Introduction à la modélisation statistique bayésienne

Ladislas Nalborczyk
GIPSA-lab, CNRS, Univ. Grenoble Alpes

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 Hackaton

Null Hypothesis Significance Testing (NHST)

On s'intéresse aux différences de taille entre hommes et femmes. On va mesurer 100 femmes et 100 hommes.

Null Hypothesis Significance Testing (NHST)

On s'intéresse aux différences de taille entre hommes et femmes. On va mesurer 100 femmes et 100 hommes.

```
men <- rnorm(100, 175, 10) # 100 tailles d'hommes
women <- rnorm(100, 170, 10) # 100 tailles de femmes
```

Null Hypothesis Significance Testing (NHST)

On s'intéresse aux différences de taille entre hommes et femmes. On va mesurer 100 femmes et 100 hommes.

```
men <- rnorm(100, 175, 10) # 100 tailles d'hommes
women <- rnorm(100, 170, 10) # 100 tailles de femmes
```

```
t.test(men, women)
```

Null Hypothesis Significance Testing (NHST)

On s'intéresse aux différences de taille entre hommes et femmes. On va mesurer 100 femmes et 100 hommes.

```
men <- rnorm(100, 175, 10) # 100 tailles d'hommes
women <- rnorm(100, 170, 10) # 100 tailles de femmes
```

```
t.test(men, women)
```

```
    Welch Two Sample t-test

data:  men and women
t = 3.0835, df = 197.78, p-value = 0.002338
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.585804 7.213028
sample estimates:
mean of x mean of y
 173.5682  169.1688
```

Null Hypothesis Significance Testing (NHST)

On va simuler des t-valeurs issues de données générées sous l'hypothèse d'une absence de différence entre hommes et femmes.

Null Hypothesis Significance Testing (NHST)

On va simuler des t-valeurs issues de données générées sous l'hypothèse d'une absence de différence entre hommes et femmes.

```
nsims <- 1e4 # nombre de simulations
t <- rep(NA, nsims) # initialisation d'un vecteur vide

for (i in 1:nsims) {

    men2 <- rnorm(100, 170, 10) # 100 tailles d'hommes
    women2 <- rnorm(100, 170, 10) # 100 tailles de femmes
    t[i] <- t.test(men2, women2)$statistic # on conserve la t-valeur

}
```

Null Hypothesis Significance Testing (NHST)

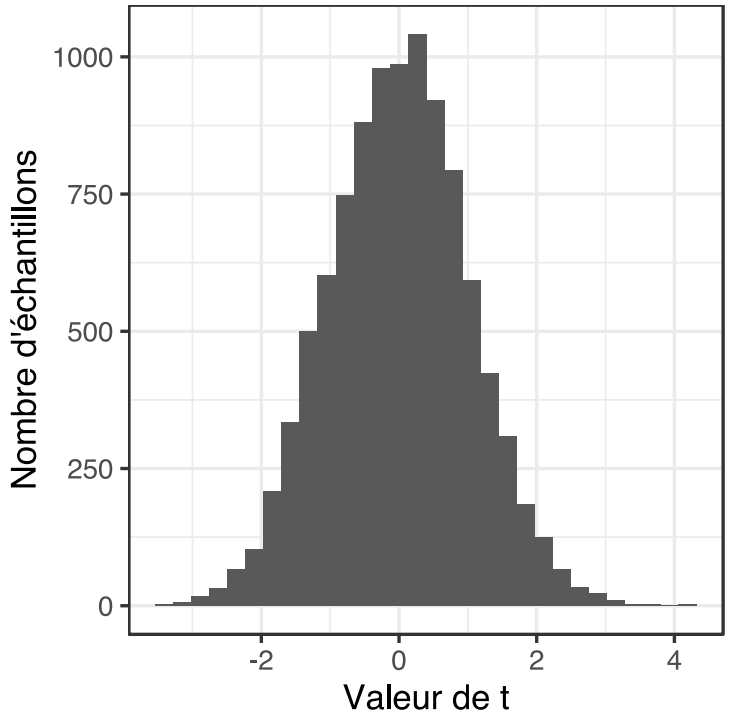
On va simuler des t-valeurs issues de données générées sous l'hypothèse d'une absence de différence entre hommes et femmes.

```
nsims <- 1e4 # nombre de simulations
t <- rep(NA, nsims) # initialisation d'un vecteur vide

for (i in 1:nsims) {

    men2 <- rnorm(100, 170, 10) # 100 tailles d'hommes
    women2 <- rnorm(100, 170, 10) # 100 tailles de femmes
    t[i] <- t.test(men2, women2)$statistic # on conserve la t-valeur

}
```

```
# une autre manière de réaliser la même opération, sans boucle for
t <- replicate(nsims, t.test(rnorm(100, 170, 10), rnorm(100, 170, 10) )$statistic)
```

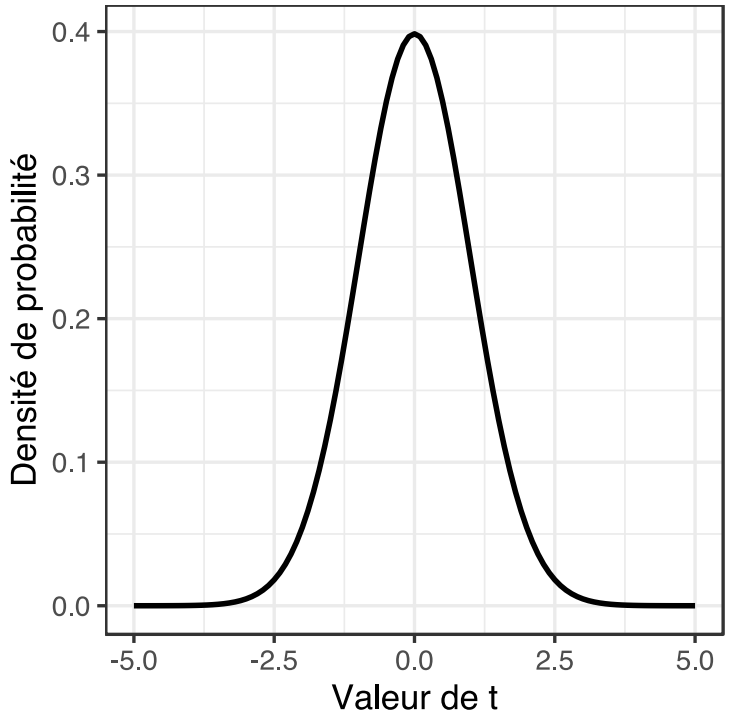
Null Hypothesis Significance Testing (NHST)

```
data.frame(t = t) %>%
    ggplot(aes(x = t) ) +
    geom_histogram() +
    theme_bw(base_size = 20) +
    labs(x = "Valeur de t", y = "Nombre d'échantillons")
```



Null Hypothesis Significance Testing (NHST)

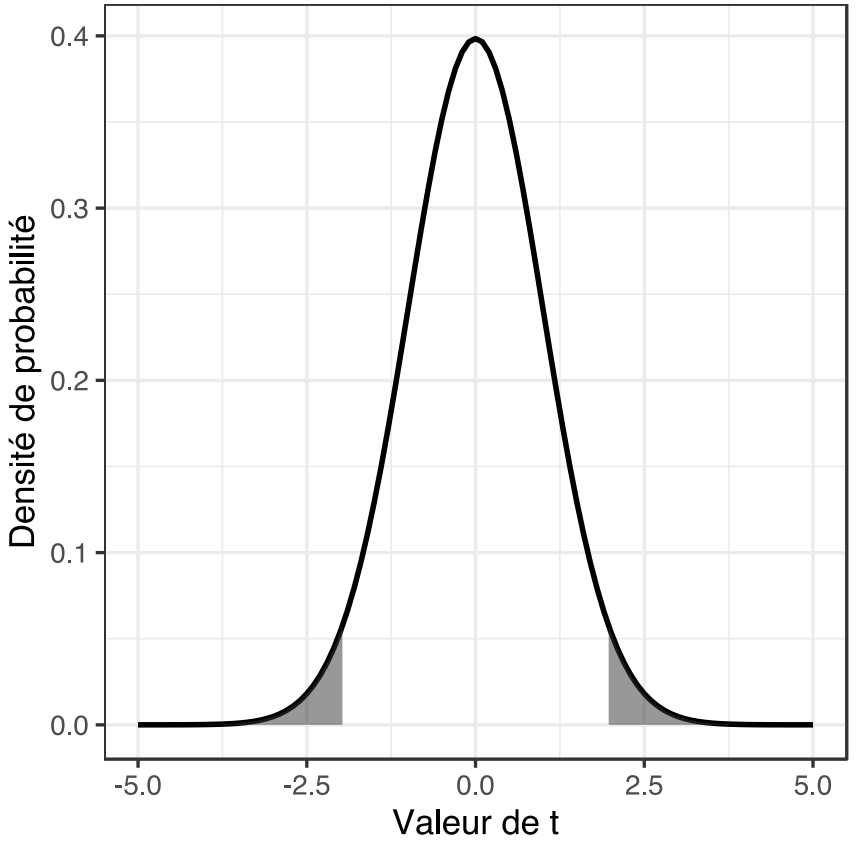
```
data.frame(x = c(-5, 5) ) %>%
    ggplot(aes(x = x) ) +
    stat_function(fun = dt, args = list(df = t.test(men, women)$parameter), size = 1.5) +
    theme_bw(base_size = 20) +
    labs(x = "Valeur de t", y = "Densité de probabilité")
```



Null Hypothesis Significance Testing (NHST)

```r
alpha <- 0.05
abs(qt(alpha / 2, df = t.test(men, women)$parameter) ) # valeur de t critique
```

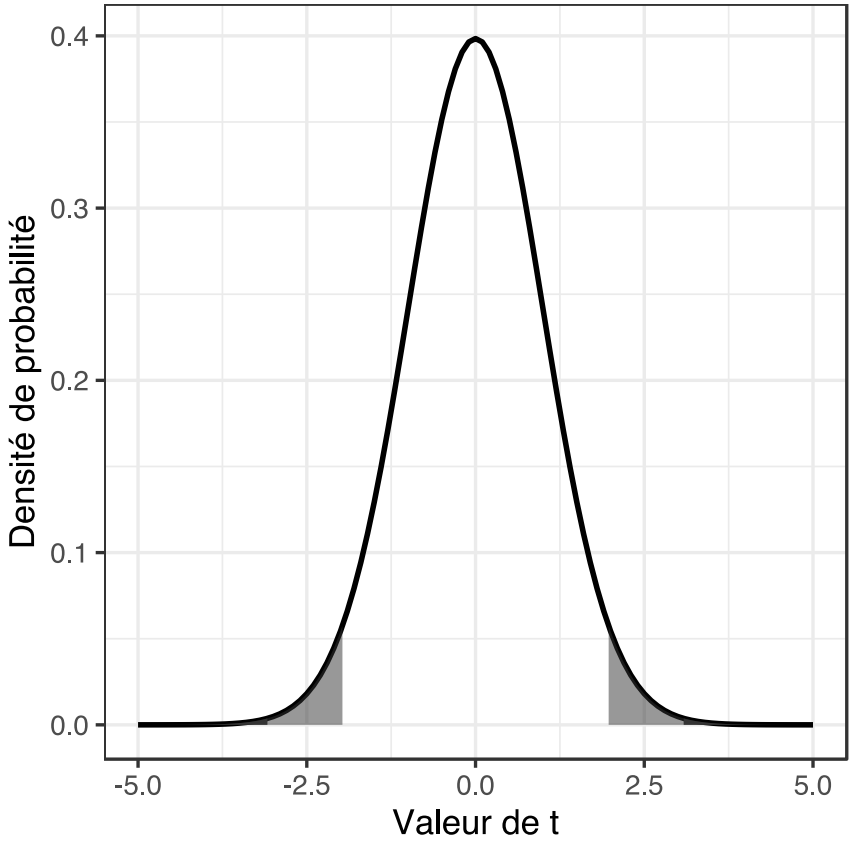
```
[1] 1.972031
```



Null Hypothesis Significance Testing (NHST)

```
tobs <- t.test(men, women)$statistic # valeur de t observée
tobs %>% as.numeric
```

```
[1] 3.083504
```



P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

```
[1] 0.002338075
```

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

```
[1] 0.002338075
```

```
tvalue <- abs(t.test(men, women)$statistic)
df <- t.test(men, women)$parameter

2 * integrate(dt, tvalue, Inf, df = df)$value
```

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

```
[1] 0.002338075
```

```
tvalue <- abs(t.test(men, women)$statistic)
df <- t.test(men, women)$parameter

2 * integrate(dt, tvalue, Inf, df = df)$value
```

```
[1] 0.002338076
```

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

```
[1] 0.002338075
```

```
tvalue <- abs(t.test(men, women)$statistic)
df <- t.test(men, women)$parameter

2 * integrate(dt, tvalue, Inf, df = df)$value
```

```
[1] 0.002338076
```

```
2 * (1 - pt(abs(t.test(men, women)$statistic), t.test(men, women)$parameter) )
```

P-valeur

Une *p*-valeur est une aire sous la courbe, une intégrale, sous la distribution de statistiques de test sous l'hypothèse nulle (i.e., étant admis que l'hypothèse nulle est vraie). La *p*-valeur indique la probabilité d'observer la valeur de la statistique de test observée, ou une valeur plus extrême, sous l'hypothèse nulle.

$$p[\mathbf{t}(\mathbf{x}^{\text{rep}}; H_0) \geq t(x)]$$

```
t.test(men, women)$p.value
```

```
[1] 0.002338075
```

```
tvalue <- abs(t.test(men, women)$statistic)
df <- t.test(men, women)$parameter

2 * integrate(dt, tvalue, Inf, df = df)$value
```

```
[1] 0.002338076
```

```
2 * (1 - pt(abs(t.test(men, women)$statistic), t.test(men, women)$parameter) )
```

```
          t
0.002338075
```

Intervalles de confiance

Les intervalles de confiance peuvent être interprétés comme des régions de significativité (ou des régions de *compatibilité* avec l'hypothèse nulle). Par conséquent, un intervalle de confiance contient la même information qu'une *p*-valeur et s'interprète de manière similaire.

Intervalles de confiance

Les intervalles de confiance peuvent être interprétés comme des régions de significativité (ou des régions de *compatibilité* avec l'hypothèse nulle). Par conséquent, un intervalle de confiance contient la même information qu'une *p*-valeur et s'interprète de manière similaire.

On ne peut pas dire qu'un intervalle de confiance a une probabilité de 95% de contenir la vraie valeur (i.e., la valeur dans la population) de θ (cf. the inverse fallacy), contrairement à l'intervalle de crédibilité bayésien.

Intervalles de confiance

Les intervalles de confiance peuvent être interprétés comme des régions de significativité (ou des régions de *compatibilité* avec l'hypothèse nulle). Par conséquent, un intervalle de confiance contient la même information qu'une *p*-valeur et s'interprète de manière similaire.

On ne peut pas dire qu'un intervalle de confiance a une probabilité de 95% de contenir la vraie valeur (i.e., la valeur dans la population) de θ (cf. the inverse fallacy), contrairement à l'intervalle de crédibilité bayésien.

Un intervalle de confiance à 95% représente une degré de "recouvrement" (*coverage*). Le "95%" fait référence à une propriété fréquentiste (i.e., sur le long-terme) de la procédure, mais ne fait pas référence au paramètre θ. Autrement dit, sur le long-terme, 95% des intervalles de confiance à 95% que l'on pourrait calculer (dans une réplication exacte de notre expérience) contiendraient la valeur du paramètre dans la population (i.e., la "vraie" valeur de θ). Cependant, nous ne pouvons pas dire qu'un intervalle de confiance en particulier a une probabilité de 95% de contenir la "vraie" valeur de θ… soit ce dernier contient la vraie valeur de θ, soit il ne la contient pas.

Facteur de Bayes

On compare deux modèles :

Facteur de Bayes

On compare deux modèles :

- $H_0 : \mu_1 = \mu_2 \rightarrow \delta = 0$

Facteur de Bayes

On compare deux modèles :

- $H_0 : \mu_1 = \mu_2 \rightarrow \delta = 0$
- $H_1 : \mu_1 \neq \mu_2 \rightarrow \delta \neq 0$

Facteur de Bayes

On compare deux modèles :

- $H_0 : \mu_1 = \mu_2 \rightarrow \delta = 0$
- $H_1 : \mu_1 \neq \mu_2 \rightarrow \delta \neq 0$

$$\underbrace{\frac{p(H_0|D)}{p(H_1|D)}}_{posterior\ odds} = \underbrace{\frac{p(D|H_0)}{p(D|H_1)}}_{Bayes\ factor} \times \underbrace{\frac{p(H_0)}{p(H_1)}}_{prior\ odds}$$

Facteur de Bayes

On compare deux modèles :

- $H_0 : \mu_1 = \mu_2 \rightarrow \delta = 0$
- $H_1 : \mu_1 \neq \mu_2 \rightarrow \delta \neq 0$

$$\underbrace{\frac{p(H_0|D)}{p(H_1|D)}}_{posterior\ odds} = \underbrace{\frac{p(D|H_0)}{p(D|H_1)}}_{Bayes\ factor} \times \underbrace{\frac{p(H_0)}{p(H_1)}}_{prior\ odds}$$

$$\text{evidence} = p(D|H) = \int p(\theta|H)p(D|\theta, H)\mathrm{d}\theta$$

Facteur de Bayes

On compare deux modèles :

- $H_0 : \mu_1 = \mu_2 \rightarrow \delta = 0$
- $H_1 : \mu_1 \neq \mu_2 \rightarrow \delta \neq 0$

$$\underbrace{\frac{p(H_0|D)}{p(H_1|D)}}_{posterior\ odds} = \underbrace{\frac{p(D|H_0)}{p(D|H_1)}}_{Bayes\ factor} \times \underbrace{\frac{p(H_0)}{p(H_1)}}_{prior\ odds}$$

$$\text{evidence} = p(D|H) = \int p(\theta|H)p(D|\theta, H)\mathrm{d}\theta$$

L'évidence en faveur d'un modèle correspond à la *vraisemblance marginale* d'un modèle (le dénominateur du théorème de Bayes), c'est à dire à la vraisemblance moyennée sur le prior… Ce qui fait du facteur de Bayes un ratio de vraisemblances, pondéré par (ou moyenné sur) le prior.

Facteur de Bayes, exemple d'application

On lance une pièce 100 fois et on essaye d'estimer la probabilité θ (le biais de la pièce) d'obtenir Face. On compare deux modèles qui diffèrent par leur prior sur θ.

Facteur de Bayes, exemple d'application

On lance une pièce 100 fois et on essaye d'estimer la probabilité θ (le biais de la pièce) d'obtenir Face. On compare deux modèles qui diffèrent par leur prior sur θ.

$$\mathcal{M}_1 : y_i \sim \text{Binomial}(n, \theta)$$
$$\theta \sim \text{Beta}(6, 10)$$

Facteur de Bayes, exemple d'application

On lance une pièce 100 fois et on essaye d'estimer la probabilité θ (le biais de la pièce) d'obtenir Face. On compare deux modèles qui diffèrent par leur prior sur θ.

$$\begin{aligned} \mathcal{M}_1 : y_i &\sim \text{Binomial}(n, \theta) \\ \theta &\sim \text{Beta}(6, 10) \end{aligned}$$

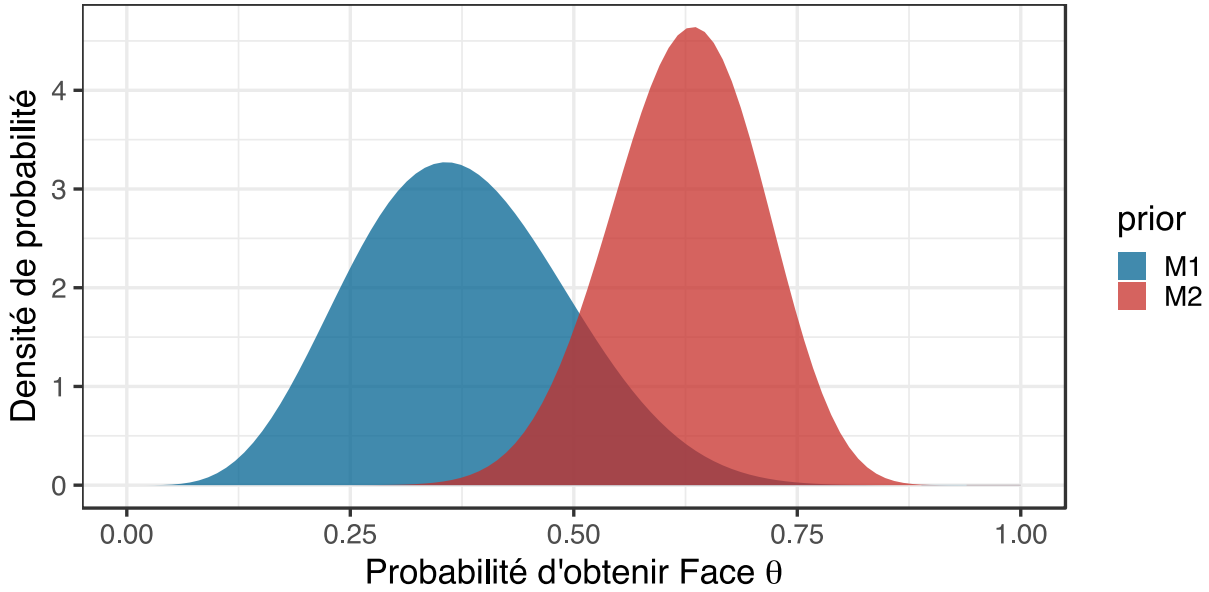
$$\begin{aligned} \mathcal{M}_2 : y_i &\sim \text{Binomial}(n, \theta) \\ \theta &\sim \text{Beta}(20, 12) \end{aligned}$$

Facteur de Bayes, exemple d'application

On lance une pièce 100 fois et on essaye d'estimer la probabilité θ (le biais de la pièce) d'obtenir Face. On compare deux modèles qui diffèrent par leur prior sur θ.

$$\mathcal{M}_1 : y_i \sim \text{Binomial}(n, \theta)$$
$$\theta \sim \text{Beta}(6, 10)$$

$$\mathcal{M}_2 : y_i \sim \text{Binomial}(n, \theta)$$
$$\theta \sim \text{Beta}(20, 12)$$



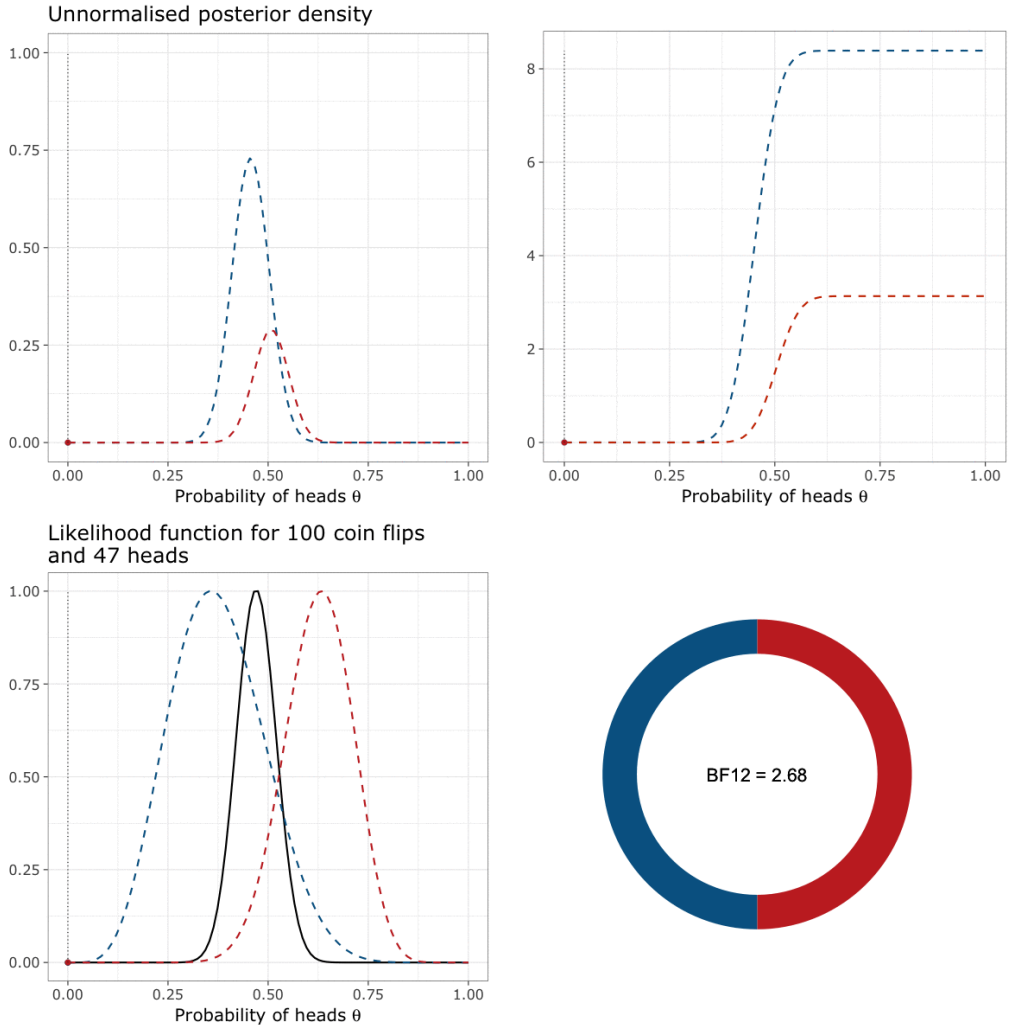
Facteur de Bayes, exemple d'application

$$\begin{aligned} \mathcal{M}_1 : y_i &\sim \text{Binomial}(n, \theta) \\ \theta &\sim \text{Beta}(6, 10) \end{aligned}$$

$$\begin{aligned} \mathcal{M}_2 : y_i &\sim \text{Binomial}(n, \theta) \\ \theta &\sim \text{Beta}(20, 12) \end{aligned}$$

$$\text{BF}_{12} = \frac{p(D|\mathcal{M}_1)}{p(D|\mathcal{M}_2)} = \frac{\int p(\theta|\mathcal{M}_1)p(D|\theta, \mathcal{M}_1)\text{d}\theta}{\int p(\theta|\mathcal{M}_2)p(D|\theta, \mathcal{M}_2)\text{d}\theta} = \frac{\int \text{Binomial}(n, \theta)\text{Beta}(6, 10)\text{d}\theta}{\int \text{Binomial}(n, \theta)\text{Beta}(20, 12)\text{d}\theta}$$

Facteur de Bayes, exemple d'application



Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

- H_0 : la précognition n'existe pas !

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

- H_0 : la précognition n'existe pas !
- H_1 : la précognition existe

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

- H_0 : la précognition n'existe pas !
- H_1 : la précognition existe

On fait une expérience et on calcule un $BF_{10} = 27$. Quelle est la probabilité a posteriori de H1 ?

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

- H_0 : la précognition n'existe pas !
- H_1 : la précognition existe

On fait une expérience et on calcule un $BF_{10} = 27$. Quelle est la probabilité a posteriori de H1 ?

Le facteur de Bayes est la nouvelle p-valeur

Attention à ne pas interpréter le BF comme un *posterior odds*…

Le BF est un facteur qui nous indique de combien nos *prior odds* doivent changer, au vue des données. Il ne nous dit pas quelle est l'hypothèse la plus probable, sachant les données (sauf si les *prior odds* sont 1:1).

- H_0 : la précognition n'existe pas !
- H_1 : la précognition existe

On fait une expérience et on calcule un $BF_{10} = 27$. Quelle est la probabilité a posteriori de H1 ?

$$\underbrace{\frac{p(H_1|D)}{p(H_0|D)}}_{posterior\ odds} = \underbrace{\frac{27}{1}}_{Bayes\ factor} \times \underbrace{\frac{1}{1000}}_{prior\ odds} = \frac{27}{1000} = 0.027$$

Facteur de Bayes - Test d'hypothèse nulle

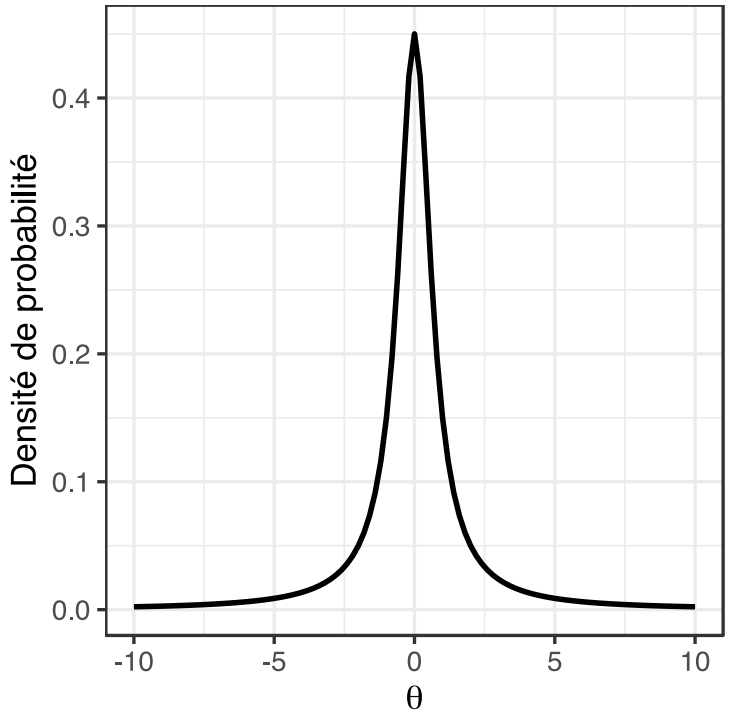
```
library(BayesFactor)

ttestBF(men, women)
```

```
Bayes factor analysis
--------------
[1] Alt., r=0.707 : 12.2944 ±0%

Against denominator:
  Null, mu1-mu2 = 0
---
Bayes factor type: BFindepSample, JZS
```

Prior JZS, en référence à Jeffreys (1961), et Zellner & Siow (1980).

Prior JZS

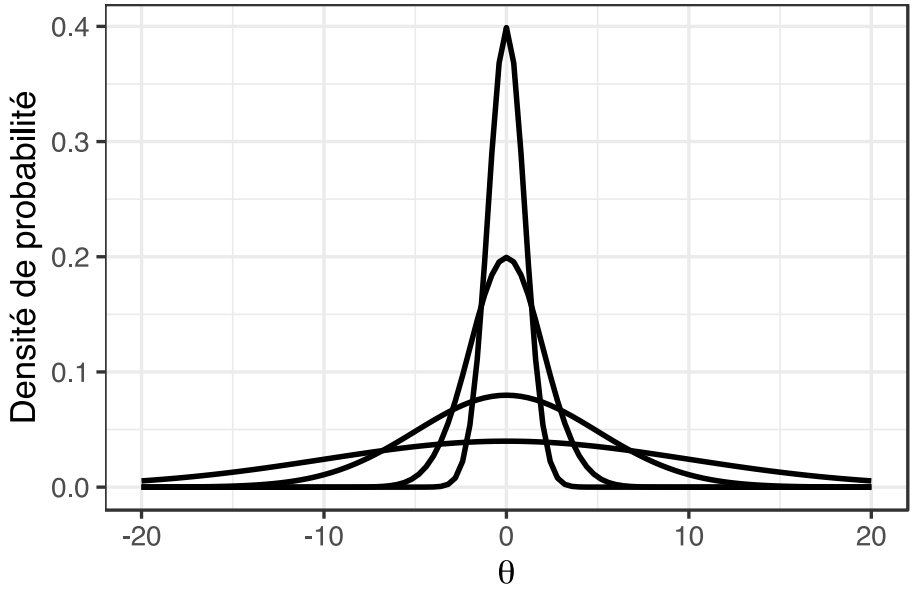
```
data.frame(x = c(-10, 10) ) %>%
    ggplot(aes(x = x) ) +
    stat_function(
        fun = dcauchy, args = list(location = 0, scale = sqrt(2) / 2), size = 1.5
        ) +
    theme_bw(base_size = 20) + labs(x = expression(theta), y = "Densité de probabilité")
```



Double sens

Dans le cadre bayésien, le terme de *prior* peut faire référence soit :

- À la probabilité *a priori* ou *a posteriori* d'un modèle (par rapport à un autre modèle), c'est à dire $\Pr(M_i)$. Voir ce blogpost.
- Aux prédictions du modèle, par exemple $\beta \sim \text{Normal}(0, 10)$. Voir ce blogpost.



Comparaison de modèles

Deux problèmes récurrents à éviter en modélisation : le sous-apprentissage et sur-apprentissage (*overffitting* et *underfitting*).

Comment s'en sortir ?

Comparaison de modèles

Deux problèmes récurrents à éviter en modélisation : le sous-apprentissage et sur-apprentissage (*overffitting* et *underfitting*). Comment s'en sortir ?

- Utiliser des priors pour *régulariser*, pour contraindre le posterior (i.e., accorder moins de poids à la vraisemblance)

Comparaison de modèles

Deux problèmes récurrents à éviter en modélisation : le sous-apprentissage et sur-apprentissage (*overffitting* et *underfitting*). Comment s'en sortir ?

- Utiliser des priors pour *régulariser*, pour contraindre le posterior (i.e., accorder moins de poids à la vraisemblance)
- Utiliser des critères d'information (e.g., AIC, WAIC)

Comparaison de modèles

Deux problèmes récurrents à éviter en modélisation : le sous-apprentissage et sur-apprentissage (*overffitting* et *underfitting*). Comment s'en sortir ?

- Utiliser des priors pour *régulariser*, pour contraindre le posterior (i.e., accorder moins de poids à la vraisemblance)
- Utiliser des critères d'information (e.g., AIC, WAIC)

Comparaison de modèles

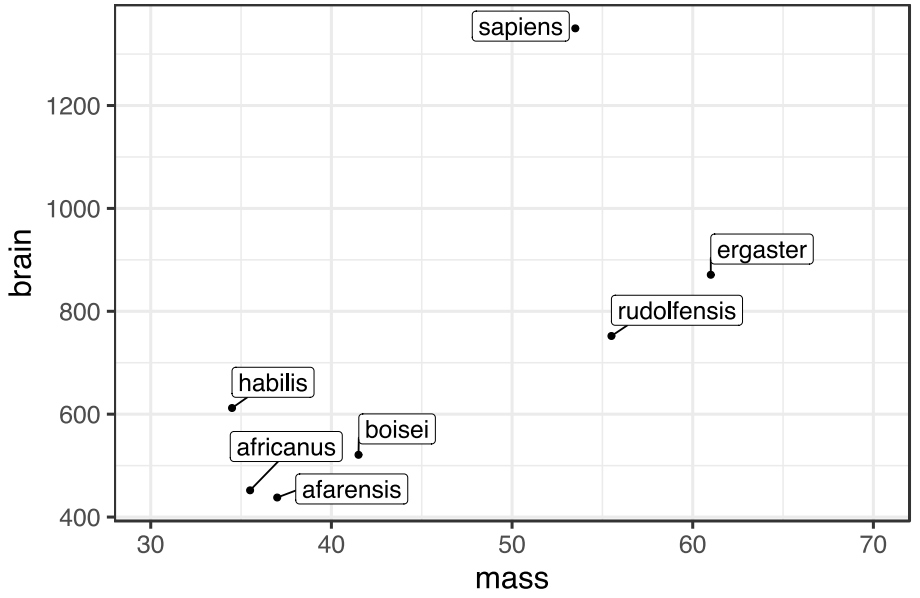
Deux problèmes récurrents à éviter en modélisation : le sous-apprentissage et sur-apprentissage (*overffitting* et *underfitting*). Comment s'en sortir ?

- Utiliser des priors pour *régulariser*, pour contraindre le posterior (i.e., accorder moins de poids à la vraisemblance)
- Utiliser des critères d'information (e.g., AIC, WAIC)

$$R^2 = \frac{\text{var(outcome)} - \text{var(residuals)}}{\text{var(outcome)}} = 1 - \frac{\text{var(residuals)}}{\text{var(outcome)}}$$

Overfitting

```
ppnames <- c("afarensis", "africanus", "habilis", "boisei",
        "rudolfensis", "ergaster", "sapiens")
brainvolcc <- c(438, 452, 612, 521, 752, 871, 1350)
masskg <- c(37.0, 35.5, 34.5, 41.5, 55.5, 61.0, 53.5)

d <- data.frame(species = ppnames, brain = brainvolcc, mass = masskg)

d %>%
    ggplot(aes(x = mass, y = brain, label = species) ) +
    geom_point() +
    ggrepel::geom_label_repel(hjust = 0, nudge_y = 50, size = 5) +
    theme_bw(base_size = 18) + xlim(30, 70)
```



Overfitting

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

```
[1] 0.490158
```

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

```
[1] 0.490158
```

```
mod1.2 <- lm(brain ~ mass + I(mass^2), data = d)
(var(d$brain) - var(residuals(mod1.2) ) ) / var(d$brain)
```

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

```
[1] 0.490158
```

```
mod1.2 <- lm(brain ~ mass + I(mass^2), data = d)
(var(d$brain) - var(residuals(mod1.2) ) ) / var(d$brain)
```

```
[1] 0.5359967
```

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

```
[1] 0.490158
```

```
mod1.2 <- lm(brain ~ mass + I(mass^2), data = d)
(var(d$brain) - var(residuals(mod1.2) ) ) / var(d$brain)
```

```
[1] 0.5359967
```

```
mod1.3 <- lm(brain ~ mass + I(mass^2) + I(mass^3), data = d)
(var(d$brain) - var(residuals(mod1.3) ) ) / var(d$brain)
```

Overfitting

```
mod1.1 <- lm(brain ~ mass, data = d)
(var(d$brain) - var(residuals(mod1.1) ) ) / var(d$brain)
```

```
[1] 0.490158
```

```
mod1.2 <- lm(brain ~ mass + I(mass^2), data = d)
(var(d$brain) - var(residuals(mod1.2) ) ) / var(d$brain)
```

```
[1] 0.5359967
```

```
mod1.3 <- lm(brain ~ mass + I(mass^2) + I(mass^3), data = d)
(var(d$brain) - var(residuals(mod1.3) ) ) / var(d$brain)
```

```
[1] 0.6797736
```


Overfitting

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

```
[1] 0.8144339
```

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

```
[1] 0.8144339
```

```
mod1.5 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5), data = d)
(var(d$brain) - var(residuals(mod1.5) ) ) / var(d$brain)
```

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

```
[1] 0.8144339
```

```
mod1.5 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5), data = d)
(var(d$brain) - var(residuals(mod1.5) ) ) / var(d$brain)
```

```
[1] 0.988854
```

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

```
[1] 0.8144339
```

```
mod1.5 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5), data = d)
(var(d$brain) - var(residuals(mod1.5) ) ) / var(d$brain)
```

```
[1] 0.988854
```

```
mod1.6 <- lm( brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5) + I(mass^6), data = d)
(var(d$brain) - var(residuals(mod1.6) ) ) / var(d$brain)
```

Overfitting

```
mod1.4 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4), data = d)
(var(d$brain) - var(residuals(mod1.4) ) ) / var(d$brain)
```

```
[1] 0.8144339
```

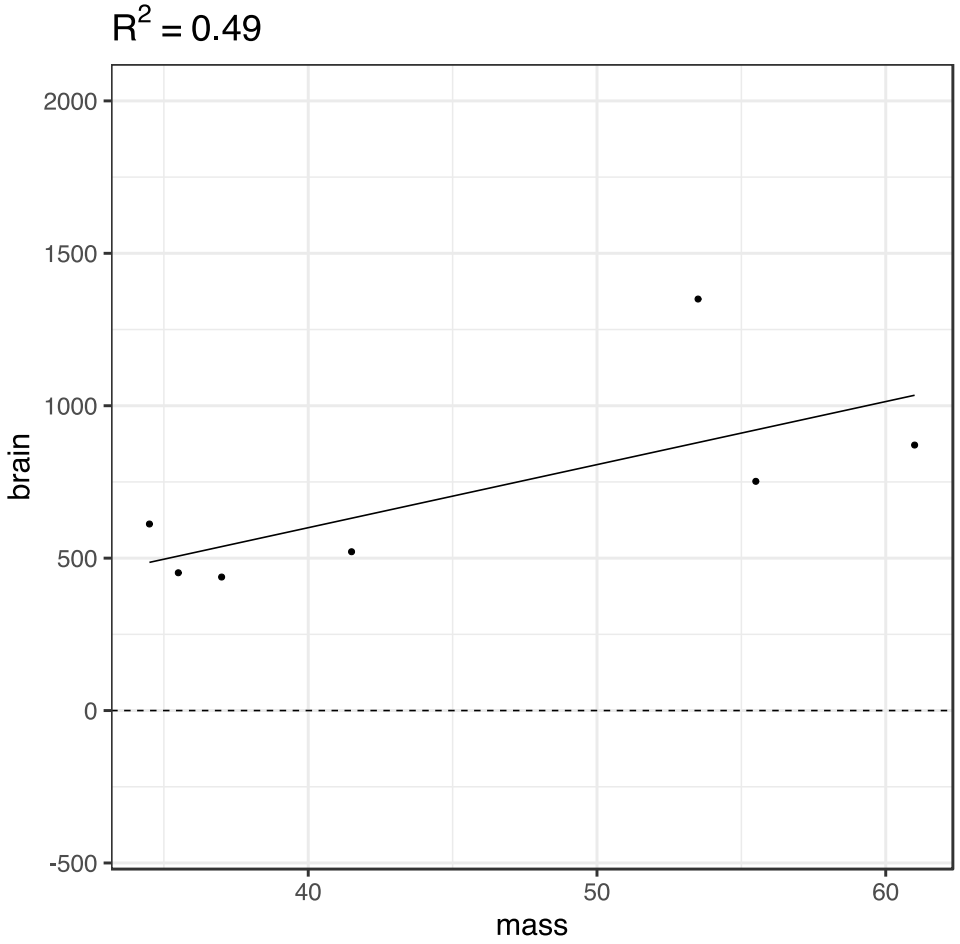
```
mod1.5 <- lm(brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5), data = d)
(var(d$brain) - var(residuals(mod1.5) ) ) / var(d$brain)
```

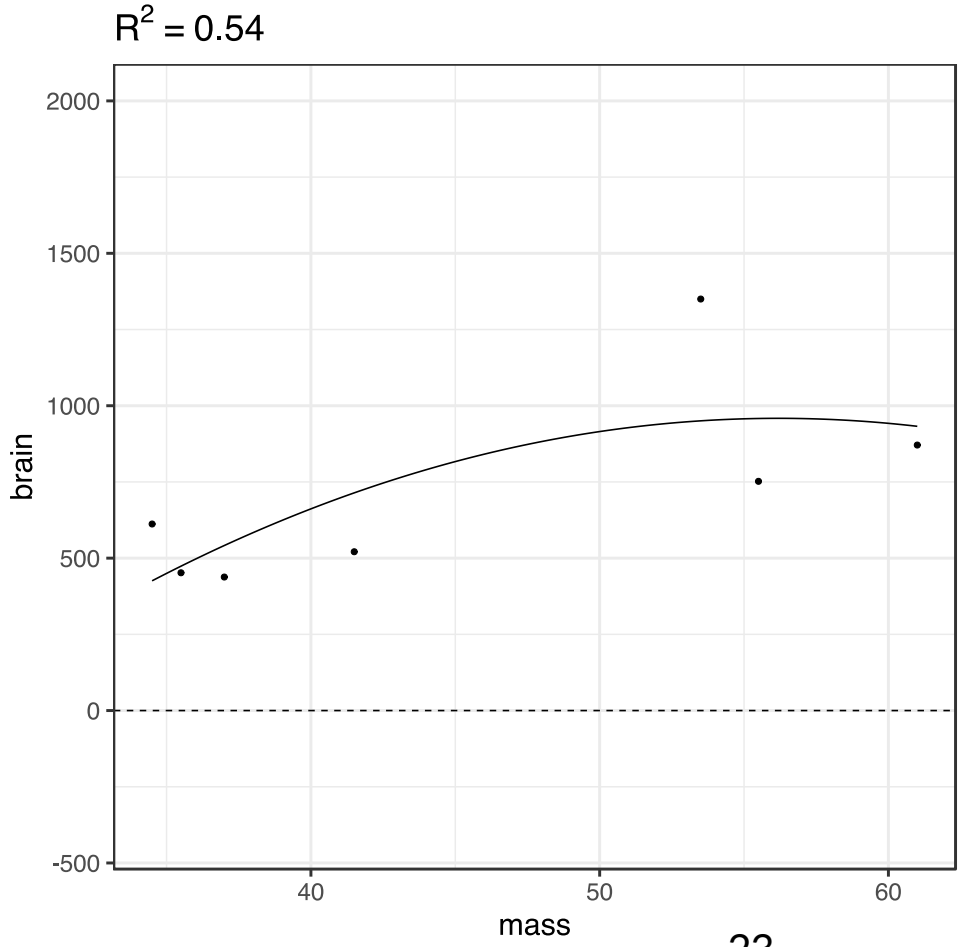
```
[1] 0.988854
```

```
mod1.6 <- lm( brain ~ mass + I(mass^2) + I(mass^3) + I(mass^4) +
    I(mass^5) + I(mass^6), data = d)
(var(d$brain) - var(residuals(mod1.6) ) ) / var(d$brain)
```

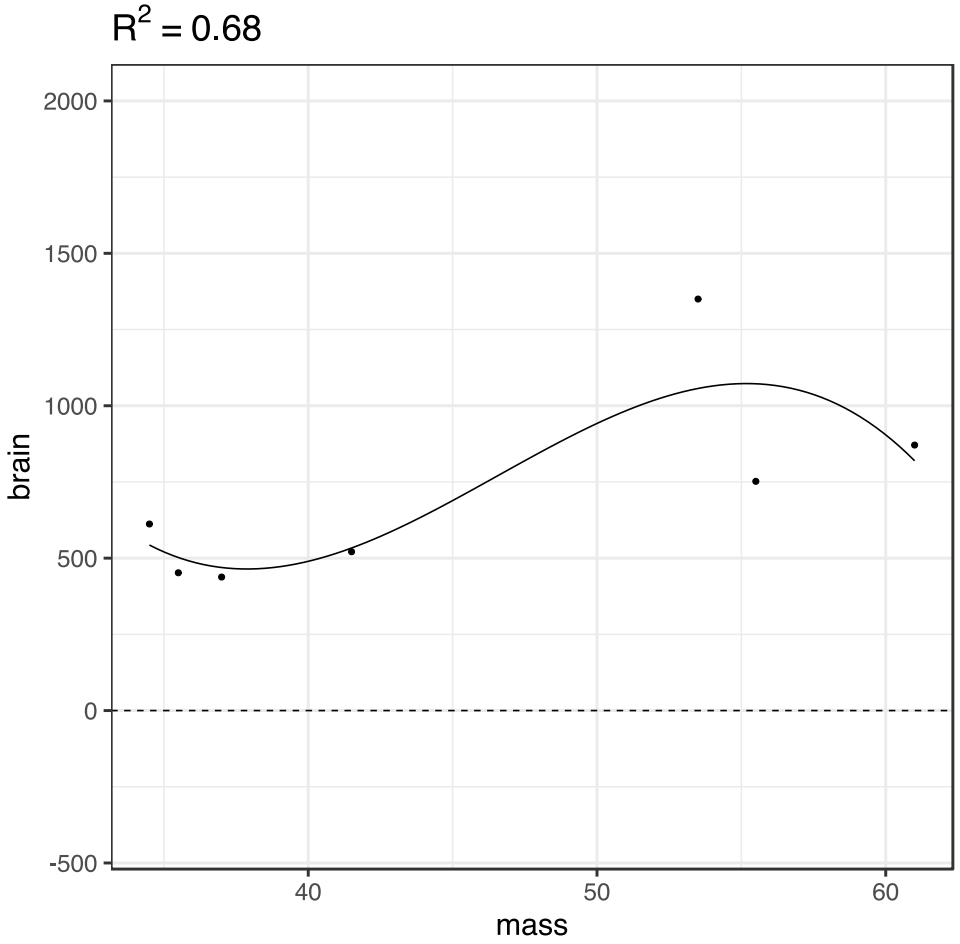
```
[1] 1
```

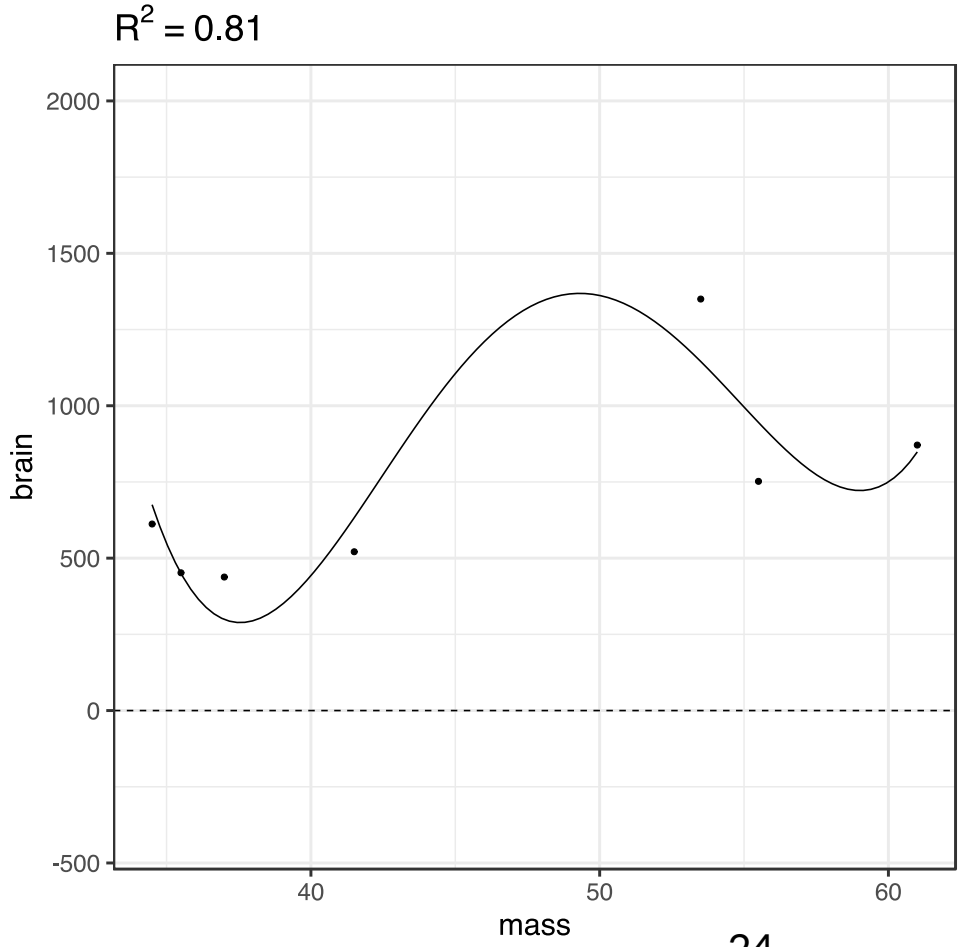

Overfitting



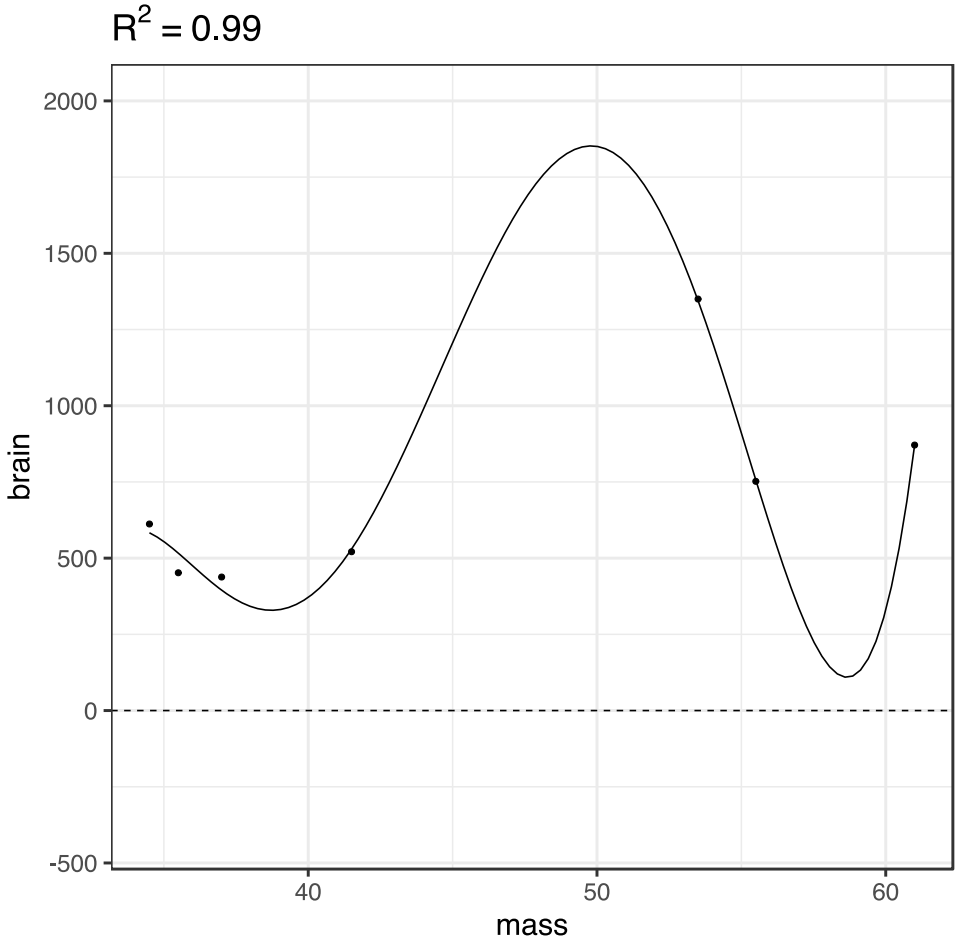


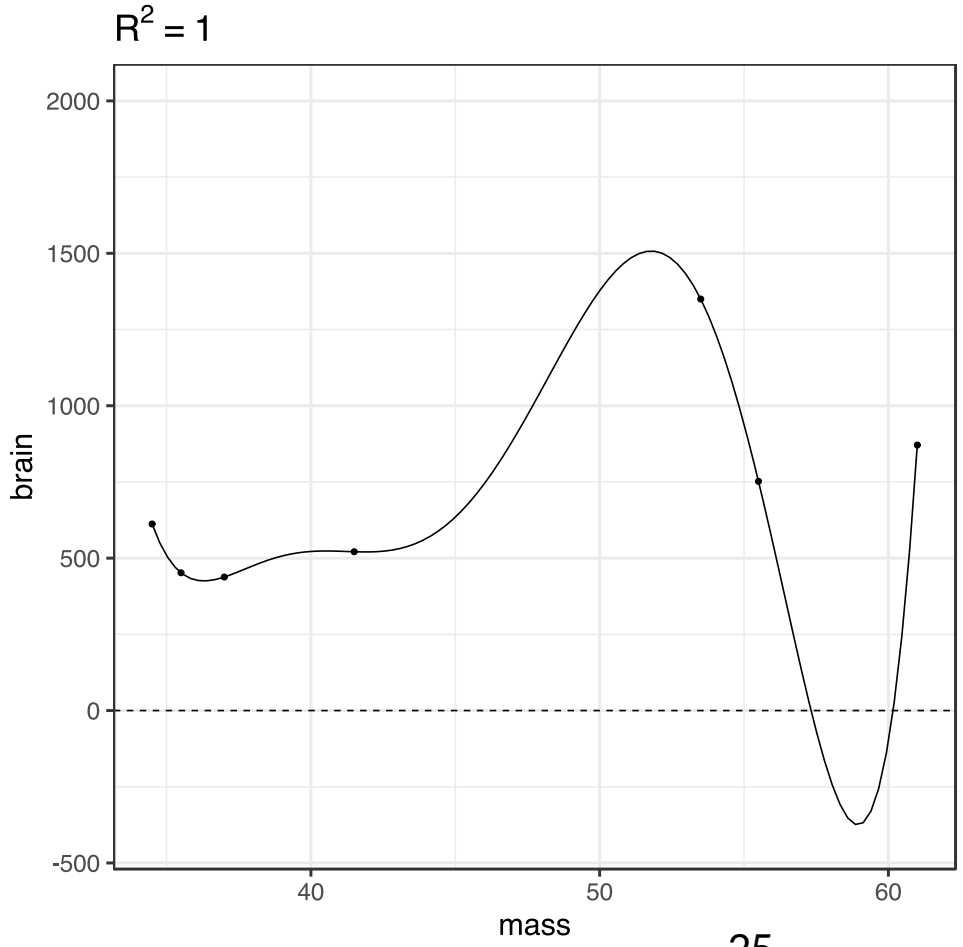
Overfitting





Overfitting

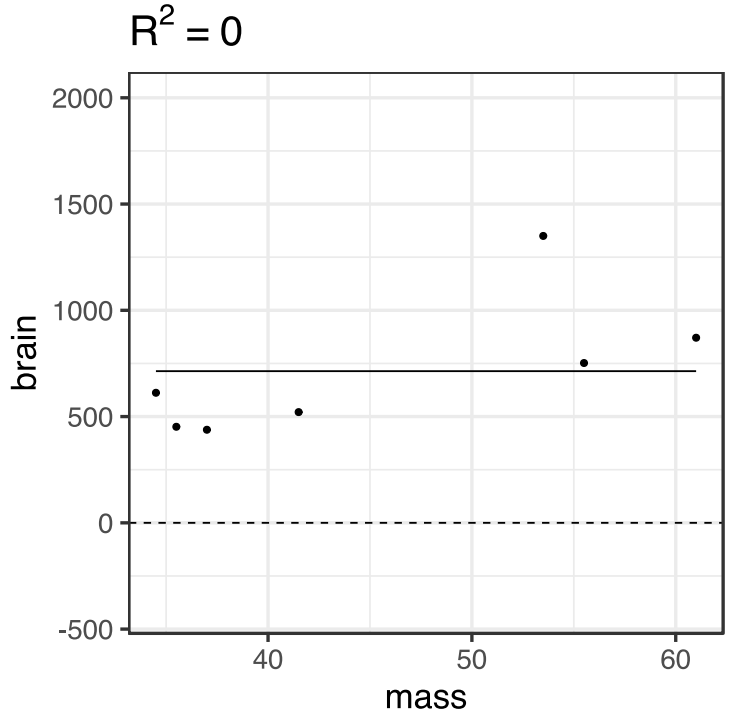




Underfitting

$$v_i \sim \text{Normal}(\mu_i, \sigma)$$
$$\mu_i = \alpha$$

```
mod1.7 <- lm(brain ~ 1, data = d)
```



Théorie de l'information

Il nous faut une autre mesure des capacités de prédiction pour évaluer nos modèles. Idéalement, on voudrait pouvoir mesurer la *distance* entre notre modèle et le *full model* (i.e., la "réalité", la nature)… mais on ne sait pas faire cela.

Théorie de l'information

Il nous faut une autre mesure des capacités de prédiction pour évaluer nos modèles. Idéalement, on voudrait pouvoir mesurer la *distance* entre notre modèle et le *full model* (i.e., la "réalité", la nature)… mais on ne sait pas faire cela.

Par contre, on peut mesure de combien notre incertitude est réduite en découvrant un *outcome* (une observation) supplémentaire. Cette réduction est la définition de l'**information**.

Théorie de l'information

Il nous faut une autre mesure des capacités de prédiction pour évaluer nos modèles. Idéalement, on voudrait pouvoir mesurer la *distance* entre notre modèle et le *full model* (i.e., la "réalité", la nature)… mais on ne sait pas faire cela.

Par contre, on peut mesure de combien notre incertitude est réduite en découvrant un *outcome* (une observation) supplémentaire. Cette réduction est la définition de l'**information**.

Mais il nous faut tout d'abord une mesure de l'incertitude (pour savoir si on l'a réduite, ou pas)… S'il existe n événements possibles, et que chaque événement i a pour probabilité p_i, alors une mesure de l'incertitude est donnée par l'entropie (de Shannon) H :

Théorie de l'information

Il nous faut une autre mesure des capacités de prédiction pour évaluer nos modèles. Idéalement, on voudrait pouvoir mesurer la *distance* entre notre modèle et le *full model* (i.e., la "réalité", la nature)… mais on ne sait pas faire cela.

Par contre, on peut mesure de combien notre incertitude est réduite en découvrant un *outcome* (une observation) supplémentaire. Cette réduction est la définition de l'**information**.

Mais il nous faut tout d'abord une mesure de l'incertitude (pour savoir si on l'a réduite, ou pas)… S'il existe n événements possibles, et que chaque événement i a pour probabilité p_i, alors une mesure de l'incertitude est donnée par l'entropie (de Shannon) H :

$$H(p) = -\mathbb{E}[\log(p_i)] = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log(p_i)$$

Théorie de l'information

Il nous faut une autre mesure des capacités de prédiction pour évaluer nos modèles. Idéalement, on voudrait pouvoir mesurer la *distance* entre notre modèle et le *full model* (i.e., la "réalité", la nature)… mais on ne sait pas faire cela.

Par contre, on peut mesure de combien notre incertitude est réduite en découvrant un *outcome* (une observation) supplémentaire. Cette réduction est la définition de l'**information**.

Mais il nous faut tout d'abord une mesure de l'incertitude (pour savoir si on l'a réduite, ou pas)… S'il existe n événements possibles, et que chaque événement i a pour probabilité p_i, alors une mesure de l'incertitude est donnée par l'entropie (de Shannon) H :

$$H(p) = -\mathbb{E}[\log(p_i)] = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log(p_i)$$

En d'autres termes, *l'incertitude contenue dans une distribution de probabilités est la log-probabilité moyenne d'un événement*.

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

```
p <- c(0.3, 0.7)
- sum(p * log(p) )
```

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

```
p <- c(0.3, 0.7)
- sum(p * log(p) )
```

```
[1] 0.6108643
```

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

```
p <- c(0.3, 0.7)
- sum(p * log(p) )
```

```
[1] 0.6108643
```

Imaginons que nous habitions à Abu Dhabi et que la probabilité qu'il y pleuve ou qu'il y fasse beau est de $p_1 = 0.01$ et $p_2 = 0.99$.

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

```
p <- c(0.3, 0.7)
- sum(p * log(p) )
```

```
[1] 0.6108643
```

Imaginons que nous habitions à Abu Dhabi et que la probabilité qu'il y pleuve ou qu'il y fasse beau est de $p_1 = 0.01$ et $p_2 = 0.99$.

```
p <- c(0.01, 0.99)
- sum(p * log(p) )
```

Incertitude

Exemple de prédiction météorologique. Si on imagine que la probabilité qu'il pleuve ou qu'il fasse beau (à Grenoble) est de $p_1 = 0.3$ et $p_2 = 0.7$.

Alors, $H(p) = -(p_1 \cdot \log(p_1) + p_2 \cdot \log(p_2)) \approx 0.61$.

```
p <- c(0.3, 0.7)
- sum(p * log(p) )
```

```
[1] 0.6108643
```

Imaginons que nous habitions à Abu Dhabi et que la probabilité qu'il y pleuve ou qu'il y fasse beau est de $p_1 = 0.01$ et $p_2 = 0.99$.

```
p <- c(0.01, 0.99)
- sum(p * log(p) )
```

```
[1] 0.05600153
```

Divergence

On a donc un moyen de quantifier l'incertitude. Comment utiliser cette mesure pour quantifier la distance entre notre modèle et la réalité ?

Divergence

On a donc un moyen de quantifier l'incertitude. Comment utiliser cette mesure pour quantifier la distance entre notre modèle et la réalité ?

Divergence : incertitude ajoutée par l'utilisation d'une distribution de probabilités pour décrire… une autre distribution de probabilités (Kullback-Leibler divergence, ou *entropie relative*).

Divergence

On a donc un moyen de quantifier l'incertitude. Comment utiliser cette mesure pour quantifier la distance entre notre modèle et la réalité ?

Divergence : incertitude ajoutée par l'utilisation d'une distribution de probabilités pour décrire… une autre distribution de probabilités (Kullback-Leibler divergence, ou *entropie relative*).

$$D_{KL}(p,q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Divergence

On a donc un moyen de quantifier l'incertitude. Comment utiliser cette mesure pour quantifier la distance entre notre modèle et la réalité ?

Divergence : incertitude ajoutée par l'utilisation d'une distribution de probabilités pour décrire… une autre distribution de probabilités (Kullback-Leibler divergence, ou *entropie relative*).

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

La divergence est la différence moyenne en log-probabilités entre la distribution cible (p) et le modèle (q).

Divergence

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Divergence

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Par exemple, supposons que la *véritable* distribution de nos événements (soleil vs pluie) soit $p_1 = 0.3$ et $p_2 = 0.7$. Si nous pensons plutôt que ces événements arrivent avec une probabilité $q_1 = 0.25$ et $q_2 = 0.75$, quelle quantité d'incertitude avons-nous ajoutée ?

Divergence

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Par exemple, supposons que la *véritable* distribution de nos événements (soleil vs pluie) soit $p_1 = 0.3$ et $p_2 = 0.7$. Si nous pensons plutôt que ces événements arrivent avec une probabilité $q_1 = 0.25$ et $q_2 = 0.75$, quelle quantité d'incertitude avons-nous ajoutée ?

```
p <- c(0.3, 0.7)
q <- c(0.25, 0.75)

sum(p * log(p / q) )
```

Divergence

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Par exemple, supposons que la *véritable* distribution de nos événements (soleil vs pluie) soit $p_1 = 0.3$ et $p_2 = 0.7$. Si nous pensons plutôt que ces événements arrivent avec une probabilité $q_1 = 0.25$ et $q_2 = 0.75$, quelle quantité d'incertitude avons-nous ajoutée ?

```
p <- c(0.3, 0.7)
q <- c(0.25, 0.75)

sum(p * log(p / q) )
```

```
[1] 0.006401457
```

Divergence

$$D_{KL}(p,q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Par exemple, supposons que la *véritable* distribution de nos événements (soleil vs pluie) soit $p_1 = 0.3$ et $p_2 = 0.7$. Si nous pensons plutôt que ces événements arrivent avec une probabilité $q_1 = 0.25$ et $q_2 = 0.75$, quelle quantité d'incertitude avons-nous ajoutée ?

```
p <- c(0.3, 0.7)
q <- c(0.25, 0.75)

sum(p * log(p / q) )
```

```
[1] 0.006401457
```

```
sum(q * log(q / p) ) # NB : La divergence n'est pas symétrique...
```

Divergence

$$D_{KL}(p, q) = \sum_i p_i \Big(\log(p_i) - \log(q_i) \Big) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right)$$

Par exemple, supposons que la *véritable* distribution de nos événements (soleil vs pluie) soit $p_1 = 0.3$ et $p_2 = 0.7$. Si nous pensons plutôt que ces événements arrivent avec une probabilité $q_1 = 0.25$ et $q_2 = 0.75$, quelle quantité d'incertitude avons-nous ajoutée ?

```
p <- c(0.3, 0.7)
q <- c(0.25, 0.75)

sum(p * log(p / q) )
```

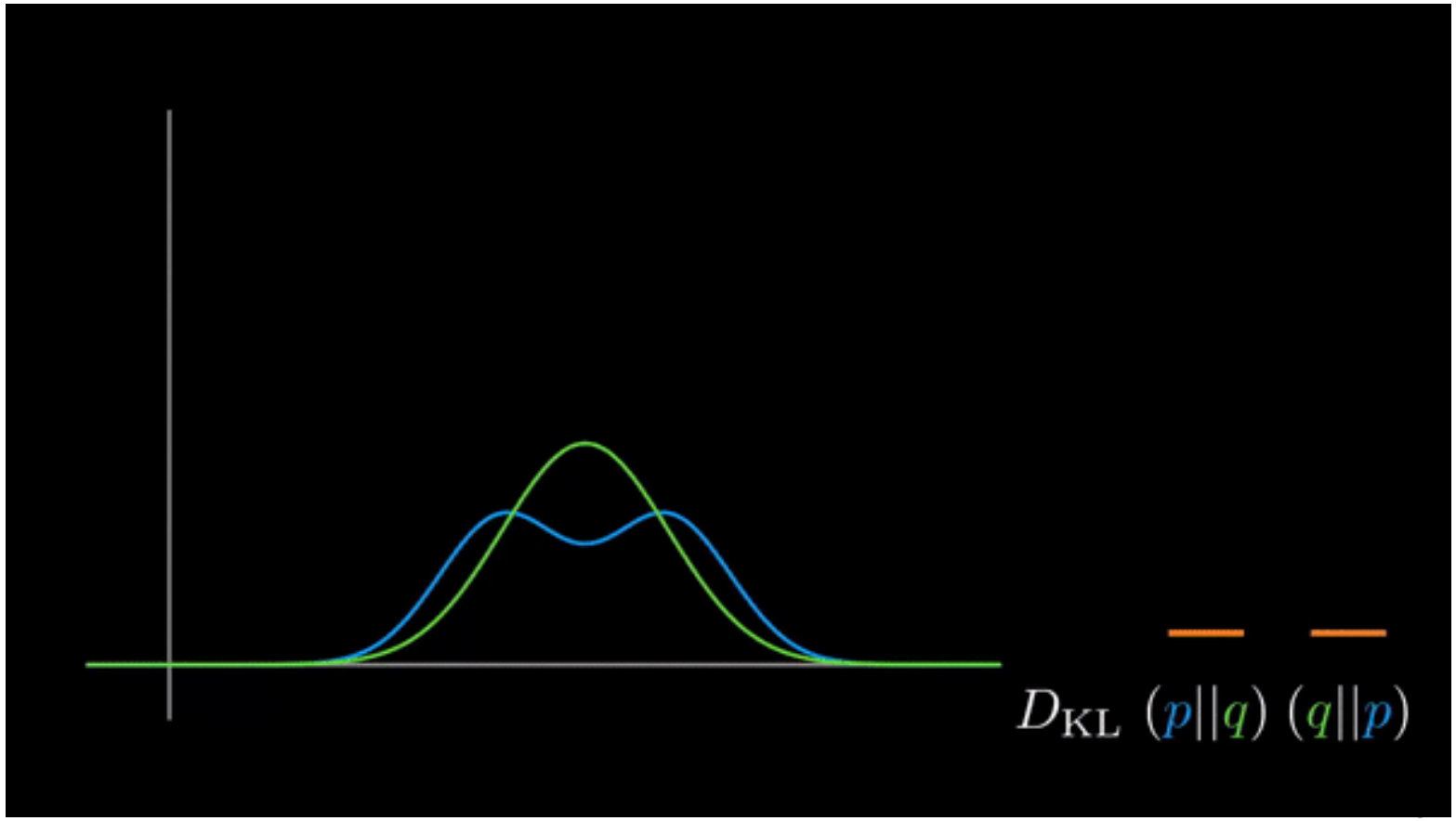
```
[1] 0.006401457
```

```
sum(q * log(q / p) ) # NB : La divergence n'est pas symétrique...
```

```
[1] 0.006164264
```

Divergence

La divergence n'est pas symétrique (ce n'est pas une distance)…



Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

```
sum(p * (log(q) ) )
```

Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

```
sum(p * (log(q) ) )
```

La **Divergence** est définie comme l'entropie additionnelle ajoutée en utilisant q pour décrire p.

Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

```
sum(p * (log(q) ) )
```

La **Divergence** est définie comme l'entropie additionnelle ajoutée en utilisant q pour décrire p.

$$
\begin{aligned}
D_{KL}(p, q) &= H(p, q) - H(p) \\
&= -\sum_i p_i \log(q_i) - \Big(-\sum_i p_i \log(p_i) \Big) \\
&= -\sum_i p_i \Big(\log(q_i) - \log(p_i) \Big)
\end{aligned}
$$

Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

```
sum(p * (log(q) ) )
```

La **Divergence** est définie comme l'entropie additionnelle ajoutée en utilisant q pour décrire p.

$$
\begin{aligned}
D_{KL}(p, q) &= H(p, q) - H(p) \\
&= -\sum_i p_i \log(q_i) - \Big(-\sum_i p_i \log(p_i) \Big) \\
&= -\sum_i p_i \Big(\log(q_i) - \log(p_i) \Big)
\end{aligned}
$$

```
- sum (p * (log(q) - log(p) ) )
```

Entropie croisée et divergence

Entropie croisée : $H(p, q) = \sum_i p_i \log(q_i)$

```
sum(p * (log(q) ) )
```

La **Divergence** est définie comme l'entropie additionnelle ajoutée en utilisant q pour décrire p.

$$
\begin{aligned}
D_{KL}(p, q) &= H(p, q) - H(p) \\
&= -\sum_i p_i \log(q_i) - \Big(-\sum_i p_i \log(p_i) \Big) \\
&= -\sum_i p_i \Big(\log(q_i) - \log(p_i) \Big)
\end{aligned}
$$

```
- sum (p * (log(q) - log(p) ) )
```

```
[1] 0.006401457
```


Vers la déviance...

OK, mais nous ne connaissons pas la distribution *target* (la réalité), à quoi cela peut donc nous servir ?

Vers la déviance...

OK, mais nous ne connaissons pas la distribution *target* (la réalité), à quoi cela peut donc nous servir ?

Astuce : si nous comparons deux modèles, q et r, pour approximer p, nous allons comparer leurs divergences… Et donc $\mathbb{E}[\log(p_i)]$ sera la même quantité pour les deux modèles !

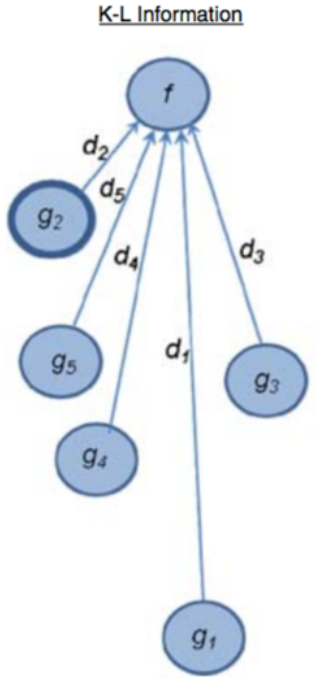
Vers la déviance...

OK, mais nous ne connaissons pas la distribution *target* (la réalité), à quoi cela peut donc nous servir ?

Astuce : si nous comparons deux modèles, q et r, pour approximer p, nous allons comparer leurs divergences… Et donc $\mathbb{E}[\log(p_i)]$ sera la même quantité pour les deux modèles !

Vers la déviance...

On peut donc utiliser $\mathbb{E}[\log(q_i)]$ et $\mathbb{E}[\log(r_i)]$ comme estimateurs de la distance *relative* entre chaque modèle et notre distribution cible. On a donc seulement besoin de la *log-probabilité moyenne des modèles*. Comme on ne connaît pas la distribution cible, cela veut dire qu'on ne peut pas interpréter cette quantité en termes absolus mais seulement en termes relatifs. Ce qui nous intéresse c'est $\mathbb{E}[\log(q_i)] - \mathbb{E}[\log(r_i)]$.



Déviance

Pour approximer la valeur de $\mathbb{E}[\log(q_i)]$, on peut utiliser la déviance d'un modèle, qui est une mesure du fit *relatif* du modèle.

Déviance

Pour approximer la valeur de $\mathbb{E}[\log(q_i)]$, on peut utiliser la déviance d'un modèle, qui est une mesure du fit *relatif* du modèle.

$$D(q) = -2 \sum_i \log(q_i)$$

Déviance

Pour approximer la valeur de $\mathbb{E}[\log(q_i)]$, on peut utiliser la déviance d'un modèle, qui est une mesure du fit *relatif* du modèle.

$$D(q) = -2 \sum_i \log(q_i)$$

où i indice chaque observation et q_i est la *vraisemblance* de chaque observation.

Déviance

Pour approximer la valeur de $\mathbb{E}[\log(q_i)]$, on peut utiliser la déviance d'un modèle, qui est une mesure du fit *relatif* du modèle.

$$D(q) = -2 \sum_i \log(q_i)$$

où i indice chaque observation et q_i est la *vraisemblance* de chaque observation.

```
d$mass.s <- scale(d$mass)
mod1.8 <- lm(brain ~ mass.s, data = d)

-2 * logLik(mod1.8) # calcul de la déviance
```

Déviance

Pour approximer la valeur de $\mathbb{E}[\log(q_i)]$, on peut utiliser la déviance d'un modèle, qui est une mesure du fit *relatif* du modèle.

$$D(q) = -2 \sum_i \log(q_i)$$

où i indice chaque observation et q_i est la *vraisemblance* de chaque observation.

```
d$mass.s <- scale(d$mass)
mod1.8 <- lm(brain ~ mass.s, data = d)

-2 * logLik(mod1.8) # calcul de la déviance
```

```
'log Lik.' 94.92499 (df=3)
```

Déviance

Déviance

```
# paramètres estimés (intercept et pente)

alpha <- coef(mod1.8)[1]
beta <- coef(mod1.8)[2]

# calcul de la log-vraisemblance

ll <- sum(dnorm(
    d$brain,
    mean = alpha + beta * d$mass.s,
    sd = sigma(mod1.8),
    log = TRUE)
    )

# calcul de la déviance

(-2) * ll
```

Déviance

```
# paramètres estimés (intercept et pente)

alpha <- coef(mod1.8)[1]
beta <- coef(mod1.8)[2]

# calcul de la log-vraisemblance

ll <- sum(dnorm(
    d$brain,
    mean = alpha + beta * d$mass.s,
    sd = sigma(mod1.8),
    log = TRUE)
    )

# calcul de la déviance

(-2) * ll
```

```
[1] 95.2803
```

Log-pointwise-predictive density

Les fréquentistes aiment bien multiplier le log-score par -2 car la différence de deux déviances suit une loi de χ^2, ce qui est utile pour tester l'hypothèse nulle. Mais sans besoin de tester l'hypothèse nulle (avec une forme prédéfinie), on peut très bien travailler directement avec le log-score $S(q) = \sum_i \log(q_i)$, qu'on traite comme une estimation de $\mathbb{E}[\log(q_i)]$.

Log-pointwise-predictive density

Les fréquentistes aiment bien multiplier le log-score par -2 car la différence de deux déviances suit une loi de χ^2, ce qui est utile pour tester l'hypothèse nulle. Mais sans besoin de tester l'hypothèse nulle (avec une forme prédéfinie), on peut très bien travailler directement avec le log-score $S(q) = \sum_i \log(q_i)$, qu'on traite comme une estimation de $\mathbb{E}[\log(q_i)]$.

On peut calculer $S(q)$ sur toute la distribution postérieure, ce qui donne la version bayésienne du log-score, la **log-pointwise-predictive density** :

Log-pointwise-predictive density

Les fréquentistes aiment bien multiplier le log-score par -2 car la différence de deux déviances suit une loi de χ^2, ce qui est utile pour tester l'hypothèse nulle. Mais sans besoin de tester l'hypothèse nulle (avec une forme prédéfinie), on peut très bien travailler directement avec le log-score $S(q) = \sum_i \log(q_i)$, qu'on traite comme une estimation de $\mathbb{E}[\log(q_i)]$.

On peut calculer $S(q)$ sur toute la distribution postérieure, ce qui donne la version bayésienne du log-score, la **log-pointwise-predictive density** :

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

Log-pointwise-predictive density

Les fréquentistes aiment bien multiplier le log-score par -2 car la différence de deux déviances suit une loi de χ^2, ce qui est utile pour tester l'hypothèse nulle. Mais sans besoin de tester l'hypothèse nulle (avec une forme prédéfinie), on peut très bien travailler directement avec le log-score $S(q) = \sum_i \log(q_i)$, qu'on traite comme une estimation de $\mathbb{E}[\log(q_i)]$.

On peut calculer $S(q)$ sur toute la distribution postérieure, ce qui donne la version bayésienne du log-score, la **log-pointwise-predictive density** :

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

Où S est le nombre d'échantillons et Θ_s est le s-ième ensemble de valeurs de paramètres échantillonnés de la distribution postérieure.

In-sample and out-of-sample

La déviance a le même problème que le R^2, lorsqu'elle est calculée sur l'échantillon observé. Dans ce cas, on l'appelle déviance **in-sample**.

In-sample and out-of-sample

La déviance a le même problème que le R^2, lorsqu'elle est calculée sur l'échantillon observé. Dans ce cas, on l'appelle déviance **in-sample**.

Si on est intéressé par les capacités de prédiction de notre modèle, nous pouvons calculer la déviance du modèle sur de nouvelles données… qu'on appellera dans ce cas déviance **out-of-sample**. Cela revient à se demander si notre modèle est performant pour prédire de nouvelles données.

In-sample and out-of-sample

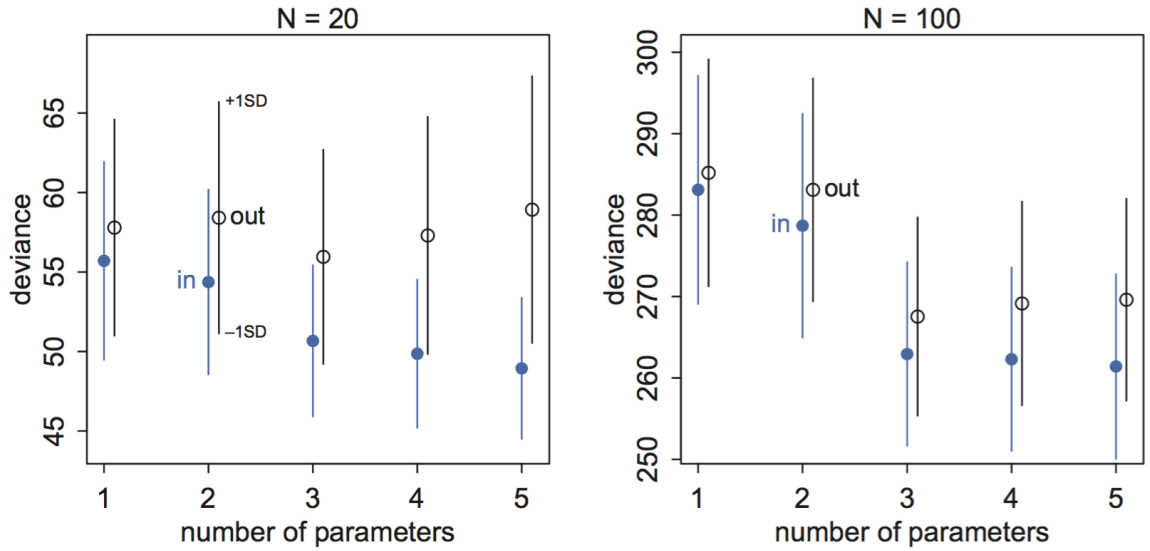
La déviance a le même problème que le R^2, lorsqu'elle est calculée sur l'échantillon observé. Dans ce cas, on l'appelle déviance **in-sample**.

Si on est intéressé par les capacités de prédiction de notre modèle, nous pouvons calculer la déviance du modèle sur de nouvelles données… qu'on appellera dans ce cas déviance **out-of-sample**. Cela revient à se demander si notre modèle est performant pour prédire de nouvelles données.

Imaginons que nous disposions d'un échantillon de taille N, que nous appellerons échantillon d'apprentissage (*training*). Nous pouvons calculer la déviance du modèle sur cet échantillon (D_{train} ou D_{in}). Si nous acquérons ensuite un nouvel échantillon de taille N issu du même processus de génération de données (que nous appellerons échantillon de test), nous pouvons calculer une déviance sur ce nouvel échantillon, en utilisant les paramètres estimés avec l'échantillon d'entraînement (que nous appellerons D_{test} ou D_{out}).

In sample and out of sample deviance

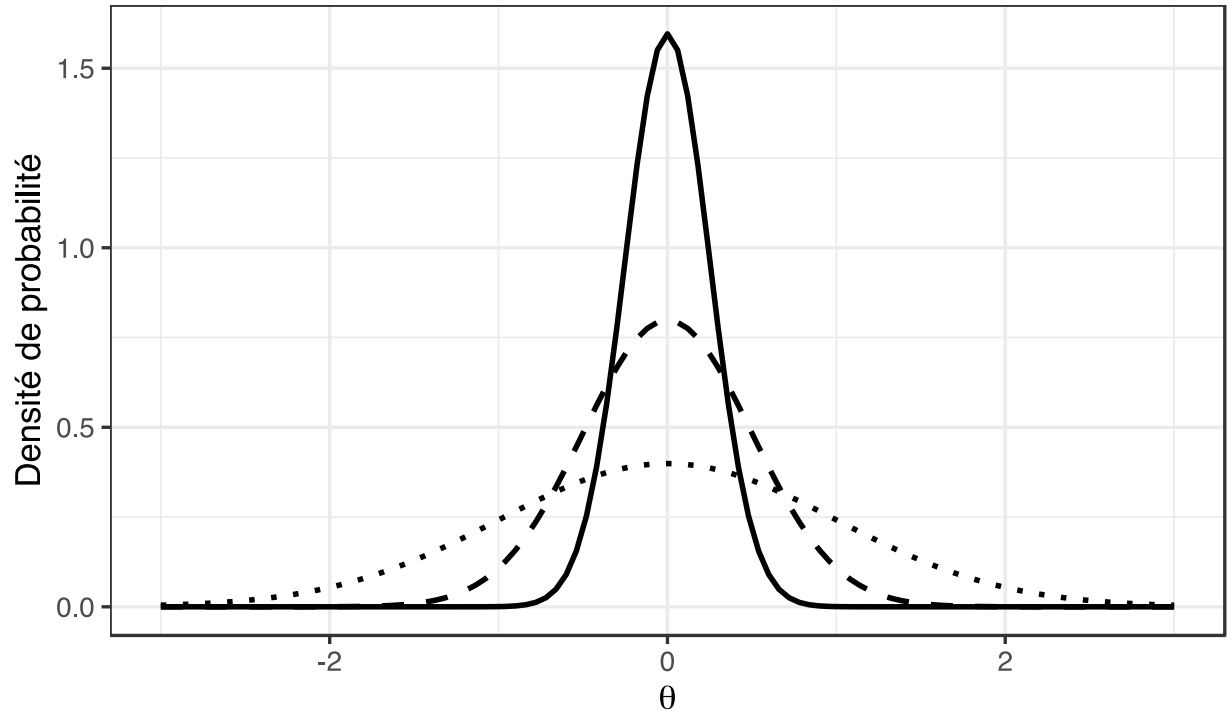
$$y_i \sim \text{Normal}(\mu_i, 1)$$
$$\mu_i = (0.15)x_{1,i} - (0.4)x_{2,i}$$



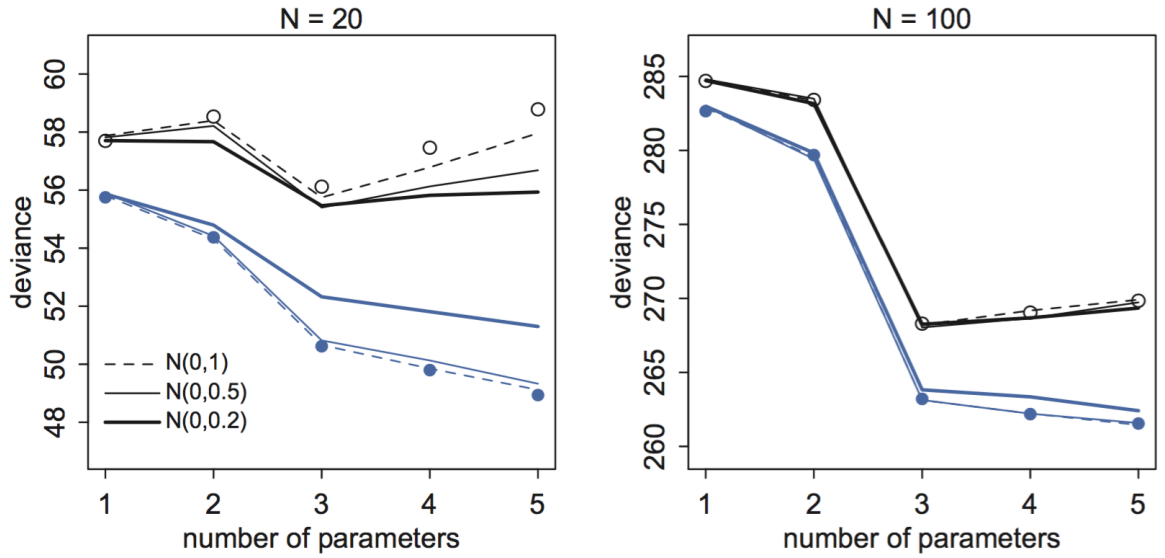
On a réalisé ce processus 10.000 fois pour cinq modèles de régression linéaire de complexité croissante. Les points bleus représentent la déviance calculée sur l'échantillon d'apprentissage et les points noirs la déviance calculée sur l'échantillon de test.

Régularisation

Une autre manière de lutter contre l'*overfitting* est d'utiliser des priors *sceptiques* qui vont venir ralentir l'apprentissage réalisé sur les données (i.e., accorder plus de poids au prior).

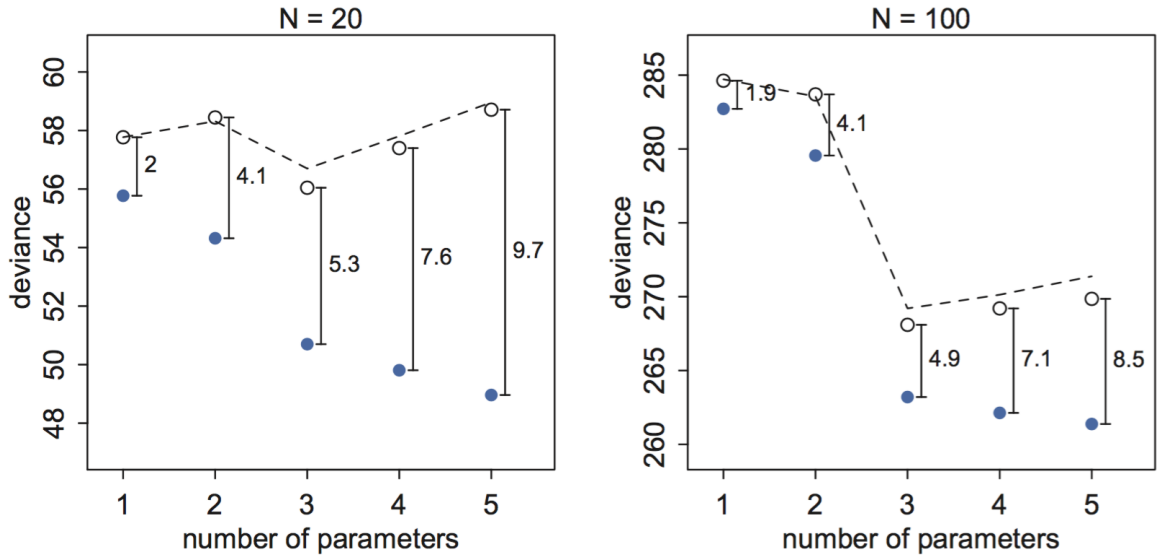


Régularisation



Comment décider de la précision du prior ? Est-ce que le prior est "assez" régularisateur ou pas ? On peut diviser le jeu de données en deux parties (*training* et *test*) afin de choisir le prior qui produit la déviance *out-of-sample* la plus faible. On appelle cette stratégie la *validation croisée* (cross-validation).

Critères d'information



On mesure ici la différence entre la déviance *in-sample* (en bleu) et la déviance *out-of-sample* (en noir). On remarque que la déviance *out-of-sample* est presque exactement égale à la déviance *in-sample*, plus deux fois le nombre de paramètres du modèle…

Akaike information criterion

L'AIC fournit une approximation de la déviance *out-of-sample*:

Akaike information criterion

L'AIC fournit une approximation de la déviance *out-of-sample*:

$$\text{AIC} = D_{\text{train}} + 2p = -2\text{lppd} + 2p \approx D_{\text{test}}$$

Akaike information criterion

L'AIC fournit une approximation de la déviance *out-of-sample*:

$$\text{AIC} = D_{\text{train}} + 2p = -2\text{lppd} + 2p \approx D_{\text{test}}$$

où p est le nombre de paramètres libres (i.e., à estimer) dans le modèle. L'AIC donne donc une approximation des capacités de prédiction du modèle.

Akaike information criterion

L'AIC fournit une approximation de la déviance *out-of-sample*:

$$\text{AIC} = D_{\text{train}} + 2p = -2\text{lppd} + 2p \approx D_{\text{test}}$$

où p est le nombre de paramètres libres (i.e., à estimer) dans le modèle. L'AIC donne donc une approximation des capacités de prédiction du modèle.

Akaike information criterion

L'AIC fournit une approximation de la déviance *out-of-sample*:

$$\text{AIC} = D_{\text{train}} + 2p = -2\text{lppd} + 2p \approx D_{\text{test}}$$

où p est le nombre de paramètres libres (i.e., à estimer) dans le modèle. L'AIC donne donc une approximation des capacités de prédiction du modèle.

NB : l'AIC fonctionne bien uniquement quand le nombre d'observations N est largement supérieur au nombre de paramètres p. Dans le cas contraire, on utilise plutôt l'AICc (*corrected AIC*, voir Burnham & Anderson, 2002; 2004).

Deviance information criterion

Une autre condition de l'AIC est que les priors soient plats (i.e., peu informatifs) ou dépassés par la vraisemblance (on a beaucoup de données). Le DIC est un indice qui ne requiert pas cette condition, en s'accommodant de priors informatifs.

Deviance information criterion

Une autre condition de l'AIC est que les priors soient plats (i.e., peu informatifs) ou dépassés par la vraisemblance (on a beaucoup de données). Le DIC est un indice qui ne requiert pas cette condition, en s'accommodant de priors informatifs.

Le DIC est calculé à partir de la distribution a posteriori de la déviance D calculée sur l'échantillon d'apprentissage (i.e., D_{train}).

Deviance information criterion

Une autre condition de l'AIC est que les priors soient plats (i.e., peu informatifs) ou dépassés par la vraisemblance (on a beaucoup de données). Le DIC est un indice qui ne requiert pas cette condition, en s'accommodant de priors informatifs.

Le DIC est calculé à partir de la distribution a posteriori de la déviance D calculée sur l'échantillon d'apprentissage (i.e., D_{train}).

$$\text{DIC} = \bar{D} + (\bar{D} - \hat{D}) = \bar{D} + p_D$$

Deviance information criterion

Une autre condition de l'AIC est que les priors soient plats (i.e., peu informatifs) ou dépassés par la vraisemblance (on a beaucoup de données). Le DIC est un indice qui ne requiert pas cette condition, en s'accommodant de priors informatifs.

Le DIC est calculé à partir de la distribution a posteriori de la déviance D calculée sur l'échantillon d'apprentissage (i.e., D_{train}).

$$\text{DIC} = \bar{D} + (\bar{D} - \hat{D}) = \bar{D} + p_D$$

où $\bar{D}$ est la moyenne de la distribution a posteriori D calculée pour chaque valeur de paramètre échantillonnée, et $\hat{D}$ la déviance calculée à la moyenne de la distribution a posteriori. La différence $\bar{D} - \hat{D} = p_D$ est analogue au nombre de paramètres utilisé dans le calcul de l'AIC (en cas de prior plat, cette différence revient à compter le nombre de paramètres).

Widely applicable information criterion

Une condition d'application de l'AIC et du DIC est que la distribution a posteriori soit une distribution gaussienne multivariée. Le WAIC relâche cette condition. Il est plus précis tout en étant souvent plus précis que le DIC.

Widely applicable information criterion

Une condition d'application de l'AIC et du DIC est que la distribution a posteriori soit une distribution gaussienne multivariée. Le WAIC relâche cette condition. Il est plus précis tout en étant souvent plus précis que le DIC.

Un aspect important du WAIC est qu'il est dit *pointwise*, c'est à dire qu'il considère l'imprécision de prédiction point par point (donnée par donnée), indépendamment pour chaque observation.

Widely applicable information criterion

Une condition d'application de l'AIC et du DIC est que la distribution a posteriori soit une distribution gaussienne multivariée. Le WAIC relâche cette condition. Il est plus précis tout en étant souvent plus précis que le DIC.

Un aspect important du WAIC est qu'il est dit *pointwise*, c'est à dire qu'il considère l'imprécision de prédiction point par point (donnée par donnée), indépendamment pour chaque observation.

On va commencer par calculer la *log-pointwise-predictive-density* (*lppd*), définie de la manière suivante :

Widely applicable information criterion

Une condition d'application de l'AIC et du DIC est que la distribution a posteriori soit une distribution gaussienne multivariée. Le WAIC relâche cette condition. Il est plus précis tout en étant souvent plus précis que le DIC.

Un aspect important du WAIC est qu'il est dit *pointwise*, c'est à dire qu'il considère l'imprécision de prédiction point par point (donnée par donnée), indépendamment pour chaque observation.

On va commencer par calculer la *log-pointwise-predictive-density* (*lppd*), définie de la manière suivante :

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

Widely applicable information criterion

Une condition d'application de l'AIC et du DIC est que la distribution a posteriori soit une distribution gaussienne multivariée. Le WAIC relâche cette condition. Il est plus précis tout en étant souvent plus précis que le DIC.

Un aspect important du WAIC est qu'il est dit *pointwise*, c'est à dire qu'il considère l'imprécision de prédiction point par point (donnée par donnée), indépendamment pour chaque observation.

On va commencer par calculer la *log-pointwise-predictive-density* (*lppd*), définie de la manière suivante :

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

En Français : *la log-densité prédictive point par point est la somme du log de la vraisemblance moyenne de chaque observation*. Il s'agit de l'analogue point par point de la déviance, moyenné sur toute la distribution postérieure.

Widely applicable information criterion

```
library(brms)
data(cars)

priors <- c(
  set_prior("normal(0, 100)", class = "Intercept"),
  set_prior("normal(0, 10)", class = "b"),
  set_prior("exponential(0.1)", class = "sigma")
  )

mod1 <- brm(
  dist ~ 1 + speed,
  prior = priors,
  data = cars
  )
```


Widely applicable information criterion

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

Widely applicable information criterion

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

```
n_obs <- nrow(cars) # number of observations

ll <-
  log_lik(mod1) %>% # pointwise log-likelihood (S rows * N observations)
  data.frame() %>% # converts to dataframe
  set_names(c(str_c(0, 1:9), 10:n_obs) ) # renaming columns

(lppd <-
  ll %>%
  pivot_longer(
    everything(), # for all columns
    names_to = "i",
    values_to = "loglikelihood"
    ) %>%
  # log-likelihood to likelihood
  mutate(likelihood = exp(loglikelihood) ) %>%
  group_by(i) %>%
  # taking the log of the average likelihood
  summarise(log_mean_likelihood = mean(likelihood) %>% log() ) %>%
  # computing the sum of these values
  summarise(lppd = sum(log_mean_likelihood) ) %>%
  pull(lppd) )
```

Widely applicable information criterion

$$\text{lppd}(y, \Theta) = \sum_i \log \frac{1}{S} \sum_s p(y_i|\Theta_s)$$

```
n_obs <- nrow(cars) # number of observations

ll <-
  log_lik(mod1) %>% # pointwise log-likelihood (S rows * N observations)
  data.frame() %>% # converts to dataframe
  set_names(c(str_c(0, 1:9), 10:n_obs) ) # renaming columns

(lppd <-
  ll %>%
  pivot_longer(
    everything(), # for all columns
    names_to = "i",
    values_to = "loglikelihood"
    ) %>%
  # log-likelihood to likelihood
  mutate(likelihood = exp(loglikelihood) ) %>%
  group_by(i) %>%
  # taking the log of the average likelihood
  summarise(log_mean_likelihood = mean(likelihood) %>% log() ) %>%
  # computing the sum of these values
  summarise(lppd = sum(log_mean_likelihood) ) %>%
  pull(lppd) )
```



```
[1] -206.5653
```

Widely applicable information criterion

La deuxième partie du calcul du WAIC est le nombre de paramètres effectif, p_{WAIC}. On définit $\text{var}_\theta \log[p(y_i|\theta)]$ comme la variance de la log-vraisemblance pour chaque observation i de l'échantillon d'entraînement.

Widely applicable information criterion

La deuxième partie du calcul du WAIC est le nombre de paramètres effectif, p_{WAIC}. On définit $\text{var}_\theta \log[p(y_i|\theta)]$ comme la variance de la log-vraisemblance pour chaque observation i de l'échantillon d'entraînement.

$$p_{\text{WAIC}} = \sum_i \text{var}_\theta \log[p(y_i|\theta)]$$

Widely applicable information criterion

La deuxième partie du calcul du WAIC est le nombre de paramètres effectif, p_{WAIC}. On définit $\text{var}_\theta \log[p(y_i|\theta)]$ comme la variance de la log-vraisemblance pour chaque observation i de l'échantillon d'entraînement.

$$p_{\text{WAIC}} = \sum_i \text{var}_\theta \log[p(y_i|\theta)]$$

```
(pwaic <-
  ll %>%
  pivot_longer(
    everything(),
    names_to = "i",
    values_to = "loglikelihood"
    ) %>%
  group_by(i) %>%
  summarise(var_loglikelihood = var(loglikelihood) ) %>%
  summarise(pwaic = sum(var_loglikelihood) ) %>%
  pull() )
```

Widely applicable information criterion

La deuxième partie du calcul du WAIC est le nombre de paramètres effectif, p_{WAIC}. On définit $\text{var}_\theta \log[p(y_i|\theta)]$ comme la variance de la log-vraisemblance pour chaque observation i de l'échantillon d'entraînement.

$$p_{\text{WAIC}} = \sum_i \text{var}_\theta \log[p(y_i|\theta)]$$

```
(pwaic <-
  ll %>%
  pivot_longer(
    everything(),
    names_to = "i",
    values_to = "loglikelihood"
    ) %>%
  group_by(i) %>%
  summarise(var_loglikelihood = var(loglikelihood) ) %>%
  summarise(pwaic = sum(var_loglikelihood) ) %>%
  pull() )
```

```
[1] 3.369924
```

Widely applicable information criterion

Ensuite, le WAIC est défini par :

Widely applicable information criterion

Ensuite, le WAIC est défini par :

$$\mathrm{WAIC}(y, \Theta) = -2\Big(\mathrm{lppd} - \underbrace{\sum_i \mathrm{var}_\theta \log[p(y_i|\theta)]}_{\text{penalty term}}\Big)$$

Widely applicable information criterion

Ensuite, le WAIC est défini par :

$$\text{WAIC}(y, \Theta) = -2\Big(\text{lppd} - \underbrace{\sum_i \text{var}_\theta \log[p(y_i|\theta)]}_{\text{penalty term}}\Big)$$

```
(WAIC <- -2 * (lppd - pwaic) )
```

Widely applicable information criterion

Ensuite, le WAIC est défini par :

$$\text{WAIC}(y, \Theta) = -2\Big(\text{lppd} - \underbrace{\sum_i \text{var}_\theta \log[p(y_i|\theta)]}_{\text{penalty term}}\Big)$$

```
(WAIC <- -2 * (lppd - pwaic) )
```

```
[1] 419.8704
```

Widely applicable information criterion

Le WAIC est aussi un estimateur de la déviance *out-of-sample*. La fonction `brms::waic()` permet de le calculer directement.

Widely applicable information criterion

Le WAIC est aussi un estimateur de la déviance *out-of-sample*. La fonction `brms::waic()` permet de le calculer directement.

```
waic(mod1)
```

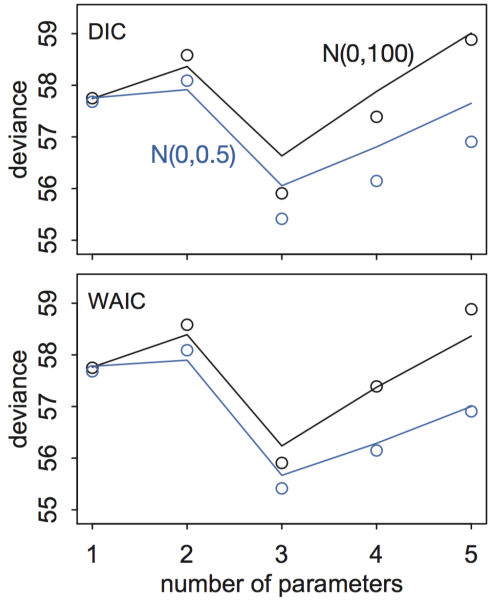
Widely applicable information criterion

Le WAIC est aussi un estimateur de la déviance *out-of-sample*. La fonction `brms::waic()` permet de le calculer directement.

```
waic(mod1)
```

```
Computed from 4000 by 50 log-likelihood matrix

          Estimate   SE
elpd_waic   -209.9  6.6
p_waic         3.4  1.2
waic         419.9 13.2

2 (4.0%) p_waic estimates greater than 0.4. We recommend trying loo instead.
```


Critères d'information et régularisation



Le DIC et le WAIC peuvent être conceptualisés (au même titre que l'AIC) comme des approximations de la déviance *out-of-sample*. On remarque que le WAIC produit des approximations plus précises que le DIC, et que l'utilisation de priors régularisateurs permet de réduire la déviance *out-of-sample*.

Et ensuite ?

Sélection de modèle : On choisit le meilleur modèle un utilisant un des outils présentés et on base nos conclusions sur les paramètres estimés par ce meilleur modèle.

Et ensuite ?

Sélection de modèle : On choisit le meilleur modèle un utilisant un des outils présentés et on base nos conclusions sur les paramètres estimés par ce meilleur modèle.

Comparaison de modèles : On utilise la validation croisée ou des critères d'informations mais aussi les outils de *posterior predictive checking* discutés précédemment, pour chaque modèle, afin d'étudier leurs forces et faiblesses.

Et ensuite ?

Sélection de modèle : On choisit le meilleur modèle un utilisant un des outils présentés et on base nos conclusions sur les paramètres estimés par ce meilleur modèle.

Comparaison de modèles : On utilise la validation croisée ou des critères d'informations mais aussi les outils de *posterior predictive checking* discutés précédemment, pour chaque modèle, afin d'étudier leurs forces et faiblesses.

Moyennage de modèles : On va construire des *posterior predictive checks* qui exploitent ce qu'on sait des capacités de prédiction de chaque modèle (e.g., via le WAIC).

Comparaison de modèles

Nous allons essayer de prédire les kg par gramme de lait (`kcal.per.g`) avec les prédicteurs `neocortex` et le logarithme de `mass`. Nous allons ensuite fitter 4 modèles qui correspondent aux 4 combinaisons possibles de prédicteurs et les comparer en utilisant le WAIC.

```
data(milk)

d <- milk[complete.cases(milk), ] # removing NAs
d$neocortex <- d$neocortex.perc / 100 # rescaling explanatory variable

head(d)
```

```
              clade            species kcal.per.g perc.fat perc.protein
1     Strepsirrhine     Eulemur fulvus       0.49    16.60        15.42
6  New World Monkey Alouatta seniculus       0.47    21.22        23.58
7  New World Monkey         A palliata       0.56    29.66        23.46
8  New World Monkey       Cebus apella       0.89    53.41        15.80
10 New World Monkey         S sciureus       0.92    50.58        22.33
11 New World Monkey   Cebuella pygmaea       0.80    41.35        20.85
   perc.lactose mass neocortex.perc neocortex
1         67.98 1.95          55.16    0.5516
6         55.20 5.25          64.54    0.6454
7         46.88 5.37          64.54    0.6454
8         30.79 2.51          67.64    0.6764
10        27.09 0.68          68.85    0.6885
11        37.80 0.12          58.85    0.5885
```


Comparaison de modèles

```
mod2.1 <- brm(
  kcal.per.g ~ 1,
  family = gaussian,
  data = d,
  prior = c(
    brms::prior(normal(0, 100), class = Intercept),
    brms::prior(exponential(0.01), class = sigma)
    ),
  iter = 2000, warmup = 1000,
  chains = 4, cores = 4
  )

mod2.2 <- brm(
  kcal.per.g ~ 1 + neocortex,
  family = gaussian,
  data = d,
  prior = c(
    brms::prior(normal(0, 100), class = Intercept),
    brms::prior(normal(0, 10), class = b),
    brms::prior(exponential(0.01), class = sigma)
    ),
  iter = 2000, warmup = 1000,
  chains = 4, cores = 4
  )
```


Comparaison de modèles

On peut utiliser la méthode `update()` qui permet de fitter plus rapidement un nouveau modèle qui ressemble à un modèle déjà existant.

```
mod2.3 <- update(
  mod2.2,
  newdata = d,
  formula = kcal.per.g ~ 1 + log(mass)
  )

mod2.4 <- update(
  mod2.3,
  newdata = d,
  formula = kcal.per.g ~ 1 + neocortex + log(mass)
  )
```


Comparaison de modèles

```
# calcul du WAIC et ajout du WAIC à chaque modèle

mod2.1 <- add_criterion(mod2.1, "waic")
mod2.2 <- add_criterion(mod2.2, "waic")
mod2.3 <- add_criterion(mod2.3, "waic")
mod2.4 <- add_criterion(mod2.4, "waic")

# comparaison des WAIC de chaque modèle

w <- loo_compare(mod2.1, mod2.2, mod2.3, mod2.4, criterion = "waic")
print(w, simplify = FALSE)
```

```
       elpd_diff se_diff elpd_waic se_elpd_waic p_waic se_p_waic waic  se_waic
mod2.4   0.0       0.0     8.4       2.6          3.1    0.8     -16.8   5.1
mod2.3  -3.8       1.7     4.6       2.1          2.0    0.4      -9.1   4.2
mod2.1  -4.0       2.4     4.4       1.8          1.3    0.3      -8.7   3.7
mod2.2  -4.7       2.5     3.6       1.6          1.9    0.3      -7.3   3.2
```


Akaike’s weights

Le poids d'un modèle est une estimation de la probabilité que ce modèle fera les meilleures prédictions possibles sur un nouveau jeu de données, conditionnellement au set de modèles considéré.

Akaike's weights

Le poids d'un modèle est une estimation de la probabilité que ce modèle fera les meilleures prédictions possibles sur un nouveau jeu de données, conditionnellement au set de modèles considéré.

$$w_i = \frac{\exp(-\frac{1}{2}\mathrm{dWAIC}_i)}{\sum_{j=1}^{m} \exp(-\frac{1}{2}\mathrm{dWAIC}_j)}$$

Akaike's weights

Le poids d'un modèle est une estimation de la probabilité que ce modèle fera les meilleures prédictions possibles sur un nouveau jeu de données, conditionnellement au set de modèles considéré.

$$w_i = \frac{\exp(-\frac{1}{2}\mathrm{dWAIC}_i)}{\sum_{j=1}^{m} \exp(-\frac{1}{2}\mathrm{dWAIC}_j)}$$

Cette fonction permet simplement de passer d'un WAIC à une probabilité (softmax). Le modèle `mod2.4` a un poids de 0.954, qui le place en tête du jeu de modèles. N'oublions cependant pas que nous disposons seulement de 12 observations…

Akaike's weights

Le poids d'un modèle est une estimation de la probabilité que ce modèle fera les meilleures prédictions possibles sur un nouveau jeu de données, conditionnellement au set de modèles considéré.

$$w_i = \frac{\exp(-\frac{1}{2}\text{dWAIC}_i)}{\sum_{j=1}^{m} \exp(-\frac{1}{2}\text{dWAIC}_j)}$$

Cette fonction permet simplement de passer d'un WAIC à une probabilité (softmax). Le modèle `mod2.4` a un poids de 0.954, qui le place en tête du jeu de modèles. N'oublions cependant pas que nous disposons seulement de 12 observations…

```
model_weights(mod2.1, mod2.2, mod2.3, mod2.4, weights = "waic") %>% round(digits = 3)
```

Akaike's weights

Le poids d'un modèle est une estimation de la probabilité que ce modèle fera les meilleures prédictions possibles sur un nouveau jeu de données, conditionnellement au set de modèles considéré.

$$w_i = \frac{\exp(-\frac{1}{2}\text{dWAIC}_i)}{\sum_{j=1}^{m} \exp(-\frac{1}{2}\text{dWAIC}_j)}$$

Cette fonction permet simplement de passer d'un WAIC à une probabilité (softmax). Le modèle `mod2.4` a un poids de 0.954, qui le place en tête du jeu de modèles. N'oublions cependant pas que nous disposons seulement de 12 observations…

```
model_weights(mod2.1, mod2.2, mod2.3, mod2.4, weights = "waic") %>% round(digits = 3)
```

```
mod2.1 mod2.2 mod2.3 mod2.4
 0.017  0.008  0.021  0.954
```

Moyennage de modèles (model averaging)

Pourquoi ne conserver uniquement le premier modèle et oublier les autres ? Une autre stratégie consisterait à pondérer les prédictions des modèles par leurs poids respectifs. C'est ce qu'on appelle le moyennage de modèles (*model averaging*).

Moyennage de modèles (model averaging)

Pourquoi ne conserver uniquement le premier modèle et oublier les autres ? Une autre stratégie consisterait à pondérer les prédictions des modèles par leurs poids respectifs. C'est ce qu'on appelle le moyennage de modèles (*model averaging*).

- Calculer le WAIC de chaque modèle

Moyennage de modèles (model averaging)

Pourquoi ne conserver uniquement le premier modèle et oublier les autres ? Une autre stratégie consisterait à pondérer les prédictions des modèles par leurs poids respectifs. C'est ce qu'on appelle le moyennage de modèles (*model averaging*).

- Calculer le WAIC de chaque modèle
- Calculer le poids de chaque modèle

Moyennage de modèles (model averaging)

Pourquoi ne conserver uniquement le premier modèle et oublier les autres ? Une autre stratégie consisterait à pondérer les prédictions des modèles par leurs poids respectifs. C'est ce qu'on appelle le moyennage de modèles (*model averaging*).

- Calculer le WAIC de chaque modèle
- Calculer le poids de chaque modèle
- Simuler des données à partir de chaque modèle

Moyennage de modèles (model averaging)

Pourquoi ne conserver uniquement le premier modèle et oublier les autres ? Une autre stratégie consisterait à pondérer les prédictions des modèles par leurs poids respectifs. C'est ce qu'on appelle le moyennage de modèles (*model averaging*).

- Calculer le WAIC de chaque modèle
- Calculer le poids de chaque modèle
- Simuler des données à partir de chaque modèle
- Combiner ces valeurs simulées dans un *ensemble* de prédictions pondérées par le poids du modèle

Moyennage de modèles (model averaging)

On peut utiliser la fonction `brms::pp_average()` qui pondère les prédictions de chaque modèle par leur poids.

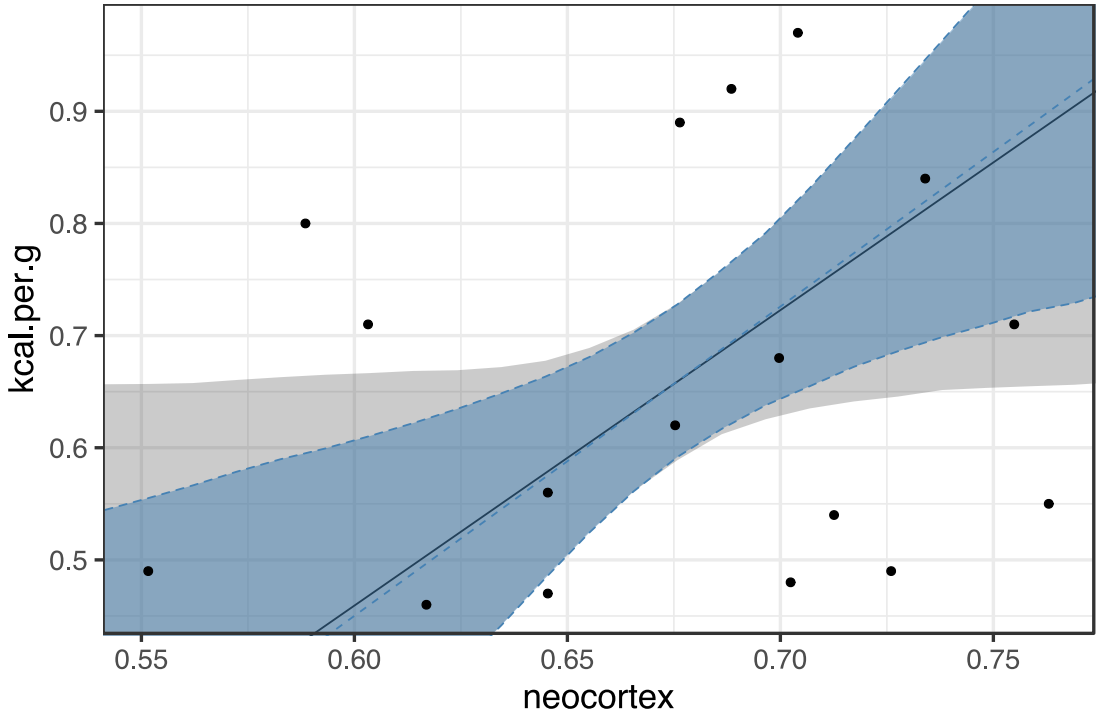
Moyennage de modèles (model averaging)

On peut utiliser la fonction `brms::pp_average()` qui pondère les prédictions de chaque modèle par leur poids.

```
# grille de valeurs pour lesquelles on va générer des prédictions
new_data <- data.frame(
  neocortex = seq(from = 0.5, to = 0.8, length.out = 30),
  mass = 4.5
  )

# prédictions du modèle mod2.4
f <- fitted(mod2.4, newdata = new_data) %>%
  as.data.frame() %>%
  bind_cols(new_data)

# prédictions moyennées sur les 4 modèles
averaged_predictions <- pp_average(
  mod2.1, mod2.2, mod2.3, mod2.4,
  weights = "waic",
  method  = "fitted",
  newdata = new_data
  ) %>%
  as.data.frame() %>%
  bind_cols(new_data)
```

Moyennage de modèles (model averaging)

Voici les prédictions de tous les modèles considérés, pondérés par leur poids respectif. Comme le modèle `mod2.4` concentrait quasiment tout le poids, il fait sens que cette prédiction moyennée soit similaire aux prédictions du modèle `mod2.4`.

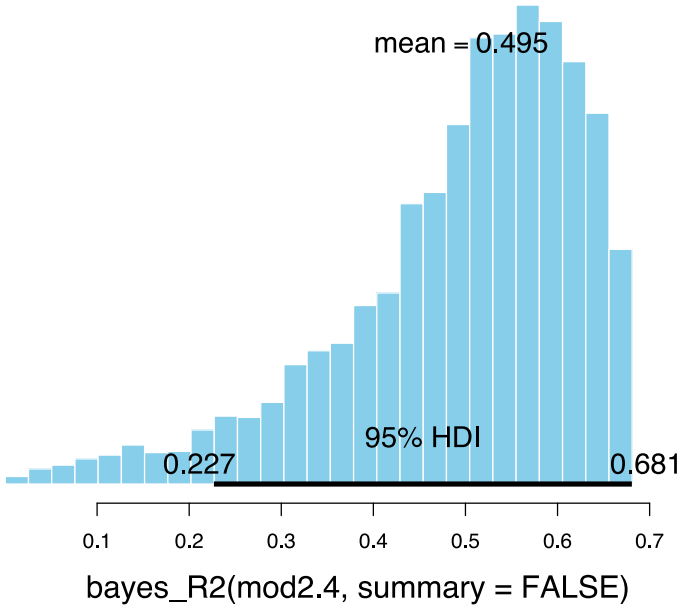


R-squared

```
bayes_R2(mod2.4) %>% round(digits = 3)
```

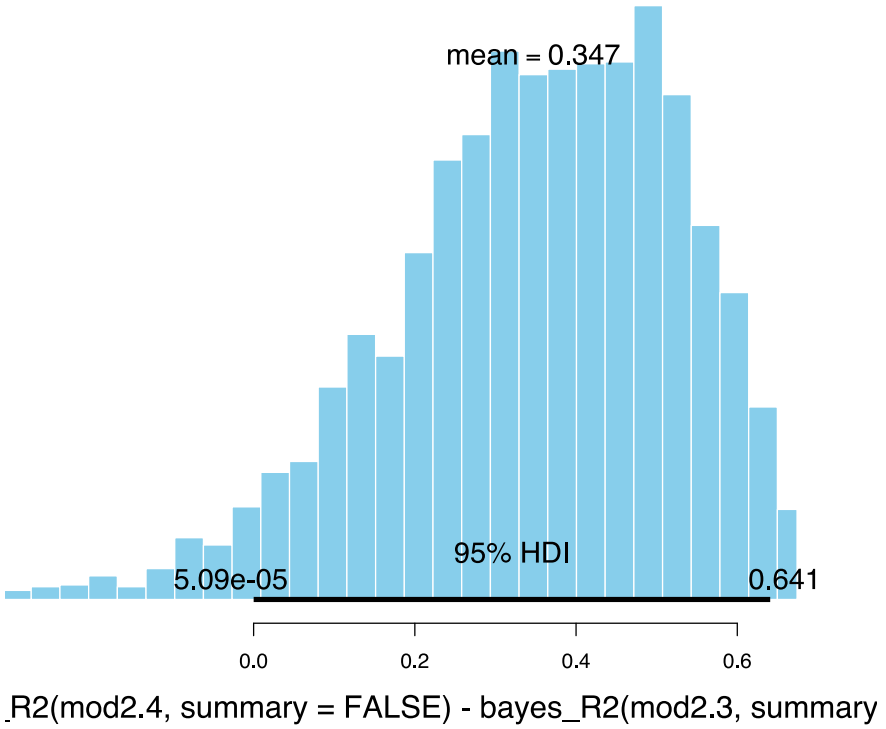
```
   Estimate Est.Error  Q2.5 Q97.5
R2    0.495     0.132 0.145 0.664
```

```
plotPost(bayes_R2(mod2.4, summary = FALSE) )
```



R-squared

```
plotPost(bayes_R2(mod2.4, summary = FALSE) - bayes_R2(mod2.3, summary = FALSE) )
```



Conclusions

Se méfier des interprétations intuitives (et souvent erronées) de la *p*-valeur, des intervalles de confiance, ou du facteur de Bayes.

Conclusions

Se méfier des interprétations intuitives (et souvent erronées) de la *p*-valeur, des intervalles de confiance, ou du facteur de Bayes.

Retenir que la difficulté en modélisation est de trouver un juste équilibre entre sous-apprentissage (*under-fitting*) et sur-apprentissage (*over-fitting*).

Conclusions

Se méfier des interprétations intuitives (et souvent erronées) de la *p*-valeur, des intervalles de confiance, ou du facteur de Bayes.

Retenir que la difficulté en modélisation est de trouver un juste équilibre entre sous-apprentissage (*under-fitting*) et sur-apprentissage (*over-fitting*).

Pour contraindre l'apprentissage réalisé par le modèle sur les données observées (et ainsi éviter que le modèle accorde trop de poids à ces données), on peut utiliser des priors dits **régularisateurs** et/ou des outils comme la validation croisée ou les critères d'informations permettant d'estimer les capacités de prédiction du modèle sur de nouvelles données.

Travaux pratiques

```
data(Howell1)
d <- Howell1 %>% mutate(age = scale(age) )

set.seed(666)
i <- sample(1:nrow(d), size = nrow(d) / 2)

d1 <- d[i, ] # échantillon d'entraînement
d2 <- d[-i, ] # échantillon de test
```

Nous avons maintenant deux dataframes, de 272 lignes chacune. On va utiliser `d1` pour fitter nos modèles et `d2` pour les évaluer.

Travaux pratiques

Soit h_i les valeurs de taille et x_i les valeurs centrées d'âge, sur la ligne i. Construisez les modèles suivants avec `d1`, en utilisant `brms::brm()` et des priors faiblement régularisateurs.

$$
\begin{aligned}
\mathcal{M}_1 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i \\
\mathcal{M}_2 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i + \beta_2 x_i^2 \\
\mathcal{M}_3 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 \\
\mathcal{M}_4 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 \\
\mathcal{M}_5 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 \\
\mathcal{M}_6 : h_i &\sim \text{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \beta_6 x_i^6
\end{aligned}
$$

Travaux pratiques

1. Comparer ces modèles en utilisant le WAIC. Comparer les rangs des modèles et leurs poids.

2. Pour chaque modèle, produire un plot de la moyenne estimée et l'intervalle de confiance à 97% de la moyenne, surimposée aux données brutes. Comment ces prédictions diffèrent-elles selon les modèles ?

3. Faire un plot des prédictions moyennées sur tous les modèles (sur les trois meilleurs). En quoi ces prédictions diffèrent-elles des prédictions du modèle avec le plus petit WAIC ?

4. Calculer la déviance *out-of-sample* pour chaque modèle. Comparer les déviances obtenues à la question précédente aux valeurs de WAIC. Basé sur les déviances obtenues, quel modèle fait les meilleures prédictions ? Est-ce que le WAIC est un bon estimateur de la déviance *out-of-sample* ?

Solution - Question 1

```
mod3.1 <- brm(
  height ~ 1 + age,
  family = gaussian(),
  data = d1,
  prior = c(
    brms::prior(normal(0, 100), class = Intercept),
    brms::prior(exponential(0.01), class = sigma)
    )
  )

mod3.2 <- update(
  mod3.1,
  newdata = d1,
  formula = height ~ 1 + age + I(age^2)
  )

mod3.3 <- update(
  mod3.1,
  newdata = d1,
  formula = height ~ 1 + age + I(age^2) + I(age^3)
  )
```


Solution - Question 1

```
mod3.4 <- update(
  mod3.1,
  newdata = d1,
  formula = height ~ 1 + age + I(age^2) + I(age^3) + I(age^4)
  )

mod3.5 <- update(
  mod3.1,
  newdata = d1,
  formula = height ~ 1 + age + I(age^2) + I(age^3) + I(age^4) + I(age^5)
  )

mod3.6 <- update(
  mod3.1,
  newdata = d1,
  formula = height ~ 1 + age + I(age^2) + I(age^3) + I(age^4) + I(age^5) + I(age^6)
  )
```


Solution - Question 1

```
# calcul du WAIC et ajout du WAIC à chaque modèle

mod3.1 <- add_criterion(mod3.1, "waic")
mod3.2 <- add_criterion(mod3.2, "waic")
mod3.3 <- add_criterion(mod3.3, "waic")
mod3.4 <- add_criterion(mod3.4, "waic")
mod3.5 <- add_criterion(mod3.5, "waic")
mod3.6 <- add_criterion(mod3.6, "waic")

# comparaison des WAIC de chaque modèle

mod_comp <- loo_compare(mod3.1, mod3.2, mod3.3, mod3.4, mod3.5, mod3.6, criterion = "waic")
print(mod_comp, digits = 2, simplify = FALSE)
```

```
       elpd_diff se_diff  elpd_waic se_elpd_waic p_waic   se_p_waic waic
mod3.6     0.00      0.00  -957.23     12.86         7.68     0.96   1914.46
mod3.4    -2.09      2.71  -959.32     13.45         6.10     0.92   1918.63
mod3.5    -3.01      2.73  -960.24     13.47         6.92     1.01   1920.47
mod3.3   -19.91      6.22  -977.14     12.49         5.47     0.86   1954.27
mod3.2  -123.69     13.52 -1080.92     11.46         5.27     1.08   2161.85
mod3.1  -247.63     15.25 -1204.86     11.23         3.36     0.41   2409.72
       se_waic
mod3.6    25.71
mod3.4    26.91
mod3.5    26.95
mod3.3    24.98
mod3.2    22.92
mod3.1    22.45
```

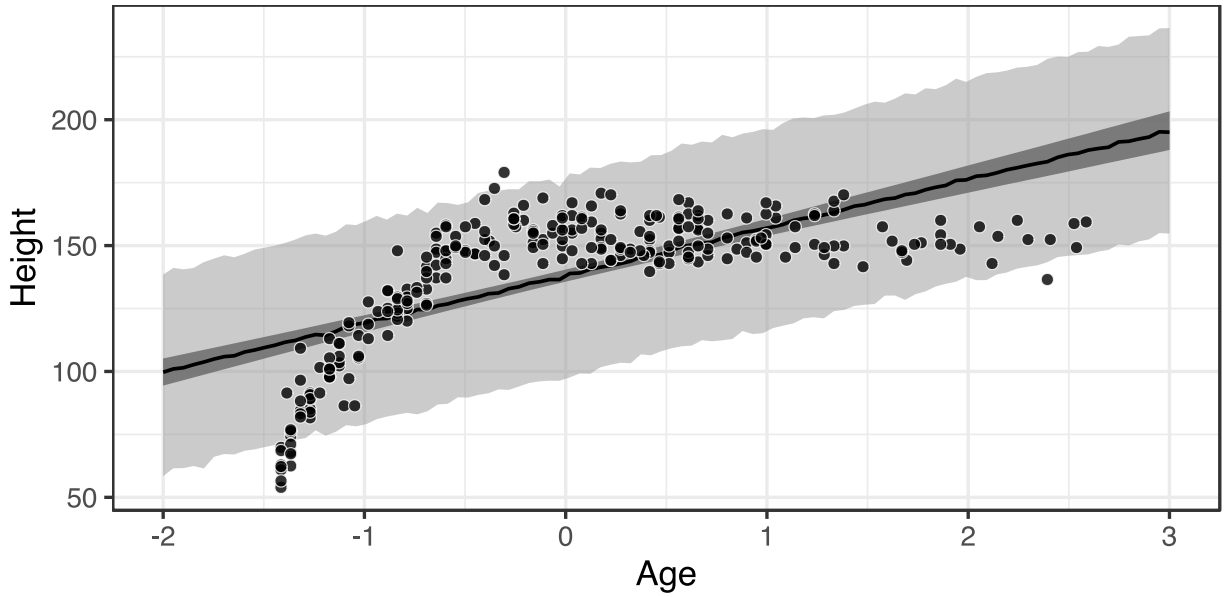

Solution - Question 2

```
# on crée un vecteur de valeurs possibles pour "age"
age_seq <- data.frame(age = seq(from = -2, to = 3, length.out = 1e2) )

# on récupère les prédictions du modèle pour ces valeurs
mu <- data.frame(fitted(mod3.1, newdata = age_seq) ) %>% bind_cols(age_seq)

# on récupère les prédictions du modèle pour ces valeurs
pred_age <- data.frame(predict(mod3.1, newdata = age_seq) ) %>% bind_cols(age_seq)

# on affiche les dix premières prédictions
head(pred_age, 10)
```

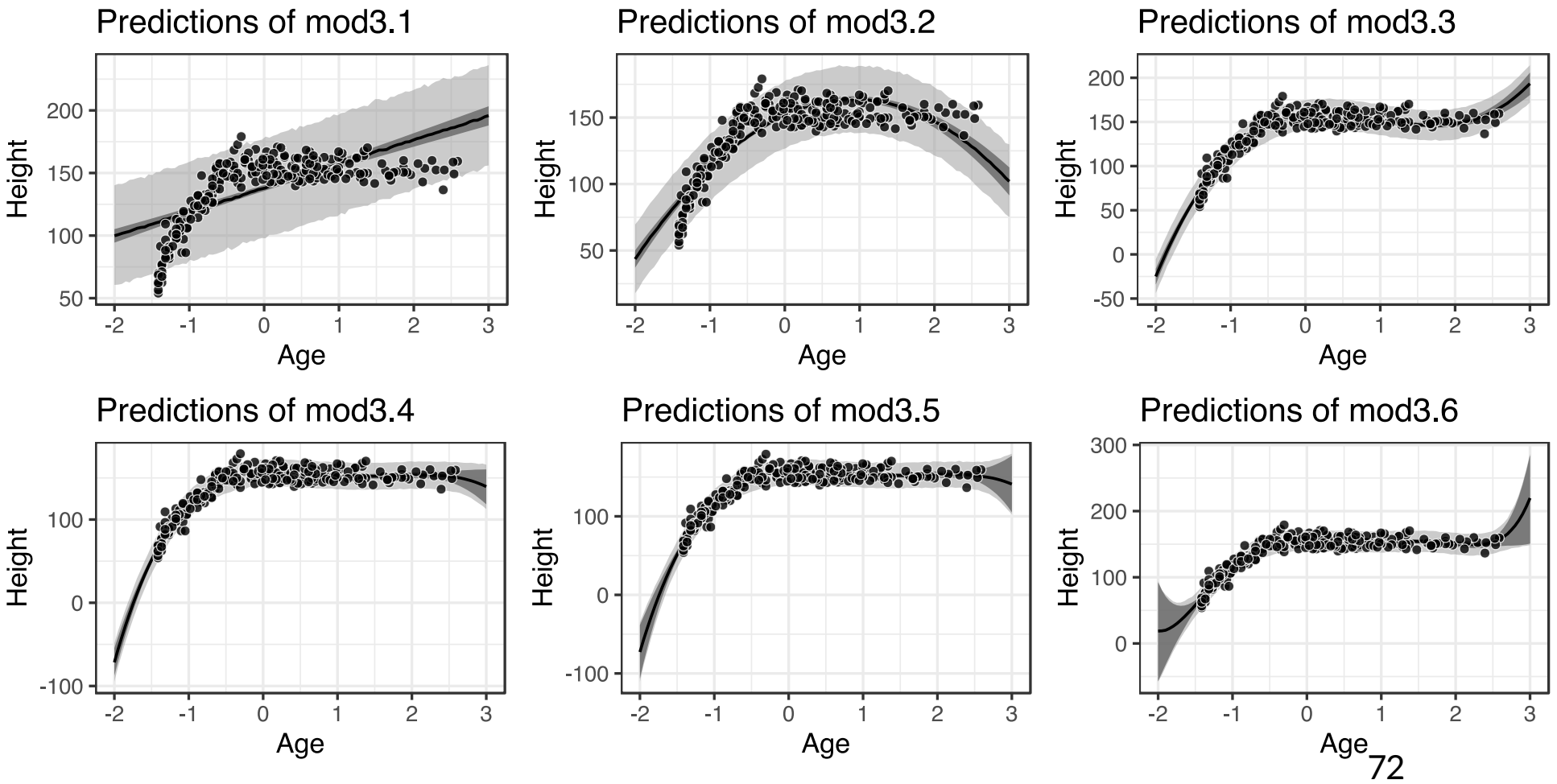
```
    Estimate Est.Error     Q2.5    Q97.5       age
1   99.69745  20.28642 58.39932 138.4827 -2.000000
2  101.00954  20.75176 61.53742 141.1399 -1.949495
3  101.48606  20.24859 61.60244 140.5623 -1.898990
4  102.71003  20.13705 62.44134 141.1389 -1.848485
5  103.74612  20.70298 61.45332 143.4339 -1.797980
6  104.92502  20.30197 66.08221 144.8113 -1.747475
7  105.88654  20.34482 67.20133 146.1468 -1.696970
8  106.23664  20.03543 67.01264 145.3628 -1.646465
9  107.72163  19.95983 68.48510 146.7435 -1.595960
10 108.52388  20.06383 69.04406 148.4426 -1.545455
```


Solution - Question 2

```
d1 %>%
  ggplot(aes(x = age, y = height) ) +
  geom_ribbon(
    data = mu, aes(x = age, ymin = Q2.5, ymax = Q97.5),
    alpha = 0.8, inherit.aes = FALSE
    ) +
  geom_smooth(
    data = pred_age, aes(y = Estimate, ymin = Q2.5, ymax = Q97.5),
    stat = "identity", color = "black", alpha = 0.5, size = 1
    ) +
  geom_point(colour = "white", fill = "black", pch = 21, size = 3, alpha = 0.8) +
  theme_bw(base_size = 20) + labs(x = "Age", y = "Height")
```



Solution - Question 2

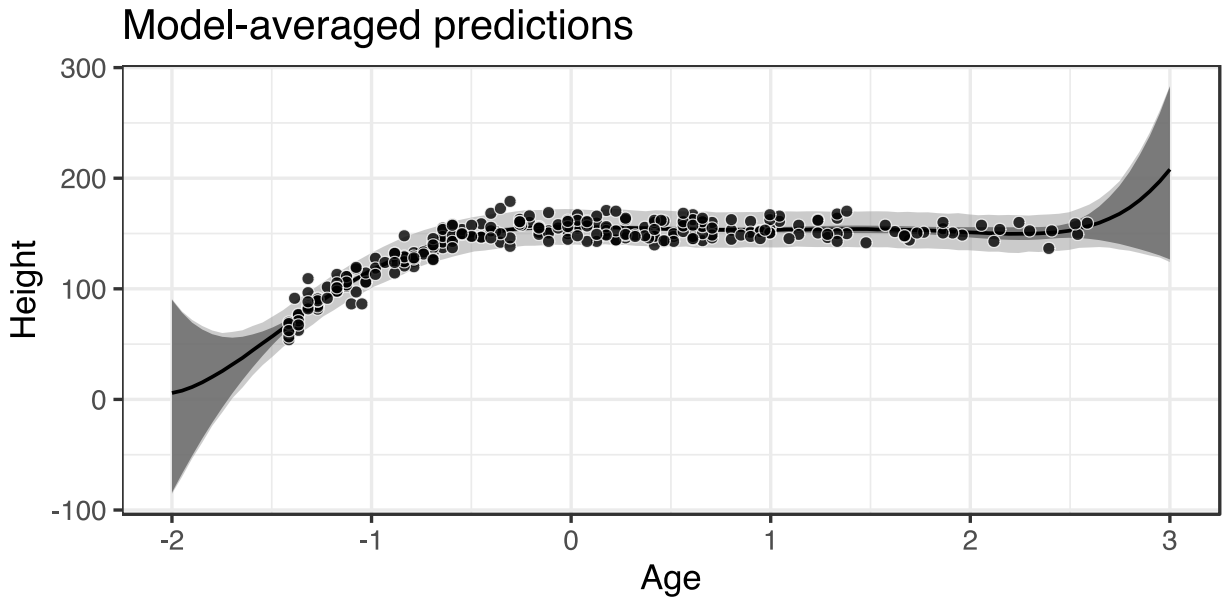


Solution - Question 3

```
# prédictions moyennées sur les 4 modèles
averaged_predictions_mu <- pp_average(
  mod3.1, mod3.2, mod3.3, mod3.4, mod3.5, mod3.6,
  weights = "waic",
  method  = "fitted",
  newdata = age_seq
  ) %>%
  as.data.frame() %>%
  bind_cols(age_seq)

# prédictions moyennées sur les 4 modèles
averaged_predictions_age <- pp_average(
  mod3.1, mod3.2, mod3.3, mod3.4, mod3.5, mod3.6,
  weights = "waic",
  method  = "predict",
  newdata = age_seq
  ) %>%
  as.data.frame() %>%
  bind_cols(age_seq)
```


Solution - Question 3

```
d1 %>%
  ggplot(aes(x = age, y = height) ) +
  geom_ribbon(
    data = averaged_predictions_mu, aes(x = age, ymin = Q2.5, ymax = Q97.5),
    alpha = 0.8, inherit.aes = FALSE
    ) +
  geom_smooth(
    data = averaged_predictions_age, aes(y = Estimate, ymin = Q2.5, ymax = Q97.5),
    stat = "identity", color = "black", alpha = 0.5, size = 1
    ) +
  geom_point(colour = "white", fill = "black", pch = 21, size = 3, alpha = 0.8) +
  theme_bw(base_size = 20) + labs(x = "Age", y = "Height", title = "Model-averaged predictions")
```

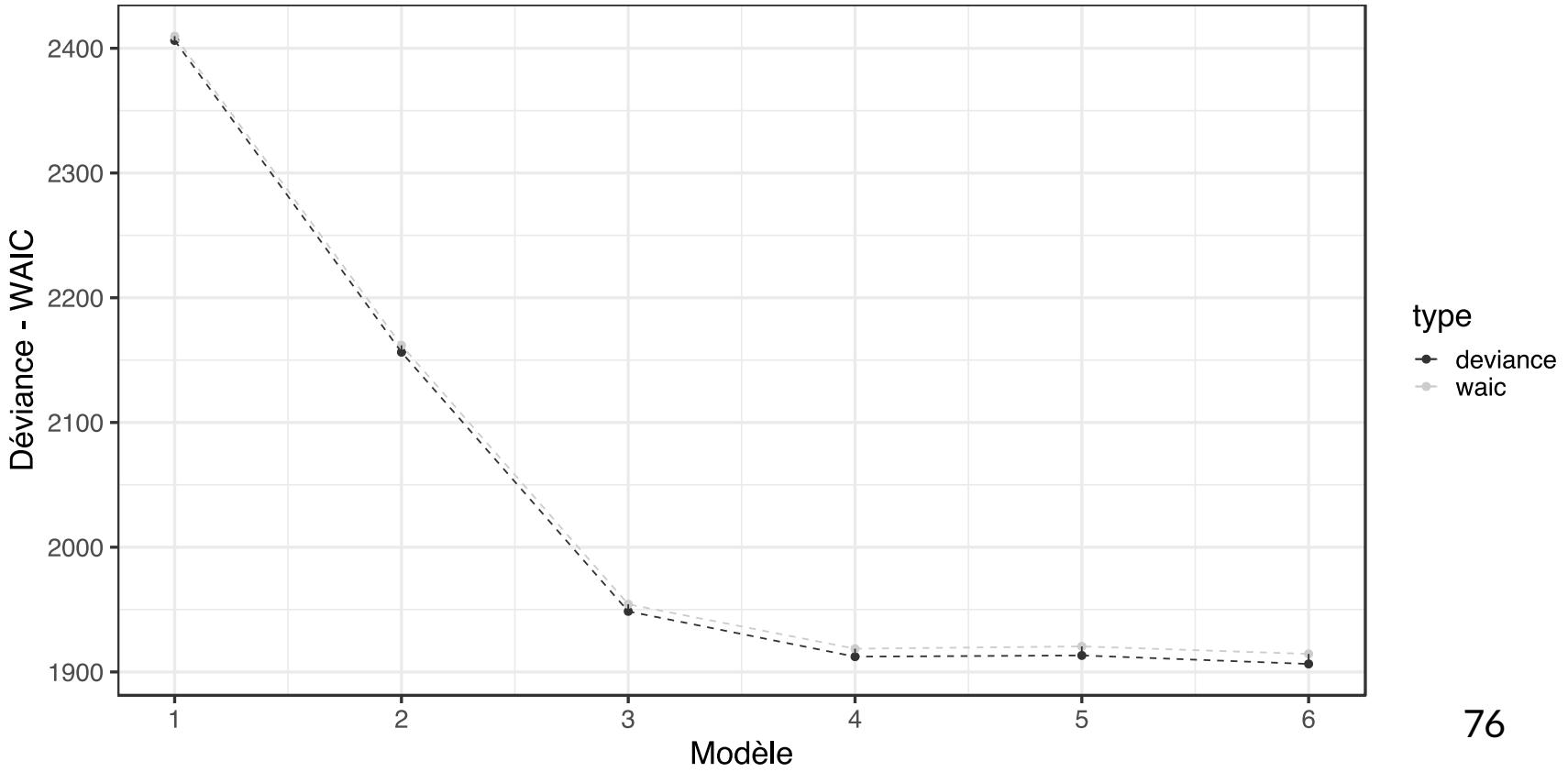


Solution - Question 4

```
# extract log-likelihood of mod3.1
# each row is an iteration and each column is an observation
log_lik_mod3.1 <- log_lik(mod3.1)

# NB: the deviance has a distribution too in the Bayesian world
dev.mod3.1 <- mean(-2 * rowSums(log_lik_mod3.1) )

# model 3.2
dev.mod3.2 <- mean(-2 * rowSums(log_lik(mod3.2) ) )

# model 3.3
dev.mod3.3 <- mean(-2 * rowSums(log_lik(mod3.3) ) )

# model 3.4
dev.mod3.4 <- mean(-2 * rowSums(log_lik(mod3.4) ) )

# model 3.5
dev.mod3.5 <- mean(-2 * rowSums(log_lik(mod3.5) ) )

# model 3.6
dev.mod3.6 <- mean(-2 * rowSums(log_lik(mod3.6) ) )
```


Solution - Question 4

```
deviances <- c(dev.mod3.1, dev.mod3.2, dev.mod3.3, dev.mod3.4, dev.mod3.5, dev.mod3.6)
comparison <- mod_comp %>% data.frame %>% select(waic) %>% rownames_to_column()
waics <- comparison %>% arrange(rowname) %>% pull(waic)
```



Solution - Question 4

```
deviances <- c(dev.mod3.1, dev.mod3.2, dev.mod3.3, dev.mod3.4, dev.mod3.5, dev.mod3.6)
comparison <- mod_comp %>% data.frame %>% select(waic) %>% rownames_to_column()
waics <- comparison %>% arrange(rowname) %>% pull(waic)
```