Lectura 2

February 28, 2024

1 Aplicaciones de Minería de Datos I

1.1 Lectura 2: Exploración y Representación Visual de un Conjunto de Datos utilizando Python

1.1.1

***Se sugiere leer brevemente el siguiente tutorial en línea, para comprender la noción de método y función en el lenguaje de programación Python*: Difference Between Method and Function in Python | Python Method Vs Function** Python: El lenguaje más empleado en el ámbito de la inteligencia artificial y el machine learning

El 57% de los científicos de datos y desarrolladores de aprendizaje automático lo emplean, y el 33% lo elige por encima de otros lenguajes para sus proyectos.

Disponible para la implementación de algoritmos de inteligencia artificial, abarcando tanto el aprendizaje automático, el aprendizaje profundo como el procesamiento del lenguaje natural (NLP – Natural Language Processing).

1.1.2 Repaso de Python

Los guiones con código pueden ser guardados con la extensión .py y ejecutados a través del intérprete, o bien, se pueden ejecutar comandos directamente desde el intérprete.

Reglas básicas de codificación Indentación: mantener siempre el mismo nivel de sangrado para cada bloque de código.

La función print() se utiliza para desplegar en la consola tanto literales como variables.

Los nombres de las variables son sensibles a mayúsculas y minúsculas, pueden incluir letras, números y el símbolo de subrayado, pero no pueden empezar con un número.

Existen funciones built-in: aquells predefinidas que vienen incluidas con Python y pueden ser utilizadas directamente en un programa sin necesidad de importar bibliotecas externas.

[1]:
```
if 3 < 5:
    print("3 es menor a 5") #primer nivel de identación
    if 3 > 2:
        print("3 es mayor a 2") #segundo nivel de identación
else:
    #esta sentencia no ejecuta acción
```

```
    pass #pass: palabra reservada en los casos en que necesitemos que no se␣
 ↪realice ninguna operación

x = 1
y = 2

print(x)
print(type(x)) #Función built-in que despliegue el tipo de clase de un objeto
suma = x + y
print(suma)
```

```
3 es menor a 5
3 es mayor a 2
1
<class 'int'>
3
```

Tipos de datos

[2]:
```
#texto
a = 'hola'
print("Texto")
print(a,"=",type(a))
print("\n")

#Numéricos
b = 1
c = 1.0
d = 1j+2
print("Numéricos")
print(b,"=",type(b))
print(c,"=",type(c))
print(d,"=",type(d))
print("\n")

#Secuencia
print("Secuencia")
e = [1,3,4]
f = (1,2)
g = range(0,1)
print(e,"=",type(e))
print(f,"=",type(f))
print(g,"=",type(g))
print("\n")

#Conjuntos
print("Conjuntos")
h = {1,2,3}
```

```
print(h,"=",type(h))
print("\n")

#Diccionarios
print("Diccionarios")
j = {"a":1,"b":2}
print(j,"=",type(j))
print("\n")

#Boolean
print("Boolean")
k = True
print(k,"=",type(k))
print("\n")

#Binarios
print("Binarios")
l = b'Hola'
print(l,"=",type(l))
print("\n")

#El valor None representa un valor especial que indica la ausencia de un valor.
print("Binarios")
m = None
print(m,"=",type(m))
```

```
Texto
hola = <class 'str'>


Numéricos
1 = <class 'int'>
1.0 = <class 'float'>
(2+1j) = <class 'complex'>


Secuencia
[1, 3, 4] = <class 'list'>
(1, 2) = <class 'tuple'>
range(0, 1) = <class 'range'>


Conjuntos
{1, 2, 3} = <class 'set'>


Diccionarios
```

```
{'a': 1, 'b': 2} = <class 'dict'>


Boolean
True = <class 'bool'>


Binarios
b'Hola' = <class 'bytes'>


Binarios
None = <class 'NoneType'>
```

Tipos de colecciones

[3]:
```
#Lista (arreglo): secuencia de objetos
print("Lista")
a = [1,2,2,"datos",1j]
print(a,"=",type(a))
print("\n")

#Tupla: secuencia de inmutable
print("Tupla")
b,c,d = 1,5.0,False
print(b,c,d)
print(type((b,c,d)))
print("\n")

#Conjunto: colección única de objetos
print("Conjunto")
e = {1,2,3,4}
print(e,"=",type(e))
#Se puede convertir una lista a diccionario
print(set(a),"=",set(a))
print("\n")

#Diccionario: pares e clave-valor sin orden
f = {"a":1,"b":2}
print(f,"=",type(f))
print("Valor de la llave 'a'=",f["a"]) #se accede al valor, por el nombre de la␣
 ↪llave
```

```
Lista
[1, 2, 2, 'datos', 1j] = <class 'list'>


Tupla
```

```
1 5.0 False
<class 'tuple'>


Conjunto
{1, 2, 3, 4} = <class 'set'>
{'datos', 1, 2, 1j} = {'datos', 1, 2, 1j}


{'a': 1, 'b': 2} = <class 'dict'>
Valor de la llave 'a'= 1
```

Operadores aritméticos

[4]:
```
x = 4
y = 2
print("Adición x + y = ",x + y)
print("Resta x - y = ",x - y)
print("Producto x * y = ",x * y)
print("División x / y = ",x / y)
print("Módulo x % y = ",x % y)
print("Exponente x ** y = ",x ** y)
print("División entera x // y = ",x // y)
print("XOR x ^ y = ",x ^ y)
```

```
Adición x + y =  6
Resta x - y =  2
Producto x * y =  8
División x / y =  2.0
Módulo x % y =  0
Exponente x ** y =  16
División entera x // y =  2
XOR x ^ y =  6
```

Operadores de asignación

[5]:
```
x = 10
y = 5
x = y
print("x = y equivale a x = y \n x = ",x)
x += y
print("x += y equivale a x = x + y \n x = ",x)
x -= y
print("x -= y equivale a x = x - y \n x = ",x)
x *= y
print("x *= y equivale a x = x * y \n x = ",x)
x /= y
print("x /= y equivale a x = x / y \n x = ",x)
x **= y
```

```
print("x **= y equivale a x = x ** y \n x = ",x)
x //= y
print("x //= y equivale a x = x // y \n x = ",x)
x %= y
print("x %= y equivale a x = x % y \n x = ",x)
```

```
x = y equivale a x = y
 x =  5
x += y equivale a x = x + y
 x =  10
x -= y equivale a x = x - y
 x =  5
x *= y equivale a x = x * y
 x =  25
x /= y equivale a x = x / y
 x =  5.0
x **= y equivale a x = x ** y
 x =  3125.0
x //= y equivale a x = x // y
 x =  625.0
x %= y equivale a x = x % y
 x =  0.0
```

Operadores de comparación

[6]:
```
x = 8
y = 4
print("Igualdad x == y = ",x == y)
print("No Igualdad x == y = ",x != y)
print("Mayor que x > y = ",x > y)
print("Menor que x > y = ",x < y)
print("Mayor o igual que x > y = ",x >= y)
print("Menor o igual que x > y = ",x >= y)
```

```
Igualdad x == y =  False
No Igualdad x == y =  True
Mayor que x > y =  True
Menor que x > y =  False
Mayor o igual que x > y =  True
Menor o igual que x > y =  True
```

Operadores lógicos

[7]:
```
x = 5

print("Producto lógico (x > 5 and x < 10)",x > 5 and x < 10)
print("Suma lógica (x > 5 and x < 10)",x > 5 or x < 10)
print("Negación lógica not (x > 5 and x < 10)",not(x > 5 and x < 10))
```

```
Producto lógico (x > 5 and x < 10) False
Suma lógica (x > 5 and x < 10) True
Negación lógica not (x > 5 and x < 10) True
```

Operadores de identidad

[8]:
```
x = 1
y = 1
print("is devuelve True si ambas variables son el mismo objeto",x is y)
print("is not devuelve True si ambas variables n oson el mismo objeto",x is not␣
 ↪y)
```

```
is devuelve True si ambas variables son el mismo objeto True
is not devuelve True si ambas variables n oson el mismo objeto False
```

Membresía

[9]:
```
x = [1,2,3,4]
print("in devuelve True si una secuencia con el valor especificado se e␣
 ↪encuentra en el objeto", 1 in x)
print("not in devuelve True si ninguna secuencia con el valor especificado se e␣
 ↪encuentra en el objeto", 10 not in x)
```

```
in devuelve True si una secuencia con el valor especificado se e encuentra en el
objeto True
not in devuelve True si ninguna secuencia con el valor especificado se e
encuentra en el objeto True
```

Librerías útiles en el análisis de datos

Existen diferentes bibliotecas del lenguaje de programación de Python que ayudan al proceso de análisis exploratorio y manipulación de datos, las dos más usadas son:

pandas: se trata de una biblioteca que facilita el análisis y la manipulación de datos a través de estructuras y operaciones que pueden ser representadas visualmente en tablas conocidas como frames.

numpy: se trata de una biblioteca diseñada para el manejo de matrices de datos y funciones matemáticas relacionadas con el álgebra lineal..

[10]:
```
#En el lenguaje de programación Python
#la palabra reservada import, importa una biblioteca,
#la palabra reservada as, es un alias de la biblioteca
import pandas as pd
# Lo anterior se lee como: "importa la biblioteca pandas como pd"
```

[11]:
```
import numpy as np
```

1.1.3 Análisis Exploratorio de Datos

Al trabajar con un conjunto de datos, resulta fundamental explorar qué intenta representar la información, con el fin de evaluar la calidad de las entradas y formular una hipótesis precisa.

El análisis exploratorio de datos representa el primer paso en las tareas de Minería de Datos, con dos enfoques principales:

Visualización de datos

Información estadística de los datos

El conjunto de datos Según (Selamat, N., & Ali, F., 2019), para analizar software malicioso en el S.O. Windows existen diferentes formas de representar una muestra: archivos desensamblados, secuencias de llamadas al API, *strings*, recursos del sistema como DLLs y como se analiza en esta Lectura **estructuras del formato PE de un archivo compilado** un poco más del formato en este enlace.

La siguiente línea de código permite importan un conjunto de datos en formato csv, mediante la biblioteca pandas:

[12]:
```
#cargar desde url
data_url = pd.read_csv("https://alhernandezsua.gitlab.io/amd-misti/datasets/
↪dataset_malwares.csv")
```

[13]:
```
data_url.head()
```

[13]:

	Name	e_magic	e_cblp	e_cp	e_crlc	\
0	VirusShare_a878ba26000edaac5c98eff4432723b3	23117	144	3	0	
1	VirusShare_ef9130570fddc174b312b2047f5f4cf0	23117	144	3	0	
2	VirusShare_ef84cdeba22be72a69b198213dada81a	23117	144	3	0	
3	VirusShare_6bf3608e60ebc16cbcff6ed5467d469e	23117	144	3	0	
4	VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb	23117	144	3	0	

```
   e_cparhdr  e_minalloc  e_maxalloc  e_ss  e_sp  …  SectionMaxChar  \
0          4           0       65535     0   184  …      3758096608
1          4           0       65535     0   184  …      3791650880
2          4           0       65535     0   184  …      3221225536
3          4           0       65535     0   184  …      3224371328
4          4           0       65535     0   184  …      3227516992

   SectionMainChar  DirectoryEntryImport  DirectoryEntryImportSize  \
0                0                     7                       152
1                0                    16                       311
2                0                     6                       176
3                0                     8                       155
4                0                     2                        43

   DirectoryEntryExport  ImageDirectoryEntryExport  ImageDirectoryEntryImport  \
0                     0                          0                      54440
1                     0                          0                     262276
2                     0                          0                      36864
3                     0                          0                     356352
4                     0                          0                      61440

   ImageDirectoryEntryResource  ImageDirectoryEntryException  \
0                        77824                         73728
1                       294912                             0
2                        40960                             0
3                      1003520                             0
4                        73728                             0

   ImageDirectoryEntrySecurity
0                            0
1                       346112
2                            0
3                     14109472
4                        90624

[5 rows x 79 columns]
```

[14]:
```
#cargar desde local
data = pd.read_csv('../datasets/dataset_malwares.csv', sep=',')
```

[15]:
```
#Primeras cinco líneas
data.head()
```

[15]:
```
                                Name  e_magic  e_cblp  e_cp  e_crlc  \
0  VirusShare_a878ba26000edaac5c98eff4432723b3    23117     144     3       0
1  VirusShare_ef9130570fddc174b312b2047f5f4cf0    23117     144     3       0
```

```
2  VirusShare_ef84cdeba22be72a69b198213dada81a    23117     144     3       0
3  VirusShare_6bf3608e60ebc16cbcff6ed5467d469e    23117     144     3       0
4  VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb    23117     144     3       0

   e_cparhdr  e_minalloc  e_maxalloc  e_ss  e_sp  …  SectionMaxChar  \
0          4           0       65535     0   184  …      3758096608
1          4           0       65535     0   184  …      3791650880
2          4           0       65535     0   184  …      3221225536
3          4           0       65535     0   184  …      3224371328
4          4           0       65535     0   184  …      3227516992

   SectionMainChar  DirectoryEntryImport  DirectoryEntryImportSize  \
0                0                     7                       152
1                0                    16                       311
2                0                     6                       176
3                0                     8                       155
4                0                     2                        43

   DirectoryEntryExport  ImageDirectoryEntryExport  ImageDirectoryEntryImport  \
0                     0                          0                      54440
1                     0                          0                     262276
2                     0                          0                      36864
3                     0                          0                     356352
4                     0                          0                      61440

   ImageDirectoryEntryResource  ImageDirectoryEntryException  \
0                        77824                         73728
1                       294912                             0
2                        40960                             0
3                      1003520                             0
4                        73728                             0

   ImageDirectoryEntrySecurity
0                            0
1                       346112
2                            0
3                     14109472
4                        90624

[5 rows x 79 columns]
```

[16]:
```
#mostrar las primeras cinco lineas
data.head()
```

[16]:
```
                                  Name  e_magic  e_cblp  e_cp  e_crlc  \
0  VirusShare_a878ba26000edaac5c98eff4432723b3    23117     144     3       0
1  VirusShare_ef9130570fddc174b312b2047f5f4cf0    23117     144     3       0
```

```
2  VirusShare_ef84cdeba22be72a69b198213dada81a    23117     144     3       0
3  VirusShare_6bf3608e60ebc16cbcff6ed5467d469e    23117     144     3       0
4  VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb    23117     144     3       0

   e_cparhdr  e_minalloc  e_maxalloc  e_ss  e_sp  …  SectionMaxChar  \
0          4           0       65535     0   184  …      3758096608
1          4           0       65535     0   184  …      3791650880
2          4           0       65535     0   184  …      3221225536
3          4           0       65535     0   184  …      3224371328
4          4           0       65535     0   184  …      3227516992

   SectionMainChar  DirectoryEntryImport  DirectoryEntryImportSize  \
0                0                     7                       152
1                0                    16                       311
2                0                     6                       176
3                0                     8                       155
4                0                     2                        43

   DirectoryEntryExport  ImageDirectoryEntryExport  ImageDirectoryEntryImport  \
0                     0                          0                      54440
1                     0                          0                     262276
2                     0                          0                      36864
3                     0                          0                     356352
4                     0                          0                      61440

   ImageDirectoryEntryResource  ImageDirectoryEntryException  \
0                        77824                         73728
1                       294912                             0
2                        40960                             0
3                      1003520                             0
4                        73728                             0

   ImageDirectoryEntrySecurity
0                            0
1                       346112
2                            0
3                     14109472
4                        90624

[5 rows x 79 columns]
```

[17]:
```
#muestra las primeras diez filas
data.head(10)
```

[17]:
```
                                   Name  e_magic  e_cblp  e_cp  e_crlc  \
0  VirusShare_a878ba26000edaac5c98eff4432723b3    23117     144     3       0
1  VirusShare_ef9130570fddc174b312b2047f5f4cf0    23117     144     3       0
```

```
2  VirusShare_ef84cdeba22be72a69b198213dada81a    23117     144     3       0
3  VirusShare_6bf3608e60ebc16cbcff6ed5467d469e    23117     144     3       0
4  VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb    23117     144     3       0
5  VirusShare_eff7676f69be2b519f3424def92d3590    23117      80     2       0
6  VirusShare_e76cac211258723745f66bd9f9e29590    23117     144     3       0
7  VirusShare_cef6cdf0e85303a461f67f19ffcc2ddf    23117      80     2       0
8  VirusShare_59af5dfb0c79537eedd3326abde3c857    23117     144     3       0
9  VirusShare_fda0add9d9a8c18c67a758ec2898d976    23117     144     3       0

   e_cparhdr  e_minalloc  e_maxalloc  e_ss  e_sp  …  SectionMaxChar  \
0          4           0       65535     0   184  …      3758096608
1          4           0       65535     0   184  …      3791650880
2          4           0       65535     0   184  …      3221225536
3          4           0       65535     0   184  …      3224371328
4          4           0       65535     0   184  …      3227516992
5          4          15       65535     0   184  …      3221225536
6          4           0       65535     0   184  …      3221225536
7          4          15       65535     0   184  …      3221225536
8          4           0       65535     0   184  …      3221225536
9          4           0       65535     0   184  …      3221225536

   SectionMainChar  DirectoryEntryImport  DirectoryEntryImportSize  \
0                0                     7                       152
1                0                    16                       311
2                0                     6                       176
3                0                     8                       155
4                0                     2                        43
5                0                     8                        96
6                0                     3                        74
7                0                     8                        96
8                0                    14                       329
9                0                     3                        41

   DirectoryEntryExport  ImageDirectoryEntryExport  ImageDirectoryEntryImport  \
0                     0                          0                      54440
1                     0                          0                     262276
2                     0                          0                      36864
3                     0                          0                     356352
4                     0                          0                      61440
5                     0                          0                      53248
6                     0                          0                      45140
7                     0                          0                      53248
8                    16                     533872                     526528
9                     0                          0                      13024

   ImageDirectoryEntryResource  ImageDirectoryEntryException  \
0                        77824                         73728
```

```
1                       294912                                0
2                        40960                                0
3                      1003520                                0
4                        73728                                0
5                        69632                                0
6                        61440                                0
7                        69632                                0
8                       643072                                0
9                        20480                                0

   ImageDirectoryEntrySecurity
0                            0
1                       346112
2                            0
3                     14109472
4                        90624
5                       664744
6                       314368
7                       476984
8                       930128
9                       327680

[10 rows x 79 columns]
```

[18]:
```
#mostrar las últimas cinco lineas
data.tail()
```

[18]:
```
                                      Name  e_magic  e_cblp  e_cp  e_crlc  \
19606                            clip.exe    23117     144     3       0
19607        VNC-Server-6.2.0-Windows.exe    23117     144     3       0
19608  Microsoft.GroupPolicy.Management.ni.dll    23117       0     0       0
19609                    cryptuiwizard.dll    23117     144     3       0
19610                          winhttp.dll    23117     144     3       0

       e_cparhdr  e_minalloc  e_maxalloc  e_ss  e_sp  …  SectionMaxChar  \
19606          4           0       65535     0   184  …      3221225536
19607          4           0       65535     0   184  …      3221225536
19608          0           0           0     0     0  …      3758096448
19609          4           0       65535     0   184  …      3221225536
19610          4           0       65535     0   184  …      3221225536

       SectionMainChar  DirectoryEntryImport  DirectoryEntryImportSize  \
19606                0                     8                        85
19607                0                    10                       391
19608                0                     0                         0
19609                0                    12                       162
19610                0                    35                       226
```

```
       DirectoryEntryExport  ImageDirectoryEntryExport  \
19606                     0                          0
19607                     0                          0
19608                     0                          0
19609                     8                      89008
19610                    58                       8348

       ImageDirectoryEntryImport  ImageDirectoryEntryResource  \
19606                      24948                        28672
19607                    1413420                      1462272
19608                          0                        61440
19609                      94904                       102400
19610                     545812                       557056

       ImageDirectoryEntryException  ImageDirectoryEntrySecurity
19606                             0                            0
19607                             0                     18855424
19608                             0                            0
19609                             0                            0
19610                             0                            0

[5 rows x 79 columns]
```

Pandas genera dos tipos de objetos DataFrame o marco de datos y Series que son una lista seriada de datos. El objeto DataFrame es un diccionario Estructuras de Datos en Python donde cada columna se accede mediante llaves (nombre de columnas) y regresa todos los valores de la misma.

[19]:
```
#Leer columna
data['Name']
```

[19]:
```
0        VirusShare_a878ba26000edaac5c98eff4432723b3
1        VirusShare_ef9130570fddc174b312b2047f5f4cf0
2        VirusShare_ef84cdeba22be72a69b198213dada81a
3        VirusShare_6bf3608e60ebc16cbcff6ed5467d469e
4        VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb
                            …
19606                                       clip.exe
19607                    VNC-Server-6.2.0-Windows.exe
19608         Microsoft.GroupPolicy.Management.ni.dll
19609                                cryptuiwizard.dll
19610                                      winhttp.dll
Name: Name, Length: 19611, dtype: object
```

[20]:
```
data['Malware']
```

[20]:
```
0        1
1        1
2        1
3        1
4        1
        ..
19606    0
19607    0
19608    0
19609    0
19610    0
Name: Malware, Length: 19611, dtype: int64
```

[21]:
```
#Leer fila
lista_dataframe = data.values
print("Primer fila",lista_dataframe[0])
```

```
Primer fila ['VirusShare_a878ba26000edaac5c98eff4432723b3' 23117 144 3 0 4 0
65535 0
 184 0 0 0 64 0 0 0 248 34404 6 1236512358 0 0 240 34 523 8 0 54784 189440
 0 51316 4096 4294967296 4096 512 6 0 6 0 5 2 1024 295281 274432 2 32832
 524288 8192 1048576 4096 0 16 1 0 0 6 0.0 0 512 0 274 0 188416 0 270336 0
 245248 0 3758096608 0 7 152 0 0 54440 77824 73728 0]
```

[22]:
```
#Métodos built-in
print("Calcular la longitud de una lista",len(lista_dataframe[0]))
#print("Calcular la suma de todos los miembros de una␣
 ↪lista",sum(lista_dataframe[0]))
#Deslizar lista
print(lista_dataframe[0][1])
#Deslizar la muestra 0 de la dimensión 1 a la 10 de dos en dos
print(sum(lista_dataframe[0][1:]))
```

```
Calcular la longitud de una lista 79
23117
9293104722.0
```

¿Qué es un conjunto de datos? Un conjunto de datos es una colección de datos El conjunto es de manera común presentado de manera tabular o en forma de matriz ($filas $\times$columnas $). Cada columna describe de manera particular una variable, también denominada característica. Por otro lado, cada fila describe un miembro del conjunto de datos, también denominada muestra u observación.

¿Qué es una característica? Una característica es una propiedad medible de un fenómeno observado

En conjuntos de datos bien definidos, la información de las características son demostradas por las columnas, a esto se le denomina dimensión

En conjuntos de datos no definidos, por ejemplo, aquellos que se conforman por secuencias o grafos, las características deben de ser generadas a partir de los criterios de dimensión de la muestra más larga

</p>

El conjunto de datos dataset_malwares.csv contiene las siguientes dimensiones: $\mathbb{R}^{19611\times79}$, es decir $19,611$ muestras (filas) por 79 columnas.

En la siguiente línea de código se muestra con la biblioteca numpy, como calcular de forma matricial las dimensiones del conjunto de datos:

[23]:
```
np.shape(data)
```

[23]:
```
(19611, 79)
```

Es importante notar que los datos trabajados por las bibliotecas pandas y numpy son compatibles entre ellas. Es decir, no se necesita hacer casting entre tipos de datos

Tipos de conjuntos de datos En estadística existen diferentes tipos de datos disponibles, para diferentes tipos de información:

Numéricos: todos aquellos que manejen exclusivamente valores numéricos

Bivariados: aquellos que contienen dos variables y miden las relaciones entre ellas

Multivariados: un conjunto con diferentes valores pero que distinguen diferentes tipos de relaciones entre ellos

Categóricos: trabaja con datos no numéricos como clases o taxonomías

De correlación: con conjuntos de valores que miden cierta relación de correlación entre ellos

¿Qué tipo de conjunto de datos es csv_malwares.csv? Es un conjunto mixto: numérico y categórico. De hecho es un conjunto diseñado para resolver un problema de Aprendizaje Supervisado es decir, enlaza una etiqueta, respuesta o clase a cada muestra. Su objetivo es realizar una función de inferencia de los datos etiquetados, durante el proceso aprendizaje con el conjunto de entrenamiento. Por convención las muestras del conjunto se denotan por la letra mayúscula X y las etiquetas de cada muestra mediante la letra y.

[24]:
```
# Dividir muestras y clases
# X, denota las muestras
#Elimina las columnas que no representan valores numéricos,
#como el nombre de la muestra
# para las muestras
# y, denota las clases
y = data['Malware']
X = data.drop(['Name','Malware'], axis=1)
```

[25]:
```
#X
```

En la siguiente línea de código se muestran los nombres de las **características** del conjunto de datos X

[26]:
```
# Mostrar las columnas
X.columns
```

[26]:
```
Index(['e_magic', 'e_cblp', 'e_cp', 'e_crlc', 'e_cparhdr', 'e_minalloc',
       'e_maxalloc', 'e_ss', 'e_sp', 'e_csum', 'e_ip', 'e_cs', 'e_lfarlc',
       'e_ovno', 'e_oemid', 'e_oeminfo', 'e_lfanew', 'Machine',
       'NumberOfSections', 'TimeDateStamp', 'PointerToSymbolTable',
       'NumberOfSymbols', 'SizeOfOptionalHeader', 'Characteristics', 'Magic',
       'MajorLinkerVersion', 'MinorLinkerVersion', 'SizeOfCode',
       'SizeOfInitializedData', 'SizeOfUninitializedData',
       'AddressOfEntryPoint', 'BaseOfCode', 'ImageBase', 'SectionAlignment',
       'FileAlignment', 'MajorOperatingSystemVersion',
       'MinorOperatingSystemVersion', 'MajorImageVersion', 'MinorImageVersion',
       'MajorSubsystemVersion', 'MinorSubsystemVersion', 'SizeOfHeaders',
       'CheckSum', 'SizeOfImage', 'Subsystem', 'DllCharacteristics',
       'SizeOfStackReserve', 'SizeOfStackCommit', 'SizeOfHeapReserve',
       'SizeOfHeapCommit', 'LoaderFlags', 'NumberOfRvaAndSizes',
       'SuspiciousImportFunctions', 'SuspiciousNameSection', 'SectionsLength',
       'SectionMinEntropy', 'SectionMaxEntropy', 'SectionMinRawsize',
       'SectionMaxRawsize', 'SectionMinVirtualsize', 'SectionMaxVirtualsize',
       'SectionMaxPhysical', 'SectionMinPhysical', 'SectionMaxVirtual',
       'SectionMinVirtual', 'SectionMaxPointerData', 'SectionMinPointerData',
       'SectionMaxChar', 'SectionMainChar', 'DirectoryEntryImport',
       'DirectoryEntryImportSize', 'DirectoryEntryExport',
       'ImageDirectoryEntryExport', 'ImageDirectoryEntryImport',
       'ImageDirectoryEntryResource', 'ImageDirectoryEntryException',
       'ImageDirectoryEntrySecurity'],
      dtype='object')
```

En la siguiente línea de código se muestra con la biblioteca **numpy**, como calcular de forma matricial las dimensiones del conjunto de datos X si nvalores no numéricos ni la etiqueta de clase y:

[27]:
```
# dimension del conjunto de datos real,
# mediante el método shape de numpy
np.shape(X)
```

[27]:
```
(19611, 77)
```

¿Cuántas muestras de software benigno y maligno existen?

[28]:
```
# Longitud de la lista de clases,
# en el lenguaje de programación Python los denominados "arreglos",
# son llamados listas
# El método len devuelve el tamaño de una lista
```

```
# Investigar que son las métodos o funciones built-in del lenguaje: https://
  ↪docs.python.org/3/library/functions.html
muestras = len(y)
```

[29]:
```
muestras
```

[29]:
```
19611
```

¿Qué son las etiquetas o clases? La etiqueta es la salida final de la muestra, es lo que representa o la categoría de clase a la que pertenece

A veces las etiquetas o clases contienen categorías, a lo que les llamamos etiquetas categóricas, como por ejemplo: $\{H, M\}$, para distinguir entre Hombre y Mujer ó $\{B, M\}$ para distinguir entre benigno y maligno Otro tipos de etiquetas o clases son denominadas discretas y representan valores numéricos bien separables

Tipos de datos Cada muestra (fila) del conjunto X toma valores en sus columnas (características) con diferentes unidades, de manera distinta, dependiendo qué es lo que representen:

1. Cualitativa descrita por una palabra o frase (p. ej. tipo de sangre, género o color)
2. Cuantitativa descrita por un número (p. ej. días de la semana, número de vuelos que salen a cierta hora o el número de personas que toman el metro al día)
3. Ordinal: las observaciones no son números, pero pueden ser ordenadas (p. ej. el servicio de un restaurante puede ser ordenado como excelente, bien, regular, mal o muy mal)

Los datos cuantitativos pueden ser:

Discretos: la variable solo puede tomar un número finito de valores contables (p. ej. número de lista de los alumnos en una clase)

Continuos la variable es una medida que puede tomar cualquier valor en un intervalo de los números reales (p. ej. los pesos de las personas)

La siguiente línea de código muestra los diferentes tipos de datos para el conjunto X, onde en su mayoría, representan valores enteros int64:

[30]:
```
X.dtypes
```

[30]:
```
e_magic                        int64
e_cblp                         int64
e_cp                           int64
e_crlc                         int64
e_cparhdr                      int64
                               ...
ImageDirectoryEntryExport      int64
ImageDirectoryEntryImport      int64
ImageDirectoryEntryResource    int64
ImageDirectoryEntryException   int64
ImageDirectoryEntrySecurity    int64
```

```
Length: 77, dtype: object
```

[31]:
```
#El método unique permite listar los valores únicos de una lista
np.unique(X.dtypes)
```

[31]:
```
array([dtype('int64'), dtype('float64')], dtype=object)
```

Propiedades del conjunto de datos Antes de realizar un análisis de minería de datos, el análisis exploratorio ayudará a entender las propiedades de los datos, para poder identificar que algoritmo será el más apropiado para trabajar con los mismos. Algunas propiedades que deben de analizarse son la siguientes:

Muestras y poblaciones

Medidas de la tendencia central

Medidas de dispersión

Oblicuidad

Presencia de extremos

Correlación entre los datos

Algunos conceptos que deben de considerarse son los siguientes:

Población Son todos los individuos o unidades de interés. De manera típica, no existen los suficientes datos para representar a todos los individuos de una población.

Muestra Es un subconjnunto de todos los individuos o unidades en una población. De manera típica sí hay datos suficientes para todos individuos dentro de una muestra.

Variable aleatoria Es una variable cuyo valor es desconocido o una función que asigna valores a la salidas de un experimento.

Distribución Normal La distribución normal se trata de una variable aleatoria continua, la cual es muy utilizada, ya que su distribución se aproxima a muchos fenómenos naturales. Es por esto, que se ha desarrollado como estándar de referencia para muchos problemas de probabilidad y estadística.

La distribución normal está compuesta por:

Frecuencia absoluta (f): numero de veces que aparece un valor

Frecuencia relativa (n): el resultado de dividir la frecuencia absoluta de un determinado valor entre el número total de datos

Frecuencia acumulada (F): la suma de las frecuencias absolutas de todos los valores, iguales o inferiores al valor considerado

Intervalo de clase rango utilizado para dividir el conjunto de posibles valores numéricos al trabajar con grandes cantidades de datos

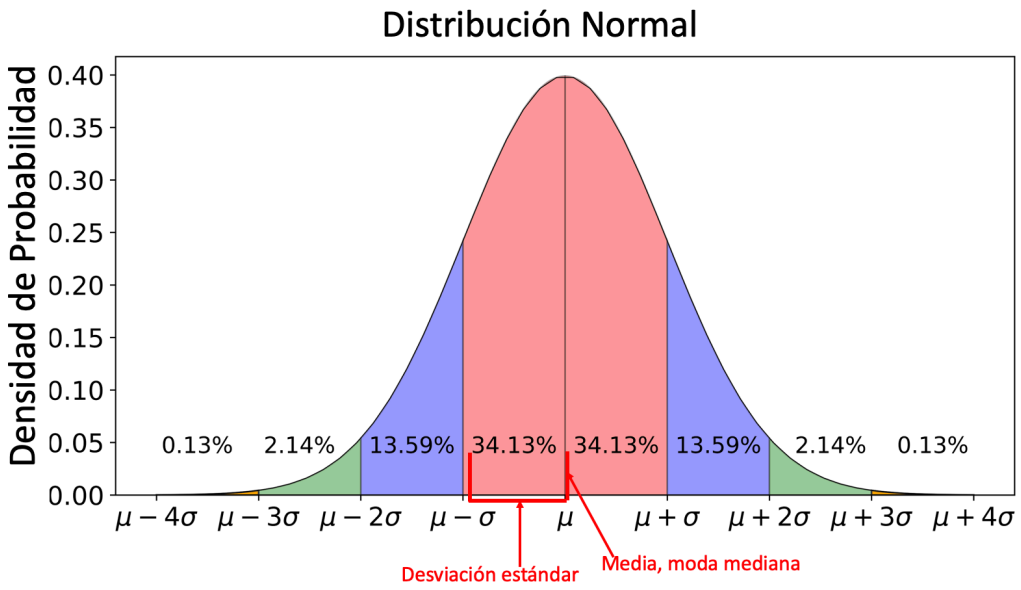
Función de densidad de probabilidad es una función cuyo valor es dado por cualquier observación en el espacio de muestras, es decir el conjunto de valores que toma la variable aleatoria. En una distribución normal toma la forma de una campana

```
<li>   <b>Media</b> ($\mu$): también conocida como promedio, es una medida del centro del conju
<li>  <b>Mediana</b>: es el valor medio, es decir el valor central con datos ordenados</li>
<li>  <b>Moda</b>: es el valor que ocurre de manera más común</li>
```

Desviación estándar (σ): indica que tan dispersos están los datos con respecto a la media. Cuanto mayor sea su valor, la dispersión de la variable es mas alta

Varianza (σ^2): identifica la diferencia promedio de cada valor con respecto a la media (el valor cuadrado tiene como fin eliminar cualquier valor que pudiera ser negativo)

En la siguiente Figura se muestra un ejemplo de la distribución normal:



El siguiente código muestra algunas medidas de la tendencia central del conjunto de datos X:

[32]:
```
# Mostrar los tipos de datos en el conjuto
X.describe()
```

[32]:

	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	\
count	19611.0	19611.000000	19611.000000	19611.000000	19611.000000	
mean	23117.0	178.615726	71.660752	49.146958	37.370710	
std	0.0	987.200729	1445.192977	1212.201919	864.515405	
min	23117.0	0.000000	0.000000	0.000000	0.000000	
25%	23117.0	144.000000	3.000000	0.000000	4.000000	
50%	23117.0	144.000000	3.000000	0.000000	4.000000	
75%	23117.0	144.000000	3.000000	0.000000	4.000000	
max	23117.0	59448.000000	63200.000000	64613.000000	43690.000000	

```
         e_minalloc    e_maxalloc          e_ss         e_sp        e_csum  \
count  19611.000000  19611.000000  19611.000000  19611.00000  19611.000000
mean      37.032635  64178.739687     10.418490    226.46530     29.689103
std      915.833139   9110.755873    637.116265   1249.68033   1015.303419
min        0.000000      0.000000      0.000000      0.00000      0.000000
25%        0.000000  65535.000000      0.000000    184.00000      0.000000
50%        0.000000  65535.000000      0.000000    184.00000      0.000000
75%        0.000000  65535.000000      0.000000    184.00000      0.000000
max    43690.000000  65535.000000  61436.000000  65464.00000  63262.000000

       …  SectionMaxChar  SectionMainChar  DirectoryEntryImport  \
count  …    1.961100e+04          19611.0          19611.000000
mean   …    3.163632e+09              0.0              6.112437
std    …    5.860332e+08              0.0              7.525158
min    …    1.073742e+09              0.0              0.000000
25%    …    3.221226e+09              0.0              2.000000
50%    …    3.221226e+09              0.0              4.000000
75%    …    3.221226e+09              0.0              8.000000
max    …    4.294967e+09              0.0            588.000000

       DirectoryEntryImportSize  DirectoryEntryExport  \
count              19611.000000          19611.000000
mean                 101.912804             14.131865
std                  127.076767            154.958102
min                    0.000000              0.000000
25%                   17.000000              0.000000
50%                   80.000000              0.000000
75%                  135.000000              0.000000
max                 4016.000000           7319.000000

       ImageDirectoryEntryExport  ImageDirectoryEntryImport  \
count               1.961100e+04               1.961100e+04
mean                3.368566e+05               4.047213e+05
std                 2.181191e+07               4.704601e+06
min                 0.000000e+00               0.000000e+00
25%                 0.000000e+00               2.960400e+04
50%                 0.000000e+00               8.222400e+04
75%                 0.000000e+00               2.477080e+05
max                 2.147484e+09               5.368914e+08

       ImageDirectoryEntryResource  ImageDirectoryEntryException  \
count                 1.961100e+04                  1.961100e+04
mean                  5.555810e+05                  1.238834e+06
std                   6.772167e+06                  5.868961e+07
min                   0.000000e+00                  0.000000e+00
25%                   4.096000e+04                  0.000000e+00
50%                   1.310720e+05                  0.000000e+00
```

```
75%                    3.870720e+05                    0.000000e+00
max                    8.304108e+08                    2.906159e+09

       ImageDirectoryEntrySecurity
count                  1.961100e+04
mean                   8.814868e+05
std                    2.167579e+07
min                    0.000000e+00
25%                    0.000000e+00
50%                    0.000000e+00
75%                    3.184640e+05
max                    2.415919e+09

[8 rows x 77 columns]
```

Matplotlib, una herramienta útil para la visualización de datos Otra de las bibliotecas que serán utilizadas ampliamente, en este curso será matplotlib, diseñada para la generación de gráficos, a partir de datos contenidos en listas.

