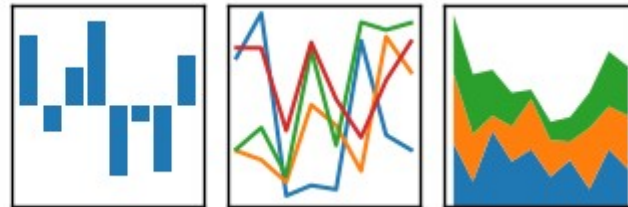


pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



<https://pandas.pydata.org/>



Pan_{el} of Da_{ta}

Pandas

- Un mòdul de Python pel tractament de dades
- Vegeu com instal·lar-ho a <https://pandas.pydata.org/>
- Per utilitzar-lo:

```
>>> import pandas
```

```
>>> s = pandas.Series([1, 4, 6, 8])
```

- O bé (més curt)

```
>>> import pandas as pd
```

```
>>> s = pd.Series([1, 4, 6, 8])
```

Pandas i numpy

- Les classes de *pandas* estan implementades sobre les del mòdul *numpy*, en particular la classe *ndarray* (taula multidimensional)
- Vegeu: numpy.org
- Per treballar en *numpy*:

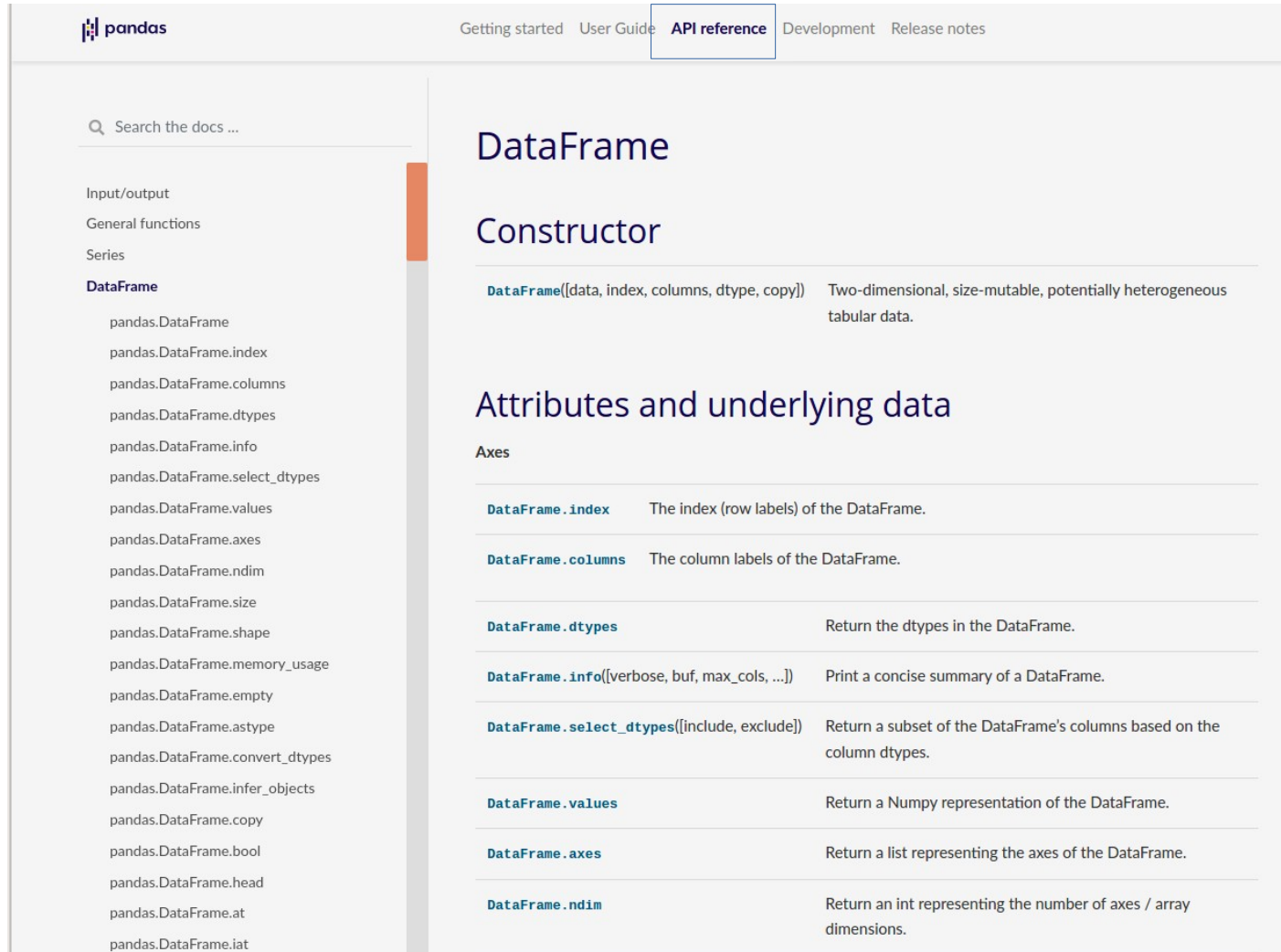
```
>>> import numpy as np
```

```
>>> a = np.ndarray(shape=(20,), dtype=np.uint8)
```

