

T5.1: Panel de control con PowerBI

Sistemas de Big Data

06/05/2025 – IES Fernando Wirtz

Ralphy Núñez Mercado

Fecha	Motivo del cambio
06/05/2025	Versión inicial
26/05/2025	Versión final

INDICE

ENUNCIADO DE LA TAREA.....	3
1. Explicación del proyecto	4
2. Obtención de los datos	5
2.1. Scraping AutoScout24.....	5
2.2. Obtención de datos de GEI por comunidad autónoma	7
2.3. Script de Python.....	13
3. PowerBi	16
3.1. Primer informe.....	16
3.2. Segundo informe.....	17
3.3. Tercer informe (relación de los datos)	23
3.4. Cuarto informe (Objeto visual de Python).....	25
3.5. Vista de móviles para los informes	26
4. Spark	28
5. Conclusiones	33

ENUNCIADO DE LA TAREA

Hai que facer un proxecto con PowerBI que teña os seguintes elementos:

- Dous informes (e vista para móbiles e ordenador por cada informe).
- Alomenos 2 gráficas ou elementos gráficos ordinarios de PowerBI.
- 1 Mapa (aínda que sexa con outros datos sen relación).
- Dúas orixes de datos, por exemplo un excel e un JSON.
- Relacións entre os datos (alomenos algunha que teña sentido).
- Un obxecto visual de Python.
- Un orixe de datos cun script de Python.

Emprega varios orixes de datos (diferentes) para comparar.

Na práctica debe verse algo de:

- Scrapping
- PANDAS
- Spark-HDFS <-> PowerBi

E ademáis que quede bonito o deseño do informe :)

1. Deberase facer con PowerBI Desktop e publicarse a app.powerbi.com coa conta @fernandowirtz.com.
2. Subirase a esta tarefa o arquivo .pbix. Se é moi grande para subir, darase unha ligazón ao arquivo en OneDrive (coa conta @fernandowirtz.com).
3. Compartirase a ligazón ao informe publicado en app.powerbi.com tamén co profe (pode poñerse no texto da tarefa ou engadir ao membro da organización co email: scj@fernandowirtz.com)

1. Explicación del proyecto

En este proyecto scrapeare la web de [AutoScout24](#) para obtener los coches en venta en España y conseguiré de la web del [miteco](#) (Ministerio para la Transición Ecológica y el Reto Demográfico) las emisiones de [GEI](#) (Gases de Efecto Invernadero) por comunidades autónomas.

El objetivo es analizar si ha aumentado la oferta de coches eléctricos e híbridos como respuesta a regulaciones ambientales más estrictas. También se buscará estimar en qué comunidades autónomas se observa una mayor conciencia ecológica o una aplicación más firme de las medidas impuestas por el gobierno en torno a las emisiones de CO₂, reflejadas en una mayor proporción de vehículos ecológicos en el mercado.

Se usará PoweBI para generar informes sobre el tema en vista para móviles y ordenador, así como un mapa para mostrar los datos geográficamente.

2. Obtención de los datos

Obtención de los orígenes de datos:

- CSV
- JSON
- Script de Python

2.1. Scraping AutoScout24

Para obtener los datos he tenido que scrapear la página de [AutoScout24](https://www.autoscout24.es) para obtener los datos de los coches en venta

Función cerrar_overlays

Busca y cierra ventanas emergentes que interfieren con la interacción automática en la web. Intenta hacer clic en diferentes botones de cerrar mediante diferentes XPath, todos representando distintas variantes de modales. Si se detecta alguno visible, se cierra.

Función limpiar_valor

Se encarga de normalizar los valores de texto obtenidos. Si el valor está vacío o coincide con ciertos textos que indican información ausente o irrelevante (como “- Kilometraje” o “unknown”), devuelve una cadena vacía. En caso contrario, devuelve el valor.

Función extraer_ubicacion_cp

Procesa un string que contiene una ubicación y extrae:

- El código postal (de 4 o 5 dígitos precedido por “ES-”).
- El nombre de la localidad, limpiando caracteres

Configuración del navegador

Se inicializa el navegador Firefox usando webdriver.Firefox. Además, se establece un user-agent personalizado para simular una navegación real desde un navegador. También se configura un WebDriverWait para sincronizar acciones con los tiempos de carga de la web.

Navegación inicial

El navegador se dirige a la URL principal de AutoScout24 con filtros ya aplicados (como país y tipo de vehículo). Si aparece el banner de cookies, se intenta hacer clic en “Aceptar”. En caso contrario, el proceso continúa con normalidad.

Obtención de marcas de vehículos

1. Se hace clic en el campo de selección de marca.
2. Se espera a que aparezca el listado de sugerencias.
3. Se recogen todas las marcas disponibles omitiendo los elementos innecesarios o divisores.

Iteración sobre cada marca

Para cada marca detectada:

- Se recarga la página principal.
- Se vuelve a abrir el selector de marca.
- Se busca la marca exacta en la lista desplegable y se selecciona.
- Si aparece un modal (como un login, que aparece de vez en cuando), se intenta cerrar usando `cerrar_overlays()`.

Scraping de coches por marca

En cada iteración de página (máximo 20):

- Se espera a que carguen los anuncios.
- Se extrae la información de cada coche:
 - Marca, modelo, precio, kilometraje, año, tipo de combustible, caballos (CV), ubicación, transmisión y código postal.
- Si se detectan errores en la extracción de un coche, se ignora ese elemento y se continúa con el siguiente.

Navegación por páginas

Al final de cada página, se intenta avanzar a la siguiente:

- Se comprueba si existe un botón "siguiente".
- Si no está deshabilitado, se hace clic y se espera la recarga de los elementos.
- Si no se encuentra el botón o no hay más páginas, se interrumpe el bucle.

Guardado de resultados

Una vez finalizado el scraping:

- Se escribe un archivo CSV llamado `coches.csv`.
- Este archivo tiene una fila por coche con los campos relevantes.
- Se imprime la cantidad total de vehículos extraídos.

Cierre del navegador

Finalmente, se cierra la instancia de Selenium con `driver.quit()` para liberar recursos.

2.2. Obtención de datos de GEI por comunidad autónoma

Para obtener los datos de las emisiones de [GEI por comunidad autónoma](#) he tenido que ir a la página del [MITECO](#). Estos datos están en tablas en un archivo pdf y mi objetivo es tenerlos en un json.

En el notebook de la tarea primero cargo el pdf y elimino la primera página ya que no me interesa al no contener tablas con información.

```
reader = PdfReader(".\\EMISIONES DE GEI POR COMUNIDADES AUTÓNOMAS Ed. 2024.pdf")
writer = PdfWriter()

for i, page in enumerate(reader.pages):
    if i != 0: # saltamos la primera página
        writer.add_page(page)

with open("Emisiones_sin_primera_pagina.pdf", "wb") as f:
    writer.write(f)
```

Después extraigo las tablas del pdf y las convierto cada una a un fichero csv.

```
pdf = ".\\Emisiones_sin_primera_pagina.pdf"

tables_page1 = tabula.io.read_pdf(pdf, pages=1, multiple_tables=False, lattice=True)
tables_page1[0].to_csv("tabla_pagina_1.csv", index=False)

tables_page2 = tabula.io.read_pdf(pdf, pages=2, multiple_tables=False, lattice=True)
tables_page2[0].to_csv("tabla_pagina_2.csv", index=False)
```

"COMUNIDAD",EMISIONES NETAS DE CO2-eq (kt),Unnamed: 2,Unnamed: 3,Unnamed: 4,Unnamed: 5,Unnamed: 6,Unnamed: 7,Unnamed: 8,Unnamed: 9,Unnamed: 10,Unnamed: 11,
 1990.0,1991.0,1992.0,1993.0,1994.0,1995.0,1996.0,1997.0,1998.0,1999.0,2000.0,2001.0,2002.0,2003.0,2004.0,2005.0
 ANDALUCÍA,34.682,36.479,39.225,36.982,38.089,40.153,37.961,37.923,44.55,47.766,49.058,49.168,52.046,54.905,57.437,62.272
 ARAGÓN,12.625,13.365,14.251,14.057,14.615,14.952,13.733,16332.0,13.431,14.513,16.477,15.019,17.314,17.159,17.578,18.399
 ASTURIAS,26.261,25.397,26.023,24.734,25.545,27.478,23.139,26.135,26.0,31.124,32.63,29.979,32.898,31.46,31.915,33.033
 BALEARES,5.59,5.979,5.842,5.83,6.405,6.705,7.018,7.067,7.553,8.224,8.485,8.632,8.896,10.419,10.066,10.153
 CANARIAS,8.231,8.067,8.2,8.523,9.144,9.152,11.538,11.474,12.326,15.083,15.474,16.249,16.303,16.753,18.803,19.303
 CANTABRIA,4.219,4.165,3.897,3.694,4.402,4.971,4.321,4.603,4.974,4.84,5.039,5.418,5.836,5.918,6.147,6.2
 CASTILLA Y LEÓN,27.467,27.867,27.87,26.951,28.161,29.136,27.528,34.636,29.612,31.819,33.16,32.601,35.192,34.753,36.683,36.522
 CASTILLA-LA MANCHA,14.986,15.529,15.74,14.078,15.732,15.341,16.2,17.409,18.107,19.633,20.238,20.591,20.523,20.247,21.199,20.358
 CATALUÑA,34.077,34.942,36.538,35.053,39.153,44.188,43.049,42.891,43.405,45.686,47.213,45.17,44.541,47.683,49.185,51.408
 CEUTA,697.0,705.0,710.0,762.0,761.0,810.0,875.0,714.0,806.0,748.0,625.0,598.0,628.0,661.0,651.0,637.0
 COMUNIDAD VALENCIANA,16.39,17.463,18.187,17.449,19.418,20.93,20.882,21.639,23.613,25.731,25.812,26.018,27.469,28.149,29.435,31.034
 EXTREMADURA,2.127,2.311,2.17,1.861,2.304,2.198,2.392,2.542,2.539,3.151,3.63,3.749,5.031,5.528,5.556,5.836
 GALICIA,23.783,23.214,25.128,24.933,24.286,26.787,25.064,23.839,25.965,24.484,25.654,25.994,28.144,27.023,28.101,26.99
 LA RIOJA,955.0,970.0,985.0,978.0,1.059,1.16,1.175,1.196,1.365,1.433,1.49,1.509,1.558,1.737,1.915,3.272
 MADRID,15.758,16.945,18.353,17.713,18.929,19.039,20.169,20.158,21.665,22.478,24.064,25.424,25.886,27.082,27.269,28.83
 MELILLA,316.0,345.0,305.0,444.0,448.0,393.0,375.0,290.0,301.0,320.0,329.0,333.0,319.0,342.0,350.0,357.0
 NAVARRA,4.029,4.232,4.076,3.853,4.359,4.636,4.56,4.663,4.93,3.593,3.564,3.899,5.127,6.015,5.757
 PAÍS VASCO,15.361,17.075,16.802,15.796,17.508,16.197,14.925,15.633,16.36,17.502,19.088,18.544,20.823,20.523,21.178,23.036
 REGIÓN DE MURCIA,5.559,5.61,5.83,5.301,5.849,5.887,5.877,5.879,6.492,6.893,7.413,7.74,8.47,7.694,7.705,8.041
 TOTAL DE ESPAÑA,253.113,260.66,270.132,258.912,276.167,290.036,280.857,294.92,303.727,326.358,339.472,336.3,355.776,363.163,377.188,391.438

Comunidad Autónoma, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016

ANDALUCÍA, 34682, 36479, 39225, 36902, 38089, 40153, 37961, 37923, 4455, 47766, 49058, 49168, 52046, 54905, 57437, 62272, 61736, 65184, 56688, 54004, 51427, 50674, 50119, 47114.

ARAGÓN, 12625, 13365, 14251, 14057, 14615, 14952, 13733, 163320, 13431, 14513, 16477, 15019, 17314, 17159, 18798, 18399, 17916, 18881, 17829, 14897, 13021, 16065, 14325, 12408, 1.

