange(4).reshape(2, 2), index=["a", "c"],
                        columns=["three", "four"])
df2
```

```
[ ]:      three  four
a         5     6
c         7     8
```

```
[ ]: pd.concat([df1, df2], axis="columns", keys=["level1", "level2"])
```

```
[ ]:      level1      level2
      one two  three four
a      0  1    5.0  6.0
b      2  3    NaN  NaN
c      4  5    7.0  8.0
```

```
[ ]: pd.concat({"level1": df1, "level2": df2}, axis="columns")
```

```
[ ]:      level1      level2
      one two  three four
a      0  1    5.0  6.0
b      2  3    NaN  NaN
c      4  5    7.0  8.0
```

```
[ ]: pd.concat([df1, df2], axis="columns", keys=["level1", "level2"],
                names=["upper", "lower"])
```

```
[ ]: upper level1      level2
lower      one two  three four
a          0  1    5.0  6.0
b          2  3    NaN  NaN
c          4  5    7.0  8.0
```

```
[ ]: df1 = pd.DataFrame(np.random.standard_normal((3, 4)),
                        columns=["a", "b", "c", "d"])
df1
```

```
[ ]:      a      b      c      d
0 -0.565837 -1.007665 -0.007756  1.232125
1 -0.852007  0.111540 -0.435110 -0.754659
2  1.023165 -0.499075  0.283348  0.851128
```

```
[ ]: df2 = pd.DataFrame(np.random.standard_normal((2, 3)),
                        columns=["b", "d", "a"])
df2
```

```
[ ]:      b      d      a
0 -1.411379  1.071315 -0.275694
1  0.802649 -0.180412  1.057268
```

```
[ ]: pd.concat([df1, df2], ignore_index=True)
```

```
[ ]:      a      b      c      d
0 -0.565837 -1.007665 -0.007756  1.232125
1 -0.852007  0.111540 -0.435110 -0.754659
2  1.023165 -0.499075  0.283348  0.851128
3 -0.275694 -1.411379      NaN  1.071315
4  1.057268  0.802649      NaN -0.180412
```

```
[ ]: a = pd.Series([np.nan, 2.5, 0.0, 3.5, 4.5, np.nan],
                  index=["f", "e", "d", "c", "b", "a"])
a
```

```
[ ]: f    NaN
e    2.5
d    0.0
c    3.5
b    4.5
a    NaN
dtype: float64
```

```
[ ]: b = pd.Series([0., np.nan, 2., np.nan, np.nan, 5.],
                  index=["a", "b", "c", "d", "e", "f"])
b
```

```
[ ]: a    0.0
b    NaN
c    2.0
d    NaN
e    NaN
f    5.0
dtype: float64
```

```
[ ]: np.where(pd.isna(a), b, a)
```

```
[ ]: array([0. , 2.5, 0. , 3.5, 4.5, 5. ])
```

```
[ ]: a.combine_first(b)
```

```
[ ]: a    0.0  
     b    4.5  
     c    3.5  
     d    0.0  
     e    2.5  
     f    5.0  
     dtype: float64
```

```
[ ]: df1 = pd.DataFrame({"a": [1., np.nan, 5., np.nan],  
                        "b": [np.nan, 2., np.nan, 6.],  
                        "c": range(2, 18, 4)})  
df1
```

```
[ ]:      a    b    c  
0  1.0  NaN    2  
1  NaN  2.0    6  
2  5.0  NaN   10  
3  NaN  6.0   14
```

```
[ ]: df2 = pd.DataFrame({"a": [5., 4., np.nan, 3., 7.],  
                        "b": [np.nan, 3., 4., 6., 8.]})  
df2
```

```
[ ]:      a    b  
0  5.0  NaN  
1  4.0  3.0  
2  NaN  4.0  
3  3.0  6.0  
4  7.0  8.0
```

