Nuria García Fuentes y Alicia Perdices Guerra

Tipología de datos. Práctica 1

**TIPOLOGÍA DE DATOS. PRÁCTICA 1**

# **Descripción de la Práctica.**

En el contexto de la práctica propuesta, y motivadas sobre todo por las ventajas y beneficios que el estudio de datos procedentes de enfermedades y de desastres naturales puede aportar a la comunidad científica, no solo para la prevención, sino para establecer protocolos, hemos decidido estudiar y hacer web scraping de los siguientes recursos:

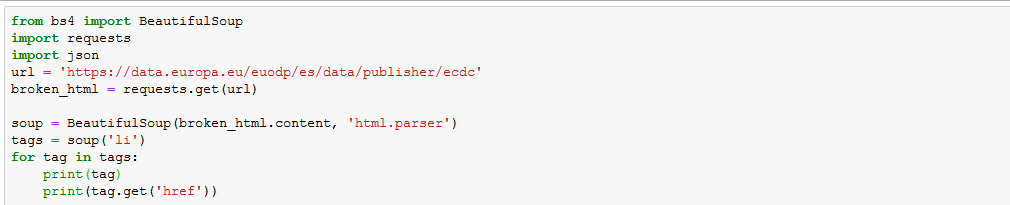
[https://data.europa.eu/euodp/es/data/publisher/ecdc](https://www.google.com/url?q=https://data.europa.eu/euodp/es/data/publisher/ecdc&sa=D&source=hangouts&ust=1584813221920000&usg=AFQjCNFXaBDxR7QEoA9NpKMW7cIcEjJQbQ)

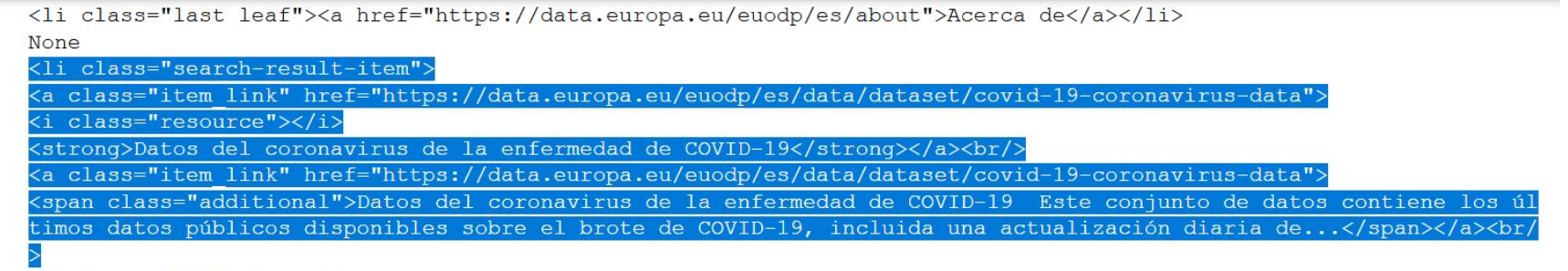
<https://reliefweb.int/disasters>

**Caso enfermedades: Descarga de un archivo xlsx**

De este modo, y con el fin, de descargar, analizar y organizar estos datos de interés, hemos completado el proceso de web scraping de la siguiente forma:

* Se ha evaluado el mapa de los sitios web para localizar el contenido del recurso, poniendo el foco en la estructura de la página, y así poder averiguar dónde y cómo descargar los datos que necesitamos para el estudio.
* Se ha convertido la página en HTML a estructura anidada.
* A continuación, se ha descargado la página web mediante las librerías **Requests** y **BeautifulSoup** en Python.
* Tras examinar las etiquetas, tags, clases de la página, hemos navegado por la estructura hasta el enlace que contiene los datos para posteriormente descargarlos y analizarlos.
  + Obtenemos las etiquetas **<li> con class: search-result-item,** que es donde tenemos las enfermedades y sus links.

