Roberto Sanchez (014587792)
October 3, 2017
EE 381 – Probability and Statistics with Applications to Computing
Lab 2 – Conditional Probabilities: Communication through a Noisy Channel

1) Probability of erroneous transmission
    a) **INTRODUCTION:**
        i) In this problem we are trying to generate the probability of a message (usually a bit 1 or 0) into a noisy communication channel.
    b) METHODOLOGY:
        i) We solved this by first generating a random 'm' to pass through our first if-statements to generate 'S'. Once generated, 'S' gets passed through the transmission definition and then gets caught in a series of if-statements to determine if the correct value comes out or not. Once the definition returns a value it gets compared to the original value. If they match it was a successful transmission. Taking the probability of success divided by the total amount gave us the probability of successful transmission. To get failed transmission we subtracted the successes from 1 to get failures.
    c) RESULTS AND CONCLUSIONS:
        i) From running the program, the probability of transmission error is the following:

Probability of Transmission Error	
Ans.	**P=0.02459**

2) Conditional Probability: P(R=1|S=1)
    a) INTRODUCTION:
        i) We create one bit message and send it to the noisy channel and see when S= 1 we count the probability of success
    b) METHODOLOGY:
        i) S is set at 1 then we send it to the noisy channel and only compare with the if statements that deal with S=1. Then returns R to be tested with the original message.

    c) RESULTS AND CONCLUSIONS:

Conditional Probability P(R=1|S=1)	
Ans.	**P=0.97035**

3) Conditional Probability: $P(S=1\mid R=1)$
   - a) **INTRODUCTION:**
     - i) Same thing as the previous questions however we now now want for R=1.
   - b) **METHODOLOGY:**
     - i) Same as the previous, instead we look for if R=1
   - c) **RESULTS AND CONCLUSIONS:**

