Package 'nlmixr2rpt'

September 26, 2026

Title Templated Word and PowerPoint Reporting of 'nlmixr2' Fitting Results

Version 0.2.3

Maintainer John Harrold <john.m.harrold@gmail.com>

Description This allows you to generate reporting workflows around 'nlmixr2' analyses with outputs in Word and PowerPoint. You can specify figures, tables and report structure in a user-definable 'YAML' file. Also you can use the internal functions to access the figures and tables to allow their including in other outputs (e.g. R Markdown).

URL https://nlmixr2.github.io/nlmixr2rpt/

BugReports https://github.com/nlmixr2/nlmixr2rpt/issues

License GPL (>= 3)

Encoding UTF-8

Imports cli, dplyr, equatags, flextable, ggforce, ggpubr, ggplot2, grDevices, knitr, stringr, nlmixr2est, nlmixr2extra (>= 2.0.7), onbrand, rxode2, utils, xpose, xpose.nlmixr2, yaml

Suggests rmarkdown, GGally, ggPMX, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author John Harrold [aut, cre] (ORCID: <https://orcid.org/0000-0003-2052-4373>)

Repository CRAN

Date/Publication 2026-09-26 03:50:02 UTC

Contents

build_figures 2
build_tables 4
eval_str 5

fetch_fdim 6
fetch_fit_example 7
fetch_option 7
gen_eqn_table 8
gen_pest_table 9
mk_error_fig 10
mk_error_tab 11
process_ph 11
report_fit 12
safe_text 13
write_figure 14
yaml_read_fit 15

Index **17**

`build_figures` *Generates Figures for an* `nlmixr2` *Report*

Description

Creates figures specified in a rptyaml file

Usage

```
build_figures(
  obnd = NULL,
  fit = NULL,
  rptdetails = NULL,
  cat_covars = NULL,
  cont_covars = NULL,
  verbose = TRUE
)
```

Arguments

`obnd`	`onbrand` report object to have report elements appended to
`fit`	`nlmixr2` fit object to be reported
`rptdetails`	Object created when reading in rptyaml file
`cat_covars`	character vector of categorical covariates to overwrite defaults in yaml file
`cont_covars`	character vector of continuous covariates to overwrite defaults in yaml file
`verbose`	Boolean variable when set to TRUE (default) messages will be displayed on the terminal

Value

List containing the figures with the following structure:

- "rptfigs" - List of figures with names corresponding to the figure ids in the yaml file. Each figure ID contains the following elements:
  - "figure" - list of figure file names for the current fid
  - "orientation" - Figure orientation ("portrait" or "landscape")
  - "isgood" - Boolean variable indicating success or failure
  - "skip" - Boolean variable indicating whether the figure should be skipped during reporting
  - "fmsgs" - Vector of messages
  - "cmd" - Original plot generation command
  - "cmd_proc" - Plot generation command after processing for placeholders
  - "height" - Figure height
  - "width" - Figure width
  - "caption" - Caption for Word
  - "caption_proc" - Caption for Word after processing for placeholders
  - "title" - Slide title for PowerPoint
  - "title_proc" - Slide title for PowerPoint after processing for placeholders
- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages

Examples

```
# We need an onbrand object to use below
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.docx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# We also need an nlmixr fit object
fit = fetch_fit_example()

# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails

# Now we will build the figures
bfres = build_figures(obnd       = obnd,
                      fit        = fit,
                      rptdetails = rptdetails)
```

build_tables *Generates Tables for an* nlmixr2 *Report*

Description

Creates tables specified in a rptyaml file

Usage

```
build_tables(
  obnd = NULL,
  fit = NULL,
  rptdetails = NULL,
  cat_covars = NULL,
  cont_covars = NULL,
  verbose = TRUE
)
```

Arguments

obnd	onbrand report object to have report elements appended to
fit	nlmixr2 fit object to be reported
rptdetails	object creating when reading in rptyaml file
cat_covars	character vector of categorical covariates to overwrite defaults in yaml file
cont_covars	character vector of continuous covariates to overwrite defaults in yaml file
verbose	Boolean variable when set to TRUE (default) messages will be displayed on the terminal

Value

List containing the tables with the following structure:

- "rpttabs" - List of tables with names corresponding to the table ids in the yaml file. It contains the elements from the yamle file and the following elements:
  - "table" - Result of build (t_res object)
  - "orientation" - Table orientation ("portrait" or "landscape")
  - "isgood" - Boolean variable indicating success or failure
  - "skip" - Boolean variable indicating whether the table should be skipped during reporting
  - "tmsgs" - Vector of messages
  - "cmd" - Original plot generation command
  - "cmd_proc" - Plot generation command after processing for placeholders
  - "height" - Table height
  - "width" - Table width

  - "caption" - Caption for Word
  - "caption_proc" - Caption for Word after processing for placeholders
  - "title" - Slide title for PowerPoint
  - "title_proc" - Slide title for PowerPoint after processing for placeholders
- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages

Examples

```
# We need an onbrand object to use below

library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.docx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# We also need an nlmixr fit object
fit = fetch_fit_example()

# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails

# Now we will build the tables
btres = build_tables(obnd        = obnd,
                     fit        = fit,
                     rptdetails = rptdetails)
```

---

eval_str *Evaluate R Code in String*

Description

Attempts to evaluate a string as a chunk of R code.

Usage

```
eval_str(estr = "", fit = NULL)
```

Arguments

estr	Object creating when reading in rptyaml file
fit	nlmixr2 fit object to be reported

Value

String containing the evaled as a character or the original string

Examples

```
res = eval_str(estr="ls()")
```

`fetch_fdim` *Gets Figure Dimensions*

Description

For a given figure id and report type this will pull out the dimensions of the figure.

Usage

```
fetch_fdim(obnd = NULL, fid = NULL, fdim = "width", rptdetails = NULL)
```

Arguments

obnd	onbrand report object to have report elements appended to
fid	Figure ID used in the figures section of the yaml file
fdim	Dimension to fetch either "width" or "height"
rptdetails	Object creating when reading in rptyaml file

Value

ggplot object

Examples

```
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.pptx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# This will create an example fit object to use in the examples below
fit = fetch_fit_example()

# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails

fetch_fdim(obnd=obnd, fid="bad_figure", fdim="width", rptdetails=rptdetails)

fetch_fdim(obnd=obnd, fid="bad_figure", fdim="height", rptdetails=rptdetails)
```

fetch_fit_example *Fetch Fit Example*

Description

Creates an nlmixr2 fit example using posthoc estimation method for testing purposes. displayed on the terminal following:

Usage

```
fetch_fit_example(use_cache = TRUE)
```

Arguments

use_cache	Boolean variable used to cache the fit process for the current R session.

Value

Example nlmixr2 fit object

Examples

```
fit = fetch_fit_example()
```

fetch_option *Fetch Analysis Options*

Description

Fetches analysis options from the report yaml applies it to strings.

Usage

```
fetch_option(rptdetails, option = NULL, fit = NULL, verbose = TRUE)
```

Arguments

rptdetails	Object creating when reading in rptyaml file
option	String containing the option to fetch (see below)
fit	nlmixr2 fit object to be reported
verbose	Boolean variable when set to TRUE (default) messages will be displayed on the terminal following:

Details

The option can be one of the following (default: NULL):

- "output_dir" - Directory to place figures that are generated (default: tempdir())
- "resolution" - Resolution of figure files (default: 300)

Value

List containing the following information about the output directory

- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages
- "value" - The value of the option or the default if not specified

Examples

```
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.pptx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# This will create an example fit object to use in the examples below
fit = fetch_fit_example()

# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails


fetch_option(rptdetails, option="output_dir", fit=fit)
```

---

gen_eqn_table *Makes* nlmixr2 *Model Equation Table for Reporting*

Description

Generates a flextable with one row per model equation rendered as native Word/PowerPoint math. The equations are taken from the LaTeX produced by the knit_print() method for nlmixr2 fit objects (provided by nlmixr2extra), split into individual rows, and converted with flextable::as_equation().

