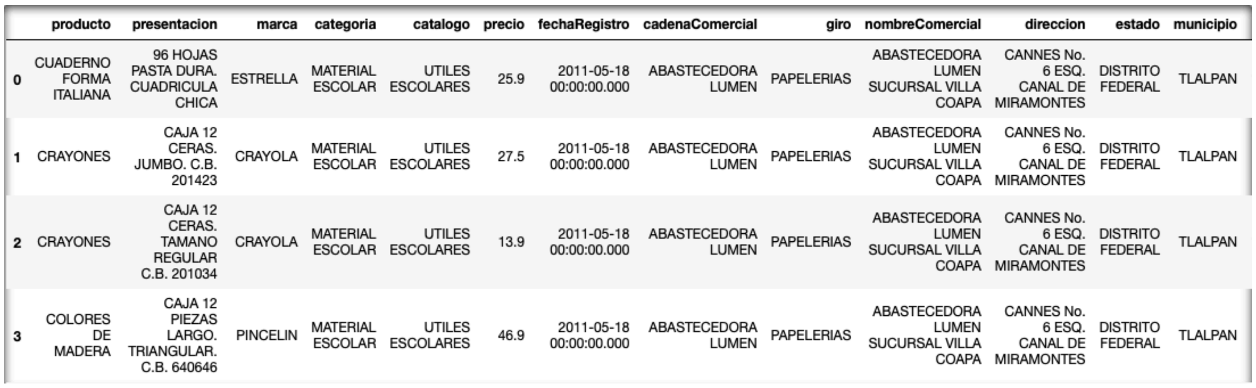
Technical challenge, Jorge Martı́nez Vega

Exploratory Analysis

Summary

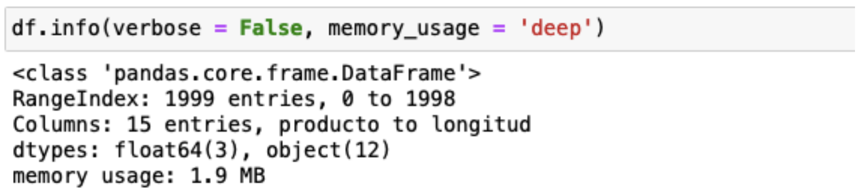
For this analysis the data base being used to solve the challenge comes from PROFECO (Mexican Consumer Protection Agency), it is composed of over 62 million “rows”, and 15 columns, the data has a structure as shown in the following image.



	producto	presentacion	marca	categoria	catalogo	precio	fechaRegistro	cadenaComercial	giro	nombreComercial	direccion	estado	municipio
0	CUADERNO FORMA ITALIANA	96 HOJAS PASTA DURA. CUADRICULA CHICA	ESTRELLA	MATERIAL ESCOLAR	UTILES ESCOLARES	25.9	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
1	CRAYONES	CAJA 12 CERAS. JUMBO. C.B. 201423	CRAYOLA	MATERIAL ESCOLAR	UTILES ESCOLARES	27.5	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
2	CRAYONES	CAJA 12 CERAS. TAMANO REGULAR C.B. 201034	CRAYOLA	MATERIAL ESCOLAR	UTILES ESCOLARES	13.9	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
3	COLORES DE MADERA	CAJA 12 PIEZAS LARGO. TRIANGULAR. C.B. 640646	PINCELIN	MATERIAL ESCOLAR	UTILES ESCOLARES	46.9	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

