financial

March 8, 2021

1 Usando a biblioteca Financial do Python

Doherty Andrade – www.metodosnumericos.com.br

Vamos ilustrar com exemplos como usar as principais funções da biblioteca Financial.

1. pmt – Calcula o pagamento mensal de um empréstimo.
2. fv – Calcula o valor futuro de um investimento.
3. pv – Calcula o valor presente de um investimento.
4. nper – Calcula o número de pagamentos (períodos).
5. rate – Calcula a taxa de juros por período de uma anuidade.
6. ppmt –Calcula o pagamento sobre o principal para um investimento em um período específico.
8. effect – Calcula a taxa de juros anual efetiva.
9. irr – Calcula a taxa de juros interna do retorno para uma série de fluxo de caixa.
10. npv – Calcula o valor presente líquido (net present value).

A sintaxe é semelhante em todas as funções. Como exemplo, vitamos a função fv:

numpy.fv(rate, nper, pmt, pv, when='end'),

onde when : {{'begin', 1}, {'end', 0}}, {string, int}, optional

When ou quando, se os pagamentos devidos são ('begin' (1) ou 'end' (0)). O defaults é {'end', 0}.

Primeiro vamos importar as bibliotecas Python necessárias. Você encontra funções financeiras na Biblioteca Python Numpy.

```
In [1]: import numpy as np
        import pandas as pd
```

1.0.1 Instalando o financial

```
In [2]: pip install numpy-financial
```

```
Requirement already satisfied: numpy-financial in c:\users\doherty\anaconda3\lib\site-packages
Requirement already satisfied: numpy>=1.15 in c:\users\doherty\anaconda3\lib\site-packages (fr
Note: you may need to restart the kernel to use updated packages.
```

```
In [3]: print(np.__version__)
```

```
1.16.2
```

```
In [4]: import numpy_financial as npf
```

1.1 Exemplos

1.1.1 Função pmt

Podemos usar a função pmt() para calcular os pagamentos mensais. Primeiramente vamos criar um dicionário e chamá-lo de Obs.

Observe o sinal negativo que indica pagamento.

Exemplo 1

Qual é a retirada mensal necessária para usufruir de um capital de $ 10.000 em 5 anos a uma taxa de juros anual de 6% ao ano?

```
In [5]: df = pd.DataFrame({'Obs':[1]})
        df['rate'] = 0.06/12
        df['nper'] = 60
        df['pv'] = 10000
        df['fv'] = 0

        df['pmt'] = np.pmt(df.rate, df.nper, df.pv, df.fv)

        df

Out[5]:    Obs   rate  nper     pv  fv         pmt
        0    1  0.005    60  10000   0 -193.328015
```

Exemplo 2

Qual é a retirada mensal necessária para usufruir de um capital de $ 200.000 em 15 anos a uma taxa de juros anual de 7.5% ao ano?

```
In [6]: df = pd.DataFrame({'Obs':[1]})
        df['rate'] = 0.075/12
        df['nper'] = 15*12
        df['pv'] = 200000
        df['fv'] = 0


        df['pmt'] = np.pmt(df.rate, df.nper, df.pv, df.fv)
        df

Out[6]:    Obs     rate  nper      pv  fv         pmt
        0    1  0.00625   180  200000   0 -1854.02472
```

1.1.2 Função fv

Exemplo 3

Vamos usar a função np.fv(rate, nper, pmt, pv). Mesma sintaxe anterior.

Neste exemplo vamos calcular o valor de um empréstimo de 10.000 se você pagar 200 por mês durante os próximos 60 meses. A taxa de juros anual é de 6% ao ano e é calculada mensalmente.

```
In [7]: df = pd.DataFrame({'Obs':[1]})
        df['rate'] = 0.06/12
```

```
df['nper'] = 60
df['pmt'] = -200
df['pv'] = 10000
df['fv'] = np.fv(df.rate, df.nper, df.pmt, df.pv)
df
```

Out[7]:

	Obs	rate	nper	pmt	pv	fv
0	1	0.005	60	-200	10000	465.504576

1.1.3 Função pv

Usando a função valor presente np.pv() do NumPy:

Exemplo 3

Qual é o valor presente (por exemplo, o investimento inicial) de um investimento que precisa totalizar \$ 15692,93 após 10 anos economizando $ 100 por mês? Suponha que a taxa de juros seja de 5% (ao ano) composta mensalmente.

Sintaxe para usar diretamente: numpy.pv(rate, nper, pmt, fv=0.0, when='end')

```
In [8]: pv = np . pv ( 0.05 / 12 ,  10 * 12 ,  - 100 ,  15692.93 )
        pv
```

Out[8]: -100.00067131625819

```
In [9]: df = pd.DataFrame({'Obs':[1]})
        df['rate'] = 0.05/12
        df['nper'] = 10*12
        df['pmt'] = -100
        df['fv'] = 15692.93
        df['pv'] = np.pv(df.rate, df.nper, df.pmt, df.fv)
        df
```

Out[9]:

	Obs	rate	nper	pmt	fv	pv
0	1	0.004167	120	-100	15692.93	-100.000671

Por convenção, o sinal negativo representa o fluxo de caixa (ou seja, dinheiro não disponível hoje). Assim, para terminar com \$ 15.692,93 em 10 anos economizando $ 100 por mês com juros anuais de 5%, o depósito inicial também deve ser de \$ 100.

Exemplo 4

Qual é o valor presente (por exemplo, o investimento inicial) de um investimento após 5 anos economizando $ 200 por mês? Suponha que a taxa de juros seja de 6% (ao ano) composta mensalmente.

```
In [10]: df = pd.DataFrame({'Obs':[1]})
         df['rate'] = 0.06/12
         df['nper'] = 12*5
         df['pmt'] = -200
         df['fv'] = 0
         df['pv'] = np.pv(df.rate, df.nper, df.pmt, df.fv)
         df
```

Out[10]:

	Obs	rate	nper	pmt	fv	pv
0	1	0.005	60	-200	0	10345.11215

1.1.4 Função nper

Agora vamos usar a função np.nper() em um exemplo.

Exemplo 5

Se você fosse sacar \$ 150 ao mês de um capital de \$8.000 com juro sde 7% ao ano, quanto tempo (meses) levaria?

```
In [11]: df = pd.DataFrame({'Obs':[1]})
         df['rate'] = 0.07/12
         df['pmt'] = -150
         df['pv'] = 8000
         df['fv'] = 0

         df['nper'] = np.nper(df.rate, df.pmt, df.pv, df.fv)
         df
```

```
Out[11]:    Obs      rate  pmt    pv  fv       nper
         0    1  0.005833 -150  8000   0  64.073349
```

1.1.5 Função rate

Agora vamos usar a função np.rate(). Mas para obter o valor anual devemos multiplicar o resultado por 12.

```
In [12]: df = pd.DataFrame({'Obs':[1]})
         df['nper'] = 60
         df['pmt'] = -200
         df['pv'] = 10000
         df['fv'] = 0

         df['rate'] = np.rate(df.nper, df.pmt, df.pv, df.fv) * 12
         df
```

```
Out[12]:    Obs  nper  pmt     pv  fv      rate
         0    1    60 -200  10000   0  0.074201
```

1.1.6 Função ppmt

Usando a função ppmt().

Para criar valores para 60 meses, usamos o range(1,61) para fazer uma lista com 60 valores. Para cada período a função np.ppmt() é usado para calcular o pagamento principal.

```
In [13]: df = pd.DataFrame()
         df['period'] = range(1, 61)
         df['rate'] = 0.06/12
         df['nper'] = 60
         df['pv'] = 10000
         df['fv'] = 0
         df['ppmt'] = np.ppmt(df.rate, df.period, df.nper, df.pv, df.fv)
         df
```

