


Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

2.7. Selección de datos

2.7.1 Por filas loc-iloc

- **NOTA:** loc-iloc indica selección por localización e índice de localización.

Ejercicio 80_01. Seleccionar subconjunto de filas loc-iloc en Python

```
# df.loc podemos seleccionar filas según su etiqueta
# df.iloc seleccionamos según la posición
## Importamos las librerías necesarias
import numpy as np
import pandas as pd
df = pd.DataFrame({'est': ['A', 'B', 'C', 'D'], 'contaminante': [np.nan, 'SO2',
'NO2', 'SO2'], 'tipo': ['Train', 'Test', 'Train', 'Test'], 'valor': [10, 22, 30, 2]})
df.index = ['Lunes', 'Martes', 'Miércoles', 'Jueves']
print(df)
```

	est	contaminante	tipo	valor
Lunes	A	NaN	Train	10
Martes	B	SO2	Test	22
Miércoles	C	NO2	Train	30
Jueves	D	SO2	Test	2

```
# selección por una sola fila
print(df.loc['Lunes'])
```

```
est          A
contaminante NaN
tipo         Train
valor        10
Name: Lunes, dtype: object
```

```
# selección por un rango de filas
```

```
print(df.loc['Lunes':'Miércoles'])
```

	est	contaminante	tipo	valor
Lunes	A	NaN	Train	10
Martes	B	SO2	Test	22
Miércoles	C	NO2	Train	30


```
# selección de filas y columnas distintas
```

```
print(df.loc['Lunes':'Miércoles'][['contaminante', 'tipo']])
```

	contaminante	tipo
Lunes	NaN	Train
Martes	SO2	Test
Miércoles	NO2	Train

```
# selección por posición.iloc
```

```
print(df.iloc[0:2, 2])
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
Lunes      Train
Martes     Test
Name: tipo, dtype: object
```

```
print(df.iloc[0, 2])
```

```
Train
```

```
# selección según criterio lógico
```

```
df[df.valor > 10]
```

	est	contaminante	tipo	valor
Martes	B	SO2	Test	22
Miércoles	C	NO2	Train	30

Ejercicio 80_02. Seleccionar subconjunto de columnas en MATLAB

```
% Creamos un DataTable similar al DataFrame en pandas
```

```
data = {
```

```
    'A', NaN, 'Train', 10;
```

```
    'B', 'SO2', 'Test', 22;
```

```
    'C', 'NO2', 'Train', 30;
```

```
    'D', 'SO2', 'Test', 2;
```

```
};
```

```
filas = {'Lunes', 'Martes', 'Miércoles', 'Jueves'};
```

```
columnas = {'est', 'contaminante', 'tipo', 'valor'};
```

```
df = table(data(:, 1), data(:, 2), data(:, 3), cell2mat(data(:, 4)), 'RowNames',
```

```
filas, 'VariableNames', columnas);
```

```
% Mostramos el DataTable generado
```

```
disp('DataTable generado:');
```

```
disp(df);
```

```
DataTable generado:
```

	est	contaminante	tipo	valor
Lunes	{'A'}	{[NaN]}	{'Train'}	10
Martes	{'B'}	{'SO2'}	{'Test' }	22
Miércoles	{'C'}	{'NO2'}	{'Train'}	30
Jueves	{'D'}	{'SO2'}	{'Test' }	2

```
% Selección por una sola fila
```

```
disp('Selección por una sola fila:');
```

```
disp(df('Lunes', :));
```


```
Selección por una sola fila:
```

	est	contaminante	tipo	valor
Lunes	{'A'}	{[NaN]}	{'Train'}	10

```
% Selección por rango de filas
```

```
disp('Selección por rango de filas:');
```

```
% disp(df('Lunes':'Miércoles', :)); % No sirve con string
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
disp(df(1:3,:));
```