Conditional Probability $P(S=1\mid R=1)$	
Ans.	**P=0.97013**

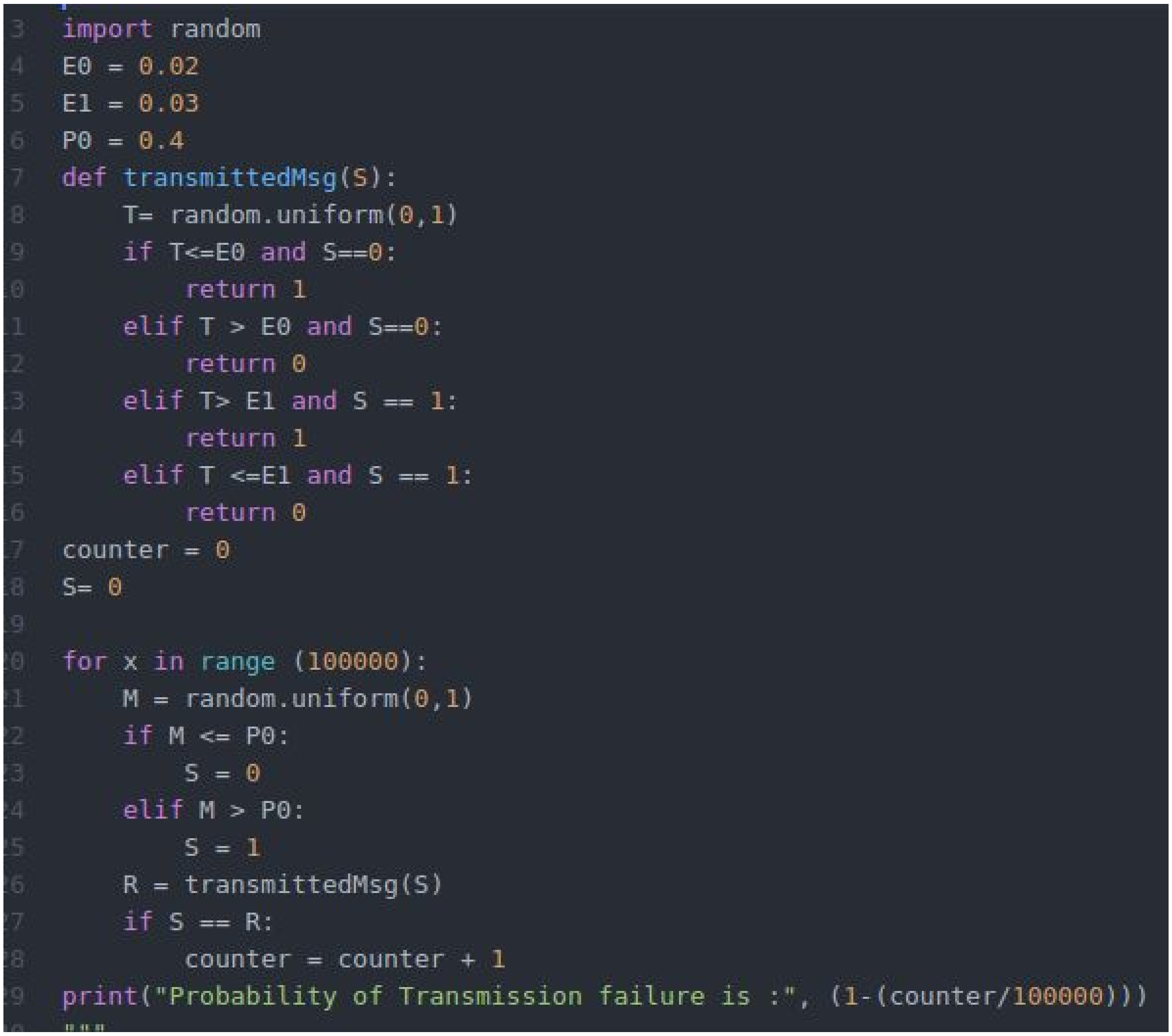
4) Enhanced Transmission Method
   - a) **INTRODUCTION:**
     - i) It's the same thing as part 1 however we are taking account that we are passing 3 digits to see how to prevent data loss be majority rules.
   - b) **METHODOLOGY:**
     - i) We run a list of 3 digits into the noisy channel and see what bits change. Then the output message comes out and there is different bits then we check which is the majority in the list. If there are more 1s than 0s then the original bit is 1.
   - c) **RESULTS AND CONCLUSIONS:**

