Exemplo - Stepwise com multicolinearidade

Fazendo a leitura dos dados do Hald.

```
suppressMessages(library(mombf))
data(hald)
```

Verificando dimensão dos dados.

```
print(hald)
```

```
##        [,1] [,2] [,3] [,4] [,5]
##  [1,]  78.5    7   26    6   60
##  [2,]  74.3    1   29   15   52
##  [3,] 104.3   11   56    8   20
##  [4,]  87.6   11   31    8   47
##  [5,]  95.9    7   52    6   33
##  [6,] 109.2   11   55    9   22
##  [7,] 102.7    3   71   17    6
##  [8,]  72.5    1   31   22   44
##  [9,]  93.1    2   54   18   22
## [10,] 115.9   21   47    4   26
## [11,]  83.8    1   40   23   34
## [12,] 113.3   11   66    9   12
## [13,] 109.4   10   68    8   12
```

Verificando informações presentes nesse banco de dados.

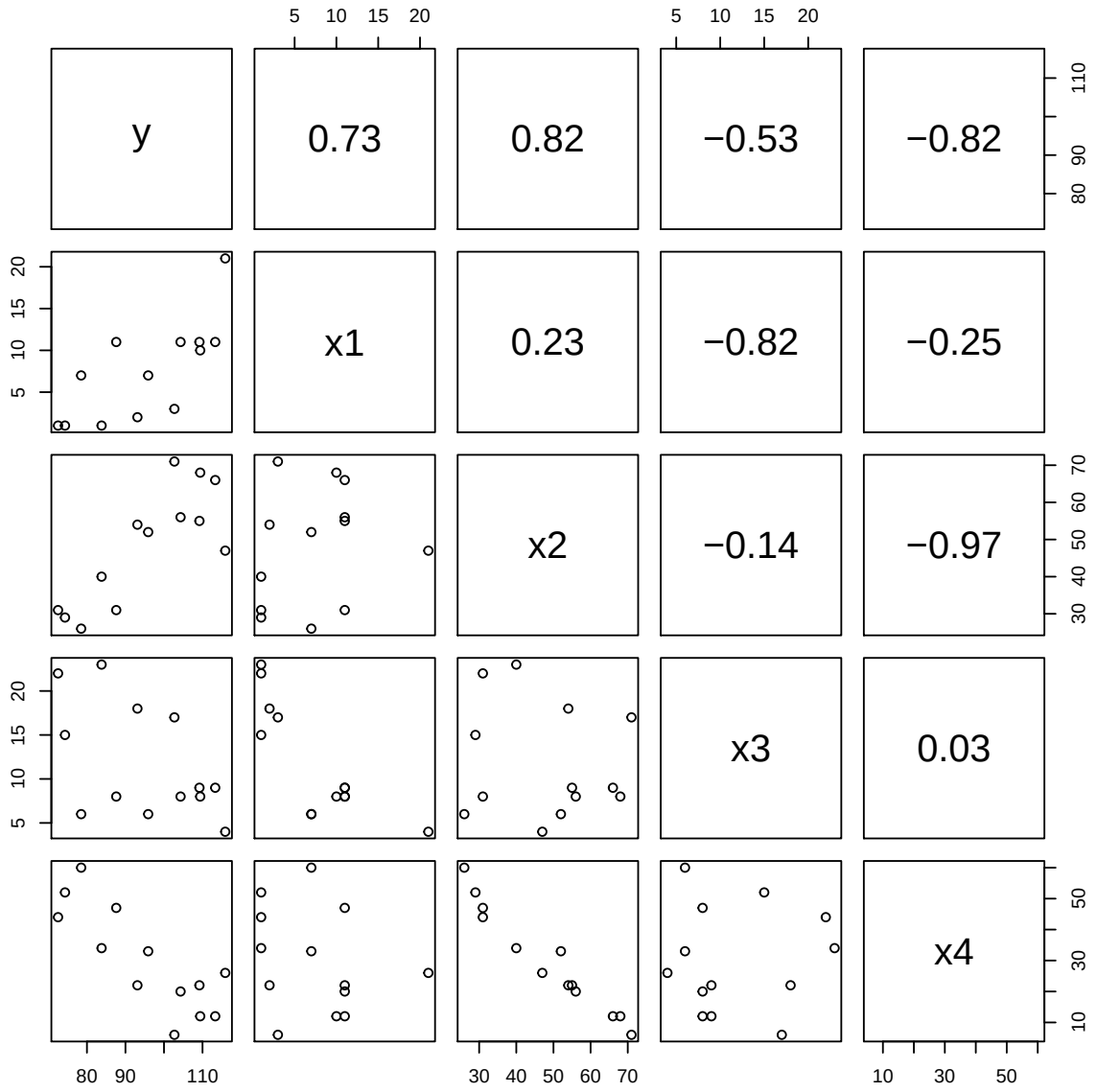
```
?hald
```

Definindo nomes das variáveis.

```
colnames(hald) <- c('y', 'x1', 'x2', 'x3', 'x4')
```

Gerando matriz com gráficos de dispersão das variáveis de interesse

```
panel.cor <- function(x, y, prefix = "", ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- cor(x, y)
    txt <- format(c(r, 0.123456789), digits = 2)[1]
    txt <- paste0(prefix, txt)
    text(0.5, 0.5, txt, cex=2)
}
pairs(hald, upper.panel = panel.cor)
```



O código para fazer o método stepwise no R pode ser encontrado em `http://orinanobworld.blogspot.ca/2011/02/stepwise-regression-in-r.html` e está reproduzido abaixo junto alguns comentários do autor.

```
stepwise <- function(full.model, initial.model, alpha.to.enter, alpha.to.leave) {
    full <- lm(full.model)
    msef <- (summary(full)$sigma)^2
    n <- length(full$residuals)
    allvars <- attr(full$terms, "predvars")
    current <- lm(initial.model)
    while (TRUE) {
        temp <- summary(current)
        rnames <- rownames(temp$coefficients)
        print(temp$coefficients)
        p <- dim(temp$coefficients)[1]
        mse <- (temp$sigma)^2
        cp <- (n - p) * mse/msef - (n - 2 * p)
        fit <- sprintf("\nS = %f, R-sq = %f, R-sq(adj) = %f, C-p = %f", temp$sigma,
            temp$r.squared, temp$adj.r.squared, cp)
        write(fit, file = "")
```

```
        write("=====", file = "")
        if (p > 1) {
            d <- drop1(current, test = "F")
            pmax <- max(d[-1, 6])
            if (pmax > alpha.to.leave) {
                var <- rownames(d)[d[, 6] == pmax]
                if (length(var) > 1) {
                  var <- var[2]
                }
                write(paste("--- Dropping", var, "\n"), file = "")
                f <- formula(current)
                f <- as.formula(paste(f[2], "~", paste(f[3], var, sep = " - ")))
                current <- lm(f)
                next
            }
        }
        a <- tryCatch(add1(current, scope = full, test = "F"), error = function(e) NULL)
        if (is.null(a)) {
            break
        }
        pmin <- min(a[-1, 6])
        if (pmin < alpha.to.enter) {
            var <- rownames(a)[a[, 6] == pmin]
            if (length(var) > 1) {
                var <- var[2]
            }
            write(paste("+++ Adding", var, "\n"), file = "")
            f <- formula(current)
            f <- as.formula(paste(f[2], "~", paste(f[3], var, sep = " + ")))
            current <- lm(f)
            next
        }
        break
    }
}
```

Utilizando a função com o banco de dados do Hald.

```
y <- hald[,1]
x1 <- hald[,2]
x2 <- hald[,3]
x3 <- hald[,4]
x4 <- hald[,5]

stepwise(full.model = y~x1+x2+x3+x4,
         initial.model = y~1,
         alpha.to.enter = 0.10,
         alpha.to.leave = 0.10)

##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 95.42308   4.172378 22.87019 2.899153e-11
##
## S = 15.043723, R-sq = 0.000000, R-sq(adj) = 0.000000, C-p = 442.916687
## =====
## +++ Adding x4
##
##                Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 117.5679312   5.262207 22.34195 1.624236e-10
## x4           -0.7381618   0.154596 -4.77478 5.762318e-04
```

```
##
## S = 8.963902, R-sq = 0.674542, R-sq(adj) = 0.644955, C-p = 138.730833
## =====
## +++ Adding x1
##
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 103.0973816 2.12398361  48.53963 3.324338e-13
## x4           -0.6139536 0.04864455 -12.62122 1.814890e-07
## x1            1.4399583 0.13841664  10.40307 1.105281e-06
##
## S = 2.734266, R-sq = 0.972471, R-sq(adj) = 0.966965, C-p = 5.495851
## =====
## +++ Adding x2
##
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 71.6483070 14.1423935  5.066208 6.753322e-04
## x4          -0.2365402  0.1732878 -1.365014 2.053954e-01
## x1           1.4519380  0.1169976 12.409981 5.780764e-07
## x2           0.4161098  0.1856105  2.241844 5.168735e-02
##
## S = 2.308745, R-sq = 0.982335, R-sq(adj) = 0.976447, C-p = 3.018233
## =====
## --- Dropping x4
##
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 52.5773489 2.28617433 22.99796 5.456571e-10
## x1           1.4683057 0.12130092 12.10465 2.692212e-07
## x2           0.6622505 0.04585472 14.44236 5.028960e-08
##
## S = 2.406335, R-sq = 0.978678, R-sq(adj) = 0.974414, C-p = 2.678242
## =====
```