Applied Bayesian Modeling - module 5

Leontine Alkema

August 29, 2022

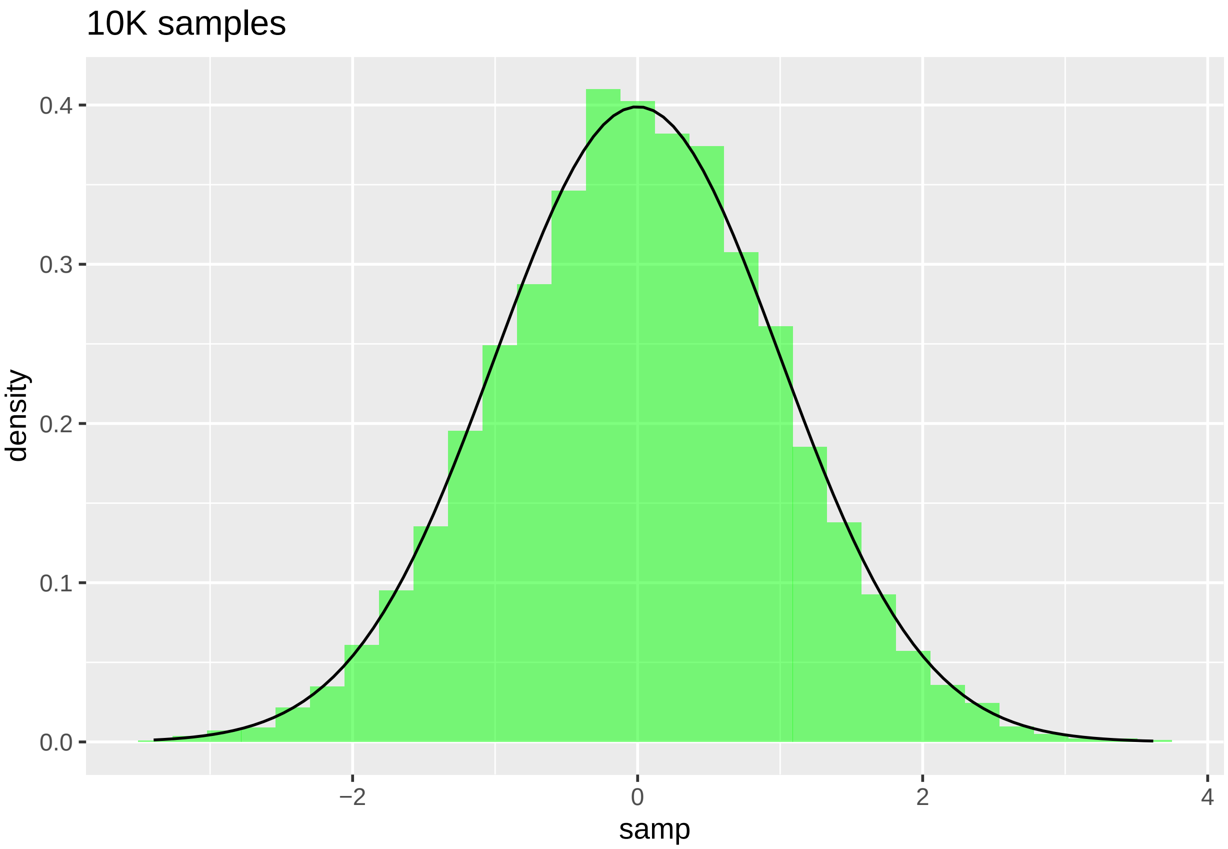
1 Illustration of MC approximation

Note: make sure to set a seed to make results that involve sampling reproducible

```
set.seed(1234)
samp <- rnorm(10000)
mean(samp)
```

```
## [1] 0.006115893
```

```
tibble(samp) %>%
  ggplot() +
  geom_histogram(aes(x = samp, y =..density..), fill = "green", alpha = 0.5) +
  stat_function(fun = dnorm, args = list(mean = 0, sd = 1)) +
 ggtitle("10K samples")
```



2 Fit a Bayesian model using brms

In this lab, we will first install the brms and rstan packages for Bayesian model fitting. The goal is to make sure that everyone has a working set-up, to not be held back by that next week when we're starting to get into details of brms/rstan.

2.1 Installation

Steps are to:
- install a C++ compiler (needed for Stan to run), see here
https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started#configuring-c-toolchain
- followed by installing the rstan and brms packages, see
https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started#latest-development-version-226x---unreleased
https://github.com/paul-buerkner/brms#how-do-i-install-brms

2.2 Verfiy that your rstan installation is working alright

You can then verify your rstan installation, see here https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started#verifying-installation I.e., run the following

```
#example(stan_model, package = "rstan", run.dontrun = TRUE)
```

2.3 Fit a model using brms

Let's fit some models to the radon data

```
library(brms)
```

Read in the radon data and process (copied from earlier module)

```
# house level data
d <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/srrs2.dat"),
                header=T, sep=",")

# deal with zeros, select what we want, make a fips (county) variable to match on
d <- d %>%
  mutate(activity = ifelse(activity==0, 0.1, activity)) %>%
  mutate(fips = stfips * 1000 + cntyfips) %>%
  dplyr::select(fips, state, county, floor, activity)

# county level data
cty <- read.table(url("http://www.stat.columbia.edu/~gelman/arm/examples/radon/cty.dat"),
                  header = T, sep = ",")
cty <-
  cty %>%
  mutate(fips = 1000 * stfips + ctfips) %>%
  dplyr::select(fips, Uppm) %>%
  rename(ura_county = (Uppm))

dmn <- d %>%
  filter(state=="MN") %>% # Minnesota data only
  dplyr::select(fips, county, floor, activity) %>%
  left_join(cty)

y <- log(dmn$activity)
ybar <- mean(y)
sd.y <- sd(y)
n <- length(y)
```

Just fit one model to check that brms is working ok. Here we are again estimating the mean μ, just like in module 3, using brms-default priors for μ and σ (to be discussed further). To estimate the mean, we can use a regression model (and corresponding formula, like in an lm call) with an intercept only.

Note: I save the fitted model in a directory "output", such that when knitting, the model is not rerun (see argument of brms, if you're interested). It's a handy option, just watch out with it when you're still updating your data inputs (as it won't update!).

```
#dir.create("output") # if you don't have an output directory yet
fit <- brm(y ~ 1, family = gaussian(), data = tibble(y),
            cores = getOption("mc.cores", 4),
           file = "output/mod5_fit" # adding this to save the fit,
           # then when knitting, the model is not rerun (see arguments)
            )
```

Check the output

```
fit
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: tibble(y) (Number of observations: 927)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.23      0.03     1.17     1.28 1.00     3338     2562
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.02     0.83     0.90 1.00     3475     2031
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

2.4 Optional extra fits

Fit models with different combinations of
- Mu: informative vs vague prior for mu
- sigma fixed at sd(y) or estimated (with default vague prior assigned by brms)

Specifying prior settings for mu (we'll discuss details later re how this is coded):

```
mu_informativepriorhw1 <- set_prior("normal(0,0.1)", class = "Intercept")
mu_vagueprior <- set_prior("normal(0,100)", class = "Intercept")
sigma_fixed <- set_prior("constant(0.86)", class = "sigma")
```

Now fit

```
fit_muinformative_sigmafixed  <- brm(y ~ 1, family = gaussian(), data = tibble(y),
            cores = getOption("mc.cores", 4),
            prior = c(mu_informativepriorhw1, sigma_fixed),
             file = "output/mod5_fit_muinformative_sigmafixed"
            )
```

```
fit_muvague_sigmafixed  <- brm(y ~ 1, family = gaussian(), data = tibble(y),
            cores = getOption("mc.cores", 4),
            prior = c(mu_vagueprior, sigma_fixed),
            file = "output/mod5_fit_muvague_sigmafixed"
            )
```

```
fit_muvague_estimatesigma  <- brm(y ~ 1, family = gaussian(), data = tibble(y),
            cores = getOption("mc.cores", 4),
            prior = c(mu_vagueprior),
            file = "output/mod5_fit_muvague_estimatesigma"
            )
```

```r
fit_muinformative_estimatesigma  <- brm(y ~ 1, family = gaussian(), data = tibble(y),
            cores = getOption("mc.cores", 4),
            prior = c(mu_informativepriorhw1),
            file = "output/mod5_fit_muinformative_estimatesigma"
            )
```

```r
fit_muinformative_sigmafixed
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: tibble(y) (Number of observations: 927)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.14      0.03     1.09     1.19 1.00     1572     1852
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.00     0.86     0.86   NA       NA       NA
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
fit_muvague_sigmafixed
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: tibble(y) (Number of observations: 927)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.23      0.03     1.17     1.28 1.00     1413     2242
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.00     0.86     0.86   NA       NA       NA
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```r
fit_muinformative_estimatesigma
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: tibble(y) (Number of observations: 927)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.13      0.03     1.08     1.19 1.00     2663     2275
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.87      0.02     0.83     0.91 1.00     2800     2657
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
fit_muvague_estimatesigma
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: tibble(y) (Number of observations: 927)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.23      0.03     1.17     1.28 1.00     3666     2963
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.02     0.83     0.90 1.00     3418     2523
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```