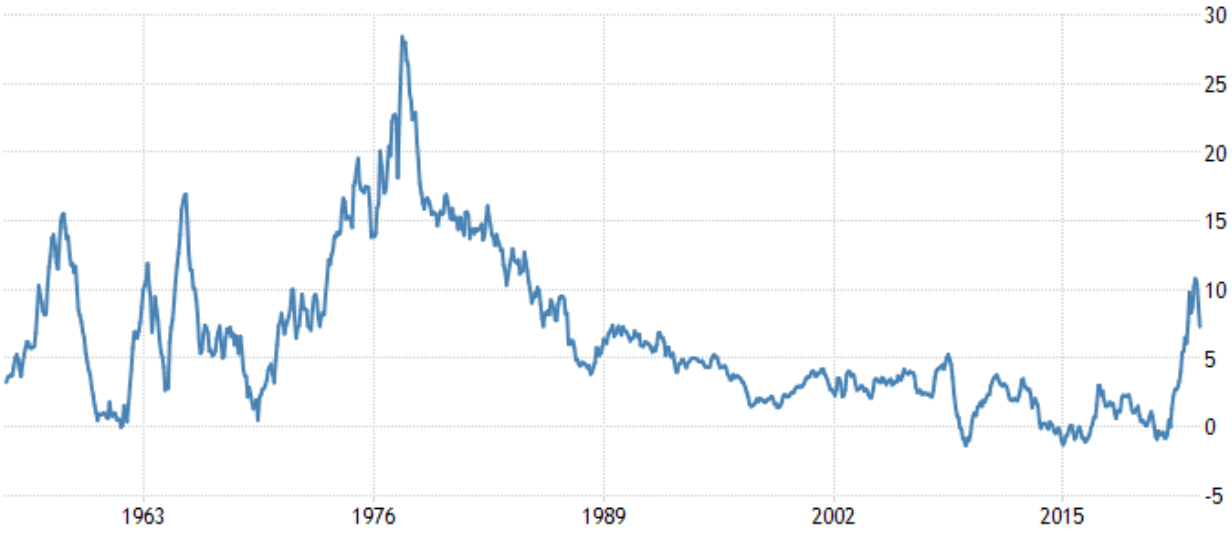
# Tipología y ciclo de vida de los datos
# Practica 1

Imanol Miguez Quintela
Ivan Cuevas Ortin

## 1.-Contexto

Al tiempo de realización de esta práctica, la tasa de inflación en España es cercana al 9%, tasa no alcanzada desde los años 80.

Tasa de inflación en España desde 1950

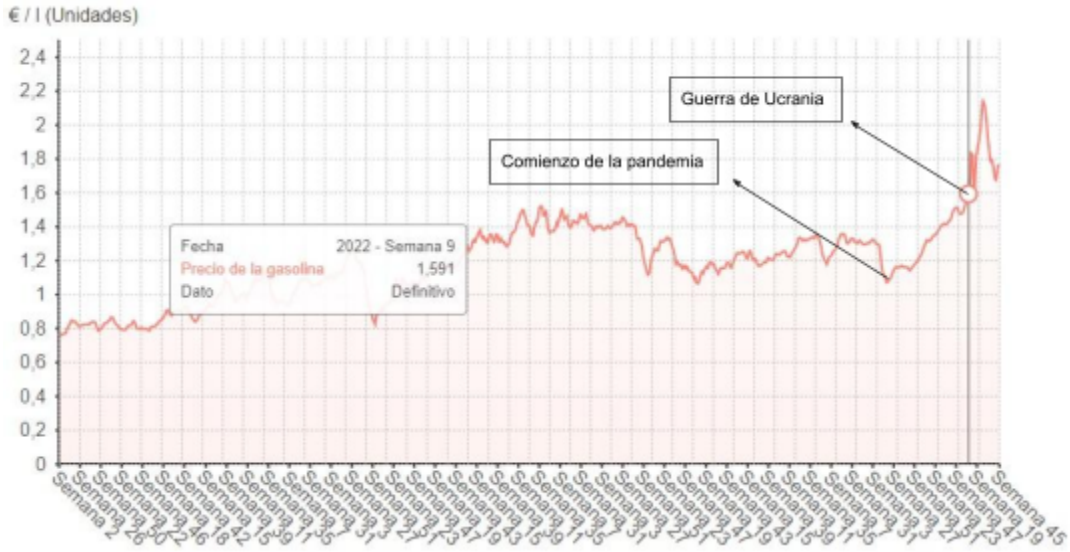



https://es.tradingeconomics.com/. Fuente:INE

Esta subida de precios viene motivada por los trastornos sufridos en la cadena de suministro durante la pandemia, los cambios de hábitos de consumo también causados por la pandemia, y finalmente por la subida de precios de los combustibles debido a la guerra de Ucrania.

Como puede observarse en el siguiente gráfico, el precio de la gasolina se ha incrementado constantemente desde una caída inicial al principio de la pandemia, y se aceleró tras el comienzo de la guerra.

Evolución del precio de la gasolina en España desde 2002




Fuente:https://www.epdata.es/ Boletín Petrolero de la Unión Europea, www.epdata.es

En este contexto inflacionista, conocer el precio de la gasolina en las diferentes estaciones de servicio puede ayudar al consumidor a reducir el gasto mensual en combustible. Por esta razón hemos decidido usar los datos disponibles en www.dieselogasolina.com para crear un conjunto de datos que nos ayude a entender la disparidad de precios entre las diferentes provincias y municipios de España.

La web dieselogasonila.com se define a sí misma como un portal de ayuda al conductor, donde se pueden encontrar datos relativos al precio de los combustibles, matriculaciones, seguros, etc… En cuanto al apartado que nos interesa, la web permite buscar gasolineras por provincia, localidad, tipo de combustible, e incluso tiene una función para encontrar las gasolineras más baratas según la región.

En nuestro caso usaremos el apartado “Buscador de gasolineras”, donde podemos generar la búsqueda por provincia. Mediante el código en Python, generamos los listados para todas las provincias, y de esta manera descargaremos todos los datos para España.

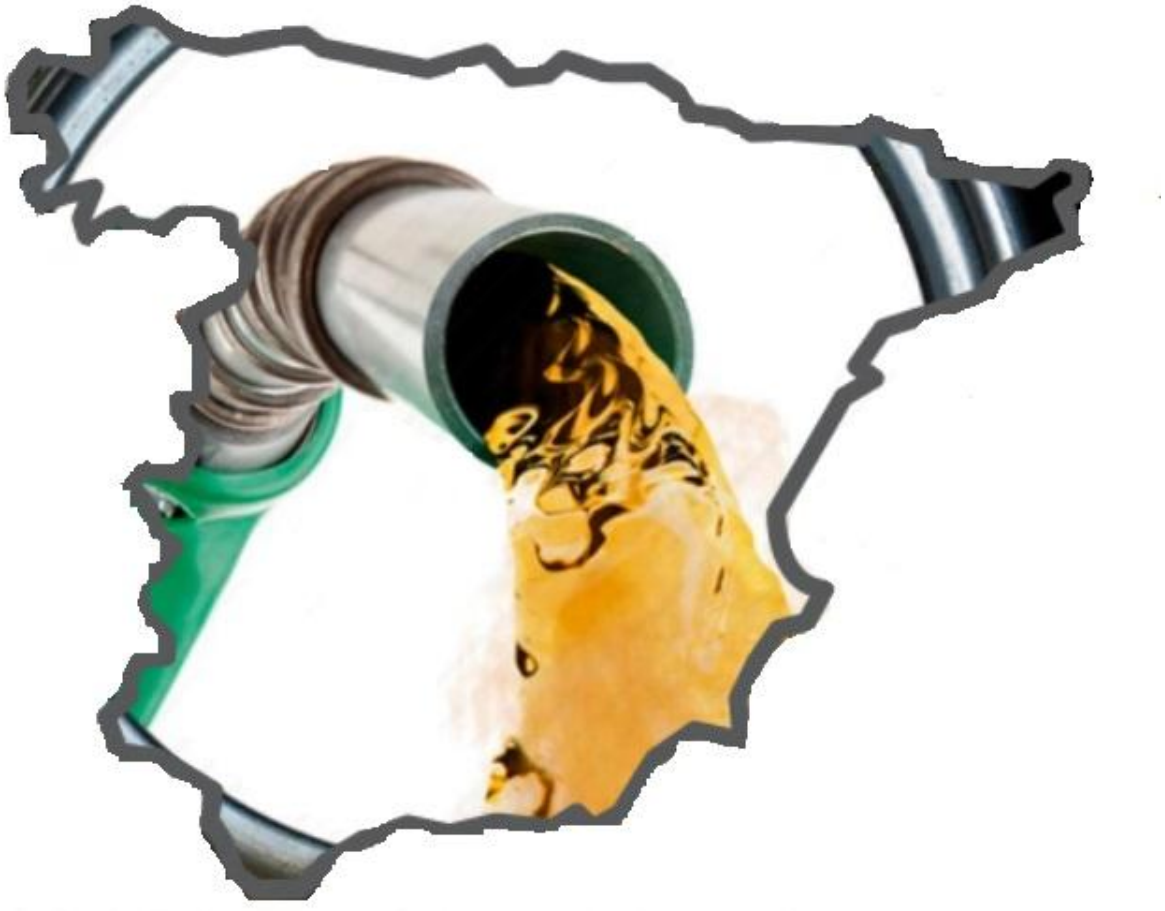
## 2.Titulo

El título elegido para el dataset es “Precio de los carburantes por estación de servicio en España”

## 3. Descripción del dataset.

Como se indica en el título, el dataset incluye datos de precios de los diferentes carburantes para cada una de las estaciones de servicios localizadas en España. También incluye información detallada sobre cada una de las gasolineras con el fin de que estas puedan ser localizadas fácilmente por el usuario final.

## 4. Representación gráfica.



## 5.- Contenido

Estos son los campos incluidos en el dataset:

- Provincia
- Localidad
- Direccion
- Margen
- Horario
- Latitud
- Longitud
- Empresa
- Precios
- fechaRevision
- fechaUltima
- ventaPublico

## 6.- Propietario

Datos de propiedad del sitio web https://www.dieselogasolina.com/

- Nombre: Jesus Lopez
- NIF: 031186775

www.dieselogasolina.com descarga datos del Ministerio de la Transición Ecológica y el Reto Demográfico a diario y posteriormente los procesa y aloja en una base de datos propia para hacerlos accesibles a sus usuarios.

Los datos de las estaciones de servicio son públicos y son proporcionados por el Ministerio de la Transición Ecológica y el Reto Demográfico para descarga de ciudadanos o empresas. Los datos pueden accederse directamente desde https://geoportalgasolineras.es/geoportal-instalaciones/Inicio. Estos datos se actualizan cada hora, y los datos son precisos ya que las gasolineras están obligadas a informar al ministerio de los cambios de precio con 12 horas de antelación.

Estos datos se usan en páginas web similares a dieselogasolina.com (precioscombustibles.com, gasolineras.eu, etc…) con el fin de mostrar los precios de los combustibles en gasolineras cercanas a los usuarios.

Otras páginas web, como rtve.es, muestran la misma información, a la que añaden cierto nivel de análisis con respecto al precio de los carburantes.

También podemos encontrar análisis más detallados como el realizado por la OCU titulado "Las cadenas de gasolineras más baratas", donde se comparan los precios de combustibles por cadena de gasolineras.

Otro uso muy común de estos datos es el desarrollo de aplicaciones móviles. Como ejemplo la aplicación "Gasolineras España", desarrollada por la empresa Mobialia, que recopila los datos desde geoportalgasolineras.es.

Finalmente, encontramos análisis académicos en los que se han utilizado estos datos (o similares):

- ESCRIBANO PABLOS, Beatriz. *Estudio de la competencia de los precios de los carburantes en España mediante Teoría de Juegos y Machine Learning.* Trabajo final de máster: UCM, 2017. Disponible en: https://eprints.ucm.es/id/eprint/45868/1/TFM_Beatriz_Escribano_Pablos.pdf
- RIOLA LUZ, Adrián. *Análisis en retrospectiva y prospectiva del precio de los carburantes en España.* Trabajo fin de grado: UPV, 2022. Disponible en: https://riunet.upv.es/bitstream/handle/10251/183961/Riola%20-%20Analisis%20en%20retrospectiva%20y%20prospectiva%20del%20precio%20de%20los%20carburantes%20en%20Espana.pdf?sequence=1&isAllowed=y

## 7.- Inspiracion

Como ya hemos apuntado anteriormente, el IPC ha alcanzado máximos históricos. Esta subida de precios general ha sido cimentada por la subida del precio de los carburantes desde el comienzo de la guerra de Ucrania.

Este conjunto de datos nos ayudará a entender la diferencia de precios de los combustibles a nivel nacional, y responderá a las siguientes preguntas:

- ¿Cuál es el precio máximo y mínimo de los combustibles a nivel nacional?
- ¿Existen diferencias de precio entre las diferentes comunidades y provincias?
- ¿Tiene influencia en el precio la empresa petrolera de la estación de servicio? (BP, REPSOL…)
- ¿El tipo de combustible tiene influencia en la distribución de precios?
- ¿Hay influencias significativas debidas a la localización de la gasolinera? (ciudad, extrarradio, autopista…)

Otros usos para estos datos podrían ser el análisis competitivo de precios entre estaciones de servicio y la evaluación de oportunidades de negocio para nuevas gasolineras.

## 8.- Licencia

Para nuestro dataset hemos elegido la licencia:

*Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)*

Nos hemos decantado por una licencia Creative Commons ya que es un tipo de licencia establecida y adoptada a nivel mundial. En cuanto al tipo, esta licencia permite compartir los datos en cualquier medio o formato, y al mismo tiempo permite mezclar y transformar, o usarlos como punto de partida para cualquier propósito, incluso comercial.

Los términos a seguir son:

1. Atribución: debe darse el crédito apropiado, proveer un link a la licencia, e indicar si se hicieron cambios.
2. ShareAlike: si los datos transforman de cualquier manera, estos deben distribuirse con la misma licencia que la original

## 9.- Código

Fases de la recolección de datos y su descripción:

1. **Obtención de las url de las provincias.** Para esta fase hacemos uso de la función get_provincias(url_padre). A dicha función le introducimos el url en cuyo HTML están las direcciones de las páginas de búsqueda de cada provincia. La función se encargará de realizar la petición a la página web y procesarla mediante el método html de la librería lxml. Finalmente devolverá una lista con las direcciones de todas las provincias.
2. **Obtención de las urls de las gasolineras.** Una vez tenemos los enlaces de las provincias usaremos la función get_gasolineras(provincias) para explorarlas mediante la librería Selenium. Retornando finalmente una lista con las direcciones web de cada gasolinera. Además, esta función hace uso de otras dos funciones:
    a. driver_configuration(). Establece la configuración de webdriver para cada vez que hacemos una llamada al servidor.
    b. get_gasolineras_provincia(url). Se le introduce un url de una provincia y lo explora mediante la librería Selenium. Retornando una lista con todas las direcciones web de las gasolineras de esa provincia.
3. **Obtención de los datos de cada gasolinera.** Exploramos cada dirección web de las gasolineras y vamos registrando los datos en una lista. Para ello usamos la función parser_content(link), a la cual le introducimos el link de una gasolinera.

La función explora la página mediante html. y recolecta los datos de interés, tras lo cual devuelve una lista con todos los datos de la gasolinera.

4. **Creación y carga de los datos en un archivo CSV**. Creamos un data frame vacío con las columnas: provincia, localidad, direccion, margen, horario, latitud, longitud, empresa, precios, fechaRevision, fechaUltima, ventaPublico. Después, añadimos uno a uno los datos de las gasolineras dentro del data frame. Finalmente, mediante el método to_csv() de la librería pandas, creamos el archivo CSV con los datos.

Otros datos relevantes:

- Utilizamos la librería lxml por comodidad a la hora de usar *XPATH* y por qué es más rápida que *Selenium*.
- Hacemos uso de la librería *fake_useragent* para generar un *User Agent* diferente en cada petición.
- Cada vez que vamos a realizar una petición mostramos con un *print* el *User Agent* utilizado.
- Utilizamos el método *WebDriverWait* para que al usar *Selenium* espere hasta cierto tiempo después de realizar la petición.
- Manejamos los errores *ConnectionError*, *Timeout* y *RequestException* de *requests*, haciendo un *print* mostrando el error del que se trata y realizando una espera de 20 segundos para realizar de nuevo la petición. Además, solo realizamos 3 intentos de petición, tras lo cual se desiste en la petición.
- Hemos creado una función llamada *conversion(tt)* a la que le introducimos unos segundos y nos devuelve un *print* con las horas minutos y segundos. Esta función la hemos usado para que nos indique cuánto tiempo ha tardado en ejecutarse la fase 2 y cuanto la fase 3.
- En el caso de la fase 3, como es la que más tarda, cada vez que se registran los datos de una gasolinera mostramos con un *print* el número de gasolineras que quedan por registrar.