Crea un array uni-dimensional de 20 elements de tipus `unsigned int 8 bits` (un byte per valor)

Documentació

Consultem habitualment el manual de referència (API)



The screenshot shows the pandas API reference documentation for the DataFrame class. The page is titled "DataFrame" and "Constructor". It includes a search bar, a navigation menu on the left, and a list of attributes and methods. The "API reference" tab is selected in the top navigation bar.

Getting started User Guide **API reference** Development Release notes

Search the docs ...

Input/output
General functions
Series
DataFrame
pandas.DataFrame
pandas.DataFrame.index
pandas.DataFrame.columns
pandas.DataFrame.dtypes
pandas.DataFrame.info
pandas.DataFrame.select_dtypes
pandas.DataFrame.values
pandas.DataFrame.axes
pandas.DataFrame.ndim
pandas.DataFrame.size
pandas.DataFrame.shape
pandas.DataFrame.memory_usage
pandas.DataFrame.empty
pandas.DataFrame.astype
pandas.DataFrame.convert_dtypes
pandas.DataFrame.infer_objects
pandas.DataFrame.copy
pandas.DataFrame.bool
pandas.DataFrame.head
pandas.DataFrame.at
pandas.DataFrame.iat

DataFrame

Constructor

`DataFrame([data, index, columns, dtype, copy])` Two-dimensional, size-mutable, potentially heterogeneous tabular data.

Attributes and underlying data

Axes

| | |
|--|--|
| <code>DataFrame.index</code> | The index (row labels) of the DataFrame. |
| <code>DataFrame.columns</code> | The column labels of the DataFrame. |
| <code>DataFrame.dtypes</code> | Return the dtypes in the DataFrame. |
| <code>DataFrame.info([verbose, buf, max_cols, ...])</code> | Print a concise summary of a DataFrame. |
| <code>DataFrame.select_dtypes([include, exclude])</code> | Return a subset of the DataFrame's columns based on the column dtypes. |
| <code>DataFrame.values</code> | Return a Numpy representation of the DataFrame. |
| <code>DataFrame.axes</code> | Return a list representing the axes of the DataFrame. |
| <code>DataFrame.ndim</code> | Return an int representing the number of axes / array dimensions. |

Classes de pandas

| Classe | Descripció | Exemple |
|-----------|--------------|---|
| Series | Una taula 1D | <pre>>>> s = pd.Series(['a', 'b', 'c']) >>> s 0 a 1 b 2 c dtype: object</pre> |
| DataFrame | Una taula 2D | <pre>>>> d = pd.DataFrame({'id456':{'edat': 24, 'nom':'Anna'}, 'id564':{'edat':26, 'nom':'pol'}}) >>> d id456 id564 edat 24 26 nom Anna pol</pre> |
| Panel | Una taula 3D | <pre>>>> d = pd.Panel({'aula_01':{'id456':{'edat': 24, 'nom':'Anna'}, 'id564':{'edat':26, 'nom':'pol'}}, 'aula_02': {'id67':{'edat':20, 'nom':'Joan'}}}) >>> d <class 'pandas.core.panel.Panel'> Dimensions: 2 (items) x 2 (major_axis) x 3 (minor_axis) Items axis: aula_01 to aula_02 Major_axis axis: edat to nom Minor_axis axis: id456 to id67</pre> |

Classes de pandas

- Treballarem sobretot amb `DataFrame` i accessoriamment amb `Series`
- Les tres classes tenen molts mètodes similars amb noms idèntics
- Alguns mètodes dels `DataFrame` retornen `Series`, per tant cal saber treballar amb elles (veure transparències al final)

Series

```
>>> s = pd.Series([1, 2, 'a', 'b', 23, 'Pep'])
```

```
>>> s
```

```
0      1
```

```
1      2
```

```
2      a
```

```
3      b
```

```
4     23
```

```
5     Pep
```

```
dtype: object
```

```
>>> s.size
```

```
6
```

```
>>> s[3]
```

```
'b'
```

