Uppsala University

Bayesian Statistics and Data Analysis

Assignment 7

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 or do additions to the assignments. 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.
- On cheating: 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. You are not allowed to use large language models, such as ChatGPT, to write assignments.
- All mathematics need to be done in digital form to simplify grading and commenting. Hence, it is not allowed to write math on paper and add an image in the assignment. If you have difficulties to write math in latex, see https://editor.codecogs.com/.

Information on this assignment

This assignment is related to Chapter 5.

Reading instructions: Chapter 5 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.

Hierarchical model: factory data with Stan

Note! Assignment 8 build upon this part of the assignment, so it is important to get this assignment correct before you start with Assignment 8.

The `factory` data in the `bsda` package contains quality control measurements from 6 machines in a factory. In the data file, each column contains the measurements for a single machine. Quality control measurements are expensive and time-consuming, so only 5 measurements were done for each machine. In addition to the existing machines, we are interested in the quality of another machine (the seventh machine). To read in the data, just use:

```
library(bsda)
data("factory")
```

For this problem, you'll use the following three Gaussian models:

- a separate model, in which each machine has its own model/parameters
- a pooled model, in which all measurements are combined and there is no distinction between machines
- a hierarchical model, which has a hierarchical structure as described in BDA3 Section 11.6.

As in the model described in the book, use the same measurement standard deviation σ for all the groups in the hierarchical model. In the separate model, however, use separate measurement standard deviation σ_j for each group j. You should use a $N(100, 100)$ for μ parameters and $N^+(0, 50)$, i.e. truncated Normal priors for σ parameters in the model.

The provided Stan code in Listing 1 given on the next page is an example of the separate model (but with very strange results, why?).

Change the Stan code to a separate model that can be summarized mathematically as:

$$y_{ij} \sim N(\mu_j, \sigma_j)$$
$$\mu_j \sim N(100, 100)$$
$$\sigma_j \sim N^+(0, 50)$$

To run Stan for that model, simply use:

```
data("factory")

sm <- rstan::stan_model(file = "[path to stan model code]")

stan_data <- list(
  y = factory,
  N = nrow(factory),
  J = ncol(factory)
)
model <- rstan::sampling(sm, data = stan_data)
model
```

```
## Inference for Stan model: 5cbfa723dd8fb382e0b647b3943db079.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##               mean se_mean   sd     2.5%     25%     50%
## mu[1]       0.11    0.01  0.98   -1.81   -0.56    0.12    0.77
## mu[2]       0.10    0.01  1.00   -1.86   -0.56    0.10    0.79
## ...
```

Note! These are *not* the results you would expect to turn in your report. You will need to change the code for the separate model code as well.

For *each of the three models* (separate, pooled, hierarchical), your tasks are the following:

a) Describe the model with mathematical notation (as is done for the separate model above). Also describe in words the difference between the three models.

b) Implement the model in Stan with the above mentioned weak priors and include the code in the report. Note that you might get divergent transitions. You should only use inference when there is no divergent transitions. To solve the issue, you can either change `adapt_delta` or reparametrize the model (see how here). If you compute the posterior correct, you should get that $\sigma_{1,separate} \approx 26.7$, $\sigma_{hiearchical} \approx 15.2$, and $\sigma_{pooled} \approx 18.8$.

c) Using the models, with the specified weakly informative priors, report the expectation/mean (and 90% uncertainty intervals) formatted in a table, comment on it and, if applicable, plot histograms for the following distributions:

    i) the posterior distribution of the mean (μ) of the quality measurements of the sixth machine

    ii) the predictive distribution $p(\tilde{y})$ for another quality measurement of the sixth machine.

    iii) the posterior distribution of the mean (μ) of the quality measurements of a seventh machine (not in the data).

d) Visualize the posterior $p(\theta|y)$ and $p(\log\tau|y)$ in the hiearchcial model, where τ is the hyperprior sd parameter and θ is hyperprior mean. Interpret these parameters.

e) Visualize the posterior $p(\mu_3, \log\tau|y)$, where τ is the hyperprior sd parameter. Describe the distribution and reason about the 'funnel' shape. What does it mean for the relationship between μ_3 and $\log\tau$?

f) Now change the prior for the σ, residual parameter(s), to a Gamma(1, 1). What happens? Why? Report the mean and 90% CI for the mean (μ) of the quality measurements of the sixth machine.

Listing 1: Stan code for a bad separate model

```
data {
  int<lower=0> N;
  int<lower=0> J;
  vector[J] y[N];
}

parameters {
  vector[J] mu;
  vector<lower=0>[J] sigma;
}

model {
  // priors
  for (j in 1:J){
    mu[j] ~ t(10, 10, 10);
    sigma[j] ~ inv_chi_square(10);
  }

  // likelihood
  for (j in 1:J)
    y[,j] ~ normal(mu[j], sigma[j]);
}

generated quantities {
  real ypred;
  // Compute predictive distribution
  // for the first machine
  ypred = normal_rng(mu[1], sigma[1]);
}
```