```
# importing libraries
from urllib.request import urlretrieve
import pandas as pd
import os
import re
from datetime import datetime
import pytz
#import geocoder
import folium
import time
import urllib.request
import json
#import geopandas
import matplotlib.pyplot as plt
from datetime import datetime
```

```
!pip install -q condacolab
import condacolab
condacolab.install()

```

https://stackoverflow.com/questions/65324533/geopandas-in-google-colab

```
!pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting geopandas
  Downloading geopandas-0.10.2-py2.py3-none-any.whl (1.0 MB)
     |████████████████████████████████| 1.0 MB 12.8 MB/s
Requirement already satisfied: pandas>=0.25.0 in /usr/local/lib/python3.7/dist-packages (from geopandas) (1.3.5)
Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-packages (from geopandas) (1.8.2)
Collecting fiona>=1.8
  Downloading Fiona-1.8.21-cp37-cp37m-manylinux2014_x86_64.whl (16.7 MB)
     |████████████████████████████████| 16.7 MB 18.5 MB/s
Collecting pyproj>=2.2.0
  Downloading pyproj-3.2.1-cp37-cp37m-manylinux2010_x86_64.whl (6.3 MB)
     |████████████████████████████████| 6.3 MB 44.0 MB/s
Collecting munch
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (7.1.2)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (1.15.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (2022.6.15)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (21.4.0)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (57.4.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopandas) (2022
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopanc
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopandas) (1.21
Installing collected packages: munch, cligj, click-plugins, pyproj, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.8.21 geopandas-0.10.2 munch-2.5.0 pyproj-3.2.1
```

```
# Activo Google Drive

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Tiene formato de código
```

Nivell 1

L'analista ha d'assegurar-se que els registres consisteixen en una gamma completa de missatges i s'interpreten segons el context. Els elements de registre han d'estandaritzar-se, utilitzant els mateixos termes o terminologia, per evitar confusions i proporcionar cohesió.

Com Científic de Dades se t'ha proporcionat accés als registres-Logs on queda registrada l'activitat de totes les visites a realitzades a la pàgina web de l'agència de viatges "akumenius.com".

```
# Abro archivo

df = list(open('/content/drive/MyDrive/Web_access_log-akumenius.com.txt'))
df[200:210]
```

```
['www.akumenius.com 180.76.5.31 - - [23/Feb/2014:03:15:03 +0100] "GET /destinos-baratos/destinos-caracteristicas/hoteles-baratos-en-Zurich_SUIZA-con-Alquiler-de-silla-de-ruedas HTTP/1.1" 200 8360 "-" "Mozilla/5.0 (compatible; Baiduspider/2.0; +http://www.baidu.com/search/spider.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:03 +0100] "GET /hoteles-baratos/ofertas-hotel-Capitolio-Apartamentos-Turisticos-en-Merida-207271b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:06 +0100] "GET /destinos-baratos/destinosEstrelles/hoteles-en-Gary_ESTADOS%20UNIDOS-con-4-estrellas HTTP/1.1" 200 8812 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:06 +0100] "GET /hoteles-baratos/ofertas-hotel-Guitart-Termes-La-Collada-en-La%20Molina-6833b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:08 +0100] "GET /hoteles-baratos/ofertas-hotel-Sol-Palmeras-en-Varadero-63381b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:10 +0100] "GET /hoteles-baratos/ofertas-hotel-The-Suites-At-San-Roque-Club-en-Sotogrande-195247b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:13 +0100] "GET /hoteles-baratos/ofertas-hotel-Punta-Serena-en-Costa%20Alegre-54540b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:13 +0100] "GET /destinos-baratos/destinosEstrelles/hoteles-en-Sant\'Arcangelo_ITALIA-con-5-estrellas HTTP/1.1" 200 8816 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:15 +0100] "GET /destinos-baratos/hoteles-baratos/ofertas-hotel-Hotel-Vantis-en-Riga-215949b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n',
 'www.akumenius.com 66.249.76.216 - - [23/Feb/2014:03:15:17 +0100] "GET /hoteles-baratos/ofertas-hotel-Tune-Hotel-Klia---LCCT-Airport-en-Airport%20Hotels-175955b-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)" VLOG=-\n']
```

Creo un fichero de IPs chopeadas del original

```
print('NUmero de registros:')
len(df)
```

```
NUmero de registros:
261873
```

Exercici 1

Estandaritza, identifica i enumera cada un dels atributs / variables de l'estructura de l'arxiu "Web_access_log-akumenius.com" que trobaràs al repositori de GitHub "Data-sources".

```
#2.- Separo la parte central.

def custom_split(str_to_split):
    separatorsCentro = "[", "]", "\s", ",",' ','+', ')', '(', '"'
    # create regular expression dynamically
    regular_exp = '|'.join(map(re.escape, separatorsCentro))

    return re.split(regular_exp, str_to_split)
```

```
def separoCorchete ( textoCortar):
  separatorsCentro = "]"
  regular_exp = '|'.join(map(re.escape, separatorsCentro))

  return re.split(regular_exp, textoCortar)

```

```
def arregloFecha(x):
    '''
    Parses datetime with timezone formatted as:
        `[day/month/year:hour:minute:second zone]`
    '''
    dt = datetime.strptime(x, '%d/%b/%Y:%H:%M:%S')

    return dt
```

```
#2.- Separo la parte central.

def custom_split_Dcha(str_to_split):
    #separatorsCentro = ["[", "]", "\s", ",",' ','+', ')', '(', '"'
    separatorsCentro = "[", "]", '"'



    # create regular expression dynamically
    regular_exp = '|'.join(map(re.escape, separatorsCentro))

    return re.split(regular_exp, str_to_split)
```

```
df1 = df.copy()
data = {'virtual_host': [],    'IPs': [],    'Fecha': [],    'request': [],  'status': [], 'size': [], '

df_salida = pd.DataFrame(data)
print('lineas total ------> ', len(df1))

for s, textoLinea  in enumerate(df1):
  #if len(textoLinea) == 23:
    #print ('---', s, textoLinea)
  problematico= 0
  if s<5000:
    textin = separoCorchete(textoLinea)
    if len(textin) > 2:
      textin[1] = textin[1] +textin[2]


    print(s, len(textin), textin[1])
    texto = custom_split(textin[0])

    textoDcha= custom_split_Dcha(textin[1])
    #for p, q in enumerate(textoDcha):
     #print(p,q)


    nueva_fila = {'request': textoDcha[1],
            'status': textoDcha[2],
            'size': textoDcha[3],
            'referer': textoDcha[5]}

```

```
2531 2  "GET /hoteles-baratos/ofertas-hotel-Clarion-Suites-en-YUMA-3720750t-destinos.html HTTP/1.1" 404 3100 "-" "Mozilla/
2532 2  "GET /destinos-caracteristicas/hoteles-baratos-en-Kusadasi_TURQUIA-con-Teatro- HTTP/1.1" 200 7189 "-" "Mozilla/5.0
2533 2  "GET /destinos-baratos/hoteles-en-Benidorm_ESPA%C3%91A HTTP/1.1" 200 34757 "http://www.akumenius.com/" "Mozilla/5.
2534 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Benavente_ESPA%C3%91A-con-Servicio-de-recepci%C3%B3n-24-horas HTT
2535 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Aix-Les-Bains_FRANCIA-con-%C3%81rea-de-juegos HTTP/1.1" 200 7915
2536 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Montecatini_ITALIA-con-Bebida-inclu%C3%ADda HTTP/1.1" 200 8327 "-
2537 2  "GET /destinos-baratos/hoteles-en-Estambul_TURQUIA HTTP/1.1" 200 61780 "http://www.akumenius.com/" "Mozilla/5.0 (c
2538 2  "GET /hoteles-baratos/ofertas-hotel-Howard-Johnson-Inn-en-YUMA-3724524t-destinos.html HTTP/1.1" 404 3100 "-" "Mozi
2539 2  "GET /destinos-baratos/destinos-caracteristicas/hoteles-baratos-en-Sunshine-Coast_AUSTRALIA-con-Billar-americano-y,
```

```
2540 2  "GET /destinosCaracteristicas/hoteles-baratos-en-COSTA-ESMERALDA-(CERDE%C3%91A)_ITALIA-con-Servicios HTTP/1.1" 200
2541 2  "GET /destinos-caracteristicas/hoteles-baratos-en-COSTA-DE-CANTABRIA_ESPA%C3%91A-con-Limpieza-semanal HTTP/1.1" 20(
2542 2  "GET /destinos-baratos/hoteles-en-Tenerife_ESPA%C3%91A HTTP/1.1" 200 47368 "http://www.akumenius.com/" "Mozilla/5.(
2543 2  "GET /destinos-baratos/destinos-caracteristicas/hoteles-baratos-en-Ciclades_GRECIA-con-Windsurf HTTP/1.1" 200 7899
2544 2  "GET /destinosCaracteristicas/hoteles-baratos-en-SEYCHELLES_SEYCHELLES-con-Ventilador-de-techo HTTP/1.1" 200 10413
2545 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Leeuwarden_HOLANDA-con-Acceso-a-Internet-sin-cables HTTP/1.1" 200
2546 2  "GET /destinos-baratos/hoteles-en-Valencia_ESPA%C3%91A HTTP/1.1" 200 31858 "http://www.akumenius.com/" "Mozilla/5.(
2547 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Navia_ESPA%C3%91A-con-Ba%C3%B1o-privado HTTP/1.1" 200 8641 "-" "M(
2548 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Maia_PORTUGAL-con-Ba%C3%B1o-privado HTTP/1.1" 200 8464 "-" "Mozil
2549 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Loja_ESPA%C3%91A-con-Ba%C3%B1o-privado HTTP/1.1" 200 8260 "-" "Mo
2550 2  "GET /destinos-baratos/hoteles-en-C%C3%B3rdoba_ESPA%C3%91A HTTP/1.1" 200 26730 "http://www.akumenius.com/" "Mozill
2551 2  "GET /hoteles-baratos/ofertas-hotel-Diplomat-Hotel-Tunis-en-T%C3%9ANEZ-3721509t-destinos.html HTTP/1.1" 404 3100 "
2552 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Lourinha_PORTUGAL-con-Ba%C3%B1o-privado HTTP/1.1" 200 8323 "-" "M(
2553 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Nabeul_T%C3%9ANEZ-con-Ba%C3%B1o-privado HTTP/1.1" 200 8216 "-" "M(
2554 2  "GET /destinos-baratos/hoteles-en-Berl%C3%ADn_ALEMANIA HTTP/1.1" 200 7691 "http://www.akumenius.com/" "Mozilla/5.0
2555 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Lagos_NIGERIA-con-Ba%C3%B1o-privado HTTP/1.1" 200 8174 "-" "Mozil
2556 2  "GET /destinosCaracteristicas/hoteles-baratos-en-Kranj_ESLOVENIA-con-Ba%C3%B1o-privado HTTP/1.1" 200 8285 "-" "Moz
2557 2  "GET /hoteles-baratos/ofertas-hotel-Polus-Palace-Thermal-Golf-Club-en-G%C3%96D-69972t-destinos.html HTTP/1.1" 404
2558 2  "GET /destinos-caracteristicas/hoteles-baratos-en-Chiang-Mai_TAILANDIA-con-Golf HTTP/1.1" 200 13658 "-" "Mozilla/5
2559 2  "GET /destinos-baratos/hoteles-en-C%C3%A1ceres ESPA%C3%91A HTTP/1.1" 200 25383 "http://www.akumenius.com/" "Mozill
```

Once the lines are printed, we see they are in the lines shows Common Log Format (CLF) :

check https://httpd.apache.org/docs/2.2/logs.html what CLF means

He dividido cada linea es 2 partes para tener mejor control y reducir el numero de accesos que hace con cada "spliter" elemento. El lado Izq va hasta que encuentra el ']' y la derecha es el otro lado y lo separare por ".

```
 1 import re
 2
 3 chopear = input('Quieres abrir el fichero de nuevo y?')
 4 if chopear == 'y' or chopper =='Y':
 5   texto=[]
 6
 7
 8   df1 = df.copy()
 9   data = {'virtual_host': [],    'IPs': [],    'Fecha': [],    'request': [],  'status': [], 'size': [],
10
11   df_salida = pd.DataFrame(data)
12   print('lineas total ------> ', len(df1))
13
14   #for s, textoLinea  in enumerate(df1):
15   #  print('Hola', s, textoLinea)
16
17   for s, textoLinea  in enumerate(df1):
18     #if len(textoLinea) == 23:
19       #print ('---', s, textoLinea)
20     problematico= 0
21     if s<10000000:
22       textin = separoCorchete(textoLinea)
23
24       #print(s, len(textin), textin[0])
25       texto = custom_split(textin[0])
26       if len(textin) > 2:
27         textin[1] = textin[1] +textin[2]
```

```
28
29      nueva_fila = {'virtual_host': texto[0],
30            'IPs': texto[1],
31            'Fecha': arregloFecha(texto[5])}  # Lo tranformo en fecha
32
33      textoDcha= custom_split_Dcha(textin[1])
34    #for p, q in enumerate(textoDcha):
35     #print(p,q)
36
37
38    nueva_fila = {'virtual_host': texto[0],
39            'IPs': texto[1],
40            'Fecha': arregloFecha(texto[5]),
41            'request': textoDcha[1],
42            'status': textoDcha[2],
43            'size': textoDcha[3],
44            'referer': textoDcha[5]}
45
46      #for p,q in enumerate(texto):
47        #print(p, q)
48
49    if s%10000 ==0 :
50            print(s, 'Z--->', nueva_fila)
51            print()
52    #df_salida=pd.DataFrame.from_dict(nueva_fila, orient='index')
53    df_salida= df_salida.append(nueva_fila, ignore_index=True)
54
55  # Guardo la informacion en drive
56  path = ('/content/drive/MyDrive/01_COLAB/' +'direcionesIP_otro.csv')
57  df_salida.to_csv(path)
58  #
59  #df_salida.type
60  print('Forma ', df_salida.shape)
61
62  df_salida.head(-20)
63
```

```
Quieres abrir el fichero de nuevo y?y
lineas total ------>  261873
0 Z---> {'virtual_host': 'localhost', 'IPs': '127.0.0.1', 'Fecha': datetime.datetime(2014, 2, 23, 3, 10, 31), 'request': '(

10000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '144.76.95.232', 'Fecha': datetime.datetime(2014, 2, 23, 10, 24,

20000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '180.76.5.171', 'Fecha': datetime.datetime(2014, 2, 23, 15, 14, 1

30000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '157.56.92.146', 'Fecha': datetime.datetime(2014, 2, 23, 19, 29, 1

40000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.76.216', 'Fecha': datetime.datetime(2014, 2, 23, 23, 40, (

50000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '87.221.174.146', 'Fecha': datetime.datetime(2014, 2, 24, 9, 13, 4

60000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '83.37.239.190', 'Fecha': datetime.datetime(2014, 2, 24, 12, 43,

70000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '83.61.241.37', 'Fecha': datetime.datetime(2014, 2, 24, 14, 5, 58

80000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.76.216', 'Fecha': datetime.datetime(2014, 2, 24, 19, 0, 1(

90000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '31.4.182.223', 'Fecha': datetime.datetime(2014, 2, 25, 0, 42, 19

100000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '88.14.201.73', 'Fecha': datetime.datetime(2014, 2, 25, 11, 14, 9

110000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '88.11.41.24', 'Fecha': datetime.datetime(2014, 2, 25, 15, 5, 35

120000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.76.216', 'Fecha': datetime.datetime(2014, 2, 25, 16, 43,

130000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.76.216', 'Fecha': datetime.datetime(2014, 2, 25, 18, 33,

140000 Z---> {'virtual_host': 'localhost', 'IPs': '127.0.0.1', 'Fecha': datetime.datetime(2014, 2, 25, 21, 33, 28), 'reque

150000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.76.216', 'Fecha': datetime.datetime(2014, 2, 26, 4, 29, 1

160000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '79.151.59.10', 'Fecha': datetime.datetime(2014, 2, 26, 12, 27,

170000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '84.122.81.236', 'Fecha': datetime.datetime(2014, 2, 26, 17, 30,

180000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '66.249.75.148', 'Fecha': datetime.datetime(2014, 2, 26, 23, 6,
```

```
190000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '87.218.139.190', 'Fecha': datetime.datetime(2014, 2, 27, 11, 24

200000 Z---> {'virtual_host': 'localhost', 'IPs': '127.0.0.1', 'Fecha': datetime.datetime(2014, 2, 27, 15, 11, 46), 'reque

210000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '81.39.17.184', 'Fecha': datetime.datetime(2014, 2, 27, 16, 46, 4

220000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '186.82.214.90', 'Fecha': datetime.datetime(2014, 2, 27, 21, 14,

230000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '37.14.223.169', 'Fecha': datetime.datetime(2014, 2, 28, 8, 57,

240000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '2.136.76.13', 'Fecha': datetime.datetime(2014, 2, 28, 20, 18, 50

250000 Z---> {'virtual_host': 'www.akumenius.com', 'IPs': '31.4.190.156', 'Fecha': datetime.datetime(2014, 3, 1, 12, 58, 2

260000 Z---> {'virtual_host': 'localhost', 'IPs': '127.0.0.1', 'Fecha': datetime.datetime(2014, 3, 1, 23, 28, 21), 'request

Forma  (261873, 8)
```

```
1 print(type(df_salida))
2 print(df_salida.dtypes)
3 print('Tmaño: ', df_salida.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
virtual_host            object
IPs                     object
Fecha           datetime64[ns]
request                 object
status                  object
size                    object
referer                 object
user_agent             float64
dtype: object
Tmaño:  (261873, 8)
```

Una vez chopeado el fichero principal. LO guardo asi no es necesario repetir este proceso que ha durado varias horas.

Abrir fichero de IPs creado

```
1 path = ('/content/drive/MyDrive/01_COLAB/' +'direcionesIP.csv')
2 df=  pd.read_csv(path)
3 print('Tamaño antes de tomar una muestra: ', df.shape)
4
5 #df = df.sample(frac = 0.5)
6 #df = df[1294:1305]
7 print(type(df))
8 print('Tamaño: ', df.shape)
```

```
Tamaño antes de tomar una muestra:  (261873, 9)
<class 'pandas.core.frame.DataFrame'>
Tamaño:  (261873, 9)
```

Exercici 2

Neteja, preprocesa, estructura i transforma (dataframe) les dades del registre d'Accés a la web.

Lo primero que voy a hacer es minimizar la base de datos. La manera que se me ha ocurrido es poner un contador por Ip y asi si hay IP repetidas solo la buscará una vez y tendre la cantidad en el df

```
1 data_ip =df["IPs"].value_counts().rename_axis('ip').reset_index(name="visits")
2 data_ip.head()
```

```
1 !pip install geocoder
```

```
1 import geocoder
2
3 ip = geocoder.ip("157.55.32.183")
4 print(ip.city)
```

```
print(ip)
print(ip.latlng)
```

Exercici 3

Geolocalitza les IP's.

```

location = geocoder.ip('66.249.76.216')
```

Este proceso dura varios minutos y lo que hago es trabajar contra un fichero que he creado y asi lo hago más eficiente.

```
posicion = []

for index, value in data_ip["ip"].items():
    with urllib.request.urlopen("https://geolocation-db.com/jsonp/"+value,timeout=500) as url:
                data = url.read().decode()
                data = data.split("(")[1].strip(")")
                data = json.loads(data)
                posicion.append(data)

```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-12-35f4654154b7> in <module>()
      2
      3 for index, value in data_ip["ip"].items():
----> 4     with urllib.request.urlopen("https://geolocation-db.com/jsonp/"+value,timeout=500) as url:
      5                 data = url.read().decode()
      6                 data = data.split("(")[1].strip(")")

                                 14 frames
/usr/lib/python3.7/ssl.py in do_handshake(self, block)
   1137             if timeout == 0.0 and block:
   1138                 self.settimeout(None)
-> 1139             self._sslobj.do_handshake()
   1140         finally:
   1141             self.settimeout(timeout)

KeyboardInterrupt:
```

SEARCH STACK OVERFLOW

```
posicion = pd.DataFrame(posicion)

type(posicion)
```

```
pandas.core.frame.DataFrame
```

```
!pip install geopandas
import geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting geopandas
  Downloading geopandas-0.10.2-py2.py3-none-any.whl (1.0 MB)
     |████████████████████████████████| 1.0 MB 7.6 MB/s
Requirement already satisfied: pandas>=0.25.0 in /usr/local/lib/python3.7/dist-packages (from geopandas) (1.3.5)
Collecting pyproj>=2.2.0
  Downloading pyproj-3.2.1-cp37-cp37m-manylinux2010_x86_64.whl (6.3 MB)
     |████████████████████████████████| 6.3 MB 43.6 MB/s
Collecting fiona>=1.8
  Downloading Fiona-1.8.21-cp37-cp37m-manylinux2014_x86_64.whl (16.7 MB)
     |████████████████████████████████| 16.7 MB 366 kB/s
Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-packages (from geopandas) (1.8.2)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (1.15.0)
Collecting munch
  Downloading munch-2.5.0-py2.py3-none-any.whl (10 kB)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (21.4.0)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (7.1.2)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (57.4.0)
```

```
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from fiona>=1.8->geopandas) (2022.6.15)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopandas) (1.21
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopandas) (2022
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=0.25.0->geopanc
Installing collected packages: munch, cligj, click-plugins, pyproj, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.8.21 geopandas-0.10.2 munch-2.5.0 pyproj-3.2.1
```

```
1 posicion['cantidad'] = data_ip['visits']
2 posicion = posicion[posicion['state']!= 'Not found']
3 posicion
```

	country_code	country_name	city	postal	latitude	longitude	IPv4	state	cantidad
0	US	United States	None	None	37.751	-97.822	66.249.76.216	None	46382
1	ES	Spain	Madrid	28045	40.4165	-3.7026	80.28.221.123	Madrid	14725
3	ES	Spain	Bormujos	41930	37.3736	-6.0723	217.125.71.222	Seville	5201
4	US	United States	None	None	37.751	-97.822	66.249.75.148	None	3558
5	US	United States	New York	10011	40.7308	-73.9975	162.243.192.191	New York	2927
6	ES	Spain	Valencia	46015	39.4698	-0.3774	62.117.197.230	Valencia	2567
7	ES	Spain	Marbella	29602	36.5154	-4.8858	89.128.176.162	Malaga	1093
8	US	United States	Chicago	60604	41.8785	-87.633	198.143.133.154	Illinois	1045
9	FR	France	None	None	48.8582	2.3387	176.31.255.177	None	1044
10	ES	Spain	Palazuelos de Eresma	40194	40.9305	-4.0607	80.58.250.94	Segovia	1043

```
# Guardo el archivo con la geoposiciones

pathPosicion = ('/content/drive/MyDrive/01_COLAB/' +'GEOdirecionesIP.csv')
```

13	ES	Spain	Coin	29100	36.6595	-4.7564	195.57.124.71	Malaga	897

```
sobreEscribir = input('Quiere sobre escribir el fichero? y/Y')
if (sobreEscribir=='Y' or sobreEscribir=='y'):
    posicion.to_csv(path)
```

```
Quiere sobre escribir el fichero? y/Yn
```

Si ya hemos geolocacliadas todas las Ips, entonces solo tenemos que descar el fichero y no hace falta repetir todos los pasos anteriores

19	ES	Spain	Barcelona	08027	41.3888	2.159	80.37.230.56	Barcelona	746

```
# Abro archivo

posicion = pd.read_csv(pathPosicion)
posicion[200:210]
```

	Unnamed: 0	country_code	country_name	city	postal	latitude	longitude	IPv4	state	cantidad	geomet
200	201	ES	Spain	Madrid	28028	40.4165	-3.7026	95.63.2.45	Madrid	176	PO (-3.7 40.41
201	202	US	United States	NaN	NaN	47.6062	-122.3321	157.56.92.174	Washington	176	PO (-122.3 47.60
202	203	ES	Spain	Madrid	28034	40.4165	-3.7026	85.48.101.193	Madrid	175	PO (-3.7 40.41
203	204	ES	Spain	Vitoria-Gasteiz	01010	42.8500	-2.6727	85.84.176.196	Araba / Álava	174	PO (-2.6 42.
204	ES	Spain	Palma	07004	39.5694	2.6502	95.59.143.223	Balearic Islands	493	PO

