ECE 50024

Homework 6

Parth Sagar Hasabnis phasabni@purdue.edu

Exercise 1:

Exercise 1

Let $\circ = 0$ and $\bullet = 1$

a) $h_1(x_n) = 1 \qquad n = 1, 2, \ldots, 8$

$h_2(x_n) = 0 \qquad n = 1, 2, \ldots, 8$

$h_1(\cdot)$ matches with $3/5$ samples
$h_2(\cdot)$ matches with $2/5$ samples

Hence the learning algorithm will pick h_1.

$g = [1, 1, 1, 1, 1, 1, 1, 1]$

g matches with:

1) 3 out-samples once (f_8)
2) 2 out-samples thrice (f_4, f_6, f_7)
3) 1 out-sample thrice (f_2, f_3, f_5)
4) 0 out-samples once (f_1)

b) In this case, the learning algorithm will pick h_2

$$g = [0, 0, 0, 0, 0, 0, 0, 0]$$

g matches with

1) 3 out samples once (f_1)
2) 2 out samples thrice (f_2, f_3, f_5)
3) 1 out sample thrice (f_4, f_6, f_7)
4) 0 out samples once (f_8)

c) $$g = [0, 1, 1, 0, 1, 0, 0, 1]$$

g matches with

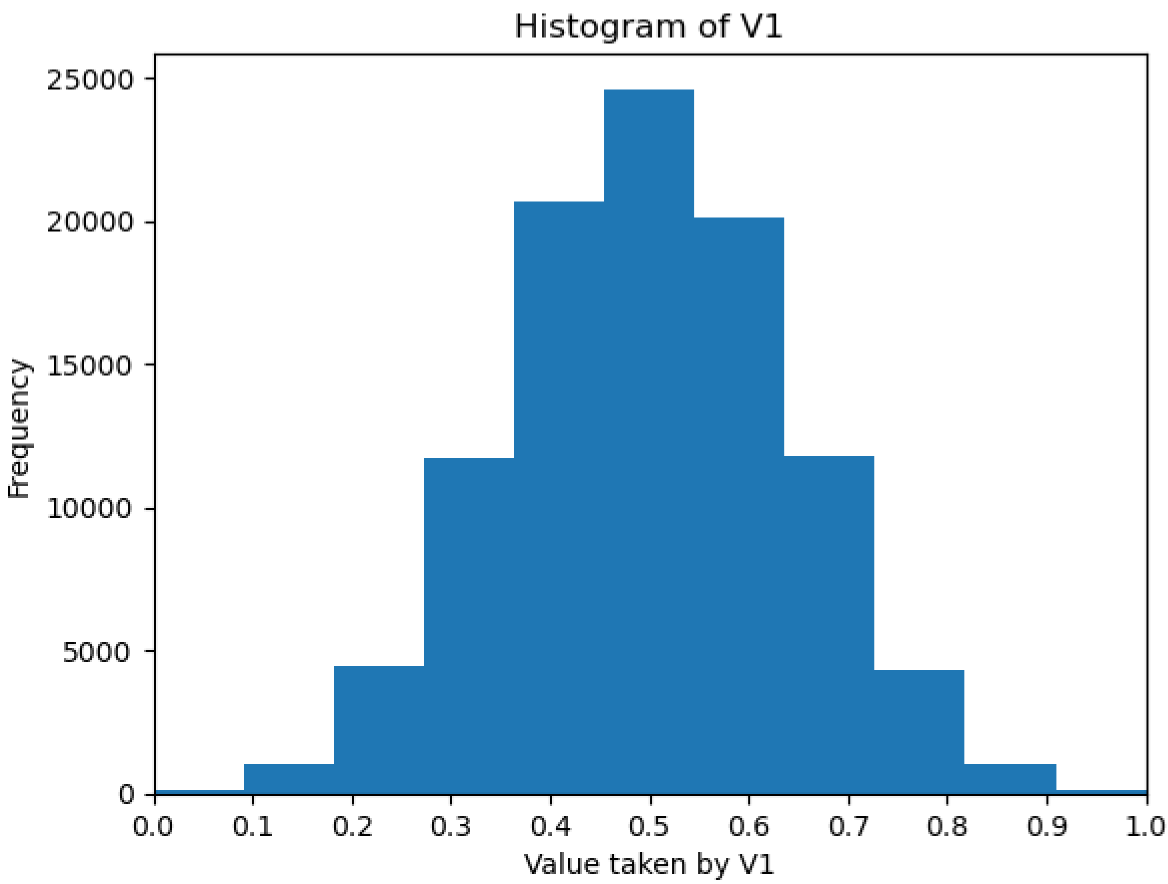
1) 3 out samples once (f_2)
2) 2 out samples thrice (f_1, f_4, f_6)
3) 1 out sample thrice (f_3, f_5, f_8)
4) 0 out samples once (f_7)

