

The Amelia Package

July 28, 2008

Version 1.1-33

Date 2008-07-22

Title Amelia II: A Program for Missing Data

Author James Honaker <tercer@ucla.edu>, Gary King <king@harvard.edu>, Matthew Blackwell <blackwel@fas.harvard.edu>

Maintainer Matthew Blackwell <blackwel@fas.harvard.edu>

Depends R (>= 2.0.0), foreign

Description Amelia II “multiply imputes” missing data in a single cross-section (such as a survey), from a time series (like variables collected for each year in a country), or from a time-series-cross-sectional data set (such as collected by years for each of several countries). Amelia II implements our bootstrapping-based algorithm that gives essentially the same answers as the standard IP or EMis approaches, is usually considerably faster than existing approaches and can handle many more variables. Unlike Amelia I and other statistically rigorous imputation software, it virtually never crashes (but please let us know if you find to the contrary!). The program also generalizes existing approaches by allowing for trends in time series across observations within a cross-sectional unit, as well as priors that allow experts to incorporate beliefs they have about the values of missing cells in their data. Amelia II also includes useful diagnostics of the fit of multiple imputation models. The program works from the R command line or via a graphical user interface that does not require users to know R.

License GPL version 2 or newer

URL <http://gking.harvard.edu/amelia>

Suggests tcltk

R topics documented:

am-internal	2
amelia	2
amelia-package	6
AmeliaView	6
combine.output	7

Index**8**

am-internal	<i>Internal Amelia functions</i>
-------------	----------------------------------

Description

Internal or currently undocumented functions in Amelia

Author(s)

James Honaker, Gary King, Matt Blackwell

amelia	<i>AMELIA: Multiple Imputation of Incomplete Multivariate Data</i>
--------	--

Description

Runs the bootstrap EM algorithm on incomplete data and creates imputed datasets.

Usage

```
amelia (data, m=5, p2s=1, frontend=FALSE, idvars=NULL,
        ts=NULL, cs=NULL, polytime=NULL, intercs=FALSE,
        lags=NULL, leads=NULL, startvals=0, tolerance=0.0001,
        logs=NULL, sqrts=NULL, lgstc=NULL, noms=NULL, ords=NULL,
        incheck=TRUE, collect=FALSE, outname="outdata",
        write.out=TRUE, archive=TRUE, arglist=NULL, keep.data=TRUE,
        empri=NULL, casepri=NULL, priors=NULL, autopri=0.05, emburn=c(0,0),
        bounds=NULL, max.resample=100)
```

Arguments

data	an incomplete dataset, organized into either a data frame or a matrix.
m	the number of imputed datasets to create.
p2s	an integer value taking either 0 for no screen output, 1 for normal screen printing of iteration numbers, and 2 for detailed screen output. See "Details" for specifics on output when p2s=2.
frontend	a logical value used internally for the GUI.
idvars	a vector of column numbers or column names that indicates identification variables. These will be dropped from the analysis but copied into the imputed datasets.
ts	column number or variable name indicating the variable identifying time in time series data.

<code>cs</code>	column number or variable name indicating the cross section variable.
<code>polytime</code>	integer between 0 and 3 indicating what power of polynomial should be included in the imputation model to account for the effects of time. A setting of 0 would indicate constant levels, 1 would indicate linear time effects, 2 would indicate squared effects, and 3 would indicate cubic time effects.
<code>intercs</code>	a logical variable indicating if the time effects of <code>polytime</code> should vary across the cross-section.
<code>lags</code>	a vector of numbers or names indicating columns in the data that should have their lags included in the imputation model.
<code>leads</code>	a vector of numbers or names indicating columns in the data that should have their leads (future values) included in the imputation model.
<code>startvals</code>	starting values, 0 for the parameter matrix from listwise deletion, 1 for an identity matrix.
<code>tolerance</code>	the convergence threshold for the EM algorithm.
<code>logs</code>	a vector of column numbers or column names that refer to variables that require log-linear transformation.
<code>sqrts</code>	a vector of numbers or names indicating columns in the data that should be transformed by a square root function. Data in this column cannot be less than zero.
<code>lgstc</code>	a vector of numbers or names indicating columns in the data that should be transformed by a logistic function for proportional data. Data in this column must be between 0 and 1.
<code>noms</code>	a vector of numbers or names indicating columns in the data that are nominal variables.
<code>ords</code>	a vector of numbers or names indicating columns in the data that should be treated as ordinal variables.
<code>incheck</code>	a logical indicating whether or not the inputs to the function should be checked before running <code>amelia</code> . This should only be set to <code>FALSE</code> if you are extremely confident that your settings are non-problematic and you are trying to save computational time.
<code>collect</code>	a logical value indicating whether or not the garbage collection frequency should be increased during the imputation model. Only set this to <code>TRUE</code> if you are experiencing memory issues as it can significantly slow down the imputation process.
<code>outname</code>	a string indicating the prefix of the file to which Amelia will write the imputed datasets. You can also specify a path in front of the prefix if you do not wish your items stored in the working directory. The files will be written as <code>.csv</code> files.
<code>write.out</code>	a logical value indicating whether or not you wish to have Amelia write your imputed datasets as comma-separated value files. If <code>TRUE</code> , Amelia will use the <code>outname</code> argument as the file prefix.
<code>archive</code>	a logical variable indicating whether a replication archive should be saved. This archive includes all of the settings, the results of each imputation and some information about the convergence. The output will be saved as <code>'amarchive.R'</code> in your working directory.

<code>arglist</code>	an output list from the <code>amelia</code> function or from a saved session from <code>AmeliaView</code> . Values from this list take precedent over any individually set arguments. See the <code>Amelia</code> manual for more information.
<code>keep.data</code>	a logical value indicating whether or not to keep the imputed datasets after each imputation. Useful if the datasets are large and you wish to avoid keeping them in memory after they have been written to a file.
<code>empri</code>	number indicating level of the empirical (or ridge) prior. This prior shrinks the covariances of the data, but keeps the means and variances the same for problems of high missingness, small N 's or large correlations among the variables. Should be kept small; a reasonable upper bound is around 10% of the rows of the data.
<code>casepri</code>	indicator matrix of size $k \times k$ (where k is the number of cases) for the degree of similarity between two cases. For example, the <code>[2,3]</code> entry would indicate how similar cases 2 and 3 were. The indicators can be 0, 1, 2, or 3. Values should only appear in the upper triangle, as values in the lower triangle are ignored.
<code>priors</code>	a four or five column matrix containing the priors for either individual missing observations or variable-wide missing values. See "Details" for more information.
<code>autopri</code>	allows the EM chain to increase the empirical prior if the path strays into a nonpositive definite covariance matrix, up to a maximum empirical prior of the value of this argument times n , the number of observations. Must be between 0 and 1, and at zero this turns off this feature.
<code>emburn</code>	a numeric vector of length 2, where <code>emburn[1]</code> is the minimum EM chain length and <code>emburn[2]</code> is the maximum EM chain length. These are ignored if they are less than 1.
<code>bounds</code>	a three column matrix to hold logical bounds on the imputations. Each row of the matrix should be of the form <code>c(column.number, lower.bound, upper.bound)</code> . See Details below.
<code>max.resample</code>	an integer that specifies how many times <code>Amelia</code> should redraw the imputed values when trying to meet the logical constraints of <code>bounds</code> . After this value, imputed values are set to the bounds.

Details

Multiple imputation is a method for analyzing incomplete multivariate data. This function will take an incomplete dataset in either data frame or matrix form and return `m` imputed datasets with no missing values. The algorithm first bootstraps a sample dataset with the same dimensions as the original data, estimates the sufficient statistics (with priors if specified) by EM, and then imputes the missing values of sample. It repeats this process `m` times to produce the `m` complete datasets where the observed values are the same and the unobserved values are drawn from their posterior distributions.

You can provide `Amelia` with informational priors about the missing observations in your data. To specify priors, pass a four or five column matrix to the `priors` argument with each row specifying a different priors as such:

```
one.prior <- c(row, column, mean, standard deviation)
```

or,

```
one.prior <- c(row, column, minimum, maximum, confidence).
```

So, in the first and second column of the priors matrix should be the row and column number of the prior being set. In the other columns should either be the mean and standard deviation of the prior, or a minimum, maximum and confidence level for the prior. You must specify your priors all as distributions or all as confidence ranges. Note that ranges are converted to distributions, so setting a confidence of 1 will generate an error.

Setting a priors for the missing values of an entire variable is done in the same manner as above, but inputting a 0 for the row instead of the row number. If priors are set for both the entire variable and an individual observation, the individual prior takes precedence.

If each imputation is taking a long time to converge, you can increase the empirical prior, `empri`. This value has the effect of smoothing out the likelihood surface so that the EM algorithm can more easily find the maximum. It should be kept as low as possible and only used if needed.

Amelia assumes the data is distributed multivariate normal. There are a number of variables that can break this assumption. Usually, though, a transformation can make any variable roughly continuous and unbounded. We have included a number of commonly needed transformations for data. Note that the data will not be transformed in the output datasets and the transformation is simply useful for climbing the likelihood.

Please refer to the Amelia manual for more information on the function or the options.

Value

A list containing the imputed datasets in objects 1 through `m`. Thus, you can refer to any of the datasets by referencing `output[[i]]`, where `i` is the number of the dataset you wish to reference.

These datasets will be returned in the same format which you passed them. For example, if you passed a data frame to `amelia` you will have `m` data frames in the output list. If you passed a matrix, you will have `m` matrices in the output.

Other objects in the list:

<code>code</code>	return code for the function. 0 indicates a successful run of Amelia. Other codes refer to various problems in data or settings. Please refer to the error message and the Amelia manual for help with errors.
<code>message</code>	error message. Only appears if return code is not 0.
<code>amelia.args</code>	list of the arguments used in the imputation along with a few diagnostics on each imputation.
<code>thetas</code>	a matrix of the output parameter matrices used to generate the imputed datasets.

Author(s)

James Honaker, Gary King, Matt Blackwell

`amelia-package`*Amelia II: A Program for Missing Data*

Description

Uses a bootstrap+EM algorithm to impute missing values from a dataset and produces multiple output datasets for analysis.

Details

Package: `amelia`
Type: `Package`
Version: `1.0`
Date: `2006-03-03`
License: `See Manual`

You can use the package in one of two ways: either by invoking the `ameliagui()` command and running the program from a graphical interface or by loading in your data and then running the `amelia` function on the data.

If you use the GUI in Windows, makes sure that you run R under a Single Window Interface (SDI) as it will try to grab focus from the GUI if you don't.

Author(s)

James Honaker, Matthew Blackwell, Gary King

References

King, Gary; James Honaker, Anne Joseph, and Kenneth Scheve. "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation," *American Political Science Review*, Vol. 95, No. 1 (March, 2001): Pp. 49-69.

`AmeliaView`*Interactive GUI for Amelia*

Description

Brings up the AmeliaView graphical interface, which allows users to load datasets, manage options and run Amelia from a traditional windowed environment.

Usage

`AmeliaView`

`combine.output` *Combine Multiple Amelia Output Lists*

Description

This function combines output lists from multiple runs of Amelia, where each run used the same arguments. The result is one list, formatted as if Amelia had been run once.

Usage

```
combine.output(...)
```

Arguments

... a list of Amelia output lists from runs of Amelia with the same arguments except the number of imputations.

Details

This function is useful for combining the output from Amelia runs that occurred at different times or in different sessions of R. It assumes that the arguments given to the runs of Amelia are the same except for m , the number of imputations, and it uses the arguments from the first output list as the arguments for the combined output list.

Index

- *Topic **internal**
 - am-internal, 2
- *Topic **models**
 - amelia, 2
- *Topic **package**
 - amelia-package, 6
- *Topic **utilities**
 - AmeliaView, 6
 - combine.output, 7

am-internal, 2

am.resample (*am-internal*), 2

amcheck (*am-internal*), 2

amelia, 2

amelia-package, 6

amelia.impute (*am-internal*), 2

amelia.prep (*am-internal*), 2

ameliaEnv (*am-internal*), 2

ameliagui (*am-internal*), 2

ameliaTclSet (*am-internal*), 2

AmeliaView, 6

amstack (*am-internal*), 2

amsubset (*am-internal*), 2

amsweep (*am-internal*), 2

amtransform (*am-internal*), 2

amunstack (*am-internal*), 2

bootx (*am-internal*), 2

combine.output, 7

compare.density (*am-internal*), 2

disperse (*am-internal*), 2

emarch (*am-internal*), 2

emfred (*am-internal*), 2

frame.to.matrix (*am-internal*), 2

framemat (*am-internal*), 2

generatepriors (*am-internal*), 2

getAmelia (*am-internal*), 2

gethull (*am-internal*), 2

impfill (*am-internal*), 2

indxs (*am-internal*), 2

mpinv (*am-internal*), 2

nametonumber (*am-internal*), 2

overimpute (*am-internal*), 2

packr (*am-internal*), 2

putAmelia (*am-internal*), 2

rmvnorm (*am-internal*), 2

scalecenter (*am-internal*), 2

sigalert (*am-internal*), 2

startval (*am-internal*), 2

unscale (*am-internal*), 2

unsubset (*am-internal*), 2

untransform (*am-internal*), 2