Chapter 5

Visualization of Multivariate Data

Packages used in these exercises include bootstrap, DAAG, ISLR, lattice, MASS, and FactoMineR.

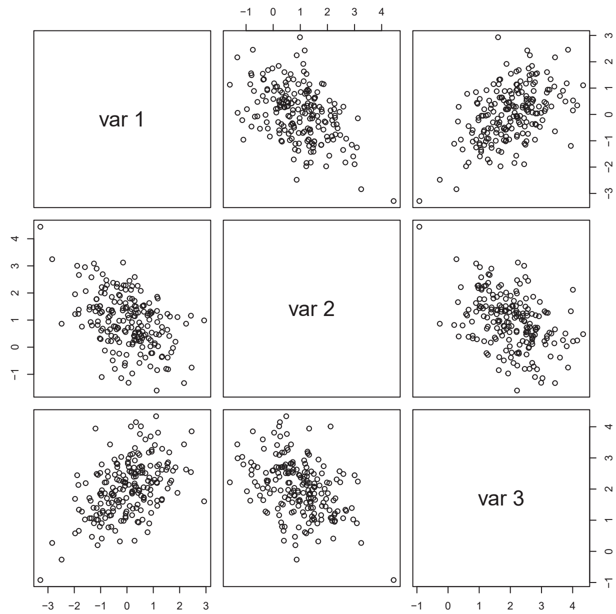
5.1 *Generate 200 random observations from the multivariate normal distribution having mean vector* $\mu = (0, 1, 2)$ *and covariance matrix*

$$\Sigma = \begin{bmatrix} 1.0 & -0.5 & 0.5 \\ -0.5 & 1.0 & -0.5 \\ 0.5 & -0.5 & 1.0 \end{bmatrix}.$$

Use any of the functions `rmvn.eigen`, `rmvn.Choleski` from Chapter 3, `mvrnorm (MASS)` or `rmvnorm (mvtnorm)`.

```
    library(MASS)
    Sigma <- matrix(c(1, -.5, .5, -.5, 1, -.5, .5, -.5, 1),
                    3, 3)
    mu <- c(0, 1, 2)
    x <- mvrnorm(200, mu, Sigma)
    colMeans(x)

## [1] 0.02900539 0.95726516 2.02953589

    cor(x)

##            [,1]       [,2]       [,3]
## [1,]  1.0000000 -0.4478050  0.4856554
## [2,] -0.4478050  1.0000000 -0.4107898
## [3,]  0.4856554 -0.4107898  1.0000000
```

```r
pairs(x)
```



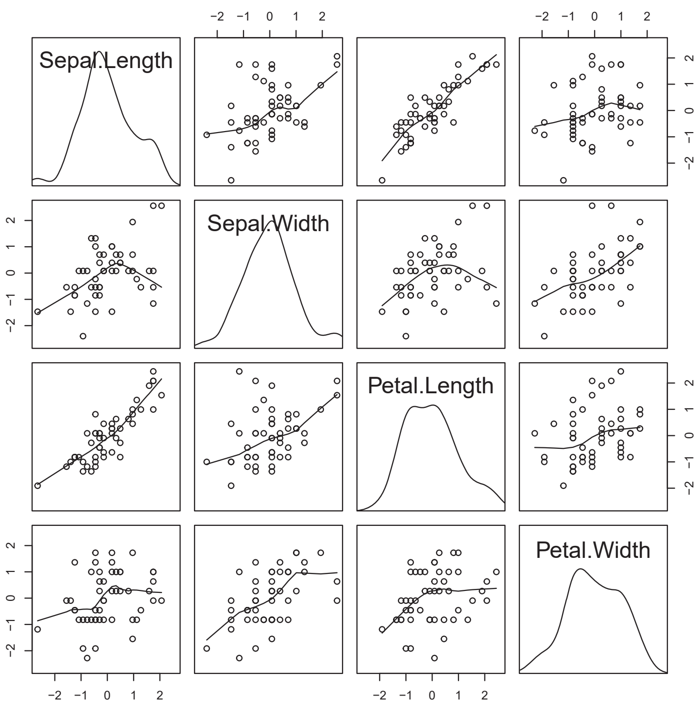
From the pairs plot it appears that the parameters for each plot approximately agree with the parameters of the corresponding bivariate distributions.

5.2 *Add a fitted smooth curve to each of the iris virginica scatterplots.*

The panel function below is similar to `panel.smooth`, with the options for color removed.

```r
panel.d <- function(x, ...) {
    usr <- par("usr")
    on.exit(par(usr))
    par(usr = c(usr[1:2], 0, .5))
    lines(density(x))
}

panel.sm <- function (x, y, bg = NA, pch = par("pch"),
  cex = 1, span = 2/3, iter = 3, ...) {
  points(x, y, pch = pch, bg = bg, cex = cex)
  ok <- is.finite(x) & is.finite(y)
  if (any(ok))
```

```
        lines(stats::lowess(x[ok], y[ok],
                            f = span, iter = iter), ...)
}

x <- scale(iris[101:150, 1:4])
r <- range(x)
pairs(x, panel = panel.sm, diag.panel = panel.d,
      xlim = r, ylim = r)
```



5.3 *The random variables X and Y are independent and identically distributed with normal mixture distributions. The components of the mixture have $N(0,1)$ and $N(3,1)$ distributions with mixing probabilities p_1 and $p_2 = 1-p_1$ respectively.*

The code below generates a bivariate random sample from the joint distribution of (X, Y).

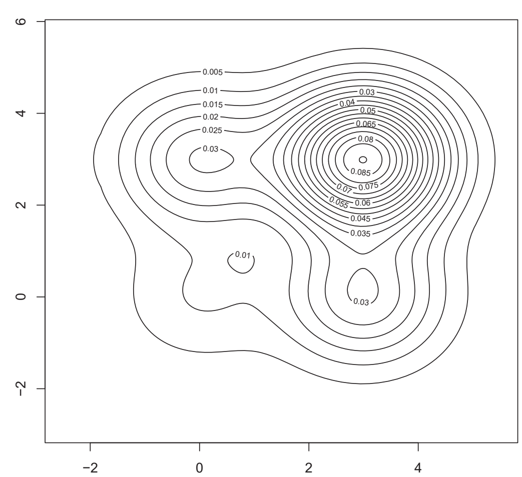
```
n <- 500
mu <- c(0, 3)
p <- .25
```

```
m <- sample(mu, size=2*n, replace=TRUE, prob=c(p, 1-p))
X <- matrix(rnorm(2*n, m), n, 2)
```

For the contour plot, we need the joint density. The random variables are independent so the joint density is the product of the marginals. (If dependent, cannot sort X and Y independently.)

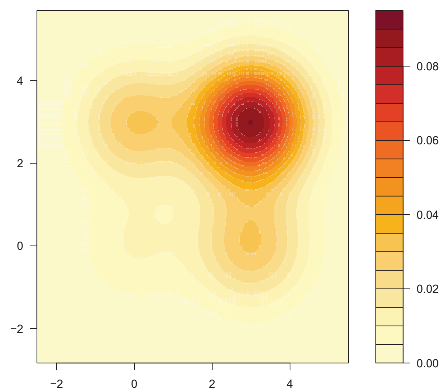
(Generally, when the joint density is available, we would not construct the contour plot from a sample, because we can generate the grid of points directly. When the joint density is not available, a density estimate can provide the z values.)

```
f <- function(x, y) {
    f1 <- p * dnorm(x, mu[1]) + (1-p) * dnorm(x, mu[2])
    f2 <- p * dnorm(y, mu[1]) + (1-p) * dnorm(y, mu[2])
    f1 * f2
}

x <- sort(X[,1])
y <- sort(X[,2])
z <- outer(x, y, f)
contour(x, y, z, nlevels=20)
```



5.4 *Construct a filled contour plot of the bivariate mixture in Exercise 5.3.*

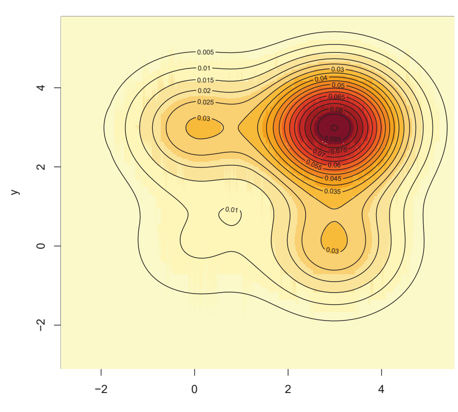
```r
filled.contour(x, y, z)  #default colors
```




```r
## other colors
#filled.contour(x, y, z, col=topo.colors(20))
#filled.contour(x, y, z,
#                col=gray(seq(.99,.01,length=20))) #no color
```

Another version using image and contour:

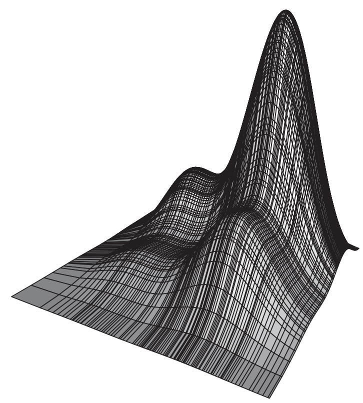
```r
image(x, y, z)
contour(x, y, z, nlevels=20, add=TRUE)
```



5.5 *Construct a surface plot of the bivariate mixture in Exercise 5.3.*

(First "thinning out" the data because the perspective plot turns out to be quite dark for the printed version.)

```
i <- seq(1, n, 5)
u <- x[i]
v <- y[i]
w <- z[i, i]
persp(u, v, w, shade=TRUE, theta=30, ltheta=30, box=FALSE)
```

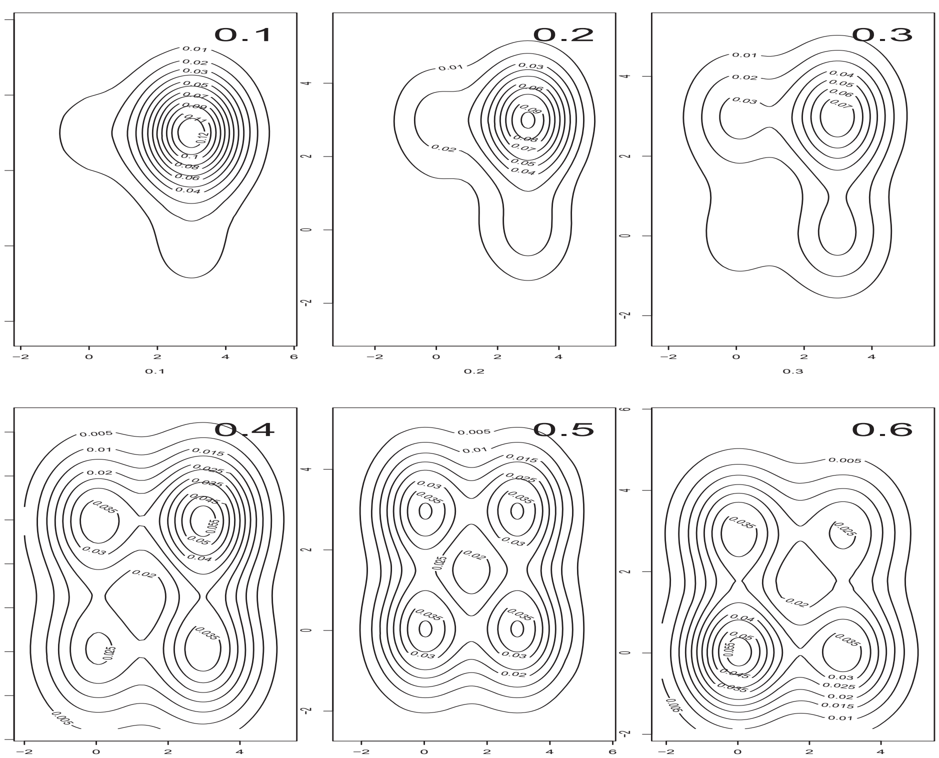


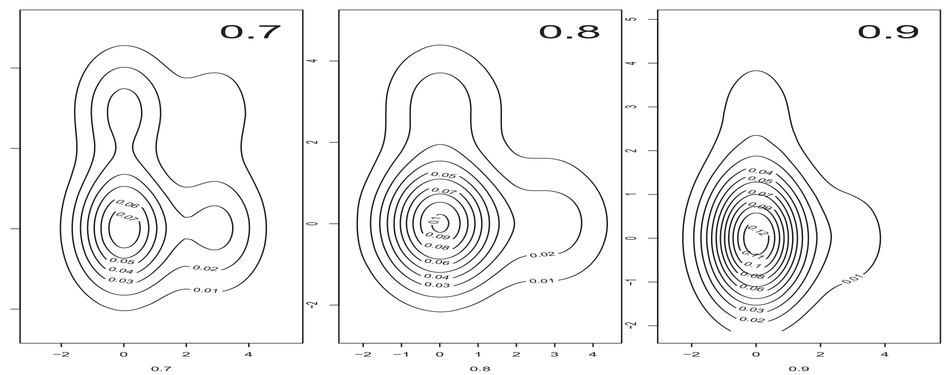
5.6 *Repeat Exercise 5.3 for various different choices of the parameters of the mixture model, and compare the distributions through contour plots.*

```
n <- 200
mu <- c(0, 3)
pr <- seq(.1, .9, .1)

f <- function(x, y) {
    f1 <- p * dnorm(x, mu[1]) + (1-p) * dnorm(x, mu[2])
    f2 <- p * dnorm(y, mu[1]) + (1-p) * dnorm(y, mu[2])
    f1 * f2
}

x <- matrix(0, n, 9)
y <- matrix(0, n, 9)
```

```
z <- array(0, c(n, n, 9))
for (i in 1:9) {
    p <- pr[i]
    m <- sample(mu, size=2*n, replace=TRUE, prob=c(p, 1-p))
    X <- matrix(rnorm(2*n, m), n, 2)

    x[,i] <- sort(X[,1])
    y[,i] <- sort(X[,2])
    z[,,i] <- outer(x[,i], y[,i], f)
    }
```

```
def.par <- par(no.readonly=TRUE)
par(mfrow = c(1, 3), mar=c(5,1,2,1))
for (i in 1:9) {
  contour(x[,i], y[,i], z[,,i], nlevels=10, xlab=pr[i])
  legend("topright", paste(pr[i]), cex=3, bty="n")
}
```






```r
par(def.par)
```

It is clear that the contours of mixtures for $1-p$ are simply a rotation of the contours for p. As p or $1-p$ increases toward 0.5 the surface is less smooth and becomes multi-modal.