TEhierarchical

ARH

2025-02-12

Meta-analysis: Hierarchical

Let's assume that the are $j = 1, \dots, J$ treatment effects from different studies such that the treatment effect

$$TE_j|\theta_j \sim N(\theta_j, \sigma_j^2),$$

where σ_j^2 is known, which is a sensible assumption given the sample sizes that are used in studies to estimate treatment effects, and

$$\theta_j \sim N(\mu, \tau),$$
$$\mu \propto c_1,$$
$$\tau \propto c_2.$$

The posterior distribution

$$\theta_j|Data \sim N(\hat{\theta}_j, \sigma^2_{\theta_j}),$$

where $\hat{\theta}_j = \frac{\frac{1}{\sigma_j^2}TE_j + \frac{1}{\tau^2}\mu}{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}}$ and $\sigma^2_{\theta_j} = \frac{1}{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}}$.

Note that $\frac{\frac{1}{\sigma_j^2}}{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}} = \frac{\tau^2}{\tau^2+\sigma_j^2} := w_j$, $0 < w_j < 1$. Consequently, $\hat{\theta}_j = w_j TE_j + (1-w_j)\mu$. Thus, if $\tau \to 0$, then $w_j \to 0$, and the posterior mean of $\theta_j \to \mu\ \forall j$, which means that there is no heterogeneity in the population treatment effects, and just sampling variability. This would suggest that the treatment effect estimates came from the same population. On the other extreme, $\tau \to \infty$, implies $w_j \to 1$, and consequently, $\theta_j \to TE_j$, which means that there is not shrinkage to μ.

The posterior distribution

$$\mu \sim N(\hat{\mu}, \sigma_\mu^2),$$

where $\hat{\mu} = \frac{\sum_{j=1}^J \left(\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}\right) TE_j}{\sum_{j=1}^J \frac{1}{\sigma_j^2}+\frac{1}{\tau^2}} = \sum_{j=1}^J \omega_j TE_j$, where $\omega_j = \frac{\frac{1}{\sigma_j^2}+\frac{1}{\tau^2}}{\sum_{j=1}^J \frac{1}{\sigma_j^2}+\frac{1}{\tau^2}} = \frac{1/\sigma^2_{\theta_j}}{\sum_{j=1}^J 1/\sigma^2_{\theta_j}} \propto \frac{1}{\sigma^2_{\theta_j}} = \frac{\tau^2+\sigma_j^2}{\tau^2\sigma_j^2} = \omega_j^*$, $0 < \omega_j < 1$ and $\sum_{j=1}^J \omega_j = 1$. Note that $\frac{d\omega_j^*}{d\sigma_j^2} = -\frac{1}{\sigma_j^4} < 0$, that is, less precise estimates have less weight to estimate $\hat{\mu}$, and given that $\frac{1}{\sigma_\mu^2} = \sum_{j=1}^J \frac{1}{\sigma_j^2} + \frac{1}{\tau^2} = \sum_{j=1}^J 1/\sigma^2_{\theta_j}$, there is less weight to estimates that contribute the most to the posterior variance of μ.

The posterior distribution of τ is proportional to $\sigma_\mu \prod_{j=1}^J (\sigma_j^2+\tau^2)^{-1/2} \exp\left(\frac{-(TE_j-\hat{\mu})^2}{2(\sigma_j^2+\tau^2)}\right)$. This is no a standard form, but we can simulate easily given a sensible grid for τ (see the code below).

```r
# Bayesian hierarchical models for treatment effects
data <- read.csv("byCountry_testscores.csv", header = TRUE)
attach(data)
TE <- X_beta # c(28, 8, -3, 7, -1, 1, 18, 12)
summary(TE)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.03993  0.00243  0.00875  0.01107  0.02189  0.06387
```

```r
StEr <- X_stderr # c(15, 10, 16, 11, 9, 11, 10, 18)
summary(StEr)
```

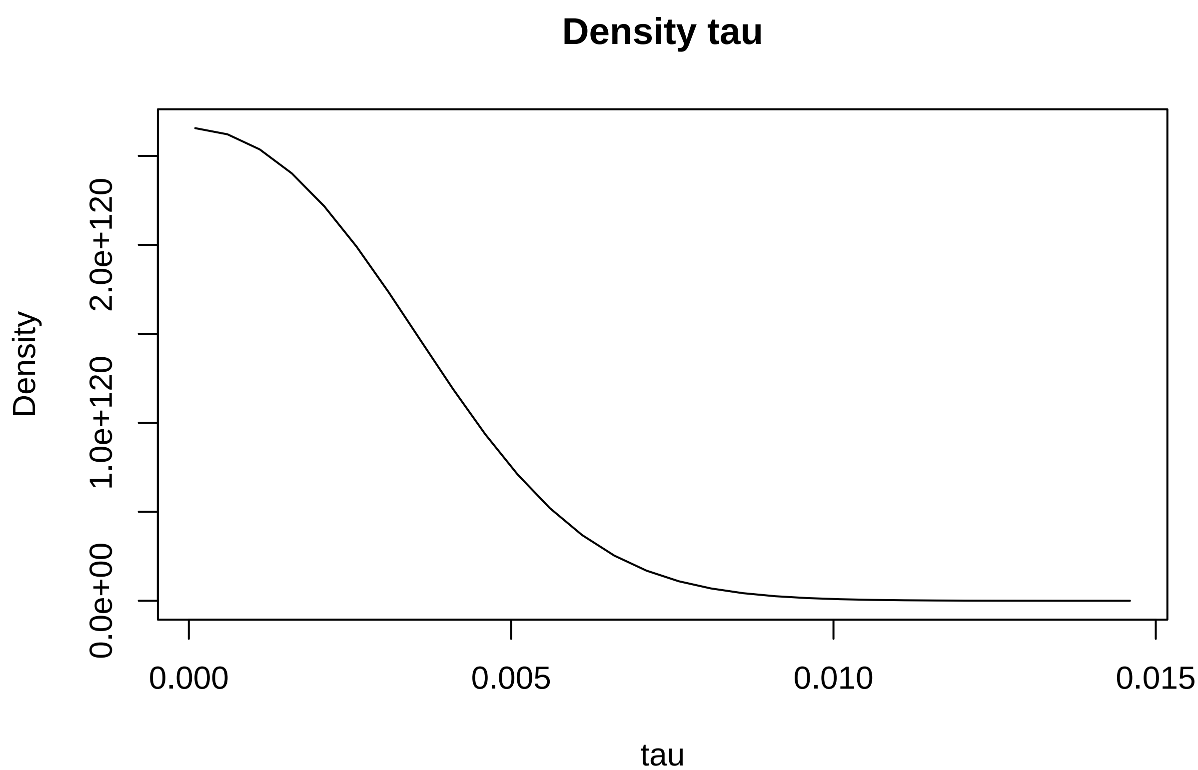
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00513 0.01164 0.01855 0.02294 0.02630 0.08538
```

```r
J <-length(TE)

# Posterior tau
PostTau <- function(tau){
  Vmui <- sum((StEr^2+tau^2)^-1)
  muhat <- sum((1/(StEr^2+tau^2))*TE)/sum((1/(StEr^2+tau^2)))
  term <- NULL
  for(j in 1:J){
    termj <- (StEr[j]^2+tau^2)^(-1/2)*exp(-(TE[j]-muhat)^2/(2*(StEr[j]^2+tau^2)))
    term <- c(term, termj)
  }
  denstau <- Vmui^(-0.5)*prod(term)
  return(denstau)
}
tau0 <- seq(0.0001,0.015,0.0005)
ProbTau <- sapply(tau0, PostTau)
plot(tau0, ProbTau, type = "l", main = "Density tau", ylab = "Density", xlab = "tau")


PostMu <- function(tau){
  Vmui <- sum((StEr^2+tau^2)^-1)
  muhat <- sum((1/(StEr^2+tau^2))*TE)/sum((1/(StEr^2+tau^2)))
  mu <- rnorm(1, muhat, (1/Vmui)^0.5)
  return(mu)
}
tau <- 0.005
mu <- PostMu(tau)

PostThetaj <- function(mu, tau, j){
  thetahatj <- (TE[j]/StEr[j]^2+mu/tau^2)/(1/StEr[j]^2+1/tau^2)
  Vj <- 1/(1/StEr[j]^2+1/tau^2)
  thetaj <- rnorm(1, thetahatj, Vj^0.5)
  return(thetaj)
}
PostThetaj(mu, tau, 7)
```

```
## [1] 0.02182585
```

```r
# Gibbs sampler
S <- 10000
THETA <- matrix(NA, S, J)
ProbTau <- sapply(tau0, PostTau)
tau <- sample(tau0, S, prob = ProbTau, replace = TRUE)
```

```r
mu <- sapply(tau, PostMu)
for(j in 1:J){
  thetaj <- sapply(1:S, function(s){PostThetaj(mu = mu[s], tau = tau[s], j = j)})
  THETA[,j] <- thetaj
}

summary(coda::mcmc(tau))
```




```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## 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
##      2.344e-03      1.831e-03      1.831e-05      1.831e-05
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## 0.0001 0.0011 0.0021 0.0036 0.0066
```

```r
summary(coda::mcmc(mu))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## 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
##      1.045e-02      1.556e-03      1.556e-05      1.556e-05
##
## 2. Quantiles for each variable:
##
##     2.5%      25%      50%      75%    97.5%
## 0.007348 0.009431 0.010444 0.011504 0.013445
```

```r
summary(coda::mcmc(THETA))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## 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
##  [1,] 0.010423 0.003215 3.215e-05      3.215e-05
##  [2,] 0.010502 0.003407 3.407e-05      3.507e-05
##  [3,] 0.010625 0.003308 3.308e-05      3.308e-05
##  [4,] 0.010432 0.003022 3.022e-05      3.022e-05
##  [5,] 0.010491 0.003323 3.323e-05      3.323e-05
##  [6,] 0.010477 0.003284 3.284e-05      3.357e-05
##  [7,] 0.010366 0.003273 3.273e-05      3.273e-05
##  [8,] 0.010579 0.003321 3.321e-05      3.274e-05
##  [9,] 0.009821 0.003150 3.150e-05      3.427e-05
## [10,] 0.010228 0.003291 3.291e-05      3.291e-05
## [11,] 0.009182 0.003120 3.120e-05      3.067e-05
## [12,] 0.010084 0.003157 3.157e-05      3.112e-05
## [13,] 0.011630 0.002841 2.841e-05      2.776e-05
## [14,] 0.010116 0.003262 3.262e-05      3.262e-05
## [15,] 0.009342 0.003419 3.419e-05      3.361e-05
## [16,] 0.010262 0.003260 3.260e-05      3.195e-05
## [17,] 0.010370 0.003234 3.234e-05      3.234e-05
## [18,] 0.010530 0.003143 3.143e-05      3.143e-05
## [19,] 0.010538 0.003322 3.322e-05      3.322e-05
## [20,] 0.010165 0.003096 3.096e-05      3.096e-05
## [21,] 0.009996 0.003281 3.281e-05      3.264e-05
```

```
## [22,] 0.010199 0.003297 3.297e-05      3.297e-05
## [23,] 0.010473 0.003279 3.279e-05      3.279e-05
## [24,] 0.010807 0.003274 3.274e-05      3.231e-05
## [25,] 0.010727 0.003292 3.292e-05      3.284e-05
## [26,] 0.010766 0.003294 3.294e-05      3.294e-05
## [27,] 0.010580 0.003224 3.224e-05      3.224e-05
## [28,] 0.011330 0.003053 3.053e-05      3.053e-05
## [29,] 0.011299 0.003253 3.253e-05      3.253e-05
## [30,] 0.010405 0.003341 3.341e-05      3.341e-05
## [31,] 0.010266 0.003093 3.093e-05      3.093e-05
## [32,] 0.010527 0.003351 3.351e-05      3.351e-05
## [33,] 0.011188 0.003399 3.399e-05      3.399e-05
## [34,] 0.010899 0.003239 3.239e-05      3.239e-05
## [35,] 0.010324 0.003363 3.363e-05      3.363e-05
## [36,] 0.011349 0.003113 3.113e-05      3.113e-05
## [37,] 0.010635 0.003324 3.324e-05      3.324e-05
## [38,] 0.010384 0.003352 3.352e-05      3.298e-05
## [39,] 0.011348 0.003258 3.258e-05      3.258e-05
## [40,] 0.010422 0.003406 3.406e-05      3.406e-05
## [41,] 0.010814 0.003208 3.208e-05      3.208e-05
## [42,] 0.010286 0.003336 3.336e-05      3.336e-05
## [43,] 0.010267 0.003297 3.297e-05      3.297e-05
## [44,] 0.011486 0.003215 3.215e-05      3.215e-05
## [45,] 0.010457 0.003245 3.245e-05      3.245e-05
## [46,] 0.009780 0.003446 3.446e-05      3.446e-05
## [47,] 0.010595 0.003324 3.324e-05      3.324e-05
## [48,] 0.010350 0.002801 2.801e-05      2.801e-05
## [49,] 0.010276 0.003257 3.257e-05      3.257e-05
## [50,] 0.010296 0.003233 3.233e-05      3.094e-05
## [51,] 0.010888 0.003034 3.034e-05      3.034e-05
## [52,] 0.009974 0.003193 3.193e-05      3.193e-05
## [53,] 0.009105 0.003578 3.578e-05      3.578e-05
## [54,] 0.010379 0.003218 3.218e-05      3.218e-05
## [55,] 0.010288 0.003218 3.218e-05      3.218e-05
## [56,] 0.010686 0.003274 3.274e-05      3.274e-05
## [57,] 0.010447 0.003200 3.200e-05      3.200e-05
## [58,] 0.010512 0.003389 3.389e-05      3.389e-05
## [59,] 0.011192 0.003437 3.437e-05      3.507e-05
## [60,] 0.010159 0.002832 2.832e-05      2.832e-05
## [61,] 0.010322 0.003249 3.249e-05      3.249e-05
## [62,] 0.010808 0.003240 3.240e-05      3.355e-05
## [63,] 0.010012 0.002607 2.607e-05      2.607e-05
## [64,] 0.010027 0.003314 3.314e-05      3.314e-05
## [65,] 0.009982 0.003146 3.146e-05      3.146e-05
## [66,] 0.010345 0.003313 3.313e-05      3.380e-05
## [67,] 0.011159 0.003383 3.383e-05      3.282e-05
## [68,] 0.009738 0.003075 3.075e-05      3.075e-05
## [69,] 0.010920 0.003215 3.215e-05      3.215e-05
## [70,] 0.009647 0.003298 3.298e-05      3.298e-05
## [71,] 0.010575 0.002877 2.877e-05      2.996e-05
## [72,] 0.010293 0.003330 3.330e-05      3.389e-05
## [73,] 0.010418 0.003260 3.260e-05      3.260e-05
## [74,] 0.010625 0.003266 3.266e-05      3.266e-05
## [75,] 0.010879 0.002800 2.800e-05      2.800e-05
```

```
## [76,] 0.010494 0.003296 3.296e-05      3.401e-05
## [77,] 0.010003 0.003056 3.056e-05      2.999e-05
## [78,] 0.010879 0.003319 3.319e-05      3.410e-05
## [79,] 0.009945 0.003353 3.353e-05      3.353e-05
##
## 2. Quantiles for each variable:
##
##            2.5%      25%      50%     75%   97.5%
## var1  0.0035904 0.008777 0.010438 0.01213 0.01710
## var2  0.0033003 0.008783 0.010483 0.01222 0.01768
## var3  0.0038451 0.008915 0.010564 0.01229 0.01755
## var4  0.0041894 0.008723 0.010412 0.01206 0.01688
## var5  0.0036492 0.008775 0.010453 0.01217 0.01754
## var6  0.0035195 0.008821 0.010479 0.01216 0.01729
## var7  0.0033558 0.008741 0.010420 0.01205 0.01712
## var8  0.0038003 0.008876 0.010540 0.01223 0.01770
## var9  0.0025620 0.008246 0.010097 0.01171 0.01538
## var10 0.0031602 0.008604 0.010330 0.01196 0.01675
## var11 0.0016320 0.007622 0.009626 0.01118 0.01425
## var12 0.0029091 0.008526 0.010241 0.01181 0.01631
## var13 0.0068026 0.009791 0.011290 0.01311 0.01844
## var14 0.0029386 0.008519 0.010263 0.01190 0.01645
## var15 0.0009689 0.007864 0.009811 0.01140 0.01487
## var16 0.0030450 0.008666 0.010383 0.01204 0.01657
## var17 0.0035161 0.008713 0.010425 0.01207 0.01695
## var18 0.0040781 0.008823 0.010493 0.01220 0.01707
## var19 0.0037354 0.008844 0.010529 0.01222 0.01744
## var20 0.0032738 0.008602 0.010306 0.01189 0.01632
## var21 0.0025732 0.008406 0.010243 0.01182 0.01603
## var22 0.0027696 0.008587 0.010299 0.01195 0.01651
## var23 0.0034394 0.008795 0.010506 0.01216 0.01731
## var24 0.0044230 0.009083 0.010671 0.01240 0.01814
## var25 0.0042156 0.008984 0.010605 0.01238 0.01789
## var26 0.0043155 0.008953 0.010656 0.01237 0.01815
## var27 0.0038999 0.008897 0.010542 0.01225 0.01730
## var28 0.0059858 0.009487 0.011014 0.01284 0.01858
## var29 0.0055699 0.009399 0.010951 0.01284 0.01912
## var30 0.0033057 0.008743 0.010458 0.01208 0.01710
## var31 0.0034229 0.008697 0.010342 0.01195 0.01654
## var32 0.0035744 0.008834 0.010507 0.01222 0.01774
## var33 0.0052132 0.009261 0.010868 0.01277 0.01936
## var34 0.0046862 0.009097 0.010713 0.01251 0.01808
## var35 0.0031682 0.008673 0.010394 0.01209 0.01695
## var36 0.0059251 0.009477 0.011020 0.01288 0.01875
## var37 0.0038000 0.008911 0.010535 0.01226 0.01783
## var38 0.0032078 0.008712 0.010444 0.01215 0.01708
## var39 0.0057639 0.009424 0.011021 0.01288 0.01917
## var40 0.0030920 0.008718 0.010450 0.01213 0.01761
## var41 0.0046267 0.009102 0.010659 0.01240 0.01787
## var42 0.0030557 0.008666 0.010368 0.01202 0.01710
## var43 0.0030033 0.008664 0.010347 0.01200 0.01695
## var44 0.0061258 0.009531 0.011101 0.01300 0.01930
## var45 0.0036646 0.008785 0.010479 0.01213 0.01722
## var46 0.0015094 0.008259 0.010043 0.01173 0.01593
```

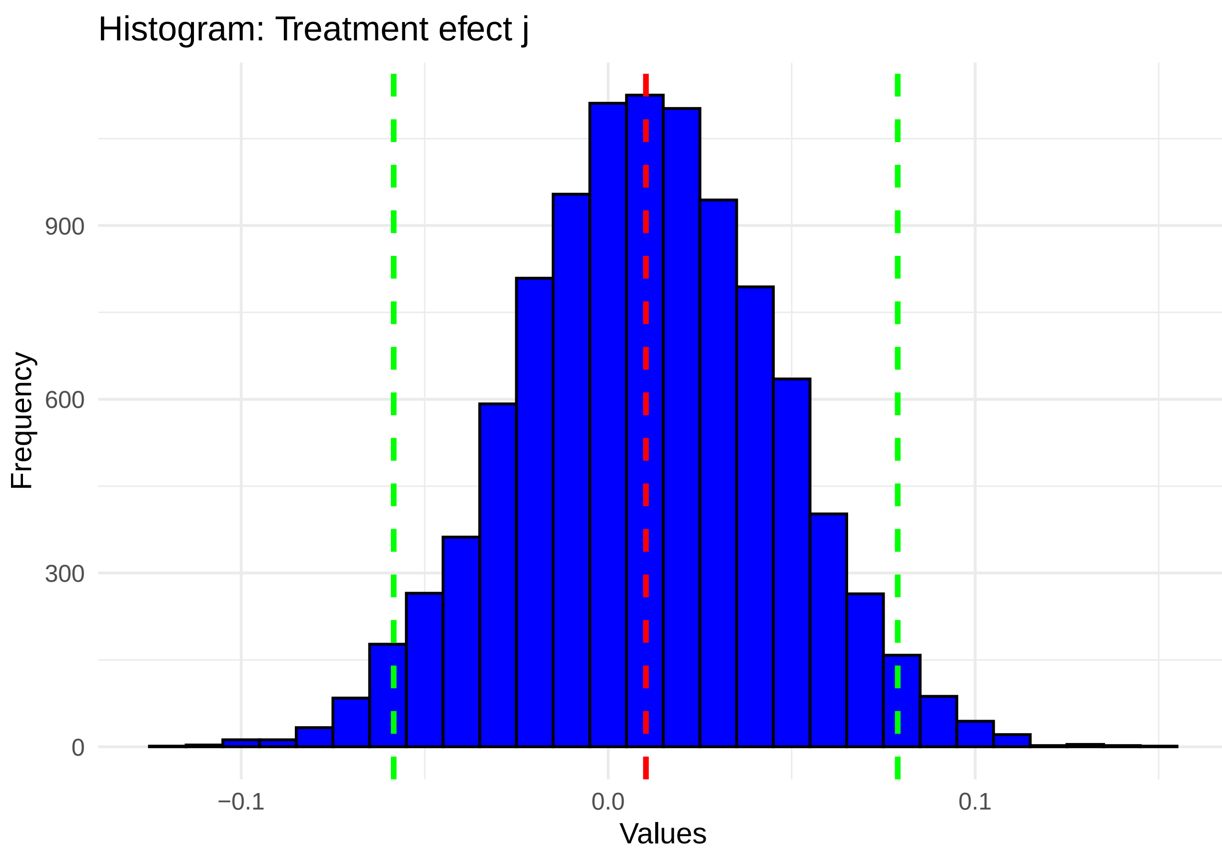
```
## var47 0.0038753 0.008831 0.010524 0.01228 0.01755
## var48 0.0046314 0.008764 0.010372 0.01196 0.01601
## var49 0.0030923 0.008661 0.010380 0.01201 0.01672
## var50 0.0032908 0.008624 0.010361 0.01203 0.01695
## var51 0.0050607 0.009151 0.010716 0.01244 0.01770
## var52 0.0027910 0.008367 0.010165 0.01177 0.01599
## var53 0.0000783 0.007634 0.009686 0.01128 0.01453
## var54 0.0033926 0.008742 0.010409 0.01207 0.01698
## var55 0.0034227 0.008704 0.010360 0.01199 0.01664
## var56 0.0041522 0.008957 0.010600 0.01228 0.01787
## var57 0.0037406 0.008757 0.010420 0.01211 0.01716
## var58 0.0034222 0.008811 0.010498 0.01221 0.01748
## var59 0.0049709 0.009282 0.010861 0.01273 0.01919
## var60 0.0041094 0.008610 0.010272 0.01186 0.01566
## var61 0.0032911 0.008707 0.010366 0.01205 0.01680
## var62 0.0044710 0.009082 0.010664 0.01239 0.01796
## var63 0.0042223 0.008559 0.010138 0.01164 0.01485
## var64 0.0024653 0.008534 0.010226 0.01184 0.01638
## var65 0.0028234 0.008433 0.010200 0.01178 0.01589
## var66 0.0032454 0.008685 0.010401 0.01208 0.01715
## var67 0.0051640 0.009272 0.010844 0.01267 0.01901
## var68 0.0024399 0.008235 0.009995 0.01160 0.01519
## var69 0.0048124 0.009115 0.010744 0.01246 0.01822
## var70 0.0016479 0.008100 0.009994 0.01161 0.01527
## var71 0.0047493 0.008945 0.010522 0.01217 0.01656
## var72 0.0029532 0.008651 0.010367 0.01201 0.01696
## var73 0.0033501 0.008767 0.010451 0.01212 0.01706
## var74 0.0039175 0.008919 0.010576 0.01227 0.01755
## var75 0.0055040 0.009225 0.010747 0.01239 0.01690
## var76 0.0035202 0.008838 0.010466 0.01212 0.01749
## var77 0.0029647 0.008423 0.010176 0.01177 0.01566
## var78 0.0046006 0.009078 0.010681 0.01245 0.01841
## var79 0.0020835 0.008391 0.010190 0.01180 0.01616
```

```r
# # Posterior distribution of thetaj given tau, and averaging over mu
# THETAtau <- matrix(NA, length(tau0), J)
# for (s in 1:length(tau0)){
#   mus <- replicate(20000, PostMu(tau0[s]))
#   for(j in 1:J){
#     thetajtau <- mean(sapply(1:S, function(l){PostThetaj(mu = mus[l], tau = tau0[s], j = j)}), na.rm
#     THETAtau[s,j] <- thetajtau
#   }
# }
#
# plot(tau0, THETAtau[,1], type = "l")

# Predictive distribution
j <- 3
TEj <- c(sapply(1:S, function(s) {rnorm(1, mean = THETA[s,j], sd = StEr[j])}))
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
dfj <- data.frame(TEj = TEj)
ggplot(dfj, aes(x = TEj)) +
  geom_histogram(binwidth = 0.01, fill = "blue", color = "black", ) +
  labs(title = "Histogram: Treatment efect j", x = "Values", y = "Frequency") +
  geom_vline(aes(xintercept = mean(TEj)), color = "red", linetype = "dashed", linewidth = 1) +
  geom_vline(aes(xintercept = quantile(TEj, 0.025)), color = "green", linetype = "dashed", linewidth = 
  geom_vline(aes(xintercept = quantile(TEj, 0.975)), color = "green", linetype = "dashed", linewidth = 
  theme_minimal()
```




```
summary(coda::mcmc(TEj))
```

```
##
## Iterations = 1:10000
## Thinning interval = 1
## Number of chains = 1
## 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
##      0.0102876      0.0348411      0.0003484      0.0003484
##
## 2. Quantiles for each variable:
##
```

```
##      2.5%      25%      50%      75%    97.5%
## -0.05844 -0.01323  0.01019  0.03416  0.07890
```