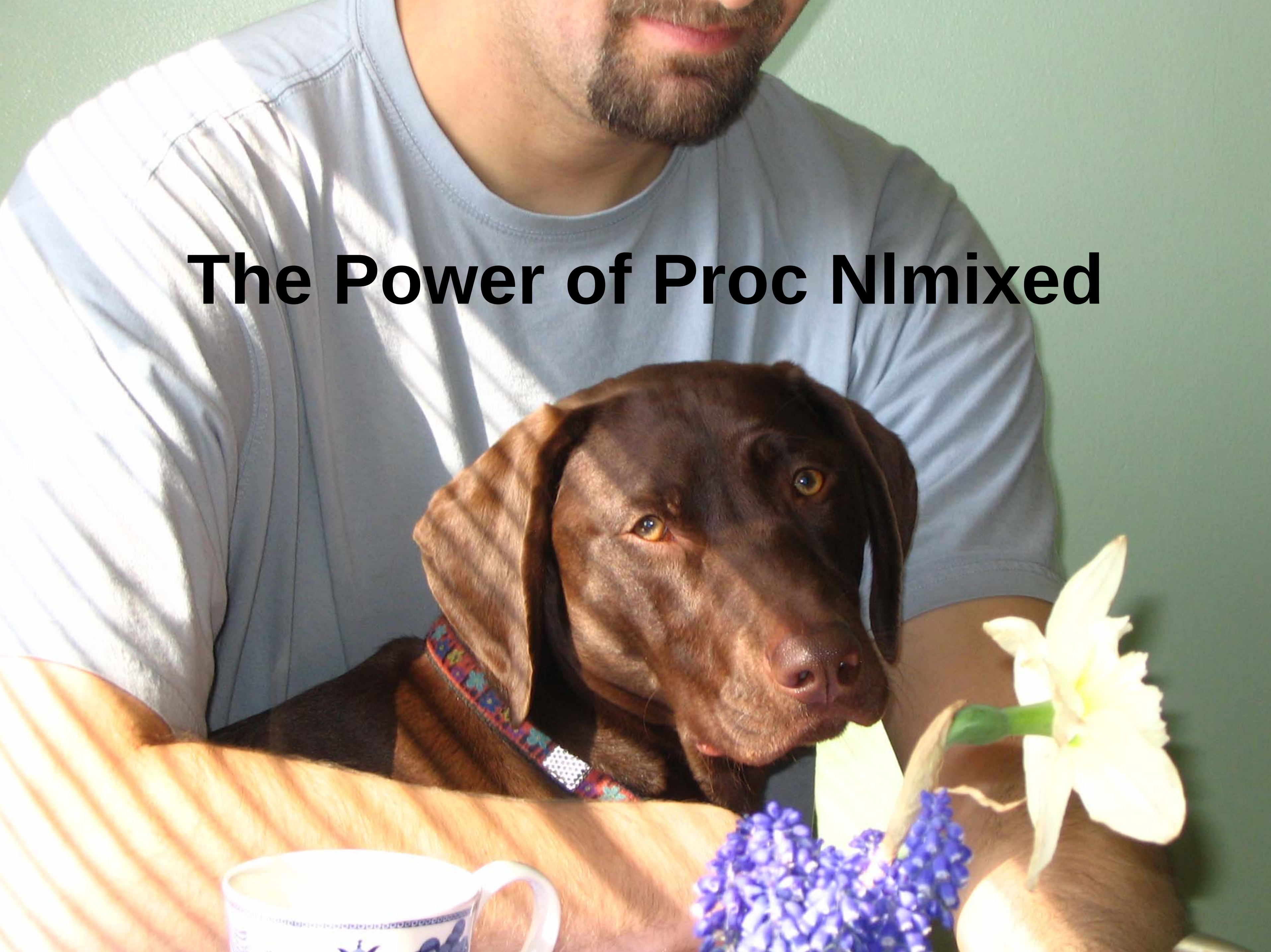
The Power of Proc Nlmixed



Introduction

- Proc Nlmixed fits *nonlinear mixed-effects models* (NLMMs) – models in which the fixed and random effects have a nonlinear relationship

- NLMMs are widespread in pharmacokinetics – ? genesis of procedure

- Nlmixed was first available in Version 7 (experimental) and then in Version 8 (production)

Introduction (cont.)

- Nlmixed is similar to the Nlinmix and Glimmix macros but uses a different estimation method, and is much easier to use

- Macros iteratively fit a set of GEEs, whereas Nlmixed directly maximizes an approximation of the likelihood, integrated over the random effects

Example: logistic regression with residual error

- Say you design an experiment with a single treatment that has $i = 1,2,3$ levels
- Each treatment level is randomly assigned to a bunch of plots, and each plot contains m = 30 trees
- Replication is balanced, so that the same number of plots $j = 1,2,\ldots,100$ occur within each treatment level
- Within each plot you measure y, the number of trees within the plot that are infected by some disease
- The objective is to see whether the treatment has an effect on the incidence of the disease

Example (cont.)

