

Package ‘Certara.Xpose.NLME’

September 30, 2026

Title Enhances 'xpose' Diagnostics for Pharmacometric Models from
'Certara.RsNLME' and Phoenix NLME

Version 2.1.0

Description Facilitates the creation of 'xpose' data objects from Nonlinear Mixed Effects (NLME) model outputs produced by 'Certara.RsNLME' or Phoenix NLME. This integration enables users to utilize all 'ggplot2'-based plotting functions available in 'xpose' for thorough model diagnostics and data visualization. Additionally, the package introduces specialized plotting functions tailored for covariate model evaluation, extending the analytical capabilities beyond those offered by 'xpose' alone.

URL <https://certara.github.io/R-Xpose-NLME/>

Depends R (>= 4.0)

License LGPL-3

Encoding UTF-8

LazyData true

LazyDataCompression xz

Suggests Certara.RsNLME, data.table, ellmer, flextable, gridExtra,
jsonlite, officer, readr, testthat (>= 3.0.0)

Imports dplyr, egg, GGally, ggplot2, magrittr, purrr, rlang, scales,
stringr, tibble, xpose

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author James Craig [aut, cre],
Michael Tomashevskiy [aut],
Soltanshahi Fred [aut],
Shuhua Hu [ctb],
Certara USA, Inc [cph, fnd]

Maintainer James Craig <james.craig@certara.com>

Repository CRAN

Date/Publication 2026-09-30 18:40:17 UTC

Contents

as_flextable.prmComparisonNlme	2
compare_etaShrinkageNlme	3
compare_prmNlme	4
eta_vs_cov	6
get_bootSummaryNlme	9
get_etaSubjectNlme	12
get_overallNlme	12
get_prmNlme	14
get_summaryNlme	15
nlme.cov.splom	18
nlme.par.vs.cov	19
nlme.ranpar.vs.cov	20
nlme.var.vs.cov	21
prm_vs_cov	21
res_vs_cov	24
update_etaShrinkageNlme	26
xpdb_ex_Nlme	26
xplot_box	27
xposeNlme	29
xposeNlmeModel	32
xpose_mcp_tools	34
Index	35

as_flextable.prmComparisonNlme

Render a parameter comparison as a flextable

Description

as_flextable() method for the prnComparisonNlme object returned by [compare_prmNlme\(\)](#). Draws solid separators between the diagnostic block, the parameter block, and the optional RSE block, and reports the number of models in the footer. Requires the flextable and officer packages (both Suggests).

Usage

```
## S3 method for class 'prnComparisonNlme'
as_flextable(x, format = NULL, max_show = NULL, ...)
```

Arguments

x	A prnComparisonNlme tibble from compare_prmNlme() .
format	One of "column" (models as columns) or "row" (models as rows, transposed). Defaults to the format attribute stored by compare_prmNlme() , or "column" when unset.

max_show	Maximum number of models to render. NULL or "" shows all models. Any CSV written by <code>compare_prmNlme()</code> still contains every model regardless of max_show.
...	Unused; present for S3 generic compatibility.

Value

A flextable object.

See Also

`compare_prmNlme()`

compare_etaShrinkageNlme

Compare original and filtered ETA shrinkage

Description

Computes filtered overall ETA shrinkage from the stored subject-by-ETA data and returns a comparison table without modifying the xpdb object.

Usage

```
compare_etaShrinkageNlme(
  xpdb,
  threshold = 1e-06,
  eta_threshold = NULL,
  eta_name = NULL,
  .problem = 1
)
```

Arguments

xpdb	An xpose_data object created by <code>xposeNlme</code> or <code>xposeNlmeModel</code> .
threshold	Global default threshold applied to all ETAs (default 1e-6). A subject/ETA row is removed when its variance-form shrinkage $\geq (1 - \text{threshold})^2$.
eta_threshold	Optional named numeric vector overriding threshold for specific ETAs, e.g. <code>c(nV = 1e-6, nC1 = 0.05)</code> .
eta_name	Optional character vector selecting a subset of ETAs to recompute. ETAs not listed keep their current summary values.
.problem	Problem number (default 1).

Value

A tibble with one row per processed ETA containing:

Eta ETA name

n_total Total number of subjects for this ETA

n_kept Number of subjects kept after filtering

n_removed Number of subjects removed

threshold_applied Resolved threshold used for this ETA

original_shrinkage Shrinkage computed from all subjects

filtered_shrinkage Shrinkage computed after filtering

omega_sd Population omega standard deviation for this ETA

compare_prmNlme

Compare NLME parameter estimates across multiple runs

Description

Builds a single wide table comparing parameter estimates (and optional %RSE) across two or more NLME runs, alongside a block of run-level diagnostics (~2LL, OFV diff, method, RetCode, condition, condition basis, nSub, nObs, and total runtime). It is the multi-model sibling of [get_summaryNlme\(\)](#): where `get_summaryNlme()` summarises one `xpose_data` object, `compare_prmNlme()` lines several up side by side for run-record style model comparison.

Usage

```
compare_prmNlme(
  x = NULL,
  dir = ".",
  runs = NULL,
  auto_detect = TRUE,
  max_runs = NULL,
  transform = c("untransformed", "sqrt_om2", "sqrt_exp_om2_minus_1"),
  param_order = c("original", "alphabetical"),
  rse_separate = FALSE,
  drop_dOFV = FALSE,
  output_file = NULL,
  format = c("column", "row"),
  log_file = "Table_log.txt"
)
```

Arguments

x	Optional named list of xpose_data objects (or a single xpose_data). When supplied, dir / runs / auto_detect are ignored. List names become the column headers and must be unique.
dir	Directory scanned for runs when x is NULL (default the working directory).
runs	Optional character vector of run names (subfolders of dir) to load explicitly, in the given order. Overrides auto-detection.
auto_detect	Logical; when x and runs are NULL, scan dir for completed runs (default TRUE).
max_runs	Optional cap on the number of successfully loaded runs to include. NULL or "" includes all runs. Failed imports do not count toward the cap.
transform	One of "untransformed" (default), "sqrt_om2", or "sqrt_exp_om2_minus_1"; applied to diagonal OMEGA only.
param_order	One of "original" (default: first run's model order, with parameters unique to later runs appended in those runs' order) or "alphabetical".
rse_separate	Logical; when TRUE, %RSE is shown in its own set of rows (suffixed " (RSE)") rather than in-line with each estimate. Accepts the strings "YES"/"NO" for backward compatibility.
drop_dOFV	Logical; when TRUE, omit the OFV diff row. Accepts "YES"/"NO".
output_file	Optional path; when set, the full table is written as a CSV in the chosen format.
format	One of "column" (default, models as columns) or "row" (models as rows, transposed). Controls the CSV layout and is stored on the returned object so as_flextable.prmComparisonNlme defaults to the same orientation.
log_file	Name of the excluded-run log written into dir during auto-detection (default "Table_log.txt"). Set to NULL to disable.

Details

Runs can be supplied three ways:

- As a pre-built named list of xpose_data objects (or a single xpose_data) via x. List names become the column labels.
- By explicit run name via runs, loaded from dir with [xposeNlme\(\)](#).
- By auto-detection (auto_detect = TRUE, the default when x and runs are both NULL): dir is scanned for model files (*.mdl or *.mmdl, case-insensitive) whose same-named run output folder exists, in alphanumeric order. Name each model so its output folder matches the model file (e.g. run001.mmdl beside a run001/ folder). A run is only included when its nlme7engine.log is present; runs that fail to load are recorded in log_file (written into dir) and skipped.

The transform setting affects only diagonal OMEGA rows. Diagonal SIGMA rows are kept on the reported [get_prmNlme\(\)](#) scale (for CEps this is the SD scale), so SIGMA values and %RSE do not change across transformation settings. %RSE on transformed OMEGA is propagated with the delta method.

