TEMA 1

MÓDULO:
DATA MANAGEMENT & DATA DIGITAL

DATA DIGITAL

FERRAN CARRASCOSA

Licenciado en Matemáticas por la UB
Data Scientist

Institut de Formació Contínua-IL3
UNIVERSITAT DE BARCELONA

ÍNDICE

OBJETIVOS ESPECÍFICOS

- Conocer los distintos orígenes de los datos digitales.
- Realizar un diagnóstico sobre cómo están estructurados los datos digitales.
- Obtener datos digitales estructurados y no estructurados mediante herramientas analíticas.

1. DATOS DIGITALES

1.1 INTRODUCCIÓN

Antes de entrar en material sobre los datos digitales, hablemos de datos: los conceptos de datos, la información, los conocimientos y la sabiduría están interrelacionados.

SABÍAS QUE...

En la literatura inglesa, el término "data" aparece por primera vez en 1640.
La palabra "data" fue utilizada por primera vez en 1946 para referirse a "información informática transmisible y almacenable".

La expresión "procesamiento de datos" se utilizó por primera vez en 1954. data | Origin and meaning of data by Online Etymology Dictionary. www.etymonline.com.

A continuación, introduciremos algunos conceptos básicos en la cadena de obtención, interpretación, almacenaje y consulta de datos digitales que iremos desarrollando conforme vayamos avanzando en el temario.

Para obtener más información, apóyate en el siguiente material complementario: modulo3_tema1_dd_01_introduccion

1.1.1. CONCEPTOS BÁSICOS

Los datos digitales, en la Teoría de la información y los sistemas de información, son la representación discreta y discontinua de información. Se representan habitualmente en forma de números y letras.

ESTADOS

Los datos digitales se pueden encontrar en tres estados:

- **Datos en reposo:** son datos almacenados en formato digital de forma persistente (bases de datos, data warehouses, hojas de cálculo, archivos, dispositivos móviles, etc.).

  Están sujetos a ser accedidos, modificados o sustraídos. Para prevenir este tipo de accesos, las organizaciones utilizan medidas de protección como las contraseñas y la encriptación.

- **Datos en tránsito:** se definen como la información que fluye sobre una red, ya sea pública (por ejemplo, Internet) o privada (red interna de una compañía o corporación).

- **Datos en uso:** son datos almacenados en formato digital de forma no persistente (memoria RAM, caché CPU, una sesión de una página web, etc.).

La identificación del estado de los datos digitales y, como veremos a continuación, su formato de archivo, es fundamental para diagnosticar la mejor forma de acceder a ellos.

FORMATO DE ARCHIVO

El formato de archivo es la forma de organizar la información y codificarla dentro del archivo o sistema informático. Algunos ejemplos de formatos que se van a utilizar en este tema, son:

Texto plano

Fichero que contiene únicamente texto formado por letras, números y signos de puntuación (incluyendo espacios), también incluye caracteres de control, como tabuladores, saltos de línea o retornos de carro (Enter o Return). Estos caracteres se pueden codificar con distintos métodos.

Un método básico para codificar texto es el sistema ASCII. Si se necesita almacenar una mayor variedad de caracteres, por ejemplo, caracteres orientales, árabes, etc. se utiliza habitualmente el sistema UTF-8.

El texto plano utiliza muchas extensiones según su estructura interna. Cuando no se presupone una estructura interna prefijada, es muy común utilizar la extensión .txt.

RECUERDA

Los ficheros de tipo .csv, vistos en los temas de programación en R y Python, son ficheros de texto plano con una estructura de tabla y valores separados por un delimitador (por ejemplo: coma o punto y coma).

JSON

Otro ejemplo de fichero de texto plano es el formato JSON (*JavaScript Object Notation*). Este formato se construye como arrays (vectores) de pares clave-valor (como los diccionarios de Python). Este formato tiene la virtud que es un formato simple de codificar y es autodefinido. Está enfocado a codificar documentos y se utiliza en bases de datos documentales, por ejemplo, MongoDB.

XML

Un precedente a JSON, aunque de propósito más general, es XML, siglas en inglés de *eXtensible Markup Language*, traducido como "Lenguaje de Marcas Extensible".

También, está construido mediante texto plano, es autodefinido como JSON y está desarrollado por el *World Wide Web Consortium* (W3C).

XML, tiene muchas variantes:

- XSLT: permite generar contenido web dinámico en HTML o XHTML.
- XPATH: permite buscar y seleccionar texto dentro de un documento XML.

Profundizaremos en estas cuestiones y algunas más en el apartado de web scraping.

HTML

HTML: *HyperText Markup Language*.

Utilizado en las páginas web. Es un formato de texto plano inventado al mismo tiempo que la web por Tim Berners Lee en 1989.

Viene estructurado con un conjunto de marcas, aunque, habitualmente, estas reglas se incumplen y los navegadores tienen que interpretar el código y corregir las ambigüedades presentes.

Veamos algunos formatos complementarios a HTML que, a su vez, son variantes de XML:

- XHTML: es un HTML con estructura XML válida, es decir, sin errores de construcción.
- CSS: son hojas de estilo para páginas web.
- RSS: o *Really Simple Syndication*, utilizado para distribuir contenido en la web y para difundir información actualizada de forma frecuente a los usuarios suscritos.

