Project 2 - Conditional Probabilities

California State University Long Beach

EE 381

Probability and Statistics

Christopher Masferrer

T/TH 5:30 - 6:20

Experiment 1: Probability of erroneous transmission

Intro: For this experiment I will be calculating the probability that a transmission error from a one bit message will occur. The results of the probability will be displayed in a chart.

Methodology: Using the nSidedDice function from a previous lab, I used it as a generator for a new function called transmissionError. I generated one number for S and based on the result from S generated a value for R. I then compared the values and recorded the result only if the transmission failed. This repeats 100,000 times and the results are displayed in the chart.

Results:

Probability of transmission error	
Ans.	**p = 0.94814**

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 20 18:13:41 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 1
import numpy as np

p0 = 0.35
e0 = 0.04
e1 = 0.07

def nSidedDie(p):
    N = 1
    n = np.size(p)
    cs = np.cumsum(p)
    cp = np.append(0,cs)

    for j in range(0, N):
        r = np.random.rand()
        for k in range (0, n):
            if r > cp[k] and r <= cp[k + 1]:
                d = k+1
        return d

def transmissionError():
    N = 100000
    count = 0
    for j in range(0, N):
        S = nSidedDie([p0, 1 - p0])
        if S == 1:
            R = nSidedDie([e1, 1 - e1])
        else:
            R = nSidedDie([1 - e0, e0])

        if S != R:
            count += 1
    prob = count / N
    print("Probability of an error: " + str(prob))
```

transmissionError()

Experiment 2: Conditional probability: P(R=1|S=1)

Intro: For this experiment, I will be finding out what the probability of getting a successful transmission is knowing that S = 1. I will be performing this 100,000 times and displaying the result on a chart.

Methodology: I used most of the code from the previous part with a few modifications made. I gave S a value of 1 and only generated a value for R. I only counted if the transmission was a success and printed out the results.

Results:

Conditional probability P(R = 1|S = 1)	
Ans.	**p = 0.06841**

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 21 08:42:59 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 1
import numpy as np

p0 = 0.35
e0 = 0.04
e1 = 0.07

def nSidedDie(p):
    N = 1
    n = np.size(p)
    cs = np.cumsum(p)
    cp = np.append(0,cs)

    for j in range(0, N):
        r = np.random.rand()
        for k in range (0, n):
            if r > cp[k] and r <= cp[k + 1]:
                d = k+1
        return d

def probabilityRS():
    N = 100000
    count = 0
    for j in range(0, N):
        S = 1
        R = nSidedDie([e1, 1 - e1])

        if S == R:
            count += 1
    prob = count / N
    print("Conditional probability P(R=1|S=1): " + str(prob))

probabilityRS()
```

Experiment 3: Conditional probability: P(S=1|R=1)

Intro: For this experiment, I will be finding out what the probability of getting a successful transmission is knowing that R = 1. I will be performing this 100,000 times and displaying the result on a chart.

Methodology: I used most of the code from the previous part with a few modifications made. I gave R a value of 1 and only generated a value for S. I only counted if the transmission was a success and printed out the results.

Results:

Conditional probability P(S = 1|R = 1)	
Ans.	**p = 0.35078**

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 22 09:36:28 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 1
import numpy as np

p0 = 0.35
e0 = 0.04
e1 = 0.07

def nSidedDie(p):
    N = 1
    n = np.size(p)
    cs = np.cumsum(p)
    cp = np.append(0,cs)

    for j in range(0, N):
        r = np.random.rand()
        for k in range (0, n):
            if r > cp[k] and r <= cp[k + 1]:
                d = k+1
        return d

def probabilitySR():
    N = 100000
    count = 0
    for j in range(0, N):
        S = nSidedDie([p0, 1 - p0])
        R = 1

        if S == R:
            count += 1
    prob = count / N
    print("Conditional probability P(S=1|R=1): " + str(prob))

probabilitySR()
```

Experiment 4: Enhanced transmission method

Intro: For this experiment I will be calculating the probability that a transmission error from a one bit message will occur if the signal S is sent 3 times. The results of the probability will be displayed in a chart.

Methodology: Using the nSidedDice function from a previous lab, I used it as a generator for a new function called eTransmission. I generated one number for S and based on the result from S generated a value for R1, R2, and R3. I then compared the values and verified what the majority was using the majority rule. I set a value for R depending on what the majority was and recorded the result only if the transmission failed. This repeats 100,000 times and the results are displayed in the chart.

Results:

Probability of error with enhanced transmission	
Ans.	**p = 0.99992**

Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 26 10:25:18 2019

@author: christophermasferrer
"""

#Christopher Masferrer
#EE 381
#Lab 1
import numpy as np

p0 = 0.35
e0 = 0.04
e1 = 0.07

def nSidedDie(p):
    N = 1
    n = np.size(p)
    cs = np.cumsum(p)
    cp = np.append(0,cs)

    for j in range(0, N):
        r = np.random.rand()
        for k in range (0, n):
            if r > cp[k] and r <= cp[k + 1]:
                d = k+1
        return d

def eTransmission():
    N = 100000
    count = 0
    for j in range(0, N):
        S = nSidedDie([p0, 1 - p0])
        R = 0
        if S == 1:
            R1 = nSidedDie([e1, 1 - e1])
            R2 = nSidedDie([e1, 1 - e1])
            R3 = nSidedDie([e1, 1 - e1])
        else:
            R1 = nSidedDie([1 - e0, e0])
            R2 = nSidedDie([1 - e0, e0])
            R3 = nSidedDie([1 - e0, e0])
```

```
        if R1 == 0 and R2 == 1 and R3 == 1:
            R = 1

        if R1 == 1 and R2 == 0 and R3 == 1:
            R = 1

        if R1 == 1 and R2 == 1 and R3 == 0:
            R = 1

        if R1 == 1 and R2 == 1 and R3 == 1:
            R = 1

        if S != R:
            count += 1
    prob = count / N
    print("Probability of an error: " + str(prob))

eTransmission()
```