

Assignment 4

anonymous

1 General information

! Reporting accuracy

For posterior statistics of interest, only report digits for which the Monte Carlo standard error (MCSE) is zero.

Example: If you estimate $E(\mu) = 1.234$ with $\text{MCSE}(E(\mu)) = 0.01$, you should report $E(\mu) = 1.2$.

2 Bioassay model

2.1 (a)

Given the data we have about the marginal distributions and co variances we can say that the mean of the bivariate normal distribution is $\mu = (0,10)$ and

the covariance matrix is $\begin{bmatrix} 4 & 12 \\ 12 & 100 \end{bmatrix}$

We get the value replacing in the following equation:

$$\text{Corr}(x, y) = \frac{\text{cov}(x, y)}{\text{sd}(x) * \text{sd}(y)}$$

2.2 (b)

Loading the library and the data.

```
# Useful functions: quantile()
# and mcse_quantile() (from aaltobda)

data("bioassay_posterior")
# The 4000 draws are now stored in the variable `bioassay_posterior`.
```

```
# The below displays the first rows of the data:  
head(bioassay_posterior)
```

```
      alpha      beta  
1 -0.02050577 10.032841  
2  1.21738518  4.504546  
3  3.04829407 16.239424  
4  1.32272770  4.924268  
5  1.36274817 12.880561  
6  1.08593225  5.943731
```

For alpha:

```
alpha_mean = mean(bioassay_posterior$alpha)  
mcse_mean_alpha = sqrt(4/4000)  
  
qs_alpha = quantile(bioassay_posterior$alpha, c(0.05, 0.95))  
  
mcse_q_alpha_5 = mcse_quantile(bioassay_posterior$alpha, 0.05)  
mcse_q_alpha_95 = mcse_quantile(bioassay_posterior$alpha, 0.95)
```

The mean of alpha is 1.0 and 5%, 95% quantiles are -0.5, 2.6 The mcse of the mean is 0.0316 and mcse of the quantiles are 0.026 and 0.0421. We round our estimates of mean and quantiles to the first digit because that's the last digit with 0 in the mcse.

For beta:

```
beta_mean = mean(bioassay_posterior$beta)  
mcse_mean_beta = sqrt(100/4000)  
  
qs_beta = quantile(bioassay_posterior$beta, c(0.05, 0.95))  
mcse_q_beta_5 = mcse_quantile(bioassay_posterior$beta, 0.05)  
mcse_q_beta_95 = mcse_quantile(bioassay_posterior$beta, 0.95)
```

The mean of beta is 11 and 5%, 95% quantiles are 4.0 and 19. The mcse of the mean is 0.1581 and mcse of the quantiles are 0.0704 and 0.2412. We round our estimates of 0.05 quantile to the first digit because that's the last digit with 0 in the mcse. For the mean and 0.95 quantiles we round the integers for the same reason.

3 Importance sampling

3.1 (c)

Since the prior and the proposal distribution are the same they cancel each other and we can just report the log likelihood as the log importance weights. It is better to compute log ratios instead of ratios because the multiplication of really small numbers can derive in rounding issues because of low precision while with log we sum and subtract making precision problems much less likely.

```
data("bioassay")
head(bioassay)

      x n y
1 -0.86 5 0
2 -0.30 5 1
3 -0.05 5 3
4  0.73 5 5

# Useful functions: bioassaylp (from aaltobda)
alpha_test = c(1.896, -3.6,  0.374, 0.964, -3.123, -1.581)
beta_test = c(24.76, 20.04, 6.15, 18.65, 8.16, 17.4)

log_importance_weights <- function(alpha, beta) {
  out = bioassaylp(alpha, beta, bioassay$x, bioassay$y, bioassay$n)
  # Do computation here, and return as below.
  # This is the correct return value for the test data provided above.
  # c(-8.95, -23.47, -6.02, -8.13, -16.61, -14.57)
}
```

3.2 (d)

Normalizing to make it sum to one transforms our values into probabilities.

```
normalized_importance_weights <- function(alpha, beta) {
  unnormalized = log_importance_weights(alpha, beta)
  exp_unnormalized = exp(unnormalized)
  denominator = sum(exp_unnormalized)

  out = exp_unnormalized/denominator
  # Do computation here, and return as below.
}
```

```

# This is the correct return value for the test data provided above.
# c(0.045, 0.000, 0.852, 0.103, 0.000, 0.000)
}

```

3.3 (e)

Write your answers and code here!

```

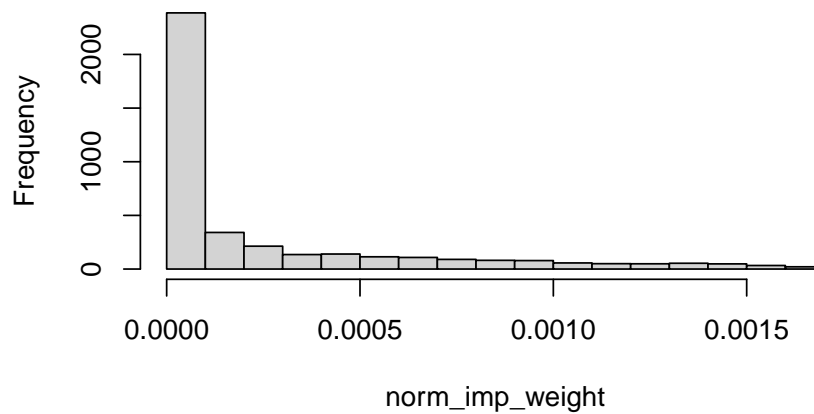
mean=c(0,10)
sigma=matrix(data=c(4,12,12,100), nrow=2, ncol=2)
samples = rmvnorm(4000, mean=mean, sigma=sigma )
samples_df = data.frame(samples)
colnames(samples_df) = c("alpha", "beta")

norm_imp_weight = normalized_importance_weights(samples_df$alpha, samples_df$beta)

hist(norm_imp_weight, main="Normalized importance ratios using samples from the
prior")

```

Normalized importance ratios using samples from the prior



3.4 (f)

```
S_eff <- function(alpha, beta) {  
  weights = normalized_importance_weights(alpha, beta)  
  squared_w = weights^2  
  out = 1 / sum(squared_w)  
  # Do computation here, and return as below.  
  # This is the correct return value for the test data provided above.  
  # 1.354  
}  
  
effective_sample = S_eff(samples_df$alpha, samples_df$beta)
```

The effective sample size is estimated to be 1151.19

3.5 (g)

The importance sampling effective sample size can be used as a measure of the quality of our estimation via importance sampling. The larger the effective sample size and closer to the number of observations, the better. It's like saying how many direct samples from the posterior are “equivalent” to the number of samples we have from the proposed distribution.

The heavy tail in the previous histogram (we would like something more uniform) corresponds to the distance we have between the 4000 samples and the actual effective sample size calculated above that is just above 1000.

3.6 (h)

```
posterior_mean <- function(alpha, beta) {  
  norm_imp_weight = normalized_importance_weights(alpha, beta)  
  mean_alpha = sum(alpha*norm_imp_weight)  
  mean_beta = sum(beta* norm_imp_weight)  
  # Do computation here, and return as below.  
  # This is the correct return value for the test data provided above.  
  # c(0.503, 8.275)  
  c(mean_alpha, mean_beta)  
}  
  
posterior_means_h = posterior_mean(samples_df$alpha, samples_df$beta)
```

```

effective_sample = S_eff(samples_df$alpha, samples_df$beta)

alpha_variance = mean(samples_df$alpha^2) - mean(samples_df$alpha)^2
beta_variance = mean(samples_df$beta^2) - mean(samples_df$beta)^2

mcse_alpha = sqrt(alpha_variance/effective_sample)
mcse_beta = sqrt(beta_variance/effective_sample)

```

Importance sampling has been computed calculating the normalized importance weights first. Those are computed using the likelihood for the data given a set of parameters. The parameters are sampled from the proposal distribution and in this case, the prior is the same as the proposed distribution. Then it's a matter of multiplying the parameter times the normalized importance weight to get the mean. In this case, since we wanted the mean of the parameter we applied the identity transformation (do nothing) but if we wanted the mean of some transformation we should have applied the function before multiplying.

Using importance sampling, the mean of alpha is estimated to be 1 and the mean of beta is estimated to be 11. Their MCSE are respectively 0.0595 and 0.296. We rounded the mean estimates digits to the last meaningful zero in the respective MCSEs.