BSDA: Assignment 4

November 19, 2023

```
knitr::opts_chunk$set(echo = TRUE)
set.seed(123)
```

Times used for reading and self-study exercise: 15
Time used for the assignment: 6
Good with the assignment: Good that we work with higher-dimensional priors
Things to improve in the assignment: More questions about Monte Carlo

Question 1; Bioassay model

For this exercise we will model the dose-response relation. The bioassay / experiment got a structure of (x_i, n_i, y_i) for $i = 1, 2, ..., k$ where x_i, n_i and y_i represents the dose level, the number of observations and the response of positive outcome, respectively. All for the ith group.

Q1a, Modeling the dose-response modeling

For this subtask we will compute the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ for parameters (α, β) where $\boldsymbol{\mu} : 2x1$ and $\Sigma : 2x2$. The joint prior distribution for parameters (α, β) is a bivariate normal distribution which gives us marginal prior distribution which is univariate normal distributed according to below.

Marginal prior distribution for α:
$\alpha \sim N(\mu_\alpha, \sigma_\alpha^2) = N(0, 2^2)$

Marginal prior distribution for β:
$\beta \sim N(\mu_\beta, \sigma_\beta^2) = N(10, 10^2)$

Based on the marginal prior distribution for α and β we get the following mean vector

$$\mu = \begin{bmatrix} \mu_\alpha \\ \mu_\beta \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

Now when the marginal prior distributions and mean vector $\boldsymbol{\mu} : 2x1$ has been calculated we can calculate the covariance matrix $\Sigma : 2x2$. It has been stated that the correlation between α and β corresponds to $\rho_{\alpha,\beta} = \rho_{\beta,\alpha} = 0.6$. The diagonal elements of Σ can extracted from the marginal distributions and the covariance can be calculated by using the $\rho_{\alpha,\beta} = 0.6$. Because we are working with multivariate statistics we can calculate the Σ by using the following formula:

Formula for the covariance matrix Σ:
$\Sigma = D^{1/2} R D^{1/2}$

where $D = \text{diag}\left(\sigma_\alpha^2, \sigma_\beta^2\right)$, $D = \text{diag}\left(\sqrt{\sigma_\alpha^2}, \sqrt{\sigma_\beta^2}\right)$ and R is a correlation matrix for α and β. So we are creating a function that computes the covariance matrix.

```
D = diag(c(2^2,10^2))
R = matrix(c(1,0.6,0.6,1),2,2)

covmatrix <- function(D = D, R = R) {

  Sigma <- D^{1/2} %*% R %*%  D^{1/2}
  return(Sigma)
}
covmatrix(D, R)
```

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

Hence, we get that Σ corresponds to

$$\Sigma = \begin{bmatrix} \sigma_{\alpha,\alpha} & \sigma_{\alpha,\beta} \\ \sigma_{\beta,\alpha} & \sigma_{\beta,\beta} \end{bmatrix} = \begin{bmatrix} 4 & 12 \\ 12 & 100 \end{bmatrix}$$

Q1b, Mean and quantiles, report digits based on MCSE

The dataset that is going to be used for this subtask consist of 4000 independent draws from the posterior distribution loaded from the bsda package.

```
library(bsda)
data("bioassay_posterior")
dataset <- bioassay_posterior

alpha <- c(1.896, -3.6, 0.374, 0.964, -3.123, -1.581)

beta <- c(24.76, 20.04, 6.15, 18.65, 8.16, 17.4)
```

Based on the data we will compute the mean and the 5% and 95% quantiles separately for parameter α and β and reporting digits based on the Monte Carlo standard errors (MCSE). The number of digits that should be reported is being stated in the Assignment. We start by creating a function that computes the mean and the MCSE for each parameter

```
# mean estimate for alpha and beta, report digits based on MCSE

mean_estimate <- function(data = bioassay_posterior) {

  mean.vector <- colMeans(data)
  MCSE.alpha <- sqrt ( var(data$alpha) / length( data$alpha) )
  MCSE.beta <- sqrt ( var(data$beta) / length( data$beta) )
  output <- c(mean.vector, MCSE.alpha, MCSE.beta)
  names(output) <- c("alpha",
                     "beta",
```

```
                      "MCSE.alpha",
                      "MCSE.beta")
  return(output)
}
mean_estimate(bioassay_posterior)
```

```
##       alpha        beta  MCSE.alpha   MCSE.beta
##  0.98522629 10.59648129  0.01482435  0.07560016
```

```
round(mean_estimate(bioassay_posterior)[1:2], digits = 1)
```

```
## alpha  beta
##   1.0  10.6
```

The mean for parameter α and β corresponds approximately to $\hat{\mu}_\alpha \approx 1$ and $\hat{\mu}_\beta \approx 10.6$ for the sample. The number of digits being reported are based on the Monte Carlo standard errors.

Now we calculate the 5% and 95% quantiles for each parameter and the result will also be presented based on the MCSE

```
post.quantile <- function(data = bioassay_posterior) {

  quantiles.alpha <- quantile(data$alpha, c(0.05, 0.95))

  lowerMCSE.alpha <- mcse_quantile(draws = data$alpha, prob = 0.05)

  higherMCSE.alpha <- mcse_quantile(draws = data$alpha, prob = 0.95)

  quantiles.beta <- quantile(data$beta, c(0.05, 0.95))

  lowerMCSE.beta <- mcse_quantile(draws = data$beta, prob = 0.05)

  higherMCSE.beta <- mcse_quantile(draws = data$beta, prob = 0.95)

  list <- list(quantiles.alpha, lowerMCSE.alpha, higherMCSE.alpha,
               quantiles.beta, lowerMCSE.beta, higherMCSE.beta)

  names(list) <- c("quantiles.alpha","lowerMCSE.alpha",
                   "higherMCSE.alpha",
                   "quantiles.beta","lowerMCSE.beta",
                   "higherMCSE.beta")

  return(list)

}
post.quantile(bioassay_posterior)
```

```
## $quantiles.alpha
##         5%        95%
## -0.4675914  2.6102028
##
## $lowerMCSE.alpha
##         mcse
```

```
## 1 0.02600412
##
## $higherMCSE.alpha
##         mcse
## 1 0.04206342
##
## $quantiles.beta
##        5%       95%
##  3.991403 19.340365
##
## $lowerMCSE.beta
##         mcse
## 1 0.07043125
##
## $higherMCSE.beta
##        mcse
## 1 0.2412129
```

```
round( post.quantile(bioassay_posterior)[["quantiles.alpha"]],
       digits = 1)
```

```
##   5%  95%
## -0.5  2.6
```

```
c(round(post.quantile(bioassay_posterior)[["quantiles.beta"]][1],
        digits = 1),
  round(post.quantile(bioassay_posterior)[["quantiles.beta"]][2],
        digits = 0))
```

```
##  5% 95%
##   4  19
```

and we got that the 5% and 95% quantiles for α corresponds approximately to $Q_{(\alpha,0.05)} \approx -0.5$ and $Q_{(\alpha,0.95)} \approx 2.6$, respectively. Also, the 5% and 95% quantiles for β corresponds approximately $Q_{(\beta,0.05)} \approx= 4$ and $Q_{(\beta,0.95)} \approx 19$, respectively.

Q1c, Important sampling, Target distribution, Proposal distribution

For this sub-task we want to implement a function for computing the log importance ratios / weights where we have the posterior distribution as the target distribution and the prior distribution as the proposal distribution. Further, the log-likelihood will be presented below and the likelihood comes from the binomial distribution for which the parameter θ is being estimated by a logistic regression. Also, the prior distribution is a bivariate normal distribution therefore the posterior will not result in a known statistical distribution. Hence, in our case we get the following vector of log-weights, log-likelihood, proposal and target distribution:

Proposal (prior) distribution $g(\theta^s)$:
$g(\theta) \sim N_2(\boldsymbol{\mu}, \Sigma)$ and

Log-likelihood function $log\left(p(y_i|\alpha, \beta, n_i, x_i)\right)$:
$log\left(p(y_i|\alpha, \beta, n_i, x_i)\right) \propto log\left(\left[\text{logit}^{-1}\left(\alpha + \beta x_i\right)\right]^{y_i}\left[1 - \text{logit}^{-1}\left(\alpha + \beta x_i\right)\right]^{n_i - y_i}\right)$

Target (posterior) distribution $q(\theta^s|y)$:
$q(\theta^s|y) \propto g(\theta) \cdot \sum_{i=i}^{k} log\left(p(y_i|\alpha, \beta, n_i, x_i)\right)$

Formula to calculate log-weights for (α, β) ie $w\left(\theta^s\right)$:
$w\left(\theta^s\right) = \frac{q(\theta^s|y)}{g(\theta^s)} = \frac{g(\theta) \cdot \sum_{i=i}^{k} log(p(y_i|\alpha,\beta,n_i,x_i))}{g(\theta)} = \sum_{i=i}^{k} log\left[p(y_i|\alpha, \beta, n_i, x_i)\right]$

We can see that the log-weights for this specific case corresponds to the log-likelihood for the whole sample, this is because we have target as posterior and proposal as the prior distribution. The prior distribution takes out each other therefore the log importance weights equals the log-likelihood for the whole sample. Below we present the source code for the function that computes the log weights.

```r
# this data is used to calculate the log likelihood
data("bioassay")

log_importance_weights <- function(alpha, beta) {

  n <- length(beta)

  loglikelihood <- c(bioassaylp(alpha = alpha,
                     beta = beta,
                     x = bioassay$x, # the dose level
                     y = bioassay$y, # nr of death in the group
                     n = bioassay$n))

  proposal <- rmvnorm(n = n, mean = c(0,10), sigma = covmatrix(D, R))

  target <- loglikelihood * proposal
  weight <- target / proposal
  weight <- weight[,1]
  return(weight)
}
```

Further, we are using the log likelihood for us to get logarithm of the posterior density and this is being done to avoid computational underflow and overflow when computing

the posterior distribution.

Q1d, Implement a function for computing normalized weights

For this sub-task will implement a function for computing the normalized ratios / weights from unnormalized ratios / weights. For previous section we calculated the log ratios so we need to exponentiate the log ratios from sub-task "Q1c" and then scale it to get it normalized. The formula to calculate the importance ratios can be found in the course book (Reference: Bayesian Data Analysis Third edition, chapter: 10 , page: 266)

Formula for calculating the normalized weights $\tilde{w}\left(\theta^s\right)$:
$\tilde{w}\left(\theta^s\right) = \frac{w(\theta^s)}{\sum_{s'=1}^{S} w\left(\theta^{s'}\right)}$

where $\tilde{w}\left(\theta^s\right)$ got the property of having the sum equal to 1. Below we present the source code for the function that computes the normalized weights.

```
normalized_importance_weights <- function(alpha = alpha,
                                          beta = beta){
  exp <- exp( log_importance_weights(alpha, beta) )
  normalized <- exp / sum(exp)
  return(normalized)
}
```

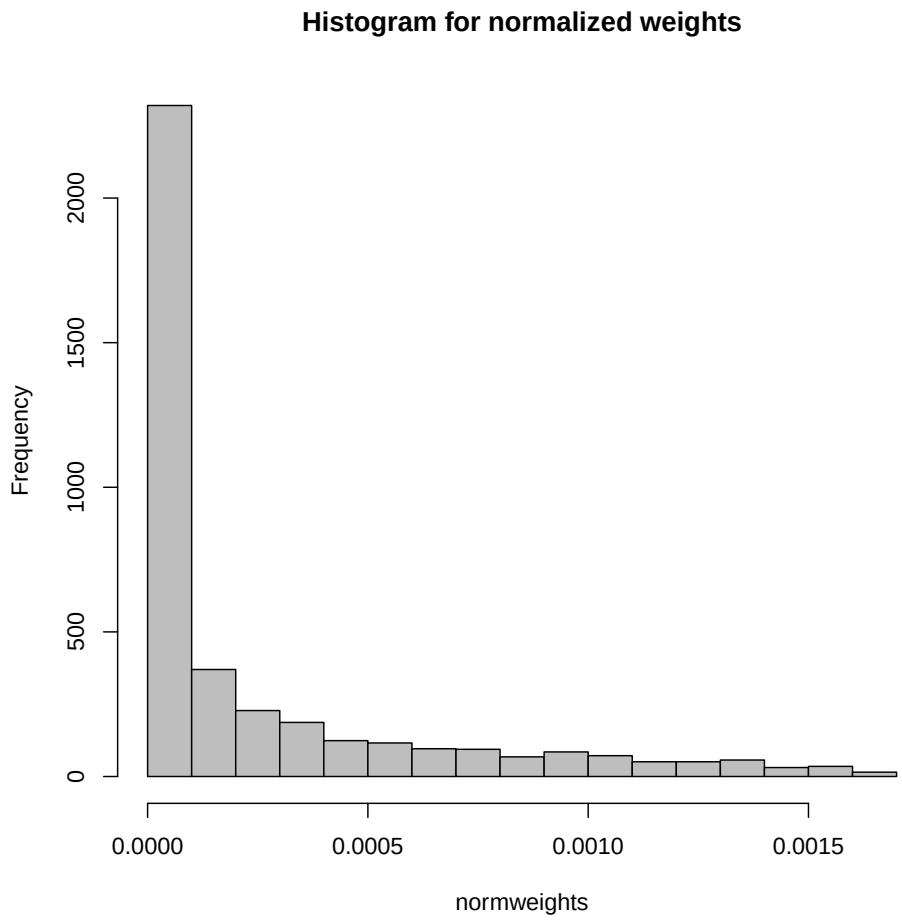
As stated above the normalized weights sum up to 1 so each normalized weight is located between 0 and 1. The sum / integral over the whole domain of a probability mass/density function equals 1 hence the different normalized weights can be seen as probabilities from a pmf/pdf.

Q1e, Compute and plot a histogram of the normalized weights

We will compute and plot a histogram of normalized weights where both α and β has been calculated based on a sample of 4000 draws from bivariate normal distribution which is the prior distribution.

```
# compute and plot a histogram of 4000 normalized weights
priorsample <- rmvnorm(n = 4000,
                       mean = c(0,10),
                       sigma = covmatrix(D, R))

normweights <- normalized_importance_weights(priorsample[,1],
                                             priorsample[,2])
hist(normweights,
     main = "Histogram for normalized weights",
     col = "gray")
```


Histogram for normalized weights
Frequency
0
500
1000
1500
2000
0.0000
0.0005
0.0010
0.0015
normweights

Q1f, Compute the importance sampling effective S_{eff}

Here we are using the importance ratios to calculate the sampling effective sample size S_{eff}. Further, because the variance for α and β is finite we can estimate the effective sample size by using formula

Effective sample size S_{eff}:
$S_{\text{eff}} = \frac{1}{\sum_{s=1}^{S}(\tilde{w}(\theta^s))^2}$

where $\tilde{w}(\theta^s)$ is the normalized weights and now we create a function that computes S_{eff}

```
S_eff <- function(alpha, beta) {

  square.weight <-  normalized_importance_weights(alpha,beta)^2

  Seff <- 1 / sum(square.weight)

  return(Seff)
}
S_eff(priorsample[,1], priorsample[,2]) # 1183.505
```

```
## [1] 1183.505
```

and the effective sample size for this sample corresponds to 1183.505.

