

# Introduction à la modélisation statistique bayésienne

Ladislav 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

# Rappels

On considère un modèle de régression linéaire gaussien avec un prédicteur continu. Ce modèle a trois paramètres à estimer : l'intercept  $\alpha$ , la pente  $\beta$ , et l'écart-type des "résidus"  $\sigma$ .

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

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

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

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

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

# Rappels

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

```
library(brms)

priors <- c(
  prior(normal(100, 10), class = Intercept),
  prior(normal(0, 10), class = b),
  prior(exponential(0.01), class = sigma)
)

model <- brm(
  y ~ 1 + x,
  family = gaussian(),
  prior = priors,
  data = df
)
```

# Régression multiple

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

- *Contrôle* des facteurs de confusion (e.g., [spurious correlations](#), [simpson's paradox](#)). Un facteur de confusion est une variable aléatoire qui influence à la fois la variable dépendante et les variables explicatives.
- 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

```
library(rethinking)
library(tidyverse)

data(WaffleDivorce) # import des données
df1 <- WaffleDivorce # import dans une dataframe nommée df1

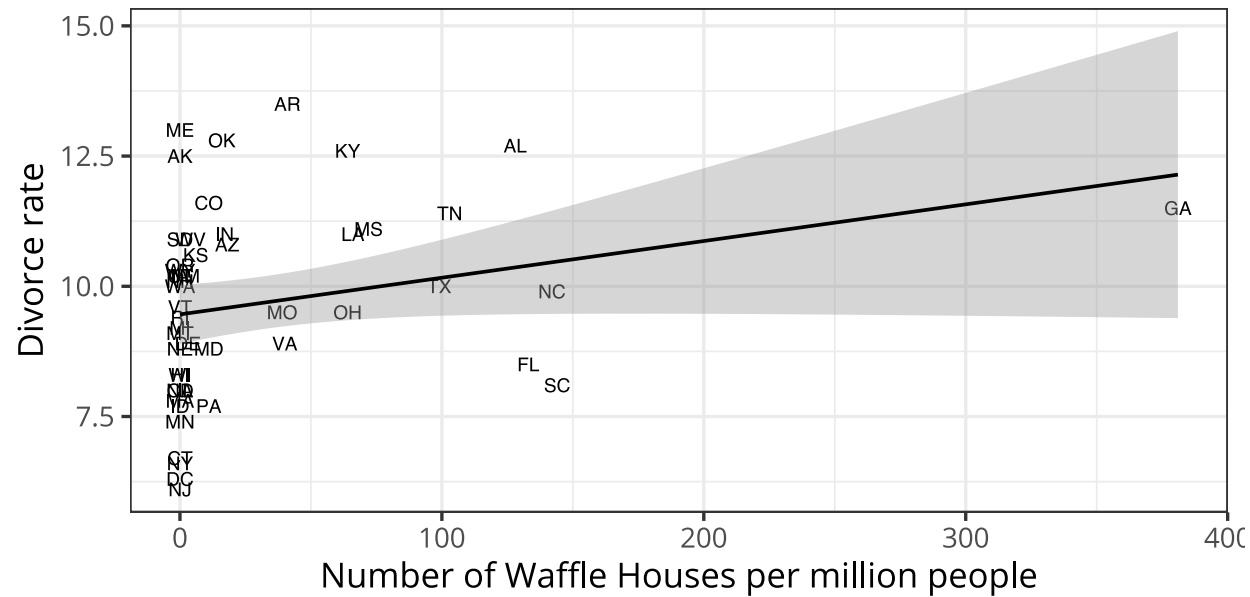
str(df1) # affiche la structure des données
```

```
'data.frame':  50 obs. of  13 variables:
 $ Location      : Factor w/ 50 levels "Alabama","Alaska",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Loc           : Factor w/ 50 levels "AK","AL","AR",...: 2 1 4 3 5 6 7 9 8 10 ...
 $ 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 0.016 0 0.44 ...
```

# Associations fortuites

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

```
df1 %>%  
  ggplot(aes(x = WaffleHouses, y = Divorce)) +  
  geom_text(aes(label = Loc)) +  
  geom_smooth(method = "lm", color = "black", se = TRUE) +  
  labs(x = "Number of Waffle Houses per million people", y = "Divorce rate")
```

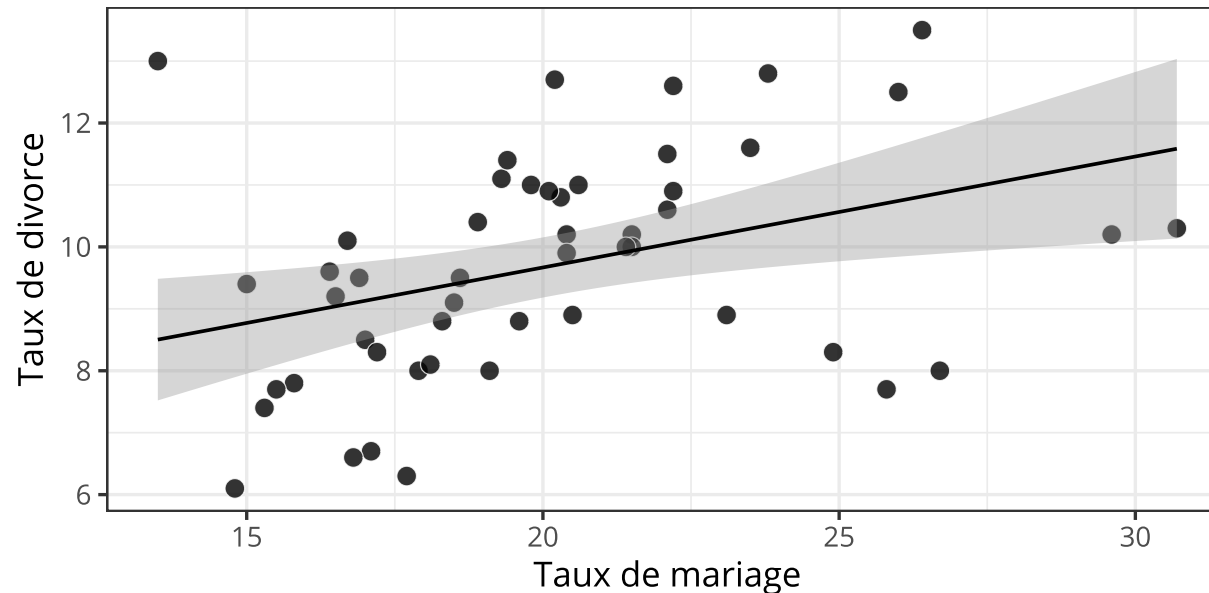


# Associations fortuites

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 ?

```
df1$Divorce.s <- (df1$Divorce - mean(df1$Divorce) ) / sd(df1$Divorce)
df1$Marriage.s <- (df1$Marriage - mean(df1$Marriage) ) / sd(df1$Marriage)

df1 %>%
  ggplot(aes(x = Marriage, y = Divorce) ) +
  geom_point(pch = 21, color = "white", fill = "black", size = 5, alpha = 0.8) +
  geom_smooth(method = "lm", color = "black", se = TRUE) +
  labs(x = "Taux de mariage", y = "Taux de divorce")
```

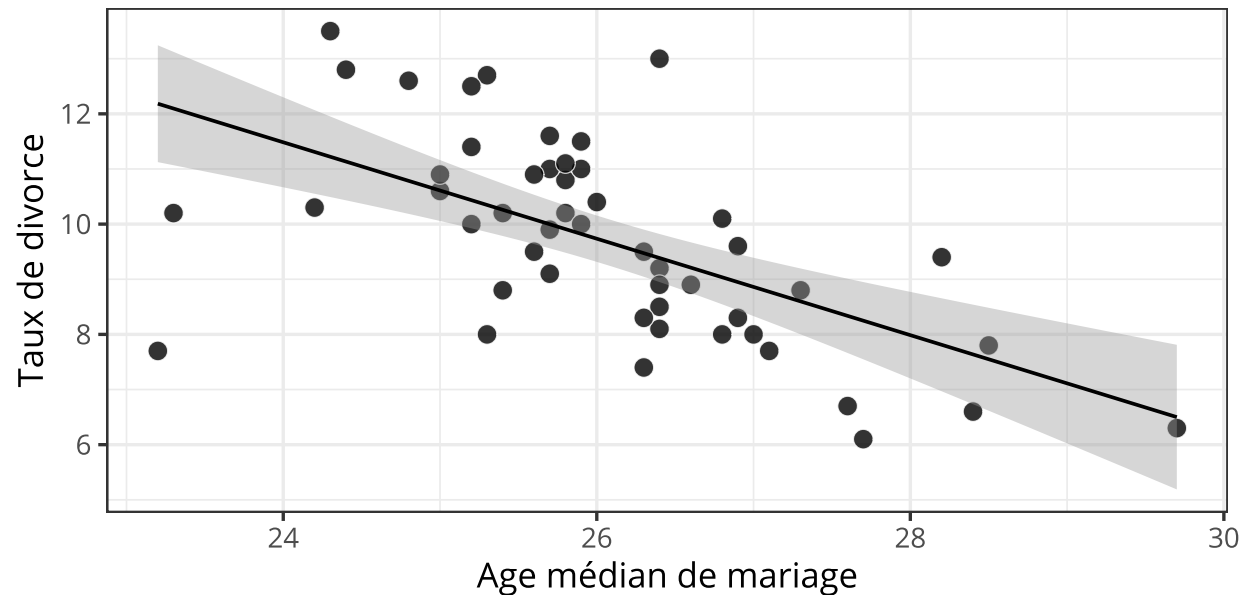




# Associations fortuites

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

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



# Influence du taux de mariage

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

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

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

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

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

```
priors <- c(  
  prior(normal(0, 10), class = Intercept),  
  prior(normal(0, 1), class = b),  
  prior(exponential(1), class = sigma)  
)  
  
mod1 <- brm(  
  Divorce.s ~ 1 + Marriage.s,  
  family = gaussian(),  
  prior = priors,  
  # for prior predictive checking  
  sample_prior = TRUE,  
  data = df1  
)
```

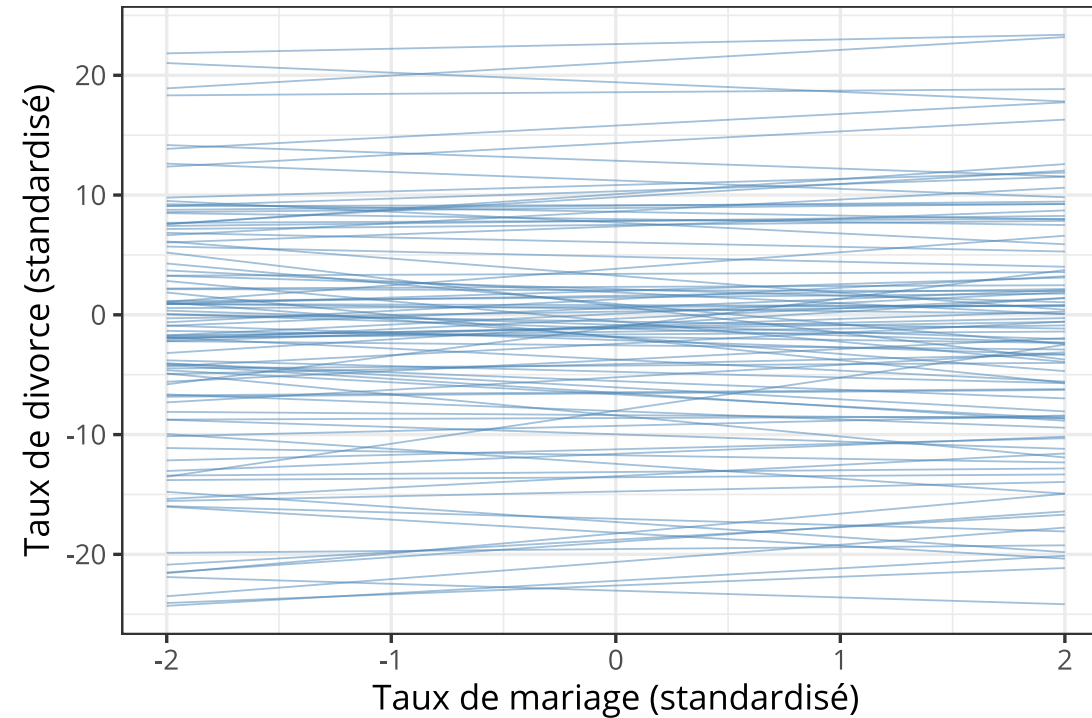
# Prior predictive checking

```
# getting the samples from the prior distribution  
prior <- prior_samples(mod1)  
  
# displaying the first six samples  
head(prior)
```

	Intercept	b	sigma
1	-18.9598991	0.9492903	1.56781849
2	-5.1251362	1.3310229	0.02805471
3	8.8314382	-0.2324045	1.77969172
4	-0.2002377	-0.8235907	0.35162553
5	4.2618860	-1.4401472	0.75244521
6	3.2694662	0.6230406	1.65426123

# Prior predictions

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



# Influence du taux de mariage

```
summary(mod1)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.00	0.14	-0.27	0.27	1.00	3433	2549
Marriage.s	0.37	0.14	0.10	0.63	1.00	4040	2867

Family Specific Parameters:

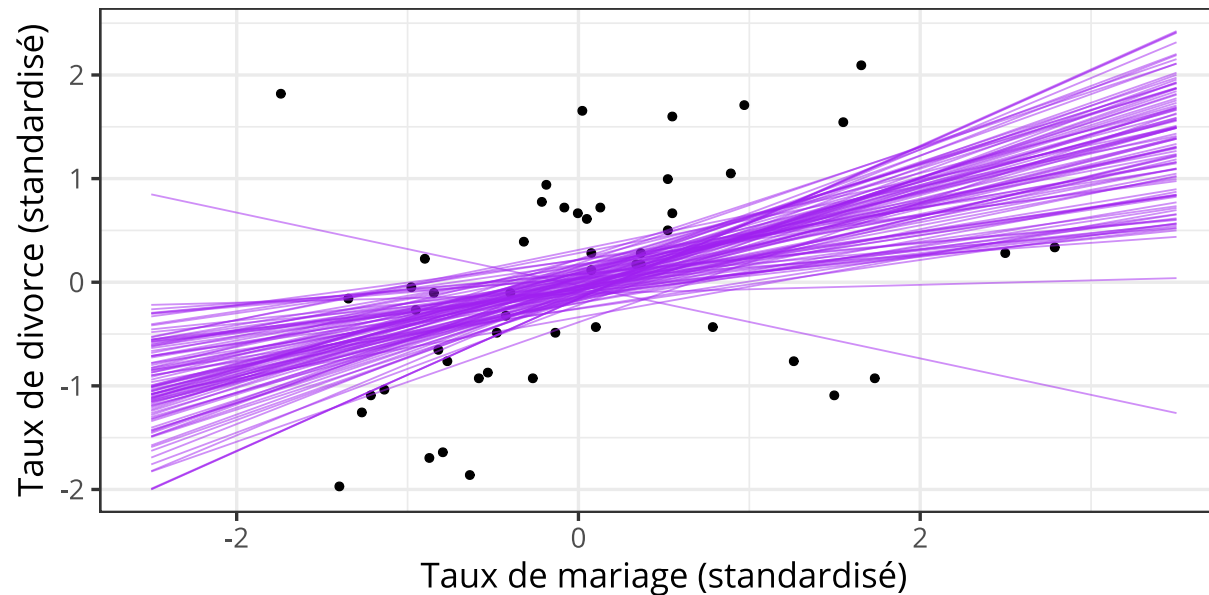
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.95	0.10	0.78	1.17	1.00	3340	2845

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).

# Posterior predictions

```
nd <- data.frame(Marriage.s = seq(from = -2.5, to = 3.5, length.out = 1e2) )

posterior_samples(mod1, pars = "^b_") %>%
  sample_n(size = 1e2) %>%
  rownames_to_column("draw") %>%
  expand(nesting(draw, b_Intercept, b_Marriage.s), a = c(-2.5, 3.5) ) %>%
  mutate(d = b_Intercept + b_Marriage.s * a) %>%
  ggplot(aes(x = a, y = d) ) +
  geom_point(data = df1, aes(x = Marriage.s, y = Divorce.s), size = 2) +
  geom_line(aes(group = draw), color = "purple", size = 0.5, alpha = 0.5) +
  labs(x = "Taux de mariage (standardisé)", y = "Taux de divorce (standardisé)")
```



# Influence de l'âge médian de mariage

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

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

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

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

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

```
priors <- c(  
  prior(normal(0, 10), class = Intercept),  
  prior(normal(0, 1), class = b),  
  prior(exponential(1), class = sigma)  
)  
  
mod2 <- brm(  
  Divorce.s ~ 1 + MedianAgeMarriage.s,  
  family = gaussian(),  
  prior = priors,  
  data = df1  
)
```

# Influence de l'âge médian de mariage

```
summary(mod2)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	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	3734	2671
MedianAgeMarriage.s	-0.59	0.12	-0.83	-0.36	1.00	3407	2746

Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	0.82	0.09	0.68	1.02	1.00	3085	2571

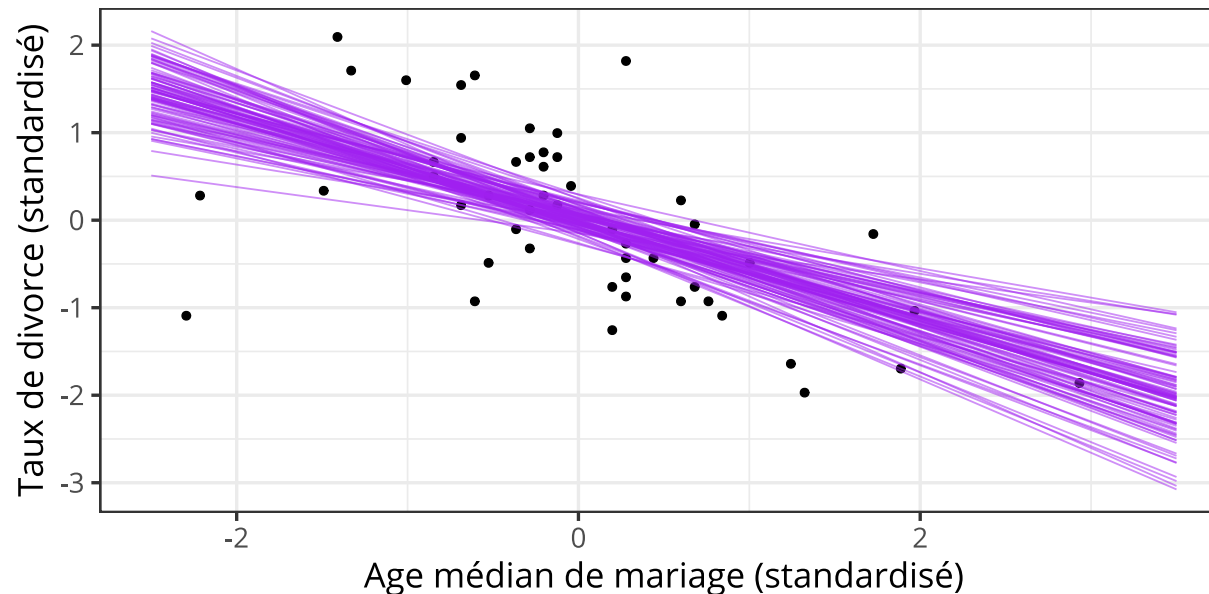
Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).



# Posterior predictions

```
nd <- data.frame(MedianAgeMarriage.s = seq(from = -3, to = 3.5, length.out = 1e2) )

posterior_samples(mod2, pars = "^b_") %>%
  sample_n(size = 1e2) %>%
  rownames_to_column("draw") %>%
  expand(nesting(draw, b_Intercept, b_MedianAgeMarriage.s), a = c(-2.5, 3.5) ) %>%
  mutate(d = b_Intercept + b_MedianAgeMarriage.s * a) %>%
  ggplot(aes(x = a, y = d) ) +
  geom_point(data = df1, aes(x = MedianAgeMarriage.s, y = Divorce.s), size = 2) +
  geom_line(aes(group = draw), color = "purple", size = 0.5, alpha = 0.5) +
  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

```
priors <- c(  
  prior(normal(0, 10), class = Intercept),  
  prior(normal(0, 1), class = b),  
  prior(exponential(1), class = sigma)  
)  
  
mod3 <- brm(  
  Divorce.s ~ 1 + Marriage.s + MedianAgeMarriage.s,  
  family = gaussian(),  
  prior = priors,  
  data = dfl  
)
```

# Régression multiple

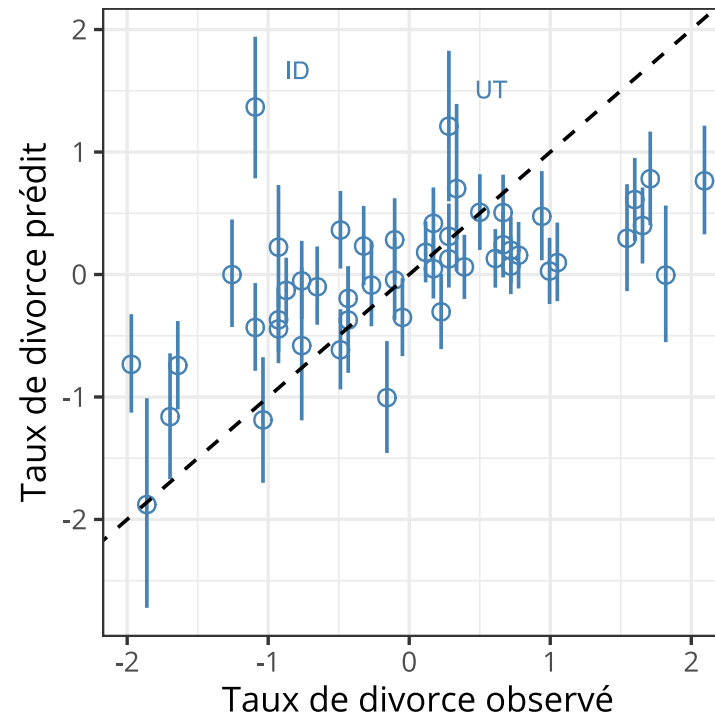
Interprétation : Une fois qu'on connaît l'âge median de mariage dans un état, connaître le taux de mariage de cet état n'apporte pas vraiment d'information supplémentaire...

```
posterior_summary(mod3, pars = "^b_")
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	2.467793e-06	0.1176557	-0.2258637	0.2342468
b_Marriage.s	-1.030547e-01	0.1657872	-0.4296366	0.2271726
b_MedianAgeMarriage.s	-6.631088e-01	0.1669599	-0.9844544	-0.3330344

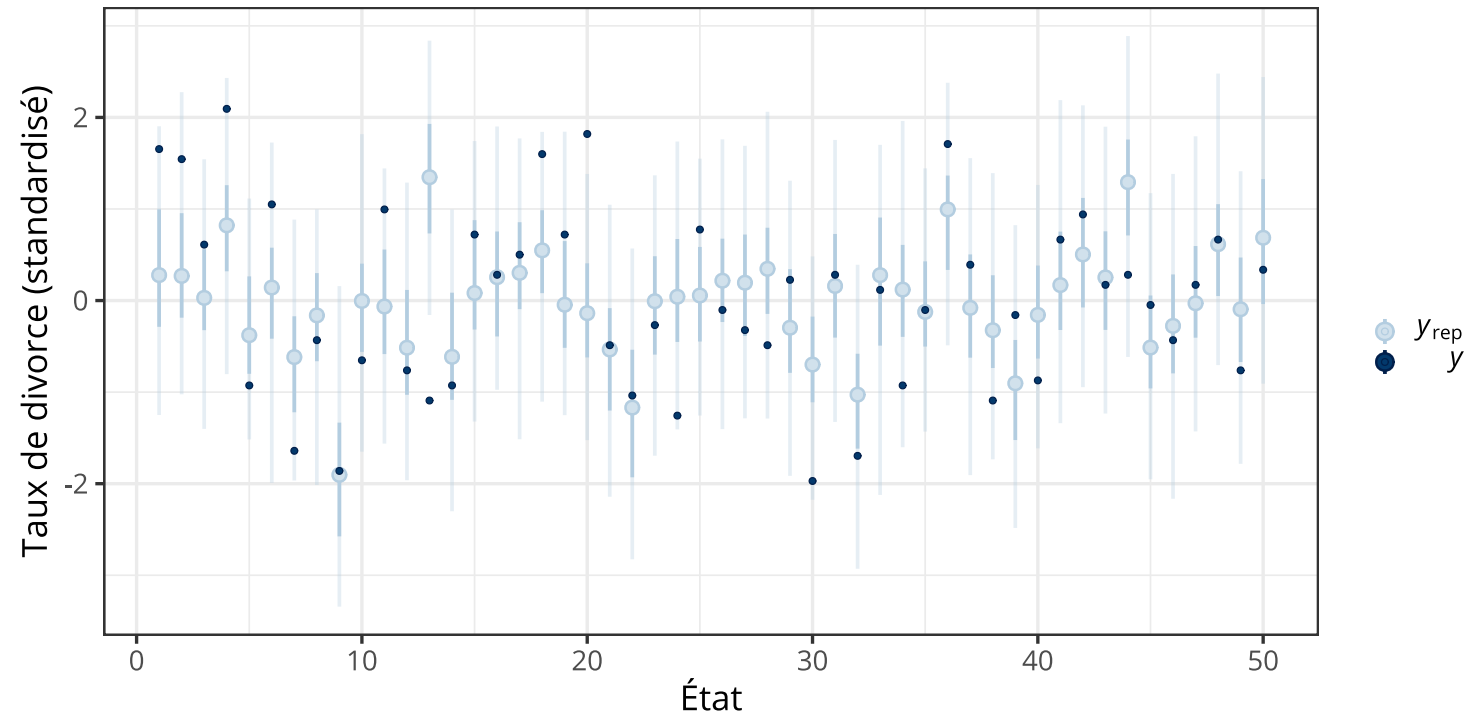
# Visualiser les prédictions du modèle

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 prédiction parfaite).

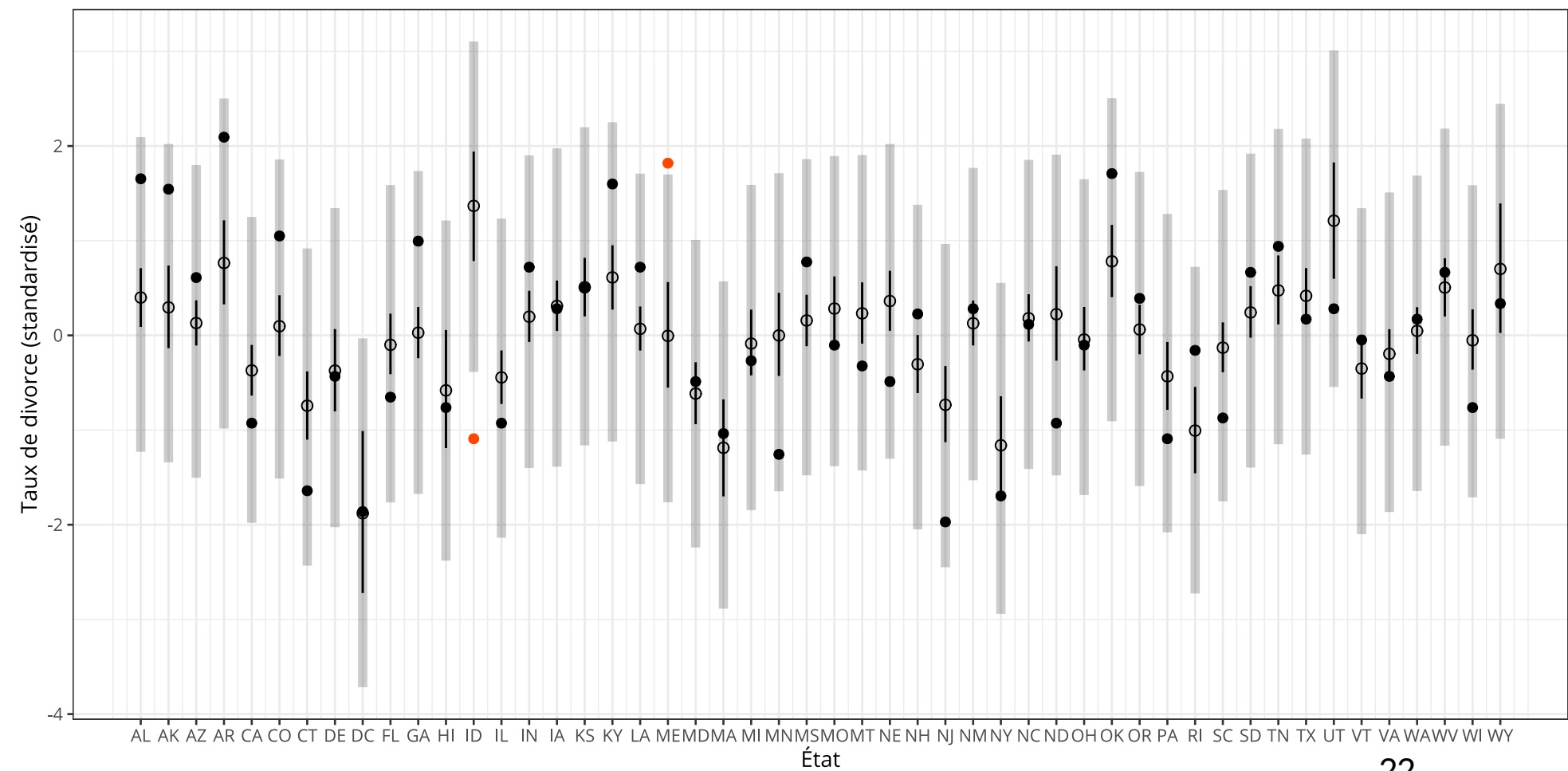


# Visualiser les prédictions du modèle

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



# Visualiser les prédictions du modèle



# Toujours plus de prédictors

Pourquoi ne pas simplement construire un modèle incluant tous les prédictors 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.

# 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.

```
set.seed(666) # afin de pouvoir reproduire les résultats

N <- 100 # nombre d'individus
height <- rnorm(N, 179, 5) # génère N observations
leg_prop <- runif(N, 0.4, 0.5) # taille des jambes (proportion taille totale)
leg_left <- leg_prop * height + rnorm(N, 0, 0.5) # taille jambe gauche (+ erreur)
leg_right <- leg_prop * height + rnorm(N, 0, 0.5) # taille jambe droite (+ erreur)
df2 <- data.frame(height, leg_left, leg_right) # création d'une dataframe

head(df2) # affiche les six première lignes
```

# 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.

```
set.seed(666) # afin de pouvoir reproduire les résultats

N <- 100 # nombre d'individus
height <- rnorm(N, 179, 5) # génère N observations
leg_prop <- runif(N, 0.4, 0.5) # taille des jambes (proportion taille totale)
leg_left <- leg_prop * height + rnorm(N, 0, 0.5) # taille jambe gauche (+ erreur)
leg_right <- leg_prop * height + rnorm(N, 0, 0.5) # taille jambe droite (+ erreur)
df2 <- data.frame(height, leg_left, leg_right) # création d'une dataframe

head(df2) # affiche les six première lignes
```

	height	leg_left	leg_right
1	182.7666	75.50967	76.00645
2	189.0718	81.10741	82.18046
3	177.2243	71.43856	71.49741
4	189.1408	82.81510	82.54405
5	167.9156	82.70860	84.00048
6	182.7920	84.86230	84.19933

# Multicolinéarité

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

```
priors <- c(  
  prior(normal(174, 10), class = Intercept),  
  prior(normal(0, 10), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
mod4 <- brm(  
  height ~ 1 + leg_left + leg_right,  
  prior = priors,  
  family = gaussian,  
  data = df2  
)
```

# 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) ?

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

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	153.08	7.41	138.95	167.65	1.00	4347	2731
leg_left	0.58	0.72	-0.85	1.99	1.00	1386	1715
leg_right	-0.26	0.72	-1.66	1.18	1.00	1374	1784

Family Specific Parameters:

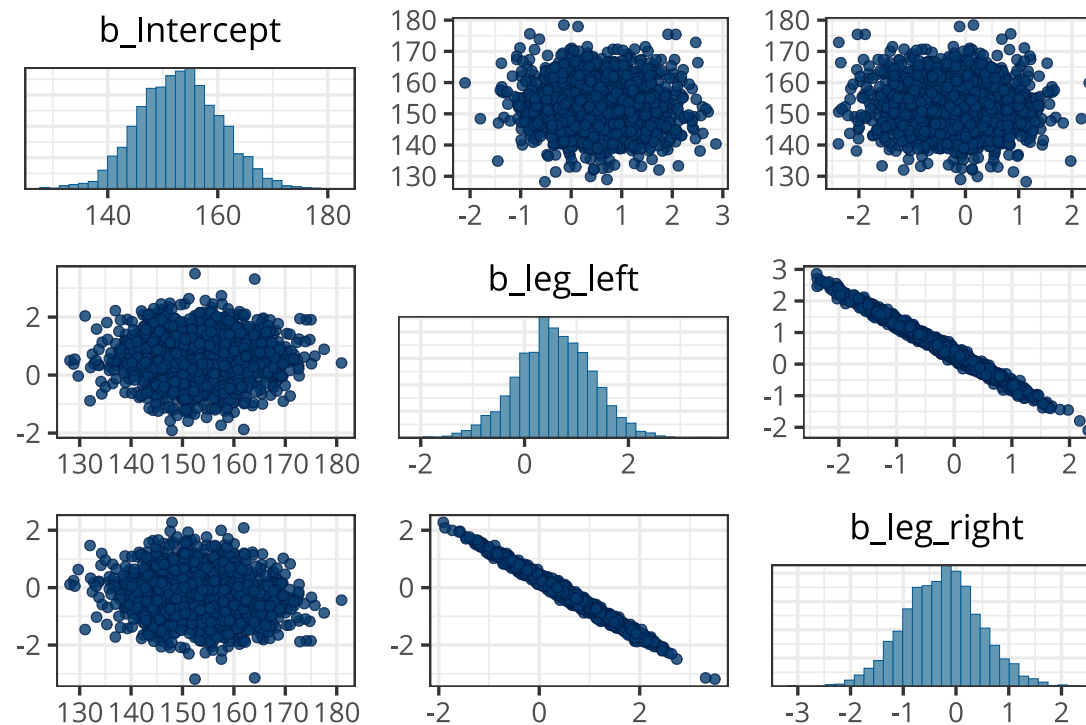
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	4.94	0.35	4.30	5.67	1.00	2016	2050

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 ? En représentant la distribution postérieure de ces deux paramètres.

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

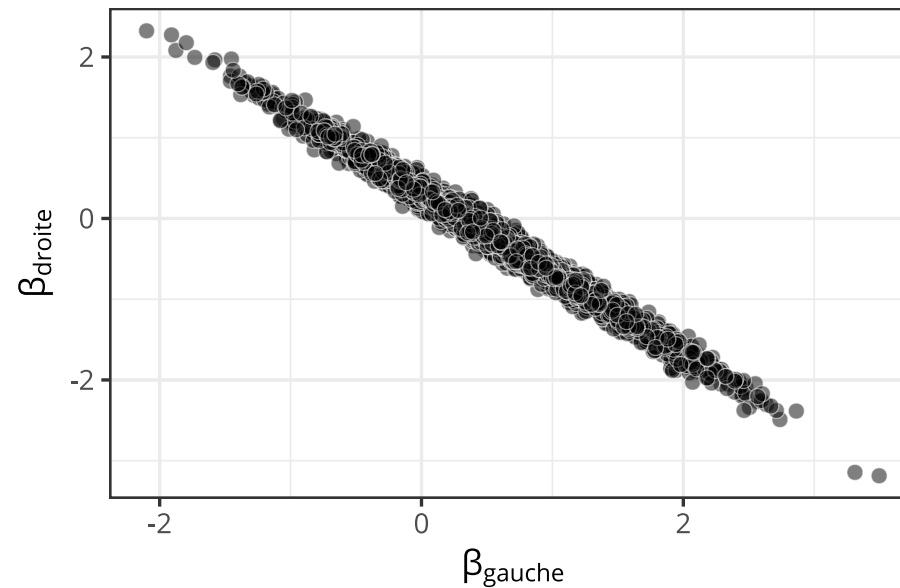


# Multicolinéarité

Comment traquer la colinéarité de deux prédicteurs ? En représentant la distribution postérieure de ces deux paramètres.

```
post <- posterior_samples(mod4)

post %>%
  ggplot(aes(x = b_leg_left, y = b_leg_right) ) +
  geom_point(pch = 21, size = 4, color = "white", fill = "black", alpha = 0.5) +
  labs(x = expression(beta[gauche]), y = expression(beta[droite]))
```

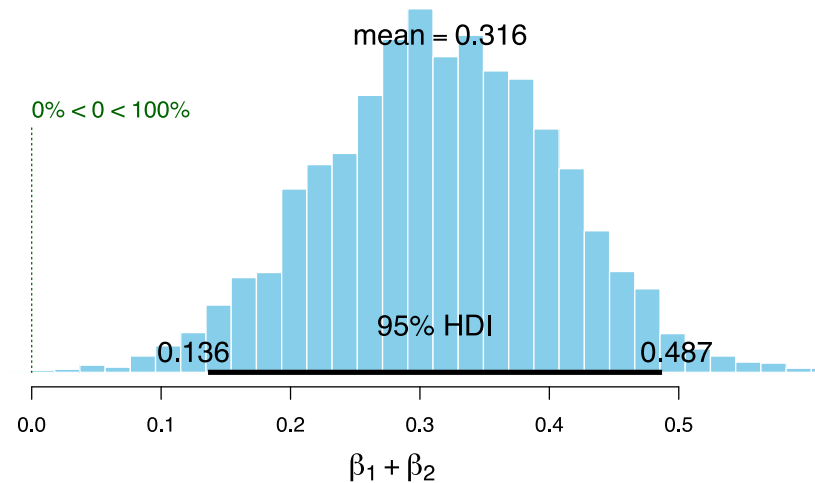


# Multicolinéarité

Le modèle précédent peut se réécrire en faisant apparaître la somme des deux prédicteurs  $\beta_1$  et  $\beta_2$ .

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha + (\beta_1 + \beta_2)x_i$$

```
library(BEST)
sum_legs <- post$b_leg_left + post$b_leg_right
plotPost(sum_legs, xlab = expression(beta[1] + beta[2]), compVal = 0)
```





# Multicolinéarité

On crée un nouveau modèle avec seulement une jambe.

```
priors <- c(  
  prior(normal(174, 10), class = Intercept),  
  prior(normal(0, 10), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
mod5 <- brm(  
  height ~ 1 + leg_left,  
  prior = priors,  
  family = gaussian,  
  data = df2  
)
```

# 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.

# 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.

```
posterior_summary(mod5)
```

# 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.

```
posterior_summary(mod5)
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	152.8377363	7.2866627	138.9360840	167.2205977
b_leg_left	0.3188943	0.0898381	0.1434423	0.4906666
sigma	4.9195822	0.3574195	4.3015601	5.7006899
lp__	-310.0692225	1.2537147	-313.2726712	-308.6577305

# 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.

```
posterior_summary(mod5)
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	152.8377363	7.2866627	138.9360840	167.2205977
b_leg_left	0.3188943	0.0898381	0.1434423	0.4906666
sigma	4.9195822	0.3574195	4.3015601	5.7006899
lp__	-310.0692225	1.2537147	-313.2726712	-308.6577305

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 prédictors qui sont eux-mêmes définis directement ou indirectement par d'autres prédictors inclus dans le modèle.

# 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.

# 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.



# 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

```
# nombre de plantes
N <- 100

# on simule différentes tailles à l'origine
h0 <- rnorm(N, mean = 10, sd = 2)

# on assigne différents traitements et on
# simule la présence de fungus et la pousse des plantes
treatment <- rep(0:1, each = N / 2)
fungus <- rbinom(N, size = 1, prob = 0.5 - treatment * 0.4)
h1 <- h0 + rnorm(N, mean = 5 - 3 * fungus)

# on rassemble les données dans une dataframe
df3 <- data.frame(h0, h1, treatment, fungus)

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

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

$$\mu_i = \alpha + \beta_1 h_{0i} + \beta_2 T_i + \beta_3 F_i$$

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

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

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

```
priors <- c(  
  prior(normal(0, 10), class = Intercept),  
  prior(normal(0, 10), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
mod6 <- brm(  
  h1 ~ 1 + h0 + treatment + fungus,  
  prior = priors,  
  family = gaussian,  
  data = df3  
)
```

# Post-treatment bias

On remarque que l'effet du traitement est négligeable. 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...

```
posterior_summary(mod6)
```

	Estimate	Est.Error	Q2.5	Q97.5
b_Intercept	4.32405061	0.49723007	3.3442629	5.3233570
b_h0	1.07329359	0.04470147	0.9846887	1.1641805
b_treatment	-0.08406279	0.19835499	-0.4743184	0.3057332
b_fungus	-2.64065745	0.22830291	-3.0894039	-2.1864978
sigma	0.91139012	0.06808294	0.7916867	1.0567342
lp__	-150.73922303	1.64077643	-154.9797577	-148.5774958

# 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 observées pourraient masquer l'effet du traitement.

```
mod7 <- brm(  
  h1 ~ 1 + h0 + treatment,  
  prior = priors,  
  family = gaussian,  
  data = df3  
)
```

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

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

# Post-treatment bias

```
summary(mod7)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	2.24	0.72	0.85	3.65	1.00	4603	3130
h0	1.17	0.07	1.04	1.30	1.00	4447	3272
treatment	0.73	0.27	0.20	1.29	1.00	4546	3231

Family Specific 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	4066	2948

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

```
data(Howell1)  
df4 <- Howell1
```

```
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 **genre** 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

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

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

$$\alpha \sim \text{Normal}(178, 100)$$

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

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

```
priors <- c(  
  prior(normal(178, 100), class = Intercept),  
  prior(normal(0, 10), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
mod8 <- brm(  
  height ~ 1 + male,  
  prior = priors,  
  family = gaussian,  
  data = df4  
)
```



# Prédicteurs catégoriels

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

# Prédicteurs catégoriels

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

```
fixef(mod8) # retrieves fixed effects
```

# Prédicteurs catégoriels

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

```
fixef(mod8) # retrieves fixed effects
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	134.840420	1.602114	131.779443	138.17051
male	7.255362	2.318785	2.598234	11.73568

# Prédicteurs catégoriels

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

```
fixef(mod8) # retrieves fixed effects
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	134.840420	1.602114	131.779443	138.17051
male	7.255362	2.318785	2.598234	11.73568

La pente  $\beta$  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  $\alpha$  et  $\beta$ .

# Prédicteurs catégoriels

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

```
fixef(mod8) # retrieves fixed effects
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	134.840420	1.602114	131.779443	138.17051
male	7.255362	2.318785	2.598234	11.73568

La pente  $\beta$  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  $\alpha$  et  $\beta$ .

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

# Prédicteurs catégoriels

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

```
fixef(mod8) # retrieves fixed effects
```

	Estimate	Est.Error	Q2.5	Q97.5
Intercept	134.840420	1.602114	131.779443	138.17051
male	7.255362	2.318785	2.598234	11.73568

La pente  $\beta$  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  $\alpha$  et  $\beta$ .

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

2.5%	50%	97.5%
138.7932	142.1269	145.3780

# Prédicteurs catégoriels

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

# Prédicteurs catégoriels

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

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

$$\mu_i = \alpha_f(1 - m_i) + \alpha_h m_i$$



# Prédicteurs catégoriels

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

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

$$\mu_i = \alpha_f(1 - m_i) + \alpha_h m_i$$

Cette formulation est strictement équivalente à la précédente car :

# Prédicteurs catégoriels

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

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

$$\mu_i = \alpha_f(1 - m_i) + \alpha_h m_i$$

Cette formulation est strictement équivalente à la précédente car :

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

# Prédicteurs catégoriels

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

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

$$\mu_i = \alpha_f(1 - m_i) + \alpha_h m_i$$

Cette formulation est strictement équivalente à la précédente car :

$$\mu_i = \alpha_f(1 - m_i) + \alpha_h m_i$$

$$= \alpha_f + (\alpha_h - \alpha_f)m_i$$

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

# Prédicteurs catégoriels

```
# on crée une nouvelle colonne pour les femmes
df4 <- df4 %>% mutate(female = 1 - male)

priors <- c(
  # il n'y a plus d'intercept dans ce modèle
  # prior(normal(178, 100), class = Intercept),
  prior(normal(0, 10), class = b),
  prior(exponential(0.01), class = sigma)
)

mod9 <- brm(
  height ~ 0 + female + male,
  prior = priors,
  family = gaussian,
  data = df4
)
```

# Prédicteurs catégoriels

```
summary(mod9)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

## Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
female	131.15	1.59	128.07	134.26	1.00	3863	3084
male	138.15	1.73	134.77	141.56	1.00	3877	3010

## Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	27.68	0.84	26.12	29.35	1.00	4314	3554

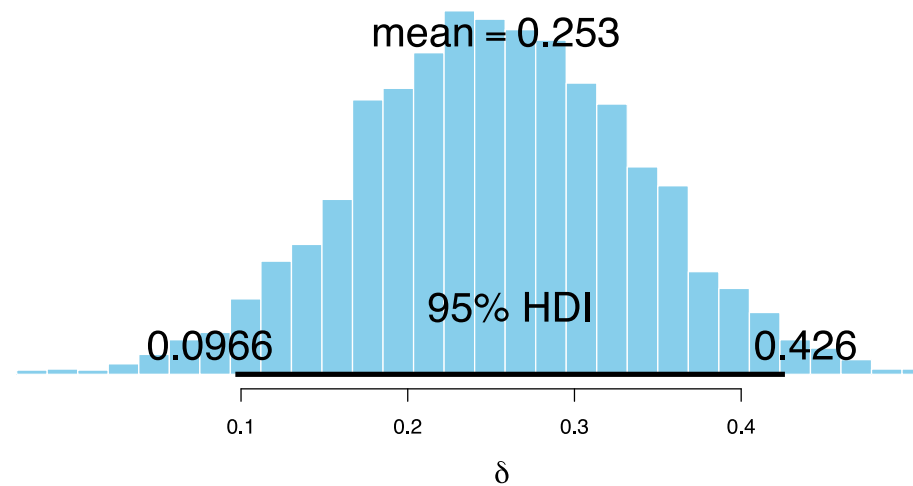
Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

# Prédicteurs catégoriels

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

```
post <- posterior_samples(mod9)

plotPost(
  (post$b_male - post$b_female) / post$sigma,
  cex = 2, xlab = expression(delta)
)
```



# Prédicteurs catégoriels

Nombre de catégories  $\geq 3$ .

# Prédicteurs catégoriels

Nombre de catégories  $\geq 3$ .

```
data(milk)  
df5 <- milk  
str(df5)
```



# Prédicteurs catégoriels

Nombre de catégories  $\geq 3$ .

```
data(milk)
df5 <- milk
str(df5)
```

```
'data.frame':  29 obs. of  8 variables:
 $ clade      : Factor w/  4 levels "Ape","New World Monkey",...: 4 4 4 4 4 2 2 2 2 2 ...
 $ species    : Factor w/ 29 levels "A palliata","Alouatta seniculus",...: 11 8 9 10 16 2 1 6 28
27 ...
 $ 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 ...
```

# Prédicteurs catégoriels

Nombre de catégories  $\geq 3$ .

```
data(milk)
df5 <- milk
str(df5)
```

```
'data.frame':  29 obs. of  8 variables:
 $ clade      : Factor w/  4 levels "Ape","New World Monkey",...: 4 4 4 4 4 2 2 2 2 2 ...
 $ species    : Factor w/ 29 levels "A palliata","Alouatta seniculus",...: 11 8 9 10 16 2 1 6 28
27 ...
 $ 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*.

# Prédicteurs catégoriels

Nombre de catégories  $\geq 3$ .

```
data(milk)
df5 <- milk
str(df5)
```

```
'data.frame':  29 obs. of  8 variables:
 $ clade      : Factor w/  4 levels "Ape","New World Monkey",...: 4 4 4 4 4 2 2 2 2 2 ...
 $ species    : Factor w/ 29 levels "A palliata","Alouatta seniculus",...: 11 8 9 10 16 2 1 6 28
27 ...
 $ 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*.

```
df5$clade.NWM <- ifelse(df5$clade == "New World Monkey", 1, 0)
df5$clade.OWM <- ifelse(df5$clade == "Old World Monkey", 1, 0)
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	$NWM_i$	$OWM_i$	$S_i$	$\mu_i$
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

```
priors <- c(  
  prior(normal(0.6, 10), class = Intercept),  
  prior(normal(0, 1), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
mod10 <- brm(  
  kcal.per.g ~ 1 + clade.NWM + clade.OWM + clade.S,  
  prior = priors,  
  family = gaussian,  
  data = df5  
)
```

# Prédicteurs catégoriels

```
summary(mod10)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	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	3200	2310
clade.NWM	0.17	0.06	0.04	0.29	1.00	3646	3038
clade.OWM	0.24	0.07	0.11	0.37	1.00	3271	2650
clade.S	-0.04	0.07	-0.18	0.11	1.00	3897	3039

Family Specific 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	3190	2830

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).

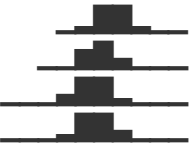
# Prédicteurs catégoriels

```
# retrieves posterior samples
post <- posterior_samples(mod10)

# retrieves posterior samples for each category
mu.ape <- post$b_Intercept
mu.NWM <- post$b_Intercept + post$b_clade.NWM
mu.OWM <- post$b_Intercept + post$b_clade.OWM
mu.S <- post$b_Intercept + post$b_clade.S

# displays a summary of the posterior samples
precis(data.frame(mu.ape, mu.NWM, mu.OWM, mu.S), prob = 0.95)
```

```
'data.frame': 4000 obs. of 4 variables:
      mean    sd 2.5% 97.5% histogram
mu.ape 0.55 0.04 0.46 0.63
mu.NWM 0.71 0.05 0.63 0.80
mu.OWM 0.79 0.05 0.69 0.89
mu.S   0.51 0.06 0.39 0.63
```



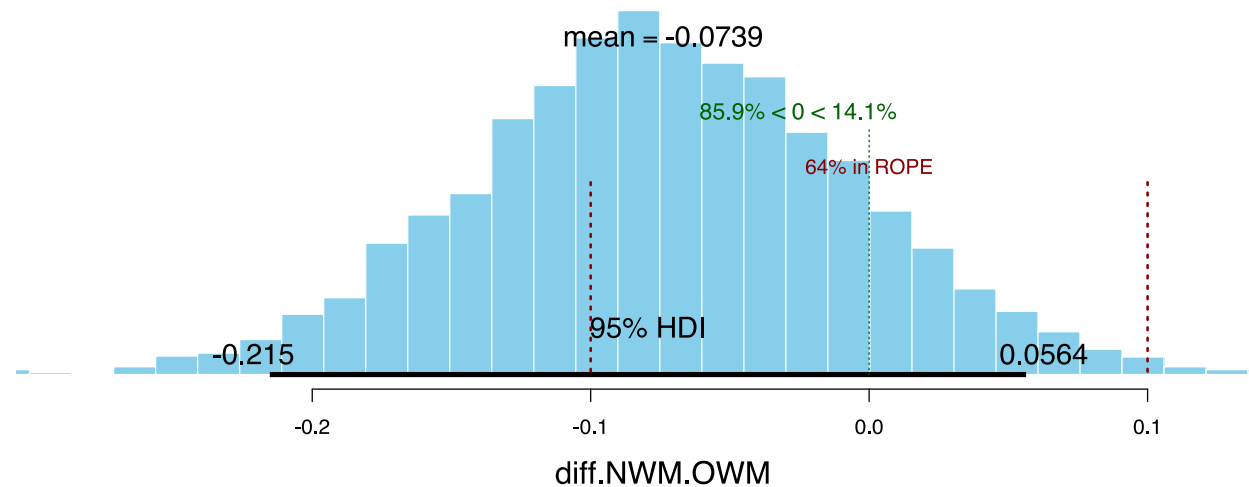
# Prédicteurs catégoriels

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

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

```
      2.5%      50%      97.5%  
-0.20948141 -0.07519787  0.06285252
```

```
plotPost(diff.NWM.OWM, compVal = 0, ROPE = c(-0.1, 0.1) )
```





# 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 genre, `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).

```
priors <- c(  
  prior(normal(0.6, 10), class = b),  
  prior(exponential(0.01), class = sigma)  
)  
  
modl1 <- brm(  
  # modèle sans intercept avec seulement un prédicteur catégoriel (facteur)  
  kcal.per.g ~ 0 + clade,  
  prior = priors,  
  family = gaussian,  
  data = df5  
)
```

# Prédicteurs catégoriels

```
summary(mod11)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	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	4421	3088
cladeNewWorldMonkey	0.71	0.05	0.63	0.81	1.00	5407	3063
cladeOldWorldMonkey	0.79	0.05	0.68	0.89	1.00	4904	2574
cladeStrepsirrhine	0.51	0.06	0.39	0.62	1.00	4859	2265

Family Specific 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	3538	2940

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

# Interaction

Jusque là, 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 tulipe soit **conditionnelle** à l'humidité du sol...

# Interaction

```
data(tulips)
df6 <- tulips

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 :

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

Modèle avec interaction :

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

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

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

# Interaction

```
priors <- c(
  prior(normal(130, 100), class = Intercept),
  prior(normal(0, 100), class = b),
  prior(exponential(0.01), class = sigma)
)

mod12 <- brm(
  blooms ~ 1 + water.c + shade.c,
  prior = priors,
  family = gaussian,
  data = df6
)
```

```
mod13 <- brm(
  blooms ~ 1 + water.c * shade.c,
  # equivalent to blooms ~ 1 + water.c + shade.c + water.c:shade.c
  prior = priors,
  family = gaussian,
  data = df6
)
```



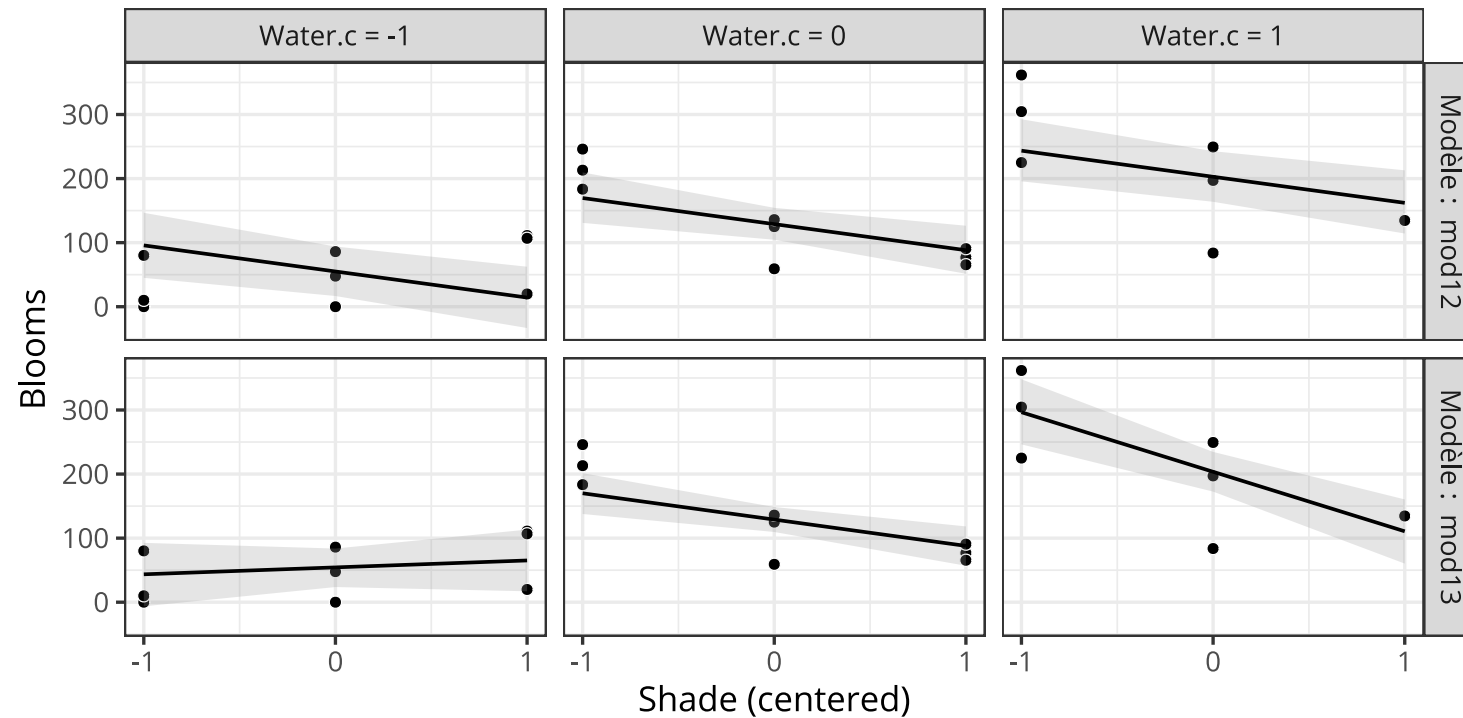
# Interaction

	term	mod12	mod13
1	b_Intercept	128.90776	128.96319
2	b_water.c	73.90532	74.61779
3	b_shade.c	-40.71308	-41.06773
4	sigma	63.60477	51.27472
5	b_water.c:shade.c	NA	-51.86903

- L'intercept  $\alpha$  représente la valeur attendue de `blooms` quand `water` et `shade` sont à 0 (i.e., la moyenne générale de la variable dépendante).
- La pente  $\beta_W$  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  $\beta_S$  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  $\beta_{WS}$  nous renseigne sur l'effet attendu de `water` sur `blooms` quand `shade` augment d'une unité (et réciproquement).

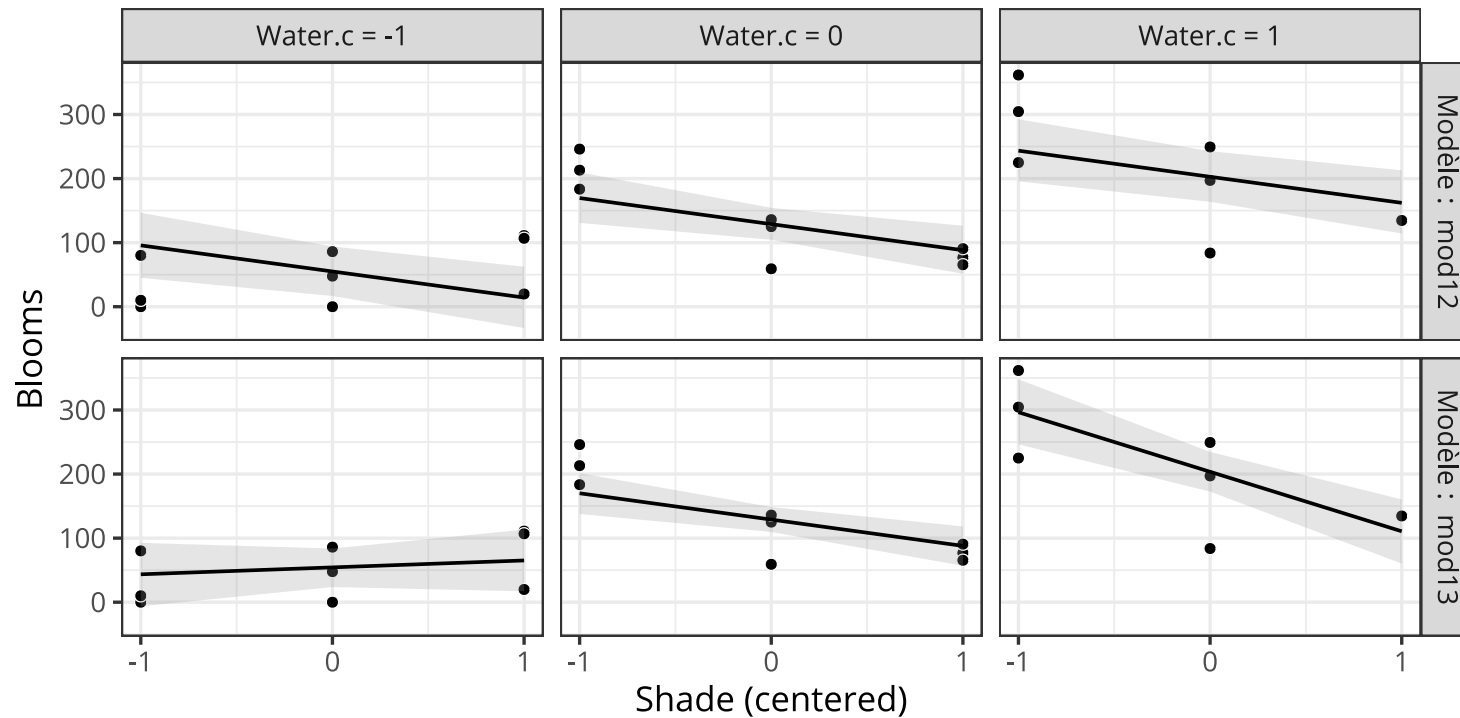
# Interaction

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 la luminosité et la 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.

# 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 étendu le modèle de régression aux prédicteurs catégoriels, et introduit le concept d'interaction entre différentes variables prédictrices.

# 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 étendu le modèle de régression aux prédicteurs catégoriels, et introduit le concept d'interaction entre différentes variables prédictives.

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.

# 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 étendu le modèle de régression aux prédicteurs catégoriels, et introduit le concept d'interaction entre différentes variables prédictrices.

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).

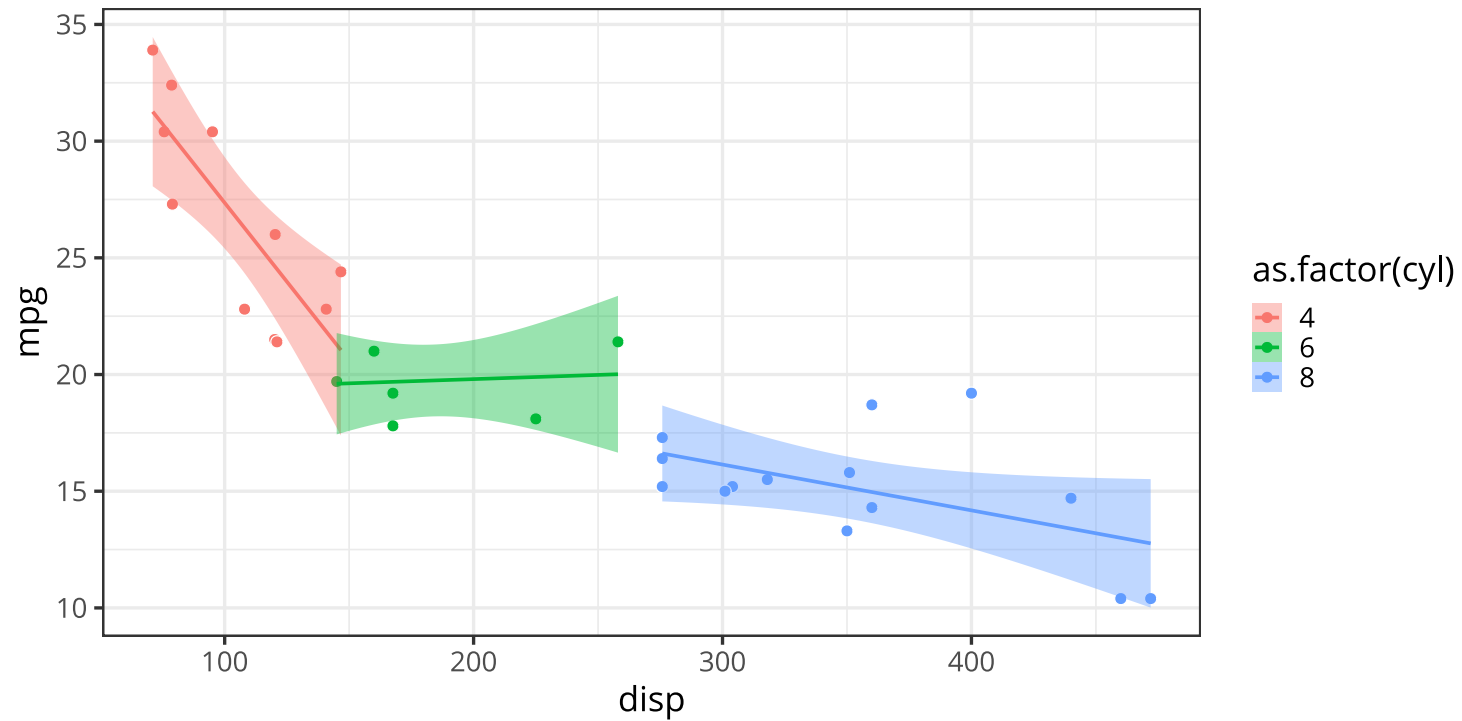
```
data(mtcars)
head(mtcars, 10)
```

	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.



# Exercise #1

```
mtcars$disp.s <- as.numeric(scale(mtcars$disp) )
mtcars$cyl.s <- as.numeric(scale(mtcars$cyl) )

m_cyl <- lm(mpg ~ disp.s * cyl.s, data = mtcars)
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

```
priors <- c(  
  prior(normal(0, 100), class = Intercept),  
  prior(normal(0, 10), class = b),  
  prior(exponential(0.1), class = sigma)  
)  
  
mod14 <- brm(  
  mpg ~ 1 + disp.s * cyl.s,  
  prior = priors,  
  family = gaussian,  
  data = mtcars  
)
```

# Proposition de réponse

```
summary(mod14)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	17.13	1.13	14.92	19.38	1.00	1846	2308
disp.s	-5.67	1.56	-8.82	-2.63	1.00	1449	2133
cyl.s	0.25	1.54	-2.80	3.30	1.00	1392	2205
disp.s:cyl.s	3.39	1.16	1.09	5.67	1.00	1696	2286

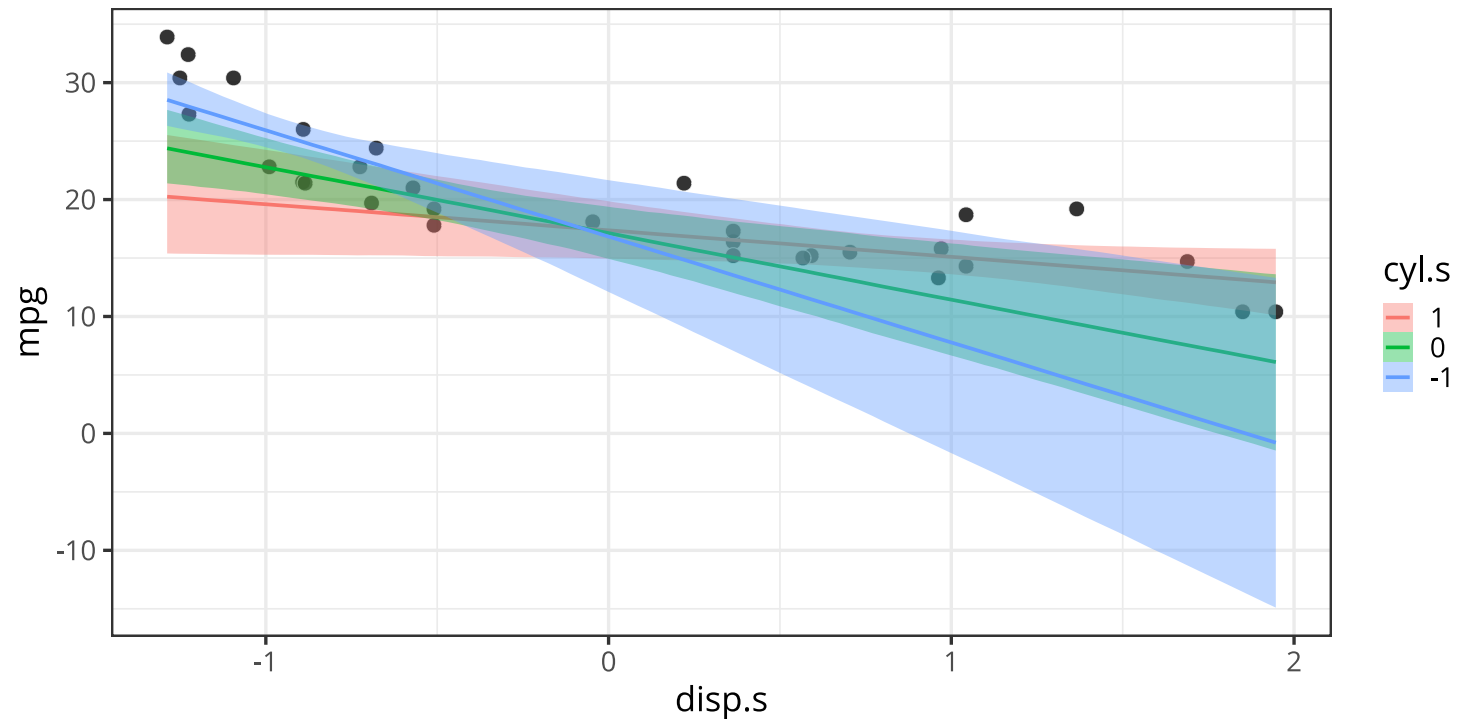
Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	2.78	0.38	2.17	3.66	1.00	2723	2563

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).

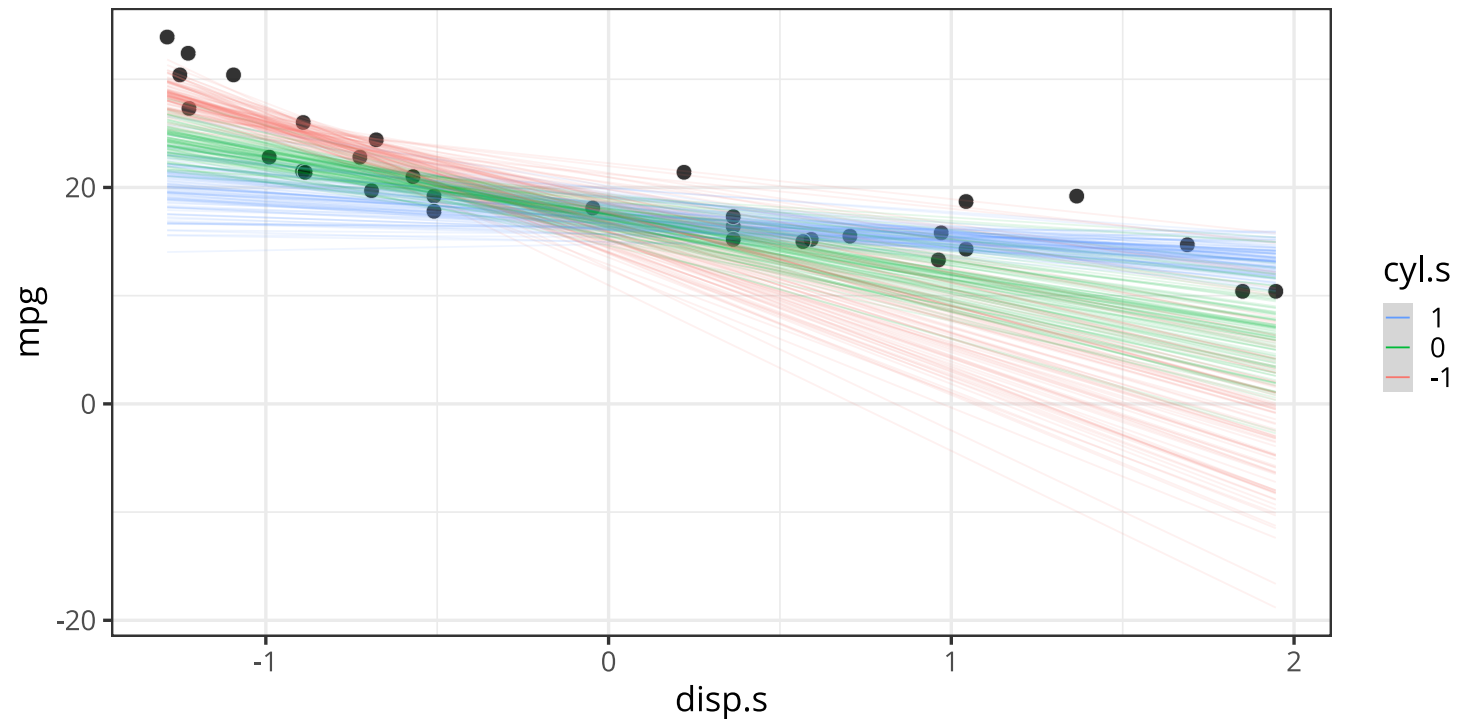
# Proposition de réponse

```
plot(  
  conditional_effects(mod14, effects = "disp.s:cyl.s"),  
  points = TRUE,  
  point_args = list(  
    alpha = 0.8, shape = 21, size = 4,  
    color = "white", fill = "black"  
  ),  
  theme = theme_bw(base_size = 20, base_family = "Open Sans")  
)
```



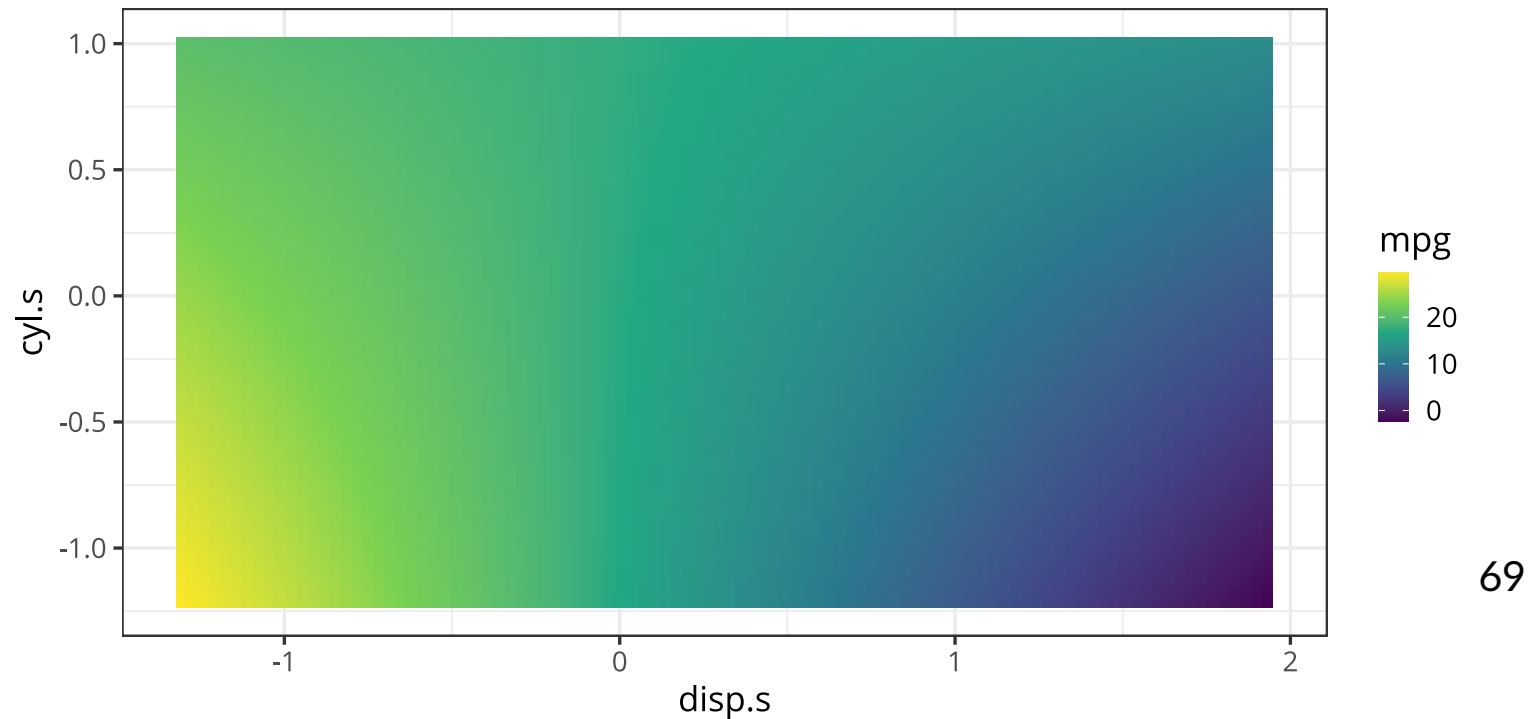
# Proposition de réponse

```
plot(  
  conditional_effects(mod14, effects = "disp.s:cyl.s", spaghetti = TRUE, nsamples = 1e2),  
  points = TRUE, mean = FALSE,  
  point_args = list(  
    alpha = 0.8, shape = 21, size = 4,  
    color = "white", fill = "black"  
  ),  
  theme = theme_bw(base_size = 20, base_family = "Open Sans")  
)
```



# Proposition de réponse

```
plot(  
  conditional_effects(  
    mod14, effects = "disp.s:cyl.s",  
    surface = TRUE, resolution = 1e2  
  ),  
  stype = "raster", # contour or raster  
  surface_args = list(hjust = 0),  
  theme = theme_bw(base_size = 20, base_family = "Open Sans")  
)
```



## 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.

```
data(airquality)
df7 <- airquality[complete.cases(airquality), ] # removes NAs

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 d'Ozone en fonction de la force du vent et de la température.

1. Écrire le modèle mathématique.
2. 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.
3. 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

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

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

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

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

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

# Proposition de réponse, fitter le modèle

```
df7$Wind.s <- scale(df7$Wind)
df7$Temp.s <- scale(df7$Temp)

priors <- c(
  prior(normal(50, 10), class = Intercept),
  prior(normal(0, 10), class = b),
  prior(exponential(0.01), class = sigma)
)

mod15 <- brm(
  Ozone ~ 1 + Wind.s + Temp.s,
  prior = priors,
  family = gaussian,
  data = df7
)
```

# Proposition de réponse, estimations du modèle

```
summary(mod15)
```

```
Family: gaussian
Links: mu = identity; sigma = 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
```

## Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	42.46	2.06	38.38	46.38	1.00	3955	3117
Wind.s	-11.52	2.28	-16.08	-6.96	1.00	3851	3100
Temp.s	16.82	2.31	12.20	21.34	1.00	3960	3058

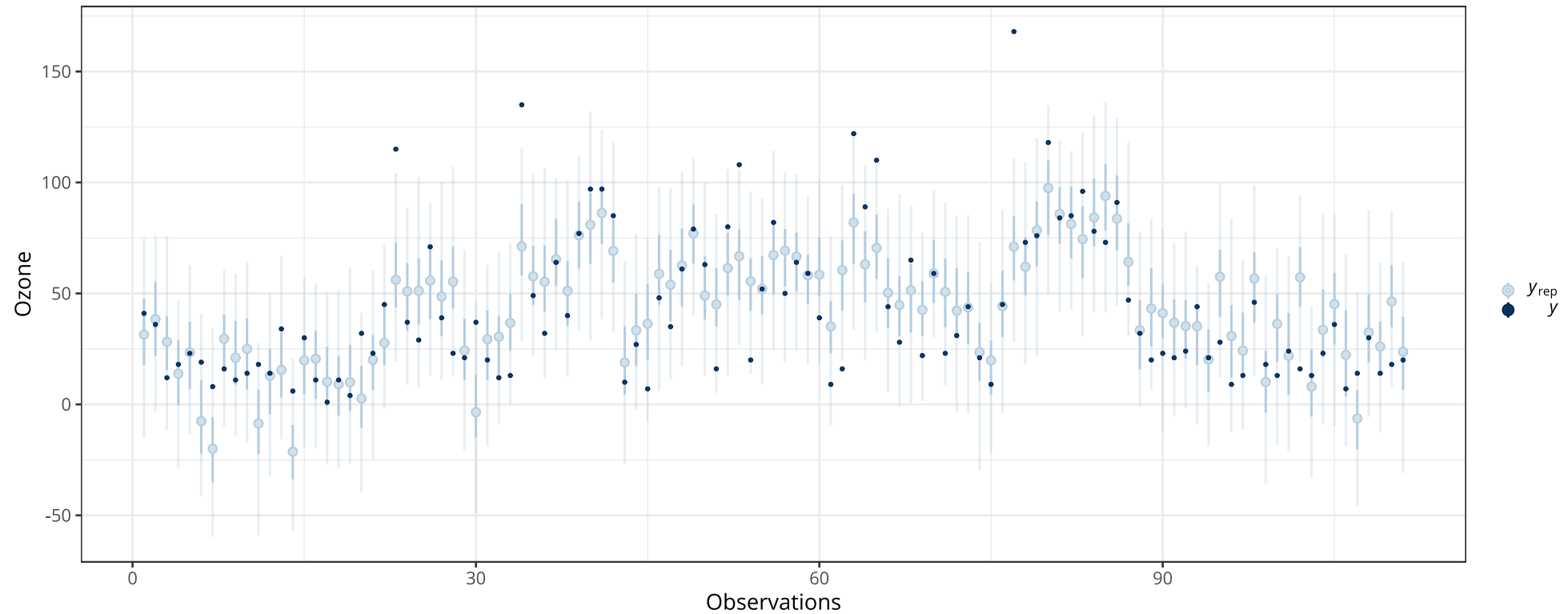
## Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	21.96	1.51	19.19	25.06	1.00	4006	2931

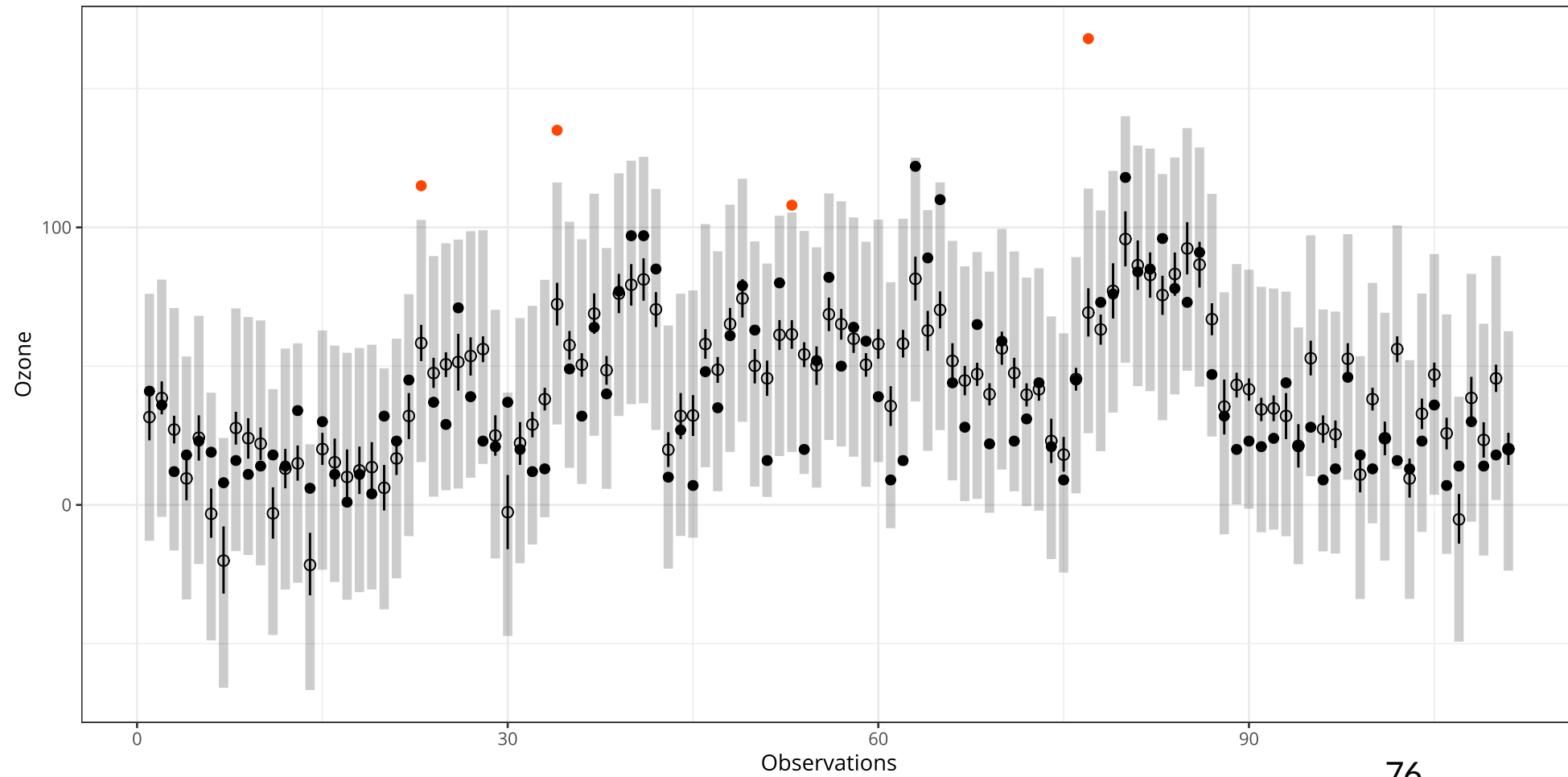
Draws were sampled using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

# Proposition de réponse, posterior predictive checking

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



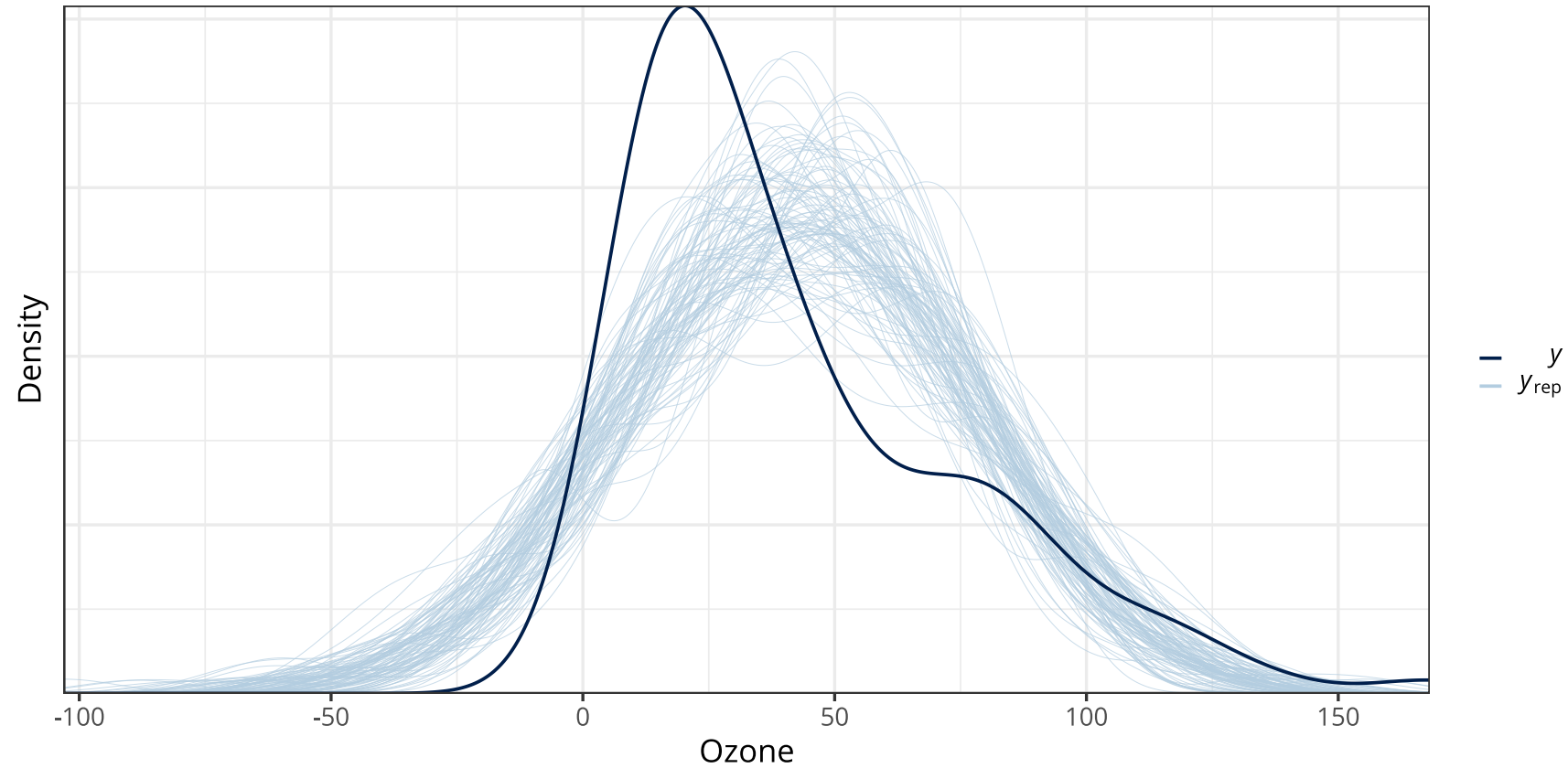
# Proposition de réponse, posterior predictive checking



76

# Proposition de réponse, posterior predictive checking

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



# Proposition de réponse, posterior predictive checking

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

