

Breve introdução a Python

Jorge Centeno



Remarks

Em Python, comentário começa com "cerquinha" #

Ex:

```
# inicio de programa
```

Geralmente você inicia o Python básico, sem muitos recursos, mas como ele é modular, você pode e deve carregar as bibliotecas que for usar. Existem muitas. Isto é feito usando o comando **import**.

Duas bibliotecas básicas são:

NumPy: pacote que suporta arrays e matrizes multidimensionais, possui uma série de funções matemáticas para trabalhar com estas estruturas.

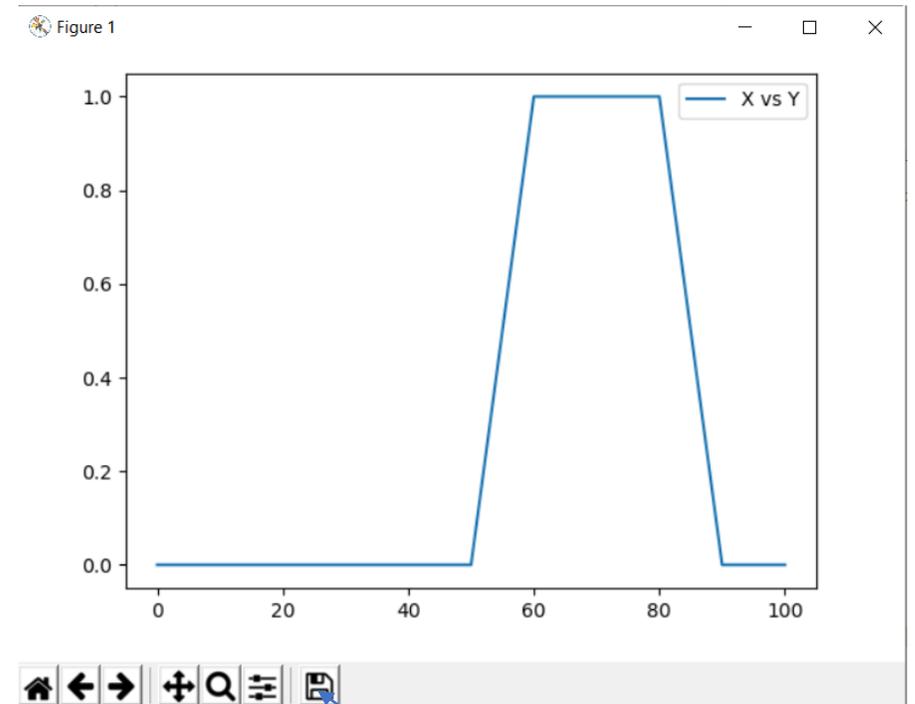
Matplotlib pacote para apresentar gráficos e figuras na tela , junto com sua extensão **pyplot**.

Um exemplo simples grafico x vs y

```
import numpy as np
import matplotlib.pyplot as plt
```

```
# definir dois vetores x e y
x=[0, 50, 60, 80, 90, 100]
y=[0, 0, 1, 1, 0, 0]
```

```
# plotar o gráfico x vs y
plt.plot(x,y)
plt.show()
# ou com legenda...
plt.plot(x,y, label='X vs Y')
plt.legend()
plt.show()
```



Aqui tem opções para “salvar” a imagem, dar zoom, etc

Carregar dados de um “txt”

```
# ler uma tabela com 2 linhas contendo 9 numeros cada uma
Nlinhas=2          # aqui defino o nro de linhas e colunas da tabela
Ncolunas=9
T = np.zeros((Nlinhas, Ncolunas),dtype = float) # criamos tabela vazia
f = open('tabela1.txt', 'r') # abrir arquivo

cl=0               # um contador de linhas
for line in f:     # varrer linhas no arquivo
    cc=0;          # um contador de colunas
    for n in line.split(' '): #varrer elementos da linha lida
        a=float(n)          # converter em float e armazenar em T(cl, cc)
        T[cl,cc]=a
        cc=cc+1
    cl=cl+1
f.close()
```