Exemple de creació a partir d'una llista

És una sola columna indexada de 0 a 5

Nombre d'elements

La indexació es fa per índex de files

Series

```
>>> s = pd.Series([1, 2, 'a', 'b', 23, 'Pep'],  
index = list('abcdef'))
```

```
>>> s
```

Exemple de creació a partir d'una llista

```
a      1  
b      2  
c      a  
d      b  
e     23  
f     Pep  
dtype: object
```

És una sola columna indexada de la 'a' a la 'f'

```
>>> s['a']
```

```
1
```

La indexació es fa per índex de files

```
>>> s['f']
```

```
'Pep'
```

Series

L'índex de les files pot ser de qualsevol tipus

L'indexació es fa tant per la posició de la fila com pel valor de l'índex.

```
>>> s2
a  123
b  245
c  125
d  234
```

```
>>> s2.index
```

```
Index(['a', 'b', 'c', 'd'], dtype='object', name='')
```

```
>>> s2[0]    La primera fila
123
```

```
>>> s2['a']   La primera fila
123
```

```
>>> s3
Abad      123
Polard    245
Junyent   125
Abad      234
```

```
Name: import, dtype: int64
```

```
>>> s3.index
```

```
Index(['Abad', 'Polard', 'Junyent',
'Abad'], dtype='object', name='valor')
```

```
>>> s3[0]    La primera fila
123
```

```
>>> s3['Abad']  Les files d'índex 'Abad'
Abad    123
Abad    234
Name: import, dtype: int64
```

Series

Alerta: L'index de les files pot estar repetida!!

```
>>> s = pd.Series([1, 2, 'a', 'b', 23, 'Pep'], index = list('abcaaf'))
```

```
>>> s
```

```
a    1
```

```
b    2
```

```
c    a
```

```
a    b
```

```
a   23
```

```
f   Pep
```

```
dtype: object
```

```
>>> s['a'] = 99
```

```
>>> s
```

```
a   99
```

```
b    2
```

```
c    a
```

```
a   99
```

```
a   99
```

```
f   Pep
```

```
dtype: object
```

Totes les files d'índex 'a' han canviat el seu valor!

DataFrame: atributs

Alguns d'ells ...

| Atribut | Descripció | Exemple amb el dataframe df |
|---------|--|---|
| axes | La llista de les etiquetes de les files i columnes | <pre>>>> df.axes [RangeIndex(start=0, stop=8, step=1), Index(['Nom', 'edat', 'marca'], dtype='object')]</pre> |
| values | La llista dels valors en forma de numpy.ndarray | <pre>>>> df.values array([['Pere', 21, 21.1], ['Pau', 22, 22.2], ... ['Carme', 24, 23.4]], dtype=object)</pre> |
| ndim | La dimensió | <pre>>>> df.ndim 2</pre> |
| size | Nombre d'elements filesxcolumnes | <pre>>>> df.size 24</pre> <pre>>>> df Nom edat marca 0 Pere 21 21.1 1 Pau 22 22.2 2 Joan 23 22.3 3 Jordi 19 21.9 4 Alba 18 28.6 5 Anna 21 20.1 6 Alicia 23 20.3 7 Carme 24 23.4</pre> |

DataFrame: mètodes (alguns)

| | | |
|-----------------|--|--|
| tail(n) | Les n darreres files del DataFrame | <pre>>>> df.tail(2) Nom edat marca 6 Alicia 23 20.3 7 Carme 24 23.4</pre> |
| head(n) | Les n primeres files del DataFrame | <pre>>>> df.head(2) Nom edat marca 0 Pere 21 21.1 1 Pau 22 22.2</pre> |
| count() | El nombre d'elements no nuls de cada columna | <pre>>>> df.count() Nom 8 edat 8 marca 8 dtype: int64</pre> |
| append(df2) | Retorna un nou DataFrame afegint al final del DataFrame les files de df2 | <pre>>>> df3= df.append(df2) >>> df3.tail(1) Nom edat marca 2 Julián 28 23.8</pre> |
| mean(),median() | La mitjana, la mediana de cada columna | <pre>>>> df.mean() edat 21.3750 marca 22.4875 dtype: float64</pre> |
| std() | Desviació típica de cada columna | <pre>>>> df.std() edat 2.065879 marca 2.701025 dtype: float64</pre> |
| max(), min() | Valor mínim i màxim de cada columna | <pre>>>> df.max() edat 24 Nom Pere marca 28.6 dtype: object</pre> |

DataFrame: mètodes

Classificació (a la documentació)

