

# 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: 0x0000000023f07238>
##
## 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.)
```

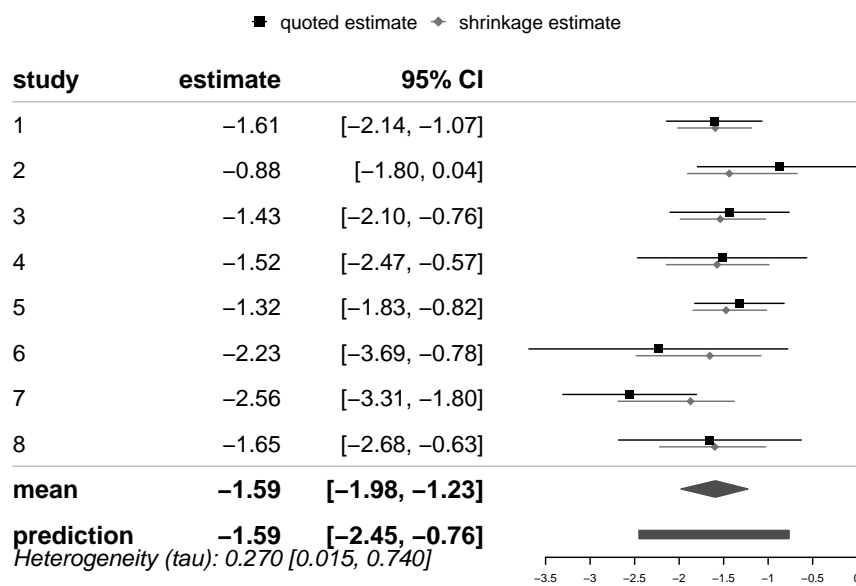
```
##
## Bayes factors:
##          tau=0          mu=0
## actual  1.0209152 3.11865e-05
## minimum 0.7030068 1.23805e-06
##
## relative heterogeneity I2 (posterior median): 0.3343206
```

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

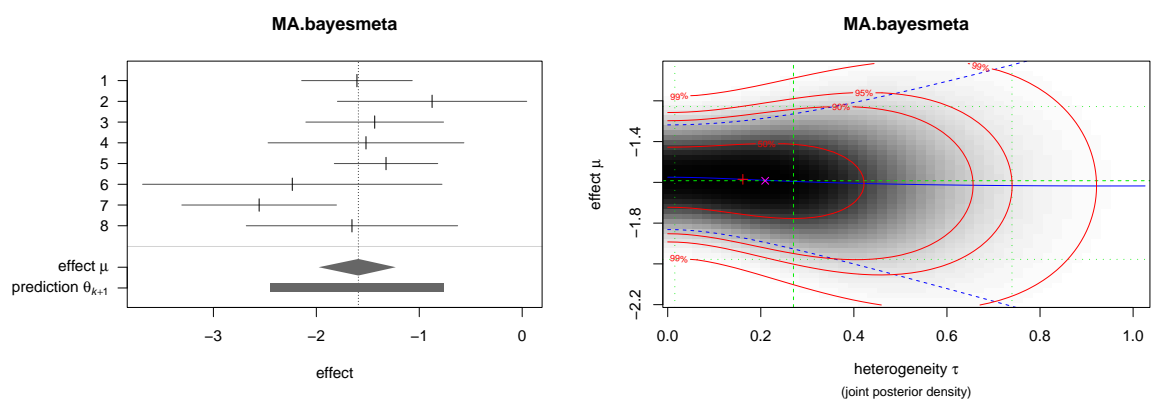
Table 2: Summary statistics for parameters (bayesmeta)

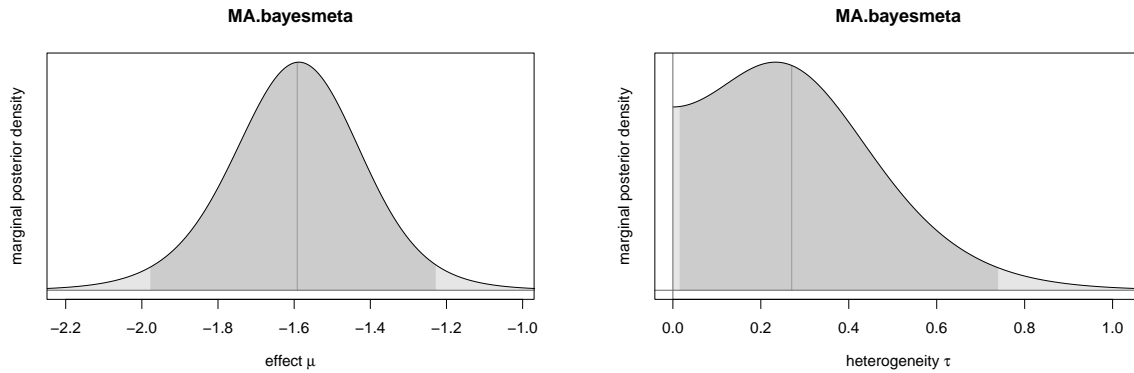
	mode	median	mean	sd	95% lower	95% upper
tau	0.2334	0.2702	0.2949	0.1941	0.0153	0.7397
mu	-1.5876	-1.5920	-1.5947	0.1880	-1.9777	-1.2282
theta	-1.5805	-1.5885	-1.5947	0.4002	-2.4510	-0.7646

```
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      1.6155 0.22030 0.0011015    0.0058266
## mu       -1.1101 0.16400 0.0008200    0.0040682
## p1.star   0.2534 0.06756 0.0003378    0.0007778
## p2.star   0.6199 0.08103 0.0004052    0.0006949
## tau       0.2993 0.13298 0.0006649    0.0033197
##
## 2. Quantiles for each variable:

```

```
##
##           2.5%    25%    50%    75%    97.5%
## beta      1.17934  1.4780  1.6157  1.7499  2.0517
## mu        -1.43915 -1.2118 -1.1088 -1.0072 -0.7894
## p1.star   0.13296  0.2113  0.2480  0.2880  0.4094
## p2.star   0.44261  0.5756  0.6226  0.6683  0.7780
## tau       0.07098  0.2097  0.2868  0.3748  0.6045

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()

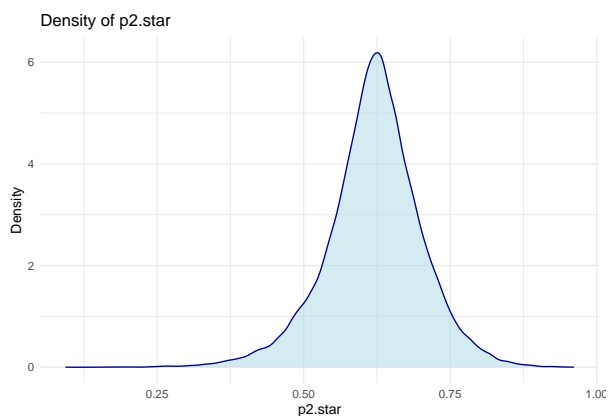
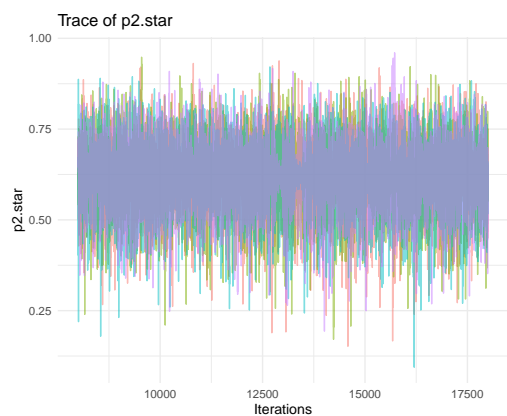
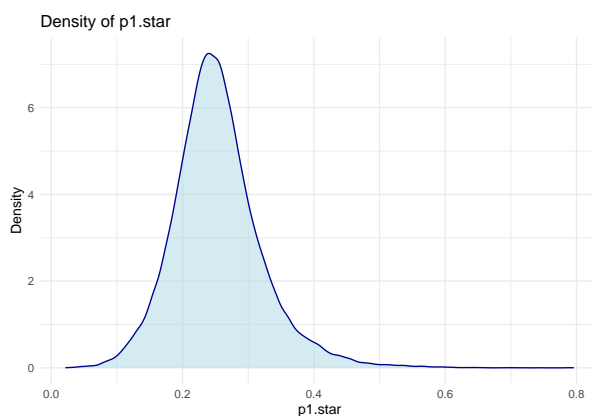
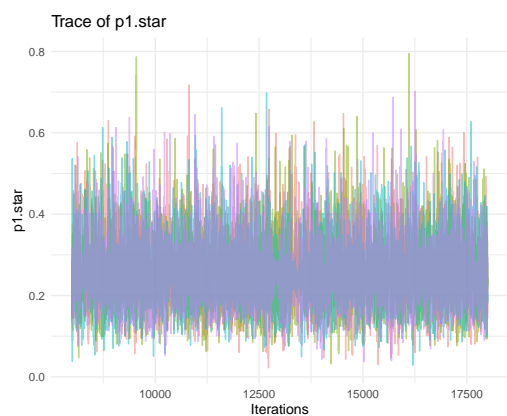
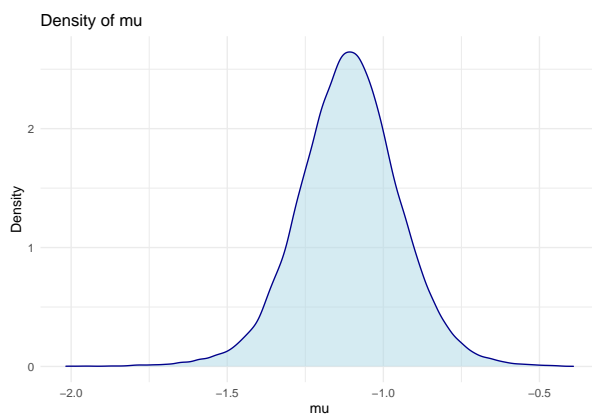
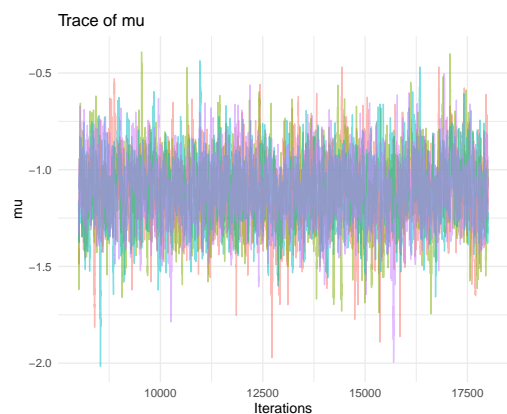
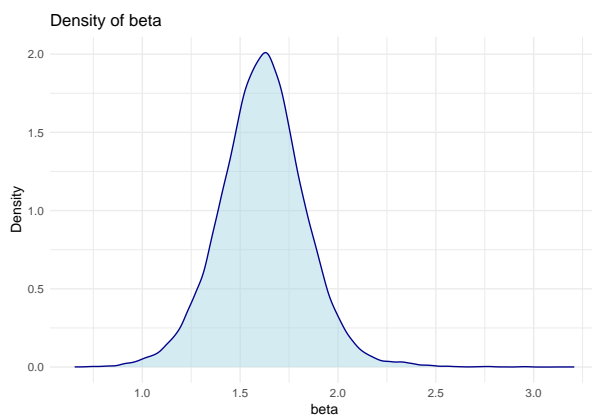
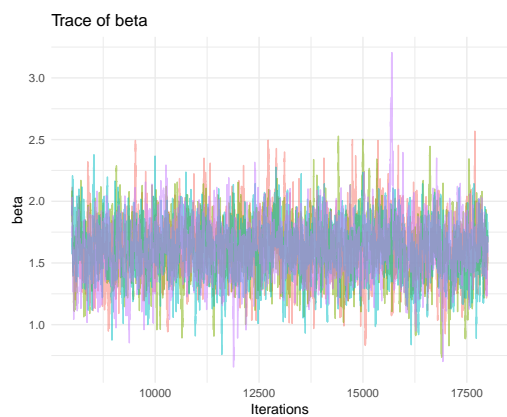
ggplot(d.chains, aes(x=p1.star, y=..density..)) +
  geom_density(color="darkblue", fill="lightblue", alpha=0.5) +
  labs(title="Density of p1.star", y="Density") + theme_minimal()

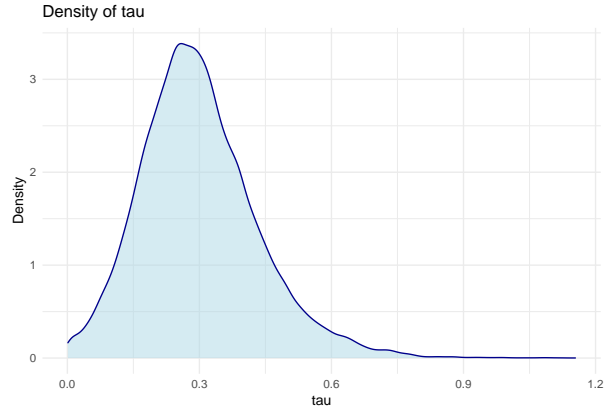
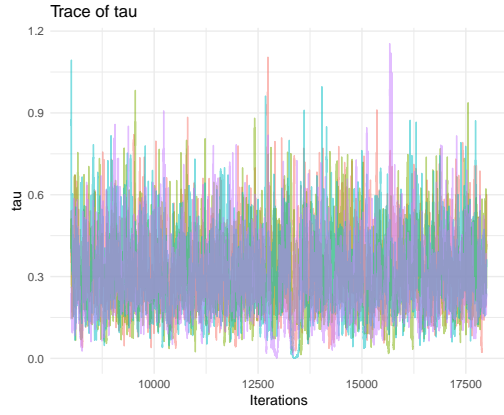
ggplot(d.chains, aes(x=iterations, y=p2.star, color=chains)) + geom_line(alpha=0.5) +
  labs(title="Trace of p2.star", x="Iterations") + theme_minimal()

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	1.6155	0.2203	1.1793	1.6157	2.0517
mu	-1.1101	0.1640	-1.4392	-1.1088	-0.7894
p1.star	0.2534	0.0676	0.1330	0.2480	0.4094
p2.star	0.6199	0.0810	0.4426	0.6226	0.7780
tau	0.2993	0.1330	0.0710	0.2868	0.6045

### 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)

Let  $Y|\lambda \sim P(\lambda)$  with  $\lambda \sim G(\alpha, \beta)$ . Use the expressions for iterated expectation

$$\mathbb{E}(Y) = \mathbb{E}_\lambda[\mathbb{E}_Y(Y | \lambda)]$$

and variance (Held and Sabanes Bove, 2020, Section A.3.4)

$$\text{Var}(Y) = \text{Var}_\lambda[\mathbb{E}_Y(Y | \lambda)] + \mathbb{E}_\lambda[\text{Var}_Y(Y | \lambda)]$$

To derive both, the expectation and the variance of the random variable  $Y$ .

Hints: Poisson distribution:  $X \sim \text{Po}(\lambda) : \mathbb{E}(X) = \lambda, \text{Var}(X) = \lambda$

Gamma distribution:  $X \sim \text{G}(\alpha, \beta) : \mathbb{E}(X) = \alpha/\beta, \text{Var}(X) = \alpha/\beta^2$

Solution:

$$\begin{aligned}\mathbb{E}(Y) &= \mathbb{E}_\lambda[\mathbb{E}_Y(Y | \lambda)] \quad \because Y | \lambda \sim \text{Po}(\lambda) \\ &= \mathbb{E}_\lambda(\lambda) \quad \because \lambda \sim \text{G}(\alpha, \beta) \\ &= \frac{\alpha}{\beta}\end{aligned}$$

$$\begin{aligned}\text{Var}(Y) &= \text{Var}_\lambda[\mathbb{E}_Y(Y | \lambda)] + \mathbb{E}_\lambda[\text{Var}_Y(Y | \lambda)] \quad \because Y | \lambda \sim \text{Po}(\lambda) \\ &= \text{Var}_\lambda(\lambda) + \mathbb{E}_\lambda(\lambda) \quad \because \lambda \sim \text{G}(\alpha, \beta) \\ &= \frac{\alpha}{\beta^2} + \frac{\alpha}{\beta} \\ &= \frac{\alpha(1 + \beta)}{\beta^2}\end{aligned}$$

## Exercise 6 (Empirical Bayes)

Consider observed numbers of lip cancer cases per district for each of 56 districts in Scotland:

```
y <- c(11, 5, 15, 9, 6, 9, 2, 3, 26, 39, 20, 31, 9, 16, 6, 16, 19, 17, 15, 11, 19,
7, 10, 0, 7, 7, 9, 2, 8, 8, 11, 6, 28, 4, 1, 1, 1, 8, 6, 3, 2, 1, 7, 10, 9, 11, 3,
11, 5, 8, 3, 7, 0, 8, 7, 13)
```

Assume that these observations are *i.i.d.* realizations of the model  $Y | \lambda \sim \text{Po}(\lambda)$  with  $\lambda \sim \text{G}(\alpha, \beta)$ . Apply and compare two different approaches to compute empirical Bayes estimates for each district:

- (a) Numerical maximization of the log-likelihood corresponding to the Poisson-gamma distribution as described by (Held and Sabanes Bove, 2020, p. 210) to obtain the marginal maximum likelihood estimator.

$$\underbrace{f(\lambda | y_{1:n})}_{\text{Posterior}} \propto \underbrace{f(y_{1:n} | \lambda)}_{\text{Likelihood}} \cdot \underbrace{f(\lambda)}_{\text{Prior}}$$

Likelihood:

$$f(y_{1:n} | \lambda) = \prod_{i=1}^n \frac{\lambda^{y_i} \exp(-\lambda)}{y_i!} \propto \lambda^{\sum_{i=1}^n y_i} \exp(-n\lambda)$$

Prior:

$$f(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda) \propto \lambda^{\alpha-1} \exp(-\beta\lambda)$$

Posterior:

$$\begin{aligned}f(\lambda | y_{1:n}) &\propto f(y_{1:n} | \lambda) \cdot f(\lambda) \\ &\propto \lambda^{\sum_{i=1}^n y_i} \exp(-n\lambda) \cdot \lambda^{\alpha-1} \exp(-\beta\lambda) \\ &= \lambda^{\sum_{i=1}^n y_i + \alpha - 1} \exp(-(n + \beta)\lambda)\end{aligned}$$

$$f(\lambda | y_{1:n}) \propto \lambda^{(\alpha + \sum_{i=1}^n y_i) - 1} \exp(-(\beta + n)\lambda)$$

Hence

$$\lambda | y_{1:n} \sim \text{G}\left(\alpha + \sum_{i=1}^n y_i, \beta + n\right)$$

In the empirical Bayes setting, we define the estimates of the prior based on the maximum likelihood estimates of the prior predictive distribution. This is also called the marginal likelihood and in our context has the Poisson-gamma form  $y_i \sim \text{PoG}(\alpha, \beta, 1)$  with the log-likelihood

Likelihood:

$$y_i \mid \lambda \sim \text{Po}(\lambda)$$

Prior:

$$\lambda \sim \text{G}(\alpha, \beta)$$

Prior predictive distribution:

$$\begin{aligned} f(y_i) &= \int_0^\infty f(y_i \mid \lambda) \cdot f(\lambda) d\lambda \\ &= \int_0^\infty \frac{\lambda^{y_i} \exp(-\lambda)}{y_i!} \cdot \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda) d\lambda \\ &= \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot \frac{1}{y_i!} \int_0^\infty \lambda^{y_i+\alpha-1} \exp(-(1+\beta)\lambda) d\lambda \\ &= \frac{\beta^\alpha}{(\beta+1)^{\alpha+y_i}} \cdot \frac{\Gamma(\alpha+y_i)}{\Gamma(\alpha)} \cdot \frac{1}{y_i!} \underbrace{\int_0^\infty \frac{(\beta+1)^{\alpha+y_i}}{\Gamma(\alpha+y_i)} \lambda^{(\alpha+y_i)-1} \exp(-(\beta+1)\lambda) d\lambda}_{\text{integrates to 1}} \\ &= \frac{\beta^\alpha}{(\beta+1)^{\alpha+y_i}} \cdot \frac{\Gamma(\alpha+y_i)}{\Gamma(\alpha)} \cdot \frac{1}{y_i!} \end{aligned}$$

Log-likelihood:

$$\begin{aligned} l(\alpha, \beta) &= \log \prod_{i=1}^n f(y_i) \\ &= \sum_{i=1}^n \log f(y_i) \\ &= \sum_{i=1}^n \log \left( \frac{\beta^\alpha}{(\beta+1)^{\alpha+y_i}} \cdot \frac{\Gamma(\alpha+y_i)}{\Gamma(\alpha)} \cdot \frac{1}{y_i!} \right) \\ &= \sum_{i=1}^n \left[ \alpha \log(\beta) - (\alpha+y_i) \log(\beta+1) + \log \left( \frac{\Gamma(\alpha+y_i)}{\Gamma(\alpha)} \right) - \log(y_i!) \right] \\ &\propto \sum_{i=1}^n \left[ \alpha \log(\beta) + \log \left( \frac{\Gamma(\alpha+y_i)}{\Gamma(\alpha)} \right) - (\alpha+y_i) \log(\beta+1) \right] \end{aligned}$$

```
## Log-likelihood function
ll <- function(x) {
  alpha <- x[1]
  beta <- x[2]
  ll <- sum(alpha * log(beta) + log( (gamma(alpha + y))/(gamma(alpha)) ) -
            (alpha + y) * log(beta + 1))
  return(ll)
}

## Maximize log-likelihood function
opt <- optim(par=c(0.1, 0.1), fn=ll, control = list(fnscale = -1))

## Print the maximum likelihood estimates
opt$par
```

```
## [1] 1.8321593 0.1914292
```

Thus, we have  $\hat{\alpha}_{\text{ML}}$  and  $\hat{\beta}_{\text{ML}}$  and can put them into the posterior formula calculated above.

- (b) Matching of moments based on the Exercise 5 above, which provides the marginal moment estimator.

In the Exercise 5, we have derived:

$$\mathbb{E}(Y) = \frac{\alpha}{\beta}$$

$$\text{Var}(Y) = \frac{\alpha(1 + \beta)}{\beta^2}$$

Let us start with  $\text{Var}(Y)$ :

$$\text{Var}(Y) = \frac{\alpha(1 + \beta)}{\beta^2}$$

$$\text{Var}(Y) = \mathbb{E}(Y) \cdot \frac{1 + \beta}{\beta}$$

$$\frac{\text{Var}(Y)}{\mathbb{E}(Y)} = \frac{1}{\beta} + 1$$

$$\beta = \frac{1}{\frac{\text{Var}(Y)}{\mathbb{E}(Y)} - 1}$$

$$\mathbb{E}(Y) = \frac{\alpha}{\beta}$$

$$\alpha = \beta \mathbb{E}(Y)$$

$$= \frac{\mathbb{E}(Y)}{\frac{\text{Var}(Y)}{\mathbb{E}(Y)} - 1}$$

$$\begin{cases} \alpha &= \frac{\mathbb{E}(Y)}{\frac{\text{Var}(Y)}{\mathbb{E}(Y)} - 1} \\ \beta &= \frac{1}{\frac{\text{Var}(Y)}{\mathbb{E}(Y)} - 1} \end{cases}$$

```
## Moments-matching function
match.moments <- function(mean, var) {
  alpha <- mean / (var/mean - 1)
  beta <- 1 / (var/mean - 1)
  return(params = c(alpha=alpha, beta=beta))
}

params <- match.moments(mean = mean(y), var = var(y)); params

##      alpha      beta
## 1.7295501 0.1806993

alpha <- params[1]
beta <- params[2]
```

Compare means and the lengths of equi-tailed 95%CrI obtained by both approaches. Report your results

```
set.seed(34324)
M <- 100000
column.names <- c("District", "Mean", "Lower", "Median", "Upper", "Length", "Method")
results.eb <- data.frame(matrix(nrow=length(y), ncol=7))
results.mm <- data.frame(matrix(nrow=length(y), ncol=7))
colnames(results.eb) <- column.names
```

```

colnames(results.mm) <- column.names

for (i in 1:length(y)) {
  lambda.eb <- rgamma(n=100000, shape=opt$par[1]+y[i], rate=opt$par[2]+1)
  results.eb[i, 1] <- i
  results.eb[i, 2] <- mean(lambda.eb)
  results.eb[i, 3:5] <- quantile(lambda.eb, probs=c(0.025, 0.5, 0.975))
  results.eb[i, 6] <- results.eb[i, 5] - results.eb[i, 3]
  results.eb[i, 7] <- "Empirical Bayes"

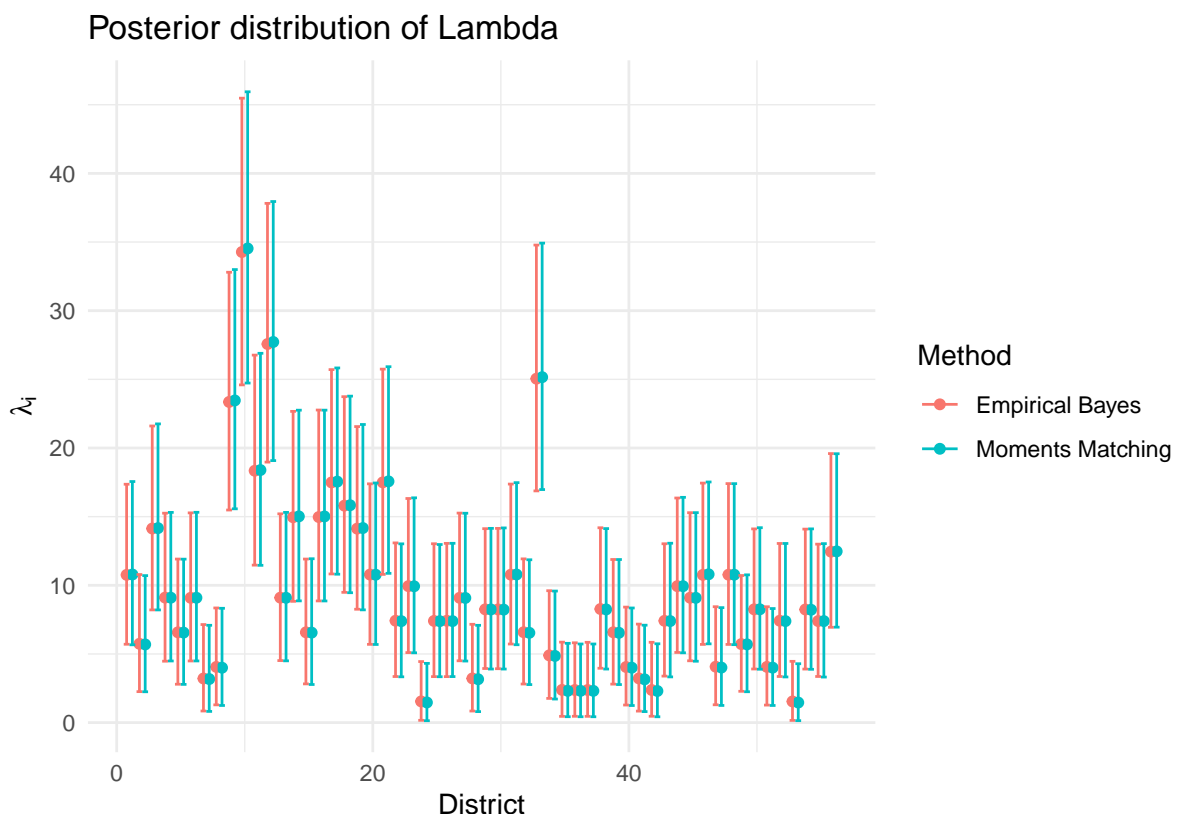
  lambda.mm <- rgamma(n=100000, shape=alpha+y[i], rate=beta+1)
  results.mm[i, 1] <- i
  results.mm[i, 2] <- mean(lambda.mm)
  results.mm[i, 3:5] <- quantile(lambda.mm, probs=c(0.025, 0.5, 0.975))
  results.mm[i, 6] <- results.mm[i, 5] - results.mm[i, 3]
  results.mm[i, 7] <- "Moments Matching"
}

```

```

d.plot <- rbind(results.eb, results.mm)
ggplot(data=d.plot, aes(x=District, y=Mean, col=Method)) +
  geom_point(position=position_dodge(.9)) +
  geom_errorbar(aes(ymin=Lower, ymax=Upper), position=position_dodge()) +
  labs(title = "Posterior distribution of Lambda",
       y=expression(lambda[i]), x="District") + theme_minimal()

```



```

mean.diff <- results.eb$Mean - results.mm$Mean
length.diff <- results.eb$Length - results.mm$Length

d.comparison <- rbind(
  c("Mean"=mean(mean.diff), quantile(mean.diff, probs=c(0.025,0.5,0.975))),

```

```

c("Mean"=mean(length.diff), quantile(length.diff, probs=c(0.025,0.5,0.975)))
)

rownames(d.comparison) <- c("Mean difference", "Length difference")
knitr::kable(d.comparison, align="c", digits=4, caption="Comparison of the two methods")

```

Table 4: Comparison of the two methods

	Mean	2.5%	50%	97.5%
Mean difference	0.0033	-0.1354	0.0136	0.0761
Length difference	-0.0349	-0.2208	-0.0332	0.1029

The Moment Matching method yields in general higher values and the width of the confidence intervals tends to be larger.