The R User Conference 2009
July 8-10, Agrocampus-Ouest, Rennes, France

An R implementation of bootstrap procedures for mixed models

José A. Sánchez-Espigares
Universitat Politècnica de Catalunya
Jordi Ocaña
Universitat de Barcelona

Outline

- Introduction and motivation
- Bootstrap methods for Mixed Models
- Implementation details
- Some examples
- Conclusions

(Generalized) Linear Mixed Models

- Repeated measures or Longitudinal data:

  Response vector Y_i for i^{th} subject $\quad Y_i = (Y_{i1},...,Y_{in_i})'$

  Observations on the same unit can be correlated

- Conditional / Hierarchical approach:

  Between-subject variability explained by Random-effects b_i usually with Normal distribution $\quad b_i \sim N(0, D)$

$$E(Y_{ij} \mid b_i) = \mu_{ij}$$

$$g(\mu_{ij}) = \theta_{ij} = X_{ij}\beta + Z_{ij}b_i$$

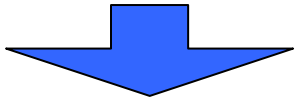
Estimation in (G)LMM

- Random-effects are not directly observed
- Estimation of parameters based on Marginal Likelihood, after integration of Random-effects

$$L(\beta, D, \phi \mid Y) = \prod_{i=1}^{n} f_i(Y_i \mid \beta, D, \phi) =$$

$$\prod_{i=1}^{n} \int \prod_{j=1}^{n_i} f_{ij}(Y_{ij} \mid \beta, b_i, \phi)\, f(b_i \mid D) db_i$$

- MLE:
  - Analytic solution in the Normal case (Linear Mixed Models)
  - Approximations are needed in the general case.



- *lme4* package: common framework for L-GL-NL/MM
  - Fast and efficient estimation for ML and REML criteria via Laplace Approximation/Adaptative Gaussian Quadrature for GLMM.

Inference (G)LMM

- Wald-type and F-tests (`summary`)
  - Asymptotic standard errors for the fixed effects parameters
- Likelihood ratio test (`anova`)
  - Comparison of likelihood of two models
- Bayesian Inference (`mcmcsamp`)
  - MCMC sampling procedure for posteriors on parameters

- Some drawbacks:
  - Asymptotic results
  - Degrees of freedom of the reference distribution in F-test
  - Likelihood Ratio test can be conservative under some conditions
  - Tests on Variance components close to the boundary of the parameter space.

Motivation

- Inference based on bootstrap for LMM and GLMM
- Inference on functions of parameters

  i.e. confidence intervals and hypothesis test for ratio of variance components

- Robust approaches

  i.e. in presence of influential data and outliers

- Effect of misspecification

  i.e. non-gaussian random effects and/or residuals

Extension of the package **`lmer`**

- ***`merBoot`*** provides methods for Monte Carlo and bootstrap techniques in generalized and linear mixed-effects models
- The implementation is object-oriented
- It takes profit of specificities of the applied algorithms to enhance efficiency, using less time and memory.
- It has a flexible interface to design complex experiments.

Bootstrap in linear models

- For (Generalized) linear models (without random effects) there is only one random component ➔ generation of the response variable according to the conditional mean.

$$\mu = X\beta \qquad Y \sim N(\mu, \sigma)$$

$$\mu = g^{-1}(X\beta) \quad Y \sim F_{\mu}$$

- Residual resampling:
  - Estimate parameters for the systematic part of the model
  - Resample random part of the model (parametric or empirical)
  - Some variants to deal with heterocedasticity (Wild bootstrap)

Bootstrap in Mixed Models

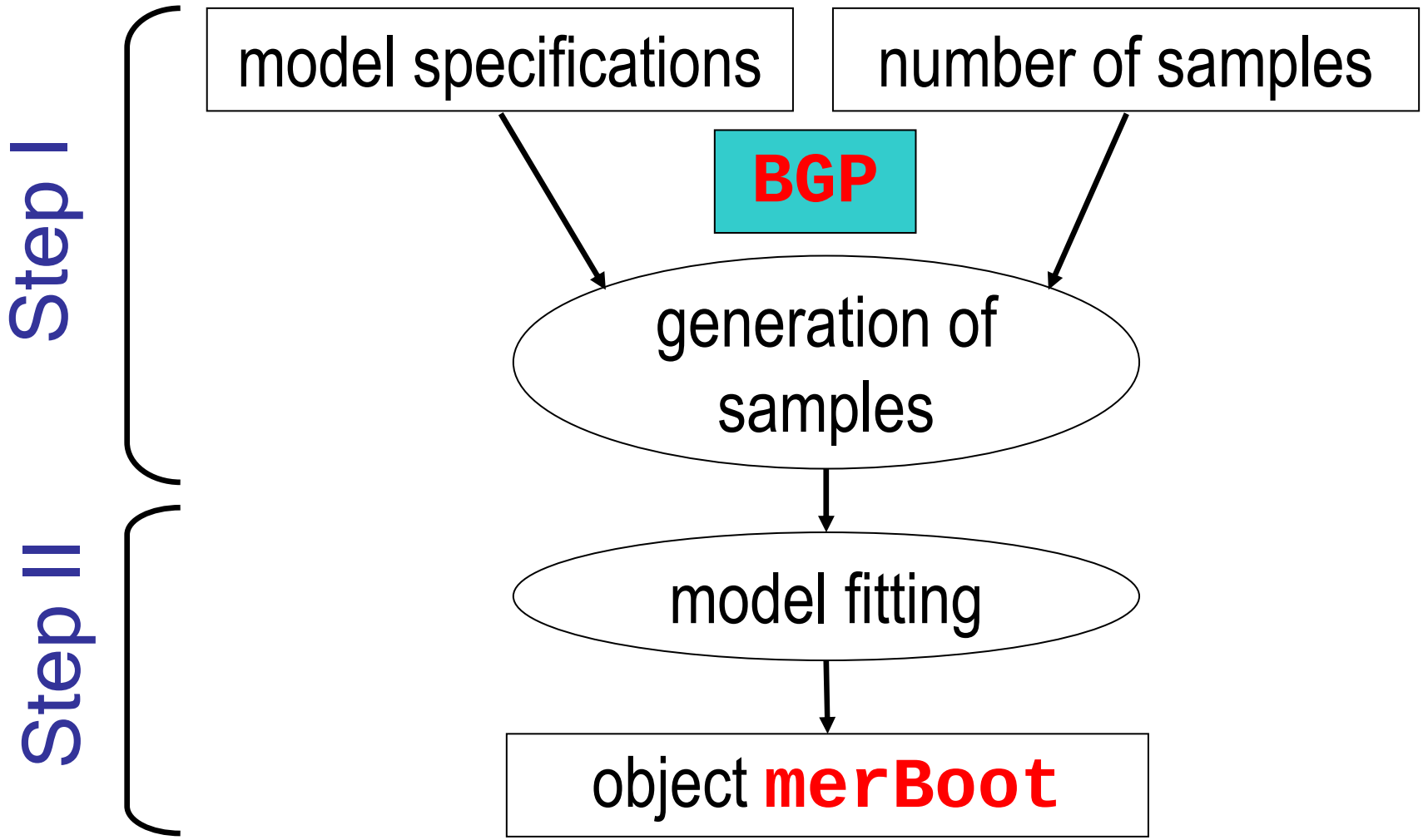
- In Mixed models, the systematic part has a random component ➔ generation of the response variable in two steps:
  - Bootstrap of the conditional mean (function of the linear predictor)
  - Bootstrap of the response variable

$$\mu = X\beta + Zb \qquad b \sim N(0,\theta) \qquad Y \sim N(\mu,\sigma)$$

$$\mu = g^{-1}(X\beta + Zb) \qquad b \sim N(0,\theta) \qquad Y \sim F_{\mu}$$

- Two objects in the **`merBoot`** implementation:
  - **`BGP`**: Set-up for the Bootstrap Generation Process
  - **`merBoot`**: Coefficients for the resamples and methods for analysis

Implementation details



Bootstrap Generation Process

Linear predictor level

- Fixed parameters β
- Design matrices X_i Z_i
- Random effects generator (b_i^*)
  - Parametric: generating b_i^* from a multivariate gaussian distribution
  - Semiparametric/Empirical (from a fitted object): sampling b_i^* from $\hat{b}_i$ with replacement.
  - User-defined: any other distribution/criteria to generate b_i^*

$$\eta_{ij}^* = X_{ij}\beta + Z_{ij}b_i^*$$

Bootstrap Generation Process

Response level

- Family (distribution F + link function g)
- Response generator (Y_{ij}^*)
  - Parametric: $\mu_{ij}^* = g^{-1}(\eta_{ij}^*)$, sample $Y_{ij}^* \sim F(.;\, \mu_{ij}^*)$
  - Semiparametric/Empirical (from a fitted object):
    - Residual-based: builds Y_{ij}^* like in linear heterocedastic models, depending on type of residuals

$$Y_{ij}^* = \mu_{ij}^* + \varepsilon_i^*$$

    - Distribution-based: resamples estimated quantiles

Residuals in GLM

- Raw residuals: $Y_{ij} - \hat{\mu}_{ij}$
- Pearson residuals: $e_{ij} = \frac{Y_{ij} - \hat{\mu}_{ij}}{\sqrt{a_{ij} V(\hat{\mu}_{ij})}}$
- Standardized Pearson residuals: $r_{ij} = \frac{e_{ij}}{\sqrt{1 - h(\hat{\mu}_{ij})}}$
- Standardized residuals on the linear predictor scale: $l_{ij} = \frac{g(Y_{ij}) - g(\hat{\mu}_{ij})}{\sqrt{a_{ij} g'(\hat{\mu}_{ij})^2 V(\hat{\mu}_{ij})(1 - h_{ij})}}$
- Deviance residuals: $r_{ij} = sign(Y_{ij} - \hat{\mu}_{ij}) \sqrt{d_{ij}}$

Empirical residual-based

- Standardized Pearson residuals:
  - Resample e_{ij}^{*} from centered e_{ij}
  - Calculate $Y_{ij}^{*} = \mu_{ij}^{*} + \left(\sqrt{\frac{\hat{\phi}}{a_i} V(\mu_{ij}^{*})} \right) e_{ij}^{*}$
- Standardized Pearson residuals on the linear predictor scale:
  - Resample l_{ij}^{*} from l_{ij}
  - Calculate $Y_{ij}^{*} = g^{-1}\left(\eta_{ij}^{*} + \left(g'(\eta_{ij}^{*}) \sqrt{\frac{\hat{\phi}}{a_i} V(\mu_{ij}^{*})} \right) l_{ij}^{*} \right)$

Empirical distribution-based

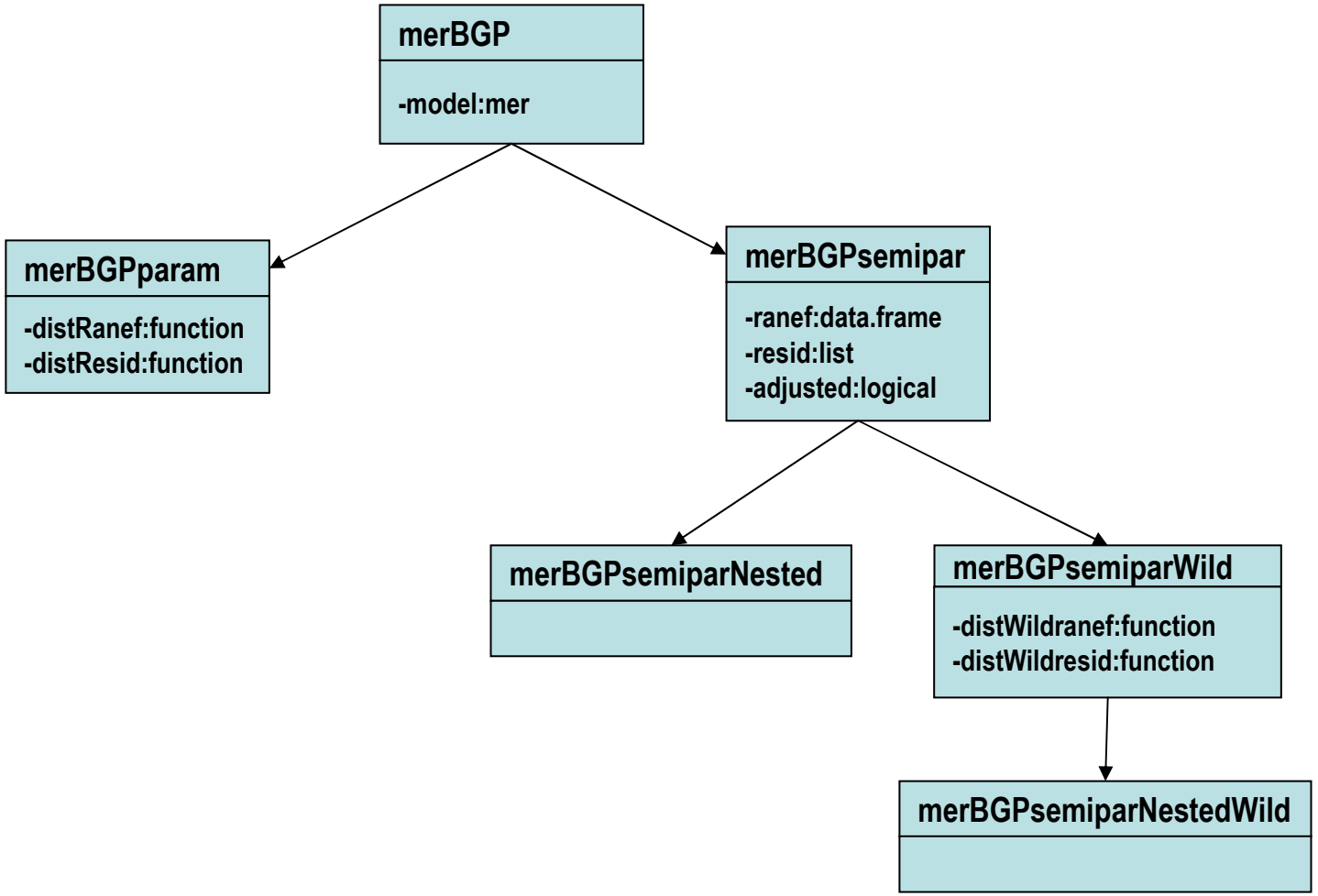
- Randomized Quantile residuals (Dunn & Smyth,1996):
  - inverting the estimated distribution function for each observation to obtain exactly uniform (q_{ij}) or standard normal residuals ($\Phi^{-1}(q_{ij})$) for diagnostics. Randomization needed for discrete distributions.

- Resampling scheme with Quantiles Residuals for (G)LMM:
  - Calculate $\hat{q}_{ij} = F(Y_{ij}; \hat{\mu}_{ij})$
  - Sample q_{ij}^* with replacement from $\hat{q}_{ij}$
  - Generate $Y_{ij}^* = F^{-1}(q_{ij}^*; \hat{\mu}_{ij}^*)$

Response generation

- For Normal family and identity link function, all three strategies (pearson, linear predictor and quantile residuals) are the same.
- In all the schemes, response is rounded to the nearest valid value, according to the family considered.
- For discrete variables, randomization of the quantiles allows for continuous uniform residuals.
- Transformation of the random effects in order to have the first and second moments equal to the parameters (adjusted bootstrap).
- For all the schemes, if resample of residuals/quantiles is restricted to the subject obtained in the linear predictor level, a nested bootstrap is performed.

Bootstrap Generation Process

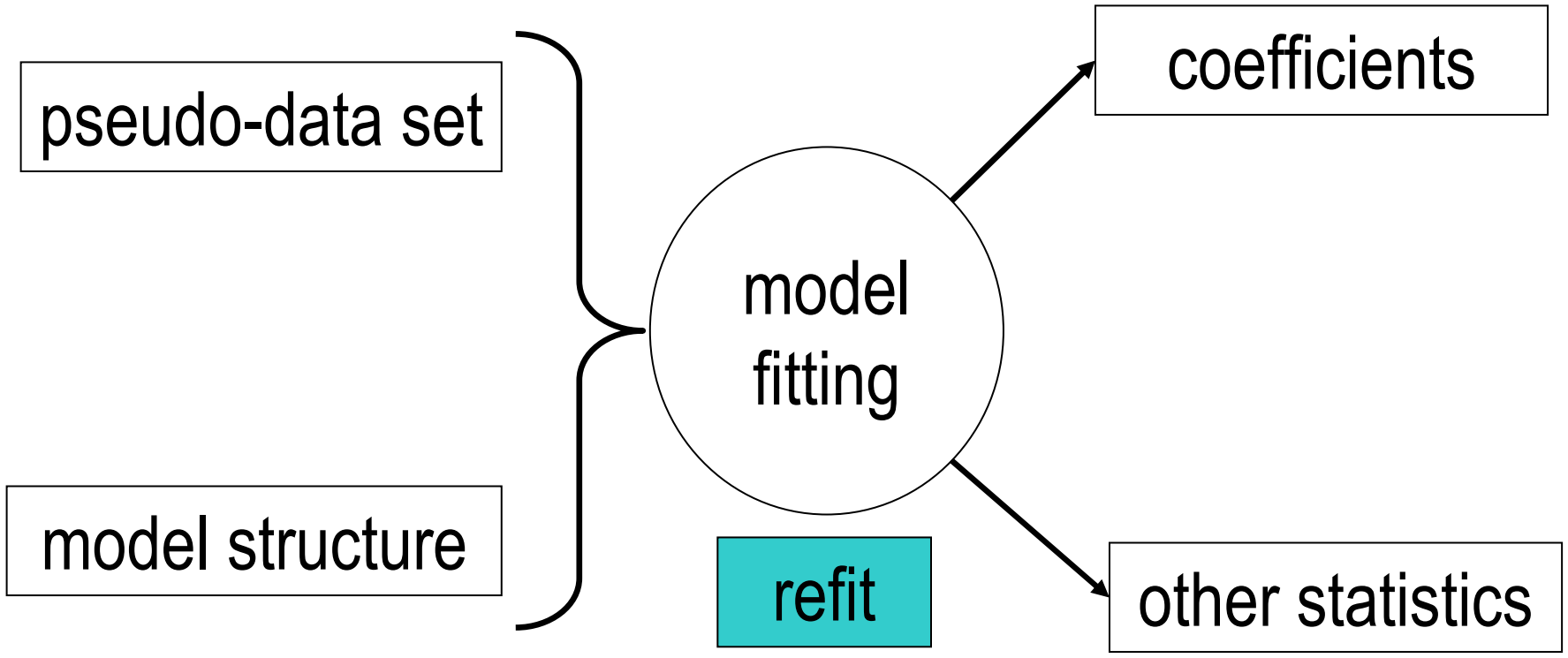


BGP Methods

- generateLinpred
  - BGPparam: $b_i^* \sim F_{\hat{\theta}}$ $\eta_{ij} = X_{ij}\beta + Z_{ij}b_i^*$
  - BGPsemipar: $b_i^* \sim \hat{F}_n(.,b_i)$ $\eta_{ij} = X_{ij}\beta + Z_{ij}b_i^*$
  - BGPsemiparWild: $b_i^* \sim \hat{F}_n(.,b_i)$ $w_i^* \sim W$ $\eta_{ij} = X_{ij}\beta + Z_{ij}w_i^*b_i^*$
- generate (lmer) ➔ generateLinpred +Residual-based
  - BGPparam: $\varepsilon_{ij}^* \sim F_\theta$ $Y_{ij}^* = \mu_{ij}^* + \varepsilon_{ij}^*$
  - BGPsemipar: $\varepsilon_{ij}^* \sim \hat{F}_n(.,e_{ij})$ $Y_{ij}^* = \mu_{ij}^* + \varepsilon_{ij}^*$
  - BGPsemiparWild: $\varepsilon_{ij}^* \sim \hat{F}_n(.,e_{ij})$ $w_i^* \sim W$ $Y_{ij}^* = \mu_{ij}^* + w_i^*\varepsilon_{ij}^*$
  - BGPsemiparNested $\varepsilon_{ij}^* \sim \hat{F}_n(.,e_{i^*j})$ $Y_{ij}^* = \mu_{ij}^* + \varepsilon_{ij}^*$
- generate (glmer) ➔ generateLinpred +Distribution-based
  - BGPparam:
  - BGPsemipar: $q_{ij}^* \sim F_{\mu_{ij}}$ $Y_{ij}^* = F_{\mu_{ij}^*}^{-1}(q_{ij}^*)$
  - BGPsemiparWild:
  - BGPSemiparNested

Object **`merBoot`**

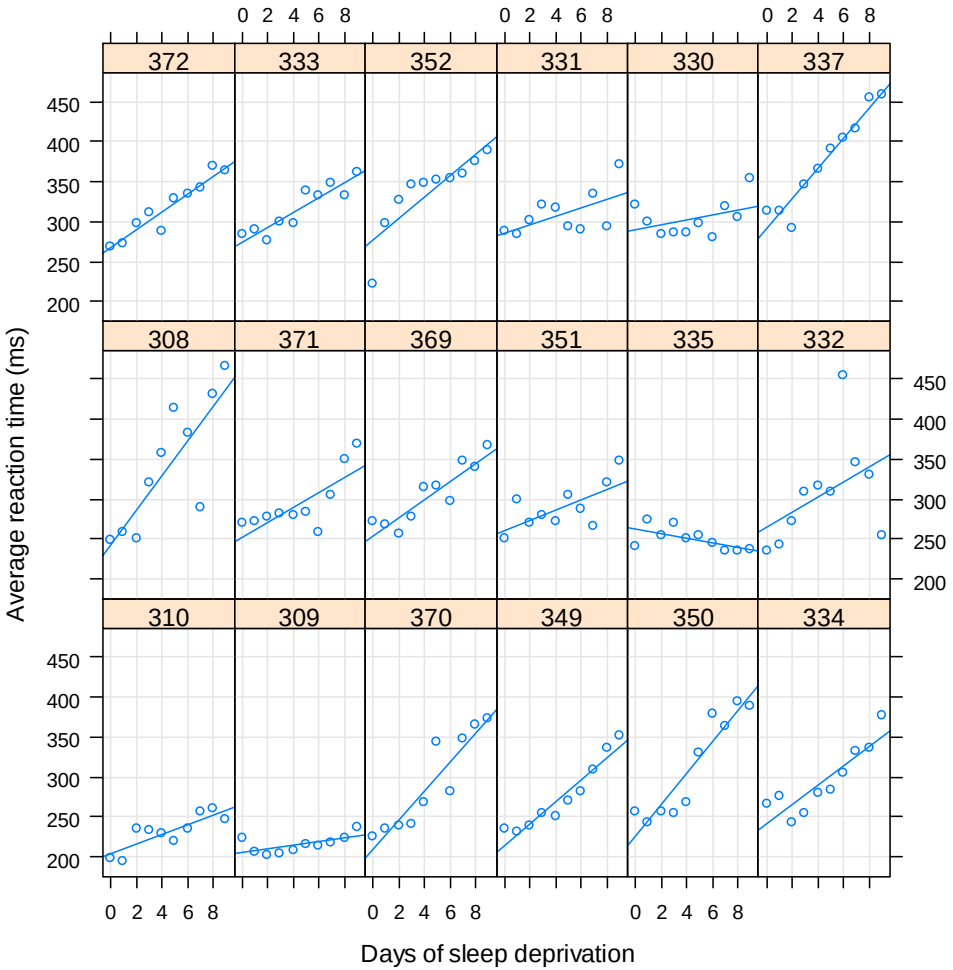
Step II: Extraction of coefficients



Bootstrap method for (g)lmer

object	• Model specification to generate pseudo-data set (*glme* object or a list). • It contains the formula describing the structure, the parameters (β,D) and the design matrix (X,Z)
distref **distres** **adj** **nest** **wild**	• Parameters to indicate how to generate variance components (random-effects) and response. • Strings (for pre-implemented options) or functions (user-defined methods)
B	• Number of samples.
model2	• Alternative model specification used to fit the pseudo-data. Default is same as object

Example: sleepstudy




```
model=lmer(Reaction~Days+(1|Subject)+
(0+Days|Subject),sleepstudy)

Linear mixed model fit by REML
Formula: form
   Data: sleepstudy
  AIC  BIC logLik deviance REMLdev
 1754 1770 -871.8     1752    1744
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept) 627.568  25.0513
 Subject  Days         35.858   5.9882
 Residual             653.584  25.5653
Number of obs: 180, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  251.405      6.885   36.51
Days          10.467      1.559    6.71

Correlation of Fixed Effects:
     (Intr)
Days -0.184
```

