Applied Bayesian Modeling - module 6

Leontine Alkema

September 17, 2022

1 Radon data

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)
```

2 Model fitting using lm and brms

data

```r
dat <- dmn %>%
  mutate(y = log(activity))
```

2.1 frequentist/tradtional

simple fit

```r
fit_lm <-  lm(y ~ 1, data = dat)
summary(fit_lm)
```

```
##
## Call:
## lm(formula = y ~ 1, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5300 -0.5856  0.0535  0.5971  2.6479
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.22745    0.02836   43.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8635 on 926 degrees of freedom
```

here the same model fitting, just coded up differently in a pipe, with output in nice form

```r
dat %>%
  lm(y ~ 1, data = .) %>%
  broom::tidy(conf.int = TRUE, conf.level = 0.95) %>%
  select(-statistic, -p.value) #%>%
```

```
## # A tibble: 1 x 5
##   term        estimate std.error conf.low conf.high
##   <chr>          <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)     1.23    0.0284     1.17      1.28
```

```r
#  knitr::kable(format = "latex", digits = 2)
```

2.2 Bayesian regression

```r
fit <- brm(y ~ 1, data = dat,
           file = "output/mod6ex2",
           chains = 4, iter = 1000, warmup = 500,
           cores = getOption("mc.cores", 4))
```

Quick summary overview

```
summary(fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: dat (Number of observations: 927)
##   Draws: 4 chains, each with iter = 1000; warmup = 500; thin = 1;
##          total post-warmup draws = 2000
##
## 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     1808     1532
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.02     0.82     0.91 1.00     1930     1329
##
## 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).
```

Aside on summaries: you can pull out whatever you're interested in

```
posterior_summary(fit, probs = c(.025, .25, .75, .975), variable = "b_Intercept")
```

```
##             Estimate Est.Error     Q2.5     Q25      Q75    Q97.5
## b_Intercept 1.226463 0.0270974 1.170451 1.20898 1.243945 1.278986
```

Note that the object fit now contains lots of info, we will go through some here

```
names(fit)
```

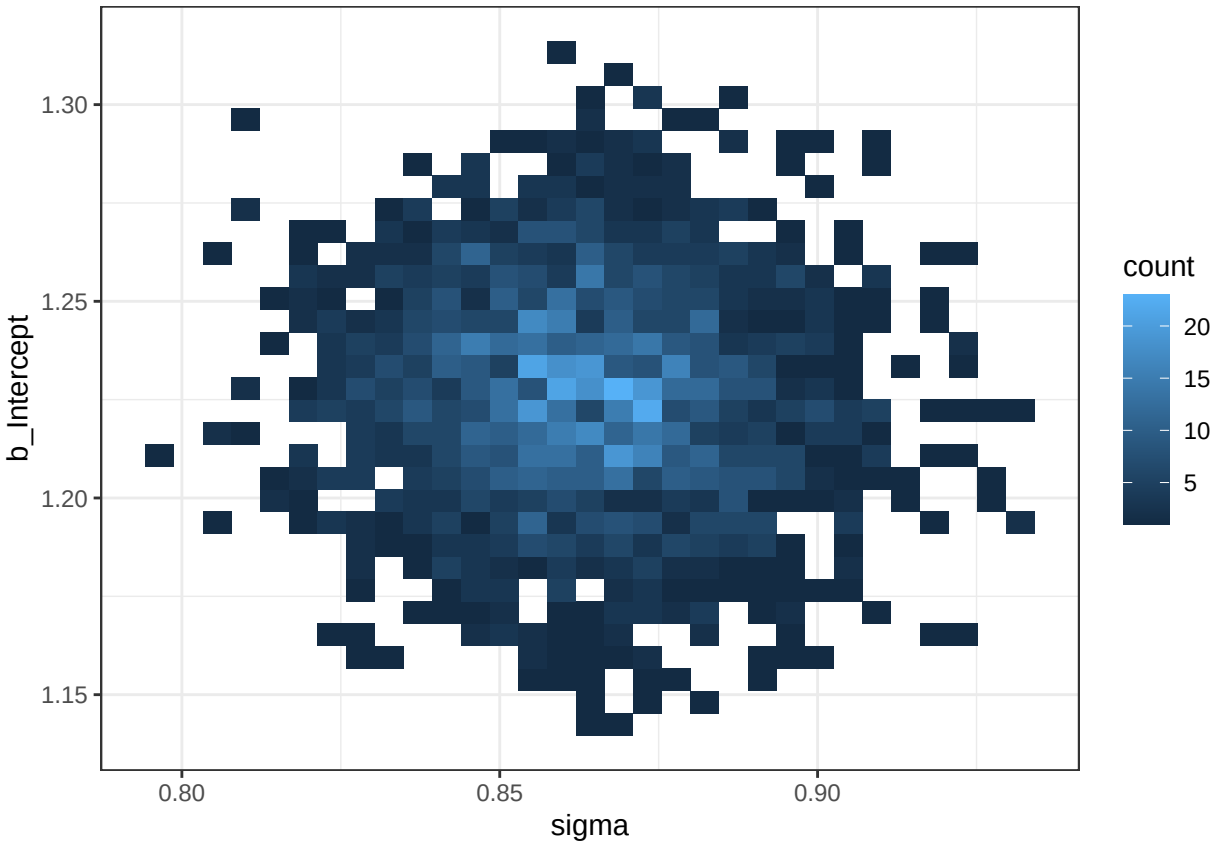
```
##  [1] "formula"   "data"      "prior"     "data2"     "stanvars"  "model"
##  [7] "ranef"     "save_pars" "algorithm" "backend"   "threads"   "opencl"
## [13] "stan_args" "fit"       "criteria"  "file"      "version"   "family"
## [19] "autocor"   "cov_ranef" "stan_funs" "data.name"
```

3 Plots to show posterior samples

Joint density, using bins

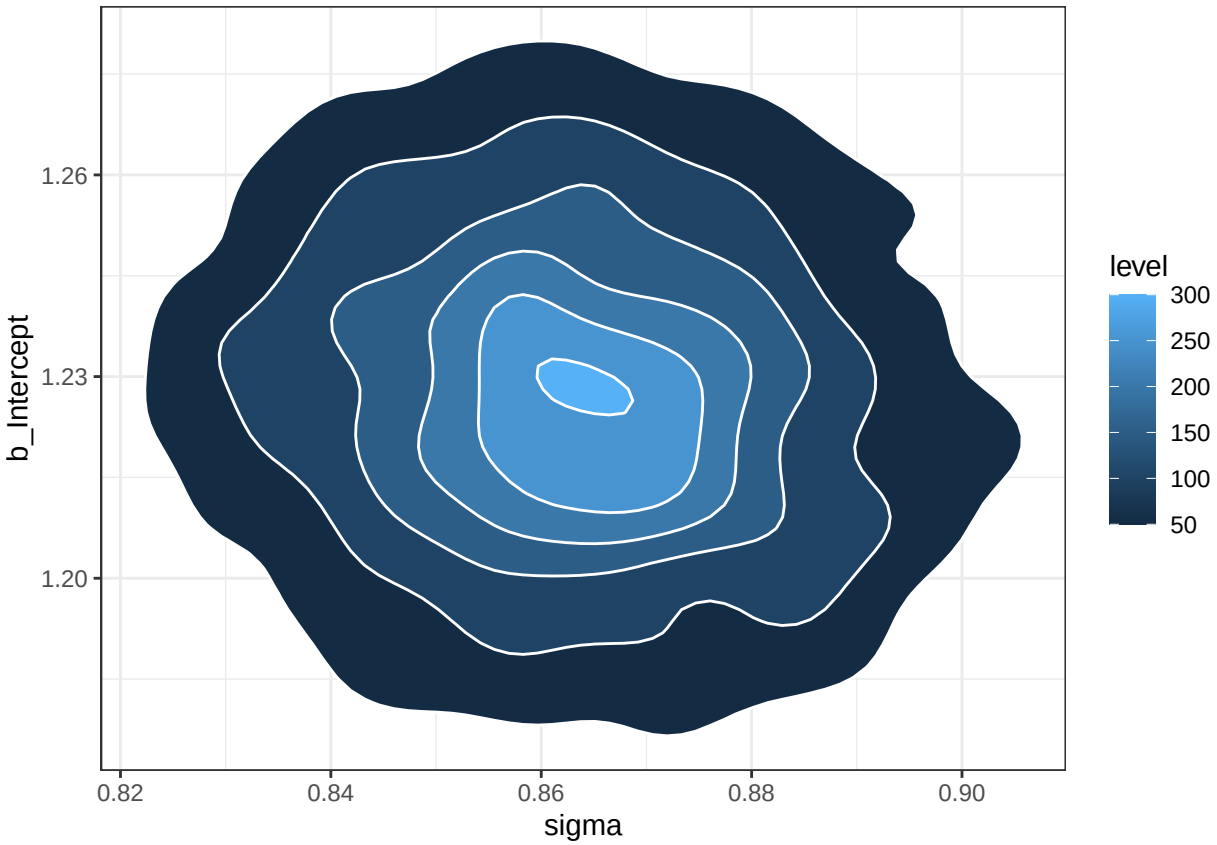
```
# posterior_samples(fit, variable = c("b_Intercept", "sigma")) %>%
#   ggplot(aes(x = sigma, y = b_Intercept)) +
#   geom_bin2d() +
#   theme_bw()

mod6ex3samples <- as_draws_df(fit, variable = c("b_Intercept", "sigma"))
mod6ex3samples %>%
  ggplot(aes(x = sigma, y = b_Intercept)) +
  geom_bin2d() +
  theme_bw()
```



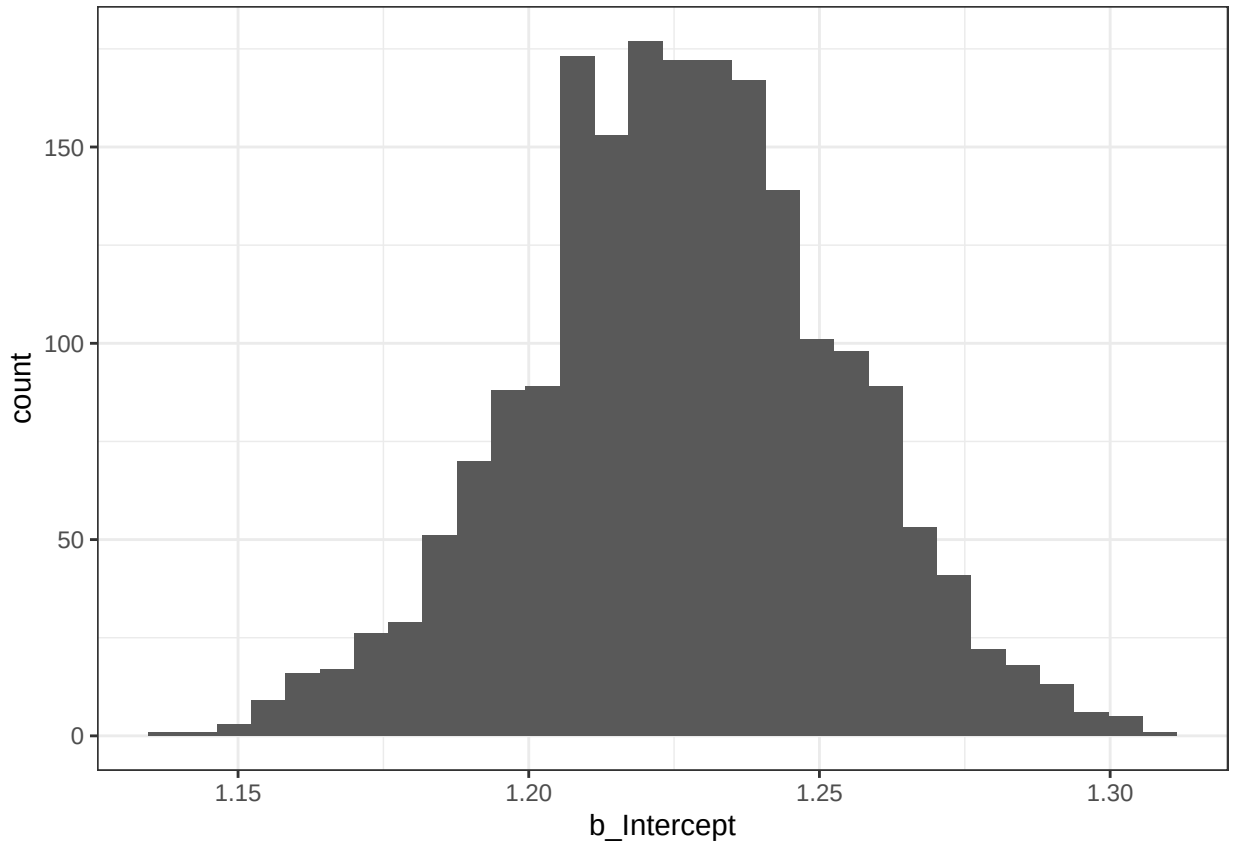
Joint density, estimated

```
# posterior_samples(fit, pars = c("b_Intercept", "sigma"))
mod6ex3samples %>%
  ggplot(aes(x = sigma, y = b_Intercept)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white")+
  theme_bw()
```

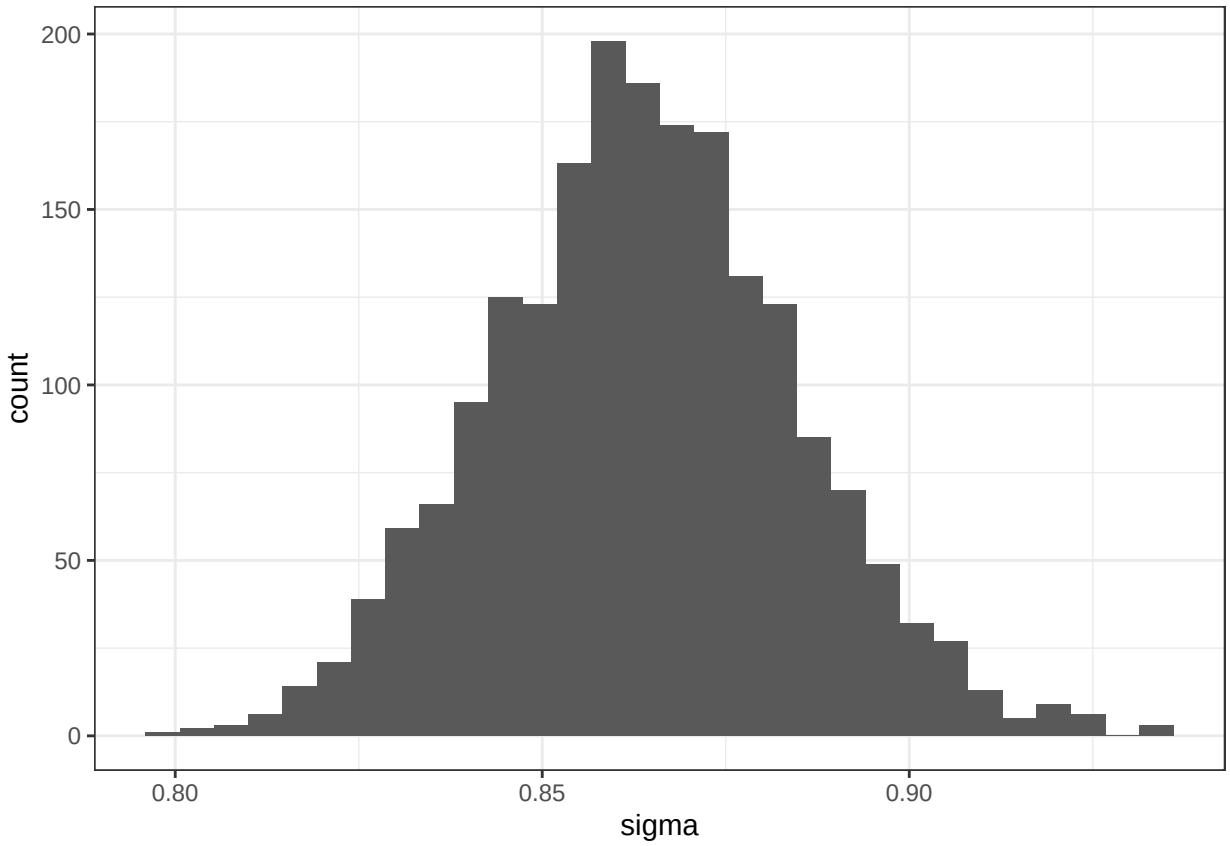


marginal densities

```
# posterior_samples(fit, pars = c("b_Intercept"))
mod6ex3samples %>%
  ggplot(aes(x = b_Intercept)) +
  geom_histogram() +
  theme_bw()
```




```
# posterior_samples(fit, pars = c("sigma"))
mod6ex3samples %>%
  ggplot(aes(x = sigma)) +
  geom_histogram() +
  theme_bw()
```

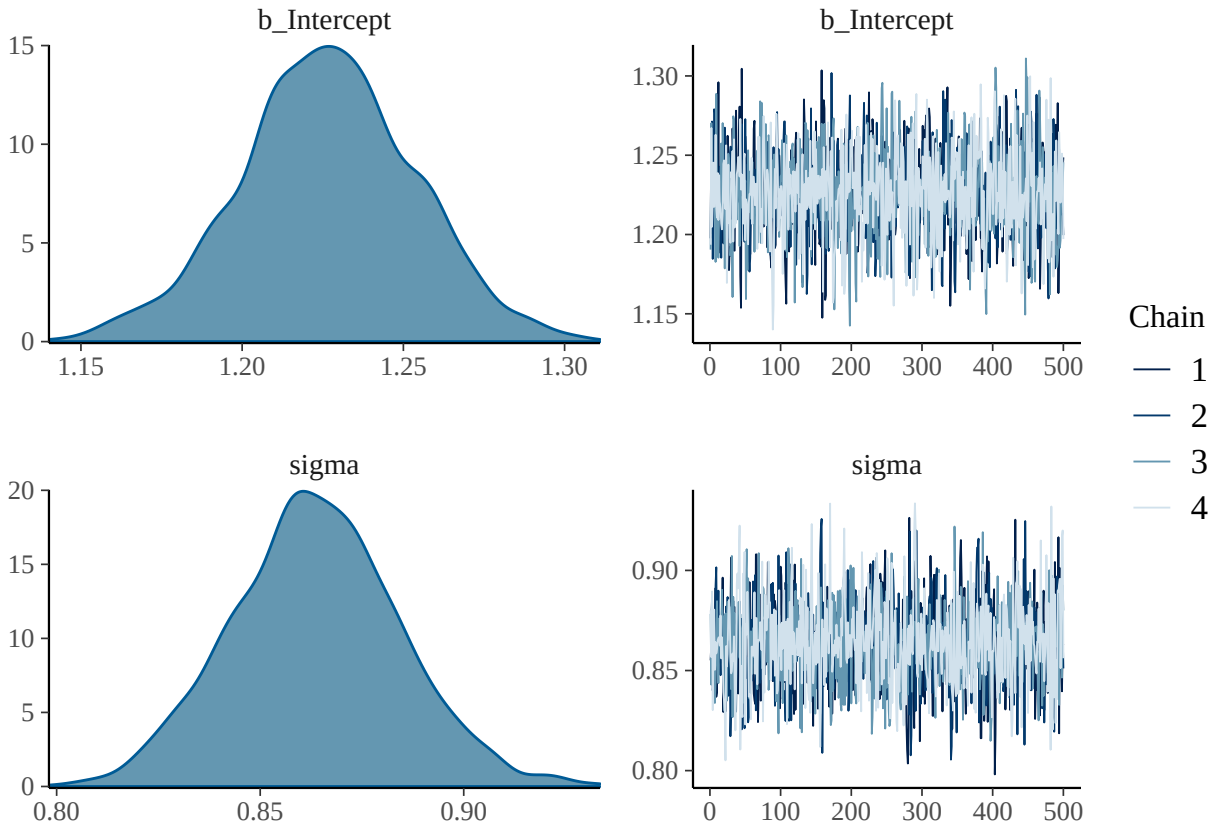


4 MCMC diagnostics

Traceplots and posterior densities. Note that you can find the help function with

```
#?plot.brmsfit
```

```
plot(fit, variable = c("b_Intercept", "sigma"))
```



Check Rhat and effective sample size

```
summary(fit)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: dat (Number of observations: 927)
##   Draws: 4 chains, each with iter = 1000; warmup = 500; thin = 1;
##          total post-warmup draws = 2000
##
## 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     1808     1532
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.02     0.82     0.91 1.00     1930     1329
##
## 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).
```

```
#names(summary(fit))
summary(fit)$fixed
```

```
##           Estimate Est.Error l-95% CI u-95% CI     Rhat Bulk_ESS Tail_ESS
```

```
## Intercept 1.226463 0.0270974 1.170451 1.278986 1.003624 1807.893 1532.356
```

```
summary(fit)$spec_pars
```

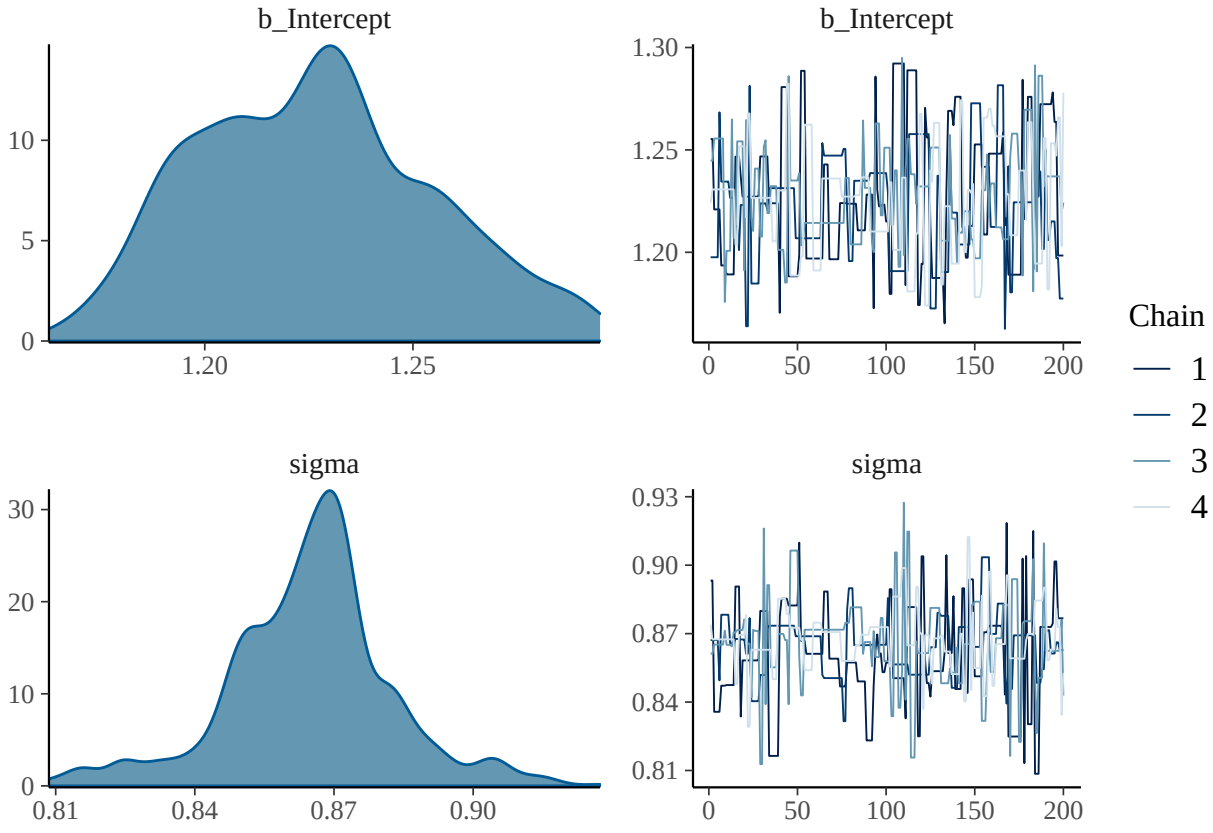
```
##        Estimate Est.Error  l-95% CI u-95% CI     Rhat Bulk_ESS Tail_ESS
## sigma 0.8637291 0.0206609 0.8241331  0.90576 1.000891  1929.88 1328.798
```

5 MCMC diagnostics in a less ideal setting. . .

Please note: These fits are based on settings that are NOT recommended. We are just creating an example here of a fit where the MCMC diagnostics (rightly) show that there are issues with the sampling.

```
fit_bad <- brm(y ~ 1, data = dat,
               file = "output/mod6ex5",
        chains = 4, iter = 400, cores = getOption("mc.cores", 4),
        control = list(adapt_delta = 0.4, max_treedepth = 4)
        # these are NOT recommended options, trying to create problems here!
)
```

```
plot(fit_bad,  variable = c("b_Intercept",  "sigma"))
```




```
summary(fit_bad)
```

```
##  Family: gaussian
```

```
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: dat (Number of observations: 927)
##   Draws: 4 chains, each with iter = 400; warmup = 200; thin = 1;
##          total post-warmup draws = 800
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.23      0.03     1.18     1.29 1.10      260      131
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.86      0.02     0.82     0.90 1.09      268      125
##
## 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).
```

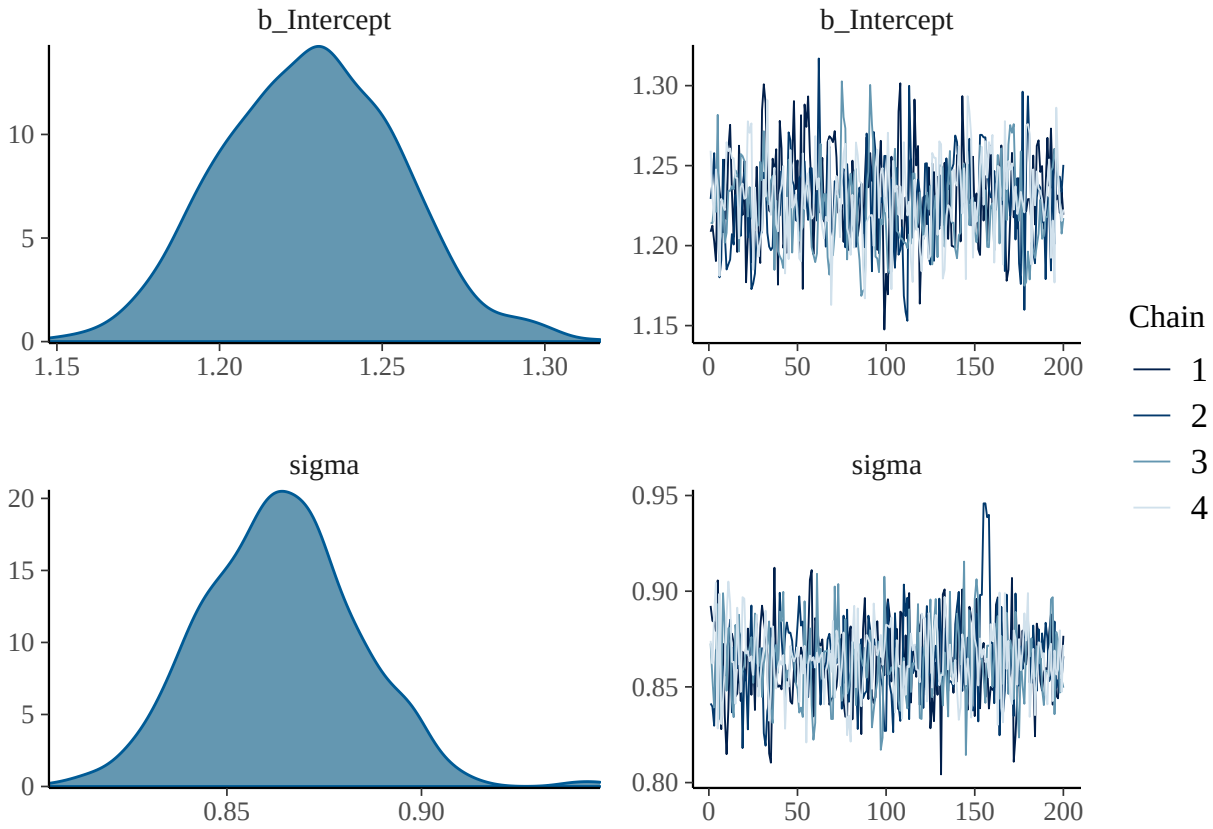
We see that the effective sample size are too low. Rhat is no longer equal to 1 (although still less than 1.05).

5.1 comparison fit

Here we create a fit with the same number of iterations for comparison

```
fit2_short <- brm(y ~ 1, data = dat,
                  file = "output/mod6ex5.1",
        chains = 4, iter = 400, cores = getOption("mc.cores", 4))
```

```
plot(fit2_short,  variable = c("b_Intercept",  "sigma"))
```




```
summary(fit2_short)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: y ~ 1
##    Data: dat (Number of observations: 927)
##   Draws: 4 chains, each with iter = 400; warmup = 200; thin = 1;
##          total post-warmup draws = 800
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     1.23      0.03     1.18     1.28 1.00      485      508
##
## 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 1.01      718      501
##
## 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).
```

Much larger effective sample sizes!