Selección por rango de filas:

	est	contaminante	tipo	valor
Lunes	{'A'}	{[NaN]}	{'Train'}	10
Martes	{'B'}	{'SO2'}	{'Test' }	22
Miércoles	{'C'}	{'NO2'}	{'Train'}	30

```
% Selección de columnas específicas por rango de filas
```

```
disp('Selección de columnas específicas por rango de filas:');
```

```
% disp(df('Lunes':'Miércoles', {'contaminante', 'tipo'})); % No sirve con string
```

```
disp(df(1:3, 2:3));
```

Selección de columnas específicas por rango de filas:

	contaminante	tipo
Lunes	{[NaN]}	{'Train'}
Martes	{'SO2'}	{'Test' }
Miércoles	{'NO2'}	{'Train'}

```
% Selección por posición iloc
```

```
disp('Selección por posición iloc:');
```

```
disp(df{1:2, 3});
```

Selección por posición iloc:

```
{'Train'}  
{'Test' }
```

```
% Selección de un elemento por posición iloc
```

```
disp('Selección de un elemento por posición iloc:');
```

```
disp(df{'Lunes', 'tipo'});
```

Selección de un elemento por posición iloc:

```
{'Train'}
```

```
% Selección según criterio lógico
```

```
disp('Selección según criterio lógico:');
```

```
disp(df(df.valor > 10, :));
```

Selección según criterio lógico:

	est	contaminante	tipo	valor
Martes	{'B'}	{'SO2'}	{'Test' }	22
Miércoles	{'C'}	{'NO2'}	{'Train'}	30

2.7.2. Selección subconjunto de columnas

Ejercicio 81_01. Seleccionar subconjunto de columnas en Python

```
# df.loc podemos seleccionar filas según su etiqueta
# df.iloc seleccionamos según la posición
# Importamos las librerías necesarias
import numpy as np
import pandas as pd

df = pd.DataFrame({'Est': ['A', 'B', 'C', 'D'], 'contaminante': [np.nan, 'SO2',
'NO2', 'SO2'], 'Tipo': ['Train', 'Test', 'Train', 'Test']})
df.index = ['Lunes', 'Martes', 'Miércoles', 'Jueves']
print(df)
print('\n Puede seleccionarse una columna específica: \n')
print(df[['Est']])
print('\n Seleccionar rango de columnas por nombre: \n')
print(df.loc[:,['contaminante','tipo']])
print('\n Seleccionar rango de columnas por posición: \n')
df.iloc[:,[0,1]])
```

	Est	contaminante	Tipo
Lunes	A	NaN	Train
Martes	B	SO2	Test
Miércoles	C	NO2	Train
Jueves	D	SO2	Test

Puede seleccionarse una columna específica:

Lunes	A
Martes	B
Miércoles	C
Jueves	D

Seleccionar rango de columnas por nombre:


	contaminante	Tipo
Lunes	NaN	Train
Martes	SO2	Test
Miércoles	NO2	Train
Jueves	SO2	Test

Seleccionar rango de columnas por posición:

	Est	contaminante
Lunes	A	NaN
Martes	B	SO2
Miércoles	C	NO2
Jueves	D	SO2

Ejercicio 81_02. Seleccionar subconjunto de columnas en MATLAB

```
% Creamos un DataTable similar al DataFrame en pandas
data = {
    'A', NaN, 'Train';
    'B', 'SO2', 'Test';
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
'C', 'N02', 'Train';
'D', 'S02', 'Test';
};
filas = {'Lunes', 'Martes', 'Miércoles', 'Jueves'};
columnas = {'Est', 'contaminante', 'Tipo'};
df = table(data(:, 1), data(:, 2), data(:, 3), 'RowNames', filas, 'VariableNames',
columnas);
% Mostramos el DataTable generado
disp('DataTable generado:');
disp(df);
% Puede seleccionarse una columna específica
disp('Puede seleccionarse una columna específica:');
disp(df.Est);
% Seleccionar rango de columnas por nombre
disp('Seleccionar rango de columnas por nombre:');
disp(df(:, {'contaminante', 'Tipo'}));
% Seleccionar rango de columnas por posición
disp('Seleccionar rango de columnas por posición:');
disp(df(:, 1:2));
```

DataTable generado:

	Est	contaminante	Tipo
Lunes	{'A'}	{[NaN]}	{'Train'}
Martes	{'B'}	{'S02'}	{'Test' }
Miércoles	{'C'}	{'N02'}	{'Train'}
Jueves	{'D'}	{'S02'}	{'Test' }

Puede seleccionarse una columna específica:

```
{'A'}
{'B'}
{'C'}
{'D'}
```

Seleccionar rango de columnas por nombre:

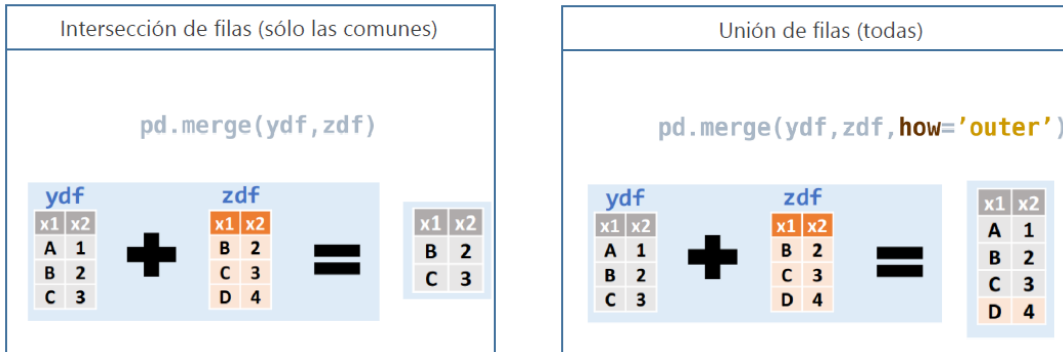
	contaminante	Tipo
Lunes	{[NaN]}	{'Train'}
Martes	{'S02'}	{'Test' }
Miércoles	{'N02'}	{'Train'}
Jueves	{'S02'}	{'Test' }

Seleccionar rango de columnas por posición:

	Est	contaminante
Lunes	{'A'}	{[NaN]}
Martes	{'B'}	{'S02'}
Miércoles	{'C'}	{'N02'}
Jueves	{'D'}	{'S02'}

2.7.3. Intersección de datasets con filas comunes

```
# combinar datasets con filas comunes:  
# pd.merge(ydf,zdf) # intersección de filas solo comunes  
# pd.merge(ydf,zdf, how='outer') # Unión de filas (todas)
```




Ejercicio 82_01. Combinar datasets con merge en Python

```
import pandas as pd  
  
df1 = pd.DataFrame({'clave': ['datos 1', 'datos 2', 'datos 3'], 'valores df1': [1, 2, 3]})  
df2 = pd.DataFrame({'clave': ['datos 1', 'datos 2', 'datos 4'], 'valores df2': [4, 5, 6]})  
df3 = pd.merge(df1, df2, on='clave')  
print("Dataframe 1: ")  
print(df1)  
print("\n Dataframe 2: ")  
print(df2)  
print("\n Dataframe intersección de ambos: ")  
print(df3)
```

```
DataTable 1:  
   clave  valores df1  
0  datos 1           1  
1  datos 2           2  
2  datos 3           3
```

```
DataTable 2:  
   clave  valores df2  
0  datos 1           4  
1  datos 2           5  
2  datos 4           6
```

```
DataTable intersección de ambos:  
   clave  valores df1  valores df2  
0  datos 1           1           4  
1  datos 2           2           5
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 82_01. Combinar datasets con merge en MATLAB