Conversion

Indexing, iteration

Binary operator functions

Function application, GroupBy & window

Computations / descriptive stats

Reindexing / selection / label manipulation

Missing data handling

Reshaping, sorting, transposing

Combining / comparing / joining / merging

Time Series-related

Metadata

Plotting

Sparse accessor

Serialization / IO / conversion

<https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>

DataFrame

Estructura de dades bi-dimensional en forma de taula

```
>>> import pandas as pd
>>> df = pd.read_csv('prova.csv')
>>> df
```

| | Nom | Cognom1 | Cognom2 | Edat | Punts |
|---|--------|----------|-----------|------|-------|
| 0 | Alba | Pérez | Juanola | 23 | 5 |
| 1 | Ricard | El Karim | Ullestres | 24 | 10 |
| 2 | Kim | Nguyen | NaN | 23 | 12 |
| 3 | Joan | Poll | Mató | 22 | 10 |
| 4 | Lluna | Picart | Hernández | 23 | 8 |
| 5 | Joana | Ranglas | Wayne | 24 | 9 |
| 6 | Alicia | Cugat | Bunyola | 23 | 9 |
| 7 | Albert | Jordan | López | 23 | 11 |
| 8 | Sàida | Blaize | Yussef | 22 | 13 |

Fitxer prova.csv

```
Nom,Cognom1,Cognom2,Edat,Punts
Alba,Pérez,Juanola,23,5
Ricard,El Karim,Ullestres,24,10
Kim,Nguyen,,23,12
Joan,Poll,Mató,22,10
Lluna,Picart,Hernández,23,8
Joana,Ranglas,Wayne,24,9
Alicia,Cugat,Bunyola,23,9
Albert,Jordan,López,23,11
Saïda,Blaize,Yussef,22,13
```

<https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>

DataFrame- Indexos

Estructura de dades bi-dimensional en forma de taula

```
>>> import pandas as pd
>>> df = pd.read_csv('prova.csv')
>>> df
```

| | Nom | Cognom1 | Cognom2 | Edat | Punts | columns |
|---|--------|----------|-----------|------|-------|---------|
| 0 | Alba | Pérez | Juanola | 23 | 5 | |
| 1 | Ricard | El Karim | Ullestres | 24 | 10 | |
| 2 | Kim | Nguyen | NaN | 23 | 12 | |
| 3 | Joan | Poll | Mató | 22 | 10 | |
| 4 | Lluna | Picart | Hernández | 23 | 8 | |
| 5 | Joana | Ranglas | Wayne | 24 | 9 | |
| 6 | Alicia | Cugat | Bunyola | 23 | 9 | |
| 7 | Albert | Jordan | López | 23 | 11 | |
| 8 | Sàida | Blaize | Yussef | 22 | 13 | |

NaN = Not a Number
No hi ha valor per la columna Cognom2 i fila d'índex 2

```
>>> df.columns
Index(['Nom', 'Cognom1', 'Cognom2', 'Edat', 'Punts'], dtype='object')
>>> df.index
RangeIndex(start=0, stop=9, step=1)
>>> df.axes
[RangeIndex(start=0, stop=9, step=1), Index(['Nom', 'Cognom1', 'Cognom2', 'Edat', 'Punts'], dtype='object')]
```

<https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>

DataFrame- Indexos

Dos altres dataframes amb les mateixes dades però diferents índexos:

```
>>> df2
      Nom      Cognom1      Cognom2      Edat      Punts
12  Alba      Pérez      Juanola      23         5
4   Ricard    El Karim    Ullestres    24        10
5   Kim       Nguyen     NaN        23        12
12  Joan      Poll       Mató        22        10
11  Lluna     Picart     Hernández  23         8
6   Joana     Ranglas    Wayne      24         9
6   Alicia    Cugat      Bunyola    23         9
7   Albert    Jordan     López      23        11
12  Saïda     Blaize     Yussef     22        13
```

```
>>> df3
      Nom      Cognom1      Cognom2      Edat      Punts
a   Alba      Pérez      Juanola      23         5
b   Ricard    El Karim    Ullestres    24        10
c   Kim       Nguyen     NaN        23        12
d   Joan      Poll       Mató        22        10
e   Lluna     Picart     Hernández  23         8
f   Joana     Ranglas    Wayne      24         9
g   Alicia    Cugat      Bunyola    23         9
h   Albert    Jordan     López      23        11
i   Saïda     Blaize     Yussef     22        13
```

Observeu que l'**índex pot estar repetit**, no ha d'estar necessàriament ordenat ni ser de tipus enter

