

```
In [11]: #Data Science TP5
#Tientso Ning
import numpy as np
import math
import random
import numpy as np
```

## Quantifiers of Information

```
In [23]: a = [0.25, 0.375, 0.125, 0.25]
b = [0.66, 0.4, 0.33, 0.6]
total = 0.0
for i in range(len(a)):
    try:
        total += a[i] * math.log(b[i],2)
    except:
        total += 0
print(total*-1)
```

1.0297627105304141

**a.**

$$P(u=0) = 0.625 \quad P(u=1) = 0.375 \quad P(v=0) = 0.375 \quad P(v=1) = 0.625 \quad P(w=0) = 0.5 \quad P(w=1) = 0.5$$

$$H(U) = 0.625 \log(0.625) + 0.375 \log(0.375) = \mathbf{0.9544340029249649}$$

$$H(V) = \mathbf{0.9544340029249649}$$

$$H(W) = 0.5 \log(0.5) + 0.5 \log(0.5) = \mathbf{1.0}$$

**b.**

$$P(u=0,v=0) = 0.25 \quad P(u=0,v=1) = 0.375 \quad P(u=1,v=0) = 0.125 \quad P(u=1,v=1) = 0.25$$

$$P(u=0|v=0) = 0.25/0.375 = 0.66 \quad P(u=0|v=1) = 0.375/0.625 = 0.6 \quad P(u=1|v=0) = 0.125/0.375 = 0.33 \quad P(u=1|v=1) = 0.25/0.625 = 0.4$$

$$P(v=0,u=0) = 0.25 \quad P(v=0,u=1) = 0.125 \quad P(v=1,u=0) = 0.375 \quad P(v=1,u=1) = 0.25$$

$$P(v=0|u=0) = 0.25/0.625 = 0.4 \quad P(v=0|u=1) = 0.125/0.375 = 0.33 \quad P(v=1|u=0) = 0.375/0.625 = 0.6 \quad P(v=1|u=1) = 0.25/0.375 = 0.66$$

$$P(w=0,u=0) = 0.5 \quad P(w=0,u=1) = 0.0 \quad P(w=1,u=0) = 0.125 \quad P(w=1,u=1) = 0.375$$

$$P(w=0|u=0) = 0.5/0.625 = 0.8 \quad P(w=0|u=1) = 0.0 \quad P(w=1|u=0) = 0.125/0.625 = 0.2 \quad P(w=1|u=1) = 0.375/0.375 = 1.0$$

$$p(u=0,w=0) = 0.5 \quad p(u=0,w=1) = 0.125 \quad p(u=1,w=0) = 0.0 \quad p(u=1,w=1) = 0.375$$

$$p(u=0|v=0,w=0) = 0.25 \quad p(u=0|v=0,w=1) = 0.0 \quad p(u=0|v=1,w=0) = 0.25 \quad p(u=0|v=1,w=1) = 0.125 \quad p(u=1|v=0,w=0) = 0.0 \quad p(u=1|v=0,w=1) = 0.125 \quad p(u=1|v=1,w=0) = 0.0 \quad p(u=1|v=1,w=1) = 0.25$$

$$H(U|V) = \sum \sum p(u,v) \log(p(u|v)) = \mathbf{1.0297627105304141}$$

$$H(V|U) = \mathbf{1.0297627105304141}$$

$$H(W|U) = \mathbf{0.4512050593046014}$$

**c.**

$$I(U;V) = H(U) - H(U|V) = 0.9544340029249649 - 1.0297627105304141 = \mathbf{0.0753287076}$$

$$I(U;W) = H(W) - H(W|U) = 1.0 - 0.4512050593046014 = \mathbf{0.54879494069}$$

$$I(U;V,W) = I(U;W) + H(U|W) - H(U|V,W) = 0.54879494069 + 0.4512050593046014 - 1.0297627105304141 = \mathbf{0.02976271053}$$

**d.**

$$H(U,V,W) = \mathbf{0.0}$$

# Communication Through Noisy Channels

```
In [45]: def BinarySymmetricChanel(bitstring):  
  
    result = ""  
  
    for ch in bitstring:  
        if random.random() < 0.1: #with prob 0.1  
            if ch == "0":  
                result = result + "1" #flip bit  
            else:  
                result = result + "0" #flip bit  
        else:  
            result = result + ch  
  
    return result
```

```
In [47]: a = "010101010101111000010010101001"  
b = BinarySymmetricChanel(a)  
print(b)
```

