Análisis de datos para la toma de una decisión

Paso 1: Importar los paquetes requeridos

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

Paso 2: Lectura de datos

In [2]:

```
df = pd.read_csv("Supermarket Transactions.csv")
df
```

Out[2]:

	Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income
0	1	18/12/2011	7223	F	S	Y	2	30K - 50K
1	2	20/12/2011	7841	M	M	Y	5	70K - 90K
2	3	21/12/2011	8374	F	M	N	2	50K - 70K
3	4	21/12/2011	9619	M	M	Y	3	30K - 50K
4	5	22/12/2011	1900	F	S	Y	3	130K - 150K
...	...	...	...	...	...	...	...	...
14054	14055	29/12/2013	9102	F	M	Y	2	10K - 30K
14055	14056	29/12/2013	4822	F	M	Y	3	10K - 30K
14056	14057	31/12/2013	250	M	S	Y	1	30K - 50K
14057	14058	31/12/2013	6153	F	S	N	4	50K - 70K
14058	14059	31/12/2013	3656	M	S	N	3	50K - 70K

14059 rows × 16 columns

In [3]:

```
# ¿Y qué es la tabla anterior?.. ¡Un DataFrame de Pandas!

print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [4]:

```
# Note que la primera columna no estaba en el archivo de excel. Esto es una indexación
 automática que hace Pandas.
# Para quitarla:

df = df.set_index("Transaction")
df
```

Out[4]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
1	18/12/2011	7223	F	S	Y	2	30K - 50K	Lo Angele
2	20/12/2011	7841	M	M	Y	5	70K - 90K	Lo Angele
3	21/12/2011	8374	F	M	N	2	50K - 70K	Bremerto
4	21/12/2011	9619	M	M	Y	3	30K - 50K	Portlan
5	22/12/2011	1900	F	S	Y	3	130K - 150K	Beverl Hill
...	...	...	...	...	...	...	...	.
14055	29/12/2013	9102	F	M	Y	2	10K - 30K	Bremerto
14056	29/12/2013	4822	F	M	Y	3	10K - 30K	Wall Wall
14057	31/12/2013	250	M	S	Y	1	30K - 50K	Portlan
14058	31/12/2013	6153	F	S	N	4	50K - 70K	Spokan
14059	31/12/2013	3656	M	S	N	3	50K - 70K	Portlan

14059 rows × 15 columns

Paso 3: Algunas operacione básicas con Dataframe

In [5]:

```
# Resumen estadístico... ¿De las variables cuantitativas? !Ojo!

print("Resúsmen estadístico de variables cuantitativas")
df.describe()
```

```
Resúsmen estadístico de variables cuantitativas
```

Out[5]:

	Customer ID	Children	Units Sold	Revenue
count	14059.000000	14059.000000	14059.000000	14059.000000
mean	5116.902127	2.530336	4.080589	13.004512
std	2920.755202	1.491852	1.174421	8.215543
min	3.000000	0.000000	1.000000	0.530000
25%	2549.000000	1.000000	3.000000	6.840000
50%	5060.000000	3.000000	4.000000	11.250000
75%	7633.000000	4.000000	5.000000	17.370000
max	10280.000000	5.000000	8.000000	56.700000

In [6]:

```
# Otras operaciones que se pueden hacer con el DaraFrame:
```

In [7]:

```
# Mostrar solo una columna:

print("Mostrar solo una columna")
df["Purchase Date"]
```

```
Mostrar solo una columna
```

Out[7]:

```
Transaction
1         18/12/2011
2         20/12/2011
3         21/12/2011
4         21/12/2011
5         22/12/2011
             ...
14055     29/12/2013
14056     29/12/2013
14057     31/12/2013
14058     31/12/2013
14059     31/12/2013
Name: Purchase Date, Length: 14059, dtype: object
```

In [8]:

```
# Mostrar las primeras cuatro filas:

print("Mostrar las primeras cuatro filas")
df.head(4)
```

```
Mostrar las primeras cuatro filas
```

Out[8]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
1	18/12/2011	7223	F	S	Y	2	30K - 50K	Lo Angele
2	20/12/2011	7841	M	M	Y	5	70K - 90K	Lo Angele
3	21/12/2011	8374	F	M	N	2	50K - 70K	Bremerto
4	21/12/2011	9619	M	M	Y	3	30K - 50K	Portlan

In [9]:

```
# Mostrar las últimas dos filas:

print("Mostrar las ultimas dos filas")
df.tail(2)
```

```
Mostrar las ultimas dos filas
```

Out[9]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
14058	31/12/2013	6153	F	S	N	4	50K - 70K	Spokane
14059	31/12/2013	3656	M	S	N	3	50K - 70K	Portland

In [10]:

```
# Mostrar los índices

print("Índices")
df.index
```

```
Índices
```

Out[10]:

```
Int64Index([    1,     2,     3,     4,     5,     6,     7,     8,     9,
               10,
            ...
            14050, 14051, 14052, 14053, 14054, 14055, 14056, 14057, 14058,
            14059],
           dtype='int64', name='Transaction', length=14059)
```

In [11]:

```
# Mostrar los nombres de las columnas

print("Nombres de las columnas")
df.columns
```

```
Nombres de las columnas
```

Out[11]:

```
Index(['Purchase Date', 'Customer ID', 'Gender', 'Marital Status', 'Homeow
ner',
       'Children', 'Annual Income', 'City', 'State or Province', 'Countr
y',
       'Product Family', 'Product Department', 'Product Category',
       'Units Sold', 'Revenue'],
      dtype='object')
```

In [12]:

```
# Organizar el DataFrame por una columna en particular

print("Organizar por género en orden alfabético")
df.sort_values(by='Gender',ascending=True)
```

```
Organizar por género en orden alfabético
```

Out[12]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
1	18/12/2011	7223	F	S	Y	2	30K - 50K	Lo Angele
5879	14/02/2013	6433	F	S	Y	1	10K - 30K	Seattl
5878	14/02/2013	8697	F	S	Y	5	30K - 50K	Hidalg
5877	14/02/2013	7720	F	M	Y	3	130K - 150K	Tacom
11115	26/08/2013	6603	F	M	N	5	30K - 50K	Merid
...	...	...	...	...	...	...	...	.
6217	25/02/2013	7308	M	M	Y	2	130K - 150K	Sa Andre
6220	25/02/2013	645	M	S	N	1	50K - 70K	Portlan
6222	26/02/2013	2510	M	S	N	1	10K - 30K	Hidalg
6269	27/02/2013	3881	M	S	Y	1	30K - 50K	Vancouve
14059	31/12/2013	3656	M	S	N	3	50K - 70K	Portlan

14059 rows × 15 columns

In [13]:

```
# Mostrar solo los registros que cumplan con una condición

print("Solo los que tengan revenue mayor a 35")
df[df.Revenue > 35]
```

```
Solo los que tengan revenue mayor a 35
```

Out[13]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
56	05/01/2012	7733	M	S	Y	5	10K - 30K	Salem
77	09/01/2012	4925	F	S	Y	0	50K - 70K	Spokane
84	10/01/2012	7823	F	M	Y	5	10K - 30K	Los Angeles
87	11/01/2012	7493	F	M	Y	3	110K - 130K	San Diego
187	22/01/2012	3001	F	M	N	1	30K - 50K	Salem
...	...	...	...	...	...	...	...	...
13834	06/12/2013	2001	F	M	Y	4	30K - 50K	Los Angeles
13838	06/12/2013	5284	F	M	N	3	50K - 70K	Acapulco
13842	06/12/2013	208	F	M	Y	2	130K - 150K	Salem
13912	11/12/2013	3239	F	S	N	4	50K - 70K	Tacoma
14036	25/12/2013	7887	M	S	N	2	30K - 50K	Portland

269 rows × 15 columns

In [14]:

```
# Remplazar un valor (primer Customer ID, por ejemplo)

print("Reemplazar el valor 7223 por remplazo en la primera transacción")
df_remplazo = df.replace(7223,"remplazo")
df_remplazo
```

```
Reemplazar el valor 7223 por remplazo en la primera transacción
```

Out[14]:

Transaction	Purchase Date	Customer ID	Gender	Marital Status	Homeowner	Children	Annual Income	City
1	18/12/2011	remplazo	F	S	Y	2	30K - 50K	Lo Angele
2	20/12/2011	7841	M	M	Y	5	70K - 90K	Lo Angele
3	21/12/2011	8374	F	M	N	2	50K - 70K	Bremerto
4	21/12/2011	9619	M	M	Y	3	30K - 50K	Portlan
5	22/12/2011	1900	F	S	Y	3	130K - 150K	Beverl Hill
...	...	...	...	...	...	...	...	.
14055	29/12/2013	9102	F	M	Y	2	10K - 30K	Bremerto
14056	29/12/2013	4822	F	M	Y	3	10K - 30K	Wall Wall
14057	31/12/2013	250	M	S	Y	1	30K - 50K	Portlan
14058	31/12/2013	6153	F	S	N	4	50K - 70K	Spokan
14059	31/12/2013	3656	M	S	N	3	50K - 70K	Portlan

14059 rows × 15 columns

In [15]:

```
#Presentar un solo registro

print("Mostrar solo el primer registro")
df_remplazo.iloc[0]
```

```
Mostrar solo el primer registro
```

Out[15]:

```
Purchase Date          18/12/2011
Customer ID              remplazo
Gender                          F
Marital Status                  S
Homeowner                       Y
Children                        2
Annual Income           30K - 50K
City                  Los Angeles
State or Province              CA
Country                       USA
Product Family               Food
Product Department    Snack Foods
Product Category      Snack Foods
Units Sold                      5
Revenue                     27.38
Name: 1, dtype: object
```

Paso 4: Visualización

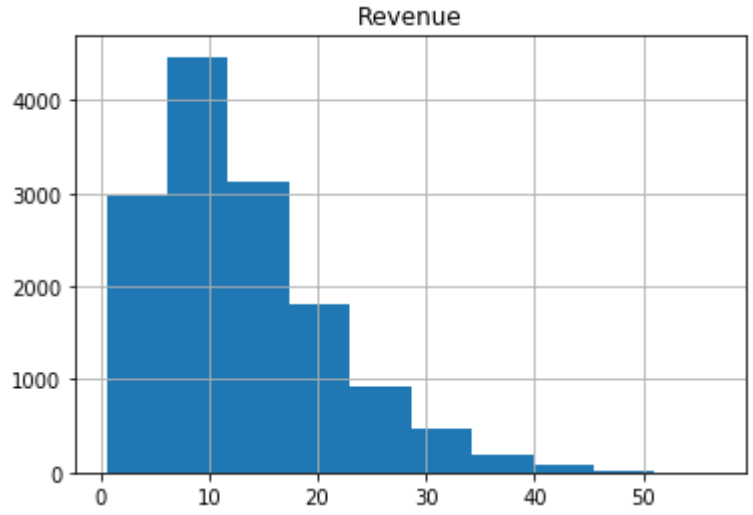
In [16]:

```
# Histogramas

pd.DataFrame.hist(df,"Revenue")
```

Out[16]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001C27D6813C
8>]],
      dtype=object)
```



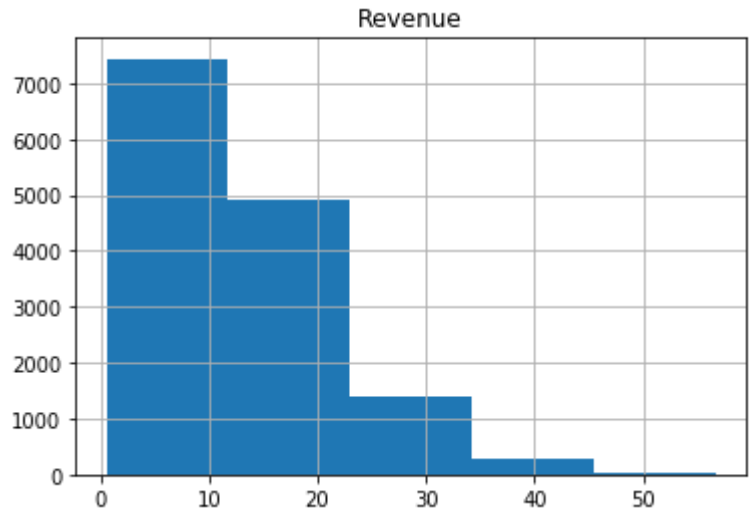
In [17]:

```
# Se puede cambiar el número de bins. ¿Cambia el análisis también?
# Bins: intervalos en el eje x

pd.DataFrame.hist(df,"Revenue",bins=5)
```

