



Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Pandas

ETSEIB/GIE

31 de maig de 2018



Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

① groupby

② merge

③ Aplicació de funcions al *DataFrame*



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({'c1': [1,2,2,2,3,3,4,4,4,4,4,4,
4,5,5],
...                    'c2': [5,6,7,8,6,6,7,7,7,7,7,7,7,
,7,8],
...                    'c3': [9,8,7,6,9,1,2,3,4,6,7,8,9,10,1]})
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({'c1': [1,2,2,2,3,3,4,4,4,4,4,4,
4,5,5],
...                    'c2': [5,6,7,8,6,6,7,7,7,7,7,7,7,
,7,8],
...                    'c3': [9,8,7,6,9,1,2,3,4,6,7,8,9,10,1]})
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({'c1': [1,2,2,2,3,3,4,4,4,4,4,4,
4,5,5],
...                    'c2': [5,6,7,8,6,6,7,7,7,7,7,7,7,
7,8],
...                    'c3': [9,8,7,6,9,1,2,3,4,6,7,8,9,10,1]})
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({'c1': [1,2,2,2,3,3,4,4,4,4,4,4,
4,5,5],
...                    'c2': [5,6,7,8,6,6,7,7,7,7,7,7,7,
7,8],
...                    'c3': [9,8,7,6,9,1,2,3,4,6,7,8,9,10,1]})
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({'c1': [1,2,2,2,3,3,4,4,4,4,4,4,
4,5,5],
...                    'c2': [5,6,7,8,6,6,7,7,7,7,7,7,7,
7,8],
...                    'c3': [9,8,7,6,9,1,2,3,4,6,7,8,9,10,1]})
```

Pandas

groupby

merge

Aplicació de funcions al DataFrame

```
>>> df
   c1  c2  c3
0    1   5   9
1    2   6   8
2    2   7   7
3    2   8   6
4    3   6   9
5    3   6   1
6    4   7   2
7    4   7   3
8    4   7   4
9    4   7   6
10   4   7   7
11   4   7   8
12   4   7   9
13   5   7  10
14   5   8   1
```

Pandas

groupby

merge

Aplicació de funcions al DataFrame

```
>>> df
   c1  c2  c3
0    1   5   9
1    2   6   8
2    2   7   7
3    2   8   6
4    3   6   9
5    3   6   1
6    4   7   2
7    4   7   3
8    4   7   4
9    4   7   6
10   4   7   7
11   4   7   8
12   4   7   9
13   5   7  10
14   5   8   1
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3
0    1   5   9
1    2   6   8
2    2   7   7
3    2   8   6
4    3   6   9
5    3   6   1
6    4   7   2
7    4   7   3
8    4   7   4
9    4   7   6
10   4   7   7
11   4   7   8
12   4   7   9
13   5   7  10
14   5   8   1
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1')
<pandas.core.groupby.DataFrameGroupBy object at ... >
>>> df.groupby('c1').groups
{1: Int64Index([0], dtype='int64'), 2: Int64Index([1, 2, 3], dtype='int64'), 3: Int64Index([4, 5], dtype='int64'), 4: Int64Index([6, 7, 8, 9, 10, 11, 12], dtype='int64'), 5: Int64Index([13, 14], dtype='int64')}
>>> df.groupby('c1').get_group(2)
   c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1')
<pandas.core.groupby.DataFrameGroupBy object at ... >
>>> df.groupby('c1').groups
{1: Int64Index([0], dtype='int64'), 2: Int64Index([1, 2, 3], dtype='int64'), 3: Int64Index([4, 5], dtype='int64'), 4: Int64Index([6, 7, 8, 9, 10, 11, 12], dtype='int64'), 5: Int64Index([13, 14], dtype='int64')}
>>> df.groupby('c1').get_group(2)
   c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1')
<pandas.core.groupby.DataFrameGroupBy object at ... >
>>> df.groupby('c1').groups
{1: Int64Index([0], dtype='int64'), 2: Int64Index([1, 2, 3], dtype='int64'), 3: Int64Index([4, 5], dtype='int64'), 4: Int64Index([6, 7, 8, 9, 10, 11, 12], dtype='int64'), 5: Int64Index([13, 14], dtype='int64')}
>>> df.groupby('c1').get_group(2)
   c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1')
<pandas.core.groupby.DataFrameGroupBy object at ... >
>>> df.groupby('c1').groups
{1: Int64Index([0], dtype='int64'), 2: Int64Index([1,
2, 3], dtype='int64'), 3: Int64Index([4, 5], dtype='i
64'), 4: Int64Index([6, 7, 8, 9, 10, 11, 12], dtype='
t64'), 5: Int64Index([13, 14], dtype='int64')}
>>> df.groupby('c1').get_group(2)
   c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
```

