

```
In [1]: # Importamos las librerías pandas y numpy
import pandas as pd
import numpy as np
os.chdir("C:/Users/Ariel/OneDrive/Escritorio/Apl-2-Modulo2")

In [2]: # Importamos los archivos csv originales con los datos que vamos a utilizar
# poblacion por provincia, año, población total, población varones, población mujeres)
poblacion = pd.read_csv("poblacion.csv", encoding="latin-1")

In [3]: # Importamos los archivos csv originales con los datos que vamos a utilizar
# esperanza de vida (provincia, año, mujeres, varones)
esperanza_de_vida = pd.read_csv("esperanza_de_vida.csv", encoding="latin-1")

In [4]: # Importamos los archivos csv originales con los datos que vamos a utilizar
# provincia_id (provincia, hogares, viviendas_particulares, viviendas_particulares_habitadas, superficie_km2)
hogares_viviendas_superficie = pd.read_csv("hogares_viviendas_superficie.csv", encoding="latin-1")

In [5]: # Exploramos las primeras filas del archivo con datos de población
poblacion.head(15)

Out[5]:
   provincia  año  población_total  población_varones  población_mujeres
0  Total País  2010      40788453      19940704      20847749
1  Total País  2011      41261490      20180791      21080699
2  Total País  2012      41733271      20420391      21312880
3  Total País  2013      42202935      20699037      21543988
4  Total País  2014      42669500      20896203      21773297
5  Total País  2015      43131966      21131346      22000620
6  Total País  2016      43990368      21364470      22225988
7  Total País  2017      44044811      21599623      22449188
8  Total País  2018      44494502      21824372      22670230
9  Total País  2019      44938712      22050332      22888380
10 Total País  2020      45376763      22273132      23103631
11 Total País  2021      45808747      22492818      23315929
12 Total País  2022      46234830      22709478      23525352
13 Total País  2023      46654581      22922281      23731700
14 Total País  2024      47067641      23132846      23934795

In [6]: # Exploramos las primeras filas del archivo con datos de esperanza de vida
esperanza_de_vida.head()

Out[6]:
   provincia  año  mujeres  varones
0  Buenos Ares  2015      80.22    73.54
1  Buenos Ares  2020      81.34    74.74
2  Buenos Ares  2025      82.32    75.80
3  Buenos Ares  2030      83.20    76.76
4  Buenos Ares  2035      83.98    77.60

In [7]: # Exploramos las primeras filas del archivo con datos de hogares, viviendas, superficie
hogares_viviendas_superficie.head()

Out[7]:
   provincia_id  provincia  hogares  viviendas_particulares  viviendas_particulares_habitadas  superficie_km2
0              2  Capital Federal  1150134      1423973      1082998      200
1              6  Buenos Aires  4789484      5377786      4425193      307571
2             10  Catamarca      96001      113634      89376      102602
3             14  Córdoba      1031843      1232211      978553      165321
4             18  Corrientes      267797      292644      248844      88199

In [8]: # Exploramos las últimas filas del archivo dataset
poblacion.tail()

Out[8]:
   provincia  año  población_total  población_varones  población_mujeres
770 Tierra del Fuego  2036      241593      122567      119026
771 Tierra del Fuego  2037      245734      124625      121109
772 Tierra del Fuego  2038      249853      126670      123183
773 Tierra del Fuego  2039      253948      128702      125246
774 Tierra del Fuego  2040      258020      130721      127299

In [9]: # Exploramos las últimas filas del archivo dataset
esperanza_de_vida.tail()

Out[9]:
   provincia  año  mujeres  varones
129 Tucumán  2020      81.05    75.11
140 Tucumán  2025      82.11    76.15
141 Tucumán  2030      83.03    77.07
142 Tucumán  2035      83.84    77.88
143 Tucumán  2040      84.54    78.58

In [10]: # Exploramos las últimas filas del archivo dataset
hogares_viviendas_superficie.tail()

Out[10]:
   provincia_id  provincia  hogares  viviendas_particulares  viviendas_particulares_habitadas  superficie_km2
19              78  Santa Cruz      81786      93881      76233      243943
20              82  Santa Fe      1023777      1143651      946399      133907
21              86  Santiago del Estero      218025      242034      197906      136351
22              90  Tucumán      368538      396040      338621      22524
23              94  Tierra del Fuego      38956      43369      36689      1002445

In [11]: # Usamos shape para ver cuantas filas y columnas tiene y print para agregarle texto
print("Las filas y columnas son:")
poblacion.shape

Las filas y columnas son:
(775, 5)

Out[11]:

In [12]: # Usamos shape para ver cuantas filas y columnas tiene y print para agregarle texto
print("Las filas y las columnas son:")
esperanza_de_vida.shape

Las filas y las columnas son:
(144, 4)

Out[12]:

In [13]: # Usamos shape para ver cuantas filas y columnas tiene y print para agregarle texto
print("Las filas y las columnas son:")
hogares_viviendas_superficie.shape

Las filas y las columnas son:
(24, 6)

Out[13]:

In [14]: # Chequeamos los tipos de variables
poblacion.dtypes

Out[14]:
provincia      object
año             int64
poblacion_total  int64
poblacion_varones  int64
poblacion_mujeres  int64
dtype: object

In [15]: # Chequeamos los tipos de variables
esperanza_de_vida.dtypes

Out[15]:
provincia      object
año             int64
mujeres         float64
varones         float64
dtype: object

In [16]: # Chequeamos los tipos de variables
hogares_viviendas_superficie.dtypes

Out[16]:
provincia_id      int64
provincia         object
hogares           int64
viviendas_particulares  int64
viviendas_particulares_habitadas  int64
superficie_km2    int64
dtype: object

In [17]: # Importamos de vuelta el archivo sin las últimas 3 filas
poblacion = pd.read_csv("poblacion.csv", skipfooter = 3, engine = "python", encoding="latin-1")
# Filtramos las observaciones/ filas que corresponden a las provincias y año
poblacion = poblacion[poblacion["provincia"] != "Corrientes", "Chaco", "Total País"]] &
(poblacion["año"] <= 2015)]

poblacion

Out[17]:
   provincia  año  población_total  población_varones  población_mujeres
0  Total País  2010      40788453      19940704      20847749
1  Total País  2011      41261490      20180791      21080699
2  Total País  2012      41733271      20420391      21312880
3  Total País  2013      42202935      20699037      21543988
4  Total País  2014      42669500      20896203      21773297
5  Total País  2015      43131966      21131346      22000620
155 Corrientes  2010      1017731      501452      516279
156 Corrientes  2011      1028248      506702      521546
157 Corrientes  2012      1038786      511969      526817
158 Corrientes  2013      1049325      517240      532085
159 Corrientes  2014      1059836      522500      537336
160 Corrientes  2015      1070283      527731      542552
186 Chaco  2010      1080017      534347      545670
187 Chaco  2011      1092625      540391      552234
188 Chaco  2012      1105280      546471      558809
189 Chaco  2013      1117953      552570      565383
190 Chaco  2014      1130608      558670      571938
191 Chaco  2015      1143201      564746      578455

In [18]: # Importamos de vuelta el archivo sin las últimas 3 filas
esperanza_de_vida = pd.read_csv("esperanza_de_vida.csv", skipfooter = 3, engine = "python", encoding = "latin-1")