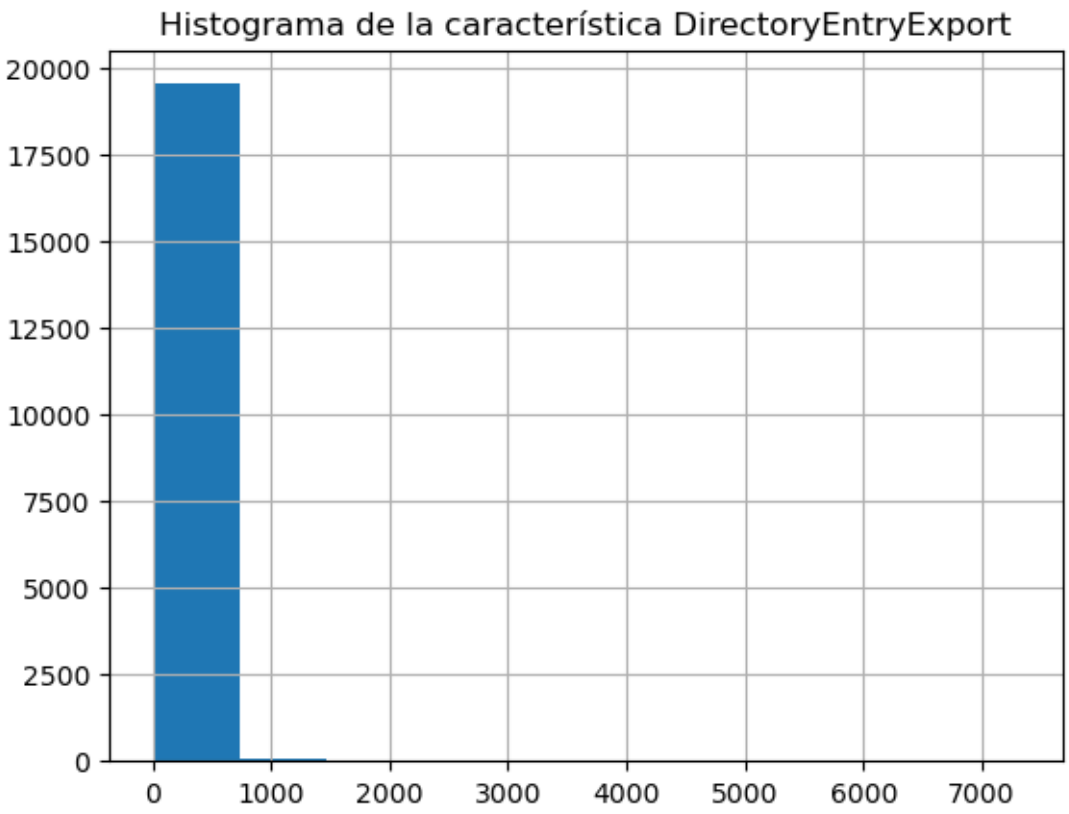
[33]:
```
import matplotlib.pyplot as plt
```

[34]:
```
#Calcular la frecuencia absoluta de una característica
DirectoryEntryExport = X.groupby('e_cparhdr').
 ↪agg(frequency=('e_cparhdr',"count"))
```

[35]:
```
DirectoryEntryExport.to_csv("tabla_de_frecuencias_e_cparhdr.csv")
```

El apoyo de un histograma Un histograma es una reprentación gráfica de los datos. Los datos en forma de matriz se pueden representar por la frecuencia de ocurrencia y categorías.

[36]:
```
hist,bin_edges = np.histogram(X['DirectoryEntryExport'],10)
plt.hist(X['DirectoryEntryExport'].values)
plt.title("Histograma de la característica DirectoryEntryExport")
plt.grid()
plt.show()
```



Las medidas de la tendencia central: Como se mencionó en las propiedades de la distribución normal, existen una serie cálculos, de los cuales, los primeros,son denominados medidas de la tendencia central (media, mediana, moda), es decir, valores que describen un conjunto de datos, identificando su posición central, dentro del mismo.

Una de las bibliotecas en el lenguaje de programación Python que incluye en gran medida muchos cálculos estadísticos es statistics. Para instalar un módulo en Python, se debe de utilizar el comando pip, que es el gestor de paquetes del lenguaje.

Si Python3 es el interprete por defecto:

pip install statistics

Si Python3 no es el interprete por defecto:

pip3 install statistics

[37]:
```
%pip install statistics
```

```
Requirement already satisfied: statistics in
/Users/aldohernandez/anaconda3/lib/python3.11/site-packages (1.0.3.5)
Requirement already satisfied: docutils>=0.3 in
/Users/aldohernandez/anaconda3/lib/python3.11/site-packages (from statistics)
```

```
(0.18.1)
Note: you may need to restart the kernel to use updated packages.
```

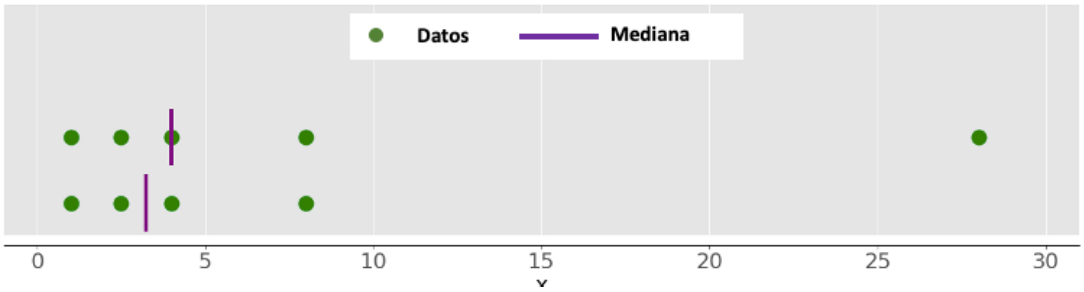
[38]:
```
import statistics

media = statistics.mean(X['DirectoryEntryExport'])
print("La media es:",media)
mediana = statistics.median(X['DirectoryEntryExport'])
print("La mediana es:",mediana)
moda = statistics.mode(X['DirectoryEntryExport'])
print("La moda es:",moda)
```

```
La media es: 14.131864769772067
La mediana es: 0
La moda es: 0
```

[39]:
```
#X.columns
```

Consideraciones de la media El cálculo de la media es sensible a variaciones de valores extremos. Valores desproporcionados no pueden mostrar realmente la media de los datos.

Consideraciones de la mediana Por otro lado la mediana no es sensible a extremos. Es el punto medio en la distribución de los datos, divida en dos partes, con una mitad igual o menor al punto medio y el otro igual o mayor.



Si la característica/atributo tiene un número de muestras impares, la mediana estará exactamente a la mitad de los valores ordenados

Si la característica/atributo tiene un número de muestras pares, la mediana será el promedio de los dos valores que se encuentran en la mitad de los valores

Medidas de dispersión Para poder entender los datos, se necesita tener conciencia de la dispersión o variabilidad de los datos, es decir que tan lejos y distribuidos están los datos ¿Se podrá dar una idea el analista de malware acerca de transacciones fuera una área normal de los patrones regulares?

Derivaciones de la media Es aquella derivación que explica que tan lejos está un valor de la media de una distribución.

1. Se calcula sustrayendo las medias de cada valor individual
2. Los valores por debajo de la media resultan en **diferencias negativas**
3. Los valores por encima de la media, indican **diferencias positivas**
4. La suma total de las diferencias es siempre **cero**

La desviación de la media absoluta

1. Describe que tan lejos está un número de la media
2. Se calcula sustrayendo la media de cada valor individual, pero las diferencias se presentan en **valores absolutos**, es decir, los signos negativos son ignorados
3. La suma de las diferencias son calculadas y divididas por el número de registros en la distribución

La varianza

1. Es una medida de dispersión que elimina el factor de las diferencias totales hacia cero y disminuye el efecto de los números negativos
2. Se calcula mediante el cuadrado de cada diferencia (valor de la característica/atributo menos la media), tomando el total de las diferencias cuadráticas y dividiéndolas por el total de las muestras

La desviación estándar

1. Es una una de las medidas de variabilidad más utilizadas
2. Es la distancia de cada valor desde el centro de la media
3. Se calcula aplicando la raíz cuadrada a la varianza
4. Es útil para indicar identificar la variabilidad en la distribución de los datos
5. Entre más lejos esté el valor de la media, mayor será el valor de la desviación

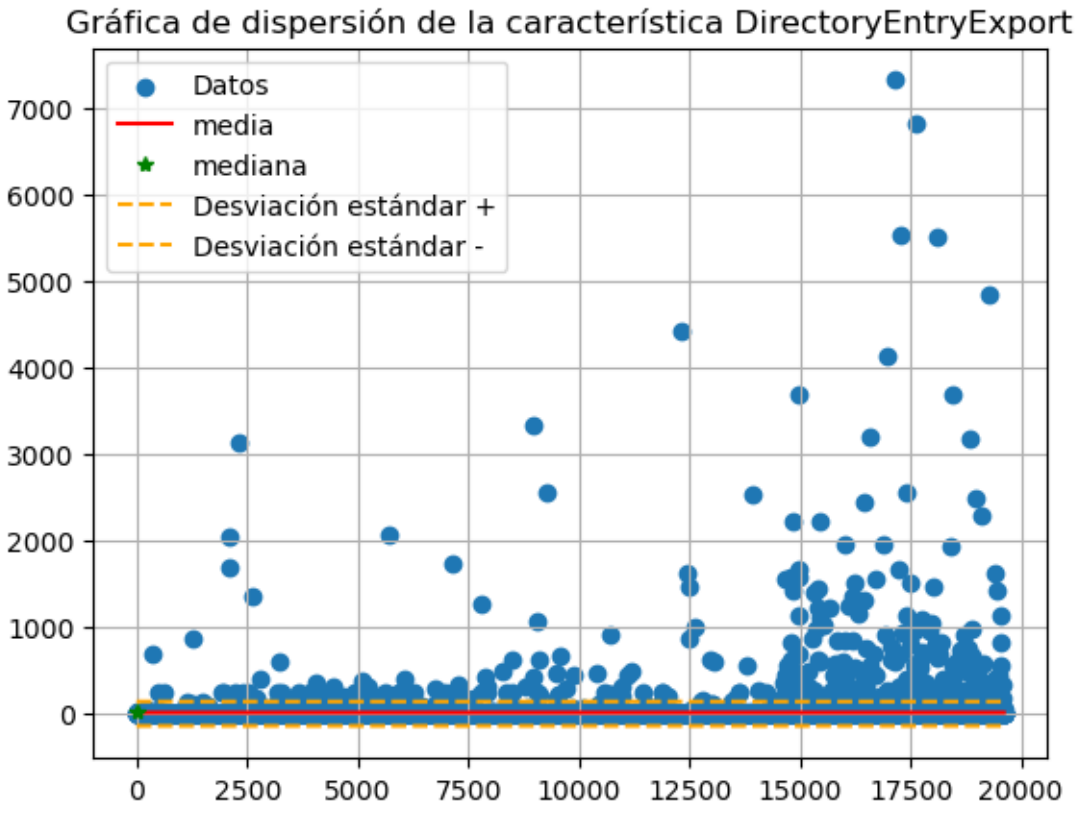
[40]:
```
devest = statistics.stdev(X['DirectoryEntryExport'])
var = statistics.variance(X['DirectoryEntryExport'])
print("La varianza es:",var)
print("La desviación estándar es",devest)
```

```
La varianza es: 24012.01341140772
La desviación estándar es 154.9581021160485
```

[41]:
```
# Graficar las medidas de la población
# Multiplicar un valor n veces, es decir de la longitud del vector de datos␣
 ↪X['DirectoryEntryExport']
longitud_datos = len(X['DirectoryEntryExport'])
# Genear los índices en el eje x [1,...,longitud_datos]
indices = range(0,longitud_datos)
media_grafica = [media]*longitud_datos
devest_grafica_pos = [devest-media]*longitud_datos
devest_grafica_neg = [(media-devest)]*longitud_datos
```

[42]:
```
#scatter es un método para graficar la dispersión de los datos
plt.scatter(indices,X['DirectoryEntryExport'],label="Datos")
```

```
plt.plot(indices,media_grafica,color="red",label="media")
plt.plot(mediana,media,'*',label='mediana',color="green")
plt.plot(indices,devest_grafica_pos,'--',label='Desviación estándar␣
 ↪+',color="orange")
plt.plot(indices,devest_grafica_neg,'--',label='Desviación estándar␣
 ↪-',color="orange")
#muestra las etiquetas de los datos
plt.legend()
#muestra el enrejado
plt.grid()
#Título del plot
plt.title('Gráfica de dispersión de la característica DirectoryEntryExport')
#Descripción en el eje de las y
#Los gráficos se pueden guardar mediante el método savefig
plt.savefig("miprimergafico.png",dpi=300)
plt.show()
```



[43]:
```
#Ejemplo de Derivación de la media de los primeros diez datos de␣
 ↪DirectoryEntryExport
for dato in X['DirectoryEntryExport'][0:10]:
    print("Derivación de la media:",dato-media)
```

```
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: -14.131864769772067
Derivación de la media: 1.8681352302279333
Derivación de la media: -14.131864769772067
```

[44]:
```
#Ejemplo de Derivación de la media absoluta de los primeros diez datos de␣
 ↪DirectoryEntryExport
derivacion_total = []
for dato in X['DirectoryEntryExport'][0:10]:
    print("Derivación de la media absoluta:",abs(dato-media))
    #append añade un valor a una lista
    derivacion_total.append(abs(dato-media))
```

```
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 14.131864769772067
Derivación de la media absoluta: 1.8681352302279333
Derivación de la media absoluta: 14.131864769772067
```

[45]:
```
sum(derivacion_total)
sum([1,2,3])
lista = [1,2]
#añade el nuevo valor al final de la lista
lista.append(1)
print(lista)
```

```
[1, 2, 1]
```

La derivación total se calcula de la siguiente forma:

$$\sum_{\forall i} \frac{|(x_i - \mu)|}{N} \tag{1}$$

[46]:
```
#El método sum regresa la suma de todos los valres de una lita
total = sum(derivacion_total)/len(derivacion_total)
print("La derivación absoluta total es:",total)
```

```
La derivación absoluta total es: 12.905491815817655
```

Diagramas de caja Son útiles para entender las medidas centrales, pero además para obtener información acerca de la dispersión de los datos. El diagrama de caja de construye mediante los siguientes pasos:

Ordenar los datos de menor a mayor valor

Determinar la media de los datos

Encontrar el primer cuartil (Q1), es decir 25% de los puntos ordenados en la secuencia

Encontrar el tercer cuartil (Q3), es decir 75% de los datos ordenados

Encontrar el valor máximo y mínimo

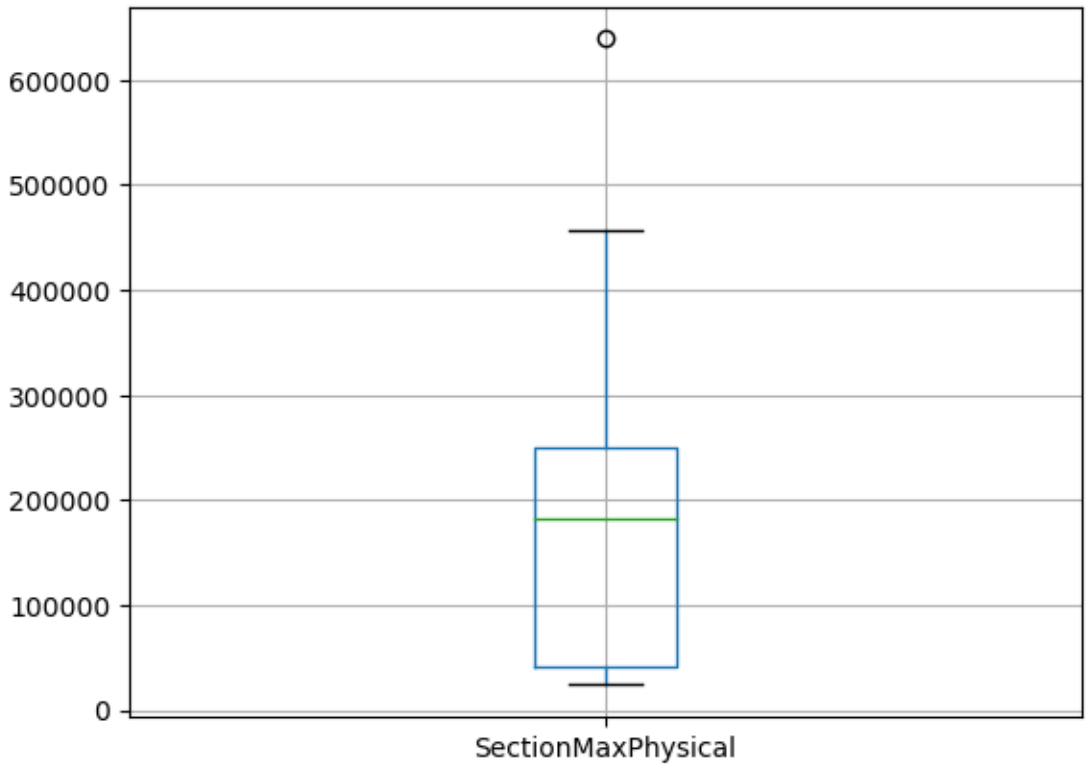
Describir una línea horizontal, que represente la escala de valores la característica/atributo

Marcar el punto de la mediana y la media de los pasos uno y dos, construyendo así la caja

Añadir el valores máximo/mínimo y construir los límites

*Los puntos atípicos representan valores que numéricamente son distantes al resto de los datos.

[47]:
```
pd.DataFrame(X['SectionMaxPhysical'][0:10]).boxplot();
```



Oblicuidad Una oblicuidad negativa indica que existen valores dominantes en el lado izquierdo de la mediana

Una oblicuidad positiva indica una mayor proporción de valores en el lado derecho de la mediana

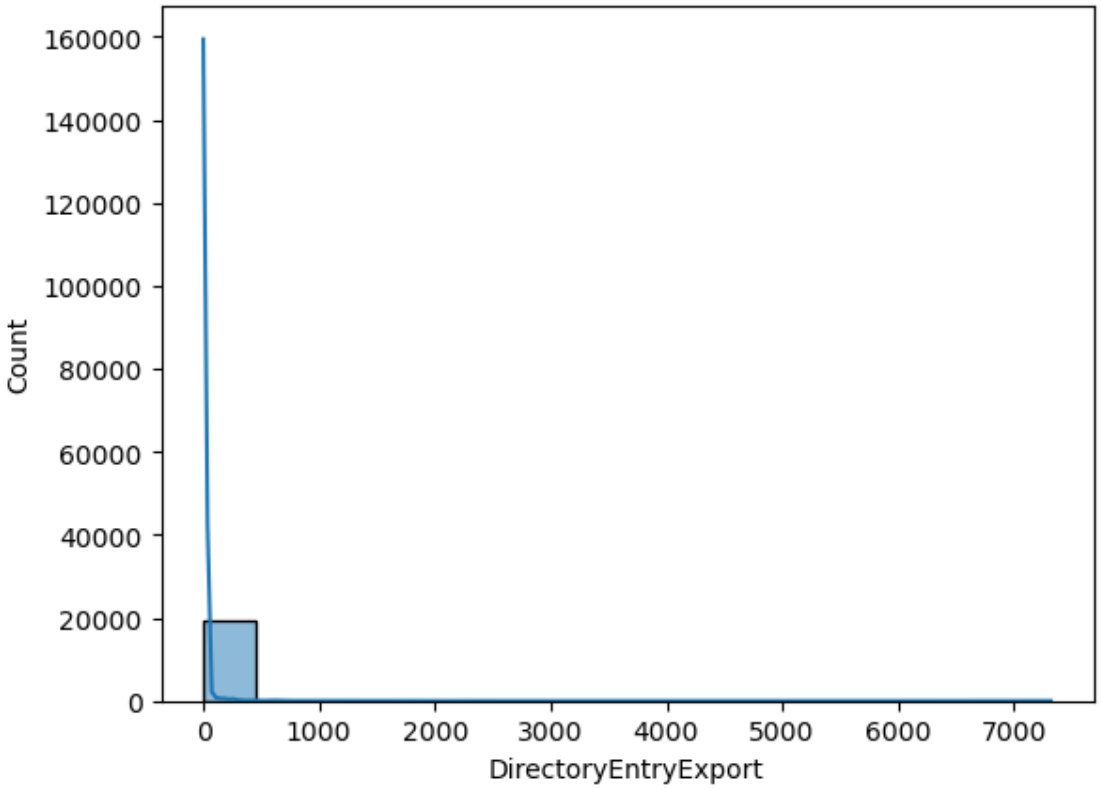
Entre el valor de la oblicuidad tienda más a cero, se concluye el atributo/característica es simétrico en sus valores

[48]:
```
print("La oblicuidad es:",X['DirectoryEntryExport'].skew())
```

```
La oblicuidad es: 24.812096643292783
```

Seaborn Seanborn es una biblioteca de visualización de datos para la representación de patrones, basado en matplotlib

[49]:
```
import seaborn as sns

sns.histplot(X['DirectoryEntryExport'],kde=True);
```



1.2 Referencias

1. Selamat, N., & Ali, F. (2019). Comparison of malware detection techniques using machine learning algorithm. Indones. J. Electr. Eng. Comput. Sci, 16, 435.

[ ]:

[ ]:

[ ]: