STA 247 — Assignment #1. Due in class on October 7.

Late assignments will be accepted only with a valid medical or other excuse.
Worth 6% of the course mark.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. Handing in work that is not your own is a serious academic offense. Fabricating results, such handing in fake output that was not actually produced by your program, is also an academic offense.

Questions 1 and 2 are to be solved using pencil and paper (and maybe a calculator), not by writing a program. Include enough details in your answers to make clear how you obtained them, and also include the final numerical answer (not just a formula). Question 3 involves writing R programs, which you should hand in along with your output and discussion.

Question 1: Suppose that we flip a coin six times and roll a 6-sided die once. Suppose also that all outcomes of this experiment (consisting of an ordered sequence of results for the flips (heads or tails) and the number showing on the die after the roll (an integer from 1 to 6)) are equally likely.

Find the following probabilities:

1) The probability that all six flips are heads and the die shows the number 6.
2) The probability that the first R flips are heads, where R is whatever number is showing on the die.
3) The probability that the number of flips that are heads times the number showing on the die is 18.
4) The probability that the number of heads in the six flips is same as the number showing on the die.

Question 2: A school has a computer lab that is closed for the summer, when no students are there, and then is re-opened in the fall. Sometimes, a computer in the lab that worked before the lab closed is found to not work properly when it is powered up again in the fall.

These computer failures are always due to one or more of the following causes:

A) Failure of the power supply.
B) Failure of the disk drive.
C) Failure of the RAM.
D) Failure of the CPU.

From long experience, it is known that these failure events are independent, and that their probabilities of occurrence (for a single computer) are $P(A) = 0.04$, $P(B) = 0.03$, $P(C) = 0.02$, and $P(D) = 0.01$.

1) Find the probability that a computer will fail for any of these reasons — that is, find $P(F)$, where $F = A \cup B \cup C \cup D$.
2) Find the probability that a non-working computer has failed for more than one reason — that is, find the conditional probability that more than one of A, B, C, and D has occurred, given that $F = A \cup B \cup C \cup D$ has occurred.

Question 3: Suppose that Joe draws k balls from an urn containing n red balls and n green balls, without replacing the balls after they are drawn. Similarly, Mary draws k balls from an urn containing m red balls and m green balls, without replacing the balls after they are drawn. We want to compute the probability that Joe and Mary will draw the the same number of red balls.

1) Write an R function to compute this, which takes n, m, and k as arguments. (These arguments must be positive integers, and $2n$ and $2m$ must be at least as big as k, but you don't have to check for this in your program.) This function should use one or more of the `sum`, `prod`, `factorial`, and `choose` functions. Note that `factorial` and `choose` can take vectors as arguments, and then return a vector of results. Note also that a vector that is a sequence can be created with an expression like `i:j`. Test your function on at least the following values for the arugments:

   $n = 20,\ m = 30,\ k = 1$
   $n = 20,\ m = 30,\ k = 2$
   $n = 200,\ m = 300,\ k = 2$
   $n = 2000,\ m = 3000,\ k = 2$
   $n = 50,\ m = 60,\ k = 17$

   Comment on the results. Can you see why some of them are what they are (at least approximately) with simple calculations?

2) Try the function you wrote above with $n = 600$, $m = 500$, and $k = 400$. You should see a result of `NaN`, indicating that floating point overflow occurred as some point in the computation, so the final result was meaningless. Write a new version of the function that avoids this problem by working in terms of the logarithms of the values, until the final result needs to be computed. The (natural) logarithm is computed with the `log` function, and its inverse, the exponential function, is computed with `exp`. The `lfactorial` and `lchoose` functions compute the log of the factorial and "choose" functions. Test your new function on the same sets of arguments as above, for which it should produce the same (or very close to the same) answer, as well as on $n = 600$, $m = 500$, and $k = 400$ and on $n = 6000$, $m = 5000$, and $k = 4000$, for which it should not produce `NaN`.