```
% Creamos dos DataTables similares a los DataFrames en pandas
data1 = {'datos 1', 1; 'datos 2', 2; 'datos 3', 3};
data2 = {'datos 1', 4; 'datos 2', 5; 'datos 4', 6};
columnas = {'clave', 'valores_df1'};
df1 = table(data1(:, 1), cell2mat(data1(:, 2)), 'VariableNames', columnas);
df2 = table(data2(:, 1), cell2mat(data2(:, 2)), 'VariableNames', columnas);
% Mostramos los DataTables generados
disp('DataTable 1:');
disp(df1);
disp('DataTable 2:');
disp(df2);
% Fusionamos los DataTables por la columna 'clave'
df3 = outerjoin(df1, df2, 'Keys', 'clave', 'MergeKeys', true);
% Renombramos las variables para reflejar el origen
df3.Properties.VariableNames = {'clave', 'valores_df1_df1', 'valores_df1_df2'};
% Mostramos el DataTable resultante
disp('DataTable intersección de ambos:');
disp(df3);
```

DataTable 1:

clave	valores_df1
{'datos 1'}	1
{'datos 2'}	2
{'datos 3'}	3

DataTable 2:

clave	valores_df1
{'datos 1'}	4
{'datos 2'}	5
{'datos 4'}	6

DataTable intersección de ambos:

clave	valores_df1_df1	valores_df1_df2
{'datos 1'}	1	4
{'datos 2'}	2	5
{'datos 3'}	3	NaN
{'datos 4'}	NaN	6

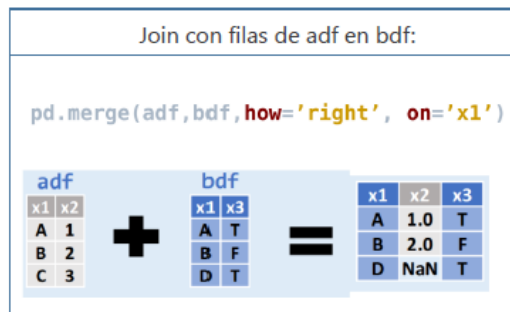
2.7.4. Combinaciones a nivel columna

Python

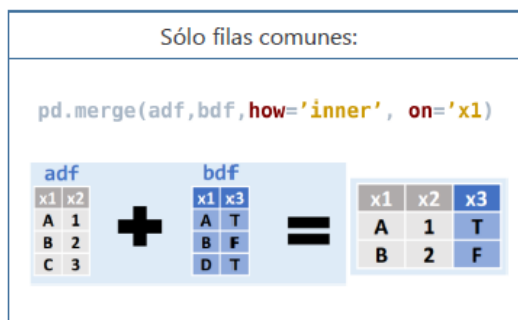
```
# join con filas de bdf en adf (ejemplo de 2 matrices 3x2)  
pd.merge(adf,bdf, how='left', on='x1')
```




```
# join con filas de adf en bdf (ejemplo de 2 matrices 3x2)  
pd.merge(adf,bdf, how='right', on='x1')
```



```
# join sólo filas comunes (ejemplo de 2 matrices 3x2)  
pd.merge(adf,bdf, how='inner', on='x1')
```



```
## join todas las filas y valores (ejemplo de 2 matrices 3x2)  
## pd.merge(adf,bdf, how='outer', on='x1')
```


Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Todas las filas y valores:

```
pd.merge(adf,bdf,how='outer', on='x1')
```

adf		bdf		Result		
x1	x2	x1	x3	x1	x2	x3
A	1	A	T	A	1	T
B	2	B	F	B	2	F
C	3	D	T	C	3	NaN
				D	NaN	T

Ejercicio 83_01. Agrupamientos en dataframes en Python

