

# incontro-02-cor

September 10, 2024

## 0.1 Liste

```
[1]: cose = ['dado', 'bullone', 'fascetta', 'vite', 'ranella']  
cose
```

```
[1]: ['dado', 'bullone', 'fascetta', 'vite', 'ranella']
```

```
[2]: numeri = [12,15,59,93,17,1]  
numeri
```

```
[2]: [12, 15, 59, 93, 17, 1]
```

```
[3]: misto = ['gatto',23,12.5,'a','zeta']  
misto
```

```
[3]: ['gatto', 23, 12.5, 'a', 'zeta']
```

### 0.1.1 Operazioni sulle liste

```
[4]: # aggiungere in coda  
cose.append('brugola')  
cose
```

```
[4]: ['dado', 'bullone', 'fascetta', 'vite', 'ranella', 'brugola']
```

```
[5]: # ordinare  
numeri.sort()  
numeri
```

```
[5]: [1, 12, 15, 17, 59, 93]
```

```
[6]: cose.sort()  
cose
```

```
[6]: ['brugola', 'bullone', 'dado', 'fascetta', 'ranella', 'vite']
```

```
misto.sort() misto
```



'G',  
'H',  
'H',  
'H',  
'H',  
'H',  
'I',  
'I',  
'I',  
'J',  
'J',  
'J',  
'J',  
'J',  
'K',  
'K',  
'L',  
'L',  
'L',  
'L',  
'M',  
'M',  
'M',  
'N',  
'N',  
'N',  
'N',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'P',  
'P',  
'P',  
'Q',  
'Q',  
'Q',  
'R',  
'R',  
'R',  
'R',  
'R',

'S',  
'S',  
'S',  
'T',  
'T',  
'T',  
'T',  
'T',  
'T',  
'T',  
'U',  
'U',  
'U',  
'U',  
'V',  
'V',  
'W',  
'W',  
'X',  
'X',  
'X',  
'X',  
'X',  
'Y',  
'Y',  
'Y',  
'Y',  
'Z',  
'Z',  
'Z'],  
['A',  
'A',  
'A',  
'A',  
'A',  
'A',  
'B',  
'B',  
'B',  
'B',  
'B',  
'B',  
'C',  
'D',  
'D',  
'E',  
'E',  
'E',  
'E',  
'E',  
'F',  
'F',

'F',  
'F',  
'F',  
'F',  
'G',  
'G',  
'G',  
'H',  
'H',  
'H',  
'H',  
'H',  
'I',  
'I',  
'I',  
'J',  
'J',  
'J',  
'J',  
'J',  
'K',  
'K',  
'L',  
'L',  
'L',  
'L',  
'M',  
'M',  
'M',  
'N',  
'N',  
'N',  
'N',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'O',  
'P',  
'P',  
'P',  
'Q',  
'Q',

```
'Q',  
'R',  
'R',  
'R',  
'R',  
'R',  
'R',  
'S',  
'S',  
'S',  
'T',  
'T',  
'T',  
'T',  
'T',  
'T',  
'U',  
'U',  
'U',  
'U',  
'V',  
'V',  
'W',  
'W',  
'X',  
'X',  
'X',  
'X',  
'Y',  
'Y',  
'Y',  
'Y',  
'Z',  
'Z',  
'Z']])
```

## 0.2 Vettori (array)

```
[13]: import numpy as np
```

```
[9]: numeri = np.array(numeri, dtype = int)  
numeri
```

```
[9]: array([ 1, 12, 15, 17, 59, 93])
```

```
[10]: inumeri = np.zeros(6, dtype = int)  
inumeri
```

```
[10]: array([0, 0, 0, 0, 0, 0])
```

```
[11]: import random
```

```
[12]: for k in range(6) :  
      x = random.randint(15,139)  
      inumeri[k] = x  
      inumeri
```

```
[12]: array([133, 50, 111, 17, 51, 105])
```

```
[13]: n = numeri + inumeri  
      n
```

```
[13]: array([134, 62, 126, 34, 110, 198])
```

```
[14]: n = numeri * inumeri  
      n
```

```
[14]: array([ 133, 600, 1665, 289, 3009, 9765])
```

```
[15]: n = numeri - inumeri  
      n
```

```
[15]: array([-132, -38, -96, 0, 8, -12])
```

```
[16]: n = np.dot(numeri,inumeri)  
      n
```

```
[16]: 15461
```

```
[23]: V = []  
      k = 0  
      n = 50  
      while k < n :  
          k += 1  
          x = random.random()*8  
          y = random.random()*8  
          if x == 0 and y == 0 :  
              k -= 1  
          else :  
              V.append([x,y])  
      for h in range(n-1) :  
          for j in range(h+1,n) :  
              if np.dot(V[h],V[j]) == 0 :  
                  print(V[h],V[j])  
      print('nessuna coppia di vettori è perpendicolare')
```

