

Práctica 1: Web Scraping - Dataset: Evolución COVID-19

Patricia García Suarez^{*}

Adrián Arnaiz-Rodríguez^{**}

10/4/2020

Índice

1. Enlaces de interés	2
2. Contexto	3
2.1. Link a Worldometers - COVID-19	4
3. Título del dataset	4
4. Descripción breve del dataset	4
5. Representación gráfica	4
6. Contenido	5
6.1. Dataset	5
6.2. Cómo se ha recogido	6
6.2.1. Travis	7
7. Agradecimientos	9
8. Inspiración	9
9. Licencia	9
10. Código fuente y dataset en Zenodo	9
11. Tabla de contribuciones	10

1

^{*}Perfil Github: <https://github.com/Kadashi>

^{**}Perfil Github: <https://github.com/AdrianArnaiz/>

¹Bibliografía al final del documento

1. Enlaces de interés

Repositorio de Github: https://github.com/AdrianArnaiz/scrap_uoc

DOI de Zenodo: [10.5281/zenodo.3748050](https://doi.org/10.5281/zenodo.3748050).

Link a Zenodo: <https://zenodo.org/record/3748050#.XpD5w8gzZ9A>

2. Contexto

En el contexto actual de la situación del virus **COVID-19** es imprescindible poder tener datos sobre la situación actual de los países para poder actuar en consecuencia. Si se disponen de buenos datos y un buen modelo, se puede incluso predecir datos futuros para poder prevenir o gestionar de manera más efectiva tanto recursos como medidas a tomar. Sin embargo, se produce una gran acumulación de datos de diversas fuentes. Existen muchas y muy diversas fuentes de los datos de personas contagiadas, casos activos, pacientes recuperados, muertes...

En este contexto de falta de homogeneidad y rigurosidad en los datos, nos adentramos a buscar fuentes de datos que pudieran ser candidatas a llevar un conteo diario (*serie temporal*) por país sobre: contagiados, casos activos, recuperados, muertes y tests realizados. El objetivo es realizar series temporales de estos datos por país con la función de realizar en un futuro análisis sobre los mismos. Nuestro contexto particular sería el de obtener estos datos tanto para **explicar comportamientos pasados, presentes como predecir futuros**. Incluso analizarlo minuciosamente comparando los datos de test realizados y casos totales o cualquier pareja de atributos, incluso analizando las series temporales de datos del COVID-19 con otros datos: PIB, exportaciones, valores bursátiles, densidades de población...

En **resumen**, digamos que nos queremos poner en la piel de instituciones como el CSIC o el INE y realizar estudio minucioso sobre el desarrollo de la enfermedad (tanto de la propia enfermedad, como su relación con datos externos). Por ello, para comenzar **necesitamos los datos centralizados y rigurosos de la evolución contagiados, casos activos, recuperados, muertes y tests realizados por país y por día**.

Hemos analizado muchas posibles fuentes de datos:

- *Oficiales (gobiernos)*: dificultad de recopilación de datos. Habría que buscar la vía por la que cada uno de los países realiza el comunicado y sacar los datos de ese comunicado (algunos comunicados son textos reales en pdf, se necesitaría una labor difícil de *NLP*.)
- *OMS*: En este caso, se debería pensar que es la fuente más fiable de los datos. Sin embargo, en el momento que empezamos a realizar esta práctica, no tenían una plataforma oficial donde se indicaran los datos por países. A esto se añade la poca fiabilidad que han tenido los datos de esta organización en las primeras fases de la epidemia.
- *European Centre for Disease Prevention and Control*: Permitía descargar un gran csv con datos. El motivo por el que no hemos elegido es porque sólo tiene datos de contagiados y muertes.
- *Universidad de Oxford - Our world in data*: Se trata de un estudio interactivo y actualizado en tiempo real que está realizando la Universidad de Oxford sobre la pandemia. Intentan analizar múltiples aspectos de la enfermedad, sin embargo obtienen los datos a través de varias organizaciones (los casos de la OMS, las muertes de la Johns Hopkins e incluso tienen acuerdos). Hemos visto que obtiene los datos a través de la universidad **Johns Hopkins**.
- **Johns Hopkins University**: Esta universidad ha tenido una visualización muy importante [Dong et al., 2020], la cual ha aparecido en todos los medios de comunicación. Sin embargo, los datos los tienen en un repositorio Github donde guardan los csv. También hemos visto que una de sus fuentes principales es **wolrdometers**.
- **Worldometers**: Nos hemos decantado por worldometers porque cumple con todas las funcionalidades que buscábamos: variedad en los datos, centralizados, rigurosos y actualizados dinámicamente. Deducimos que es **rigurosa** y de calidad por dos aspectos: el primero es que una universidad tan prestigiosa como la Johns Hopkins University la utiliza como fuente para su visualización. El segundo aspecto es por la completitud de datos (todos los atributos que queríamos, de muchos países y cada vez van añadiendo más). También, como vista de que es **dinámica** la propia página obtiene datos de diversas fuentes, pero si alguien da datos con justificantes de veracidad: los datos se actualizan (Ver Figura 1). Por ello, consideramos esta página como un repositorio central de los datos del COVID.

