Bayesian Hierarchical Modeling using SAS®

Fang Chen
SAS Institute Inc., Cary, NC, USA
fangk.chen@sas.com

BASUG Webinar, September 2022

1. SAS Bayesian Procedures

2. PROC MCMC
   - Non-nested Random-Effects Model
   - The RANDOM Statement

3. PROC BGLIMM
   - Overview
   - Logistic Random-Effects Model
   - Repeated Measurements

Bayesian Procedures in SAS/STAT®

SAS/STAT provides three ways to conduct Bayesian analysis.

Bayesian Procedures in SAS/STAT®

SAS/STAT provides three ways to conduct Bayesian analysis.

- Statement add-on access with the BAYES statement
  - Bayesian estimation to standard analyses available through existing procedures (GENMOD, PHREG, FMM, etc).
  - Suitable for quick and easy access to Bayesian analysis with minimal programming efforts.

Bayesian Procedures in SAS/STAT®

SAS/STAT provides three ways to conduct Bayesian analysis.

- Statement add-on access with the BAYES statement
  - Bayesian estimation to standard analyses available through existing procedures (GENMOD, PHREG, FMM, etc).
  - Suitable for quick and easy access to Bayesian analysis with minimal programming efforts.
- General model exploration with PROC MCMC

Bayesian Procedures in SAS/STAT®

SAS/STAT provides three ways to conduct Bayesian analysis.

- Statement add-on access with the BAYES statement
  - Bayesian estimation to standard analyses available through existing procedures (GENMOD, PHREG, FMM, etc).
  - Suitable for quick and easy access to Bayesian analysis with minimal programming efforts.
- General model exploration with PROC MCMC
- Specialized models
  - the BGLIMM and BCHOICE (discrete choice) procedures, tailored to specific models

PROC MCMC

PROC MCMC is a (mostly) programming-based procedure that fits models with

- arbitrary priors
- arbitrary likelihood function
- hierarchical structure

PROC MCMC

PROC MCMC is a (mostly) programming-based procedure that fits models with

- arbitrary priors
- arbitrary likelihood function
- hierarchical structure

You specify the model and the procedure deploys sampling techniques to generate posterior samples from a target (posterior) distribution.

PROC BGLIMM

PROC BGLIMM is the Bayesian cousin to the GLIMMIX procedure, and fits both linear and generalized linear mixed-effects models.

PROC BGLIMM

PROC BGLIMM is the Bayesian cousin to the GLIMMIX procedure, and fits both linear and generalized linear mixed-effects models.

A mixed model contains fixed and random effects. In a linear case:

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{R}) \end{aligned}$$

where $\boldsymbol{\beta}$ is fixed effects and $\boldsymbol{\gamma}$ is random effects.

PROC BGLIMM

PROC BGLIMM is the Bayesian cousin to the GLIMMIX procedure, and fits both linear and generalized linear mixed-effects models.

A mixed model contains fixed and random effects. In a linear case:

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{R}) \end{aligned}$$

where $\boldsymbol{\beta}$ is fixed effects and $\boldsymbol{\gamma}$ is random effects.

- G-side matrix, $\mathbf{G}$, is the covariance matrix of the random effects.
- R-side matrix, $\mathbf{R}$, is the covariance matrix of the residuals.

PROC BGLIMM

A generalized linear mixed model (GLMM) consists of the following:

- a linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- a link function

$$\mathrm{E}[Y|\boldsymbol{\beta}, \boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

- a response distribution in the exponential family (binary, binomial, Poisson, normal, gamma, negative binomial, ...).

PROC BGLIMM

A generalized linear mixed model (GLMM) consists of the following:

- a linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- a link function

$$\mathrm{E}[Y|\boldsymbol{\beta}, \boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

- a response distribution in the exponential family (binary, binomial, Poisson, normal, gamma, negative binomial, ...).

PROC BGLIMM samples from the joint posterior distribution of $p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{R}, \mathbf{G}|\mathbf{Y})$.

Similar Essential Statements

- MODEL: Y, X, dist, and link function
- RANDOM: random effects (Z), the G-side cov (TYPE=)
- REPEATED: the R-side cov (TYPE=)
- CLASS
- ESTIMATE: linear combination of parameters
- FREQ: frequency

Features

Additional features of PROC BGLIMM include:

- suite of covariance structures (for both G- and R-side)

Features

Additional features of PROC BGLIMM include:

- suite of covariance structures (for both G- and R-side)
- covariance heterogeneity modeling

Features

Additional features of PROC BGLIMM include:

- suite of covariance structures (for both G- and R-side)
- covariance heterogeneity modeling
- suite of priors

Features

Additional features of PROC BGLIMM include:

- suite of covariance structures (for both G- and R-side)
- covariance heterogeneity modeling
- suite of priors
- model comparison via DIC

Features

Additional features of PROC BGLIMM include:

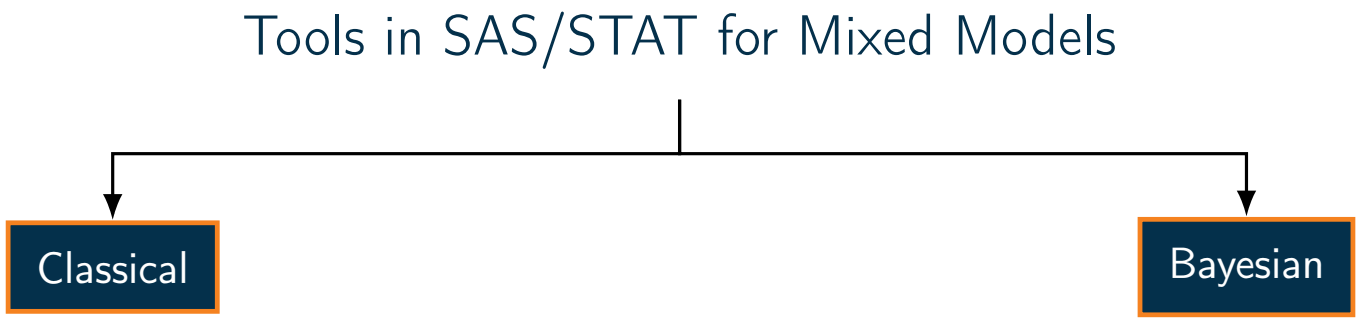
- suite of covariance structures (for both G- and R-side)
- covariance heterogeneity modeling
- suite of priors
- model comparison via DIC
- ability to handle missing data: MCAR and MAR

Features

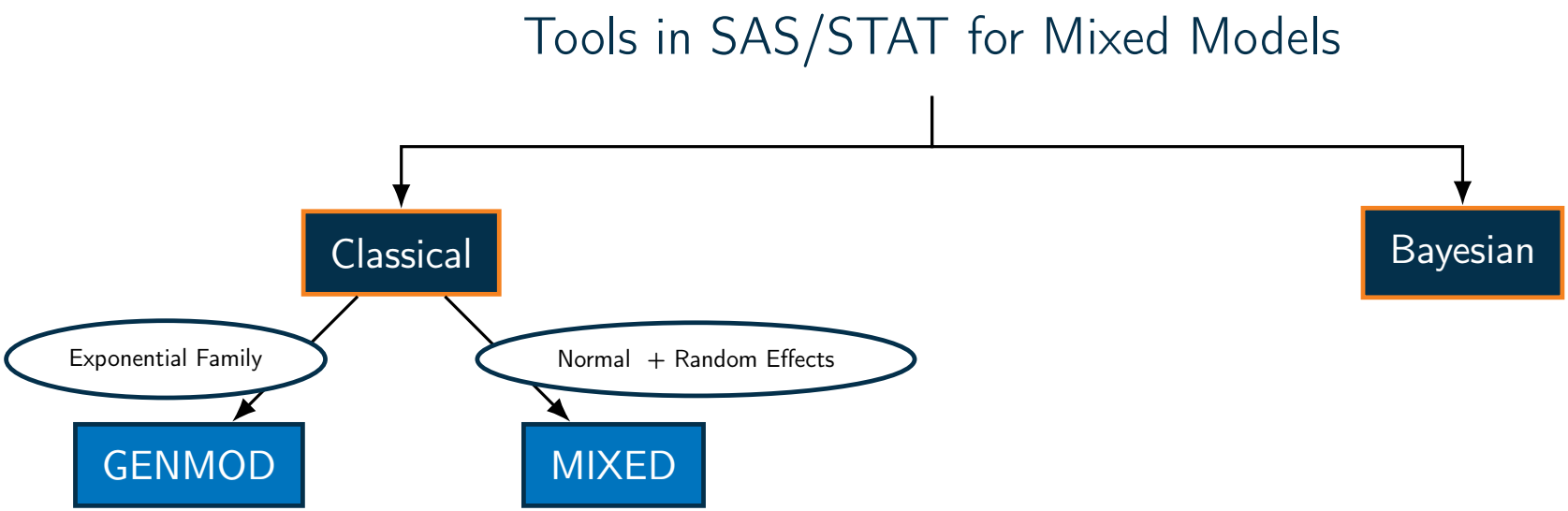
Additional features of PROC BGLIMM include:

- suite of covariance structures (for both G- and R-side)
- covariance heterogeneity modeling
- suite of priors
- model comparison via DIC
- ability to handle missing data: MCAR and MAR
- parallel computing combined with optimal sampling

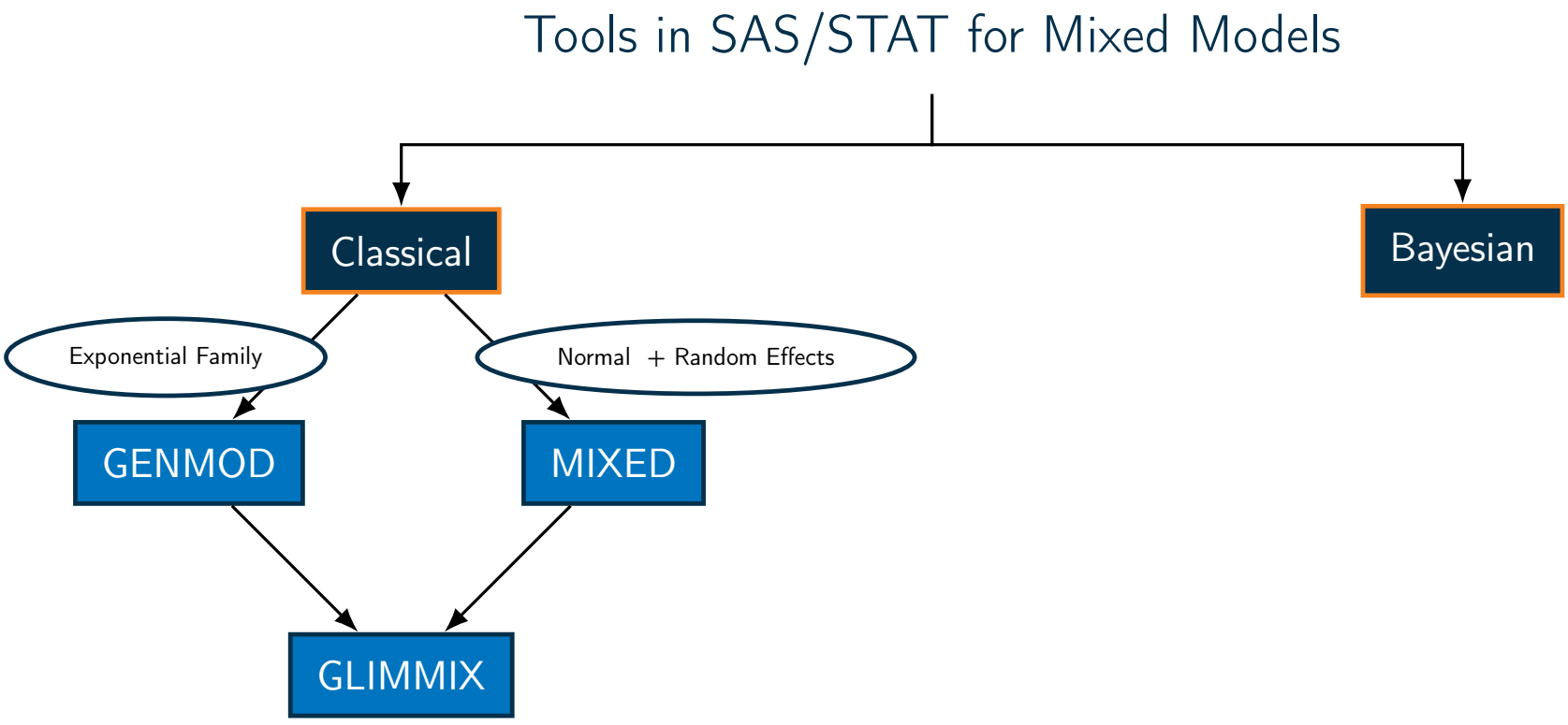
Mixed Modeling Procedures in SAS/STAT®



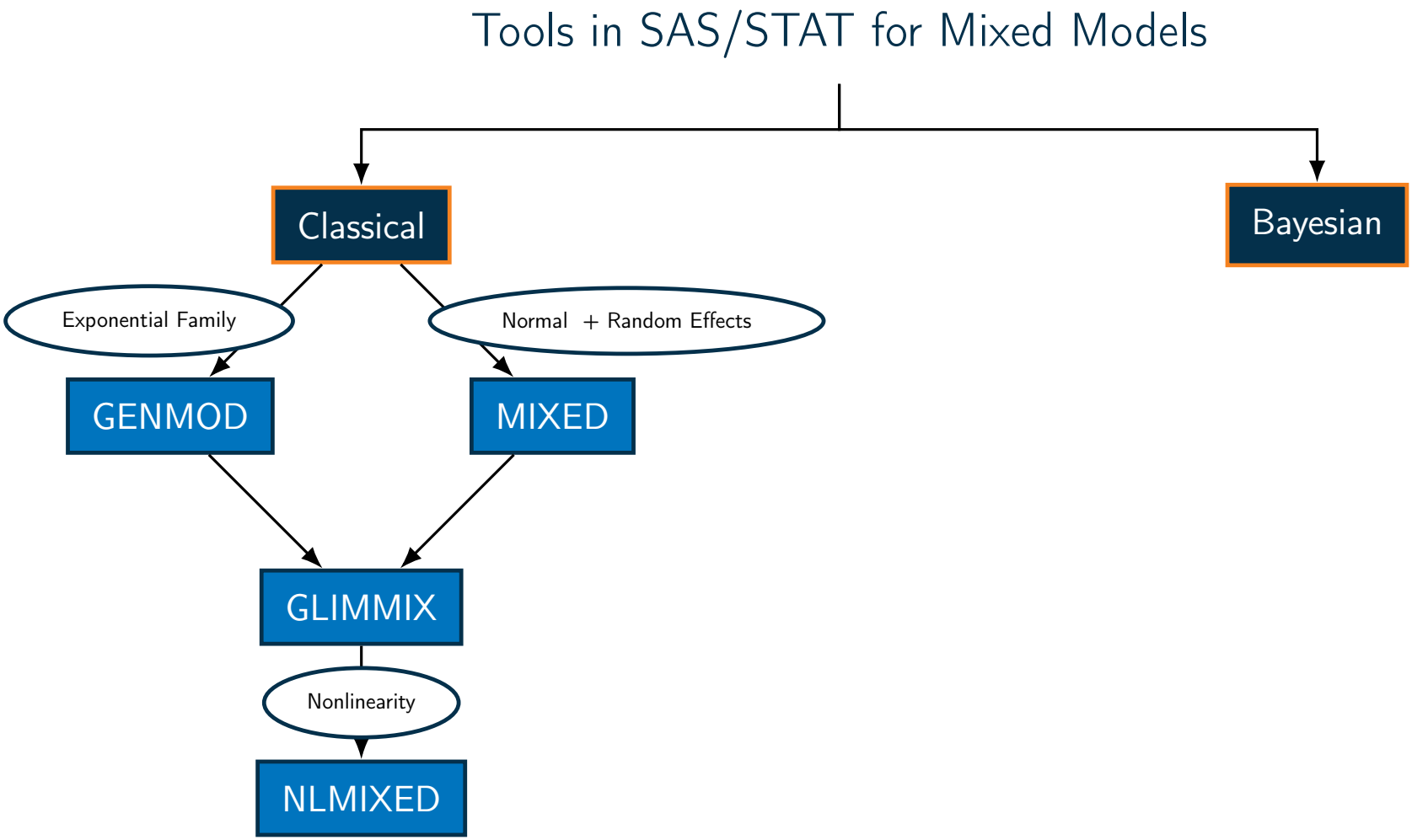
Mixed Modeling Procedures in SAS/STAT®



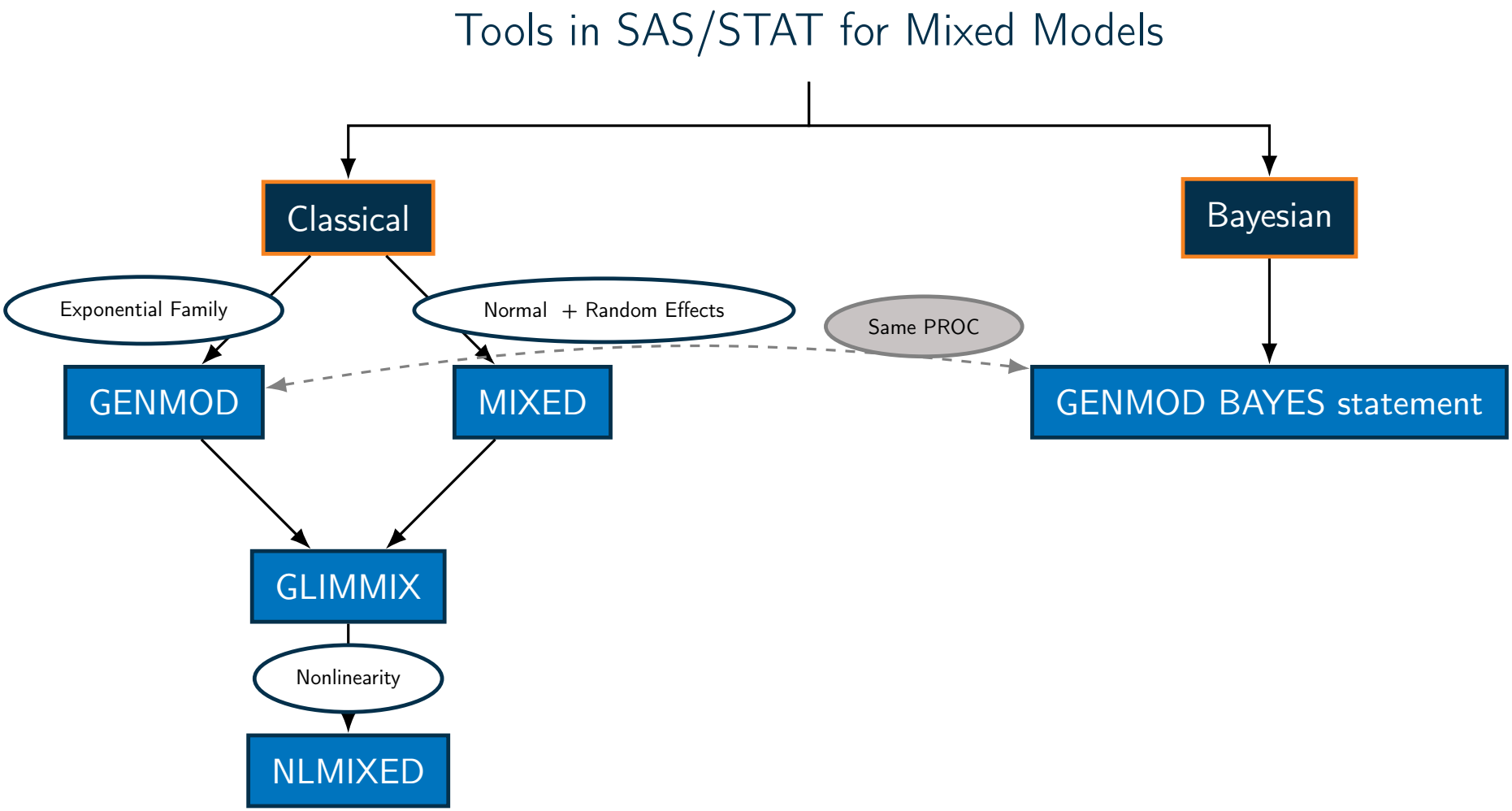
Mixed Modeling Procedures in SAS/STAT®



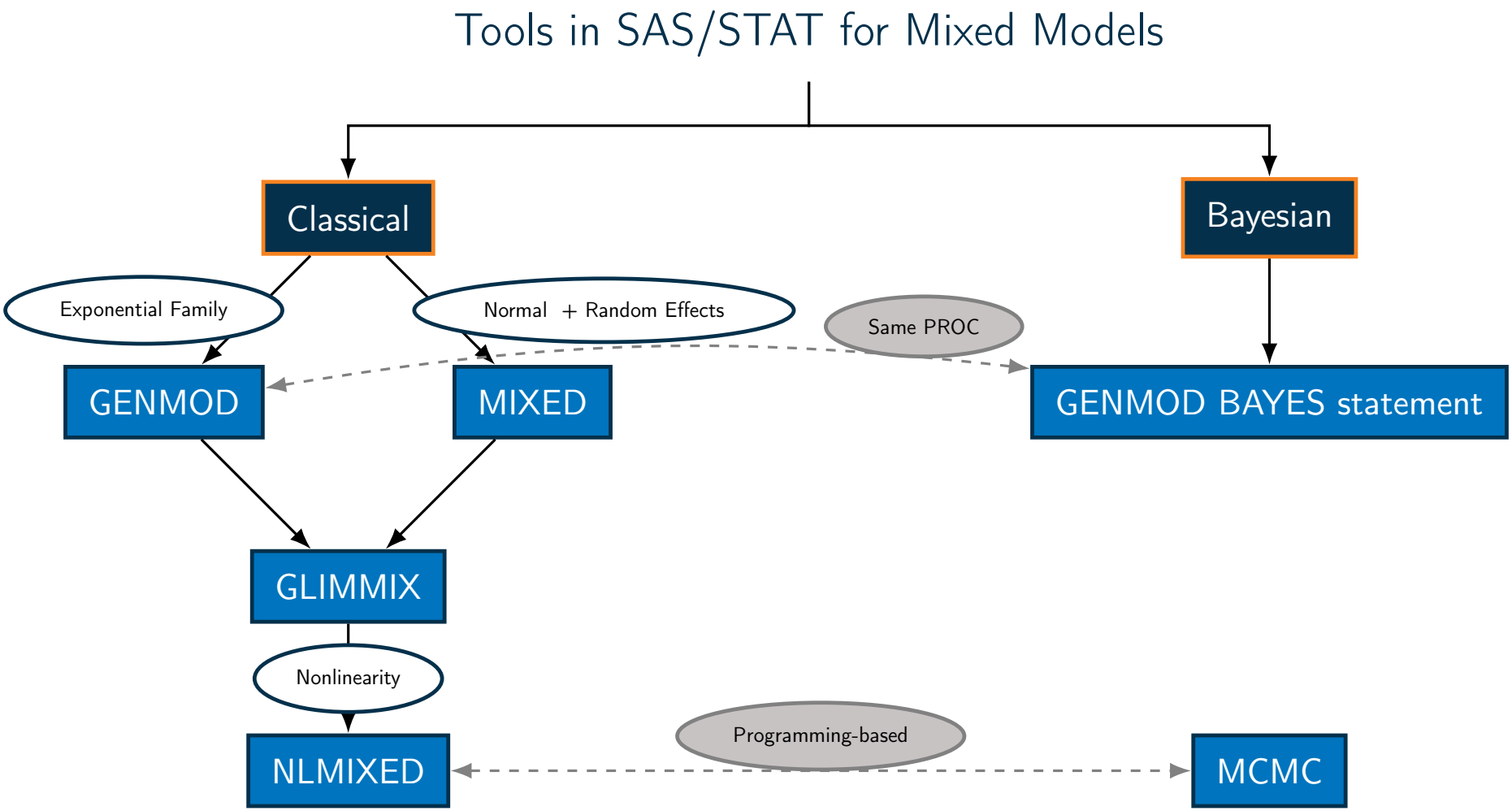
Mixed Modeling Procedures in SAS/STAT®



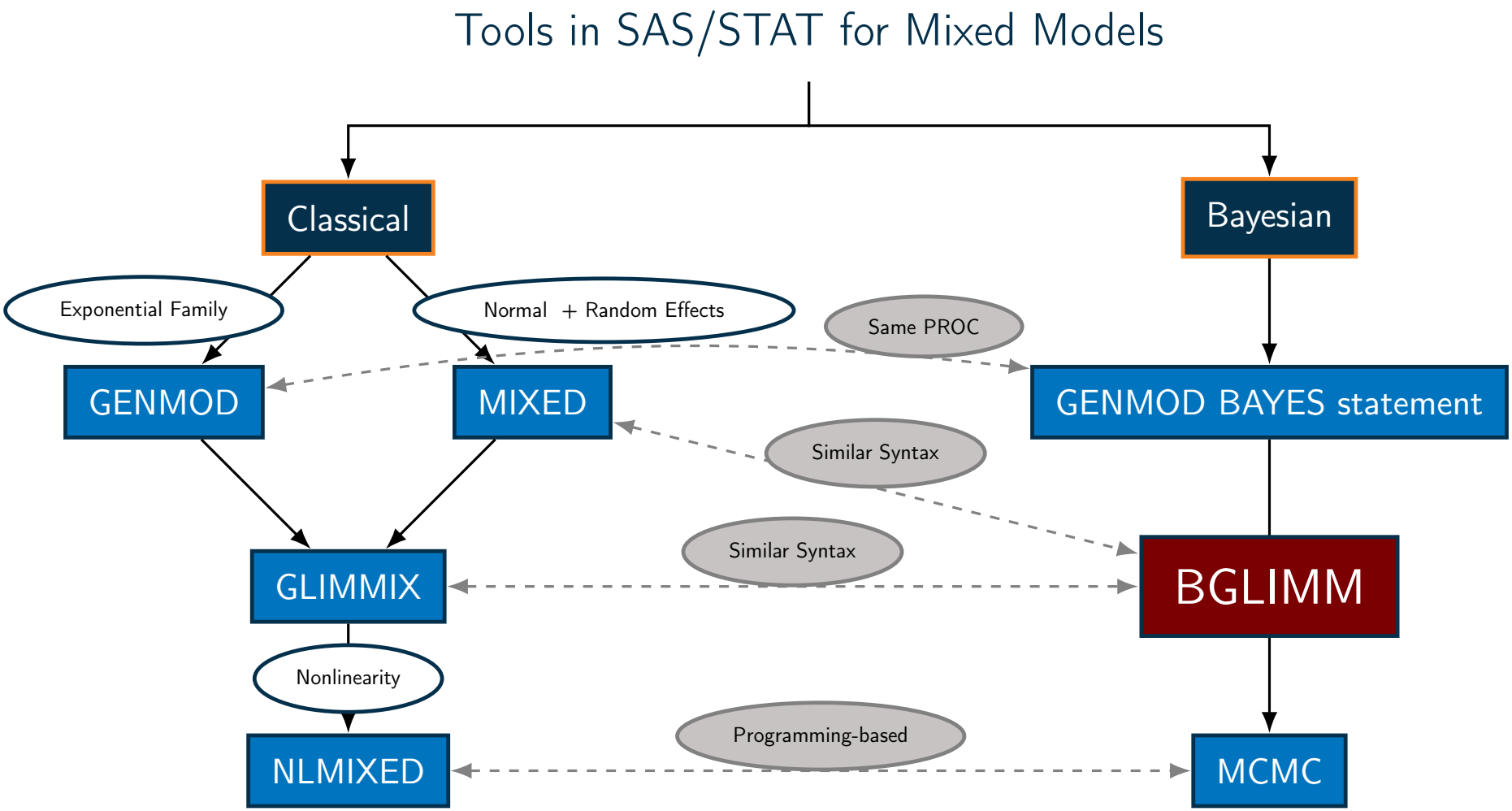
Mixed Modeling Procedures in SAS/STAT®



Mixed Modeling Procedures in SAS/STAT®



Mixed Modeling Procedures in SAS/STAT®



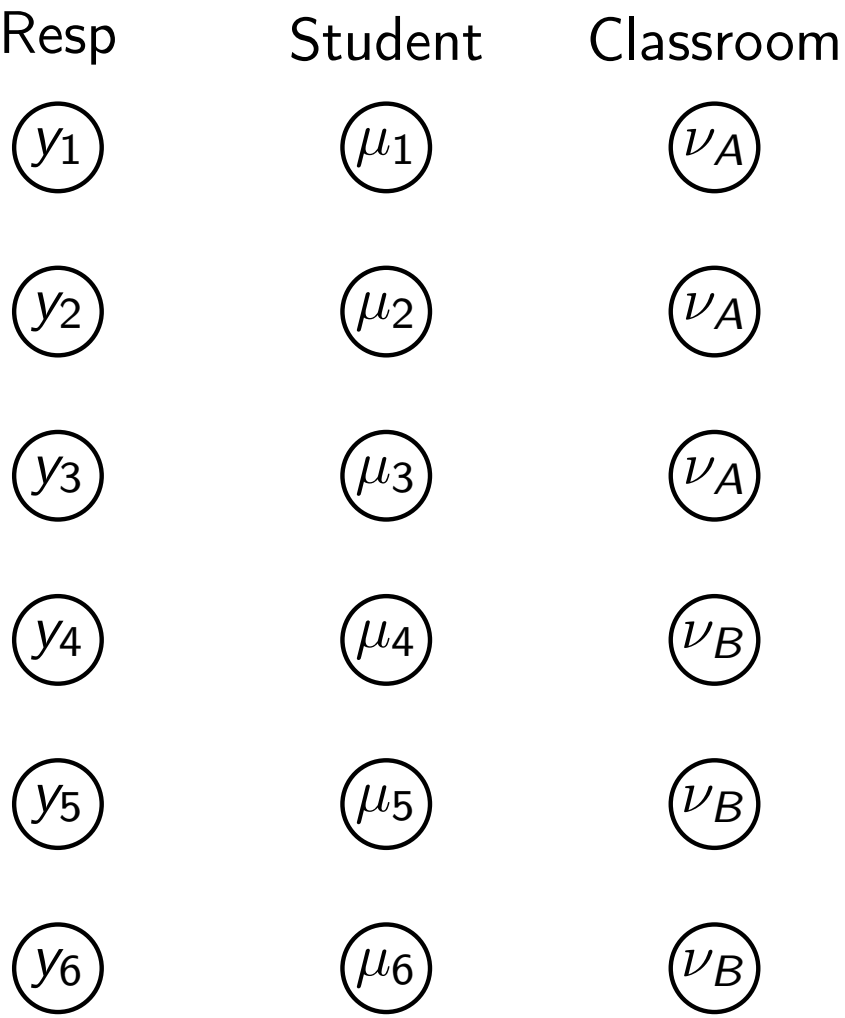
Outline

To Fit a RE Model, You Want to Start with the Data

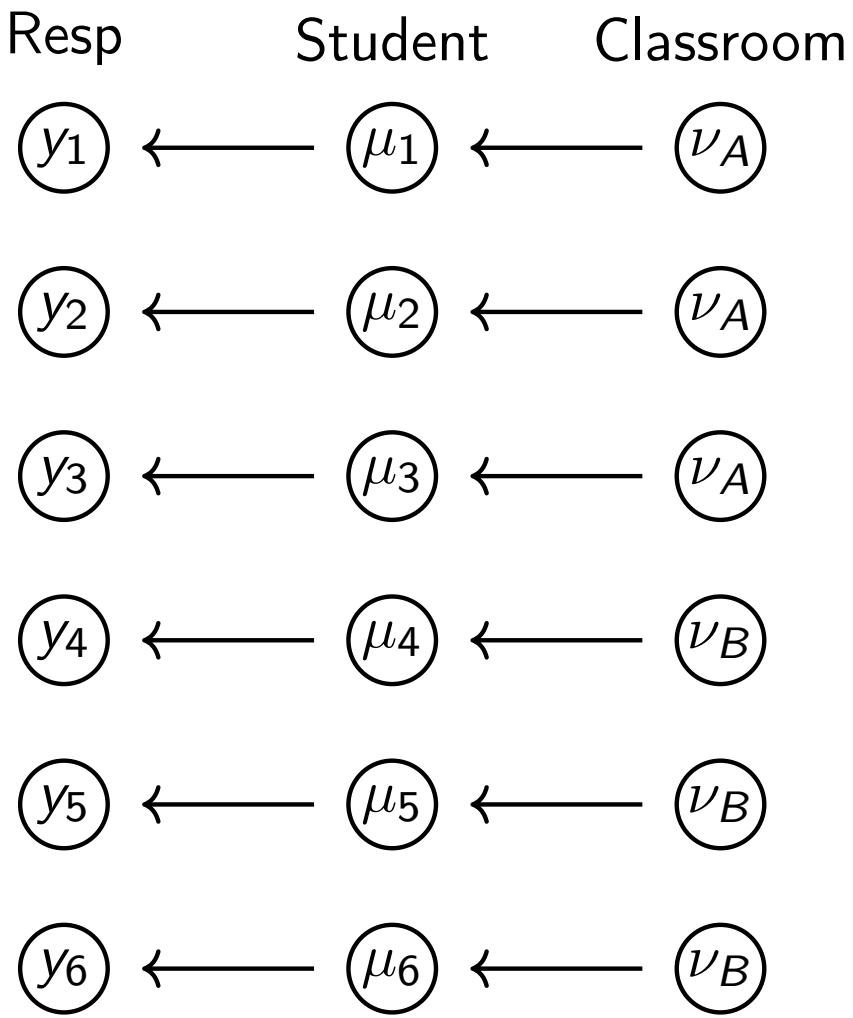
Resp	Student	Classroom
y_1	Bella C.	A
y_2	Alexa N.	A
y_3	Olive T.	A
y_4	Ian C.	B
y_5	Ben C.	B
y_6	Ava B.	B

Suppose that you want to fit a model with two random effects to these data: one student-level (μ_i) and one classroom-level (ν_j).

Underlying Representation of Random-Effects Parameters

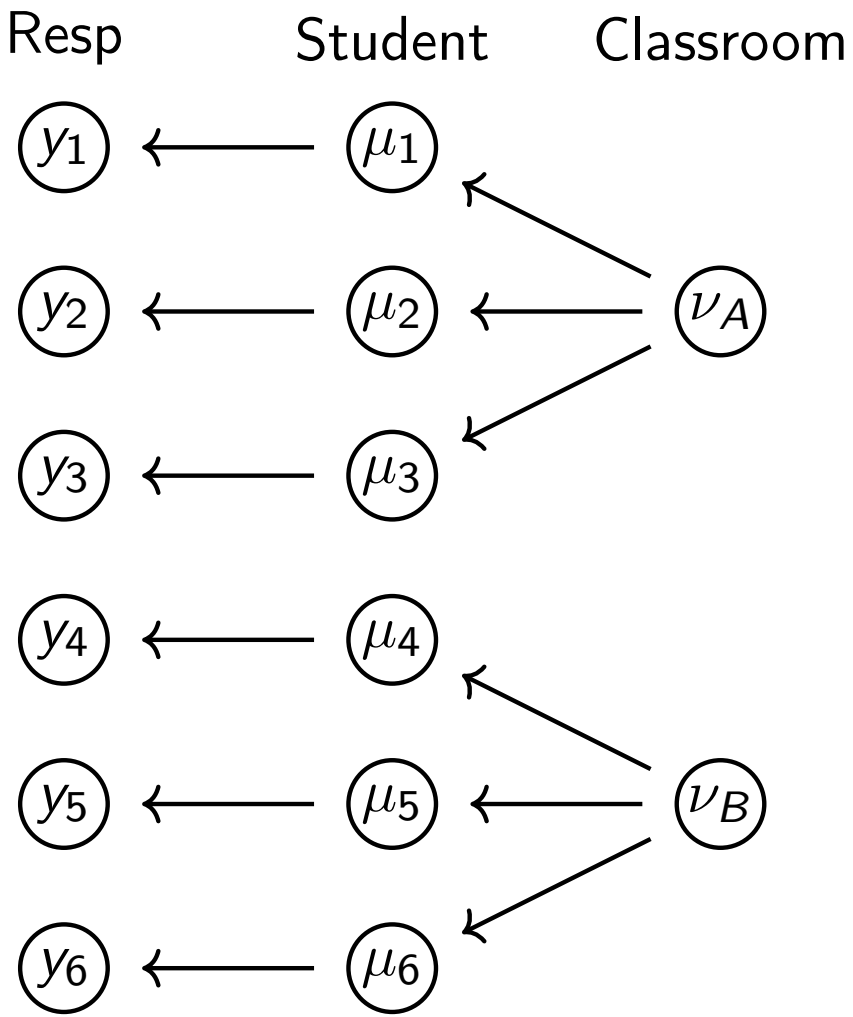


Frequently Used Conditional Independence Assumption

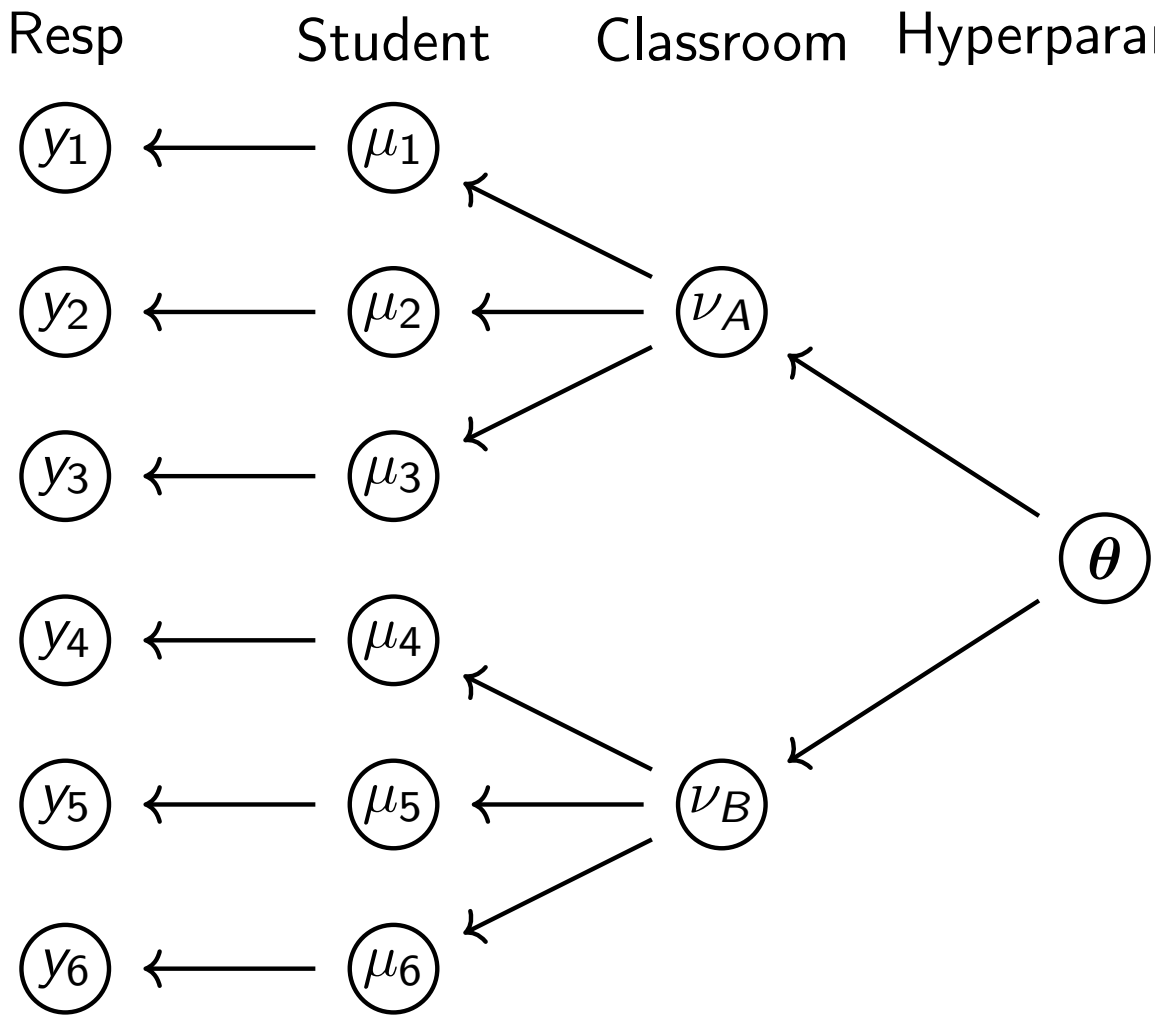


Conditional independence means random variables within each level are independent given parameters one level up.

Equivalent Representation



Multilevel Hierarchical Model



This is the class of multilevel hierarchical models:

$$\pi(\mathbf{y}|\boldsymbol{\mu}) \cdot \pi(\boldsymbol{\mu}|\boldsymbol{\nu}) \cdot \pi(\boldsymbol{\nu}|\boldsymbol{\theta})$$

Conditional Independence Assumption

This type of data occurs frequently in practice:

- By default, the MODEL statement assumes the responses are independent given the random effects (and other parameters in the model):

$$y_i \perp y_j | \{\mu\}$$

for $i \neq j$

Conditional Independence Assumption

This type of data occurs frequently in practice:

- By default, the MODEL statement assumes the responses are independent given the random effects (and other parameters in the model):

$$y_i \perp y_j | \{\mu\}$$

for $i \neq j$

- By default, the RANDOM statement assumes that the random-effects parameters within a group are independent given their hyperparameters:

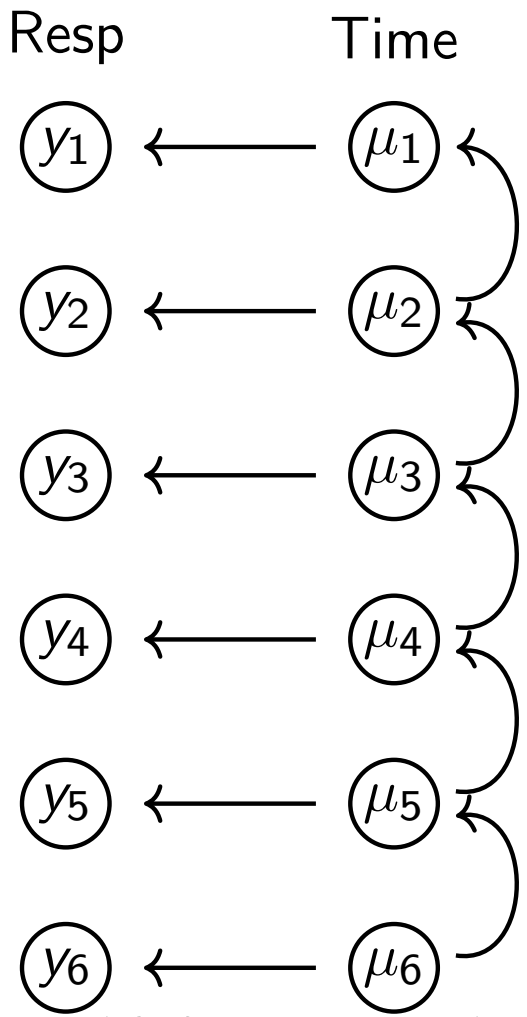
$$\mu_i \perp \mu_j | \{\nu\}$$

$$\nu_k \perp \nu_l | \theta$$

where θ are the hyperparameters.

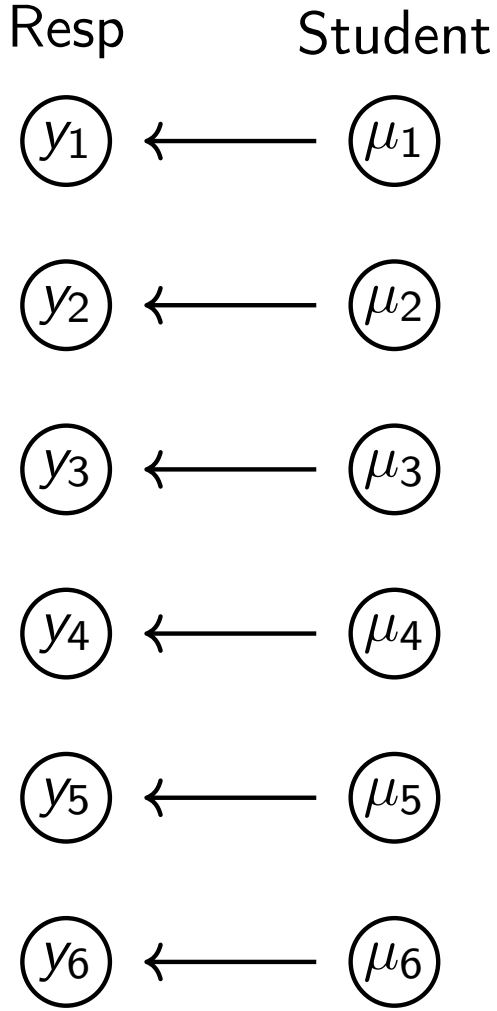
More Flexible Assumption

The conditional independence assumption might not apply in certain scenarios. For example, a model can require dependence among the μ's:

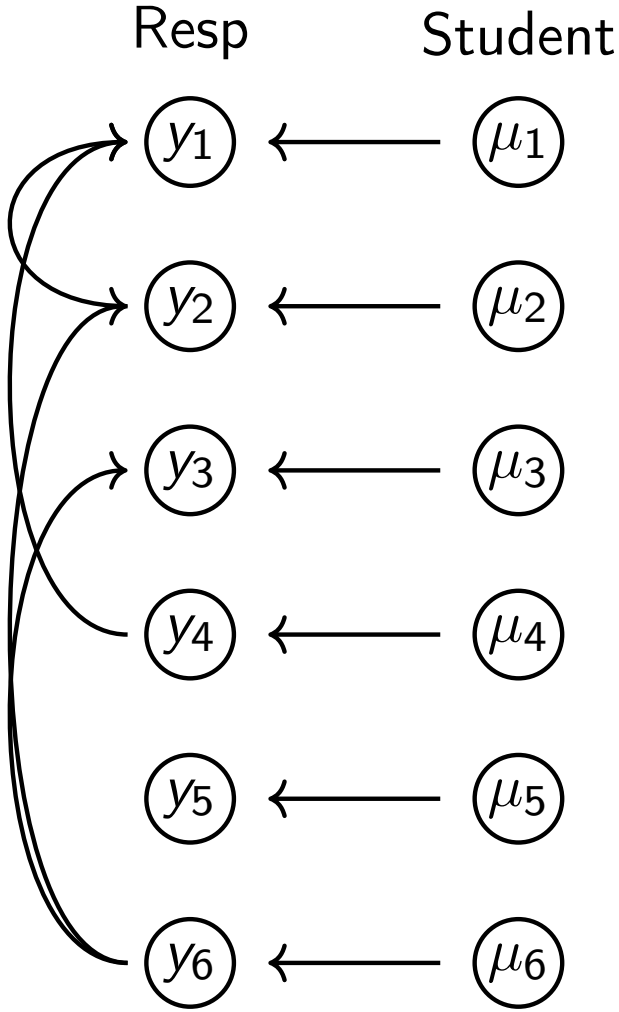


$y_i \perp y_j | \{\mu\}$ but $\mu_i | \mu_{i-1}$

Dependence Can Occur Everywhere

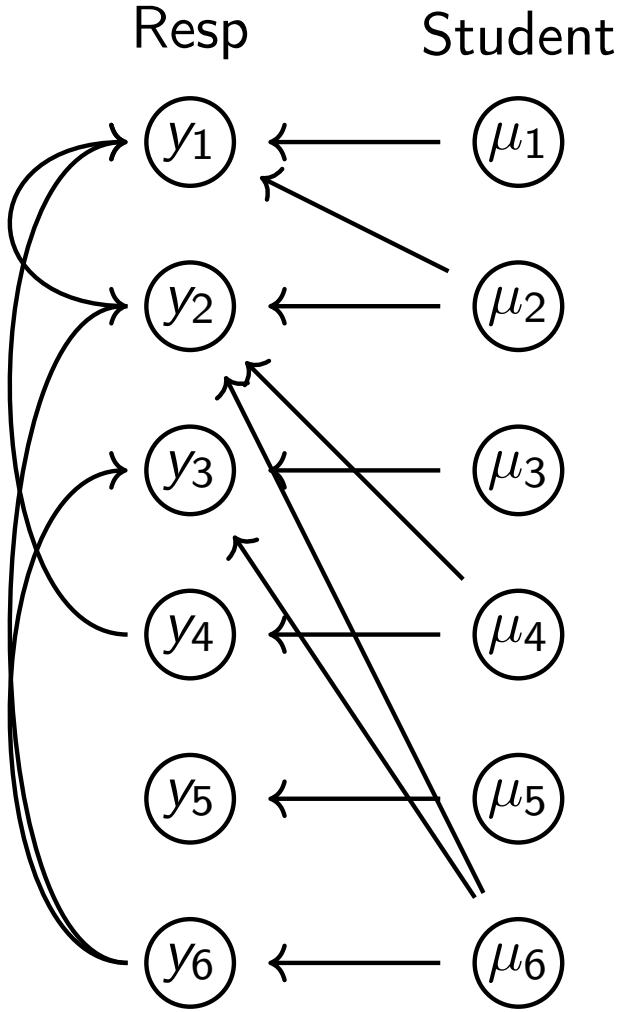


Dependence Can Occur Everywhere



$y_i|y_{-i}$, where y_i depends on other y_j

Dependence Can Occur Everywhere



$y_i|y_{-i}$, where y_i depends on other y_j

$y_i|\mu_{-i}, y_i$, where y_i can also depend on other random effects

Outline

Simulation Study

Milliren, Carly et al. (2018) "Does an uneven sample size distribution across settings matter in cross-classified multilevel modeling? Results of a simulation study"

- Non-nested hierarchical models
- Sample size considerations are unknown for these models.
- Objective is to assess small sample impact on random effect estimation.

Simulated Data

bmi	hood_id	school_id	kid_id
18.3436	5	87	5087001
21.1745	5	87	5087002
28.5420	5	87	5087003
22.1695	5	87	5087004
25.1573	5	87	5087005
27.3970	5	87	5087006

...

- sample size: 20,085
- kid ID: 19,281
- hood ID: 2,410
- school ID: 132

There is not a nested structure in the data. Kids from different neighborhood can go to different schools.

Model

Normal random-effects model with three random effects (kids, school, and hood):

```
proc glimmix  data=sim6;
   class hood_id school_id kid_id;
   model bmi= /solution ddfm=kr dist=normal link=identity;
   random int /subject=kid_id;
   random int /subject=school_id;
   random int /subject=hood_id;
run;
```

This doesn't work in GLIMMIX.

Model

Normal random-effects model with three random effects (kids, school, and hood):

```
proc glimmix  data=sim6;
   class hood_id school_id kid_id;
   model bmi= /solution ddfm=kr dist=normal link=identity;
   random int /subject=kid_id;
   random int /subject=school_id;
   random int /subject=hood_id;
run;
```

This doesn't work in GLIMMIX. If we are to reduce the model to a two random effects (school and hood), it takes about 20 minutes to run in GLIMMIX.

Speed

The challenge here, for PROC GLIMMIX, is that the subjects are not nested. As a result, PROC GLIMMIX treats the entire data set as one subject.

PROC MIXED handles it faster, but faces the same challenge.

Speed

The challenge here, for PROC GLIMMIX, is that the subjects are not nested. As a result, PROC GLIMMIX treats the entire data set as one subject.

PROC MIXED handles it faster, but faces the same challenge.

One solution could be PROC HPMIXED, which handles sparse design matrix better.

Speed

The challenge here, for PROC GLIMMIX, is that the subjects are not nested. As a result, PROC GLIMMIX treats the entire data set as one subject.

PROC MIXED handles it faster, but faces the same challenge.

One solution could be PROC HPMIXED, which handles sparse design matrix better.

Or you can use PROC MCMC.

PROC MCMC

```
proc mcmc data=sim6 nthreads=8 nmc=5000 plots=none;
   parm s2s s2h s2k s2;
   parm beta0;
   prior beta0 ~ n(0, sd=1000);
   prior s2: ~ igamma(shape=2, scale=2);
   random int_h ~ n(0, var=s2h) subject=hood_id;
   random int_s ~ n(0, var=s2s) subject=school_id;
   random int_k ~ n(0, var=s2k) subject=kid_id;
   mu = beta0 + int_s + int_h + int_k;
   model bmi ~ n(mu, var=s2);
   run;
```

Runtime is about 2 to 3 minutes.

PROC MCMC

```
proc mcmc data=sim6 nthreads=8 nmc=5000 plots=none;
   parm s2s s2h s2k s2;
   parm beta0;
   prior beta0 ~ n(0, sd=1000);
   prior s2: ~ igamma(shape=2, scale=2);
   random int_h ~ n(0, var=s2h) subject=hood_id;
   random int_s ~ n(0, var=s2s) subject=school_id;
   random int_k ~ n(0, var=s2k) subject=kid_id;
   mu = beta0 + int_s + int_h + int_k;
   model bmi ~ n(mu, var=s2);
   run;
```

Runtime is about 2 to 3 minutes.

In addition, you can use PROC MCMC to model more complex random effects structure.

Spatial Prior

RANDOM statement supports a conditional autoregressive Gaussian prior, for random effects that are spatially correlated:

$$\theta_i|\theta_{-i} \sim \text{normal}\left(\sum_{j\in N(i)} \theta_j/m_i, s^2\right)$$

- $N(i)$ is the set of neighbors of area i
- m_i is the total number of neighbors of area i
- s is the standard deviation of the normal distribution.

Spatial Prior

Suppose that now you have information on neighboring hoods (county) information to each observation in the input data set:

Let `county` by the county identifier, and the adjacency information is provided:

```
 county num     ID1        ID2        ID3       ID4      ID5     ID6     I
  Wake    6   Johnston   Franklin   Granville Durham Chatham Harnett
 Durham   5     Wake     Granville    Person   Orange Chatham    .
...
```

- The `ID` variables are the neighboring county names.

Spatial Prior

Suppose that now you have information on neighboring hoods (county) information to each observation in the input data set:

Let `county` by the county identifier, and the adjacency information is provided:

```
 county num     ID1        ID2        ID3       ID4      ID5     ID6     I
  Wake    6   Johnston   Franklin   Granville Durham Chatham Harnett
 Durham   5     Wake     Granville    Person   Orange Chatham    .
...
```

- The `ID` variables are the neighboring county names.

```
random a ~ normalcar(neighbors=ID, num=num, var=s2a)
   subject=county;
```

Outline

RANDOM Statement

In PROC MCMC, you use the RANDOM statement to specify the random-effects parameters.

RANDOM *random-effect* $\sim$ distribution SUBJECT= ;

random-effect : defines the effect

distribution : specifies its prior distribution: normal, beta, binary, gamma, mvn, general, dgeneral, table, etc.

SUBJECT= : identifies group membership

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.
2. PROC MCMC fills in the effect symbol with RE parameters as it steps through the data set:

```
random gamma ~ normal(0, var=s2g) subject=patient;
mu = b0 + gamma;
```

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.
2. PROC MCMC fills in the effect symbol with RE parameters as it steps through the data set:

```
random gamma ~ normal(0, var=s2g) subject=patient;
mu = b0 + gamma;
```

Patient
201
202
203
201
203

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.
2. PROC MCMC fills in the effect symbol with RE parameters as it steps through the data set:

```
random gamma ~ normal(0, var=s2g) subject=patient;
mu = b0 + gamma;
```

Patient	RE parameter
201	γ_{201}
202	γ_{202}
203	γ_{203}
201	γ_{201}
203	γ_{203}

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.
2. PROC MCMC fills in the effect symbol with RE parameters as it steps through the data set:

```
random gamma ~ normal(0, var=s2g) subject=patient;
mu = b0 + gamma;
```

Patient	RE parameter	Processed
201	γ_{201}	$\mu = \beta_0 + \gamma_{201}$
202	γ_{202}	$\mu = \beta_0 + \gamma_{202}$
203	γ_{203}	$\mu = \beta_0 + \gamma_{203}$
201	γ_{201}	$\mu = \beta_0 + \gamma_{201}$
203	γ_{203}	$\mu = \beta_0 + \gamma_{203}$

Three Things to Remember about the RANDOM Statement

1. The number of RE parameters = the number of **unique values** in the SUBJECT= variable.
2. PROC MCMC fills in the effect symbol with RE parameters as it steps through the data set:

```
random gamma ~ normal(0, var=s2g) subject=patient;
mu = b0 + gamma;
```

Patient	RE parameter	Processed
201	γ_{201}	$\mu = \beta_0 + \gamma_{201}$
202	γ_{202}	$\mu = \beta_0 + \gamma_{202}$
203	γ_{203}	$\mu = \beta_0 + \gamma_{203}$
201	γ_{201}	$\mu = \beta_0 + \gamma_{201}$
203	γ_{203}	$\mu = \beta_0 + \gamma_{203}$

3. You must add the random effect to the model.

The SUBJECT= Variable

This is a data set variable that indicates clustering of the random effects.

- The number of R.E. parameters is determined by the number of unique values in the SUBJECT= variable.
- The SUBJECT= variable be numeric or character, and doesn't have to be sorted

1	27513	07/13/1995	John
1	27513	01/31/2003	Mary
2	01440	10/12/1997	Ken
2	17923	08/03/2010	John
...	...	...	...

More on Random Effects

A program can have multiple RANDOM statements:

- one for the classroom-level
- one for the school-level
- ...

```
random a ~ normal(0, var=s2a) subject=school;
random b ~ normal(0, var=s2b) subject=class;
mu = ... + a + b;
```

More on Random Effects

You can fit nested or nonnested models:

- nested models: levels of one factor must cluster within the levels of another factor.
  - the classroom effect can be the hyperparameters to the student effects

```
random a ~ normal(0, var=s2a) subject=school;
random b ~ normal(a, var=s2b) subject=class;
```

More on Random Effects

You can fit nested or nonnested models:

- nested models: levels of one factor must cluster within the levels of another factor.
  - the classroom effect can be the hyperparameters to the student effects

```
random a ~ normal(0, var=s2a) subject=school;
random b ~ normal(a, var=s2b) subject=class;
```

- nonnested models: levels of factors can cross;

```
random a ~ normal(0, var=s2a) subject=student;
random b ~ normal(0, var=s2b) subject=age;
```

More on Random Effects

The RANDOM statement supports different prior distributions:

```
random z ~ binary(p) subject=patient;
random b ~ gamma(shape=g1, scale=g2) subject=class;
```

More on Random Effects

The RANDOM statement supports different prior distributions:

```
random z ~ binary(p) subject=patient;
random b ~ gamma(shape=g1, scale=g2) subject=class;
```

Including multivariate, categorical, or general prior:

```
array w[2];
random w ~ MVN(mu0, Sigma) subject=test;
random u ~ table(p) subject=obs;
random v ~ general(lreff) subject=location;
```

Outline

GLMM Models

PROC BGLIMM fits the class of generalized linear mixed model (GLMM):

- a linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- a link function

$$\mathrm{E}[Y|\boldsymbol{\beta}, \boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

- a response distribution in the exponential family (binary, binomial, Poisson, normal, gamma, negative binomial, ...).

Parameters and Priors

To complete the specification of a Bayesian GLIMM, you need prior on the following parameters:

$$
\begin{aligned}
\boldsymbol{\beta} &\sim \pi(\boldsymbol{\beta}) \\
\boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\
\mathbf{G} &\sim \pi(\mathbf{G}) \\
\mathbf{R} &\sim \pi(\mathbf{R})
\end{aligned}
$$

Parameters and Priors

To complete the specification of a Bayesian GLIMM, you need prior on the following parameters:

$$\begin{aligned} \boldsymbol{\beta} &\sim \pi(\boldsymbol{\beta}) \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \mathbf{G} &\sim \pi(\mathbf{G}) \\ \mathbf{R} &\sim \pi(\mathbf{R}) \end{aligned}$$

PROC BGLIMM samples from the joint posterior distribution of $p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{R}, \mathbf{G}|\mathbf{Y})$.

PROC BGLIMM Simplifies Model Specification

Normal regression:

```
proc bglimm data=input seed=985329;
   class A B;
   model y = A|B;
run;
```

$$
\begin{aligned}
y_i =& \beta_0 + \beta_a A_i + \beta_b B_i + \beta_{11} A_{1i} B_{1i} \\
&+ \ldots + \beta_{54} A_{5i} B_{4i} + \epsilon_i \\
\epsilon_i \sim& N(0, \sigma^2) \\
\pi(\boldsymbol{\beta}) \propto& 1, \ \pi(\sigma^2) = \text{igamma}(2, 2)
\end{aligned}
$$

PROC BGLIMM Simplifies Model Specification

Normal regression:

```
proc bglimm data=input seed=985329;
   class A B ID;
   model y = A|B / scaleprior=igamma(0.01, 0.01);
   random int / subject=ID covprior=halfCauchy(2);
run;
```

$$
\begin{aligned}
y_i =& \beta_0 + \beta_a A_i + \beta_b B_i + \beta_{11} A_{1i} B_{1i} \\
&+ \ldots + \beta_{54} A_{5i} B_{4i} + \gamma_j + \epsilon_i \\
\epsilon_i \sim& N(0, \sigma^2) \\
\pi(\boldsymbol{\beta}) \propto& 1, \ \pi(\sigma^2) = \text{igamma}(0.01, 0.01) \\
\gamma_j \sim& N(0, \sigma_\gamma^2), \ \pi(\sigma_\gamma^2) = \text{half-Cauchy}(2)
\end{aligned}
$$

PROC BGLIMM Simplifies Model Specification

Normal regression:

```
proc bglimm data=input seed=985329;
   class A B ID D;
   model y = A|B;
   random D / subject=ID type=AR(1);
run;
```

$$y_i = \beta_0 + \beta_a A_i + \beta_b B_i + \beta_{11} A_{1i} B_{1i} + \ldots + \beta_{54} A_{5i} B_{4i} + \alpha_{1j} D_{1i} + \alpha_{2j} D_{2i} \cdots \alpha_{Kj} D_{Ki} + \epsilon_i$$

$$\epsilon_i \sim N(0, \sigma^2)$$

$$\pi(\boldsymbol{\beta}) \propto 1, \ \pi(\sigma^2) = \text{igamma}(2, 2)$$

$$\begin{pmatrix} \alpha_{1j} \\ \alpha_{2j} \\ \vdots \\ \alpha_{Kj} \end{pmatrix} \sim MVN \left[\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \mathbf{G} = \sigma^2_\alpha \begin{bmatrix} 1 & \rho & \ldots & \rho^{K-1} \\ \rho & 1 & \ldots & \rho^{K-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{K-1} & \rho^{K-2} & \ldots & 1 \end{bmatrix} \right]$$

PROC BGLIMM Simplifies Model Specification

Normal regression:

```
proc bglimm data=input seed=985329;
   class A B ID D Person Hour;
   model y = A|B;
   random D / subject=ID type=AR(1);
   repeated Hour/ type=un subject=Person;
run;
```

$$Y_i = \mathbf{X}_i\beta + \mathbf{Z}_i\gamma + \epsilon_i$$

$$Y_i \text{ is a } H \times 1 \text{ vector}$$

$$\epsilon_i = \begin{pmatrix} \epsilon_{i1} \\ \vdots \\ \epsilon_{iH} \end{pmatrix} \sim MVN\left[\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}_H, \mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1H} \\ R_{21} & R_{22} & \dots & R_{2H} \\ \vdots & \vdots & \ddots & \vdots \\ R_{H1} & R_{H2} & \dots & R_{HH} \end{bmatrix}\right]$$

$$\pi(\mathbf{R}) = \text{iwishart}(H+3, (H+3)\cdot\mathbf{I})$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10);
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \log() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I})
\end{aligned}
$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10)
         offset=LogN link=loglog;
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i + \text{LogN}_i \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \text{loglog}() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I})
\end{aligned}
$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Analyst Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10)
         offset=LogN;
   random int / subject=Analyst;
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i + \text{LogN}_i + \gamma_j \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \log() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I}) \\
\gamma_j &\sim N(0, \sigma^2_\gamma), \ \pi(\sigma^2_\gamma) = \text{igamma}(2, 2)
\end{aligned}
$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Analyst Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10)
         offset=LogN;
   random int / subject=Analyst group=Gender;
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i + \text{LogN}_i + \gamma_{Fj} + \gamma_{Mj} \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \log() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I}) \\
\gamma_{Fj} &\sim N(0, \sigma_F^2), \ \pi(\sigma_F^2) = \text{igamma}(2, 2) \\
\gamma_{Mj} &\sim N(0, \sigma_M^2), \ \pi(\sigma_M^2) = \text{igamma}(2, 2)
\end{aligned}
$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Analyst Run Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10)
         offset=LogN;
   random int / subject=Analyst;
   random int / subject=Run(Analyst);
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i + \text{LogN}_i + \gamma_j + \alpha_k \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \log() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I}) \\
\gamma_j &\sim N(0, \sigma^2_\gamma), \ \pi(\sigma^2_\gamma) = \text{igamma}(2, 2) \\
\alpha_k &\sim N(0, \sigma^2_\alpha), \ \pi(\sigma^2_\alpha) = \text{igamma}(2, 2)
\end{aligned}
$$

Extension to GLMMs

Poisson regression:

```
proc bglimm data=input seed=463766;
   class Analyst Run Gender(ref="F");
   model Observed=Gender Conc / dist=poisson cprior=normal(var=10)
         offset=LogN;
   random int / subject=Analyst covprior=unif(lower=0, upper=10);
   random int / subject=Run(Analyst) covprior=normal(var=5);
run;
```

$$
\begin{aligned}
\eta_i &= \beta_0 + \beta_1 \text{Gender\_}M_i + \beta_2 \text{Conc}_i + \text{LogN}_i + \gamma_j + \alpha_k \\
y_i &\sim \text{Poisson}(\mu_i) \\
\eta_i &= g(\mu_i) \quad \text{where} \quad g = \log() \\
\pi(\boldsymbol{\beta}) &= N(0, 10 \cdot \mathbf{I}) \\
\gamma_j &\sim N(0, \sigma^2_\gamma), \ \pi(\sigma^2_\gamma) = \text{uniform}(0, 10) \\
\alpha_k &\sim N(0, \sigma^2_\alpha), \ \pi(\sigma^2_\alpha) = \text{normal}(\text{var} = 5)
\end{aligned}
$$

Outline

Example 1: Logistic Regression with Random Intercepts

Researchers investigated two medical procedures.

- 15 centers were randomly selected.
- Patients were assigned to two treatment groups, A or B.
- N is number of patients who received a given procedure.
- Sideeffect is the number of patients who reported side effects.

Example 1: Logistic Regression with Random Intercepts

Researchers investigated two medical procedures.

- 15 centers were randomly selected.
- Patients were assigned to two treatment groups, A or B.
- N is number of patients who received a given procedure.
- Sideeffect is the number of patients who reported side effects.

```
data MultiCenter;
   input Center Group$ N SideEffect @@;
   datalines;
 1  A  32  14   1  B  33  18
 2  A  30   4   2  B  28   8
 3  A  23  14   3  B  24   9
 4  A  22   7   4  B  22  10
 5  A  20   6   5  B  21  12
 6  A  19   1   6  B  20   3
 7  A  17   2   7  B  17   6
 8  A  16   7   8  B  15   9
 9  A  13   1   9  B  14   5
10  A  13   3  10  B  13   1
11  A  11   1  11  B  12   2
12  A  10   1  12  B   9   0
13  A   9   2  13  B   9   6
14  A   8   1  14  B   8   1
15  A   7   1  15  B   8   0
;
```

Example 1: Logistic Regression with Random Intercepts

- The response is the sample proportions of side effects as binomial ratios
- The fixed effect is Group
- The random effect cluster is Center

```
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
   outpost=CenterOut plots=all;
   class Center Group;
   model SideEffect/N = Group / noint;
   random int / sub = Center;
run;
```

PROC BGLIMM Output

The BGLIMM Procedure

Model Information	
Data Set	WORK.MULTICENTER
Response Variable	SideEffect
Distribution	Binomial
Link Function	Logit
Fixed Effects Included	Yes
Random Effects Included	Yes
Sampling Algorithm	Gamerman, Conjugate
Burn-In Size	500
Simulation Size	10000
Thinning	2
Random Number Seed	976352
Number of Threads	1

PROC BGLIMM Output

Class Level Information		
Class	**Levels**	**Values**
Center	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Group	2	A B

Number of Observations	
Number of Observations Read	30
Number of Observations Used	30
Number of Events	155
Number of Trials	503

Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Group A	5000	-1.3895	0.3102	-2.0071	-0.7956
Group B	5000	-0.8839	0.2968	-1.4819	-0.3186
Random Var	5000	0.9184	0.4198	0.3024	1.7515

Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Group A	5000	-1.3895	0.3102	-2.0071	-0.7956
Group B	5000	-0.8839	0.2968	-1.4819	-0.3186
Random Var	5000	0.9184	0.4198	0.3024	1.7515

- Group A vs. Group B

Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Group A	5000	-1.3895	0.3102	-2.0071	-0.7956
Group B	5000	-0.8839	0.2968	-1.4819	-0.3186
Random Var	5000	0.9184	0.4198	0.3024	1.7515

- Group A vs. Group B
- Random Var measures variability of center-level intercepts.

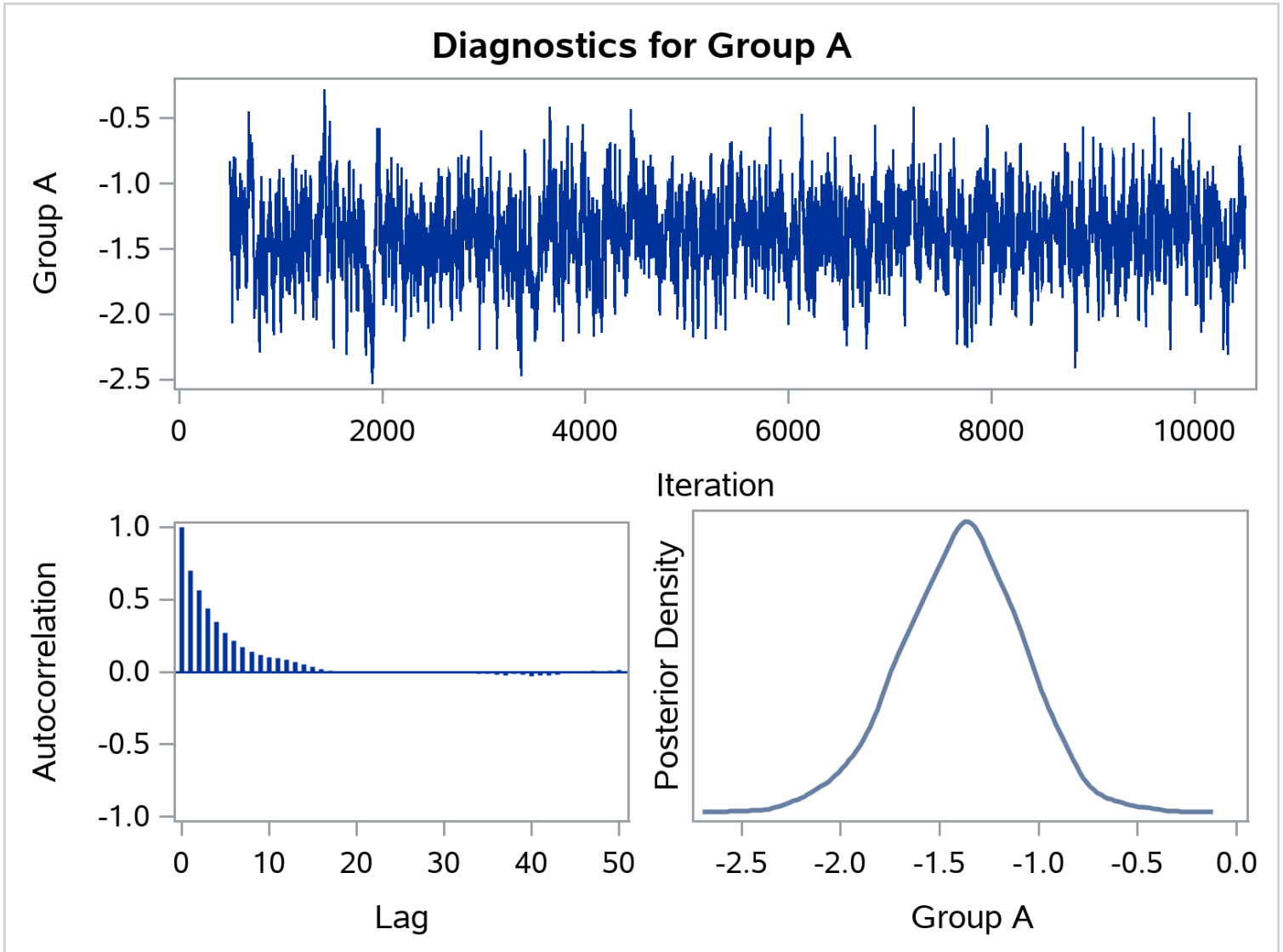
Posterior Summary Statistics

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Group A	5000	-1.3895	0.3102	-2.0071	-0.7956
Group B	5000	-0.8839	0.2968	-1.4819	-0.3186
Random Var	5000	0.9184	0.4198	0.3024	1.7515

- Group A vs. Group B
- Random Var measures variability of center-level intercepts.
- You can print each center's intercept by specifying the MONITOR option in the RANDOM statement.

TAD (Trace, Autocorrelation, Density) Plots



ESTIMATE Statement

Use the ESTIMATE statement to get the log of odds ratios between A and B:

```
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
   outpost=CenterOut;
   class Center Group;
   model SideEffect/N = Group / noint;
   random int / sub = Center;
   estimate "log OR" group 1 -1;
run;
```

ESTIMATE Statement

Use the ESTIMATE statement to get the log of odds ratios between A and B:

```
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
   outpost=CenterOut;
   class Center Group;
   model SideEffect/N = Group / noint;
   random int / sub = Center;
   estimate "log OR" group 1 -1;
run;
```

The BGLIMM Procedure

Results from ESTIMATE Statements

Label	Mean	Standard Deviation	95% HPD Interval	
log OR	-0.5056	0.2087	-0.9292	-0.1102

Functions of Parameters

Example: Probability difference between A and B:

$$\text{pDiff} = \frac{\exp(\beta_b)}{1+\exp(\beta_b)} - \frac{\exp(\beta_a)}{1+\exp(\beta_a)}$$

```
data prob;
   set CenterOut;
   pDiff = logistic(group_b)
         - logistic(group_a);
run;

%sumint(data=prob, var=pDiff)
```

Functions of Parameters

Example: Probability difference between A and B:

$$\text{pDiff} = \frac{\exp(\beta_b)}{1+\exp(\beta_b)} - \frac{\exp(\beta_a)}{1+\exp(\beta_a)}$$

```
data prob;
   set CenterOut;
   pDiff = logistic(group_b)
           - logistic(group_a);
run;

%sumint(data=prob, var=pDiff)
```

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
pDiff	5000	0.0920	0.0395	0.0195	0.1750

Functions of Parameters

Example: Probability difference between A and B:

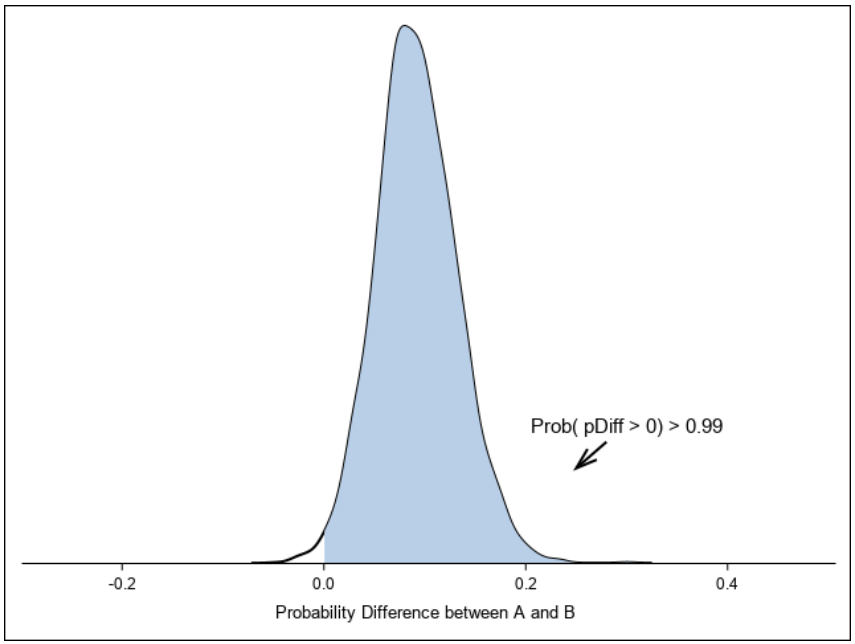
$$\text{pDiff} = \frac{\exp(\beta_b)}{1+\exp(\beta_b)} - \frac{\exp(\beta_a)}{1+\exp(\beta_a)}$$

```
data prob;
   set CenterOut;
   pDiff = logistic(group_b)
         - logistic(group_a);
run;

%sumint(data=prob, var=pDiff)
```

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
pDiff	5000	0.0920	0.0395	0.0195	0.1750



Built-In Bayesian Macros

%ESS	Effective sample sizes
%GEWEKE	Geweke diagnostic
%HEIDEL	Heidelberger-Welch diagnostic
%MCSE	Monte Carlo standard errors
%POSTACF	Autocorrelation
%POSTCOR	Correlation matrix
%POSTCOV	Covariance matrix
%POSTINT	Equal-tail and HPD intervals
%POSTSUM	Mean, std, and quantiles
%RAFTERY	Raftery diagnostic
%SUMINT	Mean, std, and HPD intervals
%TADPLOT	Trace plot, AC plot, and density plot

Outline

Example 2: Repeated Measurements with Heterogeneity

Example 2: Repeated Measurements with Heterogeneity

- A two-treatment trial for patients with rheumatoid arthritis
- 67 subjects enrolled
- Subjects were followed up three times
- A grip strength measurement was taken at each follow-up visit
- A baseline grip strength (in mmHg) was measured at the start

```
data GripData;
input Subject Baseline Treat Gender$ Time Grip;
datalines;
26  175  1  M  1  161
26  175  1  M  2  210
26  175  1  M  3  230
27  165  1  M  1  215
27  165  1  M  2  245
27  165  1  M  3  265
...

71  104  2  F  1  107
71  104  2  F  2  .
71  104  2  F  3  .
72   60  2  F  1   60
72   60  2  F  2   55
72   60  2  F  3   58
;
```

Model Details

A reasonable initial model should involve fairly general specifications for both the mean and the variance-covariance structure (Littell et al. 2006).

- The mean includes three-way interactions of the variables Gender,Treat, and Time and the Baseline measure.
- To allow general within-subject heterogeneity, the unstructured type is used in the R-side covariance matrix.
  - An advantage of considering this most general covariance type is to inspect the estimates for heterogeneous patterns in both the variances and correlations.

Use PROC BGLIMM for Repeated Measurements

- The fixed effects contain 12 cell means: 2 treatments by 2 genders by 3 visits.
- The REPEATED statement specifies that the repeated measurements were taken over Time and are grouped according to the Subject variable.

```
proc bglimm data=GripData seed=475193;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   repeated Time / sub=Subject type=un r rcorr;
   run;
```

Posterior R-Side Covariance and Correlation Matrices

Requested via the R and RCORR options.

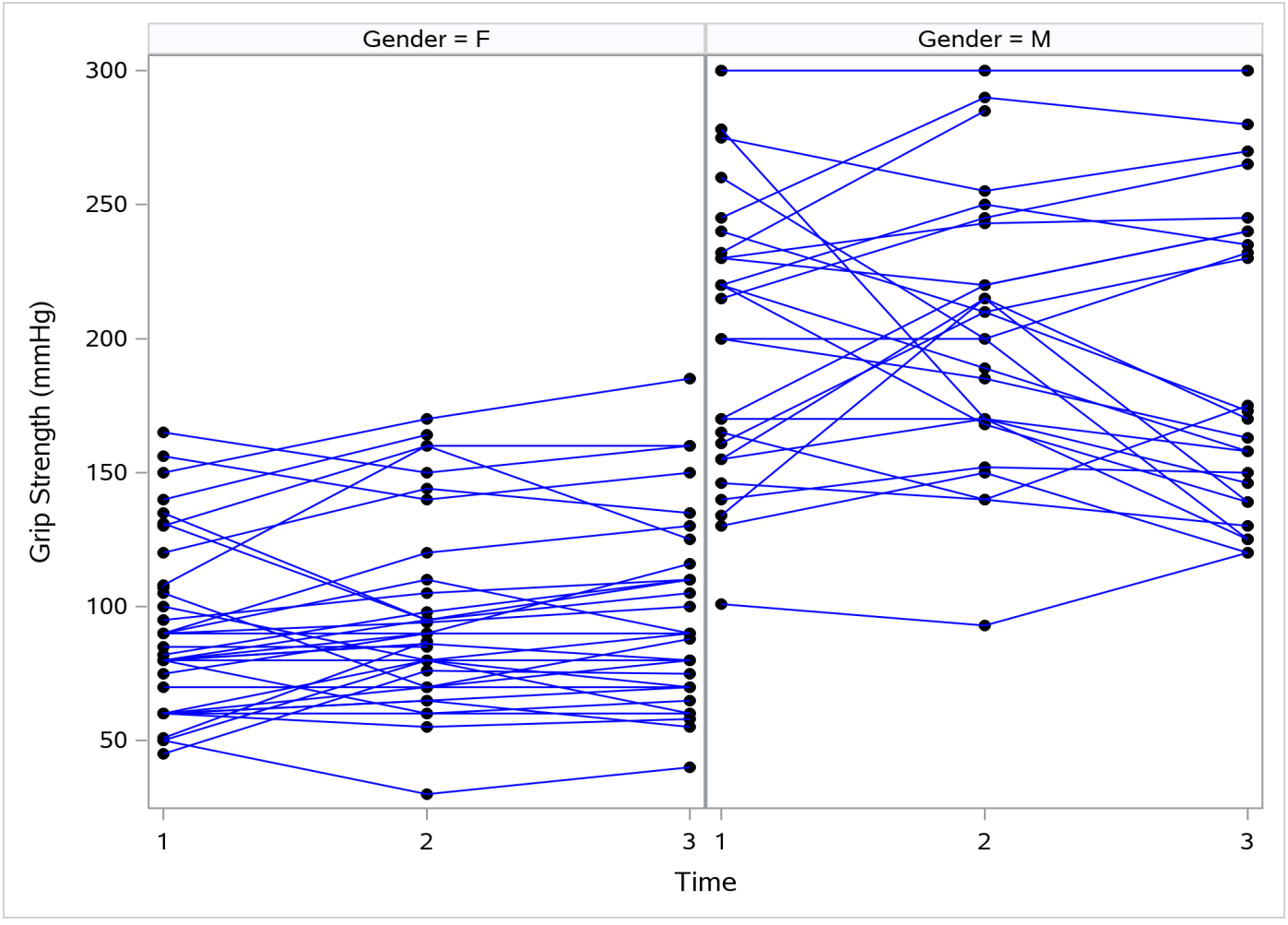
Estimated R Matrix

Row	Col 1	Col 2	Col 3
1	604.96	308.00	288.96
2	308.00	950.48	885.65
3	288.96	885.65	1304.71

Estimated R Correlation Matrix

Row	Col 1	Col 2	Col 3
1	1.0000	0.4062	0.3252
2	0.4062	1.0000	0.7953
3	0.3252	0.7953	1.0000

Grip Strength Measurements over Time by Gender



Between-Subject Heterogeneity by Gender

To account for distinct covariance structures of the two gender groups, you can fit the model by adding the GROUP=GENDER option to the REPEATED statement:

```
proc bglimm data=GripData seed=475193;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   repeated Time / sub=Subject type=un group=Gender r rcorr;
run;
```

R-Side Covariance Matrices by Gender

The BGLIMM Procedure

Estimated R Matrix

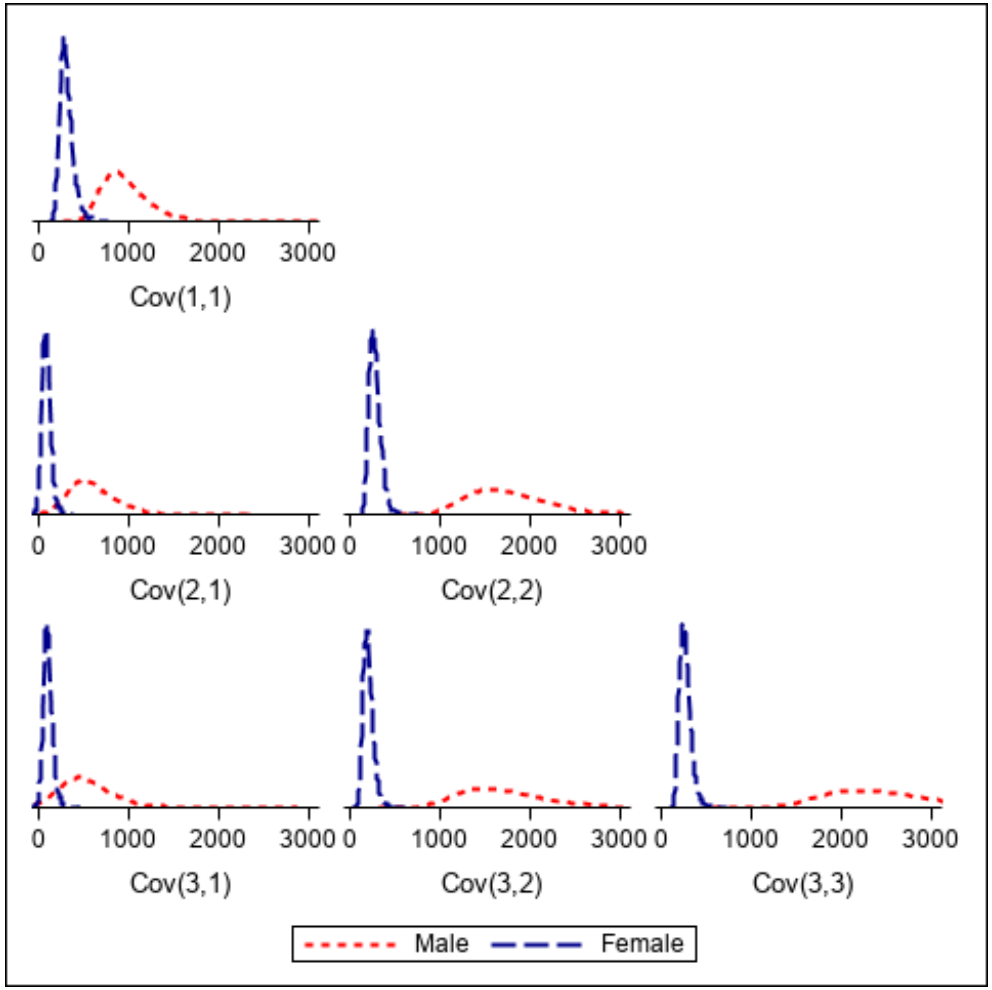
Group	Row	Col 1	Col 2	Col 3
Gender F	**1**	300.08	77.2769	95.2165
Gender F	**2**	77.2769	267.23	195.20
Gender F	**3**	95.2165	195.20	257.48
Gender M	**1**	960.37	591.43	528.93
Gender M	**2**	591.43	1773.63	1710.94
Gender M	**3**	528.93	1710.94	2504.11

R-Side Covariance Matrices by Gender

The BGLIMM Procedure

Estimated R Matrix

Group	Row	Col 1	Col 2	Col 3
Gender F	1	300.08	77.2769	95.2165
Gender F	2	77.2769	267.23	195.20
Gender F	3	95.2165	195.20	257.48
Gender M	1	960.37	591.43	528.93
Gender M	2	591.43	1773.63	1710.94
Gender M	3	528.93	1710.94	2504.11



Between-Subject Heterogeneity in Random Effects

You can account for more between-subject heterogeneity by adding a RANDOM statement.

```
proc bglimm data=GripData seed=475193 nmc=20000 thin=4;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   random int / sub=Subject group=Gender covprior=uniform;
   repeated Time / sub=Subject type=un group=Gender r rcorr
            covprior=iw(scale=500);
run;
```

Discussion

- SAS/STAT has two Bayesian procedures to fit hierarchical models: PROC MCMC and PROC BGLIMM.
  - PROC MCMC offers more flexibility (e.g. priors, non-linearity, etc.), but requires programming.
  - PROC BGLIMM focuses on a subclass of hierarchical models, is optimized, and is convinient to use.
- Both procedures use MCMC techniques to generate samples to approximate the exact posterior distributions. And they offer a rich set of features (e.g. hierarchical structures to arbituary depth, handling of missing data, etc).

Thank You!