

Laboratorio No 1

Diego Nicolas Garcia Vargas

Elías Buitrago Bolivar

Seminario Big data y gestión de la información

Universidad ECCI

Julio 2024

Desarrollo

```
[ ] 1 import pandas as pd
    2 import numpy as np
    3 import matplotlib.pyplot as plt
```

Importd de las librerías: Pandas, Numpy y matplotlib

```
[ ] 1 data = {'year': [2010, 2011, 2012,
2         2010, 2011, 2012,
3         2010, 2011, 2012],
4         'team': ['FCBarcelona', 'FCBarcelona',
5         'FCBarcelona', 'RMadrid',
6         'RMadrid', 'RMadrid',
7         'ValenciaCF', 'ValenciaCF',
8         'ValenciaCF'],
9         'wins': [30, 28, 32, 29, 32, 26, 21, 17, 19],
10        'draws': [6, 7, 4, 5, 4, 7, 8, 10, 8],
11        'losses': [2, 3, 2, 4, 2, 5, 9, 11, 11]}
12
13 football = pd.DataFrame(data, columns=['year', 'team', 'wins', 'draws', 'losses'])
14
15 football
```

Se crea un conjunto de datos y este se envía para un dataframe



	year	team	wins	draws	losses
0	2010	FCBarcelona	30	6	2
1	2011	FCBarcelona	28	7	3
2	2012	FCBarcelona	32	4	2
3	2010	RMadrid	29	5	4
4	2011	RMadrid	32	4	2
5	2012	RMadrid	26	7	5
6	2010	ValenciaCF	21	8	9
7	2011	ValenciaCF	17	10	11
8	2012	ValenciaCF	19	8	11

Resultado del dataframe

```
[ ] 1 from google.colab import drive
    2 drive.mount('/content/drive')
```

Función para trabajar sobre el repositorio de drive

```
1 edu = pd.read_csv("/content/drive/MyDrive/data/Lab_1/educ_figdp_1_Data.csv",
2
3 na_values = ': ',
4 usecols = ["TIME", "GEO", "Value"])
5
6 edu
```

Se localiza el archivo y se establece los nombres de columnas

	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95
...
379	2007	Finland	5.90
380	2008	Finland	6.10
381	2009	Finland	6.81
382	2010	Finland	6.85
383	2011	Finland	6.76

384 rows x 3 columns

Resultado de consulta de archivo

```
1 edu[edu['Value'] > 6.5].tail()
```

Se filtra por la columna value para traer aquellos con valor mayor a 6.5



	TIME	GEO	Value
286	2010	Malta	6.74
287	2011	Malta	7.96
381	2009	Finland	6.81
382	2010	Finland	6.85
383	2011	Finland	6.76

Resultado del filtro

```
1 edu[edu["Value"].isnull()].head()
```

Se filtra por la columna value para aquellos con valor nulo




	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
36	2000	Euro area (18 countries)	NaN
37	2001	Euro area (18 countries)	NaN
48	2000	Euro area (17 countries)	NaN

Resultado de filtro

```
1 edu. max(axis = 0)
```

Se filtra por los máximos de cada columna



TIME	2011
GEO	Spain
Value	8.81
dtype:	object

Resultado del filtro

```
[ ] 1 print("Pandas max function:", edu['Value'].max())
    2 print("Python max function:", max(edu['Value']))
```

Filtrando los máximos por columnas específicas

```
⇒ Pandas max function: 8.81
   Python max function: nan
```

Resultado de filtro

```
▶ 1 s = edu["Value"]. apply (np.sqrt)
   2
   3 s.head()
```

Aplicando la raíz cuadrada a cada valor de la columna Value

```
⇒ 0      NaN
   1      NaN
   2    25.0000
   3    25.3009
   4    24.5025
   Name: Value, dtype: float64
```

Resultado

```
[ ] 1 s = edu["Value"]. apply ( lambda d: d**2)
    2
    3 s.head()
```

Aplicando lambda a cada valor de la columna Value

```
⇒ 0      NaN
   1      NaN
   2    25.0000
   3    25.3009
   4    24.5025
   Name: Value, dtype: float64
```

Resultado

```
1 edu['ValueNorm'] = edu['Value']/edu['Value'].max()  
2  
3 edu.tail()
```

Creando una nueva columna en el dataframe a partir de un algoritmo matemático



	TIME	GEO	Value	ValueNorm
379	2007	Finland	5.90	0.669694
380	2008	Finland	6.10	0.692395
381	2009	Finland	6.81	0.772985
382	2010	Finland	6.85	0.777526
383	2011	Finland	6.76	0.767310

Resultado

```
1 edu.drop('ValueNorm', axis = 1, inplace = True)  
2  
3 edu.head()
```

Borrando columnas del dataframe



	TIME	GEO	Value
0	2000	European Union (28 countries)	NaN
1	2001	European Union (28 countries)	NaN
2	2002	European Union (28 countries)	5.00
3	2003	European Union (28 countries)	5.03
4	2004	European Union (28 countries)	4.95

Resultado del borrado de columna

```

1 edu.sort_values(by = 'Value', ascending = False ,
2
3 inplace = True)
4
5 edu.head()

```

Realizando ordenamiento de los datos

	TIME	GEO	Value
130	2010	Denmark	8.81
131	2011	Denmark	8.75
129	2009	Denmark	8.74
121	2001	Denmark	8.44
122	2002	Denmark	8.44

Resultado del ordenamiento

```

1 group = edu[["GEO", "Value"]].groupby('GEO').mean()
2 group.head()

```

Se realiza agrupamiento por la columna GEO

	Value
GEO	
Austria	5.618333
Belgium	6.189091
Bulgaria	4.093333
Cyprus	7.023333
Czech Republic	4.168333

Resultado de agrupamiento

```
1 filtered_data = edu[edu["TIME"] > 2005]
2
3 pivedu = pd. pivot_table( filtered_data , values = 'Value',
4
5 index = ['GEO'] ,
6 columns = ['TIME'])
7
8 pivedu.head()
```

Filtrando la tabla para los años mayores a 2005 y pivoteando las columnas por TIME y GEO

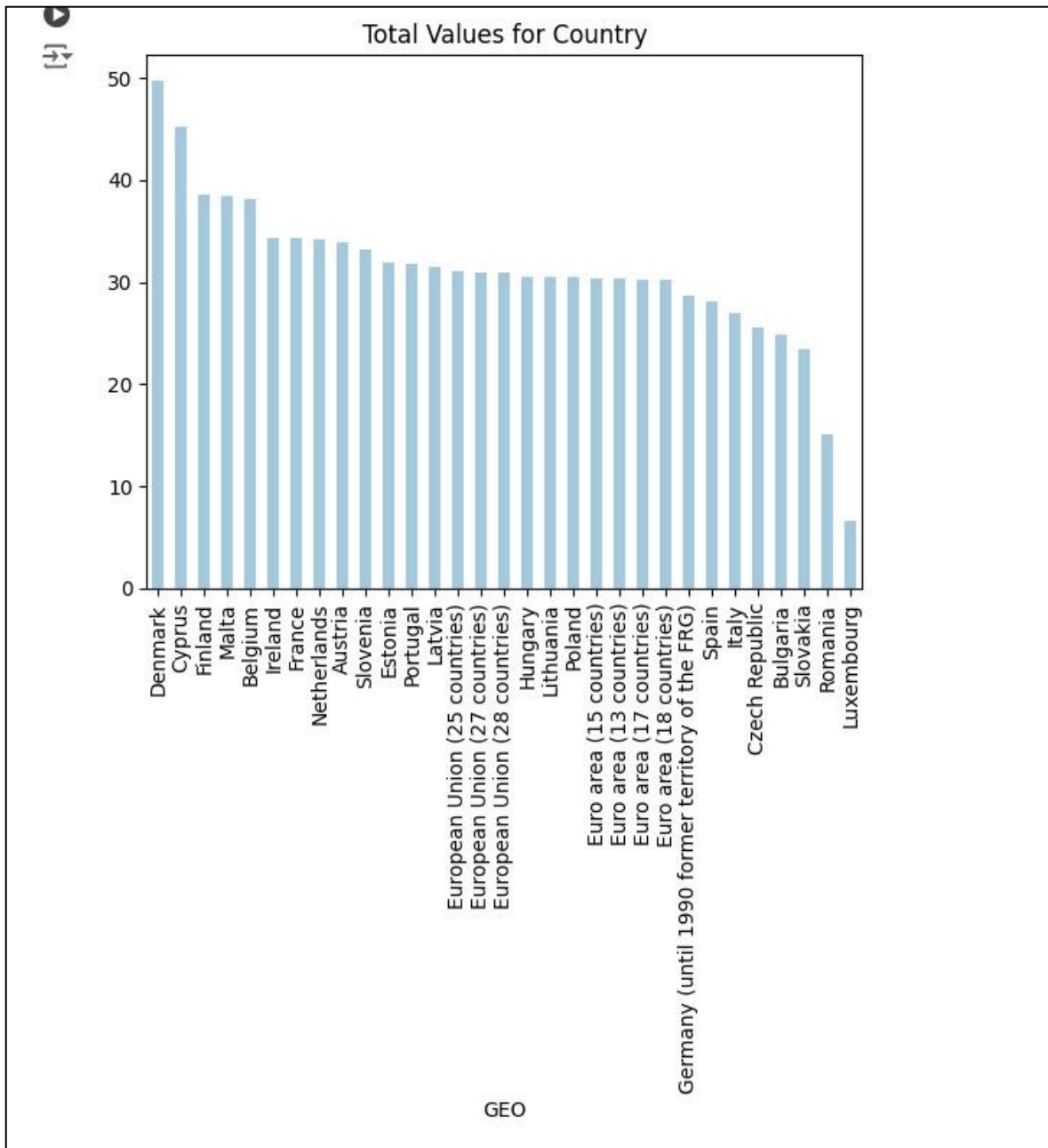


	TIME	2006	2007	2008	2009	2010	2011
	GEO						
	Austria	5.40	5.33	5.47	5.98	5.91	5.80
	Belgium	5.98	6.00	6.43	6.57	6.58	6.55
	Bulgaria	4.04	3.88	4.44	4.58	4.10	3.82
	Cyprus	7.02	6.95	7.45	7.98	7.92	7.87
	Czech Republic	4.42	4.05	3.92	4.36	4.25	4.51

Resultado de filtro y pivoteo

```
1 totalSum = pivedu. sum(axis = 1).sort_values(ascending = False)
2 totalSum. plot(kind = 'bar', style = 'b', alpha = 0.4,
3 title = "Total Values for Country")
```

Dando valor a las propiedades de graficas



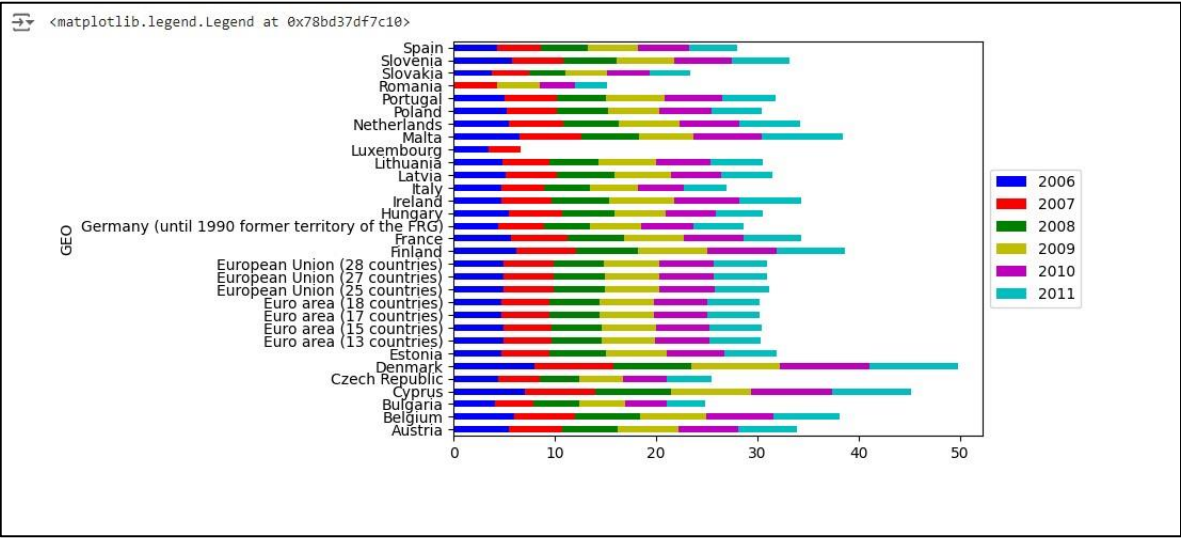
Grafica

```

1 my_colors = ['b', 'r', 'g', 'y', 'm', 'c']
2 ax = pivedu. plot(kind = 'barh',
3 stacked = True ,
4 color = my_colors)
5
6 ax.legend(loc = 'center left', bbox_to_anchor = (1, .5))

```

Otras propiedades de graficas



Grafica

Conclusiones

Usar las librerías Pandas, NumPy y Matplotlib.pyplot en la analítica de datos proporciona una poderosa combinación de herramientas que facilita la manipulación, análisis y visualización de grandes conjuntos de datos de manera eficiente y efectiva. En conjunto, estas librerías permiten a los analistas de datos llevar a cabo análisis exploratorios, preparar datos para modelos de machine learning, y comunicar sus hallazgos de manera clara y efectiva mediante visualizaciones comprensibles.