

# Random-numbers-1

April 2, 2024

```
[65]: import numpy as np
import matplotlib.pyplot as plt
```

```
[70]: np.random.rand(2)
```

```
[70]: array([0.51599229, 0.41455642])
```

```
[79]: np.random.seed(1234)
```

```
[81]: np.random.randn(2)
```

```
[81]: array([ 1.43270697, -0.3126519 ])
```

```
[84]: np.random.randint(1,21,10)
```

```
[84]: array([10, 12, 17,  4,  3, 20, 13,  2, 12, 20])
```

```
[85]: np.random.random_sample(200)
```

```
[85]: array([0.53814784, 0.7906221 , 0.46583634, 0.43533225, 0.56947866,
          0.969259 , 0.04055615, 0.54811957, 0.4625766 , 0.37647223,
          0.32791208, 0.81352893, 0.64655232, 0.04742648, 0.99495757,
          0.68923564, 0.92954594, 0.91811655, 0.97530172, 0.39700197,
          0.26262609, 0.43015136, 0.76453077, 0.59973081, 0.08094696,
          0.70454447, 0.16401332, 0.03234935, 0.32815036, 0.47386 ,
          0.06808472, 0.3827107 , 0.11855414, 0.89632852, 0.76430853,
          0.37540494, 0.02581418, 0.90653093, 0.78641778, 0.61931523,
          0.21158894, 0.292304 , 0.66154467, 0.88626098, 0.67117646,
          0.32968079, 0.2073618 , 0.01163207, 0.07154342, 0.73005591,
          0.9798396 , 0.69001621, 0.99057483, 0.69053432, 0.94812322,
          0.58364465, 0.53155327, 0.16865734, 0.15880774, 0.93796532,
          0.71826456, 0.47654265, 0.88365336, 0.40420231, 0.17146805,
          0.13183202, 0.41190816, 0.02485604, 0.5635608 , 0.78187792,
          0.26705997, 0.21425496, 0.17755704, 0.42926568, 0.97211312,
          0.04653158, 0.91735447, 0.15893009, 0.94338268, 0.76316153,
          0.05387827, 0.25408163, 0.92797263, 0.8383115 , 0.15692453,
          0.69077595, 0.36694644, 0.93747272, 0.61336482, 0.69934982,
          0.50294625, 0.7111109 , 0.13438585, 0.82893194, 0.74284587,
```

```

0.45703374, 0.07910336, 0.37304659, 0.93363636, 0.41872481,
0.23421157, 0.57248483, 0.57211088, 0.41689298, 0.62588308,
0.22036227, 0.62205913, 0.47767162, 0.97434208, 0.77298466,
0.02713854, 0.22102235, 0.12032832, 0.17527398, 0.42946156,
0.65776918, 0.56589859, 0.56903498, 0.65419601, 0.36855768,
0.95238467, 0.19677048, 0.84993003, 0.96045829, 0.38111827,
0.3309362 , 0.26092345, 0.66549109, 0.18179543, 0.37679994,
0.01425856, 0.3391353 , 0.4013514 , 0.46757367, 0.6521055 ,
0.99719214, 0.51746228, 0.40361156, 0.05844676, 0.04519571,
0.62708044, 0.00806841, 0.41788026, 0.00634233, 0.42420633,
0.36278279, 0.61878759, 0.41900078, 0.01371138, 0.41334959,
0.21985716, 0.52993668, 0.77967004, 0.52172732, 0.50547663,
0.06520625, 0.4082489 , 0.70322202, 0.92442994, 0.55011142,
0.33244958, 0.6431856 , 0.95203233, 0.21441909, 0.90481733,
0.52859399, 0.7575487 , 0.52235971, 0.59450566, 0.37082039,
0.08223248, 0.52991991, 0.75915808, 0.01777811, 0.03533422,
0.94394711, 0.40556919, 0.4479019 , 0.78263591, 0.57419279,
0.87642588, 0.36038531, 0.42729407, 0.83383316, 0.11213941,
0.86878024, 0.79134595, 0.49167366, 0.39582695, 0.03559736,
0.17168925, 0.18904467, 0.10286923, 0.90904121, 0.58232868,
0.89883124, 0.43500218, 0.07836778, 0.22470752, 0.69762604])

```

```

[86]: def compute_pi(N):
      count = 0
      for i in range(N):
          x = np.random.rand()
          y = np.random.rand()
          if x**2 + y**2 <= 1:
              count += 1

      return 4*count/N

