

**1. Genere un CSV que contenga la información alumno(ci, nombre completo, sexo, departamento, nota promedio, edad) con 20 filas mínimo como datos (El primer registro es de usted).**

La información del alumno se encuentra en el archivo lista1.csv

**a. La media y la desviación estándar y explique qué significa en cada caso mediante Python sin uso de librerías**

El código usado se encuentra en el archivo P1\_A.py

```
trabajons python')
La desviacion estandar de CI es: 2156506.621907117
La media es de CI es: 7883169.125
La desviacion estandar de NOTA es: 15.95913032028093
La media es de NOTA es: 68.54166666666667
La desviacion estandar de EDAD es: 2.3090088217957145
La media es de EDAD es: 23.125
```

La **desviación estándar de CI** es 2156506,6 como vemos es muy alta esto nos indica que los CI extienden sobre un rango de valores más amplio es decir están más dispersos.

La **media de los CI** es 7883169 nos dice que, si escogemos un estudiante al azar, este tendría su CI se encontraría un poco por encima o por debajo de la media de ci.

La **desviación estándar de NOTAS** es 15,96 es una desviación un poco alta con respecto a las notas esto nos dice que las notas están en un rango amplio.

La **media de las NOTAS** es 68.5 nos dice que, si escogemos un estudiante al azar, seria un estudiante aprobado ya que la nota media es de 68.5

La **desviación estándar de EDAD** es 2.3 es una desviación baja con respecto a las edad nos dice que los estudiantes tienen casi la misma edad que las edad no están separadas.

La **media de EDAD** es 23.1 es la edad promedio de los estudiantes podríamos esperar que si tenemos un estudiante su edad estaría en 23 años quizás 1 año mas o menos.

**b. La media, la moda, la desviación estándar con el uso de numpy y pandas**

El código usado se encuentra en el archivo P1\_B.py

```
23 9998883          ARANEDA CASTIGLIONI TOMAS IGNACIO ...    65    24

[24 rows x 6 columns]
La moda de la NOTA es:
65
La moda de la EDAD es:
24
La moda de la DPTO es:
LA PAZ
La Desviacion estandar CI es:
2156506.621907117
La Desviacion estandar EDAD es:
2.3090088217957145
La Desviacion estandar NOTA es:
15.95913032028093
El promedio de CI es:
7883169.125
El promedio de EDAD es:
23.125
El promedio de NOTA es:
68.54166666666667
```

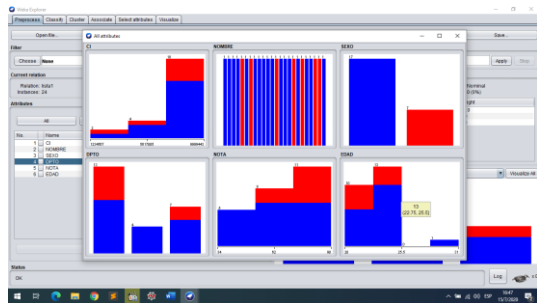
Nos quedaría comentar sobre la moda en cada columna, para comenzar no calcule la moda de CI ya que el CI solo le corresponde a una persona y así mismo la moda del nombre no

se calculo ya que es una pequeña muestra y no se encuentra 2 o mas personas con el mismo nombre y apellidos.

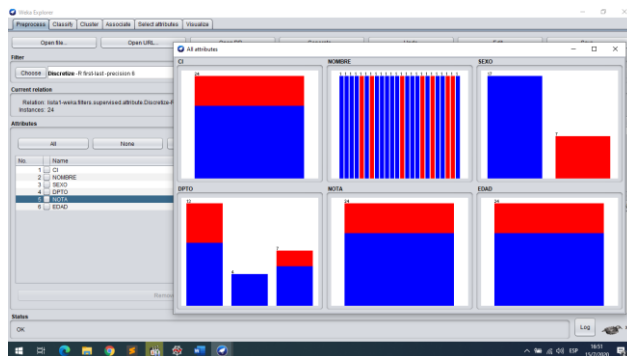
La moda en la NOTA es de 65 esto indica que existe más estudiantes con la nota de 65

Asi también vemos que la moda de la edad es de 24 años es decir tenemos mas estudiantes con la edad de 24 y también vemos que muchos\_estudiantes son del departamento de LA PAZ

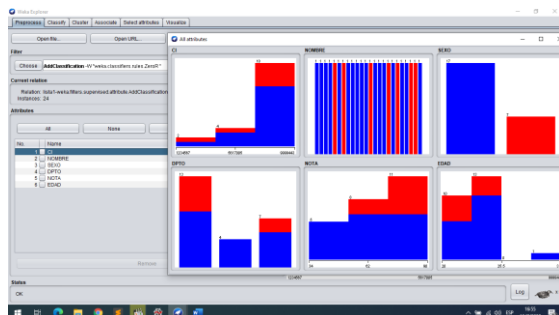
**2. Del archivo del problema 1 en WEKA y PYTHON realice el preprocesamiento utilizando al menos dos métodos del mismo.**



Choose-filters-supervised-atribute-Discretize

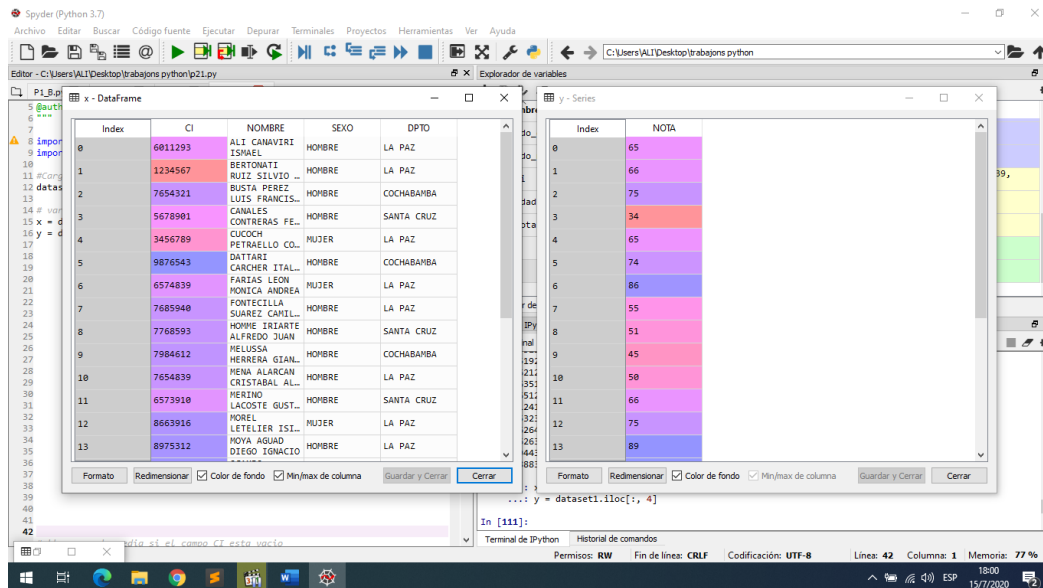


Choose-filters-supervised-atribute -Addclasification



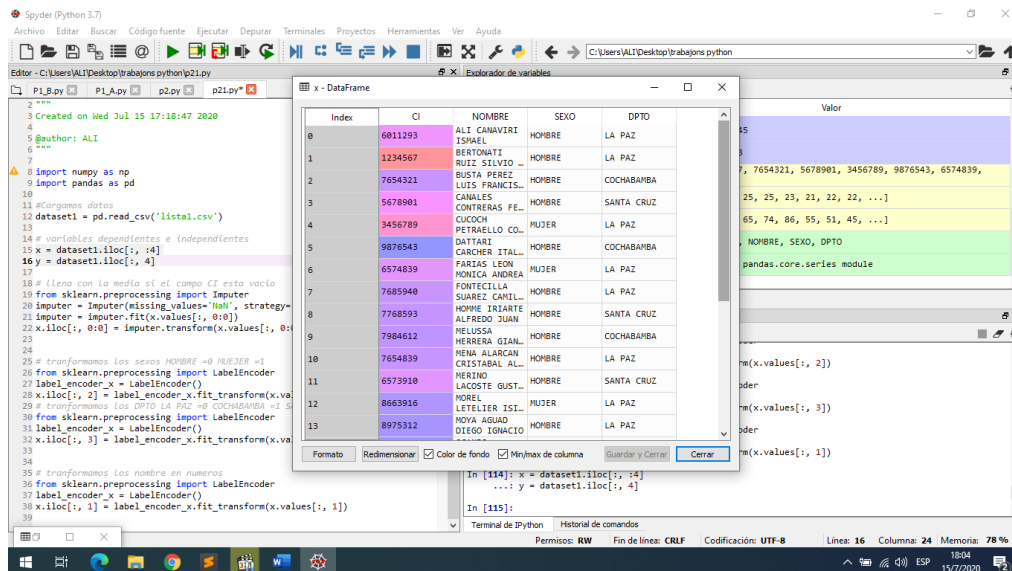
En Python las dividimos en variables independientes x e independientes y

