

Enlace al repositorio GitHub con la memoria y los archivos relativos a la práctica:

[https://github.com/CanoLuis/Practica\\_Spark\\_22160810](https://github.com/CanoLuis/Practica_Spark_22160810)

## **Memoria descriptiva**

Para la realización de esta práctica se utilizó una conexión a Spark (spark\_connect) para el programa RStudio, así como las librerías sparklyr, dplyr, tidyverse, stringr, y readxl.

En primer lugar se recurrió a la página del ministerio

<https://sedeaplicaciones.minetur.gob.es/ServiciosRESTCarburantes/PreciosCarburantes/EstacionesTerrestres/>, que fue asociada a una variable (url) y con la que se trabajó durante la práctica

### **Apartado a)**

Al estar trabajando con la información de una fuente externa, se recurrió a la librería jsonlite y, a partir de ahí, a comandos como *clean\_names*.

Además, se eliminaron columnas que no aportaban valor (margen, remisión) y se renombraron los nombres de algunas columnas, como Código Postal, para un mayor entendimiento.

- Para crear una nueva columna se recurrió a la orden *mutate()*, de manera que se identificase si en el nombre de las estaciones de servicio se encontraba el nombre de alguna de las grandes compañías petrolíferas, pudiendo así discernir entre estas, que serían las no low-cost, y el resto, que serían las catalogadas como low-cost.
- Como esto da como resultado TRUE o FALSE, se ejecutó *ifelse* de forma que, para cuando apareciera TRUE se mostrara un low-cost, y viceversa.  
*dataset\$low\_cost = ifelse(dataset\$low\_cost == TRUE, "Gasolinera low-cost", "Gasolinera no low-cost")*
- Para calcular el precio promedio, se seleccionó la columna de precio de cada combustible presente en el dataset para una vez agrupados por comunidades autónomas (*group\_by(idccaa)*) recoger (*summarise*) los precios medios de estas columnas mediante el comando *mean()*. Se hizo el mismo proceso por provincias (*group\_by(id\_provincia)*).
- Para la impresión de los mapas pedidos, se realizó (todo junto con el operador pipe *%>%*) una selección de las columnas de interés y del top elegido (las primeras 10 estaciones de servicio o las 20 últimas), para posteriormente, gracias al paquete leaflet, remarcar (*addCircleMarkers*) las estaciones de servicio mencionadas.

A continuación, se puede apreciar una vista de dichos mapas.

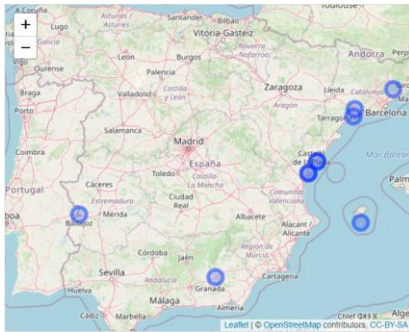


Ilustración 1. Top 10 gasolineras más caras. Precio gasóleo A

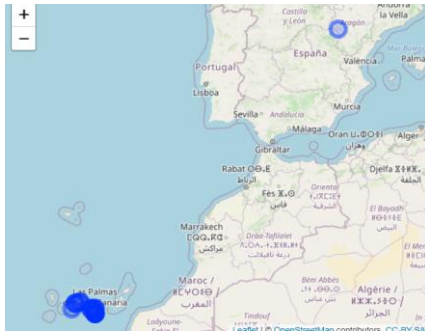


Ilustración 2. Top 20 gasolineras más baratas. Precio gasóleo A

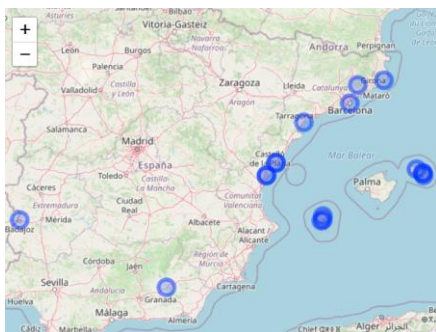


Ilustración 3. Top 10 gasolineras más caras. Precio gasolina 95.

