Trabajo individual

Identificación del trabajo

a. **Módulo:** M2 – E1 API2
b. **Asignatura:** Análisis y Visualización de Datos
c. **RA:**
d. **Docente Online:** James Tomalá Robles
e. **Fecha de entrega**: 12 de septiembre 2023

Identificación del estudiante

Nombre y Apellido	**Carrera**
Hugo Morales Paredes	Técnico en Data Science

Introducción

Se debe integrar distintas bases de datos que contienen variables demográficas de la Republica Argentina, población, hogares y viviendas, esperanza de vida, población. Todas se encuentran en formato .csv.

Importar las librerías necesarias y a través de los chequeos correspondientes y elegir las adecuadas para calcular la densidad poblacional, debiendo para ello crear un nuevo campo de densidad.

Explicar los valores y su posible causa.

Desarrollar en un notebook jupyter.

Desarrollo

Desarrollo API 2

Consigna 1

Importar librerias y realizar chequeos básicos, eleccion de librerias a usar.

Importando el primer dataset 'poblacion.csv'

In [1]:
```
import pandas as pd
import numpy as np

#df = pd.read_csv("poblacion.csv")

# crea un dataframe a partir del dataset
df = pd.read_csv('poblacion.csv', index_col=0, encoding='latin-1')
```

Visualizamos el DataFrame y caracteristicas.

In [2]:
```
# Primeros 5 registros del DataFrame
df.head()
```

Out[2]:

provincia	anio	poblacion_total	poblacion_varones	poblacion_mujeres
Total País	2010	40788453	19940704	20847749
Total País	2011	41261490	20180791	21080699
Total País	2012	41733271	20420391	21312880
Total País	2013	42202935	20659037	21543898
Total País	2014	42669500	20896203	21773297

In [3]:
```
# Ultimos 5 registros de DataFrame
df.tail()
```

Out[3]:

provincia	anio	poblacion_total	poblacion_varones	poblacion_mujeres
Tierra del Fuego	2036	241593	122567	119026
Tierra del Fuego	2037	245734	124625	121109
Tierra del Fuego	2038	249853	126670	123183
Tierra del Fuego	2039	253948	128702	125246
Tierra del Fuego	2040	258020	130721	127299

In [4]:
```
# Dimension del DataFrame
df.shape
```

Out[4]:
```
(775, 4)
```

In [5]:
```
# Label de las columnas y tipos de datos
df.columns
```

Out[5]:
```
Index(['anio', 'poblacion_total', 'poblacion_varones', 'poblacion_mujeres'], dtype='object')
```

In [6]:
```
# Informacion de la tabla y Tipos de datos de las columnas (int64=integer de 64 bits)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 775 entries, Total País to Tierra del Fuego
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   anio               775 non-null    int64
 1   poblacion_total    775 non-null    int64
 2   poblacion_varones  775 non-null    int64
 3   poblacion_mujeres  775 non-null    int64
dtypes: int64(4)
memory usage: 30.3+ KB
```

Dejando solo columnas de anio y poblacion_total

```
Como en el DataFrame 'hogares_viviendas_superficie.csv' se encuentra la superficie (km2) de cada provincia y
este censo incluye solo el año 2010,
se debe limpiar el DataFrame correspondiente a 'poblacion' que incluye los datos de 'poblacion_total'.

```

Comenzamos a filtrar el DataFrame, dejando solo el campo de 'provincia', 'anio' y 'poblacion_total'

```
Este nuevo DataFrame lo almacenamos en df_1.
```

In [7]:

```
# loc seleccion por label para ver solo las columnas de anio vs poblacion_total
df_1 = df.loc[ : , ["anio" , "poblacion_total"] ]

display(df_1.head())
```

provincia	anio	poblacion_total
Total País	2010	40788453
Total País	2011	41261490
Total País	2012	41733271
Total País	2013	42202935
Total País	2014	42669500

Seleccionamos solo el anio 2010

```
Este nuevo DataFrame lo almacenamos en df_2
```

