MTH3045: Statistical Computing
Example partial coursework answer

Ben Youngman
(b.youngman@exeter.ac.uk, Laver 817)

This example answer is a document demonstrating how the example coursework can be answered in a way that would get full marks. The answer is designed to meet the criteria outlined in 'Guidance on Courseworks'. You should aim to meet these criteria. This example answer is merely for illustration.

Question 1

The function `dmvn2(y, mu, Sigma, log)` below evaluates the log-likelihood for the multivariate Normal distribution based on a QR decomposition of the variance-covariance matrix.

```r
dmvn2 <- function(y, mu, Sigma, log = TRUE) {
  # Function to evaluate multivariate Normal pdf
  # y and mu are p-vectors
  # Sigma is a p x p matrix
  # log is a logical
  # Returns scalar, on log scale, if log == TRUE.
  p <- nrow(y)
  res <- y - mu
  QR <- qr(Sigma)
  Q <- qr.Q(QR)
  R <- qr.R(QR)
  out <- - 0.5 * sum(log(abs(diag(R)))) - 0.5 * p * log(2 * pi) -
            0.5 * colSums(res * backsolve(R, crossprod(Q, res)))
  out <- sum(out)
  if (!log)
    out <- exp(out)
  out
}
```

Question 1(a)

The following reads in the data, mean vector and variance-covariance matrix

```r
y1 <- c(0.85, 1.97, 7.35)
y2 <- c(-0.35, 2.29, 7.39)
y3 <- c(4.55, 3.85, 11.29)
y4 <- c(4.46, 0.5, 4.47)
```

```r
y <- cbind(y1, y2, y3, y4)

mu <- c(3.5, 1.5, 6.2)

Sigma <- cbind(
  c(5, 0.2, 1.1),
  c(0.2, 1.4, 1.2),
  c(1.1, 1.2, 3.6)
)
```

and then the following reads in the function `dmvn1()` given in the question.

```r
dmvn1 <- function(y, mu, Sigma, log = TRUE) {
  # Function to evaluate multivariate Normal pdf
  # y and mu are p-vectors
  # Sigma is a p x p matrix
  # log is a logical
  # Returns scalar, on log scale, if log == TRUE.
  p <- nrow(y)
  res <- y - mu
  out <- - 0.5 * determinant(Sigma)$modulus - 0.5 * p * log(2 * pi) -
            0.5 * colSums(res * (solve(Sigma) %*% res))
  out <- sum(out)
  if (!log)
    out <- exp(out)
  out
}
```

Log-likelihoods using `dmvn1()` and `dmvn2()` are calculated below and stored as `loglik1` and `loglik2`, respectively,

```r
loglik1 <- dmvn1(y, mu, Sigma)
loglik2 <- dmvn2(y, mu, Sigma)
```

which give log-likelihoods of -24.487 and -24.487, for `dmvn1()` and `dmvn2()`, respectively.

The following then checks whether the two log-likelihoods are the same

```r
all.equal(loglik1, loglik2)
```

```
## [1] TRUE
```

which they are, given that `R` returns `TRUE`.