


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

2.6. Aplicar funciones a dataframes y tablas

Funciones aplicables: `sum()`, `min()`, `max()`, `mean()`, `var()`, `std()`, `count()`, `size()`, `median()`, `apply(function)`.

Pueden aplicarse a series, agrupaciones (ejemplo ejercicio 83_01. Agrupamientos en dataframes en Python) y columnas.

Otras funciones útiles:

- Seleccionar aleatoriamente una fracción de filas: `df.sample(frac=0.5)`
- Seleccionar n filas aleatoriamente: `df.sample(n=10)`
- Seleccionar y ordenar las n entradas más grandes: `df.nlargest(n, 'columna')`
- Seleccionar y ordenar las n entradas más pequeñas: `df.nsmallest(n, 'columna')`
- Ordenar valores: `df.sort_values('name', ascending = true)`
- Ordenar índice: `df.sort_index()`
- Renombrar columnas: `df.rename(columns={'n':'nombre'})`
- Borrar columnas: `df.drop(columns=['nombre','edad'])`

En Matlab: `sum()`, `min()`, `max()`, `mean()`, `var()`, `std()`, `count()`, `size()`, `median()`, `arrayfun(@(x) x^2, A); % Aplicar función x^2 a cada elemento de A.`

Ejercicio 78_01. Aplicar funciones a dataframes en Python

```
# Importamos las librerías necesarias
import pandas as pd
import numpy as np


# Creamos un dataframe con datos aleatorios (5 filas y 4 columnas)
# Para Los nombres de filas se usan las fechas, para las columnas la lista
df = pd.DataFrame(np.random.randn(5, 4))
print("Dataframe generado: ")
print(df)
df2 = df.apply(np.cumsum)
print("\n Aplicamos función que calcula la suma acumulativa por columna: \n")
print(df2)
df3 = df.mean(axis=0)
print("\n Aplicamos función que calcula la media por columna: \n")
print(df3)
df4 = df.mean(axis=1)
print("\n Aplicamos función que calcula la media por fila: \n")
print(df4)
# Lambda son pequeñas funciones anónimas
df5 = df.apply(lambda x: x.max() - x.min())
print("\n Aplicamos función que calcula máximo menos mínimo por columna: \n")
print(df5)
```

Dataframe generado:

```
0 1 2 3
0 -1.856295 0.132211 0.107456 1.877389 1 -1.857194 -1.419337 -0.493022 -1.434923 2 -
1.782115 -0.329186 -1.112902 -1.141936 3 1.083036 0.184746 -0.088146 -0.034818 4
0.482112 -0.621373 1.612355 1.199380
```

Aplicamos función que calcula la suma acumulativa por columna:

```
0 1 2 3
```

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

```
0 -1.856295 0.132211 0.107456 1.877389 1 -3.713490 -1.287127 -0.385566 0.442466 2 -5.495604 -1.616312 -1.498468 -0.699470 3 -4.412568 -1.431566 -1.586613 -0.734288 4 -3.930456 -2.052939 0.025742 0.465091
```

Aplicamos función que calcula la media por columna:

```
0 -0.786091
1 -0.410588
2 0.005148
3 0.093018
dtype: float64
```

Aplicamos función que calcula la media por fila: 0 0.065190


```
1 -1.301119
2 -1.091534
3 0.286205
4 0.668118
dtype: float64
```

Aplicamos función que calcula máximo menos mínimo por columna:

```
0 2.940230
1 1.604084
2 2.725257
3 3.312312
dtype: float64
```

Ejercicio 78_02. Aplicar funciones a tablas en MATLAB

```
% Creamos un DataTable con datos aleatorios (5 filas y 4 columnas)
x = randn(5, 4);
df = table(x(:,1),x(:,2),x(:,3),x(:,4),...
    'VariableNames',{'A','B','C','D'},...
    'RowNames',{'2022-01-01'; '2022-01-02'; '2022-01-03'; '2022-01-04'; '2022-01-05'})
% Mostramos el DataTable generado
disp('DataTable generado:');
disp(df);
% Aplicamos la función sum() por columna
df_sum = varfun(@sum, df);
disp('Suma por columna:');
disp(df_sum);
% Aplicamos la función cumsum() por columna
df_cumsum = varfun(@cumsum, df);
disp('Suma acumulativa por columna:');
disp(df_cumsum);
% Aplicamos la función mean() por columna
df_mean_col = varfun(@mean, df);
disp('Media por columna:');
disp(df_mean_col);
% Creamos la función anónima para calcular max() - min() por columna
df_max_min_diff = varfun(@(x) max(x) - min(x), df);
disp('Máximo menos mínimo por columna:');
disp(df_max_min_diff);
% Aplicamos la función mean() por fila
df_mean_row = mean(df{:, :}, 2); % dimensión 2 indica que haga la media por filas de las columnas
disp('Media por fila:');
```

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_mean_row);
```

```
x =
```

```
  2.4245    1.8779   -0.5583   -0.2099
  0.9594    0.9407   -0.3114   -1.6989
 -0.3158    0.7873   -0.5700    0.6076
  0.4286   -0.8759   -1.0257   -0.1178
 -1.0360    0.3199   -0.9087    0.6992
```

```
df = 5x4 table
```

	A	B	C	D
2022-01-01	2.4245	1.8779	-0.55829	-0.2099
2022-01-02	0.9594	0.9407	-0.31143	-1.6989
2022-01-03	-0.31577	0.78735	-0.57001	0.6076
2022-01-04	0.42862	-0.87587	-1.0257	-0.1178
2022-01-05	-1.036	0.31995	-0.90875	0.69916

```
DataTable generado:
```

	A	B	C	D
2022-01-01	2.4245	1.8779	-0.55829	-0.2099
2022-01-02	0.9594	0.9407	-0.31143	-1.6989
2022-01-03	-0.31577	0.78735	-0.57001	0.6076
2022-01-04	0.42862	-0.87587	-1.0257	-0.1178
2022-01-05	-1.036	0.31995	-0.90875	0.69916

```
Suma por columna:
```

sum_A	sum_B	sum_C	sum_D
2.4607	3.05	-3.3742	-0.7198

```
Suma acumulativa por columna:
```


cumsum_A	cumsum_B	cumsum_C	cumsum_D
2.4245	1.8779	-0.55829	-0.2099
3.3839	2.8186	-0.86972	-1.9088
3.0681	3.6059	-1.4397	-1.3012
3.4967	2.73	-2.4655	-1.419
2.4607	3.05	-3.3742	-0.7198

```
Media por columna:
```

mean_A	mean_B	mean_C	mean_D
0.49215	0.61	-0.67484	-0.14396

```
Máximo menos mínimo por columna:
```

Fun_A	Fun_B	Fun_C	Fun_D
-------	-------	-------	-------

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

```
1.3543    0.6783    2.3467    2.6604
```

Media por fila:

```
-0.4249  
-0.6928  
-0.3411  
-0.8122  
0.0846
```

Ejercicio 79_01. Concatenar dataframes en Python

```
# Importamos Las Librerías necesarias  
import numpy as np  
import pandas as pd  
df1 = pd.DataFrame(np.random.randn(3, 3))  
df2 = pd.DataFrame(np.random.randn(3, 3))  
print('Dataframe original 1: ')  
print(df1)  
print('\n Dataframe original 2: ')  
print(df2)  
df3 = pd.concat([df1, df2])  
print('\n Dataframe concatenando los anteriores en filas: ')  
print(df3)  
df4 = pd.concat([df1, df2], axis=1)  
print('\n Dataframe concatenando los anteriores en columnas: ')  
print(df4)
```

Dataframe original 1:

```
0 1 2  
0 -0.551075 0.765915 -0.499833  
1 0.817082 -0.350316 1.319587  
2 -0.225708 0.170537 0.351697
```

Dataframe original 2:

```
0 1 2  
0 0.364047 -0.387928 -2.339652  
1 0.595656 0.830094 1.349182 2 -0.410960 -0.373968 0.428950
```

Dataframe concatenando los anteriores en filas: 0 1 2

```
0 -0.551075 0.765915 -0.499833  
1 0.817082 -0.350316 1.319587  
2 -0.225708 0.170537 0.351697  
0 0.364047 -0.387928 -2.339652  
1 0.595656 0.830094 1.349182 2 -0.410960 -0.373968 0.428950
```

Dataframe concatenando los anteriores en columnas:

```
0 1 2 0 1 2  
0 -0.551075 0.765915 -0.499833 0.364047 -0.387928 -2.339652  
1 0.817082 -0.350316 1.319587 0.595656 0.830094 1.349182 2 -0.225708 0.170537  
0.351697 -0.410960 -0.373968 0.428950
```

Ejercicio 79_02. Concatenar dataframes en MATLAB

```
% Creamos un DataTable con datos aleatorios (5 filas y 4 columnas)
x1 = randn(3, 3);
x2 = randn(3, 3);
x3 = randn(3, 3);
df1 = table(x1(:,1),x1(:,2),x1(:,3),...
    'VariableNames',{'A','B','C'});
df2 = table(x2(:,1),x2(:,2),x2(:,3),...
    'VariableNames',{'D','E','F'});
df3 = table(x3(:,1),x3(:,2),x3(:,3),...
    'VariableNames',{'A','B','C'});
% Mostramos las tablas generadas
disp('Tabla 1 generado:');
disp(df1);
disp('Tabla 2 generado:');
disp(df2);
disp('Tabla 3 generado:');
disp(df3);
% Concatenar los DataTables en columnas
df4 = [df1, df2];
disp('DataTable concatenando los anteriores en columnas:');
disp(df4);
% Concatenar los DataTables en filas
df5 = [df1; df3];
disp('DataTable concatenando los anteriores en filas:');
disp(df5);
```

Tabla 1 generado:

A	B	C
-1.8288	0.44892	-3.073
1.3845	-0.36326	0.62628
-0.062727	-1.0206	-0.28668

Tabla 2 generado:


D	E	F
-0.19734	-0.72945	-1.2813
0.40561	1.1473	-2.2033
-1.4193	0.59786	-0.57125

Tabla 3 generado:

A	B	C
0.214	-1.1223	-0.96097
0.94238	0.30616	-0.65374
0.093725	-1.1723	-1.2294

DataTable concatenando los anteriores en columnas:

A	B	C	D	E	F
---	---	---	---	---	---

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

-1.8288	0.44892	-3.073	-0.19734	-0.72945	-1.2813
1.3845	-0.36326	0.62628	0.40561	1.1473	-2.2033
-0.062727	-1.0206	-0.28668	-1.4193	0.59786	-0.57125

DataTable concatenando los anteriores en filas:

A	B	C
-1.8288	0.44892	-3.073
1.3845	-0.36326	0.62628
-0.062727	-1.0206	-0.28668
0.214	-1.1223	-0.96097
0.94238	0.30616	-0.65374
0.093725	-1.1723	-1.2294