

# Package ‘nonmem2R’

May 9, 2026

**Type** Package

**Title** Loading NONMEM Output Files with Functions for Visual Predictive Checks (VPC) and Goodness of Fit (GOF) Plots

**Version** 0.2.5

**Author** Magnus Astrand

**Maintainer** Magnus Astrand <magnus.astrand@astrazeneca.com>

**Description** Loading NONMEM (NONlinear Mixed-Effect Modeling, <<https://www.iconplc.com/solutions/technologies/nonmem/>>) and PSN (Perlspeaks-NONMEM, <<https://uupharmacometrics.github.io/PsN/>>) output files to extract parameter estimates, provide visual predictive check (VPC) and goodness of fit (GOF) plots, and simulate with parameter uncertainty.

**License** Unlimited

**Depends** R (>= 3.0.0), ggplot2, gridExtra (>= 2.3)

**Imports** mvtnorm, lattice, latticeExtra, MASS, splines2, reshape2

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-03-11 17:30:02 UTC

## Contents

add.caption . . . . .	2
basic.eta.GOF . . . . .	3
basic.GOF4 . . . . .	4
covload . . . . .	6
do.cat.GOF . . . . .	7
do.individual.GOF . . . . .	8
do.multi.GOF . . . . .	9

do.one.GOF . . . . .	10
eta.cov.GOF . . . . .	11
extload . . . . .	13
extToTable . . . . .	14
get.GOF.dictionary . . . . .	15
get.label . . . . .	17
get.model.path . . . . .	17
grid.sim . . . . .	18
histGOF . . . . .	19
merge2GOF . . . . .	20
modload . . . . .	21
position_scale . . . . .	21
qqnormGOF . . . . .	22
stat_corr . . . . .	23
stat_denx . . . . .	24
stat_dnorm . . . . .	25
stat_hmean . . . . .	26
stat_hmedian . . . . .	27
sumoR . . . . .	28
test.grid.sim . . . . .	29
vpcfig . . . . .	29
vpcfig2 . . . . .	32
vpcfig3 . . . . .	36
<b>Index</b>	<b>39</b>

---

add.caption	<i>Add caption to ggplot object</i>
-------------	-------------------------------------

---

## Description

Adds caption text as returned by `get.caption` to a ggplot object

## Usage

```
add.caption(p, control = GOF.control())
```

## Arguments

<code>p</code>	ggplot object
<code>control</code>	an optional list of control settings. See <code>GOF.control</code> for the names of the settable control values and their effect.

---

`basic.eta.GOF`*Histogram and/or QQ-norm GOF for ETA's*

---

**Description**

Provides histogram and/or QQ-norm GOF for all ETA's included in the input data.frame with caption added as returned by `get.caption`

**Usage**

```
basic.eta.GOF(  
  data,  
  title = "",  
  global.ggplot.options = NULL,  
  type = c("both", "qqnorm", "hist"),  
  ETA.subset = NULL,  
  refline = TRUE,  
  drop.fixed = TRUE,  
  id.column = "ID",  
  standardize = TRUE,  
  bins = NA,  
  control = GOF.control()  
)
```

```
eta.hist.GOF(  
  data,  
  title = "",  
  drop.fixed = TRUE,  
  refline = TRUE,  
  id.column = "ID",  
  standardize = TRUE,  
  bins = NA,  
  control = GOF.control()  
)
```

```
eta.qqnorm.GOF(  
  data,  
  title = "",  
  drop.fixed = TRUE,  
  refline = TRUE,  
  id.column = "ID",  
  standardize = TRUE,  
  control = GOF.control()  
)
```

**Arguments**

`data`            data.frame to plot

title	title
global.ggplot.options	ggplot option added to each ggplot before plotting/returning object
type	do only qq-norm plots (qqnorm), only histograms (hist), or do both (both)
ETA.subset	index for subset of ETA's to plot. If NULL (default) all ETA's are plotted.
refline	add reference line (default =TRUE) or not (FALSE)
drop.fixed	drop ETA's that are fixed, i.e. any ETA with the same value in all subjects
id.column	column name that indicate subject identifier
standardize	Standardize all ETA's (e.g. scale to unit variance)
bins	number of histogram bins, if NA, bins are set using Sturges' formula
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

### Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
basic.eta.GOF(sdtab)
```

---

basic.GOF4

*Basic 4- and 6-panel GOF*

---

### Description

basic.GOF4 provides a 4-panel GOF plot showing

- i) Observations (DV) vs population predictions (PRED)
- ii) Observations (DV) vs individual predictions (IPRED)
- iii)  $\sqrt{\text{abs}(\text{CWRES})}$  vs individual predictions (IPRED), and
- iv) CWRES vs TIME or other columns as set by idv

basic.GOF6 provides 2 additional GOF's showing histogram and qqnorm GOF's for CWRES.

Caption is added as returned by get.caption

### Usage

```
basic.GOF4(
  data,
  residual = "CWRES",
  idv1 = "IPRED",
  idv2 = "TIME",
  title = "",
  color = "",
```

```

    log.scale = FALSE,
    global.ggplot.options = NULL,
    refline = TRUE,
    add.loess = TRUE,
    control = GOF.control()
)

basic.GOF6(
  data,
  residual = "CWRES",
  idv1 = "IPRED",
  idv2 = "TIME",
  title = "",
  color = "",
  log.scale = FALSE,
  global.ggplot.options = NULL,
  refline = TRUE,
  add.loess = TRUE,
  bins = NA,
  control = GOF.control()
)

```

### Arguments

data	data.frame to plot
residual	column name for residuals, default="CWRES"
idv1	independent variable for plot of sqrt of absolute CWRES
idv2	independent variable for plot of CWRES
title	title
color	data columns to set different colors in plot, interpreted as factor
log.scale	use log scale for DV, IPRED and PRED (TRUE) or normal scale (FALSE)
global.ggplot.options	ggplot option added to each ggplot before plotting/returning object
refline	add reference line (default =TRUE) or not (FALSE)
add.loess	add loess smoother to plot (TRUE), or not (FALSE)
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.
bins	number of histogram bins, if NA, bins are set using Sturges' formula

### Examples

```

# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
## Example 4 panel basic GOF

```

```
basic.GOF4(subset(sdtab,DV>0),idv2="TAPD")
## Example 6 panel basic GOF
basic.GOF6(subset(sdtab,DV>0),idv1="TAPD",idv2="PRED")
```

---

covload	<i>Load the covariance matrix from a nonmem .cov output file.</i>
---------	---

---

## Description

Load the covariance matrix from a nonmem .cov output file. Either the covariance matrix of all THETA parameters (default) or the covariance matrix of all parameters, THETA, OMEGA and SIGMA.

For cov-files with multiple table results either only the last table result is loaded (`last.table.only=TRUE`, default) or all table results are loaded (`last.table.only=FALSE`). The result is then a named list of covariance matrix's, `LastTable` is the last table in the cov-files.

## Usage

```
covload(
  model,
  use.model.path = TRUE,
  last.table.only = TRUE,
  theta.only = TRUE
)
```

## Arguments

<code>model</code>	name of the cov file with or without the .cov extension. model may include full or relative path to the cov file. See examples.
<code>use.model.path</code>	Load file from a global defined model library ( <code>TRUE</code> =default). If so will look for a global character vector named <code>model.path</code>
<code>last.table.only</code>	Include only the last table result for cov files with multiple table results
<code>theta.only</code>	return covariance matrix of theta's only (default)

## Value

The covariance matrix or a named list of covariance matrix's

## Examples

```
##### Load the .cov file "run001.cov"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.cov", package = "nonmem2R")
# 2) Load the file using the covload function
covload(file1)
```

do.cat.GOF

*Y vs categorical X GOF-plot***Description**

GOF plot with boxplots of Y grouped by categorical data in X, with refine and with caption added as returned by get.caption

**Usage**

```
do.cat.GOF(
  data,
  x,
  y,
  color = "",
  add.points = TRUE,
  refine = c("abline", "href0", "hrefmean", "hrefmedian", "none"),
  title = "",
  lines.by.id = FALSE,
  id.column = "ID",
  fx = NULL,
  fy = NULL,
  control = GOF.control()
)
```

**Arguments**

data	data.frame to plot
x	character string with name of column for x
y	character string with name of column for y
color	data columns to set different colors in plot, interpreted as factor
add.points	add jittered points of data (TRUE), or not (FALSE)
refline	add reference line with intercept=0, slope=1 (abline), horizontal at y=0( href0), horizontal at y=mean of y( hrefmean), horizontal at y=median of y( hrefmedian), or don't add reference line (none)
title	title
lines.by.id	connect subjects by lines (TRUE), or don't (FALSE)
id.column	column name that indicate subject identifier
fx	function for transformation of x before plotting
fy	function for transformation of y before plotting
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

---

do.individual.GOF      *Individual GOF-plots*


---

### Description

DV, IPRED and PRED (default) individual GOF plots with one panel per subject. Caption added as returned by get.caption

### Usage

```
do.individual.GOF(
  data,
  x = "TIME",
  y = c("DV", "IPRED", "PRED"),
  type = c("point", "line", "line"),
  title = "",
  per.page = 20,
  fx = NULL,
  fy = NULL,
  ylab = "Observations/Predictions",
  equal.lims = TRUE,
  global.ggplot.options = NULL,
  control = GOF.control()
)
```

### Arguments

data	data.frame to plot
x	character string with name of column for x
y	character string vector with names of columns for y, default = c(DV, IPRED , PRED)
type	plot symbol or line for each y variable, length of type must match length of y
title	title
per.page	number of subjects/panels on each page
fx	function for transformation of x before plotting
fy	function for transformation of y before plotting
ylab	y axis label
equal.lims	use same x- and y-limits on all pages and panels (TRUE), or use allow to vary across pages (FALSE)
global.ggplot.options	ggplot option added to each ggplot before plotting/returning object
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

**Examples**

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
do.individual.GOF(subset(sdtab,DV>0 & ID<13))
```

---

do.multi.GOF                      *Multiple Y vs single X GOF-plot*

---

**Description**

X-Y GOF plots for multiple Y's and shared X with refines and data smoother and with caption added as returned by get.caption

**Usage**

```
do.multi.GOF(
  data,
  x,
  y,
  color = "",
  add.loess = TRUE,
  refline = c("none", "abline", "href0", "hrefmean", "hrefmedian"),
  title = "",
  lines.by.id = FALSE,
  id.column = "ID",
  fx = NULL,
  fy = NULL,
  ylab = "Observations/Predictions",
  control = GOF.control()
)
```

**Arguments**

data	data.frame to plot
x	character string with name of column for x
y	character string with name of column for y
color	data columns to set different colors in plot, interpreted as factor
add.loess	add loess smoother to plot (TRUE), or not (FALSE)
refline	add reference line with intercept=0, slope=1 (abline), horizontal at y=0( href0), horizontal at y=mean of y( hrefmean), horizontal at y=median of y( hrefmedian), or don't add reference line (none)
title	title
lines.by.id	connect subjects by lines (TRUE), or don't (FALSE)

id.column	column name that indicate subject identifier
fx	function for transformation of x before plotting
fy	function for transformation of y before plotting
ylab	y axis label
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

---

do.one.GOF

*X-Y GOF-plot*


---

### Description

X-Y GOF plot with refines and data smoother and with caption added as returned by get.caption

### Usage

```
do.one.GOF(
  data,
  x,
  y,
  color = "",
  add.loess = TRUE,
  refline = c("abline", "href0", "hrefmean", "hrefmedian", "none"),
  title = "",
  lines.by.id = FALSE,
  id.column = "ID",
  fx = NULL,
  fy = NULL,
  control = GOF.control()
)
```

### Arguments

data	data.frame to plot
x	character string with name of column for x
y	character string with name of column for y
color	data columns to set different colors in plot, interpreted as factor
add.loess	add loess smoother to plot (TRUE), or not (FALSE)
refline	add reference line with intercept=0, slope=1 (abline), horizontal at y=0( href0), horizontal at y=mean of y( hrefmean), horizontal at y=median of y( hrefmedian), or don't add reference line (none)
title	title
lines.by.id	connect subjects by lines (TRUE), or don't (FALSE)

id.column	column name that indicate subject identifier
fx	function for transformation of x before plotting
fy	function for transformation of y before plotting
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

### Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
do.one.GOF(subset(sdtab,DV>0),"IPRED","DV")
```

---

eta.cov.GOF

*Covariate and pairs ETA GOFs*

---

### Description

ETA's vs numerical covariates (eta.cov.GOF) and ETA's vs categorical covariates GOF plots on multiple or single pages with refines, data smoother, and caption added as returned by get.caption.

eta.pairs.GOF provides a pairs plot of all ETA's.

### Usage

```
eta.cov.GOF(
  data,
  covariates = c("AGE", "BWT"),
  ETA.subset = NULL,
  exclude.zero.ETA = FALSE,
  title = "",
  drop.fixed = TRUE,
  id.column = "ID",
  standardize = TRUE,
  refline = TRUE,
  type = c("all-in-one", "covariate-by-page", "eta-by-page"),
  layout = c("ETAbbyROW", "ETAbbyCOL"),
  add.loess = TRUE,
  control = GOF.control()
)
```

```
eta.cat.GOF(
  data,
  covariates = c("SEX"),
  ETA.subset = NULL,
  exclude.zero.ETA = FALSE,
```

```

title = "",
drop.fixed = TRUE,
id.column = "ID",
standardize = TRUE,
refline = TRUE,
type = c("all-in-one", "covariate-by-page", "eta-by-page"),
layout = c("ETAbYROW", "ETAbYCOL"),
add.points = TRUE,
control = GOF.control()
)

eta.pairs.GOF(
  data,
  ETA.subset = NULL,
  title = "",
  drop.fixed = TRUE,
  id.column = "ID",
  density2D = c("none", "upper", "lower"),
  standardize = TRUE,
  control = GOF.control()
)

```

### Arguments

data	data.frame to plot
covariates	covariates, list of character strings
ETA.subset	index for subset of ETA's to plot. If NULL (default) all ETA's are plotted.
exclude.zero.ETA	If set to TRUE any $ETA==0$ is excluded before plotting. This option is useful if the model have full shrinkage for subset of individuals e.g. for ETA on ED50 in patents on placebo or on dose==0.
title	title
drop.fixed	drop ETA's that are fixed, i.e. any ETA with the same value in all subjects
id.column	column name that indicate subject identifier
standardize	Standardize all ETA's (e.g. scale to unit variance)
refline	add reference line (default =TRUE) or not (FALSE)
type	do all ETA's and covariates on one page (all-in-one), one page for each covariate (covariate-by-page), or one page for each ETA (eta-by-page)
layout	Layout for ETA's and covariates
add.loess	add loess smoother to plot (TRUE), or not (FALSE)
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.
add.points	add jittered points of data (TRUE), or not (FALSE)
density2D	add 2D-density above (upper) or below (lower), or don't add (none)

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
eta.cov.GOF(sdtab,covariates=c("AGE","BWT"))
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
eta.cat.GOF(sdtab,covariates=c("SEXM"))
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab999", package = "nonmem2R")
sdtab<-read.table(file=file1,skip=1,header=TRUE)
set.script.name("MyScript.R")
eta.pairs.GOF(sdtab)
```

---

extload

*Load final parameter values from a nonmem .ext output file.*

---

## Description

Load parameter values from a nonmem .ext output file. For output files from FO, FOCE and IMP only the final parameter values are loaded. For output tables of MCMC and SAEM all parameter values from ITERATIONS>0 will be loaded unless positive.iterations=FALSE in case all the output from all iterations are loaded.

For ext-files with multiple table results either only the last table result is loaded (last.table.only=TRUE, default) or all table result are loaded (last.table.only=FALSE). All but the last table results are then returned as sub-lists to the last table result.

## Usage

```
extload(
  model,
  use.model.path = TRUE,
  positive.iterations.only = TRUE,
  last.table.only = TRUE
)
```

## Arguments

model	name of the ext file with or without the .ext extension. model may include full or relative path to the ext file. See examples.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
positive.iterations.only	Include only rows with ITERATIONS>0 from MCMC and SEAM table results (default=TRUE)

```
last.table.only
```

Include only the last table result for ext files with multiple table results

### Value

Named list including theta, theta.sd, omega, omega.sd, sigma, sigma.sd, and ofv. Here .sd is the vector(matrix) with standard errors estimated parameter values in theta(omega & sigma).

For MCMC output files each object are matrixes.

### Examples

```
##### Load the .ext file "run001.ext"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.ext", package = "nonmem2R")
# 2) Load the file using the extload function
extload(file1)
```

---

extToTable

*Compile parameter table suitable for reports*

---

### Description

Load parameter values from a nonmem .ext output file and compile to table suitable for reports. Format can be "wide" (wide=TRUE, default) with a similar look as from sumoR, or similar a 3 column layout (wide=FALSE).

### Usage

```
extToTable(
  model,
  use.model.path = TRUE,
  tableType = 2,
  wide = TRUE,
  format.estimate = "% -#6.4g",
  format.rse = "%#6.3g"
)
```

### Arguments

model	name of the ext file with or without the .ext extension. model may include full or relative path to the ext file. See examples.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
tableType	Table type for THETA's, OMEGA's and SIGMA's tableType=0: Present OMEGA and SIGMA as variance and covariances and display SE for THETA, OMEGA, SIGMA

tableType=1: Present OMEGA and SIGMA as variance and covariances and display RSE for THETA, OMEGA, SIGMA  
 tableType=2: Present OMEGA and SIGMA as standard-deviation and correlations and display RSE for THETA, OMEGA, SIGMA  
 tableType=3: Present OMEGA and SIGMA as standard-deviation and correlations and display SE for THETA, OMEGA, SIGMA  
 wide produce a wide 9-column table (wide=TRUE, default) or a thin 3-column table(wide=FALSE)  
 format.estimate format for estimated value, passed to sprintf  
 format.rse format for rse, passed to sprintf

**Value**

a character-matrix

**Examples**

```
##### Load the .ext file "run001.ext"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.ext", package = "nonmem2R")
# 2) Load the file using the extload function
extToTable(file1)
```

---

get.GOF.dictionary     *Get, set, and modify the global (or local) GOF parameters and GOF dictionary for labels*

---

**Description**

The current global GOF parameters and dictionary is automatically applied to every plot you draw. Use 'get.GOF.params' to get the current GOF parameters, and 'set.GOF.params' to change one or several parameters. To change settings for one GOF only you can use GOF.control as input to the GOF function. See details below for complete list of GOF parameters.

Similarly use 'get.GOF.dictionary' and 'set.GOF.dictionary' for the dictionary of labels. 'set.GOF.dictionary' also allows for adding new items to the dictionary.

Use 'set.ETA.labels' to set labels for ETA's in all plots

**Usage**

```
get.GOF.dictionary()

set.GOF.dictionary(...)

default.GOF.dictionary()
```

```

set.GOF.params(...)
get.GOF.params()
default.GOF.params()
set.ETA.labels(labels)
set.script.name(script.name)
GOF.control(...)

```

### Arguments

```

...          named list of GOF parameter or GOF dictionary for labels
labels       vector of character strings with the labels
script.name  character strings for script name to put at end of caption

```

### Details

Table below describe each of the global GOF parameters that can be modified by 'set.GOF.params'

Parameter	Default	Description
col.data	"gray20"	color for points in all GOFs
cex.data	1.5	cex for points in all GOFs
pch.data	19	pch for points in all GOFs
alpha.data	0.5	alpha for points in all GOFs
col.smooth	"#3366FF"	color for smoothers
lty.smooth	1	lty for smoothers
lwd.smooth	1	lwd for smoothers
se.smooth	TRUE	show confidence interval for loess smoother (TRUE) or not (FALSE)
span.smooth	0.6666667	span parameter for loess smothers
degree.smooth	1	degree parameter for loess smothers
family.smooth	"symmetric"	family parameter for loess smothers
col.refline	"red"	color for reference lines in all GOFs
lty.refline	2	lty for reference lines in all GOFs
lwd.refline	1	lwd for reference lines in all GOFs
fill.hist	"gray30"	color for filling of histograms
col.hist	"transparent"	color for border of histograms
alpha.hist	0.5	alpha for fill area in histograms
fill.box	"steelblue"	color for filling of box-plots
col.box	"black"	color for border and whisker of box-plots
alpha.box	0.6	alpha for fill area in box-plots
axis.labels	TRUE	use dictionary for labels (TRUE) or just use column names for axis labels (FALSE)
add.caption	TRUE	add caption to GOF plots (true) or not (FALSE)
size.caption	8	size for caption text
col.caption	1	color for caption text

caption.path.depth	99	number of folders levels to include in caption
corr.fontface	4	font face for correlation in eta.pairs.GOF
eta.labels	NULL	vector of labels for ETA's
script.name	""	script name for caption

### Examples

```
### Example setting ETA labels
set.ETA.labels(c("Ka", "CL", "V"))
### Example setting script name
set.script.name("MyScript.R")
```

---

get.label	<i>Labels for known NONMEM variables</i>
-----------	--

---

### Description

get.label match known NONMEM variables to the GOF-dictionary and returns the matched label. Unless trans is NULL, the label is modified to 'f(matched label)'

### Usage

```
get.label(x, trans = NULL)
```

### Arguments

x	column to get label for
trans	transformation

### See Also

[get.GOF.dictionary()], [set.GOF.dictionary()], and [default.GOF.dictionary()].

---

get.model.path	<i>Get and set model.path</i>
----------------	-------------------------------

---

### Description

The model.path in the nonmem2R set of global variables is used by functions loading NONMEM or PSN generated output file. The model.path can be retrieved by get.model.path and can be set using set-model.path. When loading nonmem2R the model-path is set to "", e.g. the current working directory. The model.path can be set absolute or relative, see examples.

## Usage

```
get.model.path()

set.model.path(model.path)
```

## Arguments

model.path      character string for path to model files

## Examples

```
## Not run:
## Example for setting absolute path
set.model.path("c:/NONMEM")
## Example for setting relative path,
set.model.path("../NONMEM")

## End(Not run)
```

---

grid.sim

*Combine fix grid and simulated grid based on multivariate normal distribution*

---

## Description

grid.sim produce comparable output to mvnorm, however for methods 2 to 4 grid.sim make use of center and scaling, and or use a fix grid of values for one column of the output. The intended use of grid.sim is for computing confidence intervals (CI) for model predictions. Method 1 is the same as rmvnorm, however can handle cases of zero variances in sigma. Method 2 is method 1 followed by centering and scaling of the simulated matrix. Thus output when using method 2 always have sample mean and covariance equal to input parameters means and sigma. Method 3 is the same as method 1 however with one column having a fixed range of values rather than a simulated range of values. Method 4 is method 3 followed by centering and scaling of the simulated matrix. Thus output when using method 4 always have sample mean and covariance equal to input parameters means and sigma.

As the number of simulations (n) goes to infinity all methods in grid.sim are identical to mvnorm in that the sample covariance (mean) of the output will converge to the input covariance matrix sigma (vector means).

The advantage with methods 2 to 4 (over method 1 and rmvnorm) is that they provides more stable results, hence number of simulations can be reduced and still have equally stable results when used to represent parameter uncertainty and or population variability in model predictions.

## Usage

```
grid.sim(n, means = NULL, sigma, grid.param = NULL, method = 4)
```

**Arguments**

n	Number of simulations
means	vector of mean values
sigma	covariance matrix
grid.param	the index of the parameter for which a fix grid (from qnorm) is used instead of a simulated grid (rnorm). If grid.param=NULL (default) the fix grid will be used for the parameter with largest variance.
method	simulation method, default=4

**Value**

row-matrix of parameters

**Examples**

```
sigma<-matrix(c(1,0.5,0.5,2),ncol=2)
sim1<-grid.sim(1000,sigma=sigma)
pairs(sim1)
cov(sim1)
```

---

histGOF

*Histogram GOF*


---

**Description**

Histogram GOF showing histogram, smooth density, and normal density as reference line and caption added as returned by get.caption

**Usage**

```
histGOF(
  data,
  x,
  title = "",
  color = "",
  fx = NULL,
  bins = NA,
  refline = TRUE,
  control = GOF.control()
)
```

**Arguments**

data	data.frame to plot
x	character string with name of column for x
title	title
color	data columns to set different colors in plot, interpreted as factor
fx	function for transformation of x before plotting
bins	number of histogram bins, if NA, bins are set using Sturges' formula
refline	add reference line (default =TRUE) or not (FALSE)
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

**Examples**

```
dd<-data.frame(CWRES=rnorm(100),gr=rep(LETTERS[1:5],20))
histGOF(dd,"CWRES",color="gr")
```

---

merge2GOF

*Merging 2, 4, or 6 GOF's into one graph*


---

**Description**

Merging multiple GOFs (ggplot objects) into one graph with caption added as returned by get.caption. Any legend present in first GOF is added to the combined graph.

**Usage**

```
merge2GOF(p1, p2, byrow = TRUE)
```

```
merge4GOF(p1, p2, p3, p4)
```

```
merge6GOF(p1, p2, p3, p4, p5, p6)
```

**Arguments**

p1	first ggplot object
p2	second ggplot object
byrow	combine plots side by side (TRUE) or one above the other (FALSE)
p3	third ggplot object
p4	4th ggplot object
p5	5th ggplot object
p6	6th ggplot object

---

modload	<i>Show model file</i>
---------	------------------------

---

**Description**

Load and dump model file in the consol.

**Usage**

```
modload(model, use.model.path = TRUE)
```

**Arguments**

model	name of the mod file with or without the .mod extension. model may include full or relative path to the mod file. See examples.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path

**Value**

data.frame

**Examples**

```
##### Load the .ext file "run001.mod"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.mod", package = "nonmem2R")
# 2) Load the file using the extload function
modload(file1)
```

---

position_scale	<i>Position for scaling y</i>
----------------	-------------------------------

---

**Description**

Position for scaling y

**Usage**

```
position_scale(y = 1)
```

**Arguments**

y	scale value
---	-------------

**Details**

Position for scaling y

---

 qqnormGOF

*QQ-norm GOF*


---

**Description**

QQ-norm GOF qqnorm plots with reference line and caption added as returned by get.caption

**Usage**

```
qqnormGOF(
  data,
  sample,
  title = "",
  color = "",
  fx = NULL,
  refline = TRUE,
  control = GOF.control()
)
```

**Arguments**

data	data.frame to plot
sample	character string with name of column for x
title	title
color	data columns to set different colors in plot, interpreted as factor
fx	function for transformation of x before plotting
refline	add reference line (default =TRUE) or not (FALSE)
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.

**Examples**

```
dd<-data.frame(CWRES=rnorm(100))
qqnormGOF(dd, "CWRES")
```

---

stat_corr	<i>Adding correlation to x-y plots</i>
-----------	--

---

### Description

Correlation to x-y plots. This stat is intended for providing correlations e.g. above the diagonal of a pairs plot

### Usage

```
stat_corr(  
  mapping = NULL,  
  data = NULL,  
  geom = "text",  
  position = "identity",  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  ...  
)
```

### Arguments

mapping	Set of aesthetic mappings created by aes or aes_.
data	The data to be displayed in this layer.
geom	Use to override the default geom
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to layer.

### Details

Provide normal QQplot

---

stat_denx	<i>Density for x in x-y plot</i>
-----------	----------------------------------

---

### Description

Density for x in x-y plot. This stat is intended for providing the a density on the diagonal of a pairs plot

### Usage

```
stat_denx(
  mapping = NULL,
  data = NULL,
  geom = "polygon",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

### Arguments

mapping	Set of aesthetic mappings created by aes or aes_.
data	The data to be displayed in this layer.
geom	Use to override the default geom
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to layer.

### Details

Provide normal QQplot

---

stat_dnorm	<i>Normal density</i>
------------	-----------------------

---

### Description

Normal density with ggplot2

### Usage

```
stat_dnorm(  
  mapping = NULL,  
  data = NULL,  
  geom = "line",  
  position = "identity",  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE,  
  ...  
)
```

### Arguments

mapping	Set of aesthetic mappings created by aes or aes_.
data	The data to be displayed in this layer.
geom	Use to override the default geom
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to layer.

### Details

Provide normal QQplot

---

stat_hmean	<i>Horizontal reference at mean</i>
------------	-------------------------------------

---

### Description

Horizontal reference at mean with ggplot2

### Usage

```
stat_hmean(
  mapping = NULL,
  data = NULL,
  geom = "hline",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

### Arguments

mapping	Set of aesthetic mappings created by aes or aes_.
data	The data to be displayed in this layer.
geom	Use to override the default geom
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to layer.

### Details

Provide normal QQplot

---

stat_hmedian	<i>Horizontal reference at median</i>
--------------	---------------------------------------

---

### Description

Horizontal reference at median with ggplot2

### Usage

```
stat_hmedian(
  mapping = NULL,
  data = NULL,
  geom = "hline",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

### Arguments

mapping	Set of aesthetic mappings created by aes or aes_.
data	The data to be displayed in this layer.
geom	Use to override the default geom
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them.
...	other arguments passed on to layer.

### Details

Provide normal QQplot

### Aesthetics

geom\_smooth understands the following aesthetics (required aesthetics are in bold):  
**x**, alpha, colour, fill, group, shape, size, stroke

---

sumoR	<i>Compile summary information for NONMEM model based on the lst file, ext file, and the cov file.</i>
-------	--

---

## Description

Compile summary information similar to that of the sumo PSN function, based on the NONMEM output files lst, ext, and if covariance setp was run, the cov file.

## Usage

```
sumoR(
  model,
  use.model.path = TRUE,
  tableType = 2,
  format.estimate = "% -#6.4g",
  format.rse = "%#6.3g"
)
```

## Arguments

model	name of the lst file with or without the .lst extension. model may include full or relative path to the lst file.
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
tableType	Table type for THETA's, OMEGA's and SIGMA's tableType=0: Present OMEGA and SIGMA as variance and covariances and display SE for THETA, OMEGA, SIGMA tableType=1: Present OMEGA and SIGMA as variance and covariances and display RSE for THETA, OMEGA, SIGMA tableType=2: Present OMEGA and SIGMA as standard-deviation and correlations and display RSE for THETA, OMEGA, SIGMA tableType=3: Present OMEGA and SIGMA as standard-deviation and correlations and display SE for THETA, OMEGA, SIGMA
format.estimate	format for estimated value, passed to sprintf
format.rse	format for RSE or SE, passed to sprintf

## Value

named list of class sumoR

**Examples**

```
##### Compile summary information from the .lst file "run001.lst"
# 1) Get path to the example file included in nonmem2R package
file1 <- system.file("extdata", "run001.lst", package = "nonmem2R")
# 2) Compile summary information from "run001.lst"
sumoR(file1)
```

test.grid.sim

*Function for testing grid.sim and compare with rmvnorm***Description**

Test grid.sim

**Usage**

```
test.grid.sim(n = 1000, k = 1:4)
```

**Arguments**

n	number of simulations
k	subset of parameters from a 4X4 sigma to use

**Value**

graphics

**Examples**

```
## Not run:
require(lattice)
test.grid.sim(n=1000)

## End(Not run)
```

vpcfig

*Visual Predictive Check (VPC) based on Perl-speaks-NONMEM (PsN) generated VPC files (lattice version).***Description**

This function creates VPC using output files from the vpc command in Pearl Speaks NONMEM (PsN). Graphs are generated using the lattice package with many arguments for plot settings are passed directly to the xyplot function and have the same flexibility as when used in xyplot.

**Usage**

```

vpcfig(
  vpcdir = NULL,
  vpctab = NULL,
  vpcresult = NULL,
  use.model.path = TRUE,
  strata.names = NULL,
  strata.subset = NULL,
  percentile = 10,
  fy = function(y) { y },
  fx = function(x) { x },
  xlab = NULL,
  ylab = NULL,
  col.data = 8,
  cex.data = 0.3,
  pch.data = 3,
  col.line = c("blue", "red", "blue"),
  lwd.line = c(1, 2, 1),
  lty.line = c(1, 1, 1),
  col.segm = c("lightblue", "pink", "lightblue"),
  alpha.segm = c(0.5, 0.5, 0.5),
  type = 3,
  ...
)

```

**Arguments**

vpcdir	Path of directory of the VPC files
vpctab	Path to the vpctab-file
vpcresult	Path to the vpcresult file
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
strata.names	Character vector for strata names. Must have length equal to number of strata's in vpc files, otherwise ignored.
strata.subset	Vector specifying subset of strata to use. Either a vector of index, e.g. strata.subset=c(1,3), or a character vector naming which strata to use. See details.
percentile	percentile to use, default=10 will display 10 Percentile argument must match columns included in the vpcresult file.
fy	transformation function for y-axis, default to identity function (f(y)=y).
fx	transformation function for x-axis, default to identity function (f(x)=x)
xlab	label for x-axis, default is IDV as found in vpcresultfile, passed to xyplot.
ylab	label for y-axis, default is DV as found in vpcresultfile, passed to xyplot.
col.data	color for data points, passed to xyplot, default=8
cex.data	cex for data points, passed to xyplot, default=0.3

<code>pch.data</code>	plot symbol ( <code>pch</code> ) for data points, passed to <code>xyplot</code> , default=3
<code>col.line</code>	line color for low quartile, median, and high quartile of data, passed to <code>xyplot</code> , default=c("blue","red","blue")
<code>lwd.line</code>	line width ( <code>lwd</code> ) for low quartile, median, and high quartile of data, passed to <code>xyplot</code> , default=c(1,2,1)
<code>lty.line</code>	line type ( <code>lty</code> ) for low quartile, median, and high quartile of data, passed to <code>xyplot</code> , default=c(1,1,1)
<code>col.segm</code>	Color of CI regions (low, median and high) for model, passed to <code>xyplot</code> , default=c("lightblue","pink","lightblue")
<code>alpha.segm</code>	alpha of CI regions (low, median and high) for model, passed to <code>xyplot</code> , default=c(0.5,0.5,0.5)
<code>type</code>	type of VPC plot, 1=display model regions only, 2=as 1 + lines (low, median and high) of data, 3= as 2 + points for data. <code>type=0</code> can also be used and then no graph is produced but instead a list with 2 dataframes is returned. one for the <code>vpresult</code> and one for the observed data in the <code>vpctab</code> file.
<code>...</code>	Further arguments, passed to <code>xyplot</code> , e.g <code>xlim</code> and <code>yylim</code> axis limits, <code>main</code> for title of plot, <code>abline</code> for adding reference lines, or scales for formatting axes, see help files for <code>xyplot</code> . Note: <code>col</code> , <code>cex</code> , <code>pch</code> , <code>lty</code> , <code>lwd</code> , and <code>alpha</code> should not be used. Instead use the corresponding arguments defined above.

## Details

The lattice package is used for creating the VPC and `vpctfig` have functionality for e.g. modify names of strata, change layout of panels, use log-scale.

The data used is either specified by the directory of the PsN generated files, or by specifying the file names of both the `vpctab`-file and the `vpresult` file. See examples 1 and 2 below.

Names of strata can be changed with the `strata.names` argument, if `strata.names` is NULL the names as specified in the `vpresult` file are used.

`strata.subset` can be used to select a subset of strata, and or to change the order of stratas. See example 3. NOTE: `strata.subset` is matched with `strata.names` unless `strata.names=NULL`. If `strata.names=NULL` then `strata.subset` is matched to names as specified in `vpresult` file. Default is to use all strata's (`strata.subset=NULL`).

The `fy` and `fx` arguments can be used to alter the scale of data plotted. For example, using `fy=function(y)(log(y+1))` is a convenient way to get log-scale for y-axis but with an off-set to show any values==0. Proper y tick marks can then be set by using the `xyplot` argument `scales`.

## Value

lattice object of VPC plot

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "vpctab004.dat", package = "nonmem2R")
file2 <- system.file("extdata", "vpc_results.csv", package = "nonmem2R")
```

```

# Ex 1, produce VPC with default setting, here specifying both vpctab and vpcresult
vpcfig(vpctab=file1,vpcresult=file2)

## Not run:
# Ex 2, produce VPC with default setting, here specifying only directory of vpc files
path1<-gsub("vpctab004.dat","",file1)
vpcfig(vpcdir=path1)

## End(Not run)

# Ex 3, produce VPC with i) modifies strata names, ii) strata in reverse order, and
#      iii) labels
strata.names<-c("Group A","Group B")
xlab<-"Time after dose (hrs)"
ylab<-"Plasma Conc(mmol/L)"
vpcfig(vpctab=file1,vpcresult=file2,strata.names=strata.names,strata.subset=2:1,
       xlab=xlab,ylab=ylab)

```

---

vpcfig2

*Visual Predictive Check (VPC) based on Perl-speaks-NONMEM (PsN) generated VPC files (ggplot2-version).*

---

## Description

This function creates VPC using output files from the vpc command in Pearl Speaks NONMEM (PsN). Graphs are generated using the ggplot2 package and the return object is an ggplot class and can be further modified, see details below and documentation for ggplot2 for further details.

## Usage

```

vpcfig2(
  vpcdir = NULL,
  vpctab = NULL,
  vpcresult = NULL,
  use.model.path = TRUE,
  strata.names = NULL,
  strata.subset = NULL,
  median.only = FALSE,
  bin.idv = c("median", "midpoint"),
  percentile = 10,
  fy = function(y) { y },
  fx = function(x) { x },
  xlab = NULL,
  ylab = NULL,
  col.data = "gray20",
  cex.data = 1,
  pch.data = 19,

```

```

alpha.data = 0.5,
col.line = "grey20",
lwd.line = 1,
lty.line = c(2, 1, 2),
col.segm = c("steelblue", "gray50", "steelblue"),
alpha.segm = 0.6,
type = 3,
panel.height.censored = 0.25,
censoring.labels = c("ALQ", "BLQ"),
ignore.censoring = FALSE,
control = GOF.control()
)

```

### Arguments

vpcdir	Path of directory of the VPC files
vpctab	Path to the vpctab-file
vpcresult	Path to the vpcresult file
use.model.path	Load file from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
strata.names	Character vector for strata names. Must have length equal to number of strata's in vpc files, otherwise ignored.
strata.subset	Vector specifying subset of strata to use. Either a vector of index, e.g. strata.subset=c(1,3), or a character vector naming which strata to use. See details.
median.only	Logical to plot only median and no quantiles for model and data (FALSE=default)
bin.idv	Method for computed idv value for plotting bin summary values. Set to "median" to use median of independent variable in each bin on x-axis (default), or set to "midpoint" to use midpoint of bins on x-axis.
percentile	percentile to use, default=10 will display 10 Percentile argument must match columns included in the vpcresult file.
fy	transformation function for y-axis, default to identity function (f(y)=y).
fx	transformation function for x-axis, default to identity function (f(x)=x)
xlab	label for x-axis, default is IDV as found in vpcresultfile, passed to labs.
ylab	label for y-axis, default is DV as found in vpcresultfile, passed to labs.
col.data	color for data points, passed to geom_point, default="gray20"
cex.data	cex for data points, passed to geom_point, default=1
pch.data	plot symbol (pch) for data points, passed to geom_point, default=19
alpha.data	alpha for plotting of data points, passed to geom_point, default=0.5
col.line	line color for low quartile, median, and high quartile of data, passed to geom_line, default="gray20"
lwd.line	line width (lwd) for low quartile, median, and high quartile of data, passed to geom_line, default=1

<code>lty.line</code>	line type (lty) for low quartile, median, and high quartile of data, passed to <code>geom_line</code> , should be vector of length 3, default= <code>c(2,1,2)</code>
<code>col.segm</code>	Color of CI regions (low, median and high) for model, passed to <code>geom_polygon</code> , should be vector of length 3, default= <code>c("steelblue", "gray50", "steelblue")</code>
<code>alpha.segm</code>	alpha of CI regions (low, median and high) for model, passed to <code>geom_polygon</code> , default=0.6
<code>type</code>	type of VPC plot: 1=display model regions only, 2=as 1 + lines (low, median and high) of data, 3= as 2 + points for data.  type=0 can also be used and then no graph is produced but instead a list with 2 dataframes is returned. one for the <code>vpcresult</code> and one for the observed data in the <code>vpctab</code> file.
<code>panel.height.censored</code>	Height for BLQ ALQ panels. Height is relative to height of Un-censored panel, default=0.25
<code>censoring.labels</code>	Character vector used as labels for panels when VPC have either BLQ or ALQ data Default set to <code>c("ALQ","BLQ")</code> . NOTE: must be character vector of length 2
<code>ignore.censoring</code>	Ignore any censoring results in VPC result file (BLQ and ALQ), default=FALSE
<code>control</code>	an optional list of control settings. See <code>GOF.control</code> for the names of the settable control values and their effect.

## Details

The `ggplot2` package is used for creating the VPC and `vpcfig2` have functionality for e.g. modify names of strata, order of and subset of strata's, and transform the x- and or y-axis before plotting.

The data used is either specified by the directory of the PsN generated files, or by specifying the file names of both the `vpctab`-file and the `vpcresult` file. See examples 1 and 2 below.

Names of strata can be changed with the `strata.names` argument, if `strata.names` is NULL the names as specified in the `vpcresult` file are used.

`strata.subset` can be used to select a subset of strata, and or to change the order of stratas. See example 3.

**Strata.subset** is matched with `strata.names` unless `strata.names=NULL`. If `strata.names=NULL` then `strata.subset` is matched to names as specified in `vpcresult` file. Default is to use all strata's (`strata.subset=NULL`).

The `fy` and `fx` arguments can be used to alter the scale of data plotted. For example, using `fy=function(y){log(y+1)}` is a convenient way to get log-scale for y-axis but with an off-set to show any values==0. Proper y tick marks can then be set using the `scale_y_continuous` `ggplot2` function.

Since the returned object is a `ggplot`-class object it can be further modified to e.g. to log-scale for y-axis;

```
vpcfig2(...)+scale_y_log10(),
```

adding/changing labels & titles;

```
vpcfig2(...)+labs(y="Modified y-label", title="New title").
```

**Axis-limits** are preferably set using the ggplot2 function `coord_cartesian(...)`. This way data points outside the axis-limit are only hidden when plotting.

**Stratified** VPC's are created with `facet_wrap(~strata)` as default but can be modified to use `facet_grid` for setting the panel grid.

However faceting must be done with `~strata`.

See examples below and documentation for ggplot2 for further details.

## Value

ggplot object of VPC plot

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "vpctab004.dat", package = "nonmem2R")
file2 <- system.file("extdata", "vpc_results.csv", package = "nonmem2R")

# Ex 1, produce VPC with default setting, here specifying both vpctab and vpcresult
vpcfig2(vpctab=file1,vpcresult=file2)

## Not run:
# Ex 2, produce VPC with default setting, here specifying only directory of vpc files
path1<-gsub("vpctab004.dat","",file1)
vpcfig2(vpmdir=path1)

## End(Not run)

# Ex 3, produce VPC with i) modifies strata names, ii) strata in reverse order, and
#      iii) labels
strata.names<-c("Group A","Group B")
xlab<-"Time after dose (hrs)"
ylab<-"Plasma Conc(mmol/L)"
vpcfig2(vpctab=file1,vpcresult=file2,strata.names=strata.names,strata.subset=2:1,
        xlab=xlab,ylab=ylab)

# Example using the fy argument to transform y-axis setting y-ticks using scale_y_continuous(...)
tmp<-c(1,2,3,4,6)
yticks<-c(0.1,tmp,tmp*10,tmp*100,tmp*1000)
vpcfig2(vpctab=file1,vpcresult=file2,fy=function(y){log(y+1)})+
  scale_y_continuous(breaks=log(yticks+1),labels=yticks,minor_breaks=NULL)

## Not run:
# Example changing to slog-scale using the ggplot2 function scale_y_log10
vpcfig2(vpctab=file1,vpcresult=file2)+scale_y_log10()

## End(Not run)
```

```
# Example changing y-axis label and adding figure title using ggplot2 function labs(...).
vpcfig2(vpctab=file1,vpcresult=file2)+labs(y="Modified y-label", title="New title")
```

---

vpcfig3	<i>Visual Predictive Check (VPC) based on raw Perl-speaks-NONMEM (PsN) generated VPC files.</i>
---------	---

---

### Description

This function compile VPC simulated data from the vpc command in Pearl Speaks NONMEM (PsN). The compiled data is then plotted (unless type=0) and returned (ggplot-object) and can then be further modified.

### Usage

```
vpcfig3(
  sdtab,
  vpcdir,
  use.model.path = TRUE,
  dv = "DV",
  idvs = "TIME",
  numerical = rep(TRUE, length(idvs)),
  strata = NULL,
  method = c("loess", "spline"),
  pred.corr = c("none", "pred-corr-pro p", "pred-corr-add"),
  maxNsim = NULL,
  knots = 5,
  minobs = 8,
  col.data = "gray20",
  cex.data = 1,
  pch.data = 19,
  alpha.data = 0.5,
  col.line = "grey20",
  lwd.line = 1,
  lty.line = 1,
  col.segm = "steelblue",
  alpha.segm = 0.6,
  type = 3,
  control = GOF.control(),
  ...
)
```

### Arguments

sdtab	file path to sdtab output from model, ALTERNATIVELY a data.frame of the loaded sdtab file. See details.
-------	---

vpcdir	Path of directory of the VPC files, ALTERNATIVELY a data.frame of the loaded matrix file. See details.
use.model.path	Load files from a global defined model library (TRUE=default). If so will look for a global character vector named model.path
dv	Dependent variable (y-axis)
idvs	Independent variable (x-axis)
numerical	Logical indicator if idvs is numerical variable(TRUE) or categorical (FALSE), default=TRUE NOTE: must be of same length as idvs
strata	Stratification column of sdtab, NOTE must be a column in the sdtab
method	either "loess" (default) or "spline" for continuous IDV's
pred.corr	Set to "pred-corr-prop" or "pred-corr-add" to perform prediction corrected VPC or set to "none" (default) otherwise. With "pred-corr-prop" correction is recommended for strictly positive data, use "pred-corr-add" if some data points are negative,
maxNsim	maximum number of simulations used to use for computing VPC. Default=NULL, meaning all simulations in the input datafile/data.frame will be used.
knots	number of knots (bins) for method spline. See details.
minobs	minimum number of observations per number of knots(bins) for method spline. See details.
col.data	color for data points, passed to geom_point, default="gray20"
cex.data	cex for data points, passed to geom_point, default=1
pch.data	plot symbol (pch) for data points, passed to geom_point, default=19
alpha.data	alpha for plotting of data points, passed to geom_point, default=0.5
col.line	line color for mean of data, passed to geom_line, default="gray20"
lwd.line	line width (lwd) for mean of data, passed to geom_line, default=1
lty.line	line type (lty) for mean of data, passed to geom_line, default=1
col.segm	Color of CI region for model, passed to geom_ribbon, default="steelblue"
alpha.segm	alpha of CI region for model, passed to geom_polygon, default=0.6
type	type of VPC plot: 1=display model regions only, 2=as 1 + mean of data, 3= as 2 + points for data.  type=0 can also be used and then no graph is produced but instead a list with 2 dataframes is returned. one for the vpcresult and one for the observed data in the vpctab file.
control	an optional list of control settings. See GOF.control for the names of the settable control values and their effect.
...	Additional named arguments (e.g. span and degree) are passed to loess.smooth.

## Details

The sdtab and vpcdir can either be file name of the sdtab file and the folder name of the psn generated VPC, or can be data.frames of sdtab and simulation data loaded outside vpcfig3, see example below for the latter case.

Loading the sdtab and matrix file outside vpcfig2 is convenient when dosing multiple VPC plots or for VPC using new strata ( or idv) variables not included in the sdtab file, see example.

If vpcdir is given as a folder names, then a matrix.csv file is loaded from the /m1 subfolder of vpcdir.

The raw simulated results from the matrix file is loaded to compute mean of data with confidence intervals For numerical idv variables loess-smooth or linear spline is used on observed actual data and on simulated data. For categorical idv mean is computed for each unique value if idv

For method=spline, the actual number of knots(bins) is set as  $\min(\text{knots}, N/\text{minobs})$  where N is the number of observations.

## Value

A ggplot object if type>0, or if type==0 a list of 3 data.frames, D1=individual data, D3=mean data, D5=mean model with confidence intervals

## Examples

```
# Get path to the example files included in nonmem2R package
file1 <- system.file("extdata", "sdtab", package = "nonmem2R")
file2 <- system.file("extdata", "DV_matrix.csv", package = "nonmem2R")

#load sdtab and matrix file
sdtab<-read.table(file=file1,skip=1,header=TRUE)
vpc<-read.table(file=file2,header=FALSE,sep=",")

# VPC stratified by gender
vpcfig3(sdtab,vpc,dv="DV",idvs="TAD",strata="SEX")

# create new strata variable and do VPC
sdtab$age.group<-cut(sdtab$AGE,c(22,30,50))
vpcfig3(sdtab,vpc,dv="DV",idvs="TAD",strata="age.group")
```

# Index

add.caption, 2

basic.eta.GOF, 3

basic.GOF4, 4

basic.GOF6 (basic.GOF4), 4

covload, 6

default.GOF.dictionary  
(get.GOF.dictionary), 15

default.GOF.params  
(get.GOF.dictionary), 15

do.cat.GOF, 7

do.individual.GOF, 8

do.multi.GOF, 9

do.one.GOF, 10

eta.cat.GOF (eta.cov.GOF), 11

eta.cov.GOF, 11

eta.hist.GOF (basic.eta.GOF), 3

eta.pairs.GOF (eta.cov.GOF), 11

eta.qqnorm.GOF (basic.eta.GOF), 3

extload, 13

extToTable, 14

get.GOF.dictionary, 15

get.GOF.params (get.GOF.dictionary), 15

get.label, 17

get.model.path, 17

GOF.control (get.GOF.dictionary), 15

grid.sim, 18

histGOF, 19

merge2GOF, 20

merge4GOF (merge2GOF), 20

merge6GOF (merge2GOF), 20

modload, 21

position\_scale, 21

qqnormGOF, 22

set.ETA.labels (get.GOF.dictionary), 15

set.GOF.dictionary  
(get.GOF.dictionary), 15

set.GOF.params (get.GOF.dictionary), 15

set.model.path (get.model.path), 17

set.script.name (get.GOF.dictionary), 15

stat\_corr, 23

stat\_denx, 24

stat\_dnorm, 25

stat\_hmean, 26

stat\_hmedian, 27

sumoR, 28

test.grid.sim, 29

vpcfig, 29

vpcfig2, 32

vpcfig3, 36