

# Multilevel Models for Big Data

## Approaches for handling very large datasets

Ömercan Mısırlıoğlu, Yordan Saputra

Department of Statistics  
TU Dortmund University

Final Presentation for Multilevel Models Seminar WS25

# Table of Contents

## ① Introduction

- Multilevel Models
- Issues with Large Datasets

## ② The split-sample approach

- Pseudo Likelihood
- Graphical representation
- Independent subsamples
- Dependent subsamples
- Overlapping subsamples

## ③ R Packages

- lme4 and mgcv
- Why use bam()?
- When to use bam()?

## ④ Summary

## ⑤ References

## What are multilevel models?

- Hierarchical Structure
- Residual Components
- Variance Partitioning

## Why use multilevel models?

- Correct Inferences
- Group Effects Estimation
- Simultaneous Estimation
- Generalization Beyond Sample <sup>1</sup>

## Statistical Model

$$y_n \sim \mathcal{N}(\mu_n, \sigma)$$

$$\mu_n = b_0 + \sum_{p=1}^P b_p x_{pn} + \tilde{b}_{0j[n]} + \sum_{p=1}^P \tilde{b}_{pj[n]} x_{pn}$$

---

<sup>1</sup>Centre for Multilevel Modelling 2025.

# Multilevel Models for Big Data

## Introduction

## Multilevel Models

## Introduction – Multilevel Models

### What are multilevel models?

- Hierarchical Structure
- Residual Components
- Variance Partitioning

### Why use multilevel models?

- Correct Inferences
- Group Effects Estimation
- Simultaneous Estimation
- Generalization Beyond Sample<sup>1</sup>

### Statistical Model

$$y_{in} \sim \mathcal{N}(\mu_{in}, \sigma^2)$$

$$\mu_{in} = b_0 + \sum_{p=1}^P b_{p0}x_{ip0} + \tilde{b}_{0[i]q} + \sum_{p=1}^P \tilde{b}_{p[i]q}x_{ipq}$$

<sup>1</sup>Centre for Multilevel Modelling 2025.

### What are multilevel models?

- **Hierarchical Structure:** Observational data often feature individuals nested within higher-level groups, such as schools, workplaces, or geographical areas.
- **Residual Components:** Multilevel models account for these hierarchies by incorporating residual components at every level of the data structure.
- **Variance Partitioning:** These models divide residual variance into between-group and within-group components to capture unobserved factors influencing outcomes.

### Why use multilevel models?

- **Correct Inferences:** Traditional methods assume independent observations, which are often false. Additionally, ignoring hierarchical structures can lead to underestimated standard errors and overstated statistical significance.
- **Group Effects Estimation:** Directly quantify between-group variation and identify outlying groups.
- **Simultaneous Estimation:** Unlike fixed effects models, the separation of observed and unobserved group characteristics is possible, allowing for simultaneous estimation of group-level and individual-level effects.
- **Generalization Beyond Sample:** Unlike fixed effects models which only describe sampled groups, multilevel models treat groups as random samples from a population, enabling generalizations to unobserved groups.

## Multilevel Models for Big Data

- └ Introduction
  - └ Multilevel Models
    - └ Introduction – Multilevel Models

Introduction – Multilevel Models

**What are multilevel models?**

- Hierarchical Structure
- Residual Components
- Variance Partitioning

**Why use multilevel models?**

- Correct Inferences
- Group Effects Estimation
- Simultaneous Estimation
- Generalization Beyond Sample <sup>1</sup>

**Statistical Model**

$$y_n \sim \mathcal{N}(\mu_n, \sigma)$$

$$\mu_n = b_0 + \sum_{p=1}^P b_p x_{pn} + \tilde{b}_{0j[n]} + \sum_{p=1}^P \tilde{b}_{pj[n]} x_{pn}$$

<sup>1</sup>Centre for Multilevel Modelling 2025.

## Statistical Model

$$y_n \sim \mathcal{N}(\mu_n, \sigma)$$

$$\mu_n = b_0 + b_1 x_{1n} + \dots + b_p x_{pn} + \tilde{b}_{0j[n]} + \tilde{b}_{1j[n]} x_{1n} + \dots + \tilde{b}_{pj[n]} x_{pn}$$

where:

- $y_n$ : dependent variable for observation  $n$
- $x_{pn}$ : predictor variable  $p$  for observation  $n$
- $b_p$ : overall slope (or intercept for  $p = 0$ ) for predictor  $p$  across all groups
- $\tilde{b}_{pj[n]}$ : random effect of predictor  $p$  for group  $j$  that observation  $n$  belongs to
- $\sigma$ : residual standard deviation (assumed constant across observations)

# Introduction — Issues with Large Datasets

As hierarchical data scales ( $N$  groups  $\times$   $n$  individuals), massive datasets create the following issues:

- High number of groups
- Large group sizes<sup>2</sup>
- Design matrix construction<sup>3</sup>

---

<sup>2</sup>Clark 2019; Speelman, Heylen, and Geeraerts 2018.

<sup>3</sup>**wood2014.**

# Multilevel Models for Big Data

## └ Introduction

### └ Issues with Large Datasets

#### └ Introduction – Issues with Large Datasets

As hierarchical data scales ( $N$  groups  $\times$   $n$  individuals), massive datasets create the following issues:

- High number of groups
- Large group sizes<sup>2</sup>
- Design matrix construction<sup>3</sup>

<sup>2</sup>Clark 2009; Spielman, Hayfin, and Geierarts 2018.  
<sup>3</sup>weiss2014.

#### Issues with Large Datasets

- Large number of groups: Computational bottleneck from numerical integration over random effects for each group at every optimization step, leading to high computational costs
- Large group sizes: High-dimensional multivariate distributions create numerical issues with large covariance matrix inversion, even in linear models
- construction of the full design matrix  $X$  leads to high computational costs. In GAM, the estimator  $\hat{\beta} = (X^T X + \sum \lambda_j S_j)^{-1} X^T y$  is hard to compute when  $X$  is large.
- it's even worse when we add another level to the hierarchy, e.g. students nested within classes nested within schools, which is common in educational research.

# Table of Contents

## ① Introduction

- Multilevel Models
- Issues with Large Datasets

## ② The split-sample approach

- Pseudo Likelihood
- Graphical representation
- Independent subsamples
- Dependent subsamples
- Overlapping subsamples

## ③ R Packages

- lme4 and mgcv
- Why use bam()?
- When to use bam()?

## ④ Summary

## ⑤ References

# The split-sample approach – Pseudo Likelihood

- Consider the log-likelihood function  $\ell(\boldsymbol{\theta}) = \sum_i \ell(\mathbf{y}_i | \boldsymbol{\theta})$  where  $\mathbf{y}_i$  is the vector of all observations in group  $i$
- Replaces the log-likelihood contribution  $\ell(\mathbf{y}_i | \boldsymbol{\theta})$  by a weighted sum of log-likelihood contributions for sub-vectors  $\mathbf{Y}_i^{(s)}$
- More specifically, the pseudo-log-likelihood function:

$$p\ell(\boldsymbol{\psi}) = \sum_i \sum_s \delta_s \ell(\mathbf{y}_i^{(s)} | \boldsymbol{\psi})$$

is maximized instead with respect to  $\boldsymbol{\psi}$ , which is not necessarily identical to  $\boldsymbol{\theta}$

- Although  $\hat{\boldsymbol{\psi}}$  is not the MLE estimate, it still has similar properties such as consistency and asymptotic normality<sup>4</sup>

---

<sup>4</sup>Clark 2019.

## Multilevel Models for Big Data

## └ The split-sample approach

## └ Pseudo Likelihood

## └ The split-sample approach – Pseudo Likelihood

- Consider the log-likelihood function  $l(\theta) = \sum_i l(y_i|\theta)$  where  $y_i$  is the vector of all observations in group  $i$
- Replaces the log-likelihood contribution  $l(y_i|\theta)$  by a weighted sum of log-likelihood contributions for sub-vectors  $y_i^{(s)}$
- More specifically, the pseudo-log-likelihood function:

$$p(\psi) = \sum_i \sum_s \delta_i l(y_i^{(s)}|\psi)$$

is maximized instead with respect to  $\psi$ , which is not necessarily identical to  $\theta$

- Although  $\psi$  is not the MLE estimate, it still has similar properties such as consistency and asymptotic normality<sup>4</sup>

---

<sup>4</sup>Clark 2018.

- Now, how do we split  $y_i$  into sub-vectors  $y_i^{(s)}$ ? There are different ways to do this, and we will discuss some of them in the next slides.
- Going back to the slide on “issues with large datasets”, it’s obvious that we can split the data in two (technically three) different ways: either we can split the data by groups, or we can split the data by observations within groups. The first one is more suitable when we have a large number of groups, while the second one is more suitable when we have a large number of observations within groups.

# The split-sample approach – Graphical representation

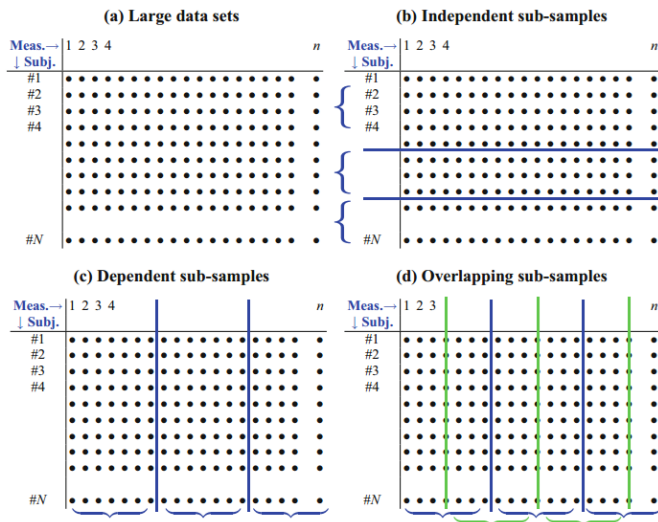


Figure 1: Graphical representation of different ways to split large samples

# The split-sample approach — Independent subsamples

- Shown in panel (b) of Figure 1, dataset with large  $N$  is partitioned into  $M$  independent sets  $S_m$  of groups, where  $m = 1, \dots, M$
- In each subsample, the model is fitted, yielding an estimate  $\hat{\theta}_m$  of  $\theta$ , equivalent to maximizing

$$p\ell(\psi) = \sum_m \sum_{i \in S_m} \ell(\mathbf{y}_i | \theta_m)$$

with respect to  $\psi = \{\theta_1, \dots, \theta_M\}$

- All  $\theta_m$  are equal to  $\theta$ , therefore the estimates  $\hat{\theta}_m$  can be averaged to obtain an overall estimate  $\hat{\theta}$

# Multilevel Models for Big Data

- └ The split-sample approach
  - └ Independent subsamples
    - └ The split-sample approach — Independent subsamples

- Shown in panel (b) of Figure 1, dataset with large  $N$  is partitioned into  $M$  independent sets  $S_m$  of groups, where  $m = 1, \dots, M$
- In each subsample, the model is fitted, yielding an estimate  $\hat{\theta}_m$  of  $\theta$ , equivalent to maximizing

$$p(\psi) = \sum_m \sum_{i \in S_m} \ell(Y_i | \theta_m)$$

with respect to  $\psi = \{\theta_1, \dots, \theta_M\}$

- All  $\theta_m$  are equal to  $\theta$ , therefore the estimates  $\hat{\theta}_m$  can be averaged to obtain an overall estimate  $\hat{\theta}$

- $\theta_m$  are all equal to  $\theta$  because the subsamples are independent
- Mention parallelization here, since we can fit the model on each subsample in parallel, which can significantly reduce the computational time.

# The split-sample approach – Dependent subsamples

- Shown in panel (c) of Figure 1, dataset with large  $n$  is partitioned into  $M$  (not independent) sets  $S_m$  of groups, where  $m = 1, \dots, M$
- Fitting the model on each subsample, equivalent to maximizing

$$p\ell(\psi) = \sum_m \sum_i \ell(\mathbf{Y}_i^{(m)} | \theta_m)$$

with respect to  $\psi = \{\theta_1, \dots, \theta_M\}$ , where  $\mathbf{Y}_i^{(m)}$  is the observations in  $\mathbf{Y}_i$  belonging to subsample  $S_m$ .

- All  $\theta_m$  are not necessarily equal to  $\theta$ , therefore the combination of all  $\hat{\theta}_m$  into a single estimator  $\hat{\theta}$  depends on the precise model and data structure.

# Multilevel Models for Big Data

- └ The split-sample approach
  - └ Dependent subsamples
    - └ The split-sample approach – Dependent subsamples

- Shown in panel (c) of Figure 1, dataset with large  $n$  is partitioned into  $M$  (not independent) sets  $S_m$  of groups, where  $m = 1, \dots, M$
- Fitting the model on each subsample, equivalent to maximizing

$$p(\psi) = \sum_m \sum_i \ell(\mathbf{Y}_i^{(m)} | \theta_m)$$

with respect to  $\psi = \{\theta_1, \dots, \theta_M\}$ , where  $\mathbf{Y}_i^{(m)}$  is the observations in  $\mathbf{Y}_i$  belonging to subsample  $S_m$ .

- All  $\theta_m$  are not necessarily equal to  $\theta$ , therefore the combination of all  $\hat{\theta}_m$  into a single estimator  $\hat{\theta}$  depends on the precise model and data structure.

- $\theta_m$  are not necessarily equal to  $\theta$  because the subsamples are not independent, and there may be some correlation between the observations in different subsamples.
- (GPT warning, dont trust 100%) The combination of all  $\hat{\theta}_m$  into a single estimator  $\hat{\theta}$  can be done using various methods, such as meta-analysis techniques, or by fitting a model to the estimates  $\hat{\theta}_m$  themselves.

# The split-sample approach — Overlapping subsamples

- Shown in panel (d) of Figure 1, dataset with large  $n$  is partitioned similarly to dependent subsamples, but association between observations is accounted for by letting the subsamples overlap
- Denoting the parameters in pair  $\{\mathbf{Y}_i^{(p)}, \mathbf{Y}_i^{(q)}\}$  by  $\theta_{p,q}$ , fitting the models on all pairs is equivalent to maximizing

$$p\ell(\psi) = \sum_{p < q} \sum_i \ell(\mathbf{Y}_i^{(p)}, \mathbf{Y}_i^{(q)} | \theta_{p,q})$$

with respect to  $\psi = \{\theta_{1,2}, \theta_{1,3}, \dots, \theta_{Q-1,Q}\}$ , where  $\mathbf{Y}_i^{(p)}$  and  $\mathbf{Y}_i^{(q)}$  are the observations in  $\mathbf{Y}_i$  belonging to subsamples  $S_p$  and  $S_q$ , respectively.

- Similarly, the combination of all  $\hat{\theta}_{p,q}$  into a single estimator  $\hat{\theta}$  depends on the precise model and data structure.

# Multilevel Models for Big Data

- └ The split-sample approach
  - └ Overlapping subsamples
    - └ The split-sample approach — Overlapping subsamples

- Shown in panel (d) of Figure 1, dataset with large  $n$  is partitioned similarly to dependent subsamples, but association between observations is accounted for by letting the subsamples overlap
- Denoting the parameters in pair  $\{\mathbf{Y}_i^{(p)}, \mathbf{Y}_i^{(q)}\}$  by  $\theta_{p,q}$ , fitting the models on all pairs is equivalent to maximizing

$$p(\psi) = \sum_{p < q} \sum_i \ell(\mathbf{Y}_i^{(p)}, \mathbf{Y}_i^{(q)} | \theta_{p,q})$$

with respect to  $\psi = (\theta_{1,2}, \theta_{1,3}, \dots, \theta_{Q-1,Q})$ , where  $\mathbf{Y}_i^{(p)}$  and  $\mathbf{Y}_i^{(q)}$  are the observations in  $\mathbf{Y}_i$  belonging to subsamples  $S_p$  and  $S_q$ , respectively.

- Similarly, the combination of all  $\hat{\theta}_{p,q}$  into a single estimator  $\hat{\theta}$  depends on the precise model and data structure.

- without the pairwise fitting, we have to fit the model on the entire dataset, which is computationally infeasible when  $n$  is very large. By fitting the model on pairs of subsamples, we can reduce the computational burden while still accounting for the association between longitudinal observations.
- not gonna go into details here since this is more suitable for longitudinal data, which is not the focus of our presentation, but the idea is similar to dependent subsamples
- similarly,  $\psi = \{\theta_{1,2}, \theta_{1,3}, \dots, \theta_{Q-1,Q}\} = \{\theta_{p,q} : p < q\}$
- what if both  $n$  and  $N$  are large? there is no mention of this in the literature, should we mention this?

# Table of Contents

## ① Introduction

- Multilevel Models
- Issues with Large Datasets

## ② The split-sample approach

- Pseudo Likelihood
- Graphical representation
- Independent subsamples
- Dependent subsamples
- Overlapping subsamples

## ③ R Packages

- lme4 and mgcv
- Why use bam()?
- When to use bam()?

## ④ Summary

## ⑤ References

## lme4

- an R package for fitting linear and generalized linear mixed-effects (multilevel) models<sup>5</sup>
- efficient, able to handle large sample sizes for simple model, and process hundreds of thousands observations on a typical laptop
- Modeling functions: `lmer()` and `glmer()`

## mgcv

- an R package for fitting generalized additive model and generalized additive mixed models<sup>6</sup>
- Modeling functions: `gam()` and `bam()`

---

<sup>5</sup>Bates et al. 2015.

<sup>6</sup>S. N. Wood 2011.

# Multilevel Models for Big Data

## └ R Packages

### └ lme4 and mgcv

### └ R Packages – lme4 and mgcv

#### lme4

- an R package for fitting linear and generalized linear mixed-effects (multilevel) models<sup>3</sup>
- efficient, able to handle large sample sizes for simple model, and process hundreds of thousands of observations on a typical laptop
- Modeling functions: `lmer()` and `glmer()`

#### mgcv

- an R package for fitting generalized additive model and generalized additive mixed models<sup>4</sup>
- Modeling functions: `gam()` and `bsm()`

<sup>3</sup>Bates et al. 2015.

<sup>4</sup>S. N. Wood 2011.

can start by saying "in practice, optimization on large datasets are much more complex than what we have shown here, and a short presentation is not enough to cover all the details. we have two packages (used for large datasets) that we want to cover here, that is lme4 and mgcv"

lme4:

- is an R package for fitting linear and generalized linear mixed-effects (multilevel) models using 'Eigen' C++ library and S4 classes. (or just say using C++ library). Eigen is a high-level C++ template library for linear algebra that provides efficient, header-only classes for managing matrices, vectors, and numerical solvers. S4 is a formal system in R for object-oriented programming that uses strictly defined classes and methods to ensure data integrity and facilitate complex statistical modeling.
- It's computationally efficient, enabling it to handle very large sample sizes for simpler mixed models and to process hundreds of thousands of observations with random effects on a typical laptop.
- `lmer()`: fits linear multilevel model using restricted maximum likelihood (REML) or maximum likelihood estimation. `glmer()`: fits generalized linear multilevel model, accommodating non-normal response distributions. basically, `lmer()` for linear models and `glmer()` for GLM

mgcv:

- is an R package for fitting generalized additive models (GAMs) and generalized additive mixed models (GAMMs) using penalized regression splines.
- `gam()`: fits generalized additive multilevel models using penalized regression splines with smooth terms that can incorporate multilevel structure through random effect splines. `bsm()`: a computationally efficient version of `gam()` optimized for very large datasets.

# R Packages – Why use bam()?

- Same underlying model between `gam()` and `lme4`, with differences in parameter estimation
- How `bam()` works:
  - QR decomposition<sup>7</sup>
  - (i) Efficient fitting algorithm, (ii) Parallel computation, and (iii) Covariate discretization<sup>8</sup>
  - Efficient crossproduct matrix  $X^T W X$  computation<sup>9</sup>
- Discretization on large datasets leads to tradeoff between accuracy and speed

---

<sup>7</sup>S. Wood, Goude, and Shaw 2015.

<sup>8</sup>S. Wood, Li, et al. 2017.

<sup>9</sup>Li and S. Wood 2020.

# Multilevel Models for Big Data

## └ R Packages

### └ Why use bam()?

#### └ R Packages – Why use bam()?

- Same underlying model between `gam()` and `lme4`, with differences in parameter estimation
- How `bam()` works:
  - QR decomposition<sup>7</sup>
  - (i) Efficient fitting algorithm, (ii) Parallel computation, and (iii) Covariate discretization<sup>8</sup>
  - Efficient crossproduct matrix  $X^T W X$  computation<sup>9</sup>
- Discretization on large datasets leads to tradeoff between accuracy and speed

<sup>7</sup>S. Wood, Goude, and Shaw 2015.

<sup>8</sup>S. Wood, Li, et al. 2017.

<sup>9</sup>Li and S. Wood 2020.

- The underlying model between `gam()` function and `lme4` is the same, with differences only in the way parameters are estimated.
- `bam()` employs parallelized computation on model matrix subsets and optional data discretization to extract minimal necessary information, enabling efficient estimation of large multilevel models.
- Discretization has negligible impact on parameter estimates (differing only at high decimal precision), but leads to dramatic speed improvements.

# Multilevel Models for Big Data

## └ R Packages

### └ Why use bam()?

#### └ R Packages – Why use bam()?

- Same underlying model between `gam()` and `lme4`, with differences in parameter estimation
- How `bam()` works:
  - QR decomposition<sup>7</sup>
  - (i) Efficient fitting algorithm, (ii) Parallel computation, and (iii) Covariate discretization<sup>8</sup>
  - Efficient crossproduct matrix  $X^T W X$  computation<sup>9</sup>
- Discretization on large datasets leads to tradeoff between accuracy and speed

<sup>7</sup>S. Wood, Goude, and Shaw 2015.<sup>8</sup>S. Wood, Li, et al. 2017.<sup>9</sup>Li and S. Wood 2020.

## QR decomposition

- QR decomposition is a method for decomposing a matrix into a lower triangular matrix and an upper triangular matrix.
- Fitting GAM  $\hat{\beta} = (X^T X + \sum \lambda_j S_j)^{-1} X^T y$  becomes  $\hat{\beta} = (R^T R + \sum \lambda_j S_j)^{-1} R^T R$  where  $X = QR$ ,  $X$  is the design matrix,  $S_j$  is the penalty matrix, and  $\lambda_j$  are smoothing parameters.

### (i) Efficient fitting algorithm, (ii) Parallel computation, and (iii) Covariate discretization

- (i) Efficient fitting algorithm: which required only basic easily parallelized matrix computations and a pivoted Cholesky decomposition
- (ii) Parallel computation: the use of a scalable parallel block pivoted Cholesky algorithm mentioned above
- (iii) Covariate discretization: an efficient approach to model matrix storage and computations with the model matrix, using discretized covariates. For example, there are only a finite number of site locations, site labels and elevations, temperature is only recorded to within  $0.1^\circ C$  (or any precision), etc
- These three elements work together, and dropping any one of them leads to an increase in fitting time of an order of magnitude or more. also, mention that this is a new algorithm that is better than QR decomposition. 3 orders of magnitude faster than QR decomposition

## Efficient crossproduct matrix $X^T W X$ computation

- the most expensive part of previous algorithm is the formation of the matrix crossproduct. this approach present a simple, novel and substantially more efficient approach to the computation of this cross product

# R Packages – When to use `bam()`?

- In general, `lme4` is preferred due to easy syntax and robust estimation
- `bam()` is particularly useful for:
  - Complex models that exceed `lme4`'s capabilities
  - Incorporating smooth (nonlinear) terms
  - Large datasets with memory issues
  - Leveraging parallel computing resources

# Multilevel Models for Big Data

## └ R Packages

### └ When to use `bam()`?

#### └ R Packages – When to use `bam()`?

R Packages – When to use `bam()`?