Probability of error with enhanced transmission	
Ans.	**P= 0.001909999**

The Probability of error with enhanced transmission is 0.002

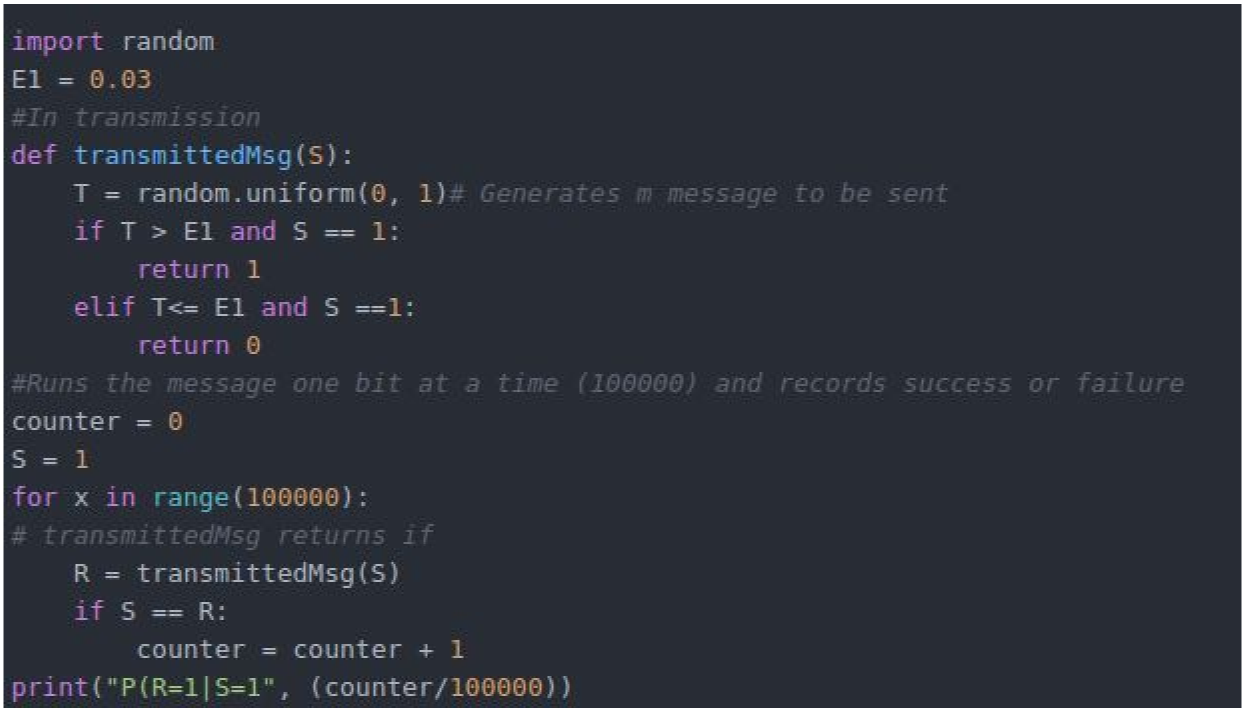
5) Appendix

a) Code for #1



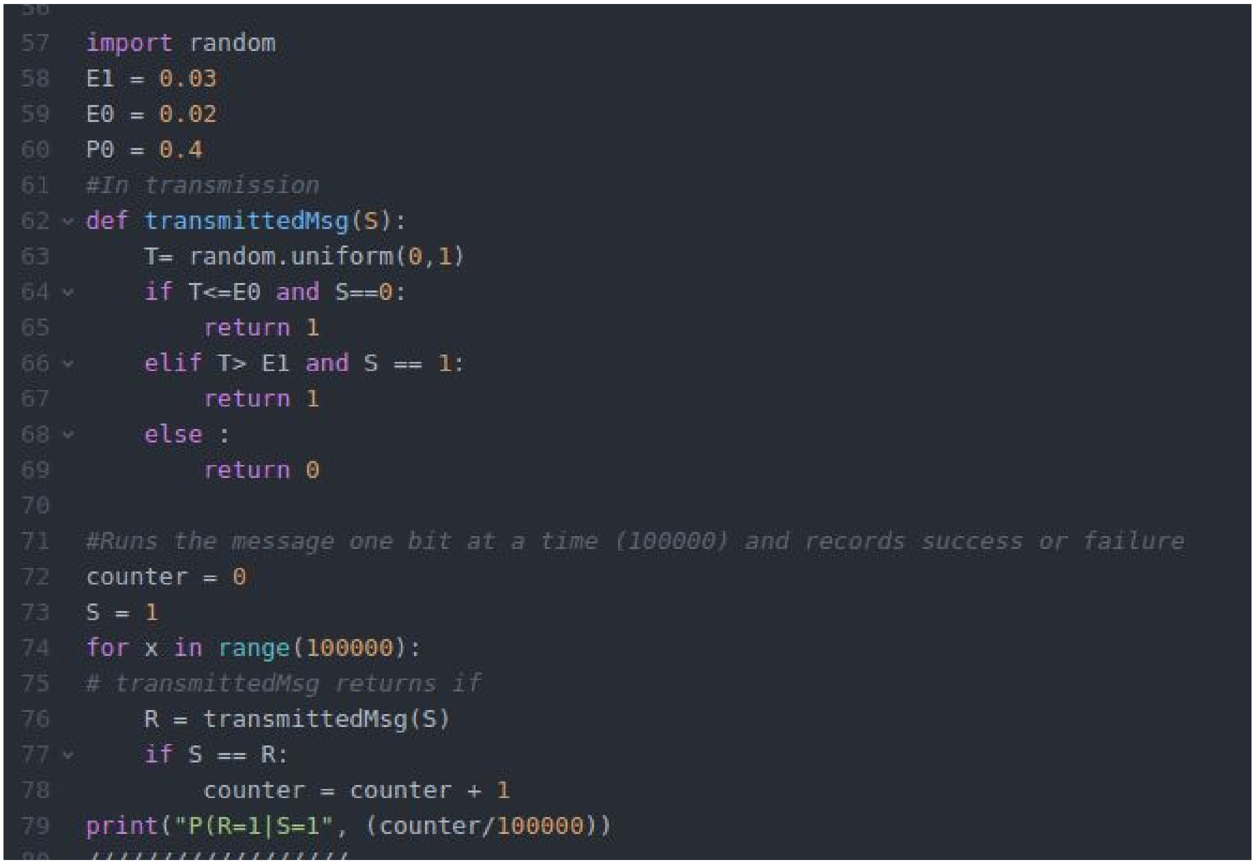
```
import random
E0 = 0.02
E1 = 0.03
P0 = 0.4
def transmittedMsg(S):
    T= random.uniform(0,1)
    if T<=E0 and S==0:
        return 1
    elif T > E0 and S==0:
        return 0
    elif T> E1 and S == 1:
        return 1
    elif T <=E1 and S == 1:
        return 0
counter = 0
S= 0

for x in range (100000):
    M = random.uniform(0,1)
    if M <= P0:
        S = 0
    elif M > P0:
        S = 1
    R = transmittedMsg(S)
    if S == R:
        counter = counter + 1
print("Probability of Transmission failure is :", (1-(counter/100000)))
"""
```

