MSM_metaAnalysis_Beta_example.r

luiz

2019-06-24

```
# 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
```

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

meta$SampleSize
```

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

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

beta_mean(av, bv)
```

```
## [1] 0.06818182 0.24190065 0.09951846 0.24164524 0.09006211 0.11750881
```

```
beta_mode(av, bv)
```

```
## [1] 0.06686930 0.24078091 0.09822866 0.24120172 0.08878505 0.11660777
```

```
beta_sd(av, bv)
```

```
## [1] 0.009803906 0.019880284 0.011983866 0.012525755 0.011271899 0.011032416
```

```
beta_sd(av, bv)^2
```

```
## [1] 9.611658e-05 3.952257e-04 1.436130e-04 1.568945e-04 1.270557e-04
## [6] 1.217142e-04
```

```
data.frame(meta$Study, round ( t( apply(cbind(av, bv), 1, stat_beta)  ), 3))
```

```
##                           meta.Study    av    V2    V3
## 1                   Tun_et_al_2008 0.068 0.050 0.089
## 2               Barcellos_et_al_2003 0.242 0.204 0.282
## 3                Carneiro_et_al_2003 0.100 0.077 0.124
## 4               Sutmoller_et_al_2002 0.242 0.218 0.267
## 5 Brazilian_Ministery_of_Health_2000 0.090 0.069 0.113
## 6                Harrison_et_al_1999 0.118 0.097 0.140
```

```
# Individual entropies
entropies <- rep(NA, K)
for(k in 1:K) entropies[k] <- entropy_beta(av[k], bv[k])
```

```
entropies
```

```
## [1] -3.212007 -2.500142 -3.009108 -2.961447 -3.070775 -3.090198
```

```
## Entropy surface (for a future dominance analysis)

library(fields)
```

```
## Loading required package: spam
```

```
## Loading required package: dotCall64
```

```
## Loading required package: grid
```

```
## Spam version 2.2-2 (2019-03-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: maps
```

```
## See https://github.com/NCAR/Fields for
##  an extensive vignette, other supplements and source code
```

