

Programando em Python

Aula 6

Prof. Dr. Marco Antonio Leonel Caetano

Importando Planilhas Excel para *Vetores (array)*

	A	B
1	1	10
2	2	9
3	3	8
4	4	9
5	5	9
6	6	8
7	7	10
8	8	11
9	9	15
10	10	7
11	11	8
12	12	7
13	13	8
14	14	10
15	15	10
16	16	10
17	17	11
18	18	9
19	19	10
20	20	8

```
1 # Retornos - estatística
2 import pandas as pd
3 import statistics as estat
4 import matplotlib.pyplot as fig
5 import numpy as np
6
7 plan=pd.read_excel('dados.xlsx',sheet_name='Planilha1')
```

Importando Planilhas Excel para *Vetores (array)*


	A	B
1	1	10
2	2	9
3	3	8
4	4	9
5	5	9
6	6	8
7	7	10
8	8	11
9	9	15
10	10	7
11	11	8
12	12	7
13	13	8
14	14	10
15	15	10
16	16	10
17	17	11
18	18	9
19	19	10
20	20	8

```
1 # Retornos - estatística
2 import pandas as pd
3 import statistics as estat
4 import matplotlib.pyplot as fig
5 import numpy as np
6
7 plan=pd.read_excel('dados.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 x=np.zeros(lin)
```

Importando Planilhas Excel para *Vetores (array)*

	A	B
1	1	10
2	2	9
3	3	8
4	4	9
5	5	9
6	6	8
7	7	10
8	8	11
9	9	15
10	10	7
11	11	8
12	12	7
13	13	8
14	14	10
15	15	10
16	16	10
17	17	11
18	18	9
19	19	10
20	20	8

```
1 # Retornos - estatística
2 import pandas as pd
3 import statistics as estat
4 import matplotlib.pyplot as fig
5 import numpy as np
6
7 plan=pd.read_excel('dados.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 x=np.zeros(lin)
11
12 for i in range(lin):
13     x[i]=plan.values[i][1]
```

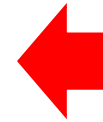


*Lembrar sempre que as colunas começam
Com 0, 1, 2... Logo a coluna B é 1*

Importando Planilhas Excel para *Vetores (array)*

	A	B
1	1	10
2	2	9
3	3	8
4	4	9
5	5	9
6	6	8
7	7	10
8	8	11
9	9	15
10	10	7
11	11	8
12	12	7
13	13	8
14	14	10
15	15	10
16	16	10
17	17	11
18	18	9
19	19	10
20	20	8

```
1 # Retornos - estatística
2 import pandas as pd
3 import statistics as estat
4 import matplotlib.pyplot as fig
5 import numpy as np
6
7 plan=pd.read_excel('dados.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 x=np.zeros(lin)
11
12 for i in range(lin):
13     x[i]=plan.values[i][1]
```



*Dados da planilha 1 do
arquivo em Excel*

Cálculo do retorno para *Vetores (array)* (sem o uso do “For”)

```
retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
```



Começa da 2ª. pos

Cálculo do retorno para *Vetores (array)* (sem o uso do “For”)

```
retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
```



Termina na última

Cálculo do retorno para *Vetores (array)* (sem o uso do “For”)

```
retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
```



Começa na 1ª. pos

Cálculo do retorno para *Vetores (array)* (sem o uso do “For”)

```
retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
```



Termina na penúltima

Cálculo do retorno para *Vetores (array)* (sem o uso do “For”)

```
retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
```



Termina na penúltima

```
array([-0.1      , -0.11111111,  0.125      ,  0.        ,  
       -0.11111111,  
         0.25     ,  0.1       ,  0.36363636, -0.53333333,  
       0.14285714,  
       -0.125    ,  0.14285714,  0.25      ,  0.        ,  0.  
       ,  
         0.1      , -0.18181818,  0.11111111, -0.2       ])
```

Importando Planilhas Excel para *Vetores (array)*

Figura da coluna B do Excel

```
1# Retornos - estatística
2import pandas as pd
3import statistics as estat
4import matplotlib.pyplot as fig
5import numpy as np
6
7plan=pd.read_excel('dados.xlsx',sheet_name='Planilha1')
8
9lin=len(plan)
10x=np.zeros(lin)
11
12for i in range(lin):
13    x[i]=plan.values[i][1]
14
15retorno=(x[1:lin]-x[0:lin-1])/x[0:lin-1]
16
17
18fig.figure()
19fig.plot(retorno)
```

