Práctica 2

Análisis de datos en Python

Joan Amorós Ramírez

Minería de Datos

3º Grado en Ingeniería Informática en Tecnologías de la Información

Noviembre 2022

Índice

1) Lectura de un fichero

En primer lugar vamos a leer el archivo .csv (en mi caso 'votaciones.csv) con la coma (,) como separador. Para ello, haremos uso de"**Pandas**", una herramienta rápida, flexible y potente de análisis y manipulación de datos.

```
import pandas as pd

datos = pd.read_csv("votaciones.csv", sep = ",")
```

Con la siguiente instrucción, se leerán todos los datos del archivo .csv separados por comas.

2) Análisis descriptivos preliminares

En este segundo análisis del programa, lo que haremos será realizar una descriptiva de los datos con la instrucción **`datos.describe()`**. Si queremos contar los valores nulos (o blancos) de la base de datos, podremos escribir en el código del programa las expresiones `print(datos.isnull().sum())` y `print(pd.isnull(datos).sum())`.

Otras dos características importantes de este análisis son: contar de los datos únicos que contiene cada variable (con el código `datos["Mi variable"].unique()`), y también hacer un recuento de los valores únicos de cada una de nuestras variables (con la expresión (`datos["Mi variable"].value_counts()`)

2.1. Valores nulos

Ninguna de las variables tiene valores nulos; por ello, el archivo .csv dispone de todos los registros.

Partido Votado	0
Comunidad	0
Edad	0
Ocupación	0
Género	0
Estado Civil	0
Patrimonio (euros)	0

dtype: int64 → Esto representa el tipo de dato con el que se está trabajando en la base de datos (en este caso, todos los registros son valores de tipo entero de 64 bits)

2.2. Datos únicos

No hay datos repetidos en cada variable, tras haber analizado completamente la base de datos:

Partido Votado	0
Comunidad	0
Edad	0
Ocupación	0
Género	0
Estado Civil	0
Patrimonio (euros)	0

dtype: int64

2.3. Valores únicos

Se ha escogido una columna, en este caso Patrimonio (euros), y casi todos los valores son únicos (es decir, solo se repiten una vez); excepto el valor 810 que se repite dos veces.

810	2
15756	1
40187	1
68205	1
88586	1
…	…
20813	1
20704	1
16004	1
40874	1
15688	1

Name: Patrimonio (euros), Length: 99, dtype: int64

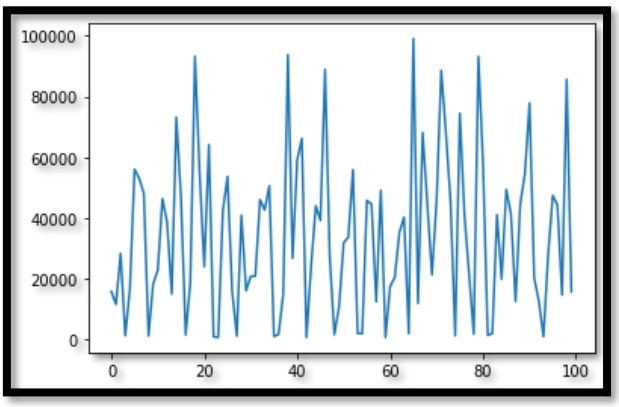
3) Gráficos descriptivos. Correlaciones

Aquí haremos uso de la librería "Matplotlib" de Python, que está especializada en crear gráficos (e incluso histogramas) de dos dimensiones.

```
import matplotlib.pyplot as plt
```

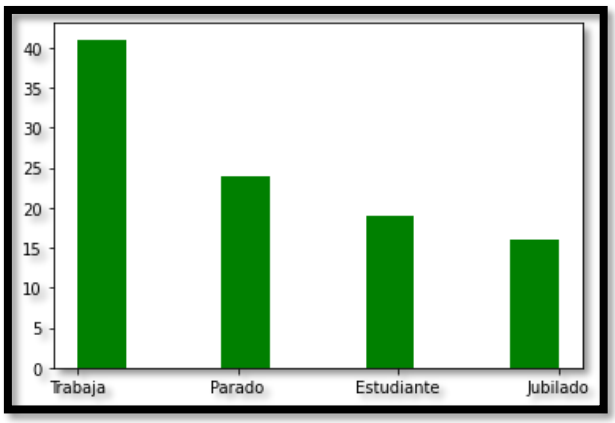
3.1. Gráfico de líneas

Con las líneas de código `plt.plot(datos["Mi variable"])` y `plt.show()` se genera un gráfico formado por puntos unidos por líneas (en este ejemplo, la variable independiente sería el número de filas de la base de datos [100] y la dependiente el intervalo de valores posibles de mi variable de la tabla [Patrimonio en euros; 0 – 100.000])



3.2. Histogramas

La función hist() en el módulo .pyplot de la biblioteca "Matplotlib" se usa generalmente para trazar un histograma, como podemos ver en el siguiente ejemplo:

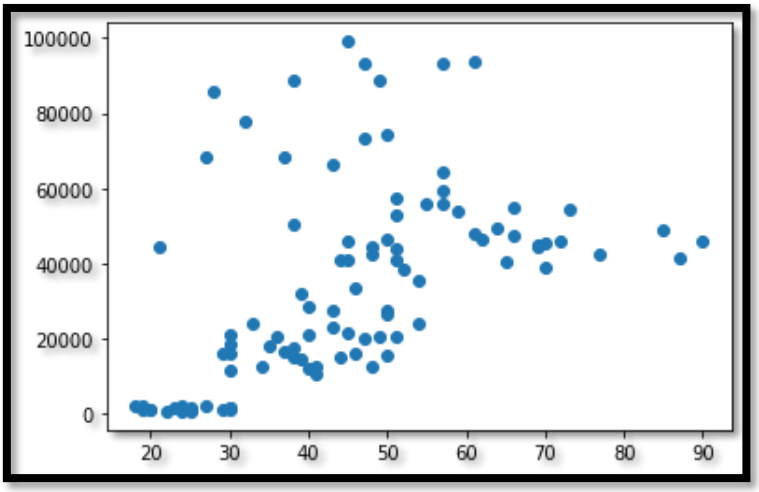



```
plt.hist(datos["Mi variable"], facecolor = 'color')    plt.show()
```

3.3. Gráfico de dispersión

La siguiente instrucción nos muestra un gráfico de dispersión con la variable independiente (en mi caso **Edad**) y la variable dependiente (**Patrimonio en euros**).

```
plt.scatter(datos["Variable independiente"],datos["Variable dependiente"])

plt.show()
```



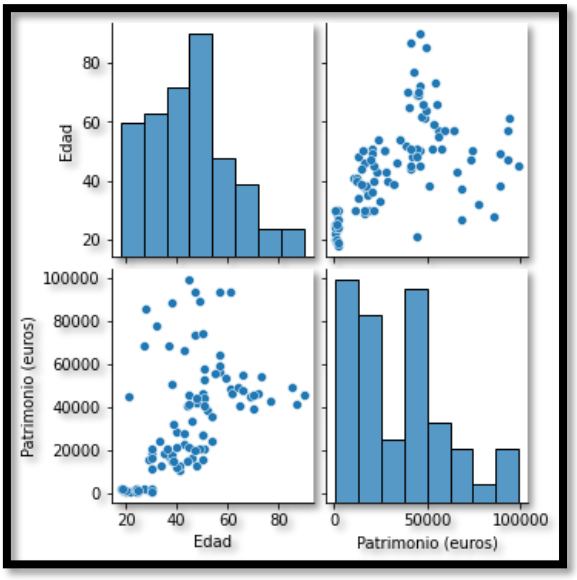
4) Seaborn

Seaborn es una librería de visualización de datos en Python, que proporciona una interfaz de un alto nivel para dibujar gráficos estadísticos atractivos e informativos.

Los gráficos que se han generado en esta sección del código han sido (en la siguiente imagen, de izquierda a derecha y de arriba abajo):

1. Número de individuos según su edad.
2. Edad en función del patrimonio (euros).
3. Patrimonio en función de la edad.
4. Número de personas según su patrimonio.

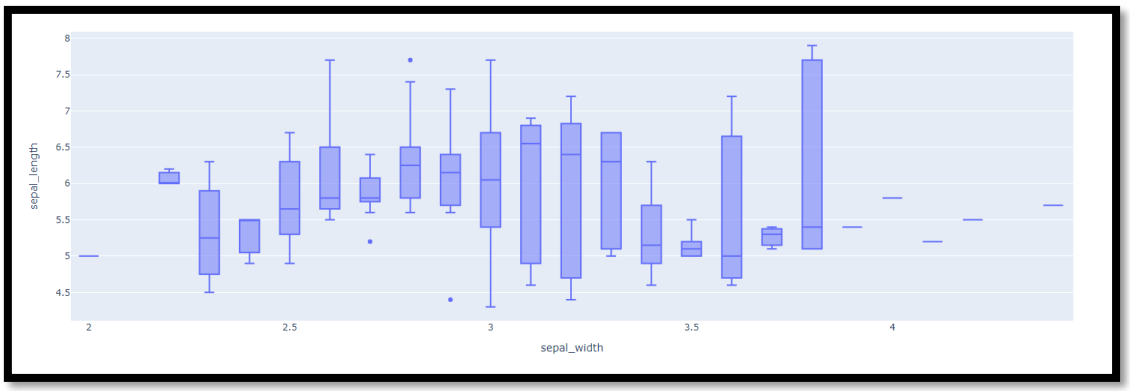
```
import seaborn as sns
sns.pairplot(datos.select_dtypes(exclude=[object]))
```



5) Plotly Express

Plotly Express es otra librería propia de Python que proporciona una gran variedad funciones para crear diferentes tipos de figuras.

