# 4. Programming (using python)

## 4(a).

```python
import numpy as np
import matplotlib.pyplot as plt

def coin_simulate(num_of_flips, coin_type):
    sequence = []
    if coin_type == "fair":
        sequence = ["H" if np.random.uniform() < 0.5 else "T" for _ in range(num_of_flips)]
    elif coin_type == "bias":
        sequence = ["H" if np.random.uniform() < 0.25 else "T" for _ in range(num_of_flips)]
    return sequence

for i in range(5):
    print("======== 40 Flip Sequence {} ========".format(i+1))
    print("============= Bias ==================")
    print(coin_simulate(40, "bias"))
    print("============= Fair ==================")
    print(coin_simulate(40, "fair"))
    print("\n")
```

## Output:

======== 40 Flip Sequence 1 ========

============= Bias ==================

['H', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T']

============= Fair ==================

['H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'H', 'H', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T']

======== 40 Flip Sequence 2 ========

============= Bias ==================

['T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T']

============= Fair ==================

['T', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'T', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T']

======== 40 Flip Sequence 3 ========

========== Bias ==========

['T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T']

========== Fair ==========

['T', 'H', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H', 'H']

======== 40 Flip Sequence 4 ========

========== Bias ==========

['T', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'H']

========== Fair ==========

['H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'H', 'H', 'H', 'H', 'T', 'T', 'T', 'H', 'T', 'H', 'T', 'H', 'H', 'H', 'H', 'H', 'T']

======== 40 Flip Sequence 5 ========

========== Bias ==========

['T', 'T', 'H', 'T', 'H', 'T', 'T', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H']

========== Fair ==========

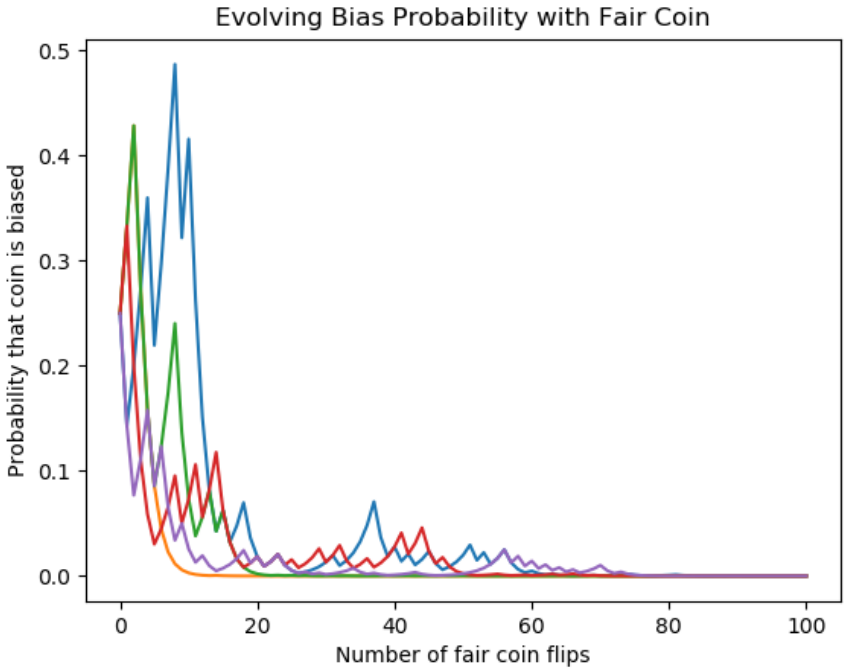
['T', 'T', 'H', 'H', 'T', 'T', 'H', 'H', 'T', 'T', 'T', 'T', 'H', 'H', 'T', 'T', 'H', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'T', 'H', 'T', 'H', 'H', 'T', 'H', 'H', 'H', 'T', 'H', 'T', 'T', 'T', 'T', 'T']

4(b).

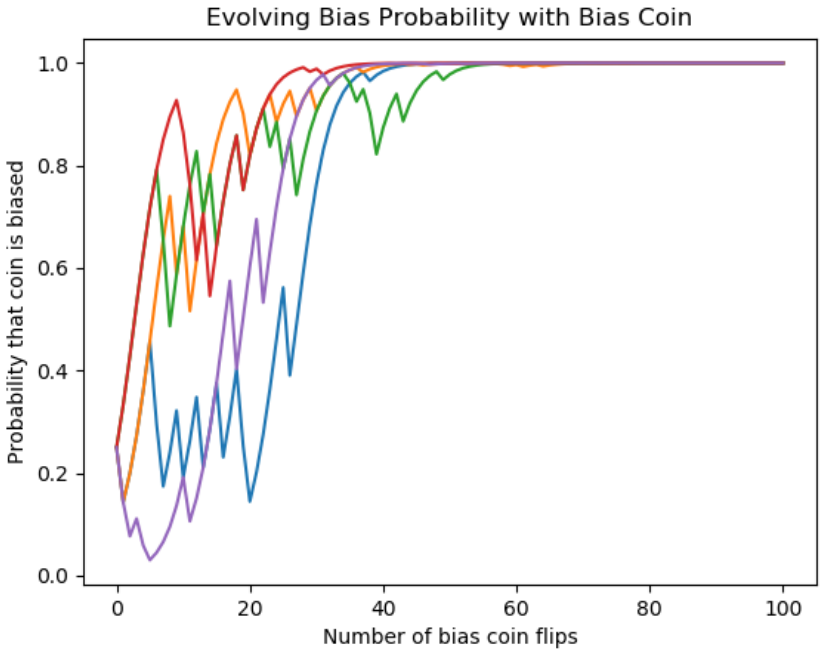
```
def coin_simulate_with_likelihood(num_of_flips, coin_type="fair"):
    """
        probability of choosing biased coin is 1/4
        probability of choosing fair coin is 3/4
        for bias:
        choosing head is 1/4
        choosing tails is 3/4
    """
    sequence = []
    bias = [0.25] # starts at 0 flips
    fair = [0.75]
    p_h = 0.5 # probability of heads
    if coin_type == "bias": p_h = 0.25
    for i in range(num_of_flips):
        if np.random.uniform() < p_h:
            sequence.append("H")
            bias.append(bias[i]*p_h)
            fair.append(fair[i]*0.50)
        else:
            sequence.append("T")
            bias.append(bias[i]*(1-p_h))
            fair.append(fair[i]*0.50)

    likelihood_bias = [bias[i]/(bias[i]+fair[i]) for i in range(len(bias))]
    return sequence, likelihood_bias
```

```
plt.figure(1)
flips = [i for i in range(101)]
for _ in range(5):
    coin_sequence, bias_prob = coin_simulate_with_likelihood(100, "fair")
    plt.plot(flips, bias_prob)
plt.title("Evolving Bias Probability with Fair Coin")
plt.ylabel("Probability that coin is biased")
plt.xlabel("Number of fair coin flips")

plt.figure(2)
for _ in range(5):
    coin_sequence, bias_prob = coin_simulate_with_likelihood(100, "bias")
    plt.plot(flips, bias_prob)
plt.title("Evolving Bias Probability with Bias Coin")
plt.ylabel("Probability that coin is biased")
plt.xlabel("Number of bias coin flips")

plt.show()
```

4(c).




4(d).




4.2. difficulty: easy

5.1

I found the probability and programming sections to be easy and did not need any review. A little under 1 hour to complete both at a leisurely pace.

The linear algebra portion required a little bit of review but was easy after brushing up for approximately 45 minutes. Took only 10ish minutes to finish afterwards.

The differential equations portion took me a bit to solve and required significant review (~2.5 hours). My last controls class was about 2 years ago, so I had to brush up on Laplace transforms and state-space representations. Was unable to come up with an answer for 2(c).