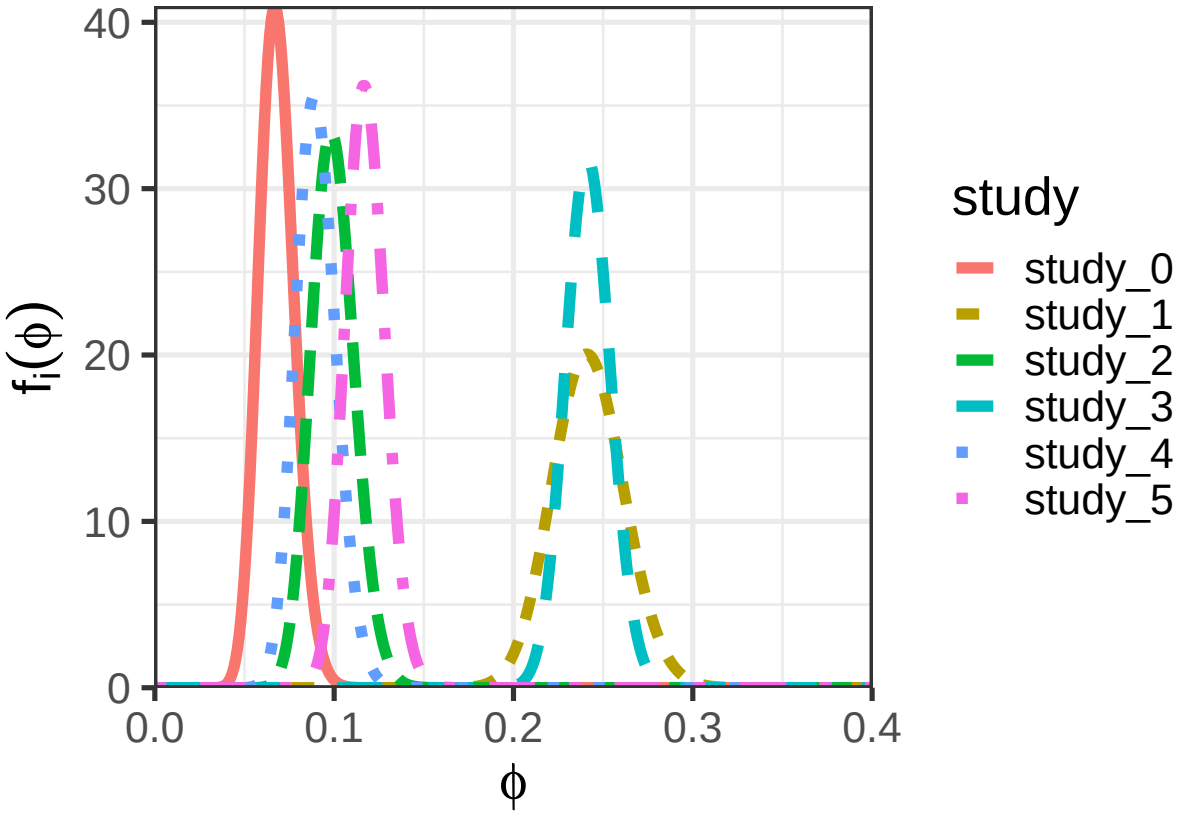
```
ES <- entropy_surface_beta(av, bv)
export <- TRUE
if(export){
  pdf("../plots/entropy_surface_MSMBetaExample.pdf")
}
image.plot(ES$as, ES$bs, ES$M,
           xlab = expression(a), ylab = expression(b), horizontal = TRUE,
           cex.lab = 1.5, cex.axis = 1.5, axis.args = list(font = 2),
           legend.cex = 1.5,
           legend.lab = expression(H[pi]), main = "Entropy Beta distribution", font = 2)
if(export){
  dev.off()
}
```

```
## pdf
##   2
```

```
############
PaperMSMBeta.tbl <- data.frame(mean.prior = rep(NA, 6), lower.prior = NA,
                               upper.prior = NA)
rownames(PaperMSMBeta.tbl) <- c("equal_weights", "maximum_entropy", "minimum_KL",
                               "hierarchical_Dirichlet", "hierarchical_LogisticNormal", "Sample_size")

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

```r
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 = dbeta(phi.grid, shape1 = av[k], shape2 = bv[k]),
                                     study = paste("study_", k-1, sep = ""))

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

study_priors <- ggplot(study.densities.df, aes(x = phi, y = dens,
                                               linetype = study, colour = study)) +
  geom_line(size = 2) +
  # scale_linetype_manual(values = c("twodash", "dotted", "longdash", "solid"))+
  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_MSMBeta.pdf")
```

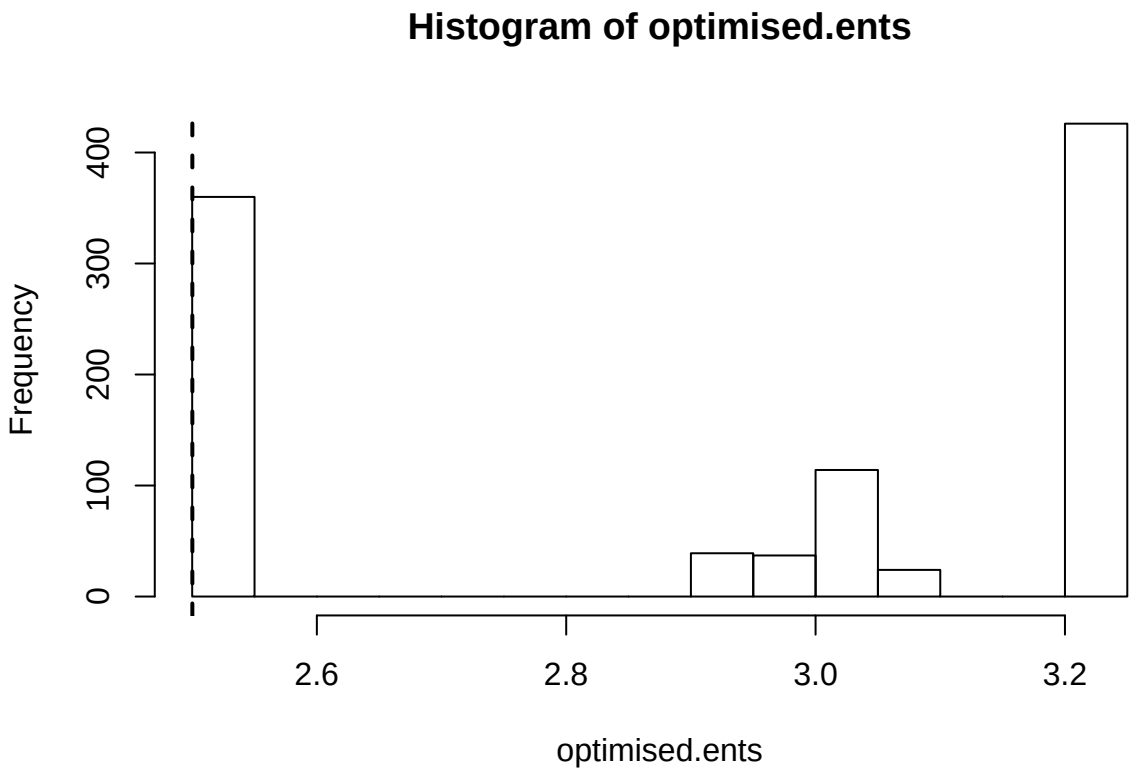
```
## Saving 6.5 x 4.5 in image
```

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

```
###### Equal weights

alphaEqual <- rep(1/K, K)

ab.Equal.star <- pool_par(alphaEqual, av, bv)
# Prior
(PaperMSMBeta.tbl[1, 1:3] <- stat_beta(ab.Equal.star))
```

```
## [1] 0.1495009 0.1246565 0.1761523
```

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

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, ], optentbeta_inv, ap = av, bp = bv)
})
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)

round(alphaMaxEnt.opt, 2)
```

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

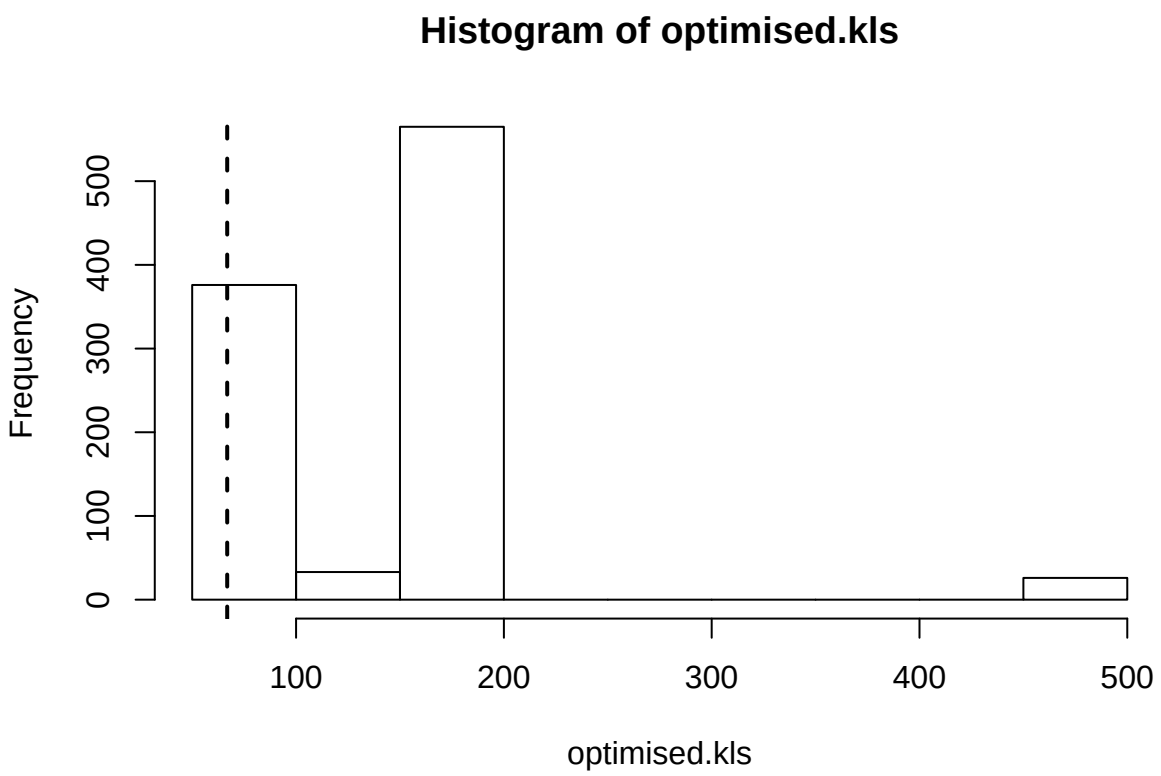
```r
( AlphasMSMBeta.tbl[1, ] <- alphaMaxEnt.opt )
```

```
## [1] 2.898465e-15 1.000000e+00 2.220446e-16 0.000000e+00 0.000000e+00
## [6] 0.000000e+00
```

```r
ab.MaxEnt.star <- pool_par(alphaMaxEnt.opt, av, bv)

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

```
## [1] 0.2419006 0.2040203 0.2818927
```

```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, ], optklbeta_inv, ap = av, bp = bv, 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(AlphasMSMBeta.tbl[2, ] <- alphaKL.opt, 2)
```

```
## [1] 0.53 0.47 0.00 0.00 0.00 0.00
```

```r
ab.KL.star <- pool_par(alphaKL.opt, av, bv)

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

```
## [1] 0.1341829 0.1074299 0.1633742
```

```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] 0.1681538 0.1649428 0.1682642 0.1654654 0.1670445 0.1661293
```

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

```
## [1] 0.1659923 0.1646214 0.1682818 0.1650614 0.1689670 0.1670761
```

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

```
## [1] 0.2955677 0.2929785 0.2964780 0.2939351 0.2942324 0.2941033
```

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

```
## [1] 0.3434359 0.3418889 0.3453067 0.3421141 0.3451033 0.3436754
```

```r
beta.par.dirichlet <- alpha.MC.dirichlet %*% cbind(av, bv)
beta.par.logisticNormal <- alpha.MC.logisticNormal %*% cbind(av, bv)

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

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



####### Using sample sizes

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

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

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

```r
ab.sampleSize <- pool_par(alphas.sampleSize, av, bv)
```

```r
# Prior
(PaperMSMBeta.tbl[6, 1:3] <- stat_beta(ab.sampleSize))
```

```
## [1] 0.1615843 0.1369570 0.1878078
```

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

round(PaperMSMBeta.tbl, 3)
```

```
##                             mean.prior lower.prior upper.prior
## equal_weights                    0.150       0.125       0.176
## maximum_entropy                  0.242       0.204       0.282
## minimum_KL                       0.134       0.107       0.163
## hierarchical_Dirichlet           0.144       0.066       0.253
## hierarchical_LogisticNormal      0.143       0.060       0.261
## Sample_size                      0.162       0.137       0.188
```

```r
round(AlphasMSMBeta.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        0.534   0.466   0.000   0.000   0.000   0.000
## Sample_size       0.150   0.105   0.141   0.265   0.146   0.193
```

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

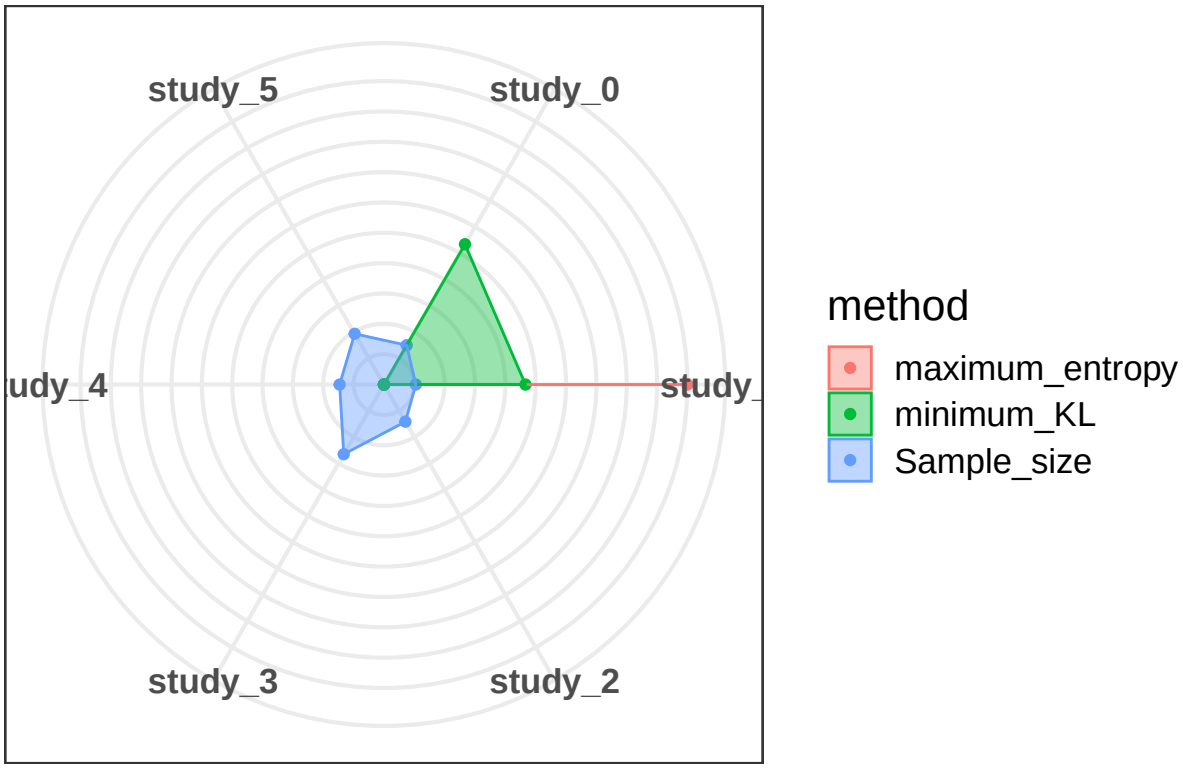
```
##                             mean.prior lower.prior upper.prior
## equal_weights                     0.15        0.12        0.18
## maximum_entropy                   0.24        0.20        0.28
## minimum_KL                        0.13        0.11        0.16
## hierarchical_Dirichlet            0.14        0.07        0.25
## hierarchical_LogisticNormal       0.14        0.06        0.26
## Sample_size                       0.16        0.14        0.19
```

```r
round(AlphasMSMBeta.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         0.53    0.47    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(PaperMSMBeta.tbl, 3), file = "../data/output/MSM_Beta_stat.csv", row.names = TRUE)
write.csv(round(AlphasMSMBeta.tbl, 3), file = "../data/output/MSM_Beta_weights.csv", row.names = TRUE)

####### Plotting

posterior_studies <- data.frame(
  alpha = as.numeric(c(AlphasMSMBeta.tbl[1, ], AlphasMSMBeta.tbl[2, ], AlphasMSMBeta.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)
)

####
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
```




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

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

```
#############
# 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)

lapply(pars, function(x) c(beta_mean(a = x[1], b = x[2]), beta_sd(a = x[1], b = x[2]) ) )
```

```
## $equal_weights
```

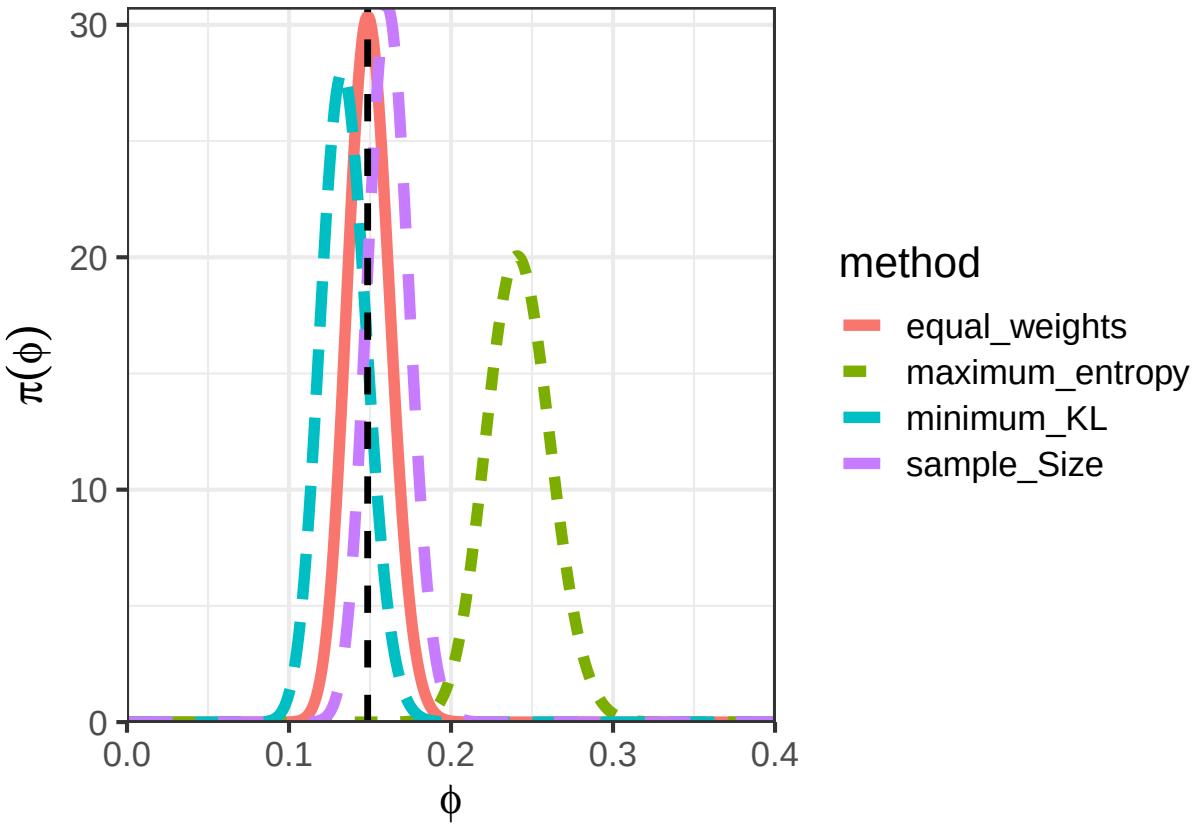
```
## [1] 0.14950091 0.01314675
##
## $maximum_entropy
## [1] 0.24190065 0.01988028
##
## $minimum_KL
## [1] 0.13418294 0.01428718
##
## $sample_Size
## [1] 0.16158429 0.01298057
```

```
apply(AlphasMSMBeta.tbl, 1, get_ratio)
```

```
## maximum_entropy      minimum_KL     Sample_size
##    3.450102e+14    1.144914e+00    1.372203e+00
```

```
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 = dbeta(phi.grid, shape1 = pars[[j]][1], shape2 = pars[[j]][2]),
    method = names(pars)[j]
  )
}

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

write.csv(posterior.densities.df, "../data/output/MSM_Beta_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_MSMBeta.pdf")
```

```
## Saving 6.5 x 4.5 in image
## Warning: Removed 2400 rows containing missing values (geom_path).
```