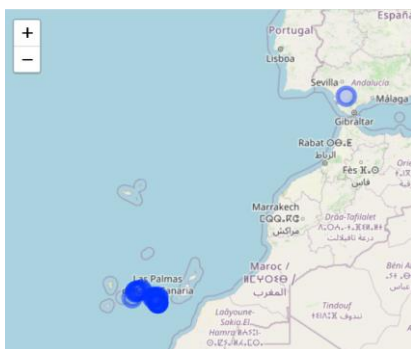


Ilustración 4. Top 20 gasolineras más baratas. Precio gasolina 95

Estos apartados, así como los posteriores, se guardaron mediante a orden `write.csv`

Apartado b)

- Para este apartado se le asignó a una variable una selección de las columnas `idccaa`, `low-cost` y provincia del dataset para después, mediante un filter obtener las estaciones de servicio de esa comunidad y mediante un count, saber la cantidad de cuántas son low-cost. El código generado para esto fue tal que: `%>% filter(idccaa=="13") %>% count(low_cost)`. Este proceso se realizó dos veces, una para la Comunidad de Madrid y otra para Cataluña.

Tabla 1. Proporción de gasolineras en la Comunidad de Madrid.

Gasolineras low-cost	474
Gasolineras no low-cost	318

Tabla 2. Proporción de gasolineras en Cataluña.

Gasolineras low-cost	623
Gasolineras no low-cost	827

- De forma similar, para saber los precios de los combustibles, se seleccionaron las variables de interés y se filtró por comunidad gracias al `idccaa`. Se eliminaron los posibles valores nulos (`drop_na`) para finalmente recoger los valores medios, máximos y mínimos de los combustibles mediante los comandos `mean()`, `max()` y `min()` respectivamente. Este proceso se llevó a cabo tanto para la Comunidad de Madrid como para Cataluña.

Tabla 3. Datos de precios de combustibles. Comunidad de Madrid.

Gasóleo A. Precio Máximo	Gasóleo A. Precio Mínimo	Gasóleo A. Precio Medio	Gasolina 95. Precio Máximo	Gasolina 95. Precio Mínimo	Gasolina 95. Precio Medio
1.459	1.339	1.427892	1.689	1.519	1.615378

Tabla 4. Datos de precios de combustibles. Cataluña.

Gasóleo A. Precio Máximo	Gasóleo A. Precio Mínimo	Gasóleo A. Precio Medio	Gasolina 95. Precio Máximo	Gasolina 95. Precio Mínimo	Gasolina 95. Precio Medio
1.469	1.189	1.38748	1.715	1.369	1.571267

### Apartado c

Para este apartado en primer lugar se filtraron los datos de forma que se quitaron los registros correspondientes a las ciudades que se debían excluir del análisis

```
%>% filter(!municipio %in% c("Madrid", "Barcelona", "Sevilla", "Valencia")) %>%
```

A partir de ahí, se contaron los registros resultantes, y por tanto, las estaciones de servicio que se deseaban saber, mediante un `nrow()`

Tabla 5. Proporción de gasolineras omitiendo grandes ciudades.

Low-Cost	2168
No Low-Cost	2471

### Apartado d

De forma similar al apartado c, en esta parte de la práctica lo que se hizo fue filtrar a partir de la variable horario para saber qué estaciones de servicio tendrían horario de apertura 24 horas

