Bayesian 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 tons of tutorials, videos and introductions to R and R-Studio online. You can find some initial hints **here**.

- You can write the report with your preferred software, but the outline of the report should follow the instruction in the R markdown template that can be found **here**.

- Report all results in a single, **anonymous** *.pdf -file and return it to **peergrade.io**.

- The course has its own R package with data and functionality to simplify coding. To install the package just run the following:

    1. `install.packages("remotes")`

    2. `remotes::install_github("avehtari/BDA_course_Aalto", subdir = "rpackage")`

- Many of the exercises can be checked automatically using the R package `markmyassignment`. Information on how to install and use the package can be found **here**.

- Additional self study exercises and solutions for each chapter in BDA3 can be found **here**.

- We collect common questions regarding installation and technical problems in a course Frequently Asked Questions (FAQ). This can be found **here**.

- Deadline for all assignments are **Sunday at 23.59**.

- If you have any suggestions or improvements to the course material, please feel free to create an issue or submit a pull request to the public repository!

Information on this assignment

This exercise is related to Chapters 10 and 11. The maximum amount of points from this assignment is 6.

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

Grading instructions: The grading will be done in peergrade. All grading questions and evaluations for assignment 6 can be found **here**

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 $E(\mu) = 1.2$.

Installing and using `rstan`: See the Stan demos on how to use Stan from R. The Aalto Ubuntu desktops have the necessary libraries installed so there should be no need to install anything. To install Stan on your laptop, see the instructions below.

In R, install package `rstan`. Installation instructions on Linux, Mac and Windows can be found at `https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started`. Additional useful packages are `loo`, `bayesplot` and `shinystan` (but you don't need these in this exercise). For Python users, the `Arviz` library may be relevant.

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

1. Generalized linear model: Bioassay with Stan (6 points)

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 and in chapter 3 reading instructions **here**. To get access to data, use the following code:

```
> library(aaltobda)
> data("bioassay")
```

Use the Gaussian prior as in Assignment 4 and 5, that is

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

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`. Note that Stan gives you these values automatically; in R you can simply type `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. **This means that you should briefly explain 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).