For all of these problems, if you need the prior probability of a genotype in the population, you can assume the following: $P(AA) = 0.95^2$, $P(BB) = 0.05^2$, and $P(AB)$ is the remainder.

Before generating random numbers, make sure to set a random seed so your results will be reproducible.

Problem 1

From biallelic models to triallelic models

In the "SNP or not SNP" lecture, we limited ourselves to biallelic variation. Fortunately, most single nucleotide variation is biallelic (estimates of multiallelic variation are 3% of SNPs). Let's assume that at this particular position in the genome, we can have 3 possibilities: $\{G, A, C\}$. This isn't crazy — sometimes for unstable mutations alleles can drift and then create more stable variation (e.g. a G can turn into a C in some finite number of generations). We are going to restrict ourselves to the case where only two chromosomes exist.

(a) How many possible genotypes are there? (i.e. GG, GA, ..., CC).

(b) If we assume an error can result in any base (e.g. a G can turn into a T), assume I observe a T in my data. How does that translate into the probabilistic model? Note: I'm not looking for a complicated answer. Simply describing how it might change the probability of observing in error is sufficient.

(c) Write the probability of observing read i for homozygous genotypes (i.e. GG, AA, CC). You can follow the example from class, but don't forget there are more possibilities than the example in class.

(d) Write the probability of observing read i given genotype GA. Remember that we have only two chromosomes but three possible alleles.

(e) Write down the probability of the remaining genotypes. If they reduce into functions of the others, feel free to be lazy and write them in terms of other probabilities.

Problem 2

Data analysis + the bootstrap

Consider the biallelic model again.

(a) Refer to the slides from `2_errors_and_snps.pdf`. If you assume $P(C_i = A) = P(C_i = B)$, does this reduce any of the likelihoods? For the remainder of this problem, please assume $P(C_i = A) = P(C_i = B)$ when there is than one allele in the truth.

(b) `reads.tsv` is some data in the following format:

observation	$P(E_i = 1)$	**indicator if actually an error**
A	0.02	FALSE
G	0.01	TRUE
...	...	...

This is a simulation and you don't need the final column; it is there for your personal enjoyment. You might be able to do something with it, but you don't need it to solve the problem. *Please don't use it to solve the problem.*

Write some code to estimate the posterior probability of the three possible genotypes given the data.

(c) Randomly sample 5 observations with replacement and re-estimate the posterior probability of each genotype. What are the results?

(d) Repeat (c) 1,000 times. That means you will have 1,000 estimates for each of your posterior probabilities, each using 5 observations. This procedure is a variation of the *bootstrap*. Make a histogram for each of the posterior probabilities. Please be mindful of the number of bins and the appearance of your histogram. No one likes an ugly histogram.

(e) Repeat (d), but this time instead of taking 5 observations, take 50. Again, make three histograms.

(f) How do the results from (d) and (e) compare? Feel free to take summary statistics like the mean and standard deviation from those resampled results.

(g) Implicitly, there are assumptions about the base caller, the prior probabilities, etc. What are these assumptions and how might they affect the results? An example, what if the base caller probability estimates were way off?

1a) $P(3, 2)$ = **6 possible genotypes**

- b) If you observe a T, which is not an expected base, you will get uncertainty w/ your genotype data. This will affect the likelihood of each genotype given your observed data. Observing T suggests an error in sequencing or recording the data. You will have to account for this error by modifying the likelihood of observing each of the expected alleles so that it includes the probability of observing T due to an error. Getting expected alleles of {G, A, C}, but appearing as T would now have to be accounted for by the error probability ϵ_i in the probabilistic model.

- c) $P(O_i = G \mid G = GG) = \underbrace{P(O_i = G \mid E_i = 0, G = GG)}_{1} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} + \underbrace{P(O_i = G \mid E_i = 1, G = GG)}_{0} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} = 1 - \epsilon_i$

  $P(O_i = A \mid G = GG) = \underbrace{P(O_i = A \mid E_i = 0, G = GG)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon} + \underbrace{P(O_i = A \mid E_i = 1, G = GG)}_{1} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} = \frac{1}{2}\epsilon_i$

  $P(O_i = C \mid G = GG) = \underbrace{P(O_i = C \mid E_i = 0, G = GG)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon} + \underbrace{P(O_i = C \mid E_i = 1, G = GG)}_{1} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} = \frac{1}{2}\epsilon_i$

  repeat for AA and CC

$P(O_i = G \mid G = GG) = 1 - \epsilon_i$ no error	$P(O_i = G \mid G = AA) = \epsilon_i/2$	$P(O_i = G \mid G = CC) = \epsilon_i/2$
$P(O_i = A \mid G = GG) = \epsilon_i/2$	$P(O_i = A \mid G = AA) = 1 - \epsilon_i$	$P(O_i = A \mid G = CC) = \epsilon_i/2$
$P(O_i = C \mid G = GG) = \epsilon_i/2$	$P(O_i = C \mid G = AA) = \epsilon_i/2$	$P(O_i = C \mid G = CC) = 1 - \epsilon_i$
$P(O_i \neq G \mid G = GG) = \epsilon_i$	$P(O_i \neq A \mid G = AA) = \epsilon_i$	$P(O_i \neq C \mid G = CC) = \epsilon_i$

* if there's an error, O_i has $\frac{1}{2}$ chance of being A or C (given)

- d) $P(O_i = G \mid G = GA)$ = same reasoning as $P(O_i = A \mid G = GA)$

$$= \underbrace{P(O_i = G \mid E_i = 0, C_i = G, G = GA)}_{1} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} \cdot \underbrace{P(C_i = G \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = G \mid E_i = 0, C_i = A, G = GA)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} \cdot \underbrace{P(C_i = A \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = G \mid E_i = 1, C_i = G, G = GA)}_{0} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = G \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = G \mid E_i = 1, C_i = A, G = GA)}_{1} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = A \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = G \mid E_i = 0, C_i = C, G = GA)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} \cdot \underbrace{P(C_i = C \mid G = GA)}_{0}$$
$$+ \underbrace{P(O_i = G \mid E_i = 1, C_i = C, G = GA)}_{0} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = C \mid G = GA)}_{0}$$

