

Capstone IX: Visualización y BI

Explorando datos de F1

Marta Bellón Castro
Curso 2022-2023

Jesús Martínez Gómez
Máster en Ciencia de Datos e Ingeniería de Datos en la Nube III
Universidad de Castilla-La Mancha

Índice

- [1. Introducción](#)
- [2. Generación de los conjuntos de datos](#)
- [3. Análisis inicial](#)
- [4. Análisis combinado](#)
- [5. Paneles y alertas](#)

1. Introducción

Con este Capstone se pretende poner en práctica algunos de los conocimientos adquiridos durante el módulo 9.

En concreto, se plantea el uso de la herramienta de Business Intelligence AWS Quicksight, la cual está totalmente gestionada por AWS, para lo que debe utilizarse el espacio de AWS creado a tal efecto (46336).

El Capstone requiere un uso correcto de las herramientas vistas durante algunas de las sesiones del módulo, pero también se valorará la iniciativa a la hora de seleccionar los widgets que aporten más valor a los datos bajo estudio, así como la organización general de los mismos.

Sobre todos los ejercicios propuestos, se valorará la correcta elección de:

- Títulos
- Leyendas
- Ordenación
- Organización



2. Generación de los conjuntos de datos.

Además del servicio AWS Quicksight, durante el Capstore haremos uso de S3 como servicio de AWS para el almacenamiento de objetos.

En concreto, la primera acción a realizar consistirá en la carga de un subconjunto de los ficheros .csv incluidos en el reto F1 incluido en Kaggle: <https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020> (<https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020>).

Los ficheros a subir son:

- circuits.csv
- drivers.csv
- races.csv
- results.csv

Estos ficheros deben subirse a un Bucket de S3 creado previamente, y la localización final de los mismos queda a la elección del alumno, pudiendo optar por dejar todos los ficheros en el raíz o usar carpetas.

Tras cargar los datos en S3, se debe acceder a AWS Quicksight y crear un conjunto de datos para los 3 primeros ficheros, creando así un conjunto de datos para circuitos, conductores y carreras, utilizando S3 como fuente de datos (por ahora no creamos el conjunto de datos de resultados).

Si es la primera vez que utilizamos Quicksight, debemos inscribirnos siguiendo las opciones por defecto, salvo la elección de un rol de AWS ya existente (como vimos en las sesiones de clase).

Para crear los conjuntos de datos iniciales (3 conjuntos diferentes), usaremos S3 como origen y crearemos un fichero manifest.json para realizar cada carga, donde se recoge la localización de los mismos.

Tras validar la carga en la vista previa, programaremos una actualización semanal para cada uno de los conjuntos.

EJERCICIO 1: Carga de datos

- Cread 3 conjuntos de datos siguiendo las directrices anteriores, e indicad en la siguiente celda los siguientes datos:
 - Por cada conjunto de datos:
 - Localización del fichero csv dentro del bucket de S3
 - Número de filas cargadas
 - Número de filas omitidas
 - Razón por la que se han omitido las filas
- Circuits.csv
 - Se importaron 74 filas
 - Se omitieron 3 filas: no se pudieron interpretar los valores numéricos del campo alt.
- Drivers.csv:
 - Se importaron 857 filas
 - Se omitieron 0 filas
- Races.csv
 - Se importaron 1102 filas
 - Se omitieron 0 filas
- Result.csv
 - Se importaron 25834 filas
 - Se omitieron 6 filas: no se pudieron interpretar los valores numéricos del campo number.



3. Analisis inicial de datos

En esta sección se plantea un análisis inicial de los datos cargados en el ejercicio anterior. A tal efecto, se creará un análisis en Quicksight llamado AnálisisS3 con una hoja por ejercicio, usando como nombre el fichero de datos usado en el ejercicio. Durante el análisis, únicamente añadiremos widgets/elementos visuales y, únicamente en caso de que se nos indique, añadiremos campos calculados.

No se aplicarán filtros, parámetros o acciones.

La validación de cada ejercicio se realizará exportando un pdf en la carpeta exports, y ejecutando la celda a continuación de cada ejercicio, tras reemplazar "sample_export.pdf" por el nombre adecuado.

EJERCICIO 2.1 CIRCUITOS:

- Cread una hoja dentro del análisis que permita explorar los siguientes datos de los circuitos:
 - Países con mayor número de circuitos

- Localización de los circuitos en el mapa, mostrando su posición concreta
- Distribución de la altitud de los circuitos.
- Exportad la hoja a un fichero llamado circuitos21.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por circuitos21.pdf en la siguiente celda de código
- Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Países con mayor número de circuitos:** un gráfico de barras. Cada país se representa con una barra cuya altura refleja la cantidad de circuitos que posee. Además, permite ordenar las barras de manera descendente para mostrar primero los países con más circuitos.
- **Localización de los circuitos en el mapa, mostrando su posición concreta:** un widget de puntos en el mapa porque está diseñado para visualizar datos geoespaciales, permitiendo mostrar los circuitos en su ubicación exacta.
- **Distribución de la altitud de los circuitos:** un histograma, ya que muestra la frecuencia de los datos en diferentes intervalos. En este caso, los intervalos son rangos de altitud y el tamaño de cada barra representa la cantidad de circuitos que se encuentran dentro de ese rango.

```
In [1]: from IPython.display import IFrame
        IFrame("./exports/circuitos21.pdf"+"#zoom=50.0", width=950, height=700)
```

Out[1]:

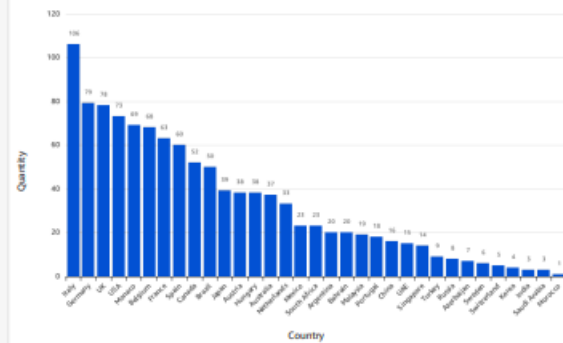
⋮ ⏏ ⏏ ⏏ ⏏ ⋮

— + ⏏ 1 de 1 🔍

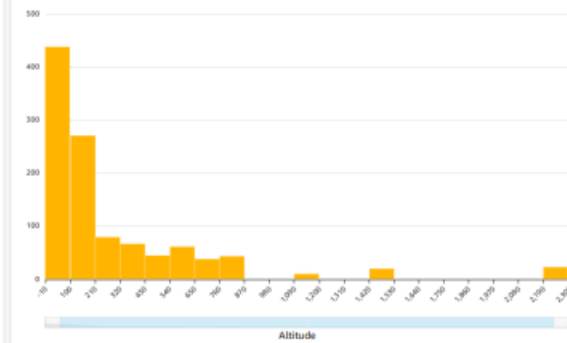
F1 Circuits Analysis

This section will analyse different parameters of F1 circuits: countries with the highest number of circuits, altitude distribution and specific location on the map.

