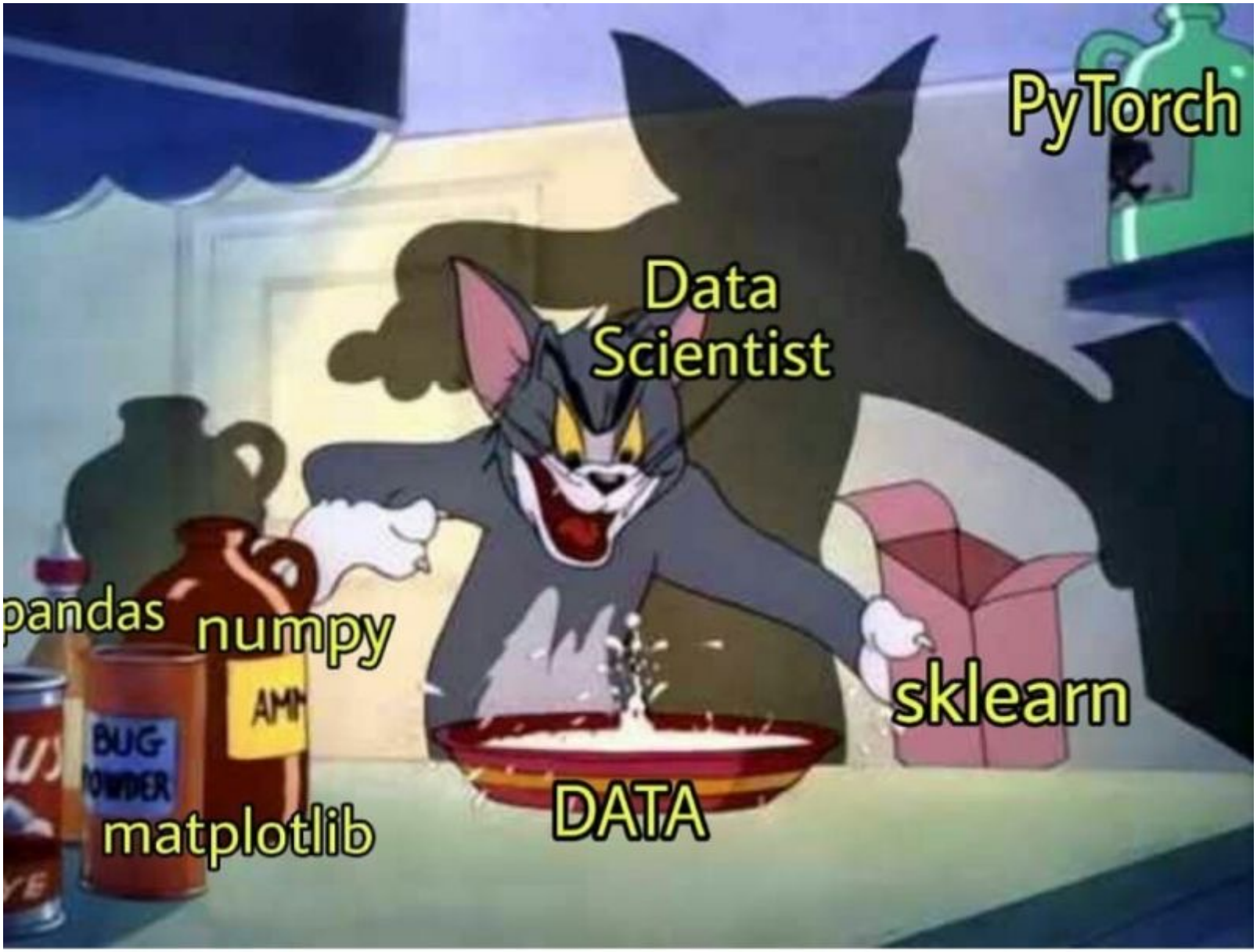
FACULTAD DE
INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBÁÑEZ

Introducción a Python

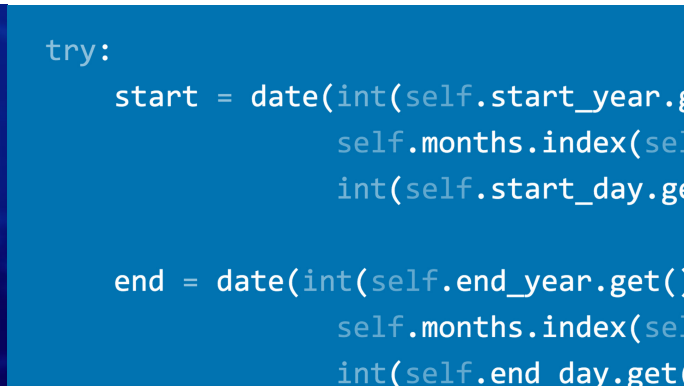
pandas y gráficos

Miguel Carrasco
miguel.carrasco@uai.cl


PyTorch
Data
Scientist
pandas
numpy
sklearn
DATA
matplotlib

Temario

- Introducción a Pandas
  - Operaciones básicas de datos
  - Agrupaciones de datos
  - Pivot table
  - CrossTab
  - Gráfica matplotlib
  - Gráfica Plotly



