Data Engineering Bootcamp Challenge

Profeco Exploratory Analysis

Alejandra Elizabeth Moreno Morales

Challenge description

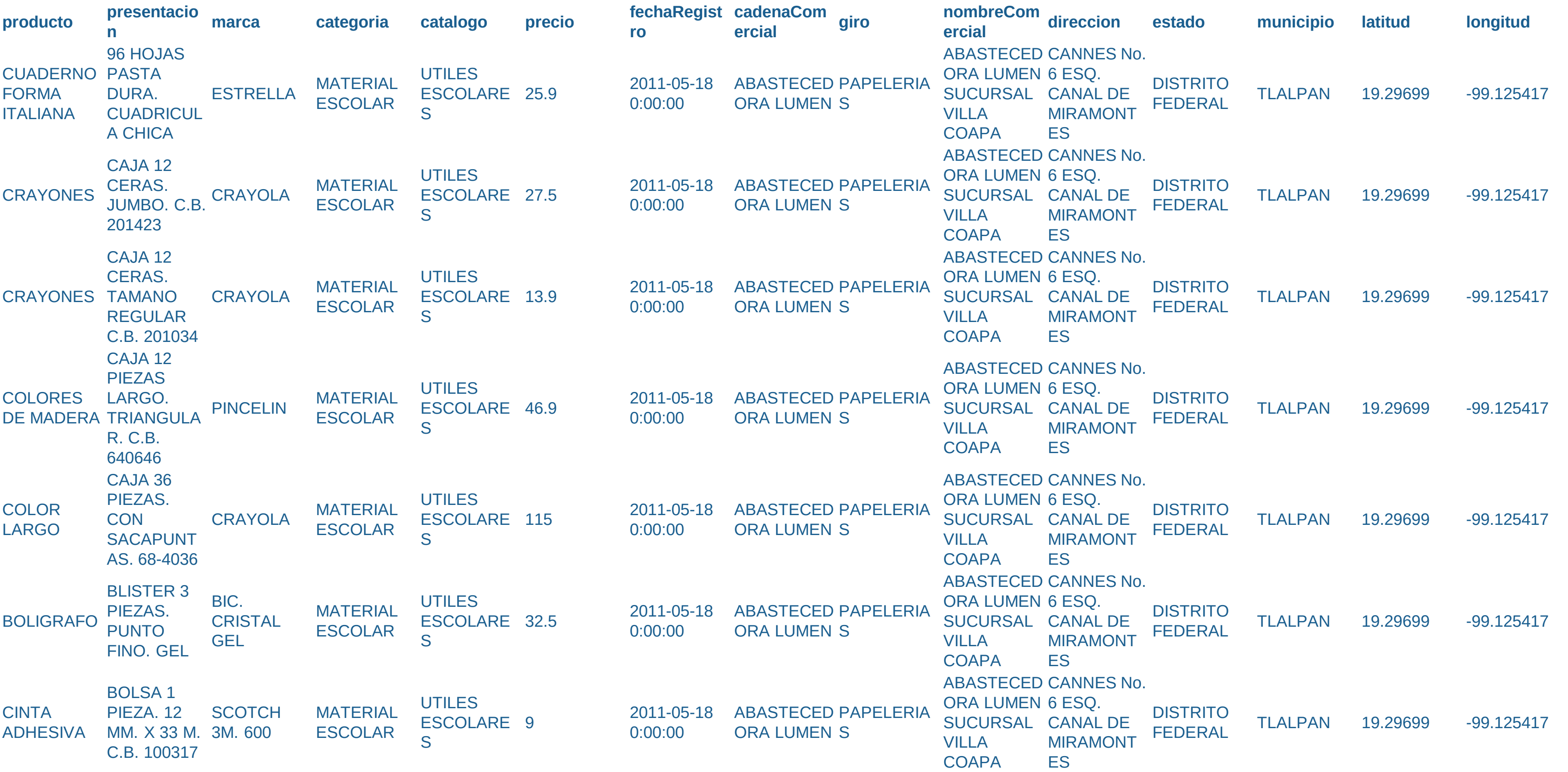
The Customer Service team at Profeco (Mexican Consumer Protection Agency) wants to analyze the monitored products in Mexico. The IT team downloaded the database into an Google Drive on a CSV file of about 20GB.

Our task as Data Engineer is processing the data and creating an exploratory analysis with Python Pandas without using pure Python functions.

Processor Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, RAM 12 Gb, Windows 10 Home

- Dataframes
- Chunks
- Methods (count(), sum(), add() and group by()

Csv file structure



producto	presentacion	marca	categoria	catalogo	precio	fechaRegistro	cadenaComercial	giro	nombreComercial	direccion	estado	municipio	latitud	longitud
CUADERNO FORMA ITALIANA	96 HOJAS PASTA DURA. CUADRICULA CHICA	ESTRELLA	MATERIAL ESCOLAR	UTILES ESCOLARES	25.9	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
CRAYONES	CAJA 12 CERAS. JUMBO. C.B. 201423	CRAYOLA	MATERIAL ESCOLAR	UTILES ESCOLARES	27.5	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
CRAYONES	CAJA 12 CERAS. TAMANO REGULAR C.B. 201034	CRAYOLA	MATERIAL ESCOLAR	UTILES ESCOLARES	13.9	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
COLORES DE MADERA	CAJA 12 PIEZAS LARGO. TRIANGULAR. C.B. 640646	PINCELIN	MATERIAL ESCOLAR	UTILES ESCOLARES	46.9	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
COLOR LARGO	CAJA 36 PIEZAS. CON SACAPUNTAS. 68-4036	CRAYOLA	MATERIAL ESCOLAR	UTILES ESCOLARES	115	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
BOLIGRAFO	BLISTER 3 PIEZAS. PUNTO FINO. GEL	BIC. CRISTAL GEL	MATERIAL ESCOLAR	UTILES ESCOLARES	32.5	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417
CINTA ADHESIVA	BOLSA 1 PIEZA. 12 MM. X 33 M. C.B. 100317	SCOTCH 3M. 600	MATERIAL ESCOLAR	UTILES ESCOLARES	9	2011-05-18 0:00:00	ABASTECEDORA LUMEN	PAPELERIAS	ABASTECEDORA LUMEN SUCURSAL VILLA COAPA	CANNES No. 6 ESQ. CANAL DE MIRAMONTES	DISTRITO FEDERAL	TLALPAN	19.29699	-99.125417

TARGET

Products

*Presentation

*Branch

*Price

Commercial chains

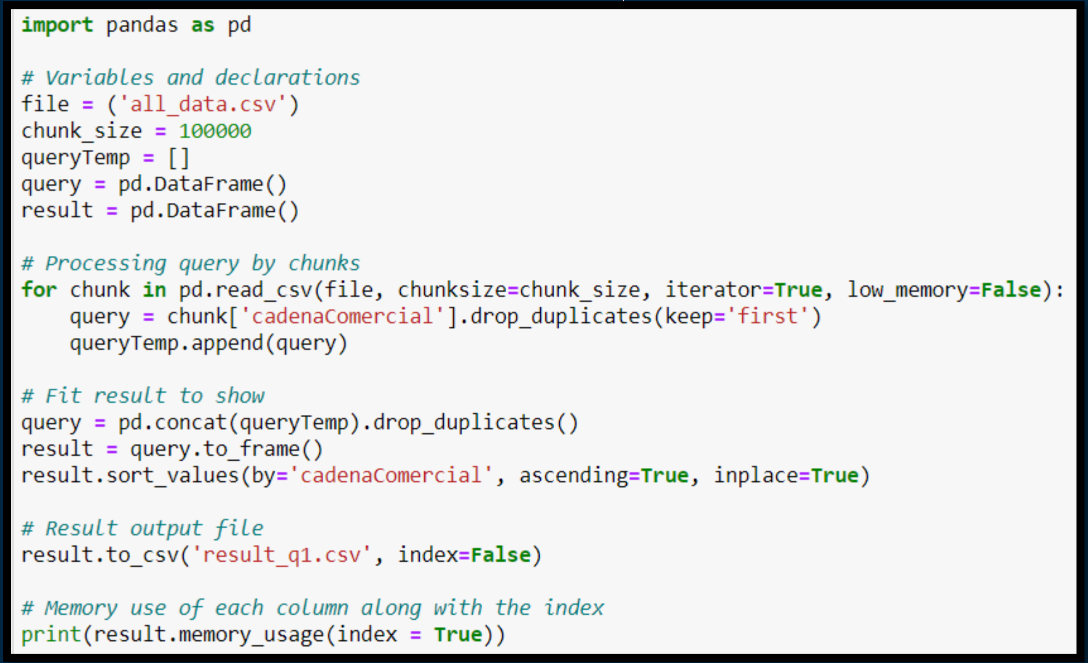
*Commercial chains offices
*Address
*State
*Location

Profeco is a institution in charge of defending the rights of consumers and its main objective is guarantee fair consumer relations.
Profeco monitor products and its characteristics around commercial chains, so need to have useful information to make data-driven decisions.

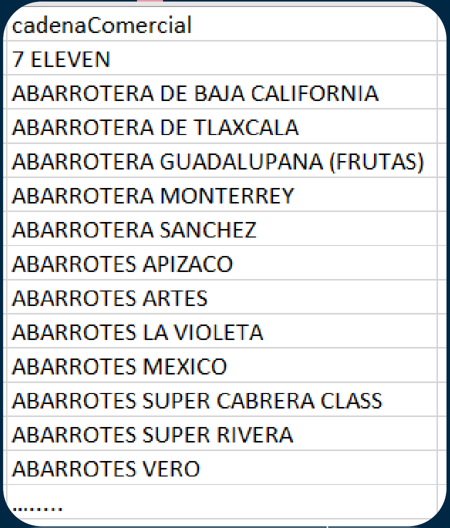
Analysis

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

01



```python
import pandas as pd

# Variables and declarations
file = ('all_data.csv')
chunk_size = 100000
queryTemp = []
query = pd.DataFrame()
result = pd.DataFrame()

# Processing query by chunks
for chunk in pd.read_csv(file, chunksize=chunk_size, iterator=True, low_memory=False):
    query = chunk['cadenaComercial'].drop_duplicates(keep='first')
    queryTemp.append(query)

# Fit result to show
query = pd.concat(queryTemp).drop_duplicates()
result = query.to_frame()
result.sort_values(by='cadenaComercial', ascending=True, inplace=True)

# Result output file
result.to_csv('result_q1.csv', index=False)

# Memory use of each column along with the index
print(result.memory_usage(index = True))
```



cadenaComercial
7 ELEVEN
ABARROTERA DE BAJA CALIFORNIA
ABARROTERA DE TLAXCALA
ABARROTERA GUADALUPANA (FRUTAS)
ABARROTERA MONTERREY
ABARROTERA SANCHEZ
ABARROTES APIZACO
ABARROTES ARTES
ABARROTES LA VIOLETA
ABARROTES MEXICO
ABARROTES SUPER CABRERA CLASS
ABARROTES SUPER RIVERA
ABARROTES VERO
....

