

Bayesian evidence synthesis: practical exercises

Anne Presanis

Full worked solutions are provided in section 4 at the end of this document, or, as an R Markdown document (with R code blocks that can be run conveniently from RStudio) in `evsyn_practical.Rmd`. Feel free to consult these as you are going along, if you are stuck on any of the exercises, or do ask any questions. Don't worry if you don't get through all the exercises during the practical sessions, they are there also for you to take away and absorb in your own time.

As you will have understood by now, there are usually multiple ways of coding something in R, so don't worry if the way you manipulate the posterior samples and produce plots is not exactly the same as in the provided solutions. You will see that in the provided solutions, we have tended to use the `posterior` R package and the `tidyverse`, including `ggplot2` to produce the plots in this practical. But if you are more comfortable coding, for example, in base R, then that is fine. The concepts of implementing and using the techniques introduced for evidence synthesis, rather than the precise way of coding them in R, are what is important to take away from this practical.

In this practical, you will:

- practise building a simple evidence synthesis;
- understand that synthesising evidence can result in lack-of-fit to the data, if some of the evidence are conflicting;
- practise using (cross-validatory) mixed- and posterior-predictive p-values to detect conflict and the DIC to compare models;
- practise resolving conflict by introducing bias parameters, possibly systematically;
- practise accounting for more flexibility in a model by accounting for over-dispersion.

Outline of exercises in this practical:

1. Evidence synthesis: exploring the HIV prevalence example
2. Conflict resolution: using the HIV prevalence example to explore bias modelling and robustifying models by introducing flexibility
3. Systematic model criticism and bias adjustment: using the sepsis meta-analysis example to carry out cross-validatory mixed-predictive checks and bias modelling.

1 Evidence synthesis

- (A) This set of questions concerns the HIV prevalence example on slides 5-6 of the lecture, using only a single observation $(y_1, n_1) = (5, 100)$.
- Implement the model using:

- (1) flat priors for the parameters;
- (2) a flat prior on π and an informative prior on δ implying 75% (66-83%) of infections are diagnosed; and
- (3) an informative prior on π implying a prevalence of 15% (9-23%), and the same prior on δ from 2.

Recall that Beta priors can be formed from elicited/prior proportions using the expressions for the mean and variance of a Beta random variable.

- (ii) Reproduce the plots of the posterior distributions of the parameters on slide 7. Recall that if you are using `ggplot2`, then `geom_density()` will give you a density of a set of samples.
 - (iii) Calculate posterior-predictive p-values for the data point in each of the three models A1-A3. What do they tell you about the consistency of the evidence?
 - (iv) Compare models A1-A3 using the DIC. Which model fits the data best, once model complexity has been taken into account? Recall that the DIC penalises a measure of model fit with a measure of the effective number of parameters in the model.
- (B) We next consider the HIV prevalence model on slides 8-10 of the lecture, using both $(y_1, n_1) = (5, 100)$ and $(y_2, n_2) = (3000, 100000)$.
- (i) Implement the model using flat priors for both π and δ . Run it and check the traces for convergence. What do you find? Using a scatter plot, plot the posterior samples of π against the corresponding samples of δ . How do you interpret what you see? Have a look also at the Rhat values.
 - (ii) Try re-running the model with 10 times as many iterations as you previously ran it, but with the posterior samples you keep thinned to every 10th iteration (use the `n.thin` setting in the `coda.samples()` function). Check again the traces, the bivariate scatter plot of the posterior samples of π and δ and the Rhat and ESS values. Do you see any difference?
 - (iii) Compare this model B1 to model A1: (1) visually, by reproducing the plots on slide 10 of the lecture comparing the posterior distributions of the parameters by model and the posterior-predictive distributions to the data; (2) more formally, using posterior-predictive p-values. Note that the DIC cannot be used to compare models A1 and B1, as they contain different amounts of data.
- (C) Finally, we consider the HIV prevalence model on slides 11-13 of the lecture, using all three data points: $(y_1, n_1) = (5, 100)$ informing π , $(y_2, n_2) = (3000, 100000)$ informing $\pi(1 - \delta)$ and $(y_3, n_3) = c(90, 100)$ informing δ .
- (i) Implement the model using flat priors for both π and δ , retaining the thinning to every 10th MCMC iteration. Repeat the same trace and scatter plots as in (B.ii) to see if adding a third data point has made a difference to the mixing and correlation.
 - (ii) Repeat (B.iii), this time comparing this model C1 to both B1 and A1, both visually and using posterior-predictive p-values. Again, the DIC cannot be used, since each model includes a different number of data points. Are there signs of conflicting evidence? How has the addition of the third data point affected the estimates?

We will break here for the second half of the evidence synthesis lecture.

2 Conflict resolution

- (A) This set of exercises considers the addition of a bias parameter to resolve the conflict detected in model C1.

- (i) Amend the model C1 to include a bias parameter β such that the study y_2 measures a quantity that is between 20 and 50% of the “true” undiagnosed prevalence $\pi(1 - \delta)$ (see slides 23-26 of the lecture). Find the parameters of a beta distribution that can represent this prior belief for β .
 - (ii) Reproduce the plots on slide 25 of the lecture, comparing the posterior distributions of each parameter from the model with the bias parameter β to those from the model C1, and the posterior-predictive distributions, calculating also the posterior-predictive p-values. Does the use of the bias parameter lead to a better fit to the data?
 - (iii) Compare the bias model with model C1 formally, using the DIC. Which does the DIC prefer?
- (B) This set of exercises considers the HIV prevalence model on slides 27-29 and 32-33 of the lecture notes, incorporating a count observation $y_4 = 400$ informing the number of people living with diagnosed HIV in the fixed population of size $N = 10,000$.
- (i) Implement the model assuming a Poisson sampling distribution for the count y_4 . Compare the estimates visually and in terms of posterior-predictive p-values to model C1, the model with only three data points and no bias modelling (i.e. reproduce the plots on slides 28-29). Which parameter estimates are most affected by the addition of Study 4?
 - (ii) Implement the model assuming instead a Negative Binomial $\text{NB}(\psi, r)$ sampling distribution, with a Beta prior for ψ expressing that it lies between 0.2 and 0.6 (i.e. with a prior mean around 0.4 and standard deviation around 0.1). Reproduce the plots on slides 32-33, calculating and plotting in addition the posterior p-values, to compare the Negative binomial model to the Poisson model. Compare them also using the DIC.
 - (iii) What happens if you change the prior for ψ so that you allow for much greater over-dispersion, with much more certainty (e.g. prior mean 0.1, prior sd 0.01)? Does the flexibility of the Negative binomial model do anything to counteract the large conflict, given how large the sample size n_2 is?

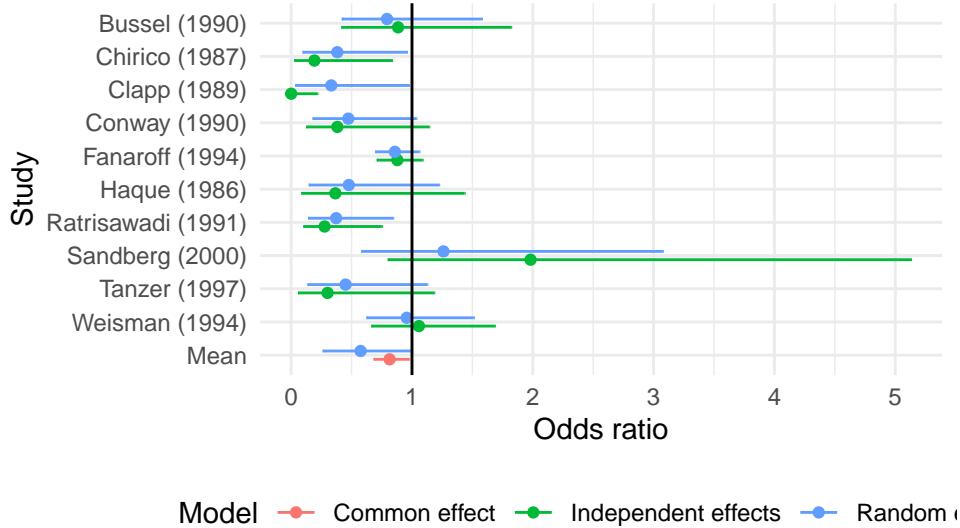
3 Systematic model criticism and bias adjustment

In this section, we will use the sepsis example to practise carrying out cross-validatory mixed-predictive checks and systematic bias adjustment. The data and implementations of the independent effects, common effect and random effects models, as seen in slide 35 of the lecture notes, are provided in the file `sepsis.R`, which you can run now using:

```
source("sepsis.R")
```

The posterior samples are saved in the three objects `sepsis_ie_out`, `sepsis_ce_out` and `sepsis_re_out`; and the treatment effects (odds ratios) from the three models are additionally saved in `sepsis_summary` for plotting, as follows:

```
# forest plot of odds ratios
sepsis_summary %>%
  ggplot() +
  aes(x = Median, y = reorder(Study, desc(Study)), col = Model) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbarh(aes(xmin = Lower, xmax = Upper, height = 0),
                 position = position_dodge(width = 0.5)) +
  geom_vline(xintercept = 1) +
  labs(x = "Odds ratio", y = "Study") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



(A) Following the outline in slides 35-38 of the lecture notes, implement the cross-validatory mixed-predictive checks, following the steps below:

- (i) Complete the code outline provided here. Replace the question marks with the appropriate expressions to obtain the replicate data $y_{i,k}^{rep}$ from the replicate parameters $p_{i,k}^{rep}$, μ_i^{rep} and δ_i^{rep} :

```
sepsis_cv_model <- "
model
{
  # Cross-validation: repeat data set/analysis, leaving i'th
  # study out each time
  for(i in 1:Ns)
  {
    # For each study, Ns = total number of studies
    for(j in 1:Ns)
    {
      # for each of the two arms, control=1, treatment=2
      for(k in 1:Na)
      {
        # Binomial likelihood
        # We don't leave the i'th study out here, but we
        # later do not link p[i,i,k] to any other parameters,
        # so that effectively the i'th study is left out
        ycv[i,j,k] ~ dbin(p[i,j,k], ncv[i,j,k])
      }

      # On a logit scale, the proportion p is the probability
      # of infection in terms of the study baselines mu and
      # the study-specific treatment effects delta (log odds ratios,
      # relative to the study baseline), if not left-out.
      # i=j refers to the i=j'th study being left out: (1 - equals(i,j))
      # is equal to 0 if i=j, 1 otherwise; so that p is linked to the
      # treatment effects only if the study is not left out.
      # Since p[i,i,k] is not linked to either mu[i,i] or delta[i,i],
      # mu[i,i] and delta[i,i] are the Replicate parameters,
      # drawn from their (random effects) distributions with
      # no data directly informing them, only the indirect information
    }
  }
}
```

```

# from all studies other than i
logit(p[i,j,1]) <- ((1 - equals(i,j)) * mu[i,j])
logit(p[i,j,2]) <- ((1 - equals(i,j)) * (mu[i,j] + delta[i,j]))

# for the treatment arm, the log odds ratios are random
# effects with mean d[i] for the i'th cross-validation
delta[i,j] ~ dnorm(d[i], prec.d[i])

# study-specific baselines, vague priors
mu[i,j] ~ dnorm(0, 0.01)
}

# Replications
# for each arm
for(k in 1:Na)
{
  # replicated data for left-out i'th study
  yrep[i,k] ~ ?
}

# replicate proportions infected
logit(prep[i,1]) <- ?
logit(prep[i,2]) <- ?

# Priors for basic parameters (mean and sd log odds ratio of
# treatment vs control)
d[i] ~ dnorm(0, 0.01)
prec.d[i] <- 1 / (sd.d[i] * sd.d[i])
sd.d[i] ~ dunif(0,10)
}
}
"

```

- (ii) The repeated data (`ycv`, `ncv`) have already been provided in a data list (`sepsis_cv_dat`) in `sepsis.R`, as have a set of initial values for two chains. Run the cross-validation model and carry out your usual checks of convergence: have you run the chains for long enough?
 - (iii) Carry out the mixed-predictive checks for the first of the two alternative test statistics discussed in the lecture on slide 36, comparing each $y_{i,k}$ to its predictive distribution $Y_{i,k}^{rep}$. Are any of the individual observations by arm and study substantially different to the corresponding observations in the same arm in each other study?
 - (iv) Carry out the mixed-predictive checks for the second of the two test statistics on slide 37, comparing $\text{logit}(y_{\{i,2\}}/n_{\{i,2\}}) - \text{logit}(y_{\{i,1\}}/n_{\{i,1\}})$ to the predictive distribution for δ_i^{rep} . Note that the `logit` function is defined as $\text{logit}(p) = \log(p / (1-p))$. Are there any outliers? Which of the two test statistics gives a more meaningful test for outliers in this example?
- (B) This final set of exercises explores the systematic bias adjustment described in slides 40-45 of the lecture notes. Turner et al, JRSS(A) (2008) introduced the idea of systematic bias adjustment for meta-analysis where some of the included studies are considered less rigorous or relevant than others. By incorporating prior information on the effect possible biases in such studies might have on the estimated treatment effect, the evidence from less rigorous or relevant studies may still be included, but down-weighted. Since then, much work has been done on systematic bias adjustment, see the Cochrane Collaboration's "Risk of bias" tool (<https://training.cochrane.org/handbook/current/chapter-08>). The "BRANDO" study (Savovic et al, Ann Intern Med, 2012) estimated the effect that different flaws in study design

have on the estimated treatment effect: these estimates can be used as prior information to incorporate in a systematic bias adjustment model as suggested by Turner et al (2008).

- (i) Below is an outline of the code to implement the bias adjustment in the random effects sepsis model. Complete the model code by replacing the question marks with appropriate expressions for the biased version of the log odds ratio, δ_i^{bias} ; and for the bias parameters β_i . Assume, as in slide 44 that there is no bias for the “safe” set of studies; and that there is a common mean and standard deviation for the biases in the “risky” set of studies. Following Savovic et al (2012), choose a mean and standard deviation such that the prior multiplicative bias effect on the odds ratio is 0.82 (0.67-1) (i.e. convert this prior information to the log scale). This prior came from averaging different types of bias effect in Savovic et al (2012):

```
sepsis_re_bias_model <- "
model
{
  # For each study, Ns = total number of studies
  for(i in 1:Ns)
  {
    # for each of the two arms
    for(k in 1:Na)
    {
      # Binomial likelihood
      y[i,k] ~ dbin(p[i,k], n[i,k])
    }

    # On a logit scale, the proportion p is the probability
    # of infection in terms of the study baselines mu and
    # the study-specific treatment effects delta (log odds ratios,
    # relative to the study baseline).
    # Here we replace delta with the biased version delta_bias
    logit(p[i,1]) <- mu[i]
    logit(p[i,2]) <- mu[i] + delta_bias[i]

    # and express the biased log odds ratio in terms of the unbiased
    delta_bias[i] <- ?

    # replicate biased log odds ratio in terms of replicate unbiased log odds ratio for
    # mixed predictive checks
    delta_bias_rep[i] <- ?

    # for the treatment arm, log odds ratios are random effects
    # with a common mean d
    delta[i] ~ dnorm(d, prec.d)

    # mixed-predictive replicates
    delta_rep[i] ~ dnorm(d, prec.d)

    # study-specific baselines, vague priors
    mu[i] ~ dnorm(0, 0.01)
  }

  # Priors for basic parameters:

  # mean log odds ratio of treatment vs control
  d ~ dnorm(0, 0.01)
```

```

# sd of study-specific log odds ratios
prec.d <- 1 / (sd.d * sd.d)
sd.d ~ dunif(0,10)

# informative priors for studies with at least one high risk of bias
# otherwise no bias
for(j in 1:Nsafe) {
  beta[safe[j]] <- beta.safe
}
beta.safe <- ?
for(j in 1:Nrisky) {
  beta[risky[j]] <- beta.risky[j]
  beta.risky[j] ~ ?
}
"

```

- (ii) Run the model to obtain estimates of the bias-adjusted mean and study-specific treatment effects. How do the bias-adjusted treatment effects compare to the unadjusted estimates from the random effects model stored in `sepsis_post` from `sepsis.R`? Reproduce the plot on slide 45 to visually compare the estimates.
- (iii) Calculate the mixed-predictive p-values (without cross-validation) for both the unadjusted replicate study-specific treatment effects from the bias model; and the study-specific replicate treatment effects from the random effects model in `sepsis_re_out` (named `delta_rep[i]` for study i). Are there any differences? Even though these p-values are more conservative than the cross-validatory ones in (A), do you still find outliers?

4 Solutions

4.1 Evidence synthesis

- (A) This set of questions concerns the HIV prevalence example on slides 5-6 of the lecture, using only a single observation $(y_1, n_1) = (5, 100)$.

- (i) Implement the model using:
 - (1) flat priors for the parameters;
 - (2) a flat prior on π and an informative prior on δ implying 75% (66-83%) of infections are diagnosed; and
 - (3) an informative prior on π implying a prevalence of 15% (9-23%), and the same prior on δ from 2.

Recall that Beta priors can be formed from elicited/prior proportions using the expressions for the mean and variance of a Beta random variable.

- First we write model A1 (reading ahead, since we will eventually need predictive distributions to calculate posterior-predictive p-values, we include them in the code from the start):

```

library(tidyverse)
library(rjags)
library(bayesplot)
library(posterior)
set.seed(2)

```

```

# (i) Model A1: using single datum (y1), flat priors
hivModelA <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi, b.pi)
  delta ~ dbeta(a.delta, b.delta)

  # Likelihood
  # prevalence data only
  for(i in 1:1)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }

  # Proportions in terms of basic and functional parameters
  p[1] <- pi
  p[2] <- pi * (1 - delta)
  p[3] <- delta
}
"

```

- Format the data as a list:

```

hivDataA <- list(
  y = c( 5),
  n = c(100),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1
)

```

- Set initial values for two MCMC chains, as well as random seeds for each chain, to obtain reproducible MCMC chains each time the model is run:

```

hivInits <- list(
  # chain 1
  list(
    pi = 0.1,
    delta = 0.9,
    ypred = 3,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(7195)
  ),
  # chain 2
  list(
    pi = 0.2,
    delta = 0.2,
    ypred = 7,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(168422)
  )
)

```

- Initialise the model, run for a burn-in period of 1000 samples, followed by a further 5000 samples to keep:

```
# Initialise model
nChains <- 2
hivA1.jm <- jags.model(textConnection(hivModelA),
                         data = hivData,
                         inits = hivInits,
                         n.chains = nChains)

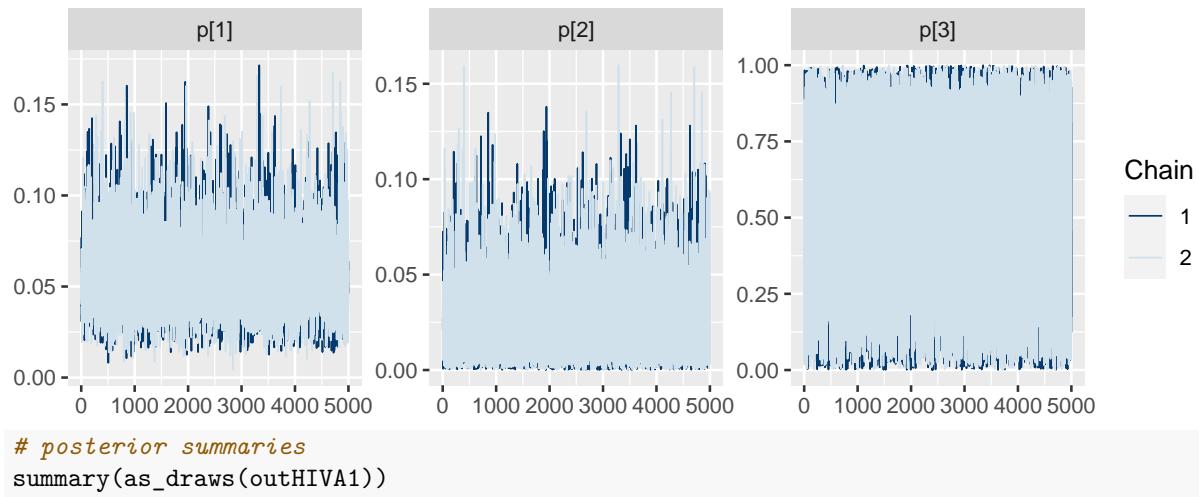
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 1
##   Unobserved stochastic nodes: 3
##   Total graph size: 12
##
## Initializing model
# burn-in
nBurn <- 1000
update(hivA1.jm, n.iter = nBurn)

# Parameters to monitor
hivParams <- c("p", "ypred")

# samples to keep
nIter <- 5000
outhIVA1 <- coda.samples(hivA1.jm,
                         variable.names = hivParams,
                         n.iter = nIter,
                         n.thin = 1)
```

- Check traces for convergence and obtain posterior summaries:

```
# traces
mcmc_trace(outhIVA1, paste0("p[", 1:3, "]"))
```



```

## # 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>
## 1 p[1]      0.0593 0.0565 0.0235 0.0230 0.0264 0.102  1.00  5101.  4429.
## 2 p[2]      0.0298 0.0255 0.0221 0.0215 0.00262 0.0718 1.00  8076.  7162.
## 3 p[3]      0.499   0.503   0.289   0.371   0.0489 0.949  1.00  9684.  8806.
## 4 ypred     5.95    6       3.34   2.97    1       12     1.00  6703.  7198.
summary(as_draws(outHIVA1),
~quantile(.x, probs = c(0.025, 0.5, 0.975)))

```



```

## # A tibble: 4 x 4
##   variable `2.5%` `50%` `97.5%`
##   <chr>     <dbl>  <dbl>  <dbl>
## 1 p[1]      0.0222 0.0565 0.113
## 2 p[2]      0.00134 0.0255 0.0824
## 3 p[3]      0.0258 0.503   0.973
## 4 ypred     1       6       14

```

Notice the trace for $p[3] = \delta$ covers the $[0, 1]$ range, demonstrating it is unidentified (posterior reflects only the flat prior, see also 95% posterior credible interval).

- Repeat for model A2, having found that a Beta(75,25) leads to a 95% prior interval of 66-83%:

```

# (ii) Add in informative prior for delta and re-run
hivData$a.delta <- 75
hivData$b.delta <- 25

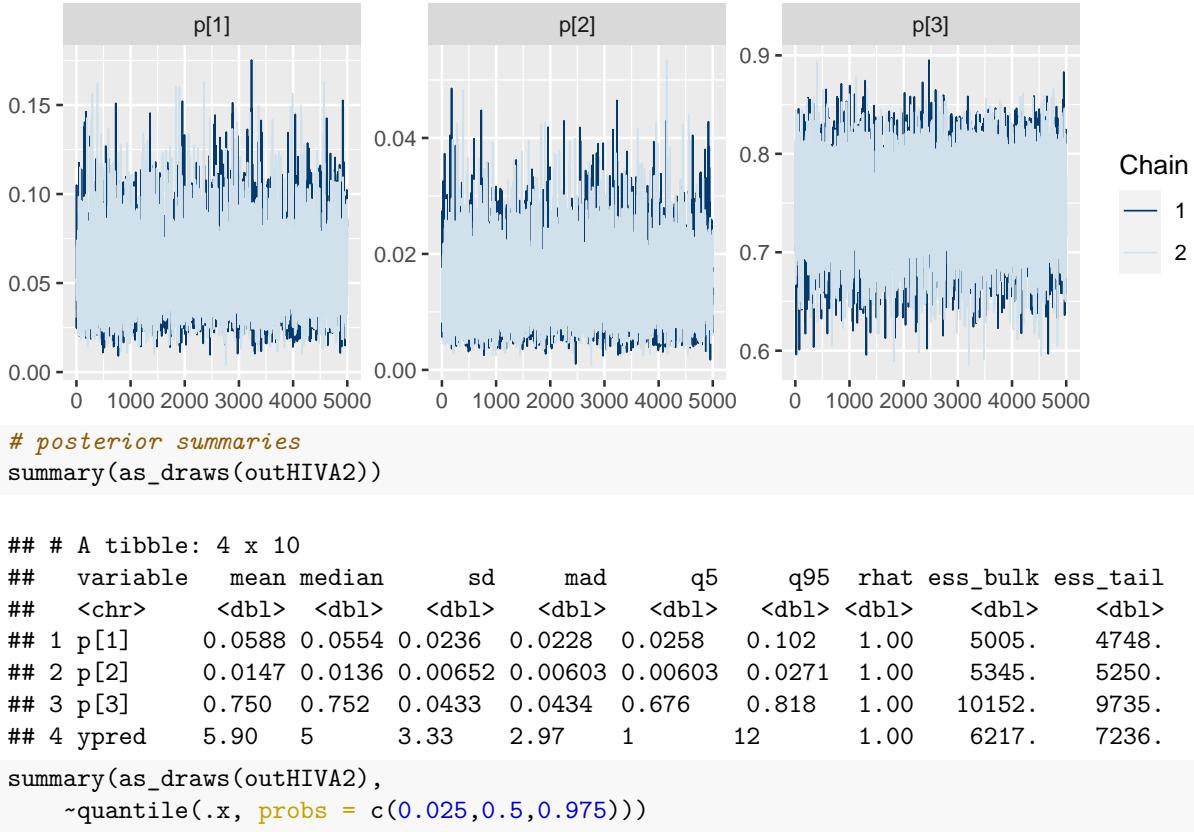
# Initialise model
hivA2.jm <- jags.model(textConnection(hivModelA),
                         data = hivData,
                         inits = hivInits,
                         n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 1
##   Unobserved stochastic nodes: 3
##   Total graph size: 12
##
## Initializing model

# burn-in
update(hivA2.jm, n.iter = nBurn)
# samples to keep
outhIVA2 <- coda.samples(hivA2.jm,
                          variable.names = hivParams,
                          n.iter = nIter,
                          n.thin = 1)

# traces
mcmc_trace(outhIVA2, paste0("p[", 1:3, "]"))

```



```
## # A tibble: 4 x 4
##   variable `2.5%` `50%` `97.5%
##   <chr>     <dbl>   <dbl>   <dbl>
## 1 p[1]      0.0222  0.0554  0.112
## 2 p[2]      0.00505 0.0136  0.0299
## 3 p[3]      0.661   0.752   0.830
## 4 ypred     1        5       13
```

Through the informative prior for $p[3] = \delta$, both parameters are now identified.

- Repeat for model A3, having found that a Beta(15,85) leads to a 95% prior interval of 9-23%:

```
# (iii) Add in informative prior for pi and re-run
hivDataA$a.pi <- 15
hivDataA$b.pi <- 85

# Initialise model
hivA3.jm <- jags.model(textConnection(hivModelA),
                        data = hivData,
                        inits = hivInits,
                        n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 1
## Unobserved stochastic nodes: 3
```

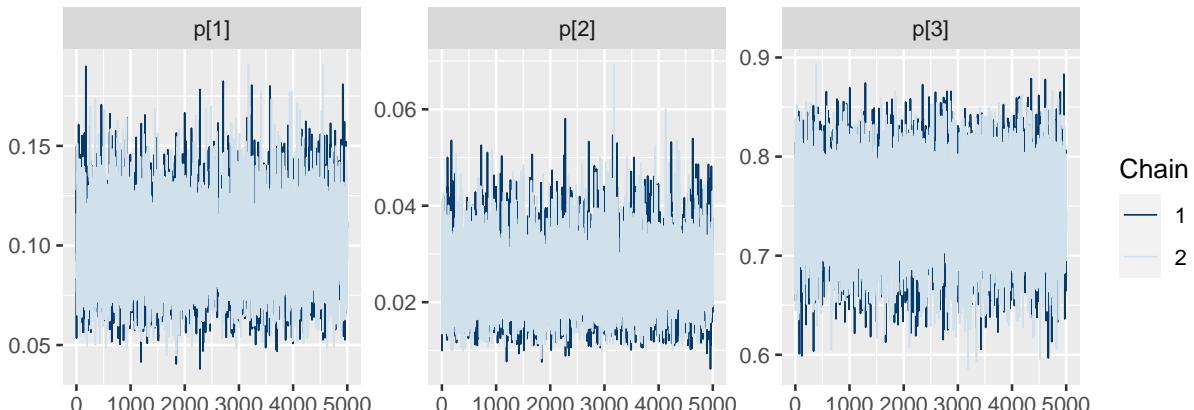
```

##      Total graph size: 12
##
## Initializing model

# burn-in
update(hivA3.jm, n.ITER = nBurn)
# samples to keep
outHIVA3 <- coda.samples(hivA3.jm,
                          variable.names = hivParams,
                          n.ITER = nIITER,
                          n.thin = 1)

# traces
mcmc_trace(outHIVA3, paste0("p[", 1:3, "]"))

```



```

# posterior summaries
summary(as_draws(outHIVA3))

```

```

## # A tibble: 4 x 10
##   variable    mean   median     sd     mad     q5     q95 rhat ess_bulk
##   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>
## 1 p[1]      0.100   0.0987  0.0213  0.0212  0.0677  0.137   1.00  5627.
## 2 p[2]      0.0250  0.0243  0.00693 0.00681  0.0150  0.0374  1.00  6564.
## 3 p[3]      0.750   0.752   0.0431  0.0430  0.676   0.818   1.00  9857.
## 4 ypred     10.0    10.0    3.71    2.97    5       17      1.00  7300.
## # i 1 more variable: ess_tail <dbl>

```

```

summary(as_draws(outHIVA3),
       ~quantile(.x, probs = c(0.025, 0.5, 0.975)))

```

```

## # A tibble: 4 x 4
##   variable `2.5%` `50%` `97.5%
##   <chr>     <dbl>   <dbl>   <dbl>
## 1 p[1]      0.0626  0.0987  0.145
## 2 p[2]      0.0136  0.0243  0.0404
## 3 p[3]      0.662   0.752   0.830
## 4 ypred     4        10      18

```

The inclusion of an informative prior for $p[1] = \pi$ introduces some conflict between the three items of evidence (1 data point and two informative priors). The informative prior for π implies a different region of support to the data, so note the posterior for $p[1] = \pi$ is a compromise between prior and likelihood.

(ii) Reproduce the plots of the posterior distributions of the parameters on slide 7. Recall that if you are using `ggplot2`, then `geom_density()` will give you a density of a set of samples.

- First, we combine the posterior samples from three models A1-A3 in a single tibble:

```
outHIVA <- as_tibble(as_draws_matrix(outHIVA1),
  rownames = "Iter") %>%
  mutate(Model = "Flat priors") %>%
  bind_rows(
  as_tibble(as_draws_matrix(outHIVA2),
    rownames = "Iter") %>%
    mutate(Model = "Informative prior delta")
  ) %>%
  bind_rows(
  as_tibble(as_draws_matrix(outHIVA3),
    rownames = "Iter") %>%
    mutate(Model = "Informative priors delta, pi")
  ) %>%
  pivot_longer(
  cols = contains("p") | contains("ypred"),
  names_to = "Parameter",
  values_to = "Posterior sample"
  ) %>%
  mutate(
Parameter = case_when(
  str_detect(Parameter, "1") ~ "p[1] = pi",
  str_detect(Parameter, "2") ~ "p[2] = pi(1-delta)",
  str_detect(Parameter, "3") ~ "p[3] = delta",
  Parameter == "ypred" ~ "ypred[1]"
),
`Posterior sample` = as.numeric(`Posterior sample`),
Model = factor(Model,
  levels = c("Flat priors",
            "Informative prior delta",
            "Informative priors delta, pi"))
)
```

- Next we plot A2 against A1:

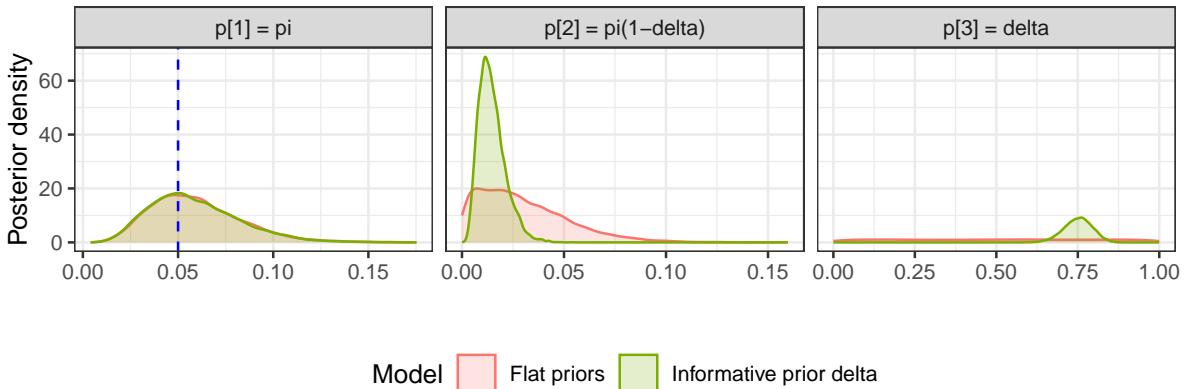
```
# tibble of observed values to plot as geom_vline()
obsA <- tibble(
  Parameter = c("p[1] = pi", "p[2] = pi(1-delta)", "p[3] = delta"),
  Obs = c(hivDataA$y / hivDataA$n, NA, NA)
)

# plot
outHIVA %>%
  filter(Model != "Informative priors delta, pi", Parameter != "ypred[1]") %>%
  ggplot() +
  aes(x = `Posterior sample`, col = Model, fill = Model) +
  geom_density(alpha = 0.2) +
  facet_grid(. ~ Parameter, scales = "free_x") +
  geom_vline(data = obsA, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
  labs(x = "", y = "Posterior density") +
  scale_color_manual(values = c("Flat priors" = "#F8766D",
```

```

    "Informative prior delta" = "#7CAE00")) +
scale_fill_manual(values = c("Flat priors" = "#F8766D",
                           "Informative prior delta" = "#7CAE00")) +
theme_bw() +
theme(legend.position = "bottom")

```

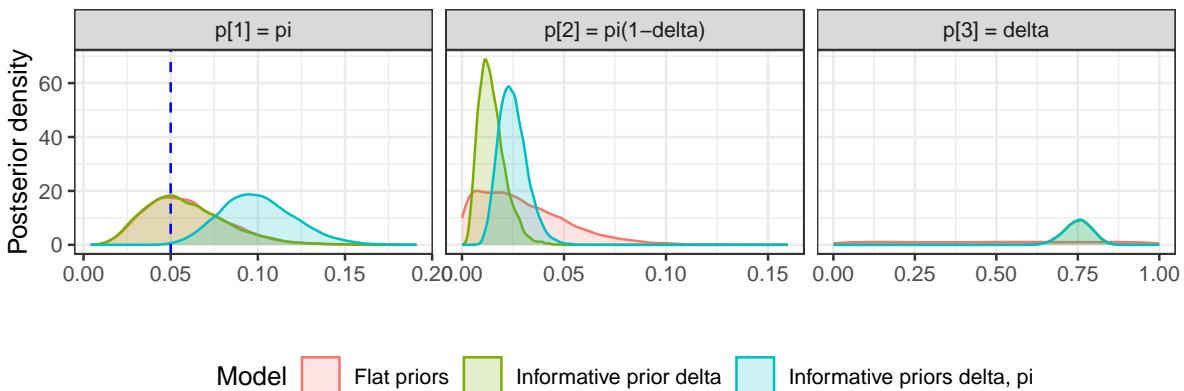


- Then add in A3:

```

outHIVA %>%
filter(Parameter != "ypred[1]") %>%
ggplot() +
aes(x = `Posterior sample`, col = Model, fill = Model) +
geom_density(alpha = 0.2) +
facet_grid(. ~ Parameter, scales = "free_x") +
geom_vline(data = obsA, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
labs(x = "", y = "Posterior density") +
scale_color_manual(values = c("Flat priors" = "#F8766D",
                             "Informative prior delta" = "#7CAE00",
                             "Informative prior delta, pi" = "#00BFC4")) +
scale_fill_manual(values = c("Flat priors" = "#F8766D",
                           "Informative prior delta" = "#7CAE00",
                           "Informative prior delta, pi" = "#00BFC4")) +
theme_bw() +
theme(legend.position = "bottom")

```



- Calculate posterior-predictive p-values for the data point in each of the three models A1-A3. What do they tell you about the consistency of the evidence?

- To calculate the posterior-predictive p-values, join the posterior-predictive samples (`ypred`) with the observed data in a single tibble, before comparing the posterior-predictive samples to the relevant observation at each MCMC iteration. We then summarise over the posterior-predictive samples, by model:

```
outhIVA %>%
  filter(Parameter == "ypred[1]") %>%
  left_join(tibble(Parameter = "ypred[1]", Obs = hivData$y), by = "Parameter") %>%
  mutate(pval = `Posterior sample` > Obs) %>%
  group_by(Model) %>% # summarise over MCMC samples, by group
  summarise(
    Median = median(`Posterior sample`),
    Lower = quantile(`Posterior sample`, probs = 0.025),
    Upper = quantile(`Posterior sample`, probs = 0.975),
    pval = mean(pval),
    .groups = "keep"
  ) %>%
  ungroup()

## # A tibble: 3 x 5
##   Model           Median Lower Upper  pval
##   <fct>          <dbl> <dbl> <dbl> <dbl>
## 1 Flat priors      6     1    14  0.504
## 2 Informative prior delta  5     1    13  0.492
## 3 Informative priors delta, pi 10     4    18  0.901
```

The p-value for model A3 is close to 1 (0.901), indicating potential lack of fit to the data informing π .

- (iv) Compare models A1-A3 using the DIC. Which model fits the data best, once model complexity has been taken into account? Recall that the DIC penalises a measure of model fit with a measure of the effective number of parameters in the model.

- We obtain the DIC from JAGS' in-built function, using unstandardised deviance:

```
load.module("dic")

## module dic loaded
dic.samples(hivA1.jm, n.iter = 1000, type = "pD")

## Mean deviance:  4.426
## penalty 0.9711
## Penalized deviance: 5.398

dic.samples(hivA2.jm, n.iter = 1000, type = "pD")

## Mean deviance:  4.464
## penalty 0.9981
## Penalized deviance: 5.462

dic.samples(hivA3.jm, n.iter = 1000, type = "pD")

## Mean deviance:  7.059
## penalty 0.5438
## Penalized deviance: 7.603
```

Models A1 and A2 fit equally well, whereas A3, while having a lower effective number of parameters

(since there are three items of evidence to inform 2 parameters) has a somewhat higher DIC, indicating the lack of fit to the data.

- (B) We next consider the HIV prevalence model on slides 8-10 of the lecture, using both y_1 and $(y_2, n_2) = (3000, 100000)$.

- (i) Implement the model using flat priors for both π and δ . Run it and check the traces for convergence. What do you find? Using a scatter plot, plot the posterior samples of π against the corresponding samples of δ . How do you interpret what you see? Have a look also at the Rhat values.

- First we implement and run the model:

```
# Model B: using two data points (y1, y2), flat priors
hivModelB <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi, b.pi)
  delta ~ dbeta(a.delta, b.delta)

  # Likelihoods
  # prevalence and undiagnosed prevalence data
  for(i in 1:2)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }

  # Proportions in terms of basic and functional parameters
  p[1] <- pi
  p[2] <- pi * (1 - delta)
  p[3] <- delta
}
"

# DATA
hivDatB <- list(
  y = c( 5, 3000),
  n = c(100, 100000),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1
)

# INITS
hivInits[[1]]$ypred <- c(3,2000)
hivInits[[2]]$ypred <- c(7,4000)

# Initialise model
hivB1.jm <- jags.model(textConnection(hivModelB),
                        data = hivDatB,
                        inits = hivInits,
                        n.chains = nChains)

## Compiling model graph
```

```

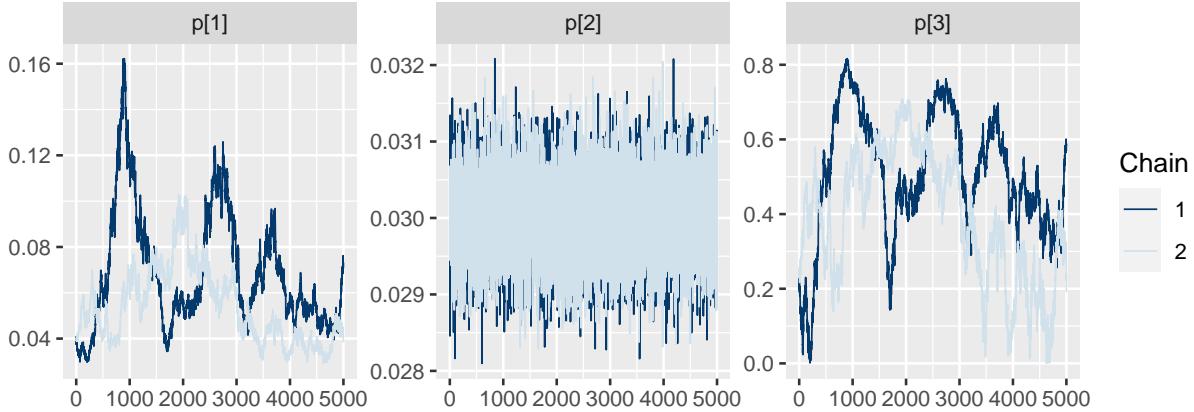
##      Resolving undeclared variables
##      Allocating nodes
## Graph information:
##      Observed stochastic nodes: 2
##      Unobserved stochastic nodes: 4
##      Total graph size: 15
##
## Initializing model
# burn-in
update(hivB1.jm, n.iter = nBurn)

# samples to keep
outhIVB1 <- coda.samples(hivB1.jm,
                          variable.names = hivParams,
                          n.iter = nIter,
                          n.thin = 1)

```

- The trace plots show that the chains are “slowly mixing”, i.e. taking very small steps to explore the joint parameter space of π and δ . This slow mixing makes it hard to judge whether the chains have converged or not.

```
mcmc_trace(outhIVB1, paste0("p[", 1:3, "]"))
```



- We can check the Rhat statistic and effective sample sizes to see whether they suggest convergence has been reached. Rhat is somewhat larger than 1 for π and δ , and the effective sample sizes are very small, suggesting lack of convergence.

```
summary(as_draws(outhIVB1))
```

```

## # A tibble: 5 x 10
##   variable     mean    median      sd      mad      q5      q95 rhat ess_bulk
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>    <dbl>
## 1 p[1]       0.0612   0.0562   0.0228  1.95e-2  3.40e-2  1.08e-1  1.19    8.40
## 2 p[2]       0.0300   0.0300   0.000543 5.49e-4  2.91e-2  3.09e-2  1.00   9329.
## 3 p[3]       0.448    0.466    0.180   1.86e-1  1.16e-1  7.24e-1  1.19    8.37
## 4 ypred[1]    6.09     6        3.27   2.97e+0  2        e+0  1.2 e+1  1.07    25.5
## 5 ypred[2]  3000.    2999     75.7   7.71e+1  2.88e+3  3.12e+3  1.00   9796.
## # i 1 more variable: ess_tail <dbl>

```

```
summary(as_draws(outhIVB1),
       ~quantile(.x, probs = c(0.025, 0.5, 0.975)))
```

```

## # A tibble: 5 x 4
##   variable    `2.5%`    `50%`    `97.5%
##   <chr>      <dbl>     <dbl>     <dbl>
## 1 p[1]       0.0322    0.0562    0.119
## 2 p[2]       0.0289    0.0300    0.0311
## 3 p[3]       0.0711    0.466     0.747
## 4 ypred[1]    1         6         14
## 5 ypred[2]  2853     2999     3148

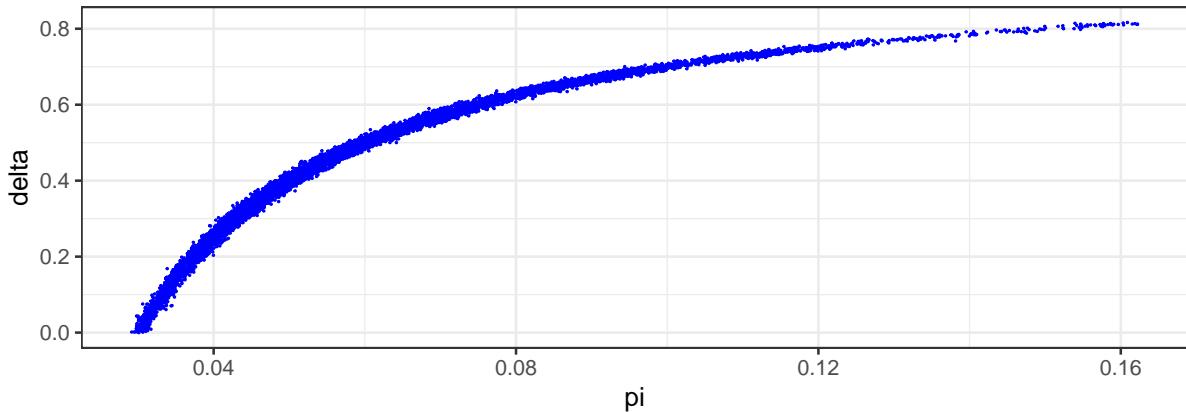
```

- Plotting the posterior samples of π against the corresponding samples of δ , we find that the two parameters are highly correlated. This occurs because the introduction of the data point (y_2, n_2) informing $\pi(1 - \delta)$ induces the correlation: for a (almost, due to the very large sample-size n_2) fixed value of $\pi(1 - \delta)$, δ has to increase as π increases. This high correlation is what makes the chains mix slowly: it is challenging for the MCMC algorithm to make large jumps in the parameter space from one iteration to the next.

```

as_tibble(as_draws_matrix(outHIVB1)) %>%
  rename(pi = "p[1]", delta = "p[3]") %>%
  mutate(pi = as.numeric(pi), delta = as.numeric(delta)) %>%
  ggplot() +
  aes(x = pi, y = delta) +
  geom_point(col = "blue", size = 0.01) +
  theme_bw()

```



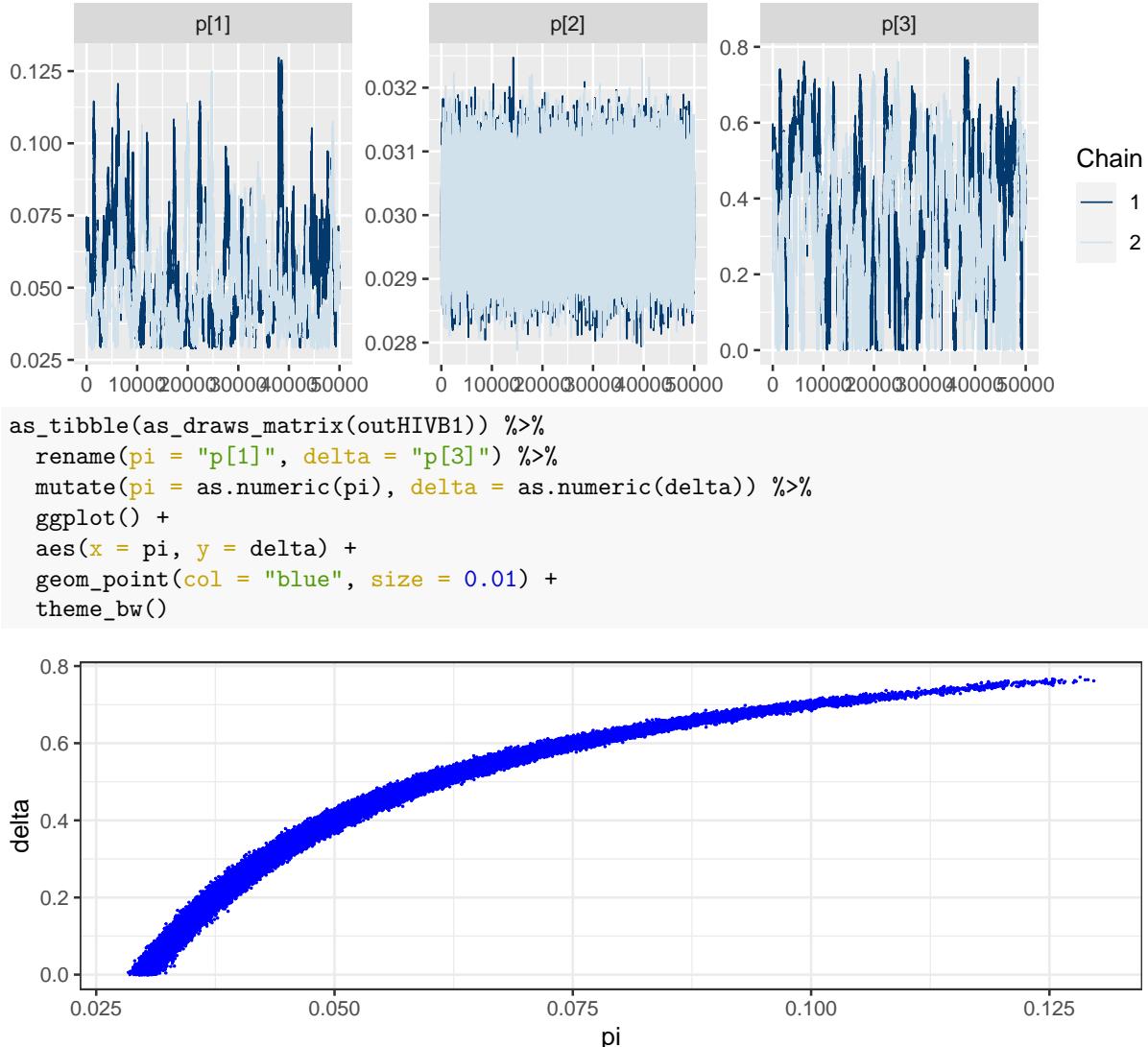
- (ii) Try re-running the model with 10 times as many iterations as you previously ran it, but with the posterior samples you keep thinned to every 10th iteration (use the `n.thin` setting in the `coda.samples()` function). Check again the traces, the bivariate scatter plot of the posterior samples of π and δ and the Rhat and ESS values. Do you see any difference?

- The traces look a little better, even though of course the parameters are still highly correlated. Running for more iterations can ensure convergence, and thinning just allows us to be sure we have adequately sampled from across the entire posterior distribution, while not needing to save all the iterations (reducing file sizes, including the size of this .pdf!).

```

outhIVB1 <- coda.samples(hivB1.jm,
                         variable.names = hivParams,
                         n.iter = nIter*10,
                         n.thin = 10)
mcmc_trace(outhIVB1, paste0("p[", 1:3, "]"))

```



- The Rhat statistic is now close to 1 and the ESS have improved:

```
summary(as_draws(outHIVB1))
```

```

## # A tibble: 5 x 10
##   variable     mean    median      sd      mad      q5      q95 rhat ess_bulk
##   <chr>      <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>    <dbl>
## 1 p[1]      0.0512   0.0476  0.0167  1.64e-2 3.16e-2 8.39e-2  1.02    117.
## 2 p[2]      0.0300   0.0300  0.000541 5.39e-4 2.91e-2 3.09e-2  1.00  87690.
## 3 p[3]      0.356    0.369   0.185   2.25e-1 5.03e-2 6.42e-1  1.02    117.
## 4 ypred[1]   5.11     5       2.76   2.97e+0 1       e+0 1       e+1  1.01    341.
## 5 ypred[2] 3000.    3000    76.4   7.56e+1 2.88e+3 3.13e+3  1.00  92200.
## # i 1 more variable: ess_tail <dbl>
summary(as_draws(outHIVB1),
       quantile(.x, probs = c(0.025, 0.5, 0.975)))

## # A tibble: 5 x 4
##   variable     `2.5%`     `50%`     `97.5%` 
##   <chr>      <dbl>      <dbl>      <dbl> 
## 1 p[1]      0.0476    0.0476    0.0512 
## 2 p[2]      0.0291    0.0300    0.0300 
## 3 p[3]      0.356     0.369     0.375  
## 4 ypred[1]  5.00      5.11      5.20  
## 5 ypred[2] 2900.     3000.    3000. 
```

```

##   <chr>     <dbl>     <dbl>     <dbl>
## 1 p[1]      0.0308    0.0476    0.0925
## 2 p[2]      0.0290    0.0300    0.0311
## 3 p[3]      0.0254    0.369     0.676
## 4 ypred[1]   1         5         11
## 5 ypred[2]  2852     3000     3151

```

- (iii) Compare this model B1 to model A1: (1) visually, by reproducing the plots on slide 10 of the lecture comparing the posterior distributions of the parameters by model and the posterior-predictive distributions to the data; (2) more formally, using posterior-predictive p-values. Note that the DIC cannot be used to compare models A1 and B1, as they contain different amounts of data.

- First, we join the posteriors from both models in one tibble:

```

outHIVAB <- as_tibble(as_draws_matrix(outhIVA1),
                       rownames = "Iter") %>%
  mutate(across(Iter:ypred, as.numeric)) %>%
  mutate(`ypred[2]` = as.numeric(NA), Model = "y1 = 5, n1 = 100") %>%
  rename(`ypred[1]` = ypred) %>%
  bind_rows(
  as_tibble(as_draws_matrix(outhIVB1),
            rownames = "Iter") %>%
  mutate(across(Iter:`ypred[2]`, as.numeric)) %>%
  mutate(Model = "y2 = 3,000, n2 = 100,000")
  ) %>%
  pivot_longer(
  cols = contains("p"),
  names_to = "Parameter",
  values_to = "Posterior sample"
  ) %>%
  mutate(
  Model = factor(Model,
                  levels = c("y1 = 5, n1 = 100",
                             "y2 = 3,000, n2 = 100,000"))
  )

```

- Next, we plot B1 against A1, adding in the observed data as vertical lines, both on the parameter scale (proportions) and on the posterior-predictive replicate scale:

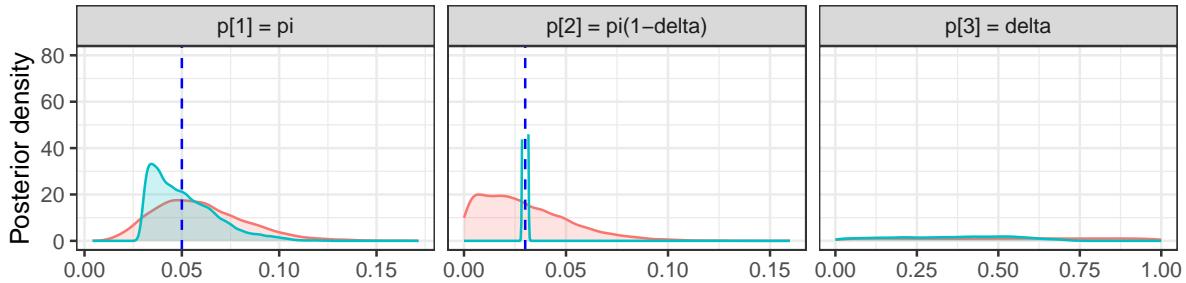
```

obsB <- tibble(
  Parameter = c("p[1] = pi", "p[2] = pi(1-delta)", "p[3] = delta"),
  Obs = c(hivDatB$y/hivDatB$n, NA)
)

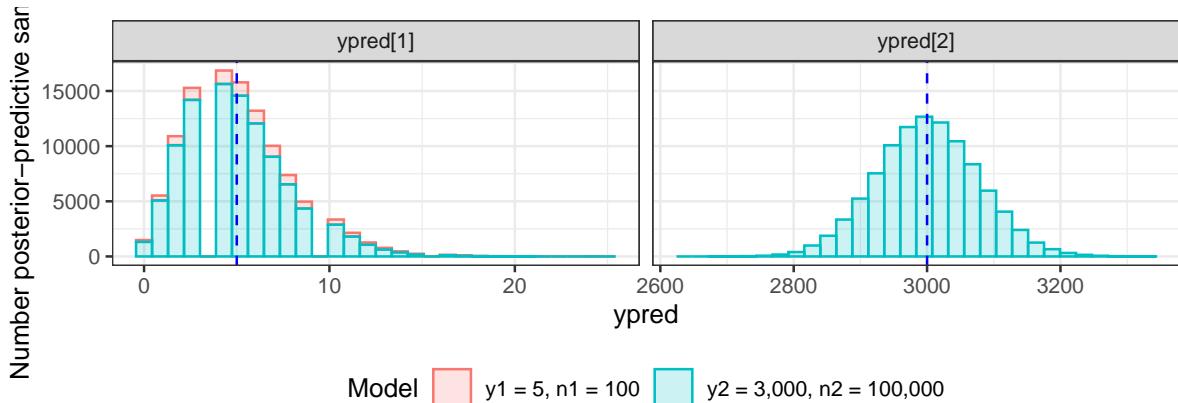
outhIVAB %>%
  filter(str_detect(Parameter, "p\\\[\")) %>%
  mutate(Parameter = factor(Parameter, levels = c("p[1]", "p[2]", "p[3]"), labels = c("p[1] = pi", "p[2] = pi(1-delta)", "p[3] = delta")),
  ggplot() +
  aes(x = `Posterior sample`, col = Model, fill = Model) +
  geom_density(alpha = 0.2) +
  facet_grid(. ~ Parameter, scales = "free_x") +
  geom_vline(data = obsB, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
  labs(x = "", y = "Posterior density") +
  lims(y = c(0, 80)) +
  theme_bw()

```

```
theme(legend.position = "bottom")
```



```
obsB <- tibble(
  Parameter = c("ypred[1]", "ypred[2]"),
  Obs = hivDatB$y
)
outHIVAB %>%
  filter(str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = `Posterior sample`, col = Model, fill = Model) +
  geom_histogram(alpha = 0.2) +
  facet_grid(. ~ Parameter, scales = "free_x") +
  geom_vline(data = obsB, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
  labs(x = "ypred", y = "Number posterior-predictive samples") +
  theme_bw() +
  theme(legend.position = "bottom")
```



- The posterior-predictive p-values demonstrate no lack of fit or conflict in either model:

```
outHIVAB %>%
  filter(str_detect(Parameter, "ypred")) %>%
  left_join(obsB, by = "Parameter") %>%
  mutate(pval = `Posterior sample` > Obs) %>%
  group_by(Model, Parameter) %>% # summarise over MCMC samples, by model
                                # and parameter
  summarise(
    Median = median(`Posterior sample`, na.rm = TRUE),
```

```

Lower = quantile(`Posterior sample`, probs = 0.025, na.rm = TRUE),
Upper = quantile(`Posterior sample`, probs = 0.975, na.rm = TRUE),
pval = mean(pval, na.rm = TRUE),
.groups = "keep"
) %>%
ungroup()

```

```

## # A tibble: 4 x 6
##   Model             Parameter Median Lower Upper    pval
##   <fct>            <chr>     <dbl> <dbl> <dbl> <dbl>
## 1 y1 = 5, n1 = 100 ypred[1]      6     1    14  0.504
## 2 y1 = 5, n1 = 100 ypred[2]      NA    NA    NA  NaN
## 3 y2 = 3,000, n2 = 100,000 ypred[1]      5     1    11  0.391
## 4 y2 = 3,000, n2 = 100,000 ypred[2]    3000  2852  3151  0.496

```

- (C) Finally, we consider the HIV prevalence model on slides 11-13 of the lecture, using all three data points: $(y_1, n_1) = (5, 100)$ informing π , $(y_2, n_2) = (3000, 100000)$ informing $\pi(1 - \delta)$ and $(y_3, n_3) = (90, 100)$ informing δ .

- (i) Implement the model using flat priors for both π and δ , retaining the thinning to every 10th MCMC iteration. Repeat the same trace and scatter plots as in (B.ii) to see if adding a third data point has made a difference to the mixing and correlation.

- Adding the third data point to both model and data and re-running:

```

# Model C: using 3 data points (y1, y2, y3), flat priors
hivModelC <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi,b.pi)
  delta ~ dbeta(a.delta,b.delta)

  # Likelihoods
  # prevalence, undiagnosed prevalence and proportion diagnosed data
  for(i in 1:3)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }

  # Proportions in terms of basic and functional parameters
  p[1] <- pi
  p[2] <- pi * (1 - delta)
  p[3] <- delta
}

"

# DATA
hivDatC <- list(
  y = c( 5,    3000,   90),
  n = c(100, 100000, 100),
  a.pi = 1, b.pi = 1,

```

```

    a.delta = 1, b.delta = 1
)

# INITS
hivInits[[1]]$ypred <- c(3,2000,80)
hivInits[[2]]$ypred <- c(7,4000,95)

# Initialise model
hivC1.jm <- jags.model(textConnection(hivModelC),
                         data = hivDatC,
                         inits = hivInits,
                         n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 3
##   Unobserved stochastic nodes: 5
##   Total graph size: 18
##
## Initializing model
# burn-in
update(hivC1.jm, n.iter = nBurn)

# samples to keep
outHIVC1 <- coda.samples(hivC1.jm,
                           variable.names = hivParams,
                           n.iter = nIter*10,
                           n.thin = 10)

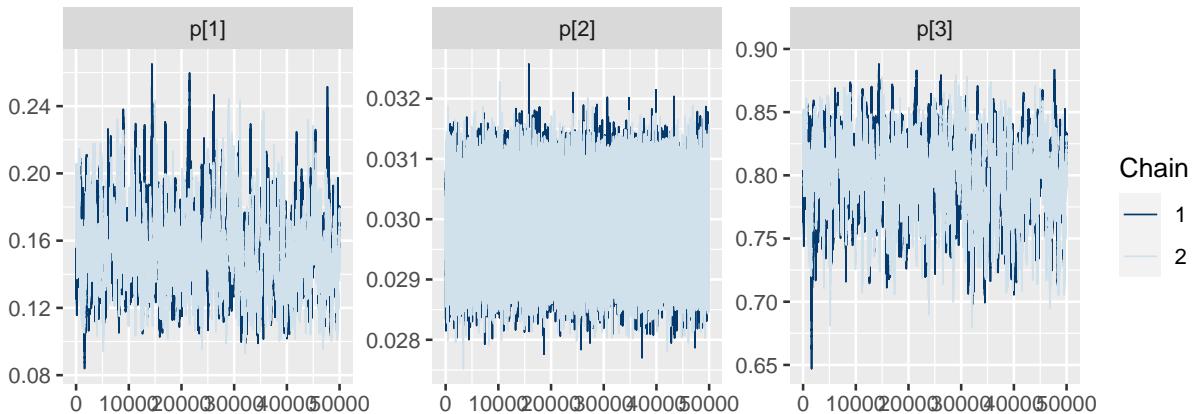
```

- We find that the mixing is improved compared to the model in (B.ii) and the strong posterior correlation between π and δ remains. The Rhat value is 1 for all quantities, and the ESS have increased, indicating convergence also:

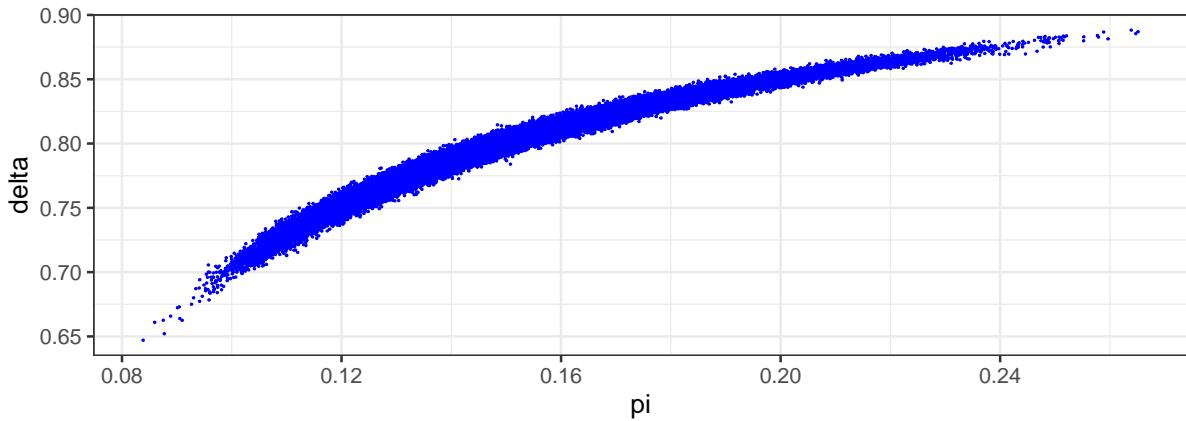
```

# traces
mcmc_trace(outHIVC1, paste0("p[", 1:3, "]"))

```



```
# Scatter plot of pi against delta, to see correlation
as_tibble(as_draws_matrix(outHIVC1)) %>%
  rename(pi = "p[1]", delta = "p[3]") %>%
  mutate(pi = as.numeric(pi), delta = as.numeric(delta)) %>%
  ggplot() +
  aes(x = pi, y = delta) +
  geom_point(col = "blue", size = 0.01) +
  theme_bw()
```



```
# posterior summaries, including Rhat
summary(as_draws(outHIVC1))
```

```
## # A tibble: 6 x 10
##   variable     mean    median      sd      mad      q5      q95    rhat ess_bulk
##   <chr>     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 p[1]      0.153    0.151  0.0229  2.23e-2 1.18e-1 1.94e-1  1.01    451.
## 2 p[2]      0.0299   0.0299  0.000539 5.41e-4 2.90e-2 3.08e-2  1.00  89459.
## 3 p[3]      0.800    0.802  0.0296  2.91e-2 7.47e-1 8.46e-1  1.01    450.
## 4 ypred[1]   15.3     15     4.26   4.45e+0 9e+0  2.3e+1  1.00   1565.
## 5 ypred[2]  2989.    2988   76.0   7.56e+1 2.86e+3 3.11e+3  1.00  93627.
## 6 ypred[3]   80.0     80     4.97   4.45e+0 7.2e+1 8.8e+1  1.00   1246.
## # i 1 more variable: ess_tail <dbl>
summary(as_draws(outHIVC1),
       quantile(.x, probs = c(0.025, 0.5, 0.975)))
```



```
## # A tibble: 6 x 4
##   variable `2.5%` `50%` `97.5%`
##   <chr>     <dbl>    <dbl>    <dbl>
## 1 p[1]      0.113    0.151    0.203
## 2 p[2]      0.0288   0.0299   0.0309
## 3 p[3]      0.736    0.802    0.853
## 4 ypred[1]   8        15       24
## 5 ypred[2]  2841     2988     3139
## 6 ypred[3]   70       80       89
```

- (ii) Repeat (B.iii), this time comparing this model C1 to both B1 and A1, both visually and using posterior-predictive p-values. Again, the DIC cannot be used, since each model includes a different number of data points. Are there signs of conflicting evidence? How has the addition of the third data point affected the estimates?

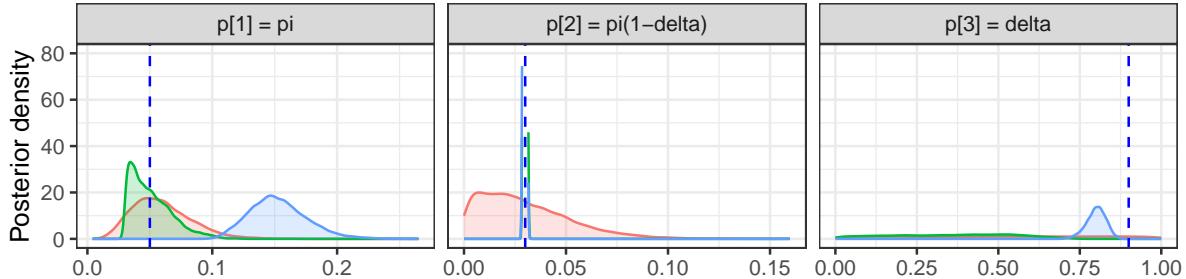
- First, we join the posteriors from all three models in one tibble:

```
outHIVABC <- as_tibble(as_draws_matrix(outHIVA1),
                        rownames = "Iter") %>%
  mutate(across(Iter:ypred, as.numeric)) %>%
  mutate(`ypred[2]` = as.numeric(NA), `ypred[3]` = as.numeric(NA), Model = "y1 = 5, n1 = 100") %>%
  rename(`ypred[1]` = ypred) %>%
  bind_rows(
  as_tibble(as_draws_matrix(outHVB1),
            rownames = "Iter") %>%
    mutate(across(Iter:`ypred[2]`, as.numeric)) %>%
    mutate(`ypred[3]` = as.numeric(NA), Model = "y2 = 3,000, n2 = 100,000")
  ) %>%
  bind_rows(
  as_tibble(as_draws_matrix(outHVC1),
            rownames = "Iter") %>%
    mutate(across(Iter:`ypred[3]`, as.numeric)) %>%
    mutate(Model = "y3 = 90, n3 = 100")
  ) %>%
  pivot_longer(
  cols = contains("p"),
  names_to = "Parameter",
  values_to = "Posterior sample"
  ) %>%
  mutate(
  Parameter = case_when(
    str_detect(Parameter, "p\\\[1") ~ "p[1] = pi",
    str_detect(Parameter, "p\\\[2") ~ "p[2] = pi(1-delta)",
    str_detect(Parameter, "p\\\[3") ~ "p[3] = delta",
    str_detect(Parameter, "ypred") ~ Parameter
  ),
  `Posterior sample` = as.numeric(`Posterior sample`),
  Model = factor(Model,
                 levels = c("y1 = 5, n1 = 100",
                           "y2 = 3,000, n2 = 100,000",
                           "y3 = 90, n3 = 100"))
  )
)
```

- Next, we plot the posterior distributions of each quantity from all three models, adding in the observed data as vertical lines, both on the parameter and posterior-predictive replicate scales:

```
obsC <- tibble(
  Parameter = c("p[1] = pi", "p[2] = pi(1-delta)", "p[3] = delta"),
  Obs = hivDatC$y / hivDatC$n
)
outHIVABC %>%
  filter(str_detect(Parameter, "p\\\[")) %>%
  ggplot() +
  aes(x = `Posterior sample`, col = Model, fill = Model) +
  geom_density(alpha = 0.2) +
  facet_grid(. ~ Parameter, scales = "free_x") +
  geom_vline(data = obsC, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
  labs(x = "", y = "Posterior density") +
  lims(y = c(0,80)) +
  theme_bw()
```

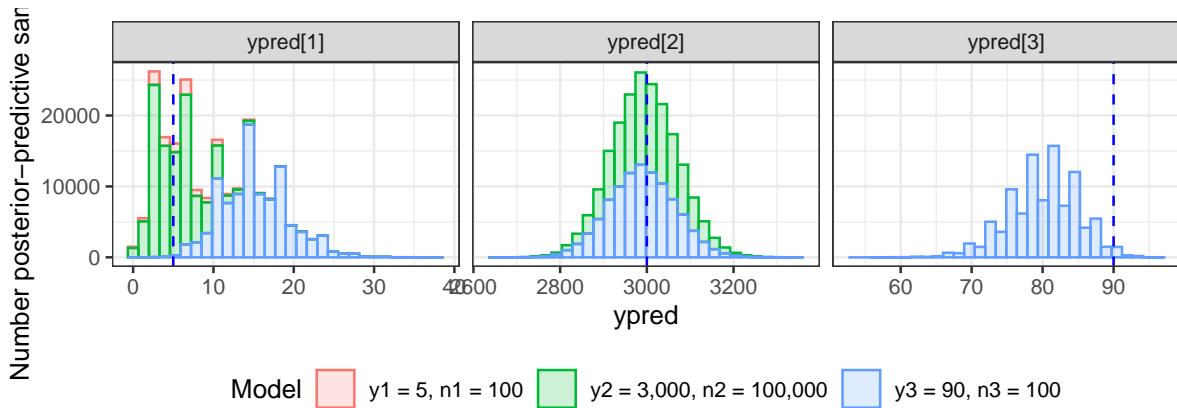
```
theme(legend.position = "bottom")
```



Model $y_1 = 5, n_1 = 100$ $y_2 = 3,000, n_2 = 100,000$ $y_3 = 90, n_3 = 100$

```
obsC <- tibble(
  Parameter = c("ypred[1]", "ypred[2]", "ypred[3]"),
  Obs = hivDatC$y
)

outHIVABC %>%
  filter(str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = `Posterior sample`, col = Model, fill = Model) +
  geom_histogram(alpha = 0.2) +
  facet_grid(. ~ Parameter, scales = "free_x") +
  geom_vline(data = obsC, aes(xintercept = Obs), col = "blue", linetype = "dashed") +
  labs(x = "ypred", y = "Number posterior-predictive samples") +
  theme_bw() +
  theme(legend.position = "bottom")
```



Model $y_1 = 5, n_1 = 100$ $y_2 = 3,000, n_2 = 100,000$ $y_3 = 90, n_3 = 100$

- The plots and the posterior-predictive p-values demonstrate the conflict between all three data points. The estimate of π is higher and more uncertain in C1 (in blue) than the corresponding estimate in model B1 (in green). Note that since the data on prevalence has the smallest sample size, it has the least influence and hence the estimate of π is closer to that implied by y_2 and y_3 than that implied by y_1 . This does not necessarily mean that y_1 is biased, any of the three studies could be biased in one or more directions:

```
outHIVABC %>%
  filter(str_detect(Parameter, "ypred")) %>%
```

```

left_join(obsC, by = "Parameter") %>%
mutate(pval = `Posterior sample` > Obs) %>%
group_by(Model, Parameter) %>% # summarise over MCMC samples, by model
# and parameter
summarise(
Median = median(`Posterior sample`, na.rm = TRUE),
Lower = quantile(`Posterior sample`, probs = 0.025, na.rm = TRUE),
Upper = quantile(`Posterior sample`, probs = 0.975, na.rm = TRUE),
pval = mean(pval, na.rm = TRUE),
.groups = "keep"
) %>%
ungroup()

## # A tibble: 9 x 6
##   Model      Parameter Median Lower Upper    pval
##   <fct>     <chr>      <dbl> <dbl> <dbl>    <dbl>
## 1 y1 = 5, n1 = 100  ypred[1]      6     1    14  0.504
## 2 y1 = 5, n1 = 100  ypred[2]      NA    NA    NA  NaN
## 3 y1 = 5, n1 = 100  ypred[3]      NA    NA    NA  NaN
## 4 y2 = 3,000, n2 = 100,000 ypred[1]      5     1    11  0.391
## 5 y2 = 3,000, n2 = 100,000 ypred[2]    3000  2852  3151  0.496
## 6 y2 = 3,000, n2 = 100,000 ypred[3]      NA    NA    NA  NaN
## 7 y3 = 90, n3 = 100     ypred[1]      15     8    24  0.996
## 8 y3 = 90, n3 = 100     ypred[2]    2988  2841  3139  0.436
## 9 y3 = 90, n3 = 100     ypred[3]      80    70    89  0.0102

```

4.2 Conflict resolution

- (A) This set of exercises considers the addition of a bias parameter to resolve the conflict detected in model C1.

- (i) Amend the model C1 to include a bias parameter β such that the study y_2 measures a quantity that is between 20 and 50% of the “true” undiagnosed prevalence $\pi(1 - \delta)$ (see slides 23-26 of the lecture). Find the parameters of a beta distribution that can represent this prior belief for β .
- We first amend the model such that y_2 informs a quantity $p[2] = \pi(1 - \delta)/\beta$, rather than the undiagnosed prevalence $\pi(1 - \delta)$:

```

hivModelCbias <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi,b.pi)
  delta ~ dbeta(a.delta,b.delta)

  # Likelihoods
  # prevalence, (biased) undiagnosed prevalence and proportion diagnosed data
  for(i in 1:3)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }
}

```

```

# Proportions in terms of basic and functional parameters
p[1] <- pi
p[2] <- pi * (1 - delta) / beta
p[3] <- delta

# bias parameter beta, expert opinion based on previous studies
# suggests the true undiagnosed prevalence is somewhere between 20% and
# 50% of the undiagnosed prevalence suggested by study 2
beta ~ dbeta(a.beta, b.beta)
}

"

```

- The values of the parameters for the prior distribution for β can be found by considering the expressions for the mean and variance of a Beta distribution. If we assume we want a prior mean of 35% (half-way through the prior interval given of 20-25%) and a prior standard deviation of $0.15/2 = 0.075$, then rearranging the expressions for the mean and variance to solve for a.beta and b.beta, we find that $a.\text{beta} = ((\text{mean}^2 * (1 - \text{mean})) - (\text{mean} * \text{variance})) / \text{variance} = 13.8$ and $b.\text{beta} = (a.\text{beta} * (1 - \text{mean})) / \text{mean} = 25.6$:

```

# DATA
hivDatCbias <- list(
  y = c( 5, 3000, 90),
  n = c(100, 100000, 100),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1,
  a.beta = 13.8, b.beta = 25.6
)

```

- We run the model:

```

# INITS
hivInitsCBias <- list(
  # chain 1
  list(
    pi = 0.1,
    delta = 0.9,
    beta = 0.2,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(7195)
  ),
  # chain 2
  list(
    pi = 0.2,
    delta = 0.2,
    beta = 0.5,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(168422)
  )
)

# Initialise model
hivCbias.jm <- jags.model(textConnection(hivModelCbias),
                           data = hivDatCbias,
                           inits = hivInitsCBias,
                           n.chains = nChains)

```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 3
##   Unobserved stochastic nodes: 6
##   Total graph size: 22
##
## Initializing model
# burn-in
update(hivCbias.jm, n.iter = nBurn)

# samples to keep
hivParamsBias <- c(hivParams, "beta")
outHIVCbias <- coda.samples(hivCbias.jm,
                            variable.names = hivParamsBias,
                            n.iter = nIter*10,
                            n.thin = 10)

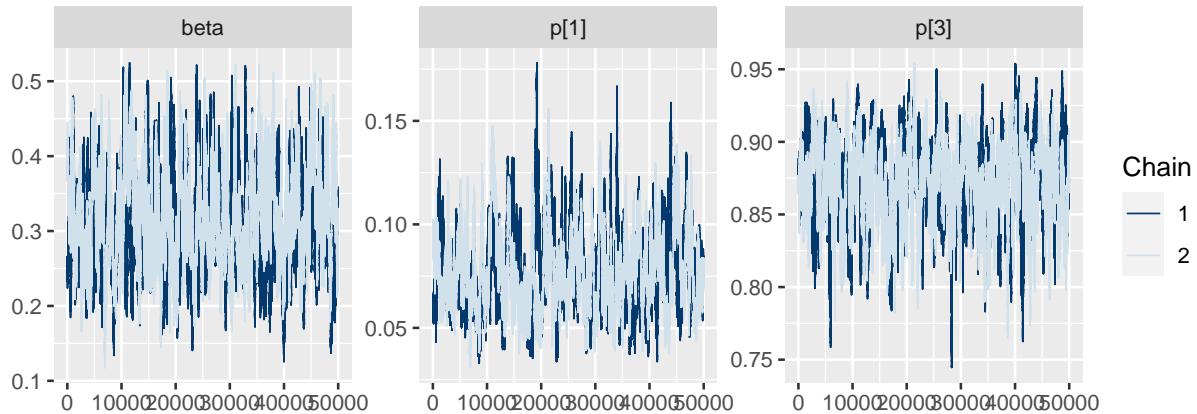
```

- Then checking the convergence and the correlation, we find reasonable mixing and some correlation between π and δ , although with a less pronounced posterior ridge:

```

# traces
mcmc_trace(outHIVCbias, c("beta", "p[1]", "p[3]"))

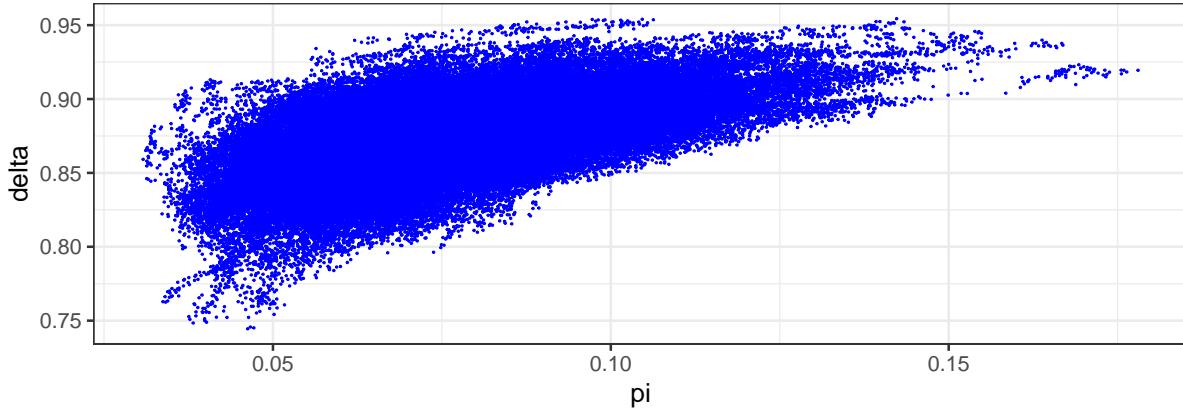
```



```

# Scatter plot of pi against delta, to see correlation
as_tibble(as_draws_matrix(outHIVCbias)) %>%
  mutate(pi = as.numeric(`p[1]`), delta = as.numeric(`p[3]`)) %>%
  ggplot() +
  aes(x = pi, y = delta) +
  geom_point(col = "blue", size = 0.01) +
  theme_bw()

```



- The Rhat values are close to 1, and the ESS are mostly alright, although they could be increased by running the MCMC chains for longer. Comparing the quantity measured by y_2 to the unadjusted $\pi(1 - \delta)$, we see that the unadjusted posterior is around a third of the adjusted posterior, consistent with the posterior estimate of β :

```
# posterior summaries and Rhat
outHIVCbiasMat <- as_draws_matrix(outHIVCbias) %>%
  mutate_variables(
    undprev = `p[1]` * (1 - `p[3]`),
    `undprev/beta` = undprev / beta
  )
summary(outHIVCbiasMat)

## # A tibble: 9 x 10
##   variable      mean   median     sd     mad     q5     q95 rhat ess_bulk
##   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>
## 1 beta       0.312   3.09e-1  6.69e-2  7.03e-2  2.06e-1  4.27e-1  1.01    280.
## 2 p[1]       0.0770  7.42e-2  2.00e-2  1.91e-2  4.81e-2  1.14e-1  1.00    199.
## 3 p[2]       0.0300  3.00e-2  5.42e-4  5.42e-4  2.91e-2  3.09e-2  1.00  94421.
## 4 p[3]       0.874   8.76e-1  2.80e-2  2.78e-2  8.26e-1  9.17e-1  1.00    280.
## 5 ypred[1]    7.69    7 e+0   3.32e+0  2.97e+0  3 e+0   1.4 e+1  1.00    559.
## 6 ypred[2]    2998.   3.00e+3  7.68e+1  7.71e+1  2.87e+3  3.13e+3  1.00  97595.
## 7 ypred[3]    87.4    8.8 e+1  4.34e+0  4.45e+0  8 e+1   9.4 e+1  1.00    669.
## 8 undprev     0.00935 9.28e-3  2.01e-3  2.11e-3  6.17e-3  1.28e-2  1.01    280.
## 9 undprev/beta 0.0300  3.00e-2  5.42e-4  5.42e-4  2.91e-2  3.09e-2  1.00  94421.
## # i 1 more variable: ess_tail <dbl>

summary(outHIVCbiasMat,
~quantile(.x, probs = c(0.025, 0.5, 0.975)))

## # A tibble: 9 x 4
##   variable    `2.5%`   `50%`   `97.5%`
##   <chr>     <dbl>     <dbl>     <dbl>
## 1 beta       0.191     0.309     0.448
## 2 p[1]       0.0441    0.0742    0.122
## 3 p[2]       0.0289    0.0300    0.0311
## 4 p[3]       0.817     0.876     0.925
## 5 ypred[1]    2          7         15
## 6 ypred[2]    2848      2998      3150
## 7 ypred[3]    78         88        95
## 8 undprev    0.00571   0.00928   0.0134
```

```
## 9 undprev/beta    0.0289    0.0300    0.0311
```

- (ii) Reproduce the plots on slide 25 of the lecture, comparing the posterior distributions of each parameter from the model with the bias parameter β to those from the model C1, and the posterior-predictive distributions, calculating also the posterior-predictive p-values. Does the use of the bias parameter lead to a better fit to the data?

- First we join both sets of posterior samples in one tibble:

```
outHIVCvsCbias <- as_tibble(as_draws_matrix(outHIVC1),
                               rownames = "Iter") %>%
  mutate(across(Iter:`ypred[3]`, as.numeric)) %>%
  mutate(beta = as.numeric(NA), Model = "Assume no bias") %>%
  bind_rows(
    as_tibble(as_draws_matrix(outHIVCbias),
              rownames = "Iter") %>%
      mutate(across(Iter:`ypred[3]`, as.numeric)) %>%
      mutate(Model = "Study 2 biased") %>%
      select(Iter, contains("p"), beta, Model)
  ) %>%
  rename(
    pi = `p[1]`, delta = `p[3]`
  ) %>%
  mutate(
    undprev = pi * (1 - delta),
    `undprev/beta` = undprev / beta
  ) %>%
  select(-`p[2]`) %>%
  pivot_longer(
    cols = c(pi, undprev, delta, beta, `undprev/beta`, contains("ypred")),
    names_to = "Parameter",
    values_to = "Posterior"
  )
```

- Next we create a tibble with the observed data in each model, on both the proportion and numerator scales:

```
obsCbias <- tibble(
  Model = rep("Assume no bias", 6),
  Parameter = c("pi", "undprev", "delta", "ypred[1]", "ypred[2]", "ypred[3]"),
  Obs = c(hivDatCbias$y / hivDatCbias$n, hivDatCbias$y)
) %>%
  bind_rows(
    tibble(
      Model = rep("Study 2 biased", 6),
      Parameter = c("pi", "undprev/beta", "delta", "ypred[1]", "ypred[2]", "ypred[3]"),
      Obs = c(hivDatCbias$y / hivDatCbias$n, hivDatCbias$y)
    )
  )
```

- We join the posterior samples with the observed data to calculate the posterior-predictive p-values:

```
pvalsCvsCbias <- outHIVCvsCbias %>%
  left_join(obsCbias, by = c("Model", "Parameter")) %>%
  filter(str_detect(Parameter, "ypred")) %>%
  mutate(
```

```

pval = (Posterior > Obs)
) %>%
group_by(Model, Parameter) %>%
summarise(
Median = median(Posterior, na.rm = TRUE),
Lower = quantile(Posterior, probs = 0.025, na.rm = TRUE),
Upper = quantile(Posterior, probs = 0.975, na.rm = TRUE),
pval = mean(pval, na.rm = TRUE),
.groups = "keep"
) %>%
ungroup()

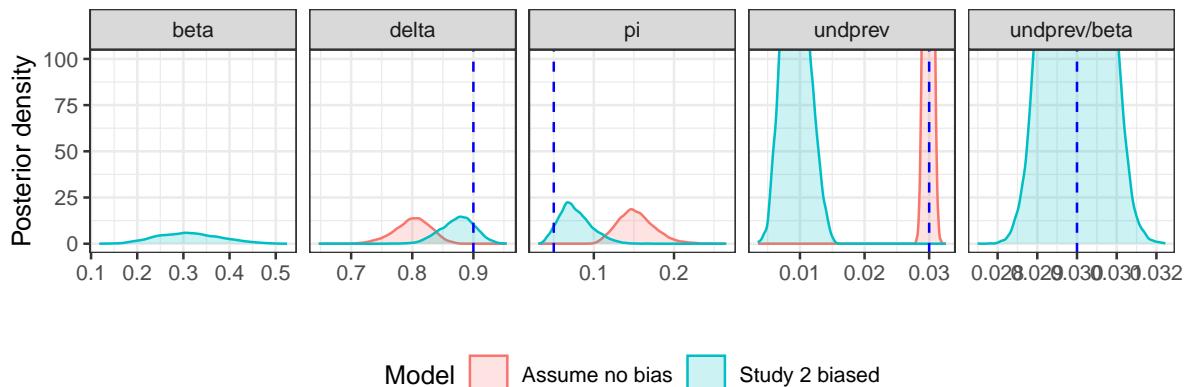
```

- Finally we create the plots, using `geom_vline()` to add the observations to the plots and `geom_text()` to add the posterior-predictive p-values to the plots. We see the better, although not perfect, fit to the observations from Studies 1 and 3 (informing π and δ respectively) in the bias model:

```

# parameter (proportion) scale
outhIVCvsCbias %>%
filter(!str_detect(Parameter, "ypred")) %>%
ggplot() +
aes(x = Posterior, col = Model, fill = Model) +
geom_density(alpha = 0.2) +
facet_grid(. ~ Parameter, scales = "free_x") +
geom_vline(data = obsCbias %>% filter(!str_detect(Parameter, "ypred")), aes(xintercept = Obs), c
labs(x = "", y = "Posterior density") +
coord_cartesian(ylim = c(0,100)) +
theme_bw() +
theme(legend.position = "bottom")

```



```

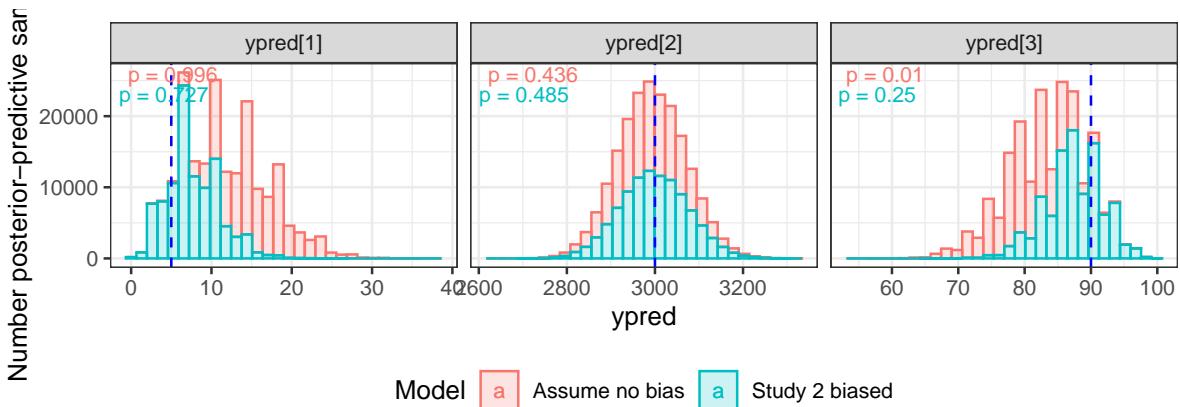
# posterior-predictive replicate scale
outhIVCvsCbias %>%
filter(str_detect(Parameter, "ypred")) %>%
ggplot() +
aes(x = Posterior, col = Model, fill = Model) +
geom_histogram(alpha = 0.2) +
facet_grid(. ~ Parameter, scales = "free_x") +
geom_vline(data = obsCbias %>% filter(str_detect(Parameter, "ypred")), aes(xintercept = Obs), c
geom_text(
data      = pvalsCvsCbias %>% filter(Model == "Assume no bias", !is.na(pval)),
mapping   = aes(x = -Inf, y = Inf, col = Model,

```

```

            label = paste0("p = ", round(pval,3)),
hjust    = -0.2,
vjust    = 1.2,
size = 3
) +
geom_text(
data   = pvalsCvsCbias %>% filter(Model == "Study 2 biased", !is.na(pval)),
mapping = aes(x = -Inf, y = Inf, col = Model,
              label = paste0("p = ", round(pval,3)),
hjust    = -0.1,
vjust    = 2.5,
size = 3
) +
labs(x = "ypred", y = "Number posterior-predictive samples") +
theme_bw() +
theme(legend.position = "bottom")

```



(iii) Compare the bias model with model C1 formally, using the DIC. Which does the DIC prefer?

- We see that despite the extra parameter in the bias model, the posterior mean deviance is much lower for the bias model than the unadjusted model, so that the DIC is also lower for the bias model.

```
dic.samples(hivC1.jm, n.iter = 1000, type = "pD")
```

```
## Mean deviance: 36.92
## penalty 1.894
## Penalized deviance: 38.81
```

```
dic.samples(hivCbias.jm, n.iter = 1000, type = "pD")
```

```
## Mean deviance: 20.64
## penalty 1.412
## Penalized deviance: 22.05
```

(B) This set of exercises considers the HIV prevalence model on slides 27-29 and 32-33 of the lecture notes, incorporating a count observation $y_4 = 400$ informing the number of people living with diagnosed HIV in the fixed population of size $N = 10,000$.

- Implement the model assuming a Poisson sampling distribution for the count y_4 . Compare the estimates visually and in terms of posterior-predictive p-values to model C1, the model with only

three data points and no bias modelling (i.e. reproduce the plots on slides 28-29). Which parameter estimates are most affected by the addition of Study 4?

- Model code, data and initial values:

```

hivModelD <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi,b.pi)
  delta ~ dbeta(a.delta,b.delta)

  # Likelihoods
  # prevalence, undiagnosed prevalence and proportion diagnosed data
  for(i in 1:3)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }
  # total number diagnosed
  y[4] ~ dpois(p[4])
  ypred[4] ~ dpois(p[4])

  # Proportions in terms of basic and functional parameters
  p[1] <- pi
  p[2] <- pi * (1 - delta)
  p[3] <- delta
  p[4] <- delta * pi * N
}
"

# DATA
hivDatD <- list(
  y = c( 5, 3000, 90, 400),
  n = c(100, 100000, 100),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1,
  N = 10000
)

# INITS
hivInitsD <- list(
  # chain 1
  list(
    pi = 0.1,
    delta = 0.9,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(7195)
  ),
  # chain 2
  list(
    pi = 0.2,
    delta = 0.2,
    .RNG.name = c("base::Mersenne-Twister"),

```

```

.RNG.seed = c(168422)
)
)

```

- Obtain the posterior samples:

```

# Initialise model
hivD1.jm <- jags.model(textConnection(hivModelD),
                        data = hivDatD,
                        inits = hivInitsD,
                        n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 4
## Unobserved stochastic nodes: 6
## Total graph size: 22
##
## Initializing model

# burn-in
update(hivD1.jm, n.iter = nBurn)

# samples to keep
hivParamsD <- c("pi", "delta", "ypred")
outHIVD1 <- coda.samples(hivD1.jm,
                         variable.names = hivParamsD,
                         n.iter = nIter*10,
                         n.thin = 10)

```

- Parameter estimates:

```

outHIVD1mat <- as_draws_matrix(outHIVD1) %>%
  mutate_variables(
    undprev = pi * (1 - delta),
    nDiag = delta * pi * hivDatD$N
  )
summary(outHIVD1mat)

## # A tibble: 8 x 10
##   variable     mean    median      sd      mad      q5     q95 rhat ess_bulk
##   <chr>      <dbl>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>
## 1 delta      0.591    0.591  0.0120  1.21e-2 5.71e-1 6.11e-1  1.00  13131.
## 2 pi         0.0727   0.0726  0.00209 2.08e-3 6.93e-2 7.62e-2  1.00  13383.
## 3 ypred[1]   7.27     7       2.61    2.97e+0 3e+0  1.2e+1  1.00  94587.
## 4 ypred[2]  2970.    2969    75.9   7.56e+1 2.85e+3 3.10e+3  1.00  94967.
## 5 ypred[3]   59.1     59      5.04   4.45e+0 5.1e+1 6.7e+1  1.00  75934.
## 6 ypred[4]   430.     429     28.9   2.82e+1 3.83e+2 4.78e+2  1.00  22368.
## 7 undprev    0.0297   0.0297  0.000536 5.39e-4 2.88e-2 3.06e-2  1.00  91436.
## 8 nDiag     430.     429.    20.1    2.01e+1 3.97e+2 4.63e+2  1.00  12179.
## # i 1 more variable: ess_tail <dbl>

```

```

summary(outhIVD1mat,
~quantile(.x, probs = c(0.025,0.5,0.975)))

## # A tibble: 8 x 4
##   variable    `2.5%`    `50%`    `97.5%
##   <chr>      <dbl>     <dbl>     <dbl>
## 1 delta      0.567     0.591     0.614
## 2 pi         0.0687    0.0726    0.0769
## 3 ypred[1]    3         7         13
## 4 ypred[2]  2822      2969      3119
## 5 ypred[3]    49        59        69
## 6 ypred[4]   374       429       488
## 7 undprev    0.0287    0.0297    0.0308
## 8 nDiag      391.      429.      470.

```

- Combine posterior samples in a tibble with those from model C1:

```

outhIVDvsC <- as_tibble(as_draws_matrix(outhIVD1),
                           rownames = "Iter") %>%
  mutate(across(Iter:`ypred[4]`, as.numeric)) %>%
  mutate(Model = "Studies 1-3 + 4: number diagnosed") %>%
  full_join(
  as_tibble(as_draws_matrix(outhIVC1),
            rownames = "Iter") %>%
  mutate(across(Iter:`ypred[3]`, as.numeric)) %>%
  rename(pi = `p[1]`, delta = `p[3]`) %>%
  mutate(Model = "Studies 1-3") %>%
  select(Iter, Model, pi, delta, contains("ypred")),
  by = c("Iter", "Model", "pi", "delta", "ypred[1]", "ypred[2]", "ypred[3]"))
) %>%
  mutate(
  undprev = pi * (1-delta),
  nDiag = delta * pi * hivDatD$N
) %>%
  pivot_longer(
  cols = c(pi, undprev, delta, nDiag, contains("ypred")),
  names_to = "Parameter",
  values_to = "Posterior"
) %>%
  mutate(Parameter = factor(Parameter,
                            levels = c("pi", "delta", "undprev", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]"))

```

- Observations in a tibble, to join with posterior samples to calculate posterior-p-values:

```

obsD <- tibble(
  Model = rep("Studies 1-3 + 4: number diagnosed", 8),
  Parameter = c("pi", "undprev", "delta", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"),
  Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[4], hivDatD$y[1:4])
) %>%
  bind_rows(
  tibble(
    Model = rep("Studies 1-3", 6),
    Parameter = c("pi", "undprev", "delta", "ypred[1]", "ypred[2]", "ypred[3]"),
    Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[1:3])
)

```

```

    )
) %>%
mutate(Parameter = factor(Parameter,
                           levels = c("pi", "delta", "undprev", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]")),
       Posterior = ifelse(Posterior < 0, 0, Posterior),
       Obs = ifelse(Obs < 0, 0, Obs))
pvalsDvsC <- outHIVDvsC %>%
left_join(obsD, by = c("Model", "Parameter")) %>%
filter(str_detect(Parameter, "ypred")) %>%
mutate(
pval = (Posterior > Obs)
) %>%
group_by(Model, Parameter) %>%
summarise(
Median = median(Posterior, na.rm = T),
Lower = quantile(Posterior, probs = 0.025, na.rm = T),
Upper = quantile(Posterior, probs = 0.975, na.rm = T),
pval = mean(pval, na.rm = T),
.groups = "keep"
) %>%
ungroup()

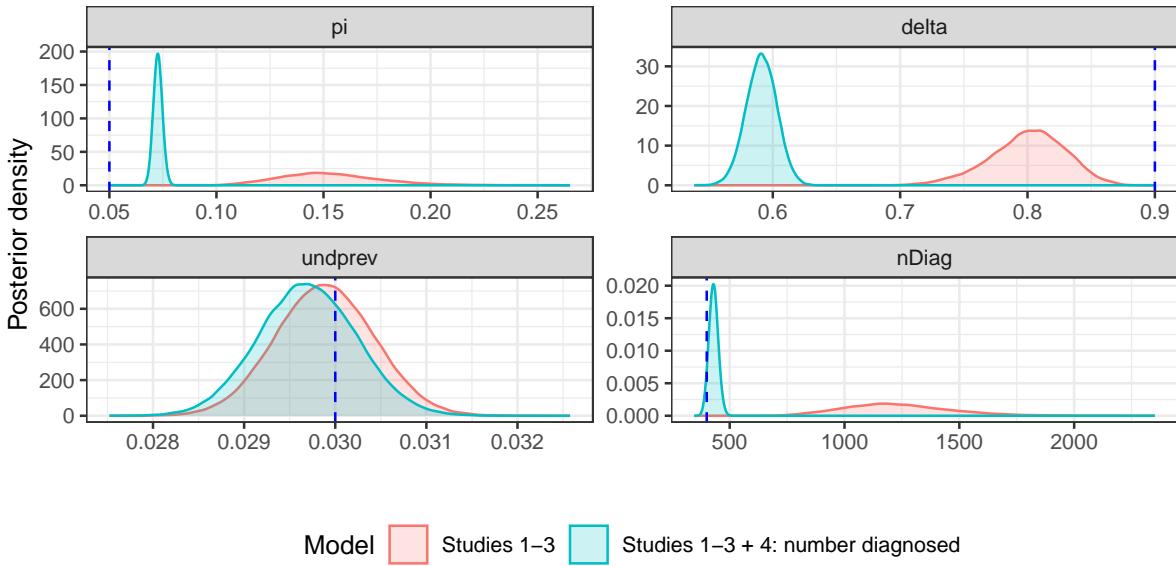
```

- Plotting this Poisson model (D) against model C1, we find that the addition of the data y_4 on the number diagnosed shifts the estimate of δ even further away from its observed value, but also shifts the estimate of π closer to its observed value. However, both sets of posterior-predictive p-values and the plots demonstrate there is still substantial conflict, as expected, since we have done nothing yet to try to resolve the conflict:

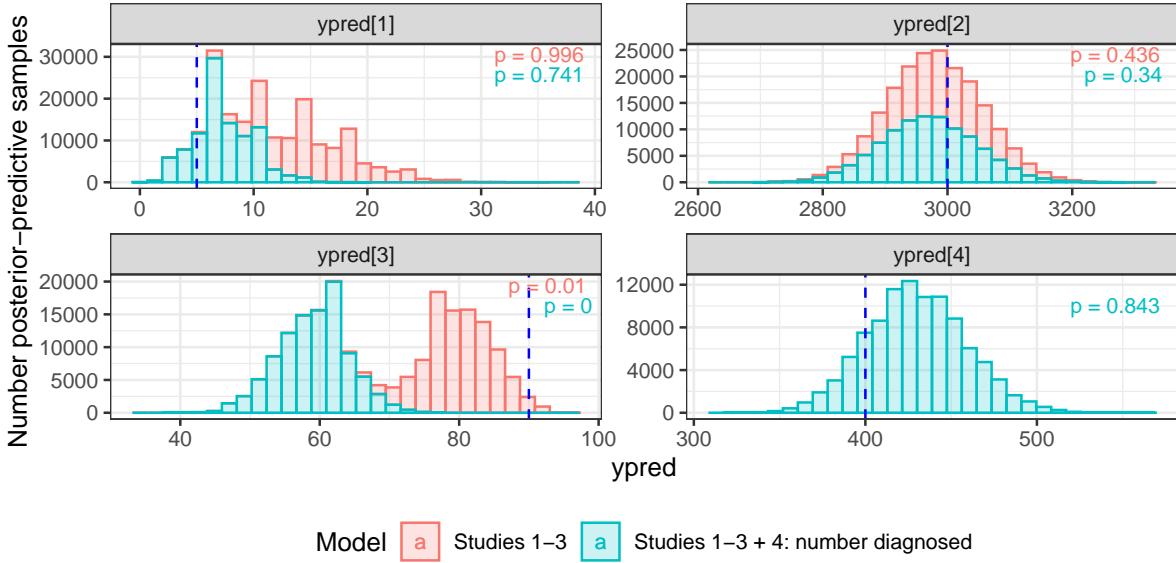
```

outHIVDvsC %>%
filter(!str_detect(Parameter, "ypred")) %>%
ggplot() +
aes(x = Posterior, col = Model, fill = Model) +
geom_density(alpha = 0.2) +
facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
geom_vline(data = obsD %>% filter(!str_detect(Parameter, "ypred")), aes(xintercept = Obs), col =
labs(x = "", y = "Posterior density") +
theme_bw() +
theme(legend.position = "bottom")

```



```
outhIVDvsC %>%
  filter(str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = Posterior, col = Model, fill = Model) +
  geom_histogram(alpha = 0.2) +
  facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
  geom_vline(data = obsD %>% filter(str_detect(Parameter, "ypred")), aes(xintercept = Obs), col = "black",
             size = 1)
  geom_text(
    data = pvalsDvsC %>% filter(Model == "Studies 1-3",
                                   !is.na(pval)),
    mapping = aes(x = Inf, y = Inf, col = Model,
                  label = paste0("p = ", round(pval,3))),
    hjust = 1.2,
    vjust = 1.2,
    size = 3
  ) +
  geom_text(
    data = pvalsDvsC %>% filter(Model == "Studies 1-3 + 4: number diagnosed",
                                   !is.na(pval)),
    mapping = aes(x = Inf, y = Inf, col = Model,
                  label = paste0("p = ", round(pval,3))),
    hjust = 1.2,
    vjust = 2.5,
    size = 3
  ) +
  labs(x = "ypred", y = "Number posterior-predictive samples") +
  theme_bw() +
  theme(legend.position = "bottom")
```



- (ii) Implement the model assuming instead a Negative Binomial $\text{NB}(\psi, r)$ sampling distribution, with a Beta prior for ψ expressing that it lies between 0.2 and 0.6 (i.e. with a prior mean around 0.4 and standard deviation around 0.1). Reproduce the plots on slides 32-33, calculating and plotting in addition the posterior p-values, to compare the Negative binomial model to the Poisson model. Compare them also using the DIC.

- Model code, data and initial values:

```

hivModelD2 <- "
model
{
  # Flat priors, i.e. each a, b = 1 (Unif(0,1))
  # Set prior parameters in data list
  pi ~ dbeta(a.pi,b.pi)
  delta ~ dbeta(a.delta,b.delta)

  # Likelihoods
  # prevalence, undiagnosed prevalence and proportion diagnosed data
  for(i in 1:3)
  {
    y[i] ~ dbin(p[i], n[i])
    ypred[i] ~ dbin(p[i], n[i])
  }
  # total number diagnosed
  y[4] ~ dnegbin(psi, r)
  ypred[4] ~ dnegbin(psi, r)

  # Proportions in terms of basic and functional parameters
  p[1] <- pi
  p[2] <- pi * (1 - delta)
  p[3] <- delta

  # Beta prior for negative binomial parameter psi giving values
  # roughly between 0.2 and 0.6, representing variances between 1.67 and
  # 5 times the mean
  r <- psi * nDiag / (1 - psi)
}

```

```

    nDiag <- delta * pi * N
    psi ~ dbeta(a.psi, b.psi)
}
"

# DATA
hivDatD2 <- list(
  y = c( 5, 3000, 90, 400),
  n = c(100, 100000, 100),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1,
  a.psi = 9.2, b.psi = 13.8,
  N = 10000
)

# INITS
hivInitsD2 <- list(
  # chain 1
  list(
    pi = 0.1,
    delta = 0.9,
    psi = 0.4,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(7195)
  ),
  # chain 2
  list(
    pi = 0.2,
    delta = 0.2,
    psi = 0.8,
    .RNG.name = c("base::Mersenne-Twister"),
    .RNG.seed = c(168422)
  )
)

```

- Obtain the posterior samples:

```

# Initialise model
hivD2.jm <- jags.model(textConnection(hivModelD2),
                        data = hivDatD2,
                        inits = hivInitsD2,
                        n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 4
##   Unobserved stochastic nodes: 7
##   Total graph size: 28
##
## Initializing model

```

```

# burn-in
update(hivD2.jm, n.ITER = nBurn)

# samples to keep
hivParamsD2 <- c("pi", "delta", "psi", "ypred")
outHIVD2 <- coda.samples(hivD2.jm,
                         variable.names = hivParamsD2,
                         n.ITER = nIter*10,
                         n.thin = 10)

```

- Parameter estimates:

```

outHIVD2mat <- as_draws_matrix(outHIVD2) %>%
  mutate_variables(
    undprev = pi * (1 - delta),
    nDiag = delta * pi * hivDatD2$N,
    r = psi * nDiag / (1 - psi)
  )
summary(outHIVD2mat)

## # A tibble: 10 x 10
##   variable     mean    median      sd      mad      q5      q95    rhat ess_bulk
##   <chr>     <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 delta      0.618    0.617  2.07e-2 1.99e-2 5.86e-1 6.54e-1  1.00    3284.
## 2 pi         0.0781   0.0776  4.37e-3 4.01e-3 7.18e-2 8.58e-2  1.00    3294.
## 3 psi        0.335    0.330  1.08e-1 1.11e-1 1.66e-1 5.20e-1  1.00    8039.
## 4 ypred[1]    7.80     8       2.72e+0 2.97e+0 4e+0 1.2e+1  1.00    59827.
## 5 ypred[2]   2972.    2971   7.61e+1 7.56e+1 2.85e+3 3.10e+3  1.00    95635.
## 6 ypred[3]    61.8     62      5.25e+0 5.93e+0 5.3e+1 7e+1  1.00    17401.
## 7 ypred[4]    484.     478    5.96e+1 5.49e+1 3.97e+2 5.87e+2  1.00    6011.
## 8 undprev    0.0297   0.0297  5.37e-4 5.38e-4 2.88e-2 3.06e-2  1.00    90615.
## 9 nDiag      484.     479.   4.32e+1 3.95e+1 4.22e+2 5.60e+2  1.00    3208.
## 10 r         259.     236.   1.26e+2 1.08e+2 1.04e+2 4.95e+2  1.00   11772.
## # i 1 more variable: ess_tail <dbl>

summary(outHIVD2mat,
~quantile(.x, probs = c(0.025, 0.5, 0.975)))

## # A tibble: 10 x 4
##   variable `2.5%`  `50%`  `97.5%`
##   <chr>     <dbl>    <dbl>    <dbl>
## 1 delta      0.580    0.617    0.662
## 2 pi         0.0708   0.0776   0.0878
## 3 psi        0.141    0.330    0.557
## 4 ypred[1]    3       8       14
## 5 ypred[2]   2824    2971    3122
## 6 ypred[3]    51      62      72
## 7 ypred[4]   383     478     615
## 8 undprev    0.0287   0.0297   0.0308
## 9 nDiag      412.    479.    580.
## 10 r         88.1    236.    571.
```

- Combine posterior samples in a tibble with those from model D1:

```

outhIVD2vsD1 <- as_tibble(as_draws_matrix(outhIVD2mat),
                           rownames = "Iter") %>%
  mutate(across(Iter:r, as.numeric)) %>%
  mutate(Model = "Negative binomial") %>%
  full_join(
    as_tibble(as_draws_matrix(outhIVD1),
              rownames = "Iter") %>%
    mutate(across(Iter:`ypred[4]`, as.numeric)) %>%
    mutate(Model = "Poisson") %>%
    select(Iter, Model, pi, delta, contains("ypred")),
    by = c("Iter", "Model", "pi", "delta", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]")
  ) %>%
  mutate(
    undprev = pi * (1 - delta),
    nDiag = delta * pi * hivDatD$N,
    Model = factor(Model, levels = c("Poisson", "Negative binomial"))
  ) %>%
  pivot_longer(
    cols = c(pi, delta, undprev, nDiag, psi, r, contains("ypred")),
    names_to = "Parameter",
    values_to = "Posterior"
  ) %>%
  mutate(
    Parameter = factor(Parameter,
                        levels = c("pi", "delta", "undprev", "nDiag", "psi", "r", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"))
  )

```

- Observations in a tibble, to join with posterior samples to calculate posterior-predictive p-values:

```

obsD2 <- tibble(
  Model = rep("Poisson", 8),
  Parameter = c("pi", "undprev", "delta", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"),
  Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[4], hivDatD$y[1:4])
) %>%
  bind_rows(
    tibble(
      Model = rep("Negative binomial", 8),
      Parameter = c("pi", "undprev", "delta", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"),
      Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[4], hivDatD$y[1:4])
    )
  ) %>%
  mutate(Parameter = factor(Parameter,
                            levels = c("pi", "delta", "undprev", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]")))

```

`pvalsD2vsD1 <- outhIVD2vsD1 %>%`

```

  filter(str_detect(Parameter, "ypred")) %>%
  left_join(obsD2 %>% filter(str_detect(Parameter, "ypred")) , by = c("Model", "Parameter")) %>%
  mutate(
    pval = (Posterior > Obs)
  ) %>%
  group_by(Model, Parameter) %>%
  summarise(
    Median = median(Posterior, na.rm = TRUE),
    Lower = quantile(Posterior, probs = 0.025, na.rm = TRUE),
    Upper = quantile(Posterior, probs = 0.975, na.rm = TRUE)
  )

```

```

Upper = quantile(Posterior, probs = 0.975, na.rm = TRUE),
pval = mean(pval),
.groups = "keep"
) %>%
ungroup()

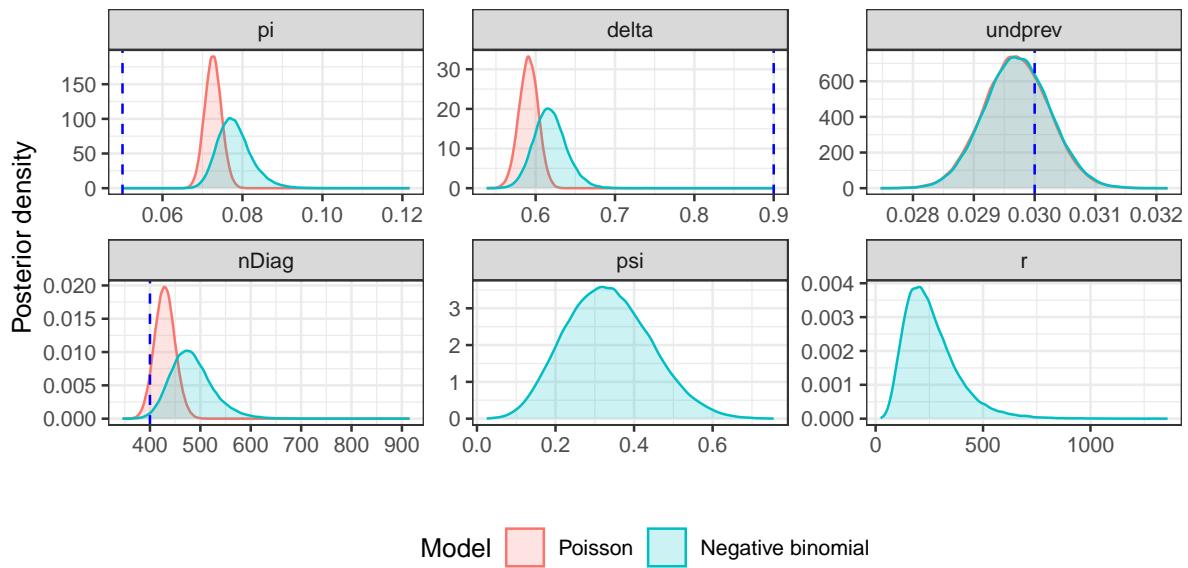
```

- Plotting this Negative Binomial model (D2) against model D1, we find that allowing for over-dispersion with a Beta(9.2, 13.8) prior for ψ does little to alleviate the large amount of conflict in this example, other than slightly widening the posterior uncertainty for the π and δ parameters and shifting them slightly higher.

```

outHIVD2vsD1 %>%
  filter(!str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = Posterior, col = Model, fill = Model) +
  geom_density(alpha = 0.2) +
  facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
  geom_vline(data = obsD2 %>% filter(!str_detect(Parameter, "ypred")), aes(xintercept = Obs), col =
  labs(x = "", y = "Posterior density") +
  theme_bw() +
  theme(legend.position = "bottom")

```



```

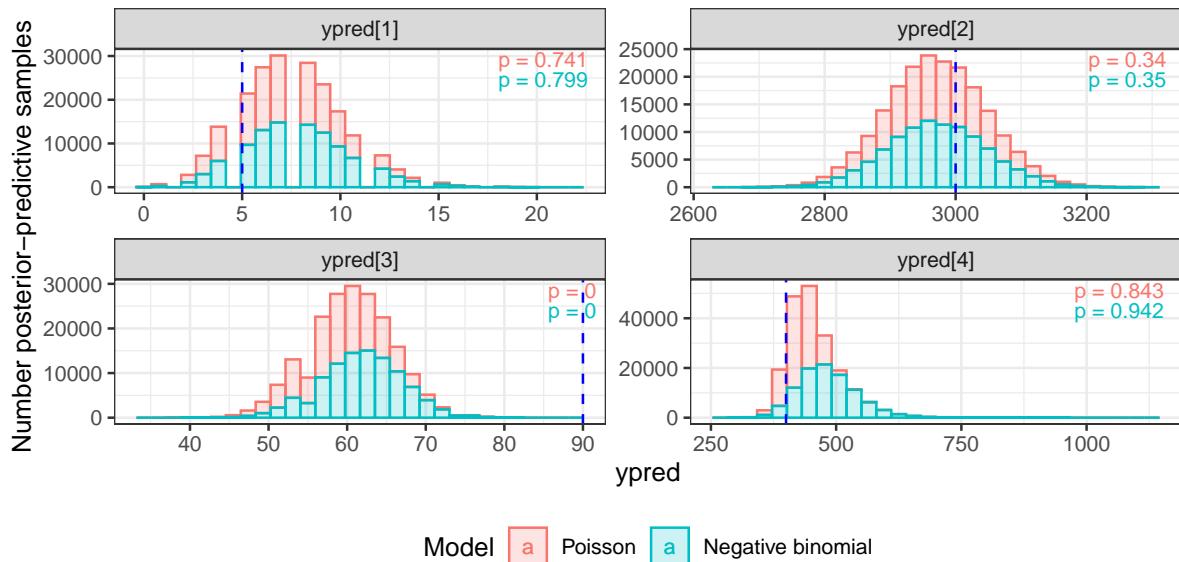
outHIVD2vsD1 %>%
  filter(str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = Posterior, col = Model, fill = Model) +
  geom_histogram(alpha = 0.2) +
  facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
  geom_vline(data = obsD2 %>% filter(str_detect(Parameter, "ypred")), aes(xintercept = Obs), col =
  geom_text(
data      = pvalsD2vsD1 %>% filter(Model == "Poisson",
                                         !is.na(pval)),
mapping   = aes(x = Inf, y = Inf, col = Model,
                 label = paste0("p = ", round(pval,3))),
hjust     = 1.2,

```

```

vjust    = 1.2,
size = 3
) +
  geom_text(
data    = pvalsD2vsD1 %>% filter(Model == "Negative binomial",
                                   !is.na(pval)),
mapping = aes(x = Inf, y = Inf, col = Model,
              label = paste0("p = ", round(pval,3))),
hjust    = 1.2,
vjust    = 2.5,
size = 3
) +
  labs(x = "ypred", y = "Number posterior-predictive samples") +
  theme_bw() +
  theme(legend.position = "bottom")

```



- Comparing the Negative Binomial model to the Poisson model using the DIC, we find that the negative binomial is marginally preferred. However, the fact the negative binomial is preferred does not mean that either model is adequate! We clearly see from the plots above and the posterior-predictive p-values that neither model fits the data particularly well.

```
dic.samples(hivD1.jm, n.iter = 1000, type = "pD")
```

```

## Mean deviance: 77.73
## penalty 1.871
## Penalized deviance: 79.6

dic.samples(hivD2.jm, n.iter = 1000, type = "pD")

## Mean deviance: 75
## penalty 2.122
## Penalized deviance: 77.13

```

- (iii) What happens if you change the prior for ψ so that you allow for much greater over-dispersion, with much more certainty (e.g. prior mean 0.1, prior sd 0.01)? Does the flexibility of the Negative binomial model do anything to counteract the large conflict, given how large the sample size n_2 is?

- Changing the data list, and re-running:

```
# DATA
hivDatD3 <- list(
  y = c( 5, 3000, 90, 400),
  n = c(100, 100000, 100),
  a.pi = 1, b.pi = 1,
  a.delta = 1, b.delta = 1,
  a.psi = 89.9, b.psi = 809.1,
  N = 10000
)

# Initialise model
hivD3.jm <- jags.model(textConnection(hivModelD2),
                        data = hivDatD3,
                        inits = hivInitsD2,
                        n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 4
##   Unobserved stochastic nodes: 7
##   Total graph size: 28
##
## Initializing model
# burn-in
update(hivD3.jm, n.iter = nBurn)

# samples to keep
outHIVD3 <- coda.samples(hivD3.jm,
                           variable.names = hivParamsD2,
                           n.iter = nIter*10,
                           n.thin = 10)
```

- Parameter estimates:

```
outHIVD3mat <- as_draws_matrix(outHIVD3) %>%
  mutate_variables(
    undprev = pi * (1 - delta),
    nDiag = delta * pi * hivDatD2$N,
    r = psi * nDiag / (1 - psi)
  )
summary(outHIVD3mat)

## # A tibble: 10 x 10
##   variable     mean    median      sd     mad     q5     q95   rhat ess_bulk
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl> <dbl>    <dbl>
## 1 delta      0.662    0.663   0.0226   2.23e-2 6.24e-1 6.98e-1  1.00    2185.
## 2 pi         0.0885   0.0883  0.00593  5.80e-3 7.91e-2 9.86e-2  1.00    2172.
## 3 psi        0.0966   0.0963  0.00996  9.90e-3 8.08e-2 1.14e-1  1.00    29778.
## 4 ypred[1]    8.84     9       2.90     2.97e+0 4e+0 1.4e+1  1.00    38573.
## 5 ypred[2]  2976     2976    76.2     7.56e+1 2.85e+3 3.10e+3  1.00    94094.
```

```

## 6 ypred[3]    66.2      66      5.24      5.93e+0 5.8 e+1 7.5 e+1 1.00 11764.
## 7 ypred[4]    587.     582     98.2      9.64e+1 4.36e+2 7.57e+2 1.00 5966.
## 8 undprev     0.0298   0.0298  0.000537 5.36e-4 2.89e-2 3.06e-2 1.00 90590.
## 9 nDiag       587.     585.    58.8      5.76e+1 4.94e+2 6.88e+2 1.00 2141.
## 10 r          62.8     62.4     8.49      8.44e+0 4.94e+1 7.74e+1 1.00 6603.
## # i 1 more variable: ess_tail <dbl>
summary(outHIVD3mat,
~quantile(.x, probs = c(0.025,0.5,0.975)))

```

```

## # A tibble: 10 x 4
##   variable `2.5%`  `50%`  `97.5%`
##   <chr>     <dbl>    <dbl>    <dbl>
## 1 delta      0.616    0.663    0.705
## 2 pi         0.0775   0.0883   0.101
## 3 psi        0.0780   0.0963   0.117
## 4 ypred[1]    4        9        15
## 5 ypred[2]   2829    2976    3127
## 6 ypred[3]    56       66       76
## 7 ypred[4]   410      582     795
## 8 undprev    0.0287   0.0298   0.0308
## 9 nDiag      478.     585.    710.
## 10 r         47.2     62.4     80.5

```

- Combine posterior samples in a tibble with those from model D2:

```

outHIVD3vsD2 <- as_tibble(outHIVD3mat,
                           rownames = "Iter") %>%
  mutate(across(Iter:r, as.numeric)) %>%
  mutate(Model = "NB more over-dispersion") %>%
  full_join(
    as_tibble(outHIVD2mat,
              rownames = "Iter") %>%
    mutate(across(Iter:r, as.numeric)) %>%
    mutate(Model = "NB less over-dispersion"),
    by = c("Iter", "Model", "pi", "delta", "undprev", "nDiag", "psi", "r", "ypred[1]", "ypred[2]", "ypred[3]"))
  ) %>%
  mutate(
    undprev = pi * (1 - delta),
    nDiag = delta * pi * hivDatD$N,
    Model = factor(Model, levels = c("NB less over-dispersion", "NB more over-dispersion"))
  ) %>%
  pivot_longer(
    cols = c(pi, delta, undprev, nDiag, psi, r, contains("ypred")),
    names_to = "Parameter",
    values_to = "Posterior"
  ) %>%
  mutate(
    Parameter = factor(Parameter,
                        levels = c("pi", "delta", "undprev", "nDiag", "psi", "r", "ypred[1]", "ypred[2]", "ypred[3]"))
  )

```

- Observations in a tibble, to join with posterior samples to calculate posterior-predictive p-values:

```

obsD3 <- tibble(
  Model = rep("NB less over-dispersion", 8),
  Parameter = c("pi", "undprev", "delta", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"),
  Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[4], hivDatD$y[1:4])
) %>%
  bind_rows(
    tibble(
      Model = rep("NB more over-dispersion", 8),
      Parameter = c("pi", "undprev", "delta", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]"),
      Obs = c(hivDatD$y[1:3] / hivDatD$n[1:3], hivDatD$y[4], hivDatD$y[1:4])
    )
  ) %>%
  mutate(Parameter = factor(Parameter,
                            levels = c("pi", "delta", "undprev", "nDiag", "ypred[1]", "ypred[2]", "ypred[3]", "ypred[4]")))
)

pvalsD3vsD2 <- outHIVD3vsD2 %>%
  filter(str_detect(Parameter, "ypred")) %>%
  left_join(obsD3 %>% filter(str_detect(Parameter, "ypred"))), by = c("Model", "Parameter")) %>%
  mutate(
    pval = (Posterior > Obs)
  ) %>%
  group_by(Model, Parameter) %>%
  summarise(
    Median = median(Posterior, na.rm = TRUE),
    Lower = quantile(Posterior, probs = 0.025, na.rm = TRUE),
    Upper = quantile(Posterior, probs = 0.975, na.rm = TRUE),
    pval = mean(pval),
    .groups = "keep"
  ) %>%
  ungroup()

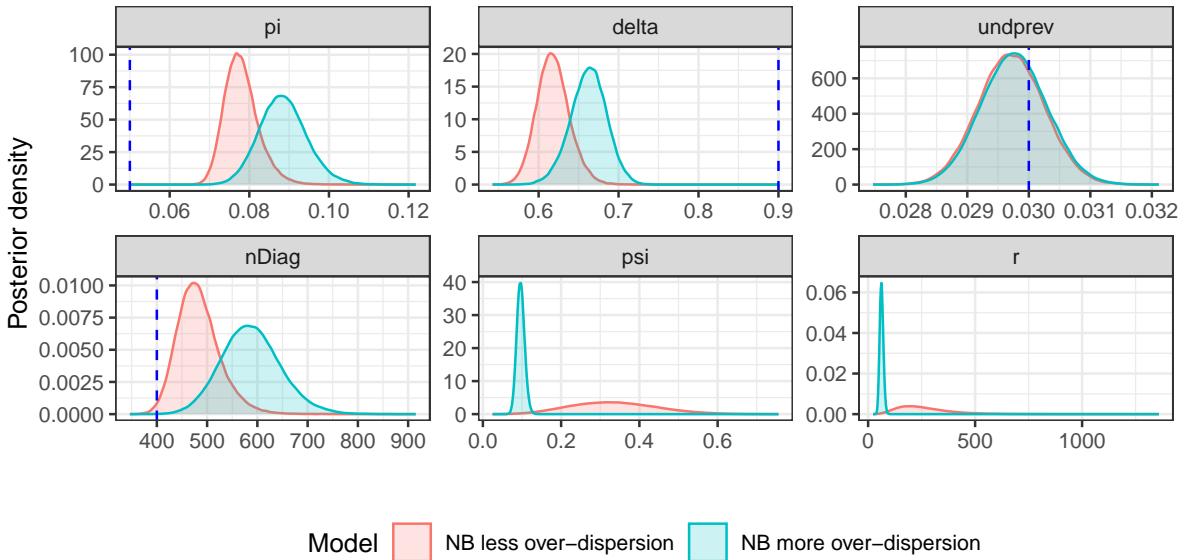
```

- Plotting this negative binomial model with more over-dispersion (D3) against the previous one with less over-dispersion (D2), we find that although the posterior uncertainty for the number diagnosed and hence for π and δ is increased, the change in prior over-dispersion actually worsens the fit to the data on π . However, it improves slightly the fit to the data on δ :

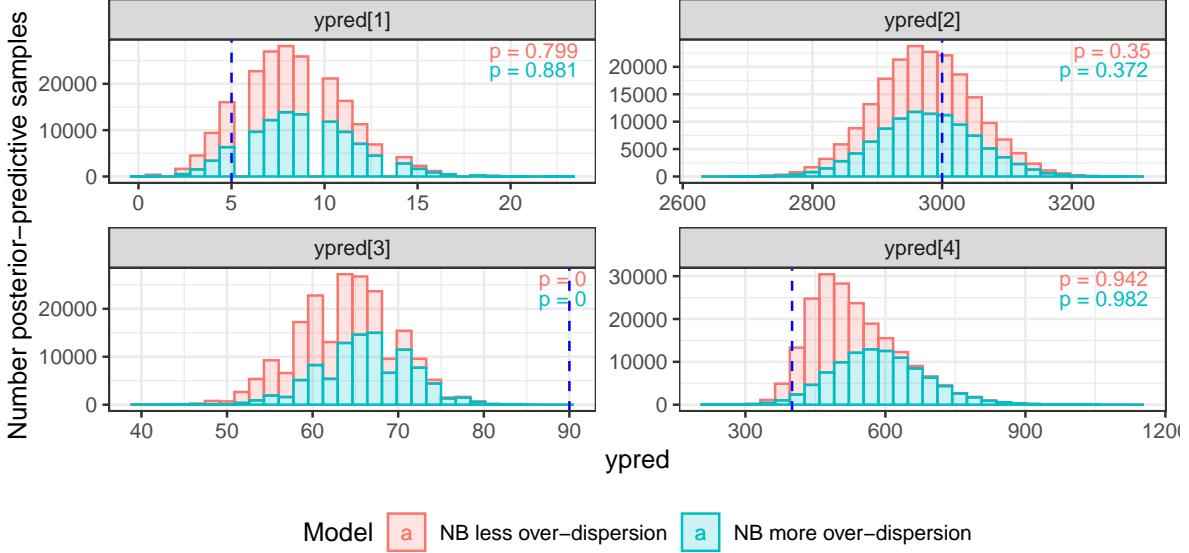
```

outHIVD3vsD2 %>%
  filter(!str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = Posterior, col = Model, fill = Model) +
  geom_density(alpha = 0.2) +
  facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
  geom_vline(data = obsD3 %>% filter(!str_detect(Parameter, "ypred"))), aes(xintercept = Obs), col =
  labs(x = "", y = "Posterior density") +
  theme_bw() +
  theme(legend.position = "bottom")

```



```
outhIVD3vsD2 %>%
  filter(str_detect(Parameter, "ypred")) %>%
  ggplot() +
  aes(x = Posterior, col = Model, fill = Model) +
  geom_histogram(alpha = 0.2) +
  facet_wrap(vars(Parameter), nrow = 2, scales = "free") +
  geom_vline(data = obsD3 %>% filter(str_detect(Parameter, "ypred")), aes(xintercept = Obs), col =
  geom_text(
    data      = pvalsD3vsD2 %>% filter(Model == "NB less over-dispersion",
                                           !is.na(pval)),
    mapping   = aes(x = Inf, y = Inf, col = Model,
                    label = paste0("p = ", round(pval,3))),
    hjust     = 1.2,
    vjust     = 1.2,
    size      = 3
  ) +
  geom_text(
    data      = pvalsD3vsD2 %>% filter(Model == "NB more over-dispersion",
                                           !is.na(pval)),
    mapping   = aes(x = Inf, y = Inf, col = Model,
                    label = paste0("p = ", round(pval,3))),
    hjust     = 1.2,
    vjust     = 2.5,
    size      = 3
  ) +
  labs(x = "ypred", y = "Number posterior-predictive samples") +
  theme_bw() +
  theme(legend.position = "bottom")
```



- The DIC nevertheless prefers slightly the model with more over-dispersion. However, again the fact the greater over-dispersion is preferred does not mean that either model is adequate! The absolute fit seen in both models is still poor:

```
dic.samples(hivD2.jm, n.iter = 1000, type = "pD")
```

```
## Mean deviance: 75.18
## penalty 2.469
## Penalized deviance: 77.65

dic.samples(hivD3.jm, n.iter = 1000, type = "pD")

## Mean deviance: 69.28
## penalty 1.784
## Penalized deviance: 71.06
```

4.3 Systematic model criticism and bias adjustment

(A) Following the outline in slides 35-38 of the lecture notes, implement the cross-validatory mixed-predictive checks, following the steps below:

- Complete the code outline provided here. Replace the question marks with the appropriate expressions to obtain the replicate data $y_{i,k}^{rep}$ from the replicate parameters $p_{i,k}^{rep}$, μ_i^{rep} and δ_i^{rep} :
- The replicated observations for the left-out study i are given by $y_{i,k}^{rep} \sim \text{Binomial}(n_{i,k}, p_{i,k}^{rep})$ and the corresponding replicated proportions are given by $\text{logit}(p_{i,1}^{rep}) = \mu_i^{rep}$ and $\text{logit}(p_{i,2}^{rep}) = \mu_i^{rep} + \delta_i^{rep}$:

```
sepsis_cv_model <- "
model
{
  # Cross-validation: repeat data set/analysis, leaving i'th
  # study out each time
  for(i in 1:Ns)
  {
    # For each study, Ns = total number of studies
    for(j in 1:Ns)
    {
```

```

# for each of the two arms, control=1, treatment=2
for(k in 1:Na)
{
  # Binomial likelihood
  # We don't leave the i'th study out here, but we
  # later do not link p[i,i,k] to any other parameters,
  # so that effectively the i'th study is left out
  ycv[i,j,k] ~ dbin(p[i,j,k], ncv[i,j,k])
}

# On a logit scale, the proportion p is the probability
# of infection in terms of the study baselines mu and
# the study-specific treatment effects delta (log odds ratios,
# relative to the study baseline), if not left-out.
# i=j refers to the i=j'th study being left out: (1 - equals(i,j))
# is equal to 0 if i=j, 1 otherwise; so that p is linked to the
# treatment effects only if the study is not left out.
# Since p[i,i,k] is not linked to either mu[i,i] or delta[i,i],
# mu[i,i] and delta[i,i] are the Replicate parameters,
# drawn from their (random effects) distributions with
# no data directly informing them, only the indirect information
# from all studies other than i
logit(p[i,j,1]) <- ((1 - equals(i,j)) * mu[i,j])
logit(p[i,j,2]) <- ((1 - equals(i,j)) * (mu[i,j] + delta[i,j]))

# for the treatment arm, the log odds ratios are random
# effects with mean d[i] for the i'th cross-validation
delta[i,j] ~ dnorm(d[i], prec.d[i])

# study-specific baselines, vague priors
mu[i,j] ~ dnorm(0, 0.01)
}

# Replications
# for each arm
for(k in 1:Na)
{
  # replicated data for left-out i'th study
  yrep[i,k] ~ dbin(prep[i,k], ncv[i,i,k])
}
# replicate proportions infected
logit(prep[i,1]) <- mu[i,i]
logit(prep[i,2]) <- mu[i,i] + delta[i,i]

# Priors for basic parameters (mean and sd log odds ratio of
# treatment vs control)
d[i] ~ dnorm(0, 0.01)
prec.d[i] <- 1 / (sd.d[i] * sd.d[i])
sd.d[i] ~ dunif(0,10)
}
}
"

```

- (ii) The repeated data (ycv, ncv) have already been provided in a data list (`sepsis_cv_dat`) in

`sepsis.R`, as have a set of initial values for two chains. Run the cross-validation model and carry out your usual checks of convergence: have you run the chains for long enough?

- Running the cross-validation with a burn-in of 1,000 samples followed by 10,000 iterations thinned to every 2 to keep:

```
# Initialise model
sepsis_cv_jm <- jags.model(textConnection(sepsis_cv_model),
                             data = sepsis_cv_dat,
                             inits = sepsis_cv_inits,
                             n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 200
##   Unobserved stochastic nodes: 240
##   Total graph size: 1296
##
## Initializing model

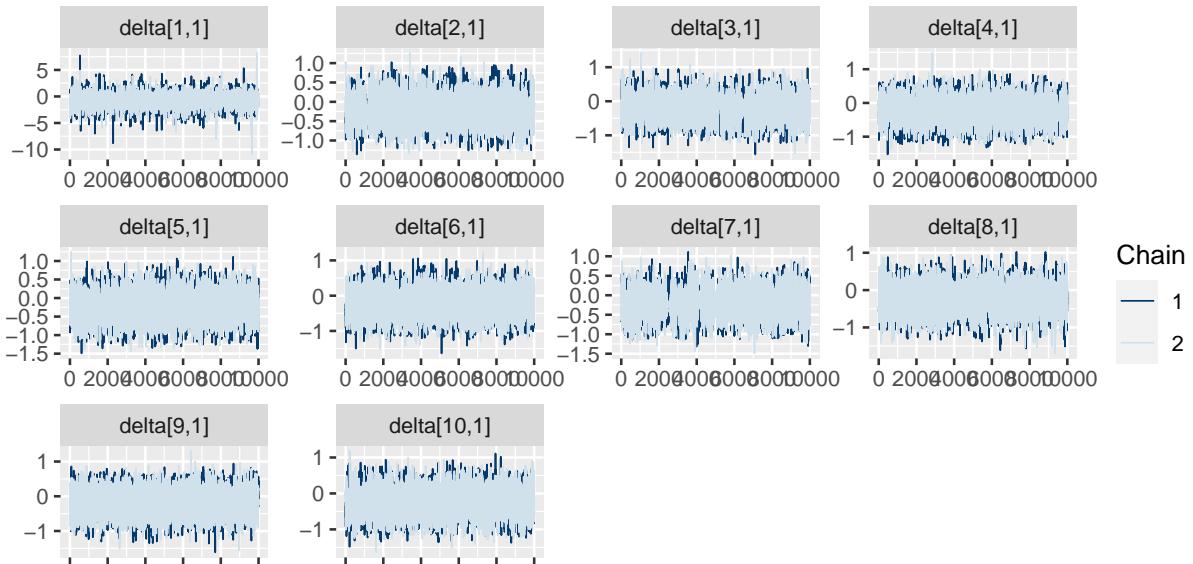
# burn-in
update(sepsis_cv_jm, n.iter = nBurn)

# Parameters to monitor
sepsis_cv_params <- c("p", "mu", "delta", "d", "sd.d", "yrep", "prep")

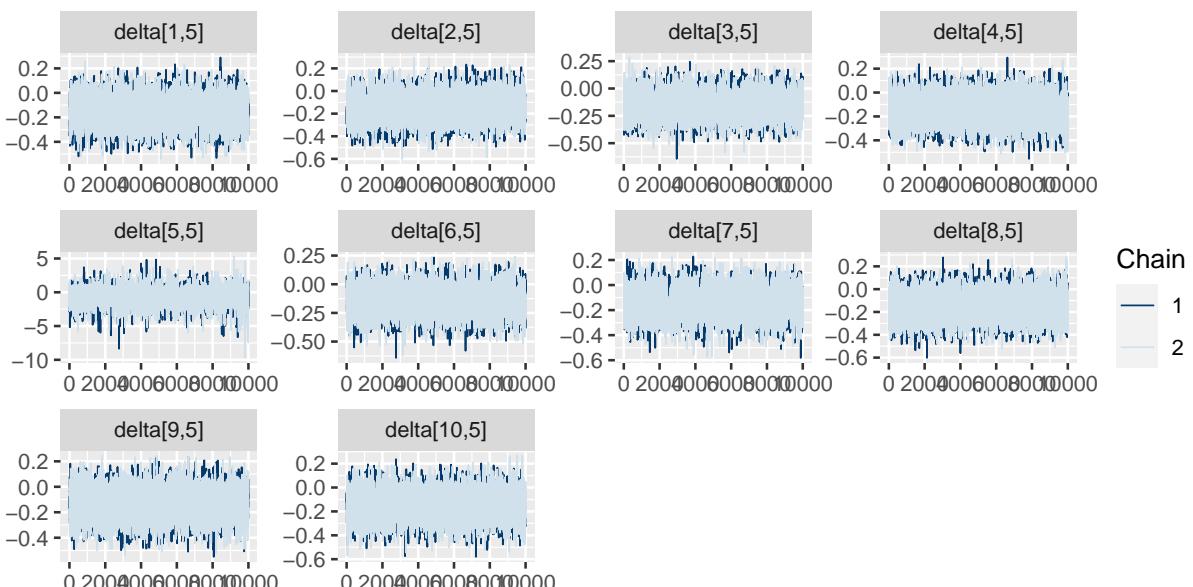
# samples to keep
sepsis_cv_out <- coda.samples(sepsis_cv_jm,
                                variable.names = sepsis_cv_params,
                                n.iter = nIter*2,
                                n.thin = 2)
```

- We find that convergence looks reasonable, see selected traces and summaries below:

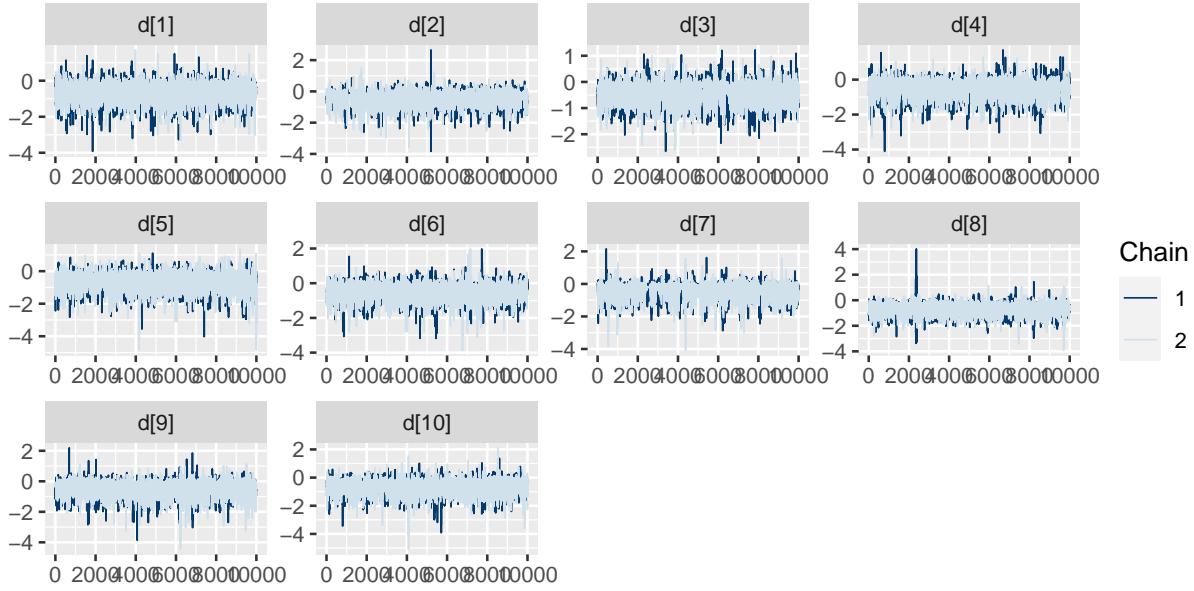
```
# traces
mcmc_trace(sepsis_cv_out,paste0("delta[",1:10," ",1,"]"))
```



```
mcmc_trace(sepsis_cv_out,paste0("delta[",1:10,"],5,")")
```



```
mcmc_trace(sepsis_cv_out,paste0("d[,1:10,]")")
```



```
# posterior summaries
summary(as_draws(sepsis_cv_out))
```

```
## # A tibble: 460 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 d[1]     -0.693 -0.654 0.410 0.354 -1.41 -0.113  1.00 3732.  5759.
## 2 d[2]     -0.484 -0.442 0.349 0.295 -1.09 -0.00618 1.00 2147.  4944.
## 3 d[3]     -0.449 -0.417 0.311 0.275 -1.00 -0.0233  1.00 2129.  5001.
## 4 d[4]     -0.554 -0.514 0.388 0.345 -1.24 -0.0135  1.00 2405.  5572.
## 5 d[5]     -0.696 -0.658 0.412 0.355 -1.41 -0.114  1.00 3187.  5232.
## 6 d[6]     -0.553 -0.512 0.373 0.323 -1.21 -0.0426  1.00 3339.  5269.
## 7 d[7]     -0.443 -0.382 0.361 0.314 -1.10 -0.00260 1.00 749.   2604.
## 8 d[8]     -0.695 -0.665 0.352 0.313 -1.30 -0.212  1.00 2395.  5154.
## 9 d[9]     -0.537 -0.500 0.385 0.330 -1.20 -0.0219  1.00 1929.  4390.
## 10 d[10]    -0.720 -0.687 0.401 0.354 -1.41 -0.159  1.00 2760.  5683.
```

i 450 more rows

```
summary(as_draws(sepsis_cv_out),
~quantile(.x, probs = c(0.025, 0.5, 0.975)))
```

```
## # A tibble: 460 x 4
##   variable `2.5%` `50%` `97.5%
##   <chr>     <dbl>  <dbl>   <dbl>
## 1 d[1]      -1.61 -0.654  0.0185
## 2 d[2]      -1.27 -0.442  0.102
## 3 d[3]      -1.16 -0.417  0.0778
## 4 d[4]      -1.43 -0.514  0.101
## 5 d[5]      -1.62 -0.658  0.0154
## 6 d[6]      -1.38 -0.512  0.0718
## 7 d[7]      -1.29 -0.382  0.0832
## 8 d[8]      -1.47 -0.665 -0.121
## 9 d[9]      -1.39 -0.500  0.0925
## 10 d[10]    -1.60 -0.687 -0.0409
## # i 450 more rows
```

- (iii) Carry out the mixed-predictive checks for the first of the two alternative test statistics discussed in the lecture on slide 36, comparing each $y_{i,k}$ to its predictive distribution $Y_{i,k}^{rep}$. Are any of the individual observations by arm and study substantially different to the corresponding observations in the same arm in each other study?

- First, we manipulate the posterior samples of $y_{i,k}^{rep}$ to get them into a format where it is easy to compare the observations to the corresponding predictive distributions:

```
sepsis_cv_yrep <- as_tibble(as_draws_matrix(sepsis_cv_out),
                           rownames = "Iteration") %>%
  select(
    Iteration, contains("yrep")
  ) %>%
  pivot_longer(
    cols = !contains("Iteration"),
    names_to = "Quantity",
    values_to = "Posterior"
  ) %>%
  mutate(
    StudyNum = as.integer(str_remove(str_remove(Quantity, "yrep\\\"["), ", [1-2]\\\"]")),
    Arm = as.integer(str_remove(str_remove(Quantity, "yrep\\\"[[0-9]+,"), "\\"]")),
    Quantity = "yrep"
  ) %>%
  pivot_wider(
    id_cols = c(Iteration, StudyNum, Arm),
    names_from = Quantity,
    values_from = Posterior
  ) %>%
  left_join(
    tibble(y = as.vector(sepsis_dat$y),
           n = as.vector(sepsis_dat$n),
           StudyNum = rep(1:10, 2),
           Arm = rep(1:2, each = 10)
    ),
    by = c("StudyNum", "Arm")
  ) %>%
  mutate(
    yrep = as.integer(yrep)
  ) %>%
  pivot_wider(
    id_cols = c(Iteration, StudyNum),
    names_from = Arm,
    names_sep = "_",
    values_from = c(yrep, y, n)
  ) %>%
  mutate(
    pval_y_1 = (yrep_1 > y_1),
    pval_y_2 = (yrep_2 > y_2),
    Study = factor(StudyNum, levels = 1:10,
                  labels = StudyNames[1:10])
  )
```

- Then we summarise over iterations, by study number, to find there are no outliers if we are comparing the within-arm observations to each other:

```

sepsis_yrep_pvals <- sepsis_cv_yrep %>%
  group_by(Study) %>%
  summarise(
  pval_y_1 = mean(pval_y_1),
  pval_y_2 = mean(pval_y_2),
  .groups = "keep"
  ) %>%
  ungroup()
sepsis_yrep_pvals

```

```

## # A tibble: 10 x 3
##   Study           pval_y_1   pval_y_2
##   <fct>        <dbl>     <dbl>
## 1 Bussel (1990) 0.528    0.504
## 2 Chirico (1987) 0.552    0.591
## 3 Clapp (1989)  0.591    0.659
## 4 Conway (1990) 0.504    0.516
## 5 Fanaroff (1994) 0.560    0.536
## 6 Haque (1986)  0.579    0.595
## 7 Ratrisawadi (1991) 0.518    0.550
## 8 Sandberg (2000)  0.534    0.480
## 9 Tanzer (1997)   0.555    0.575
## 10 Weisman (1994) 0.591   0.561

```

- (iv) Carry out the mixed-predictive checks for the second of the two test statistics on slide 37, comparing $\text{logit}(y_{\{i,2\}}/n_{\{i,2\}}) - \text{logit}(y_{\{i,1\}}/n_{\{i,1\}})$ to the predictive distribution for δ_i^{rep} . Note that the logit function is defined as $\text{logit}(p) = \log(p / (1-p))$. Are there any outliers? Which of the two test statistics gives a more meaningful test for outliers in this example?

- As in (iii), we combine the posterior samples of δ_i^{rep} with the observations in a single tibble, to compare them:

```

# define logit function
logit <- function(p) {
  log(p / (1-p))
}

# tibble
sepsis_cv_delta <- as_tibble(as_draws_matrix(sepsis_cv_out),
                                rownames = "Iteration") %>%
  select(
  Iteration, contains(paste0("delta[", 1:10, ", ", 1:10, "]"))
  ) %>%
  pivot_longer(
  cols = !contains("Iteration"),
  names_to = "Quantity",
  values_to = "Posterior"
  ) %>%
  mutate(
StudyNum = as.integer(str_remove(str_remove(Quantity, "delta\\\[\"), \",[0-9]+\\\"]")),
Quantity = "LOR",
Posterior = as.double(Posterior)
  ) %>%
  pivot_wider(

```

```

id_cols = c(Iteration, StudyNum),
names_from = Quantity,
values_from = Posterior
) %>%
left_join(
tibble(y_1 = sepsis_dat$y[,1],
y_2 = sepsis_dat$y[,2],
n_1 = sepsis_dat$n[,1],
n_2 = sepsis_dat$n[,2],
StudyNum = 1:10
),
by = "StudyNum"
) %>%
mutate(
pobs_1 = y_1 / n_1,
pobs_2 = y_2 / n_2,
LOR_loo = logit(pobs_2) - logit(pobs_1),
pval_LOR = (LOR > LOR_loo),
Study = factor(StudyNum, levels = 1:10,
               labels = StudyNames[1:10])
)

```

- Then summarise over iterations, by study number, finding that both the Clapp and Sandberg studies appear to be outliers:

```

sepsis_delta_pvals <- sepsis_cv_delta %>%
  group_by(Study) %>%
  summarise(
    pval_LOR = mean(pval_LOR),
    .groups = "keep"
  ) %>%
  ungroup()
sepsis_delta_pvals

## # A tibble: 10 x 2
##   Study           pval_LOR
##   <fct>        <dbl>
## 1 Bussel (1990) 0.256
## 2 Chirico (1987) 0.902
## 3 Clapp (1989)  1
## 4 Conway (1990) 0.703
## 5 Fanaroff (1994) 0.253
## 6 Haque (1986)  0.728
## 7 Ratrisawadi (1991) 0.871
## 8 Sandberg (2000) 0.0473
## 9 Tanzer (1997)  0.784
## 10 Weisman (1994) 0.177

```

- We can reproduce the plot on slide 39 of the lecture notes to visualise the outliers:

```

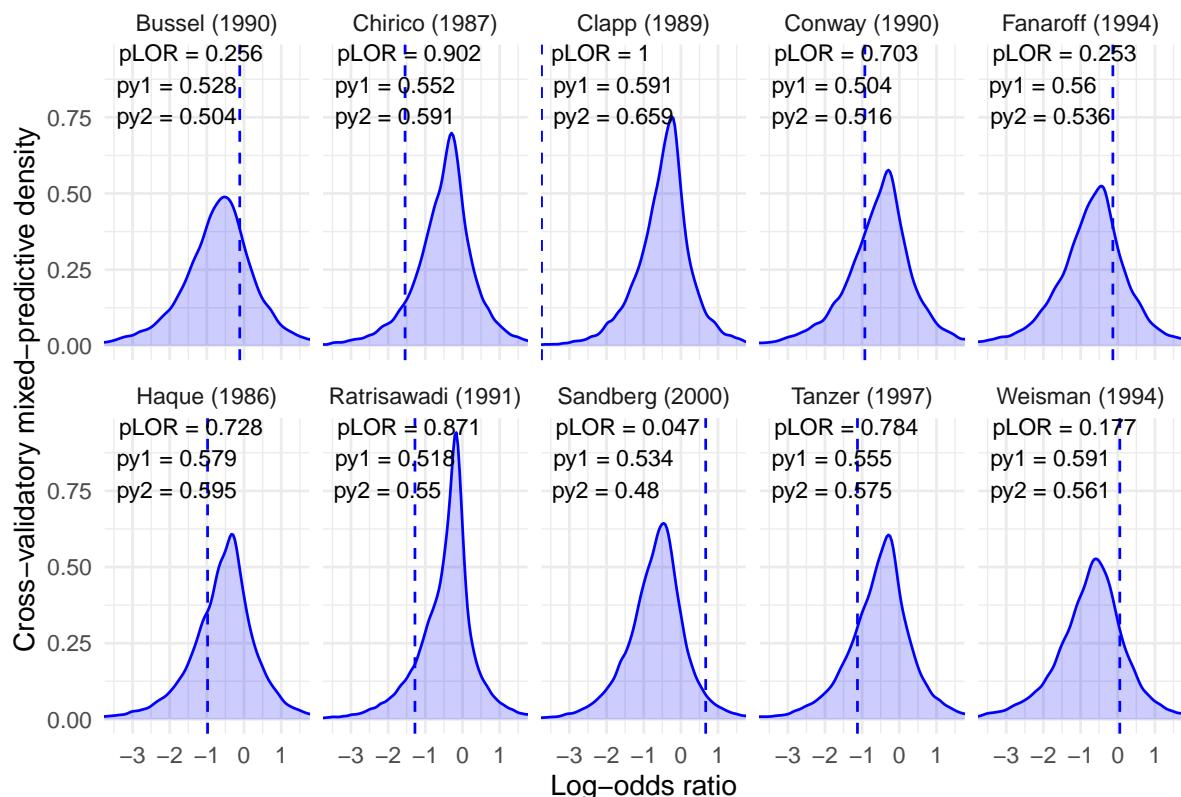
sepsis_pvals <- sepsis_yrep_pvals %>%
  left_join(sepsis_delta_pvals, by = "Study")
sepsis_cv_delta %>%
  ggplot() +

```

```

aes(x = LOR) +
  geom_density(alpha = 0.2, col = "blue", fill = "blue") +
  labs(x = "Log-odds ratio", y = "Cross-validatory mixed-predictive density") +
  coord_cartesian(xlim = c(-3.5, 1.5)) +
  facet_wrap(vars(Study), nrow = 2) +
  geom_vline(aes(xintercept = LOR_loo), col = "blue", linetype = "dashed") +
  geom_text(
    data = sepsis_pvals,
    mapping = aes(x = -Inf, y = Inf, label = paste0("pLOR = ", round(pval_LOR,3), "\n",
                                                    "py1 = ", round(pval_y_1,3), "\n",
                                                    "py2 = ", round(pval_y_2,3))),
    hjust = -0.1,
    vjust = 1,
    size = 3
  ) +
  theme_minimal() +
  theme(legend.position = "none")

```



- (B) This final set of exercises explores the systematic bias adjustment described in slides 40-45 of the lecture notes. Turner et al, JRSS(A) (2008) introduced the idea of systematic bias adjustment for meta-analysis where some of the included studies are considered less rigorous or relevant than others. By incorporating prior information on the effect possible biases in such studies might have on the estimated treatment effect, the evidence from less rigorous or relevant studies may still be included, but down-weighted. Since then, much work has been done on systematic bias adjustment, see the Cochrane Collaboration’s “Risk of bias” tool (<https://training.cochrane.org/handbook/current/chapter-08>). The “BRANDO” study (Savovic et al, Ann Intern Med, 2012) estimated the effect that different flaws in study design have on the estimated treatment effect: these estimates can be used as prior information to incorporate

in a systematic bias adjustment model as suggested by Turner et al (2008).

- (i) Below is an outline of the code to implement the bias adjustment in the random effects sepsis model. Complete the model code by replacing the question marks with appropriate expressions for the biased version of the log odds ratio, δ_i^{bias} ; and for the bias parameters β_i . Assume, as in slide 44 that there is no bias for the “safe” set of studies; and that there is a common mean and standard deviation for the biases in the “risky” set of studies. Following Savovic et al (2012), choose a mean and standard deviation such that the prior multiplicative bias effect on the odds ratio is 0.82 (0.67-1) (i.e. convert this prior information to the log scale). This prior came from averaging different types of bias effect in Savovic et al (2012):

- The biased version of the log odds ratio is $\delta_i^{bias} = \delta_i + \beta_i$, where β_i is either 0, if study i is “safe”; or comes from a prior $N(\nu = -0.2, \sigma^2 = 0.1^2)$ distribution if study i is “risky”:

```
sepsis_re_bias_model <- "
model
{
  # For each study, Ns = total number of studies
  for(i in 1:Ns)
  {
    # for each of the two arms
    for(k in 1:Na)
    {
      # Binomial likelihood
      y[i,k] ~ dbin(p[i,k], n[i,k])
    }

    # On a logit scale, the proportion p is the probability
    # of infection in terms of the study baselines mu and
    # the study-specific treatment effects delta (log odds ratios,
    # relative to the study baseline).
    # Here we replace delta with the biased version delta_bias
    logit(p[i,1]) <- mu[i]
    logit(p[i,2]) <- mu[i] + delta_bias[i]

    # and express the biased log odds ratio in terms of the unbiased
    delta_bias[i] <- delta[i] + beta[i]

    # replicate biased log odds ratio in terms of replicate unbiased log odds ratio for
    # mixed predictive checks
    delta_bias_rep[i] <- delta_rep[i] + beta[i]

    # for the treatment arm, log odds ratios are random effects
    # with a common mean d
    delta[i] ~ dnorm(d, prec.d)

    # mixed-predictive replicates
    delta_rep[i] ~ dnorm(d, prec.d)

    # study-specific baselines, vague priors
    mu[i] ~ dnorm(0, 0.01)
  }

  # Priors for basic parameters:
}
```

```

# mean log odds ratio of treatment vs control
d ~ dnorm(0, 0.01)

# sd of study-specific log odds ratios
prec.d <- 1 / (sd.d * sd.d)
sd.d ~ dunif(0,10)

# informative priors for studies with at least one high risk of bias
# otherwise no bias
for(j in 1:Nsafe) {
  beta[safe[j]] <- beta.safe
}
beta.safe <- 0
for(j in 1:Nrisky) {
  beta[risky[j]] <- beta.risky[j]
  beta.risky[j] ~ dnorm(nu, prec.beta)
}
}

# data: original data, plus indicators for the safe and risk sets
# of studies; and the parameters of the prior for the biases for
# the risky set
sepsis_re_bias_dat <- sepsis_dat
sepsis_re_bias_dat$Nsafe = 4
sepsis_re_bias_dat$Nrisky = 6
sepsis_re_bias_dat$safe = c(3,5,8,10)
sepsis_re_bias_dat$risky = c(1:2,4,6:7,9)
sepsis_re_bias_dat$nu = -0.2
sepsis_re_bias_dat$prec.beta = 1 / (0.1^2)

```

- (ii) Run the model to obtain estimates of the bias-adjusted mean and study-specific treatment effects. How do the bias-adjusted treatment effects compare to the unadjusted estimates from the random effects model stored in `sepsis_post` from `sepsis.R`? Reproduce the plot on slide 44 to visually compare the estimates.

- Setting initial values and running the model:

```

# Inits
sepsis_re_bias_inits <- list(
  list(d = 0,
       sd.d = 5,
       mu = rep(0,sepsis_dat$Ns),
       delta = rep(0,sepsis_dat$Ns),
       beta.risky = rep(0,sepsis_re_bias_dat$Nrisky),
       .RNG.name = c("base::Mersenne-Twister"),
       .RNG.seed = c(7195)),
  list(d = 0.1,
       sd.d = 1,
       mu = c(1,-1,-2,0,0,-2,1,0,2,2),
       delta = rep(1,sepsis_dat$Ns),
       beta.risky = rep(-1,sepsis_re_bias_dat$Nrisky),
       .RNG.name = c("base::Mersenne-Twister")),

```

```

.RNG.seed = c(168422)
)

# Initialise model
sepsis_re_bias_jm <- jags.model(textConnection(sepsis_re_bias_model),
                                 data = sepsis_re_bias_dat,
                                 inits = sepsis_re_bias_inits,
                                 n.chains = nChains)

## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
##   Observed stochastic nodes: 20
##   Unobserved stochastic nodes: 38
##   Total graph size: 150
##
## Initializing model
# burn-in
update(sepsis_re_bias_jm, n.iter = nBurn)

# Parameters to monitor
sepsis_re_bias_params <- c("p", "mu", "delta", "d", "sd.d", "delta_bias", "beta", "delta_bias_rep")

# samples to keep
sepsis_re_bias_out <- coda.samples(sepsis_re_bias_jm,
                                      variable.names = sepsis_re_bias_params,
                                      n.iter = nIter,
                                      n.thin = 1)

```

- We extract the estimated bias-adjusted odds ratios and add them to the tibble `sepsis_post` from `sepsis.R` that contains the estimates from the other models:

```

sepsis_bias_post <- sepsis_post %>%
  bind_rows(
  as_tibble(as_draws_matrix(sepsis_re_bias_out), rownames = "Iteration") %>%
    select(Iteration, contains("delta["), d) %>%
    pivot_longer(
      cols = contains("d"),
      names_to = "StudyNum",
      values_to = "LOR"
    ) %>%
    mutate(
      StudyNum = if_else(
        str_detect(StudyNum, "delta"),
        as.integer(str_remove(str_remove(StudyNum, "delta\\\["), "\\]")),
        as.integer(99)),
      Study = factor(StudyNum, levels = c(1:10, 99),
                    labels = c(StudyNames[1:10], "Mean")),
      Model = "Bias-adjusted random effects",
      OR = as.double(exp(LOR))
    )
  )

```

```

) %>%
  select(-StudyNum)
)

sepsis_bias_plt <- sepsis_bias_post %>%
  group_by(Model, Study) %>%
  summarise(
  Median = median(OR),
  Lower = quantile(OR, probs = 0.025),
  Upper = quantile(OR, probs = 0.975),
  .groups = "keep"
) %>%
  ungroup() %>%
  mutate(
  Model = factor(Model, levels = c("Common effect", "Independent effects",
                                    "Random effects", "Bias-adjusted random effects"))
)

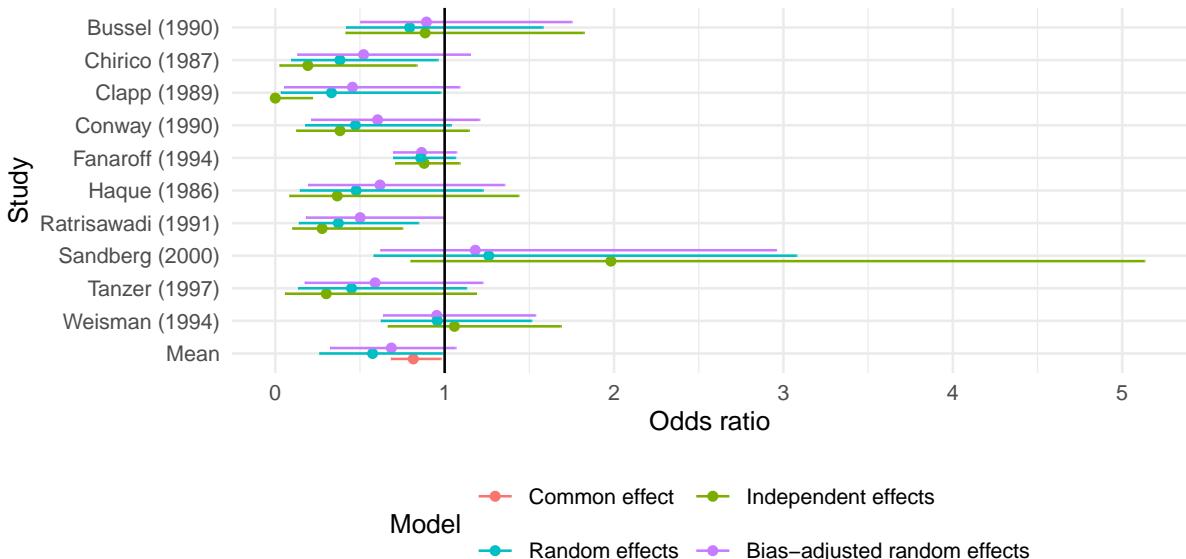
```

- And plot them to compare the estimates visually. We find that bias-adjustment moderates the estimates towards 1 for the “risky” set of studies, and that the bias-adjusted mean effect credible interval no longer excludes 1:

```

sepsis_bias_plt %>%
  ggplot() +
  aes(x = Median, y = reorder(Study, desc(Study)), col = Model) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbarh(aes(xmin = Lower, xmax = Upper, height = 0),
                 position = position_dodge(width = 0.5)) +
  geom_vline(xintercept = 1) +
  labs(x = "Odds ratio", y = "Study") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  guides(col=guide_legend(nrow=2,byrow=TRUE))

```



- (iii) Calculate the mixed-predictive p-values (without cross-validation) for both the unadjusted replicate

study-specific treatment effects from the bias model; and the study-specific replicate treatment effects from the random effects model in `sepsis_re_out` (named `delta_rep[i]` for study i). Are there any differences? Even though these p-values are more conservative than the cross-validatory ones in (A), do you still find outliers?

- First join the replicate log odds ratios in the random effects model from `sepsis_post` in `sepsis.R` with the unadjusted replicate log odds ratios from the bias model (rather than the bias-adjusted ones we considered in (ii)) and with the data in a single tibble, to compare observations to their posterior predictions:

```

sepsis_bias_unadjusted_post <- as_tibble(as_draws_matrix(sepsis_re_out),
                                         rownames = "Iteration") %>%
  select(Iteration, contains("delta_rep"), d) %>%
  pivot_longer(
    cols = contains("d"),
    names_to = "StudyNum",
    values_to = "LORrep"
  ) %>%
  mutate(
    StudyNum = if_else(
      str_detect(StudyNum, "delta_rep"),
      as.integer(str_remove(str_remove(StudyNum, "delta_rep\\\""), "\\\"")),
      as.integer(99)
    ),
    Study = factor(StudyNum, levels = c(1:10, 99),
                  labels = c(StudyNames[1:10], "Mean")),
    Model = "Random effects",
    ORrep = as.double(exp(LORrep))
  ) %>%
  bind_rows(
    as_tibble(as_draws_matrix(sepsis_re_bias_out), rownames = "Iteration") %>%
      select(Iteration, contains("delta_bias_rep["), d) %>%
      pivot_longer(
        cols = contains("d"),
        names_to = "StudyNum",
        values_to = "LORrep"
      ) %>%
      mutate(
        StudyNum = if_else(
          str_detect(StudyNum, "delta_bias"),
          as.integer(str_remove(str_remove(StudyNum, "delta_bias_rep\\\""), "\\\"")),
          as.integer(99)
        ),
        Study = factor(StudyNum, levels = c(1:10, 99),
                      labels = c(StudyNames[1:10], "Mean")),
        Model = "Bias-adjusted random effects",
        ORrep = as.double(exp(LORrep))
      ) %>%
      select(-StudyNum)
  )
)

sepsis_bias_unadjusted_post <- sepsis_bias_unadjusted_post %>%
  left_join(
  tibble(y_1 = sepsis_dat$y[, 1],
         y_2 = sepsis_dat$y[, 2],
         n_1 = sepsis_dat$n[, 1],

```

```

n_2 = sepsis_dat$n[, 2],
StudyNum = 1:10,
Study = factor(StudyNum, levels = c(1:10, 99),
               labels = c(StudyNames[1:10], "Mean"))
) %>%
  select(-StudyNum),
by = "Study"
) %>%
  mutate(
pobs_1 = y_1 / n_1,
pobs_2 = y_2 / n_2,
LOR_obs = logit(pobs_2) - logit(pobs_1),
pval_LOR = (LORrep > LOR_obs)
)

```

- Summarise over iterations, by study and model, to obtain the p-values. We find that only the Clapp study is identified as an outlier; unlike with the cross-validatory mixed-predictive p-values, where the Sandberg study was also found to be an outlier, demonstrating the conservatism of the non-cross-validatory p-values:

```

sepsis_bias_pvals <- sepsis_bias_unadjusted_post %>%
  filter(Study != "Mean") %>%
  group_by(Study, Model) %>%
  summarise(
pval_LOR = mean(pval_LOR),
.groups = "keep"
) %>%
  ungroup() %>%
  pivot_wider(
id_cols = Study,
names_from = Model,
values_from = pval_LOR
)
sepsis_bias_pvals %>% print(width = Inf)

```

```

## # A tibble: 10 x 3
##   Study           `Bias-adjusted random effects` `Random effects`
##   <fct>                      <dbl>                  <dbl>
## 1 Bussel (1990)                0.193                 0.268
## 2 Chirico (1987)               0.901                 0.883
## 3 Clapp (1989)                  1                     1
## 4 Conway (1990)                0.728                 0.706
## 5 Fanaroff (1994)               0.319                 0.271
## 6 Haque (1986)                  0.748                 0.729
## 7 Ratrisawadi (1991)              0.848                 0.824
## 8 Sandberg (2000)                0.0468                0.0614
## 9 Tanzer (1997)                  0.808                 0.777
## 10 Weisman (1994)                 0.200                 0.189

```