Number of F1 Circuits per country



F1 Circuits altitude



F1 Circuits map



1 de agosto de 2023 9:29 (GMT)

Powered by Quicklight

EJERCICIO 2.2 PILOTOS:

- Cread una hoja dentro del análisis que permita explorar los siguientes datos de los pilotos:
 - Nacionalidades con mayor número de pilotos
 - Preferencias de los pilotos por números par o impar

- Requerirá añadir un campo calculado
- Distribución de la edad de los pilotos (asumimos que todos siguen vivos).
 - Requerirá añadir un campo calculado
- Exportad la hoja a un fichero llamado pilotos22.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por pilotos22.pdf en la celda de código
- Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Nacionalidades con mayor número de pilotos:** un gráfico de barras. Cada país se representa con una barra cuya altura refleja la cantidad de pilotos que posee. Además, permite ordenar las barras de manera descendente para mostrar primero los países con más pilotos.
- **Preferencias de los pilotos por números par o impar:** un gráfico de barras. Para ello, se ha creado un campo calculado 'oddeven' que detecta si el número del piloto es par o impar.
- **Distribución de la edad de los pilotos:** un histograma porque muestra la frecuencia de los datos en diferentes intervalos. En este caso, los intervalos son rangos de edad y el tamaño de cada barra representa la cantidad de pilotos que se encuentran dentro de ese rango. Para realizar este apartado, también se ha tenido que crear un campo calculado para conocer la edad del piloto utilizando la fecha de nacimiento 'dob'.

```
In [2]: from IPython.display import IFrame
        IFrame("./exports/pilotos22.pdf"+"#zoom=50.0", width=950, height=700)
```

Out[2]:

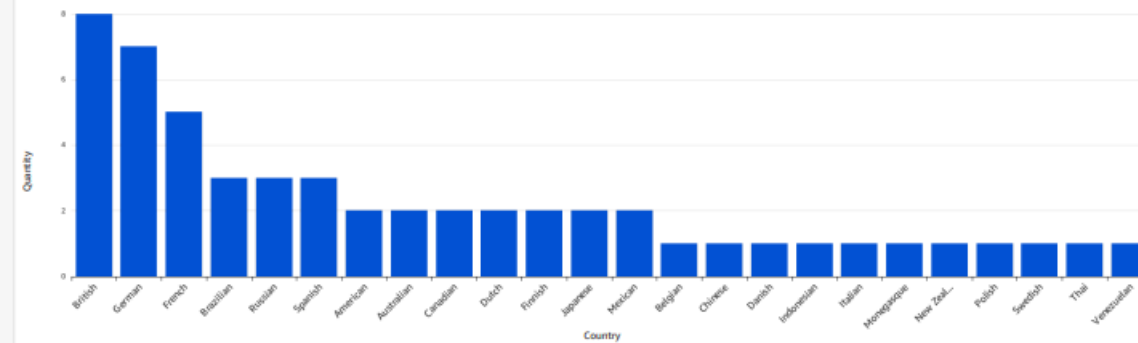
☰ ▾ ▾ ☰ ...

— + ↔ 1 de 1 🔍

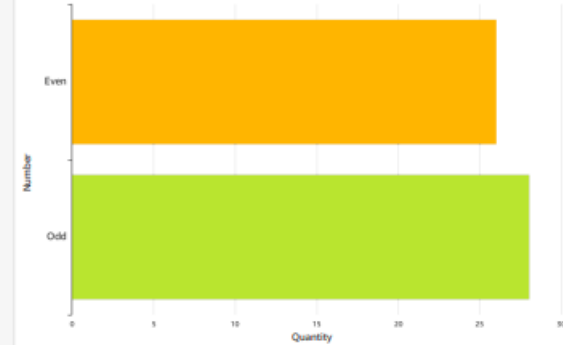
F1 Drivers Analysis

In this section, different parameters will be analyzed: nationalities with the highest number of pilots, pilots with odd or even numbers and age distribution.

Number of F1 pilots per country

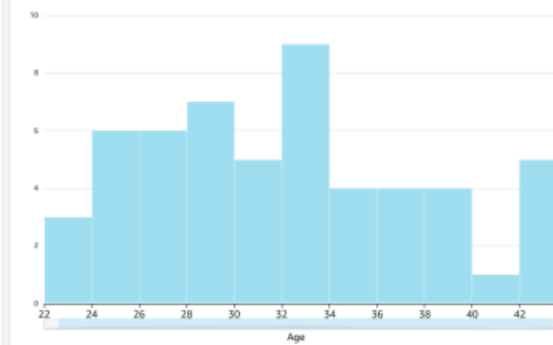


F1 pilots' odd/even numbers



1 de agosto de 2023 9:02 (GMT)

F1 drivers' age



Powered by Quicklight

✍ EJERCICIO 2.3 CARRERAS:

- Cread una hoja dentro del análisis que permita explorar los siguientes datos de las carreras:
 - Relaciones entre nombre la carrera (name) y ronda en la que se realizó (round)
 - Requerirá añadir un campo calculado --> round como string

- Relación entre la ronda, veces que se ha corrido esa ronda, y número diferente de carreras en las que se ha realizado
 - Esperamos que la ronda 1 se haya corrido siempre, pero no así la 20 o 21
- Exportad la hoja a un fichero llamado carreras23.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por carreras23.pdf en la celda de código
- Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Relaciones entre nombre la carrera (name) y ronda en la que se realizó (round):** un gráfico de barras. Cada carrera se muestra como una barra cuya altura indica la cantidad de rondas. Además, permite ordenar las barras de manera descendente para mostrar primero las carreras con más rondas. Para ello, se ha creado un campo calculado 'round-str' que toma el valor de la ronda y lo transforma en string.
- **Relación entre la ronda, veces que se ha corrido esa ronda, y número diferente de carreras en las que se ha realizado:** un diagrama de Sankey ya que es muy útil para mostrar la relación entre estas variables. En el lado izquierdo se encontrarían las rondas, en el medio las veces que se ha corrido cada ronda y en el lado derecho las diferentes de carreras en los que se ha realizado cada ronda. Las líneas conectarían estas variables y el ancho de las líneas representaría la frecuencia.


```
In [3]: from IPython.display import IFrame
        IFrame("./exports/carreras23.pdf", width=950, height=950)
```

Out[3]:

4. Analisis combinado de datos

Una vez realizado el análisis inicial, vamos estudiar cómo combinar acciones y filtros sobre diferentes conjuntos de datos, para así poder realizar un mejor estudio de forma interactiva. Crearemos un nuevo análisis en Quicksight llamado AnálisisS4 y usaremos una hoja por ejercicio.

La validación de cada ejercicio se realizará exportando un pdf en la carpeta exports, y ejecutando la celda a continuación de cada ejercicio, una vez se haya reemplazado "sample_export.pdf" por el nombre indicado.

EJERCICIO 3.1: Conjuntos de datos separados

Sobre AnálisisS4, crea una nueva hoja llamada "Conjuntos de datos separados".

En la misma hoja, vamos a añadir 3 elementos visuales que permitan cierta funcionalidad:

- Widget 1.- Elemento visual con datos de circuitos: mapa rellenado
- Widget 2.- Elemento visual con datos de circuitos: puntos en el mapa
- Widget 3.- Elemento visual con datos de carreras: tipo de elemento visual a elección

Se deben añadir las acciones, filtros y/o parámetros necesarios para obtener la siguiente interactividad:

- Al hacer clic sobre un país en Widget 1, se debe aplicar un filtro en el Widget 2 que muestre sólo los circuitos del país seleccionado
- Al hacer clic sobre un circuito en el Widget 2, se debe aplicar un filtro en el Widget 3 que muestre sólo los datos de carreras realizadas en el circuito seleccionado

Por ejemplo, si pulsamos en España sobre el Widget 1, y elegimos el circuito de Valencia en el Widget 2, el Widget 3 nos debe mostrar que se han realizado carreras en los años 2008, 2009, 2010, 2011 y 2012.

Como ayuda, podemos usar acciones personalizadas de navegación que fijen un parámetro a usar como filtro por otros elementos visuales.

- Exportad la hoja a un fichero llamado datos_separados31.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por datos_separados31.pdf en la celda de código
- Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Un mapa rellenado (filled map)** tomando el campo 'country' como objeto de análisis (circuits.csv).
- **Un widget de puntos en el mapa** usando los campos de latitud y longitud de circuits.csv.
- **Una tabla** que muestra el nombre del circuito, el nombre de la carrera, el país y el año en el que se realizó la competición.

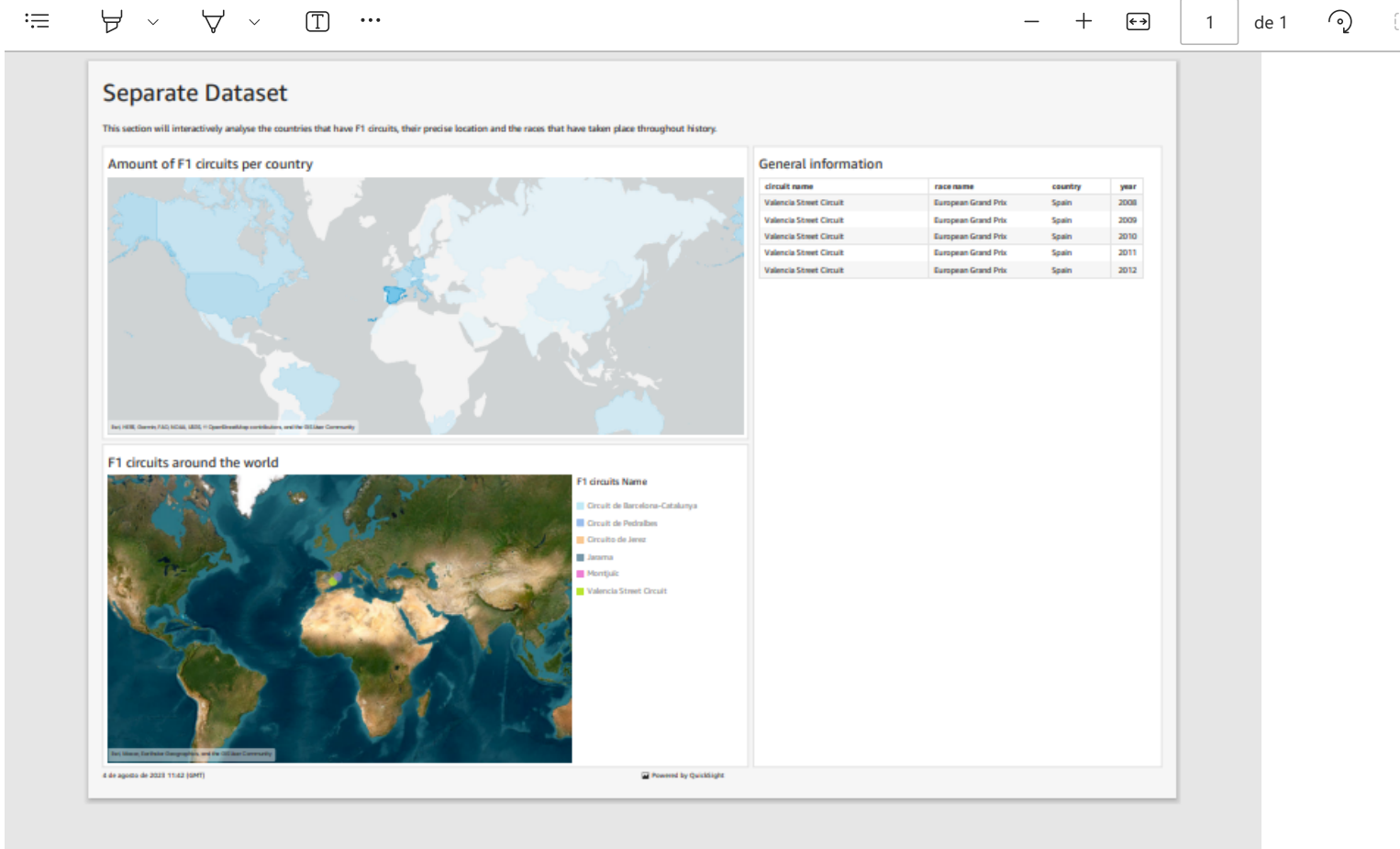
Tareas realizadas:

- Acción en el filled map para filtrar por país en el mapa de puntos.
- Se realiza un join de las tablas 'races' y 'circuits' utilizando de referencia 'circuitId'.
- Acción en el mapa de puntos para filtrar por circuito en la tabla.
- Acción adicional en la tabla, por si quiero seleccionar un circuito desde la tabla. Se filtra esta información a los otros dos widgets.


```
In [5]: # Datos filtrados: España y Circuito de Valencia
```

```
from IPython.display import IFrame
IFrame("./exports/datos_separados312.pdf" + "#zoom=50.0", width=950, height=650)
```

```
Out[5]:
```



EJERCICIO 3.2: Conjuntos de datos unidos

Vamos a crear un nuevo conjunto de datos, llamado `racas_circuits`, combinando 2 conjuntos creados previamente: carreras y circuitos.

Para ello, debes realizar una unión (JOIN) sobre el campo `circuitId`.

Sobre `AnálisisS4`, crea una nueva hoja llamada "Conjuntos de datos unidos", donde trabajaremos con el nuevo conjunto de datos.

En la misma hoja, vamos a añadir 2 elementos visuales que permitan cierta funcionalidad:

- Widget 1.- Elemento visual con datos de grandes premios: gráfico de rectángulos
- Widget 2.- Elemento visual con nombres de circuitos: world cloud

El Widget 1 debe mostrar mostrar, agrupando por el nombre del gran premio (name), el número total y distintivo de circuitos asociados a un gran premio. Por ejemplo, el gran premio de Italia se ha corrido 73 ocasiones, usando para ello 2 circuitos diferentes.

Para evitar mostrar demasiadas entradas, el Widget 1 debe aplicar un filtro que muestre únicamente los 12 grandes premios disputados en más ocasiones.

Se deben añadir las acciones, filtros y/o parámetros necesarios para obtener la siguiente interactividad:

- Al hacer clic sobre un gran premio en el Widget 1, se debe aplicar un filtro en el Widget 2 para que se muestren solamente los nombres de los circuitos asociados al gran premio seleccionado.

Por ejemplo, si pulsamos sobre el gran premio de Italia sobre el Widget 1, el Widget 2 nos debe mostrar solamente 2 nombres de circuitos: "Monza" y "Enzo e Dino Ferrari"

Como ayuda, en este caso será suficiente con una acción personalizada que filtre elementos en la misma hoja.

- Exportad la hoja a un fichero llamado datos_combinados32.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por datos_combinados32.pdf en la celda de código
- Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Un gráfico de rectángulos** porque queremos mostrar la relación proporcional entre los grandes premios y la cantidad de circuitos asociados a cada uno. Se agrupa por el nombre de la carrera, siendo el tamaño y el color correspondientes al nombre del circuito (recuento distintivo)
- **Un gráfico de nube de palabras (word cloud)** muestra los nombres de los circuitos en diferentes tamaños y colores, donde el tamaño de cada palabra es proporcional a su frecuencia en el conjunto de datos. He agrupado por el nombre del circuito y he seleccionado como tamaño el recuento de los nombres del circuito.

Tareas realizadas:

- Filtro superior a 12 en el gráfico de rectángulos para el nombre de las carreras.
- Acción en el gráfico de rectángulos para filtrar los circuitos cuando se seleccione una carrera.


```
In [6]: # Datos sin filtrar

from IPython.display import IFrame
IFrame("./exports/datos_combinados32.pdf"+"#zoom=50.0", width=950, height=800)
```

This section will interactively analyse the F1 circuits associated with Grand Prix events.

Grand Prix	Count
United States Grand Prix	6
French Grand Prix	7
Spanish Grand Prix	5
British Grand Prix	3
Belgian Grand Prix	3
Japanese Grand Prix	2
German Grand Prix	3
Canadian Grand Prix	3
Italian Grand Prix	2
Australian Grand Prix	2
Monaco Grand Prix	1
Brazilian Grand Prix	2
Hungarian Grand Prix	1

[illegible]

```
In [7]: # Datos filtrados: United States Grand Prix

from IPython.display import IFrame
IFrame("./exports/datos_combinados322.pdf", width=950, height=800)
```

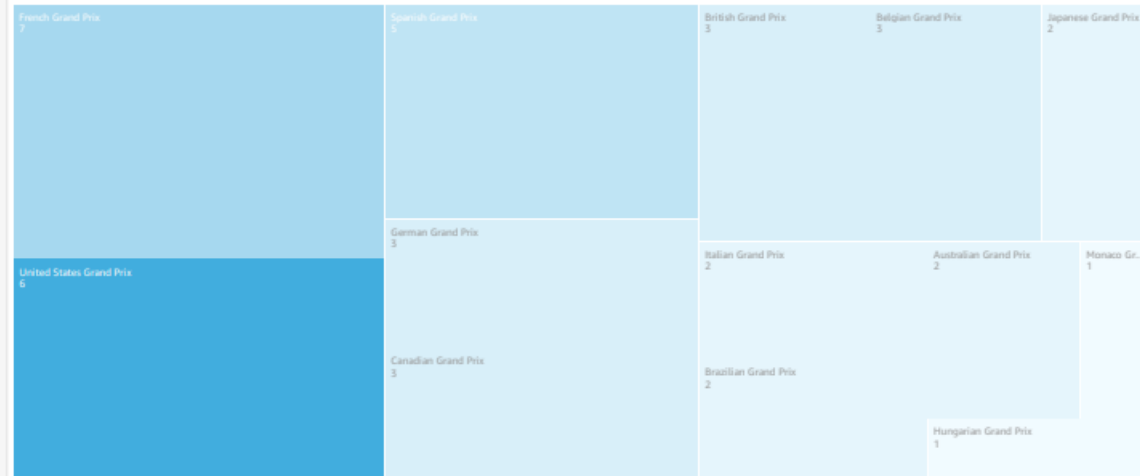
Out[7]:



Analysis of joined dataset

This section will interactively analyse the F1 circuits associated with Grand Prix events.

Total and distinctive number of F1 circuits associated with a Grand Prix



F1 circuit names associated with the selected Grand Prix.

Riverside International Raceway
Phoenix street circuit
Sebring International Raceway
Indianapolis Motor Speedway
Watkins Glen
Circuit of the Americas

4 de agosto de 2023 15:31 (GMT)

Powered by Quicklight



EJERCICIO 3.3: Conjuntos de datos unidos B

Vamos a crear un nuevo conjunto de datos combinado usando los 3 conjuntos creados previamente: carreras, circuitos y pilotos, y un nuevo conjunto de datos que crearemos ahora.

El nuevo conjunto de datos se creará de la misma manera que los de carreras, circuitos y pilotos, usando S3 como origen y tras haber subido, si no se hizo al principio, el fichero results.csv. De nuevo, tras generar y validar el conjunto de datos estableceremos una actualización semanal.

En este caso, a diferencia de los primeros conjuntos de datos, nos vamos a centrar únicamente en un subconjunto de campos, además de variar algunos tipos.

En concreto, solamente vamos a añadir los siguientes campos con sus tipos asociados:

- resultId: número
- raceId: número
- driverId: número
- positionText: cambiamos tipo a cadena
- statusId: cambiamos tipo a cadena

Una vez generado, en la vista de conjuntos de datos, seleccionamos el conjunto de resultados e indicamos: utilizar en un nuevo conjunto de datos, que llamaremos combined_all.

Sobre este nuevo conjunto llamado combined_all, añadiremos los 3 conjuntos de datos de circuitos, carreras y pilotos, usando los siguientes JOINS:

- 1.- resultados <--> carreras unidos por raceId
- 2.- resultados <--> pilotos unidos por driverId
- 3.- carreras <--> circuitos unidos por circuitId

Sobre AnálisisS4, crea una nueva hoja llamada "Conjuntos de datos combinados B", donde trabajaremos con el conjunto de datos combined_all.

En esta hoja, tienes libertad absoluta para elegir elementos visuales, filtros, parámetros y acciones, siempre que permitan, al menos, las siguientes funcionalidades:

- Mostrar todos los pilotos que hayan ganado alguna carrera (positionText es 1), de entre los nacidos (DoB) después de una fecha que pueda ser establecida.
 - Por ejemplo, de los nacidos después de 1995, solamente han ganado 5 pilotos: Ocon, Verstappen, Leclerc, Russell y Gasly
 - Mostrar los pilotos ordenados por el número de colisiones (carreras que terminan con statusId=4)
 - Mostrar los países donde ha habido más resultados con accidente (carreras que terminan con statusId=3)
 - Cualquier otra información que sea útil desde tu punto de vista
-
- Exportad la hoja a un fichero llamado datos_combinados33.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por datos_combinados33.pdf en la celda de código
 - Rellenad la siguiente celda de texto con datos explicativos sobre la elección

He optado por los siguientes widgets:

- **Un gráfico de barras** para mostrar la cantidad de victorias que tuvieron los pilotos de F1.
- **Un gráfico de barras** con el número de colisiones que tuvieron los pilotos.
- **Una tabla dinámica** que muestra más información sobre las colisiones: nombre del piloto, país en donde ocurrió, circuito y el nº de veces.

- **Un mapa de puntos** con la cantidad de accidentes de F1 por país.
- **Una tabla dinámica** en donde se visualiza más información sobre los accidentes: país, circuito y el nº de colisiones.

Tareas realizadas:

- Parámetro de fecha (paramDate) con grado de detalle del periodo de tiempo en día.
- Creación de un campo calculado (MatchingCriteria), en donde se establecía que la fecha de nacimiento de los pilotos tiene que ser mayor o igual a paramDate (`{{dob}} >= ${paramDate}`).
- Filtro con MatchingCriteria cuya condición es que sea igual a 1. Lo he aplicado a todos los widgets de la hoja.
- Filtro en el gráfico de las victorias de los pilotos, en donde 'positionText' es igual a 1.
- Filtro en el gráfico y en la tabla de las colisiones de los pilotos, en donde 'statusId' es igual a 4.
- Filtro en el mapa y en la tabla de los accidentes, en donde 'statusId' es igual a 3.

```
In [8]: # Datos sin filtrar

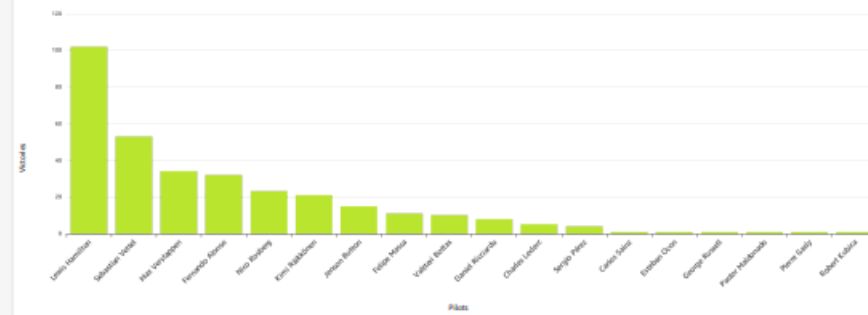
from IPython.display import IFrame
IFrame("./exports/datos_combinados33.pdf"+"#zoom=38", width=950, height=950)
```

Out[8]:

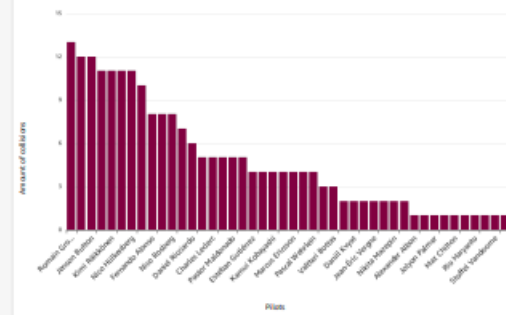
Second analysis of joined dataset

This page displays visualization of winning drivers, drivers sorted by collisions, and countries with the most accidents in Formula 1 races, all governed by a selectable birthdate filter (which can be modified at the top of the sheet).

Winning F1 drivers



F1 drivers' collisions



F1 drivers' collision information

Pilot	Country	Circuit	Count
Adrian Sutil	Bahrain	Bahrain International Circuit	1
	China	Shanghai International Circuit	1
	Hungary	Hungaroring	1
	Korea	Korea International Circuit	2
	Malaysia	Sepang International Circuit	1
	Mexico	Circuit de Mexico	1
	Singapore	Marina Bay Street Circuit	1
Alexander Albon	Spain	Circuit de Barcelona-Catalunya	2
	USA	Circuit of the Americas	1
	UK	Silverstone Circuit	1
Antonio Elisei	Italy	Autodromo Internazionale del Mugello	1
Brendon Hartley	Canada	Circuit Gilles Villeneuve	1
	Mexico	Circuit de Mexico	1
	Bahrain	Bahrain International Circuit	1
Carlos Sainz	Canada	Circuit Gilles Villeneuve	1
	Italy	Autodromo Enzo e Dino Ferrari	1
	Italy	Autodromo Internazionale del Mugello	1
Philippe Alliot	UK	Silverstone Circuit	1
	Bahrain	Bahrain International Circuit	1
	Brazil	Autodromo José Carlos Pace	1
Total			271

F1 accident information

Country	Circuit	Accidents
Australia	Albert Park Grand Prix Circuit	5
Azerbaijan	Baku City Circuit	5
Bahrain	Bahrain International Circuit	2
Belgium	Circuit de Spa-Francorchamps	9
Brazil	Autodromo José Carlos Pace	9
Canada	Circuit Gilles Villeneuve	9
China	Shanghai International Circuit	5
France	Circuit Paul Ricard	1
Germany	Hockenheimring	6
Hungary	Hungaroring	1
India	Buddh International Circuit	1
Italy	Autodromo Enzo e Dino Ferrari	1
Italy	Autodromo Nazionale di Monza	5
Japan	High Speedway	1
San Marino	Imola Circuit	9
Malaysia	Sepang International Circuit	2
Mexico	Autodromo Hermanos Rodríguez	1
Mexico	Circuit de Mexico	22
Spain	San Juan	1
South Africa	Johannesburg	2
Singapore	Marina Bay Street Circuit	10
Spain	Circuit de Barcelona-Catalunya	2
Spain	Valencia Street Circuit	1
USA	Yas Marina Circuit	2
UK	Silverstone Circuit	2
USA	Circuit of the Americas	5
USA	Indianapolis Motor Speedway	2

F1 Accidents



In [9]: *# Imagen del filtro establecido*

```
from PIL import Image
from IPython.display import display

image = Image.open("./exports/filtro.png")
display(image)
```

Hoja 1 ▾ +

Controles

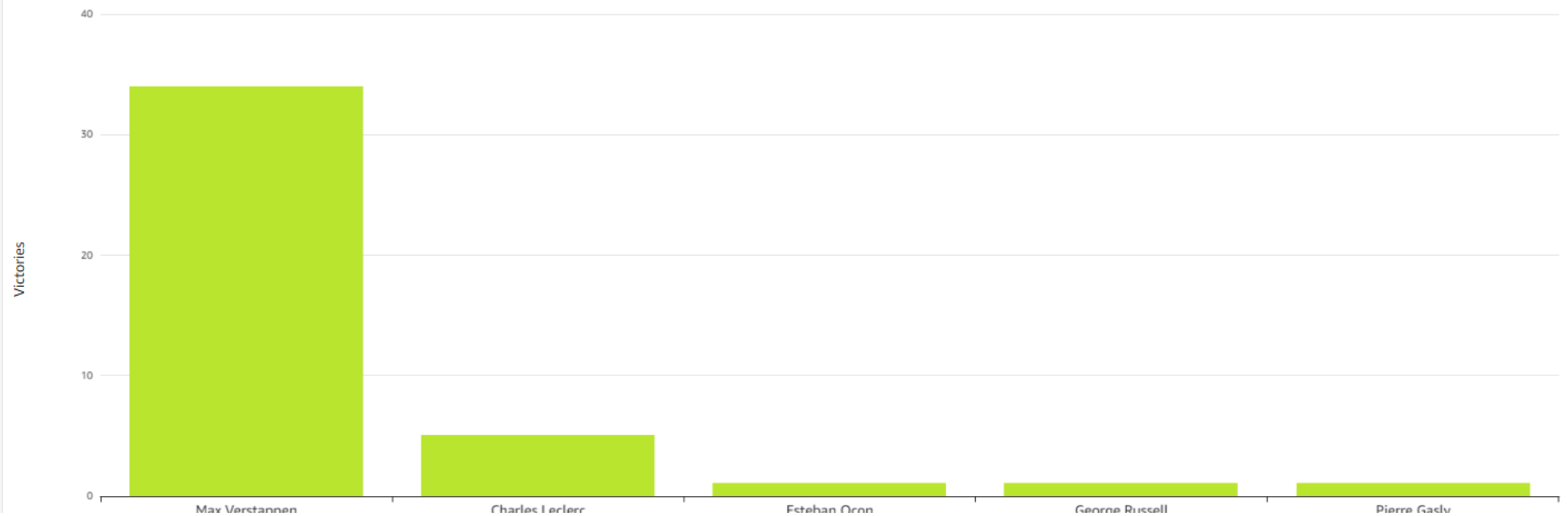
Pilots Born After Set Date

1995/05/05

Second analysis of joined dataset

This page displays visualization of winning drivers, drivers sorted by collisions, and countries with the most accidents in Formula 1 races, all governed by a selectable birthdate filter (which can be modified at the top of the sheet).

Winning F1 drivers



```
In [10]: # Datos filtrados: 1995/05/05

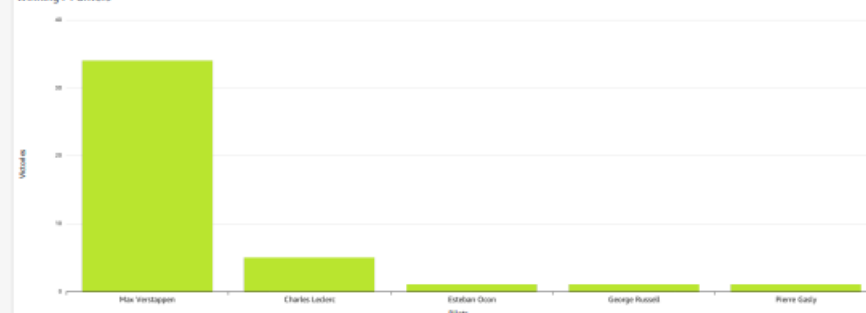
from IPython.display import IFrame
IFrame("./exports/datos_combinados332.pdf"+"#zoom=38", width=950, height=950)
```

Out[10]:

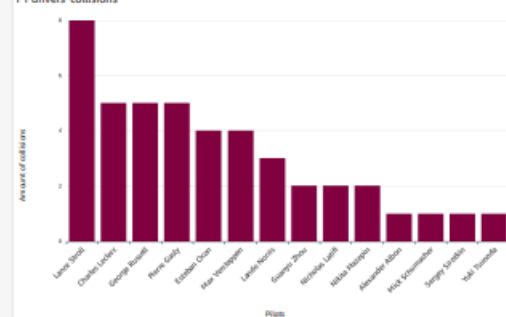
Second analysis of joined dataset

This page displays visualization of winning drivers, drivers sorted by collisions, and countries with the most accidents in Formula 1 race, all governed by a selectable birthdate filter (which can be modified at the top of the sheet).

