Práctica 1

Carga de librerías

El siguiente código carga los paquetes necesarios para la práctica y lee el fichero de datos que usaremos.

In [2]:
```
import pandas as pd
import numpy as np
```

In [8]:
```
df = pd.read_csv("../csv/dataset.csv", delimiter=";") #change path if needed
```

Contexto

He recogido la información de la página web USGS - Earthquake Hazard Program

Este sitio web es un sitio que publica información acerca de los terremotos y los ubica en un mapa interactivo. Para acceder al mapa, se debe visitar el siguiente enlace: https://earthquake.usgs.gov/earthquakes/map/?extent=-89.58992,-357.1875&extent=89.58992,717.1875&range=search&listOnlyShown=true&baseLayer=terrain&timeZone=utc&search=%7B%22name%22:%22Search%20Results%22,%22params%22:%7B%22starttime%22:%221900-01-01%2000:00:00%22,%22minmagnitude%22:7,%22orderby%22:%22time%22%7D%7D

Para mis tareas de web scrapping, he llevado a cabo las siguiente comprobaciones:

- No existe fichero robots.txt *
- La página web está servida por un servidor web NGiNX *
- La página web utiliza código avanzado de javascript y utiliza técnicas de descarga de información a medida que se navega sobre el paginador.
- La página web no ofrece bloqueos de ningún tipo y permite hacer el web scrapping sin problemas.

* En el fichero de código `technologies.py` se encuentran los comandos ejecutados para tales comprobaciones, que dan resultados:

```
{'web-servers': ['Nginx']}
{
  "domain_name": "USGS.GOV",
  "registrar": null,
  "whois_server": null,
  "referral_url": null,
  "updated_date": null,
  "creation_date": null,
  "expiration_date": null,
  "name_servers": null,
  "status": "ACTIVE",
  "emails": "security@usgs.gov",
  "dnssec": null,
  "name": null,
  "org": null,
  "address": null,
  "city": null,
  "state": null,
  "zipcode": null,
  "country": null
}
```

El sitio web ofrece la información necesaria de los terremotos a nivel mundial para poder analizar las tendencias a nivel de frecuencia, magnitud y localización.

Título

El título que he decidido darle al dataset es: **\"Earthquakes worldwide alert\"**

Descripción del dataset

Dataset with data from Earthquake Hazard Program with a subset of the latest earthquakes registered since 1984 with a Magnitude of 2.5+

Representación gráfica

Contenido

Como se puede ver en la siguiente tabla, el dataset contiene los siguientes campos:

- **time**: Indica el momento del terremoto (UTC)
- **place**: Descripción de la ubicación donde ocurrió el terremoto
- **depth**: Radio de alcance del terremoto
- **magnitude**: Magnitud del terremoto

In [9]:
```
df.head(5)
```

Out[9]:

	time	place	depth	magnitude
0	2022-03-31 05:44:01 (UTC)	279 km ESE of Tadine, New Caledonia	10.0 km	7.0
1	2022-03-16 14:36:33 (UTC)	57 km ENE of Namie, Japan	63.1 km	7.3
2	2021-12-29 18:25:51 (UTC)	125 km NNE of Lospalos, Timor Leste	165.5 km	7.3
3	2021-12-14 03:20:23 (UTC)	Flores Sea	14.3 km	7.3
4	2021-11-28 10:52:14 (UTC)	43 km NNW of Barranca, Peru	126.0 km	7.5

El dataset que se ha utilizado contiene un **total de 995 filas** de datos, si bien, el proceso de carga de datos se podría continuar ampliando hasta un total de cerca de 1500, he considerado que esta muestra es suficiente. En cuanto al periodo de los datos, el dataset contiene información de terremetos desde 1984 hasta la actualidad.

El proceso de scrap en este caso requiere de una simulación de navegación web y recoger a cada iteración los elementos del paginador ya que la página web utiliza un paginador de Angular que elimina los registros no visibles del DOM y por tanto, el scrap debe ser dinámico (y es algo lento). Este proceso se describirá a continuación.

El proceso prinicpal tiene el siguiente código (se han eliminado partes no útiles para esta explicación):

```
    pages = 20
    for page in range(1, pages):
        html = selenium_download(page)
        subset = scrap(html)
        print(len(subset))
        print(subset)
        append_csv(FILENAME, subset)
        data.append(subset)
```

Podemos ver como se repite el proceso un total de 20 veces para ir obteniendo página a página los datos del dataset y se van añadiendo al fichero de datos del dataset tras cada iteración.

El proceso tiene dos fases fundamentales para la captura y proceso de información.

La captura de la información se hace en la siguiente función:

```
def selenium_download(page_number):
    '''
    Downloads the page source of a web using selenium webdriver which allows to load
    advanced javascript webpages.
    '''
    url = ...
    browser = webdriver.Firefox()
    browser.get(url)

    # There is an overlay shown with text "Earthquakes loaded" which only appears after data been loaded. Useful.
    class_name = "cdk-global-overlay-wrapper"

    try:
        print("Waiting for data grid container to be loaded")
        elem = WebDriverWait(browser, 30).until(
            EC.presence_of_element_located((By.CLASS_NAME, class_name)) #This is a dummy element
        )
        # Scroll down.
        for i in range(1,page_number):
            scrollview = browser.find_element_by_tag_name("cdk-virtual-scroll-viewport")
            for i2 in range(1,50):
                scrollview.send_keys(Keys.DOWN)
            time.sleep(2)
        time.sleep(5)
    finally:
        print("Data grid container found. Closing driver.")
        html = browser.page_source
        browser.close()
    return html
```

El proceso espera a que se cargue el elemento paginador y entonces, simula un scroll en el elemento web por medio de "pulsar" 50 veces la flecha abajo.

Previamente, podemos ver que la tecnología a utilizar para la descarga es un `webdriver` de selenium, que es una herramienta que permite trabajar con páginas web con avanzado javascript. Las librerías `urllib` son insuficientes para estos casos y no permiten trabajar adecuadamente con página interactivas.

Finalmente, el procedimiento de carga utiliza las librebrías `BeautifulSoup` para procesar el DOM obtenido y obtener los datos requeridos.

Agradecimientos

Los datos han sido obtenidos de la página web "Earthquake Hazard Program" haciendo uso de técnicas de web scrapping como método académico de formación.

El gobierno de USGS publica una API para obtener los terremotos que podría ser utilizada en lugar de hacer uso de web scrapping.

Inspiración

En los últimos tiempos existe una tendencia muy importante a la generación de conciencia social acerca del cambio climático y los efectos devastadores que este conlleva y conllevará a nivel mundial a no ser que se lleven a cabo medidas para reducir la contaminación ambiental y las emisiones de CO2.

Aún cuando en los últimos años las noticias se sobreponen las unas sobre las otras (covid, volcán de la Palma y la invasión rusa sobre Ucrania), debemos seguir teniendo presentes y analizar las diferentes señales que nos manda el planeta.

Es por eso que decidí buscar alguna publicación acerca de desastres naturales y trabajar con estos datos

Licencia

La licencia seleccionada es `CC BY-NC-SA 4.0 License`. Dado que no se puede obtener la licencia del sitio web principal, he considerado que es importante, cuando menos, utilizar una licencia que no permita a un tercero hacer negocio a partir de este dataset.

Código

El código de la práctica se encuentra en un repositorio público de GitHub: https://github.com/mtablado/uoc2022_web_scraping_pec1

Publicación de dataset

El Dataset ha sido publicado en Zenodo: https://zenodo.org/record/6448412#.YlRPR1xBxH4

Vídeo

Enlace de descarga del vídeo: https://drive.google.com/file/d/13WGp3XGZQnywykanlbmQtvGQJJ72CKMsL/view?usp=sharing

Contribuciones

Contribuciones	Firma
Investigación previa	MTL
Redacción de las respuestas	MTL
Desarrollo del código	MTL