

Worksheet 5

Foundations of Bayesian Methodology

Wenjie Tu

Spring Semester 2022

Exercise 1 (Individual project (Part 5A))

1. Apply the logit-transformation

$$X \sim \text{Bin}(n, \pi)$$

$$\hat{\phi}_{\text{ML}} = \text{logit}(\hat{\pi}_{\text{ML}}) = \log\left(\frac{\hat{\pi}_{\text{ML}}}{1 - \hat{\pi}_{\text{ML}}}\right) = \log\left(\frac{x}{n - x}\right)$$

$$\text{se}(\hat{\phi}_{\text{ML}}) = \sqrt{\frac{1}{x} + \frac{1}{n - x}}$$

```
pl_total <- c(107, 44, 51, 39, 139, 20, 78, 35)
pl_case <- c(23, 12, 19, 9, 39, 6, 9, 10)

## Logit-transformation
pl_y <- log(pl_case / (pl_total - pl_case))

## Standard error
pl_se <- sqrt(1/pl_case + 1/(pl_total - pl_case))

prec_s <- 1 / pl_se^2
```

2. Implement the code for a random effects meta-analysis in JAGS

```
pl1_modelString <- "model{
  for (i in 1:length(y)) {
    y[i] ~ dnorm(theta[i], prec_s[i]);
    theta[i] ~ dnorm(mu, prec_tau);
  }

  theta_new ~ dnorm(mu, prec_tau); # predictive distribution for theta

  # predictive distribution at the probability scale
  p_new <- exp(theta_new) / (1 + exp(theta_new));

  mu ~ dnorm(0.0, 1.0E-4);
  prec_tau ~ dgamma(1.0E-3, 1.0E-3); # just our assumption
}"

writeLines(pl1_modelString, con="MetaAnalysis.txt")
```

Parametrization

$$\begin{aligned} y_i &\sim N(\theta_i, 1/\tau_i^s) \\ \theta_i &\sim N(\mu, 1/\tau) \end{aligned} \quad \text{for } i = 1, \dots, N$$

$$\begin{aligned} \mu &\sim N(0, 100^2) \\ \tau &\sim G(0.001, 0.001) \end{aligned} \quad \text{priors for parameters}$$

Data required for JAGS:

$$\begin{aligned} y_i &= \log\left(\frac{x_i}{n_i - x_i}\right) \\ \tau_i^s &= 1 / \left(\frac{1}{x_i} + \frac{1}{n_i - x_i}\right) \end{aligned}$$

```
library(rjags)
library(coda)
library(ggplot2)
```

```
## Generate a data list for JAGS
dat.jags <- list(y=pl_y, prec_s=prec_s)

## Generate a list for initial values for JAGS
## Set initial values and seed for reproducible results
inits.jags <- list(
  list(mu=100, prec_tau=0.01, .RNG.name="base::Wichmann-Hill", .RNG.seed=314159),
  list(mu=10, prec_tau=0.05, .RNG.name="base::Marsaglia-Multicarry", .RNG.seed=159314),
  list(mu=-100, prec_tau=0.001, .RNG.name="base::Super-Duper", .RNG.seed=413159),
  list(mu=-10, prec_tau=0.1, .RNG.name="base::Mersenne-Twister", .RNG.seed=143915)
)

## Create a JAGS model object
model.jags <- jags.model(
  file = "MetaAnalysis.txt", # text file for JAGS
  data = dat.jags,          # the data list for JAGS
  inits = inits.jags,        # the list for initialization
  n.chains = 4,              # the number of parallel chains for the model
  n.adapt = 4000             # the number of iterations for adaption
)
```

```
## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 8
##   Unobserved stochastic nodes: 11
##   Total graph size: 34
##
## Initializing model
```

```
## Use the first 4000 iterations as burn-in iterations
update(model.jags, n.iter = 4000)

## Generate posterior samples
fit.jags.coda <- coda.samples(
  model = model.jags,        # JAGS model object
  variable.names = c("p_new"), # variable to be monitored
```

```

n.iter = 10000,          # number of iterations to monitor
thin = 1                 # thinning interval for monitors
)

```

3. Explain what is being done in each line of the code

See the comments in the code chunks.

