Bayesian Computation Using PROC MCMC

Fang Chen
SAS Institute Inc.
fangk.chen@sas.com

Bayes Pharma
May 20th, 2015

Goals

- Introduce PROC MCMC
- Illustrate features of PROC MCMC relevant to the pharmaceutical industry through examples

Outline

Outline

The MCMC Procedure is a General Simulation Procedure

- single-level or multilevel (hierarchical) models
- linear or nonlinear models, such as regression, survival, ordinal multinomial
- missing data problems
- PK models, latent variable models, state space models

The current release is SAS/STAT®13.2, the second maintenance release of SAS 9.4, and Revision 14w32. The next release is SAS/STAT®14.1, Revision 15w29.

Typical PROC Call is a Mixture of Statements and DATA Step Language

PROC MCMC *options*;

PARMS; define parameters.

PRIOR; declare prior distributions

Programming statements;
MODEL; } define log-likelihood function

PREDDIST; posterior prediction

RANDOM; random effects

UDS; User-Defined Sampler

run;

Simple Example

$$
\begin{aligned}
\text{weight}_i &\sim \text{normal}(\mu_i, \text{var} = \sigma^2) \\
\mu_i &= \beta_0 + \beta_1 \cdot \text{height}_i \\
\beta_0, \beta_1 &\sim \text{normal}(0, \text{var} = 100) \\
\sigma^2 &\sim \text{inverse Gamma}(\text{shape} = 2, \text{scale} = 2)
\end{aligned}
$$

The data:

```
data class;
   input height weight;
datalines;
   69.0      112.5
   56.5       84.0
   65.3       98.0
...
   57.5       85.0
   66.5      112.0
;
```

PROC MCMC Program Reflects the Statistical Model

$$
\begin{aligned}
\text{weight}_i &\sim \text{normal}(\mu_i, \text{var} = \sigma^2) \\
\mu_i &= \beta_0 + \beta_1 \cdot \text{height}_i \\
\beta_0, \beta_1 &\sim \text{normal}(0, \text{var} = 100) \\
\sigma^2 &\sim \text{inverse Gamma}(\text{shape} = 2, \text{scale} = 2)
\end{aligned}
$$

```
proc mcmc data=class seed=1 nbi=5000 nmc=10000 outpost=regOut;
   parms beta0 beta1 s2;
   prior beta: ~ normal(0, var=100);
   prior s2 ~ igamma(shape=2, scale=2);
   mu = beta0 + beta * height;
   model weight ~ normal(mu, var=s2);
   run;
```

Built-in Flexibilities

$$
\begin{aligned}
\text{weight}_i &\sim \text{t}(\mu_i, \text{sd} = \sigma, \text{df} = 3) \\
\mu_i &= \beta_0 + \beta_1 \cdot \text{height}_i \\
\beta_0, \beta_1 &\sim \text{normal}(0, \text{var} = 100) \\
\sigma &\sim \text{uniform}(0, 25)
\end{aligned}
$$

Change the model, parameterization, and so on as required:

```
proc mcmc data=class seed=1 nbi=5000 nmc=10000 outpost=regOut;
   parms beta0 beta1 sig;
   prior beta: ~ normal(0, var=100);
   prior sig ~ uniform(0, 25);
   mu = beta0 + beta1 * height;
   model weight ~ t(mu, sd=sig, df=3);
   run;
```

Built-in Flexibilities

$$
\begin{aligned}
\text{weight}_i &\sim \text{poisson}(\lambda_i) \\
\lambda_i &= \exp(\beta_0 + \beta_1 \cdot \text{height}_i) \\
\beta_0, \beta_1 &\sim \text{normal}(0, \text{var} = 100)
\end{aligned}
$$

You can fit generalized or nonliear models:

```
proc mcmc data=class seed=1 nbi=5000 nmc=10000 outpost=regOut;
   parms beta0 beta1;
   prior beta: ~ normal(0, var=100);
   lambda = exp(beta0 + beta1 * height);
   model weight ~ poisson(lambda);
   run;
```

What Does the Procedure Produce

- samples from the posterior
- posterior statistics (mean, s.d., HPD, etc)
- convergence diagnostics (ESS, Geweke, MCSE, etc)
- graphical display (trace plot, ACF plot, KDE plot)

Sampling Algorithm Hierarchy

	Continuous Parameters	**Discrete Parameters**
When Applicable	Conjugate Direct	Conjugate Direct Inverse CDF
All Others	**RWM** RWM-t HMC NUTS slice	**Discrete RWM** Geometric RWM

Additional Features

- mix and match sampling algorithms
- control parameters updating sequence
- implement your own sampler (the UDS Statement)
- obtain optimized estimates (no MCMC run)
- multithread
  - log likelihood computation
  - sampling of conditionally independent parameters

The Posterior Distribution

PROC MCMC uses the general posterior form:

$$\pi(\theta|\boldsymbol{y},\boldsymbol{x}) \propto \pi(\theta)\cdot f(\boldsymbol{y}|\theta,\boldsymbol{x})$$

- The PRIOR statements define the prior distributions: $\pi(\theta)$.
- The MODEL statement defines the likelihood function for each observation in the data set: $f(y_i|\theta,x_i)$, for $i = 1,\cdots,n$
- The posterior distribution (on the log scale):

$$\log(\pi(\theta|\boldsymbol{y},\boldsymbol{x})) = \log(\pi(\theta)) + \sum_{i=1}^{n}\log(f(y_i|\theta,x_i))$$

where $\boldsymbol{y} = \{y_i\}$ and $\boldsymbol{x} = \{x_i\}$

The Procedure Loops Through the Input Data Set to Compute $\log(\pi(\theta|\boldsymbol{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

at the top of the data set

$$\log \pi(\theta|\boldsymbol{y}) = \log(f(y_1|\theta))$$

The Procedure Loops Through the Input Data Set to Compute $\log(\pi(\theta|\boldsymbol{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0


```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

stepping through the data set

$$\log \pi(\theta|\boldsymbol{y}) = \log \pi(\theta|\boldsymbol{y}) + \log(f(y_2|\theta))$$

The Procedure Loops Through the Input Data Set to Compute $\log(\pi(\theta|\boldsymbol{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

stepping through the data set

$$\log \pi(\theta|\boldsymbol{y}) = \log \pi(\theta|\boldsymbol{y}) + \log(f(y_3|\theta))$$

The Procedure Loops Through the Input Data Set to Compute $\log(\pi(\theta|\boldsymbol{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model;
   run;
```

at the last observation, the prior is included

$$\log \pi(\theta|\boldsymbol{y}) = \log(\pi(\theta)) + \sum_{i=1}^{n} \log(f(y_i|\theta))$$

PROC MCMC and WinBUGS Syntax

In WinBUGS, a for-loop and array indices are used to access records in variables; In PROC MCMC, the looping over the data set is implicit.

```
parms beta tau;
prior beta ~ normal(0, prec=0.1);
prior tau ~ gamma(0.1, is=0.1);
mu = beta * height;
model weight ~ normal(mu, prec=tau);
```

```
model {
  for(i in 1:19) {
    mu[i] = beta * height[i]
    weight[i] ~ dnorm(mu[i], tau)
  }
  beta ~ dnorm(0, 0.1)
  tau ~ gamma(0.1, 0.1)
}
```

Distributions

PROC MCMC supports the usual suspects in standard distributions:

- Univariate: normal, uniform, beta, igamma, gamma, t, poisson, etc
- Multivariate: mvn, iwish, dirich, multinom
- Categorical: table
- Truncation: `lower=` and `upper=`

You can also define non-standard distributions using DATA step language (more later).

DATA Step Language Offers Versatility

Most DATA step operators, functions, and statements can be used in PROC MCMC. You can

- debug the program
- compute functions of parameters
- construct general prior and/or likelihood functions

Programs and derivative computations are translated and executed in C.

Outline

Two Data Sets

Researchers are interested in evaluating the performance of a medical procedure in a multicenter study. There have been two studies, a historical data and a current data:

```
data pilot;
   input event n;
   datalines;
   5  163
;
```

```
data trials;
   input event n center;
   datalines;
   2  86  1
   2  69  2
   1  71  3
   1 113  4
   1 103  5
;
```

event: number of deaths

n: number of patients assigned to the treatment procedure

center: center index

Various Ways of Utilizing Information from the Pilot Data in Constructing a Prior

- parametrix approximation (MAP approach)
- commensurate prior
- nonparametrix approximation (use KDEs)
- power prior

$$p(\theta|D_0, a_0) \propto L(\theta; D_0)^{a_0} \cdot \pi_0(\theta)$$

Power Prior with Fixed a_0 is Relatively Straightforward

The posterior distribution can be factored in the following way:

$$p(\theta|D^*, a_0) \propto \prod_{i=1}^{n+n_0} f_i(y_i|\theta, x_i) \cdot \pi_0(\theta)$$

$$\text{where } f_i = \begin{cases} f(y_i|\theta, x_i) & \text{for each } i \text{ in the current data set} \\ f(y_{0,i}|\theta, x_{0,i})^{a_0} & \text{for each } i \text{ in the historical data set} \end{cases}$$

Need to assign the appropriate likelihood function to each observations.

But $f(y_{0,i}|\theta, x_{0,i})^{a_0}$ does not have a standard form.

Specifying a Nonstandard Distribution

The GENERAL and DGENERAL functions enable you construct your own prior or likelihood function. The "D" stands for discrete.

PRIOR `alpha` $\sim$ `dgeneral(lp);`
MODEL `y` $\sim$ `general(llike);`

The expressions `lp` and `llike` must take the values of the **logarithm** of the distribution.

The normalizing constant of the distribution can be ignored, as long as it is independent of other parameters in the model.

The GENERAL Distribution

Suppose that you want to use the following prior:

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2}$$

which is a nonstandard distribution (nonintegrable prior). The logarithm of this prior is

$$\log(\pi(\sigma^2)) = -\log(\sigma^2) + C$$

You use the following statements to declare this prior:

```
lp = -log(sigma2);
prior sigma2 ~ general(lp, lower=0);
```

Fitting Power Prior with Fixed a_0

OBS	event	n	group
1	2	86	current
2	2	69	current
3	1	71	current
4	1	113	current
5	1	103	current
6	5	163	pilot

```
proc mcmc data=alldata nmc=50000 outpost=a1;
   parm p 0.2;
   a0 = 0.2;
   prior p ~ uniform(0, 1);
   llike = logpdf("binomial", event, p, n);
   if (group = "pilot") then
      llike = a0 * llike;
   model general(llike);
   run;
```

The `general` function specifies the likelihood function, which is either a binomial (`current`) or a weighted binomial (`pilot`).

Prior on a_0 Requires Integration

In specifying $\pi(a_0)$, you must compute the normalizing constant:

$$\begin{aligned} p(\theta, a_0|D_0) &\propto p(\theta|D_0, a_0) \cdot \pi(a_0) \\ &= \frac{L(\theta; D_0)^{a_0} \cdot \pi_0(\theta)}{\int L(\theta; D_0)^{a_0} \cdot \pi_0(\theta) d\theta} \cdot \pi_0(a_0) \\ &= \frac{1}{C(a_0)} \cdot L(\theta; D_0)^{a_0} \cdot \pi_0(\theta) \cdot \pi_0(a_0) \end{aligned}$$

The CALL QUAD subroutinne computes integral of a user-specific functionn (defined using PROC FCMP)

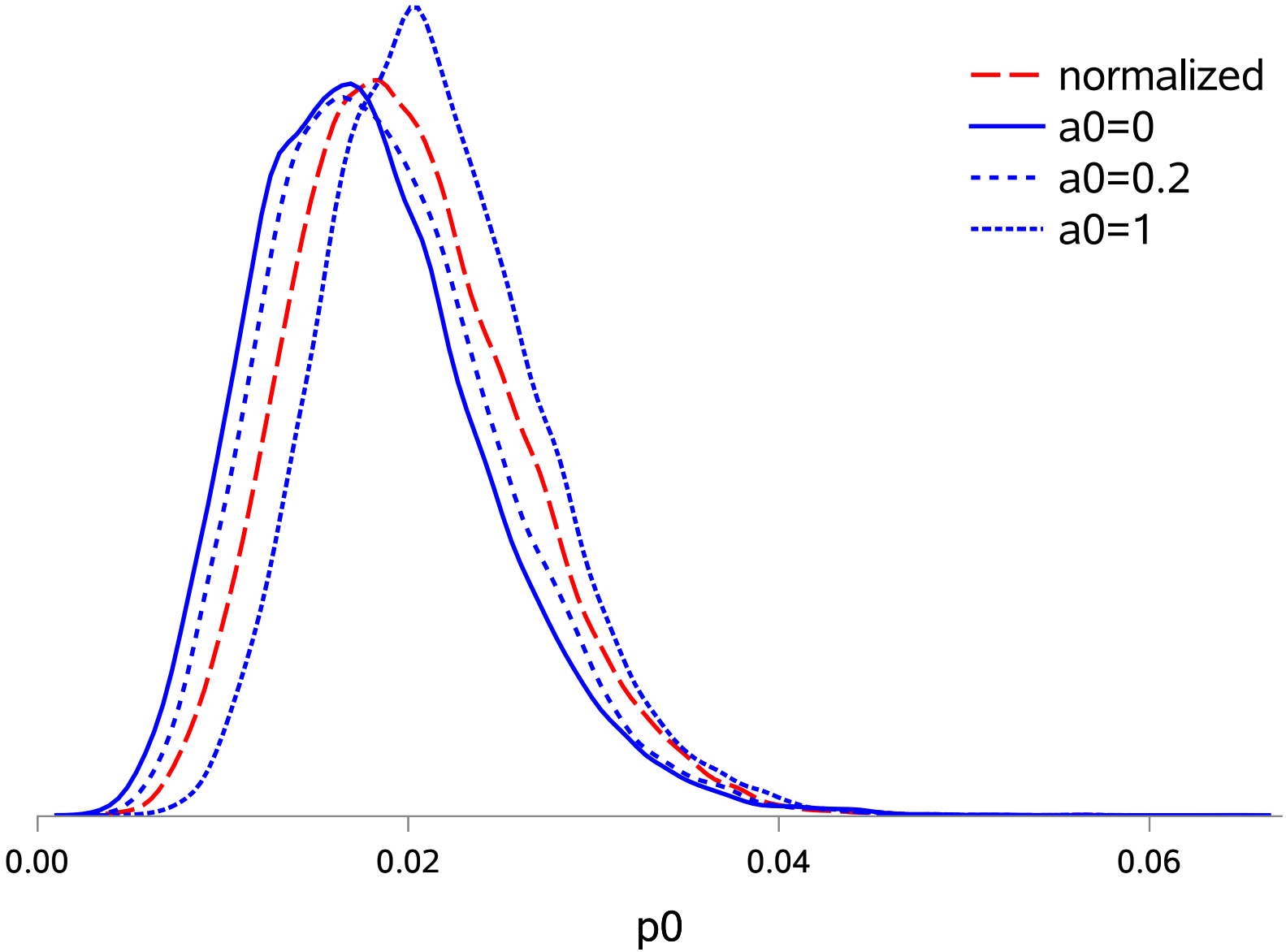
- Adaptive Romberg for univariate problem
- Laplace for multidimensional problem

Normalized Power Prior

```
proc fcmp outlib=sasuser.funcs.power;
   subroutine bPower(p, den, y, n, a0);          ! integration w.r.t. p
   outargs den;
   den = exp(a0 * logpdf("binomial", y, p, n));  ! L(p; D0)^a0
   endsub;
run;

options cmplib=sasuser.funcs;
proc mcmc data=alldata seed=17 nmc=50000 outpost=npout;
   parm p 0.5;
   parm a0 0.2;
   prior p a0 ~ uniform(0, 1);                   ! a0 is a parameter
   llike = logpdf("binomial", event, p, n);
   if (group = 'pilot') then do;
      CALL QUAD('bPower', C, 0, 1, event, n, a0); ! C is the integral
      llike = -log(C) + a0 * llike;
      end;
   model general(llike);
   run;
```

Posterior Distribution Comparison



The RANDOM Statement Constructs Random Effects in a Model

The statement supports both

- conditional independence models (most common):

$$\begin{aligned} \beta_j &\sim \pi(\theta) \\ \beta_i &\perp \beta_j \quad \textit{a priori} \end{aligned}$$

- dependence (autoregressive type of) models:

$$\begin{aligned} \beta_j &\sim \pi(\beta_{j-k}, \beta_{j+l}, \theta) \\ \beta_{j-k}, \beta_{j+l} &: k\text{-th lag or } l\text{-th lead variables} \end{aligned}$$

For example. you can model

$$\pi(\theta_j) \sim \text{normal}\left(\frac{\theta_{j-1} + \theta_{j+1}}{2}, \text{var} = 1\right)$$

PROC MCMC Supports a Variety of Random-Effects Models

- multiple effects (school, class, student, etc)
- nested or non-nested models
- linear or nonlinear
- corner-point constraint
- standard (normal, MVN, categorical, etc) and general prior distributions

Simple Binomial Random-Effects Model

A simple example:

$$\begin{aligned} y_i &\sim \text{binomial}(n_i, p_i) \\ p_i &\sim \text{beta}(a, b) \end{aligned}$$

```
random p ~ beta(a, b) subject = center;
model y ~ binomial(n, p);
```

The number of R.E. parameters is determined by the number of **unique values** in the SUBJECT= variable.

Parallel Similarity

Again, the syntax is similar in WinBUGS. The difference is in implicit indexing:

```
 y[] n[] center[]
  2   86     1
  2   69     2
  1   71     3
  1  113     4
  1  103     5
END
```

```
model {
   for(i in 1:5) {
       y[i] ~ dbin(p[i], n[i])
    }
   for(i in 1:5) {
      p[i] ~ beta(a, b)
   }
}
```

Repeats in Subjects

The nested indexing is handled internally by the procedure. For example, if there are repeats in `center`:

```
 y[] n[] center[]
  2   86      1
  2   69      2
  1   71      2
  1  113      3
  1  103      2
END
```

in WinBUGS:

```
model {
   for(i in 1:5) {
       y[i] ~ dbin(p[center[i]], n[i])
    }
   for(i in 1:3) {
      p[i] ~ beta(a, b)
   }
}
```

in PROC MCMC:

```
random p ~ beta(a, b) subject = center;
model y ~ binomial(n, p);
```

Pharmacokinetic Models

Computational challenges in fitting PK models are

- nonlinear models
  - DATA step language enables modeling flexibilities
- random-effects models
  - RANDOM statements
- differential equation solver, ODEs or piecewise ODEs
  - CALL ODE subroutine

CALL ODE

The CALL ODE subroutine solves a set of first-order ODEs, including piecewise DEs. The form is $\frac{dy}{dt} = f(t, y(t))$ over the subinterval $t \in [t_i, t_f]$ with the initial values $y(t_i) = y_0$.

```
CALL ODE("DeqFun", Soln, Init, ti, tf <, args>);
```

DeqFun : name of the subroutine function of a set of ODEs

Soln : solutions (can be numeric or an array)

Init : initial values of y

ti : initial time value of the subinterval t

tf : final time value of the subinterval t

args : input arguments to `DeqFun`

Define ODEs

The set of differential equations is specified in PROC FCMP:

```
PROC FCMP outlib=sasuser.funcs.ODE;
   SUBROUTINE DeqFun(t,y[*],dy[*], A1,A2,...,Ak);
   OUTARGS dy;
   dy[1] = -A1*y[1];
   dy[2] = A2*y[1]-Ak-1*y[2];
   ...
   endsub;
run;
```

outlib : location to store the objective function

t : the time variable

y : the w.r.t. variable

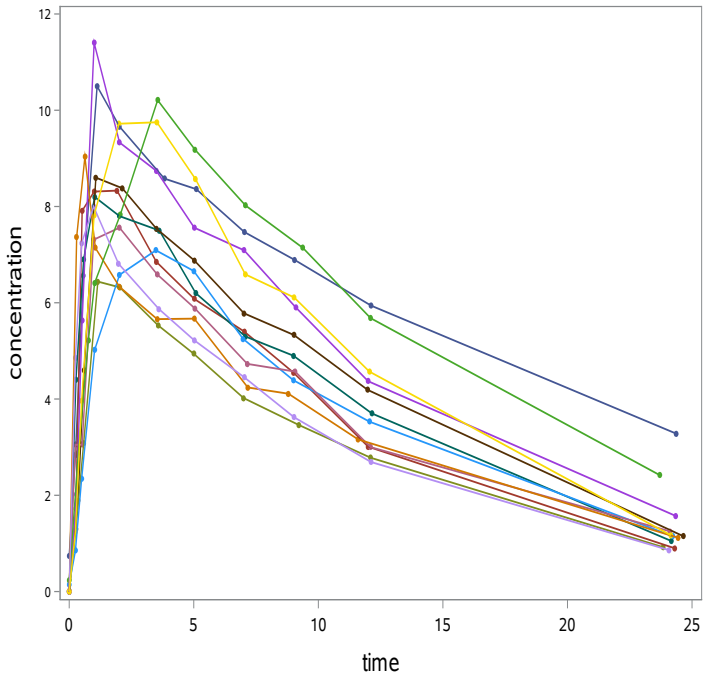
dy : DE function variable, must be declared as an OUTARGS

One-Compartment Model

A well-known case, which studied concentrations of theophylline in 12 subjects over a 25-hour period after oral administration. The differential equations and initial values are:

$$\begin{aligned} \frac{dA_0(t)}{dt} &= -K_a A_0(t) \\ \frac{dA(t)}{dt} &= K_a A_0(t) - K_e A(t) \\ A_0(t=0) &= x \\ A(t=0) &= 0 \end{aligned}$$

where K_a is the absorption rate, K_e is the elimination rate, and x is dose.



The statistical model is:

$$\begin{aligned} \mu_i(t) &= A_i(t)/Cl_i \\ y_i(t) &\sim \text{normal}(\mu_i(t), \sigma^2) \end{aligned}$$

where i is the subject index, $A_i(t)$ is the ODE solution at time t, and Cl is the clearance. Further,

$$\begin{aligned} CL_i &= \exp(\beta_1 + b_{i1}) \\ K_{a_i} &= \exp(\beta_2 + b_{i2}) \\ K_{e_i} &= \exp(\beta_3) \end{aligned}$$

where

$$\begin{aligned} \beta_1, \beta_2, \beta_3 &\sim \text{normal}(0, 100) \\ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} &\sim \text{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned}$$

Part of the data set:

```
data theoph;
   input subject time conc dose;
   datalines;
 1  0.00  0.74 4.02
 1  0.25  2.84 4.02
 1  0.57  6.57 4.02
 1  1.12 10.50 4.02
 1  2.02  9.66 4.02
 1  3.82  8.58 4.02
 1  5.10  8.36 4.02
 1  7.03  7.47 4.02
 1  9.05  6.89 4.02
 1 12.12  5.94 4.02
 1 24.37  3.28 4.02
 2  0.00  0.00 4.40
 2  0.27  1.72 4.40
 2  0.52  7.91 4.40
...
```

Define the set of ODE using PROC FCMP:

```
proc fcmp outlib=sasuser.funcs.PK;
   subroutine OneComp(t,y[*],dy[*],ka,ke);
   outargs dy;
   dy[1] = -ka*y[1];
   dy[2] = ka*y[1]-ke*y[2];
   endsub;
run;
```

with the ODEs:

$$\begin{aligned} \frac{dA_0(t)}{dt} &= -K_a A_0(t) \\ \frac{dA(t)}{dt} &= K_a A_0(t) - K_e A(t) \end{aligned}$$

```
options cmplib=sasuser.funcs;
proc mcmc data=theoph nmc=10000 seed=27 outpost=theoph0
   diag=none nthreads=8;
   array b[2];                ! two-dim random-effects
   array muB[2] (0 0);
   array cov[2,2];            ! cov matrix for b
   array S[2,2] (1 0 0 1);
   array init[2] dose 0;      ! A0(t=0) = x; A(t=0) = 0
   array sol[2];              ! solution matrix

   parms beta1 -3.22 beta2 0.47 beta3 -2.45 ;
   parms cov {0.03 0 0 0.4};
   parms s2y;
   prior beta: ~ normal(0, sd=100);
   prior cov ~ iwish(2, S);
   prior s2y  ~ igamma(shape=3, scale=2);

   random b ~ mvn(muB, cov) subject=subject;
   cl   = exp(beta1 + b1);          ! CL_i = exp(beta1 + b_i1)
   ka   = exp(beta2 + b2);          ! Ka_i = exp(beta2 + b_i2)
   ke   = exp(beta3);               ! Ke_i = exp(beta3 )
   v    = cl/ke;
   call ode('OneComp',sol,init,0,time,ka,ke);
   mu   = (sol[2]/v);               ! A_i(t) = sol[2]
   model conc ~ normal(mu,var=s2y);
run;
```

Prediction

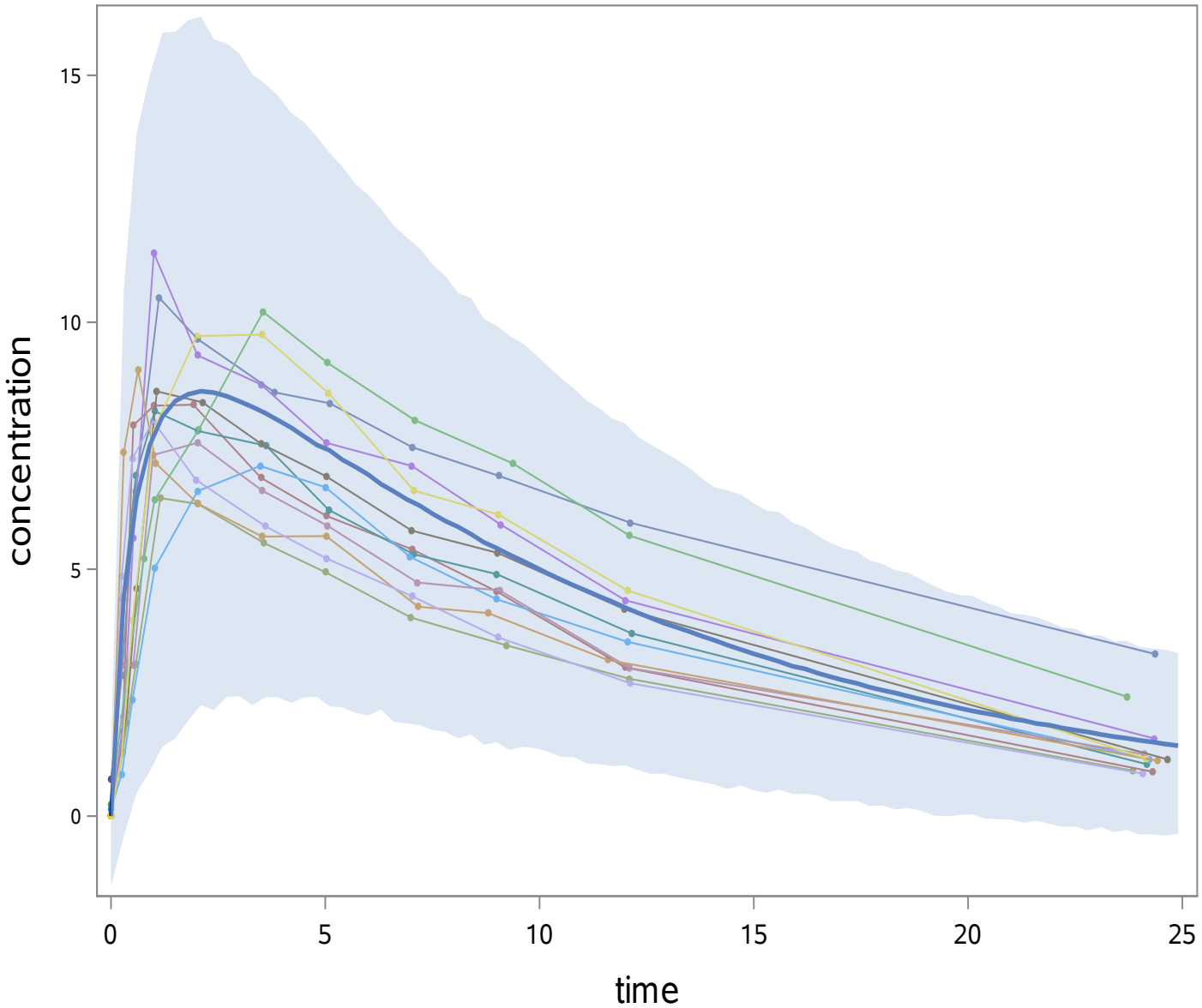
You can make prediction on a new patient with `dose = 4.7` over a period time:

```
data NewPatient;
   dose = 4.7;
   do time = 0 to 25 by 0.3;
      output;
      end;
   run;

proc mcmc ...;
   /* identical code */

   preddist outpred=NewOut covariates=NewPatient;
   run;
```

Predicted mean and 95% HPD intervals for a new patient who is given `dose = 4.7`:



Piecewise ODEs

The `CALL ODE` subroutine also solves piecewise differential equations. You

- specify the system of ODEs in PROC FCMP
- input a numerical array with interval boundaries
- specify an initial value function in PROC FCMP for initial values of the ODEs at different intervals (for example, solution to the first interval can become the initial value for the second interval).

```
CALL ODE("DeqFun", Soln, ., ti, tf, G1, G2, ... , Gk,
         ode_grid, "InitFun", H1, H2, ..., Hl <, ode_opt>);
```

For more information, refer to procedure documentation in the *SAS/STAT User's Guide*.

Missing Data in PROC MCMC

Treatment of missing data in PROC MCMC is straightforward.

- Missing values are random variables
- An additional Gibbs step is inserted in the sampling
- The objective is to obtain the joint posterior distribution conditional on observed data: $\pi(\theta, \mathbf{y}_{\text{mis}} | \mathbf{x}, \mathbf{y}_{\text{obs}})$
- The approach is model-based and capable of handling complex missing data scenarios.

Handling of Missing Values in PROC MCMC

The MODEL statement handles the estimation of all missing values:

MODEL *variable-list* $\sim$ distribution <options> ;

- The distribution is the usual likelihood function when the MODEL statement is applied to a response variable;
- It becomes a prior distribution for a covariate;

PROC MCMC

- identifies all missing values that are in *variable-list*
- creates a parameter for each missing value
- draws samples in a Gibbs fashion in simulation

- PROC MCMC models missing values only for MODEL statement variables. If there are missing values in y:

```
MODEL y ~ normal(mu, var=1);
```

Each missing value in y becomes a parameter.

- Records that contain missing values in other data set variables are discarded. If there are missing values in x:

```
mu = beta0 + beta1 * x;
MODEL y ~ normal(mu, var=1);
```

PROC MCMC does not model any missing values in x by default, unless you model it specifically:

```
model x ~ normal(alpha0 + beta1 * age, var=2);
mu = beta0 + beta1 * x;
MODEL y ~ normal(mu, var=1);
```

Modelling Missing Data in PROC MCMC

PROC MCMC handles all three types of missing data models:

- Missing Completely at Random (use option MISSING=CC)
- Missing at Random (PROC MCMC's default)
- Missing not at Random (model missing mechanism R)
  - selection model

$$f(r, y|x, \theta) \propto f(y|x, \alpha) \cdot f(r|y, x, \beta)$$

  - pattern-mixture model

$$f(r, y|x, \theta) \propto f(y|r, x, \delta) \cdot f(r|x, \gamma)$$

Modeling MNAR in PROC MCMC

You use two MODEL statements, one for the marginal model and one for the conditional model.

