

# Worksheet 6

## Foundations of Bayesian Methodology

Wenjie Tu      Lea Bühner      Jerome Sepin      Zhixuan Li  
Elia-Leonid Mastropietro      Jonas Raphael Füglistaler

Spring Semester 2022

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

### Exercise 3 (Bayesian meta-analysis with bayesmeta)

$$y = \log(\text{OR}) = \log \frac{x_P}{n_P - x_P} - \log \frac{x_T}{n_T - x_T}$$

$$\sigma = \text{SE}(\log(\text{OR})) = \sqrt{\frac{1}{x_P} + \frac{1}{n_P - x_P} + \frac{1}{x_T} + \frac{1}{n_T - x_T}}$$

```
pl_total <- c(107, 44, 51, 39, 139, 20, 78, 35)
pl_case <- c(23, 12, 19, 9, 39, 6, 9, 10)
tr_total <- c(208, 38, 150, 45, 138, 20, 201, 34)
tr_case <- c(120, 18, 107, 26, 82, 16, 126, 23)
log_or <- log(pl_case/(pl_total-pl_case)) - log(tr_case/(tr_total-tr_case))
log_or_se <- sqrt(1/pl_case + 1/(pl_total-pl_case) + 1/tr_case + 1/(tr_total-tr_case))
labels <- 1:length(pl_total)
```

```
dat <- data.frame(labels, tr_case, tr_total, pl_case, pl_total, log_or, log_or_se)
knitr::kable(dat, align="c", digits=3, caption="Historical data for meta-analysis")
```

Table 1: Historical data for meta-analysis

labels	tr_case	tr_total	pl_case	pl_total	log_or	log_or_se
1	120	208	23	107	-1.605	0.274
2	18	38	12	44	-0.875	0.469
3	107	150	19	51	-1.433	0.341
4	26	45	9	39	-1.518	0.485
5	82	138	39	139	-1.323	0.256
6	16	20	6	20	-2.234	0.742
7	126	201	9	78	-2.556	0.383
8	23	34	10	35	-1.654	0.524

Bayesian normal-normal hierarchical model (NNHM) with three levels of hierarchy:

Likelihood:

$$y_i \sim N(\theta_i, \sigma_i^2)$$

for  $i = 1, \dots, k$

Random effects:

$$\theta_i \sim N(\mu, \tau^2)$$

Priors:

$$\mu \sim N(\nu, \gamma^2)$$

$$\tau \sim |N(0, A^2)| = \text{HN}(A)$$

where  $\nu = 0, \gamma = 4, A = 0.5$

```
MA.bayesmeta <- bayesmeta(y = dat[, "log_or"],
                          sigma = dat[, "log_or_se"],
                          labels = dat[, "labels"],
                          mu.prior.mean = 0, mu.prior.sd = 4,
                          tau.prior = function(t){dhalfnormal(t, scale = 0.5)},
                          interval.type = "central" )
```

```
summary(MA.bayesmeta)
```

```
## 'bayesmeta' object.
## data (8 estimates):
##           y      sigma
## 1 -1.6054775 0.2740073
## 2 -0.8754687 0.4691896
## 3 -1.4329256 0.3412963
## 4 -1.5176304 0.4853221
## 5 -1.3229761 0.2563070
## 6 -2.2335922 0.7420210
## 7 -2.5556757 0.3832411
## 8 -1.6538897 0.5238200
##
## tau prior (proper):
## function(t){dhalfnormal(t, scale = 0.5)}
## <bytecode: 0x0000000023f64440>
##
## mu prior (proper):
## normal(mean=0, sd=4)
##
## ML and MAP estimates:
##           tau      mu
## ML joint    0.2094171 -1.592280
## ML marginal 0.2852879 -1.590235
## MAP joint    0.1614761 -1.585174
## MAP marginal 0.2334117 -1.587618
##
## marginal posterior summary:
##           tau      mu      theta
## mode      0.2334117 -1.5876182 -1.5805059
## median    0.2702386 -1.5919563 -1.5884808
## mean      0.2949284 -1.5946544 -1.5946544
## sd        0.1941244 0.1879906 0.4002409
## 95% lower 0.0153332 -1.9777397 -2.4509871
## 95% upper 0.7397310 -1.2281569 -0.7646060
##
## (quoted intervals are central, equal-tailed credible intervals.)
```

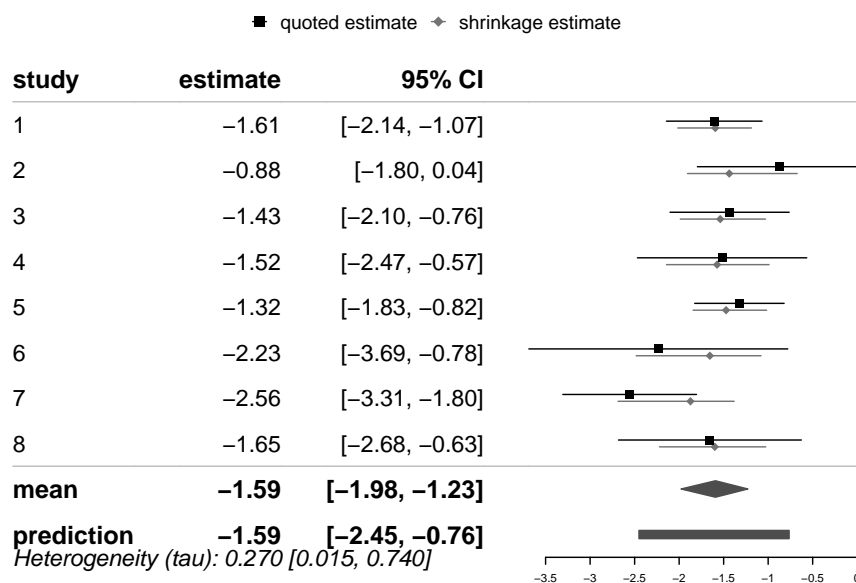
```
##
## Bayes factors:
##      tau=0      mu=0
## actual 1.0209152 3.11865e-05
## minimum 0.7030068 1.23805e-06
##
## relative heterogeneity I2 (posterior median): 0.3343206
```

```
knitr::kable(t(MA.bayesmeta$summary), align="c", digits=4,
              caption="Summary statistics for parameters (bayesmeta)")
```

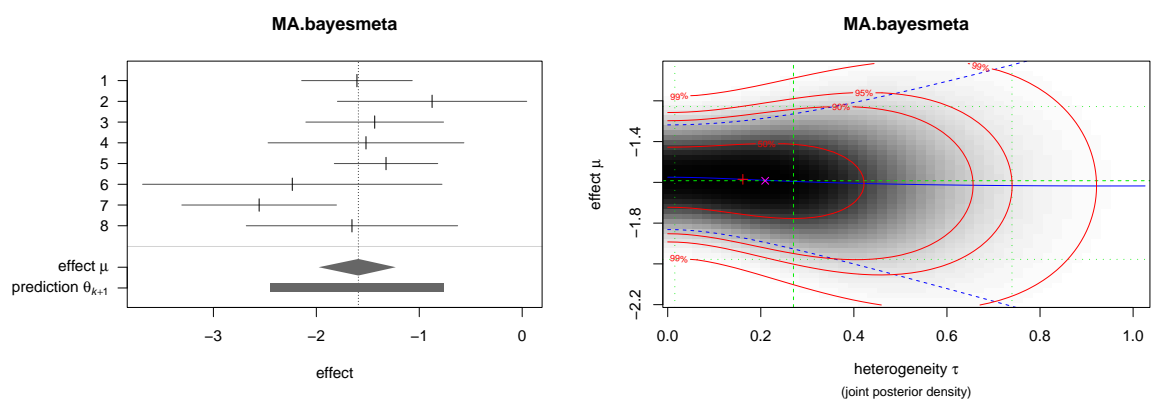
Table 2: Summary statistics for parameters (bayesmeta)

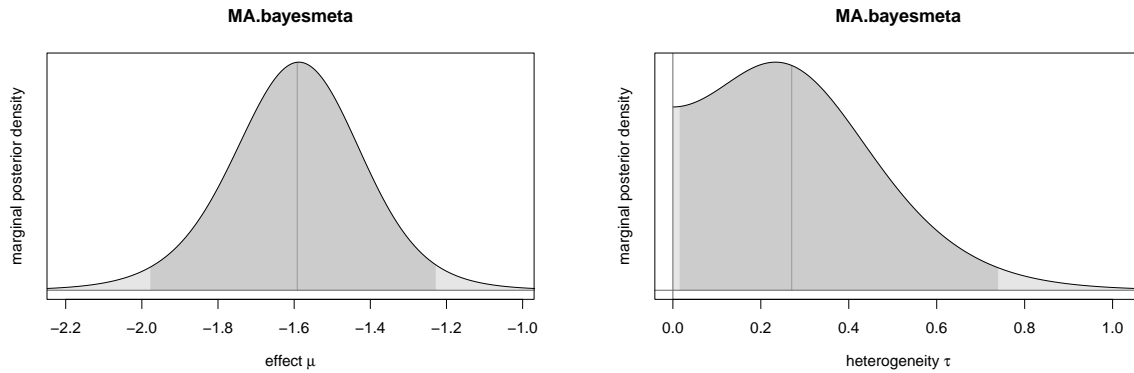
	mode	median	mean	sd	95% lower	95% upper
tau	0.2334	0.2702	0.2949	0.1941	0.0153	0.7397
mu	-1.5876	-1.5920	-1.5947	0.1880	-1.9777	-1.2282
theta	-1.5805	-1.5885	-1.5947	0.4002	-2.4510	-0.7646

```
forestplot(MA.bayesmeta)
```



```
plot(MA.bayesmeta)
```





## Exercise 4 (Bayesian meta-analysis with JAGS)

Likelihood:

$$y_j \sim \text{Bin}(n_j, p_j)$$

$$\eta_j \sim N(0, 1/\tau_{\text{prec}})$$

for  $i = 1, \dots, k$ , where  $\tau_{\text{prec}} = 1/\tau^2$

Priors:

$$\mu \sim U(-10, 10)$$

$$\beta \sim U(-10, 10)$$

$$\tau \sim U(0, 10)$$

```
p11.data <- list(
  N = 16,
  y = c(23., 12., 19., 9., 39., 6., 9., 10., 120., 18., 107., 26., 82., 16., 126., 23.),
  n = c(107., 44., 51., 39., 139., 20., 78., 35., 208., 38., 150., 45., 138., 20., 201., 34.),
  C1 = c(0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 1., 1., 1., 1., 1.)
)
```

```
p11.params <- c("mu", "beta", "tau", "p1.star", "p2.star")
```

```
p11_modelString <- "model {
  # sampling model (likelihood)
  for (j in 1:N) {
    y[j] ~ dbin(p[j], n[j])
    logit(p[j]) <- mu + beta * C1[j] + eta[j]
    eta[j] ~ dnorm(0, tau.prec)

    # prediction for posterior predictive checks
    y.pred[j] ~ dbin(p[j], n[j])
    PPC[j] <- step(y[j] - y.pred[j]) - 0.5 * equals(y[j], y.pred[j])
  }

  # priors
  mu ~ dunif(-10, 10)
  beta ~ dunif(-10, 10)
  tau ~ dunif(0, 10)
  tau.prec <- 1/tau/tau

  # population effect
  p1 <- 1/(1+exp(-mu))
  p2 <- 1/(1+exp(-mu-beta))
}
```

```

# predictive distribution for new study effect
eta.star ~ dnorm(0, tau.prec)
p1.star <- 1/(1+exp(-mu-eta.star))
p2.star <- 1/(1+exp(-mu-beta-eta.star))
}"

writeLines(pl1_modelString, con="./models/MetaAnalysis.txt")

```

```

# model initiation
rjags.pl1 <- jags.model(
  file = "./models/MetaAnalysis.txt",
  data = pl1.data,
  n.chains = 4,
  n.adapt = 4000
)

```

```

## Compiling model graph
##   Resolving undeclared variables
##   Allocating nodes
## Graph information:
##   Observed stochastic nodes: 16
##   Unobserved stochastic nodes: 36
##   Total graph size: 222
##
## Initializing model

```

```

# burn-in
update(rjags.pl1, n.iter = 4000)

# sampling/monitoring
fit.rjags.pl1.coda <- coda.samples(
  model = rjags.pl1,
  variable.names = pl1.params,
  n.iter = 10000,
  thin = 1
)

summary(fit.rjags.pl1.coda)

```

```

##
## Iterations = 8001:18000
## 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
## beta      1.6166 0.21653 0.0010827    0.0056055
## mu       -1.1086 0.15816 0.0007908    0.0037545
## p1.star   0.2542 0.06709 0.0003355    0.0007110
## p2.star   0.6213 0.08043 0.0004021    0.0006933
## tau       0.2968 0.13461 0.0006731    0.0034719
##
## 2. Quantiles for each variable:

```

```
##
##           2.5%    25%    50%    75%    97.5%
## beta      1.19700  1.4809  1.6134  1.7534  2.0479
## mu       -1.42156 -1.2126 -1.1094 -1.0058 -0.7968
## p1.star   0.13670  0.2124  0.2484  0.2885  0.4094
## p2.star   0.44526  0.5778  0.6237  0.6688  0.7785
## tau       0.04182  0.2091  0.2876  0.3734  0.5917

m.fit.rjags.pl1.coda <- as.matrix(fit.rjags.pl1.coda)
d.chains <- data.frame(
  iterations = rep(8001:18000, times=4),
  chains = rep(c("chain1", "chain2", "chain3", "chain4"), each=10000),
  beta = m.fit.rjags.pl1.coda[, "beta"],
  mu = m.fit.rjags.pl1.coda[, "mu"],
  p1.star = m.fit.rjags.pl1.coda[, "p1.star"],
  p2.star = m.fit.rjags.pl1.coda[, "p2.star"],
  tau = m.fit.rjags.pl1.coda[, "tau"]
)

ggplot(d.chains, aes(x=iterations, y=beta, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of beta", x="Iterations") + theme_minimal()

ggplot(d.chains, aes(x=beta, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of beta", y="Density") + theme_minimal()

ggplot(d.chains, aes(x=iterations, y=mu, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of mu", x="Iterations") + theme_minimal()

ggplot(d.chains, aes(x=mu, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of mu", y="Density") + theme_minimal()

ggplot(d.chains, aes(x=iterations, y=p1.star, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of p1.star", x="Iterations") + theme_minimal()

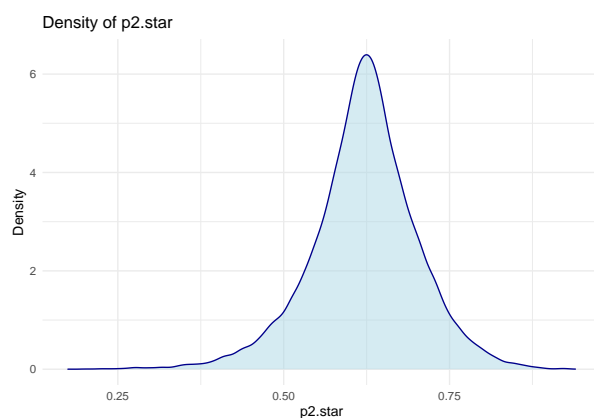
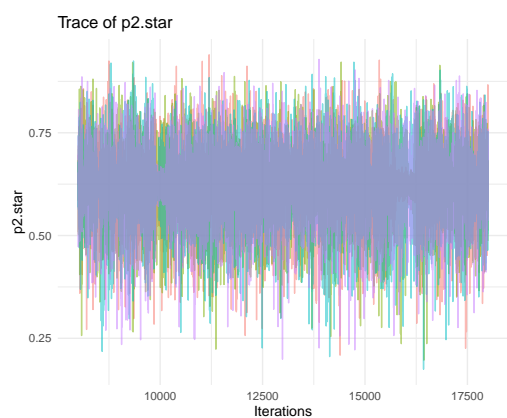
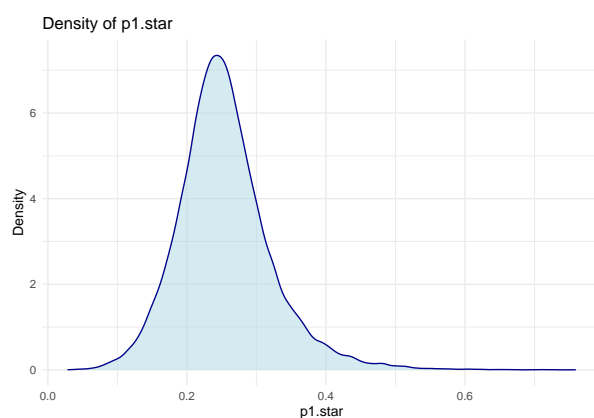
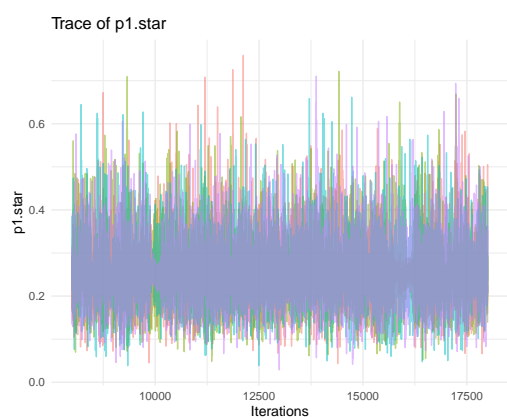
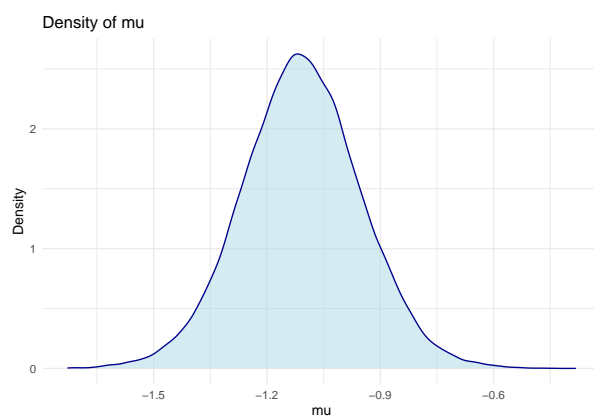
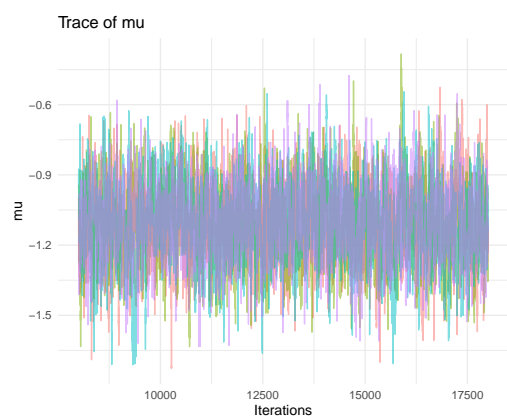
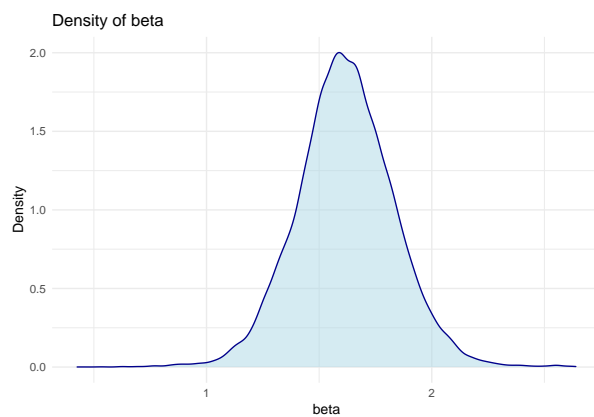
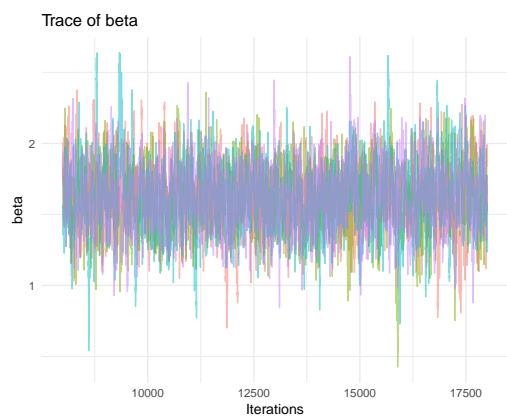
ggplot(d.chains, aes(x=p1.star, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of p1.star", y="Density") + theme_minimal()

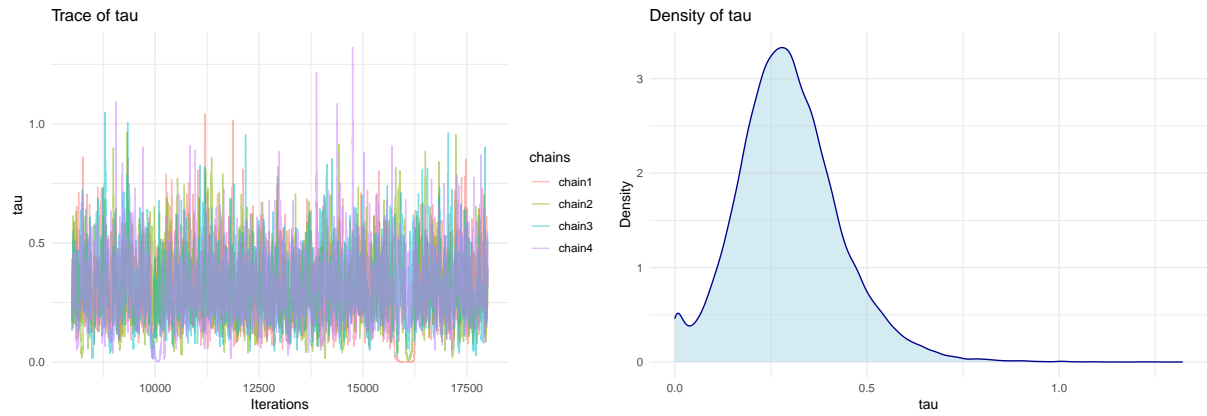
ggplot(d.chains, aes(x=iterations, y=p2.star, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of p2.star", x="Iterations") + theme_minimal()

ggplot(d.chains, aes(x=p2.star, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of p2.star", y="Density") + theme_minimal()

ggplot(d.chains, aes(x=iterations, y=tau, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of tau", x="Iterations") + theme_minimal()

ggplot(d.chains, aes(x=tau, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of tau", y="Density") + theme_minimal()
```





```
d.summary <- t(rbind(
  colMeans(m.fit.rjags.pl1.coda),
  apply(m.fit.rjags.pl1.coda, 2, function(x) sd(x)),
  apply(m.fit.rjags.pl1.coda, 2, function(x) quantile(x, probs=c(0.025, 0.5, 0.975)))
))

colnames(d.summary) <- c("Mean", "SD", "2.5%", "Median", "97.5%")
knitr::kable(d.summary, align="c", digits=4, caption="Summary statistics for parameters (JAGS)")
```

Table 3: Summary statistics for parameters (JAGS)

	Mean	SD	2.5%	Median	97.5%
beta	1.6166	0.2165	1.1970	1.6134	2.0479
mu	-1.1086	0.1582	-1.4216	-1.1094	-0.7968
p1.star	0.2542	0.0671	0.1367	0.2484	0.4094
p2.star	0.6213	0.0804	0.4453	0.6237	0.7785
tau	0.2968	0.1346	0.0418	0.2876	0.5917

### Model (Exercise 1 of Worksheet 5)

In this model, we first apply the logit-transformation to  $p_i = x_i/n_i$  to get an approximately normal distribution of logit-transformed rates. We then use the delta method to compute the standard of logit-transformed rates.

$$y_i = \text{logit}(p_i) = \log \frac{p_i}{1-p_i} = \log \frac{x_i}{n_i - x_i}$$

$$\sqrt{\frac{1}{\tau_i^s}} = \text{SE}(y_i) = \sqrt{\frac{1}{x_i} + \frac{1}{n_i - x_i}}$$

The full Bayesian meta-analysis is conducted using the Bayesian normal-normal hierarchical model (NNHM) with three levels of hierarchy:

Likelihood:

$$y_i \sim N(\theta_i, 1/\tau_i^s)$$

for  $i = 1, \dots, N$

Random effects:

$$\theta_i \sim N(\mu, 1/\tau)$$

Priors:

$$\mu \sim N(0, 100^2)$$

$$\tau \sim G(0.001, 0.001)$$



### Model (Exercise 3 of Worksheet 6)

This model uses the same idea as for the model in Exercise 1 of Worksheet 5. The only difference is that in this model we consider the historical data for both placebo and treatment groups. We first compute the so-called log odds ratio, which is simply the difference between logit-transformed rates in the placebo group and logit-transformed rates in the treatment group. We then use the formula from [Held and Sabanes Bove, 2020, p. 137–138] to compute the standard error of the log odds ratio.

$$y = \log(\text{OR}) = \log \frac{x_P}{n_P - x_P} - \log \frac{x_T}{n_T - x_T}$$

$$\sigma = \text{SE}(\log(\text{OR})) = \sqrt{\frac{1}{x_P} + \frac{1}{n_P - x_P} + \frac{1}{x_T} + \frac{1}{n_T - x_T}}$$

The full Bayesian meta-analysis is conducted using the Bayesian normal-normal hierarchical model (NNHM) with three levels of hierarchy:

Likelihood:

$$y_i \sim N(\theta_i, \sigma_i^2)$$

for  $i = 1, \dots, k$

Random effects:

$$\theta_i \sim N(\mu, \tau^2)$$

Priors:

$$\begin{aligned}\mu &\sim N(\nu, \gamma^2) \\ \tau &\sim |N(0, A^2)| = \text{HN}(A)\end{aligned}$$

where  $\nu = 0, \gamma = 4, A = 0.5$

### Model (Exercise 4 of Worksheet 6)

Unlike models stated before, this model uses a linear regression with a normal error ( $\eta_j$ ) to directly model the number of responders with only one predictor indicating whether in the treatment or not.

$$y_j = \mu + \beta \cdot \text{C1}_j + \eta_j$$

where  $\text{C1}_i$  is a binary variable which is equal to 0 if placebo and 1 otherwise.

Likelihood:

$$\begin{aligned}y_j &\sim \text{Bin}(n_j, p_j) \\ \eta_j &\sim N(0, 1/\tau_{\text{prec}})\end{aligned}$$

for  $i = 1, \dots, k$ , where  $\tau_{\text{prec}} = 1/\tau^2$

Priors:

$$\begin{aligned}\mu &\sim U(-10, 10) \\ \beta &\sim U(-10, 10) \\ \tau &\sim U(0, 10)\end{aligned}$$

## Exercise 5 (Moments of the Poisson-gamma distribution)

## Exercise 6 (Empirical Bayes)