

### Acciones previas necesarias antes del inicio del ejercicio en Spark

- Procedo a instalar o bien “llamar” a los paquetes necesarios para la realización del ejercicio, entre ellos (dplyr, httr, tidyverse, leaflet, janitor, readr, sparklyr, XML..)
- Me conecto a spark a través del siguiente código:
  - o `sc <- spark_connect(master = "local")`
- Obtengo los datos de fuentes veraces a través de [geo portal gasolineras](#)

A. De fuentes veraces, lea los archivos que se indican en el anexo, como podrá apreciar el/los archivos contienen miles de filas por decenas de columnas; solo es posible tratarlos utilizando Spark si queremos respuestas en tiempo real;

A. I y II. Anomalías durante la limpia del dataset.

Para proceder a limpiar el dataset, se ha ejecutado el siguiente código, el mismo utilizado en clase:

```
ds <- jsonlite::fromJSON(url)
```

```
ds <- ds$ListaEESSPrecio
```

```
ds <- ds %>% as_tibble() %>% clean_names()
```

```
ds <- ds %>% type_convert(locale = locale(decimal_mark = ",")) %>% view() %>%  
clean_names()
```

Con ello, lo que hacemos convertir el dataframe a clase tbl y limpia en el dataset los nombres, espacios y caracteres especiales que pueda contener. Una vez realizado esto, ejecutamos type\_covert para convertir los objetos en su formato adecuado, y como volvemos a leer los datos, se procede a llamar a la función clean\_names para que vuelva a limpiar los datos.

Durante el proceso de limpia del dataset, llama la atención una serie de anomalías que ocurrieron, como por ejemplo que al usar type\_covert para convertir los puntos por las comas, al ejecutarlo, no hace el reemplazo, pero aún así permite operar con los datos. Otra anomalía llamativa fue que al estar realizando el apartado AIII, al calcular las medias para todos los precios promedios, devolvía todos los valores como NAs, y esto se consiguió arreglar haciendo en dos partes la limpieza del dataset.

A. III. Cree una columna nueva que deberá llamarse low-cost, y determine cuál es el precio promedio de todos los combustibles a nivel comunidades autónomas, así como para las provincias, tanto para el territorio peninsular e insular, esta columna deberá clasificar las estaciones por low-cost y no-low-cost.

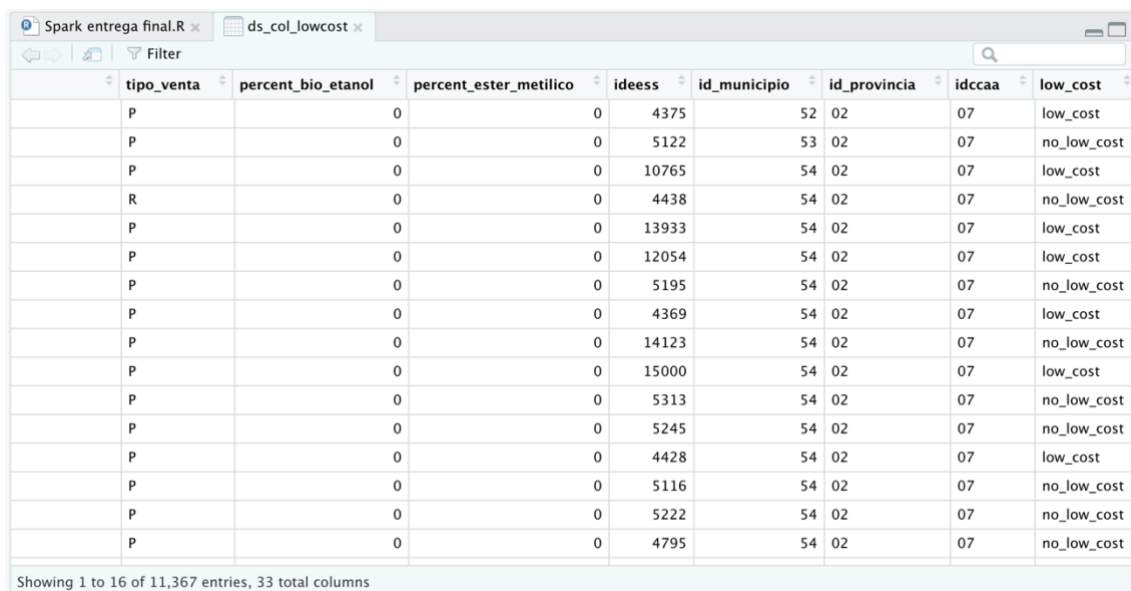
A través del siguiente código, se crea la columna low\_cost que clasifica las gasolineras según su nombre en el rótulo y devuelve un “True” para las que son no\_low\_cost, y “False” para las que son low\_cost

```
ds_col_lowcost <- ds
%>% mutate(low_cost=rotulo%in%c("REPSOL","CAMPSA","BP","SHELL","GALP",
"CEPSA")) %>% view()
```

Ahora, como aprendimos en el ModII de R, se reemplaza los valores True/False por su nombre correspondiente.

```
ds_col_lowcost$low_cost[ds_col_lowcost$low_cost == TRUE] <- "no_low_cost"
```

```
ds_col_lowcost$low_cost[ds_col_lowcost$low_cost == FALSE] <- "low_cost"
```



	tipo_venta	percent_bio_etanol	percent_ester_metilico	ideess	id_municipio	id_provincia	idccaa	low_cost
	P	0	0	4375	52	02	07	low_cost
	P	0	0	5122	53	02	07	no_low_cost
	P	0	0	10765	54	02	07	low_cost
	R	0	0	4438	54	02	07	no_low_cost
	P	0	0	13933	54	02	07	low_cost
	P	0	0	12054	54	02	07	low_cost
	P	0	0	5195	54	02	07	no_low_cost
	P	0	0	4369	54	02	07	low_cost
	P	0	0	14123	54	02	07	no_low_cost
	P	0	0	15000	54	02	07	low_cost
	P	0	0	5313	54	02	07	no_low_cost
	P	0	0	5245	54	02	07	no_low_cost
	P	0	0	4428	54	02	07	low_cost
	P	0	0	5116	54	02	07	no_low_cost
	P	0	0	5222	54	02	07	no_low_cost
	P	0	0	4795	54	02	07	no_low_cost

Showing 1 to 16 of 11,367 entries, 33 total columns

Imagen 1. Creación de la columna low\_cost con su correspondiente clasificación

- Calcular el precio promedio de todos los combustibles a nivel comunidades autónomas tanto para el territorio peninsular e insular.

Para poder realizar este apartado, se han seleccionado todas las columnas de los precios de los combustibles, así como la provincia, rotulo e id de cada comunidad autónoma, agrupado por comunidad autónoma y provincia, y se calcula la media de todos los precios de los combustibles omitiendo los valores NAs con na.rm=T.

```
#calcular la media del precio de todos los combustibles
mean_precios <- ds_col_lowcost %>% select(precio_bioetanol, precio_biodiesel, precio_gas_natural_comprimido, precio_g
  group_by(idccaa, provincia) %>% summarise(media_precio_bioetanol=mean(precio_bioetanol, na.rm=T), media_precio_bio
View(mean_precios)
```

Imagen 2. Extracto parcial del código utilizado para el cálculo del precio promedio de los distintos tipos de combustibles clasificado por CCAA.

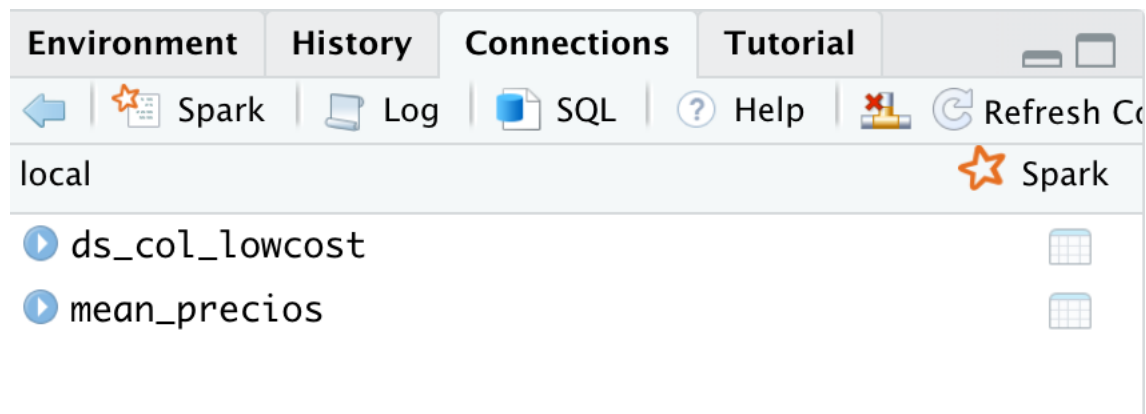


Imagen 3. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

A. IV. Imprima en un mapa interactivo, la localización del top 10 mas caras y otro mapa interactivo del top 20 mas baratas, estos 2 archivos deben guardarse en formato HTML y pdf para su posterior entrega al inversor., nombre de los archivos : top\_10.html, top\_10.pdf y top\_20.html, top\_20.pdf

Dado que en el enunciado no se especifica para que precio de combustible se debe realizar el análisis, he procedido a elegir el precio de la gasolina\_95\_e10.

Para el Top10 más caras, he utilizado el siguiente código para obtener el resultado:

```
top10_caras <- ds %>% select(rotulo, latitud, longitud_wgs84, precio_gasolina_95_e10,
  localidad, direccion) %>%
```

Alejandro Fernández Fraile (21614015)

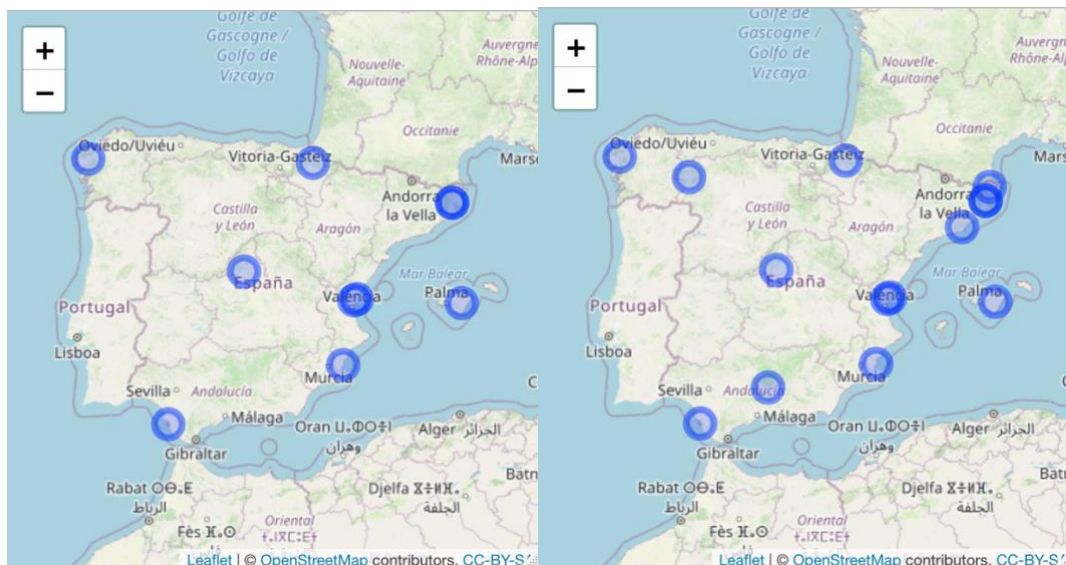
Módulo II: Posicionamiento Empresarial del Big Data

Práctica Spark

07/01/2022

```
top_n(10, precio_gasolina_95_e10) %>% leaflet() %>% addTiles() %>%  
addCircleMarkers(lng = ~longitud_wgs84, lat = ~latitud, popup = ~rotulo,label =  
~precio_gasolina_95_e10)
```

En el caso del Top20 más baratas, el código es el mismo con el único cambio de reemplazar 10 por -20 para obtener las 20 más baratas a nivel nacional.



Imágenes 4 y 5. Mapa Top10 gasolineras más baratas y Top20 gasolineras más caras..

Archivos de los mapas en [GitHub](#).

Archivos de los mapas en [G-Drive](#)

A. V. Conseguidos objetivos anteriores, debe guardar este “archivo” en una nueva tabla llamada low-cost\_num\_expediente y deberá estar disponible también en su repositorio de Github con el mismo nombre y formato csv.

Archivo tabla low\_cost en [GitHub](#).

Ambos archivos disponibles en [G-Drive](#)

- B. Este empresario tiene sus residencias habituales en Madrid y Barcelona , por lo que, en principio le gustaría implantarse en cualquiera de las dos antes citadas, y para ello quiere saber :

B. I. cuántas gasolineras tiene la comunidad de Madrid y en la comunidad de Cataluña, cuántas son low-cost, cuantas no lo son,

Para calcular el número de gasolineras en ambas comunidades autónomas y su correspondiente clasificación, se ha realizado seleccionando el id de comunidad autónoma y la clasificación de low\_cost, filtrando por las dos únicas comunidades autónomas que nos solicitan, agrupándolo por id de comunidad autónoma y haciendo un count del numero de gasolineras obtenidas según la clasificación.

```
Clasificación_lowcost_MAD_BCN <- ds_col_lowcost %>% select(idccaa, low_cost,
provincia) %>% filter(idccaa %in% c("13","09")) %>%
group_by(idccaa) %>% count(low_cost)
```

Número de Gasolineras en la Comunidad de Madrid(idccaa =13)

- Low-Cost: 317
- No Low-Cost: 474

Número de Gasolineras en Cataluña: (idccaa =09)

- Low-Cost: 827
- No Low-Cost: 653

	idccaa	low_cost	n
1	09	low_cost	827
2	09	no_low_cost	653
3	13	low_cost	317
4	13	no_low_cost	474

Imagen 6. Número de gasolineras por CCAA dividido entre low-cost y no low-cost.

B. II. además, necesita saber cuál es el precio promedio, el precio más bajo y el más caro de los siguientes carburantes: gasóleo A, y gasolina 95 e Premium.

Para calcular el precio promedio, el máximo y el mínimo, se ha obtenido el resultado con el siguiente código.

```
max_min_mean_MAD_BCN <- ds_col_lowcost %>% select(idccaa, low_cost,
provincia, precio_gasoleo_a, precio_gasolina_95_e5_premium) %>% drop_na() %>%
filter(idccaa %in% c("13","09")) %>%
group_by(idccaa, low_cost) %>%
summarise(max(precio_gasoleo_a), min(precio_gasoleo_a), mean(precio_gasoleo_a),
max(precio_gasolina_95_e5_premium), min(precio_gasolina_95_e5_premium),
mean(precio_gasolina_95_e5_premium))
```

Para ello, se han seleccionado los precios, la provincia, comunidad autónoma y su clasificación, filtrando por la comunidad de Madrid y Cataluña y calculando la media, máximo y mínimo.

	idccaa	low_cost	max(precio_gasoleo_a)	min(precio_gasoleo_a)	mean(precio_gasoleo_a)	max(precio_gasolina_95_e5_premium)	min(precio_gasolina_95_e5_premium)	mean(precio_gasolina_95_e5_premium)
1	09	low_cost	1.449	1.189	1.317850	1.659	1.369	1.461050
2	09	no_low_cost	1.469	1.325	1.415564	1.719	1.529	1.611309
3	13	low_cost	1.459	1.389	1.429000	1.659	1.519	1.602333
4	13	no_low_cost	1.465	1.345	1.429310	1.689	1.539	1.616845

Imagen 7. Precio medio, mas alto y bajo de Gasolina 95 E5 Premium y Gasóleo A en la Comunidad de Madrid y Cataluña

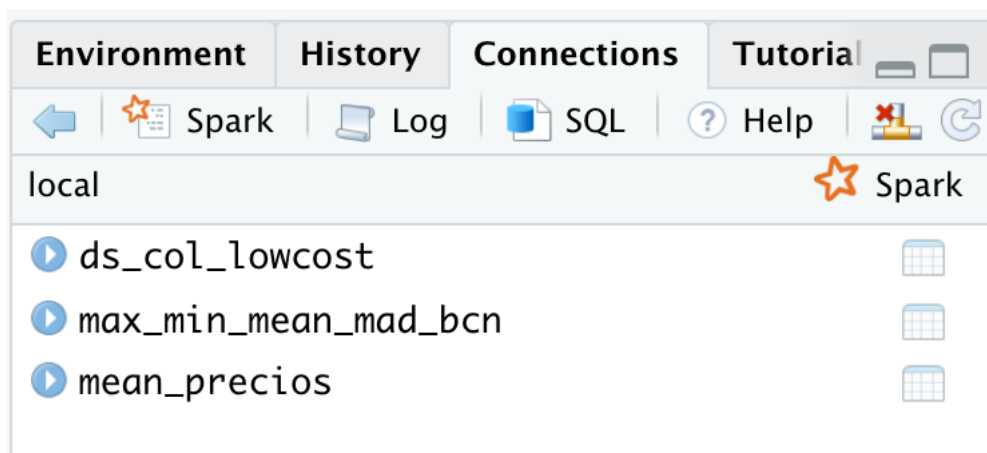


Imagen 8. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

- B. III. Conseguido el objetivo, deberá guardar este “archivo” en una nueva tabla llamada **informe\_MAD\_BCN\_expediente** y deberá estar disponible también en su repositorio con el mismo nombre en formato CSV

Archivo de la tabla informe\_MAD\_BCN en [GitHub](#).

Archivo disponible en [G-Drive](#)

- C. Por sí las comunidades de Madrid y Cataluña no se adapta a sus requerimientos, el empresario también quiere :

C.I. conocer a nivel municipios, cuántas gasolineras son low-cost, cuantas no lo son, cuál es el precio promedio, el precio más bajo y el más caro de los siguientes carburantes: gasóleo A, y gasolina 95 e5 Premium , en todo el TERRITORIO NACIONAL, exceptuando las grandes CIUDADES ESPAÑOLAS ("MADRID", "BARCELONA", "SEVILLA" y "VALENCIA")

Para conocer el resultado a nivel municipios, se ha seleccionado las columnas con los precios solicitados, municipios, y clasificación entre low\_cost de las distintas comunidades autónomas, agrupándolo por municipio y su clasificación, y filtrando para eliminar los municipios de Madrid, Barcelona, Sevilla y Valencia, para poder calcular el precio máximo, mínimo y media de los dos tipos de combustible.

```
no_grandes_ciudades <- ds_col_lowcost %>% select(idcaa, id_municipio, municipio,
low_cost,      precio_gasoleo_a,      precio_gasolina_95_e5_premium) %>%
group_by(municipio, low_cost) %>% filter(!municipio %in% c("Madrid", "Barcelona",
"Sevilla", "Valencia")) %>%
```

```
summarise(max(precio_gasoleo_a), min(precio_gasoleo_a), mean(precio_gasoleo_a),
max(precio_gasolina_95_e5_premium), min(precio_gasolina_95_e5_premium),
mean(precio_gasolina_95_e5_premium))
```

Obteniendo un total de aproximadamente 4600 municipios.

	municipio	low_cost	max(precio_gasoleo_a)	min(precio_gasoleo_a)	mean(precio_gasoleo_a)	max(precio_gasolina_95_e5_premium)
1	Abadín	low_cost	1.429	1.429	1.429000	
2	Abadín	no_low_cost	1.409	1.409	1.409000	
3	Abadiño	low_cost	1.359	1.315	1.337000	
4	Abanilla	low_cost	1.319	1.299	1.309000	
5	Abanilla	no_low_cost	1.399	1.399	1.399000	
6	Abanto y Ciérvana-Abanto Zierbena	low_cost	1.459	1.449	1.455667	
7	Abanto y Ciérvana-Abanto Zierbena	no_low_cost	1.449	1.449	1.449000	
8	Abarán	low_cost	1.379	1.359	1.369000	
9	Abarán	no_low_cost	1.359	1.359	1.359000	
10	Abegondo	low_cost	1.278	1.278	1.278000	
11	Abegondo	no_low_cost	1.415	1.415	1.415000	
12	Abejar	no_low_cost	1.379	1.379	1.379000	
13	Abengibre	low_cost	1.349	1.349	1.349000	
14	Abenójar	no_low_cost	1.419	1.419	1.419000	
15	Abia	no_low_cost	1.389	1.389	1.389000	
16	Ablitas	low_cost	1.299	1.299	1.299000	

Showing 1 to 16 of 4,630 entries, 8 total columns

Imagen 9. A nivel municipios, los precios medio, mas alto y bajo de Gasolina 95 E5 Premium y Gasóleo A.

Para obtener cuantas gasolineras son low\_cost y no low\_cost, ejecutamos el siguiente código:

```
count_lowcost_municipios <- no_grandes_ciudades %>% group_by(low_cost)
%>% count(low_cost)
```

	low_cost	n
1	low_cost	2465
2	no_low_cost	2165

Imagen 10. A nivel municipios, el número de gasolineras low-cost y no low-cost.



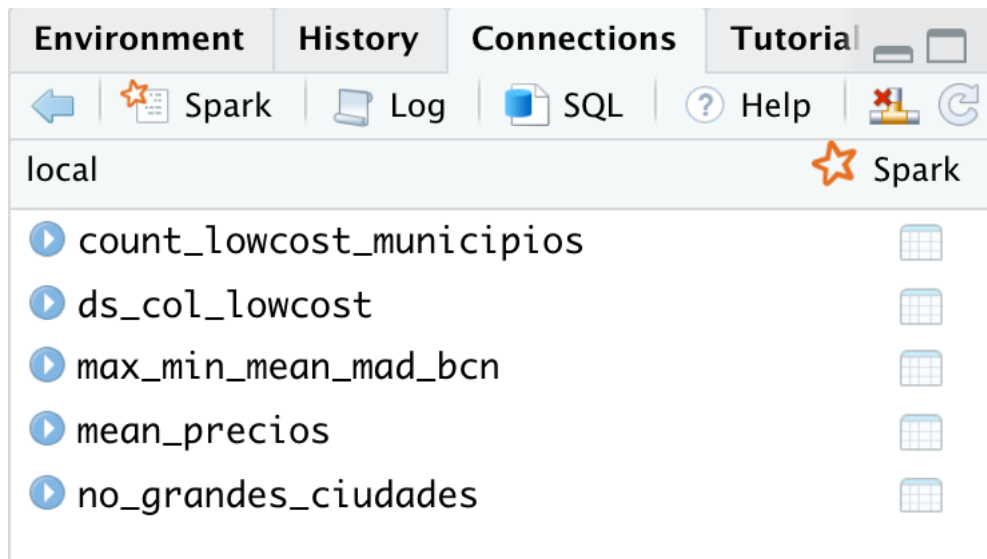


Imagen 11. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

C. II. Conseguido el objetivo, deberá guardar este “archivo” en una nueva tabla llamada `informe_no_grandes_ciudades_expediente` y deberá estar disponible también en su repositorio con el mismo nombre en formato Excel

“Descargo a través de R el archivo en csv, ya que no me permite descargarlo en Excel por un error en Java, pero luego al descargar el fichero del apartado D, ese si que me permite descargarlo con el formato Excel.”

Archivo de la tabla `informe_no_grandes_ciudades` en [GitHub](#).

Archivo disponible en [G-Drive](#)

D. I. Determine: Que gasolineras se encuentran abiertas las 24 horas exclusivamente, genere una nueva tabla llamada `no_24_horas` sin la variable `horario` ( es decir no debe aparecer esta columna).

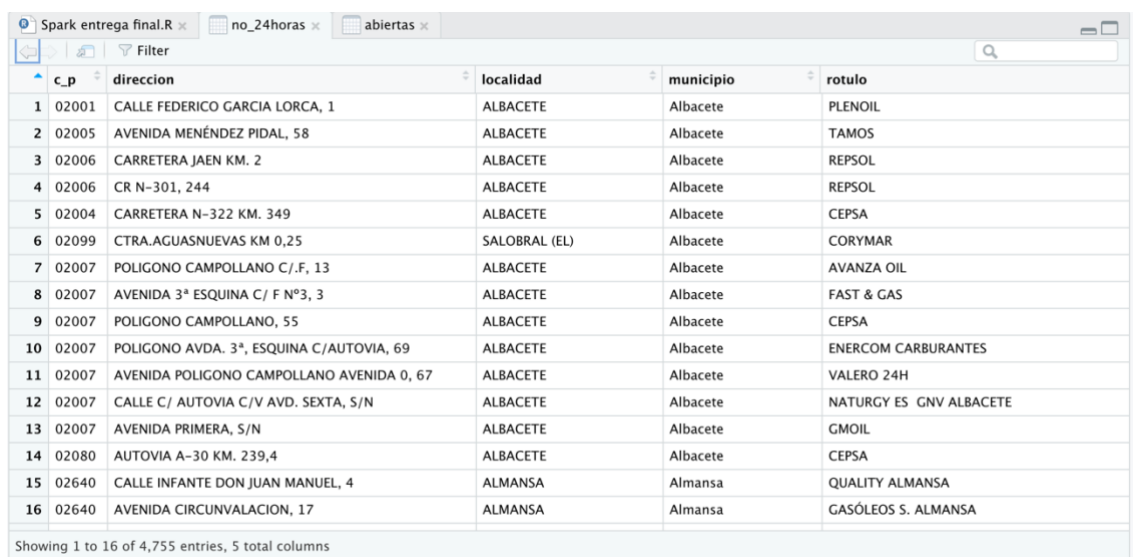
Para determinar que gasolineras se encuentran abiertas las 24 horas, selecciono los datos relevantes para el análisis de las gasolineras que encuentran abiertas como el código postal, dirección, localidad, municipio, rotulo y horario, para a continuación filtrar unicamnete por las gasolineras que se encuentran abiertas las 24 horas.

```
abiertas <- ds_col_lowcost %>% select(c_p, direccion, localidad, municipio, rotulo,
horario) %>% filter(
  horario=="L-D: 24H")
```

Para eliminar la columna `horario`, sobre el dataset, selecciono todas las columnas excepto la que contiene la variable `horario`.

```
no_24horas <- abiertas %>% select(c_p, direccion, localidad, municipio, rotulo, -horario)
```

Como resultado, obtenemos aproximadamente un total de 4750 gaoslineras abiertas las 24h en el territorio nacional.



	c_p	direccion	localidad	municipio	rotulo
1	02001	CALLE FEDERICO GARCIA LORCA, 1	ALBACETE	Albacete	PLENOIL
2	02005	AVENIDA MENÉNDEZ PIDAL, 58	ALBACETE	Albacete	TAMOS
3	02006	CARRETERA JAEN KM. 2	ALBACETE	Albacete	REPSOL
4	02006	CR N-301, 244	ALBACETE	Albacete	REPSOL
5	02004	CARRETERA N-322 KM. 349	ALBACETE	Albacete	CEPSA
6	02099	CTRA.AGUASNUEVAS KM 0,25	SALOBRA (EL)	Albacete	CORYMAR
7	02007	POLIGONO CAMPOLLANO C/.F, 13	ALBACETE	Albacete	AVANZA OIL
8	02007	AVENIDA 3ª ESQUINA C/ F Nº3, 3	ALBACETE	Albacete	FAST & GAS
9	02007	POLIGONO CAMPOLLANO, 55	ALBACETE	Albacete	CEPSA
10	02007	POLIGONO AVDA. 3ª, ESQUINA C/AUTOVIA, 69	ALBACETE	Albacete	ENERCOM CARBURANTES
11	02007	AVENIDA POLIGONO CAMPOLLANO AVENIDA 0, 67	ALBACETE	Albacete	VALERO 24H
12	02007	CALLE C/ AUTOVIA C/V AVD. SEXTA, 5/N	ALBACETE	Albacete	NATURGY ES GNV ALBACETE
13	02007	AVENIDA PRIMERA, 5/N	ALBACETE	Albacete	GMOIL
14	02080	AUTOVIA A-30 KM. 239,4	ALBACETE	Albacete	CEPSA
15	02640	CALLE INFANTE DON JUAN MANUEL, 4	ALMANSA	Almansa	QUALITY ALMANSA
16	02640	AVENIDA CIRCUNVALACION, 17	ALMANSA	Almansa	GASÓLEOS S. ALMANSA

Imagen 12. Gasolineras abiertas las 24 horas.

Alejandro Fernández Fraile (21614015)

Módulo II: Posicionamiento Empresarial del Big Data

Práctica Spark

07/01/2022

D. II. Conseguido el objetivo, deberá guardar este “archivo” en una nueva tabla llamada no\_24\_horas y deberá estar disponible también en su repositorio con el mismo nombre en formato Excel

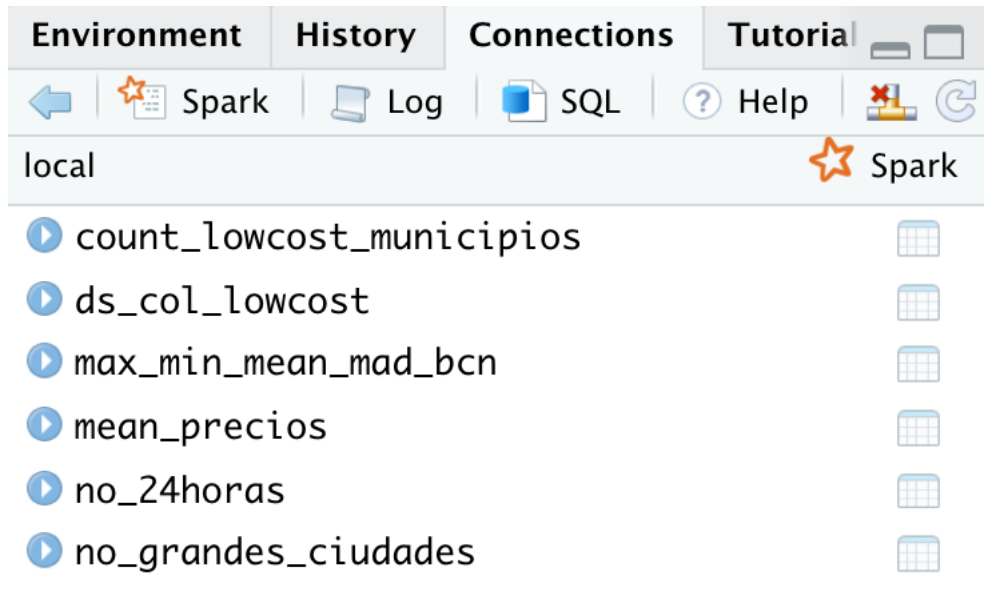


Imagen 13. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

Archivo de la tabla no\_24\_horas en [GitHub](#).

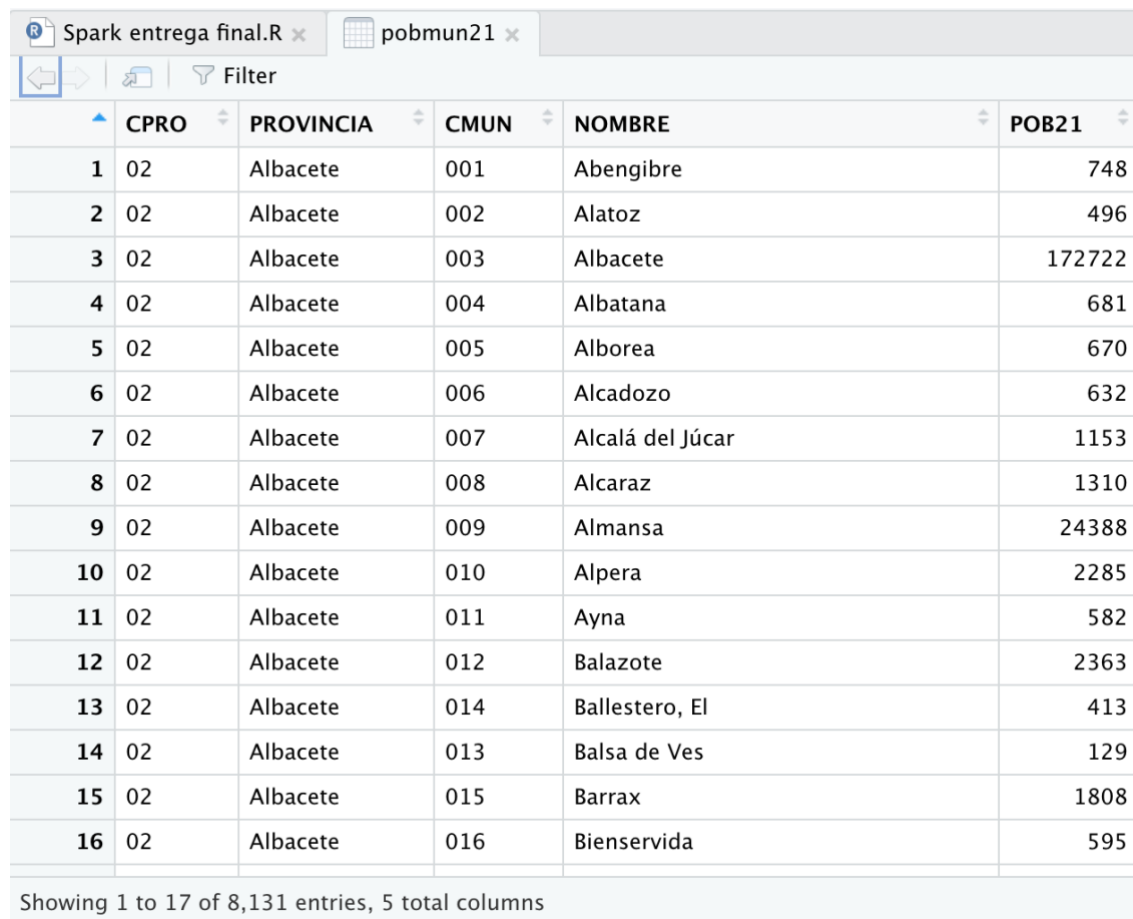
Archivo disponible en [G-Drive](#)

E. Uno de los factores más importantes para que el empresario se decante a instalar nuevas gasolineras es la demanda que viene dada por la población y la competencia existente en un municipio donde se pretenda implantar las gasolineras, para responder a esta pregunta de negocio,

Primero de todo, procedo a descargar desde el INE el dataset de población a nivel municipal para todo el territorio nacional más actualizado, que corresponde al año 2021.

Una vez descargado, importo este dataset a R para poder trabajar con él.

```
pobmun21 <- read_excel("Desktop/G- Drive/2.Posicionamiento Big Data/Spark/TAREA SPARK/pobmun21.xlsx")
```



	CPRO	PROVINCIA	CMUN	NOMBRE	POB21
1	02	Albacete	001	Abengibre	748
2	02	Albacete	002	Alatoz	496
3	02	Albacete	003	Albacete	172722
4	02	Albacete	004	Albatana	681
5	02	Albacete	005	Alborea	670
6	02	Albacete	006	Alcadozo	632
7	02	Albacete	007	Alcalá del Júcar	1153
8	02	Albacete	008	Alcaraz	1310
9	02	Albacete	009	Almansa	24388
10	02	Albacete	010	Alpera	2285
11	02	Albacete	011	Ayna	582
12	02	Albacete	012	Balazote	2363
13	02	Albacete	014	Ballestero, El	413
14	02	Albacete	013	Balsa de Ves	129
15	02	Albacete	015	Barrax	1808
16	02	Albacete	016	Bienservida	595

Showing 1 to 17 of 8,131 entries, 5 total columns

Imagen 14. Extracto del dataset recién cargado

Una vez cargado el dataset, procedo a examinarlo y prepararlo para añadirlo al dataset original renombrando las columnas para que coincidan con el otro dataset.

```
names(pobmun21) = c("id_provincia", "provincia", "cnum", "municipio", "poblacion")
```

	id_provincia	provincia	cnum	municipio	poblacion
1	02	Albacete	001	Abengibre	748

Imagen 15. Tabla Pobmun21 con las columnas renombradas

E.I. Deberá añadir la población al dataset original creando una nueva columna denominada población, esta información debe ser veraz y la más actualizada, la población debe estar a nivel municipal (todo el territorio nacional)

Para añadir la nueva columna población al dataset original, he optado por utilizar un left join para unir el dataset original y el dataset recién descargado mediante la columna en común “municipio”.

```
ds_poblacion<-left_join(x=ds, y=pobmun21, by="municipio")
```

bio_etanol	percent_ester_metilico	ideess	id_municipio	id_provincia.x	idcaa	id_provincia.y	provincia.y	cnum	poblacion
0	0	4375	52	02	07	02	Albacete	001	748
0	0	5122	53	02	07	02	Albacete	002	496
0	0	10765	54	02	07	02	Albacete	003	172722
0	0	4438	54	02	07	02	Albacete	003	172722
0	0	13933	54	02	07	02	Albacete	003	172722
0	0	12054	54	02	07	02	Albacete	003	172722
0	0	5195	54	02	07	02	Albacete	003	172722
0	0	4369	54	02	07	02	Albacete	003	172722
0	0	14123	54	02	07	02	Albacete	003	172722
0	0	15000	54	02	07	02	Albacete	003	172722
0	0	5313	54	02	07	02	Albacete	003	172722
0	0	5245	54	02	07	02	Albacete	003	172722
0	0	4428	54	02	07	02	Albacete	003	172722
0	0	5116	54	02	07	02	Albacete	003	172722
0	0	5222	54	02	07	02	Albacete	003	172722
0	0	4795	54	02	07	02	Albacete	003	172722

Showing 1 to 16 of 11,430 entries, 36 total columns

Imagen 16. Dataset original una vez añadida la columna población

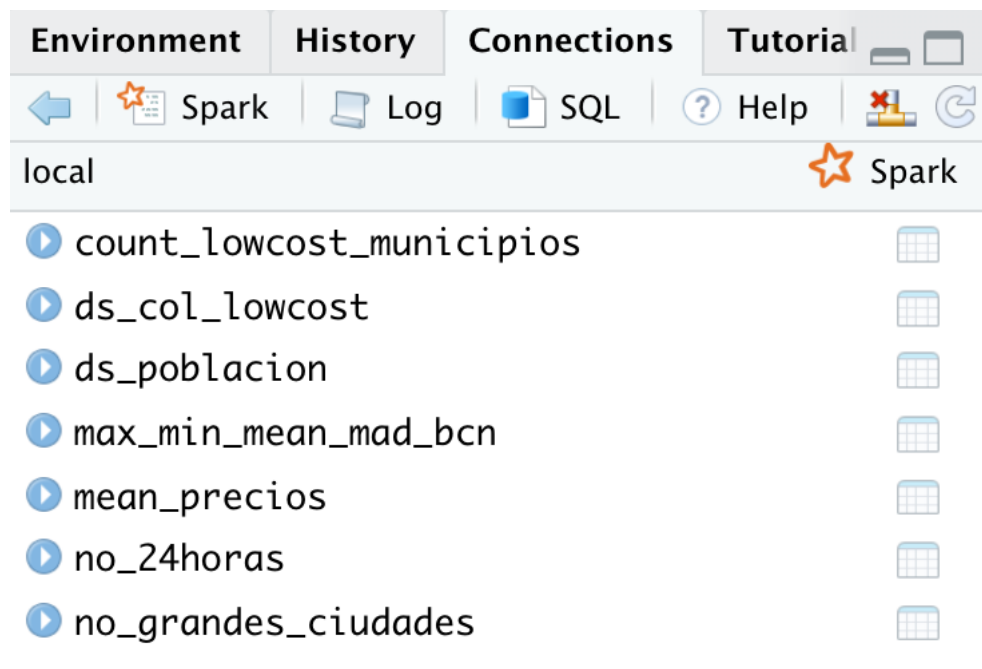


Imagen 17. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

E.II. este empresario ha visto varios sitios donde potencialmente le gustaría instalar su gasolinera, esos sitios están representados por la dirección, desde esta última calcule cuanta competencia (nombre de la gasolinera y dirección) tiene en :

Para hacer este ejercicio, me he basado en uno que hicimos en clase bastante similar.

Dado que no se especifica cuáles son los sitios que ha visto donde potencialmente le gustaría instalar la gasolinera, he generado el siguiente mapa de competencia que imprime a nivel nacional la ubicación y la marca de cada una de las gasolineras.

```
ds_poblacion %>% select(rotulo, latitud, longitud_wgs84, direccion, municipio,
provincia.x) %>%
```

```
leaflet() %>% addTiles() %>% addCircleMarkers(lng = ~longitud_wgs84, lat =
~latitud, popup = ~rotulo, label = ~municipio)
```

Además, otra opción en el caso de que se conociera la ubicación deseada, y dependiendo el análisis que se quiera obtener, se puede filtrar por las distintas variables tal y como voy a mostrar para cada uno de los siguientes cálculos pedidos:

### 1. En un radio de 1 km ( genere mapa\_competencia1.html)

Para la obtención de este resultado, decido filtrar por provincia “Madrid” para obtener las todas las gasolineras de la Comunidad, con el rótulo, y alrededor un circulo que marca el radio establecido de 1km, en este caso 1000m.

Para añadir el radio, he utilizado la función addCircles obtenida a través de la ayuda de la función en r.

```
ds_poblacion %>% select(rotulo, latitud, longitud_wgs84, direccion, municipio,
provincia.y) %>%
```

```
filter(provincia.y=='Madrid') %>%
```

```
leaflet() %>% addTiles() %>% addCircleMarkers(lng = ~longitud_wgs84, lat =
~latitud, popup = ~rotulo, label = ~provincia.y) %>% addCircles(lng = ~longitud_wgs84,
lat = ~latitud, radius = 1000)
```

Esto imprime el siguiente mapa a nivel Provincia Madrid:

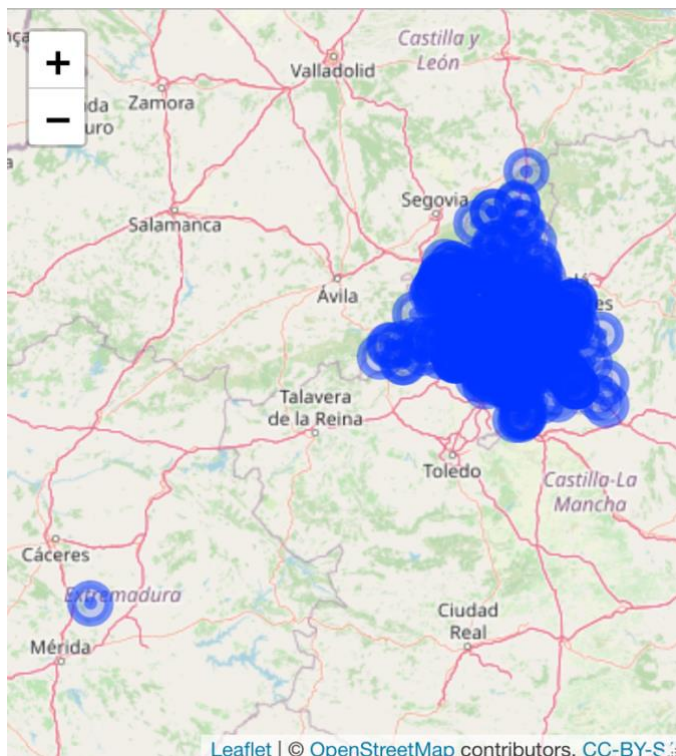


Imagen 18. Mapa competencia con 1km de radio en la Comunidad de Madrid.



Un dato llamativo es que al hacer ese filtro, salta una gasolinera de Extremadura como si estuviera en la comunidad de Madrid debido a supuestamente un error del dataset.

## 2. En un radio de 2 km ( genere mapa\_competencia2.html)

Para la obtención de este resultado, decido filtrar por municipio “San Sebastián de los Reyes” para obtener las todas las gasolineras de San Sebastián de los Reyes, con el rótulo, y alrededor un círculo que marca el radio establecido de 2km, en este caso 2000m.

```
ds_poblacion %>% select(rotulo, latitud, longitud_wgs84, direccion, municipio,
provincia.x) %>%
```

```
filter(municipio=='San Sebastián de los Reyes') %>%
```

```
leaflet() %>% addTiles() %>% addCircleMarkers(lng = ~longitud_wgs84, lat =
~latitud, popup = ~rotulo,label = ~municipio) %>% addCircles(lng = ~longitud_wgs84,
lat = ~latitud, radius = 2000)
```

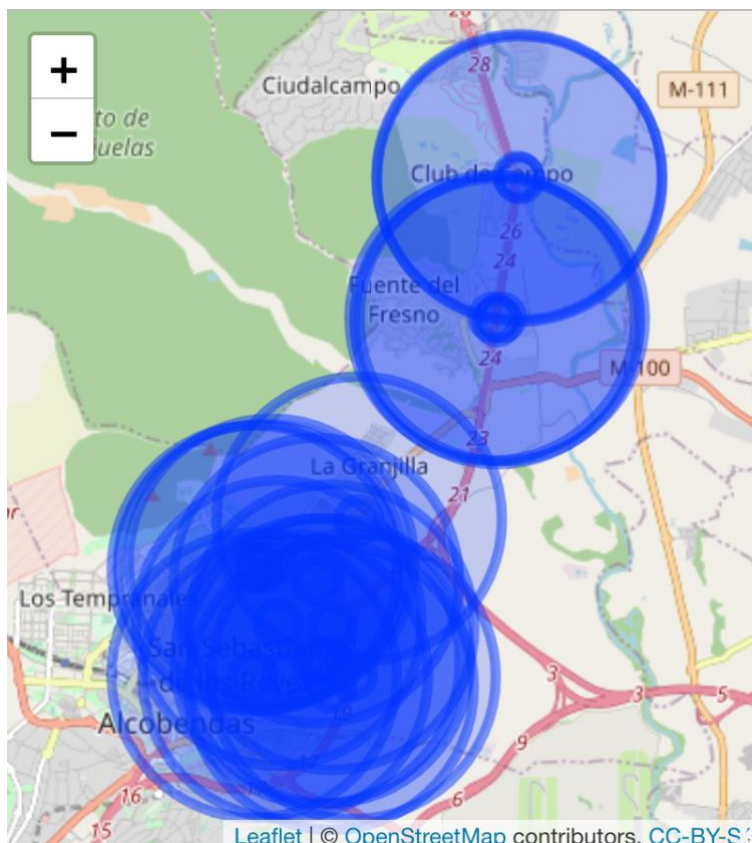


Imagen 19. Mapa competencia con 2km de radio en San Sebastián de los Reyes



3. En un radio de 4 km ( genere mapa\_competencia3.html)

Para la obtención de este resultado, decido filtrar por un municipio al azar elegido del dataset, en este caso “Mula” para obtener las todas las gasolineras de Mula, con el rótulo, y alrededor un círculo que marca el radio establecido de 4km.

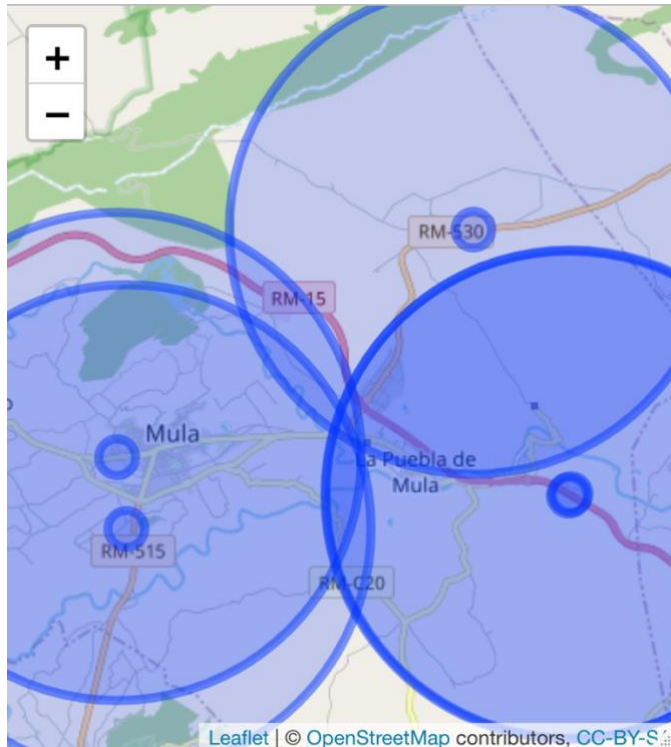


Imagen 20. Mapa competencia con 4km de radio en Mula, Murcia

Archivos de mapas competencia disponible en [GitHub](#).

Archivos de mapas competencia disponible en [G-Drive](#)

E.III. Genere el TopTen de municipios entre todo el territorio nacional excepto el territorio insular, donde no existan gasolineras 24 horas, agrupadas entre low-cost y no low-cost, deberá guardar este “archivo” en una nueva tabla llamada `informe_top_ten_expediente` y deberá estar disponible también en su repositorio con el mismo nombre en formato csv.

Para generar el top10 de municipios excluyendo las islas donde no haya gasolineras 24h y a su vez agrupadas por su clasificación `low_cost`, he procedido a realizarlo en dos pasos:

El primero, seleccionando el dataset que ya contiene la columna `low_cost`, filtrando para eliminar las provincias insulares, y a su vez las que están abiertas las 24h, agrupando el resultado por municipio y su clasificación low cost obteniendo una suma del total de gasolineras.

```
gasolineras_municipio_24h<- ds_col_lowcost %>% filter (!provincia %in%
c("BALEARS (ILLES)", "PALMAS (LAS)")) %>%
```

```
filter (!horario=="L-D: 24H" ) %>% group_by (municipio, low_cost) %>% count()
```

	municipio	low_cost	n
1	Abadín	low_cost	1
2	Abadiño	low_cost	1
3	Abanilla	low_cost	2
4	Abanilla	no_low_cost	1
5	Abanto y Ciérvana-Abanto Zierbena	low_cost	1
6	Abanto y Ciérvana-Abanto Zierbena	no_low_cost	1
7	Abarán	low_cost	1
8	Abarán	no_low_cost	1
9	Abegondo	low_cost	1
10	Abegondo	no_low_cost	1
11	Abejar	no_low_cost	1
12	Abengibre	low_cost	1
13	Abenójar	no_low_cost	1
14	Abla	no_low_cost	1
15	Abrera	low_cost	1
16	Aceuchal	no_low_cost	1

