Random-numbers

April 10, 2025

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Generating random numbers between 0.0 and 1.0

[4]:
```
np.random.rand()
```

[4]:
```
0.7609868970867905
```

Set the random number generator at a specific point

[8]:
```
np.random.seed(1234)
```

Generate (m x n) random numbers

[9]:
```
np.random.rand(5,2)
```

[9]:
```
array([[0.19151945, 0.62210877],
       [0.43772774, 0.78535858],
       [0.77997581, 0.27259261],
       [0.27646426, 0.80187218],
       [0.95813935, 0.87593263]])
```

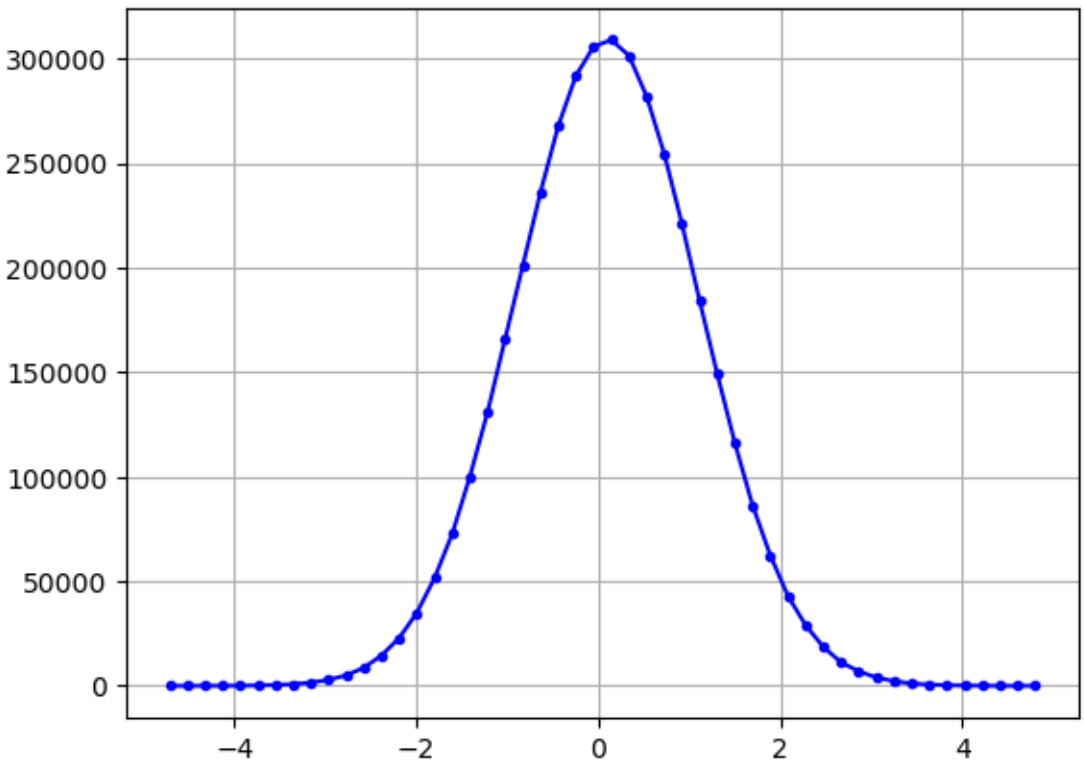
Sampling random numbers from a normal distribution

[21]:
```
np.random.randn(5)
```

[21]:
```
array([ 1.32115819, -1.54690555, -0.20264632, -0.65596934,  0.19342138])
```

Check if it is Gaussian

[138]:
```
nums = np.random.randn(4000000);
values,bins = np.histogram(nums,bins=50);
plt.plot(bins[1:], values, 'b.-')
plt.grid()
```



Simple integration

[139]:
```
def calculate_pi(N):
    count = 0
    for i in range(N):
        x, y = np.random.rand(2)
        if x**2 + y**2 <= 1.0:
            count += 1
    return 4*count/N
```