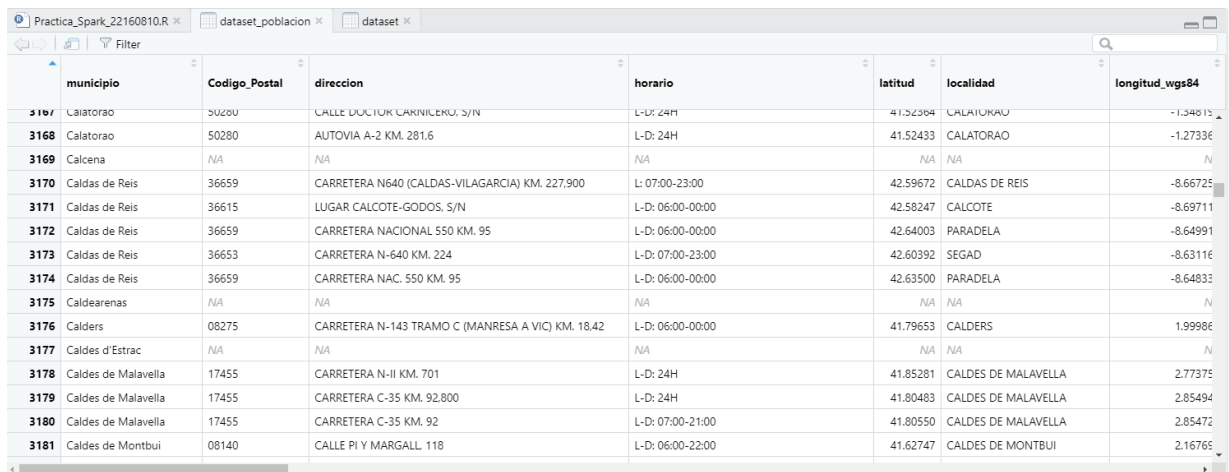
```
filter(dataset$horario == "L-D: 24H")
```

### Apartado e

- Para la realización de este apartado se recurrió a la fuente <https://www.ine.es/dynt3/inebase/es/index.htm?padre=517&capsel=525>, de donde se extrajo la información de número de habitantes por municipios. Esta información se almacena en una variable (población). También se renombraron ciertas variables y se eliminaron algunas columnas de cara a la limpieza de los datos.

Con todo esto se tiene un dataset para la población que, junto con el ya realizado de los municipios con estaciones de servicio y demás, dan pie a la realización de un merge por el cual se concluirá en una tabla con los datos de municipios, estaciones de servicio etc. y, además, la población de la fuente consultada

A continuación se puede apreciar una imagen de muestra del resultado de unir los datos de estaciones de servicio con los datos de municipios:



	municipio	Codigo_Postal	direccion	horario	latitud	localidad	longitud_wgs84
3167	Calatonia	50200	CALLE DUC TOR CARNICERO, 3/N	L-D: 24H	41.52304	CALATORAO	-1.34915
3168	Calatonia	50280	AUTOVIA A-2 KM. 281.6	L-D: 24H	41.52433	CALATORAO	-1.27336
3169	Calatonia	N/A	N/A	N/A	N/A	N/A	N
3170	Caldas de Reis	36659	CARRETERA N640 (CALDAS-VILAGARCIA) KM. 227.900	L: 07:00-23:00	42.59672	CALDAS DE REIS	-8.66725
3171	Caldas de Reis	36615	LUGAR CALCOTE-GODOS. S/N	L-D: 06:00-00:00	42.58247	CALCOTE	-8.69711
3172	Caldas de Reis	36659	CARRETERA NACIONAL 550 KM. 95	L-D: 06:00-00:00	42.64003	PARADELA	-8.64991
3173	Caldas de Reis	36653	CARRETERA N-640 KM. 224	L-D: 07:00-23:00	42.60392	SEGAD	-8.63116
3174	Caldas de Reis	36659	CARRETERA NAC. 550 KM. 95	L-D: 06:00-00:00	42.63500	PARADELA	-8.64833
3175	Caldearenas	N/A	N/A	N/A	N/A	N/A	N
3176	Calders	08275	CARRETERA N-143 TRAMO C (MANRESA A VIC) KM. 18.42	L-D: 06:00-00:00	41.79653	CALDEERS	1.99986
3177	Caldes d'Estrac	N/A	N/A	N/A	N/A	N/A	N
3178	Caldes de Malavella	17455	CARRETERA N-II KM. 701	L-D: 24H	41.85281	CALDES DE MALAVELLA	2.77375
3179	Caldes de Malavella	17455	CARRETERA C-35 KM. 92.800	L-D: 24H	41.80483	CALDES DE MALAVELLA	2.85494
3180	Caldes de Malavella	17455	CARRETERA C-35 KM. 92	L-D: 07:00-21:00	41.80550	CALDES DE MALAVELLA	2.85472
3181	Caldes de Montbui	08140	CALLE PI Y MARGALL 118	L-D: 06:00-22:00	41.62747	CALDES DE MONTBUI	2.16765

Ilustración 5. Dataset de estaciones de servicio + población.

- Después, para el siguiente punto, en que se han de trazar los mapas para diferentes radios desde las estaciones de servicio, se recurrió, a través de la librería leaflet, a cambiar el radio con que se marcan las estaciones de servicio. Para ello, se utiliza el parámetro *radius* dentro del *AddCircles*.  
*dataset %>% select(rotulo, latitud, longitud\_wgs84, precio\_gasolina\_98\_e5, localidad, direccion, ideess) %>% leaflet() %>% addTiles() %>% addCircles(lng = ~longitud\_wgs84, lat = ~latitud, popup = ~rotulo, label = ~ideess, radius = 4)*
- Por último, para este apartado, en que se pide un top de municipios en que no haya gasolineras abiertas 24 horas, se recurrió a filtrar el dataset (una vez seleccionadas las variables deseadas) por la variable horario para después agruparlas en función de que sean low-cost. Con todo esto, ya se está en disposición de elegir las 10 gasolineras deseadas mediante el comando *top* → *top\_n(-10, precio\_gasolina\_95\_e5)* (Nota: se escogió como parámetro la gasolina 95e5)

## Otros resultados

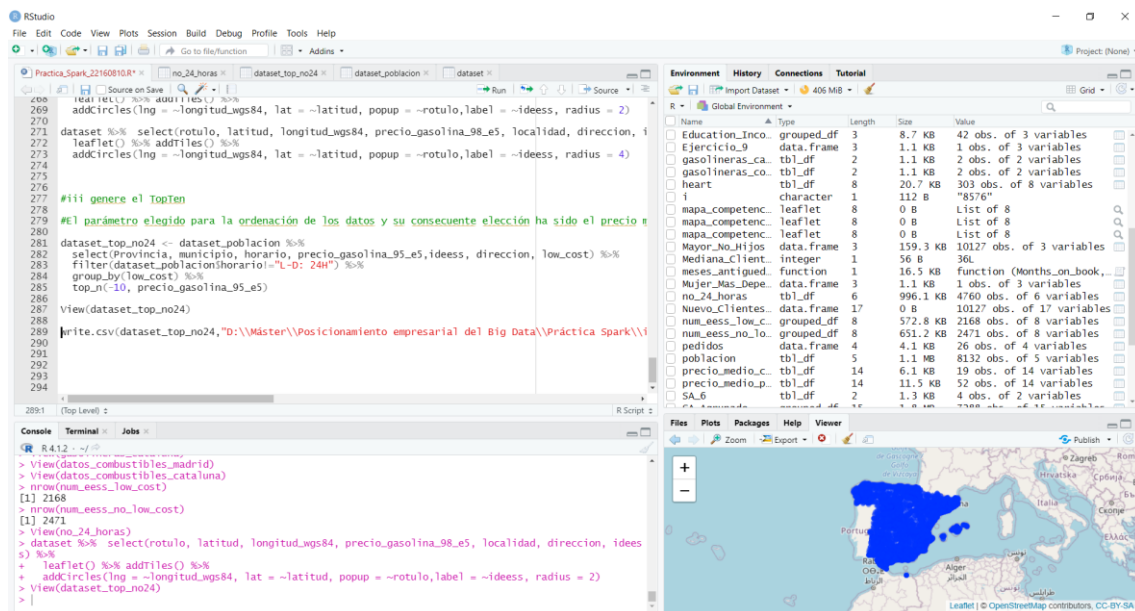


Ilustración 63. Ventana de Spark de trabajo.