  RSS es, por tanto, un ejemplo de datos públicos en tránsito. Aunque se ha utilizado desde el inicio de la web, su gran impulso vino de la mano de la red social o Web 2.0 a través de los Blogs. Es, por lo tanto, un precedente de otras formas de difusión social como Twitter y Facebook.

MÉTODOS DE ACCESO

Los métodos de acceso que se explican en el tema, cubren un abanico muy amplio de orígenes de datos, desde la habitual descarga a través de una web, hasta el uso de Python para realizar estas tareas de forma automática.

Peticiones http

Hypertext Transfer Protocol, abreviado HTTP, es el protocolo en el que se basa Internet. Los mensajes HTTP son de texto plano, tanto la petición como la respuesta.

El ejemplo más común de peticiones HTTP son las que realiza un navegador a un servidor web, para obtener el código HTML.

API

Una API, acrónimo de *application programming interface*, permite definir las llamadas o peticiones que se pueden hacer a un programa.

Si bien es un contexto muy general, se utilizará en este tema como una forma de solicitar datos a través de la web de forma estructurada, es decir, poder seleccionar los datos que queremos obtener a través de unos parámetros de entrada y una salida, generalmente, de texto (JSON, XML, etc.).

Si bien existen varias arquitecturas API, en el temario nos centramos en el tipo REST API, *representational state transfer*. Esta arquitectura, también llamada *RESTful Web services*, permite transferir datos a través de HTTP, es decir Internet, directamente en texto plano, por lo tanto, permite utilizar JSON y XML.

Los siguientes sitios web utilizan APIs de tipo REST en formato de texto de tipo JSON:

- SWAPI The Star Wars API: ¿te suena?
- Twitter API
- Facebook Social Graph API
- Flickr
- YouTube
- OpenStreetMap
- Google Maps
- Wikipedia API
- Pokemon Go

Parseado del texto

Para procesar datos digitales, no sólo es necesario acceder a los datos, además, hay que saber seleccionar, de entre todo el texto capturado, aquellos elementos que queremos analizar. Las técnicas para realizar esta selección se llaman **parseado** (o analizador sintáctico) del texto. El tema muestra distintas técnicas, desde la selección directa mediante XPATH (ver sección XML), el uso de herramientas que facilitan la explotación del HTML como BeatifulSoup, hasta el uso de expresiones regulares que permiten seleccionar texto mediante patrones de búsqueda.

Javascript y Selenium

Se trata del lenguaje de scripting más común que utilizan los navegadores para construir el contenido de una página web. Éste se ejecuta, directamente, en el navegador, lo que dificulta poder acceder, directamente, desde Python, al contenido generado de la web. Para superar este tipo de barreras, Python se apoya en un software externo, SELENIUM, que permite controlar el navegador desde Python y simular un acceso completo al contenido de la web.

ALMACENAJE DE LOS DATOS

Para guardar la información de forma persistente, se utilizan:

- Ficheros de texto (por ejemplo: .txt, .csv, .json, etc.): es un sistema muy flexible, pero poco eficiente.
- Bases de datos estructuradas SQL: son estructuras de datos muy rígidas, aunque muy eficientes.
- Bases de datos documentales (por ejemplo: MongoDB): es un mix de flexibilidad en la estructura de los datos y eficiencia en su explotación, que lo hace el sistema idóneo para este tipo de tareas.

AUTOMATIZACIÓN

El último paso consiste en automatizar, mediante lo que se llama un bot, la extracción de los datos. Para automatizar, se utilizan tiempos de espera planificados, calendarizadores que ejecutan scripts de Python según el día y la hora.

Una buena automatización, además, permite minimizar los errores por denegación de acceso producidos por un excesivo volumen de consultas a un mismo sitio. También, permite lanzar avisos frente a cambios relevantes en la estructura de la web.

1.2. FUENTES DE DATOS EXTERNAS OFICIALES

Las fuentes de datos oficiales son un gran recurso para enriquecer la calidad de los datos utilizados dentro de nuestro análisis.

Supone una fuente de datos fiable y exhaustiva, tanto en temática como en ámbito geográfico y, habitualmente, se puede actualizar de forma recurrente.

Este tema se centra en la exploración de 3 fuentes de datos en 3 ámbitos:

- **Nacional**: Instituto Nacional de Estadística (INE).
- **Europeo**: Oficina estadística de la Comisión Europea (EUROSTAT).
- **Global**: Datos abiertos del Banco Mundial (WORLD DATA BANK).

PIENSA UN MINUTO

La Clasificación Nacional de Actividades Económicas (CNAE) es utilizada por las oficinas estadísticas y está uniformizada por toda la Unión Europea. Permite identificar 630 actividades económicas distintas agrupadas, de forma jerárquica, en 2, 3 y 4 dígitos. Las "Actividades cinematográficas, de vídeo y de programas de televisión" tienen el epígrafe **"591"**. Esta información la podrás encontrar en el INE.

1.2.1. INSTITUTO NACIONAL DE ESTADÍSTICA (INE)

El INE es un organismo del estado Español con un papel destacado en la actividad estadística pública. Realiza los censos demográficos y económicos, las cuentas nacionales, las estadísticas demográficas y sociales, los indicadores económicos y sociales, la coordinación y mantenimiento de los directorios de empresas, la formación del Censo Electoral, etc.

Además, regula las relaciones con las oficinas de estadística territoriales y con la Oficina Estadística de la Unión Europea (EUROSTAT).

El INE ha creado el espacio **"Datos abiertos"** accesible tanto a través de la web del INE: www.ine.es/datosabiertos, como a través del portal datos.gob.es.

Los conjuntos de datos y aplicaciones más relevantes son:

- **Información estadística elaborada por el INE y publicada en INEbase:** permite acceder a los resultados agregados de las estadísticas por temas.
- **Microdatos anonimizados de encuestas:** permite acceder a los datos obtenidos de algunas encuestas a nivel de registros (microdatos) de forma anónima. En general, consiste en ficheros de texto, con columnas en formato de ancho fijo. Su lectura requiere el uso de un diccionario que especifica el significado de cada variable.
- **Callejero de censo electoral**: contiene el catálogo de calles, números de calle y códigos postales en España.
- **API JSON:** especifica una API de consulta de los datos de INEbase y Tempus3.

INEBASE

Los datos de INEbase se pueden consumir, directamente, desde la web.

El procedimiento es simple, se navega por el árbol temático para seleccionar aquellos datos que puedan ser de nuestro interés.

Para nuestra actividad, vamos a centrarnos en los apartados de "Demografía y población" y "Economía".

Demografía y población

Tenemos 2 apartados de nuestro interés:

- **Padrón. Población por municipios:**
  - Estadística del Padrón continuo (01/01/2020).
- **Cifras de población y Censos demográficos:**
  - Cifras de población (datos provisionales 01/01/2020).

El **Padrón Municipal** es un **registro** donde constan los vecinos del municipio. Su gestión está al cargo de los respectivos ayuntamientos. Se publican cada 1 de Enero e informan los totales de población hasta nivel inframunicipal de sección censal.

SABÍAS QUE...

La sección censal es la unidad mínima de información estadística. Se utiliza parar organizar los procesos electorales y agrupa entre 1.500 y 2.000 habitantes.

A su vez, las **Cifras de Población** son una operación estadística que utiliza distintas encuestas para elaborar las cifras oficiales de población. Se publican los datos de forma semestral a nivel de provincia.

Para nuestra actividad, nos centramos en la siguiente ruta:

Estadística del Padrón continuo > Resultados > Comunidades autónomas y provincias > 2.3 Población por edad (grupos quinquenales) y sexo

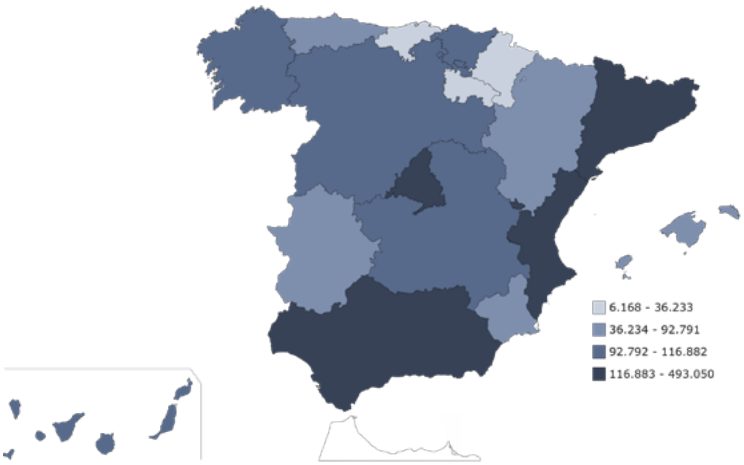
Fuente: INE

Seleccionaremos:

- "Ambos sexos".
- Todas las Comunidades.
- Edades.

Fuente: INE

Posteriormente, el formulario permite descargar los datos en distintos formatos (csv, Excel, etc.), así como generar gráficos y mapas:



Fuente: INE

En el siguiente punto, mostraremos cómo descargar los datos, directamente, desde Python, mediante la API JSON del INE. Con este objetivo, se obtiene la url de los datos:

- Botón de flecha de Descarga.
- Json.
- Se abre una ventana y copiamos la url: https://servicios.ine.es/wstempus/js/es/DATOS_TABLA//t20/e245/p04/provi/l0/0ccaa003.px?tip=AMtv=sexo:ambossexos

API JSON

La API JSON del INE permite acceder, mediante peticiones URL, a toda la información disponible en INEbase. Las consultas a la API, según la fuente, pueden ser de dos tipos:

- **Base de datos de difusión (Tempus3)**: datos principales publicados de forma periódica.
- **Repositorio de ficheros PC-Axis**: resto de datos.

Para comprender el funcionamiento de cada una de ellas, veamos dos ejemplos:

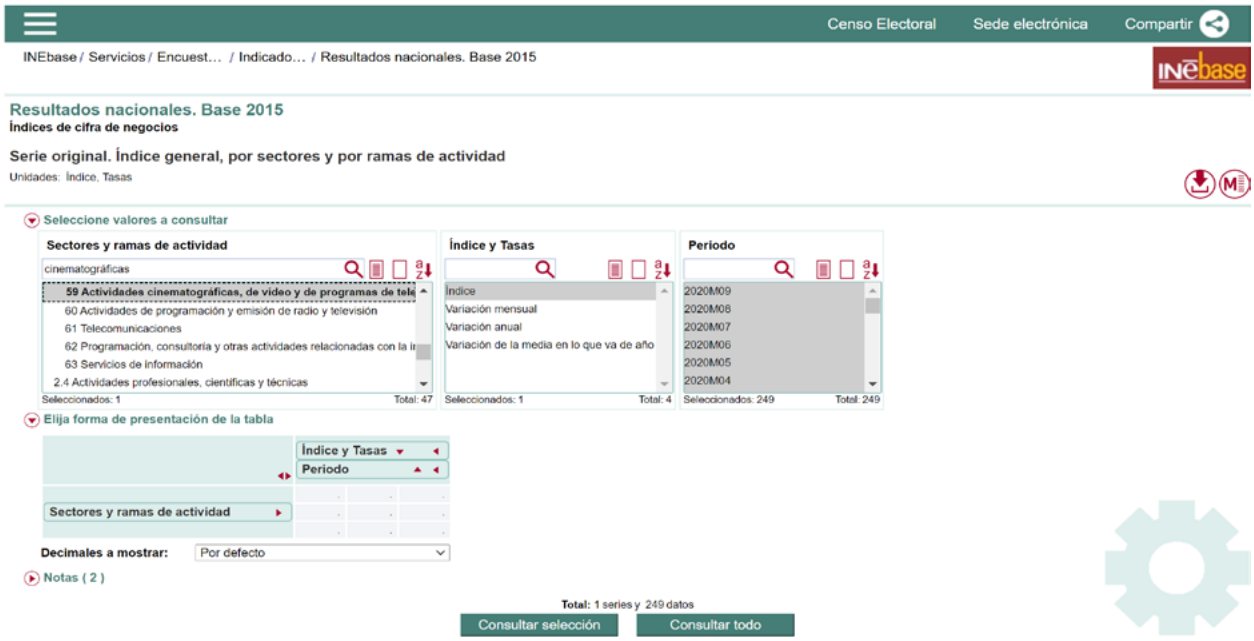
API - Tempus3

Descargamos la "Serie original. Índice general, por sectores y por ramas de actividad" disponible en la URL: https://www.ine.es/jaxiT3/Tabla.htm?t=25891.

Ahora, hacemos lo siguiente:

- Buscamos "cinematográficas" en la caja "Sectores y ramas de actividad" y pulsamos el botón Enter.

- Seleccionamos “59 Actividades cinematográficas, de vídeo y de programas de televisión, grabación de sonido y edición musical”.
- Seleccionamos “Índice”.
- Seleccionamos todos los periodos.



Índice Actividades cinematográficas. Fuente: INE

- Consultamos la selección.
- Hacemos clic en el botón Descargar (flecha superior derecha).
- Seleccionamos Json.

Descarga Índice Actividades cinematográficas. Fuente: INE

- A continuación, se abrirá una ventana con la siguiente url: https://servicios.ine.es/wstempus/js/es/DATOS_TABLA/25891?tip=AM&tv=387:17588&tv=3:83
- Para poder generar la petición desde Python, en primer lugar, se cargan las librerías y se declara la variable con la url:

```
from urllib.request import urlretrieve
from urllib.request import urlopen
import json
import pandas as pd
import numpy as np
```

```
urlTempus3 =
"https://servicios.ine.es/wstempus/js/es/DATOS_TABLA/25891?tip=AM&tv=387:17588&tv=3:83"
```

- Ahora, descargamos el contenido de la URL y lo pasamos a formato json:

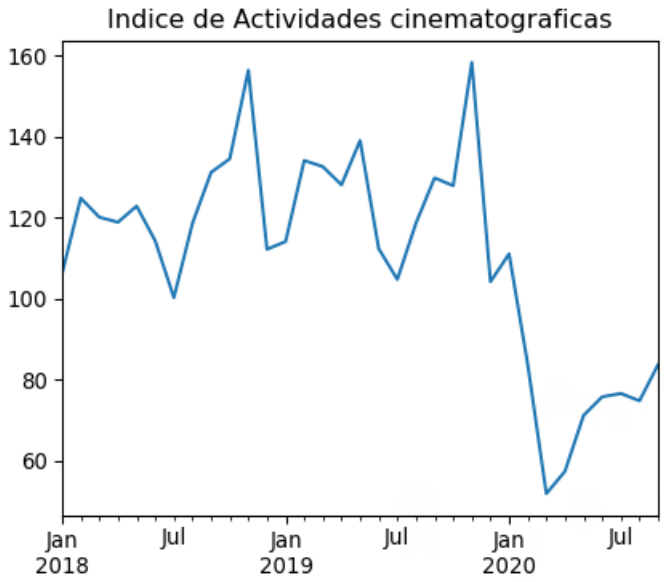
```
result = json.load(urlopen(urlTempus3))
```

- Transformamos de json a pandas.
- Ordenamos por el campo fecha.
- Generamos el índice de fecha (con frecuencia mensual).
- Eliminamos los valores faltantes.
- Seleccionamos la serie a partir de enero de 2018 en adelante:

```
df=pd.DataFrame(result[0]["Data"]).sort_values(by=["Fecha"])
#df.index = pd.to_datetime(np.array(df["Fecha"], dtype=np.datetime64)).to_period('D')
df.index = pd.to_datetime(np.array(df["Fecha"], dtype=np.datetime64)).to_period('M')
df = df[df['Valor'].notna()]
df = df['2018-01':]
```

- Se genera el plot para visualizar la serie:

```
df.Valor.plot.line(title='Indice de Actividades cinematograficas')
```



Índice de Actividades cinematográficas (fuente: elaboración propia con datos de INE)

API - PCAXIS

Los datos que se van a descargar, a continuación, son los datos vistos en la sección de Demografía y población, para ello:

- Recuperamos la url obtenida anteriormente.
- Se descargan los datos con urllib:

```
urlPCAxis =
"https://servicios.ine.es/wstempus/js/es/DATOS_TABLA//t20/e245/p04/provi/l0/0ccaa003.px?tip
tv=sexo:ambossexos"

result = json.load(urlopen(urlPCAxis))
```

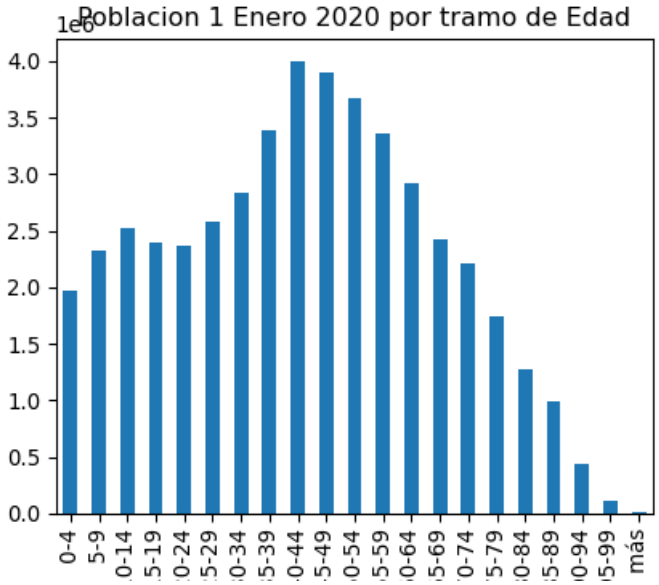
Después de analizar el contenido .json, se ha transformado a pandas filtrando que los datos correspondan con “Ambos sexos” y el total Nacional.

```
df = pd.DataFrame([x["Nombre"].split(",")+ [x["Data"][0]["Valor"]] for x in result if "Ambos
sexos" in x["Nombre"] and "Total Nacional" in x["Nombre"]])
df.shape

## (22, 4)
```

Posteriormente, se etiquetan los datos, se selecciona la edad y se realiza un plot para visualizar los datos.

```
df.columns = ["Genero","Region","Edad","Valor"]
df.index = df["Edad"]
df[df.index!=" Total"].Valor.plot.bar(title="Poblacion 1 Enero 2020 por tramo de Edad")
```



Población 1 Enero 2020 por tramo de edad (fuente: elaboración propia con datos de INE)

LIBRERÍA WORLD_BANK_DATA

Existen varias alternativas para explotar WBD desde Python. A continuación, se muestra el uso de world_bank_data.

Para obtener el listado de temas:

```
import pandas as pd
import world_bank_data as wb

wb.get_topics()["value"]

## id
## 1     Agriculture & Rural Development
## 2                   Aid Effectiveness
## 3                    Economy & Growth
```

```
## 4                          Education
## 5                     Energy & Mining
## 6                        Environment
## 7                   Financial Sector
## 8                             Health
## 9                     Infrastructure
## 10         Social Protection & Labor
## 11                           Poverty
## 12                    Private Sector
## 13                     Public Sector
## 14               Science & Technology
## 15                 Social Development
## 16                  Urban Development
## 17                             Gender
## 18        Millenium development goals
## 19                     Climate Change
## 20                      External Debt
## 21                              Trade
## Name: value, dtype: object
```

Si nos centramos en el tema 14-"Science & Technology", vemos los sub-temas disponibles:

```
wb.get_indicators(topic=14)["name"]
```

```
## id
## BM.GSR.ROYL.CD       Charges for the use of intellectual property, ...
## BX.GSR.ROYL.CD       Charges for the use of intellectual property, ...
## GB.XPD.RSDV.GD.ZS      Research and development expenditure (% of GDP)
## IP.JRN.ARTC.SC               Scientific and technical journal articles
## IP.PAT.NRES                          Patent applications, nonresidents
## IP.PAT.RESD                             Patent applications, residents
## IP.TMK.NRES                  Trademark applications, direct nonresident
## IP.TMK.RESD                     Trademark applications, direct resident
## IP.TMK.TOTL                               Trademark applications, total
## SP.POP.SCIE.RD.P6              Researchers in R&D (per million people)
## SP.POP.TECH.RD.P6              Technicians in R&D (per million people)
## TX.VAL.TECH.CD                   High-technology exports (current US$)
## TX.VAL.TECH.MF.ZS    High-technology exports (% of manufactured exp...
## Name: name, dtype: object
```

A continuación, para seleccionar un tema de interés, se seleccionan los datos del % de gasto en I+D sobre el PIB (id = GB.XPD.RSDV.GD.ZS) en España:

```
ImasD = wb.get_series('GB.XPD.RSDV.GD.ZS', country="ES")
ImasD.index = [x[2] for x in ImasD.index.values]
ImasD[ImasD.notna()]
```

```
## 1996    0.78949
## 1997    0.77964
## 1998    0.85102
## 1999    0.84052
## 2000    0.88495
## 2001    0.89019
## 2002    0.96005
## 2003    1.02219
## 2004    1.03849
## 2005    1.09577
## 2006    1.17217
## 2007    1.23448
## 2008    1.31706
## 2009    1.35134
## 2010    1.34961
## 2011    1.32508
## 2012    1.28788
## 2013    1.26859
## 2014    1.23535
```

```
## 2015    1.21832
## 2016    1.18526
## 2017    1.20580
## 2018    1.23700
## Name: GB.XPD.RSDV.GD.ZS, dtype: float64
```

La serie va de 1996 hasta 2018. Gráficamente:

```
ImasD.plot.line();
```

Se observan como posterior a la crisis de 2010, las inversiones en I+D bajan hasta que se empieza a recuperar en 2016 hasta el último año observado, 2018.