Read file → FOR conditional → By chunks

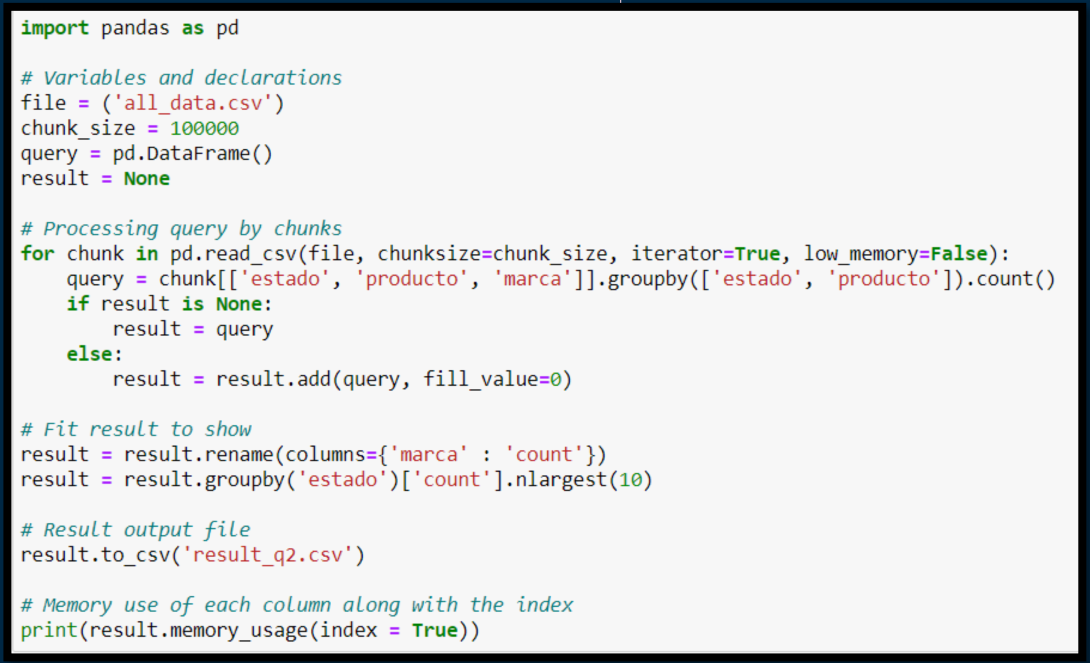
Column 'cadenaComercial'

- Drop_duplicates() → Keep='First'
- Append()
- Concat()
- Sort_values()
- To_csv()
- Memory_usage()

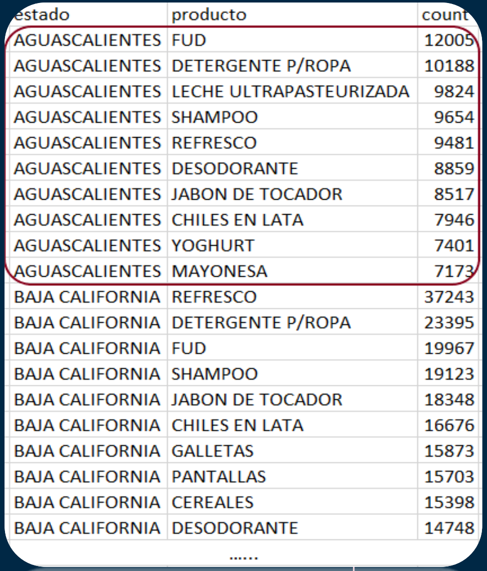
Analysis

What are the top 10
monitored products by State?

02



```python
import pandas as pd

# Variables and declarations
file = ('all_data.csv')
chunk_size = 100000
query = pd.DataFrame()
result = None

# Processing query by chunks
for chunk in pd.read_csv(file, chunksize=chunk_size, iterator=True, low_memory=False):
    query = chunk[['estado', 'producto', 'marca']].groupby(['estado', 'producto']).count()
    if result is None:
        result = query
    else:
        result = result.add(query, fill_value=0)

# Fit result to show
result = result.rename(columns={'marca' : 'count'})
result = result.groupby('estado')['count'].nlargest(10)

# Result output file
result.to_csv('result_q2.csv')

# Memory use of each column along with the index
print(result.memory_usage(index = True))
```



estado	producto	count
AGUASCALIENTES	FUD	12005
AGUASCALIENTES	DETERGENTE P/ROPA	10188
AGUASCALIENTES	LECHE ULTRAPASTEURIZADA	9824
AGUASCALIENTES	SHAMPOO	9654
AGUASCALIENTES	REFRESCO	9481
AGUASCALIENTES	DESODORANTE	8859
AGUASCALIENTES	JABON DE TOCADOR	8517
AGUASCALIENTES	CHILES EN LATA	7946
AGUASCALIENTES	YOGHURT	7401
AGUASCALIENTES	MAYONESA	7173
BAJA CALIFORNIA	REFRESCO	37243
BAJA CALIFORNIA	DETERGENTE P/ROPA	23395
BAJA CALIFORNIA	FUD	19967
BAJA CALIFORNIA	SHAMPOO	19123
BAJA CALIFORNIA	JABON DE TOCADOR	18348
BAJA CALIFORNIA	CHILES EN LATA	16676
BAJA CALIFORNIA	GALLETAS	15873
BAJA CALIFORNIA	PANTALLAS	15703
BAJA CALIFORNIA	CEREALES	15398
BAJA CALIFORNIA	DESODORANTE	14748
	......	

