

## Exercises week 1

prof. Richard Torkar

March 28, 2021

Es ist nicht genug, zu wissen, man muß auch anwenden; es ist nicht genug, zu wollen, man muß auch tun.

Johann Wolfgang von Goethe

YOU WANT TO DO THIS STEP-BY-STEP to continuously learn things. There are simply too many concepts that we will study in this course for you to learn everything the week before the exam.

Every week has exercises; you do not need to submit them, but doing them will allow you to get a better understanding of what we teach in this course. All exercises are based on McElreath's exercises, which he has online.<sup>1</sup> I will pick the ones I believe are the most important, with solutions! Each document will introduce, first, the exercises and then the solutions for all exercises (so you don't get tempted to look at the solution the first thing you do ;)

<sup>1</sup> [https://github.com/rmcelreath/stat\\_rethinking\\_2020/tree/main/homework](https://github.com/rmcelreath/stat_rethinking_2020/tree/main/homework)

EXERCISE 1. If you haven't done it, please install all software you will use in this course! Do the following,

1. Install R v4.0 or later for your platform.<sup>2</sup>
2. Install RStudio for your platform.<sup>3</sup>
3. Follow the instructions and configure your compiler etc.

<sup>2</sup> <https://ftp.acc.umu.se/mirror/CRAN/>

<sup>3</sup> <https://www.rstudio.com/products/rstudio/download/#download>

Windows: Make sure to follow the instructions carefully.<sup>4</sup>

OS X: Make sure to follow the instructions carefully.<sup>5</sup>

Linux: Make sure to follow the instructions carefully.<sup>6</sup>

<sup>4</sup> <https://github.com/stan-dev/rstan/wiki/Configuring-C---Toolchain-for-Windows>

<sup>5</sup> <https://github.com/stan-dev/rstan/wiki/Configuring-C---Toolchain-for-Mac>

<sup>6</sup> <https://github.com/stan-dev/rstan/wiki/Configuring-C-Toolchain-for-Linux>

4. Restart RStudio and then follow the instructions starting with **Installation of RStan**.<sup>7</sup>
5. We will try to use the latest version of the Stan compiler and sampler. After having restarted RStudio please enter the following in the console,

<sup>7</sup> <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>

```
devtools::install_github("stan-dev/cmdstanr")
cmdstanr::install_cmdstan()
```

6. Finally, we install the rethinking package which is the package we'll use in this course to declare our statistical models,

```
install.packages(c("coda", "mvtnorm", "devtools", "loo", "dagitty"))
devtools::install_github("rmcelreath/rethinking")
```

EXERCISE 2. Recall the globe tossing model from the chapter. Compute and plot the grid approximate posterior distribution for each of the following sets of observations. In each case, assume a uniform prior for  $p$ .

1. W,W,W
2. W,W,W,L
3. L,W,W,L,W,W,W

EXERCISE 3. Now assume a prior for  $p$  that is equal to zero when  $p < 0.5$  and is a positive constant when  $p \geq 0.5$ . Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

SOLUTION 1. I do hope everything is set up correctly now! If not, talk to me!

SOLUTION 2. Since the prior is uniform, it can be omitted from the calculations. But I'll show it here, for conceptual completeness. To compute the grid approximate posterior distribution for (1):

```
p_grid <- seq( from=0 , to=1 , length.out=100 )
# likelihood of 3 water in 3 tosses
likelihood <- dbinom( 3 , size=3 , prob=p_grid )
prior <- rep(1,100) # uniform prior
posterior <- likelihood * prior
posterior <- posterior / sum(posterior) # standardize
```

And `plot(posterior)` will produce a simple and ugly plot. This will produce something with nicer labels and a line instead of individual points:

```
plot( posterior ~ p_grid , type="l" )
```

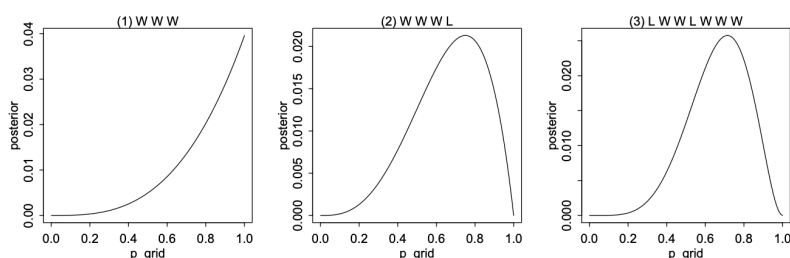
The other two data vectors are completed the same way, but with different likelihood calculations. For (2):

```
# likelihood of 3 water in 4 tosses
likelihood <- dbinom( 3 , size=4 , prob=p_grid )
```

And for (3):

```
# likelihood of 5 water in 7 tosses
likelihood <- dbinom( 5 , size=7 , prob=p_grid )
```

And this is what each plot should look like:



SOLUTION 3. Only the prior has to be changed. For the first set of observations, W W W, this will complete the calculation and plot the result:

```
p_grid <- seq( from=0 , to=1 , length.out=100 )
likelihood <- dbinom( 3 , size=3 , prob=p_grid )
prior <- ifelse( p_grid < 0.5 , 0 , 1 ) # new prior
posterior <- likelihood * prior
posterior <- posterior / sum(posterior) # standardize
plot( posterior ~ p_grid , type="l" )
```

The other two plots can be completed by changing the likelihood, just as in the previous problem. Here are the new plots, demonstrating that the prior merely truncates the posterior distribution below 0.5:

