Stanley Ung

011743626

EE 381

**Conditional Probabilities**

**Problem 1:**

**Intro:**

For this problem, we want to determine if the transmission of a bit can send successfully from the sender to a receiver and still be the same bit at the end.

**Methodology:**

Using the given probabilities, I use Boolean logic to determine if the bit will be 1 or 0. This simplifies the code and returns the bit that is needed. The methodology is simple here, I would process the bits and check if S is NOT equal to R.

**Code:**

def problem\_1(N):

M = [np.random.rand() for i in range(N)]

P\_0 = 0.4

E\_0 = 0.02

E\_1 = 0.03

success = 0

for i in range(0, N):

S = int(not M[i] <= P\_0) # create msg S

R = int(not np.random.rand() <= E\_1) if S == 1 else int(np.random.rand() <= E\_0)

if R != S: # Transmitted incorrectly

success += 1

return success / N

**Results:**

There is only a roughly 2% chance that the transmission will fail.

|  |  |
| --- | --- |
| **Probability of transmission error** |  |
| **Ans.** | **p =** 0.02661 |

**Problem 2:**

**Intro:**

This problem is a simplified version of problem 1, where we only need to pay attention to the S signals that are 1 and see what the result is for R. If the R is 1, we have succeeded in the experiment

**Methodology:**

This problem follows the same logic as problem 1, but now we are only checking the signals where S = 1. If S is equal to 1, I check that R is also equal to 1 and add it to the number of successes.

**Code:**

def problem\_2(N):

E\_1 = 0.03

success = 0

for i in range(0, N):

R = np.random.rand()

success += int(R >= E\_1)

return success / N

**Results:**

When S is 1, there is a very high chance that the bit will be sent and decoded successfully at 96%.

|  |  |
| --- | --- |
| **Conditional Probability P(R = 1 | S = 1)** |  |
| **Ans.** | **p =** 0.9697 |

**Problem 3:**

**Intro:**

This problem is the reverse of problem 2 where we must check that S = 1 given that we know R = 1. We can check the received message to see if we got 1 as the bit, but we must make sure that the sender bit was also in fact a 1.

**Methodology:**

Following the same methodology as problem 2, I now added a new counter to keep track of when S and R are both 1 and one when we have just R. We divide these to get the probability.

**Code:**

def problem\_3(N):

M = [np.random.rand() for i in range(N)]

P\_0 = 0.4

E\_0 = 0.02

E\_1 = 0.03

success = 0

R\_success = 0

for i in range(0, len(M)):

S = int(not M[i] <= P\_0) # Create msg S

R = int(not np.random.rand() <= E\_1) if S == 1 else int(np.random.rand() <= E\_0)

success += int(S == 1 and R == 1)

R\_success += int(R == 1)

return success / R\_success

**Results:**

For the R’s that are 1’s, there was a very high chance they were decoded correctly from a bit that was 1.

|  |  |
| --- | --- |
| **Conditional Probability P(S = 1 | R = 1)** |  |
| **Ans.** | **p =** 0.9875 |

**Problem 4:**

**Intro:**

This problem sends three bits instead of one to improve the efficiency of the transmission. The three bits are sent and decoded into the receiving bit in the same way they have in the other problems. We then use majority rule to determine what bit it will become in the end D.

**Methodology:**

Here I use lists to represent the three bits and sum up for majority rule. The R is decoded and compared to the S in the same way is has been and we determine if the transmission has been successful or not from that.

**Code:**

def problem\_4(N):

M = [np.random.rand() for i in range(N)]

P\_0 = 0.4

E\_0 = 0.02

E\_1 = 0.03

success = 0

for i in range(0, len(M)):

S = int(not M[i] <= P\_0) # Create msg S

S\_bits = [S, S, S] # Transmit the same bit 3 times

R = list(map(lambda r : int(np.random.rand() > E\_1) if S == 1 else int(np.random.rand() <= E\_0), S\_bits))

D = int(sum(R) > 0)

success += int(S != D)

return success / N

**Results:**

In conclusion, transmitting three bits instead of one is much more efficient because we get a lower probability of error in decoding. Compared to problem 1 at 2%, we now have and error rate of 0.2% only.

|  |  |
| --- | --- |
| **Probability of error with enhanced transmission** |  |
| **Ans.** | **p =** 0.00238 |