2.1. Link a Worldometers - COVID-19

- Link a la página que contiene la tabla: <https://www.worldometers.info/coronavirus>.

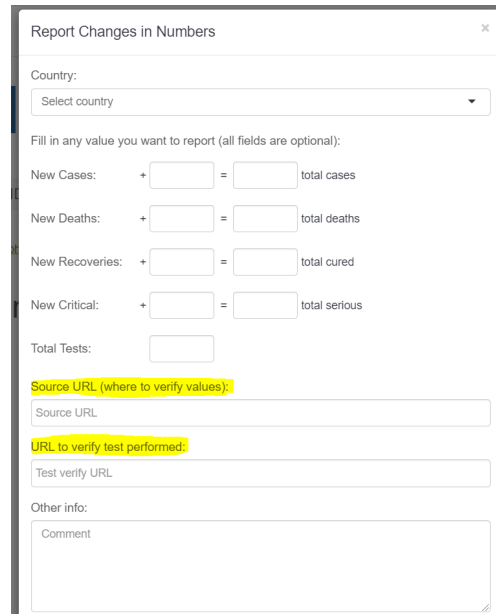


Figura 1: Como insertar nuevos datos con referencias de veracidad

3. Título del dataset

Evolución de contagio del COVID-19 por países.

4. Descripción breve del dataset

El dataset tendrá la evolución temporal de 5 datos relativos al covid por países. Guardaremos los datos relativos a **contagiados, casos activos, recuperados, muertes y tests realizados**. Es decir para cada uno de los países en los que haya casos registrados, guardaremos un dato al día (de manera automática) cada uno de los datos recién enumerados. Al final, reflejamos la serie temporal de cada uno de esos datos por países.

Como podemos ver es un dataset de 3 dimensiones (Tipo de Dato, País y fecha). En la sección 6.1, explicaremos cómo lo hemos resuelto.

5. Representación gráfica

El potencial actual de visualización que están sacando las asociaciones son gráficas de líneas para ver la evolución de los datos y gráficas de datos sobre el mapa para explicar evolución del contagio, mortalidad, etc. Entendemos que este apartado se refiere a representar el dataset de manera visual o esquemática, no a visualizaciones que se pueden obtener a partir del mismo.

Como hemos comentado tiene 3 dimensiones:

- País: Países con al menos un caso (así añade *worldometers países a su tabla*)
- Tiempo: Desde el 30/03 que empezamos a scrapear al menos 1 vez al día (la automática), hasta el día actual, ya que el scrap se lanza automáticamente con Travis todos los días a las 10:00 GMT+0
- Tipo de Dato: datos que queremos registrar: contagiados, casos activos, recuperados, muertes y tests realizados. Por tanto, lo representaremos gráficamente tal y como se ve en la Figura 2.

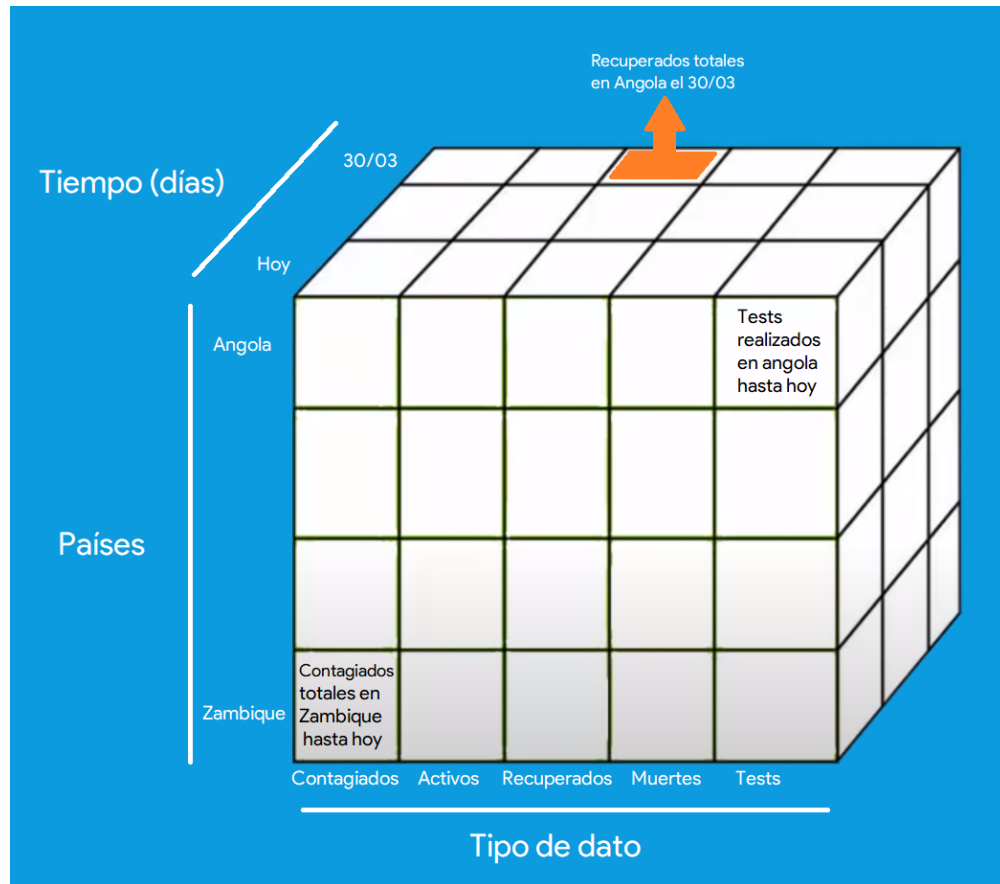


Figura 2: Representación gráfica

6. Contenido

6.1. Dataset

Nuestros datos tienen los campos que hemos contado anteriormente, sin embargo, tenemos que explicar la solución que hemos tomado para plasmarlo en formato csv. Hay dos alternativas (ver Figura 3):

- Un único csv con las siguientes columnas: *País, Fecha, Contagiados, activos, recuperados, muertes, tests*. Así tendríamos una entrada en la tabla por cada par País-Fecha. Tendría $num_filas = País \times Fecha$.
- **Varios csv:** Consiste en que una dimensión se utilice para dividir los datos en distintos csv. Es decir, un csv por cada *slice* del cubo. **Hemos utilizado esta alternativa** porque facilita la comprensión a la hora de ver los datos. Además, normalmente trabajaremos con un dato en concreto (p.e. analizar contagiados), por lo que si tenemos todo en un único csv, la mayoría de análisis empezarían con simular este slice que hemos explicado. Por lo tanto tenemos **5 csv**: la variación temporal de cada tipo de dato por país.