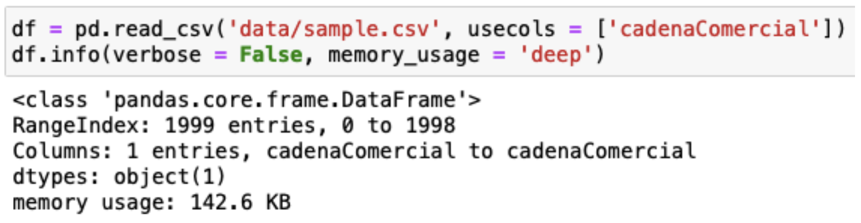
Challenge

Due to the data being so large, it isn’t convenient to just load the entire data into a pandas data frame, for efficiency purposes, it is better if only the columns that are going to be used are stored in the data frame created, a test can be done with the sample data to give an idea of how much smaller the data frame is, compared to when all the columns are included.



```
df.info(verbose = False, memory_usage = 'deep')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1999 entries, 0 to 1998
Columns: 15 entries, producto to longitud
dtypes: float64(3), object(12)
memory usage: 1.9 MB
```



```
df = pd.read_csv('data/sample.csv', usecols = ['cadenaComercial'])
df.info(verbose = False, memory_usage = 'deep')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1999 entries, 0 to 1998
Columns: 1 entries, cadenaComercial to cadenaComercial
dtypes: object(1)
memory usage: 142.6 KB
```

It can be seen that there is a significant reduction, of around *93%*, in the *memory usage*, from **1.9 MB** down to **142.6 KB**, this will have a huge impact for creating data frames to analize data in this project, mainly due to the volume of the data, over **20 GB.**

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

In order to answer, it is necessary to count how manny distinct values there are in column **cadenaComercial**, for this it will be enough to import only that column into the data frame.

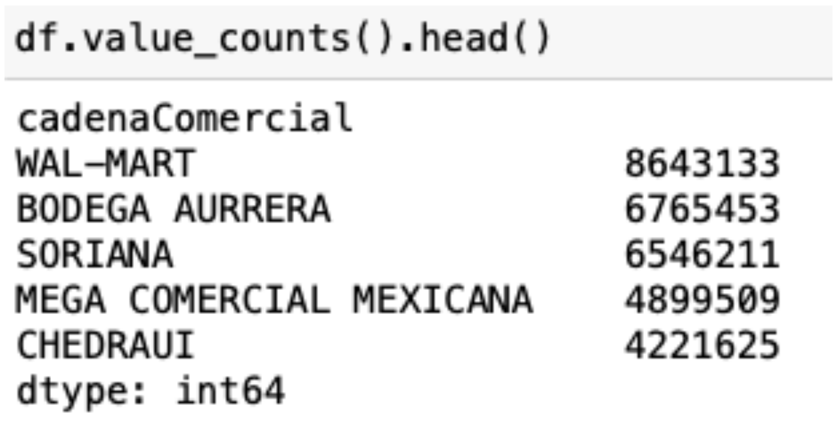
For the counting, pandas includes a method **nunique()** which will count the distinct values on an specified column, After counting, the result is 705, hence **there are 705 commercial chains monitored in the database.**

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

To further reduce *memory usage*, categorical data type can be used, that way the data frame will use even less memory.

For this question each row in the data set will be considered a **monitored product**, this is because even if two rows have the same value in the column **producto** they might have a different price or vendor, therefore they are monitored separately.

Having that in mind, counting the occurrences of each commercial chain and taking the top one will answer the question.



```
df.value_counts().head()
```

```
cadenaComercial
WAL-MART                   8643133
BODEGA AURRERA             6765453
SORIANA                    6546211
MEGA COMERCIAL MEXICANA    4899509
CHEDRAUI                   4221625
dtype: int64
```

From the output, **WAL-MART** has the highest number of monitored products with a total of **8643133**, followed by *BODEGA AURRERA* with *6765453* records.

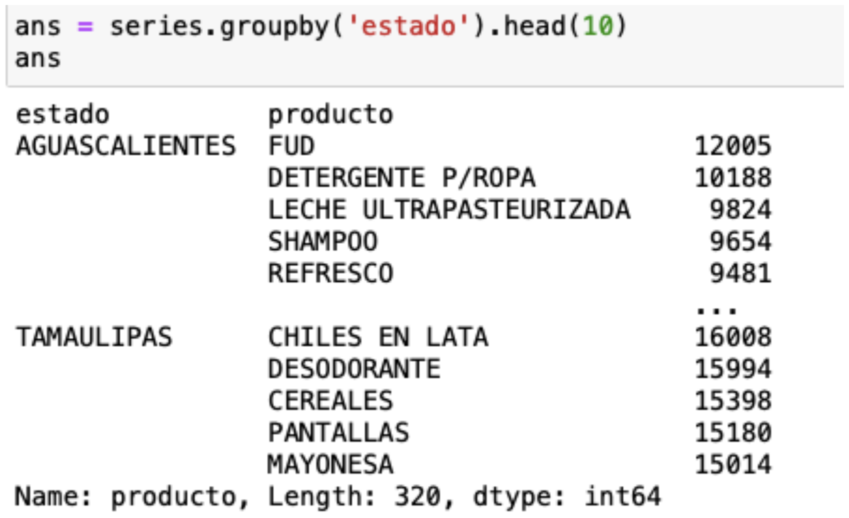
What are the top 10 monitored products by State?