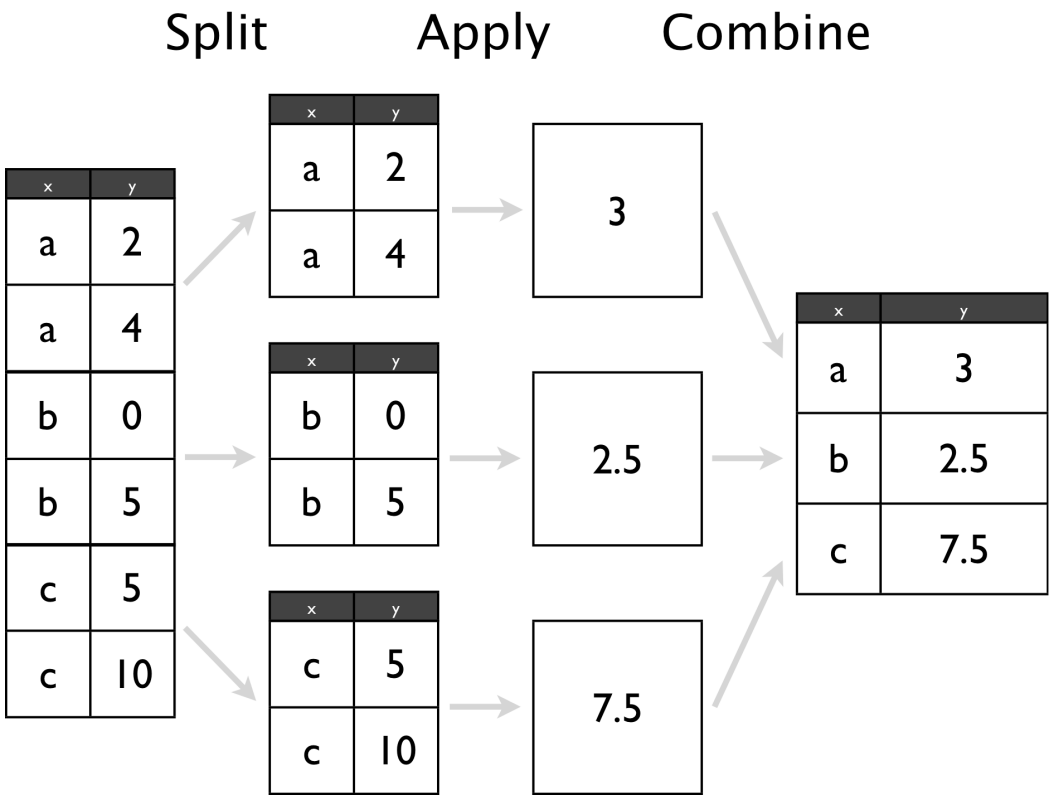
Agrupaciones

Combinar dos o más atributos en un único atributo o objeto

Reducción de datos	Reducir el número de atributos o objetos
Cambio de escala	Emplear una escala mayor para reducir el número de posibles salidas
Estabilizar datos	Agregar datos tiene a disminuir la variabilidad

Agrupaciones

Combinar dos o más atributos en un único atributo o objeto



Agrupaciones

```
import pandas as pd
df = pd.read_csv('simple_data.csv')
grupos = df.groupby('Edad')

print(grupos.describe())
```

groupby () nos permite crear agrupaciones por un campo

con el commando describe() Podemos ver un resumen estadístico de los valores agrupados según su agrupación

	Altura							
	count	mean	std	min	25%	50%	75%	max
Edad								
2.0	1.0	6.000000	NaN	6.0	6.00	6.0	6.0	6.0
12.0	1.0	13.000000	NaN	13.0	13.00	13.0	13.0	13.0
23.0	4.0	16.550000	19.349677	3.0	5.25	9.1	20.4	45.0
24.0	1.0	23.000000	NaN	23.0	23.00	23.0	23.0	23.0
32.0	1.0	3.000000	NaN	3.0	3.00	3.0	3.0	3.0
45.0	3.0	37.666667	16.502525	24.0	28.50	33.0	44.5	56.0
67.0	1.0	4.000000	NaN	4.0	4.00	4.0	4.0	4.0

df

	Nombre	Altura	Edad
0	Pedro	12.2	23.0
1	Juan	13.0	12.0
2	Diego	45.0	23.0
3	Arturo	24.0	45.0
4	Felipe	23.0	24.0
5	Andrés	56.0	45.0
6	Juan	6.0	2.0
7	Sofía	23.0	NaN
8	Antonia	6.0	23.0
9	Bernardita	4.0	67.0
10	Camila	3.0	32.0
11	Pedro	3.0	23.0
12	Alvaro	34.0	NaN
13	Magnolia	33.0	45.0

