Parameter estimation of Poisson generalized linear mixed models based on three different statistical principles: a simulation study

Martí Casals, Klaus Langohr, Josep Lluís Carrasco
and Lars Rönnegård

December 2015

The material contained herein is supplementary to the article named in the title and published in SORT-Statistics and Operations Research Transactions Volume 39 (2).

Supplemental material: plots

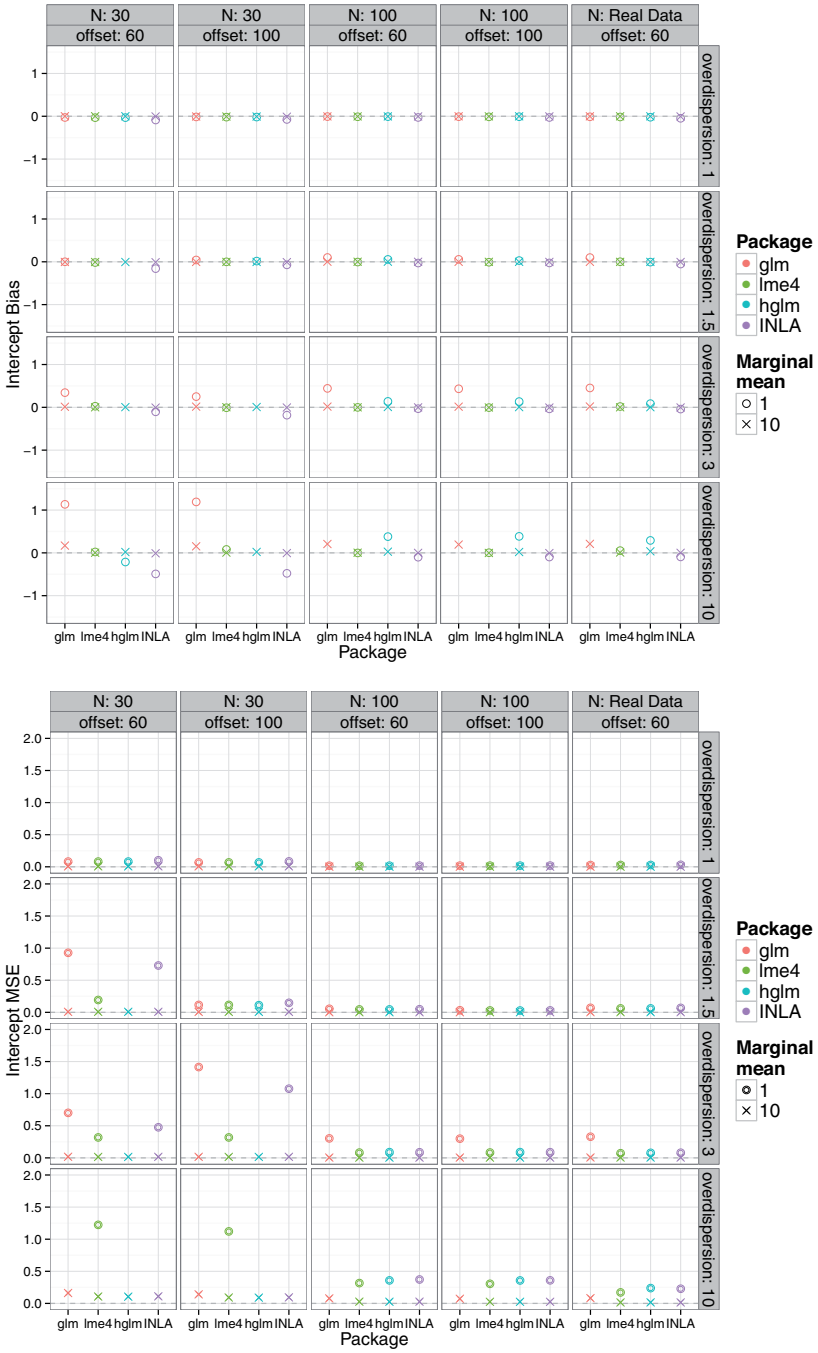

Figure 3: *Empirical bias (upper panel) and empirical MSE of the intercept estimate ($\hat{\beta}_0$) as a function of* Φ, μ, *offset, and N.*

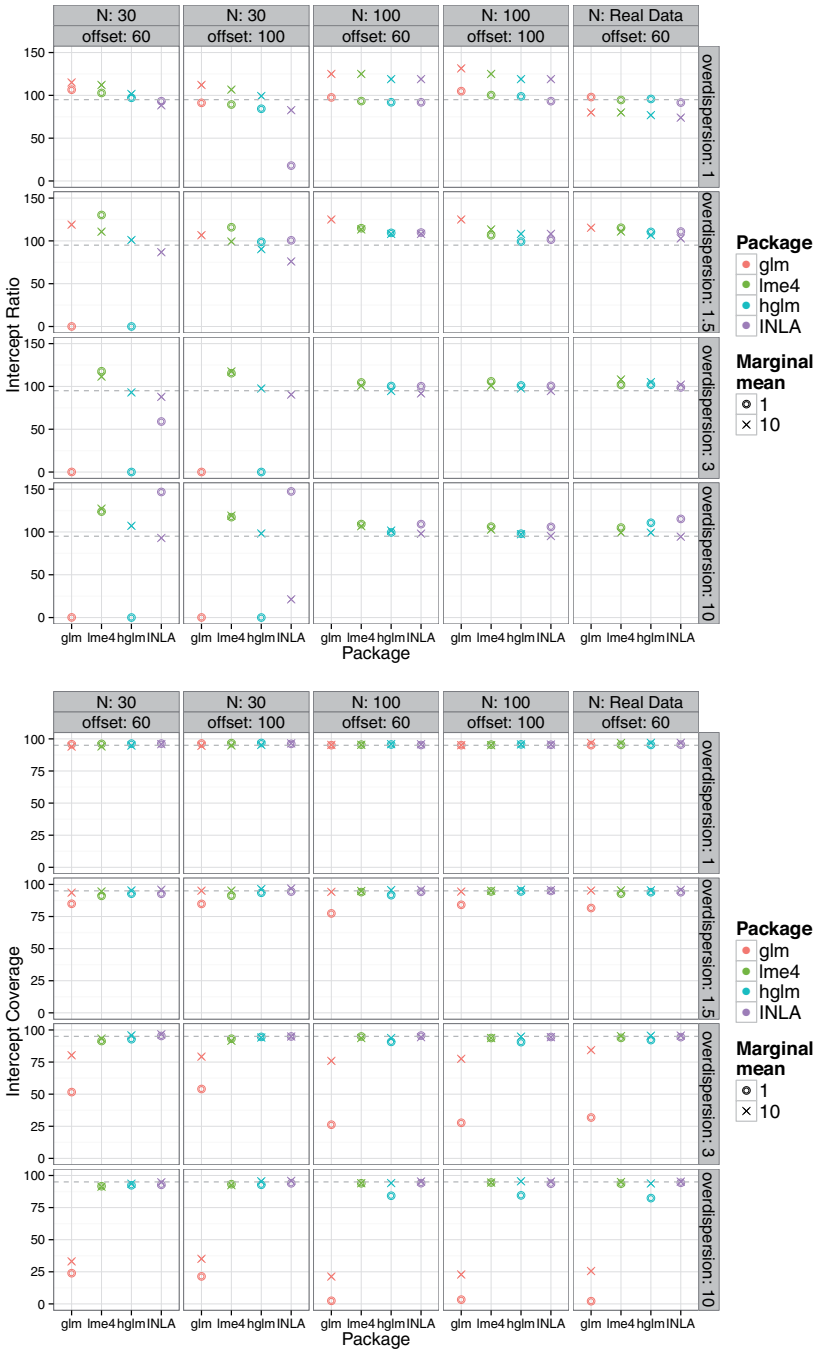

Figure 4: *Precision (upper panel) and empirical coverage of the intercept estimate ($\hat{\beta}_0$) as a function of Φ, μ, offset, and N. Precision is measured as the ratio of the estimator's empirical variance divided by the average of the squared standard errors.*

Supplemental material: tables

Table 4: *Empirical Bias of the slope ($\hat{\beta}_4$) and variance component ($\hat{\sigma}^2_u$) as a function of the marginal mean (μ), overdispersion (OD), and the cluster size (m). Sample size is $N = 100$, the offset is equal to 60.*

			lme4		**hglm**		**INLA**	
		OD	$m = 2$	$m = 5$	$m = 2$	$m = 5$	$m = 2$	$m = 5$
$\mu = 1$	$\hat{\beta}_4$	$\Phi = 1.5$	0	-0.012	0.01	0.001	-0.008	-0.017
		$\Phi = 10$	-0.01	-0.006	0.083	0.064	-0.042	-0.025
	$\hat{\sigma}^2_u$	$\Phi = 1.5$	-0.045	-0.029	-0.027	-0.023	0.009	0.009
		$\Phi = 10$	-0.492	-0.345	1.522	-1.13	0.221	0.217
$\mu = 10$	$\hat{\beta}_4$	$\Phi = 1.5$	-0.002	-0.001	-0.002	-0.001	-0.002	-0.001
		$\Phi = 10$	0.003	0.003	0.014	0.009	0.001	0.002
	$\hat{\sigma}^2_u$	$\Phi = 1.5$	0	-0.001	0.002	0.001	0.002	0
		$\Phi = 10$	-0.027	-0.025	-0.023	-0.013	0.002	0.015

Table 5: *Empirical MSE of the slope ($\hat{\beta}_4$) and variance component ($\hat{\sigma}^2_u$) as a function of the marginal mean (μ), overdispersion (OD), and the cluster size (m). Sample size is $N = 100$, the offset is equal to 60.*

			lme4		**hglm**		**INLA**	
		OD	$m = 2$	$m = 5$	$m = 2$	$m = 5$	$m = 2$	$m = 5$
$\mu = 1$	$\hat{\beta}_4$	$\Phi = 1.5$	0.041	0.025	0.04	0.024	0.043	0.026
		$\Phi = 10$	0.274	0.231	0.189	0.175	0.315	0.25
	$\hat{\sigma}^2_u$	$\Phi = 1.5$	0.017	0.006	0.013	0.005	0.02	0.007
		$\Phi = 10$	0.968	0.696	2.624	1.58	1.179	0.836
$\mu = 10$	$\hat{\beta}_4$	$\Phi = 1.5$	0.002	0.001	0.002	0.001	0.002	0.001
		$\Phi = 10$	0.023	0.022	0.022	0.022	0.023	0.022
	$\hat{\sigma}^2_u$	$\Phi = 1.5$	0	0	0	0	0	0
		$\Phi = 10$	0.007	0.005	0.006	0.005	0.008	0.006