Read file → FOR conditional → By chunks

Columns ‘estado’, ‘producto’

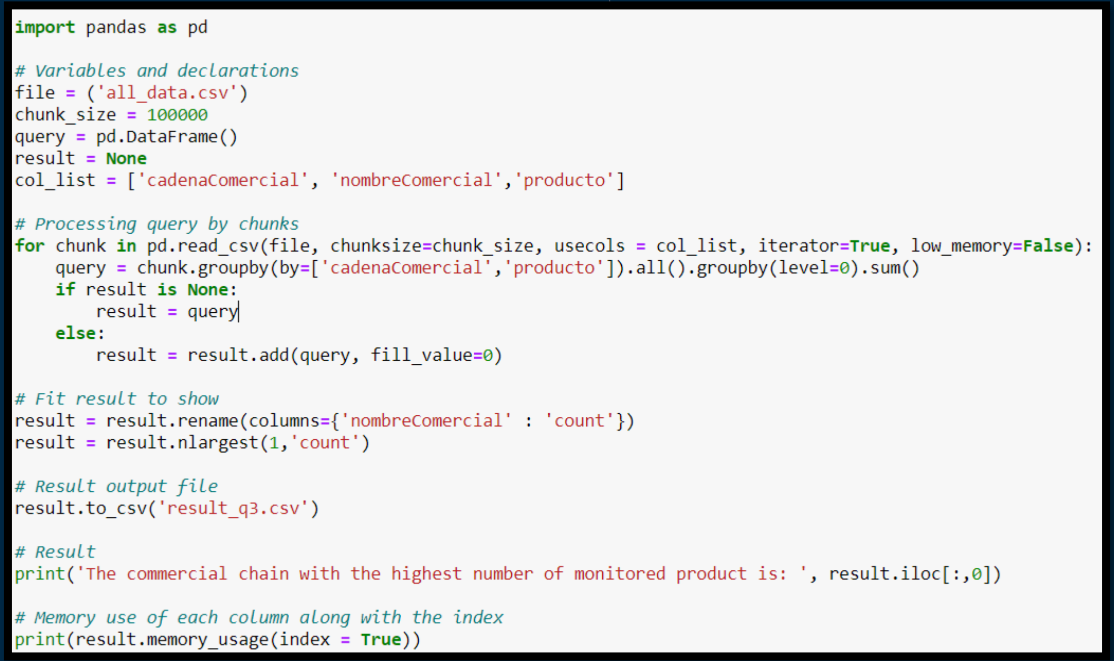
No matter its presentation, brand or in which commercial chain are sold.

- Groupby()
- IF conditional (new value, Add())
- Nlargest()
- To_csv()
- Memory_usage()

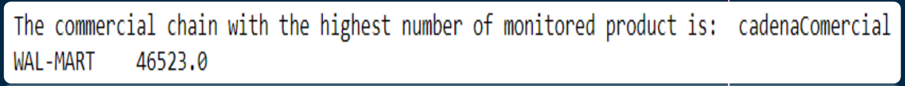
Analysis

03

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



```python
import pandas as pd

# Variables and declarations
file = ('all_data.csv')
chunk_size = 100000
query = pd.DataFrame()
result = None
col_list = ['cadenaComercial', 'nombreComercial','producto']

# Processing query by chunks
for chunk in pd.read_csv(file, chunksize=chunk_size, usecols = col_list, iterator=True, low_memory=False):
    query = chunk.groupby(by=['cadenaComercial','producto']).all().groupby(level=0).sum()
    if result is None:
        result = query
    else:
        result = result.add(query, fill_value=0)

# Fit result to show
result = result.rename(columns={'nombreComercial' : 'count'})
result = result.nlargest(1,'count')

# Result output file
result.to_csv('result_q3.csv')

# Result
print('The commercial chain with the highest number of monitored product is: ', result.iloc[:,0])

# Memory use of each column along with the index
print(result.memory_usage(index = True))
```



```
The commercial chain with the highest number of monitored product is:  cadenaComercial
WAL-MART    46523.0
```

Read file → FOR conditional → By chunks

Columns ‘cadenaComercial’, ‘producto’

No matter if the commercial chain is in one or in other state, or which branch office is, neither which branch or presentation have the products.

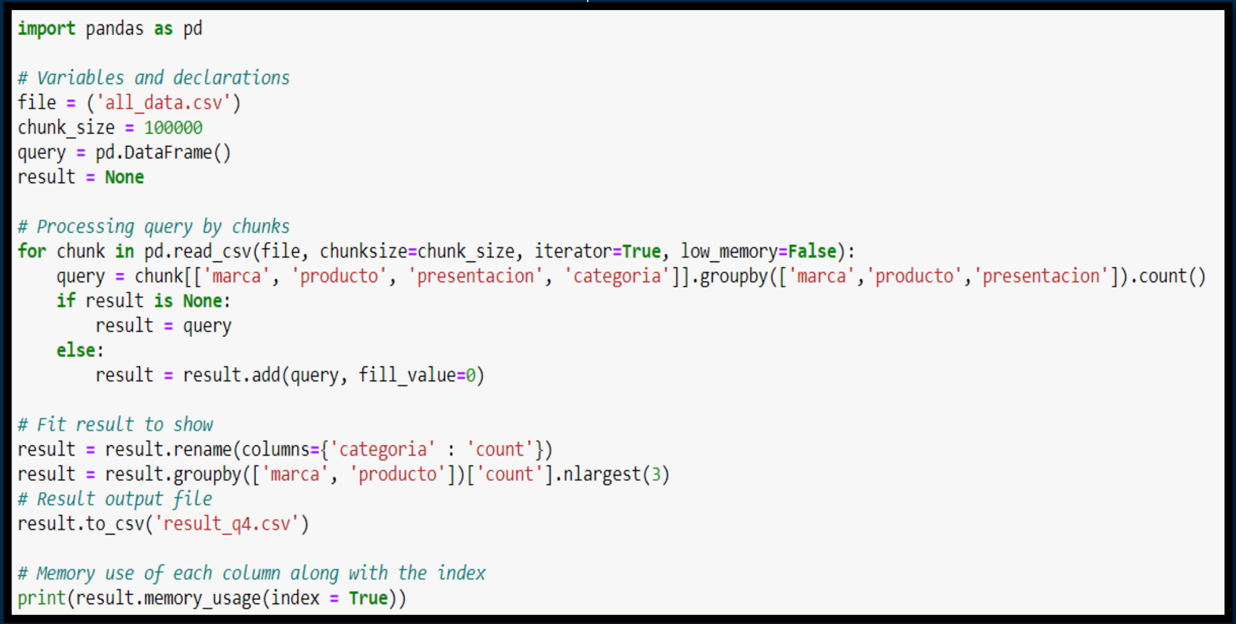
- Groupby()
- Sum()
- IF conditional (new value, Add())
- Nlargest()
- To_csv(), print()
- Memory_usage()

Analysis

Use the data to find an interesting fact

04

Top 3 leading products by trademark



```python
import pandas as pd

# Variables and declarations
file = ('all_data.csv')
chunk_size = 100000
query = pd.DataFrame()
result = None

# Processing query by chunks
for chunk in pd.read_csv(file, chunksize=chunk_size, iterator=True, low_memory=False):
    query = chunk[['marca', 'producto', 'presentacion', 'categoria']].groupby(['marca','producto','presentacion']).count()
    if result is None:
        result = query
    else:
        result = result.add(query, fill_value=0)

# Fit result to show
result = result.rename(columns={'categoria' : 'count'})
result = result.groupby(['marca', 'producto'])['count'].nlargest(3)
# Result output file
result.to_csv('result_q4.csv')

# Memory use of each column along with the index
print(result.memory_usage(index = True))
```

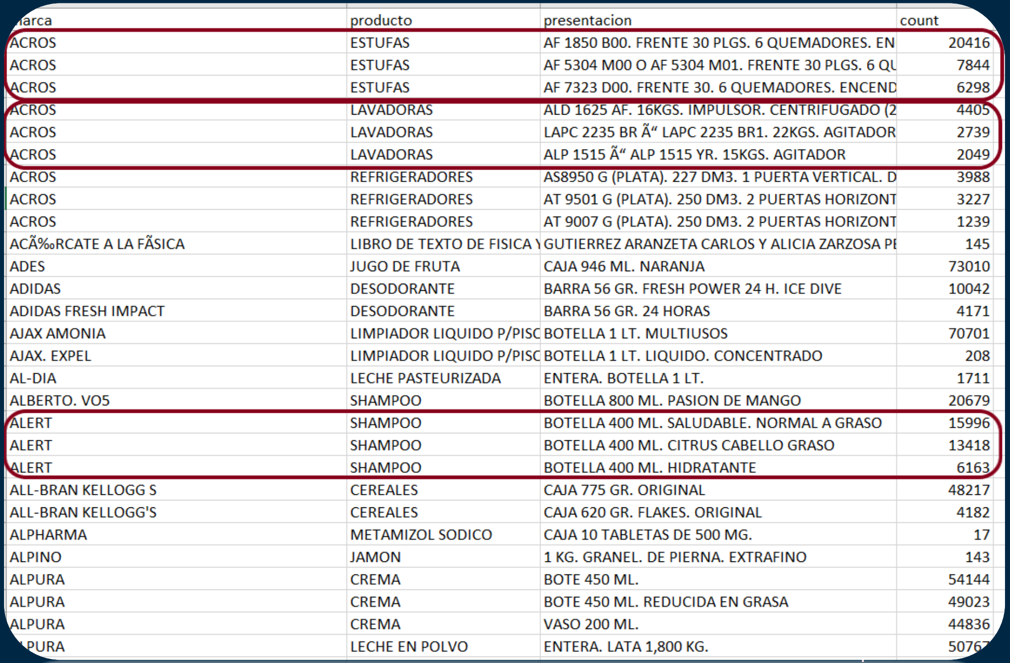
Read file → FOR conditional → By chunks

Columns ‘marca’, ‘producto’, ‘presentacion’

- Groupby()
- Count()
- IF conditional (new value, Add())
- Nlargest()
- To_csv()
- Memory_usage()

AF 1850 B00
→ 20,416 registers

ALD 1625A AF
→ 4,405 registers



arca	producto	presentacion	count
ACROS	ESTUFAS	AF 1850 B00. FRENTE 30 PLGS. 6 QUEMADORES. EN	20416
ACROS	ESTUFAS	AF 5304 M00 O AF 5304 M01. FRENTE 30 PLGS. 6 QU	7844
ACROS	ESTUFAS	AF 7323 D00. FRENTE 30. 6 QUEMADORES. ENCEND	6298
ACROS	LAVADORAS	ALD 1625 AF. 16KGS. IMPULSOR. CENTRIFUGADO (2	4405
ACROS	LAVADORAS	LAPC 2235 BR Ã“ LAPC 2235 BR1. 22KGS. AGITADOR	2739
ACROS	LAVADORAS	ALP 1515 Ã“ ALP 1515 YR. 15KGS. AGITADOR	2049
ACROS	REFRIGERADORES	AS8950 G (PLATA). 227 DM3. 1 PUERTA VERTICAL. D	3988
ACROS	REFRIGERADORES	AT 9501 G (PLATA). 250 DM3. 2 PUERTAS HORIZONT	3227
ACROS	REFRIGERADORES	AT 9007 G (PLATA). 250 DM3. 2 PUERTAS HORIZONT	1239
ACÃ‰RCATE A LA FÃSICA	LIBRO DE TEXTO DE FISICA	GUTIERREZ ARANZETA CARLOS Y ALICIA ZARZOSA PE	145
ADES	JUGO DE FRUTA	CAJA 946 ML. NARANJA	73010
ADIDAS	DESODORANTE	BARRA 56 GR. FRESH POWER 24 H. ICE DIVE	10042
ADIDAS FRESH IMPACT	DESODORANTE	BARRA 56 GR. 24 HORAS	4171
AJAX AMONIA	LIMPIADOR LIQUIDO P/PISC	BOTELLA 1 LT. MULTIUSOS	70701
AJAX. EXPEL	LIMPIADOR LIQUIDO P/PISC	BOTELLA 1 LT. LIQUIDO. CONCENTRADO	208
AL-DIA	LECHE PASTEURIZADA	ENTERA. BOTELLA 1 LT.	1711
ALBERTO. VO5	SHAMPOO	BOTELLA 800 ML. PASION DE MANGO	20679
ALERT	SHAMPOO	BOTELLA 400 ML. SALUDABLE. NORMAL A GRASO	15996
ALERT	SHAMPOO	BOTELLA 400 ML. CITRUS CABELLO GRASO	13418
ALERT	SHAMPOO	BOTELLA 400 ML. HIDRATANTE	6163
ALL-BRAN KELLOGG S	CEREALES	CAJA 775 GR. ORIGINAL	48217
ALL-BRAN KELLOGG'S	CEREALES	CAJA 620 GR. FLAKES. ORIGINAL	4182
ALPHARMA	METAMIZOL SODICO	CAJA 10 TABLETAS DE 500 MG.	17
ALPINO	JAMON	1 KG. GRANEL. DE PIERNA. EXTRAFINO	143
ALPURA	CREMA	BOTE 450 ML.	54144
ALPURA	CREMA	BOTE 450 ML. REDUCIDA EN GRASA	49023
ALPURA	CREMA	VASO 200 ML.	44836
PURA	LECHE EN POLVO	ENTERA. LATA 1,800 KG.	5076

Analysis

What are the lessons learned from this exercise?

05

Lessons learned

01

Get acquainted with the data

- ○ *what are you studying?*
- ○ *what useful information want to find?*

Amount of information and the way I compiled into a query

02

Understand the different techniques for data analysis and its methods

- *Large data →Chunks*
- *Dataframes, arrays*
- *Methods*
- *….other functions and libraries*

03

Find connections

Generate solutions and useful information

Analysis

06

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

Facing the challengue

DASK

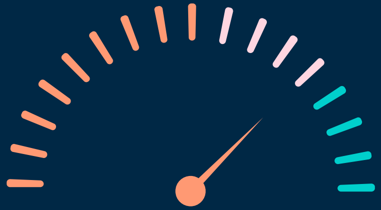
Dask API
Parallel computing library
dask.dataframe
Multiple threads to
process data in parallel.

Distributed file systems

Distributed file systems like Hadoop and Spark

Frameworks to process data in parallel across clusters on single computer.

Cloud compute services



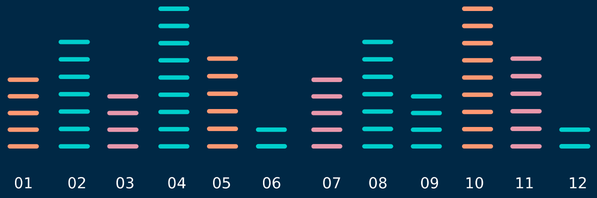
Use compute services of any of Cloud providers

Get the resources needed to perform queries more efficiently

–CONCLUSION

Exploratory data analysis (EDA), as an approach to analyze data to summarize and deepen into its structure and main characteristics, allows Analysts know the data and how to work with it. This could be the very fist step to build more complex analysis in order to make decisions and built predictions.

So, taking in mind Pandas, as a very powerful and easy to use tool, to face this first approach, it is a great advantage in order to get a quality big picture.



Do you have any questions?

aleks_moreno2@hotmail.com
+961 668 6265
https://www.linkedin.com/in/alejandramoreno-it/

THANKS

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik
Please keep this slide for attribution