ASTURIAS, 26261, 25397, 26023, 24734, 25545, 24748, 23139, 26135, 260, 31124, 3623, 29979, 32898, 3146, 31915, 33033, 2989, 32199, 24587, 21196, 19766, 21446, 22855, 21731, 221.

BALEARES, 559, 5979, 5842, 583, 6405, 6795, 7018, 7867, 7553, 8224, 8485, 8632, 8896, 10419, 10066, 10153, 10204, 10398, 10461, 10321, 10186, 9726, 9125, 8248, 7875, 8071, 825, 8873.

CANARIAS, 8231, 8067, 82, 8523, 9144, 9152, 11538, 11474, 12326, 15083, 15474, 16249, 16303, 16753, 18803, 19303, 17701, 17663, 18292, 15925, 1575, 14529, 14677, 13588, 1325, 130.

CANTABRIA, 4219, 4165, 3897, 3694, 4404, 4691, 4321, 4603, 4974, 484, 5039, 5418, 5836, 5918, 6147, 62, 6299, 6415, 6304, 5555, 5871, 5455, 5451, 482, 5055, 5055, 4922, 5272, 5449, 5.

CASTILLA Y LEÓN, 27467, 27867, 2787, 26951, 28161, 29136, 27528, 34636, 29612, 31819, 3316, 32601, 35192, 34753, 36683, 36522, 33404, 34434, 3098, 22617, 18095, 25797, 2633, 208.

CASTILLA-LA MANCHA, 14986, 15529, 1574, 14078, 15732, 15341, 162, 14709, 18107, 19633, 20238, 20591, 20523, 20247, 21199, 20358, 2065, 21883, 191, 17761, 16369, 15674, 14468, 1.

CATALUÑA, 34077, 34942, 36538, 35053, 39153, 44188, 43409, 42891, 43405, 45686, 47213, 4517, 44541, 47683, 49183, 51408, 50447, 51648, 48093, 45074, 43959, 41407, 38881, 36043, 3.

CEUTA, 6970, 7050, 7100, 7620, 7610, 8100, 8750, 7140, 8060, 7480, 6250, 5980, 6280, 6610, 6510, 6370, 5900, 8030, 6960, 5900, 5390, 4650, 4520, 3560, 3340, 3310, 3540, 4080, 3310, 50.

COMUNIDAD VALENCIANA, 1639, 17463, 18187, 17449, 19418, 2093, 20882, 21639, 23613, 25731, 25812, 26018, 27469, 28149, 29435, 31034, 30165, 31862, 31895, 2774, 26673, 25134, 23.

EXTREMADURA, 2127, 2311, 217, 1861, 2304, 2198, 2332, 2542, 2539, 3151, 363, 3749, 5031, 5528, 5556, 5836, 6593, 6818, 6704, 6397, 6403, 625, 6045, 5733, 5647, 5671, 5459, 5874, 590.

GALICIA, 23783, 23214, 25128, 24923, 24286, 26787, 25064, 23839, 29565, 24484, 25654, 25994, 28144, 27023, 28101, 2699, 26353, 27036, 31106, 18844, 17333, 1894, 21179, 18667, 186.

LA RIOJA, 9550, 9700, 9850, 9780, 1059, 116, 1175, 1196, 1365, 1433, 149, 1509, 1558, 1737, 1915, 3272, 3316, 3196, 2978, 2074, 1988, 1673, 1805, 1406, 1238, 1402, 1575, 1798, 1569, 1.

MADRID, 15758, 16945, 18353, 17713, 18929, 19939, 20169, 20158, 21665, 22478, 24064, 25424, 25886, 27082, 27269, 2888, 29189, 29514, 27083, 26396, 26116, 25738, 24001, 24322, 222.

MELILLA, 3160, 3450, 3050, 4440, 4480, 3930, 3750, 3290, 3010, 3200, 3290, 3330, 3190, 3420, 3500, 3570, 3950, 4160, 4220, 4010, 4110, 4000, 4700, 4100, 3630, 3940, 4320, 5130, 6690.

NAVARRA, 4029, 4232, 4076, 3853, 4359, 4559, 4636, 456, 4663, 493, 3593, 3564, 3899, 5127, 6015, 5757, 5537, 5487, 5755, 5292, 5285, 4177, 3722, 3508, 3424, 3813, 4178, 3873, 3787, 47.

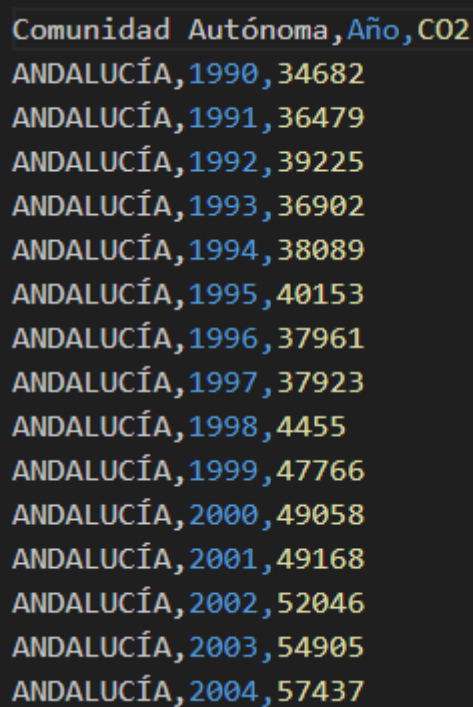
PAÍS VASCO, 15361, 17075, 16802, 15796, 17508, 16197, 14923, 15633, 1636, 17502, 19088, 18544, 20823, 20523, 21178, 23036, 22938, 21873, 21891, 20931, 20277, 16823, 17773, 14833.

REGIÓN DE MURCIA, 5559, 561, 583, 5301, 5849, 5887, 5877, 5589, 6493, 7413, 774, 847, 7694, 7785, 8041, 9896, 11203, 12328, 10071, 9073, 8486, 8899, 8273, 8473, 8118, 7855, 58.

TOTAL DE ESPAÑA, 25313, 26066, 270132, 258912, 276167, 290036, 280857, 29842, 303727, 326358, 339472, 3363, 355776, 363163, 377188, 391438, 383388, 398623, 363202, 326086.

Ahora cambiare la representación de los datos del csv ya que no me pareció la más adecuada para su uso en PowerBi.

```
df = pd.read_csv('.\\emisiones_co2.csv')
df = df[df['Comunidad Autónoma'] != 'TOTAL DE ESPAÑA']
df_long = pd.melt(
    df,
    id_vars=['Comunidad Autónoma'],
    var_name='Año',
    value_name='CO2'
)
df_long = df_long.sort_values(by=['Comunidad Autónoma', 'Año'])
df_long.to_csv('emisiones_co2.csv', index=False)
```



Comunidad Autónoma	Año	CO2
ANDALUCÍA	1990	34682
ANDALUCÍA	1991	36479
ANDALUCÍA	1992	39225
ANDALUCÍA	1993	36902
ANDALUCÍA	1994	38089
ANDALUCÍA	1995	40153
ANDALUCÍA	1996	37961
ANDALUCÍA	1997	37923
ANDALUCÍA	1998	4455
ANDALUCÍA	1999	47766
ANDALUCÍA	2000	49058
ANDALUCÍA	2001	49168
ANDALUCÍA	2002	52046
ANDALUCÍA	2003	54905
ANDALUCÍA	2004	57437

A continuación, para su uso más adelante en PowerBI añadiremos las coordenadas de las provincias autónomas a cada fila de la tabla, esto para después en PowerBi representar los puntos en un mapa.

```
Comunidad Autónoma,Año,CO2,lat,lon
ANDALUCÍA,1990,34682,37.3399964,-4.5811614
ANDALUCÍA,1991,36479,37.3399964,-4.5811614
ANDALUCÍA,1992,39225,37.3399964,-4.5811614
ANDALUCÍA,1993,36902,37.3399964,-4.5811614
ANDALUCÍA,1994,38089,37.3399964,-4.5811614
```

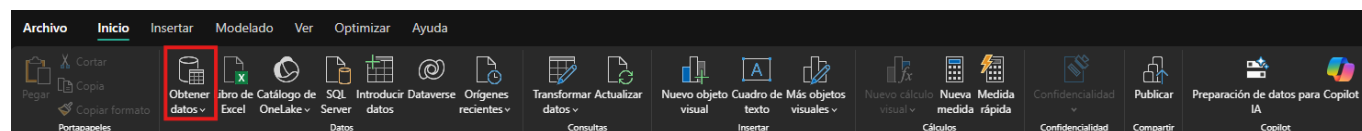
Para acabar convertiremos el archivo csv a json.

```
df = pd.read_csv('emisiones_co2.csv')
```

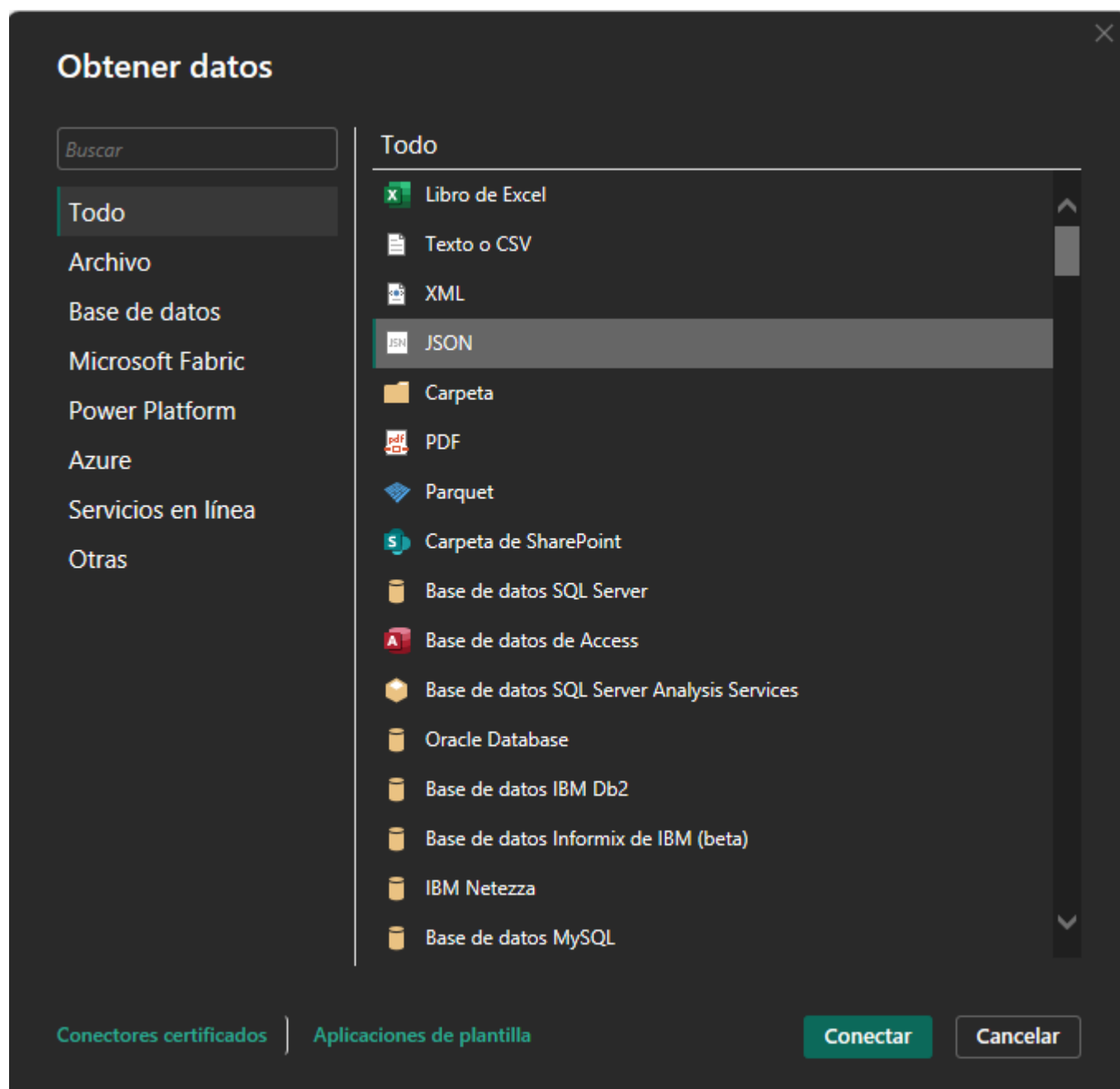
```
df.to_json('emisiones_co2.json', orient='records', lines=True, force_ascii=False)
```

```
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1990,
  "CO2": 34682,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1991,
  "CO2": 36479,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1992,
  "CO2": 39225,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1993,
  "CO2": 36902,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1994,
  "CO2": 38089,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1995,
  "CO2": 40153,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1996,
  "CO2": 37961,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1997,
  "CO2": 37923,
  "lat": 37.3399964,
  "lon": -4.5811614
},
{
  "Comunidad Autónoma": "ANDALUCÍA",
  "Año": 1998,
  "CO2": 4455,
  "lat": 37.3399964,
  "lon": -4.5811614
}
```