```
[ ]: df1.combine_first(df2)
```

```
[ ]:      a    b    c  
0  1.0  NaN    2  
1  4.0  2.0    6  
2  5.0  4.0   10  
3  3.0  6.0   14  
4  7.0  8.0   NaN
```

```
[ ]: data = pd.DataFrame(np.arange(6).reshape((2, 3)),  
                        index=pd.Index(["Ohio", "Colorado"], name="state"),  
                        columns=pd.Index(["one", "two", "three"],  
                                         name="number"))
```

```
data
```

```
[ ]: number    one  two  three
     state
Ohio         0    1    2
Colorado     3    4    5
```

```
[ ]: result = data.stack()
     result
```

```
[ ]: state      number
Ohio          one      0
           two      1
           three     2
Colorado     one      3
           two      4
           three     5
dtype: int64
```

```
[ ]: result.unstack()
```

```
[ ]: number    one  two  three
     state
Ohio         0    1    2
Colorado     3    4    5
```

```
[ ]: result.unstack(level=0)
```

```
[ ]: state  Ohio  Colorado
     number
one        0        3
two        1        4
three      2        5
```

```
[ ]: result.unstack(level="state")
```

```
[ ]: state  Ohio  Colorado
     number
one        0        3
two        1        4
three      2        5
```

```
[ ]: s1 = pd.Series([0, 1, 2, 3], index=["a", "b", "c", "d"], dtype="Int64")
     s1
```

```
[ ]: a    0
     b    1
```

```
c    2
d    3
dtype: Int64
```

```
[ ]: s2 = pd.Series([4, 5, 6], index=["c", "d", "e"], dtype="Int64")
s2
```

```
[ ]: c    4
d    5
e    6
dtype: Int64
```

```
[ ]: data2 = pd.concat([s1, s2], keys=["one", "two"])
data2
```

```
[ ]: one  a    0
      b    1
      c    2
      d    3
two    c    4
      d    5
      e    6
dtype: Int64
```

```
[ ]: data2.unstack()
```

```
[ ]:      a    b  c  d    e
one    0    1  2  3  <NA>
two  <NA> <NA> 4  5    6
```

```
[ ]: data2.unstack().stack()
```

```
[ ]: one  a    0
      b    1
      c    2
      d    3
two    c    4
      d    5
      e    6
dtype: Int64
```

```
[ ]: data2.unstack().stack(dropna=False)
```

/tmp/ipykernel_40681/3936770077.py:1: FutureWarning: The previous implementation of stack is deprecated and will be removed in a future version of pandas. See the What's New notes for pandas 2.1.0 for details. Specify future_stack=True to adopt the new implementation and silence this warning.

```
data2.unstack().stack(dropna=False)
```

```
[ ]: one  a      0
      b      1
      c      2
      d      3
      e    <NA>
two   a    <NA>
      b    <NA>
      c      4
      d      5
      e      6
dtype: Int64
```

```
[ ]: df = pd.DataFrame({"left": result, "right": result + 5},
                        columns=pd.Index(["left", "right"], name="side"))
df
```

```
[ ]: side      left  right
state  number
Ohio   one       0      5
      two       1      6
      three     2      7
Colorado one     3      8
      two     4      9
      three    5     10
```

```
[ ]: df.unstack(level="state")
```

```
[ ]: side  left      right
state Ohio Colorado Ohio Colorado
number
one      0        3      5        8
two      1        4      6        9
three    2        5      7       10
```

```
[ ]: df.unstack(level="state").stack(level="side")
```

/tmp/ipykernel_40681/2617337668.py:1: FutureWarning: The previous implementation of stack is deprecated and will be removed in a future version of pandas. See the What's New notes for pandas 2.1.0 for details. Specify future_stack=True to adopt the new implementation and silence this warning.

