Listas 3 e 4 - Matplot e Numpy

João Alonso Casella - data: 09/05/2022

In [3]:

```
# Importando bibliotecas
import pandas as pd
import pandas_datareader as reader
import pandas_ta as ta
import random
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy.stats import norm
%matplotlib inline
import time
import datetime
from scipy import stats
from IPython import display
```

Lista Matplot

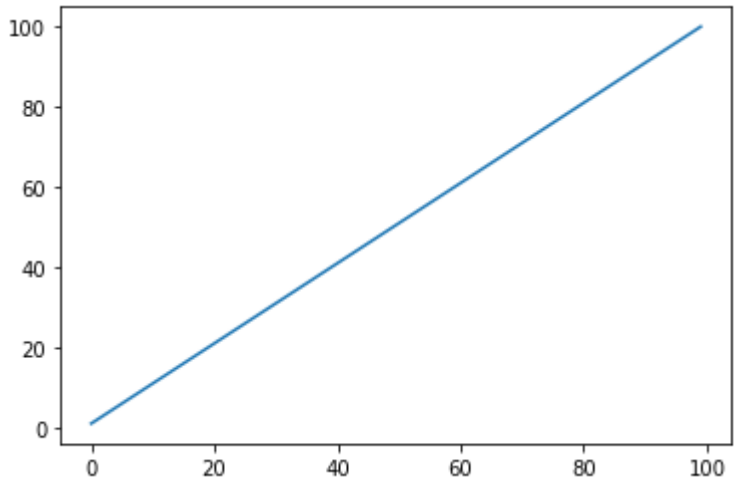
In [2]:

```
#1)
a=[]
for i in range(100):
    a.append(i+1)
    i=i+1
print(a)
plt.plot(a)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 5
8, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76,
77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 9
5, 96, 97, 98, 99, 100]
```

Out[2]:

```
[<matplotlib.lines.Line2D at 0x2d55143c3a0>]
```

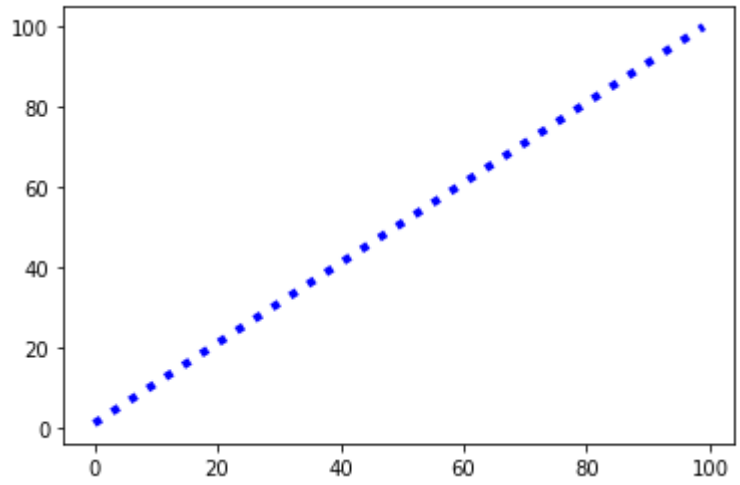


In [3]:

```
#2)
plt.plot(a, color = 'b', linestyle = ':', lw = 4)
```

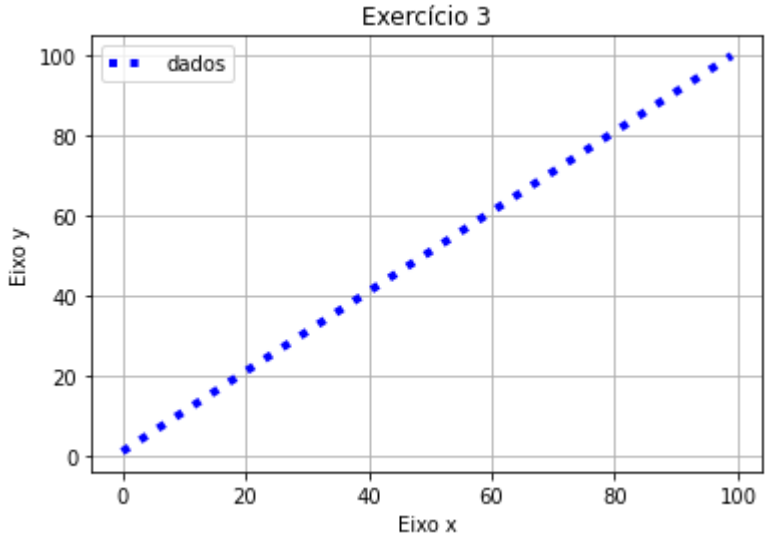
Out[3]:

```
[<matplotlib.lines.Line2D at 0x2d5514e4100>]
```



In [4]:

```
#3)
plt.plot(a, color="b", linestyle = ":", lw = 4, label = 'dados')
plt.legend()
plt.xlabel('Eixo x')
plt.ylabel('Eixo y')
plt.title("Exercício 3")
plt.grid()
```



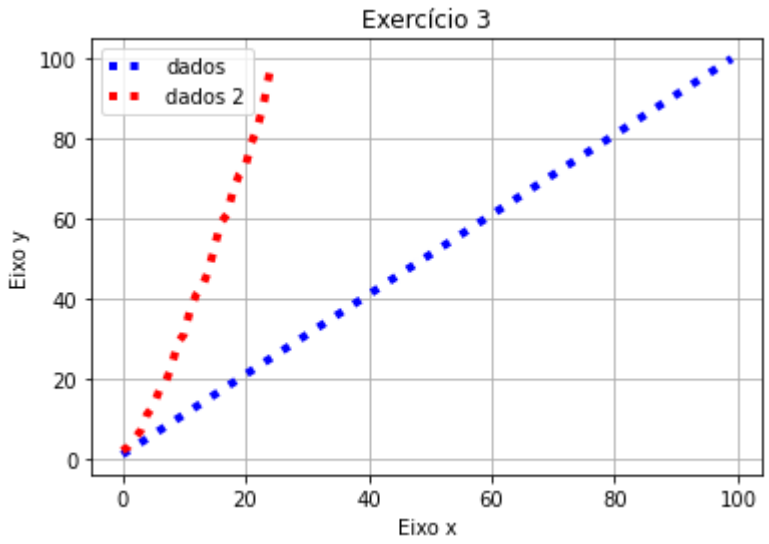
In [5]:

```
#4)
n = int(input("n: "))
b = list(range(2, n))
for i in range(2, int(math.sqrt(n)+1)):
    if i in b:
        for j in range(i**2, n, i):
            if j in b: b.remove(j)

print(b)

plt.plot(a, color="b", linestyle = ":", lw = 4, label = 'dados')
plt.legend()
plt.xlabel('Eixo x')
plt.ylabel('Eixo y')
plt.title("Exercício 3")
plt.grid()
plt.plot(b, color = "r", linestyle = ":", lw = 4, label= 'dados 2')
plt.legend()
```

```
n: 100
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 7
1, 73, 79, 83, 89, 97]
```

Out[5]:

```
<matplotlib.legend.Legend at 0x2d5515b1d60>
```

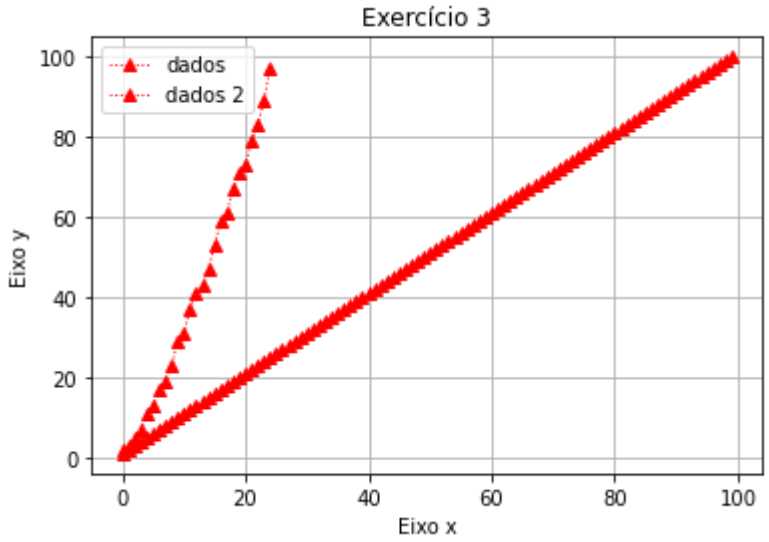


In [6]:

```
#5)
plt.plot(a, color="r", linestyle = ":", lw = 1, label = 'dados', marker = "^")
plt.legend()
plt.xlabel('Eixo x')
plt.ylabel('Eixo y')
plt.title("Exercício 3")
plt.grid()
plt.plot(b, color = "r", linestyle = ":", lw = 1, label= 'dados 2', marker = "^")
plt.legend()
```

Out[6]:

```
<matplotlib.legend.Legend at 0x2d5516193d0>
```

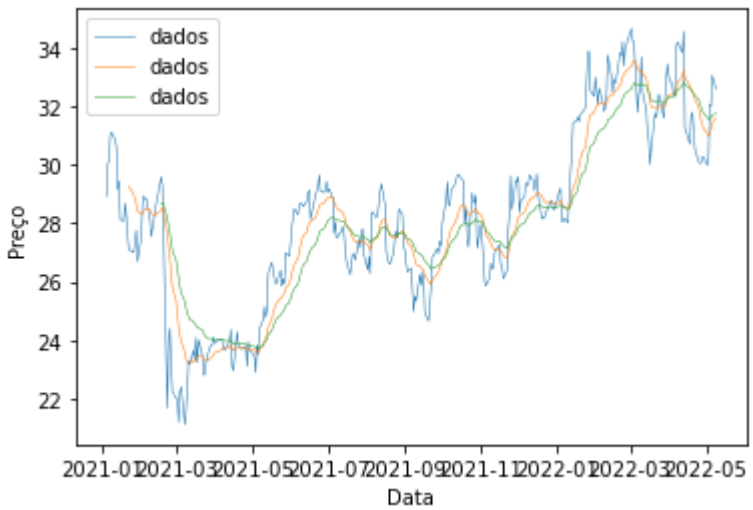


In [18]:

```
#6)
Ticker_trabalhado = 'PETR4.SA'

data = reader.get_data_yahoo(Ticker_trabalhado,start = "2021-01-01", interval="d")
data = data.dropna()  # Remove as linhas em branco
data = data[['Close']]

# Cálculo da média móvel exponencial p/ curto e longo prazo # ------------------------
-

Periodo_curto = 15
Periodo_Longo = 30

data['MMEC'] = round(ta.ema(data.iloc[:, 0], Periodo_curto), 4) # Média móvel exponenci
al de curto
data['MMEL'] = round(ta.ema(data.iloc[:, 0], Periodo_Longo), 4) # Média móvel exponenci
al de longo

# Criando listas para plotar no gráfico
Lista_EMA_curto = list(data['MMEC'])
Lista_EMA_longo = list(data['MMEL'])
Lista_PRECO_fechamento = list(data['Close'])
plt.plot(data, label="dados", lw = 0.5)
plt.legend()
plt.xlabel('Data')
plt.ylabel('Preço')
```

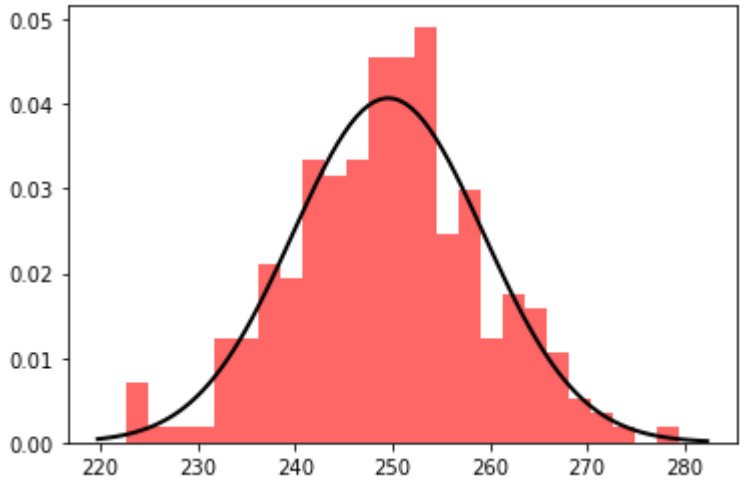
Out[18]:

```
Text(0, 0.5, 'Preço')
```



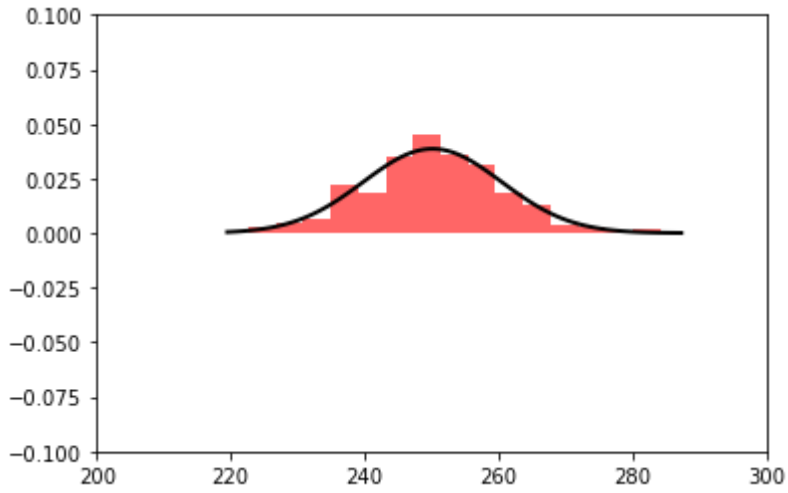
In [44]:

```
#7)
x = np.random.normal(250, 10, 250)
media, std = norm.fit(x)
plt.hist(x, bins=25, density=True, alpha=0.6, color='r')
xmin, xmax = plt.xlim()
x2 = np.linspace(xmin, xmax, 100)
x3 = norm.pdf(x2, media, std)
plt.plot(x2, x3, 'k', linewidth=2)
plt.show()
```



In [51]:

```
#8)
x = np.random.normal(250, 10, 250)
media, std = norm.fit(x)
plt.hist(x, bins=15, density=True, alpha=0.6, color='r')
xmin, xmax = plt.xlim()
x2 = np.linspace(xmin, xmax, 100)
x3 = norm.pdf(x2, media, std)
plt.xlim(200,300)
plt.ylim(-0.1,0.1)
plt.plot(x2, x3, 'k', linewidth=2)
plt.show()
```



In [79]:

```
#9)

c=[]
for i in range(101):
    c.append(i+1)
    i=i+1
c2=[]
c3=[]
c5=[]
c7=[]
co=[]
for i in range(len(c)):
    if i % 2 == 0:
        c2.append(i)
    elif i % 3 ==0:
        c3.append(i)
    elif i % 5 ==0:
        c5.append(i)
    elif i % 7 == 0:
        c7.append(i)
    else:
        co.append(i)

ct =[len(c2),len(c3),len(c5),len(c7),len(co)]
print(ct)
lab = 'c 2','c 3','c 5','c 7','c outros'
cc = '#FF4500','#B23AEE','#FF69B4','#BB0FA0','#00BFFF'
plt.pie(ct, labels = lab, colors = cc)
plt.legend()
```

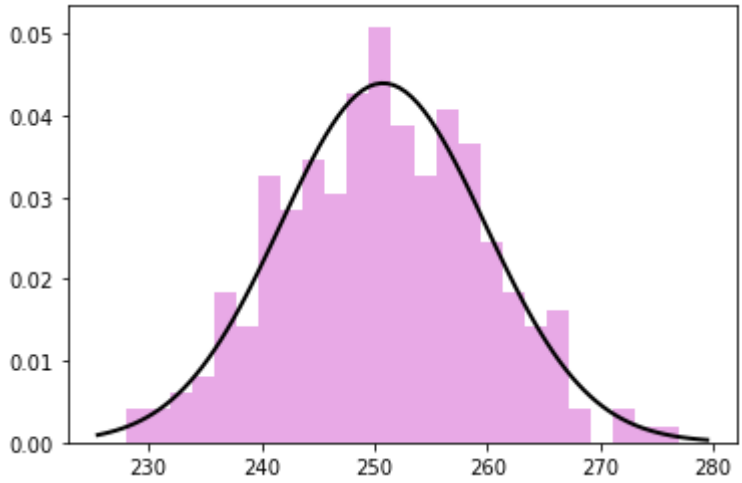
```
[51, 17, 7, 4, 22]
```

Out[79]:

```
<matplotlib.legend.Legend at 0x2d555318b20>
```

In [82]:

```
#10 - desafio)
# Já feito no próprio exercício 7
# Segue, novamente:
x = np.random.normal(250, 10, 250)
media, std = norm.fit(x)
plt.hist(x, bins=25, density=True, alpha=0.6, color='#DA70D6')
xmin, xmax = plt.xlim()
x2 = np.linspace(xmin, xmax, 100)
x3 = norm.pdf(x2, media, std)
plt.plot(x2, x3, 'k', linewidth=2)
plt.show()
```