Showing 1 to 17 of 3,416 entries, 3 total columns

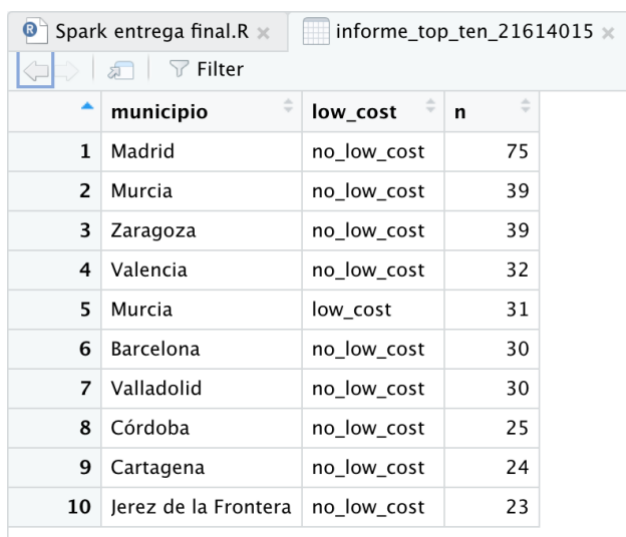
Imagen 21. Municipios de España donde no existan gasolineras 24h.

Este resultado imprime que hay aproximadamente un total de 3400 municipios donde no existen gasolineras 24h.

El segundo paso del ejercicio es ordenar los 3400 municipios en orden descendiente, y seleccionar únicamente el Top10 de ellos.

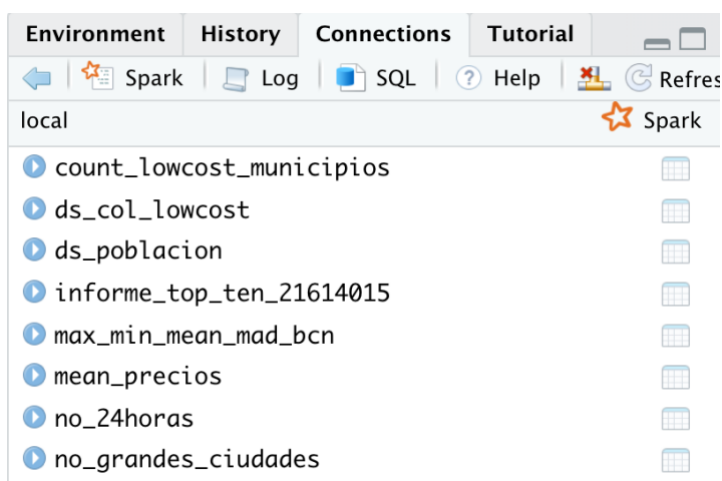
```
informe_top_ten_21614015 <-  
gasolineras_municipio_24h[order(gasolineras_municipio_24h$n, decreasing = TRUE),]  
%>% head(10)
```

Con este código obtenemos que los siguientes municipios no contienen gasolineras 24h.



	municipio	low_cost	n
1	Madrid	no_low_cost	75
2	Murcia	no_low_cost	39
3	Zaragoza	no_low_cost	39
4	Valencia	no_low_cost	32
5	Murcia	low_cost	31
6	Barcelona	no_low_cost	30
7	Valladolid	no_low_cost	30
8	Córdoba	no_low_cost	25
9	Cartagena	no_low_cost	24
10	Jerez de la Frontera	no_low_cost	23

Imagen 22. Top 10 Municipios de España donde no existan gasolineras 24h.



Environment	History	Connections	Tutorial
local	Spark	Log	SQL
count_lowcost_municipios			
ds_col_lowcost			
ds_poblacion			
informe_top_ten_21614015			
max_min_mean_mad_bcn			
mean_precios			
no_24horas			
no_grandes_ciudades			

Imagen 23. Captura de la ventana Spark que contiene las tablas recién creadas para este apartado.

Alejandro Fernández Fraile (21614015)

Módulo II: Posicionamiento Empresarial del Big Data

Práctica Spark

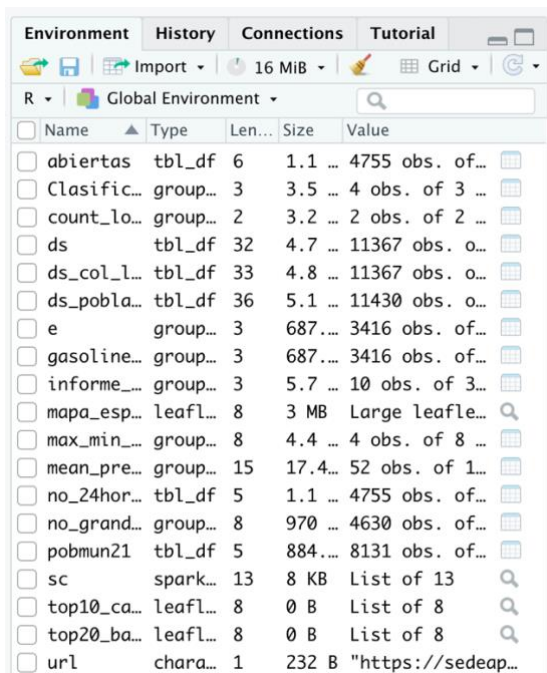
07/01/2022

Archivo de la tabla informe\_top\_10 en [GitHub](#).

Archivo disponible en [G-Drive](#)

### Información adicional:

1. Adjunto imagen del todos los objetos creados en el environment de R.



Name	Type	Len...	Size	Value
abiertas	tbl_df	6	1.1 ...	4755 obs. of...
Clasific...	group...	3	3.5 ...	4 obs. of 3 ...
count_lo...	group...	2	3.2 ...	2 obs. of 2 ...
ds	tbl_df	32	4.7 ...	11367 obs. o...
ds_col_l...	tbl_df	33	4.8 ...	11367 obs. o...
ds_pobla...	tbl_df	36	5.1 ...	11430 obs. o...
e	group...	3	687...	3416 obs. of...
gasoline...	group...	3	687...	3416 obs. of...
informe_...	group...	3	5.7 ...	10 obs. of 3...
mapa_esp...	leafl...	8	3 MB	Large leafle...
max_min...	group...	8	4.4 ...	4 obs. of 8 ...
mean_pre...	group...	15	17.4...	52 obs. of 1...
no_24hor...	tbl_df	5	1.1 ...	4755 obs. of...
no_grand...	group...	8	970 ...	4630 obs. of...
pobmun21	tbl_df	5	884...	8131 obs. of...
sc	spark...	13	8 KB	List of 13
top10_ca...	leafl...	8	0 B	List of 8
top20_ba...	leafl...	8	0 B	List of 8
url	chara...	1	232 B	"https://sedeap...

Imagen 23. Captura de la ventana environment que contiene todos los objetos creados.

2. Adjunto también por si fuera necesario el link a los archivos solicitados subidos a google drive, incluyendo el Excel de los datos de población utilizado para el Ejercicio E, por si hubiera algún error en el repositorio de github

[Google drive spark 21614015](#)

[Archivo de R con todas las operaciones, gráficos y códigos en G Drive](#)

Adjunto también el archivo de R, datos de población y este pdf en [GitHub](#).