Plota o gráfico

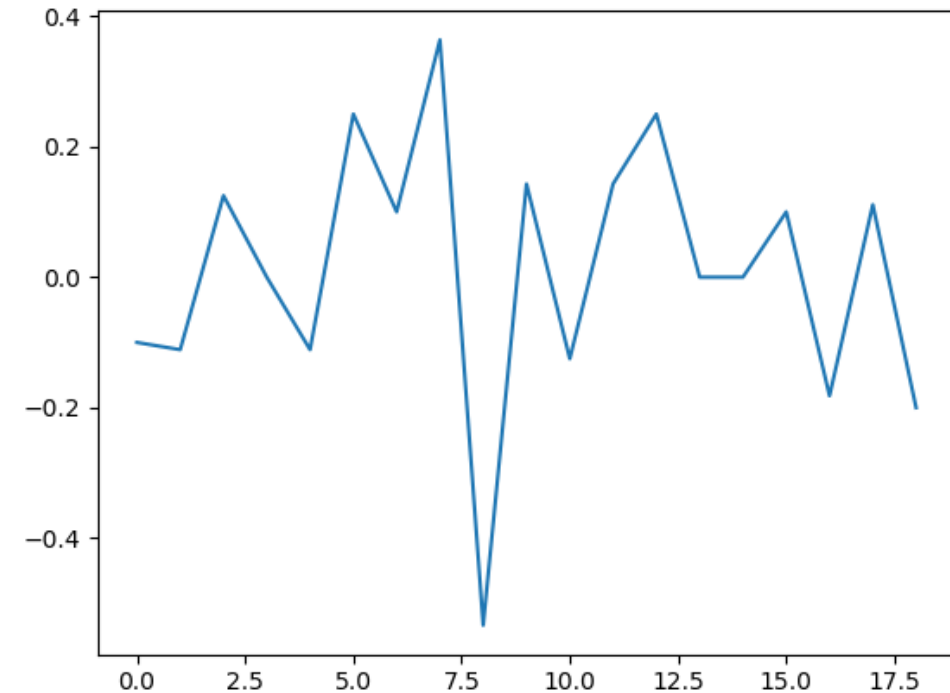
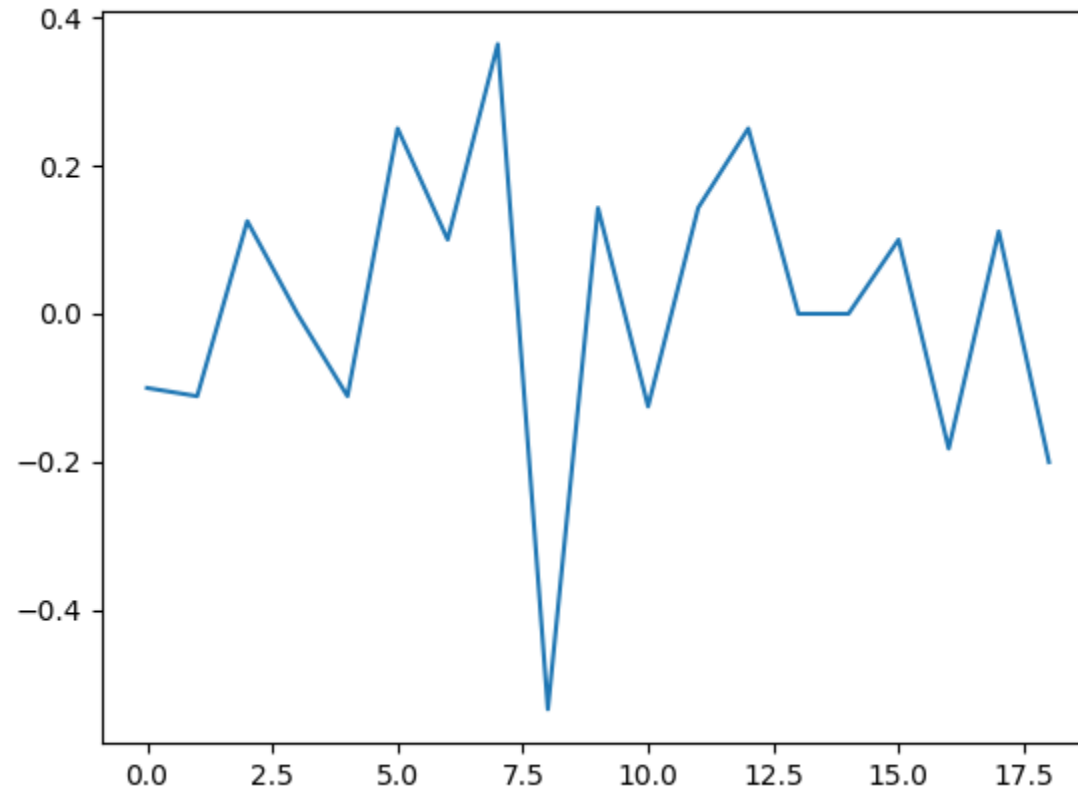


