

Filtracion_Data_Dengue

October 30, 2023

1 Filtracion de datos

Usaremos las librerias numpy y panda para manipular los datas

```
[122]: # Importamos las librerias
import numpy as np
import pandas as pd
```

Primero leemos los datas set que utilizaremos - dengue_2013_2023.csv contiene informacion sobre los casos de dengue resgistrados desde 2013 hasta 2023 - DATA_SOCIAL_CLIMA.csv continene informacion del clima y de factores sociales - reinpress_piura.csv contiene informacion sobre los codigos de reinpress del departamento del cual haremos el analisis en este caso Piura

Empezamos limpiando el data dengue_2013_2023.csv, necesitamos la columna renipress para filtrar la data por ubicacion geografica, para manipularlo de forma correcta necesitamos que el valor de la columna sea de tipo int (entero) y eliminar los valores nulos Posteriormente filtramos la columna tipo_dx para considerar solamente los casos confirmados

```
[59]: # --- Acceder a los datos
data_dengue = pd.read_csv('dengue_2013_2023.csv')
#####
# Reemplazar los puntos ('.') por un valor nulo o NaN
data_dengue['renipress'] = data_dengue['renipress'].replace('.', pd.NA)

# Eliminar los valores NaN y luego convertir a entero
data_dengue['renipress'] = data_dengue['renipress'].fillna(0).astype(int)
#Filtrando los casos confirmados
data_dengue = data_dengue[data_dengue["tipo_dx"] == "C"]

# Verificar el tipo de datos después de la conversión
print(data_dengue['renipress'].dtype)

# --- Visualizar datos
data_dengue.head()
```

int32

```
[59]:   ubigeo  renipress  diagnostic  ano  semana  tipo_dx  edad  tipo_edad  sexo  \
0   21803        1675      A97.0  2013        1        C   42         A      F
```

1	21803	1675	A97.0	2013	1	C	66	A	F
2	21803	1675	A97.0	2013	1	C	13	A	M
3	21803	1675	A97.0	2013	1	C	54	A	F
4	21803	9047	A97.0	2013	1	C	28	A	F

```

    fecha_ini
0 2013-01-02
1 2013-01-01
2 2013-01-02
3 2013-01-03
4 2013-01-05

```

Ahora leemos el archivo DATA_SOCIAL_CLIMA.csv

```
[61]: data_social = pd.read_csv('DATA_SOCIAL_CLIMA.csv', delimiter=';')
data_social.head()
```

```
[61]: Departamento Provincia  Semana Fecha (inicia) Temp. Media (°C) \
0      Piura    AYABACA      9    2023-02-26      18,79
1      Piura    AYABACA     10    2023-03-05      17,78
2      Piura    AYABACA     11    2023-03-12      17,89
3      Piura    AYABACA     12    2023-03-19      18,18
4      Piura    AYABACA     13    2023-03-26      18,12
```

	Precipitación (mm)	Humedad (%)	PobrezaTotal(%)	PobrezaExtrema(%)
0	95,17	85,99	50381116,1	2428404,6
1	143,31	92,8	50381116,1	2428404,6
2	66,84	88,2	50381116,1	2428404,6
3	73,78	88,03	50381116,1	2428404,6
4	69,04	90,66	50381116,1	2428404,6

Arrays de datos de renipress separados por provincias leemos el archivo 'reinpress_piura.csv'

```
[62]: # --- Acceder a los datos
data_reinpress = pd.read_csv('reinpress_piura.csv', delimiter=';')

# --- Visualizar datos automaticamente
data_reinpress.head()
#print(len(data_dengue))
```

```
[62]: Código Único Departamento Provincia    Distrito
0      2001      PIURA    SULLANA    BELLAVISTA
1      2018      PIURA    SULLANA    SULLANA
2     31459      PIURA    AYABACA    PAIMAS
3      2025      PIURA    PAITA    ARENAL
4      2019      PIURA    SULLANA    MIGUEL CHECA
```

1.1 Definicion de variables

definimos la provincia que deseamos analizar, utilizamos el upper() para que el string de la variable este siempre en mayusculas

```
[11]: provincia='Piura'.upper() #NOMBRE DE LA PROVINCIA
      print(provincia)
```

PIURA

1.1.1 Filtracion

Necesitamos todos los codigos de renipress pertenecientes a la provincia Piura, para ello aplicaremos un filtro de provincia y la data filtrada la almacenaremos en la variable cod_renip. Posteriormente elegimos la columna en la cual se encuentran los codigos, en este caso es la columna 0 y recuperamos solo los valores con el metodo values, imprimimos el resultado para revisar si la variable es correcta (en algunos casos puede que no existan datos para la provincia) Una vez filtrados los codigos los usaremos para filtrar los datos del data set data_dengue usando el metodo isin para que filtre todos los valores de co_renip y almacenamos el resultado en filtro_prov

```
[104]: cod_renip=data_reinpress[data_reinpress["Provincia"] == provincia]
      cod_renip=cod_renip.iloc[:,0].values
      print(cod_renip)
      #####

      # Lista de valores que quieres seleccionar
      valores_filtrar = cod_renip

      # Filtrar el data_dengue
      filtro_prov = data_dengue[data_dengue['renipress'].isin(valores_filtrar)]

      # Imprimir o trabajar con las filas filtradas
      filtro_prov.head()
```

```
[ 1978   1977 30872 33441 32588 12235   1932   2112 28409 19139 26717 18210
 12400   1943 26927 27663   1981 27210 21001]
```

```
[104]:      ubigeo  renipress diagnostic   ano  semana tipo_dx  edad tipo_edad sexo \
373    200601      1977      A97.0  2013      2      C    21      A      M
4307   200701      1977      A97.0  2013     13      C     8      A      M
5038   200114      1977      A97.0  2013     15      C    11      A      F
5039   200114      1977      A97.0  2013     15      C    16      A      F
5040   200114      1977      A97.0  2013     15      C     8      A      M

      fecha_ini
373    2013-01-10
4307   2013-03-30
5038   2013-04-12
5039   2013-04-12
```

