The mitools Package

October 28, 2008

Title Tools for multiple imputation of missing data

Version 2.0

Author Thomas Lumley

Description Tools to perform analyses and combine results from multiple-imputation datasets.

Maintainer Thomas Lumley <tlumley@u.washington.edu>

Suggests RODBC, DBI

License GPL 2

R topics documented:

`MIcombine` *Multiple imputation inference*

Description

Combines results of analyses on multiply imputed data sets. A generic function with methods for `imputationResultList` objects and a default method. In addition to point estimates and variances, `MIcombine` computes Rubin's degrees-of-freedom estimate and rate of missing information.

Usage

```
MIcombine(results, ...)
## Default S3 method:
MIcombine(results,variances,call=sys.call(),df.complete=Inf,...)
## S3 method for class 'imputationResultList':
MIcombine(results,call=NULL,df.complete=Inf,...)
```

Arguments

`results`	A list of results from inference on separate imputed datasets
`variances`	If `results` is a list of parameter vectors, `variances` should be the corresponding variance-covariance matrices
`call`	A function call for labelling the results
`df.complete`	Complete-data degrees of freedom
`...`	Other arguments, not used

Details

The `results` argument in the default method may be either a list of parameter vectors or a list of objects that have `coef` and `vcov` methods. In the former case a list of variance-covariance matrices must be supplied as the second argument.

The complete-data degrees of freedom are used when a complete-data analysis would use a t-distribution rather than a Normal distribution for confidence intervals, such as some survey applications.

Value

An object of class `MIresult` with `summary` and `print` methods

References

put references to the literature/web site here

See Also

`MIextract`, `with.imputationList`

Examples

```
data(smi)
models<-with(smi, glm(drinkreg~wave*sex,family=binomial()))
summary(MIcombine(models))

betas<-MIextract(models,fun=coef)
vars<-MIextract(models, fun=vcov)
summary(MIcombine(betas,vars))
```

MIextract *Extract a parameter from a list of results*

Description

Used to extract parameter estimates and standard errors from lists produced by `with.imputationList`.

Usage

```
MIextract(results, expr, fun)
```

Arguments

`results`	A list of objects
`expr`	an expression
`fun`	a function of one argument

Details

If `expr` is supplied, it is evaluated in each element of `results`. Otherwise each element of `results` is passed as an argument to `fun`.

Value

A list

See Also

`with.imputationList`, `MIcombine`

Examples

```
data(smi)
models<-with(smi, glm(drinkreg~wave*sex,family=binomial()))

betas<-MIextract(models,fun=coef)
vars<-MIextract(models, fun=vcov)
summary(MIcombine(betas,vars))
```

`imputationList` *Constructor for imputationList objects*

Description

Create and update `imputationList` objects to be used as input to other `MI` routines.

Usage

```
imputationList(datasets,...)
## Default S3 method:
imputationList(datasets,...)
## S3 method for class 'character':
imputationList(datasets,dbtype,dbname,...)
## S3 method for class 'imputationList':
update(object,...)
## S3 method for class 'imputationList':
rbind(...)
## S3 method for class 'imputationList':
cbind(...)
```

Arguments

`datasets`	a list of data frames corresponding to the multiple imputations, or a list of names of database tables or views
`dbtype`	"ODBC" or a database driver name for `DBI::dbDriver()`
`dbname`	Name of the database
`object`	An object of class `imputationList`
`...`	Arguments `tag=expr` to `update` will create new variables `tag` by evaluating `expr` in each imputed dataset. Arguments to `imputationList()` are passed to the database driver

Details

When the arguments to `imputationList()` are character strings a database-based imputation list is created. This can be a database accessed through ODBC with the `RODBC` package or a database with a DBI-compatible driver. The `dbname` and `...` arguments are passed to `dbConnect()` or `odbcConnect()` to create a database connection. Data are read from the database as needed.

For a database-backed object the `update()` method creates variable definitions that are evaluated as the data are read, so that read-only access to the database is sufficient.

Value

An object of class `imputationList` or `DBimputationList`

Examples

```
data.dir<-system.file("dta",package="mitools")
files.men<-list.files(data.dir,pattern="m.\.dta$",full=TRUE)
men<-imputationList(lapply(files.men, foreign::read.dta))
files.women<-list.files(data.dir,pattern="f.\.dta$",full=TRUE)
women<-imputationList(lapply(files.women, foreign::read.dta))
men<-update(men, sex=1)
women<-update(women,sex=0)
all<-rbind(men,women)
all<-update(all, drinkreg=as.numeric(drkfre)>2)
all
```

`smi` *Multiple imputations*

Description

An `imputationList` object containing five imputations of data from the Victorian Adolescent Health Cohort Study.

Usage

```
data(smi)
```

Format

The underlying data are in a data frame with 1170 observations on the following 12 variables.

id a numeric vector

wave a numeric vector

mmetro a numeric vector

parsmk a numeric vector

drkfre a factor with levels `Non drinker` `not in last wk` `<3 days last wk` `>=3 days last wk`

alcdos a factor with levels `Non drinker` `not in last wk` `av <5units/drink_day` `av =>5units/drink_day`

alcdhi a numeric vector

smk a factor with levels `non/ex-smoker` `<6 days` `6/7 days`

cistot a numeric vector

mdrkfre a numeric vector

sex a numeric vector

drinkreg a logical vector

Source

Carlin, JB, Li, N, Greenwood, P, Coffey, C. (2003) "Tools for analysing multiple imputed datasets" The Stata Journal 3; 3: 1-20.

Examples

```
data(smi)
with(smi, table(sex, drkfre))
model1<-with(smi, glm(drinkreg~wave*sex, family=binomial()))
MIcombine(model1)
summary(MIcombine(model1))
```

---

with.imputationList

Evaluate an expression in multiple imputed datasets

Description

Performs a computation of each of imputed datasets in `data`

Usage

```
## S3 method for class 'imputationList':
with(data, expr, fun, ...)
```

Arguments

`data`	An `imputationList` object
`expr`	An expression
`fun`	A function taking a data frame argument
`...`	Other arguments, passed to `fun`

Details

If `expr` is supplied, evaluate it in each dataset in `data`; if `fun` is supplied, it is evaluated on each dataset. If all the results inherit from "`imputationResult`" the return value is an `imputationResultList` object, otherwise it is an ordinary list.

Value

Either a list or an `imputationResultList` object

See Also

`imputationList`

Examples

```
data(smi)
models<-with(smi, glm(drinkreg~wave*sex,family=binomial()))
tables<-with(smi, table(drkfre,sex))
with(smi, fun=summary)
```

Index