Out[17]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001C27E13E5C
8>]],
      dtype=object)
```



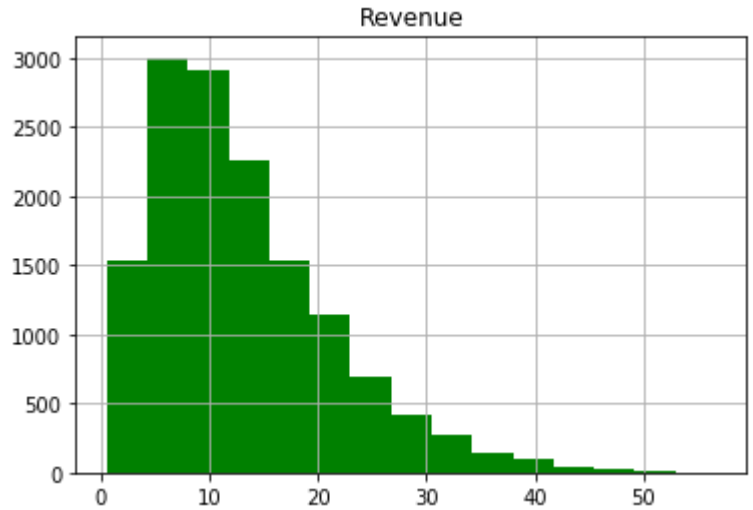
In [18]:

```
# De forma predeterminada los bins son 10...¿Qué pasará ahorá con 15?

pd.DataFrame.hist(df,"Revenue",bins=15, color="green")
```

Out[18]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001C27E1BB2C
8>]],
      dtype=object)
```



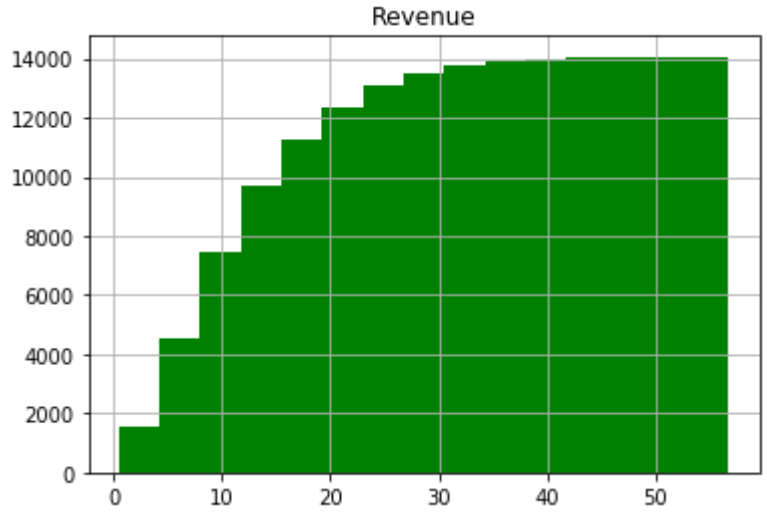
In [19]:

```
# ¿Y si "ordenamos" el histograma anterior?

pd.DataFrame.hist(df,"Revenue",bins=15,cumulative=True, color="green")
```

Out[19]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001C27E270B0
8>]],
      dtype=object)
```



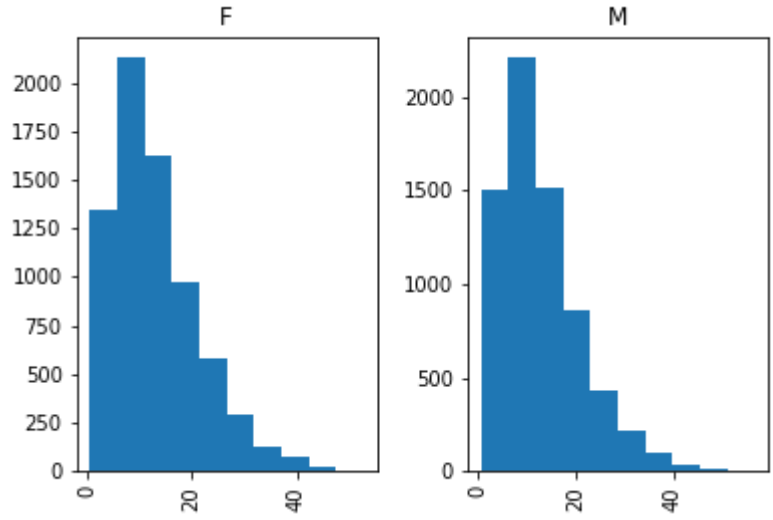
In [20]:

```
# ¡Veamos los ingresos de los hombres y las mujeres!

pd.DataFrame.hist(df, "Revenue", xlabelsize=10, ylabelsize=10, by="Gender")
```

Out[20]:

```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000001C27E2F9108
>,
       <matplotlib.axes._subplots.AxesSubplot object at 0x000001C27E32BE88
>],
      dtype=object)
```



Otros tipos de gráficos

In [21]:

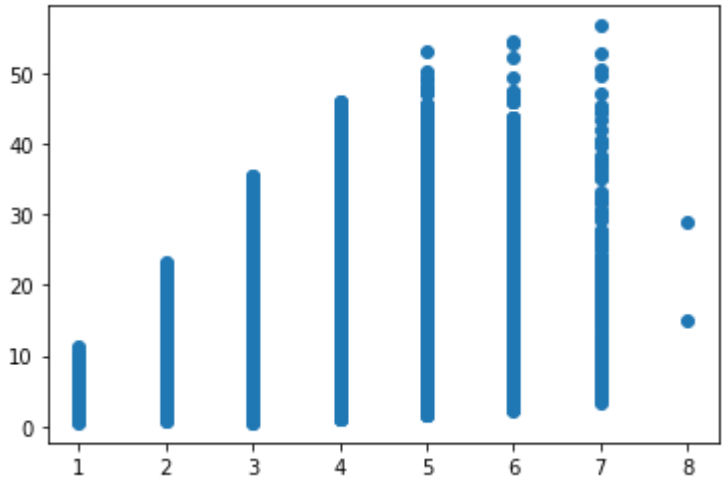
```
# Scatter plot:

y = df["Revenue"]
x = df["Units Sold"]

plt.scatter(x,y)
```

Out[21]:

```
<matplotlib.collections.PathCollection at 0x1c27e3c8d08>
```



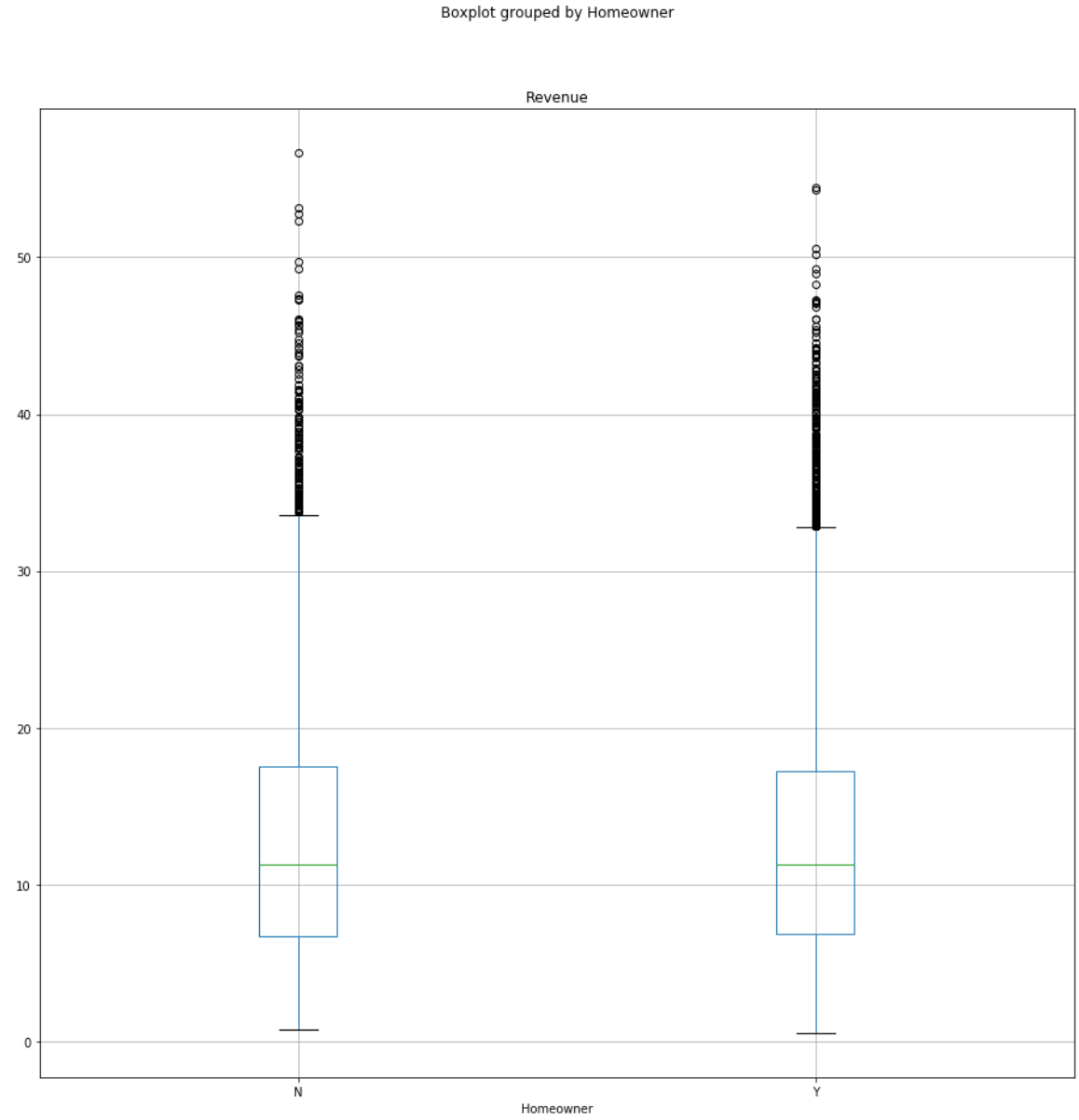
In [22]:

```
# Boxplot

df.boxplot("Revenue",figsize=(15,15),by="Homeowner")
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c27f424288>
```


Boxplot grouped by Homeowner
Revenue
50
40
30
20
10
0
N
Y
Homeowner

In [23]:

```
#  ¿Y qué podemos hacer con las variales categóricas?
```

In [24]:

```
# 1. Contemos los valores de una columna

df["Product Department"].value_counts()
```

Out[24]:

```
Produce                1994
Snack Foods            1600
Household              1420
Frozen Foods           1382
Baking Goods           1072
Canned Foods            977
Dairy                   903
Health and Hygiene      893
Deli                    699
Beverages               680
Baked Goods             425
Alcoholic Beverages     356
Snacks                  352
Starchy Foods           277
Periodicals             202
Eggs                    198
Breakfast Foods         188
Canned Products         109
Seafood                 102
Meat                     89
Checkout                 82
Carousel                 59
Name: Product Department, dtype: int64
```

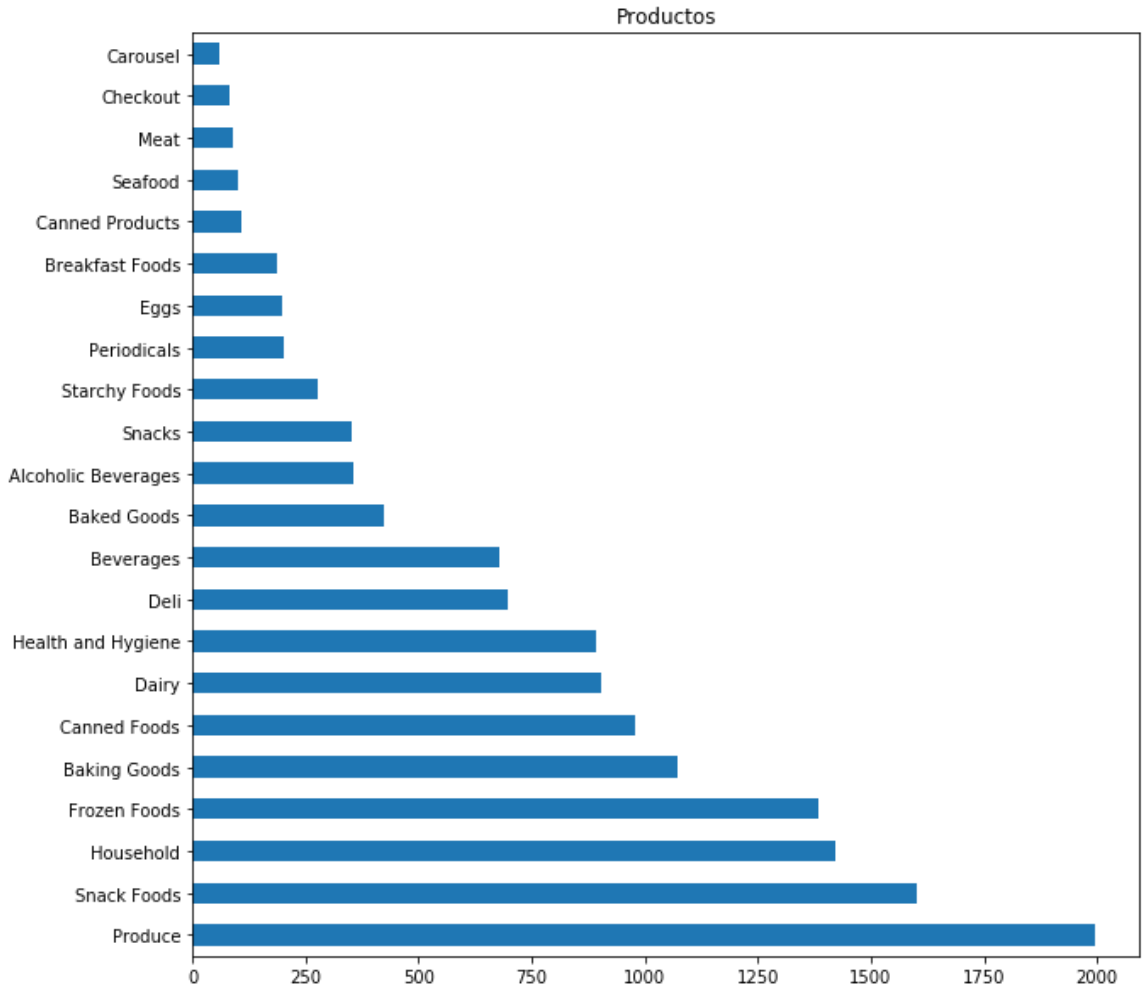
In [25]:

```
# Y ahora...¡Grafiquemos!

df["Product Department"].value_counts().plot(kind="barh",figsize=(10,10))
plt.title("Productos")
```

Out[25]:

```
Text(0.5, 1.0, 'Productos')
```



In [26]:

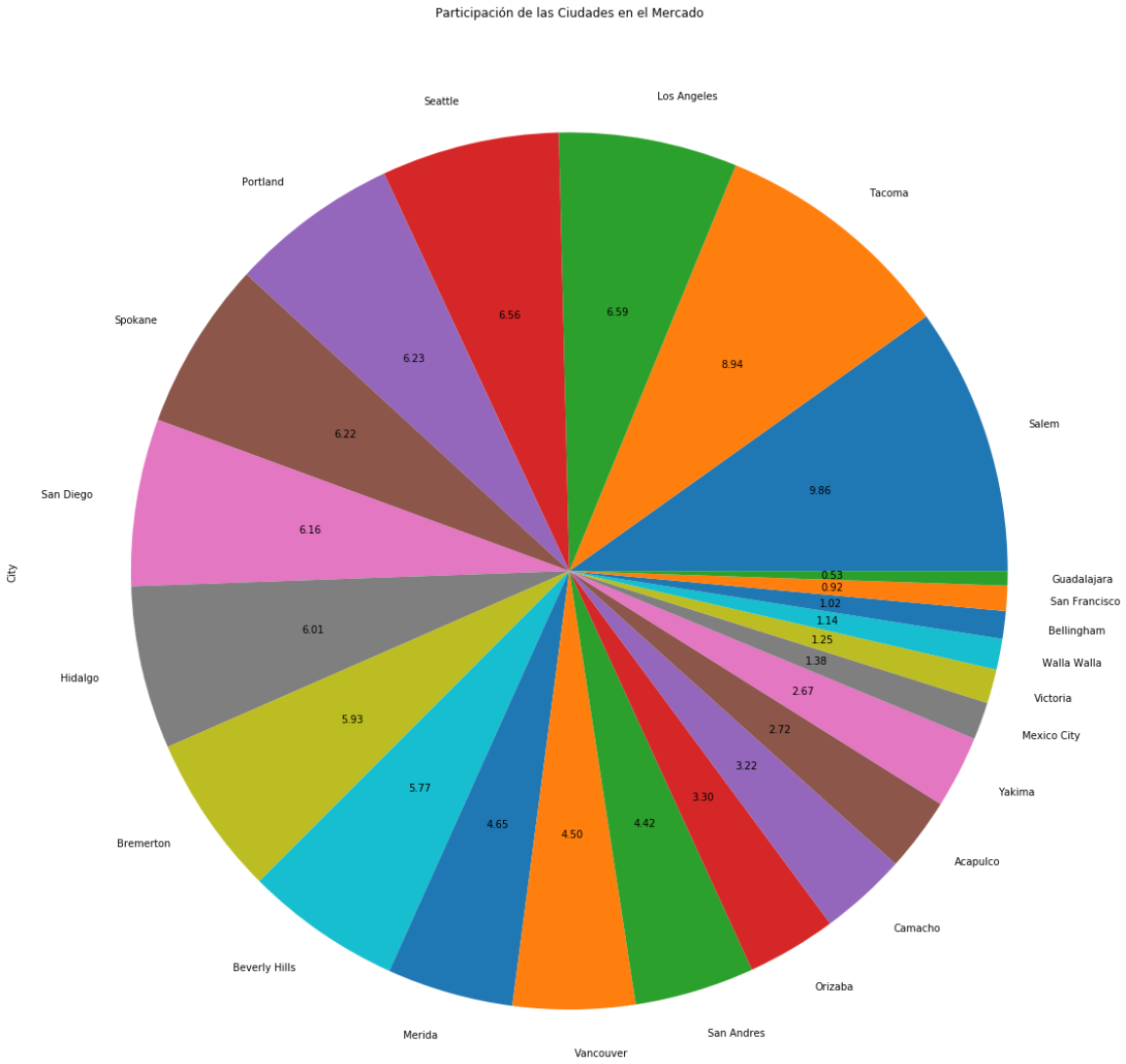
```
# ¿Y con otra variable..?

df["City"].value_counts().plot(kind="pie",figsize=(20,20),autopct="%.2f")

plt.title("Participación de las Ciudades en el Mercado")
```

Out[26]:

```
Text(0.5, 1.0, 'Participación de las Ciudades en el Mercado')
```



In [27]:

```
# Finalmente veamos un ejemplo Tablas Pivot en Pandas
```

In [28]:

```
#¿Qué pasa si queremos separar los ingresos de las personas casadas y solteras en los d
istintos estados o provincias?

df_piv2=df.pivot_table(index=["State or Province"], values=["Revenue"],columns=["Marita
l Status"],\
                       aggfunc=[np.mean],margins=True)
#values: la variable que queremos separar
#columns: criterio de separación
#index: criterio de agrupamiento
#aggfunc: permite realizar calculos matem{aticos sobre las variables, en este caso la m
edia}
#margins: crea una nueva columna("All") para presentar los cálculos de "aggfunc"

df_piv2
```

Out[28]:

	mean		
	Revenue		
Marital Status	M	S	All
State or Province			
BC	13.921085	13.458697	13.679889
CA	12.785429	12.634205	12.707797
DF	12.784347	13.444101	13.121914
Guerrero	14.278889	12.828396	13.476005
Jalisco	4.917000	7.726909	6.977600
OR	13.270420	13.367956	13.323802
Veracruz	13.156840	13.813131	13.459526
WA	12.731647	12.854021	12.791695
Yucatan	13.215471	13.515569	13.364602
Zacatecas	13.249223	13.138078	13.192151
All	12.994518	13.014051	13.004512

In [29]:

```
# ¿Y si no queremos la media sino el máximo?
df_piv=df.pivot_table(index=["State or Province"], values=["Revenue"],columns=["Marital Status"],\
                            aggfunc=[max],margins=True)
df_piv
```

Out[29]:

	max		
	Revenue		
Marital Status	M	S	All
State or Province			
BC	54.46	54.30	54.46
CA	48.25	46.08	48.25
DF	40.60	45.37	45.37
Guerrero	47.04	47.58	47.58
Jalisco	19.40	23.81	23.81
OR	46.08	53.19	53.19
Veracruz	43.68	52.80	52.80
WA	50.17	52.35	52.35
Yucatan	46.08	41.88	46.08
Zacatecas	56.70	49.75	56.70
All	56.70	54.30	56.70

In [ ]:

```

```