$P(O_i = G \mid GA) = (\frac{1}{2})(1 - \epsilon_i) + (\frac{1}{2}\epsilon_i)(\frac{1}{2})$

$\boxed{P(O_i = G \mid GA) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$

$P(O_i = A \mid GA) = P(O_i = G \mid GA)$

$\boxed{P(O_i = A \mid GA) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$ Same as before b/c we assume $P(C_i = A) = P(C_i = G)$

$$P(O_i = C \mid G = GA) = \underbrace{P(O_i = C \mid E_i = 0, C_i = G, G = GA)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} \cdot \underbrace{P(C_i = G \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = C \mid E_i = 0, C_i = A, G = GA)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon_i} \cdot \underbrace{P(C_i = A \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = C \mid E_i = 1, C_i = G, G = GA)}_{1} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = G \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = C \mid E_i = 1, C_i = A, G = GA)}_{1} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = A \mid G = GA)}_{\frac{1}{2}}$$
$$+ \underbrace{P(O_i = C \mid E_i = 0, C_i = C, G = GA)}_{0} \cdot \underbrace{P(E_i = 0)}_{1-\epsilon} \cdot \underbrace{P(C_i = C \mid G = GA)}_{0}$$
$$+ \underbrace{P(O_i = G \mid E_i = 1, C_i = C, G = GA)}_{0} \cdot \underbrace{P(E_i = 1)}_{\frac{1}{2}\epsilon_i} \cdot \underbrace{P(C_i = C \mid G = GA)}_{0}$$

$P(O_i = C \mid G = GA) = (1)(\frac{1}{2}\epsilon_i)(\frac{1}{2}) + (1)(\frac{1}{2}\epsilon_i)(\frac{1}{2})$

$= \frac{1}{4}\epsilon_i + \frac{1}{4}\epsilon_i$

$\boxed{P(O_i = C \mid G = GA) = \frac{1}{2}\epsilon_i}$

e) $P(O_i = G \mid G = CA) = P(O_i = C \mid G = GA)$

$\boxed{P(O_i = G \mid G = CA) = \frac{1}{2}\epsilon_i}$

$P(O_i = C \mid G = CA) = P(O_i = G \mid G = GA)$

$\boxed{P(O_i = C \mid G = CA) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$

$P(O_i = A \mid G = CA) = P(O_i = A \mid G = GA)$

$\boxed{P(O_i = A \mid G = CA) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$

$G = CA$

$P(O_i = G \mid G = GC) = P(O_i = G \mid G = GA)$

$\boxed{P(O_i = G \mid G = GC) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$

$P(O_i = C \mid G = GC) = P(O_i = A \mid G = GA)$

$\boxed{P(O_i = C \mid G = GC) = \frac{1}{2} - \frac{1}{4}\epsilon_i}$

$P(O_i = A \mid G = GC) = P(O_i = C \mid G = GA)$

$\boxed{P(O_i = A \mid G = GC) = \frac{1}{2}\epsilon_i}$

$G = GC$

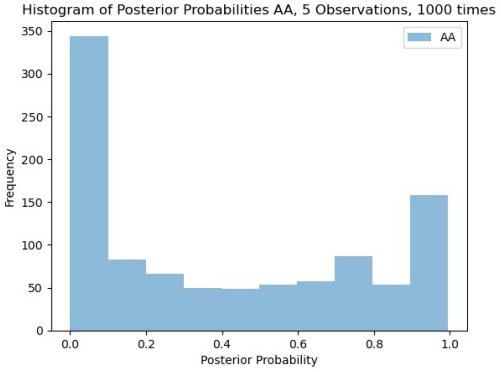
2a) If you assume $P(C_i = A) = P(C_i = B)$, this does simplify calculations for likelihoods of observing different reads.
For example, when you have a heterozygous genotype AB as the ground truth, each allele, A and B, has equal probability of contributing to the observed read. So, $P(i=A|G=AB) = P(i=A|C_i=A)\cdot P(C_i=A) + P(i=A|C_i=B)\cdot P(C_i=B)$ becomes $P(i=A|C_i=A)\cdot 0.5 + P(i=A|C_i=B)\cdot 0.5$, which then becomes $(1)\cdot(0.5)+(0)\cdot(0.5) = \frac{1}{2}$. This applies to both cases for the heterozygous genotype.
So, the likelihood does get reduced here, and gets reduced to $\frac{1}{2}$ for the heterozygous case.
For the homozygous case, there's only one allele present in the genotype so the probability of choosing between alleles doesn't matter.
For example, $P(O_i=A|G=AA) = P(O_i=A|E_i=0, C_i=A, G=AA)\cdot P(E_i=0)\cdot P(C_i=A|G=AA)$ becomes $(1)\cdot(1-\epsilon_i)\cdot(1)$, which is the exact same as originally (in the case of 2 possible alleles in lecture slides).
The likelihood does not get reduced for the homozygous case bc it's unaffected.

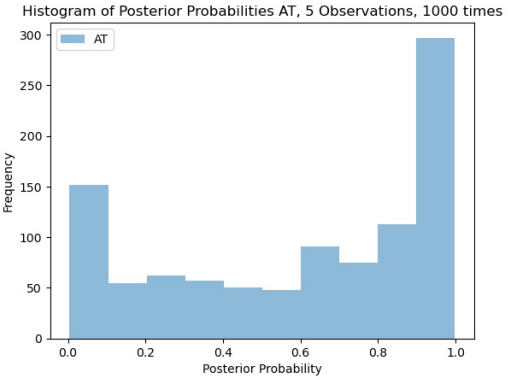
b)
```
P(AA|D)= 4.566267927309115e-300
P(AT|D)= 1.0
P(TT|D) = 6.963414923111202e-290
```
$P(G = AA \mid Data) = 4.57\cdot10^{-300}$
$P(G = AT \mid Data) = \sim 1.00$
$P(G = TT \mid Data) = 6.96\cdot10^{-290}$

