


```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import lxml
```

Cargue los datos de incidentes viales reportados por C5 (Viales\_2022\_2024.csv en Github) y realice lo siguiente:

```
ruta = 'https://raw.githubusercontent.com/jamc88/AD-MCMAI/refs/heads/main/Datos/Viales_2022_2024.csv'
df = pd.read_csv(ruta, chunksize=10_000)
df
```

 <pandas.io.parsers.readers.TextFileReader at 0x7c704aa45050>

```
for chunk in pd.read_csv(ruta, chunksize=10_000):
    print(chunk.columns)
    break
```

 Index(['fecha\_creacion', 'hora\_creacion', 'tipo\_incidente\_c4', 'incidente\_c4',  
'alcaldia\_inicio', 'clas\_con\_f\_alarma', 'tipo\_entrada',  
'alcaldia\_catalogo', 'colonia\_catalogo', 'longitud', 'latitud'],  
dtype='object')

a) Obtenga un Dataframe con los tipos de incidentes y su frecuencia. Obtenga un gráfico que ilustre esto.

```
#a) Obtenga un Dataframe con los tipos de incidentes y su frecuencia
tot = pd.Series([], dtype='int64')
```

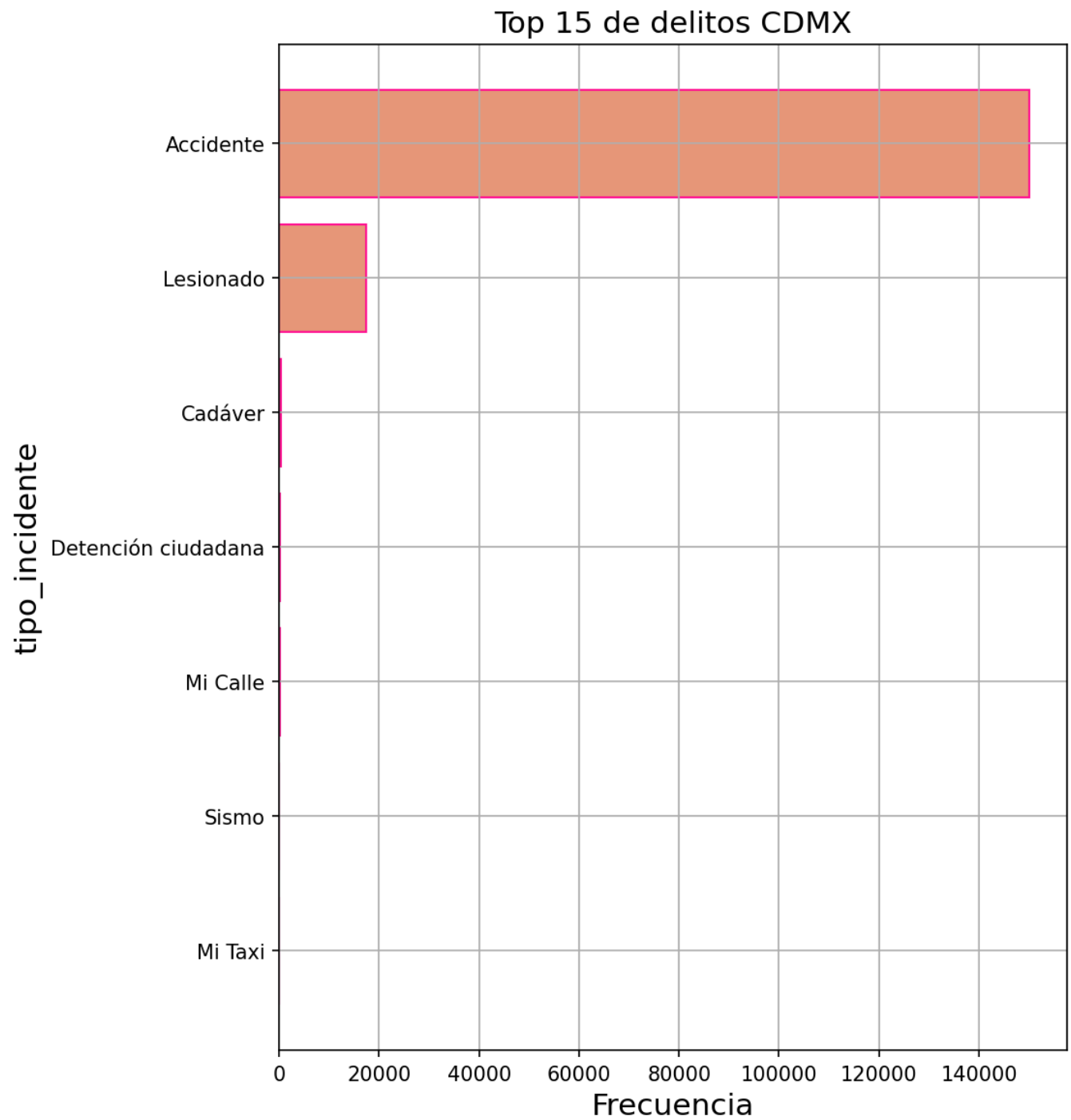
```
for piece in df:
    tot = tot.add(piece['tipo_incidente_c4'].value_counts(), fill_value=0)
    tot = tot.sort_values(ascending=False)
```

```
incidentes = pd.DataFrame(tot, columns=['Frecuencia'])
incidentes
```



