EE 381 Lab 2 Report

Binomial Coefficient Experiment

Daniel Duong and Long Nguyen

Instructor: Dr. Duc Tran

Due 10/15/20

Table Of Contents

Introduction

The purpose of this lab is to determine the probability of run simulations in regards to determining the probability using combinations, and permutations. For the first problem, we generated different arrays of people and we determined the probability of which groups of people support Party A, Party B, or Party C. For the second problem, we first generated a list of children which contains boys and girls. We then determined the probability of having the same number of boys and number of girls. For the third problem, we generated two lists with random numbers which are the list of winning lottery numbers and the list of numbers that you would get when you bought a lottery ticket. We determined the probability of winning the lottery.

Procedure:

1. **<u>Question 1:</u>**
    a. We generated a random list of four elements with the range of the elements between 0 and 1000.
    b. We create another array of size 3 to count who which person supports what group from the random array with index 0 represent the number of people support party A and 1 support party B and 2 support party C
    c. Then we iterate the random people list, if the element in the random person array is larger than 801 that person would be supporting party C, if larger than 501 that person would be supporting B and party A would be from the range 0 - 500. Increase the element inside the party counter array by 1 accordingly
    d. Iterate through the party counter list after the people in the random people list has been counted to each party, if we encounter a 4 that means all 4 people support

one party. Increase the counter for occurrences of all four people support the same party by 1

e. We then iterate the step “a” to “d” 1 million time to simulate 1 million experiments

2. **Question 2:**

    a. We generated a random list of 40 children when n = 10 and in the second scenario, we generated a random list of 200 children when n = 50. We always have a class of 4n children in any scenario.

    b. The generated random numbers were assigned to each children without repetition.

    c. For 10000 experiments, we iterated through the list of children to see if they were either a boy or a girl.

    d. We represented the even numbers as boys and the odd numbers as girls.

    e. If the assigned value of the child is even, we increment the number of boys by 1. Otherwise, we increment the number of girls by 1.

    f. Then we checked if the number of boys is equal to the number of girls.

    g. If the number of boys is equal to the number of girls, then we increment the counter by 1.

    h. To find the probability of having the same number of boys and girls, we divided the counter by the number of experiments

3. **Question 3:**

    a. We generated a list of random winning numbers and a list of random numbers that you would get when buying a lottery ticket.

b. For 1 million experiments, we iterated through the entire list of your random numbers and checked if any of your random numbers matches the list of winning numbers.
c. For every of your numbers that match the numbers on the winning lottery, we increase the number of matches by 1.
d. If the number of matches is equal to 4: then you increment the number of times that you won the lottery by 1.
e. To find the probability of winning the lottery, we divide the number of times that you won the lottery by the number of experiments.

Results and Discussions:

Lab 2 Q1 Trials:

Theoretical Results: A is 0.062125, B is 0.00798, C is 0.00156

1.Probability of all support A: 0.062654

Probability of all support B: 0.008162

Probability of all support C: 0.001554

2. Probability of all support A: 0.062618

Probability of all support B: 0.00793

Probability of all support C: 0.001597

3. Probability of all support A: 0.062441

Probability of all support B: 0.008087

Probability of all support C: 0.001616

Lab 2 Q2 Trials:

Part 1: n = 10

Theoretical Result: 0.24763

1. Number of Expeiremnts is: 100000

The probability of getting an equal number of girls is: 0.2457

2. Number of Expeiremnts is: 100000

The probability of getting an equal number of girls is: 0.24806

3. Number of Expeiremnts is: 1000000

The probability of getting an equal number of girls is: 0.24763

Part 2: n = 50

Theoretical Result: 0.11242

1. Number of Expeiremnts is: 100000

The probability of getting an equal number of girls is: 0.11199

2. Number of Expeiremnts is: 100000

The probability of getting an equal number of girls is: 0.11282

3. Number of Expeiremnts is: 100000

The probability of getting an equal number of girls is: 0.11369

Lab 2 Q3 Trials:

Theoretical Result: 0.0002064

1. The probability of winning the lottery is: 0.00021

2. The probability of winning the lottery is: 0.000199

3. The probability of winning the lottery is: 0.000189

Conclusion:

We learned that there was a new function called random.sample(range of the elements, number of elements).. This automatically returns a list without repetition. We discover that the probability of winning the lottery is very low in reality. We were able to use the classical approach to determine the probability of getting the same numbers of boys and girls, the probability of winning the lottery, and the probability of people supporting different parties. We learnt to use ranges appropriately in order to represent different parties. We overcome the challenges determining boys and girls by assigning even and odd values to boys and girls. We were trying to find a way to make a list without repetition because using a for loop with the random.randint() function returns a list with duplicates so we had to research on how to generate a list without duplicates. For question 1 we check if the people were supporting the most exclusive group first which is party C in our case with the range from 801 - 1000 followed by the second group which was party B with the range from 501 - 800 which helps us determine which person is supporting which party. We applied the knowledge where no voters vote more than once for the presidential election; thus, we thought that no person cannot support more than one party. We assume for question 3, the lottery ticket and the chosen numbers does not have duplicate numbers. But in reality, there can be duplicates for the lottery tickets.