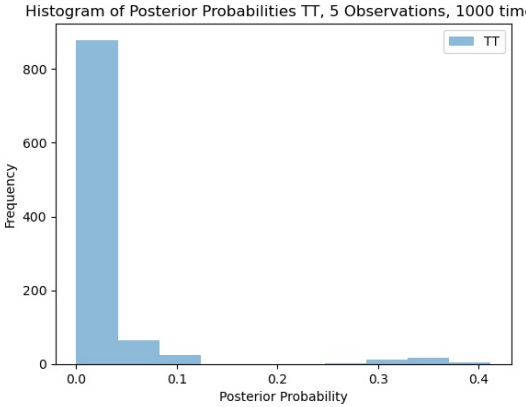
c)
```
P(AA|D)= 0.0007040016426367364
P(AT|D)= 0.9790522220051942
P(TT|D) = 0.020243776352169073
```
$P(G = AA \mid Data) = 0.000704$
$P(G = AT \mid Data) = 0.979$
$P(G = TT \mid Data) = 0.0202$

$P(AA|D)$ and $P(TT|D)$ are now farther away from 0 and are larger values.
$P(AT|D)$ is now smaller than 1.0 and deviated further.
Overall, the posterior probabilities are less skewed toward 0 and 1 when only looking at only 5 observations at a time.

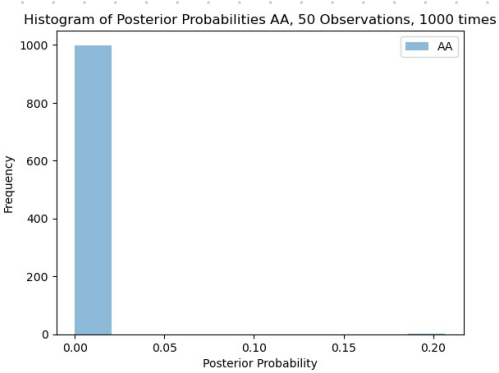
d)

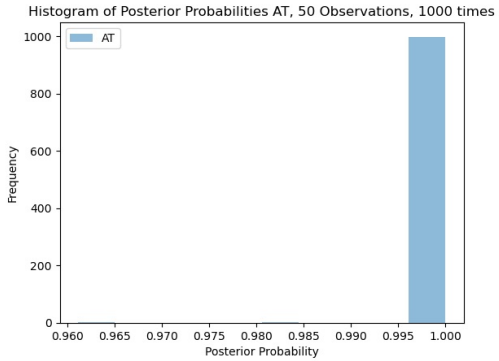


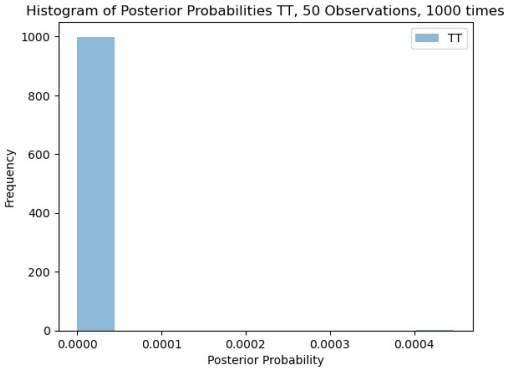




e)







f) When taking 50 observations at a time, the mean of the posterior probability for 1000 samples was closer to the posterior probability of all observations compared to taking 5 observations at a time. For example, the mean posterior probability of AA for $n=5$ was 0.379, but for $n = 50$ for AA, the mean posterior probability was 0.00022. For AT, $n=5$ resulted in a mean posterior prob. of 0.60, but $n=50$ resulted in a mean posterior prob. of 0.99. For TT, $n= 5$ resulted in a mean posterior prob. of 0.019, but $n=50$ resulted in a mean posterior prob. of $4.9 \cdot 10^{-7}$.
Also, the standard deviation for $n=5$ for all genotypes was much greater, indicating more variability in estimating the posterior prob. This can be seen by looking at the histograms. For $n=5$, there was a much larger range of possible posterior probabilities. The standard deviation of AA, AT, and TT were 0.36, 0.34, and 0.05, respectively. However, for $n=50$, the standard deviation of AA, AT, and TT were 0.0065, 0.0013, and $1.4 \cdot 10^{-5}$, respectively. For $n=5$, there's generally a larger skewed tail, and you can see more estimated posterior probabilities that are more "extreme" and farther away from the 1000-sample posterior probability estimate for all genotypes.

g) We assume the base callers' error probabilities are accurate and that the prior probabilities of each genotype (AA, AT, TT) are equal UNLESS specified otherwise. If these base caller probabilities are inaccurate, it could lead to incorrect estimates of the genotype probabilities. Also, assuming equal probabilities for all genotypes might not always be the case b/c one genotype could be more common in real life. So, these factors could lead to biases in posterior probability estimates.

In [1]:
```
import numpy as np
import pandas as pd
import random
import math
import statistics
import matplotlib.pyplot as plt
```

In [2]:
```
reads = pd.read_csv('/Users/timothyliu/Documents/121/reads.tsv', sep='\t')#, i
print(reads['observations'].unique())
print(reads)
#reads[reads['observations'] == 'A']
```

```
['A' 'T']
    observations  probability_of_error  error_truth
0              A              0.125650        False
1              T              0.092379        False
2              A              0.196982        False
3              T              0.063769        False
4              T              0.163563         True
..           ...                   ...          ...
995            A              0.037062        False
996            A              0.027094        False
997            A              0.146039        False
998            T              0.170114         True
999            T              0.038950        False

[1000 rows x 3 columns]
```

b) Write some code to estimate the posterior probability of the three possible genotypes given the data

In [3]:
```
def prob_genotype_b(reads_input, genotype, num_observations):
    reads = reads_input.copy().sample(n=num_observations)

    '''
    P(G)
    '''
    log_prior_prob_AA = np.log(0.95*0.95)
    log_prior_prob_AT = np.log(0.095)
    log_prior_prob_TT = np.log(0.05*0.05)

    # Select the log prior based on genotype
    if genotype == 'AA':
        log_prior_prob = log_prior_prob_AA
    elif genotype == 'AT':
        log_prior_prob = log_prior_prob_AT
    elif genotype == 'TT':
        log_prior_prob = log_prior_prob_TT

    '''
    P(D | G)
    '''
    def prob_data_given_genotype(reads, genotype, num_observations):
        log_P_observationsGivenGenotype = 0

        for observation in range(num_observations):
            base = reads.iloc[observation]['observations']  # Get base of that
```

```
            error = reads.iloc[observation]['probability_of_error']  # Get erro

            if base != genotype[0] and base != genotype[1]:  # If B | AA
                likelihood = error
            elif base == genotype[0] and base == genotype[1]:  # If A | AA
                likelihood = 1 - error
            else:  # If A | AB
                likelihood = 0.5

            log_P_observationsGivenGenotype += np.log(likelihood)

        return log_P_observationsGivenGenotype

    log_prob_dataGgenotype = prob_data_given_genotype(reads, genotype, num_obse

    '''
    P(D)
    '''
    # Compute likelihoods for each genotype
    log_likelihood_AA = prob_data_given_genotype(reads, "AA", num_observations
    log_likelihood_AT = prob_data_given_genotype(reads, "AT", num_observations
    log_likelihood_TT = prob_data_given_genotype(reads, "TT", num_observations

    # Log of total probability of data
    log_prob_data = np.logaddexp(np.logaddexp(log_likelihood_AA + log_prior_pro
                                              log_likelihood_AT + log_prior_pro
                                 log_likelihood_TT + log_prior_prob_TT)

    '''
    Putting it all together
    '''
    log_posterior = log_prob_dataGgenotype + log_prior_prob - log_prob_data
    posterior = np.exp(log_posterior)

    return posterior

# Assuming `reads` DataFrame is defined and available
# Example call
posterior_AA = prob_genotype_b(reads, "AA", 1000)
print("P(AA|D)=", posterior_AA)
posterior_AT = prob_genotype_b(reads, "AT", 1000)
print("P(AT|D)=", posterior_AT)
posterior_TT = prob_genotype_b(reads, "TT", 1000)
print("P(TT|D) =", posterior_TT)
```

```
P(AA|D)= 4.5662679273101534e-300
P(AT|D)= 1.0
P(TT|D) = 6.963414923136534e-290
```

c) Randomly sample 5 observations with replacement and re-estimate the posterior probability of each genotype. What are the results?

In [4]:

```
#P(G | D) = P(D | G) * P(G) / P(D)
def prob_genotype_c(reads_input, genotype, num_observations, random_seed=True):
    if random_seed:
        reads = reads_input.copy().sample(n=num_observations)
    else:
        reads = reads_input.copy().sample(n=num_observations, random_state=6)
    '''
```

```
P(G)
'''
prior_prob_AA = 0.95*0.95
prior_prob_AT = 0.095
prior_prob_TT = 0.05*0.05

if genotype == 'AA':
    prior_prob = prior_prob_AA
elif genotype == 'AT':
    prior_prob = prior_prob_AT
elif genotype == 'TT':
    prior_prob = prior_prob_TT

#print("P(G):", prior_prob)
'''
P(D | G)
'''

def calc_prob_dataGgenotype(reads, genotype, num_observations):

    #Subsample of matrix


    '''
    Genotype: "AA", "AT", or "TT"
    Reads: Dataframe of reads
    Num_observations: Number of observations (from prev)
    '''

    P_observationsGivenGenotype = 1

    for observation in range(0, num_observations):
        base = reads.iloc[observation]['observations']  #Get base of that
        error = reads.iloc[observation]['probability_of_error'] #Get error
        #print(f"Observation {observation}: Base = {base}, Error = {error}"

        #B | AA or A | BB
        #Homozygous genotype
        if base != genotype[0] and base != genotype[1]: #If B | AA
            likelihood = error

        #B | BB or A | AA
        elif base == genotype[0] and base == genotype[1]: #If A | AA
            likelihood = 1 - error

        #B | AB or A | AB
        else:
            likelihood = 1/2
        P_observationsGivenGenotype *= likelihood
    return P_observationsGivenGenotype

prob_dataGgenotype = calc_prob_dataGgenotype(reads, genotype, num_observat
#print("P(D | G):", prob_dataGgenotype)

'''
P(Data)
'''

#AA
likelihood_AA = calc_prob_dataGgenotype(reads, "AA", num_observations)
```

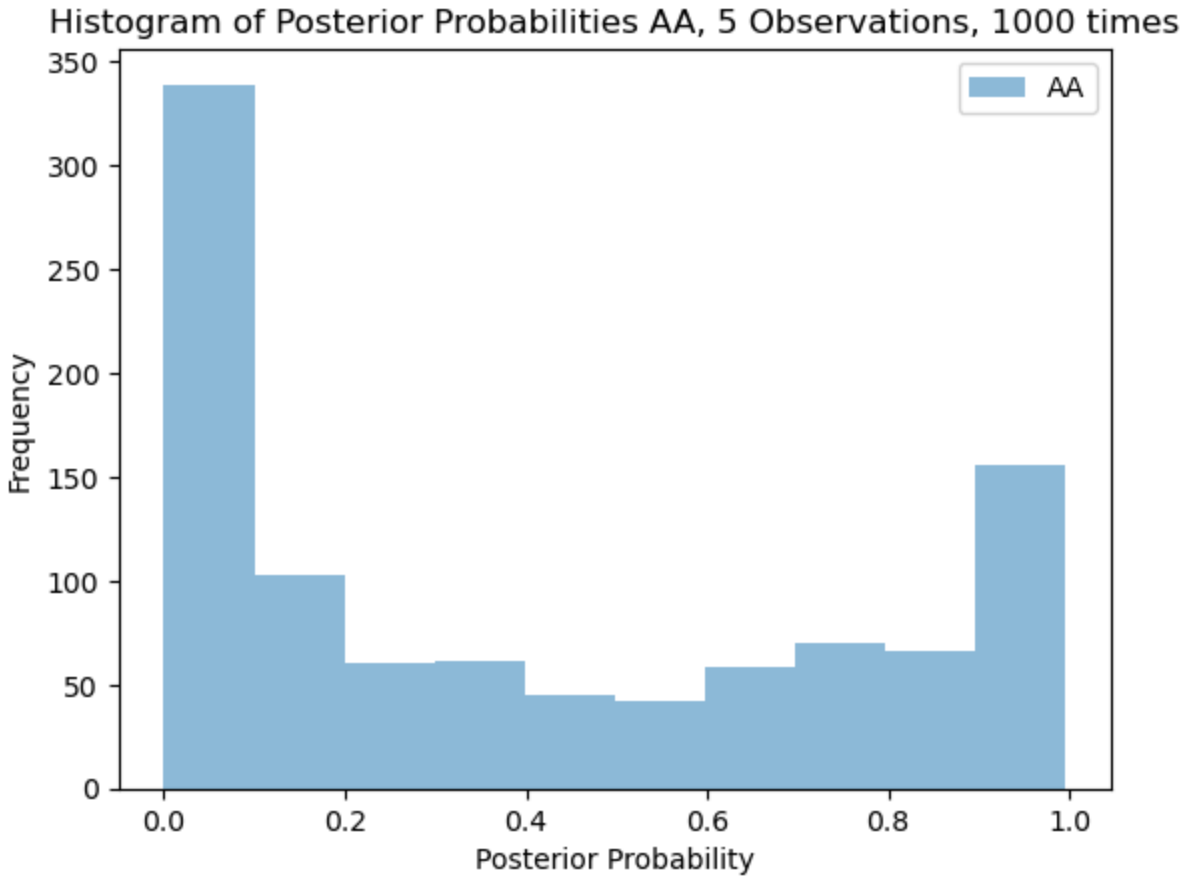
```
    likelihood_AT = calc_prob_dataGgenotype(reads, "AT",num_observations)

    likelihood_TT = calc_prob_dataGgenotype(reads, "TT", num_observations)

    prob_data = ((likelihood_AA * prior_prob_AA)
                 + (likelihood_AT * prior_prob_AT)
                 + (likelihood_TT * prior_prob_TT))
    #print("P(Data):",prob_data)

    '''
    Putting it all together
    '''

    posterior = (prob_dataGgenotype * prior_prob) / prob_data
    #print("Posterior:",posterior)
    return posterior


posterior_AA = prob_genotype_c(reads, "AA", 5, random_seed=False)
print("P(AA|D)=",posterior_AA)
posterior_AT = prob_genotype_c(reads, "AT", 5, random_seed=False)
print("P(AT|D)=",posterior_AT)
posterior_TT = prob_genotype_c(reads, "TT", 5, random_seed=False)
print("P(TT|D) =",posterior_TT)
```

```
P(AA|D)= 0.0007040016426367364
P(AT|D)= 0.9790522220051942
P(TT|D) = 0.020243776352169073
```

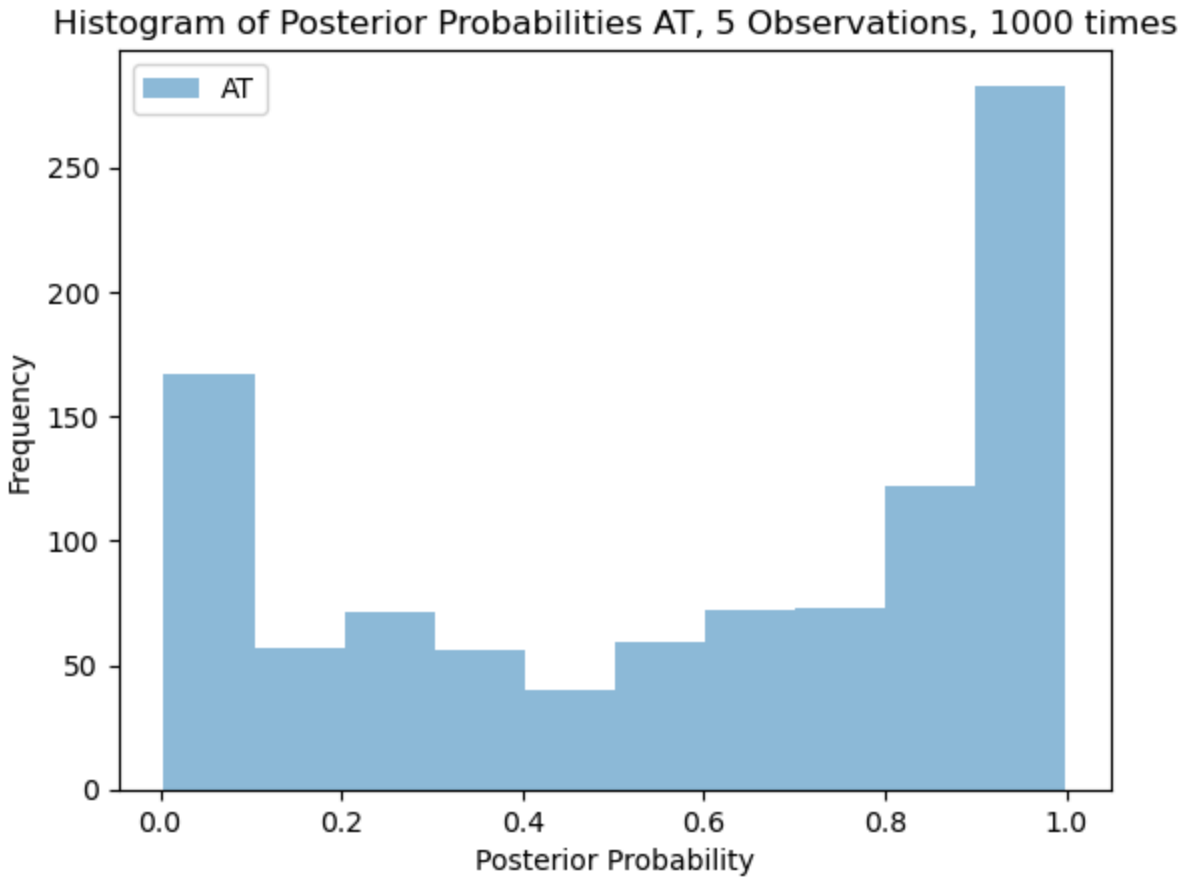
d) Repeat (c) 1,000 times. That means you will have 1,000 estimates for each of your posterior probabilities, each using 5 observations. This procedure is a variation of the bootstrap. Make a histogram for each of the posterior probabilities. Please be mindful of the number of bins and the appearance of your histogram. No one likes an ugly histogram.

In [5]:
```
def make_histogram(list, title, label, bins=10):
    plt.hist(list, bins=bins, alpha=0.5, label=label)
    plt.xlabel('Posterior Probability')
    #plt.xlim(0,1)
    plt.ylabel('Frequency')
    plt.title(title)
    plt.legend()
    plt.show()

posterior_AA_list_5 = []
for i in range(0, 1000):
    posterior_AA_list_5.append(prob_genotype_c(reads, "AA",5))
make_histogram(posterior_AA_list_5, 'Histogram of Posterior Probabilities AA, 5
mean_AA_5 = statistics.mean(posterior_AA_list_5)
stdev_AA_5 = statistics.stdev(posterior_AA_list_5)
print(f"Mean: {mean_AA_5}, Standard Deviation: {stdev_AA_5}")

posterior_AT_list_5 = []
for i in range(0, 1000):
    posterior_AT_list_5.append(prob_genotype_c(reads, "AT", 5))
make_histogram(posterior_AT_list_5, 'Histogram of Posterior Probabilities AT, 5
mean_AT_5 = statistics.mean(posterior_AT_list_5)
stdev_AT_5 = statistics.stdev(posterior_AT_list_5)
print(f"Mean: {mean_AT_5}, Standard Deviation: {stdev_AT_5}")
```

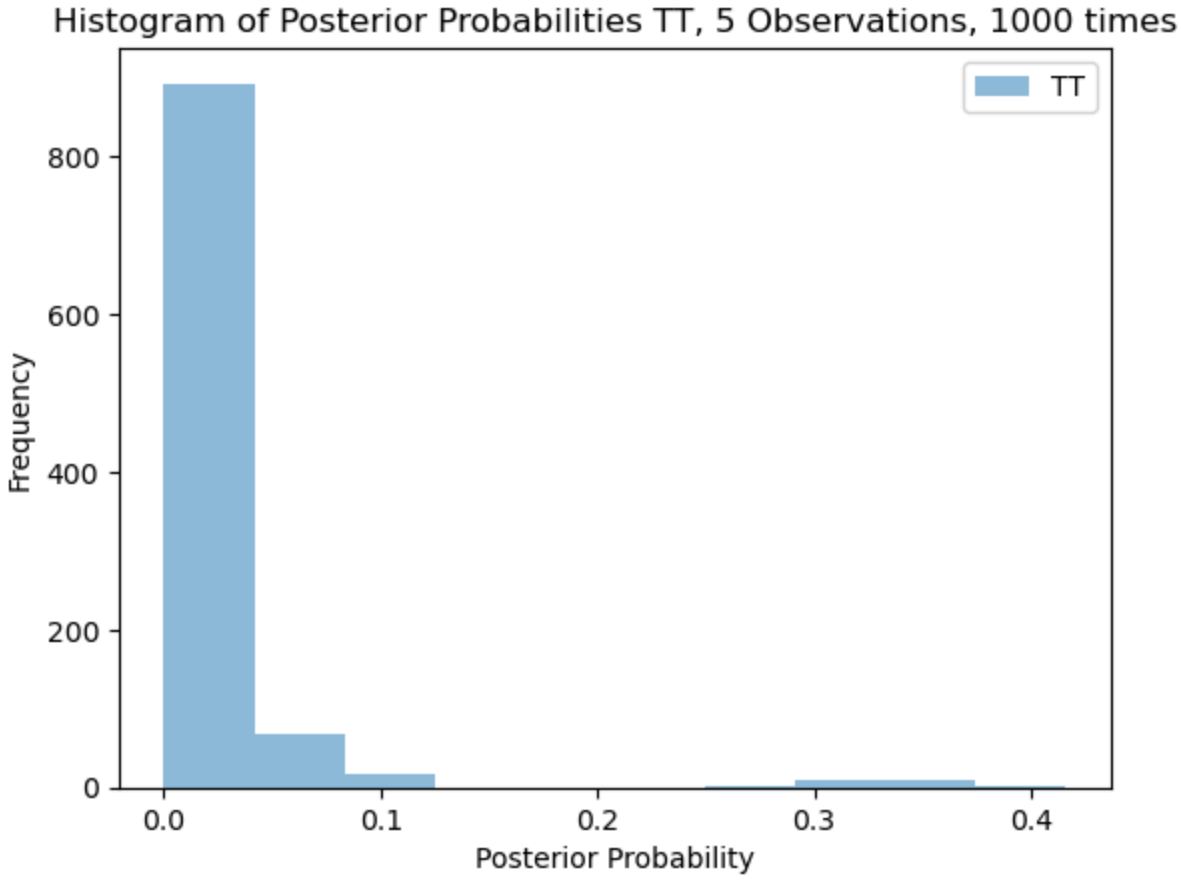
```
posterior_TT_list_5 = []
for i in range(0, 1000):
    posterior_TT_list_5.append(prob_genotype_c(reads, "TT", 5))
make_histogram(posterior_TT_list_5, 'Histogram of Posterior Probabilities TT, 5
mean_TT_5 = statistics.mean(posterior_TT_list_5)
stdev_TT_5 = statistics.stdev(posterior_TT_list_5)
print(f"Mean: {mean_TT_5}, Standard Deviation: {stdev_TT_5}")
```




```
Mean: 0.3976534234101415, Standard Deviation: 0.36266924687145935
```



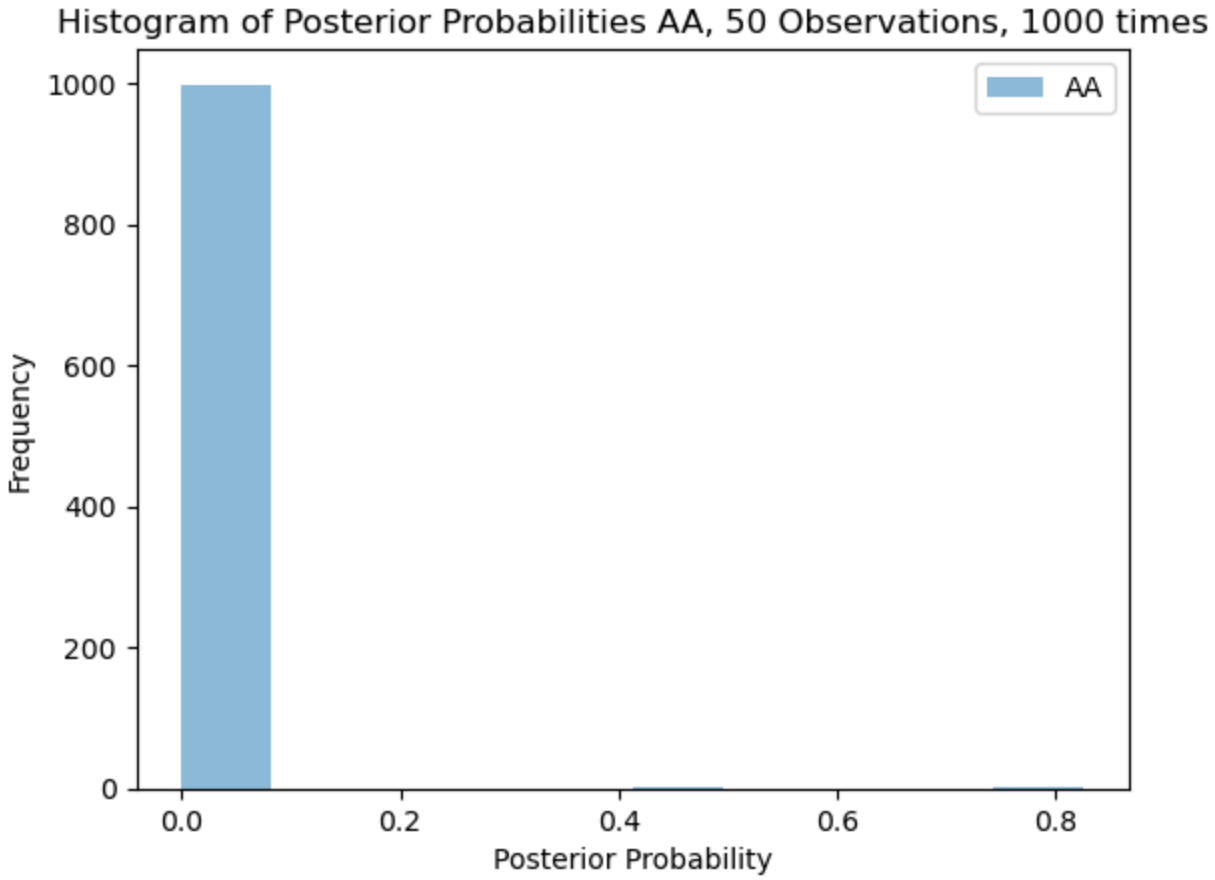

```
Mean: 0.5780704879226927, Standard Deviation: 0.34892567976833005
```



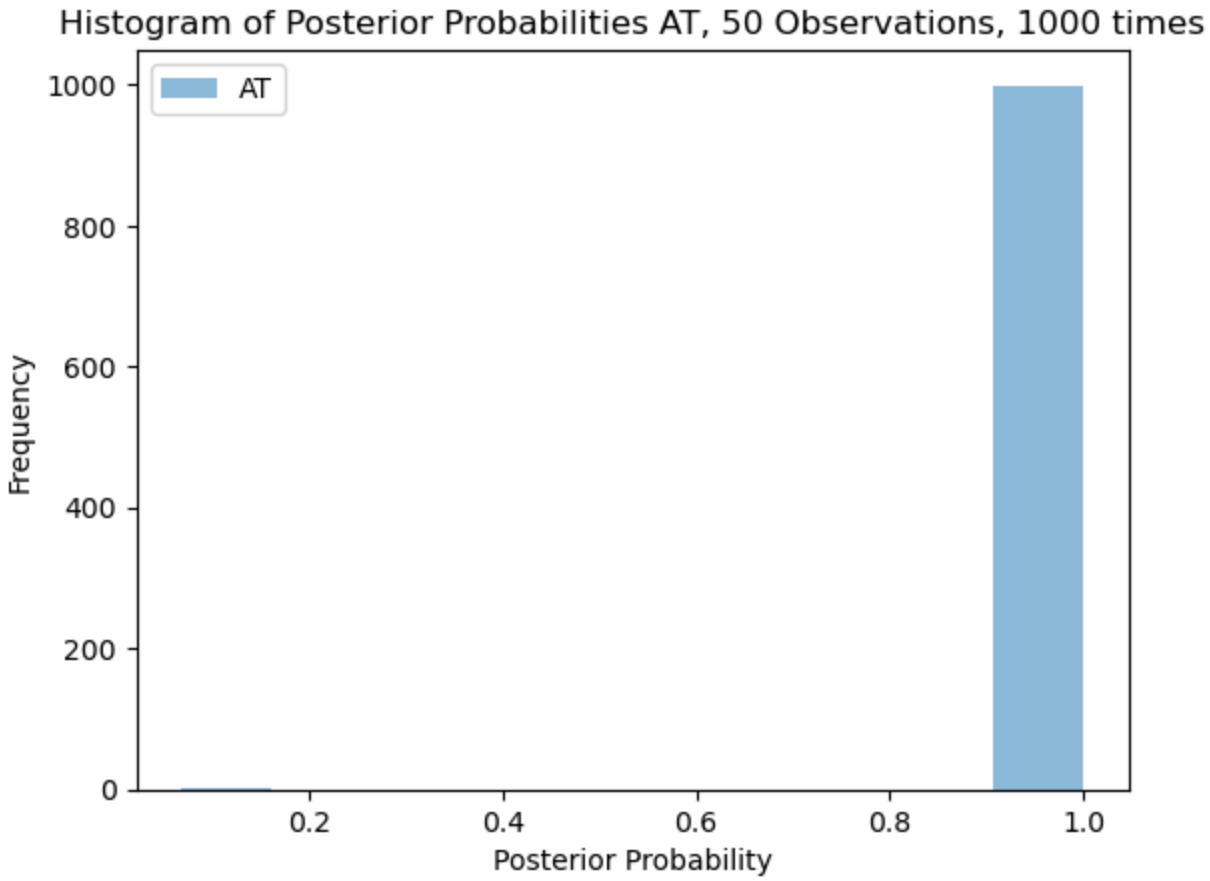

```
Mean: 0.01680041570704586, Standard Deviation: 0.05386400792756098
```

e) Repeat (d), but this time instead of taking 5 observations, take 50. Again, make three histograms.

