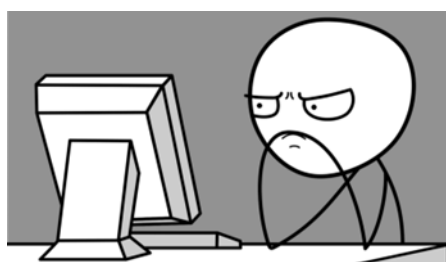
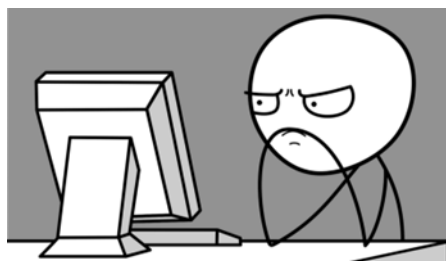


12_Ejercicios_Series

November 28, 2023



0.1 EJERCICIOS: Exploración

Para ejercitarte y afianzar lo aprendido sobre Exploración de DataFrames, completa los siguientes ejercicios. Recuerda que puedes necesitar de algunos datos que están en el directorio data que acompaña al notebook (búscalo en el repositorio de ejercicios)

La solución a los mismos se publicarán en el repo de ejercicios a lo largo del curso.

0.1.1 Ejercicio 0:

Importa pandas "como debe ser" y cualquier otra librería que emplees en el notebook (también "como debe ser" si es necesario)

```
[103]: import numpy as np
import pandas as pd
```

Ahora ejecuta la celda para crear un DataFrame con los datos de la Eurocopa de 2012

```
[104]: pd.set_option('display.max_columns', None) # con None Quita el límite a la
↳ visualización de columnas por celda
df_euro12 = pd.read_csv('./data/Euro_2012_stats_TEAM.csv')
```

0.1.2 Ejercicio 1

Muestra las primeras 7 filas de df_euro12. Luego las 12 últimas.

```
[105]: df_euro12.head(7)
```

```
[105]:
```

	Team	Goals	Shots on target	Shots off target	Shooting Accuracy \
0	Croatia	4	13	12	51.9%
1	Czech Republic	4	13	18	41.9%
2	Denmark	4	10	10	50.0%
3	England	5	11	18	50.0%
4	France	3	22	24	37.9%
5	Germany	10	32	32	47.8%
6	Greece	5	8	18	30.7%

	% Goals-to-shots	Total shots (inc. Blocked)	Hit Woodwork	Penalty goals \
0	16.0%	32	0	0
1	12.9%	39	0	0
2	20.0%	27	1	0
3	17.2%	40	0	0
4	6.5%	65	1	0
5	15.6%	80	2	1
6	19.2%	32	1	1

	Penalties not scored	Headed goals	Passes	Passes completed \
0	0	2	1076	828
1	0	0	1565	1223
2	0	3	1298	1082
3	0	3	1488	1200
4	0	0	2066	1803
5	0	2	2774	2427
6	1	0	1187	911

	Passing Accuracy	Touches	Crosses	Dribbles	Corners Taken	Tackles \
0	76.9%	1706	60	42	14	49
1	78.1%	2358	46	68	21	62
2	83.3%	1873	43	32	16	40
3	80.6%	2440	58	60	16	86
4	87.2%	2909	55	76	28	71
5	87.4%	3761	101	60	35	91
6	76.7%	2016	52	53	10	65

	Clearances	Interceptions	Clearances off line	Clean Sheets	Blocks \
0	83	56	NaN	0	10
1	98	37	2.0	1	10
2	61	59	0.0	1	10
3	106	72	1.0	2	29
4	76	58	0.0	1	7

5	73	69	0.0	1	11
6	123	87	0.0	1	23

	Goals conceded	Saves made	Saves-to-shots ratio	Fouls Won	Fouls Conceded \
0	3	13	81.3%	41	62
1	6	9	60.1%	53	73
2	5	10	66.7%	25	38
3	3	22	88.1%	43	45
4	5	6	54.6%	36	51
5	6	10	62.6%	63	49
6	7	13	65.1%	67	48

