Hierarchical models: practical exercises

Robert Goudie

Full worked solutions are provided at the back of this document. Feel free to consult these as you are going along, if you are stuck on any of the exercises.

The document and embedded code are also provided as an R Markdown document in `hierarchical_practical.Rmd`. This allows the R code blocks to be run conveniently from RStudio.

1 THM

In this example, we will explore a simulated set of data for the THM example used in the lecture.

The data are in the file `thm.rds` in the form of a data frame, with `zone` indicating the zone of the corresponing `thm` measurement:

```
(thm <- readRDS(file = "thm.rds"))
```

```
## # A tibble: 156 x 2
##    zone    thm
##    <fct> <dbl>
##  1 1      116.
##  2 1      116.
##  3 1      112.
##  4 2      126
##  5 2      129.
##  6 2      129.
##  7 2      126.
##  8 3      100.
##  9 4      108.
## 10 4      114.
## # ... with 146 more rows
```

As usual for JAGS, we will need to convert this data into a `list` containing numeric values, and provide JAGS with the number of observations. The function `compose_data()` in the `tidybayes` package is one way to do this. For factor variables, as `thm$zone` is coded here, `compose_data()` will create variable that contains the number of levels in that variable; here this is `n_zone`. It also creates an `n` variable that is the number of rows in the data frame.

Note that we have no observations from some of the `zone`s, so there are factor levels without any corresponding data: for example zone 5 has no data. Not removing these zones will lead JAGS to treat these are missing data, and produce a prediction for these zones, which will be useful. The warning from `compose_data()` about the unused levels can thus be ignored here.

```
library(tidybayes, quietly = TRUE)
thm_dat <- compose_data(thm)
```

```
## Warning in as_data_list.factor(object[[i]], name = names(object)[[i]], ...):
## Some levels of factor "zone" are unused. This may cause issues if you are using
## it as the dimension for a variable in a model.
```

```
# Compare the result and the raw data frame
thm
```

```
## # A tibble: 156 x 2
##    zone    thm
##    <fct> <dbl>
##  1 1      116.
##  2 1      116.
##  3 1      112.
##  4 2      126
##  5 2      129.
##  6 2      129.
##  7 2      126.
##  8 3      100.
##  9 4      108.
## 10 4      114.
## # ... with 146 more rows
```

```
str(thm_dat)
```

```
## List of 4
##  $ zone  : num [1:156(1d)] 1 1 1 2 2 2 2 3 4 4 ...
##  $ n_zone: int 70
##  $ thm   : num [1:156(1d)] 116 116 112 126 129 ...
##  $ n     : int 156
```

The JAGS specification for the basic THM model is as follows:

```
thm_model <- "
model {
  for (i in 1:n){
    thm[i] ~ dnorm(theta[zone[i]], sigma.prec)    # likelihood for observed data
  }
  for (z in 1:n_zone) {
    theta[z] ~ dnorm(mu, tau.prec)    # zone-specific means (random effects)
  }

  # priors on random effects mean and SD
  mu ~ dnorm(120, 1/100^2)
  # random effects SD (between-zone SD of mean THM)
  tau.sd ~ dunif(0, 100)
  tau.var <- pow(tau.sd, 2)
  tau.prec <- 1 / tau.var

  sigma.sd ~ dexp(1/10)
  sigma.prec <- 1/sigma.sd^2
  sigma.var <- sigma.sd^2
}"
```

1. Choose some initial values for `sigma.sd`, `mu` and `tau.sd`, and run this model, monitoring the zone-specific means, residual variance, overall mean, and the between-zone variance.
2. For each zone, calculate both the zone-level mean MLE (simply the zone-level sample mean) and the posterior mean of the zone-level mean `theta` under the model above. Plot these together on a single graph.

   Hints:

- *the posterior samples output by* `coda.samples()` *for* `theta` *for each zone will be labelled* `theta[1]`*,* `theta[2]` *etc, which may be inconvenient for plotting on the same graph as the MLEs. There are several ways to extract the zone number* `1` *from labels like* `theta[1]`*: one convenient function is* `spread_draws` *from the* `tidybayes` *package (or alternatively the similar* `spread_rvars` *function). See below for an example of its use.*
- *If you run* `recover_types(thm)` *(also from* `tidybayes`*) before running* `spread_draws`*, then the output will be of the same type as the corresponding column in the data frame* `thm`*: here, this makes* `zone` *into a factor variable.*

```
library(posterior, quietly = TRUE)
# thm_post here is the output from coda.samples()
thm_draws <- as_draws(thm_post)

thm_theta_post_spread <- subset_draws(thm_draws, "theta") %>%
  recover_types(thm) %>%
  spread_draws(theta[zone])
```

3. What is the variance partition coefficient, and what does this tell us?
4. Edit the model code to allow the residual error variance to be zone specific, and assume a hierarchical prior distribution for the logarithms of these variances as suggested in the Lecture.

[Hint: Change `sigma.prec` *to* `sigma.prec[zone[i]]` *in the likelihood and add an appropriate hierarchical prior for it to the* `for (z in 1:Nzone)` *loop]*

2 Hierarchical regression

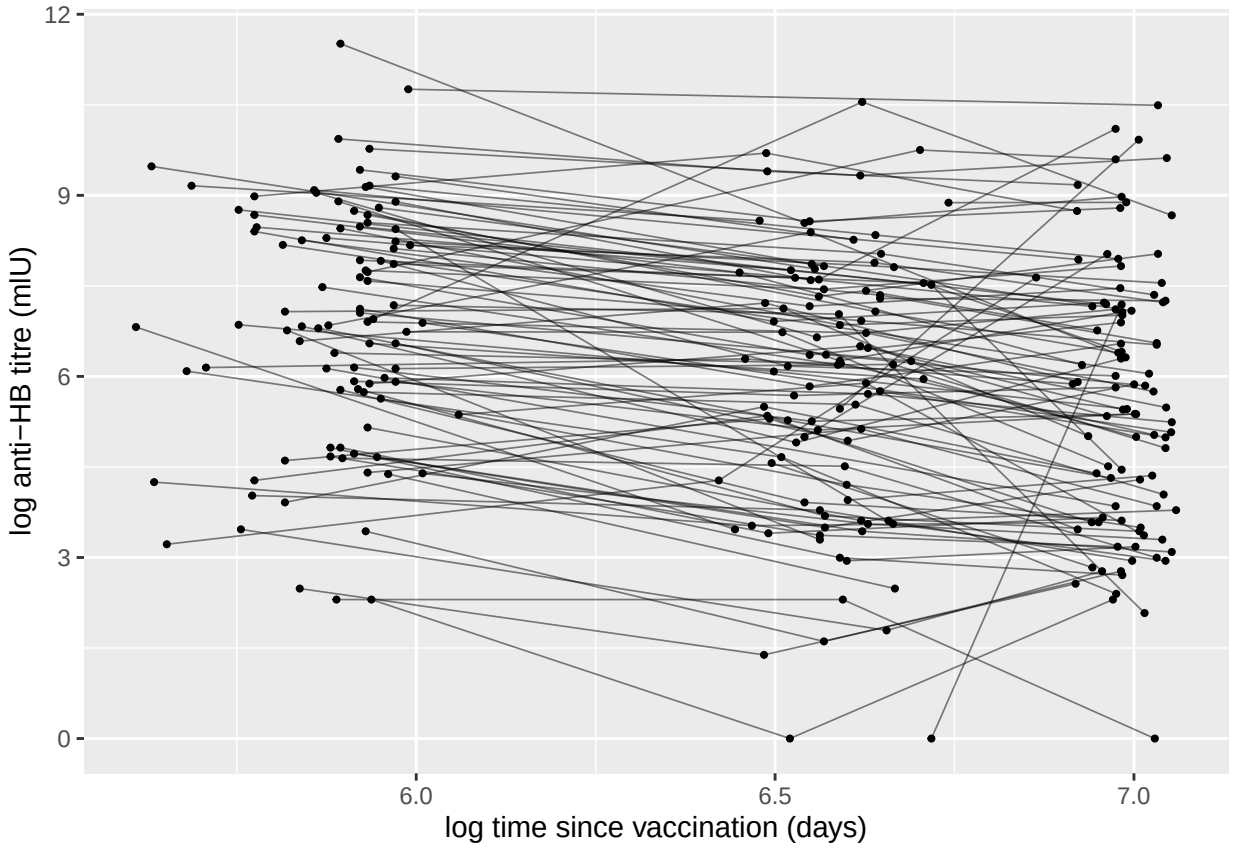
Consider the Hep B data we looked at in the lecture. The data are in the file `hepb.rds`. It contains both logged versions of the Anti-HB titre and time, as well as centred versions of these. We will use the centred versions in the model.

```
(hepb <- readRDS(file = "hepb.rds"))
```

```
## # A tibble: 288 x 6
##    child log_time log_anti_hb log_anti_hb_at_~ log_time_centred log_anti_hb_at_~
##    <fct>    <dbl>       <dbl>            <dbl>            <dbl>            <dbl>
##  1 1         6.54        5.00             8.61           0.0512             1.89
##  2 1         6.96        8.03             8.61           0.473              1.89
##  3 2         5.84        6.83             7.10          -0.649              0.386
##  4 2         6.53        4.90             7.10           0.0396             0.386
##  5 2         6.98        6.30             7.10           0.492              0.386
##  6 3         6.60        3.95             6.90           0.111              0.177
##  7 3         7.03        4.36             6.90           0.536              0.177
##  8 4         5.86        6.80             5.64          -0.626             -1.08
##  9 4         6.52        5.27             5.64           0.0279            -1.08
## 10 4         6.97        4.32             5.64           0.478             -1.08
## # ... with 278 more rows
```

The raw data can be plotted as

```
library(tidyverse)
ggplot(hepb, aes(x = log_time, y = log_anti_hb, group = child)) +
  geom_line(alpha = 0.5, size = 0.3) +
  geom_point(size = 0.75) +
  xlab("log time since vaccination (days)") +
  ylab("log anti-HB titre (mlU)")
```




```
library(rjags)
```

```
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.9.0
## - Online documentation and vignettes at mc-stan.org/bayesplot
## - bayesplot theme set to bayesplot::theme_default()
##    * Does _not_ affect other ggplot2 plots
##    * See ?bayesplot_theme_set for details on theme setting
```

```
library(posterior)
```

```
## This is posterior version 1.2.1
##
## Attaching package: 'posterior'
## The following object is masked from 'package:bayesplot':
##
##     rhat
## The following objects are masked from 'package:stats':
##
```

```
##     mad, sd, var
```

We can form the input to JAGS using:

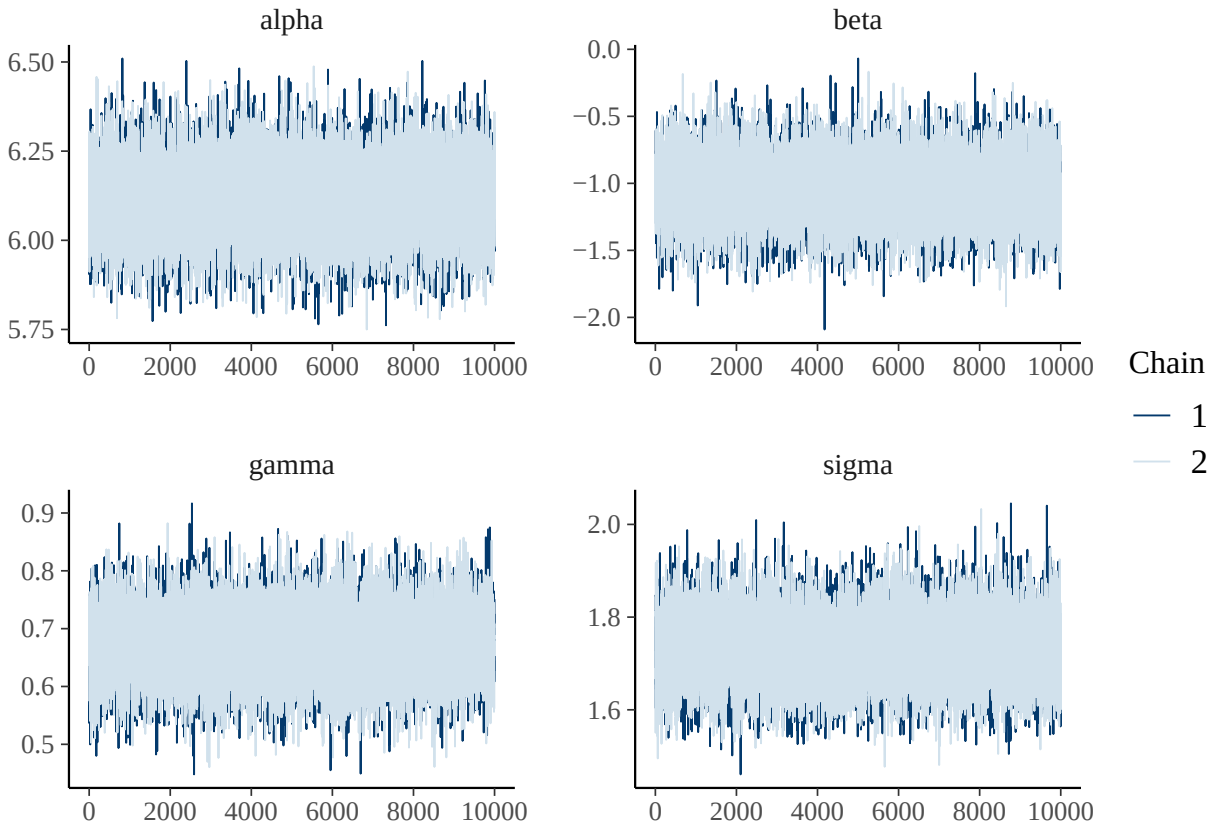
```
library(tidybayes)
dat_jag <- hepb %>%
  select(log_anti_hb, log_time_centred, log_anti_hb_at_time0_centred) %>%
  compose_data()
```

The basic (non-hierarchical) model can be coded as:

```
hepb_basic_mod <- "model {
  for (i in 1:n){
    log_anti_hb[i] ~ dnorm(mu[i], sigma.prec)
    mu[i] <- alpha +
             beta * log_time_centred[i] +
             gamma * log_anti_hb_at_time0_centred[i]
  }

  sigma.prec <- 1/sigma.sd
  sigma.sd <- pow(sigma, 2)
  log(sigma) <- log.sigma
  log.sigma ~ dunif(-10, 10)

  alpha ~ dnorm(0, 1/100^2)
  beta ~ dnorm(0, 1/100^2)
  gamma ~ dnorm(0, 1/100^2)
}"
```

And we can run the model as

```
ini <- list(list(alpha = 0, beta = 0, gamma = 0),
            list(alpha = 1, beta = 1, gamma = 1))
par <- c("alpha", "beta", "gamma", "sigma")
hepb_basic_jag <- jags.model(textConnection(hepb_basic_mod),
                             data = dat_jag,
                             inits = ini,
                             n.chains = 2,
                             quiet = TRUE)

update(hepb_basic_jag, 1000)
sam <- coda.samples(hepb_basic_jag,
                    n.iter = 10000,
                    variable.names = par)

mcmc_trace(sam)
```




```
hepb_basic_draws <- as_draws(sam)

summary(hepb_basic_draws)
```

```
## # A tibble: 4 x 10
##   variable   mean median     sd    mad     q5    q95  rhat ess_bulk ess_tail
##   <chr>     <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>    <dbl>    <dbl>
## 1 alpha     6.11   6.11  0.102  0.102   5.95   6.28   1.00   19971.   19837.
## 2 beta     -1.05  -1.05  0.225  0.224  -1.42  -0.679  1.00   19824.   18112.
## 3 gamma     0.673  0.673 0.0572 0.0567  0.579  0.768  1.00   20165.   19169.
## 4 sigma     1.73   1.73  0.0734 0.0741  1.61   1.85   1.00   12010.   11753.
```

```
summary(hepb_basic_draws, default_convergence_measures())
```

```
## # A tibble: 4 x 4
##   variable  rhat ess_bulk ess_tail
##   <chr>    <dbl>    <dbl>    <dbl>
## 1 alpha     1.00   19971.   19837.
## 2 beta      1.00   19824.   18112.
## 3 gamma     1.00   20165.   19169.
## 4 sigma     1.00   12010.   11753.
```

```
summary(hepb_basic_draws, default_mcse_measures())
```

```
## # A tibble: 4 x 6
##   variable mcse_mean mcse_median  mcse_sd  mcse_q5 mcse_q95
##   <chr>        <dbl>       <dbl>    <dbl>    <dbl>    <dbl>
## 1 alpha     0.000722    0.000952 0.000511 0.00122  0.00159
## 2 beta      0.00160     0.00218  0.00113  0.00360  0.00335
```

```
## 3 gamma     0.000403    0.000554 0.000286 0.000912 0.000849
## 4 sigma     0.000670    0.000717 0.000474 0.00129  0.00151
```

1. Adapt the model code to make the hierarchical model described in the lecture.

3 Meta analysis: sepsis example

The Sepsis data (discussed in the lecture) aim to answer the question: 'does administration of intravenous immunoglobulin (IVIG) prevent infection in hospital, compared to placebo?' The raw data from the 10 studies are reproduced in the table below.

		Treatment (IVIG)		Control	
Study	Country	Events	Total	Events	Total
Bussel (1990a)	US	20	61	23	65
Chirico (1987)	Italy	2	43	8	43
Clapp (1989)	US	0	56	5	59
Conway (1990)	UK	8	34	14	32
Fanaroff (1994)	US	186	1204	209	1212
Haque (1986)	Saudi Arabia	4	100	5	50
Ratrisawadi (1991)	Thailand	10	68	13	34
Sandberg (2000)	Sweden/Austria	19	40	13	41
Tanzer (1997)	Turkey	3	40	8	40
Weisman (1994a)	US	40	372	39	381

Table 1: Sepsis data

These data can be provided to R using:

```
# control data is in the first column, treatment in the second
sepsis.data <- list(
  Ns = 10,
  y = matrix(c(23, 8, 5, 14, 209, 5, 13, 13, 8, 39, 20, 2, 0, 8, 186, 4,
               10, 19, 3, 40),
               nrow = 10,
               ncol = 2),
  n = matrix(c(65, 43, 59, 32, 1212, 50, 34, 41, 40, 381, 61, 43, 56, 34,
               1204, 100, 68, 40, 40, 372),
               nrow = 10,
               ncol = 2)
)
```

- We wish to fit a random-effects meta-analysis model for odds ratio.

$$y_{sT} \sim \text{Bin}(n_{sT}, p_{sT}) \qquad y_{sC} \sim \text{Bin}(n_{sC}, p_{sC})$$
$$\text{logit}(p_{sT}) = \alpha_s + \mu_s \qquad \text{logit}(p_{sC}) = \alpha_s$$
$$\mu_s \sim N(\delta, \tau^2)$$

Here y_{sT} is the number of events in study s in the IVIG group, and n_{sT} is the total number in the IVIG group; y_{sC} and n_{sC} are the corresponding numbers in the control group.

Fit this meta-analysis model using the following model specification and data provided.

```
sepsis.re.model <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    ## study-specific baselines, vague priors
    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dnorm(0, 1/10^2)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"
ini <- list(list(delta = 0, alpha = rep(0, 10)))
pars <- c("p", "alpha", "mu", "delta")

sepsis_re_jag <- jags.model(textConnection(sepsis.re.model),
                            data = sepsis.data,
                            inits = ini,
                            n.chains = 1,
                            quiet = TRUE)
update(sepsis_re_jag, n.iter = 1000)
sepsis_re_post <- coda.samples(model = sepsis_re_jag,
                               variable.names = pars,
                               n.iter = 50000)
```

1. The parameter δ is the overall log odds ratio treatment effect. What does the posterior distribution of this parameter suggest about the effectiveness of the treatment?
2. The figure in the lecture shows the log odds ratio treatment effect rather than the odds ratio. Obtain a posterior sample for this quantity.
3. Modify the code to fit a common effect meta-analysis for the odds ratios. How does the estimated treatment effect change? Can you explain why this happens?
4. The table above gives the country in which each study was conducted. Is there evidence that the treatment effect is higher in US hospitals?
    - *Hint: One way to assess this would be a model in which the size of the treatment effect is related to study covariates X_s:*

      $$\mu_s \sim N(\delta + \beta X_s, \tau^2)$$

      Here your regression model would include an indicator variable for whether the study was conducted in the US. (This is called "meta-regression".) Alternatively, consider the model separately for US and non-US studies - how does this approach differ to the meta-regression approach?

4 Prior sensitivity

Transfusions of granulocytes (a type of white blood cell) have been used clinically for many years to support and treat severe infection in high risk groups of patients with neutropenia or neutrophil dysfunction. However, there is still uncertainty about whether such transfusions are beneficial. In 2005, the Cochran Collaboration published a systematic review of the available evidence. Six studies were included in their study of all-cause mortality. We will explore the dependence of the results on the priors are chosen.

```
Ns <- 6
y <- matrix(c(4, 5, 3, 1, 14, 2, 1, 0, 9, 7, 13, 18), byrow = T, ncol = 2)
n <- matrix(c(11, 13, 14, 13, 19, 17, 12, 13, 13, 17, 47, 48), byrow = T, ncol = 2)

gran_dat <- list(Ns = Ns, y = y, n = n)
```

- Consider the random-effects meta-analysis model for odds ratio.

$$y_{sT} \sim \text{Bin}(n_{sT}, p_{sT}) \qquad y_{sC} \sim \text{Bin}(n_{sC}, p_{sC})$$
$$\text{logit}(p_{sT}) = \alpha_s + \mu_s \qquad \text{logit}(p_{sC}) = \alpha_s$$
$$\mu_s \sim N(\delta, \tau^2)$$

Here y_{sT} is the number of events in study s in the IVIG group, and n_{sT} is the total number in the IVIG group; y_{sC} and n_{sC} are the corresponding numbers in the control group.

1. Imagine you want to be sceptical about the effect δ of transfusions of granulocytes. Specifically, you think *apriori* that

- there is only a 2.5% chance that transfusions will reduce the odds of mortality by 25% or more
- and that it is most likely there is no effect i.e. the mode of the prior is 1 on the odd ratio scale, and 0 on the log odds ratio scale.

Formulate a prior distribution matching these beliefs.

[Hint: the first condition is equivalent to a 2.5% chance that the log odd ratio is less than -0.29]

2. Fit the meta-analysis model with the 'vague' prior $\delta \sim \text{N}(0, 10^2)$ using the supplied model and data files. Compare the results to the results with your sceptical prior, and using an alternative vague prior $\delta \sim \text{Unif}(-100, 100)$.

- Compare the following priors for the random effect variance sd.mu^2.
  (a) Uniform$(0, 10)$ for the standard deviation
  (b) Uniform$(0, 100)$ for the standard deviation
  (c) Gamma$(1, 1)$ for the precision
  (d) Gamma$(0.001, 0.001)$ for the precision

Either run each of these in turn. Or try to run them all in the same model file, taking care to ensure the models under each prior are completely separate, using the following set of inits and data list, which repeats the data 4 times:

```
ini_gran_repeated <- list(
  list(
    delta = rep(0, 4),
    alpha = matrix(rep(rep(0, 6), 4), nrow = 4)
  )
)

gran_dat_repeated <- gran_dat
```

```
gran_dat_repeated$Np <- 4
gran_dat_repeated$y <- array(NA, dim = c(4, dim(gran_dat$y)))
gran_dat_repeated$y[1,,] <- gran_dat$y
gran_dat_repeated$y[2,,] <- gran_dat$y
gran_dat_repeated$y[3,,] <- gran_dat$y
gran_dat_repeated$y[4,,] <- gran_dat$y
gran_dat_repeated$n <- array(NA, dim = c(4, dim(gran_dat$n)))
gran_dat_repeated$n[1,,] <- gran_dat$n
gran_dat_repeated$n[2,,] <- gran_dat$n
gran_dat_repeated$n[3,,] <- gran_dat$n
gran_dat_repeated$n[4,,] <- gran_dat$n
```

- Turner et al (2012) recommend for meta-analysis of all-cause mortality with non-pharmacological interventions a log-normal$(-3.93, 1.51^2)$ informative prior for the distribution of the variance τ^2, where 3.93 is the mean on the log scale, and 1.51^2 is the variance on the log scale. How does that change the results?

[Hint: log normal distributions are specified as dlnorm(a, b) in BUGS, where a is the mean on the log scale and b is the precision on the log scale]

5 Solutions

5.1 THM example

1.

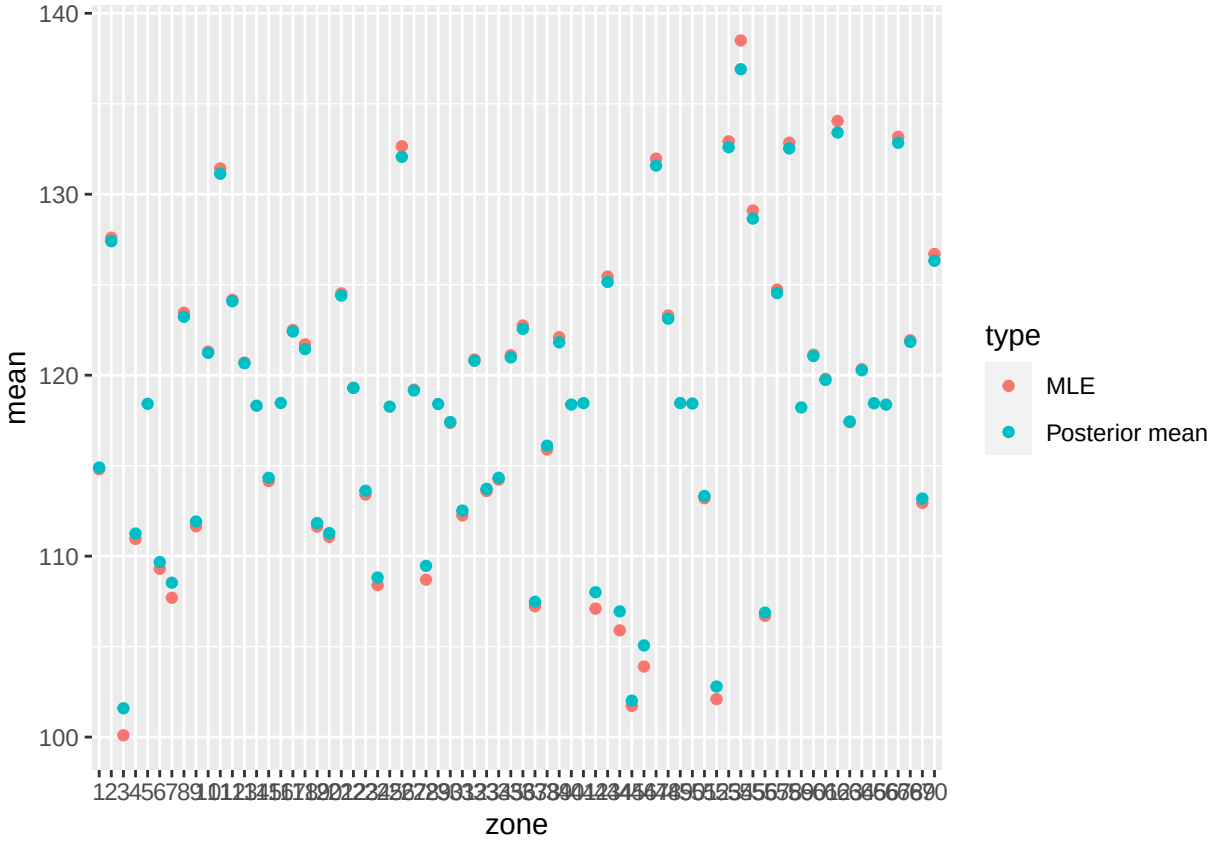
```
thm_in <- list(list(sigma.sd = 5, mu = 50, tau.sd = 1),
               list(sigma.sd = 20, mu = 200, tau.sd = 10))

library(rjags, quietly = TRUE)
thm_jag <- jags.model(file = textConnection(thm_model),
                      data = thm_dat,
                      inits = thm_in,
                      n.chains = 2,
                      quiet = TRUE)
update(thm_jag, n.iter = 5000)
pars <- c("mu", "sigma.var", "tau.var", "theta")
thm_post <- coda.samples(thm_jag,
                    variable.names = pars,
                    n.iter = 10000)
```

2.

```
thm_mle <- thm %>%
  group_by(zone) %>%
  summarise(mean = mean(thm)) %>%
  mutate(type = "MLE")

library(posterior, quietly = TRUE)
# thm_post here is the output from coda.samples()
thm_draws <- as_draws(thm_post)

thm_theta_post_spread <- subset_draws(thm_draws, "theta") %>%
  recover_types(thm) %>%
  spread_draws(theta[zone])
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# use the output of spread_draws shown in the Hint.
thm_theta_post_mean <- thm_theta_post_spread %>%
  group_by(zone) %>%
  summarise(mean = mean(theta)) %>%
  mutate(type = "Posterior mean")

thm_both <- bind_rows(thm_mle, thm_theta_post_mean)

ggplot(thm_both, aes(x = zone, y = mean, colour = type)) +
  geom_point()
```



Note again that the posterior means are "shrunk" towards the overall mean compared to the MLEs.

3. We can monitor the variable `vpc` defined as `vpc <- tau.var / (tau.var + sigma.var)` by adding this to the model. (Alternatively your could monitor `tau.var` and `sigma.var` in the original model, and then calculate `vpc` for every sample in R.)

```
thm_model <- "
model {
  for (i in 1:n){
    thm[i] ~ dnorm(theta[zone[i]], sigma.prec)    # likelihood for observed data
  }
  for (z in 1:n_zone) {
    theta[z] ~ dnorm(mu, tau.prec)    # zone-specific means (random effects)
  }
```

```
  # priors on random effects mean and SD
  mu ~ dnorm(120, 1/100^2)
  # random effects SD (between-zone SD of mean THM)
  tau.sd ~ dunif(0, 100)
  tau.var <- pow(tau.sd, 2)
  tau.prec <- 1 / tau.var

  sigma.sd ~ dexp(1/10)
  sigma.prec <- 1/sigma.sd^2
  sigma.var <- sigma.sd^2

  vpc <- tau.var / (tau.var + sigma.var)    # variance partition coefficient
}"

thm_in <- list(list(sigma.sd = 5, mu = 50, tau.sd = 1),
               list(sigma.sd = 20, mu = 200, tau.sd = 10))

library(rjags, quietly = TRUE)
thm_jag <- jags.model(file = textConnection(thm_model),
                      data = thm_dat,
                      inits = thm_in,
                      n.chains = 2,
                      quiet = TRUE)
update(thm_jag, n.iter = 5000)
pars <- c("vpc")
thm_post <- coda.samples(thm_jag,
                    variable.names = pars,
                    n.iter = 10000)

library(posterior, quietly = TRUE)
summary(as_draws(thm_post))
```

```
## # A tibble: 1 x 10
##   variable  mean median     sd    mad    q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 vpc      0.919  0.921 0.0187 0.0181 0.886 0.946  1.00    7684.   10504.
```

The VPC is around is 0.92, indicating the majority of the variation is *between* zones rather than *within* zones.

4.

```
thm_randvar_model <- "model {
  for (i in 1:n) {
    thm[i] ~ dnorm(theta[zone[i]], sigma.prec[zone[i]])    # likelihood for observed data
  }
  for (z in 1:n_zone) {
    theta[z] ~ dnorm(mu, tau.prec)    # zone-specific means (random effects)
    log.sigma.var[z] ~ dnorm(lambda, nu.prec)    # zone-specific log variances (random effects)
    sigma.prec[z] <- 1/sigma.var[z]
    sigma.var[z] <- exp(log.sigma.var[z])
  }

  # priors on random effects mean and SD
  mu ~ dnorm(120, 1/100^2)
  # random effects SD (between-zone SD of mean THM)
```

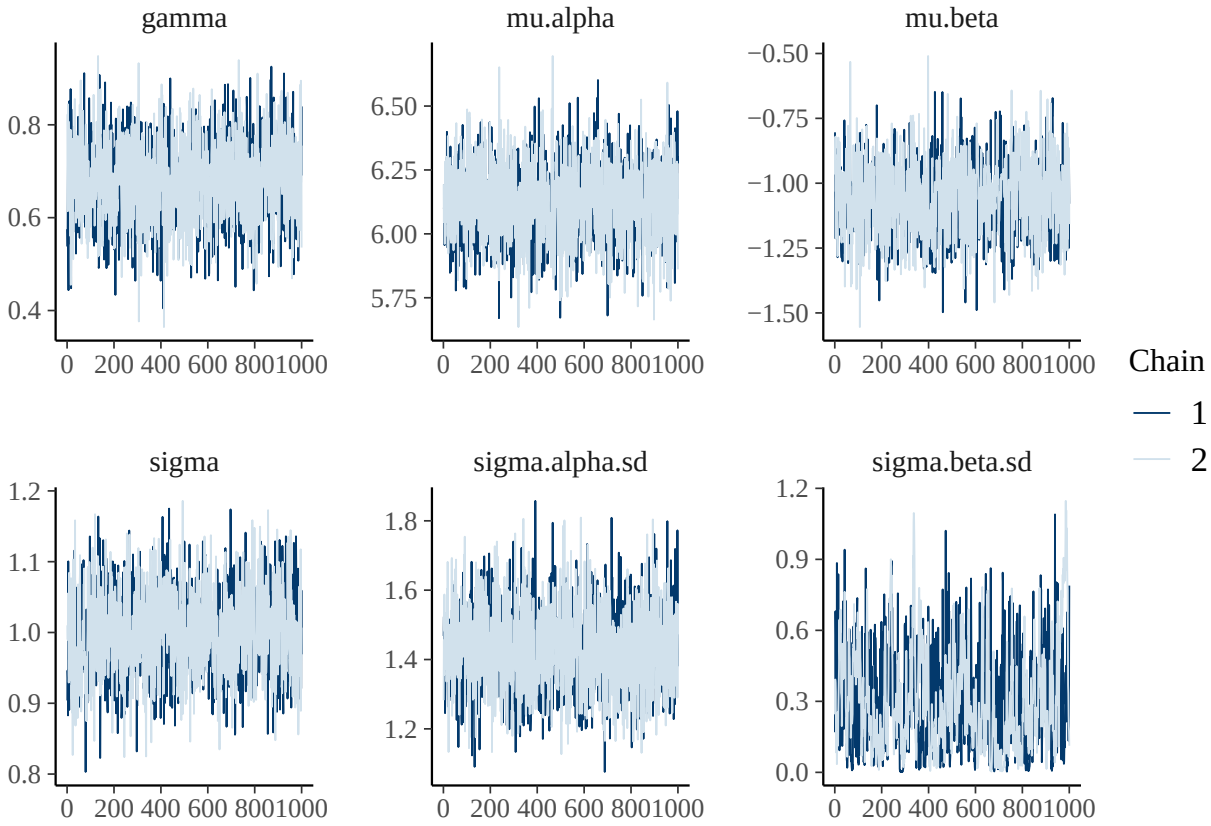
```
  tau.sd ~ dunif(0, 100)
  tau.var <- pow(tau.sd, 2)
  tau.prec <- 1 / tau.var

  lambda ~ dunif(-100, 100)    # mean log variance
  mean.var <- exp(lambda)    # mean residual error variance
  nu.sd ~ dunif(0, 100)
  nu.var <- pow(nu.sd, 2) # between-zone variance in log error variance
  nu.prec <- 1 / nu.var

  vpc <- tau.var / (tau.var + mean.var)    # average variance partition coefficient
}
"
thm_randvar_in <- list(list(mu = 50, tau.sd = 1, lambda = 2, nu.sd = 0.5),
                       list(mu = 200, tau.sd = 10, lambda = 1, nu.sd = 0.25))

thm_randvar_jag <- jags.model(file = textConnection(thm_randvar_model),
                      data = thm_dat,
                      inits = thm_randvar_in,
                      n.chains = 2,
                      quiet = TRUE)
update(thm_randvar_jag, n.iter = 5000)
pars <- c("mu","sigma.var","tau.var","vpc","theta")
sam <- coda.samples(thm_randvar_jag,
                    variable.names = pars,
                    n.iter = 10000)
```

5.2 Hierarchical example: Hep B

1.

```
hepb_mod <- "model {
  for (i in 1:n) {
    log_anti_hb[i] ~ dnorm(mu[i], sigma.prec)
    mu[i] <- alpha[child[i]] +
             beta[child[i]] * log_time_centred[i] +
             gamma * log_anti_hb_at_time0_centred[i]
  }
  for (c in 1:n_child){
    alpha[c] ~ dnorm(mu.alpha, sigma.alpha.prec)
    beta[c] ~ dnorm(mu.beta, sigma.beta.prec)
  }
  sigma.prec <- 1/sigma.sd
  sigma.alpha.prec <- 1/pow(sigma.alpha.sd, 2)
  sigma.beta.prec <- 1/pow(sigma.beta.sd, 2)
  sigma.sd <- pow(sigma, 2)
  log(sigma) <- log.sigma
  log.sigma ~ dunif(-10, 10)
  sigma.alpha.sd ~ dunif(0, 100)
  sigma.beta.sd ~ dunif(0, 100)
  mu.alpha ~ dnorm(0, 1/100^2)
  mu.beta ~ dnorm(0, 1/100^2)
  gamma ~ dnorm(0, 1/100^2)
}"
```

```
ini <- list(list(log.sigma = 0, sigma.alpha.sd = 0.1, sigma.beta.sd = 0.1,
                 mu.alpha = 0, mu.beta = 0, gamma = 0),
            list(log.sigma = 1, sigma.alpha.sd = 1, sigma.beta.sd = 1,
                 mu.alpha = 1, mu.beta = 1, gamma = 1))

dat_jag <- hepb %>%
  select(child, log_anti_hb, log_time_centred, log_anti_hb_at_time0_centred) %>%
  compose_data()

par <- c("mu.alpha", "mu.beta", "gamma", "sigma", "sigma.alpha.sd", "sigma.beta.sd")
hepb_jag <- jags.model(textConnection(hepb_mod),
                       data = dat_jag,
                       inits = ini,
                       n.chains = 2,
                       quiet = TRUE)

update(hepb_jag, 1000)
sam <- coda.samples(hepb_jag,
                    n.iter = 100000,
                    thin = 100,
                    variable.names = par)
mcmc_trace(sam)
```




```
hepb_draws <- as_draws(sam)
summary(hepb_draws)
```

```
## # A tibble: 6 x 10
##   variable      mean median     sd    mad      q5    q95  rhat ess_bulk ess_tail
```

```
##   <chr>        <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <dbl>    <dbl>    <dbl>
## 1 gamma        0.677  0.678 0.0858 0.0839  0.539   0.819  1.00    1876.    1736.
## 2 mu.alpha     6.12   6.12  0.146  0.147   5.88    6.36   1.00    2036.    1921.
## 3 mu.beta     -1.06  -1.06  0.138  0.140  -1.29   -0.832  1.00    1041.    1192.
## 4 sigma        0.996  0.993 0.0588 0.0586  0.905   1.10   1.00    1081.    1279.
## 5 sigma.alph~  1.43   1.43  0.120  0.122   1.24    1.63   1.00    2018.    1800.
## 6 sigma.beta~  0.302  0.262 0.215  0.226   0.0255  0.704  1.00     169.     281.
```

```
summary(hepb_draws, default_convergence_measures())
```

```
## # A tibble: 6 x 4
##   variable        rhat ess_bulk ess_tail
##   <chr>          <dbl>    <dbl>    <dbl>
## 1 gamma           1.00    1876.    1736.
## 2 mu.alpha        1.00    2036.    1921.
## 3 mu.beta         1.00    1041.    1192.
## 4 sigma           1.00    1081.    1279.
## 5 sigma.alpha.sd  1.00    2018.    1800.
## 6 sigma.beta.sd   1.00     169.     281.
```

```
summary(hepb_draws, default_mcse_measures())
```

```
## # A tibble: 6 x 6
##   variable       mcse_mean mcse_median mcse_sd mcse_q5 mcse_q95
##   <chr>              <dbl>       <dbl>   <dbl>   <dbl>    <dbl>
## 1 gamma            0.00198     0.00283 0.00140 0.00448  0.00451
## 2 mu.alpha         0.00322     0.00439 0.00228 0.00973  0.00543
## 3 mu.beta          0.00428     0.00707 0.00303 0.00538  0.00840
## 4 sigma            0.00177     0.00170 0.00125 0.00269  0.00283
## 5 sigma.alpha.sd   0.00270     0.00354 0.00193 0.00493  0.00797
## 6 sigma.beta.sd    0.0160      0.0176  0.0113  0.00497  0.0320
```

5.3 Meta analysis: sepsis example

1.

```
sepsis_re_draws <- as_draws(sepsis_re_post)
sepsis_re_delta_draws <- subset_draws(sepsis_re_draws, "delta")
summary(sepsis_re_delta_draws)
```

```
## # A tibble: 1 x 10
##   variable   mean median    sd   mad    q5     q95  rhat ess_bulk ess_tail
##   <chr>     <dbl>  <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>    <dbl>    <dbl>
## 1 delta    -0.576 -0.544 0.342 0.307 -1.17 -0.0938  1.00    7343.   15652.
```

The 95% posterior CI for the log odds ratio, delta, is (-1.35, -0.01), suggesting that IVIG is reducing the chances of in-hospital infection.

2. EITHER exponentiate the stored samples for delta:

```
delta <- extract_variable(sepsis_re_delta_draws, "delta")
delta_or <- exp(delta)
summary(delta_or)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02913 0.46419 0.58026 0.59391 0.70544 5.64514
```

OR rerun the model with an extra variable added

```
sepsis.re.model2 <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    ## study-specific baselines, vague priors
    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dnorm(0, 1/10^2)
  # Add the following line to get odds ratios rather than log odds ratios
  delta_or <- exp(delta)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"
pars <- c("p", "alpha", "mu", "delta", "delta_or")
ini <- list(list(delta = 0, alpha = rep(0, 10)),
            list(delta = -1, alpha = rep(2, 10)))

sepsis_re_jag <- jags.model(textConnection(sepsis.re.model2),
                            data = sepsis.data,
                            inits = ini,
                            n.chains = 2,
                            quiet = TRUE)
update(sepsis_re_jag, n.iter = 1000)
sepsis_re_post <- coda.samples(model = sepsis_re_jag,
                               variable.names = pars,
                               n.iter = 50000)

sepsis_re_draws <- as_draws(sepsis_re_post)
sepsis_re_delta_draws <- subset_draws(sepsis_re_draws, "delta_or")
summary(sepsis_re_delta_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.593  0.581 0.197 0.180 0.306 0.908  1.00   12746.   29888.
```

The 95% posterior CI for the odd ratio, delta_or, is (0.26, 0.99), with median 0.58.

3.

```
sepsis_ce_model <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
```

```
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      # Now only have a single treatment effect parameter
      logit(p[s, a]) <- alpha[s] + delta[a]
    }

    ## study-specific baselines, vague priors
    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta[1] <- 0
  delta[2] ~ dnorm(0, 1/10^2)
  delta_or <- exp(delta[2])

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"
pars <- c("p", "alpha", "delta", "delta_or")
ini <- list(list(delta = c(NA, 0), alpha = rep(0, 10)))

sepsis.ce.jag <- jags.model(textConnection(sepsis_ce_model),
                            data = sepsis.data,
                            inits = ini,
                            n.chains = 1,
                            quiet = TRUE)
update(sepsis.ce.jag, n.iter = 1000)
sepsis_ce_post <- coda.samples(model = sepsis.ce.jag,
                               variable.names = pars,
                               n.iter = 10000)

sepsis_ce_draws <- as_draws(sepsis_ce_post)
sepsis_ce_delta_draws <- subset_draws(sepsis_ce_draws, "delta_or")
summary(sepsis_ce_delta_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median     sd    mad    q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.819  0.815 0.0727 0.0724 0.706 0.945  1.00    2372.    4350.
```

The 95% posterior CI for the odds ratio is now (0.68, 0.96). See also the results shown for each study in the lecture.

Under the common effect model, the lower bound of the 95% CI for the overall odds ratio is considerably higher. This is due to the lack of shrinkage in the common effects model. e.g. the Sandberg (2000) study is not shrunk down towards the mean

4.

```
is.us <- c(1, 0, 1, 0, 1, 0, 0, 0, 0, 1)
sepsis.data$is.us <- is.us

sepsis.re.metareg.model <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
```

```
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta + beta * is.us[s], prec.mu)

    ## study-specific baselines, vague priors
    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dnorm(0, 1/10^2)
  beta ~ dnorm(0, 1/10^2)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"

ini <- list(
  list(
    delta = 0,
    alpha = rep(0, 10)
  ) ,
   list(
     delta = 0.1,
     alpha = c(1,-1,-2,0,0,-2,1,0,2,2)
   )
)
pars <- c("p", "alpha", "mu", "delta", "beta")

sepsis_re_metareg_jag <- jags.model(textConnection(sepsis.re.metareg.model),
                                    data = sepsis.data,
                                    inits = ini,
                                    n.chains = 2,
                                    quiet = TRUE)
update(sepsis_re_metareg_jag, n.iter = 1000)
sepsis_re_metareg_post <- coda.samples(model = sepsis_re_metareg_jag,
                                       variable.names = pars,
                                       n.iter = 50000,
                                       thin = 10)

sepsis_re_metareg_draws <- as_draws(sepsis_re_metareg_post)
sepsis_re_metareg_delta_draws <- subset_draws(sepsis_re_metareg_draws, "beta")
summary(sepsis_re_metareg_delta_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad     q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 beta     0.469  0.500 0.647 0.510 -0.595  1.41  1.00    8019.    8613.
```

There is a suggestion that the treatment is less effective in the US (the median for beta, the log odds ratio, is 0.52), but the 95% CI is wide (-0.95, 1.62) and includes 0, so the evidence is really quite weak for this effect.

Fitting a separate model is equivalent, except that prec.mu is shared between both the US and no-US studies in the above model, whereas they would be unconstrained if US and non-US studies were considered separately. (Of course, you could specify a meta-regression model that allowed `prec.mu` to differ between US and non-US studies

5.4 Prior sensitivity

1. For a Normal prior on the log odds ratio scale want 95% of the region to be between log(0.75)=-0.29 and 0.29, and mean = 0. So SD = 0.29/1.96 = 0.15. And precision = 1/(0.15^2). ie `delta ~ dnorm(0, 44)`.

2.

```
library(posterior)
library(bayesplot, quietly = TRUE)

# Model with vague prior
gran_re_vague_jagsmod <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dnorm(0, 1/10^2)
  delta_or <- exp(delta)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"

ini <- list(
  list(
    delta = 0,
    alpha = rep(0, 6)
  )
)

pars <- c("p", "alpha", "mu", "delta", "delta_or")
gran.re.vague.jag <- jags.model(textConnection(gran_re_vague_jagsmod),
                                data = gran_dat,
                                inits = ini,
                                n.chains = 1,
                                quiet = TRUE)
update(gran.re.vague.jag, n.iter = 1000)
gran_re_vague_post <- coda.samples(model = gran.re.vague.jag,
```

```
                                  variable.names = pars,
                                  n.iter = 50000)

gran_re_vague_draws <- as_draws(gran_re_vague_post)
gran_re_vague_delta_or_draws <- subset_draws(gran_re_vague_draws, "delta_or")
summary(gran_re_vague_delta_or_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad     q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.722  0.328  12.5 0.259 0.0484  1.41  1.00   17173.   13694.
```

```
gran_re_scept_jagsmod <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    alpha[s] ~ dnorm(0, 1/10^2)
  }

  # For a Normal prior on the log odds ratio scale want
  # 95% of the region to be between log(0.75)=-0.29 and 0.29, and
  # mean = 0. So SD = 0.29/1.96 = 0.15. And precision = 1/(0.15^2)
  delta ~ dnorm(0, 44)
  delta_or <- exp(delta)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"

ini <- list(
  list(
    delta = 0,
    alpha = rep(0, 6)
  )
)

pars <- c("p", "alpha", "mu", "delta", "delta_or")
gran.re.scept.jag <- jags.model(textConnection(gran_re_scept_jagsmod),
                                data = gran_dat,
                                inits = ini,
                                n.chains = 1,
                                quiet = TRUE)
update(gran.re.scept.jag, n.iter = 1000)
gran_re_scept_post <- coda.samples(model = gran.re.scept.jag,
                           variable.names = pars,
```

```
                              n.iter = 50000)

gran_re_scept_draws <- as_draws(gran_re_scept_post)
gran_re_scept_delta_or_draws <- subset_draws(gran_re_scept_draws, "delta_or")
summary(gran_re_scept_delta_or_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.970  0.960 0.145 0.143 0.753  1.23  1.00   39891.   46421.
```

```
# Vague delta ~ Unif(-100, 100) prior
gran_re_vague2_jagsmod <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
      y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dunif(-100, 100)
  delta_or <- exp(delta)

  prec.mu <- 1/sd.mu^2
  sd.mu ~ dunif(0, 10)
}
"

ini <- list(
  list(
    delta = 0,
    alpha = rep(0, 6)
  )
)

pars <- c("p", "alpha", "mu", "delta", "delta_or")
gran.re.vague2.jag <- jags.model(textConnection(gran_re_vague2_jagsmod),
                                 data = gran_dat,
                                 inits = ini,
                                 n.chains = 1,
                                 quiet = TRUE)
update(gran.re.vague2.jag, n.iter = 1000)
gran_re_vague2_post <- coda.samples(model = gran.re.vague2.jag,
                                    variable.names = pars,
                                    n.iter = 50000)

gran_re_vague2_draws <- as_draws(gran_re_vague2_post)
```

```
gran_re_vague2_delta_or_draws <- subset_draws(gran_re_vague2_draws, "delta_or")
summary(gran_re_vague2_delta_or_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad     q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.987  0.324  21.9 0.261 0.0458  1.40  1.00    8784.    8707.
```

Under the vague N(0, 10^2) prior delta's posterior media is -1.11 (95% CI -3.6, 0.76). Under the informative prior, posterior median -0.04 (95% CI -0.33, 0.25). Under the U(-100, 100) prior, delta's posterior median is -1.12 (95% CI -3.74, 0.89).

The posterior distribution is much more concentrated around 0 under the informative prior, but there is no substantive difference between the two vague priors, although the tails are a bit longer under the Uniform prior.

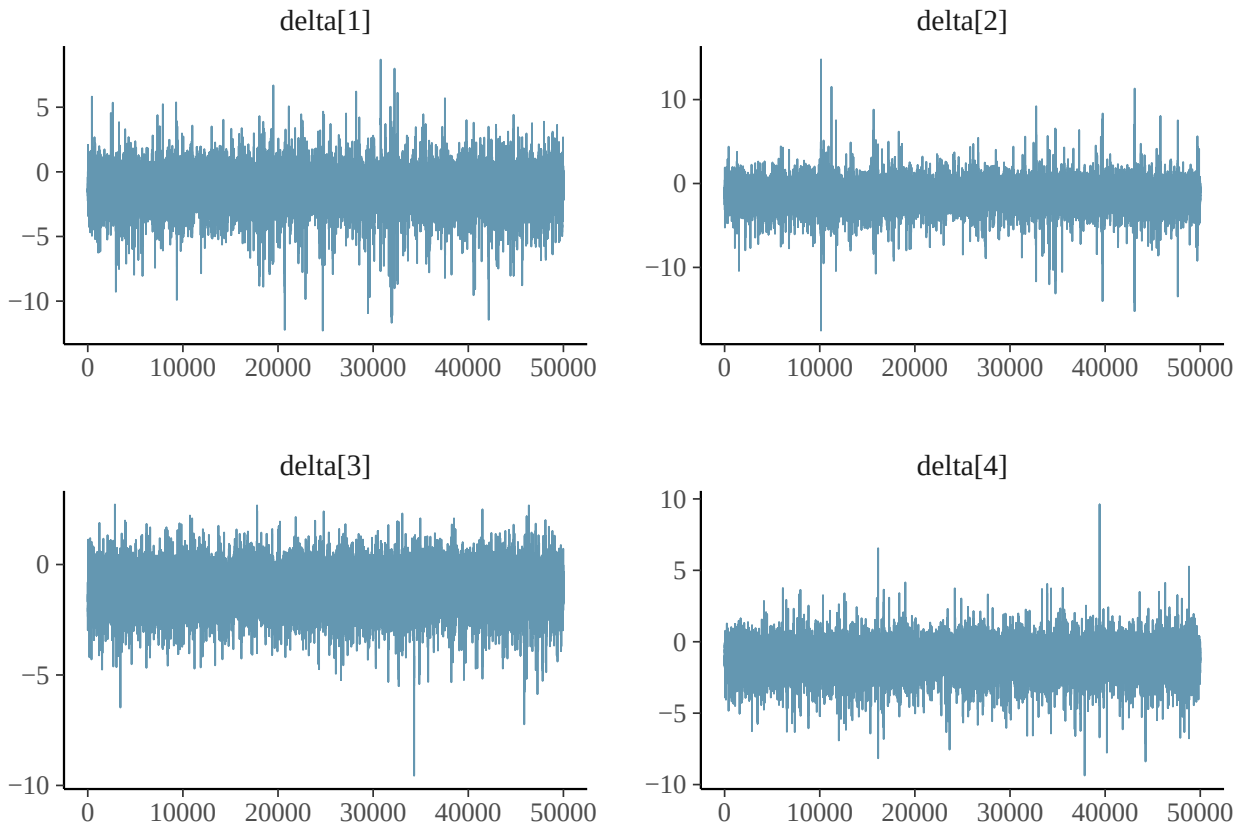
3.

```
gran_re_jagsmod <- "
model
{
  for (i in 1:Np){
    for (s in 1:Ns){ # for each study, Ns = total number of studies
      for (a in 1:2){ # for each arm
        y[i, s, a] ~ dbin(p[i, s, a], n[i, s, a])
        logit(p[i, s, a]) <- alpha[i, s] + mu[i, s, a]
      }
      # treatment effect
      mu[i, s, 1] <- 0
      mu[i, s, 2] ~ dnorm(delta[i], prec.mu[i])

      alpha[i, s] ~ dnorm(0, 1/10^2)
    }

    delta[i] ~ dnorm(0, 1/10^2)
    delta_or[i] <- exp(delta[i])
  }

  sd.mu[1] ~ dunif(0, 10)
  prec.mu[1] <- 1/sd.mu[1]^2

  sd.mu[2] ~ dunif(0, 100)
  prec.mu[2] <- 1/sd.mu[2]^2

  prec.mu[3] ~ dgamma(1, 1)
  sd.mu[3] <- 1/sqrt(prec.mu[3])

  prec.mu[4] ~ dgamma(0.001, 0.001)
  sd.mu[4] <- 1/sqrt(prec.mu[4])
}
"

pars <- c("p", "alpha", "mu", "delta", "delta_or", "sd.mu", "prec.mu")
gran_re_jag <- jags.model(textConnection(gran_re_jagsmod),
                          data = gran_dat_repeated,
                          inits = ini_gran_repeated,
```

```
                          n.chains = 1,
                          quiet = TRUE)
update(gran_re_jag, n.iter = 1000)
gran_re_post <- coda.samples(model = gran_re_jag,
                             variable.names = pars,
                             n.iter = 50000)

gran_re_draws <- as_draws(gran_re_post)
gran_re_delta_or_draws <- subset_draws(gran_re_draws, "delta_or")
summary(gran_re_delta_or_draws)
```

```
## # A tibble: 4 x 10
##   variable      mean median        sd   mad     q5   q95  rhat ess_bulk ess_tail
##   <chr>        <dbl>  <dbl>     <dbl> <dbl>  <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or[1]  0.804  0.329    30.1    0.261 0.0497  1.41  1.00   16781.   13521.
## 2 delta_or[2] 56.9    0.329 11582.     0.259 0.0503  1.44  1.00   18400.   13874.
## 3 delta_or[3]  0.447  0.356     0.416  0.223 0.105   1.05  1.00   13130.   22712.
## 4 delta_or[4]  0.800  0.367    66.9    0.244 0.0858  1.11  1.00   12383.   19634.
```

```
gran_re_delta_draws <- subset_draws(gran_re_draws, "delta")
mcmc_trace(gran_re_delta_draws)
```




```
summary(gran_re_delta_draws)
```

```
## # A tibble: 4 x 10
##   variable  mean median    sd   mad    q5    q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>    <dbl>    <dbl>
## 1 delta[1] -1.20  -1.11 1.10  0.838 -3.00 0.344   1.00   16781.   13521.
## 2 delta[2] -1.19  -1.11 1.14  0.840 -2.99 0.362   1.00   18400.   13874.
```

```
## 3 delta[3] -1.06  -1.03 0.727 0.654 -2.26 0.0459  1.00   13130.   22712.
## 4 delta[4] -1.07  -1.00 0.829 0.689 -2.46 0.105   1.00   12383.   19634.
```

The results under priors 1 and 2 are quite similar, but are fairly different to priors 3 and 4, which are fairly similar to each other, with 4 being more extreme.

```
gran_re_p_draws <- subset_draws(gran_re_draws, "p")
summary(gran_re_p_draws) %>%
  print(n = Inf)
```

```
## # A tibble: 48 x 10
##    variable   mean  median     sd     mad      q5    q95  rhat ess_bulk ess_tail
##    <chr>     <dbl>   <dbl>  <dbl>   <dbl>   <dbl>  <dbl> <dbl>    <dbl>    <dbl>
##  1 p[1,1,1] 0.401  0.395   0.135  0.141   1.87e-1 0.632   1.00    9914.   14924.
##  2 p[2,1,1] 0.398  0.392   0.135  0.141   1.86e-1 0.630   1.00   10090.   14101.
##  3 p[3,1,1] 0.412  0.407   0.133  0.139   2.01e-1 0.638   1.00   10825.   15964.
##  4 p[4,1,1] 0.410  0.406   0.133  0.139   1.97e-1 0.635   1.00   10630.   14891.
##  5 p[1,2,1] 0.210  0.197   0.100  0.0997  6.86e-2 0.394   1.00   17059.   19541.
##  6 p[2,2,1] 0.209  0.196   0.100  0.0999  6.95e-2 0.395   1.00   18640.   20890.
##  7 p[3,2,1] 0.207  0.194   0.0986 0.0978  7.01e-2 0.390   1.00   18796.   23006.
##  8 p[4,2,1] 0.208  0.194   0.0984 0.0971  7.08e-2 0.390   1.00   18813.   21543.
##  9 p[1,3,1] 0.698  0.704   0.104  0.106   5.15e-1 0.857   1.00   10854.   18419.
## 10 p[2,3,1] 0.698  0.705   0.103  0.106   5.17e-1 0.857   1.00   10732.   18820.
## 11 p[3,3,1] 0.681  0.687   0.102  0.104   5.02e-1 0.838   1.00   12274.   20704.
## 12 p[4,3,1] 0.676  0.684   0.109  0.112   4.85e-1 0.842   1.00    7238.    8753.
## 13 p[1,4,1] 0.0661 0.0462  0.0644 0.0478  3.49e-3 0.196   1.00   21288.   18974.
## 14 p[2,4,1] 0.0668 0.0471  0.0649 0.0486  3.42e-3 0.197   1.00   19648.   17869.
## 15 p[3,4,1] 0.0636 0.0447  0.0616 0.0456  3.57e-3 0.189   1.00   22361.   21192.
## 16 p[4,4,1] 0.0637 0.0445  0.0616 0.0456  3.58e-3 0.189   1.00   22583.   20901.
## 17 p[1,5,1] 0.686  0.694   0.119  0.123   4.75e-1 0.867   1.00    8457.   12675.
## 18 p[2,5,1] 0.686  0.694   0.118  0.122   4.77e-1 0.865   1.00    8251.   12366.
## 19 p[3,5,1] 0.683  0.692   0.115  0.119   4.79e-1 0.856   1.00    9300.   14849.
## 20 p[4,5,1] 0.678  0.685   0.115  0.119   4.78e-1 0.855   1.00    8607.   14217.
## 21 p[1,6,1] 0.292  0.288   0.0654 0.0656  1.89e-1 0.404   1.00    8357.   15226.
## 22 p[2,6,1] 0.293  0.291   0.0652 0.0662  1.91e-1 0.405   1.00    8156.   15478.
## 23 p[3,6,1] 0.299  0.296   0.0653 0.0657  1.96e-1 0.411   1.00    9626.   15616.
## 24 p[4,6,1] 0.302  0.299   0.0673 0.0681  1.96e-1 0.418   1.00    7127.   13580.
## 25 p[1,1,2] 0.355  0.347   0.123  0.126   1.66e-1 0.570   1.00   30995.   34085.
## 26 p[2,1,2] 0.356  0.348   0.124  0.129   1.65e-1 0.573   1.00   27831.   36517.
## 27 p[3,1,2] 0.345  0.337   0.120  0.123   1.64e-1 0.556   1.00   33481.   36530.
## 28 p[4,1,2] 0.346  0.339   0.119  0.123   1.64e-1 0.554   1.00   29635.   35581.
## 29 p[1,2,2] 0.0830 0.0659  0.0665 0.0555  9.88e-3 0.215   1.00   23012.   22906.
## 30 p[2,2,2] 0.0834 0.0666  0.0663 0.0555  1.02e-2 0.215   1.00   23716.   22625.
## 31 p[3,2,2] 0.0862 0.0706  0.0641 0.0544  1.35e-2 0.212   1.00   25885.   26247.
## 32 p[4,2,2] 0.0864 0.0711  0.0640 0.0551  1.27e-2 0.213   1.00   21266.   23235.
## 33 p[1,3,2] 0.160  0.147   0.0873 0.0863  4.26e-2 0.324   1.00   12915.   21487.
## 34 p[2,3,2] 0.161  0.147   0.0880 0.0867  4.23e-2 0.325   1.00   13141.   23185.
## 35 p[3,3,2] 0.180  0.169   0.0881 0.0873  5.69e-2 0.343   1.00   18711.   24206.
## 36 p[4,3,2] 0.185  0.171   0.0973 0.0959  5.30e-2 0.367   1.00    7293.    8073.
## 37 p[1,4,2] 0.0181 0.00726 0.0280 0.00991 6.55e-5 0.0728  1.00   11350.    8385.
## 38 p[2,4,2] 0.0181 0.00732 0.0282 0.00997 6.88e-5 0.0731  1.00   10599.    8947.
## 39 p[3,4,2] 0.0210 0.0104  0.0290 0.0128  3.87e-4 0.0790  1.00   21087.   19413.
## 40 p[4,4,2] 0.0207 0.00999 0.0288 0.0127  2.57e-4 0.0774  1.00   16094.   17136.
## 41 p[1,5,2] 0.415  0.412   0.113  0.116   2.35e-1 0.607   1.00   34755.   36052.
## 42 p[2,5,2] 0.416  0.413   0.113  0.116   2.34e-1 0.605   1.00   35098.   36571.
```

```
## 43 p[3,5,2] 0.420  0.417   0.111  0.114   2.44e-1 0.608   1.00   35225.   37156.
## 44 p[4,5,2] 0.421  0.419   0.111  0.114   2.44e-1 0.609   1.00   30593.   36002.
## 45 p[1,6,2] 0.360  0.358   0.0685 0.0692  2.51e-1 0.476   1.00   21654.   29939.
## 46 p[2,6,2] 0.360  0.358   0.0690 0.0702  2.50e-1 0.477   1.00   23318.   29698.
## 47 p[3,6,2] 0.353  0.351   0.0674 0.0679  2.46e-1 0.468   1.00   28493.   36544.
## 48 p[4,6,2] 0.351  0.349   0.0688 0.0698  2.41e-1 0.468   1.00   14636.   20336.
```

Note that the probabilities for each under are quite stable across all the priors – the data provide a lot of information about these quantites, and so the prior plays a less strong part in these estimates compared to the random effect variance parameter

4.

```
gran_re_jagsmod <- "
model
{
  for (s in 1:Ns){ # for each study, Ns = total number of studies
    for (a in 1:2){ # for each arm
    y[s, a] ~ dbin(p[s, a], n[s, a])
      logit(p[s, a]) <- alpha[s] + mu[s, a]
    }
    # treatment effect
    mu[s, 1] <- 0
    mu[s, 2] ~ dnorm(delta, prec.mu)

    alpha[s] ~ dnorm(0, 1/10^2)
  }

  delta ~ dnorm(0, 1/10^2)
  delta_or <- exp(delta)

  prec.mu <- 1/sd.mu^2
  sd.mu <- sqrt(var.d)

  # Turner et al (2012) informative prior
  # This is log normal with mean -3.93 and variance 1.51 on the log scale
  var.d ~ dlnorm(-3.93, 0.44) # 1/(1.51^2)
}
"

ini <- list(
  list(
    delta = 0,
    alpha = rep(0, 6)
  )
)

pars <- c("p", "alpha", "mu", "delta", "delta_or", "sd.mu", "prec.mu")
gran_re_jag <- jags.model(textConnection(gran_re_jagsmod),
                          data = gran_dat,
                          inits = ini,
                          n.chains = 1,
                          quiet = TRUE)
update(gran_re_jag, n.iter = 1000)
gran_re_post <- coda.samples(model = gran_re_jag,
```

```
                            variable.names = pars,
                            n.iter = 100000)

gran_re_draws <- as_draws(gran_re_post)
gran_re_delta_or_draws <- subset_draws(gran_re_draws, "delta_or")
summary(gran_re_delta_or_draws)
```

```
## # A tibble: 1 x 10
##   variable  mean median    sd   mad    q5   q95  rhat ess_bulk ess_tail
##   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 delta_or 0.503  0.473 0.222 0.192 0.209 0.892  1.00    4036.    7252.
```

Under the informative prior, the 95% CI for delta is much narrower, since the informative prior allows much lower levels of heterogeneity than our previous vague priors, which leads to greater precision in the treatment effect. The interval also doesn't include 0, in contrast to the other priors!