Examen DS Intermedio
Enrique Santibáñez Cortés

Sección B

B1.QQP

Descarga la Base de datos histórica de **Quién es Quién en los Precios** de Profeco y resuelve los siguientes incisos. Para el procesamiento de los datos y análisis exploratorio debes usar Spark SQL en el lenguanje de programación de tu elección.

Hacemos la conección Spark.

```
#Cargamos los paquetes para los datos
library(sparklyr)
library(tidyverse)
#Configuramos Spark.
config = spark_config()
config$`sparklyr.shell.driver-memory` <- "4G"
config$`sparklyr.shell.executor-memory` <- "4G"
config$`spark.yarn.executor.memoryOverhead` <- "512"
#Realizamos la conección y cargamos los datos en Spark.
sc = spark_connect(master = "local", config = config)
```

1. Procesamienoo de los datos

```
#Leemos los datos.
setwd("~/Documentos/Servicio_social/Examen intermedio/")
all_data <- spark_read_csv(sc, name = "air", path = "all_data.csv")
```

a) ¿Cuántos registros hay?

Respuesta:

```
all_data%>%tally()
```

```
## # Source: spark<?> [?? x 1]
##          n
##      <dbl>
## 1 62530715
```

b) ¿Cuántas categorías?

Respuesta:

```
#Y el número total de catedorías es: (RESPUESTA)
all_data%>%distinct(categoria)%>%arrange(categoria)%>%tally()
```

```
## # Source: spark<?> [?? x 1]
##       n
##   <dbl>
## 1    41
```

```
#Las principales categorías que más aparecen son:
head(all_data%>%group_by(categoria)%>%count()%>%arrange(desc(n)))
```

```
## # Source:     spark<?> [?? x 2]
## # Groups:     categoria
```

```
## # Ordered by: desc(n)
##   categoria                              n
##   <chr>                              <dbl>
## 1 MEDICAMENTOS                    11485813
## 2 ARTS. PARA EL CUIDADO PERSONAL   4143846
## 3 APARATOS ELECTRICOS              3471515
## 4 DETERGENTES Y PRODUCTOS SIMILARES 3085450
## 5 CARNES FRIAS SECAS Y EMBUTIDOS   2850197
## 6 DERIVADOS DE LECHE               2679630
```

c) ¿Cuántas cadenas comerciales están siendo monitoreadas?

Respuesta:

```
# En total son: (RESPUESTA)
all_data%>%distinct(cadenaComercial)%>%tally()
```

```
## # Source: spark<?> [?? x 1]
##       n
##   <dbl>
## 1   705
```

```
# Cada cadena comercial tiene un número de registros:
head(all_data%>%group_by(cadenaComercial)%>%count()%>%arrange(desc(n)))
```

```
## # Source:     spark<?> [?? x 2]
## # Groups:     cadenaComercial
## # Ordered by: desc(n)
##   cadenaComercial               n
##   <chr>                     <dbl>
## 1 WAL-MART                8643133
## 2 BODEGA AURRERA          6765453
## 3 SORIANA                 6546211
## 4 MEGA COMERCIAL MEXICANA 4899509
## 5 CHEDRAUI                4221625
## 6 COMERCIAL MEXICANA      2598903
```

d) ¿Cómo podrías determinar la calidad de los datos? ¿Detectaste algún tipo de incosistencia o error en la fuente?

Respuesta:

Me percate de varios problemas en la fuente. En algunos campos existen NAN los cuales si los eliminamos se reduce la base de datos de 62530715 a 61593556 es decir los datos perdidos son 937159, ahora en los campos estado y municipio existe varias incosistencias como por ejemplo: existe registros con estados **COL. EDUARDO GUERRA** y otros con **3 ESQ. SUR 125**, para el municipio de León Guanajuato viene como **León** y **"León**. En los productos también existen incosistencia en la fuente, como por ejemplo: **Acondicionador Y Enjuague** vs **Acondicionador / Enjuague**.

e) ¿Cuáles son los productos más monitoreados en cada entidad?

Respuesta:

```
# El top3 de articulos para cada estado es:
all_data%>%group_by(estado)%>%count(producto,sort=T)%>%top_n(3)
```

```
## Selecting by n
```

```
## Warning: `lang_name()` is deprecated as of rlang 0.2.0.
## Please use `call_name()` instead.
```

```
## This warning is displayed once per session.
```

```
## Warning: `lang()` is deprecated as of rlang 0.2.0.
## Please use `call2()` instead.
## This warning is displayed once per session.
```

```
## # Source:     spark<?> [?? x 3]
## # Groups:     estado
## # Ordered by: desc(n)
##    estado     producto                  n
##    <chr>      <chr>                 <dbl>
##  1 CAMPECHE   FUD                   12960
##  2 CAMPECHE   REFRESCO              11333
##  3 CAMPECHE   PANTALLAS             10449
##  4 GUANAJUATO REFRESCO              49441
##  5 GUANAJUATO DETERGENTE P/ROPA     36618
##  6 GUANAJUATO VARIOS                35278
##  7 NAYARIT    REFRESCO               8003
##  8 NAYARIT    PANTALLAS              7083
##  9 NAYARIT    FUD                    6644
## 10 OAXACA     LECHE ULTRAPASTEURIZADA 18078
## # ... with more rows
```

f) ¿Cuál es la cadena comercial con mayor variead de productos?

Respuesta:

```
#El top 5 de las cadenas con más variedad de productos son:
head(all_data%>%select(cadenaComercial,producto)%>%group_by(cadenaComercial)%>%distinct(cadenaComercial
```

```
## # Source:     spark<?> [?? x 2]
## # Groups:     cadenaComercial
## # Ordered by: desc(n)
##   cadenaComercial             n
##   <chr>                   <dbl>
## 1 SORIANA                  1059
## 2 WAL-MART                 1051
## 3 MEGA COMERCIAL MEXICANA  1049
## 4 COMERCIAL MEXICANA       1036
## 5 CHEDRAUI                 1026
## 6 MERCADO SORIANA          1024
```

```
#Y los productos son:
set.seed(1)
cade<-all_data%>%select(cadenaComercial,producto)%>%filter(cadenaComercial=="SORIANA")%>%distinct(produ
sample_n(cade,5)
```

```
## # A tibble: 5 x 1
##   producto
##   <chr>
## 1 BRAN FLAKES
## 2 DICCIONARIO
## 3 RELENZA
## 4 ASPIRINA PROTECT
## 5 MICONAZOL
```

3. Visualización
   a) Genera un mapa que nos permita identificar la oferta de categorías en la zona metropolitana de

León Guanajuato y el nivel de precios en cada una de ellas. Se darán puntos extra si el mapa es interactivo.

Respuesta:

Filtramos los datos que pertenecen a la zona metropolitana de Léon Guanajuato (Nota: solo vienen datos del municiopio de León), y los guardamos en un csv, para así no tener que hacer ese filtro cada vez que se ocupen los datos.

Posteriomente ocuparemos la paquetería Shiny para crear el mapa interactivo, el cuál esta formado: por un mapa, una tabla para la visualización de los datos, y dos **SelectInput** que controlan el producto y la fecha. Es decir, en el mapa se podrán consultar el precio de las categorías a una fecha específcada. Se consideró que solo mostrará el precio de la categoría con el registro más cerca a la fecha específicada.

```
# Nos desconectamos de Spark
spark_disconnect(sc)
```

```
## NULL
```