Introduction à l'analyse bayésienne

Données

Déjà utilisé pour les exercices sur le maximum de vraisemblance, le tableau de données thermal_range.csv représente le résultat d'une expérience visant à déterminer l'effet de la température (*temp*) sur le nombre d'oeufs (*num_eggs*) produits par une espèce de moustique. Trois réplicats ont été mesurés pour des valeurs de température entre 10 et 32 degrés Celsius.

```
therm <- read.csv("../donnees/thermal_range.csv")
head(therm)
```

```
##   temp num_eggs
## 1   10        1
## 2   10        1
## 3   10        2
## 4   12        4
## 5   12        4
## 6   12        6
```

Estimation bayésienne du modèle d'optimum thermique

Rappelons-nous le modèle utilisé précédemment pour ce jeu de données. Le nombre moyen d'oeufs produits est donné par une courbe gaussienne:

$$N = N_o \exp\left(-\frac{(T - T_o)^2}{\sigma_T^2}\right)$$

Dans cette équation, T_o est la température optimale, N_o est le nombre d'oeufs produits à cet optimum et σ_T représente la tolérance autour de l'optimum (plus σ_T est élevé, plus N décroit lentement autour de l'optimum).

a) Il est possible d'estimer les paramètres d'un modèle non-linéaire comme celui-ci dans *brms*. Par exemple:

```
brm(bf(num_eggs ~ No * exp(-((temp-To)/sigmaT)^2), No + To + sigmaT ~ 1, nl = TRUE),
    data = therm)
```

Note:

- Il faut entourer la formule dans une fonction `bf` et spécifier l'argument `nl = TRUE` (non-linéaire).
- Après la formule non-linéaire du modèle, il faut ajouter un terme décrivant les paramètres. Ici, `No + To + sigmaT ~ 1` signifie seulement que nous estimons un effet fixe pour chaque paramètre. Si un des paramètres variait en fonction d'une variable de groupe, nous pourrions écrire par exemple `No ~ (1|groupe), To + sigmaT ~ 1`.

Puisque nous allons utiliser une distribution binomiale négative avec un lien logarithmique pour représenter la moyenne de la réponse (`family = negbinomial` dans *brms*), nous devons modifier la formule ci-dessus pour représenter le logarithme du nombre d'oeufs moyen N. Ré-écrivez la fonction `bf` en appliquant cette transformation.

b) Choisissez des distributions *a priori* appropriées pour trois paramètres de l'équation obtenue précédemment. Dans l'instruction `set_prior`, le nom du paramètre est spécifié ave `nlpar` pour un modèle non-linéaire. Par exemple, `set_prior("normal(0, 1)", nlpar = "To")` assigne une distribution normale centrée réduite au paramètre `To`.

Note: N'oubliez pas de spécifier la borne inférieure pour `sigmaT`.

Ajoutez aussi une distribution *a priori* pour le paramètre θ de la distribution binomiale négative avec `set_prior("gamma(2, 0.1)", class = "shape")`. Vous pouvez visualiser cette distribution dans R avec `plot(density(rgamma(1E5, 2, 0.1))`. Puisque la variance de la distribution binomiale négative est de $\mu + \mu^2/\theta$, où μ est la moyenne, nous voulons éviter les valeurs de θ trop proches de zéro. Avec les paramètres spécifiés, θ est petit pour des valeurs proches de 0 et plus grandes que 50 (avec un θ si grand, la distribution binomiale négative rejoint pratiquement celle de Poisson).

c) Ajustez avec `brm` le modèle non-linéaire avec la formule et les distributions *a priori* spécifiées dans les parties précédentes, en utilisant une distribution binomiale négative de la réponse. Visualisez la forme de la fonction N vs. T estimée avec `marginal_effects`. Déterminez la valeur moyenne et l'intervalle de crédibilité à 95% pour la distribution *a posteriori* de chaque paramètre.

d) Comparez les résultats en (c) aux estimés et intervalles de confiance obtenus dans le laboratoire 3 par le maximum de vraisemblance, reproduits dans le tableau ci-dessous.

Paramètre	Estimé	Intervalle
N_o	123.2	(104.2, 147.2)
T_o	23.9	(23.4, 24.5)
σ_T	6.82	(6.33, 7.42)
k	0.103	(0.059, 0.186)

Note: Le paramètre k correpond à $1/\theta$ pour la distribution binomiale négative.

e) Vérifiez les intervalles de prédictions *a posteriori* avec `pp_check(..., type = "intervals")`. Les observations semblent-elles cohérentes avec le modèle ajusté?

f) Comme nous verrons la semaine prochaine, le programme *Stan* sur lequel se base *brms* produit un échantillon de la distribution *a posteriori* conjointe des paramètres du modèle. Par défaut, cet échantillon compte 4000 jeux de paramètres. La fonction `posterior_epred` de *brms* calcule la prédiction moyenne selon chacun de ces jeux de paramètres pour un nouveau jeu de données donné par l'argument `newdata`, un peu comme la fonction `predict` dans le cas des modèles de régression.

Utilisez la fonction `posterior_epred` pour calculer le rapport entre la production d'oeufs moyenne à 25 degrés C comparée à celle à 20 degrés C, ainsi qu'un intervalle de crédibilité à 95% pour ce rapport.