Agrupaciones

```python
import pandas as pd
df = pd.read_csv('simple_data.csv')
grupos = df.groupby('Edad')

print('datos separados por grupos: \n')
for key, item in grupos:
    print(grupos.get_group(key), "\n\n")
```

```
datos separados por grupos:

    Nombre  Altura  Edad
6     Juan     6.0     2
12  Alvaro     NaN     2


  Nombre  Altura  Edad
1   Juan    13.0    12


     Nombre  Altura  Edad
0     Pedro    12.0    23
2     Diego    45.0    23
8   Antonia     6.0    23
11    Pedro     3.0    23


   Nombre  Altura  Edad
4  Felipe    23.0    24


    Nombre  Altura  Edad
10  Camila     3.0    32


  Nombre  Altura  Edad
7  Sofía     NaN    34


      Nombre  Altura  Edad
3     Arturo    24.0    45
5     Andrés    56.0    45
13  Magnolia    33.0    45


       Nombre  Altura  Edad
9  Bernardita     4.0    67
```

df

	Nombre	Altura	Edad
0	Pedro	12.2	23.0
1	Juan	13.0	12.0
2	Diego	45.0	23.0
3	Arturo	24.0	45.0
4	Felipe	23.0	24.0
5	Andrés	56.0	45.0
6	Juan	6.0	2.0
7	Sofía	23.0	NaN
8	Antonia	6.0	23.0
9	Bernardita	4.0	67.0
10	Camila	3.0	32.0
11	Pedro	3.0	23.0
12	Alvaro	34.0	NaN
13	Magnolia	33.0	45.0

Ejemplo

Para aprender Pandas de forma más práctica, iremos paso a paso usando datos de las crisis bancarias que han sufrido países del continente Africano a lo largo de los años 1860 a 2014. Una copia de los datos se encuentra en webcursos para que los bajen

- Lo que haremos es:
  - Cargar los datos
  - Analizar un dataframe
  - Acceder a datos
  - Realizar filtros
  - Obtener estadísticas

Agrupaciones

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

grupos = df.groupby('country')

agg_df = grupos[['inflation_crises','currency_crises']].agg(['mean'])

print(agg_df)
```

groupby () nos permite crear agrupaciones por un campo

country	inflation_crises mean	currency_crises mean
Algeria	0.164706	0.105882
Angola	0.337662	0.285714
Central African Republic	0.034483	0.034483
Egypt	0.070968	0.051613
Ivory Coast	0.063492	0.015873
Kenya	0.059701	0.134328
Mauritius	0.088235	0.073529
Morocco	0.133333	0.106667
Nigeria	0.200000	0.166667
South Africa	0.008772	0.140351
Tunisia	0.106667	0.133333
Zambia	0.277778	0.263889
Zimbabwe	0.223529	0.233333

Y sobre esos grupos podemos sacar estadísticas de ciertos campos, en este caso de inflation_crisis con el commando .agg

Agrupaciones

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

grupos = df.groupby('country')

estadisticas = grupos['inflation_crises'].agg(['std', 'mean'])

print(estadisticas)
```

groupby () nos permite crear agrupaciones por un campo

Y sobre esos grupos podemos sacar estadísticas de ciertos campos, en este caso de inflation_crisis con el comando .agg

country	std	mean
Algeria	0.373116	0.164706
Angola	0.476014	0.337662
Central African Republic	0.184059	0.034483
Egypt	0.257603	0.070968
Ivory Coast	0.245805	0.063492
Kenya	0.238721	0.059701
Mauritius	0.285746	0.088235
Morocco	0.342224	0.133333
Nigeria	0.403376	0.200000
South Africa	0.093659	0.008772
Tunisia	0.310768	0.106667
Zambia	0.451046	0.277778
Zimbabwe	0.419083	0.223529

min, max, mean, sum,std, sorted, lambda

Ejercicio

Tiempo : 10 minutos

1. Determine el valor promedio y la desviación estándar del índice de precios del consumidor CPI (inflation_annual_cpi) de los países en toda su historia

2. Ahora realice la comparación anterior, pero solamente los bancos en los momentos de tiempo que NO han estado en crisis (banking_crisis)

Recuerde que tiene los siguientes campos:

- inflation_annual_cpi (valores numéricos)
- banking_crisis (crisis o no_crisis)

Ejercicio

Tiempo : 10 minutos

```python
import pandas as pd
df = pd.read_csv('african_crises.csv')

#pregunta 1
group_1 = df.groupby('country')

#determinamos variables estadísticas para cada país
stats_1 = group_1['inflation_annual_cpi'].agg(['std', 'mean'])
print(stats_1)

#pregunta 2
group_2 = df[df['banking_crisis']=='no_crisis'].groupby('country')

#determinamos variables estadísticas para cada país con restricciones
stats_2 = group_2['inflation_annual_cpi'].agg(['std', 'mean'])
print(stats_2)
```

Ejercicio

Tiempo : 10 minutos

1. Determine el número de crisis por país, y luego indique el mayor de todos ellos. Recuerde primero agrupar los datos por país.