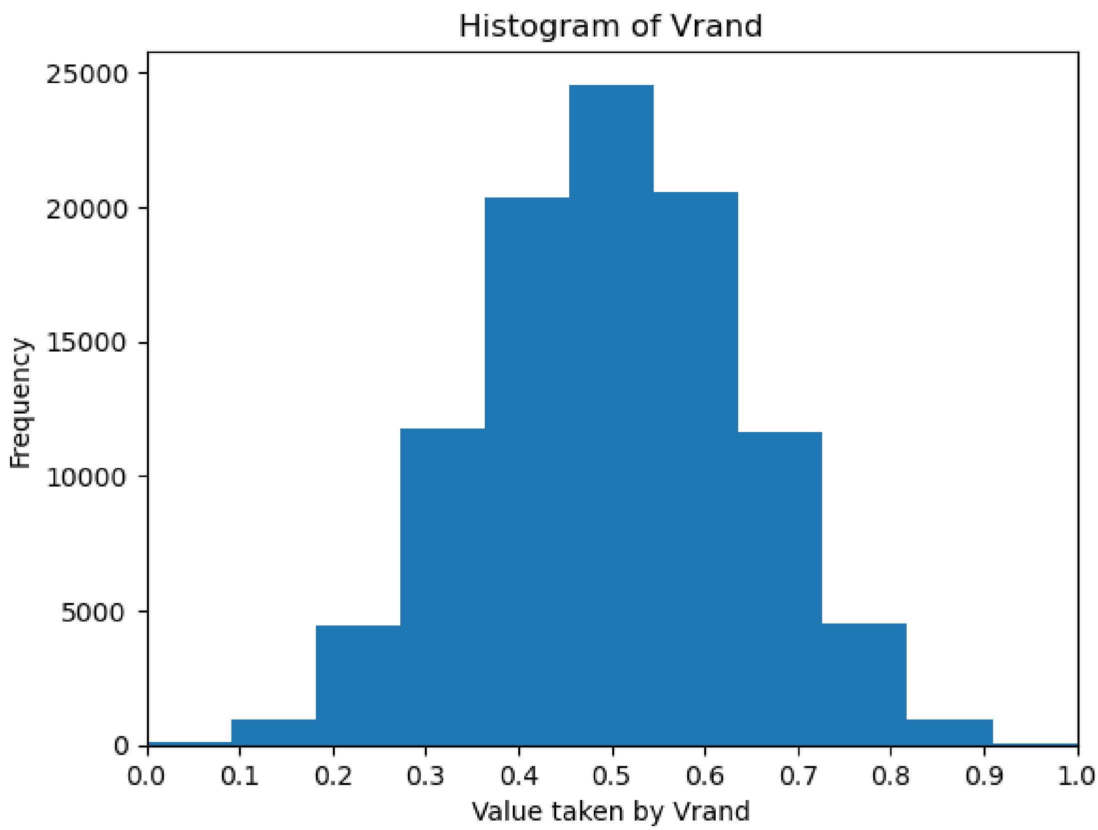
Exercise 2:

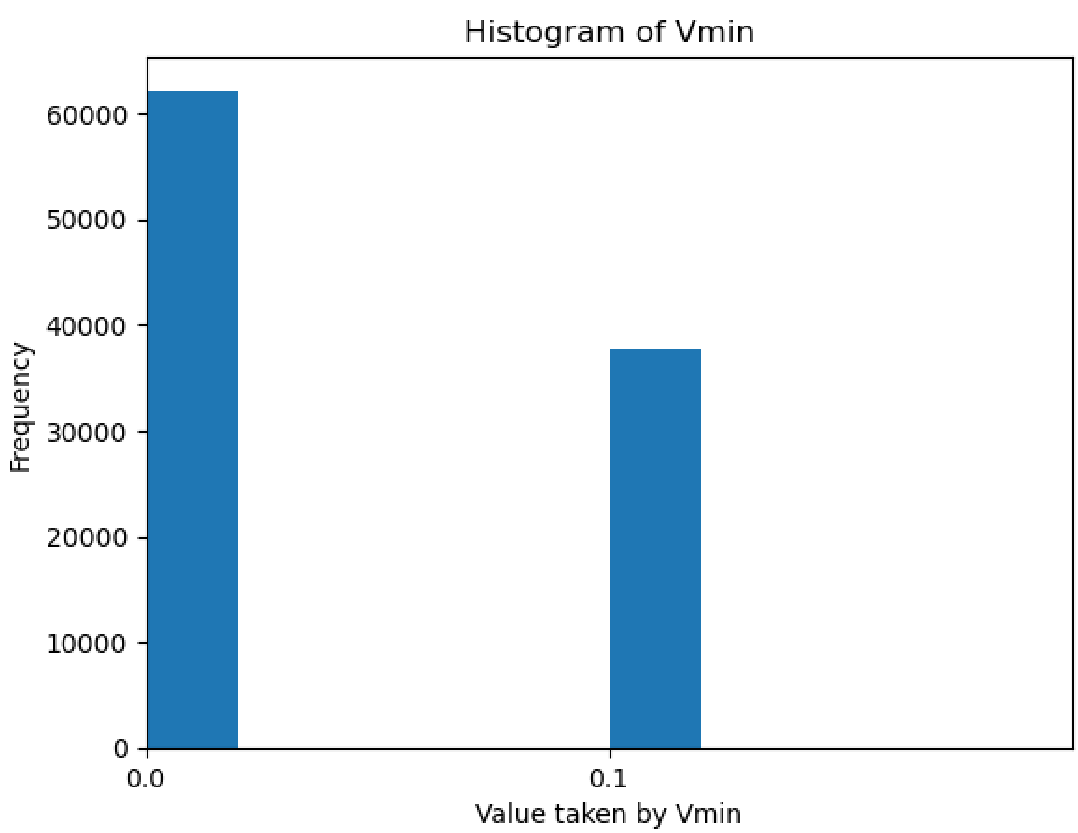
a) As all the three coins we pick are fair coins, the probability that we get a head on each of them is same and is equal to 0.5.