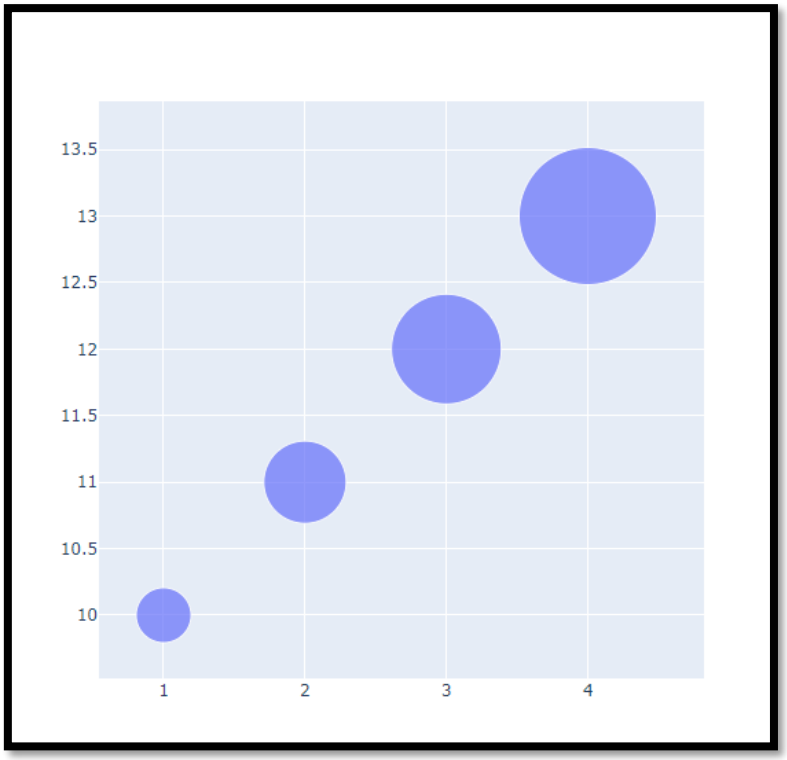
Con la instrucción px.box(df, x, y) crearemos un diagrama de caja, que es una representación estadística de la distribución de una variable a través de sus cuartiles. Los extremos de la caja representan los cuartiles inferior y superior, mientras que la mediana (segundo cuartil) está marcada por una línea dentro de la caja.




```
import plotly.express as px

df = px.data.iris()

fig = px.box(df, x="sepal_width", y="sepal_length")

fig.show()
```

6) Plotly Go Scatter

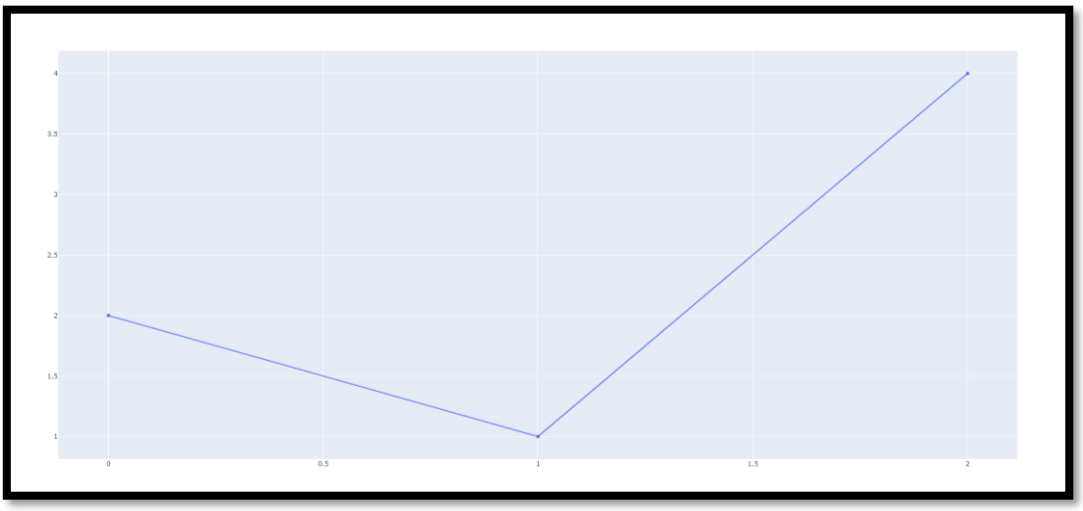
Si empleamos la línea de código `go.scatter`, estaremos representando los puntos de datos como puntos marcadores, como podemos ver en el siguiente ejemplo:




```
import plotly
from plotly.offline import plot
import plotly.graph_objs as go

plotly.offline.plot
(
      {
            "data": [go.Scatter(x    =    [1, 2, 3, 4],    y    =
            [10, 11, 12, 13], mode   =   'markers',   marker   =
            dict(size = [40, 60, 80, 100]))],

            "layout": go.Layout(showlegend = False, height =
            600, width = 600)
      }
)
```

7) Plotly Go Figure

También la librería “Plotly” de Python ofrece una función para trazar líneas según los datos introducidos. Este es uno de los ejemplos que se han realizado:

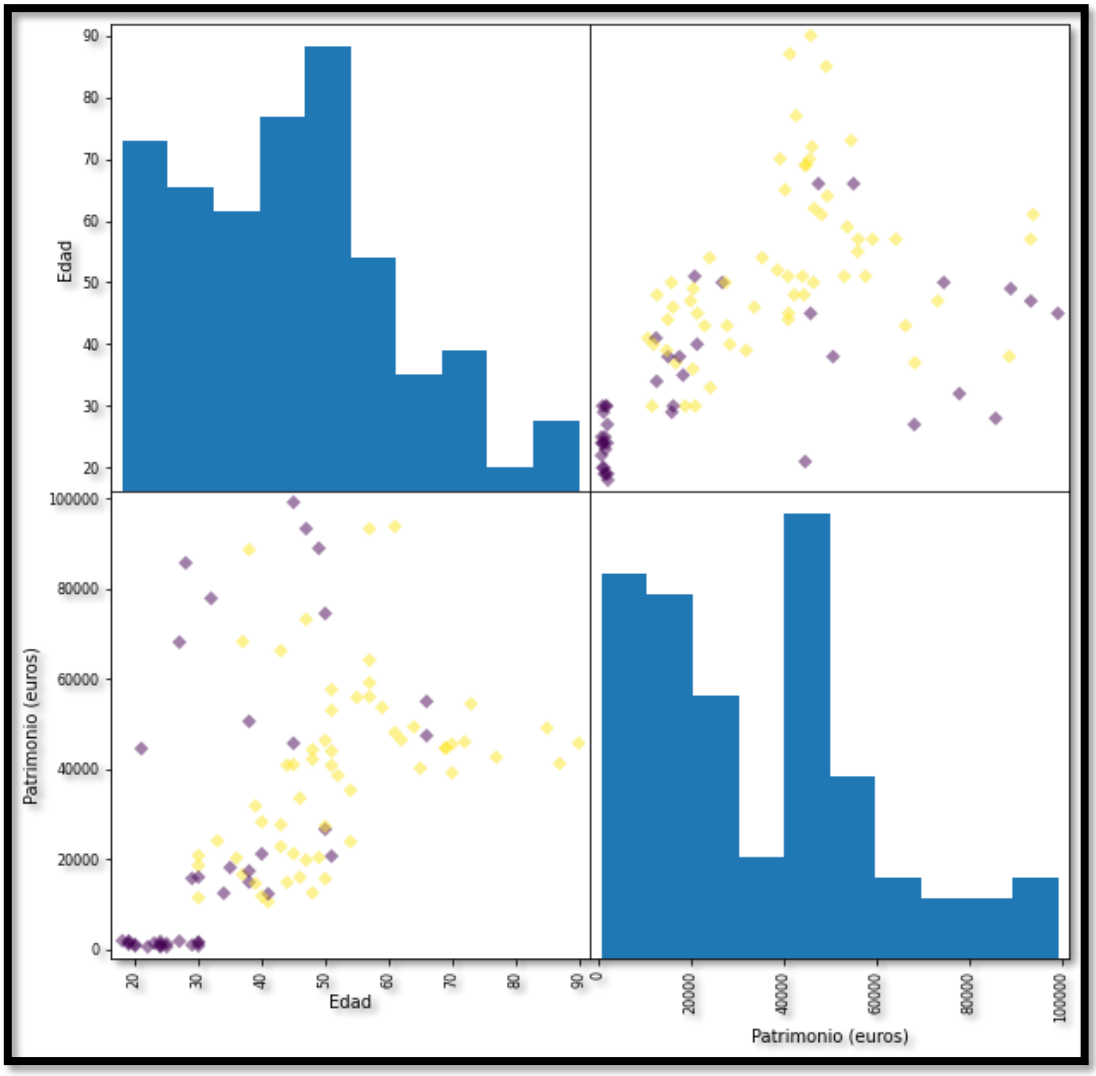


```
fig = go.Figure(data=[{'type': 'scatter', 'y': [2, 1, 4]}])

plot(fig)
```

8) Preprocesamiento de datos

En este apartado de la práctica vamos a escoger las dos variables numéricas con las que hemos estado trabajando durante la mayor parte de la práctica (edad y patrimonio en euros) y vamos a escoger otras variables categóricas tales como el género (masculino o femenino) y el estado civil (casado o soltero) para representarlas una de ellas de un color. Al fin y al cabo vamos a obtener un gráfico parecido al del apartado 4 (Seaborn), donde en el gráfico de dispersión los individuos que se encuentran solteros aparecen de color morado y los que se han casado están coloreados de amarillo.

Ya por otra parte, en este mismo apartado también aparecen los gráficos del patrimonio de cada persona según la posición en la base de datos (en la parte superior izquierda del gráfico) y la cantidad de individuos según su edad (situado en la parte inferior derecha).


Edad
Patrimonio (euros)
Edad
Patrimonio (euros)

9) Métodos supervisados

En los métodos supervisados se han implementado árboles de decisión para el análisis de la tabla, de tres tipos diferentes: clasificación, predicción y regresión:

9.1. Clasificación

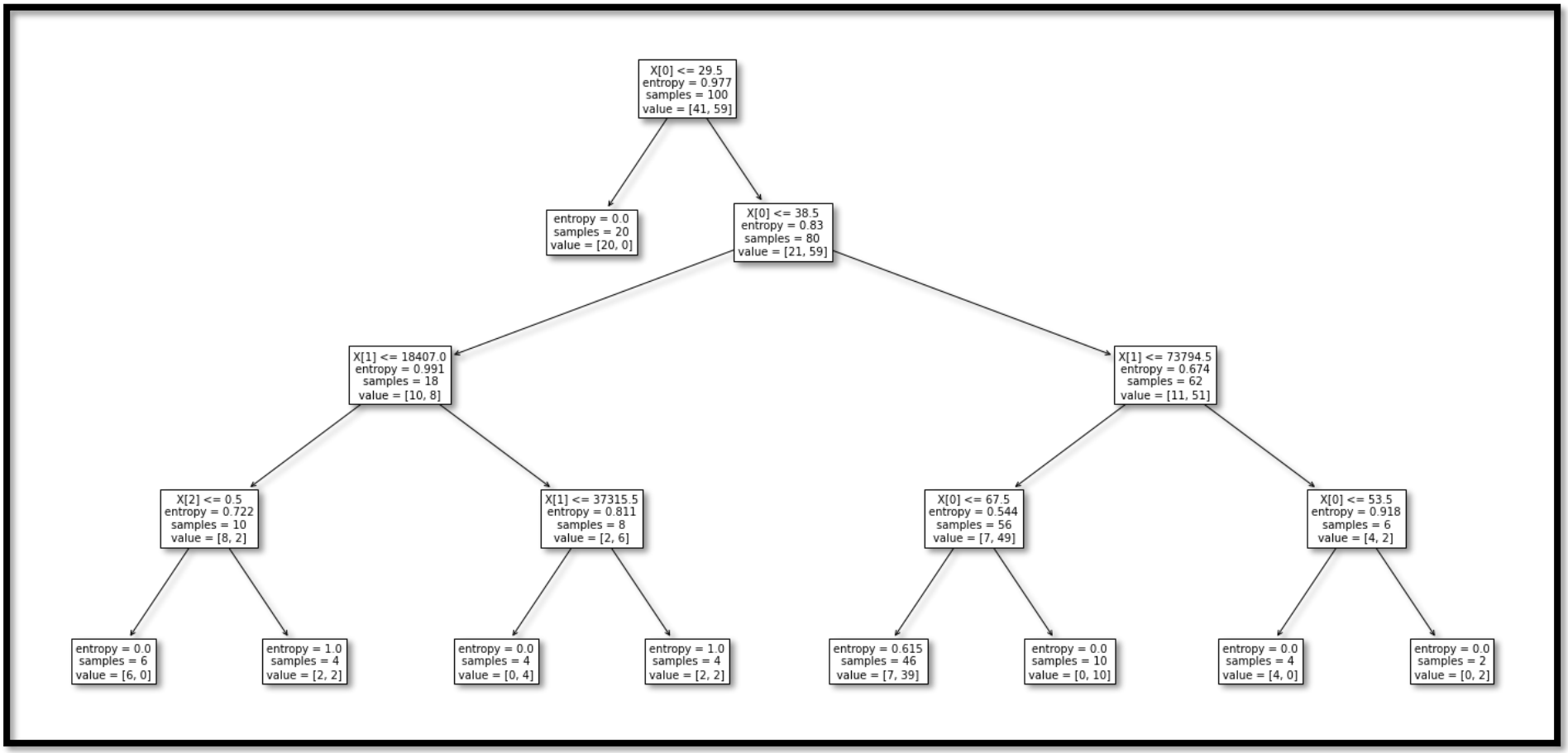
En el primer apartado del análisis de la tabla con los árboles de decisión, se han podido sacar las siguientes conclusiones:

En primer lugar se han escogido los 100 individuos y se han dividido según la edad (los que tenían menos y más de 29,5 años) junto con los valores del estado civil (41 personas solteras y 59 que se encuentran casadas). Por una parte, las 20 personas cuya edad era menor que 29,5 (años) eran todos solteros y por otra, en las 80 restantes había 21 solteros/as y 59 ya habían contraído matrimonio.

Seguidamente, se ha vuelto ha realizar la segunda ramificación del árbol en función de la edad de cada individuo. En el lado izquierdo de la subdivisión se encontraban 10 indiviuos solteros/as y 8 casados/as (todos ellos/as menores de 38,5 años); mientras que en la parte derecha había 62 personas mayores que 38,5 años (11 sin pareja y 51 casados/as).

Concretamente, en la sección izquierda, se han escogido las 18 personas con menos de 38,5 años de edad. Después el árbol ha elegido los individuos con un salario inferior a 18407 euros, donde hay 8 personas solteros/as y 2 con pareja. Además, se vuelven a dividir de nuevo por género (siendo la variable x[2] masculino cuando vale 0 y femenino cuando vale 1); donde en el árbol se han podido observar 6 hombres solteros y 4 mujeres (2 que no tienen pareja y otras dos que sí). En cuanto a las personas cuyo patrimonio está comprendido entre 18407 y 37315,5 euros, nos encontramos con 2 solteros/as y 6 casados/as (de ellos 4 hombres casados, 2 solteros y 2 mujeres sin haberse casado)

Si nos fijamos en la ramificación derecha había 62 individuos con más de 38,5 años de edad, de los cuales 56 con un patrimonio menor que 73794,5 euros (7 solteros/as y 49 casados/as), que 46 de ellos (7 sin pareja y 39 con matrimonio) tenían menos de 67,5 años y los otros 10 eran mayores que los anteriores. Finalmente, aquellas 6 personas (4 solteras y 2 casadas) cuyo patrimonio estaba por encima de 73794,5 euros, 4 de ellas tenían menos de 53,5 años y casualmente las 4 no habían contraído matrimonio. En cambio, las otras dos, sí que se encontraban actualmente casadas.


X[0] <= 29.5
entropy = 0.977
samples = 100
value = [41, 59]
entropy = 0.0
samples = 20
value = [20, 0]
X[0] <= 38.5
entropy = 0.83
samples = 80
value = [21, 59]
X[1] <= 18407.0
entropy = 0.991
samples = 18
value = [10, 8]
X[1] <= 73794.5
entropy = 0.674
samples = 62
value = [11, 51]
X[2] <= 0.5
entropy = 0.722
samples = 10
value = [8, 2]
X[1] <= 37315.5
entropy = 0.811
samples = 8
value = [2, 6]
X[0] <= 67.5
entropy = 0.544
samples = 56
value = [7, 49]
X[0] <= 53.5
entropy = 0.918
samples = 6
value = [4, 2]
entropy = 0.0
samples = 6
value = [6, 0]
entropy = 1.0
samples = 4
value = [2, 2]
entropy = 0.0
samples = 4
value = [0, 4]
entropy = 1.0
samples = 4
value = [2, 2]
entropy = 0.615
samples = 46
value = [7, 39]
entropy = 0.0
samples = 10
value = [0, 10]
entropy = 0.0
samples = 4
value = [4, 0]
entropy = 0.0
samples = 2
value = [0, 2]

9.2. Predicción

Es un modelo predictivo formado por reglas binarias con las que pretende conseguir repartir las observaciones en función de sus atributos y predecir así el valor de la variable respuesta.

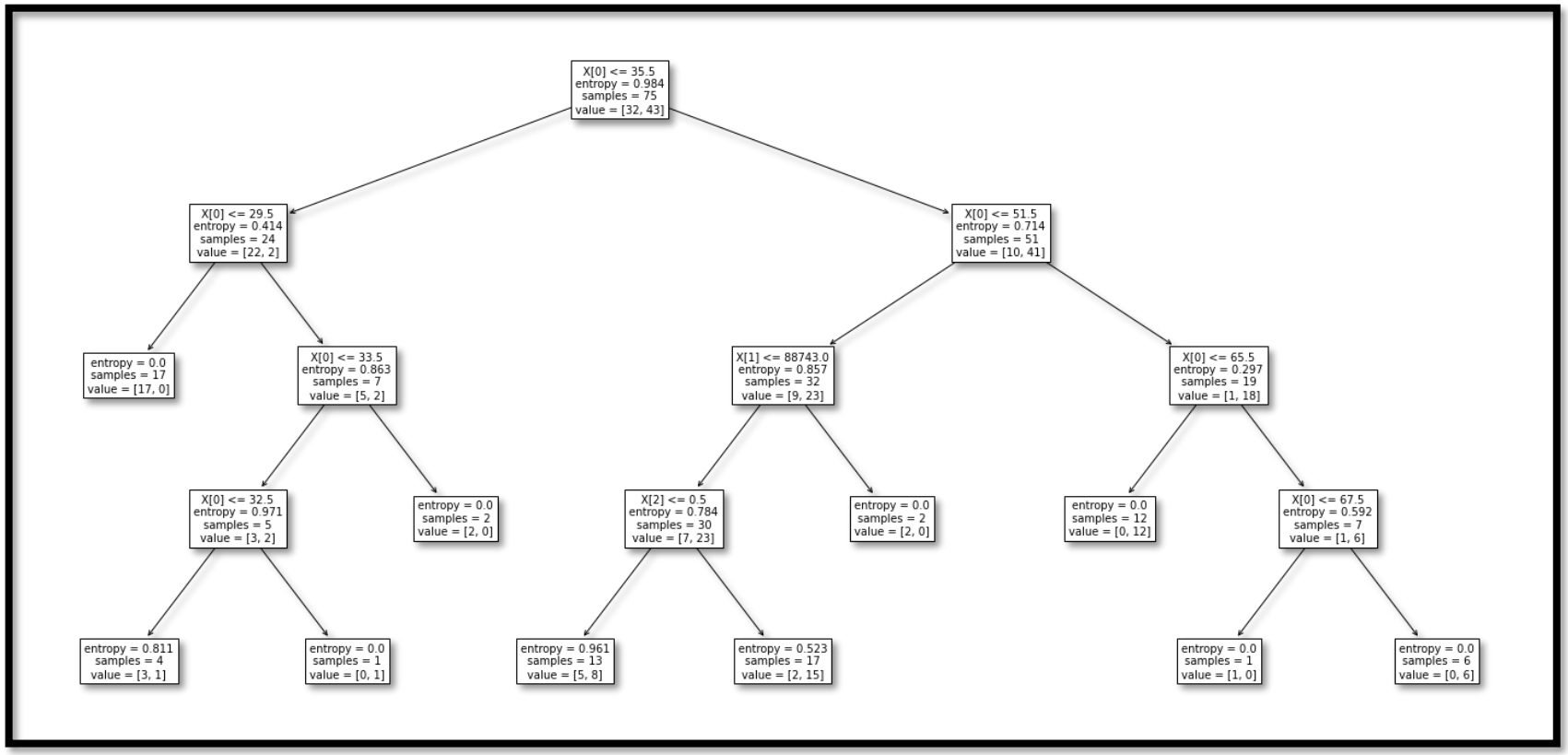
Se trata de un árbol muy similar al anterior, pero con algunas diferencias: no se toma el mismo número de muestras o de registros, algunos parámetros del árbol varían; pero a pesar de ello también se incluyen datos analíticos como los siguientes:

I. **Mean Absolute Error** (error absoluto medio): 0.2506451612903226

II. **Mean Squared Error** (error medio cuadrado): 0.16434703433922995

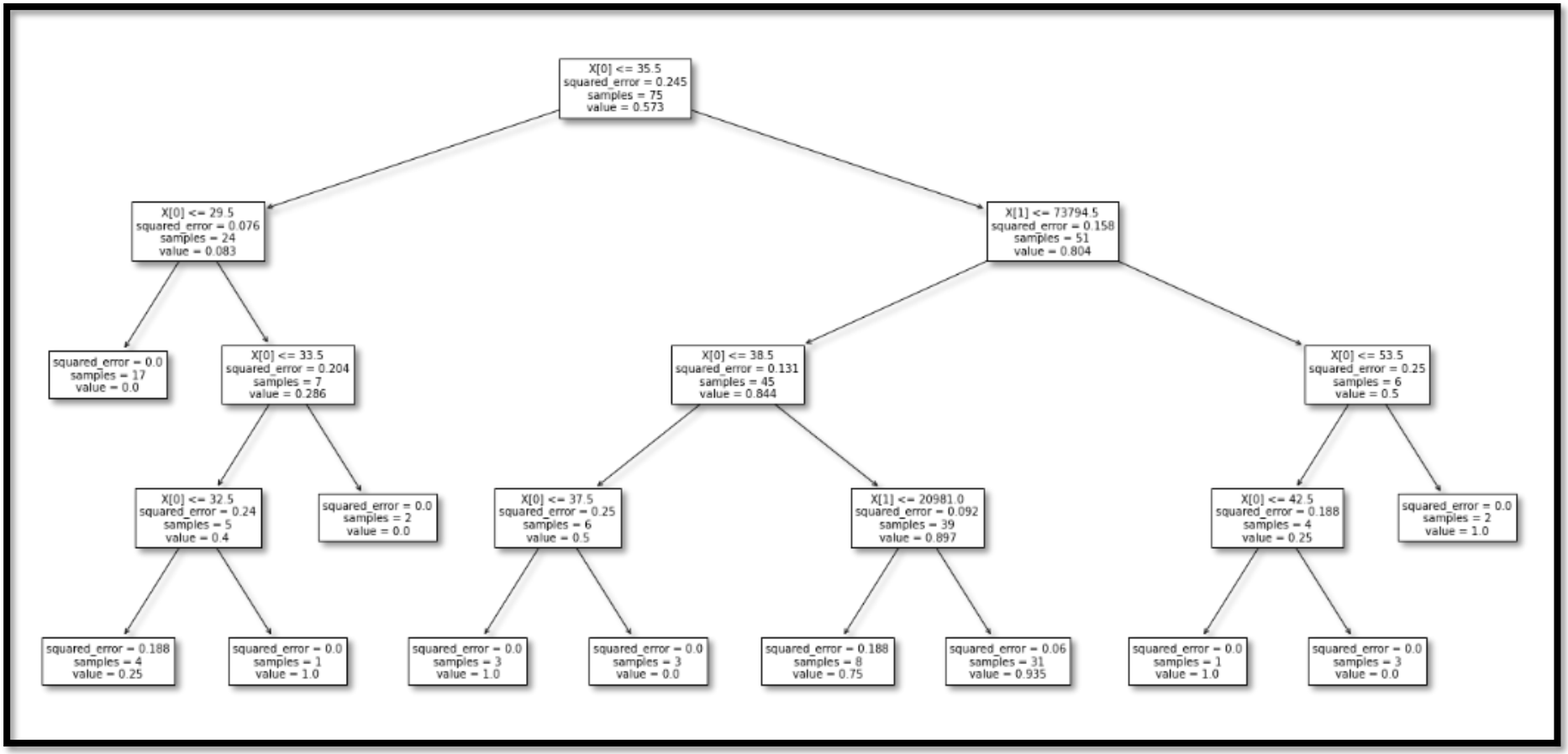
III. **Root Mean Squared Error** (Error cuadrático medio de la raíz): 0.40539737830828404

Como en el caso anterior, el árbol dibujado aparecerá en la siguiente página para que se pueda ver en detalle.