b) Code for #2



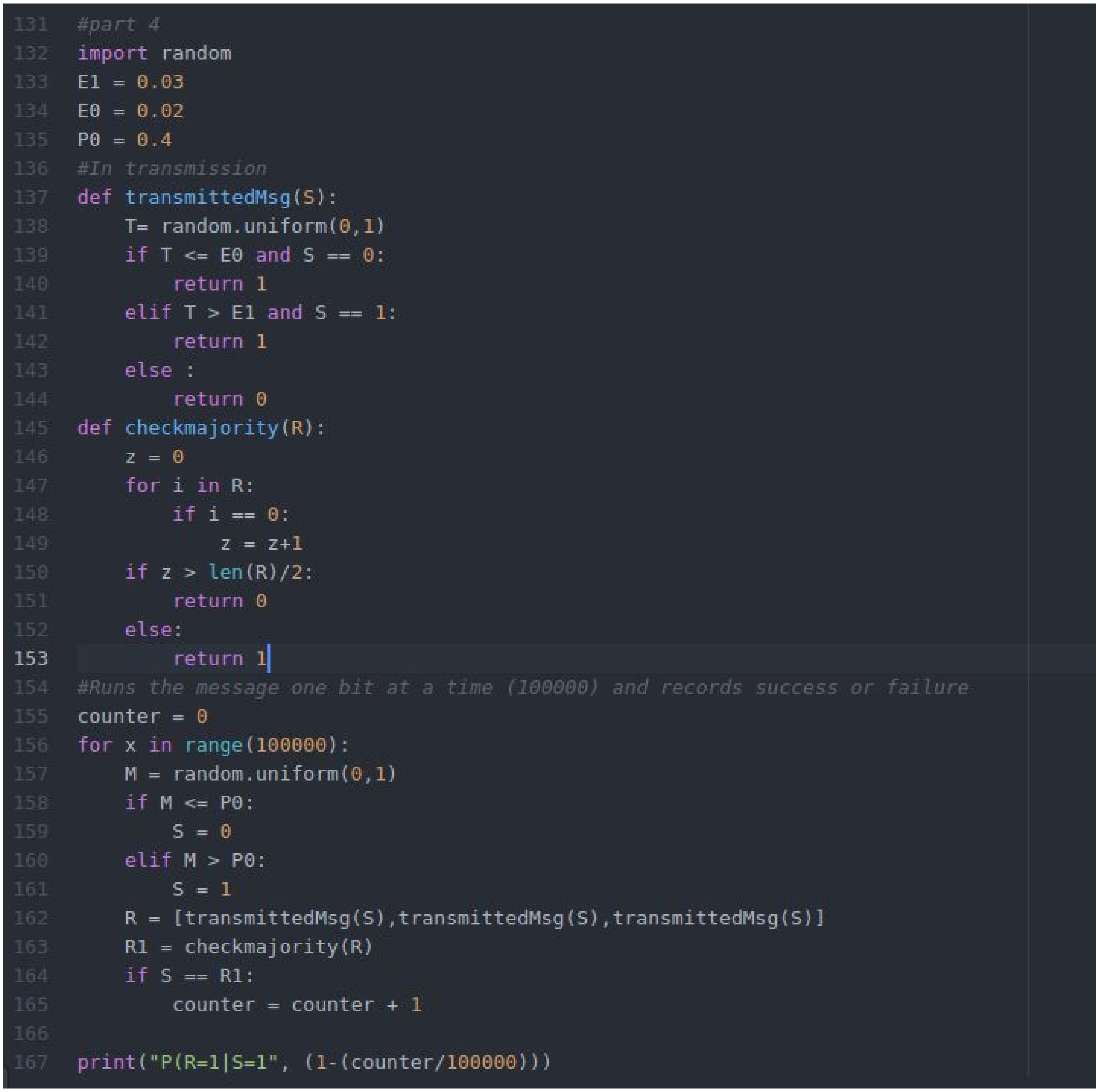
```
import random
E1 = 0.03
#In transmission
def transmittedMsg(S):
    T = random.uniform(0, 1)# Generates m message to be sent
    if T > E1 and S == 1:
        return 1
    elif T<= E1 and S ==1:
        return 0
#Runs the message one bit at a time (100000) and records success or failure
counter = 0
S = 1
for x in range(100000):
# transmittedMsg returns if
    R = transmittedMsg(S)
    if S == R:
        counter = counter + 1
print("P(R=1|S=1", (counter/100000))
```

c) Code for #3



```
import random
E1 = 0.03
E0 = 0.02
P0 = 0.4
#In transmission
def transmittedMsg(S):
    T= random.uniform(0,1)
    if T<=E0 and S==0:
        return 1
    elif T> E1 and S == 1:
        return 1
    else :
        return 0

#Runs the message one bit at a time (100000) and records success or failure
counter = 0
S = 1
for x in range(100000):
# transmittedMsg returns if
    R = transmittedMsg(S)
    if S == R:
        counter = counter + 1
print("P(R=1|S=1", (counter/100000))
//////////////////
```

d) Code for #4



```
#part 4
import random
E1 = 0.03
E0 = 0.02
P0 = 0.4
#In transmission
def transmittedMsg(S):
    T= random.uniform(0,1)
    if T <= E0 and S == 0:
        return 1
    elif T > E1 and S == 1:
        return 1
    else :
        return 0
def checkmajority(R):
    z = 0
    for i in R:
        if i == 0:
            z = z+1
    if z > len(R)/2:
        return 0
    else:
        return 1
#Runs the message one bit at a time (100000) and records success or failure
counter = 0
for x in range(100000):
    M = random.uniform(0,1)
    if M <= P0:
        S = 0
    elif M > P0:
        S = 1
    R = [transmittedMsg(S),transmittedMsg(S),transmittedMsg(S)]
    R1 = checkmajority(R)
    if S == R1:
        counter = counter + 1

print("P(R=1|S=1", (1-(counter/100000)))
```