How to use the R code that accompanies "Quantifying sources of uncertainty in projections of future climate"

Paul Northrop and Richard Chandler

June 4, 2015

1 Required software

R is a free software environment for statistical computing and graphics. It can be installed from `www.http://cran.r-project.org/`.

Our code uses the R packages lme4 (Bates and Maechler, 2009) `http://cran.r-project.org/web/packages/lme4/index.html` and arm (Gelman et al., 2010) `http://cran.r-project.org/web/packages/arm/index.html`. The latter uses the package R2WinBUGS (Sturtz et al., 2005) `http://cran.r-project.org/web/packages/R2WinBUGS/index.html` to run WinBUGS from R. The WinBUGS software performs Bayesian analyses using Markov chain Monte Carlo (MCMC) methods. It can be installed from `http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml`.

The R commands used below are given in the file `examplecode.R`.

2 Getting started

The R package arm and WinBUGS must be installed first. Then we

- load the arm package (which automatically loads the lm4 package) and R2WinBUGS packages;
- set the working directory to one containing the files `examplecode.fns` and `superandfinite.bug` and the data files `tempdata.Rdata` and `raindata.Rdata`;
- read in the functions contained in `examplecode.fns`.

```
> library(arm)
> library(R2WinBUGS)
> setwd("Z:/CLIMATE/IMSC/FOR_REC")
> # input functions MCMC.half.Cauchy.finite(), two.way.sim() and REML.results()
> source("examplecode.fns")
```

3 Analyses of temperature indices

We load and inspect the temperature data. The value of the index is named `y`.

```
> load(file="tempdata.Rdata") # load the temperature data
> summary(all.temp.data) # see which variables are present
```

```
       y          scenario                GCM            run           region       period
 Min.   :0.1133   A1B:2438   ncar_ccsm3_0   : 920   run1   :2806   Min.   : 0   Min.   :1.0
 1st Qu.:1.1145   A2 :1656   cccma_cgcm3_1  : 690   run2   :1104   1st Qu.: 5   1st Qu.:1.0
```

```
 Median :1.6957   B1 :2024   mri_cgcm2_3_2a : 690   run3   :1012   Median :11   Median :1.5
 Mean   :1.9993              mpi_echam5     : 460   run4   : 460   Mean   :11   Mean   :1.5
 3rd Qu.:2.5819              miroc3_2_medres: 414   run5   : 460   3rd Qu.:17   3rd Qu.:2.0
 Max.   :8.5460              miub_echo_g    : 414   run6   :  92   Max.   :22   Max.   :2.0
                             (Other)        :2530   (Other): 184
```

```
> all.temp.data[1:10,]   # look at first 10 rows of data
```

```
           y scenario               GCM  run region period
1  0.8937861      A1B       bccr_bcm2_0 run1      0      1
2  1.1928243      A1B     cccma_cgcm3_1 run1      0      1
3  1.2127593      A1B     cccma_cgcm3_1 run2      0      1
4  1.2114399      A1B     cccma_cgcm3_1 run3      0      1
5  1.1877431      A1B     cccma_cgcm3_1 run4      0      1
6  1.2191221      A1B     cccma_cgcm3_1 run5      0      1
7  1.4497154      A1B cccma_cgcm3_1_t63 run1      0      1
8  1.2101686      A1B          cnrm_cm3 run1      0      1
9  0.6423304      A1B       csiro_mk3_0 run1      0      1
10 1.2490990      A1B       csiro_mk3_5 run1      0      1
```

We select the period (1 for 2020–2049, 2 for 2069–2098) and region (0 for global, 1 for region 1 etc.) and extract the data for this combination. Here we analyse global temperature indices for period 1 (2020–2049).

```
> which.period <- 1          # 1 for 2020-2049, 2 for 2069-2098
> which.reg <- 0             # 0 for global, 1 for region 1 etc.
> cat(paste("region",which.reg,", period",which.period),"\n")
```

```
region 0 , period 1
```

```
> cond <- which(all.temp.data$region==which.reg & all.temp.data$period==which.period)
> temp <- all.temp.data[cond,]                                    # extract relevant data
```

3.1 Analysis using REML

We use the function `lmer` in the package lme4. The function `REML.results` extracts the REML estimates of the fixed effect μ and the random effects $\sigma_G, \sigma_S, \sigma_{GS}$ and σ_R. Setting the argument `CI` to `TRUE` means that simulation is used to estimate confidence intervals for the parameters. The argument `conf` (default 95) gives the desired confidence level and `n.sim` (default 1000) the number of simulations on the which the intervals are based. After every hundredth simulation the simulation number is printed to the screen. There is an option (`my.seed`) to specify a user-defined random number seed.

Firstly, we calcute estimates only (no confidence intervals).

```
> fit.REML <- lmer(y~1+ (1|scenario)+(1|GCM)+(1|factor(scenario):factor(GCM)), data=temp, REML=T)
> # no confidence intervals (CI=F)
> REML.res <- REML.results(fit.REML,x1=temp$GCM,x2=temp$scenario,CI=F)
> REML.res
```

```
        mu    sigma_G    sigma_S   sigma_GS    sigma_R
1.08454952 0.23077883 0.09655014 0.03948617 0.04533721
```

Now, we add estimated 95% confidence intervals.

```
> # conf% confidence intervals (CI=T, n.sim simulations, my.seed=(optional) random number seed)
> # The simulation number is printed to the screen (every 100 iterations).
> # With n.sim=1000, for example, this will take a few minutes
> REML.res <- REML.results(fit.REML,x1=temp$GCM,x2=temp$scenario,CI=TRUE,conf=95,n.sim=10,my.seed=1)
> REML.res
```

```
           estimate          SE lower limit upper limit
mu       1.08454952 0.051410567  0.97722435  1.11217234
sigma_G  0.23077883 0.039885309  0.17331124  0.30127709
sigma_S  0.09655014 0.048361379  0.01591441  0.16051255
sigma_GS 0.03948617 0.008523832  0.02486251  0.05082362
sigma_R  0.04533721 0.002537944  0.04336055  0.05123868
```

3.2 Bayesian analyses with half-Cauchy weakly-informative priors

Firstly, set the scale parameter of the half-Cauchy priors for super-population standard deviations.

```
> if (which.period==1) my.A <- 0.5 # set scale parameter of half-Cauchy prior for SDs
> if (which.period==2) my.A <- 1
```

We use the function `bugs()` in the package `R2WinBUGS` to draw an approximate sample from the posterior distribution of the parameters using MCMC. This `bugs` function is called within the function `MCMC.half.Cauchy.centre`. Use

```
> help(bugs)
```

to see what are the arguments to, and outputs from, `bugs`.

Note in particular that `R2WinBUGS` needs to know where `WinBUGS` is installed on your computer. By default it will look in somewhere like "C:/Program Files/WinBUGS14/" but if `WinBUGS` does not exist in this location then the code will fail with error messages that include "WinBUGS executable does not exist in c:/Program Files/WinBUGS14/".

You can tell `R2WinBUGS` where to look for `WinBUGS` using the `bugs.directory` argument to `bugs()`. The code is set up so that you can pass this argument to `bugs()` via `MCMC.half.Cauchy.finite()`. First we define an object containing the location of `WinBUGS` on the computer.

```
> my.bugs <- "C:/Users/paul/Documents/WinBUGS14"
```

Then in the calls to `MCMC.half.Cauchy.finite()` below we add the argument `bugs.directory=my.bugs`. Removing this argument will mean that `bugs()` will look for `WinBUGS` in the default location.

Here we simulate 5 independent Markov chains of length 2500 observations each and discard the first 5000 observations from each chain. We use all the $5 \times 2000 = 10000$ observations to estimate the posterior distribution of the parameters (the fixed effect μ and the super-population standard devations and finite-population standard deviations. When `MCMC.half.Cauchy.centre` is called a WinBUGS window will open and close when the calculations finish.

```
> #----------  Super-population and finite-population standard devations -------------#
> set.seed(200)
> my.sim.fin <- MCMC.half.Cauchy.finite(temp,A=my.A,n.iter=2500,n.chains=5,n.burnin=500,
+    n.thin=1,bugs.directory=my.bugs)
> Bayes.res.fin <- my.sim.fin$summary[-10,]
> row.names(Bayes.res.fin) <- c("mu","sigma_G","sigma_S","sigma_GS","sigma_R",
+    "s_G","s_S","s_GS","s_R")
> print(round(Bayes.res.fin,3)) # summary of the results
```

```
          mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
mu       1.091 0.176 0.754 1.022 1.090 1.160 1.422 1.005  3200
sigma_G  0.239 0.038 0.177 0.212 0.234 0.261 0.327 1.012   270
sigma_S  0.217 0.204 0.058 0.105 0.156 0.254 0.723 1.010   340
sigma_GS 0.041 0.009 0.025 0.035 0.040 0.047 0.060 1.011   310
sigma_R  0.046 0.004 0.040 0.044 0.046 0.049 0.055 1.005   670
s_G      0.231 0.009 0.214 0.225 0.231 0.236 0.248 1.002  2200
```

```
s_S      0.097 0.009 0.079 0.091 0.097 0.103 0.114 1.004   990
s_GS     0.040 0.008 0.026 0.035 0.040 0.045 0.056 1.009   350
s_R      0.046 0.002 0.042 0.044 0.046 0.047 0.051 1.004   950
```

```
> #summary(my.sim.fin)          # my.sim contains lots of other stuff!
```

This is a summary of the posterior sample mean, standard deviation and various quantiles for each of the parameters. `Rhat` is a standard diagnostic for MCMC convergence. As a rough guide values of less than 1.1 suggest that approximate convergence has been attained. `n.eff` is the effective sample size on which the estimated posterior distribution of the parameter is based.

