Random Walk

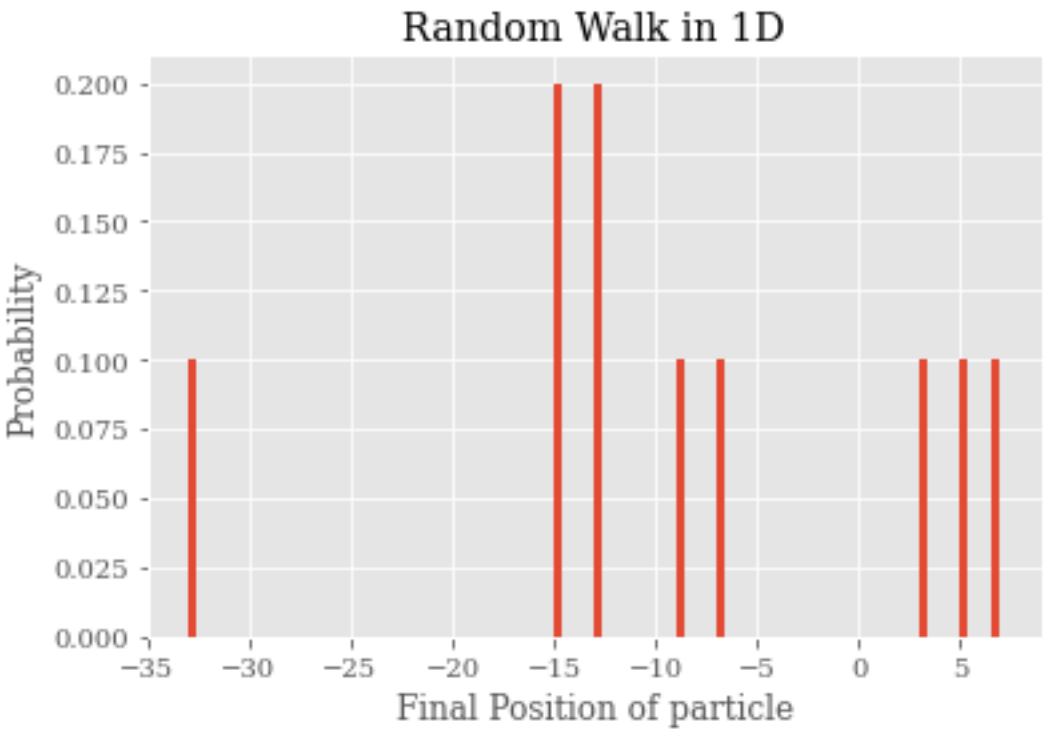
September 4, 2020

[4]:
```
import matplotlib.pyplot as plt
import pylab as pl
import numpy as np
import random
```

[64]:
```
def random_walk(N):
    N = int(N) # no.of random walks
    steps = 101 # no. of steps in each walk
    x=[] # final displacement

    step_dirn = [-1,1]

    for i in range(N):
        u = random.choices(step_dirn, k = steps)
        x.append(sum(u))

    weights = np.ones_like(x)/N


    plt.style.use('ggplot')
    plt.rcParams["font.family"] = "serif"
    plt.rcParams["mathtext.fontset"] = "dejavuserif"
    plt.hist(x, weights=weights,bins=100)
    plt.title('Random Walk in 1D')
    plt.ylabel('Probability')
    plt.xlabel('Final Position of particle')
    plt.savefig('Random_walk_hist.png',dpi=600)
    plt.show()

    return None
```

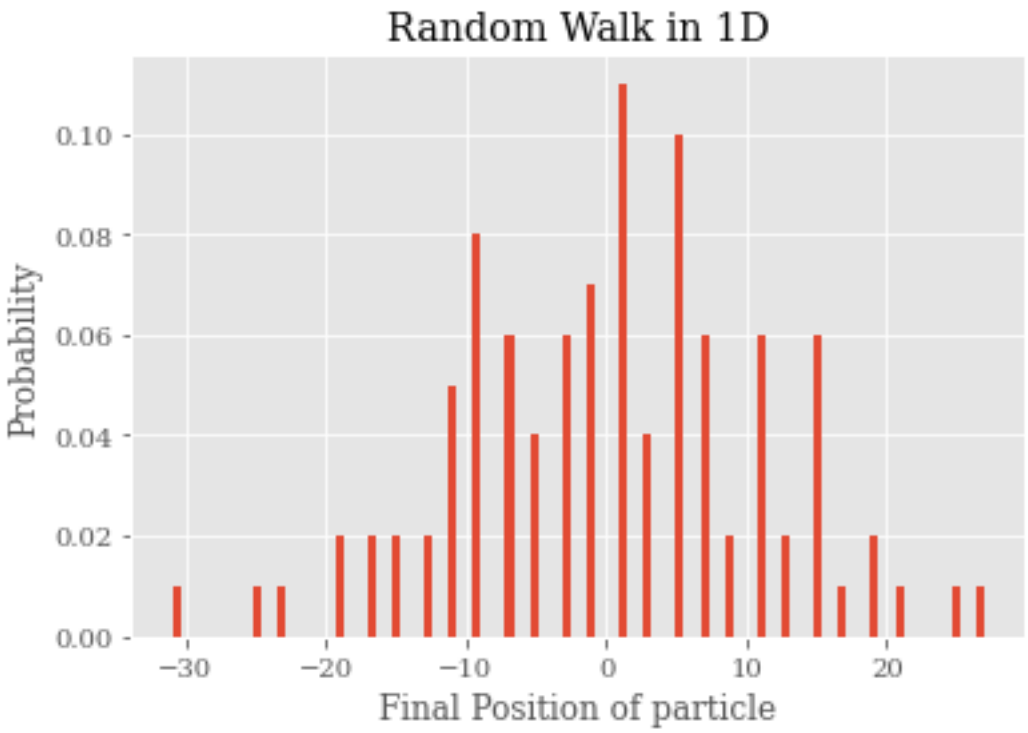
0.1 Number of random walks = 10

[65]:
```
random_walk(10)
```



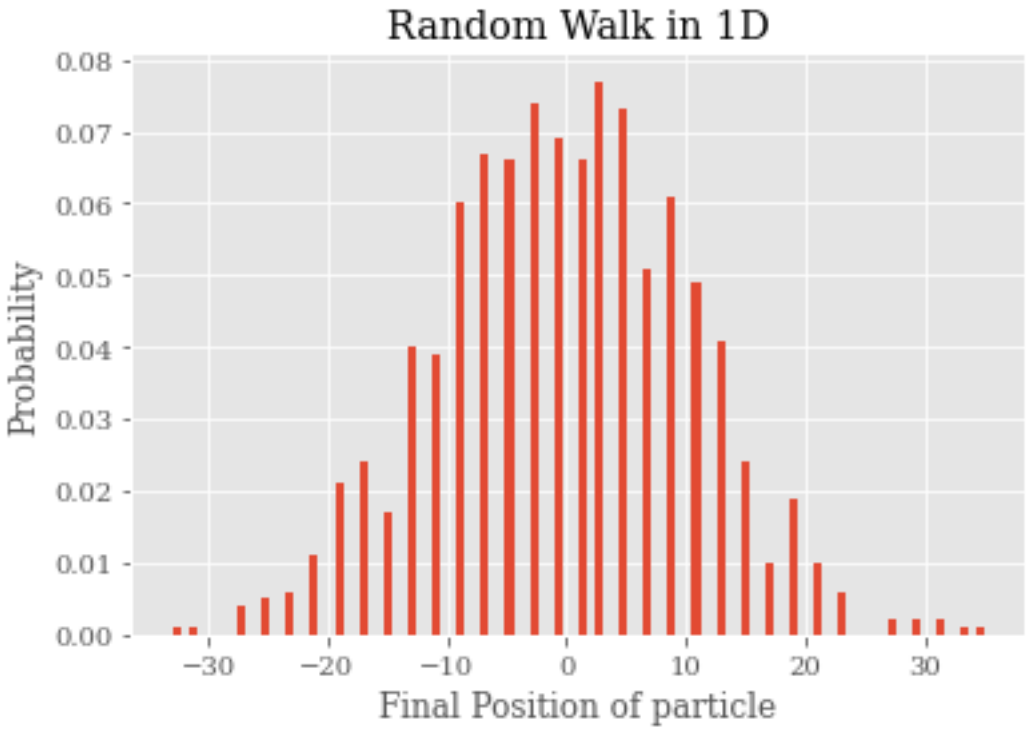
0.2 Number of random walks = 100

[66]:
```
random_walk(100)
```



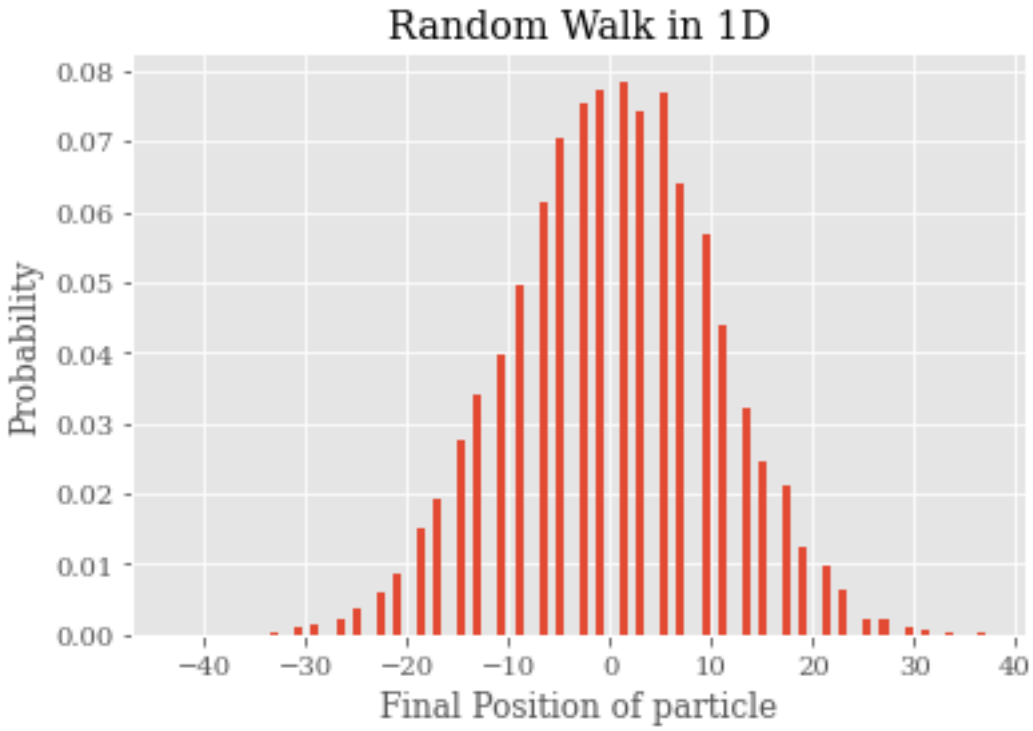
0.3 Number of random walks = 1000

[67]:
```
random_walk(1000)
```