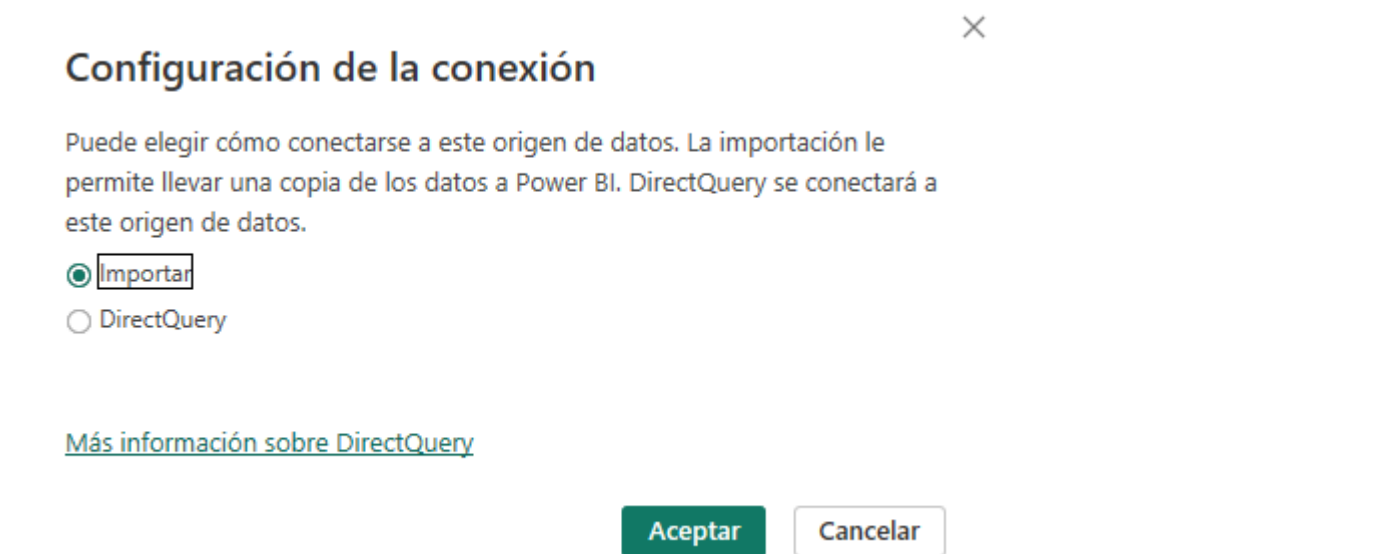
Ahora para poder obtener los datos en PowerBi tenemos que ir al boton de arriba a la izquierda “Obtener Datos”.



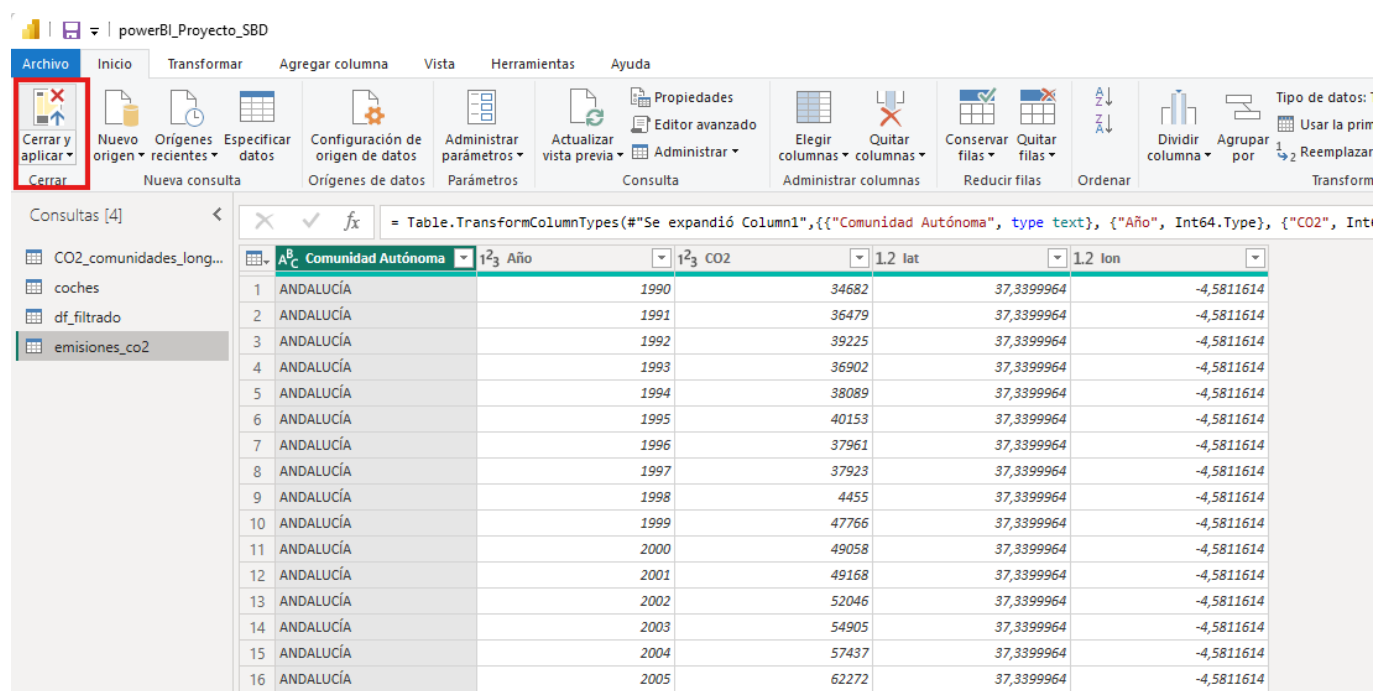
Al darle clic en el botón podremos elegir el formato de nuestros datos a importar, en este caso JSON.



Al darle a conectar tendremos que elegir la configuración de la conexión en este caso le elegiremos a importar.



Comprobamos que los datos se han importado correctamente y hacemos clic en cerrar y aplicar.



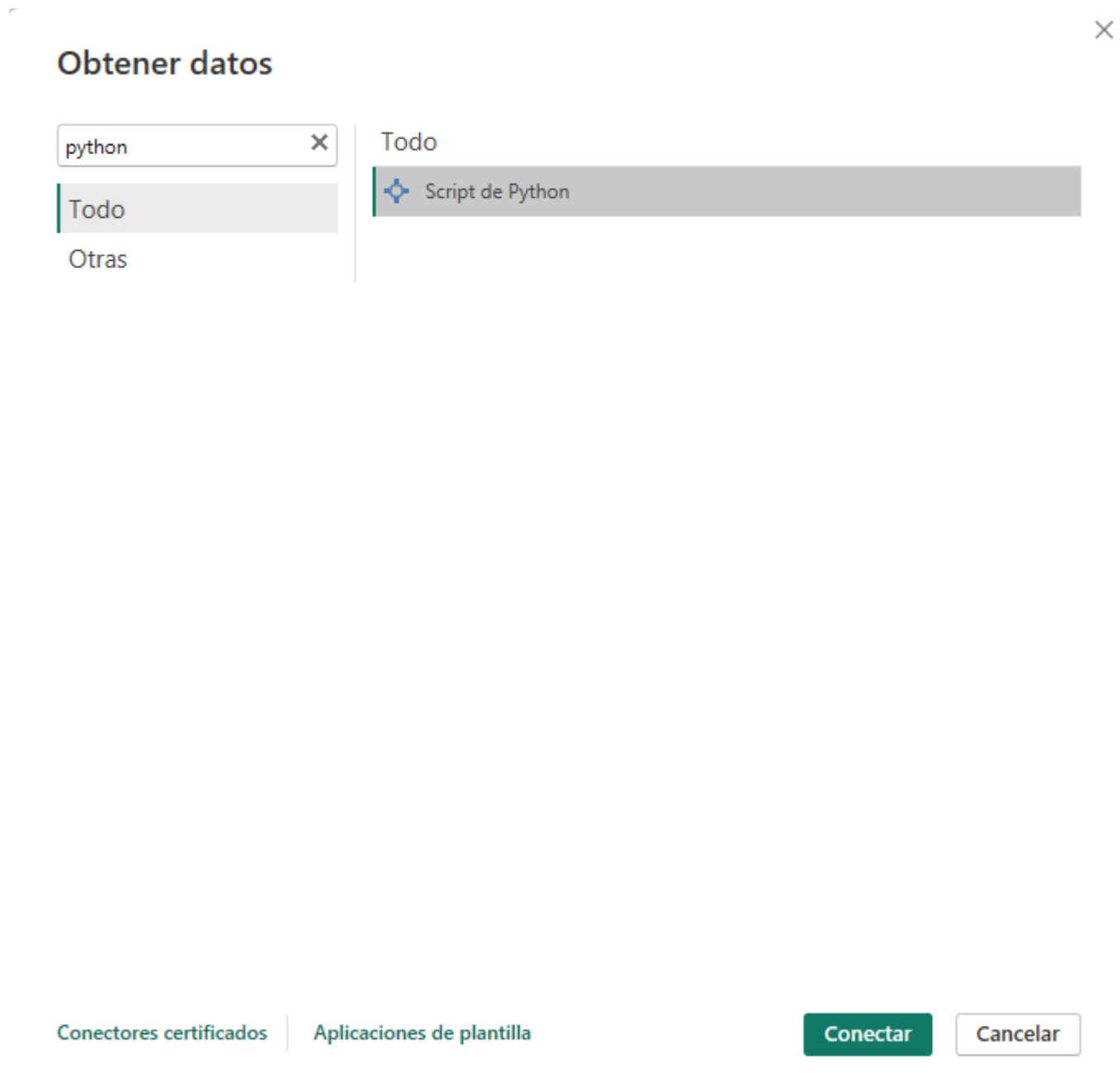
Para el csv arriba haremos algo parecido.

2.3. Script de Python

Para el origen de datos con un script de Python he ido a la página de datos públicos datos.gob y he conseguido las emisiones de GEI de [Cataluña](#).

He descargado él [.csv](#) y he usado un script para importar los datos.

Para ello iremos a obtener datos en powerBI y buscaremos Script de Python.



En el Script solo cogemos las emisiones de CO2 que pertenecen a la subclase de Transporte por carretera.

Script:

```
import pandas as pd

# Leer el CSV
df = pd.read_csv(r"C:\Users\ralfhy.nunezmercado\Downloads\Emissions_de_GEH_a_Catalunya.csv")

# Filtrar por NOM_SUBCLASSE y CONTAMINANT
df_filtrado = df[
    (df['NOM_SUBCLASSE'] == 'Transport per carretera') &
    (df['CONTAMINANT'] == 'CO2')
]

# Mostrar resultado (opcional)
print(df_filtrado)
```

Al darle a aceptar seleccionaremos el dataframe filtrado.

Navegador

Python [2]

df

☒ df_filtrado

df_filtrado

ANY	PROVINCIA	ACTIVITAT	IPCC_SECTOR	IPCC_DIVISIO	IPCC_CLASSE	IPCC_SUBCLASSE	NOM_SECTOR	NOM_DIVISIO	NOM_CLASSE	NOM_SUBCLASSE
1990	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1991	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1992	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1993	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1994	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1995	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1996	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1997	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1998	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
1999	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2000	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2001	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2002	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2003	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2004	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2005	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2006	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2007	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2008	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2009	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2010	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2011	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret
2012	BARCELONA	1A3b	1	A	30	b	Processament de l'energia	Activitats de combustió	Transport	Transport per carret

Cargar

Transformar datos

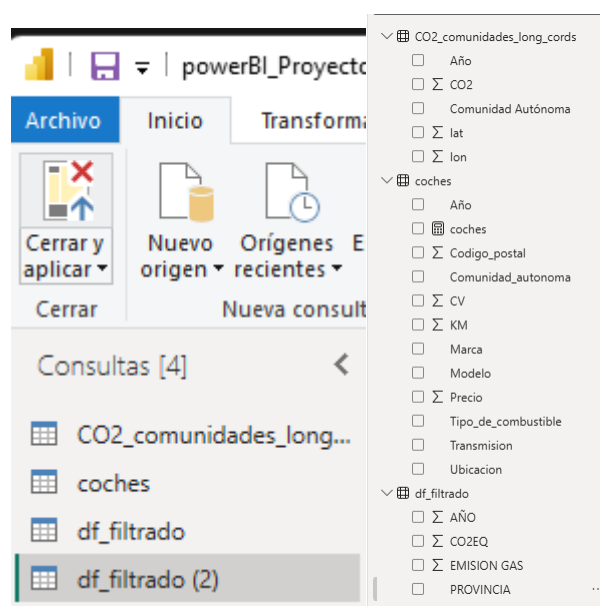
Cancelar

Y le damos a transformar datos para quitar las columnas que no nos interesan y también podemos aprovechar para renombrar las columnas.

Al final nos quedará una tabla como esta.

1 ² 3 AÑO	A ^B C PROVINCIA	1 ² 3 CO2EQ	1 ² 3 EMISION GAS
1990	BARCELONA	6473594843	6473594843
1991	BARCELONA	6786182175	6786182175
1992	BARCELONA	757275525	757275525
1993	BARCELONA	7225804185	7225804185
1994	BARCELONA	7907069269	7907069269
1995	BARCELONA	7693405069	7693405069
1996	BARCELONA	758470462	758470462
1997	BARCELONA	7512894447	7512894447
1998	BARCELONA	8175342567	8175342567
1999	BARCELONA	8024906992	8024906992
2000	BARCELONA	8365729734	8365729734

Al acabar la transformación le damos a cerrar y aplicar.

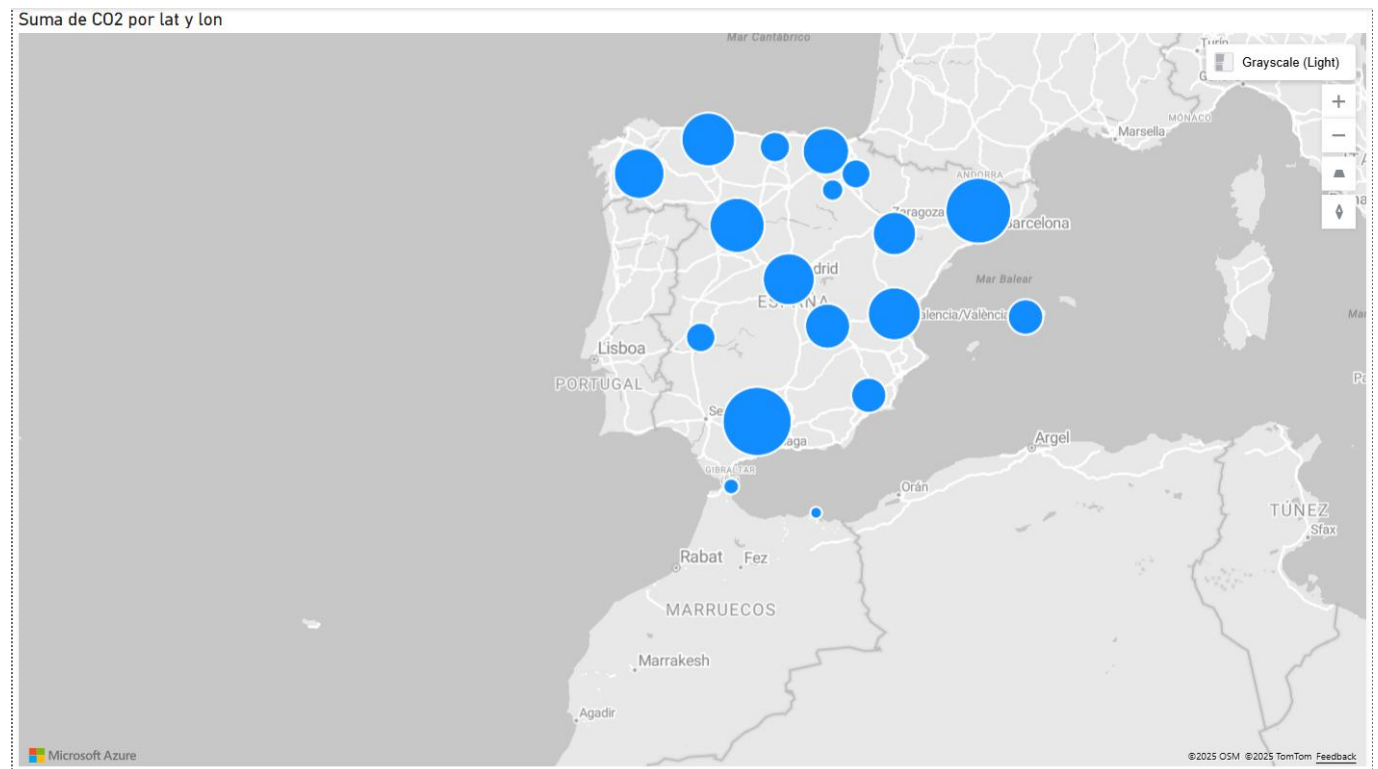


3. PowerBI

Ya con los datos importados en PowerBI podremos generar informes y gráficos.

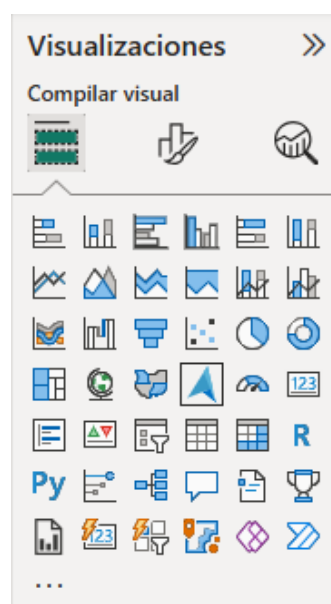
3.1. Primer informe

En el primer informe he colocado un mapa.

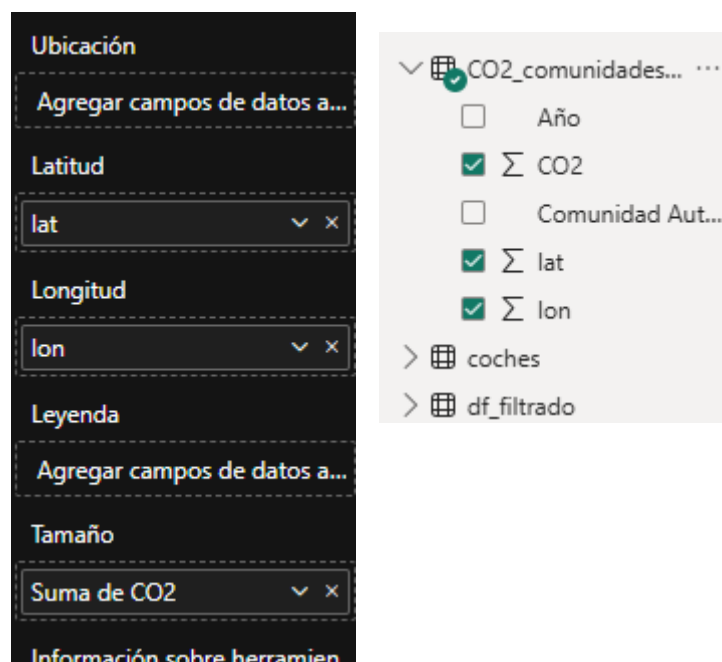


Este mapa representa las emisiones de co2 por comunidad autónoma española.

Para poder graficarlo usaremos un mapa de Azure.



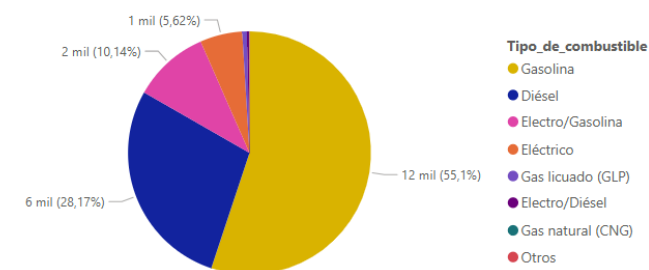
Aquí elegiremos de nuestros datos el json de Co2 y cogeremos la suma de Co2 y las coordenadas de las comunidades.



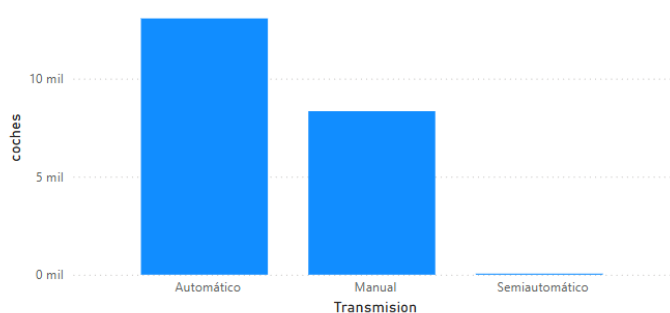
3.2. Segundo informe

En el segundo informe tendremos los siguientes gráficos

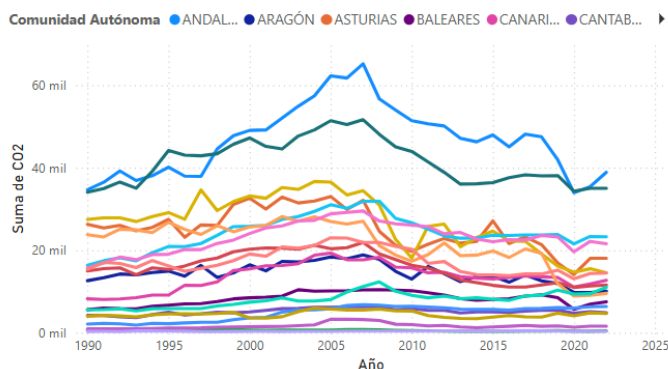
coches por Tipo_de_combustible



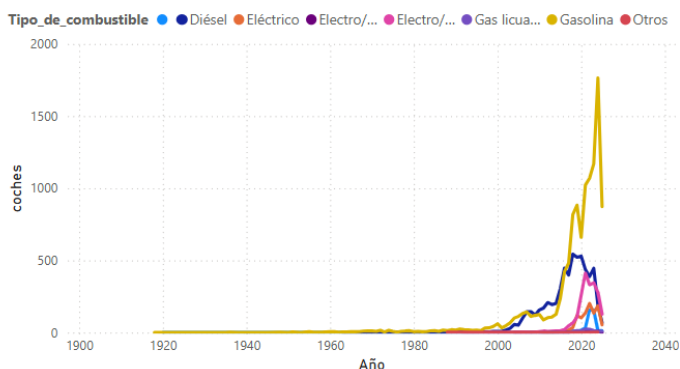
coches por Transmisión



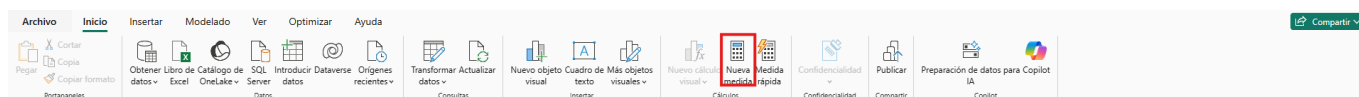
Suma de CO2 por Año y Comunidad Autónoma



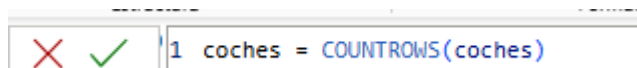
coches por Año y Tipo_de_combustible



Para hacer algunos de los gráficos de este informe y algunos de los siguientes necesitaremos hacer una medida para contar el número de coches para ello tendremos que ir al desplegable de inicio y hacer clic a la opción de “Nueva medida”.

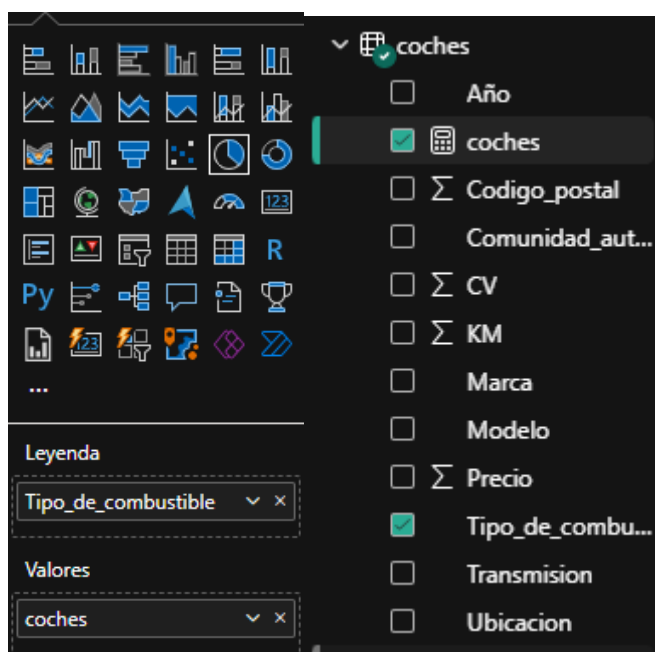


Aquí escribiremos lo siguiente para contar las filas del csv de coches: **“coches = COUNTROWS(coches)”** y le damos al botón de aceptar.



1. Gráfico circular de la cantidad de coches por tipo de combustible.

Para realizar este gráfico necesitaremos el conteo de los coches y el tipo de combustible.



Además, he deseleccionado el tipo de combustible que no tiene dato, no he eliminado directamente los coches ya que el resto de sus características nos pueden servir a la hora de hacer otras graficas.

Tipo_de_combustible
no es

Tipo de filtro ⓘ
Filtrado básico

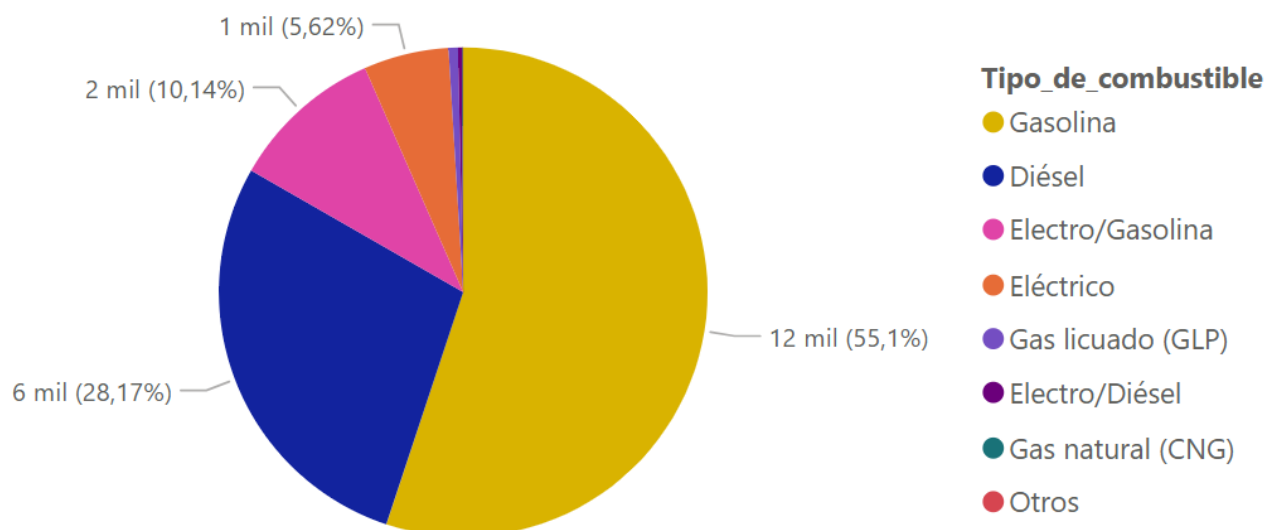
Buscar

<input checked="" type="checkbox"/>	Seleccionar todo	
<input type="checkbox"/>		462
<input checked="" type="checkbox"/>	Diésel	5951
<input checked="" type="checkbox"/>	Eléctrico	1188
<input checked="" type="checkbox"/>	Electro/Diésel	53
<input checked="" type="checkbox"/>	Electro/Gasolina	2141
<input checked="" type="checkbox"/>	Gas licuado (GLP)	131

☐ Requerir selección única

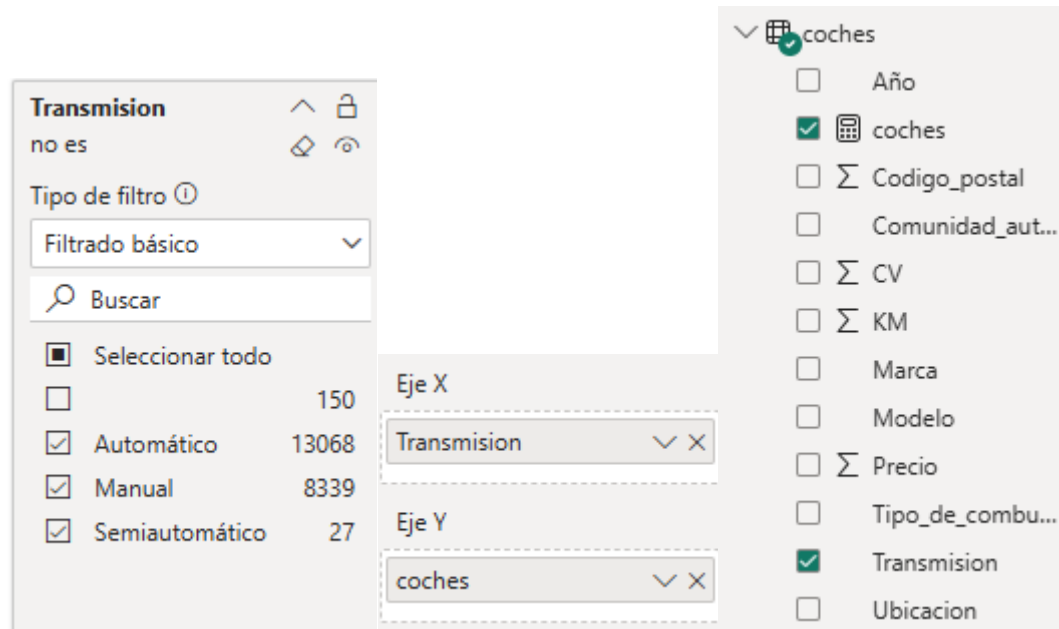
Resultado final:

coches por Tipo_de_combustible



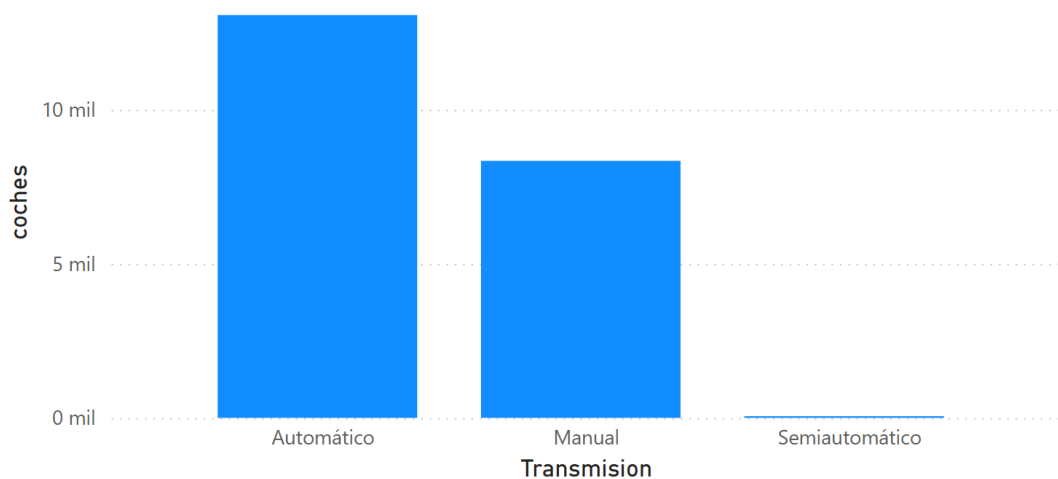
2. Gráfico de barras de la cantidad de coches por tipo de transmisión.

Para este grafico necesitaremos la cantidad coches por transmisión y como en el gráfico anterior quitaremos los datos de los coches en los que no aparezca la transmisión.



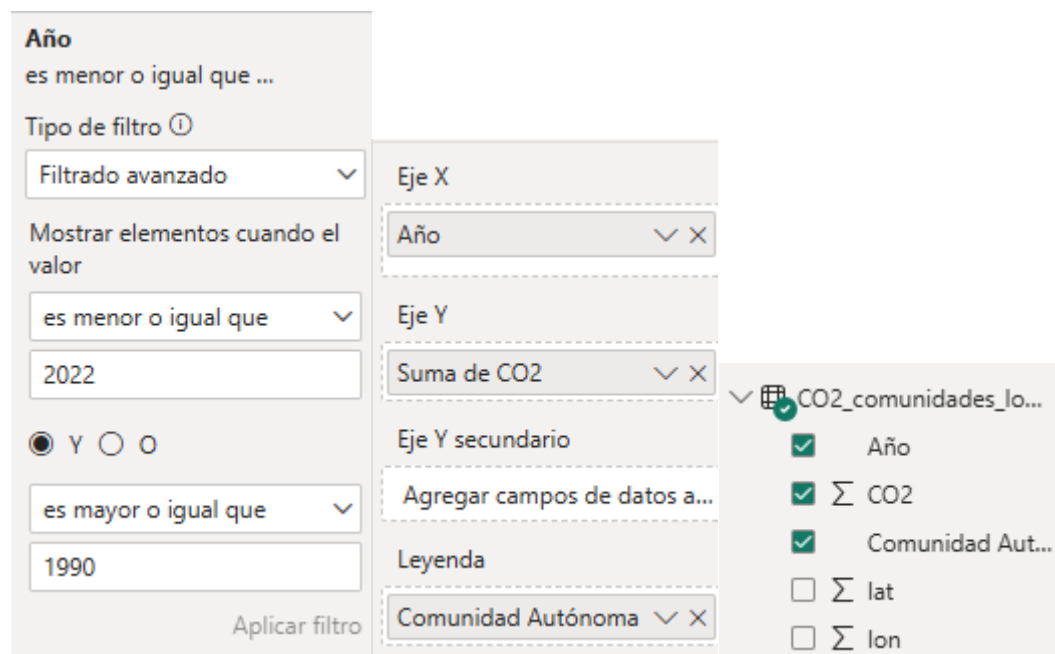
Resultado final:

coches por Transmision



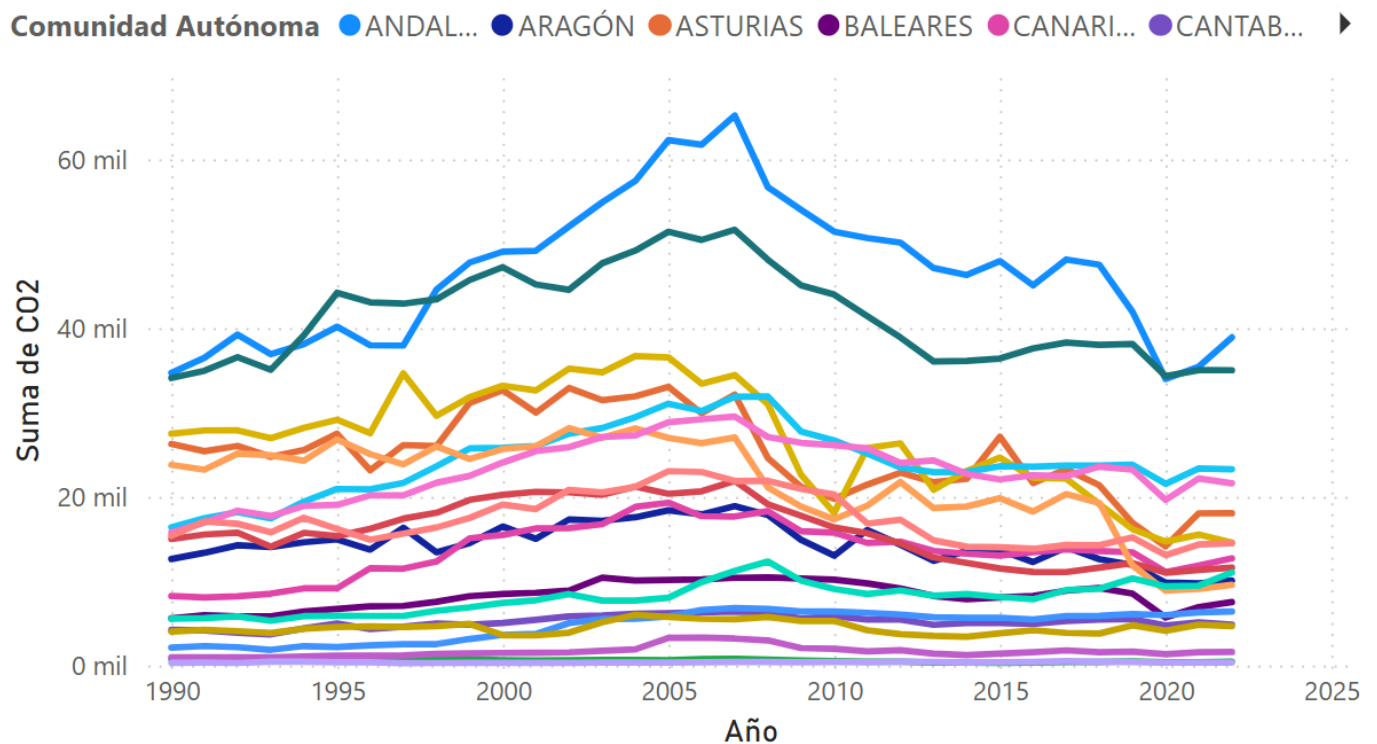
3. Gráfico de líneas de la suma de Co2 por año y por comunidad autónoma.

Para este gráfico seleccionamos el gráfico de líneas en el menú después en los datos del json de Co2 por comunidad autónoma escogemos el Año la suma de Co2 y la comunidad autónoma.



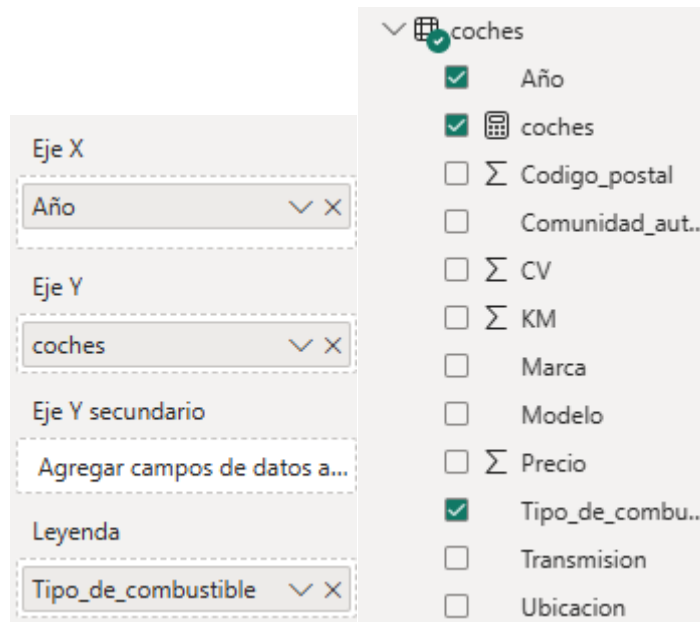
Resultado final:

Suma de CO2 por Año y Comunidad Autónoma



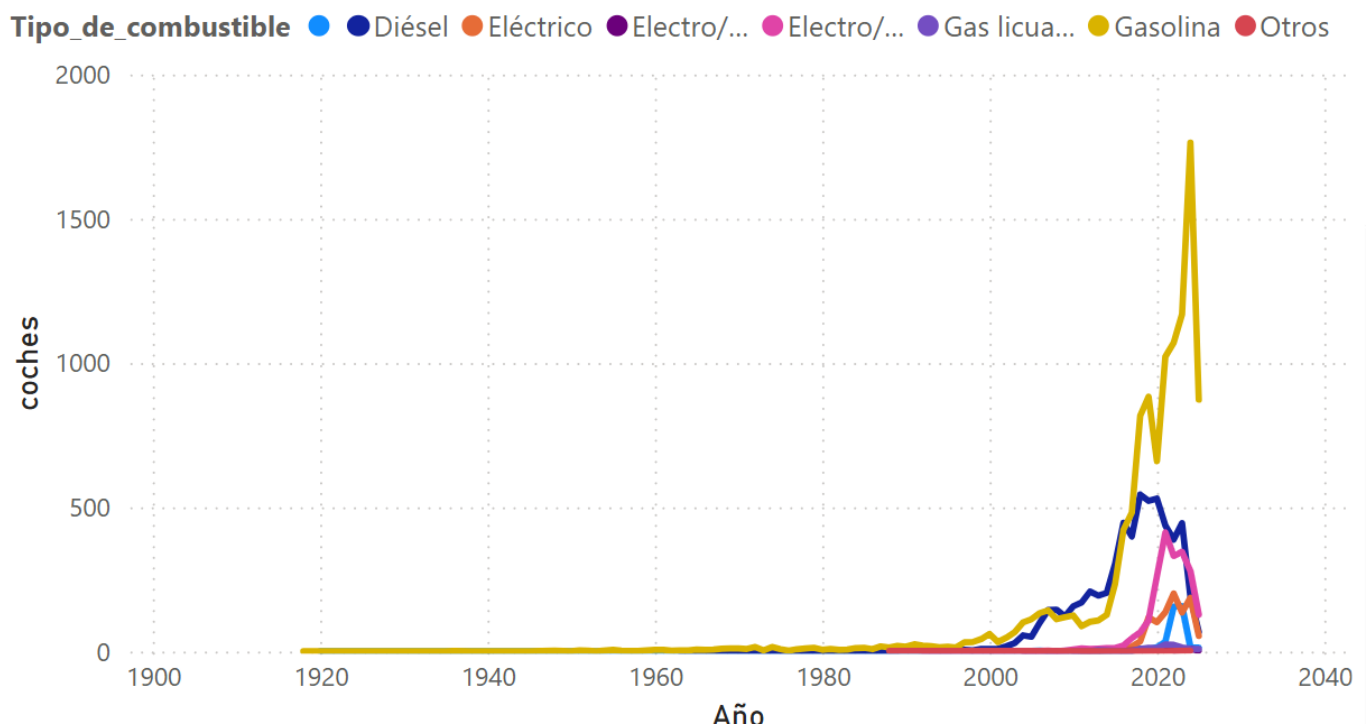
4. Gráfico de líneas de la cantidad de coches por año y tipo de combustible.

Para representar este gráfico necesitaremos de nuestro csv el año el recuento de los coches y el tipo de combustible.



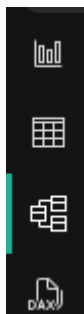
Resultado final:

