BayesianImportance example

August Arnstad

05.11.2023

Contents


```
## Loading required package: usethis
```

```r
# If not already installed, install the 'devtools' package
if (!require(devtools)) install.packages("devtools")

# Install BayesianImportance
devtools::install_github("AugustArnstad/BayesianImportance")
```

SIMULATE DATA

```r
library(BayesianImportance)
library(INLA)
library(mnormt)
library(ggplot2)
set.seed(1234)

n <- 1000
nclass_gamma <- 50
nclass_eta <- 100

mu <- c(1, 2)
sigma <- matrix(c(1, 0.9, 0.9, 1), 2, 2)

# Sample a standardized correlated design matrix
X <- rmnorm(n, mu, sigma)

# Add random effects
gamma <- rep(rnorm(nclass_gamma, 0, sqrt(1)), each = n/nclass_gamma)
eta <- rep(rnorm(nclass_eta, 0, sqrt(1)), each = n/nclass_eta)
epsilon = rnorm(n, mean = 0, sd = 1)

# Define some formula
Y <- 1 + 1 * X[, 1] + sqrt(2) * X[, 2] + gamma + eta + epsilon  # write epsilon as a random effect

# Collect as a dataframe
data_bayes = data.frame(cbind(Y, X = X))
data_bayes = data.frame(cbind(data_bayes, gamma = gamma))
```

```r
data_bayes = data.frame(cbind(data_bayes, eta = eta))
```

```r
test <- lm(Y ~ V2 + V3, data = data_bayes)
summary(test)
```

```
##
## Call:
## lm(formula = Y ~ V2 + V3, data = data_bayes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3902 -1.1559  0.0681  1.1425  5.2723
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0830     0.1698   6.378 2.74e-10 ***
## V2            0.9413     0.1381   6.815 1.63e-11 ***
## V3            1.3975     0.1381  10.119  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.822 on 997 degrees of freedom
## Multiple R-squared:  0.6219, Adjusted R-squared:  0.6211
## F-statistic: 819.8 on 2 and 997 DF,  p-value: < 2.2e-16
```

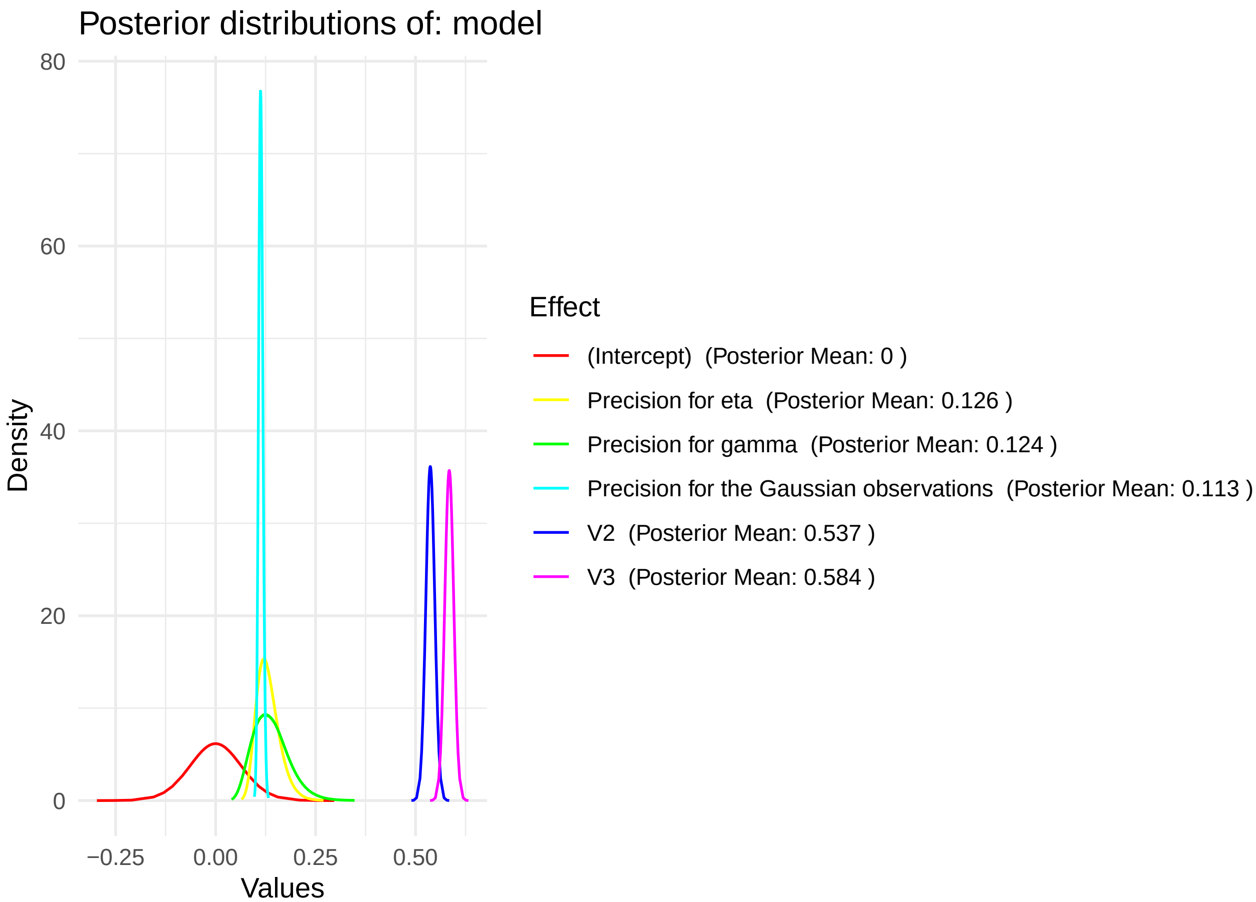
```r
library(relaimpo)
relaimpo::calc.relimp(test)
```

```
## Response variable: Y
## Total response variance: 8.764219
## Analysis based on 1000 observations
##
## 2 Regressors:
## V2 V3
## Proportion of variance explained by model: 62.19%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##          lmg
## V2 0.3003295
## V3 0.3215457
##
## Average coefficients for different model sizes:
##
##          1X       2Xs
## V2 2.216449 0.9412968
## V3 2.256335 1.3975197
```

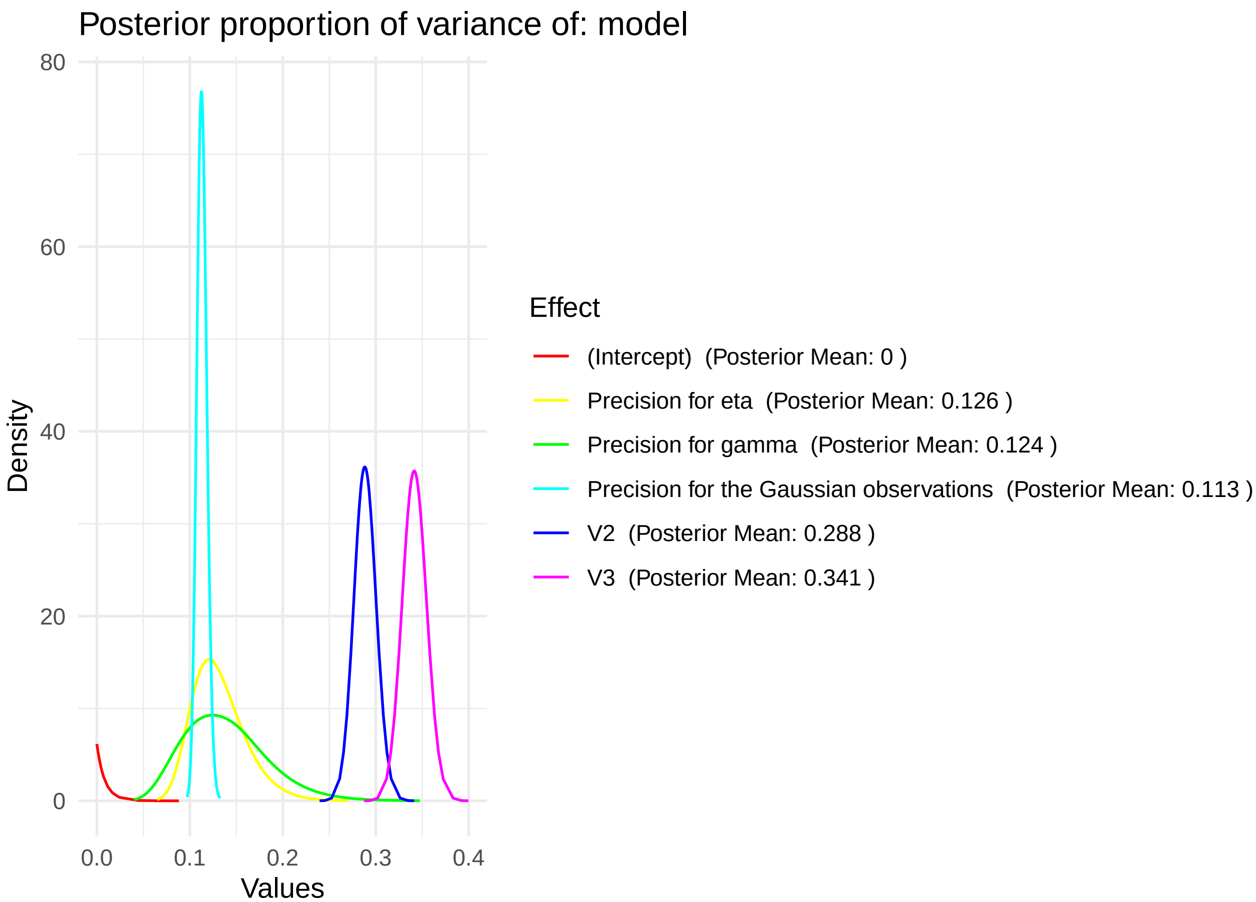
USAGE

```r
set.seed(1234)
model <- run_bayesian_imp(Y ~ V2 + V3 + (1 | gamma) + (1 | eta), data = data_bayes)
```

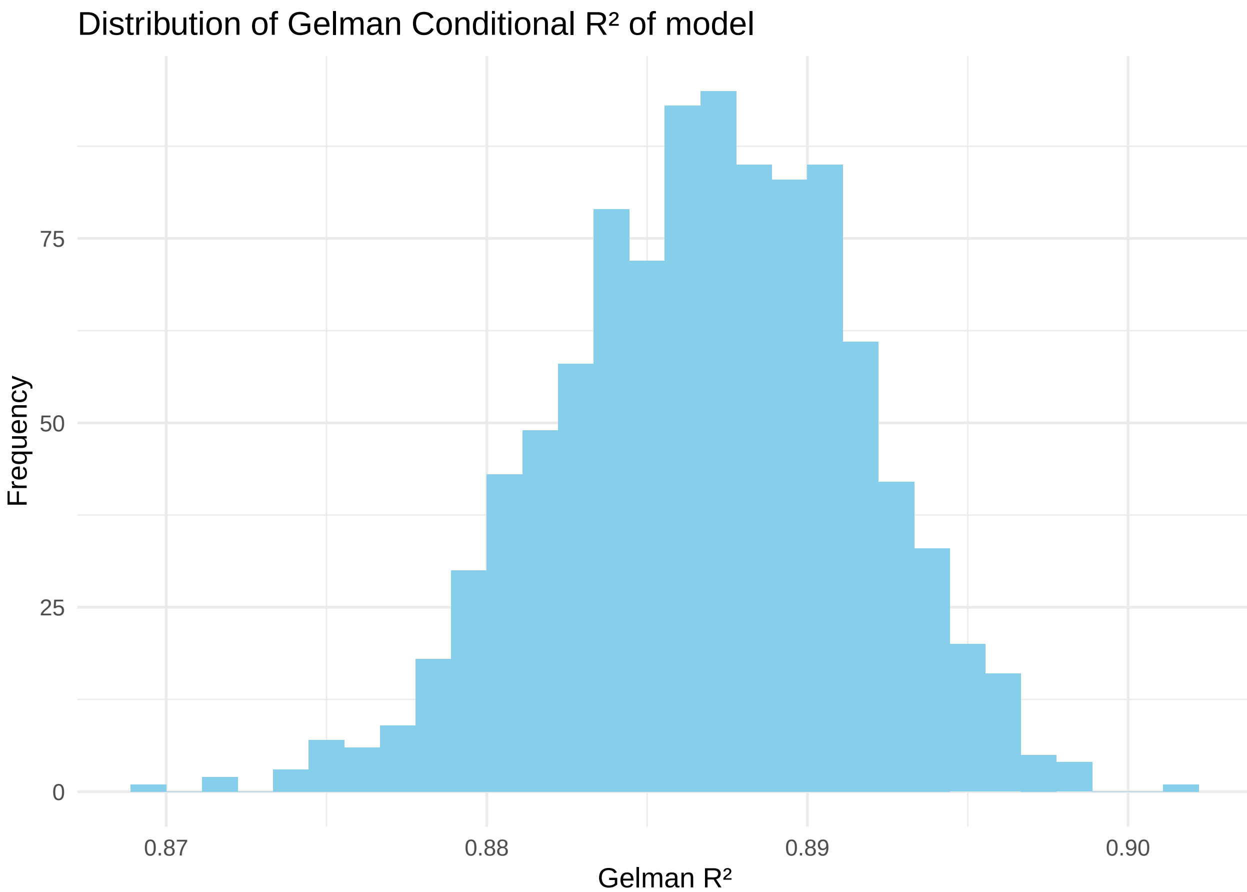
```r
plot_model = plot_posteriors(model, importance = FALSE)
plot_model$posterior_plot
```




```r
plot_model = plot_posteriors(model, importance = TRUE)
plot_model$posterior_plot
```




```
gelman_r2 = gelman_r2_metrics(model, s = 1000, plot = TRUE)
gelman_r2$plot
```




```
summary(gelman_r2$conditional_gelman_r2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.8691  0.8837  0.8871  0.8868  0.8903  0.9014
```

```
rmarkdown::render(input = "BayesianImportanceExample.Rmd", output_format = "pdf_document")
```

How do I set up the design matrix for the random effects Z?

$$Y = X\beta + Z\gamma$$

Need help with this!

Am I right now just working with random intercepts? Is that what my gamma values truly represent?

In the run_INLA function I run INLA on the Z matrix(SVD approximation of X). Is it correct to run INLA on Z? Right now I transform right before plotting/giving the summary, keeping the beta coefficients from Z before that, since that is the output of the INLA model.

Struggling to make the priors work if they are not simply the default value. Now its just the basic version. This should probably be fixed as it is very limiting. In general I am a bit unsure if my script is efficient, as I have made it pretty quickly. Need a review of this.

In addition I need general comments on my package as there might be stuff missing that I am not aware of/could be improved.