# Simulation study analysis

August Arnstad

11.11.2023

## Contents



```
## 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")
```

```
## -- R CMD build -----------------------------------------------------------------
##      checking for file '/private/var/folders/2g/tqlxxltj3g10t6vhctfp9j8h0000gn/T/Rtmp5KOpo5/remotes1
##   -  preparing 'BayesianImportance':
##    checking DESCRIPTION meta-information ...  v  checking DESCRIPTION meta-information
##   -  excluding invalid files
##      Subdirectory 'R' contains invalid file names:
##      'BayesianImportanceExample.Rmd' 'BayesianImportanceExample.pdf'
##   -  checking for LF line-endings in source and make files and shell scripts
##   -  checking for empty or unneeded directories
##    Omitted 'LazyData' from DESCRIPTION
##   -  building 'BayesianImportance_0.1.0.tar.gz'
##
##
```

```
# Install decompr2
devtools::install_git("https://gitlab.com/elonus/decompr2.git")
```

### Simulation

```
library(BayesianImportance)
library(INLA)
library(mnormt)
library(ggplot2)
library(lme4)
library(relaimpo)
```

```r
library(decompR2)
library(readr)
library(dplyr)
library(patchwork)
library(tidyr)
library(utils)
library(xtable)
```

## RETRIEVE DATA

```r
# Remember to change directory to where files are stored
lmm_results <- read_csv("lmm_results.csv")
lm_results <- read_csv("lm_results.csv")
inla_results <- read_csv("inla_results.csv")
decomp_lmg_results <- read_csv("decomp_lmg_results.csv")
decomp_rw_results <- read_csv("decomp_rw_results.csv")
```

## SUMMARIZE DATA FOR COMPARISON

```r
# Function to make data long format
make_long <- function(data) {
    pivot_longer(data, cols = starts_with("Variance"), names_to = "Variable",
        values_to = "Value")
}

# Apply the function to make data long format
lmm_results_long_data <- make_long(lmm_results)
lm_results_long_data <- make_long(lm_results)
inla_results_long_data <- make_long(inla_results)
decomp_lmg_results_long_data <- make_long(decomp_lmg_results)
decomp_rw_results_long_data <- make_long(decomp_rw_results)

# Define the covariance levels you have in your data
covariance_levels <- unique(lm_results$Covariance)

# Function to calculate summary statistics
calculate_summary <- function(data, variable_name, cov_level) {
    data %>%
        filter(Variable == variable_name, Covariance == cov_level) %>%
        summarise(Mean = mean(Value, na.rm = TRUE), Quantile_2.5 = quantile(Value,
            0.025, na.rm = TRUE), Quantile_97.5 = quantile(Value, 0.975,
            na.rm = TRUE))
}

# Function to create summary tables for a given effect
create_summary_tables <- function(effect) {
    summary_tables <- list()

    for (cov_level in covariance_levels) {
        lmm_summary <- calculate_summary(lmm_results_long_data, effect,
            cov_level) %>%
            mutate(Method = "LMM")
```

```
        lm_summary <- calculate_summary(lm_results_long_data, effect,
            cov_level) %>%
            mutate(Method = "LM")
        inla_summary <- calculate_summary(inla_results_long_data, effect,
            cov_level) %>%
            mutate(Method = "INLA")
        decomp_lmg_summary <- calculate_summary(decomp_lmg_results_long_data,
            effect, cov_level) %>%
            mutate(Method = "Decomp_LMG")
        decomp_rw_summary <- calculate_summary(decomp_rw_results_long_data,
            effect, cov_level) %>%
            mutate(Method = "Decomp_RW")

        summary_tables[[as.character(cov_level)]] <- bind_rows(lmm_summary,
            lm_summary, inla_summary, decomp_lmg_summary, decomp_rw_summary) %>%
            select(Method, everything())
    }

    return(summary_tables)
}

# List of effects
effects <- c("Variance_V1", "Variance_V2", "Variance_V3", "Variance_gamma")

# Initialize an empty list to store all summary tables
all_summary_tables <- list()

# Loop through each effect and create summary tables
for (effect in effects) {
    all_summary_tables[[effect]] <- create_summary_tables(effect)
}
```

## VIOLIN PLOTS

```
lmm_results_long <- pivot_longer(lmm_results, cols = starts_with("Variance"))
lm_results_long <- pivot_longer(lm_results, cols = starts_with("Variance"))
inla_results_long <- pivot_longer(inla_results, cols = starts_with("Variance"))
decomp_lmg_results_long <- pivot_longer(decomp_lmg_results, cols = starts_with("Variance"))
decomp_rw_results_long <- pivot_longer(decomp_rw_results, cols = starts_with("Variance"))
theoretical_values <- c(Variance_V1 = 1/8, Variance_V2 = 2/8, Variance_V3 = 3/8, Variance_gamma = 1/8,

# Combine all datasets into one with a 'Method' column
combined_results_long <- bind_rows(
  mutate(lm_results_long, Method = 'LM'),
  mutate(lmm_results_long, Method = 'LMM'),
  mutate(inla_results_long, Method = 'INLA'),
  mutate(decomp_lmg_results_long, Method = 'Decomp_LMG'),
  mutate(decomp_rw_results_long, Method = 'Decomp_RW')
)

# Create a list to store plots for each effect
effect_plots <- list()
```

```r
# Loop through each effect
for (effect in c("Variance_V1", "Variance_V2", "Variance_V3", "Variance_gamma", "Variance_epsilon")) {
  # Filter the combined dataset for the current effect
  data_for_plot <- combined_results_long %>%
    filter(name == effect) %>%
    mutate(Covariance = as.factor(Covariance))  # Ensure Covariance is a factor


  # Create the violin plot for the current effect
  p <- ggplot(data_for_plot, aes(x = Covariance, y = value, fill = Method)) +
    geom_violin(trim = FALSE) +
    geom_hline(yintercept = theoretical_values[[effect]], linetype = "dashed", color = "black") +
    labs(title = paste("Results for", effect),
         y = "Variance",
         x = "Covariance Level") +
    theme_minimal() +
    theme(legend.position = "bottom") +  # Adjust legend position as needed
    facet_wrap(~Method)  # Separate plots by Method, if not the same y-axis, add: , scales = "free_y"


  # Add the plot to the list
  effect_plots[[effect]] <- p
}
```