<https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html>

DataFrame

Consulta de columnes

Una columna : **df[índex columna]**

```
>>> df['Nom']
0    Alba
1  Ricard
2    Kim
3    Joan
4   Lluna
5   Joana
6  Alicia
7  Albert
8   Saïda
Name: Nom, dtype: object
```

Diverses columnes : **df[llista d'índexos de columnes]**

```
>>> df[['Nom', 'Edat']]
```

| | Nom | Edat |
|---|--------|------|
| 0 | Alba | 23 |
| 1 | Ricard | 24 |
| 2 | Kim | 23 |
| 3 | Joan | 22 |
| 4 | Lluna | 23 |
| 5 | Joana | 24 |
| 6 | Alicia | 23 |
| 7 | Albert | 23 |
| 8 | Saïda | 22 |

Alerta doble clau:
l'índex és una llista

<http://pandas.pydata.org/pandas-docs/stable/indexing.html>

Dataframe- View i còpies

```

>>> df
   Nom      Cognom1      Cognom2  Edat  Punts
0   Alba      Pérez      Juanola   23     5
1 Ricard  El Karim  Ullestres   24    10
2   Kim      Nguyen           NaN   23    12
>>> s = df['Nom']
>>> s
0      Alba
1   Ricard
2      Kim
Name: Nom, dtype: object
>>> df['Nom'][0] = 'Pol'
>>> df
   Nom      Cognom1      Cognom2  Edat  Punts
0   Pol      Pérez      Juanola   23     5
1 Ricard  El Karim  Ullestres   24    10
2   Pol      Nguyen           NaN   23    12
>>> s
0      Pol
1   Ricard
2      Pol
3      Joan
4      Lluna
5      Joana
6   Alicia
7   Albert
8     Saïda
Name: Nom, dtype: object

```

← És una vista, no una còpia!!!

Dataframe- View i còpies

```
>>> s[6] = 'Inés'
__main__:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a
DataFrame
```

See the caveats in the documentation:

```
https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
```

```
>>> df
   Nom Cognom1  Cognom2  Edat  Punts
0  Alba  Pérez  Juanola   23     5
1  Ricard ElKarim Ullestres  24    10
2   Kim  Nguyen      NaN   23    12
3  Joan   Poll   Mató    22    10
4  Lluna  Picart Hernández  23     8
5  Joana Ranglas  Wayne   24     9
6  Inés  Cugat  Bunyola   23     9
7  Albert Jordan   López   23    11
8  Saïda  Blaize  Yussef   22    13
```

Alerta!!! df ha quedat modificat.
És el que volíem?

DataFrame

Consulta de files

Files: `df.loc[índex de fila]`

```
>>> df.loc[0]
Nom      Alba
Cognom1  Pérez
Cognom2  Juanola
Edat     23
Punts    5
Name: 0, dtype: object
```

Series

```
>>> df2.loc[12]
      Nom Cognom1 Cognom2 Edat
Punts
12 Alba Pérez Juanola 23 5
12 Joan Poll Mató 22 10
12 Saïda Blaize Yussef 22 13
```

DataFrame

```
>>> df3.loc['a']
Nom      Alba
Cognom1  Pérez
Cognom2  Juanola
Edat     23
Punts    5
Name: a, dtype: object
```

Series

De la i-èsima fila d'un DataFrame: `df.iloc[i]`

```
>>> df.iloc[0]
Nom      Alba
Cognom1  Pérez
Cognom2  Juanola
Edat     23
Punts    5
Name: 0, dtype: object
```

Series

```
>>> df2.iloc[0]
Nom      Alba
Cognom1  Pérez
Cognom2  Juanola
Edat     23
Punts    5
Name: 12, dtype: object
```

Series

```
>>> df3.iloc[0]
Nom      Alba
Cognom1  Pérez
Cognom2  Juanola
Edat     23
Punts    5
Name: a, dtype: object
```

Series

<http://pandas.pydata.org/pandas-docs/stable/indexing.html>

DataFrame

Consulta amb llista de columnes i slices

```
>>> df2.loc[[12, 6, 7]]
   Nom Cognom1 Cognom2 Edat Punts
12 Alba  Pérez Juanola  23    5
12 Joan  Poll  Mató    22   10
12 Saïda Blaize Yussef  22   13
6  Joana Ranglas Wayne  24    9
6  Alicia Cugat Bunyola  23    9
7  Albert Jordan  López  23   11
```

Alerta doble clau:
l'índex és una llista

A partir d'una llista d'índexos o de posicions

```
>>> df[1:3]
   Nom Cognom1 Cognom2 Edat Punts
1 Ricard El Karim Ullestres  24   10
2 Kim Nguyen NaN 23 12
```