# Filtramos las observaciones/ filas que corresponden a las provincias y año
esperanza_de_vida = esperanza_de_vida[esperanza_de_vida["provincia"] != "Buenos Aires", "Catamarca"]] &
(esperanza_de_vida["año"] <= 2040)]

esperanza_de_vida

Out[18]:
   provincia  año  mujeres  varones
0  Buenos Ares  2015      80.22    73.54
1  Buenos Ares  2020      81.34    74.74
2  Buenos Ares  2025      82.32    75.80
3  Buenos Ares  2030      83.20    76.76
4  Buenos Ares  2035      83.98    77.60
5  Buenos Ares  2040      84.66    78.32
6  Catamarca  2015      80.36    74.74
7  Catamarca  2020      81.45    75.78
8  Catamarca  2025      82.45    76.70
9  Catamarca  2030      83.32    77.55
10 Catamarca  2035      84.08    78.28
11 Catamarca  2040      84.75    78.91

In [19]: # Importamos de vuelta el archivo sin las últimas 3 filas
hogares_viviendas_superficie = pd.read_csv("hogares_viviendas_superficie.csv", skipfooter = 3, engine = "python", encoding = "latin-1")
# Filtramos las observaciones/ filas que corresponden a las provincias
hogares_viviendas_superficie = hogares_viviendas_superficie[hogares_viviendas_superficie["provincia"] != "Corrientes"]]

hogares_viviendas_superficie

Out[19]:
   provincia_id  provincia  hogares  viviendas_particulares  viviendas_particulares_habitadas  superficie_km2
4             18  Corrientes      267797      292644      248844      88199

In [20]: # Eliminamos columnas
poblacion = poblacion.drop(columns = ["poblacion_mujeres"])
poblacion

Out[20]:
   provincia  año  población_total  población_varones
0  Total País  2010      40788453      19940704
1  Total País  2011      41261490      20180791
2  Total País  2012      41733271      20420391
3  Total País  2013      42202935      20699037
4  Total País  2014      42669500      20896203
5  Total País  2015      43131966      21131346
155 Corrientes  2010      1017731      501452
156 Corrientes  2011      1028248      506702
157 Corrientes  2012      1038786      511969
158 Corrientes  2013      1049325      517240
159 Corrientes  2014      1059836      522500
160 Corrientes  2015      1070283      527731
186 Chaco  2010      1080017      534347
187 Chaco  2011      1092625      540391
188 Chaco  2012      1105280      546471
189 Chaco  2013      1117953      552570
190 Chaco  2014      1130608      558670
191 Chaco  2015      1143201      564746

In [21]: # Eliminamos columnas
esperanza_de_vida = esperanza_de_vida.drop(columns = ["varones"])
esperanza_de_vida

Out[21]:
   provincia  año  mujeres
0  Buenos Ares  2015      80.22
1  Buenos Ares  2020      81.34
2  Buenos Ares  2025      82.32
3  Buenos Ares  2030      83.20
4  Buenos Ares  2035      83.98
5  Buenos Ares  2040      84.66
6  Catamarca  2015      80.36
7  Catamarca  2020      81.45
8  Catamarca  2025      82.45
9  Catamarca  2030      83.32
10 Catamarca  2035      84.08
11 Catamarca  2040      84.75

In [22]: # Eliminamos columnas
hogares_viviendas_superficie = hogares_viviendas_superficie.drop(columns = ["viviendas_particulares", "viviendas_particulares_habitadas"])
hogares_viviendas_superficie

Out[22]:
   provincia_id  provincia  hogares  superficie_km2
4             18  Corrientes      267797      88199

In [23]: # Transposición de formato ancho a largo
ndf_melt = pd.melt(poblacion, id_vars=["provincia"], value_vars=["año", "poblacion_total", "poblacion_varones"])
ndf_melt

In [24]: # Renombramos la variable año por año
poblacion = poblacion.rename(columns = ("año": "año"))
poblacion

Out[24]:
   provincia  año  población_total  población_varones
0  Total País  2010      40788453      19940704
1  Total País  2011      41261490      20180791
2  Total País  2012      41733271      20420391
3  Total País  2013      42202935      20699037
4  Total País  2014      42669500      20896203
5  Total País  2015      43131966      21131346
155 Corrientes  2010      1017731      501452
156 Corrientes  2011      1028248      506702
157 Corrientes  2012      1038786      511969
158 Corrientes  2013      1049325      517240
159 Corrientes  2014      1059836      522500
160 Corrientes  2015      1070283      527731
186 Chaco  2010      1080017      534347
187 Chaco  2011      1092625      540391
188 Chaco  2012      1105280      546471
189 Chaco  2013      1117953      552570
190 Chaco  2014      1130608      558670
191 Chaco  2015      1143201      564746

In [25]: # Renombramos la variable año por año
esperanza_de_vida = esperanza_de_vida.rename(columns = ("año": "año"))
esperanza_de_vida

Out[25]:
   provincia  año  mujeres
0  Buenos Ares  2015      80.22
1  Buenos Ares  2020      81.34
2  Buenos Ares  2025      82.32
3  Buenos Ares  2030      83.20
4  Buenos Ares  2035      83.98
5  Buenos Ares  2040      84.66
6  Catamarca  2015      80.36
7  Catamarca  2020      81.45
8  Catamarca  2025      82.45
9  Catamarca  2030      83.32
10 Catamarca  2035      84.08
11 Catamarca  2040      84.75

In [26]: # No es necesario usar la función sort para ordenar ya que la tabla está ordenada.

In [27]: # Vemos los tipos de variable de cada columna
poblacion.dtypes

Out[27]:
provincia      object
año             int64
poblacion_total  int64
poblacion_varones  int64
dtype: object

In [28]: # Vemos los tipos de variable de cada columna
esperanza_de_vida.dtypes

Out[28]:
provincia      object
año             int64
mujeres         float64
dtype: object

In [29]: # Vemos los tipos de variable de cada columna
hogares_viviendas_superficie.dtypes

Out[29]:
provincia_id      int64
provincia         object
hogares           int64
superficie_km2    int64
dtype: object

In [30]: # Convertimos algunos de los campos a tipos de variables que sirvan: provincia: string, población_varones: int
poblacion = poblacion.astype({"provincia": str, "poblacion_varones": int})
poblacion.dtypes

Out[30]:
provincia      object
año             int64
poblacion_total  int64
poblacion_varones  int32
dtype: object

In [31]: # Convertimos algunos de los campos a tipos de variables que sirvan: provincia: string, año: int
esperanza_de_vida = esperanza_de_vida.astype({"provincia": str, "año": int})
esperanza_de_vida.dtypes

Out[31]:
provincia      object
año             int32
mujeres         float64
dtype: object

In [32]: # En el archivo esperanza_de_vida y hogares_viviendas_superficie no se necesita hacer la conversión

In [33]: # Filtramos las observaciones anteriores a 2015, el último año con datos completos
poblacion = poblacion[poblacion["año"] <= 2015]
poblacion

Out[33]:
   provincia  año  población_total  población_varones
0  Total País  2010      40788453      19940704
1  Total País  2011      41261490      20180791
2  Total País  2012      41733271      20420391
3  Total País  2013      42202935      20699037
4  Total País  2014      42669500      20896203
5  Total País  2015      43131966      21131346
155 Corrientes  2010      1017731      501452
156 Corrientes  2011      1028248      506702
157 Corrientes  2012      1038786      511969
158 Corrientes  2013      1049325      517240
159 Corrientes  2014      1059836      522500
160 Corrientes  2015      1070283      527731
186 Chaco  2010      1080017      534347
187 Chaco  2011      1092625      540391
188 Chaco  2012      1105280      546471
189 Chaco  2013      1117953      552570
190 Chaco  2014      1130608      558670
191 Chaco  2015      1143201      564746

In [34]: # Filtramos las observaciones anteriores a 2040, el último año con datos completos
esperanza_de_vida = esperanza_de_vida[esperanza_de_vida["año"] <= 2040]
