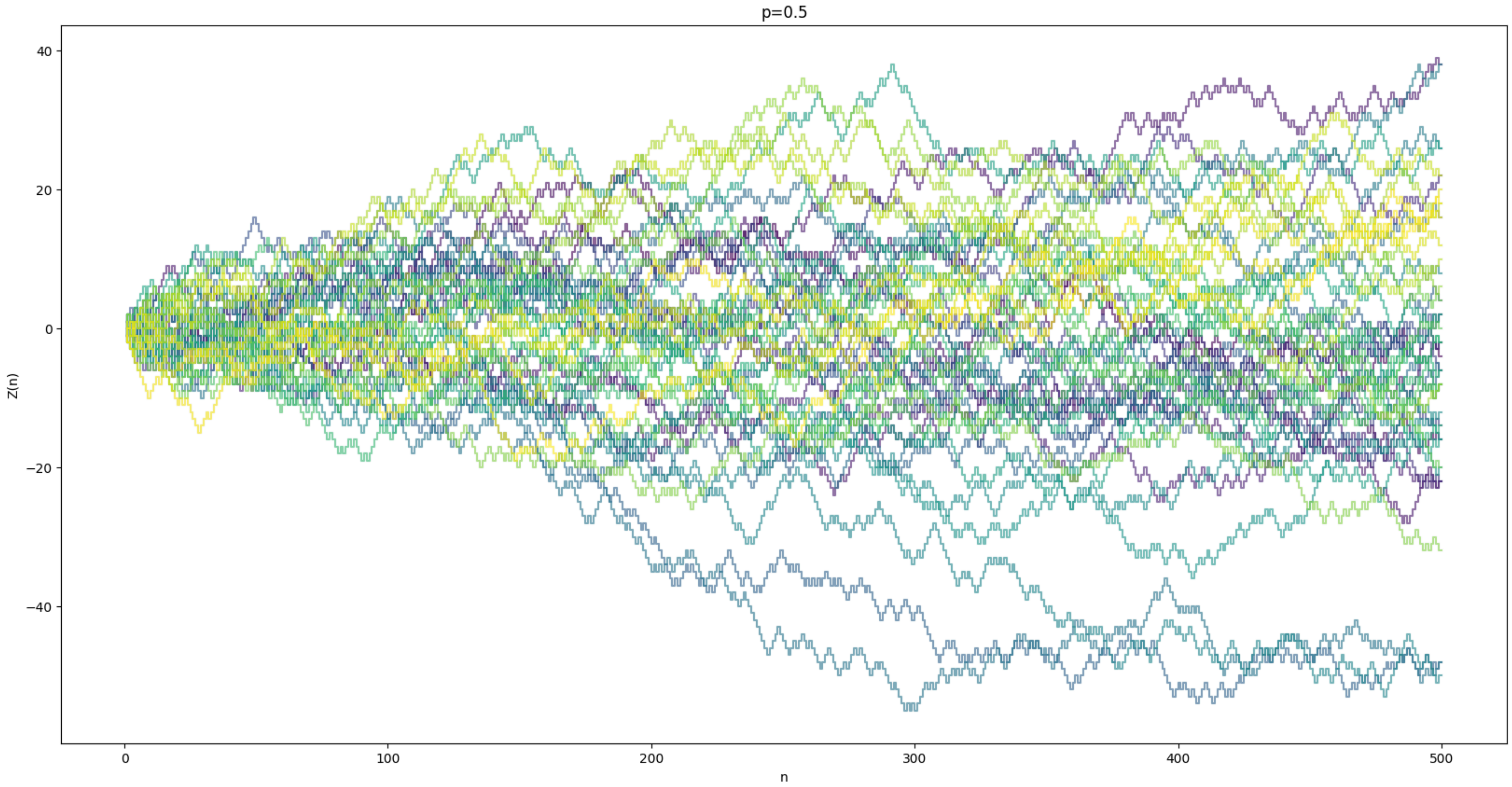
Colab Link: https://colab.research.google.com/drive/1iKhautsrHB5MBwEGzJYtr0NhAjYSoOFP?usp=sharing

In [ ]:
```
import random
import matplotlib.pyplot as plt
import numpy as np
import math
```

Question 1

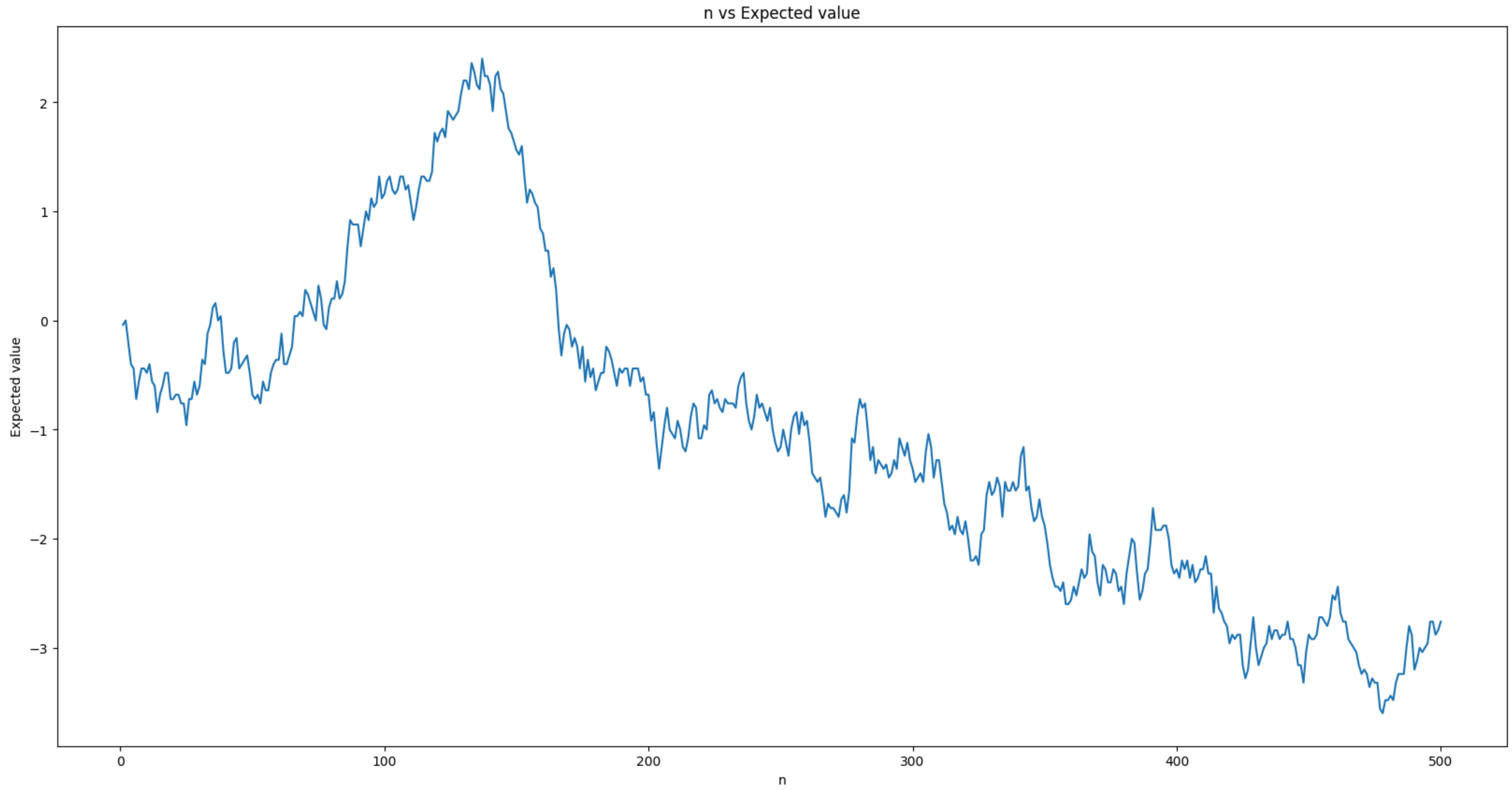
In [ ]:
```
#a)
colors = plt.cm.viridis(np.linspace(0, 1, 50))
plt.figure(figsize=(20, 10))

n = list(range(1,501))
z_matrix = []
for i in range(50):
    X = np.random.choice([-1, 1], size=500, p=[0.5, 0.5])
    z = np.cumsum(X)
    z_matrix.append(z)
    plt.step(n,z, color = colors[i],alpha=0.6)
plt.xlabel("n")
plt.ylabel("Z(n)")
plt.title("p=0.5")
plt.show()
```



In [ ]:
```
#b)
z_matrix = np.array(z_matrix)
z_matrix.shape
```

Out[ ]: (50, 500)

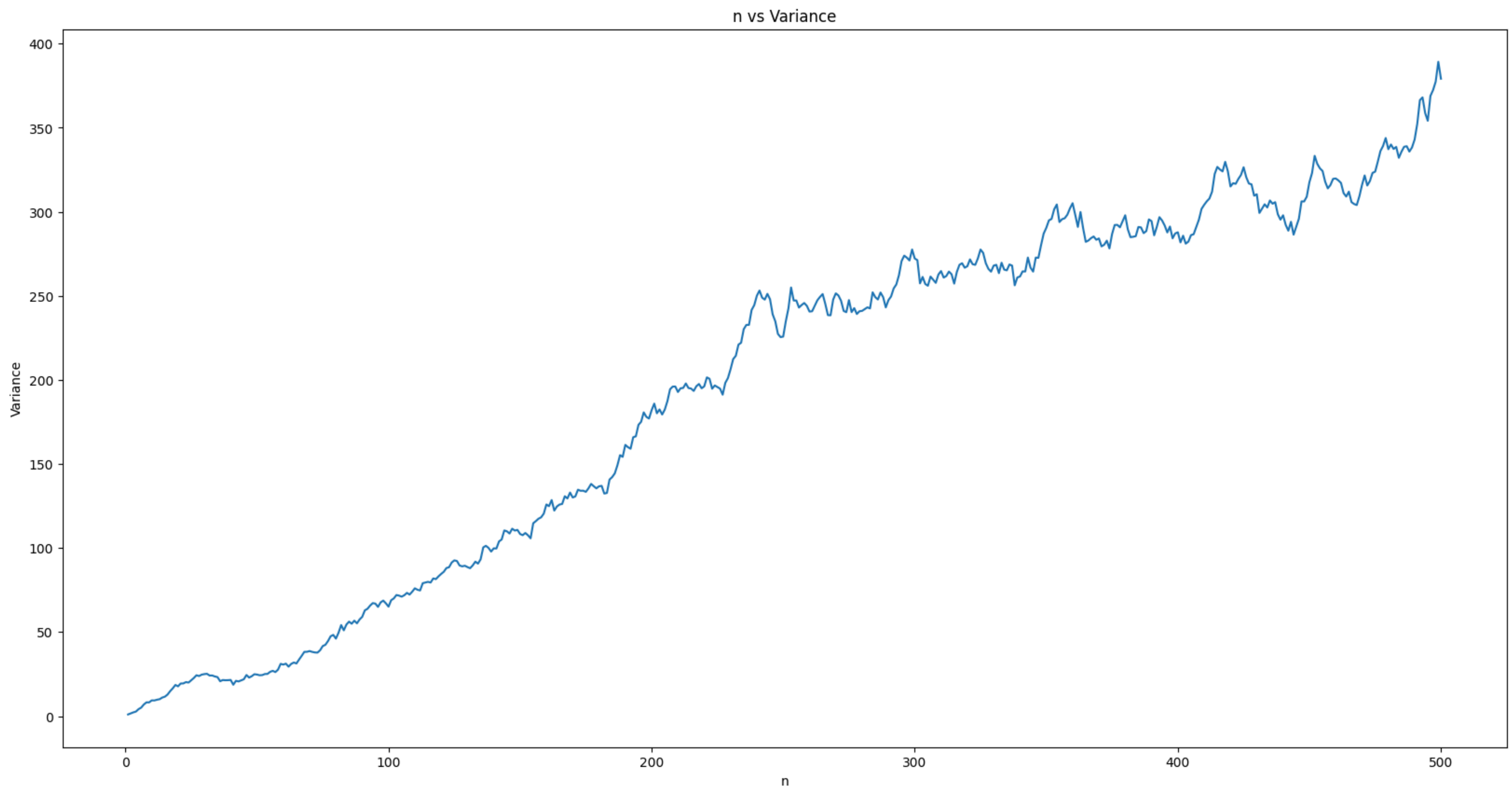
In [ ]:
```
mean_z = np.sum(z_matrix, axis = 0)
mean_z = mean_z/50
plt.figure(figsize=(20, 10))

plt.plot(n,mean_z)
plt.xlabel("n")
plt.ylabel("Expected value")
plt.title("n vs Expected value")
plt.show()
```



Explanation: The Expected value fluctuate around 0. As number of traces increase, we would expect to see fluctuation closer to 0; thus, a better approximation.

In [ ]:
```
#c)
mean_z = mean_z.reshape(1,500)
```

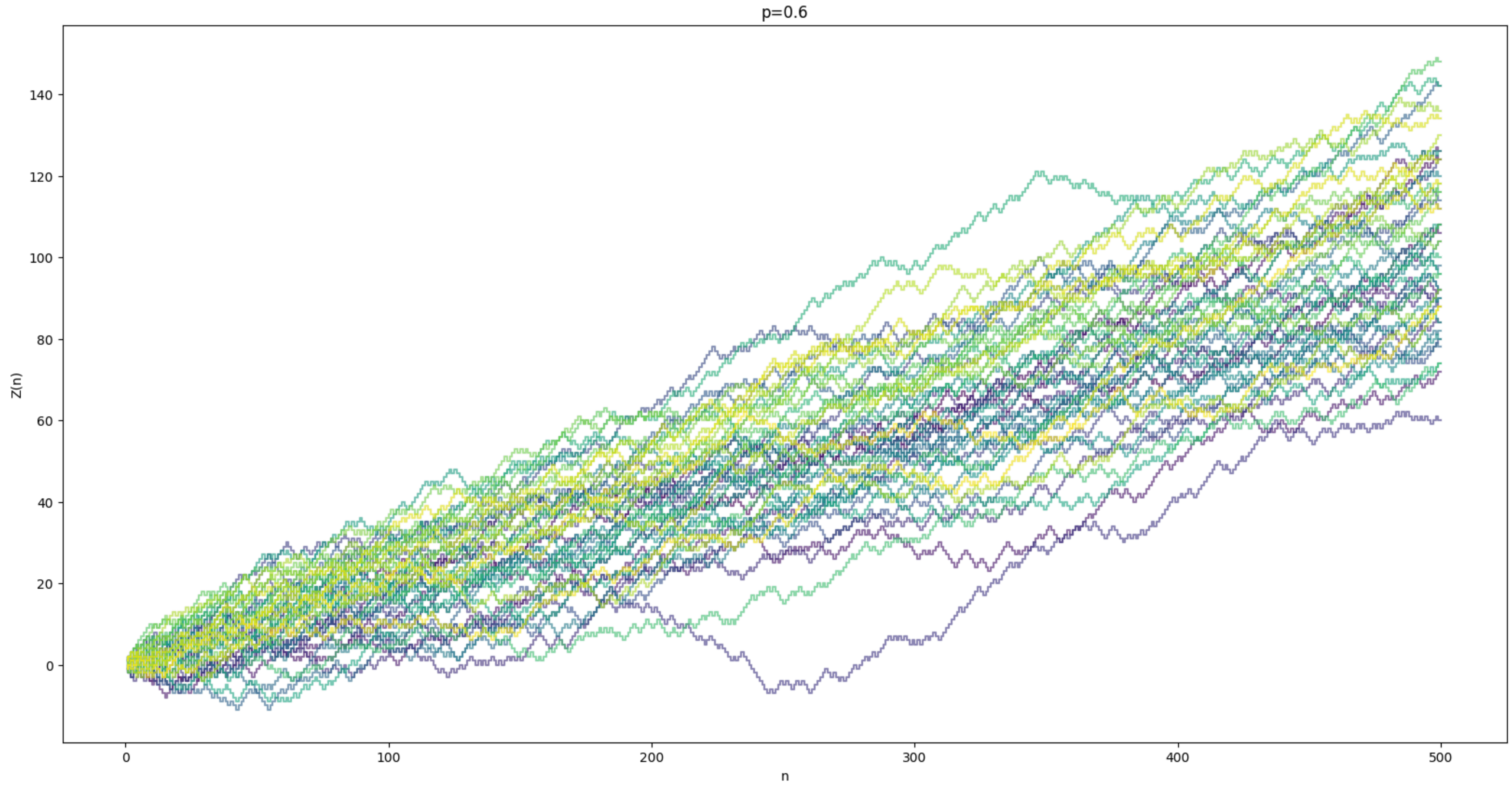
In [ ]:
```
var_z = (z_matrix-mean_z)**2
var_z = (np.sum(var_z, axis = 0))/50
plt.figure(figsize=(20, 10))

plt.plot(n,var_z)
plt.xlabel("n")
plt.ylabel("Variance")
plt.title("n vs Variance")
plt.show()
```



Explanation: Z(n) is not a stationary process. As n increases, the variance also increases; therefore, the distribution changes according to n.

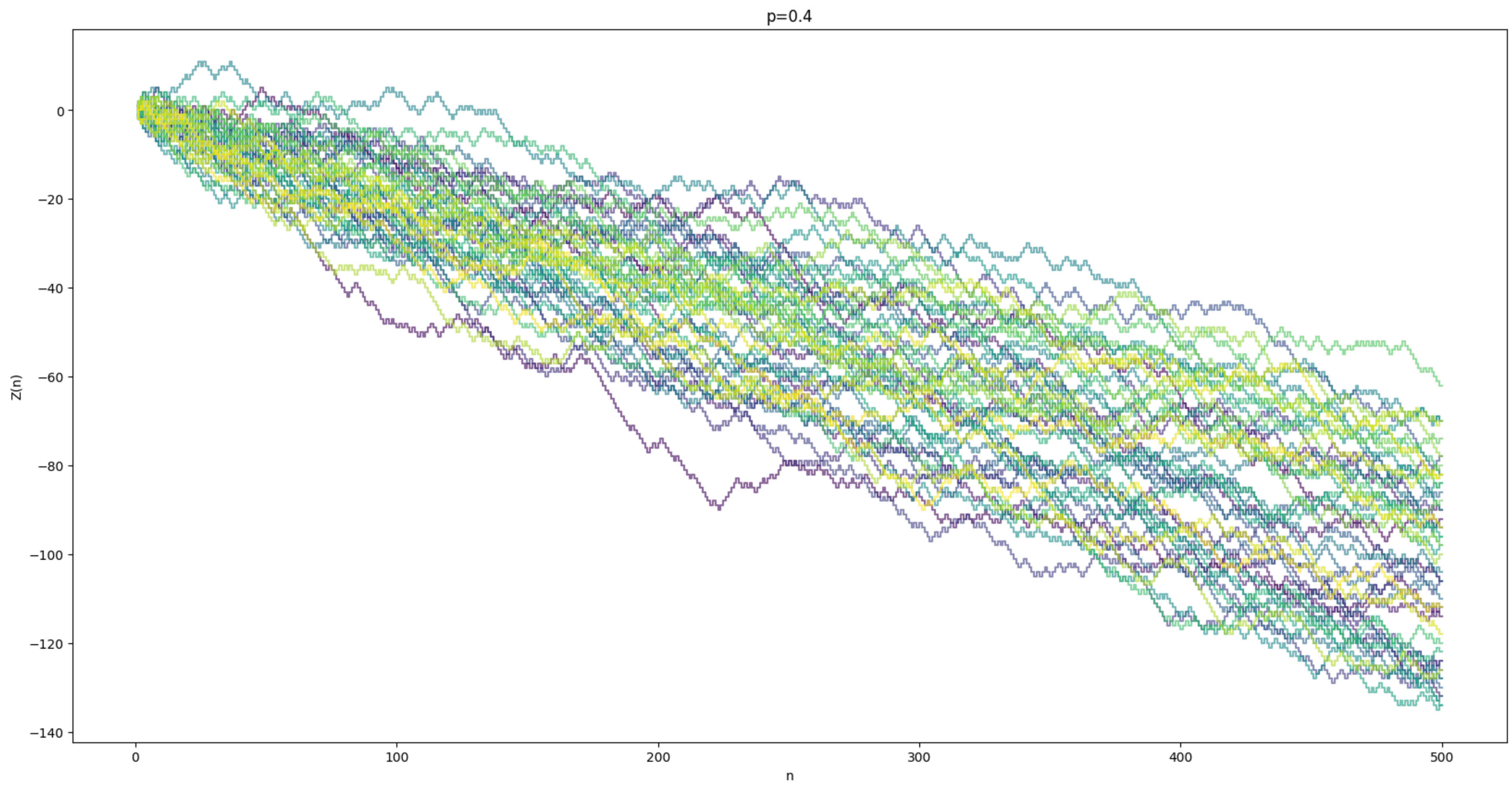
In [ ]:
```
#d)
def plot_z(p):
    colors = plt.cm.viridis(np.linspace(0, 1, 50))
    plt.figure(figsize=(20, 10))

    n = list(range(1,501))
    for i in range(50):
        X = np.random.choice([-1, 1], size=500, p=[1-p, p])
        z = np.cumsum(X)
```

```
        plt.step(n,z, color = colors[i],alpha=0.6)
    plt.xlabel("n")
    plt.ylabel("Z(n)")
    plt.title(f"p={p}")
    plt.show()
plot_z(0.6)
```



Explanation: We see a trend where Z(n) increases as n increases. This can be expected as the probability of +1 is more than probability of -1.

In [ ]:
```
#e)
plot_z(0.4)
```

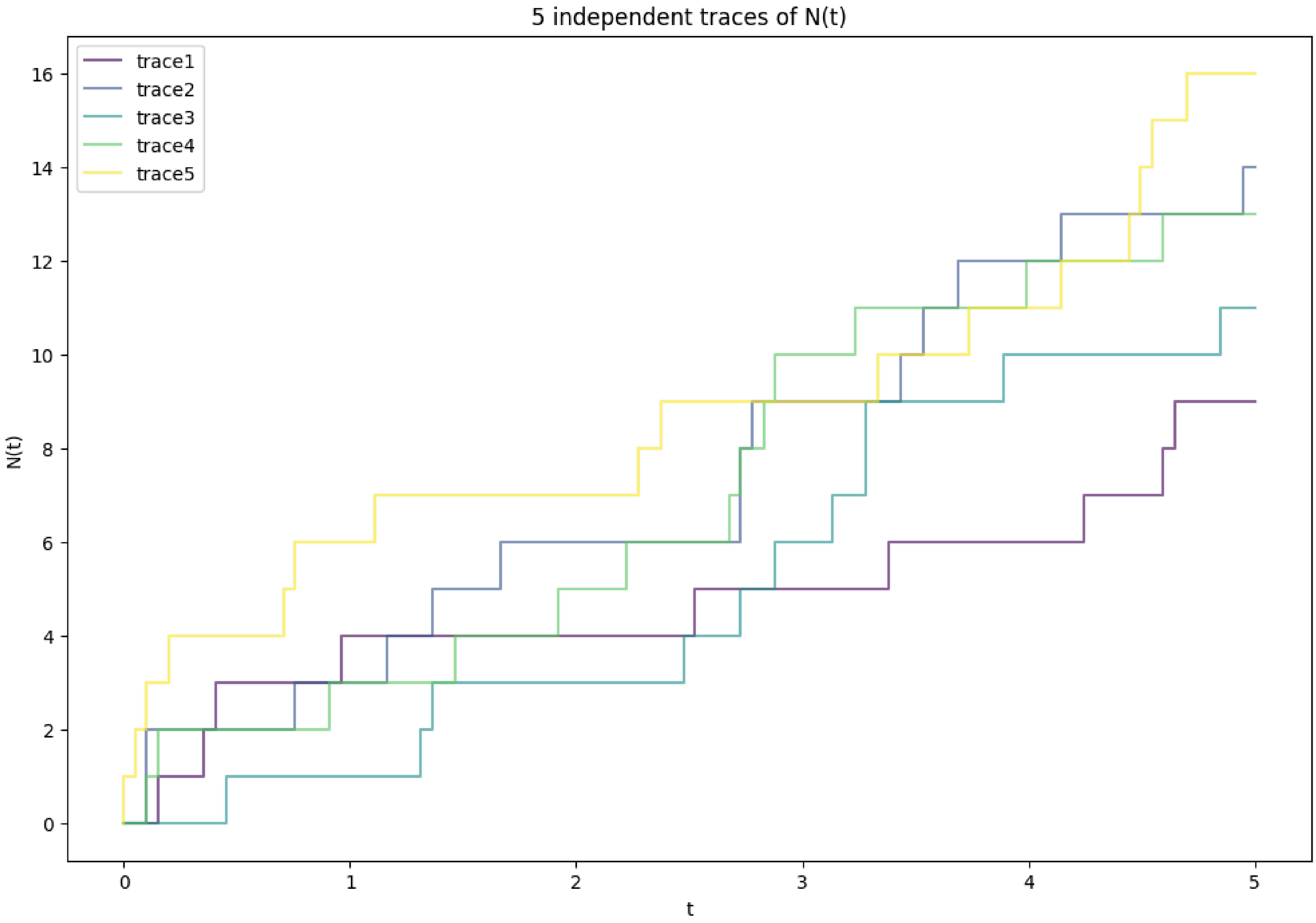


Explanation: We see a trend where Z(n) decreases as n increases. This can be expected as the probability of -1 is more than the probability of +1.

Question 2

In [ ]:
```
#a)
def N_t(sum_x, t):
    p = sum_x < t
    return np.sum(p)

plt.figure(figsize=(12, 8))
colors = plt.cm.viridis(np.linspace(0, 1, 5))
t_domain = np.linspace(0, 5, 100)
for i in range(5):
    X = np.random.exponential(scale=1/2, size=100)
    sum_x = np.cumsum(X)
```

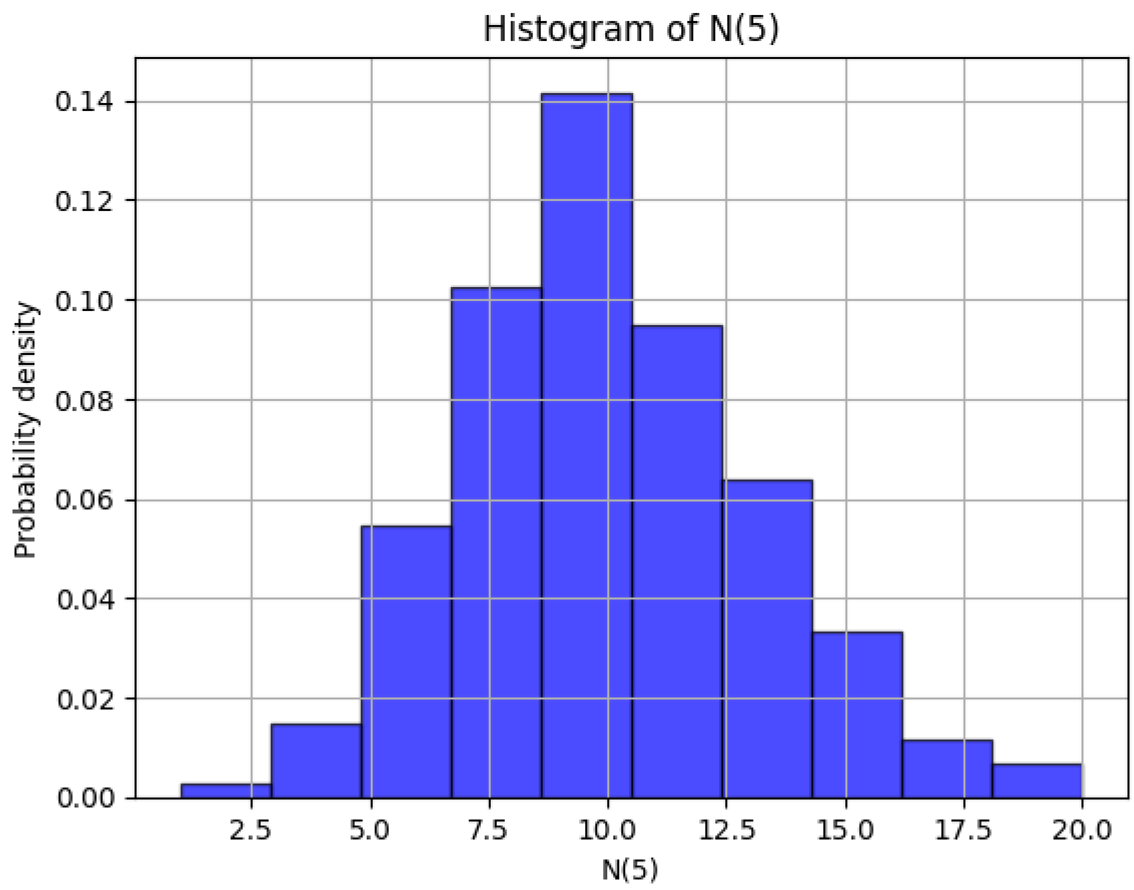
```
    N = [N_t(sum_x,t) for t in t_domain]
    plt.step(t_domain,N, color = colors[i],alpha=0.6,label=f"trace{i+1}")
plt.xlabel("t")
plt.ylabel("N(t)")
plt.title("5 independent traces of N(t)")
plt.legend()
plt.show()
```



In [ ]:
```
#b)
N = []
for i in range(1000):
    X = np.random.exponential(scale=1/2, size=100)
    sum_x = np.cumsum(X)
    N.append(N_t(sum_x, 5))
N[0:5]
```

Out[ ]: [15, 8, 12, 6, 15]

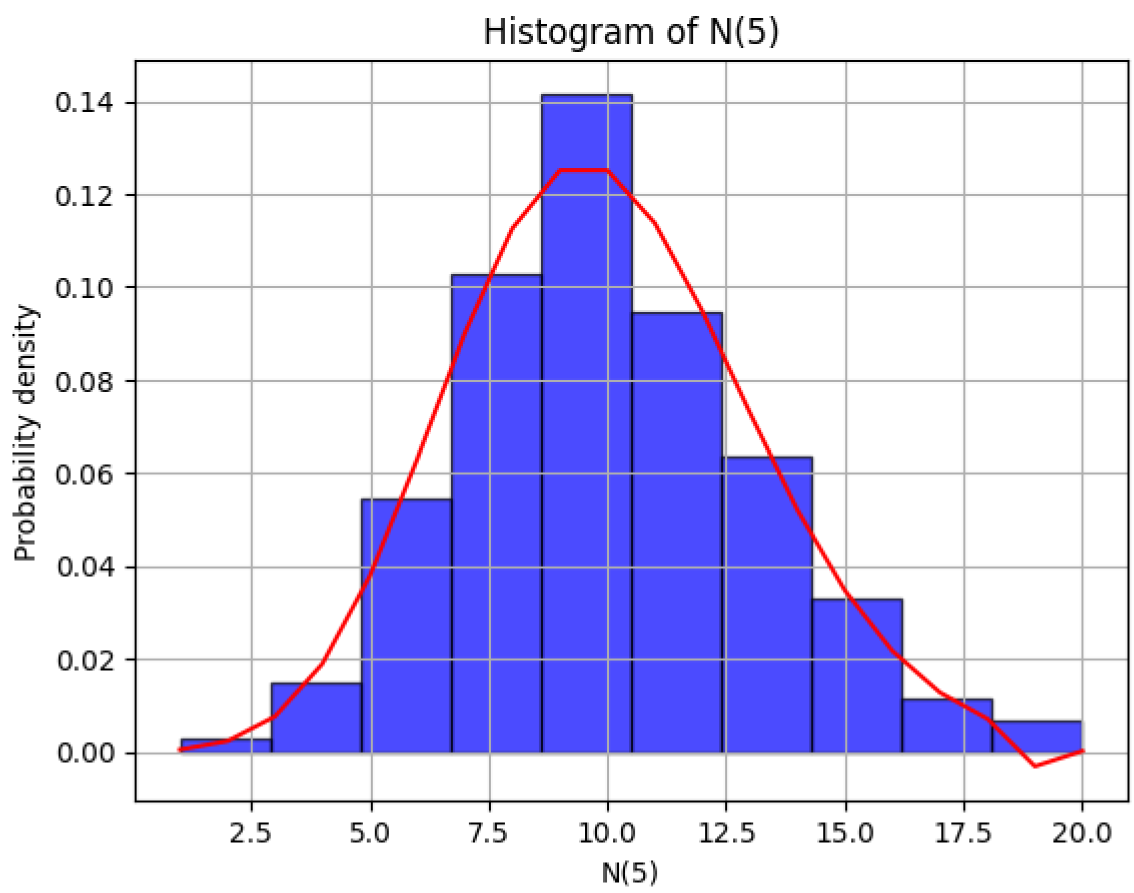
In [ ]:
```
plt.hist(N, bins=10, color='b', alpha=0.7, edgecolor='black', density=True)
plt.xlabel('N(5)')
plt.ylabel('Probability density')
plt.title('Histogram of N(5)')

plt.grid(True)

plt.show()
```



In [ ]:
```
k = np.arange(min(N), max(N)+1)
numerator = ((2*5)**k)*(math.e**(-2*5))
denominator = np.array([math.factorial(i) for i in k])
poisson = numerator/denominator
print(poisson)
```

```
[ 0.000454    0.00227     0.00756665  0.01891664  0.03783327  0.06305546
  0.09007923  0.11259903  0.12511004  0.12511004  0.1137364   0.09478033
  0.07290795  0.0520771   0.03471807  0.02169879  0.012764    0.00709111
 -0.00315246  0.00014493]
```

In [ ]:
```
plt.hist(N, bins=10, color='b', alpha=0.7, edgecolor='black', density=True)
plt.plot(k, poisson, color='r', label='Poisson PDF')

plt.xlabel('N(5)')
plt.ylabel('Probability density')
plt.title('Histogram of N(5)')

plt.grid(True)

plt.show()
```



Explanation: The distribution of N(5) closely represents a Poisson distribution with mean (2*5)

In [ ]:
```
#c)
mean = np.mean(N)
print(f"Estimated Expected Value: {mean}")
print("Theoretical Expected Value: 2*5 = 10")
```

```
Estimated Expected Value: 10.026
Theoretical Expected Value: 2*5 = 10
```

In [ ]:
```
#d)
var = (N-mean)**2
var = np.sum(var)/len(N)
print(f"Estimated Variance: {var}")
print("Theoretical Variance: 2*5 = 10")
```

```
Estimated Variance: 10.823324000000001
Theoretical Variance: 2*5 = 10
```