

Introduction à la modélisation statistique bayésienne

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).



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 `bzrms`.



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 `bzrms`.

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.



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 `bzrms`.

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 ?

```
library(tidyverse)  
  
d1 <- read.csv("data/parents.csv")  
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 ?

```
d2 <- read.csv("data/titanic.csv")
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) ?

```
d3 <- read.csv("data/apples.csv")
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 package `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.

```
library(rethinking)
d4 <- get(data(UCBadmit) )
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

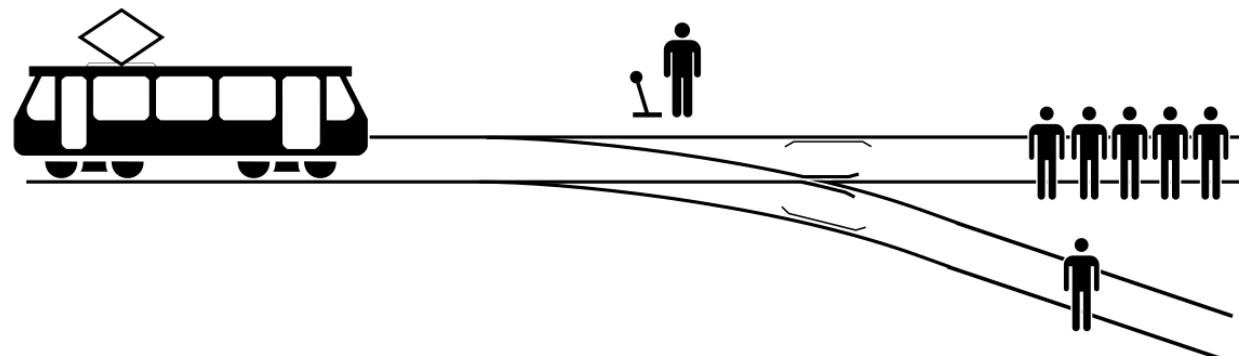


Problème n°5

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).

```
d5 <- read.csv("data/morale.csv")
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

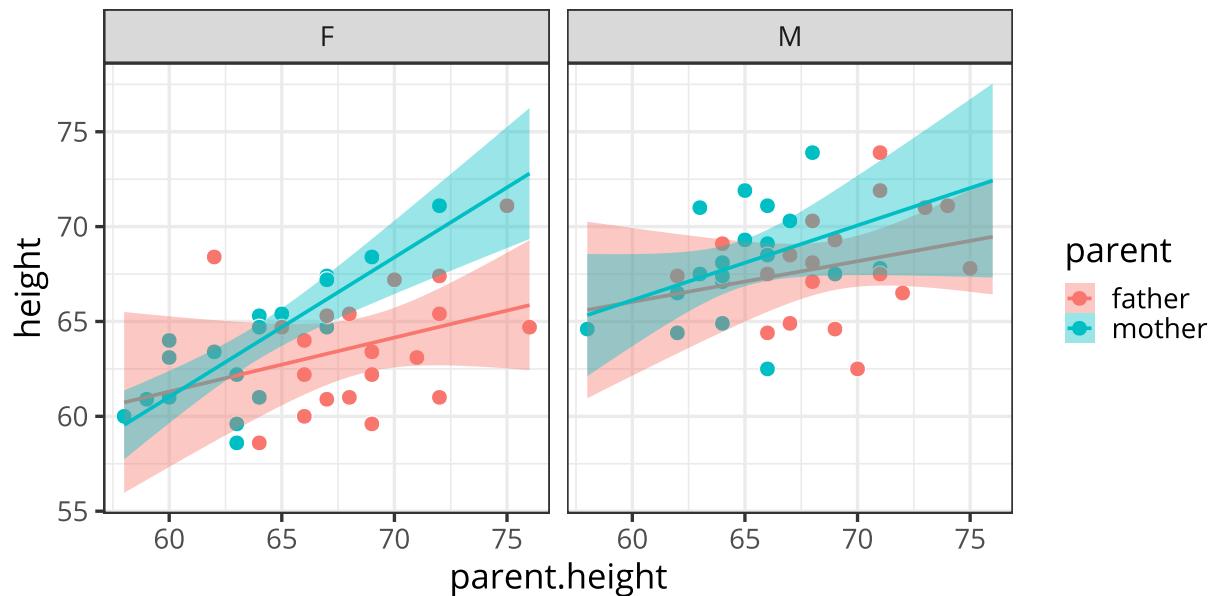
Propositions de réponses



Réponse 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 problème n°1

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

```
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,
  prior = p2,
  data = d1
)
```

12



Réponse problème n°1

```
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 problème n°1

```
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	-93.40868	5.007490	5.386806	1.361854	186.8174
2	m2	-0.111643	1.465079	-93.52032	5.381510	4.916108	1.414796	187.0406
3	m4	-1.301467	1.578262	-94.71015	5.299438	5.828583	1.640442	189.4203
4	m1	-9.395523	4.745187	-102.80420	4.257850	2.673129	0.663660	205.6084
	se_waic	weight						
1	10.01498	0.462						
2	10.76302	0.413						
3	10.59888	0.126						
4	8.51570	0.000						



Réponse problème n°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	65.98	0.39	65.19	66.74	1.00	6048	3191
gender	3.59	0.79	2.05	5.14	1.00	5581	2805
mother	1.71	0.41	0.90	2.50	1.00	5262	3146
father	0.59	0.39	-0.18	1.35	1.00	5742	2646
gender:mother	-1.07	0.77	-2.60	0.42	1.00	5583	2961

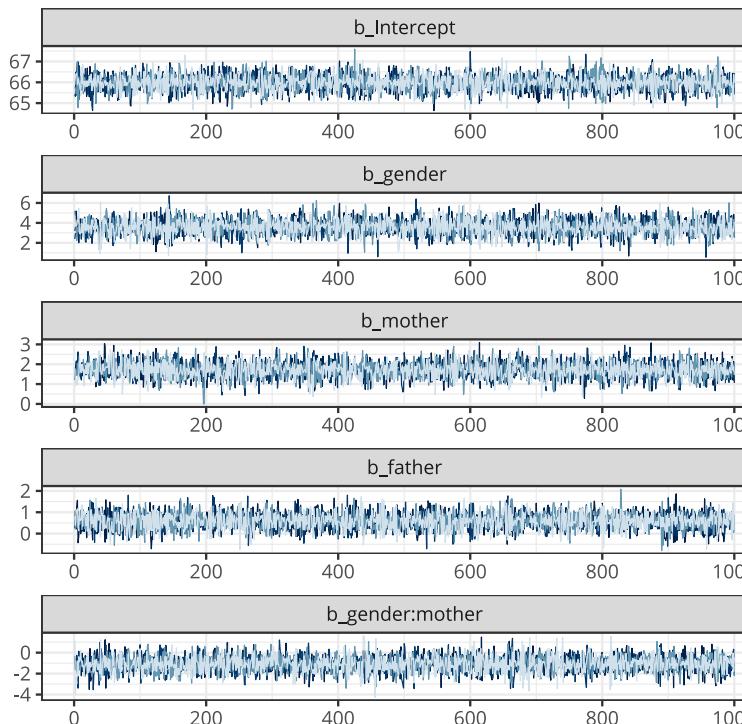
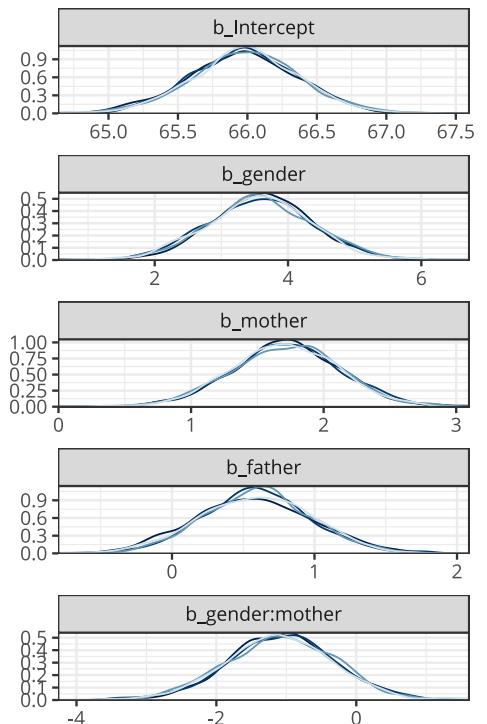
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	2.36	0.30	1.87	3.01	1.00	4032	3068

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).

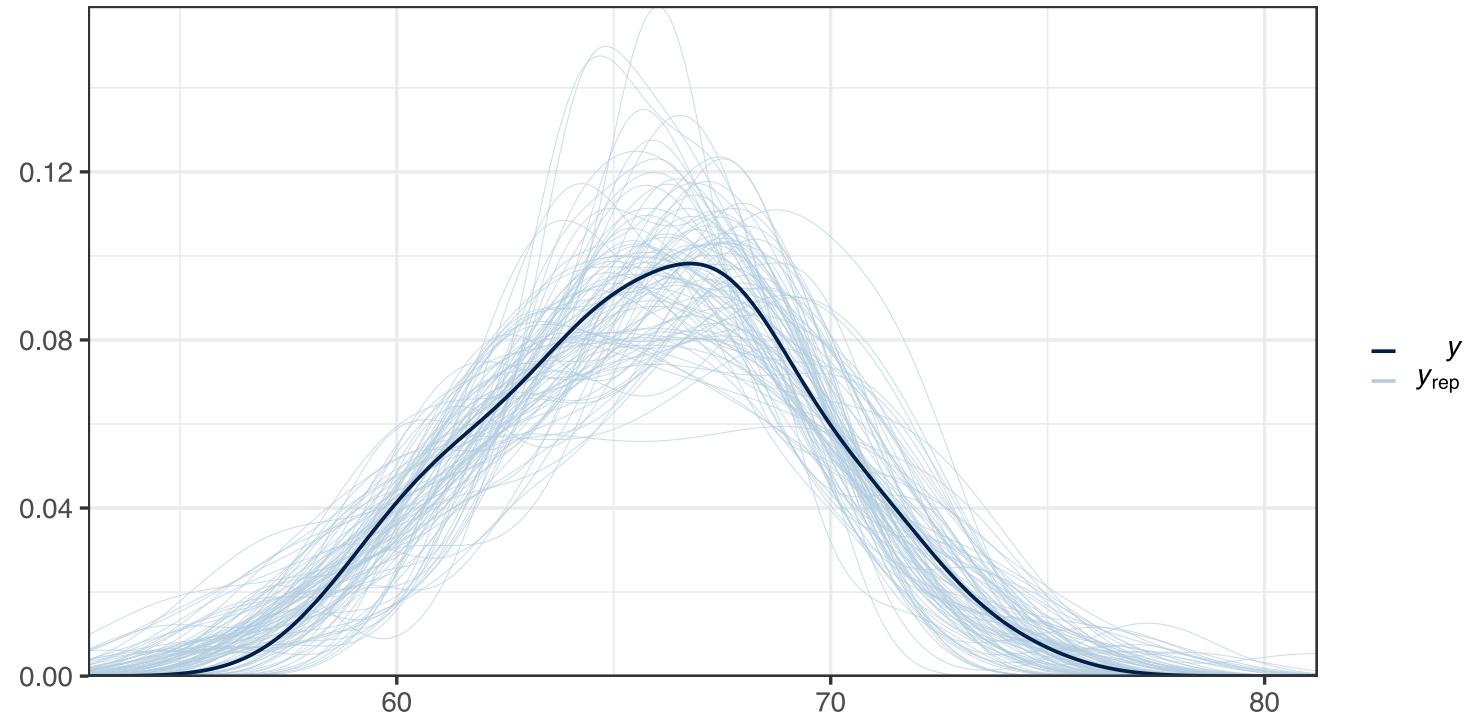
Réponse problème n°1

```
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 problème n°1

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



Réponse 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).

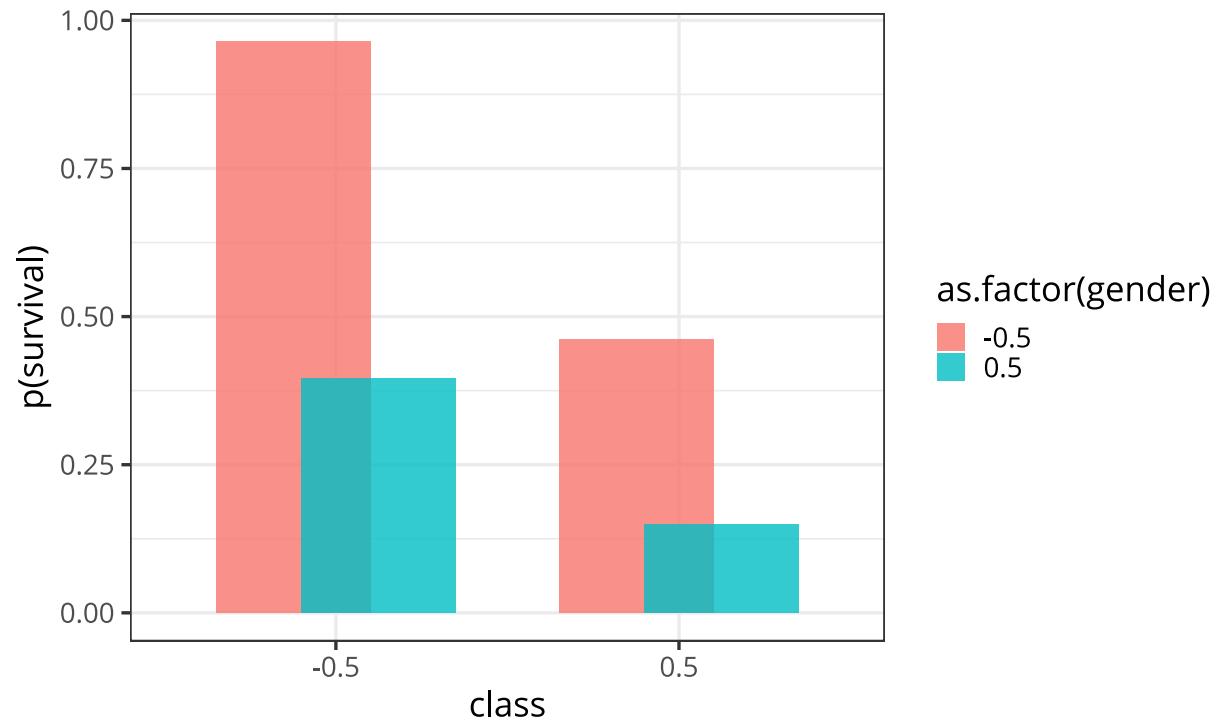
```
# centering and standardising predictors

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 problème n°2

```
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 problème n°2

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

```
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 problème n°2

```
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 problème n°2

```
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")
```

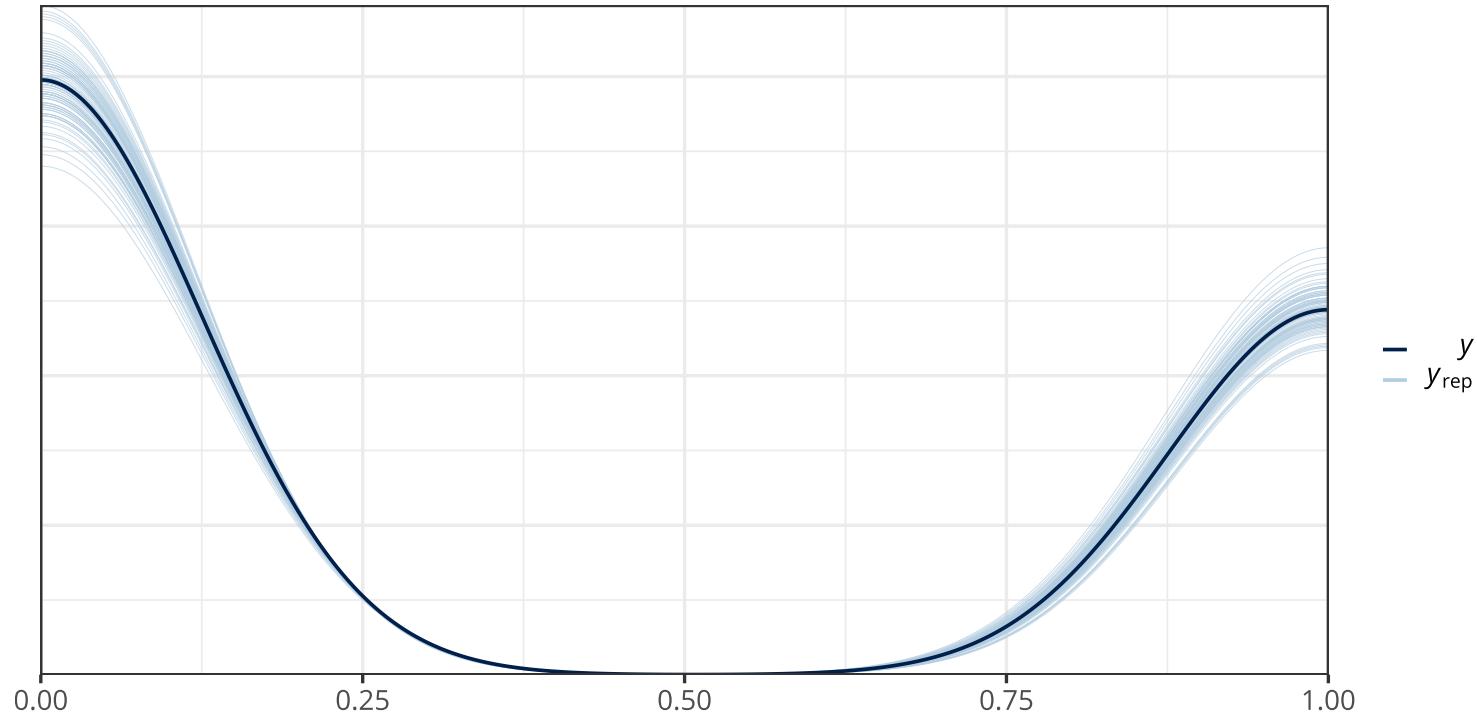
	model	elpd_diff	se_diff	elpd_waic	se_elpd_waic	p_waic	se_p_waic	waic
1	m3	0.000000	0.000000	-256.2070	13.27447	5.196429	0.7689453	512.4141
2	m1	-4.170900	4.438229	-260.3779	12.91831	4.931806	0.4429285	520.7559
3	m2	-6.275095	3.969899	-262.4821	12.72384	4.347686	0.7496830	524.9643

	se_waic	weight
1	26.54894	0.983
2	25.83661	0.015
3	25.44767	0.002



Réponse problème n°2

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



Réponse problème n°2

```
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:

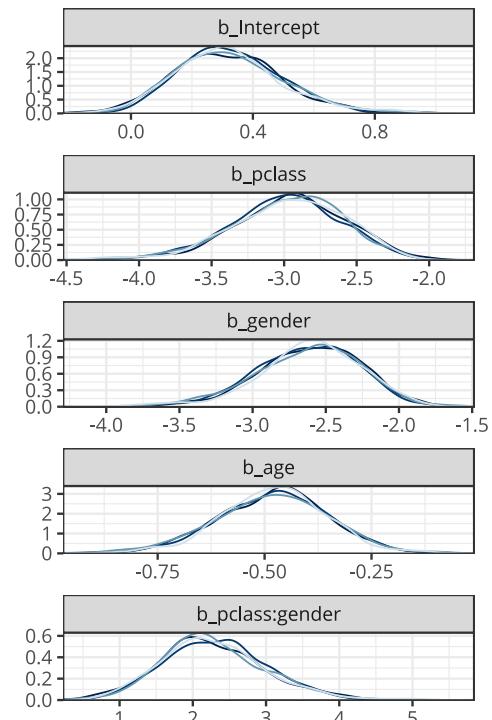
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.32	0.18	0.02	0.71	1.00	1474	1537
pclass	-2.98	0.39	-3.83	-2.29	1.00	1624	1722
gender	-2.62	0.36	-3.40	-1.99	1.00	1462	1481
age	-0.48	0.13	-0.75	-0.24	1.00	3022	2565
pclass:gender	2.29	0.71	1.01	3.78	1.00	1406	1412

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).



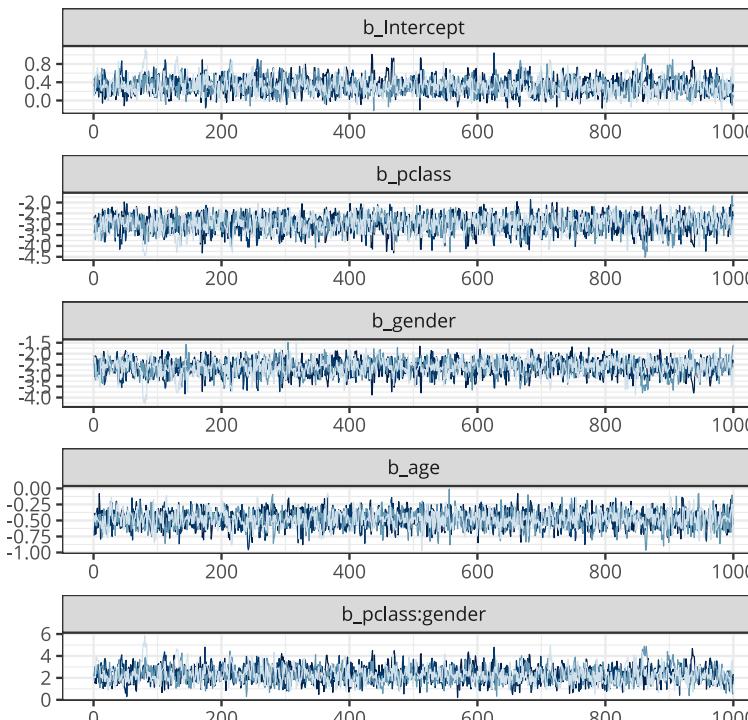
Réponse problème n°2

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



Chain

- 1
- 2
- 3
- 4



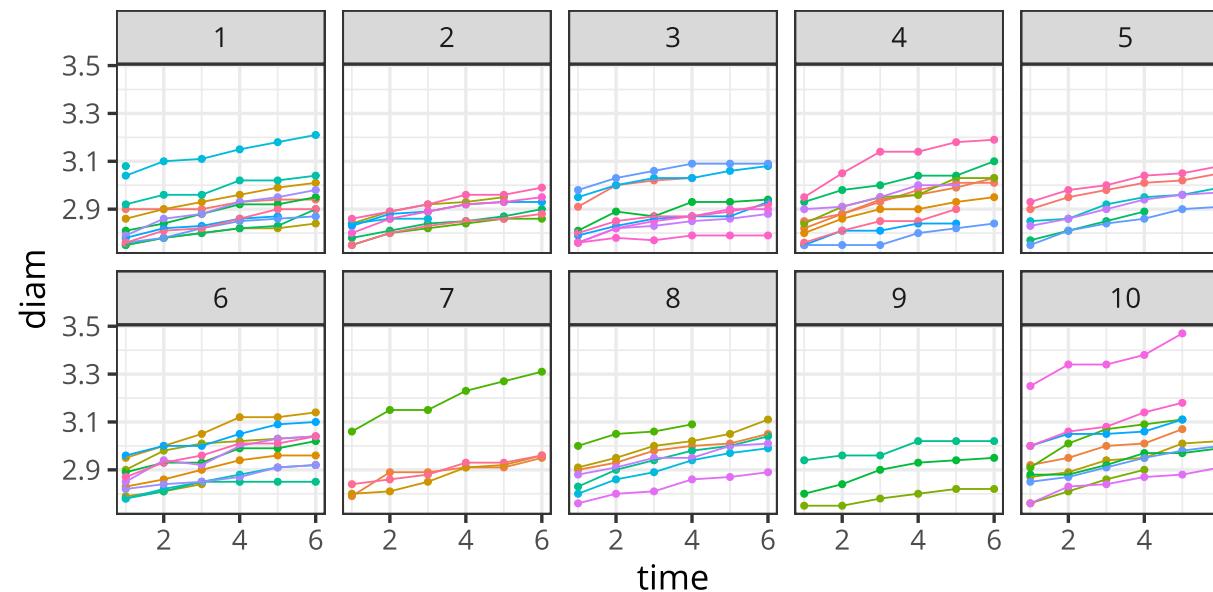
Chain

- 1
- 2
- 3
- 4

Réponse 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).

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



Solution problème n°3

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

```
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"
)
```



Solution problème n°3

```
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),
  prior = p4,
  data = d3,
  cores = parallel::detectCores(),
  control = list(adapt_delta = 0.99),
  backend = "cmdstanr"
)
```



Solution problème n°3

```
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"
)
```



Solution problème n°3

```
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	1150.2310	16.78730	113.396211	7.816722
2	m3	-737.7672	25.38164	412.4638	21.57817	11.918446	1.438018
3	m4	-739.1176	25.45435	411.1134	21.66294	14.107996	1.753265
4	m2	-760.9864	27.57277	389.2446	24.26230	4.416760	1.023097
5	m1	-799.3179	25.71044	350.9131	21.76144	2.980521	0.791171

	waic	se_waic	weight
1	-2300.4620	33.57459	1
2	-824.9275	43.15634	0
3	-822.2268	43.32588	0
4	-778.4893	48.52461	0
5	-701.8262	43.52288	0



Solution problème n°3

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

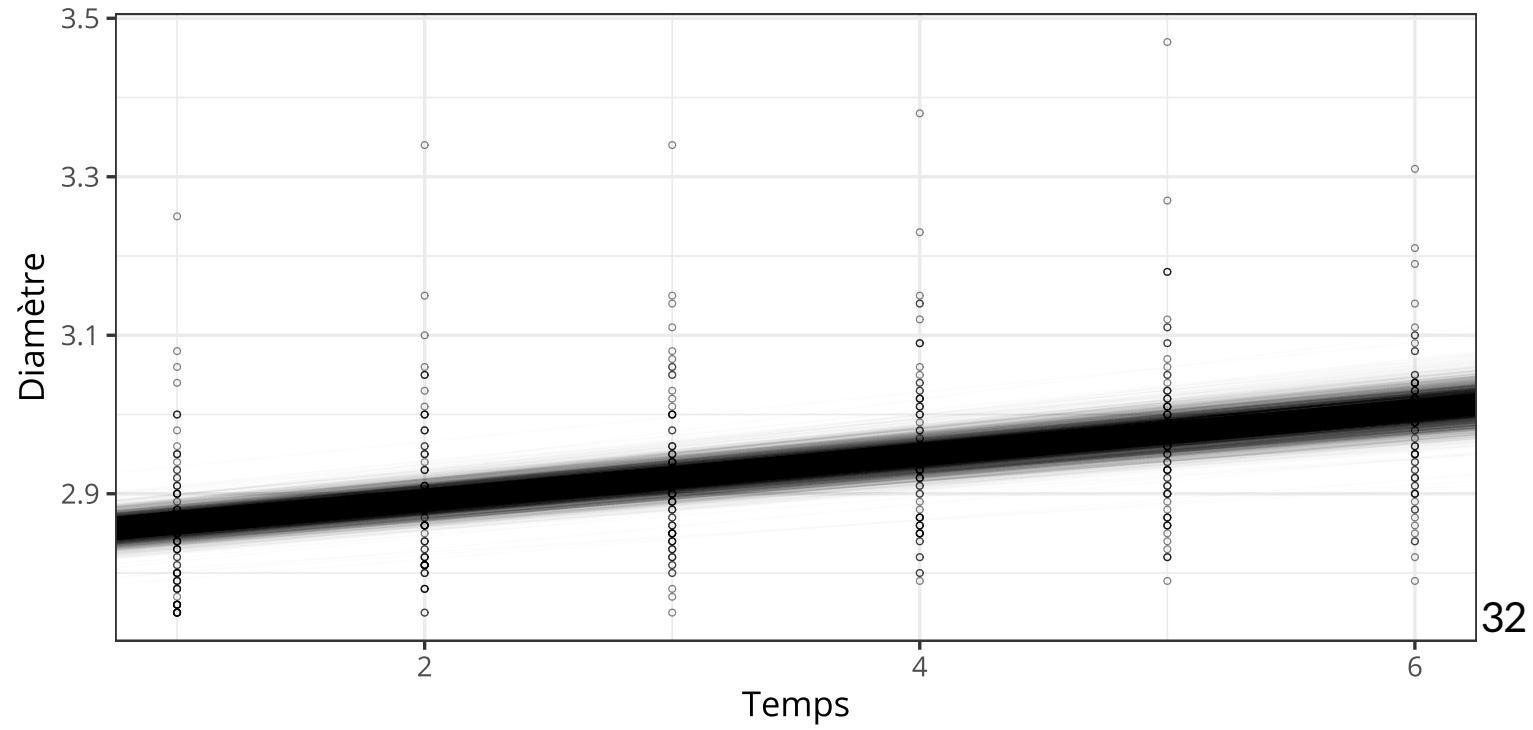
	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	2.83433475	0.0121347366	2.81096850	2.85796100
b_time	0.02853712	0.0017569523	0.02514066	0.03202085
sigma	0.01622598	0.0006714483	0.01497832	0.01758932



Solution 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")
```



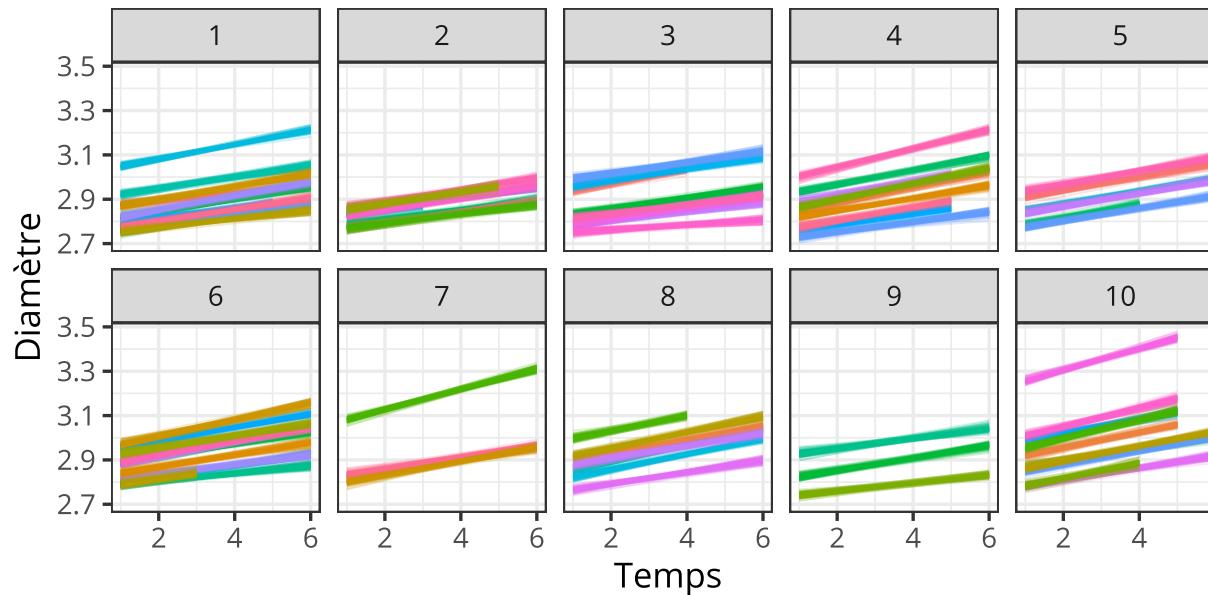
32



Solution 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 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 l'article suivant : <https://onlinelibrary.wiley.com/doi/abs/10.1111/bmsp.12195>.

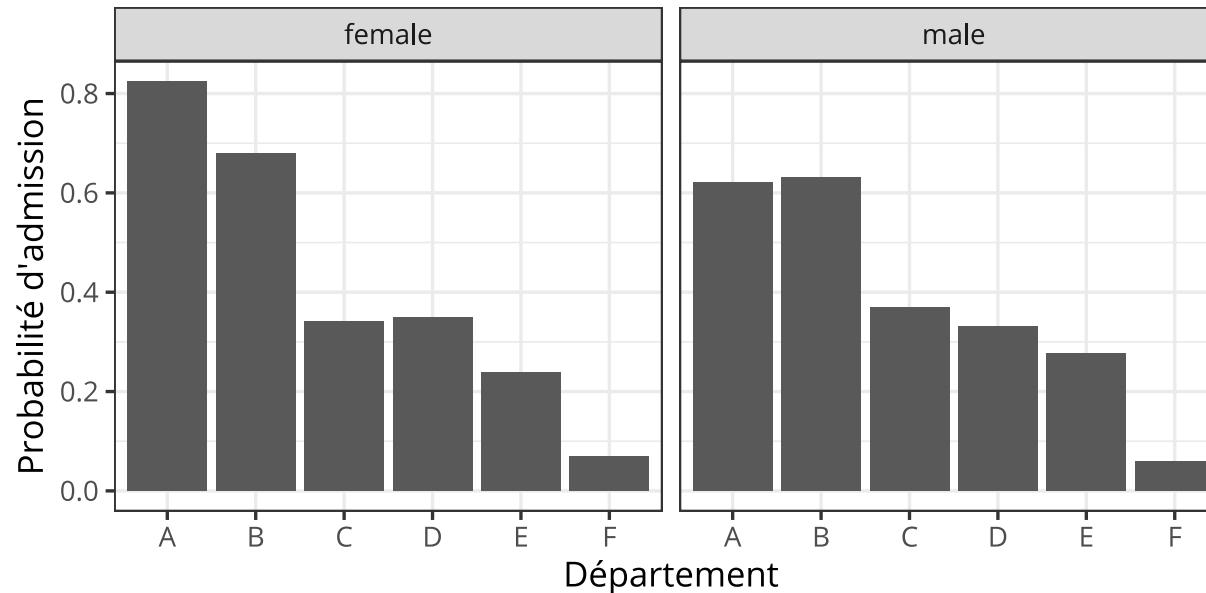
Deuxièmement, on pourrait affiner le modèle d'observation postulé pour le phénomène mesuré. Plus précisément, nous avons des informations sur la nature de la variable mesurée (le diamètre). En l'occurrence, on sait par exemple que le diamètre d'une pomme ne peut pas être négatif. On pourrait donc remplacer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



Réponse 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 problème n°4

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

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

# creating an index for department
d4$dept_id <- coercce_index(d4$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 problème n°4

```
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,
  prior = p3,
  warmup = 1000, iter = 5000,
  control = list(adapt_delta = 0.99, max_treedepth = 12),
  backend = "cmdstanr"
)
```



Réponse problème n°4

```
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	-45.39703	2.277771	6.678872	1.334714	90.79406
2	m1	-7.112165	7.622194	-52.50920	8.999190	6.431656	2.228411	105.01839
3	m2	-8.963574	6.713191	-54.36061	8.204008	9.466406	2.933153	108.72121

	se_waic	weight
1	4.555542	0.999
2	17.998380	0.001
3	16.408015	0.000



Réponse problème n°4

```
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 = 4000; warmup = 0; 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	0.84	3.09	1.00	5237
sd(gender)	0.51	0.26	0.16	1.16	1.00	4936
cor(Intercept,gender)	-0.29	0.36	-0.86	0.47	1.00	9337
		Tail_ESS				
sd(Intercept)		7942				
sd(gender)		6469				
cor(Intercept,gender)		9112				

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.57	0.67	-1.93	0.82	1.00	4105	6118
gender	-0.16	0.25	-0.65	0.33	1.00	7326	8035

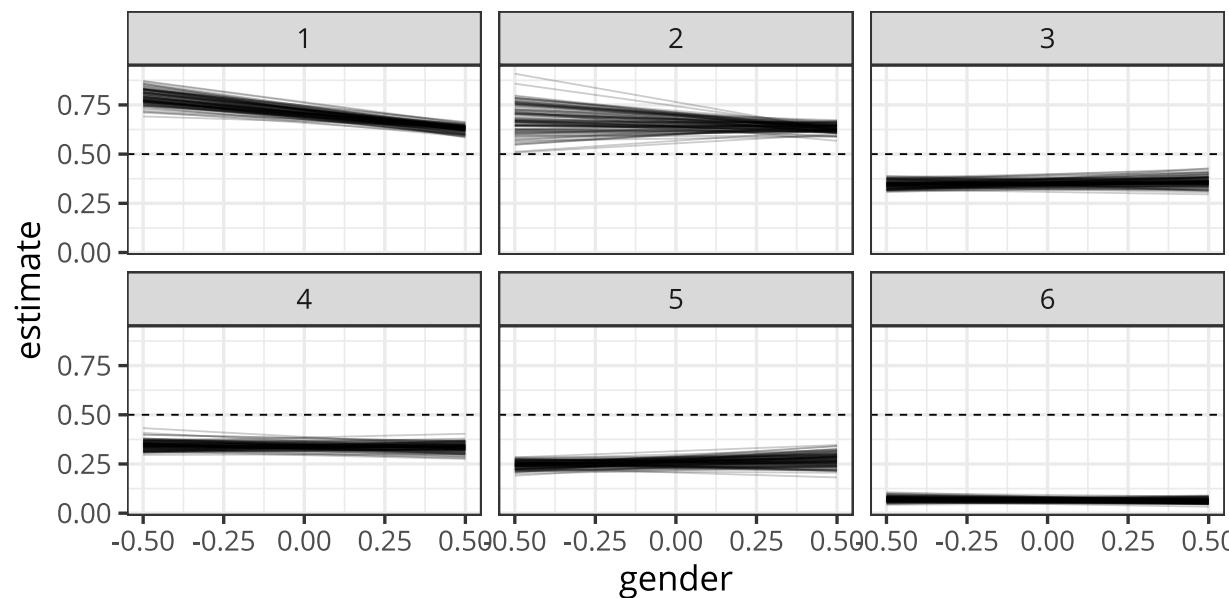
Draws were sampled using sample(hmc). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).



Réponse problème n°4

```
library(tidybayes)
library(modelr)

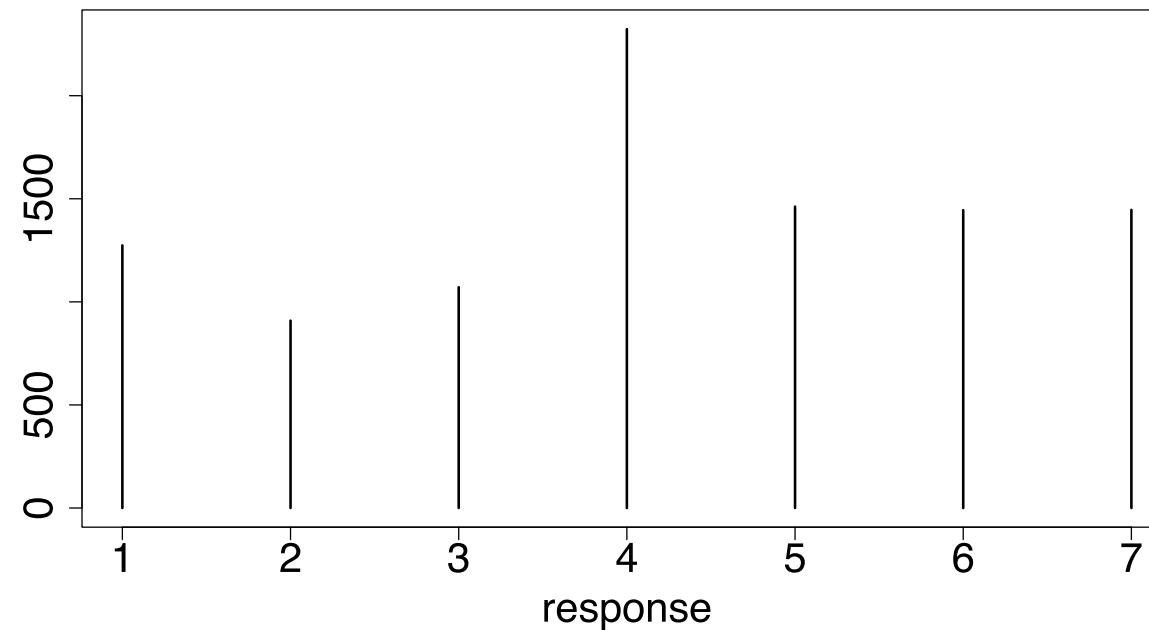
d4 %>%
  group_by(dept_id, applications) %>%
  data_grid(gender = seq_range(gender, n = 1e2) ) %>%
  add_fitted_samples(m3, newdata = ., n = 100, scale = "linear") %>%
  mutate(estimate = plogis(estimate) ) %>%
  ggplot(aes(x = gender, y = estimate, group = .iteration)) +
  geom_hline(yintercept = 0.5,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



Réponse 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...

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



Solution problème n°5

Ce type de données peut se modéliser en utilisant le modèle de régression logistique ordinaire, 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).

```
morall1 <- brm(  
  response ~ 1,  
  data = d5,  
  family = cumulative("logit"),  
  cores = parallel::detectCores(),  
  control = list(adapt_delta = 0.99),  
  backend = "cmdstanr"  
)  
  
morall2 <- brm(  
  response ~ 1 + action + intention + contact,  
  data = d5,  
  family = cumulative("logit"),  
  cores = parallel::detectCores(),  
  control = list(adapt_delta = 0.99),  
  backend = "cmdstanr"  
)
```



Solution 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*.

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

Output of model 'moral1':

Computed from 4000 by 9930 log-likelihood matrix

	Estimate	SE
elpd_waic	-18927.1	28.8
p_waic	5.9	0.0
waic	37854.2	57.7

Output of model 'moral2':

Computed from 4000 by 9930 log-likelihood matrix

	Estimate	SE
elpd_waic	-18544.9	38.1
p_waic	9.0	0.0
waic	37089.8	76.3

Model comparisons:

	elpd_diff	se_diff
moral2	0.0	0.0
moral1	-382.2	28.0



Solution problème n°5

```
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 = 1000; warmup = 0; 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	3136	2700
Intercept[2]	-2.16	0.04	-2.24	-2.08	1.00	3078	3009
Intercept[3]	-1.57	0.04	-1.65	-1.50	1.00	3405	2964
Intercept[4]	-0.55	0.04	-0.62	-0.48	1.00	3532	3141
Intercept[5]	0.12	0.04	0.05	0.19	1.00	3887	3137
Intercept[6]	1.02	0.04	0.95	1.10	1.00	3998	2722
action	-0.71	0.04	-0.79	-0.63	1.00	3505	2940
intention	-0.72	0.04	-0.79	-0.65	1.00	4242	2832
contact	-0.96	0.05	-1.06	-0.86	1.00	3693	3327

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
disc	1.00	0.00	1.00	1.00	NA	NA	NA

Draws were sampled using sample(hmc). 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).



Solution 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]]

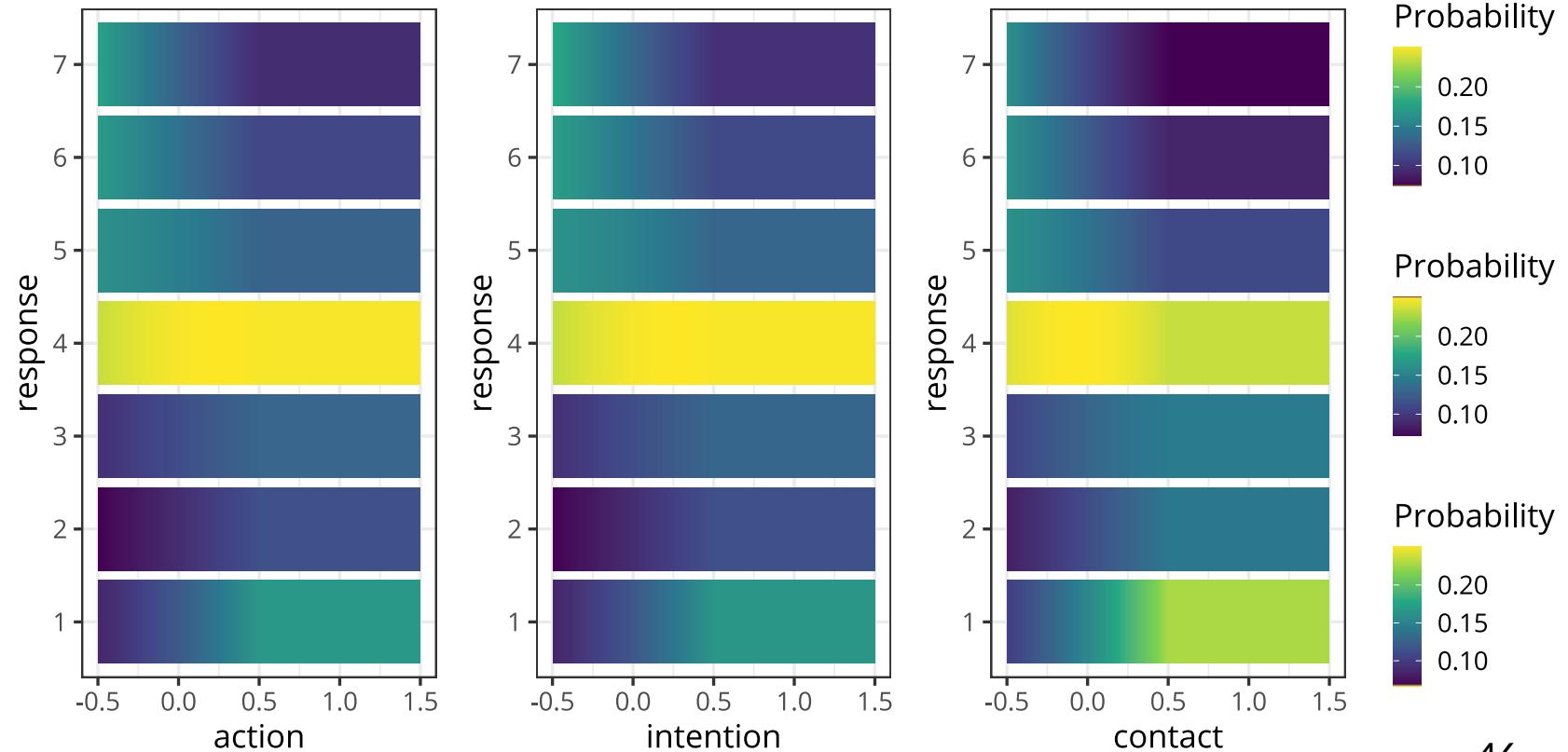
marg2 <- marginal_effects(moral2, "intention", ordinal = TRUE)
p2 <- plot(marg2, theme = theme_bw(base_size = 20, base_family = "Open Sans"), plot = FALSE) [[1]]

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



Solution problème n°5

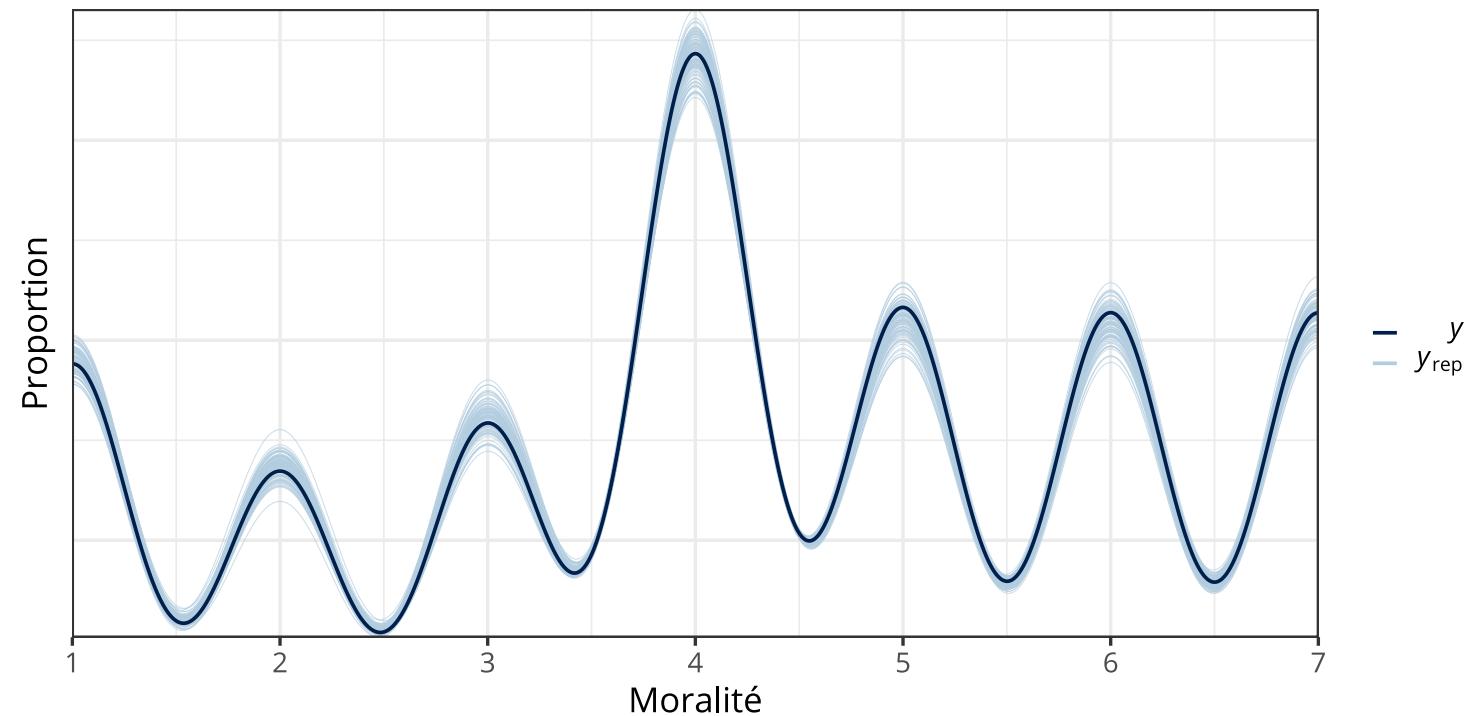
```
library(patchwork)
p1 + p2 + p3 + plot_layout(guides = "collect") & theme(legend.position = "right")
```



Solution problème n°5

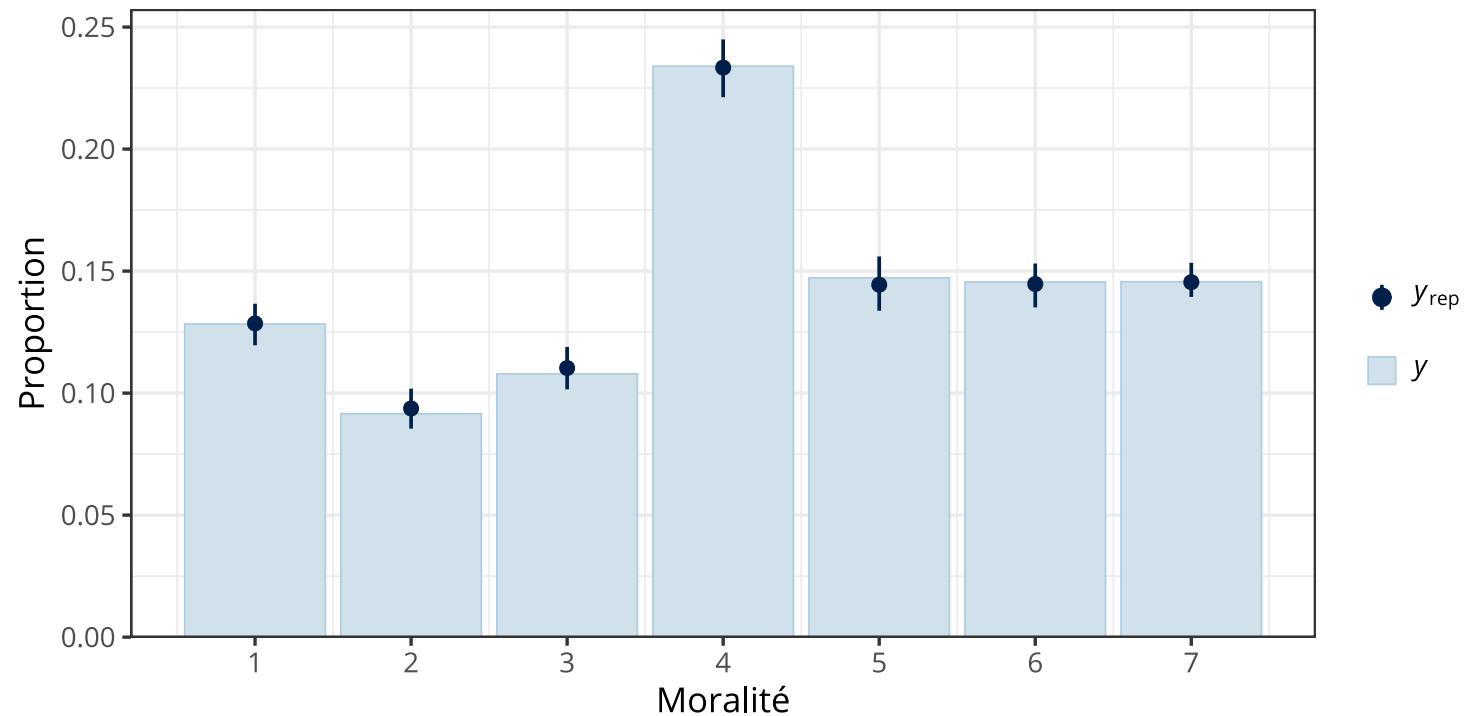
Pour plus d'informations sur la régression logistique ordinale, voir [ce lien](#), [ce tutorial](#), ou le chapitre 11 de [McElreath \(2015\)](#).

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



Solution problème n°5

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



Solution problème n°5

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