The returned object carries n_header / n_rse attributes marking the leading diagnostic block and the trailing RSE block, which [as_flextable.prmComparisonNlme\(\)](#) uses to draw separators.

Rendering with `flextable` and CSV export via `output_file` are both optional; the core computation depends only on packages already imported by `Certara.Xpose.NLME`.

Timing:

The diagnostic row `total runtime (sec)` is engine-reported **CPU** time: the sum of the `runtime` and `covtime` rows in `xpdb$summary`, which are parsed from `nlme7engine.log` at import. That total can differ substantially from the wall-clock elapsed time shown by `print.rsnlme_fit / a fit object's runTime` (especially on multi-core runs). See also `get_overallNlme()` for the same distinction, including optional `runtime_wallclock` when an `xpdb` was built via `xposeNlmeModel()`.

Value

A tibble of class `prmcComparisonNlme` with a `Description` column followed by one column per run, carrying `n_header`, `n_rse`, `nModels`, `run_labels`, and `format` attributes.

See Also

`get_summaryNlme()`, `get_prmNlme()`, `get_overallNlme()`, `xposeNlme()`

Examples

```
## Not run:
# 1) Compare two already-imported runs.
xp1 <- xposeNlme(dir = "run001")
xp2 <- xposeNlme(dir = "run002")
compare_prmNlme(list(run001 = xp1, run002 = xp2))

# 2) Auto-detect every completed run under the working directory,
#     put %RSE on its own rows, and write a CSV.
tbl <- compare_prmNlme(
  rse_separate = TRUE,
  output_file = "TableofParameters.csv"
)

# 3) Render the comparison as a flextable (requires the flextable and
#     officer packages).
flextable::as_flextable(tbl)

## End(Not run)
```

eta_vs_cov

ETAs vs covariate Plot

Description

Plot ETAs against a continuous or categorical covariate.

Usage

```
eta_vs_cov(
  xpdb,
  covariate,
  mapping = NULL,
  drop_fixed = FALSE,
  group = "ID",
  type = "bpls",
  title = "ETAs vs @x | @run",
  subtitle = "Based on @nind individuals",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  onlyfirst = TRUE,
  facets,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
covariate	Character; String of covariate name
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
drop_fixed	Logical; Logic specifying whether ETAs having same value for the given covariate value should be removed from plotting
group	Grouping variable to be used for lines. ID by default
type	Character; String setting the type of plot to be used. Must be 'b' for categorical covariates, one or a combination of 'p','l','s' for continuous covariates.
title	Character; Plot title. Use NULL to remove.
subtitle	Character; Plot subtitle. Use NULL to remove.
caption	Character; Page caption. Use NULL to remove.
tag	Character; Plot identification tag. Use NULL to remove.
log	Character; String assigning logarithmic scale to axes, can be either "", 'x', 'y' or 'xy'.
guide	Logical; Should the guide (e.g. reference distribution) be displayed.
onlyfirst	Logical; Should the data be filtered to retain first value for each group/facet.
facets	Either a character string to use facet_wrap_paginate or a formula to use facet_grid_paginate .
.problem	The \$problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter or xplot_box .

Value

An object of class `xpose_plot`, `ggplot`, and `gg`. This object represents a customized plot created using `ggplot2`. The `xpose_plot` class provides additional metadata and integration with `xpose` workflows, allowing for advanced customization and compatibility with other `xpose` functions. Users can interact with the plot object as they would with any `ggplot2` object, including modifying aesthetics, adding layers, or saving the plot.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is `layer_option` where `layer` is one of the names defined in the list below and `option` is any option supported by this layer e.g. `boxplot_fill = 'blue'`, etc.

- box plot: options to `geom_boxplot`
- point plot: options to `geom_point`
- line plot: options to `geom_line`
- smooth plot: options to `geom_smooth`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

See Also

[xplot_scatter](#) [xplot_box](#)

Examples

```
eta_vs_cov(xpose::xpdb_ex_pk,  
  covariate = "WT",  
  type = "ps",  
  smooth_color = "red",  
  point_color = "green",  
  point_shape = "square",  
  point_alpha = .5,  
  point_size = 3  
)
```

```
eta_vs_cov(xpose::xpdb_ex_pk,  
  covariate = "AGE",  
  type = "ps",  
  facets = DOSE ~ variable,  
  guide = TRUE,  
  guide_color = "red",  
  guide_slope = 0,  
  guide_intercept = 0  
)
```

get_bootSummaryNlme	<i>Build a fused fit + bootstrap parameter summary</i>
---------------------	--

Description

Companion to `get_summaryNlme()` for `Certara.RsNLME::bootstrap()` results. Combines original-fit columns (`Estimate`, `%RSE`, `Shrinkage (%)`) with per-replicate bootstrap columns (`Bootstrap estimate (<metric>)`, `Bootstrap <pct>% CI`) using the same transform / shrinkage machinery.

Usage

```
get_bootSummaryNlme(
  bootResult,
  xpdb = NULL,
  transform = list(),
  units = list(),
  metric = c("Median", "Mean"),
  ci_level = NULL,
  return_code_ok = 1:3,
  digits = 3
)

## S3 method for class 'bootSummaryNlme'
print(x, ...)
```

Arguments

<code>bootResult</code>	Object returned by <code>Certara.RsNLME::bootstrap()</code> (an <code>rsnlme_boot</code> list of CSV-derived tables). Must carry at least <code>BootOverall</code> (with <code>Replicate</code> + <code>ReturnCode</code> columns), <code>BootThetaStacked</code> , and <code>BootOmegaStacked</code> .
<code>xpdb</code>	Optional <code>xpose_data</code> object from <code>xposeNlme()</code> / <code>xposeNlmeModel()</code> . When supplied, used as the original-fit source.
<code>transform</code>	Named list of per-parameter transforms keyed by parameter label. Same contract as <code>get_summaryNlme()</code> 's <code>transform</code> argument: either a preset string from the section's catalog or a <code>list(fn = ..., dfn = ..., name = ...)</code> spec, where the optional <code>name</code> sets the scale flag appended to the parameter name (a warning fires for a custom transform supplied without name). Applied to both the original-fit rows and the per-replicate bootstrap pool.
<code>units</code>	Named character vector (or list of length-1 character) keyed by parameter label, overriding the real-units <code>Unit</code> column for any row. Forwarded to <code>get_summaryNlme()</code> (m3) / the <code>fitSummary</code> pipeline (m2) and applied to the bootstrap-only output on m1. Keys not present in the relevant section emit a single warning and are ignored.
<code>metric</code>	"Median" (default) or "Mean" – statistic for the Bootstrap estimate (<metric>) column.

ci_level	Numeric in (0, 1); width of the percentile CI. When NULL (default) it inherits the bootstrap run's confidenceLevel (stored on bootResult), falling back to 0.95 when that metadata is absent. An explicit value always overrides.
return_code_ok	Integer vector of ReturnCode values to accept; replicates outside this set are dropped. Defaults to 1:3.
digits	Significant-digits count applied via signif() to the bootstrap columns and, when an original-fit summary is present, to the stored numeric Estimate, %RSE, and Shrinkage (%). Defaults to 3. The print.bootSummaryNlme method also honours digits for displayed precision (by setting pillar.sigfig for the duration of the print), so the original-fit columns show the same number of significant figures on screen as are stored – matching the pre-formatted bootstrap columns, which already carry digits in their character values.
x	A bootSummaryNlme tibble returned by get_bootSummaryNlme().
...	Further arguments passed to the underlying tibble print method.

Details

Three input modes drive what columns appear in the output:

m1: bootResult only No original-fit source; output is bootstrap-only – Section, Parameter, optional Unit, Bootstrap estimate (<metric>), and Bootstrap <pct>% CI. A message() notes that original-fit columns are unavailable and how to include them (initialEstimates = TRUE or xpdb). The residual-transform advisory (see below) still fires, since the default's fitness can't be ruled out without PML either way.

m2: bootResult carries an embedded fitSummary fitSummary is the original-fit source (when initialEstimates = TRUE was used on the bootstrap call). Omega off-diagonal covariance rows (Diagonal = FALSE) are dropped so the original-fit side stays variance-scale, matching m3's "off-diagonals excluded" contract; older fitSummary tables without a Diagonal column are treated as diagonal-only.

m3: explicit xpdb argument Original-fit columns come from get_summaryNlme(xpdb, ...). If bootResult\$fitSummary is also non-NULL a one-shot warning fires and xpdb wins.

Per-replicate filtering: replicates whose BootOverall\$returnCode falls outside return_code_ok are dropped silently before the metric and percentile CI are computed. The BootOverall table on the rsnlme_boot is the single source of truth – no on-disk reads.

The CI bounds are the empirical $(1 - \text{ci_level}) / 2$ and $1 - (1 - \text{ci_level}) / 2$ quantiles of the kept replicates, computed with stats::quantile()'s default method (type 7).

Soft-degrade for missing stacks: BootSigmaStacked and BootSecondaryStacked are net-new in the corresponding Certara.NLME8 release. When either is NULL (older NLME8 build) the function emits NA bootstrap cells for that section and a single warning naming the missing stacks plus the installed Certara.NLME8 version. BootThetaStacked and BootOmegaStacked predate the new release and are always available.

Output formatting (shared with get_summaryNlme()): when a non-identity transform is active the values are shown on the transformed scale only, and the scale label is appended to the shared Parameter name in parentheses – nV (CV%), CEps (SD), or a custom name for list(fn=, dfn=, name=) specs. Identity / raw rows keep the bare name. The Unit column carries real units only and

is dropped when every row is dimensionless. The bootstrap CI is a single character column formatted as "lo - hi" (a dash range, no brackets). Numeric digits is applied via signif() to both the bootstrap columns and, when present, the original-fit Estimate, %RSE, and Shrinkage (%).

Residual-shape advisories: a per-sigma warning fires when PML classifies a sigma as additive or otherwise non-proportional while it is reported with the multiplicative_cv default. Classification requires PML, which is only available when xpdb is supplied (m3); on m1/m2 no per-sigma warning is emitted. The general default-transform message() is broader: it fires for any defaulted sigma that isn't *proven* proportional, which on m1/m2 (no PML to check) means it always fires – mirroring [get_summaryNlme\(\)](#)'s own no-PML behaviour. Set options(xposeNlme.summary.quiet_default_warning = TRUE) to silence it.

Value

A tibble (class bootSummaryNlme) with Section, Parameter (carrying a (CV%) / (SD) / custom scale flag when a non-identity transform is active), optional original-fit columns (Estimate, %RSE, Shrinkage (%)), an optional real-units Unit column, and the two bootstrap columns (the CI formatted as "lo - hi"). Carries the attributes n_used, n_total, ci_level, return_code_ok, transform, metric, fitSource ("xpdb" / "embedded" / "none"). The bootSummaryNlme class carries a print method that honours the digits argument for displayed precision.

See Also

[get_summaryNlme\(\)](#)

Examples

```
## Not run:
fit_boot <- Certara.RsNLME::bootstrap(model, ...)

# m1: bootstrap-only summary, no fit source.
get_bootSummaryNlme(fit_boot)

# m2: fused summary using the bootstrap's embedded fitSummary
# (initialEstimates = TRUE on the bootstrap call).
fit_boot_init <- Certara.RsNLME::bootstrap(model,
                                           initialEstimates = TRUE, ...)
get_bootSummaryNlme(fit_boot_init)

# m3: fused summary against an explicit xpdb -- preferred when the
# xpdb has its own provenance (covariates, residuals, posthoc) that
# fitSummary doesn't capture.
xp <- xposeNlmeModel(fit_model)
get_bootSummaryNlme(fit_boot, xpdb = xp)

# Custom transform on a sigma applied to both original-fit and
# bootstrap columns; `name` sets the scale flag on the parameter name.
get_bootSummaryNlme(
  fit_boot_init,
  transform = list(
    CEps = list(
      fn = function(s) 200 * s,
```

```

        dfn = function(s) 200,
        name = "2xCV%"
    )
)
)

## End(Not run)

```

get_etaSubjectNlme	<i>Access subject-level ETA data</i>
--------------------	--------------------------------------

Description

Retrieves the stored subject-by-ETA table from the `xpdb$special` slot with `type == "eta_subject"`.

Usage

```
get_etaSubjectNlme(xpdb, .problem = 1)
```

Arguments

<code>xpdb</code>	An <code>xpose_data</code> object created by xposeNlme or xposeNlmeModel .
<code>.problem</code>	The problem number to extract (default 1).

Value

A tibble with columns ID, Eta, ETA_VAL, ETA_SE.

get_overallNlme	<i>Access NLME model overall fit results</i>
-----------------	--

Description

Access model fit diagnostics from an `xpdb` object generated by `xposeNlme`.

Usage

```

get_overallNlme(
  xpdb,
  .problem = 1,
  .subprob = 0,
  .method = NULL,
  conditionNumber = NULL
)

```

Arguments

xpdb	An xpose data base object from which the model output file data will be extracted. Only objects generated by xposeNlme are supported.
.problem	The problem to be used.
.subprob	The subproblem to be used.
.method	The estimation method to be used.
conditionNumber	Optional character; overrides the basis/scope used for the returned Condition (one of "CovarianceFixef", "CorrelationFixef", "CovarianceFull", "CorrelationFull", or the legacy aliases "Covariance" / "Correlation" - same vocabulary as Certara.RsNLME::engineParams(conditionNumber = ...)). NULL (the default) returns whatever basis was recorded at import time (see Details) unchanged. When supplied, this is computed lazily by re-reading the run's dmp.txt ("...Fixef" scopes) or Covariance.csv ("...Full" scopes) from the original run directory , which must still exist. A missing directory or a missing Covariance.csv for a "...Full" request warns and sets Condition to NA while still updating ConditionBasis to the requested label. An unrecognized string warns and leaves both columns unchanged. See Details for why "...Full" scopes need Covariance.csv specifically.

Details

This function returns only the Overall.csv fit-statistics table (for example objective function value, AIC, BIC, log-likelihood). It does not contain run metadata such as estimation settings or timing.

Condition and ConditionBasis report the reported-condition- number diagnostic (see engineParams(conditionNumber = ...) in Certara.RsNLME). ConditionBasis labels which basis/scope Condition actually reflects, e.g. "Correlation (full)" vs. "Covariance (fixed effects)". Actual-over-requested precedence (this is the value returned when conditionNumber is NULL): when the engine's own out.txt reports a value (any conditionNumber mode), it is used as-is; only when that is unavailable does this function fall back to an independent, theta-only, covariance-basis recompute (always labeled "Covariance (fixed effects)") - so Condition may not be comparable across rows/fits that used different conditionNumber modes unless ConditionBasis matches.

Passing conditionNumber forces a specific basis/scope instead, regardless of what the engine actually used, by recomputing from the raw covariance data. "...Fixef" scopes (fixed effects only) can always be recomputed from dmp.txt\$varFix. "...Full" scopes (all free population parameters: fixed effects, free residual error, free Omega) require Covariance.csv specifically - dmp.txt does not carry the joint covariance needed, because dmp.txt\$omegaSE is only the per-element standard error of each Omega entry, with no cross-covariance to fixed effects/error or between different Omega entries.

Run metadata lives in xpdb\$summary (view it with summary(xpdb)). Two timing rows may appear there and are measured differently: runtime (and covtime) are engine-reported CPU times parsed from nlme7engine.log that sum across workers and can overstate the actual wait on multi-core runs, whereas runtime_wallclock is the elapsed wall-clock time captured from a self-describing Certara.RsNLME fit (via [xposeNlmeModel\(\)](#)) and is generally smaller on multi-core runs. runtime_wallclock is never available on the file-based [xposeNlme\(dir = ...\)](#) path.

Value

A tibble for single problem/subproblem.

See Also

[xposeNlme](#), [xposeNlmeModel](#)

Examples

```
# Store the parameter table
prmOverall <- get_overallNlme(xpdb_ex_Nlme)
```

get_prmNlme	<i>Access NLME model parameter estimates</i>
-------------	--

Description

Access model parameter estimates from an xpdb object generated by xposeNlme.

Usage

```
get_prmNlme(
  xpdb,
  .problem = 1,
  .subprob = 0,
  .method = NULL,
  digits = 6,
  show_all = FALSE,
  level = 0.95
)
```

Arguments

xpdb	An xpose data base object from which the model output file data will be extracted. Only objects generated by xposeNlme are supported.
.problem	The problem to be used.
.subprob	The subproblem to be used.
.method	The estimation method to be used.
digits	Integer specifying the number of significant digits to be displayed.
show_all	Logical specifying whether the 0 off-diagonal omega elements should be removed from the output or not.
level	Numeric specifying confidence level to compute confidence intervals, which are calculated based on Student's t distribution. Set to NULL (or any zero-length value) to skip confidence intervals.

Value

A tibble for single problem/subproblem.

See Also

[xposeNlme](#)

Examples

```
# Store the parameter table
prm <- get_prmNlme(xpdb_ex_Nlme)

# Set the desired number of significant digits to display results

# Note: To have results displayed in the number of significant digits
# specified in the digits argument, one needs to make sure that
# the value of pillar.sigfig option (default value is 3) is greater
# than or equal to this specified value.

options(pillar.sigfig = 6)
get_prmNlme(xpdb_ex_Nlme, digits = 4)
```

get_summaryNlme	<i>Build a parameter summary table for an NLME xpdb</i>
-----------------	---

Description

Produces a single tibble combining fixed effects, random effects, residual errors, and secondary parameters, along with their estimates and %RSE on the chosen scale. Shrinkage for random effects and residual errors is also included. The transform applied to each row controls both the displayed Estimate and the corresponding %RSE, which is computed on the transformed scale via the delta method. Built-in presets carry analytic derivatives; for transforms outside the catalog, supply a custom fn with its derivative dfn.

Usage

```
get_summaryNlme(
  xpdb,
  .problem = 1,
  .subprob = 0,
  .method = NULL,
  transform = list(),
  units = list(),
  shrinkage = c("engine", "sd", "var"),
  digits = NULL,
  append_flag = TRUE,
  emit_advisories = TRUE
```

```
)

## S3 method for class 'summaryNlme'
print(x, ...)
```

Arguments

xpdb	An xpose_data object created by xposeNlme() or xposeNlmeModel().
.problem	Problem number (default 1). Mirrors get_prmNlme().
.subprob	Subproblem number (default 0). Mirrors get_prmNlme().
.method	Estimation method filter (default NULL). Mirrors get_prmNlme().
transform	Named list of per-parameter transforms keyed by prmTable\$label. Each value is either a preset string from the section's catalog or a list(fn = ..., dfn = ..., name = ...) spec, where the optional name sets the scale flag appended to the parameter name.
units	Named character vector keyed by prmTable\$label that populates or overrides the Unit column with physical units for matching rows.
shrinkage	Shrinkage calculation method, one of "engine" (default), "sd", or "var". "engine": uses the standard-deviation-based shrinkage values reported directly by the engine (read from xpdb\$summary) – eta shrinkage $1 - SD(\eta)/\omega$ (with $\omega = \sqrt{\Omega}$, the model standard deviation) and eps shrinkage $1 - SD(IWRES)$. Note the engine computes the eta SD with denominator n (population) but the eps SD with denominator $n - 1$ (sample). "sd": recomputes the standard-deviation-based shrinkage using R's sd() (denominator $n - 1$). This differs from "engine" only for eta shrinkage, since the engine's eps path already uses $n - 1$. "var": recomputes a variance-based shrinkage using R's var() (denominator $n - 1$) – eta shrinkage $1 - Var(\eta)/\Omega$ and eps shrinkage $1 - Var(IWRES)$. The recompute paths ("sd" / "var") use subject-level etas for eta shrinkage when the model has random effects (skipped for naive-pooled or any other no-ranef() fit) and the IWRES column in xpdb\$data for eps shrinkage; multi-residual models additionally need the embedded PML source to map each ObsName row to its driving sigma.
digits	Optional significant-digits count. When non-NULL, signif() is applied to the stored numeric Estimate, %RSE, and Shrinkage (%), and the print.summaryNlme method shows that many significant figures (by setting pillar.sigfig for the duration of the print). NULL (default) keeps full precision in both the stored values and the printed display.
append_flag	When TRUE (default), the scale flag (CV% / SD / custom name) is appended to Parameter for non-identity transforms.
emit_advisories	When TRUE (default), the residual-transform advisory message and the per-sigma type warnings are emitted. Set to FALSE to silence both.
x	A summaryNlme tibble returned by get_summaryNlme().
...	Further arguments passed to the underlying tibble print method.

Details

Default transforms per section:

- Fixed effects: raw.
- Random effects (omegas): lognormal_cv, defined as $100\sqrt{\exp(\omega^2) - 1}$, where ω^2 is the variance stored in `prmTable$value`.
- Residual error (sigmas): multiplicative_cv (100σ). This default is applied **uniformly to every sigma**, regardless of the error-model shape declared in PML – and choosing the wrong scale doesn't correct itself, it just mislabels the number. For purely additive or combined error models the %CV label is misleading – override with `transform = list(<sigma> = "raw")` (the engine reports the residual error as a standard deviation) or a custom fn.

When the PML source is available, each sigma's shape is inferred directly from its `observe()` expression by differentiating it with respect to the error variable (base R's `stats::D()`) and checking whether that derivative is a constant (additive error) or proportional to the noise-free prediction with no curvature (proportional error – the one shape `multiplicative_cv` is exact for). This inference is syntax-only and best-effort: it works directly on whatever PML the `xpdb` happens to embed, without assuming it came from any particular model-building tool. The advisory message is skipped only when every defaulted sigma is proven proportional; it fires for additive error, for any other non-proportional shape (combined, power, ...), and whenever the shape can't be determined at all (a function outside `D()`'s derivative table, unusual syntax, or no PML source) – in that last case the safer default is to warn rather than assume. An extra per-sigma warning fires for any sigma whose inferred role is additive or otherwise non-proportional. Set `emit_advisories = FALSE` to silence both the message and the per-sigma warnings, or `options(xposeNlme.summary.quiet_default_warning = TRUE)` to silence only the message. `get_bootSummaryNlme()`'s `bootResult`-only and `embedded-fitSummary` input modes have no PML to check either, so they inherit this same "no PML source" behaviour: the default message always fires for a defaulted sigma, while the per-sigma warning (which needs a known shape to name) stays silent.

- Secondary: raw.

Built-in preset catalog:

- Fixed effects: raw (custom fn always available).
- Random effects: raw, lognormal_cv, normal_sd. `normal_sd` returns the standard deviation $\sqrt{\Omega}$ (i.e. `fn = sqrt(value)`), not the variance.
- Residual error: raw, log_additive_cv, multiplicative_cv. `log_additive_cv` is $100\sqrt{\exp(\sigma^2) - 1}$ and `multiplicative_cv` is 100σ .

Custom transform spec: `transform = list(<label> = list(fn = function(x) ..., dfn = function(x) ..., name = dfn` is optional; when omitted, %RSE falls back to the raw scale and a single warning per call lists the affected parameters. `name` is the optional scale flag appended to the parameter name; a custom transform supplied without name triggers a warning and leaves the name unflagged.

Scale flag and units: when a non-identity transform is active the scale label is appended to `Parameter` in parentheses – `nV (CV%)`, `CEps (SD)`, or the custom name. Identity / raw rows keep the bare name. The `Unit` column carries physical units only (user units or the model's structural-parameter units) and is dropped when every row is dimensionless.

Off-diagonal omega/sigma rows are excluded entirely. transform and units are keyed by prmTable\$label (the PML-source name – tvCl, nV, CEps); names not present among the labels emit a single warning per call and are ignored.

Value

A tibble (class summaryNlme) with columns Section, Parameter (carrying a (CV%) / (SD) / custom scale flag when a non-identity transform is active), Estimate, %RSE, Shrinkage (%), and – only when at least one row resolves to a non-empty physical unit – Unit. The summaryNlme class carries a print method that honours the digits argument for displayed precision.

See Also

[get_prmNlme\(\)](#), [get_overallNlme\(\)](#), [get_etaSubjectNlme\(\)](#)

Examples

```
## Not run:
# 1) Default output on a log-normal IIV + proportional error model.
xp <- xposeNlmeModel(fit)
get_summaryNlme(xp)

# 2) Per-parameter override on an additive error model. The engine
#   reports residual error as a standard deviation, so `raw` shows the
#   SD directly (no misleading %CV flag).
get_summaryNlme(
  xp,
  transform = list(EEps = "raw"),
  units      = list(EEps = "ng/mL")
)

# 3) Custom transform for combined add-mult error.
get_summaryNlme(
  xp,
  transform = list(
    CEps = list(
      fn = function(s) 100 * s,
      dfn = function(s) 100
    )
  )
)

## End(Not run)
```

Description

Use to create covariates scatterplot.

Usage

```
nlme.cov.splom(
  xpdb,
  covColNames,
  ggupper = list(continuous = "cor", combo = "box_no_facet", discrete = "count", na =
    "na"),
  gglower = list(continuous = GGally::wrap("smooth", alpha = 0.3, size = 0.1), combo =
    "facethist", discrete = "facetbar", na = "na"),
  ggdiag = list(continuous = "densityDiag", discrete = "barDiag", na = "naDiag"),
  ...
)
```

Arguments

xpdb	An xpose database object.
covColNames	Character vector of covariates to build the matrix
ggupper	See ggpairs() upper argument.
gglower	See ggpairs() lower argument.
ggdiag	See ggpairs() diag argument.
...	Parameters to be passed to ggpairs() .

Value

[ggmatrix](#) object.

Examples

```
nlme.cov.splom(xpdb = xpdb_ex_Nlme,
  covColNames = c("sex", "wt", "age")
)
```

nlme.par.vs.cov

Plot parameter estimates against covariates

Description

Use to create a stack of plots of parameter estimates plotted against covariates.

Usage

```
nlme.par.vs.cov(xpdb, covColNames, nrow = 1, ncol = 1, ...)
```

Arguments

xpdb	An xpose database object.
covColNames	Character vector of covariates to build the matrix.
nrow	Number of rows.
ncol	Number of columns; if ncol=1, each gtable object is treated separately.
...	Parameters to be passed to ggarrange() .

Value

List of [gtable](#)

Examples

```
nlme.par.vs.cov(
  xpdb = xpdb_ex_Nlme,
  covColNames = c("sex", "wt", "age")
)
```

nlme.ranpar.vs.cov	<i>Plot random parameter estimates against covariates</i>
--------------------	---

Description

Use to create a stack of plots of random parameter estimates plotted against covariates.

Usage

```
nlme.ranpar.vs.cov(xpdb, covColNames, nrow = 1, ncol = 1, ...)
```

Arguments

xpdb	An xpose database object.
covColNames	Character vector of covariates to build the matrix.
nrow	Number of rows.
ncol	Number of columns; if ncol=1, each gtable object is treated separately.
...	Parameters to be passed to ggarrange()

Value

List of [gtable](#)

Examples

```
nlme.ranpar.vs.cov(xpdb = xpose::xpdb_ex_pk,
  covColNames = c("SEX", "CLCR", "AGE")
)
```

nlme.var.vs.cov

*Build multiple plots for selected variable vs covariates***Description**

The type of plot depends on the type of covariate: boxplot for categorical, geom_point and geom_smooth for continuous.

Usage

```
nlme.var.vs.cov(xpdb, covColNames, nrow = 1, ncol = 1, yVar = "WRES", ...)
```

Arguments

xpdb	An xpose database object.
covColNames	Character vector of covariates to build the matrix.
nrow	Number of rows.
ncol	Number of columns; if ncol=1, each gtable object is treated separately.
yVar	Variable from xpdb data to build a plot.
...	Parameters to be passed to ggarrange()

Value

List of [gtable](#)

Examples

```
nlme.var.vs.cov(
  xpdb = xpdb_ex_Nlme,
  covColNames = c("sex", "wt", "age"),
  yVar = "WRES",
  nrow = 2,
  ncol = 2
)
```

prm_vs_cov

*Parameter vs covariate Plot***Description**

Plot Parameters against a continuous or categorical covariate.

Usage

```

prm_vs_cov(
  xpdb,
  covariate,
  mapping = NULL,
  drop_fixed = FALSE,
  group = "ID",
  type = "bpls",
  title = "Parameters vs @x | @run",
  subtitle = "Based on @nind individuals",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = FALSE,
  onlyfirst = FALSE,
  facets,
  .problem,
  quiet,
  ...
)

```

Arguments

xpdb	An xpose database object.
covariate	Character; String of covariate name
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
drop_fixed	Logical; logic specifying whether structural parameters having same value for the given covariate value should be removed from plotting
group	Grouping variable to be used for lines. ID by default
type	Character; String setting the type of plot to be used. Must be 'b' for categorical covariates, one or a combination of 'p','l','s' for continuous covariates.
title	Character; Plot title. Use NULL to remove.
subtitle	Character; Plot subtitle. Use NULL to remove.
caption	Character; Page caption. Use NULL to remove.
tag	Character; Plot identification tag. Use NULL to remove.
log	Character; String assigning logarithmic scale to axes, can be either "", 'x', 'y' or 'xy'.
guide	Logical; Enable guide display (e.g. unity line).
onlyfirst	Logical; Should the data be filtered to retain first value for each group/facet.
facets	Either a character string to use facet_wrap_paginate or a formula to use facet_grid_paginate .
.problem	The \$problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter or xplot_box .

Value

An object of class `xpose_plot`, `ggplot`, and `gg`. This object represents a customized plot created using `ggplot2`. The `xpose_plot` class provides additional metadata and integration with `xpose` workflows, allowing for advanced customization and compatibility with other `xpose` functions. Users can interact with the plot object as they would with any `ggplot2` object, including modifying aesthetics, adding layers, or saving the plot.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is `layer_option` where `layer` is one of the names defined in the list below and `option` is any option supported by this layer e.g. `boxplot_fill = 'blue'`, etc.

- box plot: options to `geom_boxplot`
- point plot: options to `geom_point`
- line plot: options to `geom_line`
- smooth plot: options to `geom_smooth`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

See Also

[xplot_scatter](#) [xplot_box](#)

Examples

```
prm_vs_cov(xpose::xpdb_ex_pk,
  covariate = "AGE", type = "ps",
  log = "y",
  yscale_breaks = scales::trans_breaks("log10", function(x) 10^x),
  yscale_labels = scales::trans_format("log10", scales::math_format(10^.x)),
  caption = NULL
)

prm_vs_cov(xpose::xpdb_ex_pk,
  covariate = "SEX",
  type = "b",
  boxplot_fill = "blue",
  boxplot_color = "black",
  boxplot_outlier_color = "red"
)
```

res_vs_cov	<i>Residuals vs covariate plot</i>
------------	------------------------------------

Description

Plot Residuals against a continuous or categorical covariate.

Usage

```
res_vs_cov(
  xpdb,
  mapping = NULL,
  covariate,
  res = "CWRES",
  group = "ID",
  type = "bpls",
  title = "Residuals vs @x | @run",
  subtitle = "Based on @nind individuals",
  caption = "@dir",
  tag = NULL,
  log = NULL,
  guide = TRUE,
  facets,
  .problem,
  quiet,
  ...
)
```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. point_color).
covariate	Character; String of covariate name
res	Character; String of residual name; CWRES by default.
group	Grouping variable to be used for lines. ID by default
type	Character; String setting the type of plot to be used. Must be 'b' for categorical covariates, one or a combination of 'p','l','s' for continuous covariates.
title	Character; Plot title. Use NULL to remove.
subtitle	Character; Plot subtitle. Use NULL to remove.
caption	Character; Page caption. Use NULL to remove.
tag	Character; Plot identification tag. Use NULL to remove.
log	Character; String assigning logarithmic scale to axes, can be either "", 'x', 'y' or 'xy'.
guide	Logical; Should the guide (e.g. reference distribution) be displayed.

facets	Either a character string to use <code>facet_wrap_paginate</code> or a formula to use <code>facet_grid_paginate</code> .
.problem	The \$problem number to be used. By default returns the last estimation problem.
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on <code>xplot_scatter</code> or <code>xplot_box</code> .

Value

An object of class `xpose_plot`, `ggplot`, and `gg`. This object represents a customized plot created using `ggplot2`. The `xpose_plot` class provides additional metadata and integration with `xpose` workflows, allowing for advanced customization and compatibility with other `xpose` functions. Users can interact with the plot object as they would with any `ggplot2` object, including modifying aesthetics, adding layers, or saving the plot.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is `layer_option` where `layer` is one of the names defined in the list below and `option` is any option supported by this layer e.g. `boxplot_fill = 'blue'`, etc.

- box plot: options to `geom_boxplot`
- point plot: options to `geom_point`
- line plot: options to `geom_line`
- smooth plot: options to `geom_smooth`
- xscale: options to `scale_x_continuous` or `scale_x_log10`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

See Also

`xplot_scatter` `xplot_box`

Examples

```
res_vs_cov(xpose::xpdb_ex_pk,
  covariate = "SEX",
  type = "b",
  res = "WRES"
)

res_vs_cov(xpose::xpdb_ex_pk,
  covariate = "AGE",
  type = "ps",
  res = c("CWRES", "WRES", "IRES", "IWRES")
)
```

update_etaShrinkageNlme	<i>Update ETA shrinkage with subject-level filtering</i>
-------------------------	--

Description

Recomputes overall ETA shrinkage from stored subject-by-ETA data, applying threshold-based filtering to remove subjects whose individual shrinkage is too close to 1 (variance form). Returns an updated xpdb with the etashk summary row replaced.

Usage

```
update_etaShrinkageNlme(  
  xpdb,  
  threshold = 1e-06,  
  eta_threshold = NULL,  
  eta_name = NULL,  
  .problem = 1  
)
```

Arguments

xpdb	An xpose_data object created by xposeNlme or xposeNlmeModel .
threshold	Global default threshold applied to all ETAs (default 1e-6). A subject/ETA row is removed when its variance-form shrinkage $\geq (1 - \text{threshold})^2$.
eta_threshold	Optional named numeric vector overriding threshold for specific ETAs, e.g. <code>c(nV = 1e-6, nC1 = 0.05)</code> .
eta_name	Optional character vector selecting a subset of ETAs to recompute. ETAs not listed keep their current summary values.
.problem	Problem number (default 1).

Value

A new xpdb with the etashk value in `xpdb$summary` updated.

xpdb_ex_Nlme	<i>XposeNlme example xpose_data object</i>
--------------	--

Description

An example `xpose::xpose_data` object for use in examples and tests. It holds a one-compartment (Clearance-parameterized) PK model with three covariates, fit with Phoenix NLME (`phx/nlme`) and imported via [xposeNlmeModel\(\)](#).

Format

An `xpose::xpose_data` object.

Details

The model was built from the real `Certara.RsNLME::pkData` dataset (16 subjects, 112 observations, single bolus dose) using a clearance-parameterized one-compartment model with a proportional residual error, diagonal random effects on V and Cl, and the covariates age and wt (continuous) plus sex (categorical: male/female). Roughly reproduced with:

```
library(Certara.RsNLME)
model <- pkmodel(
  parameterization = "Clearance",
  numCompartments = 1,
  data = pkData,
  ID = "Subject",
  Time = "Act_Time",
  A1 = "Amount",
  CObs = "Conc",
  workingDir = tempdir()
) |>
addCovariate("age") |>
addCovariate("wt") |>
addCovariate("sex", type = "Categorical",
             levels = c(0, 1), labels = c("male", "female")) |>
colMapping(age = "Age", wt = "BodyWeight", sex = "Gender")
```

Examples

```
print(xpdb_ex_Nlme)
```

`xplot_box`*Default xpose box plot function*

Description

Manually generate categorical covariate box plots against eta.

Usage

```
xplot_box(
  xpdb,
  mapping = NULL,
  type = "b",
  guide = FALSE,
  yscale = "continuous",
```

```

    title = NULL,
    subtitle = NULL,
    caption = NULL,
    tag = NULL,
    plot_name = "box_plot",
    gg_theme,
    xp_theme,
    opt,
    quiet,
    ...
  )

```

Arguments

xpdb	An xpose database object.
mapping	List of aesthetics mappings to be used for the xpose plot (e.g. <code>point_color</code>).
type	String setting the type of plot to be used. Only 'b' applicable.
guide	Enable guide display (e.g. unity line).
yscale	Scale type for y axis (e.g. 'continuous', 'discrete', 'log10').
title	Plot title. Use NULL to remove.
subtitle	Plot subtitle. Use NULL to remove.
caption	Page caption. Use NULL to remove.
tag	Plot identification tag. Use NULL to remove.
plot_name	Name to be used by <code>xpose::xpose_save()</code> when saving the plot.
gg_theme	A complete ggplot2 theme object (e.g. <code>ggplot2::theme_classic</code>), a function returning a complete ggplot2 theme, or a change to the current <code>gg_theme</code> .
xp_theme	A complete xpose theme object (e.g. <code>theme_xp_default</code>) or a list of modifications to the current <code>xp_theme</code> (e.g. <code>list(point_color = 'red', line_linetype = 'dashed')</code>).
opt	A list of options in order to create appropriate data input for ggplot2. For more information see data_opt .
quiet	Logical, if FALSE messages are printed to the console.
...	Any additional aesthetics to be passed on xplot_scatter .

Value

An object of class `xpose_plot`, `ggplot`, and `gg`. This object represents a customized plot created using `ggplot2`. The `xpose_plot` class provides additional metadata and integration with xpose workflows, allowing for advanced customization and compatibility with other xpose functions. Users can interact with the plot object as they would with any `ggplot2` object, including modifying aesthetics, adding layers, or saving the plot.

Faceting

Every xpose plot function has built-in faceting functionalities. Faceting arguments are passed to the functions `facet_wrap_paginate` when the `facets` argument is a character string (e.g. `facets = c('SEX', 'MED1')`) or `facet_grid_paginate` when `facets` is a formula (e.g. `facets = SEX~MED1`). All xpose plot functions accept all the arguments for the `facet_wrap_paginate` and `facet_grid_paginate` functions e.g. `dv_vs_ipred(xpdb_ex_pk, facets = SEX~MED1, ncol = 3, nrow = 3, page = 1, margins = TRUE, labeller = 'label_both')`.

Faceting options can either be defined in plot functions (e.g. `dv_vs_ipred(xpdb_ex_pk, facets = 'SEX')`) or assigned globally to an `xpdb` object via the `xp_theme` (e.g. `xpdb <- update_themes(xpdb_ex_pk, xp_theme = list(facets = 'SEX'))`). In the latter example all plots generate from this `xpdb` will automatically be stratified by `SEX`.

By default, some plot functions use a custom stratifying variable named `variable`, e.g. `eta_distrib()`. When using the `facets` argument, `variable` needs to be added manually e.g. `facets = c('SEX', 'variable')` or `facets = c('SEX', 'variable')`, but is optional, when using the `facets` argument in `xp_theme` `variable` is automatically added whenever needed.

Layers mapping

Plots can be customized by mapping arguments to specific layers. The naming convention is `layer_option` where `layer` is one of the names defined in the list below and `option` is any option supported by this layer e.g. `boxplot_fill = 'blue'`, etc.

- box plot: options to `geom_boxplot`
- yscale: options to `scale_y_continuous` or `scale_y_log10`

See Also

[xplot_scatter](#) [xplot_qq](#)

Examples

```
# Categorical Covariate MED1 vs ETA1
xplot_box(xpose::xpdb_ex_pk, ggplot2::aes(x = MED1, y = ETA1))

# Categorical Covariate SEX vs CL
xplot_box(xpose::xpdb_ex_pk, ggplot2::aes(x = SEX, y = CL))
```

Description

Imports results of an NLME run into xpose database Use to import NLME model output files into `xpdb` object that is compatible with existing model diagnostic function in Xpose package.

Usage

```
xposeNlme(
  dir = "",
  modelName = "",
  dmpFile = "dmp.txt",
  dmp.txt = NULL,
  dataFile = "data1.txt",
  logFile = "nlme7engine.log",
  ConvergenceData = NULL,
  progresstxt = NULL
)
```

Arguments

<code>dir</code>	Path to NLME Run directory. Current working directory is used if <code>dir</code> not given.
<code>modelName</code>	name of the model to be written in <code>xpdb\$summary\$value</code> with run label
<code>dmpFile</code>	NLME generated output file.
<code>dmp.txt</code>	NLME generated output from <code>dmpFile</code> (substitutes <code>dmpFile</code> if presented).
<code>dataFile</code>	Input file for NLME Run.
<code>logFile</code>	engine log file
<code>ConvergenceData</code>	optional convergence info, as either a data frame (the long-format Scenario, Iter, Parameter, Value shape returned by <code>Certara.RsNLME::fitmodel()</code>) or a path to a CSV with the same columns (e.g. "ConvergenceData.csv"). When NULL (the default), <code>dir</code> is scanned for <code>ConvergenceData.csv</code> , then <code>progresstxt</code> / <code>progress.txt</code> is used as a fallback.
<code>progresstxt</code>	optional NLME-generated <code>progress.txt</code> with convergence info. Useful for SCM scenario folders and Phoenix temporary result folders where <code>progress.txt</code> is retained and <code>ConvergenceData.csv</code> may be absent. For normal <code>Certara.RsNLME::fitmodel()</code> keep-dirs prefer <code>ConvergenceData.csv</code> (auto-scanned) or <code>ConvergenceData=</code> ; the job-folder <code>progress.txt</code> is usually deleted after summarize. When NULL (the default), <code>progress.txt</code> is only used as a fallback if <code>ConvergenceData</code> is not supplied and no <code>ConvergenceData.csv</code> is found in <code>dir</code> . An explicit path forces that file even if a CSV is also present; a missing explicit path is an error.

Details

Not all functionality from the `xpose` package is supported.

Bootstrap working directories are not a supported `xposeNlme()` source: artifacts may be overwritten or replicate-concatenated (`dmp.txt`, `progress.txt`). When bootstrap markers are detected (`Boot*.csv`, `bootstrap_nlme7engine.log`, `bootstrap_*.rds`):

- last-replicate `dmp.txt` typically has no residuals table and no `residuals.csv` – import **stops** with a message pointing at `get_bootSummaryNlme`;

- if a residuals table is already present (embedded or leftover residuals.csv from a prior fitmodel()), import continues after a warning. Those CSVs may belong to a different fit (including different initial estimates, or bootstrap(..., overwriteFitDir = TRUE) that overwrote dmp.txt without replacing the CSVs). GOF is not the bootstrap last replicate.

For bootstrap parameter summaries, use `get_bootSummaryNlme` on the `bootstrap()` return value (not this directory). For GOF / convergence plots, import the original fit with `xposeNlme()` on the fit folder or `xposeNlmeModel` on the fit result.

Run metadata is stored in the summary tibble of the returned `xpdb` object (`xpdb$summary`, or rendered with `summary(xpdb)`). On this file-based entry point, every metadata row beyond the long-standing `descr/ofv/method/runtime` set is parsed from `nlme7engine.log` **only** – `ode_solver`, `stderr_algorithm`, `stderr_method`, `xnorderagq`, and `fastOptimization` appear when the log records them (missing fields are silently omitted, never an error), and the method row discriminates FOCE-ELS from LAPLACIAN using the log's Hessian-variant flag. `stderr_algorithm` and `stderr_method` are both omitted entirely for IT2S-EM (which cannot estimate standard errors); `xnorderagq` and `fastOptimization` only ever appear for FOCE-ELS/LAPLACIAN, and `xnorderagq` is further omitted when the log shows OuterAD (which forces the adaptive Gaussian quadrature order to 1 internally). `epsshk/etashk` are likewise omitted when not applicable (no residual-error model / the NAIVE-POOLED method, respectively).

`xposeNlme()` deliberately never reads `rtol`, `atol`, or `nmxstep` (not written to the engine log), never reads `runtime_wallclock`, `RsNLMEVersion`, `timestart`, or `timestop` (only known to Certara.RsNLME at fit time, not to the engine). Arguments files are intentionally not read, because requested settings can differ from the engine run.

Value

`xpdb` object

See Also

`xposeNlmeModel`, `get_bootSummaryNlme`

Examples

```
## Not run:
# files in arguments supposed to be in the current working directory;
# ConvergenceData.csv (if present) is picked up automatically:
xp <- xposeNlme(
  dir = getwd(),
  modelName = "PMLModel",
  dmpFile = "dmp.txt",
  dataFile = "data1.txt",
  logFile = "nlme7engine.log"
)

# SCM / Phoenix folders that retain progress.txt but not
# ConvergenceData.csv:
xp <- xposeNlme(
  dir = "~/scm_archive/cstep000__0",
  modelName = "Base",
```

```

    dataFile = "../data1.txt",
    progresstxt = "progress.txt"
  )

  # using dmp.txt structure and Convergence Data loaded previously:
  xp <- xposeNlme(
    dir = "~/Model1/",
    modelName = "Model1",
    dmp.txt = dmp.txt,
    dataFile = "Data.csv",
    logFile = "nlme7engine.log",
    ConvergenceData = ConvergenceData
  )

  # explore unique covariate plots specific to Certara.Xpose.NLME:
  nlme.cov.splom(xp, covColNames = c("AGE", "WT"))
  nlme.par.vs.cov(xp, covColNames = c("AGE", "WT"))

  res_vs_cov(xp, covariate = "AGE", res = "IWRES")

  # or use existing plotting functions from the xpose package
  library(xpose)
  dv_vs_pred(xp)
  res_vs_idv(xp)

  ## End(Not run)

```

xposeNlmeModel

Creates xpose database from Certara.RsNLME objects

Description

Imports results of an NLME run into xpose database Use to import NLME model object and NLME object output into xpdb object that is compatible with existing model diagnostic function in Xpose package.

Usage

```
xposeNlmeModel(model, fitmodelOutput)
```

Arguments

model	NlmePmlModel model class object generated by Certara.RsNLME package. Optional when fitmodelOutput is a self-describing fit (carries \$model); in that case the model is read from fitmodelOutput\$model. When both are supplied, the explicit model argument wins.
-------	--

`fitmodelOutput` the output object of `Certara.RsNLME::fitmodel()` run. Newer versions of `Certara.RsNLME` return a self-describing list that carries the model and the resolved engine parameters; in that case `xposeNlmeModel(fitmodelOutput)` can be called with a single argument.

Details

Not all functionality from the `xpose` package is supported.

Run metadata is stored in the summary tibble of the returned `xpdb` object. Access it directly with `xpdb$summary` or render it to the console with `summary(xpdb)`. This is distinct from `get_overallNlme()`, which returns the `Overall.csv` fit-statistics table (objective function, AIC, BIC, and similar).

When `fitmodelOutput` is a self-describing fit that carries `$params`, `$runTime`, and/or `$RsNLMEVersion`, the summary tibble gains additional rows: `ode_solver`, `rtol`, `atol`, `nmxstep`, `stderr_algorithm`, `stderr_method`, `xnorderagq`, `fastOptimization`, `runtime_wallclock`, and `RsNLMEVersion`, plus `timestart` / `timestop` at the global problem. `stderr_algorithm` is the standard-error *algorithm* (Hessian / Sandwich / Fisher Score / Auto-Detect / None); `stderr_method` is the finite-difference scheme used for the standard-error computation (`params@xstderr`: none / central-difference / forward-difference) and is shown when the finite-difference flag is nonzero (`IFLAGSTDERR` / `params@xstderr`), even if the algorithm name could not be resolved (e.g. Auto-Detect with no resolution prose). Both SE rows are omitted for IT2S-EM, which cannot estimate standard errors. `xnorderagq` and `fastOptimization` only ever appear for FOCE-ELS/LAPLACIAN; `xnorderagq` is further omitted when the engine actually ran with OuterAD (which forces the adaptive Gaussian quadrature order to 1 internally). `ode_solver`, `stderr_algorithm`, and `fastOptimization` always reflect the engine's *actual* run (preferring `nlme7engine.log` over the `params` request whenever the log has the relevant flag) – `params` is only used as a fallback when the log lacks the flag (e.g. no log file at all).

The file-based `xposeNlme(dir = ...)` entry point recovers `ode_solver`, `stderr_algorithm`, `stderr_method`, `xnorderagq`, and `fastOptimization` from `nlme7engine.log` when the log records them, but never `rtol`, `atol`, `nmxstep`, `runtime_wallclock`, `RsNLMEVersion`, `timestart`, or `timestop` – those are not written to the engine log, only resolved by `Certara.RsNLME` at fit time. `xposeNlme()` also never reads `args/params` files (for example `nlmeargs.txt`, `jobArgsCombined.txt`, `jobControlFile.txt`): those record *requested* settings, which can go stale relative to what the engine actually ran, so surfacing them in `xp$summary` would be misleading. See `xposeNlme` for the full file-path contract.

The runtime row reports engine-reported CPU time, which sums across workers and can exceed the actual wait on multi-core runs; `runtime_wallclock` reports the elapsed wall-clock time from `fitmodelOutput$runTime$elapsed` and is generally smaller on multi-core runs.

Value

`xpdb` object

Examples

```
## Not run:
library(Certara.RsNLME)
library(Certara.Xpose.NLME)

model <- pkmodel(
```

```

    parameterization = "Clearance",
    numCompartments = 2,
    data = pkData,
    ID = "Subject",
    Time = "Act_Time",
    A1 = "Amount",
    CObs = "Conc"
  )

  fit <- fitmodel(model)

  # Two-argument form (works with all RsNLME versions):
  xp <- xposeNlmeModel(model = model, fitmodelOutput = fit)

  # One-argument form (RsNLME with self-describing fitmodel output):
  xp <- xposeNlmeModel(fit)

  ## End(Not run)

```

xpose_mcp_tools

Build the Certara.Xpose.NLME MCP tool set

Description

Returns a list of `ellmer::tool()` objects for the Certara.R MCP host. This is the builder referenced by `inst/mcp/tools/manifest.json`. The host calls it with the launch profile's provider groups; tools whose group is not requested are omitted.

Usage

```
xpose_mcp_tools(groups = c("data", "interpretation", "comparison"))
```

Arguments

groups	Character vector of tool groups to include. One or more of "data", "interpretation", "comparison". Defaults to all.
--------	---

Value

A list of `ellmer::tool()` objects (empty list when `ellmer` is not installed or no group matches).

Examples

```

## Not run:
tools <- xpose_mcp_tools()
tools <- xpose_mcp_tools(groups = c("data", "interpretation"))

## End(Not run)

```

Index

`as_flextable.prmComparisonNlme`, [2](#)
`as_flextable.prmComparisonNlme()`, [5](#)

`compare_etaShrinkageNlme`, [3](#)
`compare_prmNlme`, [4](#)
`compare_prmNlme()`, [2](#), [3](#)

`data_opt`, [28](#)

`ellmer:::tool()`, [34](#)
`eta_vs_cov`, [6](#)

`facet_grid_paginate`, [7](#), [22](#), [25](#), [29](#)
`facet_wrap_paginate`, [7](#), [22](#), [25](#), [29](#)

`get_bootSummaryNlme`, [9](#), [30](#), [31](#)
`get_bootSummaryNlme()`, [17](#)
`get_etaSubjectNlme`, [12](#)
`get_etaSubjectNlme()`, [18](#)
`get_overallNlme`, [12](#), [33](#)
`get_overallNlme()`, [6](#), [18](#)
`get_prmNlme`, [14](#)
`get_prmNlme()`, [5](#), [6](#), [18](#)
`get_summaryNlme`, [15](#)
`get_summaryNlme()`, [4](#), [6](#), [9–11](#)
`ggarrange`, [20](#), [21](#)
`ggmatrix`, [19](#)
`ggpairs`, [19](#)
`gtable`, [20](#), [21](#)

`nlme.cov.splom`, [18](#)
`nlme.par.vs.cov`, [19](#)
`nlme.ranpar.vs.cov`, [20](#)
`nlme.var.vs.cov`, [21](#)

`print.bootSummaryNlme`
 (`get_bootSummaryNlme`), [9](#)
`print.summaryNlme` (`get_summaryNlme`), [15](#)
`prm_vs_cov`, [21](#)

`res_vs_cov`, [24](#)

`theme_xp_default`, [28](#)

`update_etaShrinkageNlme`, [26](#)

`xpdb_ex_Nlme`, [26](#)
`xplot_box`, [7](#), [8](#), [22](#), [23](#), [25](#), [27](#)
`xplot_qq`, [29](#)
`xplot_scatter`, [7](#), [8](#), [22](#), [23](#), [25](#), [28](#), [29](#)
`xpose_mcp_tools`, [34](#)
`xposeNlme`, [3](#), [12–15](#), [26](#), [29](#), [33](#)
`xposeNlme()`, [5](#), [6](#)
`xposeNlmeModel`, [3](#), [12–14](#), [26](#), [31](#), [32](#)
`xposeNlmeModel()`, [6](#)