BayesianImportance example

August Arnstad

27.10.2023

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, 0, 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))
data_bayes = data.frame(cbind(data_bayes, eta = eta))
```

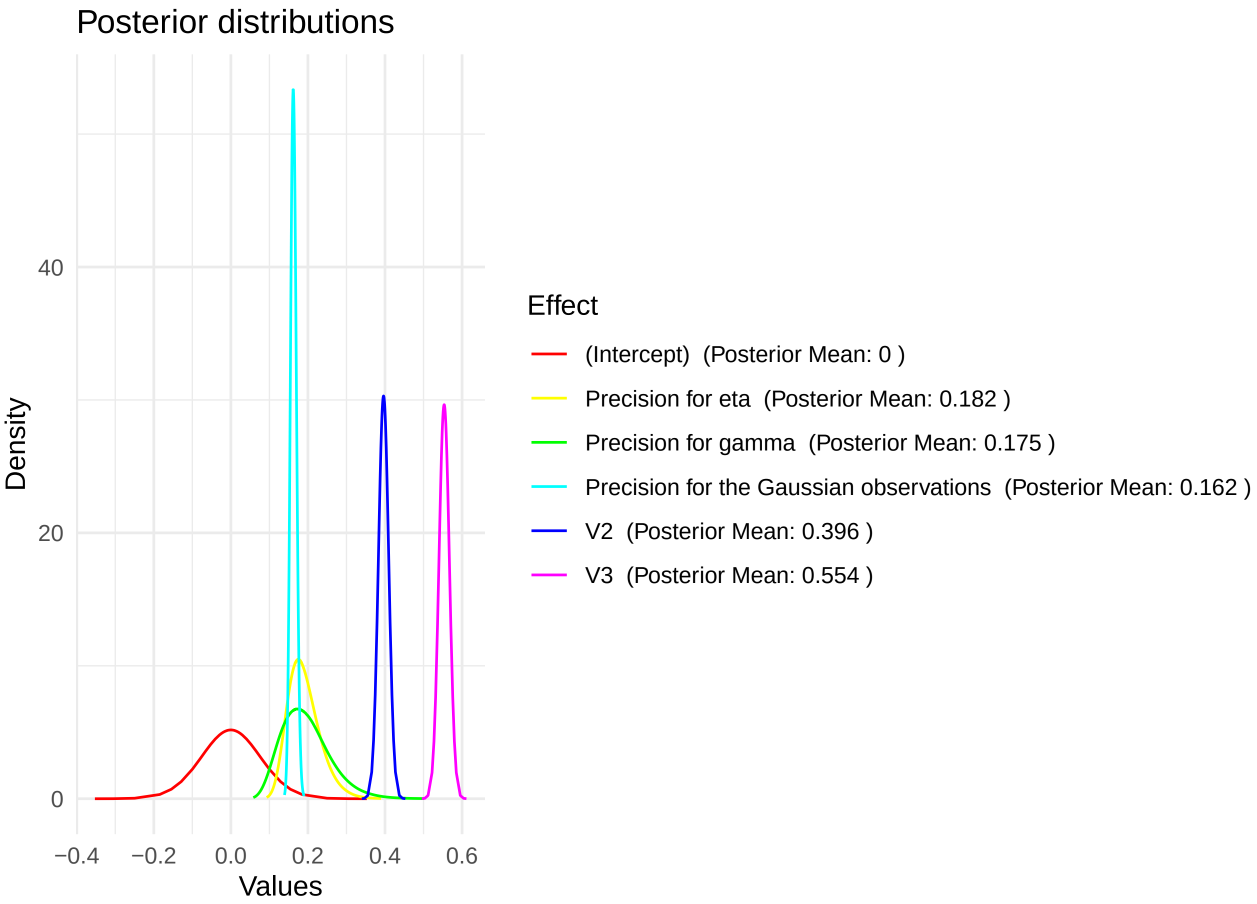
USAGE

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

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

```
## Mean            6.16636
## Stdev           0.290727
## Quantile  0.025 5.61234
## Quantile  0.25  5.96655
## Quantile  0.5   6.16023
## Quantile  0.75  6.35852
## Quantile  0.975 6.75192
## Mean            5.7054
## Stdev           1.98246
## Quantile  0.025 2.93903
## Quantile  0.25  4.30086
## Quantile  0.5   5.32049
## Quantile  0.75  6.69381
## Quantile  0.975 10.5795
## Mean            5.50388
## Stdev           1.20014
## Quantile  0.025 3.47691
## Quantile  0.25  4.64651
## Quantile  0.5   5.39388
## Quantile  0.75  6.23242
## Quantile  0.975 8.15688
## Mean            -1.40174e-13
## Stdev           0.0783614
## Quantile  0.025 -0.154428
## Quantile  0.25  -0.0524047
## Quantile  0.5   -0.000172405
## Quantile  0.75  0.0520599
## Quantile  0.975 0.154083
## Mean            0.396273
## Stdev           0.0132019
## Quantile  0.025 0.370358
## Quantile  0.25  0.387344
## Quantile  0.5   0.396246
## Quantile  0.75  0.405147
## Quantile  0.975 0.422132
## Mean            0.55355
## Stdev           0.0134942
## Quantile  0.025 0.52706
## Quantile  0.25  0.544423
## Quantile  0.5   0.553522
## Quantile  0.75  0.562621
## Quantile  0.975 0.57998
```

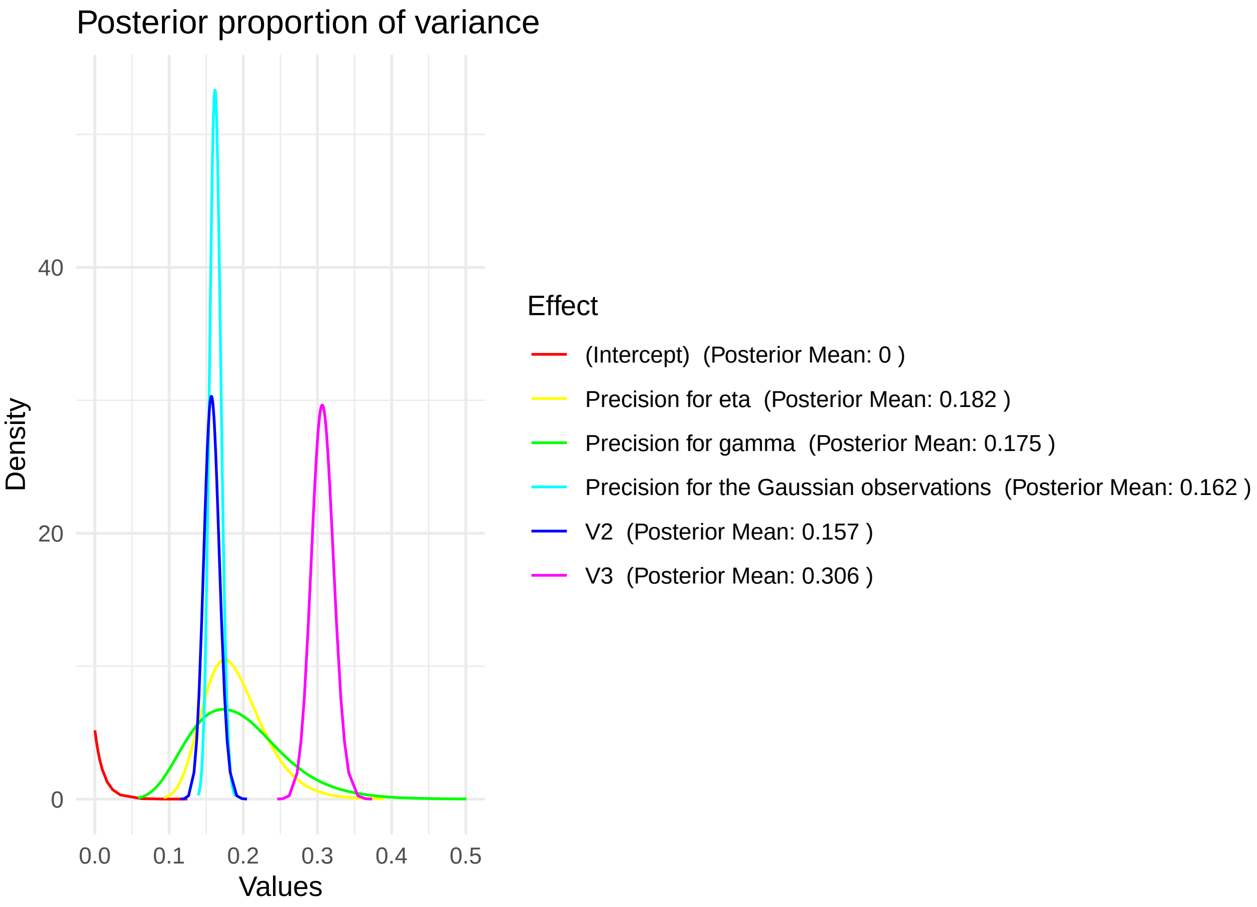
```
plot_model$posterior_plot
```



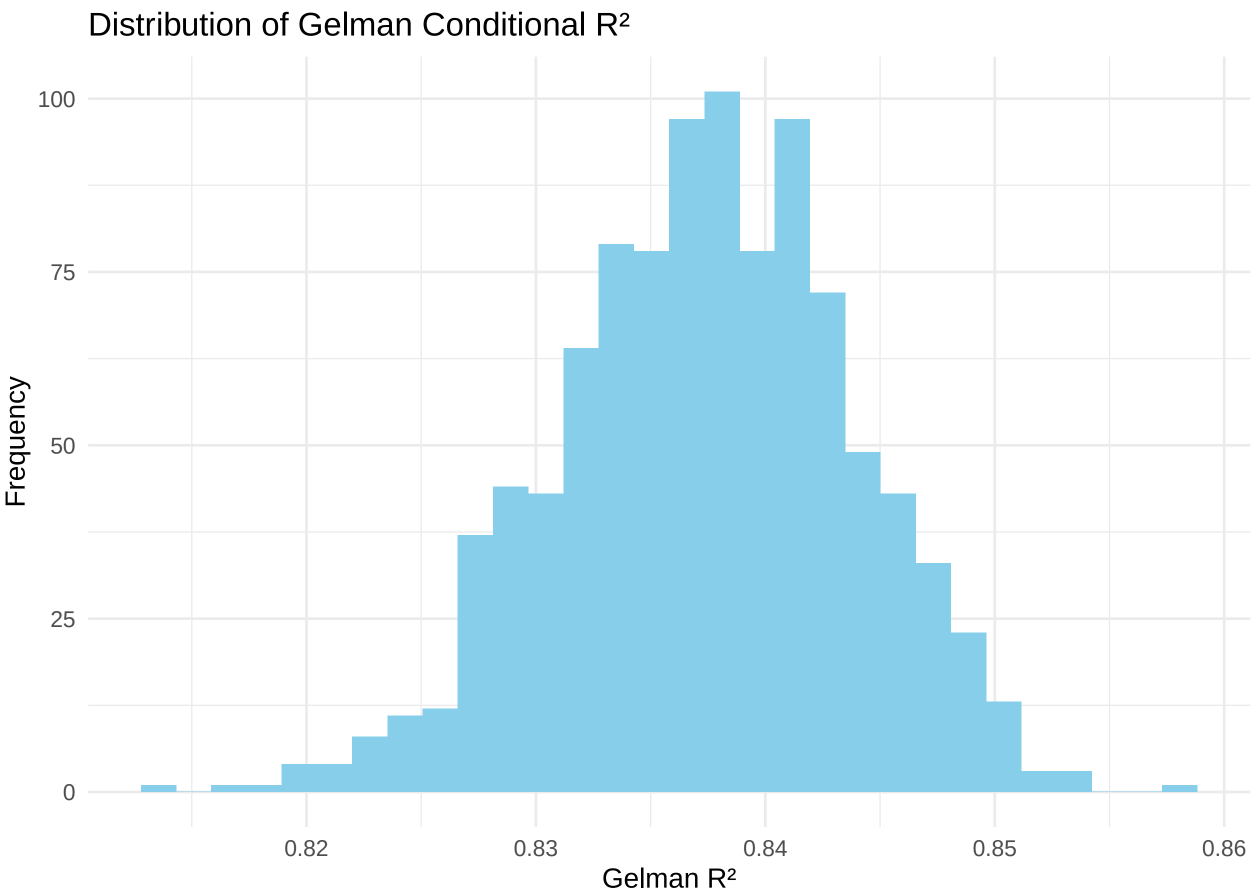
```
plot_model = plot_posteriors(model, importance = TRUE)
```

```
## Mean           6.16636
## Stdev          0.290727
## Quantile  0.025 5.61234
## Quantile  0.25  5.96655
## Quantile  0.5   6.16023
## Quantile  0.75  6.35852
## Quantile  0.975 6.75192
## Mean           5.7054
## Stdev          1.98246
## Quantile  0.025 2.93903
## Quantile  0.25  4.30086
## Quantile  0.5   5.32049
## Quantile  0.75  6.69381
## Quantile  0.975 10.5795
## Mean           5.50388
## Stdev          1.20014
## Quantile  0.025 3.47691
## Quantile  0.25  4.64651
## Quantile  0.5   5.39388
## Quantile  0.75  6.23242
## Quantile  0.975 8.15688
## Mean            -1.40174e-13
## Stdev           0.0783614
## Quantile  0.025 -0.154428
## Quantile  0.25  -0.0524047
## Quantile  0.5   -0.000172405
## Quantile  0.75  0.0520599
## Quantile  0.975 0.154083
## Mean            0.396273
## Stdev           0.0132019
## Quantile  0.025 0.370358
## Quantile  0.25  0.387344
## Quantile  0.5   0.396246
## Quantile  0.75  0.405147
## Quantile  0.975 0.422132
## Mean            0.55355
## Stdev           0.0134942
## Quantile  0.025 0.52706
## Quantile  0.25  0.544423
## Quantile  0.5   0.553522
## Quantile  0.75  0.562621
## Quantile  0.975 0.57998
```

```
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. 
##  0.8131  0.8331  0.8376  0.8374  0.8418  0.8576
```

```
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.