

MetMatAtuaria_EP1

May 13, 2016

1 Métodos Matemáticos para Atuária: EP 1

1.1 Nome:

Envie este notebook com as solucoes até 5 de junho de 2016 no email rvuspaulas@gmail.com. Nomeie seu notebook seguindo o seguinte modelo: RVicente_EP1. Comente seu código fartamente. Os comentários são parte da avaliação.

1.2 Exercício 1: Cálculo da Taxa Interna de Retorno (TIR) (4 pontos)

A TIR é a taxa x que anula o valor presente líquido de um projeto, ou seja, é a raiz de

$$VPL(x) = \frac{R_1}{(1+x)} + \frac{R_2}{(1+x)^2} + \cdots + \frac{R_n}{(1+x)^n} - \left[I_0 + \frac{I_1}{(1+x)} + \frac{I_2}{(1+x)^2} + \cdots + \frac{I_n}{(1+x)^n} \right]$$

,
onde

$$R_1, R_2, \dots, R_n$$

denota o fluxo de caixa previsto e

$$I_0, I_2, \dots, I_n$$

são investimentos previstos.

Use a função `scipy.optimize.newton` para escrever uma função que calcule a TIR dados os investimentos e o fluxos de caixa previstos.

Escreva uma função que, dados os investimentos e o fluxos de caixa previstos, desenhe um gráfico do VPL. Confira a TIR obtida na função anterior visualmente.

Comente seus códigos.

1.3 Exercício 2: Interpolação de Taxas de Juro (6 pontos)

O conjunto de dados abaixo contem uma lista de prazos e uma lista de taxas de juro ao ano em %

```
In [8]: data = [[20,40,60,80,100,140,180,220,300,350],  
               [18.75,19.15,19.40,19.75,20.0,20.4,20.6,20.58,20.4,20.3]]
```

O seguinte código representa os mesmos dados na forma de uma tabela

```
In [12]: from IPython.display import HTML, display
```

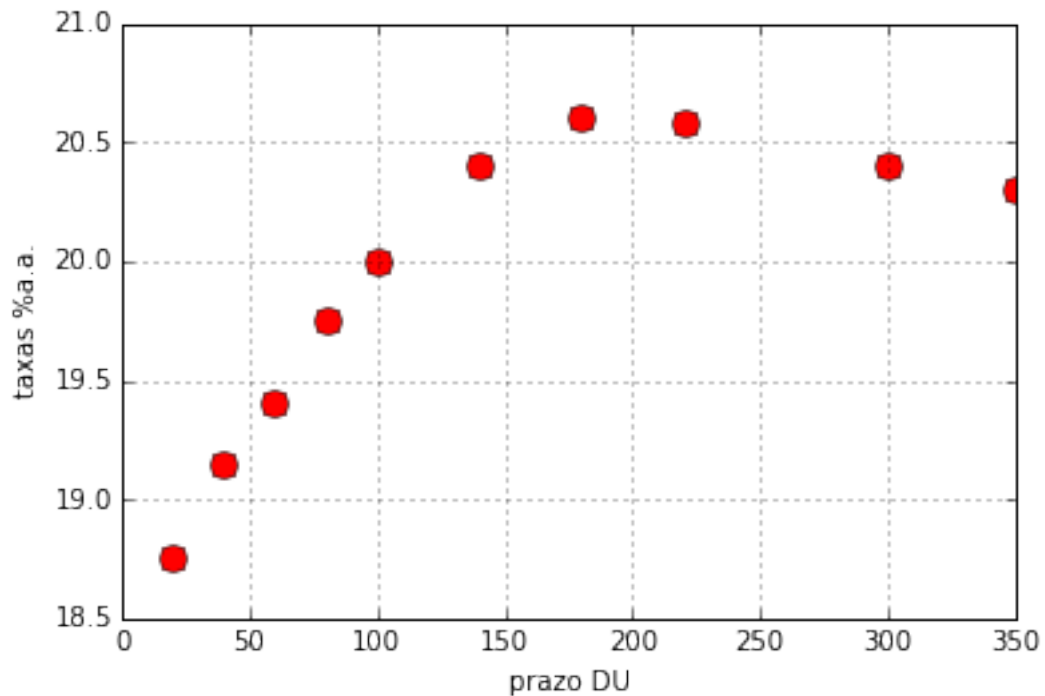
```
display(HTML(  
    '<table><tr>{</tr></table>'.  
    format(  
        '</tr><tr>'.join('<td>{</td>'.format('</td><td>'.join(str(_) for _ in row)) for row in  
    )  
))
```

<IPython.core.display.HTML object>

A seguir fazemos um gráfico com as taxas

```
In [17]: import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [35]: plt.plot(data[0],data[1], 'or',markersize=10)
         plt.grid()
         plt.ylabel('taxas %a.a.')
         plt.xlabel('prazo DU')
         plt.show()
```



Use `scipy.interpolate` para interpolar localmente uma spline cúbica e globalmente um polinômio. Se preferir escreva seu próprio código de interpolação.

Escreva uma função que forneça o menor prazo para uma taxa dada.

```
In [ ]:
```