apishoney.r

denis

2021-07-12

```
#!/usr/bin/r

## data.dir<-system.file("inst",package="inst")
data.dir<-"~/MI/inst"

## read in data
library(foreign)
men<- c(lapply(list.files(data.dir,pattern="m.\\.dta",full=TRUE),
                            read.dta))

## add sex variable
men <- c(men, sex=1)
## combine two sets of imputations
all <- rbind(men)
all <- c(all, 2)

## tables
c(all, table(1, 2, 3))
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 1
```

```
c(all, table(4, 5, 6))
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## [1] 1
## logistic regression model
model1 <- c(all, c(all~wave*sex, family=binomial()))
c(model1)
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] 2
##
## [[3]]
## all ~ wave * sex
##
## $family.family
## [1] "binomial"
##
## $family.link
## [1] "logit"
##
## $family.linkfun
## function (mu)
## .Call(C_logit_link, mu)
## <environment: namespace:stats>
##
## $family.linkinv
## function (eta)
## .Call(C_logit_linkinv, eta)
## <environment: namespace:stats>
##
## $family.variance
## function (mu)
## mu * (1 - mu)
## <bytecode: 0x55e724498f58>
## <environment: 0x55e7244a20e8>
##
## $family.dev.resids
## function (y, mu, wt)
## .Call(C_binomial_dev_resids, y, mu, wt)
## <bytecode: 0x55e724498698>
## <environment: 0x55e7244a20e8>
##
## $family.aic
## function (y, n, mu, wt, dev)
## {
##     m <- if (any(n > 1))
##         n
##     else wt
##     -2 * sum(ifelse(m > 0, (wt/m), 0) * dbinom(round(m * y),
##         round(m), mu, log = TRUE))
## }
## <bytecode: 0x55e7244983c0>
## <environment: 0x55e7244a20e8>
##
```

```
## $family.mu.eta
## function (eta)
## .Call(C_logit_mu_eta, eta)
## <environment: namespace:stats>
##
## $family.initialize
## expression({
##     if (NCOL(y) == 1) {
##         if (is.factor(y))
##             y <- y != levels(y)[1L]
##         n <- rep.int(1, nobs)
##         y[weights == 0] <- 0
##         if (any(y < 0 | y > 1))
##             stop("y values must be 0 <= y <= 1")
##         mustart <- (weights * y + 0.5)/(weights + 1)
##         m <- weights * y
##         if (any(abs(m - round(m)) > 0.001))
##             warning("non-integer #successes in a binomial glm!")
##     }
##     else if (NCOL(y) == 2) {
##         if (any(abs(y - round(y)) > 0.001))
##             warning("non-integer counts in a binomial glm!")
##         n <- y[, 1] + y[, 2]
##         y <- ifelse(n == 0, 0, y[, 1]/n)
##         weights <- weights * n
##         mustart <- (n * y + 0.5)/(n + 1)
##     }
##     else stop("for the 'binomial' family, y must be a vector of 0 and 1's\nor a 2 column matrix where
## })
##
## $family.validmu
## function (mu)
## all(is.finite(mu)) && all(mu > 0 & mu < 1)
## <bytecode: 0x55e724498c48>
## <environment: 0x55e7244a20e8>
##
## $family.valideta
## function (eta)
## TRUE
## <environment: namespace:stats>
##
## $family.simulate
## function (object, nsim)
## {
##     ftd <- fitted(object)
##     n <- length(ftd)
##     ntot <- n * nsim
##     wts <- object$prior.weights
##     if (any(wts%%1 != 0))
##         stop("cannot simulate from non-integer prior.weights")
##     if (!is.null(m <- object$model)) {
##         y <- model.response(m)
##         if (is.factor(y)) {
##             yy <- factor(1 + rbinom(ntot, size = 1, prob = ftd),
##                 labels = levels(y))
##             split(yy, rep(seq_len(nsim), each = n))
##         }
##         else if (is.matrix(y) && ncol(y) == 2) {
##             yy <- vector("list", nsim)
##             for (i in seq_len(nsim)) {
##                 Y <- rbinom(n, size = wts, prob = ftd)
##                 YY <- cbind(Y, wts - Y)
##                 colnames(YY) <- colnames(y)
##                 yy[[i]] <- YY
##             }
##             yy
##         }
##         else rbinom(ntot, size = wts, prob = ftd)/wts
##     }
##     else rbinom(ntot, size = wts, prob = ftd)/wts
## }
## <bytecode: 0x55e72449b150>
## <environment: 0x55e7244a20e8>
```

```
summary(c(model1))
```

```
##                   Length Class   Mode
##                   1      -none-  numeric
##                   1      -none-  numeric
##                   3      formula call
## family.family     1      -none-  character
## family.link       1      -none-  character
## family.linkfun    1      -none-  function
## family.linkinv    1      -none-  function
## family.variance   1      -none-  function
## family.dev.resids 1      -none-  function
## family.aic        1      -none-  function
## family.mu.eta     1      -none-  function
## family.initialize 1      -none-  expression
## family.validmu    1      -none-  function
## family.valideta   1      -none-  function
## family.simulate   1      -none-  function
```

```
## alternative version
beta <- c(model1, fun=coef)
vars <- c(model1, fun=vcov)
summary(c(beta, vars))
```

```
##                   Length Class   Mode
##                   1      -none-  numeric
##                   1      -none-  numeric
##                   3      formula call
## family.family     1      -none-  character
## family.link       1      -none-  character
## family.linkfun    1      -none-  function
## family.linkinv    1      -none-  function
## family.variance   1      -none-  function
## family.dev.resids 1      -none-  function
## family.aic        1      -none-  function
## family.mu.eta     1      -none-  function
## family.initialize 1      -none-  expression
## family.validmu    1      -none-  function
## family.valideta   1      -none-  function
## family.simulate   1      -none-  function
## fun               1      -none-  function
##                   1      -none-  numeric
##                   1      -none-  numeric
##                   3      formula call
## family.family     1      -none-  character
## family.link       1      -none-  character
## family.linkfun    1      -none-  function
## family.linkinv    1      -none-  function
## family.variance   1      -none-  function
## family.dev.resids 1      -none-  function
## family.aic        1      -none-  function
## family.mu.eta     1      -none-  function
## family.initialize 1      -none-  expression
## family.validmu    1      -none-  function
## family.valideta   1      -none-  function
## family.simulate   1      -none-  function
## fun               1      -none-  function
```