MSM_metaAnalysis_Gaussian_example.r

luiz

2019-06-24

```r
# Leo Bastos & Luiz Max Carvalho (2019)
# This example was taken from Malta et al. (2010)

source("pooling_aux.r")
```

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

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```r
meta <- read.csv("../data/meta_analysis_Malta_2010.csv")

meta$SampleSize
```

```
## [1]  658  461  621 1165  642  849
```

```r
K <- nrow(meta)
av <- meta$HIV + 1
bv <- meta$SampleSize - meta$HIV + 1

mv <- meta$HIV/meta$SampleSize
vv <- mv*(1-mv)/meta$SampleSize
sv <- sqrt(vv)

cbind(
  t(apply(cbind(av, bv), 1, stat_beta) ),
  t(apply(cbind(mv, sv), 1, stat_gauss) )
)
```

```
##              av                               mv
## [1,] 0.06818182 0.05024034 0.08859908 0.06686930 0.04778309 0.08595551
## [2,] 0.24190065 0.20402034 0.28189269 0.24078091 0.20175148 0.27981034
## [3,] 0.09951846 0.07727883 0.12419170 0.09822866 0.07482038 0.12163695
## [4,] 0.24164524 0.21752169 0.26660751 0.24120172 0.21663549 0.26576795
## [5,] 0.09006211 0.06920620 0.11332765 0.08878505 0.06678311 0.11078698
## [6,] 0.11750881 0.09675464 0.13996512 0.11660777 0.09501866 0.13819689
```

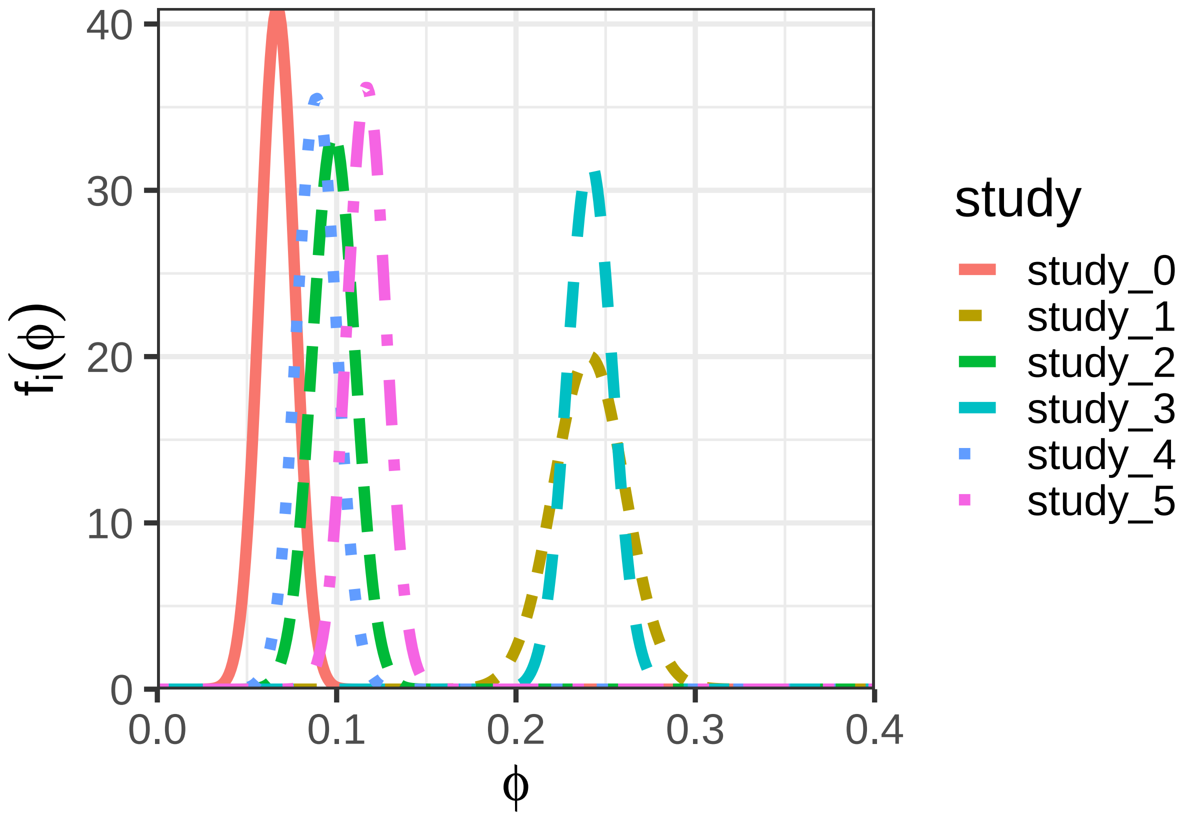
```r
# Individual entropies
entropies <- rep(NA, K)
for(k in 1:K) entropies[k] <- entropy_gauss(mv[k], vv[k])

entropies
```

```
## [1] -3.212777 -2.497427 -3.008653 -2.960370 -3.070612 -3.089554
```

```r
############
PaperMSMGauss.tbl <- data.frame(mean.prior = rep(NA, 6), lower.prior = NA,
                                upper.prior = NA)
```

```r
rownames(PaperMSMGauss.tbl) <- c("equal_weights", "maximum_entropy", "minimum_KL",
                                  "hierarchical_Dirichlet", "hierarchical_LogisticNormal", "Sample_size")

AlphasMSMGauss.tbl <- data.frame(matrix(NA, nrow = 3, ncol = length(av)))
rownames(AlphasMSMGauss.tbl) <- c("maximum_entropy", "minimum_KL", "Sample_size")
colnames(AlphasMSMGauss.tbl) <- paste("alpha_", 0:(K-1), sep = "")


library(ggplot2)

phi.grid <- seq(0, 1, length.out = 1000)
study.densities <- vector(K, mode = "list")
for(k in 1:K){
  study.densities[[k]] <- data.frame(phi = phi.grid,
                                     dens = dnorm(phi.grid, mean = mv[k], sd = sv[k] ),
                                     study = paste("study_", k-1, sep = ""))

}
study.densities.df <- do.call(rbind, study.densities)
study.densities.df$distribution <- "Gaussian"
write.csv(study.densities.df, file =  "../data/output/MSM_Gaussian_expert_densities.csv", row.names = FA

study_priors <- ggplot(study.densities.df, aes(x = phi, y = dens,
                                                linetype = study, colour = study)) +
  geom_line(size = 2) +
  scale_x_continuous(expression(phi), expand = c(0, 0), limits = c(0, .4)) +
  scale_y_continuous(expression(f[i](phi)), expand = c(0, 0)) +
  theme_bw(base_size = 20)

study_priors
```

```
## Warning: Removed 3600 rows containing missing values (geom_path).
```




```r
ggsave(study_priors, filename = "../plots/study_densities_MSMGaussian.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 3600 rows containing missing values (geom_path).
```

```r
###### Equal weights

alphaEqual <- rep(1/K, K)

ab.Equal.star <- pool_par_gauss(alphaEqual, mv, vv)
# Prior
(PaperMSMGauss.tbl[1, 1:3] <- stat_gauss(ab.Equal.star))
```

```
## [1] 0.12205302 0.09879808 0.14530795
```

```r
####### Maximum entropy

## WARNING: For the Gaussian case, we do not need to optimise, if you don't believe the maths, just run
# N <- 1000 ## could increase to, say, 10000 in order to make sure, but it's fine
# ent.many.startingPoints <- matrix(rnorm(n = (K-1)*N, mean = 0, sd = 100), ncol = K-1, nrow = N)
# many.ents <- lapply(1:N, function(i) {
#   optim(ent.many.startingPoints[i, ], optentgauss_inv, mp = mv, vp = vv)
# })
# optimised.ents <- unlist(lapply(many.ents, function(x) x$value))
#
# hist(optimised.ents)
# abline(v = optimised.ents[which.min(optimised.ents)], lty = 2, lwd = 2)
#
# alphaMaxEnt.opt <- alpha_01(many.ents[[which.min(optimised.ents)]]$par)
```

```r
## Maximum entropy "analytical" solution,
alphaMaxEnt.opt  <- rep(0, K)
alphaMaxEnt.opt[which.max(vv)] <- 1
round(alphaMaxEnt.opt, 2)
```

```
## [1] 0 1 0 0 0 0
```

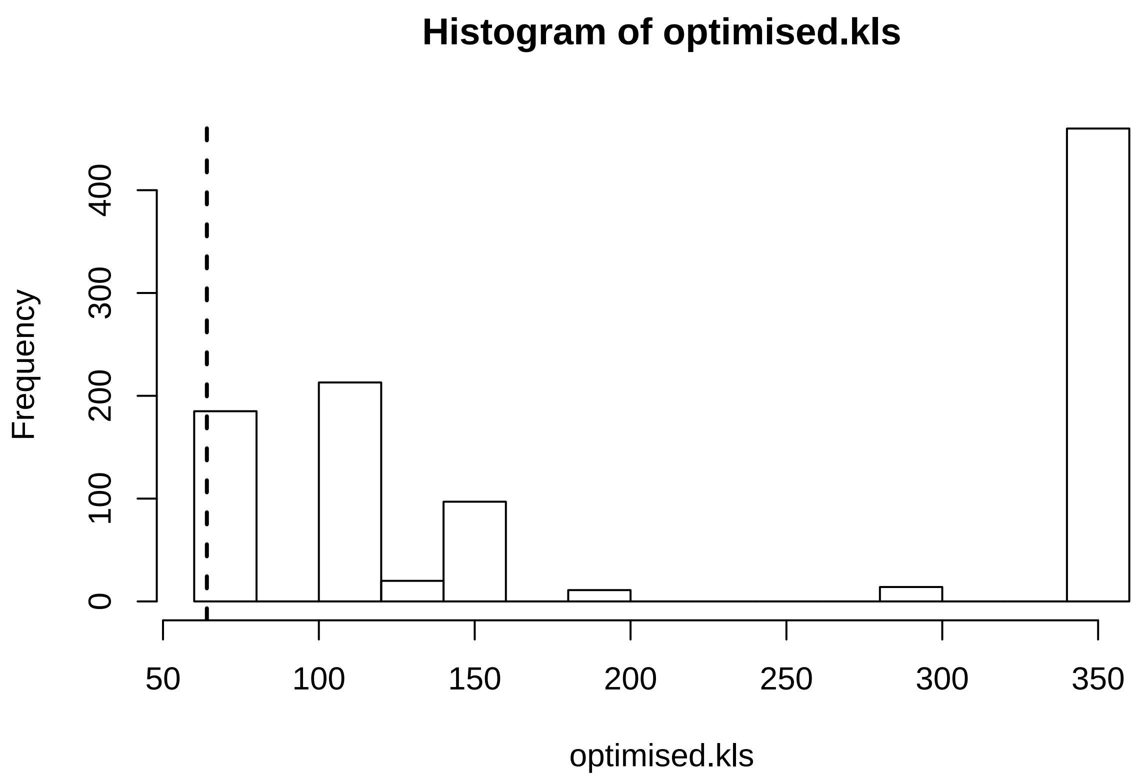
```r
( AlphasMSMGauss.tbl[1, ] <- alphaMaxEnt.opt )
```

```
## [1] 0 1 0 0 0 0
```

```r
ab.MaxEnt.star <- pool_par_gauss(alphaMaxEnt.opt, mv, vv)

# Prior
(PaperMSMGauss.tbl[2, 1:3] <- stat_gauss(ab.MaxEnt.star))
```

```
## [1] 0.2407809 0.2017515 0.2798103
```

```r
####### Minimum KL

N <- 1000 ## could increase to, say, 10000 in order to make sure, but it's fine
kl.many.startingPoints <- matrix(rnorm(n = (K-1)*N, mean = 0, sd = 100), ncol = K-1, nrow = N)
many.kls <- lapply(1:N, function(i) {
  optim(kl.many.startingPoints[i, ], optklgauss_inv, mp = mv, vp = vv, type = "fp")
})
optimised.kls <- unlist(lapply(many.kls, function(x) x$value))

hist(optimised.kls)
abline(v = optimised.kls[which.min(optimised.kls)], lty = 2, lwd = 2)
```



```r
alphaKL.opt <- alpha_01(many.kls[[which.min(optimised.kls)]]$par)

round(AlphasMSMGauss.tbl[2, ] <- alphaKL.opt, 2)
```

```
## [1] 0.17 0.83 0.00 0.00 0.00 0.00
```

```r
ab.KL.star <- pool_par_gauss(alphaKL.opt, mv, vv)

# Prior
(PaperMSMGauss.tbl[3, 1:3] <- stat_gauss(ab.KL.star))
```

```
## [1] 0.1601554 0.1287551 0.1915557
```

```r
####### Hierarchical priors
require("LearnBayes")
```

```
## Loading required package: LearnBayes
```

```r
M <- 100000
X <- c(1, 1, 1, 1, 1, 1)/10
alpha.MC.dirichlet <- rdirichlet(M, X)
alpha.MC.logisticNormal <- rlogisticnorm(N = M,
                              m = digamma(X)-digamma(X[K]),
                              Sigma = constructSigma(X))

apply(alpha.MC.dirichlet, 2, mean)
```

```
## [1] 0.1673374 0.1665987 0.1669955 0.1671761 0.1656557 0.1662365
```

```r
apply(alpha.MC.logisticNormal, 2, mean)
```

```
## [1] 0.1671627 0.1676562 0.1669083 0.1665868 0.1661364 0.1655496
```

```r
apply(alpha.MC.dirichlet, 2, sd)
```

```
## [1] 0.2955351 0.2949382 0.2955590 0.2954523 0.2937906 0.2946246
```

```r
apply(alpha.MC.logisticNormal, 2, sd)
```

```
## [1] 0.3442450 0.3445844 0.3438166 0.3436327 0.3431704 0.3427884
```

```r
gauss.par.dirichlet <- apply(alpha.MC.dirichlet, 1, function(w) pool_par_gauss(w, mv, vv))
gauss.par.logisticNormal <- apply(alpha.MC.logisticNormal, 1, function(w) pool_par_gauss(w, mv, vv))

phi.par.dirichlet <- apply(gauss.par.dirichlet, 2, function(x) rnorm(1, x[1], x[2]))
phi.par.logisticNormal <- apply(gauss.par.logisticNormal, 2, function(x) rnorm(1, x[1], x[2]))
# Prior
PaperMSMGauss.tbl[4, 1] <- mean(phi.par.dirichlet)
PaperMSMGauss.tbl[4, 2:3] <- quantile(phi.par.dirichlet, c(.025, .975))

PaperMSMGauss.tbl[5, 1] <- mean(phi.par.logisticNormal)
PaperMSMGauss.tbl[5, 2:3] <- quantile(phi.par.logisticNormal, c(.025, .975))


####### Using sample sizes

alphas.sampleSize <- meta$SampleSize/sum(meta$SampleSize)

( AlphasMSMGauss.tbl[3, ] <- alphas.sampleSize )
```

```
## [1] 0.1496815 0.1048681 0.1412648 0.2650136 0.1460419 0.1931301
```

```r
ab.sampleSize <- pool_par_gauss(alphas.sampleSize, mv, vv)

# Prior
( PaperMSMGauss.tbl[6, 1:3] <- stat_gauss(ab.sampleSize) )
```

```
## [1] 0.1322952 0.1093096 0.1552809
```

```r
#### Finally, tables!

round(PaperMSMGauss.tbl, 3)
```

```
##                               mean.prior lower.prior upper.prior
## equal_weights                      0.122       0.099       0.145
## maximum_entropy                    0.241       0.202       0.280
## minimum_KL                         0.160       0.129       0.192
## hierarchical_Dirichlet             0.133       0.063       0.250
## hierarchical_LogisticNormal        0.138       0.059       0.259
## Sample_size                        0.132       0.109       0.155
```

```r
round(AlphasMSMGauss.tbl, 3)
```

```
##                 alpha_0 alpha_1 alpha_2 alpha_3 alpha_4 alpha_5
## maximum_entropy   0.000   1.000   0.000   0.000   0.000   0.000
## minimum_KL        0.171   0.829   0.000   0.000   0.000   0.000
## Sample_size       0.150   0.105   0.141   0.265   0.146   0.193
```

```r
round(PaperMSMGauss.tbl, 2)
```

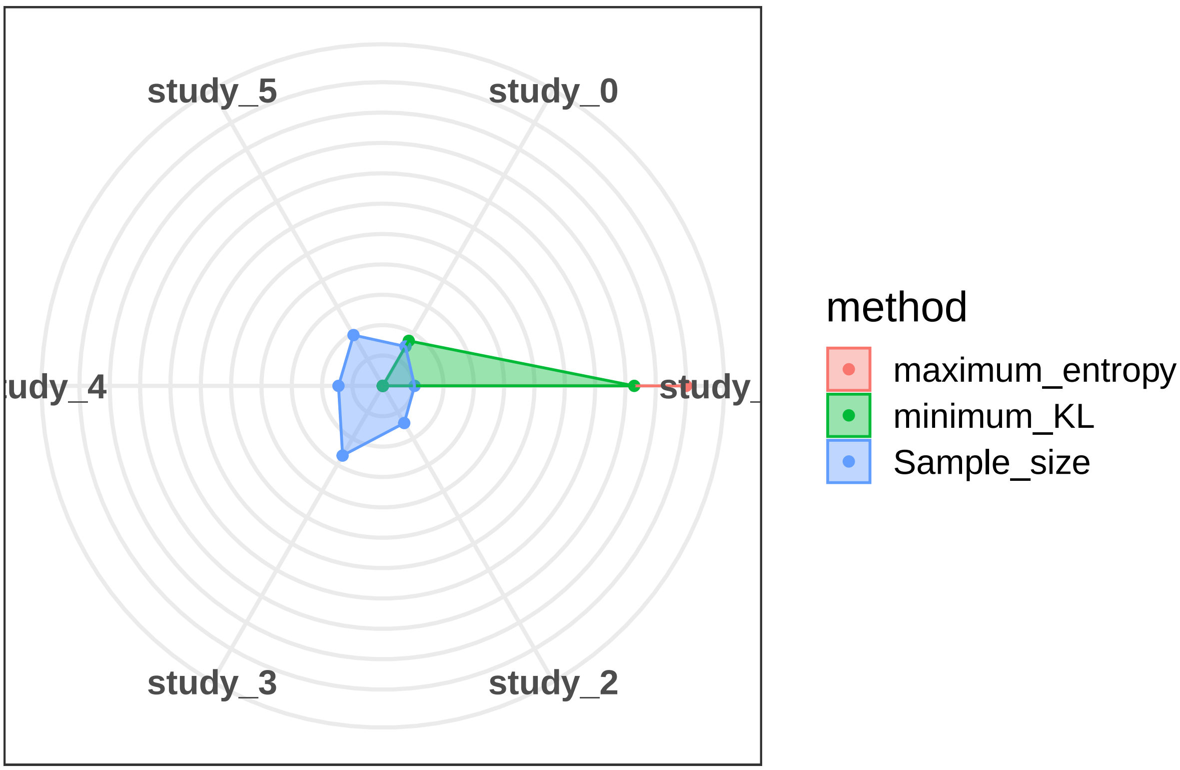
```
##                               mean.prior lower.prior upper.prior
## equal_weights                       0.12        0.10        0.15
## maximum_entropy                     0.24        0.20        0.28
## minimum_KL                          0.16        0.13        0.19
## hierarchical_Dirichlet              0.13        0.06        0.25
## hierarchical_LogisticNormal         0.14        0.06        0.26
## Sample_size                         0.13        0.11        0.16
```

```r
round(AlphasMSMGauss.tbl, 2)
```

```
##                 alpha_0 alpha_1 alpha_2 alpha_3 alpha_4 alpha_5
## maximum_entropy    0.00    1.00    0.00    0.00    0.00    0.00
## minimum_KL         0.17    0.83    0.00    0.00    0.00    0.00
## Sample_size        0.15    0.10    0.14    0.27    0.15    0.19
```

```r
write.csv(round(PaperMSMGauss.tbl, 3), file = "../data/output/MSM_Gaussian_stat.csv", row.names = TRUE)
write.csv(round(AlphasMSMGauss.tbl, 3), file = "../data/output/MSM_Gaussian_weights.csv", row.names = T

####### Plotting

posterior_studies <- data.frame(
  alpha = as.numeric(c(AlphasMSMGauss.tbl[1, ], AlphasMSMGauss.tbl[2, ], AlphasMSMGauss.tbl[3, ])),
  lwr = rep(NA, 18),
  upr = rep(NA, 18),
  study = rep(paste("study_", 0:(K-1), sep = ""), 3),
  method = rep(c("maximum_entropy", "minimum_KL", "Sample_size"), each = K)
)
```

```r
####
radar_alphas <- ggplot(data = posterior_studies,
       aes(x = study, y = alpha, group = method, colour = method, fill = method)) +
  geom_point() +
  geom_polygon(alpha = 0.4) +
  theme_bw(base_size = 16) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1),
                     breaks = number_ticks(10)) +
  coord_radar() +
  theme(axis.title.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.x = element_text(face = "bold"),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank()
  )
radar_alphas
```




```r
ggsave(plot = radar_alphas, filename = "../plots/alphas_radar_MSMGaussian.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```r
#############
# Now  let's look at marginal likelihoods for the pooled priors

pars <- list(equal_weights = ab.Equal.star,
             maximum_entropy = ab.MaxEnt.star,
             minimum_KL = ab.KL.star,
             sample_Size = ab.sampleSize)
```

```r
pars
```

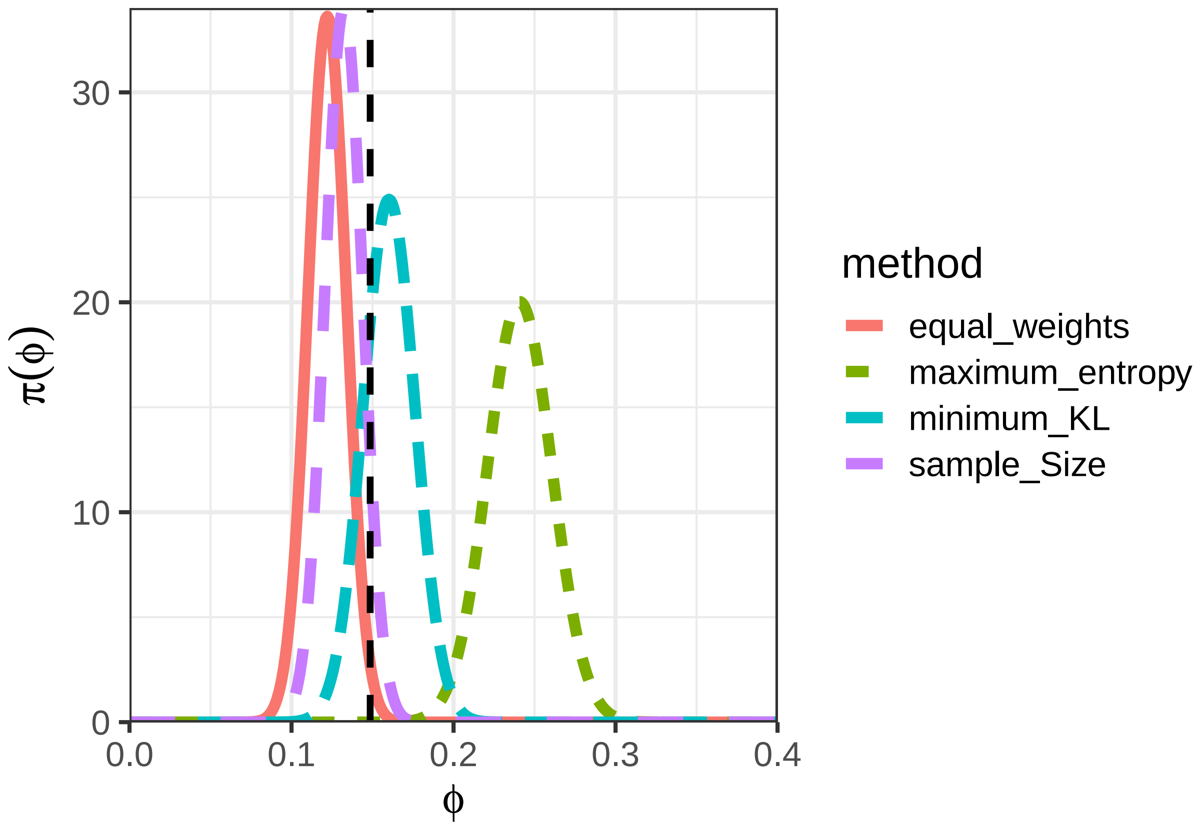
```
## $equal_weights
## [1] 0.12205302 0.01186498
##
## $maximum_entropy
## [1] 0.24078091 0.01991334
##
## $minimum_KL
## [1] 0.16015539 0.01602084
##
## $sample_Size
## [1] 0.13229523 0.01172759
```

```r
apply(AlphasMSMGauss.tbl, 1, get_ratio)
```

```
## maximum_entropy      minimum_KL     Sample_size
##             Inf        4.838260        1.372203
```

```r
J <- length(pars)
posterior.densities.list <- vector(J, mode ="list")
for (j in 1:J){
  posterior.densities.list[[j]] <- data.frame(
    phi = phi.grid,
    dens = dnorm(phi.grid, mean = pars[[j]][1], sd = pars[[j]][2]),
    method = names(pars)[j]
  )
}

posterior.densities.df <- do.call(rbind, posterior.densities.list)
posterior.densities.df$distribution <- "Gaussian"

write.csv(posterior.densities.df, "../data/output/MSM_Gaussian_densities.csv", row.names = FALSE)

method_posteriors <- ggplot(posterior.densities.df, aes(x = phi, y = dens,
                                                        linetype = method, colour = method)) +
  geom_line(size = 2) +
  scale_x_continuous(expression(phi), expand = c(0, 0), limits = c(0, .4)) +
  scale_y_continuous(expression(pi(phi)), expand = c(0, 0)) +
  geom_vline(xintercept = sum(meta$HIV)/sum(meta$SampleSize), linetype = "dashed", size = 1.2) +
  theme_bw(base_size = 16)

method_posteriors
```

```
## Warning: Removed 2400 rows containing missing values (geom_path).
```




```
ggsave(method_posteriors, filename = "../plots/method_posterior_densities_MSMGaussian.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 2400 rows containing missing values (geom_path).
```