	Frecuencia
tipo_incidente_c4	
<b>Accidente</b>	150097.0
<b>Lesionado</b>	17356.0
<b>Cadáver</b>	353.0
<b>Detención ciudadana</b>	120.0
<b>Mi Calle</b>	105.0
<b>Sismo</b>	36.0
<b>Mi Taxi</b>	20.0

```
plt.figure(figsize=(7,9), dpi=150)
plt.barh(incidentes.index, incidentes['Frecuencia']
        ,facecolor='#E9967A', edgecolor='#FF1493')
plt.title('Top 15 de delitos CDMX', fontsize = 15)
plt.xlabel('Frecuencia', fontsize = 15)
plt.ylabel('tipo_incidente', fontsize = 15)
plt.gca().invert_yaxis()
plt.grid(True)
plt.show()
```



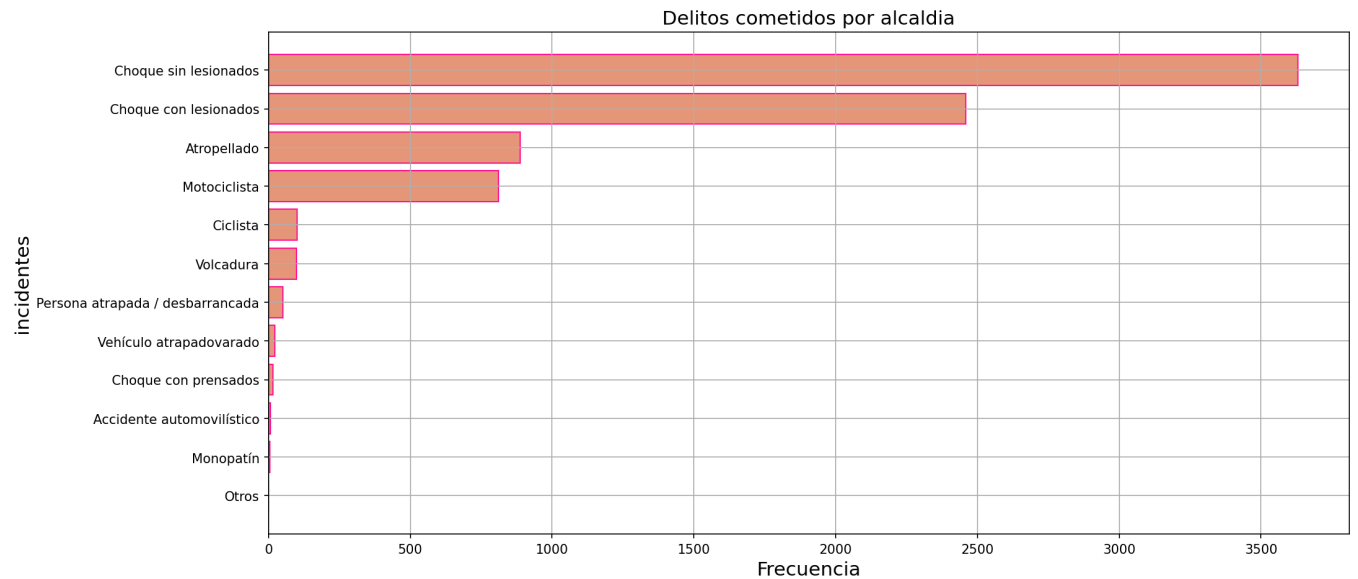
b) Obtenga un Dataframe con los incidentes y su frecuencia. Obtenga un gráfico que ilustre esto.

```
incidentes1 = piece.groupby('incidente_c4').size().reset_index(name='frecuencia_incidentes')
incidentes1 = incidentes1.sort_values(by='frecuencia_incidentes', ascending=False)
incidentes1
```



	incidente_c4	frecuencia_incidentes
4	Choque sin lesionados	3629
2	Choque con lesionados	2458
1	Atropellado	887
7	Motociclista	812
5	Ciclista	100
11	Volcadura	99
9	Persona atrapada / desbarrancada	51
10	Vehículo atrapado varado	23
3	Choque con prensados	15
0	Accidente automovilístico	8
6	Monopatín	4
8	Otros	1

```
#grafica de Data Frame de incidentes_c4
plt.figure(figsize=(15,7), dpi=150)
plt.barh(incidentes1['incidente_c4'], incidentes1['frecuencia_incidentes'],
         ,facecolor='#E9967A', edgecolor='#FF1493')
plt.title('Delitos cometidos por alcaldia', fontsize = 15)
plt.xlabel('Frecuencia', fontsize = 15)
plt.ylabel('incidentes', fontsize = 15)
plt.gca().invert_yaxis()
plt.grid(True)
```



c) Obtenga un gráfico de barras que iluste el número de incidentes por alcaldía (columna 'alcaldia\_inicio')

```
#Obtenga un gráfico de barras que iluste el número de incidentes por alcaldía (columna 'alcaldia_
incidentes2 = piece.groupby('alcaldia_inicio').size().reset_index(name='frecuencia_incidentes')
incidentes2 = incidentes2.sort_values(by='frecuencia_incidentes', ascending=False)
incidentes2
```



	alcaldia_inicio	frecuencia_incidentes
8	IZTAPALAPA	1311
6	GUSTAVO A. MADERO	947
5	CUAUHTEMOC	829
14	VENUSTIANO CARRANZA	610
13	TLALPAN	598
3	COYOACAN	586
2	BENITO JUAREZ	558
0	ALVARO OBREGON	544
10	MIGUEL HIDALGO	480
7	IZTACALCO	373
1	AZCAPOTZALCO	345
15	XOCHIMILCO	335
12	TLAHUAC	244
9	LA MAGDALENA CONTRERAS	143
4	CUAJIMALPA DE MORELOS	126
11	MILPA ALTA	58

```

conteo = piece.groupby(['alcaldia_inicio', 'incidente_c4']).size().reset_index(name='frecuencia_i
conteo = conteo.sort_values(by='frecuencia_incidentes', ascending=False)
conteo

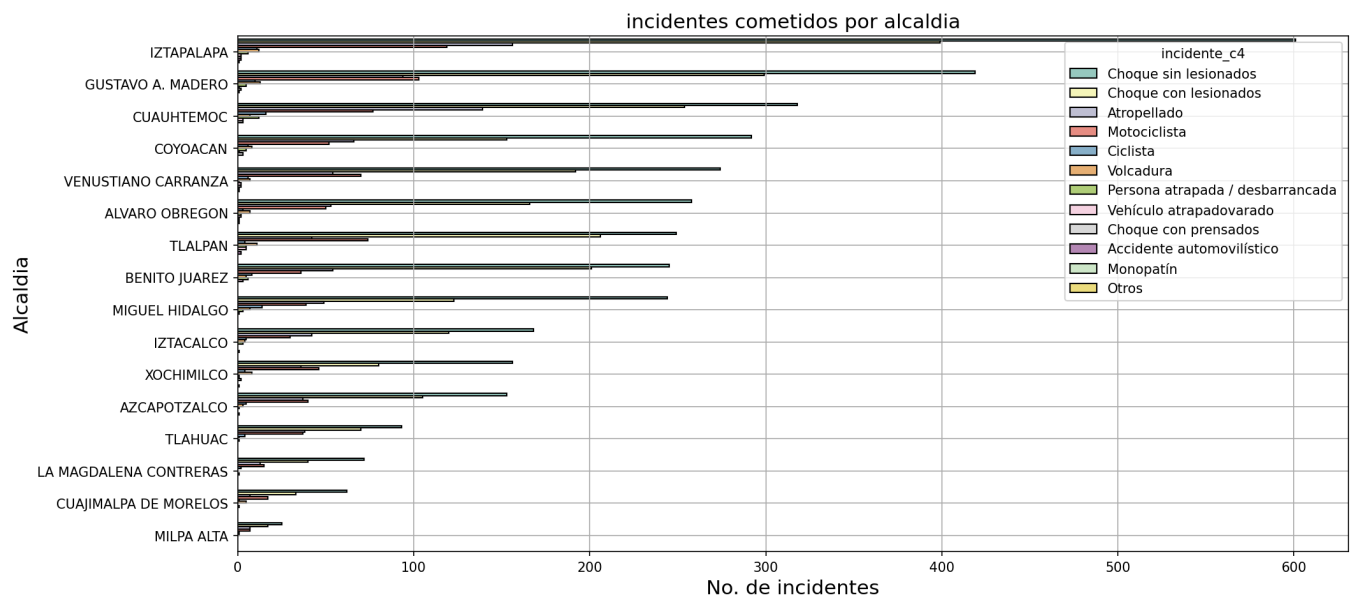
```



	alcaldia_inicio	incidente_c4	frecuencia_incidentes
74	IZTAPALAPA	Choque sin lesionados	601
56	GUSTAVO A. MADERO	Choque sin lesionados	419
72	IZTAPALAPA	Choque con lesionados	399
46	CUAUHTEMOC	Choque sin lesionados	318
54	GUSTAVO A. MADERO	Choque con lesionados	299
...	...	...	...
125	VENUSTIANO CARRANZA	Persona atrapada / desbarrancada	1
123	VENUSTIANO CARRANZA	Monopatín	1
134	XOCHIMILCO	Otros	1
135	XOCHIMILCO	Persona atrapada / desbarrancada	1
136	XOCHIMILCO	Vehículo atrapadovarado	1

138 rows × 3 columns

```
#Obtenga un gráfico de barras que ilustre el número de incidentes por alcaldía (columna 'alcaldia_
plt.figure(figsize=(15,7), dpi=150)
sns.barplot(data=conteo, y='alcaldia_inicio', x='frecuencia_incidentes',hue='incidente_c4',palett
plt.title('incidentes cometidos por alcaldia', fontsize = 15)
plt.xlabel('No. de incidentes', fontsize = 15)
plt.ylabel('Alcaldia', fontsize = 15)
#plt.gca().invert_yaxis()
plt.grid(True)
```



d) Obtenga una serie temporal con el número de atropellados por semana y grafique la serie

```
df_atropellados = df[df['incidente_c4'] == 'Atropellado']
```



```
-----  
TypeError                                Traceback (most recent call last)  
/tmp/ipython-input-122-3443500574.py in <cell line: 0>()  
----> 1 df_atropellados = df[df['incidente_c4'] == 'Atropellado']  
  
TypeError: 'TextFileReader' object is not subscriptable
```

```
df_atropellados = df[df['incidente_c4'] == 'Atropellado'].copy()
```





	fecha_creacion	hora_creacion	tipo_incidente_c4	incidente_c4	alcaldia_inicio	clas
0	2021-12-29	23:21:20	Lesionado	Atropellado	NaN	
5	2022-01-01	09:05:53	Lesionado	Atropellado	IZTAPALAPA	
7	2022-01-01	18:42:30	Lesionado	Atropellado	XOCHIMILCO	
12	2021-12-31	20:49:40	Lesionado	Atropellado	CUAUHTEMOC	
49	2022-01-01	03:43:28	Lesionado	Atropellado	GUSTAVO A. MADERO	
...	...	...	...	...	...	...
168056	2024-02-29	13:28:35	Lesionado	Atropellado	CUAUHTEMOC	
168062	2024-02-29	15:10:29	Lesionado	Atropellado	CUAUHTEMOC	
168069	2024-02-29	17:05:25	Lesionado	Atropellado	IZTACALCO	
168070	2024-02-29	17:12:49	Lesionado	Atropellado	IZTAPALAPA	
168085	2024-02-29	19:48:51	Lesionado	Atropellado	CUAJIMALPA DE MORELOS	

17559 rows × 11 columns

```
df_atropellados = df_atropellados.set_index('fecha_creacion')
atropellados_sem= df_atropellados.resample('W').size()
atropellados_sem
```

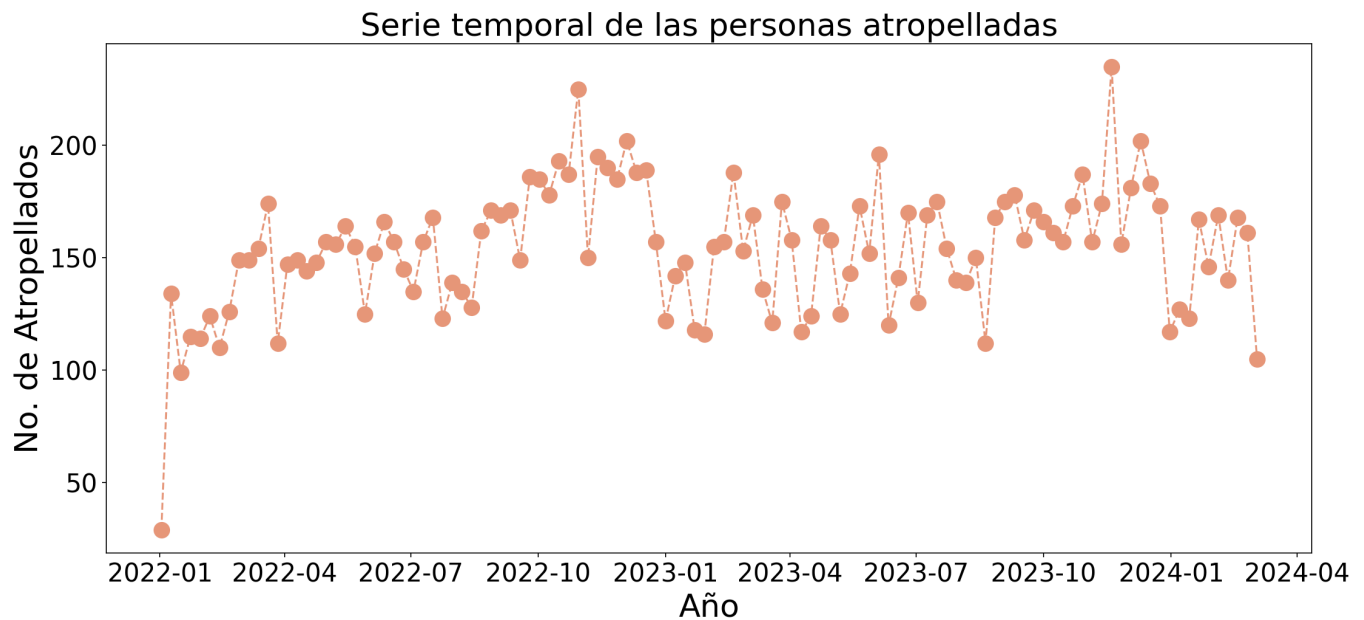


```
-----
KeyError                                Traceback (most recent call last)
/tmp/ipython-input-123-3365104961.py in <cell line: 0>()
----> 1 df_atropellados = df_atropellados.set_index('fecha_creacion')
      2 atropellados_sem= df_atropellados.resample('W').size()
      3 atropellados_sem

/usr/local/lib/python3.11/dist-packages/pandas/core/frame.py in set_index(self, keys, drop,
append, inplace, verify_integrity)
    6120
    6121         if missing:
-> 6122             raise KeyError(f"None of {missing} are in the columns")
    6123
    6124         if inplace:

KeyError: "None of ['fecha_creacion'] are in the columns"
```

```
#grafica de serie temporal de los atropellados
plt.figure(figsize=(15,7), dpi=150)
plt.plot(atropellados_sem.index, atropellados_sem.values, color='#E9967A',linestyle='dashed', mar
plt.title('Serie temporal de las personas atropelladas', fontdict={'fontsize':25})
plt.xlabel('Año', fontdict={'fontsize':25})
plt.ylabel('No. de Atropellados', fontdict={'fontsize':25})
plt.tick_params(labelsize=20)
plt.tight_layout()
plt.show()
```