Modeling this scenario…

$$y_{ij} \mid p_{ij} \sim \text{Binomial}\left(m, p_{ij}\right)$$

$$p_{ij} = \frac{\exp\left(\mu + \alpha_i + \varepsilon_{ij}\right)}{1 + \exp\left(\mu + \alpha_i + \varepsilon_{ij}\right)} \quad \text{or} \quad \log\left(\frac{p_{ij}}{1 - p_{ij}}\right) = \mu + \alpha_i + \varepsilon_{ij}$$

$$\varepsilon_{ij} \sim N\left(0, \sigma^2\right)$$

Example (cont.)

Consequences of model:

$$E(y_{ij}) = m\pi_i \quad \text{where} \quad \pi_i = E(p_{ij}) \cong \frac{\exp(\mu + \alpha_i)}{1 + \exp(\mu + \alpha_i)}$$

$$Var(y_{ij}) \cong m\pi_i(1 - \pi_i)\left[1 + \sigma^2(m-1)\pi_i(1 - \pi_i)\right]$$

Notice that the inflation factor $[\bullet]$ is a function of π_i, and not constant (i.e. it changes for each treatment level)

Example (cont.)

How do we fit this model in SAS?

- Proc Catmod or Nlin – crude model without any overdispersion
- Proc Logistic or Genmod – simple overdispersed model
- Glimmix or Nlinmix macros – iterative GEE approach
- Proc Nlmixed – exact (sort of) approach

Proc Genmod code

```
proc genmod data=fake;
 class treat;
 model y/m=treat / scale=p link=logit
  dist=binomial type3;
 title 'Random-effects Logistic Regression
  using Proc Genmod';
 output out=results pred=pred;
run;
```

Proc Genmod Output

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	297	646.7835	2.1777
Pearson Chi-Square	297	607.1128	2.0442
Log Likelihood		-2119.8941	

Analysis Of Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	0.1389	0.0523	0.0363	0.2415	7.04	0.0080
treat	1	1	-2.0194	0.0931	-2.2019	-1.8368	470.18	<.0001
treat	2	1	1.8726	0.0964	1.6838	2.0615	377.65	<.0001
treat	3	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale		0	1.4297	0.0000	1.4297	1.4297		

NOTE: The scale parameter was estimated by the square root of Pearson's Chi-Square/DOF.

LR Statistics For Type 3 Analysis

Source	Num DF	Den DF	F Value	Pr > F
treat	2	297	929.55	<.0001

Glimmix macro code

```
%inc 'h:\SASPROGS\Glimmix macro\glmm800.sas' /
  nosource;

%glimmix(data=fake, procopt=%str(method=reml
  covtest maxiter=100), maxit=100, out=results,
  stmts=%str(
  class treat ident;
  model y/m = treat / ddfm=residual; random
  ident;
  parms (0.25) (1) / hold=2;
  title 'Random-effects Logistic Regression
  using the Glimmix macro';
  ),
  error=binomial, link=logit);
run;
```

Glimmix macro output

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error	Z Value	Pr Z
ident	0.2283	0.03780	6.04	<.0001
Residual	1.0000	0	.	.

Fit Statistics

-2 Res Log Likelihood	639.5

Solution for Fixed Effects

Effect	treat	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		0.1423	0.06058	297	2.35	0.0195
treat	1	-2.0654	0.09475	297	-21.80	<.0001
treat	2	1.8989	0.09614	297	19.75	<.0001
treat	3	0	.	.	.	.

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
treat	2	297	723.42	<.0001

Proc Nlmixed code

```
proc nlmixed data=fake tech=trureg df=297;
 bounds sigma2>0;
 parms mu=1 t1=-2 t2=2 sigma2=0.25;
 if treat=1 then eta=mu + t1 + e;
 else if treat=2 then eta=mu + t2 + e;
 else eta=mu + e;
 prob=exp(eta)/(1+exp(eta));
 model y ~ binomial(m, prob);
 random e ~ normal(0, sigma2) subject=ident;
 contrast 'treat' t1, t2;
 predict prob out=results;
 title 'Random-effects Logistic Regression
  using Proc Nlmixed';
run;
```

Proc Nlmixed output

Fit Statistics

-2 Log Likelihood	1473.3

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
mu	0.1455	0.06199	297	2.35	0.0195
t1	-2.1197	0.09782	297	-21.67	<.0001
t2	1.9499	0.09887	297	19.72	<.0001
sigma2	0.2423	0.04118	297	5.88	<.0001

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
treat	2	297	694.22	<.0001

Results

Parameter	True value	Parameter estimates				
		Nlin	Genmod	Glimmix	Nlmixed	Nlinmix
μ	0.0	0.14	0.14	0.14	0.15	0.14
α_1	-2.0	-2.02	-2.02	-2.07	-2.12	-2.02
α_2	2.0	1.87	1.87	1.90	1.95	1.87
σ^2	0.25	n/a	0.144*	0.228	0.242	0.301
F	?	3163.4	929.6	723.4	694.2	699.2

- treatment is significant
- estimates are similar
- Nlimixed works

Core syntax

- Proc Nlmixed statement options
  - tech=
    - optimization algorithm
    - several available (e.g. trust region)
    - default is dual quasi-Newton
  - method=
    - controls method to approximate integration of likelihood over random effects
    - default is adaptive Gauss-Hermite quadrature

Syntax (cont.)

- Model statement
  - specify the conditional distribution of the data given the random effects
  - e.g. $\mathbf{y} \mid \mathbf{u} \sim N\left(\mathbf{X\beta} + \mathbf{Zu}, \mathbf{R}\right)$
  - Valid distributions:
    - normal(*m, v*)
    - binary(*p*)
    - binomial(*n, p*)
    - gamma(*a ,b*)
    - negbin(*n, p*)
    - Poisson(*m*)
    - general(*log likelihood*)

Syntax (cont.)

Fan-shaped error model:

$$y_i \sim N\left(a + bx_i, \sigma^2 x_i\right)$$

```
proc nlmixed;
 parms a=0.3 b=0.5 sigma2=0.5;
 var = sigma2*x;
 pred = a + b*x;
 model y ~ normal(pred, var);
run;
```

Syntax (cont.)

- Binomial:

```
model y ~ binomial(m,prob);
```

- General:

```
combin = gamma(m+1)/(gamma(y+1)*gamma(m-y+1));
loglike = y*log(prob)+(m-y)*log(1-prob)+
          log(combin);
model y ~ general(loglike);
```

Syntax (cont.)

- Random statement
  - defines the random effects and their distribution
  - e.g. $\mathbf{u} \sim N\left(\mathbf{0}, \mathbf{G}\right)$
  - The input data set must be clustered according to the SUBJECT= variable.

- Estimate and contrast statements also available

Summary

Pros

- Syntax fairly straightforward
- Common distributions (conditional on the random effects) are built-in – via the model statement
- Likelihood can be user-specified if distribution is non-standard
- More exact than glimmix or nlinmix and runs faster than both of them

Summary (cont.)

Cons

- Random effects must come from a (multivariate) normal distribution
- All random effects must share the same subject (i.e. cannot have multi-level mixed models)
- Random effects cannot be nested or crossed
- DF for Wald-tests or contrasts generally require manual intervention

A Smidgeon of Theory…

Linear mixed-effects model

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{Zu} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{R}) \quad \text{and} \quad \mathbf{u} \sim N(\mathbf{0}, \mathbf{G})$$

$$E(\mathbf{y} \mid \mathbf{u}) = \mathbf{X\beta} + \mathbf{Zu}$$

$$\mathbf{y} \mid \mathbf{u} \sim N(\mathbf{X\beta} + \mathbf{Zu}, \mathbf{R})$$

$$\mathbf{y} \sim N(\mathbf{X\beta}, \mathbf{ZGZ}' + \mathbf{R})$$

Theory (cont.)

Generalized linear mixed-effect model

$$E(\mathbf{y} \mid \mathbf{u}) = g(\mathbf{X\beta} + \mathbf{Zu})$$

$$g^{-1}(E(\mathbf{y} \mid \mathbf{u})) = \mathbf{X\beta} + \mathbf{Zu}$$

$$\mathbf{u} \sim N(\mathbf{0}, \mathbf{G})$$

$$\mathbf{y} \mid \mathbf{u} \sim f_Y(\mathbf{y} \mid \mathbf{u})$$

- The marginal distribution of $\mathbf{y}$ cannot usually be simplified due to nonlinear function $g(\cdot)$

Theory (cont.)

Nonlinear mixed-effect model

$$E(\mathbf{y} \mid \mathbf{u}) = h(\mathbf{X}, \boldsymbol{\beta}, \mathbf{Z}, \mathbf{u})$$

- Link h^{-1} (yielding linear combo) does not exist

$$\mathbf{u} \sim f_U(\mathbf{u})$$

$$\mathbf{y} \mid \mathbf{u} \sim f_Y(\mathbf{y} \mid \mathbf{u})$$

- The marginal distribution of $\mathbf{y}$ is unavailable in closed form

Conclusion

- Flexibility of proc nlmixed makes it a good choice for many non-standard applications (e.g. non-linear models), even those without random effects.

References

- Huet, S., Bouvier, A., Poursat, M.-A., and E. Jolivet. 2004. Statistical Tools for Nonlinear Regression. A Practical Guide with S-PLUS and R Examples, Second Edition. Springer-Verlag. New York.
- Littell, R.C., Milliken, G.A., Stroup, W.W., and R.D. Wolfinger. 1996. SAS System for Mixed Models. Cary, NC. SAS Institute Inc.
- McCullogh, C.E. and S.R. Searle. 2001. Generalized, Linear, and Mixed Models. John Wiley & Sons. New York
- Pinheiro, J.C. and D.M. Bates. 2000. Mixed-effects Models in S and S-PLUS. Springer-Verlag. New York.
- SAS Institute Inc. 2004. SAS OnlineDoc® 9.1.3. Cary, NC: SAS Institute Inc.