4. Summarize the posterior predictive distribution for `p_new`

```

## Print the summary results of the posterior samples generated
summary(fit.jags.coda)

```

```

##
## Iterations = 4001:14000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean           SD      Naive SE Time-series SE
##    0.2585754    0.0633004    0.0003165    0.0004144
##
## 2. Quantiles for each variable:
##
##    2.5%    25%    50%    75%   97.5%
## 0.1416 0.2259 0.2543 0.2843 0.4069

```

```

## Concatenate 4 chains
m.fit.jags.coda <- as.matrix(fit.jags.coda)

d.mcmc <- data.frame(
  Iterations = rep(4001:14000, times=4),
  ResponseRate = m.fit.jags.coda[, "p_new"],
  chainType = rep(c("chain1", "chain2", "chain3", "chain4"), each=10000)
)

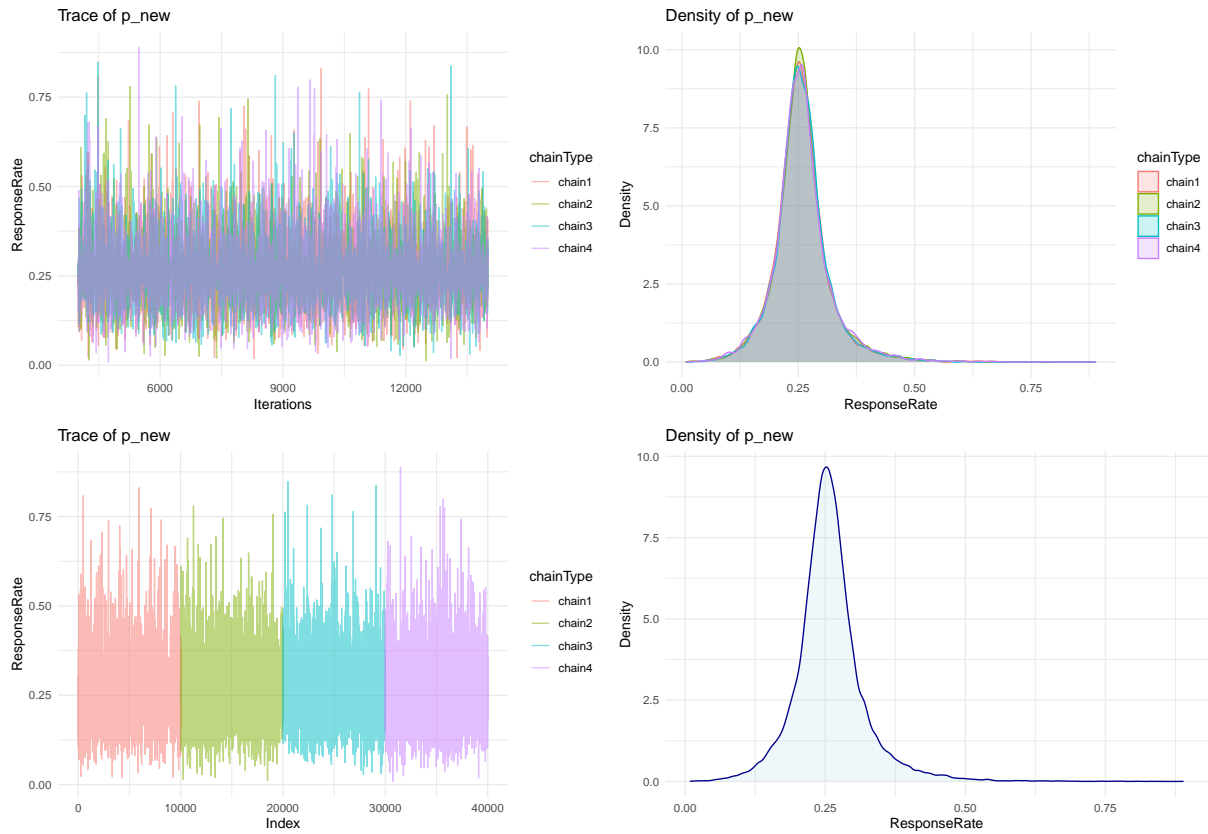
ggplot(d.mcmc, aes(x=Iterations, y=ResponseRate, color=chainType)) +
  geom_line(alpha=0.5) + labs(title="Trace of p_new") + theme_minimal()

ggplot(d.mcmc, aes(x=ResponseRate, y=..density.., fill=chainType, color=chainType)) +
  geom_density(alpha=0.2) + labs(title="Density of p_new", y="Density") + theme_minimal()

ggplot(d.mcmc, aes(x=1:nrow(d.mcmc), y=ResponseRate, color=chainType)) +
  geom_line(alpha=0.5) + labs(title="Trace of p_new", x="Index") + theme_minimal()

ggplot(d.mcmc, aes(x=ResponseRate, y=..density..)) +
  geom_density(alpha=0.2, color="darkblue", fill="lightblue") +
  labs(title="Density of p_new", y="Density") + theme_minimal()

```



```
d.summary <- data.frame(t(rbind(
  colMeans(m.fit.jags.coda),
  apply(m.fit.jags.coda, 2, function(x) sd(x)),
  apply(m.fit.jags.coda, 2, function(x) var(x)),
  apply(m.fit.jags.coda, 2, function(x) quantile(x, probs=c(0.025, 0.5, 0.975)))
)))
colnames(d.summary) <- c("Mean", "SD", "Variance", "2.5%", "Median", "97.5%")
knitr::kable(d.summary, digits=5, align="c", caption="Summary statistics")
```

Table 1: Summary statistics

	Mean	SD	Variance	2.5%	Median	97.5%
p_new	0.25858	0.0633	0.00401	0.14158	0.25427	0.4069

5. Compare with the results obtained in Baeten's study

```
d.comparison <- data.frame(
  c(d.summary$Mean, 0.25),
  c(d.summary$`2.5%`, 0.13),
  c(d.summary$`97.5%`, 0.4)
)
colnames(d.comparison) <- c("Mean", "2.5%", "97.5%")
rownames(d.comparison) <- c("Meta-analysis", "Baeten's study")
knitr::kable(d.comparison, digits=3, align="c", caption="Comparison")
```

Table 2: Comparision

	Mean	2.5%	97.5%
Meta-analysis	0.259	0.142	0.407
Baeten's study	0.250	0.130	0.400

The estimate from the meta-analysis and its 95% credible interval agree with those reported in Baeten et al. (2013)

6. Apply the function for moment matching

In Exercise 2 of Worksheet 2, we derived

$$\begin{cases} \alpha = \left(\frac{1-\mu}{\sigma^2} - \frac{1}{\mu} \right) \mu^2 \\ \beta = \alpha \left(\frac{1}{\mu} - 1 \right) \end{cases}$$

```
## Moment matching function
estBetaParams <- function(mean, var) {
  alpha <- ((1 - mean) / var - 1 / mean) * mean ^ 2
  beta <- alpha * (1 / mean - 1)
  return(params = c(alpha=alpha, beta=beta))
}
```

7. Report the estimates

```
estBetaParams(mean=d.summary$Mean, var=d.summary$Variance)
```

```
##      alpha      beta
## 12.11310 34.73241
```

8. Explain the approach to a client

In this exercise, we attempt to replicate the results from Baeten's study and see whether the similar results can be obtained.

- We first gather the data from 8 historical studies
- We next apply logit-transformation to response rate for placebo group to obtain an approximately normal distribution of logit-transformed rates
- We then conduct a random-effects meta-analysis to derive the posterior predictive distribution of response rate
- We next summarize the posterior predictive distribution of response rates (i.e., mean and 95% credible interval)
- We then apply the function for moment matching to derive the parameters α and β of the Beta prior for the response rate in the placebo group.

Exercise 2 (Individual project (Part 5B))

```
## Set seed for reproducible results
set.seed(44566)

## Monte Carlo sample size
```

```
M <- 100000
```

```
## Generate i.i.d. random samples
```

```
p.p <- rbeta(M, shape1=11, shape2=32) # response rate in Placebo
```

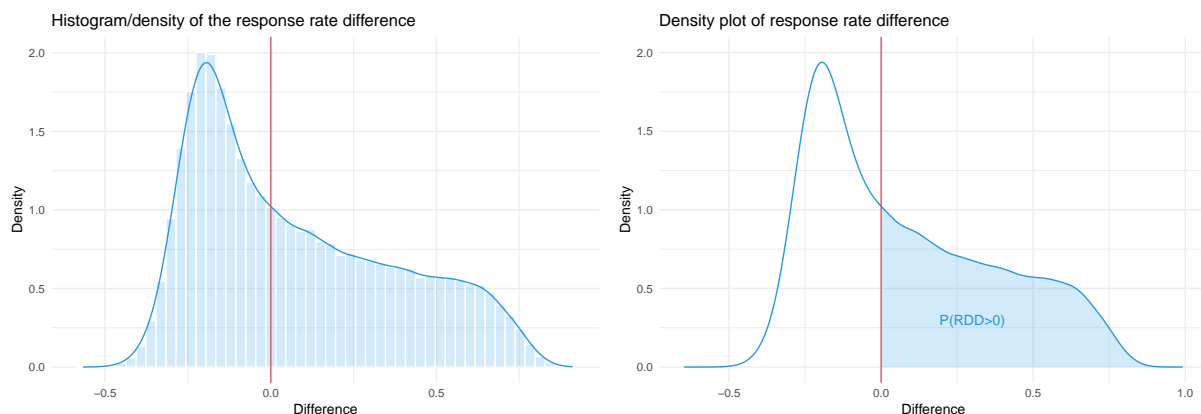
```
p.t <- rbeta(M, shape1=0.5, shape2=1) # response rate in Treatment
```

```
## Construct response rate difference (RRD)
```

```
rrd <- p.t - p.p
```

```
ggplot(data.frame(Difference=rrd), aes(x=Difference, y=..density..)) +
  geom_histogram(position="identity", bins=50, alpha=0.2, color="white", fill=4) +
  geom_density(alpha=0.2, color=4) + geom_vline(xintercept=0, color=2) + theme_minimal() +
  labs(title="Histogram/density of the response rate difference", y="Density")
```

```
ggplot(with(density(rrd), data.frame(x, y)), aes(x=x, y=y)) +
  geom_area(aes(x=ifelse(x>0, x, 0), y=y), fill=4, alpha=0.2) +
  geom_line(color=4) + geom_vline(xintercept=0, color=2) +
  labs(title="Density plot of response rate difference", x="Difference", y="Density") +
  geom_text(aes(x=0.3, y=0.3), label="P(RDD>0)", color=4, check_overlap=TRUE) +
  ylim(0, 2) + theme_minimal()
```



```
## Posterior probability of superiority of treatment over placebo
```

```
pps <- mean(rrd>0); pps
```

```
pps.se <- sqrt(var(rrd>0)/M); pps.se
```

```
## 99% confidence interval
```

```
c(pps-3*pps.se, pps+3*pps.se)
```

```
## [1] 0.50046
```

```
## [1] 0.001581146
```

```
## [1] 0.4957166 0.5052034
```

Beta(0.5,1) is used for treatment group and it has been shown that with such priors there is an approximately 50:50 chance that the response rate on active treatment would be greater than the response rate on placebo.

Approach:

- Generate a vector of i.i.d. random samples from Beta(11, 32) as the response rates for the placebo group
- Generate a vector of i.i.d. random samples from Beta(0.5, 1) as the response rates for the treatment group

- Construct the response rate difference between the response rates in the treatment group and the response rates in the placebo group
- Calculate the posterior probability of superiority and the corresponding Monte Carlo standard error