Worksheet 6

Foundations of Bayesian Methodology

Wenje Tu   Lea Bührer   Jerome Sepin   Zhixuan Li
Elia-Leonid Mastropietro   Jonas Raphael Füglistaler

Spring Semester 2022

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

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

$$y = \log(\mathrm{OR}) = \log \frac{x_\mathrm{P}}{n_\mathrm{P} - x_\mathrm{P}} - \log \frac{x_\mathrm{T}}{n_\mathrm{T} - x_\mathrm{T}}$$

$$\sigma = \mathrm{SE}\left(\log(\mathrm{OR})\right) = \sqrt{\frac{1}{x_\mathrm{P}} + \frac{1}{n_\mathrm{P} - x_\mathrm{P}} + \frac{1}{x_\mathrm{T}} + \frac{1}{n_\mathrm{T} - x_\mathrm{T}}}$$

```r
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)
```

```r
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:

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

for $i = 1, \cdots, k$

Random effects:

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

Priors:

$$\mu \sim \mathrm{N}(\nu, \gamma^2)$$
$$\tau \sim |\mathrm{N}(0, A^2)| = \mathrm{HN}(A)$$

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: 0x564795169668>
##
## 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
##
```

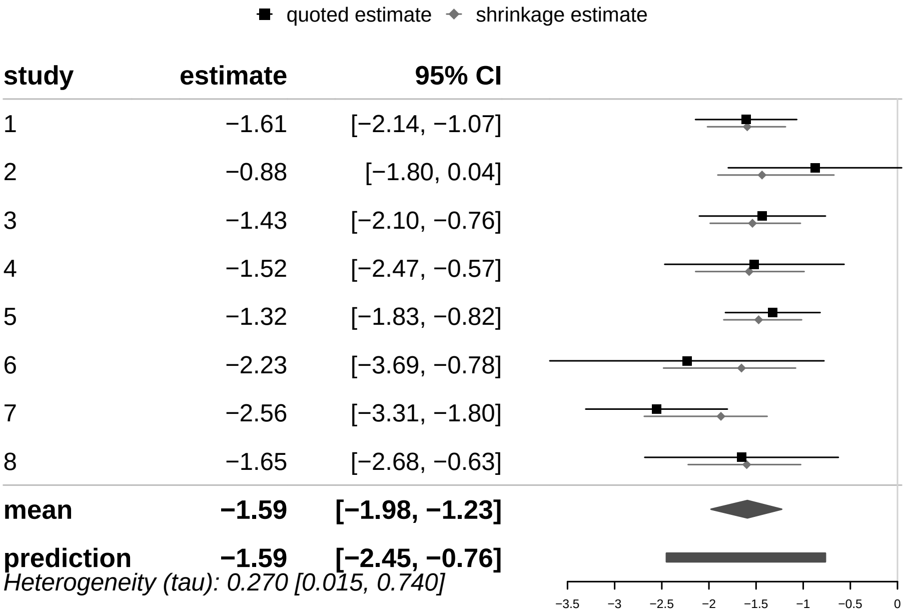
```
## relative heterogeneity I^2 (posterior median): 0.3343206
```

```
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

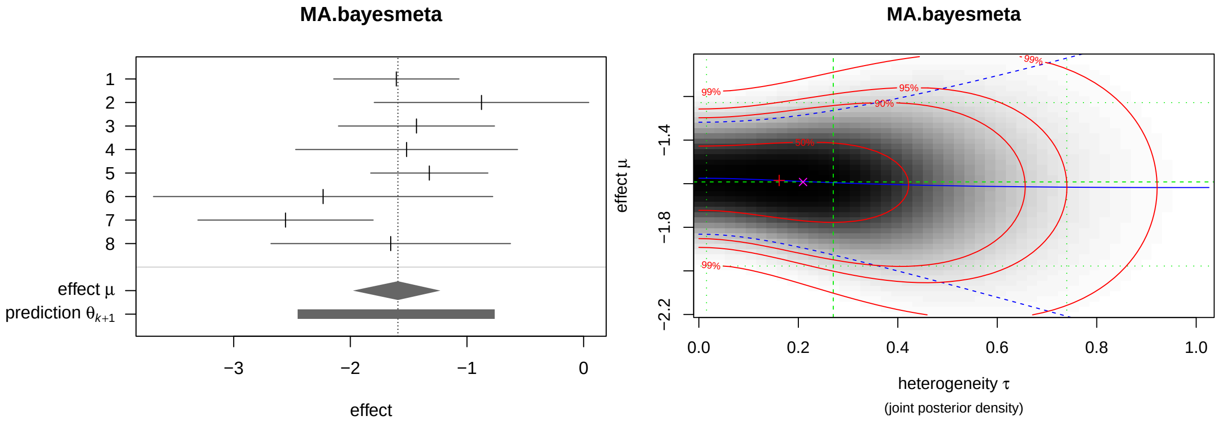
```
forestplot(MA.bayesmeta)
```

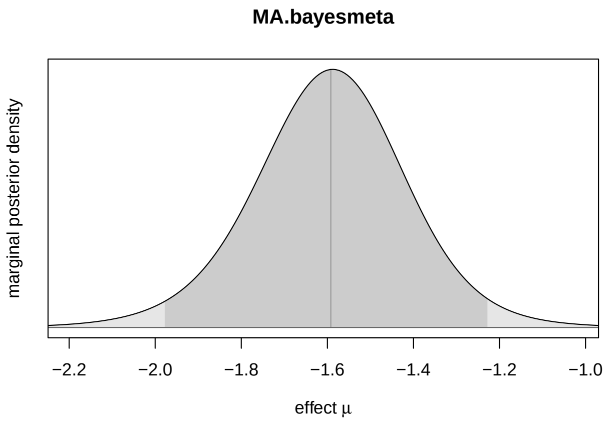


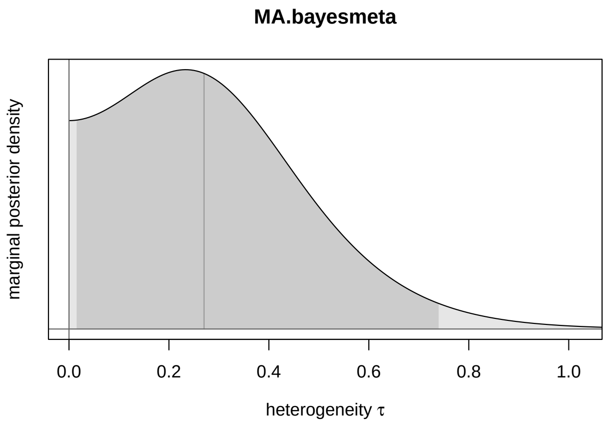
study	estimate	95% CI
1	−1.61	[−2.14, −1.07]
2	−0.88	[−1.80, 0.04]
3	−1.43	[−2.10, −0.76]
4	−1.52	[−2.47, −0.57]
5	−1.32	[−1.83, −0.82]
6	−2.23	[−3.69, −0.78]
7	−2.56	[−3.31, −1.80]
8	−1.65	[−2.68, −0.63]
mean	**−1.59**	**[−1.98, −1.23]**
prediction	**−1.59**	**[−2.45, −0.76]**

Heterogeneity (tau): 0.270 [0.015, 0.740]

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







Exercise 4 (Bayesian meta-analysis with JAGS)

Likelihood:

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

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

Priors:

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

```
pl1.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.)
)

pl1.params <- c("mu", "beta", "tau", "p1.star", "p2.star")
```

```
pl1_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
)
```

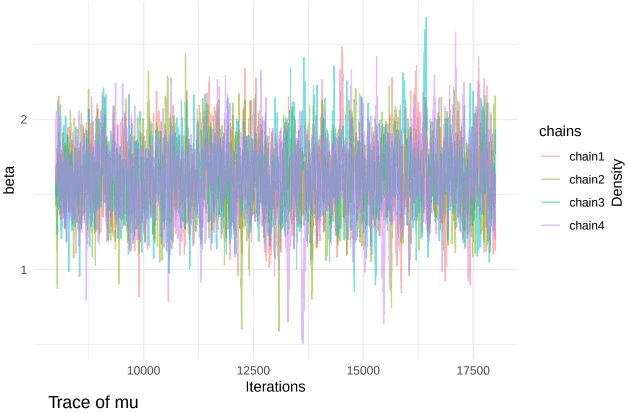
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 16
##    Unobserved stochastic nodes: 36
##    Total graph size: 222
##
## Initializing model
```

```
# 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)
```

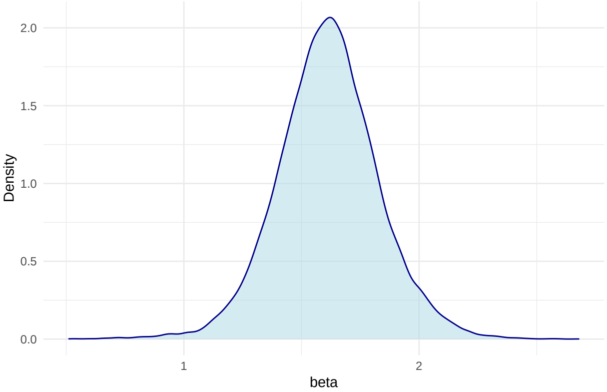
```
##
## Iterations = 8001:18000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 10000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean      SD  Naive SE Time-series SE
## beta     1.6118 0.21531 0.0010766      0.0054751
## mu      -1.1035 0.15800 0.0007900      0.0036498
## p1.star  0.2549 0.06708 0.0003354      0.0007737
## p2.star  0.6213 0.07931 0.0003965      0.0006785
## tau      0.2937 0.13328 0.0006664      0.0035281
##
## 2. Quantiles for each variable:
##
##             2.5%     25%     50%     75%   97.5%
## beta     1.18607  1.4784  1.6121  1.7447  2.0466
## mu      -1.41475 -1.2056 -1.1042 -1.0028 -0.7904
## p1.star  0.13808  0.2135  0.2489  0.2881  0.4119
## p2.star  0.44773  0.5785  0.6235  0.6685  0.7761
## tau      0.05062  0.2047  0.2828  0.3706  0.5942
```

```
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()
```

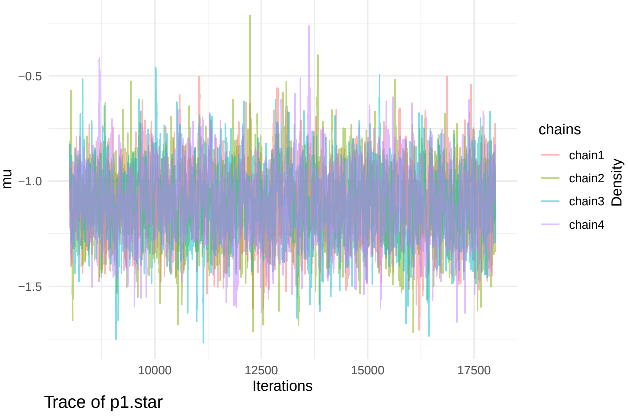
Trace of beta



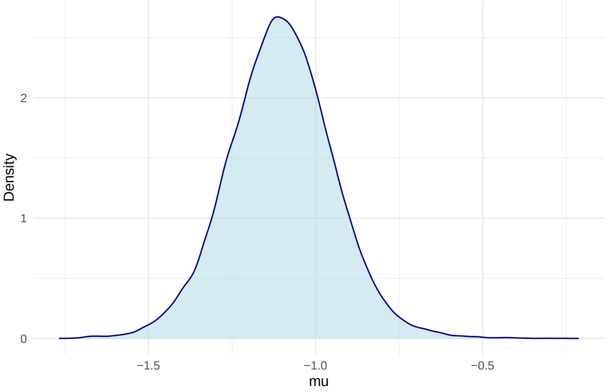
Density of beta



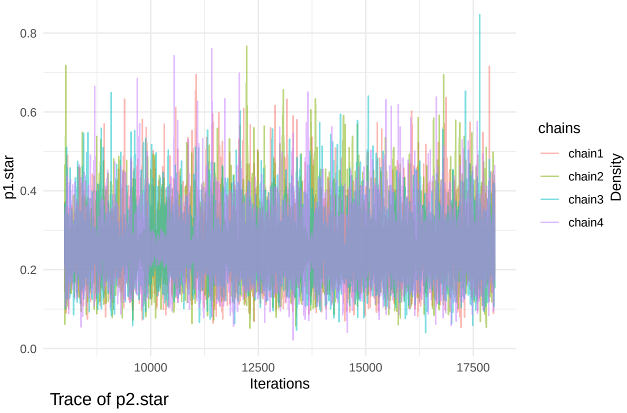
Trace of mu



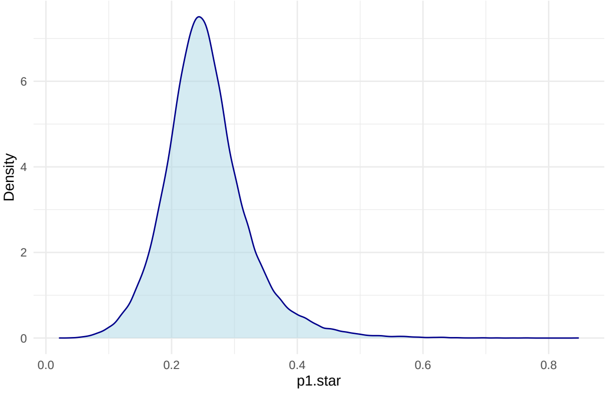
Density of mu



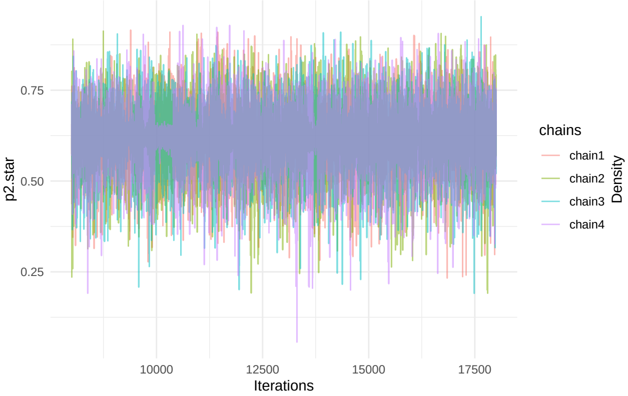
Trace of p1.star



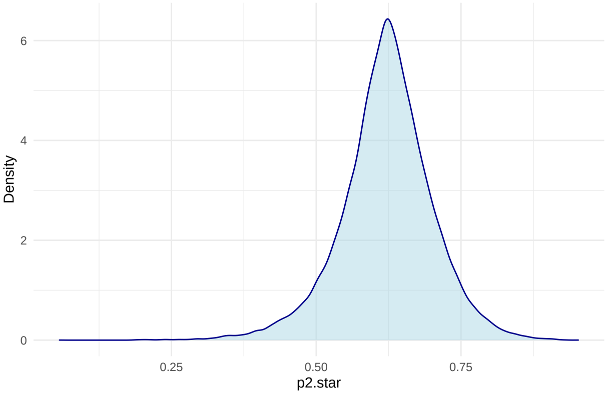
Density of p1.star

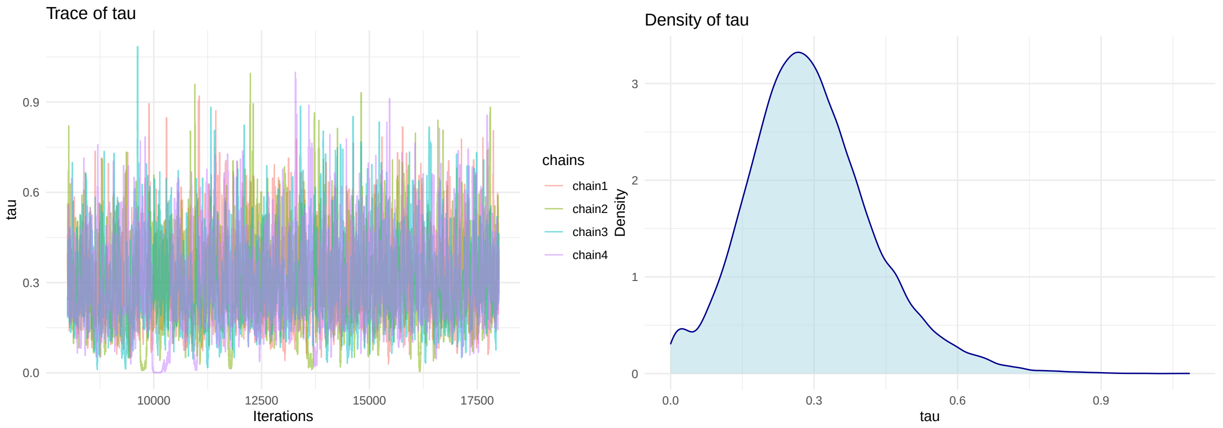


Trace of p2.star



Density of p2.star





```r
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	1.6118	0.2153	1.1861	1.6121	2.0466
mu	-1.1035	0.1580	-1.4148	-1.1042	-0.7904
p1.star	0.2549	0.0671	0.1381	0.2489	0.4119
p2.star	0.6213	0.0793	0.4477	0.6235	0.7761
tau	0.2937	0.1333	0.0506	0.2828	0.5942

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:

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

for $i = 1, \cdots, N$

Random effects:

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

Priors:

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

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.

$$y = \log(\mathrm{OR}) = \log \frac{x_\mathrm{P}}{n_\mathrm{P} - x_\mathrm{P}} - \log \frac{x_\mathrm{T}}{n_\mathrm{T} - x_\mathrm{T}}$$

$$\sigma = \mathrm{SE}\left(\log(\mathrm{OR})\right) = \sqrt{\frac{1}{x_\mathrm{P}} + \frac{1}{n_\mathrm{P} - x_\mathrm{P}} + \frac{1}{x_\mathrm{T}} + \frac{1}{n_\mathrm{T} - x_\mathrm{T}}}$$

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

Likelihood:

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

for $i = 1, \cdots, k$

Random effects:

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

Priors:

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

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 (η_j) to directly model the number of responders with only one predictor indicating whether in the treatment or not.

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

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

Likelihood:

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

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

Priors:

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

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

$$E(Y) = E_\lambda(E_Y(Y|\lambda))$$

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

$$Var(Y) = Var_\lambda(E_Y(Y|\lambda)) + E_\lambda(Var_Y(Y|\lambda))$$

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

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

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

Solution:

$$
\begin{aligned}
&\text{Expectation:} \\
&E(Y) = E_\lambda(E_Y(Y|\lambda)) && \text{Because } Y|\lambda \sim P(\lambda) \\
& = E_\lambda(\lambda) && \text{Because } \lambda \sim G(\alpha, \beta) \\
& = \frac{\alpha}{\beta}
\end{aligned}
$$

$$
\begin{aligned}
&\text{Variance:} \\
&Var(Y) = Var_\lambda(E_Y(Y|\lambda)) + E_\lambda(Var_Y(Y|\lambda)) && \text{Because } Y|\lambda \sim P(\lambda) \\
& = Var_\lambda(\lambda) + E_\lambda(\lambda) && \text{Because } \lambda \sim G(\alpha, \beta) \\
& = \frac{\alpha}{\beta^2} + \frac{\alpha}{\beta} = \frac{\alpha(1+\beta)}{\beta^2}
\end{aligned}
$$

Exercise 6 (Empirical Bayes)

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

```r
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.* realisations of the model $Y|\lambda \sim P(\lambda)$ with $\lambda \sim G(\alpha, \beta)$. Apply and compare two different approaches to compute empirical Bayes estimates for each district:

(a) Numerical maximisation 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.

$$
\begin{aligned}
&\text{we want} \\
f(\lambda|y_{1:n}) &\propto f(y_{1:n}|\lambda) f(\lambda) \\
&\text{Likelihood:} \\
f(y_{1:n}|\lambda) &= \prod_{i=1}^{n} \frac{\lambda^{x_i} exp(-\lambda)}{x_i!} \\
&\propto \lambda^{\sum_{i=1}^n x_i} exp(-n\lambda) \\
&\text{Prior:} \\
f(\lambda) &= \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} exp(-\beta\lambda) \\
&\propto \lambda^{\alpha-1} exp(-\beta\lambda) \\
&\text{Posterior:} \\
f(\lambda|y_{1:n}) &\propto \lambda^{\sum_{i=1}^n x_i} exp(-n\lambda) \lambda^{\alpha-1} exp(-\beta\lambda) \\
&\propto \lambda^{(\sum_{i=1}^n x_i + \alpha)-1} exp(-(\beta+n)\lambda) \\
f(\lambda|y_{1:n}) &\sim G(\sum_{i=1}^n x_i + \alpha, \beta + n)
\end{aligned}
$$

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 $y_i \sim PoG(\alpha, \beta)$ with the log-likelihood

$$l(\alpha,\beta)=\sum_{i=1}^{n}\left[\alpha log(\beta)+log\left(\frac{\Gamma(\alpha+x_i)}{\Gamma(\alpha)}\right)-(\alpha+x_i)log(\beta+\nu)\right]$$

ν is in our case 1.

```r
#implement log-likelihood
ll <- function(par, x){
  ll<-sum(dnbinom(x, size = par[1], prob =par[2]/(par[2]+1),log = T) )
  return(ll)
}
#optimizing
opt <- optim(par=c(0.1,0.1), ll, x = y, method = "BFGS",
             control = list(fnscale = -1),
             hessian = T)
opt$par
```

```
## [1] 1.8330830 0.1915152
```

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

```r
#posterior distribution of lambda
set.seed(34324)
result <- c()
for(i in 1:length(y)){
  lambda_post <- rgamma(n=100000, shape =opt$par[1]+y[i], rate = opt$par[2] )
  current_result <- c("Mean"=mean(lambda_post),
                      "CrI:2.5%"=quantile(lambda_post,c(0.025)),
                      "Median"=quantile(lambda_post,c(0.5)),
                      "CrI:97.5%"=quantile(lambda_post,c(0.975))
  )
  result <- rbind(result, current_result)
}
result <- data.frame(result)
rownames(result) <- NULL
colnames(result) <- c("Mean","Lower","Median" ,"Upper")
result$Method <- "EmpiricalBayes"
result$district <- 1:nrow(result)
result$width <-  result$Upper-result$Lower
```

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

Matching moments:

$$\begin{aligned}
Var(Y)&=\frac{\alpha(1+\beta)}{\beta^2}\rightarrow\alpha=\beta^2Var(Y)/(1+\beta)\\
E(Y)&=\frac{\alpha}{\beta}\rightarrow\alpha=\beta E(Y)\\
\beta^2Var(Y)/(1+\beta)&=\beta E(Y)\\
\frac{\beta}{1+\beta}&=\frac{E(Y)}{Var(Y)}\\
\beta&=\frac{E(Y)}{Var(Y)}+\frac{E(Y)}{Var(Y)}\beta\\
\beta\left(1-\frac{E(Y)}{Var(Y)}\right)&=\frac{E(Y)}{Var(Y)}\\
\beta&=\frac{\frac{E(Y)}{Var(Y)}}{1-\frac{E(Y)}{Var(Y)}}
\end{aligned}$$

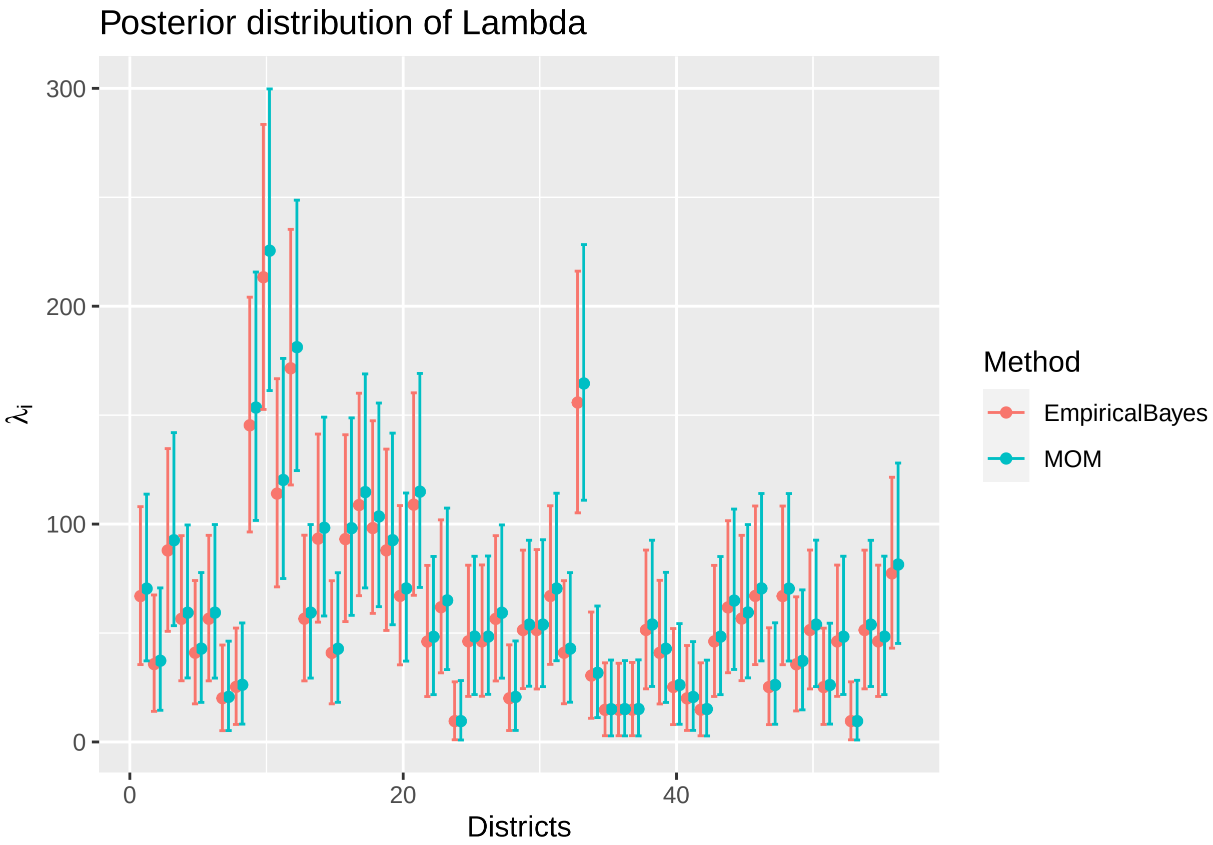
```
#apply formulas
var_y <- var(y)
ex_y <- mean(y)
beta <- (ex_y/var_y) /(1-(ex_y/var_y))
alpha <- beta*ex_y
print(c(alpha,beta))
```

```
## [1] 1.7295501 0.1806993
```

```
#posterior distribution of lambda
set.seed(34324)
result_MaMo <- c()
for(i in 1:length(y)){
  lambda_post <- rgamma(n=100000, shape =alpha+y[i], rate = beta )
  current_result <- c("Mean"=mean(lambda_post),
                      "CrI:2.5%"=quantile(lambda_post,c(0.025)),
                      "Median"=quantile(lambda_post,c(0.5)),
                      "CrI:97.5%"=quantile(lambda_post,c(0.975))
  )
  result_MaMo <- rbind(result_MaMo, current_result)
}
result_MaMo <- data.frame(result_MaMo)
rownames(result_MaMo) <- NULL
colnames(result_MaMo) <- c("Mean","Lower","Median" ,"Upper")
result_MaMo$Method <- "MOM"
result_MaMo$district <- 1:nrow(result_MaMo)
result_MaMo$width <-  result_MaMo$Upper-result_MaMo$Lower
```

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

```
results_comp <- rbind(result,result_MaMo )
ggplot(data = results_comp,aes(x=district, y=Mean, col=Method))+
  geom_point(position=position_dodge(.9))+
  geom_errorbar(aes(ymin=Lower, ymax=Upper), width=.9,
                position=position_dodge(.9))+
  labs(title = "Posterior distribution of Lambda",y = expression(lambda[i]),
     x = "Districts")
```



```r
# comparison of mean and width of the two methods
mean_diff <- result$Mean-result_MaMo$Mean
width_diff <- result$width-result_MaMo$width

mean_and_width <- rbind(
"Mean"=c("Mean "=mean(mean_diff),quantile(mean_diff,c(0.025,0.5,0.975))),
"Width"=c("Mean "=mean(width_diff),quantile(width_diff,c(0.025,0.5,0.975)))
)
```

Table 4: Comparison of the two methods.

	Mean	2.5%	50%	97.5%
Mean	-2.991674	-9.336656	-2.502364	-0.1129776
Width	-3.478733	-6.653901	-3.436985	-0.9754468

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