

## tablas

March 22, 2022

**2.3 Ejemplo** En un proceso de control de calidad, durante un periodo de tiempo determinado, se tiene un conjunto de 7,614 lotes de cien resistencias (resistores) cada uno. Para cada lote se registra el número de resistencias con algún tipo de inconformidad, obteniéndose los resultados de la tabla que se presenta a continuación. Completar la tabla de frecuencias.

Clase	F.Absoluta	F.Relativa
0	145	
1	2415	
2	3456	
3	852	
4		0.0603
5	157	
6		
Total	7,614	

```
[ ]: # Tamaño de la muestra
n = 7614
# Frecuencia absoluta clase 4
n_4 = 0.0603*n
print(n_4)
```

459.1242

```
[ ]: # Frecuencia absoluta clase 6
n_6 = n - sum([145, 2415, 3456, 852, 459 + 157])
print(n_6)
```

130

```
[ ]: # Frecuencias absolutas
nj = [145, 2415, 3456, 852, 459, 157, 130]
# Frecuencias relativas
hj = list(map(lambda x: round(x/n, 3), nj))
print(hj)
```

[0.019, 0.317, 0.454, 0.112, 0.06, 0.021, 0.017]

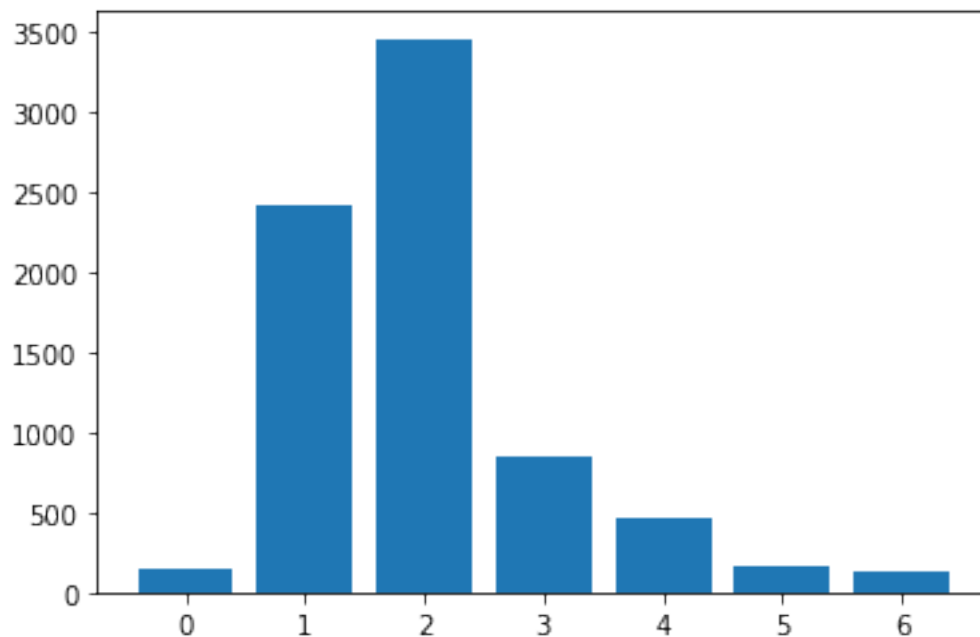
**2.4 Análisis gráfico** En general las variables categóricas y las cuantitativas discretas se grafican por medio de diagramas de barras:

- Eje x categorías de la característica observada.
- Eje y frecuencia absoluta o frecuencia relativa.

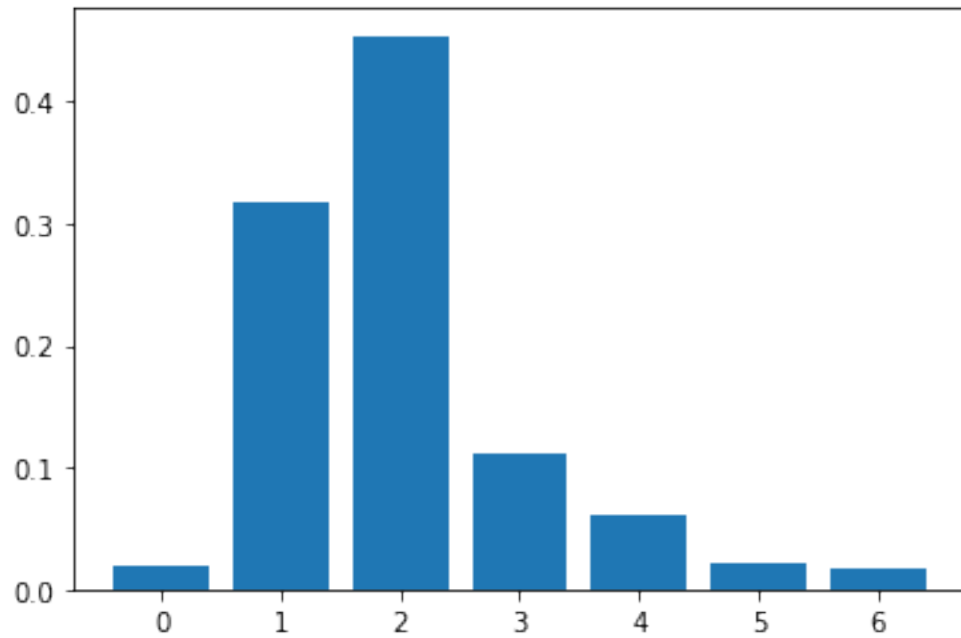
## 2.5 Ejemplo Graficar las frecuencias relativas del ejemplo de las resistencias.

```
[ ]: import numpy as np
import matplotlib.pyplot as plt

# Tamaño de la muestra
n = 7614
# Frecuencias absolutas
nj = np.array([145, 2415, 3456, 852, 459, 157, 130])
# Hacer una grafica de barras
plt.bar(np.arange(len(nj)), nj)
# plt.xticks(np.arange(len(nj)), ['145', '2415', '3456', '852', '459', '157', '130'])
# Mostrar la grafica
plt.show()
```

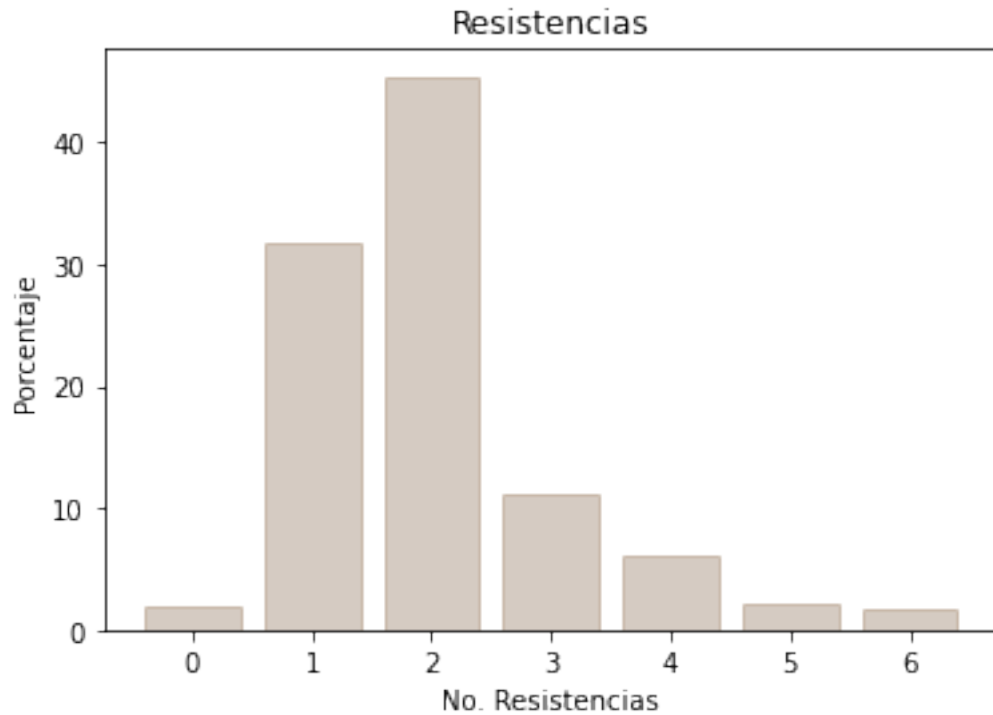


```
[ ]: # Frecuencias relativas
hj = np.array(list(map(lambda x: round(x/n, 3), nj)))
plt.bar(np.arange(len(hj)), hj)
plt.show()
```



#### AÑADIR DISEÑO A LA GRAFICA

```
[ ]: plt.bar(np.arange(len(hj)), list(map(lambda x: x*100 ,hj)), color=(0.32, 0.16, 0.40, 0.24), edgecolor=(0.40, 0.20, 0, 0.24))
plt.title("Resistencias")
plt.xlabel("No. Resistencias")
plt.ylabel("Porcentaje")
plt.show()
```



#### *Frecuencias absolutas y relativas acumuladas*

**2.8.1 Ejemplo** Considerar el siguiente conjunto de datos asociados con el nivel educativo de una muestra de empleados (Bachillerato (B), Pregrado (P), Maestría (M), y Doctorado (D)). Elaborar la tabla de frecuencias correspondiente.

B, D, M, B, B, P, B, M, B, B, B, P, B, M, B, B, M, B, M, B, B, B, B, B, B, P, B, B, B, B, M, B, P, B, B, M, B, B, B, D, B, M, B, P, B, B, B, P, P

Clase	F.Absoluta	F.Relativa	F.Abs.Acumulada	F.Rel.Acumulada
Bachillerato	33	66.0%	33	66.0%
Pregrado	7	14.0%	40	80.0%
Maestría	8	16.0%	48	96.0%
Doctorado	2	4.0%	50	100%
Total	50	100%	N.A.	N.A.

```
[ ]: # Datos
edu = ["B", "D", "M", "B", "B", "P", "B", "M", "B", "B", "B", "P", "B", "M",
      "B", "B", "M", "B", "M", "B", "B", "B", "B", "B", "B", "B", "P", "B",
      "B", "B", "B", "M", "B", "P", "B", "B", "M", "B", "B", "B", "D", "B",
      "M", "B", "P", "B", "B", "B", "P", "P"]

# Tamaño de la muestra
n = len(edu)
print(n)
```

50

```
[ ]: # Importar modulo pandas pandas
import pandas as pd
```

```
[ ]: # Frecuencias absolutas
ps = pd.Series(edu)
nj = ps.value_counts()
print(nj)
```

```
B    33
M     8
P     7
D     2
dtype: int64
```

```
[ ]: nj = nj.reindex(["B", "P", "M", "D"])
print(nj)
```

```
B    33
P     7
M     8
D     2
dtype: int64
```

```
[ ]: # Frecuencias relativas (hj)
hj = nj.div(n)
print(hj)
```

```
B    0.66
P    0.14
M    0.16
D    0.04
dtype: float64
```

```
[ ]: # Frecuencias absolutas acumuladas
Nj = nj.cumsum()
print(Nj)
```

```
B    33
P    40
M    48
D    50
dtype: int64
```

```
[ ]: # Frecuencias relativas acumuladas
Hj = hj.cumsum()
print(Hj)
```

```
B    0.66
```

```
P    0.80
M    0.96
D    1.00
dtype: float64
```

```
[ ]: table = pd.DataFrame({'nj':nj,'hj':hj,'Nj': Nj, 'Hj': Hj})
print(table)
```

```
   nj  hj  Nj  Hj
B  33  0.66  33  0.66
P   7  0.14  40  0.80
M   8  0.16  48  0.96
D   2  0.04  50  1.00
```

```
[ ]: nombres_f = ["Bachillerato", "Pregrado", "Maestria", "Doctorado"]
nombres_c = ["F. Absoluta", "F. Relativa", "F. Abs. Acumulada", "F. Rel. Acumulada"]
table.columns = nombres_c
table.index = nombres_f
print(table)
```

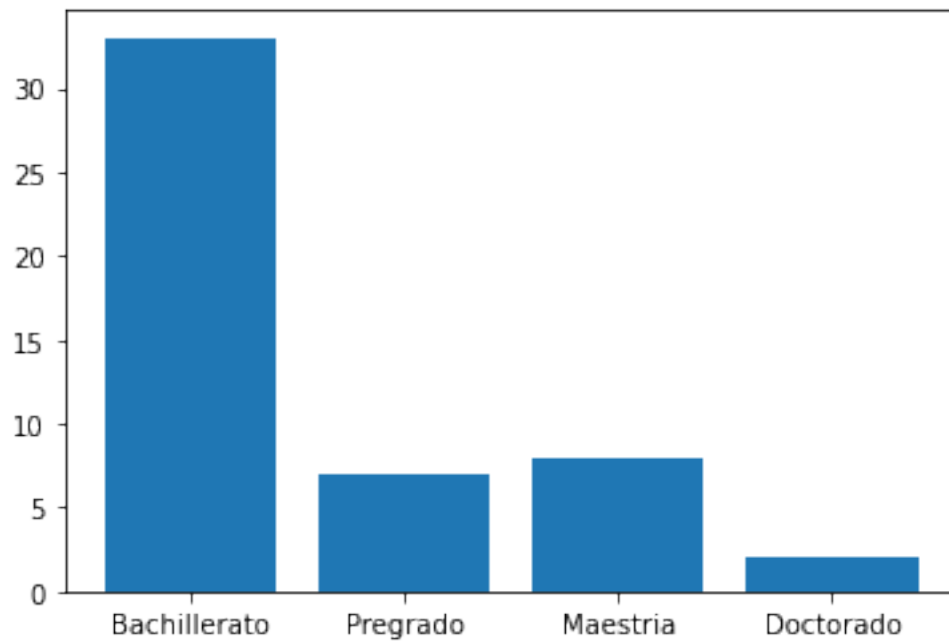
	F. Absoluta	F. Relativa	F. Abs. Acumulada	F. Rel. Acumulada
Bachillerato	33	0.66	33	0.66
Pregrado	7	0.14	40	0.80
Maestria	8	0.16	48	0.96
Doctorado	2	0.04	50	1.00

```
[ ]: # Print table using with markdown style (In R using ktnir)
print(table.to_markdown())
```

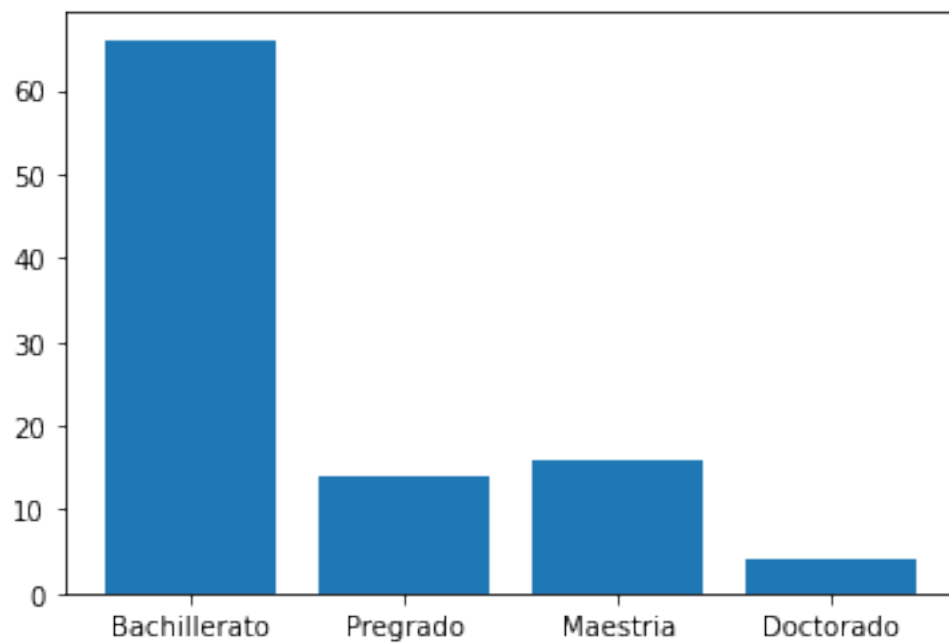
	F. Absoluta	F. Relativa	F. Abs. Acumulada	F. Rel. Acumulada
Bachillerato	33	0.66	33	0.66
Pregrado	7	0.14	40	0.8
Maestria	8	0.16	48	0.96
Doctorado	2	0.04	50	1

*Graficas de las frecuencias*

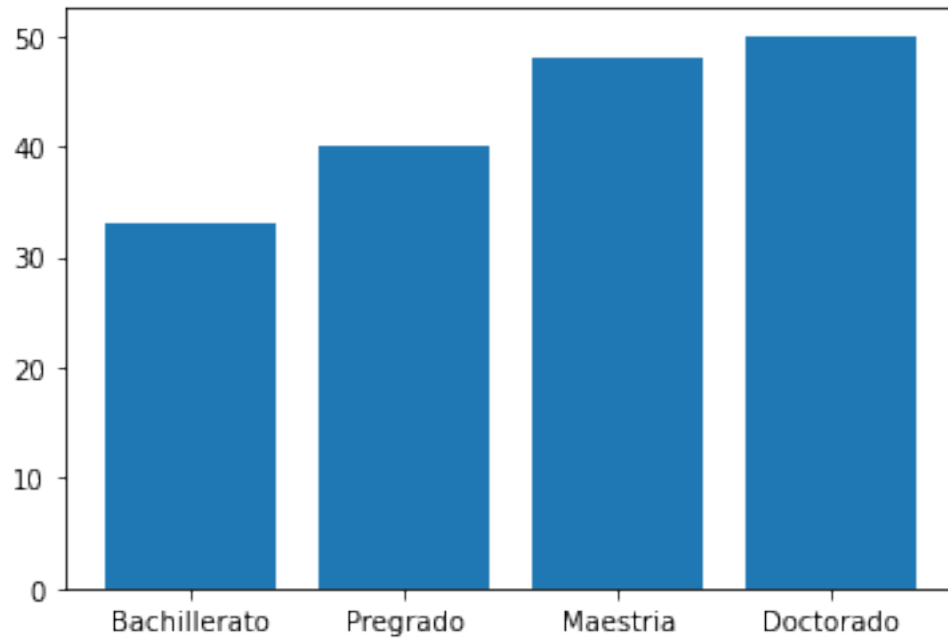
```
[ ]: # Frecuencia absoluta
plt.bar(np.arange(len(nj)), nj)
plt.xticks(np.arange(len(nj)), nombres_f)
plt.show()
```



```
[ ]: # Frecuencia relativa (%)  
plt.bar(np.arange(len(hj)), [x*100 for x in hj])  
plt.xticks(np.arange(len(hj)), nombres_f)  
plt.show()
```

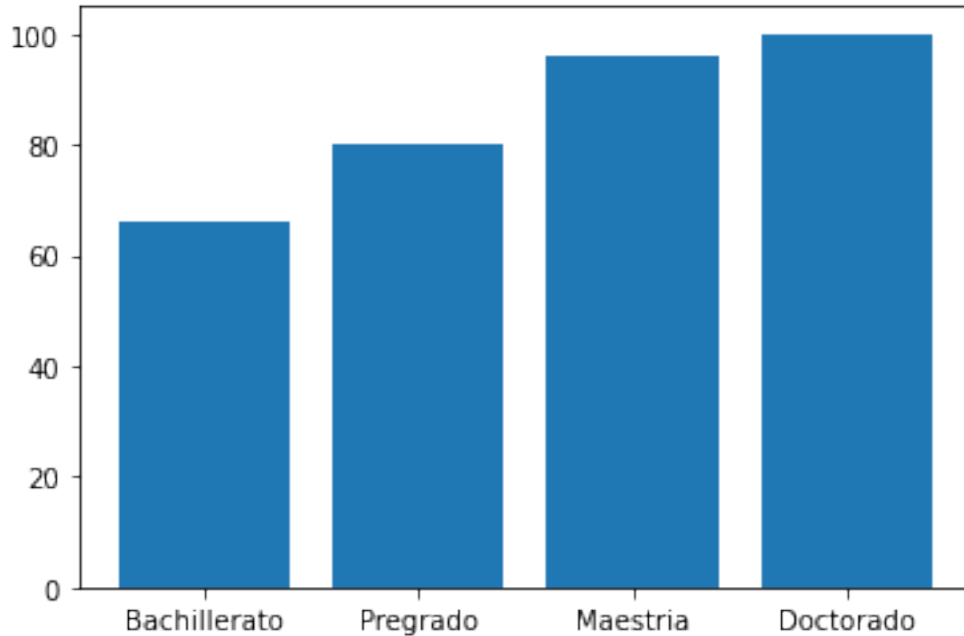


```
[ ]: # Frecuencia absoluta acumulada
plt.bar(np.arange(len(Nj)), Nj)
plt.xticks(np.arange(len(Nj)), nombres_f)
plt.show()
```



```
[ ]: # Frecuencia relativa acumulada (%)
plt.bar(np.arange(len(Hj)), [x*100 for x in Hj])
plt.xticks(np.arange(len(Hj)), nombres_f)
plt.show()
```





### *Variables cuantitativas*

**2.10 Ejemplo** Los siguientes datos están asociados con el peso (kg) de un conjunto de materiales. Elaborar la distribución de frecuencias correspondiente.

103.1, 82.1, 106.2, 100.9, 91.8, 96.1, 126.9, 119.8, 93.1, 86.8, 75.2, 93.0, 82.3, 94.8, 64.2, 105.3, 108.0, 86.3, 81.8, 138.1, 92.5, 66.3, 66.6, 142.2, 96.5, 74.8, 95.4, 100.1, 81.9, 112.0, 116.8, 103.2, 66.1, 60.4, 78.7

La variable “peso” es una variable cuantitativa de razón. Es claro que esta variable no está dada en categorías, por lo que es necesario elaborar las clases pertinentes como sigue:

- Se eligen  $m = 6$  clases dado que  $\sqrt{35} = 5.916 \approx 6$  y  $1 + 3.3\log_{10}(35) = 6.095 \approx 6$

-  $x_{min} = 60.4$  y  $x_{max} = 142.2$ .

-  $R = x_{max} - x_{min} = 142.2 - 60.4 = 81.8$ .

-  $a = \frac{81.8}{6} = 13.63$ .

- Las clases resultantes son:

1.  $C_1 = \{x : l_0 \leq x \leq l_1\}$  donde  $l_0 = 60.40$  y  $l_1 = 60.40 + 13.63 = 74.03$ .
2.  $C_2 = \{x : l_1 \leq x \leq l_2\}$  donde  $l_1 = 74.03$  y  $l_2 = 74.03 + 13.63 = 87.67$ .
3.  $C_3 = \{x : l_2 \leq x \leq l_3\}$  donde  $l_2 = 87.67$  y  $l_3 = 87.67 + 13.63 = 101.30$ .
4.  $C_4 = \{x : l_3 \leq x \leq l_4\}$  donde  $l_3 = 101.30$  y  $l_4 = 101.30 + 13.63 = 114.93$ .
5.  $C_5 = \{x : l_4 \leq x \leq l_5\}$  donde  $l_4 = 114.93$  y  $l_5 = 114.93 + 13.63 = 128.57$ .
6.  $C_6 = \{x : l_5 \leq x \leq l_6\}$  donde  $l_5 = 128.57$  y  $l_6 = 128.57 + 13.63 = 142.20$ .

Así, la distribución de frecuencias está dada por:

Clase	F. Absoluta	F. Relativa	F. Abs. Acumulada	F. Rel. Acumulada
[60.40; 74.03]	5	14.3%	5	14.3%
(74.03; 87.67]	9	25.7%	14	40.0%
(87.67; 101.30]	10	28.6%	24	68.6%
(101.30; 114.93]	6	17.1%	30	85.7%
(114.93; 128.57]	3	8.6%	33	94.3%
(128.57; 142.20]	2	5.7%	35	100%
Total	35	100%	N.A.	N.A.

```
[ ]: # Datos
peso = [103.1, 82.1, 106.2, 100.9, 91.8, 96.1, 126.9, 119.8, 93.1, 86.8, 75.
↪2, 93.0,
      82.3, 94.8, 64.2, 105.3, 108.0, 86.3, 81.8, 138.1, 92.5, 66.3, 66.
↪6, 142.2,
      96.5, 74.8, 95.4, 100.1, 81.9, 112.0, 116.8, 103.2, 66.1, 60.4, 78.
↪7]
# Tamaño de la muestra
n = len(peso)
print(n)
```

35

```
[ ]: # Número de intervalos
m = round(1+3.33*np.log10(n))
print(m)
```

6

```
[ ]: # Rango
R = max(peso) - min(peso)
print(round(R,1))
```

81.8

```
[ ]: # Amplitud
a = R/m
print(round(a,5))
```

13.63333

```
[ ]: # Límites
lim = list(map(lambda x: round((x*a)+min(peso),5), list(range(m+1))))
print(lim)
```

[60.4, 74.03333, 87.66667, 101.3, 114.93333, 128.56667, 142.2]

```
[ ]: # Frecuencias absolutas
nj = pd.cut(sorted(peso), lim, include_lowest=True).value_counts()
print(nj)
```

```
(60.399, 74.033]      5
(74.033, 87.667]      9
(87.667, 101.3]     10
(101.3, 114.933]      6
(114.933, 128.567]    3
(128.567, 142.2]      2
dtype: int64
```

```
[ ]: # Frecuencias relativas
hj = nj.div(n)
print(hj)
```

```
(60.399, 74.033]      0.142857
(74.033, 87.667]      0.257143
(87.667, 101.3]       0.285714
(101.3, 114.933]      0.171429
(114.933, 128.567]    0.085714
(128.567, 142.2]      0.057143
dtype: float64
```

```
[ ]: # Frecuencias absolutas acumuladas
Nj = nj.cumsum()
print(Nj)
```

```
(60.399, 74.033]      5
(74.033, 87.667]     14
(87.667, 101.3]     24
(101.3, 114.933]     30
(114.933, 128.567]   33
(128.567, 142.2]     35
dtype: int64
```

```
[ ]: # Frecuencias relativas acumuladas
Hj = hj.cumsum()
print(Hj)
```

```
(60.399, 74.033]      0.142857
(74.033, 87.667]      0.400000
(87.667, 101.3]       0.685714
(101.3, 114.933]      0.857143
(114.933, 128.567]    0.942857
(128.567, 142.2]      1.000000
dtype: float64
```

```
[ ]: # Tabla
tab = pd.concat([nj, hj, Nj, Hj], axis=1)
print(tab)
```

```
(60.399, 74.033]      0      1      2      3
                    5  0.142857  5  0.142857
```

```
(74.033, 87.667]      9  0.257143  14  0.400000
(87.667, 101.3]      10 0.285714  24  0.685714
(101.3, 114.933]     6  0.171429  30  0.857143
(114.933, 128.567]   3  0.085714  33  0.942857
(128.567, 142.2]     2  0.057143  35  1.000000
```

```
[ ]: # Edicion
# nombres_f <- ["Bachillerato", "Pregrado", "Maestria", "Doctorado"]
nombres_c = ["F. Absoluta", "F. Relativa", "F. Abs. Acumulada", "F. Rel.
↪Acumulada"]
tab.columns = nombres_c
# Imprimir tabla
print(tab.to_string())
```

	F. Absoluta	F. Relativa	F. Abs. Acumulada	F. Rel.
Acumulada				
(60.399, 74.033]	5	0.142857	5	0.142857
(74.033, 87.667]	9	0.257143	14	0.400000
(87.667, 101.3]	10	0.285714	24	0.685714
(101.3, 114.933]	6	0.171429	30	0.857143
(114.933, 128.567]	3	0.085714	33	0.942857
(128.567, 142.2]	2	0.057143	35	1.000000

```
[ ]: # Imprimir tabla a markdown (In R using ktnir)
print(tab.round(3).to_markdown())
```

	F. Absoluta	F. Relativa	F. Abs. Acumulada	F. Rel. Acumulada
(60.399, 74.033]	5	0.143	5	0.143
(74.033, 87.667]	9	0.257	14	0.4
(87.667, 101.3]	10	0.286	24	0.686
(101.3, 114.933]	6	0.171	30	0.857
(114.933, 128.567]	3	0.086	33	0.943
(128.567, 142.2]	2	0.057	35	1