

# MA213 Basic Statistics and Probability - Lab3

## Lab 3: Simulation and Probability

In this lab session, we will go over random variables and compute the expectation and variance of random variables and their linear combinations. We will verify whether the theory holds using R.

### Pre-lab activities

- We are going to work in groups. This time, each group should have 3 people. You can discuss the exercise questions and post-lab activities with your members.

```
num_ppl_each_group = 3
student_list <- read.csv("StudentList.csv")
N <- nrow(student_list)

# Sort by last name
student_list <- student_list %>% arrange(Last)
student_list$number <- 1:N

# Shuffle the student numbers randomly
shuffled <- sample(student_list$number)

# Create group
group_ids <- rep(1:ceiling(N / num_ppl_each_group), num_ppl_each_group)[1:N]

# Assign data frame
group_df <- data.frame(number = shuffled, group = group_ids)
grouped_students <- left_join(group_df, student_list, by = "number")

grouped_students <- grouped_students %>% arrange(group)

# Output
grouped_students %>% select(!number)
```

```
##      group First Last
## 1         1    C2  CC2
## 2         1     X   XX
## 3         1     H   HH
## 4         2     L   LL
## 5         2     G   GG
## 6         2     R   RR
## 7         3     M   MM
## 8         3     D   DD
## 9         3     A   AA
## 10        4     T   TT
## 11        4     E   EE
## 12        4     Q   QQ
## 13        5     O   OO
```

```
## 14      5      S   SS
## 15      5      F   FF
## 16      6      V   VV
## 17      6      U   UU
## 18      6     A2  AA2
## 19      7      I   II
## 20      7     B2  BB2
## 21      7      Y   YY
## 22      8      N   NN
## 23      8      J   JJ
## 24      8      Z   ZZ
## 25      9      P   PP
## 26      9      C   CC
## 27      9      W   WW
## 28     10      B   BB
## 29     10      K   KK
```

**Q)** Let's roll a die! You are playing a dice-rolling game and you're curious about how this game goes at the end. The rule is you roll it once and other player roll it once and add the outcome. What can you expect from this game in the end? In other words, what are the expected value and variance?

We let  $X_1$  be the number on the top face of the first die rolled, and  $X_2$  be the number on the top face of the second die rolled.

Then, what is  $E(X_1 + X_2)$  and  $Var(X_1 + X_2)$ ?

How do you justify it?

## Lab activities

Let's consider if we want to do  $i = 1, \dots, 10$ . How do you want to do it?

1. Rolling dice part
2. Record each run
3. Calculation at the end

We need three people. Two people roll the dice each time and the other person record each turn.

i	1	2	3	4	5	6	7	8	9	10
person 1										
person 2										
outcome										

How do you roll each run using R?

Sample function —

## Random Samples and Permutations

**Description** `sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

### Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

## Arguments

---

x	either a vector of one or more elements from which to choose, or a positive integer. See ‘Details.’
n	a positive number, the number of items to choose from. See ‘Details.’
size	a non-negative integer giving the number of items to choose.
replace	should sampling be with replacement?
prob	a vector of probability weights for obtaining the elements of the vector being sampled.

---

Let’s roll one dice and show the output.

```
outcome <- sample(c(1,2,3,4,5,6), 1)
outcome
```

```
## [1] 1
```

*# or*

```
outcome <- sample(1:6, 1)
outcome
```

```
## [1] 5
```

*#so outcome from those two dice*

```
dice1 <- sample(1:6, 1)
dice2 <- sample(1:6, 1)
sum_of_two <- dice1 + dice2
sum_of_two
```

```
## [1] 9
```

**Loops** for loops are used to iterate operations over items in a vector.

The basic syntax is of the form:

```
for (item in vector) {
  perform_action
}
```

For example, if you want to print 1 to 10 each iteration.

```
my_vector = 1:10
```

```
for (i in my_vector){
  print(i)
}
```

```
## [1] 1
```

```
## [1] 2
```

```
## [1] 3
```

```
## [1] 4
```

```
## [1] 5
```

```
## [1] 6
```

```
## [1] 7
```

```
## [1] 8
```

```
## [1] 9
```

```
## [1] 10
```

```
myoutcome = rep(0, 100) # make a vector of size 100 that has all zero's

for (i in 1:100){
  dice1 <- sample(1:6, 1)
  dice2 <- sample(1:6, 1)
  sum_of_two <- dice1 + dice2
  myoutcome[i] <- sum_of_two
}

head(myoutcome)
```

Using for loop, assign outcome of two dice rolled into myoutcome from 1000 iterations.

```
## [1] 6 2 7 4 7 9
```

```
twodice_outcome <- function(){
  dice1 <- sample(1:6, 1)
  dice2 <- sample(1:6, 1)
  sum_of_two <- dice1 + dice2
  myoutcome <- sum_of_two
  return(myoutcome)
}

twodice_outcome()
```

Use replicate(), it carries out repeated tasks more efficient way.

```
## [1] 8
```

```
replicate(n=10, twodice_outcome()) # this will work as above lines
```

```
## [1] 3 9 8 10 6 5 3 8 9 7
```

Calculate probability of each outcome.

```
factored_outcome <- factor(myoutcome)
outcome_table <- table(factored_outcome)
n <- length(myoutcome)
prob_outcome <- outcome_table/n
prob_outcome
```

```
## factored_outcome
##      2      3      4      5      6      7      8      9     10     11     12
## 0.08 0.05 0.10 0.11 0.08 0.19 0.16 0.09 0.07 0.04 0.03
# how to ensure that it is proper probability?
sum(prob_outcome)
```

```
## [1] 1
```

Calculate the simulated expected value of the random variable.

```
outcome_table * prob_outcome
```

```
## factored_outcome
##      2      3      4      5      6      7      8      9     10     11     12
```

```
## 0.64 0.25 1.00 1.21 0.64 3.61 2.56 0.81 0.49 0.16 0.09
```

```
names(outcome_table) # we need this
```

```
## [1] "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"
```

```
values <- as.numeric(names(outcome_table) )
```

```
# or
```

```
values <- 2:12
```

```
sum(values * prob_outcome)
```

```
## [1] 6.66
```

```
mu_hat = sum(values * prob_outcome)
```

Calculate the simulated variance of the random variable.

```
sigma_2_hat <- sum( (values-mu_hat)^2 * prob_outcome )
```

```
sigma_2_hat
```

```
## [1] 110.2716
```

Make a function that has an input `n` and a list of output that gives you the simulated expected value and variance.

```
sim_fn <- function(n=1000){
```

```
  myoutcome = rep(0, n)
```

```
  for (i in 1:n){
```

```
    dice1 <- sample(1:6, 1)
```

```
    dice2 <- sample(1:6, 1)
```

```
    sum_of_two <- dice1 + dice2
```

```
    myoutcome[i] <- sum_of_two
```

```
  }
```

```
  factored_outcome <- factor(myoutcome)
```

```
  outcome_table <- table(factored_outcome)
```

```
  prob_outcome <- outcome_table/n
```

```
  values <- as.numeric(names(outcome_table) )
```

```
  mu_hat = sum(values * prob_outcome)
```

```
  sigma_2_hat <- sum( (values-mu_hat)^2 * prob_outcome )
```

```
  output <- list(mu = mu_hat, sigma2 = sigma_2_hat)
```

```
  return(output)
```

```
}
```

```
sim_fn(100)
```

```
## $mu
```

```
## [1] 7.13
```

```
##
```

```
## $sigma2
```

```
## [1] 109.1859
```

```
sim_fn(1000)
```

```
## $mu
```

```
## [1] 7.001
```

```
##
```

```
## $sigma2
```

```
## [1] 109
```

```
sim_fn(10000)
```

```
## $mu
```

```
## [1] 6.9607
```

```
##
```

```
## $sigma2
```

```
## [1] 109.017
```

```
sim_fn(100000)
```

```
## $mu
```

```
## [1] 6.99516
```

```
##
```

```
## $sigma2
```

```
## [1] 109.0003
```

This time, you invented another challenging game now. If one person roll a dice, then the outcome is multiplied by 3. Also when second person roll a dice, then the outcome is multiplied by 4. The final outcome is the sum of those two values. What is the expected value and variance of the random variable?

Q. What is the expected value of the total outcome?

```
#  
#
```

Q. Let's simulate it. Make a function to obtain one output from this setting.

```
#  
#  
#
```

Q. Using the function above, Make an another function to obtain the expected value from  $n$  iteration.

```
#  
#  
#
```

Q. Using the function above, could you plot a line where x-axis is  $n$  and y-axis is expected value of the outcome?

Q. What is the expected value of one die? And variance of one die? Compare your answers to  $3E(X_1) + 4E(X_2)$ . Also, What is  $Var(3X_1 + 4X_2)$ ?

### Post-lab activities

Please name your submission as `lab3.R`

Now, You flip a slightly biased coin where  $P(head) = 0.49$  and  $P(tail) = 0.51$ . If the result is head, you are given \$10, otherwise you pay \$10. How much money you have at the end?

Calculate theoretical value of  $\mu$  and  $\sigma^2$ . Assign them into `mu` and `sigma_2`, respectively.

You are going to make an function named `my_sim` which takes input of `n`.

(Gradescope will test with  $n = 10, 500$  and  $1000$  ).