Pre Lab 05 Group Work

You will begin this activity as group work, and the rest will be submitted as Lab 05. You are encouraged to work together, but you should submit your individual work.

Scenario

Let's suppose a traffic light is green with probability 3/4 and red with probability 1/4. You want to guess what the next color will be.

Consider the following two guessing strategies:

- Strategy A: guess green 3/4 of the time and red 1/4 of the time, choosing "randomly" with these probabilities each time
- Strategy B: guess green every time

Task 1

Which strategy do you think is better? Why?

Task 2

Write psuedocode for how you could simulate both strategies. Some tips for getting started:

- Simulate one (large) vector you consider to be the "true" sequence of colors, which you will then compare your A and B guesses to
- You can make use of the `rbernoulli()` function
- Make sure to define what TRUE and FALSE will represent
- You want to end up with three vectors: one for the true values, one for the A guesses, and one for the B guesses. All should have the same (large) number of elements (e.g. 10,000)

Task 3

Conduct your simulations in R to estimate the probability of success for each strategy. You should set a seed at the top of your code chunk so you don't get a different answer everytime you run the code. Download the `Lab_05.qmd` file from our course template for a template document to type up your work.

Tasks 4 - 6 should be done "by hand" on separate paper (not in R).

Task 4

Note the sample space of (actual,guess) pairs, where "g" represents "green" and "r" represents "red":

$$S = \{gg, gr, rg, rr\}$$

Consider the following two random variables:

- $C_A = \{1$ when strategy A leads to a correct guess, 0 otherwise$\}$
- $C_B = \{1$ when strategy B leads to a correct guess, 0 otherwise$\}$

Mathematically develop a probability distribution for these two random variables. That is, find:

- $P(C_A = 1)$
- $P(C_A = 0)$
- $P(C_B = 1)$
- $P(C_B = 0)$

How do your simulations compare to these probabilities?

Task 5

Consider a general strategy that makes the correct guess with probability p. Suppose you are interested in the random variable, Y, that gives the number of trials needed until the first correct guess is made. Develop a probability distribution for this new random variable.

Hint: think in terms of slots, and where correct and incorrect guesses must fall

Task 6

Suppose you are interested in the random variable, Z, that gives the number of trials needed until the 20th correct guess is made. Develop a probability distribution for this new random variable.

Hint: again think in terms of slots, and where correct and incorrect guesses must fall. Don't neglect the multiple ways to achieve k successes in a series of trials.

Simulating Y

The following code simulates 10,000 values of the random variable described in Task 5, which we'll call `Y` (number of trials until the 1st success), and plots the resulting distribution. We use $p = 0.75$ to mimic the traffic light example above.

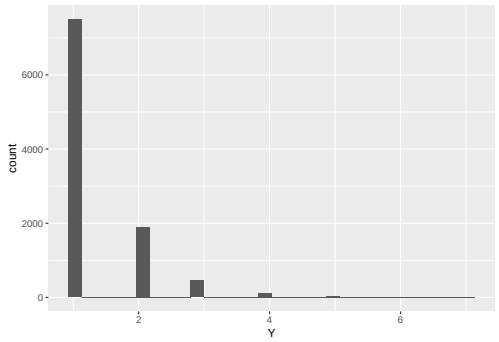
```
set.seed(437)
#initialize empty vector
#will fill with the # of trials needed to reach first success
Y <- c()
for(i in 1:10000){
  #initialize the # of successes and # of trials to be 0
  num_successes <- 0
  trials <- 0

  #while # of successes remains 0,
  #generate a new observation x of a bernoulli random variable
  #update the number of trials
  #once x = TRUE, num_successes will be updated to 1
  #and while loop will end
  while(num_successes < 1){
    x <- rbernoulli(1, .75)
    num_successes <- num_successes + x
    trials <- trials + 1
  }

  #fill element i of the vector Y with the number of trials
  #it took to reach the first success
  Y[i] <- trials
  #for loop repeats this process 10,000 times
}

#place simulation results in a dataframe
sims_Y <- data.frame(Y)

#plot simulation results
ggplot(sims_Y, aes(x = Y)) +
  geom_histogram()
```



Task 7

Comprehension check: how many values will be stored in the vector `Y` by the end of the for loop? What are the possible values of Y?

Comparing Simulated and Theoretical Probabilities

We can calculate the theoretical probabilities using the probability distribution and compare them to the simulated probabilities.

```
#calculate the theoretical probabilities from the pmf
#create vector with all possible values of Y from the simulations
support_Y <- seq(1, max(Y))
support_Y
```

```
## [1] 1 2 3 4 5 6 7
```

```
#probability of one success and y-1 failures
#calculated for each value of y in the support
pY <-  .75*(.25)^(support_Y-1)
pY
```

```
## [1] 0.7500000000 0.1875000000 0.0468750000 0.0117187500 0.0029296875
## [6] 0.0007324219 0.0001831055
```

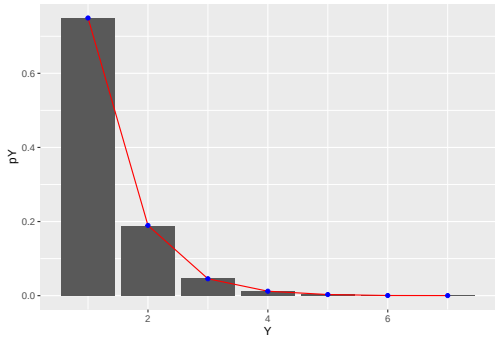
```
prob_dist_Y <- data.frame(Y = support_Y,
                          pY)

#add a column with the simulated relative frequencies
prob_dist_Y$pY_sims <- proportions(table(sims_Y))

prob_dist_Y
```

```
##   Y           pY pY_sims
## 1 1 0.7500000000  0.7493
## 2 2 0.1875000000  0.1895
## 3 3 0.0468750000  0.0456
## 4 4 0.0117187500  0.0120
## 5 5 0.0029296875  0.0030
## 6 6 0.0007324219  0.0004
## 7 7 0.0001831055  0.0002
```

```
#plot the theoretical probabilites on top of the simulated distribution
ggplot(prob_dist_Y) +
  geom_col(aes(x = Y, y = pY)) +
  geom_line(aes(x = Y, y = pY_sims), color = "red") +
  geom_point(aes(x = Y, y = pY_sims), color = "blue")
```



Task 8

Adapt the code provided above to simulate the distribution of the random variable described in Task 6 and compare it to its theoretical distribution. Call this random variable Z. Note this simulation will take longer to run, so you may want to use only 1000 iterations.