```
df.unstack(level="state").stack(level="side")
```

```
[ ]: state      Ohio  Colorado
number side
one   left      0          3
      right     5          8
```

two	left	1	4
	right	6	9
three	left	2	5
	right	7	10

```
[ ]: data = pd.read_csv(base_url + "examples/macrodata.csv")
data = data.loc[:, ["year", "quarter", "realgdp", "infl", "unemp"]]
data.head()
```

```
[ ]:   year  quarter  realgdp  infl  unemp
0  1959         1  2710.349  0.00    5.8
1  1959         2  2778.801  2.34    5.1
2  1959         3  2775.488  2.74    5.3
3  1959         4  2785.204  0.27    5.6
4  1960         1  2847.699  2.31    5.2
```

```
[ ]: periods = pd.PeriodIndex(year=data.pop("year"),
                             quarter=data.pop("quarter"),
                             name="date")
periods
```

/tmp/ipykernel_40681/1359989538.py:1: FutureWarning: Constructing PeriodIndex from fields is deprecated. Use PeriodIndex.from_fields instead.

```
periods = pd.PeriodIndex(year=data.pop("year"),
```

```
[ ]: PeriodIndex(['1959Q1', '1959Q2', '1959Q3', '1959Q4', '1960Q1', '1960Q2',
                '1960Q3', '1960Q4', '1961Q1', '1961Q2',
                ...
                '2007Q2', '2007Q3', '2007Q4', '2008Q1', '2008Q2', '2008Q3',
                '2008Q4', '2009Q1', '2009Q2', '2009Q3'],
                dtype='period[Q-DEC]', name='date', length=203)
```

```
[ ]: data.index = periods.to_timestamp("D")
data.head()
```

```
[ ]:   realgdp  infl  unemp
date
1959-01-01  2710.349  0.00    5.8
1959-04-01  2778.801  2.34    5.1
1959-07-01  2775.488  2.74    5.3
1959-10-01  2785.204  0.27    5.6
1960-01-01  2847.699  2.31    5.2
```

```
[ ]: data = data.reindex(columns=["realgdp", "infl", "unemp"])
data.columns.name = "item"
data.head()
```

```
[ ]: item      realgdp  infl  unemp
     date
1959-01-01  2710.349  0.00   5.8
1959-04-01  2778.801  2.34   5.1
1959-07-01  2775.488  2.74   5.3
1959-10-01  2785.204  0.27   5.6
1960-01-01  2847.699  2.31   5.2
```

```
[ ]: long_data = (data.stack()
                  .reset_index()
                  .rename(columns={0: "value"}))
```

```
[ ]: long_data[:10]
```

```
[ ]:
     date      item      value
0 1959-01-01  realgdp  2710.349
1 1959-01-01    infl    0.000
2 1959-01-01   unemp    5.800
3 1959-04-01  realgdp  2778.801
4 1959-04-01    infl    2.340
5 1959-04-01   unemp    5.100
6 1959-07-01  realgdp  2775.488
7 1959-07-01    infl    2.740
8 1959-07-01   unemp    5.300
9 1959-10-01  realgdp  2785.204
```

```
[ ]: pivoted = long_data.pivot(index="date", columns="item",
                               values="value")
     pivoted.head()
```

```
[ ]: item      infl  realgdp  unemp
     date
1959-01-01  0.00  2710.349   5.8
1959-04-01  2.34  2778.801   5.1
1959-07-01  2.74  2775.488   5.3
1959-10-01  0.27  2785.204   5.6
1960-01-01  2.31  2847.699   5.2
```

```
[ ]: long_data.index.name = None
```

```
[ ]: long_data["value2"] = np.random.standard_normal(len(long_data))
     long_data[:10]
```

```
[ ]:
     date      item      value  value2
0 1959-01-01  realgdp  2710.349 -1.321280
1 1959-01-01    infl    0.000  0.885264
2 1959-01-01   unemp    5.800 -0.655461
```



