Wizeline Data Engineering Bootcamp Challenge

Luis David Hernandez Zamarripa

To approach this challenge first I loaded the sample data into a Pandas data frame to answer the requested questions

```
url="https://raw.githubusercontent.com/DavidHZama/data-engineering-bootcamp/main/data/sample.csv"
s=requests.get(url).content
df=pd.read_csv(io.StringIO(s.decode('utf-8')))
df
```

categoria	catalogo	precio	fechaRegistro	cadenaComercial	giro	nombreComercial	direccion	estado	municipio	latitud	longitud
MATERIAL ESCOLAR	UTILES ESCOLARES	25.90	2011-05-18 00:00:00.000	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.296990	-99.125417
MATERIAL ESCOLAR	UTILES ESCOLARES	27.50	2011-05-18 00:00:00.000	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.296990	-99.125417
MATERIAL ESCOLAR	UTILES ESCOLARES	13.90	2011-05-18 00:00:00.000	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.296990	-99.125417
MATERIAL ESCOLAR	UTILES ESCOLARES	46.90	2011-05-18 00:00:00.000	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.296990	-99.125417
						ABASTECEDORA	CANNES No.				

Once I had tested my code with the sample data I decided to load the big .csv file, which was a bit more than 20gb, so i decided to load only the columns i needed to solve this challenge, this way processing times were faster

```
df = pd.read_csv('all_data.csv', index_col=None,low_memory=False, usecols=['cadenaComercial','estado','producto'])
df
```

	producto	cadenaComercial	estado
0	CUADERNO FORMA ITALIANA	ABASTECEDORA LUMEN	DISTRITO FEDERAL
1	CRAYONES	ABASTECEDORA LUMEN	DISTRITO FEDERAL
2	CRAYONES	ABASTECEDORA LUMEN	DISTRITO FEDERAL
3	COLORES DE MADERA	ABASTECEDORA LUMEN	DISTRITO FEDERAL

Memory usage was a bit more than 1.4 GB instead of the 20+ GB

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62530715 entries, 0 to 62530714
Data columns (total 3 columns):
producto           object
cadenaComercial    object
estado             object
dtypes: object(3)
memory usage: 1.4+ GB
```

How many commercial chains are monitored, and therefore, included in this database?

```
print("Number of monitored commercial chains")
print(len(pd.unique(df['cadenaComercial'])))
```

```
Number of monitored commercial chains
706
```

What are the top 10 monitored products by State?

```
Top10perstate =
df.groupby(['estado','producto'])['producto'].count().reset_index(name='
numReportes').sort_values(['estado','numReportes'],
ascending=False).groupby(['estado']).head(10)
top10perstate
```

	estado	producto	numReportes
29662	ZACATECAS	DETERGENTE P/ROPA	20884
29904	ZACATECAS	LECHE ULTRAPASTEURIZADA	17309
29899	ZACATECAS	LAVADORAS	16072
29961	ZACATECAS	MAYONESA	15927
29843	ZACATECAS	JABON DE TOCADOR	15926
29764	ZACATECAS	FUD	15541
30218	ZACATECAS	SHAMPOO	15012
29560	ZACATECAS	CHILES EN LATA	14866
29610	ZACATECAS	COMPONENTES DE AUDIO	14799
30165	ZACATECAS	REFRESCO	13925
28865	YUCATÁN	LECHE ULTRAPASTEURIZADA	35991
28621	YUCATÁN	DETERGENTE P/ROPA	33390
29132	YUCATÁN	REFRESCO	33235
28724	YUCATÁN	FUD	31885
28860	YUCATÁN	LAVADORAS	27013
28519	YUCATÁN	CHILES EN LATA	24472
28925	YUCATÁN	MAYONESA	23629
29185	YUCATÁN	SHAMPOO	23433
29082	YUCATÁN	PLANCHAS	22816
28862	YUCATÁN	LECHE EN POLVO	22696

Which is the commercial chain with the highest number of monitored products?

```
comercialChain =
df.groupby('cadenaComercial')['producto'].nunique().reset_index(name='nu
mProductos').sort_values(['numProductos'], ascending=False)

print('The commercial chain with the highest number of monitored
products is')
print(comercialChain.iloc[0]['cadenaComercial'])

comercialChain.head()
```

```
The commercial chain with the highest number of monitored products is
SORIANA
```

	cadenaComercial	numProductos
574	SORIANA	1059
683	WAL-MART	1051
301	MEGA COMERCIAL MEXICANA	1049
65	COMERCIAL MEXICANA	1036
58	CHEDRAUI	1026

Use the data to find an interesting fact.
I decided to rank the states by number of reports:

```
df.groupby(['estado'])['estado'].count().reset_index(name='numReportes')
.sort_values(['numReportes'],
ascending=False).groupby(['estado']).head(32)
```

estado	numReportes
DISTRITO FEDERAL	11284102
MÉXICO	8173302
JALISCO	4552128
NUEVO LEÓN	3171091
GUANAJUATO	2638456
YUCATÁN	2300994
MICHOACÁN DE OCAMPO	2093037
TLAXCALA	2081024
QUINTANA ROO	2076525
PUEBLA	2021476
TABASCO	1842633

What are the lessons learned from this exercise?

It was nice to review .groupby and experiment nesting them.

Can you identify other ways to approach this problem? Explain.

I would approach this using Dask instead of Pandas, this way we could use the entire dataset and not be limited by my computer's RAM