Práctica 1: Web scraping

Datos sobre los resultados de la Fórmula 1 a lo largo de la historia.

Tabla de contenido

1. Contexto

Se ha optado por crear un set de datos que tenga la información de los resultados de la Fórmula 1 desde sus inicios. Según los organizadores este tipo de eventos las primera carreras oficiales son del 1950, y se realizan anualmente en diferentes circuitos del mundo.

Los datos han sido recopilados desde la web oficial de la Fórmula 1. http://www.formula1.com. Con un poco más de detalle, la página que muestra los resultados de cada carrera a lo largo de los años está en el siguiente enlace: https://www.formula1.com/en/results.html.

Primer contacto

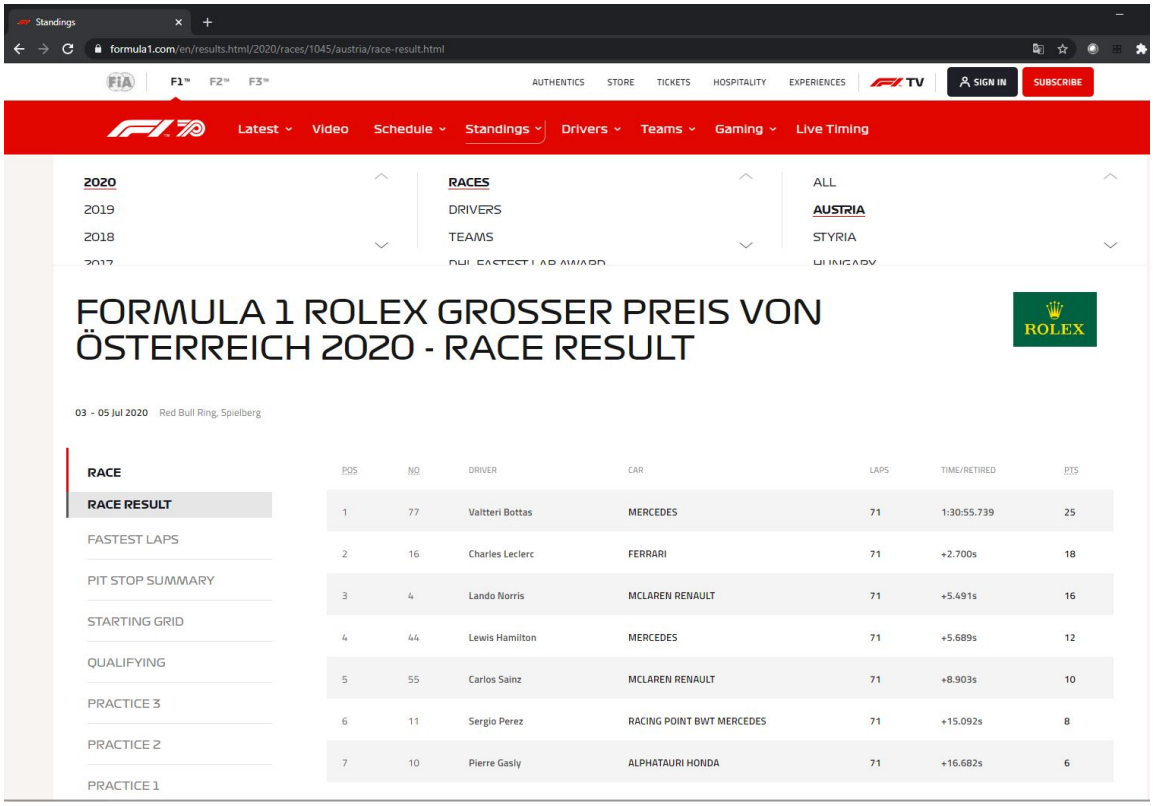
La página permite navegar por los diferentes años, y escoger un año concreto, cargando la información específica de ese año se puede escoger por tipología de información, en nuestro caso serían las carreras (Races), finalmente podemos seleccionar la información específica de cada circuito donde se ha realizado la carrera. Por defecto carga un resumen de quién ha ganado dicha competición, pero en nuestro caso queremos la información completa de como han quedado todos los pilotos que han participado en el evento deportivo.

Una vez escogido el circuito, observamos en la parte izquierda que hay la información de las fechas del evento, el nombre comercial del circuito y la localidad donde se celebra. Además justo debajo observamos diferentes selectores sobre la información a visualizar, en nuestro caso la que se obtiene por defecto, Race Result, es la que contiene la tabla con los resultados que queremos extraer.

Observamos también la tabla de resultados de la carrera. Cada línea es uno de los participantes, y para cada uno se facilitan varios datos, la posición que ha terminado en la carrera, el número del vehículo, el nombre y apellidos del conductor, la escudería o equipo, el número de vueltas que ha dado al circuito, el tiempo de la carrera, que como se observa es el tiempo del primero, el resto que ha entrado en la misma vuelta tiene la diferencia con el primero, y el resto de vehículos doblados indica cuantas vueltas respecto al primero. Y finalmente el número de puntos obtenidos según la posición en la que han finalizado la carrera.

Además, se ha observado que hay información que se visualiza en función del dispositivos, en especial hay el acrónimo o alias de los pilotos, y que también recopilaremos.

A modo de ejemplo, esta es la vista de los resultados del primer gran premio de este año 2020.



Robots.txt

Revisamos las restricciones que se han definido en el fichero robots.txt, para ver si hay alguna restricción que nos pueda afectar a nuestra extracción de datos.

https://www.formula1.com/robots.txt
Sitemap: https://www.formula1.com/content/fom-website/en.sitemap-index.xml
User-Agent: *
Disallow:
Allow: /

Observamos que no tenemos restricciones, ya que se permite a cualquier robot, y no hay directorios excluidos.

Mapa del sitio web

El propio fichero robots.txt nos indica dónde buscar la información relativa a la estructura web. Si accedemos a la misma, obtenemos un árbol con diferentes sitmaps.

formula1.com/content/fom-website/en.sitemap-index.xml

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/sitemap.xml.gz</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.0.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.1.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.2.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.3.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.4.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.5.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.6.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.7.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.8.xml</loc>
  </sitemap>
  <sitemap>
    <loc>https://www.formula1.com/content/fom-website/en/latest/all.sitemap.9.xml</loc>
  </sitemap>
</sitemapindex>
```


Tras revisar los diferentes enlaces, todos ellos apuntan a artículos y noticias, no a la web de resultados de las carreras.

Esta estructura es ideal para mantenerse informado de cualquier noticia oficial que emita la organización.

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://www.formula1.com/en/latest/article.opinion-mercedes-have-earned-the-right-to-call-themselves-f1-greats.LeYBL7NLFmhXxfhCCptrh.html</loc>
    <lastmod>2020-11-05T10:40:25.083Z</lastmod>
  </url>
  <url>
    <loc>https://www.formula1.com/en/latest/article.tsunoda-surprised-by-physical-challenge-of-f1-car-after-maiden-test-as-fp1.3vh0DRWxMpsj9KqLGrLVF7.html</loc>
    <lastmod>2020-11-05T11:52:08.428Z</lastmod>
  </url>
  <url>
    <loc>https://www.formula1.com/en/latest/article.ferraris-all-new-2021-engine-delivering-very-promising-dyno-figures-says.2ea6l0jR9IspOvxNTYgtRo.html</loc>
    <lastmod>2020-11-05T11:55:28.790Z</lastmod>
  </url>
  <url>
    <loc>https://www.formula1.com/en/latest/article.f1-adds-saudi-arabian-grand-prix-night-race-to-2021-calendar.49pVgTPyYV0KBJrOwtqUCN.html</loc>
    <lastmod>2020-11-05T16:50:53.978Z</lastmod>
  </url>
  <url>
    <loc>https://www.formula1.com/en/latest/article.everything-you-need-to-know-about-f1s-new-race-in-saudi-arabia.6aetpPHHw73sKnbsXwSIKA.html</loc>
    <lastmod>2020-11-05T16:51:12.449Z</lastmod>
  </url>
```

Tamaño

Si realizamos la búsqueda en Google, observamos que hay indexados unos 31.700 resultados sobre el site www.formula1.com.

En nuestro caso, nuestra búsqueda es sobre todos los eventos de Fórmula 1 que ha habido desde los inicios. A modo de cálculo rápido, son 71 años de historia, y este año 2020 hay planificados 17 eventos, aunque este número puede variar cada año, nos pone en una cifra cercana a las 1200 carreras con resultados, que serán las peticiones que realizaremos para extraer la información.

Tecnología

Revisamos la tecnología que se ha utilizado en el diseño de la web.

```
import builtwith
builtwith.builtwith("http://www.formula1.com")
```

```
{'cms': ['Adobe CQ5'],
 'programming-languages': ['Java'],
 'advertising-networks': ['DoubleClick for Publishers (DFP)'],
 'tag-managers': ['Google Tag Manager'],
 'javascript-frameworks': ['Prototype', 'RequireJS'],
 'web-frameworks': ['Twitter Bootstrap']}
```

Propietario

Y revisamos quién es el propietario del dominio.

```
import whois
whois.whois("http://www.formula1.com")
```

```
{'domain_name': ['FORMULA1.COM', 'formula1.com'],
 'registrar': 'NOM-IQ Ltd dba Com Laude',
 'whois_server': 'whois.comlaude.com',
 'referral_url': None,
 'updated_date': [datetime.datetime(2020, 3, 10, 23, 1, 53),
  datetime.datetime(2020, 10, 27, 13, 24, 2)],
 'creation_date': datetime.datetime(1999, 4, 9, 4, 0),
 'expiration_date': [datetime.datetime(2021, 4, 9, 4, 0),
  datetime.datetime(2021, 4, 9, 0, 0)],
 'name_servers': ['DNS1.COMLAUDE-DNS.COM',
  'DNS2.COMLAUDE-DNS.NET',
  'DNS3.COMLAUDE-DNS.CO.UK',
  'DNS4.COMLAUDE-DNS.EU',
  'dns1.comlaude-dns.com',
  'dns2.comlaude-dns.net',
  'dns3.comlaude-dns.co.uk',
  'dns4.comlaude-dns.eu'],
 'status': ['clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited',
  'clientTransferProhibited https://icann.org/epp#clientTransferProhibited',
  'clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited',
  'clientDeleteProhibited https://www.icann.org/epp#clientDeleteProhibited',
  'clientTransferProhibited https://www.icann.org/epp#clientTransferProhibited',
  'clientUpdateProhibited https://www.icann.org/epp#clientUpdateProhibited'],
 'emails': ['abuse@comlaude.com',
  'formula1.com-Registrant@anonymised.email',
  'formula1.com-Admin@anonymised.email',
  'formula1.com-Tech@anonymised.email'],
 'dnssec': ['unsigned', 'Unsigned'],
 'name': 'REDACTED FOR PRIVACY',
 'org': 'Formula One Asset Management Limited',
 'address': 'REDACTED FOR PRIVACY',
 'city': 'REDACTED FOR PRIVACY',
 'state': None,
 'zipcode': 'REDACTED FOR PRIVACY',
 'country': 'GB'}
```

2. Definir un título para el dataset

En este caso, un título descriptivo podría ser el de "***Formula_1_historical_results.csv***".

El nombre permite identificar claramente el contenido del dataset.

3. Descripción del dataset

El dataset contiene 12 atributos, que identifican cada resultado de cada una de las carreras realizadas a lo largo de la historia de la Fórmula 1.

Cada entrada tiene la fecha de la carrera, el país donde se realizó el evento, el nombre del circuito, la posición con la que terminó la carrera, en número de vehículo, el nombre del piloto, su apellido y su alias, el equipo con el que corría, cuántas vueltas dio al circuito, la duración de la carrera y los puntos obtenido en la misma.

A modo visual, se muestran las primeras y las últimas entradas del dataset.

	date	country	circuit	position	car_num	name	surname	alias	team	laps	duration	points
0	2020-07-05	Austria	Red Bull Ring, Spielberg	1	77	Valtteri	Bottas	BOT	Mercedes	71	1:30:55.739	25.0
1	2020-07-05	Austria	Red Bull Ring, Spielberg	2	16	Charles	Leclerc	LEC	Ferrari	71	+2.700s	18.0
2	2020-07-05	Austria	Red Bull Ring, Spielberg	3	4	Lando	Norris	NOR	McLaren Renault	71	+5.491s	16.0
3	2020-07-05	Austria	Red Bull Ring, Spielberg	4	44	Lewis	Hamilton	HAM	Mercedes	71	+5.689s	12.0
4	2020-07-05	Austria	Red Bull Ring, Spielberg	5	55	Carlos	Sainz	SAI	McLaren Renault	71	+8.903s	10.0

	date	country	circuit	position	car_num	name	surname	alias	team	laps	duration	points
23133	1950-09-03	Italy	Autodromo Nazionale Monza, Italy	NC	42	Maurice	Trintignant	TRI	Simca-Gordini	13	DNF	0.0
23134	1950-09-03	Italy	Autodromo Nazionale Monza, Italy	NC	46	Consalvo	Sanesi	SAN	Alfa Romeo	11	DNF	0.0
23135	1950-09-03	Italy	Autodromo Nazionale Monza, Italy	NC	44	Robert	Manzon	MAN	Simca-Gordini	7	DNF	0.0
23136	1950-09-03	Italy	Autodromo Nazionale Monza, Italy	NC	30	Prince	Bira	BIR	Maserati	1	DNF	0.0
23137	1950-09-03	Italy	Autodromo Nazionale Monza, Italy	NC	28	Paul	Pietsch	PIE	Maserati	0	DNF	0.0

4. Representación gráfica

La Fórmula 1 desde 1950 hasta 2020.

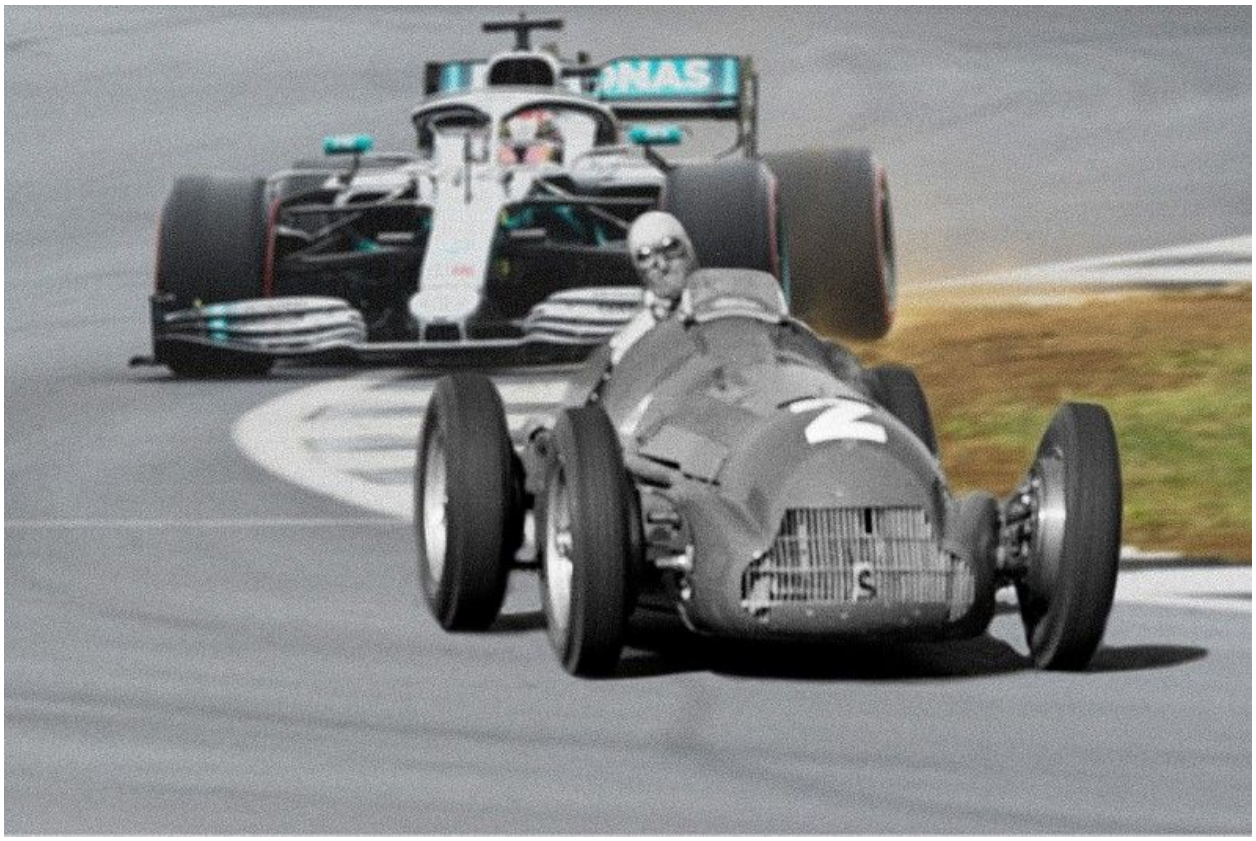


Figura 1*: Composición que visualiza la evolución de los vehículos desde 1950 a la actualidad.

