

## Objects contained in R workspace DEf1b.Rdata

---

### Datasets

EA	Ethnographic Atlas dataset
EAkey	Ethnographic Atlas metadata file
EAfact	Ethnographic Atlas dataset with factor labels
EAcov	Ethnographic Atlas variable covariates for imputation
LRB	Binford forager dataset
LRBkey	Binford forager metadata file
LRBfact	Binford forager dataset with factor labels
LRBcov	Binford forager variable covariates for imputation
SCCS	Standard Cross-Cultural Sample dataset
SCCSkey	Standard Cross-Cultural Sample metadata file
SCCSfact	Standard Cross-Cultural Sample dataset with factor labels
SCCScov	Standard Cross-Cultural Sample variable covariates for imputation
WNAI	Western North American Indians dataset
WNAIkey	Western North American Indians metadata file
WNAIfact	Western North American Indians dataset with factor labels
WNAIcov	Western North American Indians variable covariates for imputation
llm	Matrix of linguistic proximities between all pairs of societies

### Undocumented functions

kln	auxiliary function that converts all variables in a dataframe to either numeric or character
mmgg	auxiliary function that cleans up output from aggregate() function
rnkd	auxiliary function that assigns ranks to values (1=lowest)
resc	auxiliary function that rescales a variable
spmang	auxiliary function that removes leading and trailing spaces from string
widen	auxiliary function that widens the range of a variable
chK	auxiliary function that finds some characteristics of variables in dataframe
chkpmc	auxiliary function that checks variables for high collinearity
gSimpStat	auxiliary function that obtains descriptive statistics for numeric variables in dataframe

### Documented functions

setDS	sets up environment to work with one of the four datasets (EA, LRB, SCCS, WNAI)
mkdummy	makes dummy variable and creates entry for it in metadata
addesc	adds or changes description of variable in metadata
fv4scale	helper function to find variables for use in a scale
doMI	creates multiple imputed datasets
mkyscale	makes a scale (composite index) from several similar variables
doOLS	estimates regression model using OLS with imputed datasets, including network lag term
doLogit	estimates regression model using logit with imputed datasets, including network lag term
CSVwrite	writes objects to csv format file
mkmappng	plots a variable on world map and writes a png format file

---

### Description

Prior to running any other function, one must select the particular ethnological dataset one is using. The function creates the appropriate weight matrices and other auxiliary files.

### Usage

```
setDS(dsname)
```

### Arguments

dsname name of ethnological dataset (one of : "SCCS", "LRB", "WNAI", "EA")

### Value

The function writes the following objects to the general environment, where they are accessible to the other functions.

cov Names of covariates to use during imputation step  
dx The selected ethnological dataset is now called dx  
dx The factor version of dx  
key A metadata file for dx  
wdd A geographic proximity weight matrix for the societies in dx  
wee An ecological similarity weight matrix for the societies in dx  
wll A linguistic proximity weight matrix for the societies in dx

### Details

### Note

### Author(s)

Anthon Eff Anthon.Eff@mtsu.edu

### Examples

```
setDS("SCCS")
```

### Description

The function makes a dummy variable from a variable, and creates a description which is used in `doOLS` output.

### Usage

```
mkdummy(varb, val, showname=TRUE)
```

### Arguments

`varb`      name of a variable  
`val`        the value of variable `vv` for which the dummy equals one.  
`showname`    should variable name and description print to the console?

### Value

The function returns a variable named `vv.dval`, which equals one when `vv==val`, and equals zero otherwise.

### Details

There are two reasons why one should use this function to create dummy variables. First, it makes it possible to use the predetermined set of best covariates, found in the auxiliary file “`cov`”, for multiple imputation in `doMI`. Second, the function will automatically append a description for the dummy variable to the key file, which is then available for use in `doOLS` output. The description is created using the variable name from the key file and the description of the value from the factor version of the ethnological dataset.

### Note

#### Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

### Examples

```
mkdummy("v245", 2)
```

---

addesc	Add a variable description to the key file
--------	--

---

### **Description**

The function adds a variable description to the key file. This is useful in cases where a new variable is created, whose description is not yet in the key file. The description is then available for use in `doOLS` output.

### **Usage**

```
addesc (nvbs, nvbsdes)
```

### **Arguments**

nvbs      name of variable  
nvbsdes   description of nvbs

### **Value**

The function appends the description to the key file.

### **Details**

### **Note**

### **Author(s)**

Anthon Eff      Anthon.Eff@mtsu.edu

### **Examples**

```
dx$valchild <- (dx$v473+dx$v474+dx$v475+dx$v476)  
addesc("valchild", "Degree to which society values children")
```

## Description

The function scans the metadata for keywords and returns a list of variable names that might be suitable for combining into a scale. Can be helpful in quickly identifying potential components, but care should be taken to eliminate those that are unsuitable.

## Usage

```
fv4scale(lookword, dropword=NULL, keepword=NULL, coreword=NULL, nmin=93,  
         minalpha=.7, chklevels=FALSE, verbose=TRUE)
```

## Arguments

lookword	keywords to look for in variable descriptions (from metadata)
dropword	if identified variables contain these keywords, then they should be dropped
keepword	keep only identified variables also containing these keywords
corewords	these are the most important keywords, keep only those correlating highly with this set
nmin	look only for variables with at least this many non-missing values
minalpha	minimum value of Cronbach's alpha for set of variables (those least conforming will be eliminated until this target is hit)
chklevels	should factor levels also be scanned for keywords (in addition to variable descriptions)?
verbose	should function write information about variables to console (can help in deciding which variables to keep).

## Value

The function returns a string of variable names.

## Details

The function should be used with caution. It provides only candidate variables, not necessarily the best variables, to include in a scale. The widest set of candidate variables can be found by setting `chklevels=TRUE`, which creates dummy variables for those variables containing a keyword within a factor level label. After identifying variables with keywords in `lookword`, retaining those meeting the `keepword` condition and dropping those meeting the `dropword` condition, the procedure will narrow down the set of retained variables further by looking at the covariances among the variables. It does this in two ways. First, if the `coreword` option is used, those variables containing the `coreword` keywords are compared to those not containing the `coreword` keywords, and of the latter set, only those correlating most strongly with the `coreword` set are retained. Second, Cronbach's alpha is calculated for the set of candidate variables, and if  $\alpha < \text{minalpha}$  then that variable is dropped that most increases alpha by being dropped. This procedure is repeated until  $\alpha \geq \text{minalpha}$ .

The function `fv4scale` is run on the original data `dx`, as created by the function `setDS`. The  $\alpha$  produced here is calculated using listwise deletion, and might be lower when a scale is created with multiply imputed data, using the function `mkscale`.

## Note

### Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

## Examples

```
# --finds SCCS variables related to female economic contribution--  
femecon<-fv4scale(lookword=c("market", "exchange", "wage", "trade", "subsistence",  
                             "goods", "product", "labor"), keepword=c("female", "women", "woman"),  
                 coreword=c("subsistence"), nmin=60, chklevels=TRUE, verbose=FALSE)
```

**Description**

The function produces multiply imputed datasets from an ethnological dataset, using methods from the *mice* package.

**Usage**

```
smi<-doMI(varbnames, nimp=10, maxit=7)
```

**Arguments**

**varbnames** names of variables to include in the imputed data.  
**nimp** the number of imputed datasets to create (default=10)  
**maxit** the number of iterations used to estimate imputed data (default=7).

**Value**

The function `doMI` returns a dataframe containing the number of imputed datasets specified by the `nimp` option. The datasets are stacked one atop the other, and indexed by the variable `".imp"`.

**Details**

This function imputes several new datasets, using covariates for each variable to create a conditional distribution of estimates for each missing value, and then replacing the missing value with a draw from the distribution; as a result, each of the imputed datasets will typically have slightly different values for the estimated cells. The key to successful imputation is to have good covariates for each variable. The auxiliary file `"cov"` lists the best covariates found in a lengthy specification search. For those variables with no covariates found in `"cov"` (such as user-created variables), the best covariates are selected from a set of variables with no missing values, including network lag variables (based on geographic distance, language, and ecology).

The first argument is a list of variable names—all of these must be found in the ethnological dataset (transformed variables must be added to the ethnological dataset prior to running `doMI`). These will be the data used in model building. One should include all data one thinks might be useful, including all transformed data, but no additional data. The second argument is the number of imputed datasets to create: between 5 and 10 imputed datasets are considered adequate, but there is no harm in choosing more; the default is 10. The third argument is the number of iterations to perform in creating each imputed dataset; the default is 7.

It is usually a good idea to take a look at the returned dataframe, to see what variables it contains. It will contain not only the variables listed in `varbnames`, but also a set of normalized (mean=0, sd=1) climate and ecology variables that will be used as exogenous variables in the function `doOLS`. In addition, all variables with at least three discrete values, and with a maximum absolute value less than 300, will have a squared variable also entered (the squared variables all have the suffix `"Sq"`). Finally, the data.frame contains a variable called `".imp"`, which identifies the imputed dataset, and a variable called `".id"` which gives the society name.

**Note**

Based on the methods proposed by Malcolm M. Dow and E. Anthon Eff.

**Author(s)**

Anthon Eff      Anthon.Eff@mtsu.edu

**Examples**

```
scnn<-c("v1649", "v1127", "v2137", "v1265")
smi<-doMI(scnn, nimp=10, maxit=7)
dim(smi) # dimensions of new dataframe smi
smi[1:2, ] # first two rows of new dataframe smi
```

## Description

The function calculates a scale from a multiply imputed dataset.

## Usage

```
mkscale(compvarbs, udnavn=NULL, impdata, type="LP", add.descrip=NULL)
```

## Arguments

`compvarbs` names of component variables to include in the scale.  
`udnavn` the name of the scale.  
`impdata` the name of the multiply imputed dataset containing component variables.  
`type` the method to use in calculating the scale (one of "LP", "mean", "pc1").  
`add.descrip` the description of the scale, to add to the metadata file.

## Value

`scales` a dataframe, with two values for each observation in the input data: the calculated scale, and the square of the scale.  
`stats` Cronbach's alpha for the scale components.  
`corrs` correlation between scale and scale components.

## Details

The function can calculate three different kinds of scales: 1) based on linear programming as described in Eff (2010); 2) the mean of the standardized values; 3) the first principal component of the standardized values. Those components that vary negatively with the total scale are multiplied by -1; all components are then standardized (mean=10, sd=1).

Output is a list that includes the scale itself, as well as some statistics to help assess whether the scale is performing as desired. The `corrs` object should be examined: all correlations between components and total scale are positive since those that originally correlated negatively were multiplied by -1. The column labeled "inv" indicates with a "-1" those components that were inverted. The column "levels" reports the factor level labels, and provides a way to understand what higher values of a variable mean. If one variable correlates with the total scale in a way inconsistent with the other variables, then one should try again to find good component variables.

## Note

Based on the methods proposed by Malcolm M. Dow and E. Anthon Eff.

Eff, E. A. (2010). A scale for markets and property using the Standard Cross-Cultural Sample: a linear programming approach. *World Cultures eJournal*. 17(2). Retrieved from: <http://escholarship.org/uc/item/12k7z4st>

## Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

## Examples

```
scnn<-c(femecon, "v1649", "v1127", "v2137", "v1265")
smi<-doMI(scnn, nimp=10, maxit=7)

fec<-mkscale(compvarbs="femecon", udnavn="femecon.lp", impdata=smi,
  type="LP", add.descrip="female economic contribution (LP scale)")
#--check reasonableness of scale--
fec$stats
fec$corrs

smi[,names(fec$scales)]<-fec$scales
```

### Description

The function estimates an unrestricted and restricted OLS model, with network lag term, providing common diagnostics.

### Usage

```
doOLS(MIdata, depvar, indpv, rindpv=NULL, othexog=NULL, dw=TRUE,  
      lw=TRUE, ew=FALSE, stepW=FALSE, relimp=FALSE, slmtests=FALSE, haustest=NULL,  
      boxcox=FALSE, getismat=FALSE, mean.data=TRUE, doboot=0)
```

### Arguments

MIdata	a multiply imputed dataset, created by the function doMI
depvar	the name of the dependent variable (must be in MIdata)
indpv	the names of the independent variables for the unrestricted model (must be in MIdata)
rindpv	names of restricted model independent variables (must be in indpv; when default of NULL is executed, the restricted model independent variables will be the same as the unrestricted model, minus the last variable)
othexog	names of additional exogenous variables (must be in MIdata; will be added to a list of 21 variables; default is NULL)
dw	Should geographic proximity be used in constructing composite weight matrix (default=TRUE)
lw	Should linguistic proximity be used in constructing composite weight matrix (default=TRUE)
ew	Should ecological proximity be used in constructing composite weight matrix (default=FALSE)
stepW	Should stepwise regression be done to show most-selected variables from unrestricted model (default=FALSE)
relimp	Should relative importance be calculated for independent variables of restricted model (default=FALSE)
slmtests	Should spatial error tests be run for the three weight matrices (default=FALSE)
haustest	Hausman tests (H0: variable exogenous) are run for each independent variable listed here (variable must be in the restricted model). Default of NULL runs no tests.
boxcox	When boxcox=TRUE, a Box-Cox transformation is applied to the dependent variable, to make residuals as normal as possible. Default is FALSE.
getismat	When getismat=TRUE, the distance weight matrix is modified in the way suggested by Getis and Aldstadt (2003). Default is FALSE.
mean.data	When mean.data=TRUE (the default), output file includes a dataframe with mean values (across imputations) of the unrestricted model variables for each society, as well as significant dfbeta scores for restricted model independent variables, and latitude and longitude. mean.data=FALSE returns the entire, unaggregated set of data.
doboot	Enter the number of bootstrap repetitions to calculate bootstrap standard errors. Legal values lie between 10 and 10,000. The default (doboot=0) does not calculate bootstrap standard errors.



## Value

Returns a list with 12 elements:

DependVarb	Description of dependent variable
URmodel	Coefficient estimates from the unrestricted model (includes standardized coefficients and VIFs). Two pvalues are given for $H_0: \beta = 0$ . One is the usual pvalue, the other ( <i>hcvpval</i> ) is heteroskedasticity consistent. If <i>stepkept</i> =TRUE, the table will also include the proportion of times a variable is retained in the model using stepwise regression.
Rmodel	Coefficient estimates from the restricted model. If <i>relimp</i> =TRUE, the $R^2$ assigned to each independent variable is shown here.
EndogeneityTests	Hausman tests ( $H_0$ : variable is exogeneous), with F-statistic for weak instruments (a rule of thumb is that the instrument is weak if the F-stat is below 10), and Sargan test ( $H_0$ : instrument is uncorrelated with second-stage 2SLS residuals).
Diagnostics	Regression diagnostics for the restricted model: RESET test ( $H_0$ : model has correct functional form); Wald test ( $H_0$ : appropriate variables dropped); Breusch-Pagan test ( $H_0$ : residuals homoskedastic); Shapiro-Wilkes test ( $H_0$ : residuals normal); Hausman test ( $H_0$ : $W_y$ is exogenous); Sargan test ( $H_0$ : residuals uncorrelated with instruments for $W_y$ ). If <i>slmtests</i> =TRUE, the LaGrange multiplier tests ( $H_0$ : spatial error model not appropriate) are reported here.
OtherStats	Other statistics: Composite weight matrix weights (see details); $R^2$ for restricted model and unrestricted model; number of imputations; number of observations; Fstat for weak instruments for $W_y$ .
DescripStatsImputedData	Descriptive statistics for variables in unrestricted model.
DescripStatsOriginalData	Descriptive statistics for variables in unrestricted model.
totry	Character string of variables that were most significant in the unrestricted model as well as additional variables that proved significant using the <i>add1</i> function on the restricted model.
didwell	Character string of variables that were most significant in the unrestricted model.
dfbetas	Influential observations for <i>dfbetas</i> (see details)
data	Data as used in the estimations. Observations with missing values of the dependent variable have been dropped.

## Details

Users can choose any of three kinds of proximity/similarity weight matrices for constructing a network lag term: geographic, linguistic, and ecological. In most cases, users should choose only geographic and linguistic (the defaults). The optimal composite weight matrix, constructed as the weighted sum of the chosen weight matrices, is that which returns the most significant LaGrange multiplier statistic on the unrestricted model without network lag term (i.e., the composite matrix that finds the most autocorrelated structure in the unrestricted model residuals). The network lag term is entered in each model as the variable “ $W_y$ ”.

The *dfbetas* are scaled changes in restricted model coefficient estimates caused by adding an observation to the restricted model. Negative values indicate that including that observation lowers the coefficient estimate; positive values indicate that inclusion raises the estimate. Only the most influential *dfbetas* are output.

The stepwise procedure can provide additional insight into which independent variables provide the best model fit. Since the imputed datasets differ slightly from each other, the variables selected by a stepwise procedure typically differ slightly for each imputed dataset. If the *stepW*=TRUE option is chosen, a column labeled “*stepkept*” will be added to the table reporting unrestricted model results. The column reports the proportion of times the independent variable was retained in the model by a stepwise procedure using both forward and backward selection.

The *add1* function tests whether the members of a list of variables prove significant when added singly to a model. The list of variables includes all numeric variables in the imputed dataset, as well as squared terms of variables currently in the unrestricted regression. Variables proving significant in over 80 percent of the  $m$  estimated models are returned in the character string “*totry*”.

Relative importance is a method of assigning  $R^2$  to each independent variable. The method repeatedly estimates a model, first with one independent variable, then with two, etc. and calculates the change in  $R^2$  as each variable is introduced. The order of entry is changed, and the process repeated, to consider all possible orders of entry. The relative importance measure is the average change in  $R^2$  when introducing an independent variable across all these different orders of entry. With large

numbers of independent variables, the calculations are prohibitively slow. Setting `relimp=TRUE` will calculate the relative importance of independent variables in the restricted model, and report these in the column labeled “*relimp*”.

Endogeneity is a recognized problem with network lag terms. The Hausman test for endogenous regressors is performed on  $Wy$ , which is replaced by an instrumental variable which is the fitted value from regressing  $Wy$  on the network lagged other independent variables. The instrumental variable should be highly correlated with the endogenous variable, but not correlated with the 2SLS second-stage residual. A test for the latter is the Sargan test, with  $H_0$ : residuals are uncorrelated with instruments. A test for the former is to calculate the F-statistic with  $H_0$ : the excluded instruments are irrelevant in the first-stage regression; the rule of thumb is that this “weak identification F-stat” should be larger than 10. Since the weak identification F-stat will be low if irrelevant instruments are chosen, a stepwise procedure is used to select among a set of possible instruments including both the network lagged independent variables and the climate and ecology variables.

All independent variables can be tested for endogeneity (squared variables are tested in their original form). For these, the potential instruments consist of the climate, location, and ecology variables, and stepwise regression is used to pick a significant subset. While these variables are certainly exogenous, they are unlikely to be good instruments, since finding good instruments is a process requiring a great deal of creativity and patience on the part of the econometrician, and is not something that can be automated. Thus, one should think carefully about variables that might serve as instruments for any variable one wishes to test for endogeneity, and include these in the `othexog=` option.

Heteroskedasticity biases the standard errors of estimated coefficients. If the Breusch-Pagan test rejects the null that errors are homoskedastic, one should use either the heteroskedasticity consistent p-values (*hcpval*) in the *URmodel* and *Rmodel* results, or the p-values from bootstrap standard errors. Bootstraps take a fairly long time to calculate, so one shouldn't set the number of repetitions too high; in most cases, good results can be obtained with `doboot=500`.