Agora pode teclar “T” na outra janela e verificar se tem a tabela na memoria



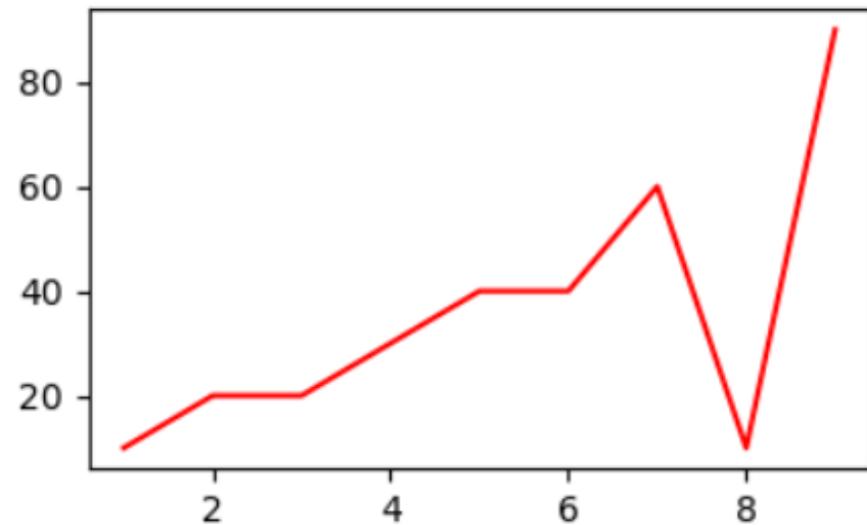
```
# separar X e Y, cada um em uma linha ...
```

```
x=T[0,:]
```

```
y=T[1,:]
```

```
plt.plot(x,y, color='red')
```

```
plt.show()
```

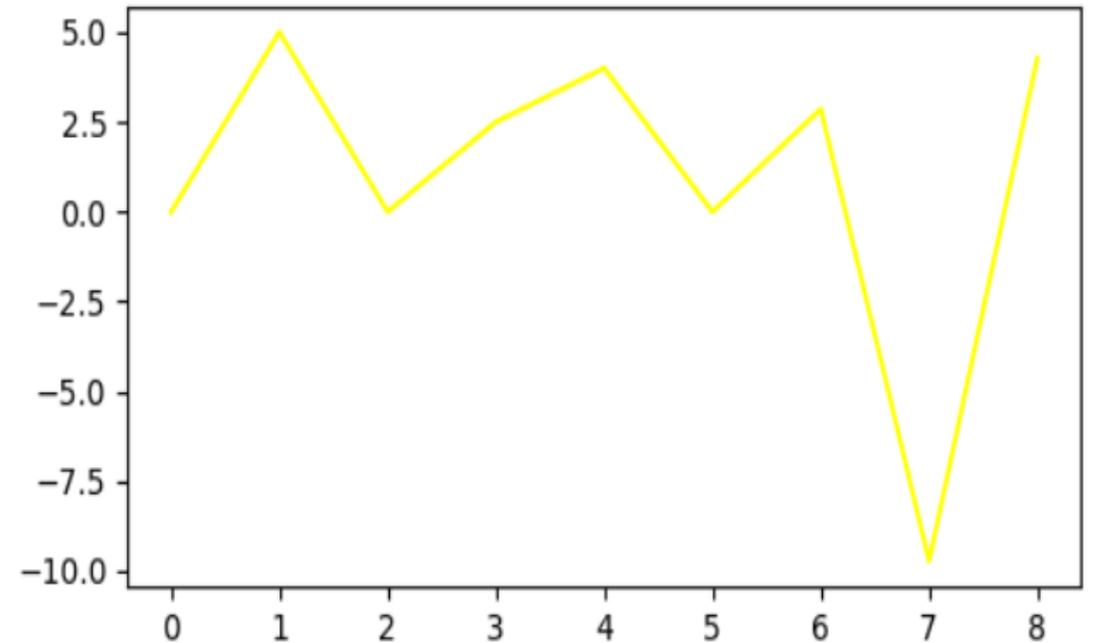


operações

```
# combinar duas variáveis  
R= y-z  
plt.plot(R, color='yellow')  
plt.show()
```