$$\mu_1 = 0.5$$
$$\mu_{rand} = 0.5$$
$$\mu_{min} = 0.5$$

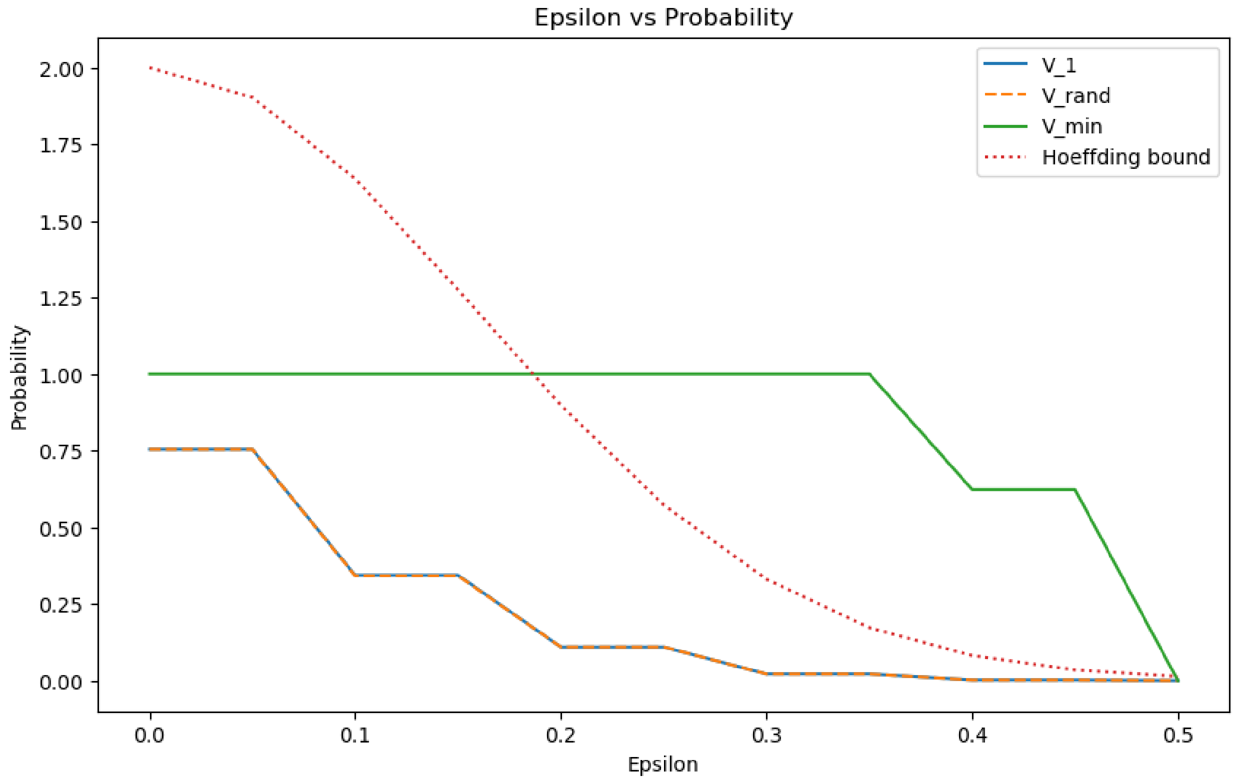
b) Histograms of the random variables




Histogram of Vrand
Frequency
25000
20000
15000
10000
5000
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Value taken by Vrand


Histogram of Vmin
Frequency
60000
50000
40000
30000
20000
10000
0
0.0
0.1
Value taken by Vmin

c) Hoeffding's inequality for the 3 coins



Note: The curves for V_1 and V_{rand} coincide.

d) The coins $coin_1$and $coin_{rand}$ follow the Hoeffding bound, while $coin_{min}$ does not. This is because the coins $coin_1$and $coin_{rand}$ are selected before we look at the data, while $coin_{min}$ is selected after we have the data. Hoeffding inequality is valid only if we apply it before we look at the data.

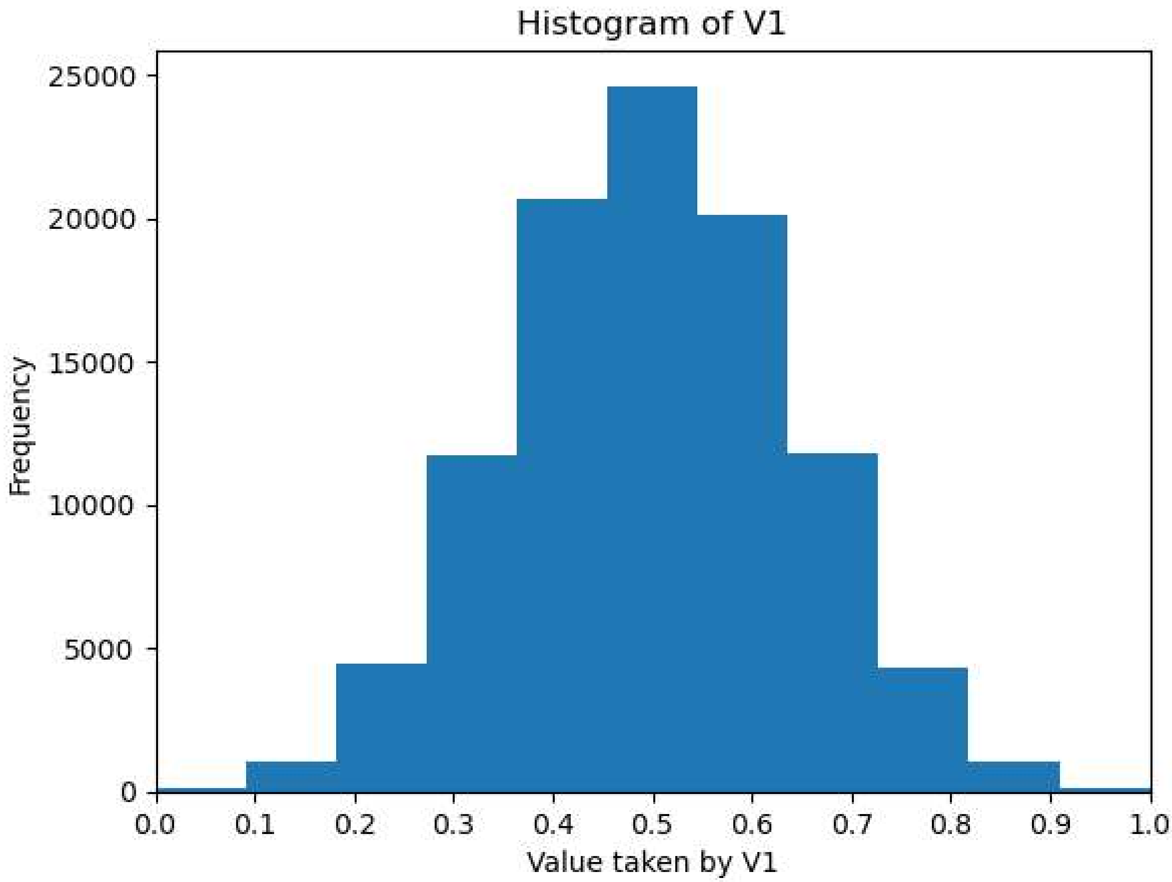
APPENDIX

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

In [ ]:
```
V_1 = []
V_min = []
V_rand = []

M = 1000
trials = 100000
N = 10
p=0.5

heads_array = []

for trial in range(trials):
    for coin in range(M):
        n_heads = np.random.binomial(N, p)
        heads_array.append(n_heads)
    V_1.append(heads_array[0])
    V_min.append(np.min(heads_array))
    idx = np.random.randint(0,M)
    V_rand.append(heads_array[idx])
    heads_array = []
```

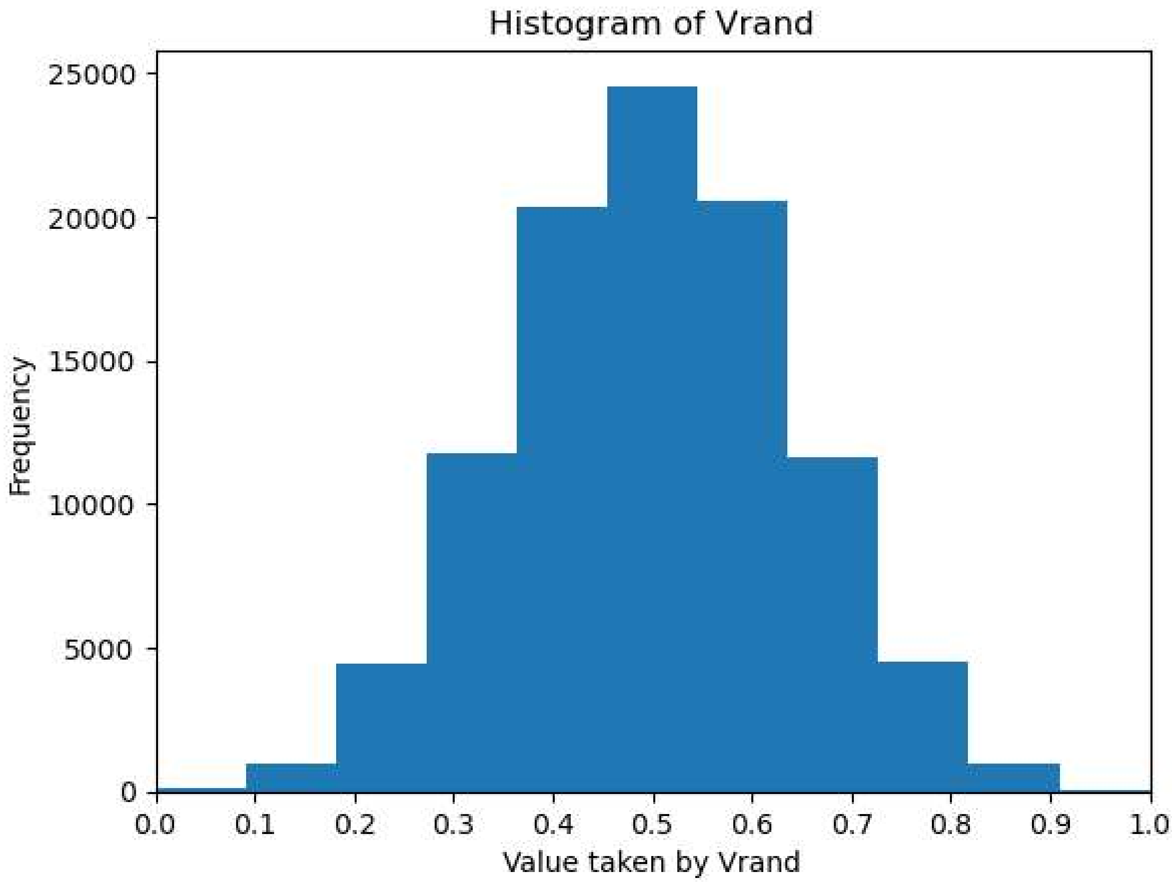
In [ ]:
```
V1 = np.array(V_1)/N
plt.hist(V1, bins=11)
plt.title("Histogram of V1")
plt.xlabel("Value taken by V1")
plt.ylabel("Frequency")
plt.xticks(np.arange(0,12,1)/10)
plt.xlim([0,1])
```