Dificultades y sus resoluciones:

Hemos tenido varias dificultades a la hora de recolectar los datos. Por un lado, el robot.txt nos denegaba el uso de */Buscador/Búsqueda/* que era de donde podíamos sacar los datos. Por otro lado, para conseguir los datos que necesitábamos teníamos que interactuar con una tabla, ya que necesitábamos ir dándole a *next* para poder ir recolectando los datos de la tabla. Además, el robots.txt nos pedía una espera de 30 segundos entre llamadas a la página web.

Hemos tenido varias dificultades a la hora de recolectar los datos. Por un lado, el robot.txt nos denegaba el uso de */Buscador/Búsqueda/* que era de donde podíamos sacar los datos. Por otro lado, para conseguir los datos que necesitábamos teníamos que interactuar con una tabla, ya que necesitábamos ir dándole a *next* para poder ir recolectando los datos de la

tabla. Además, el robots.txt nos pedía una espera de 30 segundos entre llamadas a la página web.

Para solucionar el problema de no poder usar */Buscador/Búsqueda/* tuvimos que encontrar un atajo. Para ello primero exploramos el enlace *…/buscador-gasolineras.html* donde recolectamos las direcciones *…/gasolineras-en-X.html* (X = nombre de cada provincia). Finalmente, de cada provincia sacábamos los enlaces que contenían los datos de cada gasolinera con la extensión */Buscador/Ficha/XXXX* (XXXX = número de la ficha).

En cuanto a la tabla, después de varios intentos vimos que con Selenium se podía hacer clic en botones de la página. Por lo que lo solucionamos explorando con Selenium la parte que necesitaba interacción con la página. De esta manera, para cada enlace de provincia activamos el webdriver hacemos clic en aceptar cookies, recolectamos todos los enlaces de gasolineras de la página de la tabla, hacemos clic en siguiente, volvemos a recolectar los enlaces y así sucesivamente hasta terminar con las páginas de la tabla.

Así mismo, el problema del tiempo es el único que hemos resuelto a medias. La solución era crear una espera, para lo que creamos la función *delay(s)*, que haciendo uso de la librería *time*, creaba una espera de los segundos introducidos. Sin embargo, esto alargaba mucho la recolección de datos, llegando fácilmente a las 90h, lo que lo hacía inviable para usarlo con jupyter y como ejercicio de una asignatura. Por lo que como el robots.txt solo son recomendaciones y no obligaciones, optamos por crear esperas más conservadoras. Teniendo en cuenta que en la fase 1 no eran necesarias las esperas, ya que solo se hacía una llamada al servidor, solo hay que preocuparse de las fases 2 y 3. Por un lado, en la fase 2 sin esperas tarda 4 minutos en ejecutarse y añadiendo a cada llamada una espera de 30 segundos tarda casi 30 minutos, lo que es un tiempo asumible por lo que aquí cumpliremos lo solicitado en *robots.txt*. Por otro lado, la fase 3 tardaría, sin contar la propia ejecución, más de 90 horas. Esto se debe a que hay más de 11000 gasolineras, lo que implica 330000 segundos (30 segundos x 11000) de espera en total, que serían unas 91 horas. Es por ello que para no saturar tanto el servidor escogimos añadir una espera de 30 segundos, pero cada 500 llamadas, haciendo que la ejecución de esta fase tarde en torno a 1 hora 10 minutos.

## 10.- Dataset

El dataset ha sido publicado a Zenodo, y se puede encontrar en el siguiente enlace:

https://zenodo.org/record/7339542

El enlace del DOI es el siguiente:

https://doi.org/10.5281/zenodo.7339542

**11.- Video**

https://drive.google.com/file/d/1T-rZAaol07Z2GjFbCJTYxHt61wsHsH1T/view?usp=sharing

| Contribuciones | Firma |
|---|---|
| Investigación previa | IMQ, ICO |
| Redacción de las respuestas | IMQ, ICO |
| Desarrollo del código | IMQ, ICO |
| Participación en el video | IMQ, ICO |