Modèles hiérarchiques bayésiens 2

> **Truc**: Dans RMarkdown, vous pouvez ajouter l'argument `cache = TRUE` à un bloc de code (`{r, cache = TRUE}`) pour enregistrer le résultat du bloc. Dans ce cas, tant que le code reste le même, le calcul n'est pas répété à chaque compilation du document RMarkdown. Cette fonction est particulièrement utile pour les opérations qui prennent beaucoup de temps, comme l'ajustement d'un modèle bayésien avec `brm`.

Données

Nous utiliserons le jeu de données *gapminder* présenté lors des exercices sur la régression robuste (labo 4). Ce tableau inclut l'espérance de vie (*lifeExp*), la population (*pop)* et le PIB par habitant (*gdpPercap*) pour 142 pays et 12 années (aux 5 ans entre 1952 et 2007).

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

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1704 obs. of  6 variables:
##  $ 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  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 16317921 22
##  $ gdpPercap: num  779 821 853 836 740 ...
```

Comme pour le labo 4, nous transformons d'abord les prédicteurs:

- *gdp_norm* est le logarithme de *gdpPercap*, normalisé pour avoir une moyenne de 0 et un écart-type de 1.
- *dyear* est le nombre d'années écoulées depuis 1952.

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

1. Modèle bayésien de l'espérance de vie en fonction du PIB et du temps

Dans le labo 4, nous avions d'abord effectué une régression linéaire de `lifeExp` en fonction de `gdp_norm`, `dyear` et leur interaction. Pour cette première partie, nous estimerons ces mêmes effets dans un contexte bayésien, en ajoutant des effets aléatoires du pays sur l'ordonnée à l'origine et sur les coefficients de `gdp_norm` et `dyear`.

Notes:

- La formule du modèle dans `brm` suit la même syntaxe que `lmer` pour la spécification des effets fixes et aléatoires.
- Bien qu'il serait possible d'ajouter un effet aléatoire du pays sur l'interaction `gdp_norm:dyear`, nous l'omettons ici afin de réduire le temps de calcul des modèles.

(a) Choisissez des distributions *a priori* pour les paramètres du modèle décrit ci-dessus. Voici un exemple de code où il ne manque que la spécification des distributions. Les quatre premières lignes définissent les distributions *a priori* pour l'ordonnée à l'origine et les coefficients des trois effets fixes, les trois suivantes définissent les distributions pour les écarts-types des effets aléatoires (`class = "sd"`), tandis que la dernière réfère à l'écart-type des observations individuelles (`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"))
```

Il est recommandé de choisir des distributions normales dans tous les cas. Pour "sigma" et "sd", ces distributions seront interprétées comme des demi-normales car il est sous-entendu que ces paramètres sont ≥ 0. Pour choisir la moyenne et l'écart-type de chaque distribution normale, considérez l'interprétation de chaque paramètre et en particulier les échelles des prédicteurs `gdp_norm` et `dyear`.

- Pour l'effet de l'interaction, on peut supposer que celui-ci n'est pas plus fort que les effets principaux des deux prédicteurs, donc `gdp_norm:year` peut prendre la même distribution *a priori* que l'effet supposé le plus petit entre `gdp_norm` et `year`.
- Quant aux écarts-types des effets aléatoires ("sd"), leur distribution *a priori* peut avoir la même largeur que celle du coefficient "b" correspondant.

b) Tirez maintenant un échantillon de la distribution conjointe *a priori* des paramètres avec `brm`. Je suggère de spécifier `chains = 1, iter = 1500, warmup = 1000` pour produire une seule chaîne de Markov avec 1000 itérations de rodage et 500 itérations d'échantillonnage. Visualisez ensuite la distribution de `lifeExp` prédite pour chaque itération des paramètres *a priori*.

En raison du grand nombre d'effets estimés et du fait que nous n'imposons que des contraintes légères sur chaque distribution *a priori*, on doit s'attendre à des valeurs extrêmes voire impossibles (grandes valeurs positives et négatives); l'important est que la densité soit plus grande dans une plage de valeurs réalistes. Il peut être utile de faire un "zoom" sur une partie du graphique `ggplot` en y ajoutant `coord_cartesian(xlim = c(..., ...), ylim = c(..., ...))` avec des limites en x et y.

c) Ajustez maintenant le modèle avec `brm`. Vous pouvez réduire le nombre de chaînes de Markov à 2 pour sauver du temps, mais conservez les valeurs par défaut pour le nombre d'itérations. (Vous pouvez ignorer l'avertissement selon lequel la taille effective de l'échantillon ou ESS est faible.) Comment pouvez-vous évaluer la convergence du modèle?

2. Régression robuste avec la distribution t

Dans le labo 4, nous avions vu qu'une régression robuste était préférable pour ce jeu de données. Afin de permettre des résidus plus extrêmes dans un contexte bayésien, nous allons remplacer la distribution normale pour les résidus par une distribution t de Student.

a) Réajustez le modèle précédent en ajoutant l'argument `family = student` dans `brm`. Cet argument indique que les résidus normalisés par `sigma`, $(y - \hat{y})/\sigma$, suivent une distribution t avec ν degrés de liberté.

Conservez les mêmes distributions *a priori* pour tous les paramètres du modèle. Laissez `brm` choisir une distribution *a priori* pour ν (`nu`). En appelant la fonction `prior_summary` à partir du modèle ajusté, pouvez-vous déterminer quelle est cette distribution *a priori* par défaut?

b) Décrivez les principales différences entre les estimés des paramètres de ce modèle, comparés à ceux du modèle en 1.

c) Comparez l'ajustement des deux modèles avec la méthode PSIS-LOO. Dans le contexte de ce devoir, vous pouvez ignorer les valeurs élevées de k (dans la pratique, il faudrait effectuer la validation croisée avec `reloo = TRUE`).

d) Appliquez `predict` au modèle ajusté pour obtenir la moyenne, l'écart-type et l'intervalle à 95% pour la prédiction *a posteriori* de chaque point du tableau de données et rattachez ces prédictions au jeu de données original avec `cbind`. Choisissez quelques pays du jeu de données et illustrez les observations, les prédictions des deux modèles et leurs intervalles de crédibilité.