VariableImportanceINLA draft

August Arnstad

15.03.2024

Contents

Questions and deficiencies

1) The package so far only works for LMMs. I need to get a dataset I can use for binomial and Poisson GLMMs.
2) The package cannot handle categorical covariates, as the relative weights part of the package is not capable of handling this yet
3) Is the heritability caluclated correctly? In this equation

$$h^2 = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2}$$

   does σ_e^2 refer to only residual variance, or does it also include the variance of the random effects? I think it should only include the residual variance, but I am not sure. Currently the heritability is implemented using all random effects, not just the residual variance.
4) Should I compute multiple heritabilities? Currently, I let the user specify the "additive parameter", and the heritability is calculated using this as the additive dispersion parameter. However, I could also calculate the heritability for all random effects, and let the user decide which one to use. Does this make sense?

Example code for binary model

```r
library(INLA)
library(ggplot2)
library(reshape2)
library(tidyr)
set.seed(1)

# Number of observations and groups
n <- 10000
n_groups <- 100

# Simulate fixed effects
X1 <- rnorm(n, mean = 0, sd = 1)
X2 <- rnorm(n, mean = 0, sd = 1)
X3 <- rnorm(n, mean = 0, sd = 1)
```

```r
# Simulate random effects groups
Z1 <- sample(1:n_groups, n, replace = TRUE)
Z2 <- sample(1:n_groups, n, replace = TRUE)


# Coefficients for fixed effects
beta0 <- 1
beta1 <- 1
beta2 <- 1
beta3 <- 1

# Coefficients for random effects (standard deviations)
sigma_z1 <- 1
sigma_z2 <- 1

# Simulate outcome variable Y
random_effect_contributions_z1 <- rnorm(n_groups, mean = 0, sd = sigma_z1)[Z1]
random_effect_contributions_z2 <- rnorm(n_groups, mean = 0, sd = sigma_z2)[Z2]
error_term <- rnorm(n, mean = 0, sd = 1)

eta <- beta0 + beta1 * X1 + beta2 * X2 + beta3 * X3 + random_effect_contributions_z1 +
    random_effect_contributions_z2 + error_term

probit_p <- pnorm(eta)

# Simulate binomial response based on these probabilities
y_bin <- rbinom(n, size = 1, prob = probit_p)

# Combine into a data frame
data_inla_binomal <- data.frame(y_bin, X1, X2, X3, Z1, Z2)  #, Z3)
```

```r
lr_formula <- y_bin ~ X1 + X2 + X3
lmm_1re_formula <- y_bin ~ X1 + X2 + X3 + f(Z1, model = "iid")
lmm_2re_formula <- y_bin ~ X1 + X2 + X3 + f(Z1, model = "iid") + f(Z2,
    model = "iid")
lmm_0fe_formula <- y_bin ~ f(Z1, model = "iid") + f(Z2, model = "iid")

# prior = list(theta = list(initial=log(0.5), prior='pc.prec',
# param=c(sqrt(2),0.05)))

model_bin_lm <- VariableImportanceINLA::perform_inla_analysis(data_inla_binomal,
    lr_formula, family = "binomial", link_func = "probit")
model_bin_lmm1 <- VariableImportanceINLA::perform_inla_analysis(data_inla_binomal,
    lmm_1re_formula, family = "binomial", link_func = "probit")
model_bin_lmm2 <- VariableImportanceINLA::perform_inla_analysis(data_inla_binomal,
    lmm_2re_formula, family = "binomial", link_func = "probit")
model_bin_lmm0 <- VariableImportanceINLA::perform_inla_analysis(data_inla_binomal,
    lmm_0fe_formula, family = "binomial", link_func = "probit")
```

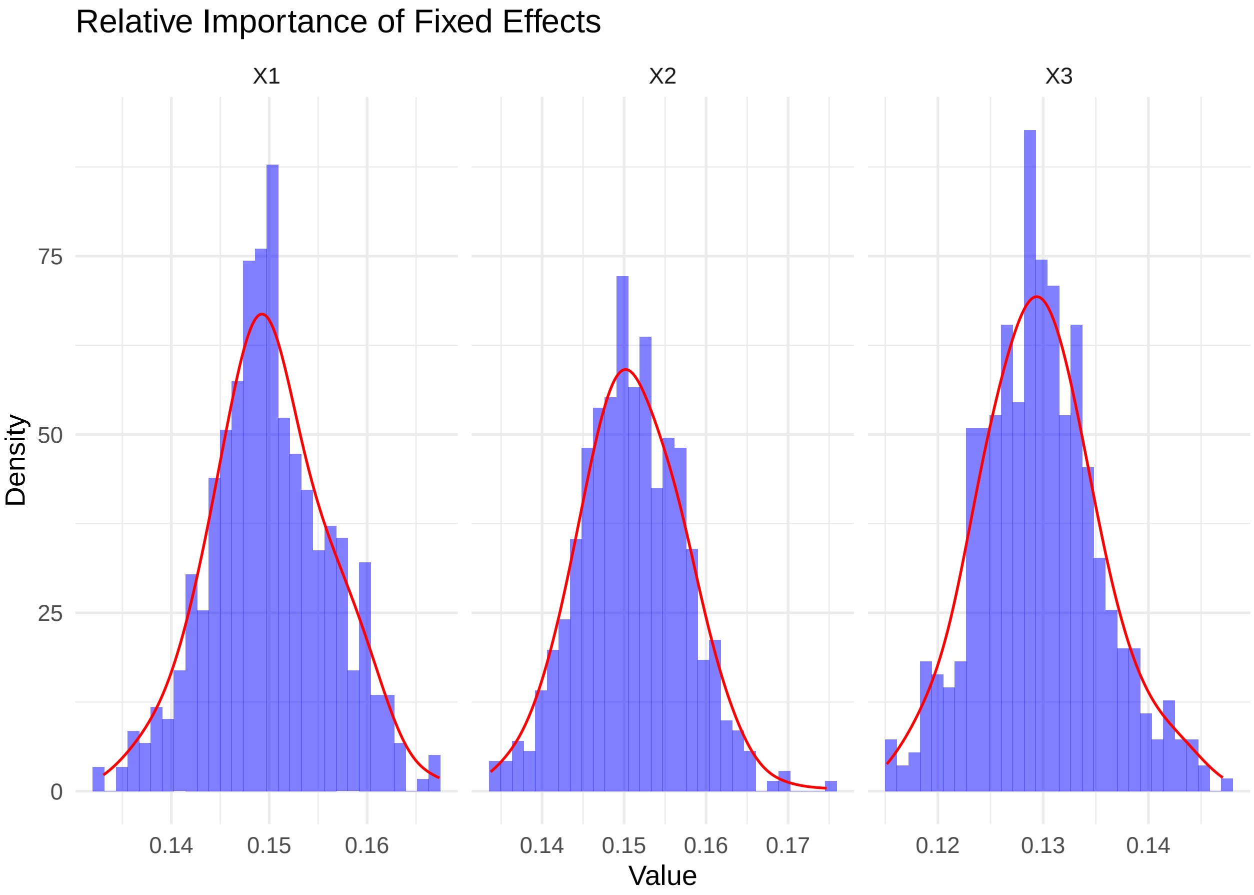
```r
bin_lm_samples <- VariableImportanceINLA::sample_posterior_count(model_bin_lm,
    lr_formula, data_inla_binomal, n_samp = 500)
```

```
## [1] "No random effects, only resiudal variance"
```

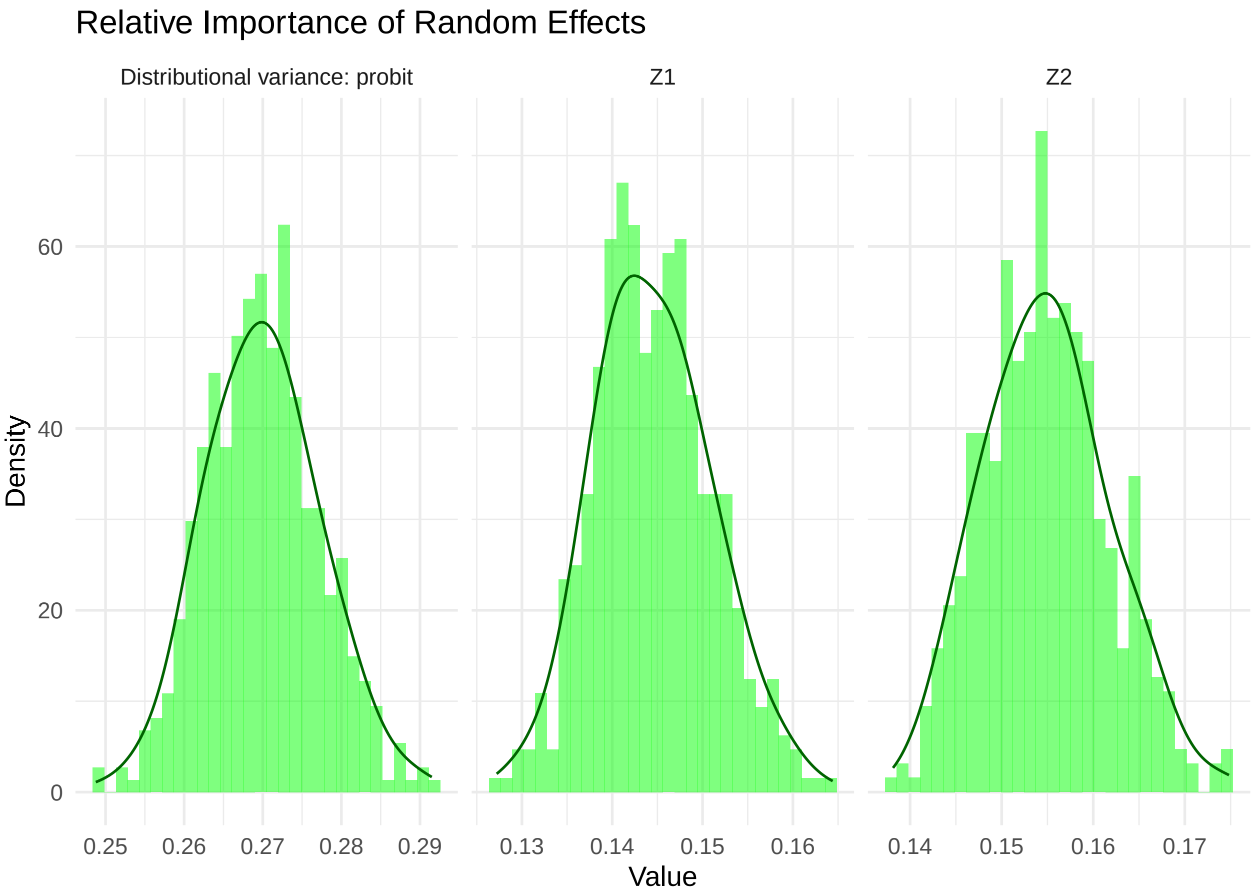
```r
bin_lmm1_samples <- VariableImportanceINLA::sample_posterior_count(model_bin_lmm1,
    lmm_1re_formula, data_inla_binomal, n_samp = 500, additive_param = "Z1")

bin_lmm2_samples <- VariableImportanceINLA::sample_posterior_count(model_bin_lmm2,
    lmm_2re_formula, data_inla_binomal, n_samp = 500, additive_param = "Z1")

bin_lmm0_samples <- VariableImportanceINLA::sample_posterior_count(model_bin_lmm0,
    lmm_0fe_formula, data_inla_binomal, n_samp = 500, additive_param = "Z1")
```

```r
plot_bin_lm <- VariableImportanceINLA::plot_samples(bin_lm_samples)

plot_bin_lmm1 <- VariableImportanceINLA::plot_samples(bin_lmm1_samples)

plot_bin_lmm2 <- VariableImportanceINLA::plot_samples(bin_lmm2_samples)

plot_bin_lmm0 <- VariableImportanceINLA::plot_samples(bin_lmm0_samples)
```