In [8]:

```
df_2=df_1[ df['anio'] == 2010 ]
df_2
```

Out[8]:

provincia	anio	poblacion_total
Total País	2010	40788453
Capital Federal	2010	3028481
Buenos Aires	2010	15716942
Catamarca	2010	377676
Córdoba	2010	3373025
Corrientes	2010	1017731
Chaco	2010	1080017
Chubut	2010	513433
Entre Ríos	2010	1255574
Formosa	2010	551626
Jujuy	2010	683513
La Pampa	2010	327028
La Rioja	2010	342582
Mendoza	2010	1774737
Misiones	2010	1113279
Neuquén	2010	571910
Río Negro	2010	648277
Salta	2010	1239111

San Juan	2010	696076
San Luis	2010	443944
Santa Cruz	2010	275452
Santa Fe	2010	3257907
Santiago del Estero	2010	879246
Tucumán	2010	1489225
Tierra del Fuego	2010	131661

Sacamos la fila de "Total País"

```
Este nuevo DataFrame lo almacenamos en df_3
```

In [9]:
```
df_3 = df_2.drop(['Total País'])
#df_2 = df_1.drop(['0'])
```

In [10]:
```
df_3
```

Out[10]:

provincia	anio	poblacion_total
Capital Federal	2010	3028481
Buenos Aires	2010	15716942
Catamarca	2010	377676
Córdoba	2010	3373025
Corrientes	2010	1017731
Chaco	2010	1080017
Chubut	2010	513433
Entre Ríos	2010	1255574
Formosa	2010	551626
Jujuy	2010	683513
La Pampa	2010	327028
La Rioja	2010	342582
Mendoza	2010	1774737
Misiones	2010	1113279
Neuquén	2010	571910
Río Negro	2010	648277
Salta	2010	1239111
San Juan	2010	696076
San Luis	2010	443944
Santa Cruz	2010	275452
Santa Fe	2010	3257907
Santiago del Estero	2010	879246
Tucumán	2010	1489225
Tierra del Fuego	2010	131661

```
Ya tenemos el DataFrame df_3 filtrado a solo las 'provincia' y censo de año 2010.
Ahora importaremos el DataSet donde se encuentra el campo de ´superficie' (km2) que usaremos para el calculo de
la 'densidad' y que tambien corresponde al año 2010.
```

Importando los dataset 'hogares_viviendas_superficie.csv'

Visualizamos el DataFrame y caracteristicas. ¶

In [11]:
```
df_4 = pd.read_csv('hogares_viviendas_superficie.csv', index_col=0, encoding='latin-1')

```

In [12]:
```
df_4.head(30)
```

Out[12]:

provincia_id	provincia	hogares	viviendas_particulares	viviendas_particulares_habitadas	superficie_km2
2	Capital Federal	1150134	1423973	1082998	200
6	Buenos Aires	4789484	5377786	4425193	307571
10	Catamarca	96001	113634	89376	102602
14	Córdoba	1031843	1232211	978553	165321
18	Corrientes	267797	292644	248844	88199
22	Chaco	288422	312602	270133	99633
26	Chubut	157166	177985	147176	224686
30	Entre Ríos	375121	425591	357250	78781
34	Formosa	140303	154458	130134	72066
38	Jujuy	174630	195785	154911	53219
42	La Pampa	107674	133186	104797	143440
46	La Rioja	91097	108967	86367	89680
50	Mendoza	494841	538056	459550	148827
54	Misiones	302953	330049	290263	29801
58	Neuquén	170057	193733	159302	94078
62	Río Negro	199189	236609	190597	203013
66	Salta	299794	315186	267075	155488
70	San Juan	177155	188655	162204	89651
74	San Luis	126922	142049	117766	76748
78	Santa Cruz	81796	93881	76233	243943
82	Santa Fe	1023777	1143651	948369	133007
86	Santiago del Estero	218025	242034	197906	136351
90	Tucumán	368538	396040	335821	22524
94	Tierra del Fuego	38956	43360	36689	1002445