Utilice la siguiente columna para conocer su tuvo o no crisis

- banking_crisis (crisis o no_crisis)

Ejercicio

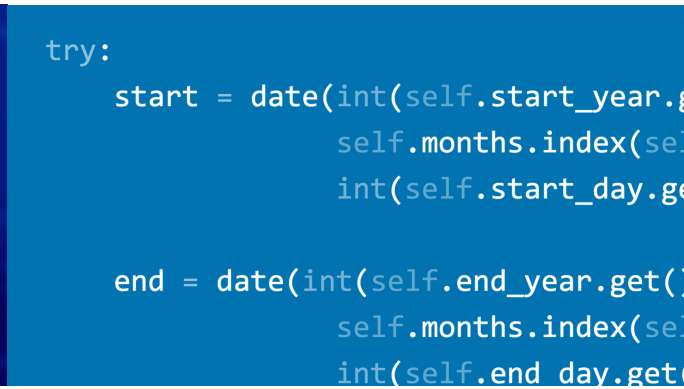
Tiempo : 10 minutos

```python
import pandas as pd
import numpy as np

df = pd.read_csv('data/african_crises.csv')

# agrupamos por país
grp = df.groupby(['country'])

#determiamos aquellos valores por país con crisis
tbl = grp['banking_crisis'].agg([('cont',lambda col:np.sum(col=='crisis'))])

print(f'Número de crisis por país {tbl}')

#buscamos el mayor de toda la tabla
id = pd.Series.argmax(tbl['cont'])

# mostramos los resultados
no_crisis = tbl.iloc[id].cont
print(f'{tbl.iloc[id].name} tiene un máximo de {no_crisis} crisis ')
```

empleamos una función específica en este ejemplo

Temario

- Introducción a Pandas
  - Operaciones básicas de datos
  - Agrupaciones de datos
  - Pivot table
  - CrossTab
  - Gráfica matplotlib
  - Gráfica Plotly



En muchas ocasiones deseamos organizar los datos de una tabla de modo de extraer información significante de los datos a través de operaciones entre las columnas y sus datos. Esto es lo que conocemos como tabla pivote (muy empleado en excel)

Data

	A	B	C	D	E	F
1	Order ID	Product	Category	Amount	Date	Country
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

Preguntas

- *¿Qué país vende más plátanos?*
- *¿Cuál es el monto promedio de ventas de naranjas?*
- *¿Cuál es el monto total de ventas de los vegetales?*
- *¿Cuantas productos han sido vendidos?*
- *¿Qué país ha vendido menos vegetales?*

En muchas ocasiones deseamos organizar los datos de una tabla de modo de extraer información significante de los datos a través de operaciones entre las columnas y sus datos. Esto es lo que conocemos como tabla pivote (muy empleado en excel)

Data

	A	B	C	D	E	F
1	Order ID	Product	Category	Amount	Date	Country
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

Tabla Pivote

- *¿Cuantos productos han sido vendidos?*

	A	B
3	**Etiquetas de fila**	**Suma de Amount**
4	Apple	191257
5	Banana	340295
6	Beans	57281
7	Broccoli	142439
8	Carrots	136945
9	Mango	57079
10	Orange	104438
11	**Total general**	**1029734**

Pivot table

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')
```

Vamos a leer un archivo excel con los datos en la hoja "Sheet1"

pivot-tables.xlsx

	A	B	C	D	E	F
1	**Order ID**	**Product**	**Category**	**Amount**	**Date**	**Country**
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

Pivot (index)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Product')
print(pt)
```

con la opción **index** indicamos la (o las) columna (s) sobre la cual vamos a agrupar las filas.

	A	B	C	D	E	F
1	Order ID	Product	Category	Amount	Date	Country
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

index

Product	Amount	Order ID
Apple	191257	4306
Banana	340295	7648
Beans	57281	1381
Broccoli	142439	2579
Carrots	136945	3174
Mango	57079	918
Orange	104438	2785

Como observamos, las filas quedan **agrupados** los productos, y las **columnas** son los resultados del pivote (en este caso son sumas por defecto)

Pivot (values)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Product', values='Amount')
print(pt)
```

	A	B	C	D	E	F
1	Order ID	Product	Category	Amount	Date	Country
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

index

Product	Amount
Apple	191257
Banana	340295
Beans	57281
Broccoli	142439
Carrots	136945
Mango	57079
Orange	104438

con la opción **values** indicamos la (o las) columna(s) que deseamos utilizar para el cálculo del pivote

Los valores expresados en la columna son sumas (por defecto)

Pivot (aggfunc)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Product',
                    values='Amount',
                    aggfunc=['max', 'min', 'mean'])

print(pt)
```

aggfunc

	max	min	mean
Product	Amount	Amount	Amount
Apple	8892	330	4781.425000
Banana	9543	107	4792.887324
Beans	8416	680	4406.230769
Broccoli	9630	277	5275.518519
Carrots	9127	339	5072.037037
Mango	9333	1641	5189.000000
Orange	9990	220	4351.583333

index

con la opción **aggfunc** indicamos la (o las) los calculos estadísticos sobre una determinada columna