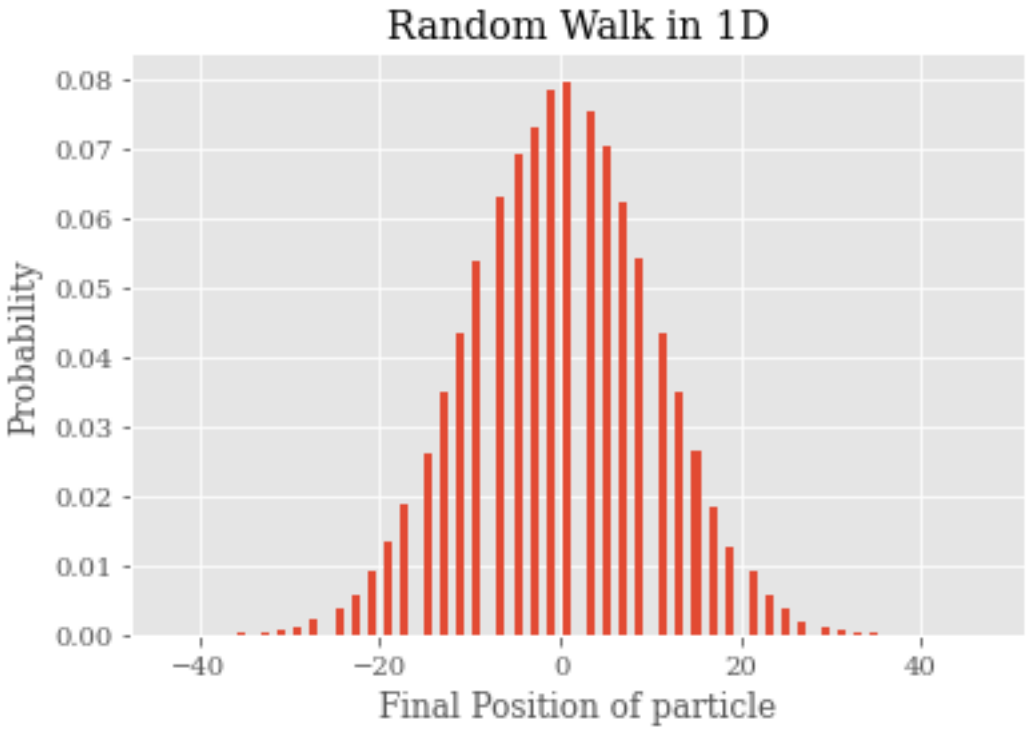
0.4 Number of random walks = 10^4

[68]:
```
random_walk(10000)
```



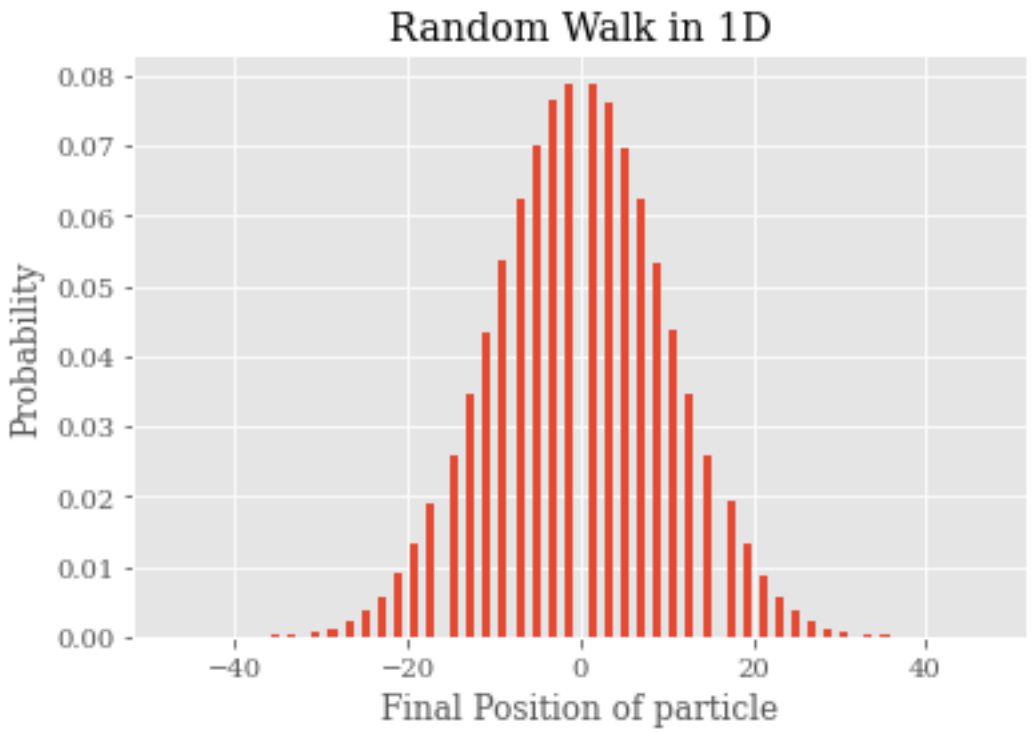
0.5 Number of random walks = 10^5

[69]:
```
random_walk(100000)
```



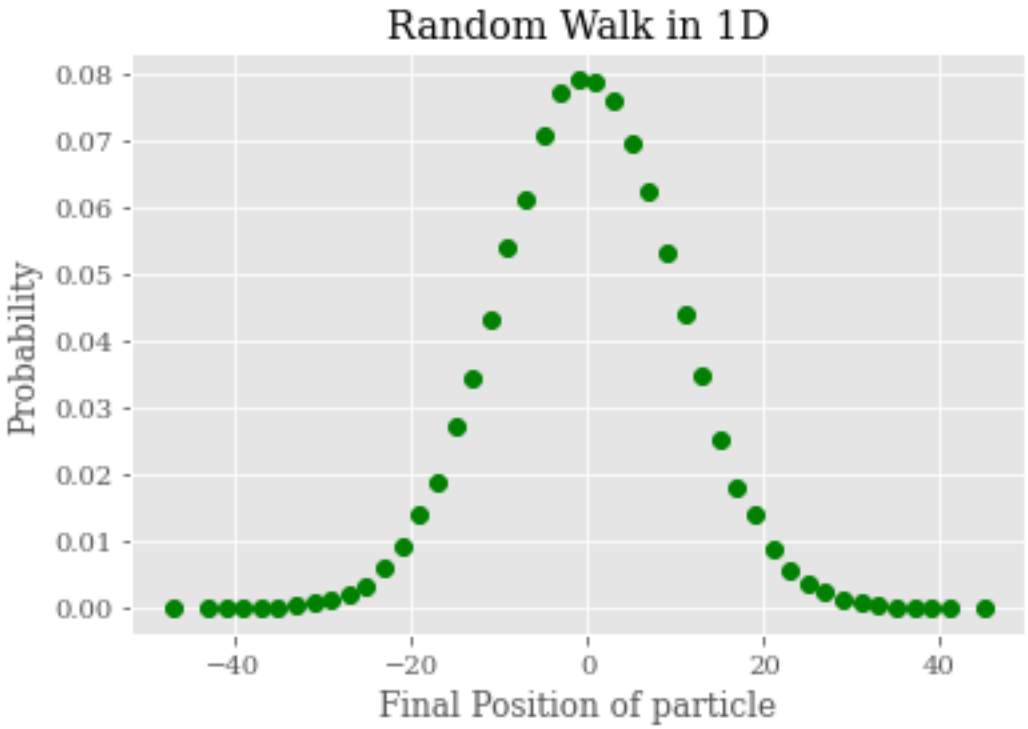
0.6 Number of random walks = 10^6

[70]:
```
random_walk(1e6)
```



[78]:
```
N = int(1e5) # no.of random walks
steps = 101 # no. of steps in each walk
x=[] # final displacement

step_dirn = [-1,1]

for i in range(N):
    u = random.choices(step_dirn, k = steps)
    x.append(sum(u))

weights = np.ones_like(x)/N

unique_elements, counts_elements = np.unique(x, return_counts=True)
counts_elements = counts_elements/N

plt.style.use('ggplot')
plt.rcParams["font.family"] = "serif"
plt.rcParams["mathtext.fontset"] = "dejavuserif"
#plt.hist(x, weights=weights,bins=100)
plt.plot(unique_elements,counts_elements,'go')
plt.title('Random Walk in 1D')
plt.ylabel('Probability')
plt.xlabel('Final Position of particle')
```

```
plt.savefig('Random_walk_hist.png',dpi=600)
plt.show()
```



[ ]: