Bayesian Estimation Supersedes the t-Test

Mike Meredith and John Kruschke

June 1, 2013

1 Introduction

The BEST package provides a Bayesian alternative to a t test, providing much richer information about the samples and the difference in means than a simple p value.

Bayesian estimation for two groups provides complete distributions of credible values for the effect size, group means and their difference, standard deviations and their difference, and the normality of the data. For a single group, distributions for the mean, standard deviation and normality are provided. The method handles outliers.

The decision rule can accept the null value (unlike traditional t tests) when certainty in the estimate is high (unlike Bayesian model comparison using Bayes factors).

The package also provides methods to estimate statistical power for various research goals.

The code used for the computations is the same as that used in Kruschke (2013).

2 The Model

To accommodate outliers we describe the data with a distribution that has fatter tails than the normal distribution, namely the t distribution. (Note that we are using this as a convenient description of the data, not as a sampling distribution from which p values are derived.) The relative height of the tails of the t distribution is governed by the shape parameter ν: when ν is small, the distribution has heavy tails, and when it is large (e.g., 100), it is nearly normal. Here we refer to ν as the normality parameter.

The data (y) are assumed to be independent and identically distributed (i.i.d.) draws from a t distribution with different mean (μ) and standard deviation (σ) for each population, and with a common normality parameter (ν), as indicated in the lower portion of Figure 1.

The priors used are minimally informative: normal priors with large standard deviation for (μ), broad uniform priors for (σ), and a shifted-exponential prior for (ν), as shown in the upper part of Figure 1. Full details of the priors are given in Kruschke (2013).

For a general discussion see chapters 11 and 12 of Kruschke (2011).

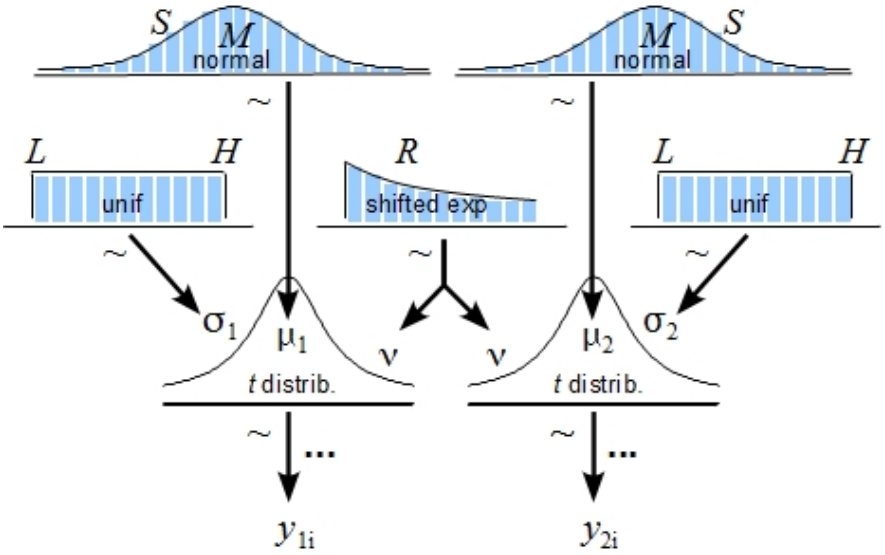

Figure 1: *Hierarchical diagram of the descriptive model for robust Bayesian estimation.*

3 Preparing to run BEST

BEST uses the JAGS package (Plummer, 2003) to produce samples from the posterior distribution of each parameter of interest. You will need to download JAGS from `http://sourceforge.net/projects/mcmc-jags/` and install it before running BEST.

BEST also requires the packages `rjags` and `coda`, which should normally be installed at the same time as package BEST if you use the `install.packages` function in R.

Once installed, we need to load the BEST package at the start of each R session, which will also load rjags and coda and link to JAGS:

```
> library(BEST)

Loading required package: rjags
Loading required package: coda
Loading required package: lattice
Linked to JAGS 3.3.0
Loaded modules: basemod,bugs
```

4 An example with two groups

4.1 Some example data

We will use the hypothetical data from Kruschke (2013): "Consider data from two groups of people who take an IQ test. Group 1 ($N_1 = 47$) consumes a "smart drug" while Group 2 ($N_2 = 42$) is a control group that consumes a placebo."

```
> y1 = c(101,100,102,104,102,97,105,105,98,101,100,123,105,103,100,95,102,106,
         109,102,82,102,100,102,102,101,102,102,103,103,97,97,103,101,97,104,
         96,103,124,101,101,100,101,101,104,100,101)
> y2 = c(99,101,100,101,102,100,97,101,104,101,102,102,100,105,88,101,100,
         104,100,100,100,101,102,103,97,101,101,100,101,99,101,100,100,
         101,100,99,101,100,102,99,100,99)
```

4.2 Running the model

We run BESTmcmc and save the result in BESTout. This will take several minutes:

```
> BESTout <- BESTmcmc(y1, y2)
```

```
Setting up the JAGS model...
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 197
Initializing model
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100%
Burning in the MCMC chain...
  |**************************************************| 100%
Sampling final MCMC chain...
  |**************************************************| 100%
```

4.3 Basic inferences

The default plot (Figure 2) is a histogram of the posterior distribution of the difference in means.

```
> plot(BESTout)
```