A partir d'un slice directe

```
>>> df.loc[1:3]
   Nom Cognom1 Cognom2 Edat Punts
1 Ricard El Karim Ullestres  24   10
2 Kim Nguyen NaN 23 12
3 Joan Poll Mató 22 10
```

A partir d'un slice de loc. Observeu que ara la fila d'índex màxim està inclosa

```
>>> df.iloc[1:3]
   Nom Cognom1 Cognom2 Edat Punts
1 Ricard El Karim Ullestres  24   10
2 Kim Nguyen NaN 23 12
```

A partir d'un slice de iloc. Observeu que és equivalent a df[1:3]

<http://pandas.pydata.org/pandas-docs/stable/indexing.html>

DataFrame

Consulta amb slice

```
>>> df2[1:3]
      Nom  Cognom1  Cognom2  Edat  Punts
2  Ricard  El Karim  Ullestres   24    10
12  Kim    Nguyen    NaN      23    12
```

A partir d'un slice directe

```
>>> df2.iloc[1:3]
      Nom  Cognom1  Cognom2  Edat  Punts
2  Ricard  El Karim  Ullestres   24    10
12  Kim    Nguyen    NaN      23    12
```

A partir d'un slice de iloc. Observeu que és equivalent a `df2[1:3]`

```
>>> df2.loc[1:3]
      Nom  Cognom1  Cognom2  Edat  Punts
1  Alba   Pérez   Juanola   23     5
2  Ricard  El Karim  Ullestres   24    10
12  Kim    Nguyen    NaN      23    12
3  Joan   Poll     Mató     22    10
```

A partir d'un slice de loc. Són els files entre l'índex (label) 1 i 3

```
>>> df2.loc[12:8]
Dóna un error perquè els índexos estan repetits:
KeyError: 'Cannot get left slice bound for non-unique label: 12'
```

<http://pandas.pydata.org/pandas-docs/stable/indexing.html>

DataFrame

Consulta d'elements

`df[índex_columna][índex_fila]`

```
>>> df['Nom'][0]
'Alba'
```

Millor utilitzar l'indexació amb tuple!!



`df.loc[índex_fila, índex_columna]`

```
>>> df.loc[0, 'Nom']
'Alba'
```

`df.at[índex_fila, índex_columna]`

```
>>> df3.at['c', 'Nom']
'Kim'
```

```
>>> df2['Nom'][12]
12    Alba
12    Joan
12    Saïda
Name: Nom, dtype:
object
>>> df3['Nom']['a']
'Alba'
```

Alerta
Hi ha més d'una fila
amb aquest índex
No és un sol element

```
>>> df.loc[12, 'Nom']
12    Alba
12    Joan
12    Saïda
Name: Nom, dtype: object
```

```
>>> df2.at[12, 'Nom']
array(['Alba', 'Joan', 'Saïda'],
      dtype=object)
>>> df2.at[12, 'Nom'][0]
'Alba'
```

Observeu: és 1 array

<http://pandas.pydata.org/pandas-docs/stable/indexing.html>

DataFrame

Modificació d'elements

`df.at[index_fila, index_columna] = valor`

```
>>> df.at[0, 'Nom'] = 'Pol'
```

```
>>> df2.at[12, 'Nom'] = 'Pol'
```

```
>>> df3['a', 'Nom'] = 'Pol'
23
```

Alerta!
Modificarà tots els noms
de totes les files amb índex 12

```
>>> df2
```

| | Nom | Cognom1 | Cognom2 | Edat | Punts |
|----|--------|----------|-----------|------|-------|
| 12 | Pol | Pérez | Juanola | 23 | 5 |
| 4 | Ricard | El Karim | Ullestres | 24 | 10 |
| 5 | Kim | Nguyen | NaN | 23 | 12 |
| 12 | Pol | Poll | Mató | 22 | 10 |
| 11 | Lluna | Picart | Hernández | 23 | 8 |
| 6 | Joana | Ranglas | Wayne | 24 | 9 |
| 6 | Alicia | Cugat | Bunyola | 23 | 9 |
| 7 | Albert | Jordan | López | 23 | 11 |
| 12 | Pol | Blaize | Yussef | 22 | 13 |

DataFrame

Selecció avançada

Les files en les que una o més columnes compleixen una condició

```
>>> df[df['Edat'] == 23]
   Nom Cognom1 Cognom2 Edat Punts
0   Pol   Pérez  Juanola   23     5
2   Kim  Nguyen      NaN   23    12
4  Lluna  Picart  Hernández  23     8
6  Alicia  Cugat  Bunyola   23     9
7  Albert  Jordan   López   23    11
```

df[df[columna] op_relació valor]
op_relació: {>, <, >=, <=, ==, !=}

df[(selecció) op_bool (selecció)]
op_bool = {& (and), | (or), ~(not)}

```
>>> df[(df['Edat'] == 23) & (df['Punts'] > 10)]
   Nom Cognom1 Cognom2 Edat Punts
2   Kim  Nguyen      NaN   23    12
7  Albert  Jordan   López   23    11
```

```
>>> df[~(df['Edat'] == 23) | (df['Punts'] == 11)]
   Nom Cognom1 Cognom2 Edat Punts
1  Ricard  El Karim  Ullestres   24    10
3   Joan    Poll    Mató     22    10
5   Joana  Ranglas  Wayne     24     9
7  Albert  Jordan   López     23    11
8   Saïda  Blaize  Yussef     22    13
```

Alerta!
Cada expressió entre ()

DataFrame

Selecció avançada

Condicions sobre columnes de tipus str: utilitzar els mètodes **Series.str.mètode**

```
>>> df[df['Nom'].str.startswith('A')]
   Nom Cognom1 Cognom2 Edat Punts
0  Alba  Pérez  Juanola   23     5
6  Alicia  Cugat  Bunyola   23     9
7  Albert  Jordan   López   23    11
```

```
>>> df[(df['Nom'].str.contains('a')
| (df['Nom'].str.contains('A')))]
   Nom Cognom1 Cognom2 Edat Punts
0  Alba  Pérez  Juanola   23     5
1  Ricard  El Karim  Ullestres   24    10
3   Joan   Poll   Mató   22    10
4  Lluna  Picart  Hernández   23     8
5  Joana  Ranglas   Wayne   24     9
6  Alicia  Cugat  Bunyola   23     9
7  Albert  Jordan   López   23    11
8  Saïda  Blaize  Yussef   22    13
```

[pandas.Series.str.casefold](#) [pandas.Series.str.rjust](#)
[pandas.Series.str.cat](#) [pandas.Series.str.rpartition](#)
[pandas.Series.str.center](#) [pandas.Series.str.rstrip](#)
[pandas.Series.str.contains](#) [pandas.Series.str.slice](#)
[pandas.Series.str.count](#) [pandas.Series.str.slice_replace](#)
[pandas.Series.str.decode](#) [pandas.Series.str.split](#)
[pandas.Series.str.encode](#) [pandas.Series.str.rsplit](#)
[pandas.Series.str.endswith](#) [pandas.Series.str.startswith](#)
[pandas.Series.str.extract](#) [pandas.Series.str.strip](#)
[pandas.Series.str.extractall](#) [pandas.Series.str.swapcase](#)
[pandas.Series.str.find](#) [pandas.Series.str.title](#)
[pandas.Series.str.findall](#) [pandas.Series.str.translate](#)
[pandas.Series.str.get](#) [pandas.Series.str.upper](#)
[pandas.Series.str.index](#) [pandas.Series.str.wrap](#)
[pandas.Series.str.join](#) [pandas.Series.str.zfill](#)
[pandas.Series.str.len](#) [pandas.Series.str.isalnum](#)
[pandas.Series.str.ljust](#) [pandas.Series.str.isalpha](#)
[pandas.Series.str.lower](#) [pandas.Series.str.isdigit](#)
[pandas.Series.str.lstrip](#) [pandas.Series.str.isspace](#)
[pandas.Series.str.match](#) [pandas.Series.str.islower](#)
[pandas.Series.str.normalize](#) [pandas.Series.str.isupper](#)
[pandas.Series.str.pad](#) [pandas.Series.str.istitle](#)
[pandas.Series.str.partition](#) [pandas.Series.str.isnumeric](#)
[pandas.Series.str.repeat](#) [pandas.Series.str.isdecimal](#)
[pandas.Series.str.replace](#) [pandas.Series.str.get_dummies](#)
[pandas.Series.str.rfind](#)
[pandas.Series.str.rindex](#)

Estadístiques

pandas.DataFrame.corr
 pandas.DataFrame.corrwith
 pandas.DataFrame.count
 pandas.DataFrame.cov
pandas.DataFrame.cummax
 pandas.DataFrame.cummin
 pandas.DataFrame.cumprod
 pandas.DataFrame.cumsum
 pandas.DataFrame.describe
 pandas.DataFrame.kurt
 pandas.DataFrame.kurtosis
 pandas.DataFrame.mad
 pandas.DataFrame.max
 pandas.DataFrame.mean
 pandas.DataFrame.median
 pandas.DataFrame.min
 pandas.DataFrame.mode
 pandas.DataFrame.pct_change
 pandas.DataFrame.prod
 pandas.DataFrame.product
 pandas.DataFrame.quantile
 pandas.DataFrame.rank

I algunes altres...