* Imagen obtenida de https://www.autosport.com/f1/news/151152/f1EURTMs-70th-anniversary-how-f1-cars-changed-since-1950

5. Contenido

Tipología de datos

El conjunto de datos está compuesto por 12 atributos, que son:

Date: Fecha del evento.
Formato: aaaa-mm-dd
Tipo de dato: datetime

Country: País donde se realiza el evento.
Formato: Texto libre
Tipo de dato: string

Circuit: Nombre comercial del circuito y localización.
Formato: Texto libre
Tipo de dato: string

Position: Resultado de la carrera.
Formato: Mixto. Existen valores enteros y valores de texto.
Tipo de dato: string
Observación: En algunos casos donde el vehículo no ha llegado a finalizar la competición, en este campo aparece el texto “NC”.

Car_num: Número que tiene asignado el vehículo.
Formato: número
Tipo de dato: int

Name: Nombre del piloto.
Formato: Texto libre
Tipo de dato: string

Surname: Apellido del piloto.
Formato: Texto libre
Tipo de dato: string

Alias: Alias corto del piloto.
Formato: Texto libre
Tipo de dato: string

Team: Equipo o escudería.
Formato: Texto libre
Tipo de dato: string

Laps: Número de vueltas que se ha dado al circuito.
Formato: Mixto. Números y en algún caso un valor vacío.
Tipo de dato: string

Duration: Tiempo requerido por el ganador.
Formato: Mixto. Existen valores de tiempo, valores enteros y texto.
Tipo de dato: string
Observaciones: El vehículo ganador tiene el dato del tiempo de carrera (hh:mm:ss:dd), el resto de vehículos que no han sido doblados, tienen la diferencia en segundos respecto al primero (+ss:dddd) , los doblados tienen el número de vueltas de diferencia (+n) y los que han abandonado tienen un texto de tres letras (“DNF”).

Points: Puntos ganados en función de la posición.
Formato: número
Tipo de dato: float
Observaciones: Se ha detectado que en alguna carrera se ha otorgado una puntuación decimal, por lo que no es posible tratarlo como entero.

Marco temporal de los datos

Los registros van desde 1950, año de la primera carrera oficial de Fórmula 1, hasta la actualidad. La ejecución del código descarga toda la información hasta la última carrera que tenga los datos subidos en la web.
Esto son actualmente 71 años de registros, con un total 1034 carreras hasta la fecha de entrega del ejercicio. Tal y como se ha programado el algoritmo, una nueva ejecución obtiene toda la información hasta la fecha de ejecución.

Extracción de datos

El proceso de obtención de información vía web scraping lo vamos a realizar en tres pasos, en primer lugar, y partiendo de una página web donde están los resultados de la Fórmula 1, procederemos a extraer todas las url de cada año que ha habido carreras. Posteriormente, para cada una de dichas urls, procederemos a obtener el listado de las urls de cada uno de los grandes premios que se han realizado. Finalmente, para cada url de cada gran premio,

obtendremos la información de detalle de la carrera, que es la que incorporaremos en nuestro dataset.

El proceso de descarga empieza con una primera extracción de todos los años que existen datos, y genera las url de cada año. Esta extracción se guarda en una lista.

Para realizar esta extracción se ha creado una función, *F1_year_extract(link)*, a la cual se le provee del link básico donde empezar a trabajar.

Se ha incorporado un delay a cada conexión, es este caso solo se aplica una vez.

A modo de ejemplo ponemos las primera entradas generadas del listado.

```
F1_url_by_year[0:5]
```

```
['http://www.formula1.com/en/results.html/2020/races.html',
 'http://www.formula1.com/en/results.html/2019/races.html',
 'http://www.formula1.com/en/results.html/2018/races.html',
 'http://www.formula1.com/en/results.html/2017/races.html',
 'http://www.formula1.com/en/results.html/2016/races.html']
```

El segundo paso es muy parecido, ya que buscamos obtener el listado de urls de todos los grandes premios que se han realizado, hay un número variable de eventos por cada año.

Se ha definido una función que permite la extracción del listado de eventos de un año concreto al pasarle la url específica de dicho año, que hemos buscado previamente.

Para realizar este proceso, creamos una función, *"F1_prix_extract(link)"*, y buscamos obtener la información de una tabla, por lo que realizaremos la búsqueda de la tabla con un identificador de clase concreto.

A modo de ejemplo, podemos ver una de las tuplas de datos que obtenemos.

```
prix_url_list [0]
```

```
('Austria',
 '05 Jul 2020',
 'http://www.formula1.com/en/results.html/2020/races/1045/austria/race-result.html')
```

El tercer paso es definir una función que partiendo de la url de un evento concreto nos extraiga la información de como ha quedado la carrera, posición de los ganadores, nombres de pilotos, nombres de equipos, etcétera. Para ello creamos la función *"F1_data_extract(link)"*.

En este caso se han detectado elementos de texto que contienen diéresis y acentos circunflejos, por lo que es necesario tener en cuenta la codificación de la página web.

A modo de ejemplo visualizamos un registro de la extracción.

```
records[0]
```

```
('Nürburgring, Nürburgring',
 '1',
 '44',
 'Lewis',
 'Hamilton',
 'HAM',
 'Mercedes',
 '60',
 '1:35:49.641',
 '25')
```

En la parte del código que llama a las funciones que se han definido, se controla la iteración sobre las listas, y se va ordenando adecuadamente la información obtenida.

La ejecución de todo el proceso es superior a los 30 minutos.

Procesado de datos

Los datos se han obtenido como texto, por lo que en los casos que son números enteros, se ha convertido el dato antes de ser incorporado al dataset.

En el caso del atributo del número de vueltas, se ha detectado valores vacíos, por lo que se ha dejado el valor como texto para ser procesado posteriormente en el análisis de datos y validar el valor que debería tener.

Además, se ha procesado la fecha, con el objetivo de dejarla en un formato más adecuado para el tratamiento posterior de los datos.

Buenas prácticas

Se ha verificado que no existe una API oficial por parte de la web que provee los datos.
Se ha implementado un delay para evitar saturación del servidor web.
No es necesario la modificación del *user agent* ya que la web permite la descarga de datos para uso educativo o de investigación.
Se han incorporado textos para indicar que las conexiones son erróneas, indicando la url.
Los datos son obtenidos de la fuente, y se ha asegurado que su extracción es correcta a nivel de codificación. Se han localizado datos faltantes en el dataset, pero es fácil deducir su valor. Dado que estamos en una fase de extracción, no se ha realizado la tarea de completitud de valores faltantes.
Se han revisado los aspectos legales y se permite el uso de los datos con fines educativos.

6. Agradecimientos

Los datos se han extraído de la página http://www.formula1.com, gestionada por "**Formula One Digital Media Limited**".

Se agradece a dicha organización el esfuerzo que han realizado para habilitar la posibilidad de utilizar cierta información de su web adaptando los derechos de autor para fines educativos, permitiendo a los estudiantes descargar la información para propósitos educativos y de investigación, siempre que no se altere la misma. Dichos términos pueden ser consultados en esta página web: https://www.formula1.com/en/toolbar/legal-notices.html.

Existen trabajos similares que se pueden encontrar fácilmente en algunos repositorios, en especial hay algunos que me han llamado la atención.

Se ha localizado un trabajo con el mismo objetivo de extraer información de la información sobre los resultados de la Fórmula 1. En el análisis detalla como analiza la web, extrae la información, y posteriormente realiza algunos gráficos de presentación de la información que permite visualizar algunos de los usos del dataset.

https://towardsdatascience.com/formula-one-extracting-and-analysing-historical-results-19c950cda1d1

No existe una API oficial por parte de la organización, pero existe una API desarrollada por un usuario que permite la extracción de datos.

http://ergast.com/mrd/

El uso de esta API ha sido utilizado por diferentes analistas de datos para descargar la información, a modo de ejemplo, este es un usuario de Kaggel que ha creado diferentes ficheros csv para su posterior análisis. Dichos ficheros contienen más información que la que se ha extraído en esta práctica.

https://www.kaggle.com/cjgdev/formula-1-race-data-19502017

Finalmente, hay un trabajo de análisis de datos de Fórmula 1 que me ha parecido original en su presentación de la información. Dichos datos han sido obtenidos de la web anterior.

https://towardsdatascience.com/formula-1-grand-prix-analysis-d05d73b1e79c

7. Inspiración

La Fórmula 1 es un deporte muy competitivo, donde los equipos el éxito de un campeonato depende de muchos factores, tanto de la innovación tecnológica como de la estabilidad y consistencia del piloto.