This time two columns are needed in order to answer the question, these being **producto** and **estado**, the column **estado** is supposed to have at most **32** different values, so it is the perfect candidate to use the categorical data type.

While exploring the newly created data frame, there are actually more than 32 different values, with a further investigation, the cause was that some of the rows where corrupted, they weren't **null** but the value wasn't giving any valuable information, for example, the value for column **estado** was the string 'estado'.

The index of the rows with the '*estado*' value can be found and with that those lines can be skipped for future data frames in the challenge.

This *clean* data frame will be used to find the answer, to start, pandas includes the method **groupby()** that can be combined with **value_counts()** to first group by a column and then get the count of unique values of anoter, this will return a Series object with the counts of products by state, this object can be grouped by the column **estado** and with the use of the method **head()** the top ten products can be fetched.



```
ans = series.groupby('estado').head(10)
ans
```

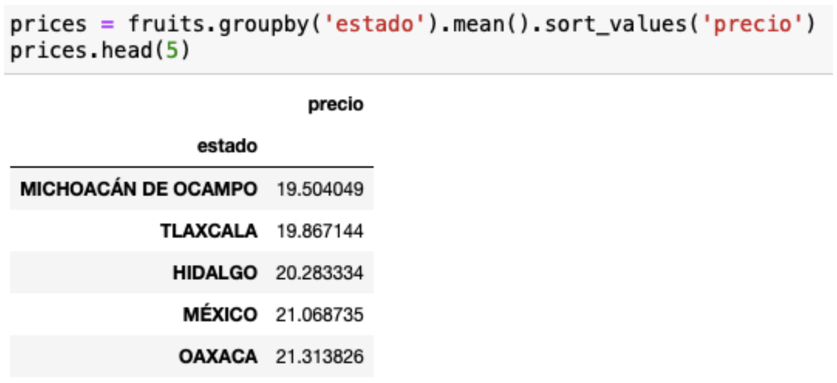
```
estado         producto
AGUASCALIENTES FUD                        12005
               DETERGENTE P/ROPA          10188
               LECHE ULTRAPASTEURIZADA     9824
               SHAMPOO                     9654
               REFRESCO                    9481
                                           ...
TAMAULIPAS     CHILES EN LATA             16008
               DESODORANTE                15994
               CEREALES                   15398
               PANTALLAS                  15180
               MAYONESA                   15014
Name: producto, Length: 320, dtype: int64
```

All the results corresponding to each state can be found in the file *topten.csv* inside the forked repository.

Interesting Fact

One of the things that is of general interest is the price of fruits and legumes, with this premise, a question araise, **which state has the lowest price on average for this kind of products?**, to answer this, three columns of the data set will be used, **catalogo**, **precio** and **estado.**

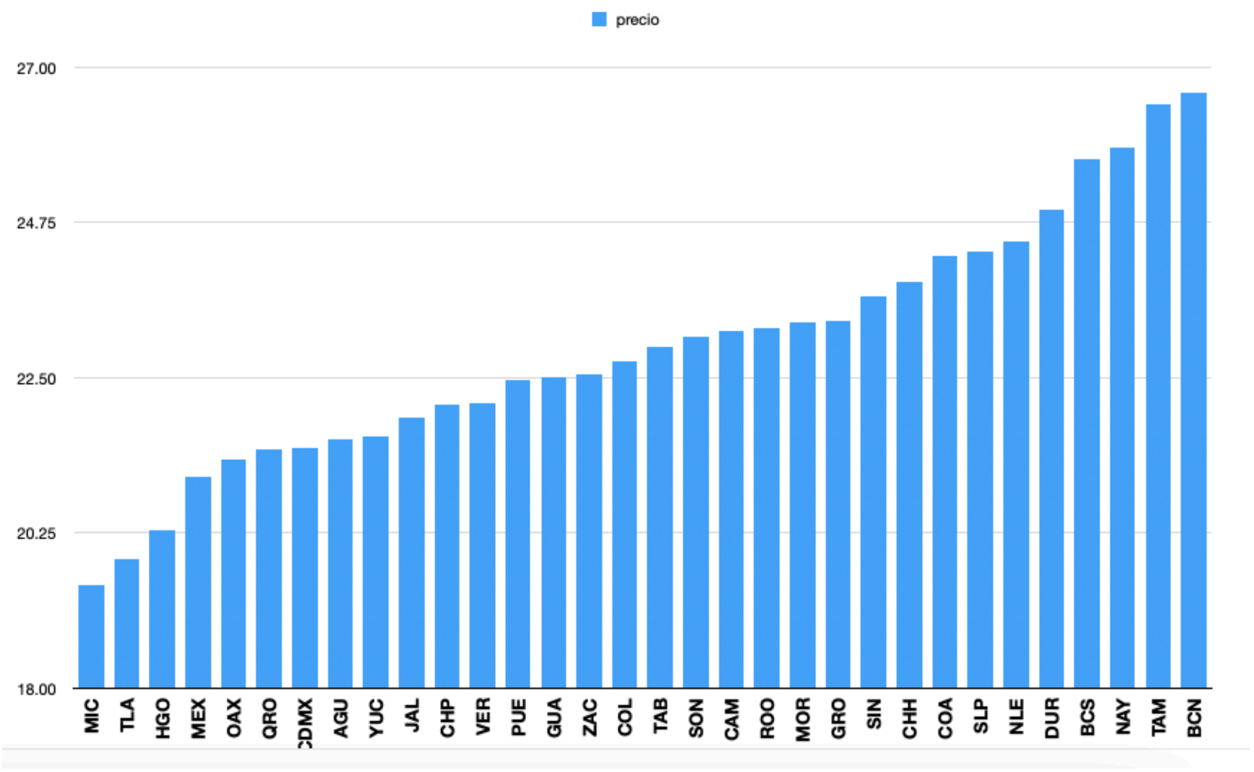
A new data frame called **fruits** will be defined, it will only contain the rows that have *FRUTAS Y LEGUMBRES* as value on the column **catalogo** This new data frame is first grouped by state, the average price for all products is calculated and at the end the result is sorted.



```
prices = fruits.groupby('estado').mean().sort_values('precio')
prices.head(5)
```

	precio
estado	
MICHOACÁN DE OCAMPO	19.504049
TLAXCALA	19.867144
HIDALGO	20.283334
MÉXICO	21.068735
OAXACA	21.313826

Using this data, a plot can be made to see more easily the states that have the lower and higher prices



The interesting fact found is that, according to the data, the states with the highest average price are mainly the ones in the north of the country, my hypothesis is that this kinds of products are produced and harvested on the center and south part of the country, this will imply that they need to be shipped to the north states carrying an extra fee because of that.

Lessons learn from this exercise

- The first thing I learned from was how to handle “big” sets of data, of around 20 GB, so far I had used pandas for analysis on much smaller datasets, small enough to load them into memory without a problem, but for this challenge a different path must be taken, I learn how to make a better use of the memory by optimizing what’s loaded, using a categorical data type was one of the things that helped on that regard.
- I have to clean the data as good as possible, when I was trying to solve the question about the top 10 most monitored problems I tested my code with the sample data, it worked great so I tried with the whole data, I was getting an issue not present when testing with the sample data, after some digging I found that the complete data set had a few rows with problems, so what I did was create a new sample data with some rows similar to the ones found in the data and tested the code, this led to the same issue I was having so I knew that was the problem, I skipped the mentioned lines while defining the data frame and it solve the problem.

Other way to approach the problem

Since my main problem was the size of the data, one solution could be create or recreate a data base, the replicated data, by building a data base with a python script we can create multiple tables with foreign keys so there’s no replicated data, by doing so we can use SQL queries to obtain the same results for the questions of the challenge, and unlike in a data frame from pandas, where we needed to load only part of the data, we can use the entire data set that is now contained in our new created data base.