	Offsides	Yellow Cards	Red Cards	Subs on	Subs off	Players Used
0	2	9	0	9	9	16
1	8	7	0	11	11	19
2	8	4	0	7	7	15
3	6	5	0	11	11	16
4	5	6	0	11	11	19
5	12	4	0	15	15	17
6	12	9	1	12	12	20

```
[106]: df_euro12.tail(12)
```

```
[106]:
```

	Team	Goals	Shots on target	Shots off target \
4	France	3	22	24
5	Germany	10	32	32
6	Greece	5	8	18
7	Italy	6	34	45
8	Netherlands	2	12	36
9	Poland	2	15	23
10	Portugal	6	22	42
11	Republic of Ireland	1	7	12
12	Russia	5	9	31
13	Spain	12	42	33
14	Sweden	5	17	19
15	Ukraine	2	7	26

	Shooting Accuracy %	Goals-to-shots	Total shots (inc. Blocked) \
4	37.9%	6.5%	65
5	47.8%	15.6%	80
6	30.7%	19.2%	32
7	43.0%	7.5%	110
8	25.0%	4.1%	60
9	39.4%	5.2%	48
10	34.3%	9.3%	82
11	36.8%	5.2%	28
12	22.5%	12.5%	59

13	55.9%	16.0%	100
14	47.2%	13.8%	39
15	21.2%	6.0%	38

	Hit Woodwork	Penalty goals	Penalties not scored	Headed goals	Passes \
4	1	0	0	0	2066
5	2	1	0	2	2774
6	1	1	1	0	1187
7	2	0	0	2	3016
8	2	0	0	0	1556
9	0	0	0	1	1059
10	6	0	0	2	1891
11	0	0	0	1	851
12	2	0	0	1	1602
13	0	1	0	2	4317
14	3	0	0	1	1192
15	0	0	0	2	1276

	Passes completed	Passing Accuracy	Touches	Crosses	Dribbles \
4	1803	87.2%	2909	55	76
5	2427	87.4%	3761	101	60
6	911	76.7%	2016	52	53
7	2531	83.9%	4363	75	75
8	1381	88.7%	2163	50	49
9	852	80.4%	1724	55	39
10	1461	77.2%	2958	91	64
11	606	71.2%	1433	43	18
12	1345	83.9%	2278	40	40
13	3820	88.4%	5585	69	106
14	965	80.9%	1806	44	29
15	1043	81.7%	1894	33	26

	Corners Taken	Tackles	Clearances	Interceptions	Clearances off line \
4	28	71	76	58	0.0
5	35	91	73	69	0.0
6	10	65	123	87	0.0
7	30	98	137	136	1.0
8	22	34	41	41	0.0
9	14	67	87	62	0.0
10	41	78	92	86	0.0
11	8	45	78	43	1.0
12	21	65	74	58	0.0
13	44	122	102	79	0.0
14	7	56	54	45	0.0
15	18	65	97	29	0.0

Clean Sheets	Blocks	Goals conceded	Saves made	Saves-to-shots ratio \
--------------	--------	----------------	------------	------------------------

4	1	7	5	6	54.6%
5	1	11	6	10	62.6%
6	1	23	7	13	65.1%
7	2	18	7	20	74.1%
8	0	9	5	12	70.6%
9	0	8	3	6	66.7%
10	2	11	4	10	71.5%
11	0	23	9	17	65.4%
12	0	8	3	10	77.0%
13	5	8	1	15	93.8%
14	1	12	5	8	61.6%
15	0	4	4	13	76.5%

	Fouls Won	Fouls Conceded	Offsides	Yellow Cards	Red Cards	Subs on \
4	36	51	5	6	0	11
5	63	49	12	4	0	15
6	67	48	12	9	1	12
7	101	89	16	16	0	18
8	35	30	3	5	0	7
9	48	56	3	7	1	7
10	73	90	10	12	0	14
11	43	51	11	6	1	10
12	34	43	4	6	0	7
13	102	83	19	11	0	17
14	35	51	7	7	0	9
15	48	31	4	5	0	9

	Subs off	Players Used
4	11	19
5	15	17
6	12	20
7	18	19
8	7	15
9	7	17
10	14	16
11	10	17
12	7	16
13	17	18
14	9	18
15	9	18

[]:

0.1.3 Ejercicio 2

Écha un primer vistazo, mostrando las columnas, cuantos valores no nulos tienen y su tipo. Utiliza un sólo comando para todo.

```
[111]: df_euro12.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 35 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Team                                  16 non-null     object
1   Goals                                16 non-null     int64
2   Shots on target                      16 non-null     int64
3   Shots off target                    16 non-null     int64
4   Shooting Accuracy                   16 non-null     object
5   % Goals-to-shots                    16 non-null     object
6   Total shots (inc. Blocked)          16 non-null     int64
7   Hit Woodwork                        16 non-null     int64
8   Penalty goals                       16 non-null     int64
9   Penalties not scored                16 non-null     int64
10  Headed goals                        16 non-null     int64
11  Passes                              16 non-null     int64
12  Passes completed                    16 non-null     int64
13  Passing Accuracy                    16 non-null     object
14  Touches                            16 non-null     int64
15  Crosses                            16 non-null     int64
16  Dribbles                           16 non-null     int64
17  Corners Taken                      16 non-null     int64
18  Tackles                            16 non-null     int64
19  Clearances                         16 non-null     int64
20  Interceptions                      16 non-null     int64
21  Clearances off line                 15 non-null     float64
22  Clean Sheets                       16 non-null     int64
23  Blocks                             16 non-null     int64
24  Goals conceded                     16 non-null     int64
25  Saves made                         16 non-null     int64
26  Saves-to-shots ratio                16 non-null     object
27  Fouls Won                          16 non-null     int64
28  Fouls Conceded                     16 non-null     int64
29  Offsides                           16 non-null     int64
30  Yellow Cards                       16 non-null     int64
31  Red Cards                          16 non-null     int64
32  Subs on                            16 non-null     int64
33  Subs off                           16 non-null     int64
34  Players Used                       16 non-null     int64
dtypes: float64(1), int64(29), object(5)
memory usage: 4.5+ KB
```

0.1.4 Ejercicio 3

Haz un primer análisis de las columnas numéricas, mostrando al menos la media el máximo y el mínimo.

```
[112]: primerA=round(df_euro12.describe(),2)
primerA
```

```
[112]:      Goals  Shots on target  Shots off target  Total shots (inc. Blocked) \
count    16.00             16.00             16.00             16.00
mean      4.75             17.12             24.94             54.94
std       2.89             10.58             10.68             26.07
min       1.00              7.00             10.00             27.00
25%       2.75              9.75             18.00             36.50
50%       4.50             13.00             23.50             44.00
75%       5.25             22.00             32.25             68.75
max      12.00            42.00             45.00            110.00
```

```
      Hit Woodwork  Penalty goals  Penalties not scored  Headed goals \
count           16.00           16.00           16.00           16.00
mean            1.25            0.19            0.06            1.38
std             1.61            0.40            0.25            1.02
min             0.00            0.00            0.00            0.00
25%             0.00            0.00            0.00            0.75
50%             1.00            0.00            0.00            1.50
75%             2.00            0.00            0.00            2.00
max             6.00            1.00            1.00            3.00
```

```
      Passes  Passes completed  Touches  Crosses  Dribbles  Corners Taken \
count    16.00             16.00    16.00    16.00    16.00             16.00
mean   1763.38            1467.38  2579.19    57.19    52.31             21.56
std    906.18             827.58  1122.16    18.62    22.70             11.24
min    851.00             606.00  1433.00    33.00    18.00              7.00
25%   1190.75             951.50  1856.25    43.75    37.25             14.00
50%   1522.00            1211.50  2220.50    53.50    51.00             19.50
75%   1934.75            1546.50  2921.25    62.25    65.00             28.50
max   4317.00            3820.00  5585.00   101.00   106.00             44.00
```

```
      Tackles  Clearances  Interceptions  Clearances off line  Clean Sheets \
count    16.00           16.00           16.00             15.00           16.00
mean     68.38           86.38           63.56              0.33            1.06
std     22.82           24.61           25.77              0.62            1.29
min     34.00           41.00           29.00              0.00            0.00
25%     54.25           73.75           44.50              0.00            0.00
50%     65.00           85.00           58.50              0.00            1.00
75%     80.00           99.00           73.75              0.50            1.25
max    122.00          137.00          136.00              2.00            5.00
```

	Blocks	Goals conceded	Saves made	Fouls Won	Fouls Conceded	\
count	16.00	16.00	16.00	16.00	16.00	
mean	12.56	4.75	12.12	52.94	55.62	
std	6.94	1.98	4.57	22.92	18.97	
min	4.00	1.00	6.00	25.00	30.00	
25%	8.00	3.00	9.75	35.75	44.50	
50%	10.00	5.00	11.00	45.50	51.00	
75%	13.50	6.00	13.50	64.00	64.75	
max	29.00	9.00	22.00	102.00	90.00	

	Offsides	Yellow Cards	Red Cards	Subs on	Subs off	Players Used
count	16.00	16.00	16.00	16.00	16.00	16.00
mean	8.12	7.44	0.19	10.88	10.88	17.25
std	4.91	3.27	0.40	3.54	3.54	1.53
min	2.00	4.00	0.00	7.00	7.00	15.00
25%	4.00	5.00	0.00	8.50	8.50	16.00
50%	7.50	6.50	0.00	10.50	10.50	17.00
75%	11.25	9.00	0.00	12.50	12.50	18.25
max	19.00	16.00	1.00	18.00	18.00	20.00

0.1.5 Ejercicio 4

¿Cuántos equipos participaron en la Euro2012?

```
[113]: display(len(df_euro12["Team"]))
```

16

0.1.6 Ejercicio 5

Muestra por pantalla las selecciones y los goles que marcó cada una

```
[10]: df_euro12[["Team", "Goals"]]
```

```
[10]:
```

	Team	Goals
0	Croatia	4
1	Czech Republic	4
2	Denmark	4
3	England	5
4	France	3
5	Germany	10
6	Greece	5
7	Italy	6
8	Netherlands	2
9	Poland	2
10	Portugal	6
11	Republic of Ireland	1
12	Russia	5

13	Spain	12
14	Sweden	5
15	Ukraine	2

0.1.7 Ejercicio 6

¿Cuál es el número de columnas del conjunto de datos?

```
[11]: print(len(df_euro12.columns))
```

35

0.1.8 Ejercicio 7

Quédate sólo las columnas Equipo, Tarjetas Amarillas y Tarjetas Rojas y asígnalas a un dataframe de datos llamado discipline

```
[115]: # Seleccionar las columnas de interés
columnas = df_euro12[['Team', 'Yellow Cards', 'Red Cards']]
#print(columnas)
# crear el nuevo DF
df_discipline = pd.DataFrame(columnas)
df_discipline
```

```
[115]:
```

	Team	Yellow Cards	Red Cards
0	Croatia	9	0
1	Czech Republic	7	0
2	Denmark	4	0
3	England	5	0
4	France	6	0
5	Germany	4	0
6	Greece	9	1
7	Italy	16	0
8	Netherlands	5	0
9	Poland	7	1
10	Portugal	12	0
11	Republic of Ireland	6	1
12	Russia	6	0
13	Spain	11	0
14	Sweden	7	0
15	Ukraine	5	0

0.1.9 Ejercicio 8

Ordena los equipos por tarjetas rojas y luego por tarjetas amarillas. Recuerda el método `sort_values` de la práctica final de la unidad anterior.

```
[116]:
```

```
# ordeno en orden ascendente ambas columnas(valor predeterminado de ascending_
↳=True)
x_rojas =df_discipline.sort_values("Red Cards")
display(x_rojas)
print("\n")
x_amarillas = df_discipline.sort_values("Yellow Cards")
display(x_amarillas)
```

	Team	Yellow Cards	Red Cards
0	Croatia	9	0
1	Czech Republic	7	0
2	Denmark	4	0
3	England	5	0
4	France	6	0
5	Germany	4	0
7	Italy	16	0
8	Netherlands	5	0
10	Portugal	12	0
12	Russia	6	0
13	Spain	11	0
14	Sweden	7	0
15	Ukraine	5	0
6	Greece	9	1
9	Poland	7	1
11	Republic of Ireland	6	1

	Team	Yellow Cards	Red Cards
2	Denmark	4	0
5	Germany	4	0
3	England	5	0
8	Netherlands	5	0
15	Ukraine	5	0
4	France	6	0
11	Republic of Ireland	6	1
12	Russia	6	0
1	Czech Republic	7	0
9	Poland	7	1
14	Sweden	7	0
0	Croatia	9	0
6	Greece	9	1
13	Spain	11	0
10	Portugal	12	0
7	Italy	16	0

0.1.10 Ejercicio 9

Investiga como podrías hacer para que ordenase por tarjetas rojas de forma decreciente y creciente por tarjetas amarillas.

```
[118]: #orden decreciente cambiando valor predeterminado de ascending
x_rojas =df_discipline.sort_values(["Red Cards"], ascending = False)
print(x_rojas)
print("\n")
#crecientes que es lo mismo que poner ascending =True
x_amarillas = df_discipline.sort_values("Yellow Cards")
display(x_amarillas)
```

	Team	Yellow Cards	Red Cards
6	Greece	9	1
9	Poland	7	1
11	Republic of Ireland	6	1
0	Croatia	9	0
1	Czech Republic	7	0
2	Denmark	4	0
3	England	5	0
4	France	6	0
5	Germany	4	0
7	Italy	16	0
8	Netherlands	5	0
10	Portugal	12	0
12	Russia	6	0
13	Spain	11	0
14	Sweden	7	0
15	Ukraine	5	0

	Team	Yellow Cards	Red Cards
2	Denmark	4	0
5	Germany	4	0
3	England	5	0
8	Netherlands	5	0
15	Ukraine	5	0
4	France	6	0
11	Republic of Ireland	6	1
12	Russia	6	0
1	Czech Republic	7	0
9	Poland	7	1
14	Sweden	7	0
0	Croatia	9	0
6	Greece	9	1
13	Spain	11	0
10	Portugal	12	0

7	Italy	16	0
---	-------	----	---

0.1.11 Ejercicio 10

Calcular el número medio de tarjetas amarillas por equipo (pista método mean)

```
[140]: # media por tarjetas amarillas en total
media_a = df_euro12["Yellow Cards"].mean()
print(media_a)
# Creo la nueva columna con la media total de tarjetas amarillas por equipo en %
df_euro12["Y_Cards Mean(%)"] = round((df_euro12["Yellow Cards"] / media_a) * 100, 2)
# print(df_euro12["Y_Cards Mean(%)"])

# muestro equipos con la nueva columna
resultado = df_euro12[["Team", "Yellow Cards", "Y_Cards Mean(%)"]]
display(resultado)
```

7.4375

	Team	Yellow Cards	Y_Cards Mean(%)
0	Croatia	9	121.01
1	Czech Republic	7	94.12
2	Denmark	4	53.78
3	England	5	67.23
4	France	6	80.67
5	Germany	4	53.78
6	Greece	9	121.01
7	Italy	16	215.13
8	Netherlands	5	67.23
9	Poland	7	94.12
10	Portugal	12	161.34
11	Republic of Ireland	6	80.67
12	Russia	6	80.67
13	Spain	11	147.90
14	Sweden	7	94.12
15	Ukraine	5	67.23

[]:

0.1.12 Ejercicio 11

Muestra el nombre, número de goles y la precisión en el tiro de los equipos que marcaron más de 6 goles

```
[142]: # precision de tiro en %
df_euro12["Shooting Accuracy(%)"] = round((df_euro12["Shots on target"] /
df_euro12["Shots off target"]) * 100, 2)
# print(df_euro12["Shooting Accuracy(%)"])
```

```
#muestro resultado con las columna nueva d ela ppresion por equipo
res =df_euro12[["Team","Shots on target", "Shots off target", "Shooting_
↳Accuracy(%)"] ]
display(res)
```

	Team	Shots on target	Shots off target	\
0	Croatia	13	12	
1	Czech Republic	13	18	
2	Denmark	10	10	
3	England	11	18	
4	France	22	24	
5	Germany	32	32	
6	Greece	8	18	
7	Italy	34	45	
8	Netherlands	12	36	
9	Poland	15	23	
10	Portugal	22	42	
11	Republic of Ireland	7	12	
12	Russia	9	31	
13	Spain	42	33	
14	Sweden	17	19	
15	Ukraine	7	26	

	Shooting Accuracy(%)
0	108.33
1	72.22
2	100.00
3	61.11
4	91.67
5	100.00
6	44.44
7	75.56
8	33.33
9	65.22
10	52.38
11	58.33
12	29.03
13	127.27
14	89.47
15	26.92

0.1.13 Ejercicio 12

Presenta sólo la Shooting Accuracy de Inglaterra, Italia y España

```
[82]: # saco el subconjunto total de las dos columnas
Total =df_euro12[["Team","Shooting Accuracy(%)"] ]
#display(Total)
```

```
#hago variable con sus indices
indices = [3,7,13]
#hago una serie panda para que me extraiga los datos con los indices indicados
↳del subconjunto
Subtotal = Total.loc[indices]
display(Subtotal)
```

	Team	Shooting Accuracy(%)
3	England	61.11
7	Italy	75.56
13	Spain	127.27

0.1.14 Ejercicio 13

Cambia el nombre de las columnas Goal, Yellow Cards, Red Cards por su traducción al castellano.

```
[ ]:
```

```
[100]: df_euro12.rename(columns= {"Goals": "Goles", "Yellow Cards": "Tarjetas
↳Amarillas", "Red Cards": "Tarjetas Rojas"}, inplace =True)
display(df_euro12)
```

	Team	Goles	Shots on target	Shots off target	\
0	Croatia	4	13	12	
1	Czech Republic	4	13	18	
2	Denmark	4	10	10	
3	England	5	11	18	
4	France	3	22	24	
5	Germany	10	32	32	
6	Greece	5	8	18	
7	Italy	6	34	45	
8	Netherlands	2	12	36	
9	Poland	2	15	23	
10	Portugal	6	22	42	
11	Republic of Ireland	1	7	12	
12	Russia	5	9	31	
13	Spain	12	42	33	
14	Sweden	5	17	19	
15	Ukraine	2	7	26	

	Shooting Accuracy %	Goals-to-shots	Total shots (inc. Blocked)	\
0	51.9%	16.0%	32	
1	41.9%	12.9%	39	
2	50.0%	20.0%	27	
3	50.0%	17.2%	40	
4	37.9%	6.5%	65	
5	47.8%	15.6%	80	
6	30.7%	19.2%	32	

7	43.0%	7.5%	110
8	25.0%	4.1%	60
9	39.4%	5.2%	48
10	34.3%	9.3%	82
11	36.8%	5.2%	28
12	22.5%	12.5%	59
13	55.9%	16.0%	100
14	47.2%	13.8%	39
15	21.2%	6.0%	38

	Hit Woodwork	Penalty goals	Penalties not scored	Headed goals	Passes \
0	0	0	0	2	1076
1	0	0	0	0	1565
2	1	0	0	3	1298
3	0	0	0	3	1488
4	1	0	0	0	2066
5	2	1	0	2	2774
6	1	1	1	0	1187
7	2	0	0	2	3016
8	2	0	0	0	1556
9	0	0	0	1	1059
10	6	0	0	2	1891
11	0	0	0	1	851
12	2	0	0	1	1602
13	0	1	0	2	4317
14	3	0	0	1	1192
15	0	0	0	2	1276

	Passes completed	Passing Accuracy	Touches	Crosses	Dribbles \
0	828	76.9%	1706	60	42
1	1223	78.1%	2358	46	68
2	1082	83.3%	1873	43	32
3	1200	80.6%	2440	58	60
4	1803	87.2%	2909	55	76
5	2427	87.4%	3761	101	60
6	911	76.7%	2016	52	53
7	2531	83.9%	4363	75	75
8	1381	88.7%	2163	50	49
9	852	80.4%	1724	55	39
10	1461	77.2%	2958	91	64
11	606	71.2%	1433	43	18
12	1345	83.9%	2278	40	40
13	3820	88.4%	5585	69	106
14	965	80.9%	1806	44	29
15	1043	81.7%	1894	33	26

	Corners Taken	Tackles	Clearances	Interceptions	Clearances off line \
0	14	49	83	56	NaN

1	21	62	98	37	2.0
2	16	40	61	59	0.0
3	16	86	106	72	1.0
4	28	71	76	58	0.0
5	35	91	73	69	0.0
6	10	65	123	87	0.0
7	30	98	137	136	1.0
8	22	34	41	41	0.0
9	14	67	87	62	0.0
10	41	78	92	86	0.0
11	8	45	78	43	1.0
12	21	65	74	58	0.0
13	44	122	102	79	0.0
14	7	56	54	45	0.0
15	18	65	97	29	0.0

	Clean Sheets	Blocks	Goals conceded	Saves made	Saves-to-shots ratio \
0	0	10	3	13	81.3%
1	1	10	6	9	60.1%
2	1	10	5	10	66.7%
3	2	29	3	22	88.1%
4	1	7	5	6	54.6%
5	1	11	6	10	62.6%
6	1	23	7	13	65.1%
7	2	18	7	20	74.1%
8	0	9	5	12	70.6%
9	0	8	3	6	66.7%
10	2	11	4	10	71.5%
11	0	23	9	17	65.4%
12	0	8	3	10	77.0%
13	5	8	1	15	93.8%
14	1	12	5	8	61.6%
15	0	4	4	13	76.5%

	Fouls Won	Fouls Conceded	Offsides	Tarjetas Amarillas	Tarjetas Rojas \
0	41	62	2	9	0
1	53	73	8	7	0
2	25	38	8	4	0
3	43	45	6	5	0
4	36	51	5	6	0
5	63	49	12	4	0
6	67	48	12	9	1
7	101	89	16	16	0
8	35	30	3	5	0
9	48	56	3	7	1
10	73	90	10	12	0
11	43	51	11	6	1
12	34	43	4	6	0

13	102	83	19	11	0
14	35	51	7	7	0
15	48	31	4	5	0

	Subs on	Subs off	Players Used	Y_Cards	Mean(%)	Total Accur.	Target(%) \
0	9	9	16		121.01		108.33
1	11	11	19		94.12		72.22
2	7	7	15		53.78		100.00
3	11	11	16		67.23		61.11
4	11	11	19		80.67		91.67
5	15	15	17		53.78		100.00
6	12	12	20		121.01		44.44
7	18	18	19		215.13		75.56
8	7	7	15		67.23		33.33
9	7	7	17		94.12		65.22
10	14	14	16		161.34		52.38
11	10	10	17		80.67		58.33
12	7	7	16		80.67		29.03
13	17	17	18		147.90		127.27
14	9	9	18		94.12		89.47
15	9	9	18		67.23		26.92

	Shooting Accuracy(%)
0	108.33
1	72.22
2	100.00
3	61.11
4	91.67
5	100.00
6	44.44
7	75.56
8	33.33
9	65.22
10	52.38
11	58.33
12	29.03
13	127.27
14	89.47
15	26.92

0.1.15 Ejercicio 14

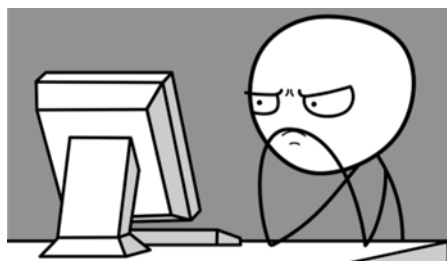
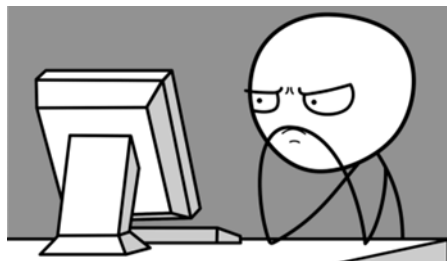
¿Cuántos goles se marcaron en la Euro12?

```
[102]: suma = df_euro12["Goles"].sum()
display(suma)
```

[]:

14_Ejercicios_Series

November 28, 2023



0.1 EJERCICIOS: Series

Para ejercitarte y afianzar lo aprendido sobre Series, completa los siguiente ejercicios.

La solución a los mismos se publicarán en el repo de ejercicios a lo largo del curso.