coches por Año y Tipo_de_combustible

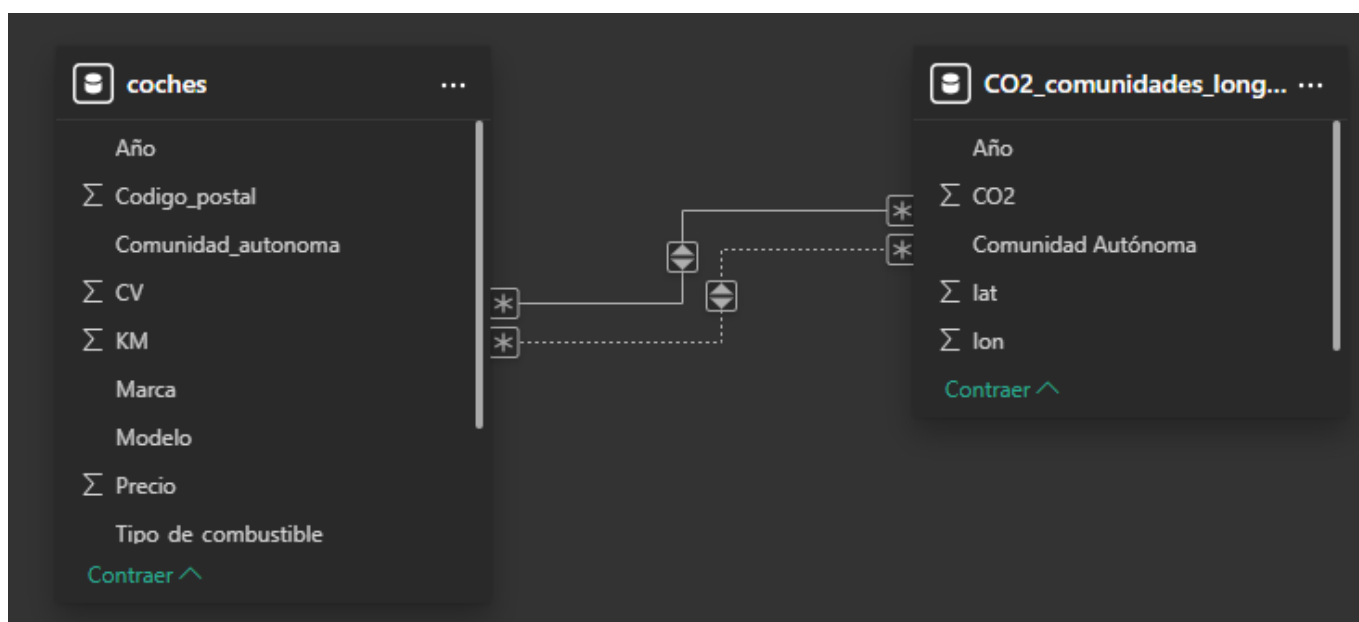


3.3. Tercer informe (relación de los datos)

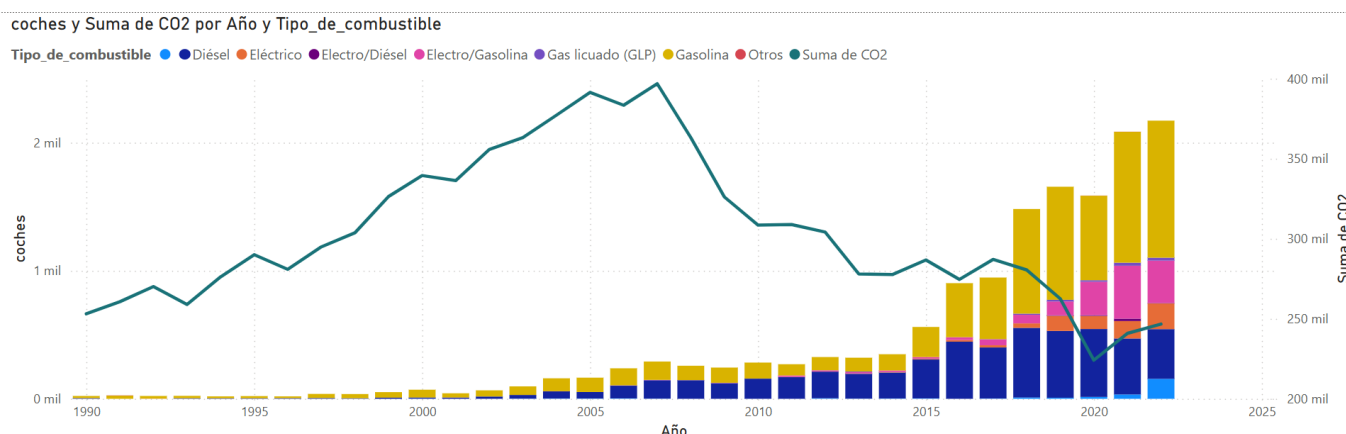
En este informe tengo dos gráficos en unos de ellos es necesario hacer una relación entre los años y/o la comunidad autónoma de la tabla de coches y la de emisiones para ello tendremos que ir a la barra de herramientas de la derecha en la que encontramos este icono:



Aquí podremos hacer relaciones entre las tablas arrastrando los campos que queramos relacionar.

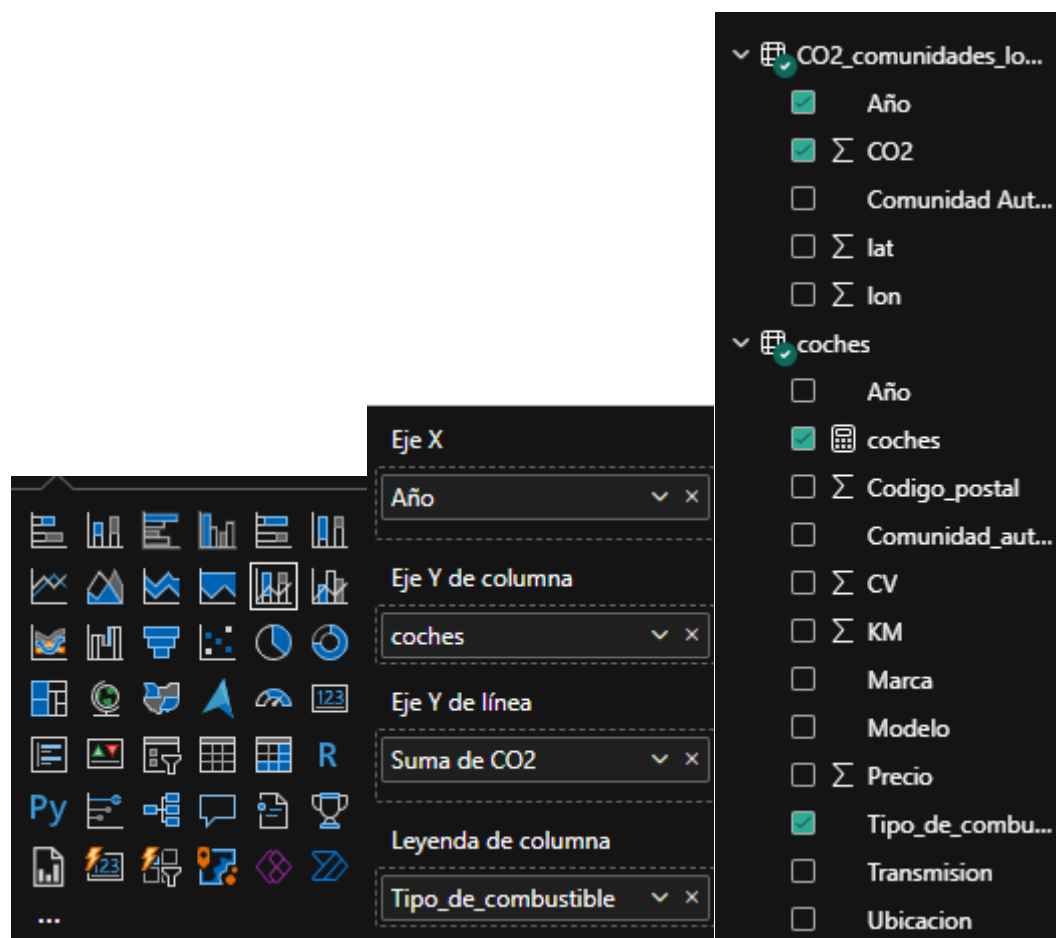


Una vez hecho esto podremos realizar el siguiente gráfico.



En este gráfico podemos apreciar en las barras la cantidad de coche por año y en diferente color el tipo de combustible al que pertenece y la línea correspondería a la evolución de las emisiones de Co2 a lo largo de los años.

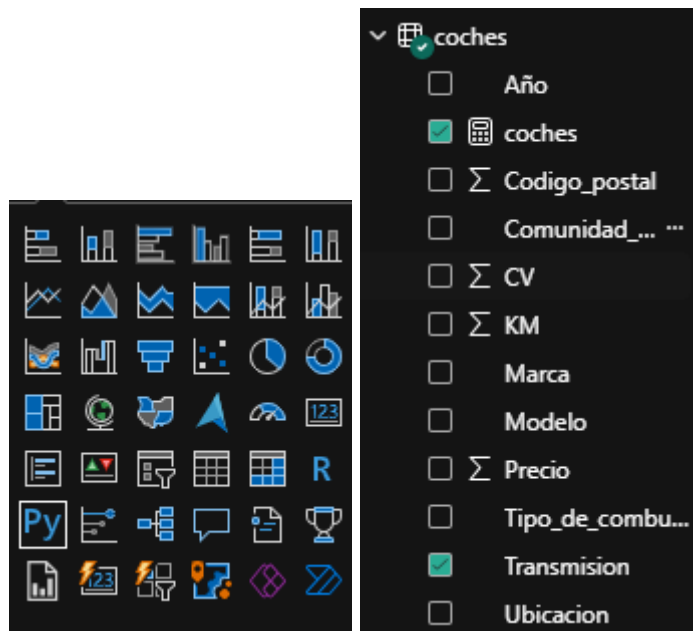
Para ello en el eje X he puesto el año, en el eje Y de las columnas la cantidad de coches, en el Y de la línea el Co2 y en la leyenda de la columna el tipo de combustible.



3.4. Cuarto informe (Objeto visual de Python)

En este informe he puesto otra vez la gráfica de la cantidad de coches por tipo de transmisión, pero hecho con un objeto visual de Python.

Para ello he vuelto a usar la cantidad de coches y los tipos de transmisión.

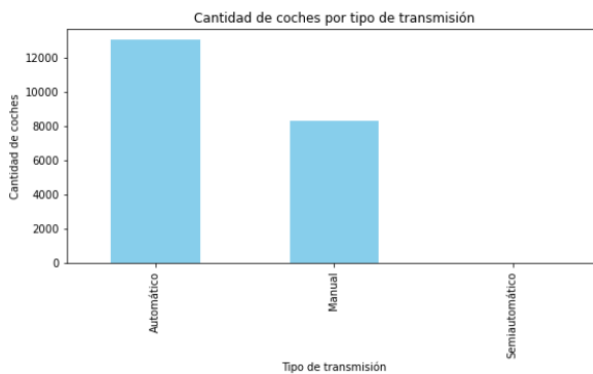


Y para la parte del código he puesto lo siguiente:

```
import matplotlib.pyplot as plt
import pandas as pd

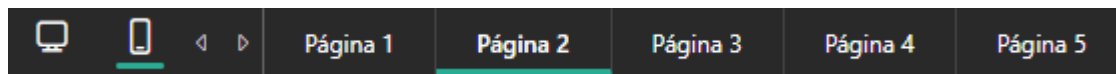
cochesxtransmision = dataset.groupby('Transmision')['coches'].sum()
cochesxtransmision.plot(kind='bar', color='skyblue', figsize=(8, 5))
plt.title("Cantidad de coches por tipo de transmisión")
plt.xlabel("Tipo de transmisión")
plt.ylabel("Cantidad de coches")
plt.tight_layout()
plt.show()
```

coches y Transmision

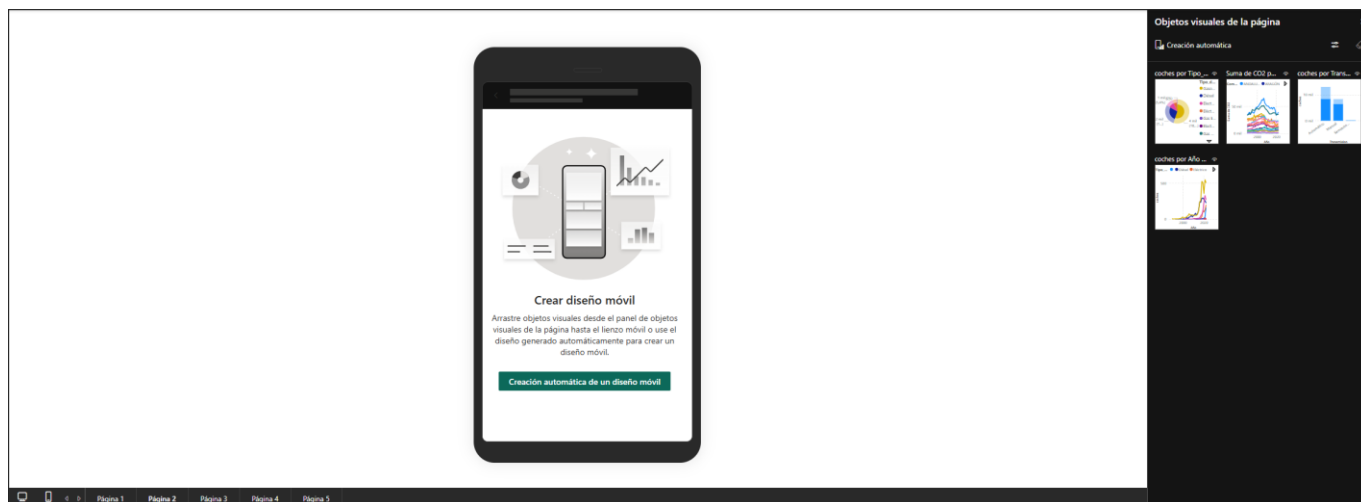


3.5. Vista de móviles para los informes

Para poder visualizar los informes en la vista para dispositivos móviles tendremos que hacer clic en el icono de abajo a la derecha con la forma de este.