Teniendo la información de los resultados de todas las carreras, se puede obtener información estadística sobre el rendimiento de los pilotos y las escuderías, detectar que circuitos se le dan mejor a que pilotos, o incluso se podría obtener una visión de la evolución de los pilotos durante su periódo profesional, y de las escuderías durante periodos de tiempo más extensos, permitiendo analizar la tendencia de los resultados.

8. Licencia

Los datos de este dataset tienen ciertas restricciones de uso por parte del propietario de los mismos.

Es necesario referenciar al autor de los datos, en este caso www.formula1.com.
No se pueden utilizar para fines comerciales.
No se pueden cambiar los datos.
No se pueden cambiar el tipo de licenciamiento de los datos en su redistribución.

Tras revisar los diferentes tipos de licenciamiento disponibles, el que creo más se ajusta a dichos requisitos es el CC BY-NC-SA.

Attribution-NonCommercial-ShareAlike *This license lets others remix, adapt, and build upon your work non-commercially, as long as they credit you and license their new creations under the identical terms.* *Source:* https://creativecommons.org/licenses/	

Este licenciamiento permitiría a otros trabajar con el conjunto de datos, pero siempre referenciando al generador de los datos, no utilizándolos para fines comerciales, no cambiando los datos y no cambiando el tipo de licenciamiento tras su uso.

Para ver las diferencias se ha utilizado la información que se encuentra en la web de Creative Commons: https://creativecommons.org/licenses/