0.1.1 Ejercicio 0:

Importa pandas "como debe ser" y cualquier otra librería que emplees en el notebook

```
[1]: import pandas as pd
```

0.1.2 Ejercicio 1

Crea una serie con los siguientes valores e índice

- Valores = [45,46,12,10]
- Índices = ["Mum", "Dad", "Jing", "Jang"]

Dale un nombre que creas que refleje el sentido de la serie. Muestra el resultado por pantalla.

```
[2]: valores = [45,46,12,10]
indices = ["Mum","Dad","Jing","Jang"]

serie1 =pd.Series(valores, indices)
print(serie1)
```

```
Mum      45
Dad      46
Jing     12
Jang     10
dtype: int64
```

0.1.3 Ejercicio 2

Para la serie anterior y utilizando atributos de **Series**, muestra sus valores y su índice

```
[3]: print(serie1.values)
print(serie1.index)
```

```
[45 46 12 10]
Index(['Mum', 'Dad', 'Jing', 'Jang'], dtype='object')
```

0.1.4 Ejercicio 3

Crea una series con tus 4 series preferidas, pero como valores no uses los nombres, utiliza el año de su primera temporada y como índices los nombres de las series.

```
[4]: indices= ["Fauda", "Breaking Bad", "Stranger things", "Juego de Tronos"]
valores =[2015, 2008,2016, 2011]
#creo un dictionary, usando la funcion zip
dict_tv = dict(zip(indices, valores))
dict_tv
# creo la serie panda
serie2 = pd.Series(dict_tv)
serie2
```

```
[4]: Fauda                2015
Breaking Bad            2008
Stranger things        2016
Juego de Tronos        2011
dtype: int64
```

0.1.5 Ejercicio 4

¿Cuántos años tiene la más antigua? (Resuélvelo utilizando un método, además de lo que necesites adicionalmente)

```
[5]: años_fin=[2022, 2013,2023,2019]
#creo nuevo serie con años fin de las series
serie3 = pd.Series(años_fin, indices)
serie3
#la serie mas antigua es
#print(serie1.min())
# la serie2 con el año menor correspnde a la serie mas antigua
#print(serie2.min())
# saber los años que ha durado la mas antigua
resta_3_2 = serie3 - serie2
#print(resta_1_2)
print(f" La serie mas antigua se ha retrasmitido durante {resta_3_2.min()}_
↳anyos, desde el año {serie2.min()} hasta el año {serie3.min()}")
```

La serie mas antigua se ha retrasmitido durante 5 años, desde el año 2008 hasta el año 2013

Encuentra cual es la más antigua y la más reciente empleando métodos de Series

```
[6]: print(serie2.min())
print(serie2.max())
```

2008

2016

0.1.6 Ejercicio 6

Accede a la más reciente empleando el índice explícito y el implícito

```
[13]: serie1
#metodo implicito .loc accedemos por etiqueta o condcion(bool)
reciente_i = serie2.loc["Breaking Bad"]
print(f"La serie mas reciente de la serie Panda 1 por el metodo implicito es_
↳del año {reciente_i}")
#metodo explicito, se accede por slicing o posicion en python
reciente_e = serie2.iloc[1:2]
print(f"La serie mas reciente de la serie Panda 1 por el metodo explicito es_
↳{reciente_e}")
```

La serie mas reciente de la serie Panda 1 por el metodo implicito es del anyo 2008

La serie mas reciente de la serie Panda 1 por el metodo explicito es Breaking Bad 2008
dtype: int64

0.1.7 Ejercicio 7

Haz un slicing de la primera serie (ejercicio 1) de forma que no incluya a "Mum" y usando índices explícitos (el índice de la Serie, vaya)

```
[14]: serie1  
      serie1.iloc[1:]
```

```
[14]: Dad      46  
      Jing     12  
      Jang     10  
      dtype: int64
```

0.1.8 Ejercicio 8

Crea un objeto Series de 10 elementos todos ellos con el valor "DataScience", cambia el elemento 5 por "TheBridge" y el 8 por "Bootcamp".

```
[23]: lista= ["DataScience"]*10  
      #creo serie panda  
      serie4= pd.Series(lista)  
      serie4  
      cambio1 = serie4[4] ="theBridge"  
      serie4  
      cambio2 = serie4[7] = "Bootcamp"  
      serie4
```

```
[23]: 0    DataScience  
      1    DataScience  
      2    DataScience  
      3    DataScience  
      4    theBridge  
      5    DataScience  
      6    DataScience  
      7    Bootcamp  
      8    DataScience  
      9    DataScience  
      dtype: object
```

```
[ ]:
```