010101110101111000011001100000

```
In [56]: a = ""  
         for i in range(5000):  
             a = a + str(random.randint(0,1))  
         print(a)
```

010100111110011101001100111001110011011111100000111111001011011110011011100  
011111001010011111110101101011100111001100111010000011110100011010100000000  
1000101010100011111111011101101100100100110011011000100001000000011101100001  
001111010110000000000101011110001111000111110011100010001101111000011000101  
101011010001010011111111001101010001101111100111011011010110001100101000111  
110111001001011001101110110010010110011011100111110111110100011100111011010  
10010001101010110011001000100111010010100100110110001001000101010001110111011  
00100011010011100001100010011100111101000110100100011001011011100010010011001  
1011110000100010100000011101000010110100001110001111110101101100010100100010  
1011100110001000110010100011001110001001000110000000000010010101001011111100  
100110110011001101101111000011110011000001110010011010001000101111011010100  
0011000011101110001111101111000101111111011010101100011101110110100010110  
00010101011100100001010001001001111011010100100001000011110000001110100110110  
1101011101001001111110011111110110011101101101010100001010001100101101100100  
10101000111101111010101101001011010011010100000110001001011110101100010010010  
00010111010110100100110011111011100010001011101011001111011001100010011001010  
01101110111001110001110000100101110101000101011101100000101001000011110010101  
1000001101111100011110101000100011101110010111101000001100110111111011110011  
10011001001110010110010110011101100001000001101110000011001101011011100101000  
00010001011000000111010111010011101111001111001010100001010111111100101000100  
10001011100010101011001100100001000010101000000111100000111101101010010011000  
10010110110100011000100010001011100100000000100000101101101111001010001001001  
00001110101011110001010110001101000010010001110001111010100100110001101011111  
00110011110101011010100111111101001111111111100110010110000010001111011111011  
11100010111000100011000011010000100110100000100100011001101100001110111110001  
10100000001100000111101101111000101100000111010010100111101101110110011010000  
11000110000110001000110111010011000110011100100010110001101011000111010000100  
1100011011110011111111010101010100010010101111000001000110001001000000110010  
00101010100111100111101000000101100011100010011100001000100101011000100001  
01111101111001010110110100110011010001111110100011010101110011100111010101010  
01001000010110110001010011011110110111101010111110101000011101000011010000000  
00000001100011000011100011001101001100010110100111100110101011100011000000001  
01000101100011001110110001001011100001011011100100101000110011101111000110100  
00011111000100100111011001101001100111011011101001101101000001100100000101010  
1000100011000100101001110011110000011110111000111110100101010110011101111010  
10001100010111011110001000100101000010111000111110111011100001110100111000010  
00101000011100010100111010100001101010011001000101011100010110010000010011111  
11000100011011010101101100110000010000000011110011001111101000010100011001111  
11100011010100101011110101110111010001110000111111111100010110111000111101100  
10101100001011000000100001011110101110100111001000101000111110000011100011010  
01101100011010011000010101010001110011100110011111111000111010010011101111011  
0010110110110110111011011100010101110101101101000101101000000010010111111010  
10011000111100001001001000001110100100000000000010010001111010111100100001010  
0111010011001101000100101001011011110100001010111001110000100101001010111010  
10101011001111101000110001001000011010100010100111010001011100000101110110011  
110000110001001011111100010101010110000110011111101001011101011010110101110  
11010000111000110111100101111001110111101111000001010010111011010001110110101  
011011001111001011110010001010101011101011100100011101011100010110101001000  
10111010100111111001100011011110101110000110100110100001100100101100111000100  
11011100101100100100001111000011001000100001101101011101001011001111000011101  
1101001101100100001110000101101101111101101111101100111101011011011010000111  
0100010010000111011000111111110011110001011001001111010111100111100111111011  
10001100011011010000001111101010110011111001110001110100000101010001010101111  
11111100100111100010001001111000011000111100011011101111101001100110100100110  
00101110100001111001110001110011100001100000010111101001010110100011011000100  
00100010111110111010111100001100001111011101111101011100100000101101100010101  
0100011111000111000000100101001100000011101001111101011111011010110110000

10110101010001100001010110001000101011110011000001010010101010100111101011000  
10110000000100000110110000011100000010100101111001100011100011010011000110011  
11111000010001101001001010001000000010010100000001111111101000011000111101000  
10111000100010111001001011001111000110010111101110110100110011011111000010001  
01000100011110001100000000101001100100111111011110011010010000110100001001011  
11001110101010010001001001000011110011110001101101011010111111001011100011000  
01001001110000011111100111000010111100011010011011000000111100001010001101110  
111110100001000000010100000000100001010000010010101001110110010100011101

```
In [57]: b = BinarySymmetricChanel(a)
print(b)
```

010100111110010101001100101001111011011111000000111111001011011100110011100  
01111100001001111011010110101101100111001110111010000001110100011010100000000  
1000101010000011111101011101101101100100110011000000110001000001001101100001  
0011110101100000000000111011110001111000111110011100010100001111000011000101  
1110110100010000111111110001010100011011001101111011011010111001100101000101  
11011100000101100111111011000101011001101110011100100111110100011100111011010  
10011001101110010011001000100111000100100100011010001001000101011001110111010  
00100011000001100001100010001100111101000110100000001011001011101010001010001  
10111101001000101000000111010000001101000011100011001110101101101010000100010  
1011100010001001110110100001001110000001000110000100000001001010100111111100  
10011011010100110100111110001111100111000001110010001011001000101111010110100  
001100111111110101110100111100010111111101111010111000111010101101010111  
01010101011100100001010001001011111011010100100001000011110000000010100110100  
10010111010010011110111111111111000111011011110101011011000100101101100100  
10101000110101010010101101001011010010010100000110001001010111011100010010010  
00011110010110100100110011111011100110001111111011001111011001100010011001010  
0111111011100011000111011010010111010100011111101100000101001100011110010101  
1000001101111100011110111000110011101110010111001001100110111111011110011  
110110011011000101101101110101100001000001101110000011001101001011100101010  
001110010110000001110101010100111001110011011010100011010111110100100000100  
10001011100010101001000101100001000010100000100111100000111101101010010011000  
10010111110100011000100010001011101100000000100000101000101111001010001101101  
0000111011101101000100111000110100001001000111000011110010010111000110101111  
001100111101010010101001111011110011111111111100100010110000010001111011011011  
10100010111000110011000011110001100110110100101110011001101100001110110110001  
10100000011100000111101101111100101100000111000010101011101100100110111010100  
1100011000011000100011011111011100011101010110001011000111011000111010000100  
10000110011100111110110010100011100011110101111100001000110001001010110110000  
00111010100101101111101000000001101011100010011100100000100000101011000101001  
0111110111100111011011110011001101100111110010001101110111001110111101010101  
01010101000110100001010011011110111111101010111110101000011101000011010000001  
00000101000011000011100011000101101101000110100111100110101000100000000000100  
01000100110011001010110001001011100001011011100100101000110011101011100110100  
01011111010100100111010000000001100111011011101001101101000101100100001001010  
10001000110001001010010100111100010111001110001111101001010101100111011110010  
10001100010111011110001000100101000110111000101110111010100011100110111001011  
10101000111100010100011010100001101010011011000111011110010110010000010011111  
11000100011011010100101100110000000000000011110011001111100000010100011001111  
11000011100000101001110101110111110001100010111110111100010110111000111101000  
11001100011011000010100001000111101100100011001000101000111100000001000010111  
0110110001101011000001011001000111101111010011111111001111010010011101101011  
00001101101111101110110001001101011101011011010001011010000011000110111111010  
10011001111000001001001001001110100100000000000011010001111010101100010001010  
0111010111010100000110101001011011010100001110111001110001000101001010111010  
10101010101011101100110001001000011110100010100111010001011100100101110110001  
11000011000100101111100001010101011000011101111110110101010101101110110101110  
100100000110001101111001011110011101111101111000001010010111110010001110110111  
00101100101100101111000000101010000101010111000101011101011000010110101001000  
10011010000011111001100010011110101110010110100100110011100100101100111010100  
11011100101100100100001111000111000000011001101101011111001011001110001011100  
1100001101100000001110000011101101111101101111101000101101011011010011000111  
01100100100001110010001111111000010100010110010011110101111001111011111111011  
10000111011011010000001111101010100011101001110001110100000101011001010101110  
11111100100111000010000101111000011000111110011011001111101000100110100100110  
00101100100001111101110001110011110001100000110011101001010111100011001100100  
10110010011100011010011100001100001111011101111101011000000000101101100010101  
01000111110001110000010010100100000001101001100100101111110110111010110000



```

11110101001001100001010110101100101011010011000001110010001000110111101011100
10110010000100000110110000011100100110010101111011100011100011010011000110011
11111000010001111001001010001000000110010100000111110111101010011000111111100
1011100010001111100100101000111100011101011110111101110010011110010010001
01000100011111001100000100101001100110111101011110010010010000111100001010011
11000100001110011001001101000010100011110001101111010010101111001001100011000
01001001110000111110100011000010111100010000011011000000101100001010001101110
111110100001101000010100000000100001011000010010101001110110010100111111

```

```

In [76]: #compare a and b
a_list = []
b_list = []

for ch in a:
    a_list.append(int(ch))
for ch in b:
    b_list.append(int(ch))

erroneous = 0
matches = np.logical_xor(a_list, b_list)
for i in matches:
    if i:
        erroneous += 1
print(erroneous/5000) #should be around 10%

0.0994

```

```

In [102]: def calc_erroneous(a,b):
    a_list = []
    b_list = []

    for ch in a:
        a_list.append(int(ch))
    for ch in b:
        b_list.append(int(ch))

    erroneous = 0
    matches = np.logical_xor(a_list, b_list)
    for i in matches:
        if i:
            erroneous +=1
    return erroneous/5000

```

```
In [156]: a = ""
for i in range(1000):

    #create a word
    b = ""
    for j in range(5):
        b = b + str(random.randint(0,1))

    #for each word, repeat each character
    c = ""
    for x in range(len(b)):
        c = c + b[x]+b[x]+b[x]

    a = a + c
print(len(a))
print(a)
```

[illegible]

---

[illegible]

[illegible]

[illegible]

```
In [157]: b = BinarySymmetricChanel(a)
          print(b)
```

[illegible]



1111111100011111111000111111000111111100011100011111100000101100000011111  
100011100011111111100100111001101000000000111000000000000110000110000001111  
01000000011000000111111000111101000011011000011000000110011000111000010000000  
11100000011100000110010111100010000111010011100011100010000011111100000000001  
111111111101111111000111000001000111111000000111111111011001111110000000000  
001111110111111111011010000000001111100111101111100011100011111011101000000  
10100000000011011100011110110110100011111111000111111111111110011100010011  
11110000000011111101011100001111111000101000000000010110001010101111110000  
001110000001110001111110011111111000011110001111111100000011111111000111111  
01010100011100011111100011111111000000001111000000000011100100000000001100  
00001000000000000000000111111111111111111001010000001111001111110001111010  
0000011100000000011101111100100011001000010110110011100011111000000000011111  
000000010111111111111100010101111000000011100011111100010000011110100000000  
0000100111110000010101000111101000000110100000111111001001110111000000000000  
001111111110101000000111110000000000000000000111101000111001101101111000111000  
010000000000011110011111101000111011001111000000111101100000111101111011111  
10000110001100011111011111111111111000111001001000000111000111101111111111  
111011111010101001110000001110000101101111011100111100011110001011011111100  
000111111111000111111111110001000100000001110001111100011110011000011100010  
111100000000010111110110111011100000011111111010000001111011100110000111110  
11000111110000111010000111001111000001000110000111000010000000000110100111000  
000111111000000100000111001111000000000111101110111000111100010000000000111  
1111000000111000111000100111100101000111111110000001111110001110000001110000  
0011100000011111111001111111111111011101110001111110011111100000000110010  
11100001100011111111100111000111111101110000000000000000011001100011100000  
00110001110001111111000010011100000011111100010111000111000010111011111011  
0000100111100001000001111100011100001100000011110101000000011000111000101000  
001000010000111111101111111000011100011000000010100010100001011111101100100  
011111110111000000111111110001111110000100000001100110000001010010000001110  
0010000011110110011001011100010111111000000000000000111111000011011000111111  
10011100111110111111111111010001110000101111110000001111111000000000011110000  
10000001110000011001110001111110010000001111110100001111011110000000000001100  
0011111111111111010000101000000111111100000000000000011111011111111000111110  
01000000010001111111100001000011101111100011100011100000011000001100001  
1111000111111111000000111111000111000111000000111110001110001111010001110000  
0010111011101011110000011100011101111111000000010000111011101111000010000000  
0000010001110000001110001111111101010000101110001011110110101111111111111110  
1001111111000000000000111000000111111100111111000111110101001100011111  
01100101000000111111001001000011111101000111100111010010110101000100000101000  
111000000000000000000001001111111000111000010000001000000111111101111111111111  
1100111000001000111111101110001110001111111111110111111101111001100111111110  
1111110100011111110000000011000011110001111111011111000111000000001001100000  
101111111111000110001000101011111100101011100000000000000001011100000111100000  
00000000000000010001110111100010011111110111110011111011100000000000000001  
1111000011111111111100100011110100011111100010111111000111010111000000000100  
11001011101111111111111100011100011100011100011100000011100010000011100011100  
01100001111111111111010001010000110111111100000111111111111110001111110000101  
1100000011100111110000011100000111111000011011111000000000000100101000000111  
110000010000001001000111000111000000111110000000101110001111111111111100011  
11110000101111110001111110000001001110001111001010110000000001111010110001001  
11101000000000000000000011100010000000011111011101101111100001000111000111000  
00000100000011100011110100011110110111100010100010100001111111111110010100001  
00001100001111110111000000001101010000000000000000001111110111110001000001110  
00101000000101110101001111010100001111000000000000111011110111110000111000111  
01011100100100000011100011100011100010111100011111111111110111101011110011100  
000000000000100110100000001111011110001111100000001110101111111111110000110000  
00000111000000011110001000011111100000011111111000111101101111110010100001

0011111011111100000000000011100001100011111100010011100000000011100000011111  
1111000000001111001001000000000000000010001110000001110001110001110001000001  
00111111111011001000110000011111000111000000111110000001111111111111111000111  
11111111111111000011111111110100001010000011111101000000000000000011100011  
10000100001110001010000000000001110100001110000000000111110000001110000010000  
010111111011111100011011100000000011011110111111000111010111000000111111000  
0001110000001110110100100001111110000001001101010000000000000000000011100001000  
001010000011000000011111111000111111111111111110000011110000001110000000001  
0100000011111111111111010111000101000000111001000111000111001111000000000000  
11100110011100011100011100010000011100001011100100011111100000010100011100000  
00001010001111110000110001111110110000000001110001110011110000110101111000111  
1111100001000001001101100001100000010111111111111111111111100101111000000101111  
011000111100111000001111111010011111011000000000000111111111100011111000011  
1000010111000001100111001000000111000111111100001111110101101110001111110001  
10001000100111111111000010111011100110100000000111110101000000111111000001111  
00000000011111101100001111100011100010100000010111011111100000001100000000011  
111100000000011000000000111000001000101111000000000011110001110001111111110  
0000011100011000011100110011001100001010100000111000000000111000010110111000  
0000001111111110001110001011110001100001111111100011111000000011100011100011  
10001100111110000000000011100000000111111111000000001110001000000000110000  
0011100011100010100011100111111001111111010111111111000000111000111000000111  
00111011101111110010000000000110011111100000100011000000011111100011100011110  
0100101111011111111111111111111100011000011111100100101000000000111111111101111  
1110010111000000011111010111010111100011111110000111111000111010000111000000  
0000010001001110001110011110000000001111100001000001100011011111111100000000  
0000000000000101110001000000000000000001110001110001111011111111111110101111  
1111111111111100000111010000011100000100111111100111111100000011111101111111  
111111000111000000111111000000111111001110000001100001111001111110111111111  
100000011100000000110000001111111011110001100000100001111100001111000011000  
000100111110001001111111111111011011100000011101111000111111000001110110111  
1110001110000001100000110101110001110101111110101111000100001000011101100011  
0000111001000111101001001000111111001111111110001110001111010001111110001110  
00101000000100111000001111000000111110001110111000000000000000000011110111  
111111111111111111111100011100000011111111110010001110000010000000000000000  
01111000000001110011110001110001110001110000001000001111110001111010111110001  
0111100000010110000011101000000011010000101000110111000111000000000000000110  
11001111100000011101111100010011111101001011111111100010000011001110100011100  
0000000111101000000100000000011111110001000010001001010000100000001110000000  
00000111000111000000111010110000110000000111111000010000111000111111111100111  
00000011100000011011100000011100010110111100000011100011110111100011100000001  
1111111000111111000000111111110001111001001110001001110000110110110001100011  
11111111111001111000111000111111000000111111000000000000000000000000011101101  
11110010011111100011100011100111101100100000010000000011101100000000000000011  
1101111000111010111101001100100101111011111111111010000001111101011111111  
1100000011110111111111100000000011111111100011111111000111000111000000111111  
1001110000001111000001110001110000001111111110100000001110001111111110000000  
01110001100000001111010001110000111111001111110000000000000001111110000001000  
00110111101000011001111011000000111000111111000000001111110001111101111000101  
01000000000000000000001110000101110100001110000001001110000011111111110100111  
1000110000111111111111000111111110000110000001110001111000101101110000000001  
01111111111101000000000111010000111000000111001111000000100000000110110111000  
1010001100000001100010100000000000000010011111110001110000000001111111111100  
01111000001110000001101010000000000000000110000000001001111110100110001110000  
0000011100011110011111100011100111111111000000001111000111011111111000000000  
11100011100111110000010011100000000011100011111100001000011111111111011101011  
1100011111010111000111111111000111111101011111000000001000001111101000010000  
11100000000000000010011100011100011111010111111001111111011111011110111000001

```

00011100011100000000011100000011111100011100001000111100011111100010010011110
10001001111111111110000001110000000110000000110000110000111010100010100001111
1000000011111111010011100011100010010110000011101011111100000001011111011000
0001110010000000000001010010010001001110001001011110000000000100000101111101
00000000000001110110001001111100000001101110001001110100011110001101110001111
11011000000111111000000000001111000000000111000101000111010011001000100010010
00010000011111111100000011011101000000000000011100001011110000010011111111111
0101000000111010111000111000001000111101010100101111111110000001011111110000
001100001110001110001110001010001111110000001111110111000111111011111011000
00011011100011111101111000000010000000111111110100010001111111100001111100
0000000010010100000010000000001000000111111011000111111110111111000000101111
11111100111001010000000000001000000111000111000011101000111010000110000111011
1111000000101111111010001110001011110101111110000110011110001111110001111111
10000000001111000001110001010000111111110000000000000001100101010001011111110
00111101000110011111000111111111111011011110111000000110010010000110111111010
11100000100011100000000000011000000111111100011000001100000000011000010100000
0000000001110000100000111000000111111111100011100011111111110001111010000
011111111000000000000000000000011100000111110100011111101100000000011100011111000
11111111100011101100110001100000011100011111100000011100011001000100011100000
011100011101011011100011100011000011100011111111111111111001110001101101111001
11111000101111011111000111111111110000001111100001111110011111000000110001111
1000001110000001111100001110001111110001000000001110001110001111111111111101
11011111111111110001110001011111110011111110000100000001110000111010101110111
10000000000010111100011111000111111111111111111111111111110001001111000

```

```

In [158]: def decode(bitstring, step):
            decoded = ""
            for i in range(0, len(bitstring)-2, step):
                if sum([int(x) for x in bitstring[i:i+step]]) > (step//2): #bit is a 1
                    decoded = decoded + "1"
                else:
                    decoded = decoded + "0"
            return decoded

```

```

In [159]: c = decode(a,3)
            d = decode(b,3)
            print(calc_erroneous(c,d))

```

0.0296

```
In [139]: a = ""
for i in range(1000):

    #create a word
    b = ""
    for j in range(5):
        b = b + str(random.randint(0,1))

    #for each word, repeat each character
    c = ""
    for x in range(len(b)):
        c = c + b[x]*5

    a = a + c

b = BinarySymmetricChanel(a)
c = decode(a,5)
d = decode(b,5)
print(calc_erroneous(c,d))
```

0.0086

```
In [146]: a = ""
for i in range(1000):

    #create a word
    b = ""
    for j in range(5):
        b = b + str(random.randint(0,1))

    #for each word, repeat each character
    c = ""
    for x in range(len(b)):
        c = c + b[x]*11

    a = a + c

b = BinarySymmetricChanel(a)
c = decode(a,11)
d = decode(b,11)
print(calc_erroneous(c,d))
```

0.0002

```
In [148]: a = ""
          for i in range(1000):

              #create a word
              b = ""
              for j in range(5):
                  b = b + str(random.randint(0,1))

              #for each word, repeat each character
              c = ""
              for x in range(len(b)):
                  c = c + b[x]*23

              a = a + c

          b = BinarySymmetricChanel(a)
          c = decode(a,23)
          d = decode(b,23)
          print(calc_erroneous(c,d))
```

0.0

```
In [161]: def com_rate(size, n):
          return math.log(size,2)/n
```

```
In [163]: print(com_rate(3,15000))
          print(com_rate(5,25000))
          print(com_rate(11,55000))
          print(com_rate(23,115000))
```

0.00010566416671474376  
9.287712379549448e-05  
6.289875670249632e-05  
3.93353213570175e-05

As we can see, increasing the number of repetitions decreases the rate of communication (speed reduces). But as we can also see from above, the probability of errors goes down as well.