

1	# Fuente de datos
2	

In [1]:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 datos = pd.read_excel('C:/Users/Villamar/Desktop/JOSS/SEXTOSEMESTRE/MODELAM1
6 datos
7
8
```

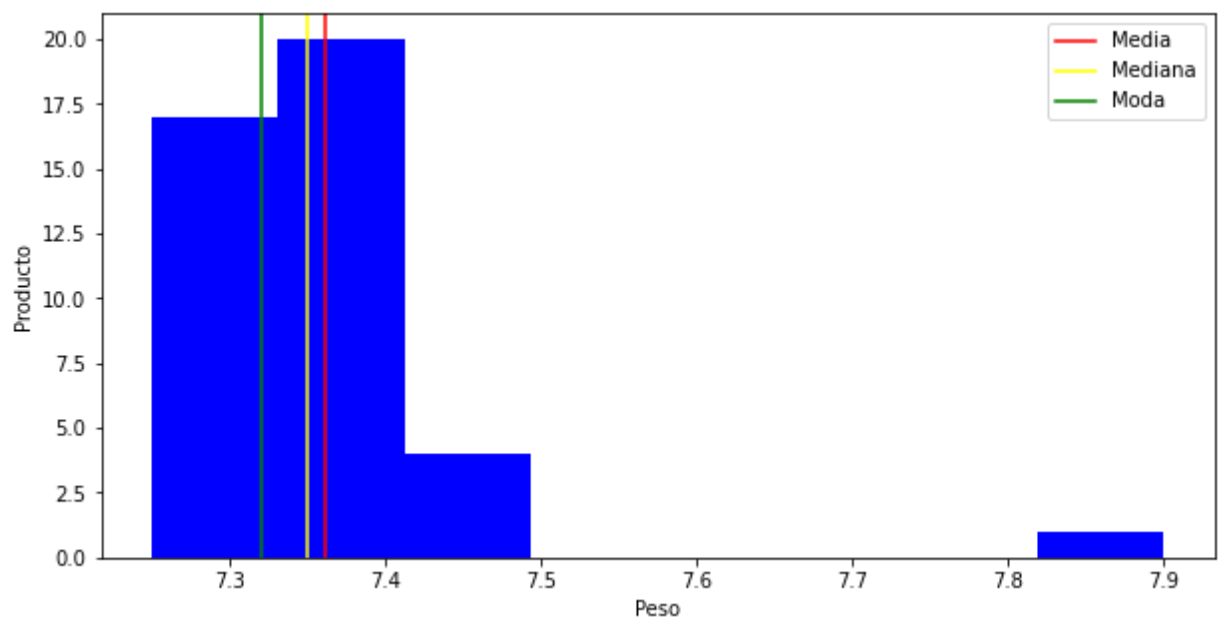
Out[1]:

	articulos	peso_libra
0	1	7.25
1	2	7.26
2	3	7.27
3	4	7.28
4	5	7.28
5	6	7.29
6	7	7.30
7	8	7.30
8	9	7.31
9	10	7.31
10	11	7.32
11	12	7.32
12	13	7.32
13	14	7.32
14	15	7.33
15	16	7.33
16	17	7.33
17	18	7.34
18	19	7.34
19	20	7.35
20	21	7.35
21	22	7.35
22	23	7.35
23	24	7.36
24	25	7.36
25	26	7.36
26	27	7.36
27	28	7.37
28	29	7.37
29	30	7.37

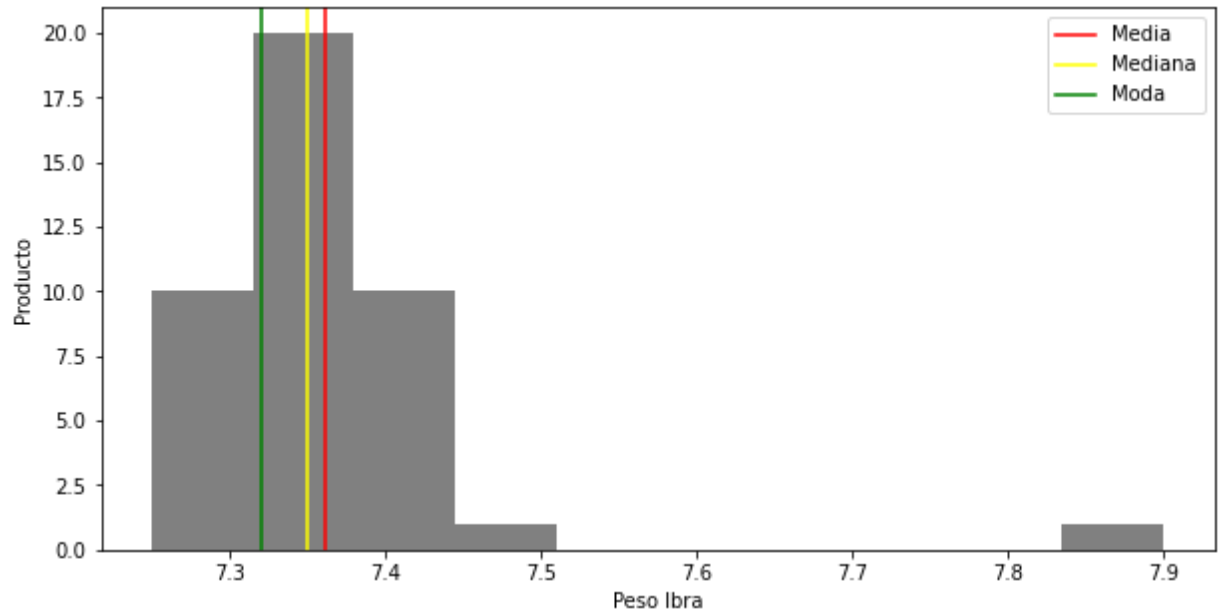
	articulos	peso_libra
30	31	7.38
31	32	7.39
32	33	7.39
33	34	7.40
34	35	7.40
35	36	7.41
36	37	7.41
37	38	7.42
38	39	7.43
39	40	7.44
40	41	7.48
41	42	7.90

Histograma

```
In [2]: 1 x=datos["peso_libra"]
2 plt.figure(figsize=(10,5))
3 plt.hist(x,bins=8,color='blue')
4 plt.axvline(x.mean(),color='red',label='Media')
5 plt.axvline(x.median(),color='yellow',label='Mediana')
6 plt.axvline(x.mode()[0],color='green',label='Moda')
7 plt.xlabel('Peso')
8 plt.ylabel('Producto')
9 plt.legend()
10 plt.show()
```



```
In [3]: 1 x=datos["peso_libra"]
2 plt.figure(figsize=(10,5))
3 plt.hist(x,bins=None,color='grey')
4 plt.axvline(x.mean(),color='red',label='Media')
5 plt.axvline(x.median(),color='yellow',label='Mediana')
6 plt.axvline(x.mode()[0],color='green',label='Moda')
7 plt.xlabel('Peso libra')
8 plt.ylabel('Producto')
9 plt.legend()
10 plt.show()
```



Mediana

```
In [4]: 1 # enviando las medias a t1, t2, t3 para su utilización
2 print("Mediana:", )
3 t1 = datos.median()
4 print( "la Mediana de pesos es: ", t1)
5 print("DIRECTAMENTE DEL DATAFRAME ")
6 datos.median()
```

```
Mediana:
la Mediana de pesos es:  artículos      21.50
peso_libra      7.35
dtype: float64
DIRECTAMENTE DEL DATAFRAME
```

```
Out[4]: artículos      21.50
peso_libra      7.35
dtype: float64
```

Media

```
In [5]: 1 # enviando las medias a t1, t2, t3 para su utilización
        2 print("Media:", )
        3 t1 = datos.mean()
        4 print( "la Media de pesos: ", t1)
        5 print("DIRECTAMENTE DEL DATAFRAME ")
        6 datos.mean()
```

```
Media:
la Media de pesos:  articulos      21.500000
peso_libra      7.361905
dtype: float64
DIRECTAMENTE DEL DATAFRAME
```

```
Out[5]: articulos      21.500000
        peso_libra      7.361905
        dtype: float64
```

Moda

```
In [6]: 1 # enviando las modas a mo1, mo2, mo3 para su utilización
        2 print("Moda:")
        3 mo1 = datos["peso_libra"].mode()
        4 print( "la Moda de peso de articulos: ", mo1)
        5 pd.DataFrame(mo1)
        6
```

```
Moda:
la Moda de peso de articulos:  0    7.32
1    7.35
2    7.36
dtype: float64
```

```
Out[6]:
```

	0
0	7.32
1	7.35
2	7.36

```
In [7]: 1 # Tomamos Los datos de Las columnas
2 datos[['peso_libra']].describe()
3 # describe(), nos presenta directamente la media, desviación standar, el val
4 #o, valor máximo, el 1er cuartil, 2do Cuartil, 3er Cuartil
```

Out[7]:

	peso_libra
count	42.000000
mean	7.361905
std	0.099001
min	7.250000
25%	7.320000
50%	7.350000
75%	7.387500
max	7.900000

```
In [8]: 1 # seleccionamos los datos del mes de abril 30 registros para calcular estadi
2 df_1 = datos[:30]
3 print ("Estadísticos de 30 REGISTROS")
4 print ("-----")
5 df_1[['peso_libra']].describe()
```

Estadísticos de 30 REGISTROS

Out[8]:

	peso_libra
count	30.000000
mean	7.325000
std	0.034215
min	7.250000
25%	7.302500
50%	7.330000
75%	7.350000
max	7.370000

```
In [9]: 1 df_2 = datos[10:21]
        2 df_2
```

Out[9]:

	articulos	peso_libra
10	11	7.32
11	12	7.32
12	13	7.32
13	14	7.32
14	15	7.33
15	16	7.33
16	17	7.33
17	18	7.34
18	19	7.34
19	20	7.35
20	21	7.35

```
In [10]: 1 print ("Estadisticos del mes de Mayo ")
        2 print ("-----")
        3 df_2[['peso_libra']].describe()
```

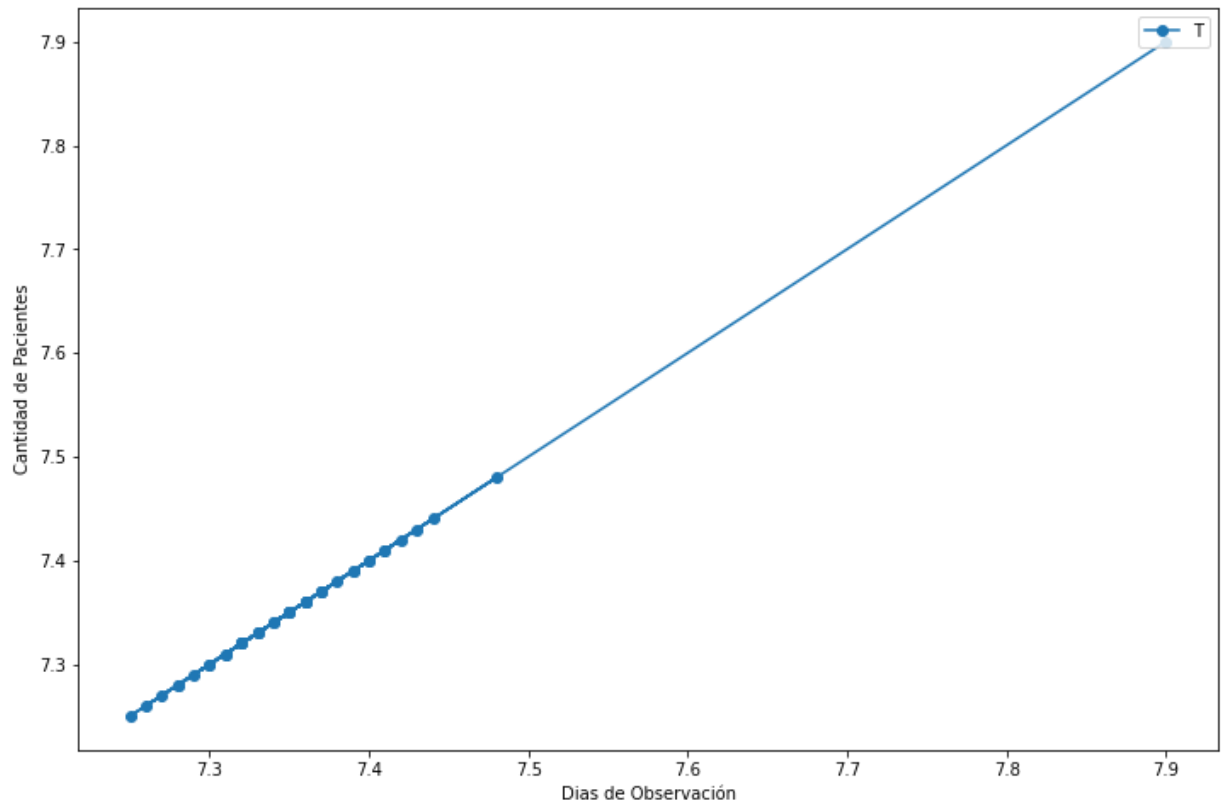
Estadisticos del mes de Mayo

Out[10]:

	peso_libra
count	11.000000
mean	7.331818
std	0.011677
min	7.320000
25%	7.320000
50%	7.330000
75%	7.340000
max	7.350000

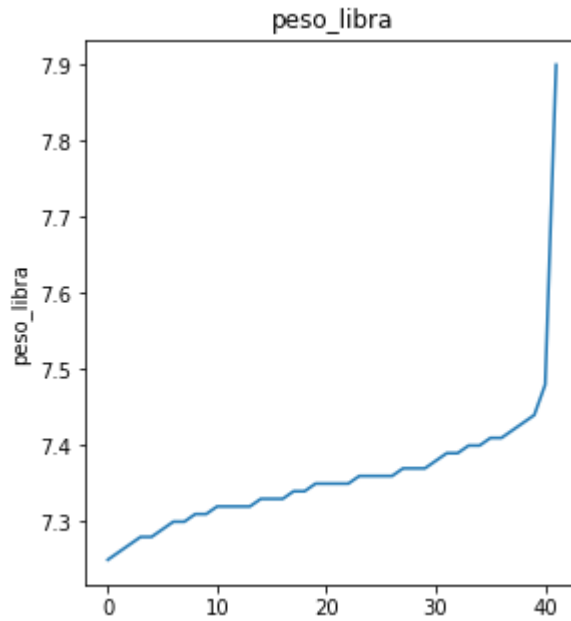
```
In [17]: 1 #x = np.arange(61)
2 x = datos["peso_libra"]
3 t1 = datos["peso_libra"]
4 plt.figure(figsize=(12,8))
5 plt.plot(x,t1,marker='o')
6 plt.xlabel('Dias de Observación')
7 plt.ylabel('Cantidad de pesos')
8 plt.legend(('TOTAL PESOS'), prop = {'size':10},loc='upper right')
```

Out[17]: <matplotlib.legend.Legend at 0x23724e67d30>




```
In [11]: 1 x = range(42)
2 plt.figure(figsize=(15,5))
3 plt.subplot(131)
4 t1 = datos["peso_libra"]
5 p1, = plt.plot(x,t1)
6 plt.ylabel('peso_libra')
7 plt.title('peso_libra')
8
```

Out[11]: Text(0.5, 1.0, 'peso_libra')



```
In [12]: 1 # OBTENER LOS DATOS UNICOS DE LA TABLA
2 lis = datos["peso_libra"].unique()
3 lis
4 dfclases=pd.DataFrame(lis,columns=["peso_libra"])
5 dfclases
```

Out[12]:

	peso_libra
0	7.25
1	7.26
2	7.27
3	7.28
4	7.29
5	7.30
6	7.31
7	7.32
8	7.33
9	7.34
10	7.35
11	7.36
12	7.37
13	7.38
14	7.39
15	7.40
16	7.41
17	7.42
18	7.43
19	7.44
20	7.48
21	7.90

Frecuencias Absolutas

```

In [13]: 1 #TABLA DE FRECUENCIAS ABSOLUTAS
2 # OBTENER FRECUENCIAS ABSOLUTAS DE CADA CLASE
3 datafi=pd.crosstab(index=datos["peso_libra"], columns = "Fi")
4 # Creamos una lista con los valores de las frecuencias
5 li = datafi.values
6 # agregamos una columna al dataframe
7 dfclases["Fi"] = li
8 #observamos dfclase
9 dfclases

```

Out[13]:

	peso_libra	Fi
0	7.25	1
1	7.26	1
2	7.27	1
3	7.28	2
4	7.29	1
5	7.30	2
6	7.31	2
7	7.32	4
8	7.33	3
9	7.34	2
10	7.35	4
11	7.36	4
12	7.37	3
13	7.38	1
14	7.39	2
15	7.40	2
16	7.41	2
17	7.42	1
18	7.43	1
19	7.44	1
20	7.48	1
21	7.90	1

Frecuencias Relativas

```
In [14]: 1 # Columna de Frecuencia relativa
2 total = dfclases.sum(axis=0)
3 datahi = dfclases["Fi"]/total["Fi"] # aqui calculamos la frecuencia
4 datahi.values
5 # agregamos nueva columna de frecuencia relativa
6 dfclases["hi"] = datahi
7 dfclases
```

Out[14]:

	peso_libra	Fi	hi
0	7.25	1	0.023810
1	7.26	1	0.023810
2	7.27	1	0.023810
3	7.28	2	0.047619
4	7.29	1	0.023810
5	7.30	2	0.047619
6	7.31	2	0.047619
7	7.32	4	0.095238
8	7.33	3	0.071429
9	7.34	2	0.047619
10	7.35	4	0.095238
11	7.36	4	0.095238
12	7.37	3	0.071429
13	7.38	1	0.023810
14	7.39	2	0.047619
15	7.40	2	0.047619
16	7.41	2	0.047619
17	7.42	1	0.023810
18	7.43	1	0.023810
19	7.44	1	0.023810
20	7.48	1	0.023810
21	7.90	1	0.023810

```
In [15]: 1 total1 = dfclases.sum(axis=0) # totales
2 total1
```

```
Out[15]: peso_libra    162.28
Fi                42.00
hi                 1.00
dtype: float64
```

```

In [16]: 1 # La suma de las frecuencias Relativas nos da 1
          2 # aqui vamos a calcular la frecuencia absoluta
          3 FA = dfclases["Fi"].values
          4 # obtenemos FA
          5 a=[]
          6 b=0
          7 for c in FA:
          8     b = c + b
          9     a.append(b)
         10 dfclases["FA"] = a
         11 HI = dfclases["hi"].values
         12 # obtenemos HI
         13 a=[]
         14 b=0
         15 for c in HI:
         16     b = c + b
         17     a.append(b)
         18 dfclases["HI"] = a
         19 dfclases

```

Out[16]:

	peso_libra	Fi	hi	FA	HI
0	7.25	1	0.023810	1	0.023810
1	7.26	1	0.023810	2	0.047619
2	7.27	1	0.023810	3	0.071429
3	7.28	2	0.047619	5	0.119048
4	7.29	1	0.023810	6	0.142857
5	7.30	2	0.047619	8	0.190476
6	7.31	2	0.047619	10	0.238095
7	7.32	4	0.095238	14	0.333333
8	7.33	3	0.071429	17	0.404762
9	7.34	2	0.047619	19	0.452381
10	7.35	4	0.095238	23	0.547619
11	7.36	4	0.095238	27	0.642857
12	7.37	3	0.071429	30	0.714286
13	7.38	1	0.023810	31	0.738095
14	7.39	2	0.047619	33	0.785714
15	7.40	2	0.047619	35	0.833333
16	7.41	2	0.047619	37	0.880952
17	7.42	1	0.023810	38	0.904762
18	7.43	1	0.023810	39	0.928571
19	7.44	1	0.023810	40	0.952381
20	7.48	1	0.023810	41	0.976190
21	7.90	1	0.023810	42	1.000000

f) Usando los resultados del problema determinar cual es la probabilidad de

que un peso de un artículo sea 7.30?

La probabilidad es 0.190476

g) Usando los resultados del problema determinar cuál es la probabilidad de

que un peso de un artículo sea 7.35 o menor?

La probabilidad es 0.547619

h) Usando los resultados del problema determinar cuál es la probabilidad de

que un peso de un artículo sea 7.36 o menor?

1	La probabilidad es 0.642857
---	-----------------------------

Generar 10 números aleatorios por el método congruencial multiplicativo

determinar lo valores en peso en libras de cada uno.

$$X_{n+1} = (1140671485X_n + C) \bmod(M) \quad X_n=81, C=12820163; M= 264$$

In [22]:

```

1  # -  $X_{n+1} = (1140671485X_n + C) \bmod(M)$   $X_n=81$ ,  $C=12820163$ ;  $M= 264$ 
2  n, m, a, x0, c = 10, 264, 1140671485, 81, 12820163
3  x = [1] * n
4  r = [0.1] * n
5  print (" Generador Congruencial multiplicativo")
6  print ("-----")
7  for i in range(0, n):
8      x[i] = ((a*x0)+c) % m
9      x0 = x[i]
10     r[i] = x0 / m
11 d = {'Xn': x, 'ri': r }
12 df1 = pd.DataFrame(data=d)
13 # df1.head()
14 df1

```

Generador Congruencial multiplicativo

Out[22]:

	Xn	ri
0	80	0.303030
1	259	0.981061
2	162	0.613636
3	101	0.382576
4	196	0.742424
5	87	0.329545
6	62	0.234848
7	49	0.185606
8	0	0.000000
9	59	0.223485

In []:

1