

Evaluación Primer parcial

Nombre: MACIAS PICO JOSSELYN STEFANY

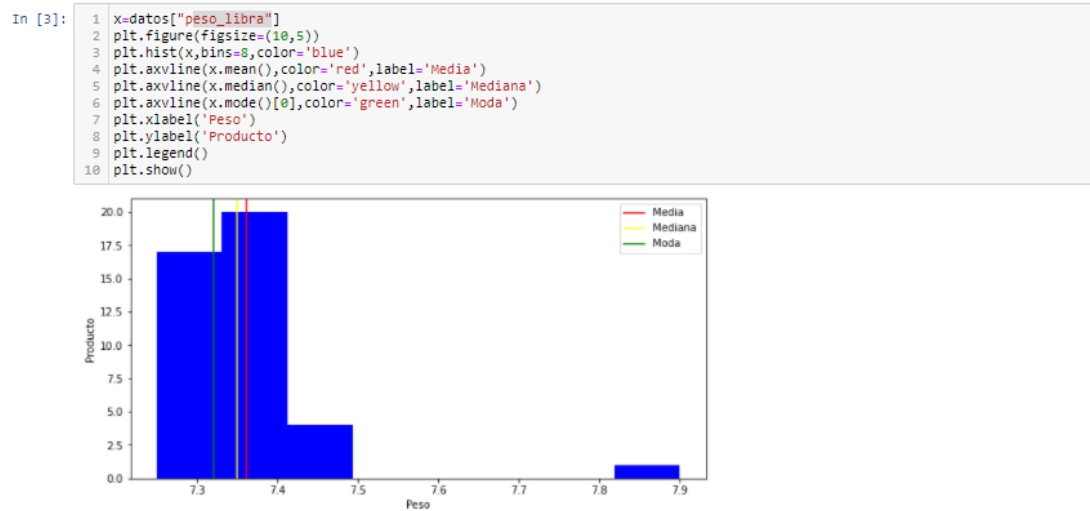
Curso: SEXTO NIVEL "B"

Los datos tabulados muestran el peso en libras de una muestra de 42 artículos manufacturados por una compañía.

7.9	7.31	7.36	7.31	7.35	7.32	7.48
7.37	7.32	7.28	7.36	7.32	7.37	7.33
7.36	7.35	7.34	7.33	7.26	7.33	7.38
7.30	7.25	7.35	7.34	7.40	7.40	7.27
7.32	7.42	7.39	7.41	7.44	7.30	7.36
7.37	7.43	7.28	7.39	7.35	7.29	7.41

a) Dibujar el histograma de datos

Histograma



b) Dibujar la distribución de frecuencia relativa

Frecuencias Relativas

```
In [25]: 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[25]:
```

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

c) Dibujar la distribución de frecuencia acumulada

Frecuencias Absolutas

```
In [24]: 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[24]:
```

	peso_libra	Fi
0	7.90	1
1	7.31	1
2	7.36	1
3	7.35	2
4	7.32	1
5	7.48	2
6	7.37	2
7	7.28	4
8	7.33	3
9	7.34	2
10	7.26	4
11	7.38	4
12	7.30	3
13	7.25	1
14	7.40	2

d) Calcular la media, y la desviación típica

Media

```
In [8]: 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[8]: articulos      21.500000
peso_libra      7.361905
dtype: float64
```

e) Cual es la media, moda y mediana

Media ¶

```
In [8]: 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[8]: articulos      21.500000
peso_libra      7.361905
dtype: float64
```

Moda

```
In [11]: 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[11]:
```

	0
0	7.32
1	7.35
2	7.36

Mediana

```
In [7]: 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:  articulos      21.50
peso_libra      7.35
dtype: float64
DIRECTAMENTE DEL DATAFRAME
```

```
Out[7]: articulos      21.50
peso_libra      7.35
dtype: float64
```

Media

- Usando los resultados del problema determinar cual es la probabilidad de que un peso de un artículo sea 7.30?
- Usando los resultados del problema determinar cuál es la probabilidad de que un peso de un artículo sea 7.35 o menor?
- Usando los resultados del problema determinar cuál es la probabilidad de que un peso de un artículo sea 7.36 o menor?
- Genere 10 números aleatorios por el método congruencial multiplicativo y determine los valores en peso en libras de cada uno.

Genere 10 números aleatorios por el método congruencial multiplicativo

determine los valores en peso en libras de cada uno.

$X_{n+1} = (1140671485X_n + C) \bmod(M)$ $X_n=81$, $C=12820163$; $M= 2^{64}$

```
In [33]: 1 # - Xn+1 = (1140671485Xn+C) mod(M) Xn=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[33]:
```

	Xn	ri
0	80	0.303030
1	259	0.981061
2	162	0.613636
3	101	0.382576
4	198	0.742424
5	87	0.329545

$X_{n+1} = (1140671485X_n + C) \bmod(M)$ $X_n=81$, $C=12820163$; $M= 2^{64}$

	Aleatorio generado	Probabilidad	Peso en libras
1			
2			
3			

4			
5			
6			
7			
8			
9			
10			