Pivot (columns)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Country',
                        values='Amount',
                        columns='Product').round(2)

print(pt)
```

con la opción **columns** seleccionamos la columna sobre la cual vamos a mostrar los resultados

columns

index

Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange
Australia	5158.5	5272.1	4811.0	8976.5	2702.0	4593.0	2893.33
Canada	4144.5	4825.0		4135.67		3767.0	6643.0
France	5012.06	5156.29	680.0	5341.0	9104.0	7388.0	2256.0
Germany	4541.0	4409.56	4984.17	4649.62	3606.0	8775.0	8887.0
New Zealand	5166.0	5006.25		4390.0			4003.33
United Kingdom	4383.5	6129.71	2550.0	4804.5	5973.57	5600.0	4348.8
United States	4769.17	4133.09	7163.0	6678.75	5628.4	4472.6	3866.5

Utilice variables categóricas para las filas(index) y columnas

Pivot (columns)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Country',
                        values='Amount',
                        columns='Product',
                        fill_value=0).round(2)

print(pt)
```

con la opción **fill_value** definimos un determinado valor para aquellas celdas faltantes

columns

index

Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange
Australia	5158.5	5272.1	4811.0	8976.5	2702.0	4593.0	2893.33
Canada	4144.5	4825.0	0.0	4135.67	0.0	3767.0	6643.0
France	5012.06	5156.29	680.0	5341.0	9104.0	7388.0	2256.0
Germany	4541.0	4409.56	4984.17	4649.62	3606.0	8775.0	8887.0
New Zealand	5166.0	5006.25	0.0	4390.0	0.0	0.0	4003.33
United Kingdom	4383.5	6129.71	2550.0	4804.5	5973.57	5600.0	4348.8
United States	4769.17	4133.09	7163.0	6678.75	5628.4	4472.6	3866.5

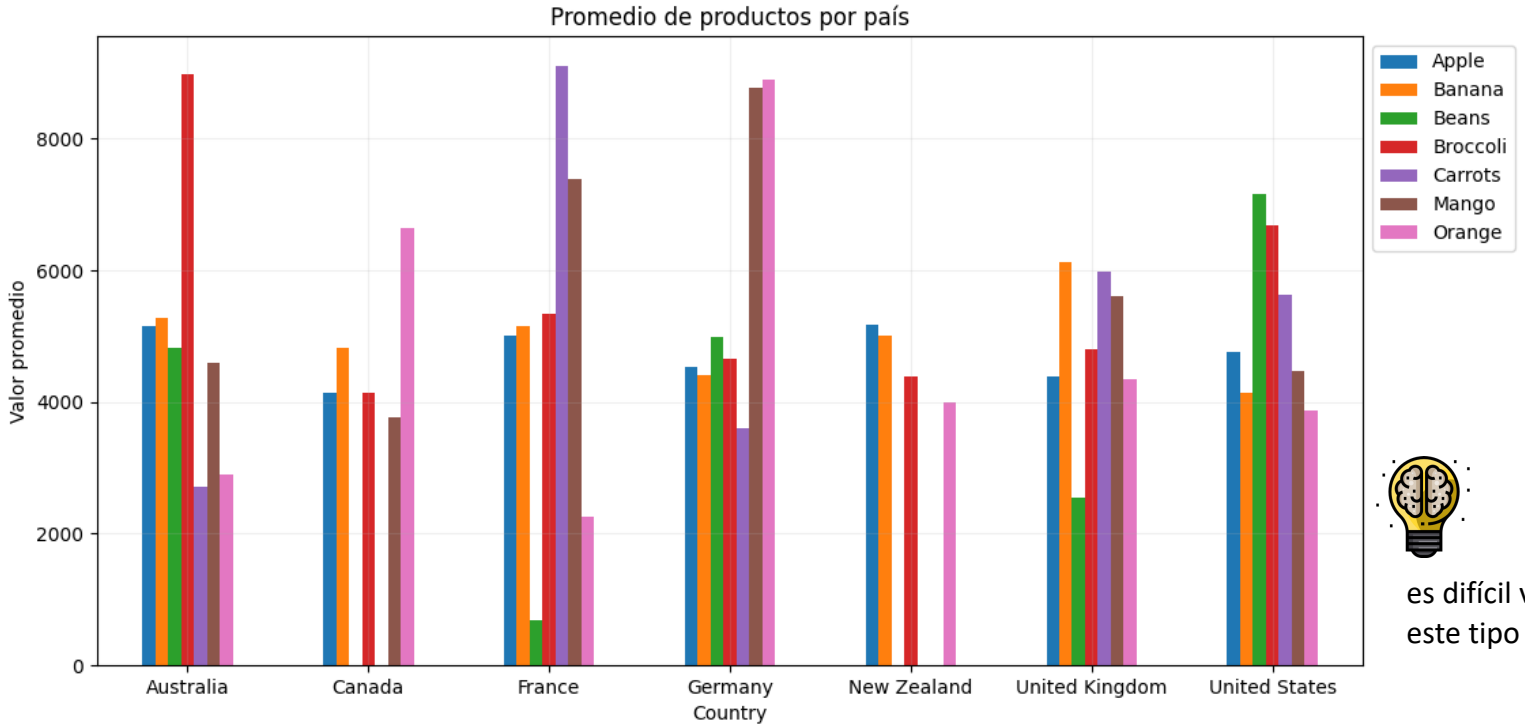
Pivot (plot BAR)