- In general, `lme4` is preferred due to easy syntax and robust estimation
- `bam()` is particularly useful for:
  - Complex models that exceed `lme4`'s capabilities
  - Incorporating smooth (nonlinear) terms
  - Large datasets with memory issues
  - Leveraging parallel computing resources

- In general, `lme4` is preferred for most multilevel datasets due to its straightforward syntax and robust estimation methods.
- `bam()` is particularly useful when:
  - You have complicated structure that begins to bog down `lme4`
  - You want to add smooth terms<sup>1</sup>
  - You have memory issues
  - You have a computing setup that can take advantage of `bam`

# Table of Contents

## ① Introduction

- Multilevel Models
- Issues with Large Datasets

## ② The split-sample approach

- Pseudo Likelihood
- Graphical representation
- Independent subsamples
- Dependent subsamples
- Overlapping subsamples

## ③ R Packages

- lme4 and mgcv
- Why use bam()?
- When to use bam()?

## ④ Summary

## ⑤ References

- Large multilevel datasets pose significant computational challenges
- The split-sample approach offers a practical solution
- R packages like `lme4` and `mgcv` provide robust tools for fitting multilevel models
- Approach and tools selection depends on dataset and research questions

# Multilevel Models for Big Data

## └ Summary

## └ Summary

### Summary

- Large multilevel datasets pose significant computational challenges
- The split-sample approach offers a practical solution
- R packages like `lme4` and `mgcv` provide robust tools for fitting multilevel models
- Approach and tools selection depends on dataset and research questions

- However, large multilevel datasets pose significant computational challenges, including memory constraints and slow estimation times.
- The split-sample approach offers a practical solution by partitioning data into manageable subsamples, enabling efficient model fitting while retaining key statistical properties.
- R packages like `lme4` and `mgcv` provide robust tools for fitting multilevel models, with `bam()` in `mgcv` being particularly suited for very large datasets due to its computational efficiency.
- Choosing the right approach and tools depends on the specific characteristics of the dataset and the research questions at hand.

# Table of Contents

## ① Introduction

- Multilevel Models
- Issues with Large Datasets

## ② The split-sample approach

- Pseudo Likelihood
- Graphical representation
- Independent subsamples
- Dependent subsamples
- Overlapping subsamples







## ③ R Packages

- lme4 and mgcv
- Why use bam()?
- When to use bam()?

## ④ Summary

## ⑤ References

# References I

-  Bates, Douglas et al. (2015). “Fitting Linear Mixed-Effects Models Using lme4”. In: *Journal of Statistical Software* 67.1, pp. 1–48. DOI: 10.18637/jss.v067.i01.
-  Centre for Multilevel Modelling (2025). *What are multilevel models and why should I use them?* University of Bristol. URL: <https://www.bristol.ac.uk/cmm/learning/multilevel-models/what-why.html>.
-  Clark, Michael (2019). *Mixed Models for Big Data*. URL: <https://m-clark.github.io/posts/2019-10-20-big-mixed-models/> (visited on 02/17/2025).
-  Li, Zheyuan and Simon Wood (2020). “Faster model matrix crossproducts for large generalized linear models with discretized covariates”. In: *Statistics and Computing* 30. DOI: 10.1007/s11222-019-09864-2.
-  Speelman, Dirk, Kris Heylen, and Dirk Geeraerts (2018). *Mixed-Effects Regression Models in Linguistics*. Springer International Publishing, pp. 11–28.
-  Wood, Simon, Yannig Goude, and Simon Shaw (2015). “Generalized Additive Models for Large Data Sets”. In: *Journal of the Royal Statistical Society Series C-Applied Statistics* 64, pp. 139–155. DOI: 10.1111/rssc.12068.

# References II



Wood, Simon, Zheyuan Li, et al. (2017). “Generalized Additive Models for Gigadata: Modeling the U.K. Black Smoke Network Daily Data”. In: *Journal of the American Statistical Association* 112, pp. 1–40. DOI: 10.1080/01621459.2016.1195744.



Wood, Simon N. (2011). “Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models”. In: *Journal of the Royal Statistical Society (B)* 73.1, pp. 3–36. DOI: 10.1111/j.1467-9868.2010.00749.x.