```
>>> df.groupby('c1')
<pandas.core.groupby.DataFrameGroupBy object at ... >
>>> df.groupby('c1').groups
{1: Int64Index([0], dtype='int64'), 2: Int64Index([1, 2, 3], dtype='int64'), 3: Int64Index([4, 5], dtype='int64'), 4: Int64Index([6, 7, 8, 9, 10, 11, 12], dtype='int64'), 5: Int64Index([13, 14], dtype='int64')}
>>> df.groupby('c1').get_group(2)
   c1  c2  c3
1    2    6    8
2    2    7    7
3    2    8    6
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').get_group(4)
   c1  c2  c3
6   4   7   2
7   4   7   3
8   4   7   4
9   4   7   6
10  4   7   7
11  4   7   8
12  4   7   9
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').get_group(4)
   c1  c2  c3
6   4   7   2
7   4   7   3
8   4   7   4
9   4   7   6
10  4   7   7
11  4   7   8
12  4   7   9
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').get_group(4)
   c1  c2  c3
6   4   7   2
7   4   7   3
8   4   7   4
9   4   7   6
10  4   7   7
11  4   7   8
12  4   7   9
```

```
>>> for g, dg in df.groupby('c1'):
...     print('--- Grup: ', g, ' ---\n', dg, sep='')
--- Grup: 1 ---
   c1  c2  c3
0    1   5   9
--- Grup: 2 ---
   c1  c2  c3
1    2   6   8
2    2   7   7
3    2   8   6
--- Grup: 3 ---
   c1  c2  c3
4    3   6   9
5    3   6   1
--- Grup: 4 ---
   c1  c2  c3
6    4   7   2
7    4   7   3
8    4   7   4
9    4   7   6
10   4   7   7
11   4   7   8
12   4   7   9
--- Grup: 5 ---
   c1  c2  c3
13   5   7  10
14   5   8   1
```

```
>>> for g, dg in df.groupby('c1'):
...     print('--- Grup: ', g, ' ---\n', dg, sep='')
--- Grup: 1 ---
  c1  c2  c3
0   1   5   9
--- Grup: 2 ---
  c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
--- Grup: 3 ---
  c1  c2  c3
4   3   6   9
5   3   6   1
--- Grup: 4 ---
  c1  c2  c3
6   4   7   2
7   4   7   3
8   4   7   4
9   4   7   6
10  4   7   7
11  4   7   8
12  4   7   9
--- Grup: 5 ---
  c1  c2  c3
13  5   7  10
14  5   8   1
```

```
>>> for g, dg in df.groupby('c1'):
...     print('--- Grup: ', g, ' ---\n', dg, sep='')
--- Grup: 1 ---
  c1  c2  c3
0   1   5   9
--- Grup: 2 ---
  c1  c2  c3
1   2   6   8
2   2   7   7
3   2   8   6
--- Grup: 3 ---
  c1  c2  c3
4   3   6   9
5   3   6   1
--- Grup: 4 ---
  c1  c2  c3
6   4   7   2
7   4   7   3
8   4   7   4
9   4   7   6
10  4   7   7
11  4   7   8
12  4   7   9
--- Grup: 5 ---
  c1  c2  c3
13  5   7  10
14  5   8   1
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg(np.mean)
      c2      c3
c1
1    5.0  9.000000
2    7.0  7.000000
3    6.0  5.000000
4    7.0  5.571429
5    7.5  5.500000
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg(np.mean)
      c2      c3
c1
1    5.0  9.000000
2    7.0  7.000000
3    6.0  5.000000
4    7.0  5.571429
5    7.5  5.500000
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg(np.mean)
      c2      c3
c1
1    5.0  9.000000
2    7.0  7.000000
3    6.0  5.000000
4    7.0  5.571429
5    7.5  5.500000
```



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg([np.sum, np.mean])
```

	c2		c3	
	sum	mean	sum	mean
c1				
1	5	5.0	9	9.000000
2	21	7.0	21	7.000000
3	12	6.0	10	5.000000
4	49	7.0	39	5.571429
5	15	7.5	11	5.500000



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg([np.sum, np.mean])
```

	c2		c3	
	sum	mean	sum	mean
c1				
1	5	5.0	9	9.000000
2	21	7.0	21	7.000000
3	12	6.0	10	5.000000
4	49	7.0	39	5.571429
5	15	7.5	11	5.500000



groupby

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.groupby('c1').agg([np.sum, np.mean])
```

	c2		c3	
	sum	mean	sum	mean
c1				
1	5	5.0	9	9.000000
2	21	7.0	21	7.000000
3	12	6.0	10	5.000000
4	49	7.0	39	5.571429
5	15	7.5	11	5.500000



