Studio 5 Discrete Bayesian Updating
18.05, Spring 2022

Overview of the studio

This studio explores Bayesian updating of discrete priors

R introduced in this studio

The R needed is introduced in `mit18_05_s22_studio5-samplecode.r`.

New functions: `barplot()`.

We will use `barplot()` to make what is called a stacked bar plot. You should make sure you understand how to read these plots. If you have any confusion, please ask one of the instructors or TAs.

Download the zip file

- You should have downloaded the file `mit18_05_s22_studio5.zip` from our MITx site.
- Unzip it in your 18.05 studio folder.
- You should see the following R files
  `mit18_05_s22_studio5.r`
  `mit18_05_s22_studio5-samplecode.r`
  `mit18_05_s22_studio5-test.r`

  and the following other files

  `mit18_05_s22_studio5-test-answers.html`

Prepping R Studio

- In R studio, open `mit18_05_s22_studio5-samplecode.r` and `mit18_05_s22_studio5.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 `mit18_05_s22_studio5.r`
- Solution code will be posted tomorrow at 10 pm

Detailed instructions for the studio

• Go through **mit18_05_s22_studio5-samplecode.r** as a tutorial. Pay special to Example 2.

Setup

- We have our five Platonic dice: 4, 6, 8, 12, 20 sided.
- There is a prior distribution of the quantity of each die.
- One die is chosen at random and rolled repeatedly.
- Our job is to use Bayesian updating to figure out which die was chosen.

Note. The variable `DICE = c(4,6,8,12,20)` is set at the top of `mit18_05_s22_studio5.r`. This is our list of dice for the entire studio.

Problem 0

This is a warmup to make sure the concepts are clear.

Problem 0a. Here you will finish the code for the function: `studio5_problem_0a()`

Use a cat or print statement to list all the hypotheses for the Bayesian updating.

Problem 0b. Here you will finish the code for the function: `studio5_problem_0b()`

Use a cat or print statement to list all the possible outcomes for one roll of the chosen die.

Problem 0c. Here you will finish the code for the function: `studio5_problem_0c()`

Construct and print the full likelihood table. That is a matrix giving the likelihood of every possible outcome for every possible die.

The sample code in either example should be very helpful. Remember the sample code uses just 3 dice and we now have all 5 Platonic dice.

Problem 1

This problem will do Bayesian updating of the probabilities for the type of die being rolled.

Problem 1a. There is no code to write for this problem. All you need to do is read and run the sample code for example 2.

This code shows how to choose a random die, roll it `nrolls` times and update the probabilities after each roll. It shows how to use `barplot()` to plot the posterior after each update. It also uses `barplot()` to make a single stacked barplot showing the entire sequence of posteriors.

Problem 1b. Here you will finish the code for the function
`studio5_problem_1b(prior, nrolls, plot_individual_posteriors)`

Arguments:
`prior` = prior probilities for the type of die use to generate data
`nrolls` = number of rolls to simulate
`plot_individual_posteriors` = says whether or not to make individual bar charts. When `nrolls` is large, we don't want to plot the posterior after each roll, so we set this to FALSE.

The given code fixes `random_die = 8` and then simulates `nrolls` rolls. You should not alter the die or the data.

You should finish the code by having it go through the rolls one at a time. For each roll, the probabilities for the type of die being rolled should be updated and the resulting posterior should be saved.

If `plot_individual_posterior = TRUE`, then after each update, a barplot of the posterior should be drawn.

After the last update, a stacked barplot of all the updates should be made.

Problem 1c. Here you will finish the code for the function: `studio5_problem_1c()`

Run the function `studio5_problem_1b()` twice. (Both of these are conveniently in `mit18_05_s22_studio5-test.r`

Both times set `nrolls = 20` and `plot_individual_posteriors = FALSE`.

The first time, set `prior = c(0.2, 0.2, 0.2, 0.2, 0.2)`.

The second time, set `prior = c(0.001, 0.001, 0.001, 0.001, 0.996)`.

Your code for this problem should print a few sentences, comparing and contrasting the resulting stacked bar plots. In particular, we know the chosen die is 8-sided. How do the different priors affect the number of rolls needed for the posteriors to become fairly certain of which die is being rolled?

Problem 1d. Here you will finish the code for the function: `studio5_problem_1d()`

The given code runs studio5_problem_1b() with a prior that gives the hypothesis D8 zero probability.

Your code should print out a few sentences describing what happens in the updating. Remember, we know the chosen die is 8-sided.

Optional Problem 2

All of Problem 2 is optional.

Sometimes data is censored in some way. In this problem, the data will be censored by being reported as 1 if the roll is a 1 and 0 if it is not. Our goal is still to use Bayesian updating to guess which die is being rolled.

OPTIONAL Problem 2a. Here you will finish the code for the function: `studio5_problem_2a()`

Your code should print out the following:
- A list of all the hypotheses for the Bayesian updating.

- All the possible outcomes of one roll.
- A full likelihood table for the likelihood of each outcome give each hypothesis.

OPTIONAL Problem 2b. This problem is optional. In it you will finish code for the function

`studio5_problem_2b(prior, nrolls)`

Arguments:
`prior` = prior probilities for the type of die use to generate data
`nrolls` = number of rolls to simulate

This code will simulate the entire scenario.

- Your code should pick a random die according to the given `prior`.
- It should simulate `nrolls` of that die.
- It should go through the rolls one at a time. For each roll, the probabilities for the type of die being rolled should be updated and the resulting posterior should be saved.
- After the last update, make a stacked barplot of all the updates.
- Print out the actual die chosen and the final posterior.

Do not plot the individual posteriors.

Suggestion: This problem only requires a few modifications of your code for problem 1. You need to modify the likelihood table so it is for censored data. Also, instead of fixing the `random_die = 8`, you need to choose it randomly. Finally, you need to censor the data for each roll.

Testing your code

For each problem, we ran the problem function with certain parameters. You can see the function call and the output in `mit18_05_s22_studio5-test-answers.html`. If you call the same function with the same parameters, you should get the same results as in `mit18_05_s22_studio5-test-answers.html` – if there is randomness involved the answers should be close but not identical.

For your convenience, the file `mit18_05_s22_studio5-test.r` contains all the function calls used to make `mit18_05_s22_studio5-test-answers.html`.

Before uploading your code

1. Make sure all your code is in `mit18_05_s22_studio5.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 `mit18_05_s22_studio5-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 `mit18_05_s22_studio5-test-answers.html`.

Upload your code

Use the upload link on our MITx site to upload your code for grading.

Leave the file name as `mit18_05_s22_studio5.r`. (The upload script will automatically add your name and a timestamp to the file.)

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 10 PM ET the day after the studio.

Solutions uploaded: Solution code will be posted on MITx at 10 PM the day after the studio.