nessuna coppia di vettori è perpendicolare

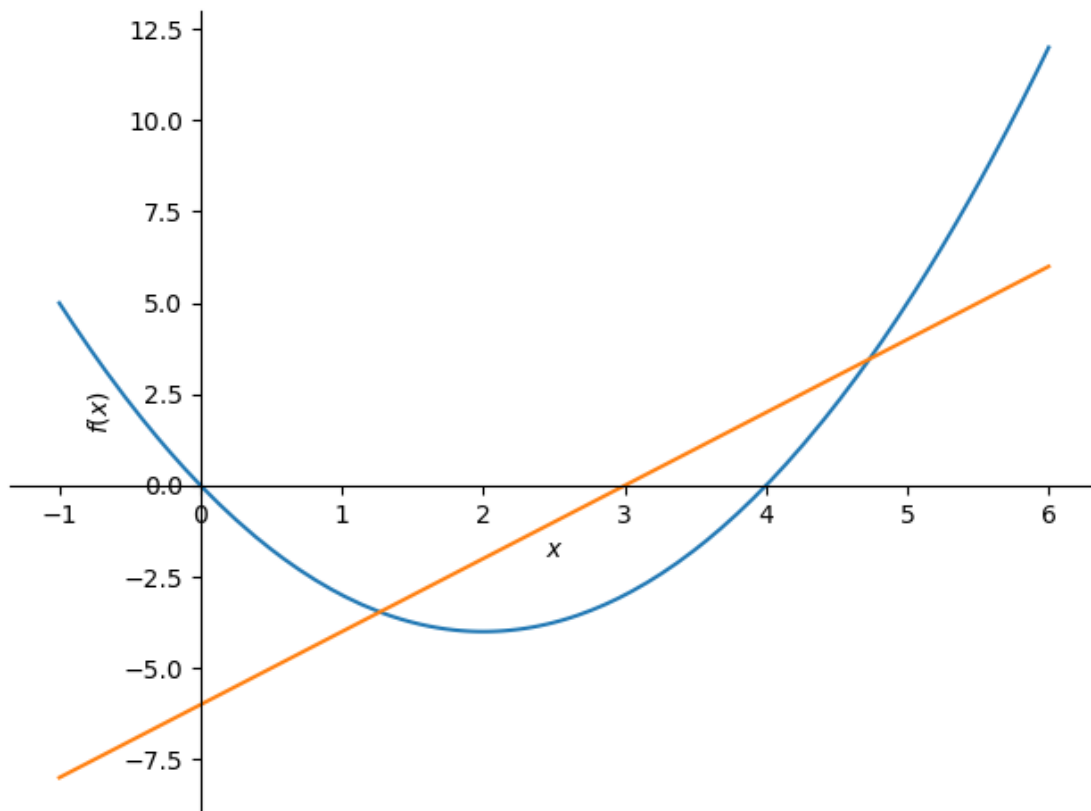
### 0.3 Grafica in Sympy

Parabola e retta

```
[17]: from sympy import *  
      init_printing()
```

```
[18]: #parabola  
      def p(x) :  
          return x**2-4*x  
      #retta  
      def r(x) :  
          return 2*x-6
```

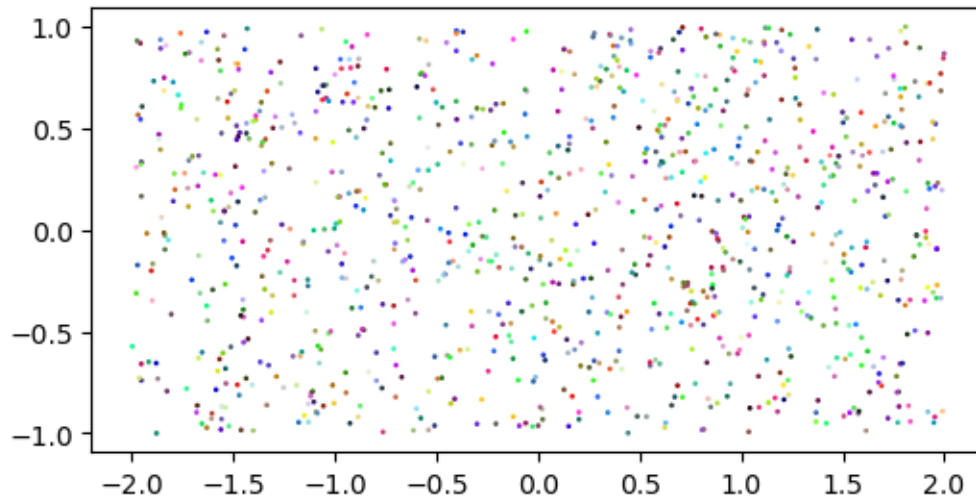
```
[19]: x = symbols('x')  
      Gp = plot(p(x), (x,-1,6), show = False)  
      Gr = plot(r(x), (x,-1,6), show = False)  
      Gp.append(Gr[0])  
      Gp.show()
```



