

apl-1

September 7, 2023

```
[1]: #importacion principal
import pandas as pd
import numpy as np
import os
os.chdir("C:/Users/Lucas/Desktop/Lucas/Programación/TECLAB - DATA SCIENCE/
↳MATERIAS/1ER CUATRIMESTRE/BIMESTRE A/Análisis y visualización de datos/
↳modulo 2/API2/ARCHIVOS")
```

```
[2]: #declaracion de variables
poblacion = pd.read_csv("poblacion.csv", encoding = 'latin-1')

hog_viv_sup = pd.read_csv("hogares_viviendas_superficie.csv", encoding =
↳'latin-1')
```

```
[3]: #en el calculo deberemos extraer ==> poblacion, superficiekm2 de
↳hogares-superficie
```

```
[3]: #filtros

poblacion.drop_duplicates
hog_viv_sup.drop_duplicates
```

```
[3]: <bound method DataFrame.drop_duplicates of      provincia_id      provincia
hogares viviendas_particulares \
0           2      Capital Federal  1150134      1423973
1           6      Buenos Aires  4789484      5377786
2          10      Catamarca    96001      113634
3          14      Córdoba    1031843      1232211
4          18      Corrientes   267797      292644
5          22      Chaco      288422      312602
6          26      Chubut    157166      177985
7          30      Entre Ríos   375121      425591
8          34      Formosa    140303      154458
9          38      Jujuy     174630      195785
10         42      La Pampa   107674      133186
11         46      La Rioja    91097      108967
12         50      Mendoza   494841      538056
```

13	54	Misiones	302953	330049
14	58	Neuquén	170057	193733
15	62	Río Negro	199189	236609
16	66	Salta	299794	315186
17	70	San Juan	177155	188655
18	74	San Luis	126922	142049
19	78	Santa Cruz	81796	93881
20	82	Santa Fe	1023777	1143651
21	86	Santiago del Estero	218025	242034
22	90	Tucumán	368538	396040
23	94	Tierra del Fuego	38956	43360

	viviendas_particulares_habitadas	superficie_km2
0	1082998	200
1	4425193	307571
2	89376	102602
3	978553	165321
4	248844	88199
5	270133	99633
6	147176	224686
7	357250	78781
8	130134	72066
9	154911	53219
10	104797	143440
11	86367	89680
12	459550	148827
13	290263	29801
14	159302	94078
15	190597	203013
16	267075	155488
17	162204	89651
18	117766	76748
19	76233	243943
20	948369	133007
21	197906	136351
22	335821	22524
23	36689	1002445 >

```
[4]: #filtros 2
hog_viv_sup.dropna(axis=1)
poblacion.dropna(axis=1)
```

[4]:	provincia	anio	poblacion_total	poblacion_varones \
0	Total País	2010	40788453	19940704
1	Total País	2011	41261490	20180791
2	Total País	2012	41733271	20420391
3	Total País	2013	42202935	20659037

4	Total País	2014	42669500	20896203
..
770	Tierra del Fuego	2036	241593	122567
771	Tierra del Fuego	2037	245734	124625
772	Tierra del Fuego	2038	249853	126670
773	Tierra del Fuego	2039	253948	128702
774	Tierra del Fuego	2040	258020	130721

	poblacion_mujeres
0	20847749
1	21080699
2	21312880
3	21543898
4	21773297
..	...
770	119026
771	121109
772	123183
773	125246
774	127299

[775 rows x 5 columns]

```
[6]: #filtros para usar los elementos que nos sirven para calcular la nueva
      ↪variable, en este caso vamos a necesitar
      # la poblacion total de cada provincia y la superficie
```

```
[5]: poblacion.drop(columns="poblacion_varones", inplace = True)
      poblacion.drop(columns="poblacion_mujeres", inplace = True)
```

```
[6]: hog_viv_sup.drop(columns="viviendas_particulares", inplace = True)
      hog_viv_sup.drop(columns="viviendas_particulares_habitadas", inplace = True)
```

```
[7]: hog_viv_sup.drop(columns="hogares", inplace = True)
```

```
[8]: #aplicamos otro filtro
      poblacion_f = poblacion[poblacion["anio"] <= 2024]
      poblacion_f1 = poblacion_f[poblacion_f['provincia'] != 'Total País']
```

```
[9]: poblacion_densidad = pd.merge(poblacion_f1, hog_viv_sup, left_on =
      ↪["provincia"], right_on = ["provincia"], how = 'left')
```

```
[10]: #calculo de densidad poblacional
       poblacion_densidad['densidad_poblacional'] =
       ↪poblacion_densidad['poblacion_total']/ poblacion_densidad['superficie_km2']
       poblacion_densidad
```

```
[10]:
```

	provincia	anio	poblacion_total	provincia_id	superficie_km2	\
0	Capital Federal	2010	3028481	2	200	
1	Capital Federal	2011	3033639	2	200	
2	Capital Federal	2012	3038860	2	200	
3	Capital Federal	2013	3044076	2	200	
4	Capital Federal	2014	3049229	2	200	
..	
355	Tierra del Fuego	2020	173432	94	1002445	
356	Tierra del Fuego	2021	177697	94	1002445	
357	Tierra del Fuego	2022	181983	94	1002445	
358	Tierra del Fuego	2023	186285	94	1002445	
359	Tierra del Fuego	2024	190601	94	1002445	

	densidad_poblacional
0	15142.405000
1	15168.195000
2	15194.300000
3	15220.380000
4	15246.145000
..	...
355	0.173009
356	0.177264
357	0.181539
358	0.185831
359	0.190136

[360 rows x 6 columns]

```
[11]: #valores extremos de densidad de poblacion
mostrar1 = np.max(poblacion_densidad["densidad_poblacional"])
mostrar1
```

```
[11]: 15427.415
```

```
[12]: mostrar2 = np.min(poblacion_densidad["densidad_poblacional"])
mostrar2
```

```
[12]: 0.1313398740080503
```

```
[13]: #una vez obtenidos los valores extremos haremos un filtro para saber de donde
      ↪vienen
poblacion_densidad.dtypes
```

```
[13]: provincia      object
      anio           int64
      poblacion_total int64
      provincia_id   int64
```

```
superficie_km2          int64
densidad_poblacional    float64
dtype: object
```

```
[14]: mostrar_extremo_menor =
      ↪poblacion_densidad[poblacion_densidad["densidad_poblacional"] == 0.
      ↪1313398740080503]
      mostrar_extremo_menor
```

```
[14]:      provincia  anio  poblacion_total  provincia_id  superficie_km2  \
345  Tierra del Fuego  2010           131661           94          1002445

      densidad_poblacional
345              0.13134
```

```
[15]: mostrar_extremo_mayor =
      ↪poblacion_densidad[poblacion_densidad["densidad_poblacional"] == 15427.415]
      mostrar_extremo_mayor
```

```
[15]:      provincia  anio  poblacion_total  provincia_id  superficie_km2  \
14   Capital Federal  2024          3085483           2           200

      densidad_poblacional
14              15427.415
```

```
[ ]: # Aquí identificamos ambos valores extremos en densidad poblacional, en el caso
      ↪del menor tiene muy poca poblacion con respecto
      # a su superficie total, y caso contrario en el mayor, demasiado poblacion para
      ↪muy poco espacio en superficie x km2
```