PRÁCTICA 2

1. Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El dataset en cuestión, contiene los datos correspondientes a las cifras de población de las comunidades autónomas y provincias españolas, junto a la total. Además, diferencia el sexo, lugar de procedencia, grupo quinquenal de edades, periodo de los datos y pobación que corresponde.

Vistos uno a uno los distintos campos del dataset:

* Edad: variable categórica que indica los rangos quinquenales de edad. Desde “De 0 a 4 años” hasta “100 y más años”.
* Sexo: variable categórica que indica el sexo de la población que corresponda. “Mujer”, “Hombre” o “Total”.
* Periodo: variable numérica que indica el año al que corresponde la medida de la población. Desde 2015 hasta 2020.
* Lugar: variable categórica que indica la procedencia donde se está realizando la medida. Desde el Total, comunidades y ciudades autónomas como Aragón o Melilla, y las provincias españolas.
* Nacimiento: variable categórica que indica el lugar de procedencia de la población que se está midiendo.
* Total: variable numérica que indica la población que corresponde.

Estos datos han sido recopilados del Instituto Nacional de Estadística de España y nos servirán para ver el envejecimiento en España, junto con las pirámides poblacionales para comparar población en cuanto a términos de edad y de localizaciones.

Una pirámide poblacional representa las características de una población en una localización concreta en función de rangos quinquenales de edad, para ver la distribución y comprar momentos y lugares distintos.

Con los datos de los que se parte se podrían hacer un montón de medidas y comparativas, pero nosotros nos vamos a centrar en el envejecimiento de España y las pirámides poblacionales de ciertas provincias y comunidades, más y menos pobladas, y la total par hacer estudios demográficos y equilibrio entre sexos.

Las preguntas a las que responderemos serán:

* ¿Es España un país viejo?
* ¿Hay equilibrio entre sexos en provincias más pobladas o menos pobladas?

**2. Integración y selección de los datos de interés a analizar.**

Para poder realizar el análisis que se ha detallado anteriormente, se van a leer los datos del fichero “population\_spain\_dataset.csv”, generado en la práctica anterior, y convertir en un dataframe para un manejo mejor de los datos:

*df = pd.read\_csv ('population\_spain\_dataset.csv')*

*df.tail()*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Lugar** | **Nacimiento** | **Edad** | **Sexo** | **Periodo** | **Total** |
| **3763579** | 52 Melilla | Resto de Países de Oceanía | 100 y más años | Mujeres | 2019 | 0 |
| **3763580** | 52 Melilla | Resto de Países de Oceanía | 100 y más años | Mujeres | 2018 | 0 |
| **3763581** | 52 Melilla | Resto de Países de Oceanía | 100 y más años | Mujeres | 2017 | 0 |
| **3763582** | 52 Melilla | Resto de Países de Oceanía | 100 y más años | Mujeres | 2016 | 0 |
| **3763583** | 52 Melilla | Resto de Países de Oceanía | 100 y más años | Mujeres | 2015 | 0 |

Una vez se han cargado en un dataframe, como en este caso no se va a utilizar el dato del origen de procedencia, se van a seleccionar las que correspondan a “Todas” las procedencias y se van a eliminar las demás, además de eliminar dicha columna una vez filtradas dichas filas:

*df = df.loc[df['Nacimiento'] == 'Total']*

*del(df['Nacimiento'])*

Como en los campos de ‘Sexo’ tenemos “Total”, “Hombre” y “Mujer” y el resultado de la suma de hombres y mujeres corresponde al total, se van a eliminar las filas que tengan ese total:

*df = df.loc[df['Sexo'] != 'Total']*

Con el campo de “edad” pasa algo similar, se tienen los grupos quinquenales además de un campo que tiene “todas las edades” que vamos a filtrar y quedarnos con el resto:

*df = df.loc[df['Edad'] != 'Todas las edades']*

*# también hay algún espacio al principio, lo quitamos*

*df['Edad'] = df['Edad'].str.strip()*

Con el campo de ‘Lugar’ debemos pensar que está el “total nacional”, todas las comunidades y las provincias. Nos vamos a quedar solamente con los datos de las provincias y el total nacional, ya que las pirámides de población la haremos de alguna provincia, cuando se quiera hacer de una comunidad se escogen las provincias que correspondan a cada comunidad y el total nacional se va a conservar ya que es solamente un dato y para calcular el envejecimiento se hará mucho más rápido en una sola consulta:

*comun = ['01 Andalucía', '02 Aragón', '03 Asturias, Principado de', '04 Balears, Illes', '05 Canarias', '06 Cantabria', '07 Castilla y León',*

*'08 Castilla - La Mancha', '09 Cataluña', '10 Comunitat Valenciana', '11 Extremadura', '12 Galicia', '13 Madrid, Comunidad de', '14 Murcia, Región de',*

*'15 Navarra, Comunidad Foral de', '16 País Vasco', '17 Rioja, La', '18 Ceuta', '19 Melilla']*

*df = df.loc[~df['Lugar'].isin(comun)]*

Se eliminan los números y espacios que haya en los nombres:

*df['Lugar'] = df['Lugar'].str.strip()*

Ahora se tienen los datos que se van a analizar posteriormente en los siguientes apartados.

**3. Limpieza de los datos.**

En este apartado, a partir del dataframe preparado con los datos a analizar que se ha realizado anteriormente, se va a ver si el dataset contiene elementos vacíos, ceros o valores extremos.

**3.1. ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?**

En este caso, para saber si el dataframe contiene elementos vacíos se utilizan funciones de pandas que dan dicha información:

*df.isna().sum()*

*Lugar 0*

*Edad 0*

*Sexo 0*

*Periodo 0*

*Total 0*

*dtype: int64*

Se puede ver como no existen elementos vacíos dentro del set de datos. De haber datos vacíos, solamente tendría sentido en el campo de población ya que el resto son variables definidas previamente que son categóricas. El que un dato tenga el campo de población total vacío, indica que no se tiene sabiduría de dicho dato y que no podría obtenerse mediante ninguno de los métodos que conocemos, ya que no tendría sentido utilizar el método de los vecinos cercanos para obtener o predecir dicho valor por ejemplo. Por esto, se tendría que prescindir de dicho valor.

En cuanto a elementos nulos pasa algo similar que en el caso anterior pero en este caso no se tendría que prescindir, ya que el valor nulo en el campo de población sí que nos da información. El que el campo población sea 0, indica que para un lugar en concreto, un periodo, un grupo quinquenal y un sexo concreto no existe población, por lo que no se deben de eliminar.

**3.2. Identificación y tratamiento de valores extremos.**

De la misma forma que en el caso anterior, el que exista un valor extremo no se debe de tratar como algo a eliminar ya que tiene sentido para las conclusiones que se deben de sacar.

No existen valores extremos en este caso, pero de haberlos se deberían de dejar porque indican que para un lugar en concreto, un periodo, un grupo quinquenal y un sexo concreto existe un pico de población.

**4. Análisis de los datos.**

**4.1. Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).**

Para este trabajo, se ha decidido seleccionar la provincia de Madrid, la provincia de Zaragoza, la comunidad autónoma de Extremadura y los datos de España para calcular las pirámides de población correspondientes y el envejecimiento.

*# Madrid 2020*

*df\_madrid = df.loc[(df['Lugar']=='Madrid') & (df['Periodo']==2020)]*

*df\_mad = pd.DataFrame({'Edad': list(df\_madrid.loc[(df\_madrid['Sexo']=='Mujeres')]['Edad']),*

*'Hombres': list(df\_madrid.loc[(df\_madrid['Sexo']=='Hombres')]['Total']),*

*'Mujeres': list(df\_madrid.loc[(df\_madrid['Sexo']=='Mujeres')]['Total'])})*

*# Zaragoza 2020*

*df\_zaragoza = df.loc[(df['Lugar']=='Zaragoza') & (df['Periodo']==2020)]*

*df\_zgz = pd.DataFrame({'Edad': list(df\_zaragoza.loc[(df\_zaragoza['Sexo']=='Mujeres')]['Edad']),*

*'Hombres': list(df\_zaragoza.loc[(df\_zaragoza['Sexo']=='Hombres')]['Total']),*

*'Mujeres': list(df\_zaragoza.loc[(df\_zaragoza['Sexo']=='Mujeres')]['Total'])})*

*# Extremadura 2020*

*df\_extremadura = df.loc[((df['Lugar']=='Cáceres') | (df['Lugar']=='Badajoz')) & (df['Periodo']==2020)]*

*edades = list(df\_extremadura.loc[(df\_extremadura['Sexo']=='Mujeres') & (df\_extremadura['Lugar']=='Badajoz')]['Edad'])*

*total\_mujeres\_ext = df\_extremadura.loc[(df\_extremadura['Sexo']=='Mujeres') & (df\_extremadura['Lugar'].isin(['Badajoz', 'Cáceres']))]*

*total\_hombres\_ext = df\_extremadura.loc[(df\_extremadura['Sexo']=='Hombres') & (df\_extremadura['Lugar'].isin(['Badajoz', 'Cáceres']))]*

*mujeres\_ex = []*

*hombres\_ex = []*

*for edad in edades:*

*mujeres\_ex.append(total\_mujeres\_ext.loc[total\_mujeres\_ext['Lugar'].isin(['Cáceres', 'Badajoz']) & (total\_mujeres\_ext['Edad'] == edad)]['Total'].sum())*

*hombres\_ex.append(total\_hombres\_ext.loc[total\_hombres\_ext['Lugar'].isin(['Cáceres', 'Badajoz']) & (total\_hombres\_ext['Edad'] == edad)]['Total'].sum())*

*df\_ext = pd.DataFrame({'Edad': edades,*

*'Hombres': hombres\_ex,*

*'Mujeres': mujeres\_ex)*

Para el caso de españa se va a aplicar la fórmula del envejecimiento, que corresponde con la proporción de mayores de 64 años= ((Población >64 años / Población total) x100), por lo que los datos serán:

*# España 2015-2020*

*df\_espana = df.loc[(df['Lugar']=='Total Nacional')]*

*periods = list(df\_espana[‘Periodo’].unique())*

*edades\_mas\_64 = ['De 65 a 69 años', 'De 70 a 74 años', 'De 75 a 79 años', 'De 80 a 84 años', 'De 85 a 89 años', 'De 90 a 94 años', 'De 95 a 99 años', '100 y más años']*

*dict\_env = {}*

*for per in periods:*

*total\_pob = df\_espana.loc[df\_espana['Periodo'] == per]['Total'].sum()*

*total\_pob\_mas\_64 = df\_espana.loc[df\_espana['Edad'].isin(edades\_mas\_64) & (df\_espana['Periodo'] == per)]['Total'].sum()*

*env = (total\_pob\_mas\_64/total\_pob)\*100*

*dict\_env[per] = env*

*# dataframe pirámide 2020*

*df\_esp = pd.DataFrame({'Edad': list(df\_espana.loc[(df\_espana['Sexo']=='Mujeres') & (df\_espana['Periodo'] == 2020)]['Edad']),*

*'Hombres': list(df\_espana.loc[(df\_espana['Sexo']=='Hombres') & (df\_espana['Periodo'] == 2020)]['Total']),*

*'Mujeres': list(df\_espana.loc[(df\_espana['Sexo']=='Mujeres') & (df\_espana['Periodo'] == 2020)]['Total'])})*

En este punto, ya tenemos los datos preparados para realizar el estudio.

**4.2. Comprobación de la normalidad y homogeneidad de la varianza.**

\*\*

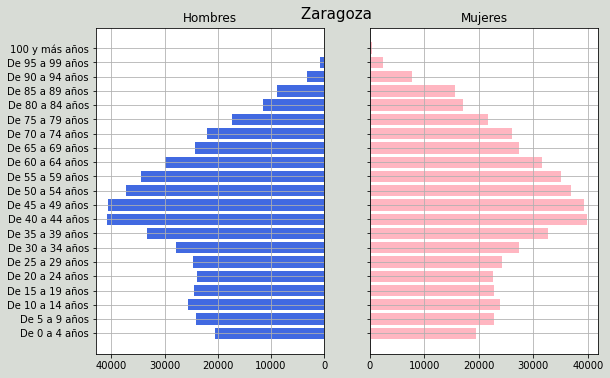
**4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.**

**\*\***

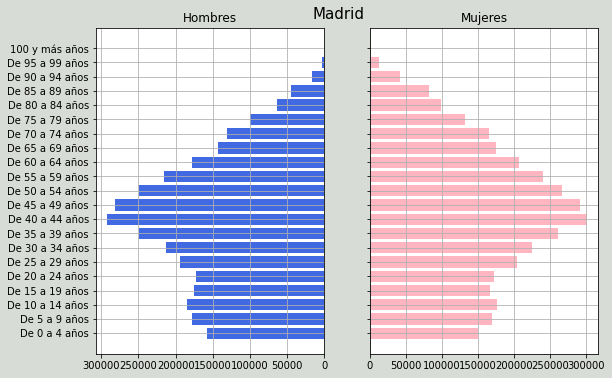
**5. Representación de los resultados a partir de tablas y gráficas.**

En este punto se van a representar los datos anteriormente seleccionados:

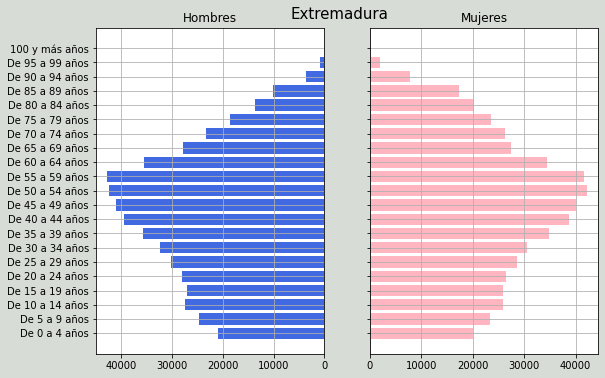
* Pirámide poblacional de la provincia de Zaragoza en 2020



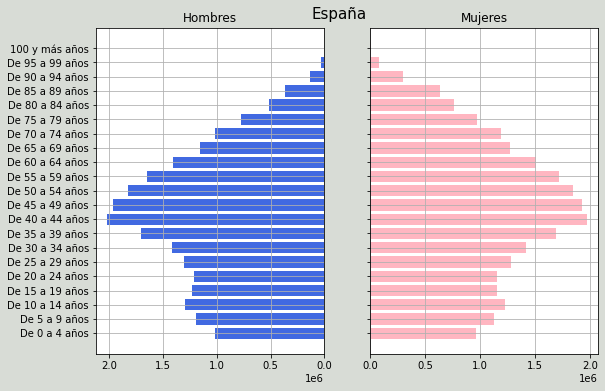
* Pirámide poblacional de la provincia de Madrid (Comunidad autónoma) en 2020



* Pirámide poblacional de la Comunidad autónoma de Extremadura en 2020



* Pirámide poblacional de España en 2020



* Envejecimiento de España de 2015 a 2020

**6. Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?**

**7. Código: Adjuntado en el fichero clean\_data.py dentro de github.**