1.2.2. OFICINA ESTADÍSTICA DE LA COMISIÓN EUROPEA (EUROSTAT)

La Oficina Estadística de la Unión Europea (EUROSTAT) tiene como misión proveer estadísticas y datos de alta calidad sobre Europa.

Para cumplir con este objetivo, participa con el rol de coordinador dentro del European Statistical System (ESS), conjuntamente con el resto de oficinas estadísticas oficiales de los estados del Área Económica Europea y Suiza. Además, se coordina con otros organismos superiores como el World Data Bank.

El EUROSTAT trabaja para que exista un alto grado de armonización de los distintos estudios estadísticos realizados por los estados de Europa. Gracias a esta armonización existe un gran número de indicadores socioeconómicos y demográficos comparables entre los estados.

BASES DE DATOS

Los datos de EUROSTAT, se pueden ver, de forma temática, en Estadísticas por temas: estadística regionales, economía y finanzas...

Tablas resumen

Estas estadísticas, al igual que en el INE, se pueden consultar, desde la web, en formato de tablas agregadas y, posteriormente descargar, los datos en distintos formatos.

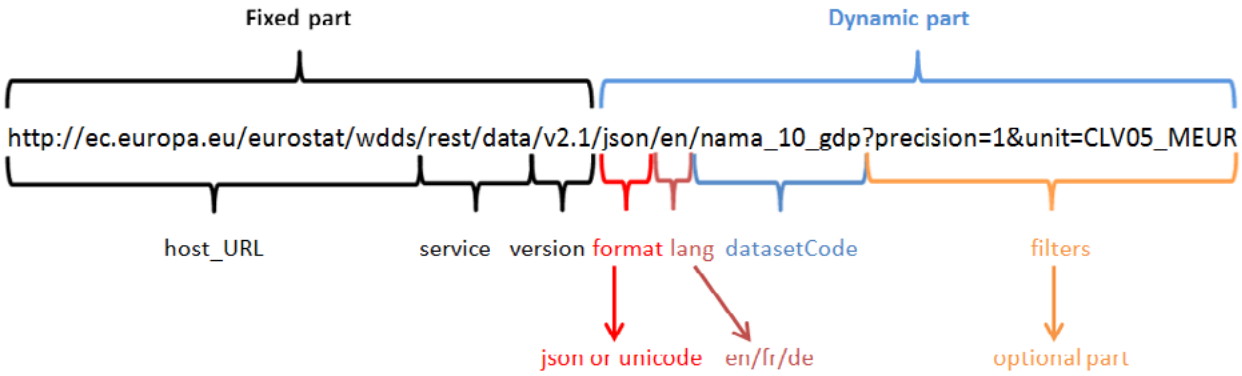
Ficheros de microdatos

Otro apartado relevante lo conforman los ficheros de microdatos. Estos datos permiten acceder a datos individuales sobre las respuestas a las encuestas estadísticas (raw data).

Algunos ejemplos de interés son: encuesta de ingresos y condiciones de vida o encuesta de la fuerza laboral.

API REST

Finalmente, EUROSTAT también ofrece una API REST. Esta API se puede llamar bajo la siguiente estructura:



La petición REST. Fuente: EUROSTAT

Dónde la llamada:

{host_url}/rest/data/{version}/{format}/{lang}/{datasetCode}?{filters}

Con parámetros:

- **host_url, service, version:** parte fija de la llamada.
- **format:** formato de retorno de los datos (json o unicode).
- **lang:** idioma de los metadatos (en/fr/de).
- **datasetCode:** código único identificado de la consulta de datos.
- **filters:** ambito de la consulta. Cada consulta tiene un límite de 50 sub-indicadores por consulta.
- **precision:** número de decimales.
- **unit:** filtro de la unidad solicitada.

Veamos un ejemplo de consulta del conjunto de datos "nama_10_gdp" para:

- **Periodo:** 2010 y 2011.
- **Geo:** European Union (28 países).
- **Precisión:** 1 decimal.
- **Indicador de Contabilidad nacional:** B1GQ – Gross domestic product at market prices.
- **Unidad:** CP_MEUR - Precios corriente, Millon de Euros.

```
urlEUROSTAT =
"http://ec.europa.eu/eurostat/wdds/rest/data/v2.1/json/en/nama_10_gdp?geo=EU28&precision=1&na_
item=B1GQ&unit=CP_MEUR&time=2010&time=2011"

result = json.load(urlopen(urlEUROSTAT))

print(result["dimension"])
```

```
## {'unit': {'label': 'unit', 'category': {'index': {'CP_MEUR': 0}, 'label': {'CP_MEUR':
'Current prices, million euro'}}}, 'na_item': {'label': 'na_item', 'category': {'index':
{'B1GQ': 0}, 'label': {'B1GQ': 'Gross domestic product at market prices'}}}, 'geo': {'label':
'geo', 'category': {'index': {'EU28': 0}, 'label': {'EU28': 'European Union - 28 countries
(2013-2020)'}}}, 'time': {'label': 'time', 'category': {'index': {'2010': 0, '2011': 1},
'label': {'2010': '2010', '2011': '2011'}}}}
```

Como en el caso del INE, resulta difícil configurar una llamada.

Para facilitar esta tarea, existe un generador de consultas donde el dato de entrada requerido es el nombre del dataset, que se puede buscar con el buscador de Estadísticas por temas.

La librería de Python Eurostat

Por último, para facilitar la explotación de datos de Eurostat, resulta extremadamente útil la librería eurostat de python.

Con ella, se pueden obtener fácilmente las temáticas y sus códigos de tablas:

```
import eurostat

toc_df = eurostat.get_toc_df()

toc_df.head()
```

```
##                                                title  ... data end
## 0                                   Database by themes  ...
## 1                     General and regional statistics  ...
## 2  European and national indicators for short-ter...  ...
## 3  Business and consumer surveys (source: DG ECFIN)  ...
## 4                 Consumer surveys (source: DG ECFIN)  ...
##
## [5 rows x 7 columns]
```

Para nuestra actividad, filtraremos los temas que contengan la palabra “cinema”:

```
toc_df[(toc_df.type=="dataset")&(toc_df.title.str.contains("cinema"))][['code']]
```

```
##             code
## 2011  hlth_ds010
```

Buscamos en Google el código: *hlth_ds010*.

El resultado es la página de Eurostat: [Frequency of going to cinema, live performances, cultural sites or attending live sport events by level of activity limitation, sex and age] (http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=hlth_ds010&lang=en)

Una vez analizados los datos, filtramos la estadística del % de población por país que cumple:

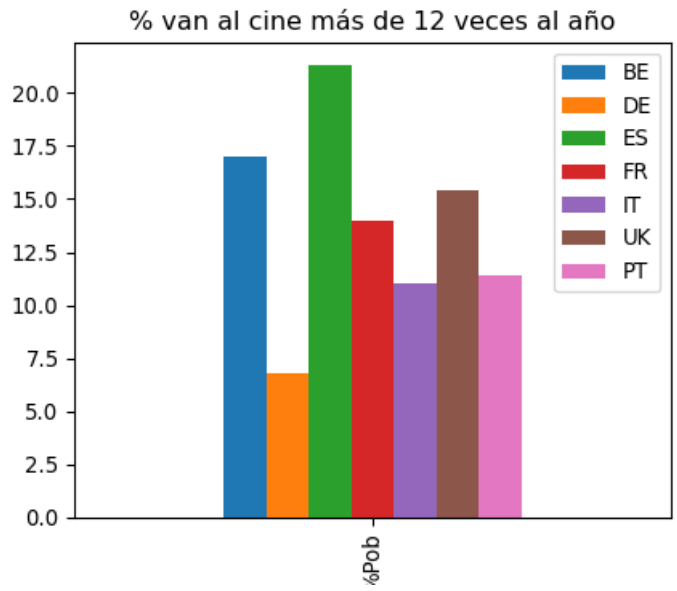
- Personas de edad entre 16 y 29 años.
- Van al cine (AC521) doce o más veces.
- Países: Bélgica, Alemania, España, Francia, Italia, Reino Unido o Portugal.

```
df = eurostat.get_data_df("hlth_ds010", flags=False)

paises = ['BE', 'DE','ES', 'FR','IT','UK', 'PT']
```

```
cinema = df[(df.age=="Y16-29") & (df.frequenc=="GT12") & (df.sex=="T") &
(df.lev_limit=="NONE") & (df.acl00=="AC521")][paises]
cinema.index=["%Pob"]

cinema.plot.bar(title='% van al cine más de 12 veces al año')
```



% van al cine más de 12 veces al año (fuente: elaboración propia con datos de EUROSTAT)

El resultado, no obstante el año del estudio es 2006, nos muestra que España era un país perfecto para la industria del Cine, ya que tenía la mayor proporción de población de este colectivo.

1.2.3. DATOS ABIERTOS DEL BANCO MUNDIAL (WORLD BANK DATA)

El World Bank Data, en los últimos años, de la mano del auge del Open Data, se ha convertido en una herramienta muy potente para comparar las estadísticas de los países y analizar su evolución.

ESTRUCTURA WBD

La estructura es muy similar al resto de oficinas estadísticas oficiales:

- Un navegador temático: permite visualizar datos agregados y descargar datos en distintos formatos.
- Acceso a microdatos: para acceder a datos individualizados de encuesta.
- Acceso por API para desarrolladores.

LIBRERÍA WORLD_BANK_DATA

Existen varias alternativas para explotar WBD desde Python. A continuación, mostramos el uso de world_bank_data.

