Mixture Analysis of the **Old Faithful Geyser** Data Using the Package **mixAK**

Arnošt Komárek
Faculty of Mathematics and Physics, Charles University in Prague

Abstract

This document supplements a paper Komárek (2009) and shows an analysis of the Old Faithful Geyser data introduced in Härdle (1991) using the R package mixAK. The data have been analysed using mixtures by several researchers, e.g., Stephens (2000), Dellaportas and Papageorgiou (2006).

Keywords: density estimation, normal mixture, R package.

This document was built on **December 20, 2011**.

1. Introduction

• Due to the fact that some code (especially MCMC) is time consuming, the code chunks found in this vignette are not run when compiling the package. You should set the variable RUN.TIMECONSUMING.CODE to TRUE to run full MCMC and related code.

• Having run full MCMC and related code, setting the variable RUN.ALLOUT to TRUE will cause that all output shown in this vignette is re-created and not taken from previously computed results.

R⇒ Setting variables RUN.ALLOUT and RUN.TIMECONSUMING.CODE.

```
> RUN.TIMECONSUMING.CODE <- FALSE
> RUN.ALLOUT <- FALSE
```

R⇒ Directory to store postscript files with figures. Figures which require chains are stored in FIGKEEPDIR directory all other figures are stored in FIGDIR directory.

```
> FIGDIR <- "./figures/"
> FIGKEEPDIR <- "./figuresKeep/"
```

R⇒ Directories with results computed in past. Objects with chains will be stored in directory specified by variable RESULTDIR All other objects will be stored in directory RESULT2DIR. The user must create these directories on his/her disk and change appropriately the values of the variables RESULTDIR and RESULT2DIR below.

```
> RESULTDIR <- "/home/komarek/RESULT_OBJ/mixAK-Faithful-S081115/"   ### must be changed by
> RESULT2DIR <- "./RESULT_OBJ/"      ### must be changed by the user
```

R⇒ Display options.

```
> options(width=80)
```

R⇒ Check whether directories where the results are to be stored exist.

```
> if (!file.exists(RESULTDIR)){
+   stop(paste("Directory ", RESULTDIR, " does not exist.\nYou have to create it or change
+ }
> if (!file.exists(RESULT2DIR)){
+   stop(paste("Directory ", RESULT2DIR, " does not exist.\nYou have to create it or chang
+ }
```

R⇒ Load results computed in past (if calculated in past). Variable Kshow determines which value for fixed number of components K is used in section 4.

```
> Kshow <- 3
> if ("Faithful-Result.RData" %in% dir(RESULT2DIR)){
+   load(paste(RESULT2DIR, "Faithful-Result.RData", sep=""))
+      ## contains ModelK (without chains), MPDensModelK, JPDensModelK
+   Model0 <- ModelK[[Kshow]]
+   MPDensModel0 <- MPDensModelK[[Kshow]]
+   JPDensModel0 <- JPDensModelK[[Kshow]]
+ }else{
+   if (!RUN.TIMECONSUMING.CODE){
+     stop(paste("Directory ", RESULT2DIR, " does not contain necessary files.\nSet RUN.TI
+   }
+ }
> if (RUN.ALLOUT){
+   if (paste("Faithful-Model0", Kshow, ".RData", sep="") %in% dir(RESULTDIR)){
+     load(paste(RESULTDIR, "Faithful-Model0", Kshow, ".RData", sep=""))
+        ## contains Model0=ModelK[[Kshow]] (chains included)
+   }else{
+     if (!RUN.TIMECONSUMING.CODE){
+       stop(paste("Directory ", RESULTDIR, " does not contain necessary files.\nSet RUN.T
+     }
+   }
+ }
```

R⇒ Load the package mixAK and packages coda and colorspace. Package coda is used to perform some basic convergence diagnostics, package colorspace is used to draw nicer image plots with estimated bivariate densities.

```
> library("mixAK")
> library("coda")
> library("colorspace")
```

2. Exploration of the data

R⇒ The data are read and summarized as follows.

```
> data("Faithful", package="mixAK")
> summary(Faithful)
```

```
   eruptions        waiting
 Min.   :1.600   Min.   :43.0
 1st Qu.:2.163   1st Qu.:58.0
 Median :4.000   Median :76.0
 Mean   :3.488   Mean   :70.9
 3rd Qu.:4.454   3rd Qu.:82.0
 Max.   :5.100   Max.   :96.0
```

R⇒ Additionally, Figure 1 shows the scatterplot and histograms of the data.

```
> postscript(paste(FIGDIR, "figFaithful01.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(bty="n")
> layout(matrix(c(0,1,1,1,1,0, 2,2,2,3,3,3), nrow=2, byrow=TRUE))
> plot(Faithful, col="red", pch=16,
+      xlab="Eruptions (min)", ylab="Waiting (min)")
> hist(Faithful$eruptions, prob=TRUE, col="sandybrown",
+      xlab="Eruptions (min)", ylab="Density", main="",
+      breaks=seq(1.4, 5.6, by=0.3))
> hist(Faithful$waiting, prob=TRUE, col="sandybrown",
+      xlab="Waiting (min)", ylab="Density", main="")
> dev.off()
```

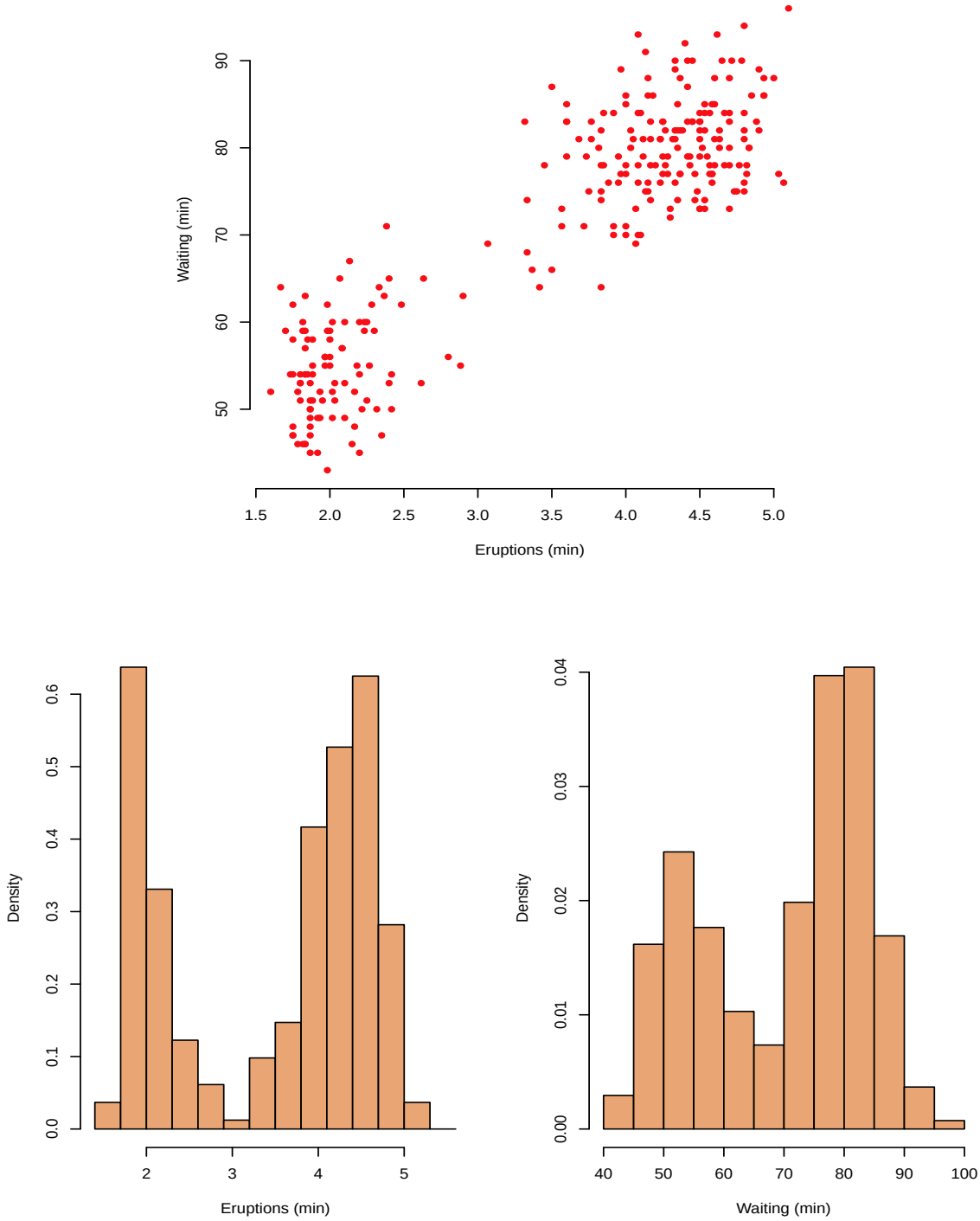

Figure 1: Scatterplot and histograms of Faithful data.

3. Preparation of the MCMC

R⇒ Length of the MCMC simulation for all models in this document (burn-in of 100 000 iterations, additional 500 000 iterations are kept for the inference, thinning of 1:10):

```
> nMCMC <- c(burn=100000, keep=500000, thin=10, info=10000)
```

R⇒ Grid of values where we evaluate and subsequently plot the predictive density for all models in this document:

```
> ygrid <- list(eruptions=seq(1, 6, length=100),
+               waiting=seq(40, 100, length=100))
```

4. Model with three mixture components

In this section, we will fit a mixture model with $K = 3$ components.

4.1. Specification of the prior distributions and MCMC simulation

R⇒ The minimal specification of the prior distribution and running MCMC with default values for all prior parameters and $K = 3$:

```
> Prior0 <- list(priorK="fixed", Kmax=3)
```

R⇒ Two chains will be generated since the argument PED is set to TRUE (output is shown from MCMC simulation performed by author):

```
> if (RUN.TIMECONSUMING.CODE){
+   set.seed(777988621)
+   Model0 <- NMixMCMC(y0=Faithful, prior=Prior0, nMCMC=nMCMC, PED=TRUE)
+ }


Chain number 1
==============
MCMC sampling started on Sat Nov 15 21:18:35 2008.
Burn-in iteration 100000
Iteration 600000
MCMC sampling finished on Sat Nov 15 21:38:44 2008.

Chain number 2
==============
MCMC sampling started on Sat Nov 15 21:38:52 2008.
Burn-in iteration 100000
Iteration 600000
MCMC sampling finished on Sat Nov 15 21:58:44 2008.

Computation of penalized expected deviance started on Sat Nov 15 21:58:53 2008.
Computation of penalized expected deviance finished on Sat Nov 15 22:09:28 2008.
```

R⇒ The prior distribution for the function NMixMCMC was the same as with

```
> Prior0 <- list(priorK="fixed", Kmax=3,
+                delta=1,
+                priormuQ="independentC",
+                xi=c(-0.1207, -0.1028), D=diag(c(9.4033, 15.1983)),
+                zeta=3, g=0.2, h=c(1.0635, 0.6580))
```

Note that due to the fact that the argument scale has not been specified in function NMixMCMC, the MCMC has been run on standardized data. Adding scale=list(shift=0, scale=1) would lead to the MCMC run on the original data.

4.2. Posterior inference

R⇒ Basic posterior summary of the fitted model is obtained using the command `print(Model0)`.

R⇒ Quantities shown in section labeled "Penalized expected deviance" are computed from two sampled chains and have the following meaning: D.expect is $\hat{D}_e$ from Komárek (2009), p(opt) is estimated optimism $\hat{p}_{opt}$ computed with unit weights, PED equals D.expect + p(opt) gives the estimate of penalized expected deviance with optimism computed without the use of importance sampling. Further, wp(opt) is estimated optimism $\hat{p}_{opt}$ computed as described in Komárek (2009), i.e., using importance sampling. Finally, wPED equals D.expect + wp(opt) gives the estimate of penalized expected deviance as described in Komárek (2009).

R⇒ Quantities shown in section labeled "Deviance information criteria" are computed separately from the first and the second sampled chain. They have the following meaning: DIC is deviance information criterion denoted as DIC in Komárek (2009), pD is the effective dimension p_D, D.bar is approximated posterior mean $\overline{D}$ of the deviance and D.in.bar is $\tilde{D}$ – deviance evaluated in the "estimate".

R⇒ Section "Posterior summary statistics for moments of mixture for original data" gives posterior summary statistics for $\mathrm{E}(\boldsymbol{Y}) = \boldsymbol{m} + \boldsymbol{S}\mathrm{E}(\boldsymbol{Y}^*)$ and quantities derived from $\mathrm{var}(\boldsymbol{Y}) = \boldsymbol{S}\,\mathrm{var}(\boldsymbol{Y}^*)\,\boldsymbol{S}'$ in the notation of Komárek (2009), separately for each generated chain.

```
> print(Model0)


     3 component normal mixture estimated using MCMC
     ================================================

Penalized expected deviance:
----------------------------
  D.expect     p(opt)        PED    wp(opt)       wPED
2254.01481   49.83073 2303.84554   65.44018 2319.45499

Deviance information criteria:
------------------------------
             DIC       pD    D.bar D.in.bar
Chain 1 2274.618 20.41526 2254.203 2233.788
Chain 2 2274.640 20.46532 2254.175 2233.710

Posterior summary statistics for moments of mixture for original data:
----------------------------------------------------------------------
Means (chain 1):
           y.Mean.1   y.Mean.2
Mean     3.48276703 70.8387857
Std.Dev. 0.06887599  0.8228232
Min.     3.12980123 67.0204657
2.5%     3.34647886 69.2058761
1st Qu.  3.43665720 70.2872820
Median   3.48330318 70.8461866
3rd Qu.  3.52945459 71.3983297
```

```
97.5%    3.61615508 72.4281424
Max.     3.77453766 74.5192929

Means (chain 2):
          y.Mean.1   y.Mean.2
Mean     3.48275249 70.8387119
Std.Dev. 0.06880969  0.8228206
Min.     3.18001025 66.6570918
2.5%     3.34689706 69.2133404
1st Qu.  3.43646715 70.2859397
Median   3.48338237 70.8451770
3rd Qu.  3.52955716 71.3972907
97.5%    3.61603051 72.4315395
Max.     3.80148421 74.6265588

Standard deviations and correlations (chain 1):
            y.SD.1  y.Corr.2.1     y.SD.2
Mean     1.13958563  0.89591889 13.6159714
Std.Dev. 0.02583515  0.01113585  0.4003479
Min.     0.06771792 -0.61547798  1.0371961
2.5%     1.08782841  0.87375280 12.8423673
1st Qu.  1.12268084  0.88950179 13.3480601
Median   1.14014370  0.89662570 13.6132654
3rd Qu.  1.15703242  0.90317118 13.8808271
97.5%    1.18828521  0.91426570 14.4002020
Max.     2.37819603  0.97004335 24.5887557

Standard deviations and correlations (chain 2):
            y.SD.1  y.Corr.2.1     y.SD.2
Mean     1.13958293  0.89597106 13.6165269
Std.Dev. 0.02599097  0.01118269  0.4034239
Min.     0.10101659 -0.84440306  1.3401567
2.5%     1.08780340  0.87391881 12.8434681
1st Qu.  1.12274437  0.88955113 13.3482394
Median   1.14012439  0.89667413 13.6142453
3rd Qu.  1.15701185  0.90321251 13.8814610
97.5%    1.18816105  0.91430662 14.4060457
Max.     2.82812950  0.98656318 36.3131940
```

R⇒ Computation of the marginal (univariate) predictive densities (separately for chain 1 and chain 2):

```
> if (RUN.TIMECONSUMING.CODE){
+   MPDensModel0 <- list()
+   MPDensModel0[[1]] <- NMixPredDensMarg(Model0[[1]], grid=ygrid)
+   MPDensModel0[[2]] <- NMixPredDensMarg(Model0[[2]], grid=ygrid)
+ }
```

R⇒ Default `plot` method for the computed object (see Figure 2):

```
> postscript(paste(FIGDIR, "figFaithful02.ps", sep=""), width=7, height=5,
+            horizontal=FALSE)
> plot(MPDensModel0[[1]])
> dev.off()
```

R⇒ Computation of the joint (bivariate) predictive densities (separately for chain 1 and chain 2):

```
> if (RUN.TIMECONSUMING.CODE){
+   JPDensModel0 <- list()
+   JPDensModel0[[1]] <- NMixPredDensJoint2(Model0[[1]], grid=ygrid)
+   JPDensModel0[[2]] <- NMixPredDensJoint2(Model0[[2]], grid=ygrid)
+ }
```

R⇒ Default `plot` method for the computed object (see Figure 3):

```
> postscript(paste(FIGDIR, "figFaithful03.ps", sep=""), width=7, height=5,
+            horizontal=FALSE)
> plot(JPDensModel0[[1]])
> dev.off()
```

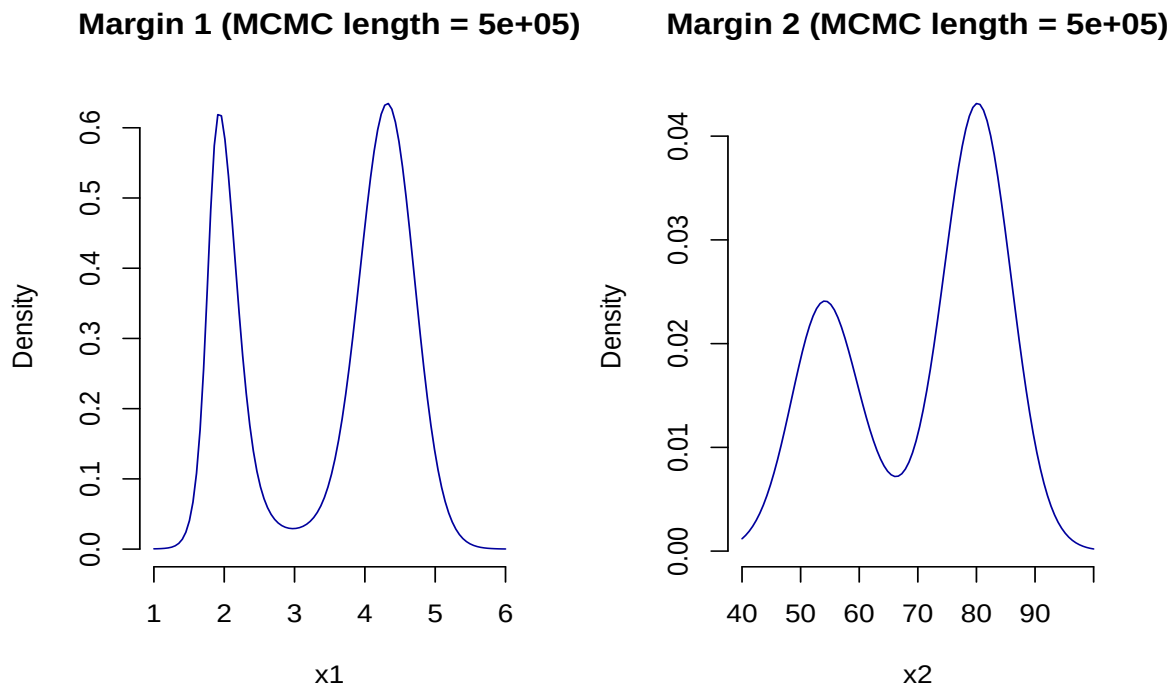

Figure 2: Default `plot` method for the marginal predictive densities based on the model with three mixture components (results from chain 1).

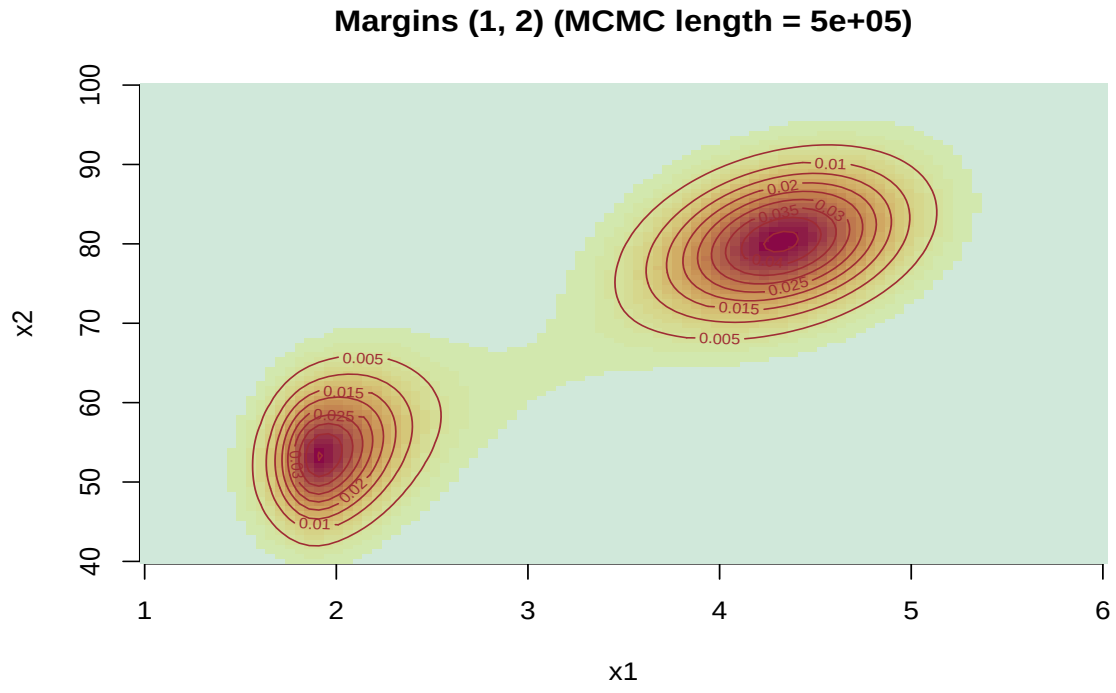

Figure 3: Default `plot` method for the joint predictive density based on the model with three mixture components (results from chain 1).

4.3. Nicer figures of posterior predictive densities

R⇒ Plot of the marginal predictive densities together with the histograms of the data, separate figures for two generated chains (see Figure 4):

```
> postscript(paste(FIGDIR, "figFaithful04.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mfrow=c(2, 2), bty="n")
> ylimE <- c(0, max(c(MPDensModel0[[1]]$dens[[1]], MPDensModel0[[2]]$dens[[1]])))
> ylimW <- c(0, max(c(MPDensModel0[[1]]$dens[[2]], MPDensModel0[[2]]$dens[[2]])))
> for (CH in 1:2){
+   hist(Faithful$eruptions, prob=TRUE, col="sandybrown",
+        xlab="Eruptions (min)", ylab="Density", main=paste("Chain", CH),
+        breaks=seq(1.4, 5.6, by=0.3), ylim=ylimE)
+   lines(MPDensModel0[[CH]]$x$x1, MPDensModel0[[CH]]$dens[[1]], col="darkblue", lwd=2)
+   hist(Faithful$waiting, prob=TRUE, col="sandybrown",
+        xlab="Waiting (min)", ylab="Density", main="", ylim=ylimW)
+   lines(MPDensModel0[[CH]]$x$x2, MPDensModel0[[CH]]$dens[[2]], col="darkblue", lwd=2)
+ }
> dev.off()
```

R⇒ Contour plot of the joint predictive density together with the scatterplot of the data, separate figures for two generated chains (see Figure 5):

```
> postscript(paste(FIGDIR, "figFaithful05.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mfrow=c(2, 1), bty="n")
> for (CH in 1:2){
+   plot(Faithful, col="red", xlab="Eruptions (min)", ylab="Waiting (min)",
+        main=paste("Chain", CH))
+   contour(JPDensModel0[[CH]]$x$x1, JPDensModel0[[CH]]$x$x2,
+           JPDensModel0[[CH]]$dens[["1-2"]],
+           col="darkblue", add=TRUE)
+ }
> dev.off()
```

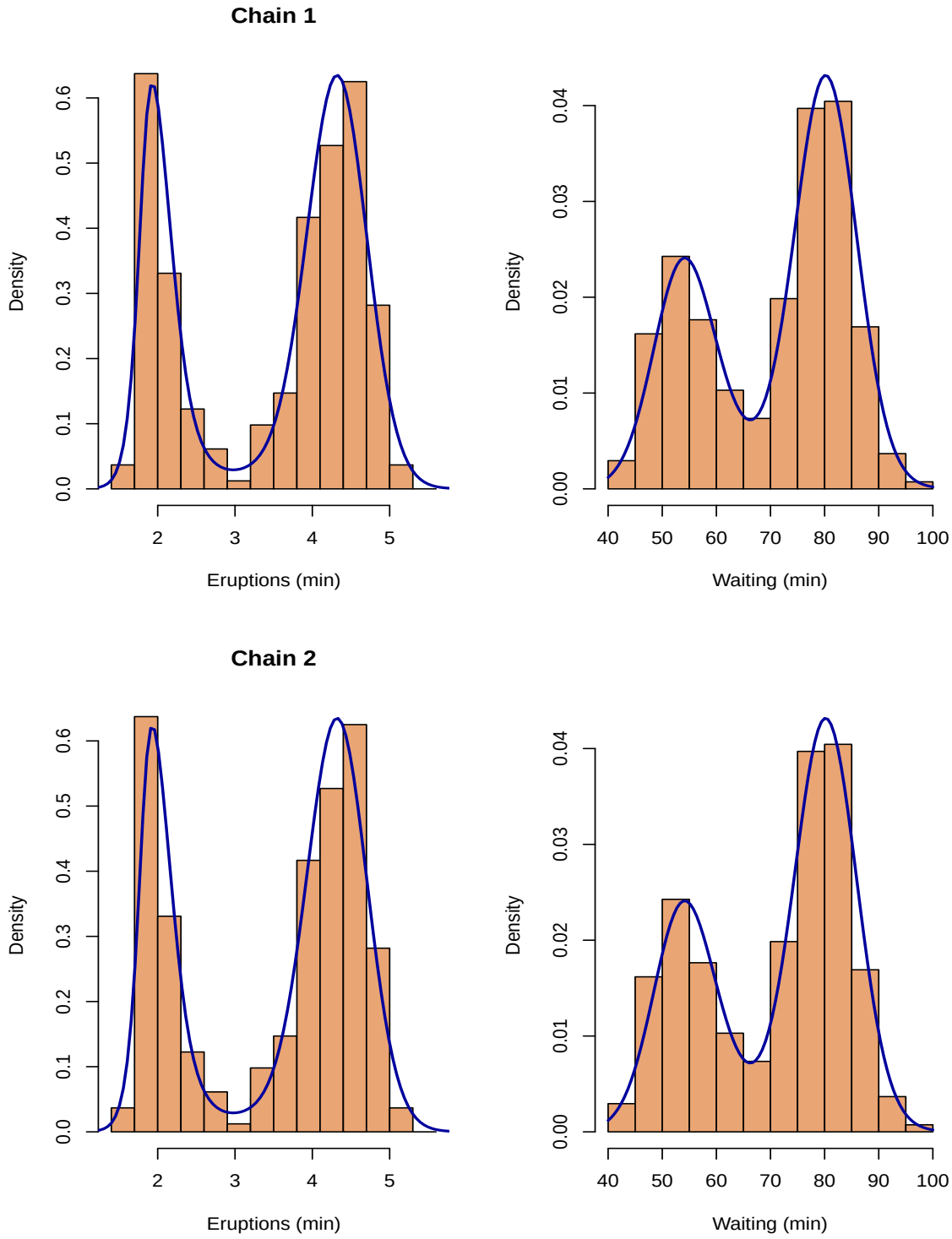

Figure 4: Marginal predictive densities (blue line) based on the model with three mixture components, separately for chain 1 and chain 2.

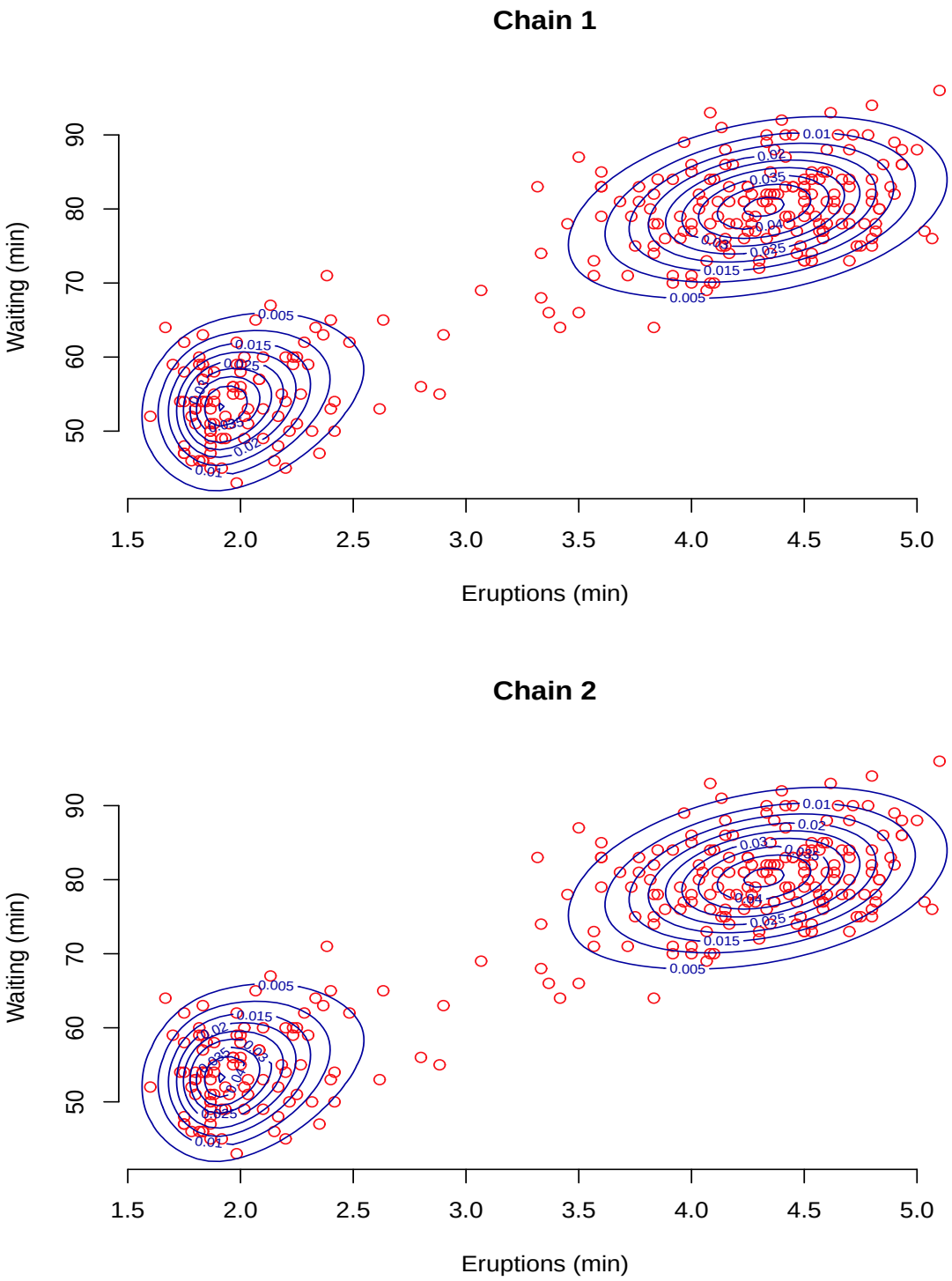

Figure 5: Contour plot of the joint predictive density based on the model with three mixture components, separately for chain 1 and chain 2.

R⇒ Image plot of the joint predictive density together with the scatterplot of the data, separate figures for two generated chains (see Figure 6):

```
> postscript(paste(FIGDIR, "figFaithful06.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mfrow=c(2, 1), bty="n")
> for (CH in 1:2){
+   plot(Faithful, col="darkblue", xlab="Eruptions (min)", ylab="Waiting (min)",
+        main=paste("Chain", CH))
+   image(JPDensModel0[[CH]]$x$x1, JPDensModel0[[CH]]$x$x2,
+         JPDensModel0[[CH]]$dens[["1-2"]], add=TRUE,
+         col=rev(heat_hcl(33, c=c(80, 30), l=c(30, 90), power=c(1/5, 1.3))))
+   points(Faithful, col="darkblue")
+ }
> dev.off()
```

Note that package `colorspace` is needed to specify the colors in the plot.

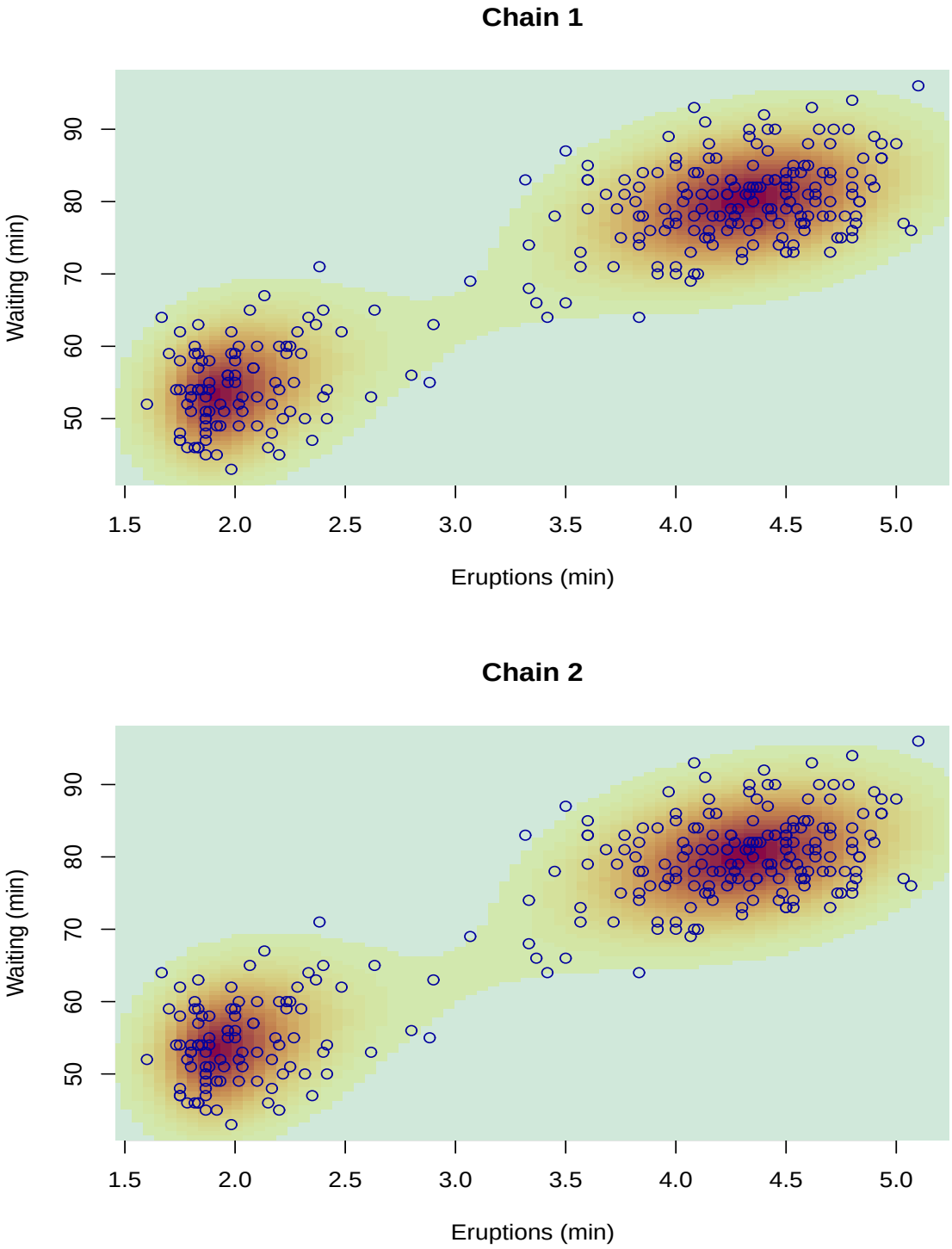

Figure 6: Image plot of the joint predictive density based on the model with three mixture components, separately for chain 1 and chain 2.

4.4. Convergence diagnostics

R⇒ Single chain convergence diagnostics using chain 1 will be shown here.

```
> CH <- 1
```

R⇒ Converting the chains into `mcmc` objects to be used in the package `coda`:

```
> if (RUN.ALLOUT){
+   start <- Model0[[CH]]$nMCMC["burn"] + 1
+   end <- Model0[[CH]]$nMCMC["burn"] + Model0[[CH]]$nMCMC["keep"]
+   chgammaInv <- mcmc(Model0[[CH]]$gammaInv, start=start, end=end)
+   chmixture <- mcmc(Model0[[CH]]$mixture, start=start, end=end)
+   chdeviance <- mcmc(Model0[[CH]]$deviance, start=start, end=end)
+ }
```

R⇒ Traceplots for selected parameters (output not shown).

R⇒ Traceplots will be drawn for last 5 000 iterations only:

```
> if (RUN.ALLOUT){
+   tstart <- 495001
+   tend <- 500000
+   titers <- tstart:tend
+   chgammaInv2 <- mcmc(Model0[[CH]]$gammaInv[titers,], start=tstart, end=tend)
+   chmixture2 <- mcmc(Model0[[CH]]$mixture[titers,], start=tstart, end=tend)
+   chdeviance2 <- mcmc(Model0[[CH]]$deviance[titers,], start=tstart, end=tend)
+ }


> if (RUN.ALLOUT){
+   lwd <- 0.5
+   postscript(paste(FIGKEEPDIR, "figFaithful07.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(mfrow=c(2, 3), bty="n")
+   traceplot(chmixture2[, "y.Mean.1"], smooth=FALSE, col="darkblue", lwd=lwd,
+             main="E(eruptions)")
+   traceplot(chmixture2[, "y.Mean.2"], smooth=FALSE, col="darkblue", lwd=lwd,
+             main="E(waitings)")
+   traceplot(chgammaInv2[, "gammaInv1"], smooth=FALSE, col="brown", lwd=lwd,
+             main="gamma^{-1}")
+   traceplot(chmixture2[, "y.SD.1"], smooth=FALSE, col="darkgreen", lwd=lwd,
+             main="sd(eruptions)")
+   traceplot(chmixture2[, "y.SD.2"], smooth=FALSE, col="darkgreen", lwd=lwd,
+             main="sd(waitings)")
+   traceplot(chmixture2[, "y.Corr.2.1"], smooth=FALSE, col="red", lwd=lwd,
+             main="corr(eruptions, waitings)")
+   dev.off()
+ }
```

```
> if (RUN.ALLOUT){
+   postscript(paste(FIGKEEPDIR, "figFaithful08.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(mfrow=c(2, 2), bty="n")
+   traceplot(chdeviance2[, "LogL0"], smooth=FALSE, col="red", lwd=lwd,
+             main="Log(L0)")
+   traceplot(chdeviance2[, "LogL1"], smooth=FALSE, col="red", lwd=lwd,
+             main="Log(L1)")
+   traceplot(chdeviance2[, "dev.complete"], smooth=FALSE, col="red", lwd=lwd,
+             main="D(complete)")
+   traceplot(chdeviance2[, "dev.observed"], smooth=FALSE, col="red", lwd=lwd,
+             main="D(observed)")
+   dev.off()
+ }
```

R⇒ Posterior density estimates for selected parameters (see Figure 7):

```
> if (RUN.ALLOUT){
+   postscript(paste(FIGKEEPDIR, "figFaithful09.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(mfrow=c(2, 3), bty="n")
+   densplot(chmixture[, "y.Mean.1"], show.obs=FALSE, col="darkblue",
+            main="E(eruptions)")
+   densplot(chmixture[, "y.Mean.2"], show.obs=FALSE, col="darkblue",
+            main="E(waitings)")
+   densplot(chgammaInv[, "gammaInv1"], show.obs=FALSE, col="brown",
+            main="gamma^{-1}")
+   densplot(chmixture[, "y.SD.1"], show.obs=FALSE, col="darkgreen",
+            main="sd(eruptions)")
+   densplot(chmixture[, "y.SD.2"], show.obs=FALSE, col="darkgreen",
+            main="sd(waitings)")
+   densplot(chmixture[, "y.Corr.2.1"], show.obs=FALSE, col="red",
+            main="corr(eruptions, waitings)")
+   dev.off()
+ }
```

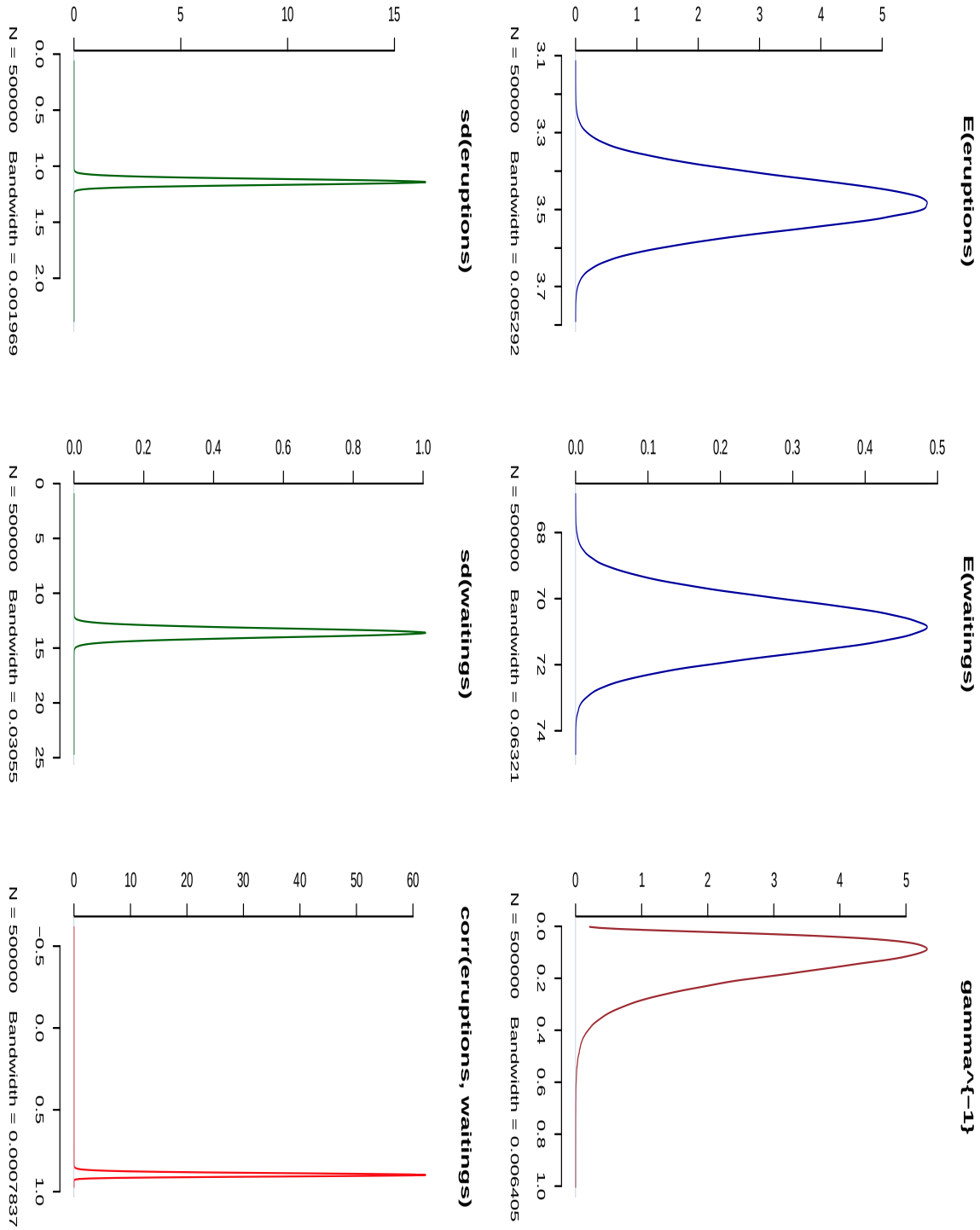

Figure 7: Model with three mixture components. Posterior density estimates for selected parameters.

R⇒ Autocorrelation plots for selected parameters (see Figure 8):

```
> if (RUN.ALLOUT){
+   postscript(paste(FIGKEEPDIR, "figFaithful10.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(mfrow=c(2, 3), bty="n")
+   autocorr.plot(chmixture[, "y.Mean.1"], auto.layout=FALSE,
+                 ask=FALSE, col="darkblue", main="E(eruptions)")
+   autocorr.plot(chmixture[, "y.Mean.2"], auto.layout=FALSE,
+                 ask=FALSE, col="darkblue", main="E(waitings)")
+   autocorr.plot(chgammaInv[, "gammaInv1"], auto.layout=FALSE,
+                 ask=FALSE, col="brown", main="gamma^{-1}")
+   autocorr.plot(chmixture[, "y.SD.1"], auto.layout=FALSE,
+                 ask=FALSE, col="darkgreen", main="sd(eruptions)")
+   autocorr.plot(chmixture[, "y.SD.2"], auto.layout=FALSE,
+                 ask=FALSE, col="darkgreen", main="sd(waitings)")
+   autocorr.plot(chmixture[, "y.Corr.2.1"], auto.layout=FALSE,
+                 ask=FALSE, col="red", main="corr(eruptions, waitings)")
+   dev.off()
+ }
```

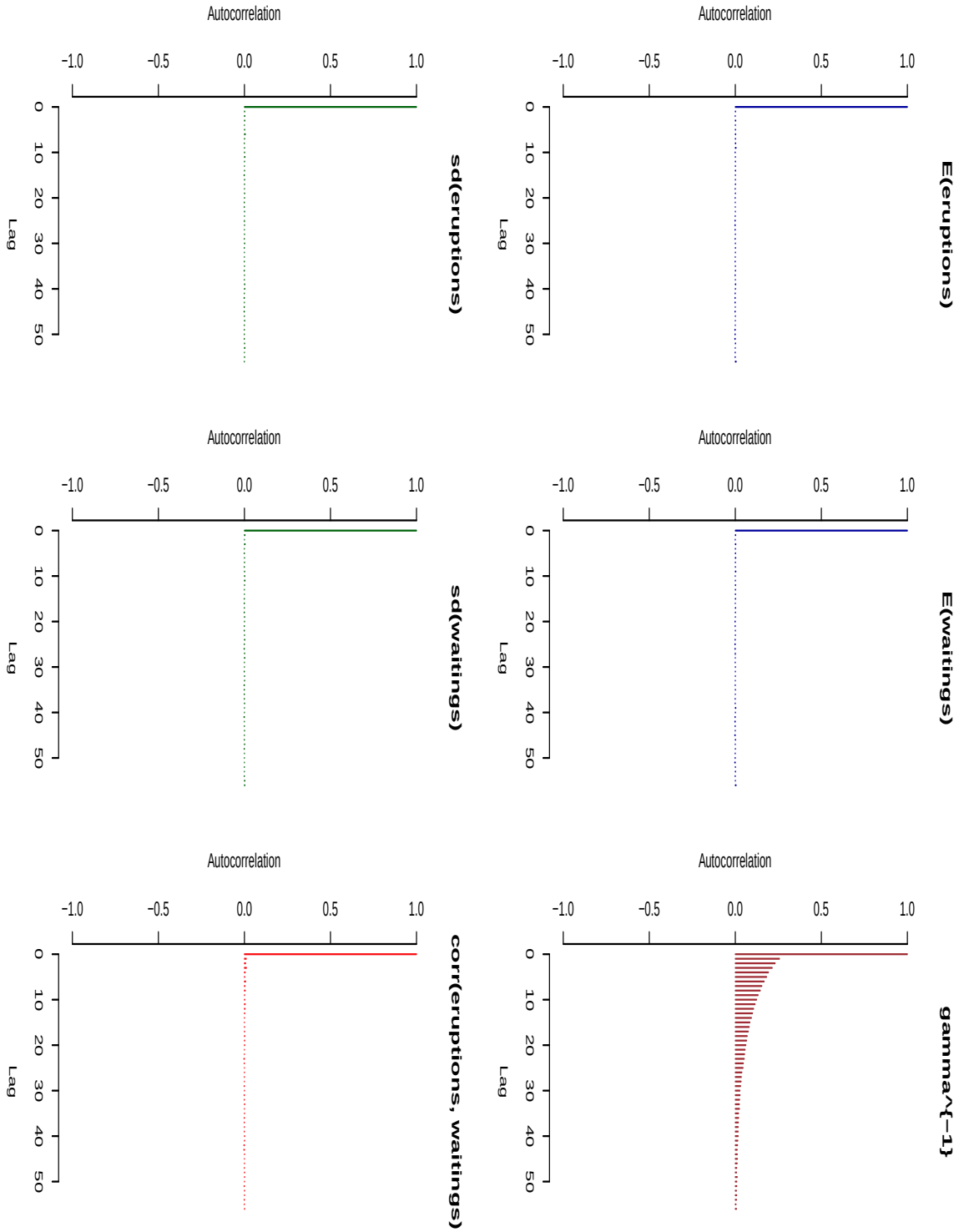

Figure 8: Model with three mixture components. Autocorrelation plots for selected parameters.

R⇒ In advance, we know that the posterior distribution has 3! symmetric modes which should all be visited by the Markov chain. Indication of failing to visit all the modes is provided by exploration of posterior distribution of mixture weights, means and variance-covariance matrices. If no identifiability constraints are imposed (as is done in our implementation of MCMC), posterior distributions of mixture weights, means and variance-covariance matrices should be identical for all mixture components if the chain suceeded to visit all the modes of the posterior. The following figures show possibility for exploration of these posterior distributions (see Figures 9, 10, 11 for results). To make the size of the resulting pdf file reasonable, all scatterplots are drawn for 500 randomly selected iterations only.

```
> if (RUN.ALLOUT){
+   PCH <- 1;  CEX <- 0.5
+   set.seed(770328)
+   SELECT <- sample(end-start+1, size=500, replace=FALSE)
+
+   MuDens1 <- MuDens2 <- list()
+   for (j in 1:3){
+     MuDens1[[j]] <- density(Model0[[CH]]$mu[,(j-1)*2+1])
+     MuDens2[[j]] <- density(Model0[[CH]]$mu[,j*2])
+   }
+   LTY <- c(1, 2, 4)
+   COL <- c("blue", "red", "darkgreen")
+
+   XLIM <- c(-2, 2);  YLIM <- c(-2, 2);
+   XLAB <- "Mean, margin 1";  YLAB <- "Mean, margin 2"
+   postscript(paste(FIGKEEPDIR, "figFaithful16.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(bty="n")
+   layout(matrix(c(1,0, 1,4, 2,4, 2,5, 3,5, 3,6), ncol=2, byrow=TRUE))
+   plot(Model0[[CH]]$mu[SELECT, "mu.1.1"], Model0[[CH]]$mu[SELECT, "mu.1.2"],
+        col="blue", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(mu[1]))
+   plot(Model0[[CH]]$mu[SELECT, "mu.2.1"], Model0[[CH]]$mu[SELECT, "mu.2.2"],
+        col="blue", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(mu[2]))
+   plot(Model0[[CH]]$mu[SELECT, "mu.3.1"], Model0[[CH]]$mu[SELECT, "mu.3.2"],
+        col="blue", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(mu[3]))
+   #
+   plot(MuDens1[[1]]$x, MuDens1[[1]]$y, type="l", lty=LTY[1], col=COL[1],
+        xlab="Mean, margin 1", ylab="Posterior density", xlim=XLIM,
+        main="Margin 1")
+   for (j in 2:3) lines(MuDens1[[j]]$x, MuDens1[[j]]$y, lty=LTY[j], col=COL[j])
+   #
+   plot(MuDens2[[1]]$x, MuDens2[[1]]$y, type="l", lty=LTY[1], col=COL[1],
+        xlab="Mean, margin 2", ylab="Posterior density", xlim=YLIM,
+        main="Margin 2")
```

```
+   for (j in 2:3) lines(MuDens2[[j]]$x, MuDens2[[j]]$y, lty=LTY[j], col=COL[j])
+   #
+   plot(0:100, 0:100, type="n", xlab="", ylab="", xaxt="n", yaxt="n")
+   legend(10, 99, paste("Comp.", 1:3), lty=LTY, col=COL, bty="n", y.intersp=1)
+   dev.off()
+
+   XLIM <- c(0, 1)
+   YLIM <- c(0, 1.5)
+   XLAB <- "Variance, margin 1"
+   YLAB <- "Variance, margin 2"
+   postscript(paste(FIGKEEPDIR, "figFaithful17.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(bty="n")
+   layout(matrix(c(0,1,1,0, 2,2,3,3), nrow=2, byrow=TRUE))
+   plot(Model0[[CH]]$Sigma[SELECT, "Sigma1.1.1"],
+        Model0[[CH]]$Sigma[SELECT, "Sigma1.2.2"],
+        col="red", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(Sigma[1]))
+   plot(Model0[[CH]]$Sigma[SELECT, "Sigma2.1.1"],
+        Model0[[CH]]$Sigma[SELECT, "Sigma2.2.2"],
+        col="red", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(Sigma[2]))
+   plot(Model0[[CH]]$Sigma[SELECT, "Sigma3.1.1"],
+        Model0[[CH]]$Sigma[SELECT, "Sigma3.2.2"],
+        col="red", pch=PCH, cex=CEX, xlim=XLIM, ylim=YLIM,
+        xlab=XLAB, ylab=YLAB, main=expression(Sigma[3]))
+   dev.off()
+
+   XLIM <- c(0, 1)
+   YLIM <- c(0, 4.5)
+   XLAB <- "Weight"
+   YLAB <- "Density"
+   postscript(paste(FIGKEEPDIR, "figFaithful18.ps", sep=""), width=7, height=10,
+              horizontal=FALSE)
+   par(bty="n")
+   layout(matrix(c(0,1,1,0, 2,2,3,3), nrow=2, byrow=TRUE))
+   hist(Model0[[CH]]$w[, "w1"], prob=TRUE, col="sandybrown",
+        xlim=XLIM, ylim=YLIM, xlab=XLAB, ylab=YLAB, main=expression(w[1]))
+   hist(Model0[[CH]]$w[, "w2"], prob=TRUE, col="sandybrown",
+        xlim=XLIM, ylim=YLIM, xlab=XLAB, ylab=YLAB, main=expression(w[2]))
+   hist(Model0[[CH]]$w[, "w3"], prob=TRUE, col="sandybrown",
+        xlim=XLIM, ylim=YLIM, xlab=XLAB, ylab=YLAB, main=expression(w[3]))
+   dev.off()
+ }
```

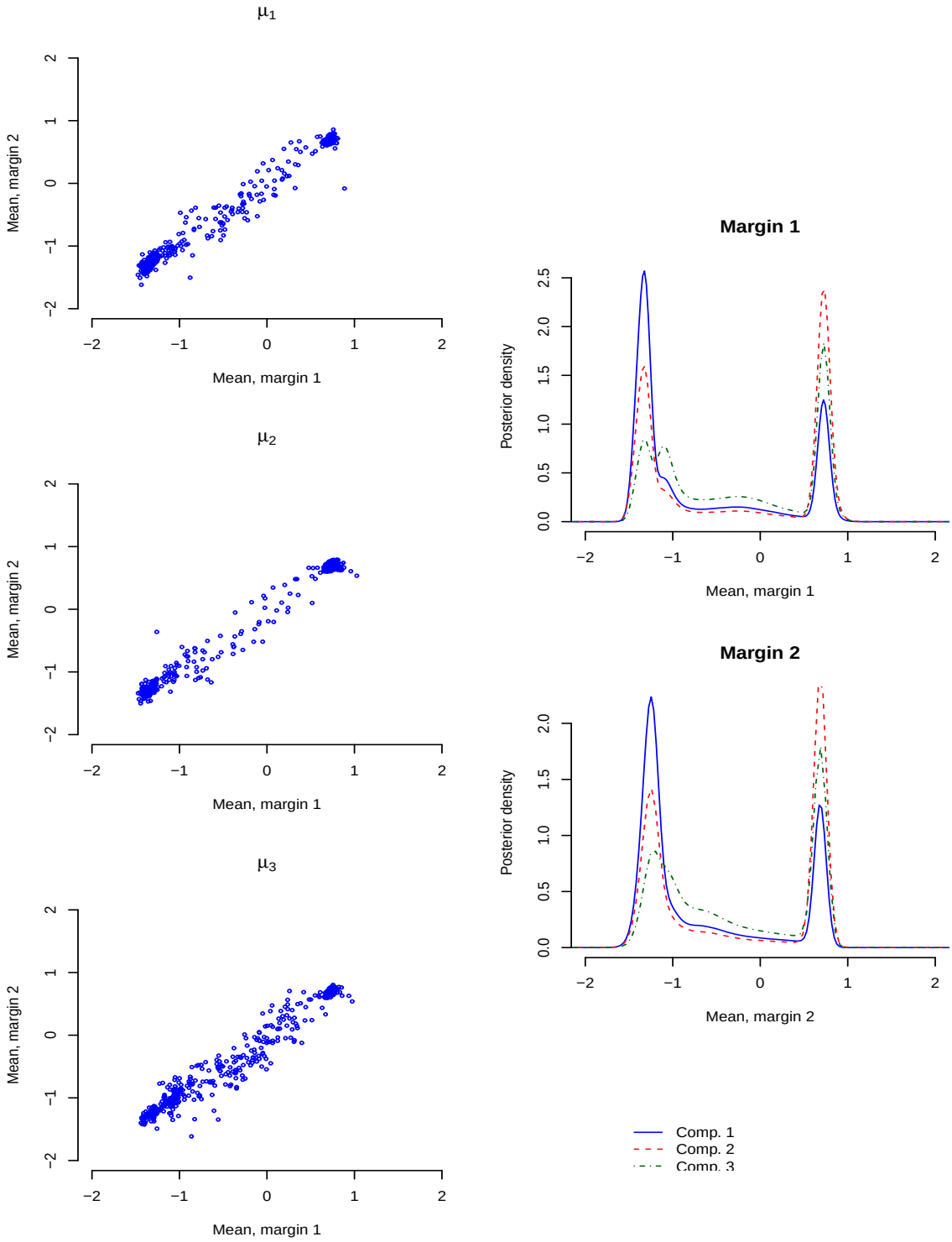

Figure 9: Model with three mixture components. Scatterplots of sampled mixture means (500 randomly selected iterations) without imposing any identifiability constraints and posterior densities for marginal mixture means in the three components.

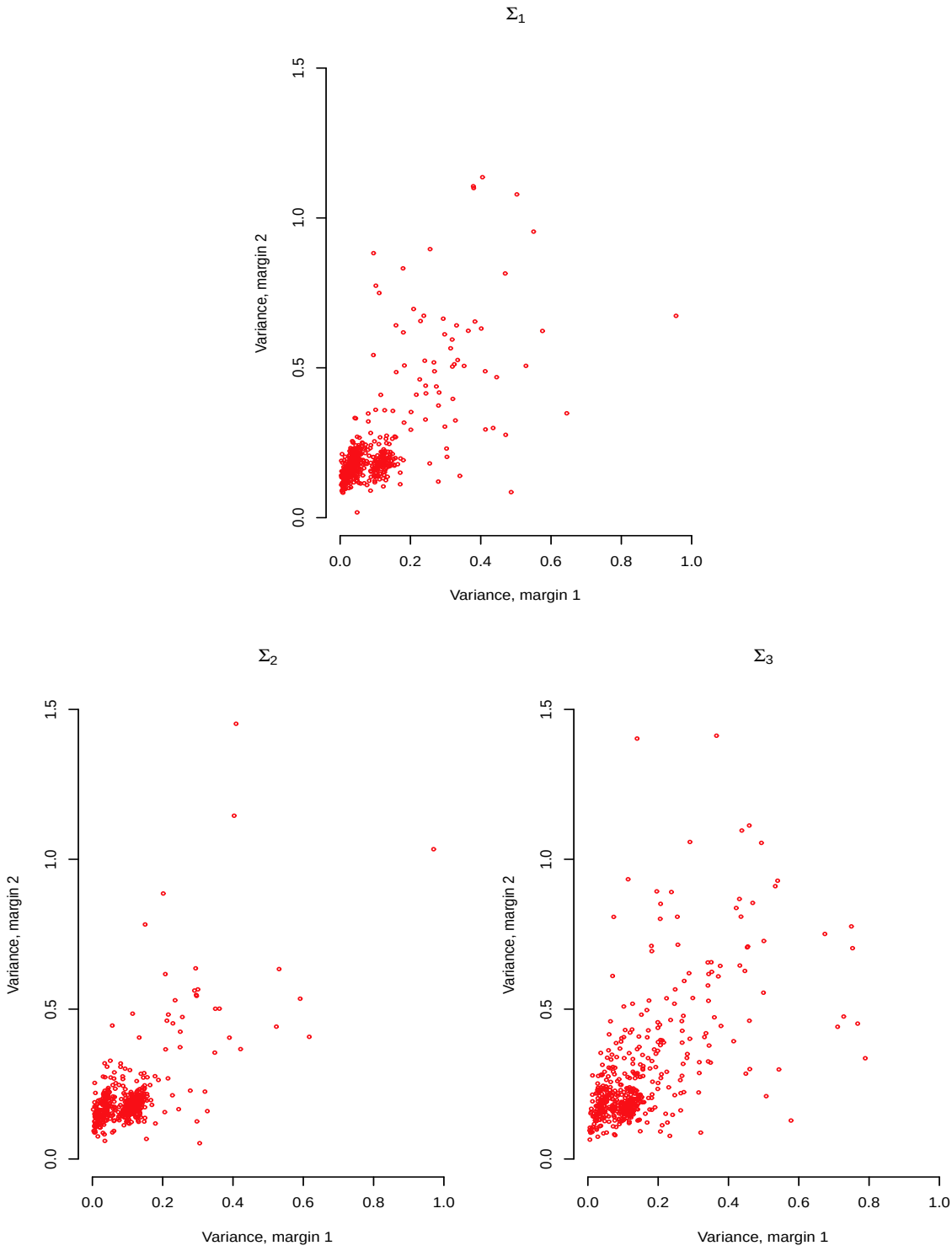

Figure 10: Model with three mixture components. Scatterplots of sampled mixture variances (500 randomly selected iterations) without imposing any identifiability constraints.

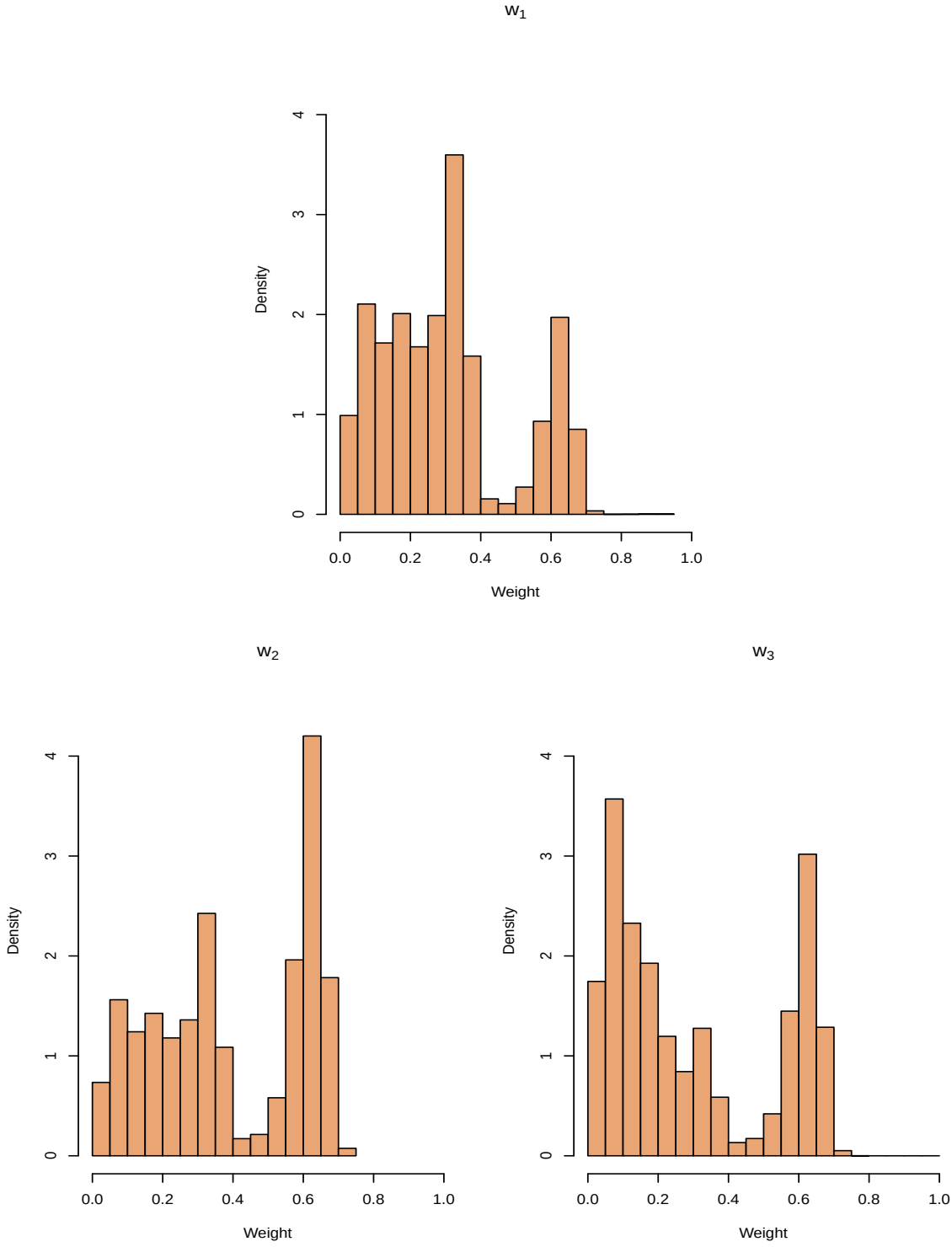

Figure 11: Model with three mixture components. Histograms of sampled mixture weights without imposing any identifiability constraints.

5. Models with different fixed numbers of components

In this section, we will fit a mixture model for $K = 1, \ldots, 10$, compare the deviance based quantities and predictive densities.

R⇒ Running the MCMC simulation for $K = 1, \ldots, 10$ (output printed during the MCMC run on the screen not shown), computation of predictive densities.

R⇒ After predictive densities are computed, we remove all chains from resulting objects (to save some memory).

```
> if (RUN.TIMECONSUMING.CODE){
+   Seed <- c(777988621, 777988621, 777988621, 777988621, 780830,
+             780830, 777988621, 777988621, 777988621, 777988621)
+ #
+   Keep <- c("iter", "nMCMC", "dim", "prior", "init", "RJMCMC",
+             "scale", "state", "freqK", "propK", "DIC", "moves",
+             "pm.y", "pm.z", "pm.indDev", "pred.dens", "summ.y.Mean",
+             "summ.y.SDCorr", "summ.z.Mean", "summ.z.SDCorr")
+ #
+   ModelK <- list()
+   MPDensModelK <- list()
+   JPDensModelK <- list()
+   for (k in 1:10){
+     set.seed(Seed[k])
+     cat(paste("K = ", k, "\n-------------------------------\n", sep=""))
+     PriorNow <- Prior0
+     PriorNow$Kmax <- k
+     ModelK[[k]] <- NMixMCMC(y0=Faithful, prior=PriorNow, nMCMC=nMCMC, PED=TRUE)
+ #
+     cat(paste("\nComputation of marginal pred. densities started on ", date(),
+               "\n", sep=""))
+     MPDensModelK[[k]] <- list()
+     MPDensModelK[[k]][[1]] <- NMixPredDensMarg(ModelK[[k]][[1]], grid=ygrid)
+     MPDensModelK[[k]][[2]] <- NMixPredDensMarg(ModelK[[k]][[2]], grid=ygrid)
+     cat(paste("Computation of marginal pred. densities finished on ", date(),
+               "\n\n", sep=""))
+ #
+     cat(paste("Computation of joint pred. densities started on ", date(),
+               "\n", sep=""))
+     JPDensModelK[[k]] <- list()
+     JPDensModelK[[k]][[1]] <- NMixPredDensJoint2(ModelK[[k]][[1]], grid=ygrid)
+     JPDensModelK[[k]][[2]] <- NMixPredDensJoint2(ModelK[[k]][[2]], grid=ygrid)
+     cat(paste("Computation of joint pred. densities finished on ", date(),
+               "\n\n\n", sep=""))
+ #
+     ModelK[[k]][[1]] <- ModelK[[k]][[1]][Keep]
+     ModelK[[k]][[2]] <- ModelK[[k]][[2]][Keep]
+     class(ModelK[[k]][[1]]) <- class(ModelK[[k]][[2]]) <- "NMixMCMC"
```

```
+   }
+ }
```

R⇒ Summary of PED and DIC's for the fitted models:

```
> PED <- ModelK[[1]]$PED
> DIC <- list(Chain1=ModelK[[1]][[1]]$DIC, Chain2=ModelK[[1]][[2]]$DIC)
> for (k in 2:length(ModelK)){
+   PED <- rbind(PED, ModelK[[k]]$PED)
+   DIC[[1]] <- rbind(DIC[[1]], ModelK[[k]][[1]]$DIC)
+   DIC[[2]] <- rbind(DIC[[2]], ModelK[[k]][[2]]$DIC)
+ }
> rownames(PED) <- rownames(DIC[[1]]) <- rownames(DIC[[2]]) <- paste("K=", 1:length(ModelK
```

```
> print(PED)
```

```
     D.expect    p(opt)      PED   wp(opt)     wPED
K=1  2584.614  10.07656 2594.691  10.10785 2594.722
K=2  2271.574  22.90130 2294.475  22.97497 2294.549
K=3  2254.015  49.83073 2303.846  65.44018 2319.455
K=4  2261.309 128.95254 2390.261 329.28420 2590.593
K=5  2270.776 300.87314 2571.649 617.23516 2888.011
K=6  2271.931 433.32046 2705.252 686.77180 2958.703
K=7  2271.061 544.32400 2815.385 733.33584 3004.396
K=8  2270.613 642.65954 2913.273 782.67754 3053.291
K=9  2271.691 736.80114 3008.492 860.15062 3131.842
K=10 2271.252 818.84434 3090.097 907.26121 3178.513
```

```
> print(DIC)
```

```
$Chain1
          DIC        pD    D.bar D.in.bar
K=1  2588.514  3.897188 2584.617 2580.720
K=2  2282.605 11.033362 2271.572 2260.538
K=3  2274.618 20.415258 2254.203 2233.788
K=4  2300.796 38.538356 2262.258 2223.720
K=5  2322.212 48.817365 2273.394 2224.577
K=6  2319.594 47.292302 2272.302 2225.010
K=7  2320.086 47.384916 2272.701 2225.316
K=8  2316.717 45.617064 2271.100 2225.483
K=9  2319.168 46.840837 2272.327 2225.486
K=10 2317.516 46.040202 2271.476 2225.435

$Chain2
```

```
           DIC        pD    D.bar D.in.bar
K=1   2588.501  3.888788 2584.612 2580.723
K=2   2282.611 11.034632 2271.576 2260.542
K=3   2274.640 20.465325 2254.175 2233.710
K=4   2301.159 38.693754 2262.465 2223.772
K=5   2323.386 49.862152 2273.524 2223.662
K=6   2322.783 48.834225 2273.948 2225.114
K=7   2318.592 46.659287 2271.932 2225.273
K=8   2318.144 46.332396 2271.812 2225.480
K=9   2318.270 46.417038 2271.853 2225.436
K=10 2316.980 45.758336 2271.222 2225.463
```

R⇒ Plot of the marginal predictive densities (eruptions) for different values of K, densities computed from chain 1 (see Figures 12 and 14):

```
> CH <- 1
> postscript(paste(FIGDIR, "figFaithful11.ps", sep=""), width=6, height=6,
+            horizontal=FALSE)
> par(mfrow=c(1, 1), bty="n")
> hist(Faithful$eruptions, prob=TRUE, col="grey90",
+      xlab="Eruptions (min)", ylab="Density", main="",
+      breaks=seq(1.4, 5.6, by=0.3))
> for (k in 1:10){
+   lines(MPDensModelK[[k]][[CH]]$x$x1, MPDensModelK[[k]][[CH]]$dens[[1]],
+         col="red")
+ }
> dev.off()


> postscript(paste(FIGDIR, "figFaithful12.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mar=c(3, 2, 2, 1)+0.1)
> par(mfrow=c(5, 2), bty="n")
> for (k in 1:10){
+   hist(Faithful$eruptions, prob=TRUE, col="lightblue",
+        xlab="", ylab="", main=paste("K = ", k, sep=""),
+        breaks=seq(1.4, 5.6, by=0.3))
+   lines(MPDensModelK[[k]][[CH]]$x$x1, MPDensModelK[[k]][[CH]]$dens[[1]],
+         col="red", lwd=2)
+ }
> dev.off()
```

R⇒ Plot of the marginal predictive densities (waiting) for different values of K, densities computed from chain 1 (see Figures 13 and 15):

```
> CH <- 1
> postscript(paste(FIGDIR, "figFaithful13.ps", sep=""), width=6, height=6,
+            horizontal=FALSE)
> par(mfrow=c(1, 1), bty="n")
> hist(Faithful$waiting, prob=TRUE, col="grey90",
+      xlab="Waiting (min)", ylab="Density", main="")
> for (k in 1:10){
+   lines(MPDensModelK[[k]][[CH]]$x$x2, MPDensModelK[[k]][[CH]]$dens[[2]],
+         col="red")
+ }
> dev.off()


> postscript(paste(FIGDIR, "figFaithful14.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mar=c(3, 2, 2, 1)+0.1)
> par(mfrow=c(5, 2), bty="n")
> for (k in 1:10){
+   hist(Faithful$waiting, prob=TRUE, col="lightblue",
+        xlab="", ylab="", main=paste("K = ", k, sep=""))
+   lines(MPDensModelK[[k]][[CH]]$x$x2, MPDensModelK[[k]][[CH]]$dens[[2]],
+         col="red", lwd=2)
+ }
> dev.off()
```

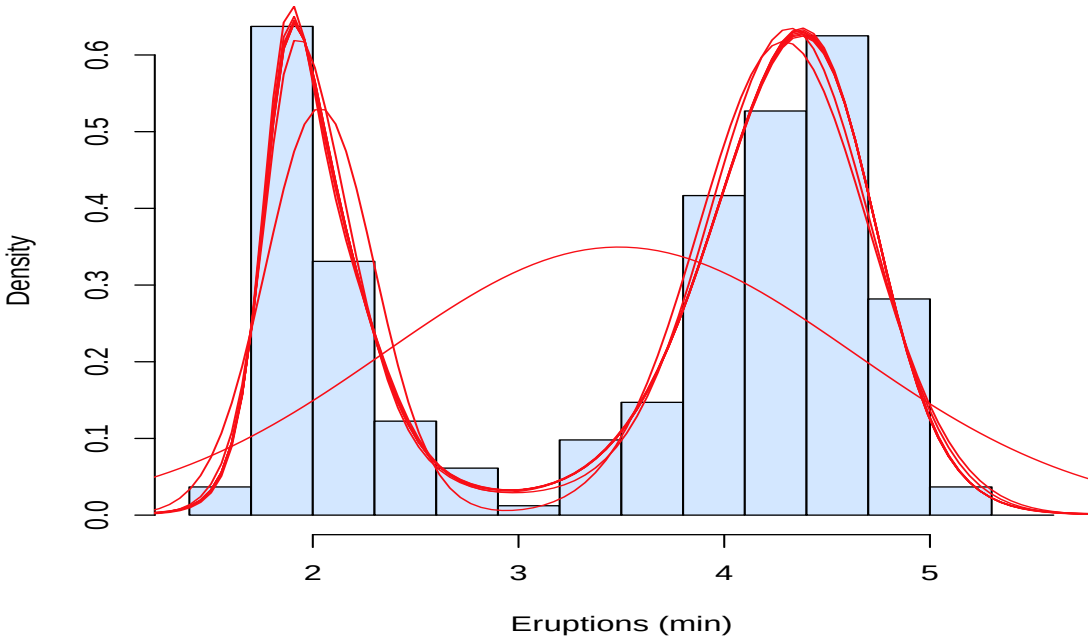

Figure 12: Marginal predictive density for eruptions based on the models with a fixed number of mixture components, results from chain 1.

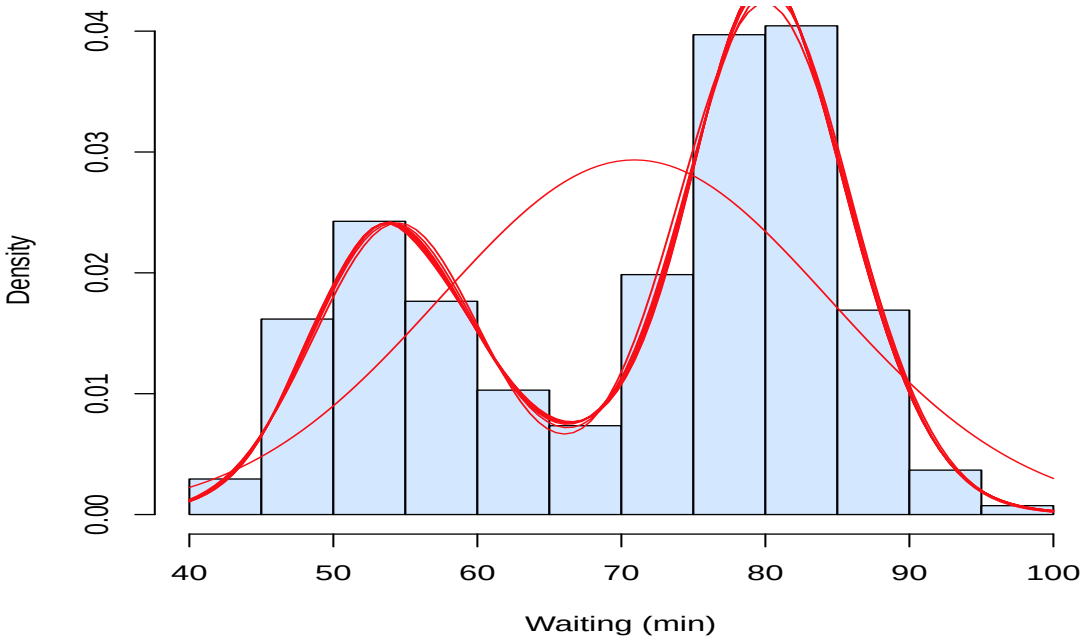

Figure 13: Marginal predictive density for waiting based on the models with a fixed number of mixture components, results from chain 1.

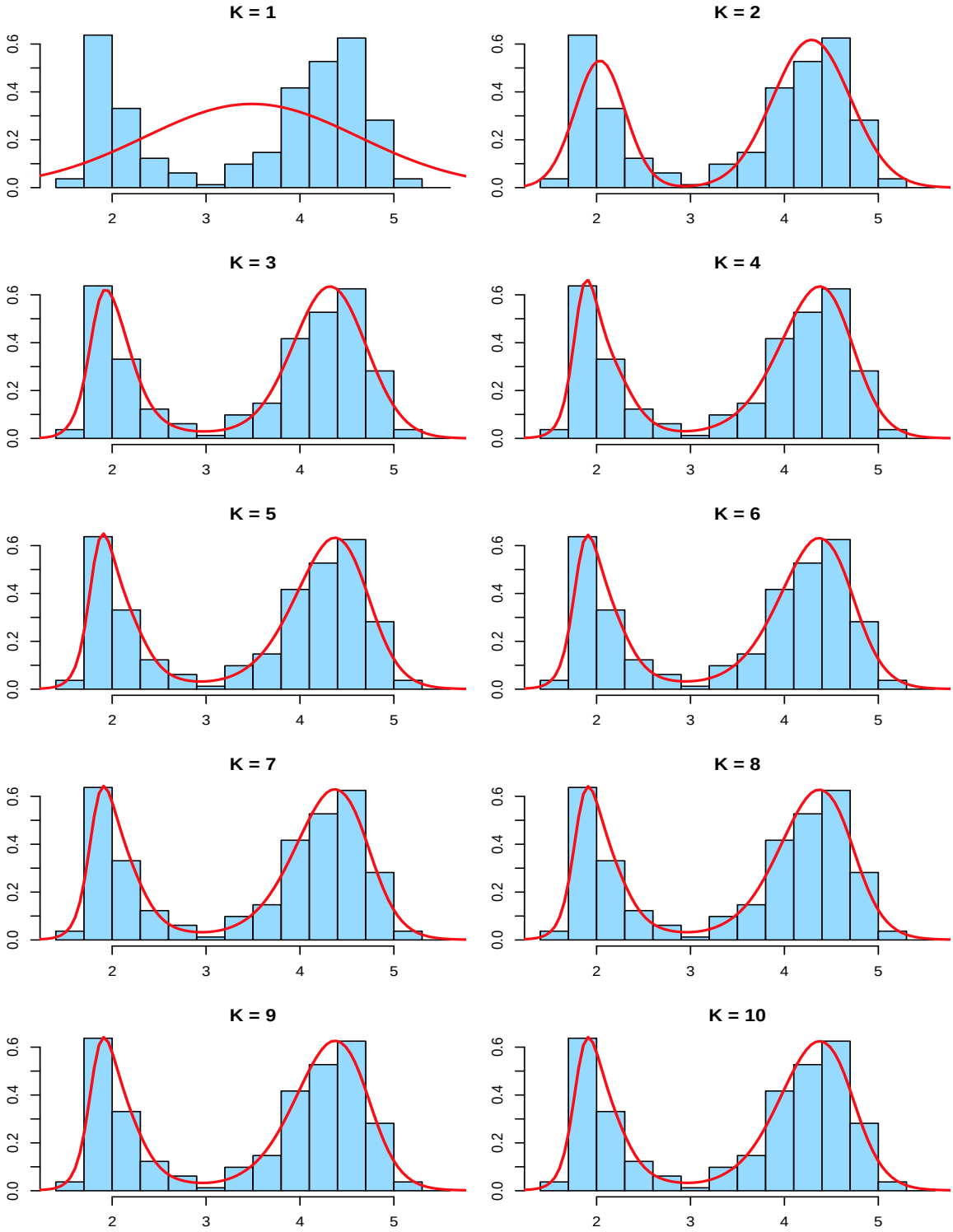

Figure 14: Marginal predictive density for eruptions based on the models with a fixed number of mixture components, results from chain 1.

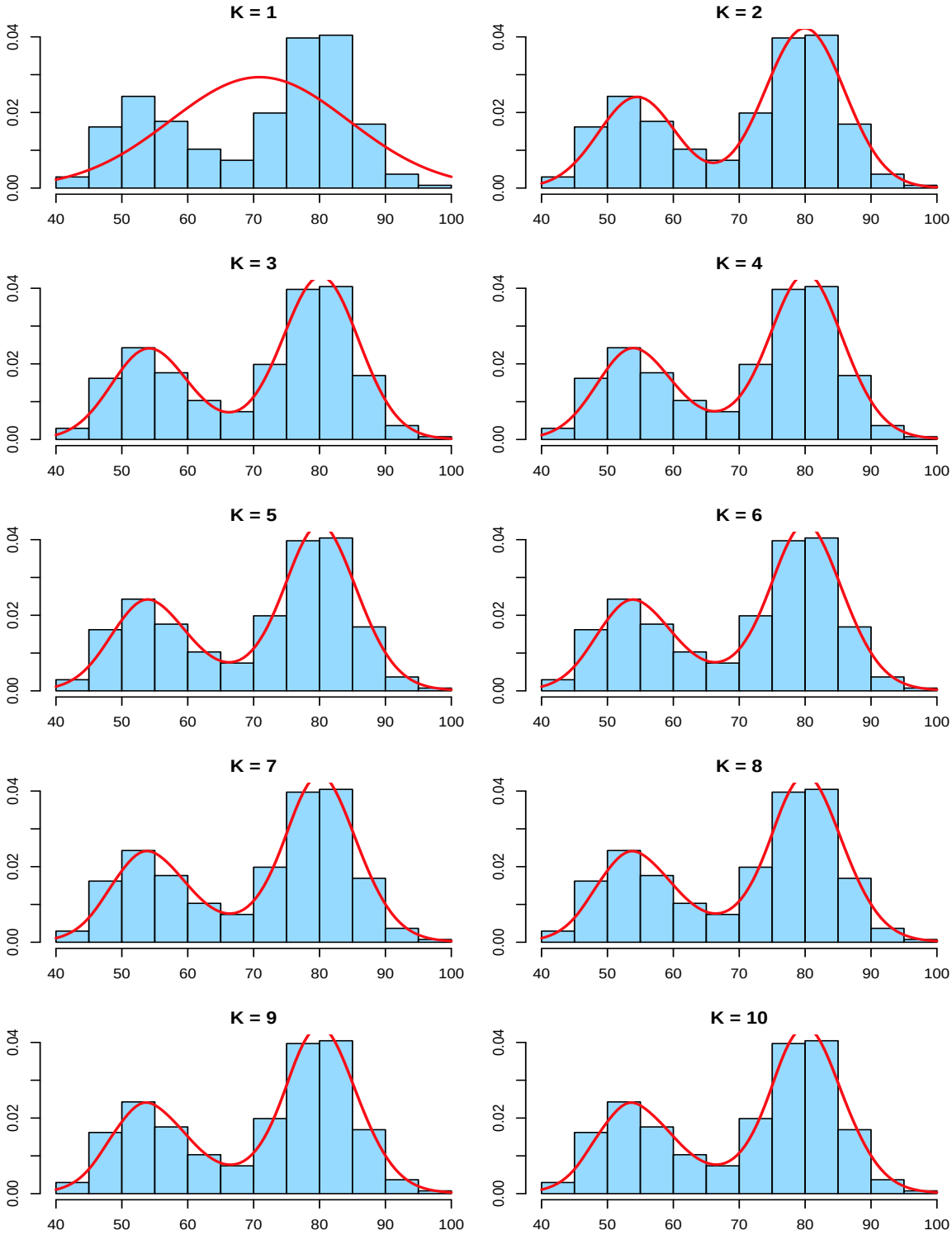

Figure 15: Marginal predictive density for waiting based on the models with a fixed number of mixture components, results from chain 1..

R⇒ Plots of the joint predictive density (contour plots) for different values of K, densities computed from chain 1 (see Figure 16):

```
> CH <- 1
> postscript(paste(FIGDIR, "figFaithful15.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mar=c(3, 2, 2, 1)+0.1)
> par(mfrow=c(5, 2), bty="n")
> for (k in 1:10){
+   plot(Faithful, col="red", xlab="", ylab="",
+        main=paste("K = ", k, sep=""))
+   contour(JPDensModelK[[k]][[CH]]$x$x1, JPDensModelK[[k]][[CH]]$x$x2,
+           JPDensModelK[[k]][[CH]]$dens[["1-2"]],
+           col="darkblue", add=TRUE)
+ }
> dev.off()
```

R⇒ Plots of the joint predictive density (image plots) for different values of K, densities computed from chain 1 (see Figure 17):

```
> CH <- 1
> postscript(paste(FIGDIR, "figFaithful19.ps", sep=""), width=7, height=10,
+            horizontal=FALSE)
> par(mar=c(3, 2, 2, 1)+0.1)
> par(mfrow=c(5, 2), bty="n")
> for (k in 1:10){
+   plot(Faithful, col="darkblue", xlab="", ylab="",
+        main=paste("K = ", k, sep=""))
+   image(JPDensModelK[[k]][[CH]]$x$x1, JPDensModelK[[k]][[CH]]$x$x2,
+         JPDensModelK[[k]][[CH]]$dens[["1-2"]], add=TRUE,
+         col=rev(heat_hcl(33, c=c(80, 30), l=c(30, 90), power=c(1/5, 1.3))))
+ #  points(Faithful, col="darkblue")
+ }
> dev.off()
```

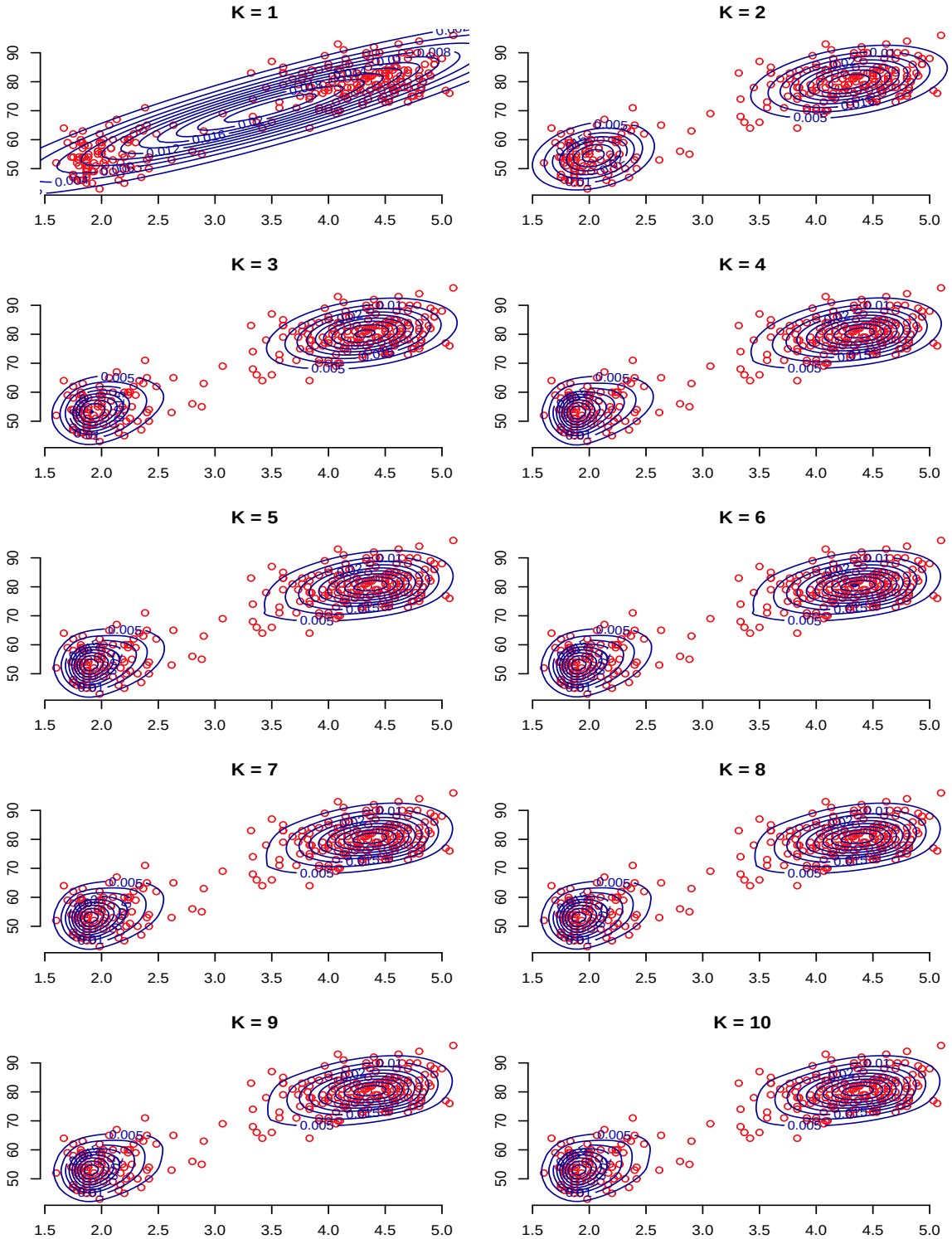

Figure 16: Joint predictive density based on the models with a fixed number of mixture components, results from chain 1.

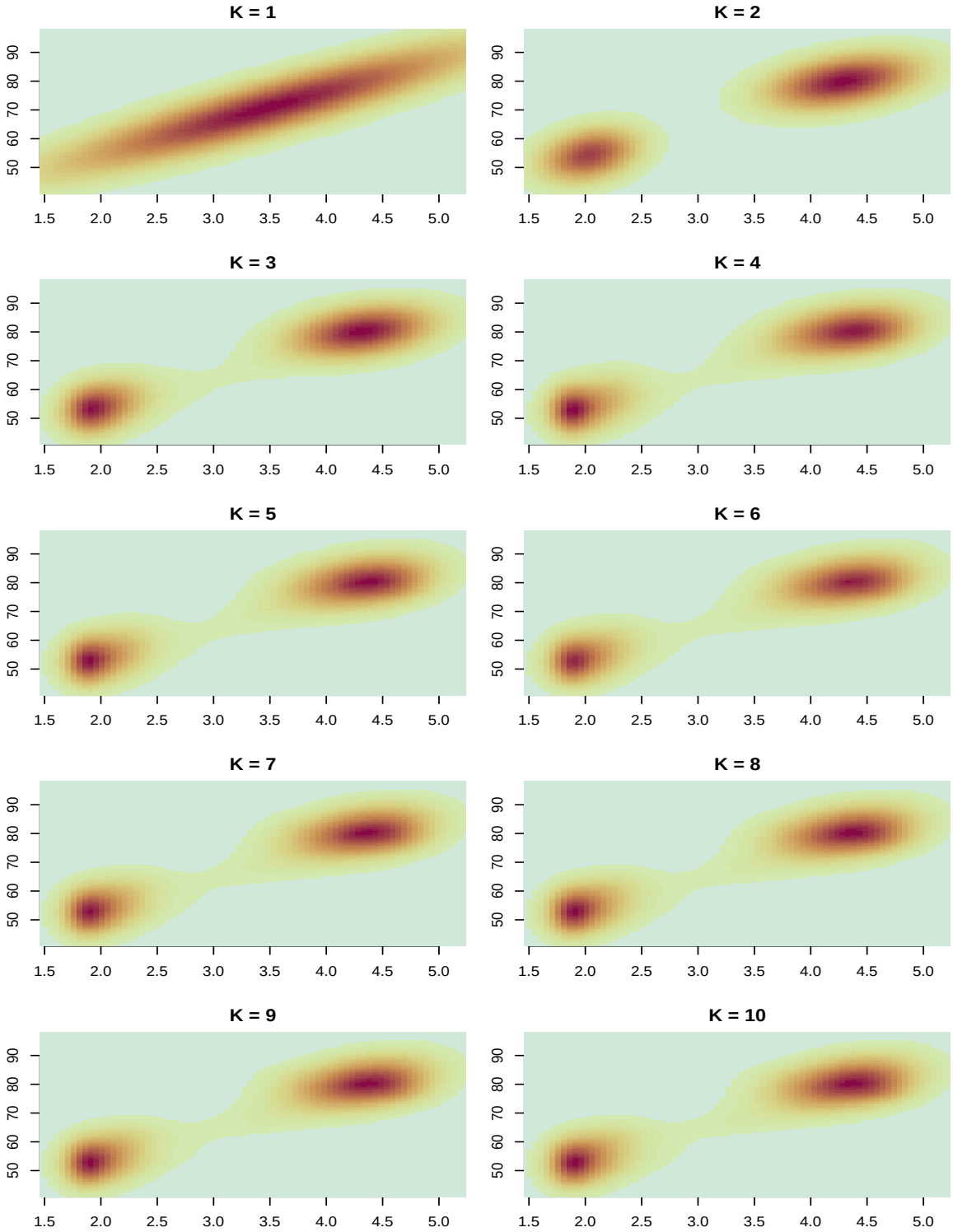

Figure 17: Joint predictive density based on the models with a fixed number of mixture components, results from chain 1.

R⇒ Save results for future use:

```
> if (RUN.TIMECONSUMING.CODE){
+   save(list="Model0",
+        file=paste(RESULTDIR, "/Faithful-Model0", Kshow, ".RData", sep=""))
+   save(list=c("ModelK", "MPDensModelK", "JPDensModelK"),
+        file=paste(RESULT2DIR, "/Faithful-Result.RData", sep=""))
+ }
```

References

Dellaportas P, Papageorgiou I (2006). "Multivariate mixtures of normals with unknown number of components." *Statistics and Computing*, **16**, 57–68.

Härdle W (1991). *Smoothing Techniques with Implementation in S*. Springer Verlag, New York. ISBN 978-0-387-97367-8.

Komárek A (2009). "A new R package for Bayesian estimation of multivariate normal mixtures allowing for selection of the number of components and interval-censored data." *Computational Statistics and Data Analysis*, **53**(12), 3932–3947. doi:10.1016/j.csda.2009.05.006.

Stephens M (2000). "Dealing with label switching in mixture models." *Journal of the Royal Statistical Society, Series B*, **62**, 795–809.

Affiliation:

Arnošt Komárek
Dept. of Probability and Mathematical Statistics
Faculty of Mathematics and Physics, Charles University in Prague
Sokolovská 83
186 75 Praha 8 – Karlín, Czech Republic
E-mail: Arnost.Komarek@mff.cuni.cz
URL: http://www.karlin.mff.cuni.cz/~komarek/