Package ‘PathSelectMP’

December 19, 2017

Type Package
Title Backwards Variable Selection for Paths using M Plus
Version 1.1
Date 2017-12-19
Author William Terry, Meredith Ray, Hongmei Zhang
Maintainer William Terry <wterryj6a214@gmail.com>
Description Primarily for use with datasets containing only categorical variables, although continuous variables may be included as independent variables in paths. Using M Plus, backward variable selection is performed on all Total, Total Indirect, and then Direct effects until none of these effects have p-values greater than the specified target p-value. If there are missing values in the data, imputations are performed using the Mice package. Then selection is performed with the imputed data sets, and results are averaged.
License GPL (>= 2)
Depends R (>= 3.1.2)
Imports mice, MplusAutomation
NeedsCompilation no
Repository CRAN
Date/Publication 2017-12-19 21:06:10 UTC

R topics documented:

PathSelectMP-package .................................................. 2
AddOnAllInd ................................................................. 4
AddOnINDStatements ..................................................... 6
AllBackwardSelect ....................................................... 7
AllSummary ................................................................. 8
AllSummary2 ................................................................. 10
AllTotalEffOutput ......................................................... 11
AverageRRs ................................................................. 13
CalculatRiskRatios ....................................................... 14
CheckVarNames ........................................................... 15
ConvertData2 ............................................................... 16
PathSelectMP-package

Backwards Variable Selection for paths using M Plus

Description

M Plus must be installed. This package is primarily for use with datasets containing only categorical variables, although continuous variables may be included as independent variables in paths. Backward variable selection is performed on all Total, Total Indirect, and then Direct effects until none of these effects have p-values greater than the specified target p-value. In some cases a given starting set of paths may produce singularity issues, in which case, the user should revise the set of possible paths. It’s very important to delete all folders which are specified in Initialize and other functions where M Plus is called to read and write .inp and .out files. Or if the default is used and the user doesn’t specify directories and folders, delete the default folder used before performing selection again or performing with a new example with the same default folder name.

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>PathSelectMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Date</td>
<td>2016-04-20</td>
</tr>
<tr>
<td>License</td>
<td>GPL (&gt;= 2)</td>
</tr>
</tbody>
</table>
The most important functions to use are `simulate` to simulate data for an example, `initialize` to write initial paths and input file, `AllBackwardSelect` to perform backwards selection with .inp and .out files saved in current working directory or user specified directory, `AllSummary2` for a summary of all direct effects, `CreateTotalSummary` for a summary of all total effects, `CalculateRiskRatios` to calculate direct effect risk ratios, `TotalRiskRatios` to calculate total effect risk ratios, and `AverageRRs` to calculate average risk ratios for direct effects or total effects when comparing multiple imputed datasets.

It is recommended to follow and review examples since many functions require other functions to be run first and certain data files to be saved and created before running a function. Also note that all variable names must begin with a capital letter and contain only letters and numbers in this version.

Author(s)

William Terry, Meredith Ray, Hongmei Zhang

Maintainer: <hzhang6@memphis.edu>

References

MPlus, MPlusAutomation, mice

Examples

```c
## Not run:
#example 1
#creates generated data set and stores as InitD
InitD=Simulate(exampleNum=1)
xxx=Initialize(InitD,WhichCat=c(1,1,1,1))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])

###DirectEffects
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
www=CalculateRiskRatios(AllDat1[[1]],NADes=c(-99),WhichCat=c(1,1,1,1),
WhichRiskCalc=c(0,0,0,0,0))

###TotalEffects
www=CalculateRiskRatios(AllDat1[[1]],NADes=c(-99),WhichCat=c(1,1,1,1),
WhichRiskCalc=c(0,0,0,list(c(3,4,5)),0,0))
RRT1=lapply(AllDat1,TotalRiskRatios,InputDepVal=1)

#example 2
Simulated=Simulate(n=900,MissingYN=1,exampleNum=2)
#MissingYN is 1 for add missing data 0 is default which is don’t add missing data
uu=Initialize(Simulated,NumImpute=3,WhichCat=c(1,1,1,1,0,1,0,0),
DataFileName="Example2",Namefile="Example2D",
Directry=getwd(),WhichRowsImp=c(1:450))

#NumImpute is the number of imputed datasets
AB=AllBackwardSelect(uu[[1]],Directry=getwd(),PSig=0.04)
DE=AllSummary2(uu[[1]],Directry=getwd())

###DirectEffects
AllDat1=ConvertData2(uu[[1]],uu[[2]],DE[[1]])
www=lapply(AllDat1,CalculateRiskRatios,NADes=c(-99),
Directry=getwd()}
```
```

AddOnAllInd

Description

wrapper function for creating indirect effects from direct effects and function writes indirect effects to file

Usage

AddOnAllInd(filename, indstatements, directory = getwd())

```
AddOnAllInd

Arguments

FileName  name without path of mplus .inp and .out files to read and write which is the base filename such as "New"
IndStatements  the indirect statements to write created by AddOnINDStatements
Directry  the path where all of the .inp and .out files are located

Details

No Details required.

Value

no value returned directly instead an extra .inp file is written and run with all IND effects

Note

This function is a helper function for Initialize indirectly through WriteInitialInpFile. Not used as a helper function is is useful only when PasteIND=0 options is utilized by Initialize due to having a very large number of variables, and then after backwards selection is performed, the user desires to add back to the model any possible indirect effects not in the model but possible with the final model.

Author(s)

William Terry

References

No references

Examples

## Not run:

InitD=Simulate(MissingYN=0,exampleNum=3)  
xxx=Initialize(InitD,NumImpute=0,WhichCat=c(1,1,1,1,0,1,0),PasteIND=0)  
ggg=AllBackwardSelect(xxx[[1]])  
zzz=AllSummary(xxx[[1]][[2]])  
qqq=AddOnAllInd(xxx[[1]],zzz)  

## End(Not run)
AddOnINDStatements   *Indirect Effect List Creator*

**Description**

Creates list of all indirect effects

**Usage**

AddOnINDStatements(MeanDirectList, PasteIND)

**Arguments**

- **MeanDirectList**: Matrix with direct effects or the initialize matrix with 1s designating paths used in analysis
- **PasteIND**: a value of 1 indicates to use all possible indirect effects in modelling and a value of 0 is input to only use direct effects in modelling

**Details**

This function is a helper function and is not necessary for the regular user. The user is instead referred to `AddOnAllInd`

**Value**

INDlist: each element is a list which is an "IND" statement

**Note**

when `PasteIND=0` IND statements are created in the .inp files but only for direct effects

**Author(s)**

William Terry

**References**

No references

**Examples**

InitD=Simulate(n=1000)
InputInitializeMat=CreateInitializeMatrix(InitD,WhichCat=c(1,1,1,1,0))
IndList=AddOnINDStatements(InputInitializeMat,PasteIND=1)
AllBackwardSelect

Run Backward Variable Selection in Path Analysis with M Plus

Description

performs backward selection in M Plus for all input data sets which is either 1 data set or more if imputations specified

Usage

AllBackwardSelect(AllNames, Directry = getwd(), PSig = 0.05)

Arguments

AllNames  string name of folder containing .inp and .out M Plus files, and also start of filenames for .inp and .out files
Directry   the path containing the folder where the M Plus files are saved and run default is current working directory
PSig       the alpha value to use for accepting or rejecting null hypothesis that coefficient is equal to zero in path

Details

Backward variable selection is performed for an entire path where 1 variable at a time is deleted where the variable which is deleted has the highest p value for a total effect which is greater than PSig, and if no such variable in a path with corresponding p-value exists then the variable in a path with the highest p value for an indirect effect which is greater than PSig is deleted, and finally if no such variable in a path with corresponding p-value exists for the first two cases, then the variable in a path with the highest p value for a direct effect which is greater than PSig is deleted. This process repeats until no p-values for effects are greater than PSig.

Value

There is no output from this function rather M Plus .inp and .out files are written and saved in the specified folder and path one at a time as backward stepwise variable selection is performed and updated.

Note

This is one of the most important functions for the user in this package. Also make sure the folder where the .inp and .out files are saved does not exist yet or exists but is empty. Also if re-running with same folder, delete previous results before re-running. If the m plus first .inp file fails to run due to a singularity issue, consider creating a new InputInitializeMat for Initialize. See CreateInitializeMatrix for the format of the InputInitializeMat, and consider replacing some 1 values for included paths with 0s to not include the path.
Author(s)
William Terry

References
M Plus and MplusAutomation

Examples
## Not run:

```r
InitD=Simulate(n=1000,MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
## End(Not run)
```

<table>
<thead>
<tr>
<th>AllSummary</th>
<th>Direct Effects Path Summaries</th>
</tr>
</thead>
</table>

Description
One of the main and most important functions. Ties together Indirect Statements and summary output of direct effects

Usage

```r
AllSummary(AllNames, Directry = getwd(), GreaterThanNum = 0, PasteIND = 1)
```

Arguments

<table>
<thead>
<tr>
<th>AllNames</th>
<th>AllNames is the filename where the M Plus.inp and .out files are located</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directry</td>
<td>this the path where the folder specified by AllNames is located and default is working directory</td>
</tr>
<tr>
<td>GreaterThanNum</td>
<td>number of output files containing path over which to average which must be less than the number of imputations and is only used if multiple imputations are performed. Default 0 which is to use all data sets in mean calculations.</td>
</tr>
<tr>
<td>PasteIND</td>
<td>a value of 1 indicates to use all possible indirect effects in modelling and a value of 0 is input to only use direct effects in modelling</td>
</tr>
</tbody>
</table>

Details
Must initialize and run backwards selection before using this function
Value

- **Average**: list with the following objects:
  - **DirectEffectCounts**: count matrix for number of times path appears which will be 1’s and 0’s if no imputed data sets are used.
  - **MeanDirectEffects**: mean values of direct effects for paths which are just the direct effects if no imputations are performed.
  - **MeanStandardError**: the mean square error of the effect parameters averaged over imputed data sets if they exist.
  - **MeanPValue**: mean p values of these direct effects.
  - **MinPVals**: minimum p values of these direct effects.
  - **MaxPVals**: maximum p value of these direct effects.
  - **MedianPVals**: median p value of these direct effects.
  - **INDStatements**: lists of indirect effect relations.

Note

This function must be run before `AddOnAllInd` can be run (see examples), but otherwise is not a very useful summary function. The user is instead referred to `AllSummary2`. The means in the matrices above are only calculated for those paths and parameters which appear in the count matrix with a value greater than the GreaterThanNum.

Author(s)

William Terry

References

M Plus

See Also

- `Initialize AllBackwardSelect`

Examples

```r
## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
zzz=AllSummary(xxx[[1]][[2]])

## End(Not run)
```
AllSummary2

Direct Effect Summary

Description

summarizes direct effects from path analysis for all imputed datasets

Usage

AllSummary2(AllNames, Directory = getwd(), GreaterThanNum = 0)

Arguments

AllNames is a list of the filenames as strings where the M Plus.inp and .out files are located which is length 1 if no imputations

Directory this the path where the folder specified by AllNames is located and default is working directory

GreaterThanNum number of output files containing path over which to average which must be less than the number of imputations and is only used if multiple imputations are performed. Default 0 which is to use all data sets in mean calculations.

Details

this function is essentially the same as AllSummary except it does not return the indirect effect list

Value

list of length 2:

DirectEffects Direct Effects of each imputed data set or just direct effects of one data set if no imputation

Average list with the following objects:

DirectEffectCounts count matrix for number of times path appears which will be 1's and 0's if no imputed data sets are used

MeanDirectEffects mean values of direct effects for paths which are just the direct effects if no imputations are performed

MeanStandardError the mean square error of the effect parameters averaged over imputed data sets if they exist

MeanPValue mean p values of these direct effects

MinPVals minimum p values of these direct effects

MaxPVals maximum p value of these direct effects

MedianPVals median p value of these direct effects
**Note**

Must initialize and run backwards selection before using this function

**Author(s)**

William Terry

**References**

M Plus

**See Also**

AllSummary

**Examples**

```r
## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])

## End(Not run)
```

---

**AllTotalEffOutput**  
*Summary of Total Effects*

**Description**

total effects averaged over imputed data sets, if there are imputations, with accompanying p value summaries

**Usage**

```r
AllTotalEffOutput(AllDat, GreaterThanNum = 0, Directry = getwd())
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllDat</td>
<td>output from ConvertData2</td>
</tr>
<tr>
<td>GreaterThanNum</td>
<td>number of output files containing path over which to average which must be</td>
</tr>
<tr>
<td></td>
<td>less than the number of imputations and is only used if multiple imputations</td>
</tr>
<tr>
<td></td>
<td>are performed. Default 0 which is to use all data sets in mean calculations.</td>
</tr>
<tr>
<td>Directry</td>
<td>this the path where the folder specified by AllNames is located and default is working directory</td>
</tr>
</tbody>
</table>
Details

Must initialize and run backwards selection before using this function

Value

list of matrices which are respectively:

Count
number of times total effect appears in imputed data sets final path selection

AverageEffects
mean values of total effects for paths which are just the total effects if no imputations are performed

AverageStandardError
the mean square error of the effect parameters averaged over imputed data sets if they exist

AveragePVal
mean p values of these total effects

MinPVal
minimum p values of these total effects

MaxPVal
maximum p value of these total effects

MedianPVal
median p value of these total effects

Note

total effects and direct effect averages are done separately so a path may be present in one summary and not the other when using the same GreaterThanNum which is not equal to zero

Author(s)

William Terry

References

M Plus

See Also

AllSummary2

Examples

## Not run:

```r
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
      ggg=AllBackwardSelect(xxx[[1]])
      yyy=AllSummary2(xxx[[1]])
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
TotsAllEff=AllTotalEffOutput(AllDat1)

## End(Not run)
```
AverageRRs

Calculate Average Risk Ratios over all imputed data sets

Description

Averages the risk ratio values for all paths in imputed data sets if the specific path relationship
appears frequently enough as specified by user.

Usage

AverageRRs(ListORatioMats, GreaterThanCountNum)

Arguments

ListORatioMats a list of dataframe where each dataframe is the risk ratios matrix calculated for
each imputed data set

GreaterThanCountNum

the number of times a path relationship should appear more times than in order
to be included in averaging of risk ratios for this relationship default is 0

Details

adds risk ratios for relationship and divides by number of times this relationship appears as long as
the relationship appears more than GreaterThanCountNum

Value

AverageRiskRatios

data frame consisting of averaged risk ratios

CountInAverage

same data frame as AverageRiskRatios except in place of averaged risk ratio
is corresponding count number for path relationship over all imputed data sets

Note

no notes

Author(s)

William Terry

References

no references
CalculatRiskRatios

Calculate Risk Ratios from Direct Effects

Description

calculates risk ratios from direct effects for each imputed data set path analyssi or just one data set
path analysis if no imputations

Usage

CalculatRiskRatios(AllDat, NADes, Directry = getwd(), WhichCat, WhichRiskCalc)

Arguments

- **AllDat**: output from ConvertData2
- **NADes**: a one element vector containing a string which is the missing value designation
- **Directry**: the working directory or the path where the folder for backward selection is located
- **WhichCat**: list length of number of variables in initial data with 1 specifying variable is
categorical and 0 specifying variable is continuous
- **WhichRiskCalc**: list length of number of variables in initial data with 1 specifying use default
for risk ratio calculation which is to calculate a risk ratio for all category levels
compared to the lowest level and include an averaging over all of these levels
when calculating other variable risk ratios and if variable is continuous 1 specifies
using average of all values for variable compared to zero. Otherwise a list
can be given to specify which values to use for the risk ratios which are averaged
over when calculated risk ratios for other variables.

Examples

```r
## Not run:
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
www=lapply(AllDat1,CalculatRiskRatios,NADes=c(-99),WhichCat=c(1,1,1,1,0),WhichRiskCalc=c(0,
list(c(0,1,2)),list(c(0,1,2,4)),list(c(0,1,2)),0))
Avgwwa=AverageRRs(wwwA,0)

RRT1=lapply(AllDat1,TotalRiskRatios,InputDepVal=1)
RRT1avg=AverageRRs(RRT1,0)

## End(Not run)
```
Details

if there is more than one threshold value for a categorical dependent variable the calculations are performed for all thresholds according to M Plus Manual Chapter 14 on probit calculations

Value

A dataframe where the row names are dependent variables and the columns are independent variables with the corresponding risk ratios.

Note

averaging over possible combinations of levels of other variables in the risk ratio calculations is necessary since the direct effects are from a probit model using WLSMV in M Plus rather than from a logistic regression model which was not possible for categorical (non-binary) variables which are dependent and independent in path when this program was written

Author(s)

William Terry

References

M Plus method WLSMV

Examples

```r
## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
www=lapply(AllDat1,CalculateRiskRatios,NADes=c(-99),WhichCat=c(1,1,1,1,0),WhichRiskCalc=c(0,
                list(c(0,1,2)),list(c(0,1,2,4)),list(c(0,1,2)),0))

## End(Not run)
```

---

<table>
<thead>
<tr>
<th>CheckVarNames</th>
<th>Variable Name Check</th>
</tr>
</thead>
</table>

Description

Variable Names must begin with a capital letter and can contain only letters and numbers

Usage

CheckVarNames(InitialData)
ConvertData2

Converts Necessary Inputs for CalculatRiskRatios

Description

converts inputs to list structure

Usage

ConvertData2(List1, List2, jjlist1)
Arguments

List1  list of strings where each string is the name of the folder for each imputed backward selection or just one name if no imputation

List2  list of strings here each string is the name of the .dat file (without .dat in string) for the imputed data set or just one .dat file name if no imputation

jjlist1 list of dataframes where each dataframe is the direct effects dataframe from backward selection with the imputed dataset which is only a list of length 1 with one matrix of direct effects if no imputations

Details

follow example and in future versions a wrapper function will be written

Value

list of length number of imputed data sets (length 1 if no imputations) containing three elements in each list. The first element is the name of the imputed selection folder contained in List1, the second element is the name of the dataset contained in List2, and the 3rd element is the dataframe of direct effects in jjlist1

Note

no notes

Author(s)

William Terry

References

No references

See Also

CalculatRiskRatios

Examples

## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])

## End(Not run)
ConvertData3

Used to make multiple list names for imputations

Description

uses imputation seed numbers and names of data to create names of imputed data, helper function

Usage

ConvertData3(N1, N2, List1)

Arguments

N1            name of file
N2            name of data file
List1        list of imputation seeds

Details

regular user doesn’t need this function but it useful for creating new imputation folder names

Value

returns list of new folders for imputed analysis

Note

helper function

Author(s)

William Terry

References

no reference

Examples

NumImpute=3
startSeedImputations=1000
NumImplist=seq(startSeedImputations,(startSeedImputations+NumImpute-1),1)
DataFileName="NewData"
NameFile="New"
NameFile=paste0(NameFile,"_")
DataFileName=paste0(DataFileName,"_")
AllData=ConvertData3(NameFile,DataFileName,NumImplist)
**CountImputedEffect**  
*Count and Average Effects for All Imputed Datasets*

---

**Description**

averages direct effects and corresponding standard errors and p values for results from each imputed data set

**Usage**

`CountImputedEffect(LL, LLse, LLPVal, GreaterThanNum)`

**Arguments**

- **LL**  
  list containing square matrices as each element, 1 matrix for each imputed data set, with the final direct effects
- **LLse**  
  list containing square matrices as each element, 1 matrix for each imputed data set, with the final standard errors for direct effects
- **LLPVal**  
  list containing square matrices as each element, 1 matrix for each imputed data set, with the final p values for direct effects
- **GreaterThanNum**  
  an integer less than the number of imputed data sets representing the number of times a path must be counted more than this number of times in order to be included in averaging of effects, effect standard errors, and effect p values

**Details**

this function is a helper function and does not need to be used by the regular user

**Value**

list which contains the following objects:

- **DirectEffectCounts**  
  count matrix for number of times path appears which will be 1’s and 0’s if no imputed data sets are used
- **MeanDirectEffects**  
  mean values of direct effects for paths which are just the direct effects if no imputations are performed
- **MeanStandardError**  
  the mean square error of the effect parameters averaged over imputed data sets if they exist
- **MeanPValue**  
  mean p values of these direct effects
- **MinPVals**  
  minimum p values of these direct effects
- **MaxPVals**  
  maximum p value of these direct effects
- **MedianPVals**  
  median p value of these direct effects
Note
No notes

Author(s)
William Terry

References
no references

See Also
AllSummary

Examples
## Not run:

InitD=Simulate()
xxx=Initialize(InitD)
bbb=AllBackwardSelect(xxx[[1]])
AllNames=xxx[[1]]
LL1=lapply(AllNames,CreateSummaryMats,OutputSE=FALSE,OutputPVal=FALSE,Dirctry=getwd())
LL1se=lapply(AllNames,CreateSummaryMats,OutputSE=TRUE,OutputPVal=FALSE,Dirctry=getwd())
LL1p=lapply(AllNames,CreateSummaryMats,OutputSE=FALSE,OutputPVal=TRUE,Dirctry=getwd())
LLL=CountImputedEffect(LL1,LL1se,LL1p,GreaterThanNum=0)

## End(Not run)

CreateInitializeMatrix

*Create Possible Path Matrix*

Description
creates a dataframe with 1’s and 0’s where a 1 represents inclusion in the path and 0 represents exclusion

Usage
CreateInitializeMatrix(InitialData, WhichCat, empty = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InitialData</td>
<td>the dataframe with the initial data</td>
</tr>
<tr>
<td>WhichCat</td>
<td>a vector of 0’s and 1’s length of number of columns of InitialData where 1 is</td>
</tr>
<tr>
<td></td>
<td>a categorical variable and 0 is a continuous variable</td>
</tr>
<tr>
<td>empty</td>
<td>if empty is TRUE then a dataframe of all 0’s is returned to allow user to individually assign 1’s for path inclusion more easily</td>
</tr>
</tbody>
</table>
Details

the rows represent dependent variables in the path and the columns represent independent variables in the paths with a 1 representing in M Plus terminology row variable ON column variable

Value

returns matrix required by Initialize for all initial paths to run the default is an upper triangular matrix. This is a square matrix and names of rows is the same as the names of columns

Note

In this version only Categorical variables can be dependent variables and if using the default upper triangular matrix, due to the current way indirect effects are created, only approximately 8 or 9 variables can be used with reasonable computing time. The input to Initialize must be in the same format as this output so user may want to use this function with empty as FALSE and substitute 1’s for desired possible paths. Also note that row names and column names must be the same and must start with a capital letter and only contains numbers and letters in this version.

Author(s)

William Terry

References

M Plus

Examples

```r
InitD=Simulate()
I=CreateInitializeMatrix(InitD,WhichCat=c(rep(1,ncol(InitD)-1),0))
```

CreateSummaryMats Extract and Summarize Direct Effects

Description

helper function for AllSummary2 and not necessary for regular user

Usage

```r
CreateSummaryMats(FileName, OutputSE = FALSE, OutputPVal = FALSE, Directry, OutputFinalMat = TRUE)
```
Arguments

FileName string which is the name of the folder where the .inp and .out files are stored
OutputSE TRUE outputs standard error dataframe and FALSE does not output standard error dataframe
OutputPVal TRUE outputs p-value dataframe and FALSE does not p-value dataframe
Directry this the path where the folder specified by AllNames is located and default is working directory
OutputFinalMat a input value of 1 returns matrix and value of 0 returns a list

Details

helper function

Value

summaries of direct effects

Note

no notes

Author(s)

William Terry

References

M Plus

See Also

AllSummary2

Examples

```r
# Not run:
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
rgg=AllBackwardSelect(xxx[[1]])
lll=lapply(xxx[[1]],CreateSummaryMats,OutputSE=FALSE,
OutputPVal=FALSE,Directry=getwd(),OutputFinalMat=1)

# End(Not run)
```
CreateSummaryMats2

helper function for AddOnAllInd

Description

helper function for AddOnAllInd, not necessary for regular user

Usage

CreateSummaryMats2(AllDat, Directry)

Arguments

AllDat  not important see AddOnAllInd
Directry  this the path where the folder specified by AllNames is located and default is working directory

Details

see AddOnAllInd

Value

the direct output is not important, rather the function writes new .inp file and runs a .out file with MplusAutomation with all of the newly added indirect effects, some of which were deleted during backward selection

Note

helper function no need to run by itself and function is called by AddOnAllInd which user should see for example

Author(s)

William Terry

References

M Plus

See Also

AddOnAllInd

Examples

#see AddOnAllInd which it is a helper function for
**CreateTotalEffMat**

*Summarizes Total Effects*

**Description**

this is a helper function for AllTotEffOutput1 which is better for the regular user and it is a helper function for CreateTotalSummary, and it is used to extract total effects from M Plus .out files

**Usage**

`CreateTotalEffMat(FileNaName, Directry)`

**Arguments**

- **FileName**
  a list of length 1 containing the string which is the folder name which contains .inp and .out M Plus files after backwards selection
- **Directry**
  the path where the folder specified by FileName is located

**Details**

extracts total effects with standard errors and p values from final .out file

**Value**

returns list of length 4:

- **TotalEffects**
  dataframe containing the total effects
- **TotalEffectsStandardError**
  data frame with the standard errors of the total effects
- **TotalEffectsPVals**
  dataframe contains the p values for these effects
- **TotalEffectsCount**
  data frame with a 0 if the total effect is not present and a 1 if it is present in the analysis

**Note**

this function is applied over all imputed data sets or can be used with one data set and analysis at a time, but this function is a helper function that is likely not necessary for the regular user

**Author(s)**

William Terry

**References**

M Plus
CreateTotalSummary

Summarizes and Averages Total Effects

### Description

this is a helper function for `AllTotalEffOutput` which is better for the regular user

### Usage

```r
CreateTotalSummary(AllTots, GreaterThanNum)
```

### Arguments

- **AllTots**: is output from `CreateTotalEffMat` applied to each imputed data sets so it is a list of length number of imputed data sets (or length 1 if no imputed data sets) containing each list a list of dataframes where each data frame is total effects, standard errors, p-values, path designation respectively
- **GreaterThanNum**: number of output files containing path over which to average which must be less than the number of imputations and is only used if multiple imputations are performed.

### Details

not necessary for regular user

### Value

output is same as `AllTotalEffOutput` which is easier to run

---

See Also

- `AllTotalEffOutput`

Examples

```r
## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD, NumImpute=3, WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
zzz=AllSummary(xxx[[1]])[[2]]
qqq=AddOnAll1Ind(xxx[[1]], zzz)
AllDat1=ConvertData2(xxx[[1]], xxx[[2]], yyy[[1]])
AllFileNames1=sapply(AllDat1, OnlyNumberElement, 1)
Tots=CreateTotalEffMat(AllFileNames1[[1]], Directry=getwd())

## End(Not run)
```
Note

must run backwards selection first and some other necessary functions see example

Author(s)

William Terry

References

M Plus

See Also

AllTotalEffOutput

Examples

## Not run:

```r
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
zzz=AllSummary(xxx[[1]])[[2]]
qqq=AddOnAllInd(xxx[[1]],zzz)
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
AllFileNames1=sapply(AllDat1,OnlyNumberElement,1)
Tots1=lapply(AllFileNames1,CreateTotalEffMat,Directry=getwd())
AllTotalEffOutput1=CreateTotalSummary(Tots1,GreaterThanNum=0)
```

## End(Not run)
Arguments

InitialData    dataframe of data to be used in model creation
ImputeSeed    the seed for imputations
NADes         the missing value designation
DataFileNameS name of imputed data file
WhichCat      vector length of names of InitialData specifying '1' for categorical and '0' for continuous variable
WhichImpute   vector length of names of InitialData specifying '1' for Impute column and '0' for no Imputation
WhichRowsImp  vector length of number of rows of InitialData specifying '1' for Impute row and '0' for no Imputation
AllMethods    default is set at "logreg" for binary imputation "polr" for categorical imputation "pmm" for continuous

Details

this is a helper function for Initialize when imputations are desired by user, and it should not be used as standalone function by regular user

Value

No value is returned rather a new data set is saved

Note

This function is best used by specifying the number of desired imputations with NumImpute in Initialize

Author(s)

William Terry

References

mice https://cran.r-project.org/web/packages/mice/mice.pdf

See Also

mice https://cran.r-project.org/web/packages/mice/mice.pdf

Examples

## Not run:

InitD=Simulate(MissingYN=1,exampleNum=1)
WhichCat=c(1,1,1,1,0)
WhichImpute=c(rep(1, ncol(InitD)))
WhichRowsImp=c(rep(1, nrow(InitD)))
Initialize

Create Initial Objects

Description
write data files and initial .inp files to setup file directory system for running backwards selection using M Plus

Usage
Initialize(InitialData, NumImpute = 0, DataFileName = "NewData", NameFile = "New", Directry = getwd(), NADes = c(-99), startSeedImputations = 1000, InputInitializeMat = "N", WhichCat = rep(1, ncol(InitialData)), AllMethods = c("logreg", "polr", "pmm"), WhichImpute = rep(1, ncol(InitialData)), WhichRowsImp = c(1:nrow(InitialData)), PasteIND = 1)

Arguments
InitialData data frame of initial data
NumImpute the number of imputed datasets to create default is 0 for no imputation
DataFileName the name of the .dat file to be created from the initial data or imputed datasets default is "NewData"
NameFile the name of the folder where the .inp and .out files with the same base name are stored default is "New"
Directry the path where the folder containing the .inp .out and .dat files are saved during backward selection
NADes a list of length 1 with the value for missing data default is c(-99)
startSeedImputations the seed for the first imputed data sets all other data sets seed is plus one from this value default is 1000
Initialize

InputInitializeMat

square dataframe where row name and column names are equal and are the variable names of InitialData with 1 for include path and 0 for exclude created by CreateInitializeMatrix default creates upper triangular data frame

WhichCat

list of length number of variables in InitialData with 1 for categorical and 0 for continuous variables default is all 1s for all categorical data

AllMethods

default is set at "logreg" for binary imputation "polr" for categorical imputation "pmm" for continuous

WhichImpute

list of length of names of InitialData specifying '1' for Impute column and '0' for no Imputation

WhichRowsImp

list of length of number of rows of InitialData specifying '1' for Impute row and '0' for no Imputation

PasteIND

a value of 1 indicates to use all possible indirect effects in modelling and a value of 0 is input to only use direct effects in modelling

Details

this function must be run before using any other of the important functions in order to create initial input files

Value

Output is list of length 2:

AllNames

list containing the names of the folders containing each imputed data set path analysis

AllDataNames

list containing the name of the imputed datasets which is just length one element if there are no imputations. Also files are written and saved.

Note

the PasteIND=0 option should be used when the set of variables is too large to specify all indirect effects. In selection, AllBackwardSelect, if the mplus first .inp file fails to run due to a singularity issue, consider creating a new InputInitializeMat. See CreateInitializeMatrix for the format of the InputInitializeMat, and consider replacing some 1 values for included paths with 0s to not include the path. It is also very important that all variables start with a capital letter and contain only numbers and letters (no spaces or special characters) in this version

Author(s)

William Terry

References

MplusAutomation and mice

See Also

CreateInitializeMatrix
Examples

## Not run:

```r
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))

## End(Not run)
```

---

mod  

**Simple Modular arithmetic**

### Description

mod operator created for syntax reasons

### Usage

```r
mod(x, m)
```

### Arguments

- `x`  
  number
- `m`  
  modulus m number

### Details

helper function created for syntax reasons

### Value

number which is x mod m

### Note

no notes

### Author(s)

R help

### References

no references

### Examples

```r
mod(10, 3)
```
### MPlusBackwardSelect

**Backward Selection Helper Function**

#### Description

performs backward selection for `AllBackwardSelect`

#### Usage

```r
MPlusBackwardSelect(FileNmae, Directry, PSig)
```

#### Arguments

- **FileName**: name of folder where .inp and .out files saved
- **Directry**: the path where folder is located containing .inp and .out M Plus files
- **PSig**: the significance level used for backward selection

#### Details

the regular user should not use this function and is referred to instead use the wrapper function `codeAllBackwardSelect` which calls this function

#### Value

does not return value but rather reads and writes .inp and .out files to specified folder and directory

#### Note

see `AllBackwardSelect`

#### Author(s)

William Terr

#### References

M Plus

#### Examples

```r
## Not run:

InitD=Simulate()
xxx=Initialize(InitD)
ggg=MPlusBackwardSelect(xxx[[1]],getwd(),0.05)

## End(Not run)
```
NewBinseqWrap

Possible Combinations of Elements, 1 from each list

Description
creates a list of lists where each list element has n elements 1 element from each of the first n lists

Usage
NewBinseqWrap(n, PossibleCoefs)

Arguments
n an integer less than or equal to the number of lists used to choose from which specifies the length of each combination elements from the list choices
PossibleCoefs a list of lists where one element at a time will be combined with elements of the other list

Details
this is a helper function and not necessary for the regular user

Value
returns the list of lists for possible combinations of 1 element at a time from each list

Note
The global variable V is used to return the list and thus V should not be used for any other variable name. This may need to be corrected in future versions

Author(s)
William Terry

References
No references

Examples
## Not run:
NewBinseqWrap(n=3,PossibleCoefs=c(list(c(0,1,2)),list(c(0,1,2,3)),list(c(4,5,6))))
## End(Not run)
**NewNamesThresh**

<table>
<thead>
<tr>
<th>NewNamesThresh</th>
<th>Threshold Names</th>
</tr>
</thead>
</table>

**Description**

Converts M Plus threshold names to the name of the category given in the data.

**Usage**

```r
NewNamesThresh(FileName, DataName, ThreshName,
InitialData, Directry = getwd(), NADes = c(-99))
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileName</td>
<td>name of .out Mplus file</td>
</tr>
<tr>
<td>DataName</td>
<td>name of .inp Mplus file</td>
</tr>
<tr>
<td>ThreshName</td>
<td>string threshold name given by M Plus</td>
</tr>
<tr>
<td>InitialData</td>
<td>dataframe used</td>
</tr>
<tr>
<td>Directry</td>
<td>the working directory which contains the folder which contains .inp and .out files</td>
</tr>
<tr>
<td>NADes</td>
<td>the value for missing data</td>
</tr>
</tbody>
</table>

**Details**

Threshold names given by M Plus and consequently many output matrices in this package are designated starting with 1 and ordered, and this function converts 1 or other designation to the actual category in the data.

**Value**

returns string of variable value which is used for the threshold.

**Note**

Make sure working directory is set to location of folder containing folder which contains .out mplus files.

**Author(s)**

William Terry

**References**

M Plus
Examples

```r
## Not run:
InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
zzz=AllSummary(xxx[[1]])[[2]]
qqq=AddOnAllInd(xxx[[1]],zzz)
NewNamesThresh("New_1000","NewData_1000","X$1",InitD)

## End(Not run)
```

<table>
<thead>
<tr>
<th>NumCat</th>
<th>Number of Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

used as a helper function to return number of categories of categorical variable

Usage

```r
NumCat(ColDes, DataMat, NADes)
```

Arguments

- **ColDes**: which column number is the number for the data vector for which number of unique categories to be returned
- **DataMat**: this the dataframe or can be a matrix of data with different variables as the columns
- **NADes**: this is the value to be used for the missing category of data

Details

helper function which specifies number of unique categories excluding missing category of a column of a dataframe or matrix

Value

- **comp1**: returns numeric value of number of unique categories excluding missing category for vector of data

Note

No notes

Author(s)

William Terry
NumEndFile

References
No References

Examples

InitD=Simulate(MissingYN=1)
NumCat(2,InitD,c(99))

NumEndFile Extract Number From INP and OUT Files

Description
helper function used to find last input and output file which is the highest numbered file

Usage
NumEndFile(NameOfFile, pattern1, pattern2)

Arguments
NameFile list of strings
pattern1 the string pattern before which the desired number is located
pattern2 2nd character of string pattern before which the desired number is located

Details
helper function not to be used by regular user

Value
returns number

Note
No notes

Author(s)
William Terry

References
No reference
OnlyNumberElement

Examples

```r
files <- c("new_1.out","new_10.out","new_11.out","new_12.out")
hh <- lapply(strsplit(files,"_"), NumEndFile, pattern1=".out", pattern2="o")
```

Description

simple helper function to return only specified list element of set of lists to be used with apply functions

Usage

```r
OnlyNumberElement(AllData, Number)
```

Arguments

- `AllData`: list of lists
- `Number`: index number of list element to return

Details

created for easy indexing of lists of lists with apply functions

Value

- `comp1`: designated list element

Note

No notes

Author(s)

William Terry

References

No references

Examples

```r
Dat <- list(c(list("new1"),list("new2")),c(list("Old1"),list("Old2")))
AllFileNames1 <- sapply(Dat, OnlyNumberElement, 1)
```
**ParseTotalEffects**  
*Parse Total, Direct, and Indirect Effects*

**Description**

read .out M Plus files to return total, indirect, and direct effect p-values and names of effects

**Usage**

`ParseTotalEffects(OutFile, FileName, Directry)`

**Arguments**

- **OutFile**
  a read, scanned string of the .out M Plus file, see example
- **FileName**
  the name of the folder where the .out files are stored for backward selection
- **Directry**
  the working directory used to store the folders containing .out files

**Details**

the Total effects and Total Indirect Effects results are returned reading IND statements output, and the Direct Effects results are returned by reading the Model ON statements

**Value**

returns a list with the following objects:

- **TotalPVals**
  A list where each element is the p-value of the Total Effect found under Model Indirect IND statements
- **INDPVals**
  A list where each element is the p-value of the Total Indirect Effect found under Model Indirect IND statements
- **INDNames**
  A matrix where each row contains the names of the variables found under Model Indirect IND statements corresponding to the Total and Total Indirect p-values respectively
- **DirectPVals**
  A list where each element is the p-value of the Direct Effect found under Model created using ON statements
- **DirectNames**
  A matrix where each row contains the names of the variables found under Model ON statements corresponding to the Direct Effect p-values respectively

**Note**

this function is a helper function used in `AllBackwardSelect` to determine which variable relations to delete

**Author(s)**

William Terry
ParseTotalEffects2

Parse and Extract Total Effects

Description

finds names and values of total effects and returns as lists

Usage

ParseTotalEffects2(OutFile, StandardError = FALSE, PVal = FALSE, Indirect = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutFile</td>
<td>name of folder where .inp and .out files are located</td>
</tr>
<tr>
<td>StandardError</td>
<td>if TRUE then standard errors are returned default is false and if also PVal is false then effects returned</td>
</tr>
<tr>
<td>PVal</td>
<td>if TRUE then p values for total effects are returns default is FALSE</td>
</tr>
<tr>
<td>Indirect</td>
<td>if 1 then total indirect effects are returned if 0 then total effects are returned</td>
</tr>
</tbody>
</table>

Details

helper function for TotalRiskRatios, CreateTotalEffMat and all other total effect functions and does not need to be used by regular user
Value

TotalVals 1st list is values returned either total effects or standard errors of total effects or p values of total effects

TotalEffectNames list of names of total effects

Note

helper function, but useful for parsing if modifying package or certain functions

Author(s)

William Terry

References

M Plus

See Also

TotalRiskRatios and CreateTotalEffMat

Examples

```r
## Not run:

InitD=Simulate(MissingYN=0,exampleNum=3)
xxx=Initialize(InitD,NumImpute=0,WhichCat=c(1,1,1,1,1,0,0),PasteIND=1)
ggg=AllBackwardSelect(xxx[[1]]))
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
Dirrectry=getwd()
LastFileOut=AllDat1[[1]][[1]]
TO=paste(Directry,"/",LastFileOut,sep="")
files <- list.files(path=TO,pattern = ".out$")
hh=lapply(strsplit(files,"_"),NumEndFile,pattern1=".out",pattern2='out')
hh=as.numeric(paste(hh))
LastFileName=files[which(hh==max(hh))]
x=scan(paste(TO,"/",LastFileName,sep=""),what=character())
GG=ParseTotalEffects2(x)

## End(Not run)
```
**PathNames**

*M Plus Path description*

**Description**
converts path statements from the initialize matrix to MPlus format

**Usage**
PathNames(rowNum, InputInitializeMat)

**Arguments**
- **rowNum** the row number of the initialize matrix from which to return path name
- **InputInitializeMat** the square matrix of all variable names as rows and columns with 1s for include relationship and 0s for don’t include relationship

**Details**
helper function not to be used as standalone by regular user

**Value**
- **path** string which represents path relationship for M plus .inp file

**Note**
No Notes

**Author(s)**
William Terry

**References**
M Plus

**Examples**
InitD=Simulate()
cc=CreateInitializeMatrix(InitD,WhichCat=c(1,1,1,1,0))
PathNames(2,cc)
Description

a data set for example purposes

Usage

Simulate(n = 1000, seedNum = 1000, MissingYN = 0, exampleNum = 1)

Arguments

n                the sample size
seedNum            the seed number for random number generation
MissingYN      0 is no missing values 1 is missing values
exampleNum  1, 2, or 3 for the example to use

Details

generates data frame

Value

returns generated dataframe

Note

Note in the examples that the names of the variables all start with a capital letter and contain no special symbols or spaces as required in this version

Author(s)

William Terry

References

no references

Examples

InitD=Simulate(n=500,seedNum=1001,MissingYN=1,exampleNum=1)
**SpecialMatch**

**Match and delete elements**

**Description**

used to delete superfluous spaces when parsing M Plus output

**Usage**

SpecialMatch(List0, DelL)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List0</td>
<td>list of lists which contain strings as each element</td>
</tr>
<tr>
<td>DelL</td>
<td>list of lists of same length as List0 and those elements which are NA are kept for output while all other elements are removed</td>
</tr>
</tbody>
</table>

**Details**

see example

**Value**

input list of lists without elements specified by DelL

**Note**

this is a helper function that does not need to be used by regular user

**Author(s)**

William Terry

**References**

No references

**Examples**

```r
x=c()
x=c(1.388, 0.068, 20.514, 0.000)
x=c(x," A$1 1.858 0.139 13.340 0.000"
)x=c(x," B$1 1.426 0.081 17.542 0.000"
)x=c(x," C$1 1.644 0.092 17.934 0.000"
Thresh=x
Thresh=apply(Thresh, strsplit,"
DL=apply(AllThresh,match,""
AllThreshNoSp=SpecialMatch(AllThresh,DL)
```
TotalRiskRatios  

*Calculate Risk Ratios for total effects*

**Description**

Risk Ratios for the total effects are calculated using threshold values obtained for direct effects and the total effect values see in AllTotalEffOutput.

**Usage**

TotalRiskRatios(AllDat, Directry = getwd(), InputDepVal = 1)

**Arguments**

- `AllDat`: output from `ConvertData2`
- `Directry`: this the path where the folder specified by AllNames is located and default is working directory
- `InputDepVal`: a number which is the value of the dependent variable used for risk ratio which is compared to 0 in risk ratio calculation, default value is 1

**Details**

Total effects are sum of direct and indirect effects where indirect effects are the product of all direct effects in the chain which comprises the indirect variable chain, this total effect value is converted to its probability using the threshold values for direct effects and multiplying the total effect with the specified input value.

**Value**

returns data frame with risk ratios in place of total effects

**Note**

of course backwards selection and other functions must run first, see example

**Author(s)**

William Terry

**References**

M Plus

**See Also**

codeAllTotalEffOutput, codeAverageRRs
Examples

## Not run:

InitD=Simulate(MissingYN=1)
xxx=Initialize(InitD,NumImpute=3,WhichCat=c(1,1,1,0))
ggg=AllBackwardSelect(xxx[[1]])
yyy=AllSummary2(xxx[[1]])
AllDat1=ConvertData2(xxx[[1]],xxx[[2]],yyy[[1]])
RRT1=lapply(AllDat1,TotalRiskRatios,InputDepVal=1)

## End(Not run)

WriteInitialInpFile  Write Inp File

Description

function writes first M Plus .inp files for each imputed data set or just one file if no imputed data sets

Usage

WriteInitialInpFile(AllData, InitialData, InputInitializeMat, IndList, Directry, NADes, WhichCat, WhichImpute, WhichRowsImp, AllMethods)

Arguments

- **AllData**: list of length three with the first element a list of strings with each string as the name of the folder where .inp and .out files are written, the second element is a list of strings with each string the name of the .dat file containing imputed data set or one string name of data set if no imputations, and the third element is a list of numbers of the imputation seed used by MICE for imputation which is empty if there are no imputed data sets
- **InitialData**: dataframe of initial data
- **InputInitializeMat**: square dataframe where row name and column names are equal and are the variable names of InitialData with 1 for include path and 0 for exclude created by CreateInitializeMatrix
- **IndList**: the indirect statements to write created by AddOnINDStatements
- **Directry**: list of length number of variables in InitialData with 1 for categorical and 0 for continuous variables
- **NADes**: a list of length 1 with the value for missing data
- **WhichCat**: list of length number of variables in InitialData with 1 for categorical and 0 for continuous variables
- **WhichImpute**: list of length of names of InitialData specifying ’1’ for Impute column and ’0’ for no Imputation
WriteInitialImpFile

WhichRowsImp list of length of number of rows of InitialData specifying '1' for Impute row and '0' for no Imputation

AllMethods list of length 3 for MICE imputation for binary, unordered categorical, and continuous imputations default in Initialize set as "logreg" for binary imputation "polr" for categorical imputation "pmm" for continuous

Details

this is a helper function for Initialize and should not be used by regular user

Value

output is string which is the .inp file which is to be written to file using Initialize function

Note

helper function. Also note that the input file in this version should not be modified by the user since some of the line numbers are set for reading and should not be altered

Author(s)

William Terry

References

MPlusAutomation MICE

See Also

Initialize

Examples

```r
# Not run:
InitD=Simulate(MissingYN=1)
NumImpute=3
startSeedImputations=1000
DataFileName="NewData"
NameFile="New"
Dirctry=getwd()
NADes=c(-99)
WhichCat=c(1,1,1,1,0)
AllMethods=c("logreg","polr","pmm")
WhichImpute=rep(1,ncol(InitD))
WhichRowsImp=c(1:nrow(InitD))
NumImplist=seq(startSeedImputations,(startSeedImputations+NumImpute-1),1)
NameFile=paste0(NameFile,"_")
DataFileName=paste0(DataFileName,"_")
AllData=ConvertData3(NameFile,DataFileName,NumImplist)
InputInitializeMat=CreateInitializeMatrix(InitD,WhichCat,empty=FALSE)
IndList=AddOnINDStatements(InputInitializeMat,PasteIND=1)
```
WRt=lapply(AllData,WriteInitialInpFile,InitD,InputInitializeMat,IndList,Directry, NADes,WhichCat,WhichImpute,WhichRowsImp,AllMethods)

## End(Not run)
Index

*Topic **Categorical**
  NumCat, 34

*Topic **Data**
  Simulate, 41

*Topic **Direct Effects**
  CountImputedEffect, 19
  CreateSummaryMats, 21
  ParseTotalEffects, 37

*Topic **Impute**
  DatImputations, 26
  Initialize, 28

*Topic **Index**
  OnlyNumberElement, 36

*Topic **Indirect Effects**
  AddOnAllInd, 4
  AddOnINDStatements, 6
  CreateSummaryMats2, 23
  ParseTotalEffects, 37

*Topic **Initialize**
  CreateInitializeMatrix, 20
  PathNames, 40
  WriteInitialInpFile, 44

*Topic **Parse**
  NumEndFile, 35
  ParseTotalEffects, 37
  ParseTotalEffects2, 38
  SpecialMatch, 42

*Topic **Risk Ratios**
  AverageRRs, 13
  CalculatRiskRatios, 14
  ConvertData2, 16
  NewBinseqWrap, 32
  TotalRiskRatios, 43

*Topic **Summary**
  AllSummary, 8
  CountImputedEffect, 19
  CreateSummaryMats, 21
  CreateTotalEffMat, 24
  CreateTotalSummary, 25

*Topic **Threshold**
  NewNamesThresh, 33

*Topic **Total Effects**
  AllTotalEffOutput, 11
  CreateTotalEffMat, 24
  ParseTotalEffects, 37
  TotalRiskRatios, 43

*Topic **Variable Selection**
  AllBackwardSelect, 7
  Initialize, 28
  MPlusBackwardSelect, 31
  PathSelectMP-package, 2

*Topic **data**
  Initialize, 28
  NewNamesThresh, 33

*Topic **helper**
  AddOnAllInd, 4
  AddOnINDStatements, 6
  CheckVarNames, 15
  ConvertData2, 16
  ConvertData3, 18
  CountImputedEffect, 19
  CreateSummaryMats, 21
  CreateSummaryMats2, 23
  DatImputations, 26
  mod, 30
  MPlusBackwardSelect, 31
  NewBinseqWrap, 32
  NumCat, 34
  NumEndFile, 35
  OnlyNumberElement, 36
  PathNames, 40
  SpecialMatch, 42
  WriteInitialInpFile, 44

*Topic **summary**
  AllSummary2, 10

*Topic **variables**
  CheckVarNames, 15

AddOnAllInd, 4, 6, 9, 23
AddOnINDStatements, 5, 6
AllBackwardSelect, 3, 7, 9, 29, 31, 37, 38
AllSummary, 8, 11, 20
AllSummary2, 3, 9, 10, 12, 22
AllTotalEffOutput, 11, 25, 26, 43
AverageRRs, 3, 13, 43

CalculatRiskRatios, 3, 14, 16, 17
CheckVarNames, 15
ConvertData2, 11, 14, 16, 43
ConvertData3, 18
CountImputedEffect, 19
CreateInitializeMatrix, 7, 20, 29, 44
CreateSummaryMats, 21
CreateSummaryMats2, 23
CreateTotalEffMat, 24, 38, 39
CreateTotalSummary, 3, 25

DatImputations, 26
Initialize, 2, 3, 5, 7, 9, 16, 21, 27, 28, 45

mice, 26, 27, 29
mod, 30
MplusAutomation, 8, 23, 29
MPlusBackwardSelect, 31

NewBinseqWrap, 32
NewNamesThresh, 33
NumCat, 34
NumEndFile, 35

OnlyNumberElement, 36

ParseTotalEffects, 37
ParseTotalEffects2, 38
PathNames, 40
PathSelectMP (PathSelectMP-package), 2
PathSelectMP-package, 2

Simulate, 3, 41
SpecialMatch, 42

TotalRiskRatios, 3, 38, 39, 43

WriteInitialInpFile, 5, 44