model2rjfun.R

john

Tue Sep 2 11:44:06 2014

```
model2rjfun <- function(modelformula, pvec, data = NULL, jacobian = TRUE,
                        testresult = TRUE) {

    cat("model2rjfun: modelformula = ")
    print(modelformula)
    print(class(modelformula))

    stopifnot(inherits(modelformula, "formula"))

    if (length(modelformula) == 2) {
        residexpr <- modelformula[[2]]
    } else if (length(modelformula) == 3) {
        residexpr <- call("-", modelformula[[3]], modelformula[[2]])
    } else stop("Unrecognized formula")

    if (is.null(names(pvec)))
    names(pvec) <- paste0("p", seq_along(pvec))

    if (jacobian)
        residexpr <- deriv(residexpr, names(pvec))
## SHOULD TRY:
##  residexpr <- fnDeriv(residexpr, names(pvec))

    if (is.null(data))
    data <- environment(modelformula)
    else if (is.list(data))
    data <- list2env(data, parent = environment(modelformula))
    else if (!is.environment(data))
    stop("'data' must be a dataframe, list, or environment")
## ??140730JN: Why do we have no data=something here? We cannot change data
##   for the function.
    rjfun <- function(prm) {
        if (is.null(names(prm)))
        names(prm) <- names(pvec)
    localdata <- list2env(as.list(prm), parent = data)
    eval(residexpr, envir = localdata)
        # Saves Jacobian matrix as "gradient" attribute (consistent with deriv())
    }

    if (testresult) {
    resids <- rjfun(pvec)
    if (any(bad <- !is.finite(resids)))
        stop("residuals contain ", unique(resids[bad]))
    if (jacobian && any(bad <- !is.finite(attr(resids, "gradient"))))
        stop("Jacobian contains ", unique(attr(resids, "gradient")[bad]))
    rm(resids, bad)  # Don't want to capture these in the environment of rjfun
    }
```

```
    rjfun
}

model2ssgrfun <- function(modelformula, pvec, data = NULL, gradient = TRUE,
                          testresult = TRUE) {
    rjfun <- model2rjfun(modelformula, pvec, data = data, jacobian = gradient,
                         testresult = testresult)

    function(prm) {
    resids <- rjfun(prm)
    ss <- as.numeric(crossprod(resids))
    if (gradient) {
        jacval <- attr(resids, "gradient")
        grval <- 2*as.numeric(crossprod(jacval, resids))
        attr(ss, "gradient") <- grval
    }
    ss
    }
}

modelexpr <- function(fun) {
    env <- environment(fun)
    if (exists("rjfun", env))
    env <- environment(env$rjfun)
    env$residexpr
}
```