```

3 1959-04-01  realgdp  2778.801  0.856157
4 1959-04-01    infl      2.340  0.843170
5 1959-04-01   unemp      5.100 -0.950790
6 1959-07-01  realgdp  2775.488  0.414294
7 1959-07-01    infl      2.740 -0.329745
8 1959-07-01   unemp      5.300  0.233431
9 1959-10-01  realgdp  2785.204  0.273288

```

```
[ ]: pivoted = long_data.pivot(index="date", columns="item")
pivoted.head()
```

```
[ ]:
      value                value2
item      infl  realgdp unemp      infl  realgdp    unemp
date
1959-01-01  0.00  2710.349   5.8  0.885264 -1.321280 -0.655461
1959-04-01  2.34  2778.801   5.1  0.843170  0.856157 -0.950790
1959-07-01  2.74  2775.488   5.3 -0.329745  0.414294  0.233431
1959-10-01  0.27  2785.204   5.6  0.208112  0.273288 -0.225871
1960-01-01  2.31  2847.699   5.2  0.472371 -1.525169  0.669444

```

```
[ ]: pivoted["value"].head()
```

```
[ ]: item      infl  realgdp  unemp
date
1959-01-01  0.00  2710.349    5.8
1959-04-01  2.34  2778.801    5.1
1959-07-01  2.74  2775.488    5.3
1959-10-01  0.27  2785.204    5.6
1960-01-01  2.31  2847.699    5.2

```

```
[ ]: unstacked = long_data.set_index(["date", "item"]).unstack(level="item")
unstacked.head()
```

```
[ ]:
      value                value2
item      infl  realgdp unemp      infl  realgdp    unemp
date
1959-01-01  0.00  2710.349   5.8  0.885264 -1.321280 -0.655461
1959-04-01  2.34  2778.801   5.1  0.843170  0.856157 -0.950790
1959-07-01  2.74  2775.488   5.3 -0.329745  0.414294  0.233431
1959-10-01  0.27  2785.204   5.6  0.208112  0.273288 -0.225871
1960-01-01  2.31  2847.699   5.2  0.472371 -1.525169  0.669444

```

```
[ ]: df = pd.DataFrame({"key": ["foo", "bar", "baz"],
                        "A": [1, 2, 3],
                        "B": [4, 5, 6],
                        "C": [7, 8, 9]})
df
```

```
[ ]:   key  A  B  C
      0  foo  1  4  7
      1  bar  2  5  8
      2  baz  3  6  9
```

```
[ ]: melted = pd.melt(df, id_vars="key")
      melted
```

```
[ ]:   key variable  value
      0  foo         A      1
      1  bar         A      2
      2  baz         A      3
      3  foo         B      4
      4  bar         B      5
      5  baz         B      6
      6  foo         C      7
      7  bar         C      8
      8  baz         C      9
```

```
[ ]: reshaped = melted.pivot(index="key", columns="variable",
                              values="value")
      reshaped
```

```
[ ]: variable  A  B  C
      key
      bar      2  5  8
      baz      3  6  9
      foo      1  4  7
```

```
[ ]: reshaped.reset_index()
```

```
[ ]: variable  key  A  B  C
      0         bar  2  5  8
      1         baz  3  6  9
      2         foo  1  4  7
```

```
[ ]: pd.melt(df, id_vars="key", value_vars=["A", "B"])
```

```
[ ]:   key variable  value
      0  foo         A      1
      1  bar         A      2
      2  baz         A      3
      3  foo         B      4
      4  bar         B      5
      5  baz         B      6
```

```
[ ]: pd.melt(df, value_vars=["A", "B", "C"])
```

```
[ ]:   variable  value
      0         A      1
      1         A      2
      2         A      3
      3         B      4
      4         B      5
      5         B      6
      6         C      7
      7         C      8
      8         C      9
```

```
[ ]: pd.melt(df, value_vars=["key", "A", "B"])
```

```
[ ]:   variable value
      0      key  foo
      1      key  bar
      2      key  baz
      3         A    1
      4         A    2
      5         A    3
      6         B    4
      7         B    5
      8         B    6
```