merge. Dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> mecànica = pd.DataFrame({'DNI': ['1A', '2B', '3C']
...                           'Nota': [3.0, 5.0, 7.0]})
>>> informàtica = pd.DataFrame({'DNI': ['1A', '2B', '4
D'],
...                              'Nota': [5.0, 8.0, 9.0]})
```



merge. Dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> mecànica = pd.DataFrame({'DNI': ['1A', '2B', '3C']
...                           'Nota': [3.0, 5.0, 7.0]})
>>> informàtica = pd.DataFrame({'DNI': ['1A', '2B', '4
D'],
...                              'Nota': [5.0, 8.0, 9.0]})
```



merge. Dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> mecànica = pd.DataFrame({'DNI': ['1A', '2B', '3C']
...                          'Nota': [3.0, 5.0, 7.0]})
>>> informàtica = pd.DataFrame({'DNI': ['1A', '2B', '4
D'],
...                              'Nota': [5.0, 8.0, 9.0]})
```



merge. Dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> mecànica = pd.DataFrame({'DNI': ['1A', '2B', '3C']
,
...                          'Nota': [3.0, 5.0, 7.0]})
>>> informàtica = pd.DataFrame({'DNI': ['1A', '2B', '4
D'],
...                              'Nota': [5.0, 8.0, 9.0]})
```



merge. Dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> mecànica = pd.DataFrame({'DNI': ['1A', '2B', '3C']
,
...                          'Nota': [3.0, 5.0, 7.0]})
>>> informàtica = pd.DataFrame({'DNI': ['1A', '2B', '4
D'],
...                              'Nota': [5.0, 8.0, 9.0]})
```



dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> mecànica
```

```
   DNI  Nota
0  1A    3.0
1  2B    5.0
2  3C    7.0
```

```
>>> informàtica
```

```
   DNI  Nota
0  1A    5.0
1  2B    8.0
2  4D    9.0
```



dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> mecànica
```

```
   DNI  Nota
0  1A    3.0
1  2B    5.0
2  3C    7.0
```

```
>>> informàtica
```

```
   DNI  Nota
0  1A    5.0
1  2B    8.0
2  4D    9.0
```



dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> mecànica
```

	DNI	Nota
0	1A	3.0
1	2B	5.0
2	3C	7.0

```
>>> informàtica
```

	DNI	Nota
0	1A	5.0
1	2B	8.0
2	4D	9.0



dades inicials

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> mecànica
   DNI  Nota
0  1A   3.0
1  2B   5.0
2  3C   7.0

>>> informàtica
   DNI  Nota
0  1A   5.0
1  2B   8.0
2  4D   9.0
```



merge. left

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...         informàtica,  
...         how='left',  
...         on='DNI',  
...         suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN



merge. left

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='left',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN



merge. left

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='left',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='right',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	4D	NaN	9.0

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...         informàtica,  
...         how='right',  
...         on='DNI',  
...         suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	4D	NaN	9.0



merge. right

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> pd.merge(mecànica,  
...         informàtica,  
...         how='right',  
...         on='DNI',  
...         suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	4D	NaN	9.0



merge. inner

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Intersecció de claus

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='inner',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0



merge. inner

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Intersecció de claus

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='inner',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0



merge. inner

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Intersecció de claus

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='inner',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0



merge. outer

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Unió de claus

```
>>> pd.merge(mecànica,  
...         informàtica,  
...         how='outer',  
...         on='DNI',  
...         suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN
3	4D	NaN	9.0



merge. outer

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Unió de claus

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='outer',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN
3	4D	NaN	9.0



Unió de claus

```
>>> pd.merge(mecànica,  
...          informàtica,  
...          how='outer',  
...          on='DNI',  
...          suffixes=['_mecànica', '_informàtica'])
```

	DNI	Nota_mecànica	Nota_informàtica
0	1A	3.0	5.0
1	2B	5.0	8.0
2	3C	7.0	NaN
3	4D	NaN	9.0



Aplicació de funcions al *DataFrame*

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

- Aplicació a la taula: *pipe()*
- Aplicació a fila o columna: *apply()*
- Aplicació a element: *applymap()*



pipe()

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> df = pd.DataFrame([[1,2,3,4,5],
...                    [5,6,7,8,9],
...                    [0,1,2,3,4]],
...                   columns=['c1', 'c2', 'c3', 'c4', 'c5'])
>>> def suma(a,b):
...     return a+b
```



pipe()

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> df = pd.DataFrame([[1,2,3,4,5],
...                    [5,6,7,8,9],
...                    [0,1,2,3,4]],
...                   columns=['c1', 'c2', 'c3', 'c4', 'c5'])
>>> def suma(a,b):
...     return a+b
```



pipe()