Gráfico do retorno para *Vetores (array)*

```
fig.figure()  
fig.plot(retorno)
```



Estatísticas básicas do retorno

```
--  
30 minimo=retorno.min()  
31 maximo=retorno.max()  
32 media=retorno.mean()  
33 desvio=retorno.std()  
34  
35 print("+++++ ESTATISTICAS +++++")  
36 print("MÁXIMO = ",maximo)  
37 print("MÍNIMO = ",minimo)  
38 print("MÉDIA = ",media)  
39 print("DESVIO PADRAO = ",desvio)  
40
```

```
+++++ ESTATISTICAS +++++  
MÁXIMO =  0.36363636363636365  
MÍNIMO = -0.5333333333333333  
MÉDIA =  0.011741474899369638  
DESVIO PADRAO =  0.19884107404063106
```

Tendência Linear (“polyfit”)

Importando os dados e
iniciando um vetor

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
```

	A	B
1	preços	
2	10	
3	9	
4	8	
5	9	
6	9	
7	8	
8	10	
9	11	
10	15	
11	19	
12	16	
13	14	
14	20	
15	25	
16	30	
17	27	
18	29	
19	26	
20	31	
21	32	

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
```


Criando um vetor

	A	B
1	preços	
2	10	
3	9	
4	8	
5	9	
6	9	
7	8	
8	10	
9	11	
10	15	
11	19	
12	16	
13	14	
14	20	
15	25	
16	30	
17	27	
18	29	
19	26	
20	31	
21	32	

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
```


“linspace” cria o eixo x
começando pelo primeiro
índice até a última linha, com
número total de todas as linhas


linspace(1,lin,lin)

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
```

“linspace” cria o eixo x
começando pelo primeiro
índice até a última linha, com
número total de todas as linhas


linspace(1,lin,lin)

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
```

“**linspace**” cria o eixo x
começando pelo primeiro
índice até a última linha, com
número total de todas as linhas

Total de números



`.linspace(1,lin,lin)`

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
16 coef=np.polyfit(x,y,1)
```

Descobre os coeficientes
“a” e “b” da reta:

$$y = a.x + b$$




Coef[0] é “a”

Coef[1] é “b”

Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
16 coef=np.polyfit(x,y,1)
17
18 tendencia= coef[0]*x + coef[1]
```

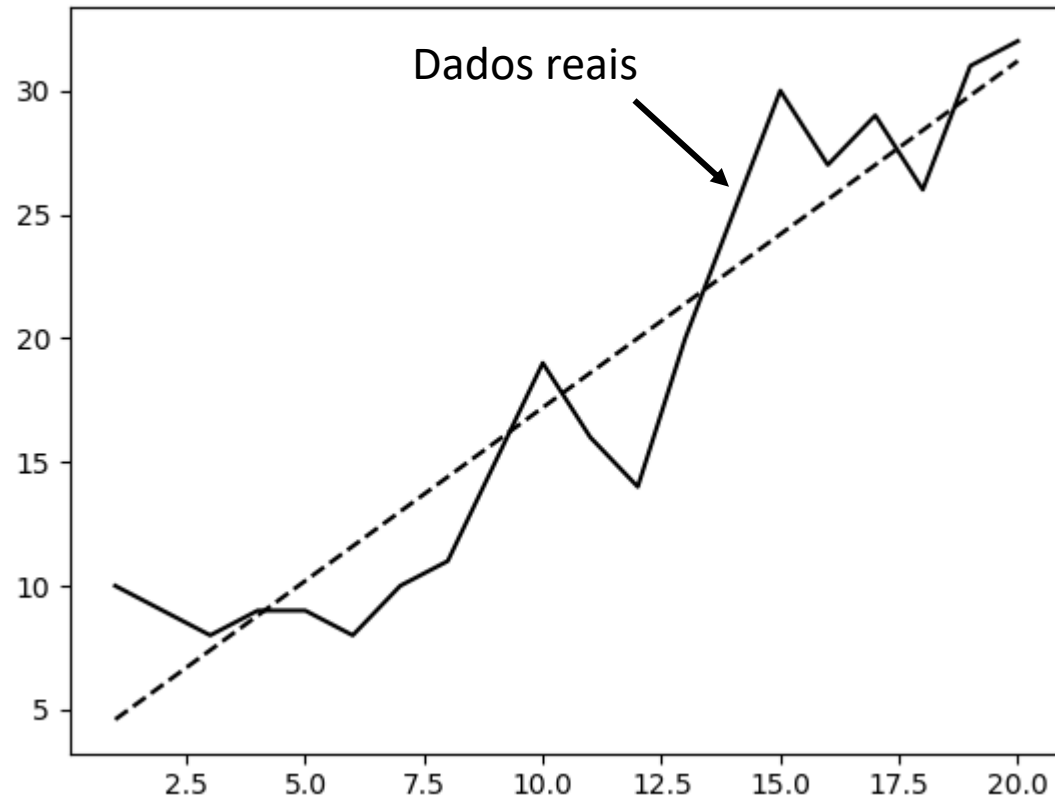


Constrói a reta com os coeficientes (a,b)

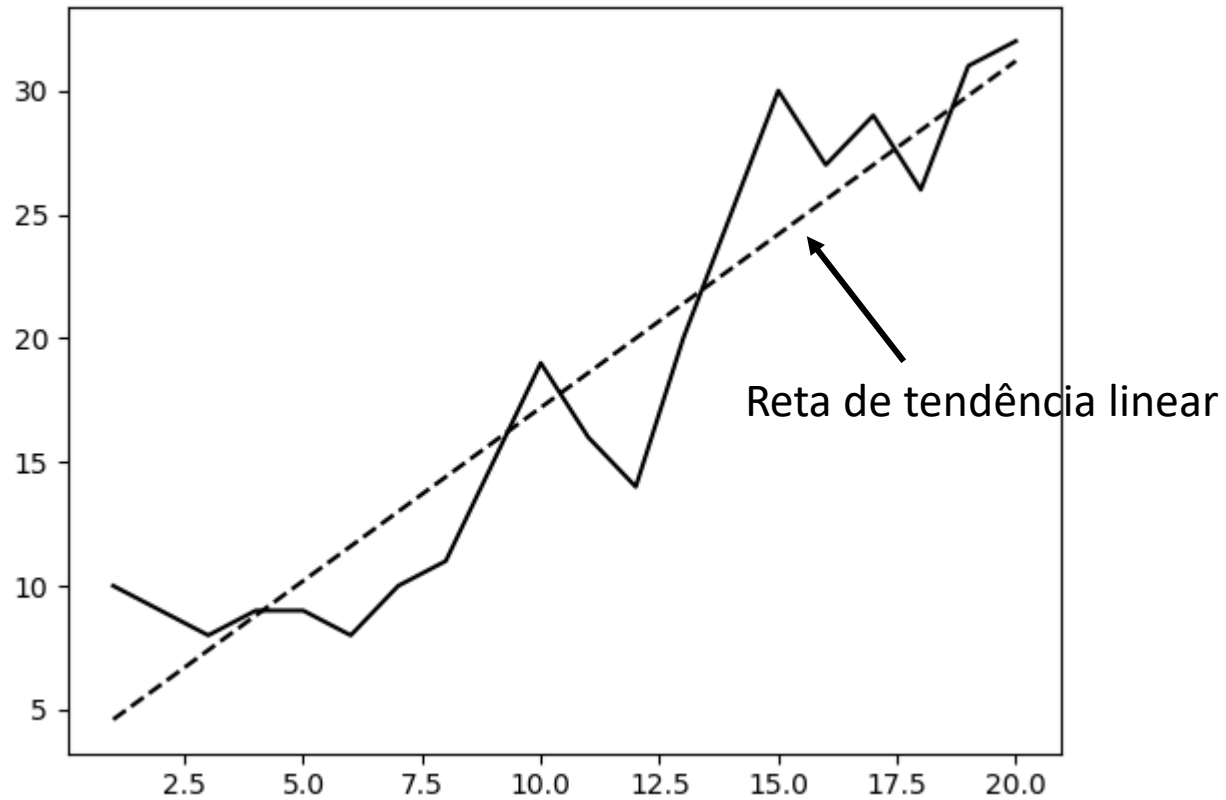
Tendência Linear (“polyfit”)

```
1 # Retornos - estatística
2 import pandas as pd
3 import matplotlib.pyplot as fig
4 import numpy as np
5 import matplotlib.mlab as m
6
7 plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9 lin=len(plan)
10 y=np.zeros(lin)
11
12 for i in range(lin):
13     y[i]=plan.values[i][0]
14
15 x=np.linspace(1,lin,lin)
16 coef=np.polyfit(x,y,1)
17
18 tendencia= coef[0]*x + coef[1]
19
20 print("++++++ coeficientes tendência linear y = b + ax +++++")
21 print("a = ", coef[0],"    b = ",coef[1])
22
23 fig.plot(x,y,'-k',x,tendencia,'--k')
```

Tendência Linear (“polyfit”)

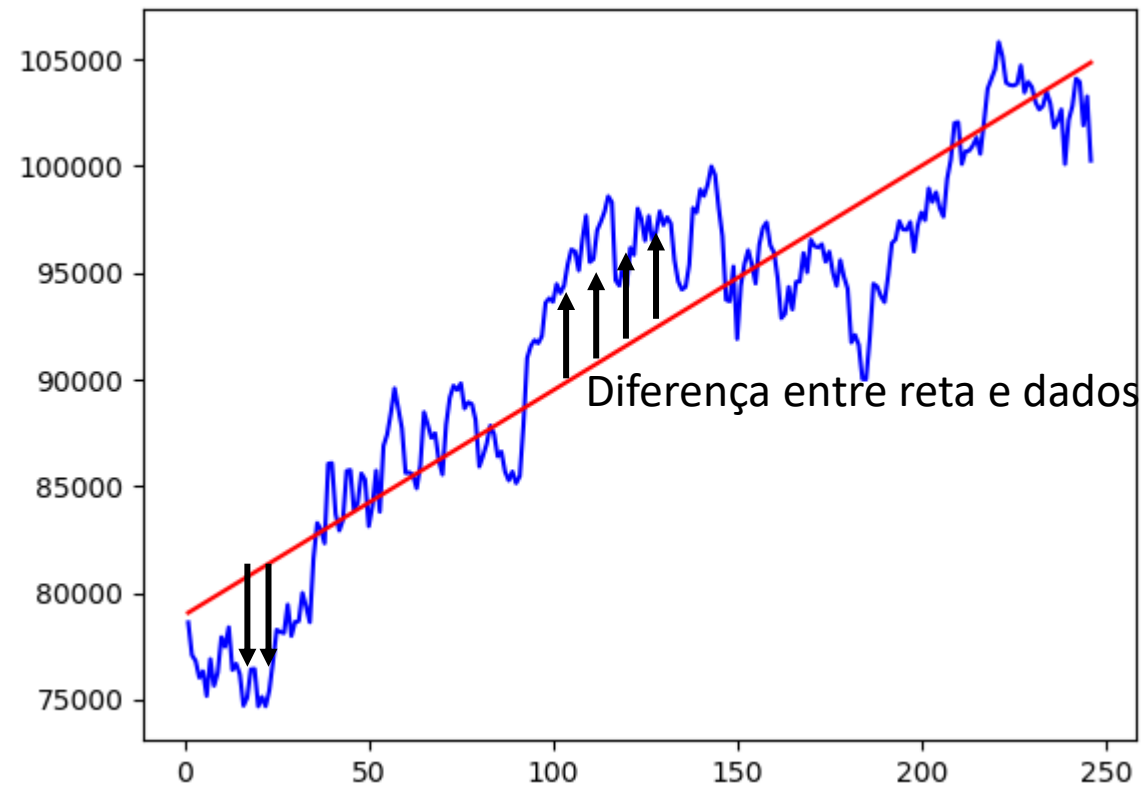


Tendência Linear (“polyfit”)



Oscilação dos dados(“matplotlib.mlab”)

A biblioteca “matplotlib.mlab” já possui uma rotina para fazer a diferença entre a reta de tendência linear e os Dados. A rotina é “**detrend**”.



Oscilação dos dados("matplotlib.mlab")

Passo-1: importar a biblioteca
usando variável "m"



```
1# Retornos - estatística
2import pandas as pd
3import matplotlib.pyplot as fig
4import numpy as np
5import matplotlib.mlab as m
6
7plan=pd.read_excel('Tendencia.xlsx',sheet_name='Planilha1')
8
9lin=len(plan)
10y=np.zeros(lin)
11
12for i in range(lin):
13    y[i]=plan.values[i][0]
14
15x=np.linspace(1,lin,lin)
16coef=np.polyfit(x,y,1)
17
18tendencia= coef[0]*x + coef[1]
19
20print("++++++ coeficientes tendência linear y = b + ax +++++")
21print("a = ", coef[0],"    b = ",coef[1])
22
23fig.plot(x,y,'-k',x,tendencia,'--k')
```

Oscilação dos dados("matplotlib.mlab")

Passo-2: chamar a rotina "detrend_linear"



```
25 filtro = m.detrend_linear(y)
26 fig.figure()
27 fig.plot(x,filtro, '-k')
```

