

# Python para Análisis de Datos

Módulo 04

# Pandas

# Indexing

El hecho de utilizar etiquetas hace que Pandas ofrezca una variedad de formas para manipular los datos.

La indexación directa sobre una serie devuelve uno de sus elementos y sobre un dataframe devuelve una de sus columnas (cada una de ellas es una serie). Hay que usar como índice el elemento que se tenga como etiqueta, similar a lo que sucede con diccionarios. Se puede usar un rango en caso de las series.



# Indexing

```
serie
```

```
a    5
b    7
c    2
d    4
e    3
f    9
g    8
dtype: int64
```

```
serie["d"]
```

```
4
```

```
serie["b": "e"]
```

```
b    7
c    2
d    4
e    3
dtype: int64
```

```
data["Age"]
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
...
886     27.0
887     19.0
888      NaN
889     26.0
890     32.0
```

```
Name: Age, Length: 891, dtype: float64
```

## Indexing

Un dataframe también puede ser indexado con una lista de columnas, devolviendo un dataframe.

```
data[["Survived", "Age", "Name"]]
```

|     | Survived | Age  | Name  |
|-----|----------|------|---|
| 0   | 0        | 22.0 | Braund, Mr. Owen Harris                           |
| 1   | 1        | 38.0 | Cumings, Mrs. John Bradley (Florence Briggs Th... |
| 2   | 1        | 26.0 | Heikkinen, Miss. Laina                            |
| 3   | 1        | 35.0 | Futrelle, Mrs. Jacques Heath (Lily May Peel)      |
| 4   | 0        | 35.0 | Allen, Mr. William Henry                          |
| ... | ...      | ...  | ...   |
| 886 | 0        | 27.0 | Montvila, Rev. Juozas                             |
| 887 | 1        | 19.0 | Graham, Miss. Margaret Edith                      |
| 888 | 0        | NaN  | Johnston, Miss. Catherine Helen "Carrie"          |
| 889 | 1        | 26.0 | Behr, Mr. Karl Howell                             |
| 890 | 0        | 32.0 | Dooley, Mr. Patrick                               |

891 rows × 3 columns

# Label Indexing

Pandas provee un método general muy versátil para seleccionar elementos en base a las etiquetas. Esto se hace a través del atributo `.loc`. Para seleccionar un elemento se ponen las etiquetas que le correspondan. Recordemos que las etiquetas pueden ser cualquier objeto de python, incluso números. Cuando se usan números con `.loc`, estos representan las etiquetas y no las posiciones.

También se pueden hacer slices con etiquetas. En este caso el slice es inclusivo tanto con la primer etiqueta como con la segunda. También se pueden poner listas para seleccionar etiquetas específicas.



## Label Indexing

```
data.loc[3, "Name"]
```

```
'Futrelle, Mrs. Jacques Heath (Lily May Peel)'
```

```
data.loc[[25,30], ["Survived", "Age"]]
```

|    | Survived | Age  |
|----|----------|------|
| 25 | 1        | 38.0 |
| 30 | 0        | 40.0 |

```
data.loc[25:30, "PassengerId": "Pclass"]
```

|    | PassengerId | Survived | Pclass |
|----|-------------|----------|--------|
| 25 | 26          | 1        | 3      |
| 26 | 27          | 0        | 3      |
| 27 | 28          | 0        | 1      |
| 28 | 29          | 1        | 3      |
| 29 | 30          | 0        | 3      |
| 30 | 31          | 0        | 1      |

# Integer Indexing

A su vez existe el método `iloc` para acceder a los elementos según las posiciones. En este caso los números representan las posiciones y no las etiquetas. Funciona como los slices de `numpy`.

La primera posición empieza a contar desde cero, los slices incluyen el primer índice pero excluyen el segundo. También se pueden poner listas de índices.





## Integer Indexing

```
data.iloc[125,4]
```

'male'

```
data.iloc[[25,30], [2, 5]]
```

|    | Pclass | Age  |
|----|--------|------|
| 25 | 3      | 38.0 |
| 30 | 1      | 40.0 |

```
data.iloc[10:18, -3:]
```

|    | Fare    | Cabin | Embarked |
|----|---------|-------|----------|
| 10 | 16.7000 | G6    | S        |
| 11 | 26.5500 | C103  | S        |
| 12 | 8.0500  | NaN   | S        |
| 13 | 31.2750 | NaN   | S        |
| 14 | 7.8542  | NaN   | S        |
| 15 | 16.0000 | NaN   | S        |
| 16 | 29.1250 | NaN   | Q        |
| 17 | 13.0000 | NaN   | S        |

# Filtros

También es posible seleccionar elementos según condiciones de sus valores. La expresión condicional produce un arreglo de booleanos y se devuelven los valores para los que el valor de ese arreglo es verdadero. `data["Age"] >=30` devuelve una serie de booleanos y `data[data["Age"] >= 30]` devuelve todas las filas del dataframe para las cuales el valor de la edad es mayor o igual a 30.

También se pueden usar condiciones dentro de `loc` y de `iloc`, para una o las dos dimensiones. Por ejemplo, `data.loc[data["Age"] >= 30, ["Name", "Sex"]]` selecciona las filas de las columnas *Name* y *Sex* cuyas edades son mayores o iguales a 30.



## Filtros

```
data[data["Age"] >= 30 ]
```

| PassengerId | Survived | Pclass | Name | Sex   | Age    | SibSp | Parch | Ticket | Fare     | Cabin   | Embarked |     |
|-------------|----------|--------|------|---|--------|-------|-------|--------|----------|---------|----------|-----|
| 1           | 2        | 1      | 1    | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0  | 1     | 0      | PC 17599 | 71.2833 | C85      | C   |
| 3           | 4        | 1      | 1    | Futrelle, Mrs. Jacques Heath (Lily May Peel)      | female | 35.0  | 1     | 0      | 113803   | 53.1000 | C123     | S   |
| 4           | 5        | 0      | 3    | Allen, Mr. William Henry                          | male   | 35.0  | 0     | 0      | 373450   | 8.0500  | NaN      | S   |
| 6           | 7        | 0      | 1    | McCarthy, Mr. Timothy J                           | male   | 54.0  | 0     | 0      | 17463    | 51.8625 | E46      | S   |
| 11          | 12       | 1      | 1    | Bonnell, Miss. Elizabeth                          | female | 58.0  | 0     | 0      | 113783   | 26.5500 | C103     | S   |
| ...         | ...      | ...    | ...  | ...   | ...    | ...   | ...   | ...    | ...      | ...     | ...      | ... |
| 873         | 874      | 0      | 3    | Vander Cruyssen, Mr. Victor                       | male   | 47.0  | 0     | 0      | 345765   | 9.0000  | NaN      | S   |
| 879         | 880      | 1      | 1    | Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)     | female | 56.0  | 0     | 1      | 11767    | 83.1583 | C50      | C   |
| 881         | 882      | 0      | 3    | Markun, Mr. Johann                                | male   | 33.0  | 0     | 0      | 349257   | 7.8958  | NaN      | S   |
| 885         | 886      | 0      | 3    | Rice, Mrs. William (Margaret Norton)              | female | 39.0  | 0     | 5      | 382652   | 29.1250 | NaN      | Q   |
| 890         | 891      | 0      | 3    | Dooley, Mr. Patrick                               | male   | 32.0  | 0     | 0      | 370376   | 7.7500  | NaN      | Q   |

330 rows × 12 columns

## Filtros

```
data.loc[data["Age"] >= 30, ["Name", "Sex"]]
```

|     | Name  | Sex    |
|-----|---|--------|
| 1   | Cumings, Mrs. John Bradley (Florence Briggs Th... | female |
| 3   | Futrelle, Mrs. Jacques Heath (Lily May Peel)      | female |
| 4   | Allen, Mr. William Henry                          | male   |
| 6   | McCarthy, Mr. Timothy J                           | male   |
| 11  | Bonnell, Miss. Elizabeth                          | female |
| ... | ...   | ...    |
| 873 | Vander Cruyssen, Mr. Victor                       | male   |
| 879 | Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)     | female |
| 881 | Markun, Mr. Johann                                | male   |
| 885 | Rice, Mrs. William (Margaret Norton)              | female |
| 890 | Dooley, Mr. Patrick                               | male   |

330 rows × 2 columns

# ¡Muchas gracias!

¡Sigamos trabajando!