In [13]:
```
# Informacion de la tabla y Tipos de datos de las columnas (int64=integer de 64 bits)
df_4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 24 entries, 2 to 94
Data columns (total 5 columns):
 #   Column                           Non-Null Count  Dtype 
---  ------                           --------------  ----- 
 0   provincia                        24 non-null     object
 1   hogares                          24 non-null     int64 
 2   viviendas_particulares           24 non-null     int64 
 3   viviendas_particulares_habitadas  24 non-null     int64 
 4   superficie_km2                   24 non-null     int64 
dtypes: int64(4), object(1)
memory usage: 1.1+ KB
```

```
Revisamos que ambos DataFrame contengan las mismas provincias y en el mismo orden
```

In [14]:
```
# loc seleccion por label para ver solo las columnas de anio vs poblacion_total
df_5 = df_4.loc[ : , ["provincia" , "superficie_km2"] ]

display(df_5.head(30))
```

provincia_id	provincia	superficie_km2
2	Capital Federal	200
6	Buenos Aires	307571
10	Catamarca	102602
14	Córdoba	165321
18	Corrientes	88199
22	Chaco	99633
26	Chubut	224686
30	Entre Ríos	78781
34	Formosa	72066
38	Jujuy	53219
42	La Pampa	143440
46	La Rioja	89680
50	Mendoza	148827
54	Misiones	29801
58	Neuquén	94078
62	Río Negro	203013
66	Salta	155488
70	San Juan	89651
74	San Luis	76748
78	Santa Cruz	243943
82	Santa Fe	133007
86	Santiago del Estero	136351
90	Tucumán	22524
94	Tierra del Fuego	1002445

Conclusión 1:

Las librerias a usar son: 'poblacion.csv' y 'hogares_viviendas_superficie.csv', ya que estas contienen datos en común y necesarios para el calculo solicitado 'densidad'.

Consigna 2

Calcular un campo nuevo, densidad (población/superficie) y usar la función descrita sobre ese campo nuevo.

Igualando el 'index' en ambas tablas

In [15]:
```
# Index utilizado: 'provincia'

df_6 = df_5.set_index('provincia')
```

Concatenando ambas tablas

In [16]:
```
# pegamos la columna 'poblacion_total' de df_3 en df_6

df_den = pd.merge(df_3, df_6, on='provincia')
```

In [17]:
```
1 df_den
```

Out[17]:

provincia	anio	poblacion_total	superficie_km2
Capital Federal	2010	3028481	200
Buenos Aires	2010	15716942	307571
Catamarca	2010	377676	102602
Córdoba	2010	3373025	165321
Corrientes	2010	1017731	88199
Chaco	2010	1080017	99633
Chubut	2010	513433	224686
Entre Ríos	2010	1255574	78781
Formosa	2010	551626	72066
Jujuy	2010	683513	53219
La Pampa	2010	327028	143440
La Rioja	2010	342582	89680
Mendoza	2010	1774737	148827
Misiones	2010	1113279	29801
Neuquén	2010	571910	94078
Río Negro	2010	648277	203013
Salta	2010	1239111	155488
San Juan	2010	696076	89651
San Luis	2010	443944	76748
Santa Cruz	2010	275452	243943
Santa Fe	2010	3257907	133007

Santiago del Estero	2010	879246	136351
Tucumán	2010	1489225	22524
Tierra del Fuego	2010	131661	1002445

Agregando una nueva columna con un campo nuevo "densidad_km2" (habitantes/superficie)

```
1 El valor de 'densidad_km2' se va a redondear con 3 decimales.
```

In [18]:
```
1 df_den['densidad_km2'] = round (df_den['poblacion_total']/df_den['superficie_km2'] ,3)
```

In [19]:
```
1 df_den.head(24)
```

Out[19]:

provincia	anio	poblacion_total	superficie_km2	densidad_km2
Capital Federal	2010	3028481	200	15142.405
Buenos Aires	2010	15716942	307571	51.100
Catamarca	2010	377676	102602	3.681
Córdoba	2010	3373025	165321	20.403
Corrientes	2010	1017731	88199	11.539
Chaco	2010	1080017	99633	10.840
Chubut	2010	513433	224686	2.285
Entre Ríos	2010	1255574	78781	15.938
Formosa	2010	551626	72066	7.654
Jujuy	2010	683513	53219	12.843
La Pampa	2010	327028	143440	2.280
La Rioja	2010	342582	89680	3.820
Mendoza	2010	1774737	148827	11.925
Misiones	2010	1113279	29801	37.357
Neuquén	2010	571910	94078	6.079
Río Negro	2010	648277	203013	3.193
Salta	2010	1239111	155488	7.969
San Juan	2010	696076	89651	7.764
San Luis	2010	443944	76748	5.784
Santa Cruz	2010	275452	243943	1.129
Santa Fe	2010	3257907	133007	24.494
Santiago del Estero	2010	879246	136351	6.448
Tucumán	2010	1489225	22524	66.117
Tierra del Fuego	2010	131661	1002445	0.131

La densidad poblacional en el año 2010 por 'provincia' es la descrita en la tabla superior.

Consigna 3

Identificar si existe algún valor extremo en la densidad de población y explicar a qué podría deberse esto.

Valores extremos.

```
Igualmente se redondeara con 3 decimales.
```

In [21]:
```
round (df_den['densidad_km2'].describe(), 3)
```

Out[21]:
```
count       24.000
mean       644.299
std       3088.126
min          0.131
25%          3.785
50%          7.866
75%         17.054
max      15142.405
Name: densidad_km2, dtype: float64
```

```
En este caso, la columna densidad tiene 24 valores no nulos. La media de los valores en la columna es 644.299.
La desviación estándar de los valores en la columna es 3088.126. El valor mínimo en la columna es 0.131. El
percentil 25 de los valores en la columnda es 3.785. La mediana de los valores en la columna es 7.866. El
percentil 75 de los valores en la columna es 17.053. El valor máximo en la columna es 15142.405.
```

```
El valor maximo de densidad (habitantes/km2) se muestra en 'Capital_Federal' y esto se debe a que es la capital
de la Republica Argentina.
Normalmente es donde se acumula la mayor parte de la poblacion.
```

Desviacion Estandar de la Muestra

```
Usando metodo numpy y pandas
```

In [23]:
```
# calcular la desviacion estandar para una muestra con numpy:...debe incluir el ddof=1 (sin ddof=1 seria para una p
print ( np.std(df_den['densidad_km2'], ddof=1))

# calcular la desviacion estandar para una muestra con pandas:
print( df_den['densidad_km2'].std())
```

```
3088.126408554278
3088.126408554278
```

Estandarizando 'densidad_km2'

In [33]:
```
#estandarizar la columna densidad con numpy
df_den['estandarizado'] = (df_den['densidad_km2'] - np.mean(df_den['densidad_km2'])) / np.std(df_den['densidad_km2'
df_den['estandarizado'].head(24)
```

Out[33]:
```
provincia
Capital Federal        4.694790
Buenos Aires          -0.192090
Catamarca             -0.207446
Córdoba               -0.202031
Corrientes            -0.204901
Chaco                 -0.205127
Chubut                -0.207898
Entre Ríos            -0.203476
Formosa               -0.206159
Jujuy                 -0.204479
La Pampa              -0.207899
La Rioja              -0.207401
Mendoza               -0.204776
Misiones              -0.196541
Neuquén               -0.206669
Río Negro             -0.207604
Salta                 -0.206057
San Juan              -0.206123
San Luis              -0.206765
Santa Cruz            -0.208272
Santa Fe              -0.200706
Santiago del Estero   -0.206550
Tucumán               -0.187227
Tierra del Fuego      -0.208595
Name: estandarizado, dtype: float64
```

Outlier

In [32]:
```
#Identificar Outliers usando el criterio de desviacion estandar con umbral de 3:

outlier= df_den[(df_den['estandarizado']>3) | (df_den['estandarizado']< -3)]

display(outlier['densidad_km2'])
```

```
provincia
Capital Federal    15142.405
Name: densidad_km2, dtype: float64
```

El único que cumple la condición de un valor atípico es Capital Federal, pero se puede considerar un fenomeno real, ya que es la capital del país y por ello se encuentra muy alejado de los datos del resto de las provincias.

Percentiles

In [37]:
```
# Calculo de los percentiles 25%, 50%, 75%.
percentiles= np.percentile(df_den['densidad_km2'], [25, 50, 75])

print(percentiles)
```

```
[ 3.78525  7.8665  17.05425]
```

De acuerdo a los valores entregados de percentiles podemos decir que:
que el 25% de los datos es menor a 3.78525 y el resto es mayor a 3.78525.
que el 50% de los datos es menor a 7.8665 y el resto es mayor a 7.8665.
que el 75% de los datos es menor a 17.05425 y el resto es mayor a 17.05425.

In [87]:
```
# Determinando los outliers de 'densidad_km2' que sean mayor a 20% y mayor a 80%

p20 = np.percentile (df_den['densidad_km2'], 20)

p80 = np.percentile (df_den['densidad_km2'], 80)

print (p20)
print (p80)
```

```
3.4858000000000002
22.039400000000008
```

En el caso de un percentil específico deseado, como en este caso el 10% es de 2.2825

In [91]:
```
# Determinando el outlier mayor al percentil 20% y menor al 80% (20% por cada extremo)

outlier = df_den[(df_den['densidad_km2'] > p80) | (df_den['densidad_km2'] < p20)]

print (outlier)
```

```
                anio  poblacion_total  superficie_km2  densidad_km2  \
provincia
Capital Federal   2010          3028481             200     15142.405
Buenos Aires      2010         15716942          307571        51.100
Chubut            2010           513433          224686         2.285
La Pampa          2010           327028          143440         2.280
Misiones          2010          1113279           29801        37.357
Río Negro         2010           648277          203013         3.193
Santa Cruz        2010           275452          243943         1.129
Santa Fe          2010          3257907          133007        24.494
Tucumán           2010          1489225           22524        66.117
Tierra del Fuego  2010           131661         1002445         0.131

                estandarizado
provincia
Capital Federal      4.694790
Buenos Aires        -0.192090
Chubut              -0.207898
La Pampa            -0.207899
Misiones            -0.196541
Río Negro           -0.207604
Santa Cruz          -0.208272
Santa Fe            -0.200706
Tucumán             -0.187227
Tierra del Fuego    -0.208595
```

Podemos decir que hay 10 provincias con un percentil entre 20% y 80%

Conclusión

Los data set elegidos fueron 2: poblacion.csv y hogares_viviendas_superficie.csv dado que en ellos se encontraba la variable población y superficie, necesarias para calcular la densidad poblacional (habitantes/km2).

En este caso, la columna densidad tiene 24 valores no nulos, lo que significa que no existen inconsistencias en los datos.

La media de los valores en la columna densidad es 644,299(habitantes/km2). La desviación estándar de los valores en la columna es 3088,126(habitantes/km2).

El valor mínimo en la columna densidad es 0,131(habitantes/km2).

El percentil 25 de los valores en la columna densidad es 3,785(habitantes/km2).

La mediana de los valores en la columna densidad es 7,866(habitantes/km2).

El percentil 75 de los valores en la columna densidad es 17,053(habitantes/km2).

El valor máximo en la columna densidad es 15142,405(habitantes/km2).

El valor máximo de densidad (habitantes/km2) se muestra en 'Capital_Federal' y esto se debe a que es la capital de la República Argentina.

Normalmente en las capitales es donde se concentra la mayor parte de la población.

Bibliografía

Bases de datos públicas de la República Argentina:

- empleo.csv
- esperanza_de_vida.csv
- exportaciones.csv
- hogares_viviendas_superficie.csv
- poblacion.csv