Winning F1 drivers



F1 drivers' collisions



F1 drivers' collision information

Driver	Country	Circuit	Count
Alexander Albon	UK	Silverstone Circuit	1
Charles Leclerc	Bahrain	Bahrain International Circuit	1
Charles Leclerc	Brazil	Autodromo Jose Carlos Pace	1
Charles Leclerc	Hungary	Hungaroring	1
Charles Leclerc	Monaco	Circuit de Monaco	2
Charles Leclerc	Austria	Red Bull Ring	1
Charles Leclerc	Azerbaijan	Baku City Circuit	1
Esteban Ocon	France	Circuit Paul Ricard	1
Esteban Ocon	Singapore	Marina Bay Street Circuit	1
George Russell	Germany	Nürburgring	1
George Russell	Italy	Autodromo Enzo e Dino Ferrari	1
George Russell	Saudi Arabia	Jeddah Corniche Circuit	1
George Russell	Singapore	Marina Bay Street Circuit	1
George Russell	UK	Silverstone Circuit	1
George Russell	Singapore	Marina Bay Street Circuit	1
George Zhou	UK	Silverstone Circuit	1
George Zhou	Bahrain	Bahrain International Circuit	2
George Zhou	Canada	Circuit Gilles Villeneuve	1
George Zhou	China	Shanghai International Circuit	1
Lance Stroll	Hungary	Hungaroring	1
Total			32

F1 accident information

Country	Circuit	Accidents
Azerbaijan	Baku City Circuit	2
Bahrain	Bahrain International Circuit	2
Belgium	Circuit de Spa-Francorchamps	2
Brazil	Autodromo Jose Carlos Pace	1
Canada	Circuit Gilles Villeneuve	1
France	Circuit Paul Ricard	1
Germany	Hockenheimring	1
Italy	Autodromo Enzo e Dino Ferrari	1
Italy	Autodromo Nazionale di Monza	1
Monaco	Circuit de Monaco	2
Monaco	Circuit de Monaco	1
Saudi Arabia	Jeddah Corniche Circuit	1
Saudi Arabia	Jeddah Corniche Circuit	2
Singapore	Marina Bay Street Circuit	2
UK	Yas Viceroy Circuit	1

F1 Accidents



Country

Place born after last race

100000000


```
In [11]: # Datos filtrados: 1995/05/05, seleccionando el piloto 'Lance Stroll' y 'Monaco'

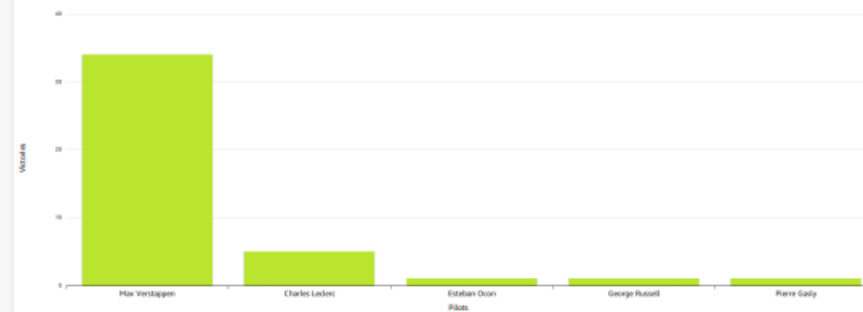
from IPython.display import IFrame
IFrame("./exports/datos_combinados333.pdf"+"#zoom=38", width=950, height=950)
```

Out[11]:

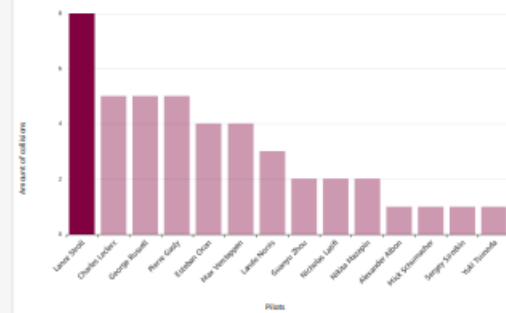
Second analysis of joined dataset

This page displays visualization of winning drivers, drivers sorted by collisions, and countries with the most accidents in Formula 1 races, all governed by a selectable birthdate filter (which can be modified at the top of the sheet).

Winning F1 drivers



F1 drivers' collisions



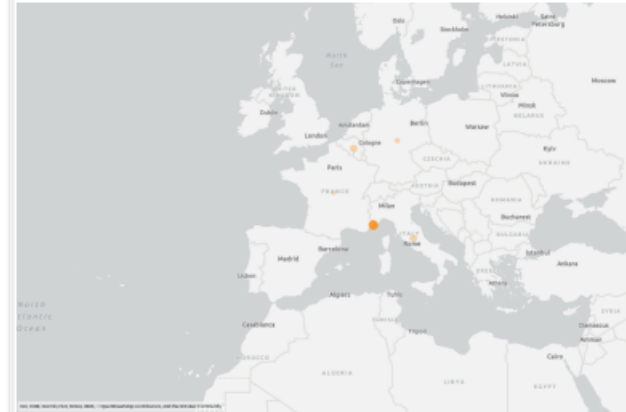
F1 drivers' collision information

Pilot	Country	Circuit	Count
	Bahrain	Bahrain International Circuit	2
	Canada	Circuit Gilles Villeneuve	1
	China	Shanghai International Circuit	1
Lance Stroll	Hungary	Hungaroring	1
Romain Grosjean	Russia	Suzuki Autodrom	1
Yuki Tsunoda	Spain	Circuit de Barcelona-Catalunya	1
	USA	Circuit of the Americas	1
Total			8

F1 accident information

Country	Circuit	Accidents
Monaco	Circuit de Monaco	8

F1 Accidents



Source: F1 Race Data
Data from F1 Race Data

5. Paneles y Alertas

EJERCICIO 4.1: Exportar panel con acceso y alertas

Vamos a crear un análisis que acabaremos exportando como panel, para lo cual creamos un análisis llamado `AnálisisS5` usando el conjunto de datos `combined_all`, usado en el último apartado.