Para obtener el listado de temas:

```
import pandas as pd
import world_bank_data as wb

wb.get_topics()["value"]
```

```
## id
## 1     Agriculture & Rural Development
## 2                   Aid Effectiveness
## 3                    Economy & Growth
## 4                           Education
## 5                    Energy & Mining
## 6                        Environment
## 7                   Financial Sector
## 8                             Health
## 9                     Infrastructure
## 10          Social Protection & Labor
## 11                            Poverty
## 12                    Private Sector
## 13                     Public Sector
## 14                Science & Technology
## 15                  Social Development
## 16                   Urban Development
## 17                             Gender
## 18        Millenium development goals
## 19                      Climate Change
## 20                       External Debt
## 21                               Trade
## Name: value, dtype: object
```

Si nos centramos en el tema 14-"Science & Technology", vemos los sub-temas disponibles:

```
wb.get_indicators(topic=14)["name"]
```

```
## id
## BM.GSR.ROYL.CD       Charges for the use of intellectual property, ...
## BX.GSR.ROYL.CD       Charges for the use of intellectual property, ...
## GB.XPD.RSDV.GD.ZS      Research and development expenditure (% of GDP)
## IP.JRN.ARTC.SC               Scientific and technical journal articles
## IP.PAT.NRES                          Patent applications, nonresidents
## IP.PAT.RESD                             Patent applications, residents
## IP.TMK.NRES                  Trademark applications, direct nonresident
## IP.TMK.RESD                     Trademark applications, direct resident
## IP.TMK.TOTL                               Trademark applications, total
## SP.POP.SCIE.RD.P6              Researchers in R&D (per million people)
## SP.POP.TECH.RD.P6              Technicians in R&D (per million people)
## TX.VAL.TECH.CD                   High-technology exports (current US$)
## TX.VAL.TECH.MF.ZS    High-technology exports (% of manufactured exp...
## Name: name, dtype: object
```

A continuación, para seleccionar un tema de interés, seleccionamos los datos del % de gasto en I+D sobre el PIB (id = GB.XPD.RSDV.GD.ZS) en España:

```
ImasD = wb.get_series('GB.XPD.RSDV.GD.ZS', country="ES")
ImasD.index = [x[2] for x in ImasD.index.values]
ImasD[ImasD.notna()]
```

```
## 1996    0.78949
## 1997    0.77964
## 1998    0.85102
## 1999    0.84052
```

```
## 2000    0.88495
## 2001    0.89019
## 2002    0.96005
## 2003    1.02219
## 2004    1.03849
## 2005    1.09577
## 2006    1.17217
## 2007    1.23448
## 2008    1.31706
## 2009    1.35134
## 2010    1.34961
## 2011    1.32508
## 2012    1.28788
## 2013    1.26859
## 2014    1.23535
## 2015    1.21832
## 2016    1.18526
## 2017    1.20580
## 2018    1.23700
## Name: GB.XPD.RSDV.GD.ZS, dtype: float64
```

La serie va de 1996 hasta 2018. Gráficamente:

```
ImasD.plot.line();
```

Observemos cómo, posterior a la crisis de 2010, las inversiones en I+D bajan hasta que se empieza a recuperar en 2016 hasta el último año observado, 2018.

Para obtener más información, apóyate en el siguiente material complementario: modulo3_tema1_dd_02_oficiales

1.3. GOOGLE ANALYTICS

Para acceder al módulo de Google Analytics, deberás ir al material complementario y abrir el siguiente notebook:

modulo3_tema1_dd_03_ganalytics

1.4. SOCIAL ANALYTICS

Para acceder al módulo de Social Analytics (Twitter, Facebook, Hootsuite), deberás ir al material complementario y abrir el siguiente notebook:

modulo3_tema1_dd_04_sanalytics

1.5. WEB SCRAPING

Para acceder al módulo de Web Scraping, deberás ir al material complementario y abrir el siguiente notebook:

modulo3_tema1_dd_05_webscraping

IDEAS CLAVE

- Los procesos de digitalización generan una fuente de datos continua y con un gran potencial de generación de riqueza.
- Algunos ejemplos relevantes son los datos oficiales, los datos de analítica web, los datos de redes sociales o los datos disponibles en la web.
- Python permite generar procesos de extracción para obtener y estructurar estos datos.
- Previo a extraer los datos, es necesario realizar un buen diagnóstico de los datos a obtener.
- Una vez se ha diagnosticado, hay que escoger las mejoras estrategias para explotar esta información: API o scraping de contenido estático o dinámico.

ANEXO: README

PREPARACIÓN DEL ENTORNO COLAB

Desde Colab, hay que clonar el repositorio y preparar el entorno, cada vez que inicias un nuevo libro. En los libros se incluye el código necesario:

```
if 'google.colab' in str(get_ipython()):
    !git clone https://github.com/griu/mbdds_fc20.git /content/mbdds_fc20
    !git -C /content/mbdds_fc20 pull
    %cd /content/mbdds_fc20/Python
    !python -m pip install -r requirementsColab.txt
    %cd /content/mbdds_fc20/datos_digitales
```

PREPARACIÓN ENTORNO LOCAL-JUPYTER

Para las prácticas de Selenium y MongoDB, es necesario preparar el entorno local de Jupyter:

Clonar repositorio

En local, puedes utilizar el mismo proyecto que has clonado en el **README DE R**.

Para actualizarlo de nuevo, desde consola:

```
cd mbdds_fc20
git pull
cd Python
```

Environment de Anaconda

En local, puedes utilizar el mismo environment que has preparado en el **README DE PYTHON**, no obstante, es necesario que lo actualices:

```
# Activar entorno
conda activate mbdds_rpy20
# Actualizar paquetes de de la carepta Python
python -m pip install -r requirementsColab.txt
# publicar el kernel
python -m ipykernel install --user --name mbdds_rpy20 --display-name "mbdds_rpy20"
# Abrir los notebooks
jupyter notebook
```