**Simulating a Genetic Screen**

**Introduction to the R statistical programming language with R-snippets or R-Studio**

https://rdrr.io/snippets/

R is a statistical programming language that is probably the most widely used platform for statistical analysis in biology. It is open source, meaning it is entirely free. R can be used to analyze data, perform statistical tests, run simulations and generate excellent visuals. You are going to be introduced to R using the web browser based R-snippets

The web has an abundant amount of information on R. Nearly all R users are self-taught. Some useful links:

Wikipedia: https://en.wikipedia.org/wiki/R_(programming_language)
Introduction to R PDF: *https://cran.**r**-project.org/doc/manuals/**R**-intro.pdf*
Online Starter Course: https://www.datacamp.com/courses/free-introduction-to-r

**About R-snippets**

https://rdrr.io/snippets/

Most people install R on their machine. One can install R alone, which runs in a command line.For this assignment, we will be using a version of R that runs in a web browser. This should make it easy for this class project.

You may also chose to use RStudio, which is user friendly. One option is to install it yourself on your own computer (*https://www.**rstudio**.com/)*. Alternately, you can also use the computers in the BTRC 1004 Haworth Hall. Computers are free to use there and have R-Studio installed. Computers are also available in Anschutz library and RStudio should be installed (if not, it can be installed quickly through the Software Center)

**2. Introduction to the Project**

Geneticists use mutations to study the function of living organisms. Basically, it's like studying how a car works by breaking components with a hammer and studying what the effect is. This is called a ***Genetic Screen***.

How many mutant strains must one generate to hit all the genes involved in a process? This is a key question. However, the answer depends on several critical factors.

1) How many different gene mutations do you make each time you add a mutagen to a genome? For the purpose of the exercise, we will assume just **ONE** mutation (though usually it is many more - perhaps dozens).

2) How many genes are in the genome? For this purpose, we will assume **10,000.**

3) How many of the total number of genes are critical for a process? For this project, we will assume **100.**

***Note: The project is due on the date provided by the syllabus. However, after handing it in, I will give you one more chance to meet with me and fix the project if you want. This way, everyone should be able to get 25 points.***

**3. R commands you will find useful**

**1. Assign a variable**

**<-**

Example:

x <- 5
x
[1] 5

**2. Adding a value to a variable**, example:

x <-x + 1
x
[1] 6

**3. Generating a random number between X and Y**, example:

**x <- sample(1:10, 1)**
**[1] 10**

**x <- sample(1:100, 1)**
**x**
**[1] 77**

3.5 Making a list(vector)

x <- c(1,2,3,4,5)
x
1,2,3,4,5

**4. Calculations can be performed on a vector.**

Example

y <- x/20
y
[1] 0.45 0.60 0.70 0.65 0.45 0.60 0.50 0.30 0.50 0.55

**5. Summarizing data**

hist(x): histogram
*note: limits of X on histogram can be set with xlim. For example,* hist(x,xlim = c(0,200)) sets the x values between 0 and 200

**Control Commands**

**7. Adding something to a vector**

x <- c(x,10)

Example:

x <- c() *establishing a null vector*

```
x <- c(x,10)
x
[1] 10
x <- c(x,10,20,30)
x
[1] 10 10 20 30
```

**8. Logic. If statements**

```
if (test_expression) {
statement
}
```

Example:

```
x <- 0
y <- 0
z <- 0
x
[1] 0
y
[1] 0
z
[1] 0

if(x==y) {z<-z+1}  #NOTE the double equals!
z
[1] 1

if(x==z) {z<-z+1}  #NOTE the double equals!
z
[1] 1

if(x==y) {z<-z+1}
z
[1] 2
```

Also, other logical operators:

```
x < y
[1] FALSE

z >x
[1] TRUE
```

not equals:

```
x != y
[1] FALSE
```

**8. For Loops**

```
for (i in 1:10) {...
}
```

Example:

```
x<-c()
for (i in 1:10) {
x <- c(x,2*i)
print(x)
}
x<-c()
for (i in 1:100) {
z<-sample(1:100,1)
x <- c(x,z)
}
x
hist(x)
```

**x <- sample(1:10, 1)**

**9. While loops**

**The logic of a while loop is this - you want something to happen until something is achieved.** In this project, what you want to happen is to continue making new mutants until you obtain one mutant. Then, you want to see how long it took until you obtained the mutation. Finally, you want to record how long it took to obtain a mutation for one experiment. By simulating many experiments, you want to get a sense as to how long it will typically take to obtain your first mutant.

**Note**: In this project, we will assume only one mutation is generated for each genome screened. We make this assumption for simplicity. Typically, many mutations are made.

```
while (test_expression) {
statement
}

index <-0
num_mutant_found <-0

while (num_mutant_found == 0){
        x <- sample(1:10000, 1)
        if (x < 101) {num_mutant_found <- num_mutant_found +1}
        index <- index + 1
}
```

**Note: Think about how you are going to record index and make sure you reset index and num_mutant_found back to zero when you need to use them again. Also, why did I elect to use x < 101 in the if statement?**

**Finally, while statements and for loops can of course be nested in other while statements and for loops. For the project, you will probably need to nest a while loop inside a for loop.**

*Note: To run a program, you will need to run it above the single line, in the text editor above, then click Run code.*

**4. Assignment**

You are interested in performing a mutagenesis experiment in the bread mold *Neurospora crassa* to identify genes that are critical for the biosynthesis of methionine. Mutants that are defective in the synthesis of methionine must have the amino acid added to their growth media.

For the assignment, you want to use simulation to get a sense of how long it will take to obtain one mutant strain. Sometimes it will take a long time. Other times you might get lucky and find a mutation early in the experiment.

To do:

Simulate 100 experiments to figure out the expected waiting times until a first mutation is obtained. In each experiment, you will keep generating mutants until you obtain one mutation that leads to a defect in the process. For that one experiment, record how long it took. Simulate this process 100 times.

To submit:

Please submit a print out of:

a. The code you used.
b. A histogram of the distribution of times until the first mutation was found for 100 experiments.
c. Provided answers to the following questions.

1) Do you feel that this is an efficient approach to identifying novel genes critical for the synthesis of methionine?
2) How might you change the experimental design to make it more efficient?
3) Briefly describe how you could use a simulation to determine whether this change might make the screen more efficient.