

Discrete Probability Assessment

Fabio Carvalho Lima

Assessment: Discrete Probability

Question 1: Olympic running

Libraries that will be used for the calculations in this assessment. This assessment has 3 sections, all related to *Discrete Probability*.

```
library(gtools)
library(tidyverse)
```

In the 200m dash finals in the Olympics, 8 runners compete for 3 medals (order matters). In the 2012 Olympics, 3 of the 8 runners were from Jamaica and the other 5 were from different countries. The three medals were all won by Jamaica (Usain Bolt, Yohan Blake, and Warren Weir).

Use the information above to help you answer the following four questions.

Question 1a - How many different ways can the 3 medals be distributed across 8 runners?

Solution 1a : To answer this question we need to remember in this kind of competition the orders matters, for the first runner, he will get gold medal, and so on, it seems we have to use permutations function to answer this question.

```
all_medals <- nrow(permutations(8,3)) # if orders matters, we must use
permutations                                # nrow will give us the number of
the observations
all_medals
## [1] 336
```

Question 1b - How many different ways can the three medals be distributed among the 3 runners from Jamaica?

Solution 1b : To answer this question we can use the same approach for previous question, the only difference, we have to make the permutations between the runners from Jamaica only.

```
nrow(permutations(3,3)) # all the permutations between 3 runners from the
same country
## [1] 6
```

Question 1c - What is the probability that all 3 medals are won by Jamaica?

Solution 1c:

S = all different ways the 3 medals distributed between 8 runners *(answer question 1a)

$n_{runnersfromJamaica}$ = all ways the Jamaica runners can win

$Pr(runnersfromJamaica)$ = **probability that all 3 medals are won by Jamaica**

```
runners_Jamaica <- nrow(permutations(3,3)) # Jamaica cases
all_8_runners <- nrow(permutations(8,3))  # all cases

Pr_runners_Jamaica <- runners_Jamaica / all_8_runners
Pr_runners_Jamaica # our answer

## [1] 0.0179
```

Question 1d - Run a Monte Carlo simulation on this vector representing of the 8 runners in this race:

```
runners <- c("Jamaica", "Jamaica", "Jamaica", "USA", "Ecuador",
            "Netherlands",
            "France", "South Africa")
```

Calculate the probability that all the runners are from Jamaica.

Solution 1d :

In order to create a Monte Carlo Simulation, we need to 3 steps:

1.Trials:

We need to create a trial which it will represent the sample from the runners vectors, then take 3 samples at time.

```
runners_medals <- sample(runners, size = 3)
```

2.Success

Here we must get all winners from Jamaica.

```
all(runners_medals == "Jamaica")

## [1] FALSE
```

3. Monte Carlo Simulation

Using the definitions we create on step 1 and step 2, here we code the simulation

```

set.seed(1)
B <- 10^5 # number of simulations
medals <- replicate(B, {
  runners_medals <- sample(runners, size = 3)
  all(runners_medals %in% "Jamaica")
})
(Pr_runners_Jamaica <- mean(medals)) # show all the probabilities for the
Jamaica runners
## [1] 0.0177

```

Question 2: Restaurant Management

A restaurant manager wants to advertise that his lunch special offers enough choices to eat different meals every day of the year. He doesn't think his current special actually allows that number of choices, but wants to change his special if needed to allow at least 365 choices.

A meal at the restaurant includes 1 entree, 2 sides, and 1 drink. He currently offers a choice of 1 entree from a list of 6 options, a choice of 2 different sides from a list of 6 options, and a choice of 1 drink from a list of 2 options.

- Question 2a: How many meal combinations are possible with the current menu ?

Solution: All we need to do here is use counting: 1 have only one 1 entree from a list of 6 options: `nrow(permutations(6,1))`

```
entree <- 6
```

For the sides we can write: `nrow(combinations(6,2))`

```
nrow(combinations(6,2))
```

```
## [1] 15
```

For the drink options we will have:

```
drink <- 2
```

To calculate all possible combinations for the menu we can multiply the combinations above:

```
(all_menu_combinations <- entree * nrow(combinations(6,2)) *
  drink)
```

```
## [1] 180
```

- Question 2b: The manager has one additional drink he could add to the special.

How many combinations are possible if he expands his original special to 3 drink options ?

Solution: For this question we just to update the combinations for drink, instead use `nrow(permutations(2,1)) -> nrow(permutations(3,1))`

```
(all_new_menu_combinations <-  
nrow(permutations(6,1))*nrow(combinations(6,2))*  
  nrow(permutations(3,1)))  
## [1] 270
```

- Question 2c:

The manager decides to add the third drink but needs to expand the number of options. The manager would prefer not to change his menu further and wants to know if he can meet his goal by letting customers choose more sides.

How many meal combinations are there if customers can choose from 6 entrees, 3 drinks, and select 3 sides from the current 6 options?

Solution 2c: we need to update our combinations into this new one: for 6 entrees and 6 options:

```
(n_entrees <- 6)  
## [1] 6
```

3 sides from the current 6 options:

```
(n_sides <- nrow(combinations(6,3)))  
## [1] 20
```

3 drinks from a list of 2 options

```
(n_drinks <- 3)  
## [1] 3
```

new menu `n_entrees``n_sides``n_drinks`

```
(new_menu <- n_entrees*n_sides*n_drinks)  
## [1] 360
```

- Question 2d:

The manager is concerned that customers may not want 3 sides with their meal.

He is willing to increase the number of entree choices instead, but if he add too many expensive options it could eat into profits. He wants to know how many entree choices he would have to offer in order to meet his goal.

Use `sapply` to apply the function to entree option counts ranging from 1 to 12. What is the minimum number of entree options required in order to generate more than 365 combinations?

```
entree_choices <- c(1:12)

menu <- function(entree_choices){
  # for n entrees and 6 options: n = entree_choices
  entrees <- entree_choices
  n_sides <- nrow(combinations(6,2)) # sides combinations
  n_drinks <- 3
  new_menu <- entrees*n_sides*n_drinks # menu combinations
}

new_menu <- sapply(entree_choices, menu)
data.frame(entree_choices = 1:12, new_menu = new_menu) %>%
  filter(new_menu > 365) %>%
  min(.$entree_choice)

## [1] 9
```

The manager isn't sure he can afford to put that many entree choices on the lunch menu and thinks it would be cheaper for him to expand the number of sides. He wants to know how many sides he would have to offer to meet his goal of at least 365 combinations.

Write a function that takes a number of side choices and returns the number of meal combinations possible given 6 entree choices, 3 drink choices, and a selection of 2 sides from the specified number of side choices.

```
side_choices <- function(x){
  6 * nrow(combinations(x, 2)) * 3
}

combos <- sapply(2:12, side_choices)

data.frame(sides = 2:12, combos = combos) %>%
  filter(combos > 365) %>%
  min(.$sides)

## [1] 7
```