Out[ ]:
```
(0.0, 1.0)
```



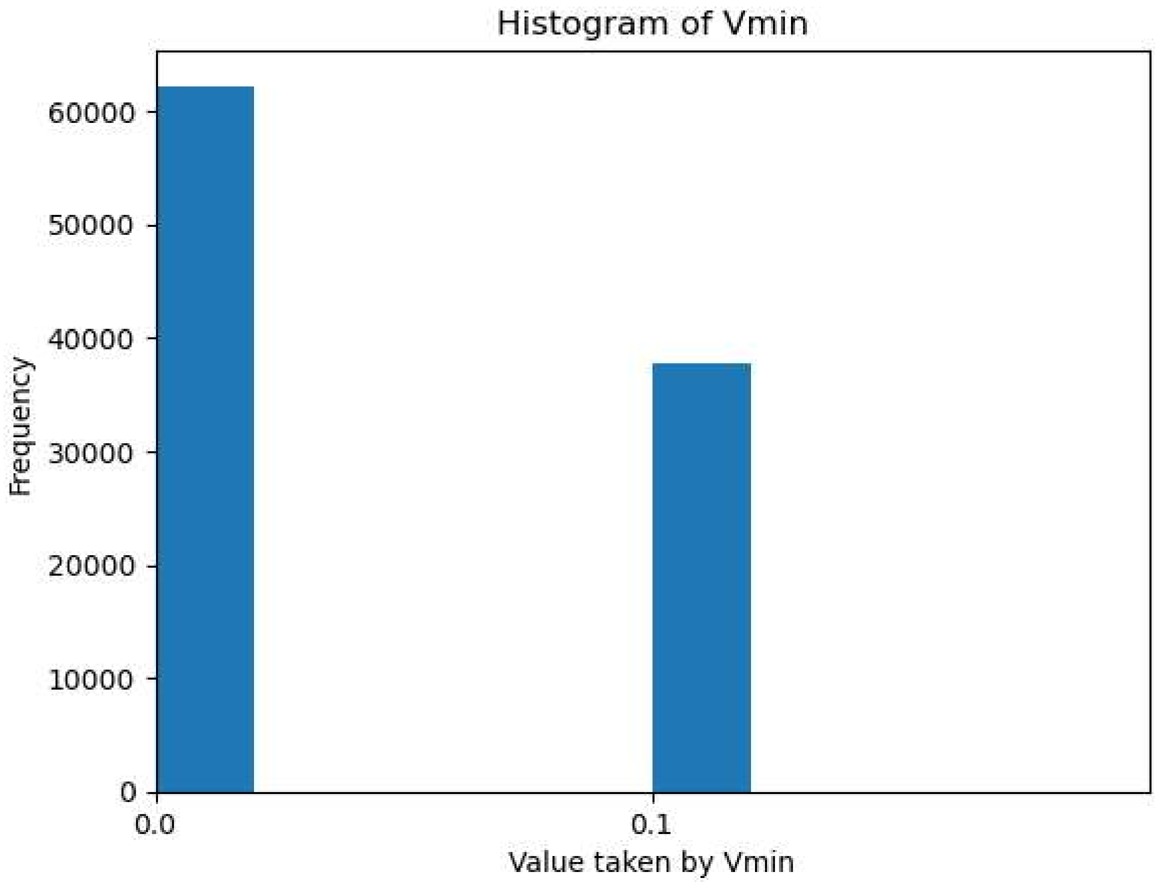
In [ ]:
```
Vrand = np.array(V_rand)/N
plt.hist(Vrand, bins=11)
plt.title("Histogram of Vrand")
plt.xlabel("Value taken by Vrand")
plt.ylabel("Frequency")
plt.xticks(np.arange(0,12,1)/10)
plt.xlim([0,1])
```

Out[ ]:
```
(0.0, 1.0)
```



In [ ]:
```
Vmin = np.array(V_min)/N
plt.hist(Vmin)
plt.title("Histogram of Vmin")
plt.xlabel("Value taken by Vmin")
plt.ylabel("Frequency")
plt.xticks(np.arange(0,2,1)/10)
plt.xlim([0,0.2])
```

