

Package ‘PKreport’

July 2, 2014

Title A reporting pipeline for checking population pharmacokinetic model assumption

Version 1.5

Date 2014-03-16

Depends R (>= 2.7.0), methods, lattice, ggplot2

Author Xiaoyong Sun

Maintainer Xiaoyong Sun <johnsunx1@gmail.com>

Description PKreport aims to 1) provide automatic pipeline for users to visualize data and models. It creates a flexible R framework with automatically generated R scripts to save time and cost for later usage; 2) implement an archive-oriented management tool for users to store, retrieve and modify figures. 3) offer powerful and convenient service to generate high-quality graphs based on two R packages: lattice and ggplot2.

License GPL (>= 2)

LazyData true

Repository CRAN

Date/Publication 2014-03-19 00:18:16

NeedsCompilation no

R topics documented:

PKreport-package	2
non.coi-methods	3
non.cor-methods	3
non.cov-methods	4
non.lst-methods	4
non.lst.meth-methods	4
non.lst.objs-methods	4

non.lst.objt-methods	5
non.lst.objv-methods	5
non.lst.term-methods	5
non.phi-methods	5
non.select-methods	6
non.tab-methods	6
nonmem-class	6
pdata	8
PKadjust	9
PKclean	9
PKcode	10
PKconfig	10
PKdata	11
PKfigure	12
PKnum	13
PKoutput	13
PKreport.1	14
PKreport.2	15
PKreport.3	15
PKreport.4	16
PKreport.5	17
PKreport.6	17
PKreport.7	18
PKreport.8	19
PKshow	19

Index	21
--------------	-----------

PKreport-package	<i>An automatically pipeline for population pharmacokinetic models</i>
------------------	--

Description

This package provides an automatically pipeline, an R package called PKreport, for population pharmacokinetic models to test model assumptions, visualize data and diagnose models.

Details

Package:	PKreport
Type:	Package
Version:	1.0
Date:	2010-06-28
License:	GNU license

Author(s)

Xiaoyong Sun Maintainer: johnsunx1<johnsunx1@gmail.com>

Examples

```
# setup configuration
general.list <- list(save.format="bmp", width = 480, height = 480, package=2)
hist.list <- list(type=c("count"), layout=c(1,1), ind.layout=c(5,5))
scatter.list <- list(span=0.25, type=c("p", "smooth"), layout=c(1,1), ind.layout=c(5,5))

var.name <- list(ID="ID", DV="CONC", TIME="TIME", PRED="PRED", RES="RES",
WRES="WRES", IPRE="IPRE", IDV=c("CLCR", "WT"), COV=c("WT", "AGE"),
ETA=c("ETA1", "ETA2"), PARA=c("CL", "V"))

data(pdata)
# PKdata(data=pdata, match.term=var.name)

# PKconfig(general.list, hist.list, scatter.list)
# PKfigure(pdata, 1)
# PKshow()
# PKclean()
```

non.coi-methods

Output coi file

Description

Methods for function non.coi in Package 'PKreport'

Methods

object = "nonmem" Access coi content in the file.coi slot.

non.cor-methods

Output cor file

Description

Methods for function non.cor in Package 'PKreport'

Methods

object = "nonmem" Access cor content in the file.cor slot.

non.cov-methods *Output cov file*

Description

Methods for function non.cov in Package 'PKreport'

Methods

object = "nonmem" Access cov content in the file.cov slot.

non.lst-methods *Output lst content*

Description

Methods for function non.lst in Package 'PKreport'

Methods

object = "nonmem" Access lst content in the file.lst slot.

non.lst.meth-methods *output method in lst file*

Description

Methods for function non.lst.meth in Package 'PKreport'

Methods

object = "nonmem" Access method section (\#meth tag) in the file.lst slot.

non.lst.objs-methods *Output objective function standard deviation in lst file*

Description

Methods for function non.lst.objs in Package 'PKreport'

Methods

object = "nonmem" Access objective function standard deviation (\#OBS tag) in the file.lst slot.

non.lst.objt-methods *Output objective function in lst file*

Description

Methods for function non.lst.objt in Package 'PKreport'

Methods

object = "nonmem" Access objective function (\#OBJT tag) in the file.lst slot.

non.lst.objv-methods *Output objective function value in lst file*

Description

Methods for function non.objv in Package 'PKreport'

Methods

object = "nonmem" Access objective function value (\#OBJV tag) in the file.lst slot.

non.lst.term-methods *output analysis section in lst file*

Description

Methods for function non.lst.term in Package 'PKreport'

Methods

object = "nonmem" Access analysis section (\#TERM, \#TERE tag) in the file.lst slot.

non.phi-methods *Output phi file*

Description

Methods for function non.phi in Package 'PKreport'

Methods

object = "nonmem" Access phi content in the file.coi slot.

non.select-methods *Output selected lines in lst files*

Description

Methods for function non.select in Package 'PKreport'

Methods

object = "nonmem" output the selected lines in lst files.

non.tab-methods *Output tab content in tab file*

Description

Methods for function non.tab in Package 'PKreport'

Methods

object = "nonmem" Access tab content in the file.tab slot.

nonmem-class *Class nonmem: contain and describe all nonmem output*

Description

This is a class representation for nonmem output.

Objects from the Class

Objects can be created by calls of the form `new("nonmem", output.lst, output.tab, output.dir, delim, ...)`.

This creates a nonmem object containing all nonmem output.

Slots

- `file.cov`: Object of class "list". This list includes title (character) and data (data.frame) for cov file.
- `file.cor`: Object of class "list". This list includes title (character) and data (data.frame) for cor file.
- `file.coi`: Object of class "list". This list includes title (character) and data (data.frame) for coi file.
- `file.phi`: Object of class "list". This list includes title (character) and data (data.frame) for phi file.
- `file.lst`: Object of class "character". This character vector contains the information from NONMEM lst file.
- `method`: Object of class "character". This character vector matches `\#METH` tag in lst file and contains the estimation method.
- `analysis`: Object of class "list". This list matches text between `\#TERM` tag and `\#TERE` tag in lst file and contains the analysis information.
- `objt`: Object of class "character". This character vector matches `\#OBJT` tag in lst file and describes the objective function.
- `objv`: Object of class "character". This character vector matches `\#OBJV` tag in lst file and describes the objective function value.
- `objs`: Object of class "character". This character vector matches `\#OBJT` tag in lst file and describes the objective function standard deviation.
- `tabid`: Object of class "character". This character vector is from the first line of tab file and describes the title of tab file.
- `tabdata`: Object of class "data.frame". This data frame matches data in tab file and describes the title of tab file.

Methods**Class-specific methods**

- `non.lst(nonmem)`: Access lst content in the `file.lst` slot.
- `non.lst.meth(nonmem)`: Access method section (`\#meth` tag) in the `file.lst` slot.
- `non.lst.term(nonmem)`: Access analysis section (`\#TERM`, `\#TERE` tag) in the `file.lst` slot.
- `non.lst.objt(nonmem)`: Access objective function (`\#OBJT` tag) in the `file.lst` slot.
- `non.lst.objv(nonmem)`: Access objective function value (`\#OBJV` tag) in the `file.lst` slot.
- `non.lst.objs(nonmem)`: Access objective function standard deviation (`\#OBS` tag) in the `file.lst` slot.
- `non.tab(nonmem)`: Access tab content in the `file.tab` slot.
- `non.cov(nonmem)`: Access cov content in the `file.cov` slot.
- `non.cor(nonmem)`: Access cor content in the `file.cor` slot.
- `non.coi(nonmem)`: Access coi content in the `file.coi` slot.
- `non.phi(nonmem)`: Access phi content in the `file.phi` slot.

`non.select(nonmem, lines, sep)`: Select the lines in the lst file.

Standard generic methods

`initialize(object, output.lst, output.tab, output.dir)`: Object instantiation. `output.lst`: the file name for lst file from NONMEM 7; `output.tab`: the filename for tab file from any NONMEM version; `output.dir(optional)`: the output directory including cor, cov, coi and phi files from NONMEM 7. `output.lst` and `output.dir` only works for NONMEM 7.

`nonmem` instances are created through `new("nonmem", ...)`. The arguments to `new` should include `output.lst` and `output.tab`. If `output.dir` is missing, `NULL` will be assigned to proper slots.

Author(s)

Xiaoyong Sun

Examples

```
showClass("nonmem")
```

pdata

A sample data

Description

A sample data.

Usage

```
data(pdata)
```

Author(s)

Xiaoyong Sun

Examples

```
data(pdata)
```

PKadjust	<i>Update figures</i>
----------	-----------------------

Description

This function is to update or modify certain figures with figure ID for particular purpose.

Usage

```
PKadjust(figno, save=FALSE,...)
```

Arguments

figno	the figure number generated from PKshow().
save	logical. To only display the modified figure, users need chose FALSE; to save in the management system, users need choose TRUE.
...	All new figure configurations for this figure number.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

PKclean	<i>Clean related archives</i>
---------	-------------------------------

Description

This function is to delete all archives (file directories and figures) and clean global variables in R environment.

Usage

```
PKclean()
```

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# PKclean()
```

PKcode	<i>Generate R scripts</i>
--------	---------------------------

Description

This function is to generate R scripts. To improve efficiency and help users to generate high-quality figures, users have option to modify related R scripts to meet their specific requirements. All generated R scripts match the order of figures generated in PKshow().

Usage

```
PKcode(filename="PKcode.txt")
```

Arguments

filename the file name to store the R scripts.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

PKconfig	<i>Data configuration</i>
----------	---------------------------

Description

This function is to configure data for analysis.

Usage

```
PKconfig(general.list, hist.list, scatter.list)
```

Arguments

general.list a list. It includes figure configuration: save.format, width, height, and also graphic packages (0: use only lattice package. 1: use only ggplot2 package. 2: use both packages).

hist.list a list. It includes histogram configuration: type, layout, ind.layout (for individual plots).

scatter.list a list. It includes scatterplot configuration: type, layout, span, ind.layout (for individual plots).

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# setup configuration
general.list <- list(save.format="bmp", width = 480, height = 480, package=2)
hist.list <- list(type=c("count"), layout=c(1,1), ind.layout=c(5,5))
scatter.list <- list(span=0.25, type=c("p", "smooth"), layout=c(1,1), ind.layout=c(5,5))

PKconfig(general.list, hist.list, scatter.list)
```

PKdata

Data input

Description

This function is to read data, match default naming system to data variables, and setup global variables.

Usage

```
PKdata(data, match.term=NULL)
```

Arguments

data	a data frame to analyze.
match.term	a list. It matches the package metrics to the variable names in the data.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```

# setup configuration
general.list <- list(save.format="bmp", width = 480, height = 480, package=2)
hist.list <- list(type=c("count"), layout=c(1,1), ind.layout=c(5,5))
scatter.list <- list(span=0.25, type=c("p", "smooth"), layout=c(1,1), ind.layout=c(5,5))

var.name <- list(ID="ID", DV="CONC", TIME="TIME", PRED="PRED", RES="RES",
WRES="WRES", IPRE="IPRE", IDV=c("CLCR", "WT"), COV=c("WT", "AGE"),
ETA=c("ETA1", "ETA2"), PARA=c("CL", "V"))

data(pdata)
# PKdata(data=pdata, match.term=var.name)

# PKconfig(general.list, hist.list, scatter.list)
# PKfigure(pdata, 1)
# PKshow()
# PKclean()

```

 PKfigure

Diagnose model and generate figures

Description

This function is for diagnosing specific models and generate figures.

Usage

```
PKfigure(pdata, methods, clean)
```

Arguments

<code>pdata</code>	a data frame to analyze. Generally it is from tab file generated from NONMEM.
<code>methods</code>	a numeric vector. It includes all diagnostics methods according to Census, 1: Exploratory data analysis; 2: Individual plots; 3: Goodness-of-fit plots; 4: Structural model diagnostics; 5: Residual model diagnostics; 6: Parameters diagnostics; 7: Covariate model; 8: Random effects.
<code>clean</code>	a logical value indicating whether or not to keep results from previous PKreport for final report.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# PKfigure(pdata, c(3,6,8))
```

PKnum	<i>Output numerical value</i>
-------	-------------------------------

Description

This function is to convert data frame or matrix (the output from non.select) to number.

Usage

```
PKnum(exp.data)
```

Arguments

exp.data a data frame or matrix selected from lst file.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# exp.data <- non.select(myclass, c(50:56))
# options(scipen=100)
# PKnum(exp.data)
# options(scipen=-100)
# PKnum(exp.data)
```

PKoutput	<i>Output all results to folders and files</i>
----------	--

Description

This function is to translate analysis from PKreport() to folders and files in the current working directory.

Usage

```
PKoutput(nonmemObj, table.Rowv, table.Colv)
```

Arguments

nonmemObj	a object of class NONMEM.
table.Rowv	determine whether the row dendrogram should be reordered in the heatmap-like table.
table.Colv	determine whether the column dendrogram should be reordered in the heatmap-like table.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# PKoutput(nonmemObj)
```

PKreport.1

Exploratory data analysis

Description

This function is for exploratory data analysis.

Usage

```
PKreport.1(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.1(pdata)
```

PKreport.2

Individual plots

Description

This function is for individual plots.

Usage

```
PKreport.2(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.2(pdata)
```

PKreport.3

Goodness-of-fit plots

Description

This function is for goodness-of-fit plots.

Usage

```
PKreport.3(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.3(pdata)
```

PKreport.4

Structural model diagnostics

Description

This function is for structural model diagnostics.

Usage

```
PKreport.4(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.4(pdata)
```

PKreport.5

Residual model diagnostics

Description

This function is for residual model diagnostics.

Usage

```
PKreport.5(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.5(pdata)
```

PKreport.6

Parameters diagnostics

Description

This function is for parameters diagnostics.

Usage

```
PKreport.6(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.6(pdata)
```

PKreport.7

Covariate model diagnostics

Description

This function is for covariate model diagnostics.

Usage

```
PKreport.7(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.7(pdata)
```

PKreport.8	<i>Random effects diagnostics</i>
------------	-----------------------------------

Description

This function is for random effects diagnostics.

Usage

```
PKreport.8(pdata)
```

Arguments

pdata a data frame.

Value

Input is a data frame.

Author(s)

Xiaoyong Sun

Examples

```
# PKreport.8(pdata)
```

PKshow	<i>Display results</i>
--------	------------------------

Description

This function is to display results from analysis.

Usage

```
PKshow(nonmemObj, table.Rowv, table.Colv)
```

Arguments

nonmemObj a object of class NONMEM.
table.Rowv determine whether the row dendrogram should be reordered in the heatmap-like table.
table.Colv determine whether the column dendrogram should be reordered in the heatmap-like table.

Details

Please see vignettes for details.

Author(s)

Xiaoyong Sun

Examples

```
# PKshow(nonmemObj, table.Rowv=TRUE, table.Rowv=TRUE)

# Only generate figure report
# PKshow()
```

Index

- *Topic **classes**
 - nonmem-class, 6
- *Topic **methods**
 - non.coi-methods, 3
 - non.cor-methods, 3
 - non.cov-methods, 4
 - non.lst-methods, 4
 - non.lst.meth-methods, 4
 - non.lst.objs-methods, 4
 - non.lst.objt-methods, 5
 - non.lst.objv-methods, 5
 - non.lst.term-methods, 5
 - non.phi-methods, 5
 - non.select-methods, 6
 - non.tab-methods, 6
 - pdata, 8
 - PKadjust, 9
 - PKclean, 9
 - PKcode, 10
 - PKconfig, 10
 - PKdata, 11
 - PKfigure, 12
 - PKnum, 13
 - PKoutput, 13
 - PKreport.1, 14
 - PKreport.2, 15
 - PKreport.3, 15
 - PKreport.4, 16
 - PKreport.5, 17
 - PKreport.6, 17
 - PKreport.7, 18
 - PKreport.8, 19
 - PKshow, 19
- *Topic **package**
 - PKreport-package, 2
- initialize, nonmem-method
(nonmem-class), 6
- non.coi (nonmem-class), 6
- non.coi, nonmem-method (nonmem-class), 6
- non.coi, nonmem-method
(non.coi-methods), 3
- non.coi-methods, 3
- non.cor (nonmem-class), 6
- non.cor, nonmem-method (nonmem-class), 6
- non.cor, nonmem-method
(non.cor-methods), 3
- non.cor-methods, 3
- non.cov (nonmem-class), 6
- non.cov, nonmem-method (nonmem-class), 6
- non.cov, nonmem-method
(non.cov-methods), 4
- non.cov-methods, 4
- non.lst (nonmem-class), 6
- non.lst, nonmem-method (nonmem-class), 6
- non.lst, nonmem-method
(non.lst-methods), 4
- non.lst-methods, 4
- non.lst.meth (nonmem-class), 6
- non.lst.meth, nonmem-method
(nonmem-class), 6
- non.lst.meth, nonmem-method
(non.lst.meth-methods), 4
- non.lst.meth-methods, 4
- non.lst.objs (nonmem-class), 6
- non.lst.objs, nonmem-method
(nonmem-class), 6
- non.lst.objs, nonmem-method
(non.lst.objs-methods), 4
- non.lst.objs-methods, 4
- non.lst.objt (nonmem-class), 6
- non.lst.objt, nonmem-method
(nonmem-class), 6
- non.lst.objt, nonmem-method
(non.lst.objt-methods), 5
- non.lst.objt-methods, 5
- non.lst.objv (nonmem-class), 6
- non.lst.objv, nonmem-method

- (nonmem-class), 6
- non.lst.objv, nonmem-method
 - (non.lst.objv-methods), 5
- non.lst.objv-methods, 5
- non.lst.term (nonmem-class), 6
- non.lst.term, nonmem-method
 - (non.lst.term-methods), 5
- non.lst.term-methods, 5
- non.phi (nonmem-class), 6
- non.phi, nonmem-method (nonmem-class), 6
- non.phi, nonmem-method
 - (non.phi-methods), 5
- non.phi-methods, 5
- non.select (nonmem-class), 6
- non.select, nonmem-method
 - (nonmem-class), 6
- non.select, nonmem-method
 - (non.select-methods), 6
- non.select-methods, 6
- non.tab (nonmem-class), 6
- non.tab, nonmem-method (nonmem-class), 6
- non.tab, nonmem-method
 - (non.tab-methods), 6
- non.tab-methods, 6
- non.term, nonmem-method (nonmem-class),
6
- nonmem-class, 6

pdata, 8

- PKadjust, 9
- PKclean, 9
- PKcode, 10
- PKconfig, 10
- PKdata, 11
- PKfigure, 12
- PKnum, 13
- PKoutput, 13
- PKreport (PKreport-package), 2
- PKreport-package, 2
- PKreport.1, 14
- PKreport.2, 15
- PKreport.3, 15
- PKreport.4, 16
- PKreport.5, 17
- PKreport.6, 17
- PKreport.7, 18
- PKreport.8, 19
- PKshow, 19