4 Analysis of precipitation data

We repeat the Bayesian analysis the global precipitation indices for period 1 (2020–2049).

Firstly, we set up the data and select the region and time period.

```
> #=================== Precipitation analyses (global or regional) ====================#
> load(file="raindata.Rdata") # load the rainfall data
> summary(all.rain.data)
```

```
       y             scenario                GCM            run           region        period
 Min.   :-40.8427   A1B:2254   ncar_ccsm3_0   : 920   run1   :2714   Min.   : 0   Min.   :1.0
 1st Qu.: -0.2442   A2 :1656   cccma_cgcm3_1  : 690   run2   :1104   1st Qu.: 5   1st Qu.:1.0
 Median :  3.0333   B1 :2024   mri_cgcm2_3_2a : 690   run3   : 966   Median :11   Median :1.5
 Mean   :  3.6342              mpi_echam5     : 460   run4   : 460   Mean   :11   Mean   :1.5
 3rd Qu.:  7.0937              miroc3_2_medres: 414   run5   : 414   3rd Qu.:17   3rd Qu.:2.0
 Max.   : 37.8825              miub_echo_g    : 414   run6   :  92   Max.   :22   Max.   :2.0
                               (Other)        :2346   (Other): 184
```

```
> which.period <- 1          # 1 for 2020-2049, 2 for 2069-2098
> which.reg <- 0             # 0 for global, 1 for region 1 etc.
> cat(paste("region",which.reg,", period",which.period),"\n")
```

```
region 0 , period 1
```

```
> cond <- which(all.rain.data$region==which.reg & all.rain.data$period==which.period)
> rain <- all.rain.data[cond,]                                    # extract relevant data
> #---------------------------------- Bayesian analysis ------------------------------#
>
> if (which.period==1) my.A <- 2.5 # set scale parameter of half-Cauchy prior for SDs
> if (which.period==2) my.A <- 5
>
> #---------------  Super-population standard devations only --------------------------#
```

Now, use MCMC as before.

```
> set.seed(200)
> my.sim.fin.prec <- MCMC.half.Cauchy.finite(rain,A=my.A,n.iter=2500,n.chains=5,n.burnin=500,
+    n.thin=1,bugs.directory=my.bugs)
> Bayes.prec.res.fin <- my.sim.fin.prec$summary[-10,]
> row.names(Bayes.prec.res.fin) <- c("mu","sigma_G","sigma_S","sigma_GS","sigma_R",
+   "s_G","s_S","s_GS","s_R")
> print(round(Bayes.prec.res.fin,3)) # summary of the results
```

```
          mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
mu       1.543 0.394 0.840 1.387 1.534 1.700 2.289 1.029   480
sigma_G  0.647 0.109 0.473 0.571 0.634 0.709 0.897 1.007   510
sigma_S  0.402 0.529 0.060 0.138 0.237 0.467 1.679 1.023   140
sigma_GS 0.199 0.032 0.144 0.176 0.197 0.218 0.269 1.006   600
sigma_R  0.111 0.010 0.094 0.105 0.111 0.117 0.132 1.004  1100
s_G      0.615 0.031 0.552 0.595 0.616 0.637 0.676 1.003  1600
s_S      0.124 0.035 0.055 0.101 0.124 0.147 0.191 1.016   240
s_GS     0.194 0.025 0.149 0.177 0.192 0.210 0.248 1.005  1000
s_R      0.110 0.007 0.099 0.105 0.109 0.114 0.125 1.002  2200
```

References

Bates, D. and M. Maechler (2009). *lme4: Linear mixed-effects models using S4 classes*. R package version 0.999375-32.

Gelman, A., Y.-S. Su, M. Yajima, J. Hill, M. G. Pittau, J. Kerman, and T. Zheng (2010). *arm: Data Analysis Using Regression and Multilevel/Hierarchical Models*. R package version 1.3-02.

Sturtz, S., U. Ligges, and A. Gelman (2005). R2WinBUGS: A package for running WinBUGS from R. *Journal of Statistical Software 12*(3), 1–16.