5040 2013-04-13

Una vez filtrada la ubicacion de los datos filtraremos el rango de fechas, en este caso decidimos filtrar por semana epidemiologica y año

```
[109]: prov_semana = filtro_prov[(filtro_prov['semana'] > 8) & (filtro_prov['semana'] ≤ 18) & (filtro_prov['ano'] == 2023)]

prov_semana= prov_semana.assign(casos=1)
prov_semana.head()
```

```
[109]:      ubigeo  renipress diagnostic  ano  semana tipo_dx  edad tipo_edad  \
355001  200105      2112      A97.1  2023      10      C      8      A
355043  200114      1977      A97.0  2023      10      C     26      A
355158  200105      2112      A97.1  2023      10      C      5      A
355168  200107      2112      A97.0  2023      10      C     53      A
355555  200114      1977      A97.0  2023      10      C     27      A

      sexo  fecha_ini  casos
355001   F  2023-03-05      1
355043   F  2023-03-05      1
355158   F  2023-03-06      1
355168   M  2023-03-06      1
355555   M  2023-03-08      1
```

Verificamos que las semanas hayan sido filtradas correctamente y si existen datos de esas semanas, en este caso solo tenemos datos desde la semana 10 a la 18

```
[110]: semana=prov_semana.iloc[:,4]

valores_unicos_numpy = np.unique(semana)
print("Valores únicos con NumPy:", valores_unicos_numpy)
```

Valores únicos con NumPy: [10 11 12 13 14 15 16 17 18]

1.1.2 Agrupando valores por fecha

Utilizando el metodo groupby agrupamos los casos para conseguir el numero de casos confirmados por semana

```
[118]: filtro_semana = prov_semana.groupby(['semana', 'ano'])['casos'].sum().
      ↪reset_index()
filtro_semana
```

```
[118]:      semana  ano  casos
0         10  2023      5
1         11  2023     12
2         12  2023     43
```

3	13	2023	60
4	14	2023	34
5	15	2023	34
6	16	2023	68
7	17	2023	95
8	18	2023	128

1.1.3 Ahora filtramos el data set social

Usando la variable provincia filtramos data_social

```
[116]: data_social.head()
data_filtrada=data_social[data_social["Provincia"] == provincia]
data_filtrada.head()
```

```
[116]: Departamento Provincia semana Fecha (inicia) Temp. Media (°C) \
40 Piura PIURA 9 2023-02-26 27,27
41 Piura PIURA 10 2023-03-05 25,16
42 Piura PIURA 11 2023-03-12 25,46
43 Piura PIURA 12 2023-03-19 25,99
44 Piura PIURA 13 2023-03-26 25,49

Precipitación (mm) Humedad (%) PobrezaTotal(%) PobrezaExtrema(%)
40 83,75 66,93 14322066,9 1895291,5
41 108,06 83 14322066,9 1895291,5
42 55,48 84,46 14322066,9 1895291,5
43 62,67 81,69 14322066,9 1895291,5
44 63,02 84,65 14322066,9 1895291,5
```

1.1.4 Creacion del data set final

Juntamos el data set filtro_semana que contiene el numero de casos junto con el data_social que contiene datos climaticos y los cruzamos mediante la columna semana

```
[119]: data_final = filtro_semana.merge(data_filtrada[['semana', 'Temp. Media (°C)',
↳ 'Precipitación (mm)', 'Humedad (%)', 'PobrezaTotal(%)']], on='semana',
↳ how='left')

data_final.head()
```

```
[119]: semana ano casos Temp. Media (°C) Precipitación (mm) Humedad (%) \
0 10 2023 5 25,16 108,06 83
1 11 2023 12 25,46 55,48 84,46
2 12 2023 43 25,99 62,67 81,69
3 13 2023 60 25,49 63,02 84,65
4 14 2023 34 25,12 81,76 89,12
```

	PobrezaTotal(%)
0	14322066,9
1	14322066,9
2	14322066,9
3	14322066,9
4	14322066,9

1.2 Exportar data a un archivo cvs

Exportamos el `data_final` como `dataFinal.csv`, omitimos el `index` ya que no sera necesario. Este comando crea un cvs en el mismo directorio de origen del notebook.

```
[121]: data_final.to_csv('dataFinal.csv', index=False)
```