Methods **merBoot**: print

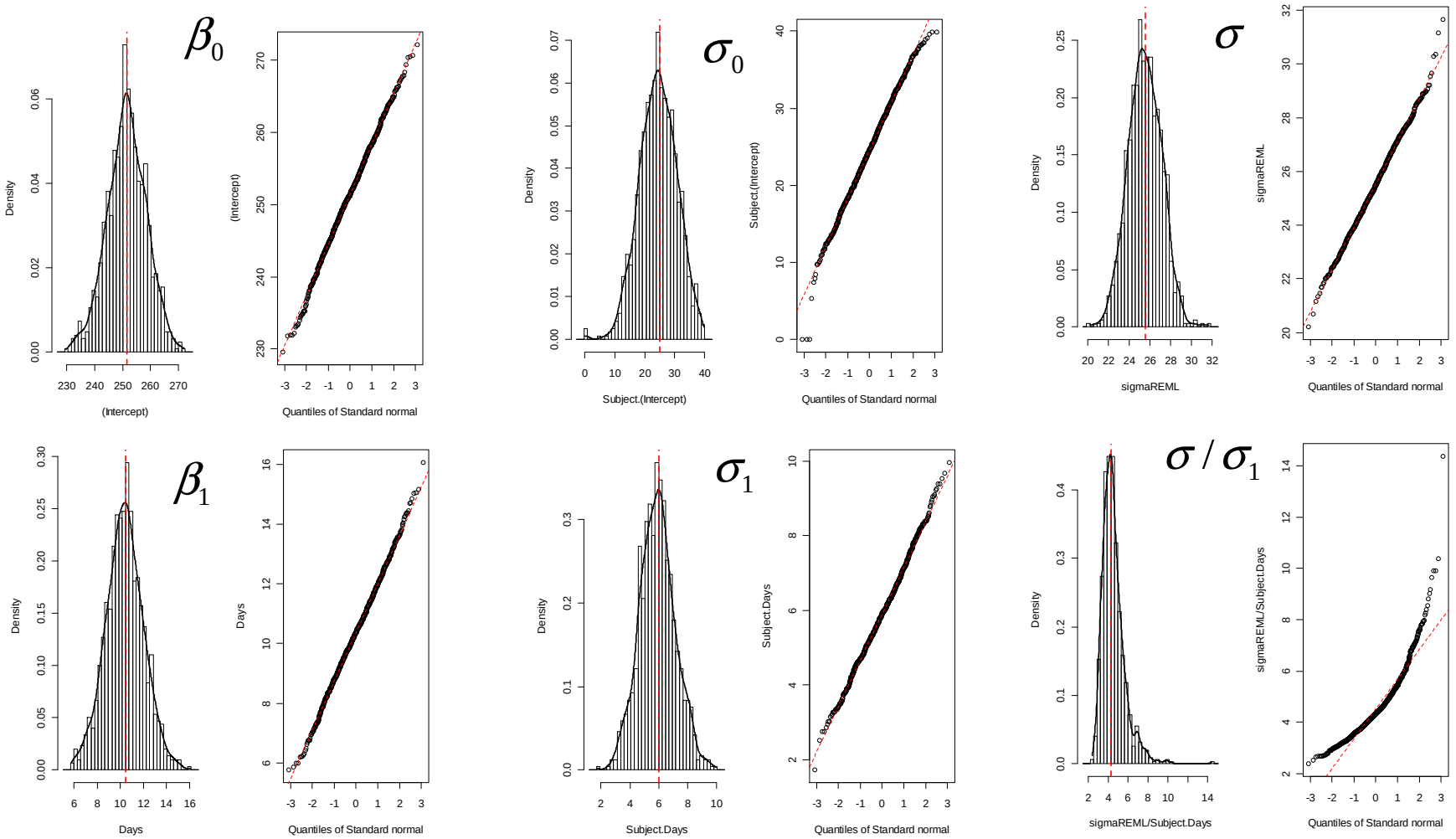
```
> sleep.boot=bootstrap(model,B=1000)
> print(sleep.boot)

lmer(formula = lmer(Reaction~Days+(1|Subject)+(0+Days|Subject),
data = sleepstudy)

Resampling Method: BGPparam

Bootstrap Statistics :
                     original      bias       mean    std. error
(Intercept)        251.405105  0.04710710 251.452212    6.971430
Days                10.467286 -0.08913201  10.378154    1.627212
Subject.(Intercept) 25.051299 -0.56896799  24.482331    6.222008
Subject.Days         5.988173 -0.08323923   5.904934    1.231645
sigmaREML           25.565287 -0.04783798  25.517449    1.582567
```

Methods **merBoot**: intervals

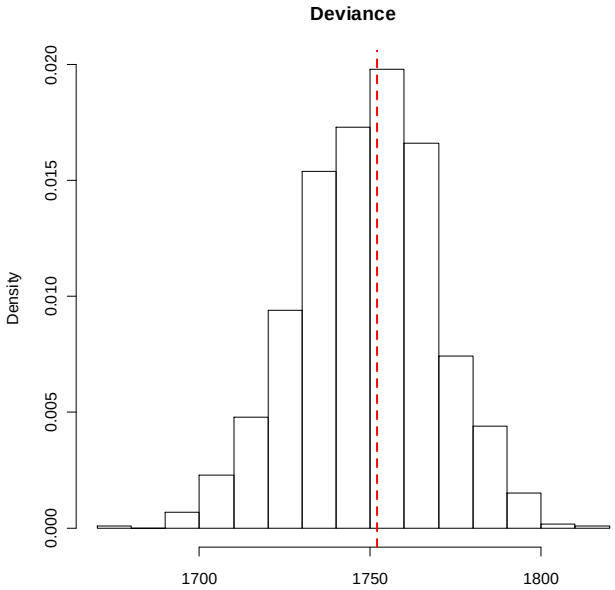
```
> intervals(sleep.boot)
$norm
                         lower      upper
(Intercept)          237.694245 265.021750
Days                   7.367141  13.745695
Subject.(Intercept)   13.425355  37.815179
Subject.Days           3.657432   8.485393
sigmaREML             22.511352  28.714899

$basic
                         lower      upper
(Intercept)          237.913080 265.836054
Days                   7.351741  13.869182
Subject.(Intercept)   13.456205  37.647603
Subject.Days           3.600202   8.499955
sigmaREML             22.527143  28.691811

$perc
                         lower      upper
(Intercept)          236.974156 264.897129
Days                   7.065390  13.582831
Subject.(Intercept)   12.454995  36.646393
Subject.Days           3.476392   8.376144
sigmaREML             22.438764  28.603432
```

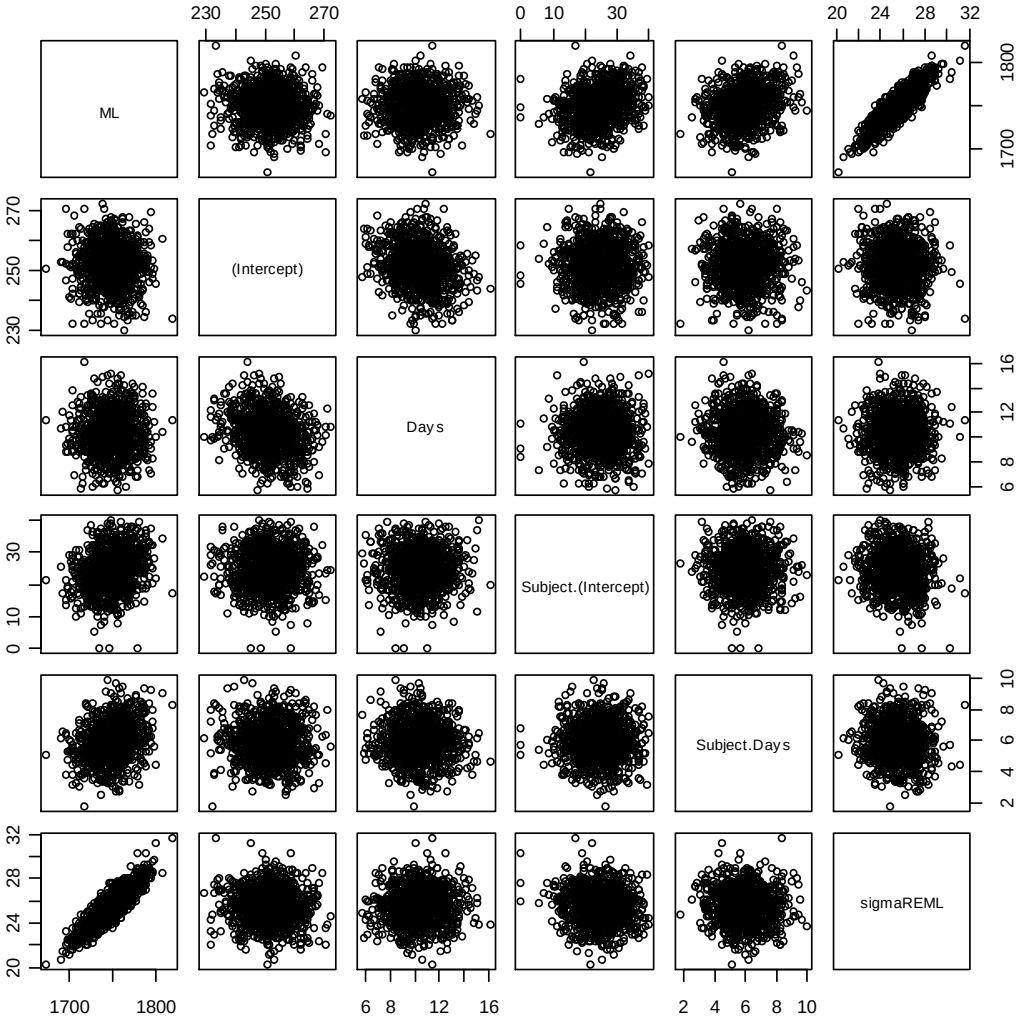
```
> HPDinterval(mcmcsamp(model,n=1000))
$fixef
                   lower     upper
(Intercept) 241.741098 261.77774
Days          7.370217  13.88645
attr(,"Probability")
[1] 0.95

$ST
          lower     upper
[1,] 0.3117272 0.8478526
[2,] 0.1485967 0.3317120
attr(,"Probability")
[1] 0.95

$sigma
         lower    upper
[1,] 23.88066 29.66232
attr(,"Probability")
[1] 0.95
```

Methods **merBoot**: plot



Methods **merBoot**: LR & pairs





Conclusions

- (G)LMM provide a framework to model longitudinal data for a wide range of situations (continuous, count and binary among others) but approximate estimation methods imply weaker inference.
- Monte-Carlo simulation and bootstrap can enhance inference providing empirical p-values and bootstrap estimators.
- The **`BGP/merBoot`** objects developed allow implementation of different bootstrap options and evaluation of the techniques via simulation.