

Introduction à la modélisation statistique bayésienne

Un cours en R et Stan avec brms

Ladislas Nalborczyk (CNRS, LPL, 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 (généralisés)

Cours n°09 : Examen final

Rappels

On considère un modèle de régression linéaire gaussien avec un prédicteur continu x . Ce modèle contient trois paramètres à estimer : l'intercept α , la pente β , et l'écart-type des "résidus" σ .

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i$$

$$\alpha \sim \text{Normal}(100, 10)$$

$$\beta \sim \text{Normal}(0, 10)$$

$$\sigma \sim \text{Exponential}(0.01)$$

Rappels

Ce modèle s'implémente simplement via `brms::brm()`.

```
1 library(brms)
2
3 priors <- c(
4   prior(normal(100, 10), class = Intercept),
5   prior(normal(0, 10), class = b),
6   prior(exponential(0.01), class = sigma)
7 )
8
9 model <- brm(
10   formula = y ~ 1 + x,
11   family = gaussian(),
12   prior = priors,
13   data = df
14 )
```

Régression multiple

On va étendre le modèle précédent en ajoutant plusieurs prédicteurs, continus et/ou catégoriels.

Pourquoi ?

- “Contrôle” des facteurs de confusion (e.g., spurious correlations, simpson’s paradox). Un facteur de confusion est un facteur (une variable aléatoire) qui “perturbe” l’association entre deux variables d’intérêts.
- Multiples causes : un phénomène peut émerger sous l’influence de multiples causes.
- Interactions : l’influence d’un prédicteur sur la variable observée peut dépendre de la valeur d’un autre prédicteur.

Associations fortuites

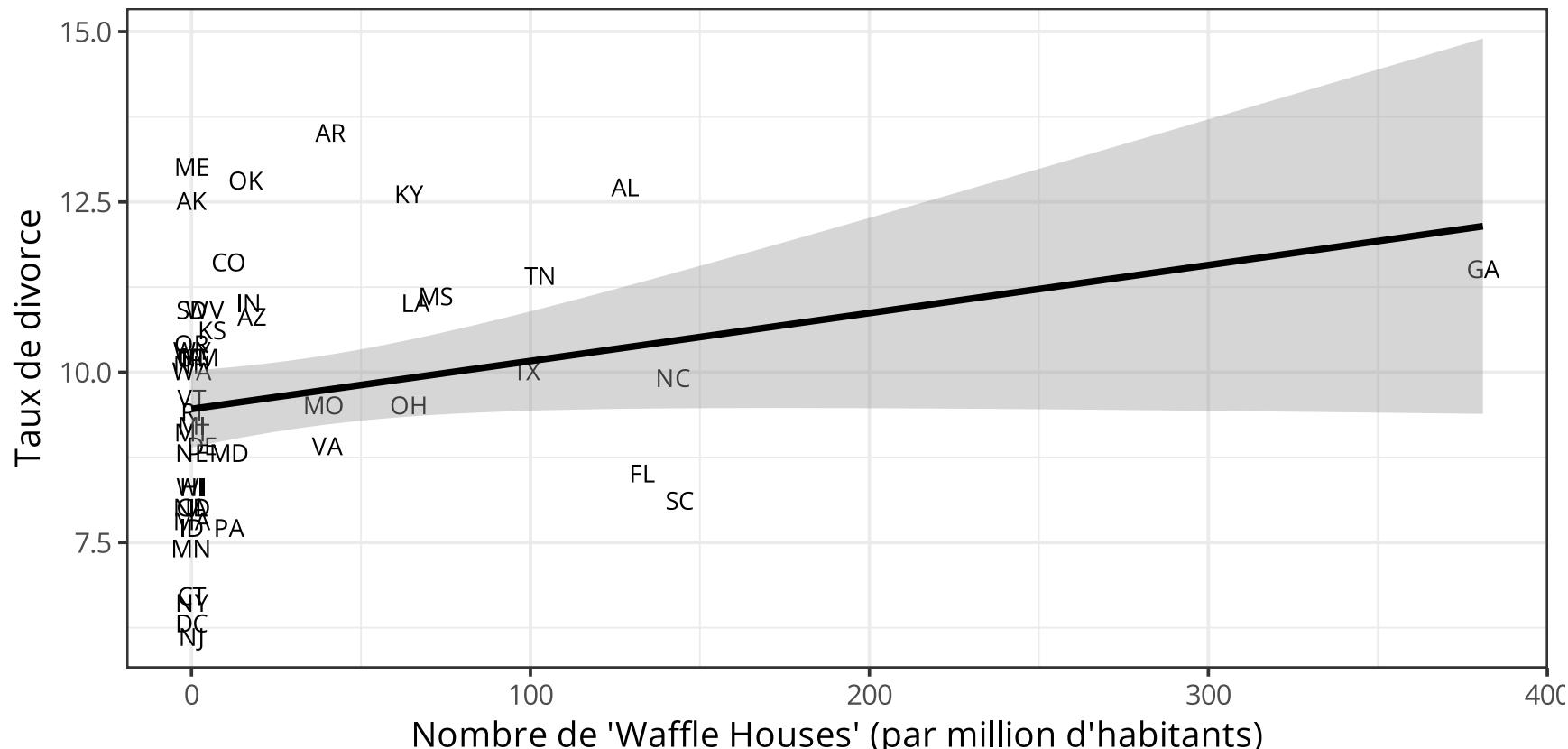
```
1 library(tidyverse)
2 library(imsb)
3
4 df1 <- open_data(waffle) # import des données dans une dataframe
5 str(df1) # affiche la structure des données
```

```
'data.frame': 50 obs. of 13 variables:
 $ Location      : chr  "Alabama" "Alaska" "Arizona" "Arkansas" ...
 $ Loc           : chr  "AL" "AK" "AZ" "AR" ...
 $ Population    : num  4.78 0.71 6.33 2.92 37.25 ...
 $ MedianAgeMarriage: num  25.3 25.2 25.8 24.3 26.8 25.7 27.6 26.6 29.7 26.4 ...
 $ Marriage      : num  20.2 26 20.3 26.4 19.1 23.5 17.1 23.1 17.7 17 ...
 $ Marriage.SE   : num  1.27 2.93 0.98 1.7 0.39 1.24 1.06 2.89 2.53 0.58 ...
 $ Divorce       : num  12.7 12.5 10.8 13.5 8 11.6 6.7 8.9 6.3 8.5 ...
 $ Divorce.SE    : num  0.79 2.05 0.74 1.22 0.24 0.94 0.77 1.39 1.89 0.32 ...
 $ WaffleHouses  : int  128 0 18 41 0 11 0 3 0 133 ...
 $ South          : int  1 0 0 1 0 0 0 0 0 1 ...
 $ Slaves1860    : int  435080 0 0 111115 0 0 0 1798 0 61745 ...
 $ Population1860: int  964201 0 0 435450 379994 34277 460147 112216 75080 140424 ...
 $ PropSlaves1860: num  0.45 0 0 0.26 0 0 0.016 0 0.44 ...
```

Associations fortuites

On observe un lien positif entre le nombre de “waffle houses” et le taux de divorce...

```
1 df1 %>%
2   ggplot(aes(x = WaffleHouses, y = Divorce) ) +
3   geom_text(aes(label = Loc), size = 4) +
4   geom_smooth(method = "lm", color = "black", se = TRUE) +
5   labs(x = "Nombre de 'Waffle Houses' (par million d'habitants)", y = "Taux de divorce")
```



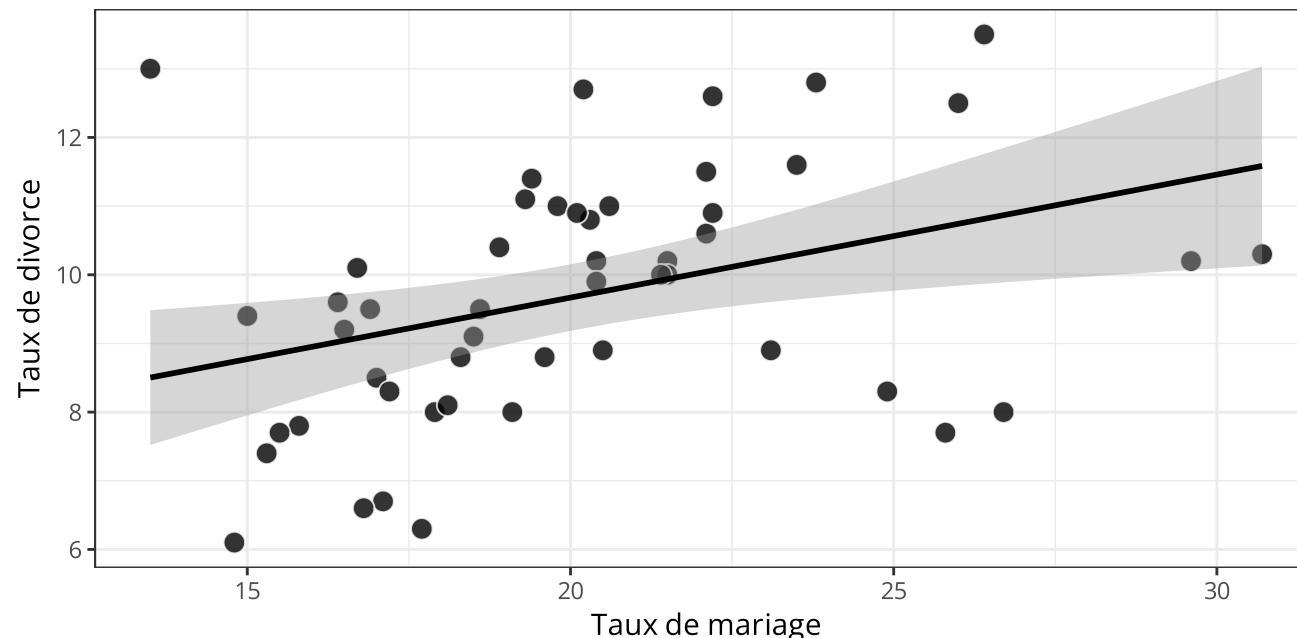
Associations fortuites

On laisse de côté les Waffle Houses. On observe un lien positif entre le taux de mariage et le taux de divorce... mais est-ce qu'on peut vraiment dire que le mariage “cause” le divorce ?

```

1 df1$Divorce.s <- scale(x = df1$Divorce, center = TRUE, scale = TRUE)
2 df1$Marriage.s <- scale(x = df1$Marriage, center = TRUE, scale = TRUE)
3
4 df1 %>%
5   ggplot(aes(x = Marriage, y = Divorce) ) +
6   geom_point(pch = 21, color = "white", fill = "black", size = 5, alpha = 0.8) +
7   geom_smooth(method = "lm", color = "black", se = TRUE) +
8   labs(x = "Taux de mariage", y = "Taux de divorce")

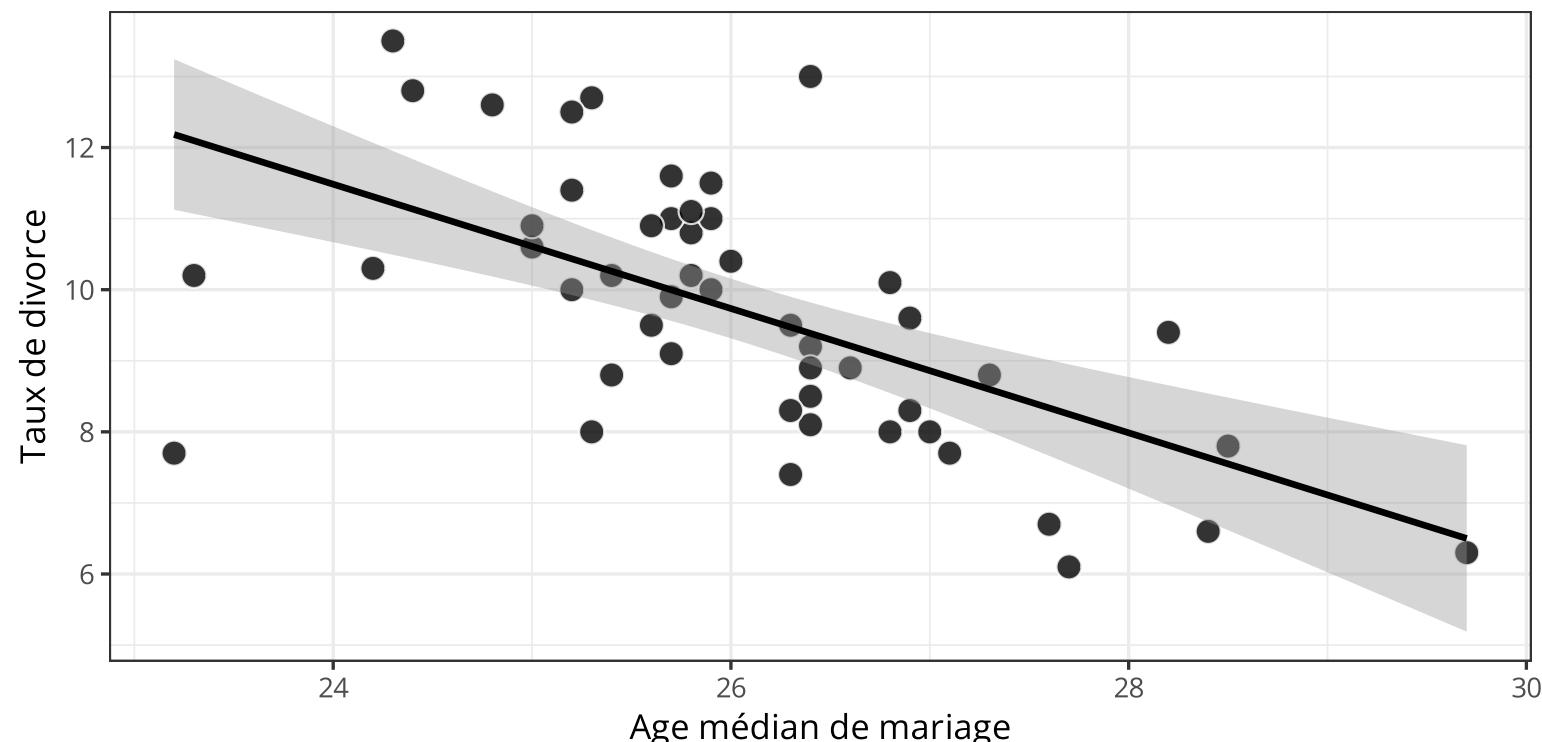
```



Associations fortuites

On observe l'association inverse entre le taux de divorce et l'âge médian de mariage.

```
1 df1$MedianAgeMarriage.s <- scale(x = df1$MedianAgeMarriage, center = TRUE, scale = TRUE)
2
3 df1 %>%
4   ggplot(aes(x = MedianAgeMarriage, y = Divorce)) +
5   geom_point(pch = 21, color = "white", fill = "black", size = 5, alpha = 0.8) +
6   geom_smooth(method = "lm", color = "black", se = TRUE) +
7   labs(x = "Age médian de mariage", y = "Taux de divorce")
```



Influence du taux de mariage sur le taux de divorce

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_R R_i$$

$$\alpha \sim \text{Normal}(0, 10)$$

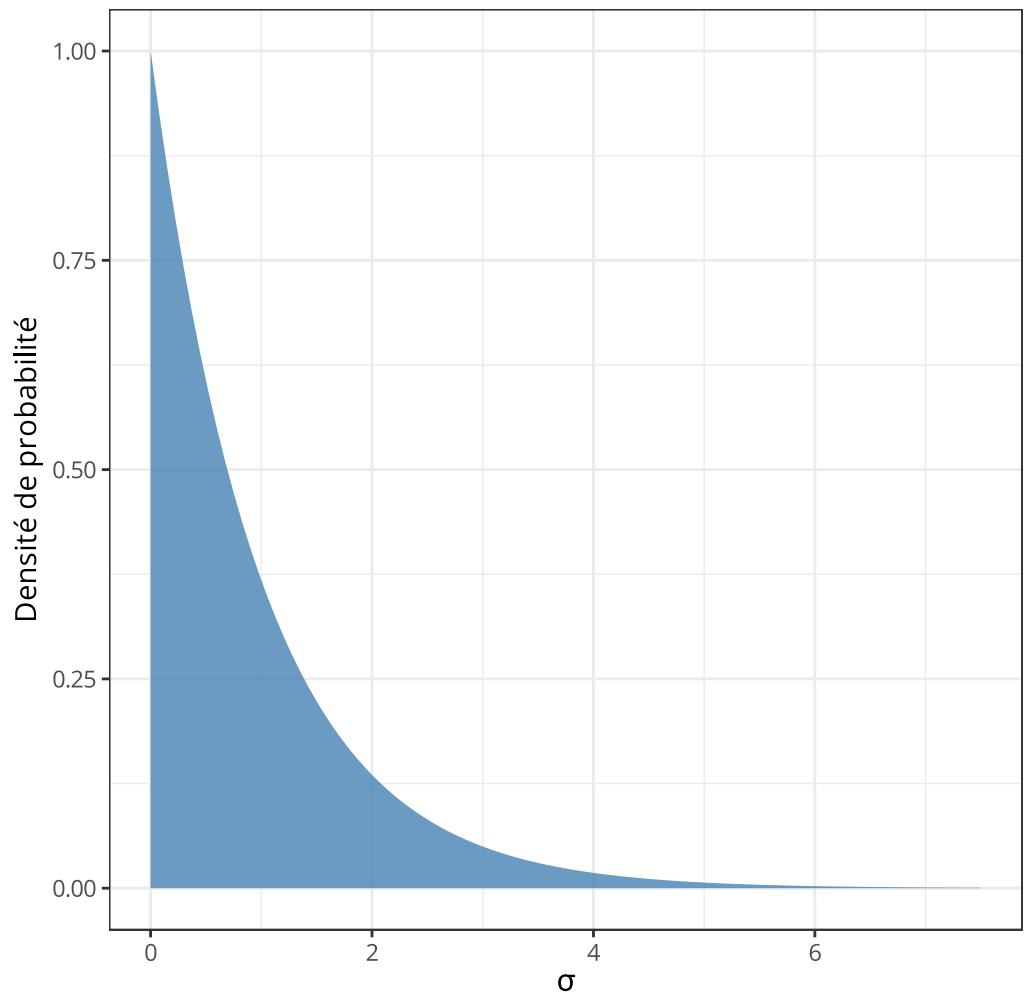
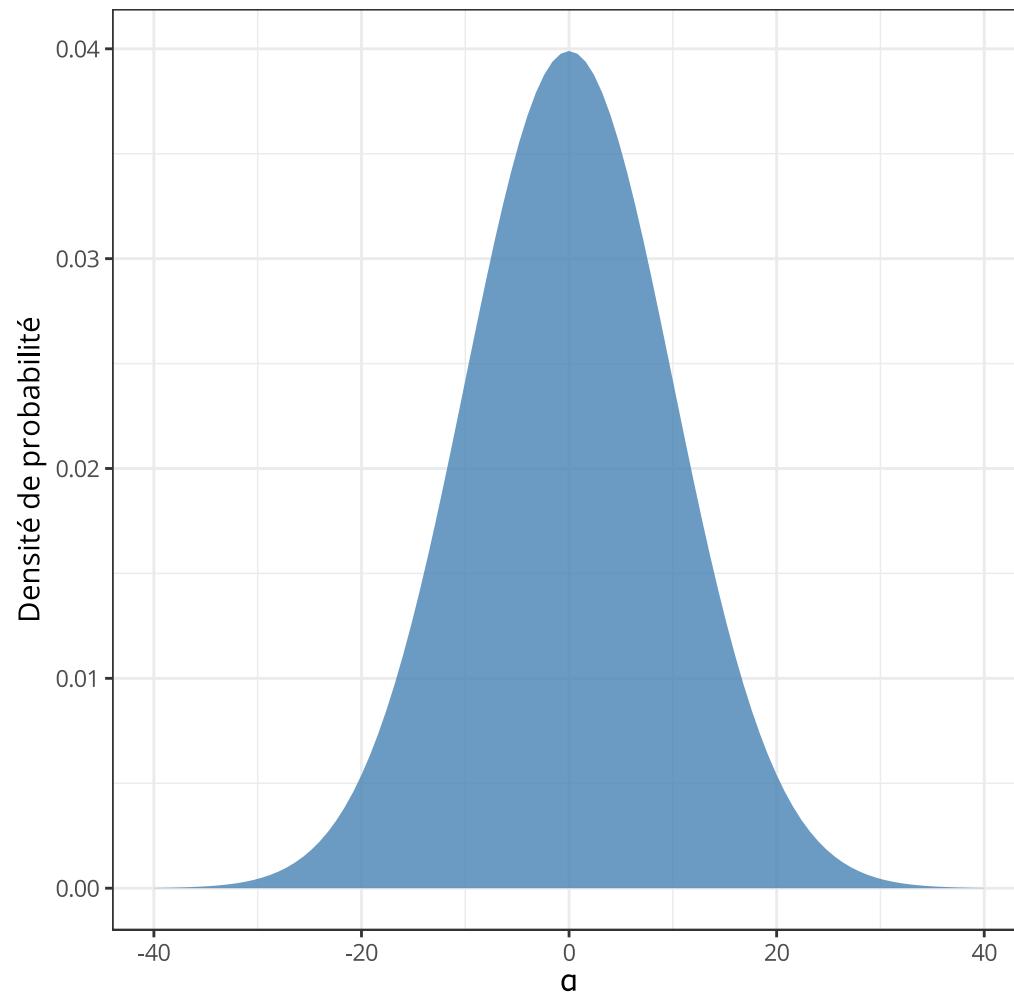
$$\beta_R \sim \text{Normal}(0, 1)$$

$$\sigma \sim \text{Exponential}(1)$$

```

1 priors <- c(
2   prior(normal(0, 10), class = Intercept),
3   prior(normal(0, 1), class = b),
4   prior(exponential(1), class = sigma)
5 )
6
7 mod1 <- brm(
8   formula = Divorce.s ~ 1 + Marriage.s,
9   family = gaussian(),
10  prior = priors,
11  # for prior predictive checking
12  sample_prior = TRUE,
13  data = df1
14 )
```

Représentation visuelle des priors



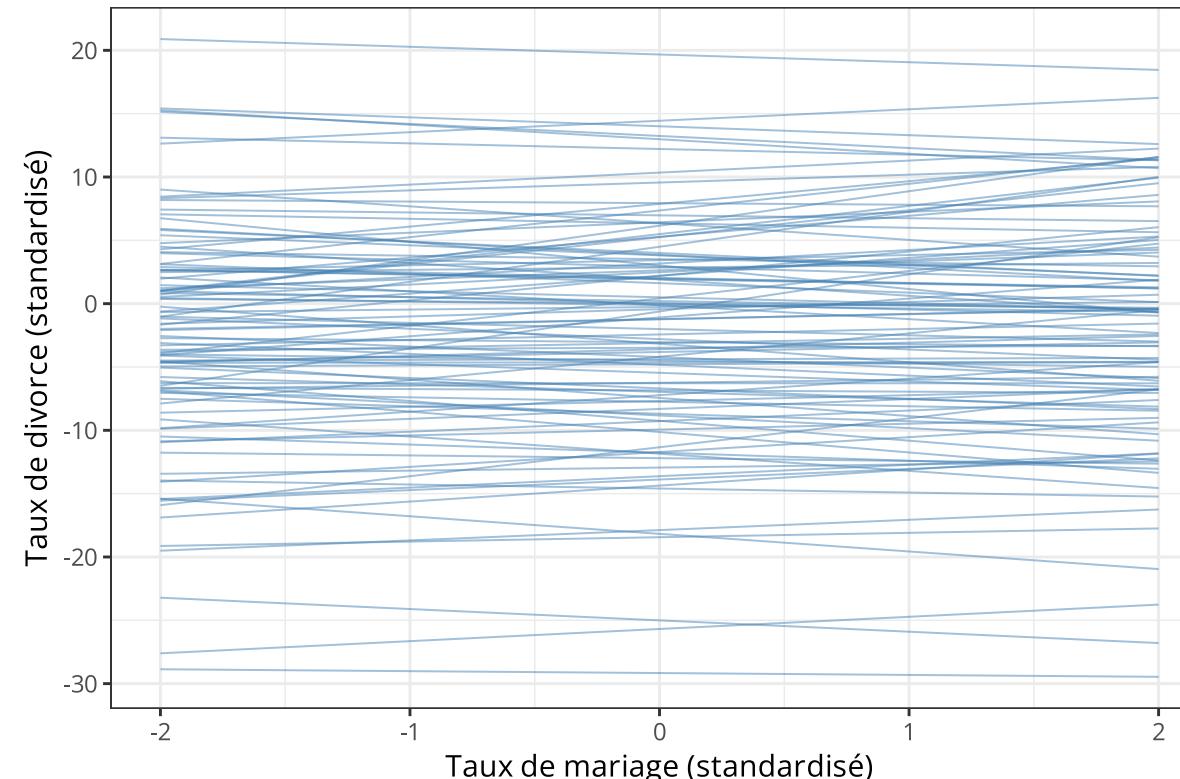
Prédictions a priori (pour μ)

```
1 # getting the samples from the prior distribution
2 prior <- prior_samples(mod1)
3
4 # displaying the first six samples
5 head(prior)
```

	Intercept	b	sigma
1	15.47924	-1.2758797	0.4283846
2	17.44202	-0.2332308	0.2330008
3	16.89468	1.1127670	1.3821642
4	-11.09762	0.1962791	0.5363238
5	-12.76362	-0.9736568	0.5207182
6	10.50564	-0.8778513	1.0662448

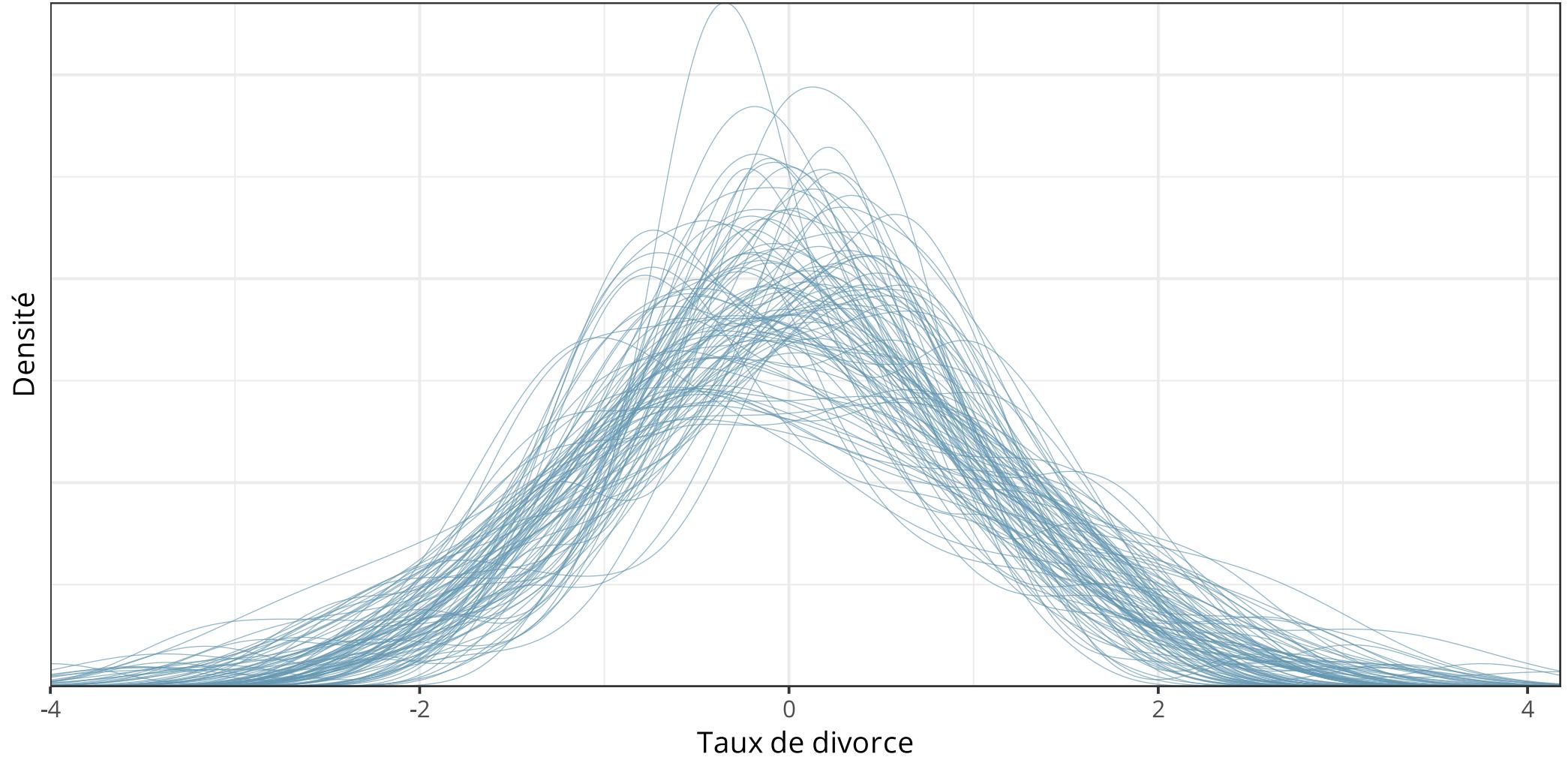
Prédictions a priori (pour μ)

```
1 prior %>%
2   sample_n(size = 1e2) %>%
3   rownames_to_column("draw") %>%
4   expand(nesting(draw, Intercept, b), a = c(-2, 2)) %>%
5   mutate(d = Intercept + b * a) %>%
6   ggplot(aes(x = a, y = d)) +
7   geom_line(aes(group = draw), color = "steelblue", size = 0.5, alpha = 0.5) +
8   labs(x = "Taux de mariage (standardisé)", y = "Taux de divorce (standardisé)")
```



Prédictions a priori (pour D_i)

```
1 pp_check(object = mod1, prefix = "ppd", ndraws = 1e2) + labs(x = "Taux de divorce", y = "Densité")
```



Influence du taux de mariage

```
1 summary(mod1)
```

Family: gaussian
 Links: mu = identity
 Formula: Divorce.s ~ 1 + Marriage.s
 Data: df1 (Number of observations: 50)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.00	0.14	-0.27	0.26	1.00	3076	2626
Marriage.s	0.37	0.14	0.10	0.63	1.00	3849	2800

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.95	0.10	0.78	1.16	1.00	4295	3240

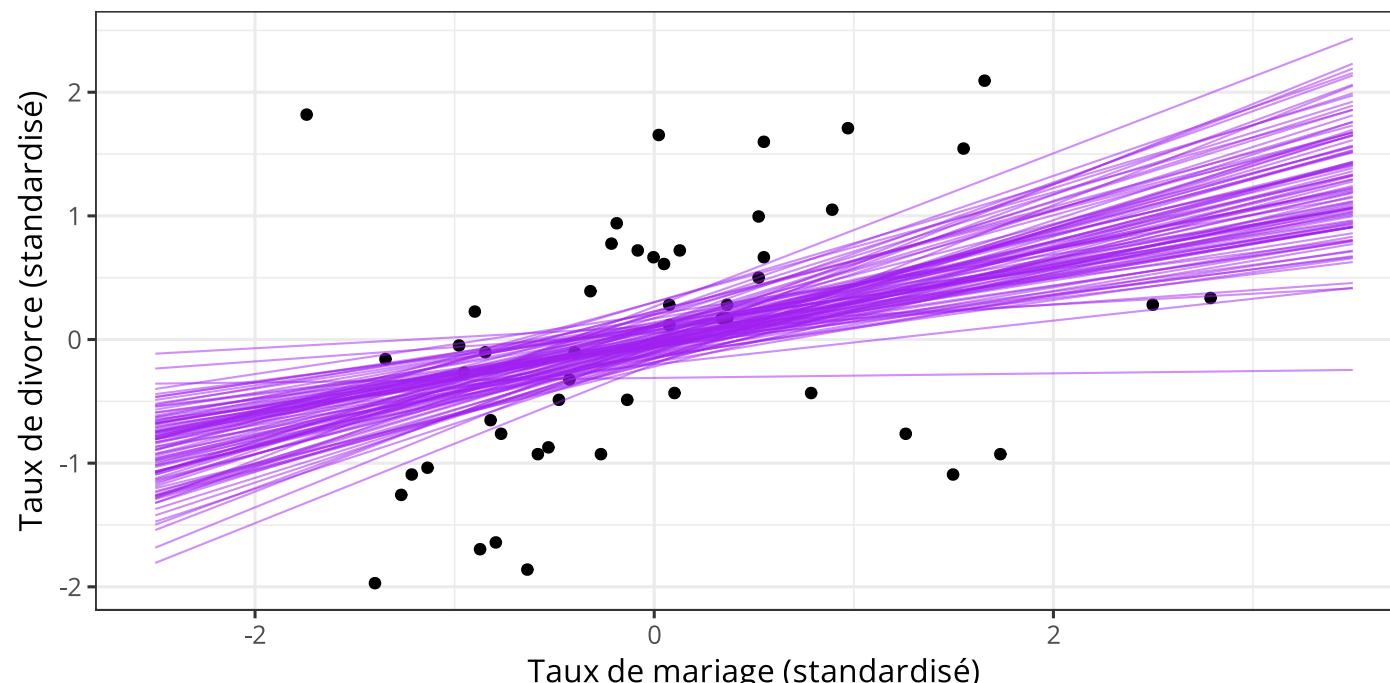
Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Prédictions a posteriori

```

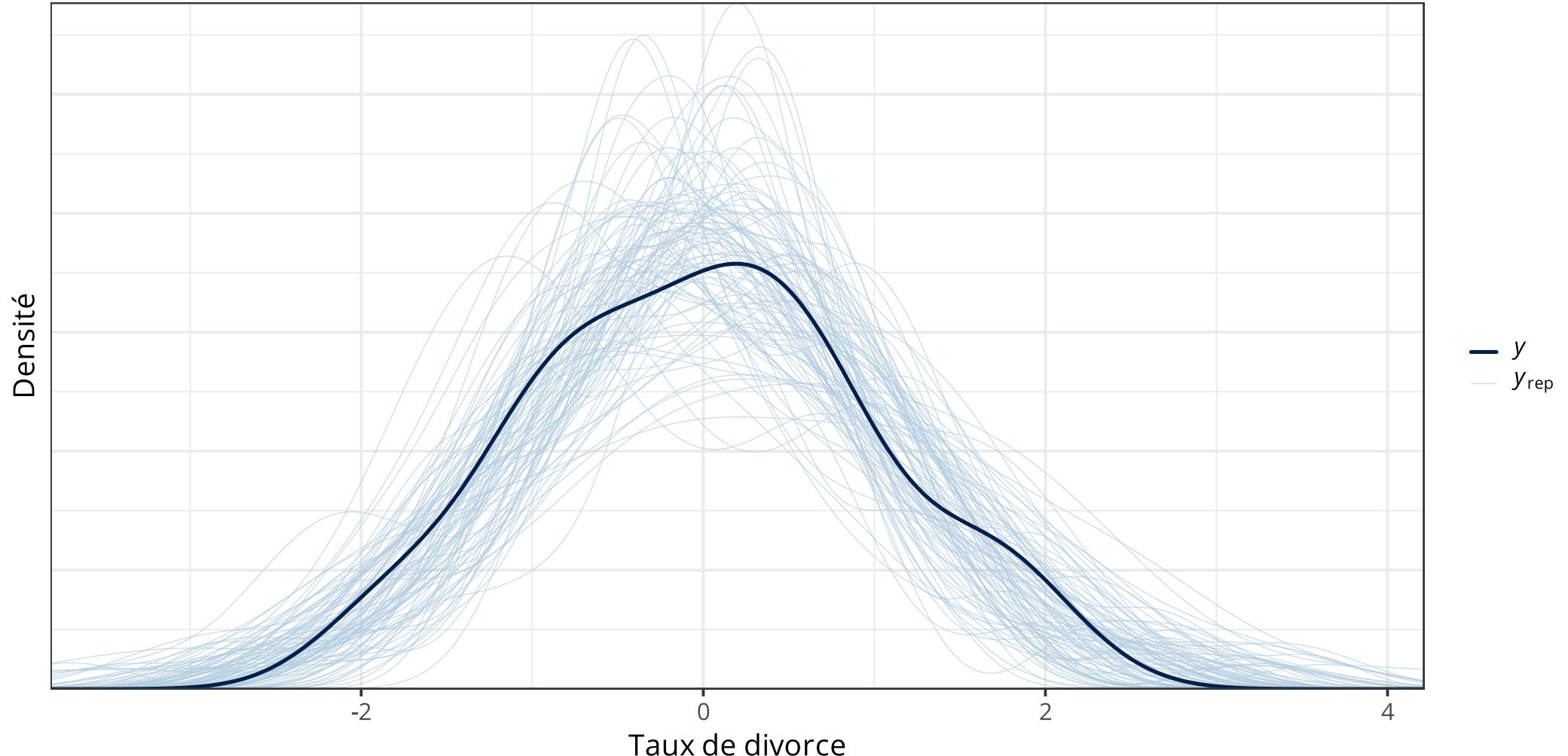
1 nd <- data.frame(Marriage.s = seq(from = -2.5, to = 3.5, length.out = 1e2) )
2
3 as_draws_df(x = mod1, pars = "^b_") %>%
4   sample_n(size = 1e2) %>%
5   expand(nesting(.draw, b_Intercept, b_Marriage.s), a = c(-2.5, 3.5) ) %>%
6   mutate(d = b_Intercept + b_Marriage.s * a) %>%
7   ggplot(aes(x = a, y = d) ) +
8   geom_point(data = df1, aes(x = Marriage.s, y = Divorce.s), size = 2) +
9   geom_line(aes(group = .draw), color = "purple", size = 0.5, alpha = 0.5) +
10  labs(x = "Taux de mariage (standardisé)", y = "Taux de divorce (standardisé)")

```



Prédictions a posteriori (pour D_i)

```
1 pp_check(object = mod1, ndraws = 1e2) + labs(x = "Taux de divorce", y = "Densité")
```



Influence de l'âge médian de mariage

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_A A_i$$

$$\alpha \sim \text{Normal}(0, 10)$$

$$\beta_A \sim \text{Normal}(0, 1)$$

$$\sigma \sim \text{Exponential}(1)$$

```

1 priors <- c(
2   prior(normal(0, 10), class = Intercept),
3   prior(normal(0, 1), class = b),
4   prior(exponential(1), class = sigma)
5 )
6
7 mod2 <- brm(
8   formula = Divorce.s ~ 1 + MedianAgeMarriage.s,
9   family = gaussian(),
10  prior = priors,
11  data = df1
12 )
```

Influence de l'âge médian de mariage

```
1 summary(mod2)
```

Family: gaussian
 Links: mu = identity
 Formula: Divorce.s ~ 1 + MedianAgeMarriage.s
 Data: df1 (Number of observations: 50)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.00	0.12	-0.23	0.23	1.00	4166	2429
MedianAgeMarriage.s	-0.59	0.12	-0.82	-0.35	1.00	3873	2865

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.82	0.08	0.68	1.01	1.00	3376	2824

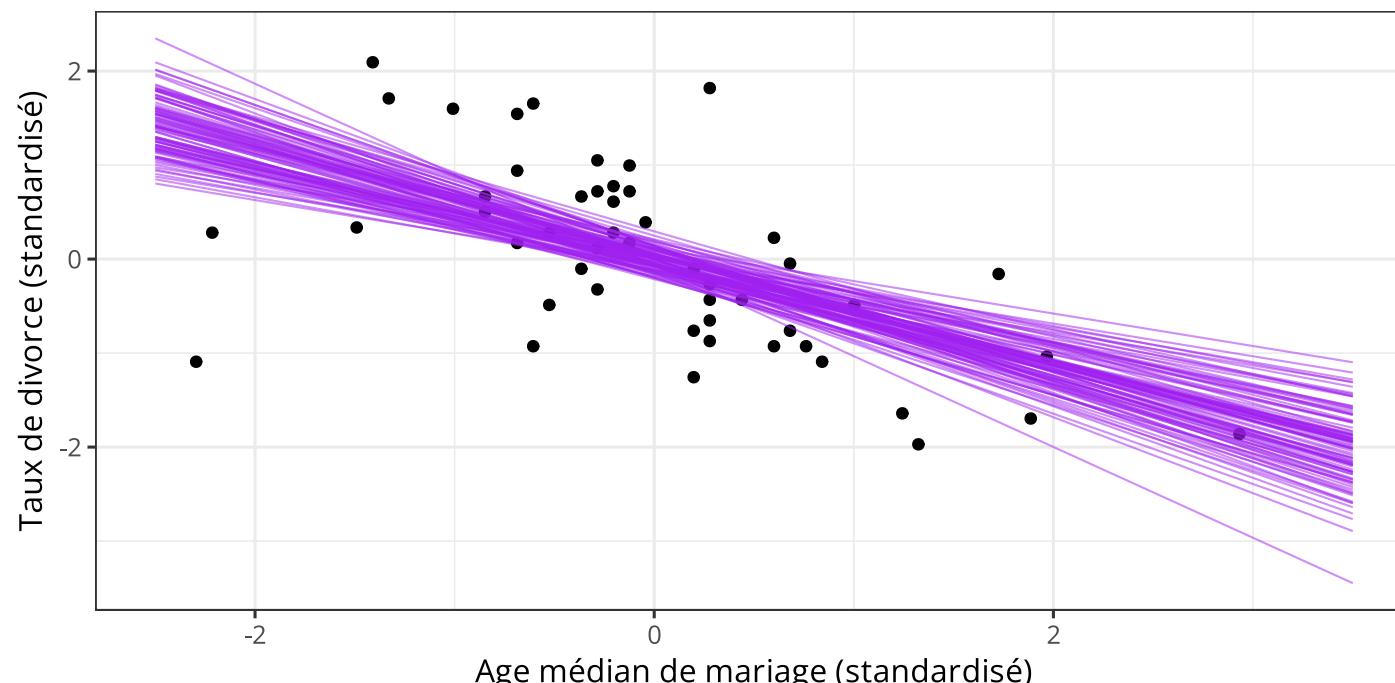
Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS and Tail_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Prédictions a posteriori

```

1 nd <- data.frame(MedianAgeMarriage.s = seq(from = -3, to = 3.5, length.out = 1e2) )
2
3 as_draws_df(x = mod2, pars = "^b_") %>%
4   sample_n(size = 1e2) %>%
5   expand(nesting(.draw, b_Intercept, b_MedianAgeMarriage.s), a = c(-2.5, 3.5) ) %>%
6   mutate(d = b_Intercept + b_MedianAgeMarriage.s * a) %>%
7   ggplot(aes(x = a, y = d) ) +
8   geom_point(data = df1, aes(x = MedianAgeMarriage.s, y = Divorce.s), size = 2) +
9   geom_line(aes(group = .draw), color = "purple", size = 0.5, alpha = 0.5) +
10  labs(x = "Age médian de mariage (standardisé)", y = "Taux de divorce (standardisé)")

```



Régression multiple

Quelle est la valeur prédictive d'une variable, une fois que je connais tous les autres prédicteurs ?

$$D_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_R R_i + \beta_A A_i$$

$$\alpha \sim \text{Normal}(0, 10)$$

$$\beta_R, \beta_A \sim \text{Normal}(0, 1)$$

$$\sigma \sim \text{Exponential}(1)$$

Ce modèle répond à deux questions :

- Une fois connu le taux de mariage, quelle valeur ajoutée apporte la connaissance de l'âge médian de mariage ?
- Une fois connu l'âge médian de mariage, quelle valeur ajoutée apporte la connaissance du taux de mariage ?

Régression multiple

```
1 priors <- c(  
2   prior(normal(0, 10), class = Intercept),  
3   prior(normal(0, 1), class = b),  
4   prior(exponential(1), class = sigma)  
5 )  
6  
7 mod3 <- brm(  
8   formula = Divorce.s ~ 1 + Marriage.s + MedianAgeMarriage.s,  
9   family = gaussian(),  
10  prior = priors,  
11  data = df1  
12 )
```

Régression multiple

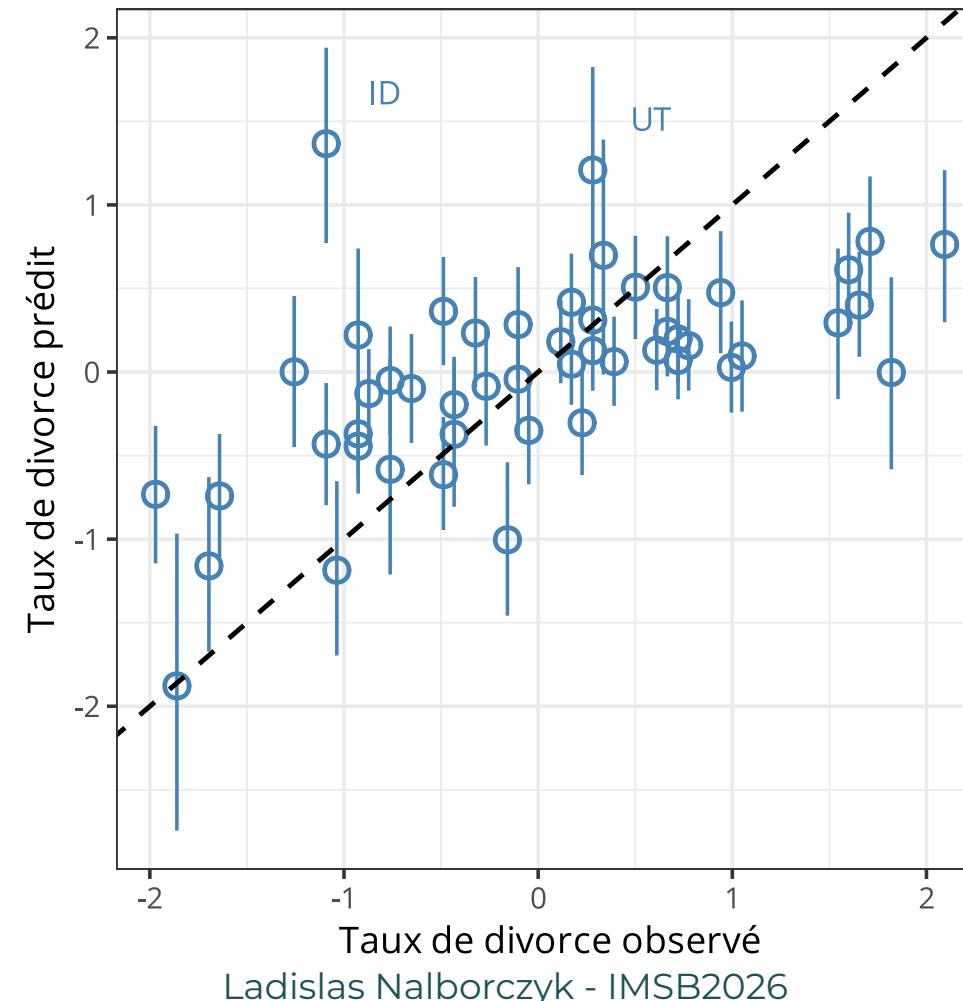
Interprétation : Une fois qu'on connaît l'âge median de mariage dans un état, connaître le taux de mariage de cet état n'apporte pas vraiment d'information supplémentaire...

```
1 posterior_summary(x = mod3, pars = "^.b_")
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	0.0002723668	0.1174717	-0.2291173	0.2364241
b_Marriage.s	-0.1042221910	0.1710225	-0.4388631	0.2390672
b_MedianAgeMarriage.s	-0.6632815118	0.1719006	-1.0101372	-0.3205263

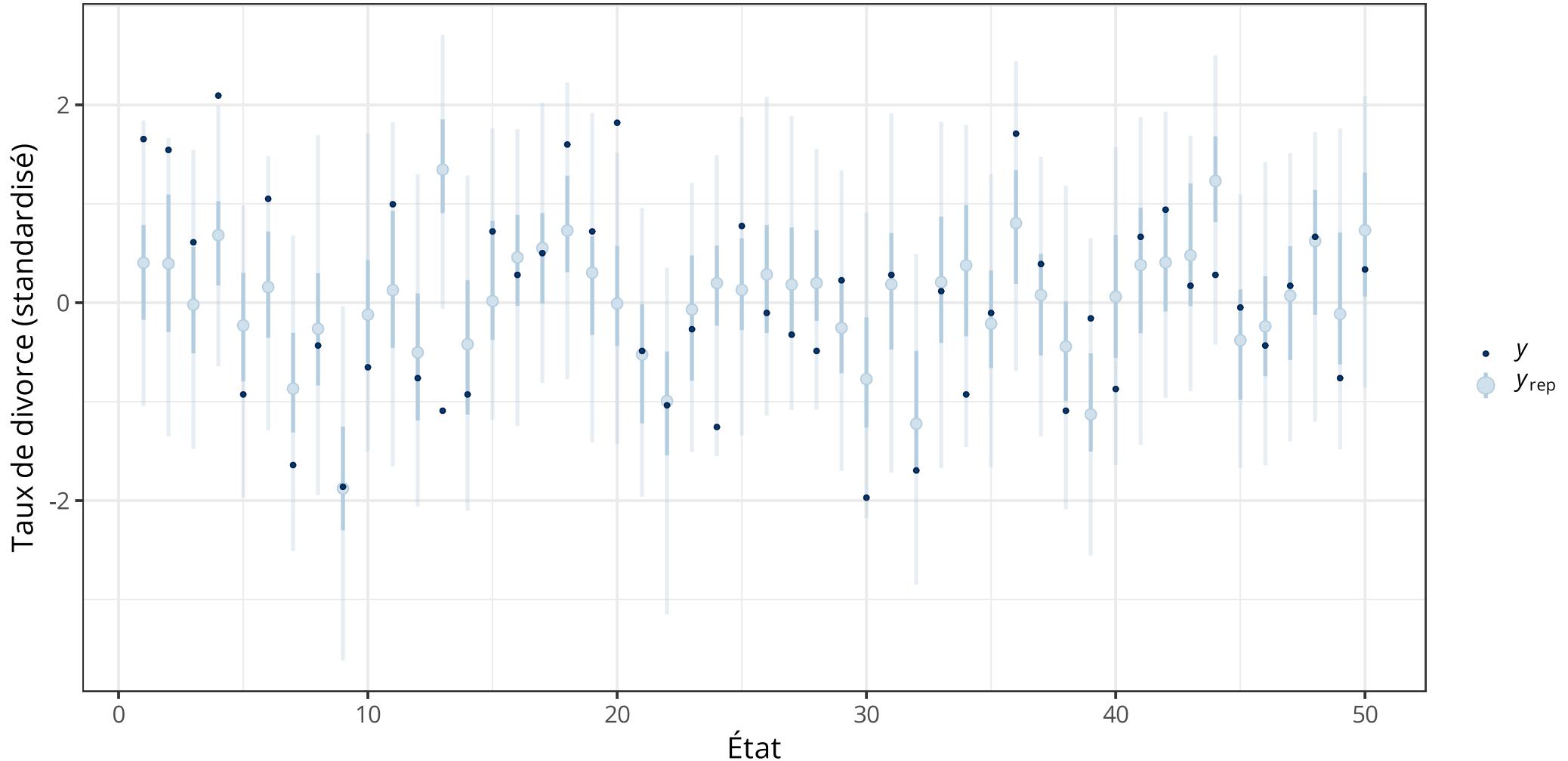
Visualiser les prédictions du modèle

En plus de l'interprétation des paramètres, il est important d'évaluer les prédictions du modèle en les comparant aux données observées. Cela nous permet de savoir si le modèle rend bien compte des données et (surtout) où est-ce que le modèle échoue. On peut comparer le taux de divorce observé dans chaque état au taux de divorce prédit par notre modèle (la ligne diagonale représente une prédition parfaite).

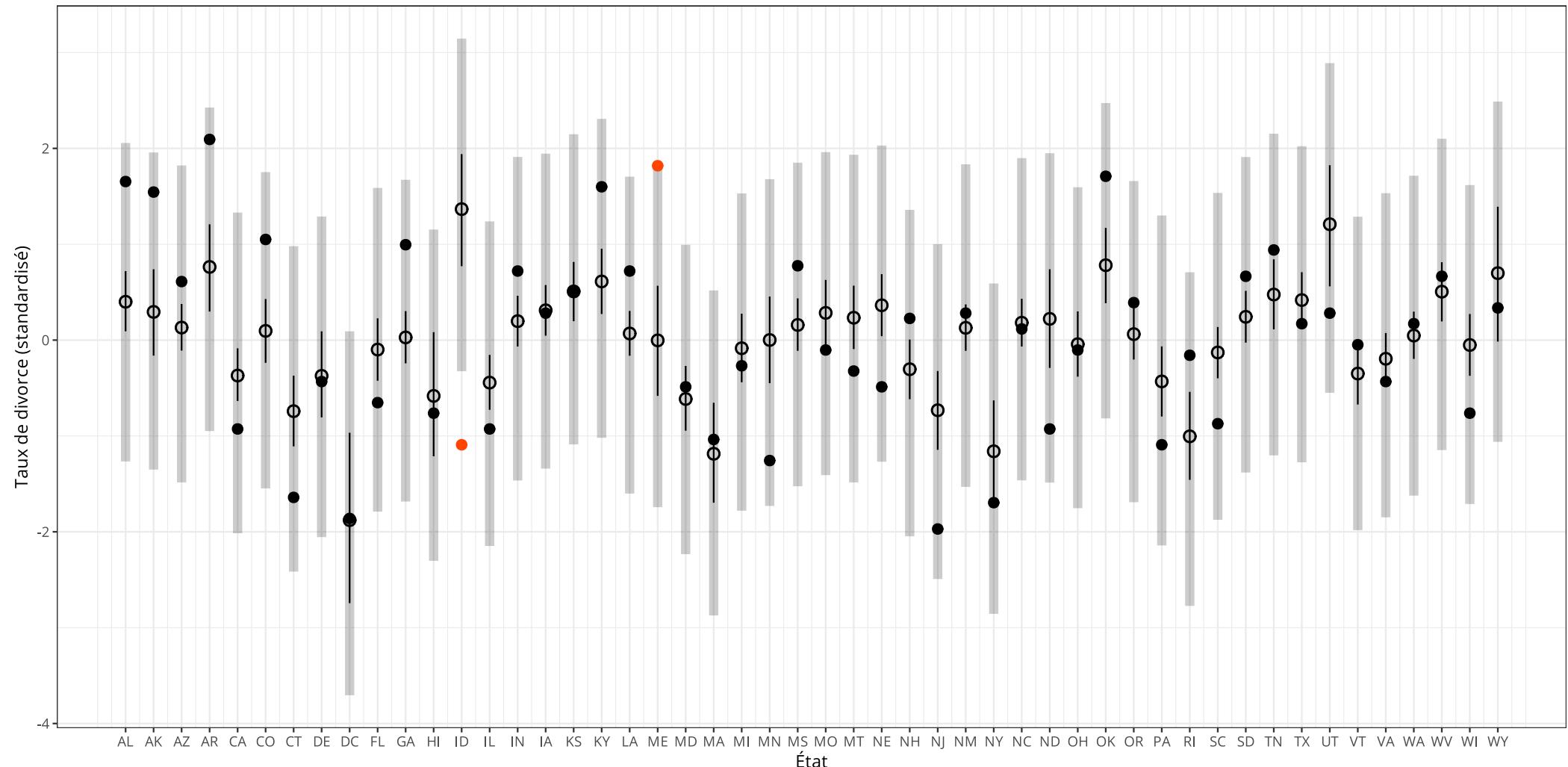


Visualiser les prédictions du modèle

```
1 pp_check(object = mod3, type = "intervals", ndraws = 1e2, prob = 0.5, prob_outer = 0.95) +  
2   labs(x = "État", y = "Taux de divorce (standardisé)")
```



Visualiser les prédictions du modèle



Toujours plus de prédicteurs

Pourquoi ne pas simplement construire un modèle incluant tous les prédicteurs et regarder ce qu'il se passe ?

- Raison n°1 : Multicolinéarité
- Raison n°2 : Post-treatment bias
- Raison n°3 : Overfitting (cf. Cours n°07)

Multicolinéarité

Situation dans laquelle certains prédicteurs sont très fortement corrélés. Par exemple, essayons de prédire la taille d'un individu par la taille de ses jambes.

```
1 set.seed(666) # afin de pouvoir reproduire les résultats
2
3 N <- 100 # nombre d'individus
4 height <- rnorm(n = N, mean = 178, sd = 10) # génère N observations
5 leg_prop <- runif(n = N, min = 0.4, max = 0.5) # taille des jambes (proportion taille totale)
6 leg_left <- leg_prop * height + rnorm(n = N, mean = 0, sd = 1) # taille jambe gauche (+ erreur)
7 leg_right <- leg_prop * height + rnorm(n = N, mean = 0, sd = 1) # taille jambe droite (+ erreur)
8 df2 <- data.frame(height, leg_left, leg_right) # création d'une dataframe
9
10 head(df2) # affiche les six première lignes
```

	height	leg_left	leg_right
1	185.5331	75.92846	76.92202
2	198.1435	84.11099	86.25709
3	174.4487	70.25378	70.37146
4	198.2817	86.82322	86.28113
5	155.8313	75.84092	78.42467
6	185.5840	86.41507	85.08914

Multicolinéarité

On fit un modèle avec deux prédicteurs : un pour la taille de chaque jambe.

```
1 priors <- c(  
2   prior(normal(178, 10), class = Intercept),  
3   prior(normal(0, 10), class = b),  
4   prior(exponential(0.01), class = sigma)  
5 )  
6  
7 mod4 <- brm(  
8   formula = height ~ 1 + leg_left + leg_right,  
9   prior = priors,  
10  family = gaussian,  
11  data = df2  
12 )
```

Multicolinéarité

Les estimations semblent étranges... mais le modèle ne fait que répondre à la question qu'on lui pose : Une fois que je connais la taille de la jambe gauche, quelle est la valeur prédictive de la taille de la jambe droite (et vice versa) ?

```
1 summary(mod4) # look at the SE...
```

Family: gaussian
 Links: mu = identity
 Formula: height ~ 1 + leg_left + leg_right
 Data: df2 (Number of observations: 100)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	100.86	10.31	81.42	121.20	1.00	3934	2539
leg_left	0.73	0.61	-0.46	1.90	1.00	1985	2149
leg_right	0.23	0.62	-0.96	1.43	1.00	1961	2167

Further Distributional Parameters:

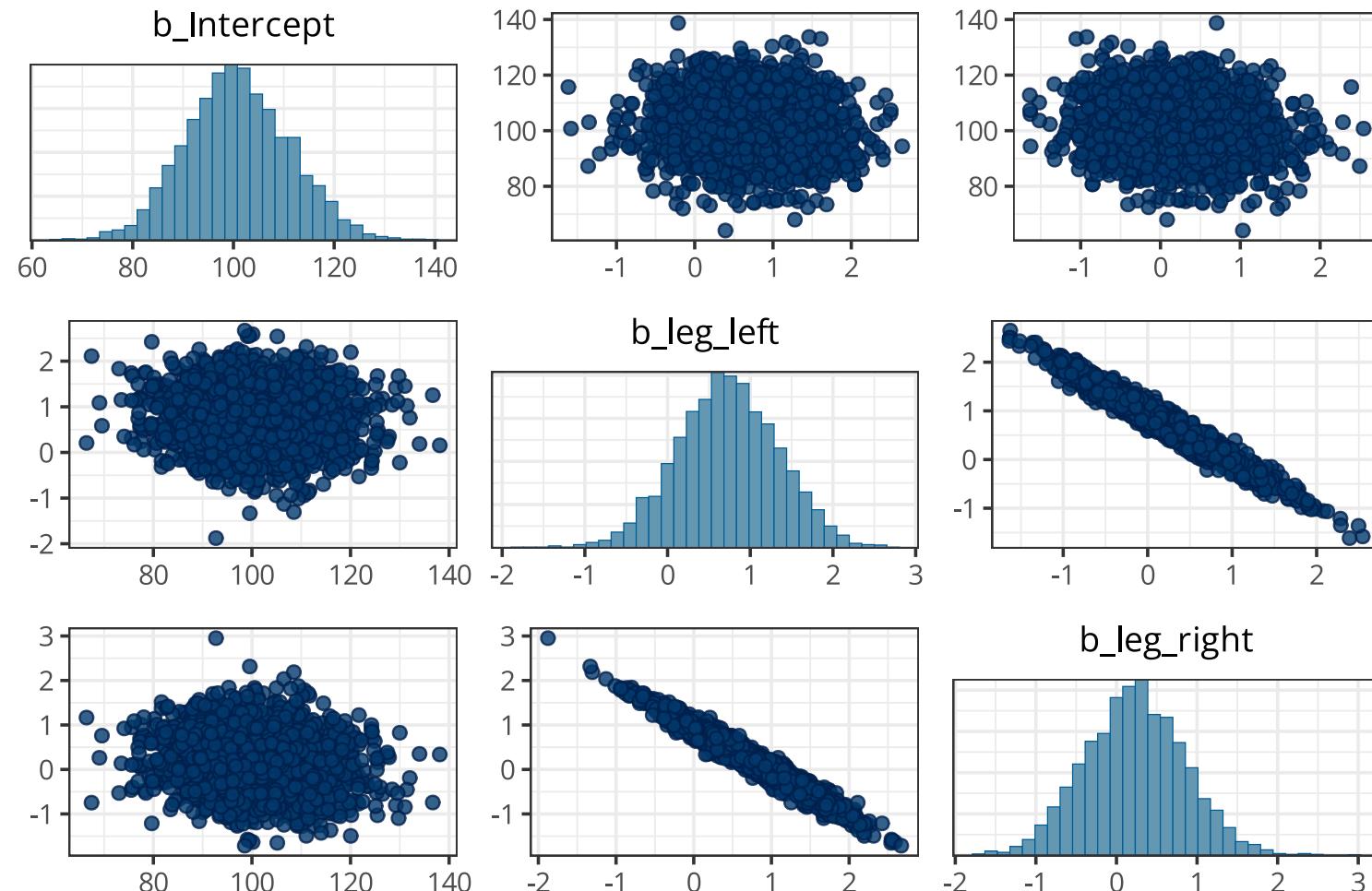
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	8.28	0.61	7.21	9.61	1.00	2517	2273

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).

Multicolinéarité

Comment traquer la colinéarité de deux prédicteurs ? On peut examiner la distribution postérieure de ces deux paramètres.

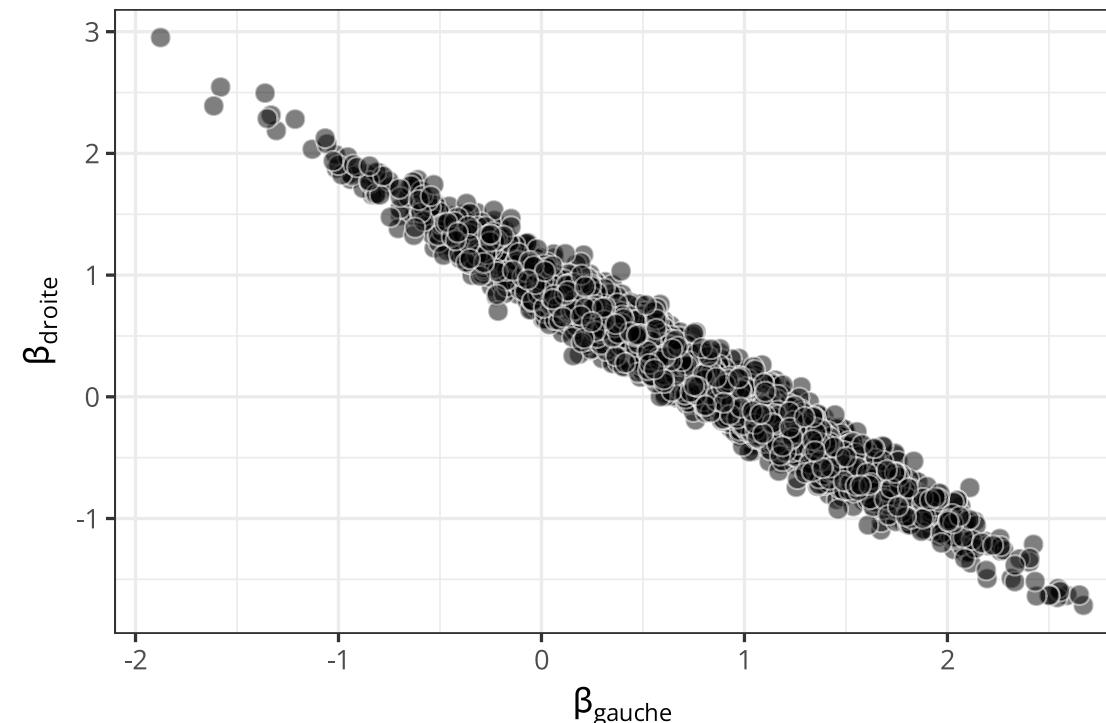
```
1 pairs(mod4, pars = parnames(mod4)[1:3])
```



Multicolinéarité

Comment traquer la colinéarité de deux prédicteurs ? On peut examiner la distribution postérieure de ces deux paramètres.

```
1 post <- as_draws_df(x = mod4)
2
3 post %>%
4   ggplot(aes(x = b_leg_left, y = b_leg_right) ) +
5   geom_point(pch = 21, size = 4, color = "white", fill = "black", alpha = 0.5) +
6   labs(x = expression(beta[gauche]), y = expression(beta[droite]) )
```



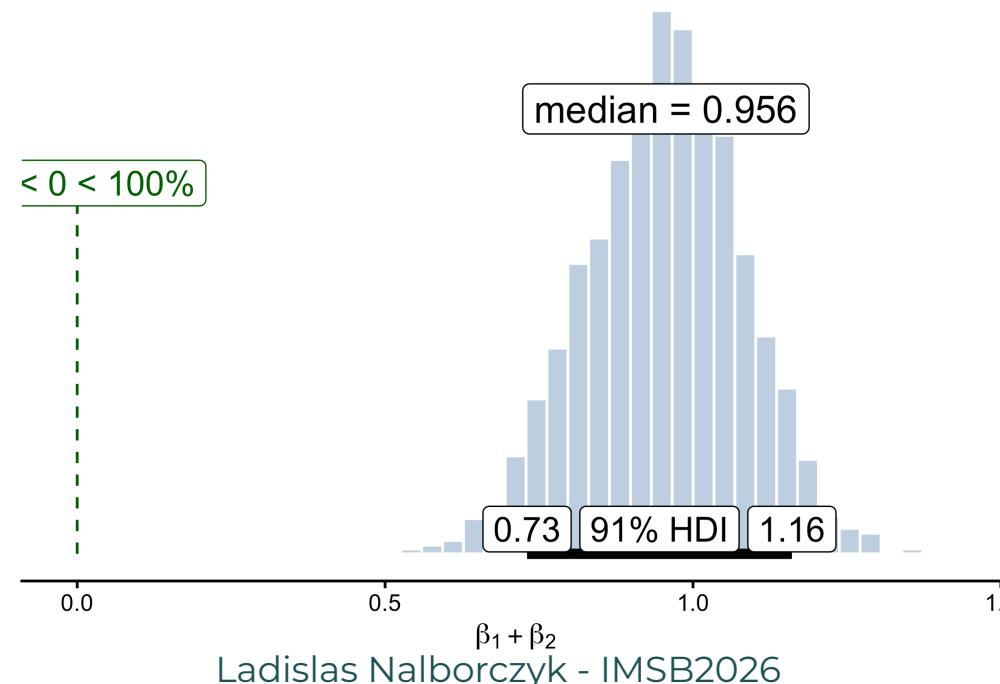
Multicolinéarité

Lorsqu'on inclut deux prédicteurs qui contiennent presque exactement la même information dans un modèle, c'est comme si on incluait deux fois le même prédicteur x . Du point de vue du modèle, les deux pentes ne sont pas dissociables, elles agissent sur le même prédicteur. C'est comme si on avait fitté le modèle suivant :

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + (\beta_1 + \beta_2)x_i$$

```
1 sum_legs <- post$b_leg_left + post$b_leg_right
2 posterior_plot(samples = sum_legs, compval = 0) + labs(x = expression(beta[1] + beta[2]))
```



Multicolinéarité

On crée un nouveau modèle avec seulement une jambe.

```
1 priors <- c(  
2   prior(normal(178, 10), class = Intercept),  
3   prior(normal(0, 10), class = b),  
4   prior(exponential(0.01), class = sigma)  
5 )  
6  
7 mod5 <- brm(  
8   formula = height ~ 1 + leg_left,  
9   prior = priors,  
10  family = gaussian,  
11  data = df2  
12 )
```

Régression multiple

En utilisant comme prédicteur une seule jambe, on retrouve l'estimation qui correspondait à la somme des deux pentes dans le modèle précédent.

```
1 posterior_summary(mod5)
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	101.3264415	10.149876199	80.7049504	120.739667
b_leg_left	0.9462465	0.125665105	0.7058043	1.201531
sigma	8.2387741	0.588088548	7.2130829	9.515633
Intercept	177.3540177	0.827425187	175.7612841	179.013804
lprior	-11.1406698	0.009671137	-11.1644623	-11.126230
lp__	-360.8778921	1.243216294	-364.0600633	-359.458427

Conclusion : Lorsque deux variables sont fortement corrélées (conditionnellement aux autres variables du modèle), les inclure toutes les deux dans un même modèle de régression peut produire des estimations aberrantes.

Post-treatment bias

Problèmes qui arrivent lorsqu'on inclut des prédicteurs qui sont eux-mêmes définis directement ou indirectement par d'autres prédicteurs inclus dans le modèle.

Supposons par exemple qu'on s'intéresse à la pousse des plantes en serre. On voudrait savoir quel traitement permettant de réduire la présence de champignons améliore la pousse des plantes.

On commence donc par planter et laisser germer des graines, mesurer la taille initiale des pousses, puis appliquer différents traitements.

Enfin, on mesure à la fin de l'expérience la taille finale de chaque plante et la présence de champignons.

Post-treatment bias

```

1 # nombre de plantes
2 N <- 100
3
4 # on simule différentes tailles à l'origine
5 h0 <- rnorm(n = N, mean = 10, sd = 2)
6
7 # on assigne différents traitements et on
8 # simule la présence de fungus et la pousse des plantes
9 treatment <- rep(x = 0:1, each = N / 2)
10 fungus <- rbinom(n = N, size = 1, prob = 0.5 - treatment * 0.4)
11 h1 <- h0 + rnorm(n = N, mean = 5 - 3 * fungus)
12
13 # on rassemble les données dans une dataframe
14 df3 <- data.frame(h0, h1, treatment, fungus)
15
16 # on affiche les six premières lignes
17 head(df3)

```

	h0	h1	treatment	fungus
1	8.842591	13.820383	0	0
2	5.094913	7.844256	0	1
3	9.423155	10.763637	0	1
4	13.008697	17.141846	0	0
5	11.566223	17.161368	0	0
6	9.520248	16.648277	0	0

Post-treatment bias

$$h_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_1 h0_i + \beta_2 T_i + \beta_3 F_i$$

$$\alpha \sim \text{Normal}(0, 10)$$

$$\beta_1, \beta_2, \beta_3 \sim \text{Normal}(0, 10)$$

$$\sigma \sim \text{Exponential}(0.01)$$

```

1 priors <- c(
2   prior(normal(0, 10), class = Intercept),
3   prior(normal(0, 10), class = b),
4   prior(exponential(0.01), class = sigma)
5 )
6
7 mod6 <- brm(
8   formula = h1 ~ 1 + h0 + treatment + fungus,
9   prior = priors,
10  family = gaussian,
11  data = df3
12 )
```

Post-treatment bias

On remarque que l'effet du traitement est négligeable. Mais la présence des champignons (`fungus`) est une conséquence de l'application du `treatment`. On demande au modèle si le traitement a une influence sachant que la plante a (ou n'a pas) développé de champignons...

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	4.32464727	0.48138580	3.3685557	5.2677963
b_h0	1.07372335	0.04385020	0.9868862	1.1618226
b_treatment	-0.08855845	0.19587086	-0.4670871	0.3017734
b_fungus	-2.64643541	0.22970916	-3.1147779	-2.1922780
sigma	0.91107578	0.06688833	0.7908050	1.0484024
Intercept	14.54789781	0.08999321	14.3680339	14.7241319
lprior	-18.59990947	0.01441497	-18.6289475	-18.5722350
lp__	-150.66059411	1.63298429	-154.6642151	-148.5555256

Post-treatment bias

Nous nous intéressons plutôt à l'influence du traitement sur la pousse. Il suffit de fitter un modèle sans la variable `fungus`. Remarque : il fait sens de prendre en compte h_0 , la taille initiale, car les différences de taille initiale pourraient masquer l'effet du traitement.

```
1 mod7 <- brm(  
2   formula = h1 ~ 1 + h0 + treatment,  
3   prior = priors,  
4   family = gaussian,  
5   data = df3  
6 )
```

Note : on pourrait également utiliser la méthode `update()`.

```
1 mod7 <- update(mod6, formula = h1 ~ 1 + h0 + treatment)
```

Post-treatment bias

```
1 summary(mod7)
```



Family: gaussian
 Links: mu = identity
 Formula: h1 ~ 1 + h0 + treatment
 Data: df3 (Number of observations: 100)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	2.25	0.71	0.85	3.63	1.00	4464	2828
h0	1.17	0.07	1.04	1.30	1.00	4104	2724
treatment	0.74	0.28	0.21	1.28	1.00	4650	3276

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	1.42	0.10	1.23	1.63	1.00	4397	2989

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).

L'influence du traitement est maintenant forte et positive.

Prédicteurs catégoriels

```
1 df4 <- open_data(howell)
2 str(df4)
```

```
'data.frame': 544 obs. of 4 variables:
 $ height: num 152 140 137 157 145 ...
 $ weight: num 47.8 36.5 31.9 53 41.3 ...
 $ age    : num 63 63 65 41 51 35 32 27 19 54 ...
 $ male   : int 1 0 0 1 0 1 0 1 0 1 ...
```

Le sexe est codé comme une **dummy variable**, c'est à dire une variable où chaque modalité est représentée soit par 0 soit par 1. On peut imaginer que cette nouvelle variable **active** le paramètre uniquement pour la catégorie codée 1, et le **désactive** pour la catégorie codée 0.

Prédicteurs catégoriels

$$h_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_m m_i$$

$$\alpha \sim \text{Normal}(178, 20)$$

$$\beta_m \sim \text{Normal}(0, 10)$$

$$\sigma \sim \text{Exponential}(0.01)$$

```

1 priors <- c(
2   prior(normal(178, 20), class = Intercept),
3   prior(normal(0, 10), class = b),
4   prior(exponential(0.01), class = sigma)
5 )
6
7 mod8 <- brm(
8   formula = height ~ 1 + male,
9   prior = priors,
10  family = gaussian,
11  data = df4
12 )
```



Prédicteurs catégoriels

L'intercept α représente la taille moyenne des femmes, car $\mu_i = \alpha + \beta_m(m_i = 0) = \alpha$.

```
1 fixef(mod8) # récupère les effets "fixes"
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	134.991927	1.592278	131.84965	138.00518
male	7.216882	2.262270	2.91364	11.65205

La pente β nous indique la différence de taille moyenne entre les hommes et les femmes. Pour obtenir la taille moyenne des hommes, il suffit donc d'ajouter α et β , car $\mu_i = \alpha + \beta_m(m_i = 1) = \alpha + \beta_m$.

```
1 post <- as_draws_df(x = mod8)
2 mu.male <- post$b_Intercept + post$b_male
3 quantile(x = mu.male, probs = c(0.025, 0.5, 0.975))
```

	2.5%	50%	97.5%
	138.9695	142.1934	145.4566

Prédicteurs catégoriels

Au lieu d'utiliser un paramètre pour la différence entre les deux catégories, on pourrait estimer un paramètre par catégorie...

$$\begin{aligned} h_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha_f(1 - m_i) + \alpha_h m_i \end{aligned}$$

Cette formulation est strictement équivalente à la précédente car :

$$\begin{aligned} \mu_i &= \alpha_f(1 - m_i) + \alpha_h m_i \\ &= \alpha_f + (\alpha_m - \alpha_f)m_i \end{aligned}$$

où $(\alpha_m - \alpha_f)$ est égal à la différence entre la moyenne des hommes et la moyenne des femmes (i.e., β_m).

Prédicteurs catégoriels

```
1 # on crée une nouvelle colonne pour les femmes
2 df4 <- df4 %>% mutate(female = 1 - male)
3
4 priors <- c(
5   # il n'y a plus d'intercept dans ce modèle
6   prior(normal(178, 20), class = b),
7   prior(exponential(0.01), class = sigma)
8 )
9
10 mod9 <- brm(
11   formula = height ~ 0 + female + male,
12   prior = priors,
13   family = gaussian,
14   data = df4
15 )
```

Prédicteurs catégoriels

```
1 summary(mod9)
```



Family: gaussian
 Links: mu = identity
 Formula: height ~ 0 + female + male
 Data: df4 (Number of observations: 544)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
female	134.92	1.65	131.70	138.09	1.00	3603	3039
male	142.57	1.68	139.28	145.89	1.00	4437	3300

Further Distributional Parameters:

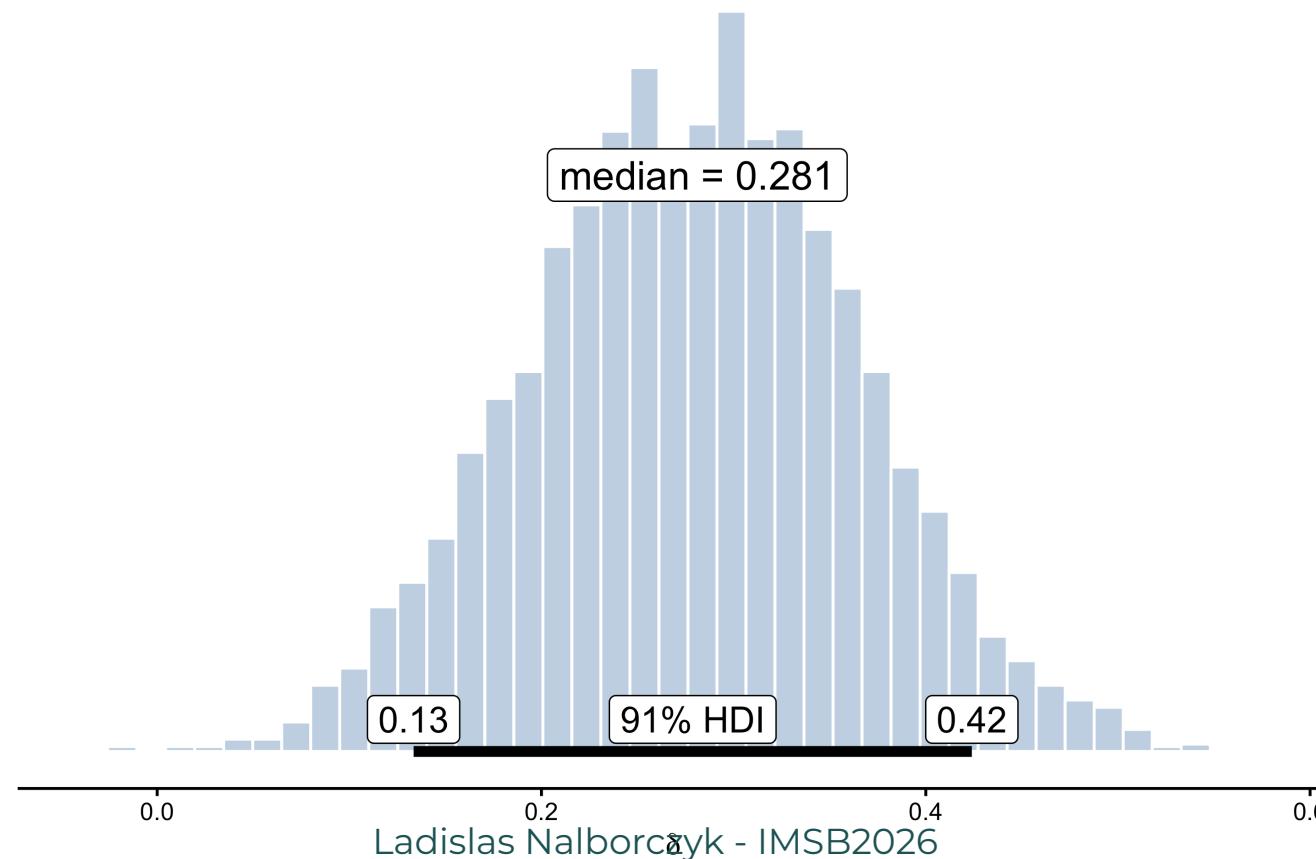
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	27.42	0.82	25.90	29.08	1.00	3990	2504

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).

Prédicteurs catégoriels

$$\text{Cohen's } d = \frac{\text{différence de moyennes}}{\text{écart-type}}$$

```
1 post <- as_draws_df(x = mod9)
2
3 posterior_plot(samples = (post$b_male - post$b_female) / post$sigma) +
4   labs(x = expression(delta))
```



Prédicteurs catégoriels

Nombre de catégories > 2.

```
1 df5 <- open_data(milk)
2 str(df5)
```

```
'data.frame': 29 obs. of 8 variables:
 $ clade      : chr "Strepsirrhine" "Strepsirrhine" "Strepsirrhine" "Strepsirrhine" ...
 $ species    : chr "Eulemur fulvus" "E macaco" "E mongoz" "E rubriventer" ...
 $ kcal.per.g : num 0.49 0.51 0.46 0.48 0.6 0.47 0.56 0.89 0.91 0.92 ...
 $ perc.fat   : num 16.6 19.3 14.1 14.9 27.3 ...
 $ perc.protein: num 15.4 16.9 16.9 13.2 19.5 ...
 $ perc.lactose: num 68 63.8 69 71.9 53.2 ...
 $ mass        : num 1.95 2.09 2.51 1.62 2.19 5.25 5.37 2.51 0.71 0.68 ...
 $ neocortex.perc: num 55.2 NA NA NA NA ...
```

Règle : pour k catégories, nous aurons besoin de $k - 1$ dummy variables. Pas la peine de créer une variable pour `ape`, qui sera notre intercept.

```
1 df5$clade.NWM <- ifelse(df5$clade == "New World Monkey", 1, 0)
2 df5$clade.OWM <- ifelse(df5$clade == "Old World Monkey", 1, 0)
3 df5$clade.S <- ifelse(df5$clade == "Strepsirrhine", 1, 0)
```

Prédicteurs catégoriels

$$k_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_{NWM} NWM_i + \beta_{OWM} OWM_i + \beta_S S_i$$

$$\alpha \sim \text{Normal}(0.6, 10)$$

$$\beta_{NWM}, \beta_{OWM}, \beta_S \sim \text{Normal}(0, 1)$$

$$\sigma \sim \text{Exponential}(0.01)$$

Category	NWM_i	OWM_i	S_i	μ_i
Ape	0	0	0	$\mu_i = \alpha$
New World monkey	1	0	0	$\mu_i = \alpha + \beta_{NWM}$
Old World monkey	0	1	0	$\mu_i = \alpha + \beta_{OWM}$
Strepsirrhine	0	0	1	$\mu_i = \alpha + \beta_S$

Prédicteurs catégoriels

```
1 priors <- c(  
2   prior(normal(0.6, 10), class = Intercept),  
3   prior(normal(0, 1), class = b),  
4   prior(exponential(0.01), class = sigma)  
5 )  
6  
7 mod10 <- brm(  
8   formula = kcal.per.g ~ 1 + clade.NWM + clade.OWM + clade.S,  
9   prior = priors,  
10  family = gaussian,  
11  data = df5  
12 )
```

Prédicteurs catégoriels

```
1 summary(mod10)
```



Family: gaussian
 Links: mu = identity
 Formula: kcal.per.g ~ 1 + clade.NWM + clade.OWM + clade.S
 Data: df5 (Number of observations: 29)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.55	0.04	0.46	0.63	1.00	3165	2881
clade.NWM	0.17	0.06	0.04	0.29	1.00	3325	3045
clade.OWM	0.24	0.07	0.10	0.38	1.00	3517	2792
clade.S	-0.04	0.07	-0.19	0.11	1.00	3067	2381

Further Distributional Parameters:

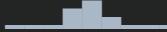
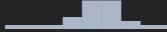
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.13	0.02	0.10	0.18	1.00	3493	2512

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).

Prédicteurs catégoriels

```
1 # récupère les échantillons de la distribution postérieure
2 post <- as_draws_df(x = mod10)
3
4 # récupère les échantillons pour chaque clade
5 mu.ape <- post$b_Intercept
6 mu.NWM <- post$b_Intercept + post$b_clade.NWM
7 mu.OWM <- post$b_Intercept + post$b_clade.OWM
8 mu.S <- post$b_Intercept + post$b_clade.S
```

```
1 # résumé de ces échantillons par clade
2 rethinking::precis(data.frame(mu.ape, mu.NWM, mu.OWM, mu.S), prob = 0.95)
```

	mean	sd	2.5%	97.5%	histogram
mu.ape	0.5450298	0.04386319	0.4596593	0.6327154	
mu.NWM	0.7145975	0.04439188	0.6252298	0.8043641	
mu.OWM	0.7883000	0.05549405	0.6762624	0.8981664	
mu.S	0.5078894	0.05901674	0.3900350	0.6238376	

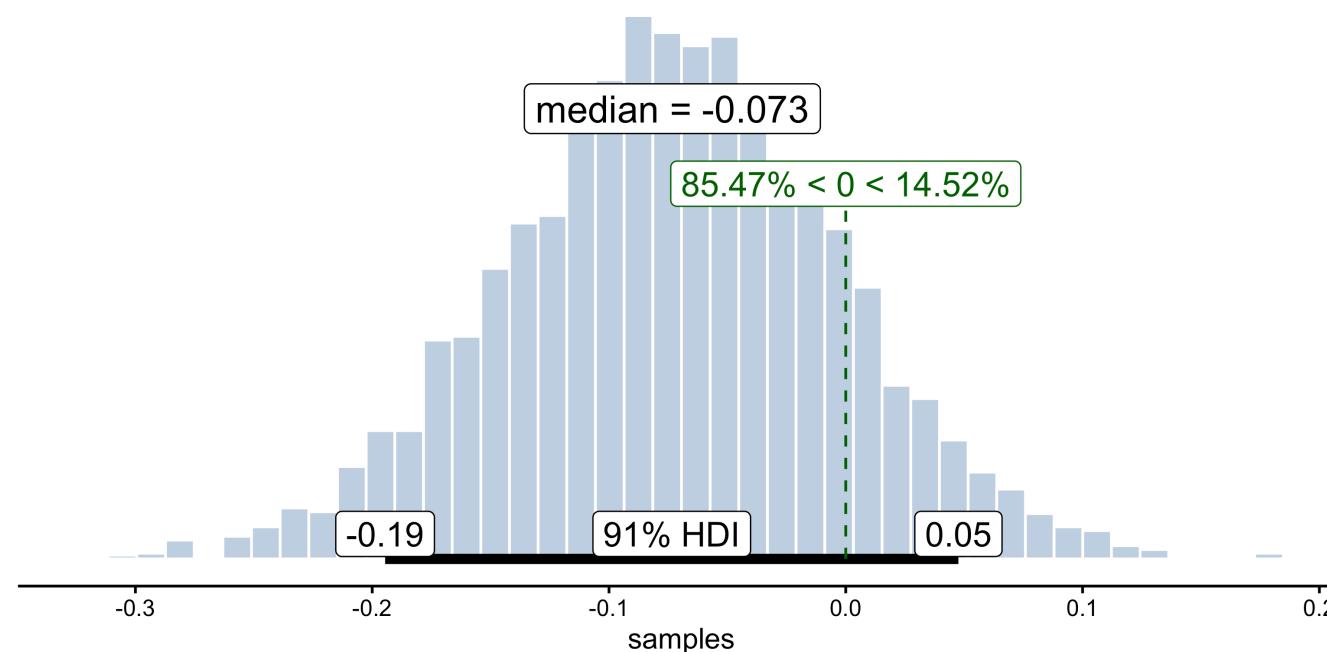
Prédicteurs catégoriels

Si on s'intéresse à la différence entre deux groupes, on peut calculer la distribution postérieure de cette différence.

```
1 diff.NWM.OWM <- mu.NWM - mu.OWM
2 quantile(diff.NWM.OWM, probs = c(0.025, 0.5, 0.975) )
```

2.5%	50%	97.5%
-0.21389086	-0.07324267	0.06576515

```
1 posterior_plot(samples = diff.NWM.OWM, compval = 0)
```



Prédicteurs catégoriels

Une autre manière de considérer les variables catégorielles consiste à construire un vecteur d'intercepts, avec un intercept par catégorie.

$$k_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha_{\text{clade}[i]}$$

$$\alpha_{\text{clade}[i]} \sim \text{Normal}(0.6, 10)$$

$$\sigma \sim \text{Exponential}(0.01)$$

Prédicteurs catégoriels

Comme on a vu avec l'exemple du sexe, `brms` “comprend” automatiquement que c'est ce qu'on veut faire lorsqu'on fit un modèle sans intercept et avec un prédicteur catégoriel (codé en facteur).

```
1 priors <- c(  
2   prior(normal(0.6, 10), class = b),  
3   prior(exponential(0.01), class = sigma)  
4 )  
5  
6 mod11 <- brm(  
7   # modèle sans intercept avec seulement un prédicteur catégoriel (facteur)  
8   formula = kcal.per.g ~ 0 + clade,  
9   prior = priors,  
10  family = gaussian,  
11  data = df5  
12 )
```

Prédicteurs catégoriels

```
1 summary(mod11)
```



Family: gaussian
 Links: mu = identity
 Formula: kcal.per.g ~ 0 + clade
 Data: df5 (Number of observations: 29)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
cladeApe	0.55	0.04	0.46	0.63	1.00	5183	2709
cladeNewWorldMonkey	0.72	0.04	0.63	0.80	1.00	5334	2839
cladeOldWorldMonkey	0.79	0.05	0.69	0.89	1.00	5565	3281
cladeStrepsirrhine	0.51	0.06	0.39	0.63	1.00	5278	2813

Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.13	0.02	0.10	0.17	1.00	3772	3013

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).

Interaction

Jusque maintenant, les prédicteurs du modèle entretenaient des relations mutuellement indépendantes. Et si nous souhaitions que ces relations soient **conditionnelles**, ou **dépendantes** les unes des autres ?

Par exemple : on s'intéresse à la pousse des tulipes selon la quantité de lumière reçue et l'humidité du sol. Il se pourrait que la relation entre quantité de lumière reçue et pousse des tulipes soit différente selon l'humidité du sol. En d'autres termes, il se pourrait que la relation entre quantité de lumière reçue et pousse des tulipes soit **conditionnelle** à l'humidité du sol...

Interaction

```
1 df6 <- open_data(tulips)
2 head(df6, 10)
```

	bed	water	shade	blooms
1	a	1	1	0.00
2	a	1	2	0.00
3	a	1	3	111.04
4	a	2	1	183.47
5	a	2	2	59.16
6	a	2	3	76.75
7	a	3	1	224.97
8	a	3	2	83.77
9	a	3	3	134.95
10	b	1	1	80.10

Interaction

Modèle sans interaction :

$$\begin{aligned}B_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha + \beta_W W_i + \beta_S S_i\end{aligned}$$

Modèle avec interaction :

$$\begin{aligned}B_i &\sim \text{Normal}(\mu_i, \sigma) \\ \mu_i &= \alpha + \beta_W W_i + \beta_S S_i + \beta_{WS} W_i S_i\end{aligned}$$

On centre les prédicteurs (pour faciliter l'interprétation des paramètres).

```
1 df6$shade.c <- df6$shade - mean(df6$shade)
2 df6$water.c <- df6$water - mean(df6$water)
```



Interaction

```
1 priors <- c(  
2   prior(normal(130, 100), class = Intercept),  
3   prior(normal(0, 100), class = b),  
4   prior(exponential(0.01), class = sigma)  
5 )  
6  
7 mod12 <- brm(  
8   formula = blooms ~ 1 + water.c + shade.c,  
9   prior = priors,  
10  family = gaussian,  
11  data = df6  
12 )
```

```
1 mod13 <- brm(  
2   formula = blooms ~ 1 + water.c * shade.c,  
3   # équivalent à blooms ~ 1 + water.c + shade.c + water.c:shade.c  
4   prior = priors,  
5   family = gaussian,  
6   data = df6  
7 )
```

Interaction

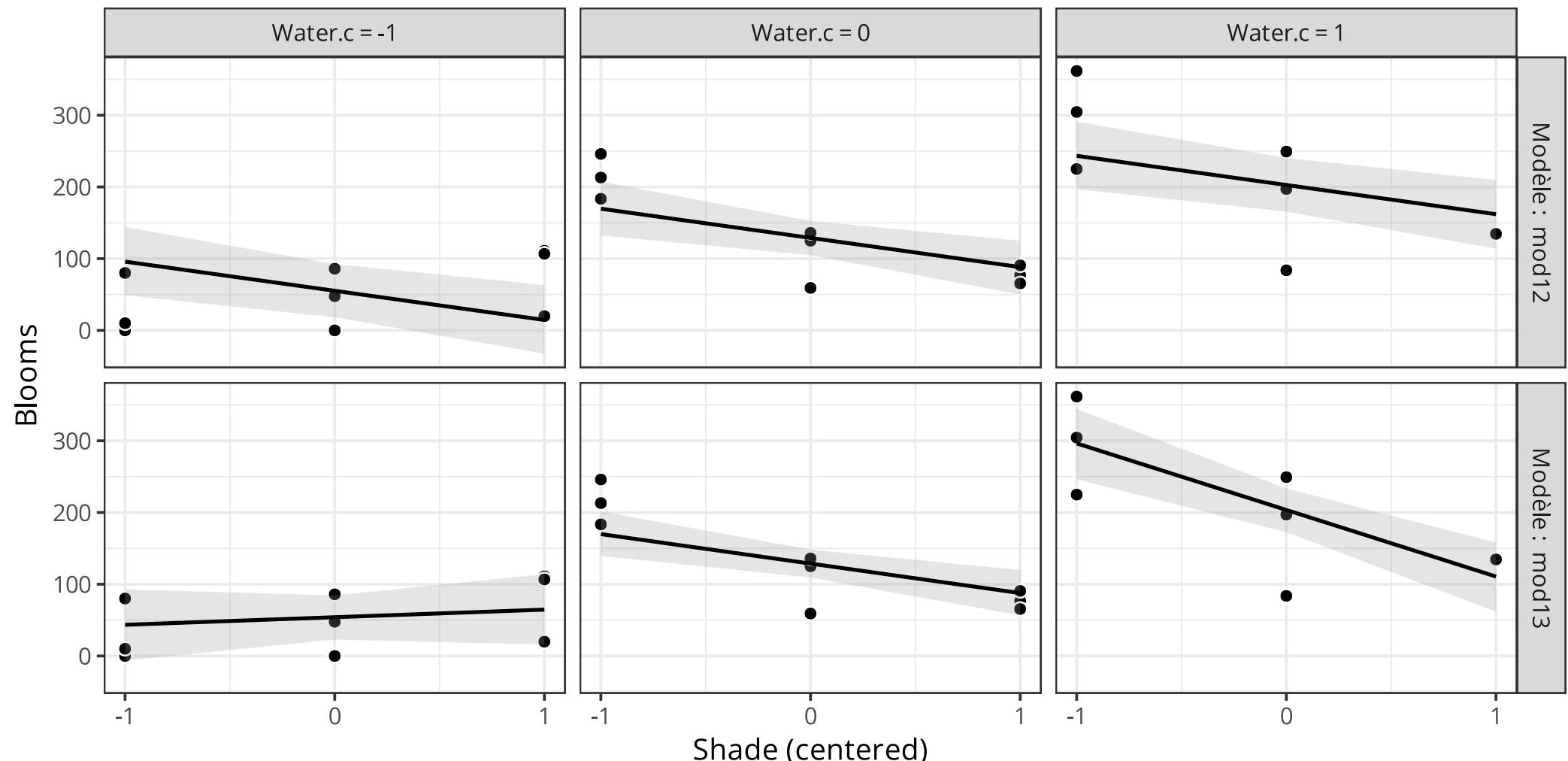
On compare les estimations des deux modèles.

	term	mod12	mod13
1	b_Intercept	128.94676	128.79906
2	b_water.c	73.75194	74.76939
3	b_shade.c	-40.68546	-41.12131
4	sigma	63.24143	51.18229
5	Intercept	128.94676	128.79906
6	lprior	-22.19385	-27.74216
7	b_water.c:shade.c	NA	-51.71338

- L'intercept α représente la valeur attendue de `blooms` quand `water` et `shade` sont à 0 (i.e., la moyenne générale de la variable dépendante).
- La pente β_W nous donne la valeur attendue de changement de `blooms` quand `water` augmente d'une unité et `shade` est à sa valeur moyenne. On voit qu'augmenter la quantité d'eau est très bénéfique.
- La pente β_S nous donne la valeur attendue de changement de `blooms` quand `shade` augmente d'une unité et `water` est à sa valeur moyenne. On voit qu'augmenter la "quantité d'ombre" (diminuer l'exposition à la lumière) est plutôt délétère.
- La pente β_{WS} nous renseigne sur l'effet attendu de `water` sur `blooms` quand `shade` augmente d'une unité (et réciproquement).

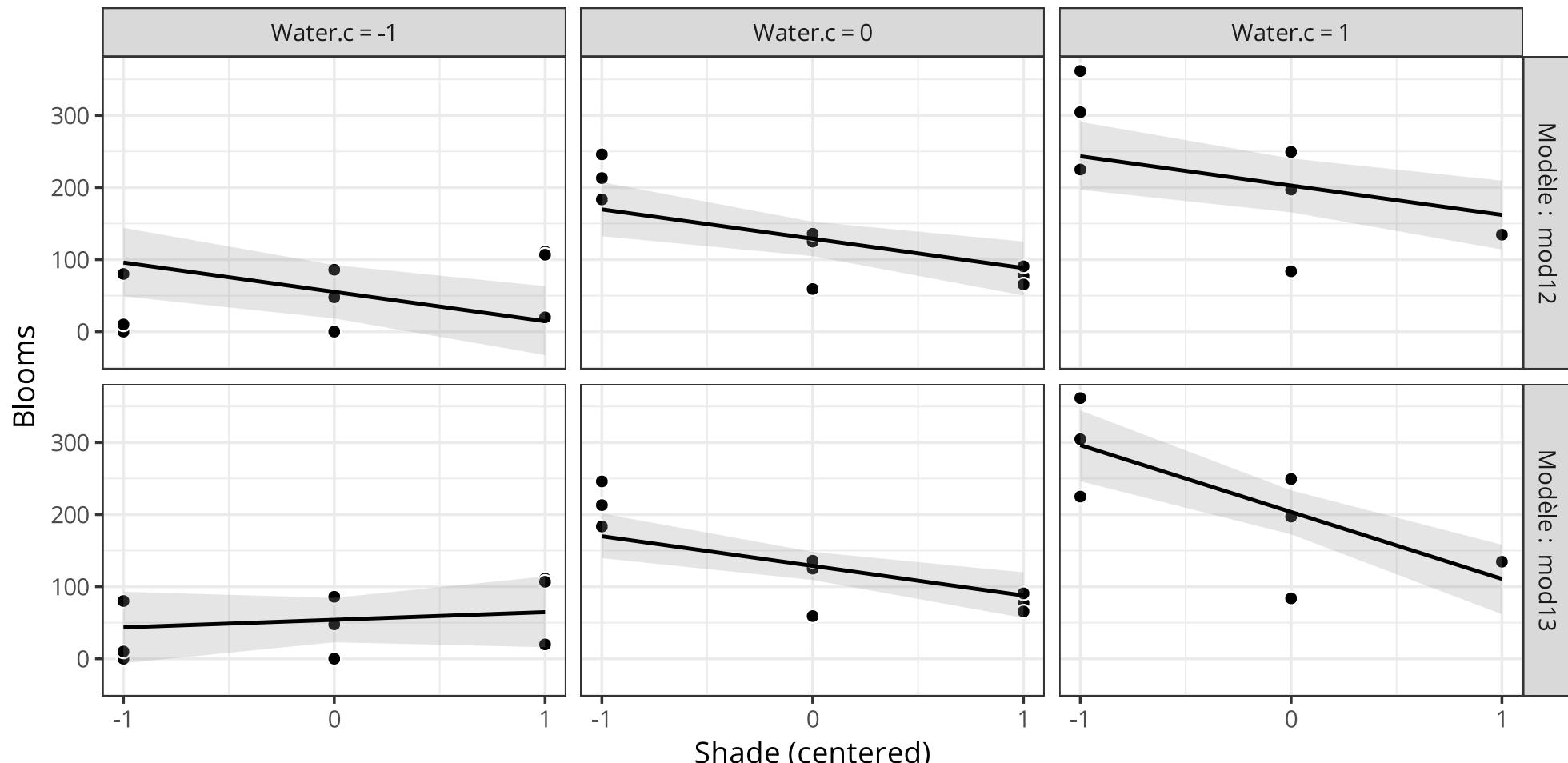
Interaction

Dans un modèle qui inclut un effet d'interaction, l'effet d'un prédicteur sur la mesure va dépendre de la valeur de l'autre prédicteur. La meilleure manière de représenter cette dépendance est de représenter visuellement la relation entre un prédicteur et la mesure, à différentes valeurs de l'autre prédicteur.



Interaction

L'effet d'interaction nous indique que les tulipes ont besoin à la fois d'eau et de lumière pour pousser, mais aussi qu'à de faibles niveaux d'humidité, la luminosité a peu d'effet, tandis que cet effet est plus important à haut niveau d'humidité. Cette explication vaut de manière **symétrique** pour l'effet de l'humidité sur la relation entre luminosité et pousse des plantes.



Résumé du cours

Nous avons étendu le modèle de régression à plusieurs prédicteurs. Ce modèle de régression multiple permet de distinguer les influences causales de différents prédicteurs, lorsque les prédicteurs sont inclus (ou pas) dans le modèle, en considérant la structure causale sous-jacente.

Nous avons également étendu le modèle de régression aux prédicteurs catégoriels, et introduit le concept d'interaction entre différents prédicteurs.

Plus nous ajoutons de variables dans notre modèle, plus les estimations “brutes” (numériques) sont difficiles à interpréter. Il devient donc plus simple, pour comprendre les prédictions du modèle, de les représenter graphiquement. Nous avons également souligné l’importance des prior et posterior predictive checks dans ce contexte.

Comme précédemment, le théorème de Bayes est utilisé pour mettre à jour nos connaissances a priori quant à la valeur des paramètres en une connaissance a posteriori, synthèse entre nos priors et l’information contenue dans les données.

Exercice #1

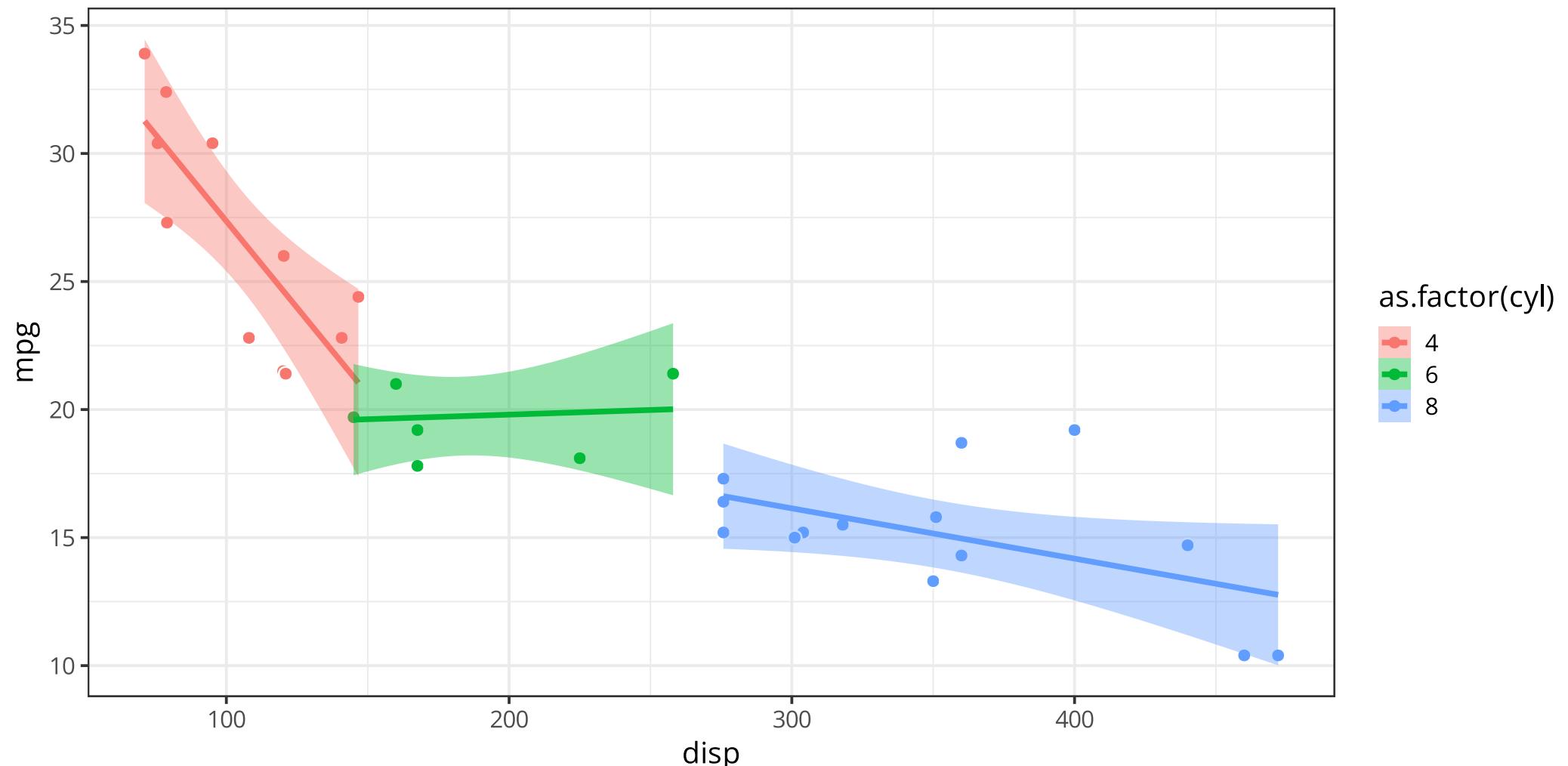
Cet exemple est basé sur le jeu de données `mtcars`, issu du volume de 1974 de “Motor Trend US”. La mesure qui nous intéresse est la consommation de carburant, en miles per gallon (mpg).

```
1 data(mtcars)
2 head(mtcars, 10)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Exercice #1

Imaginons que nous souhaitions savoir comment la cylindrée affecte la relation entre le nombre de cylindres et la consommation de carburant et / ou comment le nombre de cylindres affecte la relation entre la cylindrée et la consommation de carburant. Ce genre d'effet appelle une analyse d'interaction.



Exercice #1

```
1 mtcars$disp.s <- as.numeric(scale(mtcars$disp) )
2 mtcars$cyl.s <- as.numeric(scale(mtcars$cyl) )
3
4 m_cyl <- lm(mpg ~ disp.s * cyl.s, data = mtcars)
5 summary(m_cyl)
```

Call:

`lm(formula = mpg ~ disp.s * cyl.s, data = mtcars)`

Residuals:

Min	1Q	Median	3Q	Max
-4.0809	-1.6054	-0.2948	1.0546	5.7981

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	17.0242	1.0663	15.966	1.36e-15 ***
disp.s	-5.8784	1.5176	-3.873	0.000589 ***
cyl.s	0.4511	1.5088	0.299	0.767156
disp.s:cyl.s	3.5092	1.0952	3.204	0.003369 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.66 on 28 degrees of freedom

Multiple R-squared: 0.8241, Adjusted R-squared: 0.8052

F-statistic: 43.72 on 3 and 28 DF, p-value: 1.078e-10

Proposition de réponse

```
1 priors <- c(  
2   prior(normal(0, 100), class = Intercept),  
3   prior(normal(0, 10), class = b),  
4   prior(exponential(0.1), class = sigma)  
5 )  
6  
7 mod14 <- brm(  
8   formula = mpg ~ 1 + disp.s * cyl.s,  
9   prior = priors,  
10  family = gaussian,  
11  data = mtcars  
12 )
```

Proposition de réponse

```
1 summary(mod14)
```



Family: gaussian
 Links: mu = identity
 Formula: mpg ~ 1 + disp.s * cyl.s
 Data: mtcars (Number of observations: 32)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	17.12	1.10	14.92	19.25	1.00	2126	2448
disp.s	-5.67	1.51	-8.66	-2.75	1.00	1513	2103
cyl.s	0.25	1.51	-2.71	3.31	1.00	1511	1936
disp.s:cyl.s	3.39	1.11	1.27	5.58	1.00	1909	2261

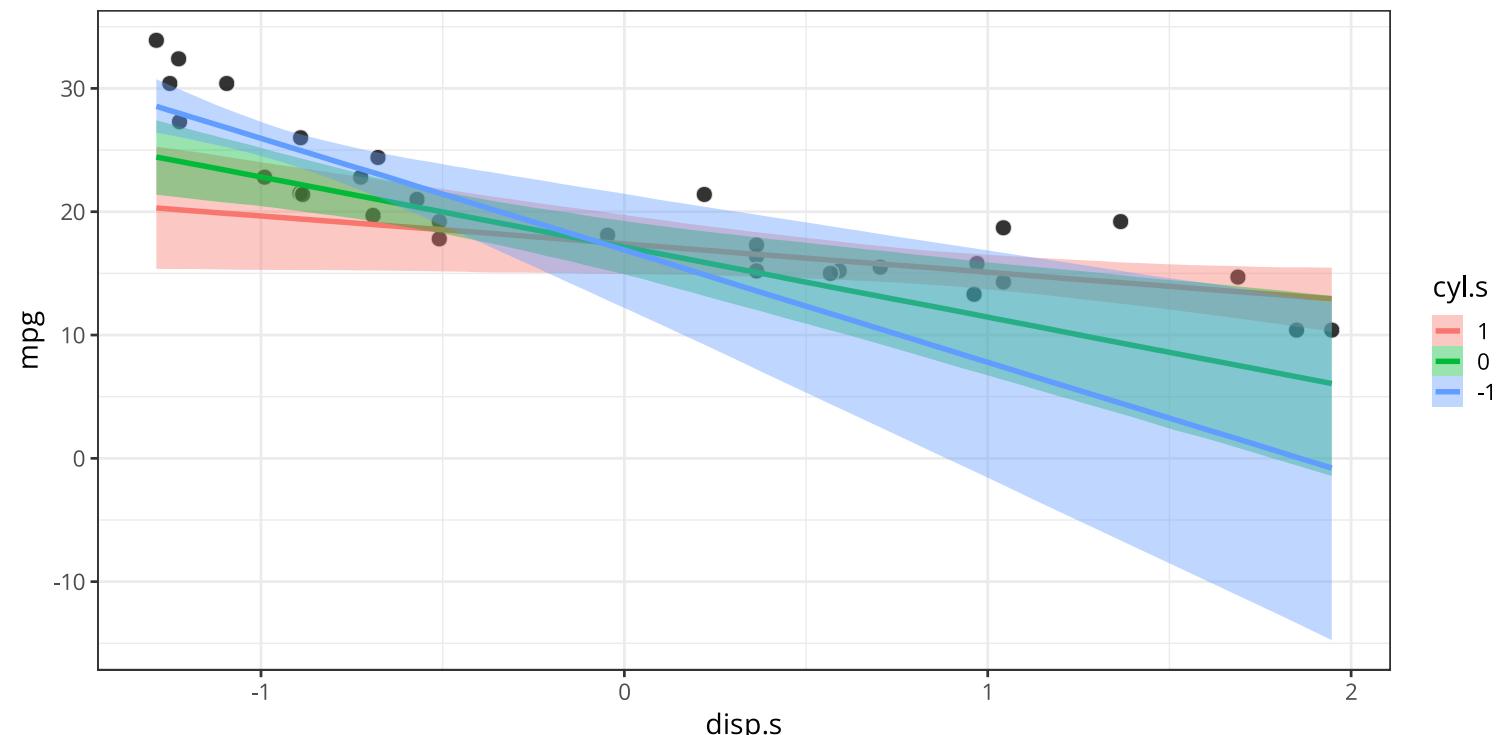
Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	2.76	0.39	2.12	3.63	1.00	2518	2357

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).

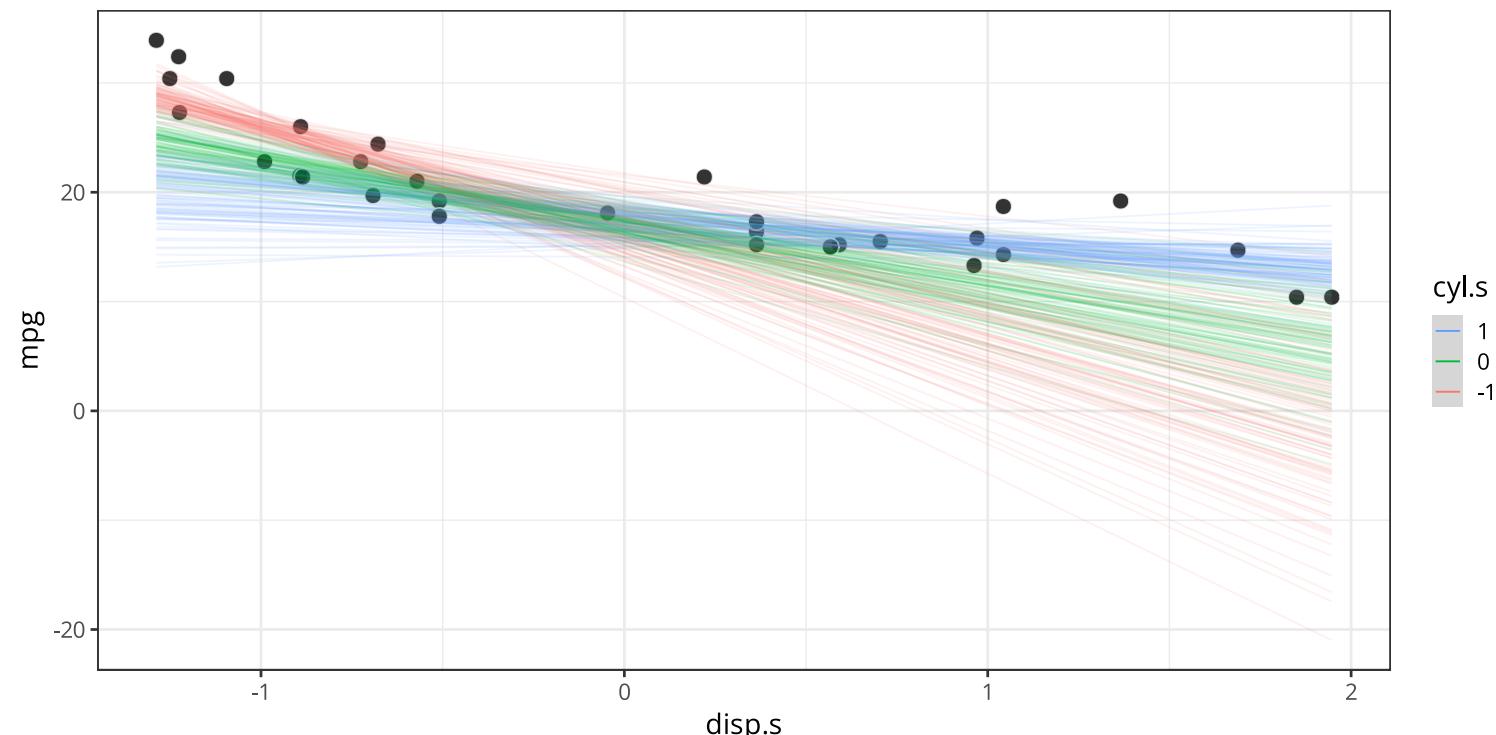
Proposition de réponse

```
1 plot(
2   conditional_effects(x = mod14, effects = "disp.s:cyl.s"),
3   points = TRUE,
4   point_args = list(
5     alpha = 0.8, shape = 21, size = 4,
6     color = "white", fill = "black"
7   ),
8   theme = theme_bw(base_size = 16, base_family = "Open Sans")
9 )
```



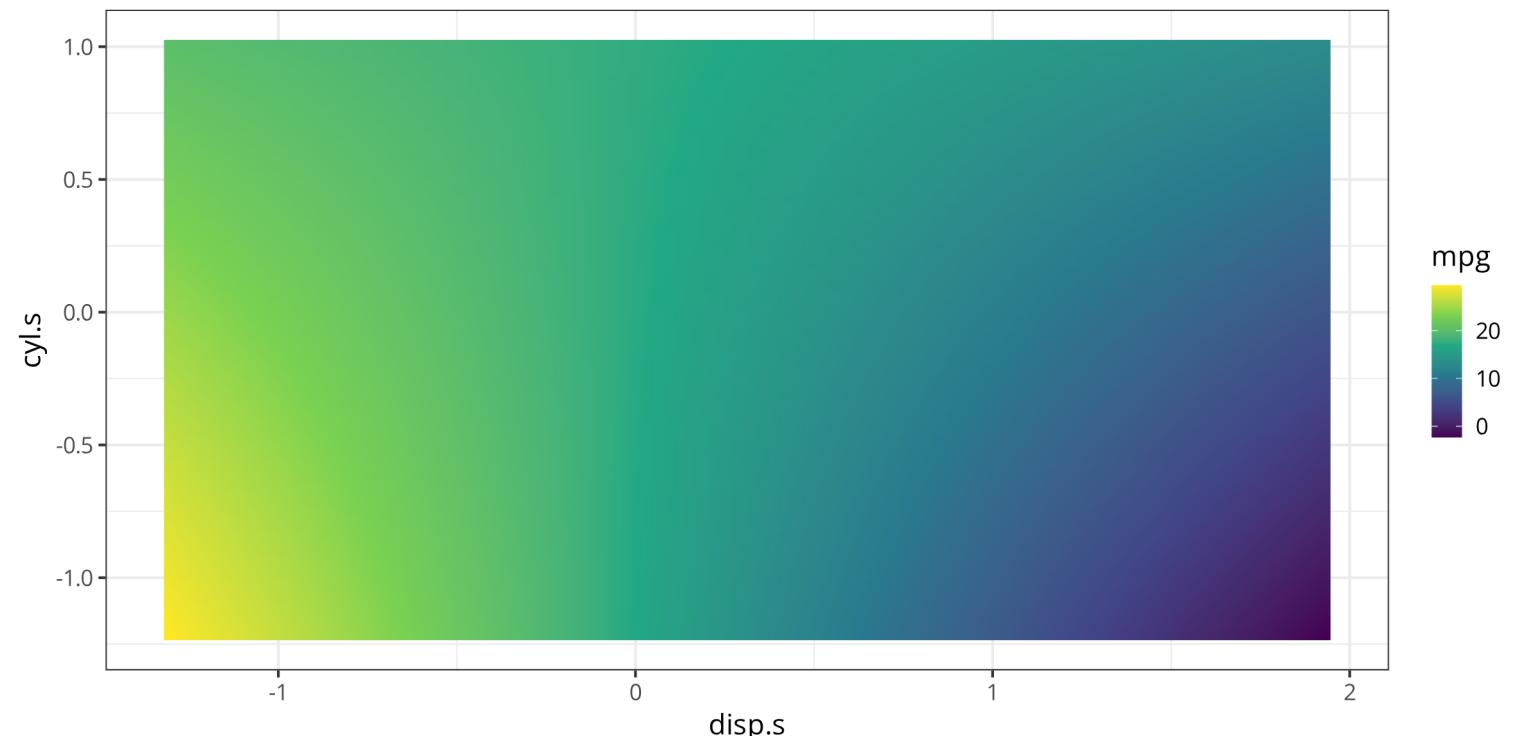
Proposition de réponse

```
1 plot(  
2   conditional_effects(x = mod14, effects = "disp.s:cyl.s", spaghetti = TRUE, ndraws = 1e2),  
3   points = TRUE, mean = FALSE,  
4   point_args = list(  
5     alpha = 0.8, shape = 21, size = 4,  
6     color = "white", fill = "black"  
7   ),  
8   theme = theme_bw(base_size = 16, base_family = "Open Sans")  
9 )
```



Proposition de réponse

```
1 plot(
2   conditional_effects(
3     x = mod14, effects = "disp.s:cyl.s",
4     surface = TRUE, resolution = 1e2
5   ),
6   stype = "raster", # contour or raster
7   surface_args = list(hjust = 0),
8   theme = theme_bw(base_size = 16, base_family = "Open Sans")
9 )
```



Exercice #2

Le jeu de données `airquality` recense des mesures de la qualité de l'air réalisées à New York, de Mai à Septembre 1973.

```
1 data(airquality)
2 df7 <- airquality[complete.cases(airquality), ] # removes NAs
3
4 head(df7, 10)
```

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8
9	8	19	20.1	61	5	9
12	16	256	9.7	69	5	12
13	11	290	9.2	66	5	13
14	14	274	10.9	68	5	14

Exercice #2

On s'intéresse à la concentration de l'air en Ozone en fonction de la force du vent et de la température.

- Écrire le modèle mathématique.
- Fitter ce modèle avec `brms::brm()`, interpréter les estimations du modèle, et conclure sur l'effet de la force du vent et de la température.
- Évaluer le modèle en faisant du **posterior predictive checking**.

Utilisez les fonctions suivantes (et lisez la documentation !) :

- `brms::brm()` : permet de construire le modèle
- `summary()` : affiche les estimations du modèle
- `brms::pp_check()` : posterior predictive checking

Proposition de réponse, modèle mathématique

On construit un modèle de régression multiple avec un intercept α et deux prédicteurs : la force du vent W et la température T , pour prédire la concentration de l'air en Ozone O .

$$O_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta_W W_i + \beta_T T_i$$

$$\alpha \sim \text{Normal}(50, 10)$$

$$\beta_W, \beta_T \sim \text{Normal}(0, 10)$$

$$\sigma \sim \text{Exponential}(0.01)$$

Proposition de réponse, fitter le modèle

```
1 df7$Wind.s <- scale(df7$Wind)
2 df7$Temp.s <- scale(df7$Temp)
3
4 priors <- c(
5   prior(normal(50, 10), class = Intercept),
6   prior(normal(0, 10), class = b),
7   prior(exponential(0.01), class = sigma)
8 )
9
10 mod15 <- brm(
11   formula = Ozone ~ 1 + Wind.s + Temp.s,
12   prior = priors,
13   family = gaussian,
14   data = df7
15 )
```

Proposition de réponse, estimations du modèle

```
1 summary(mod15)
```



Family: gaussian
 Links: mu = identity
 Formula: Ozone ~ 1 + Wind.s + Temp.s
 Data: df7 (Number of observations: 111)
 Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup draws = 4000

Regression Coefficients:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	42.41	2.06	38.42	46.41	1.00	3745	2592
Wind.s	-11.56	2.31	-15.97	-6.98	1.00	3700	3149
Temp.s	16.74	2.33	12.27	21.44	1.00	3757	3115

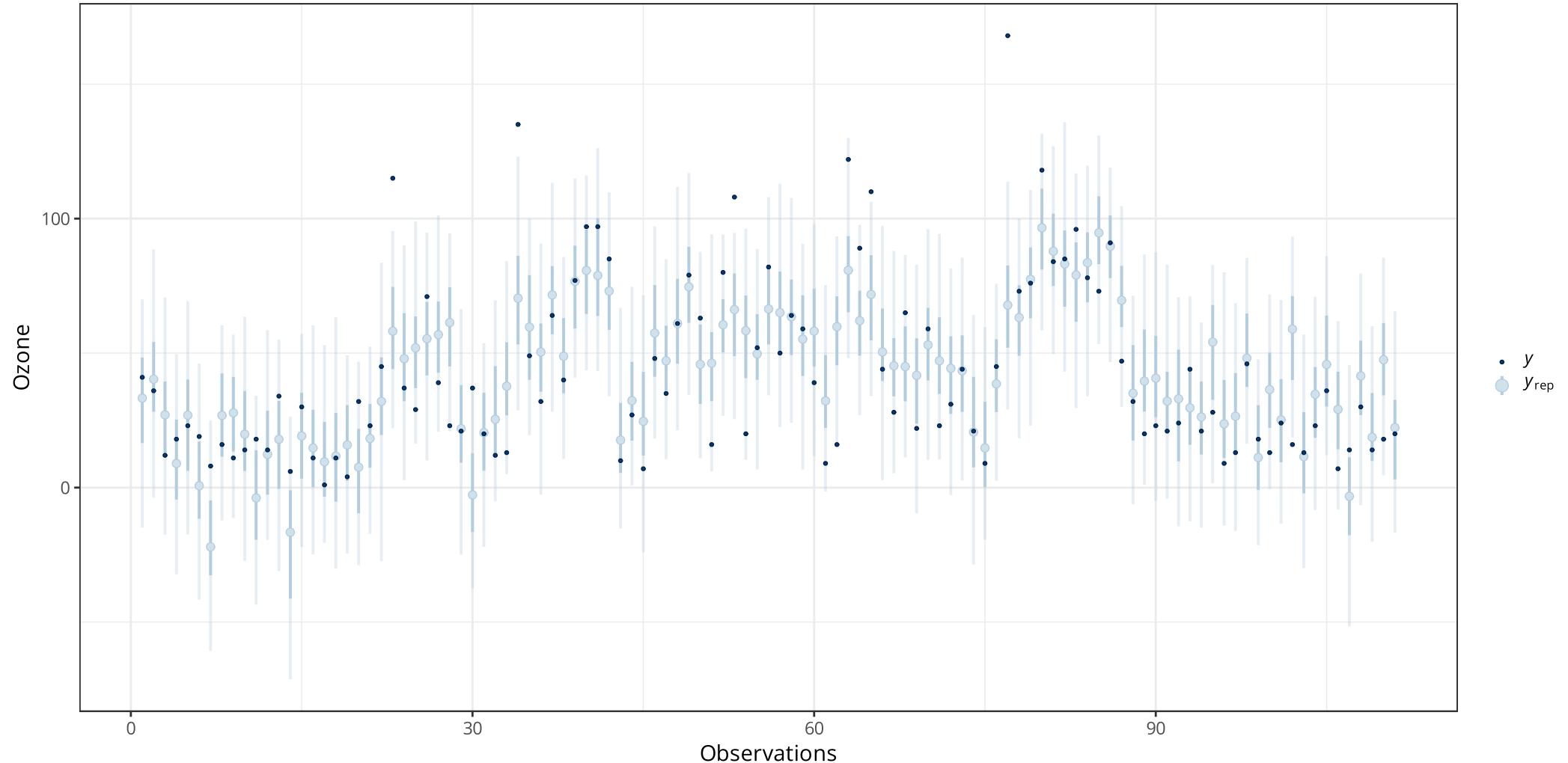
Further Distributional Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	21.94	1.49	19.28	25.01	1.00	3429	2824

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).

Proposition de réponse, posterior predictive checking

```
1 pp_check(mod15, type = "intervals", ndraws = 1e2, prob = 0.5, prob_outer = 0.95) +  
2   labs(x = "Observations", y = "Ozone")
```



Proposition de réponse, posterior predictive checking

```
1 pp_check(object = mod15, ndraws = 1e2) + labs(x = "Ozone", y = "Density")
```