9. Código

```
from bs4 import BeautifulSoup
from bs4.dammit import EncodingDetector
from datetime import datetime
from tqdm import tqdm
import pandas as pd
import requests
import random
import time
import csv


# Raíz de la página sobre la que trabajaremos
url_root = "http://www.formula1.com"
# Página específica donde iniciaremos nuestra extracción de información
url_start = "https://www.formula1.com/en/results.html"


# Creamos un dataset vacio con todas las columnas que vamos a obtener
df1 = pd.DataFrame(columns=['date', 'country','circuit','position'], dtype=str)
df2 = pd.DataFrame(columns=['car_num'], dtype=int)
df3 = pd.DataFrame(columns=['name','surname','alias', 'team'], dtype=str)
df4 = pd.DataFrame(columns=['laps'], dtype=int)
df5 = pd.DataFrame(columns=['duration'], dtype=str)
df6 = pd.DataFrame(columns=['points'], dtype=int)
Formula_1 = pd.concat([df1, df2, df3, df4, df5, df6], axis=1)

# Función extracción url relativos a los diferentes años de los que hay datos.

def F1_year_extract(link):

    # Incorporamos un retraso de entre 0.5 segundos hasta 2 segundos después de cada obtención de
datos
    sleepTimes = [0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 2]
    time.sleep(random.choice(sleepTimes))

    # Realizamos la petición a la web de la Fórmula 1.
    connection = requests.get(link)

    # Verificamos si la petición ha sido correcta, revisando el código que nos ha devuelto la petición.
    if connection.status_code == 200:

        # Descargamos la página raíz de los resultados
        soup = BeautifulSoup(connection.content, "lxml")

        # Creamos una lista vacía para alojar las url de cada año
        years_url_list = []

        # Acotamos al código donde se especifican las url de cada año
```

```
        cod_url_years = soup.find('div', {'class': 'resultsarchive-filter-container'})

        # Buscamos cada atributo "li" que tiene el año definido como texto.
        # Extraemos la url relativa, construimos la url completa y la guardamos.
        for tag in cod_url_years.find_all("li"):
            if tag.text.strip('\n').isdigit():
                link = tag.find("a").get("href")
                url = "%s%s" % (url_root, link)
                years_url_list.append(url)

        return years_url_list
    else:
        print("Error de carga en la página inicial:",link)

# Función extracción url relativos a cada gran premio de cada año.

def F1_prix_extract(link):

    # Incorporamos un retraso de entre 0.5 segundos hasta 2 segundos después de cada obtención de
datos
    sleepTimes = [0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 2]
    time.sleep(random.choice(sleepTimes))

    # Realizamos la petición a la url específica de un año concreto.
    connection = requests.get(link)

    # Verificamos si la petición ha sido correcta, revisando el código que nos ha devuelto la petición.
    if connection.status_code == 200:

        # Creamos una lista vacía para alojar la información de cada Gran Premio de Formula 1
        prix_url_list = []

        # Descargamos la página de cada año
        soup = BeautifulSoup(connection.content, "lxml")

        # Seleccionamos la división de donde extreremos los links de los eventos.
        table = soup.find('table', {'class':'resultsarchive-table'})

        for event in table.tbody.find_all("tr"):
            country = event.a.text.strip('\n ')
            date = event.select('td')[2].text
            link = event.a['href']
            url = "%s%s" % (url_root, link)
            prix_url_list.append((country,date,url))

        return(prix_url_list)

    else:
        print("Error de carga en la página de un año concreto:",link)


# Función extracción información de resultados para cada gran premio.
```

```
def F1_data_extract(link):

    # Incorporamos un retraso de entre 0.5 segundos hasta 2 segundos después de cada obtención de datos
    sleepTimes = [0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 2]
    time.sleep(random.choice(sleepTimes))

    # Realizamos la petición a la web de la Fórmula 1 y revisamos la codificación que tiene.
    connection = requests.get(link)
    http_encoding = connection.encoding if 'charset' in connection.headers.get('content-type', '').lower() else None
    html_encoding = EncodingDetector.find_declared_encoding(connection.content, is_html=True)
    encoding = html_encoding or http_encoding

    # Verificamos si la petición ha sido correcta, revisando el código que nos ha devuelto la petición.
    if connection.status_code == 200:

        # Descargamos la página de cada evento
        soup = BeautifulSoup(connection.content, "lxml", from_encoding=encoding)

        # Acotamos al código donde se especifica el nombre del circuito
        circuit = soup.select('span.circuit-info')[0].text

        # Seleccionamos la tabla donde extraemos la información.
        table = soup.find('table', {'class':'resultsarchive-table'})

        # Creamos una lista donde guardaremos la tupla de datos que obtengamos.
        records = []

        # Recorremos la tabla fila a fila extrayendo la información que nos interesa
        for event in table.tbody.find_all("tr"):
            pos = event.select('td')[1].text
            no = event.select('td')[2].text
            name = event.select('span')[0].text
            surname = event.select('span')[1].text
            alias = event.select('span')[2].text
            team = event.select('td')[4].text
            laps = event.select('td')[5].text
            duration = event.select('td')[6].text
            points = event.select('td')[7].text
            records.append((circuit,pos,no,name,surname,alias,team,laps,duration,points))


        return(records)

    else:
        print("Error de carga en la página de un evento concreto:",link)


# Obtenemos el listado de url de cada año.
F1_url_by_year = F1_year_extract(url_start)
```

```
# Obtenemos el listado de urls de cada gran premio según el año.
F1_event_url_list = []

# Vamos a extraer la información para cada url de cada año
for season in tqdm(F1_url_by_year):

    # Extraemos la lista de url de eventos por cada año
    url_event_list = F1_prix_extract(season)

    # Extraemos los datos de la lista para crear una única lista con la información que nos interesa.
    for event in url_event_list:
        country = event[0]
        date = event[1]
        link_e = event[2]
        reg = (country,date,link_e)

        # Esta lista contendrá todas las url de todos los eventos.
        F1_event_url_list.append(reg)

# Obtenemos los resultados de cada carrera

#for race in F1_event_url_list:
for race in tqdm(F1_event_url_list):

    # Identificamos la información que tenemos en cada elemento de la lista
    country = race[0]
    date = race[1]
    link = race[2]

    # Extraemos los datos de cada carrera
    data = F1_data_extract(link)

    #recorrer la lista data de los resultado de ese evento concreto
    for result in data:

        date_f = datetime.strptime(date, '%d %b %Y').date()

        # Incorporar en el dataset cada registro.
        new_reg = {'date':date_f,
                   'country':country,
                   'circuit':result[0],
                   'position':result[1],
                   'car_num':int(result[2]),
                   'name':result[3],
                   'surname':result[4],
                   'alias':result[5],
                   'team':result[6],
                   'laps':result[7],
                   'duration':result[8],
                   'points':float(result[9])
                  }
```

```
    Formula_1.loc[len(Formula_1)] = new_reg


# Escribimos el dataset obtenido en un fichero csv
Formula_1.to_csv('Formula_1_historical_results.csv', index=False, encoding='utf-8')
```

10. Dataset

Se ha creado una cuenta en Github. https://github.com/Idomingog donde se dejará la información del proyecto.

Se ha subido el fichero csv a https://zenodo.org/, y se ha obtenido un código identificativo DOI: 10.5281/zenodo.4262827

Url de acceso: https://zenodo.org/record/4262827