Simulation in R

1. Let's re-visit de Montmort's matching problem: We have a well-shuffled deck of n cards, labeled 1 through n. You flip over the cards one by one, saying the numbers 1 through n as you go. You win the game if at some point, the number you say aloud is the same of the number on the card flipped over. For large n, we learned that the probability of winning is approximately $1 - e^{-1}$.

   (a) For $n = 5$, write `R` code that uses the `sample()` function to play one iteration of the game. Your code should output whether the game results in a win or loss. This can be done by reporting `TRUE` or 1 if you won and `FALSE` or 0 if you lost. Your code will probably use the following:

   - The function `sample()`
   - Comparison of two vectors with `==`
   - The function `sum()`. If `x_vec` is a vector, then `sum(x_vec)` adds up all the values in the vector.

   (b) Optionally copy-and-pasting your code from (a), now use the `replicate()` function to simulate 10000 iterations of the game. Store the results in a variable called `sims`. Then use `sims` to approximate the probability of winning, still for $n = 5$.

   (c) Now generalize (b) by creating a function called `matching_prob` that returns the simulated probability of winning for some number of cards n and some number of iterations B used to approximate the probability. That is, n and B should be arguments in your function.

   (d) Using your new `matching_prob()` function, approximate the probability of winning in game with $n =$ each of 2, 3, and 10 cards using 5000 simulations. Verify that these get closer to the value $1 - 1/e$.