Package 'mitools'

October 13, 2022

Title Tools for Multiple Imputation of Missing Data

Version 2.4

Author Thomas Lumley

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

Maintainer Thomas Lumley <t.lumley@auckland.ac.nz>

Suggests RODBC, foreign

Imports DBI, methods, stats

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2019-04-26 05:00:03 UTC

R topics documented:

---

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

```
## Not run:
## CRAN doesn't like this example
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

## End(Not run)
```

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))
```

pisamaths	*Maths Performance Data from the PISA 2012 survey in New Zealand*

Description

Data on maths performance, gender, some problem-solving variables and some school resource variables. This is actually a weighted survey: see withPV.survey.design in the survey package for a better analyis.

Usage

```
data("pisamaths")
```

Format

A data frame with 4291 observations on the following 26 variables.

SCHOOLID School ID

CNT Country id: a factor with levels New Zealand

STRATUM a factor with levels NZL0101 NZL0102 NZL0202 NZL0203

OECD Is the country in the OECD?

STIDSTD Student ID

ST04Q01 Gender: a factor with levels Female Male

ST14Q02 Mother has university qualifications No Yes

ST18Q02 Father has university qualifications No Yes

`MATHEFF` Mathematics Self-Efficacy: numeric vector

`OPENPS` Mathematics Self-Efficacy: numeric vector

`PV1MATH`,`PV2MATH`,`PV3MATH`,`PV4MATH`,`PV5MATH` 'Plausible values' (multiple imputations) for maths performance

`W_FSTUWT` Design weight for student

`SC35Q02` Proportion of maths teachers with professional development in maths in past year

`PCGIRLS` Proportion of girls at the school

`PROPMA5A` Proportion of maths teachers with ISCED 5A (math major)

`ABGMATH` Does the school group maths students: a factor with levels `No ability grouping between any classes` `One of these forms of ability grouping between classes for s` `One of these forms of ability grouping for all classes`

`SMRATIO` Number of students per maths teacher

`W_FSCHWT` Design weight for school

`condwt` Design weight for student given school

Source

A subset extracted from the `PISA2012lite` R package, <https://github.com/pbiecek/PISA2012lite>

References

OECD (2013) PISA 2012 Assessment and Analytical Framework: Mathematics, Reading, Science, Problem Solving and Financial Literacy. OECD Publishing.

Examples

```
data(pisamaths)

means<-withPV(list(maths~PV1MATH+PV2MATH+PV3MATH+PV4MATH+PV5MATH), data=pisamaths,
       action= quote(by(maths, ST04Q01, mean)), rewrite=TRUE)
means

models<-withPV(list(maths~PV1MATH+PV2MATH+PV3MATH+PV4MATH+PV5MATH), data=pisamaths,
       action= quote(lm(maths~ST04Q01*PCGIRLS)), rewrite=TRUE)
summary(MIcombine(models))
```

---

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)
```

---

withPV   *Analyse plausible values in surveys*

Description

Repeats an analysis for each of a set of 'plausible values' in a data set, returning a list suitable for `MIcombine`. That is, the data set contains some sets of columns where each set are multiple imputations of the same variable. With `rewrite=TRUE`, the `action` is rewritten to reference each plausible value in turn; with coderewrite=FALSE a new data set is constructed for each plausible value, which is slower but more general.

Usage

```
withPV(mapping, data, action, rewrite=TRUE, ...)
## Default S3 method:
withPV(mapping, data, action, rewrite=TRUE,...)
```

Arguments

mapping	A formula or list of formulas describing each variable in the analysis that has plausible values. The left-hand side of the formula is the name to use in the analysis; the right-hand side gives the names in the dataset.
data	A data frame. Methods for `withPV` dispatch on this argument, so can be written for, eg, survey designs or out-of-memory datasets.
action	With `rewrite=TRUE`, a quoted expression specifying the analysis, or a function taking a data frame as its only argument. With `rewrite=FALSE`, A function taking a data frame as its only argument, or a quoted expression with `.DATA` referring to the newly-created data frame to be used.
rewrite	Rewrite `action` before evaluating it (versus constructing new data sets)
...	For methods

Value

A list of the results returned by each evaluation of `action`, with the call as an attribute.

Note

I would be interested in seeing naturally-occurring examples where `rewrite=TRUE` does not work

See Also

[pisamaths](#)

[with.imputationList](#)

Examples

```
data(pisamaths)

models<-withPV(list(maths~PV1MATH+PV2MATH+PV3MATH+PV4MATH+PV5MATH), data=pisamaths,
        action= quote(lm(maths~ ST04Q01*(PCGIRLS+SMRATIO)+MATHEFF+OPENPS,
        data=.DATA)),
        rewrite=FALSE
)

summary(MIcombine(models))

## equivalently
models2<-withPV(list(maths~PV1MATH+PV2MATH+PV3MATH+PV4MATH+PV5MATH), data=pisamaths,
        action=quote( lm(maths~ST04Q01*(PCGIRLS+SMRATIO)+MATHEFF+OPENPS)), rewrite=TRUE)


summary(MIcombine(models2))
```

Index