Es poden aplicar a tot el dataframe, una o més columnes 1 o més files

```
>>> df.describe()

```

| | Edat | Punts |
|-------|-----------|-----------|
| count | 9.000000 | 9.000000 |
| mean | 23.000000 | 9.666667 |
| std | 0.707107 | 2.345208 |
| min | 22.000000 | 5.000000 |
| 25% | 23.000000 | 9.000000 |
| 50% | 23.000000 | 10.000000 |
| 75% | 23.000000 | 11.000000 |
| max | 24.000000 | 13.000000 |

```
>>> df.count()
Nom          9
Cognom1      9
Cognom2      8
Edat         9
Punts        9
dtype: int64
>>> len(df)
9
```

Alerta! Observeu la diferència entre `df.count()` i `len(df)`
Entre `max(df)` i `df.max()`. No és mateix!!

Exploreu les altres funcions

Cerca de la fila en la qual hi ha el màxim en una columna

El nom de la persona amb més punts

```
>>> df['Punts'].max()
13
>>> df['Punts'].idxmax()
8
>>> df.at[8, 'Nom']
'Saïda'
>>> df.at[8, 'Punts']= 12
>>> df['Punts'].max()
12
>>> df['Punts'].idxmax()
2
```

És el valor del màxim de punts

És l'índex de la persona que ha obtingut més punts

És el nom de la persona amb més punts

Ara el màxim és 12 i hi ha 2 persones amb aquests punts

En cas d'empat, la primera ocurrència

DataFrame

Creació

A partir d'un diccionari o d'una llista i afegint files i columnes:

```
>>> import pandas as pd
>>> df = pd.DataFrame()
>>> df['nom'] = ['Pol', 'Pere', 'Pau', 'Joan', 'Júlia', 'Laia', 'Rosa', 'Alicia']
>>> df['edat'] = [23, 22, 23, 22, 22, 21, 21, 22]
>>> df['grup'] = ['A', 'A', 'B', 'A', 'C', 'A', 'B', 'A']
>>> df['nota'] = [5.6, 3.5, 2.1, 6.7, 8.7, 4.5, 6.7, 9.1]
>>> df2 = pd.DataFrame([{'nom': 'Quique', 'edat': 23, 'grup': 'C', 'nota': 6.4}])
>>> df3 = df.append(df2)
>>> df3.tail(2)
```

| | edat | grup | nom | nota |
|---|------|------|--------|------|
| 7 | 22 | A | Alicia | 9.1 |
| 0 | 23 | C | Quique | 6.4 |

També a partir de json, html i molts altres!

Llegint d'un fitxer csv

```
>>> df = pd.read_csv('dades.csv')
>>> df2 = pd.read_csv('dades.csv', use_cols= ['nom', 'edat'])
```

Llegint d'una url

```
>>> df = pd.read_csv(url)
```

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html

DataFrame

Crear i desar a fitxer

Es poden desar les dades a disc amb `to_csv` (i molts altres)

```
>>> df3.to_csv('dades.csv', sep = ';', index = False)
>>> df4 = pd.read_csv('dades.csv', sep = ';')
>>> df4.head()
```

| | edat | grup | nom | nota |
|---|------|------|-------|------|
| 0 | 23 | A | Pol | 5.6 |
| 1 | 22 | A | Pere | 3.5 |
| 2 | 23 | B | Pau | 2.1 |
| 3 | 22 | A | Joan | 6.7 |
| 4 | 22 | C | Júlia | 8.7 |

Es poden llegir només algunes columnes amb `use_cols` i especificar el format de les columnes

```
>>> df5 = pd.read_csv('dades.csv', sep = ';', usecols=['edat', 'grup'])
>>> df5
```

| | edat | grup |
|---|------|-------|
| 0 | 23 | A |
| 1 | 22 | A ... |

```
>>> df4 = pd.read_csv('dades.csv', sep = ';', usecols=['edat', 'grup'],
dtype={'edat':float})
>>> df4
```

| | edat | grup |
|---|------|-------|
| 0 | 23.0 | A |
| 1 | 22.0 | A ... |

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html

La setmana vinent

Operacions sobre columnes

Groupby

Tractament del valors perduts