[140]:
```
calculate_pi(1000000), np.pi
```

[140]:
```
(3.142452, 3.141592653589793)
```

0.0.1 Random walk

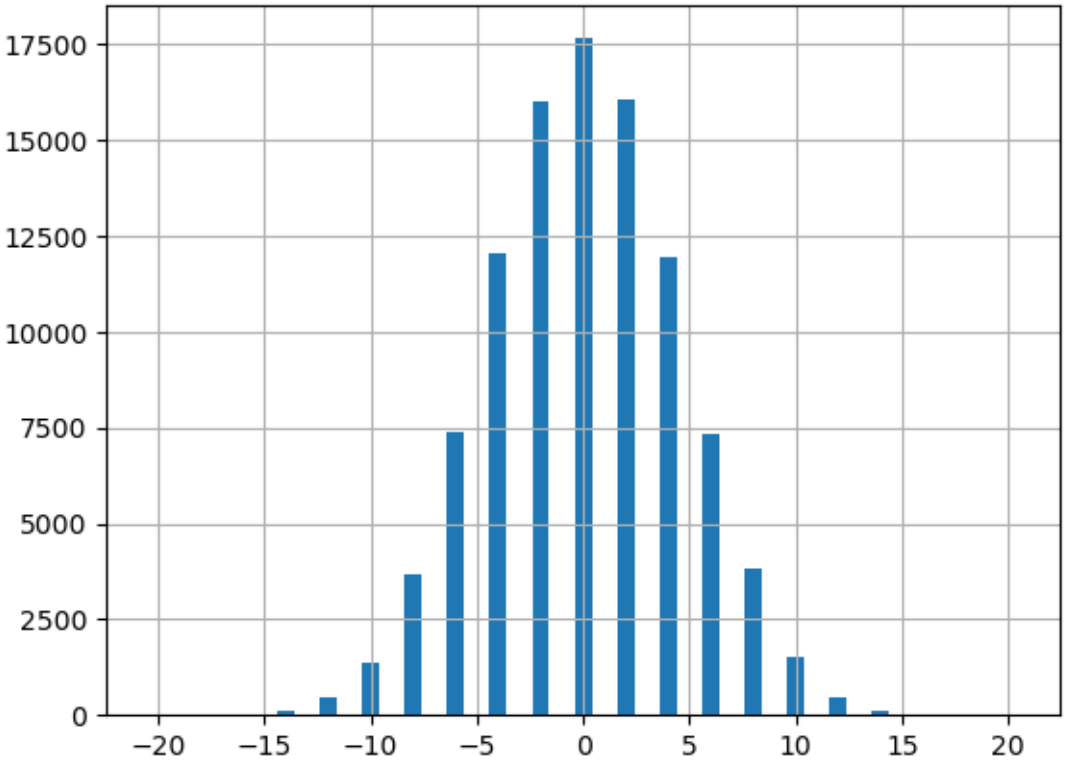
[141]:
```
def take_a_step():
    s = np.random.rand()
    if s > 0.5:
        return 1
    else:
        return -1
```

[143]:
```
num_walkers = 100000
num_step = 20

final_positions = np.zeros(num_step+1)
for walker in range(num_walkers):
    finpos = 0
    for step in range(num_step):
        finpos += take_a_step()
    pos_in_array = int((finpos+num_step)/2.0)
    #print(finpos, pos_in_array)
    final_positions[pos_in_array] += 1

positions = np.linspace(-num_step, num_step, num_step+1)
```

[144]:
```
plt.bar(positions, final_positions,)
plt.grid()
```



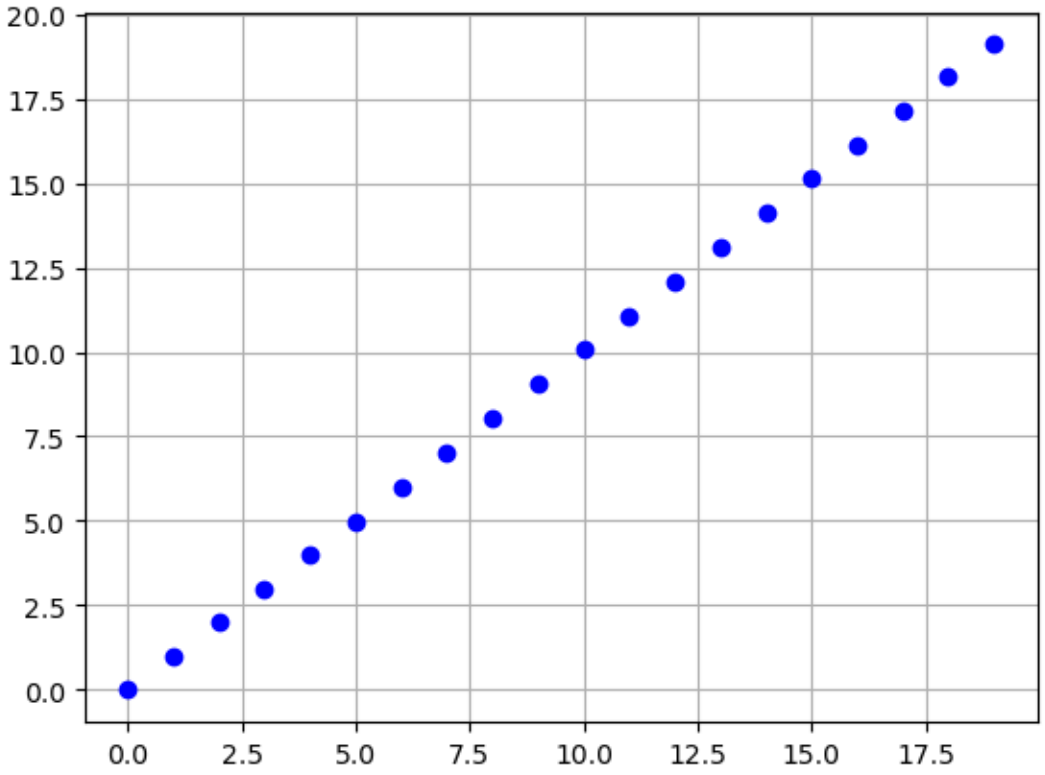
[153]:
```
num_walkers = 100000
num_steps = 20

positions = np.zeros(num_walker, dtype=int)
xaverage = np.zeros(num_steps)
```

```
for step in range(num_steps):
    xaverage[step] = sum([i**2 for i in positions])/num_walkers
    for walker in range(num_walkers):
        positions[walker] += take_a_step()

steps = [i for i in range(num_steps)]
```

[154]:
```
plt.plot(steps,xaverage,'bo')
plt.grid()
```



[155]:
```
xaverage
```

[155]:
```
array([ 0.     ,  1.     ,  1.99224,  2.98832,  3.98444,  4.99096,
        5.99876,  7.01392,  8.0442 ,  9.05328, 10.05872, 11.0592 ,
       12.08488, 13.09272, 14.10644, 15.12976, 16.11456, 17.152  ,
       18.13808, 19.09352])
```

[ ]:
```

```