

# **SAS/STAT<sup>®</sup> 15.1**

## **User's Guide**

### **The HPFMM Procedure**

This document is an individual chapter from *SAS/STAT® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/STAT® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

### **SAS/STAT® 15.1 User's Guide**

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

**For a hard-copy book:** No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

**For a web download or e-book:** Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

**U.S. Government License Rights; Restricted Rights:** The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

# Chapter 55

## The HPFMM Procedure

### Contents

---

Overview: HPFMM Procedure . . . . .	<b>4426</b>
Basic Features . . . . .	4427
PROC HPFMM Contrasted with PROC FMM . . . . .	4428
Assumptions . . . . .	4429
Notation for the Finite Mixture Model . . . . .	4429
Homogeneous Mixtures . . . . .	4429
Special Mixtures . . . . .	4430
Getting Started: HPFMM Procedure . . . . .	<b>4430</b>
Mixture Modeling for Binomial Overdispersion: “Student,” Pearson, Beer, and Yeast . . . . .	4430
Modeling Zero-Inflation: Is It Better to Fish Poorly or Not to Have Fished at All? . . . . .	4436
Looking for Multiple Modes: Are Galaxies Clustered? . . . . .	4444
Comparison with Roeder’s Method . . . . .	4451
Syntax: HPFMM Procedure . . . . .	<b>4454</b>
PROC HPFMM Statement . . . . .	4454
BAYES Statement . . . . .	4466
BY Statement . . . . .	4476
CLASS Statement . . . . .	4476
FREQ Statement . . . . .	4477
ID Statement . . . . .	4477
MODEL Statement . . . . .	4477
Response Variable Options . . . . .	4479
Model Options . . . . .	4480
OUTPUT Statement . . . . .	4485
PERFORMANCE Statement . . . . .	4488
PROBMODEL Statement . . . . .	4488
RESTRICT Statement . . . . .	4490
WEIGHT Statement . . . . .	4492
Details: HPFMM Procedure . . . . .	<b>4492</b>
A Gentle Introduction to Finite Mixture Models . . . . .	4492
The Form of the Finite Mixture Model . . . . .	4492
Mixture Models Contrasted with Mixing and Mixed Models: Untangling the Terminology Web . . . . .	4493
Overdispersion . . . . .	4495
Log-Likelihood Functions for Response Distributions . . . . .	4495
Bayesian Analysis . . . . .	4502
Conjugate Sampling . . . . .	4502

Metropolis-Hastings Algorithm . . . . .	4503
Latent Variables via Data Augmentation . . . . .	4504
Prior Distributions . . . . .	4504
Parameterization of Model Effects . . . . .	4505
Computational Method . . . . .	4506
Multithreading . . . . .	4506
Choosing an Optimization Algorithm . . . . .	4507
First- or Second-Order Algorithms . . . . .	4507
Algorithm Descriptions . . . . .	4508
Output Data Set . . . . .	4510
Default Output . . . . .	4511
Performance Information . . . . .	4511
Model Information . . . . .	4511
Class Level Information . . . . .	4511
Number of Observations . . . . .	4511
Response Profile . . . . .	4512
Default Output for Maximum Likelihood . . . . .	4512
Default Output for Bayes Estimation . . . . .	4514
ODS Table Names . . . . .	4515
ODS Graphics . . . . .	4517
Examples: HPFMM Procedure . . . . .	<b>4517</b>
Example 55.1: Modeling Mixing Probabilities: All Mice Are Equal, but Some Mice Are More Equal Than Others . . . . .	4517
Example 55.2: The Usefulness of Custom Starting Values: When Do Cows Eat? . . .	4525
Example 55.3: Enforcing Homogeneity Constraints: Count and Dispersion—It Is All Over! . . . . .	4534
References . . . . .	<b>4539</b>

---

## Overview: HPFMM Procedure

The HPFMM procedure is a high-performance counterpart of the FMM procedure that fits statistical models to data for which the distribution of the response is a finite mixture of univariate distributions—that is, each response comes from one of several random univariate distributions that have unknown probabilities. You can use PROC HPFMM to model the component distributions in addition to the mixing probabilities. For more precise definitions and a discussion of similar but distinct modeling methodologies, see the section “[A Gentle Introduction to Finite Mixture Models](#)” on page 4492.

The HPFMM procedure is designed to fit finite mixtures of regression models or finite mixtures of generalized linear models in which the covariates and regression structure can be the same across components or can be different. You can fit finite mixture models by maximum likelihood or Bayesian methods. Note that classical statistical models are a special case of the finite mixture models in which the distribution of the data has only a single component.

PROC HPFMM runs in either single-machine mode or distributed mode.

**NOTE:** Distributed mode requires SAS High-Performance Statistics.

---

## Basic Features

The HPFMM procedure estimates the parameters in univariate finite mixture models and produces various statistics to evaluate parameters and model fit. The following list summarizes some basic features of the HPFMM procedure:

- maximum likelihood estimation for all models
- Markov chain Monte Carlo estimation for many models, including zero-inflated Poisson models
- many built-in link and distribution functions for modeling, including the beta, shifted  $t$ , Weibull, beta-binomial, and generalized Poisson distributions, in addition to many standard members of the exponential family of distributions
- specialized built-in mixture models such as the binomial cluster model (Morel and Nagaraj 1993; Morel and Neerchal 1997; Neerchal and Morel 1998)
- acceptance of multiple **MODEL** statements to build mixture models in which the model effects, distributions, or link functions vary across mixture components
- model-building syntax using **CLASS** and effect-based **MODEL** statements familiar from many other SAS/STAT procedures (for example, the GLM, GLIMMIX, and MIXED procedures)
- evaluation of sequences of mixture models when you specify ranges for the number of components
- simple syntax to impose linear equality and inequality constraints among parameters
- ability to model regression and classification effects in the mixing probabilities through the **PROB-MODEL** statement
- ability to incorporate full or partially known component membership into the analysis through the **PARTIAL=** option in the **PROC HPFMM** statement
- **OUTPUT** statement that produces a SAS data set with important statistics for interpreting mixture models, such as component log likelihoods and prior and posterior probabilities
- ability to add zero-inflation to any model
- output data set with posterior parameter values for the Markov chain
- multithreading and distributed computing for high-performance optimization and Monte Carlo sampling

The HPFMM procedure uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, “[Statistical Graphics Using ODS](#).” For specific information about the statistical graphics available with the HPFMM procedure, see the **PLOTS** options in the **PROC HPFMM** statement.

Because the HPFMM procedure is a high-performance analytical procedure, it also does the following:

- enables you to run in distributed mode on a cluster of machines that distribute the data and the computations

- enables you to run in single-machine mode on the server where SAS is installed
- exploits all the available cores and concurrent threads, regardless of execution mode

For more information, see the section “Processing Modes” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

---

## PROC HPFMM Contrasted with PROC FMM

For general contrasts between SAS high-performance analytical procedures and other SAS procedures, see the section “Common Features of SAS High-Performance Statistical Procedures” (Chapter 3, *SAS/STAT User’s Guide: High-Performance Procedures*).

The HPFMM procedure is somewhat distinct from other high-performance analytical procedures in being very nearly a twin of its counterpart, PROC FMM. You can fit the same kinds of models and get the same kinds of tabular, graphical, and data set results from PROC HPFMM as from PROC FMM. The main difference is that PROC HPFMM was developed primarily to work in a distributed environment, and PROC FMM primarily for a single (potentially multithreaded) host.

PROC HPFMM and PROC FMM have several differences because of their respective underlying technology:

- The **ORDER** option that specifies the sort order for the levels of **CLASS** variables is not available in the **PROC** statement of the HPFMM procedure. Instead the HPFMM procedure makes this option available in the **CLASS** statement.
- The **CLASS** statement in the HPFMM procedure provides many more options than the **CLASS** statement in the FMM procedure.
- The **PERFORMANCE** statement in the HPFMM procedure includes a superset of the options that are available in the **PERFORMANCE** statement in the FMM procedure.
- The **NOVAR** option in the **OUTPUT** statement in the FMM procedure is not available in the **OUTPUT** statement of the HPFMM procedure.

The **OUTPUT** statement in PROC HPFMM produces observationwise statistics. However, as is customary for SAS high-performance analytical procedures, PROC HPFMM’s **OUTPUT** statement does not by default include the input and **BY** variables in the output data set. This is to avoid data duplication for large data sets. In order to include any input or **BY** variables in the output data set, you must list these variables in the **ID** statement. Furthermore, PROC HPFMM’s **OUTPUT** statement includes the predicted values of the response variable if you do not specify any output statistics.

In contrast, when you request that the posterior sample be saved to a SAS data by specifying the **OUTPOST=** option in the **BAYES** statement, PROC HPFMM includes the **BY** variables in the data set.

## Assumptions

The HPFMM procedure makes the following assumptions in fitting statistical models:

- The number of components  $k$  in the finite mixture is known a priori and is not a parameter to be estimated.
- The parameters of the components are distinct a priori.
- The observations are uncorrelated.

## Notation for the Finite Mixture Model

The general expression for the finite mixture model fitted with the HPFMM procedure is as follows:

$$f(y) = \sum_{j=1}^k \pi_j(\mathbf{z}, \boldsymbol{\alpha}_j) p_j(y; \mathbf{x}'_j \boldsymbol{\beta}_j, \phi_j)$$

The number of components in the mixture is denoted as  $k$ . The mixture probabilities  $\pi_j$  can depend on regressor variables  $\mathbf{z}$  and parameters  $\boldsymbol{\alpha}_j$ . By default, the HPFMM procedure models these probabilities using a logit transform if  $k = 2$  and as a generalized logit model if  $k > 2$ . The component distributions  $p_j$  can also depend on regressor variables in  $\mathbf{x}_j$ , regression parameters  $\boldsymbol{\beta}_j$ , and possibly scale parameters  $\phi_j$ . Notice that the component distributions  $p_j$  are indexed by  $j$  since the distributions might belong to different families. For example, in a two-component model, you might model one component as a normal (Gaussian) variable and the second component as a variable with a  $t$  distribution with low degrees of freedom to manage overdispersion.

The mixture probabilities  $\pi_j$  satisfy  $\pi_j \geq 0$ , for all  $j$ , and

$$\sum_{j=1}^k \pi_j(\mathbf{z}, \boldsymbol{\alpha}_j) = 1$$

## Homogeneous Mixtures

If the component distributions are of the same distributional form, the mixture is called homogeneous. In most applications of homogeneous mixtures, the mixing probabilities do not depend on regression parameters. The general model then simplifies to

$$f(y) = \sum_{j=1}^k \pi_j p(y; \mathbf{x}'_j \boldsymbol{\beta}_j, \phi_j)$$

Since the component distributions depend on regression parameters  $\beta_j$ , this model is known as a homogeneous regression mixture. A homogeneous regression mixture assumes that the regression effects are the same across the components, although the HPFMM procedure does not impose such a restriction. If the component distributions do not contain regression effects, the model

$$f(y) = \sum_{j=1}^k \pi_j p(y; \mu_j, \phi_j)$$

is *the* homogeneous mixture model. A classical case is the estimation of a continuous density as a  $k$ -component mixture of normal distributions.

## Special Mixtures

The HPFMM procedure enables you to fit several special mixture models. The Morel-Neerchal binomial cluster model (Morel and Nagaraj 1993; Morel and Neerchal 1997; Neerchal and Morel 1998) is a mixture of binomial distributions in which the success probabilities depend on the mixing probabilities.

Zero-inflated count models are obtained as two-component mixtures where one component is a classical count model—such as the Poisson or negative binomial model—and the other component is a distribution that is concentrated at zero. If the nondegenerate part of this special mixture is a zero-truncated model, the resulting two-component mixture is known as a hurdle model (Cameron and Trivedi 1998).

---

## Getting Started: HPFMM Procedure

---

### Mixture Modeling for Binomial Overdispersion: “Student,” Pearson, Beer, and Yeast

The following example demonstrates how you can model a complicated, two-component binomial mixture distribution, either with maximum likelihood or with Bayesian methods, with a few simple PROC HPFMM statements.

William Sealy Gosset, a chemist at the Arthur Guinness Son and Company brewery in Dublin, joined the statistical laboratory of Karl Pearson in 1906–1907 to study statistics. At first Gosset—who published all but one paper under the pseudonym “Student” because his employer forbade publications by employees after a co-worker had disclosed trade secrets—worked on the Poisson limit to the binomial distribution, using haemocytometer yeast cell counts. Gosset’s interest in studying small-sample (and limit) problems was motivated by the small sample sizes he typically saw in his work at the brewery.

Subsequently, Gosset’s yeast count data have been examined and revisited by many authors. In 1915, Karl Pearson undertook his own examination and realized that the variability in “Student’s” data exceeded that consistent with a Poisson distribution. Pearson (1915) bemoans the fact that if this were so, “it is certainly most unfortunate that such material should have been selected to illustrate Poisson’s limit to the binomial.”

Using a count of Gosset’s yeast cell counts on the 400 squares of a haemocytometer (Table 55.1), Pearson argues that a mixture process would explain the heterogeneity (beyond the Poisson).



**Table 55.1** "Student's" Yeast Cell Counts

Number of Cells	0	1	2	3	4	5
Frequency	213	128	37	18	3	1

Pearson fits various models to these data, chief among them a mixture of two binomial series

$$\nu_1(p_1 + q_1)^\theta + \nu_2(p_2 + q_2)^\theta$$

where  $\theta$  is real-valued and thus the binomial series expands to

$$(p + q)^\theta = \sum_{k=0}^{\infty} \frac{\Gamma(\theta + 1)}{\Gamma(k + 1)\Gamma(\theta - k + 1)} p^k q^{\theta-k}$$

Pearson's fitted model has  $\theta = 4.89997$ ,  $\nu_1 = 356.986$ ,  $\nu_2 = 43.014$  (corresponding to a mixing proportion of  $356.986/(43.014 + 356.986) = 0.892$ ), and estimated success probabilities in the binomial components of 0.1017 and 0.4514, respectively. The success probabilities indicate that although the data have about a 90% chance of coming from a distribution with small success probability of about 0.1, there is a 10% chance of coming from a distribution with a much larger success probability of about 0.45.

If  $\theta$  is an integer, the binomial series is the cumulative mass function of a binomial random variable. The value of  $\theta$  suggests that a suitable model for these data could also be constructed as a two-component mixture of binomial random variables as follows:

$$f(y) = \pi \text{ binomial}(5, \mu_1) + (1 - \pi) \text{ binomial}(5, \mu_2)$$

The binomial sample size  $n=5$  is suggested by Pearson's estimate of  $\theta = 4.89997$  and the fact that the largest cell count in Table 55.1 is 5.

The following DATA step creates a SAS data set from the data in Table 55.1.

```
data yeast;
  input count f;
  n = 5;
  datalines;
  0      213
  1      128
  2       37
  3       18
  4        3
  5        1
;
```

The two-component binomial model is fit with the HPFMM procedure with the following statements:

```
proc hpfmm data=yeast;
  model count/n = / k=2;
  freq f;
run;
```

Because the events/trials syntax is used in the **MODEL** statement, PROC HPFMM defaults to the binomial distribution. The **K=2** option specifies that the number of components is fixed and known to be two. The **FREQ** statement indicates that the data are grouped; for example, the first observation represents 213 squares on the haemocytometer where no yeast cells were found.

The “Model Information” and “Number of Observations” tables in Figure 55.1 convey that the fitted model is a two-component homogeneous binomial mixture with a logit link function. The mixture is *homogeneous* because there are no model effects in the **MODEL** statement and because both component distributions belong to the same distributional family. By default, PROC HPFMM estimates the model parameters by maximum likelihood.

Although only six observations are read from the data set, the data represent 400 observations (squares on the haemocytometer). Since a constant binomial sample size of 5 is assumed, the data represent 273 successes (finding a yeast cell) out of 2,000 Bernoulli trials.

**Figure 55.1** Model Information for Yeast Cell Model

The HPFMM Procedure	
Model Information	
Data Set	WORK.YEAST
Response Variable (Events)	count
Response Variable (Trials)	n
Frequency Variable	f
Type of Model	Homogeneous Mixture
Distribution	Binomial
Components	2
Link Function	Logit
Estimation Method	Maximum Likelihood

Number of Observations Read	6
Number of Observations Used	6
Sum of Frequencies Read	400
Sum of Frequencies Used	400
Number of Events	273
Number of Trials	2000

The estimated intercepts (on the logit scale) for the two binomial means are  $-2.2316$  and  $-0.2974$ , respectively. These values correspond to binomial success probabilities of 0.09695 and 0.4262, respectively (Figure 55.2). The two components mix with probabilities 0.8799 and  $1 - 0.8799 = 0.1201$ . These values are generally close to the values found by Pearson (1915) using infinite binomial series instead of binomial mass functions.

**Figure 55.2** Maximum Likelihood Estimates

Parameter Estimates for Binomial Model						
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z	Inverse Linked Estimate
1	Intercept	-2.2316	0.1522	-14.66	<.0001	0.09695
2	Intercept	-0.2974	0.3655	-0.81	0.4158	0.4262

**Figure 55.2** *continued*

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	Logit(Prob)	Linked Scale		
			Standard Error	z Value	Pr >  z
1	0.8799	1.9913	0.5725	3.48	0.0005
2	0.1201	-1.9913			

To obtain fitted values and other observationwise statistics under the stipulated two-component model, you can add the **OUTPUT** statement to the previous PROC HPFMM run. The following statements request componentwise predicted values and the posterior probabilities:

```
proc hpfmm data=yeast;
  model count/n = / k=2;
  freq f;
  id f n;
  output out=hpfmmtout pred(components) posterior;
run;
data hpfmmtout;
  set hpfmmtout;
  PredCount_1 = post_1 * f;
  PredCount_2 = post_2 * f;
run;
proc print data=hpfmmtout;
run;
```

The DATA step following the PROC HPFMM step computes the predicted cell counts in each component (Figure 55.3). Note that the predicted means in the components, 0.48476 and 2.13099, are close to the values determined by Pearson (0.4983 and 2.2118), as are the predicted cell counts.

**Figure 55.3** Predicted Cell Counts

Obs	f	n	Pred_1	Pred_2	Post_1	Post_2	PredCount_1	PredCount_2
1	213	5	0.096951	0.42620	0.98606	0.01394	210.030	2.9698
2	128	5	0.096951	0.42620	0.91089	0.08911	116.594	11.4058
3	37	5	0.096951	0.42620	0.59638	0.40362	22.066	14.9341
4	18	5	0.096951	0.42620	0.17598	0.82402	3.168	14.8323
5	3	5	0.096951	0.42620	0.02994	0.97006	0.090	2.9102
6	1	5	0.096951	0.42620	0.00444	0.99556	0.004	0.9956

Gosset, who was interested in small-sample statistical problems, investigated the use of prior knowledge in mathematical-statistical analysis—for example, deriving the sampling distribution of the correlation coefficient after having assumed a uniform prior distribution for the coefficient in the population (Aldrich 1997). Pearson also was not opposed to using prior information, especially uniform priors that reflect “equal distribution of ignorance.” Fisher, on the other hand, would not have any of it: the best estimator in his opinion is obtained by a criterion that is absolutely independent of prior assumptions about probabilities of particular values. He objected to the insinuation that his derivations in the work on the correlation were deduced from Bayes theorem (Fisher 1921).

The preceding analysis of the yeast cell count data uses maximum likelihood methods that are free of prior assumptions. The following analysis takes instead a Bayesian approach, assuming a beta prior distribution for the binomial success probabilities and a uniform prior distribution for the mixing probabilities. The changes from the previous run of PROC HPFMM are the addition of the ODS GRAPHICS, **PERFORMANCE**, and **BAYES** statements and the **SEED=12345** option.

```
ods graphics on;
proc hpfmm data=yeast seed=12345;
  model count/n = / k=2;
  freq f;
  performance nthreads=2;
  bayes;
run;
ods graphics off;
```

When ODS Graphics is enabled, PROC HPFMM produces diagnostic trace plots for the posterior samples. Bayesian analyses are sensitive to the random number seed and thread count; the **SEED=** and **NTHREADS=** options in the **PERFORMANCE** statement ensure consistent results for the purposes of this example. The **SEED=12345** option in the **PROC HPFMM** statement determines the random number seed for the random number generator that the analysis used. The **NTHREADS=2** option in the **PERFORMANCE** statement sets the number of threads to be used by the procedure to two. The **BAYES** statement requests a Bayesian analysis.

The “Bayes Information” table in Figure 55.4 provides basic information about the Markov chain Monte Carlo sampler. Because the model is a homogeneous mixture, the HPFMM procedure applies an efficient conjugate sampling algorithm with a posterior sample size of 10,000 samples after a burn-in size of 2,000 samples. The “Prior Distributions” table displays the prior distribution for each parameter along with its mean and variance and the initial value in the chain. Notice that in this situation all three prior distributions reduce to a uniform distribution on (0, 1).

**Figure 55.4** Basic Information about MCMC Sampler

The HPFMM Procedure					
Bayes Information					
Sampling Algorithm		Conjugate			
Data Augmentation		Latent Variable			
Initial Values of Chain		Data Based			
Burn-In Size		2000			
MC Sample Size		10000			
MC Thinning		1			
Parameters in Sampling		3			
Mean Function Parameters		2			
Scale Parameters		0			
Mixing Prob Parameters		1			
Prior Distributions					
Component	Parameter	Distribution	Mean	Variance	Initial Value
1	Success Probability	Beta(1, 1)	0.5000	0.08333	0.1365
2	Success Probability	Beta(1, 1)	0.5000	0.08333	0.1365
1	Probability	Dirichlet(1, 1)	0.5000	0.08333	0.6180

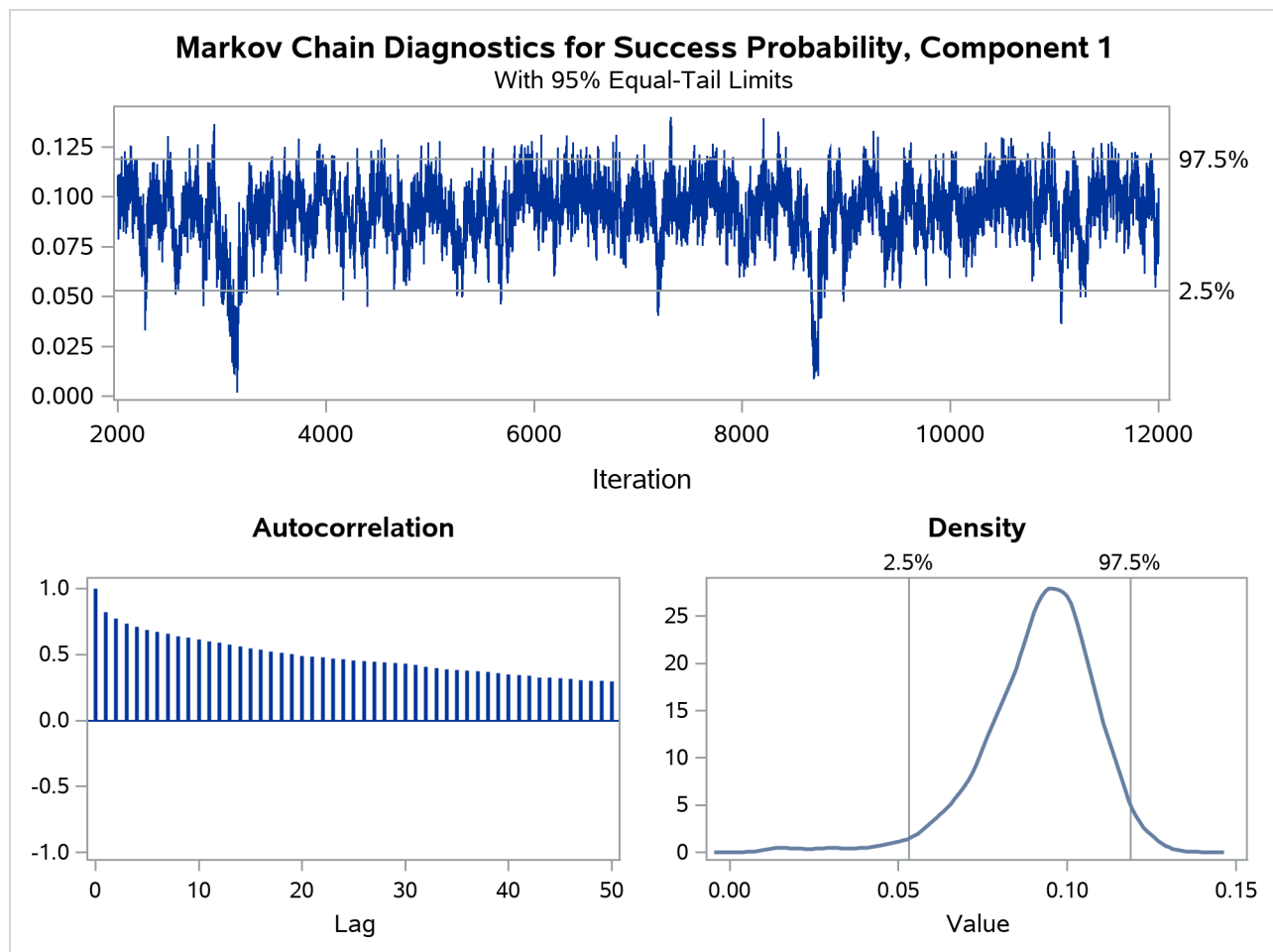
The HPFMM procedure produces a log note for this model, indicating that the sampled quantities are not the linear predictors on the logit scale, but are the actual population parameters (on the data scale):

**NOTE: Bayesian results for this model (no regressor variables, non-identity link) are displayed on the data scale, not the linked scale. You can obtain results on the linked (=linear) scale by requesting a Metropolis-Hastings sampling algorithm.**

The trace panel for the success probability in the first binomial component is shown in Figure 55.5. Note that the first component in this Bayesian analysis corresponds to the second component in the MLE analysis. The graphics in this panel can be used to diagnose the convergence of the Markov chain. If the chain has not converged, inferences cannot be made based on quantities derived from the chain. You generally look for the following:

- a smooth unimodal distribution of the posterior estimates in the density plot displayed on the lower right
- good mixing of the posterior samples in the trace plot at the top of the panel (good mixing is indicated when the trace traverses the support of the distribution and appears to have reached a stationary distribution)

**Figure 55.5** Trace Panel for Success Probability in First Component



The autocorrelation plot in [Figure 55.5](#) shows fairly high and sustained autocorrelation among the posterior estimates. While this is generally not a problem, you can affect the degree of autocorrelation among the posterior estimates by running a longer chain and thinning the posterior estimates; see the **NMC=** and **THIN=** options in the **BAYES** statement.

Both the trace plot and the density plot in [Figure 55.5](#) are indications of successful convergence.

[Figure 55.6](#) reports selected results that summarize the 10,000 posterior samples. The arithmetic means of the success probabilities in the two components are 0.0917 and 0.3974, respectively. The posterior mean of the mixing probability is 0.8312. These values are similar to the maximum likelihood parameter estimates in [Figure 55.2](#) (after swapping components).

**Figure 55.6** Summaries for Posterior Estimates

Posterior Summaries					Percentiles		
Component	Parameter	N	Mean	Standard Deviation	25	50	75
1	Success Probability	10000	0.0917	0.0168	0.0830	0.0938	0.1027
2	Success Probability	10000	0.3974	0.0871	0.3379	0.3957	0.4556
1	Probability	10000	0.8312	0.1045	0.7986	0.8578	0.8984

Posterior Intervals					Equal-Tail Interval		HPD Interval
Component	Parameter	Alpha					
1	Success Probability	0.050	0.0530	0.1187	0.0585	0.1212	
2	Success Probability	0.050	0.2272	0.5722	0.2343	0.5780	
1	Probability	0.050	0.5464	0.9454	0.6325	0.9642	

Note that the standard errors in [Figure 55.2](#) are not comparable to those in [Figure 55.6](#), since the standard errors for the MLEs are expressed on the logit scale and the Bayes estimates are expressed on the data scale. You can add the **METROPOLIS** option in the **BAYES** statement to sample the quantities on the logit scale.

The “Posterior Intervals” table in [Figure 55.6](#) displays 95% credible intervals (equal-tail intervals and intervals of highest posterior density). It can be concluded that the component with the higher success probability contributes less than 40% to the process.

## Modeling Zero-Inflation: Is It Better to Fish Poorly or Not to Have Fished at All?

The following example shows how you can use PROC HPFMM to model data with more zero values than expected.

Many count data show an excess of zeros relative to the frequency of zeros expected under a reference model. An excess of zeros leads to overdispersion since the process is more variable than a standard count data model. Different mechanisms can lead to excess zeros. For example, suppose that the data are generated from two processes with different distribution functions—one process generates the zero counts, and the other process generates nonzero counts. In the vernacular of Cameron and Trivedi (1998), such a model is called a *hurdle* model. With a certain probability—the probability of a nonzero count—a hurdle is crossed, and events are being generated. Hurdle models are useful, for example, to model the number of doctor visits

per year. Once the decision to see a doctor has been made—the hurdle has been overcome—a certain number of visits follow.

Hurdle models are closely related to zero-inflated models. Both can be expressed as two-component mixtures in which one component has a degenerate distribution at zero and the other component is a count model. In a hurdle model, the count model follows a zero-truncated distribution. In a zero-inflated model, the count model has a nonzero probability of generating zeros. Formally, a zero-inflated model can be written as

$$\Pr(Y = y) = \pi p_1 + (1 - \pi) p_2(y, \mu)$$

$$p_1 = \begin{cases} 1 & y = 0 \\ 0 & \text{otherwise} \end{cases}$$

where  $p_2(y, \mu)$  is a standard count model with mean  $\mu$  and support  $y \in \{0, 1, 2, \dots\}$ .

The following data illustrates the use of a zero-inflated model. In a survey of park attendees, randomly selected individuals were asked about the number of fish they caught in the last six months. Along with that count, the gender and age of each sampled individual was recorded. The following DATA step displays the data for the analysis:

```
data catch;
  input gender $ age count @@;
  datalines;
  F 54 18 M 37 0 F 48 12 M 27 0
  M 55 0 M 32 0 F 49 12 F 45 11
  M 39 0 F 34 1 F 50 0 M 52 4
  M 33 0 M 32 0 F 23 1 F 17 0
  F 44 5 M 44 0 F 26 0 F 30 0
  F 38 0 F 38 0 F 52 18 M 23 1
  F 23 0 M 32 0 F 33 3 M 26 0
  F 46 8 M 45 5 M 51 10 F 48 5
  F 31 2 F 25 1 M 22 0 M 41 0
  M 19 0 M 23 0 M 31 1 M 17 0
  F 21 0 F 44 7 M 28 0 M 47 3
  M 23 0 F 29 3 F 24 0 M 34 1
  F 19 0 F 35 2 M 39 0 M 43 6
  ;
```

At first glance, the prevalence of zeros in the DATA set is apparent. Many park attendees did not catch any fish. These zero counts are made up of two populations: attendees who do not fish and attendees who fish poorly. A zero-inflation mechanism thus appears reasonable for this application since a zero count can be produced by two separate distributions.

The following statements fit a standard Poisson regression model to these data. A common intercept is assumed for men and women, and the regression slope varies with gender.

```
proc hpfit data=catch;
  class gender;
  model count = gender*age / dist=Poisson;
run;
```

Figure 55.7 displays information about the model and data set. The “Model Information” table conveys that the model is a single-component Poisson model (a Poisson GLM) and that parameters are estimated by maximum likelihood. There are two levels in the CLASS variable gender, with females preceding males.

**Figure 55.7** Model Information and Class Levels in Poisson Regression

The HPFMM Procedure	
Model Information	
Data Set	WORK.CATCH
Response Variable	count
Type of Model	Generalized Linear (GLM)
Distribution	Poisson
Components	1
Link Function	Log
Estimation Method	Maximum Likelihood
Class Level Information	
Class	Levels Values
gender	2 F M
Number of Observations Read 52	
Number of Observations Used 52	

The “Fit Statistics” and “Parameter Estimates” tables from the maximum likelihood estimation of the Poisson GLM are shown in Figure 55.8. If the model is not overdispersed, the Pearson statistic should roughly equal the number of observations in the data set minus the number of parameters. With  $n=52$ , there is evidence of overdispersion in these data.

**Figure 55.8** Fit Results in Poisson Regression

Fit Statistics					
-2 Log Likelihood		182.7			
AIC (Smaller is Better)		188.7			
AICC (Smaller is Better)		189.2			
BIC (Smaller is Better)		194.6			
Pearson Statistic		85.9573			

Parameter Estimates for Poisson Model					
Effect	gender	Estimate	Standard Error	z Value	Pr >  z
Intercept		-3.9811	0.5439	-7.32	<.0001
age*gender	F	0.1278	0.01149	11.12	<.0001
age*gender	M	0.1044	0.01224	8.53	<.0001

Suppose that the cause of overdispersion is zero-inflation of the count data. The following statements fit a zero-inflated Poisson model.

```
proc hpfmm data=catch;
  class gender;
  model count = gender*age / dist=Poisson ;
  model      +           / dist=Constant;
run;
```



There are two **MODEL** statements, one for each component of the mixture. Because the distributions are different for the components, you cannot specify the mixture model with a single **MODEL** statement. The first **MODEL** statement identifies the response variable for the model (count) and defines a Poisson model with intercept and gender-specific slopes. The second **MODEL** statement uses the continuation operator (“+”) and adds a model with a degenerate distribution by using **DIST=CONSTANT**. Because the mass of the constant is placed by default at zero, the second **MODEL** statement adds a zero-inflation component to the model. It is sufficient to specify the response variable in one of the **MODEL** statements; you use the “=” sign in that statement to separate the response variable from the model effects.

Figure 55.9 displays the “Model Information” and “Optimization Information” tables for this run of the HPFMM procedure. The model is now identified as a zero-inflated Poisson (ZIP) model with two components, and the parameters continue to be estimated by maximum likelihood. The “Optimization Information” table shows that there are four parameters in the optimization (compared to three parameters in the Poisson GLM model). The four parameters correspond to three parameters in the mean function (intercept and two gender-specific slopes) and the mixing probability.

**Figure 55.9** Model and Optimization Information in the ZIP Model

The HPFMM Procedure	
Model Information	
Data Set	WORK.CATCH
Response Variable	count
Type of Model	Zero-inflated Poisson
Components	2
Estimation Method	Maximum Likelihood
Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	4
Mean Function Parameters	3
Scale Parameters	0
Mixing Prob Parameters	1

Results from fitting the ZIP model by maximum likelihood are shown in Figure 55.10. The  $-2$  log likelihood and the information criteria suggest a much-improved fit over the single-component Poisson model (compare Figure 55.10 to Figure 55.8). The Pearson statistic is reduced by factor 2 compared to the Poisson model and suggests a better fit than the standard Poisson model.

**Figure 55.10** Maximum Likelihood Results for the ZIP model

Fit Statistics	
-2 Log Likelihood	145.6
AIC (Smaller is Better)	153.6
AICC (Smaller is Better)	154.5
BIC (Smaller is Better)	161.4
Pearson Statistic	43.4467
Effective Parameters	4
Effective Components	2

Figure 55.10 continued

Parameter Estimates for Poisson Model						
Component	Effect	gender	Estimate	Standard Error	z Value	Pr >  z
1	Intercept		-3.5215	0.6448	-5.46	<.0001
1	age*gender	F	0.1216	0.01344	9.04	<.0001
1	age*gender	M	0.1056	0.01394	7.58	<.0001

Parameter Estimates for Mixing Probabilities						
Component	Mixing Probability	Logit(Prob)	Linked Scale			
			Standard Error	z Value	Pr >  z	
1	0.6972	0.8342	0.4768	1.75	0.0802	
2	0.3028	-0.8342				

The number of effective parameters and components shown in Figure 55.8 equals the values from Figure 55.9. This is not always the case because components can collapse (for example, when the mixing probability approaches zero or when two components have identical parameter estimates). In this example, both components and all four parameters are identifiable. The Poisson regression and the zero process mix, with a probability of approximately 0.6972 attributed to the Poisson component.

The HPFMM procedure enables you to fit some mixture models by Bayesian techniques. The following statements add the **BAYES** statement to the previous PROC HPFMM statements:

```
proc hpfmm data=catch seed=12345;
  class gender;
  model count = gender*age / dist=Poisson;
  model      +           / dist=constant;
  performance nthreads=2;
  bayes;
run;
```

The “Model Information” table indicates that the model parameters are estimated by Markov chain Monte Carlo techniques, and it displays the random number seed (Figure 55.11). This is useful if you did not specify a seed to identify the seed value that reproduces the current analysis. The “Bayes Information” table provides basic information about the Monte Carlo sampling scheme. The sampling method uses a data augmentation scheme to impute component membership and then the Gamerman (1997) algorithm to sample the component-specific parameters. The 2,000 burn-in samples are followed by 10,000 Monte Carlo samples without thinning.

Figure 55.11 Model, Bayes, and Prior Information in the ZIP Model

The HPFMM Procedure	
Model Information	
Data Set	WORK.CATCH
Response Variable	count
Type of Model	Zero-inflated Poisson
Components	2
Estimation Method	Markov Chain Monte Carlo
Random Number Seed	12345

**Figure 55.11** *continued*

Bayes Information						
Sampling Algorithm		Gamerman				
Data Augmentation		Latent Variable				
Initial Values of Chain		ML Estimates				
Burn-In Size		2000				
MC Sample Size		10000				
MC Thinning		1				
Parameters in Sampling		4				
Mean Function Parameters		3				
Scale Parameters		0				
Mixing Prob Parameters		1				

Prior Distributions						
Component	Effect	gender	Distribution	Mean	Variance	Initial Value
1	Intercept		Normal(0, 1000)	0	1000.00	-3.5215
1	age*gender	F	Normal(0, 1000)	0	1000.00	0.1216
1	age*gender	M	Normal(0, 1000)	0	1000.00	0.1056
1	Probability		Dirichlet(1, 1)	0.5000	0.08333	0.6972

The “Prior Distributions” table identifies the prior distributions, their parameters for the sampled quantities, and their initial values. The prior distribution of parameters associated with model effects is a normal distribution with mean 0 and variance 1,000. The prior distribution for the mixing probability is a Dirichlet(1,1), which is identical to a uniform distribution (Figure 55.11). Since the second mixture component is a degeneracy at zero with no associated parameters, it does not appear in the “Prior Distributions” table in Figure 55.11.

Figure 55.12 displays descriptive statistics about the 10,000 posterior samples. Recall from Figure 55.10 that the maximum likelihood estimates were  $-3.5215$ ,  $0.1216$ ,  $0.1056$ , and  $0.6972$ , respectively. With this choice of prior, the means of the posterior samples are generally close to the MLEs in this example. The “Posterior Intervals” table displays 95% intervals of equal-tail probability and 95% intervals of highest posterior density (HPD) intervals.

**Figure 55.12** Posterior Summaries and Intervals in the ZIP Model

Posterior Summaries								
Component	Effect	gender	N	Mean	Standard Deviation	Percentiles		
						25	50	75
1	Intercept		10000	-3.5456	0.6463	-3.9822	-3.5286	-3.1082
1	age*gender	F	10000	0.1219	0.0135	0.1129	0.1216	0.1310
1	age*gender	M	10000	0.1057	0.0140	0.0962	0.1053	0.1148
1	Probability		10000	0.6923	0.0950	0.6280	0.6960	0.7589

Posterior Intervals							
Component	Effect	gender	Alpha	Equal-Tail Interval		HPD Interval	
1	Intercept		0.050	-4.8615	-2.3246	-4.8979	-2.3808
1	age*gender	F	0.050	0.0955	0.1490	0.0963	0.1495
1	age*gender	M	0.050	0.0784	0.1338	0.0786	0.1339
1	Probability		0.050	0.4999	0.8683	0.5139	0.8799

You can generate trace plots for the posterior parameter estimates by enabling ODS Graphics:

```
ods graphics on;
ods select TADPanel;
proc hpfmm data=catch seed=12345;
  class gender;
  model count = gender*age / dist=Poisson;
  model      +           / dist=constant;
  performance nthreads=2;
  bayes;
run;
ods graphics off;
```

A separate trace panel is produced for each sampled parameter, and the panels for the gender-specific slopes are shown in Figure 55.13. There is good mixing in the chains: the modest autocorrelation that diminishes after about 10 successive samples. By default, the HPFMM procedure transfers the credible intervals for each parameter from the “Posterior Intervals” table to the trace plot and the density plot in the trace panel.

**Figure 55.13** Trace Panels for Gender-Specific Slopes

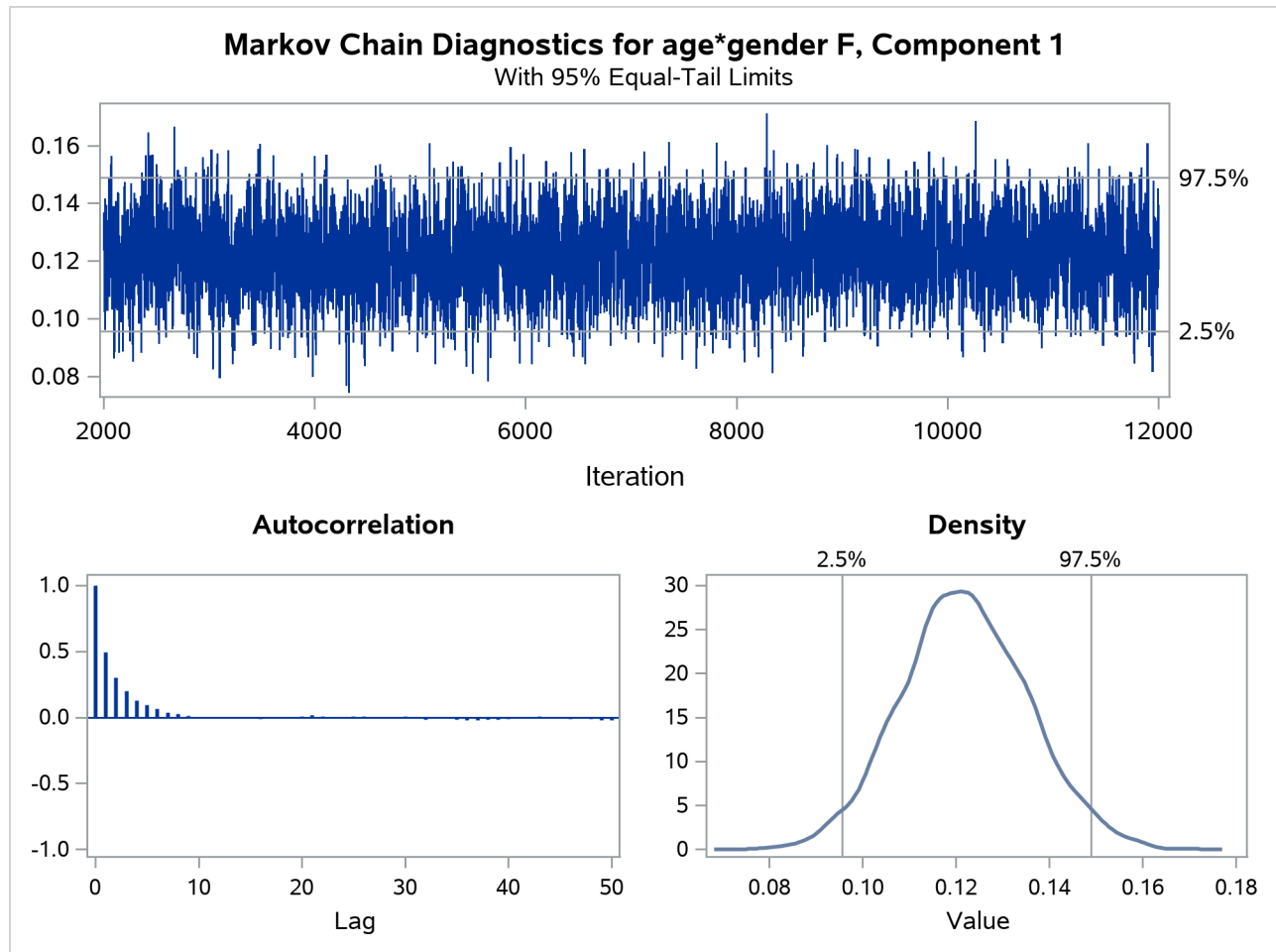
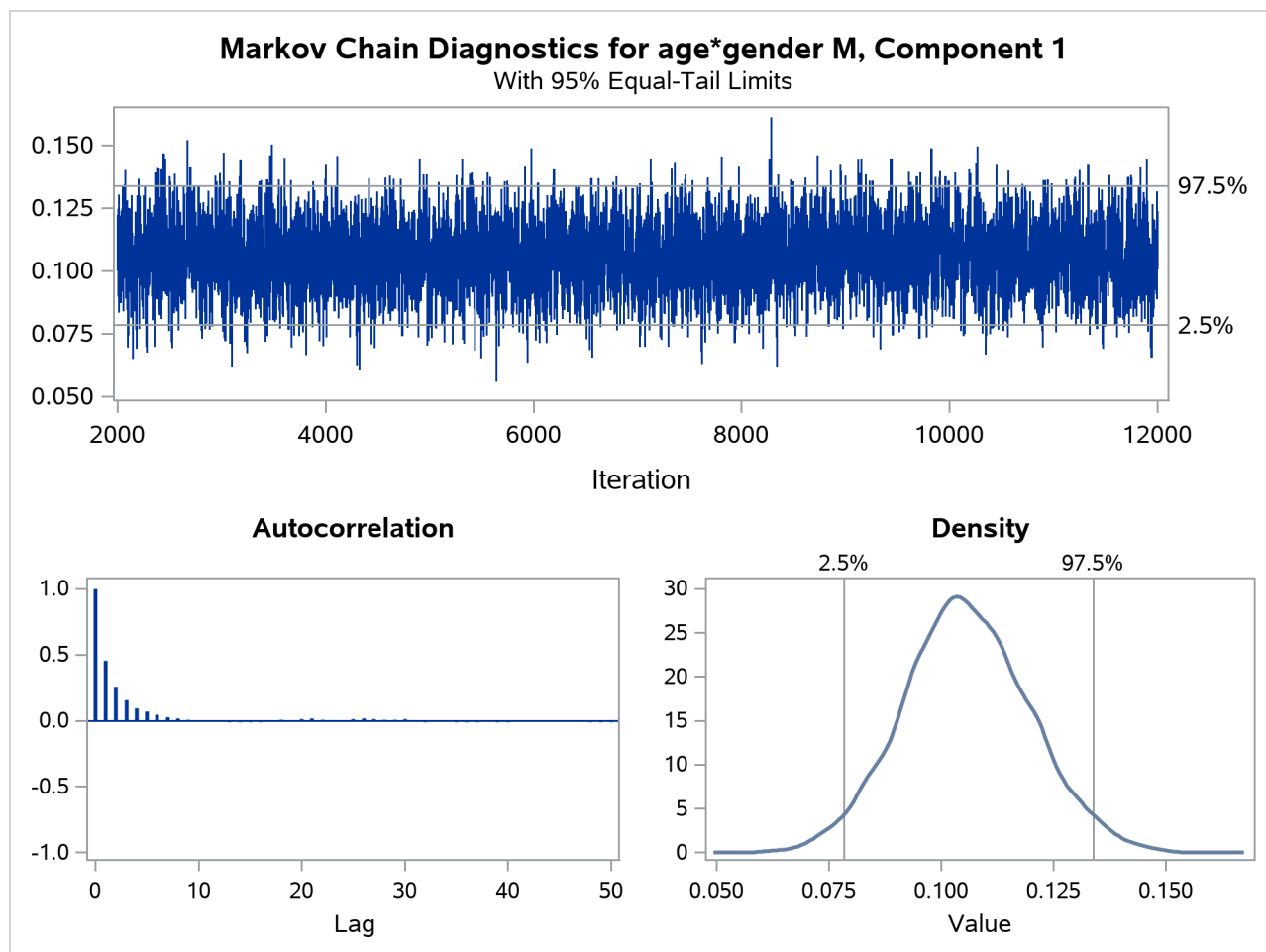


Figure 55.13 continued



## Looking for Multiple Modes: Are Galaxies Clustered?

Mixture modeling is essentially a generalized form of one-dimensional cluster analysis. The following example shows how you can use PROC HPFMM to explore the number and nature of Gaussian clusters in univariate data.

Roeder (1990) presents data from the Corona Borealis sky survey with the velocities of 82 galaxies in a narrow slice of the sky. Cosmological theory suggests that the observed velocity of each galaxy is proportional to its distance from the observer. Thus, the presence of multiple modes in the density of these velocities could indicate a clustering of the galaxies at different distances.

The following DATA step recreates the data set in Roeder (1990). The computed variable *v* represents the measured velocity in thousands of kilometers per second.

```

title "HPFMM Analysis of Galaxies Data";
data galaxies;
  input velocity @@;
  v = velocity / 1000;
  datalines;
9172  9350  9483  9558  9775  10227  10406  16084  16170  18419
18552 18600 18927 19052 19070 19330 19343 19349 19440 19473
19529 19541 19547 19663 19846 19856 19863 19914 19918 19973
19989 20166 20175 20179 20196 20215 20221 20415 20629 20795
20821 20846 20875 20986 21137 21492 21701 21814 21921 21960
22185 22209 22242 22249 22314 22374 22495 22746 22747 22888
22914 23206 23241 23263 23484 23538 23542 23666 23706 23711
24129 24285 24289 24366 24717 24990 25633 26960 26995 32065
32789 34279
;

```

Analysis of potentially multimodal data is a natural application of finite mixture models. In this case, the modeling is complicated by the question of the variance for each of the components. Using identical variances for each component could obscure underlying structure, but the additional flexibility granted by component-specific variances might introduce spurious features.

You can use PROC HPFMM to prepare analyses for equal and unequal variances and use one of the available fit statistics to compare the resulting models. You can use the model selection facility to explore models with varying numbers of mixture components—say, from three to seven as investigated in Roeder (1990). The following statements select the best unequal-variance model using Akaike’s information criterion (AIC), which has a built-in penalty for model complexity:

```

title2 "Three to Seven Components, Unequal Variances";
ods graphics on;
proc hpfmm data=galaxies criterion=AIC;
  model v = / kmin=3 kmax=7;
  ods exclude IterHistory OptInfo ComponentInfo;
run;

```

The **KMIN=** and **KMAX=** options indicate the smallest and largest number of components to consider. The ODS GRAPHICS and ODS SELECT statements request a density plot. The output for unequal variances is shown in [Figure 55.14](#) and [Figure 55.15](#).

**Figure 55.14** Model Selection for Galaxy Data Assuming Unequal Variances

### HPFMM Analysis of Galaxies Data Three to Seven Components, Unequal Variances

#### The HPFMM Procedure

Model Information	
Data Set	WORK.GALAXIES
Response Variable	v
Type of Model	Homogeneous Mixture
Distribution	Normal
Min Components	3
Max Components	7
Link Function	Identity
Estimation Method	Maximum Likelihood

#### Component Evaluation for Mixture Models

Model ID	Number of Components		Parameters		Eff. -2 Log L	AIC	AICC	BIC	Pearson	Max Gradient
	Total	Eff.	Total	Eff.						
1	3	3	8	8	406.96	422.96	424.94	442.22	82.00	0.000025
2	4	4	11	11	406.96	428.96	432.74	455.44	82.00	0.00011
3	5	5	14	14	406.96	434.96	441.23	468.66	82.00	0.000054
4	6	6	17	17	406.96	440.96	450.53	481.88	82.00	0.000020
5	7	7	20	20	406.96	446.96	460.73	495.10	82.00	0.000039

The model with 3 components (ID=1) was selected as 'best' based on the AIC statistic.

#### Fit Statistics

-2 Log Likelihood	407.0
AIC (Smaller is Better)	423.0
AICC (Smaller is Better)	424.9
BIC (Smaller is Better)	442.2
Pearson Statistic	82.0001
Effective Parameters	8
Effective Components	3

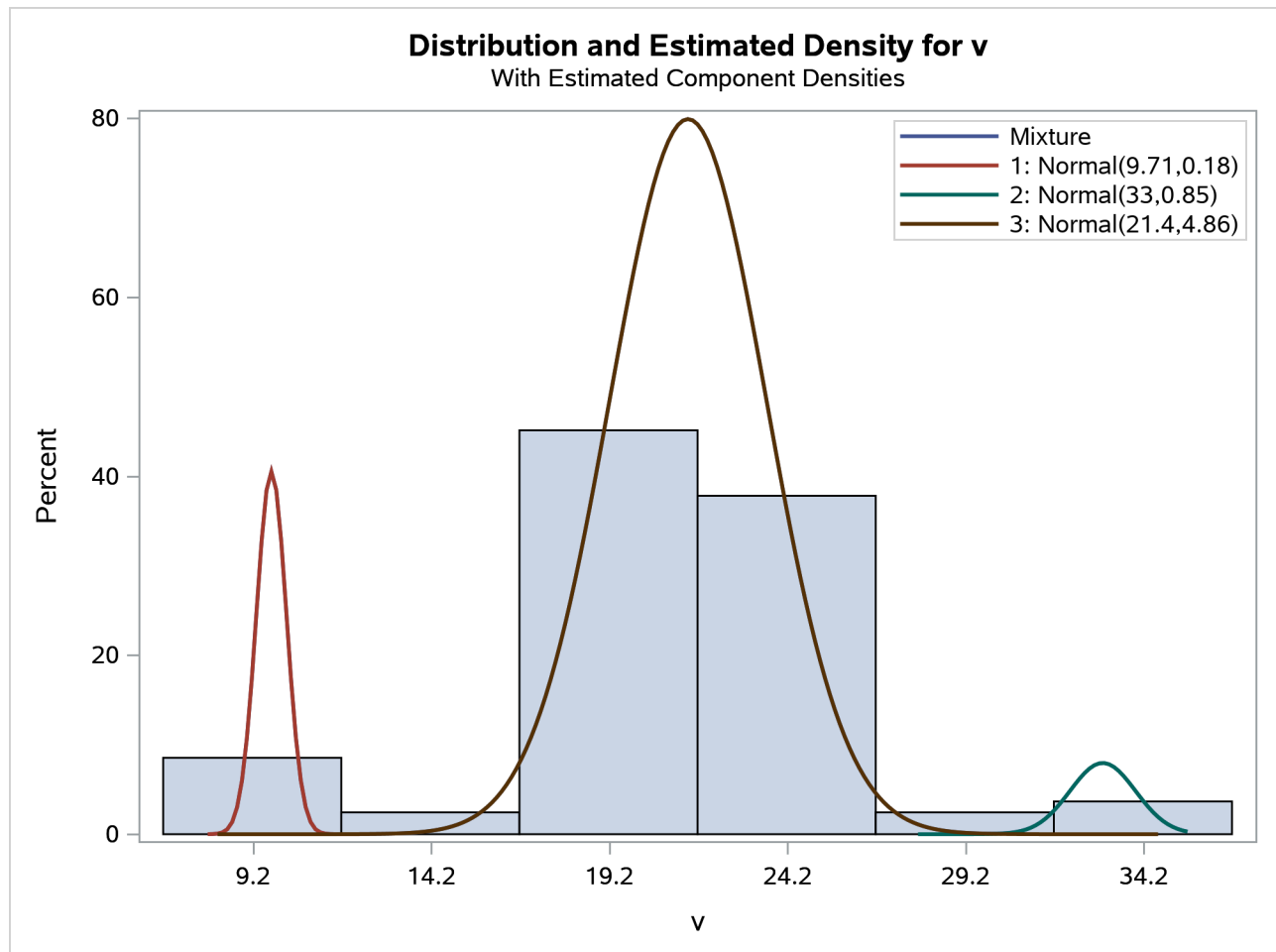
#### Parameter Estimates for Normal Model

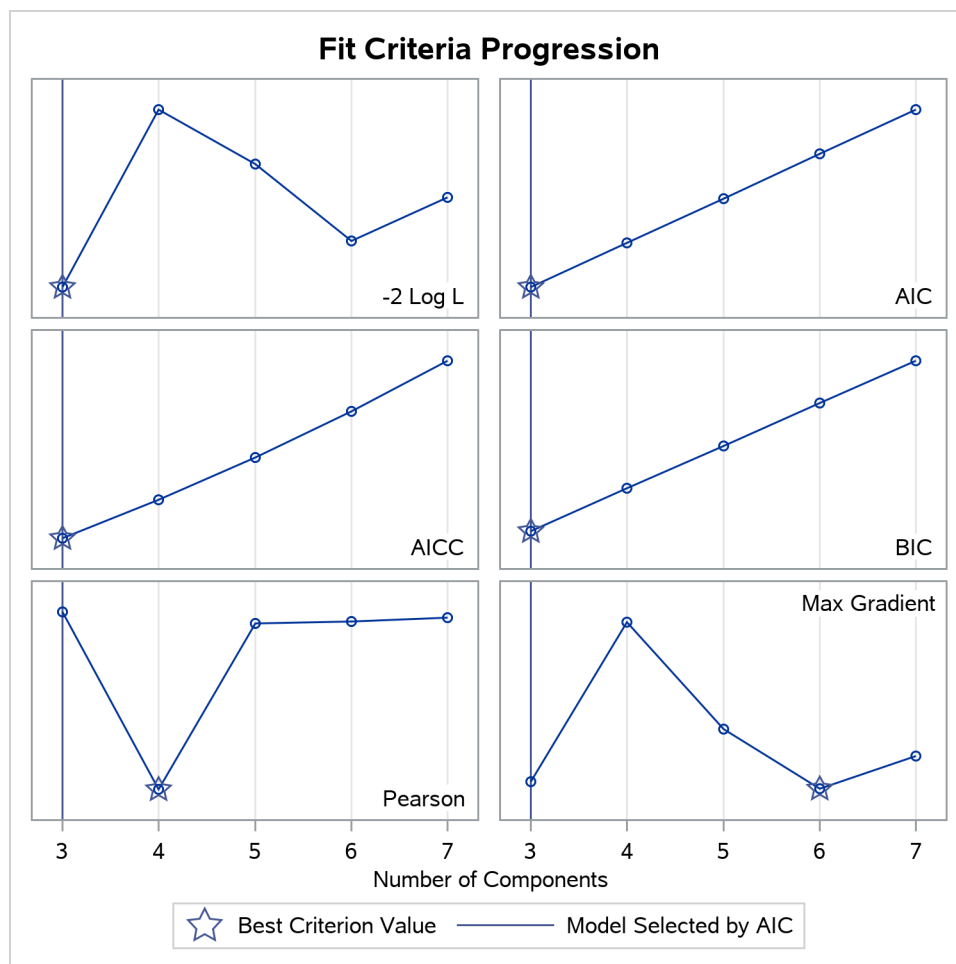
Component	Parameter	Standard			
		Estimate	Error	z Value	Pr >  z
1	Intercept	9.7101	0.1597	60.80	<.0001
2	Intercept	33.0444	0.5322	62.09	<.0001
3	Intercept	21.4039	0.2597	82.41	<.0001
1	Variance	0.1785	0.09542		
2	Variance	0.8496	0.6937		
3	Variance	4.8567	0.8098		



**Figure 55.14** *continued*

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	GLogit(Prob)	Linked Scale		
			Standard Error	z Value	Pr >  z
1	0.0854	-2.3308	0.3959	-5.89	<.0001
2	0.0366	-3.1781	0.5893	-5.39	<.0001
3	0.8781	0			

**Figure 55.15** Density Plot for Best (Three-Component) Model Assuming Unequal Variances

**Figure 55.16** Criterion Panel Plot for Model Selection Assuming Unequal Variances

This example uses the AIC for model selection. Figure 55.16 shows the AIC and other model fit criteria for each of the fitted models.

To require that the separate components have identical variances, add the **EQUATE=SCALE** option in the **MODEL** statement:

```
title2 "Three to Seven Components, Equal Variances";
proc hpfmm data=galaxies criterion=AIC gconv=0;
  model v = / kmin=3 kmax=7 equate=scale;
run;
```

The **GCONV=** convergence criterion is turned off in this PROC HPFMM run to avoid the early stoppage of the iterations when the relative gradient changes little between iterations. Turning the criterion off usually ensures that convergence is achieved with a small absolute gradient of the objective function.

The output for equal variances is shown in Figure 55.17 and Figure 55.18.

**Figure 55.17** Model Selection for Galaxy Data Assuming Equal Variances

### HPFMM Analysis of Galaxies Data Three to Seven Components, Equal Variances

#### The HPFMM Procedure

Model Information	
Data Set	WORK.GALAXIES
Response Variable	v
Type of Model	Homogeneous Mixture
Distribution	Normal
Min Components	3
Max Components	7
Link Function	Identity
Estimation Method	Maximum Likelihood

#### Component Evaluation for Mixture Models

Model ID	Number of Components		Parameters		Eff. -2 Log L	AIC	AICC	BIC	Pearson	Max Gradient
	Total	Eff.	Total	Eff.						
1	3	3	6	6	478.74	490.74	491.86	505.18	82.00	1.197E-6
2	4	4	8	8	416.49	432.49	434.47	451.75	82.00	4.052E-6
3	5	5	10	10	416.49	436.49	439.59	460.56	82.00	4.316E-6
4	6	6	12	12	416.49	440.49	445.02	469.37	82.00	6.294E-6
5	7	7	14	14	416.49	444.49	450.76	478.19	82.00	4.885E-6

The model with 4 components (ID=2) was selected as 'best' based on the AIC statistic.

#### Fit Statistics

-2 Log Likelihood	416.5
AIC (Smaller is Better)	432.5
AICC (Smaller is Better)	434.5
BIC (Smaller is Better)	451.7
Pearson Statistic	82.0000
Effective Parameters	8
Effective Components	4

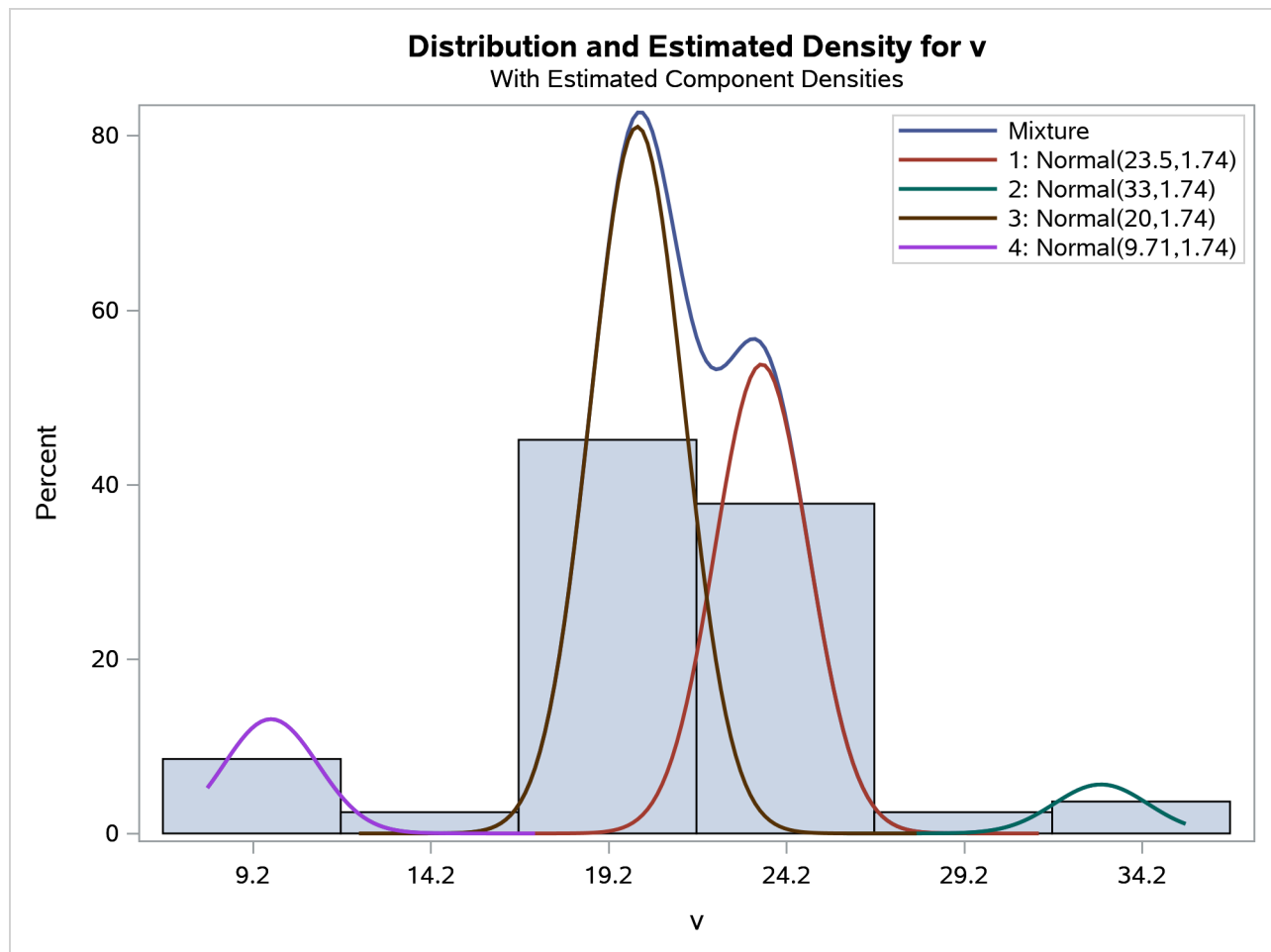
#### Parameter Estimates for Normal Model

Component	Parameter	Estimate	Standard		
			Error	z Value	Pr >  z
1	Intercept	23.5058	0.3460	67.93	<.0001
2	Intercept	33.0440	0.7610	43.42	<.0001
3	Intercept	20.0086	0.3029	66.06	<.0001
4	Intercept	9.7103	0.4981	19.50	<.0001
1	Variance	1.7354	0.3905		
2	Variance	1.7354	0.3905		
3	Variance	1.7354	0.3905		
4	Variance	1.7354	0.3905		

Figure 55.17 continued

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	GLogit(Prob)	Linked Scale		
			Standard Error	z Value	Pr >  z
1	0.3503	1.4118	0.4497	3.14	0.0017
2	0.0366	-0.8473	0.6901	-1.23	0.2195
3	0.5277	1.8216	0.4205	4.33	<.0001
4	0.0854	0			

Figure 55.18 Density Plot for Best (Six-Component) Model Assuming Equal Variances



Not surprisingly, the two variance specifications produce different optimal models. The unequal variance specification favors a three-component model while the equal variance specification favors a four-component model. Comparison of the AIC fit statistics, 423.0 and 432.5, indicates that the three-component, unequal variance model provides the best overall fit.

## Comparison with Roeder's Method

It is important to note that Roeder's original analysis proceeds in a different manner than the finite mixture modeling presented here. The technique presented by Roeder first develops a "best" range of scale parameters based on a specific criterion. Roeder then uses fixed scale parameters taken from this range to develop optimal equal-scale Gaussian mixture models.

You can reproduce Roeder's point estimate for the density by specifying a five-component Gaussian mixture. In addition, use the **EQUATE=SCALE** option in the **MODEL** statement and a **RESTRICT** statement fixing the first component's scale parameter at 0.9025 (Roeder's  $h = 0.95$ ,  $\text{scale} = h^2$ ). The combination of these options produces a mixture of five Gaussian components, each with variance 0.9025. The following statements conduct this analysis:

```
title2 "Five Components, Equal Variances = 0.9025";
proc hpfmm data=galaxies;
  model v = / K=5 equate=scale;
  restrict int 0 (scale 1) = 0.9025;
run;
ods graphics off;
```

The output is shown in [Figure 55.19](#) and [Figure 55.20](#).

**Figure 55.19** Reproduction of Roeder's Five-Component Analysis of Galaxy Data

### HPFMM Analysis of Galaxies Data Five Components, Equal Variances = 0.9025

#### The HPFMM Procedure

Model Information	
Data Set	WORK.GALAXIES
Response Variable	v
Type of Model	Homogeneous Mixture
Distribution	Normal
Components	5
Link Function	Identity
Estimation Method	Maximum Likelihood

Fit Statistics	
-2 Log Likelihood	412.2
AIC (Smaller is Better)	430.2
AICC (Smaller is Better)	432.7
BIC (Smaller is Better)	451.9
Pearson Statistic	82.5549
Effective Parameters	9
Effective Components	5

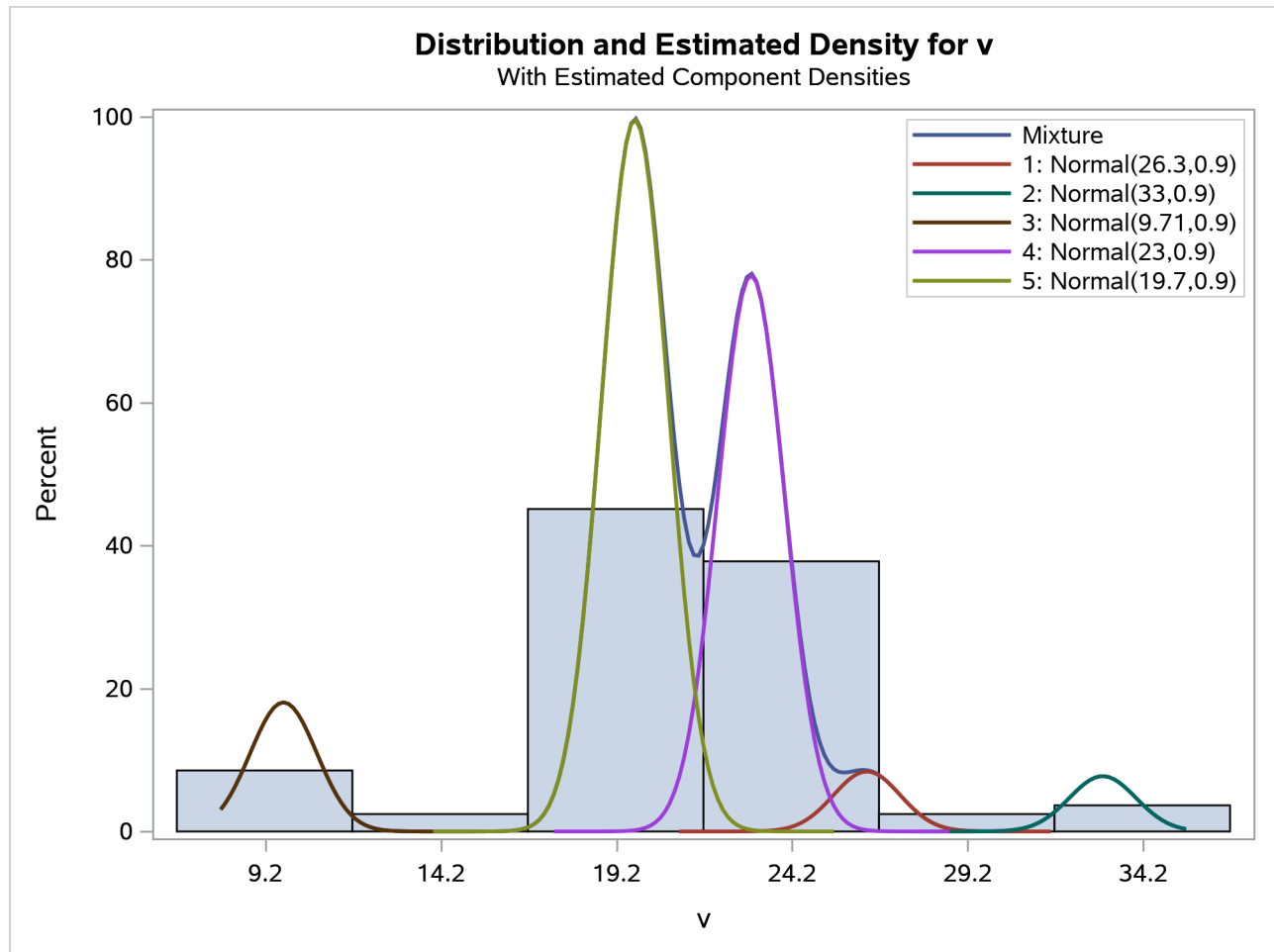
Linear Constraints at Solution	
k = 1	Constraint Active
Variance = 0.90	Yes

**Figure 55.19** *continued*

Parameter Estimates for Normal Model						
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z	
1	Intercept	26.3266	0.7778	33.85	<.0001	
2	Intercept	33.0443	0.5485	60.25	<.0001	
3	Intercept	9.7101	0.3591	27.04	<.0001	
4	Intercept	23.0295	0.2294	100.38	<.0001	
5	Intercept	19.7187	0.1784	110.55	<.0001	
1	Variance	0.9025	0			
2	Variance	0.9025	0			
3	Variance	0.9025	0			
4	Variance	0.9025	0			
5	Variance	0.9025	0			

Parameter Estimates for Mixing Probabilities						
Component	Mixing Probability	GLogit(Prob)	Standard Error	z Value	Pr >  z	
1	0.0397	-2.4739	0.7084	-3.49	0.0005	
2	0.0366	-2.5544	0.6016	-4.25	<.0001	
3	0.0854	-1.7071	0.4141	-4.12	<.0001	
4	0.3678	-0.2466	0.2699	-0.91	0.3609	
5	0.4706	0				

**Figure 55.20** Density Plot for Roeder's Analysis

## Syntax: HPFMM Procedure

The following statements are available in the HPFMM procedure:

```

PROC HPFMM < options > ;
  BAYES bayes-options ;
  BY variables ;
  CLASS variables ;
  FREQ variable ;
  ID variables ;
  MODEL response< (response-options) > = < effects > < / model-options > ;
  MODEL events/trials = < effects > < / model-options > ;
  MODEL                + < effects > < / model-options > ;
  OUTPUT < OUT=SAS-data-set >
    < keyword< (keyword-options) > < =name > > ...
    < keyword< (keyword-options) > < =name > > < / options > ;
  PERFORMANCE performance-options ;
  PROBMODEL < effects > < / probmodel-options > ;
  RESTRICT < 'label' > constraint-specification < , ... , constraint-specification >
    < operator < value > > < / option > ;
  WEIGHT variable ;

```

The **PROC HPFMM** statement and at least one **MODEL** statement is required. The **CLASS**, **RESTRICT** and **MODEL** statements can appear multiple times. If a **CLASS** statement is specified, it must precede the **MODEL** statements. The **RESTRICT** statements must appear after the **MODEL** statements.

## PROC HPFMM Statement

```
PROC HPFMM < options > ;
```

The **PROC HPFMM** statement invokes the HPFMM procedure. Table 55.2 summarizes the *options* available in the **PROC HPFMM** statement. These and other *options* in the **PROC HPFMM** statement are then described fully in alphabetical order.

**Table 55.2** PROC HPFMM Statement Options

Option	Description
<b>Basic Options</b>	
<b>DATA=</b>	Specifies the input data set
<b>EXCLUSION=</b>	Specifies how the procedure responds to support violations in the data
<b>NAMELEN=</b>	Specifies the length of effect names
<b>SEED=</b>	Specifies the random number seed for analyses that require random number draws



**Table 55.2** *continued*

Option	Description
<b>Displayed Output</b>	
COMPONENTINFO	Displays information about the mixture components
CORR	Displays the asymptotic correlation matrix of the maximum likelihood parameter estimates or the empirical correlation matrix of the Bayesian posterior estimates
COV	Displays the asymptotic covariance matrix of the maximum likelihood parameter estimates or the empirical covariance matrix of the Bayesian posterior estimates
COVI	Displays the inverse of the covariance matrix of the parameter estimates
FITDETAILS	Displays fit information for all examined models
ITDETAILS	Adds estimates and gradients to the “Iteration History” table
NOCLPRINT	Suppresses the “Class Level Information” table completely or partially
NOITPRINT	Suppresses the “Iteration History Information” table
NOPRINT	Suppresses tabular and graphical output
PARMSTYLE=	Specifies how parameters are displayed in ODS tables
PLOTS	Produces ODS statistical graphics
<b>Computational Options</b>	
CRITERION=	Specifies the criterion used in model selection
NOCENTER	Prevents centering and scaling of the regressor variables
PARTIAL=	Specifies a variable that defines a partial classification
<b>Options Related to Optimization</b>	
ABSCONV=	Tunes an absolute function convergence criterion
ABSFCNV=	Tunes an absolute function difference convergence criterion
ABSGCONV=	Tunes the absolute gradient convergence criterion
FCONV=	Specifies a relative function convergence criterion that is based on a relative change of the function value
FCONV2=	Specifies a relative function convergence criterion that is based on a predicted reduction of the objective function
GCONV=	Tunes the relative gradient convergence criterion
MAXITER=	Specifies the maximum number of iterations in any optimization
MAXFUNC=	Specifies the maximum number of function evaluations in any optimization
MAXTIME=	Specifies the upper limit of CPU time in seconds for any optimization
MINITER=	Specifies the minimum number of iterations in any optimization
TECHNIQUE=	Selects the optimization technique

**Table 55.2** *continued*

Option	Description
<b>Singularity Tolerances</b>	
<b>INVALIDLOGL=</b>	Tunes the value assigned to an invalid component log likelihood
<b>SINGCHOL=</b>	Tunes singularity for Cholesky decompositions
<b>SINGRES=</b>	Tunes singularity for the residual variance
<b>SINGULAR=</b>	Tunes general singularity criterion

You can specify the following *options* in the PROC HPFMM statement.

**ABSCONV=*r*****ABSTOL=*r***

specifies an absolute function convergence criterion. For minimization, the termination criterion is  $f(\boldsymbol{\psi}^{(k)}) \leq r$ , where  $\boldsymbol{\psi}$  is the vector of parameters in the optimization and  $f(\cdot)$  is the objective function. The default value of  $r$  is the negative square root of the largest double-precision value, which serves only as a protection against overflows.

**ABSFCONV=*r*****ABSFTOL=*r***

specifies an absolute function difference convergence criterion. For all techniques except NMSIMP, the termination criterion is a small change of the function value in successive iterations:

$$|f(\boldsymbol{\psi}^{(k-1)}) - f(\boldsymbol{\psi}^{(k)})| \leq r$$

Here,  $\boldsymbol{\psi}$  denotes the vector of parameters that participate in the optimization, and  $f(\cdot)$  is the objective function. The same formula is used for the NMSIMP technique, but  $\boldsymbol{\psi}^{(k)}$  is defined as the vertex with the lowest function value, and  $\boldsymbol{\psi}^{(k-1)}$  is defined as the vertex with the highest function value in the simplex. The default value is  $r=0$ .

**ABSGCONV=*r*****ABSGTOL=*r***

specifies an absolute gradient convergence criterion. The termination criterion is a small maximum absolute gradient element:

$$\max_j |g_j(\boldsymbol{\psi}^{(k)})| \leq r$$

Here,  $\boldsymbol{\psi}$  denotes the vector of parameters that participate in the optimization, and  $g_j(\cdot)$  is the gradient of the objective function with respect to the  $j$ th parameter. This criterion is not used by the NMSIMP technique. The default value is  $r=1\text{E}-5$ .

**COMPONENTINFO****COMPINFO****CINFO**

produces a table with additional details about the fitted model components.

**COV**

produces the covariance matrix of the parameter estimates. For maximum likelihood estimation, this matrix is based on the inverse (projected) Hessian matrix. For Bayesian estimation, it is the empirical covariance matrix of the posterior estimates. The covariance matrix is shown for all parameters, even if they did not participate in the optimization or sampling.

**COVI**

produces the inverse of the covariance matrix of the parameter estimates. For maximum likelihood estimation, the covariance matrix is based on the inverse (projected) Hessian matrix. For Bayesian estimation, it is the empirical covariance matrix of the posterior estimates. This matrix is then inverted by sweeping, and rows and columns that correspond to linear dependencies or singularities are zeroed.

**CORR**

produces the correlation matrix of the parameter estimates. For maximum likelihood estimation this matrix is based on the inverse (projected) Hessian matrix. For Bayesian estimation, it is based on the empirical covariance matrix of the posterior estimates.

**CRITERION=***keyword*

**CRIT=***keyword*

specifies the criterion by which the HPFMM procedure ranks models when multiple models are evaluated during maximum likelihood estimation. You can choose from the following *keywords* to rank models:

<b>AIC</b>	based on Akaike's information criterion
<b>AICC</b>	based on the bias-corrected AIC criterion
<b>BIC</b>	based on the Bayesian information criterion
<b>GRADIENT</b>	based on the largest element of the gradient (in absolute value)
<b>LOGL   LL</b>	based on the mixture log likelihood
<b>PEARSON</b>	based on the Pearson statistic

The default is CRITERION=BIC.

**DATA=***SAS-data-set*

names the SAS data set to be used by PROC HPFMM. The default is the most recently created data set.

**EXCLUSION=**NONE | ANY | ALL

**EXCLUDE=**NONE | ANY | ALL

specifies how the HPFMM procedure handles support violations of observations. For example, in a mixture of two Poisson variables, negative response values are not possible. However, in a mixture of a Poisson and a normal variable, negative values are possible, and their likelihood contribution to the Poisson component is zero. An observation that violates the support of one component distribution of the model might be a valid response with respect to one or more other component distributions. This requires some nuanced handling of support violations in mixture models.

The default exclusion technique, EXCLUSION=ALL, removes an observation from the analysis only if it violates the support of all component distributions. The other extreme, EXCLUSION=NONE, permits an observation into the analysis regardless of support violations. EXCLUSION=ANY removes observations from the analysis if the response violates the support of any component distributions. In the single-component case, EXCLUSION=ALL and EXCLUSION=ANY are identical.

**FCONV=*r*****FTOL=*r***

specifies a relative function convergence criterion that is based on the relative change of the function value. For all techniques except NMSIMP, PROC HPFMM terminates when there is a small relative change of the function value in successive iterations:

$$\frac{|f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k-1)})|}{|f(\boldsymbol{\psi}^{(k-1)})|} \leq r$$

Here,  $\boldsymbol{\psi}$  denotes the vector of parameters that participate in the optimization, and  $f(\cdot)$  is the objective function. The same formula is used for the NMSIMP technique, but  $\boldsymbol{\psi}^{(k)}$  is defined as the vertex with the lowest function value, and  $\boldsymbol{\psi}^{(k-1)}$  is defined as the vertex with the highest function value in the simplex.

The default is  $r = 10^{-\text{FDIGITS}}$ , where FDIGITS is by default  $-\log_{10}\{\epsilon\}$ , and  $\epsilon$  is the machine precision.

**FCONV2=*r*****FTOL2=*r***

specifies a relative function convergence criterion that is based on the predicted reduction of the objective function. For all techniques except NMSIMP, the termination criterion is a small predicted reduction

$$df^{(k)} \approx f(\boldsymbol{\theta}^{(k)}) - f(\boldsymbol{\theta}^{(k)} + \mathbf{s}^{(k)})$$

of the objective function. The predicted reduction

$$\begin{aligned} df^{(k)} &= -\mathbf{g}^{(k)'} \mathbf{s}^{(k)} - \frac{1}{2} \mathbf{s}^{(k)'} \mathbf{H}^{(k)} \mathbf{s}^{(k)} \\ &= -\frac{1}{2} \mathbf{s}^{(k)'} \mathbf{g}^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function  $f$  by the first two terms of the Taylor series and substituting the Newton step:

$$\mathbf{s}^{(k)} = -[\mathbf{H}^{(k)}]^{-1} \mathbf{g}^{(k)}$$

For the NMSIMP technique, the termination criterion is a small standard deviation of the function values of the  $n + 1$  simplex vertices  $\boldsymbol{\theta}_l^{(k)}$ ,  $l = 0, \dots, n$ ,

$$\sqrt{\frac{1}{n+1} \sum_l \left[ f(\boldsymbol{\theta}_l^{(k)}) - \bar{f}(\boldsymbol{\theta}^{(k)}) \right]^2} \leq r$$

where  $\bar{f}(\boldsymbol{\theta}^{(k)}) = \frac{1}{n+1} \sum_l f(\boldsymbol{\theta}_l^{(k)})$ . If there are  $n_{act}$  boundary constraints active at  $\boldsymbol{\theta}^{(k)}$ , the mean and standard deviation are computed only for the  $n + 1 - n_{act}$  unconstrained vertices.

The default value is  $r = 1\text{E-}6$  for the NMSIMP technique and  $r = 0$  otherwise.

**FITDETAILS**

requests that the “Optimization Information,” “Iteration History,” and “Fit Statistics” tables be produced for all optimizations when models with different number of components are evaluated. For example, the following statements fit a binomial regression model with up to three components and produces fit and optimization information for all three:

```
proc hpfmm fitdetails;
  model y/n = x / kmax=3;
run;
```

Without the FITDETAILS option, only the “Fit Statistics” table for the selected model is displayed.

In Bayesian estimation, the FITDETAILS option displays the following tables for each model that the procedure fits: “Bayes Information,” “Iteration History,” “Prior Information,” “Fit Statistics,” “Posterior Summaries,” “Posterior Intervals,” and any requested diagnostics tables. The “Iteration History” table appears only if the **BAYES** statement includes the **INITIAL=MLE** option.

Without the FITDETAILS option, these tables are listed only for the selected model.

**GCONV=*r*****GTOL=*r***

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, the termination criterion is a small normalized predicted function reduction:

$$\frac{\mathbf{g}(\boldsymbol{\psi}^{(k)})' [\mathbf{H}^{(k)}]^{-1} \mathbf{g}(\boldsymbol{\psi}^{(k)})}{|f(\boldsymbol{\psi}^{(k)})|} \leq r$$

Here,  $\boldsymbol{\psi}$  denotes the vector of parameters that participate in the optimization,  $f(\cdot)$  is the objective function, and  $\mathbf{g}(\cdot)$  is the gradient. For the CONGRA technique (where a reliable Hessian estimate  $\mathbf{H}$  is not available), the following criterion is used:

$$\frac{\|\mathbf{g}(\boldsymbol{\psi}^{(k)})\|_2^2 \|\mathbf{s}(\boldsymbol{\psi}^{(k)})\|_2}{\|\mathbf{g}(\boldsymbol{\psi}^{(k)}) - \mathbf{g}(\boldsymbol{\psi}^{(k-1)})\|_2 |f(\boldsymbol{\psi}^{(k)})|} \leq r$$

This criterion is not used by the NMSIMP technique. The default value is  $r=1\text{E}-8$ .

**HESSIAN**

displays the Hessian matrix of the model. This option is not available for Bayesian estimation.

**INVALIDLOGL=*r***

specifies the value assumed by the HPFMM procedure if a log likelihood cannot be computed (for example, because the value of the response variable falls outside of the response distribution’s support). The default value is  $-1\text{E}20$ .

**ITDETAILS**

adds parameter estimates and gradients to the “Iteration History” table. If the HPFMM procedure centers or scales the model variables (or both), the parameter estimates and gradients reported during the iteration refer to that scale. You can suppress centering and scaling with the **NOCENTER** option.

**MAXFUNC=*n*****MAXFU=*n***

specifies the maximum number of function calls in the optimization process. The default values are as follows, depending on the optimization technique:

- TRUREG, NRRIDG, and NEWRAP: 125
- QUANEW and DBLDOG: 500
- CONGRA: 1000
- NMSIMP: 3000

The optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed the number that is specified by the MAXFUNC= option. You can choose the optimization technique with the [TECHNIQUE=](#) option.

**MAXITER=*n*****MAXIT=*n***

specifies the maximum number of iterations in the optimization process. The default values are as follows, depending on the optimization technique:

- TRUREG, NRRIDG, and NEWRAP: 50
- QUANEW and DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1000

These default values also apply when *n* is specified as a missing value. You can choose the optimization technique with the [TECHNIQUE=](#) option.

**MAXTIME=*r***

specifies an upper limit of *r* seconds of CPU time for the optimization process. The time is checked only at the end of each iteration. Therefore, the actual run time might be longer than the specified time. By default, CPU time is not limited.

**MINITER=*n*****MINIT=*n***

specifies the minimum number of iterations. The default value is 0. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms can behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations.

**NAMELEN=*number***

specifies the length to which long effect names are shortened. The default and minimum value is 20.

**NOCENTER**

requests that regressor variables not be centered or scaled. By default the HPFMM procedure centers and scales columns of the **X** matrix if the models contain intercepts. If [NOINT](#) options in [MODEL](#) statements are in effect, the columns of **X** are scaled but not centered. Centering and scaling can help with the stability of estimation and sampling algorithms. The HPFMM procedure does not produce a table of the centered and scaled coefficients and provides no user control over the type of centering and scaling that is applied. The NOCENTER option turns any centering and scaling off and processes the raw values of the continuous variables.

**NOCLPRINT=<number>**

suppresses the display of the “Class Level Information” table if you do not specify *number*. If you specify *number*, the values of the classification variables are displayed for only those variables whose number of levels is less than *number*. Specifying a *number* helps to reduce the size of the “Class Level Information” table if some classification variables have a large number of levels.

**NOITPRINT**

suppresses the display of the “Iteration History Information” table.

**NOPRINT**

suppresses the normal display of tabular and graphical results. The NOPRINT option is useful when you want to create only one or more output data sets with the procedure. This option temporarily disables the Output Delivery System (ODS); see Chapter 20, “Using the Output Delivery System,” for more information.

**PARMSTYLE=EFFECT | LABEL**

specifies the display style for parameters and effects. The HPFMM procedure can display parameters in two styles:

- The EFFECT style (which is used by the MIXED and GLIMMIX procedure, for example) identifies a parameter with an “Effect” column and adds separate columns for the **CLASS** variables in the model.
- The LABEL style creates one column, named Parameter, that combines the relevant information about a parameter into a single column. If your model contains multiple **CLASS** variables, the LABEL style might use space more economically.

The EFFECT style is the default for models that contain effects; otherwise the LABEL style is used (for example, in homogeneous mixtures). You can change the display style with the PARMSTYLE= option. Regardless of the display style, ODS output data sets that contain information about parameter estimates contain columns for both styles.

**PARTIAL=variable****MEMBERSHIP=variable**

specifies a variable in the input data set that identifies component membership. You can specify missing values for observations whose component membership is undetermined; this is known as a partial classification (McLachlan and Peel 2000, p. 75). For observations with known membership, the likelihood contribution is no longer a mixture. If observation  $i$  is known to be a member of component  $m$ , then its log likelihood contribution is

$$\log \{ \pi_m(\mathbf{z}, \boldsymbol{\alpha}_m) p_m(y; \mathbf{x}'_m \boldsymbol{\beta}_m, \phi_m) \}$$

Otherwise, if membership is undetermined, it is

$$\log \left\{ \sum_{j=1}^k \pi_j(\mathbf{z}, \boldsymbol{\alpha}_j) p_j(y; \mathbf{x}'_j \boldsymbol{\beta}_j, \phi_j) \right\}$$

The variable specified in the PARTIAL= option can be numeric or character. In case of a character variable, the variable must appear in the **CLASS** statement. If the PARTIAL= variable appears in the

**CLASS** statement, the membership assignment is made based on the levelized values of the variable, as shown in the “Class Level Information” table. Invalid values of the **PARTIAL=** variable are ignored.

In a model in which label switching is a problem, the switching can sometimes be avoided by assigning just a few observations to categories. For example, in a three-component model, switches might be prevented by assigning the observation with the smallest response value to the first component and the observation with the largest response value to the last component.

**PLOTS** *<(global-plot-options)> <=plot-request <(options)>>*

**PLOTS** *<(global-plot-options)> <=(plot-request <(options)> <... plot-request <(options)>>)>*

controls the plots produced through ODS Graphics.

ODS Graphics must be enabled before plots can be requested. For example:

```
ods graphics on;
proc hpfmm data=yeast seed=12345;
  model count/n = / k=2;
  freq f;
  performance nthreads=2;
  bayes;
run;
ods graphics off;
```

For more information about enabling and disabling ODS Graphics, see the section “[Enabling and Disabling ODS Graphics](#)” on page 623 in Chapter 21, “[Statistical Graphics Using ODS](#).”

## Global Plot Options

The *global-plot-options* apply to all relevant plots generated by the HPFMM procedure. The *global-plot-options* supported by the HPFMM procedure are as follows:

### UNPACKPANEL

#### UNPACK

displays each graph separately. (By default, some graphs can appear together in a single panel.)

#### ONLY

produces only the specified plots. This option is useful if you do not want the procedure to generate all default graphics, but only the ones specified.



## Specific Plot Options

The following listing describes the specific plots and their *options*.

### ALL

requests that all plots appropriate for the analysis be produced.

### NONE

requests that no ODS graphics be produced.

### DENSITY <(density-options)>

requests a plot of the data histogram and mixture density function. This graphic is a default graphic in models without effects in the **MODEL** statements and is available only in these models. Furthermore, all distributions involved in the mixture must be continuous. You can specify the following *density-options* to modify the plot:

#### CUMULATIVE

#### CDF

displays the histogram and densities in cumulative form.

#### NBINS=*n*

#### BINS=*n*

specifies the number of bins in the histogram; *n* is greater than or equal to 0. By default, the HPFMM procedure computes a suitable bin width and number of bins, based on the range of the response and the number of usable observations. The option has no effect for binary data.

### NOCOMPONENTS

#### NOCOMP

suppresses the component densities from the plot. If the component densities are displayed, they are scaled so that their sum equals the mixture density at any point on the graph. In single-component models, this option has no effect.

### NODENSITY

#### NODENS

suppresses the computation of the mixture density (and the component densities if the COMPONENTS suboption is specified). If you specify the NOHISTOGRAM and the NODENSITY option, no graphic is produced.

### NOLABEL

suppresses the component identification with labels. By default, the HPFMM procedure labels component densities in the legend of the plot. If you do not specify a model label with the **LABEL=** option in the **MODEL** statement, an identifying label is constructed from the parameter estimates that are associated with the component. In this case the parameter values are not necessarily the mean and variance of the distribution; the values used to identify the densities on the plot are chosen to simplify linking between graphical and tabular results.

**NOHISTOGRAM****NOHIST**

suppresses the computation of the histogram of the raw values. If you specify the NOHISTOGRAM and the NODENSITY option, no graphic is produced.

**NPOINTS=*n*****N=*n***

specifies the number of values used to compute the density functions; *n* is greater than or equal to 0. The default is N=200.

**WIDTH=*value*****BINWIDTH=*value***

specifies the bin width for the histogram. The *value* is specified in units of the response variable and must be positive. The option has no effect for binary data.

**TRACE <(tadpanel-options)>**

requests a trace panel with posterior diagnostics for a Bayesian analysis. If a BAYES statement is present, the trace panel plots are generated by default, one for each sampled parameter. You can specify the following *tadpanel-options* to modify the graphic:

**BOX****BOXPLOT**

replaces the autocorrelation plot with a box plot of the posterior sample.

**SMOOTH=NONE | MEAN | SPLINE**

adds a reference estimate to the trace plot. By default, SMOOTH=NONE. SMOOTH=MEAN uses the arithmetic mean of the trace as the reference. SMOOTH=SPLINE adds a penalized B-spline.

**REFERENCE= *reference-style***

adds vertical reference lines to the density plot, trace plot, and box plot. The available options for the *reference-style* are:

**NONE** suppresses the reference lines

**EQT** requests equal-tail intervals

**HPD** requests intervals of highest posterior density. The level for the credible or HPD intervals is chosen based on the “Posterior Interval Statistics” table.

**PERCENTILES** (or PERC) for percentiles. Up to three percentiles can be displayed, as based on the “Posterior Summary Statistics” table.

The default is REFERENCE=EQT.

**UNPACK**

unpacks the panel graphic and displays its elements as separate plots.

**CRITERIONPANEL** <(critpanel-options)>

requests a plot for comparing the model fit criteria for different numbers of components. This plot is available only if you also specify the **KMAX** option in at least one **MODEL** statement. The plot includes different criteria, depending on whether you are using maximum likelihood or Bayesian estimation. You can specify the following *critpanel-option* to modify the plot:

**UNPACK**

unpacks the panel plot and displays its elements as separate plots, one for each fit criterion.

**SEED**=*n*

determines the random number seed for analyses that depend on a random number stream. If you do not specify a seed or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. The largest possible value for the seed is  $2^{31} - 1$ . The seed value is reported in the “Model Information” table.

You can use the **SYSRANDOM** and **SYSRANEND** macro variables after a PROC HPFMM run to query the initial and final seed values. However, using the final seed value as the starting seed for a subsequent analysis does not continue the random number stream where the previous analysis left off. The **SYSRANEND** macro variable provides a mechanism to pass on seed values to ensure that the sequence of random numbers is the same every time you run an entire program.

Analyses that use the same (nonzero) seed are not completely reproducible if they are executed with a different number of threads since the random number streams in separate threads are independent. You can control the number of threads used by the HPFMM procedure with system options or through the **PERFORMANCE** statement in the HPFMM procedure.

**SINGCHOL**=*number*

tunes the singularity criterion in Cholesky decompositions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**SINGRES**=*number*

sets the tolerance for which the residual variance or scale parameter is considered to be zero. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**SINGULAR**=*number*

tunes the general singularity criterion applied by the HPFMM procedure in sweeps and inversions. The default is 1E4 times the machine epsilon; this product is approximately 1E–12 on most computers.

**TECHNIQUE**=*keyword***TECH**=*keyword*

specifies the optimization technique to obtain maximum likelihood estimates. You can choose from the following techniques by specifying the appropriate *keyword*:

<b>CONGRA</b>	performs a conjugate-gradient optimization.
<b>DBLDOG</b>	performs a version of double-dogleg optimization.
<b>NEWRAP</b>	performs a Newton-Raphson optimization combining a line-search algorithm with ridging.
<b>NMSIMP</b>	performs a Nelder-Mead simplex optimization.
<b>NONE</b>	performs no optimization.

<b>NRRIDG</b>	performs a Newton-Raphson optimization with ridging.
<b>QUANEW</b>	performs a dual quasi-Newton optimization.
<b>TRUREG</b>	performs a trust-region optimization.

The default is TECH=QUANEW.

For more details about these optimization methods, see the section “[Choosing an Optimization Algorithm](#)” on page 4507.

#### **ZEROPROB=***number*

tunes the threshold (a value between 0 and 1) below which the HPFMM procedure considers a component mixing probability to be zero. This affects the calculation of the number of effective components. The default is the square root of the machine epsilon; this is approximately 1E–8 on most computers.

---

## **BAYES Statement**

**BAYES** *bayes-options* ;

The BAYES statement requests that the parameters of the model be estimated by Markov chain Monte Carlo sampling techniques. The HPFMM procedure can estimate by maximum likelihood the parameters of all models supported by the procedure. Bayes estimation, on the other hand, is available for only a subset of these models.

In Bayesian analysis, it is essential to examine the convergence of the Markov chains before you proceed with posterior inference. With ODS Graphics turned on, the HPFMM procedure produces graphs at the end of the procedure output; these graphs enable you to visually examine the convergence of the chain. Inferences cannot be made if the Markov chain has not converged.

The output produced for a Bayesian analysis is markedly different from that for a frequentist (maximum likelihood) analysis for the following reasons:

- Parameter estimates do not have the same interpretation in the two analyses. Parameters are fixed unknown constants in the frequentist context and random variables in a Bayesian analysis.
- The results of a Bayesian analysis are summarized through chain diagnostics and posterior summary statistics and intervals.
- The HPFMM procedure samples the mixing probabilities in Bayesian models directly, rather than mapping them onto a logistic (or other) scale.

The HPFMM procedure applies highly specialized sampling algorithms in Bayesian models. For single-component models without effects, a conjugate sampling algorithm is used where possible. For models in the exponential family that contain effects, the sampling algorithm is based on Gamerman (1997). For the normal and *t* distributions, a conjugate sampler is the default sampling algorithm for models with and without effects. In multi-component models, the sampling algorithm is based on latent variable sampling through data augmentation (Frühwirth-Schnatter 2006) and the Gamerman or conjugate sampler. Because of this specialization, the options for controlling the prior distributions of the parameters are limited.

Table 55.3 summarizes the *bayes-options* available in the BAYES statement. The full assortment of options is then described in alphabetical order.

**Table 55.3** BAYES Statement Options

Option	Description
<b>Options Related to Sampling</b>	
INITIAL=	Specifies how to construct initial values
NBI=	Specifies the number of burn-in samples
NMC=	Specifies the number of samples after burn-in
METROPOLIS	Forces a Metropolis-Hastings sampling algorithm even if conjugate sampling is possible
OUTPOST=	Generates a data set that contains the posterior estimates
THIN=	Controls the thinning of the Markov chain
<b>Specification of Prior Information</b>	
MIXPRIORPARMS	Specifies the prior parameters for the Dirichlet distribution of the mixing probabilities
BETAPRIORPARMS=	Specifies the parameters of the normal prior distribution for individual parameters in the $\beta$ vector
MUPRIORPARMS=	Specifies the parameters of the prior distribution for the means in homogeneous mixtures without effects
PHIPRIORPARMS=	Specifies the parameters of the inverse gamma prior distribution for the scale parameters in homogeneous mixtures
PRIOROPTIONS	Specifies additional options used in the determination of the prior distribution
<b>Posterior Summary Statistics and Convergence Diagnostics</b>	
DIAGNOSTICS=	Displays convergence diagnostics for the Markov chain
STATISTICS	Displays posterior summary information for the Markov chain
<b>Other Options</b>	
ESTIMATE=	Specifies which estimate is used for the computation of OUTPUT statistics and graphics
TIMEINC=	Specifies the time interval to report on sampling progress (in seconds)

You can specify the following *bayes-options* in the BAYES statement.

**BETAPRIORPARMS=***pair-specification*

**BETAPRIORPARMS**(*pair-specification* ... *pair-specification*)

specifies the parameters for the normal prior distribution of the parameters that are associated with model effects ( $\beta$ s). The *pair-specification* is of the form ( $a, b$ ), and the values  $a$  and  $b$  are the mean and variance of the normal distribution, respectively. This option overrides the PRIOROPTIONS option.

The form of the BETAPRIORPARMS with an equal sign and a single pair is used to specify one pair of prior parameters that applies to all components in the mixture. In the following example, the two intercepts and the two regression coefficients all have a  $N(0, 100)$  prior distribution:

```
proc hpfmm;
  model y = x / k=2;
  bayes betapriorparms=(0,100);
run;
```

You can also provide a list of pairs to specify different sets of prior parameters for the various regression parameters and components. For example:

```
proc hpfmm;
  model y = x/ k=2;
  bayes betapriorparms( (0,10) (0,20) (.,.) (3,100) );
run;
```

The simple linear regression in the first component has a  $N(0, 10)$  prior for the intercept and a  $N(0, 20)$  prior for the slope. The prior for the intercept in the second component uses the HPFMM default, whereas the prior for the slope is  $N(3, 100)$ .

**DIAGNOSTICS=ALL | NONE | (keyword-list)**

**DIAG=ALL | NONE | (keyword-list)**

controls the computation of diagnostics for the posterior chain. You can request all posterior diagnostics by specifying **DIAGNOSTICS=ALL** or suppress the computation of posterior diagnostics by specifying **DIAGNOSTICS=NONE**. The following *keywords* enable you to select subsets of posterior diagnostics; the default is **DIAGNOSTICS=(AUTOCORR)**.

**AUTOCORR <(LAGS= numeric-list)>**

computes for each sampled parameter the autocorrelations of lags specified in the **LAGS=** list. Elements in the list are truncated to integers, and repeated values are removed. If the **LAGS=** option is not specified, autocorrelations are computed by default for lags 1, 5, 10, and 50. See the section “[Autocorrelations](#)” on page 157 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for details.

**ESS**

computes an estimate of the effective sample size (Kass et al. 1998), the correlation time, and the efficiency of the chain for each parameter. See the section “[Effective Sample Size](#)” on page 157 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for details.

**GEWEKE <(geweke-options)>**

computes the Geweke spectral density diagnostics (Geweke 1992), which are essentially a two-sample  $t$  test between the first  $f_1$  portion and the last  $f_2$  portion of the chain. The default is  $f_1 = 0.1$  and  $f_2 = 0.5$ , but you can choose other fractions by using the following *geweke-options*:

**FRAC1=value**

specifies the fraction  $f_1$  for the first window.

**FRAC2=value**

specifies the fraction  $f_2$  for the second window.

See the section “[Geweke Diagnostics](#)” on page 151 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for details.

**HEIDELBERGER** <(Heidel-options)>**HEIDEL** <(Heidel-options)>

computes the Heidelberg and Welch diagnostic (which consists of a stationarity test and a half-width test) for each variable. The stationary diagnostic test tests the null hypothesis that the posterior samples are generated from a stationary process. If the stationarity test is passed, a half-width test is then carried out. See the section “[Heidelberg and Welch Diagnostics](#)” on page 153 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for more details.

These diagnostics are not performed by default. You can specify the **DIAGNOSTICS=HEIDELBERGER** option to request these diagnostics, and you can also specify suboptions, such as **DIAGNOSTICS=HEIDELBERGER(EPS=0.05)**, as follows:

**SALPHA=***value*

specifies the  $\alpha$  level ( $0 < \alpha < 1$ ) for the stationarity test. By default, **SALPHA=0.05**.

**HALPHA=***value*

specifies the  $\alpha$  level ( $0 < \alpha < 1$ ) for the half-width test. By default, **HALPHA=0.05**.

**EPS=***value*

specifies a small positive number  $\epsilon$  such that if the half-width is less than  $\epsilon$  times the sample mean of the retaining iterates, the half-width test is passed. By default, **EPS=0.1**.

**MCERROR****MCSE**

computes an estimate of the Monte Carlo standard error for each sampled parameter. See the section “[Standard Error of the Mean Estimate](#)” on page 158 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for details.

**MAXLAG=***n*

specifies the largest lag used in computing the effective sample size and the Monte Carlo standard error. Specifying this option implies the **ESS** and **MCERROR** options. The default is **MAXLAG=250**.

**RAFTERY** <(Raftery-options)>**RL** <(Raftery-options)>

computes the Raftery and Lewis diagnostics, which evaluate the accuracy of the estimated quantile ( $\hat{\theta}_Q$  for a given  $Q \in (0, 1)$ ) of a chain.  $\hat{\theta}_Q$  can achieve any degree of accuracy when the chain is allowed to run for a long time. The algorithm stops when the estimated probability  $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$  reaches within  $\pm R$  of the value  $Q$  with probability  $S$ ; that is,  $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$ . See the section “[Raftery and Lewis Diagnostics](#)” on page 154 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for more details. The *Raftery-options* enable you to specify  $Q$ ,  $R$ ,  $S$ , and a precision level  $\epsilon$  for a stationary test.

These diagnostics are not performed by default. You can specify the **DIAGNOSTICS=RAFTERY** option to request these diagnostics, and you can also specify suboptions, such as **DIAGNOSTICS=RAFTERY(QUANTILE=0.05)**, as follows:

**QUANTILE=***value***Q=***value*

specifies the order (a value between 0 and 1) of the quantile of interest. By default, QUANTILE=0.025.

**ACCURACY=***value***R=***value*

specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. By default, ACCURACY=0.005.

**PROB=***value***S=***value*

specifies the probability of attaining the accuracy of the estimation of the quantile. By default, PROB=0.95.

**EPS=***value*

specifies the tolerance level (a small positive number between 0 and 1) for the stationary test. By default, EPS=0.001.

**MIXPRIORPARMS=K****MIXPRIORPARMS**(*value-list*)

specifies the parameters used in constructing the Dirichlet prior distribution for the mixing parameters. If you specify MIXPRIORPARMS=K, the parameters of the  $k$ -dimensional Dirichlet distribution are a vector that contains the number of components in the model ( $k$ ), whatever that might be. You can specify an explicit list of parameters in *value-list*. If the MIXPRIORPARMS option is not specified, the default Dirichlet parameter vector is a vector of length  $k$  of ones. This results in a uniform prior over the unit simplex; for  $k=2$ , this is the uniform distribution. See the section “[Prior Distributions](#)” on page 4504 for the distribution function of the Dirichlet as used by the HPFMM procedure.

**ESTIMATE=MEAN | MAP**

determines which overall estimate is used, based on the posterior sample, in the computation of [OUTPUT](#) statistics and certain ODS graphics. By default, the arithmetic average of the (thinned) posterior sample is used. If you specify ESTIMATE=MAP, the parameter vector is used that corresponds to the maximum log posterior density in the posterior sample. In any event, a message is written to the SAS log if postprocessing results depend on a summary estimate of the posterior sample.

**INITIAL=DATA | MLE | MODE | RANDOM**

determines how initial values for the Markov chain are obtained. The default when a conjugate sampler is used is INITIAL=DATA, in which case the HPFMM procedure uses the same algorithm to obtain data-dependent starting values as it uses for maximum likelihood estimation. If no conjugate sampler is available or if you use the METROPOLIS option to explicitly request that it not be used, then the default is INITIAL=MLE, in which case the maximum likelihood estimates are used as the initial values. If the maximum likelihood optimization fails, the HPFMM procedure switches to the default INITIAL=DATA.



The options INITIAL=MODE and INITIAL=RANDOM use the mode and random draws from the prior distribution, respectively, to obtain initial values. If the mode does not exist or if it falls on the boundary of the parameter space, the prior mean is used instead.

### METROPOLIS

requests that the HPFMM procedure use the Metropolis-Hastings sampling algorithm based on Gamerman (1997), even in situations where a conjugate sampler is available.

### MUPRIORPARMS=*pair-specification*

#### MUPRIORPARMS(*pair-specification* ... *pair-specification*)

specifies the parameters for the means in homogeneous mixtures without regression coefficients. The *pair-specification* is of the form  $(a, b)$ , where  $a$  and  $b$  are the two parameters of the prior distribution, optionally delimited with a comma. The actual distribution of the parameter is implied by the distribution selected in the MODEL statement. For example, it is a normal distribution for a mixture of normals, a gamma distribution for a mixture of Poisson variables, a beta distribution for a mixture of binary variables, and an inverse gamma distribution for a mixture of exponential variables. This option overrides the PRIOROPTIONS option.

The parameters correspond as follows:

Beta:	The parameters correspond to the $\alpha$ and $\beta$ parameters of the beta prior distribution such that its mean is $\mu = \alpha/(\alpha + \beta)$ and its variance is $\mu(1 - \mu)/(\alpha + \beta + 1)$ .
Normal:	The parameters correspond to the mean and variance of the normal prior distribution.
Gamma:	The parameters correspond to the $\alpha$ and $\beta$ parameters of the gamma prior distribution such that its mean is $\alpha/\beta$ and its variance is $\alpha/\beta^2$ .
Inverse gamma:	The parameters correspond to the $\alpha$ and $\beta$ parameters of the inverse gamma prior distribution such that its mean is $\mu = \beta/(\alpha - 1)$ and its variance is $\mu^2/(\alpha - 2)$ .

The two techniques for specifying the prior parameters with the MUPRIORPARMS option are as follows:

- Specify an equal sign and a single pair of values:

```
proc hpfmm seed=12345;
  model y = / k=2;
  bayes mupriorparms=(0,50);
run;
```

- Specify a list of parameter pairs within parentheses:

```
proc hpfmm seed=12345;
  model y = / k=2;
  bayes mupriorparms( (.,.) (1.4,10.5));
run;
```

If you specify an invalid value (outside of the parameter space for the prior distribution), the HPFMM procedure chooses the default value and writes a message to the SAS log. If you want to use the default values for a particular parameter, you can also specify missing values in the *pair-specification*. For example, the preceding list specification assigns default values for the first component and uses the values 1.4 and 10.5 for the mean and variance of the normal prior distribution in the second component. The first example assigns a  $N(0, 50)$  prior distribution to the means in both components.

**NBI=*n***

specifies the number of burn-in samples. During the burn-in phase, chains are not saved. The default is NBI=2000.

**NMC=*n*****SAMPLE=*n***

specifies the number of Monte Carlo samples after the burn-in. Samples after the burn-in phase are saved unless they are thinned with the THIN= option. The default is NMC=10000.

**OUTPOST<(outpost-options)>=*data-set***

requests that the posterior sample be saved to a SAS data set. In addition to variables that contain log likelihood and log posterior values, the OUTPOST data set contains variables for the parameters. The variable names for the parameters are generic (Parm\_1, Parm\_2, ..., Parm\_p). The labels of the parameters are descriptive and correspond to the “Parameter Mapping” table that is produced when the OUTPOST= option is in effect.

You can specify the following *outpost-options* in parentheses:

**LOGPRIOR**

adds the value of the log prior distribution to the data set.

**NONSINGULAR | NONSING | COMPRESS**

eliminates parameters that correspond to singular columns in the design matrix (and were not sampled) from the posterior data set. This is the default.

**SINGULAR | SING**

adds columns of zeros to the data set in positions that correspond to singularities in the model or to parameters that were not sampled for other reasons. By default, these columns of zeros are not written to the posterior data set.

**PHIPRIORPARMS=*pair-specification*****PHIPRIORPARMS( *pair-specification* ... *pair-specification* )**

specifies the parameters for the inverse gamma prior distribution of the scale parameters ( $\phi$ 's) in the model. The *pair-specification* is of the form ( $a, b$ ), and the values are chosen such that the prior distribution has mean  $\mu = b/(a - 1)$  and variance  $\mu^2/(a - 2)$ .

The form of the PHIPRIORPARMS with an equal sign and a single pair is used to specify one pair of prior parameters that applies to all components in the mixture. For example:

```
proc hpfmm seed=12345;
  model y = / k=2;
  bayes phipriorparms=(2.001,1.001);
run;
```

The form with a list of pairs is used to specify different prior parameters for the scale parameters in different components. For example:

```
proc hpfmm seed=12345;
  model y = / k=2;
  bayes phipriorparms( (.,1.001) (3.001,2.001) );
run;
```

If you specify an invalid value (outside of the parameter space for the prior distribution), the HPFMM procedure chooses the default value and writes a message to the SAS log. If you want to use the default values for a particular parameter, you can also specify missing values in the *pair-specification*. For example, the preceding list specification assigns default values for the first component *a* prior parameter and uses the value 1.001 for the *b* prior parameter. The second pair assigns 3.001 and 2.001 for the *a* and *b* prior parameters, respectively.

#### **PRIOROPTIONS** <=>(prior-options)

#### **PRIOROPTS** <=>(prior-options)

specifies options related to the construction of the prior distribution and the choice of their parameters. Some *prior-options* apply only in particular models. The **BETAPRIORPARMS=** and **MUPRIORPARMS=** options override this option.

You can specify the following *prior-options*:

#### **CONDITIONAL** | **COND**

chooses a conditional prior specification for the homogeneous normal and *t* distribution response components. The default prior specification in these models is an independence prior where the mean of the *h*th component has prior  $\mu_h \sim N(a, b)$ . The conditional prior is characterized by  $\mu_h \sim N(a, \sigma_h^2/b)$ .

#### **DEPENDENT** | **DEP**

chooses a data-dependent prior for the homogeneous models without effects. The prior parameters *a* and *b* are chosen as follows, based on the distribution in the **MODEL** statement:

Binary and binomial:  $a = \bar{y}/(1 - \bar{y})$ ,  $b=1$ , and the prior distribution for the success probability is  $\text{beta}(a, b)$ .

Poisson:  $a = 1$ ,  $b = 1/\bar{y}$ , and the prior distribution for  $\mu$  is  $\text{gamma}(a, b)$ . See Frühwirth-Schnatter (2006, p. 280) and Viallefont, Richardson, and Greene (2002).

Exponential:  $a = 3$ ,  $b = 2\bar{y}$ , and the prior distribution for  $\mu$  is inverse gamma with parameters *a* and *b*.

Normal and  $t$ : Under the default independence prior, the prior distribution for  $\mu$  is  $N(\bar{y}, fs^2)$  where  $f$  is the variance factor from the **VAR=** option and

$$s^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

Under the default conditional prior specification, the prior for  $\mu_h$  is  $N(a, \sigma_h^2/b)$  where  $a = \bar{y}$  and  $b = 2.6/(\max\{y\} - \min\{y\})$ . The prior for the scale parameter is inverse gamma with parameters 1.28 and  $0.36s^2$ . For further details, see Raftery (1996) and Frühwirth-Schnatter (2006, p. 179).

#### **VAR=f**

specifies the variance for normal prior distributions. The default is VAR=1000. This factor is used, for example, in determining the prior variance of regression coefficients or in determining the prior variance of means in homogeneous mixtures of  $t$  or normal distributions (unless a data-dependent prior is used).

#### **MLE=<r>**

specifies that the prior distribution for regression variables be based on a multivariate normal distribution centered at the MLEs and whose dispersion is a multiple  $r$  of the asymptotic MLE covariance matrix. The default is MLE=10. In other words, if you specify PRIOROPTIONS(MLE), the HPFMM procedure chooses the prior distribution for the regression variables as  $N(\hat{\beta}, 10\text{Var}[\hat{\beta}])$  where  $\hat{\beta}$  is the vector of maximum likelihood estimates. The prior for the scale parameter is inverse gamma with parameters 1.28 and  $0.36s^2$  where

$$s^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

For further details, see Raftery (1996) and Frühwirth-Schnatter (2006, p. 179). If you specify PRIOROPTIONS(MLE) for the regression parameters, then the data-dependent prior is used for the scale parameter; see the PRIOROPTIONS(DEPENDENT) option above.

The MLE option is not available for mixture models in which the parameters are estimated directly on the data scale, such as homogeneous mixture models or mixtures of distributions without model effects for which a conjugate sampler is available. By using the **METROPOLIS** option, you can always force the HPFMM procedure to abandon a conjugate sampler in favor of a Metropolis-Hastings sampling algorithm to which the MLE option applies.

**STATISTICS** <(global-options)> = **ALL** | **NONE** | keyword | (keyword-list)

**SUMMARIES** <(global-options)> = **ALL** | **NONE** | keyword | (keyword-list)

controls the number of posterior statistics produced. Specifying STATISTICS=ALL is equivalent to specifying STATISTICS=(SUMMARY INTERVAL). To suppress the computation of posterior statistics, specify STATISTICS=NONE. The default is STATISTICS=(SUMMARY INTERVAL). See the section “[Summary Statistics](#)” on page 158 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for more details.

The *global-options* include the following:

**ALPHA=***numeric-list*

controls the coverage levels of the equal-tail credible intervals and the credible intervals of highest posterior density (HPD) credible intervals. The ALPHA= values must be between 0 and 1. Each ALPHA= value produces a pair of  $100(1 - \alpha)\%$  equal-tail and HPD credible intervals for each sampled parameter. The default is ALPHA=0.05, which results in 95% credible intervals for the parameters.

**PERCENT=***numeric-list*

requests the percentile points of the posterior samples. The values in *numeric-list* must be between 0 and 100. The default is PERCENT=(25 50 75), which yields for each parameter the 25th, 50th, and 75th percentiles, respectively.

The list of *keywords* includes the following:

**SUMMARY**

produces the means, standard deviations, and percentile points for the posterior samples. The default is to produce the 25th, 50th, and 75th percentiles; you can modify this list with the global PERCENT= option.

**INTERVAL**

produces equal-tail and HPD credible intervals. The default is to produce the 95% equal-tail credible intervals and 95% HPD credible intervals, but you can use the ALPHA= *global-option* to request credible intervals for any probabilities.

**THIN=***n*

**THINNING=***n*

controls the thinning of the Markov chain after the burn-in. Only one in every  $k$  samples is used when THIN= $k$ , and if NBI= $n_0$  and NMC= $n$ , the number of samples kept is

$$\left[ \frac{n_0 + n}{k} \right] - \left[ \frac{n_0}{k} \right]$$

where  $[a]$  represents the integer part of the number  $a$ . The default is THIN=1—that is, all samples are kept after the burn-in phase.

**TIMEINC=***n*

specifies a time interval in seconds to report progress during the burn-in and sampling phase. The time interval is approximate, since the minimum time interval in which the HPFMM procedure can respond depends on the multithreading configuration.

---

## BY Statement

**BY** *variables* ;

You can specify a BY statement in PROC HPFMM to obtain separate analyses of observations in groups that are defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement in the HPFMM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

---

## CLASS Statement

**CLASS** *variable* < (*options*) > . . . < *variable* < (*options*) > > < / *global-options* > ;

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. The CLASS statement must precede the **MODEL** statement.

The CLASS statement for SAS high-performance analytical procedures is documented in the section “CLASS Statement” (Chapter 3, *SAS/STAT User’s Guide: High-Performance Procedures*). The HPFMM procedure also supports the following *global-option* in the CLASS statement:

### UPCASE

uppercases the values of character-valued CLASS variables before levelizing them. For example, if the UPCASE option is in effect and a CLASS variable can take the values ‘a,’ ‘A,’ and ‘b,’ then ‘a’ and ‘A’ represent the same level and the CLASS variable is treated as having only two values: ‘A’ and ‘B.’

---

## FREQ Statement

**FREQ** *variable* ;

The *variable* in the FREQ statement identifies a numeric variable in the data set that contains the frequency of occurrence for each observation. SAS high-performance analytical procedures that support the FREQ statement treat each observation as if it appeared  $f$  times, where  $f$  is the value of the FREQ variable for the observation. If the frequency value is not an integer, it is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

---

## ID Statement

**ID** *variables* ;

The ID statement lists one or more variables from the input data set that are transferred to output data sets created by SAS high-performance analytical procedures, provided that the output data set produces one or more records per input observation.

For more information about the common ID statement in SAS high-performance analytical procedures, see the section “ID Statement” (Chapter 3, *SAS/STAT User’s Guide: High-Performance Procedures*).

---

## MODEL Statement

**MODEL** *response* <(response-options)> = <effects> </model-options> ;

**MODEL** *events/trials* = <effects> </model-options> ;

**MODEL** + <effects> </model-options> ;

The MODEL statement defines elements of the mixture model, such as the model effects, the distribution, and the link function. At least one MODEL statement is required. You can specify more than one MODEL statement. Each MODEL statement identifies one or more components of a mixture. For example, if components differ in their distributions, link functions, or regressor variables, then you can use separate MODEL statements to define the components. If the finite mixture model is homogeneous—in the sense that all components share the same regressors, distribution, and link function—then you can specify the mixture model with a single MODEL statement by using the **K=** option.

An intercept is included in each model by default. It can be removed with the **NOINT** option.

The dependent variable can be specified by using either the *response* syntax or the *events/trials* syntax. The *events/trials* syntax is specific to models for binomial-type data. A  $\text{binomial}(n, \pi)$  variable is the sum of  $n$  independent Bernoulli trials with event probability  $\pi$ . Each Bernoulli trial results in either an event or a nonevent (with probability  $1 - \pi$ ). The value of the second variable, *trials*, gives the number  $n$  of Bernoulli trials. The value of the first variable, *events*, is the number of events out of  $n$ . The values of both *events* and (*trials*–*events*) must be nonnegative, and the value of *trials* must be positive. Other distributions that allow the *events/trials* syntax are the beta-binomial distribution and the binomial cluster model.

If the *events/trials* syntax is used, the HPFMM procedure defaults to the binomial distribution. If you use the *response* syntax, the procedure defaults to the normal distribution unless the response variable is a character variable or listed in the **CLASS** statement.

The HPFMM procedure supports a continuation-style syntax in MODEL statements. Since a mixture has only one response variable, it is sufficient to specify the response variable in one MODEL statement. Other MODEL statements can use the continuation symbol “+” before the specification of effects. For example, the following statements fit a three-component binomial mixture model:

```
class A;
model y/n = x / k=2;
model      + A;
```

The first MODEL statement uses the “=” sign to separate response from effect information and specifies the response variable by using the *events/trials* syntax. This determines the distribution as binomial. This MODEL statement adds two components to the mixture models with different intercepts and regression slopes. The second MODEL statement adds another component to the mixture where the mean is a function of the classification main effect for variable A. The response is also binomial; it is a continuation from the previous MODEL statement.

There are two sets of options in the MODEL statement. The *response-options* determine how the HPFMM procedure models probabilities for binary data. The *model-options* control other aspects of model formation and inference. Table 55.4 summarizes the *response-options* and *model-options* available in the MODEL statement. These are subsequently discussed in detail in alphabetical order by option category.

**Table 55.4** Summary of MODEL Statement Options

Option	Description
<b>Response Variable Options</b>	
DESCENDING	Reverses the order of response categories
EVENT=	Specifies the event category in binary models
ORDER=	Specifies the sort order for the response variable
REFERENCE=	Specifies the reference category in categorical models
<b>Model Building</b>	
DIST=	Specifies the response distribution
LINK=	Specifies the link function
K=	Specifies the number of mixture components
KMAX=	Specifies the maximum number of mixture components
KMIN=	Specifies the minimum number of mixture components
KRESTART	Requests that the starting values for each analysis be determined separately instead of sequentially
NOINT	Excludes fixed-effect intercept from model
OFFSET=	Specifies the offset variable for linear predictor
<b>Statistical Computations and Output</b>	
ALPHA= $\alpha$	Determines the confidence level ( $1 - \alpha$ )
CL	Displays confidence limits for fixed-effects parameter estimates
EQUATE=	Imposes simple equality constraints on parameters in this model
LABEL=	Identifies the model



**Table 55.4** *continued*

Option	Description
PARMS	Provides starting values for the parameters in this model

## Response Variable Options

Response variable options determine how the HPFMM procedure models probabilities for binary data.

You can specify the following *response-options* by enclosing them in parentheses after the *response* variable. The default is ORDER=FORMATTED.

### DESCENDING

#### DESC

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC HPFMM orders the response categories according to the ORDER= option and then reverses that order.

#### EVENT='category' | keyword

specifies the event category for the binary response model. PROC HPFMM models the probability of the event category. You can specify the value (formatted, if a format is applied) of the event category in quotes, or you can specify one of the following *keywords*:

#### FIRST

designates the first ordered category as the event. This is the default.

#### LAST

designates the last ordered category as the event.

#### ORDER=order-type

specifies the sort order for the levels of the response variable. You can specify the following values for *order-type*:

#### DATA

sorts the levels by order of appearance in the input data set.

#### FORMATTED

sorts the levels by external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value.

#### FREQ

sorts the levels by descending frequency count; levels with the most observations come first in the order.

#### INTERNAL

sorts the levels by unformatted value.

**FREQDATA**

sorts the levels by order of descending frequency count, and within counts by order of appearance in the input data set when counts are tied.

**FREQFORMATTED**

sorts the levels by order of descending frequency count, and within counts by formatted value (as above) when counts are tied.

**FREQINTERNAL**

sorts the levels by order of descending frequency count, and within counts by unformatted value when counts are tied.

When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC HPFMM run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. If you specify the ORDER= option in the MODEL statement and the ORDER= option in the **CLASS** statement, the former takes precedence.

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL values, the sort order is machine-dependent.

For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

**REFERENCE**=*'category'* | *keyword*

**REF**=*'category'* | *keyword*

specifies the reference category for categorical models. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes, or you can specify one of the following *keywords*:

**FIRST**

designates the first ordered category as the reference category.

**LAST**

designates the last ordered category as the reference category. This is the default.

**Model Options**

**ALPHA**=*number*

requests that confidence intervals be constructed for each of the parameters with confidence level  $1 - \textit{number}$ . The value of *number* must be between 0 and 1; the default is 0.05.

**CL**

requests that confidence limits be constructed for each of the parameter estimates. The confidence level is 0.95 by default; this can be changed with the **ALPHA**= option.

**DISTRIBUTION=***keyword*

**DIST=***keyword*

specifies the probability distribution for a mixture component.

If you specify the DIST= option and you do not specify a link function with the LINK= option, a default link function is chosen according to [Table 55.5](#). If you do not specify a distribution, the HPFMM procedure defaults to the normal distribution for continuous response variables and to the binary distribution for classification or character variables, unless the *events/trial* syntax is used in the MODEL statement. If you choose the *events/trial* syntax, the HPFMM procedure defaults to the binomial distribution.

[Table 55.5](#) lists *keywords* that you can specify for the DISTRIBUTION= option and the corresponding default link functions. For generalized linear models with these distributions, you can find expressions for the log-likelihood functions in the section “[Log-Likelihood Functions for Response Distributions](#)” on page 4495.

**Table 55.5** Keyword Values of the DIST= Option

<i>keyword</i>	Alias	Distribution	Default Link Function
<b>BETA</b>		Beta	Logit
<b>BETABINOMIAL</b>	<b>BETABIN</b>	Beta-binomial	Logit
<b>BINARY</b>	<b>BERNOULLI</b>	Binary	Logit
<b>BINOMIAL</b>	<b>BIN</b>	Binomial	Logit
<b>BINOMCLUSTER</b>	<b>BINOMCLUS</b>	Binomial cluster	Logit
<b>CONSTANT</b> <(c)>	<b>DEGENERATE</b> <(c)>	Degenerate	N/A
<b>EXPONENTIAL</b>	<b>EXPO</b>	Exponential	Log
<b>FOLDEDNORMAL</b>	<b>FNORMAL</b>	Folded normal	Identity
<b>GAMMA</b>	<b>GAM</b>	Gamma	Log
<b>GAUSSIAN</b>	<b>NORMAL</b>	Normal	Identity
<b>GENPOISSON</b>	<b>GPOISSON</b>	Generalized Poisson	Log
<b>GEOMETRIC</b>	<b>GEOM</b>	Geometric	Log
<b>INVGAUSS</b>	<b>IGAUSSIAN, IG</b>	Inverse Gaussian	Inverse squared (power(-2))
<b>LOGNORMAL</b>	<b>LOGN</b>	Lognormal	Identity
<b>NEGBINOMIAL</b>	<b>NEGBIN, NB</b>	Negative binomial	Log
<b>POISSON</b>	<b>POI</b>	Poisson	Log
<b>T</b> <(v)>	<b>STUDENT</b> <(v)>	<i>t</i>	Identity
<b>TRUNCEXPO</b> <(a,b)>	<b>TEXPO</b> <(a,b)>	Truncated exponential	Log
<b>TRUNCLOGN</b> <(a,b)>	<b>TLOGN</b> <(a,b)>	Lognormal	Identity
<b>TRUNCNEGBIN</b>	<b>TNEGBIN, TNB</b>	Negative binomial	Log
<b>TRUNCNORMAL</b> <(a,b)>	<b>TNORMAL</b> <(a,b)>	Truncated normal	Identity
<b>TRUNCPOISSON</b>	<b>TPOISSON, TPOI</b>	Truncated Poisson	Log
<b>UNIFORM</b> <(a,b)>	<b>UNIF</b> <(a,b)>	Uniform	N/A
<b>WEIBULL</b>		Weibull	Log

Note that the PROC HPFMM default link for the gamma or exponential distribution is not the canonical

link (the reciprocal link).

The binomial cluster model is a two-component model described in Morel and Nagaraj (1993); Morel and Neerchal (1997); Neerchal and Morel (1998). See [Example 55.1](#) for an application of the binomial cluster model in a teratological experiment.

If the *events/trials* syntax is used, the default distribution is the binomial and only the following choices are available: DIST=BINOMIAL, DIST=BETABINOMIAL, and DIST=BINOMCLUSTER. The *trials* variable is ignored for all other distributions. This enables you to fit models in which some components have a binomial or binomial-like distribution. For example, suppose that variable *n* is a binomial denominator and variable *logn* is its logarithm. Then the following statements model a two-component mixture of a binomial and Poisson count model:

```
model y/n = ;
model      + / dist=Poisson offset=logn;
```

The **OFFSET=** option is used in the second MODEL statement to specify that the Poisson counts refer to different base counts, since the trial variable *n* is ignored in the second model.

If DIST=BINOMIAL is specified without the *events/trials* syntax, then *n*=1 is used for the default number of trials.

DIST=TRUNCNEGBIN and DIST=TRUNCPOISSON are zero-truncated versions of DIST=NEGBINOMIAL and DIST=POISSON, respectively—that is, only the value of 0 is excluded from the support.

For DIST=TRUNCEXPO, DIST=TRUNCLOGN, and DIST=TRUNCNORMAL, you must specify the lower (*a*) and upper (*b*) truncation points of the distribution. For example:

```
DIST=TRUNCEXPO<(a,b)>
DIST=TRUNCLOGN<(a,b)>
DIST=TRUNCNORMAL<(a,b)>
```

Each of these distributions is the conditional version of its corresponding nontruncated distribution that is confined to the support  $[a, b]$  (inclusive). You can specify a missing value (.) for either *a* or *b* to truncate only on the other side; that is, *a*=. indicates a right-truncated distribution, and *b*=. indicates a left-truncated distribution.

For several distribution specifications you can provide additional optional parameters to further define the distribution. These optional parameters are listed in the following:

**CONSTANT**<(c)> The number *c* specifies the value where the mass is concentrated. The default is DIST=CONSTANT(0), so you can add zero-inflation to any model by adding a [MODEL](#) statement with DIST=CONSTANT.

**T**<(ν)> The number *ν* specifies the degrees of freedom for the (shifted) *t* distribution. The default is DIST=T(3); this leads to a heavy-tailed distribution for which the variance is defined. See the section “[Log-Likelihood Functions for Response Distributions](#)” on page 4495 for the density function of the shifted *t*<sub>ν</sub> distribution.

**UNIFORM**<(a,b)> The values *a* and *b* define the support of the uniform distribution, *a* < *b*. By default, *a* = 0 and *b* = 1.

**EQUATE=MEAN | SCALE | NONE | EFFECTS**(*effect-list*)

specifies simple sets of parameter constraints across the components in a MODEL statement; the default is EQUATE=NONE. This option is available only for maximum likelihood estimation. If you specify EQUATE=MEAN, the parameters that determine the mean are reduced to a single set that is applicable to all components in the MODEL statement. If you specify EQUATE=SCALE, a single parameter represents the common scale for all components in the MODEL statement. The EFFECTS option enables you to force the parameters for the chosen model effects to be equal across components; however, the number of parameters is unaffected.

For example, the following statements fit a two-component multiple regression model in which the coefficients for variable *logd* vary by component and the intercepts and coefficients for variable *dose* are the same for the two components:

```
proc hpfmm;
  model num = dose logd / equate=effects(int dose) k=2;
run;
```

To fix all coefficients across the two components, you can write the MODEL statement as

```
model num = dose logd / equate=effects(int dose logd) k=2;
```

or

```
model num = dose logd / equate=mean k=2;
```

If you restrict all parameters in a  $k$ -component MODEL statement to be equal, the HPFMM procedure reduces the model to  $k=1$ .

**K=n****NUMBER=n**

specifies the number of components the MODEL statement contributes to the overall mixture. For the binomial cluster model, this option is not available, since this model is a two-component model by definition.

**KMAX=n**

specifies the maximum number of components the MODEL statement contributes to the overall mixture.

If the maximum number of components in the mixture, as determined by all KMAX= options, is larger than the minimum number of components, the HPFMM procedure fits all possible models and displays summary fit information for the sequence of evaluated models. The “best” model according to the **CRITERION=** option in the **PROC HPFMM** statement is then chosen, and the remaining output and analyses performed by PROC HPFMM pertain to this “best” model.

When you use MCMC methods to estimate the parameters of a mixture, you need to ensure that the chain for a given value of  $k$  has converged; otherwise, comparisons among models that have varying numbers of components might not be meaningful. You can use the **FITDETAILS** option to display summary and diagnostic information for the MCMC chains from each model.

If you specify the **KMIN=** option but not the **KMAX=** option, then the default value for the **KMAX=** option is the value of the **KMIN=** option (unless **KMIN=0**, in which case the **KMAX=** option is set to 1).

**KMIN=*n***

specifies the minimum number of components that the **MODEL** statement contributes to the overall mixture. When you use MCMC methods to estimate the parameters of a mixture, you need to ensure that the chain for a given value of *k* has converged; otherwise, comparisons among models that have varying numbers of components might not be meaningful.

**KRESTART**

requests that the starting values for each analysis (that is, for each unique number of components as determined by the **KMIN=** and **KMAX=** options) be determined separately, in the same way as if no other analyses were performed. If you do not specify the **KRESTART** option, then the starting values for each analysis are based on results from the previous analysis with one less component.

**LABEL='label'**

specifies an optional label for the model that is used to identify the model in printed output, on graphics, and in data sets created from ODS tables.

**LINK=keyword**

specifies the link function in the model. The *keywords* and expressions for the associated link functions are shown in Table 55.6.

**Table 55.6** Link Functions in MODEL Statement of the HPFMM Procedure

<b>LINK=</b>	<b>Alias</b>	<b>Link Function</b>	$g(\mu) = \eta =$
<b>CLOGLOG</b>	<b>CLL</b>	Complementary log-log	$\log(-\log(1 - \mu))$
<b>IDENTITY</b>	<b>ID</b>	Identity	$\mu$
<b>LOG</b>		Log	$\log(\mu)$
<b>LOGIT</b>		Logit	$\log(\mu/(1 - \mu))$
<b>LOGLOG</b>		Log-log	$-\log(-\log(\mu))$
<b>PROBIT</b>	<b>NORMIT</b>	Probit	$\Phi^{-1}(\mu)$
<b>POWER(<math>\lambda</math>)</b>	<b>POW(<math>\lambda</math>)</b>	Power with exponent $\lambda = \text{number}$	$\begin{cases} \mu^\lambda & \text{if } \lambda \neq 0 \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$
<b>POWERMINUS2</b>		Power with exponent -2	$1/\mu^2$
<b>RECIPROCAL</b>	<b>INVERSE</b>	Reciprocal	$1/\mu$

The default link functions for the various distributions are shown in Table 55.5.

**NOINT**

requests that no intercept be included in the model. An intercept is included by default, unless the distribution is **DIST=CONSTANT** or **DIST=UNIFORM**.

**OFFSET=variable**

specifies the offset variable function for the linear predictor in the model. An offset variable can be thought of as a regressor variable whose regression coefficient is known to be 1. For example, you can use an offset in a Poisson model when counts have been obtained in time intervals of different lengths. With a log link function, you can model the counts as Poisson variables with the logarithm of the time interval as the offset variable.

**PARAMETERS(parameter-specification)****PARMS(parameter-specification)**

specifies starting values for the model parameters. If no PARMS option is given, the HPFMM procedure determines starting values by a data-dependent algorithm. To determine initial values for the Markov chain with Bayes estimation, see also the **INITIAL=** option in the **BAYES** statement. The specification of the parameters takes the following form: parameters in the mean function precede the scale parameters, and parameters for different components are separated by commas.

The following statements specify starting parameters for a two-component normal model. The initial values for the intercepts are 1 and -3; the initial values for the variances are 0.5 and 4.

```
proc hpfmm;
  model y = / k=2 parms (1 0.5, -3 4);
run;
```

You can specify missing values for parameters whose starting values are to be determined by the default method. Only values for parameters that participate in the optimization are specified. The values for model effects are specified on the linear (linked) scale.

---

## OUTPUT Statement

**OUTPUT < OUT=SAS-data-set >**

< keyword< (keyword-options) > < =name > > ...  
 < keyword< (keyword-options) > < =name > > < / options > ;

The OUTPUT statement creates a data set that contains observationwise statistics that are computed after fitting the model. The variables in the input data set are *not* included in the output data set to avoid data duplication for large data sets; however, variables specified in the **ID statement** are included.

The output statistics are computed based on the parameter estimates of the converged model if the parameters are estimated by maximum likelihood. If a Bayesian analysis is performed, the output statistics are computed based on the arithmetic mean in the posterior sample. You can change to the maximum posterior estimate with the **ESTIMATE=MAP** option in the **BAYES** statement.

You can specify the following syntax elements in the OUTPUT statement before the slash (/).

**OUT=SAS-data-set**

specifies the name of the output data set. If the OUT= option is omitted, the procedure uses the **DATA***n* convention to name the output data set.

**keyword< (keyword-options) > < =name >**

specifies a statistic to include in the output data set and optionally assigns the variable the name *name*. If you do not provide a name, the HPFMM procedure assigns a default name based on the type of

statistic requested. If you provide a name for a statistic that leads to multiple output statistics, the name is modified to index the associated component number.

You can use the *keyword-options* to control which type of a particular statistic is computed. The following are valid values for *keyword* and *keyword-options*:

**PREDICTED<(COMPONENT | OVERALL)>**

**PRED<(COMPONENT | OVERALL)>**

**MEAN<(COMPONENT | OVERALL)>**

requests predicted values (predicted means) for the response variable. The predictions in the output data set are mapped onto the data scale in all cases except for a binomial or binary response with *events/trials* syntax and when **PREDTYPE=COUNT** has not been specified. In that case the predictions are predicted success probabilities.

The default is to compute the predicted value for the mixture (OVERALL). You can request predictions for the means of the component distributions by adding the COMPONENT suboption in parentheses. The predicted values for some distributions are not identical to the parameter modeled as  $\mu$ . For example, in the lognormal distribution the predicted mean is  $\exp\{\mu + 0.5\phi\}$  where  $\mu$  and  $\phi$  are the parameters of an underlying normal process; see the section “[Log-Likelihood Functions for Response Distributions](#)” on page 4495 for details.

**RESIDUAL<(COMPONENT | OVERALL)>**

**RESID<(COMPONENT | OVERALL)>**

requests residuals for the response or residuals in the component distributions. Only “raw” residuals on the data scale are computed (observed minus predicted).

**VARIANCE<(COMPONENT | OVERALL)>**

**VAR<(COMPONENT | OVERALL)>**

requests variances for the mixture or the component distributions.

**LOGLIKE<(COMPONENT | OVERALL)>**

**LOGL<(COMPONENT | OVERALL)>**

requests values of the log-likelihood function for the mixture or the components. For observations used in the analysis, the overall computed value is the observations’ contribution to the log likelihood; if a **FREQ** statement is present, the frequency is accounted for in the computed value. In other words, if all observations in the input data set have been used in the analysis, adding the value of the log-likelihood contributions in the OUTPUT data set produces the negative of the final objective function value in the “Iteration History” table. By default, the log-likelihood contribution to the mixture is computed. You can request the individual mixture component contributions with the COMPONENT suboption.

**MIXPROBS<(COMPONENT | MAX)>**

**MIXPROB<(COMPONENT | MAX)>**

**PRIOR<(COMPONENT | MAX)>**

**MIXWEIGHTS<(COMPONENT | MAX)>**

requests that the prior weights  $\pi_j(\mathbf{z}, \boldsymbol{\alpha}_j)$  be added to the OUTPUT data set. By default, the probabilities are output for all components. You can limit the output to a single statistic, the largest mixing probability, with the MAX suboption.



**NOTE:** The keyword “prior” is used here because of long-standing practice to refer to the mixing probabilities as prior weights. This must not be confused with the prior distribution and its parameters in a Bayesian analysis.

**POSTERIOR<(COMPONENT | MAX)>**

**POST<(COMPONENT | MAX)>**

**PROB<(COMPONENT | MAX)>**

requests that the posterior weights

$$\frac{\pi_j(\mathbf{z}, \boldsymbol{\alpha}_j) p_j(y; \mathbf{x}'_j \boldsymbol{\beta}_j, \phi_j)}{\sum_{j=1}^k \pi_j(\mathbf{z}, \boldsymbol{\alpha}_j) p_j(y; \mathbf{x}'_j \boldsymbol{\beta}_j, \phi_j)}$$

be added to the OUTPUT data set. By default, the probabilities are output for all components. You can limit the output to a single statistic, the largest posterior probability, with the MAX suboption.

**NOTE:** The keyword “posterior” is used here because of long-standing practice to refer to these probabilities as posterior probabilities. This must not be confused with the posterior distribution in a Bayesian analysis.

**LINP**

**XBETA**

requests that the linear predictors for the models be added to the OUTPUT data set.

**CLASS | CATEGORY | GROUP**

adds the estimated component membership to the OUTPUT data set. An observation is associated with the component that has the highest posterior probability.

**MAXPOST | MAXPROB**

adds the highest posterior probability to the OUTPUT data set.

A *keyword* can appear multiple times. For example, the following OUTPUT statement requests predicted values for the mixture in addition to the predicted means in the individual components:

```
output out=hpfmout pred=MixtureMean pred(component)=CompMean;
```

In a three-component model, this produces four variables in the hpfmout data set: MixtureMean, CompMean\_1, CompMean\_2, and CompMean\_3.

You can specify the following *options* in the OUTPUT statement after a slash (/).

**ALLSTATS**

requests that all statistics are computed. If you do not use a *keyword* to assign a name, the HPFMM procedure uses the default name.

**PREDTYPE=PROB | COUNT**

specifies the type of predicted values that are produced for a binomial or binary response with *events/trials* syntax. If PREDTYPE=PROB, the predicted values are success probabilities. If PREDTYPE=COUNT, the predicted values are success counts. The default is PREDTYPE=PROB.

## PERFORMANCE Statement

**PERFORMANCE** < *performance-options* > ;

The PERFORMANCE statement defines performance parameters for multithreaded and distributed computing, passes variables that describe the distributed computing environment, and requests detailed results about the performance characteristics of the HPFMM procedure.

You can also use the PERFORMANCE statement to control whether the HPFMM procedure executes in single-machine mode or distributed mode.

The PERFORMANCE statement is documented further in the section “PERFORMANCE Statement” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

## PROBMODEL Statement

**PROBMODEL** < *effects* > < / *probmodel-options* > ;

The PROBMODEL statement defines the model effects for the mixing probabilities and their link function and starting values. Model effects (other than the implied intercept) are not supported with Bayesian estimation. By default, the HPFMM procedure models mixing probabilities on the logit scale for two-component models and as generalized logit models in situations with more than two components. The PROBMODEL statement is not required.

The generalized logit model with  $k$  categories has a common vector of regressor or design variables,  $\mathbf{z}$ ,  $k - 1$  parameter vectors that vary with category, and one linear predictor whose value is constant. The constant linear predictor is assigned by the HPFMM procedure to the last component in the model, and its value is zero ( $\alpha_k = 0$ ). The probability of observing category  $1 \leq j \leq k$  is then

$$\pi_j(\mathbf{z}, \alpha_j) = \frac{\exp\{\mathbf{z}'\alpha_j\}}{\sum_{i=1}^k \exp\{\mathbf{z}'\alpha_i\}}$$

For  $k=2$ , the generalized logit model reduces to a model with the logit link (a logistic model); hence the attribute *generalized* logit.

By default, an intercept is included in the model for the mixing probabilities. If you suppress the intercept with the **NOINT** option, you must specify at least one effect in the statement.

You can specify the following *probmodel-options* in the PROBMODEL statement after the slash (/):

**ALPHA=***number*

requests that confidence intervals that have the confidence level  $1 - \textit{number}$  be constructed for the parameters in the probability model. The value of *number* must be between 0 and 1; the default is 0.05. If the probability model is simple—that is, it does not contain any effects—the confidence intervals are produced for the estimated parameters (on the logit scale) and for the mixing probabilities. This option has no effect when you perform Bayesian estimation. You can modify credible interval settings by specifying the **STATISTICS(ALPHA=)** option in the **BAYES** statement.

**CL**

requests that confidence limits be constructed for each of the parameter estimates. The confidence level is 0.95 by default; this can be changed with the **ALPHA=** option.

**LINK=keyword**

specifies the link function in the model for the mixing probabilities. The default is a logit link for models with two components. For models with more than two components, only the generalized logit link is available. The *keywords* and expressions for the associated link functions for two-component models are shown in [Table 55.7](#).

**Table 55.7** Link Functions in the PROBMODEL Statement

<b>LINK=</b>	<b>Link Function</b>	$g(\mu) = \eta =$
<b>CLOGLOG</b>   <b>CLL</b>	Complementary log-log	$\log(-\log(1 - \mu))$
<b>LOGIT</b>	Logit	$\log(\mu/(1 - \mu))$
<b>LOGLOG</b>	Log-log	$-\log(-\log(\mu))$
<b>PROBIT</b>   <b>NORMIT</b>	Probit	$\Phi^{-1}(\mu)$

**NOINT**

requests that no intercept be included in the model for the mixing probabilities. An intercept is included by default. If you suppress the intercept with the **NOINT** option, you must specify at least one other effect for the mixing probabilities—since an empty probability model is not meaningful.

**PARAMETERS**(*parameter-specification*)**PARMS**(*parameter-specification*)

specifies starting values for the parameters. The specification of the parameters takes the following form: parameters in the mean function appear in a list, and parameters for different components are separated by commas. Starting values are given on the linked scale, not in terms of probabilities. Also, you need to specify starting values for each of the first  $k-1$  components in a  $k$ -component model. The linear predictor for the last component is always assumed to be zero.

The following statements specify a three-component mixture of multiple regression models. The PROBMODEL statement does not list any effects, a standard “intercept-only” generalized logit model is used to model the mixing probabilities.

```
proc hpfmm;
  model y = x1 x2 / k=3;
  probmodel / parms (2, 1);
run;
```

There are three linear predictors in the model for the mixing probabilities,  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ . With starting values of  $\alpha_1 = 2$ ,  $\alpha_2 = 1$ , and  $\alpha_3 = 0$ , this leads to initial mixing probabilities of

$$\pi_1 = \frac{e^2}{e^2 + e^1 + e^0} = 0.24$$

$$\pi_2 = \frac{e^1}{e^2 + e^1 + e^0} = 0.66$$

$$\pi_3 = \frac{e^0}{e^2 + e^1 + e^0} = 0.1$$

You can specify missing values for parameters whose starting values are to be determined by the default method.

---

## RESTRICT Statement

**RESTRICT** < 'label' > *constraint-specification* < , ... , *constraint-specification* >  
 < operator < value > > < / option > ;

The RESTRICT statement enables you to specify linear equality or inequality constraints among the parameters of a mixture model. These restrictions are incorporated into the maximum likelihood analysis. The RESTRICT statement is not available for a Bayesian analysis with the HPFMM procedure.

Following are reasons why you might want to place constraints and restrictions on the model parameters:

- to fix a parameter at a particular value
- to equate parameters in different components in a mixture
- to impose order conditions on the parameters in a model
- to specify contrasts among the parameters that the fitted model should honor

A restriction is composed of a left-hand side and a right-hand side, separated by an operator. If the operator and right-hand side are not specified, the restriction is assumed to be an equality constraint against zero. If the right-hand side is not specified, the value is assumed to be zero.

An individual *constraint-specification* is written in (nearly) the same form as estimable linear functions are specified in the ESTIMATE statement of the GLM, MIXED, or GLIMMIX procedure. The *constraint-specification* takes the form

*model-effect value-list* < ... *model-effect value-list* > < (SCALE= value) >

At least one *model-effect* must be specified followed by one or more values in the *value-list*. The values in the list correspond to the multipliers of the corresponding parameter that is associated with the position in the model effect. If you specify more values in the *value-list* than the *model-effect* occupies in the model design matrix, the extra coefficients are ignored.

To specify restrictions for effects in specific components in the model, separate the *constraint-specification* by commas. The following statements provide an example:

```

proc hpfmm;
  class A;
  model y/n = A x / k = 2;
  restrict A 1 0 -1;
  restrict x 2, x -1 >= 0.5;
run;

```

The linear predictors for this two-component model can be written as

$$\eta_1 = \beta_{10} + \alpha_{11}A_1 + \cdots + \alpha_{1a}A_a + x\beta_{11}$$

$$\eta_2 = \beta_{20} + \alpha_{21}A_1 + \cdots + \alpha_{2a}A_a + x\beta_{21}$$

where  $A_k$  is the binary variable associated with the  $k$ th level of A.

The first RESTRICT statement applies only to the first component and specifies that the parameter estimates that are associated with the first and third level of the A effect are identical. In terms of the linear predictor, the restriction can be written as

$$\alpha_{11} - \alpha_{13} = 0$$

Now suppose that A has only two levels. Then the HPFMM procedure ignores the value  $-1$  in the first RESTRICT statement and imposes the restriction

$$\alpha_{11} = 0$$

on the fitted model.

The second RESTRICT statement involves parameters in two different components of the model. In terms of the linear predictors, the restriction can be written as

$$2\beta_{11} - \beta_{21} \geq \frac{1}{2}$$

When restrictions are specified explicitly through the RESTRICT statement or implied through the [EQUATE=EFFECTS](#) option in the [MODEL](#) statement, the HPFMM procedure lists all restrictions after the model fit in a table of linear constraints and indicates whether a particular constraint is active at the converged solution.

The following operators can be specified to separate the left- and right-hand sides of the restriction: =, >, <, >=, <=.

Some distributions involve scale parameters (the parameter  $\phi$  in the expressions of the log likelihood) and you can also use the *constraint-specification* to involve a component's scale parameter in a constraint. To this end, assign a value to the keyword SCALE, separated from the model effects and value lists with parentheses. The following statements fit a two-component normal model and restrict the component variances to be equal:

```

proc hpfmm;
  model y = / k=2;
  restrict int 0 (scale 1),
           int 0 (scale -1);
run;

```

The intercept specification is necessary because each *constraint-specification* requires at least one model effect. The zero coefficient ensures that the intercepts are not involved in the restriction. Instead, the RESTRICT statement leads to  $\phi_1 - \phi_2 = 0$ .

You can specify the following *option* in the RESTRICT statement after a slash (/).

**DIVISOR=***value*

specifies a *value* by which all coefficients on the right-hand side and left-hand side of the restriction are divided.

---

## WEIGHT Statement

**WEIGHT** *variable* ;

The WEIGHT statement is used to perform a weighted analysis. Consult the section “[Log-Likelihood Functions for Response Distributions](#)” on page 4495 for expressions on how weight variables are included in the log-likelihood functions. Because the probability structure of a mixture model is different from that of a classical statistical model, the presence of a weight variable in a mixture model *cannot* be interpreted as altering the variance of an observation.

Observations with nonpositive or missing weights are not included in the PROC HPFMM analysis. If a WEIGHT statement is not included, all observations used in the analysis are assigned a weight of 1.

---

## Details: HPFMM Procedure

---

### A Gentle Introduction to Finite Mixture Models

#### The Form of the Finite Mixture Model

Suppose that you observe realizations of a random variable  $Y$ , the distribution of which depends on an unobservable random variable  $S$  that has a discrete distribution.  $S$  can occupy one of  $k$  states, the number of which might be unknown but is at least known to be finite. Since  $S$  is not observable, it is frequently referred to as a latent variable.

Let  $\pi_j$  denote the probability that  $S$  takes on state  $j$ . Conditional on  $S = j$ , the distribution of the response  $Y$  is assumed to be  $f_j(y; \alpha_j, \beta_j | S = j)$ . In other words, each distinct state  $j$  of the random variable  $S$  leads to a particular distributional form  $f_j$  and set of parameters  $\{\alpha_j, \beta_j\}$  for  $Y$ .

Let  $\{\alpha, \beta\}$  denote the collection of  $\alpha_j$  and  $\beta_j$  parameters across all  $j = 1$  to  $k$ . The marginal distribution of  $Y$  is obtained by summing the joint distribution of  $Y$  and  $S$  over the states in the support of  $S$ :

$$\begin{aligned} f(y; \alpha, \beta) &= \sum_{j=1}^k \Pr(S = j) f(y; \alpha_j, \beta_j | S = j) \\ &= \sum_{j=1}^k \pi_j f(y; \alpha_j, \beta_j | S = j) \end{aligned}$$

This is a mixture of distributions, and the  $\pi_j$  are called the mixture (or prior) probabilities. Because the number of states  $k$  of the latent variable  $S$  is finite, the entire model is termed a finite mixture (of distributions) model.

The finite mixture model can be expressed in a more general form by representing  $\alpha$  and  $\beta$  in terms of regressor variables and parameters with optional additional scale parameters for  $\beta$ . The section “[Notation for the Finite Mixture Model](#)” on page 4429 develops this in detail.

## Mixture Models Contrasted with Mixing and Mixed Models: Untangling the Terminology Web

Statistical terminology can have its limitations. The terms mixture, mixing, and mixed models are sometimes used interchangeably, causing confusion. Even worse, the terms arise in related situations. One application needs to be eliminated from the discussion in this documentation: mixture experiments, where design factors are the proportions with which components contribute to a blend, are not mixture models and do not fall under the purview of the HPFMM procedure. However, the data from a mixture experiment might be analyzed with a mixture model, a mixing model, or a mixed model, besides other types of statistical models.

Suppose that you observe realizations of random variable  $Y$  and assume that  $Y$  follows some distribution  $f(y; \alpha, \beta)$  that depends on parameters  $\alpha$  and  $\beta$ . Furthermore, suppose that the model is found to be deficient in the sense that the variability implied by the fitted model is less than the observed variability in the data, a condition known as overdispersion (see the section “[Overdispersion](#)” on page 4495). To tackle the problem the statistical model needs to be modified to allow for more variability. Clearly, one way of doing this is to introduce additional random variables into the process. Mixture, mixing, and mixed models are simply different ways of adding such random variables. The section “[The Form of the Finite Mixture Model](#)” on page 4492 explains how mixture models add a discrete state variable  $S$ . The following two subsections explain how mixing and mixed models instead assume variation for a natural parameter or in the mean function.

### Mixing Models

Suppose that the model is modified to allow for some random quantity  $U$ , which might be one of the parameters of the model or a quantity related to the parameters. Now there are two distributions to cope with: the conditional distribution of the response given the random effect  $U$ ,

$$f(y; \alpha, \beta | u)$$

and the marginal distribution of the data. If  $U$  is continuous, the marginal distribution is obtained by integration:

$$f(y; \alpha, \beta) = \int f(y; \alpha, \beta | u) f(u) du$$

Otherwise, it is obtained by summation over the support of  $U$ :

$$f(y; \alpha, \beta) = \sum_u \Pr(U = u) f(y; \alpha, \beta | u)$$

The important entity for statistical estimation is the marginal distribution  $f(y; \alpha, \beta)$ ; the conditional distribution is often important for model description, genesis, and interpretation.

In a mixing model the marginal distribution is known and is typically of a well-known form. For example, if  $Y|n$  has a binomial( $n, \mu$ ) distribution and  $n$  follows a Poisson distribution, then the marginal distribution of  $Y$  is Poisson. The preceding operation is called *mixing* a binomial distribution with a Poisson distribution. Similarly, when mixing a Poisson( $\lambda$ ) distribution with a gamma( $a, b$ ) distribution for  $\lambda$ , a negative

binomial distribution results as the marginal distribution. Other important mixing models involve mixing a  $\text{binomial}(n, \mu)$  random variable with a  $\text{beta}(a, b)$  distribution for the binomial success probability  $\mu$ . This results in a distribution known as the beta-binomial.

The finite mixtures have in common with the mixing models the introduction of random effects into the model to vary some or all of the parameters at random.

### **Mixed Models**

The difference between a mixing and a mixed model is that the conditional distribution is not that important in the mixing model. It matters to motivate the overdispersed reference model and to arrive at the marginal distribution. Inferences with respect to the conditional distribution, such as predicting the random variable  $U$ , are not performed in mixing models. In a mixed model the random variable  $U$  typically follows a continuous distribution—almost always a normal distribution. The random effects usually do not model the natural parameters of the distribution; instead, they are involved in linear predictors that relate to the conditional mean. For example, a linear mixed model is a model in which the response and the random effects are normally distributed, and the random effects enter the conditional mean function linearly:

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

The conditional and marginal distributions are then

$$\begin{aligned}\mathbf{Y}|\mathbf{U} &\sim N(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{U} + \boldsymbol{\epsilon}, \mathbf{R}) \\ \mathbf{Y} &\sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R})\end{aligned}$$

For this model, because of the linearity in the mean and the normality of the random effects, you could also refer to mixing the normal vector  $\mathbf{Y}$  with the normal vector  $\mathbf{U}$ , since the marginal distribution is known. The linear mixed model can be fit with the MIXED procedure. When the conditional distribution is not normal and the random effects are normal, the marginal distribution does not have a closed form. In this class of mixed models, called generalized linear mixed models, model approximations and numerical integration methods are commonly used in model fitting; see for example, those models fit by the GLIMMIX and NLMIXED procedures. Chapter 6, “[Introduction to Mixed Modeling Procedures](#),” contains details about the various classes of mixed models and about the relevant SAS/STAT procedures.

The previous expression for the marginal variance in the linear mixed model,  $\text{var}[\mathbf{Y}] = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$ , emphasizes again that the variability in the marginal distribution of a model that contains random effects exceeds the variability in a model without the random effects ( $\mathbf{R}$ ).

The finite mixtures have in common with the mixed models that the marginal distribution is not necessarily a well-known model, but is expressed through a formal integration over the random-effects distribution. In contrast to the mixed models, in particular those involving nonnormal distributions or nonlinear elements, this integration is rather trivial; it reduces to a weighted and finite sum of densities or mass functions.



## Overdispersion

Overdispersion is the condition by which the data are more dispersed than is permissible under a reference model. Overdispersion arises only if the variability a model can capture is limited (for example, because of a functional relationship between mean and variance). For example, a model for normal data can never be overdispersed in this sense, although the reasons that lead to overdispersion also negatively affect a misspecified model for normal data. For example, omitted variables increase the residual variance estimate because variability that should have been modeled through changes in the mean is now “picked up” as error variability.

Overdispersion is important because an overdispersed model can lead to misleading inferences and conclusions. However, diagnosing and remedying overdispersion is complicated. In order to handle it appropriately, the source of overdispersion must be identified. For example, overdispersion can arise from any of the following conditions alone or in combination:

- omitted variables and model effects
- omitted random effects (a source of random variation is not being modeled or is modeled as a fixed effect)
- correlation among the observations
- incorrect distributional assumptions
- incorrectly specified mean-variance relationships
- outliers in the data

As discussed in the previous section, introducing randomness into a system increases its variability. Mixture, mixed, and mixing models have thus been popular in modeling data that appear overdispersed. Finite mixture models are particularly powerful in this regard, because even low-order mixtures of basic, symmetric distributions (such as two- or three-component mixtures of normal or  $t$  distributions) enable you to model data with multiple modes, heavy tails, and skewness. In addition, the latent variable  $S$  provides a natural way to accommodate omitted, unobservable variables into the model.

One approach to remedy overdispersion is to apply simple modifications of the variance function of the reference model. For example, with binomial-type data this approach replaces the variance of the binomial count variable  $Y \sim \text{Binomial}(n, \mu)$ ,  $\text{Var}[Y] = n \times \mu(1 - \mu)$  with a scaled version,  $\phi n \times \mu(1 - \mu)$ , where  $\phi$  is called an overdispersion parameter,  $\phi > 0$ .

In addressing overdispersion problems, it is important to tackle the problem at its root. A missing scale factor on the variance function is hardly ever the root cause of overdispersion; it is only the easiest remedy.

---

## Log-Likelihood Functions for Response Distributions

The HPFMM procedure calculates the log likelihood that corresponds to a particular response distribution according to the following formulas. The response distribution is the distribution specified (or chosen by default) through the **DIST=** option in the **MODEL** statement. The parameterizations used for log-likelihood functions of these distributions were chosen to facilitate expressions in terms of mean parameters that are modeled through an (inverse) link functions and in terms of scale parameters. These are not

necessarily the parameterizations in which parameters of prior distributions are specified in a Bayesian analysis of homogeneous mixtures. See the section “[Prior Distributions](#)” on page 4504 for details about the parameterizations of prior distributions.

The HPFMM procedure includes all constant terms in the computation of densities or mass functions. In the expressions that follow,  $l$  denotes the log-likelihood function,  $\phi$  denotes a general scale parameter,  $\mu_i$  is the “mean”, and  $w_i$  is a weight from the use of a [WEIGHT](#) statement.

For some distributions (for example, the Weibull distribution)  $\mu_i$  is not the mean of the distribution. The parameter  $\mu_i$  is the quantity that is modeled as  $g^{-1}(\mathbf{x}'\boldsymbol{\beta})$ , where  $g^{-1}(\cdot)$  is the inverse link function and the  $\mathbf{x}$  vector is constructed based on the effects in the [MODEL](#) statement. Situations in which the parameter  $\mu$  does not represent the mean of the distribution are explicitly mentioned in the list that follows.

The parameter  $\phi$  is frequently labeled as a “Scale” parameter in output from the HPFMM procedure. It is not necessarily the scale parameter of the particular distribution.

### Beta( $\mu, \phi$ )

$$\begin{aligned} l(\mu_i, \phi; y_i, w_i) = & \log \left\{ \frac{\Gamma(\phi/w_i)}{\Gamma(\mu_i\phi/w_i)\Gamma((1-\mu_i)\phi/w_i)} \right\} \\ & + (\mu_i\phi/w_i - 1) \log\{y_i\} \\ & + ((1-\mu_i)\phi/w_i - 1) \log\{1 - y_i\} \end{aligned}$$

This parameterization of the beta distribution is due to Ferrari and Cribari-Neto (2004) and has properties  $E[Y] = \mu$ ,  $\text{Var}[Y] = \mu(1 - \mu)/(1 + \phi)$ ,  $\phi > 0$ .

### Beta-binomial( $n; \mu, \phi$ )

$$\begin{aligned} \phi &= (1 - \rho^2)/\rho^2 \\ l(\mu_i, \rho; y_i) &= \log\{\Gamma(n_i + 1)\} - \log\{\Gamma(y_i + 1)\} \\ &\quad - \log\{\Gamma(n_i - y_i + 1)\} \\ &\quad + \log\{\Gamma(\phi)\} - \log\{\Gamma(n_i + \phi)\} + \log\{\Gamma(y_i + \phi\mu_i)\} \\ &\quad + \log\{\Gamma(n_i - y_i + \phi(1 - \mu_i))\} - \log\{\Gamma(\phi\mu_i)\} \\ &\quad - \log\{\Gamma(\phi(1 - \mu_i))\} \\ l(\mu_i, \rho; y_i, w_i) &= w_i l(\mu_i, \rho; y_i) \end{aligned}$$

where  $y_i$  and  $n_i$  are the *events* and *trials* in the *events/trials* syntax and  $0 < \mu < 1$ . This parameterization of the beta-binomial model presents the distribution as a special case of the Dirichlet-Multinomial distribution—see, for example, Neerchal and Morel (1998). In this parameterization,  $E[Y] = n\mu$  and  $\text{Var}[Y] = n\mu(1 - \mu)(1 + (n - 1)/(\phi + 1))$ ,  $0 \leq \rho \leq 1$ . The HPFMM procedure models the parameter  $\phi$  and labels it “Scale” on the procedure output. For other parameterizations of the beta-binomial model, see Griffiths (1973) or Williams (1975).

### Binomial( $n; \mu$ )

$$\begin{aligned} l(\mu_i; y_i) &= y_i \log\{\mu_i\} + (n_i - y_i) \log\{1 - \mu_i\} \\ &\quad + \log\{\Gamma(n_i + 1)\} - \log\{\Gamma(y_i + 1)\} \\ &\quad - \log\{\Gamma(n_i - y_i + 1)\} \\ l(\mu_i; y_i, w_i) &= w_i l(\mu_i; y_i) \end{aligned}$$

where  $y_i$  and  $n_i$  are the *events* and *trials* in the *events/trials* syntax and  $0 < \mu < 1$ . In this parameterization  $E[Y] = n\mu$ ,  $\text{Var}[Y] = n\mu(1 - \mu)$ .

### Binomial cluster( $n; \mu, \pi$ )

$$\begin{aligned} z &= \log\{\Gamma(n_i + 1)\} - \log\{\Gamma(y_i + 1)\} - \log\{\Gamma(n_i - y_i + 1)\} \\ \mu_i^* &= (1 - \mu_i)\pi \\ l(\mu_i, \pi; y_i) &= \log\{\pi\} + z + y_i \log\{\mu_i^* + \mu_i\} \\ &\quad + (n_i - y_i) \log\{1 - \mu_i^* - \mu_i\} \\ &\quad + \log\{1 - \pi\} + z + y_i \log\{\mu_i^*\} \\ &\quad + (n_i - y_i) \log\{1 - \mu_i^*\} \\ l(\mu_i, \pi; y_i, w_i) &= w_i l(\mu_i, \pi; y_i) \end{aligned}$$

In this parameterization,  $E[Y] = n\pi$  and  $\text{Var}[Y] = n\pi(1 - \pi) \{1 + \mu^2(n - 1)\}$ . The binomial cluster model is a two-component mixture of a  $\text{binomial}(n, \mu^* + \mu)$  and a  $\text{binomial}(n, \mu^*)$  random variable. This mixture is unusual in that it fixes the number of components and because the mixing probability  $\pi$  appears in the moments of the mixture components. For further details, see Morel and Nagaraj (1993); Morel and Neerchal (1997); Neerchal and Morel (1998) and [Example 55.1](#) in this chapter. The expressions for the mean and variance in the binomial cluster model are identical to those of the beta-binomial model shown previously, with  $\pi_{bc} = \mu_{bb}$ ,  $\mu_{bc} = \rho_{bb}$ .

The HPFMM procedure models the parameter  $\mu$  through the [MODEL](#) statement and the parameter  $\pi$  through the [PROBMODEL](#) statement.

### Constant( $c$ )

$$l(y_i) = \begin{cases} 0 & y_i = c \\ -1\text{E}20 & y_i \neq c \end{cases}$$

The extreme value when  $y_i \neq c$  is chosen so that  $\exp\{l(y_i)\}$  yields a likelihood of zero. You can change this value with the [INVALIDLOGL=](#) option in the [PROC HPFMM](#) statement. The constant distribution is useful for modeling overdispersion due to zero-inflation (or inflation of the process at support  $c$ ).

The [DIST=CONSTANT](#) distribution is useful for modeling an inflated probability of observing a particular value (zero, by default) in data from other discrete distributions, as demonstrated in “[Modeling Zero-Inflation: Is It Better to Fish Poorly or Not to Have Fished at All?](#)” on page 4436. While it is syntactically valid to mix a constant distribution with a continuous distribution, such as [DIST=LOGNORMAL](#), such a mixture is not mathematically appropriate, because the constant log-likelihood is the log of a probability, while a continuous log-likelihood is the log of a probability density function. If you want to mix a constant distribution with a continuous distribution, you could model the constant as a very narrow continuous distribution, such as [DIST=UNIFORM\( \$c - \delta, c + \delta\$ \)](#) for a small value  $\delta$ . However, using [PROC HPFMM](#) to analyze such mixtures is sensitive to numerical inaccuracy and ultimately unnecessary. Instead, the following approach is mathematically equivalent and more numerically stable:

1. Estimate the mixing probability  $P(Y = c)$  as the proportion of observations in the data set such that  $|y_i - c| < \epsilon$ .

2. Estimate the parameters of the continuous distribution from the observations for which  $|y_i - c| \geq \epsilon$ .

**Exponential( $\mu$ )**

$$l(\mu_i; y_i, w_i) = \begin{cases} -\log\{\mu_i\} - y_i/\mu_i & w_i = 1 \\ w_i \log\left\{\frac{w_i y_i}{\mu_i}\right\} - \frac{w_i y_i}{\mu_i} - \log\{y_i \Gamma(w_i)\} & w_i \neq 1 \end{cases}$$

In this parameterization,  $E[Y] = \mu$  and  $\text{Var}[Y] = \mu^2$ .

**Folded normal( $\mu, \phi$ )**

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2} \log\{2\pi\} - \frac{1}{2} \log\{\phi/w_i\} \\ + \log \left\{ \exp \left\{ \frac{-w_i (y_i - \mu_i)^2}{2\phi} \right\} + \exp \left\{ \frac{-w_i (y_i + \mu_i)^2}{2\phi} \right\} \right\}$$

If  $X$  has a normal distribution with mean  $\mu$  and variance  $\phi$ , then  $Y = |X|$  has a folded normal distribution and log-likelihood function  $l(\mu, \phi; y, w)$  for  $y \geq 0$ . The folded normal distribution arises, for example, when normally distributed measurements are observed, but their signs are not observed. The mean and variance of the folded normal in terms of the underlying  $N(\mu, \phi)$  distribution are

$$E[Y] = \frac{1}{\sqrt{2\pi\phi}} \exp \left\{ -\frac{\mu^2}{2\phi} \right\} + \mu \left( 1 - 2\Phi \left( -\mu/\sqrt{\phi} \right) \right) \\ \text{Var}[Y] = \phi + \mu^2 - E[Y]^2$$

The HPFMM procedure models the folded normal distribution through the mean  $\mu$  and variance  $\phi$  of the underlying normal distribution. When the HPFMM procedure computes output statistics for the response variable (for example when you use the OUTPUT statement), the mean and variance of the response  $Y$  are reported. Similarly, the fit statistics apply to the distribution of  $Y = |X|$ , not the distribution of  $X$ . When you model a folded normal variable, the response input variable should be positive; the HPFMM procedure treats negative values of  $Y$  as a support violation.

**Gamma( $\mu, \phi$ )**

$$l(\mu_i, \phi; y_i, w_i) = w_i \phi \log \left\{ \frac{w_i y_i \phi}{\mu_i} \right\} - \frac{w_i y_i \phi}{\mu_i} - \log\{y_i\} - \log\{\Gamma(w_i \phi)\}$$

In this parameterization,  $E[Y] = \mu$  and  $\text{Var}[Y] = \mu^2/\phi$ ,  $\phi > 0$ . This parameterization of the gamma distribution differs from that in the GLIMMIX procedure, which expresses the log-likelihood function in terms of  $1/\phi$  in order to achieve a variance function suitable for mixed model analysis.

**Geometric( $\mu$ )**

$$l(\mu_i; y_i, w_i) = y_i \log \left\{ \frac{\mu_i}{w_i} \right\} - (y_i + w_i) \log \left\{ 1 + \frac{\mu_i}{w_i} \right\} \\ + \log \left\{ \frac{\Gamma(y_i + w_i)}{\Gamma(w_i) \Gamma(y_i + 1)} \right\}$$

In this parameterization,  $E[Y] = \mu$  and  $\text{Var}[Y] = \mu + \mu^2$ . The geometric distribution is a special case of the negative binomial distribution with  $\phi = 1$ .

**Generalized Poisson**( $\mu, \phi$ )

$$\begin{aligned}\xi_i &= 1 - \exp\{-\phi\}/w_i \\ \mu_i^* &= \mu_i - \xi(\mu_i - y_i) \\ l(\mu_i^*, \xi_i; y_i, w_i) &= \log\{\mu_i^* - \xi_i y_i\} + (y_i - 1) \log\{\mu_i^*\} \\ &\quad - \mu_i^* - \log\{\Gamma(y_i + 1)\}\end{aligned}$$

In this parameterization,  $E[Y] = \mu$ ,  $\text{Var}[Y] = \mu/(1 - \xi)^2$ , and  $\phi \geq 0$ . The HPFMM procedure models the mean  $\mu$  through the effects in the **MODEL** statement and applies a log link by default. The generalized Poisson distribution provides an overdispersed alternative to the Poisson distribution;  $\phi = \xi_i = 0$  produces the mass function of a regular Poisson random variable. For details about the generalized Poisson distribution and a comparison with the negative binomial distribution, see Joe and Zhu (2005).

**Inverse Gaussian**( $\mu, \phi$ )

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2} \left[ \frac{w_i (y_i - \mu_i)^2}{y_i \phi \mu_i^2} + \log \left\{ \frac{\phi y_i^3}{w_i} \right\} + \log\{2\pi\} \right]$$

The variance is  $\text{Var}[Y] = \phi \mu^3$ ,  $\phi > 0$ .

**Lognormal**( $\mu, \phi$ )

$$\begin{aligned}z_i &= \log\{y_i\} - \mu_i \\ l(\mu_i, \phi; y_i, w_i) &= -\frac{1}{2} \left( 2 \log\{y_i\} + \log \left\{ \frac{\phi}{w_i} \right\} + \log\{2\pi\} + \frac{w_i z_i^2}{\phi} \right)\end{aligned}$$

If  $X = \log\{Y\}$  has a normal distribution with mean  $\mu$  and variance  $\phi$ , then  $Y$  has the log-likelihood function  $l(\mu_i, \phi; y_i, w_i)$ . The HPFMM procedure models the lognormal distribution and not the “shortcut” version you can obtain by taking the logarithm of a random variable and modeling that as normally distributed. The two approaches are not equivalent, and the approach taken by PROC HPFMM is the actual lognormal distribution. Although the lognormal model is a member of the exponential family of distributions, it is not in the “natural” exponential family because it cannot be written in canonical form.

In terms of the parameters  $\mu$  and  $\phi$  of the underlying normal process for  $X$ , the mean and variance of  $Y$  are  $E[Y] = \exp\{\mu\} \sqrt{\omega}$  and  $\text{Var}[Y] = \exp\{2\mu\} \omega(\omega - 1)$ , respectively, where  $\omega = \exp\{\phi\}$ . When you request predicted values with the **OUTPUT** statement, the HPFMM procedure computes  $E[Y]$  and not  $\mu$ .

**Negative binomial**( $\mu, \phi$ )

$$\begin{aligned}l(\mu_i, \phi; y_i, w_i) &= y_i \log \left\{ \frac{\phi \mu_i}{w_i} \right\} - (y_i + w_i/\phi) \log \left\{ 1 + \frac{\phi \mu_i}{w_i} \right\} \\ &\quad + \log \left\{ \frac{\Gamma(y_i + w_i/\phi)}{\Gamma(w_i/\phi) \Gamma(y_i + 1)} \right\}\end{aligned}$$

The variance is  $\text{Var}[Y] = \mu + \phi \mu^2$ ,  $\phi > 0$ .

For a given  $\phi$ , the negative binomial distribution is a member of the exponential family. The parameter  $\phi$  is related to the scale of the data because it is part of the variance function. However, it cannot be factored from the variance, as is the case with the  $\phi$  parameter in many other distributions.

**Normal**( $\mu, \phi$ )

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2} \left[ \frac{w_i (y_i - \mu_i)^2}{\phi} + \log \left\{ \frac{\phi}{w_i} \right\} + \log \{2\pi\} \right]$$

The mean and variance are  $E[Y] = \mu$  and  $\text{Var}[Y] = \phi$ , respectively,  $\phi > 0$

**Poisson**( $\mu$ )

$$l(\mu_i; y_i, w_i) = w_i (y_i \log \{\mu_i\} - \mu_i - \log \{\Gamma(y_i + 1)\})$$

The mean and variance are  $E[Y] = \mu$  and  $\text{Var}[Y] = \mu$ .

**(Shifted) T**( $\nu; \mu, \phi$ )

$$\begin{aligned} z_i &= -0.5 \log \{\phi/w_i\} + \log \{\Gamma(0.5(\nu + 1))\} \\ &\quad - \log \{\Gamma(0.5\nu)\} - 0.5 \times \log \{\pi\nu\} \\ l(\mu_i, \phi; y_i, w_i) &= -\left(\frac{\nu + 1}{2}\right) \log \left\{ 1 + \frac{w_i (y_i - \mu_i)^2}{\nu \phi} \right\} + z_i \end{aligned}$$

$\nu > 0, \phi > 0$ . In this parameterization,  $E[Y] = \mu$  when  $\nu > 1$  and  $\text{Var}[Y] = \phi\nu/(\nu - 2)$  when  $\nu > 2$ . Note that this form of the  $t$  distribution is not a noncentral distribution, but that of a shifted central  $t$  random variable.

**Truncated Exponential**( $\mu; a, b$ )

$$\begin{aligned} l(\mu_i; a, b, y_i, w_i) &= w_i \log \left\{ \frac{w_i y_i}{\mu_i} \right\} - \frac{w_i y_i}{\mu_i} - \log \{y_i \Gamma(w_i)\} \\ &\quad - \log \left[ \frac{\gamma \left( w_i, \frac{w_i b}{\mu_i} \right)}{\Gamma(w_i)} - \frac{\gamma \left( w_i, \frac{w_i a}{\mu_i} \right)}{\Gamma(w_i)} \right] \end{aligned}$$

where

$$\gamma(c_1, c_2) = \int_0^{c_2} t^{c_1-1} \exp(-t) dt$$

is the lower incomplete gamma function. The mean and variance are

$$\begin{aligned} E[Y] &= \frac{(a + \mu_i) \exp(-a/\mu_i) - (b + \mu_i) \exp(-b/\mu_i)}{\exp(-a/\mu_i) - \exp(-b/\mu_i)} \\ \text{Var}[Y] &= \frac{(a^2 + 2a\mu_i + 2\mu_i^2) \exp(-a/\mu_i) - (b^2 + 2b\mu_i + 2\mu_i^2) \exp(-b/\mu_i)}{\exp(-a/\mu_i) - \exp(-b/\mu_i)} \\ &\quad - (E[Y])^2 \end{aligned}$$

**Truncated Lognormal**( $\mu, \phi; a, b$ )

$$\begin{aligned} z_i &= \log \{y_i\} - \mu_i \\ l(\mu_i, \phi; a, b, y_i, w_i) &= -\frac{1}{2} \left( 2 \log \{y_i\} + \log \left\{ \frac{\phi}{w_i} \right\} + \log \{2\pi\} + \frac{w_i z_i^2}{\phi} \right) \\ &\quad - \log \left\{ \Phi \left[ \sqrt{w_i/\phi} (\log b - \mu_i) \right] - \Phi \left[ \sqrt{w_i/\phi} (\log a - \mu_i) \right] \right\} \end{aligned}$$

where  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution. The mean and variance are

$$E[Y] = \exp(\mu_i + 0.5\phi) \frac{\Phi\left(\sqrt{\phi} - \frac{\log a - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\sqrt{\phi} - \frac{\log b - \mu_i}{\sqrt{\phi}}\right)}{\Phi\left(\frac{\log b - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\frac{\log a - \mu_i}{\sqrt{\phi}}\right)}$$

$$\text{Var}[Y] = \exp(2\mu_i + 2\phi) \frac{\Phi\left(2\sqrt{\phi} - \frac{\log a - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(2\sqrt{\phi} - \frac{\log b - \mu_i}{\sqrt{\phi}}\right)}{\Phi\left(\frac{\log b - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\frac{\log a - \mu_i}{\sqrt{\phi}}\right)} - (E[Y])^2$$

### Truncated Negative binomial( $\mu, \phi$ )

$$l(\mu_i, \phi; y_i, w_i) = y_i \log \left\{ \frac{\phi \mu_i}{w_i} \right\} - (y_i + w_i/\phi) \log \left\{ 1 + \frac{\phi \mu_i}{w_i} \right\}$$

$$+ \log \left\{ \frac{\Gamma(y_i + w_i/\phi)}{\Gamma(w_i/\phi) \Gamma(y_i + 1)} \right\}$$

$$- \log \left\{ 1 - \left( \frac{\phi \mu_i}{w_i} + 1 \right)^{-w_i/\phi} \right\}$$

The mean and variance are

$$E[Y] = \mu_i \left\{ 1 - (\phi \mu_i + 1)^{-1/\phi} \right\}^{-1}$$

$$\text{Var}[Y] = (1 + \phi \mu_i + \mu_i) E[Y] - (E[Y])^2$$

### Truncated Normal( $\mu, \phi; a, b$ )

$$l(\mu_i, \phi; a, b, y_i, w_i) = -\frac{1}{2} \left[ \frac{w_i (y_i - \mu_i)^2}{\phi} + \log \left\{ \frac{\phi}{w_i} \right\} + \log\{2\pi\} \right]$$

$$- \log \left\{ \Phi \left[ \sqrt{w_i/\phi} (b - \mu_i) \right] - \Phi \left[ \sqrt{w_i/\phi} (a - \mu_i) \right] \right\}$$

where  $\Phi(\cdot)$  is the cumulative distribution function of the standard normal distribution. The mean and variance are

$$E[Y] = \mu_i + \sqrt{\phi} \frac{\text{phi}\left(\frac{a - \mu_i}{\sqrt{\phi}}\right) - \text{phi}\left(\frac{b - \mu_i}{\sqrt{\phi}}\right)}{\Phi\left(\frac{b - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\frac{a - \mu_i}{\sqrt{\phi}}\right)}$$

$$\text{Var}[Y] = \phi \left[ 1 + \frac{\frac{a - \mu_i}{\sqrt{\phi}} \text{phi}\left(\frac{a - \mu_i}{\sqrt{\phi}}\right) - \frac{b - \mu_i}{\sqrt{\phi}} \text{phi}\left(\frac{b - \mu_i}{\sqrt{\phi}}\right)}{\Phi\left(\frac{b - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\frac{a - \mu_i}{\sqrt{\phi}}\right)} \right. \\ \left. - \left\{ \frac{\text{phi}\left(\frac{a - \mu_i}{\sqrt{\phi}}\right) - \text{phi}\left(\frac{b - \mu_i}{\sqrt{\phi}}\right)}{\Phi\left(\frac{b - \mu_i}{\sqrt{\phi}}\right) - \Phi\left(\frac{a - \mu_i}{\sqrt{\phi}}\right)} \right\}^2 \right]$$

where  $\text{phi}(\cdot)$  is the probability density function of the standard normal distribution.

**Truncated Poisson( $\mu$ )**

$$l(\mu_i; y_i, w_i) = w_i(y_i \log\{\mu_i\} - \log\{\exp(\mu_i) - 1\} - \log\{\Gamma(y_i + 1)\})$$

The mean and variance are

$$E[Y] = \frac{\mu}{1 - \exp(-\mu)}$$

$$\text{Var}[Y] = \frac{\mu [1 - \exp(-\mu) - \mu \exp(-\mu)]}{[1 - \exp(-\mu)]^2}$$

**Uniform( $a, b$ )**

$$l(\mu_i; y_i, w_i) = -\log\{b - a\}$$

The mean and variance are  $E[Y] = 0.5(a + b)$  and  $\text{Var}[Y] = (b - a)^2/12$ .

**Weibull( $\mu, \phi$ )**

$$l(\mu_i, \phi; y_i) = -\frac{\phi - 1}{\phi} \log \left\{ \frac{y_i}{\mu_i} \right\} - \log\{\mu_i \phi\}$$

$$- \exp \left\{ \log \left\{ \frac{y_i}{\mu_i} \right\} / \phi \right\}$$

In this particular parameterization of the two-parameter Weibull distribution, the mean and variance of the random variable  $Y$  are  $E[Y] = \mu\Gamma(1 + \phi)$  and  $\text{Var}[Y] = \mu^2 \{\Gamma(1 + 2\phi) - \Gamma^2(1 + \phi)\}$ .

---

## Bayesian Analysis

### Conjugate Sampling

The HPFMM procedure uses Bayesian analysis via a conjugate Gibbs sampler if the model belongs to a small class of mixture models for which a conjugate sampler is available. See the section “[Gibbs Sampler](#)” on page 139 in Chapter 7, “[Introduction to Bayesian Analysis Procedures](#),” for a general discussion of Gibbs sampling. [Table 55.8](#) summarizes the models for which conjugate and Metropolis-Hastings samplers are available.

**Table 55.8** Availability of Conjugate and Metropolis Samplers in the HPFMM Procedure

Effects (exclusive of intercept)	Distributions	Available Samplers
No	Normal or T	Conjugate or Metropolis-Hastings
Yes	Normal or T	Conjugate or Metropolis-Hastings
No	Binomial, binary, Poisson, exponential	Conjugate or Metropolis-Hastings
Yes	Binomial, binary, Poisson, exponential	Metropolis-Hastings only



The conjugate sampler enjoys greater efficiency than the Metropolis-Hastings sampler and has the advantage of sampling in terms of the natural parameters of the distribution.

You can always switch to the Metropolis-Hastings sampling algorithm in any model by adding the **METROPOLIS** option in the **BAYES** statement.

## Metropolis-Hastings Algorithm

If Metropolis-Hastings is the only sampler available for the specified model (see Table 55.8) or if the **METROPOLIS** option is specified in the **BAYES** statement, PROC HPFMM uses the Metropolis-Hastings approach of Gamerman (1997). See the section “Metropolis and Metropolis-Hastings Algorithms” on page 138 in Chapter 7, “Introduction to Bayesian Analysis Procedures,” for a general discussion of the Metropolis-Hastings algorithm.

The Gamerman (1997) algorithm derives a specific density that is used to generate proposals for the component-specific parameters  $\beta_j$ . The form of this proposal density is multivariate normal, with mean  $\mathbf{m}_j$  and covariance matrix  $\mathbf{C}_j$  derived as follows.

Suppose  $\beta_j$  is the vector of model coefficients in the  $j$ th component and suppose that  $\beta_j$  has prior distribution  $N(\mathbf{a}, \mathbf{R})$ . Consider a generalized linear model (GLM) with link function  $g(\mu) = \eta = \mathbf{x}'\beta$  and variance function  $a(\mu)$ . The pseudo-response and weight in the GLM for a weighted least squares step are

$$y^* = \eta + (y - \mu) / \frac{\partial \mu}{\partial \eta}$$

$$w = \frac{\partial \mu}{\partial \eta} / a(\mu)$$

If the model contains offsets or **FREQ** or **WEIGHT** statements, or if a *trials* variable is involved, suitable adjustments are made to these quantities.

In each component,  $j = 1, \dots, k$ , form an adjusted cross-product matrix with a “pseudo” border

$$\begin{bmatrix} \mathbf{X}_j' \mathbf{W}_j \mathbf{X}_j + \mathbf{R}^{-1} & \mathbf{X}_j' \mathbf{W}_j \mathbf{y}_j^* + \mathbf{R}^{-1} \mathbf{a} \\ \mathbf{y}_j^{*'} \mathbf{W}_j \mathbf{X}_j + \mathbf{a}' \mathbf{R}^{-1} & c \end{bmatrix}$$

where  $\mathbf{W}_j$  is a diagonal matrix formed from the pseudo-weights  $w$ ,  $\mathbf{y}^*$  is a vector of pseudo-responses, and  $c$  is arbitrary. This is basically a system of normal equations with ridging, and the degree of ridging is governed by the precision and mean of the normal prior distribution of the coefficients. Sweeping on the leading partition leads to

$$\mathbf{C}_j = (\mathbf{X}_j' \mathbf{W}_j \mathbf{X}_j + \mathbf{R}^{-1})^-$$

$$\mathbf{m}_j = \mathbf{C}_j (\mathbf{X}_j' \mathbf{W}_j \mathbf{y}_j^* + \mathbf{R}^{-1} \mathbf{a})$$

where the generalized inverse is a reflexive,  $g_2$ -inverse (see the section “Linear Model Theory” on page 55 in Chapter 3, “Introduction to Statistical Modeling with SAS/STAT Software,” for details).

PROC HPFMM then generates a proposed parameter vector from the resulting multivariate normal distribution, and then accepts or rejects this proposal according to the appropriate Metropolis-Hastings thresholds.

## Latent Variables via Data Augmentation

In order to fit finite Bayesian mixture models, the HPFMM procedure treats the mixture model as a missing data problem and introduces an assignment variable  $\mathbf{S}$  as in Dempster, Laird, and Rubin (1977). Since  $\mathbf{S}$  is not observable, it is frequently referred to as a latent variable. The unobservable variable  $\mathbf{S}$  assigns an observation to a component in the mixture model. The number of states,  $k$ , might be unknown, but it is known to be finite. Conditioning on the latent variable  $\mathbf{S}$ , the component memberships of each observation is assumed to be known, and Bayesian estimation is straightforward for each component in the finite mixture model. That is, conditional on  $S = j$ , the distribution of the response is now assumed to be  $f(y; \alpha_j, \beta_j | S = j)$ . In other words, each distinct state of the random variable  $\mathbf{S}$  leads to a distinct set of parameters. The parameters in each component individually are then updated using a conjugate Gibbs sampler (where available) or a Metropolis-Hastings sampling algorithm.

The HPFMM procedure assumes that the random variable  $\mathbf{S}$  has a discrete multinomial distribution with probability  $\pi_j$  of belonging to a component  $j$ ; it can occupy one of  $k$  states. The distribution for the latent variable  $\mathbf{S}$  is

$$f(S_i = j | \pi_1, \dots, \pi_k) = \text{multinomial}(1, \pi_1, \dots, \pi_k)$$

where  $f(\cdot | \cdot)$  denotes a conditional probability density. The parameters in the density  $\pi_j$  denote the probability that  $S$  takes on state  $j$ .

The HPFMM procedure assumes a conjugate Dirichlet prior distribution on the mixture proportions  $\pi_j$  written as:

$$p(\boldsymbol{\pi}) = \text{Dirichlet}(a_1, \dots, a_k)$$

where  $p(\cdot)$  indicates a prior distribution.

Using Bayes' theorem, the likelihood function and prior distributions determine a conditionally conjugate posterior distribution of  $\mathbf{S}$  and  $\boldsymbol{\pi}$  from the multinomial distribution and Dirichlet distribution, respectively.

## Prior Distributions

The following list displays the parameterization of prior distributions for situations in which the HPFMM procedure uses a conjugate sampler in mixture models without model effects and certain basic distributions (binary, binomial, exponential, Poisson, normal, and  $t$ ). You specify the parameters  $a$  and  $b$  in the formulas below in the **MUPRIORPARMS=** and **PHIPRIORPARMS=** options in the **BAYES** statement in these models.

### Beta( $a, b$ )

$$f(y) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} y^{a-1} (1-y)^{b-1}$$

where  $a > 0, b > 0$ . In this parameterization, the mean and variance of the distribution are  $\mu = a/(a+b)$  and  $\mu(1-\mu)/(a+b+1)$ , respectively. The beta distribution is the prior distribution for the success probability in binary and binomial distributions when conjugate sampling is used.

### Dirichlet( $a_1, \dots, a_k$ )

$$f(\mathbf{y}) = \frac{\Gamma\left(\sum_{i=1}^k a_i\right)}{\prod_{i=1}^k \Gamma(a_i)} y_1^{a_1-1} \dots y_k^{a_k-1}$$

where  $\sum_{i=1}^k y_i = 1$  and the parameters  $a_i > 0$ . If any  $a_i$  were zero, an improper density would result. The Dirichlet density is the prior distribution for the mixture probabilities. You can affect the choice of the  $a_i$  through the [MIXPRIORPARMS](#) option in the [BAYES](#) statement. If  $k=2$ , the Dirichlet is the same as the  $\text{beta}(a, b)$  distribution.

### **Gamma( $a, b$ )**

$$f(y) = \frac{b^a}{\Gamma(a)} y^{a-1} \exp\{-by\}$$

where  $a > 0, b > 0$ . In this parameterization, the mean and variance of the distribution are  $\mu = a/b$  and  $\mu/b$ , respectively. The gamma distribution is the prior distribution for the mean parameter of the Poisson distribution when conjugate sampling is used.

### **Inverse gamma( $a, b$ )**

$$f(y) = \frac{b^a}{\Gamma(a)} y^{-a-1} \exp\{-b/y\}$$

where  $a > 0, b > 0$ . In this parameterization, the mean and variance of the distribution are  $\mu = b/(a - 1)$  if  $a > 1$  and  $\mu^2/(a - 2)$  if  $a > 2$ , respectively. The inverse gamma distribution is the prior distribution for the mean parameter of the exponential distribution when conjugate sampling is used. It is also the prior distribution for the scale parameter  $\phi$  in all models.

### **Multinomial( $1, \pi_1, \dots, \pi_k$ )**

$$f(\mathbf{y}) = \frac{1}{y_1! \dots y_k!} \pi_1^{y_1} \dots \pi_k^{y_k}$$

where  $\sum_{j=1}^k y_j = n$ ,  $y_j \geq 0$ ,  $\sum_{j=1}^k \pi_j = 1$ , and  $n$  is the number of observations included in the analysis. The multinomial density is the prior distribution for the mixture proportions. The mean and variance of  $Y_j$  are  $\mu_j = \pi_j$  and  $\mu_j(1 - \mu_j)$ , respectively.

### **Normal( $a, b$ )**

$$f(y) = \frac{a}{\sqrt{2\pi b}} \exp\left\{-\frac{1}{2} \frac{(y - a)^2}{b}\right\}$$

where  $b > 0$ . The mean and variance of the distribution are  $\mu = a$  and  $b$ , respectively. The normal distribution is the prior distribution for the mean parameter of the normal and  $t$  distribution when conjugate sampling is used.

When a [MODEL](#) statement contains effects or if you specify the [METROPOLIS](#) option, the prior distribution for the regression parameters is multivariate normal, and you can specify the means and variances of the parameters in the [BETAPRIORPARMS=](#) option in the [BAYES](#) statement.

---

## **Parameterization of Model Effects**

PROC HPFMM constructs a finite mixture model according to the specifications in the [CLASS](#), [MODEL](#), and [PROBMODEL](#) statements. Each effect in the [MODEL](#) statement generates one or more columns in the matrix **X** for that model. The same **X** matrix applies to all components that are associated with the [MODEL](#)

statement. Each effect in the **PROBMODEL** statement generates one or more columns in the matrix **Z** from which the linear predictors in the model for the mixture probability models is formed. The same **Z** matrix applies to all components.

The formation of effects from continuous and classification variables in the HPFMM procedure follows the same general rules and techniques as for other linear modeling procedures. For information about constructing the model effects, see the section “Specification and Parameterization of Model Effects” (Chapter 3, *SAS/STAT User’s Guide: High-Performance Procedures*).

---

## Computational Method

### Multithreading

Threading refers to the organization of computational work into multiple tasks (processing units that can be scheduled by the operating system). A task is associated with a thread. Multithreading refers to the concurrent execution of threads. When multithreading is possible, substantial performance gains can be realized compared to sequential (single-threaded) execution.

The number of threads spawned by the HPFMM procedure is determined by the number of CPUs on a machine and can be controlled in the following ways:

- You can specify the CPU count with the **CPUCOUNT=** SAS system option. For example, if you specify the following statements, the HPFMM procedure schedules threads as if it were executing on a system that had four CPUs, regardless of the actual CPU count:

```
options cpucount=4;
```

- You can specify the **NTHREADS=** option in the **PERFORMANCE** statement to determine the number of threads. This specification overrides the system option. Specify **NTHREADS=1** to force single-threaded execution.

The number of threads per machine is displayed in the “Performance Information” table, which is part of the default output. The HPFMM procedure allocates one thread per CPU.

The tasks that are multithreaded by the HPFMM procedure are primarily defined by dividing the data processed on a single machine among the threads—that is, the HPFMM procedure implements multithreading through a data-parallel model. For example, if the input data set has 1,000 observations and you are running with four threads, then 250 observations are associated with each thread. All operations that require access to the data are then multithreaded. This operations include the following:

- variable levelization
- effect levelization
- formation of the crossproducts matrix
- objective function, gradient, and Hessian evaluations
- scoring of observations

In addition, operations on matrices such as sweeps might be multithreaded if the matrices are of sufficient size to realize performance benefits from managing multiple threads for the particular matrix operation.

## Choosing an Optimization Algorithm

### First- or Second-Order Algorithms

The factors that affect how you choose a particular optimization technique for a particular problem are complex. Occasionally, you might benefit from trying several algorithms.

For many optimization problems, computing the gradient takes more computer time than computing the function value. Computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that do use a Hessian matrix; as a result, the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can terminate more easily at stationary points than at global optima.

Table 55.9 shows which derivatives are required for each optimization technique.

**Table 55.9** Derivatives Required

Algorithm	First-Order	Second-Order
TRUREG	x	x
NEWRAP	x	x
NRRIDG	x	x
QUANEW	x	-
DBLDOG	x	-
CONGRA	x	-
LEVMAR	x	-
NMSIMP	-	-

The second-derivative methods (TRUREG, NEWRAP, and NRRIDG) are best for small problems for which the Hessian matrix is not expensive to compute. Sometimes the NRRIDG algorithm can be faster than the TRUREG algorithm, but TRUREG can be more stable. The NRRIDG algorithm requires only one matrix with  $p(p + 1)/2$  double words; TRUREG and NEWRAP require two such matrices. Here,  $p$  denotes the number of parameters in the optimization.

The first-derivative methods QUANEW and DBLDOG are best for medium-sized problems for which the objective function and the gradient are much faster to evaluate than the Hessian. In general, the QUANEW and DBLDOG algorithms require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG algorithms require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP.

The first-derivative method CONGRA is best for large problems for which the objective function and the gradient can be computed much faster than the Hessian and for which too much memory is required to store the (approximate) Hessian. In general, the CONGRA algorithm requires more iterations than QUANEW or

DBLDOG, but each iteration can be much faster. Because CONGRA requires only a factor of  $p$  double-word memory, many large applications can be solved only by CONGRA.

The no-derivative method NMSIMP is best for small problems for which derivatives are not continuous or are very difficult to compute.

The LEVMAR method is appropriate only for least squares optimization problems.

Each optimization method uses one or more convergence criteria that determine when it has converged. An algorithm is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW algorithm converges if  $\text{ABSGCONV} < 1\text{E}-5$ ,  $\text{FCONV} < 2 \times \epsilon$ , or  $\text{GCONV} < 1\text{E}-8$ .

By default, the HPFMM procedure applies the NRRIDG algorithm because it can take advantage of multi-threading in Hessian computations and inversions. If the number of parameters becomes large, specifying  $\text{TECHNIQUE}=\text{QUANEW}$  (which is a first-order method with good overall properties) is recommended.

## Algorithm Descriptions

The following subsections provide details about each optimization technique and follow the same order as Table 55.9.

### Trust Region Optimization (TRUEG)

The trust region method uses the gradient  $\mathbf{g}(\boldsymbol{\psi}^{(k)})$  and the Hessian matrix  $\mathbf{H}(\boldsymbol{\psi}^{(k)})$ ; thus, it requires that the objective function  $f(\boldsymbol{\psi})$  have continuous first- and second-order derivatives inside the feasible region.

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region that has radius  $\Delta$ . The radius constrains the step size that corresponds to the quality of the quadratic approximation. The trust region method is implemented based on Dennis, Gay, and Welsch (1981); Gay (1983); Moré and Sorensen (1983).

The trust region method performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the quasi-Newton or conjugate gradient algorithms might be more efficient.

### Newton-Raphson Optimization with Line Search (NEWRAP)

The NEWRAP technique uses the gradient  $\mathbf{g}(\boldsymbol{\psi}^{(k)})$  and the Hessian matrix  $\mathbf{H}(\boldsymbol{\psi}^{(k)})$ ; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

If second-order derivatives are computed efficiently and precisely, the NEWRAP method can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This algorithm uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive-definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive-definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation (LIS=2).

### **Newton-Raphson Ridge Optimization (NRRIDG)**

The NRRIDG technique uses the gradient  $\mathbf{g}(\boldsymbol{\psi}^{(k)})$  and the Hessian matrix  $\mathbf{H}(\boldsymbol{\psi}^{(k)})$ ; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This algorithm uses a pure Newton step when the Hessian is positive-definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

The NRRIDG method performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the quasi-Newton or conjugate gradient algorithms might be more efficient.

Because the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than an iteration of the NEWRAP technique, which works with a Cholesky decomposition. However, NRRIDG usually requires fewer iterations than NEWRAP.

### **Quasi-Newton Optimization (QUANEW)**

The (dual) quasi-Newton method uses the gradient  $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ , and it does not need to compute second-order derivatives because they are approximated. It works well for medium-sized to moderately large optimization problems, where the objective function and the gradient can be computed much faster than the Hessian. However, in general it requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. QUANEW is the default optimization algorithm because it provides an appropriate balance between the speed and stability that are required for most nonlinear mixed model applications.

The QUANEW technique that is implemented by the HPFMM procedure is the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size  $\alpha$  that satisfies the Goldstein conditions (Fletcher 1987). One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive-definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted by using an identity matrix, resulting in the steepest descent or ascent search direction.

The QUANEW algorithm uses its own line-search technique.

### **Double-Dogleg Optimization (DBLDOG)**

The double-dogleg optimization method combines the ideas of the quasi-Newton and trust region methods. In each iteration, the double-dogleg algorithm computes the step  $\mathbf{s}^{(k)}$  as the linear combination of the steepest descent or ascent search direction  $\mathbf{s}_1^{(k)}$  and a quasi-Newton search direction  $\mathbf{s}_2^{(k)}$ :

$$\mathbf{s}^{(k)} = \alpha_1 \mathbf{s}_1^{(k)} + \alpha_2 \mathbf{s}_2^{(k)}$$

The step is requested to remain within a prespecified trust region radius (Fletcher 1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search.

The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient are much faster to compute than the Hessian. The implementation is based on Dennis and Mei (1979); Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG,



NEWRAP, and NRRIDG techniques, which require second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for the update of the Cholesky factor of an approximate Hessian.

### **Conjugate Gradient Optimization (CONGRA)**

Second-order derivatives are not required by the CONGRA algorithm and are not even approximated. The CONGRA algorithm can be expensive in function and gradient calls, but it requires only  $O(p)$  memory for unconstrained optimization. In general, the algorithm must perform many iterations to obtain a precise solution, but each of the CONGRA iterations is computationally cheap.

The CONGRA algorithm should be used for optimization problems that have large  $p$ . For the unconstrained or boundary-constrained case, the CONGRA algorithm requires only  $O(p)$  bytes of working memory, whereas all other optimization methods require order  $O(p^2)$  bytes of working memory. During  $p$  successive iterations, uninterrupted by restarts or changes in the working set, the CONGRA algorithm computes a cycle of  $p$  conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size  $\alpha$  that satisfies the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size. Other line-search algorithms can be specified with the LIS= option.

### **Levenberg-Marquardt Optimization (LEVVAR)**

The LEVVAR algorithm performs a highly stable optimization; however, for large problems, it consumes more memory and takes longer than the other techniques. The Levenberg-Marquardt optimization technique is a slightly improved variant of the Moré (1978) implementation.

### **Nelder-Mead Simplex Optimization (NMSIMP)**

The Nelder-Mead simplex method does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for  $p \gg 40$ .

The original Nelder-Mead simplex algorithm is implemented and extended to boundary constraints. This algorithm does not compute the objective for infeasible points, but it changes the shape of the simplex by adapting to the nonlinearities of the objective function. This adaptation contributes to an increased speed of convergence. NMSIMP uses a special termination criterion.

---

## **Output Data Set**

Many procedures in SAS software add the variables from the input data set when an observationwise output data set is created. The assumption of high-performance analytical procedures is that the input data sets can be large and contain many variables. For performance reasons, the output data set contains the following:

- those variables explicitly created by the statement
- variables listed in the **ID** statement

This enables you to add output data set information that is necessary for subsequent SQL joins without copying the entire input data set to the output data set. For more information about output data sets that are



produced when PROC HPFMM is run in distributed mode, see the section “Output Data Sets” (Chapter 2, *SAS/STAT User’s Guide: High-Performance Procedures*).

---

## Default Output

The following sections describe the output that PROC HPFMM produces by default. The output is organized into various tables, which are discussed in the order of appearance for maximum likelihood and Bayes estimation, respectively.

### Performance Information

The “Performance Information” table is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

### Model Information

The “Model Information” table displays basic information about the model, such as the response variable, frequency variable, link function, and the model category that the HPFMM procedure determined based on your input and options. The “Model Information” table is one of a few tables that are produced irrespective of estimation technique. Most other tables are specific to Bayes or maximum likelihood estimation.

If the analysis depends on generated random numbers, the “Model Information” table also displays the random number seed used to initialize the random number generators. If you repeat the analysis and pass this seed value in the **SEED=** option in the **PROC HPFMM** statement, an identical stream of random numbers results.

### Class Level Information

The “Class Level Information” table lists the levels of every variable specified in the **CLASS** statement. You should check this information to make sure that the data are correct. You can adjust the order of the **CLASS** variable levels with the **ORDER=** option in the **CLASS** statement. You can suppress the “Class Level Information” table completely or partially with the **NOCLPRINT=** option in the **PROC HPFMM** statement.

### Number of Observations

The “Number of Observations” table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a **FREQ** statement, the table also displays the sum of frequencies read and used. If the *events/trials* syntax is used for the response, the table also displays the number of events and trials used in the analysis.

Note that the number of observations “used” in the analysis is not unambiguous in a mixture model. An observation that is “unusable” for one component distribution (because the response value is outside of the support of the distribution) might still be usable in the mixture model when the response value is in the support of another component distribution. You can affect the way in which PROC HPFMM handles exclusion of observations due to support violations with the **EXCLUSION=** option in the **PROC HPFMM** statement.

## Response Profile

For binary data, the “Response Profile” table displays the ordered value from which the HPFMM procedure determines the probability being modeled as an event for binary data. For each response category level, the frequency used in the analysis is reported.

## Default Output for Maximum Likelihood

### Optimization Information

The “Optimization Information” table displays basic information about the optimization setup to determine the maximum likelihood estimates, such as the optimization technique, the parameters that participate in the optimization, and the number of threads used for the calculations. This table is not produced during model selection—that is, if the **KMAX=** option is specified in the **MODEL** statement.

### Iteration History

The “Iteration History” table displays for each iteration of the optimization the number of function evaluations (including gradient and Hessian evaluations), the value of the objective function, the change in the objective function from the previous iteration, and the absolute value of the largest (projected) gradient element. The objective function used in the optimization in the HPFMM procedure is the negative of the mixture log likelihood; consequently, PROC HPFMM performs a minimization. This table is not produced if you specify the **KMAX=** option in the **MODEL** statement. If you wish to see the “Iteration History” table in this setting, you must also specify the **FITDETAILS** option in the **PROC HPFMM** statement.

### Convergence Status

The convergence status table is a small ODS table that follows the “Iteration History” table in the default output. In the listing, it appears as a message that identifies whether the optimization succeeded and which convergence criterion was met. If the optimization fails, the message indicates the reason for the failure. If you save the “Convergence Status” table to an output data set, a numeric Status variable is added that allows you to assess convergence programmatically. The values of the Status variable encode the following:

- |   |   |
|---|---|
| 0 | Convergence was achieved or an optimization was not performed (because of <b>TECHNIQUE=NONE</b> ).  |
| 1 | The objective function could not be improved.   |
| 2 | Convergence was not achieved because of a user interrupt or because a limit was exceeded, such as the maximum number of iterations or the maximum number of function evaluations. To modify these limits, see the <b>MAXITER=</b> , <b>MAXFUNC=</b> , and <b>MAXTIME=</b> options in the <b>PROC HPFMM</b> statement. |
| 3 | Optimization failed to converge because function or derivative evaluations failed at the starting values or during the iterations or because a feasible point that satisfies the parameter constraints could not be found in the parameter space.   |

### Fit Statistics

The “Fit Statistics” table displays a variety of fit measures based on the mixture log likelihood in addition to the Pearson statistic. All statistics are presented in “smaller is better” form. If you are fitting a single-component normal, gamma, or inverse Gaussian model, the table also contains the unscaled Pearson statistic. If you are fitting a mixture model or the model has been fitted under restrictions, the table also contains the number of effective components and the number of effective parameters.

The calculation of the information criteria uses the following formulas, where  $p$  denotes the number of effective parameters,  $n$  denotes the number of observations used (or the sum of the frequencies used if a **FREQ** statement is present), and  $l$  is the log likelihood of the mixture evaluated at the converged estimates:

$$\begin{aligned} \text{AIC} &= -2l + 2p \\ \text{AICC} &= \begin{cases} -2l + 2pn/(n - p - 1) & n > p + 2 \\ -2l + 2p(p + 2) & \text{otherwise} \end{cases} \\ \text{BIC} &= -2l + p \log(n) \end{aligned}$$

The Pearson statistic is computed simply as

$$\text{Pearson statistic} = \sum_{i=1}^n f_i \frac{(y_i - \hat{\mu}_i)^2}{\widehat{\text{Var}}[Y_i]}$$

where  $n$  denotes the number of observations used in the analysis,  $f_i$  is the frequency associated with the  $i$ th observation (or 1 if no frequency is specified),  $\mu_i$  is the mean of the mixture, and the denominator is the variance of the  $i$ th observation in the mixture. Note that the mean and variance in this expression are not those of the component distributions, but the mean and variance of the mixture:

$$\begin{aligned} \mu_i &= E[Y_i] = \sum_{j=1}^k \pi_{ij} \mu_{ij} \\ \text{Var}[Y_i] &= -\mu_i^2 + \sum_{j=1}^k \pi_{ij} (\sigma_{ij}^2 + \mu_{ij}^2) \end{aligned}$$

where  $\mu_{ij}$  and  $\sigma_{ij}^2$  are the mean and variance, respectively, for observation  $i$  in the  $j$ th component distribution and  $\pi_{ij}$  is the mixing probability for observation  $i$  in component  $j$ .

The unscaled Pearson statistic is computed with the same expression as the Pearson statistic with  $n$ ,  $f_i$ , and  $\mu_i$  as previously defined, but the scale parameter  $\phi$  is set to 1 in the  $\widehat{\text{Var}}[Y_i]$  expression.

The number of effective components and the number of effective parameters are determined by examining the converged solution for the parameters that are associated with model effects and the mixing probabilities. For example, if a component has an estimated mixing probability of zero, the values of its parameter estimates are immaterial. You might argue that all parameters should be counted towards the penalty in the information criteria. But a component with zero mixing probability in a  $k$ -component model effectively reduces the model to a  $(k - 1)$ -component model. A situation of an overfit model, for which a parameter penalty needs to be taken when calculating the information criteria, is a different situation; here the mixing probability might be small, possibly close to zero.

### Parameter Estimates

The parameter estimates, their estimated (asymptotic) standard errors, and  $p$ -values for the hypothesis that the parameter is zero are presented in the “Parameter Estimates” table. A separate table is produced for each **MODEL** statement, and the components that are associated with a **MODEL** statement are identified with an overall component count variable that counts across **MODEL** statements. If you assign a label to a model with the **LABEL=** option in the **MODEL** statement, the label appears in the title of the “Parameter Estimates” table. Otherwise, the internal label generated by the HPFMM procedure is used.

If the **MODEL** statement does not contain effects and the link function is not the identity, the inversely linked estimate is also displayed in the table. For many distributions, the inverse linked estimate is the estimated mean on the data scale. For example, in a binomial or binary model, it represents the estimated probability of an event. For some distributions (for example, the Weibull distribution), the inverse linked estimate is not the component distribution mean.

If you request confidence intervals with the **CL** or **ALPHA=** option in the **MODEL** statement, confidence limits are produced for the estimate on the linear scale. If the inverse linked estimate is displayed, confidence intervals for that estimate are also produced by inversely linking the confidence bounds on the linear scale.

### **Mixing Probabilities**

If you fit a model with more than one component, the table of mixing probabilities is produced. If there are no effects in the **PROBMODEL** statement or if there is no **PROBMODEL** statement, the parameters are reported on the linear scale and as mixing probabilities. If model effects are present, only the linear parameters (on the scale of the logit, generalized logit, probit, and so on) are displayed.

## **Default Output for Bayes Estimation**

### **Bayes Information**

This table provides basic information about the sampling algorithm. The HPFMM procedure uses either a conjugate sampler or a Metropolis-Hastings sampling algorithm based on Gamerman (1997). The table reveals, for example, how many model parameters are sampled, how many parameters associated with mixing probabilities are sampled, and how many threads are used to perform multithreaded analysis.

### **Prior Distributions**

The “Prior Distributions” table lists for each sampled parameter the prior distribution and its parameters. The mean and variance (if they exist) for those values of the parameters are also displayed, along with the initial value for the parameter in the Markov chain. The Component column in this table identifies the mixture component to which a particular parameter belongs. You can control how the HPFMM procedure determines initial values with the **INITIAL=** option in the **BAYES** statement.

### **Bayesian Fit Statistics**

The “Bayesian Fit Statistics” table shows three measures based on the posterior sample. The “Average -2 Log Likelihood” is derived from the average mixture log-likelihood for the data, where the average is taken over the posterior sample. The deviance information criterion (DIC) is a Bayesian measure of model fit and the effective number of parameters ( $p_D$ ) is a penalization term used in the computation of the DIC. See the section “**Summary Statistics**” on page 158 in Chapter 7, “**Introduction to Bayesian Analysis Procedures**,” for a detailed discussion of the DIC and  $p_D$ .

### **Posterior Summaries**

The arithmetic mean, standard deviation, and percentiles of the posterior distribution of the parameter estimates are displayed in the “Posterior Summaries” table. By default, the HPFMM procedure computes the 25th, 50th (median), and 75th percentiles of the sampling distribution. You can modify the percentiles through suboptions of the **STATISTICS** option in the **BAYES** statement. If a parameter corresponds to a singularity in the design and was removed from sampling for that purpose, it is also displayed in the table of posterior summaries (and in other tables that relate to output from the **BAYES** statement). The posterior sample size for such a parameter is shown as  $N = 0$ .

### Posterior Intervals

The table of “Posterior Intervals” displays equal-tail intervals and intervals of highest posterior density for each parameter. By default, intervals are computed for an  $\alpha$ -level of 0.05, which corresponds to 95% intervals. You can modify this confidence level by providing one or more  $\alpha$  values in the ALPHA= suboption of the STATISTICS option in the BAYES statement. The computation of these intervals is detailed in section “Summary Statistics” on page 158 in Chapter 7, “Introduction to Bayesian Analysis Procedures.”

### Posterior Autocorrelations

Autocorrelations for the posterior estimates are computed by default for autocorrelation lags 1, 5, 10, and 50, provided that a sufficient number of posterior samples is available. See the section “Assessing Markov Chain Convergence” on page 144 in Chapter 7, “Introduction to Bayesian Analysis Procedures,” for the computation of posterior autocorrelations and their utility in diagnosing convergence of Markov chains. You can modify the list of lags for which posterior autocorrelations are calculated with the AUTOCORR suboption of the DIAGNOSTICS= option in the BAYES statement.

## ODS Table Names

Each table created by PROC HPFMM has a name associated with it, and you must use this name to reference the table when you use ODS statements. These names are listed in Table 55.10.

**Table 55.10** ODS Tables Produced by PROC HPFMM

Table Name	Description	Required Statement / Option
Autocorr	Autocorrelation among posterior estimates	BAYES
BayesInfo	Basic information about Bayesian estimation	BAYES
ClassLevels	Level information from the CLASS statement	CLASS
CompDescription	Component description in models with varying number of components	KMAX= in MODEL with ML estimation
CompEvaluation	Comparison of mixture models with varying number of components	KMAX= in MODEL with ML estimation
CompInfo	Component information	COMPONENTINFO option in PROC HPFMM statement
ConvergenceStatus	Status of optimization at conclusion of optimization	Default output
Constraints	Linear equality and inequality constraints	RESTRICT statement or EQUATE=EFFECTS option in MODEL statement
Corr	Asymptotic correlation matrix of parameter estimates (ML) or empirical correlation matrix of the Bayesian posterior estimates	CORR option in PROC HPFMM statement

**Table 55.10** *continued*

Table Name	Description	Required Statement / Option
Cov	Asymptotic covariance matrix of parameter estimates (ML) or empirical covariance matrix of the Bayesian posterior estimates	COV option in PROC HPFMM statement
CovI	Inverse of the covariance matrix of the parameter estimates	COVI option in PROC HPFMM statement
ESS	Effective sample size	DIAG=ESS option in BAYES statement
FitStatistics	Fit statistics	Default output
Geweke	Geweke diagnostics (Geweke 1992) for Markov chain	DIAG=GEWEKE option in BAYES statement
Hessian	Hessian matrix from the maximum likelihood optimization, evaluated at the converged estimates	HESSIAN
IterHistory	Iteration history	Default output for ML estimation
MCSE	Monte Carlo standard errors	DIAG=MCERROR in BAYES statement
MixingProbs	Solutions for the parameter estimates associated with effects in PROB-MODEL statements	Default output for ML estimation if number of components is greater than 1
ModelInfo	Model information	Default output
NObs	Number of observations read and used, number of trials and events	Default output
OptInfo	Optimization information	Default output for ML estimation
ParameterEstimates	Solutions for the parameter estimates associated with effects in MODEL statements	Default output for ML estimation
ParameterMap	Mapping of parameter names to OUTPOST= data set	OUTPOST= option in BAYES statement
PriorInfo	Prior distributions and initial value of Markov chain	BAYES
PostSummaries	Summary statistics for posterior estimates	BAYES
PostIntervals	Equal-tail and highest posterior density intervals for posterior estimates	BAYES
ResponseProfile	Response categories and category modeled	Default output in models with binary response

## ODS Graphics

You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC HPFMM generates are listed in [Table 55.11](#), along with the required statements and options.

**Table 55.11** Graphs Produced by PROC HPFMM

ODS Graph Name	Plot Description	Option
<a href="#">TADPanel</a>	Panel of diagnostic graphics to assess convergence of Markov chains	<a href="#">BAYES</a>
<a href="#">DensityPlot</a>	Histogram and density with component distributions	Default plot for homogeneous mixtures
<a href="#">CriterionPanel</a>	Panel of plots showing progression of model fit criteria for mixtures with different numbers of components	<a href="#">KMIN=</a> and <a href="#">KMAX=</a> options in <a href="#">MODEL</a> statement

## Examples: HPFMM Procedure

### Example 55.1: Modeling Mixing Probabilities: All Mice Are Equal, but Some Mice Are More Equal Than Others

This example demonstrates how you can model the means and mixture proportions separately in a binomial cluster model. It also compares the binomial cluster model to the beta-binomial model.

In a typical teratological experiment, the offspring of animals that were exposed to a toxin during pregnancy are studied for malformation. If you count the number of malformed offspring in a litter of size  $n$ , then this count is typically not binomially distributed. The responses of the offspring from the same litter are not independent; hence their sum does not constitute a binomial random variable. Relative to a binomial model, data from teratological experiments exhibit overdispersion because ignoring positive correlation among the responses tends to overstate the precision of the parameter estimates. Overdispersion mechanisms are briefly discussed in the section “[Overdispersion](#)” on page 4495.



In this application, the focus is on mixtures and models that involve a mixing mechanism. The mixing approach (Williams 1975; Haseman and Kupper 1979) supposes that the binomial success probability is a random variable that follows a  $\text{beta}(\alpha, \beta)$  distribution:

$$\begin{aligned} Y|\mu &\sim \text{Binomial}(n, \mu) \\ \mu &\sim \text{Beta}(\alpha, \beta) \\ Y &\sim \text{Beta-binomial}(n, \mu, \phi) \\ E[Y] &= n\pi \\ \text{Var}[Y] &= n\pi(1 - \pi) \{1 + \mu^2(n - 1)\} \end{aligned}$$

If  $\mu = 0$ , then the beta-binomial distribution reduces to a standard binomial model with success probability  $\pi$ . The parameterization of the beta-binomial distribution used by the HPFMM procedure is based on Neerchal and Morel (1998), see the section “[Log-Likelihood Functions for Response Distributions](#)” on page 4495 for details.

Morel and Nagaraj (1993); Morel and Neerchal (1997); Neerchal and Morel (1998) propose a different model to capture dependency within binomial clusters. Their model is a two-component mixture that gives rise to the same mean and variance function as the beta-binomial model. The genesis is different, however. In the binomial cluster model of Morel and Neerchal, suppose there is a cluster of  $n$  Bernoulli outcomes with success probability  $\pi$ . The number of responses in the cluster decomposes into  $N \leq n$  outcomes that all respond with either “success” or “failure”; the important aspect is that they all respond identically. The remaining  $n - N$  Bernoulli outcomes respond independently, so the sum of successes in this group is a  $\text{binomial}(n - N, \pi)$  random variable. Denote the probability with which cluster members fall into the group of identical respondents as  $\mu$ . Then  $1 - \mu$  is the probability that a response belongs to the group of independent Bernoulli outcomes.

It is easy to see how this process of dividing the individual Bernoulli outcomes creates clustering. The binomial cluster model can be written as the two-component mixture

$$\Pr(Y = y) = \pi \Pr(U = y) + (1 - \pi) \Pr(V = y)$$

where  $U \sim \text{Binomial}(n, \mu^* + \mu)$ ,  $V \sim \text{Binomial}(n, \mu^*)$ , and  $\mu^* = (1 - \mu)\pi$ . This mixture model is somewhat unusual because the mixing probability  $\pi$  appears as a parameter in the component distributions. The two probabilities involved,  $\pi$  and  $\mu$ , have the following interpretation:  $\pi$  is the unconditional probability of success for any observation, and  $\mu$  is the probability with which the Bernoulli observations respond identically. The complement of this probability,  $1 - \mu$ , is the probability with which the Bernoulli outcomes respond independently. If  $\mu = 0$ , then the two-component mixture reduces to a standard Binomial model with success probability  $\pi$ . Since both  $\pi$  and  $\mu$  are involved in the success probabilities of the two Binomial variables in the mixture, you can affect these binomial means by specifying effects in the [PROBMODEL](#) statement (for the  $\pi$ s) or the [MODEL](#) statement (for the  $\mu$ s). In a “straight” two-component Binomial mixture,

$$\pi \text{Binomial}(n, \mu_1) + (1 - \pi) \text{Binomial}(n, \mu_2)$$

you would vary the success probabilities  $\mu_1$  and  $\mu_2$  through the [MODEL](#) statement.

With the HPFMM procedure, you can fit the beta-binomial model by specifying [DIST=BETABIN](#) and the binomial cluster model by specifying [DIST=BINOMCLUS](#) in the [MODEL](#) statement.

Morel and Neerchal (1997) report data from a completely randomized design that studies the teratogenicity of phenytoin in 81 pregnant mice. The treatment structure of the experiment is an augmented factorial. In



addition to an untreated control, mice received 60 mg/kg of phenytoin (PHT), 100 mg/kg of trichloropropene oxide (TCPO), and their combination. The design was augmented with a control group that was treated with water. As in Morel and Neerchal (1997), the two control groups are combined here into a single group.

The following DATA step creates the data for this analysis as displayed in Table 1 of Morel and Neerchal (1997). The second DATA step creates continuous variables x1–x3 to match the parameterization of these authors.

```
data ossi;
  length tx $8;
  input tx$ n @@;
  do i=1 to n;
    input y m @@;
    output;
  end;
  drop i;
  datalines;
Control 18 8 8 9 9 7 9 0 5 3 3 5 8 9 10 5 8 5 8 1 6 0 5
      8 8 9 10 5 5 4 7 9 10 6 6 3 5
Control 17 8 9 7 10 10 10 1 6 6 6 1 9 8 9 6 7 5 5 7 9
      2 5 5 6 2 8 1 8 0 2 7 8 5 7
PHT      19 1 9 4 9 3 7 4 7 0 7 0 4 1 8 1 7 2 7 2 8 1 7
      0 2 3 10 3 7 2 7 0 8 0 8 1 10 1 1
TCPO     16 0 5 7 10 4 4 8 11 6 10 6 9 3 4 2 8 0 6 0 9
      3 6 2 9 7 9 1 10 8 8 6 9
PHT+TCPO 11 2 2 0 7 1 8 7 8 0 10 0 4 0 6 0 7 6 6 1 6 1 7
;

data ossi;
  set ossi;
  array xx{3} x1-x3;
  do i=1 to 3; xx{i}=0; end;
  pht = 0;
  tcpo = 0;
  if (tx='TCPO') then do;
    xx{1} = 1;
    tcpo = 100;
  end; else if (tx='PHT') then do;
    xx{2} = 1;
    pht = 60;
  end; else if (tx='PHT+TCPO') then do;
    pht = 60;
    tcpo = 100;
    xx{1} = 1; xx{2} = 1; xx{3}=1;
  end;
run;
```

The HPFMM procedure models the mean parameters  $\mu$  through the **MODEL** statement and the mixing proportions  $\pi$  through the **PROBMODEL** statement. In the binomial cluster model, you can place a regression structure on either set of probabilities, and the regression structure does not need to be the same. In the following statements, the unconditional probability of ossification is modeled as a two-way factorial, whereas the intralitter effect—the propensity to group within a cluster—is assumed to be constant:

```
proc hpfmm data=ossi;
  class pht tcpo;
  model y/m = / dist=binomcluster;
  probmodel pht tcpo pht*tcpo;
run;
```

The **CLASS** statement declares the PHT and TCPO variables as classification variables. They affect the analysis through their levels, not through their numeric values. The **MODEL** statement declares the distribution of the data to follow a binomial cluster model. The HPFMM procedure then automatically assumes that the model is a two-component mixture. An intercept is included by default. The **PROBMODEL** statement declares the effect structure for the mixing probabilities. The unconditional probability of ossification of a fetus depends on the main effects and the interaction in the factorial.

The “Model Information” table displays important details about the model fit with the HPFMM procedure (Output 55.1.1). Although no **K=** option was specified in the **MODEL** statement, the HPFMM procedure recognizes the model as a two-component model. The “Class Level Information” table displays the levels and values of the PHT and TCPO variables. Eighty-one observations are read from the data and are used in the analysis. These observations comprise 287 events and 585 total outcomes.

**Output 55.1.1** Model Information in Binomial Cluster Model with Constant Clustering Probability

The HPFMM Procedure	
Model Information	
Data Set	WORK.OSSI
Response Variable (Events) y	
Response Variable (Trials) m	
Type of Model	Binomial Cluster
Distribution	Binomial Cluster
Components	2
Link Function	Logit
Estimation Method	Maximum Likelihood
Class Level Information	
Class	Levels Values
pht	2 0 60
tcpo	2 0 100
Number of Observations Read	81
Number of Observations Used	81
Number of Events	287
Number of Trials	585

The “Optimization Information” table in Output 55.1.2 gives details about the maximum likelihood optimization. By default, the HPFMM procedure uses a quasi-Newton algorithm. The model contains five parameters, four of which are part of the model for the mixing probabilities. The fifth parameter is the intercept in the model for  $\mu$ .

**Output 55.1.2** Optimization in Binomial Cluster Model with Constant Clustering Probability

Optimization Information				
Optimization Technique	Dual Quasi-Newton			
Parameters in Optimization	5			
Mean Function Parameters	1			
Scale Parameters	0			
Mixing Prob Parameters	4			

Iteration History				
Iteration	Evaluations	Objective Function	Change	Max Gradient
0	5	174.92723892		43.78769
1	2	154.13180744	20.79543149	11.2346
2	3	153.26693611	0.86487133	6.888215
3	2	152.84974281	0.41719329	3.541977
4	3	152.61756033	0.23218248	2.783556
5	3	152.54795303	0.06960730	1.146807
6	3	152.52684929	0.02110374	0.034367
7	3	152.52671214	0.00013715	0.011511
8	3	152.52670799	0.00000415	0.000202
9	3	152.52670799	0.00000000	4.001E-6

Convergence criterion (GCONV=1E-8) satisfied.				
---	--	--	--	--

Fit Statistics	
-2 Log Likelihood	305.1
AIC (Smaller is Better)	315.1
AICC (Smaller is Better)	315.9
BIC (Smaller is Better)	327.0
Pearson Statistic	89.2077
Effective Parameters	5
Effective Components	2

After nine iterations, the iterative optimization converges. The  $-2$  log likelihood at the converged solution is 305.1, and the Pearson statistic is 89.2077. The HPFMM procedure computes the Pearson statistic as a general goodness-of-fit measure that expresses the closeness of the fitted model to the data.

The estimates of the parameters in the conditional probability  $\mu$  and in the unconditional probability  $\pi$  are given in [Output 55.1.3](#). The intercept estimate in the model for  $\mu$  is 0.3356. Since the default link in the binomial cluster model is the logit link, the estimate of the conditional probability is

$$\hat{\mu} = \frac{1}{1 + \exp\{-0.3356\}} = 0.5831$$

This value is displayed in the “Inverse Linked Estimate” column. There is greater than a 50% chance that the individual fetuses in a litter provide the same response. The clustering tendency is substantial.

**Output 55.1.3** Parameter Estimates in Binomial Cluster Model with Constant Clustering Probability

Parameter Estimates for Binomial Cluster Model						
Component	Effect	Estimate	Standard Error	z Value	Pr >  z	Inverse Linked Estimate
1	Intercept	0.3356	0.1714	1.96	0.0503	0.5831

Parameter Estimates for Mixing Probabilities						
Component	Effect	pht	tcpo	Estimate	Standard Error	z Value Pr >  z
1	Intercept			-1.2194	0.4690	-2.60 0.0093
1	pht	0		0.9129	0.5608	1.63 0.1036
1	pht	60		0	.	. .
1	tcpo		0	0.3295	0.5534	0.60 0.5516
1	tcpo		100	0	.	. .
1	pht*tcpo	0	0	0.6162	0.6678	0.92 0.3561
1	pht*tcpo	0	100	0	.	. .
1	pht*tcpo	60	0	0	.	. .
1	pht*tcpo	60	100	0	.	. .

The “Mixing Probabilities” table displays the estimates of the parameters in the model for  $\pi$  on the logit scale (Output 55.1.3). Table 55.12 constructs the estimates of the unconditional probabilities of ossification.

**Table 55.12** Estimates of Ossification Probabilities

PHT	TCPO	$\hat{\eta}$	$\hat{\pi}$
0	0	$-1.2194 + 0.9129 + 0.3295 + 0.6162 = 0.6392$	0.6546
60	0	$-1.2194 + 0.3295 = -0.8899$	0.2911
0	100	$-1.2194 + 0.9129 = -0.3065$	0.4240
60	100	$-1.2194$	0.2280

Morel and Neerchal (1997) considered a model in which the intralitter effects also depend on the treatments. This model is fit with the HPFMM procedure with the following statements:

```
proc hpfmm data=ossi;
  class pht tcpo;
  model y/m = pht tcpo pht*tcpo / dist=binomcluster;
  probmodel pht tcpo pht*tcpo;
run;
```

The  $-2$  log likelihood of this model is much reduced compared to the previous model with constant conditional probability (compare 287.8 in Output 55.1.4 with 305.1 in Output 55.1.2). The likelihood-ratio statistic of 17.3 is significant,  $\Pr(\chi^2_3 > 17.3 = 0.0006)$ . Varying the conditional probabilities by treatment improved the model fit significantly.

**Output 55.1.4** Fit Statistics and Parameter Estimates in Binomial Cluster Model

**The HPFMM Procedure**

Fit Statistics	
-2 Log Likelihood	287.8
AIC (Smaller is Better)	303.8
AICC (Smaller is Better)	305.8
BIC (Smaller is Better)	323.0
Pearson Statistic	85.5998
Effective Parameters	8
Effective Components	2

Parameter Estimates for Binomial Cluster Model						
Component	Effect	pht	tcpo	Estimate	Standard Error	z Value Pr >  z
1	Intercept			1.8213	0.5889	3.09 0.0020
1	pht	0		-1.4962	0.6630	-2.26 0.0240
1	pht	60		0	.	. .
1	tcpo		0	-3.1828	1.1261	-2.83 0.0047
1	tcpo		100	0	.	. .
1	pht*tcpo	0	0	3.3736	1.1953	2.82 0.0048
1	pht*tcpo	0	100	0	.	. .
1	pht*tcpo	60	0	0	.	. .
1	pht*tcpo	60	100	0	.	. .

Parameter Estimates for Mixing Probabilities						
Component	Effect	pht	tcpo	Estimate	Standard Error	z Value Pr >  z
1	Intercept			-0.7394	0.5395	-1.37 0.1705
1	pht	0		0.4351	0.6203	0.70 0.4830
1	pht	60		0	.	. .
1	tcpo		0	-0.5342	0.5893	-0.91 0.3646
1	tcpo		100	0	.	. .
1	pht*tcpo	0	0	1.4055	0.7080	1.99 0.0471
1	pht*tcpo	0	100	0	.	. .
1	pht*tcpo	60	0	0	.	. .
1	pht*tcpo	60	100	0	.	. .

Table 55.13 computes the conditional probabilities in the four treatment groups. Recall that the previous model estimated a constant clustering probability of 0.5831.

**Table 55.13** Estimates of Clustering Probabilities

PHT	TCPO	$\hat{\eta}$	$\hat{\mu}$
0	0	$1.8213 - 1.4962 - 3.1828 + 3.3736 = 0.5159$	0.6262
60	0	$1.8213 - 3.1828 = -1.3615$	0.2040
0	100	$1.8213 - 1.4962 = 0.3251$	0.5806
60	100	1.8213	0.8607

The presence of phenytoin alone reduces the probability of response clustering within the litter. The presence of trichloropropene oxide alone does not have a strong effect on the clustering. The simultaneous presence of both agents substantially increases the probability of clustering.

The following statements fit the binomial cluster model in the parameterization of Morel and Neerchal (1997).

```
proc hpfmm data=ossi;
  model y/m = x1-x3 / dist=binomcluster;
  probmodel x1-x3;
run;
```

The model fit is the same as in the previous model (compare the “Fit Statistics” tables in [Output 55.1.5](#) and [Output 55.1.4](#)). The parameter estimates change due to the reparameterization of the treatment effects and match the results in Table III of Morel and Neerchal (1997).

#### Output 55.1.5 Fit Statistics and Estimates (Morel and Neerchal Parameterization)

##### The HPFMM Procedure

Fit Statistics	
-2 Log Likelihood	287.8
AIC (Smaller is Better)	303.8
AICC (Smaller is Better)	305.8
BIC (Smaller is Better)	323.0
Pearson Statistic	85.5999
Effective Parameters	8
Effective Components	2

Parameter Estimates for Binomial Cluster Model					
		Standard			
Component	Effect	Estimate	Error	z Value	Pr >  z
1	Intercept	0.5159	0.2603	1.98	0.0475
1	x1	-0.1908	0.4006	-0.48	0.6339
1	x2	-1.8774	0.9946	-1.89	0.0591
1	x3	3.3736	1.1953	2.82	0.0048

Parameter Estimates for Mixing Probabilities					
		Standard			
Component	Effect	Estimate	Error	z Value	Pr >  z
1	Intercept	0.5669	0.2455	2.31	0.0209
1	x1	-0.8712	0.3924	-2.22	0.0264
1	x2	-1.8405	0.3413	-5.39	<.0001
1	x3	1.4055	0.7080	1.99	0.0471

The following sets of statements fit the binomial and beta-binomial models, respectively, as single-component mixtures in the parameterization akin to the first binomial cluster model. Note that the model effects that affect the underlying Bernoulli success probabilities are specified in the **MODEL** statement, in contrast to the binomial cluster model.

```
proc hpfmm data=ossi;
  model y/m = x1-x3 / dist=binomial;
run;
```

```
proc hpfmm data=ossi;
  model y/m = x1-x3 / dist=betabinomial;
run;
```

The Pearson statistic for the beta-binomial model (Output 55.1.6) indicates a much better fit compared to the single-component binomial model (Output 55.1.7). This is not surprising since these data are obviously overdispersed relative to a binomial model because the Bernoulli outcomes are not independent. The difference between the binomial cluster and the beta-binomial model lies in the mechanism by which the correlations are induced:

- a mixing mechanism in the beta-binomial model that leads to a common shared random effect among all offspring in a cluster
- a mixture specification in the binomial cluster model that divides the offspring in a litter into identical and independent responders

#### Output 55.1.6 Fit Statistics in Binomial Model

##### The HPFMM Procedure

Fit Statistics	
-2 Log Likelihood	401.8
AIC (Smaller is Better)	409.8
AICC (Smaller is Better)	410.3
BIC (Smaller is Better)	419.4
Pearson Statistic	252.1

#### Output 55.1.7 Fit Statistics in Beta-Binomial Model

##### The HPFMM Procedure

Fit Statistics	
-2 Log Likelihood	306.6
AIC (Smaller is Better)	316.6
AICC (Smaller is Better)	317.4
BIC (Smaller is Better)	328.5
Pearson Statistic	87.5379

## Example 55.2: The Usefulness of Custom Starting Values: When Do Cows Eat?

This example with a mixture of normal and Weibull distributions illustrates the benefits of specifying starting values for some of the components.

The data for this example were generously provided by Dr. Luciano A. Gonzalez of the Lethbridge Research Center of Agriculture and Agri-Food Canada and his collaborator, Dr. Bert Tolcamp, from the Scottish Agricultural College.

The outcome variable of interest is the logarithm of a time interval between consecutive visits by cattle to feeders. The intervals fall into three categories:

- short breaks within meals—such as when an animal stops eating for a moment and resumes shortly thereafter
- somewhat longer breaks when eating is interrupted to go have a drink of water
- long breaks between meals

Modeling such time interval data is important to understand the feeding behavior and biology of the animals and to derive other biological parameters such as the probability of an animal to stop eating after it has consumed a certain amount of a given food. Because there are three distinct biological categories, data of this nature are frequently modeled as three-component mixtures. The point at which the second and third components cross over is used to separate feeding events into meals.

The original data set comprises 141,414 observations of log feeding intervals. For the purpose of presentation in this document, where space is limited, the data have been rounded to precision 0.05 and grouped by frequency. The following DATA step displays the modified data used in this example. A comparison with the raw data and the results obtained in a full analysis of the original data show that the grouping does not alter the presentation or conclusions in a way that matters for the purpose of this example.

```
data cattle;
  input LogInt Count @@;
  datalines;
0.70 195 1.10 233 1.40 355 1.60 563
1.80 822 1.95 926 2.10 1018 2.20 1712
2.30 3190 2.40 2212 2.50 1692 2.55 1558
2.65 1622 2.70 1637 2.75 1568 2.85 1599
2.90 1575 2.95 1526 3.00 1537 3.05 1561
3.10 1555 3.15 1427 3.20 2852 3.25 1396
3.30 1343 3.35 2473 3.40 1310 3.45 2453
3.50 1168 3.55 2300 3.60 2174 3.65 2050
3.70 1926 3.75 1849 3.80 1687 3.85 2416
3.90 1449 3.95 2095 4.00 1278 4.05 1864
4.10 1672 4.15 2104 4.20 1443 4.25 1341
4.30 1685 4.35 1445 4.40 1369 4.45 1284
4.50 1523 4.55 1367 4.60 1027 4.65 1491
4.70 1057 4.75 1155 4.80 1095 4.85 1019
4.90 1158 4.95 1088 5.00 1075 5.05 912
5.10 1073 5.15 803 5.20 924 5.25 916
5.30 784 5.35 751 5.40 766 5.45 833
5.50 748 5.55 725 5.60 674 5.65 690
5.70 659 5.75 695 5.80 529 5.85 639
5.90 580 5.95 557 6.00 524 6.05 473
6.10 538 6.15 444 6.20 456 6.25 453
6.30 374 6.35 406 6.40 409 6.45 371
6.50 320 6.55 334 6.60 353 6.65 305
6.70 302 6.75 301 6.80 263 6.85 218
6.90 255 6.95 240 7.00 219 7.05 202
7.10 192 7.15 180 7.20 162 7.25 126
7.30 148 7.35 173 7.40 142 7.45 163
```



```

7.50 152 7.55 149 7.60 139 7.65 161
7.70 174 7.75 179 7.80 188 7.85 239
7.90 225 7.95 213 8.00 235 8.05 256
8.10 272 8.15 290 8.20 320 8.25 355
8.30 307 8.35 311 8.40 317 8.45 335
8.50 369 8.55 365 8.60 365 8.65 396
8.70 419 8.75 467 8.80 468 8.85 515
8.90 558 8.95 623 9.00 712 9.05 716
9.10 829 9.15 803 9.20 834 9.25 856
9.30 838 9.35 842 9.40 826 9.45 834
9.50 798 9.55 801 9.60 780 9.65 849
9.70 779 9.75 737 9.80 683 9.85 686
9.90 626 9.95 582 10.00 522 10.05 450
10.10 443 10.15 375 10.20 342 10.25 285
10.30 254 10.35 231 10.40 195 10.45 186
10.50 143 10.55 100 10.60 73 10.65 49
10.70 28 10.75 36 10.80 16 10.85 9
10.90 5 10.95 6 11.00 4 11.05 1
11.15 1 11.25 4 11.30 2 11.35 5
11.40 4 11.45 3 11.50 1
;

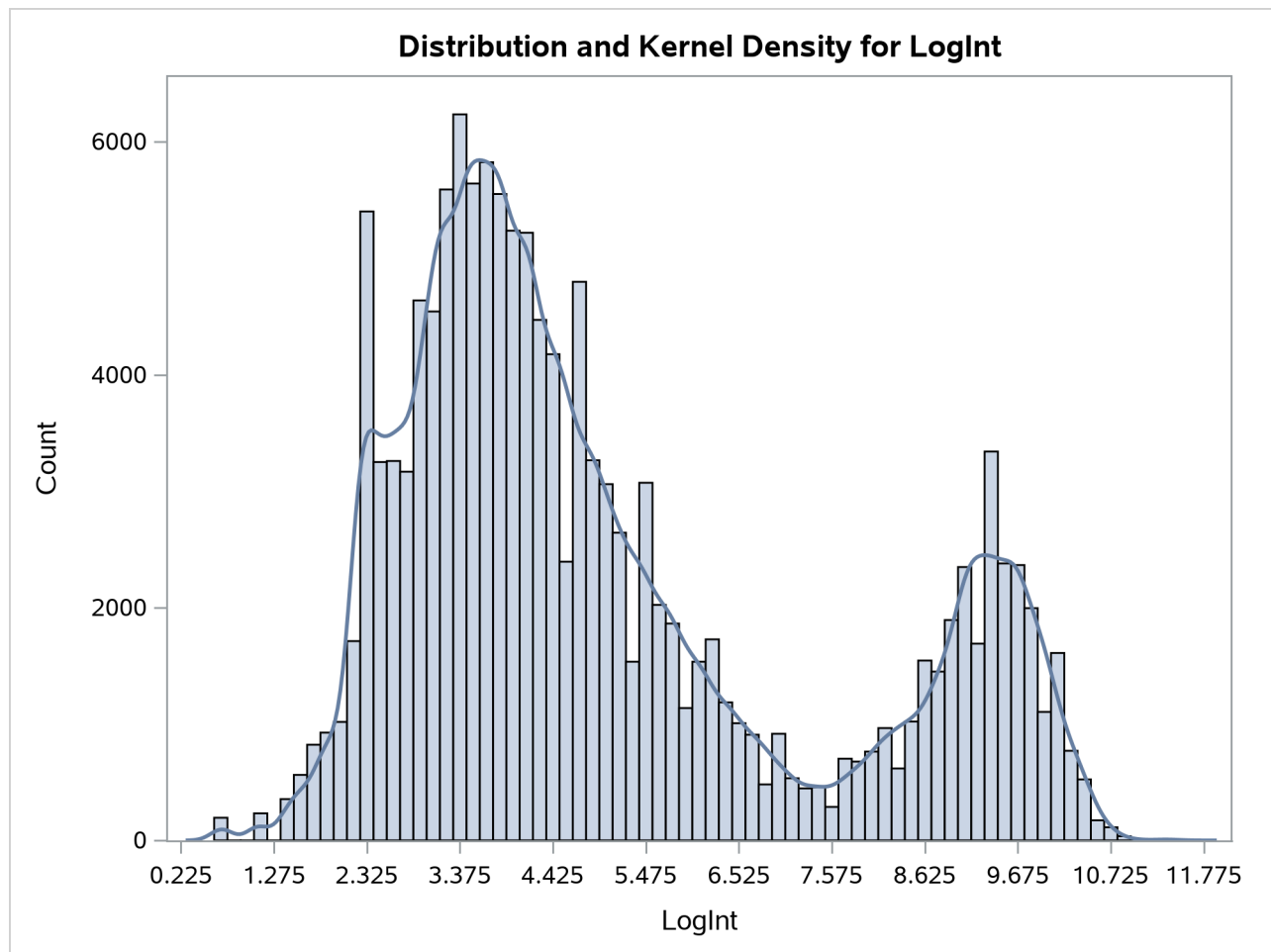
```

If you scan the columns for the Count variable in the DATA step, the prevalence of values between 2 and 5 units of LogInt is apparent, as is a long right tail. To explore these data graphically, the following statements produce a histogram of the data and a kernel density estimate of the density of the LogInt variable.

```

ods graphics on;
proc kde data=cattle;
  univar LogInt / bwm=4;
  freq count;
run;

```

**Output 55.2.1** Histogram and Kernel Density for LogInt

Two modes are clearly visible in [Output 55.2.1](#). Given the biological background, one would expect that three components contribute to the mixture. The histogram would suggest either a two-component mixture with modes near 4 and 9, or a three-component mixture with modes near 3, 5, and 9.

Following Dr. Gonzalez' suggestion, the process is modeled as a three-component mixture of two normal distributions and a Weibull distribution. The Weibull distribution is chosen because it can have long left and right tails and it is popular in modeling data that relate to time intervals.

```
proc hpfmm data=cattle gconv=0;
  model LogInt = / dist=normal k=2 parms(3 1, 5 1);
  model      + / dist=weibull;
  freq count;
run;
```

The `GCONV=` convergence criterion is turned off in this PROC HPFMM run to avoid the early stoppage of the iterations when the relative gradient changes little between iterations. Turning the criterion off usually ensures that convergence is achieved with a small absolute gradient of the objective function. The `PARMS` option in the first `MODEL` statement provides starting values for the means and variances for the parameters of the normal distributions. The means for the two components are started at  $\mu = 3$  and  $\mu = 5$ , respectively. Specifying starting values is generally not necessary. However, the choice of starting values can play an

important role in modeling finite mixture models; the importance of the choice of starting values in this example is discussed further below.

The “Model Information” table shows that the model is a three-component mixture and that the HPFMM procedure considers the estimation of a density to be the purpose of modeling. The procedure draws this conclusion from the absence of effects in the **MODEL** statements. There are 187 observations in the data set, but these actually represent 141,414 measurements ([Output 55.2.2](#)).

#### Output 55.2.2 Model Information and Number of Observations

##### The HPFMM Procedure

Model Information	
Data Set	WORK.CATTLE
Response Variable	LogInt
Frequency Variable	Count
Type of Model	Density Estimation
Components	3
Estimation Method	Maximum Likelihood
<hr/>	
Number of Observations Read	187
Number of Observations Used	187
Sum of Frequencies Read	141414
Sum of Frequencies Used	141414

There are eight parameters in the optimization: the means and variances of the two normal distributions, the  $\mu$  and  $\phi$  parameter of the Weibull distribution, and the two mixing probabilities ([Output 55.2.3](#)). At the converged solution, the  $-2$  log likelihood is 563,153 and all parameters and components are effective—that is, the model is not overspecified in the sense that components have collapsed during the model fitting. The Pearson statistic is close to the number of observations in the data set, indicating a good fit.

#### Output 55.2.3 Optimization Information and Fit Statistics

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	8
Mean Function Parameters	3
Scale Parameters	3
Mixing Prob Parameters	2
Lower Boundaries	3
Upper Boundaries	0
<hr/>	
Fit Statistics	
-2 Log Likelihood	563153
AIC (Smaller is Better)	563169
AICC (Smaller is Better)	563169
BIC (Smaller is Better)	563248
Pearson Statistic	141458
Effective Parameters	8
Effective Components	3

Output 55.2.4 displays the parameter estimates for the three models and for the mixing probabilities. The order in which the “Parameter Estimates” tables appear in the output corresponds to the order in which the **MODEL** statements were specified.

**Output 55.2.4** Optimization Information and Fit Statistics

Parameter Estimates for Normal Model						
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z	
1	Intercept	3.3415	0.01260	265.16	<.0001	
2	Intercept	4.8940	0.05447	89.84	<.0001	
1	Variance	0.6718	0.01287			
2	Variance	1.4497	0.05247			

Parameter Estimates for Weibull Model						
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z	Inverse Linked Estimate
3	Intercept	2.2531	0.000506	4452.11	<.0001	9.5174
3	Scale	0.06848	0.000427			

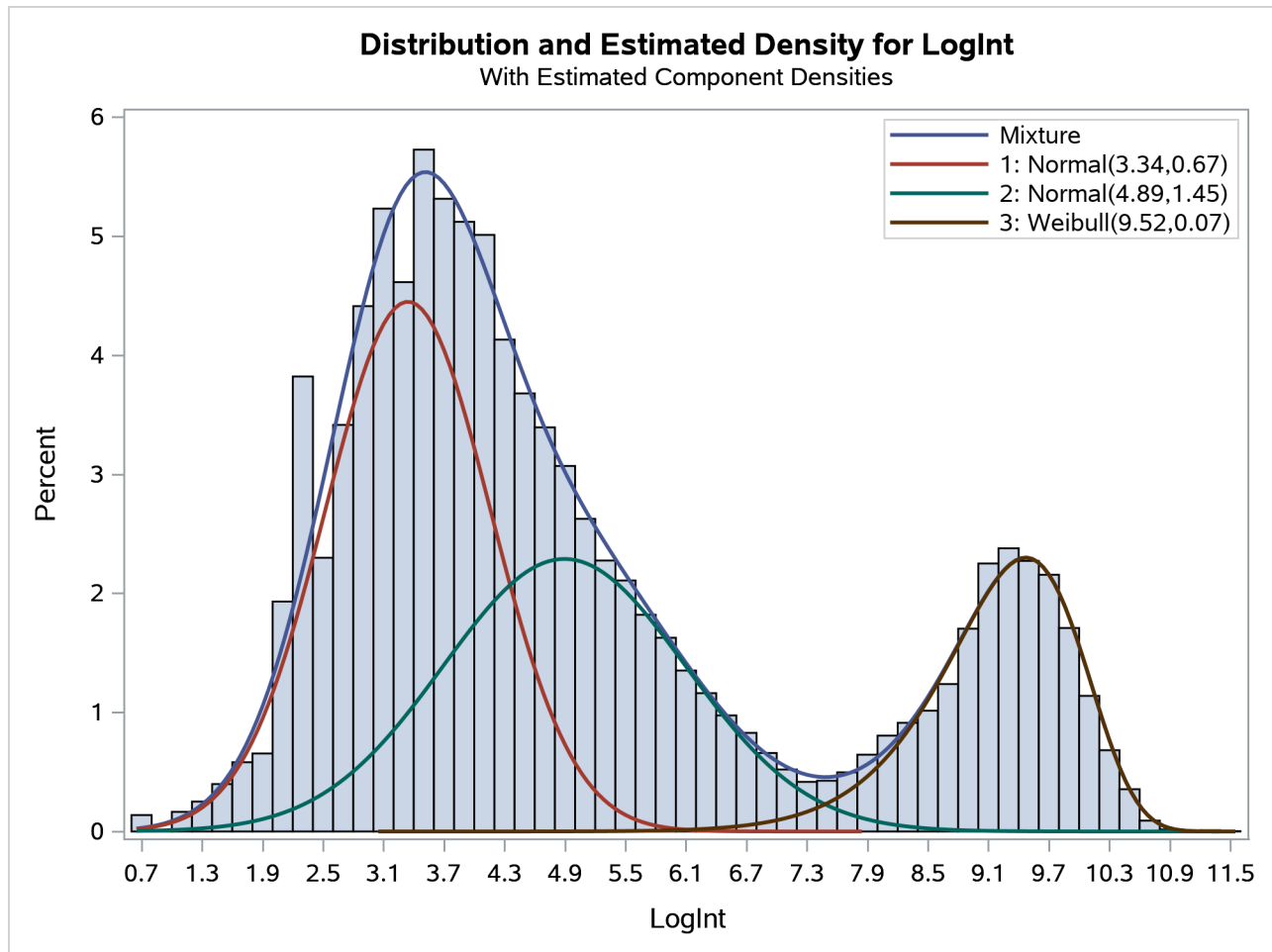
Parameter Estimates for Mixing Probabilities						
Component	Mixing Probability	GLogit(Prob)	Standard Error	z Value	Pr >  z	
1	0.4545	0.8106	0.03409	23.78	<.0001	
2	0.3435	0.5305	0.04640	11.43	<.0001	
3	0.2021	0				

The estimated means of the two normal components are 3.3415 and 4.8940, respectively. Note that the means are displayed here as Intercept. The inverse linked estimate is not produced because the default link for the normal distribution is the identity link; hence the Estimate column represents the means of the component distributions. The parameter estimates in the Weibull model are  $\hat{\beta}_0 = 2.2531$ ,  $\hat{\phi} = 0.06848$ , and  $\hat{\mu} = \exp\{\hat{\beta}_0\} = 9.5174$ . In the Weibull distribution, the  $\mu$  parameter does not estimate the mean of the distribution, the maximum likelihood estimate of the distribution's mean is  $\hat{\mu}\Gamma(\hat{\phi} + 1) = 9.1828$ .

The estimated mixing probabilities are  $\hat{\pi}_1 = 0.4545$ ,  $\hat{\pi}_2 = 0.3435$ , and  $\hat{\pi}_3 = 0.2021$ . In other words, the estimated distribution of log feeding intervals is a 45:35:20 mixture of an  $N(3.3415, 0.6718)$ , a  $N(4.8940, 1.4497)$ , and a  $Weibull(9.5174, 0.06848)$  distribution.

You can obtain a graphical display of the observed and estimated distribution of these data by enabling ODS Graphics. The **PLOTS** option in the **PROC HPFMM** statement modifies the default density plot by adding the densities of the mixture components:

```
ods select DensityPlot;
proc hpfmm data=cattle gconv=0;
  model LogInt = / dist=normal k=2 parms(3 1, 5 1);
  model      + / dist=weibull;
  freq count;
run;
```

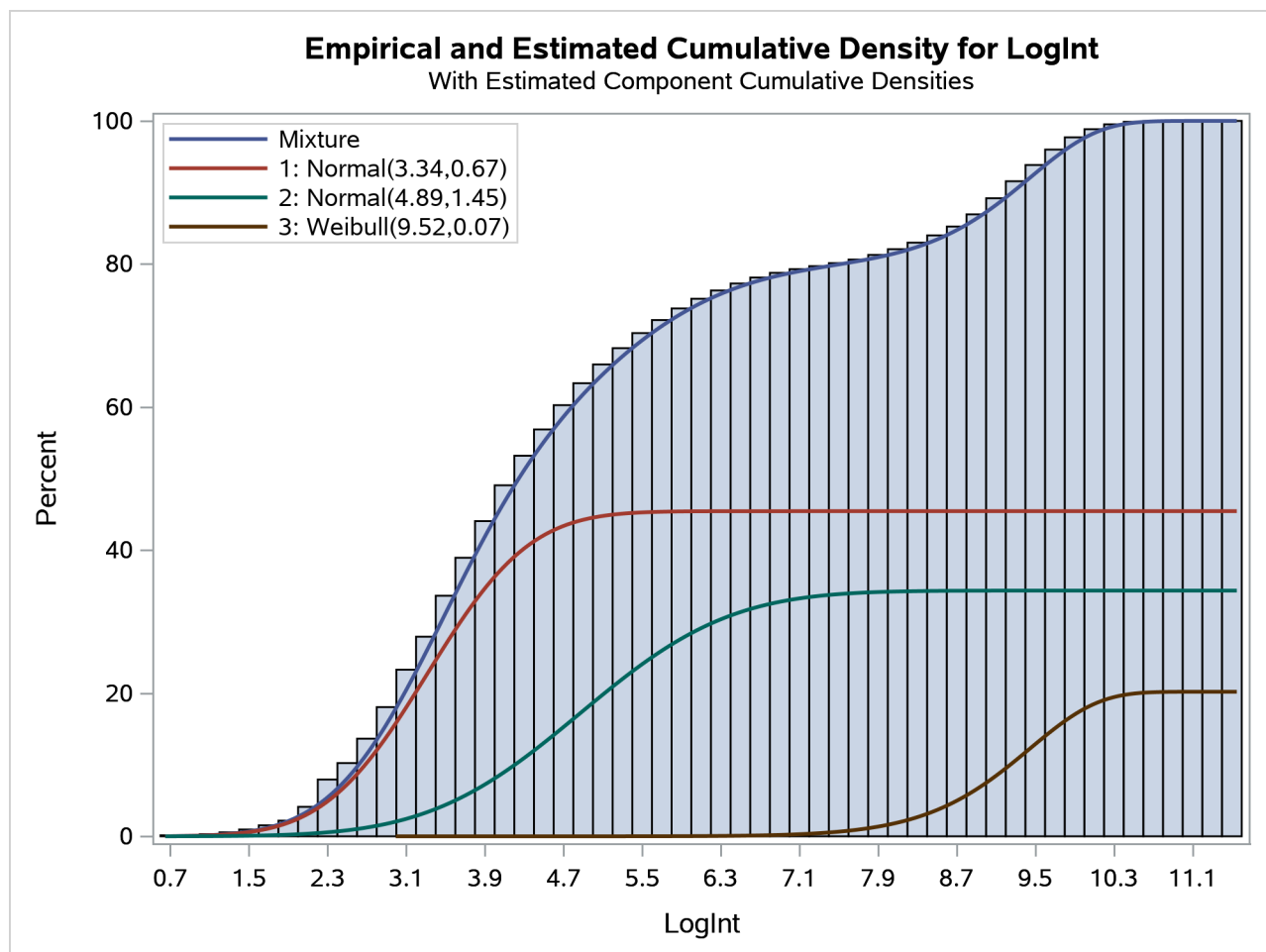
**Output 55.2.5** Observed and Estimated Densities in the Three-Component Model

The estimated mixture density matches the histogram of the observed data closely ([Output 55.2.5](#)). The component densities are displayed in such a way that, at each point in the support of the LogInt variable, their sum combines to the overall mixture density. The three components in the mixtures are well separated.

The excellent quality of the fit is even more evident when the distributions are displayed cumulatively by adding the CUMULATIVE option in the [DENSITY](#) option ([Output 55.2.6](#)):

```
ods select DensityPlot;
proc hpfmm data=cattle plot=density(cumulative) gconv=0;
  model LogInt = / dist=normal k=2 parms(3 1, 5 1);
  model      + / dist=weibull;
  freq count;
run;
```

The component cumulative distribution functions are again scaled so that their sum produces the overall mixture cumulative distribution function. Because of this scaling, the percentage reached at the maximum value of LogInt corresponds to the mixing probabilities in [Output 55.2.4](#).

**Output 55.2.6** Observed and Estimated Cumulative Densities in the Three-Component Model

The importance of starting values for the parameter estimates was mentioned previously. Suppose that different starting values are selected for the three components (for example, the default starting values).

```
proc hpfmm data=cattle gconv=0;
  model LogInt = / dist=normal k=2;
  model      + / dist=weibull;
  freq count;
run;
ods graphics off;
```

The fit statistics and parameter estimates from this run are displayed in [Output 55.2.7](#), and the density plot is shown in [Output 55.2.8](#).

**Output 55.2.7** Fit Statistics and Parameter Estimates**The HPFMM Procedure**

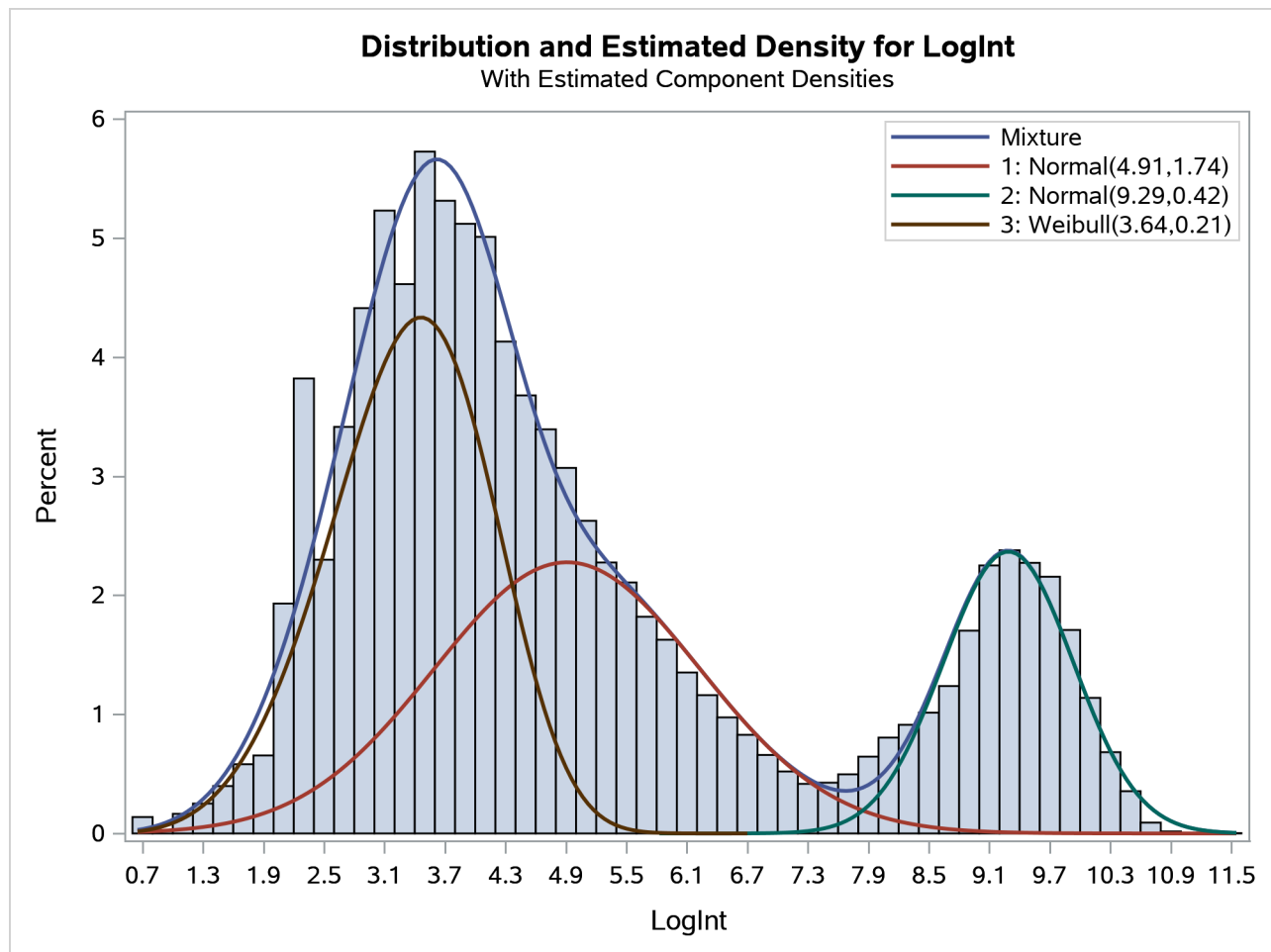
Fit Statistics	
-2 Log Likelihood	564431
AIC (Smaller is Better)	564447
AICC (Smaller is Better)	564447
BIC (Smaller is Better)	564526
Pearson Statistic	141228
Effective Parameters	8
Effective Components	3

Parameter Estimates for Normal Model					
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z
1	Intercept	4.9106	0.02604	188.56	<.0001
2	Intercept	9.2883	0.005031	1846.28	<.0001
1	Variance	1.7410	0.02753		
2	Variance	0.4158	0.005086		

Parameter Estimates for Weibull Model						
Component	Parameter	Estimate	Standard Error	z Value	Pr >  z	Inverse Linked Estimate
3	Intercept	1.2908	0.002790	462.71	<.0001	3.6358
3	Scale	0.2093	0.001311			

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	GLogit(Prob)	Linked Scale Standard Error	z Value	Pr >  z
1	0.3745	-0.1505	0.03678	-4.09	<.0001
2	0.1902	-0.8280	0.01922	-43.08	<.0001
3	0.4353	0			

All components are active; no collapsing of components occurred. However, a closer look at the “Parameter Estimates” tables in [Output 55.2.7](#) shows an important difference from the tables in [Output 55.2.4](#). The means of the two normal distributions are now 4.9106 and 9.2883. Previously, the means were 3.3415 and 4.8940. The “position” of the Weibull distribution has moved from right to left, and the third component is now modeled by a symmetric normal distribution ([Output 55.2.8](#)). The mixture probabilities have also changed—in particular, for the first and third component.

**Output 55.2.8** Three-Component Model with Default Starting Values

Such switching is not uncommon in mixture modeling. As judged by the information criteria, the model in which the Weibull distribution is the component with the smallest mean does not fit the data as well as the first model in which the specification of the starting values guided the optimization towards placing the normal distributions first. The converged solution found in the last HPFMM run represents a local minimum of the log-likelihood surface. There are other local minima—for example, when components are removed from the model, which is tantamount to estimating the associated mixture probabilities as zero.

### Example 55.3: Enforcing Homogeneity Constraints: Count and Dispersion—It Is All Over!

The following example demonstrates how you can use either the `EQUATE=` option in the `MODEL` statement or the `RESTRICT` statement to impose homogeneity constraints on chosen model effects.

The data for this example were presented by Margolin, Kaplan, and Zeiger (1981) and analyzed by various authors applying a number of techniques. The following `DATA` step shows the number of revertant salmonella colonies (variable `num`) at six levels of quinoline dosing (variable `dose`). There are three replicate plates at each dose of quinoline.



```

data assay;
  label dose = 'Dose of quinoline (microg/plate) '
        num = 'Observed number of colonies';
  input dose @;
  logd = log(dose+10);
  do i=1 to 3; input num@; output; end;
  datalines;
    0  15 21 29
   10  16 18 21
   33  16 26 33
  100  27 41 60
  333  33 38 41
 1000  20 27 42
;

```

The basic notion is that the data are overdispersed relative to a Poisson distribution in which the logarithm of the mean count is modeled as a linear regression in dose (in  $\mu\text{g}/\text{plate}$ ) and in the derived variable  $\log\{\text{dose} + 10\}$  (Lawless 1987). The log of the expected count of revertants is thus

$$\beta_0 + \beta_1 \text{dose} + \beta_2 \log\{\text{dose} + 10\}$$

The following statements fit a standard Poisson regression model to these data:

```

proc hpfmm data=assay;
  model num = dose logd / dist=Poisson;
run;

```

The Pearson statistic for this model is rather large compared to the number of degrees of freedom ( $18 - 3 = 15$ ). The ratio  $46.2707/15 = 3.08$  indicates an overdispersion problem in the Poisson model ([Output 55.3.1](#)).

### Output 55.3.1 Result of Fitting Poisson Regression Models

#### The HPFMM Procedure

Number of Observations Read	18
Number of Observations Used	18

Fit Statistics	
-2 Log Likelihood	136.3
AIC (Smaller is Better)	142.3
AICC (Smaller is Better)	144.0
BIC (Smaller is Better)	144.9
Pearson Statistic	46.2707

Parameter Estimates for Poisson Model				
Standard				
Effect	Estimate	Error	z Value	Pr >  z
Intercept	2.1728	0.2184	9.95	<.0001
dose	-0.00101	0.000245	-4.13	<.0001
logd	0.3198	0.05700	5.61	<.0001

Breslow (1984) accounts for overdispersion by including a random effect in the predictor for the log rate and applying a quasi-likelihood technique to estimate the parameters. Wang et al. (1996) examine these data using mixtures of Poisson regression models. They fit several two- and three-component Poisson regression mixtures. Examining the log likelihoods, AIC, and BIC criteria, they eventually settle on a two-component model in which the intercepts vary by category and the regression coefficients are the same. This mixture model can be written as

$$f(y) = \pi \frac{1}{y!} \lambda_1^y \exp\{-\lambda_1\} + (1 - \pi) \frac{1}{y!} \lambda_2^y \exp\{-\lambda_2\}$$

$$\lambda_1 = \exp\{\beta_{01} + \beta_1 \text{dose} + \beta_2 \log[\text{dose} + 10]\}$$

$$\lambda_2 = \exp\{\beta_{02} + \beta_1 \text{dose} + \beta_2 \log[\text{dose} + 10]\}$$

This model is fit with the HPFMM procedure with the following statements:

```
proc hpfmm data=assay;
  model num = dose logd / dist=Poisson k=2
    equate=effects(dose logd);
run;
```

The **EQUATE=** option in the **MODEL** statement places constraints on the optimization and makes the coefficients for dose and logd homogeneous across components in the model. **Output 55.3.2** displays the “Fit Statistics” and parameter estimates in the mixture. The Pearson statistic is drastically reduced compared to the Poisson regression model in **Output 55.3.1**. With  $18 - 5 = 13$  degrees of freedom, the ratio of the Pearson and the degrees of freedom is now  $16.1573/13 = 1.2429$ . Note that the effective number of parameters was used to compute the degrees of freedom, not the total number of parameters, because of the equality constraints.

**Output 55.3.2** Result for Two-Component Poisson Regression Mixture

The HPFMM Procedure					
Fit Statistics					
-2 Log Likelihood		121.8			
AIC (Smaller is Better)		131.8			
AICC (Smaller is Better)		136.8			
BIC (Smaller is Better)		136.3			
Pearson Statistic		16.1573			
Effective Parameters		5			
Effective Components		2			

Parameter Estimates for Poisson Model					
		Standard			
Component	Effect	Estimate	Error	z Value	Pr >  z
1	Intercept	1.9097	0.2654	7.20	<.0001
1	dose	-0.00126	0.000273	-4.62	<.0001
1	logd	0.3639	0.06602	5.51	<.0001
2	Intercept	2.4770	0.2731	9.07	<.0001
2	dose	-0.00126	0.000273	-4.62	<.0001
2	logd	0.3639	0.06602	5.51	<.0001

**Output 55.3.2** *continued*

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	Logit(Prob)	Linked Scale		
			Standard Error	z Value	Pr >  z
1	0.8173	1.4984	0.6875	2.18	0.0293
2	0.1827	-1.4984			

You could also have used **RESTRICT** statements to impose the homogeneity constraints on the model fit, as shown in the following statements:

```
proc hpfmm data=assay;
  model num = dose logd / dist=Poisson k=2;
  restrict 'common dose' dose 1, dose -1;
  restrict 'common logd' logd 1, logd -1;
run;
```

The first **RESTRICT** statement equates the coefficients for the dose variable in the two components, and the second **RESTRICT** statement accomplishes the same for the coefficients of the logd variable. If the right-hand side of a restriction is not specified, PROC HPFMM defaults to equating the left-hand side of the restriction to zero. The “Linear Constraints” table in [Output 55.3.3](#) shows that both linear equality constraints are active. The parameter estimates match the previous HPFMM run.

**Output 55.3.3** Result for Two-Component Mixture with RESTRICT Statements**The HPFMM Procedure**

Linear Constraints at Solution					
Label	k = 1	k = 2	Constraint Active		
common dose	dose	-dose	=	0	Yes
common logd	logd	-logd	=	0	Yes

Parameter Estimates for Poisson Model					
Component	Effect	Estimate	Standard		
			Error	z Value	Pr >  z
1	Intercept	1.9097	0.2654	7.20	<.0001
1	dose	-0.00126	0.000273	-4.62	<.0001
1	logd	0.3639	0.06602	5.51	<.0001
2	Intercept	2.4770	0.2731	9.07	<.0001
2	dose	-0.00126	0.000273	-4.62	<.0001
2	logd	0.3639	0.06602	5.51	<.0001

Parameter Estimates for Mixing Probabilities					
Component	Mixing Probability	Logit(Prob)	Linked Scale		
			Standard Error	z Value	Pr >  z
1	0.8173	1.4984	0.6875	2.18	0.0293
2	0.1827	-1.4984			

Wang et al. (1996) note that observation 12 with a revertant colony count of 60 is comparably high. The following statements remove the observation from the analysis and fit their selected model:

```
proc hpfmm data=assay(where=(num ne 60));
  model num = dose logd / dist=Poisson k=2
          equate=effects(dose logd);
run;
```

#### Output 55.3.4 Result for Two-Component Model without Outlier

##### The HPFMM Procedure

Fit Statistics	
-2 Log Likelihood	111.5
AIC (Smaller is Better)	121.5
AICC (Smaller is Better)	126.9
BIC (Smaller is Better)	125.6
Pearson Statistic	16.5987
Effective Parameters	5
Effective Components	2

##### Parameter Estimates for Poisson Model

Component	Effect	Estimate	Standard Error	z Value	Pr >  z
1	Intercept	2.2272	0.3022	7.37	<.0001
1	dose	-0.00065	0.000445	-1.46	0.1440
1	logd	0.2432	0.1045	2.33	0.0199
2	Intercept	2.5477	0.3331	7.65	<.0001
2	dose	-0.00065	0.000445	-1.46	0.1440
2	logd	0.2432	0.1045	2.33	0.0199

##### Parameter Estimates for Mixing Probabilities

		Linked Scale			
Component	Mixing Probability	Logit(Prob)	Standard Error	z Value	Pr >  z
1	0.5777	0.3134	1.7261	0.18	0.8559
2	0.4223	-0.3134			

The ratio of Pearson Statistic over degrees of freedom (12) is only slightly worse than in the previous model; the loss of 5% of the observations carries a price (Output 55.3.4). The parameter estimates for the two intercepts are now fairly close. If the intercepts were identical, then the two-component model would collapse to the Poisson regression model:

```
proc hpfmm data=assay(where=(num ne 60));
  model num = dose logd / dist=Poisson;
run;
```

**Output 55.3.5** Result of Fitting Poisson Regression Model without Outlier**The HPFMM Procedure**

Number of Observations Read	17
Number of Observations Used	17

Fit Statistics	
-2 Log Likelihood	114.1
AIC (Smaller is Better)	120.1
AICC (Smaller is Better)	121.9
BIC (Smaller is Better)	122.5
Pearson Statistic	27.8008

Parameter Estimates for Poisson Model				
Effect	Estimate	Standard Error	z Value	Pr >  z
Intercept	2.3164	0.2244	10.32	<.0001
dose	-0.00072	0.000258	-2.78	0.0055
logd	0.2603	0.05996	4.34	<.0001

Compared to the same model applied to the full data, the Pearson statistic is much reduced (compare 46.2707 in [Output 55.3.1](#) to 27.8008 in [Output 55.3.5](#)). The outlier—or *overcount*, if you will—induces at least some of the *overdispersion*.

---

## References

- Aldrich, J. (1997). “R. A. Fisher and the Making of Maximum Likelihood, 1912–1922.” *Statistical Science* 12:162–176.
- Breslow, N. E. (1984). “Extra-Poisson Variation in Log-Linear Models.” *Journal of the Royal Statistical Society, Series C* 33:38–44.
- Cameron, A. C., and Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge: Cambridge University Press.
- Celeux, G., Forbes, F., Robert, C. P., and Titterton, D. M. (2006). “Deviance Information Criteria for Missing Data Models.” *Bayesian Analysis* 1:651–674.
- Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society, Series B* 39:1–38.
- Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981). “An Adaptive Nonlinear Least-Squares Algorithm.” *ACM Transactions on Mathematical Software* 7:348–368.
- Dennis, J. E., and Mei, H. H. W. (1979). “Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values.” *Journal of Optimization Theory and Applications* 28:453–482.

- Eskow, E., and Schnabel, R. B. (1991). "Algorithm 695: Software for a New Modified Cholesky Factorization." *ACM Transactions on Mathematical Software* 17:306–312.
- Everitt, B. S., and Hand, D. J. (1981). *Finite Mixture Distributions*. London: Chapman & Hall.
- Ferrari, S. L. P., and Cribari-Neto, F. (2004). "Beta Regression for Modelling Rates and Proportions." *Journal of Applied Statistics* 31:799–815.
- Fisher, R. A. (1921). "On the 'Probable Error' of a Coefficient of Correlation Deduced from a Small Sample." *Metron* 1:3–32.
- Fletcher, R. (1987). *Practical Methods of Optimization*. 2nd ed. Chichester, UK: John Wiley & Sons.
- Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. New York: Springer.
- Gamerman, D. (1997). "Sampling from the Posterior Distribution in Generalized Linear Models." *Statistics and Computing* 7:57–68.
- Gay, D. M. (1983). "Subroutines for Unconstrained Minimization." *ACM Transactions on Mathematical Software* 9:503–524.
- Geweke, J. (1992). "Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments." In *Bayesian Statistics*, vol. 4, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, 169–193. Oxford: Clarendon Press.
- Griffiths, D. A. (1973). "Maximum Likelihood Estimation for the Beta-Binomial Distribution and an Application to the Household Distribution of the Total Number of Cases of a Disease." *Biometrics* 29:637–648.
- Haseman, J. K., and Kupper, L. L. (1979). "Analysis of Dichotomous Response Data from Certain Toxicological Experiments." *Biometrics* 35:281–293.
- Joe, H., and Zhu, R. (2005). "Generalized Poisson Distribution: The Property of Mixture of Poisson and Comparison with Negative Binomial Distribution." *Biometrical Journal* 47:219–229.
- Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998). "Markov Chain Monte Carlo in Practice: A Roundtable Discussion." *American Statistician* 52:93–100.
- Lawless, J. F. (1987). "Negative Binomial and Mixed Poisson Regression." *Canadian Journal of Statistics* 15:209–225.
- Margolin, B. H., Kaplan, N. L., and Zeiger, E. (1981). "Statistical Analysis of the Ames Salmonella Microsome Test." *Proceedings of the National Academy of Sciences* 76:3779–3783.
- McLachlan, G. J., and Peel, D. (2000). *Finite Mixture Models*. New York: John Wiley & Sons.
- Moré, J. J. (1978). "The Levenberg-Marquardt Algorithm: Implementation and Theory." In *Lecture Notes in Mathematics*, vol. 30, edited by G. A. Watson, 105–116. Berlin: Springer-Verlag.
- Moré, J. J., and Sorensen, D. C. (1983). "Computing a Trust-Region Step." *SIAM Journal on Scientific and Statistical Computing* 4:553–572.
- Morel, J. G., and Nagaraj, N. K. (1993). "A Finite Mixture Distribution for Modelling Multinomial Extra Variation." *Biometrika* 80:363–371.

- Morel, J. G., and Neerchal, N. K. (1997). "Clustered Binary Logistic Regression in Teratology Data Using a Finite Mixture Distribution." *Statistics in Medicine* 16:2843–2853.
- Neerchal, N. K., and Morel, J. G. (1998). "Large Cluster Results for Two Parametric Multinomial Extra Variation Models." *Journal of the American Statistical Association* 93:1078–1087.
- Pearson, K. (1915). "On Certain Types of Compound Frequency Distributions in Which the Components Can Be Individually Described by Binomial Series." *Biometrika* 11:139–144.
- Raftery, A. E. (1996). "Hypothesis Testing and Model Selection." In *Markov Chain Monte Carlo in Practice*, edited by W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, 163–188. London: Chapman & Hall.
- Richardson, S. (2002). "Discussion of Spiegelhalter et al." *Journal of the Royal Statistical Society, Series B* 64:631.
- Roeder, K. (1990). "Density Estimation with Confidence Sets Exemplified by Superclusters and Voids in the Galaxies." *Journal of the American Statistical Association* 85:617–624.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002). "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society, Series B* 64:583–616. With discussion.
- Titterton, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. New York: John Wiley & Sons.
- Viallefont, V., Richardson, S., and Greene, P. J. (2002). "Bayesian Analysis of Poisson Mixtures." *Journal of Nonparametric Statistics* 14:181–202.
- Wang, P., Puterman, M. L., Cockburn, I., and Le, N. (1996). "Mixed Poisson Regression Models with Covariate Dependent Rates." *Biometrics* 52:381–400.
- Williams, D. A. (1975). "The Analysis of Binary Responses from Toxicological Experiments Involving Reproduction and Teratogenicity." *Biometrics* 31:949–952.

# Subject Index

- alpha level
  - HPFMM procedure, 4480, 4488
- Bayes information
  - HPFMM procedure, 4514
- Bayesian analysis
  - HPFMM procedure, 4466
- Bayesian fit statistics
  - HPFMM procedure, 4514
- Bernoulli distribution
  - HPFMM procedure, 4481
- beta distribution
  - HPFMM procedure, 4481
- beta-binomial distribution
  - HPFMM procedure, 4481
- binary distribution
  - HPFMM procedure, 4481
- binomial distribution
  - HPFMM procedure, 4481
- class level
  - HPFMM procedure, 4461, 4511
- computational method
  - HPFMM procedure, 4506
- confidence limits
  - model parameters (HPFMM), 4480, 4489
- constrained analysis
  - HPFMM procedure, 4490
- convergence criterion
  - HPFMM procedure, 4456, 4459
- convergence status
  - HPFMM procedure, 4512
- default output
  - HPFMM procedure, 4511
- effect
  - name length (HPFMM), 4460
- examples, HPFMM
  - binomial data, 4519
  - cattle feeding data, 4526
  - logistic model, binomial cluster, 4519
  - ossification data, 4519
  - PROBMODEL specification, 4519
  - salmonella assay, 4534
  - three-component mixture, 4528
  - Weibull distribution, 4528
- exponential distribution
  - HPFMM procedure, 4481
- finite mixture models
  - See HPFMM procedure, 4426
- fit statistics
  - HPFMM procedure, 4512
- folded normal distribution
  - HPFMM procedure, 4481
- frequency variable
  - HPFMM procedure, 4477
- gamma distribution
  - HPFMM procedure, 4481
- Gaussian distribution
  - HPFMM procedure, 4481
- generalized Poisson distribution
  - HPFMM procedure, 4481
- geometric distribution
  - HPFMM procedure, 4481
- heavy-tailed density
  - finite mixture models (HPFMM), 4482, 4495
- HPFMM procedure, 4426
  - alpha level, 4480, 4488
  - Bayes information, 4514
  - Bayesian analysis, 4466
  - Bayesian fit statistics, 4514
  - Bernoulli distribution, 4481
  - beta distribution, 4481
  - beta-binomial distribution, 4481
  - binary distribution, 4481
  - binomial distribution, 4481
  - centering and scaling, 4460
  - class level, 4461, 4511
  - computational method, 4506
  - confidence limits, 4480, 4489
  - constrained analysis, 4490
  - convergence criterion, 4456, 4459
  - convergence status, 4512
  - default output, 4511
  - effect name length, 4460
  - exponential distribution, 4481
  - fit statistics, 4512
  - folded normal distribution, 4481
  - function-based convergence criteria, 4456, 4458
  - gamma distribution, 4481
  - Gaussian distribution, 4481
  - generalized Poisson distribution, 4481
  - geometric distribution, 4481
  - gradient-based convergence criteria, 4456, 4459
  - heavy-tailed density, 4482, 4495



- hurdle model, 4430, 4436
- input data sets, 4457
- introductory example, 4430
- inverse Gaussian distribution, 4481
- iteration details, 4459
- iteration history, 4512
- link function, 4484, 4489
- lognormal distribution, 4481
- mixing probabilities, 4514
- model information, 4511
- multimodal density, 4444, 4445, 4495
- multithreading, 4488, 4506
- negative binomial distribution, 4481
- normal distribution, 4481
- number of observations, 4511
- ODS Graphics, 4462, 4517
- ODS table names, 4515
- offset variable, 4485
- optimization information, 4512
- optimization technique, 4507
- output data set, 4510
- overdispersion, 4429, 4430, 4436, 4438, 4493–4495, 4517, 4525, 4535
- parameter estimates, 4513
- parameterization, 4505
- performance information, 4511
- plots, 4427
- Poisson distribution, 4481
- posterior autocorrelations, 4515
- posterior intervals, 4515
- posterior summaries, 4514
- prior distributions, 4514
- random number seed, 4465
- residual variance tolerance, 4465
- response level ordering, 4479
- response profile, 4512
- response variable options, 4479
- restricted analysis, 4490
- statistical graphics, 4517
- t* distribution, 4481
- Weibull distribution, 4481
- weighting, 4492
- zero-inflated model, 4427, 4430, 4436, 4437, 4482, 4497

hurdle model

- finite mixture models (HPFMM), 4430, 4436

inverse Gaussian distribution

- HPFMM procedure, 4481

iteration details

- HPFMM procedure, 4459

iteration history

- HPFMM procedure, 4512

link function

- HPFMM procedure, 4484, 4489

lognormal distribution

- HPFMM procedure, 4481

mixing probabilities

- HPFMM procedure, 4514

mixture model (HPFMM)

- parameterization, 4505

model

- information (HPFMM), 4511

multimodal density

- finite mixture models (HPFMM), 4444, 4445, 4495

multithreading

- HPFMM procedure, 4488, 4506

negative binomial distribution

- HPFMM procedure, 4481

normal distribution

- HPFMM procedure, 4481

number of observations

- HPFMM procedure, 4511

ODS Graphics

- HPFMM procedure, 4462, 4517

offset variable

- HPFMM procedure, 4485

optimization information

- HPFMM procedure, 4512

optimization technique

- HPFMM procedure, 4507

options summary

- BAYES statement, 4467
- MODEL statement (HPFMM), 4478
- PROC HPFMM statement, 4454

output data set

- HPFMM procedure, 4510

overdispersion

- finite mixture models (HPFMM), 4429, 4430, 4436, 4438, 4493–4495, 4517, 4525, 4535

parameter estimates

- HPFMM procedure, 4513

parameterization

- HPFMM procedure, 4505
- mixture model (HPFMM), 4505

performance information

- HPFMM procedure, 4511

plots

- finite mixture models (HPFMM), 4427

Poisson distribution

- HPFMM procedure, 4481

posterior autocorrelations

- HPFMM procedure, 4515

posterior intervals

- HPFMM procedure, [4515](#)
- posterior summaries
  - HPFMM procedure, [4514](#)
- prior distributions
  - HPFMM procedure, [4514](#)
- probability distributions
  - HPFMM procedure, [4481](#)
- random number seed
  - HPFMM procedure, [4465](#)
- response level ordering
  - HPFMM procedure, [4479](#)
- response profile
  - HPFMM procedure, [4512](#)
- response variable options
  - HPFMM procedure, [4479](#)
- restricted analysis
  - HPFMM procedure, [4490](#)
- reverse response level ordering
  - HPFMM procedure, [4479](#)
- statistical graphics
  - HPFMM procedure, [4517](#)
- $t$  distribution
  - HPFMM procedure, [4481](#)
- Weibull distribution
  - HPFMM procedure, [4481](#)
- weighting
  - HPFMM procedure, [4492](#)
- zero-inflated model
  - finite mixture models (HPFMM), [4427](#), [4430](#),  
[4436](#), [4437](#), [4482](#), [4497](#)



# Syntax Index

- ABSCONV option
  - PROC HPFMM statement, [4456](#)
- ABSFCNV option
  - PROC HPFMM statement, [4456](#)
- ABSGCONV option
  - PROC HPFMM statement, [4456](#)
- ABSGTOL option
  - PROC HPFMM statement, [4456](#)
- ABSTOL option
  - PROC HPFMM statement, [4456](#)
- ALLSTATS option
  - OUTPUT statement (HPFMM), [4487](#)
- ALPHA= option
  - MODEL statement (HPFMM), [4480](#)
  - PROBMODEL statement (HPFMM), [4488](#)
- BAYES statement
  - HPFMM procedure, [4466](#)
- BETAPRIORPARMS option
  - BAYES statement (HPFMM), [4467](#)
- BY statement
  - HPFMM procedure, [4476](#)
- CINFO option
  - PROC HPFMM statement, [4456](#)
- CL option
  - MODEL statement (HPFMM), [4480](#)
  - PROBMODEL statement (HPFMM), [4489](#)
- CLASS statement
  - HPFMM procedure, [4476](#)
- COMPINFO option
  - PROC HPFMM statement, [4456](#)
- COMPONENTINFO option
  - PROC HPFMM statement, [4456](#)
- CORR option
  - PROC HPFMM statement, [4457](#)
- COV option
  - PROC HPFMM statement, [4457](#)
- COVI option
  - PROC HPFMM statement, [4457](#)
- CRIT= option
  - PROC HPFMM statement, [4457](#)
- CRITERION= option
  - PROC HPFMM statement, [4457](#)
- DATA= option
  - PROC HPFMM statement, [4457](#)
- DESCENDING option
  - MODEL statement, [4479](#)
- DIAGNOSTICS option
  - BAYES statement (HPFMM), [4468](#)
- DIST= option
  - MODEL statement (HPFMM), [4481](#)
- DISTRIBUTION= option
  - MODEL statement (HPFMM), [4481](#)
- DIVISOR= option
  - RESTRICT statement (HPFMM), [4492](#)
- EQUATE= option
  - MODEL statement (HPFMM), [4483](#)
- ESTIMATE= option
  - BAYES statement (HPFMM), [4470](#)
- EVENT= option
  - MODEL statement, [4479](#)
- EXCLUDE= option
  - PROC HPFMM statement, [4457](#)
- EXCLUSION= option
  - PROC HPFMM statement, [4457](#)
- FCONV option
  - PROC HPFMM statement, [4458](#)
- FCONV2 option
  - PROC HPFMM statement, [4458](#)
- FITDETAILS option
  - PROC HPFMM statement, [4459](#)
- FREQ statement
  - HPFMM procedure, [4477](#)
- FTOL option
  - PROC HPFMM statement, [4458](#)
- FTOL2 option
  - PROC HPFMM statement, [4458](#)
- GCONV option
  - PROC HPFMM statement, [4459](#)
- GTOL option
  - PROC HPFMM statement, [4459](#)
- HESSIAN option
  - PROC HPFMM statement, [4459](#)
- HPFMM procedure, [4454](#)
  - BAYES statement, [4466](#)
  - FREQ statement, [4477](#)
  - ID statement, [4477](#)
  - MODEL statement, [4477](#)
  - OUTPUT statement, [4485](#)
  - PERFORMANCE statement, [4488](#)
  - PROBMODEL statement, [4488](#)
  - PROC HPFMM statement, [4454](#)

RESTRICT statement, 4490  
 syntax, 4454  
 WEIGHT statement, 4492  
 HPFMM procedure, BAYES statement, 4466  
   BETAPRIORPARMS option, 4467  
   DIAGNOSTICS option, 4468  
   ESTIMATE= option, 4470  
   INITIAL= option, 4470  
   METROPOLIS option, 4471  
   MIXPRIORPARMS option, 4470  
   MUPRIORPARMS option, 4471  
   NBI= option, 4472  
   NMC= option, 4472  
   OUTPOST= option, 4472  
   PHIPRIORPARMS option, 4472  
   PRIOROPTIONS option, 4473  
   PRIOROPTS option, 4473  
   STATISTICS option, 4474  
   SUMMARIES option, 4474  
   THIN= option, 4475  
   THINNING= option, 4475  
   TIMEINC= option, 4475  
 HPFMM procedure, BY statement, 4476  
 HPFMM procedure, CLASS statement, 4476  
   UPCASE option, 4476  
 HPFMM procedure, FREQ statement, 4477  
 HPFMM procedure, ID statement, 4477  
 HPFMM procedure, MODEL statement, 4477  
   ALPHA= option, 4480  
   CL option, 4480  
   DESCENDING option, 4479  
   DIST= option, 4481  
   DISTRIBUTION= option, 4481  
   EQUATE= option, 4483  
   EVENT= option, 4479  
   K= option, 4483  
   KMAX= option, 4483  
   KMIN= option, 4484  
   KRESTART option, 4484  
   LABEL= option, 4484  
   LINK= option, 4484  
   NOINT option, 4484  
   NUMBER= option, 4483  
   OFFSET= option, 4485  
   ORDER= option, 4479  
   PARAMETERS option, 4485  
   PARMS option, 4485  
   REFERENCE= option, 4480  
 HPFMM procedure, OUTPUT statement, 4485  
   ALLSTATS option, 4487  
   keyword= option, 4485  
   OUT= option, 4485  
   PREDTYPE option, 4487  
 HPFMM procedure, PERFORMANCE statement, 4488  
 HPFMM procedure, PROBMODEL statement, 4488  
   ALPHA= option, 4488  
   CL option, 4489  
   LINK= option, 4489  
   NOINT option, 4489  
   PARAMETERS option, 4489  
   PARMS option, 4489  
 HPFMM procedure, PROC HPFMM statement, 4454  
   ABSCONV option, 4456  
   ABSFCONV option, 4456  
   ABSFTOL option, 4456  
   ABSGCONV option, 4456  
   ABSGTOL option, 4456  
   ABSTOL option, 4456  
   CINFO option, 4456  
   COMPINFO option, 4456  
   COMPONENTINFO option, 4456  
   CORR option, 4457  
   COV option, 4457  
   COVI option, 4457  
   CRIT= option, 4457  
   CRITERION= option, 4457  
   DATA= option, 4457  
   EXCLUDE= option, 4457  
   EXCLUSION= option, 4457  
   FCONV option, 4458  
   FCONV2 option, 4458  
   FITDETAILS option, 4459  
   FTOL option, 4458  
   FTOL2 option, 4458  
   GCONV option, 4459  
   GTOL option, 4459  
   HESSIAN option, 4459  
   INVALIDLOGL= option, 4459  
   ITDETAILS option, 4459  
   MAXFUNC= option, 4460  
   MAXITER= option, 4460  
   MAXTIME= option, 4460  
   MEMBERSHIP= option, 4461  
   NAMELEN= option, 4460  
   NOCENTER option, 4460  
   NOCLPRINT option, 4461  
   NOITPRINT option, 4461  
   NOPRINT option, 4461  
   PARMSTYLE= option, 4461  
   PARTIAL= option, 4461  
   PLOTS option, 4462  
   SEED= option, 4465  
   SINGCHOL= option, 4465  
   SINGULAR= option, 4465  
   TECHNIQUE= option, 4465  
   ZEROPROB= option, 4466

HPFMM procedure, RESTRICT statement, 4490  
DIVISOR= option, 4492  
HPFMM procedure, WEIGHT statement, 4492

ID statement  
HPFMM procedure, 4477  
INITIAL= option  
BAYES statement (HPFMM), 4470  
INVALIDLOGL= option  
PROC HPFMM statement, 4459  
ITDETAILS option  
PROC HPFMM statement, 4459

K= option  
MODEL statement (HPFMM), 4483  
keyword= option  
OUTPUT statement (HPFMM), 4485  
KMAX= option  
MODEL statement (HPFMM), 4483  
KMIN= option  
MODEL statement (HPFMM), 4484  
KRESTART option  
MODEL statement (HPFMM), 4484

LABEL= option  
MODEL statement (HPFMM), 4484  
LINK= option  
MODEL statement (HPFMM), 4484  
PROBMODEL statement (HPFMM), 4489

MAXFUNC= option  
PROC HPFMM statement, 4460  
MAXITER= option  
PROC HPFMM statement, 4460  
MAXTIME= option  
PROC HPFMM statement, 4460  
MEMBERSHIP= option  
PROC HPFMM statement, 4461  
METROPOLIS option  
BAYES statement (HPFMM), 4471  
MIXPRIORPARMS option  
BAYES statement (HPFMM), 4470  
MODEL statement  
HPFMM procedure, 4477  
MUPRIORPARMS option  
BAYES statement (HPFMM), 4471

NAMELEN= option  
PROC HPFMM statement, 4460  
NBI= option  
BAYES statement (HPFMM), 4472  
NMC= option  
BAYES statement (HPFMM), 4472  
NOCENTER option  
PROC HPFMM statement, 4460

NOCLPRINT option  
PROC HPFMM statement, 4461  
NOINT option  
MODEL statement (HPFMM), 4484  
PROBMODEL statement (HPFMM), 4489  
NOITPRINT option  
PROC HPFMM statement, 4461  
NOPRINT option  
PROC HPFMM statement, 4461  
NUMBER= option  
MODEL statement (HPFMM), 4483

OFFSET= option  
MODEL statement (HPFMM), 4485  
ORDER= option  
MODEL statement, 4479  
OUT= option  
OUTPUT statement (HPFMM), 4485  
OUTPOST= option  
BAYES statement (HPFMM), 4472  
OUTPUT statement  
HPFMM procedure, 4485

PARAMETERS option  
MODEL statement (HPFMM), 4485  
PROBMODEL statement (HPFMM), 4489  
PARMS option  
MODEL statement (HPFMM), 4485  
PROBMODEL statement (HPFMM), 4489

PARMSTYLE= option  
PROC HPFMM statement, 4461  
PARTIAL= option  
PROC HPFMM statement, 4461  
PERFORMANCE statement  
HPFMM procedure, 4488  
PHIPRIORPARMS option  
BAYES statement (HPFMM), 4472  
PLOTS option  
PROC HPFMM statement, 4462  
PREDTYPE option  
OUTPUT statement (HPFMM), 4487

PRIOROPTIONS option  
BAYES statement (HPFMM), 4473  
PRIOROPTS option  
BAYES statement (HPFMM), 4473  
PROBMODEL statement  
HPFMM procedure, 4488  
PROC HPFMM procedure, PROC HPFMM statement  
SINGRES= option, 4465  
PROC HPFMM statement  
HPFMM procedure, 4454

REFERENCE= option  
MODEL statement (HPFMM), 4480  
RESTRICT statement

HPFMM procedure, [4490](#)

SEED= option

PROC HPFMM statement, [4465](#)

SINGCHOL= option

PROC HPFMM statement, [4465](#)

SINGRES= option

PROC HPFMM statement, [4465](#)

SINGULAR= option

PROC HPFMM statement, [4465](#)

STATISTICS option

BAYES statement (HPFMM), [4474](#)

SUMMARIES option

BAYES statement (HPFMM), [4474](#)

syntax

HPFMM procedure, [4454](#)

TECHNIQUE= option

PROC HPFMM statement, [4465](#)

THIN= option

BAYES statement (HPFMM), [4475](#)

THINNING= option

BAYES statement (HPFMM), [4475](#)

TIMEINC= option

BAYES statement (HPFMM), [4475](#)

UPCASE option

CLASS statement (HPFMM), [4476](#)

WEIGHT statement

HPFMM procedure, [4492](#)

ZEROPROB= option

PROC HPFMM statement, [4466](#)