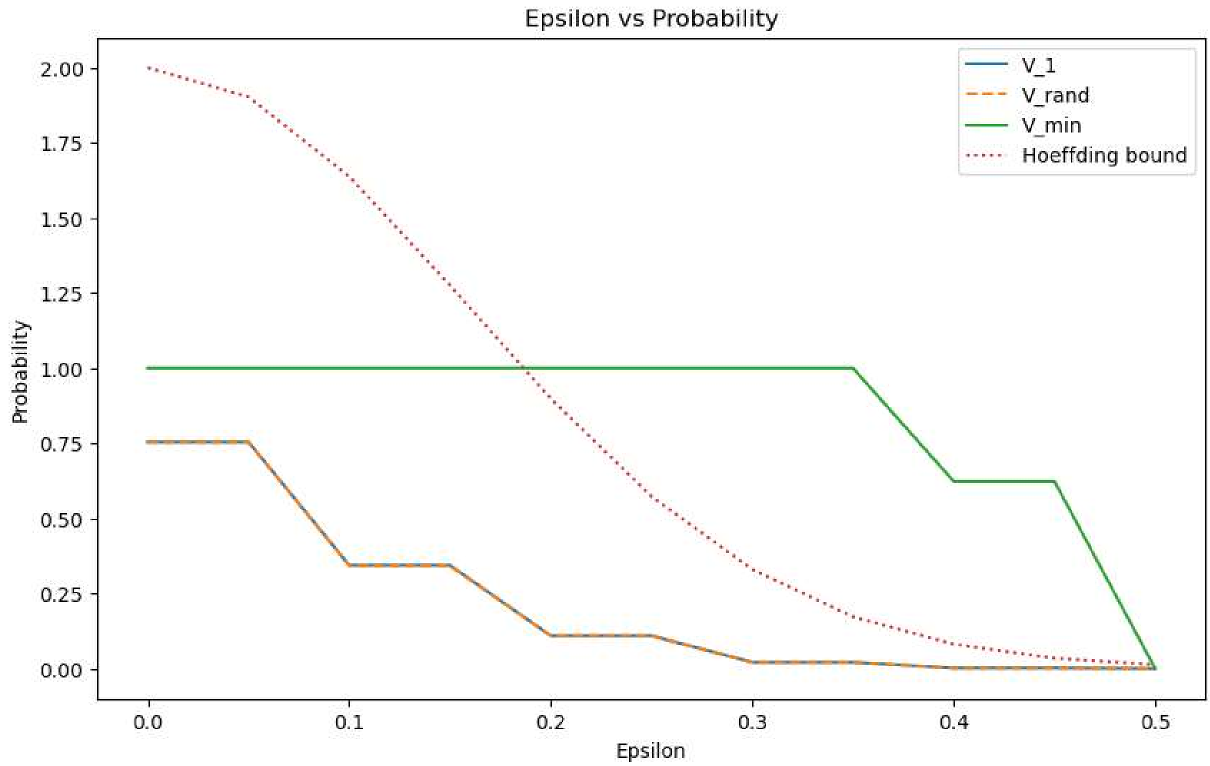
Out[ ]:
```
(0.0, 0.2)
```



In [ ]:
```
mu_1 = 0.5
mu_rand = 0.5
mu_min = 0.5
V = np.array([V1, Vrand, Vmin])
mu = np.array([0.5, 0.5, 0.5])

epilson = np.linspace(0,0.5,11)
Prob = np.zeros((len(V)+1, len(epilson)))
for e in range(len(epilson)):
    for idx in range(len(V)):
        count = np.count_nonzero((np.abs(V[idx] - mu[idx]))>epilson[e])
        Prob[idx,e] = count/trials
    Prob[idx+1,e] = 2*np.exp(-2*(epilson[e]**2)*N)

Prob = Prob
```

In [ ]:
```
marker = ["solid", "dashed", "solid", "dotted"]
plt.figure(figsize=(10,6))
for i, vector in enumerate(Prob):
    plt.plot(epilson, vector, linestyle=marker[i])
plt.legend(["V_1", "V_rand", "V_min", "Hoeffding bound"])
plt.title("Epsilon vs Probability")
plt.xlabel("Epsilon")
plt.ylabel("Probability")
```

Out[ ]:
```
Text(0, 0.5, 'Probability')
```


Epsilon vs Probability
V_1
V_rand
V_min
Hoeffding bound
Probability
Epsilon
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.1
0.2
0.3
0.4
0.5