> morris.fun

function (X)

{

w <- 2 \* (X - 0.5)

w[, c(3, 5, 7)] <- 2 \* (1.1 \* X[, c(3, 5, 7)]/(X[, c(3, 5,

7)] + 0.1) - 0.5)

y <- b0

for (i in 1:20) {

y <- y + b1[i] \* w[, i]

}

for (i in 1:19) {

for (j in (i + 1):20) {

y <- y + b2[i, j] \* w[, i] \* w[, j]

}

}

for (i in 1:18) {

for (j in (i + 1):19) {

for (k in (j + 1):20) {

y <- y + b3[i, j, k] \* w[, i] \* w[, j] \* w[,

k]

}

}

}

for (i in 1:17) {

for (j in (i + 1):18) {

for (k in (j + 1):19) {

for (l in (k + 1):20) {

y <- y + b4[i, j, k, l] \* w[, i] \* w[, j] \*

w[, k] \* w[, l]

}

}

}

}

y

}

-----------------------------------------------------------------------------

> morris

function (model = NULL, factors, r, design, binf = 0, bsup = 1,

scale = TRUE, ...)

{

if (is.character(factors)) {

X.labels <- factors

p <- length(X.labels)

}

else if (is.numeric(factors)) {

p <- factors

X.labels <- paste("X", 1:p, sep = "")

}

else {

stop("invalid argument 'factors', waiting for a scalar (number) or a character string vector (names)")

}

if (length(r) == 1) {

r.max <- r

}

else {

r.max <- r[2]

r <- r[1]

}

if (!"type" %in% names(design)) {

design$type <- "oat"

warning("argument 'design$type' not found, set at 'oat'")

}

if (design$type == "oat") {

if (!"levels" %in% names(design)) {

stop("argument 'design$levels' not found")

}

nl <- design$levels

if (length(nl) == 1)

nl <- rep(nl, p)

if ("grid.jump" %in% names(design)) {

jump <- design$grid.jump

if (round(jump, 0) != jump)

stop("grid.jump must be integer")

if (length(jump) == 1)

jump <- rep(jump, p)

}

else {

jump <- rep(1, p)

warning("argument 'design$grid.jump' not found, set at 1")

}

}

else if (design$type == "simplex") {

if (!"scale.factor" %in% names(design)) {

stop("argument 'design$scale.factor' not found")

}

h <- design$scale.factor

}

else {

stop("invalid argument design$type, waiting for \"oat\" or \"simplex\"")

}

if (length(binf) == 1)

binf <- rep(binf, p)

if (length(bsup) == 1)

bsup <- rep(bsup, p)

if (design$type == "oat") {

X <- random.oat(p, r.max, binf, bsup, nl, jump)

}

else if (design$type == "simplex") {

X <- random.simplexes(p, r.max, binf, bsup, h)

}

X.unique <- array(t(X), dim = c(p, p + 1, r.max))

X.unique <- unique(X.unique, MARGIN = 3)

X <- matrix(X.unique, ncol = p, byrow = TRUE)

colnames(X) <- X.labels

r.unique <- nrow(X)/(p + 1)

if (r.unique < r.max) {

warning(paste("keeping", r.unique, "repetitions out of",

r.max))

}

r.max <- r.unique

if (r < r.max) {

ind <- morris.maximin(X, r)

X <- X[sapply(ind, function(i) ind.rep(i, p)), ]

}

x <- list(model = model, factors = factors, r = r, design = design,

binf = binf, bsup = bsup, scale = scale, X = X, call = match.call())

class(x) <- "morris"

if (!is.null(x$model)) {

response(x, other\_types\_allowed = TRUE, ...)

tell(x)

}

return(x)

}