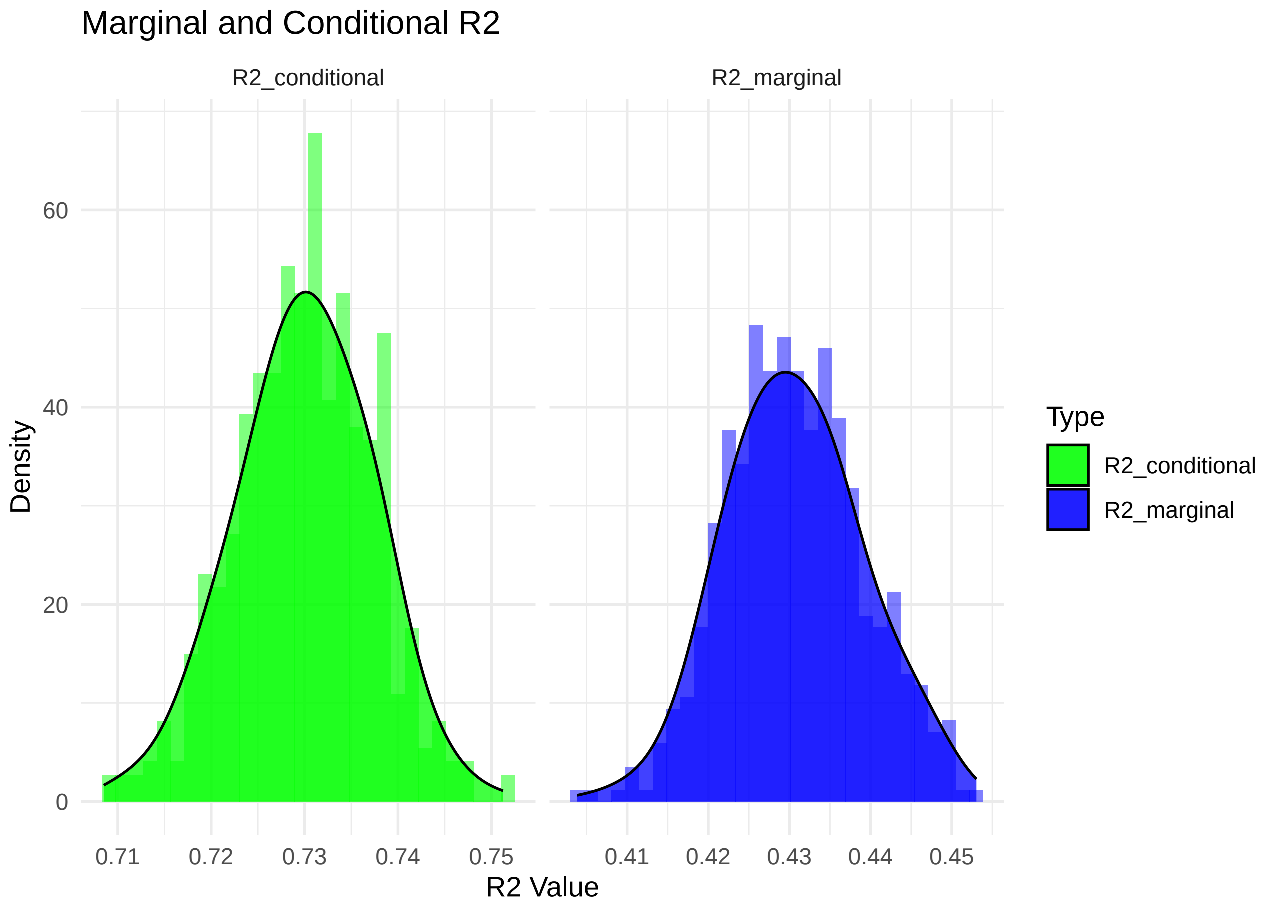
```r
plot_bin_lmm2$fixed_effects
```



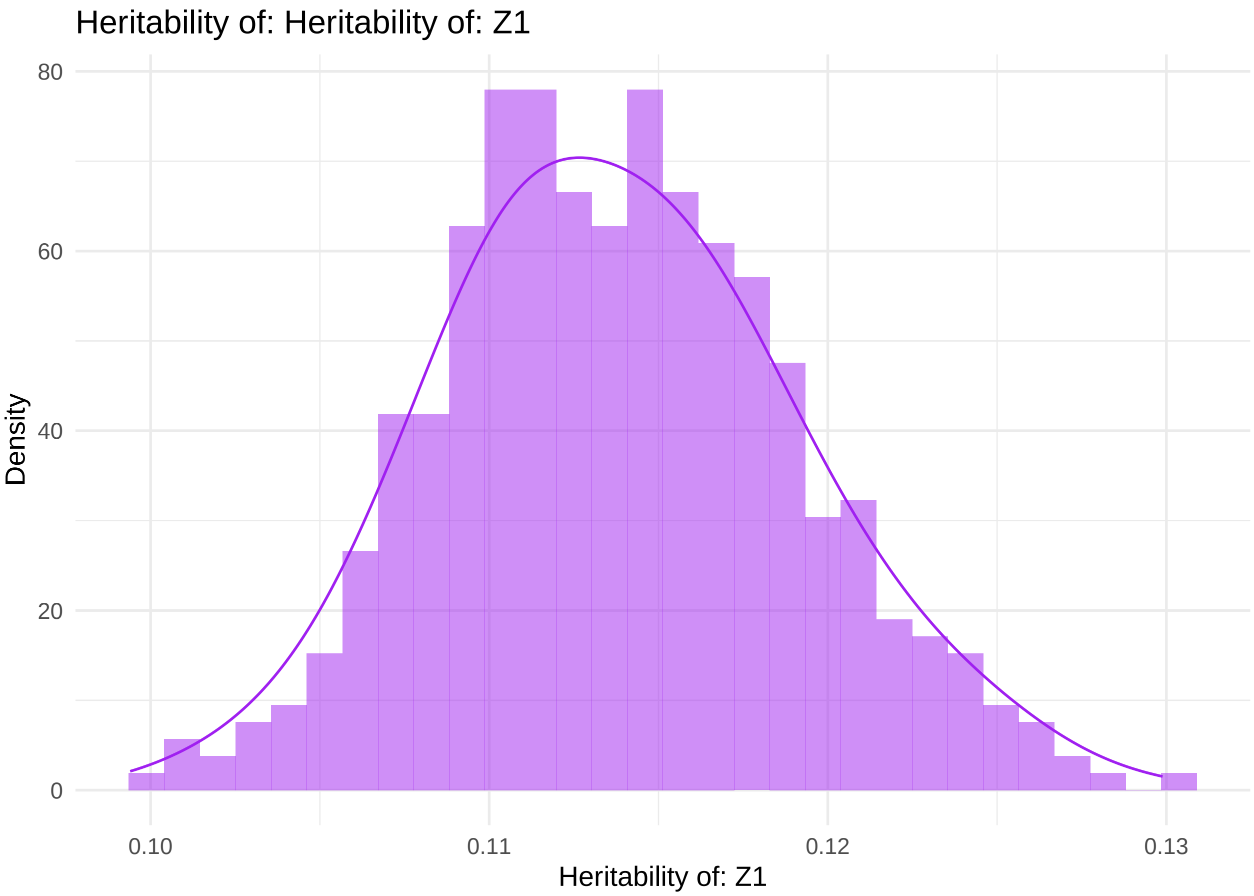
```r
plot_bin_lmm2$random_effects
```



```
plot_bin_lmm2$R2
```



```
plot_bin_lmm2$heritability
```



Example code for poisson model

```
library(INLA)
library(ggplot2)
library(reshape2)
library(tidyr)
set.seed(1)

# Number of observations and groups
n <- 10000
n_groups <- 100

# Simulate fixed effects
X1 <- rnorm(n, mean = 0, sd = 1)
X2 <- rnorm(n, mean = 0, sd = 1)
X3 <- rnorm(n, mean = 0, sd = 1)

# Simulate random effects groups
Z1 <- sample(1:n_groups, n, replace = TRUE)
Z2 <- sample(1:n_groups, n, replace = TRUE)


# Coefficients for fixed effects
beta0 <- 1
beta1 <- 1
```

```
beta2 <- 1
beta3 <- 1

# Coefficients for random effects (standard deviations)
sigma_z1 <- 1
sigma_z2 <- 1

# Simulate outcome variable Y
random_effect_contributions_z1 <- rnorm(n_groups, mean = 0, sd = sigma_z1)[Z1]
random_effect_contributions_z2 <- rnorm(n_groups, mean = 0, sd = sigma_z2)[Z2]
error_term <- rnorm(n, mean = 0, sd = 1)

eta <- beta0 + beta1 * X1 + beta2 * X2 + beta3 * X3 + random_effect_contributions_z1 +
    random_effect_contributions_z2 + error_term

# Linear predictor and expected count
lambda <- exp(eta)

# Simulate Poisson response
y_pois <- rpois(n, lambda = lambda)


# Combine into a data frame
data_inla_poisson <- data.frame(y_pois, X1, X2, X3, Z1, Z2)  #, Z3)
```

```
lr_formula <- y_pois ~ X1 + X2 + X3
lmm_1re_formula <- y_pois ~ X1 + X2 + X3 + f(Z1, model = "iid")
lmm_2re_formula <- y_pois ~ X1 + X2 + X3 + f(Z1, model = "iid") + f(Z2,
    model = "iid")
lmm_0fe_formula <- y_pois ~ f(Z1, model = "iid") + f(Z2, model = "iid")

# prior = list(theta = list(initial=log(0.5), prior='pc.prec',
# param=c(sqrt(2),0.05)))

model_pois_lm <- VariableImportanceINLA::perform_inla_analysis(data_inla_poisson,
    lr_formula, family = "poisson", link_func = "log")
model_pois_lmm1 <- VariableImportanceINLA::perform_inla_analysis(data_inla_poisson,
    lmm_1re_formula, family = "poisson", link_func = "log")
model_pois_lmm2 <- VariableImportanceINLA::perform_inla_analysis(data_inla_poisson,
    lmm_2re_formula, family = "poisson", link_func = "log")
model_pois_lmm0 <- VariableImportanceINLA::perform_inla_analysis(data_inla_poisson,
    lmm_0fe_formula, family = "poisson", link_func = "log")
```

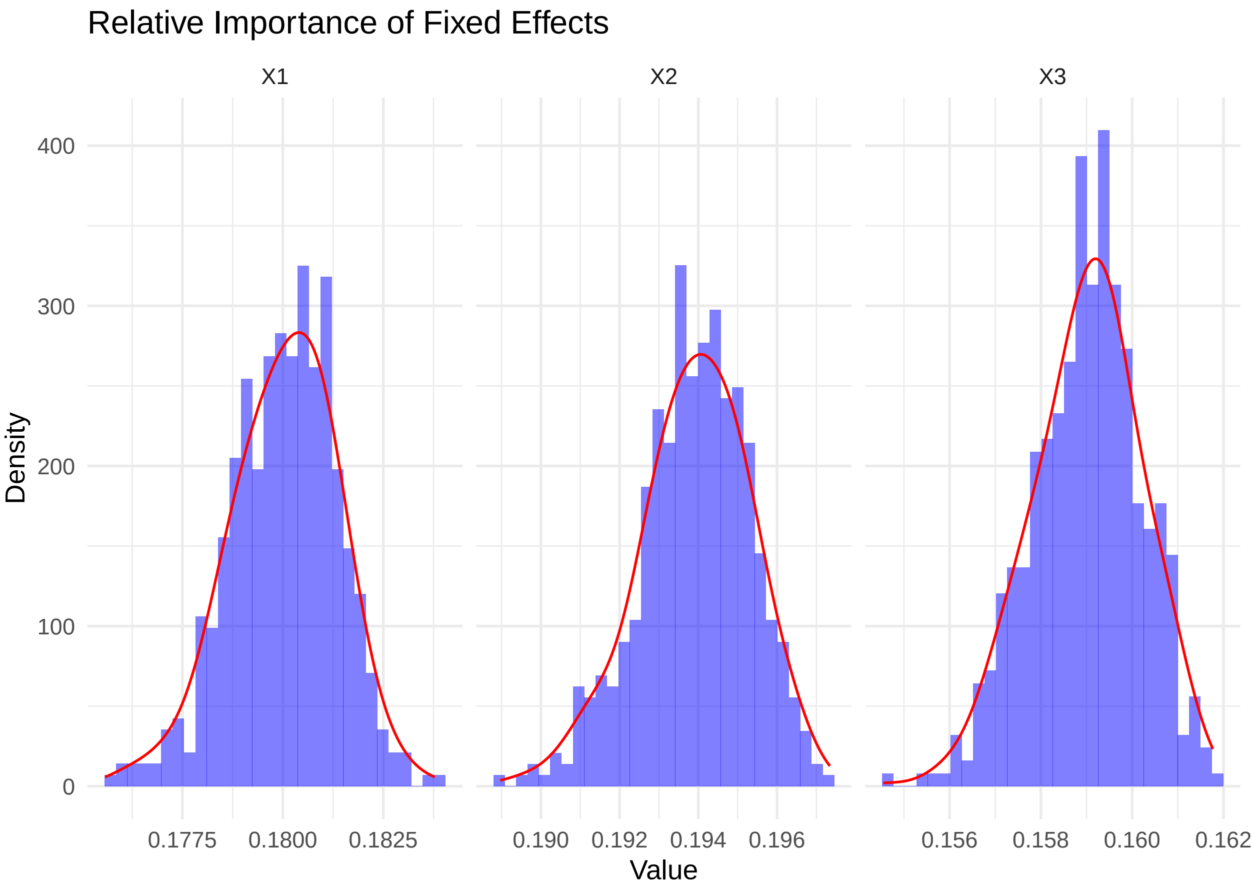
```
pois_lm_samples <- VariableImportanceINLA::sample_posterior_count(model_pois_lm,
    lr_formula, data_inla_poisson, n_samp = 500)
```

```
## [1] "No random effects, only resiudal variance"
```

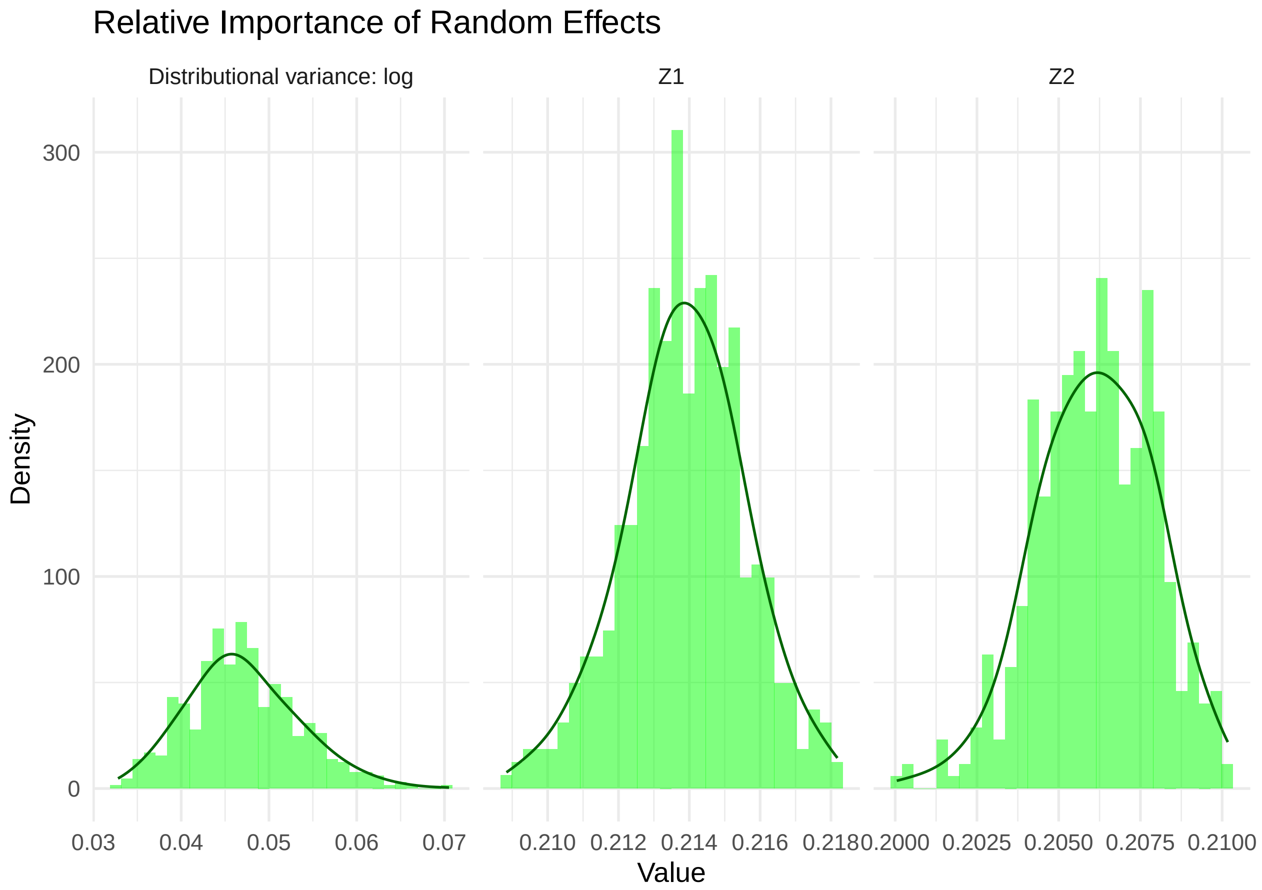
```
pois_lmm1_samples <- VariableImportanceINLA::sample_posterior_count(model_pois_lmm1,
    lmm_1re_formula, data_inla_poisson, n_samp = 500, additive_param = "Z1")

pois_lmm2_samples <- VariableImportanceINLA::sample_posterior_count(model_pois_lmm2,
    lmm_2re_formula, data_inla_poisson, n_samp = 500, additive_param = "Z1")

pois_lmm0_samples <- VariableImportanceINLA::sample_posterior_count(model_pois_lmm0,
    lmm_0fe_formula, data_inla_poisson, n_samp = 500, additive_param = "Z1")
```

```
plot_pois_lm <- VariableImportanceINLA::plot_samples(pois_lm_samples)

plot_pois_lmm1 <- VariableImportanceINLA::plot_samples(pois_lmm1_samples)

plot_pois_lmm2 <- VariableImportanceINLA::plot_samples(pois_lmm2_samples)

plot_pois_lmm0 <- VariableImportanceINLA::plot_samples(pois_lmm0_samples)
```

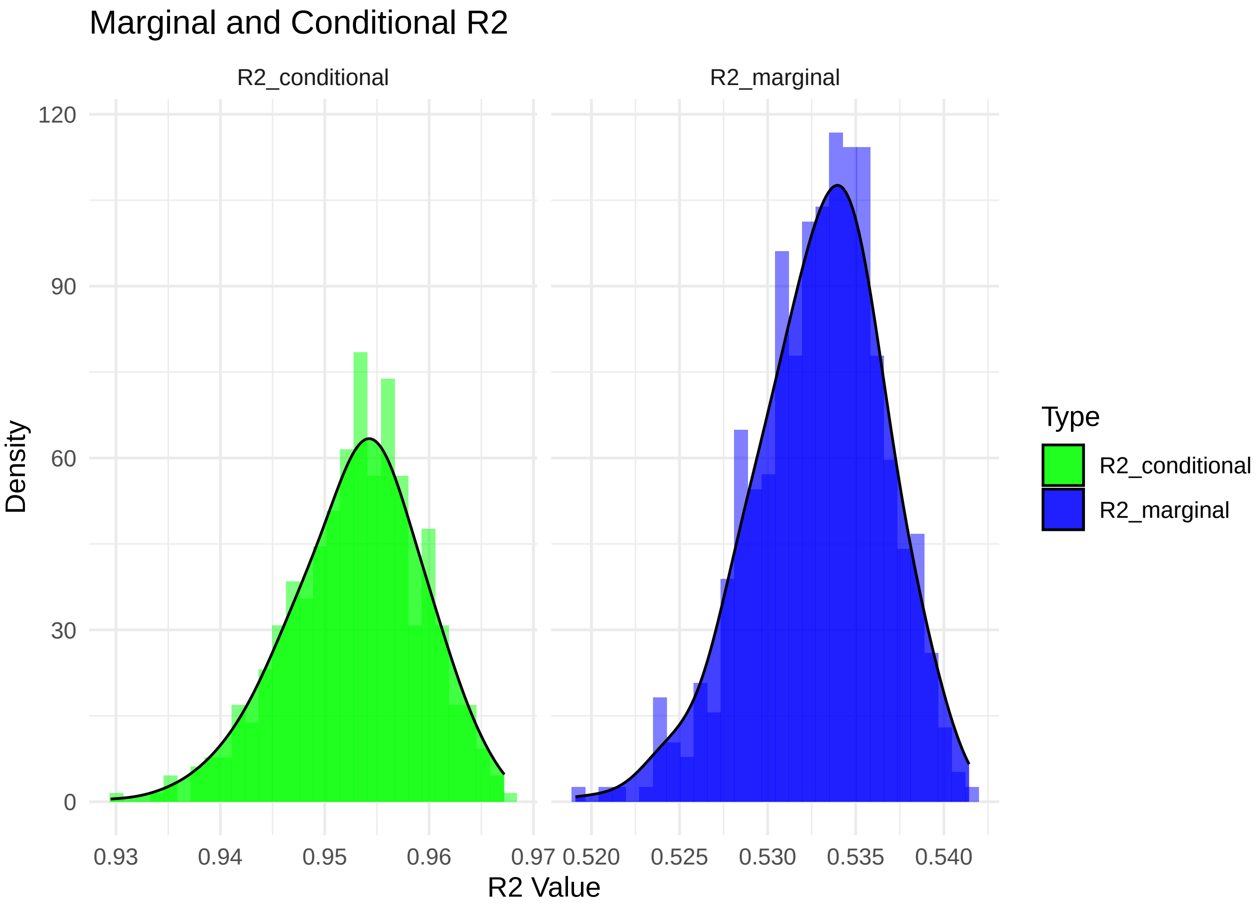
```
plot_pois_lmm2$fixed_effects
```



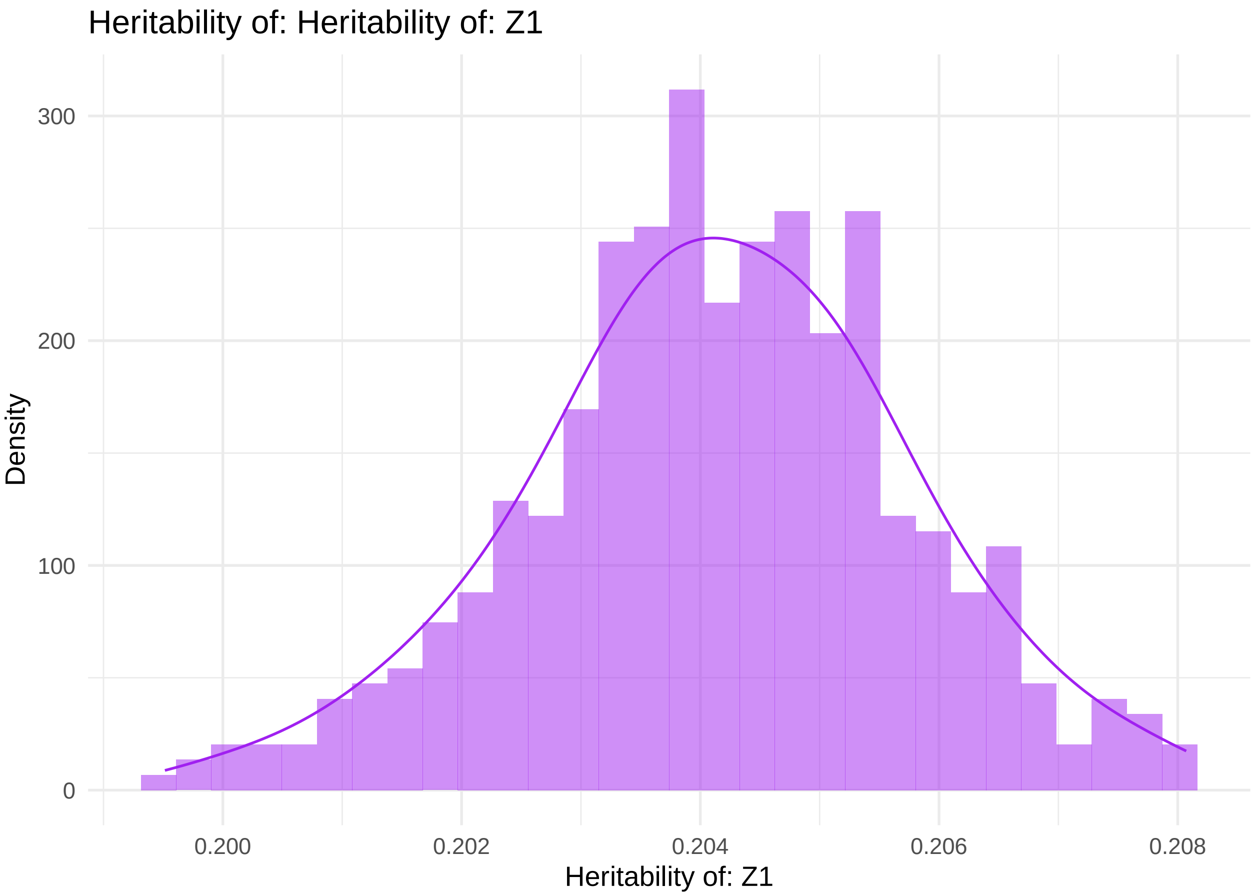
```
plot_pois_lmm2$random_effects
```



```
plot_pois_lmm2$R2
```



```
plot_pois_lmm2$heritability
```


Heritability of: Heritability of: Z1
Density
300
200
100
0
0.200
0.202
0.204
0.206
0.208
Heritability of: Z1