MAS8403 Statistical Foundations of Data Science

R Solution

Practical 1: The Birthday Problem

Part 1:

1. #Part 1
2. #Is there any people share birthday
3. # param:
4. #   Num\_Peo (int): Number of People in Room
6. ShareBirth\_Part1 = function(NUm\_Peo){
8. #Random Create the Date of People in the Room
9. birthDay = c(sample(1:365,replace = TRUE, NUm\_Peo))
11. # Judge whether the People Share Birthday Comparing the length
12. **if** (length(birthDay) == length(unique(birthDay))){
14. #No People share birthday with  no repetitive Date
15. **print**('No people share Birthday')
17. **return** (birthDay)#Reture the Date of People in the Room
18. }
19. **else**{
21. #People share birthday with Date repetitive
22. **print**('There are some people share birthday')
24. **return** (birthDay)# Reture the Date of People in the Room
25. }
27. }
29. Main\_Part1 = function(){ #The Main function to Run the Part 1
30. NUM = 10 #If there are NUM of people in the room
31. ShareBirth\_Part1(NUM) # Run to Judge
32. }
34. Main\_Part1()# Run the Part 1

Results:

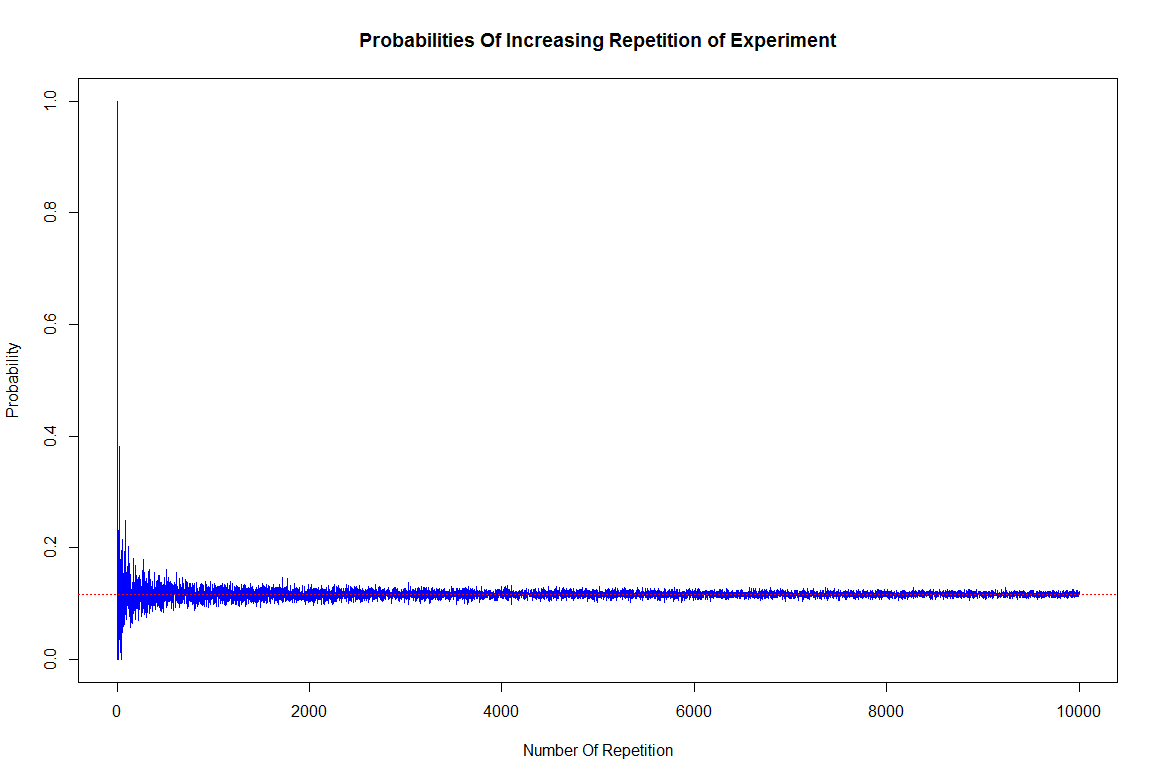
There are some people share birthday OR No people share Birthday

And RETURN the DATE of people’s birthday

Part 2:

1. #Part 2
2. #Param :
3. #    NUm\_Re: Number of Experiments will Repeate
4. #
5. ShareBirth\_Part2 = function(NUm\_Re){
7. # Initialization
8. least\_one\_Peo = 0 # At least one people share birthday in ALL experiments
9. RE\_Exp = NUm\_Re # Flag of how many experiments
11. **while**(RE\_Exp != 0){ # Loop the Num\_Re Times for Experiments
13. #Random Create the Date of 10 People in the Room
14. birthDay = c(sample(1:365,replace = TRUE, 10))
16. # Judge whether the People Share Birthday Comparing the length
17. **if** (length(birthDay) == length(unique(birthDay))){
19. #No People share birthday with  no repetitive Date
20. **print**('No people share Birthday')
21. }
22. **else**{
24. #People share birthday with Date repetitive
25. **print**('There are some people share birthday')
27. #Count the Number of People Share Birthday
28. least\_one\_Peo = least\_one\_Peo + 1
29. }
31. #In Order to Repeat the Experiment
32. RE\_Exp = RE\_Exp - 1
33. }
35. #Probability of People Share Birthday in ALL Experiments in One Setting
36. Pr = least\_one\_Peo / NUm\_Re
37. **print**(Pr)
39. **return**(Pr)
40. }
42. # Main Function For Part 2
43. Main\_Part2 = function(){
44. Num\_Re = c(1:10000) # Number of Repetition of Experiments
46. Count\_Pro = 1 # Index the Vector to Store the each Probabilities
48. Pr\_Vec = c()# Store the Num\_Re times of Probabilities
50. # Get the Pro babillity of Each Experiments and Store Them
51. **for** (count **in** Num\_Re){
52. Pr\_Vec[Count\_Pro] = ShareBirth\_Part2(count)
53. Count\_Pro = Count\_Pro + 1
54. }
56. #Plot the Trend of Probabilities Of Increasing Repetition of Experiment
57. #x : Numeber of Repetition of the Experiments
58. #y : The Probabilities Of Each Experiment
59. **print**(Pr\_Vec)
60. **print**(Num\_Re)
61. plot(Num\_Re, Pr\_Vec, type = 'l', main = 'Probabilities Of Increasing Repetition
62. of Experiment',xlab = 'Number Of Repetition',
63. ylab = 'Probability', col = 'blue')
65. #Always have the Exact probabillity to Check
66. abline(h=0.117,col = 'red', lty =3)
67. }
69. Main\_Part2()# Run the part 2

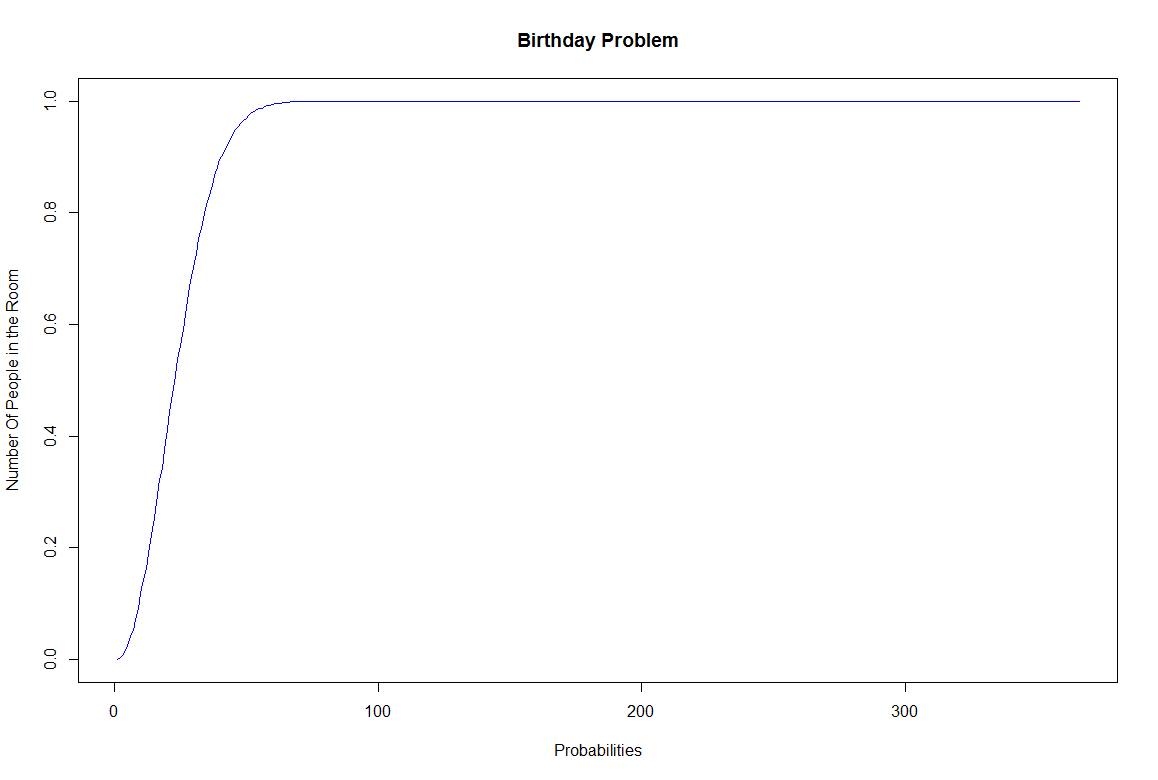
Results:



Part 3:

1. #Part 3
2. #Param :
3. #      Num\_Peo = Number of the people in the room
4. #      Num\_Re = Number of repetition of the experiments
5. ShareBirth\_Part3 = function(NUm\_Peo, NUm\_Re){
7. Count\_Pro  = 0 # Index the Vector to Store the each Probabilities
8. Pr\_Vec\_3 = c() # Store the Num\_Re times of Probabilities
9. Flag\_Num = 0 #The number of people are required before the probability becomes at least 0.5
10. Flag\_Pr = TRUE # Flag of find the number get probability at least 0.5
12. **for** (count **in** 1:NUm\_Peo){ #loop of 1->366 people
14. # At least one people share birthday in ALL experiments
15. least\_one\_Peo = 0
16. # Flag of how many Repetitions of Experiments
17. RE\_Exp = NUm\_Re
19. **while**(RE\_Exp != 0){ #loop of Repetition Of Experiment
21. #Random Create the Date of 10 People in the Room
22. birthDay = c(sample(1:365,replace = TRUE,count))
24. **if** (length(birthDay) != length(unique(birthDay))){
26. #People share birthday with Date repetitive
27. # Count the Number of People Share Birthday
28. least\_one\_Peo = least\_one\_Peo + 1
29. }
31. #In Order to Repeat the Experiment
32. RE\_Exp = RE\_Exp - 1
34. }#end of experiment
36. #Probability of People Share Birthday in ALL Experiments in One Setting
37. Pr\_3 = least\_one\_Peo / NUm\_Re
39. # Find the the number of people are required
40. # before the probability becomes at least 0.5
41. **if**(Pr\_3 >= 0.5 & Flag\_Pr){
43. #Get the number of people required before the probability becomes at least 0.5
44. Flag\_Num = count
46. # After a number, probabilities are always 0.5,so just get the first number
47. Flag\_Pr = FALSE # Keep the first number of people (at least 0.5)
48. }
50. # Get the Pro babillity of Each Experiments and Store Them
51. Count\_Pro = Count\_Pro  + 1
52. Pr\_Vec\_3[Count\_Pro] = Pr\_3
53. }
55. #Probabilities of ALL experiments
56. #the fisrt number of at least 0.5
57. **print**(Flag\_Num)
58. **return**(Pr\_Vec\_3)
59. }
61. Main = function(){
63. NUm\_Peo = 366 #Number of People in Room
64. NUm\_Re = 20000 # Number of Repetition of Experiments
65. #Get the all probabilities
66. Pr\_Vec\_3= ShareBirth\_Part3(NUm\_Peo, NUm\_Re)
67. #Plot it
68. plot(1:NUm\_Peo, Pr\_Vec\_3, type = 'l', main = 'Birthday Problem',xlab = 'Probabilities',
69. ylab = 'Number Of People in the Room', col = 'blue')
70. }
72. Main()# Run the Part 3

Results:



PRINT : 23