

CORRELATED MULTINOMIAL DATA

ANIKO SZABO

ABSTRACT. We define a class for describing data from toxicology experiments with multinomial outcomes and implement fitting of a variety of existing models and trend tests.

1. DEFINING `CMDATA` – A CLASS FOR CLUSTERED MULTINOMIAL Data

We start with defining an S3 class describing data from toxicology experiments with multinomial outcomes of type $1, 2, \dots, K + 1$. Here K denotes the “degrees of freedom” of the outcome. $K = 1$ corresponds to binary data. The class is a data frame with the following columns:

Trt: a factor defining (treatment) groups

ClusterSize: an integer-valued variable defining the cluster size

NResp.1–NResp.K+1: integer-valued variables defining the number of responses of type $1, 2, \dots, K + 1$

Freq: an integer-valued variable defining frequency for each Trt/ClusterSize/NResp.1/.../NResp.K combination

While having all the counts and the clustersize is somewhat redundant (sum of counts = clustersize), this can be helpful for many computations.

We are using some generics defined for `CBData` objects, so we refer to that here to ensure correct collation order.

```
"..\R\CMData.R" 1≡
```

```
#'@include CBData.R
#'@include Reprod.R
◇
```

`CMData` converts a data frame to a `CMData` object. `x` is the input data frame; `trt`, `clustersize`, and `freq` could be strings or column indices defining the appropriate variable in `x`. `nresp` should be a vector of variable names or column indices of length K (if `clustersize` is given) or $K + 1$ (in which case `clustersize` will be calculated). `freq` can also be `NULL`, in which case it is assumed that each combination has frequency 1.

```
"../R/CMData.R" 2≡
```

```
#'Create a 'CMdata' object from a data frame.
#
#The \code{CMData} function creates an object of class \dfn{CMData} that is
#used in further analyses. It identifies the variables that define treatment
#group, clustersize and the number of responses for each outcome type.
#
# @export
# @param x a data frame with one row representing a cluster or potentially a
# set of clusters of the same size and number of responses for each outcome
# @param trt the name of the variable that defines treatment group
# @param nresp either a character vector with the names or a numeric vector with indices
# of the variables that define the number of responses in
# the cluster for each outcome type. If \code{clustersize} is \code{NULL}, then it will be
# calculated assuming that the \code{nresp} vector contains all the possible outcomes.
# If \code{clustersize} is given, then an additional category is created for the excess cluster members.
```

Date: December 23, 2014.

```

#'@param clustersize the name or index of the variable that defines cluster size, or \code{NULL}. If \code{NULL}
#'its value will be calculated by adding the counts from the \code{nresp} variables. If defined,
#'an additional response type will be created for the excess cluster members.
#'@param freq the name or index of the variable that defines the number of clusters
#'represented by the data row. If \code{NULL}, then each row is assumed to
#'correspond to one cluster.
#'@return A data frame with each row representing all the clusters with the
#'same trt/size/number of responses, and standardized variable names:
#'@return \item{Trt}{factor, the treatment group}
#'@return \item{ClusterSize}{numeric, the cluster size}
#'@return \item{NResp.1--NResp.K+1}{numeric, the number of responses for each of the K+1 outcome types}
#'@return \item{Freq}{numeric, number of clusters with the same values}
#'@author Aniko Szabo
#'@seealso \code{\link{read.CMDData}} for creating a \code{CMDData} object
#'directly from a file.
#'@keywords classes manip
#'@examples
#
#data(dehp)
#dehp <- CMDData(dehp, trt="Trt", nresp=c("NResp.1", "NResp.2", "NResp.3"))
#str(dehp)
#
◇

```

File defined by [2](#), [3a](#), [4abc](#), [5](#), [?](#).

Uses: `read.CMDData` [4a](#).

"../R/CMDData.R" 3a≡

```

CMDData <- function(x, trt, nresp, clustersize=NULL, freq=NULL){
  if (!is.data.frame(x)) stop("x has to be a data frame")
  nms <- names(x)
  K <- if (is.null(clustersize)) length(nresp)-1 else length(nresp)

  process.var <- function(var){
    if (is.character(var)){
      if (var %in% nms) res <- x[[var]]
      else stop(paste("Variable '", var, "' not found"))
    }
    else {
      if (is.numeric(var)){
        if (var %in% seq(along=nms)) res <- x[[var]]
        else stop(paste("Column", var, " not found"))
      }
      else stop(paste("Invalid variable specification:", var))
    }
  }

  trtvar <- factor(process.var(trt))
  nrespvar <- sapply(nresp, process.var)
  if (is.null(freq)) freqvar <- rep(1, nrow(x))
  else freqvar <- process.var(freq)

  if (!is.null(clustersize)){
    csvar <- process.var(clustersize) # read cluster sizes
    nrespvar <- cbind(nrespvar, csvar - rowSums(nrespvar)) # calculate last category
  }
  else {
    csvar <- rowSums(nrespvar) #calculate sample sizes
  }
}

```

```
colnames(nrespvar) <- 1:(K+1)

d <- data.frame(Trt=trtvar, ClusterSize=csvar, NResp=nrespvar, Freq=freqvar)
nrespnames <- grep("NResp", names(d), value=TRUE)
d <- aggregate(d$Freq, d[,c("Trt", "ClusterSize", nrespnames)], sum)
names(d)[length(names(d))] <- "Freq"

attr(d, "ncat") <- K+1
class(d) <- c("CMDData", "data.frame")
d}
```

File defined by 2, 3ab, 4abc, 5, ?.
Defines: CMdata.data.frame Never used.

The `read.CMData` function reads in clustered multinomial data from a tab-delimited text file. There are two basic data format options: either the counts of responses of all categories are given (and the cluster size is the sum of these counts), or the total cluster size is given with the counts of all but one category. The first column should always give the treatment group, then either the counts for each category (first option, chosen by setting `with.clustersize = FALSE`), or the size of the cluster followed by the counts for all but one category (second option, chosen by setting `with.clustersize = TRUE`). Optionally, a last column could give the number of times the given combination occurs in the data.

"../R/CMDData.R" 3b≡

```
#'Read data from external file into a CMDData object
#
#'#A convenience function to read data from specially structured file directly
#'#into a CMDData object. There are two basic data format options: either the counts of responses of
#'#cluster size is the sum of these counts), or the total cluster size is given with the counts of all but one
#'#The first column should always give the treatment group, then either the counts for each category (first
#'#with.clustersize = FALSE), or the size of the cluster followed by the counts for all but one category
#'#chosen by setting with.clustersize = TRUE). Optionally, a last column could
#'#give the number of times the given combination occurs in the data.
#
#'#@export
#'#@param file name of file with data. The data in the file should be structured as described above.
#'#@param with.clustersize logical indicator of whether a cluster size variable is present
#'#in the file
#'#@param with.freq logical indicator of whether a frequency variable is present
#'#in the file
#'#@param ... additional arguments passed to \link[utils]{read.table}
#'#@return a CMDData object
#'#@author Aniko Szabo
#'#@seealso \link{CMDData}
#'#@keywords IO file
#
#'
```

File defined by 2, 3ab, 4abc, 5, ?.

"../R/CMDData.R" 4a≡

```
read.CMDData <- function(file, with.clustersize=TRUE, with.freq=TRUE, ...){
  d <- read.table(file, ...)
  K <- ncol(d) - with.freq - 2 #subtracting Trt & either ClusterSize or last category column

  if (with.clustersize){
    d2 <- CMDData(d, trt=1, clustersize=2, nresp=3:(K+2), freq=if (with.freq) "Freq" else NULL)
  }
  else {
    d2 <- CMDData(d, trt=1, nresp=2:(K+2), freq=if (with.freq) "Freq" else NULL)
  }
  d2}

```

File defined by 2, 3ab, 4abc, 5, ?.

Defines: read.CMDData 2.

The [.CMDData function defines subsetting of CMDData objects. If the subsetting is only affecting the rows, then the appropriate attributes are preserved, and the unused levels of Trt are dropped. Otherwise the returned object does not have a CMDData class anymore.

"../R/CMDData.R" 4b≡

```
#'@rdname Extract
#'#export

```

File defined by 2, 3ab, 4abc, 5, ?.

"../R/CMDData.R" 4c≡

```
".[.CMDData" <- function(x, i, j, drop){
  res <- NextMethod("[")
  if (NCOL(res) == ncol(x)){
    res <- ".[.data.frame"(x, i, )
    if (is.factor(res$Trt)) res$Trt <- droplevels(res$Trt)
    attr(res, "ncat") <- attr(x, "ncat")
    res
  }
  else {
    class(res) <- setdiff(class(res), "CMDData")
  }
  res
}

```

File defined by 2, 3ab, 4abc, 5, ?.

Defines: [.CMDData Never used.

unwrap.CMDData is a utility function that reformats a CMDData object so that each row is one observation (instead of one cluster). A new 'ID' variable is added to indicate clusters.

"../R/CMDData.R" 5≡

```
#'\code{unwrap.CMDData} is a utility function that reformats a CMDData object so
#'#that each row is one observation (instead of one or more clusters). A new
#'#'ID' variable is added to indicate clusters. This form can be useful for
#'#setting up the data for a different package.
#'#
#'#@rdname unwrap
#'#method unwrap CMDData

```

```

#'@export
#'@return For \code{unwrap.CMData}: a data frame with one row for each cluster element (having a multinom
#'outcome) with the following standardized column names
#'@return \item{Trt}{factor, the treatment group}
#'@return \item{ClusterSize}{numeric, the cluster size}
#'@return \item{ID}{factor, each level representing a different cluster}
#'@return \item{Resp}{numeric with integer values giving the response type of the cluster
#'element}
#'@examples
#'
#'data(dehp)
#'dehp.long <- unwrap(dehp)
#'head(dehp.long)
#'

```

File defined by 2, 3ab, 4abc, 5, ?.
 Uses: `unwrap.CMData` ?.

"../R/CMData.R" ?≡

```

unwrap.CMData <- function(object,...){
  #unwrap Freq variable
  freqs <- rep(1:nrow(object), object$Freq)
  cm1 <- object[freqs,]
  cm1$Freq <- NULL

  #create ID
  cm1$ID <- factor(1:nrow(cm1))

  ncat <- attr(object, "ncat")
  nrespvars <- paste("NResp", 1:ncat, sep=".")

  #reshape to have one row per category within cluster
  cm2 <- reshape(cm1, direction="long", varying=list(nrespvars), v.names="Count",
                 idvar="ID", timevar="Resp", times=1:ncat)

  #unwrap categories
  counts <- rep(1:nrow(cm2), cm2$Count)
  res <- cm2[counts,]
  res$Count <- NULL
  class(res) <- "data.frame"
  res <- res[order(res$ID),c("Trt","ID","ClusterSize","Resp")]
  rownames(res) <- NULL

  res
}

```

File defined by 2, 3ab, 4abc, 5, ?.
 Defines: `unwrap.CMData` 5.