9.3. Regresión

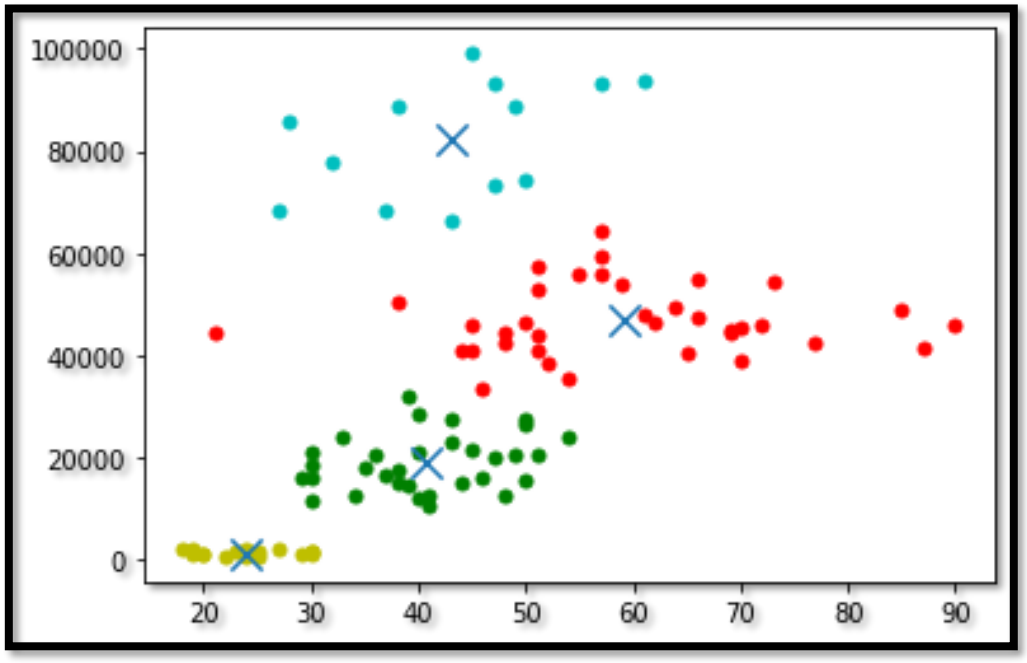
Es otro tipo de árbol cuyo propósito es predecir las variables dependientes en función de las independientes. Ha salido similar a los dos anteriores, y además ha aparecido con las mismas variables que estábamos trabajando.


X[0] <= 35.5
squared_error = 0.245
samples = 75
value = 0.573
X[0] <= 29.5
squared_error = 0.076
samples = 24
value = 0.083
X[1] <= 73794.5
squared_error = 0.158
samples = 51
value = 0.804
squared_error = 0.0
samples = 17
value = 0.0
X[0] <= 33.5
squared_error = 0.204
samples = 7
value = 0.286
X[0] <= 38.5
squared_error = 0.131
samples = 45
value = 0.844
X[0] <= 53.5
squared_error = 0.25
samples = 6
value = 0.5
X[0] <= 32.5
squared_error = 0.24
samples = 5
value = 0.4
squared_error = 0.0
samples = 2
value = 0.0
X[0] <= 37.5
squared_error = 0.25
samples = 6
value = 0.5
X[1] <= 20981.0
squared_error = 0.092
samples = 39
value = 0.897
X[0] <= 42.5
squared_error = 0.188
samples = 4
value = 0.25
squared_error = 0.0
samples = 2
value = 1.0
squared_error = 0.188
samples = 4
value = 0.25
squared_error = 0.0
samples = 1
value = 1.0
squared_error = 0.0
samples = 3
value = 1.0
squared_error = 0.0
samples = 3
value = 0.0
squared_error = 0.188
samples = 8
value = 0.75
squared_error = 0.06
samples = 31
value = 0.935
squared_error = 0.0
samples = 1
value = 1.0
squared_error = 0.0
samples = 3
value = 0.0

10) Métodos no supervisados

10.1. Clustering - KMeans, Sklearn

En los métodos no supervisados, hemos escogido el método de clustering (también explicado en la primera práctica) utilizando los algoritmos “KMeans” y “Sklearn”. En mi gráfico se pueden observar cuatro grandes clusters (agrupaciones de datos muy similares entre ellos), cada uno de un diferente color:



Para ello, se ha empleado la instrucción principal
`kmeans = KMeans(n_clusters = 4)`, siendo 4 el número de clusters.

11) Reglas de asociación

Por último, hemos podido estudiar algunas de las reglas de asociación de mi base de datos. Se ha hecho con el algoritmo “Apriori”, que consiste en obtener información sobre las relaciones estructuradas entre los diferentes elementos involucrados. Estos han sido los resultados que hemos podido ver en el algoritmo aplicado a nuestras variables: `"Partido Votado"`, `"Comunidad"`, `"Ocupación"`, `"Genero"`.

	Left_Hand_Side	Right_Hand_Side	Support	Confidence	Lift
0	Aragon	PP	0.04	0.80	3.20
1	Asturias	PNV	0.01	0.25	6.25
2	Baleares	PSOE	0.02	0.67	3.33
3	Canarias	PSOE	0.04	0.67	3.33
4	Castilla La Mancha	Podemos	0.02	0.50	3.57
5	Catalunya	JxCat	0.05	0.50	10.00
6	Ceuta	Jubilado	0.01	0.50	3.13
7	Melilla	Ciudadanos	0.01	1.00	8.33
8	Navarra	Ciudadanos	0.02	0.67	5.56
9	Extremadura	Estudiante	0.02	1.00	5.26
10	La Rioja	Jubilado	0.01	0.50	3.13
11	PNV	Jubilado	0.03	0.75	4.69
12	La Rioja	Podemos	0.01	0.50	3.57
13	Melilla	Parado	0.01	1.00	4.17
14	PNV	Pais Vasco	0.03	0.75	10.71