function (formula = NULL, data, sel = NULL, resp = NULL, treat = NULL,

...)

{

if (missing(data))

stop("Argument 'data' is missing.")

if (!is.null(formula)) {

if (dim(model.frame(formula, data))[2] == 2) {

warnings("Variables to be checked are missed.")

}

else {

name.sel <- names(model.frame(formula, data))[-c(1:2)]

sel <- as.data.frame(data[, name.sel])

name.resp <- names(model.frame(formula, data))[1]

resp <- data[, name.resp]

if (any(name.sel == name.resp))

stop("Argument 'sel' contains argument 'resp'.")

name.treat <- names(model.frame(formula, data))[2]

treat <- data[, name.treat]

if (any(name.sel == name.treat))

stop("Argument 'sel' contains argument 'treat'.")

}

}

else {

if (is.null(treat)) {

stop("Argument 'treat' is needed.")

}

else {

if (is.logical(treat)) {

stop("Argument 'treat' may not be logical.")

}

else {

if (is.character(treat) | is.numeric(treat)) {

A <- find.treat(data = data, treat = treat)

treat <- A[[1]]

name.treat <- A[[2]]

}

else {

stop("Argument 'treat' has to be either numeric or a string.")

}

}

}

if (is.null(resp)) {

stop("Argument 'resp' is needed.")

}

else {

if (is.logical(resp)) {

stop("Argument 'resp' may not be logical.")

}

else {

if (is.character(resp) | is.numeric(resp)) {

A <- find.resp(data = data, resp = resp)

resp <- A[[1]]

name.resp <- A[[2]]

}

else {

stop("Argument 'resp' has to be either numeric or a string.")

}

}

}

if (is.null(sel)) {

name.sel <- names(data)[which(names(data) != name.resp &

names(data) != name.treat)]

sel <- data[, name.sel]

}

else {

sel <- find.sel(data = data, sel = sel)

name.sel <- names(sel)

if (any(name.sel == name.treat))

stop("Argument 'sel' contains argument 'treat'.")

if (any(name.sel == name.resp))

stop("Argument 'sel' contains argument 'resp'.")

}

}

fam <- ifelse(nlevels(as.factor(resp)) == 2, "binomial",

"gaussian")

if (nlevels(as.factor(treat)) != 2)

stop(paste("Argument 'treat'=", name.treat, " has more than two values",

sep = ""))

eff.cov <- rel.eff <- vector(length = dim(sel)[2])

names(eff.cov) <- names(rel.eff) <- name.sel

null.model <- glm(resp ~ treat, family = fam, ...)

eff.treat <- as.numeric(null.model$coeff[2])

for (i in 1:dim(sel)[2]) {

cov <- sel[, i]

cov.model <- glm(resp ~ treat + cov, family = fam, ...)

eff.cov[i] <- as.numeric(cov.model$coeff[2])

if (fam == "binomial") {

rel.eff[i] <- 100 \* (abs((exp(eff.cov[i]) - exp(eff.treat))/exp(eff.treat)))

}

else {

rel.eff[i] <- 100 \* (abs((eff.cov[i] - eff.treat)/eff.treat))

}

}

output <- list(unadj.treat = eff.treat, adj.treat.cov = eff.cov,

rel.eff.treat = rel.eff, name.treat = name.treat, name.resp = name.resp,

name.sel = name.sel, family = fam)

class(output) <- c("relative.effect")

return(output)

}