esperanza_de_vida

Out[34]:
   provincia  año  mujeres
0  Buenos Ares  2015      80.22
1  Buenos Ares  2020      81.34
2  Buenos Ares  2025      82.32
3  Buenos Ares  2030      83.20
4  Buenos Ares  2035      83.98
5  Buenos Ares  2040      84.66
6  Catamarca  2015      80.36
7  Catamarca  2020      81.45
8  Catamarca  2025      82.45
9  Catamarca  2030      83.32
10 Catamarca  2035      84.08
11 Catamarca  2040      84.75

In [35]: # Calculamos las principales estadísticas descriptivas de la variable población_varones
poblacion["poblacion_varones"].describe()

Out[35]:
count      1.899890e+1
mean       7.269737e+06
std        9.770366e+06
min        5.214520e+05
25%        5.238878e+05
50%        5.495295e+05
75%        2.612877e+07
max        2.113135e+07
Name: población_varones, dtype: float64

In [36]: # Calculamos las principales estadísticas descriptivas de la variable mujeres
esperanza_de_vida["mujeres"].describe()

Out[36]:
count      12.000000
mean       82.677500
std        1.517995
min        80.220000
25%        81.422500
50%        82.825000
75%        84.085000
max        84.750000
Name: mujeres, dtype: float64

In [37]: # Calculamos las principales estadísticas descriptivas de la variable hogares
hogares_viviendas_superficie["hogares"].describe()

Out[37]:
count      1.0
mean       267797.0
std        267797.0
min        267797.0
25%        267797.0
50%        267797.0
75%        267797.0
max        267797.0
Name: hogares, dtype: float64

In [38]: # Concatenamos la tabla poblacion con esperanza_de_vida
# resultado = pd.concat([poblacion, esperanza_de_vida])
# Combinamos la tabla hogares_viviendas_superficie con la tabla concatenada
# resultado = pd.merge(hogares_viviendas_superficie, resultado, left_on = ["provincia"], right_on = ["provincia"], how = "outer")
# resultado

In [39]: # Filtramos datos para quedarnos con la provincia de Corrientes del año 2010
Corrientes = poblacion[poblacion["provincia"] == "Corrientes"] & (poblacion["año"] == 2010)
print(Corrientes)

   provincia  año  población_total  población_varones
155 Corrientes  2010      1017731      501452

In [40]: # Obtenemos la población total
pt = (Corrientes["poblacion_total"])
print(pt)

155      1017731
Name: población_total, dtype: int64

In [41]: # Obtenemos la provincia y la superficie
dt = hogares_viviendas_superficie[["provincia", "superficie_km2"]]
print(dt)

   provincia  superficie_km2
4  Corrientes      88199

In [42]: # Obtenemos la superficie
sup = (dt["superficie_km2"])
print(sup)

4      88199
Name: superficie_km2, dtype: int64

In [43]: # Calculamos un campo nuevo de densidad
densidad_Corrientes = pt/88199
print(densidad_Corrientes)

13520853.1
Name: población_total, dtype: float64

In [44]: # Identificamos outliers calculando la variable estandarizada
esperanza_de_vida["esperanza_de_vida_estandarizada"] = (esperanza_de_vida["mujeres"] - np.mean(esperanza_de_vida["mujeres"])/ np.std(esperanza_de_vida["mujeres"])/ np.std(esperanza_de_vida["mujeres"]))

Out[44]:
   provincia  año  población_total  población_varones  población_estandarizada
0  Buenos Ares  2015      80.22      -1.624546
1  Buenos Ares  2020      81.34      -0.884163
2  Buenos Ares  2025      82.32      -0.236328
3  Buenos Ares  2030      83.20      0.345402
4  Buenos Ares  2035      83.98      0.861026
5  Buenos Ares  2040      84.66      1.310545
6  Catamarca  2015      80.36      -1.531988
7  Catamarca  2020      81.45      -0.811447
8  Catamarca  2025      82.45      -0.150390
9  Catamarca  2030      83.32      0.424279
10 Catamarca  2035      84.08      0.927132
11 Catamarca  2040      84.75      1.370040

Al estandarizar la variable queda una media de 0 y una desviación estandar de 1.

