BDA - Project

Anonymous

Contents


```
knitr::opts_chunk$set(echo = TRUE, fig.height = 6, fig.width = 10)
# load libraries
pkgs <- c("tidyr", "ggplot2", "rstan", "bayesplot", "parallel", "loo",
          "readxl")
sapply(pkgs, require, character.only = TRUE)
```

```
##     tidyr   ggplot2     rstan bayesplot  parallel       loo    readxl
##      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
```

```
# set number of default cores
options(mc.cores = detectCores())
# set plot theme
theme_set(theme_minimal())
# set colour scheme
color_scheme_set("blue")
```

Note: this is the contents page. The assignment starts on the following page.

1. Introduction

This project is motivated by a thesis I finished in 2020. The thesis included a pairwise meta-analysis to estimate the efficacy of the Bivalent Human Papillomavirus vaccine. It was originally coded in JAGS, did not have any proper priors, did not include coefficients, and it was used to inform a fully-integrated Bayesian health economics model. The original JAGS model was coded as:

```
# likelihood and prior. Model parameters are abbreviated by .vac.
  for (i in 1:Nstud.vac) {
    # Likelihood:
     rA.vac[i] ~ dbin(pA.vac[i], nA.vac[i])
     rB.vac[i] ~ dbin(pB.vac[i], nB.vac[i])

    # Logistic link function:
     logit(pA.vac[i]) <- mu.vac[i]
     logit(pB.vac[i]) <- mu.vac[i] + delta.vac[i]

    # Average effect prior for SUB-MODEL 2:
     mu.vac[i] ~ dnorm(0, 1e-4)
    # Prior for sub-model 2 (Random. pop. effect):
     delta.vac[i] ~ dt(psi.vac, prec.vac, 1)
     # if desired can be ~ dnorm(psi.vac, prec.vac)

     ### Mixed predictive check for SUB-MODEL 2:
       # Predictive likelihood:
        rA.mxd[i] ~ dbin(pA.new[i], nA.vac[i])

       # Predictive logit link function:
        logit(pA.new[i]) <- mu.vac[i] + delta.new

       # Mixed predictve p-value:
        pA.mxd[i] <- step(rA.mxd[i] - rA.vac[i]) - 0.5 * equals(rA.mxd[i], rA.vac[i])
```

Since finishing the thesis, I have wanted to redo the meta-analysis in Stan. Note that this project won't, however, include the economic model. Please see a simple illustration of the model below (note that some of the final model assumptions may differ).

Given the above motivation, I will implement the original model in Stan and include proper priors, which will be given logical boundaries when necessary. Coefficients will also be included where applicable. I will also compare grouped and hierarchical models.

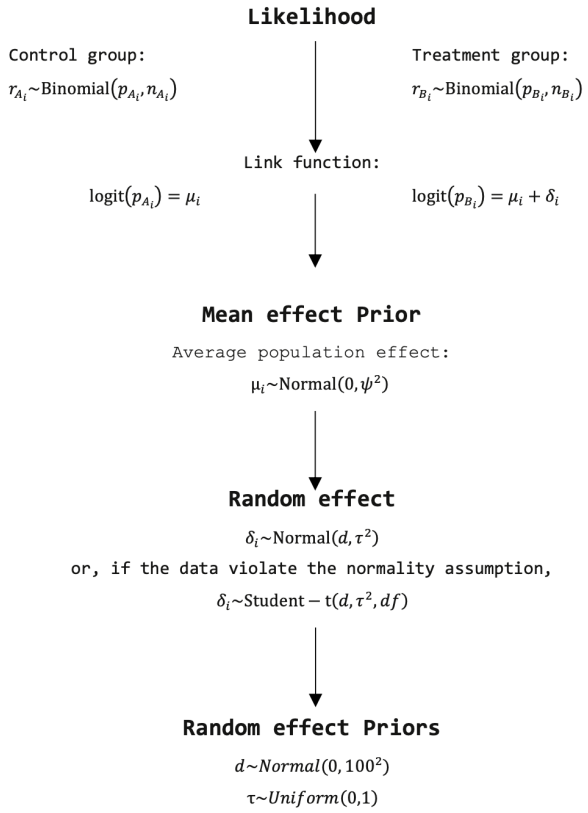

Figure 1: Model illustration. Note that some parameter distributions for the final are different, since informative priors have been added

2. Data

The data for the model are printed below. Note that references for each input can be found in the original Excel file.

```
# load data
df <- read_excel("data/data_vaccine_case_control.xlsx")
```

```
## New names:
## * `` -> `...5`
```

```
# print data frame
df
```

```
## # A tibble: 11 x 5
##    y_vaccine n_vaccine y_control n_control ...5
##        <dbl>     <dbl>     <dbl>     <dbl> <chr>
##  1        22      5331       212      5291 Apter et al. (2015). Efficacy of Hum~
##  2        12       366        41       355 Harper et al. (2004). Efficacy of a ~
##  3         1       310        28       277 Harper et al. (2006). Sustained effi~
##  4         3      2190        38      2239 Herrero et al. (2011). Prevention of~
##  5        61      2910       219      2924 Herrero et al. (2013). Reduced Preva~
##  6         0       387        15       392 Konno et al. (2010). Efficacy of Hum~
##  7         0       193        10       175 Naud et al. (2014). Sustained effica~
##  8         2      7788        21      7838 Paavonen et al. (2007). Efficacy of ~
##  9         4      7344        56      7838 Paavonen et al. (2009). Efficacy of ~
## 10         0       401        20       372 Romanowski et al. (2009). Sustained ~
## 11         1      2497        17      2502 Zhu et al. (2014). Efficacy, immunog~
```

We can then prepare the data for Stan as follows:

```
# n studies
n_s <- nrow(df)
# y events in control arm
y_0 <- df$y_control
# y events in vaccine arm
y_1 <- df$y_vaccine
# n observations in control arm
n_0 <- df$n_control
# n observations in vaccine arm
n_1 <- df$n_vaccine
# data list
data_list <- list(n_s = n_s, y_0 = y_0, n_0 = n_0, y_1 = y_1, n_1 = n_1)
```

The model will thus loop through an $i \times j$ matrix, with i rows and j columns, where there are i studies with j arms.

3. Model

The general model is a simple binomial model with several priors and hyperiors (same as above). However, to provide an understanding of whether a separate or hierarchical model is better suited to the problem at

hand, we can ought to run the model as both a separate or hierarchical model, respectively. These different approaches to the model have a slightly different structure, which is detailed in the '*Model background and results*' section below.

3.1 Model background and results

See below for a brief discussion and illustration of each model variation.

3.3.1 The Separate model

The separate model is the simpler of the two types (in terms of the number of parameters). Notationally, this model can be represented as having a likelihood

$$y_{ij} \sim Bin(n_{ij}, p_{ij})$$

The Stan code for the separate model is coded as

```
# print results
print(project_sep, pars = c("p_eff[1]", "p_eff[2]", "p_eff[3]", "p_eff[4]",
                            "p_eff[5]", "p_eff[6]", "p_eff[7]", "p_eff[8]",
                            "p_eff[9]", "p_eff[10]", "p_eff[11]"))
```

```
## Inference for Stan model: project_sep.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
## p_eff[1]  0.91       0 0.02 0.87 0.90 0.91 0.92  0.94  1831    1
## p_eff[2]  0.80       0 0.05 0.68 0.76 0.80 0.83  0.89  1675    1
## p_eff[3]  0.97       0 0.03 0.90 0.96 0.98 0.99  1.00  2365    1
## p_eff[4]  0.93       0 0.04 0.84 0.91 0.93 0.96  0.99  2359    1
## p_eff[5]  0.79       0 0.02 0.75 0.78 0.79 0.81  0.84  1921    1
## p_eff[6]  1.00       0 0.01 0.99 1.00 1.00 1.00  1.00  4032    1
## p_eff[7]  1.00       0 0.01 0.98 1.00 1.00 1.00  1.00  3866    1
## p_eff[8]  0.92       0 0.05 0.78 0.89 0.93 0.96  0.99  2519    1
## p_eff[9]  0.93       0 0.03 0.85 0.91 0.94 0.95  0.98  2145    1
## p_eff[10] 1.00       0 0.01 0.99 1.00 1.00 1.00  1.00  3222    1
## p_eff[11] 0.95       0 0.05 0.82 0.93 0.96 0.98  1.00  2610    1
##
## Samples were drawn using NUTS(diag_e) at Wed May 18 16:02:44 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

3.3.2 The Hierarchical model

```
# print results
project_hrchl
```

```
## Inference for Stan model: project_hrchl.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##              mean se_mean   sd     2.5%      25%      50%      75%    97.5%
## p_eff        0.92    0.00 0.02     0.88     0.91     0.92     0.94     0.96
## odds_eff     0.09    0.00 0.02     0.04     0.07     0.08     0.10     0.14
## psi         -2.50    0.01 0.30    -3.18    -2.69    -2.48    -2.30    -1.96
## tau          0.41    0.01 0.22     0.13     0.25     0.36     0.51     0.96
## lp__     -3630.42    0.22 5.09 -3641.05 -3633.72 -3630.28 -3626.83 -3621.06
##          n_eff Rhat
## p_eff     1525 1.00
## odds_eff  1557 1.00
## psi       1395 1.00
## tau        835 1.01
## lp__       540 1.01
##
## Samples were drawn using NUTS(diag_e) at Wed May 18 16:03:08 2022.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

3.2 Analysing the models