function (formula, data, method = "nearest", distance = "logit",

distance.options = list(), discard = "none", reestimate = FALSE,

...)

{

mcall <- match.call()

if (is.null(data))

stop("Dataframe must be specified", call. = FALSE)

if (!is.data.frame(data)) {

stop("Data must be a dataframe", call. = FALSE)

}

if (sum(is.na(data)) > 0)

stop("Missing values exist in the data")

ischar <- rep(0, dim(data)[2])

for (i in 1:dim(data)[2]) if (is.character(data[, i]))

data[, i] <- as.factor(data[, i])

if (!is.numeric(distance)) {

fn1 <- paste("distance2", distance, sep = "")

if (!exists(fn1))

stop(distance, "not supported.")

}

if (is.numeric(distance)) {

fn1 <- "distance2user"

}

fn2 <- paste("matchit2", method, sep = "")

if (!exists(fn2))

stop(method, "not supported.")

tryerror <- try(model.frame(formula), TRUE)

if (distance %in% c("GAMlogit", "GAMprobit", "GAMcloglog",

"GAMlog", "GAMcauchit")) {

library(mgcv)

tt <- terms(mgcv::interpret.gam(formula)$fake.formula)

}

else {

tt <- terms(formula)

}

attr(tt, "intercept") <- 0

mf <- model.frame(tt, data)

treat <- model.response(mf)

X <- model.matrix(tt, data = mf)

if (method == "exact") {

distance <- out1 <- discarded <- NULL

if (!is.null(distance))

warning("distance is set to `NULL' when exact matching is used.")

}

else if (is.numeric(distance)) {

out1 <- NULL

discarded <- discard(treat, distance, discard, X)

}

else {

if (is.null(distance.options$formula))

distance.options$formula <- formula

if (is.null(distance.options$data))

distance.options$data <- data

out1 <- do.call(fn1, distance.options)

discarded <- discard(treat, out1$distance, discard, X)

if (reestimate) {

distance.options$data <- data[!discarded, ]

distance.options$weights <- distance.options$weights[!discarded]

tmp <- out1

out1 <- do.call(fn1, distance.options)

tmp$distance[!discarded] <- out1$distance

out1$distance <- tmp$distance

}

distance <- out1$distance

}

if (fn1 == "distance2mahalanobis") {

is.full.mahalanobis <- TRUE

}

else {

is.full.mahalanobis <- FALSE

}

out2 <- do.call(fn2, list(treat, X, data, distance = distance,

discarded, is.full.mahalanobis = is.full.mahalanobis,

...))

if (fn1 == "distance2mahalanobis") {

distance[1:length(distance)] <- NA

class(out2) <- c("matchit.mahalanobis", "matchit")

}

out2$call <- mcall

out2$model <- out1$model

out2$formula <- formula

out2$treat <- treat

if (is.null(out2$X)) {

out2$X <- X

}

out2$distance <- distance

out2$discarded <- discarded

nn <- matrix(0, ncol = 2, nrow = 4)

nn[1, ] <- c(sum(out2$treat == 0), sum(out2$treat == 1))

nn[2, ] <- c(sum(out2$treat == 0 & out2$weights > 0), sum(out2$treat ==

1 & out2$weights > 0))

nn[3, ] <- c(sum(out2$treat == 0 & out2$weights == 0 & out2$discarded ==

0), sum(out2$treat == 1 & out2$weights == 0 & out2$discarded ==

0))

nn[4, ] <- c(sum(out2$treat == 0 & out2$weights == 0 & out2$discarded ==

1), sum(out2$treat == 1 & out2$weights == 0 & out2$discarded ==

1))

dimnames(nn) <- list(c("All", "Matched", "Unmatched", "Discarded"),

c("Control", "Treated"))

out2$nn <- nn

return(out2)

}