Bayesian 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 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**.
- 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 Chapter 5. The maximum amount of points from this assignment is 6.

Reading instructions: Chapter 5 in BDA3, see reading instructions **here**.

Grading instructions: The grading will be done in peergrade. All grading questions and evaluations for assignment 7 can be found **here**

Reporting accuracy: As many significant digits as justified by the Monte Carlo error and posterior accuracy.

Installing and using `rstan`: See the Stan demos on how to use Stan from R. The university 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. Linear model: drowning data with Stan (3p)

The provided data `drowning` in the `aaltobda` package contains the number of people drown per year in Finland 1980–2016. A statistician are going to fit a linear model with Gaussian noise to these data using time as the predictor and number of drownings as the target variable (see the related linear model example for the Kilpisjärvi-temperature data in the example Stan codes). She has two objective questions:

i) What is the trend of the number of people drown per year? We would plot the histogram of the slope of the linear model.

ii) What is the prediction for the year 2019? We would plot the histogram of the posterior predictive distribution for the number of people drowning at $\tilde{x} = 2019$.

Your task is to fix the stan code to be able to run this linear regression model.

To access the data, use:

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

1. The provided Stan code in Listing 1 given on the next page is almost correct for the given problem. However, there are two crucial mistakes. Find these two mistakes and fix them. Report the original mistakes and your fixes clearly in your report. Include the *full* Stan code in your report.

2. The provided broken code does not define any prior for the parameters. In Stan, this corresponds to using a uniform prior. In addition to the two fixes discussed above, we would like to apply a weakly-informative prior $\mathrm{N}(0, \tau^2)$ for the slope parameter `beta` into the code. It is very unlikely that the mean number of drownings changes more than 50 % in one year. The approximate historical mean yearly number of drownings is 138. Hence, set τ so that the following holds for the prior probability for `beta`: $\Pr(-69 < \texttt{beta} < 69) = 0.99$. Determine suitable value for τ and report the approximate numerical value for it in the report.

3. Using the obtained τ, implement the desired prior in the Stan code. In the report, in a separate section, indicate clearly how you carried out your prior implementation, e.g. "Added line … in block …".

   Hint! Example resulting plots for the problem, with the fixes and the desired prior applied, are shown in Figure 1. If you want, you can use these plots as a reference for testing if your modified Stan code produces similar results. However, running the inference and comparing the plots is not required.

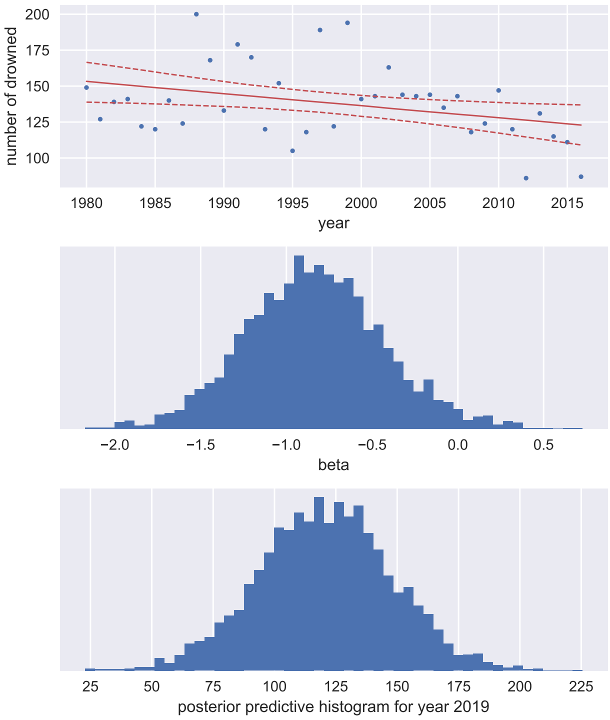

Figure 1: Example plots for the results obtained for problem in the question 1. In the first subplot, the red lines indicate the resulting 5 %, 50 %, and 95 % posterior quantiles for the transformed parameter `mu` at each year.

Listing 1: Broken Stan code for question 1

```
data {
    int<lower=0> N;  // number of data points
    vector[N] x;     // observation year
    vector[N] y;     // observation number of drowned
    real xpred;      // prediction year
}
parameters {
    real alpha;
    real beta;
    real<upper=0> sigma;
}
transformed parameters {
    vector[N] mu;
    mu = alpha + beta*x;
}
model {
    y ~ normal(mu, sigma);
}
generated quantities {
    real ypred;
    ypred = normal_rng(mu, sigma);
}
```

2. Hierarchical model: factory data with Stan (3p)

Note! Both Assignment 8 and 9 build upon this assignment, hence it is important to get this assignment correct before you start with assignment 8 and 9.

The `factory` data in the `aaltobda` package contains quality control measurements from 6 machines in a factory (units of the measurements are irrelevant here). 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(aaltobda)
> data("factory")
```

Implement a separate, a pooled, and a hierarchical Gaussian model described in Section 11.6 using Stan. In the pooled model, all the measurements are combined and no distinction is made between the machines. In the separate model, each machine has its own model. Similarly, 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 weakly informative priors for all your model.

The provided Stan code in Listing 2 given on the next page is an example the separate model (but with very strange results, why?). This separate model can be summarized mathematically as:

$$\begin{aligned} y_{ij} &\sim N(\mu_j, \sigma_j) \\ \mu_j &\sim N(0, 1) \\ \sigma_j &\sim Inv - \chi^2(10) \end{aligned}$$

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* results you would expect to turn in in your report. You will need to change the separate model as well.

Using each of the three models with weakly informative priors– separate, pooled, and hierarchical – report, comment and, if applicable, plot histogram for the following distributions:

i) the posterior distribution of the mean of the quality measurements of the sixth machine.

ii) the predictive distribution for another quality measurement of the sixth machine.

iii) the posterior distribution of the mean of the quality measurements of the seventh machine.

The report should (at least) contain:

a) The plots used to answer question i)-iii) above.

b) Stan models as model code that use weakly informative priors for all your models.

c) Each model described with mathematical notation (as is done for the separate model above). Describe in words the difference between the three models.

d) Include the posterior expectation for μ_1 with a 90% credibility interval for all three models but using a $N(0, 10)$ prior for the μ paramater(s) and a $\text{Gamma}(1, 1)$ prior for the σ parameter(s). In the hierarchical model, use the $N(0, 10)$ and $\text{Gamma}(1, 1)$ as hyper-prior.

Hint! See the example Stan-codes **here** for the comparison of k groups with and without the hierarchical structure.

Listing 2: Stan code for a bad seperate 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] ~ normal(0, 1);
    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]);
}
```