```
posicion = pd.DataFrame(posicion)

type(posicion)
```

```
pandas.core.frame.DataFrame
```

```
posicion.head(-1)
```

	Unnamed: 0	country_code	country_name	city	postal	latitude	longitude	IPv4	state	cantidad	geometr
0	0	US	United States	NaN	NaN	37.7510	-97.8220	66.249.76.216	NaN	46382	POI (-97.8 37.75
1	1	ES	Spain	Madrid	28045	40.4165	-3.7026	80.28.221.123	Madrid	14725	POI (-3.70 40.416
2	3	ES	Spain	Bormujos	41930	37.3736	-6.0723	217.125.71.222	Seville	5201	POI (-6.07

```
# Elimino todos los registros que no tengan informacion
# por ejemplo los 'not found'

posicion['cantidad'] = data_ip['visits']
posicion = posicion[posicion['latitude']!= 'Not found']
posicion.shape
```

```
(2919, 11)
```

```
posicion['latitude'].astype(float, errors = 'raise')
posicion.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2919 entries, 0 to 2918
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Unnamed: 0    2919 non-null   int64
 1   country_code  2906 non-null   object
 2   country_name  2906 non-null   object
 3   city          1804 non-null   object
 4   postal        1631 non-null   object
 5   latitude      2919 non-null   float64
 6   longitude     2919 non-null   float64
 7   IPv4          2919 non-null   object
 8   state         2200 non-null   object
 9   cantidad      2919 non-null   int64
 10  geometry      2919 non-null   object
dtypes: float64(2), int64(2), object(7)
memory usage: 273.7+ KB
```

```
gdf = geopandas.GeoDataFrame(
    posicion, geometry=geopandas.points_from_xy(posicion.longitude,posicion.latitude))
```

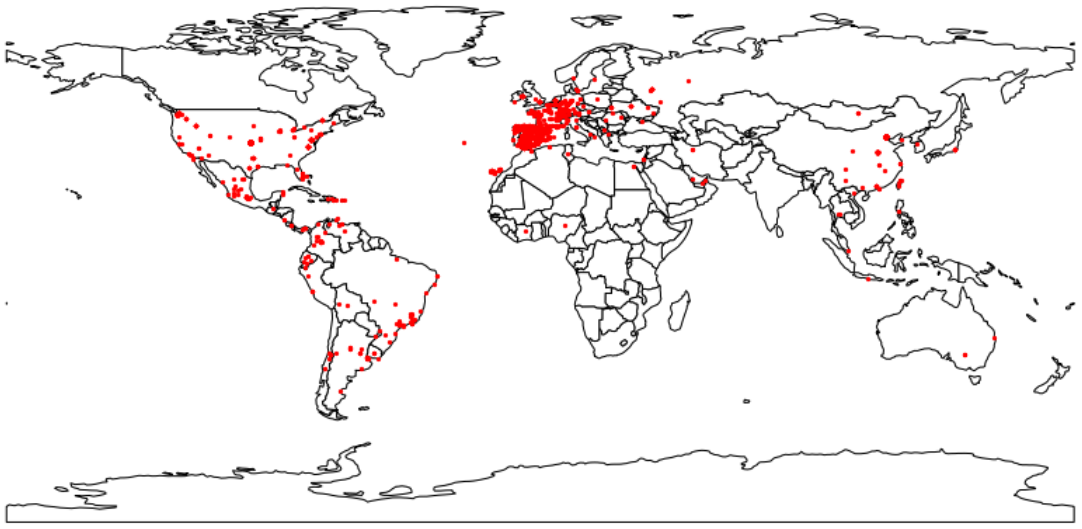
```
gdf.head(-1)
```

	Unnamed: 0	country_code	country_name	city	postal	latitude	longitude	IPv4	state	cantidad	geome
0	0	US	United States	NaN	NaN	37.7510	-97.8220	66.249.76.216	NaN	46382	PO (-97.82 37.751
1	1	ES	Spain	Madrid	28045	40.4165	-3.7026	80.28.221.123	Madrid	14725	PO (-3.70 40.416
2	3	ES	Spain	Bormujos	41930	37.3736	-6.0723	217.125.71.222	Seville	13892	PO (-6.07 37.373
3	4	US	United States	NaN	NaN	37.7510	-97.8220	66.249.75.148	NaN	5201	PO (-97.82 37.751
4	5	US	United States	New York	10011	40.7308	-73.9975	162.243.192.191	New York	3558	PO (-73.99 40.730
...	...	...	...	...	...	...	...	...	...	...	...

```
world = geopandas.read_file(geopandas.datasets.get_path('naturalearth_lowres'))
```

```
fig, ax = plt.subplots(figsize=(15,10))

```

```
ax.set_aspect('equal')

world.plot(ax=ax, color='white', edgecolor='black')
ax.set_axis_off()

gdf.plot(ax=ax, marker='o', color='red', markersize=5)
plt.show()
```



```
! pip install folium
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: folium in /usr/local/lib/python3.7/dist-packages (0.8.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from folium) (2.23.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from folium) (2.11.3)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/dist-packages (from folium) (0.5.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from folium) (1.21.6)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from folium) (1.15.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->folium) (2.0.1)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->folium) (2022.6.1
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->folium) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from reques
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->folium) (2.10)
```

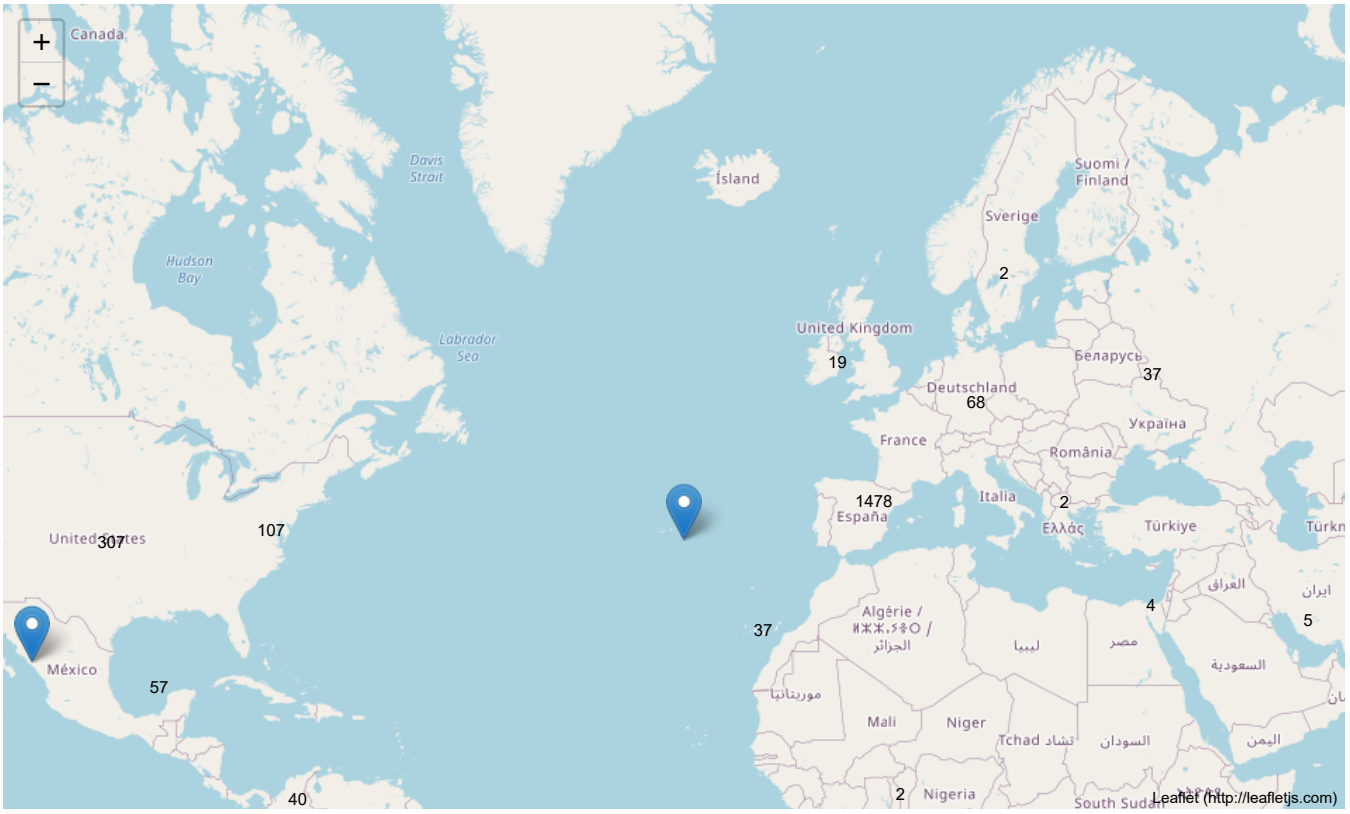
```
from folium.plugins import MarkerCluster

m = folium.Map(location=[40.965, -5.664], zoom_start=3)
mc = MarkerCluster()
```

```
#! pip install geocoder
```

+ Código + Texto

```
for i in range(0,len(posicion)):
    mc.add_child(folium.Marker(
        location=[posicion.iloc[i]["latitude"],posicion.iloc[i]["longitude"]],
        popup=str(posicion.iloc[i]["cantidad"]),))
```

