Hierarchical Bayesian models 2

Answers for this lab must be submitted on Moodle before March 31st at 5pm.

> **Tip**: In RMarkdown you can add the argument `cache = TRUE` to a block of code (`{r, cache = TRUE}`) to save the result of the block. In this case, as long as the code remains the same, the calculation is not repeated every time the RMarkdown document is compiled. This function is especially useful for time-consuming operations, such as fitting a Bayesian model with `brm`.

Data

We will use the *gapminder* dataset presented during the exercises on robust regression (lab 4). This data frame includes life expectancy (*lifeExp*), population (*pop)* and GDP per capita (*gdpPercap*) for 142 countries and 12 years (every 5 years between 1952 and 2007).

```r
library(gapminder)
data(gapminder)
str(gapminder)
```

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

As in lab 4, we first transform the predictors:

- *gdp_norm* is the logarithm of *gdpPercap*, scaled to have a mean of 0 and a standard deviation of 1.
- *dyear* is the number of years since 1952.

```r
library(dplyr)
gapminder <- mutate(gapminder, gdp_norm = scale(log(gdpPercap)),
                    dyear = year - 1952)
```

Bayesian model of life expectancy as a function of GDP and time

In lab 4, we first performed a linear regression of `lifeExp` as a function of `gdp_norm`, `dyear` and their interaction. For this first part, we will estimate these same effects in a Bayesian context, by adding random effects of the country on the intercept and the coefficients of `gdp_norm` and `dyear`.

Notes:

- The model formula in `brm` follows the same syntax as `lmer` for the specification of fixed and random effects.
- Although it would be possible to add a random effect of country on the `gdp_norm:dyear` interaction, we omit it here in order to reduce model computation time.

a) Choose prior distributions for the model parameters described above. Here is a code sample where only the specification of the distributions is missing. The first four lines define the prior distributions for the intercept and coefficients of the three fixed effects, the next three define the distributions for the standard deviations of the random effects (`class = "sd"`), while the last line refers to the standard deviation of the individual observations (`class = "sigma"`).

```
gap_prior <- c(set_prior("", class = "Intercept"),
               set_prior("", class = "b", coef = "gdp_norm"),
               set_prior("", class = "b", coef = "dyear"),
               set_prior("", class = "b", coef = "gdp_norm:dyear"),
               set_prior("", class = "sd", coef = "Intercept", group = "country"),
               set_prior("", class = "sd", coef = "gdp_norm", group = "country"),
               set_prior("", class = "sd", coef = "dyear", group = "country"),
               set_prior("", class = "sigma"))
```

It is recommended to choose normal distributions in all cases. For "sigma" and "sd", these distributions will be interpreted as half-normal because it is implied that these parameters are ≥ 0. To choose the μ and σ values for each normal distribution, consider the interpretation of each parameter and in particular the scales of the predictors `gdp_norm` and `dyear`.

- For the effect of the interaction, we can assume that it is not stronger than the main effects of the two predictors, so `gdp_norm:year` can take the same prior distribution as the smallest assumed effect between `gdp_norm` and `year`.
- As for the standard deviations of the random effects ("sd"), their prior distribution can have the same width as that of the corresponding coefficient "b".

b) Now draw a sample of the joint prior distribution of the parameters with `brm`. I suggest specifying `chains = 1, iter = 1500, warmup = 1000` to produce a single Markov chain with 1000 warmup iterations and 500 sampling iterations. Then visualize the predicted distribution of `lifeExp` for each iteration of the prior parameters.

Due to the large number of estimated effects and the fact that we impose only weak constraints on each prior distribution, extreme or even impossible values (large positive and negative values) are to be expected; the important thing is that the density is greater within a realistic range of values. It may be useful to "zoom" into a part of the `ggplot` graph by adding `coord_cartesian(xlim = c(..., ...), ylim = c(..., ...))` with limits in x and y.

c) Now fit the model with `brm`. You can reduce the number of Markov chains to 2 to save time, but keep the default values for the number of iterations. (You can ignore the warning that the effective sample size or ESS is small.) How can you assess the convergence of the model?

2. Robust regression with the t distribution

In lab 4, we saw that a robust regression was preferable for this dataset. In order to allow more extreme residuals in a Bayesian context, we will replace the normal distribution for the residuals by a Student t distribution.

a) Refit the previous model by adding the argument `family = student` in `brm`. This argument indicates that the residuals normalized by `sigma`, $(y - \hat{y})/\sigma$, follow a t distribution with ν degrees of freedom.

Keep the same prior distributions for all parameters of the model. Let `brm` choose a prior distribution for ν (`nu`). By calling the `prior_summary` function from the fitted model, can you determine what this prior distribution is by default?

b) Describe the main differences between the parameter estimates from this model, compared with those of the model in part 1.

c) Compare the fit of the two models with the PSIS-LOO method. In the context of this assignment, you can ignore the high values of k (in practice, cross-validation should be performed with `reloo = TRUE`).

d) Apply `predict` to the fitted model to obtain the mean, standard deviation and 95% credibility interval for the posterior prediction for each point in the data frame and attach these predictions to the original data set with `cbind`. Select a few countries from the dataset and illustrate the observations, predictions of the two models and their credibility intervals.