### Theoretical explained variance

```r
# Model parameters
beta <- c(1, sqrt(2), sqrt(3))
vi <- rep(1, length(beta))  # Assuming v_i = 1 for all i
sigma_epsilon_sq <- 1  # Residual variance (replace with your value)
sigma_gamma_sq <- 1    # Random effect variance (replace with your value)

# Covariance levels
rho_values <- c(0, 0.1, 0.5, 0.9)

# Calculate conditional and marginal R-squared for each covariance level
variance_explained <- data.frame(
  rho = rho_values,
  R2_conditional = numeric(length(rho_values)),
  R2_marginal = numeric(length(rho_values))
)

for (i in 1:length(rho_values)) {
  rho <- rho_values[i]

  # Calculate Var(Y) for the current rho
  var_Y <- sum((beta^2) * vi) + sum(beta[1] * beta[2] * sqrt(vi[1] * vi[2]) * rho) + sigma_gamma_sq + s
  cat(var_Y)

  # Calculate conditional R-squared
  R2_conditional <- (var_Y - sigma_epsilon_sq) / var_Y

  # Calculate marginal R-squared
```

```
  R2_marginal <- (var_Y - sigma_epsilon_sq - sigma_gamma_sq) / var_Y

  variance_explained[i, "R2_conditional"] <- R2_conditional
  variance_explained[i, "R2_marginal"] <- R2_marginal
}
```

```
## 88.1414218.7071079.272792
```

```
data_for_marginal_variance <- combined_results_long %>%
  filter(name == "Variance_total_marginal", Method!="LMM") %>%
  mutate(Covariance = as.factor(Covariance))  # Ensure Covariance is a factor

data_for_conditional_variance <- combined_results_long %>%
  filter(name == "Variance_total_conditional", Method!="LMM") %>%
  mutate(Covariance = as.factor(Covariance))  # Ensure Covariance is a factor


 # Create the violin plot for the current effect
marginal_variance_plot <- ggplot(data_for_marginal_variance, aes(x = Covariance, y = value, fill=Method)
  geom_violin(trim = FALSE) +
  labs(title = paste("Results for total marginal variance"),
        y = "Variance",
        x = "Covariance Level") +
  theme_minimal() +
  theme(legend.position = "bottom") +  # Adjust legend position as needed
  facet_wrap(~Method) +  # Separate plots by Method
  geom_hline(data = variance_explained, aes(yintercept = R2_marginal, color = ), linetype = "dashed")


conditional_variance_plot <- ggplot(data_for_conditional_variance, aes(x = Covariance, y = value, fill=M
  geom_violin(trim = FALSE) +
  labs(title = paste("Results for total conditional variance"),
        y = "Variance",
        x = "Covariance Level") +
  theme_minimal() +
  theme(legend.position = "bottom") +  # Adjust legend position as needed
  facet_wrap(~Method) +  # Separate plots by Method
  geom_hline(data = variance_explained, aes(yintercept = R2_conditional, color = ), linetype = "dashed")
```

## RESULTS

```
all_summary_tables
```

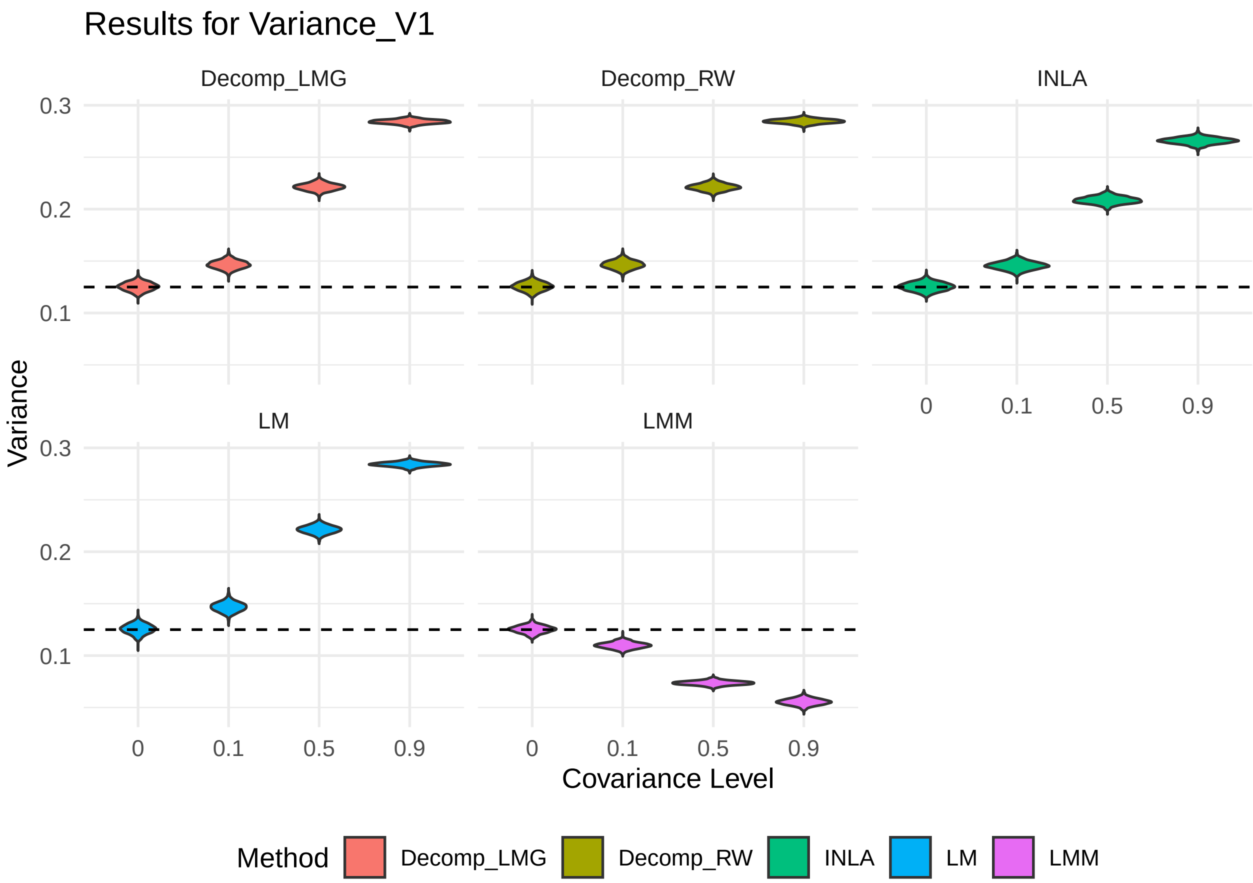
```
## $Variance_V1
## $Variance_V1$`0`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.125        0.118         0.132
## 2 LM         0.126        0.116         0.134
## 3 INLA       0.125        0.118         0.133
## 4 Decomp_LMG 0.125        0.118         0.133
## 5 Decomp_RW  0.125        0.117         0.133
##
```

```
## $Variance_V1$`0.1`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.110        0.104         0.116
## 2 LM         0.147        0.138         0.156
## 3 INLA       0.146        0.138         0.153
## 4 Decomp_LMG 0.147        0.139         0.154
## 5 Decomp_RW  0.147        0.139         0.154
##
## $Variance_V1$`0.5`
## # A tibble: 5 x 4
##   Method       Mean Quantile_2.5 Quantile_97.5
##   <chr>       <dbl>        <dbl>         <dbl>
## 1 LMM        0.0736       0.0696        0.0779
## 2 LM         0.221        0.214         0.229
## 3 INLA       0.209        0.202         0.216
## 4 Decomp_LMG 0.221        0.215         0.228
## 5 Decomp_RW  0.221        0.215         0.228
##
## $Variance_V1$`0.9`
## # A tibble: 5 x 4
##   Method       Mean Quantile_2.5 Quantile_97.5
##   <chr>       <dbl>        <dbl>         <dbl>
## 1 LMM        0.0554       0.0491        0.0614
## 2 LM         0.284        0.280         0.288
## 3 INLA       0.266        0.260         0.272
## 4 Decomp_LMG 0.284        0.280         0.288
## 5 Decomp_RW  0.285        0.280         0.289
##
##
## $Variance_V2
## $Variance_V2$`0`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.250        0.239         0.262
## 2 LM         0.250        0.237         0.262
## 3 INLA       0.250        0.239         0.262
## 4 Decomp_LMG 0.250        0.239         0.261
## 5 Decomp_RW  0.250        0.238         0.262
##
## $Variance_V2$`0.1`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.219        0.210         0.229
## 2 LM         0.261        0.250         0.273
## 3 INLA       0.262        0.251         0.272
## 4 Decomp_LMG 0.261        0.250         0.272
## 5 Decomp_RW  0.261        0.250         0.272
##
## $Variance_V2$`0.5`
## # A tibble: 5 x 4
```

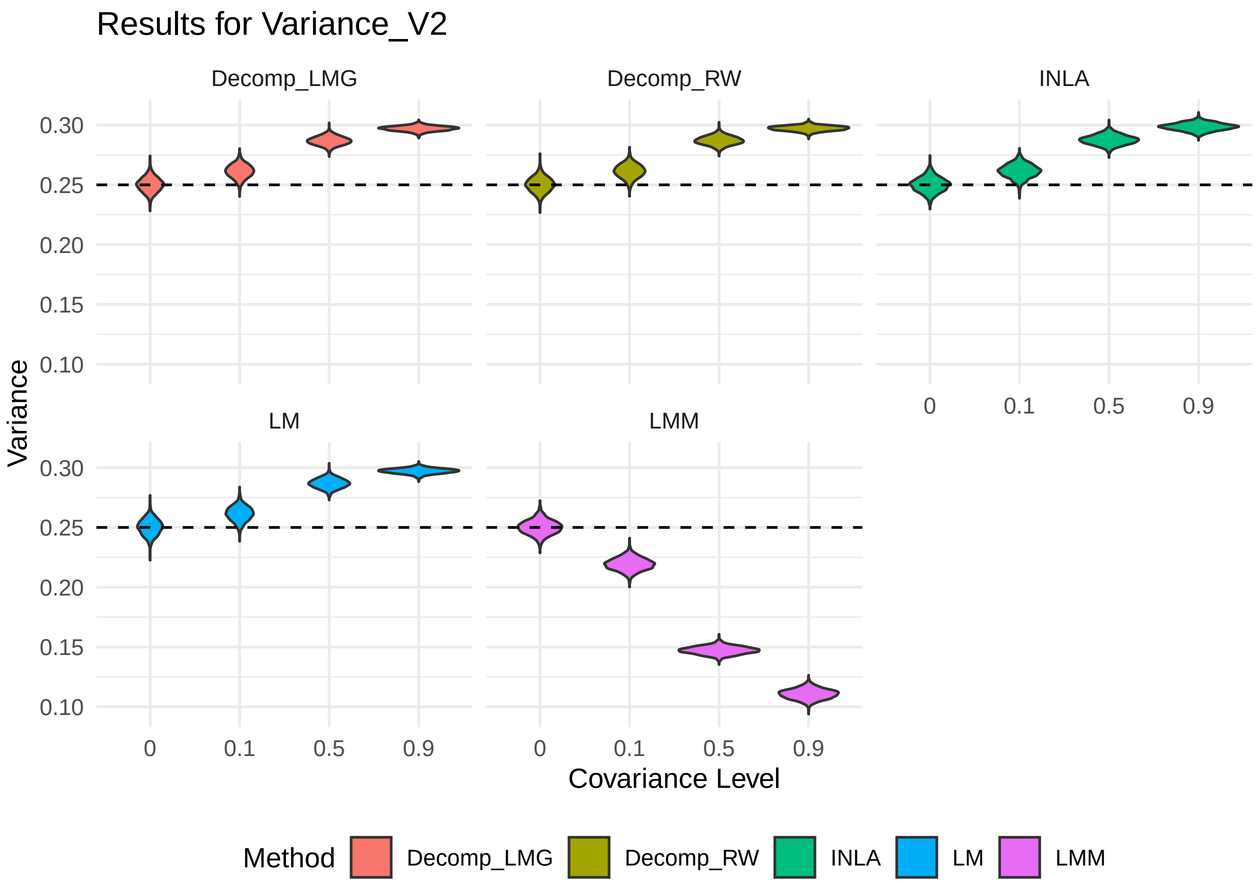
```
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.147        0.141         0.154
## 2 LM         0.287        0.279         0.294
## 3 INLA       0.288        0.280         0.296
## 4 Decomp_LMG 0.287        0.280         0.294
## 5 Decomp_RW  0.287        0.280         0.294
##
## $Variance_V2$`0.9`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.111        0.103         0.119
## 2 LM         0.297        0.293         0.301
## 3 INLA       0.299        0.292         0.305
## 4 Decomp_LMG 0.297        0.293         0.301
## 5 Decomp_RW  0.297        0.293         0.302
##
##
## $Variance_V3
## $Variance_V3$`0`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.375        0.359         0.390
## 2 LM         0.374        0.359         0.389
## 3 INLA       0.375        0.360         0.389
## 4 Decomp_LMG 0.374        0.360         0.389
## 5 Decomp_RW  0.374        0.360         0.389
##
## $Variance_V3$`0.1`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.329        0.317         0.341
## 2 LM         0.373        0.359         0.386
## 3 INLA       0.374        0.361         0.386
## 4 Decomp_LMG 0.373        0.360         0.385
## 5 Decomp_RW  0.373        0.360         0.386
##
## $Variance_V3$`0.5`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
## 1 LMM        0.221        0.213         0.229
## 2 LM         0.345        0.336         0.353
## 3 INLA       0.357        0.348         0.366
## 4 Decomp_LMG 0.345        0.337         0.353
## 5 Decomp_RW  0.345        0.337         0.353
##
## $Variance_V3$`0.9`
## # A tibble: 5 x 4
##   Method      Mean Quantile_2.5 Quantile_97.5
##   <chr>      <dbl>        <dbl>         <dbl>
```

```
## 1 LMM        0.166        0.157         0.176
## 2 LM         0.308        0.304         0.312
## 3 INLA       0.325        0.319         0.331
## 4 Decomp_LMG 0.308        0.304         0.312
## 5 Decomp_RW  0.307        0.304         0.311
##
##
## $Variance_gamma
## $Variance_gamma$`0`
## # A tibble: 5 x 4
##   Method        Mean Quantile_2.5 Quantile_97.5
##   <chr>        <dbl>        <dbl>         <dbl>
## 1 LMM          0.125        0.103         0.149
## 2 LM         NaN           NA            NA
## 3 INLA         0.125        0.103         0.149
## 4 Decomp_LMG   0.125        0.103         0.150
## 5 Decomp_RW    0.125        0.103         0.151
##
## $Variance_gamma$`0.1`
## # A tibble: 5 x 4
##   Method        Mean Quantile_2.5 Quantile_97.5
##   <chr>        <dbl>        <dbl>         <dbl>
## 1 LMM          0.110       0.0919         0.129
## 2 LM         NaN          NA             NA
## 3 INLA         0.110       0.0918         0.129
## 4 Decomp_LMG   0.110       0.0912         0.130
## 5 Decomp_RW    0.110       0.0910         0.130
##
## $Variance_gamma$`0.5`
## # A tibble: 5 x 4
##   Method         Mean Quantile_2.5 Quantile_97.5
##   <chr>         <dbl>        <dbl>         <dbl>
## 1 LMM          0.0736       0.0609        0.0873
## 2 LM         NaN           NA            NA
## 3 INLA         0.0735       0.0608        0.0872
## 4 Decomp_LMG   0.0734       0.0604        0.0876
## 5 Decomp_RW    0.0735       0.0599        0.0880
##
## $Variance_gamma$`0.9`
## # A tibble: 5 x 4
##   Method         Mean Quantile_2.5 Quantile_97.5
##   <chr>         <dbl>        <dbl>         <dbl>
## 1 LMM          0.0555       0.0455        0.0661
## 2 LM         NaN           NA            NA
## 3 INLA         0.0557       0.0457        0.0662
## 4 Decomp_LMG   0.0553       0.0452        0.0659
## 5 Decomp_RW    0.0553       0.0446        0.0661
```

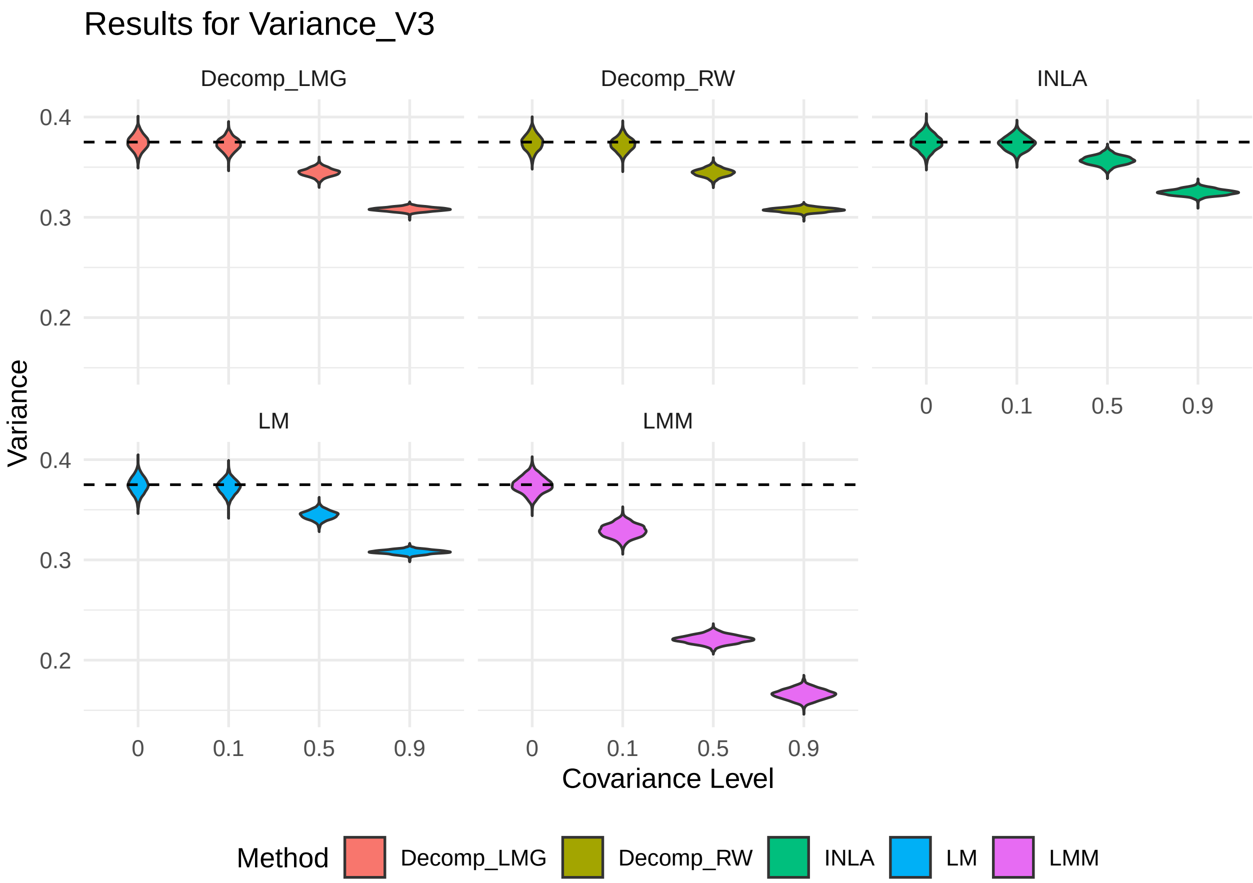
```
# The INLA value is the posterior mean of the fixed effect

effect_plots[["Variance_V1"]]
```



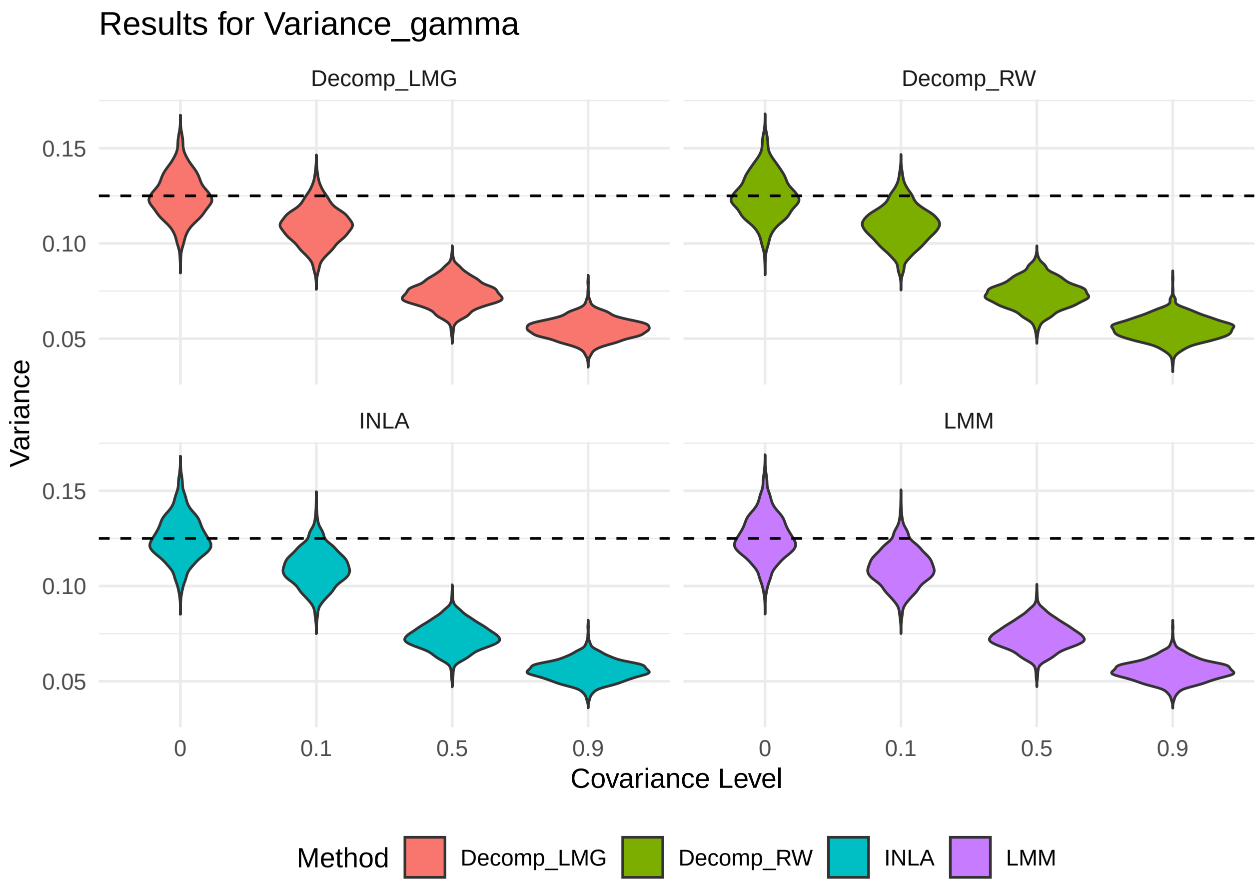

```
effect_plots[["Variance_V2"]]
```



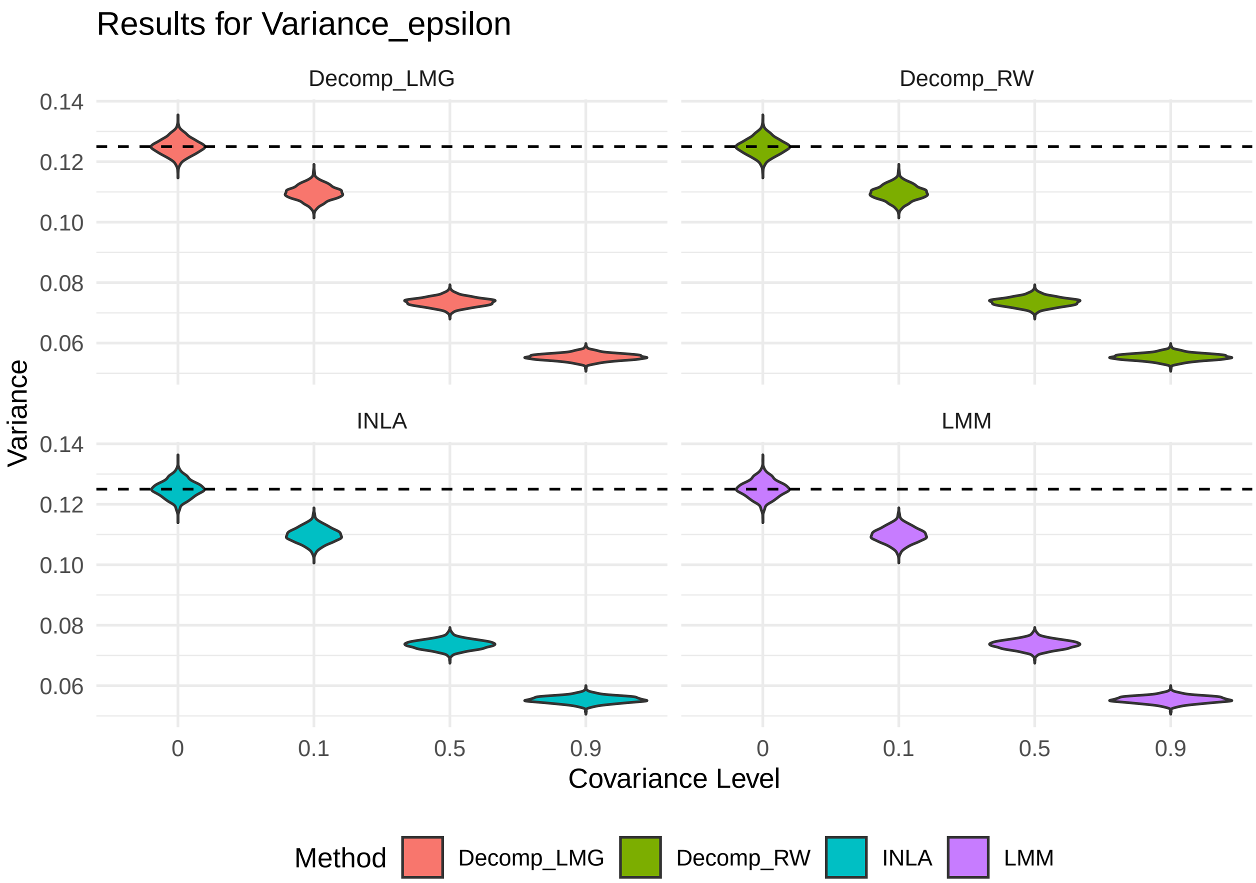

```
effect_plots[["Variance_V3"]]
```



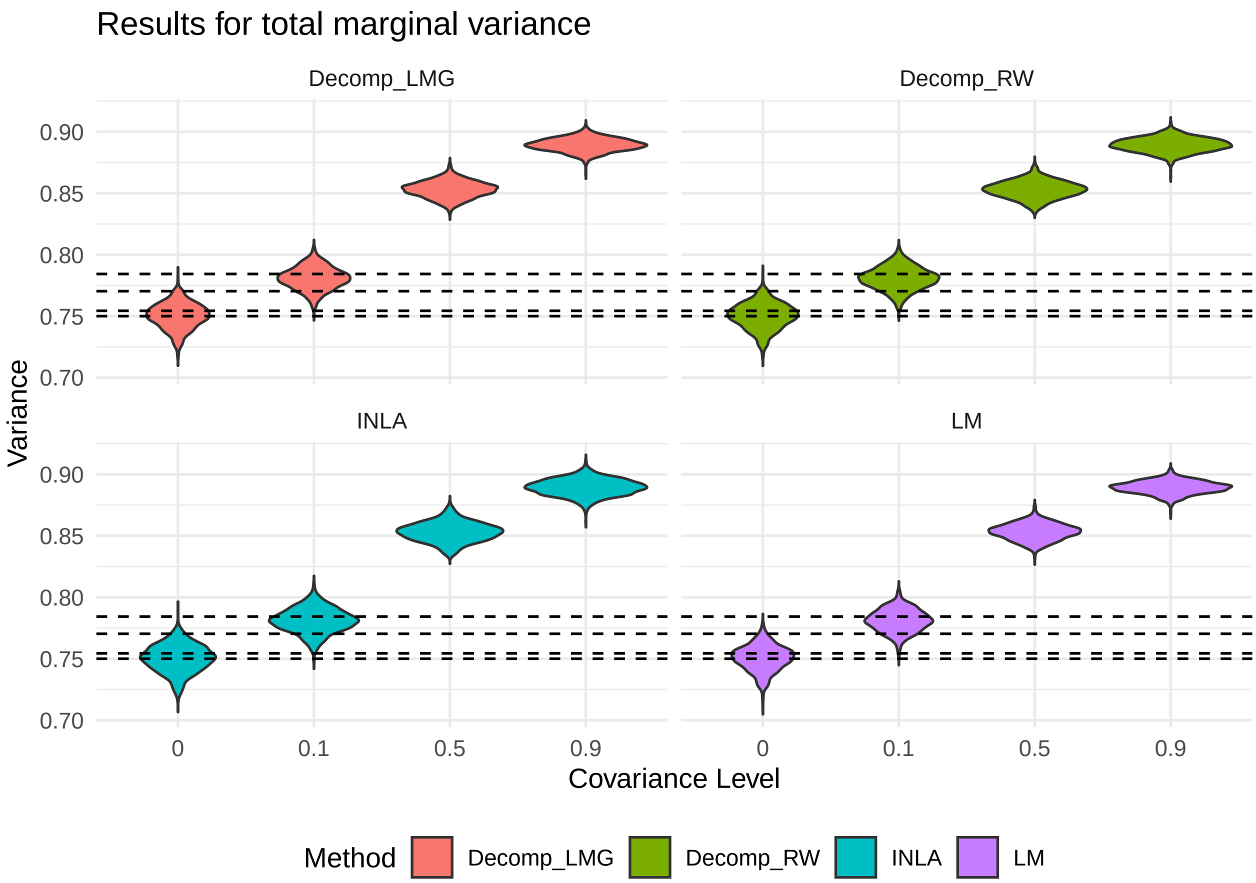

```
effect_plots[["Variance_gamma"]]
```



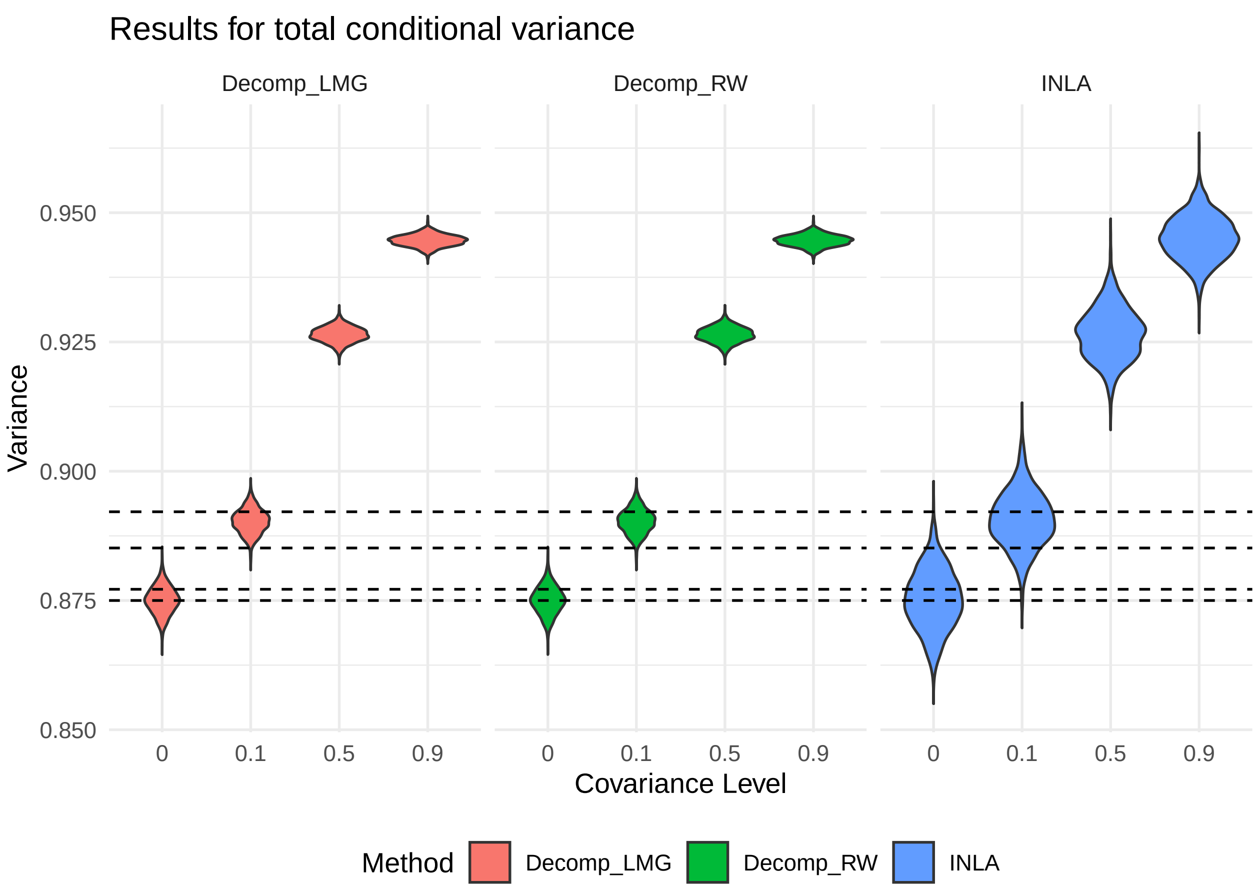

```
effect_plots[["Variance_epsilon"]]
```




```
marginal_variance_plot
```




```
conditional_variance_plot
```




## WRITE TO LATEX

The files are stored in the working directory. You can change the working directory to the "Tables" folder under "Latex" under "Prosjekt" to update the tables traight into Latex.

```
# Define the folder where you want to save the tables
output_folder_table <- "/Users/augustarnstad/Library/CloudStorage/OneDrive-NTNU/Semester 9/Prosjekt/Late

# Create the folder if it doesn't exist
if (!dir.exists(output_folder_table)) {
    dir.create(output_folder_table, recursive = TRUE)
}

# Save tables for Variance_V1
for (cov_level in names(all_summary_tables$Variance_V1)) {
    latex_table <- xtable(all_summary_tables$Variance_V1[[cov_level]],
        include.rownames = FALSE, digits = 6)
    file_name <- file.path(output_folder_table, paste("Variance_V1_summary_cov_",
        cov_level, ".tex", sep = ""))
    print(latex_table, file = file_name, include.rownames = FALSE)
}
# Save tables for Variance_V2
for (cov_level in names(all_summary_tables$Variance_V2)) {
    latex_table <- xtable(all_summary_tables$Variance_V2[[cov_level]],
        include.rownames = FALSE, digits = 6)
    file_name <- file.path(output_folder_table, paste("Variance_V2_summary_cov_",
```

```r
        cov_level, ".tex", sep = ""))
    print(latex_table, file = file_name, include.rownames = FALSE)
}
# Save tables for Variance_V3
for (cov_level in names(all_summary_tables$Variance_V3)) {
    latex_table <- xtable(all_summary_tables$Variance_V3[[cov_level]],
        include.rownames = FALSE, digits = 6)
    file_name <- file.path(output_folder_table, paste("Variance_V3_summary_cov_",
        cov_level, ".tex", sep = ""))
    print(latex_table, file = file_name, include.rownames = FALSE)
}
# Save tables for Variance_gamma
for (cov_level in names(all_summary_tables$Variance_gamma)) {
    latex_table <- xtable(all_summary_tables$Variance_gamma[[cov_level]],
        include.rownames = FALSE, digits = 6)
    file_name <- file.path(output_folder_table, paste("Variance_gamma_summary_cov_",
        cov_level, ".tex", sep = ""))
    print(latex_table, file = file_name, include.rownames = FALSE)
}
```

```r
# Define the folder where you want to save the plots
output_folder_plots <- "/Users/augustarnstad/Library/CloudStorage/OneDrive-NTNU/Semester 9/Prosjekt/Late

# Create the folder if it doesn't exist
if (!dir.exists(output_folder_plots)) {
    dir.create(output_folder_plots, recursive = TRUE)
}

# Save plots Assuming you have a list of ggplot objects named
# effect_plots
for (effect_name in names(effect_plots)) {
    file_name <- file.path(output_folder_plots, paste(effect_name, ".png",
        sep = ""))
    ggsave(file_name, plot = effect_plots[[effect_name]], width = 10,
        height = 6, dpi = 300)
}

file_name <- file.path(output_folder_plots, paste("Marginal_Variance.png",
    sep = ""))
ggsave(file_name, plot = marginal_variance_plot, width = 10, height = 6,
    dpi = 300)

file_name <- file.path(output_folder_plots, paste("Conditional_Variance.png",
    sep = ""))
ggsave(file_name, plot = conditional_variance_plot, width = 10, height = 6,
    dpi = 300)
```

### Summary

I am a bit unceratin if I sampled the lmm good enough. I think this needs some investigation, because it seems to take no effect of the higher covariance.

In general it seems that INLA resists the negative effect of higher covariance the same or even better than the decomposition from relaimpo and Matre. Judging by the table for V1 it seems that the mean value of the

INLA importance does not get dragged as much up as the others. Very exciting results. The results are in general very similar, but I notice that INLA often hugs a bit closer to the theoretical value then the others as covariance increases.

How should I format this and include it in my thesis in a suitable and appropriate way?