Ou

```
# combinar duas variáveis  
tg= x  
tg[0]=0  
for i in range(1,9):  
    dx=x[i]-x[i-1]  
    dy=y[i]-y[i-1]  
    tg[i]=dy/dx  
plt.plot(tg, color='yellow')  
plt.show()
```



Balanço hídrico

Considere um balanço hídrico simples, onde desejamos calcular o volume da reserva de água no solo:

$$P = ETR + EX + DR$$

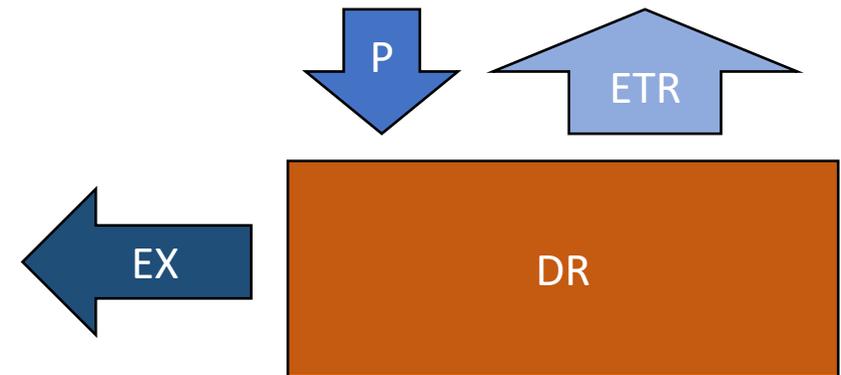
Onde:

P = precipitação em mm;

ETR = evapotranspiração real em mm;

EX = excedente de água (escoamento e infiltração);

DR = variação da reserva de água utilizada pelas plantas.



Balanço hídrico

Considerando que em final de setembro o reservatório estava vazio, calcule a variação da reserva de água ao longo do ano, dado...

mes	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET
evapotranspiração potencial	114	99.2	102	100	87.6	93.5	83	63.9	59	55	75	80
Precipitação	31.7	107.7	272.7	175.4	220.7	138.6	66.1	0.0	0.0	30.0	0.0	0.0
Excedente de água	0	0	13.3	16.5	43.3	17.2	0	0	0	0	0	0

Elabore um programa em Python que leia o arquivo de entrada contendo estes dados em 3 linhas e no final mostre um gráfico com a reserva calculada para cada mês.



```
import numpy as np
import matplotlib.pyplot as plt
# ler uma tabela com 3 linhas contendo 12 dados (mensais)
Nlinhas=3 # aqui defino o nro de linhas e colunas da tabela
Ncolunas=12
T = np.zeros((Nlinhas, Ncolunas),dtype = float) # criação de tabela vazia
f = open('dadosbalanco.txt', 'r') # abrir arquivo
cl=0 # um contador de linhas
for line in f: # varrer linhas no arquivo
    cc=0
    for n in line.split(' '):
        a=float(n) # converter em float e armazenar em T(cl, cc)
        T[cl,cc]=a
        cc=cc+1 # incrementar linha para o seguinte elemento
    cl=cl+1 # incrementar linha para seguinte conjunto de elementos
f.close() # fechar arquivo
```

separar X e Y, cada um em uma linha ...

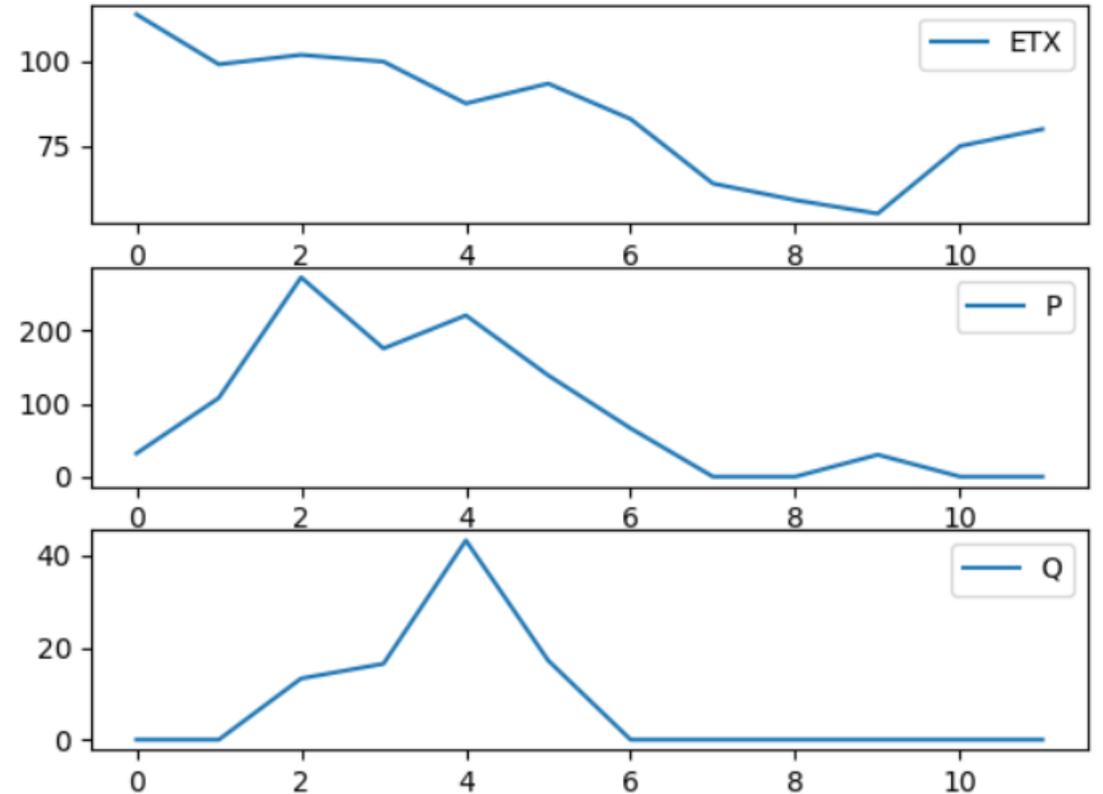
```
ETX=T[0,:]
```

```
P=T[1,:]
```

```
Q=T[2,:]
```

#mostrar as entradas

```
sub1 = plt.subplot(3,1,1)
plt.plot(ETX, label='ETX')
plt.legend()
sub2 = plt.subplot(3,1,2)
plt.plot(P, label='P')
plt.legend()
sub3 = plt.subplot(3,1,3)
plt.plot(Q, label='Q')
plt.legend()
plt.show()
```



```
# Balanço mesmo ...
```

```
DR= [0 for x in range(12)]
```

```
# calcular reserva como  $R(i)=R(i-1) +P -ETX$ 
```

```
meses=12
```

```
for i in range(0,meses):
```

```
    if i==0:
```

```
        DR[i]=P[i]-ETX[i]-Q[i]
```

```
    else:
```

```
        DR[i]=DR[i-1]+P[i]-ETX[i]-Q[i]
```

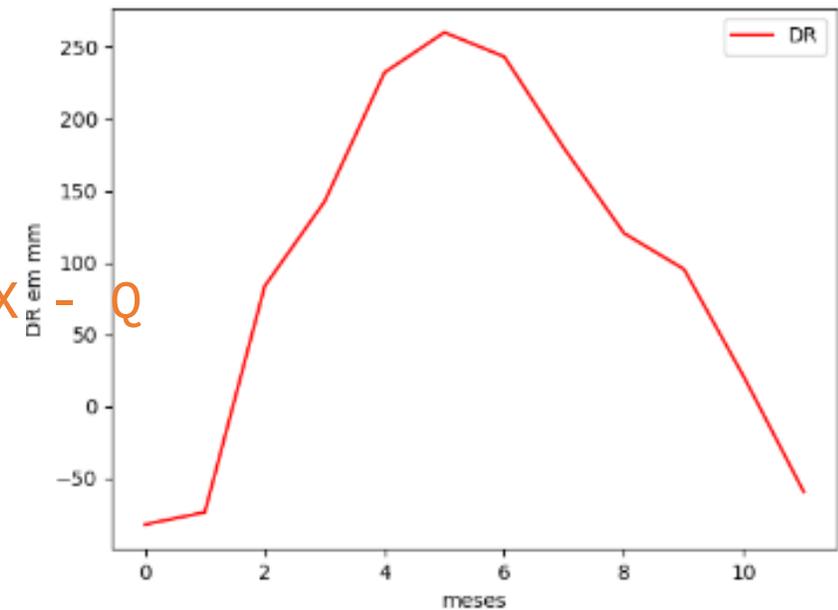
```
plt.plot(DR, label='DR', color='red')
```

```
plt.legend()
```

```
plt.xlabel('meses')
```

```
plt.ylabel('DR em mm')
```

```
plt.show()
```

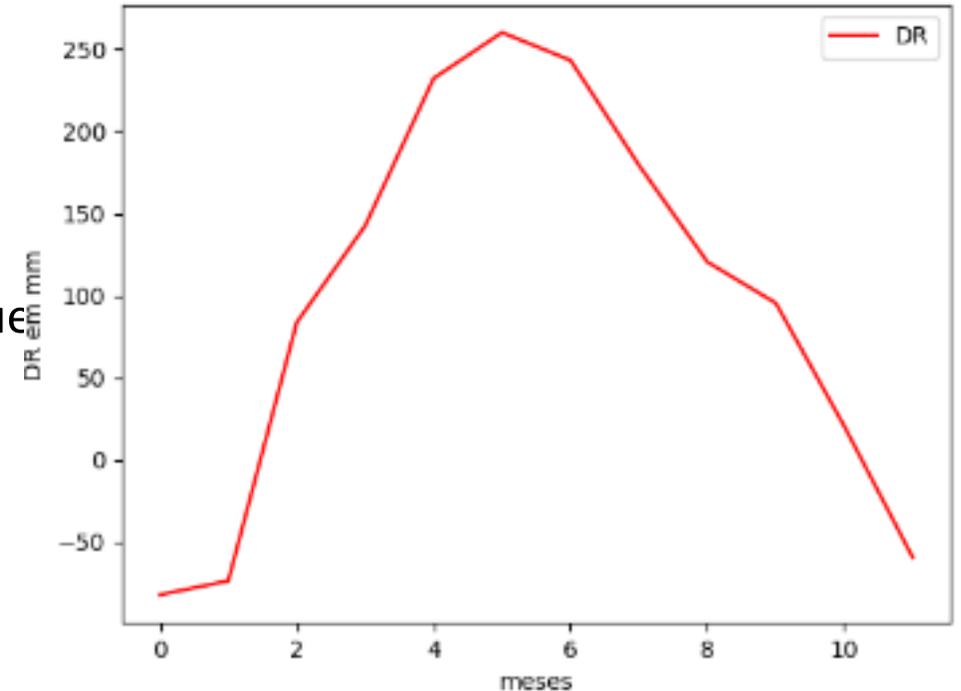


Agora...

Como não pode existir $DR < 0$, por favor modifique o código para que, quando seja calculado um valor negativo de DR, o valor armazenado seja $RD=0$.

Também deve incluir um “alerta” e imprimir uma mensagem quando o armazenamento for maior que 300.

```
print('Armazenamento', DR[i], 'no mês', i)
```





Python

Elabore um programa em Python para solucionar este problema