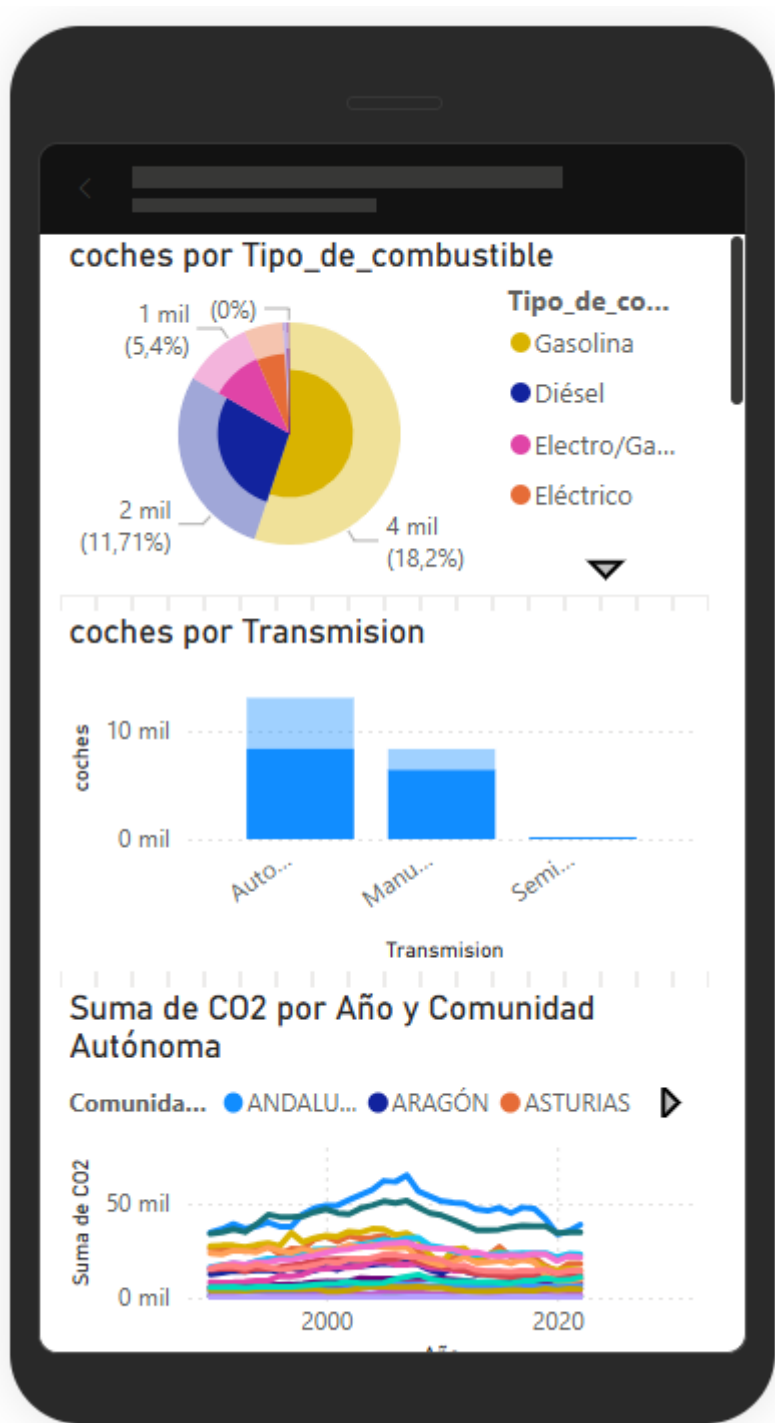


Al darle clic podremos ver la vista de móviles vacía.



Como vemos en el centro está el teléfono, si queremos hacerlo de forma sencilla y rápida podemos darle a botón verde de “creación automática de un diseño móvil” esto pondrá de manera dinámica todos los gráficos que tenemos en nuestro informe (en el panel de la derecha), si queremos colocar nosotros los gráficos podremos hacerlo al arrastrar el que queramos de dicho panel al móvil.

Como resultado podremos ver los gráficos en este formato.



4. Spark

Para empezar, tendremos que conectarnos al cesga mediante ssh.

```
PS C:\Users\ralph> ssh xuedua093@hadoop.cesga.es
xuedua093@hadoop.cesga.es's password:
Last login: Wed May 21 20:04:17 2025 from 10.121.254.3
*****
*                               BIENVENIDO A LA PLATAFORMA HADOOP3                               *
*                                                                                               *
* Financiada por el Ministerio de Economía y Competitividad,                               *
* la Xunta de Galicia y FEDER (Fondo Europeo de Desarrollo Regional)                       *
*****
* En caso de problemas/dudas: telefono ---> 981 56 98 10, ext. 814                               *
*                               web      ---> https://bigdata.cesga.es                               *
*                               e-mail   ---> helpdesk_bigdata@cesga.es                               *
*                                                                                               *
* Documentacion de la plataforma:                                                             *
*   - Guia de usuario:  https://bigdata.cesga.es/user-guide/                               *
*   - Tutoriales:      http://bigdata.cesga.es/#tutorials                               *
*                                                                                               *
*****

Envio de un trabajo Spark:
spark-submit --driver-memory 4g --executor-memory 2g --num-executors 4 run.py

Ver logs de un trabajo:
yarn logs -applicationId application_1554218728255_0007 | less

Ver informacion sistema de colas (YARN):
yarn top

[xuedua093@cdh61-login1 ~]$
```

Ponemos “**start_jupyter-lab**” y aquí podremos editar nuestro notebook para hacer consulta sobre nuestro csv que enviaremos a continuación

```
[xuedua093@cdh61-login5 ~]$ start_jupyter-lab
[I 16:59:15.423 LabApp] Writing notebook server cookie secret to /run/user/46193/jupyter/notebook_cookie_secret
[I 16:59:15.903 LabApp] JupyterLab extension loaded from /opt/cesga/anaconda/Anaconda2-2018.12/lib/python2.7/site-packages/jupyterlab
[I 16:59:15.903 LabApp] JupyterLab application directory is /opt/cesga/anaconda/Anaconda2-2018.12/share/jupyter/lab
[W 16:59:15.909 LabApp] JupyterLab server extension not enabled, manually loading...
[I 16:59:15.910 LabApp] JupyterLab extension loaded from /opt/cesga/anaconda/Anaconda2-2018.12/lib/python2.7/site-packages/jupyterlab
[I 16:59:15.910 LabApp] JupyterLab application directory is /opt/cesga/anaconda/Anaconda2-2018.12/share/jupyter/lab
[I 16:59:15.918 LabApp] Serving notebooks from local directory: /home/xunta/edu/a093/notebook
[I 16:59:15.918 LabApp] The Jupyter Notebook is running at:
[I 16:59:15.918 LabApp] http://10.121.242.35:8888/?token=cecdf98b892101a6921e95b6ddc29d375c4f21bc3ba1b1ea
[I 16:59:15.918 LabApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:59:15.922 LabApp]

To access the notebook, open this file in a browser:
file:///run/user/46193/jupyter/nbserver-15671-open.html
Or copy and paste one of these URLs:
http://10.121.242.35:8888/?token=cecdf98b892101a6921e95b6ddc29d375c4f21bc3ba1b1ea
[I 16:59:22.397 LabApp] 302 GET /?token=cecdf98b892101a6921e95b6ddc29d375c4f21bc3ba1b1ea (172.16.10.28) 0.67ms
```

Para enviar el csv los haremos con SCP de la siguiente manera.

```
scp -r -p "C:\Users\ralphynunezmercado\Documents\Directorio\BigData\T5.1_Panel de control con PowerBI\coches.csv" xuedua093@hadoop.cesga.es:/home/xunta/edu/a093
```

Después tendremos que mandar el archivo con put a nuestro hdfs

```
[xuedua093@cdh61-login1 ~]$ hdfs dfs -put coches.csv
[xuedua093@cdh61-login1 ~]$ |
```

En jupyter notebook podemos descargar el archivo como .py y ejecutarlo en la consola o podemos ejecutar el notebook directamente allí.

```
spark-submit --driver-memory 4g --executor-memory 2g --num-executors 4 graficas_spark.py
```

Por consola:

```
[xuedua093@cdh61-login1 ~]$ hdfs dfs -put coches.csv
[xuedua093@cdh61-login1 ~]$ export PYTHONIOENCODING=utf8
[xuedua093@cdh61-login1 ~]$ spark-submit --driver-memory 4g --executor-memory 2g --num-executors 4 graficas_spark.py
Coches por comunidad autonoma
```

Comunidad_autonoma	count
Madrid	5446
Andalucía	3549
Cataluña	3349
Comunidad Valenciana	2676
Galicia	1034
País Vasco	1001
Región de Murcia	915
Castilla-La Mancha	618
Aragón	492
Castilla y León	480
Asturias	429
Navarra	348
Canarias	320
Extremadura	285
Cantabria	230
Baleares	204
La Rioja	175
Desconocida	29
Ceuta	3
Melilla	1

```
Coches por transmision
```

Transmision	count
Automático	13068
null	150
Manual	8339
Semiautomático	27

Coche con más kilometraje

Marca	Modelo	KM
Mercedes-benz	sl 280	9999999.0

only showing top 1 row

Precio medio por marca

Marca	precio_medio
Pagani	5800000.0
Ferrari	461820.03
Rolls-royce	455771.26
McLaren	400837.48
Lamborghini	394570.92
Maybach	307933.33
Aston martin	256495.19
Bentley	216352.65
Wiesmann	189850.0
Ram	116642.38
Alpina	115380.0
Lotus	105701.24
Ac	103680.0
De tomaso	103300.0
Gmc	100549.47
Cadillac	94345.08
Ineos	89944.63
Voyah	86870.0
Ktm	86366.67
Morgan	84406.47

only showing top 20 rows

[xuedua093@cdh61-login1 ~]\$ |

Por el notebook:

```
<class 'pyspark.sql.dataframe.DataFrame'>
```

Coches por comunidad autonoma

Comunidad_autonoma	count
Madrid	5446
Andalucía	3549
Cataluña	3349
Comunidad Valenciana	2676
Galicia	1034
País Vasco	1001
Región de Murcia	915
Castilla-La Mancha	618
Aragón	492
Castilla y León	480
Asturias	429
Navarra	348
Canarias	320
Extremadura	285
Cantabria	230
Baleares	204
La Rioja	175
Desconocida	29
Ceuta	3
Melilla	1

Coches por transmision

Transmision	count
Automático	13068
null	150
Manual	8339
Semiautomático	27

Coche con más kilometraje

Marca	Modelo	KM
Mercedes-benz	sl 280	99999999.0

only showing top 1 row

Precio medio por marca

Marca	precio_medio
Pagani	5800000.0
Ferrari	461820.03
Rolls-royce	455771.26
McLaren	400837.48
Lamborghini	394570.92
Maybach	307933.33
Aston martin	256495.19
Bentley	216352.65
Wiesmann	189850.0
Ram	116642.38
Alpina	115380.0
Lotus	105701.24
Ac	103680.0
De tomaso	103300.0
Gmc	100549.47
Cadillac	94345.08
Ineos	89944.63
Voyah	86870.0
Ktm	86366.67
Morgan	84406.47

only showing top 20 rows

5. Conclusiones

He llegado a la conclusión de que las emisiones se reducen a partir el año 2007, esto seguramente sea más por las leyes europeas hacia las industrias que por los coches, ya que en el dataset tengo más coches en el tramo de 2015 – 2022, también hay que decir que hubo la pandemia por el COVID-19 entre el 2019 y el 2022 en donde se empiezan a rebajar las restricciones, como resultado las emisiones también habrán bajado por eso.

Aun con lo anterior dicho, es verdad que se aprecia un aumento en la venta de coches eléctricos y sobre todo de coches híbridos Eléctrico/Gasolina.

Sumando todos los coches con combustibles de bajas emisiones podemos ver cómo hay un aumento significativo en los siguientes años, aunque también hay un repunte del resto de combustibles que como he dicho anteriormente puede ser por los datos del dataset ya que tengo más datos en los años recientes.

Además, empieza a haber un repunte en las emisiones que puede ser por la vuelta a la normalidad después de la pandemia.

También hay que decir que según avanza la tecnología y con la imposición de leyes de bajas emisiones en los coches de combustible fósil, se fabrican coches que emiten menos Co2 a la atmósfera. Leyes como la de las zonas de bajas emisiones favorecen esto y por tanto aumenta la venta de coches eléctricos/híbridos o coches que emitan menos emisiones.

