

Professor: *Duc Tran*

Team members: *Alyssa Faiferlick – Thanh Nguyen*

Class: *EE 381 – Section 13*

Due date: *10/15/2020*

**Project 02**

**BINOMIAL COEFFICIENT**

EE 381 – Probability and Statistic Computing

*CALIFORNIA STATE UNIVERSITY LONG BEACH*

# **Question 1 – GROUp Support**

1. INTRODUCTION:

There are 1000 people support three different parties. 500 of them support party A, 300 of them support party B, and 200 of them support party C. Choose randomly a group of four people from the population. This program is written to calculate the probability that all 4 people in the chosen group support either party A, or B, or C.

1. PROCEDURE:

* Use a for loop from 1 to N (we choose N = 1,000,000 – the number of experiments)
* For each experiment:

+ Create a sample space from 1 to 1000

+ Choose random 4 numbers from the sample space by random.sample() function.

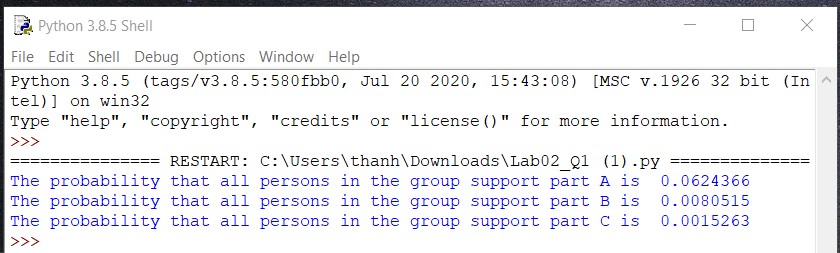
+ For each number, we assign that a number from 1 to 500 will support party A, a number from 501 to 800 supports party B, and the number from 801 to 1000 will support party C.

+ Check that if all 4 numbers just support one party, then increase the counter for each party by 1.

* After N experiments, divide the counter of each party to N, we will have the probability that all 4 people in the chosen group support each party.

1. RESULTS AND DISCUSSION:

* Results:



* The results are what us expected them to be. As we calculated the probability of these parties in class, the probability that all persons in the group support party A is for party B, and for party C. Therefore, our results are quite close to these numbers.

1. CONCLUSION:

* For this question, we used probability and combinations from the lecture to calculate exactly what the probability that all 4 people in the chosen group support each party.
* At the beginning, we had to read some materials to find out how to get the number randomly without duplicates in python. We changed from random.randint() function to random.sample() because we figured out that there is a chance to get a duplicate random number in a list if we use random.randint(). Therefore, random.sample() is the solution since it never repeats the element, so that, we can get a list of random numbers without duplicates.

1. APPENDIX:
2. """
3. Class: EE 381 - Section 13
4. Project 02 - Binomial Coefficient \_Part 01
5. """
6. **import** random
7. """
8. A certain population consists of N=1000 people. 500 of them support party A; 300 of them
9. support party B; and 200 support party C. A group of 4 people is chosen at random from the
10. population. What is the probability that all persons in the group support of party A, party B,
11. and party C.
12. """
13. **def** findFourPeopleParty**():**
14. fourPeople **=** random**.**sample**(range(**1**,** 1001**),** 4**)**
15. fourPeopleParty **=** **[]**
16. **for** position **in** **range(len(**fourPeople**)):**
17. **if** 1 **<=** fourPeople**[**position**]** **<=** 500**:**
18. fourPeopleParty**.**append**(**'A'**)**
19. **elif** 501 **<=** fourPeople**[**position**]** **<=** 800**:**
20. fourPeopleParty**.**append**(**'B'**)**
21. **elif** 801 **<=** fourPeople**[**position**]** **<=** 1000**:**
22. fourPeopleParty**.**append**(**'C'**)**
23. **return** fourPeopleParty
24. **def** allPersonsInGroup**(**N**):**
25. countA **=** 0
26. countB **=** 0
27. countC **=** 0
28. **for** i **in** **range(**0**,** N**):**
29. fourPeopleParty **=** findFourPeopleParty**()**
30. **if** fourPeopleParty**[**0**]** **==** 'A' **and** fourPeopleParty**[**1**]** **==** 'A' **and** fourPeopleParty**[**2**]** **==** 'A' **and** fourPeopleParty**[**3**]** **==** 'A'**:**
31. countA **+=** 1
32. **elif** fourPeopleParty**[**0**]** **==** 'B' **and** fourPeopleParty**[**1**]** **==** 'B' **and** fourPeopleParty**[**2**]** **==** 'B' **and** fourPeopleParty**[**3**]** **==** 'B'**:**
33. countB **+=** 1
34. **elif** fourPeopleParty**[**0**]** **==** 'C' **and** fourPeopleParty**[**1**]** **==** 'C' **and** fourPeopleParty**[**2**]** **==** 'C' **and** fourPeopleParty**[**3**]** **==** 'C'**:**
35. countC **+=** 1
36. **print(**"The probability that all persons in the group support part A is "**,** countA **/** N**)**
37. **print(**"The probability that all persons in the group support part B is "**,** countB **/** N**)**
38. **print(**"The probability that all persons in the group support part C is "**,** countC **/** N**)**
39. allPersonsInGroup**(**1000000**)**