Out[13]:

	period	rate	nper	pv	fv	ppmt
0	1	0.005	60	10000	0	-143.328015
1	2	0.005	60	10000	0	-144.044655
2	3	0.005	60	10000	0	-144.764879
3	4	0.005	60	10000	0	-145.488703
4	5	0.005	60	10000	0	-146.216147
5	6	0.005	60	10000	0	-146.947227
6	7	0.005	60	10000	0	-147.681963
7	8	0.005	60	10000	0	-148.420373
8	9	0.005	60	10000	0	-149.162475
9	10	0.005	60	10000	0	-149.908287
10	11	0.005	60	10000	0	-150.657829
11	12	0.005	60	10000	0	-151.411118
12	13	0.005	60	10000	0	-152.168174
13	14	0.005	60	10000	0	-152.929015
14	15	0.005	60	10000	0	-153.693660
15	16	0.005	60	10000	0	-154.462128
16	17	0.005	60	10000	0	-155.234439
17	18	0.005	60	10000	0	-156.010611
18	19	0.005	60	10000	0	-156.790664
19	20	0.005	60	10000	0	-157.574617
20	21	0.005	60	10000	0	-158.362490
21	22	0.005	60	10000	0	-159.154303
22	23	0.005	60	10000	0	-159.950074
23	24	0.005	60	10000	0	-160.749825
24	25	0.005	60	10000	0	-161.553574
25	26	0.005	60	10000	0	-162.361342
26	27	0.005	60	10000	0	-163.173148
27	28	0.005	60	10000	0	-163.989014
28	29	0.005	60	10000	0	-164.808959
29	30	0.005	60	10000	0	-165.633004
30	31	0.005	60	10000	0	-166.461169
31	32	0.005	60	10000	0	-167.293475
32	33	0.005	60	10000	0	-168.129942
33	34	0.005	60	10000	0	-168.970592
34	35	0.005	60	10000	0	-169.815445
35	36	0.005	60	10000	0	-170.664522
36	37	0.005	60	10000	0	-171.517845
37	38	0.005	60	10000	0	-172.375434
38	39	0.005	60	10000	0	-173.237311
39	40	0.005	60	10000	0	-174.103498
40	41	0.005	60	10000	0	-174.974015
41	42	0.005	60	10000	0	-175.848885
42	43	0.005	60	10000	0	-176.728129
43	44	0.005	60	10000	0	-177.611770
44	45	0.005	60	10000	0	-178.499829
45	46	0.005	60	10000	0	-179.392328
46	47	0.005	60	10000	0	-180.289290

```
47      48  0.005    60  10000   0 -181.190736
48      49  0.005    60  10000   0 -182.096690
49      50  0.005    60  10000   0 -183.007173
50      51  0.005    60  10000   0 -183.922209
51      52  0.005    60  10000   0 -184.841820
52      53  0.005    60  10000   0 -185.766029
53      54  0.005    60  10000   0 -186.694860
54      55  0.005    60  10000   0 -187.628334
55      56  0.005    60  10000   0 -188.566475
56      57  0.005    60  10000   0 -189.509308
57      58  0.005    60  10000   0 -190.456854
58      59  0.005    60  10000   0 -191.409139
59      60  0.005    60  10000   0 -192.366184
```

1.1.7 Função rate

Você mesmo pode calcular a taxa de juros efetiva. Vamos mostrar como fazer isso em Python.

```
In [14]: rate = 0.06
         nper = 12
         effect = (1 + rate/nper)**nper - 1

         df = pd.DataFrame({'Obs':[1]})
         df['rate'] = rate
         df['nper'] = nper
         df['effect'] = (1 + df.rate/df.nper)**df.nper - 1
         df
```

```
Out[14]:    Obs  rate  nper    effect
         0    1  0.06    12  0.061678
```

1.1.8 Função irr

Também temos a função np.irr(), você não precisa ter o fluxo de caixa em variáveis.

```
In [15]: df = pd.DataFrame()
         df['cashflow'] = [-40000, 5000, 8000, 12000, 30000]
         df['irr'] = np.irr(df['cashflow'])
         df
```

```
Out[15]:    cashflow       irr
         0    -40000  0.105823
         1      5000  0.105823
         2      8000  0.105823
         3     12000  0.105823
         4     30000  0.105823
```

1.1.9 Função npv

Exemplo

Calculando o valor presente líquido

```
In [16]: df = pd.DataFrame()
         df['values'] = [-40000, 5000, 8000, 12000, 30000]
         df['rate'] = 0.08
         df['npv'] = np.npv(df['rate'],df['values'])
         df
```

```
Out[16]:    values  rate          npv
         0  -40000  0.08  3065.222668
         1    5000  0.08  3065.222668
         2    8000  0.08  3065.222668
         3   12000  0.08  3065.222668
         4   30000  0.08  3065.222668
```

```
In [ ]:
```