```
# Importamos las librerías necesarias
import numpy as np
import pandas as pd
df = pd.DataFrame({'Estaciones': ['Est1', 'Est2', 'Est1', 'Est1', 'Est2', 'Est1',
'Est2', 'Est2'], 'contaminantes': ['CO', 'NO2', 'CO', 'PM10', 'PM10', 'CO', 'PM10',
'O3'], 'tipo': ['Train', 'Test', 'Train', 'Train', 'Test', 'Test', 'Test', 'Train'],
'concentraciones': np.random.rand(8), # Aleatorias})
print('Dataframe original: ')
print(df)
print('\n Sumamos agrupando estaciones: ')
df2 = df.groupby('Estaciones').sum()
print(df2)
print('\n Sumamos agrupando por estaciones y contaminantes: ')
df3 = df.groupby(['Estaciones', 'Contaminantes']).sum()
print(df3)
print('\n Sumamos agrupando por estaciones, contaminantes y tipos: ')
df4 = df.groupby(['Estaciones', 'Contaminantes', 'Tipo']).sum()
print(df4)
```

Dataframe original:


	Estaciones	Contaminantes	Tipo	Concentraciones
0	Est1	CO	Train	0.53063
1	Est2	NO	Test	0.83242
2	Est1	CO	Train	0.59749
3	Est1	PM10	Train	0.33531
4	Est2	PM10	Test	0.29923
5	Est1	CO	Test	0.45259
6	Est2	PM10	Test	0.42265
7	Est2	O3	Train	0.35961

Sumamos agrupando estaciones:

Estaciones	Concentraciones
Est1	1.916
Est2	1.9139

Sumamos agrupando por estaciones y contaminantes:

Estaciones	Contaminantes	Concentraciones
Est1	CO	1.5807

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

	PM10	0.33531
Est2	NO2	0.83242
	PM10	0.72187
	O3	0.35961

Sumamos agrupando por estaciones, contaminantes y tipos:

Estaciones	Contaminantes	Tipo	Concentraciones
Est1	CO	Test	1.12810
		Train	0.45259
Est2	PM10	Train	0.33531
	NO2	Test	0.83242
	O3	Train	0.72187
	PM10	Test	0.35961


Ejercicio 83_02. Agrupaciones en tablas en MATLAB

% Creamos una tabla similar a un DataFrame en pandas

```
data = {
    'Est1', 'CO', 'Train', rand();
    'Est2', 'NO2', 'Test', rand();
    'Est1', 'CO', 'Train', rand();
    'Est1', 'PM10', 'Train', rand();
    'Est2', 'PM10', 'Test', rand();
    'Est1', 'CO', 'Test', rand();
    'Est2', 'PM10', 'Test', rand();
    'Est2', 'O3', 'Train', rand();
};
columnas = {'Estaciones', 'Contaminantes', 'Tipo', 'Concentraciones'};
df = table(data(:, 1), data(:, 2), data(:, 3), cell2mat(data(:, 4)), 'VariableNames',
columnas);
% Mostramos la tabla original
disp('Tabla original:');
disp(df);
% Sumamos agrupando por estaciones
df2 = grpstats(df, 'Estaciones', 'sum', 'DataVars', 'Concentraciones');
% Mostramos el resultado
disp('Sumamos agrupando estaciones:');
disp(df2);
% Sumamos agrupando por estaciones y contaminantes
df3 = grpstats(df, {'Estaciones', 'Contaminantes'}, 'sum', 'DataVars',
'Concentraciones');
% Mostramos el resultado
disp('Sumamos agrupando por estaciones y contaminantes:');
disp(df3);
% Sumamos agrupando por estaciones, contaminantes y tipos
df4 = grpstats(df, {'Estaciones', 'Contaminantes', 'Tipo'}, 'sum', 'DataVars',
'Concentraciones');
% Mostramos el resultado
disp('Sumamos agrupando por estaciones, contaminantes y tipos:');
disp(df4);
```

Tabla original:

Estaciones	Contaminantes	Tipo	Concentraciones
------------	---------------	------	-----------------

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```

{'Est1'}      {'CO'  }      {'Train'}      0.53063
{'Est2'}      {'NO2' }      {'Test' }      0.83242
{'Est1'}      {'CO'  }      {'Train'}      0.59749
{'Est1'}      {'PM10'}     {'Train'}      0.33531
{'Est2'}      {'PM10'}     {'Test' }      0.29923
{'Est1'}      {'CO'  }      {'Test' }      0.45259
{'Est2'}      {'PM10'}     {'Test' }      0.42265
{'Est2'}      {'O3'   }      {'Train'}      0.35961

```

Sumamos agrupando estaciones:

	Estaciones	GroupCount	sum_Concentraciones
Est1	{'Est1'}	4	1.916
Est2	{'Est2'}	4	1.9139

Sumamos agrupando por estaciones y contaminantes:

	Estaciones	Contaminantes	GroupCount	sum_Concentraciones
Est1_CO	{'Est1'}	{'CO' }	3	1.5807
Est1_PM10	{'Est1'}	{'PM10'}	1	0.33531
Est2_NO2	{'Est2'}	{'NO2' }	1	0.83242
Est2_PM10	{'Est2'}	{'PM10'}	2	0.72187
Est2_O3	{'Est2'}	{'O3' }	1	0.35961


Sumamos agrupando por estaciones, contaminantes y tipos:

	Estaciones	Contaminantes	Tipo	GroupCount
Est1_CO_Train	{'Est1'}	{'CO' }	{'Train'}	2
Est1_CO_Test	{'Est1'}	{'CO' }	{'Test' }	1
Est1_PM10_Train	{'Est1'}	{'PM10'}	{'Train'}	1
Est2_NO2_Test	{'Est2'}	{'NO2' }	{'Test' }	1
Est2_PM10_Test	{'Est2'}	{'PM10'}	{'Test' }	2
Est2_O3_Train	{'Est2'}	{'O3' }	{'Train'}	1
sum_Concentraciones				
				1.12810
				0.45259
				0.33531
				0.83242
				0.72187
				0.35961

2.7.5. Tablas dinámicas

Esta funcionalidad nos permite agrupar, ordenar, calcular datos y manejar datos de una forma muy similar a la que se hace con las hojas de cálculo de Excel.

Index: Especifica el agrupamiento a nivel fila.
Column: Especifica el agrupamiento a nivel columna.
Values: Valores numéricos que queremos resumir.

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)


Ejercicio 84_01. Tablas dinámicas en Python

```
# Importamos las librerías necesarias
import numpy as np
import pandas as pd
df = pd.DataFrame({'Estaciones': ['Est1', 'Est2', 'Est1', 'Est1', 'Est2', 'Est1',
'Est2', 'Est2'], 'contaminantes': ['CO', 'NO2', 'CO', 'PM10', 'PM10', 'CO', 'PM10',
'03'], 'tipo': ['Train', 'Test', 'Train', 'Train', 'Test', 'Test', 'Test', 'Train'],
'concentraciones': np.random.rand(8), # Aleatorias})
print('Dataframe original: \n')
print(df)
# Concentraciones según el tipo, agrupados por estaciones y contaminantes
df2 = pd.pivot_table(df, values='concentraciones', index=['Estaciones',
'Contaminantes'], cols=['Tipo'])
print ('\n Agrupamos datos: concentraciones según el tipo, agrupados por estaciones y
contaminantes: \n')
print(df2)
# Añadimos funciones
df3 = pd.pivot_table(df, values='concentraciones', index=['Estaciones',
'Contaminantes'], cols=['Tipo'], aggfunc=[np.size, np.sum, np.mean])
print('\n Agrupamos datos y contamos, hacemos la suma y la media: \n')
print(df3)
```

Ejercicio 84_02. Tablas dinámicas en MATLAB

```
% Creamos una tabla similar a un DataFrame en pandas
data = {
    'Est1', 'CO', 'Train', rand();
    'Est2', 'NO2', 'Test', rand();
    'Est1', 'CO', 'Train', rand();
    'Est1', 'PM10', 'Train', rand();
    'Est2', 'PM10', 'Test', rand();
    'Est1', 'CO', 'Test', rand();
    'Est2', 'PM10', 'Test', rand();
    'Est2', '03', 'Train', rand();
};

columnas = {'Estaciones', 'Contaminantes', 'Tipo', 'Concentraciones'};
df = table(data(:, 1), data(:, 2), data(:, 3), cell2mat(data(:, 4)), 'VariableNames',
columnas);
% Mostramos la tabla original
disp('Tabla original:');
disp(df);
% Agrupamos datos: concentraciones según el tipo, agrupados por estaciones y
contaminantes
df2 = unstack(df, 'Concentraciones', 'Tipo');
% Mostramos el resultado
disp('Agrupamos datos: concentraciones según el tipo, agrupados por estaciones y
contaminantes:');
disp(df2);
% Agrupamos datos y contamos, hacemos la suma y la media
df3 = varfun(@size, df, 'InputVariables', 'Concentraciones', 'GroupingVariables',
{'Estaciones', 'Contaminantes', 'Tipo'});
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
df3.Properties.VariableNames = {'Count'};
df3.Sum_Concentraciones = varfun(@sum, df, 'InputVariables', 'Concentraciones',
'GroupingVariables', {'Estaciones', 'Contaminantes', 'Tipo'});
df3.Mean_Concentraciones = varfun(@mean, df, 'InputVariables', 'Concentraciones',
'GroupingVariables', {'Estaciones', 'Contaminantes', 'Tipo'});
% Mostramos el resultado
disp('Agrupamos datos y contamos, hacemos la suma y la media:');
disp(df3);
```