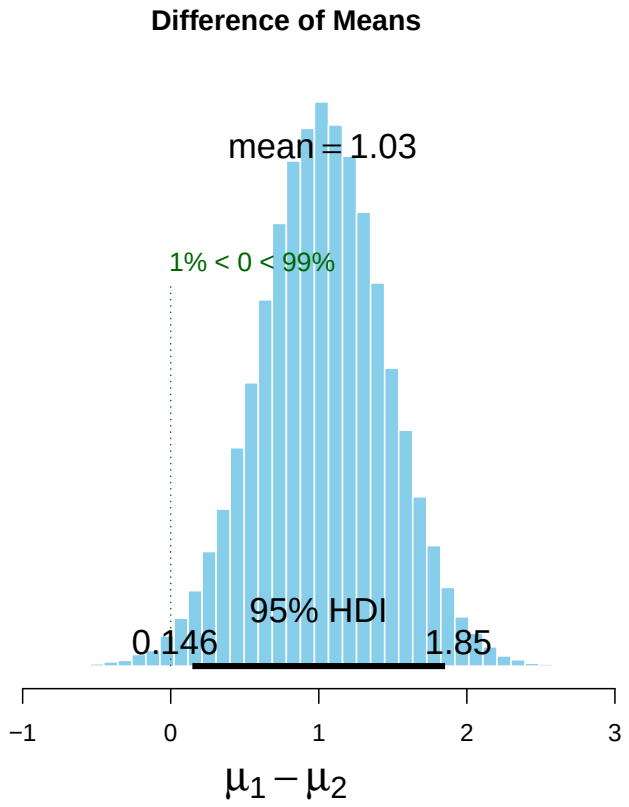

Figure 2: *Default plot: posterior probability of the difference in means.*

Also shown is the mean of the posterior probability, which is an appropriate point estimate of the true difference in means, the 95% Highest Density Interval (HDI), and the posterior probability that the difference is greater than zero. The 95% HDI does not include zero, and the probability that the true value is greater than zero is shown as 99%. Compare this with the output from a t test:

```
> t.test(y1, y2)

        Welch Two Sample t-test

data:  y1 and y2
t = 1.6222, df = 63.039, p-value = 0.1098
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3611848  3.4766863
sample estimates:
mean of x mean of y
 101.9149  100.3571
```

Because we are dealing with a Bayesian posterior probability distribution, we can extract much more information:

- We can estimate the probability that the true difference in means is above (or below) an arbitrary *comparison value*. For example, the drug is claimed to increase IQ scores by at least 1.5 units.
- The probability that the difference in yields is precisely zero is zero. More interesting is the probability that the difference may be too small to matter. We can define a *region of practical equivalence* (ROPE) around zero, and obtain the probability that the true value lies therein. For the IQ example, a difference of $\pm$ 0.1 may be too small to matter.

```
> plot(BESTout, compVal=1.5, ROPE=c(-0.1,0.1))
```

The annotations in (Figure 3) show that the probability that the increased IQ is > 1.5 is not plausible. In this case it's clear that the effect is large, but if most of the probability mass (say, 95%) lay within the ROPE, we would accept the null value for practical purposes.

BEST deals appropriately with differences in standard deviations between the samples and departures from normality due to outliers. We can check the difference in standard deviations or the normality parameter with `plot` (Figure 4).

```
> par(mfrow=1:2)
> plot(BESTout, which="sd")
> plot(BESTout, which="nu")
```

The `summary` method gives us more information on the parameters of interest, including derived parameters:

```
> summary(BESTout)

             mean  median    mode HDI%    HDIlo   HDIup compVal %>compVal
mu1       101.550 101.550 101.552   95 100.8003 102.300
mu2       100.524 100.523 100.533   95 100.1057 100.938
muDiff      1.025   1.025   1.009   95   0.1462   1.853       0      99.0
sigma1      2.058   2.018   1.954   95   1.2610   2.912
```

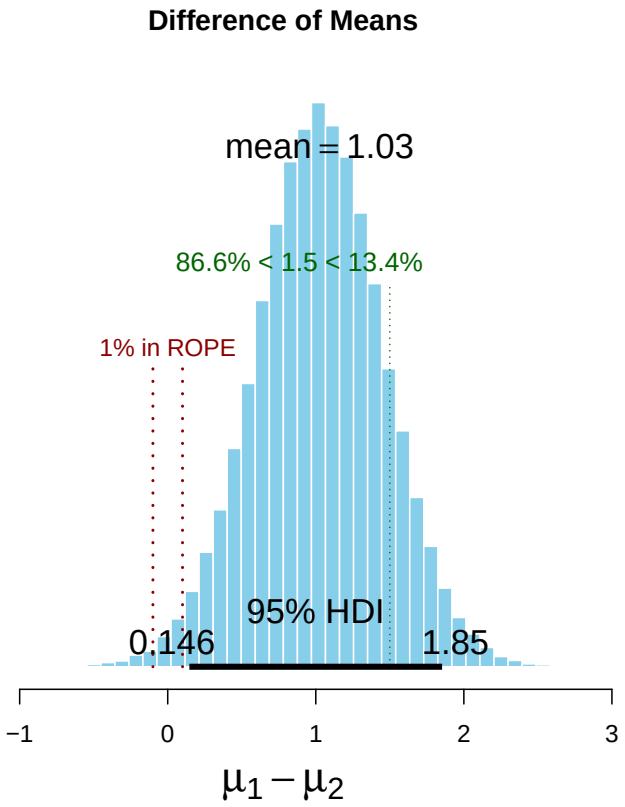

Figure 3: *Posterior probability of the difference in means with compVal=1.5 and ROPE ± 0.1.*

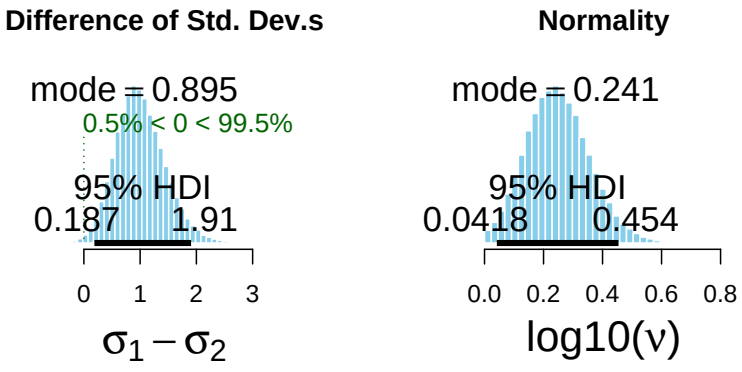

Figure 4: *Posterior plots for difference in standard deviation and for the normality parameter.*

```
sigma2      1.049   1.030   0.985   95   0.6653   1.456
sigmaDiff   1.009   0.979   0.895   95   0.1869   1.906       0      99.5
nu          1.831   1.756   1.593   95   1.0343   2.733
log10nu     0.249   0.244   0.241   95   0.0418   0.454
effSz       0.646   0.637   0.616   95   0.0692   1.220       0      99.0
```

Here we have summaries of posterior distributions for the derived parameters: difference in means (`muDiff`), difference in standard deviations (`sigmaDiff`) and effect size (`effSz`). As with the plot command, we can set values for `compVal` and `ROPE` for each of the parameters of interest:

```
> summary(BESTout, credMass=0.8, ROPEm=c(-0.1,0.1), ROPEsd=c(-0.15,0.15),
            compValeff=1)

             mean  median    mode HDI%   HDIlo   HDIup compVal %>compVal
mu1       101.550 101.550 101.552   80 101.061 102.020
mu2       100.524 100.523 100.533   80 100.247 100.787
muDiff      1.025   1.025   1.009   80   0.471   1.571       0      99.0
sigma1      2.058   2.018   1.954   80   1.463   2.531
sigma2      1.049   1.030   0.985   80   0.777   1.285
sigmaDiff   1.009   0.979   0.895   80   0.430   1.528       0      99.5
nu          1.831   1.756   1.593   80   1.196   2.276
log10nu     0.249   0.244   0.241   80   0.108   0.381
effSz       0.646   0.637   0.616   80   0.271   1.018       1      11.6
          ROPElow ROPEhigh %InROPE
mu1
mu2
muDiff      -0.10     0.10    1.14
sigma1
sigma2
sigmaDiff   -0.15     0.15    1.42
nu
log10nu
effSz
```

4.4 Checking convergence and fit

The output from `BESTmcmc` has class BEST, which has a `print` method:

```
> class(BESTout)

[1] "BEST"      "mcmc.list"

> print(BESTout)

MCMC fit results for BEST analysis:
100002 simulations saved.
            mean     sd  median    HDIlo   HDIup Rhat n.eff
mu[1]    101.550 0.3796 101.550 100.8003 102.300    1 62979
mu[2]    100.524 0.2121 100.523 100.1057 100.938    1 63154
nu         1.831 0.4750   1.756   1.0343   2.733    1 23421
sigma[1]   2.058 0.4312   2.018   1.2610   2.912    1 29573
sigma[2]   1.049 0.2064   1.030   0.6653   1.456    1 33182
```

```
'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
'Rhat' is the potential scale reduction factor (at convergence, Rhat=1).
'n.eff' is a crude measure of effective sample size.
```

This shows the mean, standard deviation and median of the posterior distributions of the parameters in the model, together with a 95% Highest Density Interval: see the help page for the `hdi` function for details. Two convergence diagnostic measures are also displayed:

- `Rhat` is the Brooks-Gelman-Rubin scale reduction factor, which is 1 on convergence. Gelman and Shirley (2011) consider values below 1.1 to be acceptable. Increase the `burnInSteps` argument to `BESTmcmc` if any of the `Rhat`s are too big.
- `n.eff` is the effective sample size, which is less than the number of simulations because of autocorrelation between successive values in the sample. Values of `n.eff` around 10,000 are needed for stable estimates of 95% credible intervals.[1] If any of the values is too small, you can increase the `numSavedSteps` or `thinSteps` arguments.

See the help pages for the `coda` package for more information. With the current version of the `BEST` package, any of the diagnostic tests in `coda` can be used with `BESTmcmc` output; this may change in future versions.

As a further check, we can compare *posterior predictive distributions* with the original data:

```
> plotPostPred(BESTout)
```

Each panel of Figure 5 corresponds to one of the samples, and shows curves produced by selecting 30 random steps in the MCMC chain and plotting the t distibution with parameters (μ), (σ) and (ν) for that step. Also shown is a histogram of the actual data. We can visually assess whether the model is a reasonably good fit to the sample data.

The function `plotAll` puts histograms of all the posterior distributions and the posterior predictive plots onto a single page (Figure 6).

```
> plotAll(BESTout)
```

4.5 Working with individual parameters

Objects of class `BEST` contain long vectors of simulated draws from the posterior distribution of each of the parameters in the model. To access these values, we attach the `BEST` object to the search path with `attachBEST`:

```
> attachBEST(BESTout)

You can now access mu1, mu2, nu, sigma1, sigma2 by name.

> length(nu)

[1] 100002

> meanDiff <- (mu1 - mu2)
> meanDiffGTzero <- mean(meanDiff > 0)
> meanDiffGTzero

[1] 0.9899302
```

[1] See `http://doingbayesiandataanalysis.blogspot.com/2011/07/how-long-should-mcmc-chain-be-to-get.html` for some simulation results.