Usage

```
gen_eqn_table(fit, obnd)
```

Arguments

fit	nlmixr2 fit object to be reported
obnd	onbrand report object used to determine the report type ("Word" or "PowerPoint")

Details

The math is rendered by `flextable::as_equation()`, which relies on the `equatags` package. Line spacing and padding are adjusted based on the report type in obnd: Word documents get extra line spacing so rows containing fractions do not crowd each other, while PowerPoint sizes the math itself and keeps the rows tight.

Value

`flextable` object containing the model equations, one equation per row, with no header and no borders

Examples

```
# We need an onbrand object to use below
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.docx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# This will create a fit object to use in the examples below
fit = fetch_fit_example()

ft = gen_eqn_table(fit = fit, obnd = obnd)
```

---

gen_pest_table *Makes* nlmixr2 *Parameter Estimate Table for Reporting*

Description

Generates a flextable containting the parameter estimates.

Usage

```
gen_pest_table(obnd = NULL, fit = NULL, rptdetails = NULL, verbose = TRUE)
```

Arguments

obnd	onbrand report object to have report elements appended to
fit	nlmixr2 fit object to be reported
rptdetails	object creating when reading in rptyaml file
verbose	Boolean variable when set to TRUE (default) messages will be displayed on the terminal

Value

List with the following elements

- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages
- "ft" - Parameter estimates as a flextable object
- "df" - Parameter estimates as a data.frame

Examples

```
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.pptx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# This will create an example fit object to use in the examples below
fit = fetch_fit_example()

#'# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails

gen_pest_table(obnd = obnd, fit = fit, rptdetails = rptdetails, verbose = TRUE)
```

---

mk_error_fig *Generates* ggplot *Object with Error Message*

Description

Takes a vector of messages and returns a ggplot object with the text in the figure. This can be used in automated figure generation to cascade an error message to the end user.

Usage

```
mk_error_fig(msgs)
```

Arguments

msgs	Vector of error messages

Value

ggplot object

Examples

```
mk_error_fig("This is an error")
```

mk_error_tab *Generates a* flextable *Object with Error Message*

Description

Takes a vector of messages and returns a `flextable` object with the text in the table. This can be used in automated table generation to cascade an error message to the end user.

Usage

```
mk_error_tab(msgs)
```

Arguments

`msgs`	Vector of error messages

Value

list with a single `flextable` object

Examples

```
error_tab = mk_error_tab("This is an error")
error_tab$ft[[1]]
```

process_ph *Substitutes Placehodlers in Strings*

Description

Takes placeholder information from the rptyaml file and applies it to strings.

Usage

```
process_ph(str, rptdetails)
```

Arguments

`str`	String to process
`rptdetails`	Object creating when reading in rptyaml file (default: `NULL`)

Value

processed string

Examples

```
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.docx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# We also need an nlmixr fit object
fit = fetch_fit_example()

# This reads in the report details as well
rptdetails = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)$rptdetails

str = "This is ===CMPD==="

process_ph(str, rptdetails)
```

---

report_fit *Report* nlmixr2 *Fit Results to PowerPoint and Word*

Description

Appends nlmixr2 fit results to an onbrand report object with the content and format of the report in the supplied yaml file

Usage

```
report_fit(
  obnd = NULL,
  fit = NULL,
  placeholders = NULL,
  cat_covars = NULL,
  cont_covars = NULL,
  parameters = NULL,
  rptyaml = system.file(package = "nlmixr2rpt", "templates", "report_fit.yaml"),
  verbose = FALSE
)
```

Arguments

obnd	onbrand report object to have report elements appended to.
fit	nlmixr2 fit object to be reported.
placeholders	Manual placeholders, see yaml_read_fit for more.
cat_covars	character vector of categorical covariates to overwrite defaults in yaml file.
cont_covars	character vector of continuous covariates to overwrite defaults in yaml file.

parameters	list with element names for each parameter to overwrite defaults in yaml file.
rptyaml	yaml file containing the report elements and structure.
verbose	Boolean variable when set to TRUE messages will be .