Tabla original:

Estaciones	Contaminantes	Tipo	Concentraciones
{'Est1'}	{'CO' }	{'Train'}	0.67973
{'Est2'}	{'NO2' }	{'Test' }	0.036563
{'Est1'}	{'CO' }	{'Train'}	0.8092
{'Est1'}	{'PM10'}	{'Train'}	0.74862
{'Est2'}	{'PM10'}	{'Test' }	0.12019
{'Est1'}	{'CO' }	{'Test' }	0.52505
{'Est2'}	{'PM10'}	{'Test' }	0.32583
{'Est2'}	{'O3' }	{'Train'}	0.54645

Agrupamos datos: concentraciones según el tipo, agrupados por estaciones y contaminantes:

Estaciones	Contaminantes	Test	Train
{'Est1'}	{'CO' }	0.52505	1.4889
{'Est2'}	{'NO2' }	0.036563	NaN
{'Est1'}	{'PM10'}	NaN	0.74862
{'Est2'}	{'PM10'}	0.44602	NaN
{'Est2'}	{'O3' }	NaN	0.54645

Agrupamos datos y contamos, hacemos la suma y la media:


Estaciones	Contaminantes	Tipo	GroupCount	size_Concentraciones
{'Est1'}	{'CO' }	{'Test' }	1	1 1
{'Est1'}	{'CO' }	{'Test' }	1	0.52505
{'Est1'}	{'CO' }	{'Test' }	1	0.52505
{'Est1'}	{'CO' }	{'Train'}	2	2 1

Agrupamos datos y contamos, hacemos la suma y la media:

Estaciones	Contaminantes	Tipo	GroupCount	size_Concentraciones
{'Est1'}	{'CO' }	{'Test' }	1	1 1
{'Est1'}	{'CO' }	{'Train'}	2	2 1
{'Est1'}	{'PM10'}	{'Train'}	1	1 1
{'Est2'}	{'NO2' }	{'Test' }	1	1 1
{'Est2'}	{'O3' }	{'Train'}	1	1 1
{'Est2'}	{'PM10'}	{'Test' }	2	2 1

Sum_Concentraciones

Estaciones	Contaminantes	Tipo	GroupCount	sum_Concentraciones
{'Est1'}	{'CO' }	{'Test' }	1	0.52505
{'Est1'}	{'CO' }	{'Train'}	2	1.4889
{'Est1'}	{'PM10'}	{'Train'}	1	0.74862

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

{'Est2'}	{'NO2' }	{'Test' }	1	0.036563
{'Est2'}	{'O3' }	{'Train'}	1	0.54645
{'Est2'}	{'PM10'}	{'Test' }	2	0.44602
Mean_Concentraciones				
Estaciones	Contaminantes	Tipo	GroupCount	mean_Concentraciones
{'Est1'}	{'CO' }	{'Test' }	1	0.52505
{'Est1'}	{'CO' }	{'Train'}	2	0.74447
{'Est1'}	{'PM10'}	{'Train'}	1	0.74862
{'Est2'}	{'NO2' }	{'Test' }	1	0.036563
{'Est2'}	{'O3' }	{'Train'}	1	0.54645
{'Est2'}	{'PM10'}	{'Test' }	2	0.22301