Assignment 8

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 7.

Note! This assignment build upon assignment 7, so be sure that assignment 7 is correct before you start with this assignment. You can ask TAs for help to check that your assignment 7 answer is good.

Reading instructions: Chapter 7 in BDA3, see reading instructions. The ['loo' package vignette on the basics of LOO](https://mc-stan.org/loo/articles/loo2-with-rstan.html) shows an example of how to modify Stan code and use the package with Stan models. Also read about the PSIS-LOO that in **this article**.

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 $\mathrm{MCSE}(E(\mu)) = 0.01$, you should report $E(\mu) = 1.2$.

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

1. Model assessment: PSIS LOO-CV for factory data with Stan

Use leave-one-out cross-validation (LOO-CV) to assess the predictive performance of the pooled, separate and hierarchical Gaussian models for the factory dataset (see the second exercise in Assignment 7). To read in the data, just use:

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

PSIS-LOO is a recently developed method for approximating the exact LOO and is thus not in BDA3. For more information, see the original paper **here** or **here**.

Use Stan for fitting the models, and the `loo` R package for computing the approximate LOO-CV given the posterior samples provided by Stan. You can install the package as

```
install.packages("loo")
```

Python users can use PSIS-LOO implementation in ArviZ library.

The report should include the following parts.

1. Fit the models with Stan as instructed in Assignment 7. With the $N(100, 100)$ prior for the μ and θ parameters and the $N(0, 50)$ prior σ and τ parameters.

   To use the `loo` or `psisloo` functions, you need to compute the log-likelihood values of each observation for every posterior draw (i.e. an S-by-N matrix, where S is the number of posterior draws and $N = 30$ is the total number of observations). This can be done in the `generated quantities` block in the Stan code.

2. Compute the PSIS-LOO elpd values and the $\hat{k}$-values for each of the three models.

   Hint 1! The elpd values for all models should be between -100 and -150.

   Hint 2! It will be convenient to visualize the $\hat{k}$-values for each model so that you can easily see how many of these values fall in the range $\hat{k} > 0.7$ to assess the reliability of the PSIS-LOO estimate for each model. You can read more about the theoretical guarantees for the accuracy of the estimate depending on $\hat{k}$ from the original article (see **here** or **here**), but regarding this assignment, it suffices to understand that if all the $\hat{k}$-values are $\hat{k} \lesssim 0.7$, the PSIS-LOO estimate can be considered to be reliable, otherwise there is a concern that it may be biased (too optimistic, overestimating the predictive accuracy of the model).

3. Compute the effective number of parameters p_{eff} for each of the three models. Interpret the p_{eff} and compare it with the number of estimated parameters. Does p_{eff} make sense? Why or why not?

   Hint! The estimated effective number of parameters in the model can be computed from equation (7.15) in the book, where $\text{elpd}_{\text{loo-cv}}$ is the PSIS-LOO value (sum of the LOO log densities) and lpd is given by equation (7.5) in the book.

4. Assess how reliable the PSIS-LOO estimates are for the three models based on the $\hat{k}$-values.

5. An assessment of whether there are differences between the models with regard to the $\text{elpd}_{\text{loo-cv}}$, and if so, which model should be selected according to PSIS-LOO.

6. Both the Stan and R code should be included in your report.

2. Model assessment: Standard LOO-CV for factory data with Stan (1.5 points, only required for VG)

PSIS LOO-CV approximates standard leave out cross validation. To compare the two, implement standard leave one out cross validation for the separate model.

1. Include your code

2. Report the elpd$_{\text{loo-cv}}$ value for the separate model

3. Compare the value to the one obtained via PSIS-LOO.