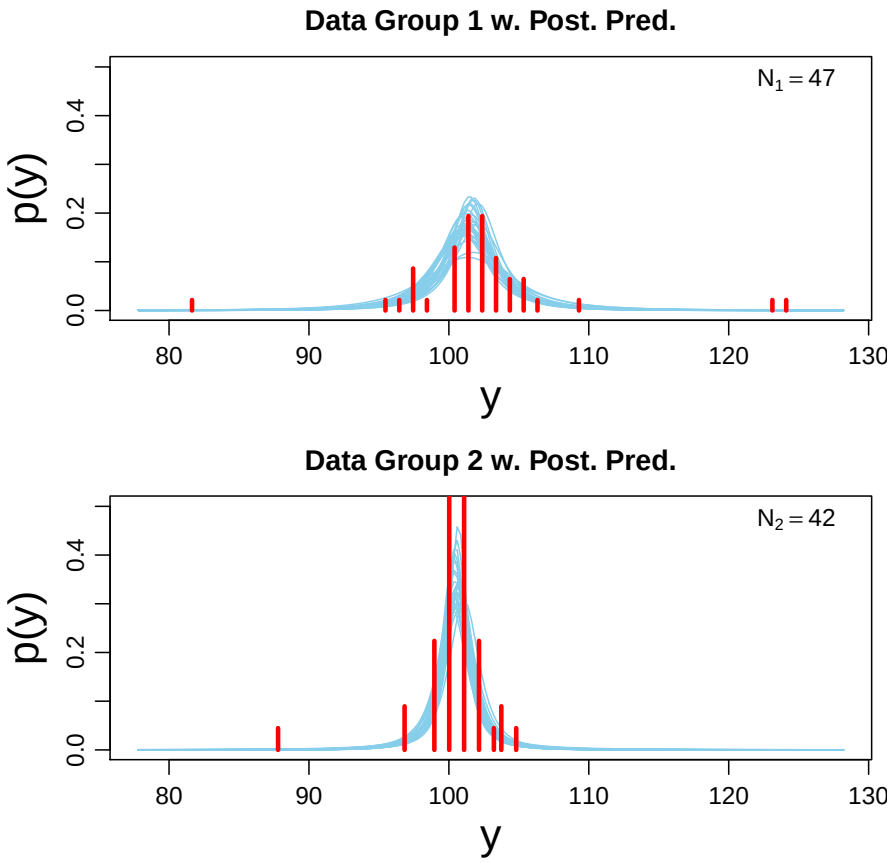

Figure 5: *Posterior predictive plots together with a histogram of the data.*

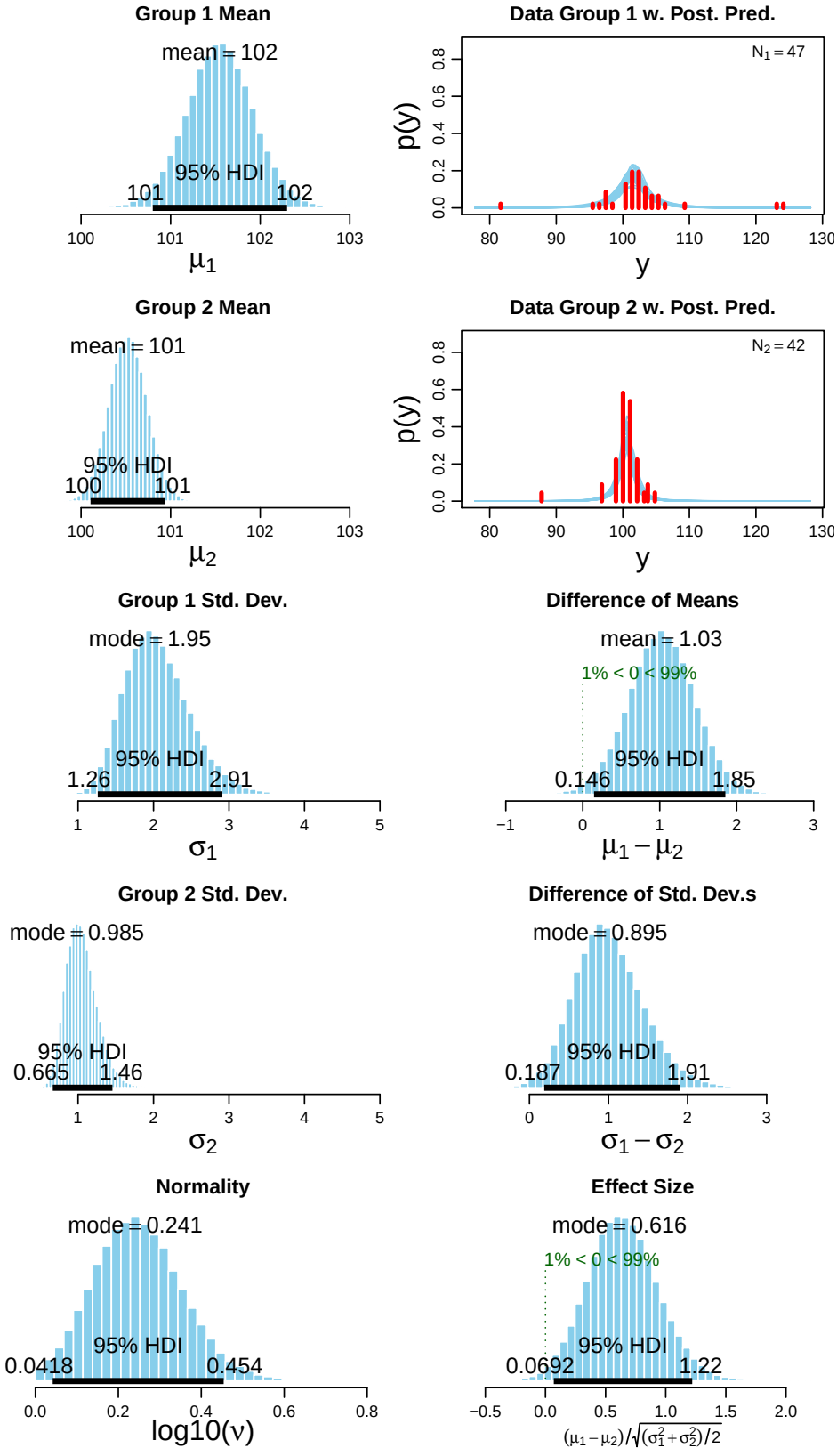

Figure 6: *All the posterior distributions and the posterior predictive plots.*

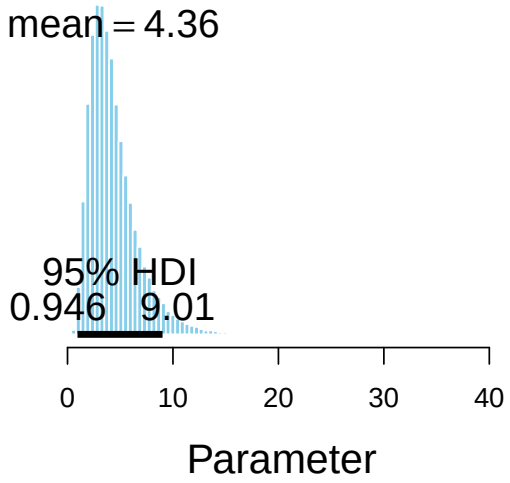

Figure 7: *Posterior distribution of the ratio of the sample variances.*

For example, you may wish to look at the ratio of the variances rather than the difference in the standard deviations. You can calculate a vector of draws from the posterior distribution, calculate summary statistics, and plot the distribution with `plotPost` (Figure 7):

```
> varRatio <- sigma1^2 / sigma2^2
> median(varRatio)

[1] 3.845709

> hdi(varRatio)

    lower     upper
0.9456415 9.0113173
attr(,"credMass")
[1] 0.95

> mean(varRatio > 1)

[1] 0.9948201

> plotPost(varRatio)
```

5 An example with a single group

Applying BEST to a single sample, or for differences in paired observations, works in much the same way as the two-sample method and uses the same function calls. To run the model, simply use `BESTmcmc` with only one vector of observations.

```
> y0 <- c(1.89, 1.78, 1.30, 1.74, 1.33, 0.89)
> BESTout1g <- BESTmcmc(y0)
```

```
Setting up the JAGS model...
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 21

Initializing model

  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100%
Burning in the MCMC chain...
  |**************************************************| 100%
Sampling final MCMC chain...
  |**************************************************| 100%
```

This time we have a single mean and standard deviation. The default plot (Figure 8) shows the posterior distribution of the mean.

```
> BESTout1g

MCMC fit results for BEST analysis:
100002 simulations saved.
         mean      sd  median  HDIlo  HDIup  Rhat n.eff
mu     1.4954  0.2401  1.4955 1.0166  1.971 1.000 45706
nu    31.9751 29.0148 23.4600 1.0010 89.833 1.001 20945
sigma  0.5178  0.2799  0.4518 0.1867  1.014 1.001 10624

'HDIlo' and 'HDIup' are the limits of a 95% HDI credible interval.
'Rhat' is the potential scale reduction factor (at convergence, Rhat=1).
'n.eff' is a crude measure of effective sample size.

> plot(BESTout1g)
```

Standard deviation, the normality parameter and effect size can be plotted individually, or on a single page with `plotAll` (Figure 9).

```
> plotAll(BESTout1g)
```

And we can access the draws from the posterior distributions after attaching them to the search path:

```
> attachBEST(BESTout1g)

You can now access mu, nu, sigma by name.

> length(nu)

[1] 100002

> variance <- sigma^2
> plotPost(variance, xlim=c(0, 3))
> detachBEST()
```

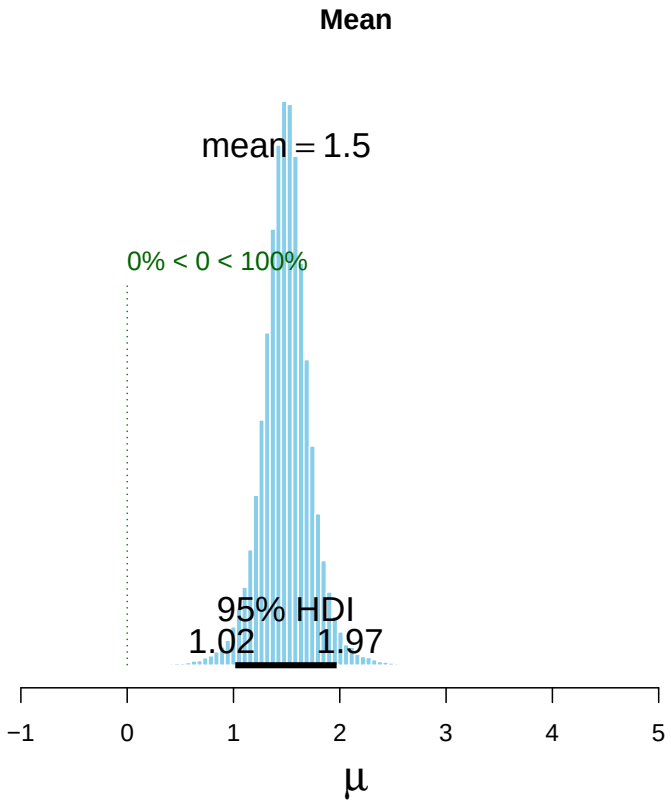

Figure 8: *Default plot: posterior probability distribution for the mean.*

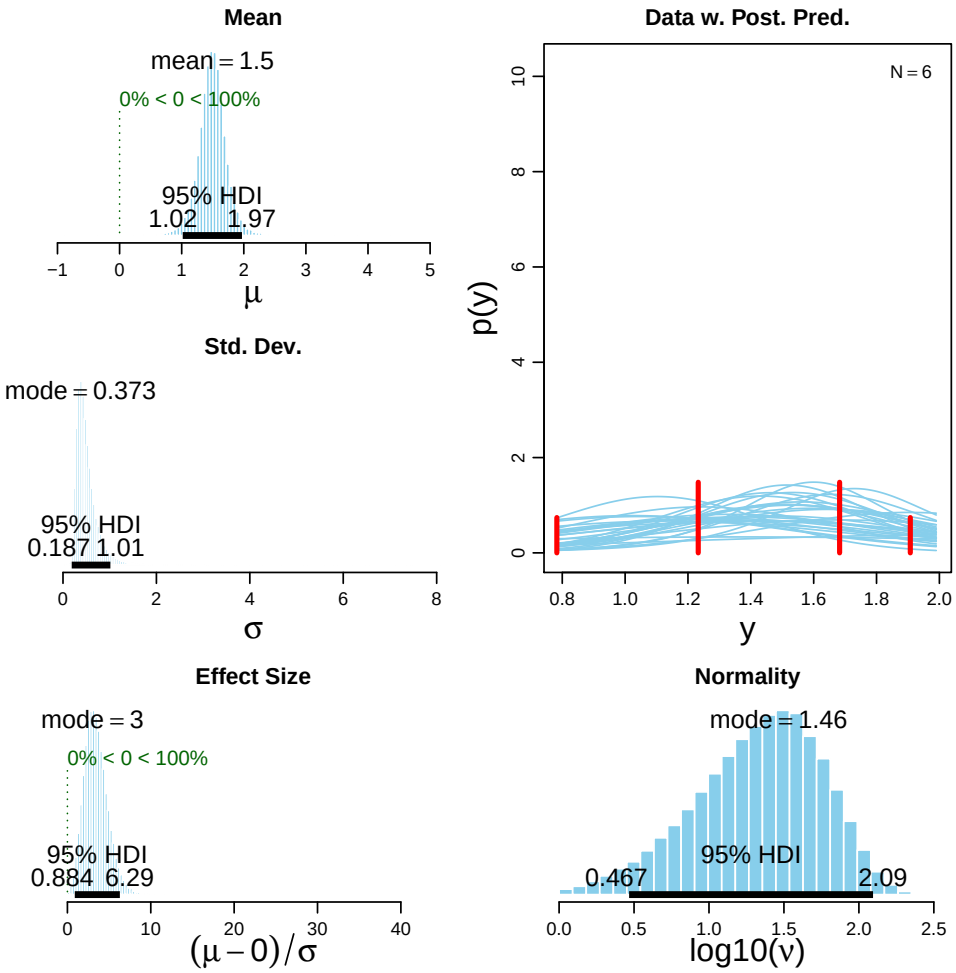

Figure 9: *All the posterior distributions and the posterior predictive plots.*

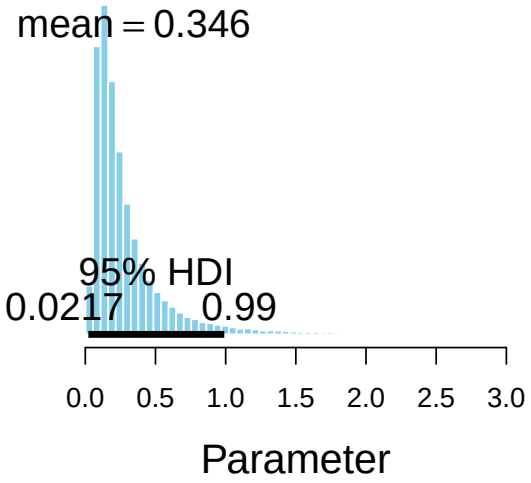

Figure 10: *Posterior distribution of the sample variance.*

6 Power analysis

Power analysis seeks to answer the question: "What is the probability that I will meet my research goals with my intended study design?" Research goals here may be:

- precision: the width of the HDI is less than a predetermined criterion;
- accept the null: the HDI falls entirely within the ROPE;
- reject the null: the HDI falls entirely outside the ROPE, either above or below it.

Of course, success depends on the real state of the populations studied, and assumptions about means, standard deviations, and normality are unavoidable. For a *retrospective* power analysis, we assume that the estimates derived from a past study are correct; for a *prospective* power analysis, we provide a fresh set of assumed values.

The approach used in BEST recognises uncertainty about the parameter values, and generates simulated samples based on plausible sets of values. The simulated samples are analysed using `BESTmcmc` and a tally kept of the number of simulations in which the research criteria were satisfied. The tally is used to calculate the posterior probability of meeting the criterion, and the mean and 95% HDI of the posterior are returned.

6.1 Retrospective power analysis

Retrospective power analysis makes use of the output from a previous study. We will continue the example from Section 4. For the difference in means, we want the 95% HDI width to be less than 2 tonnes/ha and set the ROPE at ± 0.1. By default, sample sizes match those of the original study. Running 200 simulations (the default) takes about 30 mins.

```
> powerRet1 <- BESTpower(BESTout, ROPEm = c(-0.1,0.1), maxHDIWm = 2.0,
                saveName = NULL)
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
Power computation: Simulated Experiment 1 of 200 :

Setting up the JAGS model...
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 197
...
After 200 Simulated Experiments, Posterior Probability
       of meeting each criterion is (mean and 95% CrI):
                       mean CrIlo CrIhi
  mean:   HDI > ROPE 0.446 0.377 0.514
  mean:   HDI < ROPE 0.005 0.000 0.015
  mean:  HDI in ROPE 0.005 0.000 0.015
  mean: HDI width ok 0.589 0.521 0.656
```

We did not set criteria for standard deviation or effect size, so only the results for the mean are displayed. It is possible to plot the results for the first few simulations (cf. Figure 6) by setting the argument `showFirstNrep` to a positive integer (*Note for Mac users*: this uses X11 which is no longer included in Mac OS X 10.8. The first time you try to run it, you will be prompted to install XQuartz; it should work thereafter.)

Retrospective power analysis for the single-sample case is implemented in the same way, but the criteria now apply to the mean or standard deviation of the sample rather than the difference between samples.

6.2 Prospective power analysis

This is implemented as a three-step process. For a discussion of the advantages of this procedure, see Kruschke (2013).

6.2.1 Generate an idealized data set

The first step is to specify the means and standard deviations of two normally distributed groups, and the sample size for each group. We also specify the percentage of the simulated data that should be outliers, and the (larger) standard deviation for the outliers.

With these inputs, the function `makeData` generates (and plots) an ideal data set matching our specification:

```
> proData <- makeData(mu1=108, sd1=17, mu2=100, sd2=15, nPerGrp=20,
                      pcntOut=10, sdOutMult=2.0)
```

6.2.2 Generate plausible parameter values

Next we generate sets of parameter values which are plausible in view of our ideal data set. This is accomplished by `BESTmcmc`. We plan to do 200 simulations; since values tend to be highly correlated, we generate 2000 and will select 200 from these. Figure 12 summarized the distributions of the parameters we will use.

```
> proMCMC <- BESTmcmc(proData$y1, proData$y2, numSavedSteps=2000)
> plotAll(proMCMC)
```

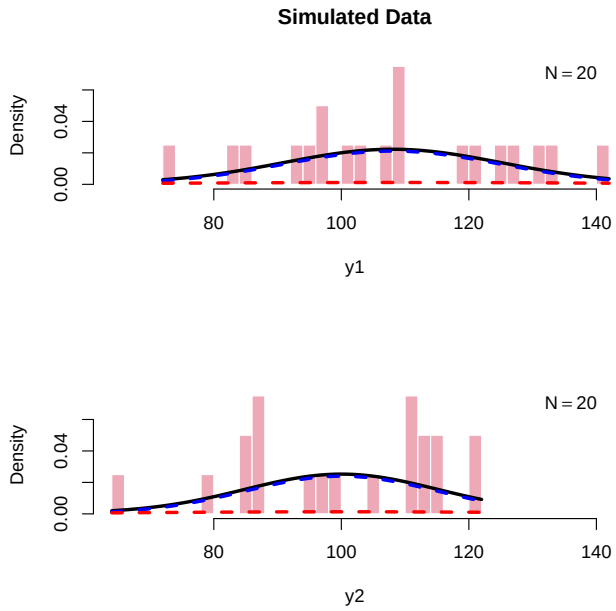

Figure 11: *Distribution of the idealised samples for each group (histograms), together with the true distributions of the outliers (red curve) and non-outliers (blue curve). The solid black curve is the combined distribution.*

```
Setting up the JAGS model...
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
   Graph Size: 99

Initializing model
  |++++++++++++++++++++++++++++++++++++++++++++++++++| 100%
Burning in the MCMC chain...
  |**************************************************| 100%
Sampling final MCMC chain...
  |**************************************************| 100%
```

6.2.3 Simulate and analyse the data sets

Finally, we simulate samples based on plausible parameter values and tally the number of times when our research criteria are met, much as we did with the retrospective power analysis.

`BESTpower` allows for the number of observations to vary across simulations. Suppose the number of volunteer subjects varies but averages around 100; we divide the number we have into two approximately equal groups. With 300 replicates, this takes over 3 hrs.

```
> nReplicates <- 200
> Nsubj <- rpois(nReplicates, 100)
> N1plan <- round(Nsubj / 2)
> N2plan <- Nsubj - N1plan
> powerPro <- BESTpower(proMCMC, N1=N1plan, N2=N2plan,
```

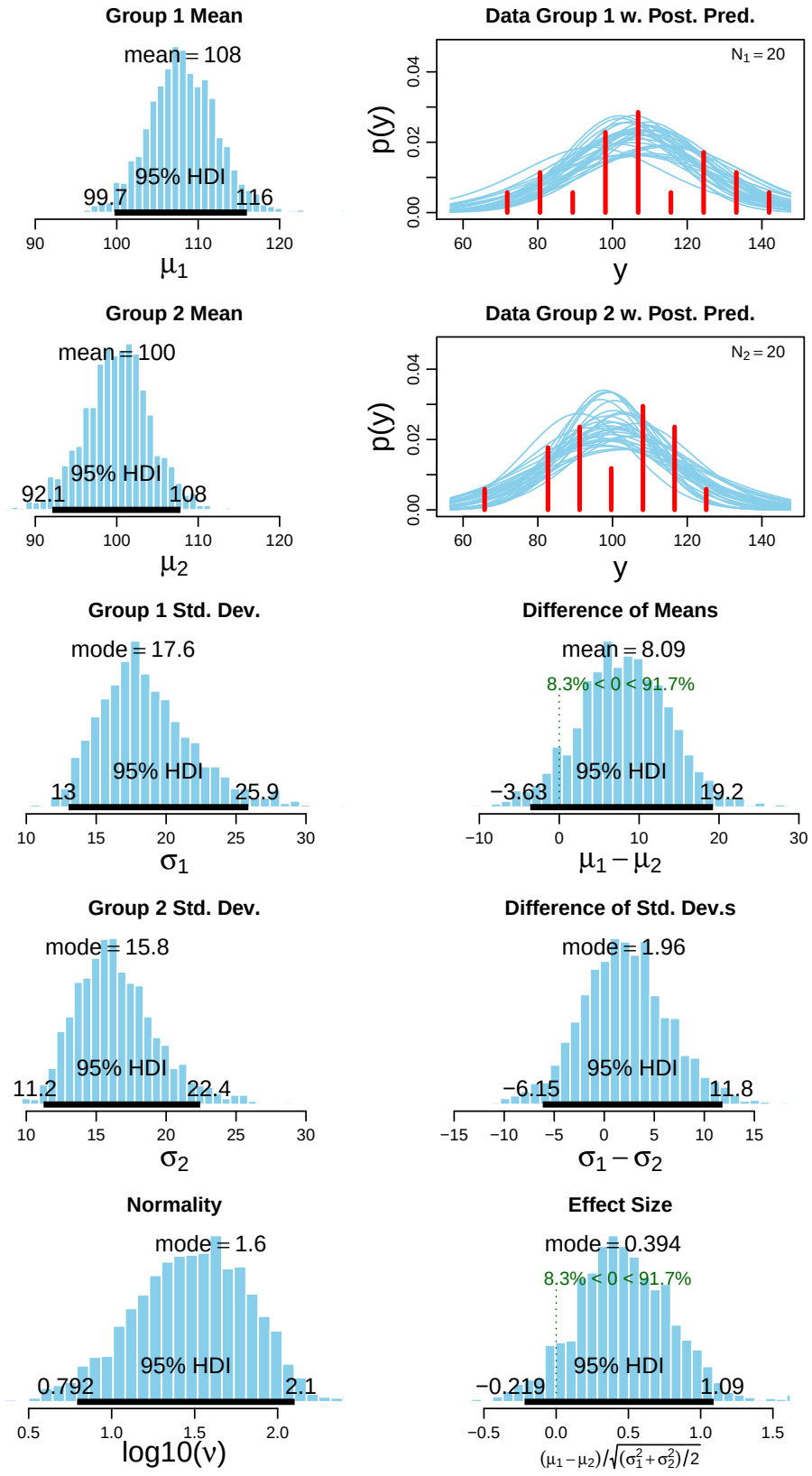

Figure 12: *Summary plot of the parameter values to use for the power analysis.*

```
                    ROPEm=c(-1.5,1.5), ROPEsd=c(-2,2), ROPEeff=c(-0.5,0.5),
                    maxHDIWm=15.0, maxHDIWsd=10.0, maxHDIWeff=1.0,
                    nRep=nReplicates)

...
After 200 Simulated Experiments, Posterior Probability
       of meeting each criterion is (mean and 95% CrI):
                        mean CrIlo CrIhi
  mean:   HDI > ROPE 0.406 0.339 0.474
  mean:   HDI < ROPE 0.010 0.000 0.023
  mean:  HDI in ROPE 0.005 0.000 0.015
  mean: HDI width ok 0.495 0.426 0.564
    sd:   HDI > ROPE 0.233 0.175 0.291
    sd:   HDI < ROPE 0.030 0.009 0.053
    sd:  HDI in ROPE 0.005 0.000 0.015
    sd: HDI width ok 0.223 0.167 0.281
effect:   HDI > ROPE 0.089 0.052 0.129
effect:   HDI < ROPE 0.005 0.000 0.015
effect:  HDI in ROPE 0.074 0.040 0.111
effect: HDI width ok 0.975 0.954 0.994
```

Power analysis for the single-sample case is analogous: provide values for arguments `mu1` and `sd1` in `makeData`, and put `mu1=NULL` and `sd1=NULL`.

7 What next?

If you want to know the details of how the functions in the `BEST` package work, you can download the R source code from GitHub `https://github.com/mikemeredith/BEST` or find almost the same code at `http://www.indiana.edu/~kruschke/BEST/` together with links to articles, videos, and the blog.

Bayesian analysis with computations performed by JAGS is a powerful approach to analysis. For a practical introduction see Kruschke (2011).

8 References

Gelman A, Shirley K (2011). "Inference from simulations and monitoring convergence." In S Brooks, A Gelman, G Jones, XL Meng (eds.), *Handbook of Markov chain Monte Carlo*, pp. 163–174. Chapman & Hall.

Kruschke JK (2011). *Doing Bayesian data analysis: a tutorial with R and BUGS*. Elsevier, Amsterdam etc.

Kruschke JK (2013). "Bayesian estimation supersedes the t test." *Journal of Experimental Psychology: General*, **142**(2), 573–603.

Plummer M (2003). "JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling." In *3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Vienna, Austria.