El código usado esta en el archivo p21.py

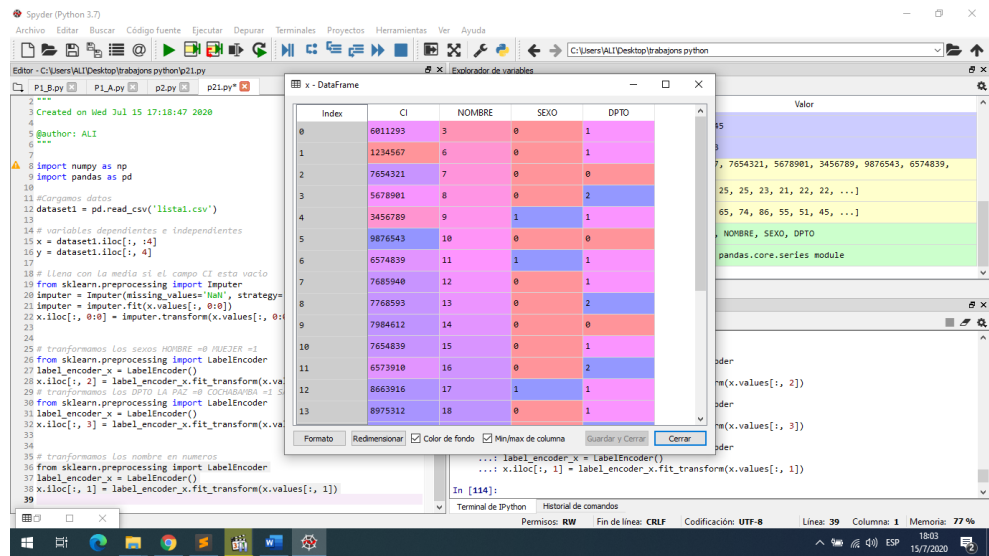


Podemos usar la tranformacion

from sklearn.preprocessing import Imputer para llenar los campos vacios pero en nuestro dataset no tenemos campos vacios



## # transformamos el NOMBRE, SEXO, DPTO en números

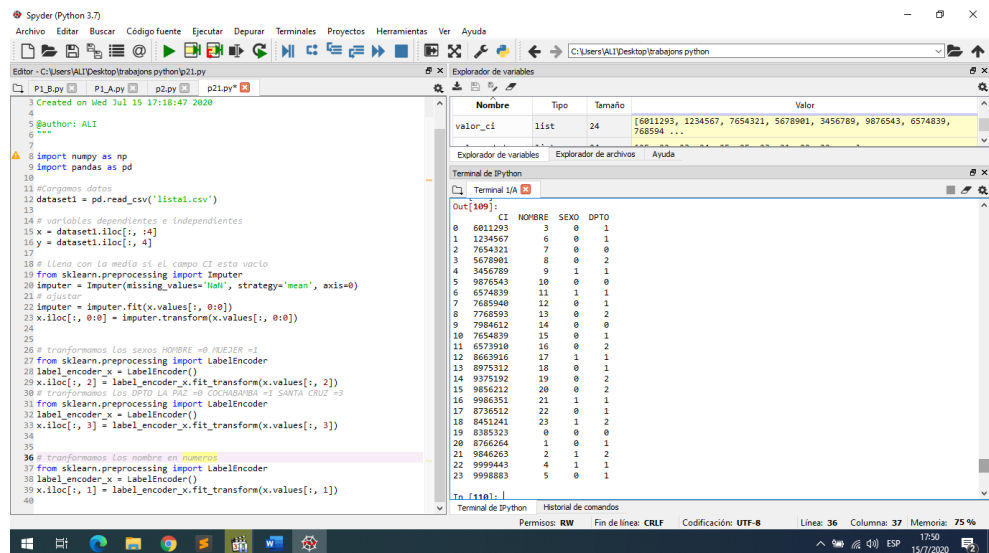


The screenshot shows the Spyder Python 3.7 IDE. The main window displays a DataFrame with the following data:

Index	CI	NOMBRE	SEXO	DPTO
0	6811293	3	0	1
1	1234567	6	0	1
2	7654321	7	0	0
3	5678901	8	0	2
4	3456789	9	1	1
5	9876543	10	0	0
6	6574839	11	1	1
7	7685940	12	0	1
8	7768593	13	0	2
9	7984612	14	0	0
10	7654839	15	0	1
11	6573910	16	0	2
12	8663916	17	1	1
13	8975312	18	0	1

The code in the editor shows the following steps:

```
1 # Importamos librerías
2 #
3 Created on Wed Jul 15 17:18:47 2020
4
5 @author: ALI
6
7
8 import numpy as np
9 import pandas as pd
10
11 #Cargamos datos
12 dataset1 = pd.read_csv('lista1.csv')
13
14 # variables dependientes e independientes
15 x = dataset1.iloc[:, :4]
16 y = dataset1.iloc[:, 4]
17
18 # Lleno con la media si el campo CI está vacío
19 from sklearn.preprocessing import Imputer
20 imputer = Imputer(missing_values='NaN', strategy='mean', axis=0)
21 imputer = imputer.fit(x.values[:, 0:0])
22 x.iloc[:, 0:0] = imputer.transform(x.values[:, 0:0])
23
24
25 # transformamos los sexos HOMBRE =0 MUJER =1
26 from sklearn.preprocessing import LabelEncoder
27 label_encoder_x = LabelEncoder()
28 x.iloc[:, 1] = label_encoder_x.fit_transform(x.values[:, 1])
29 # transformamos los DPTO LA PAZ =0 COCHABAMBA =1 SANTA CRUZ =2
30 from sklearn.preprocessing import LabelEncoder
31 label_encoder_x = LabelEncoder()
32 x.iloc[:, 2] = label_encoder_x.fit_transform(x.values[:, 2])
33
34
35 # transformamos los nombres en números
36 from sklearn.preprocessing import LabelEncoder
37 label_encoder_x = LabelEncoder()
38 x.iloc[:, 1] = label_encoder_x.fit_transform(x.values[:, 1])
39
```



The screenshot shows the Spyder Python 3.7 IDE. The main window displays the same DataFrame as the first screenshot, but with the NOMBRE, SEXO, and DPTO columns transformed into numerical values. The DataFrame is displayed in a table view with 14 rows and 4 columns. The NOMBRE column contains numerical values like '6811293', '1234567', etc. The SEXO column contains values 0 and 1. The DPTO column contains values 1, 2, 3, etc. The CI column contains values like '6811293', '1234567', etc.

The code in the editor shows the following steps:

```
1 # Importamos librerías
2 #
3 Created on Wed Jul 15 17:18:47 2020
4
5 @author: ALI
6
7
8 import numpy as np
9 import pandas as pd
10
11 #Cargamos datos
12 dataset1 = pd.read_csv('lista1.csv')
13
14 # variables dependientes e independientes
15 x = dataset1.iloc[:, :4]
16 y = dataset1.iloc[:, 4]
17
18 # Lleno con la media si el campo CI está vacío
19 from sklearn.preprocessing import Imputer
20 imputer = Imputer(missing_values='NaN', strategy='mean', axis=0)
21 imputer = imputer.fit(x.values[:, 0:0])
22 # ajustar
23 x.iloc[:, 0:0] = imputer.transform(x.values[:, 0:0])
24
25
26 # transformamos los sexos HOMBRE =0 MUJER =1
27 from sklearn.preprocessing import LabelEncoder
28 label_encoder_x = LabelEncoder()
29 x.iloc[:, 1] = label_encoder_x.fit_transform(x.values[:, 1])
30 # transformamos los DPTO LA PAZ =0 COCHABAMBA =1 SANTA CRUZ =2
31 from sklearn.preprocessing import LabelEncoder
32 label_encoder_x = LabelEncoder()
33 x.iloc[:, 2] = label_encoder_x.fit_transform(x.values[:, 2])
34
35
36 # transformamos los nombres en números
37 from sklearn.preprocessing import LabelEncoder
38 label_encoder_x = LabelEncoder()
39 x.iloc[:, 1] = label_encoder_x.fit_transform(x.values[:, 1])
40
```