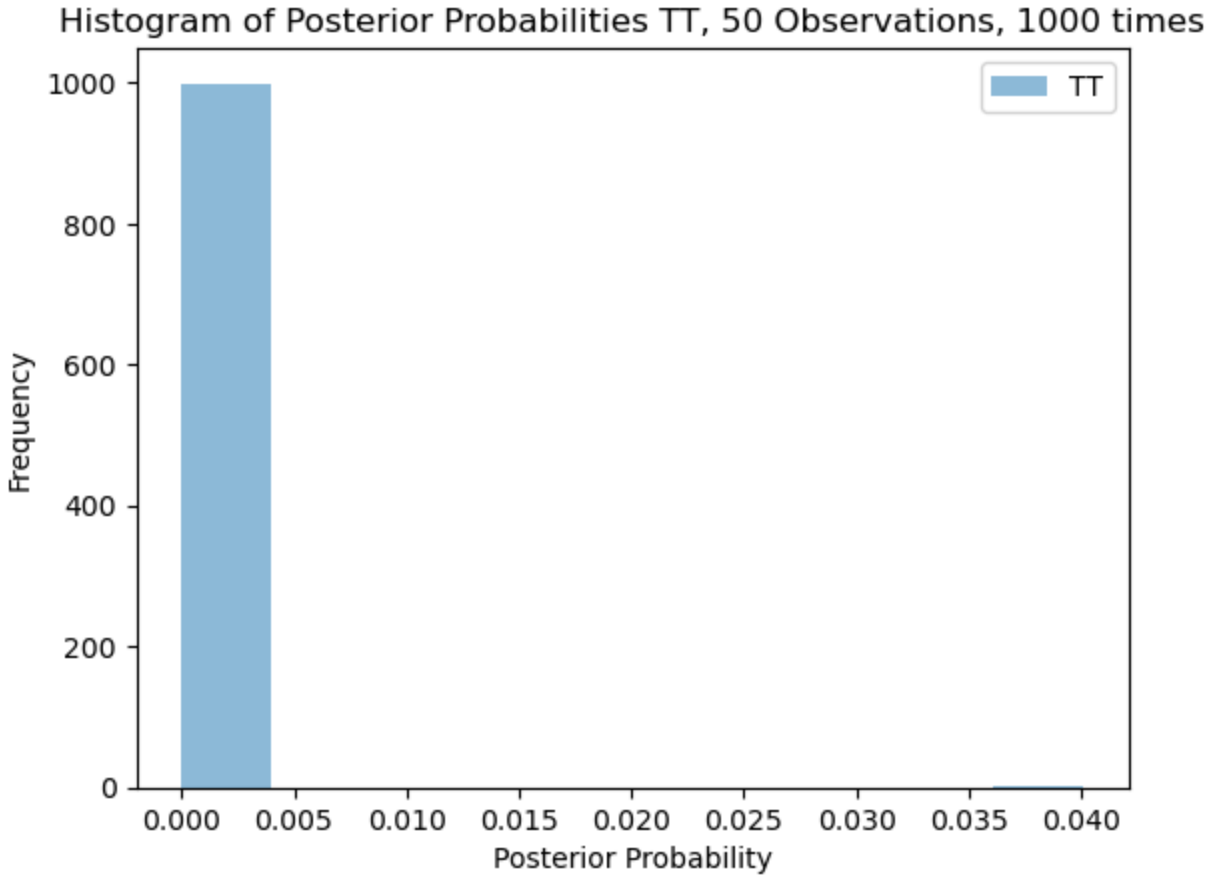
In [6]:
```
posterior_AA_list_50 = []
for i in range(0, 1000):
    posterior_AA_list_50.append(prob_genotype_c(reads, "AA", 50))
make_histogram(posterior_AA_list_50, 'Histogram of Posterior Probabilities AA,
mean_AA_50 = statistics.mean(posterior_AA_list_50)
stdev_AA_50 = statistics.stdev(posterior_AA_list_50)
print(f"Mean: {mean_AA_50}, Standard Deviation: {stdev_AA_50}")

posterior_AT_list_50 = []
for i in range(0, 1000):
    posterior_AT_list_50.append(prob_genotype_c(reads, "AT", 50))
make_histogram(posterior_AT_list_50, 'Histogram of Posterior Probabilities AT,
mean_AT_50 = statistics.mean(posterior_AT_list_50)
stdev_AT_50 = statistics.stdev(posterior_AT_list_50)
print(f"Mean: {mean_AT_50}, Standard Deviation: {stdev_AT_50}")

posterior_TT_list_50 = []
for i in range(0, 1000):
    posterior_TT_list_50.append(prob_genotype_c(reads, "TT", 50))
make_histogram(posterior_TT_list_50, 'Histogram of Posterior Probabilities TT,
mean_TT_50 = statistics.mean(posterior_TT_list_50)
stdev_TT_50 = statistics.stdev(posterior_TT_list_50)
print(f"Mean: {mean_TT_50}, Standard Deviation: {stdev_TT_50}")
```




```
Mean: 0.0013459565004476288, Standard Deviation: 0.029880141213319496
```




```
Mean: 0.9990643202691482, Standard Deviation: 0.029461578623106385
```




```
Mean: 4.0193286182381496e-05, Standard Deviation: 0.0012679733649842238
```