## 0.4 Grafica con liste

```
[20]: import matplotlib.pyplot as plt
```

```
[44]: n = 1000
figura = plt.figure(figsize=(6,3))
plt.axis('equal')
punti = []
for k in range(n) :
    x = 4*random.random()-2
    y = 2*random.random()-1
    r = random.random()
    g = random.random()
    b = random.random()
    punti.append([x,y,r,g,b])
tot = len(punti)
punti = np.array(punti,dtype=np.float64)
iColori = np.zeros((tot,4))
iColori[:,0] = punti[:,2]
iColori[:,1] = punti[:,3]
iColori[:,2] = punti[:,4]
iColori[:,3] = 1.0
plt.scatter(punti[:,0], punti[:,1], 0.8, color=iColori)
plt.show()
```



## 0.5 Problema zombie

La porta degli zombie è una fessura di larghezza  $\ell$

in un muro impenetrabile. Incessantemente migliaia di zombie camminano in modo disordinato verso questo muro e a ogni secondo uno di essi lo raggiunge in un punto aleatorio. Solo gli zombie

fortunati si trovano di fronte alla porta e possono superarla, gli altri invece urtano contro il muro. Sia  $\ell$  la larghezza della porta e  $d$  la larghezza del muro,  $t_{oss}$  il tempo, in secondi, di osservazione.

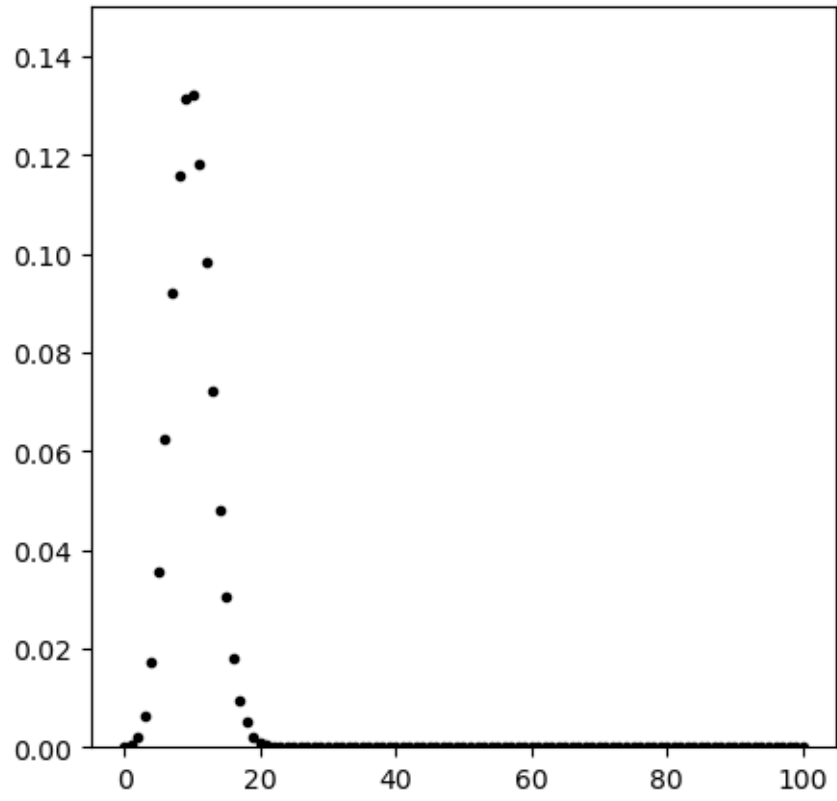
Costruisci una statistica, in cui si registra la frequenza di zombie che attraversano la porta nel primo, secondo, ..., centesimo secondo. È bene ripetere molte volte l'osservazione per meglio stimare quanto richiesto.

Output sia numerico che grafico.

```
[1]: # alcuni già utilizzati ma indico quelli necessari
import math
import random
import numpy as np
import matplotlib.pyplot as plt
```

```
[4]: def passaZombie() :
    passa = False
    posizione = d/2-random.random()*d
    if posizione>=-elle/2 and posizione <=elle/2:
        passa = True
    return passa
```

```
[10]: d = 10
elle = 1
X = []
Y = []
poisp = []
#
toss = 100
prove = 100000
zombie = np.zeros(toss+1)
for k in range(1,prove):
    conta = 0
    for n in range(1,toss):
        if passaZombie():
            conta += 1
    zombie[conta] += 1
plt.figure(figsize=(5,5))
plt.ylim(ymax = 0.15, ymin = 0)
zombie = np.array(zombie,dtype=np.float64)
for h in range(0,toss+1):
    X.append(h)
    p = zombie[h]/prove
    Y.append(p)
Y = np.array(Y, dtype=np.float64)
m = (elle/d)*toss
plt.plot(X,Y, 'k.')
plt.show()
```



[ ]: