Introduction à la modélisation statistique bayésienne

Un cours en R et Stan avec brms

Ladislas Nalborczyk (LPC, LNC, CNRS, Aix-Marseille Univ)

Planning

Cours n°01 : Introduction à l'inférence bayésienne
Cours n°02 : Modèle Beta-Binomial
Cours n°03 : Introduction à brms, modèle de régression linéaire
Cours n°04 : Modèle de régression linéaire (suite)
Cours n°05 : Markov Chain Monte Carlo
Cours n°06 : Modèle linéaire généralisé
Cours n°07 : Comparaison de modèles
Cours n°08 : Modèles multi-niveaux
Cours n°09 : Modèles multi-niveaux généralisés
Cours n°10 : Data Hackathon

Introduction

Cinq problèmes, cinq jeux de données. Le but est de comprendre et d'analyser ces données pour répondre à une (ou plusieurs) question(s) théorique(s).

Vous devrez écrire le modèle mathématique, puis fitter ce modèle en utilisant `brms`.

Ensuite, vous devrez évaluer le modèle, interpréter les résultats, et écrire un paragraphe de résultats (de type article) pour décrire vos analyses et vos conclusions.

Les problèmes sont classés par ordre croissant de difficulté. Vous pouvez travailler individuellement ou par groupe, et des propositions de correction sont disponibles à la suite des énoncés.

Problème n°1

Peut-on prédire la taille d'un individu par la taille de ses parents ?

```r
library(tidyverse)
library(imsb)

d1 <- open_data(parents)
head(d1, 10)
```

```
   gender height mother father
1       M   62.5     66     70
2       M   64.6     58     69
3       M   69.1     66     64
4       M   73.9     68     71
5       M   67.1     64     68
6       M   64.4     62     66
7       M   71.1     66     74
8       M   71.0     63     73
9       M   67.4     64     62
10      M   69.3     65     69
```


Problème n°2

Les données suivantes documentent le naufrage du titanic. La colonne `pclass` indique la classe dans laquelle chaque passager voyageait (un proxy pour le statut socio-économique), tandis que la colonne `parch` indique le nombre de parents et enfants à bord.

Peut-on prédire la survie d'un passager grâce à ces informations ?

```
1  d2 <- open_data(titanic)
2  head(d2, 10)
```

```
   survival pclass gender age parch
1         0  upper   male  22     0
2         1  lower female  38     0
3         1  upper female  26     0
4         1  lower female  35     0
5         0  upper   male  35     0
6         0  lower   male  54     0
7         0  upper   male   2     1
8         1  upper female  27     2
9         1  upper female   4     1
10        1  lower female  58     0
```


Problème n°3

Ce jeu de données recense des informations sur le diamètre (colonne `diam`) de 80 pommes (chaque pomme étant identifiée par la colonne `id`), poussant sur 10 arbres différents (colonne `tree`). On a mesuré ce diamètre pendant 6 semaines successives (colonne `time`).

Que peut-on dire de la pousse de ces pommes, tout en considérant les structures de dépendance existant dans les données (i.e., chaque pomme poussait sur un arbre différent) ?

```r
d3 <- open_data(apples)
head(d3, 10)
```

```
   tree apple id time diam
1     1     1  1    1 2.90
2     1     1  1    2 2.90
3     1     1  1    3 2.90
4     1     1  1    4 2.93
5     1     1  1    5 2.94
6     1     1  1    6 2.94
7     1     4  4    1 2.86
8     1     4  4    2 2.90
9     1     4  4    3 2.93
10    1     4  4    4 2.96
```


Problème n°4

Ces données recensent le nombre de candidatures pour 6 départements (colonne `dept`) à Berkeley (données disponibles dans le paquet `rethinking`). La colonne `admit` indique le nombre de candidatures acceptées et la colonne `reject` le nombre de candidatures rejetées (la colonne `applications` est simplement la somme des deux), en fonction du sexe des candidats (`applicant.gender`).

On veut savoir s'il existe un biais lié au sexe dans l'admission des étudiants à Berkeley.

```
1  d4 <- open_data(admission)
2  head(d4, 10)
```

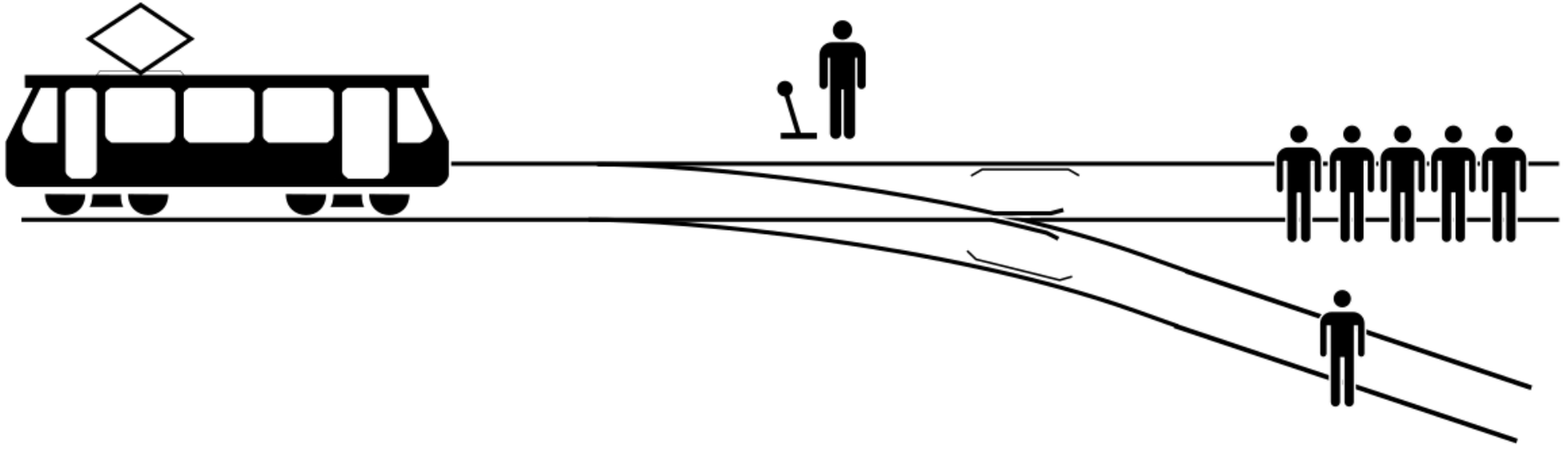
```
   dept applicant.gender admit reject applications
1     A             male   512    313          825
2     A           female    89     19          108
3     B             male   353    207          560
4     B           female    17      8           25
5     C             male   120    205          325
6     C           female   202    391          593
7     D             male   138    279          417
8     D           female   131    244          375
9     E             male    53    138          191
10    E           female    94    299          393
```

Problème n°5

Le dilemme du tramway (trolley problem) est une expérience de pensée qui permet d'étudier les déterminants des jugements de moralité (i.e., qu'est-ce qui fait qu'on juge une action comme morale, ou pas ?).

Sous une forme générale, ce dilemme consiste à poser la question suivante : si une personne peut effectuer un geste qui bénéficiera à un groupe de personnes A, mais, ce faisant, nuira à une personne B (seule); est-il moral pour la personne d'effectuer ce geste ?

Voir ce lien pour plus d'informations.



Problème n°5

Généralement, on fait lire des scénarios aux participants de l'étude, dans lesquels un individu doit prendre une décision dans une situation similaire à celle décrite à la slide précédente. Par exemple, imaginons que Denis ait le choix entre ne rien faire et laisser un train tuer cinq personnes, ou faire dérailler ce train mais tuer une personne. Ensuite, on demande aux participants de juger de la moralité de l'action choisie par Denis, sur une échelle de 1 à 7.

Des études antérieures ont montré que ces jugements de moralité sont grandement influencés par trois mécanismes de raisonnement inconscients :

- Le **principe d'action** : un préjudice causé par une action est jugé moralement moins acceptable qu'un préjudice causé par omission.
- Le **principe d'intention** : un préjudice causé comme étant le moyen vers un but est jugé moralement moins acceptable qu'un préjudice étant un effet secondaire (non désiré) d'un but.
- Le **principe de contact** : un préjudice causé via contact physique est jugé moralement moins acceptable qu'un préjudice causé sans contact physique.

Problème n°5

Ce jeu de données comprend 12 colonnes et 9930 lignes, pour 331 individus. L'outcome qui nous intéresse est `response`, un entier pouvant aller de 1 à 7, qui indique à quel point il est permis (moralement) de réaliser l'action décrite dans le scénario correspondant, en fonction de l'âge (`age`) et genre (`male`) du participant (`id`).

On se demande comment les jugements d'acceptabilité sont influencés par les trois principes décrits slide précédente. Ces trois principes correspondent aux trois variables, `action`, `intention`, et `contact` (dummy-coded).

```
1 d5 <- open_data(morale)
2 head(d5)
```

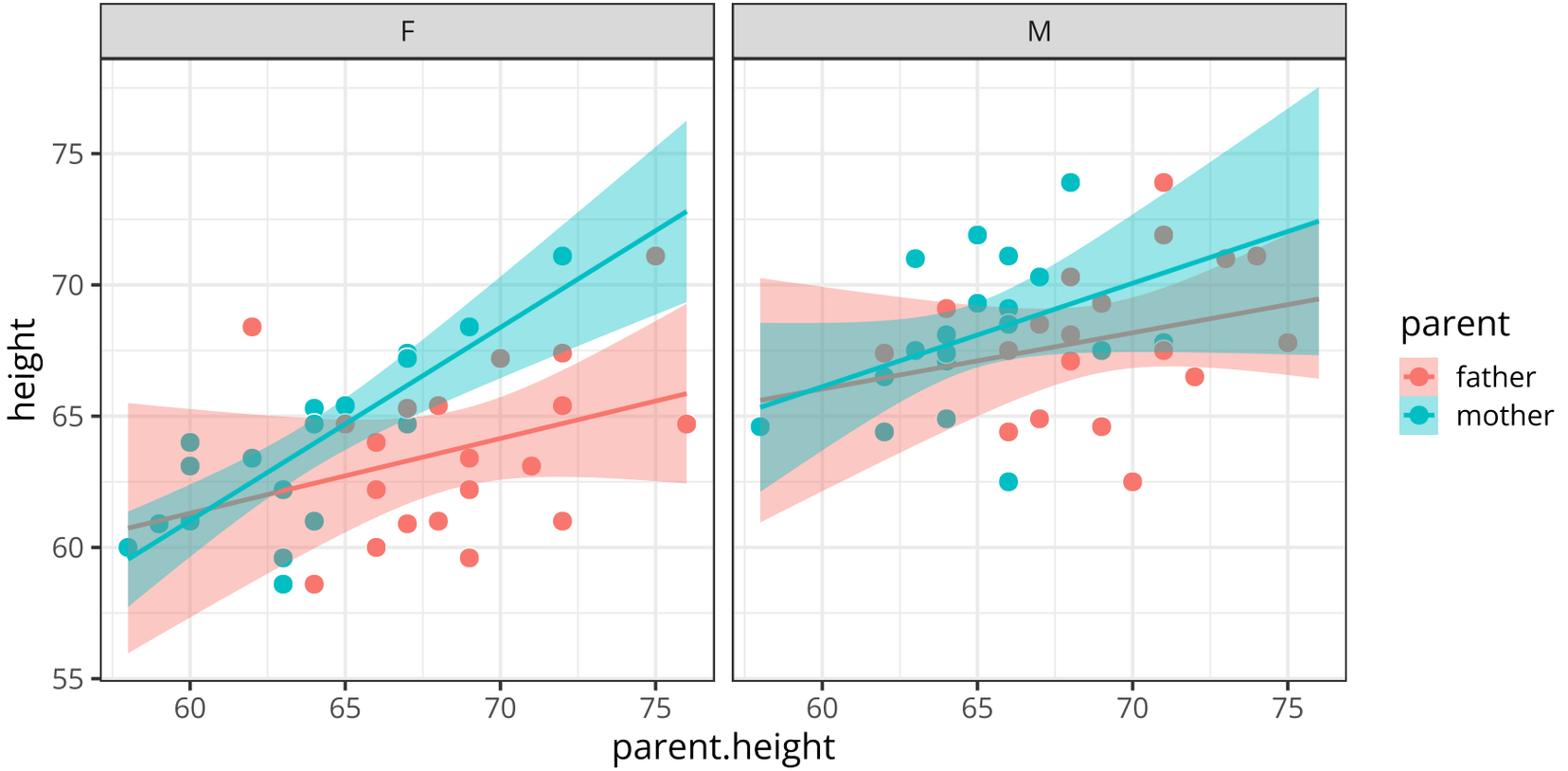
```
  response     id age male action intention contact
1        4 96;434  14    0      0         0       1
2        3 96;434  14    0      0         0       1
3        4 96;434  14    0      0         0       1
4        3 96;434  14    0      0         1       1
5        3 96;434  14    0      0         1       1
6        3 96;434  14    0      0         1       1
```


Propositions de réponses

Réponse possible problème n°1

La taille de la mère a l'air "plus" prédictive de la taille d'un individu, et ce d'autant plus si cet individu est une femme...

```
d1 %>%
    gather(parent, parent.height, 3:4) %>%
    ggplot(aes(x = parent.height, y = height, colour = parent, fill = parent) ) +
    geom_point(pch = 21, size = 4, color = "white", alpha = 1) +
    stat_smooth(method = "lm", fullrange = TRUE) +
    facet_wrap(~ gender)
```



Réponse possible problème n°1

On peut fitter plusieurs modèles avec `brms::brm()`, et les comparer en utilisant le WAIC.

```r
library(brms)

d1$gender <- ifelse(d1$gender == "F", -0.5, 0.5)
d1$mother <- scale(d1$mother) %>% as.numeric
d1$father <- scale(d1$father) %>% as.numeric

p1 <- c(
    prior(normal(70, 10), class = Intercept),
    prior(cauchy(0, 10), class = sigma)
    )

m1 <- brm(
    height ~ 1 + gender,
    prior = p1,
    data = d1
    )

p2 <- c(
    prior(normal(70, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sigma)
    )

m2 <- brm(
    height ~ 1 + gender + mother + father,
```

Réponse possible problème n°1

```r
p3 <- c(
    prior(normal(70, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sigma)
    )

m3 <- brm(
    height ~ 1 + gender + mother + father + gender:mother,
    prior = p3,
    data = d1
    )

p4 <- c(
    prior(normal(70, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sigma)
    )

m4 <- brm(
    height ~ 1 + gender + mother + father + gender:father,
    prior = p4,
    data = d1
    )
```


Réponse possible problème n°1

```r
m1 <- add_criterion(m1, "waic")
m2 <- add_criterion(m2, "waic")
m3 <- add_criterion(m3, "waic")
m4 <- add_criterion(m4, "waic")

model_comparison_table <- loo_compare(m1, m2, m3, m4, criterion = "waic") %>%
  data.frame %>%
  rownames_to_column(var = "model")

weights <- data.frame(weight = model_weights(m1, m2, m3, m4, weights = "waic") ) %>%
  round(digits = 3) %>%
  rownames_to_column(var = "model")

left_join(model_comparison_table, weights, by = "model")
```

```
  model elpd_diff  se_diff  elpd_waic se_elpd_waic   p_waic se_p_waic     waic
1    m3  0.000000 0.000000  -93.24063     5.048439 5.317052  1.361108 186.4813
2    m2 -0.227792 1.424670  -93.46842     5.348427 4.860775  1.418229 186.9368
3    m4 -1.521233 1.646477  -94.76187     5.402762 5.941276  1.744878 189.5237
4    m1 -9.551493 4.784951 -102.79213     4.241509 2.641759  0.647044 205.5843
    se_waic weight
1 10.096877  0.496
2 10.696854  0.395
3 10.805523  0.108
4  8.483018  0.000
```

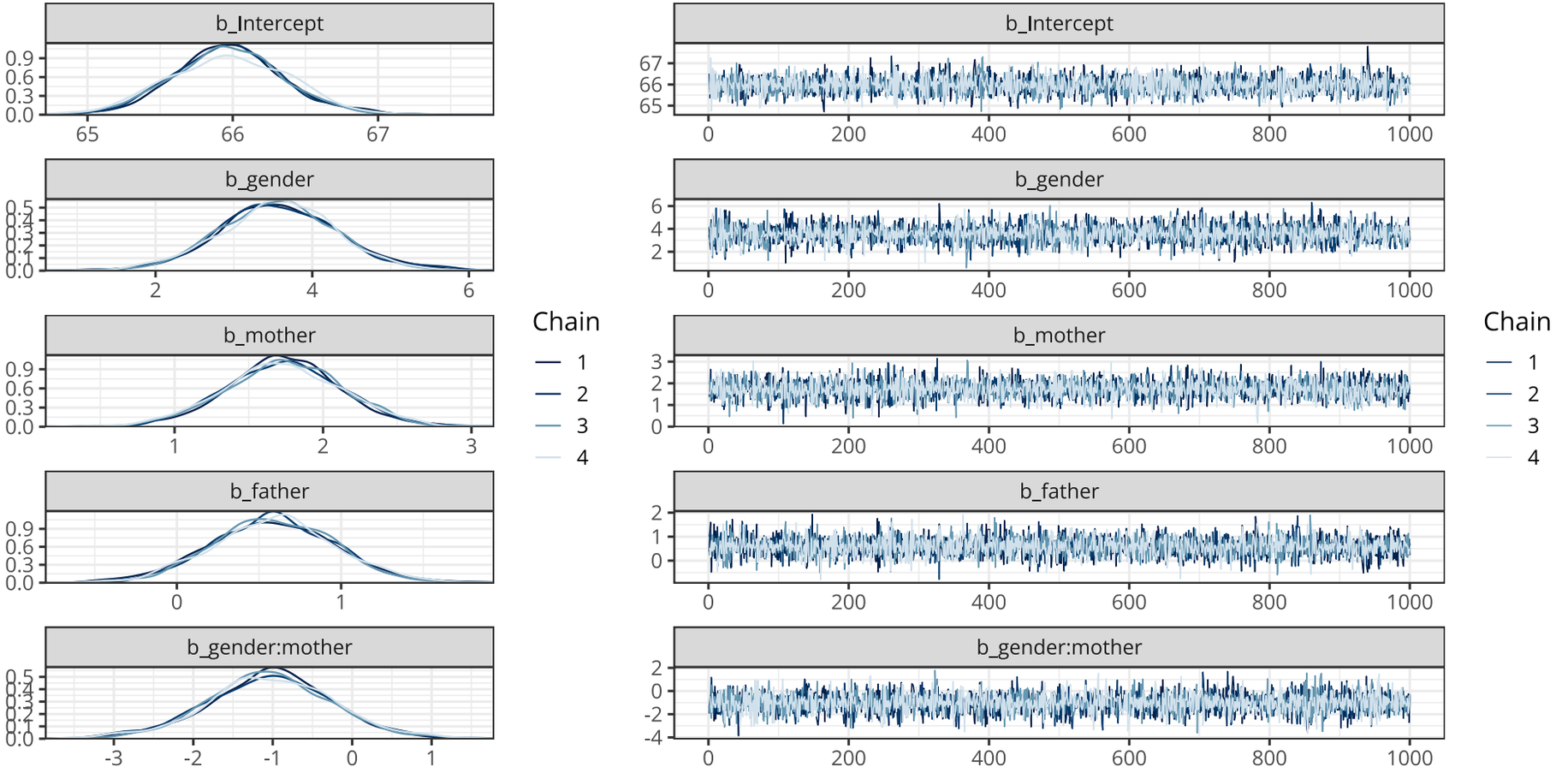
Réponse possible problème n°1

```
1 summary(m3)
```

```
 Family: gaussian
  Links: mu = identity; sigma = identity
Formula: height ~ 1 + gender + mother + father + gender:mother
   Data: d1 (Number of observations: 40)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept       65.97      0.38    65.21    66.74 1.00     5277     2285
gender           3.57      0.75     2.07     5.09 1.00     5819     2990
mother           1.73      0.39     0.95     2.49 1.00     5693     3249
father           0.59      0.37    -0.13     1.31 1.00     5315     2508
gender:mother   -1.04      0.79    -2.68     0.53 1.00     5321     2607

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma     2.35      0.29     1.86     2.98 1.00     4163     2755

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
```

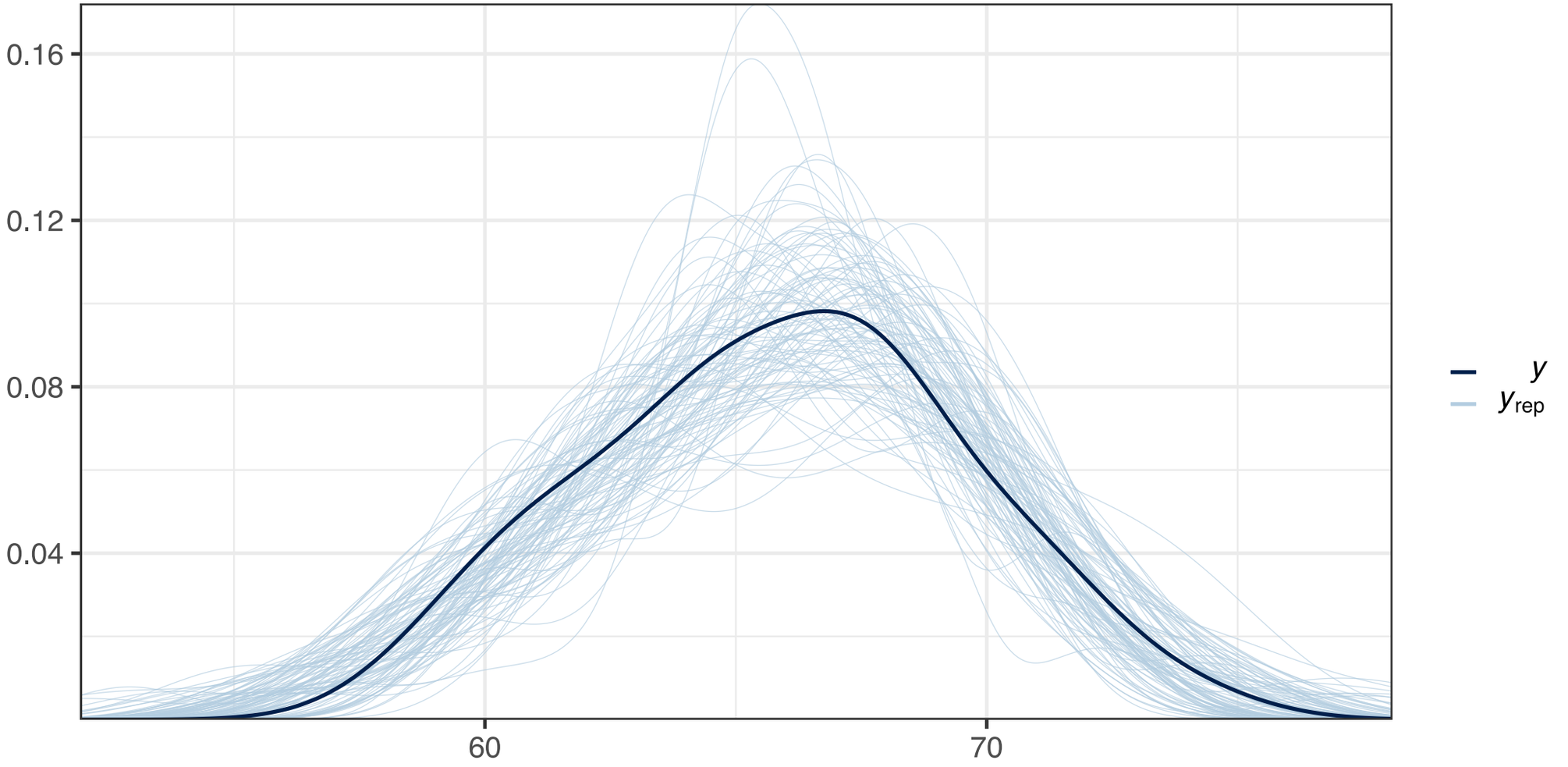

Réponse possible problème n°1

```
m3 %>%
    plot(
        pars = "^b_",
        combo = c("dens_overlay", "trace"), widths = c(1, 1.5),
        theme = theme_bw(base_size = 14, base_family = "Open Sans")
        )
```



Réponse possible problème n°1

```
1  pp_check(m3, nsamples = 1e2) + theme_bw(base_size = 20)
```



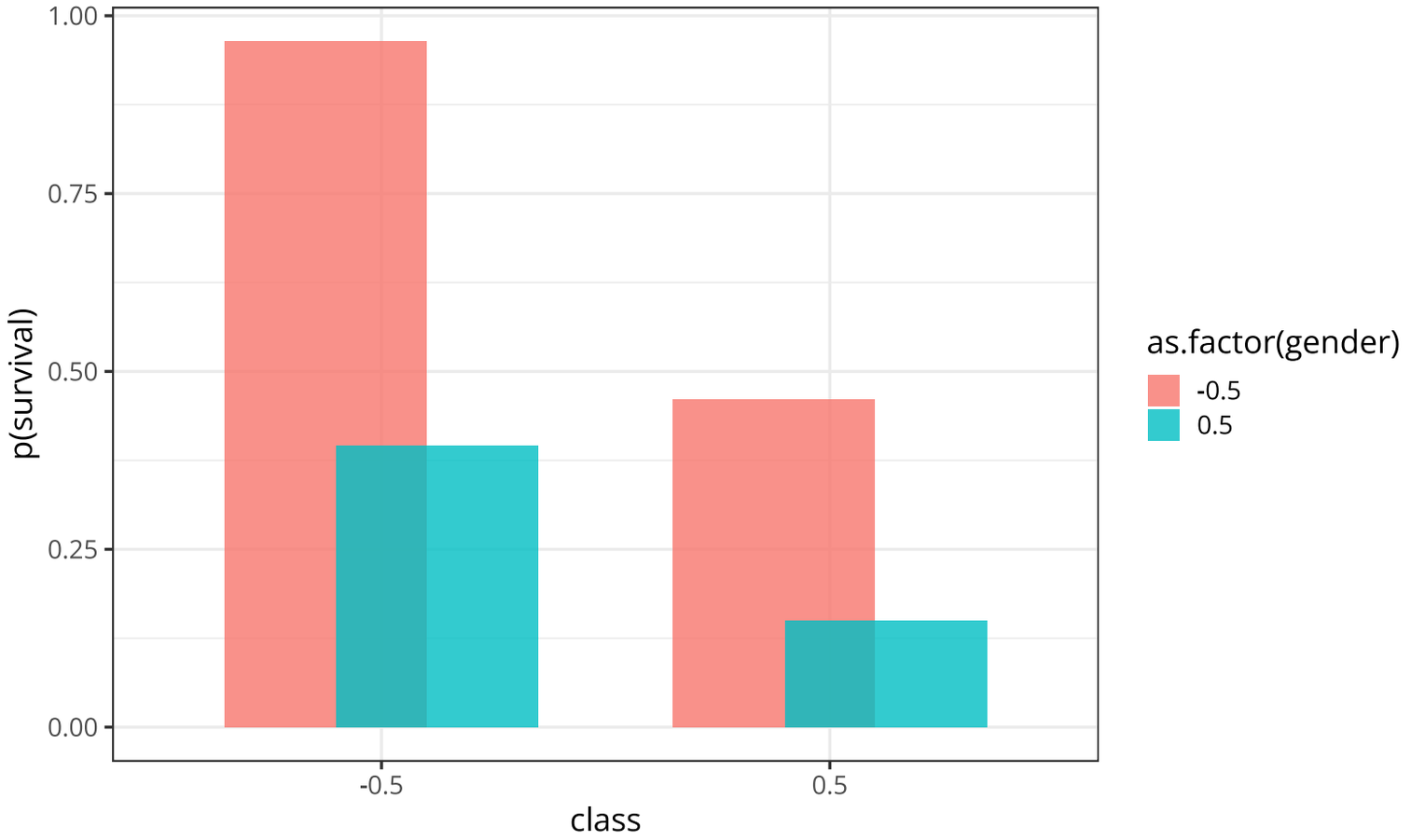
Réponse possible problème n°2

Cette situation revient à essayer de prédire un outcome dichotomique à l'aide de prédicteurs continus et / ou catégoriels. On peut utiliser un modèle de **régression logistique** (cf. Cours n°06).

```r
# centering and standardising predictors

d2 <-
    d2 %>%
    mutate(
        pclass = ifelse(pclass == "lower", -0.5, 0.5),
        gender = ifelse(gender == "female", -0.5, 0.5),
        age = scale(age) %>% as.numeric,
        parch = scale(parch) %>% as.numeric
        )
```

Réponse possible problème n°2

```r
d2 %>%
    group_by(pclass, gender) %>%
    summarise(p = mean(survival) ) %>%
    ggplot(aes(x = as.factor(pclass), y = p, fill = as.factor(gender) ) ) +
    geom_bar(position = position_dodge(0.5), stat = "identity", alpha = 0.8) +
    xlab("class") + ylab("p(survival)")
```



Réponse possible problème n°2

On peut fitter plusieurs modèles avec `brms::brm()`, et les comparer en utilisant le WAIC.

```r
prior0 <- prior(normal(0, 10), class = Intercept)

m0 <- brm(
    survival ~ 1,
    family = binomial(link = "logit"),
    prior = prior0,
    data = d2,
    cores = parallel::detectCores()
    )

prior1 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b)
    )

m1 <- brm(
    # using the dot is equivalent to say "all predictors" (all columns)
    survival ~ .,
    family = binomial(link = "logit"),
    prior = prior1,
    data = d2,
    cores = parallel::detectCores()
    )
```

Réponse possible problème n°2

```r
m2 <- brm(
    survival ~ 1 + pclass + gender + pclass:gender,
    family = binomial(link = "logit"),
    prior = prior1,
    data = d2,
    cores = parallel::detectCores()
    )

m3 <- brm(
    survival ~ 1 + pclass + gender + pclass:gender + age,
    family = binomial(link = "logit"),
    prior = prior1,
    data = d2,
    cores = parallel::detectCores()
    )
```

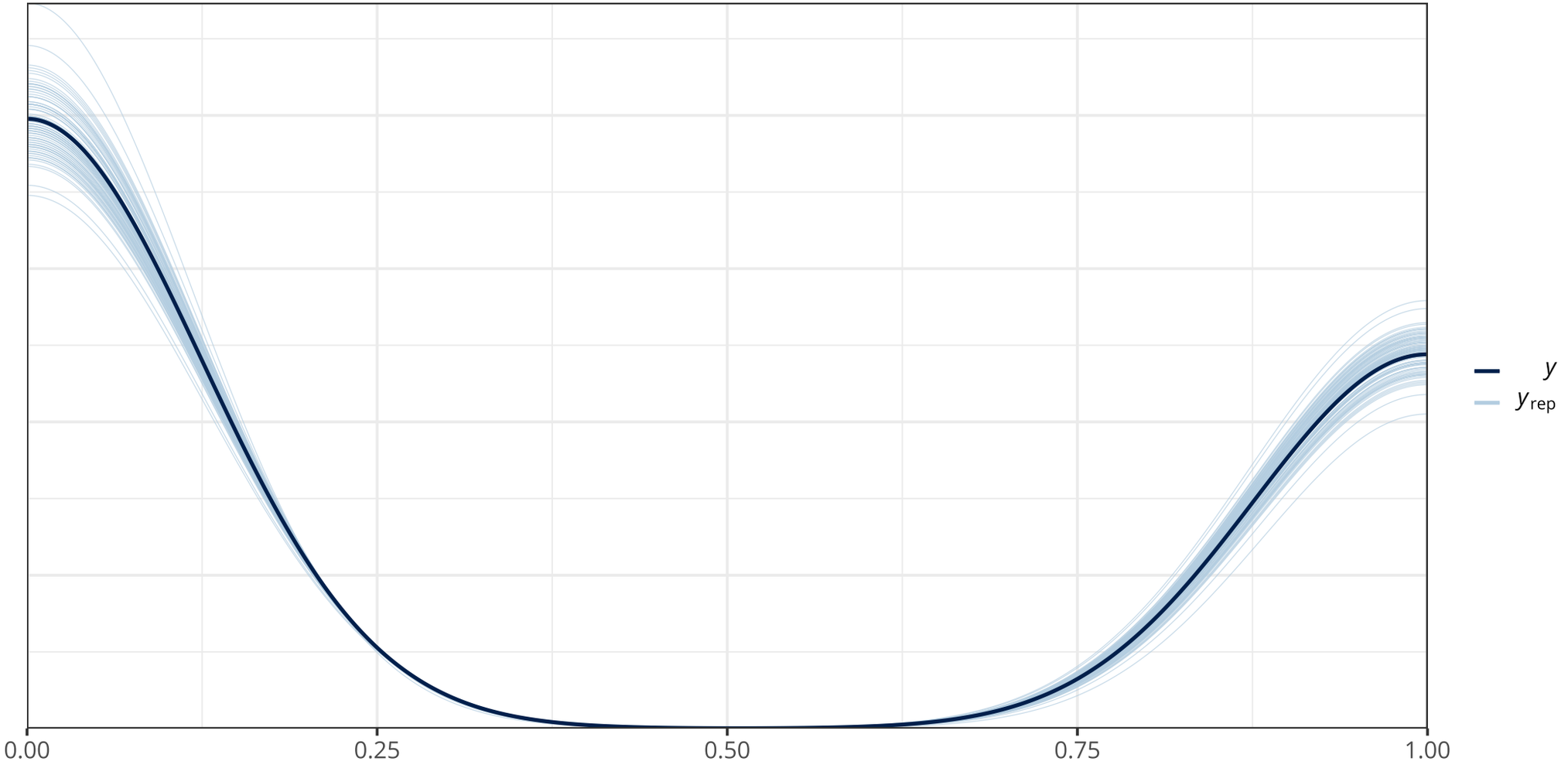

Réponse possible problème n°2

```r
m1 <- add_criterion(m1, "waic")
m2 <- add_criterion(m2, "waic")
m3 <- add_criterion(m3, "waic")

model_comparison_table <- loo_compare(m1, m2, m3, criterion = "waic") %>%
  data.frame %>%
  rownames_to_column(var = "model")

weights <- data.frame(weight = model_weights(m1, m2, m3, weights = "waic") ) %>%
  round(digits = 3) %>%
  rownames_to_column(var = "model")

left_join(model_comparison_table, weights, by = "model")
```

```
  model elpd_diff  se_diff elpd_waic se_elpd_waic   p_waic se_p_waic     waic
1    m3  0.000000 0.000000 -256.3215     13.22400 5.298333 0.7540592 512.6430
2    m1 -4.171201 4.365226 -260.4927     12.96645 5.037803 0.4487408 520.9854
3    m2 -5.883362 3.972515 -262.2049     12.66002 4.069651 0.6408615 524.4097
   se_waic weight
1 26.44801  0.982
2 25.93291  0.015
3 25.32004  0.003
```

Réponse possible problème n°2

```
1  pp_check(m3, nsamples = 1e2)
```



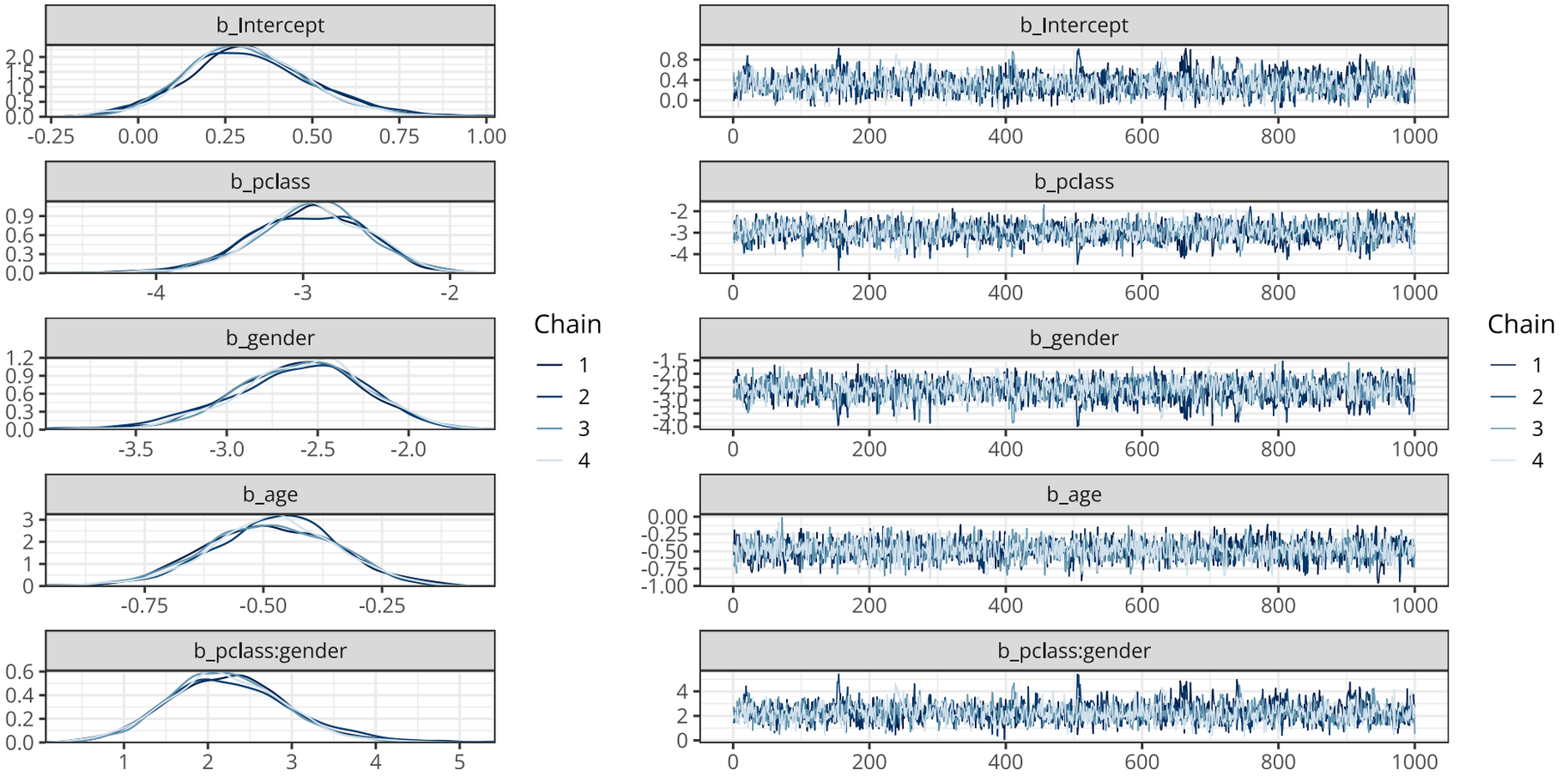
Réponse possible problème n°2

```
1 summary(m3)
```

```
 Family: binomial
  Links: mu = logit
Formula: survival ~ 1 + pclass + gender + pclass:gender + age
   Data: d2 (Number of observations: 539)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept        0.32      0.18    -0.02     0.70 1.00     1720     1235
pclass          -2.95      0.39    -3.76    -2.24 1.00     1795     1455
gender          -2.60      0.36    -3.33    -1.95 1.01     1691     1490
age             -0.48      0.13    -0.74    -0.23 1.00     2835     2270
pclass:gender    2.24      0.71     0.94     3.77 1.01     1676     1392

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```


Réponse possible problème n°2

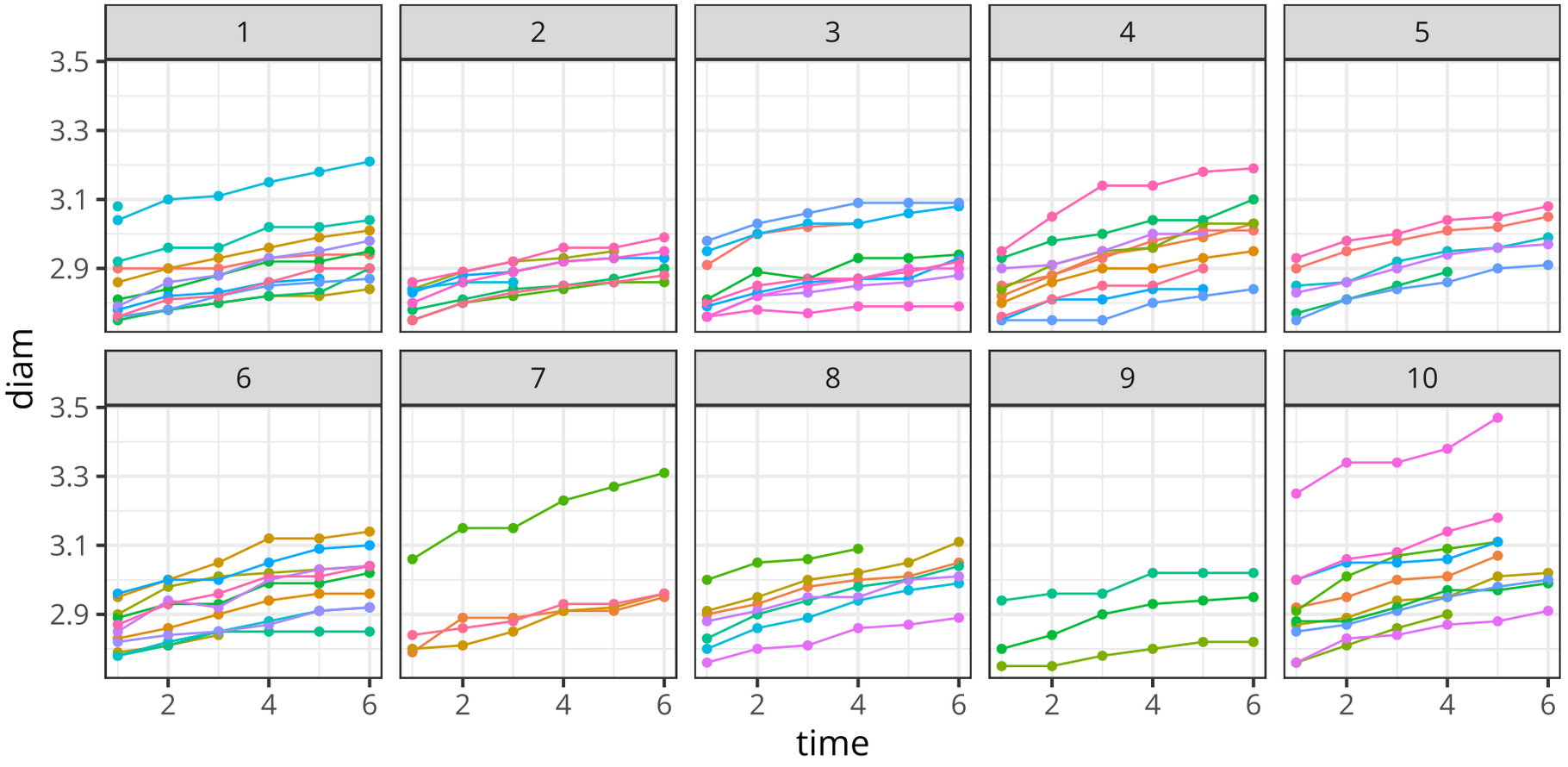
```
m3 %>%
    plot(
        pars = "^b_",
        combo = c("dens_overlay", "trace"), widths = c(1, 1.5),
        theme = theme_bw(base_size = 14, base_family = "Open Sans")
        )
```



Réponse possible problème n°3

Cette situation revient à essayer de prédire une variable continue (le diamètre) à l'aide de prédicteurs continus ordonnés (le temps), en sachant que le diamètre d'une pomme dépend de l'arbre sur lequel cette pomme pousse. On peut utiliser un modèle multi-niveaux (ou modèle mixte, cf. Cours n°08).

```
1  d3 <- d3 %>% filter(diam != 0) # removing null data
```



Réponse possible problème n°3

On peut fitter plusieurs modèles avec `brms::brm()` et les comparer ensuite en utilisant le WAIC.

```r
p1 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(cauchy(0, 10), class = sigma)
    )

m1 <- brm(
    diam ~ 1,
    prior = p1,
    data = d3,
    cores = parallel::detectCores(),
    backend = "cmdstanr"
    )

p2 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sigma)
    )

m2 <- brm(
    diam ~ 1 + time,
    prior = p2,
    data = d3,
    cores = parallel::detectCores(),
    backend = "cmdstanr"
```

Réponse possible problème n°3

```r
p3 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sd),
    prior(cauchy(0, 10), class = sigma)
    )

m3 <- brm(
    diam ~ 1 + time + (1 | tree),
    prior = p3,
    data = d3,
    cores = parallel::detectCores(),
    backend = "cmdstanr"
    )

p4 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sd),
    prior(cauchy(0, 10), class = sigma),
    prior(lkj(2), class = cor)
    )

m4 <- brm(
    diam ~ 1 + time + (1 + time | tree),
```


Réponse possible problème n°3

```r
p5 <- c(
    prior(normal(0, 10), class = Intercept),
    prior(normal(0, 10), class = b),
    prior(cauchy(0, 10), class = sd),
    prior(cauchy(0, 10), class = sigma),
    prior(lkj(2), class = cor)
    )

m5 <- brm(
    diam ~ 1 + time + (1 + time | tree / apple),
    prior = p5,
    data = d3,
    cores = parallel::detectCores(),
    control = list(adapt_delta = 0.99),
    backend = "cmdstanr"
    )
```


Réponse possible problème n°3

```r
m1 <- add_criterion(m1, "waic")
m2 <- add_criterion(m2, "waic")
m3 <- add_criterion(m3, "waic")
m4 <- add_criterion(m4, "waic")
m5 <- add_criterion(m5, "waic")

model_comparison_table <- loo_compare(m1, m2, m3, m4, m5, criterion = "waic") %>%
  data.frame %>%
  rownames_to_column(var = "model")

weights <- data.frame(weight = model_weights(m1, m2, m3, m4, m5, weights = "waic") ) %>%
  round(digits = 3) %>%
  rownames_to_column(var = "model")

left_join(model_comparison_table, weights, by = "model")
```

```
  model elpd_diff  se_diff elpd_waic se_elpd_waic     p_waic se_p_waic
1    m5    0.0000  0.00000 1149.3668     16.86258 114.316560 7.8899015
2    m3 -736.8958 25.44029  412.4709     21.57051  11.856321 1.4417254
3    m4 -737.9083 25.48815  411.4585     21.62494  13.768811 1.6953789
4    m2 -760.1188 27.64534  389.2479     24.26186   4.416807 1.0130718
5    m1 -798.6114 25.82514  350.7553     21.81975   3.159251 0.8253935
       waic  se_waic weight
1 -2298.7335 33.72516      1
2  -824.9419 43.14101      0
3  -822.9170 43.24988      0
4  -778.4959 48.52372      0
5  -701.5106 43.63950      0
```

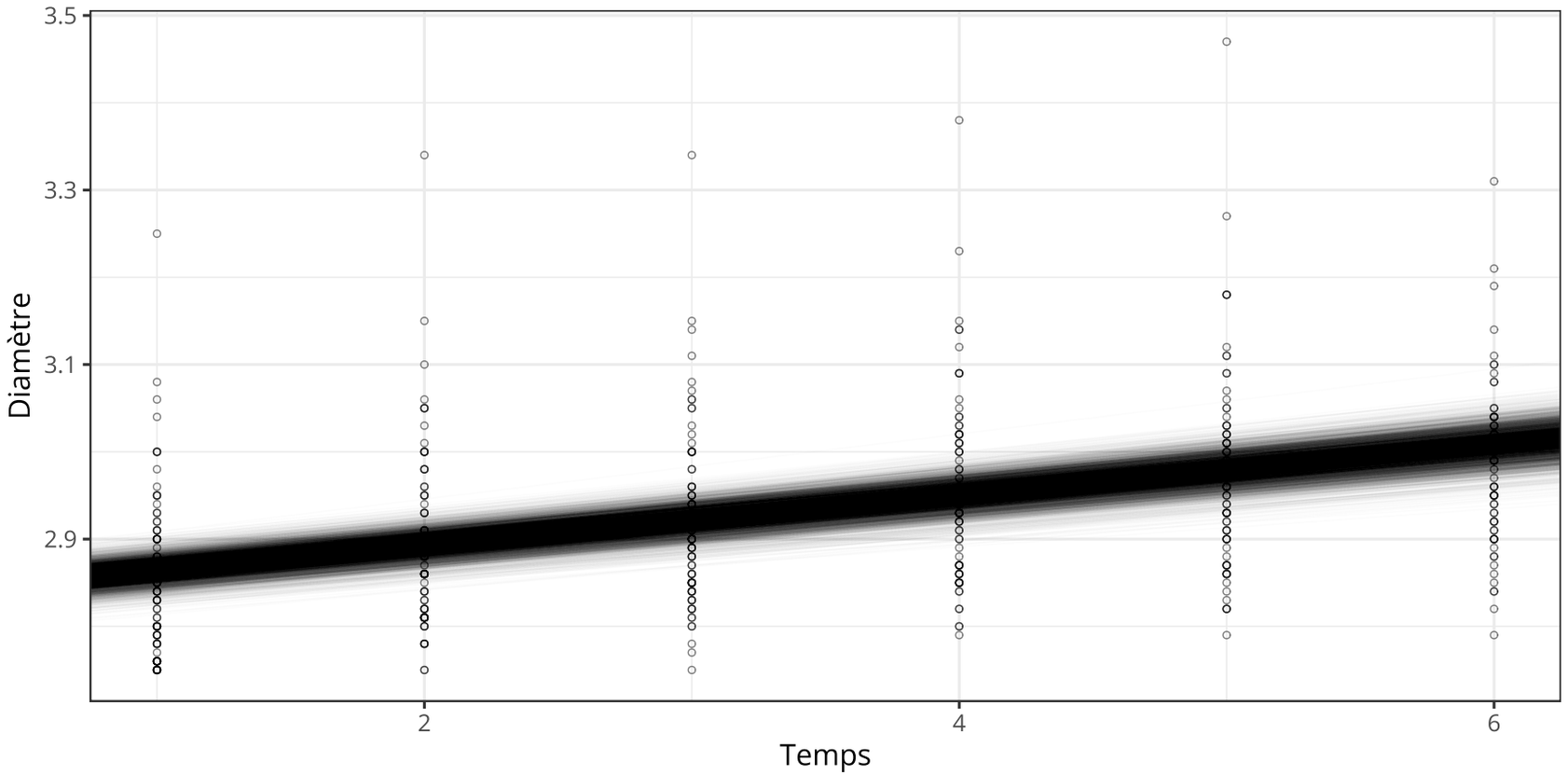

Réponse possible problème n°3

```
1 posterior_summary(m5, pars = c("^b", "sigma") )
```

```
             Estimate    Est.Error       Q2.5      Q97.5
b_Intercept 2.83590625 0.0122725983 2.81070950 2.85976125
b_time      0.02859656 0.0016768687 0.02517868 0.03185300
sigma       0.01622215 0.0006532224 0.01501562 0.01757954
```

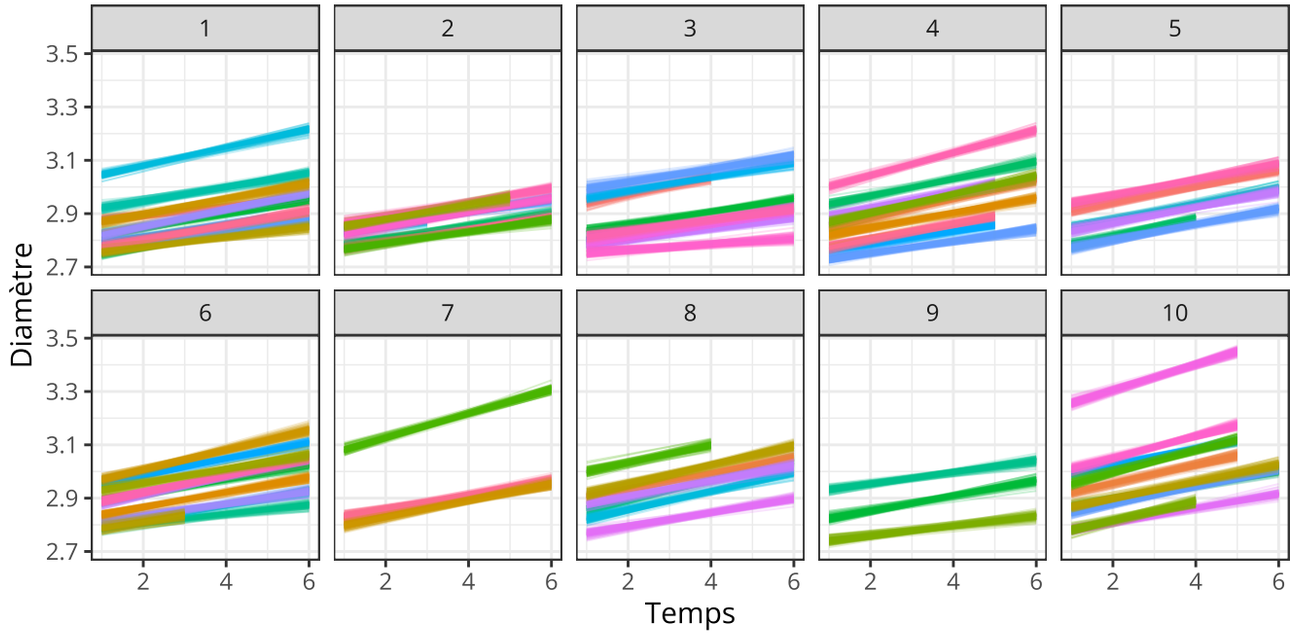

Réponse possible problème n°3

```
post <- posterior_samples(m5, "b") # extracts posterior samples

ggplot(data = d3, aes(x = time, y = diam) ) +
    geom_point(alpha = 0.5, shape = 1) +
    geom_abline(
        data = post, aes(intercept = b_Intercept, slope = b_time),
        alpha = 0.01, size = 0.5) +
    labs(x = "Temps", y = "Diamètre")
```



Réponse possible problème n°3

```
library(tidybayes)
library(modelr)

d3 %>%
    group_by(tree, apple) %>%
    data_grid(time = seq_range(time, n = 1e2) ) %>%
    add_fitted_samples(m5, n = 1e2) %>%
    ggplot(aes(x = time, y = diam, colour = factor(apple) ) ) +
    geom_line(
        aes(y = estimate, group = paste(apple, .iteration) ),
        alpha = 0.2, show.legend = FALSE) +
    facet_wrap(~tree, ncol = 5) +
    labs(x = "Temps", y = "Diamètre")
```



Réponse possible problème n°3

Quelques notes sur la proposition de réponse concernant ce problème. Les modèles proposés ici pourraient être améliorés sur plusieurs aspects... est-ce que vous avez des idées ?

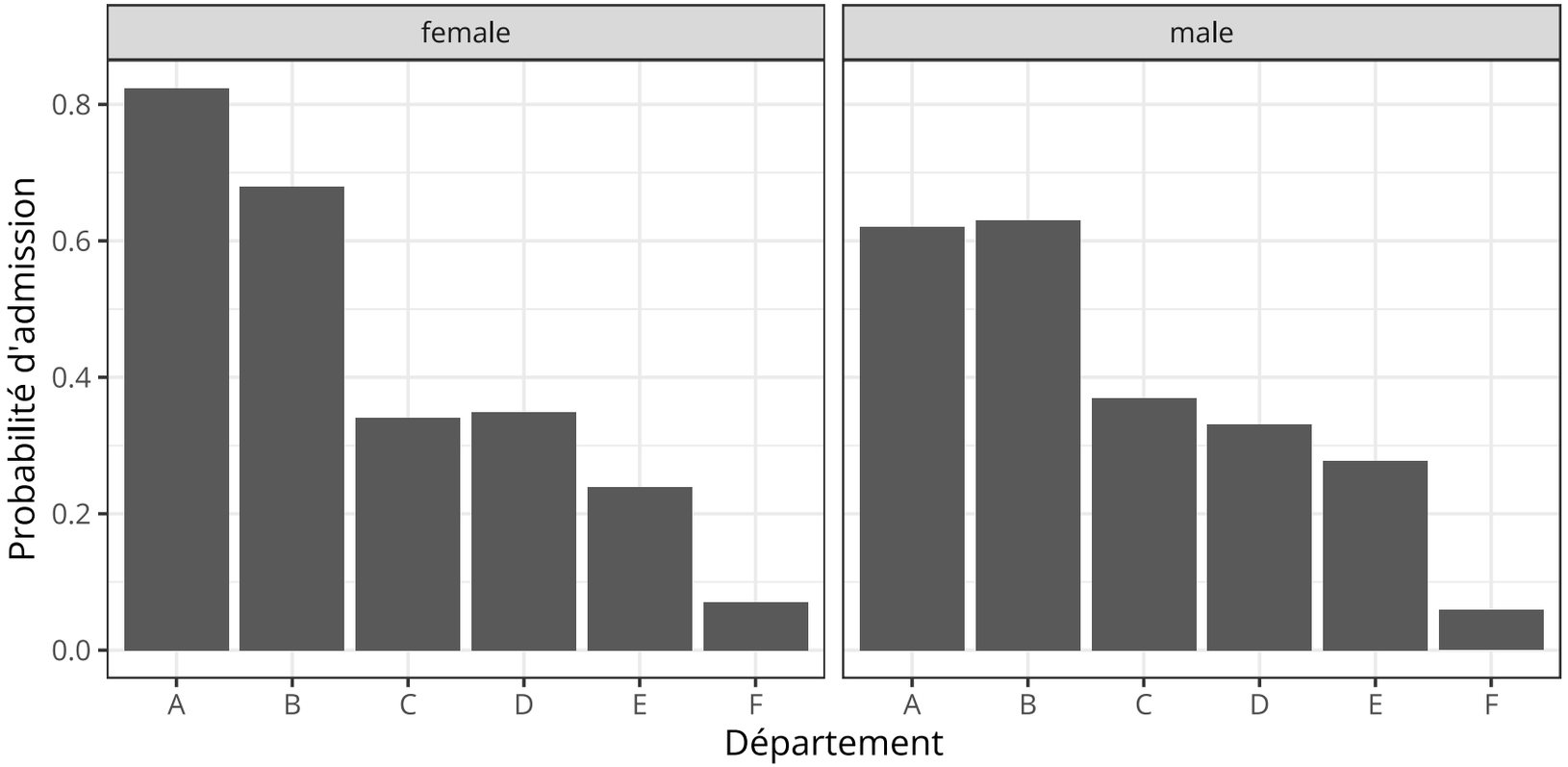
Premièrement, notre prédicteur (temps) est mesuré en utilisant une échelle discrète (i.e., le nombre de semaines). Il s'agit d'un prédicteur ordinal (i.e., un prédicteur avec différentes catégories entre elles) et un meilleur modèle pour ce genre de données est présenté dans Bürkner & Charpentier (2020).

Deuxièmement, on pourrait affiner le modèle d'observation postulé pour le phénomène mesuré. Plus précisément, nous avons des informations sur la nature de la variable mesurée (le diamètre). En l'occurrence, on sait par exemple que le diamètre d'une pomme ne peut pas être négatif. On pourrait donc remplacer la fonction de vraisemblance Gaussienne par une fonction de vraisemblance Log-Normale ou Ex-Gaussienne (par exemple).

Réponse possible problème n°4

Cette situation revient à essayer de prédire un outcome dichotomique (admit, reject) à l'aide de prédicteurs continus et/ou catégoriels.

```
d4 %>%
    ggplot(aes(x = dept, y = admit / applications) ) +
    geom_bar(stat = "identity") +
    facet_wrap(~ applicant.gender) +
    labs(x = "Département", y = "Probabilité d'admission")
```



Réponse possible problème n°4

On peut fitter plusieurs modèles avec `brms::brm()` et les comparer ensuite en utilisant le WAIC.

```r
# centering gender predictor
d4$gender <- ifelse(d4$applicant.gender == "female", -0.5, 0.5)

# creating an index for department
d4$dept_id <- as.integer(as.factor(d4$dept) )

p1 <- c(
    prior(normal(0, 10), class = "Intercept"),
    prior(cauchy(0, 2), class = "sd")
    )

m1 <- brm(
    admit | trials(applications) ~ 1 + (1 | dept_id),
    data = d4, family = binomial,
    prior = p1,
    warmup = 1000, iter = 5000,
    control = list(adapt_delta = 0.99, max_treedepth = 12),
    backend = "cmdstanr"
    )
```


Réponse possible problème n°4

```r
p2 <- c(
    prior(normal(0, 10), class = "Intercept"),
    prior(normal(0, 1), class = "b"),
    prior(cauchy(0, 2), class = "sd")
    )

m2 <- brm(
    admit | trials(applications) ~ 1 + gender + (1 | dept_id),
    data = d4, family = binomial,
    prior = p2,
    warmup = 1000, iter = 5000,
    control = list(adapt_delta = 0.99, max_treedepth = 12),
    backend = "cmdstanr"
    )

p3 <- c(
    prior(normal(0, 10), class = "Intercept"),
    prior(normal(0, 1), class = "b"),
    prior(cauchy(0, 2), class = "sd"),
    prior(lkj(2), class = "cor")
    )

m3 <- brm(
    admit | trials(applications) ~ 1 + gender + (1 + gender | dept_id),
    data = d4, family = binomial,
```


Réponse possible problème n°4

```r
m1 <- add_criterion(m1, "waic")
m2 <- add_criterion(m2, "waic")
m3 <- add_criterion(m3, "waic")

model_comparison_table <- loo_compare(m1, m2, m3, criterion = "waic") %>%
  data.frame %>%
  rownames_to_column(var = "model")

weights <- data.frame(weight = model_weights(m1, m2, m3, weights = "waic") ) %>%
  round(digits = 3) %>%
  rownames_to_column(var = "model")

left_join(model_comparison_table, weights, by = "model")
```

```
  model elpd_diff  se_diff elpd_waic se_elpd_waic   p_waic se_p_waic      waic
1    m3  0.000000 0.000000 -45.40642     2.245900 6.707272  1.246378  90.81283
2    m1 -7.196737 7.711346 -52.60315     8.983006 6.545705  2.279898 105.20630
3    m2 -8.786955 6.895374 -54.19337     8.265119 9.328847  2.984583 108.38674
    se_waic weight
1  4.491801  0.999
2 17.966012  0.001
3 16.530239  0.000
```

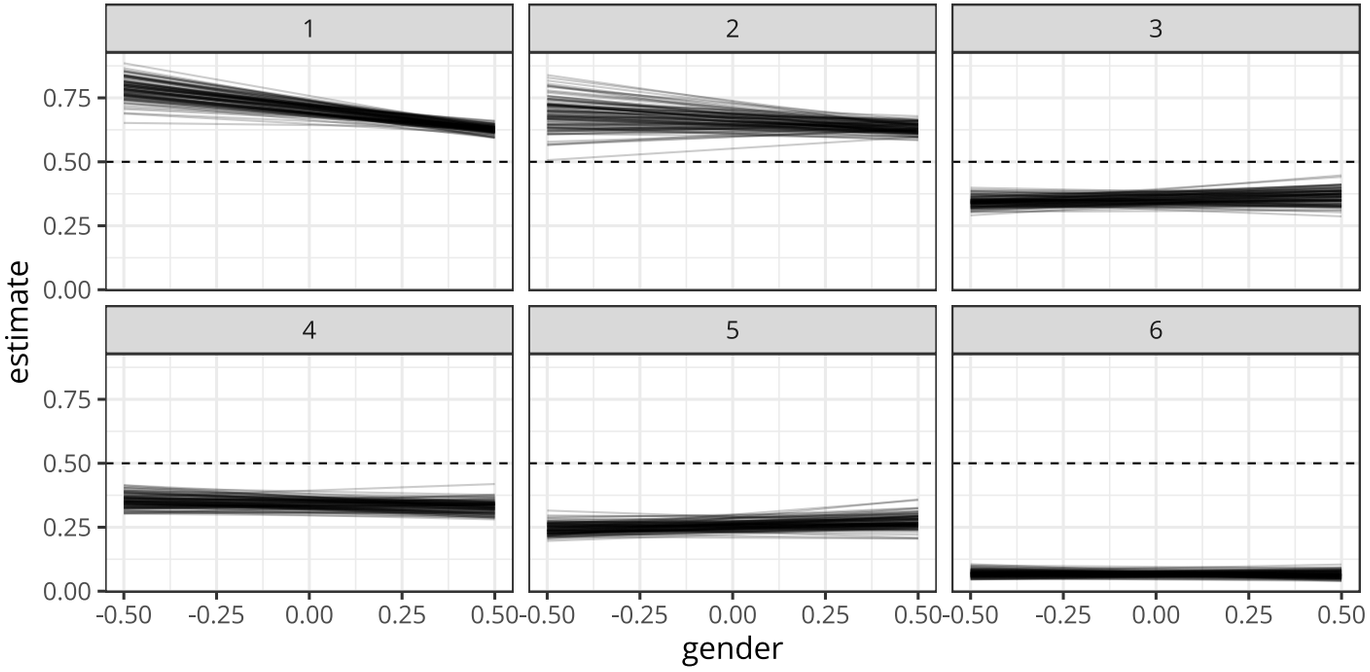
Réponse possible problème n°4

```
1 summary(m3)
```

```
 Family: binomial
  Links: mu = logit
Formula: admit | trials(applications) ~ 1 + gender + (1 + gender | dept_id)
   Data: d4 (Number of observations: 12)
  Draws: 4 chains, each with iter = 5000; warmup = 1000; thin = 1;
         total post-warmup draws = 16000

Group-Level Effects:
~dept_id (Number of levels: 6)
                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
sd(Intercept)           1.57      0.61     0.83     3.12 1.00     4733
sd(gender)              0.51      0.27     0.17     1.18 1.00     5071
cor(Intercept,gender)  -0.28      0.36    -0.86     0.49 1.00     9476
                    Tail_ESS
sd(Intercept)           7053
sd(gender)              6753
cor(Intercept,gender)   9787

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept    -0.56      0.68    -1.95     0.81 1.00     4358     6398
```

Réponse possible problème n°4

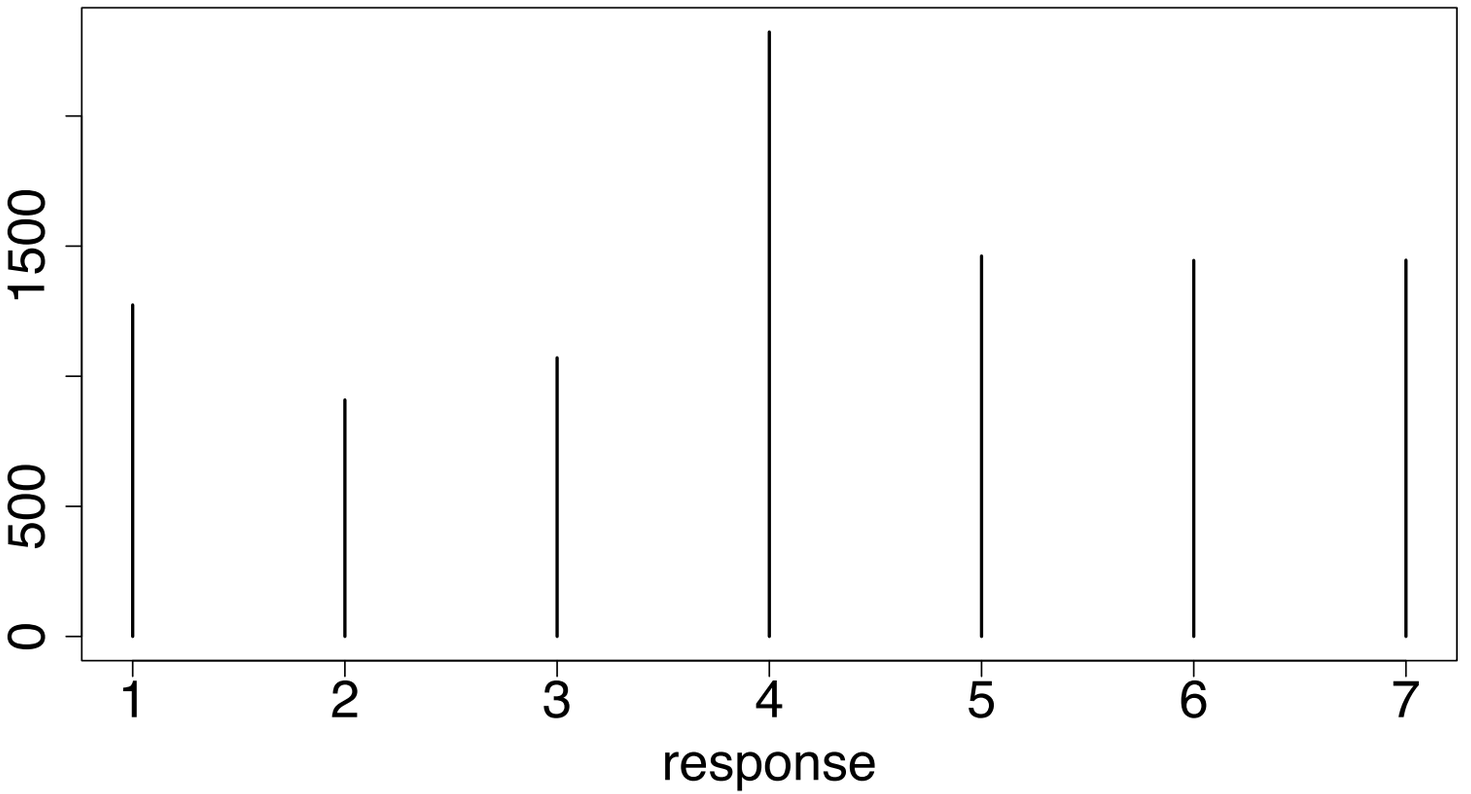
```r
library(tidybayes)
library(modelr)

d4 %>%
    group_by(dept_id, applications) %>%
    data_grid(gender = seq_range(gender, n = 1e2) ) %>%
    add_fitted_samples(m3, newdata = ., n = 100, scale = "linear") %>%
    mutate(estimate = plogis(estimate) ) %>%
    ggplot(aes(x = gender, y = estimate, group = .iteration) ) +
    geom_hline(yintercept = 0.5, lty = 2) +
    geom_line(aes(y = estimate, group = .iteration), size = 0.5, alpha = 0.2) +
    facet_wrap(~dept_id, nrow = 2)
```



Réponse possible problème n°5

On essaye de prédire un jugement exprimé sous forme d'entier entre 1 et 7. Autrement dit, la variable qu'on essaye de prédire est une variable catégorielle, dont les catégories sont ordonnées de 1 à 7...

```
1  d5$response %>% table %>%
2    plot(xlab = "response", ylab = "", cex.axis = 2, cex.lab = 2)
```



Réponse possible problème n°5

Ce type de données peut se modéliser en utilisant le modèle de régression logistique ordinale, brièvement discuté à la fin du Cours n°09. Ci-dessous un exemple en utilisant `brms`, et les priors par défaut (NB : ces modèles peuvent être un peu longs à fitter).

```
moral1 <- brm(
    response ~ 1,
    data = d5,
    family = cumulative("logit"),
    cores = parallel::detectCores(),
    control = list(adapt_delta = 0.99),
    backend = "cmdstanr"
    )

moral2 <- brm(
    response ~ 1 + action + intention + contact,
    data = d5,
    family = cumulative("logit"),
    cores = parallel::detectCores(),
    control = list(adapt_delta = 0.99),
    backend = "cmdstanr"
    )
```


Réponse possible problème n°5

Toutes les pentes sont négatives... ce qui signifie que chaque facteur réduit la réponse moyenne (i.e., le jugement de moralité). Ces pentes représentent des changements dans les log-odds cumulatifs.

```
1 brms::waic(moral1, moral2)
```

```
Output of model 'moral1':

Computed from 4000 by 9930 log-likelihood matrix

          Estimate   SE
elpd_waic -18927.2 28.8
p_waic         5.9  0.0
waic       37854.3 57.6

Output of model 'moral2':

Computed from 4000 by 9930 log-likelihood matrix

          Estimate   SE
elpd_waic -18544.7 38.1
p_waic         8.8  0.0
waic       37089.5 76.2

Model comparisons:
       elpd_diff se_diff
moral2    0.0       0.0
```


Réponse possible problème n°5

```
1 summary(moral2, prob = 0.95)
```

```
 Family: cumulative
  Links: mu = logit; disc = identity
Formula: response ~ 1 + action + intention + contact
   Data: d5 (Number of observations: 9930)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Population-Level Effects:
            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]    -2.84      0.05    -2.93    -2.75 1.00     2926     2434
Intercept[2]    -2.16      0.04    -2.24    -2.07 1.00     3374     3016
Intercept[3]    -1.57      0.04    -1.65    -1.50 1.00     3629     3188
Intercept[4]    -0.55      0.04    -0.62    -0.48 1.00     3785     2888
Intercept[5]     0.12      0.04     0.05     0.19 1.00     4031     3264
Intercept[6]     1.02      0.04     0.95     1.10 1.00     4569     3323
action          -0.71      0.04    -0.79    -0.63 1.00     4060     3005
intention       -0.72      0.04    -0.79    -0.65 1.00     4484     2690
contact         -0.96      0.05    -1.06    -0.86 1.00     3746     2993

Family Specific Parameters:
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

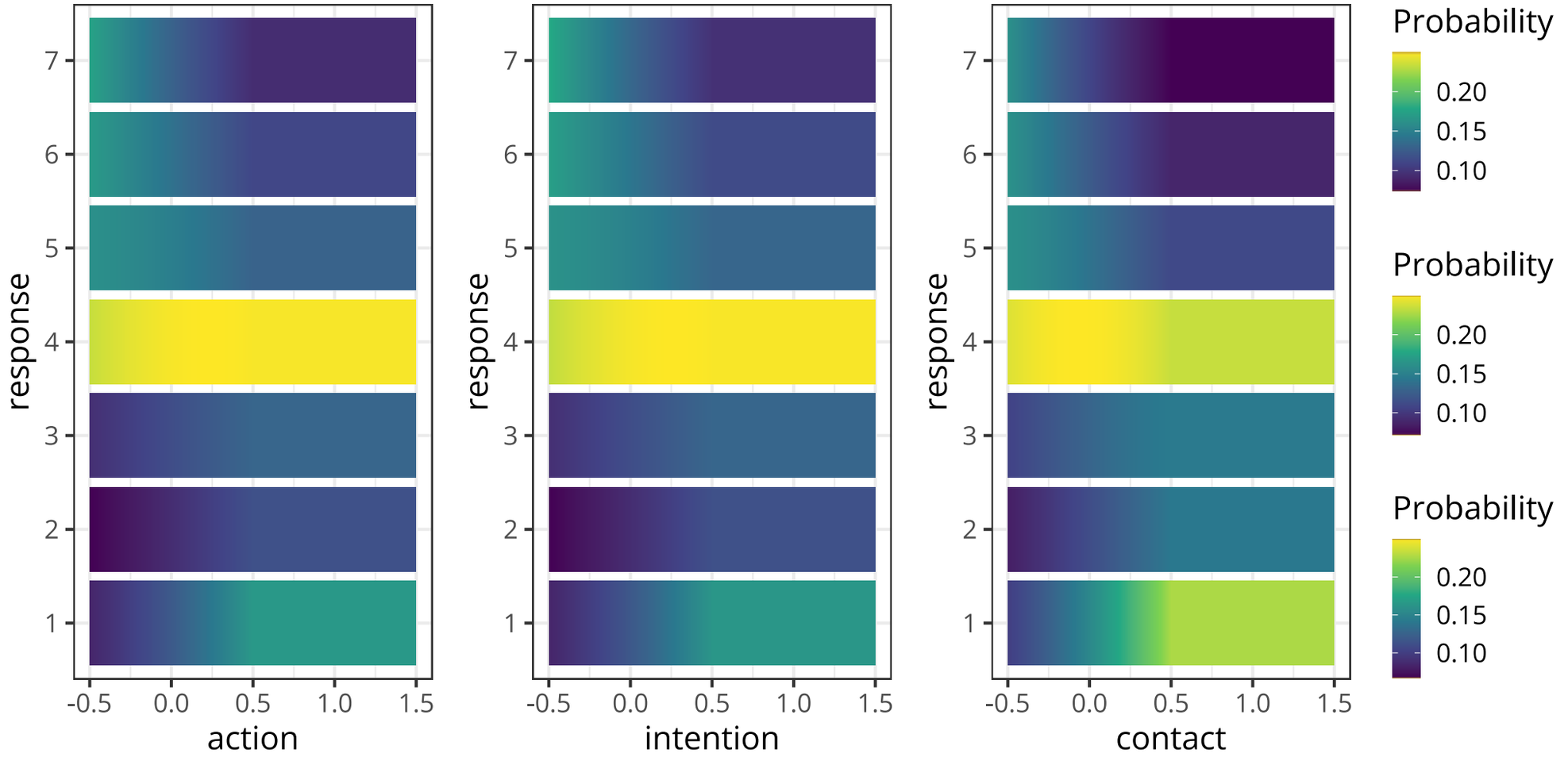

Réponse possible problème n°5

On peut représenter les prédictions du modèle en utilisant la fonction `brms::marginal_effects()`.

```
marg1 <- marginal_effects(moral2, "action", ordinal = TRUE)
p1 <- plot(marg1, theme = theme_bw(base_size = 20, base_family = "Open Sans"), plot = FALSE)[[1]]

marg2 <- marginal_effects(moral2, "intention", ordinal = TRUE)
p2 <- plot(marg2, theme = theme_bw(base_size = 20, base_family = "Open Sans"), plot = FALSE)[[1]]

marg3 <- marginal_effects(moral2, "contact", ordinal = TRUE)
p3 <- plot(marg3, theme = theme_bw(base_size = 20, base_family = "Open Sans"), plot = FALSE)[[1]]
```


Réponse possible problème n°5

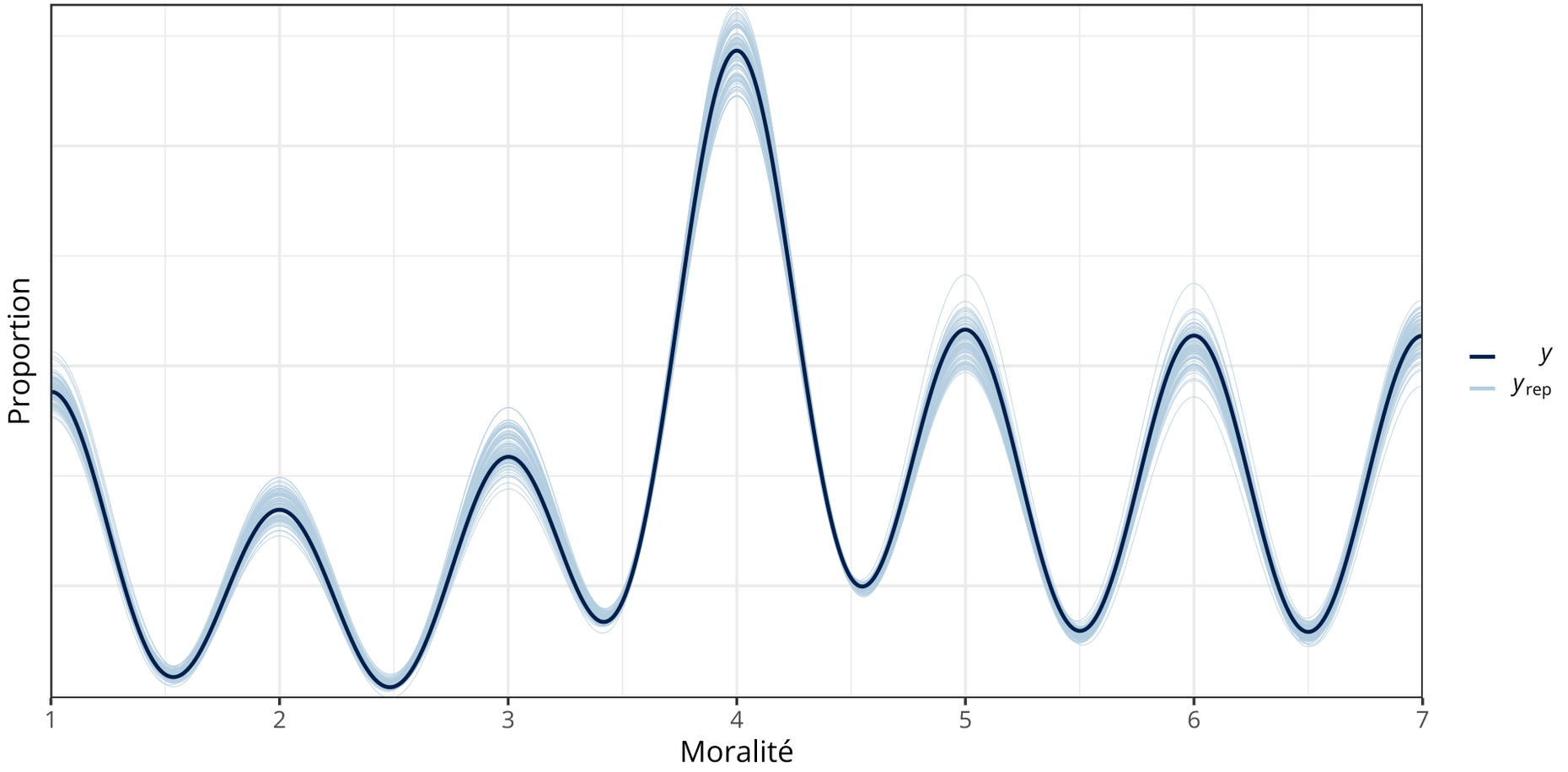
```
1  library(patchwork)
2  p1 + p2 + p3 + plot_layout(guides = "collect") & theme(legend.position = "right")
```



Réponse possible problème n°5

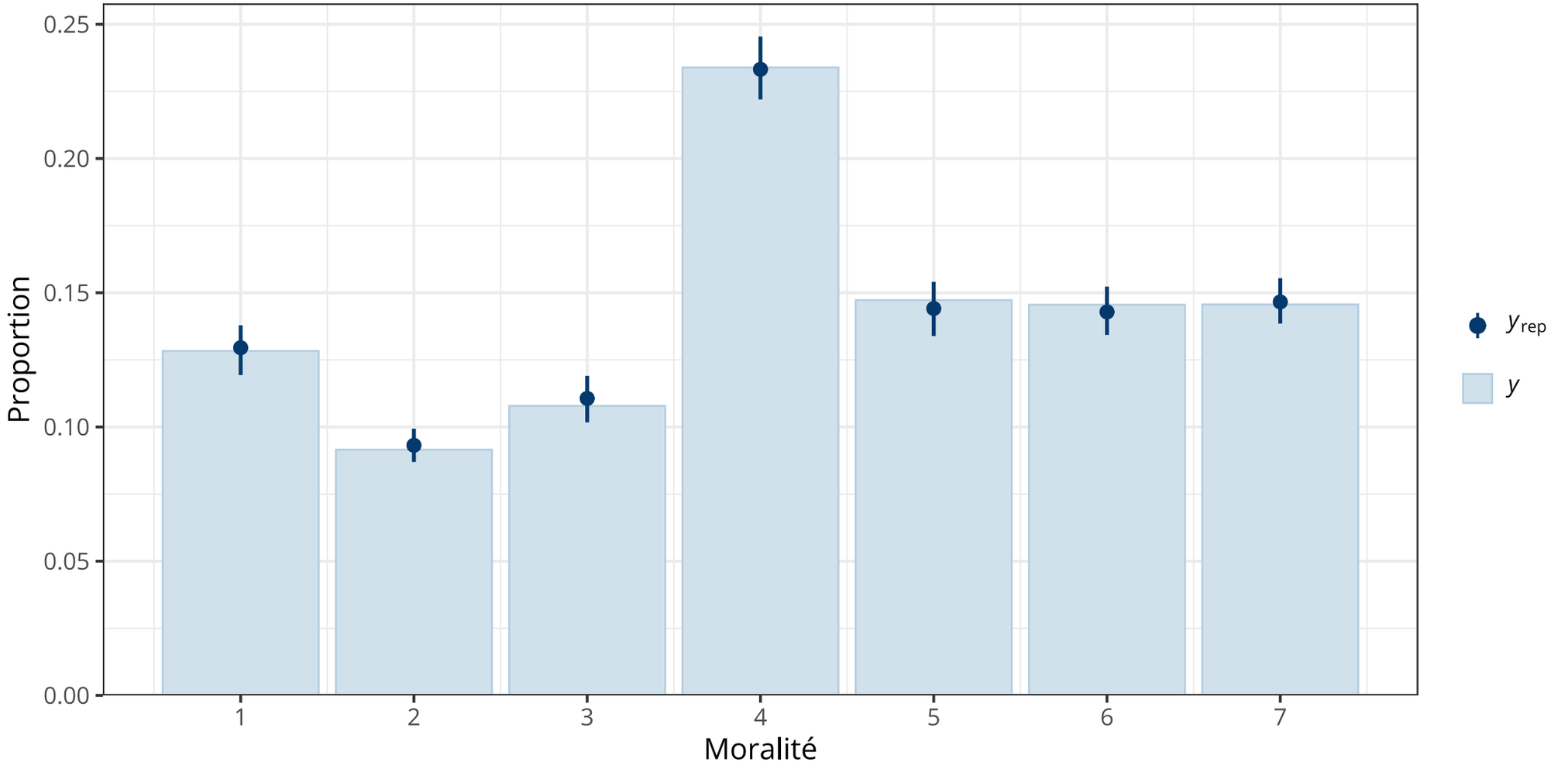
Pour plus d'informations sur la régression logistique ordinale, voir Liddell & Kruschke (2018), Bürkner & Vuorre (2019), ou le chapitre 11 de McElreath (2020).

```
1  pp_check(moral2, nsamples = 1e2) +
2    labs(x = "Moralité", y = "Proportion")
```



Réponse possible problème n°5

```
1  pp_check(moral2, nsamples = 1e2, type = "bars", prob = 0.95, freq = FALSE) +
2    scale_x_continuous(breaks = 1:7) +
3    labs(x = "Moralité", y = "Proportion")
```



Références

Bürkner, P.-C., & Charpentier, E. (2020). Modelling monotonic effects of ordinal predictors in Bayesian regression models. *British Journal of Mathematical and Statistical Psychology*, *73*(3), 420–451. https://doi.org/10.1111/bmsp.12195

Bürkner, P.-C., & Vuorre, M. (2019). Ordinal Regression Models in Psychology: A Tutorial. *Advances in Methods and Practices in Psychological Science*, *2*(1), 77–101. https://doi.org/10.1177/2515245918823199

Liddell, T. M., & Kruschke, J. K. (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, *79*, 328–348. https://doi.org/10.1016/j.jesp.2018.08.009

McElreath, R. (2020). *Statistical rethinking: A bayesian course with examples in r and stan* (2nd ed.). Taylor; Francis, CRC Press.