Q1g, Sampling effective sample size

Often in statistical inference we assume independence between observation however this is often not the case in practice. The effective sample size can be seen as a measure in increase uncertainty of estimates when doing posterior inference due to correlations between between observations. Hence, if there would be no correlation in the sample then the sample size, let say S, would correspond to the effective sample size, ie $S = 4000 = S_{\text{eff}}$. But, given that we have dependent samples, S_{eff} corresponds to the number of independent samples with the same estimation power as the autocorrelated samples.

Q1h, Posterior mean using and Compute the mean

Here, the posterior mean will be calculated by using the importance sampling method and the mean / expected value will be calculated based on 4000 random draws. The formula can be found in the course literature. (Reference: Bayesian Data Analysis Third edition, chapter: 10 , page: 265)

We aim to calculate the expected value for the posterior distribution for α and β separately. Formula for the $E(\theta|y)$, likelihood, the weights w_i and the posterior means, ie $E(\alpha|y)$ and $E(\beta|y)$ are being presented below:

Formula for the weight w_i:
$w_i = \frac{p(y_i|\theta_i)}{\sum_{i=1}^{k} p(y_i|\theta_i)}, \forall i \in (1, ..., k)$

Formula for $E(\theta|y)$:
$E(\theta|y) = \frac{\frac{1}{S}\sum_{s=1}^{S}\theta^s w(\theta^s)}{\frac{1}{S}\sum_{s=1}^{S} w(\theta^s)} = \frac{\sum_{s=1}^{S}\theta^s w(\theta^s)}{\sum_{s=1}^{S} w(\theta^s)} = \sum_{s=1}^{S}\tilde{w}\theta^s$

Because $\boldsymbol{\theta}$ is a parameter vector of dimensions $\boldsymbol{\theta}' = (\alpha, \beta) : 2x1$ then this calculations needs to be done for respective parameter in $\boldsymbol{\theta}$. Further, by using the bioassaylp function we get the log likelihood so we need to take the exponentiate to get the likelihood and further calculate the normalized weights.

```
posterior_mean <- function(alpha = alpha, beta = beta) {

  n = length(alpha) # number of observation

  loglikelihood <- bioassaylp(alpha = alpha,
                      beta = beta,
                      x = bioassay$x, # the dose level
                      y = bioassay$y, # response of positive outcome
                      n = bioassay$n) # nr of observations

  likelihood <- exp(loglikelihood)

  weight <- likelihood / sum(likelihood)

  alpha.mean <- sum(alpha * weight) / sum(weight)

  beta.mean <- sum(beta * weight) / sum(weight)

  MCSE.alpha <- sqrt ( var(alpha) / S_eff(alpha, beta) )

  MCSE.beta <- sqrt ( var(beta) / S_eff(alpha, beta) )

  output <- c(alpha.mean, beta.mean,
              MCSE.alpha, MCSE.beta,
              S_eff(alpha, beta) )

  names(output) <- c("Alpha mean", "Beta mean",
                     "MCSE.alpha", "MCSE.beta",
                     "S_eff")

  return(output)
}

posterior_mean(alpha = priorsample[,1],
               beta = priorsample[,2])
```

```
##   Alpha mean    Beta mean   MCSE.alpha    MCSE.beta        S_eff
## 9.686982e-01 1.052381e+01 5.759682e-02 2.848615e-01 1.183505e+03
```

```
round(posterior_mean(alpha = priorsample[,1],
               beta = priorsample[,2])[1], digits = 1)
```

```
## Alpha mean
##          1
```

```
round(posterior_mean(alpha = priorsample[,1],
               beta = priorsample[,2])[2], digits = 0)
```

```
## Beta mean
```

```
##         11
```

The posterior mean vector for parameter α and β corresponds approximately to $\hat{\mu}_\alpha \approx 1$ and $\hat{\mu}_\beta \approx 11$ based on the importance sampling. Hence, the number of digits being reported are based on the Monte Carlo standard errors for the importance sampling.