If the residuals are not normal, and introduction of new independent variables and functional form changes do not make them normal, one can use the Box-Cox transformation where the dependent variable  $y$  is now equal to  $(y^\lambda - 1)/\lambda$  and  $\lambda$  is chosen so as to make the residuals as normal as possible.

#### Note

Based on the methods proposed by Malcolm M. Dow and E. Anthon Eff.

Getis, A. and Aldstadt, J. (2002). Constructing the spatial weights matrix using a local statistic. *Geographical Analysis* 36:90-104.

#### Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

#### Examples

```
scnn<-c("valchild", "v1649", "v1127", "v2137", "v1265", "v245.d2")
smi<-doMI(scnn, nimp=10, maxit=7)

iv<-c("v1649", "v1127", "v2137", "v1265", "v245.d2")
riv<- c("v1649", "v1127", "v2137")

h<-doOLS(MIdata=smi, depvar="valchild", indpv=iv, rindpv=riv, othexog=NULL, dw=TRUE,
         lw=TRUE, ew=FALSE, stepW=FALSE, relimp=FALSE, slmtests=FALSE, haustest=NULL,
         boxcox=FALSE, getismat=FALSE, mean.data=TRUE, doboot=0)

# look at first 10 elements in h
h[1:10]
```

### Description

The function estimates an unrestricted and restricted logit model in a multiple imputation environment, with network lag term, providing common diagnostics.

### Usage

```
doLogit(MIdata, depvar, indpv, rindpv=NULL, dw=TRUE, lw=TRUE, ew=FALSE, doboot=500,
        mean.data=TRUE, getismat=FALSE, othexog=NULL)
```

### Arguments

MIdata	a multiply imputed dataset, created by the function doMI
depvar	the name of the dependent variable (must be in MIdata)
indpv	the names of the independent variables for the unrestricted model (must be in MIdata)
rindpv	names of restricted model independent variables (must be in indpv; when default of NULL is executed, the restricted model independent variables will be the same as the unrestricted model, minus the last variable)
dw	Should geographic proximity be used in constructing composite weight matrix (default=TRUE)
lw	Should linguistic proximity be used in constructing composite weight matrix (default=TRUE)
ew	Should ecological proximity be used in constructing composite weight matrix (default=FALSE)
doboot	Enter the number of bootstrap repetitions to calculate bootstrap standard errors. Legal values lie between 10 and 10,000. The default (doboot=500) is usually sufficient.
mean.data	When mean.data=TRUE (the default), output file includes a dataframe with mean values (across imputations) of the unrestricted model variables for each society, as well as predicted value and residuals for the restricted model, and latitude and longitude. mean.data=FALSE returns the entire, unaggregated set of data.
getismat	When getismat=TRUE, the distance weight matrix is modified in the way suggested by Getis and Aldstadt (2003). Default is FALSE.
othexog	names of additional exogenous variables (must be in MIdata; will be added to a list of 21 variables; default is NULL)

### Value

Returns a list with 7 elements:

DependVarb	Description of dependent variable
URmodel	Coefficient estimates from the unrestricted; pvalues are from bootstrap standard errors.
Rmodel	Coefficient estimates from the restricted model.
Diagnostics1	Three likelihood ratio tests: LRtestNull-R (H0: all variables in restricted model have coefficients equal zero); LRtestNull-UR (H0: all variables in unrestricted model have coefficients equal zero); LRtestR-R (H0: variables in unrestricted model, not carried over to restricted model, have coefficients equal zero). One Wald test: waldtest-R (H0: variables in unrestricted model, not carried over to restricted model, have coefficients equal zero).
Diagnostics2	Statistics without formal hypothesis tests. pLargest: the largest of proportion 1s or proportion 0s; the model should be able to outperform simply picking the most common outcome. pRight: proportion of fitted values that equal actual value of dependent variable. NetpRight=pRight-pLargest; this is positive in a good model. McIntosh.Dorfman: (num. correct 0s/num. 0s) + (num. correct 1s/num. 1s); this exceeds one in a good model; McFadden.R2 and Nagelkerke.R2 are two versions of pseudo R <sup>2</sup> .
OtherStats data	Other statistics: Composite weight matrix weights; number of imputations; number of observations. Data as used in the estimations. Observations with missing values of the dependent variable have been dropped.

## Details

Users can choose any of three kinds of proximity/similarity weight matrices for constructing a network lag term: geographic, linguistic, and ecological. In most cases, users should choose only geographic and linguistic (the defaults). The optimal composite weight matrix, constructed as the weighted sum of the chosen weight matrices, is that which returns the most significant LaGrange multiplier statistic on the unrestricted model without network lag term, estimated with OLS. The network lag term is entered in each model as the variable "Wy".

Endogeneity is a recognized problem with network lag terms. In the logit context, the network lag term will generate incorrect standard errors, so that the only legitimate p-values will be those coming from bootstrap standard errors. Bootstraps take a fairly long time to calculate, so one shouldn't set the number of repetitions too high; in most cases, good results can be obtained with `doboot=500` (the default).

## Note

Based on the methods proposed by Malcolm M. Dow and E. Anthon Eff.

Getis, A. and Aldstadt, J. (2002). Constructing the spatial weights matrix using a local statistic. *Geographical Analysis* 36:90-104.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in Econometrics*. New York: Academic Press.

McIntosh, C. S., & Dorfman, J. H. (1992). Qualitative forecast evaluation: A test for information value. *American Journal of Agricultural Economics*, 74, 209-214.

Nagelkerke, N. J. D. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78, 691-692.

## Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

## Examples

```
dpV<-"v67.d3"
UiV<-c("v2002.d2", "v1845", "v1649", "v1127.d2", "v2137", "v279.d5", "v213.d3",
      "v1265", "v1", "v234", "femecon.lp", "rectang")
RiV<-c("v1649", "v1127.d2", "v2137", "v1265")

q<-doLogit(smi, depvar=dpV, indpv=UiV, rindpv=RiV, dw=TRUE, lw=TRUE, ew=FALSE,
  doboot=1000, mean.data=TRUE, getismat=FALSE, othexog=NULL)

#--look at first six objects in q--
q[1:6]
```

---

CSVwrite      Write object to \*.csv file

---

### Description

The function writes an object, with elements capable of being coerced to a dataframe, to a csv file. It is used to write the output from `doOLS` or `doLogit` to a file that can be read by a spreadsheet.

### Usage

```
CSVwrite(object, filestem, appnd2=FALSE)
```

### Arguments

`object`    Object to be written—typically output from function `doOLS` or `doLogit`  
`filestem`    The base name of the \*.csv file (do not include the “.csv” extension)  
`appnd2`    Should the object be appended to the existing file? (default=FALSE)

### Value

No values are returned in the R environment; only changes occur to the specified \*.csv file.

### Details

Set the option `appnd2=TRUE` to append the output of object to an existing file with base name “filestem”. The default will simply overwrite any existing csv file with base name “filestem”.

### Note

#### Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

### Examples

```
CSVwrite(h, "olsresults", FALSE)
```

### Description

This function writes a png format Pacific-centered world map file to the working directory. Dots represent societies, and the size and color of the dots reflects the value of a variable specified by the user. Options allow presentation of information about local autocorrelation and dfbetas.

### Usage

```
mkmappng (usedata, varb, filetitle, show="ydata", numnb.lg=3, numnb.lm=20, numch=0,
          pv1m=.05, dfbeta.show=FALSE, map.width=8, map.height=5, map.units="in",
          map.pointsize=10, map.res=500)
```

### Arguments

usedata	Name of a dataframe. It must contain a column named “lati” and a column named “long” (latitude and longitude in decimal degrees)
varb	Name of a variable in the dataframe.
filetitle	Stem title of png file (“.png” suffix added automatically)
show	Type of value to display. Legal values are <i>lgt</i> (local G), <i>ydata</i> (original data values), <i>lmtp</i> (classifies points into significant and non-significant local autocorrelation, based on local Moran), and <i>lmtz</i> (local Moran z-value). Default is <i>lgt</i> .
numnb.lg	Number of nearest neighbors to use when creating local G. Default is 3.
numnb.lm	Number of nearest neighbors to use when creating local Moran. Default is 20.
numch	Number of convex hulls to draw around regions of local autocorrelation. Default is 0.
pv1m	Cut-off p-value for considering a local Moran statistic significant. Default is 0.05.
dfbeta.show	Should map indicate points with significant dfbeta values for this variable. Default is FALSE.
map.width	Parameter for png map file. This gives width of map. Default is 8.
map.height	Parameter for png map file. This gives height of map. Default is 5.
map.units	Parameter for png map file. This gives units in which width and height are measured. Default is “in”.
map.pointsize	Parameter for png map file. This gives pointsize. Default is 10.
map.res	Parameter for png map file. This gives resolution of map file. Default is 500 dpi.

### Value

The function writes a png format map to a file in the working directory. Larger values of the mapped variable are shown as larger and darker (reddish) circles; smaller values are shown as smaller and lighter (yellowish) circles.

### Details

Option `show=lgt` gives the local G statistic, which is essentially a spatial moving average, converted to a z-score. It is a reasonable way to smooth—spatially—map points. The default uses only the three nearest neighbors, plus self, to calculate this spatial moving average.

The local Moran is a test for autocorrelation, i.e. the degree to which a society has values similar to those of its neighbors, where the default number of neighbors is 20. Option `show=lmtz` will display the local Moran z-score, and option `show=lmtp` displays the binary significant/not significant for the z-score, using the p-value given in option `pv1m`. Convex hulls are drawn around areas of significant positive local autocorrelation; one must input the number of convex hulls to draw, but otherwise assignment of a point to a specific convex hull is automatic, based on distances between points. Usually some experimentation is needed to find the correct number of convex hulls, and it is easiest to do this experimentation on maps where `show=lmtp`.

This function is intended for use with data relevant to models estimated by the function `doOLS`. The function `doOLS` has the option `mean.data`, when this is set to TRUE (the default), the output from `doOLS` contains a dataframe with values for the dependent and independent variables (including *Wy*) calculated as the mean across all imputed datasets. There are also latitude and longitude coordinates, and the mean values of the dfbetas for variables used in the restricted model. The societies which, when included, cause a significant change in the estimated parameter in the restricted model, can be shown

in the map when `dfbeta.show=TRUE`. Triangles pointing upward indicate societies which, when included, significantly increase the value of the coefficient; triangles pointing downward indicate societies whose inclusion significantly lowered the value of the coefficient.

## Note

### Author(s)

Anthon Eff      Anthon.Eff@mtsu.edu

### Examples

```
dpV<-"v67.d3"
UiV<-c("v2002.d2", "v1845", "v1649", "v1127.d2", "v2137", "v279.d5", "v213.d3",
      "v1265", "v1", "v234", "femecon.lp", "rectang")
RiV<-c("v1649", "v1127.d2", "v2137", "v1265")

h<-doOLS(MIdata=smi, depvar=dpV, indpv=UiV, rindpv=RiV, othexog=NULL,
        dw=TRUE, lw=TRUE, ew=FALSE, stepW=TRUE, boxcox=FALSE, getismat=FALSE,
        relimp=TRUE, slmtests=FALSE, haustest=NULL, mean.data=TRUE, doboot=500)

p<-h[[12]]

# experimenting to find the right number of convex hulls
sapply(2:11, function(x) mkmappng(p, "femecon.lp", paste("Womenswork", x, sep=""),
  show="lmtpr", numch=x, dfbeta.show=TRUE))

# creates file called "Womenswork_ydata.png"
mkmappng(usedata=p,   varb="femecon.lp",   filetitle="Womenswork",   show="ydata",
numch=8, dfbeta.show=TRUE)
```