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Country',
                        values='Amount',
                        columns ='Product'
                        fill_value=0).round(2)

pt.plot.bar(figsize=(12,8))

plt.title('Promedio de productos por país')
plt.xticks(rotation=0)
plt.ylabel('Valor promedio')
plt.grid(alpha=0.2)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```

vamos a generar un gráfico de barras a partir de los datos de la tabla pivote

Opciones para los gráficos

Pivot (plot BAR)



es difícil visualizar este tipo de gráfico

index

Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange
Australia	5158.5	5272.1	4811.0	8976.5	2702.0	4593.0	2893.33
Canada	4144.5	4825.0	0.0	4135.67	0.0	3767.0	6643.0
France	5012.06	5156.29	680.0	5341.0	9104.0	7388.0	2256.0
Germany	4541.0	4409.56	4984.17	4649.62	3606.0	8775.0	8887.0
New Zealand	5166.0	5006.25	0.0	4390.0	0.0	0.0	4003.33
United Kingdom	4383.5	6129.71	2550.0	4804.5	5973.57	5600.0	4348.8
United States	4769.17	4133.09	7163.0	6678.75	5628.4	4472.6	3866.5

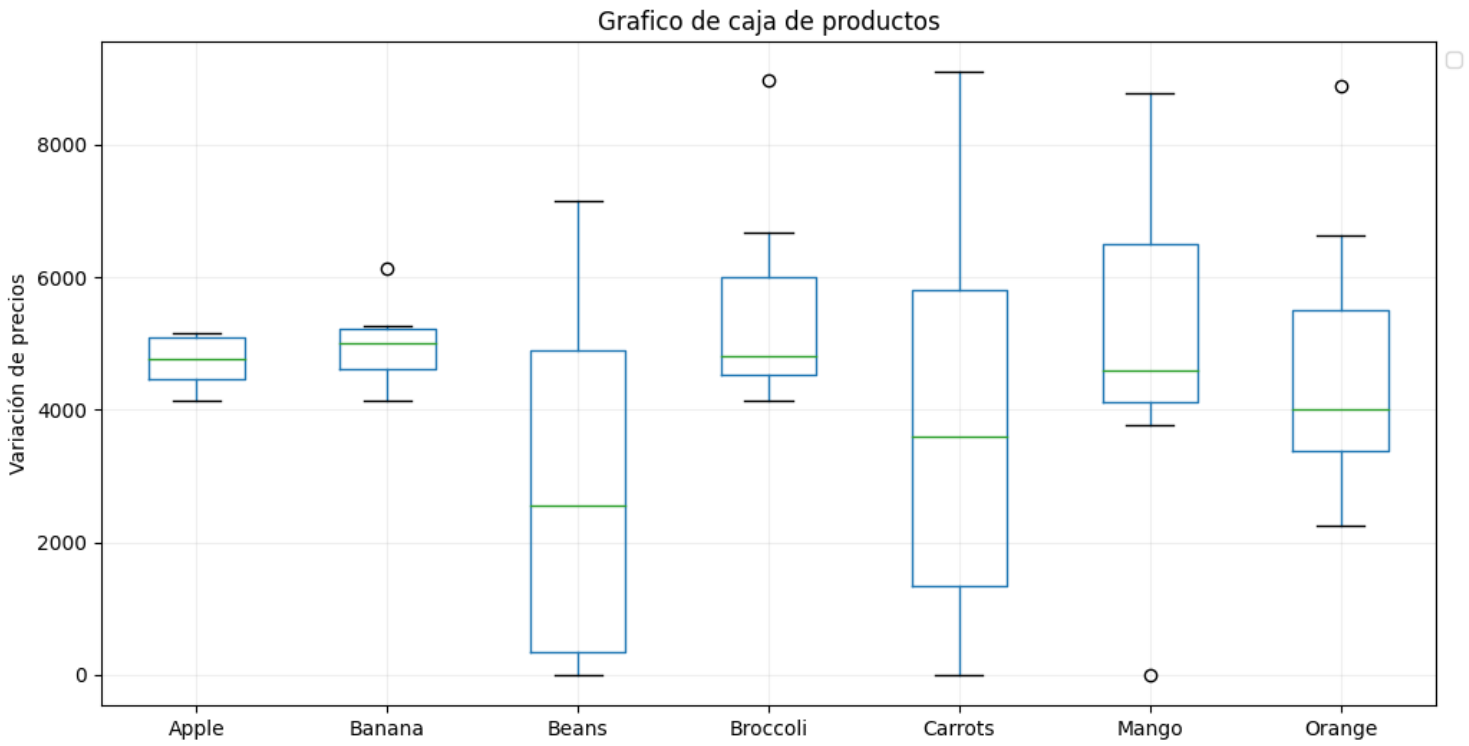
Pivot (plot BOX)

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Country',
                        values='Amount',
                        columns ='Product'
                        fill_value=0).round(2)

pt.plot.box(figsize=(12,8))

plt.title('Promedio de productos por país')
plt.xticks(rotation=0)
plt.ylabel('Valor promedio')
plt.grid(alpha=0.2)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```

vamos a generar un gráfico de dispersión tipo boxplit de la tabla pivote

Opciones para los gráficos

Pivot (plot BOX)



index

Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange
Australia	5158.5	5272.1	4811.0	8976.5	2702.0	4593.0	2893.33
Canada	4144.5	4825.0	0.0	4135.67	0.0	3767.0	6643.0
France	5012.06	5156.29	680.0	5341.0	9104.0	7388.0	2256.0
Germany	4541.0	4409.56	4984.17	4649.62	3606.0	8775.0	8887.0
New Zealand	5166.0	5006.25	0.0	4390.0	0.0	0.0	4003.33
United Kingdom	4383.5	6129.71	2550.0	4804.5	5973.57	5600.0	4348.8
United States	4769.17	4133.09	7163.0	6678.75	5628.4	4472.6	3866.5

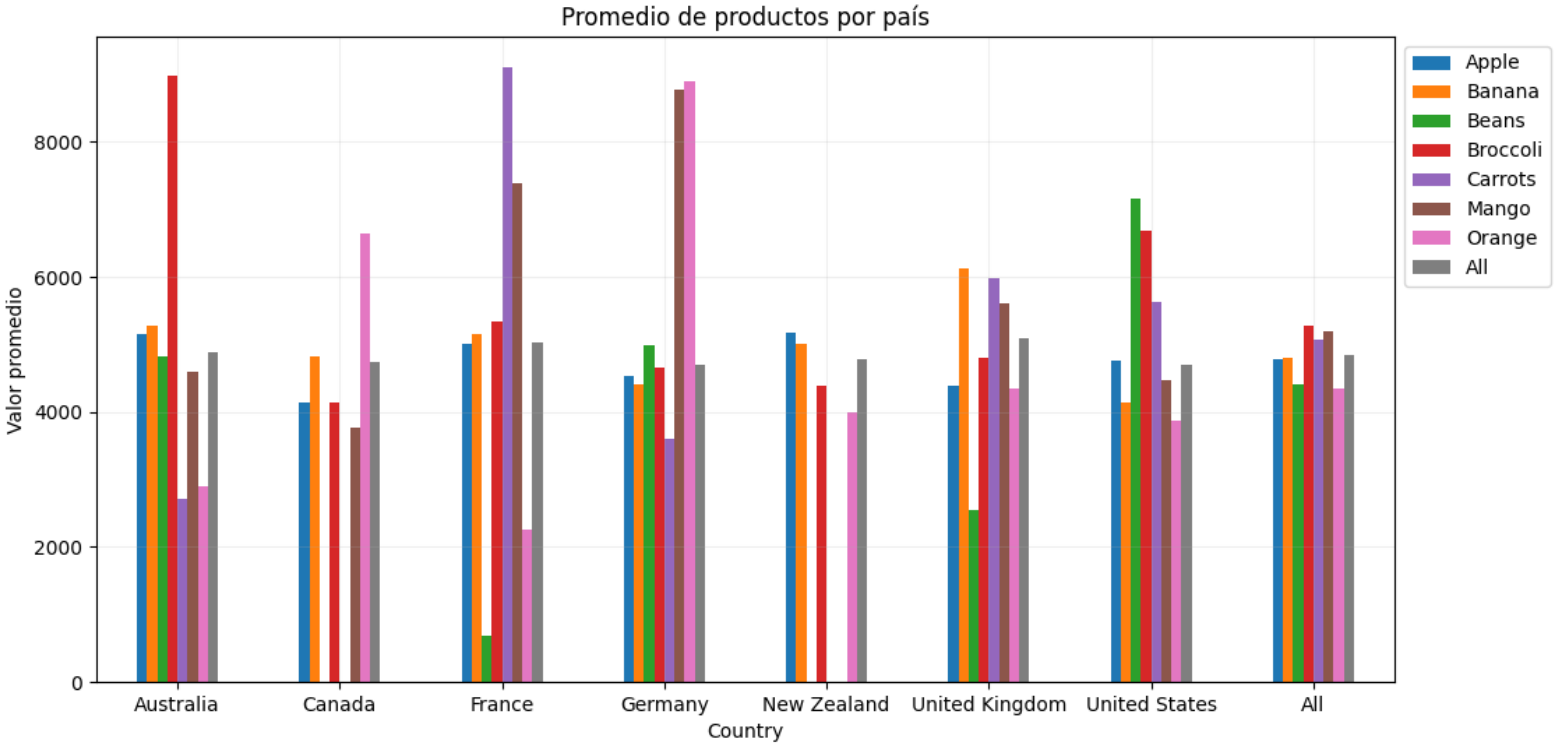
Pivot (margins)

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.pivot_table(df, index='Country',
                        values='Amount',
                        columns ='Product'
                        fill_value=0
                        margins=True).round(2)

pt.plot.bar(figsize=(12,8))
plt.title('Promedio de productos por país')
plt.xticks(rotation=0)
plt.ylabel('Valor promedio')
plt.grid(alpha=0.2)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```

Agrega una columa y fila adicional con los totales

Opciones para los gráficos

Pivot (margins)

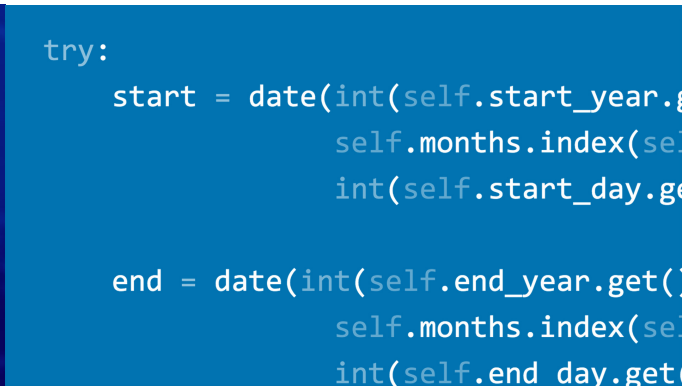


index

Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange	All
Australia	5158.5	5272.1	4811.0	8976.5	2702.0	4593.0	2893.33	4878.26
Canada	4144.5	4825.0	0.0	4135.67	0.0	3767.0	6643.0	4737.25
France	5012.06	5156.29	680.0	5341.0	9104.0	7388.0	2256.0	5037.71
Germany	4541.0	4409.56	4984.17	4649.62	3606.0	8775.0	8887.0	4702.06
New Zealand	5166.0	5006.25	0.0	4390.0	0.0	0.0	4003.33	4770.14
United Kingdom	4383.5	6129.71	2550.0	4804.5	5973.57	5600.0	4348.8	5092.26
United States	4769.17	4133.09	7163.0	6678.75	5628.4	4472.6	3866.5	4686.54
All	4781.42	4792.89	4406.23	5275.52	5072.04	5189.0	4351.58	4834.43

Temario

- Introducción a Pandas
  - Operaciones básicas de datos
  - Agrupaciones de datos
  - Pivot table
  - CrossTab



Pandas. Crosstab

En algunas ocasiones deseamos realizar una tabla cruzada. Una tabla cruzada muestra la relación entre dos variables (factores). Cada casilla representa la frecuencia de aparición entre las dos variables

Data

	A	B	C	D	E	F
1	Order ID	Product	Category	Amount	Date	Country
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

Preguntas

- *¿Cuántos productos por país han sido vendidos?*
- *¿Cuantos productos en total han sido vendidos en United States?*
- *¿Cuántas manzanas han sido vendidas en Alemania?*

Pivot table

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')
```

Vamos a leer un archivo excel con los datos en la hoja "Sheet1"

pivot-tables.xlsx

	A	B	C	D	E	F
1	**Order ID**	**Product**	**Category**	**Amount**	**Date**	**Country**
2	1	Carrots	Vegetables	$4'270	06-01-16	United States
3	2	Broccoli	Vegetables	$8'239	07-01-16	United Kingdom
4	3	Banana	Fruit	$617	08-01-16	United States
5	4	Banana	Fruit	$8'384	10-01-16	Canada
6	5	Beans	Vegetables	$2'626	10-01-16	Germany
7	6	Orange	Fruit	$3'610	11-01-16	United States
8	7	Broccoli	Vegetables	$9'062	11-01-16	Australia
9	8	Banana	Fruit	$6'906	16-01-16	New Zealand
10	9	Apple	Fruit	$2'417	16-01-16	France
11	10	Apple	Fruit	$7'431	16-01-16	Canada
12	11	Banana	Fruit	$8'250	16-01-16	Germany
13	12	Broccoli	Vegetables	$7'012	18-01-16	United States
14	13	Carrots	Vegetables	$1'903	20-01-16	Germany
15	14	Broccoli	Vegetables	$2'824	22-01-16	Canada
16	15	Apple	Fruit	$6'946	24-01-16	France
17	16	Banana	Fruit	$2'320	27-01-16	United Kingdom
18	17	Banana	Fruit	$2'116	28-01-16	United States
19	18	Banana	Fruit	$1'135	30-01-16	United Kingdom
20	19	Broccoli	Vegetables	$3'595	30-01-16	United Kingdom
21	20	Apple	Fruit	$1'161	02-02-16	United States
22	21	Orange	Fruit	$2'256	04-02-16	France
23	22	Banana	Fruit	$1'004	11-02-16	New Zealand
24	23	Banana	Fruit	$3'642	14-02-16	Canada

Pivot (index)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.cross_table(df['Country'], df['Product'])

print(pt)
```

Variable 1

Variable

Product

Country	Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange
	Australia	4	10	3	2	3	2	3
	Canada	6	7	0	3	0	1	3
	France	16	7	1	1	1	1	1
	Germany	2	9	6	8	6	1	1
	New Zealand	2	8	0	1	0	0	3
	United Kingdom	4	7	2	8	7	1	5
	United States	6	23	1	4	10	5	8

Pivot (index)

```
import pandas as pd
df = pd.read_excel('data/pivot-tables.xlsx',sheet_name='Sheet1')

pt = pd.cross_table(df['Country'], df['Product'], margins=True)

print(pt)
```

Agrega una columa y fila adicional con los totales

Product

Country	Country	Apple	Banana	Beans	Broccoli	Carrots	Mango	Orange	All
	Australia	4	10	3	2	3	2	3	27
	Canada	6	7	0	3	0	1	3	20
	France	16	7	1	1	1	1	1	28
	Germany	2	9	6	8	6	1	1	33
	New Zealand	2	8	0	1	0	0	3	14
	United Kingdom	4	7	2	8	7	1	5	34
	United States	6	23	1	4	10	5	8	57
	All	40	71	13	27	27	11	24	213