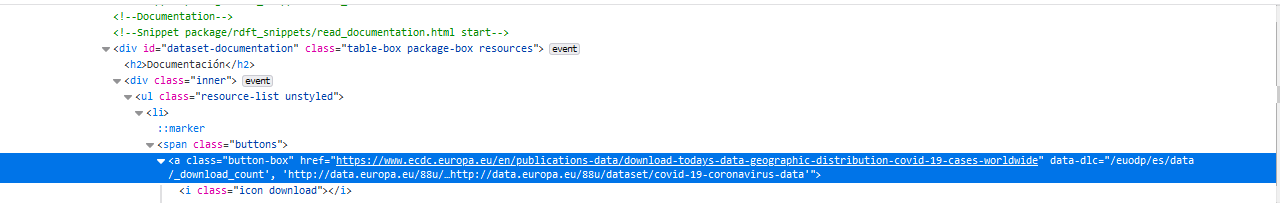


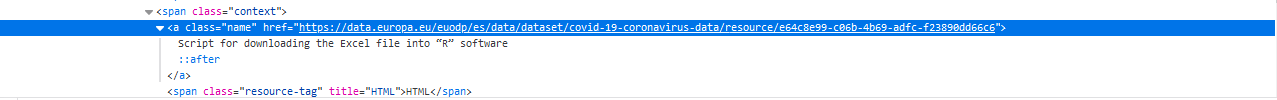


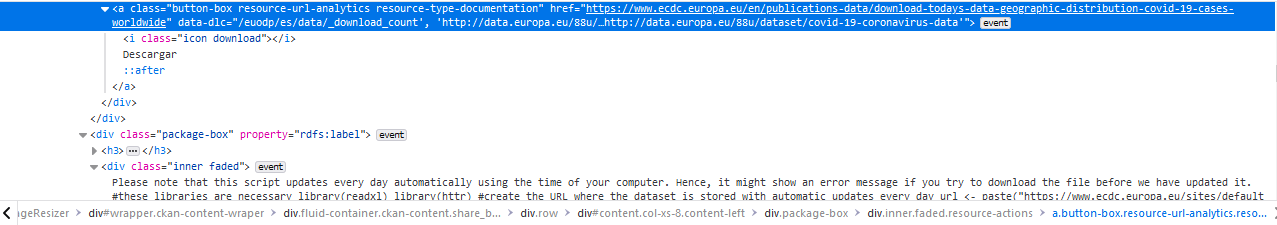
* + Repetimos el proceso para cada link de enfermedad elegida con **tag <a>** y **clase 'item\_link'**.

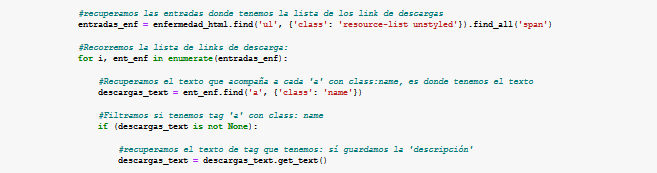


* + Recuperamos las entradas donde tenemos los links de descarga con tag **<ul>** y clase **'resource-list unstyled'**
  + Recorremos la lista de links de descarga **con tag <a>** y **class =’name’**, accediendo primero a la descripción.



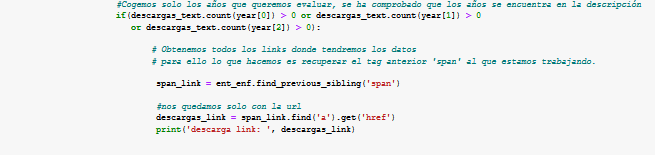




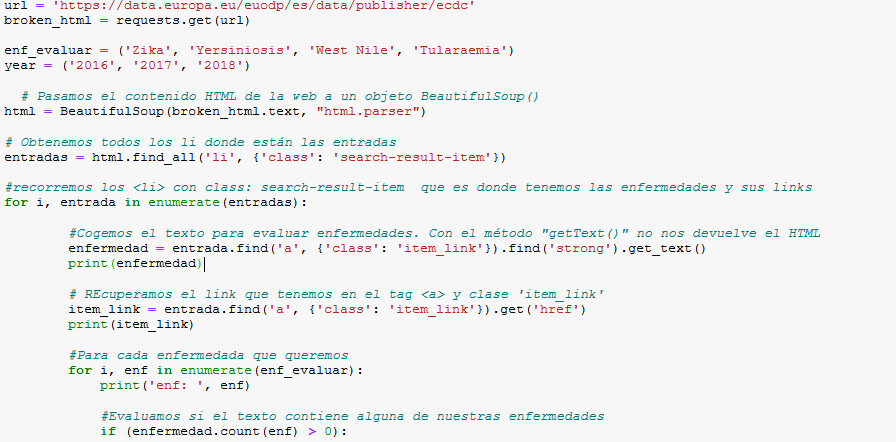


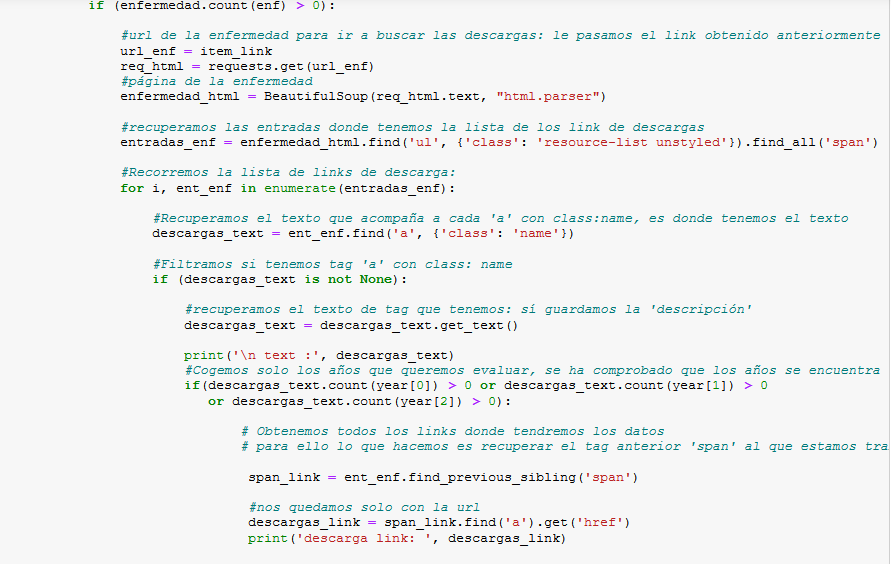
* + Obtenemos los años que queremos evaluar, dato que se encuentra en la descripción. A continuación, obtenemos con **previous\_sibling** los links donde se encuentras los datos (**tag ‘span’ anterior** **a esta descripción**).





En definitiva, tenemos:



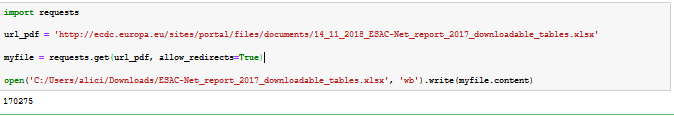


* Una vez obtenidos los enlaces desde los que nos descargaremos los datos, procedemos con su descarga y su análisis.
* En resumen:

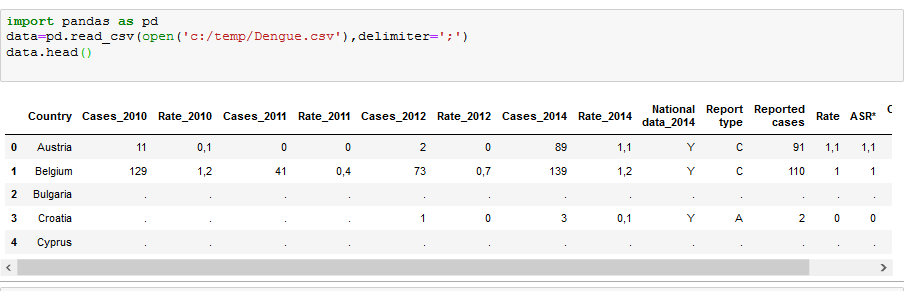


A continuación, vamos a hacer un pequeño análisis para los datos obtenidos del enlace de la enfermedad del **dengue**.

* Descargamos los datos en local.



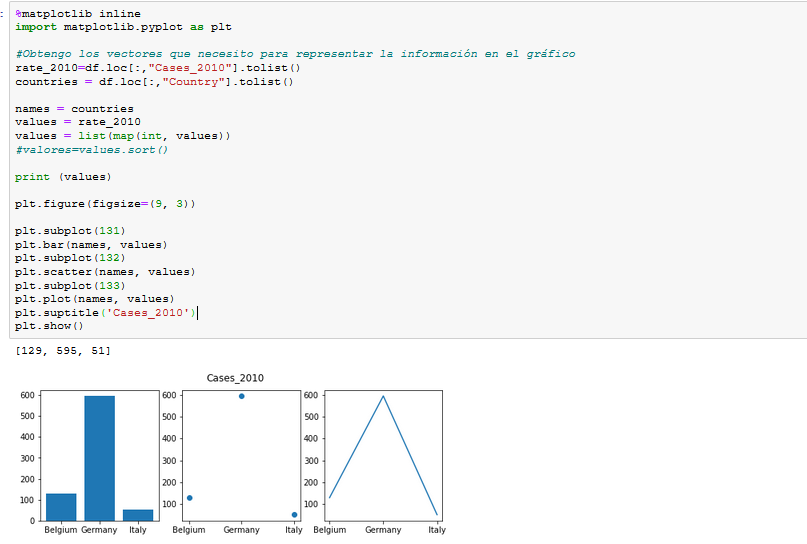
* Importamos el archivo descargado:



* Creamos un dataframe para hacer una comparativa de los casos de dengue en Italia, Alemania y Bélgica en 2010.









Podemos comprobar que en Alemania se han diagnosticado más casos de Dengue que en Italia y Bélgica. Observamos también que en 2011 hay un descenso de los casos en todos los países.

**Caso enfermedades: Descarga de un archivo pdf**

(…………….)

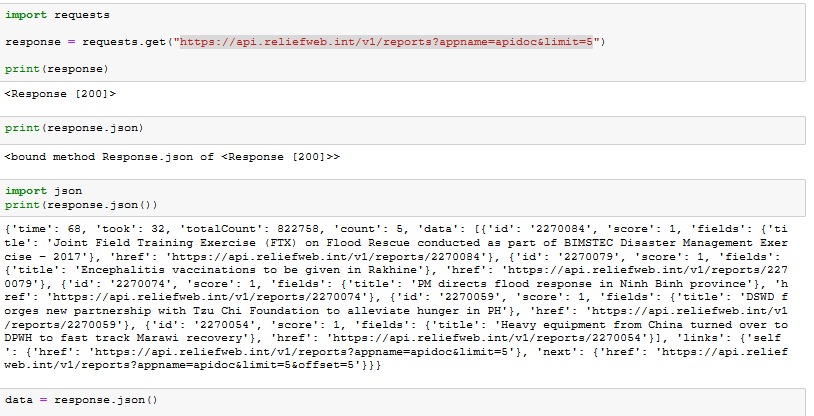
**Caso desastres naturales. Descarga pdf con mapa de la actividad del volcán Vanuatu en diciembre de 2018.**

En este caso utilizaremos las técnicas del web scraping, para obtener imágenes sobre elementos geográficos que puedan ser potenciales áreas peligrosas.

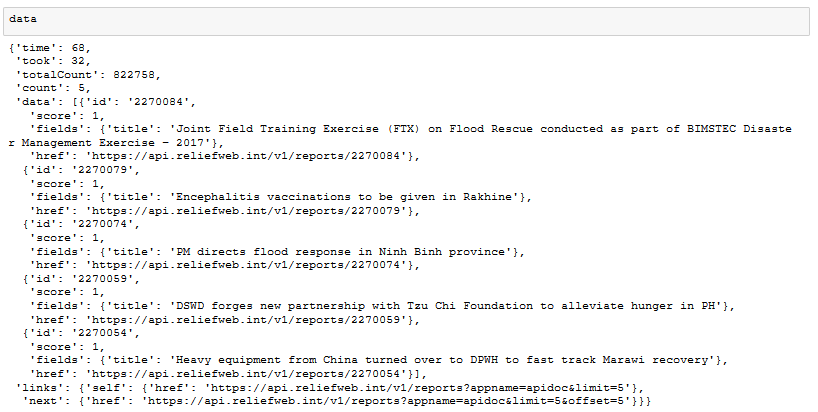
Por ejemplo, el volcán Monte Yasur que se encuentra en la isla de Tanna, y forma parte del estado de Vanuatu en el Pacífico Sur, es uno de los pocos volcanes activos que podemos encontrar. Nos podría interesar obtener mapas de actividad, para establecer protocolos de salvamento en el área.

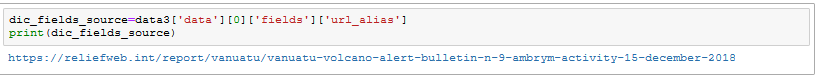
Para ello y desde el recurso <https://reliefweb.int/disasters>, comenzamos el proceso de web scraping del siguiente modo:

Comprobamos que todo el contenido de la página ReliefWeb es accesible mediante su API (<https://apidoc.rwlabs.org/>). Es de acceso público mediante solicitudes HTTP y devuelve datos JSON.

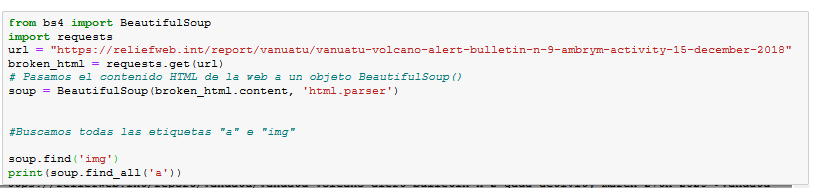


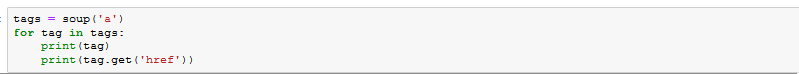
A continuación, navegamos por la estructura JSON, que consta de una colección de pares nombre /valor (objeto) y una lista ordenada de valores (array), hasta obtener el enlace con la información deseada. (Ver archivo **web\_scarping\_desastres\_Garcia\_Perdices\_Prac1.ipynb)**





Después, paseamos el contenido HTML a un objeto BeautifulSoup(). Buscamos todos **los enlaces e imágenes** con etiqueta **‘a’**.





Y, por último, descargamos el enlace con el mapa que nos interesa.



Referencias

**Básicos**

* Subirats, L., Calvo, M. (2019). **Web Scraping**. Editorial UOC.

**Complementarios**

* Lawson, R. (2015). **Web Scraping with Python**. Packt Publishing Ltd. Chapter 2

Web

* **Centro Europeo para la prevención y control de las enfermedades**. Recuperado de:

<https://data.europa.eu/euodp/es/data/publisher/ecdc>

* **Disasters**. Recuperado de:

<https://reliefweb.int/disasters>

* **Scraping en Python con ejemplos.** Recuperado de:

<https://jarroba.com/scraping-python-beautifulsoup-ejemplos/>

* **Monte Yasur en Vanuatu: El volcán en activo más accesible del mundo.** Recuperado de:

<https://tiempodexplorar.com/monte-yasur-en-vanuatu-el-volcan-en-activo-mas-accesible-del-mundo/>