Value

onbrand object with the report elements added.

Examples

```
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.pptx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# This will create an example fit object to use in the examples below
fit = fetch_fit_example()

# Appening fit results
obnd_pptx = report_fit(
 fit     = fit,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 obnd    = obnd)

# Writing the report to a file
save_report(obnd, file.path(tempdir(), "report.pptx"))
```

---

safe_text *Escapes Curly Braces for cli functions*

Description

Converts curly braces to double curly braces in order to prevent interpertation with the cli package

Usage

```
safe_text(x)
```

Arguments

x	String with curly braces

Value

String with curly braces escaped

Examples

```
safe_text("This string has curly {braces}")
```

write_figure *Writes Figures to File*

Description

Takes a figure object, optionally stamps the image, and writes to a file dimensions of the figure.

Usage

```
write_figure(
  p_res = NULL,
  page = NULL,
  width = 3,
  height = 3,
  resolution = NULL,
  fig_file = NULL,
  fig_stamp = NULL,
  verbose = TRUE
)
```

Arguments

p_res	ggplot, ggforce paginated object, or ggarrange object.
page	Page number to write or NULL for a ggplot object.
width	Width in inches.
height	Height in inches.
resolution	resolution in dpi.
fig_file	File name to write the figure to.
fig_stamp	Character object containing the text to stamp on the figure with optional ===file=== placeholder.
verbose	Boolean variable when set to TRUE (default) messages will be. displayed on the terminal

Value

list with the following

- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages

Examples

```
library(ggplot2)
write_figure(
 p_res = ggplot(),
 page = NULL,
 width = 3,
 height = 3,
 resolution = 200,
 fig_file = file.path(tempdir(), "write_figure_example.png"),
 fig_stamp = "stamp",
 verbose = TRUE)
```

---

yaml_read_fit *Reads and Checks* report_fit.yaml *File*

Description

Reads in the report yaml file and looks it to make sure it has all the necessary fields for the given report.

Usage

```
yaml_read_fit(
  obnd = NULL,
  rptyaml = NULL,
  placeholders = NULL,
  parameters = NULL,
  fit = NULL
)
```

Arguments

obnd	onbrand report object to have report elements appended to.
rptyaml	yaml file containing the report elements and structure.
placeholders	list of placeholders to overwrite defaults in the yaml file.
parameters	list with element names for each parameter to overwrite at runtime. with a named list for example RUN may be "RUNN" in the yaml file. To overwrite this just provide list(RUN="RUN_1") (default: NULL)
fit	nlmixr2 fit object to be reported.

Value

List containing the following information about the report

- "isgood" - Boolean variable indicating success or failure
- "msgs" - Vector of messages

- "rpttype" - Type of onbrand report ("Word" or "PowerPoint")
- "rptfigfmt" - Default figure formatting (orientation and dimensions)
- "rpttabfmt" - Default table formatting (orientation and dimensions)
- "rptdetails" - Contents of the yaml file
- "rptcont" - Contents of the report to generate

Examples

```
# We need an onbrand object to use below
library(onbrand)
obnd = read_template(
 template = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.docx"),
 mapping  = system.file(package="nlmixr2rpt", "templates","nlmixr_obnd_template.yaml"))

# We also need an nlmixr fit object
fit = fetch_fit_example()

# This reads in the report details
yaml_read_res = yaml_read_fit(
 obnd    = obnd,
 rptyaml = system.file(package="nlmixr2rpt", "examples", "report_fit_test.yaml"),
 fit     = fit)
```

Index

build_figures, 2
build_tables, 4

eval_str, 5

fetch_fdim, 6
fetch_fit_example, 7
fetch_option, 7

gen_eqn_table, 8
gen_pest_table, 9

mk_error_fig, 10
mk_error_tab, 11

process_ph, 11

report_fit, 12

safe_text, 13

write_figure, 14

yaml_read_fit, *12*, 15