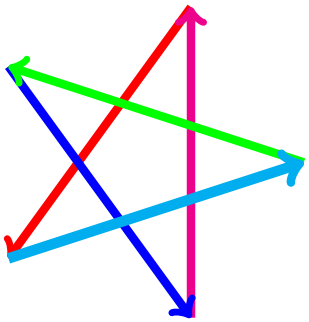
Studio 2 Binomial distributions; Derangements
18.05, Spring 2025



Overview of the studio

This studio works with the binomial distributions Binomial(n, p). It does both exact computations and simulations.

R introduced in this studio

The R needed is all introduced in `studio2-samplecode.r`. We will use

1. Functions for the binomial distribution `rbinom(), dbinom(), pbinom()`.

In R terminology the r in rbinom stands for random, i.e. `rbinom()` generates random samples from the binomial distribution. `dbinom()` computes the pmf and `pbinom()` computes the cdf.

2. `choose(n,k)` function (choose k things out of n).

3. `plot()` for making plots

4. We will also do arithmetic with lists. For exaple, if `x` is a list then `x^2` squares every element in the list.

5. Finally, the sample code shows how to write code for a 'for loop'.

Download the zip file

- You should have downloaded the studio2 zip file from our Canvas site.
- Unzip it in your 18.05 studio folder.
- You should see the following R files
  - studio2.r
  - studio2-samplecode.r
  - studio2-test.r

and the following other files

- studio2-instructions.pdf (this file)
- studio2-test-answers.html

Prepping R Studio

- In R studio, open studio2-samplecode.r and studio2.r
- Using the Session menu, set the working directory to source file location. (This is a good habit to develop!)
- Answer the questions in the detailed instructions just below. Your answers should be put in studio2.r
- Solution code will be posted on Saturday at 4 am

Detailed instructions for the studio

• Go through **studio2-samplecode.r** as a tutorial. In that file, it will tell you when you've done enough to code one of the problems here.

Problem 1. This problem is on the random variable $Y \sim \text{Binomial}(n, p)$. We will view Y as giving the number of heads in n tosses of a coin with probability p of heads.

Problem 1a. Here you will finish the code for the function

`studio2_problem_1a(ntosses, phead, k, ntrials)`

This function should do a simulation to estimate $P(Y = k)$ and $P(Y \leq k)$.

The arguments to this function are:

`ntosses` = n = `the number of tosses of the coin`
`pheads` = p = `the probability of heads`
`k = the value of` Y `we are interested in`
`ntrials = number of trials to run in the simulation`

You should use the built in R function `rbinom()` to generate random values for the simulation.

The output will be the estimates of the two probabilities. The lines printing the output are already in the code.

Problem 1b. Here you will finish the code for the function

`studio2_problem_1b(ntosses, phead, k)`

This function should compute the exact probability for $P(Y = k)$.

The arguments to this function are:

`ntosses` = n = `the number of tosses of the coin`
`pheads` = p = `the probability of heads`
`k = the value of` Y `we are interested in`

You should use the built in R function `choose()` in your computation. That is, implement the formula for the pmf. (Even though it's available, do not use the dbinom function. We want you to get practice with the formula.)

The output will be the value of the probability. The lines printing the output are already in the code.

Problem 2. This problem is based on a gambling game. A coin is tossed 10 times and the payoff to the player is based on the number of heads. The payoff function for k heads is

$$\$(k^2 - 7k).$$

For example, if $k = 1$ the player loses \$6 and if $k = 10$, they win \$30.

Problem 2a. Here you will finish the code for the function

`studio2_problem_2a()`

This function should plot the payoff function $k^2 - 7k$ vs. k. Since the game has 10 tosses, k should go from 0 to 10.

The function has no arguments. The code defines `ntosses = 10` for you.

The sample code will show you how to make simple plots.

Problem 2b. Here you will finish the code for the function

`studio2_problem_2b()`

We give you that phead = 0.6, and ntosses = 10. Your job is to decide whether or not the game is a good bet. To do this, you need to decide what to compute and then compute it exactly. Use your results to declare the game a good or bad bet.

For computing binomial probabilities you can use the function `dbinom()` which is described in studio2-samplecode.r.

Be sure to put the exact value you compute in the variable `exact_value`

The output say what you computed (use an English sentence for this), its value and whether or not the game is a good bet. The lines printing the output are already in the code.

Problem 2c. Here you will finish the code for the function

`studio2_problem_2c(ntrials)`

In this part you will simulate playing the game `ntrials` times and compute the simulated average payoff.

The arguments to `studio2_problem_2c(ntrials)` function are:

`ntrials = the number of times the simulation should play the game.`

As before, we give you that phead = 0.6, and ntosses = 10. You should use `rbinom()` to generate random values

The output will give the simulated average payoff. The lines printing the output are already in the code.

Problem 3. (**Optional**) This problem is about derangements, i.e. a permutation of n objects such that none of the objects ends up in its original position.

You will finish the code for the function

`studio2_problem_3(n, ntrials)`

The arguments to this function are:

```
n = the number of objects.
ntrials = the number of times the simulation should permute the objects.
```

You should use the `sample()` function to permute the numbers 1 to n and count the number of times the permutation is a derangement.

The output is the fraction of trials that produced derangements. The lines printing the output are already in the code.

Testing your code

For each problem, we ran the problem function with certain parameters. You can see the function call and the output in studio2-test-answers.html. If you call the same function with the same parameters, you should get the same results as in studio2-test-answers.html – if there is randomness involved the answers should be close but not identical.

For your convenience, the file studio2-test.r contains all the function calls used to make studio2-test-answers.html.

Before uploading your code

1. Make sure all your code is in studio2.r. Also make sure it is all inside the functions for the problems.
2. Clean the environment and plots window.
3. Source the file.
4. Call each of the problem functions with the same parameters as the test file studio2-test-answers.html.
5. Make sure it runs without error and outputs just the answers asked for in the questions.
6. Compare the output to the answers given in studio2-test-answers.html.

Upload your code

Upload your code to Gradescope.

Leave the file name as studio2.r.

You can upload more than once. We will grade the last file you upload.

Due date

Due date: The goal is to upload your work by the end of class. If you need extra time, you can upload your work any time before 6 PM ET on the day of the studio (Friday).

Solutions uploaded: Solution code will be posted on Canvas at 4 AM the day after the studio.