- In select model, you use one MODEL statement to model $f(y|x,\alpha)$, one to model $f(r|y,x,\beta)$
- In pattern-mixture model, you use one MODEL statement to model $f(r|x,\gamma)$, one to model $f(y|r,x,\delta)$

The marginal model must appear before the conditional model.

Example Data Set

- The data are based on a double-blind antidepressant clinical trial originally reported by Goldstein et al (2004).
- The Drug Information Association (DIA) working group on missing data have made this data set available at www.missingdata.org.uk.
- To avoid implications for marketed drugs, all patients who took active medication are grouped into a single `DRUG` group and only a subset of the original trial patients are included.
- There are 171 subjects in the data set, 88 in the control arm, and 83 in the active arm.

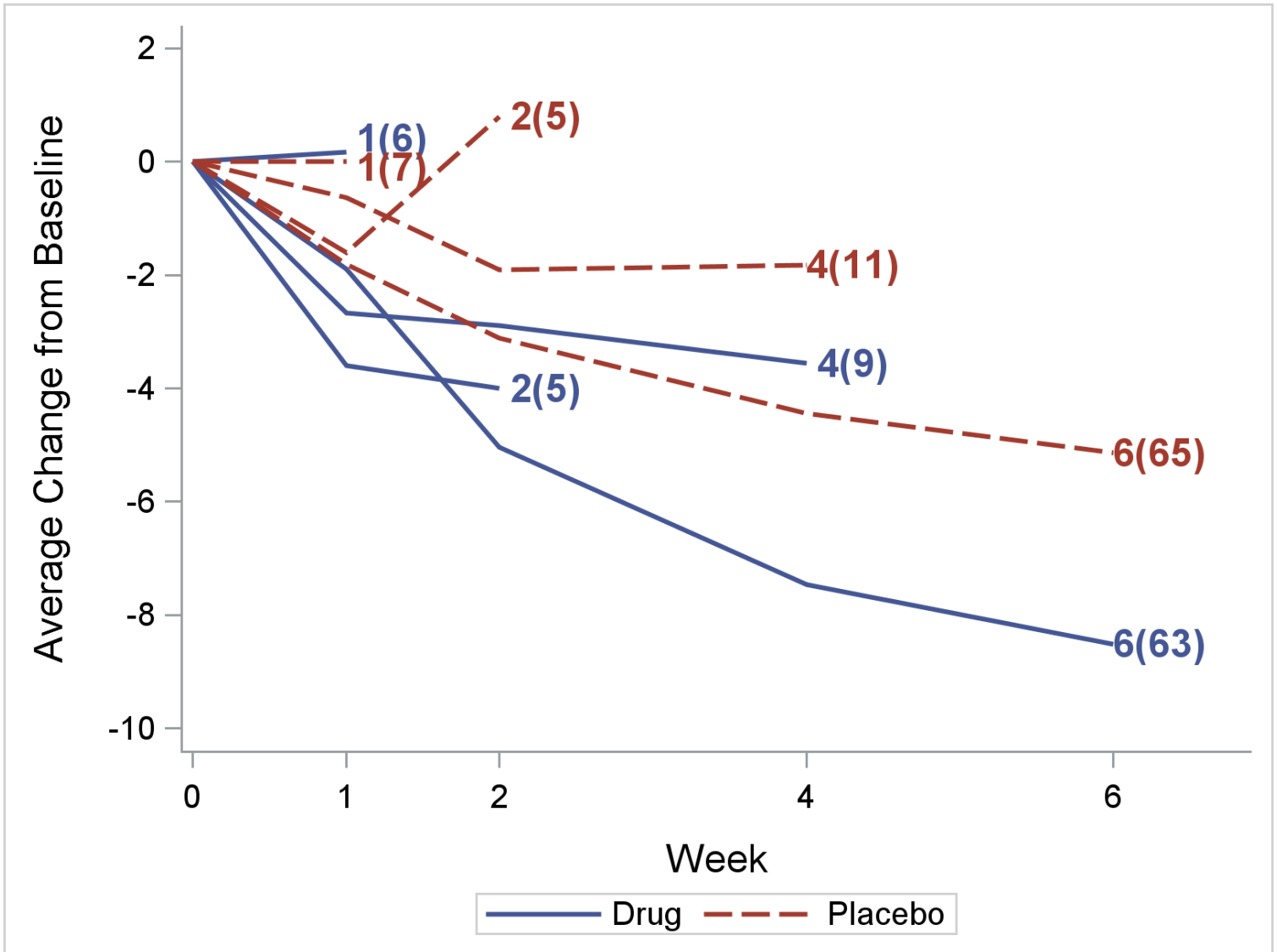
Variables in the Data Set

- `patient`: patient ID
- `baseval`: baseline assessment on the Hamilton 17-item rating scale for depression ($HAMD_{17}$, Hamilton 1960).
- `change1`–`change4`: change in $HAMD_{17}$ at weeks 1, 2, 4, and 6.
- `r1`–`r4`: missing data indicator for each of the `change` variables.
- `therapy`: treatment (`DRUG` vs `PLACEBO`)
- `poolinv`: blocking information (Groups formed by pooling investigator).
- `last`: week index to last non-missing `change` value. Patient's last visit week.
- `wkMax`: maximum number of weeks to be included in the analysis.

The first few observations of the `selection` data set:

```
data selection;
input PATIENT baseval change1-change4 r1-r4 THERAPY $ POOLINV $ last wkMax;
datalines;
 1503   32  -11  -12  -13  -15  0  0  0  0  DRUG      006   4  4
 1507   14   -3    0   -5   -9  0  0  0  0  PLACEBO   006   4  4
...
```

Average Mean Changes of $HAMD_{17}$ by Withdrawal Pattern



Data Characteristics

- Dropout probabilities appear to be correlated with the observed level of improvement (change in score).
- Patients failing to see improvement (flat or up-swinging lines), are more likely to withdraw.
- The probability of withdrawal *could* also depend on how they felt at the first unobserved visit - *the MNAR part of the model*.
- Fit a selection model:

$$f(\texttt{change}|\texttt{x}, \theta) \cdot f(\texttt{r}|\texttt{change}, \phi)$$

Outcome Model

For every subject i, $\texttt{change}_i = \{\texttt{change}_{ji}\}$ is modeled using a MVN($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}$), where $j = \{1, 2, 3, 4\}$ is the week index.

The mean variables, $\boldsymbol{\mu}_i = (\mu_{1i}, \mu_{2i}, \mu_{3i}, \mu_{4i})$, are modeled via:

$$\mu_{ji} = m_{kj} + \beta_j \cdot (\texttt{baseval-18}) + \gamma_l$$

where $k = \{1, 2\}$ indexes the treatment, l indexes pooling investigator.

The following prior distributions are used in the analysis:

$$\begin{aligned} \pi(m_{kj}, \beta_j, \gamma_l) &\propto 1 \\ \boldsymbol{\Sigma} &\sim \text{iWishart}(4, \boldsymbol{I}) \end{aligned}$$

The Selection Model

The selection model (Diggle-Kenward model) includes the previous and current (possibly missing) response variables for each week:

$$\begin{aligned} \mathtt{r}_{kj\,i} &\sim \text{binary}(q_{kj\,i}) \\ q_{kj\,i} &= \text{logistic}(\phi_{k1} + \phi_{k2} \cdot \text{change}_{(j-1)\,i} + \phi_{3k} \cdot \text{change}_{j\,i}) \end{aligned}$$

The parameters $\phi_{k\cdot}$ account for treatment effect in separate regression models. Flat prior is used:

$$\pi(\phi_{k\cdot}) \quad \propto 1$$

```
proc mcmc data=selection nmc=20000 seed=176 outpost=seleout;
   array Change[4] Change1-Change4;                    ! response
   array mu[4];                                        ! μ_i
   array Sigma[4,4];                                   ! Σ
   array S[4,4] (1 0 0 0, 0 1 0 0, 0 0 1 0, 0 0 0 1);! S = I
   array beta[4] ;                                     ! β_j
   array M[2,4] m1-m8;                                 ! m_kj
   array phi[2,3] phi1-phi6;                           ! φ_k.

   parms beta: 0 ;
   parms m1-m8 0;
   parms phi1-phi6 0;
   parms Sigma ;
   prior beta: m1-m8 phi: ~ general(0);                ! π(m_kj, β_j, φ_k) ∝ 1
   prior Sigma ~ iwish(4, S);                          ! π(Σ) = iWishart(4, S)

   /* outcome model */
   random gamma ~ general(0) subject=poolinv zero=first init=0;   ! π(γ_l) ∝ 1
   do j=1 to 4;
      if therapy eq "DRUG" then do;
         mu[j] = m[1,j] + gamma + beta[j]*(baseval-18); ! μ_{k = DRUG}j
      end; else do;
         mu[j] = m[2,j] + gamma + beta[j]*(baseval-18); ! μ_{k = PLACEBO}j
      end;
   end;
   model Change ~ mvn(mu, Sigma);                      ! likelihood
```

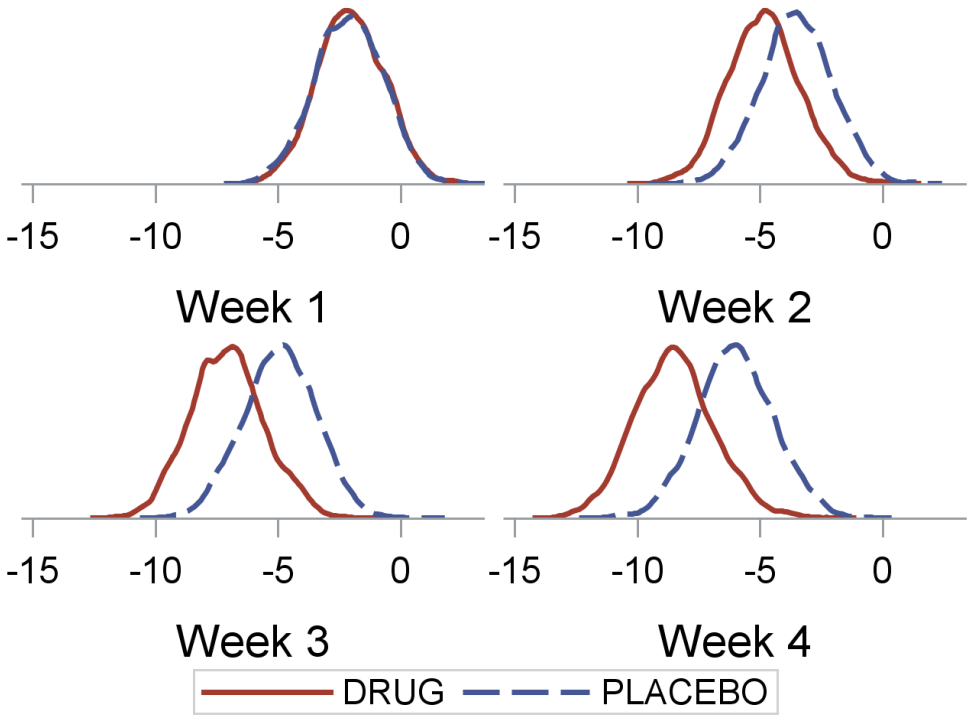
MCMC Code for the Selection Model

```
   /* selection mechanism */
   array r[4] r1-r4;                                    ! missing data indicator
   llike = 0;
   do j = 2[1] to wkMax
      if therapy eq "DRUG" then do;
         mn = phi[1,1] + phi[1,2] * change[j-1] + phi[1,3] * change[j];
         q = logistic(mn);                              ! q{k=DRUG}j
      end; else do;
         mn = phi[2,1] + phi[2,2] * change[j-1] + phi[2,3] * change[j];
         q = logistic(mn);                              ! q{k=PLACEBO}j
      end;
      llike = llike + lpdfbern(r[i], q);                ! accumulates binary
                                                        ! likelihood over weeks
   end;
   model r2 r3 r4 ~ general(llike);                     ! declares joint likelihood
run;
```

[1] Variable change1 doesn't contain any missing values, making r1 irrelevant to the analysis.

Outcome Model Estimates

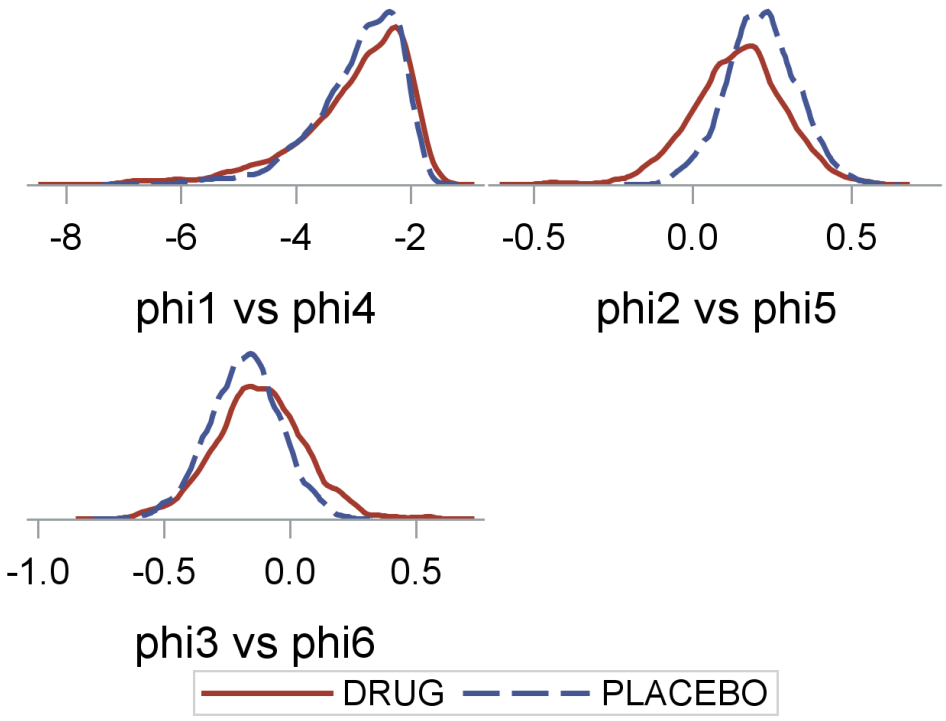
Comparison of posterior distributions of $m_{\text{drug},j}$ and $m_{\text{placebo},j}$ over the weeks:



- The treatment difference at week 1 is negligible.
- The difference becomes larger as the trial progresses, with the predicted score change for the DRUG group declining at a faster pace. The difference (mean difference is -2.42) is largest at the end of the trial.

Selection Model Estimates, When All are Estimated

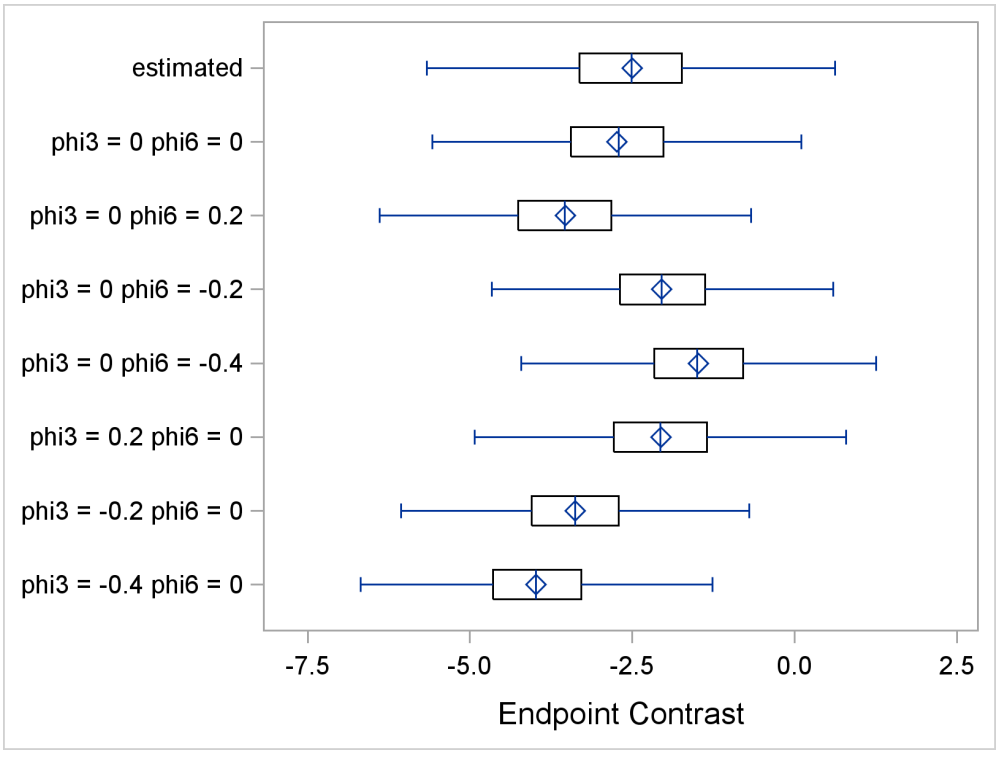
Posterior distributions of $\phi_{k\cdot}$, which model the change in the probability of dropouts given the score changes in the last and the current, potentially missing, week:



- $\phi_{\text{drug},2}$ (`phi2`) and $\phi_{\text{placebo},2}$ (`phi5`) are positive, suggesting that as the patient felt *worse* (increase in HAMD_{17} score) in their previous visit, they were *more* likely to dropout.
- $\phi_{\text{drug},2}$ (`phi3`) $\phi_{\text{placebo},2}$ (`phi6`) are negative, suggesting that patients were *less* likely to withdraw from the trial had they felt *worse* in the current week.

Sensitivity Analysis Fixing MNAR Parameter Values

The parameters in this complete model are poorly estimated. An idea is to fix the regression on the potentially unobserved values (`phi3` and `phi6`) and observe sensitivity to changing these. The estimated model (1st boxplot) produces similar point estimates (but larger s.d.) to the MAR model (2nd).



- when `phi3` $<$ `phi6`, boxplots shift to the left (3rd, 7th, and 8th). `DRUG` patients were more likely to drop out if they felt improvement in the current week. This results in stronger estimated treatment effect as the estimate is *corrected* for these missed patients.
- when `phi3` $>$ `phi6`, boxplots shift to the right (4th, 5th, and 6th), resulting in weaker treatment effect estimates.

Finishing Thoughts

Bayesian modeling is a key development area for SAS and we plan to continuously make improvement to the procedure. Some areas of keen interests include:

- distributed computing
- spatial inference
- approximation-based algorithms
- class variables
- automatic model selection
- wish list?

We always welcome your comments and feedback.

Additional Information

The current release: SAS/STAT®13.2, Revision 14w32.

The upcoming release: SAS/STAT®14.1, Revision 15w29.

You can use the following web resources:

`http://support.sas.com/documentation/onlinedoc/stat/index.html`

`http://support.sas.com/rnd/app/Bayesian/MCMC.html`

`http://support.sas.com/rnd/app/examples/STATwebexamples.html`

Or send me an email!

Copyrights