# **QUESTION 2 – AN EQUAL BOYS AND GIRLS**

1. INTRODUCTION:

A class of 4n children contains 2n boys and 2n girl. Choose 2n children from the class at random. The purpose of this program is calculating the probability that the chosen group has an equal number of boys and girls. The program will print the probability for two different ns (n = 10 and n = 50).

1. PROCEDURE:

* Write a function and pass a parameter n (the number of students in class is 4\*n) and N (the number of experiments, we chose N = 1,000,000)

+ Initialize the counter to count the number of experiments that meet the requirement (the number of boys and girls are equal)

+ Use the for loop from 1 to N. For each experiment:

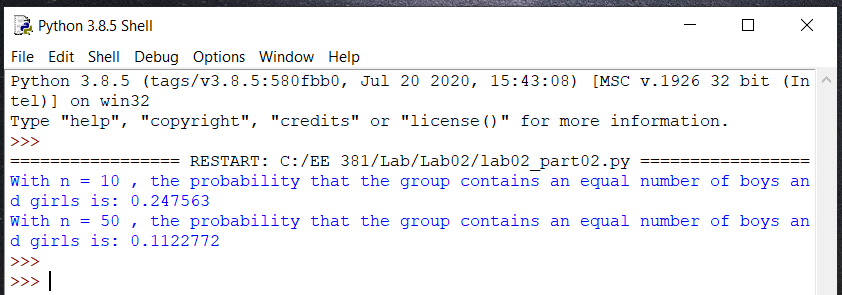
* Create a sample space – a sequence of numbers from 1 to 4\*n
* Create a list of random items of length 2\*n from the sample space. Each number k in the list is represent for a student number k is chosen.
* Go over 2\*n random numbers. If this number is less than or equal 2\*n, we assign that the student has this number is a boy and increase the counter by 1. Otherwise, the number is from 2\*n+1 to 4\*n is a symbol for a girl.
* Then check the counter. If the counter equal n (n boys and n girls are chosen), increase the counter of experiments by 1.

+ Divide the counter of experiments to N, we have the probability we need to find.

* Pass two different n numbers into the function to get the probabilities.

1. RESULTS AND DISCUSSION:

* Results:



* The probability that the chosen group has the same number of boys and girls is for n = 10 or for n = 50. Our results are close to these results, so they are what we expected.

1. CONCLUSION:

In this part of the lab, some aspects we used from lectures include probability and combinations. We found that no problems or errors occurred when writing this program. We also used random.sample() function to make sure each number in the random list is unique.

1. APPENDIX:
2. """
3. Class: EE 381 - Section 13
4. Project 02 - Binomial Coefficient \_Part 02
5. """
6. **import** numpy **as** np
7. **import** random
8. #Function
9. """
10. Find the probability
11. of getting a group of students that has an equal number of boys and girls
12. Input: N (The number of experiments), n (the number of students equal 4\*n)
13. Output: The probability
14. """
15. **def** getEqualBoyGirl**(**N**,** n**):**
16. result **=** 0 #count the number of experiments that meet the requirement
17. **for** i **in** **range(**0**,** N**):** #for each experiment
18. S **=** **range(**1**,** 4**\***n**+**1**)** # create a sequence of numbers from 1 to 4\*n
19. #create a list of random items of length 2\*n from the given string
20. #choose 2\*n student from a 4\*n-student class
21. choices **=** random**.**sample**(**S**,** 2**\***n**)**
22. count **=** 0 # a counter to get the number of boys
23. **For** k **in** choices**:**
24. **if** **(**k **<=** 2**\***n**):** # boy is marked with the number from 1 to 2\*n
25. count **+=** 1
26. # if the group contains an equal number of boys and girls
27. # => the number of boys equal n
28. **if** **(**count **==** n**):**
29. result **+=** 1
30. **print(**"With n ="**,** n**,** ", the probability that the group contains an equal number of boys and girls is:"**,** result**/**N**)**
31. #Main
32. **if** \_\_name\_\_ **==** "\_\_main\_\_"**:**
33. getEqualBoyGirl**(**1000000**,** 10**)**
34. getEqualBoyGirl**(**1000000**,** 50**)**

# **QUESTION 3 – A LOTTERY GAME**

1. INTRODUCTION:

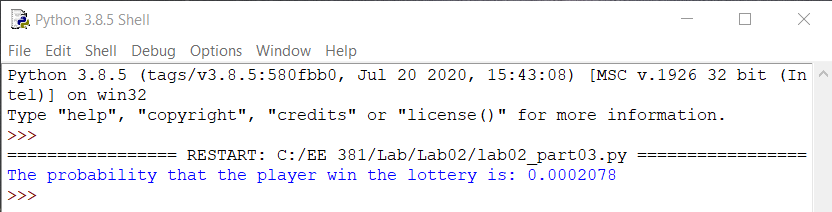
This program calculates the probability that the player will win a lottery game. In this game, the player picks 4 numbers from a sequence of 1 through 20. At lottery drawing, 4 balls are drawn at random from a box containing 20 balls numbered 1 through 20. If the player gets 4 matches in any order, the play will win. Otherwise, the player will lose.

1. PROCEDURE:

* Initialize the sample space – a sequence from 1 through 20, and the counter to count the number of experiments that the player wins.
* Use the for loop from 1 to N (the number of experiments, we chose N = 10,000,000)
* For each experiment:
* Generate 4 random numbers from the sample space. They are the number that the player chose.
* Generate 4 random numbers from the sample space again. These numbers are the numbers that the lottery game drawn.
* Sort 2 lists above.
* Check if there are 4 matches, it means the player wins the lottery, then increase the counter of experiments by 1.
* Divide the counter of experiments to N, we have the probability we need to find.

1. RESULTS AND DISCUSSION:

* Result:



* The result is what us expected. This result is close to the probabilities that we calculate in class. As we calculated, the probability that the player wins the game is .

1. CONCLUSION:

* For this question, we used probability and combinations from the lecture to calculate the probability that the player will win.
* We also used random.sample() to get a list of random numbers without duplicates.

1. APPENDIX:
2. """
3. Class: EE 381 - Section 13
4. Project 02 - Binomial Coefficient \_Part 03
5. """
6. **import** numpy **as** np
7. **import** random
8. #Function
9. """
10. Find the probability that the player win the lottery
11. (player choose 4 numbers from a sequence of 1 through 20 and
12. these numbers have to match with 4 random balls are drawn)
13. Input: N (The number of experiments)
14. Output: The probability
15. """
16. **def** winLotteryGame**(**N**):**
17. count **=** 0 #count the number of experiments that meet the requirement
18. S **=** **range(**1**,** 21**)** # create a sample space - a sequence of numbers from 1 to 20
19. **for** i **in** **range(**0**,** N**):** #for each experiment
20. # the player picks 4 numbers from 1 to 20
21. numbers **=** random**.**sample**(**S**,** 4**)**
22. # 4 balls are drawn at random from a box containing 20 balls numbered 1 through 20
23. balls **=** random**.**sample**(**S**,** 4**)**
24. numbers**.**sort**()**
25. balls**.**sort**()**
26. # if get 4 matches => the player win the lottery game => increase count
27. **if** **(**numbers**[**0**]** **==** balls**[**0**]** **and** numbers**[**1**]** **==** balls**[**1**]** **and** numbers**[**2**]** **==** balls**[**2**]** **and** numbers**[**3**]** **==** balls**[**3**]):**
28. count **+=** 1
29. #print(numbers)
30. #print(balls)
31. #print(count)
33. **print(**"The probability that the player win the lottery is:"**,** count**/**N**)**
34. #Main
35. **if** \_\_name\_\_ **==** "\_\_main\_\_"**:**
36. winLotteryGame**(**10000000**)**