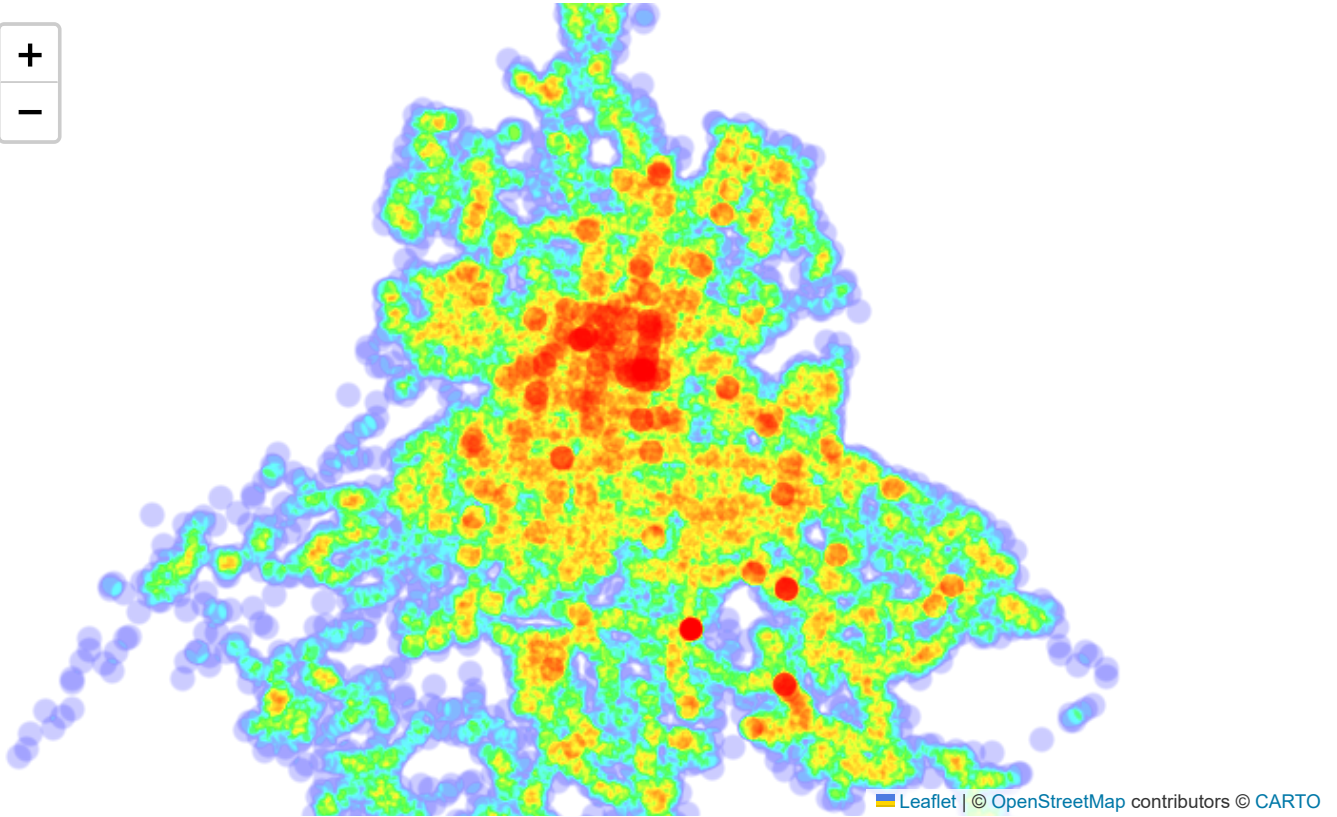
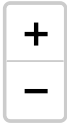


e) Obtenga un mapa de calor (HeatMap) que muestre la ubicación geográfica de los reportes de personas atropelladas, utilizando las coordenadas de latitud y longitud asociadas a cada incidente.

```
import folium

from folium.plugins import HeatMap
import folium
```

```
m = folium.Map(location=[19.4, -99.1],  
                zoom_start=11, tiles="Cartodb Positron")  
  
HeatMap(df_atropellados[["latitud", "longitud"]].dropna(), radius=7,  
        blur=1, min_opacity=0.2).add_to(m)  
  
m
```



Ejercicio 2. (50 puntos) Extraiga la tabla de "Terremotos de mayor magnitud" del siguiente enlace:

[https://es.wikipedia.org/wiki/Anexo:Terremotos\\_de\\_mayor\\_magnitud](https://es.wikipedia.org/wiki/Anexo:Terremotos_de_mayor_magnitud)

y realice lo siguiente:

a) Obtenga el siguiente Dataframe.

```
ruta = 'https://es.wikipedia.org/wiki/Anexo:Terremotos_de_mayor_magnitud'
df1 = pd.read_html(ruta)
df1
```



[ Visualiza un mapa con todas las coordenadas

Sitúa todas las coordenadas utilizando OpenStreetMap Sitúa hasta 200 coordenadas utilizando Bing

- 0 Exporta todas las coordenadas en formato KML
- 1 Exporta todas las coordenadas en formato GeoRSS
- 2 Exporta todas las coordenadas en formato GPX
- 3 Sitúa en el mapa todas las coordenadas microfo...

```
,
      N.º          Fecha y hora UTC          Magnitud \
0      1      22 de mayo de 1960, 15:11      9,5 MW[2][3]
1      2      26 de diciembre de 2004, 00:58      9,3 MW[5]
2      2      28 de marzo de 1964, 03:36      9,2 MW[3][7]
3      4      11 de marzo de 2011, 05:46      9,1 MW[9]
4      4      4 de noviembre de 1952, 16:58      9,0 MW[11][12]
5      4      13 de agosto de 1868, 21:30      9,0 MW[15]
6      4      28 de octubre de 1746, 22:30      9,0 MW
7      4      26 de enero de 1700, 21:30      9,0 MW
8      9      27 de febrero de 2010, 03:34      8,8 MW
9      9      31 de enero de 1906, 15:36      8,8 MW[21]
10     9      25 de noviembre de 1833, 20:00      8,8 MW
11     12     1 de noviembre de 1755, 10:16      8,7 MW[26][27]
12     12      8 de julio de 1730, 04:45      8,7 Mw
13     12     14 de febrero de 1619, 11:30      8,7 Mw
14     15     11 de abril de 2012, 15:38      8,6 MW
15     15     28 de marzo de 2005, 23:09      8,6 MW
16     15      9 de marzo de 1957, 14:22      8,6 MS[30]
17     15      15 de agosto de 1950      8,6 MW
18     15     10 de noviembre de 1922, 23:53      8,6 MW
19     15     19 de octubre de 1609, 19:00      8,6 MW
20     15     28 de marzo de 1787, 11:30      8,6 MW[32]
```