

UPPSALA UNIVERSITY



BAYESIAN STATISTICS AND DATA ANALYSIS

Assignment 6

General information

- The recommended tool in this course is R (with the IDE R-Studio). You can download R [here](#) and R-Studio [here](#). There are many tutorials, videos and introductions to R and R-Studio online. You can find some initial hints from [RStudio Education pages](#).
- When working with R, we recommend writing the report using R markdown and the provided [R markdown template](#). The template includes the formatting instructions and how to include code and figures.
- Instead of R markdown, you can use other software to make the PDF report, but you should use the same instructions for formatting. These instructions are also available in [the PDF produced from the R markdown template](#).
- We supply a Google Colab notebook that you can also use for the assignments. We have included the installation of all necessary R packages; hence, this can be an alternative to using your own local computer. You can find the notebook [here](#). You can also open the notebook in Colab [here](#).
- Report all results in a single and *anonymous* pdf. Note that no other formats are allowed.
- The course has its own R package `bsda` with data and functionality to simplify coding. To install the package, just run the following (`upgrade="never"` skips question about updating other packages):

```
1. install.packages("remotes")
2. remotes::install_github("MansMeg/BSDA",
  subdir = "rpackage", upgrade="never")
```

- Many of the exercises can be checked automatically using the R package `markmyassignment`. you can find information on how to install and use the package [here](#). There is no need to include `markmyassignment` results in the report.
- You can find common questions and answers regarding the installation and technical problems in [Frequently Asked Questions \(FAQ\)](#).
- You can find deadlines and information on how to turn in the assignments in Studium.
- You are allowed to discuss assignments with your friends, but it is not permitted to copy solutions directly from other students or the internet. Try to solve the actual assignment problems with your code and explanations. Do not share your answers publicly. We compare the answers with the "urkund" system. We will report all suspected plagiarism.
- If you have any suggestions or improvements to the course material, please post in the course chat feedback channel, create an issue, or submit a pull request to the public repository [here](#).

- It is *mandatory* to include the following parts in all assignments (these are included already in the template):
 1. Time used for reading: How long time took the reading assignment (in hours)
 2. Time used for the assignment: How long time took the basic assignment (in hours)
 3. Good with assignment: Write one-two sentences of what you liked with the assignment/what we should keep for next year.
 4. Things to improve in the assignment: Write one-two sentences of what you think can be improved in the assignment. Can something be clarified further? Did you get stuck on stuff unrelated to the content of the assignment etc.
 - You can find information on how each assignment will be graded and how points are assigned [here](#). **Note!** This grading information can change during the course, for example, if we find errors or inconsistencies. Please feel free to comment on these grading instructions, ideally before turning in your assignment, if you think something is missing or is incorrect.
 - To pass (G) the assignment, you need 70% of the total points. To pass with distinction (VG), you need 90% of the total points. See the grading information on the point allocations for each assignment.
 - You are not allowed to show your assignments (text or code) to anyone. Only discuss the assignments with your fellow students. The student that show their assignment to anyone else could also be considered to cheat. Similarly, on zoom labs, only screen share when you are in a separate zoom room with teaching assistants.
-

Information on this assignment

This assignment is related to Chapters 10 and 11.

Reading instructions: Chapter 10 and 11 in BDA3, see reading instructions.

Reporting accuracy: For posterior statistics of interest, only report digits for which the Monte Carlo standard error (MCSE) is zero. *Example:* If you estimate $E(\mu) = 1.234$ with $\text{MCSE}(E(\mu)) = 0.01$, you should report $E(\mu) = 1.2$.

When computing the \hat{R} diagnostics, you only need to include two decimals.

Installing and using stan: To install Stan on your laptop, <https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started>. If you encounter problems, see additional answers in the [FAQ](#). Recently there have been reports of installation problems with Windows and R 4.0 (see Stan discourse for more).

Installing and using CmdStanR: If you want to use Stan in R on local computer, it can be easier to install CmdStanR interface mc-stan.org/cmdstanr/.

General information on using stan: Additional useful packages are `loo`, `bayesplot` and `shinystan`. For Python users, `PyStan`, `CmdStanPy`, and `Arviz` packages are useful.

Stan manual can be found at <https://mc-stan.org/users/documentation/>. From this website, you can also find a lot of other useful material about Stan.

R-studio enables you to automatically check the Stan syntax. Just open a stan file (with file extension `.stan` in R-studio). Then you can use the button "Check" to check the Stan syntax.

1. Generalized linear model: Bioassay with Stan

Replicate the computations for the bioassay example of section 3.7 (BDA3) using Stan.

1. Write down the model for the bioassay data in Stan syntax. For instructions in reporting your implementation, you can refer to parts 2 c) - g) in Assignment 5. More information on the bioassay data can be found in Section 3.7 of the course book. To get access to data, use the following code:

```
library(bsda)
data("bioassay")
```

Use the Gaussian prior

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \quad \text{where} \quad \boldsymbol{\mu}_0 = \begin{bmatrix} 0 \\ 10 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 2^2 & 12 \\ 12 & 10^2 \end{bmatrix}.$$

Hint! You will need Stan functions `multi_normal` and `binomial_logit` for implementing the prior and observation model, respectively. In Stan code, it is easiest to declare a variable (say `theta`) which is a two-element vector so that the first value denotes α and latter one β . This is because the `multi_normal` function that you need for implementing the prior requires a vector as an input.

2. Use \hat{R} for convergence analysis. You can either use Eq. (11.4) in BDA3 or the later version that can be found [here](#). You should specify which \hat{R} you used. In R the best choice is to use function `Rhat` from package `rstan` (see `?rstan::Rhat`). To check \hat{R} and other diagnostics, you can also call `print(fit)`, where `fit` is the fit object returned by Stan's sampling function. Report the \hat{R} values both for α and β and discuss the convergence of the chains. **Briefly explain in your own words how to interpret the obtained \hat{R} values.**
3. Plot the draws for α and β (scatter plot) and include this plot in your report. You can compare the results to Figure 3.3b in BDA3 to verify that your code gives sensible results. Notice though that the results in Figure 3.3b are generated from posterior with a uniform prior, so even when your algorithm works perfectly, the results will look slightly different (although fairly similar).