Los datos por encima de 0 significan que por ejemplo para el año 2015 en el total de Buenos Aires se estima que la esperanza de vida de las mujeres aumente a un valor de 84.66 años, con 1.310545 desviaciones por encima de la media original debido a una combinación de factores, incluyendo avances en la nutrición, avances en la higiene y saneamiento, y estilo de vida más saludables. Los datos por debajo de la media original como en el caso de Buenos Aires en el año 2015, se logran de la misma manera para las mujeres se menor: 80.22 años con -1.624546 desviaciones por debajo de la media original debido a que no había suficientes avances en la medicina, la nutrición y la higiene, así como el acceso a atención médica de calidad.

In [45]: # Otra forma de identificar outliers usando percentiles
p99 = np.percentile(esperanza_de_vida["mujeres"], 99)
p99

84.7401

Out[45]:

In [46]: # Vemos las observaciones por encima del percentil 99
esperanza_de_vida[esperanza_de_vida["mujeres"] >= p99]

Out[46]:
   provincia  año  mujeres  esperanza_de_vida_estandarizada
0  Buenos Ares  2015      80.22      -1.624546
1  Buenos Ares  2020      81.34      -0.884163
2  Buenos Ares  2025      82.32      -0.236328
3  Buenos Ares  2030      83.20      0.345402
4  Buenos Ares  2035      83.98      0.861026
5  Buenos Ares  2040      84.66      1.310545
6  Catamarca  2015      80.36      -1.531988
7  Catamarca  2020      81.45      -0.811447
8  Catamarca  2025      82.45      -0.150390
9  Catamarca  2030      83.32      0.424279
10 Catamarca  2035      84.08      0.927132
11 Catamarca  2040      84.75      1.370040

In [47]: # Identificamos outliers calculando la variable estandarizada
poblacion["poblacion_estandarizada"] = (poblacion["poblacion_total"] - np.mean(poblacion["poblacion_total"])/ np.std(poblacion["poblacion_total"])/ np.std(poblacion["poblacion_total"]))

Out[47]:
   provincia  año  población_total  población_varones  población_estandarizada
0  Total País  2010      40788453      19940704      1.352801
1  Total País  2011      41261490      20180791      1.377336
2  Total País  2012      41733271      20420391      1.401806
3  Total País  2013      42202935      20699037      1.426167
4  Total País  2014      42669500      20896203      1.450366
5  Total País  2015      43131966      21131346      1.474354
155 Corrientes  2010      1017731      501452      -0.710019
156 Corrientes  2011      1028248      506702      -0.709474
157 Corrientes  2012      1038786      511969      -0.708927
158 Corrientes  2013      1049325      517240      -0.708381
159 Corrientes  2014      1059836      522500      -0.707835
160 Corrientes  2015      1070283      527731      -0.707294
186 Chaco  2010      1080017      534347      -0.706789
187 Chaco  2011      1092625      540391      -0.706135
188 Chaco  2012      1105280      546471      -0.705478
189 Chaco  2013      1117953      552570      -0.704821
190 Chaco  2014      1130608      558670      -0.704165
191 Chaco  2015      1143201      564746      -0.703512

Al estandarizar la variable queda una media de 0 y una desviación estandar de 1.

Los datos por encima de 0 significan que por ejemplo para el año 2015 en el total de los países la población fue de: 43.131.966 de habitantes y se ha incrementado con el paso de los años con 1.474354 desviaciones por encima de la media original debido a que está experimentando un crecimiento y un desarrollo positivo. Los datos con desviación negativa, como en el caso de las provincias de Corrientes y Chaco la población es menor debido a menos desarrollo económico, falta de servicios, dificultades en el acceso a bienes y servicios o menos oportunidades de educación pero vemos que esos valores se van acercando a la media con el correr de los años, por lo que hay avances.
```