

Exercise 1:

In the following we are defining the function `em2mvnorm()`, which implements the EM algorithm for mixtures with 2 multivariate normal distributions.

The function has also an argument `plot` which allows to specify if the current allocation of the observations to one of the components based on the a-posteriori probabilities and the estimated mean values and their prediction regions at the 50% and 95% level are shown in a plot.

```
library("mvtnorm")
library("ellipse")
add <- function(x) Reduce("+", x)

library("colorspace")
getColor <- function(x, n = 9) diverge_hcl(n + 1)[round(x * n) + 1]

plotem <- function(y, z, pars, which = 1:2, ...) {
  plot(y[, which, drop = FALSE], col = getColor(z), pch = 19, ...)
  points(rbind(pars$mu1, pars$mu2)[, which], col = getColor(1:0), pch = 4, lwd = 2)
  lines(ellipse(pars$Sigma1[which, which], centre = pars$mu1[which],
    level = 0.5), col = getColor(1), lwd = 2)
  lines(ellipse(pars$Sigma1[which, which], centre = pars$mu1[which],
    level = 0.95), col = getColor(1), lty = 2)
  lines(ellipse(pars$Sigma2[which, which], centre = pars$mu2[which],
    level = 0.5), col = getColor(0), lwd = 2)
  lines(ellipse(pars$Sigma2[which, which], centre = pars$mu2[which],
    level = 0.95), col = getColor(0), lty = 2)
}

em2mvnorm <- function(y, pars, iter.max = 100, eps = 10^-4, plot = FALSE, ...) {
  loglik_old <- -Inf
  N <- nrow(y)
  for (i in seq_len(iter.max)) {
    z1 <- pars$pi * dmvnorm(y, pars$mu1, pars$Sigma1)
    z2 <- (1 - pars$pi) * dmvnorm(y, pars$mu2, pars$Sigma2)
    loglik_n <- z1 + z2
    z <- z1 / loglik_n

    loglik <- sum(log(loglik_n))
    if (abs((loglik - loglik_old) / loglik) < eps) break
    loglik_old <- loglik

    pars$pi <- mean(z)
    pars$mu1 <- colMeans(z * y) / pars$pi
    pars$mu2 <- colMeans((1 - z) * y) / (1 - pars$pi)
    pars$Sigma1 <- add(lapply(seq_len(N), function(i)
      z[i] * crossprod(y[i, , drop = FALSE] - pars$mu1))) / (N * pars$pi)
    pars$Sigma2 <- add(lapply(seq_len(N), function(i)
      (1 - z[i]) * crossprod(y[i, , drop = FALSE] - pars$mu2))) / (N * (1 - pars$pi))
    if (plot & interactive()) {
      Sys.sleep(1)
      plotem(y, z, pars, ...)
    }
  }
}
```

```

}
if (plot) plotem(y, z, pars, ...)
list(pars = pars, z = z, iter = i)
}

```

We source the function and generate 100 observations from a mixture of 2 bivariate normal distributions.

```

> source("prac-em2mvnorm.R")
> y1 <- rmvnorm(50, c(-3, -3), diag(2))
> y2 <- rmvnorm(50, c(2, 2), matrix(c(1, 0.8, 0.8, 1), nrow = 2))
> y <- rbind(y1, y2)

```

For initialization we are using the following parameter values:

```

> pars <- list(pi = 0.5,
+             mu1 = c(0.1, 0.1),
+             mu2 = c(-0.1, -0.1),
+             Sigma1 = diag(2),
+             Sigma2 = diag(2))

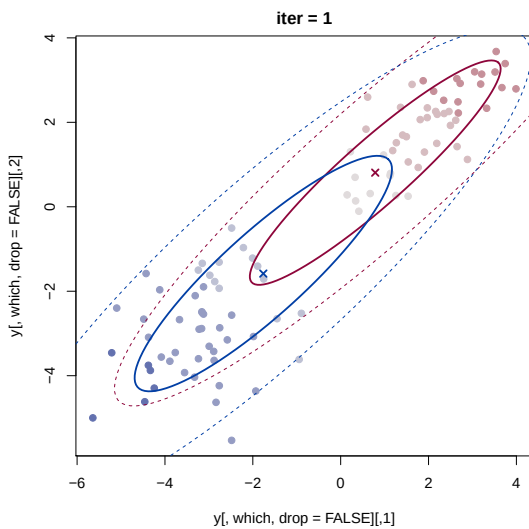
```

The development of the estimated parameter values over the iterations of the EM algorithm are visualized in the following.

```

> z1 <- em2mvnorm(y, pars, iter.max = 1, plot = TRUE, main = "iter = 1" )

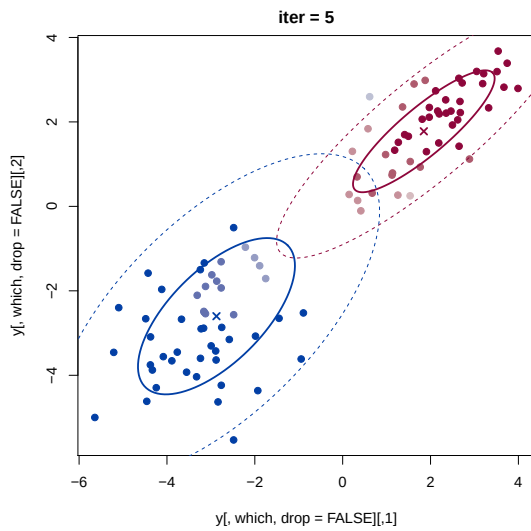
```



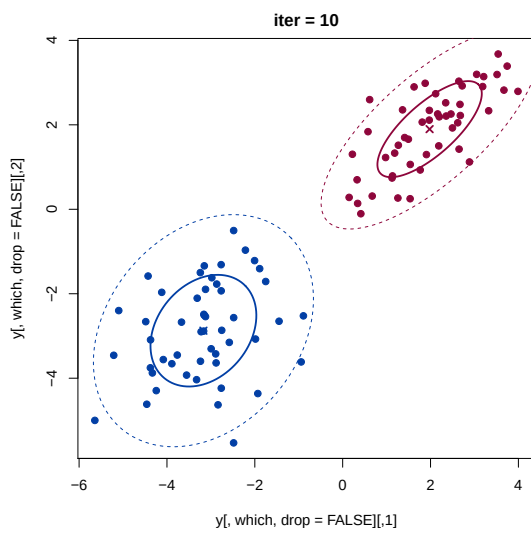
```

> z5 <- em2mvnorm(y, pars, iter.max = 5, plot = TRUE, main = "iter = 5" )

```



```
> z10 <- em2mvnorm(y, pars, iter.max = 10, plot = TRUE, main = "iter = 10" )
```



- Vary the initial values. How does the convergence behavior of the EM algorithm change?
- Draw observations from another mixture of 2 multivariate normal distributions.

How does the convergence behavior of the EM algorithm change if the component sizes, the mean values and the variances are modified?