Sobre `AnálisisS5`, usando una única hoja, añadiremos 3 elementos visuales:

- Widget 1.- KPI con el número de victorias obtenidas por pilotos nacidos desde el año 2000
- Widget 2.- Word cloud con los nombres de los pilotos nacidos desde el año 2000
- Widget 3.- Gráfico de barras y líneas que muestre cómo ha evolucionado, año a año, el número de pilotos y nacionalidades diferentes, así como el número total de carreras por año
 - Por ejemplo, en 2021, hubo 21 pilotos de 14 nacionalidades diferentes

Una vez generado el análisis, crearemos un panel llamado `PanelS5` que compartiremos con otro usuario, para lo cual debemos invitarnos, usando un correo secundario que deseemos, a través de "Administrar Quicksight", desde la esquina superior derecha.

Una vez invitado el usuario, dentro del panel, haremos clic en la esquina superior derecha, icono de compartir, compartir panel, e indicaremos el correo al cual hemos realizado la invitación.

Sobre el panel generado, crearemos una alarma para que se nos informe (al usuario administrador) si ha ganado algún piloto nacido en los años 2000 (por ahora sólo participa Tsunoda).

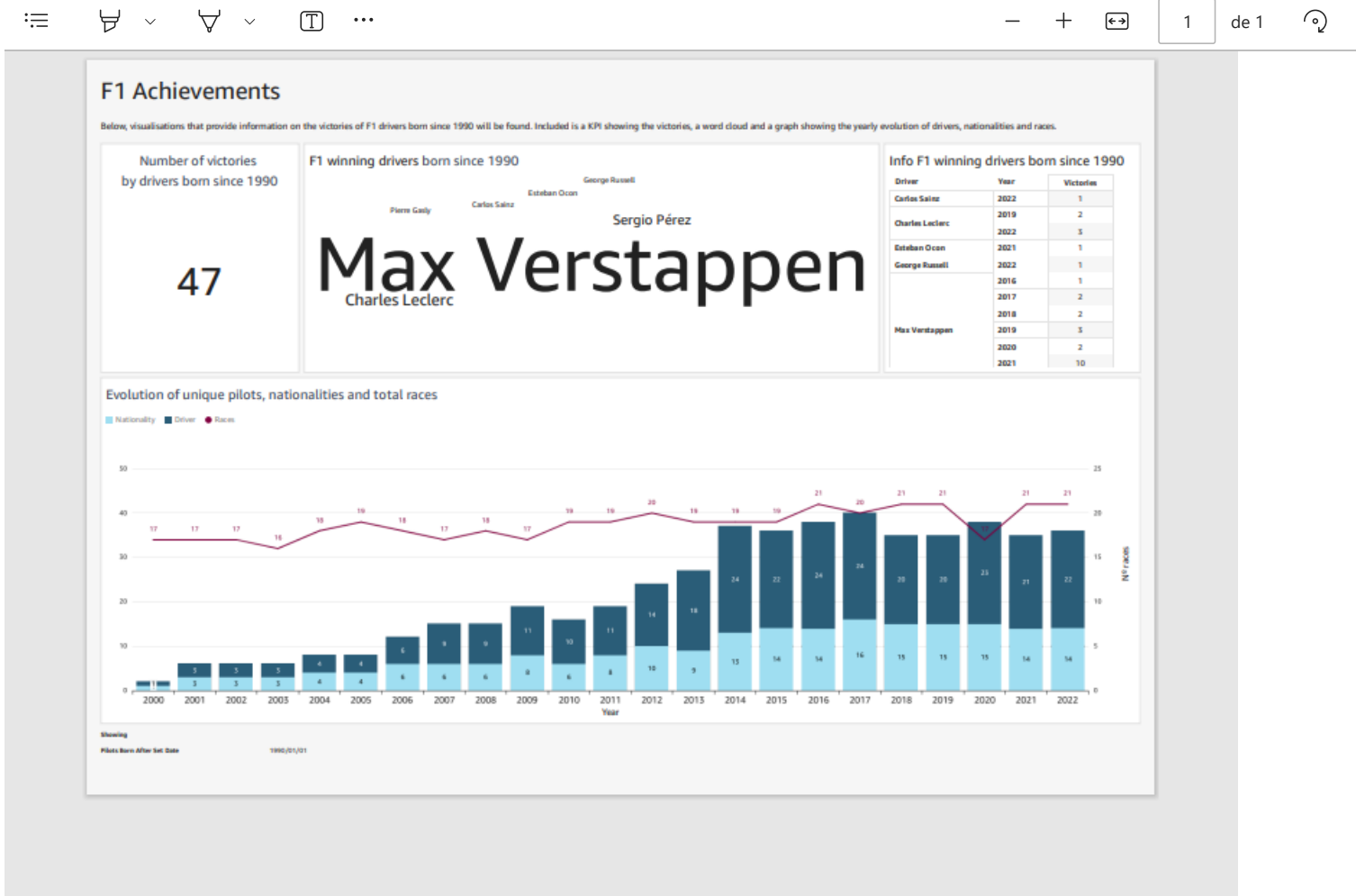
También programaremos el envío de un informe semanal, haciendo clic en la esquina superior derecha, icono del reloj, programaciones. Este informe semanal incluirá un fichero PDF, usando el usuario al que acabamos de dar acceso como destinatario.

- Exportad la hoja del análisis a un fichero llamado `datos_analisis4.pdf` y guardadlo en la carpeta `exports`
 - Reemplazad `sample_export.pdf` por `datos_analisis4.pdf` en la celda de código

In [12]: *# He establecido el año 1990 para realizar el ejercicio, ya que con el año 2000 no aparecía ningún piloto.*

```
from IPython.display import IFrame
IFrame("./exports/datos_analisis4.pdf", width=950, height=700)
```

Out[12]:



- Guardad el PDF recibido en el correo del usuario secundario como datos_correo4.pdf y guardadlo en la carpeta exports
 - Reemplazad sample_export.pdf por datos_correo4.pdf en la celda de código

```
from IPython.display import IFrame
IFrame("./exports/sample_export.pdf", width=640, height=480) # REEMPLAZAR CON EL NOMBRE
```