Supplemental material: R code for simulations

Simulation study 1

```
## Note: Simulation study 1 uses the data set of Folk wrestling data (Section 6)
##       This data cannot be provided. Contrary to that, the code for
##       simulation study 2 does not require any external data.
################################################################################


################################################
## Simulation 1: On Real Data (Poisson response)
################################################

rm(list=ls(all=TRUE))
setwd("C:\\Users\\klangohr\\Documents\\A_UPC\\Marti\\PhD")
load("Aluche2.RData")


### Global settings and needed packages
#######################################
library(lme4)
library(hglm)
library(INLA)
library(R.utils)

### The real data set (Subset of "veran" containing data of regular fighters)
################################################################################

# Regular fighters only
dd.reg <- subset(veran, Regular=="Yes")
dd.reg <- with(dd.reg, dd.reg[order(Nwrestler, Season), ])
# Discard variables that are no longer needed
dd.reg <- dd.reg[c(2, 3, 9, 17, 28)]
rownames(dd.reg) <- 1:nrow(dd.reg)
names(dd.reg)[1:2] <- tolower(names(dd.reg)[1:2])
names(dd.reg)[4] <- 'ncombat'

formula1 <- with(dd.reg, LT~category+fallswinner)
X <- model.matrix(formula1)
formula2 <- ~0+factor(dd.reg$nwrestler)
Z <- model.matrix(formula2)
rm(formula1, formula2)

### Function to generate data based on data set dd.reg
### (adds new variable Y to existing data set dd.reg)
#####################################################
## Info:
## b0: intercept;
## b1--b4 correspond to 'category': b1, heavy; b2, light; b3, semi-heavy; b4, Winner

simfun <- function(X, Z, b0=-2.09, b1=0.40, b2=0.24, b3=0.10, b4=-0.46, sigmaZu){
  b <- c(b0, b1, b2, b3, b4)
  u=rnorm(length(unique(dd.reg$nwrestler)), 0, sigmaZu)
  eta=exp(X%*%b + Z%*%u + log(dd.reg$ncombat))
  dd.reg$Y <- rpois(length(eta), eta)
  dd.reg
}
```

```
### Simulation settings
#######################
# Fixed values
b1 <- 0.40
b2 <- 0.24
b3 <- 0.10
b4 <- -0.46

# Varying parameters
bet0 <- c(-4.1, -1.7, -4.5, -1.75, -4.35, -1.85, -4.8, -2.09)
sigmaZu <- c(0, 0, 0.5, 0.05, 1, 0.2, 2.1, 0.67)
# Save in data frame
SimSettings1to8 <- data.frame(Combination=1:8, Beta0=bet0, Sigma=sigmaZu)
comment(SimSettings1to8) <- 'Simulation settings for real data simulations'
rm(bet0, sigmaZu)

# Number of data sets to be generated
nrep <- 1000


### Preparation of lists with results from simulation settings 1 to 8
#########################################################################
SimDetails1to8 <- vector('list', 8)
SimComparison1to8 <- vector('list', 8)

### SimDetails1to8 contains 8 lists, each with all results of each method
#########################################################################
names(SimDetails1to8) <- paste("SimSetting", 1:8, sep='')
for (i in 1:8){
  SimDetails1to8[[i]] <- vector('list', 4)
  names(SimDetails1to8[[i]]) <- c('GLM', 'GLMM', 'HGLM', 'INLA')
}
rm(i)

### SimComparison1to8 contains 8 lists, each with the comparison criteria
#########################################################################
names(SimComparison1to8) <- paste("SimSetting", 1:8, sep='')
for (i in 1:8){
  SimComparison1to8[[i]] <- vector('list', 3)
  names(SimComparison1to8[[i]]) <- c('ConvergenceRates', 'EstimationResults',
                                     'ComputationTimes')
}
rm(i)


### La simulació
#################
for (s in 1:8)){
  set.seed(111114)
  b0    <- SimSettings1to8$Beta0[s]
  sigZu <- SimSettings1to8$Sigma[s]

  # Generate nrep data sets for each setting
  alldata <- vector('list', nrep)
  for (i in 1:nrep){
    alldata[[i]] <- simfun(X=X, Z=Z, b0=b0, sigmaZu=sigZu)
```

```
}
rm(i)

### Application of all packages to nrep data sets:
### * Temporary lists for each package are created.
### * 3 data frames will contain results for each setting
############################################################
# List for function glm
estglm <- vector('list', 8)
names(estglm) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4', 'time',
                   'coverBet0glm', 'coverBet4glm')

# List for package lm4
estglmer <- vector('list', 9)
names(estglmer) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4', 'var_id',
                     'time', 'coverBet0lmer', 'coverBet4lmer')

# List for package hglm
esthglm <- vector('list', 10)
names(esthglm) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4', 'var_id',
                    'se_var_id', 'time', 'coverBet0hglm', 'coverBet4hglm')

# List for package INLA
estinla <- vector('list', 10)
names(estinla) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4', 'var_id',
                    'se_var_id', 'time', 'coverBet0inla', 'coverBet4inla')

# Default values (FALSE and NA)
estglm$conv   <- rep(FALSE, nrep)
estglmer$conv <- rep(FALSE, nrep)
esthglm$conv  <- rep(FALSE, nrep)
estinla$conv  <- rep(FALSE, nrep)

for (l in 2:8){
  estglm[[l]]   <- rep(NA, nrep)
  estglmer[[l]] <- rep(NA, nrep)
  esthglm[[l]]  <- rep(NA, nrep)
  estinla[[l]]  <- rep(NA, nrep)
}
estglmer[[9]] <- rep(NA, nrep)
for (l in 9:10){
  esthglm[[l]] <- rep(NA, nrep)
  estinla[[l]] <- rep(NA, nrep)
}
rm(l)
for (i in 1:nrep){
  # Data set to be used
  dd <- alldata[[i]]

  # Fit of an GLM
  ###############
  estglm$time[i] <- system.time(mod0 <- try(glm(Y~offset(log(ncombat))+category
                                +fallswinner, family='poisson', data=dd)))[3]
  if (is.list(mod0)){
    estglm$conv[i] <- mod0$converged
    if (mod0$converged){
      estglm$beta0[i] <- coef(summary(mod0))[1, 1]
```

```
    estglm$beta4[i] <- coef(summary(mod0))[5, 1]
    estglm$se_beta0[i] <- summary(mod0)$coef[1, 2]
    estglm$se_beta4[i] <- summary(mod0)$coef[5, 2]
    n.glm <- mod0$df.residual
    t05 <- qt(0.975, n.glm)
    estglm$coverBet0glm[i] <-
      coef(summary(mod0))[1, 1]-t05*summary(mod0)$coef[1, 2] <= b0 &
      coef(summary(mod0))[1, 1]+t05*summary(mod0)$coef[1, 2] >= b0
    estglm$coverBet4glm[i] <-
      coef(summary(mod0))[5, 1]-t05*summary(mod0)$coef[5, 2] <= b4 &
      coef(summary(mod0))[5, 1]+t05*summary(mod0)$coef[5, 2] >= b4
    rm(n.glm, t05)
  } else{
      estglm$time[i]
  }
} else{
    estglm$time[i] <- NA
}
rm(mod0)

# Fit of an GLMM
################
estglmer$time[i] <- system.time(mod1 <- try(glmer(Y~category+fallswinner+
  (1|nwrestler)+offset(log(ncombat)), family="poisson", data=dd)))[3]
mod1 <- update(mod1, nAGQ=5)
if (class(mod1)=="glmerMod"){
  estglmer$conv[i] <- is.null(mod1@optinfo$conv$lme4$messages)
  if (is.null(mod1@optinfo$conv$lme4$messages)){
    row.fw <- which(rownames(summary(mod1)$coefficient) == 'fallswinnerYes')
    estglmer$beta0[i] <- mod1@beta[1]
    estglmer$beta4[i] <- mod1@beta[row.fw]
    estglmer$se_beta0[i] <-coef(summary(mod1))[1, 2]
    estglmer$se_beta4[i] <- coef(summary(mod1))[row.fw, 2]
    estglmer$var_id[i] <- attr(lme4::VarCorr(mod1)$nwrestler, "stddev")^2

    n <- length(fitted(mod1))
    k <- attr(logLik(mod1), "df")
    t05 <- qt(0.975, n-k)
    estglmer$coverBet0lmer[i] <-
      mod1@beta[1]-t05*coef(summary(mod1))[1, 2] <= b0 &
      mod1@beta[1]+t05*coef(summary(mod1))[1, 2] >= b0
    estglmer$coverBet4lmer[i] <-
      mod1@beta[5]-t05*coef(summary(mod1))[row.fw, 2] <= b4 &
      mod1@beta[5]+t05*coef(summary(mod1))[row.fw, 2] >= b4
    rm(n, k, t05, row.fw)
  } else{
      estglmer$time[i] <- NA
  }
} else{
    estglmer$time[i] <- NA
}
rm(mod1)

# Fit of an HGLM
################
esthglm$time[i] <- system.time(mod2 <- try(hglm(fixed=Y~category+fallswinner,
  random=~1|nwrestler, offset=(log(ncombat)), fix.disp=1, family=poisson(),
```

```
  method="HL11", data=dd, maxit=200, conv=1e-8)))[3]
if (is.list(mod2)){
  esthglm$conv[i] <- mod2$Converge=='converged'
  if (mod2$Converge=='converged'){
    esthglm$beta0[i] <- mod2$fixef[1]
    esthglm$beta4[i] <- mod2$fixef[5]
    esthglm$se_beta0[i] <- mod2$SeFe[1]
    esthglm$se_beta4[i] <- mod2$SeFe[5]
    esthglm$var_id[i] <- mod2$varRanef
    esthglm$se_var_id[i] <- 0.5*sqrt(mod2$varRanef)*mod2$SummVC2[[1]][2]

    n <- length(mod2$fv)
    k <- (n-(mod2$dfReFe))
    t05 <- qt(0.975, n-k)
    esthglm$coverBet0hglm[i] <-
      mod2$fixef[1]-t05*mod2$SeFe[1]<= b0 &
      mod2$fixef[1]+t05*mod2$SeFe[1]>= b0
    esthglm$coverBet4hglm[i] <-
      mod2$fixef[5]-t05*mod2$SeFe[5]<= b4 &
      mod2$fixef[5]+t05*mod2$SeFe[5]>= b4
    rm(n, k, t05)
  } el  se{
      esthglm$time[i] <- NA
  }
} else{
    esthglm$time[i] <- NA
}
rm(mod2)

# Fit of an INLA model
######################
formula <- Y~offset(log(ncombat))+ category+fallswinner+f(nwrestler, model="iid",
           hyper=list(theta=list(prior="logtgaussian", param=c(0, 0.0001))))
estinla$time[i] <- system.time(mod3 <- try(inla(formula, data=dd,
   family="poisson", control.compute=list(dic=T, cpo=TRUE),
   control.inla=list(strategy="laplace", int.strategy="grid"))))[3]
if (is.list(mod3)& !is.null(mod3$mode$mode.status)){
  estinla$conv[i] <- mod3$mode$mode.status==0
  if (mod3$mode$mode.status==0){
    estinla$beta0[i] <- mod3$summary.fixed[1, 1]
    estinla$beta4[i] <- mod3$summary.fixed[5, 1]
    estinla$se_beta0[i] <- mod3$summary.fixed[1, 2]
    estinla$se_beta4[i] <- mod3$summary.fixed[5, 2]

    prec.marg <- (mod3$marginals.hyperpar$'Precision for nwrestler')
    marg.variance <- inla.tmarginal(
            function(x) 1/x,
            prec.marg
    )
    m <- inla.emarginal(function(x) x, marg.variance)
    mm  <- inla.emarginal(function(x) x^2, marg.variance)

    estinla$var_id[i] <- m
    estinla$se_var_id[i] <- sqrt(mm-m^2)

    estinla$coverBet0inla[i] <- mod3$summary.fixed[1, 3] <= b0 &
                                mod3$summary.fixed[1, 5] >= b0
```

```
      estinla$coverBet4inla[i] <- mod3$summary.fixed[5, 3] <= b4 &
                                  mod3$summary.fixed[5, 5] >= b4
      rm(prec.marg, marg.variance, m, mm)
    } else{
        estinla$time[i] <- NA
    }
  } else{
      estinla$time[i] <- NA
  }
  rm(mod3, formula, dd)
}
rm(i)

### Save lists in list SimDetails1to8
#################################
SimDetails1to8[[s]][[1]] <- estglm
SimDetails1to8[[s]][[2]] <- estglmer
SimDetails1to8[[s]][[3]] <- esthglm
SimDetails1to8[[s]][[4]] <- estinla

### Computation of comparison criteria
######################################

### Convergence rates
conv.rate.glm   <- sum(estglm$conv)/nrep*100
conv.rate.glmer <- sum(estglmer$conv)/nrep*100
conv.rate.hglm  <- sum(esthglm$conv)/nrep*100
conv.rate.inla  <- sum(estinla$conv)/nrep*100


### Bias of beta0 and beta4
bias.glm.b0  <- mean(estglm$beta0, na.rm=T)-b0
bias.glm.b4  <- mean(estglm$beta4, na.rm=T)-b4
bias.lme4.b0 <- mean(estglmer$beta0, na.rm=T)-b0
bias.lme4.b4 <- mean(estglmer$beta4, na.rm=T)-b4
bias.hglm.b0 <- mean(esthglm$beta0, na.rm=T)-b0
bias.hglm.b4 <- mean(esthglm$beta4, na.rm=T)-b4
bias.inla.b0 <- mean(estinla$beta0, na.rm=T)-b0
bias.inla.b4 <- mean(estinla$beta4, na.rm=T)-b4

### Variances and standard errors
var1.glm.b0 <- mean(estglm$se_beta0^2, na.rm=T)
var1.glm.b4 <- mean(estglm$se_beta4^2, na.rm=T)
se.glm.b0   <- sqrt(var1.glm.b0)
se.glm.b4   <- sqrt(var1.glm.b4)
var2.glm.b0 <- var(estglm$beta0, na.rm=T)
var2.glm.b4 <- var(estglm$beta4, na.rm=T)

var1.lme4.b0 <- mean(estglmer$se_beta0^2, na.rm=T)
var1.lme4.b4 <- mean(estglmer$se_beta4^2, na.rm=T)
se.lme4.b0   <- sqrt(var1.lme4.b0)
se.lme4.b4   <- sqrt(var1.lme4.b4)
var2.lme4.b0 <- var(estglmer$beta0, na.rm=T)
var2.lme4.b4 <- var(estglmer$beta4, na.rm=T)

var1.hglm.b0 <- mean(esthglm$se_beta0^2, na.rm=T)
var1.hglm.b4 <- mean(esthglm$se_beta4^2, na.rm=T)
```

```
se.hglm.b0   <- sqrt(var1.hglm.b0)
se.hglm.b4   <- sqrt(var1.hglm.b4)
var2.hglm.b0 <- var(esthglm$beta0, na.rm=T)
var2.hglm.b4 <- var(esthglm$beta4, na.rm=T)

var1.inla.b0 <- mean(estinla$se_beta0^2, na.rm=T)
var1.inla.b4 <- mean(estinla$se_beta4^2, na.rm=T)
se.inla.b0   <- sqrt(var1.inla.b0)
se.inla.b4   <- sqrt(var1.inla.b4)
var2.inla.b0 <- var(estinla$beta0, na.rm=T)
var2.inla.b4 <- var(estinla$beta4, na.rm=T)

### Confidence intervals for standard errors
CI.se.glm.b0  <- quantile(estglm$se_beta0, c(0.025, 0.975))
CI.se.glm.b4  <- quantile(estglm$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.lme4.b0 <- quantile(estglmer$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.lme4.b4 <- quantile(estglmer$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.hglm.b0 <- quantile(esthglm$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.hglm.b4 <- quantile(esthglm$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.inla.b0 <- quantile(estinla$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.inla.b4 <- quantile(estinla$se_beta4, c(0.025, 0.975), na.rm=T)

### MSE for both parameters
MSE.glm.b0  <- bias.glm.b0^2 + var2.glm.b0
MSE.glm.b4  <- bias.glm.b4^2 + var2.glm.b4
MSE.lme4.b0 <- bias.lme4.b0^2+ var2.lme4.b0
MSE.lme4.b4 <- bias.lme4.b4^2+ var2.lme4.b4
MSE.hglm.b0 <- bias.hglm.b0^2+ var2.hglm.b0
MSE.hglm.b4 <- bias.hglm.b4^2+ var2.hglm.b4
MSE.inla.b0 <- bias.inla.b0^2+ var2.inla.b0
MSE.inla.b4 <- bias.inla.b4^2+ var2.inla.b4

### Coverage
coverBet0glm  <- sum(estglm$coverBet0glm, na.rm=T)/sum(estglm$conv)*100
coverBet4glm  <- sum(estglm$coverBet4glm, na.rm=T)/sum(estglm$conv)*100
coverBet0lmer <- sum(estglmer$coverBet0lmer, na.rm=T)/sum(estglmer$conv)*100
coverBet4lmer <- sum(estglmer$coverBet4lmer, na.rm=T)/sum(estglmer$conv)*100
coverBet0hglm <- sum(esthglm$coverBet0hglm, na.rm=T)/sum(esthglm$conv)*100
coverBet4hglm <- sum(esthglm$coverBet4hglm, na.rm=T)/sum(esthglm$conv)*100
coverBet0inla <- sum(estinla$coverBet0inla, na.rm=T)/sum(estinla$conv)*100
coverBet4inla <- sum(estinla$coverBet4inla, na.rm=T)/sum(estinla$conv)*100

### Variances of random effects
bias.lme4.var_id <- mean(estglmer$var_id, na.rm=T)-sigZu^2
se.lme4.var_id   <- sd(estglmer$var_id, na.rm=T)
MSE.lme4.var_id  <- bias.lme4.var_id^2 + var(estglmer$var_id, na.rm=T)
CI.lme4.var_id   <- quantile(estglmer$var_id, c(0.025, 0.975), na.rm=T)

bias.hglm.var_id <- mean(esthglm$var_id, na.rm=T)-sigZu^2
se.hglm.var_id   <- sd(esthglm$var_id, na.rm=T)
MSE.hglm.var_id  <- bias.hglm.var_id^2 + var(esthglm$var_id, na.rm=T)
CI.hglm.var_id   <- quantile(esthglm$se_var_id, c(0.025, 0.975), na.rm=T)
se.bias.hglm.var_id <- mean(esthglm$se_var_id, na.rm=T)

bias.inla.var_id <- mean(estinla$var_id, na.rm=T)-sigZu^2
MSE.inla.var_id  <- bias.inla.var_id^2 + var(estinla$var_id, na.rm=T)
CI.inla.var_id   <- quantile(estinla$se_var_id, c(0.025, 0.975), na.rm=T)
```

```
se.bias.inla.var_id <- mean(estinla$se_var_id, na.rm=T)


### Mean computation times
mean.time.glm   <- mean(estglm$time, na.rm=T)
mean.time.glmer <- mean(estglmer$time, na.rm=T)
mean.time.hglm  <- mean(esthglm$time, na.rm=T)
mean.time.inla  <- mean(estinla$time, na.rm=T)

### Save comparison criteria in list SimComparison1to8
#####################################################
package <- c('glm', 'glmer', 'hglm', 'INLA')

### Data frame with convergence rates
conv.rt <- c(conv.rate.glm, conv.rate.glmer, conv.rate.hglm, conv.rate.inla)
conv.df <- data.frame("Conv.Rates"=round(conv.rt, 1))
rownames(conv.df) <- package
SimComparison1to8[[s]][[1]] <- conv.df

# Cleaning up
rm(conv.rt, conv.df)
rm(conv.rate.glm, conv.rate.glmer, conv.rate.hglm, conv.rate.inla)

### Data frame with main results
method <- paste(rep(c('Intercept', 'Winner', 'SigmaWrest'), each=4),
                    c('glm', 'glmer', 'hglm', 'INLA'), sep='.')
bias   <- c(bias.glm.b0, bias.lme4.b0, bias.hglm.b0, bias.inla.b0,
            bias.glm.b4, bias.lme4.b4, bias.hglm.b4, bias.inla.b4,
            NA, bias.lme4.var_id, bias.hglm.var_id, bias.inla.var_id)
MSE    <- c(MSE.glm.b0, MSE.lme4.b0, MSE.hglm.b0, MSE.inla.b0,
            MSE.glm.b4, MSE.lme4.b4, MSE.hglm.b4, MSE.inla.b4,
            NA, MSE.lme4.var_id, MSE.hglm.var_id, MSE.inla.var_id)
Var    <- c(var2.glm.b0, var2.lme4.b0, var2.hglm.b0, var2.inla.b0,
            var2.glm.b4, var2.lme4.b4, var2.hglm.b4, var2.inla.b4,
            NA, var(estglmer$var_id, na.rm=T), var(esthglm$var_id, na.rm=T),
            var(estinla$var_id, na.rm=T))
sder   <- c(se.glm.b0, se.lme4.b0, se.hglm.b0, se.inla.b0,
            se.glm.b4, se.lme4.b4, se.hglm.b4, se.inla.b4,
            rep(NA, 2), se.bias.hglm.var_id, se.bias.inla.var_id)
ic95l  <- c(CI.se.glm.b0[1], CI.se.lme4.b0[1], CI.se.hglm.b0[1], CI.se.inla.b0[1],
            CI.se.glm.b4[1], CI.se.lme4.b4[1], CI.se.hglm.b4[1], CI.se.inla.b4[1],
            rep(NA, 2), CI.hglm.var_id[1], CI.inla.var_id[1])
ic95r  <- c(CI.se.glm.b0[2], CI.se.lme4.b0[2], CI.se.hglm.b0[2], CI.se.inla.b0[2],
            CI.se.glm.b4[2], CI.se.lme4.b4[2], CI.se.hglm.b4[2], CI.se.inla.b4[2],
            rep(NA, 2), CI.hglm.var_id[2], CI.inla.var_id[2])
cove <- c(coverBet0glm, coverBet0lmer, coverBet0hglm, coverBet0inla,
          coverBet4glm, coverBet4lmer, coverBet4hglm, coverBet4inla, rep(NA, 4))
results <- data.frame(Bias=round(bias, 3), MSE=round(MSE, 3), Var=round(Var, 3),
                      Std.Error=round(sder, 3), "SE.CI95%Lower"=round(ic95l, 3),
                      "SE.CI95%Upper"=round(ic95r, 3), Coverage=round(cove, 2))
rownames(results) <- method
SimComparison1to8[[s]][[2]] <- results

# Cleaning up
rm(results, method, bias, MSE, Var, sder, ic95l, ic95r, cove)
rm(bias.glm.b0, bias.lme4.b0, bias.hglm.b0, bias.inla.b0, bias.glm.b4,
   bias.lme4.b4, bias.hglm.b4, bias.inla.b4, bias.lme4.var_id,
```

```
    bias.hglm.var_id, bias.inla.var_id)
  rm(MSE.glm.b0, MSE.lme4.b0, MSE.hglm.b0, MSE.inla.b0, MSE.glm.b4, MSE.lme4.b4,
    MSE.hglm.b4, MSE.inla.b4, MSE.lme4.var_id, MSE.hglm.var_id, MSE.inla.var_id)
  rm(var1.glm.b0, var1.lme4.b0, var1.hglm.b0, var1.inla.b0, var1.glm.b4,
    var1.lme4.b4, var1.hglm.b4, var1.inla.b4)
  rm(var2.glm.b0, var2.lme4.b0, var2.hglm.b0, var2.inla.b0, var2.glm.b4,
    var2.lme4.b4, var2.hglm.b4, var2.inla.b4)
  rm(se.glm.b0, se.lme4.b0, se.hglm.b0, se.inla.b0, se.glm.b4, se.lme4.b4,
    se.hglm.b4, se.inla.b4, se.bias.hglm.var_id, se.bias.inla.var_id,
    se.hglm.var_id, se.lme4.var_id)
  rm(CI.se.glm.b0, CI.se.lme4.b0, CI.se.hglm.b0, CI.se.inla.b0, CI.se.glm.b4,
    CI.se.lme4.b4, CI.se.hglm.b4, CI.se.inla.b4, CI.inla.var_id, CI.hglm.var_id,
    CI.lme4.var_id)
  rm(coverBet0glm, coverBet0lmer, coverBet0hglm, coverBet0inla, coverBet4glm,
    coverBet4lmer, coverBet4hglm, coverBet4inla)


  ### Data frame with computation times
  comp.time  <- c(mean.time.glm, mean.time.glmer, mean.time.hglm, mean.time.inla)
  ctime.df <- data.frame(ElapsedTime=round(comp.time, 2))
  rownames(ctime.df) <- package
  SimComparison1to8[[s]][[3]] <- ctime.df

  # Cleaning up
  rm(comp.time, ctime.df)
  rm(mean.time.glm, mean.time.glmer, mean.time.hglm, mean.time.inla)
  rm(alldata, b0, sigZu, estglm, estglmer, esthglm, estinla, package)

  list.comparison <- SimComparison1to8[[s]]
  list.details    <- SimDetails1to8[[s]]
  save(list.comparison, list.details, file=paste0("SimulationResults", s, ".RData"))
  rm(list.comparison, list.details)
}

rm(s, X, Z, b1, b2, b3, b4, dd.reg, nrep, simfun)
comment(SimDetails1to8) <-
  'List for all 8 simulations with detailed results of all packages'
comment(SimComparison1to8) <-
  'List for all 8 simulations with comparison criteria'
save(SimSettings1to8, SimDetails1to8, SimComparison1to8,
  file='SimulationResults1to8.RData')
\end{Sinput}
\end{Schunk}
}

\subsection{Simulation study 2}
{\small
\begin{Schunk}
\begin{Sinput}
################################################
## Simulation 2: Simulated Data (Poisson response)
################################################

rm(list=ls(all=TRUE))
setwd("C:\\Users\\klangohr\\Documents\\A_UPC\\Marti\\PhD")

### Global settings and needed packages
```

```
#######################################
library(lme4)
library(hglm)
library(INLA)
library(R.utils)


### Function to generate simulation data
#######################################
## Info: b0: intercept;
## b1--b4 correspond to 'category': b1, heavy; b2, light; b3, semi-heavy; b4, Winner

simfun2 <- function(nw=100, ns=6, nco=100, sigmaZu=2.1, b0=-4.4, b1=0.4, b2=0.25,
                    b3=0.1, b4=-0.5)
{
  each <- sample(1:6, nw, replace=T)  # number of seasons of each wrestler
  nwrestler <- rep(1:nw, each)
  ntot <- length(nwrestler)
  category <- factor(rep(sample(c('Light', 'Medium', 'Semi heavy', 'Heavy'),
           nw, replace=T), each), levels=c('Light', 'Medium', 'Semi heavy', 'Heavy'))
  fallswinner <- factor(rep(sample(c('No', 'Yes'), nw, replace=T), each))
  ncombat <- rpois(ntot, nco)
  dd <- data.frame(nwrestler, ncombat, category, fallswinner)
  obs <- 1:length(nwrestler)
  b <- c(b0, b1, b2, b3, b4)
  eta0 <- model.matrix(~category+fallswinner, data=dd)%*%b
  alfa <- rep(rnorm(nw, sd=sigmaZu), each)
  eta <- with(dd, log(ncombat)+eta0+alfa)
  mu <- exp(eta)
  dd$Y <- with(dd, rpois(ntot, lambda=mu))
  dd
}


### Simulation settings
#######################
# Fixed values
# b1 <- 0.4; b2 <- 0.25; b3 <- 0.1;
b4 <- -0.5

# Varying parameters
nw <- rep(c(30, 100), each=16)
nco <- rep(rep(c(60, 100), each=8), 2)
bet0 <- c(-4.1, -1.7, -4.5, -1.75, -4.35, -1.85, -4.8, -2.09, -4.6, -2.21, -4.7,
          -2.27, -4.87, -2.34, -5.35, -2.55, -4.1, -1.7, -4.5, -1.75, -4.35,
          -1.85, -4.8, -2.09, -4.6, -2.21, -4.7, -2.27, -4.87, -2.34, -5.35, -2.55)
sigmaZu <- c(0, 0, 0.5, 0.05, 1, 0.2, 2.1, 0.67, 0, 0, 0.4, 0.05, 1, 0.2, 2.1,
             0.65, 0, 0, 0.5, 0.05, 1, 0.2, 2.1, 0.67, 0, 0, 0.4, 0.05, 1, 0.2,
             2.1, 0.65)
# Save in data frame
SimSettings940 <- data.frame(Combination=9:40, NumbWrestlers=nw, NumbCombats=nco,
                             Beta0=bet0, Sigma=sigmaZu)
comment(SimSettings940) <- 'Simulation settings for Scenario 2'
rm(bet0, sigmaZu, nco, nw)


# Number of data sets to be generated
```

```
nrep <- 1000


### Preparation of lists with results from simulation settings 1 to 8
#########################################################################
SimDetails940 <- vector('list', 32)
SimComparison940 <- vector('list', 32)

### SimDetails940 contains 8 lists, each with all results of each method
#########################################################################
names(SimDetails940) <- paste("SimSetting", 9:40, sep='')
for (i in 1:32){
  SimDetails940[[i]] <- vector('list', 4)
  names(SimDetails940[[i]]) <- c('GLM', 'GLMM', 'HGLM', 'INLA')
}
rm(i)

### SimComparison940 contains 8 lists, each with the comparison criteria
#########################################################################
names(SimComparison940) <- paste("SimSetting", 9:40, sep='')
for (i in 1:32){
  SimComparison940[[i]] <- vector('list', 3)
  names(SimComparison940[[i]]) <- c('ConvergenceRates', 'EstimationResults',
                                    'ComputationTimes')
}
rm(i)


### La simulació
################
for (s in 1:32){
  set.seed(111114)
  nw    <- SimSettings940$NumbWrestlers[s]
  nco   <- SimSettings940$NumbCombats[s]
  b0    <- SimSettings940$Beta0[s]
  sigZu <- SimSettings940$Sigma[s]

  # Generate nrep data sets for each setting
  alldata <- vector('list', nrep)
  for (i in 1:nrep){
    alldata[[i]] <- simfun2(nw=nw, ns=6, nco=nco, b0=b0, sigmaZu=sigZu)
  }
  rm(i)


  ### Application of all packages to nrep data sets:
  ### * Temporary lists for each package are created.
  ### * 3 data frames will contain results for each setting
  ###########################################################
  # List for function glm
  estglm <- vector('list', 8)
  names(estglm) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4', 'time',
                     'coverBet0glm', 'coverBet4glm')

  # List for package lm4
  estglmer <- vector('list', 9)
  names(estglmer) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4',
```

```
                    'var_id', 'time', 'coverBet0lmer', 'coverBet4lmer')

# List for package hglm
esthglm <- vector('list', 10)
names(esthglm) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4',
                    'var_id', 'se_var_id', 'time', 'coverBet0hglm', 'coverBet4hglm')

# List for package INLA
estinla <- vector('list', 10)
names(estinla) <- c('conv', 'beta0', 'beta4', 'se_beta0', 'se_beta4',
                    'var_id', 'se_var_id', 'time', 'coverBet0inla', 'coverBet4inla')

# Default values (FALSE and NA)
estglm$conv   <- rep(FALSE, nrep)
estglmer$conv <- rep(FALSE, nrep)
esthglm$conv  <- rep(FALSE, nrep)
estinla$conv  <- rep(FALSE, nrep)

for (l in 2:8){
  estglm[[l]]   <- rep(NA, nrep)
  estglmer[[l]] <- rep(NA, nrep)
  esthglm[[l]]  <- rep(NA, nrep)
  estinla[[l]]  <- rep(NA, nrep)
}
estglmer[[9]] <- rep(NA, nrep)
for (l in 9:10){
  esthglm[[l]] <- rep(NA, nrep)
  estinla[[l]] <- rep(NA, nrep)
}
rm(l)

for (i in 1:nrep){
  # Data set to be used
  dd <- alldata[[i]]

  # Fit of an GLM
  ################
  estglm$time[i] <- system.time(mod0 <- try(glm(Y~offset(log(ncombat))+category
    +fallswinner, family='poisson', data=dd)))[3]
  if (is.list(mod0)){
    estglm$conv[i] <- mod0$converged
    if (mod0$converged){
      estglm$beta0[i] <- coef(summary(mod0))[1, 1]
      estglm$beta4[i] <- coef(summary(mod0))["fallswinnerYes", 1]
      estglm$se_beta0[i] <- summary(mod0)$coef[1, 2]
      estglm$se_beta4[i] <- summary(mod0)$coef["fallswinnerYes", 2]
      n.glm <- mod0$df.residual
      t05 <- qt(0.975, n.glm)
      estglm$coverBet0glm[i] <-
        coef(summary(mod0))[1, 1]-t05*summary(mod0)$coef[1, 2] <= b0 &
        coef(summary(mod0))[1, 1]+t05*summary(mod0)$coef[1, 2] >= b0
      estglm$coverBet4glm[i] <-
        coef(summary(mod0))["fallswinnerYes", 1]-
          t05*summary(mod0)$coef["fallswinnerYes", 2] <= b4 &
        coef(summary(mod0))["fallswinnerYes", 1]+
          t05*summary(mod0)$coef["fallswinnerYes", 2] >= b4
      rm(n.glm, t05)
```

```
  } else{
      estglm$time[i]
  }
} else{
    estglm$time[i] <- NA
}
rm(mod0)

# Fit of an GLMM
################
estglmer$time[i] <- system.time(mod1 <- try(glmer(Y~category+fallswinner
  +(1|nwrestler)+offset(log(ncombat)), family="poisson", data=dd)))[3]
mod1 <- update(mod1, nAGQ=5)
if (class(mod1)=="glmerMod"){
  estglmer$conv[i] <- is.null(mod1@optinfo$conv$lme4$messages)
  if (is.null(mod1@optinfo$conv$lme4$messages)){
    row.fw <- which(rownames(summary(mod1)$coefficient) == 'fallswinnerYes')
    estglmer$beta0[i] <- mod1@beta[1]
    estglmer$beta4[i] <- mod1@beta[row.fw]
    estglmer$se_beta0[i] <-coef(summary(mod1))[1, 2]
    estglmer$se_beta4[i] <- coef(summary(mod1))[row.fw, 2]
    estglmer$var_id[i] <- attr(lme4::VarCorr(mod1)$nwrestler, "stddev")^2

    n <- length(fitted(mod1))
    k <- attr(logLik(mod1), "df")
    t05 <- qt(0.975, n-k)
    estglmer$coverBet0lmer[i] <-
      mod1@beta[1]-t05*coef(summary(mod1))[1, 2] <= b0 &
      mod1@beta[1]+t05*coef(summary(mod1))[1, 2] >= b0
    estglmer$coverBet4lmer[i] <-
      mod1@beta[5]-t05*coef(summary(mod1))[row.fw, 2] <= b4 &
      mod1@beta[5]+t05*coef(summary(mod1))[row.fw, 2] >= b4
    rm(n, k, t05, row.fw)
  } else{
      estglmer$time[i] <- NA
  }
} else{
    estglmer$time[i] <- NA
}
rm(mod1)

# Fit of an HGLM
################
esthglm$time[i] <- system.time(mod2 <- try(hglm(fixed=Y~category+fallswinner,
  random=~1|nwrestler, offset=(log(ncombat)), fix.disp=1, family=poisson(),
  data=dd, maxit=200, conv=1e-8)))[3]
if (is.list(mod2)){
  esthglm$conv[i] <- mod2$Converge=='converged'
  if (mod2$Converge=='converged'){
    esthglm$beta0[i] <- mod2$fixef[1]
    esthglm$beta4[i] <- mod2$fixef["fallswinnerYes"]
    esthglm$se_beta0[i] <- mod2$SeFe[1]
    ene <- length(mod2$SeFe)
    esthglm$se_beta4[i] <- mod2$SeFe[ene]
    esthglm$var_id[i] <- mod2$varRanef
    esthglm$se_var_id[i] <- 0.5*sqrt(mod2$varRanef)*mod2$SummVC2[[1]][2]
```

```
      n <- length(mod2$fv)
      k <- (n-(mod2$dfReFe))
      t05 <- qt(0.975, n-k)
      esthglm$coverBet0hglm[i] <-
        mod2$fixef[1]-t05*mod2$SeFe[1]<= b0 &
        mod2$fixef[1]+t05*mod2$SeFe[1]>= b0
      esthglm$coverBet4hglm[i] <-
        mod2$fixef["fallswinnerYes"]-t05*mod2$SeFe[ene]<= b4 &
        mod2$fixef["fallswinnerYes"]+t05*mod2$SeFe[ene]>= b4
      rm(n, k, t05, ene)
    } else{
        esthglm$time[i] <- NA
    }
  } else{
      esthglm$time[i] <- NA
  }
  rm(mod2)

  # Fit of an INLA model
  ######################
  formula <- Y~offset(log(ncombat))+ category+fallswinner+f(nwrestler, model="iid",
    hyper=list(theta=list(prior="logtgaussian", param=c(0, 0.0001))))
 estinla$time[i] <- system.time(mod3 <- try(inla(formula, data=dd, family="poisson",
    control.compute=list(dic=T, cpo=TRUE), control.inla=list(diff.logdens = 10,
    strategy="laplace", int.strategy="grid"))))[3]
  if (is.list(mod3)& !is.null(mod3$mode$mode.status)){
    estinla$conv[i] <- mod3$mode$mode.status==0
    if (mod3$mode$mode.status==0){
      estinla$beta0[i] <- mod3$summary.fixed[1, 1]
      estinla$beta4[i] <- mod3$summary.fixed["fallswinnerYes", 1]
      estinla$se_beta0[i] <- mod3$summary.fixed[1, 2]
      estinla$se_beta4[i] <- mod3$summary.fixed["fallswinnerYes", 2]

      prec.marg <- (mod3$marginals.hyperpar$'Precision for nwrestler')
      marg.variance <- inla.tmarginal(
        function(x) 1/x,                        prec.marg
      )
      m <- inla.emarginal(function(x) x, marg.variance)
      mm  <- inla.emarginal(function(x) x^2, marg.variance)

      estinla$var_id[i] <- m
      estinla$se_var_id[i] <- sqrt(mm-m^2)

      estinla$coverBet0inla[i] <- mod3$summary.fixed[1, 3] <= b0 &
                                  mod3$summary.fixed[1, 5] >= b0
      estinla$coverBet4inla[i] <- mod3$summary.fixed["fallswinnerYes", 3] <= b4 &
                                  mod3$summary.fixed["fallswinnerYes", 5] >= b4
      rm(prec.marg, marg.variance, m, mm)
    } else{
        estinla$time[i] <- NA
    }
  } else{
      estinla$time[i] <- NA
  }
  rm(mod3, formula, dd)
}
rm(i)
```

```
### Save lists in list SimDetails940
#################################
SimDetails940[[s]][[1]] <- estglm
SimDetails940[[s]][[2]] <- estglmer
SimDetails940[[s]][[3]] <- esthglm
SimDetails940[[s]][[4]] <- estinla

### Computation of comparison criteria
######################################

### Convergence rates
conv.rate.glm   <- sum(estglm$conv)/nrep*100
conv.rate.glmer <- sum(estglmer$conv)/nrep*100
conv.rate.hglm  <- sum(esthglm$conv)/nrep*100
conv.rate.inla  <- sum(estinla$conv)/nrep*100


### Bias of beta0 and beta 4
bias.glm.b0  <- mean(estglm$beta0, na.rm=T)-b0
bias.glm.b4  <- mean(estglm$beta4, na.rm=T)-b4
bias.lme4.b0 <- mean(estglmer$beta0, na.rm=T)-b0
bias.lme4.b4 <- mean(estglmer$beta4, na.rm=T)-b4
bias.hglm.b0 <- mean(esthglm$beta0, na.rm=T)-b0
bias.hglm.b4 <- mean(esthglm$beta4, na.rm=T)-b4
bias.inla.b0 <- mean(estinla$beta0, na.rm=T)-b0
bias.inla.b4 <- mean(estinla$beta4, na.rm=T)-b4

### Variances and standard errors
var1.glm.b0 <- mean(estglm$se_beta0^2, na.rm=T)
var1.glm.b4 <- mean(estglm$se_beta4^2, na.rm=T)
se.glm.b0   <- sqrt(var1.glm.b0)
se.glm.b4   <- sqrt(var1.glm.b4)
var2.glm.b0 <- var(estglm$beta0, na.rm=T)
var2.glm.b4 <- var(estglm$beta4, na.rm=T)

var1.lme4.b0 <- mean(estglmer$se_beta0^2, na.rm=T)
var1.lme4.b4 <- mean(estglmer$se_beta4^2, na.rm=T)
se.lme4.b0   <- sqrt(var1.lme4.b0)
se.lme4.b4   <- sqrt(var1.lme4.b4)
var2.lme4.b0 <- var(estglmer$beta0, na.rm=T)
var2.lme4.b4 <- var(estglmer$beta4, na.rm=T)

var1.hglm.b0 <- mean(esthglm$se_beta0^2, na.rm=T)
var1.hglm.b4 <- mean(esthglm$se_beta4^2, na.rm=T)
se.hglm.b0   <- sqrt(var1.hglm.b0)
se.hglm.b4   <- sqrt(var1.hglm.b4)
var2.hglm.b0 <- var(esthglm$beta0, na.rm=T)
var2.hglm.b4 <- var(esthglm$beta4, na.rm=T)

var1.inla.b0 <- mean(estinla$se_beta0^2, na.rm=T)
var1.inla.b4 <- mean(estinla$se_beta4^2, na.rm=T)
se.inla.b0 <- sqrt(var1.inla.b0)
se.inla.b4 <- sqrt(var1.inla.b4)
var2.inla.b0 <- var(estinla$beta0, na.rm=T)
var2.inla.b4 <- var(estinla$beta4, na.rm=T)

### Confidence intervals for standard errors
```

```
CI.se.glm.b0  <- quantile(estglm$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.glm.b4  <- quantile(estglm$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.lme4.b0 <- quantile(estglmer$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.lme4.b4 <- quantile(estglmer$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.hglm.b0 <- quantile(esthglm$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.hglm.b4 <- quantile(esthglm$se_beta4, c(0.025, 0.975), na.rm=T)
CI.se.inla.b0 <- quantile(estinla$se_beta0, c(0.025, 0.975), na.rm=T)
CI.se.inla.b4 <- quantile(estinla$se_beta4, c(0.025, 0.975), na.rm=T)

### MSE for both parameters
MSE.glm.b0  <- bias.glm.b0^2 + var2.glm.b0
MSE.glm.b4  <- bias.glm.b4^2 + var2.glm.b4
MSE.lme4.b0 <- bias.lme4.b0^2+ var2.lme4.b0
MSE.lme4.b4 <- bias.lme4.b4^2+ var2.lme4.b4
MSE.hglm.b0 <- bias.hglm.b0^2+ var2.hglm.b0
MSE.hglm.b4 <- bias.hglm.b4^2+ var2.hglm.b4
MSE.inla.b0 <- bias.inla.b0^2+ var2.inla.b0
MSE.inla.b4 <- bias.inla.b4^2+ var2.inla.b4

### Coverage
coverBet0glm  <- sum(estglm$coverBet0glm, na.rm=T)/sum(estglm$conv)*100
coverBet4glm  <- sum(estglm$coverBet4glm, na.rm=T)/sum(estglm$conv)*100
coverBet0lmer <- sum(estglmer$coverBet0lmer, na.rm=T)/sum(estglmer$conv)*100
coverBet4lmer <- sum(estglmer$coverBet4lmer, na.rm=T)/sum(estglmer$conv)*100
coverBet0hglm <- sum(esthglm$coverBet0hglm, na.rm=T)/sum(esthglm$conv)*100
coverBet4hglm <- sum(esthglm$coverBet4hglm, na.rm=T)/sum(esthglm$conv)*100
coverBet0inla <- sum(estinla$coverBet0inla, na.rm=T)/sum(estinla$conv)*100
coverBet4inla <- sum(estinla$coverBet4inla, na.rm=T)/sum(estinla$conv)*100

### Variances of random effects
bias.lme4.var_id <- mean(estglmer$var_id, na.rm=T)-sigZu^2
se.lme4.var_id   <- sd(estglmer$var_id, na.rm=T)
MSE.lme4.var_id  <- bias.lme4.var_id^2+ var(estglmer$var_id, na.rm=T)
CI.lme4.var_id   <- quantile(estglmer$var_id, c(0.025, 0.975), na.rm=T)

bias.hglm.var_id <- mean(esthglm$var_id, na.rm=T)-sigZu^2
se.hglm.var_id   <- sd(esthglm$var_id, na.rm=T)
MSE.hglm.var_id  <- bias.hglm.var_id^2+var(esthglm$var_id, na.rm=T)
CI.hglm.var_id   <- quantile(esthglm$se_var_id, c(0.025, 0.975), na.rm=T)
se.bias.hglm.var_id <- mean(esthglm$se_var_id, na.rm=T)

bias.inla.var_id <- mean(estinla$var_id, na.rm=T)-sigZu^2
MSE.inla.var_id  <- bias.inla.var_id^2+var(estinla$var_id, na.rm=T)
CI.inla.var_id   <- quantile(estinla$se_var_id, c(0.025, 0.975), na.rm=T)
se.bias.inla.var_id <- mean(estinla$se_var_id, na.rm=T)


### Mean computation times
mean.time.glm   <- mean(estglm$time, na.rm=T)
mean.time.glmer <- mean(estglmer$time, na.rm=T)
mean.time.hglm  <- mean(esthglm$time, na.rm=T)
mean.time.inla  <- mean(estinla$time, na.rm=T)

### Save comparison criteria in list SimComparison940
#################################################
package <- c('glm', 'glmer', 'hglm', 'INLA')
```

```
### Data frame with convergence rates
conv.rt <- c(conv.rate.glm, conv.rate.glmer, conv.rate.hglm, conv.rate.inla)
conv.df <- data.frame("Conv.Rates"=round(conv.rt, 1))
rownames(conv.df) <- package
SimComparison940[[s]][[1]] <- conv.df

# Cleaning up
rm(conv.rt, conv.df)
rm(conv.rate.glm, conv.rate.glmer, conv.rate.hglm, conv.rate.inla)

### Data frame with main results
method <- paste(rep(c('Intercept', 'Winner', 'SigmaWrest'), each=4),
                    c('glm', 'glmer', 'hglm', 'INLA'), sep='.')
bias   <- c(bias.glm.b0, bias.lme4.b0, bias.hglm.b0, bias.inla.b0,
            bias.glm.b4, bias.lme4.b4, bias.hglm.b4, bias.inla.b4,
            NA, bias.lme4.var_id, bias.hglm.var_id, bias.inla.var_id)
MSE    <- c(MSE.glm.b0, MSE.lme4.b0, MSE.hglm.b0, MSE.inla.b0,
            MSE.glm.b4, MSE.lme4.b4, MSE.hglm.b4, MSE.inla.b4,
            NA, MSE.lme4.var_id, MSE.hglm.var_id, MSE.inla.var_id)
Var    <- c(var2.glm.b0, var2.lme4.b0, var2.hglm.b0, var2.inla.b0,
            var2.glm.b4, var2.lme4.b4, var2.hglm.b4, var2.inla.b4,
            NA, var(estglmer$var_id, na.rm=T), var(esthglm$var_id, na.rm=T),
            var(estinla$var_id, na.rm=T))
sder   <- c(se.glm.b0, se.lme4.b0, se.hglm.b0, se.inla.b0,
            se.glm.b4, se.lme4.b4, se.hglm.b4, se.inla.b4,
            rep(NA, 2), se.bias.hglm.var_id, se.bias.inla.var_id)
ic95l  <- c(CI.se.glm.b0[1], CI.se.lme4.b0[1], CI.se.hglm.b0[1], CI.se.inla.b0[1],
            CI.se.glm.b4[1], CI.se.lme4.b4[1], CI.se.hglm.b4[1], CI.se.inla.b4[1],
            rep(NA, 2), CI.hglm.var_id[1], CI.inla.var_id[1])
ic95r  <- c(CI.se.glm.b0[2], CI.se.lme4.b0[2], CI.se.hglm.b0[2], CI.se.inla.b0[2],
            CI.se.glm.b4[2], CI.se.lme4.b4[2], CI.se.hglm.b4[2], CI.se.inla.b4[2],
            rep(NA, 2), CI.hglm.var_id[2], CI.inla.var_id[2])
cove <- c(coverBet0glm, coverBet0lmer, coverBet0hglm, coverBet0inla,
          coverBet4glm, coverBet4lmer, coverBet4hglm, coverBet4inla, rep(NA, 4))
results <- data.frame(Bias=round(bias, 3), MSE=round(MSE, 3), Var=round(Var, 3),
                      Std.Error=round(sder, 3), "SE.CI95%Lower"=round(ic95l, 3),
                      "SE.CI95%Upper"=round(ic95r, 3), Coverage=round(cove, 2))
rownames(results) <- method
SimComparison1to8[[s]][[2]] <- results

# Cleaning up
rm(results, method, bias, MSE, Var, sder, ic95l, ic95r, cove)
rm(bias.glm.b0, bias.lme4.b0, bias.hglm.b0, bias.inla.b0, bias.glm.b4,
   bias.lme4.b4, bias.hglm.b4, bias.inla.b4, bias.lme4.var_id,
   bias.hglm.var_id, bias.inla.var_id)
rm(MSE.glm.b0, MSE.lme4.b0, MSE.hglm.b0, MSE.inla.b0, MSE.glm.b4, MSE.lme4.b4,
   MSE.hglm.b4, MSE.inla.b4, MSE.lme4.var_id, MSE.hglm.var_id, MSE.inla.var_id)
rm(var1.glm.b0, var1.lme4.b0, var1.hglm.b0, var1.inla.b0, var1.glm.b4,
   var1.lme4.b4, var1.hglm.b4, var1.inla.b4)
rm(var2.glm.b0, var2.lme4.b0, var2.hglm.b0, var2.inla.b0, var2.glm.b4,
   var2.lme4.b4, var2.hglm.b4, var2.inla.b4)
rm(se.glm.b0, se.lme4.b0, se.hglm.b0, se.inla.b0, se.glm.b4, se.lme4.b4,
   se.hglm.b4, se.inla.b4, se.bias.hglm.var_id, se.bias.inla.var_id,
   se.hglm.var_id, se.lme4.var_id)
rm(CI.se.glm.b0, CI.se.lme4.b0, CI.se.hglm.b0, CI.se.inla.b0, CI.se.glm.b4,
   CI.se.lme4.b4, CI.se.hglm.b4, CI.se.inla.b4, CI.inla.var_id, CI.hglm.var_id,
   CI.lme4.var_id)
```

```
  rm(coverBet0glm, coverBet0lmer, coverBet0hglm, coverBet0inla, coverBet4glm,
     coverBet4lmer, coverBet4hglm, coverBet4inla)


  ### Data frame with computation times
  comp.time  <- c(mean.time.glm, mean.time.glmer, mean.time.hglm, mean.time.inla)
  ctime.df <- data.frame(ElapsedTime=round(comp.time, 2))
  rownames(ctime.df) <- package
  SimComparison940[[s]][[3]] <- ctime.df

  # Cleaning up
  rm(comp.time, ctime.df)
  rm(mean.time.glm, mean.time.glmer, mean.time.hglm, mean.time.inla)
  rm(alldata, nco, nw, b0, sigZu, estglm, estglmer, esthglm, estinla, package)

  list.comparison <- SimComparison940[[s]]
  list.details    <- SimDetails940[[s]]
  save(list.comparison, list.details, file=paste0("SimulationResults", s+8, ".RData"))
  rm(list.comparison, list.details)
}

rm(s, b4, nrep, simfun2)
comment(SimDetails940) <-
  'List for all 32 simulations with detailed results of all packages'
comment(SimComparison940) <-
  'List for all 32 simulations with comparison criteria'
save(SimSettings940, SimDetails940, SimComparison940,
  file='SimulationResults9to40.RData')
```