

LABORATORIO 7 - INFORME  
RODRIGO URQUIZO 20193339

PREGUNTA 1

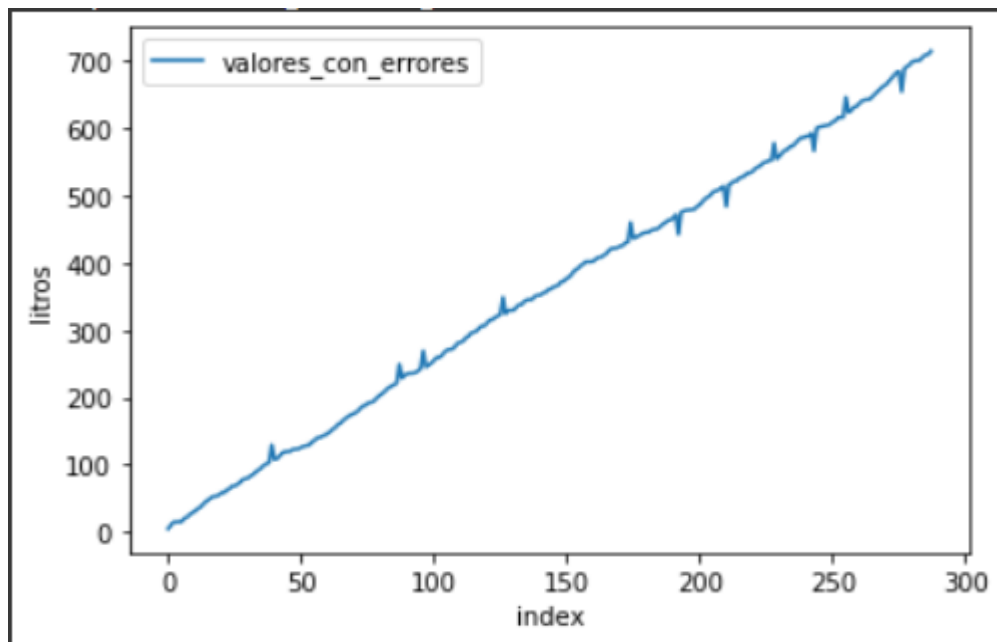
- Esta pregunta se realizó en Google Colab, los datasets de los departamentos están en una carpeta llamada data.

A) Obtengo la cantidad de errores para el dataset 'dpto\_39.csv'

```
45
46 print('Cantidad de errores: ', int(erradas) )
47 #print(erradas)
```

↳ Cantidad de errores: 11

Se comprueba la cantidad de errores con la grafica

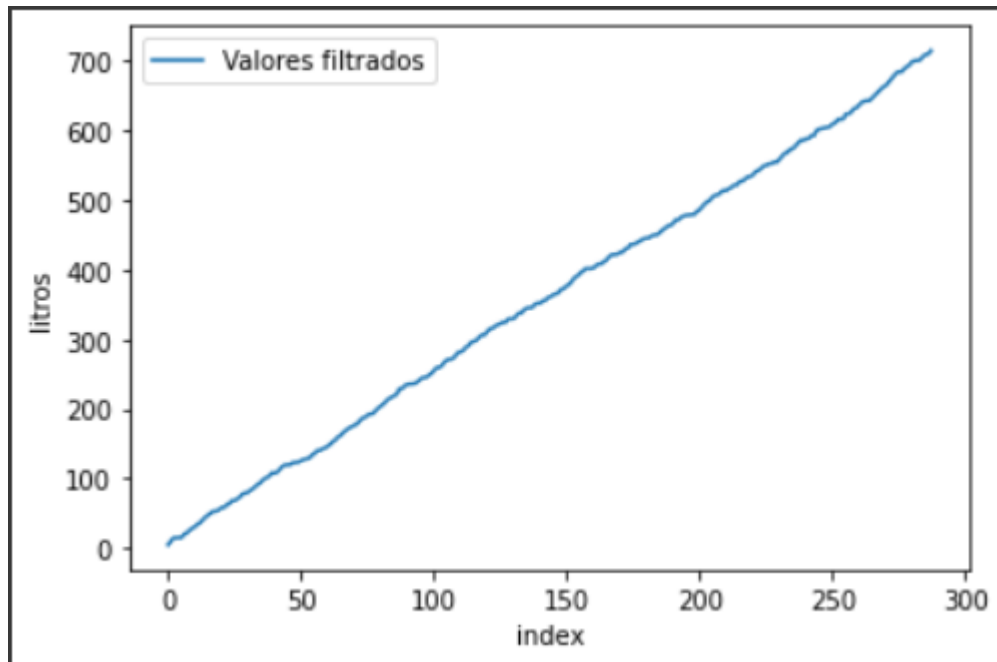


B) Con 9 valores del dataset obtengo la mediana

```
21
22 print('Valores_b:', valores_b)
23 mediana_ = mediana(valores_b)
24 print('Mediana:', mediana_)
25
```

↳ Valores\_b: [4.568183127564, 9.36099061242182, 14.3273829046456, 14.6026146986018, 14.7141574916741, 14.9  
Mediana: 14.7141574916741

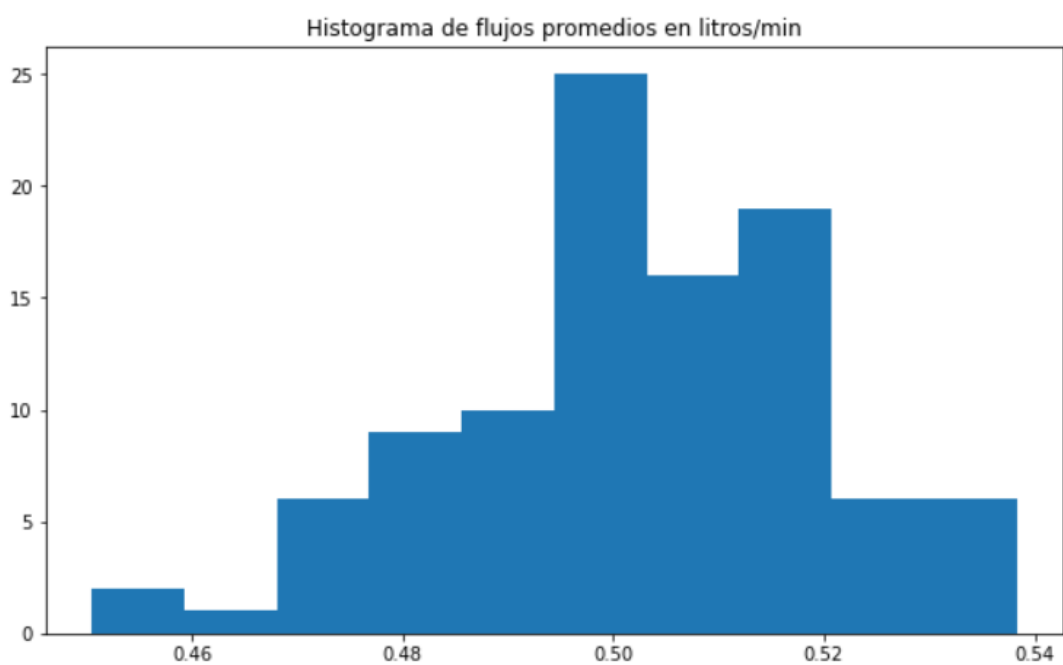
D) Gráfico de los valores corregidos (aplicando el filtro mediana) para el dataset del inciso a empleando la función `filtro_mediana()`



C) y E) Aplico el filtro a todos los datasets empleando la función `filtro_mediana()`, obtengo listas con los valores corregidos (valores filtrados). Con dichas listas obtengo los flujos para cada dataset y hallo el flujo promedio de cada dataset empleando la función `flujo_promedio()`, este valor lo guardo en una lista de flujos promedios. Finalmente con esta lista de promedios obtengo el gráfico del histograma

El flujo lo hallo en litros por minuto, como cada medición se realiza en intervalos de 5 min, entonces el flujo viene a ser el delta de litros (muestra actual - muestra anterior) entre el delta de tiempo (5 min)

Histograma de flujos promedios:



## PREGUNTA 2:

- Esta pregunta se realizó en el archivo pregunta2.py
- Empleo la librería csv para la lectura y escritura

### A) Lectura del archivo

```
tiempo_inicial = time.time()*1000
file = open('data_encoders.txt', 'r')
reader = csv.reader(file)
tiempo_final = time.time()*1000
print(f'El tiempo para leer el archivo es de: {tiempo_final-tiempo_inicial} msec')
print('-----')
```

B) Obtengo la cantidad de reinicios que tiene cada rueda empleando una función en la cual defino que para detectar un reinicio tiene que haber una diferencia mayor a 60 000

```
def cantidad_reinicios(list_encoder):
    reinicios = 0
    anterior = list_encoder[0]
    for i in range(1, len(list_encoder)):
        pulso = list_encoder[i]
        if abs(pulso - anterior) > 60000:
            reinicios += 1
        anterior = pulso
    return reinicios

print(f'El encoder de la rueda 1 ha presentado: {cantidad_reinicios(list_encoder1)} reinicios')
print(f'El encoder de la rueda 2 ha presentado: {cantidad_reinicios(list_encoder2)} reinicios')
print(f'El encoder de la rueda 3 ha presentado: {cantidad_reinicios(list_encoder3)} reinicios')
print(f'El encoder de la rueda 4 ha presentado: {cantidad_reinicios(list_encoder4)} reinicios')
print(f'El encoder de la rueda 5 ha presentado: {cantidad_reinicios(list_encoder5)} reinicios')
print(f'El encoder de la rueda 6 ha presentado: {cantidad_reinicios(list_encoder6)} reinicios')
```

### C) Escritura de datos en el nuevo archivo

```
new_file = open("PulsesEncodersRobot6Wheels.csv", "w", newline='')
file = open('data_encoders.txt', 'r')
reader = csv.reader(file)
tiempo_inicial = time.time()*1000
for row in reader:
    writer = csv.writer(new_file)
    writer.writerow(row)
tiempo_final = time.time()*1000
print(f'El tiempo para escribir el archivo es de: {tiempo_final-tiempo_inicial} msec')
```

Impresión de resultados en la terminal

```
El tiempo para leer el archivo es de: 0.8387999534606934 msec
-----
El encoder de la rueda 1 ha presentado: 216 reinicios
El encoder de la rueda 2 ha presentado: 204 reinicios
El encoder de la rueda 3 ha presentado: 208 reinicios
El encoder de la rueda 4 ha presentado: 206 reinicios
El encoder de la rueda 5 ha presentado: 211 reinicios
El encoder de la rueda 6 ha presentado: 206 reinicios
-----
El tiempo para escribir el archivo es de: 141.82879996299744 msec
PS C:\Users\51996\Desktop\LAB7> █
```

#### D)Tiempo de escritura vs tiempo de lectura

Sí se aprecia una diferencia notable entre los tiempos de lectura y escritura, donde el tiempo de escritura es mucho mayor al tiempo de lectura. Esto se debe a que la memoria caché acelera el tiempo de acceso al contenido de la memoria principal, especialmente si la data que va a ser leída ya está en la caché, sin embargo la escritura es mucho más compleja que la lectura debido a que se tiene que actualizar la memoria principal así también como la caché. Desde la perspectiva de la RAM el problema ocurre cuando se están realizando bloques grandes de escrituras y vamos más allá de las capacidades del buffer, esto hace que se tenga que liberar al buffer de forma más frecuente lo cual afecta significativamente el tiempo de escritura.

Desde la perspectiva del programador en la lectura solamente se tiene que acceder a los datos del archivo, mientras que en la escritura se está accediendo a los datos del archivo(fila por fila) a través de un bucle for y a la vez se tiene que escribir dichos datos en el nuevo archivo de extensión .csv lo que hace que vaya incrementando el tamaño de este archivo, por ello la escritura es más lenta que la lectura.