# 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 from **RStudio Education pages**.

- Instead of installing R and RStudio on you own computer, see **how to use R and RStudio remotely**.

- When working with R, we recommend writing the report using R markdown and the provided **R markdown template**. The remplate 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 the the same instructions for formatting should be used. These instructions are available also in **the PDF produced from the R markdown template**.

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

- The course has its own R package `aaltobda` 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("avehtari/BDA_course_Aalto", subdir = "rpackage", upgrade="never")`

- 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**. There is no need to include `markmyassignment` results in the report.

- Recommended additional self study exercises for each chapter in BDA3 are listed in the course web page.

- Common questions and answers regarding installation and technical problems can be found in Frequently Asked Questions (FAQ).

- Deadlines for all assignments can be found on the course web page and in peergrade. You can set email alerts for trhe deadlines in peergrade settings.

- You are allowed to discuss assignments with your friends, but it is not allowed to copy solutions directly from other students or from internet. You can copy, e.g., plotting code from the course demos, but really try to solve the actual assignment problems with your own code and explanations. Do not share your answers publicly. Do not copy answers from the internet or from previous years. We compare the answers to the answers from previous years and to the answers from other students this year. All suspected plagiarism will be reported and investigated. See more about the **Aalto University Code of Academic Integrity and Handling Violations Thereof**.

- Do not submit empty PDFs or almost empty PDFs as these are just harming the other students as they can't do peergrading for the empty or almost empty submissions. Violations of this

rule will be reported and investigated in the same way was 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!

**Information on this assignment**

This assignment 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 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 in R (or Python) . jupyter.cs.aalto.fi has working R and RStan environment and is probably the easiest way to use RStan. The Aalto Ubuntu desktops also have the necessary libraries installed. To install Stan on your laptop, run 'install.packages("rstan")' in R. If you encounter problems, see additional answers in **FAQ**. Recently there have been reports of installation problems with Windows and R 4.0 (see Stan discourse for more), so if you don't succeed in short amount of time, it is probably easier to use jupyter.cs.aalto.fi.

If you use `jupyter.cs.aalto.fi`, all necessary packages have been pre-installed. In your laptop, 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 assignment). For Python users, `PyStan`, `CmdStanPy`, and `Arviz` packages are useful.

**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/. The course Stan demos include the same demo in RStan and CmdStanR.

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.

**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

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

**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 $\alpha$ and latter one $\beta$. This is because the `multi_normal` function that you need for implementing the prior requires a vector as an input.

2. Use $\widehat{R}$ 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 $\widehat{R}$ you used. In R the best choice is to use function `Rhat` from package `rstan` (see `?rstan::Rhat`). To check $\widehat{R}$ and other diagnostics, you can also call `monitor(fit)`, where `fit` is the fit object returned by Stan's sampling function. Report the $\widehat{R}$ values both for $\alpha$ and $\beta$ and discuss the convergence of the chains. **Briefly explain in your own words how to interpret the obtained $\widehat{R}$ values**.

3. Plot the draws for $\alpha$ and $\beta$ (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).

4. To develop the course and provide feedback to Stan developers, we collect information on which Stan setup you used and whether you had any problems in setting it up or using it. Please report,

   - Operating system (Linux, Mac, Windows) or jupyter.cs.aalto.fi?

   - Programming environment used: R or Python?

   - Interface used: RStan, CmdStanR, PyStan, or CmdStanPy?

   - Did you have installation or compilation problems?

   - Did you try first installing locally, but switched to jupyter.cs.aalto.fi?

   - In addition of these you can write what other things you found out difficult (or even frustrating) when making this assignment with Stan.