Appendix

Lab2Q1.py

```
#Authors: Daniel Duong and Long Nguyen
from math import factorial
import random
'''
Here the combination formula without repetition is: C(n, r) = n! / (n - r)! r!

translated into python: math.factorial(n) / math.factorial(n - r) * math.factorial(r)

Permutation Formula: P(n, r) = n! / (n - r)!

Ex: 1 #x = factorial(6)
#print("The factorial of 6 is: ", x)

this prints 24.

Ex: 2 We are taking the C(5, 3) = 10
y = factorial(5) / (factorial(5 - 3) * factorial(3))

print(y)

this prints 10.

'''


numPeople = 1000
numExperiments = 1000000

partyA = 0
```

```
partyB = 501
partyC = 801

randomNum = 0

counter = 0
sameParty = [0, 0, 0]       #Keep the total number all four support same party
                        # party A is index 0, party B is index 1, C is index 2
for i in range(numExperiments):
    peopleList = [] # for every experiment, you reset the list of people and the same party list
    partyCounter = [0, 0, 0] #index 0 is party A, index 1 is party B and index 2 is party C

    for j in range(4): #generating 4 random numbers to have a group of 4 random people
        randomNum = random.randint(1, numPeople)
        #print(randomNum)
        peopleList.append(randomNum)

    for k in range(len(peopleList)):

        if peopleList[k] >= partyC:
            partyCounter[2] += 1 # you want to index the list where you keep of which people
support party C
            #print("supports party C")

        elif peopleList[k] >= partyB:
            partyCounter[1] += 1 # you want to index the list where you keep of which people
support party C
            #print("supports party B")

        else:
            partyCounter[0] += 1 # you want to index the list where you keep of which people
support party C
            #print("supports party A")

    for l in range(len(partyCounter)):
        if partyCounter[l] == 4:
            sameParty[l] += 1

    # print(partyCounter)
    # print(peopleList)
    # print()

print("Same party list: ", sameParty)
print("Probability of all support A: ", sameParty[0] / numExperiments)
print("Probability of all support B: ", sameParty[1] / numExperiments)
print("Probability of all support C: ", sameParty[2] / numExperiments)
```

```
#print("List: ", peopleList)
#print(len(peopleList))
```

Lab2Q2.py

```
#Authors: Daniel Duong and Long Nguyen

#Professors Result: 0.24763

import random

numExperiments = 100000

'''random.sample is a new module for the probability function
2nd parameter is the number of children you want to include
and the 2nd parameter is to return the number of random elements
will return a list
'''
children = random.sample(range(40), 20)
#print(children)
#print(len(children))

'''
odds and evens

ODDS - Girls
EVENS - Boys
'''

counter = 0

for i in range(numExperiments):
    randomList = random.sample(range(40), 20)

    numBoys = 0
    numGirls = 0

    for j in range(len(randomList)):
        if randomList[j] % 2 == 0: #Even numbers are boys
            numBoys += 1

        elif randomList[j] % 2 != 0: #odds are girls
            numGirls += 1
```

```
    if numBoys == numGirls:
        counter += 1

    #print(randomList)
    #print("Number of Boys is: ",  numBoys)
    #print("Number of Girls is: ",  numGirls)
    #print("Counter: ", counter)
    #print()

print("Number of Expeiremnts is: ", numExperiments)
print("The probability of getting an equal number of girls is: ", counter / numExperiments)
print()

counter2 = 0

for k in range(numExperiments):
    randomList = random.sample(range(200), 100)

    numBoys = 0
    numGirls = 0

    for l in range(len(randomList)):
        if randomList[l] % 2 == 0: #Even numbers are boys
            numBoys += 1

        elif randomList[l] % 2 != 0: #odds are girls
            numGirls += 1

    if numBoys == numGirls:
        counter2 += 1

    #print(randomList)
    #print("Number of Boys is: ",  numBoys)
    #print("Number of Girls is: ",  numGirls)
    #print("Counter: ", counter)
    #print()

print("Number of Expeiremnts is: ", numExperiments)
print("The probability of getting an equal number of girls is: ", counter2 / numExperiments)


#totalChildren = set() # empty set

#totalChildren.append(children)
#for i in range(20):
    #children = random.sample(0, 39)
```

```
    #totalChildren.add(children)

#print(totalChildren)
#print(len(totalChildren))
```

Lab2Q3.py

```
#Authors: Daniel Duong and Long Nguyen
''' There is no replacements at all '''
# Theoretical Result: 0.0002064

import random

numExperiments = 1_000_000

#for i in range(20):
    #print("i is: ", i)

numWinLottery = 0

for i in range(numExperiments):
    counterLottery = 0 # if you declare counterLottery when comparing every index, then you will
reset it every time you compare your numbers to the winning numbers
    myNumList = random.sample(range(1, 21), 4)
    #print("This is my numbers for the lottery: ", myNumList)

    drawingList = random.sample(range(1, 21), 4)
    #print("Winning lottery list: ", drawingList)

    for j in range(len(myNumList)):

        if myNumList[j] in drawingList:
            counterLottery += 1

    print("Number of matches found: ", counterLottery)
    if counterLottery == 4:
        numWinLottery += 1

    print("Won lottery: ", numWinLottery, "\n")

print("The probability of winning the lottery is: ", numWinLottery / numExperiments)
```