The R package `nclbayes`

The `nclbayes` package is available on all university PC clusters which run R. Please do not install it in your university account.

You can install the package on your own PC by typing (within Rstudio or R)

```
install.packages("nclbayes",repos="http://R-Forge.R-project.org")
```

Note that you only need to install the package once on your own PC – not every time you want to use it.

Distributions

- Inverse Chi distribution [*Inv-Chi(a,b)*]
  density, distribution function, quantile function and random numbers:
  `dinvchi, pinvchi, qinvchi, rinvchi`
- Generalised t distribution [$t_a(b, c)$]
  density, distribution function, quantile function and random numbers:
  `dgt, pgt, qgt, rgt`
- Normal-Gamma distribution [$NGa(b, c, g, h)$]
  density, random numbers, confidence regions and elicitation of parameter values:
  `dnormgamma, rnormgamma, NGacontour, elicitNGa`
- Normal-InvChi distribution
  density and random numbers: `dnorminvchi, rnorminvchi`
- Bivariate Normal distribution [$N_2(\boldsymbol{\mu}, \Sigma)$]
  density: `dbvnorm`
- Bivariate t distribution [$t_a(\boldsymbol{b}, c)$]
  density: `dbvt`

Highest density intervals (HDIs)

- Beta distribution [*Beta(a,b)*]: `hdiBeta`
- Gamma distribution [*Ga(a,b)*]: `hdiGamma`
- Inv-Chi distribution [*Inv-Chi(a,b)*]: `hdiInvchi`

MCMC algorithms

- `gibbsNormal`: Gibbs sampler for a normal random sample with semi-conjugate prior
- `gibbsNormal2`: Gibbs sampler for a normal random sample with conjugate prior
- `gibbsReffects`: Gibbs sampler for a one-way normal random effects model with semi-conjugate prior
- `metropolis`: Metropolis algorithm for simulating from a standard normal distribution
- `mwgGamma`: Metropolis within Gibbs algorithm for a gamma random sample
- `mhReffects`: Metropolis–Hastings algorithm for a one-way normal random effects model using normal and log normal random walk proposals

Other functions

- `mcmcAnalysis`: summarises and plots MCMC output from the above algorithms
- `mcmcProcess`: chops off burnin and then thins MCMC output
- `mcmcCi`: calculates equi-tailed confidence intervals from MCMC output

Demos

- `review`: code used in Chapter 1
- `cavendish`: analyses Cavendish's data on the earth's density
- `gibbs`: code to demonstrate the Gibbs sampler
- `mh`: code to demonstrate the Metropolis-Hastings algorithm
- `sundries`: code used in the notes but not in any other demo

As an example, you can run the demo `cavendish` using `demo(cavendish)` and view the commands in this demo using `demoCommands(cavendish)`.