

BayesianImportance example

August Arnstad

24.11.2023

Contents

SIMULATE DATA	1
USAGE	3

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

```
# If not already installed, install the 'devtools' package
```

```
if (!require(devtools)) install.packages("devtools")
```

```
# Install BayesianImportance
```

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

SIMULATE DATA

```
library(BayesianImportance)
```

```
library(INLA)
```

```
library(mnormt)
```

```
library(ggplot2)
```

```
set.seed(1234)
```

```
n <- 10000
```

```
nclass_gamma <- 200
```

```
nclass_eta <- 100
```

```
mu <- c(1, 2, 3)
```

```
sigma <- matrix(c(1, 0.9, 0.9, 0.9, 1, 0.9, 0.9, 0.9, 1), 3, 3)
```

```
# 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 = sqrt(1))
```

```
# Define some formula
```

```
Y <- 1 + 1 * X[, 1] + sqrt(2) * X[, 2] + sqrt(3) * X[, 3] + gamma + eta +  
  epsilon
```

```
# Collect as a dataframe
```

```
data_bayes = data.frame(cbind(Y, X = X))
```

```

data_bayes = data.frame(cbind(data_bayes, gamma = gamma))
data_bayes = data.frame(cbind(data_bayes, eta = eta))

test <- lm(Y ~ V2 + V3 + V4, data = data_bayes)
summary(test)

##
## Call:
## lm(formula = Y ~ V2 + V3 + V4, data = data_bayes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.6392 -1.2435 -0.0132  1.1926  5.8460
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.13125    0.08913   12.69  <2e-16 ***
## V2             1.04090    0.04611   22.57  <2e-16 ***
## V3             1.36404    0.04605   29.62  <2e-16 ***
## V4             1.68774    0.04645   36.33  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.764 on 9996 degrees of freedom
## Multiple R-squared:  0.8354, Adjusted R-squared:  0.8353
## F-statistic: 1.691e+04 on 3 and 9996 DF, p-value: < 2.2e-16

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

## Response variable: Y
## Total response variance: 18.89101
## Analysis based on 10000 observations
##
## 3 Regressors:
## V2 V3 V4
## Proportion of variance explained by model: 83.54%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##          lmg
## V2 0.2686478
## V3 0.2779598
## V4 0.2887473
##
## Average coefficients for different model sizes:
##
##          1X          2Xs          3Xs
## V2 3.793494 1.760680 1.040897
## V3 3.830943 2.003873 1.364036
## V4 3.851704 2.269482 1.687741

```

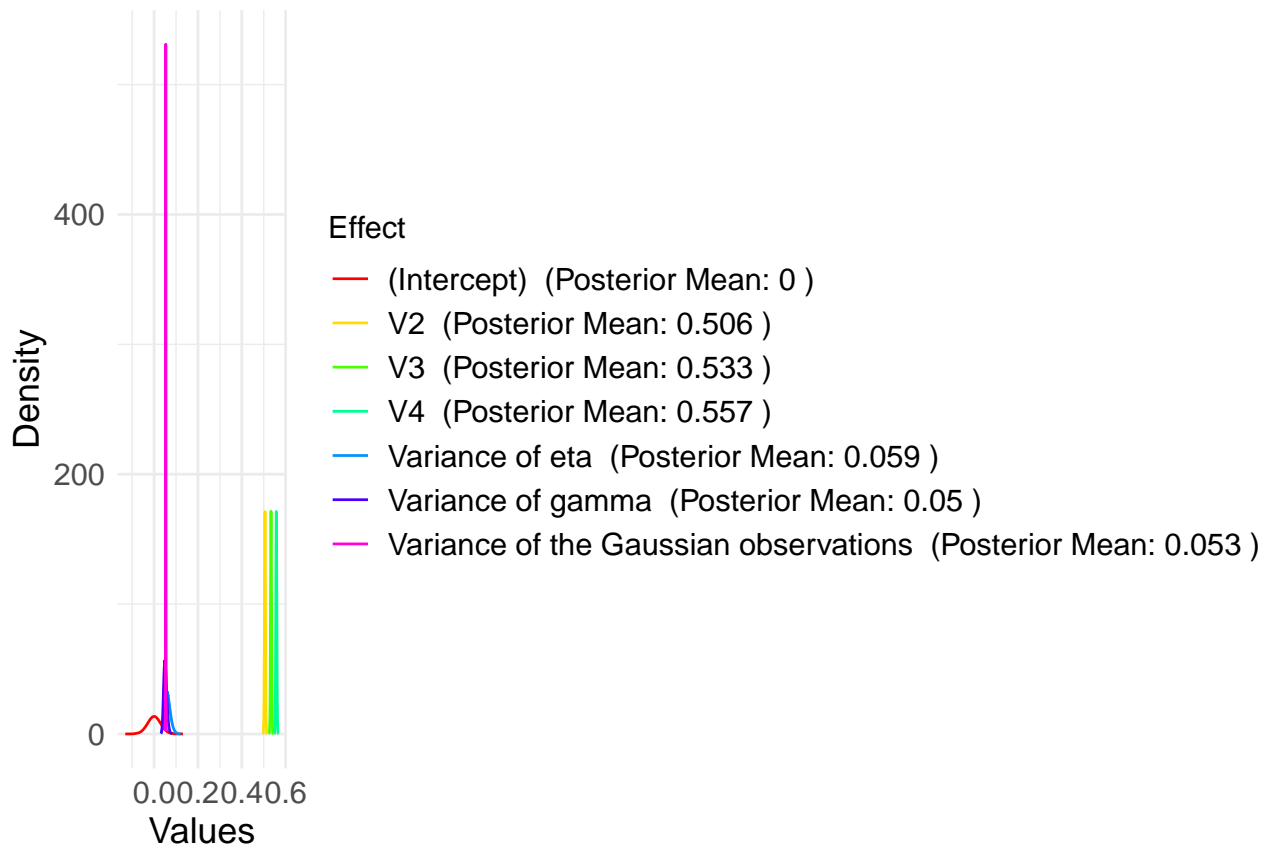
USAGE

```
set.seed(1234)
model <- run_bayesian_imp(Y ~ V2 + V3 + V4 + (1 | gamma) + (1 | eta),
  data = data_bayes)
```

*# Note that it says precision of the random effects, while this is
the variance!*

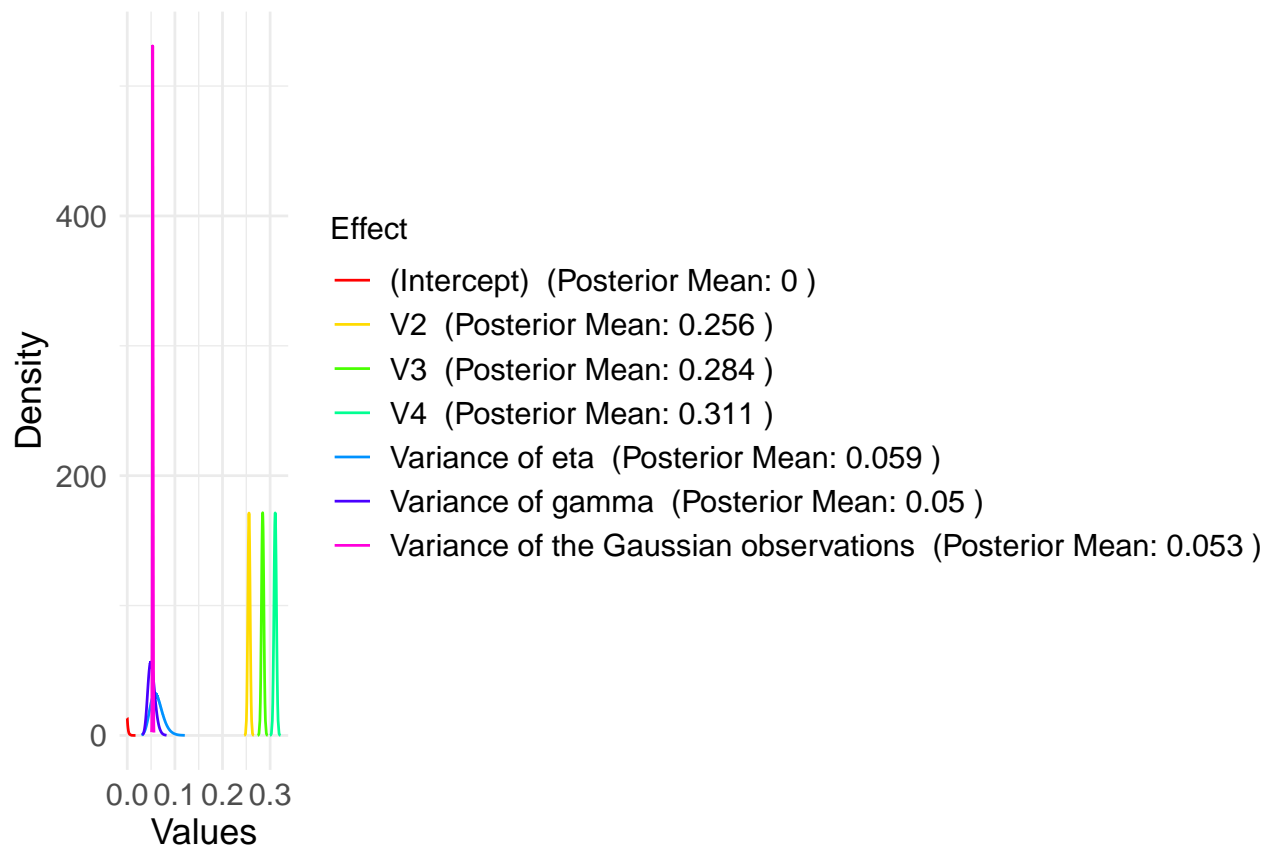
```
plot_model = plot_posteriors(model, importance = FALSE)
plot_model$posterior_plot
```

Posterior distributions



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

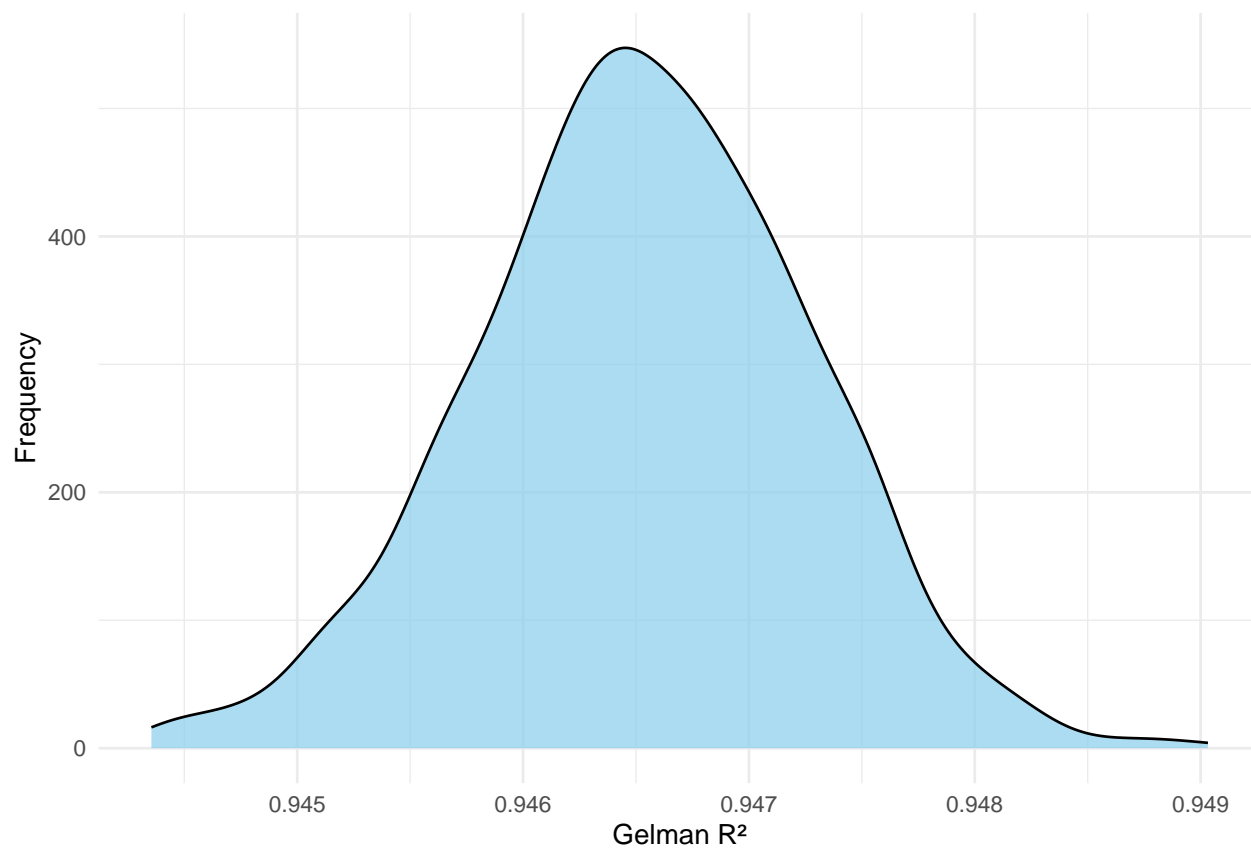
Posterior proportion of variance



```
sampled_posteriors = BayesianImportance::sample_posteriors(Y ~ V2 + V3 +
  V4 + (1 | gamma) + (1 | eta), data = data_bayes, n_samp = 5000, n,
  n_classes = (200 + 100))
```

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

Distribution of Gelman Conditional R^2 of model



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

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.9444  0.9461  0.9465  0.9465  0.9470  0.9490
```

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