36-463/663: Multilevel & Hierarchical Models
Spring 2022
HW05 – Due Tue Mar 15, 11:59pm

- Please turn the homework in, as a single pdf, online in GradeScope using the link provided on the Assignment page on canvas.cmu.edu. Upload *one* file per person.
- Reading for the week after spring break:
  - See the subfolder "reading" in the week07 and week08 folders on Canvas.
- There are 3 main exercises, all with "parts"...

Exercises

1. Install and verify the software package Stan and the library rstan, on your personal computer. Instructions for doing this are appended at the end of this hw assignment. Then run the following short example to verify that rstan is intalled and working correctly (note that the text string assigned to the variable "model" runs over several lines; that's perfectly OK):

```
text_model <- "
   data {
     real y_mean;
   }
   parameters {
     real y;
   }
   model {
     y ~ normal(y_mean,1);
   }"
compiled_model <- stan(model_code=text_model,data=list(y_mean=0),chains=0)
fit <- stan(fit=compiled_model,data=list(y_mean=0))
fit2 <- stan(fit=compiled_model,data=list(y_mean=5))
```

This Stan program is not doing anything Bayesian, it's just making draws from a normal distribution with mean=0 and sd=1 (fit) or mean 5 and sd=1 (fit2).

(a) Turn in the output for both `print(fit)` (or just `fit`), and `summary(fit)`. Write a sentence or two explaining the difference between the two outputs (besides a different number of decimal places printed!).

(b) Use commands like those in the appendix (from `library(bayesplot)`) of lecture 13 to produce and turn in, for the fitted model `fit`:

- Trace plots of `y` (the variable being simulated) and `lp__` (the value of the log-posterior (in this case, not a posterior but just the normal density) at each simulated value of `y`).
- Autocorrelation plots of `y` and `lp__`.
- A histogram of just `y`.

(c) Use the `extract` function to extract simulated values for `y` from `fit` and make a qqplot of the values against the Normal distribution. Is it plausible that the simulated values for `y` are normally distributed?

(d) Print out and turn in output from `print(fit)` and `print(fit2)`. Write a sentence noting any similarities or differences between them.

(e) Use the `extract` function to extract simulated values for `y` from `fit2` and make a qqplot of the values against the Normal distribution. Is it plausible that the simulated values for `y` are normally distributed?

2. Problem 2(c)(iv) on hw #04 asked you to generate a point estimate and 95% CI for the model

$$\begin{aligned} y_i &\sim Poiss(\lambda),\ i = 1, 2, \ldots, n \\ \lambda &\sim Gamma(\alpha, \beta) \end{aligned}$$

with $\alpha = 1$ and $\beta = 1$, for the $n = 16$ data values

$$3, 7, 1, 7, 6, 1, 5, 4, 6, 2, 11, 3, 1, 6, 4, 4$$

(a) Write a stan program that specifies the likelihood and prior for this model, and use it to simulate draws from the posterior. You can use the default number of chains (4) and samples in each chain (2000). Include in your solutions:

- Your stan program
- your R code
- the output from `print()`'ing fitted stan simulation.

Hint: You will find the documentation at

`https://mc-stan.org/docs/2_29/functions-reference/index.html`

useful, especially the sections on Discrete and Continuous Distributions. You may also find this documentation useful

`https://mc-stan.org/docs/2_29/stan-users-guide/index.html`

`https://mc-stan.org/docs/2_29/reference-manual/index.html`

and, of course, you should imitate examples given in class to save yourself time, wherever possible.

(b) Use functions in `library(bayesplot)` to produce and turn in graphs of

- trace plots
- acf plots
- rhat plots
- neff (ratio) plots

for your fitted simulation.

(c) Use functions in `library(bayesplot)` to produce and turn in graphs of

- A 95% CI for λ
- A density plot for λ, with the 95% CI shaded in underneath it.

Compare your answer here with the answers you got on hw04 problem #2.

3. In the files area of Canvas there is a folder "presentation project ideas".

(a) Read the "overview.pdf" file in that folder, and browse through the subfolders there[1]. If you might have a project of your own to present, think about that too. Turn in your first, second and third choices for projects (these are tentative, not final, choices!). Also, indicate tentative team members if you are organizing a team.

(b) Sign up for a time to meet me in the week after spring break to talk about your presentation project. (You don't have to turn anything in for this, but it will be counted as part of your project grade.)

[1] If there is only a readme.txt file there now, it means I am still organizing and uploading content. It should all be there by Wednesday evening.

Installing Stan for R

1. Go to `https://mc-stan.org/users/interfaces/rstan`
   (can also find by googling "rstan").

2. Click on the "RStan Quick Start Guide" under "Download and Get Started"

   (*Note:* We will be installing Stan to run in R. There are also installations for python, matlab, julia, stata, mathematica, scala, etc… If you are interested, see `https://mc-stan.org/` for details.)

3. Follow the instructions for configuring the C++ Toolchain on your computer

   - This is the trickiest part of installing Stan in R on your computer.
   - There are separate instructions for Windows, Mac, & Linux
   - Make sure you are using R version 4.0 or higher. You may want to update Rstudio to the current version as well (you need at least version 1.2.5042).
   - **MAKE SURE YOU ARE** following the instructions for your version of R (version 4.0 or higher) not an earlier version of R such as version 3.6, etc.

4. Follow the instructions under "Installation of RStan" and "Verify Installation"

   - The most important parts are the "install.packages" command, to actually do the installation, and the "example" command to verify that your installation is good. If you can get these two commands to run successfully, you are golden.
     - Note: the "example" command will take from several seconds to up to a minute or so to run. It will also generate a ton of output that may not make much sense right now… Be patient…
   - Please note the other instructions under these two headings, which handle some common special cases that you may need to deal with on your laptop.

5. After you have verified that your installation is good, feel free to browse through the longer "How to Use RStan" section. As you get more accustomed to Stan, you'll see that the Stan modeling language is a kind of extension of R's usual programming language – much more general and flexibe than `lmer()`'s modeling language.

6. Check out `https://mc-stan.org/` for background info on stan.