In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#from ip2geotools.databases.noncommercial import DbIpCity   not worked
import re
import requests as rq

logs_column = ["DOMAIN",'IP', 'NO1', "NO2",'TIME', "NO3", "MET_PROT_VERSION", "SERVER-ANS", "NO4", "NO5", 'BROWSER', "LOG"

p = pd.read_csv('Web_access_log-akumenius.com.txt', sep='\s+', names = logs_column)

#columnes amb informació
df = p[["DOMAIN",'IP','TIME', "MET_PROT_VERSION", "SERVER-ANS", 'BROWSER']]


df["SERVER-ANS"] = df["SERVER-ANS"].apply(str)

print(type(df["TIME"][0]))
pe = re.compile(r'\[')
df["TIME"] = [pe.sub('', x) for x in df["TIME"]]
df["TIME"] = [re.sub(r'\:',' ', x, count = 1) for x in df["TIME"]]
df['TIME']= pd.to_datetime(df['TIME'], format="%d/%b/%Y %H:%M:%S")
print((df["TIME"][0]))
print(type(df["TIME"][0]))

#comprovar que les les IP són vàlides
import socket

for element in df['IP'].unique():
    try:
        socket.inet_aton(element)
    except socket.error:
        print(df[df['IP'] == element])
```

```
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\3542864638.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df["SERVER-ANS"] = df["SERVER-ANS"].apply(str)
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\3542864638.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df["TIME"] = [pe.sub('', x) for x in df["TIME"]]
<class 'str'>
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\3542864638.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df["TIME"] = [re.sub(r'\:',' ', x, count = 1) for x in df["TIME"]]
2014-02-23 03:10:31
<class 'pandas._libs.tslibs.timestamps.Timestamp'>
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\3542864638.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['TIME']= pd.to_datetime(df['TIME'], format="%d/%b/%Y %H:%M:%S")
```

In [2]:

```
#Geolocalitzar les IP
#realitzar les peticions amb una velocitat menor de 45 request per minut
import time

city = {}
country = {}
latitude = {}
longitude = {}
for element in df['IP'].unique():
    if element != "127.0.0.1":
        response = rq.get("http://ip-api.com/json/"+element+"?lang=es").json()
        city[element] = response['city']
        country[element] = response['country']
        latitude[element] = response['lat']
        longitude[element] = response['lon']
        time.sleep(1.5)
    else:
        continue
df['CIUTAT'] =df['IP'].map(city)
df['PAIS'] =df['IP'].map(country)
df['LATITUD'] =df['IP'].map(latitude)
df['LONGITUD'] =df['IP'].map(longitude)
```

```
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\1655054823.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['CIUTAT'] =df['IP'].map(city)
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\1655054823.py:20: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['PAIS'] =df['IP'].map(country)
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\1655054823.py:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['LATITUD'] =df['IP'].map(latitude)
C:\Users\Ripley\AppData\Local\Temp\ipykernel_5968\1655054823.py:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-v
iew-versus-a-copy
  df['LONGITUD'] =df['IP'].map(longitude)
```

In [4]:

```
df
```

Out[4]:

	DOMAIN	IP	TIME	MET_PROT_VERSION	SERVER-ANS	BROWSER	CIUTAT	PAIS	LATITUD	LONGITUD
0	localhost	127.0.0.1	2014-02-23 03:10:31	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
1	localhost	127.0.0.1	2014-02-23 03:10:31	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
2	localhost	127.0.0.1	2014-02-23 03:10:31	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
3	localhost	127.0.0.1	2014-02-23 03:10:31	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
4	localhost	127.0.0.1	2014-02-23 03:10:31	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	..
261868	www.akumenius.com	5.255.253.53	2014-03-02 03:05:39	GET / HTTP/1.1	200	Mozilla/5.0 (compatible; YandexBot/3.0; +http:...	Moscú	Rusia, (la) Federación de	55.7332	37.583
261869	www.akumenius.com	74.86.158.107	2014-03-02 03:09:52	HEAD / HTTP/1.1	200	Mozilla/5.0+ (compatible; UptimeRobot/2.0; http...	Dallas	Estados Unidos	32.7767	-96.805
261870	localhost	127.0.0.1	2014-03-02 03:10:18	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
261871	localhost	127.0.0.1	2014-03-02 03:10:18	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na
261872	localhost	127.0.0.1	2014-03-02 03:10:18	OPTIONS * HTTP/1.0	200	Apache (internal dummy connection)	NaN	NaN	NaN	Na

261873 rows × 10 columns

In [3]:

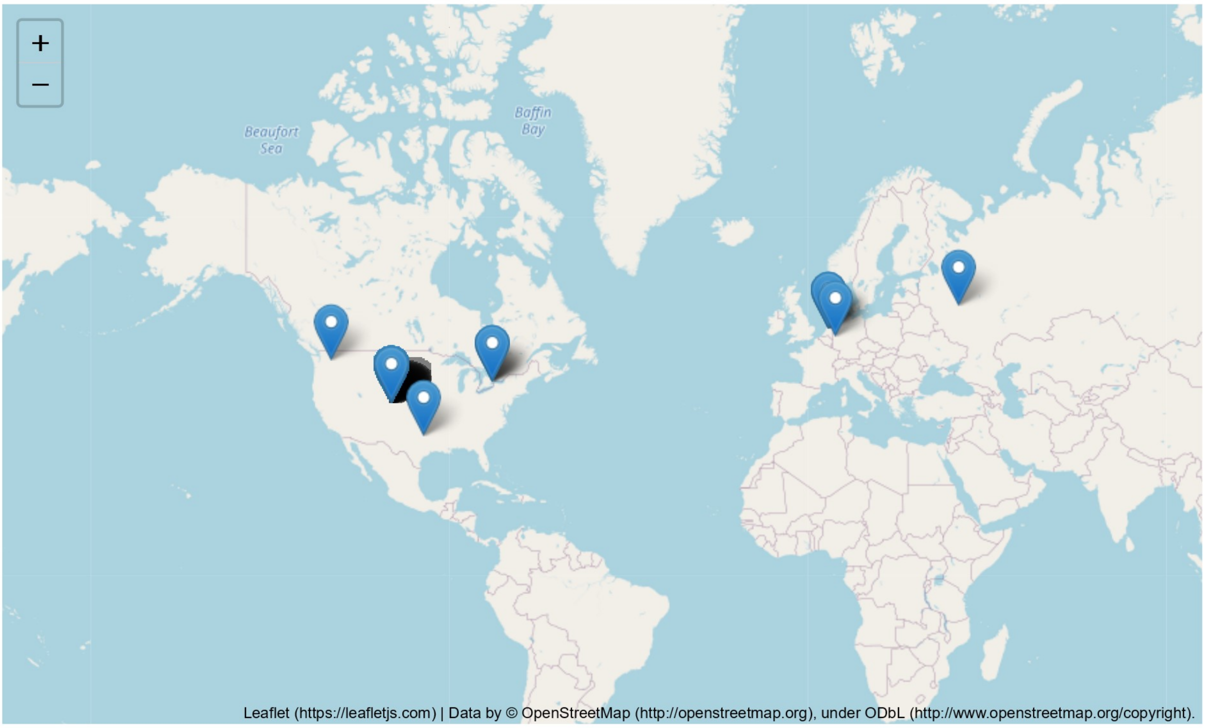
```
import folium
from folium import plugins

df2 = df[df['CIUTAT'].notna()][:200]
mapa = folium.Map(location=[0.00, 0.00], titles = "Mapbox Bright", zoom_scale = 12)

df2.apply(lambda row: folium.Marker(location = [row["LATITUD"], row["LONGITUD"]], tooltip= row["CIUTAT"]).add_to(mapa), ax

mapa
```

Out[3]:



Leaflet (https://leafletjs.com) | Data by © OpenStreetMap (http://openstreetmap.org), under ODbL (http://www.openstreetmap.org/copyright).