```

```

[87]: compute_pi(2000)

```

```

[87]: 3.114

```

```

[64]: vals = [compute_pi(i) for i in range(2,20000,10)];

```

```

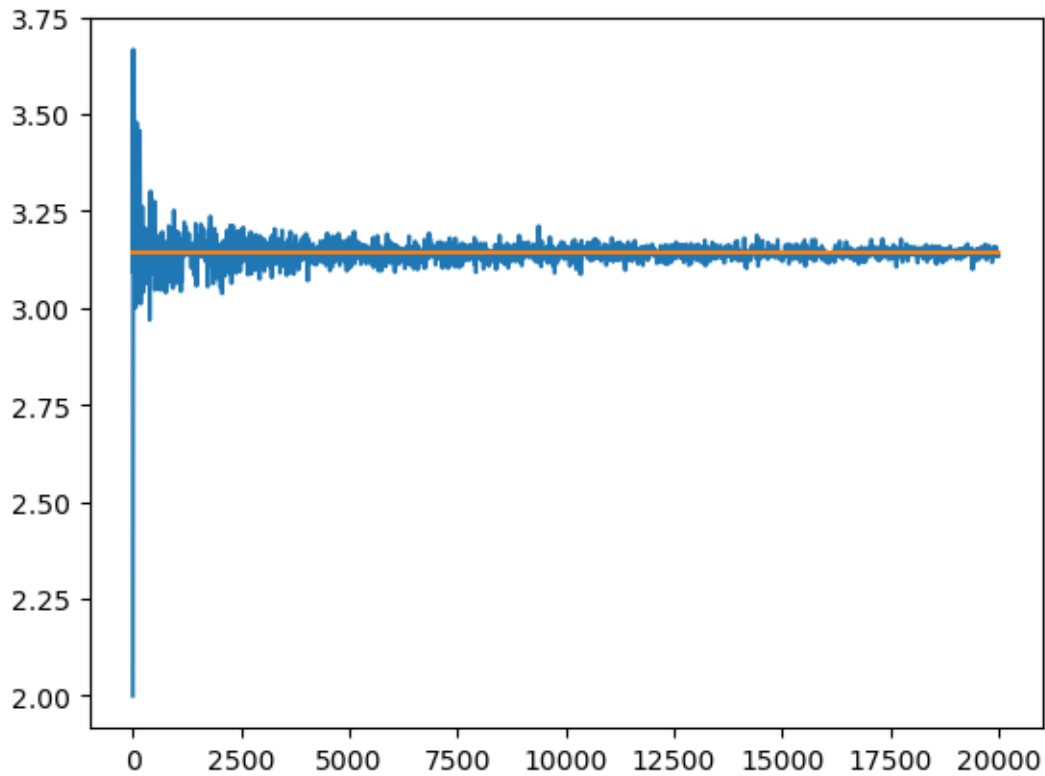
[66]: plt.plot(range(2,20000,10), vals)
      plt.plot([0, 20000],[3.141596, 3.141596])

```

```

[66]: [<matplotlib.lines.Line2D at 0x7f18d8105400>]

```



```
[23]: vals = np.random.rand(20000);  
      f, x = np.histogram(vals, bins=10);
```

```
[24]: np.size(x)
```

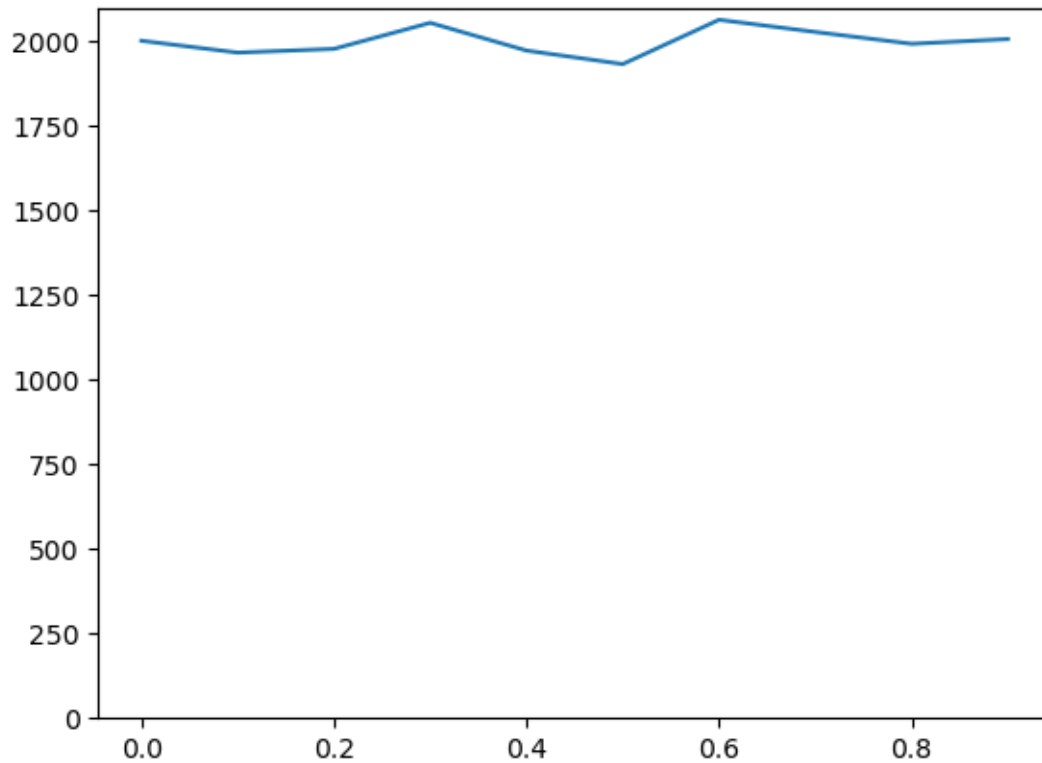
```
[24]: 11
```

```
[25]: x
```

```
[25]: array([7.51677490e-05, 1.00056350e-01, 2.00037533e-01, 3.00018716e-01,  
          3.99999899e-01, 4.99981081e-01, 5.99962264e-01, 6.99943447e-01,  
          7.99924629e-01, 8.99905812e-01, 9.99886995e-01])
```

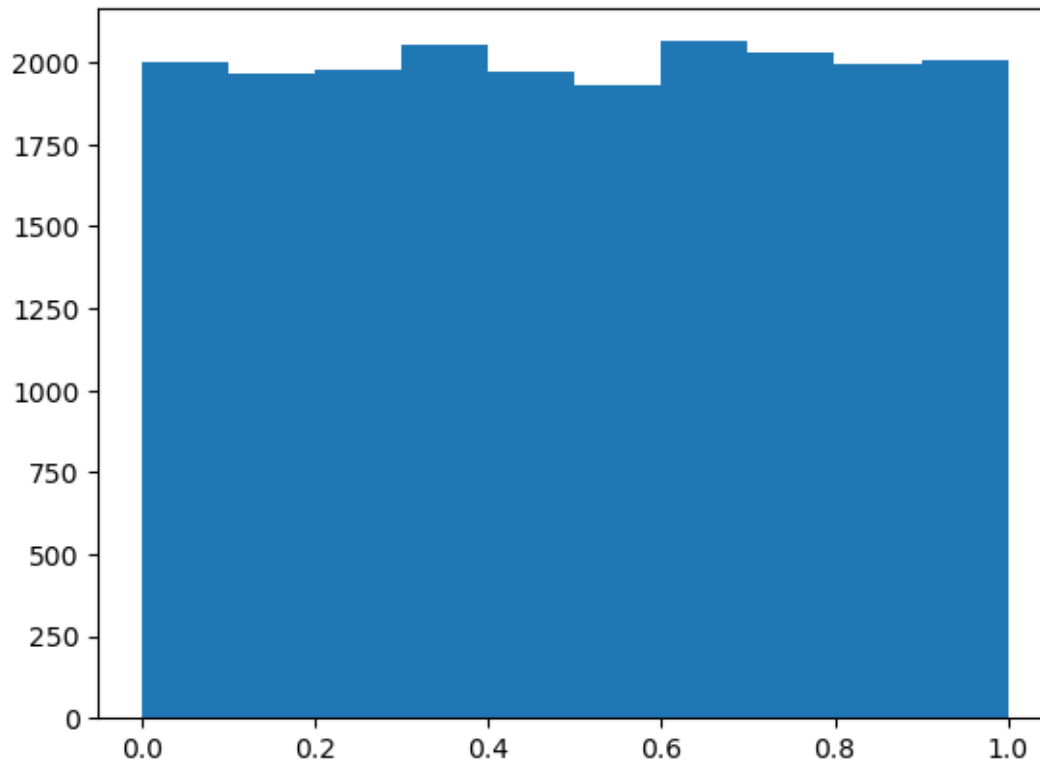
```
[26]: plt.plot(x[0:-1],f)  
      plt.ylim(0,2100)
```

```
[26]: (0.0, 2100.0)
```



```
[27]: plt.hist(vals)
```

```
[27]: (array([2002., 1967., 1978., 2055., 1973., 1933., 2064., 2028., 1993.,
        2007.]),
      array([7.51677490e-05, 1.00056350e-01, 2.00037533e-01, 3.00018716e-01,
        3.99999899e-01, 4.99981081e-01, 5.99962264e-01, 6.99943447e-01,
        7.99924629e-01, 8.99905812e-01, 9.99886995e-01])),
      <BarContainer object of 10 artists>)
```

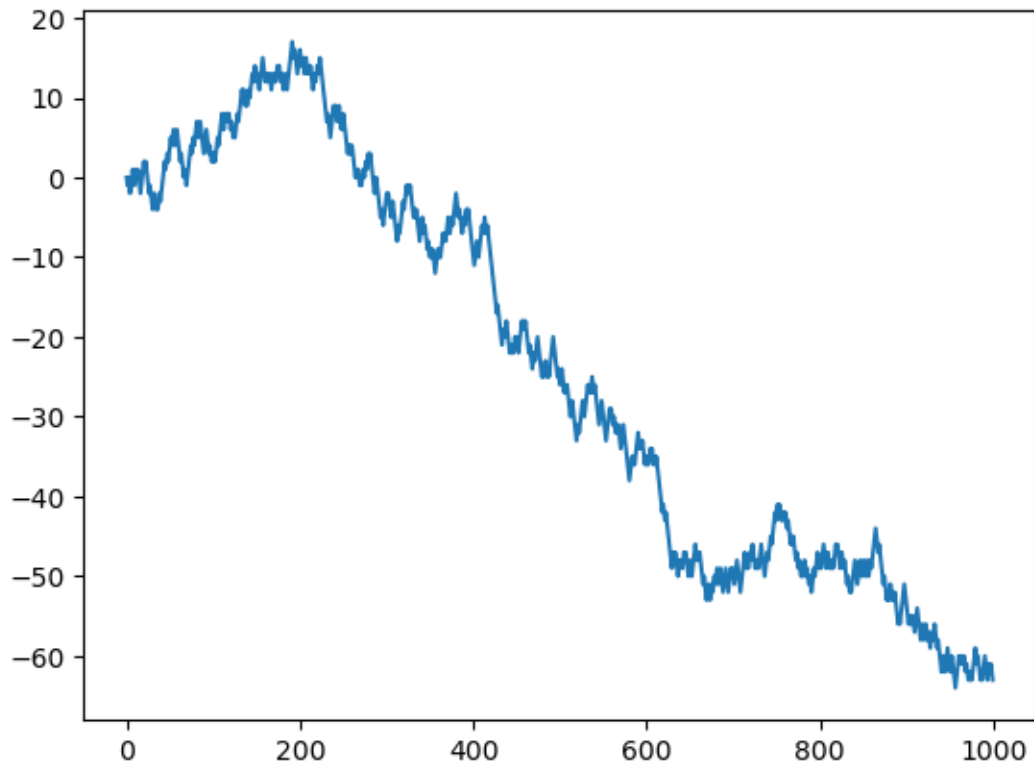


```
[92]: def take_a_step():  
        if np.random.rand() < 0.5:  
            return -1  
        return 1
```

```
steps = 1000  
path = np.zeros(steps)  
pos = 0; k = 0  
for s in range(steps):  
    path[k] = pos  
    pos += take_a_step()  
    k += 1
```

