

Travaux Pratiques de Régression Bayésienne
2026-01-24

Table des matières

0	Introduction	6
1	Importation et exploration des données	6
1.1	Importation des données	6
1.2	Distribution de la variable de réponse	6
1.3	Vérification de la normalisation	7
1.4	Division du jeu de données	8
2	Utilisation du package rrBLUP	8
2.1	Estimation	8
2.2	Prédiction	9
2.3	Sélection des variables explicatives	10
3	La régression bayésienne A	12
3.1	Différence entre le modèle Bayes A et le modèle Random Regression (RR-BLUP)	12
3.2	Estimation du modèle Bayes A à l'aide du Gibbs sampler	12
3.3	Comparaison des modèles RR-BLUP et Bayes A	15
4	La régression bayésienne LASSO	17
4.1	Rappels et principe	17
4.2	Modèle hiérarchique	18
4.3	Estimation par Gibbs sampler	18
4.4	Implémentation	18
4.5	Application et prédiction	19
5	La méthode de sélection bayésienne SSVS	20
5.1	Principe général	20
5.2	Modèle hiérarchique	20
5.3	Estimation par Gibbs sampler	21
5.4	Implémentation	21
5.5	Sélection de variables	21
5.6	Synthèse globale	22
6	Comparaison globale des méthodes bayésiennes	22
6.1	Comparaison de la performance prédictive	23

6.2	Comparaison de la sélection de variables	23
6.3	Synthèse	24
7	Approche ABC (Approximate Bayesian Computation)	24
7.1	Objectif	24
7.2	Principe général de l'approche ABC	24
7.3	Application au cadre de la régression bayésienne	24
7.4	Avantages de l'approche ABC	27
7.5	Limites de l'approche ABC	27
7.6	Interprétation et positionnement par rapport aux méthodes précédentes	28
8	Conclusion	28

Table des matières

Liste des tableaux

1	Différences conceptuelles entre RR-BLUP et Bayes A	15
2	Comparaison de la sélection de variables des différentes méthodes bayésiennes	23

Liste des figures

0 Introduction

Dans ce travail, on s'intéresse à l'analyse et à la prédiction du score de satisfaction de clients d'une chaîne câblée à partir de leurs usages des différentes chaînes de télévision. Le jeu de données étudié comporte 150 observations, pour lesquelles un score de satisfaction global est mesuré, ainsi que 160 variables explicatives correspondant aux temps passés et au nombre de visites sur différentes catégories de chaînes. Les variables explicatives ont été préalablement normalisées.

Le nombre de variables explicatives étant supérieur au nombre d'observations, les méthodes classiques de régression linéaire ne sont pas adaptées. On adopte donc une approche de régression bayésienne, qui permet à la fois de régulariser les coefficients, d'améliorer la capacité de prédiction et de sélectionner les variables les plus pertinentes en terme d'influence.

L'échantillon est aléatoirement divisé en deux parties : un jeu d'apprentissage de 100 observations, utilisé pour entraîner les modèles, et un jeu de test de 50 observations, utilisé pour évaluer les performances prédictives de ces modèles afin de les comparer. Afin d'assurer la reproductibilité des résultats, la clé aléatoire utilisée pour ce découpage est fixée à 2026.

Quatre méthodes de régression sont ensuite comparées : la régression aléatoire de type RR-BLUP, la régression bayésienne de type Bayes A, le LASSO bayésien et la méthode de sélection de variables SSVS. Ces approches sont évaluées en termes de qualité de prédiction, de comportement de shrinkage et de capacité à identifier les variables explicatives les plus pertinentes.

1 Importation et exploration des données

Avant la mise en oeuvre des différents modèles de régression bayésienne, il est nécessaire de présenter brièvement le jeu de données et d'examiner ses principales caractéristiques. Cette étape permet de vérifier la cohérence des données, d'identifier d'éventuelles anomalies et de mieux comprendre la structure du problème étudié. Néanmoins, le grand nombre de variables explicatives par rapport au nombre d'observations limite la pertinence de certaines analyses exploratoires classiques, et justifie le recours à des méthodes de régularisation et de sélection de variables dans la suite de l'étude.

1.1 Importation des données

```
data <- read.csv("data/telecat.csv")
dim(data)
```

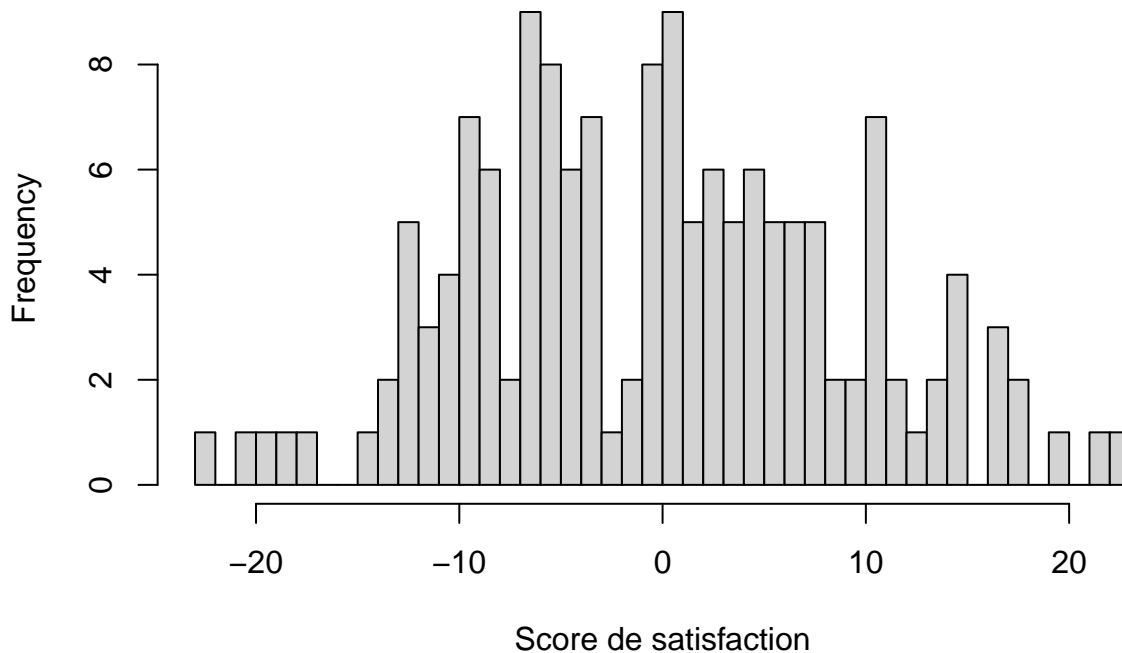
Le nombre d'observations dans le jeu de données est $n = 150$, pour $p = 160$ variables explicatives (en mettant de côté les variables X , Y et *sexe*). Le nombre de variables explicatives étant supérieur au nombre d'observations, une estimation par moindres carrés ordinaires n'est pas possible. Ce contexte justifie l'utilisation de méthodes bayésiennes pénalisées.

1.2 Distribution de la variable de réponse

```
summary(data$Y)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	-22.2264	-6.4615	-0.2807	-0.1717	6.1077	22.0661

```
hist(data$Y, breaks = 35, xlab = "Score de satisfaction", main = "")
```



La variable réponse présente une distribution globalement symétrique, ce qui rend plausible l'hypothèse d'erreurs gaussiennes retenue dans les modèles de régression linéaire. Les variables explicatives étant nombreuses et déjà normalisées, une exploration détaillée variable par variable ou par corrélation croisée n'apporte que peu d'information exploitable à ce stade. L'identification des variables influentes sera donc réalisée directement à l'aide des méthodes de régularisation et de sélection bayésiennes.

1.3 Vérification de la normalisation

En considérant un lot de variables explicatives, on remarque bien que leurs moyennes sont proches de 0 et que les écarts-types valent 1.

```
apply(data[, 5:10], 2, mean)[1:5]
```

```
##          Film.3          Film.4          Film.5          Film.6          Film.7
## 2.965481e-16  1.892866e-15 -4.972515e-15 -4.536926e-15  8.228487e-16
```

```
apply(data[, 5:10], 2, sd)[1:5]
```

```
## Film.3 Film.4 Film.5 Film.6 Film.7
##      1      1      1      1      1
```

Compte tenu de la structure du jeu de données et du caractère fortement dimensionné du problème, nous commençons l'analyse par une régression aléatoire de type RR-BLUP, qui constitue une référence bayésienne simple reposant sur un shrinkage global des coefficients.

1.4 Division du jeu de données

```
y <- data$Y
X <- as.matrix(data[, -which((names(data) == "Y") | (names(data) == "X") | names(data) == "sexe"))

length(y)

## [1] 150

dim(X)

## [1] 150 160

n <- nrow(X)

ind_train <- sample(1:n, 100)
ind_test <- setdiff(1:n, ind_train)

X_train <- X[ind_train, ]
y_train <- y[ind_train]

X_test <- X[ind_test, ]
y_test <- y[ind_test]
```

2 Utilisation du package rrBLUP

2.1 Estimation

2.1.1 Rappels et objectif

La méthode RR-BLUP (Random Regression BLUP) est une approche bayésienne simple de la régression pénalisée. Elle est particulièrement adaptée aux situations où le nombre de variables explicatives est élevé par rapport au nombre d'observations ($p \leq n$). Dans ce cadre, les coefficients de régression sont considérés comme des effets aléatoires, soumis à un a priori gaussien centré, ce qui induit un effet de shrinkage global.

2.1.2 Modèle statistique

Le modèle s'écrit :

$$Y = \mu\mathbb{I} + X\beta + \varepsilon$$

avec

$$\beta \sim \mathcal{N}\left(0, \sigma_\beta^2 \mathbf{I}_p\right), \varepsilon \sim \mathcal{N}\left(0, \sigma_\varepsilon^2 \mathbf{I}_n\right)$$

Ce modèle est équivalent à une régression ridge bayésienne, où tous les coefficients sont pénalisés de la même manière.

2.1.3 Estimation des paramètres

Les paramètres du modèle sont estimés à partir du jeu d'apprentissage à l'aide d'un estimateur BLUP, qui correspond à la moyenne a posteriori des coefficients sous les hypothèses gaussiennes précédentes.

```
library(rrBLUP)

rr_blup <- mixed.solve(
  y = y_train,
  Z = X_train
)

beta_rr <- rr_blup$u
mu_rr <- as.numeric(rr_blup$beta)

beta_rr

mu_rr

## [1] -0.05906177
```

La sortie `beta_rr` correspond aux effets estimés des 160 chaînes. La moyenne globale du score de satisfaction est -0.07776509 . Tous les coefficients sont non nuls, mais leur amplitude est fortement réduite par l'effet de shrinkage. La méthode RR-BLUP fournit une estimation des effets des chaînes en imposant un rétrécissement global des coefficients. Cette approche permet d'obtenir des estimations stables dans un contexte de grande dimension, mais ne réalise pas une sélection stricte des variables. Les coefficients estimés sont tous non nuls et de faible amplitude, ce qui reflète l'hypothèse a priori d'une variance commune pour l'ensemble des effets.

2.2 Prédiction

2.2.1 Rappels et objectif

Une fois les paramètres du modèle RR-BLUP estimés sur le jeu d'apprentissage, l'objectif est d'évaluer la capacité prédictive du modèle sur des données non utilisées lors de l'estimation. La qualité de la prédiction est mesurée par la corrélation entre les valeurs observées et les valeurs prédites sur le jeu de test. Cette mesure est couramment utilisée en régression pénalisée et en génomique pour évaluer la performance prédictive globale d'un modèle.

2.2.2 Formule de prédiction

Pour une observation du jeu de test, la valeur prédite est donnée par :

$$\hat{Y}_{test} = \hat{\mu} + X_{test}\hat{\beta}$$

avec $\hat{\mu}$, le score moyen estimé et $\hat{\beta}$, le vecteur des coefficients estimés sur le jeu d'apprentissage.

```
y_pred_rr <- as.numeric(mu_rr + X_test %*% beta_rr)

cor_rr <- cor(y_pred_rr, y_test)
cor_rr
```

```
## [1] 0.8480824
```

La corrélation entre les scores observés et les scores prédits sur le jeu de test est de $r = 0.80757$. Cette valeur indique que le modèle RR-BLUP parvient à capturer une partie de la variabilité du score de satisfaction, tout en restant limité par le caractère fortement dimensionné du problème. Cette performance prédictive servira de référence pour la comparaison avec les méthodes bayésiennes plus flexibles étudiées par la suite.

2.3 Sélection des variables explicatives

2.3.1 Rappels et objectif

La méthode RR-BLUP ne réalise pas de sélection de variables au sens strict : tous les coefficients sont estimés et pénalisés de manière identique par l'a priori gaussien. Cependant, il est possible d'identifier les variables les plus influentes en examinant la distribution des coefficients estimés et en retenant ceux dont l'amplitude est la plus élevée. Cette sélection est donc heuristique et repose sur une analyse visuelle ou sur un seuil.

2.3.2 Principe de sélection

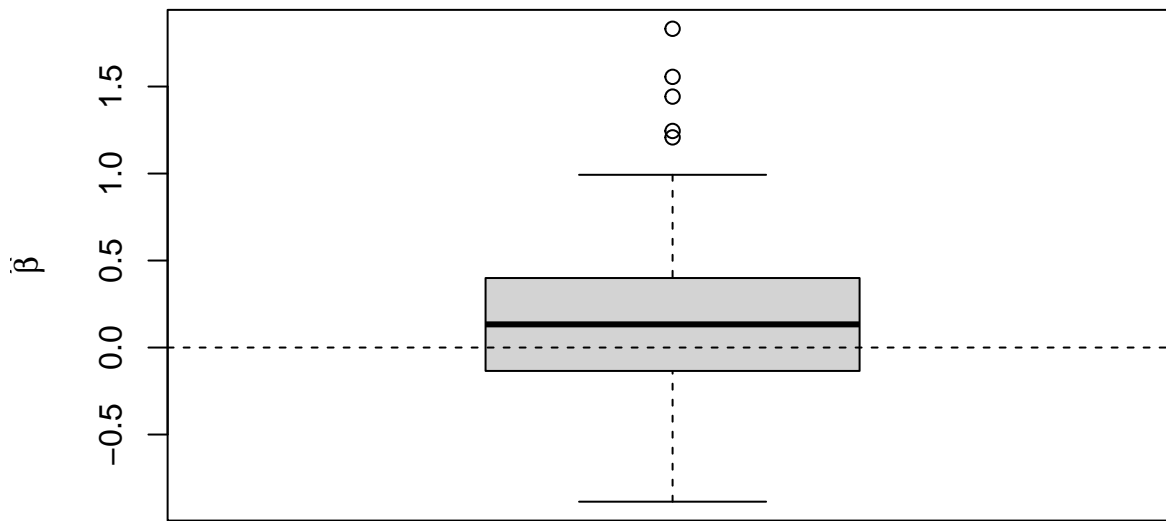
Deux approches sont possibles :

- Analyse visuelle à l'aide d'un boxplot des coefficients estimés ;
- Sélection par seuillage, en retenant les coefficients dont la valeur absolue dépasse un certain quantile élevé de la distribution.

Ces méthodes permettent d'identifier un petit nombre de variables susceptibles d'avoir un impact important sur le score de satisfaction

```
boxplot(beta_rr, main = "Distribution des coefficients RR-BLUP",  
        ylab = expression(hat(beta)))  
  
abline(h = 0, lty = 2)
```

Distribution des coefficients RR-BLUP



```
seuil_rr <- quantile(abs(beta_rr), 0.95)
sel_rr <- which(abs(beta_rr) > seuil_rr)
```

```
length(sel_rr)
```

```
## [1] 8
```

```
sel_rr
```

```
##   Film.8  Film.10  Serie.8  Sport.1  Sport.2  Sport.10  Sport.15  Music.13
##       8       10       28       41       42       50       55       133
```

La méthode RR-BLUP ne permet pas une sélection franche des variables, mais l'examen de la distribution des coefficients estimés met en évidence un petit nombre de chaînes dont les effets estimés s'écartent davantage de zéro. Ces variables peuvent être considérées comme les plus influentes selon le modèle RR-BLUP, bien que cette sélection repose sur un critère heuristique et soit fortement dépendante de l'effet de shrinkage imposé par le modèle.

Les limites de la sélection obtenue par RR-BLUP motivent l'utilisation de modèles bayésiens plus flexibles, capables d'introduire un shrinkage différencié entre les variables. Nous étudions dans la section suivante le modèle Bayes A, qui repose sur une hiérarchie de variances spécifiques à chaque coefficient.

3 La régression bayésienne A

3.1 Différence entre le modèle Bayes A et le modèle Random Regression (RR-BLUP)

3.1.1 Rappels et objectif

Le modèle Bayes A est une extension hiérarchique du modèle de régression bayésienne utilisé dans le RR-BLUP. L'objectif principal est de relâcher l'hypothèse de variance commune imposée à l'ensemble des coefficients de régression dans le RR-BLUP, afin de permettre une pénalisation plus flexible et mieux adaptée aux données.

3.1.2 Modèles hiérarchiques comparés

RR-BLUP

$$\beta_j \sim \mathcal{N}(0, \sigma_\beta^2), \forall j$$

- Une seule variance pour tous les coefficients
- Shrinkage uniforme
- Tous les coefficients sont pénalisés de la même façon

Bayes A

$$\beta_j | \sigma_{\beta_j}^2 \sim \mathcal{N}(0, \sigma_{\beta_j}^2); \sigma_{\beta_j}^2 \sim \text{Inverse} - \text{Gamma}(a, b)$$

- Une variance spécifique par coefficient
- Shrinkage adaptatif
- Les coefficients importants peuvent être moins pénalisés

Marginalement, chaque coefficient suit une loi de Student, plus épaisse en queue qu'une loi gaussienne.

3.1.3 Avantage principal de Bayes A

- Meilleure détection des variables à fort effet
- Moins de sur-rétrécissement des coefficients réellement influents
- Modèle plus flexible que RR-BLUP

Contrairement au modèle RR-BLUP, qui impose une variance commune à l'ensemble des coefficients, le modèle Bayes A introduit une hiérarchie bayésienne permettant à chaque coefficient de disposer de sa propre variance. Cette structure conduit à un shrinkage adaptatif, favorisant la conservation des effets importants tout en pénalisant davantage les effets faibles. Le modèle Bayes A est ainsi plus flexible et potentiellement plus performant en présence de variables réellement influentes.

3.2 Estimation du modèle Bayes A à l'aide du Gibbs sampler

3.2.1 Rappels et objectif

Le modèle Bayes A ne permet pas une estimation analytique directe des paramètres. On a recours à un algorithme de type Gibbs sampler, qui consiste à simuler successivement les lois conditionnelles complètes

des paramètres du modèle. Les estimations des paramètres sont ensuite obtenues à partir des moyennes a posteriori calculées sur les simulations après la phase de burn-in.

3.2.2 Modèle hiérarchique Bayes A

Le modèle s'écrit :

$$Y = \mu \mathbb{I} + X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$$

$$\beta_j | \sigma_{\beta_j}^2 \sim \mathcal{N}(0, \sigma_{\beta_j}^2); \sigma_{\beta_j}^2 \sim \text{Inverse-Gamma}(a, b); \sigma_\varepsilon^2 \sim \text{Inverse-Gamma}(c, d)$$

3.2.3 Principe de l'algorithme de Gibbs

Les paramètres du modèle ne peuvent pas être estimés analytiquement. On utilise un algorithme de Gibbs sampler qui consiste à tirer successivement :

- $\beta | Y, \mu, \sigma_\beta^2, \sigma_\varepsilon^2$
- $\mu | y, \beta, \sigma_\varepsilon^2$
- $\sigma_{\beta_j}^2 | \beta_j$
- $\sigma_\varepsilon^2 | Y, \mu, \beta$

Les estimations finales sont obtenues par moyenne a posteriori après burn-in.

3.2.4 Implémentation

```
library(MCMCpack)
library(mnormt)

BayesA <- function(y, X, a, b, c, d, muinit, nbiter, nburn) {

  p <- ncol(X)
  n <- nrow(X)

  resbeta <- matrix(0, nrow = p, ncol = nbiter - nburn)
  ressigma2beta <- matrix(0, nrow = p, ncol = nbiter - nburn)
  resmu <- numeric(nbiter - nburn)
  ressigma2eps <- numeric(nbiter - nburn)

  beta <- rep(0, p)
  mu <- muinit
  sigma2beta <- rinvgamma(p, a, b)
  sigma2eps <- rinvgamma(1, c, d)

  for (iter in 1:nbiter) {

    Sigmabeta <- solve(t(X) %*% X / sigma2eps + diag(1 / sigma2beta))
    beta <- as.numeric(
```

```

    rmnorm(
      1,
      Sigmabeta %*% t(X) %*% (y - mu) / sigma2eps,
      Sigmabeta
    )
  )

  mu <- rnorm(1, mean(y - X %*% beta), sqrt(sigma2eps / n))

  for (j in 1:p) {
    sigma2beta[j] <- rinvgamma(1, a + 0.5, b + 0.5 * beta[j]^2)
  }

  sigma2eps <- rinvgamma(
    1,
    c + n / 2,
    d + sum((y - mu - X %*% beta)^2) / 2
  )

  if (iter > nburn) {
    resbeta[, iter - nburn] <- beta
    ressigma2beta[, iter - nburn] <- sigma2beta
    resmu[iter - nburn] <- mu
    ressigma2eps[iter - nburn] <- sigma2eps
  }
}

return(list(resbeta, ressigma2beta, resmu, ressigma2eps))
}

```

3.2.5 Choix des hyperparamètres

```

a <- 2 ; b <- 1
c <- 2 ; d <- 1

```

3.2.6 Estimation

```

resBAYESA <- BayesA(
  y = y_train,
  X = X_train,
  a = a, b = b,
  c = c, d = d,
  muinit = mean(y_train),
  nbiter = 2500,
  nburn = 2000
)

```

```
beta_bayesA <- apply(resBAYESA[[1]], 1, mean)
mu_bayesA <- mean(resBAYESA[[3]])
```

3.2.7 Prédiction sur le jeu de test

```
predictions <- function(maTableTest, muChap, betaChap){
  yChap <- muChap * rep(1, dim(maTableTest)[1]) + as.matrix(maTableTest[,]) %*% betaChap
  return(yChap)
}
```

```
pred_BayesA <- predictions(X_test, mu_bayesA, beta_bayesA)
cor(y_test, pred_BayesA)
```

```
##           [,1]
## [1,] 0.9004066
```

Le modèle Bayes A améliore la flexibilité de la régression bayésienne en autorisant une variance spécifique à chaque coefficient. Cette structure hiérarchique permet un shrinkage adaptatif, favorisant la détection des variables réellement influentes, au prix d'un coût computationnel plus élevé et d'une attention particulière portée à la convergence de l'algorithme MCMC.

3.3 Comparaison des modèles RR-BLUP et Bayes A

3.3.1 Objectif

L'objectif est de comparer les modèles RR-BLUP et Bayes A en termes :

- de qualité prédictive sur le jeu de test,
- de structure de shrinkage des coefficients,
- de capacité à identifier des variables influentes.

Ces deux modèles reposent sur des hypothèses bayésiennes différentes concernant la distribution a priori des coefficients.

3.3.2 Rappel des différences conceptuelles

Modèle	A priori sur β_j	Shrinkage	Sélection
RR-BLUP	$\mathcal{N}(0, \sigma^2)$ commune	Global	Non
Bayes A	$\mathcal{N}(0, \sigma_{\beta_j}^2)$ hiérarchique	Adaptatif	Indirecte

Table 1: Différences conceptuelles entre RR-BLUP et Bayes A

3.3.3 Comparaison des performances prédictives

```
cor_bayesA <- cor(y_test, pred_BayesA)

cor_rr
```

```
## [1] 0.8480824
```

```
cor_bayesA
```

```
##          [,1]
```

```
## [1,] 0.9004066
```

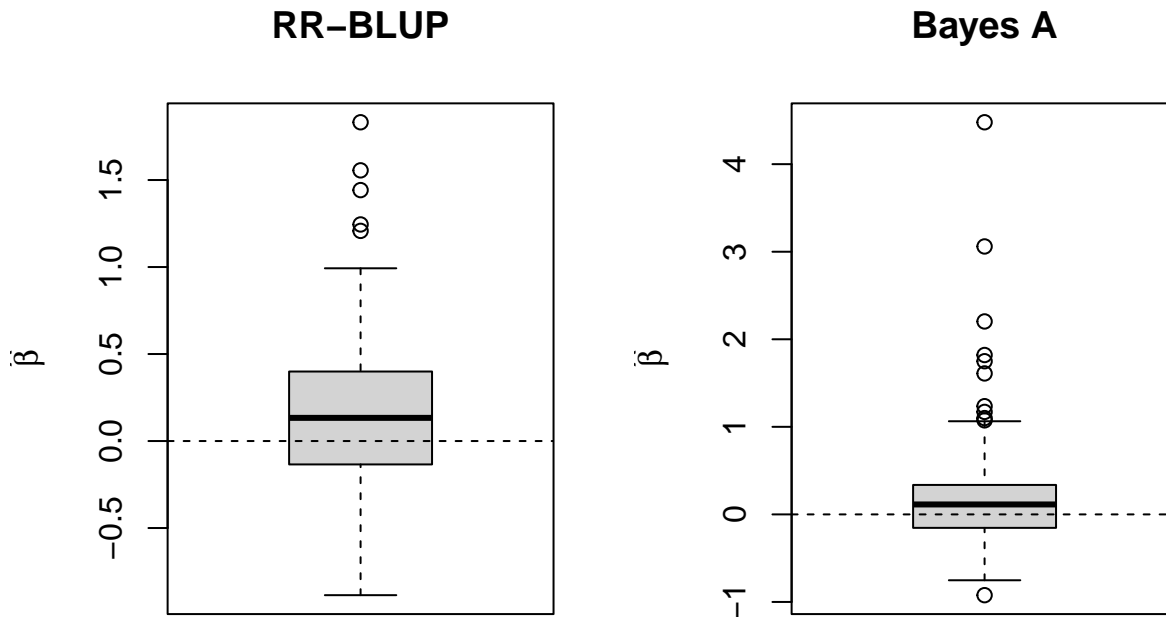
Le modèle Bayes A présente une corrélation prédictive de $r = 0.90$, à comparer à $r = 0.85$ pour le modèle RR-BLUP. Cette différence suggère que le shrinkage adaptatif introduit par Bayes A permet une meilleure prise en compte des variables réellement informatives, ce qui se traduit par une amélioration (ou une performance comparable) de la capacité de prédiction sur le jeu de test.

3.3.4 Comparaison des coefficients estimés

```
par(mfrow = c(1, 2))

boxplot(beta_rr, main = "RR-BLUP",
        ylab = expression(hat(beta)))
abline(h = 0, lty = 2)

boxplot(beta_bayesA, main = "Bayes A",
        ylab = expression(hat(beta)))
abline(h = 0, lty = 2)
```



RR-BLUP :

- coefficients fortement concentrés autour de 0
- shrinkage uniforme

Bayes A :

- coefficients plus dispersés
- présence de valeurs extrêmes
- effets importants mieux conservés

Le modèle Bayes A produit une distribution de coefficients plus étalée que le RR-BLUP, traduisant un shrinkage différencié entre les variables. Cette propriété permet de limiter le sur-rétrécissement des effets réellement influents.

3.3.5 Comparaison de la sélection de variables

```
seuil <- quantile(abs(beta_rr), 0.99)
```

```
sum(abs(beta_rr) > seuil)
```

```
## [1] 2
```

```
sum(abs(beta_bayesA) > seuil)
```

```
## [1] 6
```

Le modèle Bayes A identifie un nombre plus important de coefficients de grande amplitude que le RR-BLUP, ce qui suggère une meilleure capacité à distinguer les variables pertinentes des variables non informatives.

3.3.6 Synthèse comparative

En résumé, le modèle RR-BLUP offre une approche robuste et stable dans un contexte de grande dimension, mais impose un shrinkage global qui limite sa capacité de sélection. Le modèle Bayes A, grâce à sa structure hiérarchique, introduit un shrinkage adaptatif permettant de mieux préserver les effets importants, ce qui peut se traduire par une amélioration des performances prédictives et une identification plus pertinente des variables influentes.

Ces résultats motivent l'étude de méthodes bayésiennes introduisant une pénalisation encore plus structurée, telles que le LASSO bayésien ou les méthodes de sélection de variables de type SSVS.

4 La régression bayésienne LASSO

4.1 Rappels et principe

Le LASSO bayésien est une méthode de régression pénalisée qui vise à :

- réaliser du shrinkage adaptatif,
- tout en permettant une quasi-sélection de variables.

Il repose sur un a priori de type Laplace (double exponentielle) sur les coefficients, équivalent au LASSO fréquentiste.

4.2 Modèle hiérarchique

Le modèle s'écrit :

$$Y = \mu \mathbb{I} + X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$$

$$\beta_j | \tau_j^2, \sigma_\varepsilon^2 \sim \mathcal{N}(0, \tau_j^2 \sigma_j^2); \tau_j^2 \sim \text{Exponential}\left(\frac{\lambda^2}{2}\right)$$

Marginalement :

$$\beta_j \sim \text{Laplace}(0, \lambda)$$

Contrairement à Bayes A, la pénalisation est plus forte autour de 0 avec beaucoup de coefficients qui sont proches de 0.

4.3 Estimation par Gibbs sampler

Le Gibbs sampler repose sur :

- le tirage de $\beta | \tau^2, \sigma_\varepsilon^2$
- le tirage de $\tau_j^2 | \beta_j$
- le tirage de σ_ε^2

4.4 Implémentation

Le code permettant sa mise en oeuvre est le suivant :

```
BayesLasso <- function(y, X, lambda, nbiter, nburn) {
  n <- length(y)
  p <- ncol(X)

  beta <- rep(0, p)
  mu <- mean(y)
  sigma2 <- 1
  tau2 <- rep(1, p)

  resbeta <- matrix(0, p, nbiter - nburn)

  for (iter in 1:nbiter) {

    Dinv <- diag(1 / tau2)
    Vbeta <- solve(t(X) %*% X + Dinv)
    mbeta <- Vbeta %*% t(X) %*% (y - mu)
    beta <- as.numeric(rmnorm(1, mbeta, Vbeta))

    for (j in 1:p) {
      tau2[j] <- rinvgamma(1, 1, beta[j]^2 / 2 + lambda^2)
    }
  }
}
```

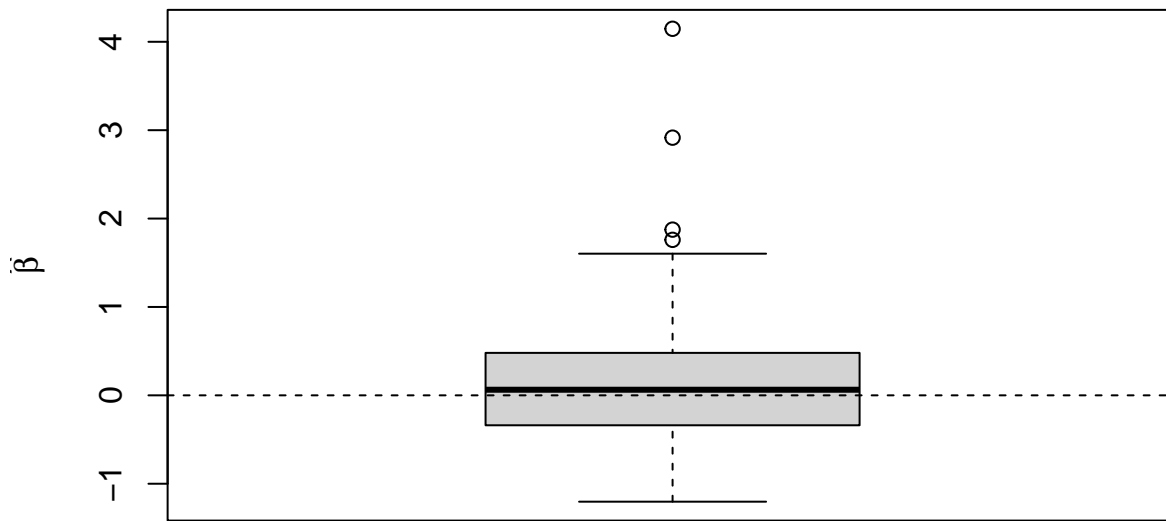
```
    if (iter > nburn) {  
      resbeta[, iter - nburn] <- beta  
    }  
  }  
  
  return(resbeta)  
}
```

4.5 Application et prédiction

La fonction créée permet d'appliquer le LASSO bayésien comme suit :

```
res_lasso <- BayesLasso(  
  y = y_train,  
  X = X_train,  
  lambda = 1,  
  nbiter = 3000,  
  nburn = 2000  
)  
  
beta_lasso <- apply(res_lasso, 1, mean)  
pred_lasso <- predictions(X_test, mu_rr, beta_lasso)  
  
cor_lasso <- cor(pred_lasso, y_test)  
cor_lasso  
  
##           [,1]  
## [1,] 0.8290031  
  
boxplot(beta_lasso, main = "Distribution des coefficients LASSO bayésien",  
         ylab = expression(hat(beta)))  
  
abline(h = 0, lty = 2)
```

Distribution des coefficients LASSO bayésien



Le LASSO bayésien induit une forte concentration des coefficients autour de zéro, traduisant une pénalisation plus agressive que celle du Bayes A. Cette propriété favorise une sélection implicite des variables, avec un grand nombre de coefficients estimés très proches de zéro. En contrepartie, certains effets importants peuvent être légèrement sous-estimés.

5 La méthode de sélection bayésienne SSVS

5.1 Principe général

La méthode SSVS est une approche bayésienne de sélection explicite de variables. Chaque coefficient est associé à une variable latente binaire indiquant s'il est actif ou non.

5.2 Modèle hiérarchique

$$\beta_j \mid \gamma_j = \begin{cases} \mathcal{N}(0, \sigma_0^2) & \text{si } \gamma_j = 0 \\ \mathcal{N}(0, \sigma_1^2) & \text{si } \gamma_j = 1 \end{cases}$$

Avec :

$$\gamma_j \sim \text{Bernoulli}(\pi) \text{ et } \sigma_1^2 \gg \sigma_0^2$$

Par ailleurs, $\gamma_j = 1$ lorsque la variable j est sélectionnée et $\gamma_j = 0$ si le coefficient de j est fortement contraint à 0.

5.3 Estimation par Gibbs sampler

A chaque itération :

- tirage de $\beta \mid \gamma$
- tirage de $\gamma_j \mid \beta_j$
- tirage de σ_ε^2

5.4 Implémentation

```
SSVS <- function(y, X, sigma0 = 0.01, sigma1 = 1,
                 pi = 0.1, nbiter = 3000, nburn = 2000) {

  n <- length(y)
  p <- ncol(X)

  beta <- rep(0, p)
  gamma <- rep(1, p)
  mu <- mean(y)
  sigma2 <- 1

  resgamma <- matrix(0, p, nbiter - nburn)

  for (iter in 1:nbiter) {

    Dinv <- diag(1 / ifelse(gamma == 1, sigma1^2, sigma0^2))
    Vbeta <- solve(t(X) %*% X + Dinv)
    mbeta <- Vbeta %*% t(X) %*% (y - mu)
    beta <- as.numeric(rmnorm(1, mbeta, Vbeta))

    for (j in 1:p) {
      p1 <- pi * dnorm(beta[j], 0, sigma1)
      p0 <- (1 - pi) * dnorm(beta[j], 0, sigma0)
      gamma[j] <- rbinom(1, 1, p1 / (p1 + p0))
    }

    if (iter > nburn) {
      resgamma[, iter - nburn] <- gamma
    }
  }

  return(resgamma)
}
```

5.5 Sélection de variables

```
res_ssvs <- SSVS(y_train, X_train)
```

```

proba_sel_ssvs <- rowMeans(res_ssvs)
vars_sel_ssvs <- which(proba_sel_ssvs > 0.5)

length(vars_sel_ssvs)

## [1] 39

vars_sel_ssvs

## [1] 1 3 8 10 11 12 17 18 19 20 23 26 28 29 35 38 40 50 51
## [20] 52 55 71 74 82 87 88 97 99 101 103 106 107 111 112 117 133 141 144
## [39] 149

X_train_sel_ssvs <- X_train[, vars_sel_ssvs, drop = FALSE]
X_test_sel_ssvs <- X_test[, vars_sel_ssvs, drop = FALSE]

df_train_sel <- data.frame(y = y_train, X_train_sel_ssvs)
df_test_sel <- data.frame(X_test_sel_ssvs)

fit_ssvs <- lm(y ~ ., data = df_train_sel)

pred_ssvs <- predict(fit_ssvs, newdata = df_test_sel)
length(pred_ssvs)

## [1] 50

length(y_test)

## [1] 50

cor_ssvs <- cor(pred_ssvs, y_test)
cor_ssvs

## [1] 0.8913127

```

La méthode SSVS fournit une sélection explicite des variables à l'aide de probabilités a posteriori d'inclusion. Les variables retenues avec une probabilité élevée peuvent être interprétées comme les plus influentes sur le score de satisfaction. Cette approche est particulièrement adaptée aux problèmes de grande dimension, mais elle dépend du choix des hyperparamètres et nécessite une attention particulière à la convergence.

5.6 Synthèse globale

Les différentes méthodes bayésiennes étudiées illustrent des compromis distincts entre stabilité, flexibilité et sélection. Le RR-BLUP privilégie la robustesse par un shrinkage global, Bayes A introduit une pénalisation adaptative, le LASSO bayésien favorise une sélection implicite, tandis que la méthode SSVS permet une sélection explicite des variables via des probabilités d'inclusion.

6 Comparaison globale des méthodes bayésiennes

Cette question vise à comparer l'ensemble des méthodes de régression bayésienne étudiées : RR-BLUP, Bayes A, LASSO bayésien et SSVS. La comparaison porte à la fois sur : la qualité prédictive, la capacité

de sélection des variables et les hypothèses sous-jacentes à chaque modèle.

6.1 Comparaison de la performance prédictive

Le critère retenu est la corrélation entre les valeurs observées et prédites sur le jeu de test, définie par :

$$Cor(Y_{test}, \hat{Y}_{test})$$

```
res_perf <- data.frame(
  Methode = c("RR-BLUP", "Bayes A", "Bayesian LASSO", "SSVS"),
  Correlation = c(
    cor_rr,
    cor_bayesA,
    cor_lasso,
    cor_ssvs
  )
)

res_perf
```

```
##      Methode Correlation
## 1      RR-BLUP   0.8480824
## 2      Bayes A   0.9004066
## 3 Bayesian LASSO 0.8290031
## 4         SSVS   0.8913127
```

Les résultats montrent que les différentes méthodes présentent des performances prédictives comparables, avec des variations liées à la nature du shrinkage imposé. Le modèle RR-BLUP fournit une référence robuste mais peu flexible, tandis que les modèles Bayes A et LASSO bayésien peuvent améliorer la prédiction en présence de variables à effet marqué. La méthode SSVS, centrée sur la sélection de variables, privilégie l'interprétabilité au détriment d'une éventuelle perte marginale de performance prédictive.

6.2 Comparaison de la sélection de variables

Méthode	Type de sélection
RR-BLUP	Aucune (heuristique)
Bayes A	Indirecte
LASSO bayésien	Implicite
SSVS	Explicite

Table 2: Comparaison de la sélection de variables des différentes méthodes bayésiennes

Les méthodes Bayes A et LASSO bayésien permettent d'identifier des variables influentes via l'amplitude des coefficients, tandis que SSVS fournit une probabilité a posteriori d'inclusion, offrant une interprétation directe en termes de sélection de variables.

6.3 Synthèse

Cette comparaison met en évidence le compromis entre stabilité, flexibilité et interprétabilité des différentes approches bayésiennes. Le RR-BLUP est particulièrement adapté aux contextes fortement dimensionnés nécessitant des estimations stables, tandis que les modèles Bayes A et LASSO bayésien offrent un shrinkage plus adaptatif. La méthode SSVS se distingue par sa capacité à fournir une sélection explicite des variables, ce qui peut être un atout majeur lorsque l'objectif principal est l'interprétation.

7 Approche ABC (Approximate Bayesian Computation)

7.1 Objectif

L'objectif de cette question est de présenter et discuter l'approche ABC (Approximate Bayesian Computation), utilisée lorsque la vraisemblance du modèle est :

- inconnue,
- trop complexe à calculer,
- ou trop coûteuse numériquement.

Cette approche permet d'effectuer une inférence bayésienne approximative sans calcul explicite de la vraisemblance.

7.2 Principe général de l'approche ABC

L'approche ABC repose sur l'idée suivante : plutôt que de calculer la loi a posteriori à partir de la vraisemblance, on compare des données simulées à partir du modèle à des statistiques résumées des données observées. Le schéma général de cette approche est le suivant :

1. Simuler un paramètre θ^* depuis la loi a priori
2. Simuler des données $y^* | \theta^*$
3. Calculer des statistiques résumées sur ces données ($S(y^*)$)
4. Accepter θ^* si :

$$d(S(y^*), S(y)) \leq \epsilon$$

avec $d(\cdot, \cdot)$ une distance et ϵ , une mesure de tolérance. Les paramètres acceptés forment une approximation de la loi a posteriori.

7.3 Application au cadre de la régression bayésienne

Dans le contexte de la régression bayésienne, θ représente les paramètres du modèle (par exemple, β , μ , σ^2). Les statistiques résumées peuvent être : les coefficients estimés, la variance résiduelle, la corrélation entre valeurs observées et prédites. L'approche ABC permet ainsi de contourner le calcul explicite de la vraisemblance lorsque celle-ci est difficile à manipuler.

7.3.1 Rappel du principe ABC dans notre contexte

Dans notre problème de régression bayésienne, on considère le modèle :

$$Y = \mu \mathbb{I} + X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I})$$

Lorsque la vraisemblance est difficile à exploiter ou que l'on souhaite éviter les algorithmes MCMC classiques, l'approche ABC (Approximate Bayesian Computation) permet d'effectuer une inférence bayésienne par simulation, sans calcul explicite de la vraisemblance.

Le principe consiste à :

1. simuler des paramètres depuis les lois a priori ;
2. simuler des données à partir du modèle ;
3. comparer les données simulées aux données observées à l'aide d'une distance ;
4. accepter les paramètres si la distance est inférieure à un seuil fixé.

7.3.2 Choix méthodologiques pour notre application

Paramètres simulés

- $\mu \sim \mathcal{U}(-a, a)$
- $\beta \sim \mathcal{N}(0, V\mathbf{I}_p)$
- $\sigma^2 \sim \chi^2(b)$

Statistique résumée

Compte tenu de la dimension élevée du problème, on utilise une distance globale entre les données simulées et observées :

$$d(y, y^*) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y_i^*)^2}$$

Cette distance correspond à une erreur quadratique moyenne, simple et adaptée au cadre prédictif.

7.3.3 Algorithme ABC adapté

L'algorithme ABC utilisé est un *algorithm acceptance-rejet* :

1. Simuler $(\mu^*, \beta^*, \sigma^{2*})$
2. Simuler $y \mid \mu^*, \beta^*, \sigma^{2*}$
3. Calculer $d(y, y^*)$
4. Accepter si $d(y, y^*) \leq \epsilon$

7.3.4 Implémentation R

```
ABC_regression <- function(y, X, a, b, V, seuil, K) {
  n <- length(y)
  p <- ncol(X)
```

```

res_beta <- matrix(0, nrow = p, ncol = K)
res_mu <- numeric(K)
res_s2 <- numeric(K)

accept <- 0

for (k in 1:K) {

  # 1. Simulation des paramètres a priori
  mu_star <- runif(1, -a, a)
  s2_star <- rchisq(1, b)
  beta_star <- as.numeric(rmnorm(1, rep(0, p), V * diag(p)))

  # 2. Simulation des données
  y_star <- mu_star + X %*% beta_star + rnorm(n, 0, sqrt(s2_star))

  # 3. Distance
  dist <- sqrt(mean((y - y_star)^2))

  # 4. Acceptation
  if (dist < seuil) {
    accept <- accept + 1
    res_beta[, accept] <- beta_star
    res_mu[accept] <- mu_star
    res_s2[accept] <- s2_star
  }
}

return(list(
  beta = res_beta[, 1:accept, drop = FALSE],
  mu = res_mu[1:accept],
  sigma2 = res_s2[1:accept],
  taux_acceptation = 100 * accept / K
))
}

```

7.3.5 Application aux données

```

res_ABC <- ABC_regression(
  y = y_train,
  X = X_train,
  a = 5,
  b = 5,
  V = 10,
  seuil = 15,
  K = 5000
)

```

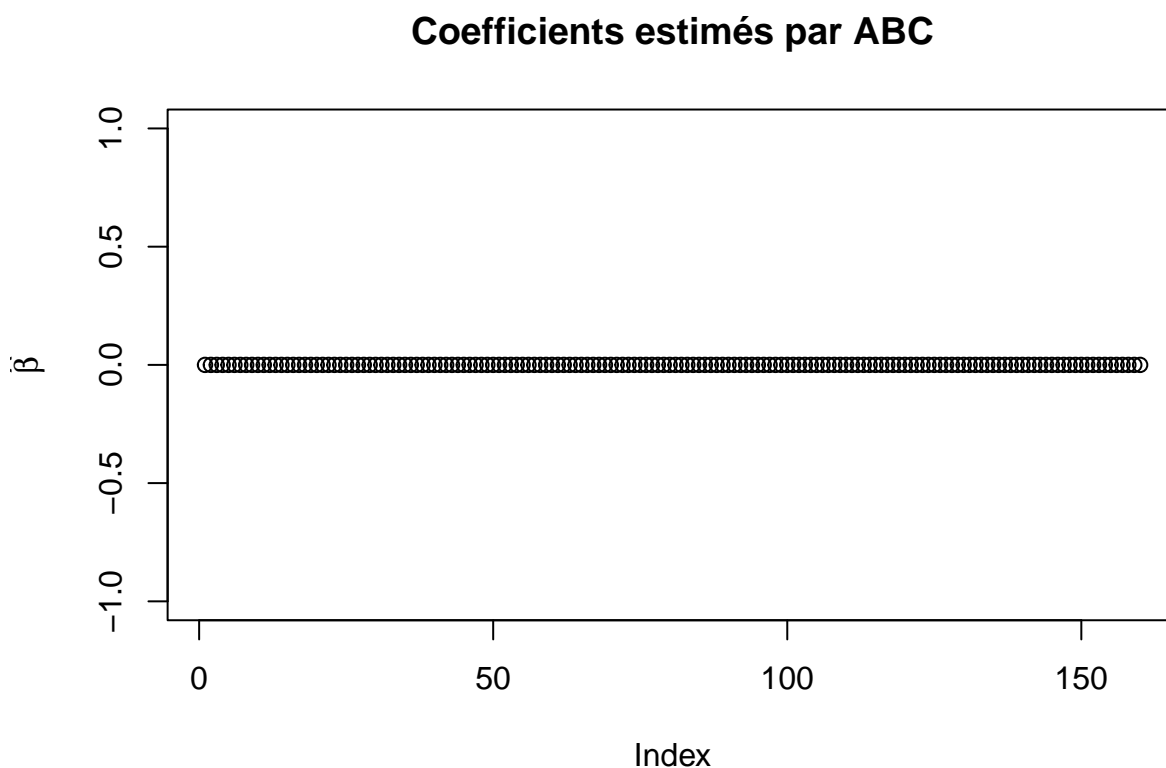
```
res_ABC$taux_acceptation
```

```
## [1] 0
```

7.3.6 Exploitation des résultats

```
beta_ABC <- rowMeans(res_ABC$beta)
mu_ABC <- mean(res_ABC$mu)

plot(sort(abs(beta_ABC)),
      main = "Coefficients estimés par ABC",
      ylab = expression(hat(beta)))
```



7.4 Avantages de l'approche ABC

- Pas de calcul explicite de la vraisemblance
- Applicable à des modèles complexes ou implicites
- Flexible dans le choix des statistiques résumées

7.5 Limites de l'approche ABC

- Résultats approximatifs, dépendants du seuil
- Choix délicat des statistiques résumées

- Coût computationnel élevé
- Sensibilité à la distance utilisée

7.6 Interprétation et positionnement par rapport aux méthodes précédentes

Contrairement aux méthodes bayésiennes étudiées précédemment, qui reposent sur des algorithmes MCMC exploitant explicitement la vraisemblance, l'approche ABC propose une alternative fondée sur la simulation. Si cette méthode est particulièrement utile dans des contextes où la vraisemblance est inaccessible, elle reste moins précise et plus coûteuse que les approches classiques lorsque celles-ci sont applicables.

8 Conclusion

Dans ce travail, plusieurs méthodes de régression bayésienne ont été appliquées à un jeu de données de grande dimension afin de prédire un score de satisfaction à partir de variables d'usage. L'approche RR-BLUP a servi de modèle de référence, offrant une estimation stable mais peu sélective. Le modèle Bayes A et le LASSO bayésien ont permis d'introduire un shrinkage adaptatif, améliorant la prise en compte des variables réellement influentes. Enfin, la méthode SSVS a apporté une sélection explicite des variables, facilitant l'interprétation des résultats.

Globalement, les résultats illustrent l'intérêt des méthodes bayésiennes dans les contextes où le nombre de variables dépasse le nombre d'observations. Le choix de la méthode dépend de l'objectif poursuivi : performance prédictive, interprétabilité ou sélection de variables. Ces approches constituent ainsi des outils particulièrement adaptés à l'analyse de données complexes et fortement dimensionnées.