```
#/usr/bin/python
#!*-* coding:utf-8 *-*
# Este script es sofware libre. Puede redistribuirlo y/o modificarlo bajo
# los terminos de la licencia pública general de GNU, según es publicada
# por la free software fundation bien la versión 3 de la misma licencia
# o de cualquier versión posterior. (según su elección ).
# Si usted hace alguna modificación en esta aplicación, deberá siempre
# mencionar el autor original de la misma.
# Autor:
# Universidad  Distrital Francisco Jose
# Grupo de fisica e informatica
# Dr Julian Andres Salamanca Bernal
# Diego Alberto Parra Garzón
# Colombia, Bogota D.C.


f = load('datos/dats1.dat'); #------------entrada de datos a través del archivo .dat
X = f(:,1)/100 #-----------datos de distancia en metros
Y = f(:,2)/1000000 #-----------datos de voltaje  en voltios
ff = [X, Y]; # ---------------voltaje como funcion X e Y
U = log(X); # ----------------logaritmo natural de la distancia
V = log(Y); #---------------- logaritmo natural del voltaje
U2 = U .* U; #-------------------- el cuadrado del logarito de la distancia
UV = U .* V; # .....................logaritmo del producto del voltaje y la distancia
z = size(f); #---------------------- tamaño de filas y columnas del archivo .dat
z1 = z(:,1); #---------------------- tamaño de columnas del archivo .dat
sumX = sum(X); # --------------------- sumatoria de todos los datos de la distancia en metros
sumY = sum(Y); # --------------------- sumatoria de todos los datos del voltaje en voltios
sumU = sum(U); # --------------------- sumatoria de todos los datos del logaritmo de la distancia
sumV = sum(V); # --------------------- sumatoria de todos los datos del logaritmo del voltaje
sumU2 = sum(U2); # --------------------- sumatoria de todos los datos del cuadrado del logaritmo de la distancia
sumUV = sum(UV);  # --------------------- sumatoria de todos los datos del producto de el logaritmo del voltaje y la distancia
promX = sumX/z1;  # --------------------- el promedio de la distancia
promY = sumY/z1;  # --------------------- el promedio del voltaje
promU = sumU/z1; # --------------------- promedio del logaritmo de la distancia
promV = sumV/z1; # --------------------- promedio del logaritmo del voltaje
promU2 = sumU2/z1; # --------------------- promedio del logaritmo la distancia
promUV = sumUV/z1; # --------------------- promedio del cociente del logaritmo de la distancia y el voltaje.
#-----------------------------Primer calculo-----------------------------
Suv = promUV - promU*promV;
Su2 = promU2 - promU*promU;
#b = (Suv / Su2) - 0.5
b = (Suv / Su2)
A = (promV - b*promU)
#a = exp(A)/5 + 0.02
a = exp(A)
```

```
47  Yest = a* (X .^ b);
48  Yteo= a * (X .^(-2));
49  error = Y .- Yest;
50  ECM = sum(error .^(2)) /z1
51  save -ascii 'datos/a.dat' a;
52  save -ascii 'datos/F.dat' ff;
53  save -ascii 'datos/b.dat' b;
54  save -ascii 'datos/ECM.dat' ECM;
55  save -ascii 'datos/Yest.dat' Yest;
56  #------------------------------segundo calculo-----------------------------
57  Vteo = 0.000259 * (X .^(-2))
58  q1 = [0.01:0.001:5];
59  Pprueba = (Y .* Y)/125 #---------- Potencia en el diodo
60  Voltaje_entrante = 100*(Y ./ Vteo)
61  Pteori = (Vteo .* Vteo)/125
62  DV = Vteo .- Y;
63  T = DV ./ Vteo;
64  sT = sum(T);
65  Tp = sT / z1;
66  Trasmitancia = Tp
67  Reflectancia = 1 - Tp
68  EEp= (Reflectancia)*(0.000259);
69  densidad_voltaje = EEp
70  Voltaje_aproximado = (EEp)*(X .^(-2));
71  BX = (0.000259)*(q1 .^(-2));
72  BY = EEp * (q1 .^(-2));
73  V_estimado = EEp*(X .^(-2));
74  error1 = Y .- V_estimado;
75  ECM1 = sum(error1 .^(2)/z1)
76  save -ascii 'datos/Yteo.dat' Yteo;
77  save -ascii 'datos/Vteo.dat' Vteo;
78  save -ascii 'datos/Amplitud.dat' EEp;
79  save -ascii 'datos/Transmitancia.dat' Tp;
80  save -ascii 'datos/V_aproximado.dat' Voltaje_aproximado;
81  save -ascii 'datos/q1.dat' q1;
82  save -ascii 'datos/BX.dat' BX;
83  save -ascii 'datos/BY.dat' BY;
84  save -ascii 'datos/ECM1.dat' ECM1;
85
86  #plot (q1, BX, '-', q1, BY, 'o-', X, Voltaje_aproximado, "-" )
87  #pause
88
89
```