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> df = pd.DataFrame([[1,2,3,4,5],
...                    [5,6,7,8,9],
...                    [0,1,2,3,4]],
...                   columns=['c1', 'c2', 'c3', 'c4', 'c5'])
>>> def suma(a,b):
...     return a+b
```



pipe()

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> df = pd.DataFrame([[1,2,3,4,5],
...                    [5,6,7,8,9],
...                    [0,1,2,3,4]],
...                   columns=['c1', 'c2', 'c3', 'c4', 'c5'])
>>> def suma(a,b):
...     return a+b
```



pipe()

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> import pandas as pd
>>> df = pd.DataFrame([[1,2,3,4,5],
...                    [5,6,7,8,9],
...                    [0,1,2,3,4]],
...                   columns=['c1', 'c2', 'c3', 'c4', 'c5'])
>>> def suma(a,b):
...     return a+b
```

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.pipe(suma,10)
   c1  c2  c3  c4  c5
0  11  12  13  14  15
1  15  16  17  18  19
2  10  11  12  13  14

>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4
```

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.pipe(suma, 10)
   c1  c2  c3  c4  c5
0  11  12  13  14  15
1  15  16  17  18  19
2  10  11  12  13  14

>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4
```



Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df.pipe(suma,10)
```

	c1	c2	c3	c4	c5
0	11	12	13	14	15
1	15	16	17	18	19
2	10	11	12	13	14

```
>>> df
```

	c1	c2	c3	c4	c5
0	1	2	3	4	5
1	5	6	7	8	9
2	0	1	2	3	4



```
>>> df.pipe(suma,10)
   c1  c2  c3  c4  c5
0  11  12  13  14  15
1  15  16  17  18  19
2  10  11  12  13  14

>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4
```

Mitjana de columnes

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> import numpy as np
>>> df.apply(np.mean)
c1    2.0
c2    3.0
c3    4.0
c4    5.0
c5    6.0
dtype: float64
```

Mitjana de columnes

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> import numpy as np
>>> df.apply(np.mean)
c1    2.0
c2    3.0
c3    4.0
c4    5.0
c5    6.0
dtype: float64
```

Mitjana de columnes

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> import numpy as np
>>> df.apply(np.mean)
c1    2.0
c2    3.0
c3    4.0
c4    5.0
c5    6.0
dtype: float64
```

Mitjana de columnes

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> import numpy as np
>>> df.apply(np.mean)
c1    2.0
c2    3.0
c3    4.0
c4    5.0
c5    6.0
dtype: float64
```

Mitjana de columnes

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> import numpy as np
>>> df.apply(np.mean)
c1    2.0
c2    3.0
c3    4.0
c4    5.0
c5    6.0
dtype: float64
```



apply

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Mitjana de files

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> df.apply(np.mean,axis=1)
0    3.0
1    7.0
2    2.0
dtype: float64
```



apply

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

Mitjana de files

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> df.apply(np.mean,axis=1)
0    3.0
1    7.0
2    2.0
dtype: float64
```



Mitjana de files

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> df.apply(np.mean,axis=1)
0    3.0
1    7.0
2    2.0
dtype: float64
```



Mitjana de files

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> df.apply(np.mean,axis=1)
0    3.0
1    7.0
2    2.0
dtype: float64
```



apply

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> df.apply(lambda x: x.max() - x.min())
c1    5
c2    5
c3    5
c4    5
c5    5
dtype: int64
```



apply

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> df.apply(lambda x: x.max() - x.min())
c1    5
c2    5
c3    5
c4    5
c5    5
dtype: int64
```



apply

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4

>>> df.apply(lambda x: x.max() - x.min())
c1    5
c2    5
c3    5
c4    5
c5    5
dtype: int64
```

```
>>> df
   c1  c2  c3  c4  c5
0    1   2   3   4   5
1    5   6   7   8   9
2    0   1   2   3   4
>>> df.apply(lambda x: x.max() - x.min())
c1    5
c2    5
c3    5
c4    5
c5    5
dtype: int64
```

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4

>>> df.applymap(lambda x:x*100)
   c1   c2   c3   c4   c5
0  100  200  300  400  500
1  500  600  700  800  900
2    0  100  200  300  400
```



applymap

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4

>>> df.applymap(lambda x:x*100)
   c1   c2   c3   c4   c5
0  100  200  300  400  500
1  500  600  700  800  900
2    0  100  200  300  400
```



applymap

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4

>>> df.applymap(lambda x:x*100)
   c1   c2   c3   c4   c5
0  100  200  300  400  500
1  500  600  700  800  900
2    0  100  200  300  400
```



applymap

Pandas

groupby

merge

Aplicació de
funcions al
DataFrame

```
>>> df
   c1  c2  c3  c4  c5
0   1   2   3   4   5
1   5   6   7   8   9
2   0   1   2   3   4

>>> df.applymap(lambda x:x*100)
   c1   c2   c3   c4   c5
0  100  200  300  400  500
1  500  600  700  800  900
2    0  100  200  300  400
```