Lista Numpy

In [129]:

```
#1)
a=np.zeros(40)
A = np.array(a)
print(A)
type(A)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

Out[129]:

```
numpy.ndarray
```

In [138]:

```
#2)
lista2=[]
for i in range(270,300):
    lista2.append(i+1)
    i=i+1
B = np.array(lista2)
print(B)
type(B)
```

```
[271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
 289 290 291 292 293 294 295 296 297 298 299 300]
```

Out[138]:

```
numpy.ndarray
```

In [139]:

```
#3)
lista3=[]
for i in range(10):
    if i%3==0:
        lista3.append(i)
C = np.array(lista3)
print(C)
type(C)
```

```
[0 3 6 9]
```

Out[139]:

```
numpy.ndarray
```

In [140]:

```
#4)
D1 = np.random.randint(1,50,(1,50))
print(D1)
type(D1)
```

```
[[37 29  9 37 34  3 21 45 49 32 16 21 12 41 10 30 42 48 39 29 34  7 47 40
  13 42  1 45  4 39 34 38 22 44 39 34 35 48 37 23 12 23  7 27 38 25 33 41
  27 14]]
```

Out[140]:

```
numpy.ndarray
```

In [151]:

```
#5)
D = np.concatenate((C,B))
print("D =",D)
print("\nValores com índices pares: \n")
for i in range(len(D)):
    if i % 2 == 0:
        print(D[i])
```

```
D = [  0   3   6   9 271 272 273 274 275 276 277 278 279 280 281 282 283 2
84
 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300]

Valores com índices pares:

0
6
271
273
275
277
279
281
283
285
287
289
291
293
295
297
299
```

In [3]:

```
# 6)

def transpor(matriz):
    t =np.transpose(matriz)
    return t
transpor([[0,1,2,3,4],[5,6,7,8,9]])
```

Out[3]:

```
array([[0, 5],
       [1, 6],
       [2, 7],
       [3, 8],
       [4, 9]])
```

In [21]:

```
#7)



import time
import datetime
from scipy import stats
from IPython import display
import yahoofinancials as yf
ticker = 'BAC' # Bank of America

start_date = '2019-03-15'
end_date =  '2021-03-15'

data = yf.YahooFinancials(ticker).get_historical_price_data(start_date, end_date, 'dail
y')
raw = pd.DataFrame(data[ticker]['prices']).dropna()
# Converter a data para o tipo correto datetime
raw['formatted_date'] = pd.to_datetime(raw['formatted_date'])
# Indica a data como o índice de cada linha
raw = raw.set_index('formatted_date')
# Removendo as colunas que não interessam
df = raw.iloc[:,1:-1]
# Visualiza o resultado
df.head()


o = np.std(df['close'])
p = np.mean(df['close'])
q = np.max(df['close'])
r = np.min(df['close'])
print("-------- ticker:BAC = Bank of America ----------------------------------")
print("Desvio Padrão:",o)
print("Média:",p)
print("Máximo:",q)
print("Mínimo:",r)

opqr = np.array([o,p,q,r])
print(opqr)
type(opqr)
```

```
-------- ticker:BAC = Bank of America ----------------------------------
Desvio Padrão: 4.032344787388834
Média: 28.430357832557874
Máximo: 37.939998626708984
Mínimo: 18.079999923706055
[ 4.03234479 28.43035783 37.93999863 18.07999992]
```

Out[21]:

```
numpy.ndarray
```

In [26]:

```
#8)
Beta=[[2,8,9,4],[9,4,9,4],[4,5,9,7],[2,9,4,3]]
Alpha = repr(Beta).count('9')
Fi = repr(Beta).count('4')
Gama = repr(Beta).count("9, 4")
print("Contagem de 9s: ",Alpha)
print("Contagem de 4s: ",Fi)
print("Contagem da sequência: ",Gama)
print("Contagem de vezes em que o 9 não está em uma sequência: ",Alpha - Gama)
print("Contagem de vezes em que o 4 não está em uma sequência: ",Fi - Gama)
```

```
Contagem de 9s:  5
Contagem de 4s:  5
Contagem da sequência:  4
Contagem de vezes em que o 9 não está em uma sequência:  1
Contagem de vezes em que o 4 não está em uma sequência:  1
```

In [ ]: