MacMCMC (v1.6)
User Guide Examples

Michael P. McLaughlin

November, 2022

This is a brief synopsis of the examples included with the *MacMCMC* (v1.6) package. Details and output for each to follow. Goodness-of-fit plots created by *MacMCMC* show, where requested, the (optional) 95-percent credible interval band for predicted values of future observations. With one exception (Fig. 27), marginal plots are raw (not smoothed).

Unless otherwise noted, all examples used the default run options. Runtimes[1] cited include the MCMC run, refining the MAP parameters, integration to get the marginal likelihood, unless relabeled, and plotting the first marginal. Runtimes varied from 2 s (TriplePoint) to 10 m 30 s (Salaries).

Table 1: Examples

1	Body	Height (cm) vs. weight (kg) of adult men
2	Carbon-14	^{14}C specific activity (counts/min-g) vs. age (years)
3	Daytime	Daytime in Boston, MA, USA (min) vs. day number
4	Enzyme	Concentration vs. rate
5	FishCount	Count of fish caught
6	Gumbel	US major-league baseball: best batting averages (1876-2019)
7	Hale-Bopp	Cyanide radial release: rate vs. distance
8	Hyphens	Count per page of line-ending hyphens
9	Iris	Fischer's iris data
10	Lizards	Capture-recapture data
11	MP	Historical melting point data for *n*-octadecane
12	NormalTemp	Normal body temperature for adult males
13	Salaries	USA college faculty
14	SAT	Northern Viriginia, USA, schools (2014)
15	TriplePoint	Data for the triple-point temperature of *n*-nonadecane

Table 2: Functionality vs. Examples

Functionality (or example type)	Example # 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Continuous Distribution	•					•			•		•	•	•	•	•
Discrete Distribution					•			•		•					
Equation		•	•	•			•								
BivariateNormal Dist.	•								•						
Generic Distribution					•	•				•					•
Truncated Distribution															•
Mixture Distribution					•				•				•		
Relabeled Mixture									•				•		
Goodness-of-fit			•				•	•				•			•
Multidimensional Input	•								•						
Extras			•	•			•	•						•	

[1] on a standard 2017 iMac

List of Models

1 Weight-height Pairs 4
2 Carbon-14 Specific Activity 9
3 Daytime for Boston, MA, USA 11
4 Enzyme Activity (model 1) 16
5 Enzyme Activity (model 2) 17
6 FishCount 20
7 Gumbel Model for Best Batting Averages 22
8 Hale-Bopp 25
9 Hyphens 28
10 Iris 32
11 Lizards 34
12 Melting Point of *n*-Octadecane 36
13 Normal Temperature 38
14 Salaries (model 1) 41
15 Salaries (model 2) 43
16 SAT 47
17 TriplePoint 49

1 Body

Here, each datum is a {weight, height} ordered pair so the likelihood is bivariate.
Runtime: 2 m 30 s

Model 1: Weight-height Pairs

```
Constants:
N = 247;  // # of points (rows)
Data:
WtHt[N][2];
Variables:
m1, m2, sig1, sig2, rho, row;
Priors:
m1 ~ Uniform(50, 100);
m2 ~ Uniform(150, 250);
sig1 ~ Jeffreys(0.1, 20);
sig2 ~ Jeffreys(0.1, 20);
rho ~ Uniform(-1, 1);
Likelihood:
for (row, 1:N) {
   WtHt[row][] ~ BivariateNormal(m1, m2, sig1, sig2, rho);
}
Extras:
Monitored:
m1, m2, sig1, sig2, rho;
```

Reports are structured as shown on the following page. The sample size reported does not include thinned iterations.[2] The marginal likelihood is quite small in this example mainly because there are a lot of points.

This run used the default parameters. Unless otherwise noted, the same is true for all of these examples. MAP parameters should be stable since they are further optimized. Gelman-Rubin statistics should all be very close to one.

Output precision and smoothness of marginal plots can be improved by collecting a larger sample of the posterior.

The trace can be used to gather further information and/or to make additional plots. To show this, a bivariate plot was constructed[3] showing the data and (mean) posterior model together (Fig. 5).

[2] i.e., this run comprised 2,500,500 iterations (thinned iterations not saved)

[3] using Mathematica™ with SetAxes

```
Data: bodydata.dat   Model: body.mcmc   Nov 8, 2022 at 2:30:41 PM

# chains x sample/chain: 175 x 1429 = 250075 (thinning = 10)

log(marginal likelihood): -1745.24

m1
MAP, Mean, Median, Mode, G-R stat: 78.1445 78.1489 78.1473 78.0305 1
Credible Intervals: 76.4312 76.8207 77.0334 79.2397 79.4552 79.9107

m2
MAP, Mean, Median, Mode, G-R stat: 177.745 177.747 177.746 177.8 1.001
Credible Intervals: 176.565 176.845 176.994 178.504 178.649 178.941

rho
MAP, Mean, Median, Mode, G-R stat: 0.533205 0.528988 0.530497 0.524484 1.025
Credible Intervals: 0.404884 0.438104 0.454052 0.604286 0.616632 0.639718

sig1
MAP, Mean, Median, Mode, G-R stat: 10.4644 10.5472 10.529 10.4596 1.002
Credible Intervals: 9.40211 9.6084 9.7682 11.3268 11.4731 11.8627

sig2
MAP, Mean, Median, Mode, G-R stat: 7.15051 7.20666 7.19477 7.15361 1.006
Credible Intervals: 6.4119 6.57106 6.67634 7.74094 7.84504 8.08568
```

Figure 1: A Typical Report

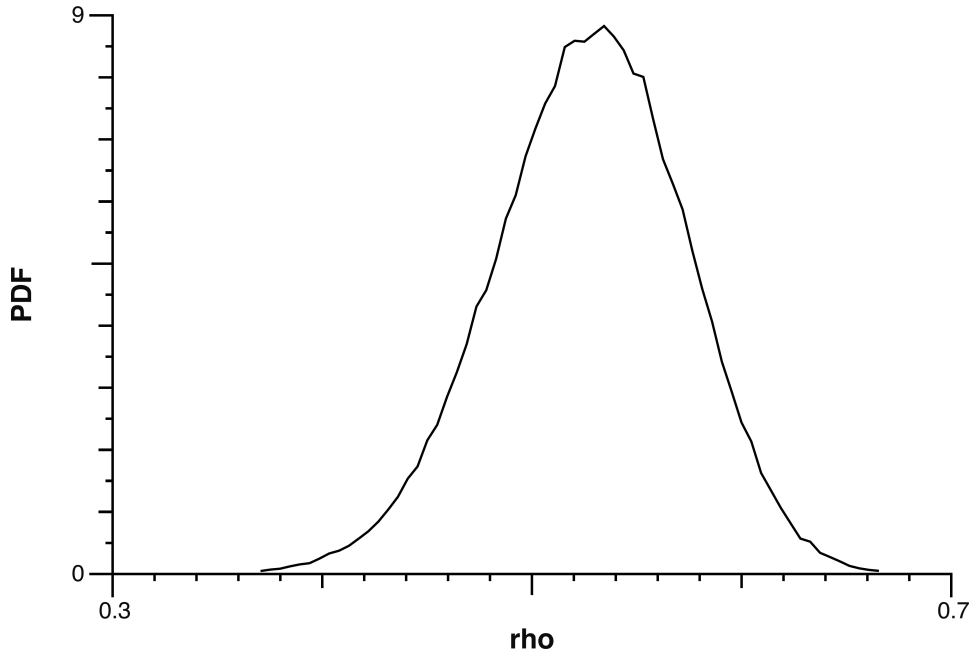


Figure 2: Marginal for Correlation Coefficient

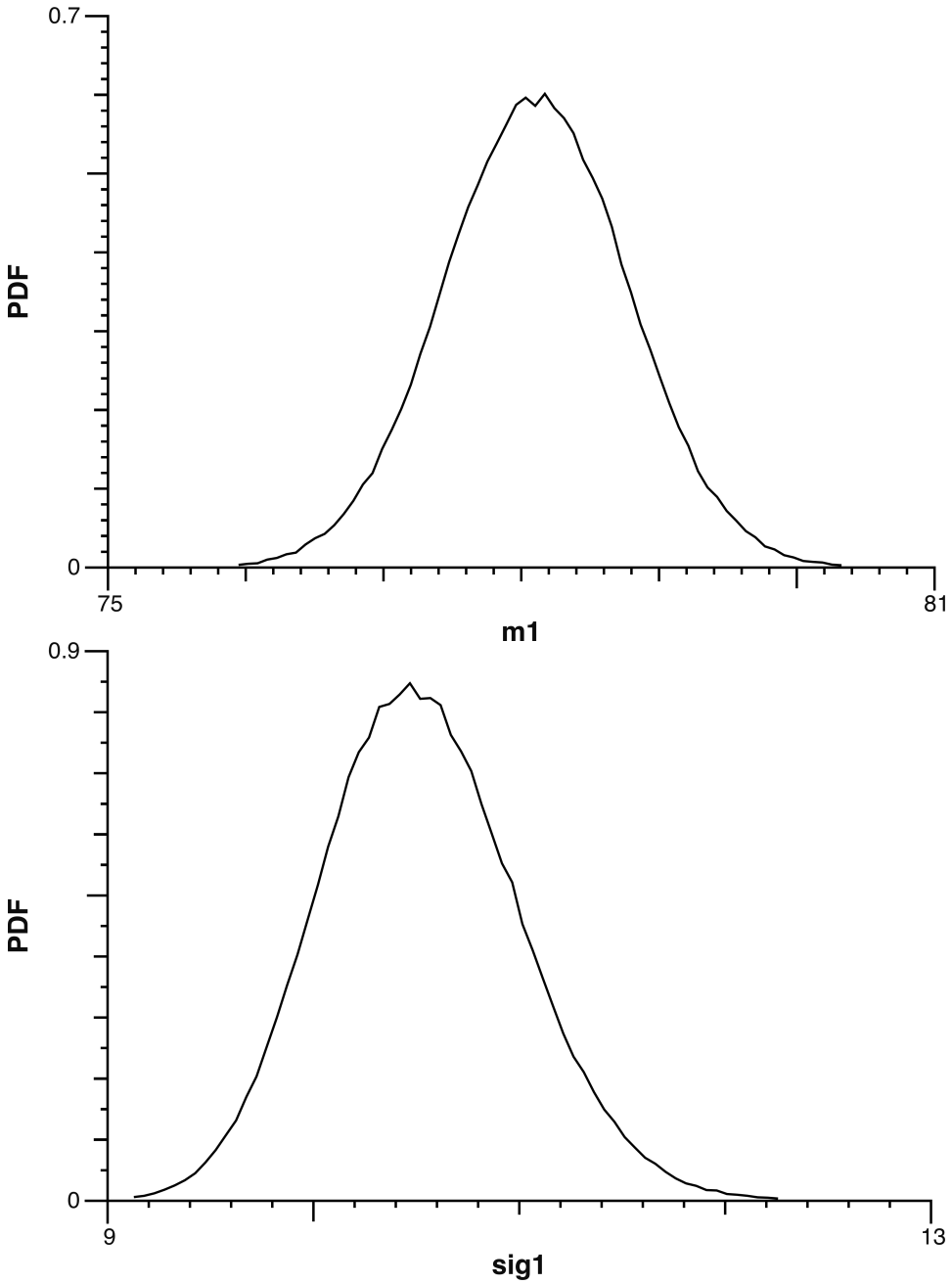


Figure 3: Marginals for Weight Dimension

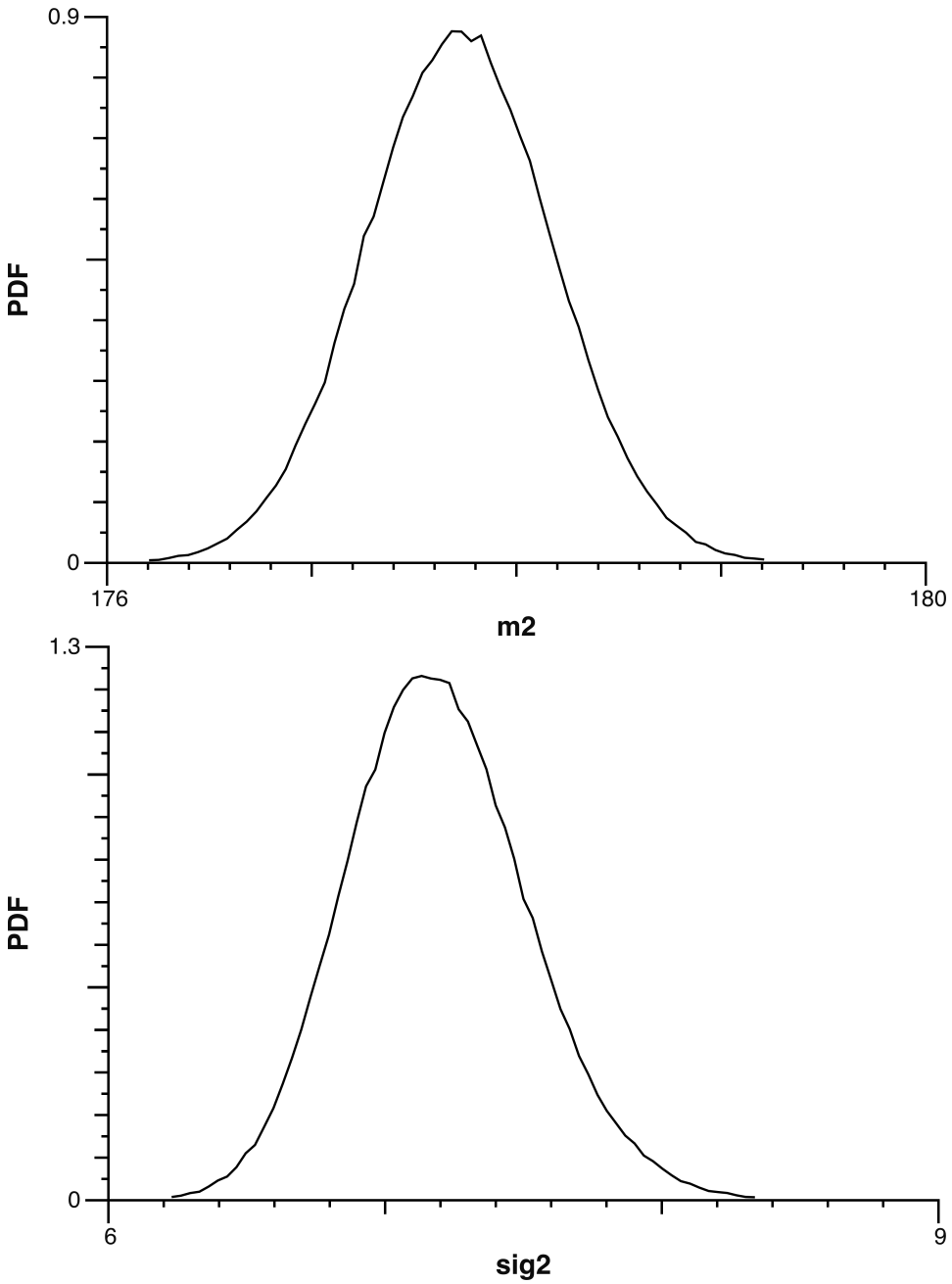


Figure 4: Marginals for Height Dimension

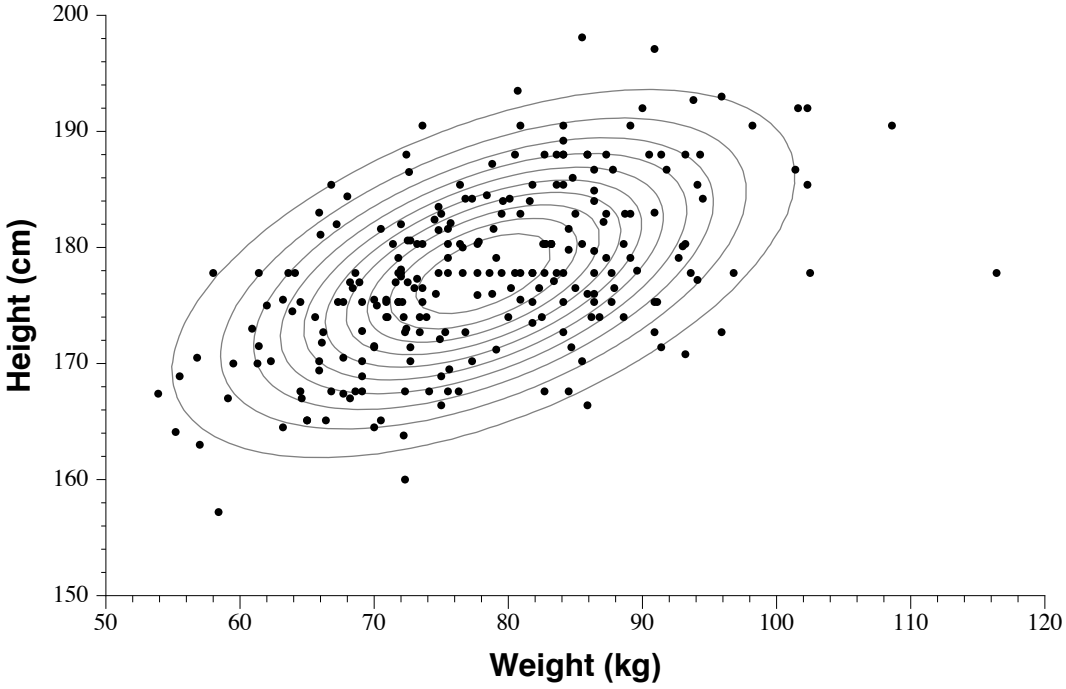


Figure 5: Goodness-of-fit for Body Model

2 Carbon-14

Specific activity of $^{14}\mathrm{C}$ with errors in independent as well as dependent variable. Here, the default burn-in setting was insufficient (did not reach equilibrium) so it was doubled. Runtime: 1 m 54 s

Model 2: Carbon-14 Specific Activity

```
Constants:
N = 18, // # of points
ln2 = log(2);
Data:
y[N], sigY[N], t[N], sigT[N];
Variables:
trueT[N], mu[N], A, halfLife, i;
Priors:
// prior info
A ~ Normal(12.5, 0.23);
halfLife ~ Normal(5720, 47);
// uncertain "ground truth"
for (i, 1:N) {
   trueT[i] ~ Jeffreys(100, 10000);
}
Likelihood:
for (i, 1:N) {
   t[i] ~ Normal(trueT[i], sigT[i]);
   mu[i] = A*exp(-(ln2/halfLife)*trueT[i]);
   y[i] ~ Normal(mu[i], sigY[i]);
}
Extras:
Monitored:
A, halfLife;
```

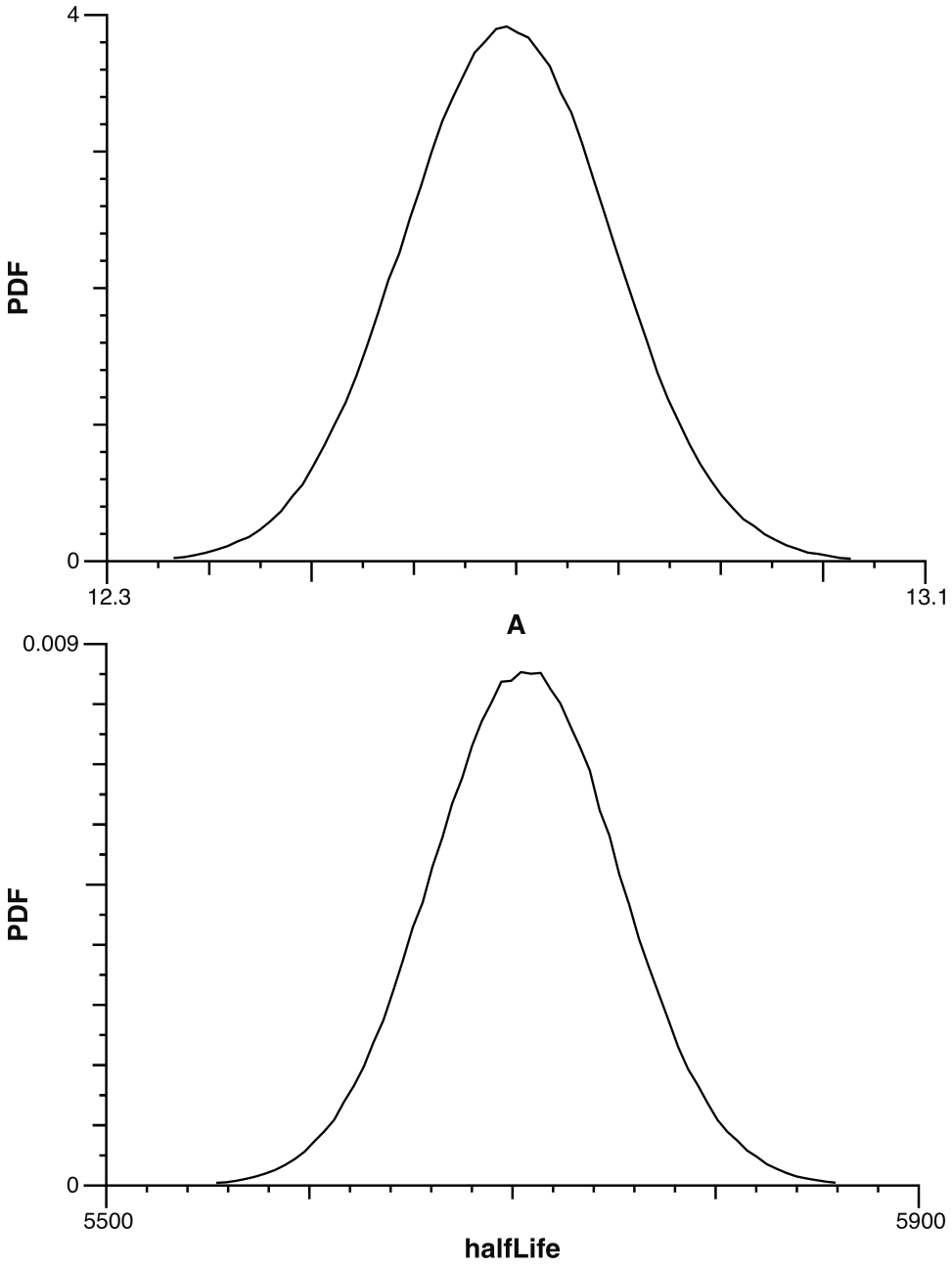


Figure 6: Marginals for $^{14}\mathrm{C}$ Activity Model Parameters

3 Daytime

Daytimes, rounded off to the nearest minute, over three years modeled as sinusoidal which is almost correct. Note that parameter sigma includes the known quantization variance in addition to the unknown model variance. Here, the (optional) goodness-of-fit plot is created. Also, as an Extra, a prediction is made for Daytime[1].
Runtime: 34 s

Model 3: Daytime for Boston, MA, USA

```
Constants:
N = 43,            // # of points
qVar = 1/12;  // quantization variance
Data:
day[N], daytime[N];
Variables:
mu[N], A, P, phi, D, procVar, sigma, pred1, i;
Priors:
A ~ Uniform(0, 720);
P ~ Normal(365.25, 1);
phi ~ Uniform(-Pi/2, 0);
D ~ Uniform(0, 1440);
procVar ~ Jeffreys(0.1, 50);
Likelihood:
sigma = sqrt(procVar + qVar);
for (i, 1:N) {
   mu[i] = A*sin(2*Pi*day[i]/P + phi) + D;
   daytime[i] ~ Normal(mu[i], sigma);
}
Extras:
pred1 = A*sin(2*Pi/P + phi) + D;
Monitored:
A, P, phi, D, sigma, pred1;
Goodness:
gdnsY = A*sin(2*Pi*gdnsX/P + phi) + D;
```

To create a goodness-of-fit plot, the dummy symbols gdnsX and gdnsY are used to enable the user to specify the variables to appear on the abscissa and the ordinate, resp. The default is to use mean parameter estimates for the model curve in this plot. In this case, the fit is so good that the 95-percent credible interval band is nearly invisible.

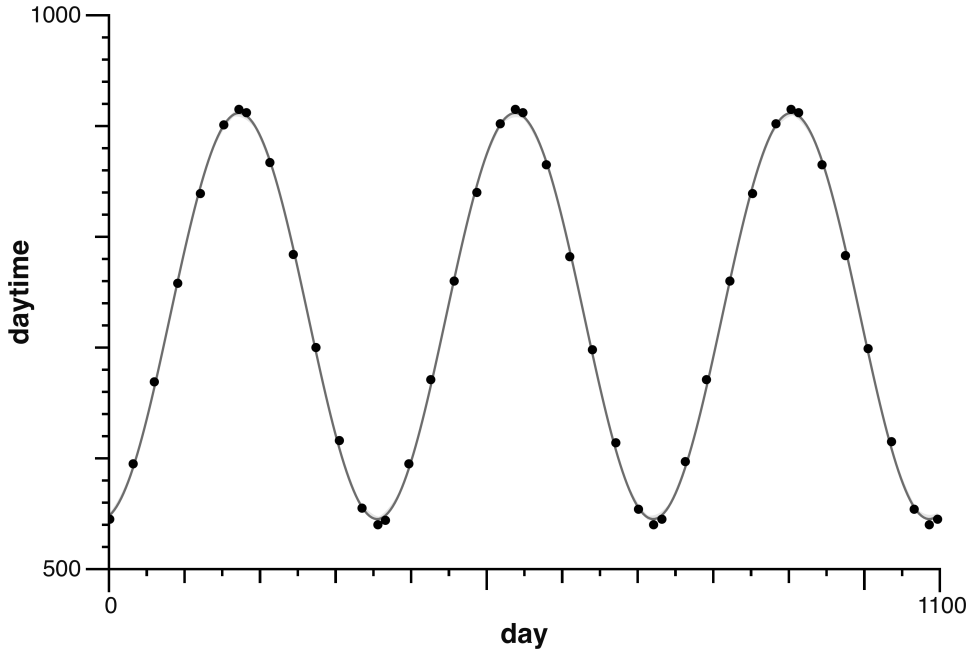


Figure 7: Goodness-of-fit for Sinusoidal Model

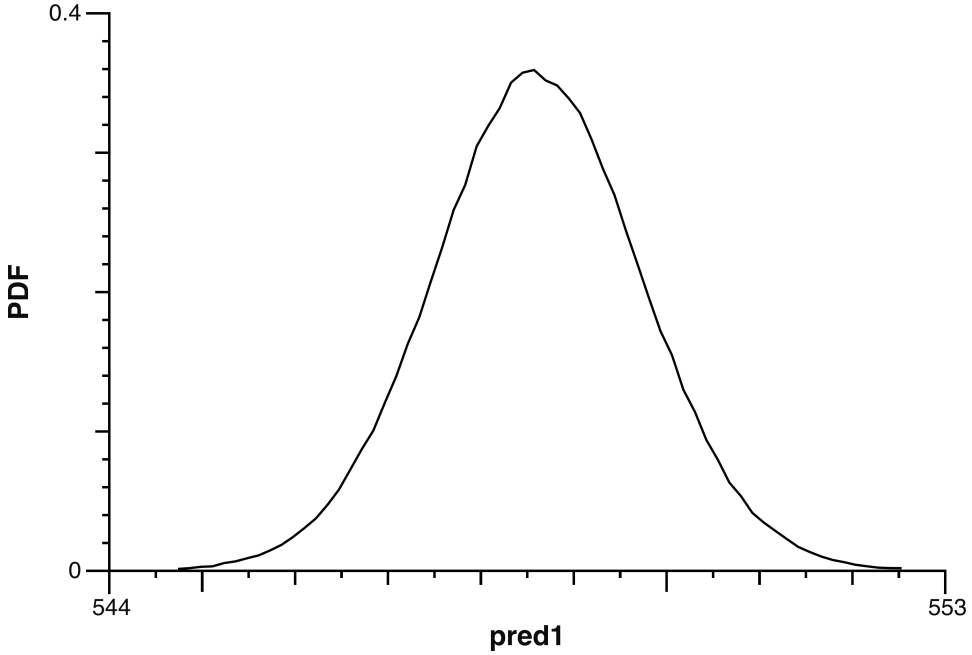


Figure 8: Posterior Prediction for Daytime[1]

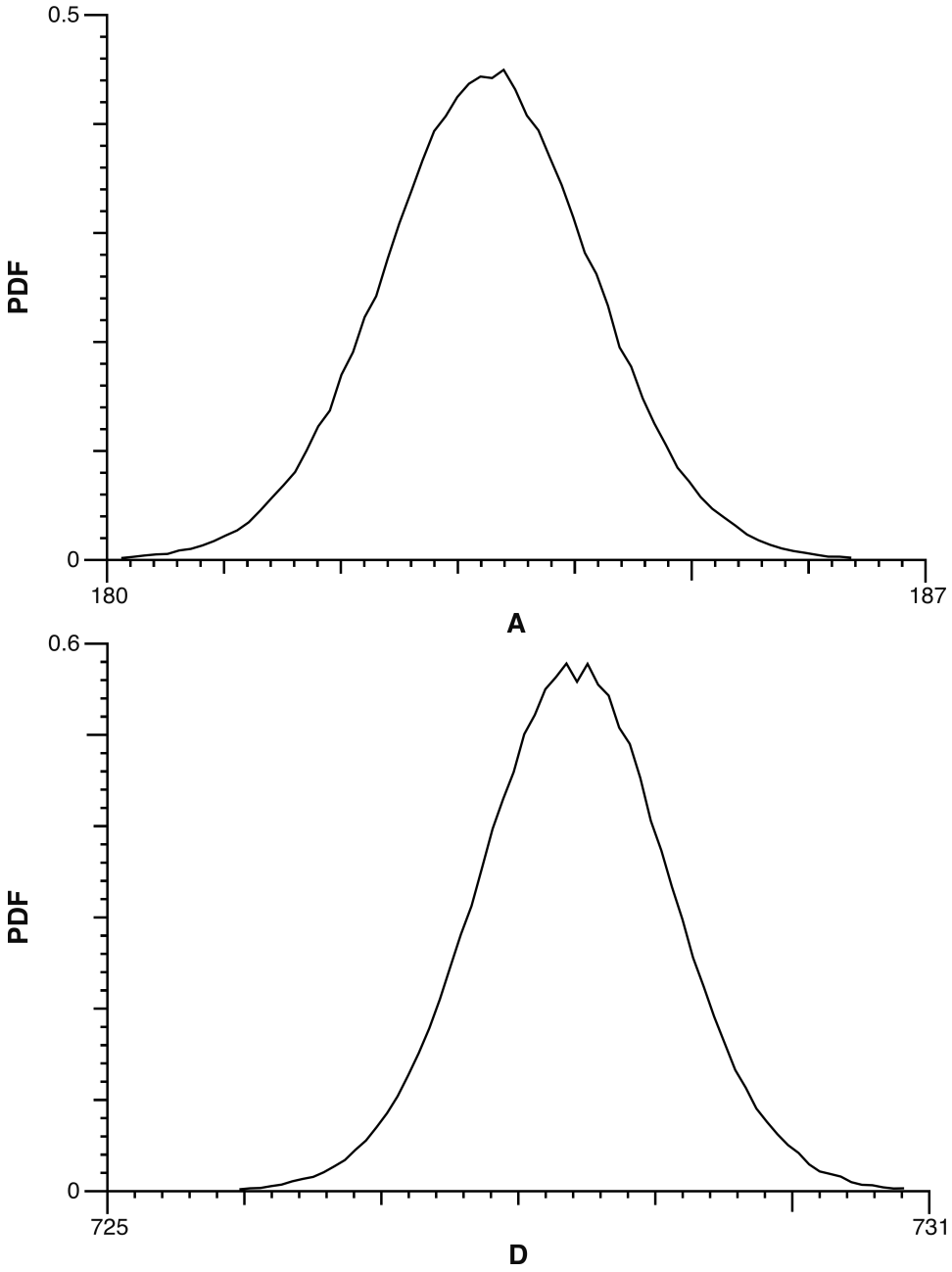


Figure 9: Marginals for Amplitude and Offset

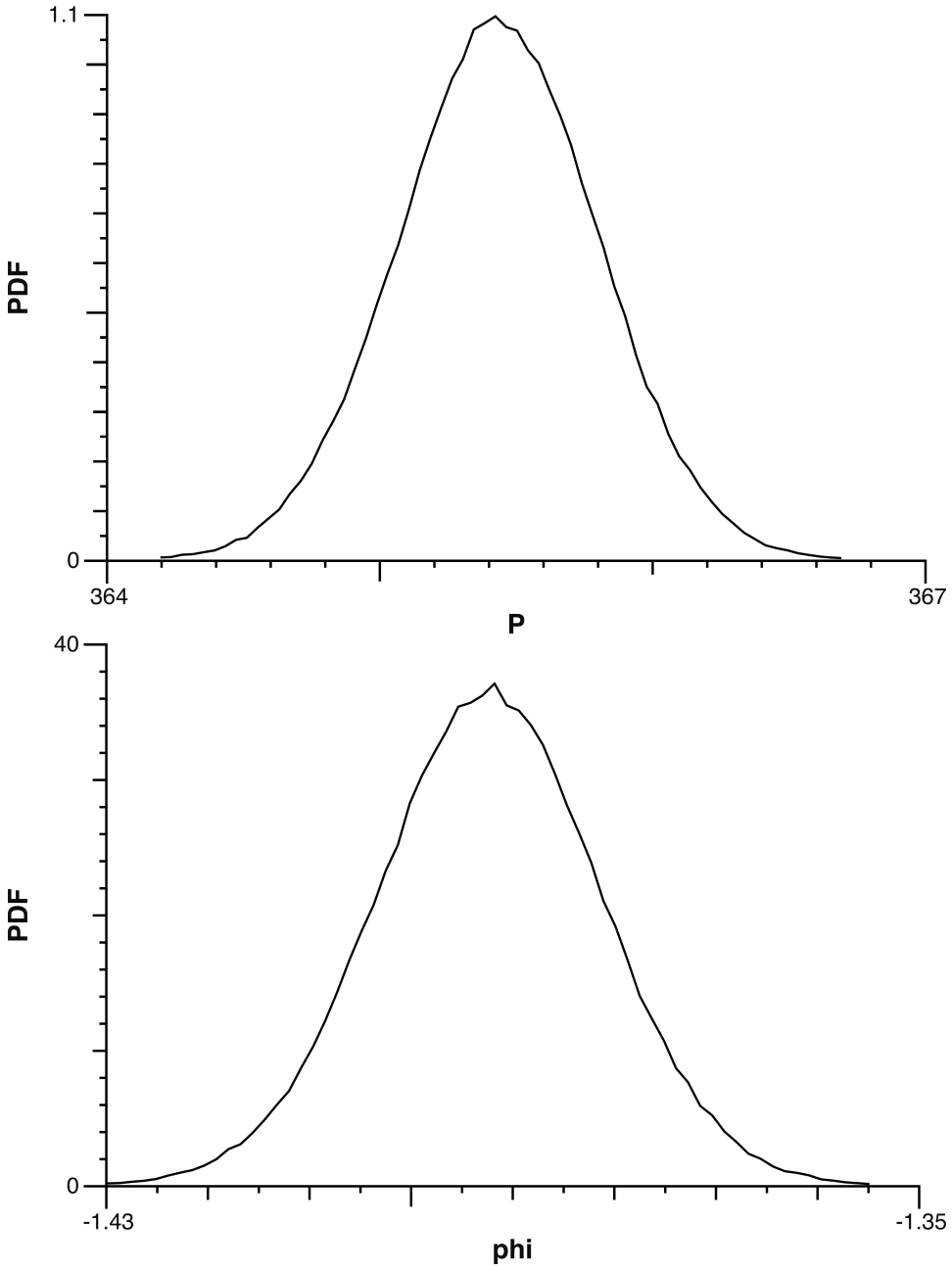


Figure 10: Marginals for Period and Phase

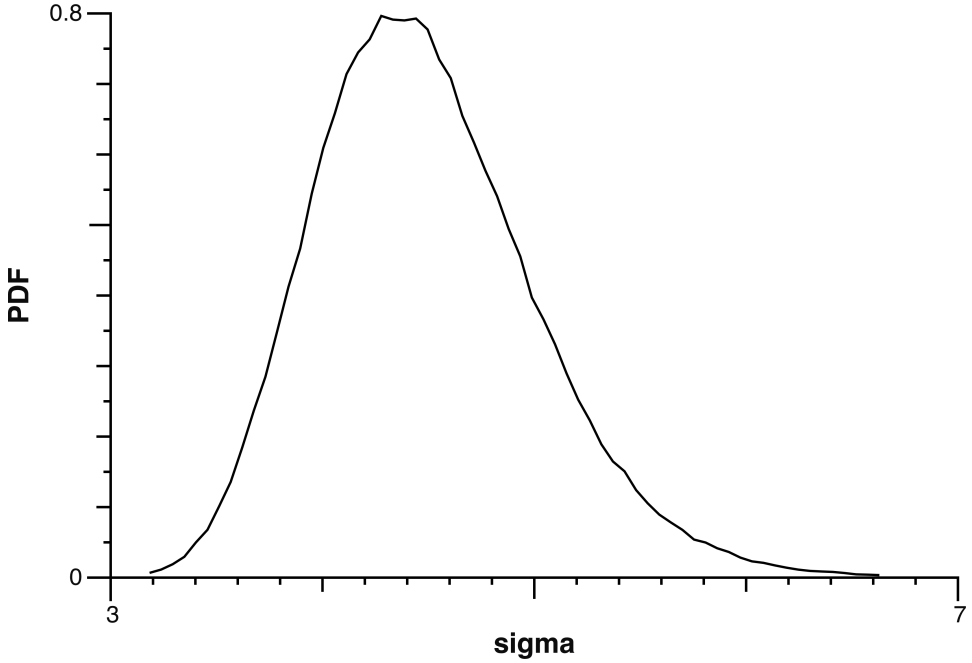


Figure 11: Marginal for Combined Sigma

4 Enzyme

Here are two models describing the activity of an enzyme as a function of its concentration. The first model includes measurement error only; the second adds model error. We shall use the marginal likelihoods of the two to do a model comparison so as to determine, in a robust fashion, which model is better (more credible). The R-squared metric is computed as an Extra.
Runtime: 7 s (model 1), 11 s (model2)

Model 4: Enzyme Activity (model 1)

```
// Simple model: measurement error only
Constants:
N = 14,   // # of points
TSS = 420.05;    // total_sum_squares
Data:
conc[N], rate[N], sigma[N];
Variables:
mu[N], A, B, err, ESS, RSq, i;
Priors:
A ~ Jeffreys(1, 100);
B ~ Jeffreys(1, 100);
Likelihood:
for (i, 1:N) {
   mu[i] = (A*conc[i])/(B + conc[i]);
   rate[i] ~ Normal(mu[i], sigma[i]);
}
Extras:
ESS = 0;
for (i, 1:N) { // weighted error
   err = (rate[i] - mu[i])/sigma[i];
   ESS = ESS + err*err;        // error_sum_squares
}
RSq = 1 - ESS/TSS;
Monitored:
A, B, RSq;
```

Model 2 is shown on the following page †

Model 5: Enzyme Activity (model 2)

```
// Measurement error plus (hypothetical) model error
Constants:
N = 14,   // # of points
TSS = 420.05;    // total_sum_squares
Data:
conc[N], rate[N], sigma[N];
Variables:
mu[N], A, B, sig, err, ESS, RSq, i;
Priors:
A ~ Jeffreys(1, 100);
B ~ Jeffreys(1, 100);
sig ~ HalfNormal(5);
Likelihood:
for (i, 1:N) { // with RMS error
   mu[i] = (A*conc[i])/(B + conc[i]);
   rate[i] ~ Normal(mu[i], sqrt(sigma[i]^2 + sig^2));
}
Extras:
ESS = 0;
for (i, 1:N) { // RMS-weighted error
   err = (rate[i] - mu[i])/sqrt(sigma[i]^2 + sig^2);
   ESS = ESS + err*err;        // error_sum_squares
}
RSq = 1 - ESS/TSS;
Monitored:
A, B, sig, RSq;
```

The figures in Table 3 show a comparison of the parameter values and also illustrate relabeling of plot axes. Table 4 shows some quantitative comparisons.

Table 3: Marginals for Enzyme Model Parameters

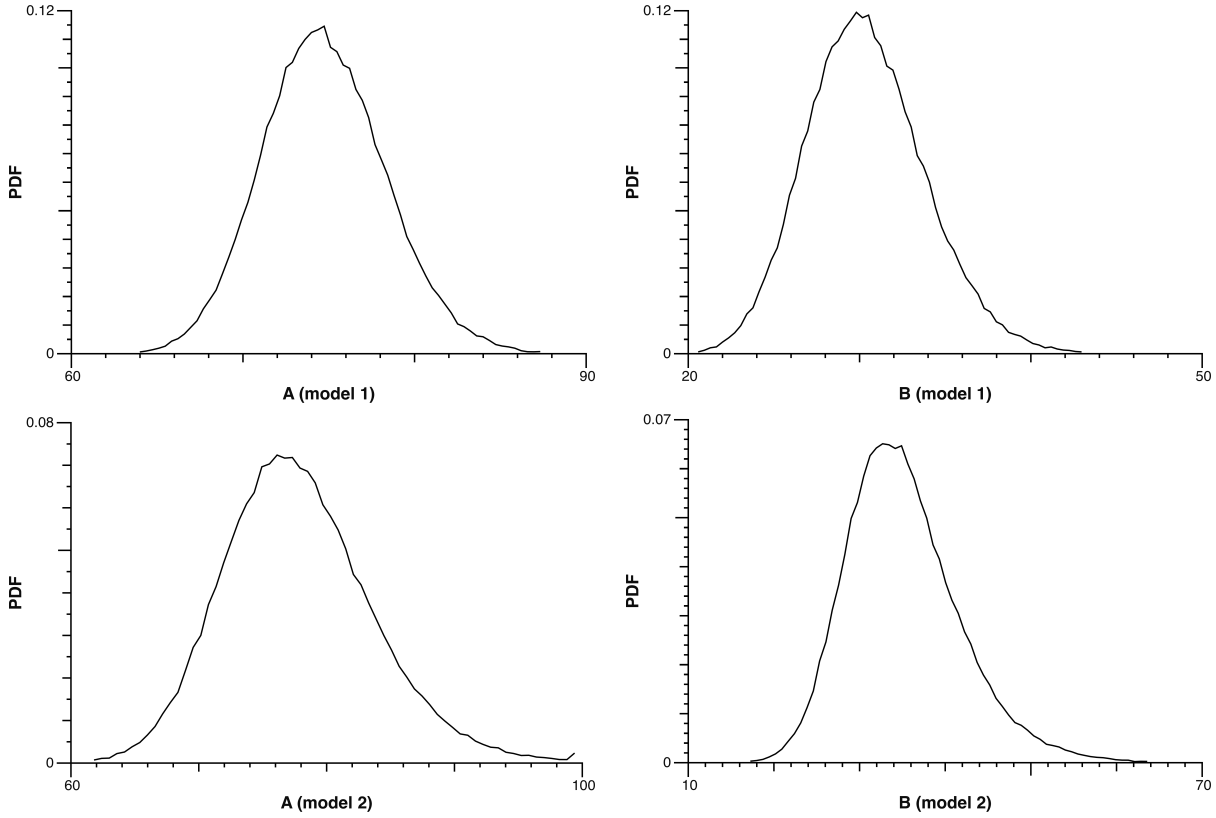


Table 4: Posterior Values for Enzyme Models

Model	Parameter	Estimate		95% Credible Interval	
		MAP	Mean	Lower Limit	Upper Limit
1	A	74.3387	74.9080	67.9552	82.1232
	B	29.8825	30.5384	23.8795	37.4809
	RSq	0.925168	0.920465	0.9109001	0.925228
2	A	76.4444	78.0646	66.9975	89.9513
	B	33.3419	35.5137	23.3249	49.0680
	RSq	0.970641	0.970092	0.949538	0.988485

Credible intervals for model 2 are wider and marginal plots broader since this model incorporates two sources of error. Nevertheless, as we show below, it is a much better model.

A picture is always nice (even if you have to create it offline).

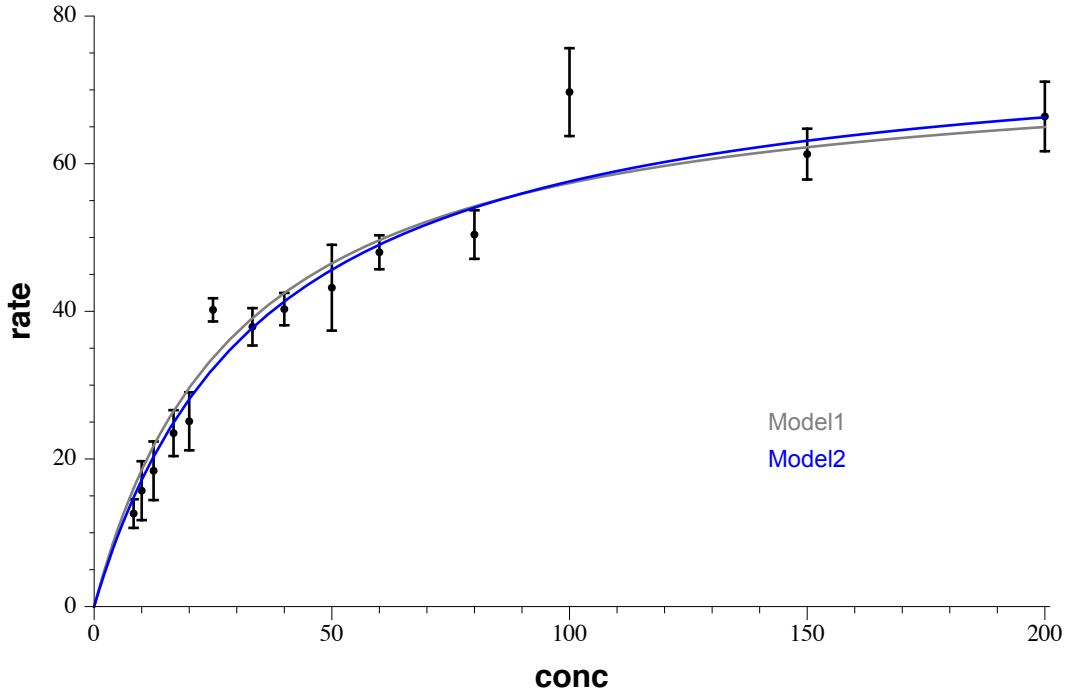


Figure 12: Another Comparison of the Two Models

In this run, the log(marginal likelihood) values for models 1 and 2 were -52.3626 and -47.2738, respectively.[4] The fact that model 2 has a larger marginal (global) likelihood indicates that it is preferred (more credible) in spite of the increased number of parameters.

We can (and should) be even more quantitative, as follows:

$$\text{odds model 2 better} \equiv \mathit{oddsM2} = \exp(-47.2738 - (-52.3626)) = 162.195$$

Hence,

$$\text{Prob(model 2 better)} = \frac{\mathit{oddsM2}}{1 + \mathit{oddsM2}} = 0.99387$$

[4]There should be very little variation from run to run.

5 FishCount

The data for this model came from a survey of 250 campers to see how many fish they caught. Most replied "zero", in some cases because they did not even go fishing! The data are modeled using a zero-inflated Poisson (ZIP) distribution. The latter is not a particularly good model here but does illustrate a mixture model for discrete count data. The weight parameter, wt, refers to the non-Poisson component of the mixture.

Since the zeroes are special, the data were sorted and the likelihood split accordingly using a Generic distribution for both parts.
Runtime: 50 s

Model 6: FishCount

```
Constants:
N = 250,         // # of points
Nzeroes = 142;
Data:
count_[N];
Variables:
lambda, wt, temp, i;
Priors:
lambda ~ Exponential(5);
wt ~ Uniform(0, 1);
Likelihood:
for (i, 1:Nzeroes) {
   temp = wt + (1 - wt)*exp(-lambda);
   count_[i] ~ Generic(log(temp), 0, 100);
}
for (i, Nzeroes+1:N) {
   temp = count_[i];
   count_[i] ~ Generic(log(1 - wt) + temp*log(lambda)
      - logFactorial(temp) - lambda, 0, 100);
}
Extras:
Monitored:
lambda, wt;
```

The mean weight was 0.567 which gives a count of 142 and suggests that all zeroes came from the non-Poisson component which seems unlikely. After all, some campers who did go fishing did not catch anything. The model could use some improvement.

Marginal plots are shown in Figure 13.

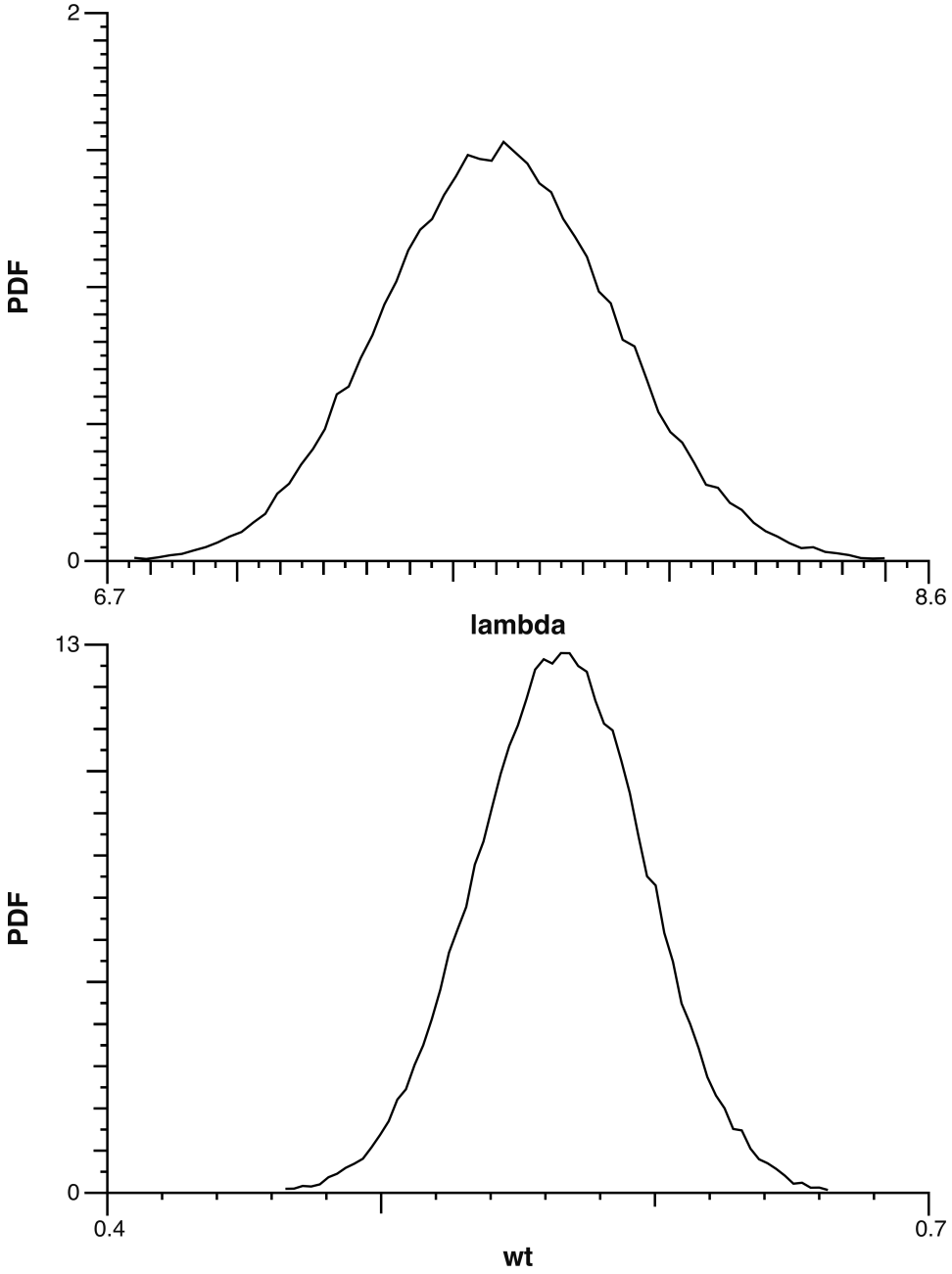


Figure 13: Marginals for FishCount

6 Gumbel

Another example of a distribution not included in the built-in repertoire of *MacMCMC*. The data are the season batting averages of the batting champions for US major-league baseball for the years (1876–2019). The Gumbel distribution is usually quite good at modeling record values of quantities that are normally distributed and seems a reasonable choice for this dataset.
Runtime: 33 s

Model 7: Gumbel Model for Best Batting Averages

```
Constants:
N = 144;  // # of points
Data:
x[N];
Variables:
a, b, z, i;
Priors:
a ~ Uniform(0, 1);
b ~ Uniform(0, 1);
Likelihood:   // Gumbel
for (i, 1:N) {
   z = (a - x[i])/b;
   x[i] ~ Generic(-log(b) + z - exp(z), 0, 1);
}
Extras:
Monitored:
a, b;
```

This is the simplest example of the Generic distribution implemented by *MacMCMC*. The first argument is the (natural) log of the probability density function (PDF). The two remaining arguments are taken as bounds for a uniform distribution from which values are selected to initialize the model. Were this a built-in distribution, *MacMCMC* would know how to select random variates for the initialization phase since there would be enough information coming ultimately from founder nodes.

Figure 14 shows the two marginals and Figure 15 shows a goodness-of-fit plot (created offline). *MacMCMC* cannot create such a plot from a Generic distribution.

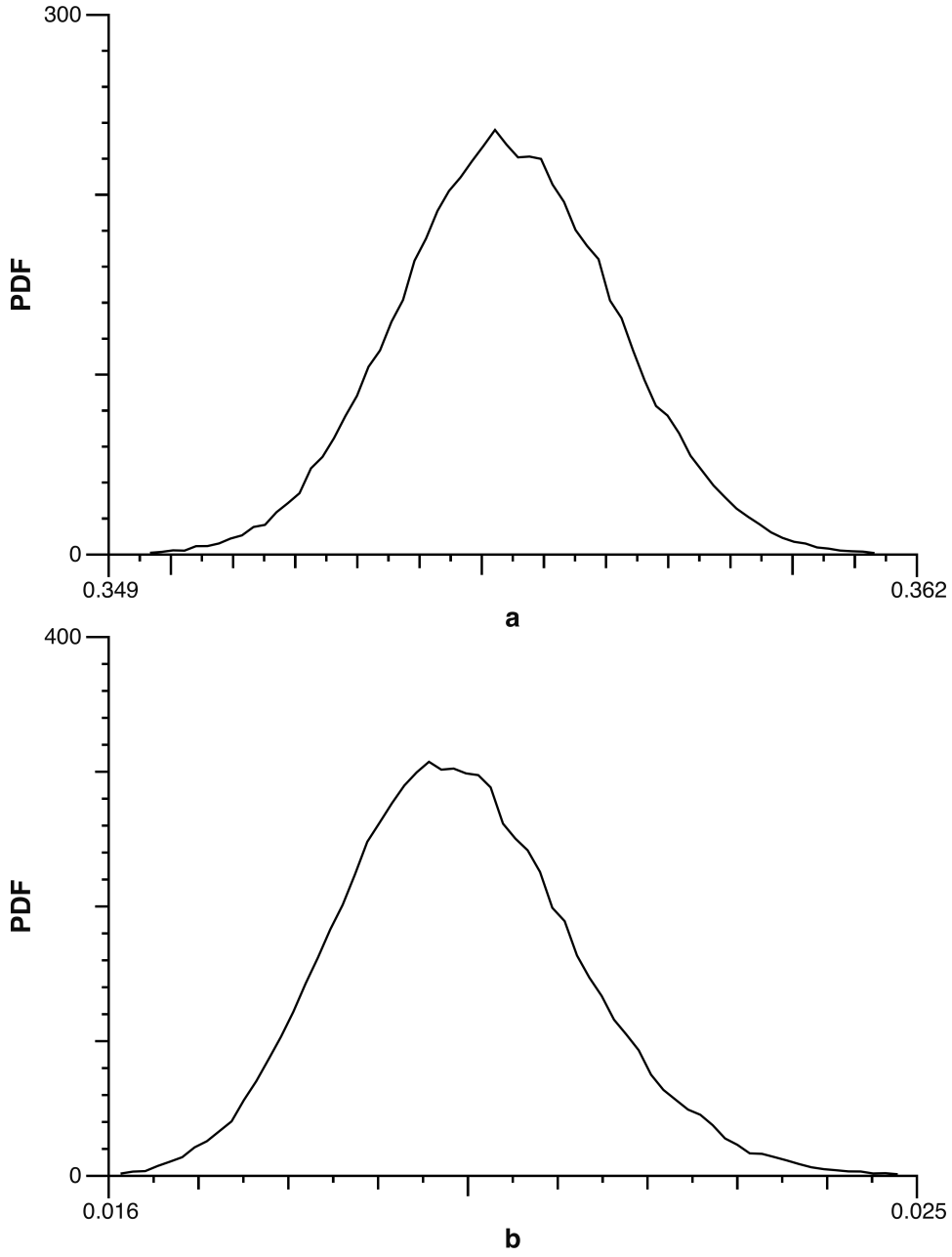


Figure 14: Marginals for the Gumbel Parameters

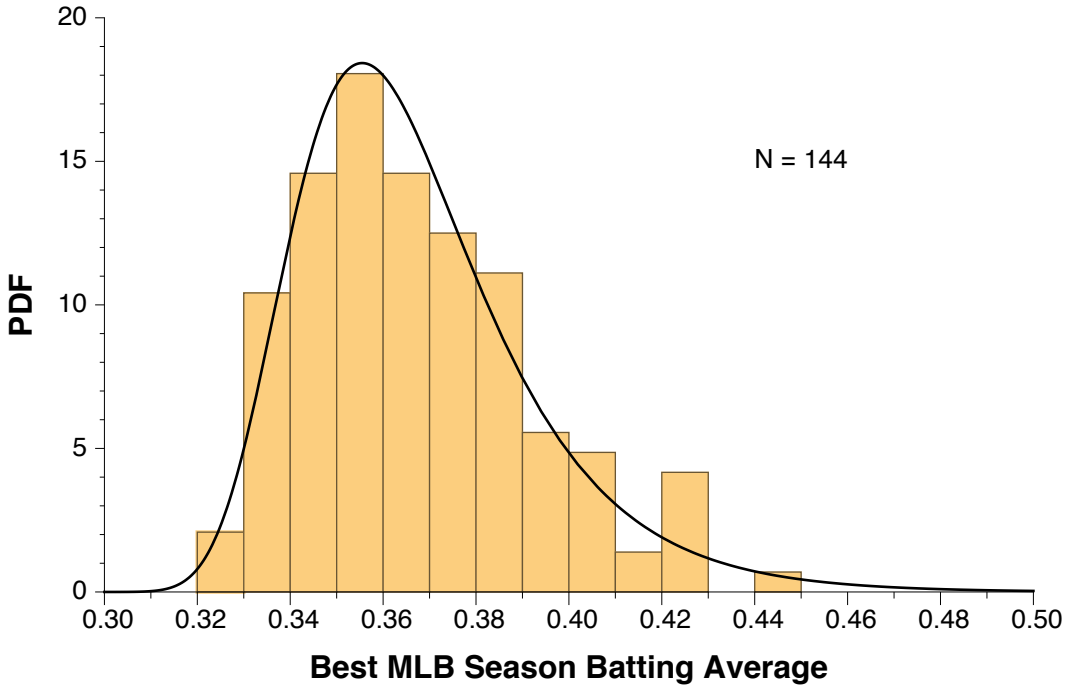


Figure 15: Goodness-of-fit for Gumbel Model

7 Hale-Bopp

This example illustrates an analysis which, in a frequentist context, would be treated as nonlinear, weighted least-squares. The data come from observations of the release of cyanide radical (CN) from the comet Hale-Bopp as it approached the sun in 1995. There are only seven points and a large amount of measurement error. The goodness-of-fit plot (Fig. 18) shows a very wide 95-percent credible interval band. For comparison, the (weighted) R-squared metric is computed as an Extra.
Runtime: 3 s

Model 8: Hale-Bopp

```
Constants:
N = 7,   // # of points
TSS = 628.66;    // total_sum_squares
Data:
rate[N], distance[N], sig[N];
Variables:
mu[N], A, B, err, ESS, RSq, i;
Priors:
A ~ Jeffreys(100, 10000);
B ~ Jeffreys(0.1, 10);
Likelihood:
for (i, 1:N) {
   mu[i] = A*exp(-B*distance[i]);
   rate[i] ~ Normal(mu[i], sig[i]);
}
Extras:
ESS = 0;
for (i, 1:N) {
   err = (rate[i] - mu[i])/sig[i]; // weighted error
   ESS = ESS + err*err;        // error_sum_squares
}
RSq = 1 - ESS/TSS;
Monitored:
A, B, RSq;
Goodness:
gdnsY = A*exp(-B*gdnsX);
```

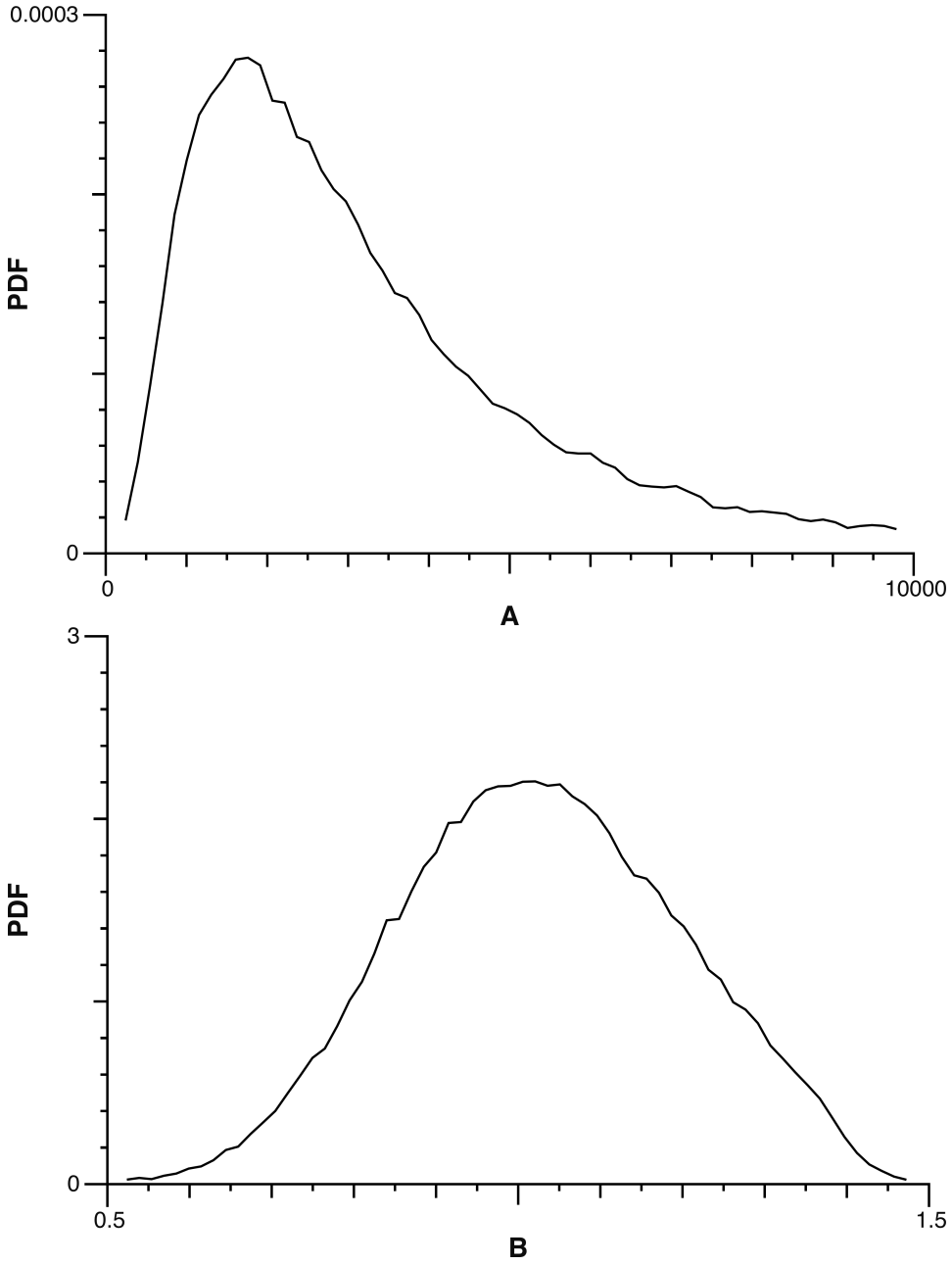


Figure 16: Marginals for the Hale-Bopp Model Parameters

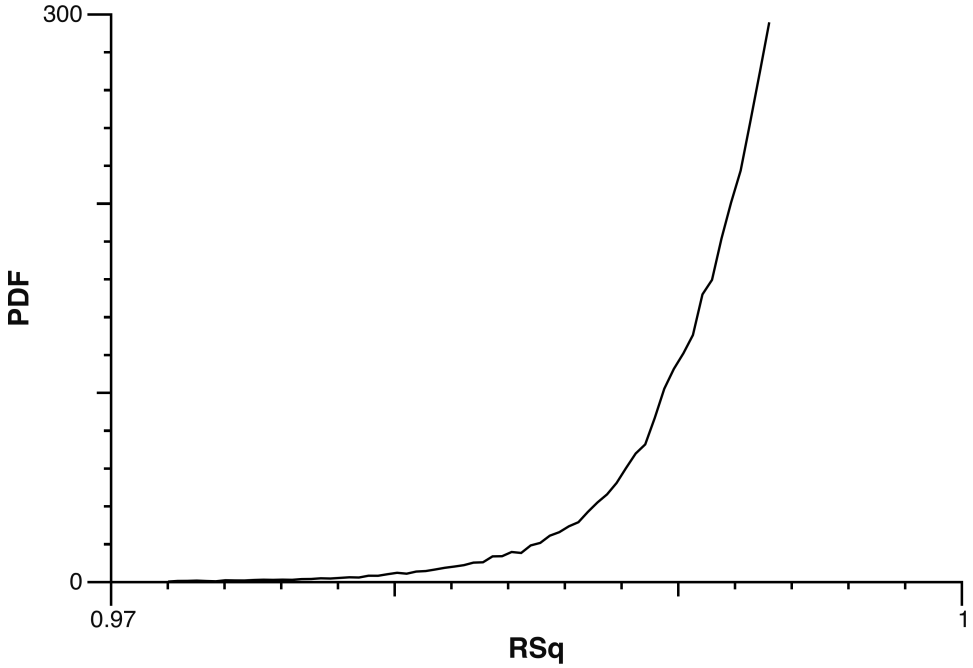


Figure 17: Marginal for R-squared

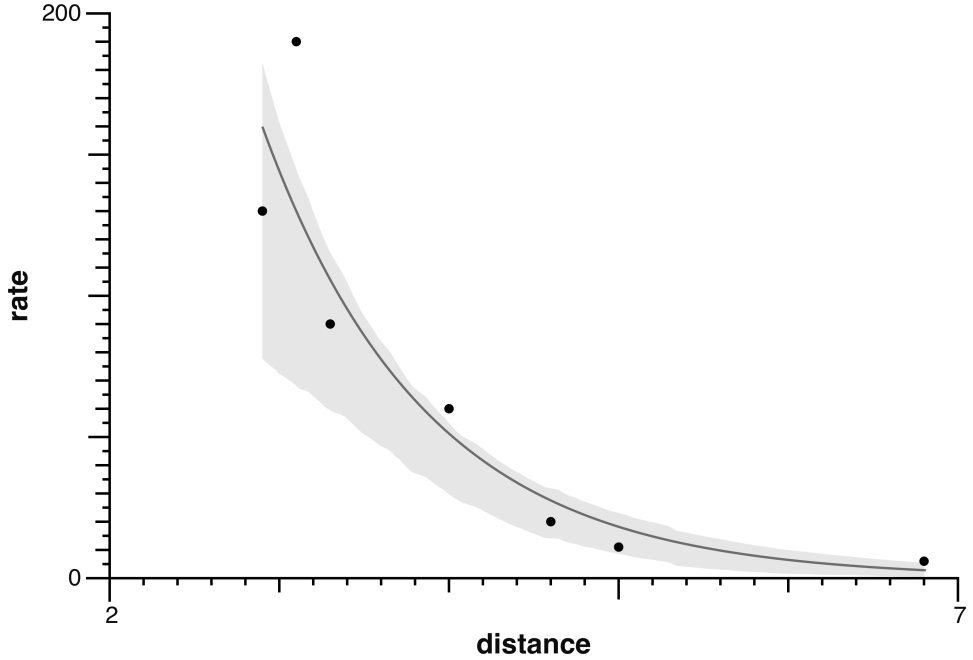


Figure 18: Hale-Bopp Goodness-of-fit (with 95% credible interval)

8 Hyphens

The data for this example are the number of line-ending hyphens on each page of a book that I was reading for all pages full of text. The model is Poisson which was just a guess. The goodness-of-fit plot shows how good a guess it was. Predictions are made for all observed values of these counts (0–8).[5]

The data and predictions are here forced to be integers by affixing an underscore to the symbol. This affects the Report and all plots for such variables.
Runtime: 26 s

Model 9: Hyphens

```
Constants:
N = 378,   // # of points
logN = log(N);
Data:
hpp_[N];
Variables:
lambda, pred_[9], logPoisson, i;
Priors:
lambda ~ Gamma(3, 3);
Likelihood:
for (i, 1:N) {
   hpp_[i] ~ Poisson(lambda);
}
Extras:
// Compute in log space to avoid possible overflow.
for (i, 0:N) {
   logPoisson = logN + i*log(lambda) - lambda - logFactorial(i);
   pred_[i] = rnd(exp(logPoisson)); // round
}
Monitored:
lambda, pred_[0], pred_[];
Goodness:
gdnsX_ ~ Poisson(lambda);
```

Figures 19 and 20 show the marginal for lambda as well as the goodness-of-fit plot (plus credible interval). Figure 21 shows the marginal for predictions of pred_[0] and the mode, pred_[2].

[5]The prediction for pred_[0] is computed separately because the pred_[] syntax starts with one.

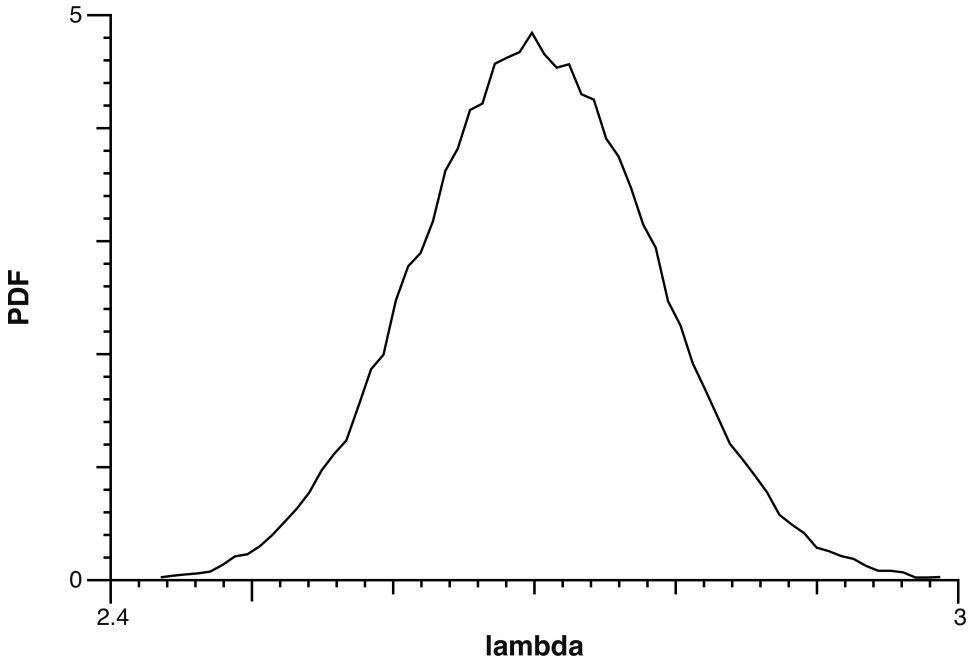


Figure 19: Marginal for lambda

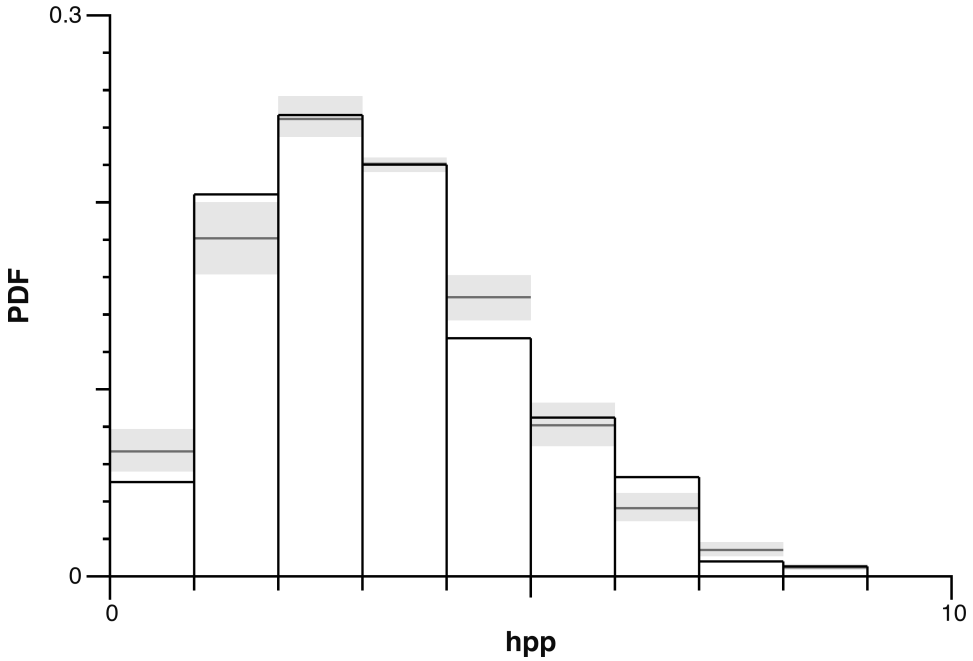


Figure 20: Goodness-of-fit for Hyphens Model

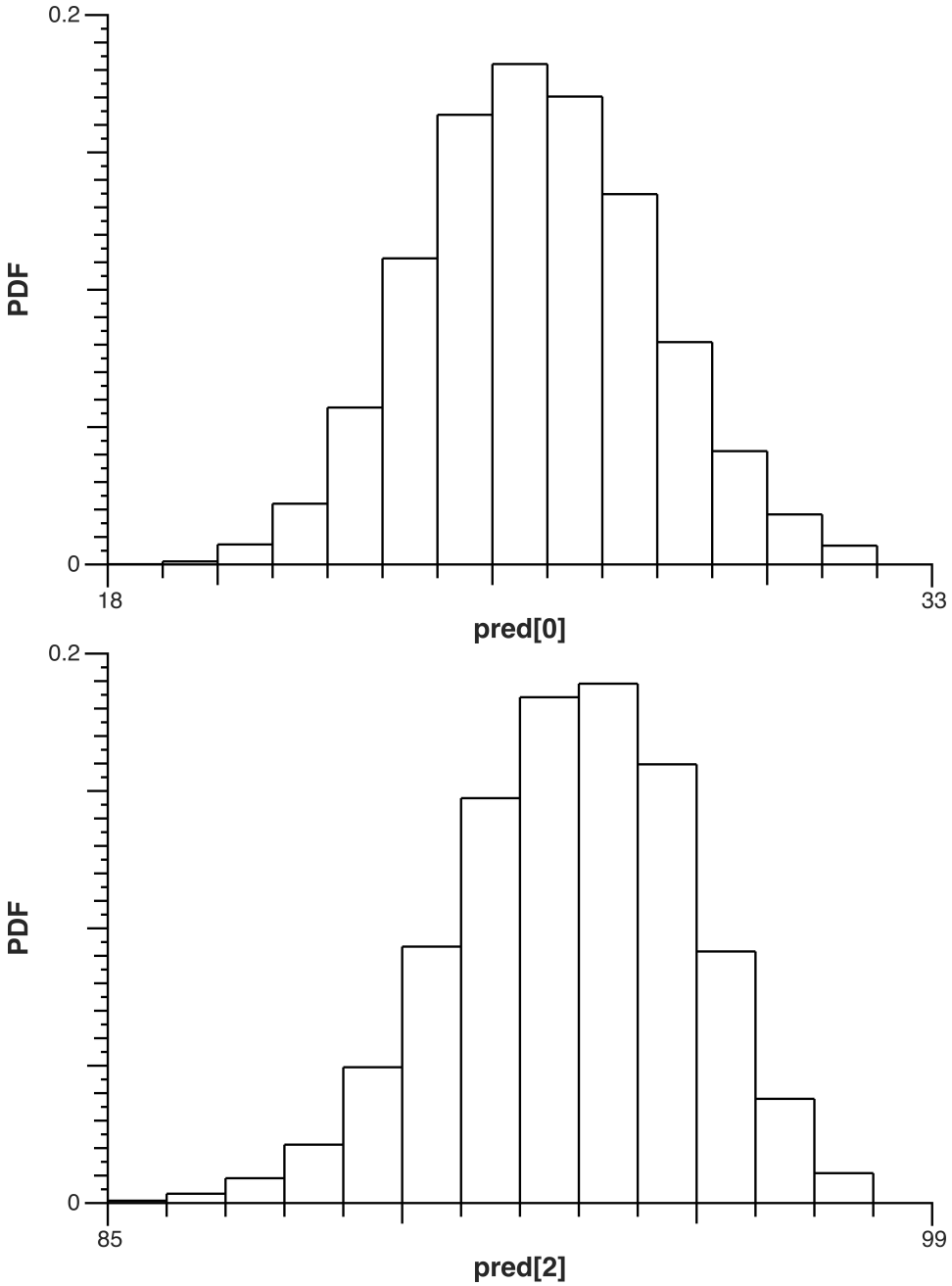


Figure 21: Posterior Predictions for Zero Count and Mode (two count)

9 Iris

This example uses the well-known Iris dataset of Fischer describing 150 samples of three species of iris (50 of each). This is a classification problem with a known answer.

For a *MacMCMC* example, two principal components were computed from the data accounting for nearly 98 percent of the observed variation.[6] Therefore, prior information suggests that a bivariate mixture with three components should be a very good model. Here, we use the built-in BivariateNormalMixture distribution.

The BivariateNormal distribution has five parameters: two means, two sigmas and one correlation coefficient (see pg. 4). Therefore, a mixture of these with three components has 17 parameters, including two (assumed) unknown weights. With 17 parameters and only 150 points, one should not expect wonderful results. Typically, 30 points per parameter are desirable and mixtures demand even more information, for the weights especially. Fortunately, "ground truth" is known so a goodness-of-fit plot should be revealing.

In this analysis, there is a special problem known as *label switching* which occurs when the identity (label) of two components are occasionally interchanged. This can easily happen because a logPosterior value for a mixture does not depend on the labeling given to the components. The problem is particularly severe when the mixture is homogeneous (all components of the same analytical form). Also, it is not peculiar to MCMC; it happens with maximum-likelihood analyses as well. Algorithms to undo this switching are an ongoing subject of research. Currently, there is no perfect solution. In fact, there *is* no perfect solution in general since nothing prevents datapoints in different components from overlapping. In that case, component identity is not even well-defined.

MacMCMC implements mixture relabeling for its built-in mixtures (see *User Guide*). When the output is relabeled, there will be two trace files, original and relabeled. There will also be an additional output file with an entry for each datum giving the probability that it "belongs" to each of the components.

The model is shown on the following page. In this mixture model, the weights are computed "manually" using an auxiliary variable, theta (see lines 24–27). Note, however, that weights may not be monitored since relabeling is going to change them.

Figure 22 shows a contour plot of the data as well as the (mean) solution. The color coding reflects the known answer. It appears that this model does a good job of classifying the three species. There is little overlap.

The marginals are collected in Figure 23. This model makes the sigmas all the same. There are not enough points to do anything more elaborate.
Runtime: 7 m 8 s (including relabeling)

[6]The ebook *Data, Uncertainty and Inference* referenced on this webpage provides more details.

Model 10: Iris

```
Constants:
N = 150,         // # of points (rows)
nc = 3,         // # of components
sq = sqrt(N/nc);
Data:
pc1pc2[N][2];
Variables:
m1[nc], m2[nc], sig1[nc], sig2[nc], rho[nc], wt[nc],
   theta, s, mum1, mum2, mum3, row, i;
Priors:
s ~ Gamma(3, 0.5);
mum1 ~ Normal(-8, s/sq);
mum2 ~ Normal(-6, s/sq);
mum3 ~ Normal(-3, s/sq);
m1[1] ~ Normal(mum1, s);
m1[2] ~ Normal(mum2, s);
m1[3] ~ Normal(mum3, s);
for (i, 1:nc) {
   m2[i] ~ Normal(5.5, s/sq);
   sig1[i] = s;
   sig2[i] = s;
   rho[i] ~ Uniform(-1, 1);
}
theta ~ Uniform(0, 1);
wt[1] ~ Uniform(0, 0.5);
wt[2] = theta*(1 - wt[1]);
wt[3] = 1 - wt[1] - wt[2];
Likelihood:
for (row, 1:N) {
   pc1pc2[row][] ~ BivariateNormalMixture(m1[], m2[],
      sig1[], sig2[], rho[], wt[], nc);
}
Extras:
Monitored:
m1[], m2[], sig1[], sig2[], rho[];
```

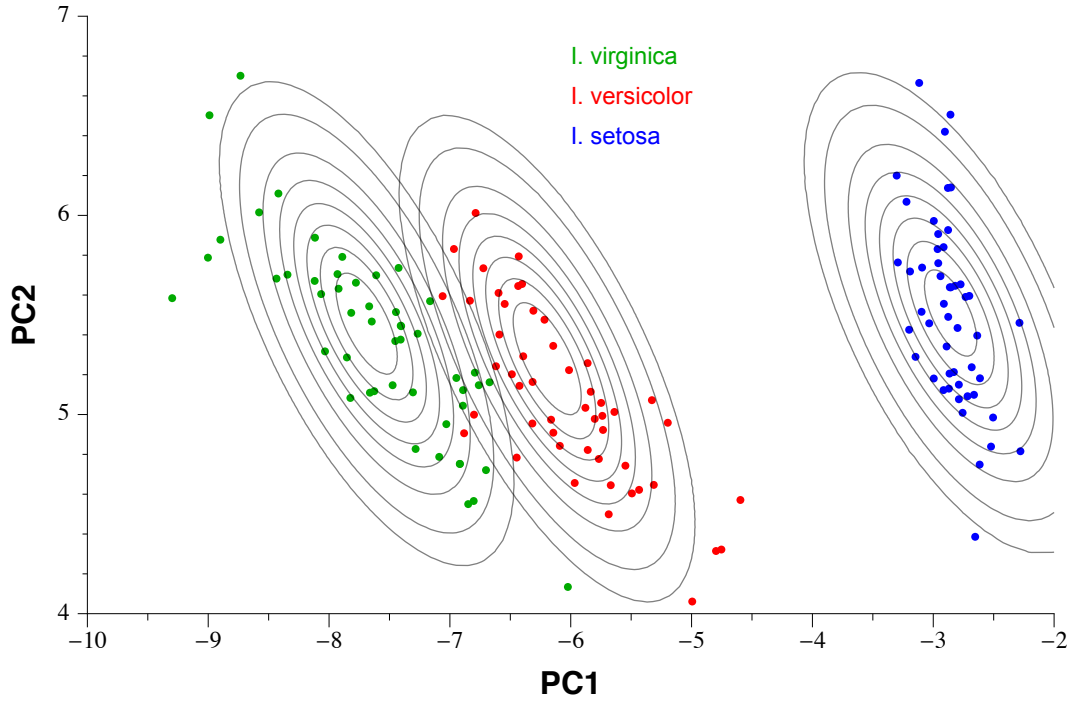


Figure 22: Iris Posterior Contours

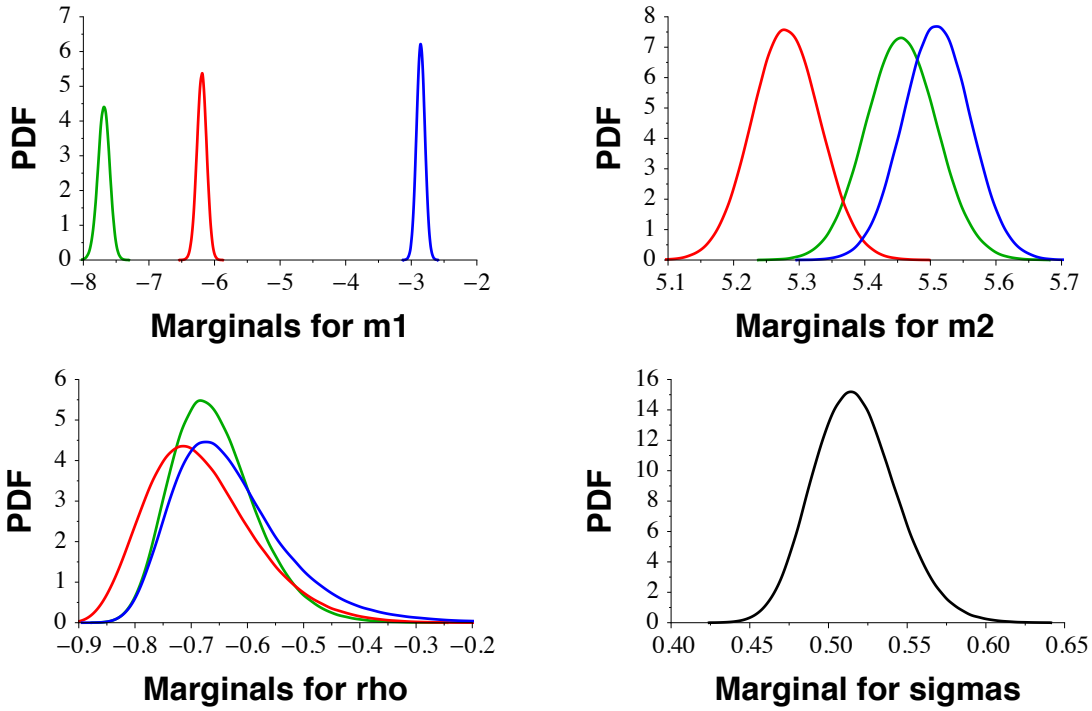


Figure 23: Collected Marginals

10 Lizards

The dataset for this example are capture-recapture counts of horned lizards in a fixed area. The purpose of the experiment, repeated for a total of 14 occasions, was to estimate the population of lizards. There is a large literature on this subject.

The likelihood suggested by analyses of this sort would typically be multinomial, with a probability for each value of n—the number observed in each category. Sadly, it cannot, in this case, be a simple multinomial since the n[0] category here represents lizards never captured for which data are obviously missing!

The model has to be more complicated so we compute some parts "by hand".
Runtime: 6 s

Model 11: Lizards

```
Constants:
J = 14, J1 = J+1, Nobs = 68;
Data:   // augmented for unobserved
n_[J1];
Variables:
N_, p, logLikTerm[J1], logp, log1mp, k;
Priors:
N_ ~ DiscreteUniform(68, 150);
p ~ Uniform(0, 1);
Likelihood:
// Multinomial EXCEPT that there is an unobserved category!
// logLikelihood computation
logp = log(p);
log1mp = log(1-p);
for (k, 1:J) { // binomial bin probability
   logLikTerm[k] = n_[k]*(k*logp + (J-k)*log1mp);
}
// remaining terms
logLikTerm[J1] = logFactorial(N_) - logFactorial(N_ - Nobs)
                 + J*(N_ - Nobs)*log1mp;
// likelihood statement
for (k, 1:J1) {
   n_[k] ~ Generic(logLikTerm[k], 0, 100);
}
Extras:
Monitored:
N_, p;
```

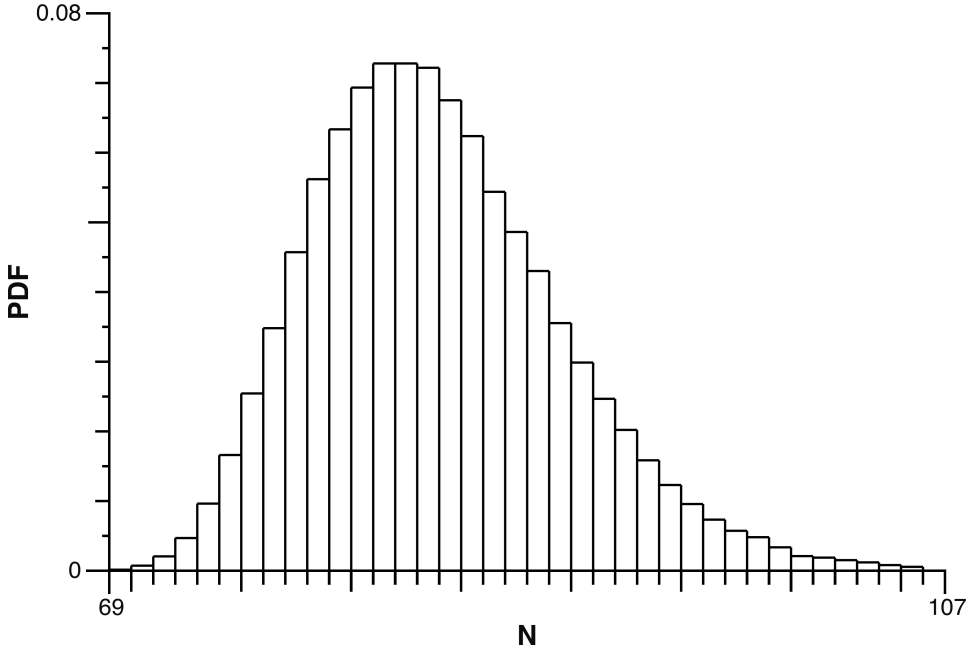


Figure 24: Marginal for Lizard Population

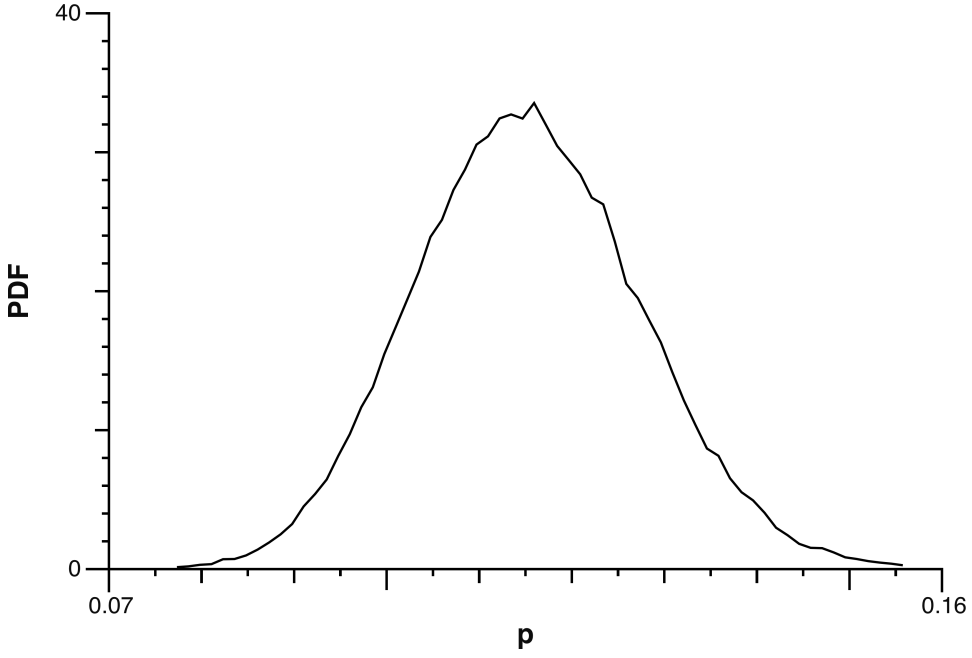


Figure 25: Marginal for Capture Probability

Mean estimates are $N\_ = 82$ and $p = 0.116$. Since there were 68 lizards observed, there were an estimated 14 that went unobserved even after 14 tries. Clearly, horned lizards are hard to capture.

11 MP

Melting point of *n*-octadecane. Measurements with varying, known measurement error. This is a typical metrology analysis where the unknown is a value that *should* be a constant. Runtime: 3 s

Model 12: Melting Point of *n*-Octadecane

```
Constants:
N = 35;  // # of points
Data:
MP[N], sig.meas[N];
Variables:
trueMP, i;
Priors:
trueMP ~ Uniform(295, 305);
Likelihood:
for (i, 1:N) {
   MP[i] ~ Normal(trueMP, sig.meas[i]);
}
Extras:
Monitored:
trueMP;
```

The marginal for trueMP is presented in Figure 26.

When a raw marginal plot is considered "too rough" for show-and-tell, it may be smoothed as shown in Figure 27. *MacMCMC* performs smoothing in Fourier space.

Note that the abscissas on these two marginal plots were *not* edited although that is a built-in capability. *MacMCMC* will automatically insert an offset onto an axis whenever failure to do so would result in tick labels that are too long. Very often, an analyst would do this sort of encoding ahead of time so as to increase the precision of the analysis by eliminating unnecessary (redundant) digits in the data. In *MacMCMC* , all variables are double-precision internally (roughly, 16 decimal digits of precision) so no harm done!

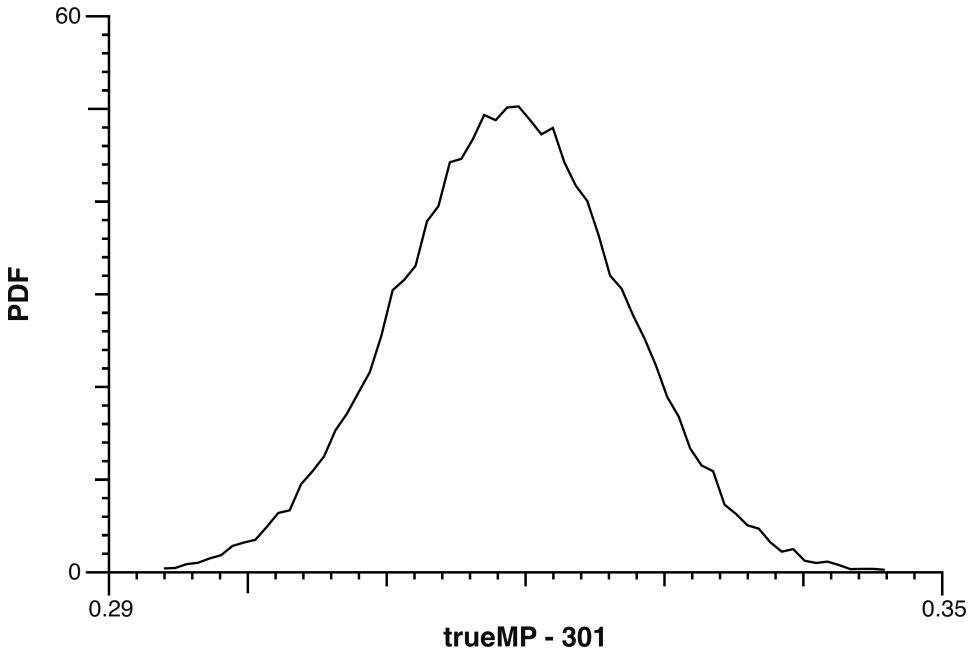


Figure 26: Marginal for trueMP

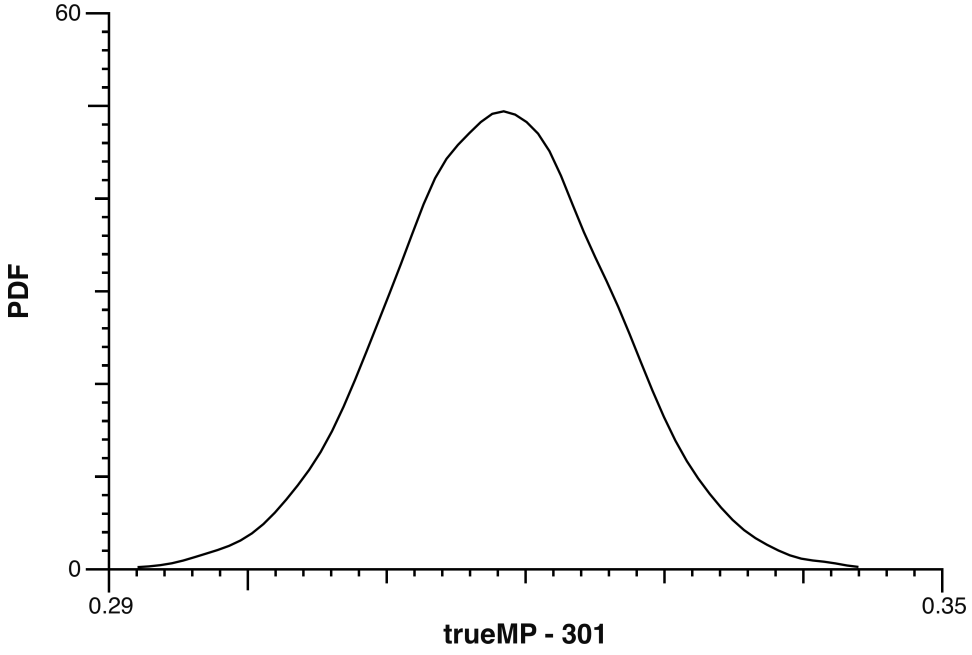


Figure 27: Smoothed Marginal for trueMP

12 NormalTemp

This is simply an analysis of a set of 65 temperatures of adult males. These are modeled as Normal (Gaussian) with both parameters unknown. We include this trivial example because it shows a goodness-of-fit plot for a continuous distribution. *MacMCMC* uses a standard q-q plot for this purpose.

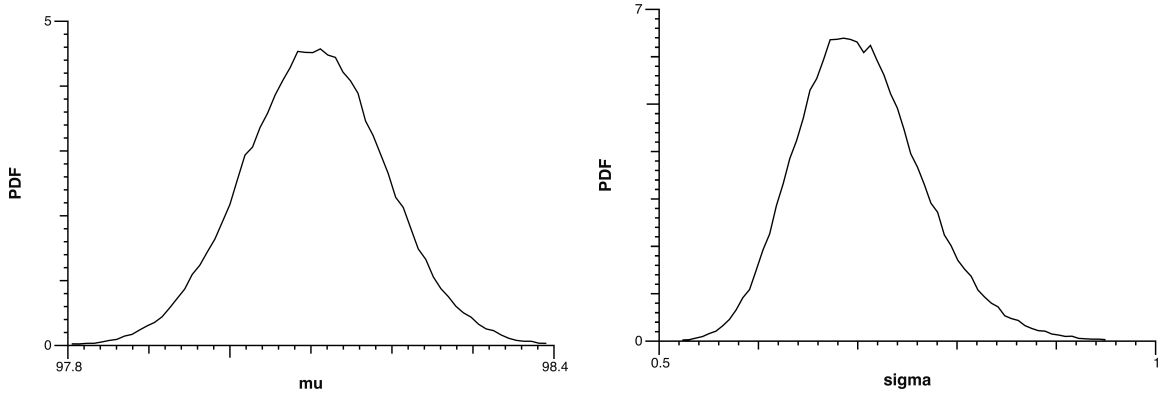
Here, as in most of these examples, the priors (lines 8–9) are intended to be vague. Runtime: 9 s

Model 13: Normal Temperature

```
1  Constants:
2  N = 65;  // # of points
3  Data:
4  y[N];
5  Variables:
6  mu, sigma, j;
7  Priors:
8  mu ~ Normal(100, 2);
9  sigma ~ Jeffreys(0.01, 10);
10 Likelihood:
11 for (j, 1:N) {
12    y[j] ~ Normal(mu, sigma);
13 }
14 Extras:
15 Monitored:
16 mu, sigma;
17 Goodness:
18 gdnsX ~ Normal(mu, sigma);
```

Marginals and goodness-of-fit are shown on the following page. A Normal distribution for the likelihood was a good choice, as expected.

Table 5: Marginals for Normal Temperature



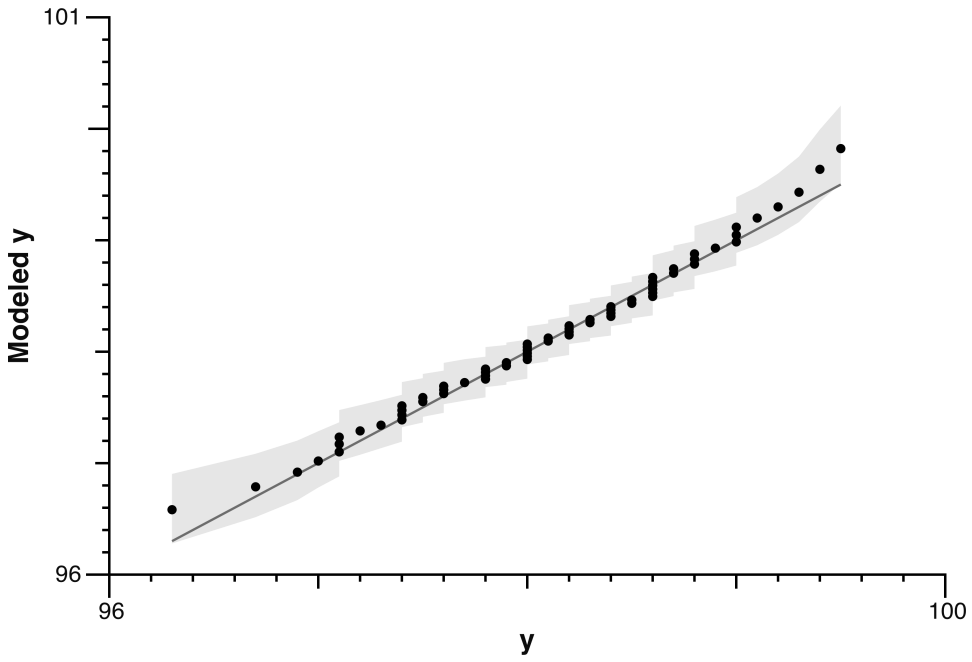


Figure 28: Goodness-of-fit for Normal Temperature

In the goodness-of-fit plot above, the credible-interval band shows the *posterior-pedictive* uncertainty for a measured temperature with the value given on the abscissa.[7] For a more obvious example, see page 50.

[7] In *MacMCMC*, the band-height limits are estimated using a random sample of 1,000 rows (parameter vectors) from the trace along with the value on the abscissa.

13 Salaries

These data comprise a set of salaries (in hundreds of dollars) obtained from a survey of college faculty in the USA. Preliminary results (prior information) suggested that such data might best be described as a mixture of two Normal distributions. The model below uses the built-in NormalMixture distribution. The MCMC run is, by default, followed by a relabeling phase to try and correct the expected label switching (see pg. 31).
Runtime: 10 m 30 s (including relabeling)

The data, shown in Figure 29, look like a mixture of two Normal distributions but they are severely overlapping which will make this a difficult analysis.

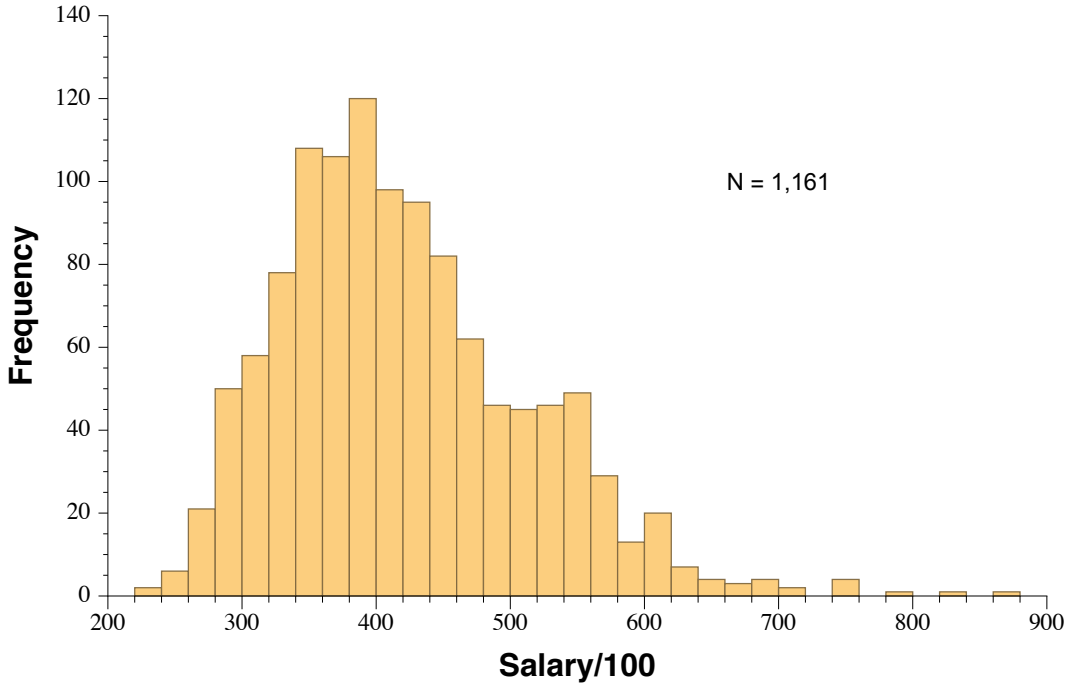


Figure 29: Salaries Data

The big problem is how to specify the two means without getting them mixed up. A common practice is to *force* the second mean to be greater (or smaller) than the first. For instance, priors might be

$$\begin{aligned} \text{mean}[1] &\sim \text{Uniform}[200, 600] \\ \text{mean}[2] &\sim \text{Uniform}[\text{mean}[1], 600] \end{aligned} \tag{1}$$

This works well when the means are far apart but, otherwise, results in marginals for the means that are skewed more than they should be. We shall try a simpler model in which nothing special is done in this regard. See model 1, shown below.
Runtime: 10 m 30 s (including relabeling)

Model 14: Salaries (model 1)

```
Constants:
N = 1161, // # of points
nc = 2; // # of components
Data:
salary[N];
Variables:
mean[nc], sigma[nc], wt[nc], i;
Priors:
for (i, 1:nc) {
   mean[i] ~ Uniform(200, 600);
   sigma[i] ~ HalfNormal(50);
}
wt[1] ~ Uniform(0.25, 0.75);
wt[2] = 1 - wt[1];
Likelihood:
for (i, 1:N) {
   salary[i] ~ NormalMixture(mean[], sigma[], wt[], nc);
}
Extras:
Monitored:
mean[], sigma[];
```

This mixture model has priors that are intentionally vague. As always, this means that all useful information comes only from the data. One result of this is that *MacMCMC* had to be run many times before the relabeling algorithm worked successfully, giving unimodal marginals for the parameters. (In this case, failure gives bimodal marginals.)

The final marginals are compared in Figure 30.

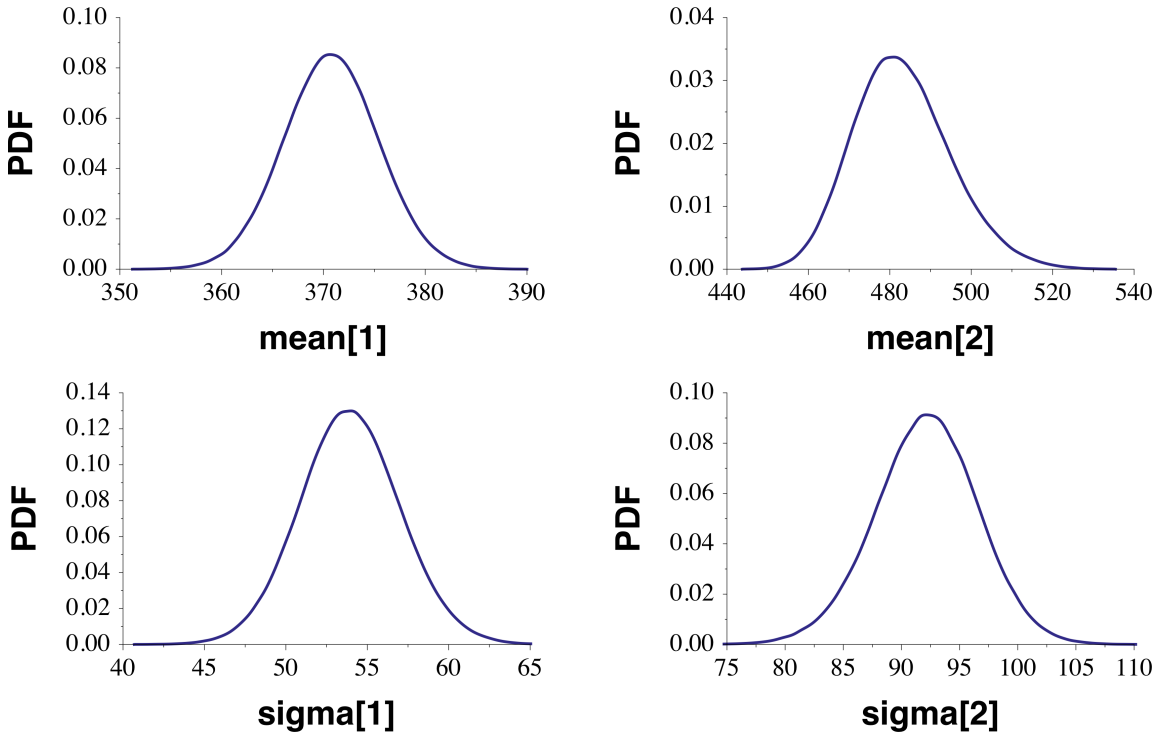


Figure 30: Marginals for Salaries Mixture (model 1)

Even with a relabeling algorithm, there is a much better idea for mixtures such as this. Consider again the forcing strategy shown in (1). We can soften the constraint. Just make the model *hierarchical* by applying the constraint to the "grandparents" of the means instead of to the "parents". One way to do this is shown below in Salaries (model 2). Here, *gp* refers to the "grandparents". The relationship should be clear from lines 10–15.

Model 2 simplifies the trace making relabeling *much* more efficient so that only one or two runs are needed to give the desired results.
Runtime: 16 m 30 s (including relabeling)

Model 15: Salaries (model 2)

```
1  Constants:
2  N = 1161,   // # of points
3  gpsig = sqrt((400^2)/(12*N)),   // for std. error
4  nc = 2;   // # of components
5  Data:
6  salary[N];
7  Variables:
8  mean[nc], sigma[nc], wt[nc], gp[nc], i;
9  Priors:
10 gp[1] ~ Uniform(200, 600);
11 gp[2] ~ Uniform(gp[1], 600);
12 for (i, 1:nc) {
13    mean[i] ~ Normal(gp[i], gpsig);
14    sigma[i] ~ HalfNormal(50);
15 }
16 wt[1] ~ Uniform(0.25, 0.75);
17 wt[2] = 1 - wt[1];
18 Likelihood:
19 for (i, 1:N) {
20    salary[i] ~ NormalMixture(mean[], sigma[], wt[], nc);
21 }
22 Extras:
23 Monitored:
24 mean[], sigma[];
```

In the literature, nearly all Bayesian models are hierarchical; it is the standard method for incorporating prior information into the model. In this document, other hierarchical models include Iris and SAT.

Figure 31 shows again a comparison of the marginals. They are virtually identical to those from model 1.

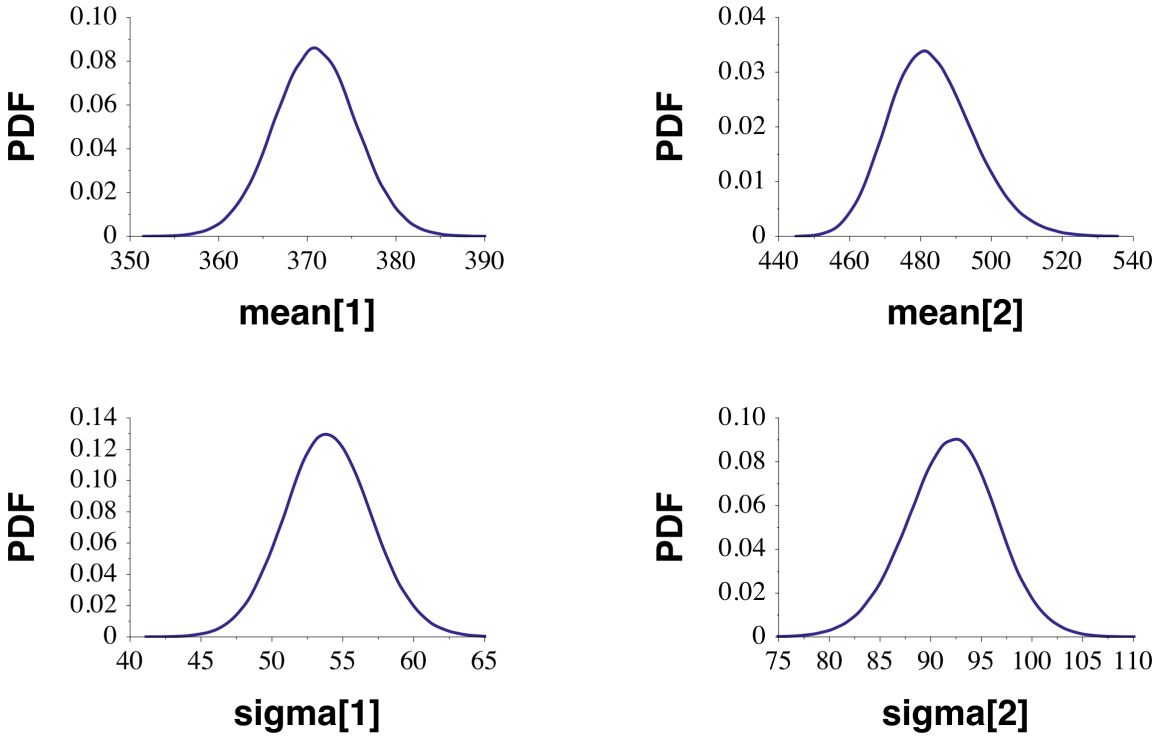


Figure 31: Marginals for Salaries Mixture (model 2)

Table 6 lists posterior values for the parameters in both models. As with the marginal plots, corresponding values are very similar.

Table 6: Posterior Values for Salaries Models

Model	Parameter	Estimate		95% Credible Interval	
		MAP	Mean	Lower Limit	Upper Limit
1	mean[1]	369.753	370.744	361.473	379.925
	mean[2]	482.296	483.043	460.394	506.770
	sigma[1]	53.4228	53.9258	47.8811	60.1002
	sigma[2]	91.9098	92.1396	83.3319	100.869
2	mean[1]	370.013	370.885	361.773	380.204
	mean[2]	482.284	483.292	460.274	506.966
	sigma[1]	53.4762	53.9802	48.8472	60.0801
	sigma[2]	91.8688	92.0678	83.0155	100.707

Figure 32 shows the data plus the model-2 posterior (solid line) as well as the separate components (dashed lines). There is considerable overlap of the components so posterior weights are a bit problematical. The weights after relabeling are listed in the Report. In this mixture, they were given as 0.524 for component #1 and 0.476 for component #2 (from component PDFs averaged over the entire trace). The *probAssign.txt* file listed 707 points with probability > 0.5 for component #1 (60.9 percent).

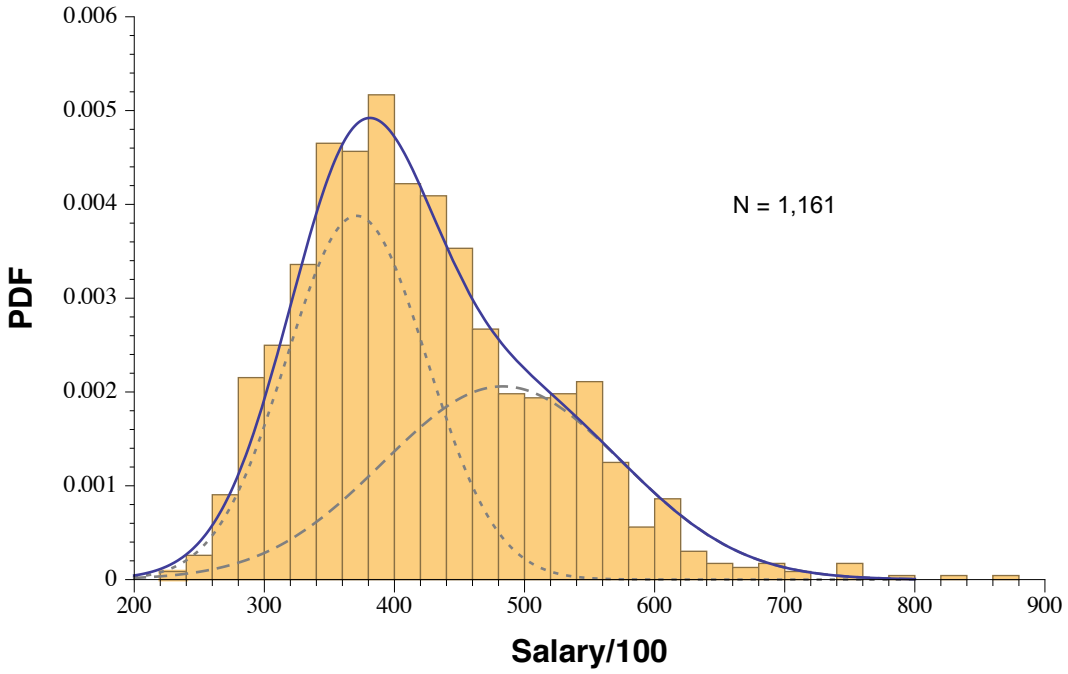


Figure 32: Data vs. Model2 (mean parameters)

14 SAT

This example illustrates one-way analysis of variance (ANOVA). The data are SAT scores for five high schools in northern Virginia, USA, for 2014. The question in this analysis is whether the putative differences, implied by distinguishing the high schools, are real. The usual ANOVA metric is the fraction of the total variance due to this distinction. The remaining variance comes from the fact that the scores in the individual rows are different as well.

The standard equations used in ANOVA compare rows with each other but assume that the within-row variance is the same for all rows. The model here makes that same assumption. It would be easy to relax this assumption (exercise left for the reader).
Runtime: 39 s

With the model as shown, one gets the following solution:

Table 7: SAT ANOVA (solution)

Unknown	Estimate		95% Credible Interval	
	MAP	Mean	Lower Limit	Upper Limit
fb	0.00159	0.301	0.0512	0.558
var.between	26	7316	300	16766
var.within	16321	15320	10399	20786
theta	1602	1586	1499	1671

The model Extra, fb, quantifies the fraction of the variance due to distinguishing the rows (high schools). Clearly, it is outweighed by the within-row variance so one cannot justifiably claim that the high schools are different in this regard.

Figure 33 shows a variance comparison.

Model 16: SAT

```
Constants:
N = 5, n1 = 25, n2 = 13, n3 = 4, n4 = 11, n5 = 5;
Data:
s1[n1], s2[n2], s3[n3], s4[n4], s5[n5];
Variables:
mu[N], theta, var.within, var.between, sig, fb, hs, s;
Priors:
theta ~ HalfNormal(10000);
var.within ~ HalfNormal(10000);
var.between ~ HalfNormal(10000);
sig = sqrt(var.between);
for (hs, 1:N) {
   mu[hs] ~ Normal(theta, sig);
}
Likelihood:
sig = sqrt(var.within);
for (s, 1:n1) {
   s1[s] ~ Normal(mu[1], sig);
}
for (s, 1:n2) {
   s2[s] ~ Normal(mu[2], sig);
}
for (s, 1:n3) {
   s3[s] ~ Normal(mu[3], sig);
}
for (s, 1:n4) {
   s4[s] ~ Normal(mu[4], sig);
}
for (s, 1:n5) {
   s5[s] ~ Normal(mu[5], sig);
}
Extras:
fb = var.between/(var.within + var.between);
Monitored:
theta, var.within, var.between, fb;
```

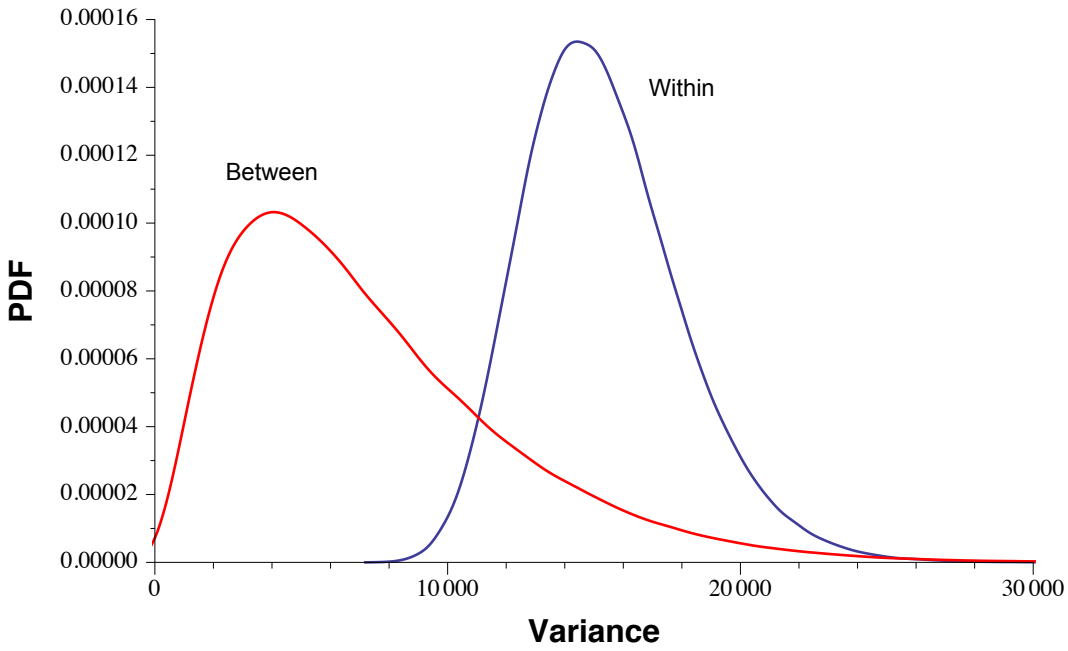


Figure 33: Variance Within vs. Variance Between

15 TriplePoint

The data for this example, just four points, comprise the current literature values for the triple-point temperature of *n*-nonadecane. The triple point of a pure substance is the {temperature, pressure} ordered pair at which the substance has its solid, liquid and gas phases all in equilibrium at the same time which is just barely possible. The *phase rule* of thermodynamics indicates that, with this threefold equilibrium, the number of degrees of freedom shrink to zero. Therefore, this point is unique making it a useful, well-defined standard for calibration purposes.

Triple-point temperatures for the normal alkanes tend to increase monotonically with carbon number and values for even-numbered homologues are generally well known since they are much more common than the odd-numbered ones (for reasons having to do with biochemistry). The triple-point temperatures for *n*-octadecane (C18) and *n*-eicosane (C20) are 301.0 K and 309.6 K, respectively.

Given the allowable range for the target temperature, a prior of Normal(305.3, 2.5) would usually be vague enough to yield good results. However, that prior distribution would stretch into invalid territory so we *truncate* it to keep it from going where Nature says it doesn't belong. Lines 8–9 in the model implement this truncation.
Runtime: 2 s

Model 17: TriplePoint

```
1  Constants:
2  N = 4;  // # of points
3  Data:
4  Tp[N];
5  Variables:
6  trueTp, sigma, i;
7  Priors:
8  trueTp ~ Generic(log((trueTp >= 301.0) && (trueTp <= 309.6))
9  - 1.83523 - 0.08*(trueTp - 305.3)^2 + 0.0893039, 301.0, 309.6);
10 sigma ~ Jeffreys(0.1, 20);
11 Likelihood:
12 for (i, 1:N) {
13    Tp[i] ~ Normal(trueTp, sigma);
14 }
15 Extras:
16 Monitored:
17 trueTp, sigma;
18 Goodness:
19 gdnsX ~ Normal(trueTp, sigma);
```

In line 8, note that *MacMCMC* defines true = 1 and false = 0, both numeric as well as Boolean. Also, the odd-looking constants in line 9 readjust the normalization constant to the truncated range.

The mean posterior estimate for the true triple-point temperatutre, trueTp, is 306.0 K with a 95-percent credible interval of [302.9, 309.3].[8]

In the q-q plot[9] below, the credible-interval band range indicates that a measurement of Tp = 314 has, *by itself*, an uncertainty that exceeds what is acceptable. Posterior-predictive uncertainties are *conditioned* on the selected observation (cf. footnote, page 39) and so are not restricted to any limits applicable to the parameter, trueTp, which has the credible interval given above.

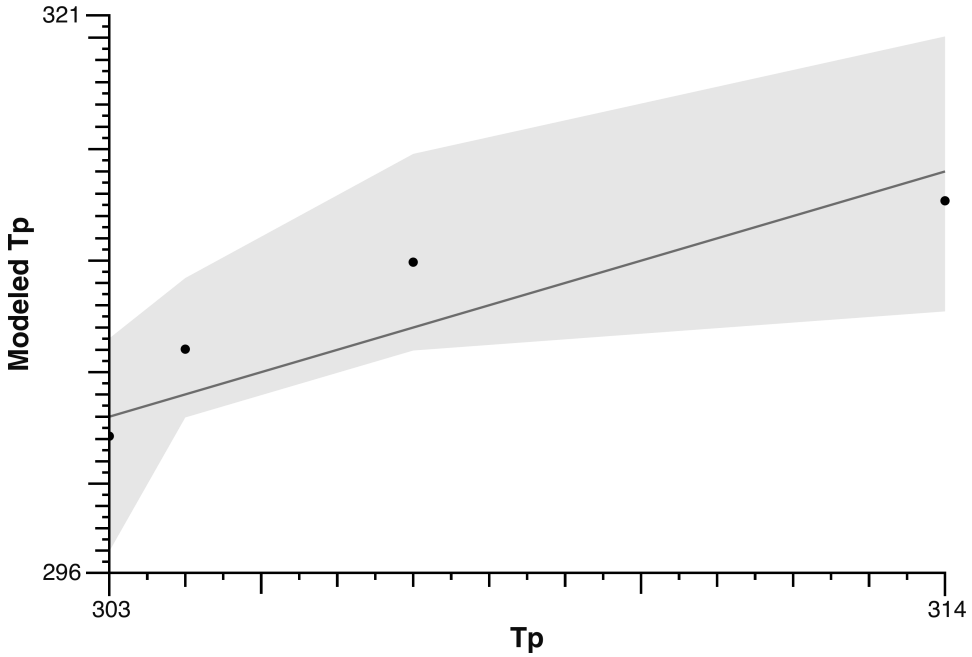


Figure 34: Goodness-of-fit for TriplePoint

Finally, FWIW, if we carry out a traditional frequentist analysis, which uses the data alone, ignoring prior information, and the t-distribution (generally recommended for a small sample), we get a maximum-likelihood value for trueTp = 307.0 K and a 95-percent *central* confidence interval of [299.1, 314.9]. Clearly, the latter cannot describe "trueTp" since it stretches far beyond the known bounds.

[8]The 99-percent interval is [301.9, 309.6].

[9]The leftmost point on this plot is half-hidden by the axis.