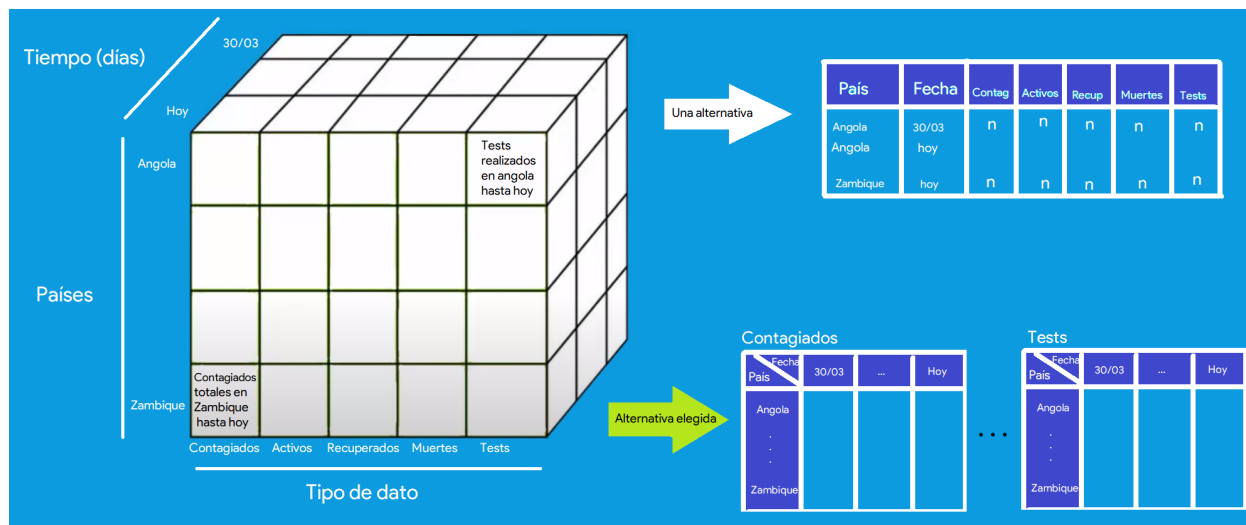


Figura 3: Representación gráfica

La estructura individual de cada uno de estos csv es la misma. Se encontrarán en el directorio `csv/covid19_series` y cada archivo tendrá el nombre `{TipoDato}_covid19_timeserie.csv`, siendo `TipoDato` cada dato que hemos nombrado: *TotalCases*, *TotalDeaths*, *TotalRecovered*, *ActiveCases*, *TotalTests*. Cada csv tendrá como filas los países y como columna la fecha de cuando hemos realizado un scrapping (idealmente una al día, pero durante la fase de desarrollo hay días que hemos hecho más → se refinará en posteriores prácticas en la fase de limpieza de datos).

6.2. Cómo se ha recogido

Hemos hecho scrapping sobre la página de Worldometers-Coronavirus, en el script alojado en el directorio `src\Scraping_covid19.py`. En el tenemos una tabla que muestra los valores de los datos (contagiados, activos, etc) por país en el momento actual. Es decir, las filas los países y las columnas los datos del momento actual. El sitio web no dispone ni de `robots.txt` ni de `sitemap.xml`. En cuanto a las características del scrapping (pocas llamadas a una sola página), que no haya `robots.txt` ni `sitemap.xml` tampoco nos plantea un gran contratiempo.

Hemos hecho scrapping sobre ese link, utilizando *BeautifulSoup* para encontrar la tabla y poder navegar sobre filas y columnas para recuperar los elementos. Hemos debido hacer la búsqueda en la tabla para quedarnos con los datos necesarios (p.e. a la hora de hacer el scrapping de la tabla, devuelve 8 filas el principio que no siguen el formato de la tabla, se deben corresponder con los resúmenes por continente, esas filas las hemos saltado en el scrapping).

Tras recuperar los datos de la tabla actual, está la **fase de actualización de datos** (o creación si es la primera vez que se lanza el script). Actualizamos las tablas con los datos nuevos, se añaden países si hay nuevos países en la tabla, y si hay países que antes estaban y ahora no queda su casilla a `Nan`. Todo ello con *Pandas*.

Puede surgir una pregunta: *¿Cómo se han obtenido datos de otras fechas?*. Nuestro enfoque ha sido **automatizar el lanzamiento del scrapping** para que se ejecute una vez al día y se vayan **actualizando automáticamente los csv de las series temporales de los datos por país**. Por ello, la primera fecha de la que tenemos datos es del 30/03, que fue el primer día que teníamos desarrollado el scrapping y lo lanzamos. La herramienta **Travis** ha sido utilizada para automatizar el lanzamiento del script y el *autodeploy* a *GitHub* (*Travis* permite que, en su plataforma, una vez al día y de forma planificada y automática se ejecute el scrapping, se actualizan las tablas de datos y se haga un commit automático para actualizar los datos en el github). Explicaremos como lo hemos hecho con *travis* en la sección 6.2.1.

Al código del scraping se le ha añadido una **salida tipo logfile** donde se reflejan los resultados de la ejecución de cada scraping. En nuestro caso es muy útil, ya que como se hace de manera automática nos deja ver cómo ha sido, la hora, si todo ha ido bien... Esto se ve reflejado en el fichero `src\log_covid.log`.

En cuanto a estrategias anti-bloqueo cabe destacar que hemos implementado pocas por la naturaleza de nuestro scraping. Tal y como hemos explicado: solo hacemos **una petición al día** para recuperar la tabla de datos actuales y trabajamos con esa página ya descargada. En el paso del *request* a la página web sí que hemos **implementado reintentos con espera** (espaciando entre peticiones http), teniendo en cuenta tanto fallos de conexión (y nos dará excepción que capturamos y reintentamos), como fallos de mensajes (reintentamos si conectamos bien con la página pero nos devuelve un código diferente al 200). No vemos sentido a otras medidas como cambiar el *user-agent*, ya que con una petición al día no nos creará dificultades.

6.2.1. Travis

Comentaremos un poco en detalle cómo hemos automatizado la ejecución del scraping para auto-actualizar los csv una vez al día mediante un bot, ya que creemos que es una parte con mucho potencial de éste scraping.

Travis es una herramienta de automatización de pruebas. Como muchos *frameworks* de automatización de pruebas, pueden utilizarse también para labores de *RPA* (automatizar procesos). *Travis* se conecta con Github, para ejecutar tests de pruebas que tengas codificados en el script que le indiques (debe estar en el repositorio). Se ejecuta automáticamente cada vez que haces un push o de manera planificada (diaria o mensualmente). Hemos utilizado esto para indicarle a **Travis que ejecute nuestro script de scraping una vez al día (no cada push) y haga un deploy de los resultados a Github**.

Para realizar ésto, solo necesitamos conectar nuestro repositorio con *Travis* y añadir la configuración en el archivo `.travis.yml` en la carpeta raíz de nuestro repositorio.

En el `Readme.md` de la raíz del proyecto, hay una marca que indica cómo ha ido el último scraping (*passing* o *faillure*), ver Figura 4. Si se clica sobre ella, se ven los resultados del ultimo scraping en la plataforma de *Travis* (aunque no se tenga usuario se puede ver los resultados).

🔗 Practica 1: Web Scrapping

Resultado del ultimo Scraping automático: build passing

Figura 4: Marca resultado scraping automatico Travis

Los pasos para su configuración son:

- Configurar el archivo `.travis.yml`.
- Registrarte en *Travis* y **activar** la conexión con tu repositorio. A partir de ahí, *Travis* ya sabe automáticamente que tiene que ejecutar por la configuración del anterior paso. Ver Figura 5.
- Crear un **token de conexión segura** para que se pueda realizar el deploy automáticamente (el push desde el bot de *Travis* para la actualización del *Github*). Ver Figura 6.
- Configurar lanzamiento automático una vez al día: Cron. Ver Figura 7.

Ahora, todos los días a las 12:11 hora de España, el bot de Travis nos hará un commit actualizando los datos. El resultado de la ejecución en su máquina se ve clicando la marca en el `readme.md` anteriormente descrita.

Legacy Services Integration

Filter repositories

DisVoice	<input type="checkbox"/>	Settings
models	<input type="checkbox"/>	Settings
scrap_uoc	<input checked="" type="checkbox"/>	Settings
TFG-Neurodegenerative-Disease-Detection	<input type="checkbox"/>	Settings

Figura 5: Activar conexión Travis-repositorio

GITHUB

Personal access tokens

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

scrap_uoc — repo	Never used	Delete
------------------	------------	--------

TRAVIS

Environment Variables

Customize your build using environment variables. For secure tips on generating private keys [read our documentation](#)

If your secret variable has special characters like `&`, escape them by adding `\` in front of each special character. For example, `ma&w!d0`

NAME	VALUE	BRANCH
GITHUB_TOKEN		master clear

Figura 6: Conexión con token seguro

Cron Jobs

BRANCH	INTERVAL	OPTIONS
master	Daily	Always run

↓

🕒 Ran 15 minutes ago	🕒 Scheduled in a day from now	2020-04-09T10:11:04Z
----------------------	-------------------------------	----------------------

Te muestra la siguiente fecha de ejecución. Se ejecuta todos los días a las 10:11 gmst+0 (12 en España)

Figura 7: Conexión con token seguro

7. Agradecimientos

- Principalmente, agradecer a la asociación *Worldometers* [Worldometers, 2020], asociación de estadísticas mundiales en tiempo real, por tener los datos actualizados de manera tan rápida y en abierto.
- Después, tanto a los estudios de la *Johns Hopkins University* [Dong et al., 2020], como a la asociación *Our world in Data* de la *Oxford University* [Roser et al., 2020], por sus trabajos que nos han permitido descubrir fuentes de calidad.
- Agradecer los recursos encontrados para realizar el scraping, tanto en [Lawson, 2015] como en el módulo [Subirats Mate and Calvo Gonzalez, (sf), propio de la UOC].

8. Inspiración

Esta pregunta ha sido respondida en anteriores apartados. Pero resumiendo, pretendemos analizar el cómo se ha desarrollado en el pasado la enfermedad, **explicar en qué etapa estamos en el presente y poder predecir los valores futuros para una mejor prevención**. A parte, en el futuro podrán analizarse otros datos externos para ver su correlación o su influencia en la propagación o viceversa (valores bursátiles, etc).

La inspiración ha sido el clima actual de la pandemia y repitiendo lo dicho en el primer apartado, buscar homogeneidad y rigurosidad en los datos para elaborar correctamente las anteriores tareas.

9. Licencia

Hemos elegido la licencia **CC BY-SA 4.0 License** por los siguiente motivos:

- *Deberán seguir con esta licencia las contribuciones a éste trabajo*: se asegura que se seguirá utilizando como mínimo una licencia tan restrictiva como esta.
- *Cuando se hagan cambios sobre el trabajo, habrá que indicar dichos cambios y al autor original*: se sigue reconociendo la autoría y qué cosas ha aportado cada uno.
- *Se permite comercializar*, con lo que podremos conseguir llegar a más cantidad de personas si alguien utiliza nuestro dataset con este fin.

10. Código fuente y dataset en Zenodo

- El código fuente del scraping, actualización de datos y autoimatización mediante *Travis* se encuentra en este enlace. Además hay archivos **readmemd** en los directorios que dan más información del proyecto.
- El dataset (conjunto de 5 csv) se sube a *Zenodo*, sin embargo, cabe **destacar que debido al potencial añadido de la autoactualización con *Travis*, este dataset está en continua actualización diaria de los datos**. El dataset con mayor actualización se corresponderá al que tenemos en el repositorio.
 - **DOI de Zenodo**: 10.5281/zenodo.3748050.
 - **Link a Zenodo**: <https://zenodo.org/record/3748050#.XpD5w8gzZ9A>

11. Tabla de contribuciones

Contribuciones	Firma
Investigación previa	P.G.S, A.A.R
Redacción respuestas	P.G.S, A.A.R
Desarrollo de código	P.G.S, A.A.R

Referencias

- Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases*, 2020.
- Richard Lawson. *Web scraping with Python*. Packt Publishing Ltd, 2015.
- Max Roser, Hannah Ritchie, and Esteban Ortiz-Ospina. Coronavirus disease (covid-19)–statistics and research. *Our World in Data*, 2020.
- Laia Subirats Mate and Mireia Calvo Gonzalez. Web scraping. Technical report, UOC, Barcelona, (sf). PID00256970.
- Worldometers. Covid-19 coronavirus pandemic. <https://www.worldometers.info/coronavirus/>, 2020.