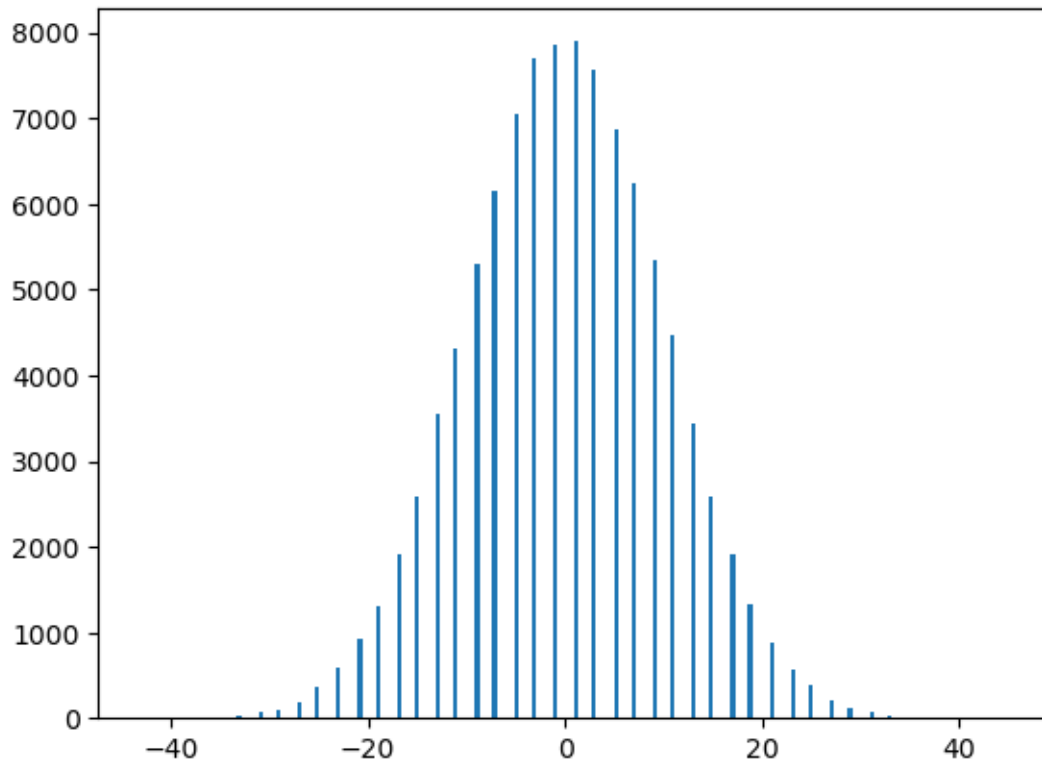
```
[93]: plt.plot(range(steps), path)
```

```
[93]: [<matplotlib.lines.Line2D at 0x7f2fc6f559a0>]
```



```
[61]: steps = 101
walkers = 100000
finpos = np.zeros(walkers)
k = 0
for w in range(walkers):
    pos = 0
    for s in range(steps):
        pos += take_a_step()
    finpos[k] = pos
    k += 1
```

```
[64]: plt.hist(finpos,bins=200);
```



```
[100]: a = 1.0
k = 0
while a+1.0>1.0:
    print(k, a)
    a = a/2
    k += 1
```

```
0 1.0
1 0.5
2 0.25
3 0.125
4 0.0625
5 0.03125
6 0.015625
7 0.0078125
8 0.00390625
9 0.001953125
10 0.0009765625
11 0.00048828125
12 0.000244140625
13 0.0001220703125
14 6.103515625e-05
```

15 3.0517578125e-05  
16 1.52587890625e-05  
17 7.62939453125e-06  
18 3.814697265625e-06  
19 1.9073486328125e-06  
20 9.5367431640625e-07  
21 4.76837158203125e-07  
22 2.384185791015625e-07  
23 1.1920928955078125e-07  
24 5.960464477539063e-08  
25 2.9802322387695312e-08  
26 1.4901161193847656e-08  
27 7.450580596923828e-09  
28 3.725290298461914e-09  
29 1.862645149230957e-09  
30 9.313225746154785e-10  
31 4.656612873077393e-10  
32 2.3283064365386963e-10  
33 1.1641532182693481e-10  
34 5.820766091346741e-11  
35 2.9103830456733704e-11  
36 1.4551915228366852e-11  
37 7.275957614183426e-12  
38 3.637978807091713e-12  
39 1.8189894035458565e-12  
40 9.094947017729282e-13  
41 4.547473508864641e-13  
42 2.2737367544323206e-13  
43 1.1368683772161603e-13  
44 5.684341886080802e-14  
45 2.842170943040401e-14  
46 1.4210854715202004e-14  
47 7.105427357601002e-15  
48 3.552713678800501e-15  
49 1.7763568394002505e-15  
50 8.881784197001252e-16  
51 4.440892098500626e-16  
52 2.220446049250313e-16

[99]: 2\*\*(-1020)\*2\*\*(-52)

[99]: 2e-323

[ ]: