BDA - Assignment 4

10/10/2021

Contents

a)

We begin by calculating the mean and the covariance matrix. The covariance matrix is calculated using the variance σ^2 and the covariance by the product of the standard variances scaled by the correlation $\sigma_\alpha \sigma_\beta \text{cor}(\alpha, \beta)$.

```
mean = c(0,10)
corr = 0.6
a_std = 2
b_std = 10

cov = matrix( c(a_std^2, a_std*b_std*corr, a_std*b_std*corr, b_std^2),nrow = 2)
mean
```

```
## [1]  0 10
```

```
cov
```

```
##      [,1] [,2]
## [1,]    4   12
## [2,]   12  100
```

The mean vector is and the covariance matrix can be seen printed above.

b)

```
alpha_mean = mean(bioassay_posterior$alpha)
beta_mean = mean(bioassay_posterior$beta)

alpha_interval = quantile(probs = c(0.05, 0.95), bioassay_posterior$alpha)
```

```
beta_interval = quantile(probs = c(0.05, 0.95), bioassay_posterior$beta)

alpha_mean_MSCE = sqrt(var(bioassay_posterior$alpha)/length(bioassay_posterior$alpha))
beta_mean_MSCE = sqrt(var(bioassay_posterior$beta)/length(bioassay_posterior$beta))

alpha_interval_MSCE = mcse_quantile(bioassay_posterior$alpha, 0.9)
beta_interval_MSCE = mcse_quantile(bioassay_posterior$beta, 0.9)
```

The mean alpha is 1.0 with MCSE 0.015 and beta 10.6 with MSCE 0.076.

The qunatile interval for alpha is [-0.5, 2.6] with a MSCE of 0.029 and for beta [4, 19] with a MSCE of 0.167.

The MSCE allows us to estimate how much inaccuracy or noise the simulation contains.

c)

```
log_importance_weights = function(alpha, beta){
  bioassaylp(alpha, beta, bioassay$x, bioassay$y, bioassay$n)
}
```

```
liw = log_importance_weights(bioassay_posterior$alpha, bioassay_posterior$beta)
```

The some of the log importance weights are: -7.1657482, -7.2312935, -7.5522297, -7.1692339, -6.3182446, -6.2787463.

It's better to compute log ratios than to compute raw ratios due to the two reasons. One reason is that for large numbers log ratios are smaller numbers and thus easier to handle computationally. The other is that for small numbers problems with floating point errors becomes small.

d

```
normalized_importance_weights = function(alpha, beta){
  liw = log_importance_weights(alpha,beta)
  exp_liw = exp(liw)
  norm_exp_liw = exp_liw/sum(exp_liw)
  return(norm_exp_liw)
}
```

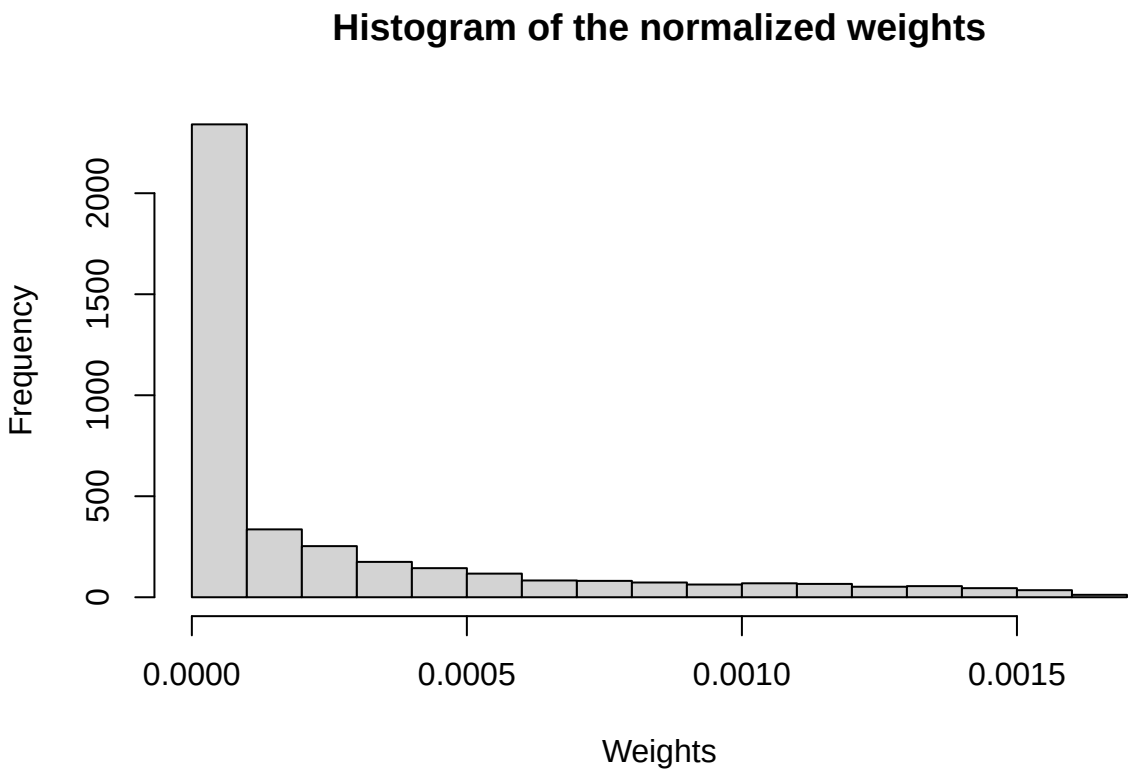
```
niw = normalized_importance_weights(bioassay_posterior$alpha,bioassay_posterior$beta)
```

The some of the normalized importance weights are: 1.3946567×10^{-4}, 1.3061751×10^{-4}, 9.4759021×10^{-5}, 1.3898039×10^{-4}, 3.2548689×10^{-4}, 3.3860032×10^{-4}.

By exponenting we remove the logarithms to get the real ratios. By dividing so that the sum becomes one we normalize the ratios. This allows us to scale the samples based on their ratios.

e)

```
n=4000
draws = rmvnorm(n, mean, cov)
normal_niw = normalized_importance_weights(draws[,1], draws[,2])
hist(normal_niw,
     main = 'Histogram of the normalized weights',
     xlab = 'Weights')
```




```
  S_eff = function(alpha, beta){
  niw = normalized_importance_weights(alpha,beta)
  1/sum(niw^2)
}
```

```
isess = S_eff(alpha = draws[,1], beta = draws[,2])
```

The importance sampling effective sample size is 1168.843.

g)

h)

```
posterior_mean = function(alpha, beta){
  seff = S_eff(alpha,beta)

  niw = normalized_importance_weights(alpha,beta)

  alpha_mean = mean(niw*alpha)/mean(niw)
  beta_mean = mean(niw*beta)/mean(niw)

  alpha_var = mean(niw*alpha^2)/mean(niw) - alpha_mean^2
  beta_var = mean(niw*beta^2)/mean(niw) - beta_mean^2

  alpha_mean_MSCE = sqrt(alpha_var/seff)
```

```
  beta_mean_MSCE = sqrt(beta_var/seff)
  res = list(alpha_mean = alpha_mean, beta_mean = beta_mean, alpha_mean_MSCE = alpha_mean_MSCE, beta_mea
  return(res)
}
```

```
post_mean = posterior_mean(draws[,1],draws[,2])
```

The posterior mean for alpha is 1.0 with a MCSE of 0.027 and beta is 11 with MCSE of 0.135.