Exercise 3 (Bayesian meta-analysis with bayesmeta - 10 points)

Use the description of the package **bayesmeta** provided by Röver (2020) to compute a Bayesian meta-analysis of log(OR) of treatment and placebo based on 8 historical studies. . . . Explore and use plot functions provided by **bayesmeta** and report results obtained.

Solution:

```
remove(list=ls())

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 <- function(x_p, n_p, x_t, n_t) {
  #estimates the log of odds ratio
  ##x_p: number of respondents in Placebo group
  ##x_t: number of respondents in Treatment group
  ##n_p: total number of patients in Placebo group
  ##n_t: total number of patients in Treatment group

logOR <- log(x_p/(n_p-x_p)) - log(x_t/(n_t-x_t))

return(logOR)
}

se_log_OR <- function(x_p, n_p, x_t, n_t) {
  #estimates the SE of log of odds ratio
  ##x_p: number of respondents in Placebo group
  ##x_t: number of respondents in Treatment group
  ##n_p: total number of patients in Placebo group
  ##n_t: total number of patients in Treatment group

  selogOR <- sqrt(1/x_p + 1/(n_p - x_p) + 1/x_t + 1/(n_t - x_t))

  return(selogOR)
}
```

```
y <- log_OR(x_p = pl_case, n_p = pl_total, x_t = tr_case, n_t = tr_total)
sigma <- se_log_OR(x_p = pl_case, n_p = pl_total, x_t = tr_case, n_t = tr_total)
df <- data.frame(labels = 1:8, responders_t = tr_case, total_t = tr_total,
                 responders_p = pl_case, total_p = pl_total,
                 y = y, sigma = sigma)
```

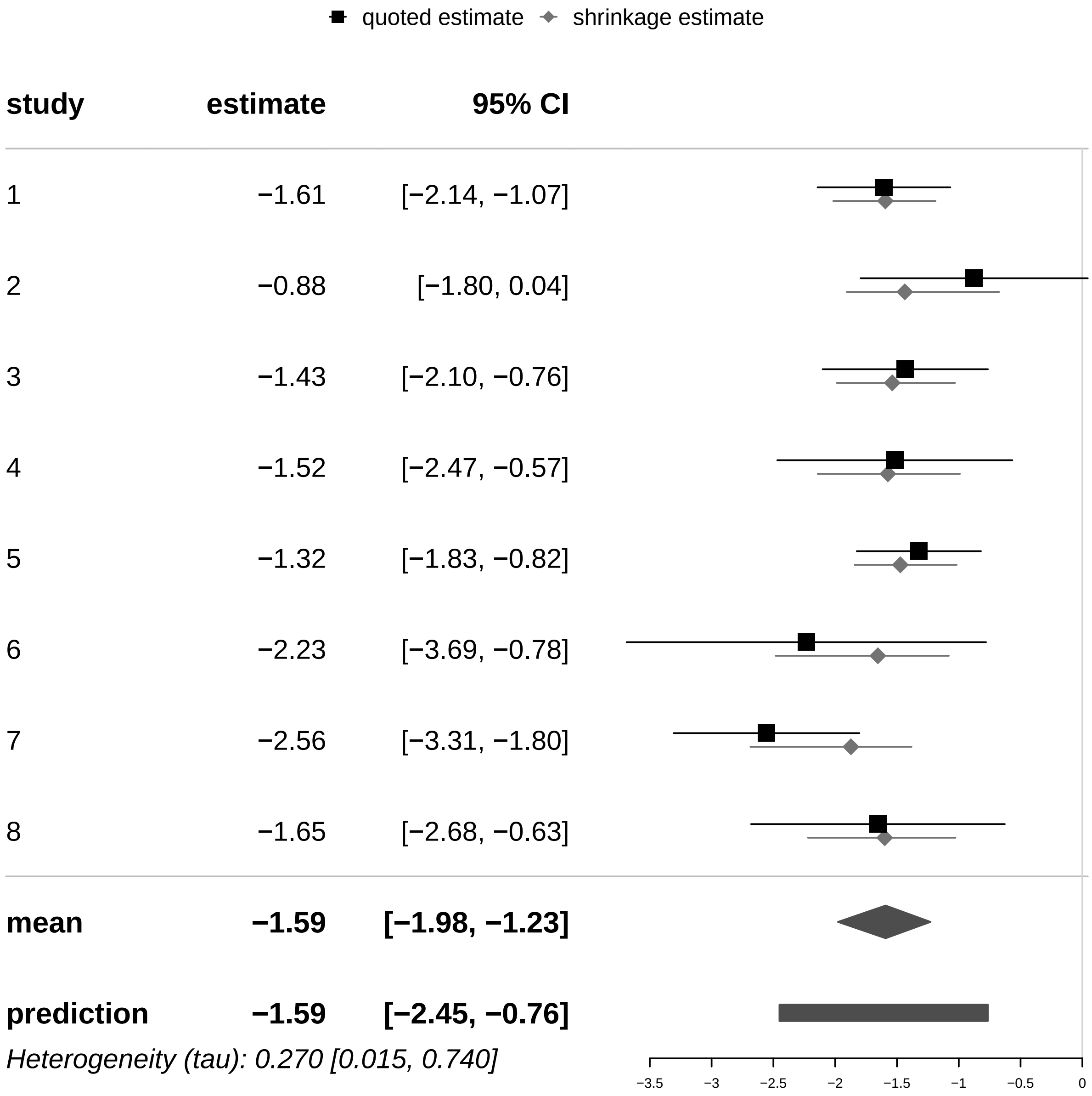
```
library(bayesmeta)
res <- bayesmeta(y = df[ , "y"], sigma = df[ , "sigma"],
                  labels = df[ , "labels"], mu.prior.mean = 0,
                  mu.prior.sd = 4,
                  tau.prior = function(t) {dhalfnormal(t, scale = 0.5)},
                  interval.type = "central")
summary(res)

##  '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: 0x000000001fe817d8>
##
## 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
```

Figure 1 shows the forest plot generated using the `forestplot()` function from `bayesmeta` package with default settings, showing the input data, effect estimate, prediction interval and shrinkage estimates.

Figure 2 shows four figures generated given the result obtained by `bayesmeta`. The top left plot is a simple forest plot showing estimates and 95% intervals. The top right plot illustrates the joint posterior density of heterogeneity τ and effect μ, with darker shading corresponding to higher probability density. The two bottom plots show the marginal posterior densities of effect μ and heterogeneity τ. 95% credible intervals are indicated with a darker shading, and the posterior median is shown by a vertical line.

```r
forestplot(res)
```



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]

Figure 1: Forest plot.

```r
par(mfrow = c(2,2))
plot(res)
```

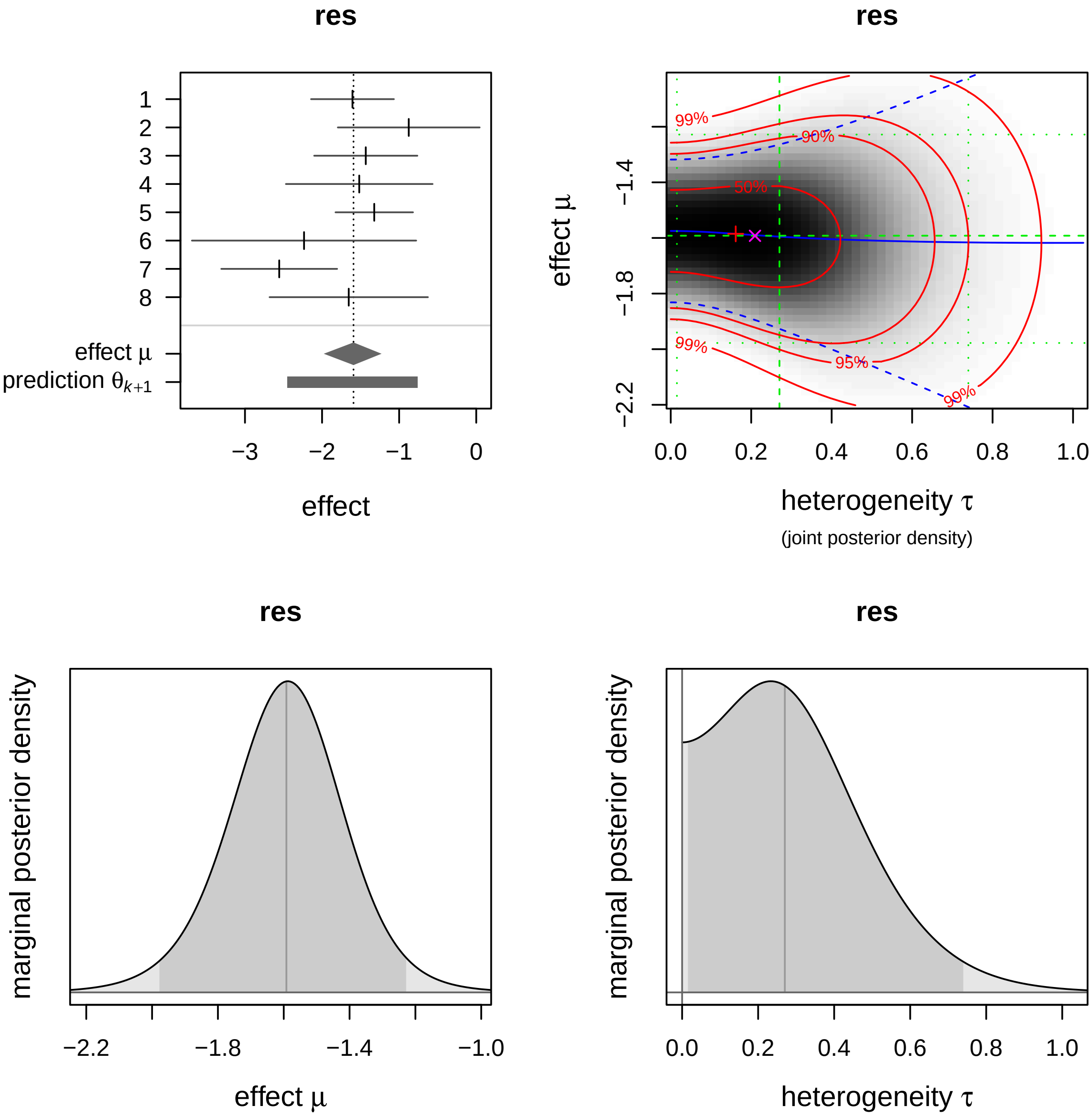


Figure 2: The four plots generated via the `plot()` function from the `bayesmeta` package.

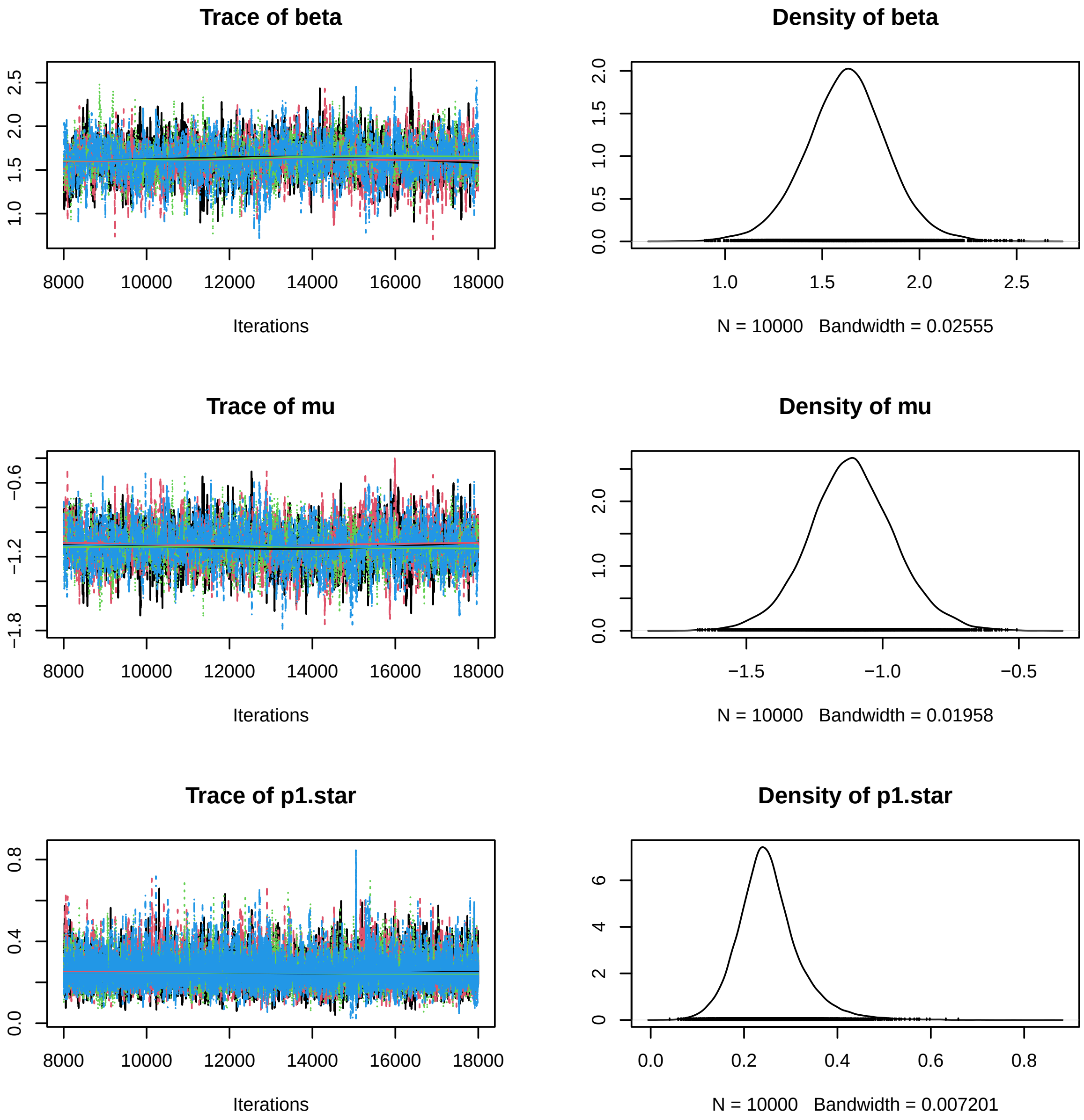
Exercise 4 (Bayesian meta-analysis with JAGS - 5 points)

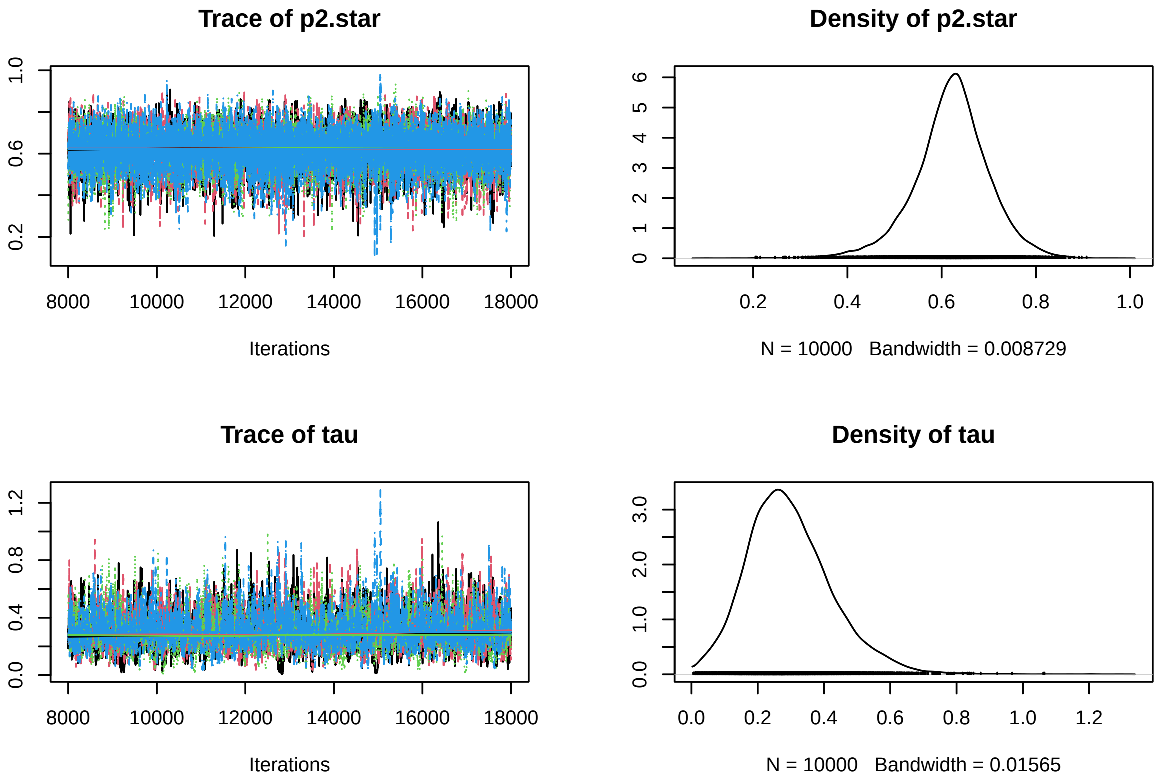
Run R code provided in the file `06worksheet_JAGSextension.R`. This model provides an alternative analysis of data considered in the Exercise 3 above. Discuss similarities and differences of the model provided in this R code and models that were used for Bayesian meta-analyses in the Exercise 3 above and in the individual project (Exercise 1 in Worksheet 5).

Solution:

```
#####################################
# STA421 FBM FS21: 06Worksheet Exercise 3
#####################################



library(rjags)
library(coda)

###############
# rjags interface with code in a model string
###############
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="TempModel.txt") # write to a file

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


# 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     1.6246 0.21280 0.0010640      0.0053224
## mu      -1.1126 0.16043 0.0008022      0.0037947
## p1.star  0.2527 0.06689 0.0003345      0.0007552
## p2.star  0.6214 0.07985 0.0003992      0.0006447
## tau      0.2964 0.12906 0.0006453      0.0030256
##
## 2. Quantiles for each variable:
##
##             2.5%     25%     50%     75%   97.5%
## beta     1.19437  1.4909  1.6268  1.7598  2.0461
## mu      -1.43131 -1.2163 -1.1149 -1.0102 -0.7834
## p1.star  0.13629  0.2108  0.2466  0.2865  0.4080
## p2.star  0.44578  0.5776  0.6246  0.6695  0.7759
## tau      0.07512  0.2069  0.2830  0.3716  0.5865

plot(fit.rjags.pl1.coda)
```


Trace of beta
1.0
1.5
2.0
2.5
8000
10000
12000
14000
16000
18000
Iterations
Density of beta
0.0
0.5
1.0
1.5
2.0
1.0
1.5
2.0
2.5
N = 10000   Bandwidth = 0.02555
Trace of mu
−1.8
−1.2
−0.6
8000
10000
12000
14000
16000
18000
Iterations
Density of mu
0.0
1.0
2.0
−1.5
−1.0
−0.5
N = 10000   Bandwidth = 0.01958
Trace of p1.star
0.0
0.4
0.8
8000
10000
12000
14000
16000
18000
Iterations
Density of p1.star
0
2
4
6
0.0
0.2
0.4
0.6
0.8
N = 10000   Bandwidth = 0.007201



In general, convergence diagnostics should be considered but we do not discuss them here. In this model the `p1.star` is the object that is used for elicitation of the β prior for placebo group in the individual project (Baeten et al. 2013). In the approach used for estimation by `bayesmeta` we first transform the raw data to normality $y_i = \log(\text{OR}_i)$, $\sigma_i = \text{SE}(\log(\text{OR}_i))$ and then assume NNHM. In the individual project (Exercise 1 in Worksheet 5) we performed a similar transformation but only for the placebo $y_i = \log(\text{odds}_i)$, $\sigma_i = \text{SE}(\log(\text{odds}_i))$. Moreover, in this exercise we model the raw data directly by assuming binomial distribution and a logit model is assumed.

Another difference is regarding the prior distribution of the heterogeneity, τ. The heterogeneity parameter describes the variance between the different studies. When the heterogeneity parameter is fixed to zero, it is assumed that the mean of the process generating all the studies is identical. Here, we use a $uniform(0, 10)$ distribution as a prior for heterogeneity, accepting that the means of included studies may not be identical. Moreover, a uniform distribution prior indicates that we have no knowledge regarding the differences between the studies. It is a weakly informative prior because it accepts all possible scenarios of heterogeneity with equal probabilities. In the exercise 3, we used a half normal prior with scale $= 0.5$ for the heterogeneity parameter. The prior used in exercise 3 is an informative prior because it suggests some scenarios of heterogeneity with greater probabilities. Nevertheless, it still allows for a reasonable amount of heterogeneity between studies.

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

Let $Y \mid \lambda \sim \mathrm{Po}(\lambda)$ with $\lambda \sim \mathrm{G}(\alpha, \beta)$. Use the expressions for iterated expectation

$$\mathrm{E}(Y) = \mathrm{E}_\lambda[\mathrm{E}_Y(Y \mid \lambda)]$$

and variance (Held and Sabanés Bové, 2020, Section A.3.4)

$$\mathrm{Var}(Y) = \mathrm{Var}_\lambda[\mathrm{E}_Y(Y \mid \lambda)] + \mathrm{E}_\lambda[\mathrm{Var}_Y(Y \mid \lambda)]$$

to derive both the expectation and the variance of the random variable Y.

Hints:
Poisson distribution: $X \sim \mathrm{Po}(\lambda)$: $\mathrm{E}(X) = \lambda$, $\mathrm{Var}(X) = \lambda$
Gamma distribution: $X \sim \mathrm{G}(\alpha, \beta)$: $\mathrm{E}(X) = \alpha/\beta$, $\mathrm{Var}(X) = \alpha/\beta^2$
Solution:

$$\mathrm{E}(Y) = \mathrm{E}_\lambda[\mathrm{E}_Y(Y|\lambda)] = \mathrm{E}_\lambda[\lambda] = \alpha/\beta$$

$$\mathrm{Var}(Y) = \mathrm{Var}_\lambda[\mathrm{E}_Y(Y|\lambda)] + \mathrm{E}_\lambda[\mathrm{Var}_Y(Y|\lambda)] = \mathrm{Var}_\lambda[\lambda] + \mathrm{E}_\lambda[\lambda] = \alpha/\beta^2 + \alpha/\beta = \frac{\alpha(1+\beta)}{\beta^2}$$

Exercise 6 (Empirical Bayes - 10 points)

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)
```

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

  Solution:

  ```
  nll <- function(alp, bet) {
      lll <- alp * log(bet) + log(gamma(alp + y)/gamma(alp)) -
        (alp + y)*log(bet + 1)
      nll <-  -sum(lll)

      return(nll)
  }
  est <- stats4::mle(minuslog=nll, start=list(alp = 0.1, bet = 0.2))
  summary_est <- summary(est)
  (alpha_ML <- summary_est@coef["alp","Estimate"])

  ## [1] 1.832898

  (beta_ML <- summary_est@coef["bet","Estimate"])

  ## [1] 0.1914985
  ```

  Another way of calculating the maximum likelihood estimates is using the function `optim`.

  ```
  ll <- function(pars, dd) {
    alp <- pars[1]
    bet <- pars[2]
      lll <- alp * log(bet) + log(gamma(alp + dd)/gamma(alp)) -
        (alp + dd)*log(bet + 1)
      ll <-  sum(lll)

      return(ll)
  ```

```
}
est1 <- optim(par = c(1.8,0.19), fn = ll, dd = y,
              control = list( fnscale = -1), hessian = TRUE)
(alpha_ML1 <- est1$par[1])

## [1] 1.83369

(beta_ML1 <- est1$par[2])

## [1] 0.1916102
```

`optim` and `mle` provide similar results. In the following we use the values 1.8329 and 0.1915.

The posterior turns out to be $\lambda_i \mid y_i \sim \mathrm{G}(\alpha + y_i, \beta + 1)$.

```
(post_mean_emprical_Bayes_ML <- (alpha_ML+y)/(beta_ML+1) )

##  [1] 10.770385  5.734710 14.127503  9.091827  6.573989  9.091827  3.216872
##  [8]  4.056151 23.359575 34.270205 18.323899 27.555971  9.091827 14.966782
## [15]  6.573989 14.966782 17.484620 15.806061 14.127503 10.770385 17.484620
## [22]  7.413268  9.931106  1.538313  7.413268  7.413268  9.091827  3.216872
## [29]  8.252548  8.252548 10.770385  6.573989 25.038133  4.895430  2.377593
## [36]  2.377593  2.377593  8.252548  6.573989  4.056151  3.216872  2.377593
## [43]  7.413268  9.931106  9.091827 10.770385  4.056151 10.770385  5.734710
## [50]  8.252548  4.056151  7.413268  1.538313  8.252548  7.413268 12.448944

(post_CrI_low_emprical_Bayes_ML <- qgamma(p = 0.025, shape = alpha_ML+y,
                                          rate = beta_ML+1) )

##  [1]  5.7076343  2.2744304  8.2047916  4.5101988  2.8076627  4.5101988
##  [7]  0.8444442  1.2847973 15.5012940 24.5748138 11.4605870 18.9485645
## [13]  4.5101988  8.8457289  2.8076627  8.8457289 10.7999750  9.4921869
## [19]  8.2047916  5.7076343 10.7999750  3.3600194  5.1037891  0.1614795
## [25]  3.3600194  3.3600194  4.5101988  0.8444442  3.9283267  3.9283267
## [31]  5.7076343  2.8076627 16.8726898  1.7647000  0.4598865  0.4598865
## [37]  0.4598865  3.9283267  2.8076627  1.2847973  0.8444442  0.4598865
## [43]  3.3600194  5.1037891  4.5101988  5.7076343  1.2847973  5.7076343
## [49]  2.2744304  3.9283267  1.2847973  3.3600194  0.1614795  3.9283267
## [55]  3.3600194  6.9415696

(post_CrI_upper_emprical_Bayes_ML <- qgamma(p = 0.975, shape = alpha_ML+y,
                                            rate = beta_ML+1) )
```

```
##  [1] 17.413830 10.767220 21.633129 15.252402 11.914896 15.252402  7.146478
##  [8]  8.391915 32.803547 45.552624 26.771741 37.749710 15.252402 22.671147
## [15] 11.914896 22.671147 25.753534 23.703601 21.633129 17.413830 25.753534
## [22] 13.042896 16.338319  4.430356 13.042896 13.042896 15.252402  7.146478
## [29] 14.154566 14.154566 17.413830 11.914896 34.789550  9.595190  5.839582
## [36]  5.839582  5.839582 14.154566 11.914896  8.391915  7.146478  5.839582
## [43] 13.042896 16.338319 15.252402 17.413830  8.391915 17.413830 10.767220
## [50] 14.154566  8.391915 13.042896  4.430356 14.154566 13.042896 19.538276
```

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

  Solution:

  By method of moments:

  $$\begin{aligned} \text{mean}(y) &= \frac{\alpha_{MM}}{\beta_{MM}} \\ \text{var}(y) &= \frac{\alpha_{MM}}{\beta_{MM}^2} + \frac{\alpha_{MM}}{\beta_{MM}}, \end{aligned}$$

  equivalently,

  $$\begin{aligned} \alpha_{MM} &= \frac{\text{mean}(y)^2}{\text{var}(y) - \text{mean}(y)} \\ \beta_{MM} &= \frac{\text{mean}(y)}{\text{var}(y) - \text{mean}(y)} \end{aligned}$$

  ```
  #find alpha and beta by method of moments
  (alpha_MM <- (mean(y)^2)/(var(y) - mean(y)))

  ## [1] 1.72955

  (beta_MM <- mean(y)/(var(y) - mean(y)))

  ## [1] 0.1806993
  ```

  Note that for this data set the values of α and β parameters obtained by ML and MM methods are similar.

  ```
  (post_mean_emprical_Bayes_MM <- (alpha_MM+y)/(beta_MM+1) )

  ##  [1] 10.781365  5.699631 14.169188  9.087454  6.546587  9.087454  3.158764
  ##  [8]  4.005720 23.485701 34.496126 18.403967 27.720480  9.087454 15.016144
  ```

```
## [15]  6.546587 15.016144 17.557011 15.863100 14.169188 10.781365 17.557011
## [22]  7.393542  9.934410  1.464852  7.393542  7.393542  9.087454  3.158764
## [29]  8.240498  8.240498 10.781365  6.546587 25.179613  4.852675  2.311808
## [36]  2.311808  2.311808  8.240498  6.546587  4.005720  3.158764  2.311808
## [43]  7.393542  9.934410  9.087454 10.781365  4.005720 10.781365  5.699631
## [50]  8.240498  4.005720  7.393542  1.464852  8.240498  7.393542 12.475277

(post_CrI_low_emprical_Bayes_MM <- qgamma(p = 0.025, shape = alpha_MM+y,
                                          rate = beta_MM+1) )

##  [1]  5.6964194  2.2408836  8.2133283  4.4901897  2.7767726  4.4901897
##  [7]  0.8090843  1.2485628 15.5718617 24.7255127 11.4963185 19.0493616
## [13]  4.4901897  8.8595245  2.7767726  8.8595245 10.8301051  9.5113410
## [19]  8.2133283  5.6964194 10.8301051  3.3323517  5.0880612  0.1388657
## [25]  3.3323517  3.3323517  4.4901897  0.8090843  3.9043162  3.9043162
## [31]  5.6964194  2.7767726 16.9552412  1.7292628  0.4282177  0.4282177
## [37]  0.4282177  3.9043162  2.7767726  1.2485628  0.8090843  0.4282177
## [43]  3.3323517  5.0880612  4.4901897  5.6964194  1.2485628  5.6964194
## [49]  2.2408836  3.9043162  1.2485628  3.3323517  0.1388657  3.9043162
## [55]  3.3323517  6.9399388

(post_CrI_upper_emprical_Bayes_MM <- qgamma(p = 0.975, shape = alpha_MM+y,
                                            rate = beta_MM+1) )

##  [1] 17.461385 10.744702 21.722397 15.277997 11.905168 15.277997  7.078868
##  [8]  8.340993 32.999720 45.868275 26.910612 37.992428 15.277997 22.770518
## [15] 11.905168 22.770518 25.882649 23.812974 21.722397 17.461385 25.882649
## [22] 13.045362 16.375013  4.314686 13.045362 13.045362 15.277997  7.078868
## [29] 14.168775 14.168775 17.461385 11.905168 35.004450  9.559065  5.751860
## [36]  5.751860  5.751860 14.168775 11.905168  8.340993  7.078868  5.751860
## [43] 13.045362 16.375013 15.277997 17.461385  8.340993 17.461385 10.744702
## [50] 14.168775  8.340993 13.045362  4.314686 14.168775 13.045362 19.606984
```

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

  Solution:

  The difference between the summary statistics of ML and MM methods is almost zero.

```
summary(post_mean_emprical_Bayes_ML -
                     post_mean_emprical_Bayes_MM)

##        Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -2.259e-01 -1.098e-02  1.205e-02 -1.352e-05  3.700e-02  7.346e-02
```

```
quantile(post_mean_emprical_Bayes_ML - post_mean_emprical_Bayes_MM,
         probs = c(0.025, 0.975))

##        2.5%       97.5% 
## -0.15587285  0.07058239 

length_ML <- post_CrI_upper_emprical_Bayes_ML - post_CrI_low_emprical_Bayes_ML
length_MM <- post_CrI_upper_emprical_Bayes_MM - post_CrI_low_emprical_Bayes_MM
summary(length_ML - length_MM)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -0.16495 -0.05877 -0.03822 -0.03389 -0.00810  0.09306 

quantile(length_ML - length_MM, probs = c(0.025, 0.975))

##        2.5%       97.5% 
## -0.13833172  0.07918047 
```

Figure 3 shows that maximum likelihood and method of moments provide very similar estimates.

```r
x <- 1:56
y1 <- post_mean_emprical_Bayes_ML
y2 <- post_mean_emprical_Bayes_MM
plot(x, y1, axes=FALSE, xlab="random effect", ylab="equi-tailed 95% CrI",
     col = "black", ylim = c(0,47))
points(x+0.5, y2, col = "green3")
axis(1, 1:56)
axis(2, 1:60)
# Vertical arrow
arrows(x0=x, y0=post_CrI_low_emprical_Bayes_ML, x1=x, angle=90, length = 0.05,
       y1=post_CrI_upper_emprical_Bayes_ML, code=3, col="black")

arrows(x0=x+0.5, y0=post_CrI_low_emprical_Bayes_MM, x1=x+0.5, angle=90,
       length = 0.05,
       y1=post_CrI_upper_emprical_Bayes_MM, code=3, col="green3")
box()
```

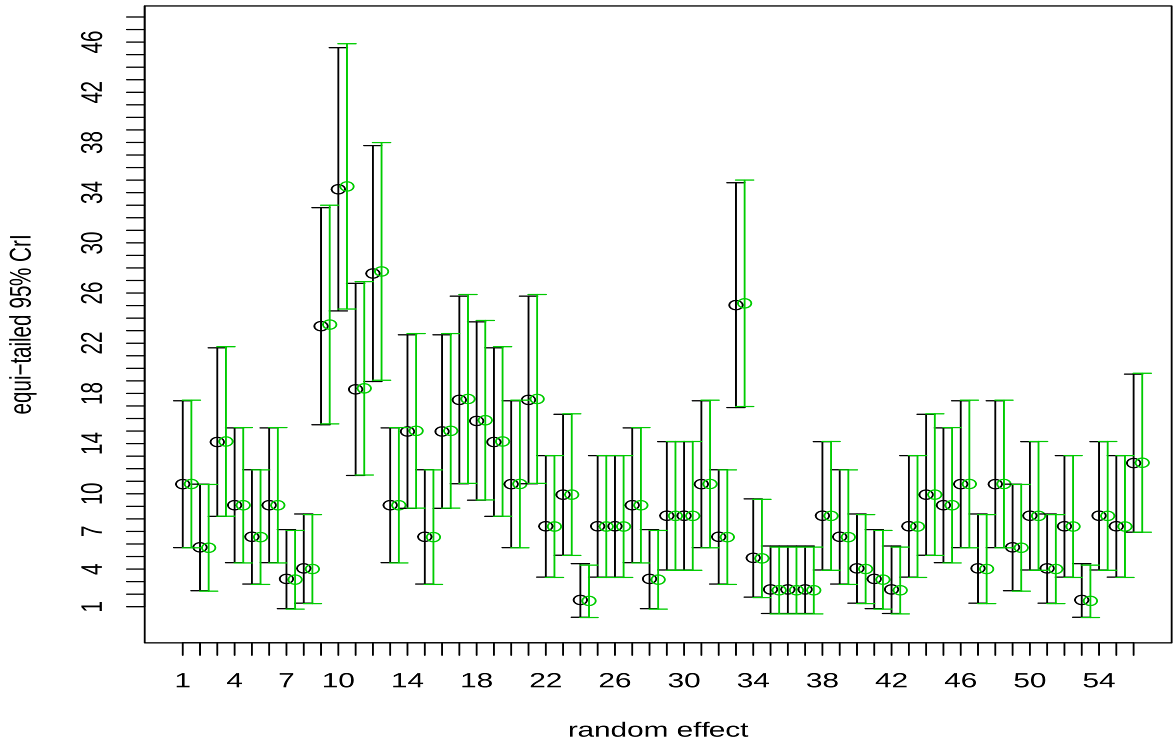


Figure 3: Posterior equi-tailed 95% credible intervals of random effect parameters provided by the by the empirical Bayes estimates with ML (black) and with method of moments (green).