Exercises for Lecture 11

Statistical Computing with R, 2023-24

Exercise 1

Consider the following `for` loop:

```
set.seed(13)
n.repl = 1000
weib.shape = c(2, 3)
weib.scale = c(5, 4)
min.weibs = rep(NA, n.repl)
for (i in 1:n.repl) {
  x1 = rweibull(1, shape = weib.shape[1], scale = weib.scale[1])
  x2 = rweibull(1, shape = weib.shape[2], scale = weib.scale[2])
  min.weibs[i] = min(x1, x2)
}
```

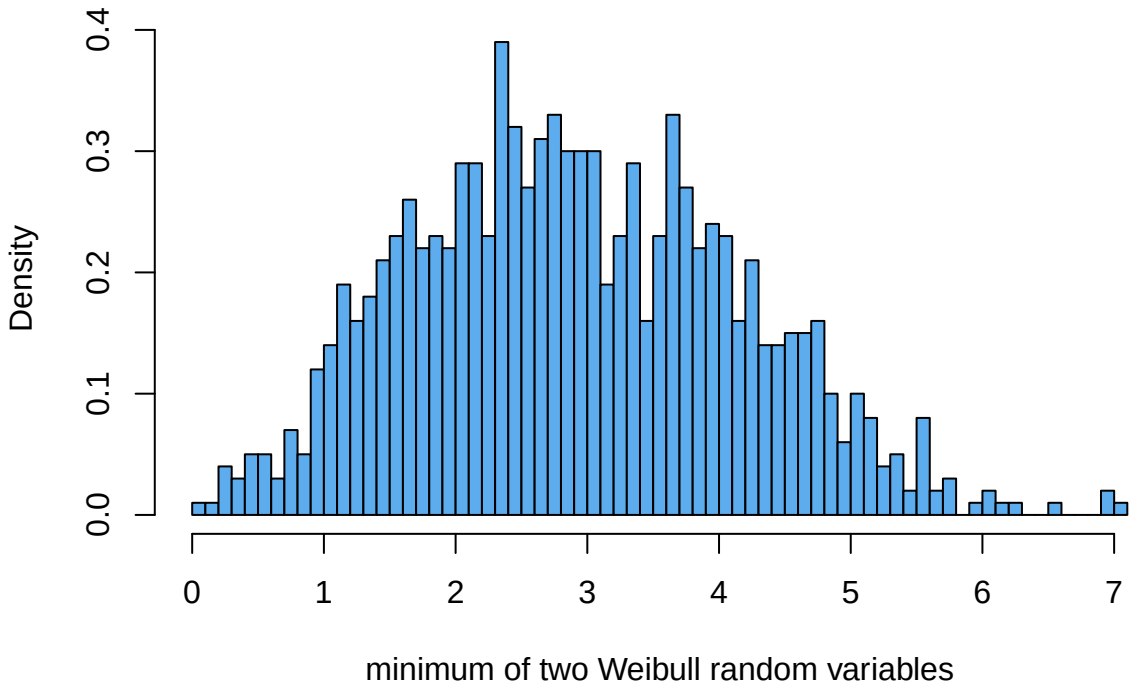
The code above implements a simulation / Monte Carlo experiment designed to evaluate the distribution of the random variable

$$Y = \min(X_1, X_2),$$

where $X_1 \sim Weib(\lambda_1, \nu_1)$ and $X_2 \sim Weib(\lambda_2, \nu_2)$ are two Weibull distributions with different scale (λ) and shape (ν) parameters.

The estimated density of Y that can be obtained from the code above is

```
hist(min.weibs, 50, prob = T, main = '', col = 'steelblue2',
     xlab = 'minimum of two Weibull random variables')
```



For the time being, you are not expected to know what a Monte Carlo experiment is, and how it can be used for estimation: we will cover this in the *Computational Statistics* course! What you need to understand, for now, is just what the code is doing so you can adjust it and reuse it to answer the questions given below.

1. Rewrite the `for` loop using `replicate( )`.
2. Compare the execution time of the `for` loop to that of the `replicate( )` implementation using the `benchmark( )` function. Set its argument `replications = 100`. Which of the two implementations is faster?
3. Use the code you wrote in (1) to perform two additional experiments:

a. one with $\lambda_1 = 0.5$, $\lambda_2 = 2$, $\nu_1 = 0.7$ and $\nu_2 = 1$;
b. one with $\lambda_1 = \lambda_2 = 3$ and $\nu_1 = \nu_2 = 1$.

Create a histogram showing the estimated (empirical) distribution of $Y = \min(X_1, X_2)$ in experiments (a) and (b).

Exercise 2

During the lecture we implemented the EM algorithm for a mixture of two normal distributions whose variances were assumed to be known and equal to 1, i.e. $\sigma_1 = \sigma_2 = 1$. Now, let's consider the more general case of a mixture of two normals

$$f_X(x) = \pi_1 f_{X_1}(x; \mu_1, \sigma_1) + \pi_2 f_{X_2}(x; \mu_2, \sigma_2)$$

where $X_1 \sim N(\mu_1, \sigma_1^2)$, $X_2 \sim N(\mu_2, \sigma_2^2)$, and all density parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$ and mixing proporton parameters π_1, π_2 are unknown.

1. Adapt the negative log-likelihood function used in the lecture to this new problem where besides μ_1 and μ_2, also σ_1 and σ_2 are unknown.

Now consider the following simulated dataset:

```
set.seed(13)
n = 2000; pi1 = 0.35
mu1 = 0.8; mu2 = 2.5
sigma1 = 0.8; sigma2 = 0.6
group = sample(1:2, n, replace = T, prob = c(pi1, 1-pi1))
table(group)
```

```
## group
##    1    2
##  698 1302
```

```
x = rep(NA, n)
x[group == 1] = rnorm(sum(group == 1), mu1, sd = sigma1)
x[group == 2] = rnorm(sum(group == 2), mu2, sd = sigma2)
```

2. Implement the EM algorithm to estimate $\mu_1, \mu_2, \sigma_1, \sigma_2, \pi_1, \pi_2$ for this dataset. You may reuse and adjust suitably the code used during lecture 11 to solve this problem.
3. Use your algorithm implementation to estimate the parameters of interest using a single starting point. Make sure to check whether the algorithm has converged.
4. Consider 10 different starting points for the algorithm, and choose the solution that yields the highest loglikelihood. What are your maximum likelihood estimates of $\mu_1, \mu_2, \sigma_1, \sigma_2, \pi_1, \pi_2$?
5. Compare your results to the true parameter values. Which inferred component corresponds to which group?
6. What is the percentage of misclassified observations?