```
m.add_child(mc)
```



Exercici 3

Mostra'm la teva creativitat, Sorprèn-me fes un pas més enllà amb l'anàlisi anterior.

https://datasciencesphere.com/project/track-location-ip-address-python-geocoder/

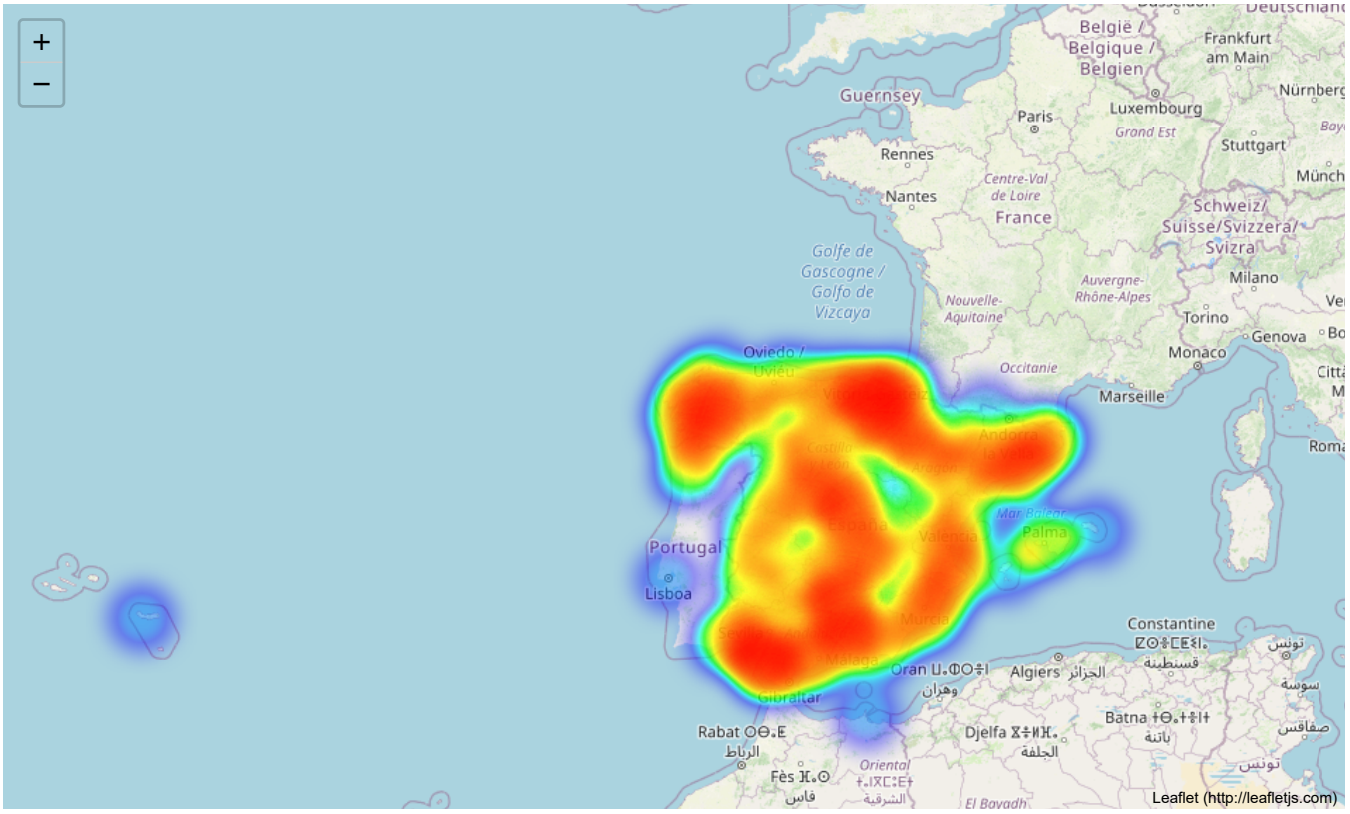
```
Spain = posicion[(posicion['country_name'] == 'Spain') | (posicion['country_name'] == 'Portugal')]
```

```
from folium.plugins import MarkerCluster

m = folium.Map(location=[40.965, -5.664], zoom_start=7)
mc = MarkerCluster()
```

```
Spain.iloc[10]
```

```
Unnamed: 0                         19
country_code                       ES
country_name                    Spain
city                         Barcelona
postal                           08027
latitude                       41.3888
longitude                        2.159
IPv4                      80.37.230.56
state                        Barcelona
cantidad                           782
geometry         POINT (2.159 41.3888)
Name: 18, dtype: object
```

```
posiciones=[]

from folium import plugins

mapa = folium.Map(location=[40.4167, -3.70325], zoom_start=4.9) #, width= 800, height =700)

for i in range(0,len(Spain)):
  posiciones.append([Spain.iloc[i]["latitude"],Spain.iloc[i]["longitude"]])


mapa.add_child(plugins.HeatMap(posiciones[:]))#.add_to(feature_group)

```



✓ 4 s completado a las 12:47