

Package: pmxdchk (via r-universe)

October 29, 2026

Title Data Visualization and Quality Checking for Pharmacometric Datasets

Version 0.0.0.9000

Description Data quality checks for NONMEM-style and ADPPK pharmacometric datasets, with a Shiny application to review the findings alongside individual and population concentration-time profiles.

License Apache License (≥ 2)

URL <https://github.com/pharmaverse/pmxdchk>

BugReports <https://github.com/pharmaverse/pmxdchk/issues>

Depends R ($\geq 4.1.0$)

Imports bslib, config, dplyr, DT, ggplot2, golem, purrr, readr, rlang, shiny, stats, tibble

Suggests pharmaverseadam, rmarkdown, shinytest2, testthat ($\geq 3.0.0$), knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

LazyData true

Config/pak/sysreqs cmake make libuv1-dev libx11-dev zlib1g-dev

Repository <https://pharmaverse.r-universe.dev>

Date/Publication 2026-10-10 01:12:20 UTC

RemoteUrl <https://github.com/pharmaverse/pmxdchk>

RemoteRef HEAD

RemoteSha a67c1d8ebd3be830fd67e1470d666630ce3b55ad

Contents

adppk_corrupted	2
adppk_example	2
check_catalogue	3
default_thresholds	3
infer_study_type	4
run_app	5
run_nmpk_checks	6
Index	7

adppk_corrupted	<i>Example ADPPK dataset with injected data-quality issues</i>
-----------------	--

Description

A variant of [adppk_example] with known problems injected across several domains so the checks and the profile flag overlay have something to detect. Injected issues include an invalid EVID, a zero and an extreme dose amount, a decreasing time, a high predose and an extreme concentration, an MDV/DV inconsistency, an inconsistent within-subject SEX, an implausible baseline weight, a subject with observations but no dose, a duplicate record, and an exclusion flag without a reason (via an added ‘EXCLFL’ column).

Usage

adppk_corrupted

Format

The same structure as [adppk_example] plus an ‘EXCLFL’ column.

Source

Derived from pharmaverseadam::adppk; see data-raw/corrupted_examples.R.

adppk_example	<i>Example ADPPK dataset (clean)</i>
---------------	--------------------------------------

Description

A copy of the ADPPK analysis dataset from **pharmaverseadam**, provided as a built-in clean example for trying the app and the checks. It uses CDISC ADPPK variable names (e.g. ‘USUBJIDN’, ‘AFRLT’, ‘NFRLT’, ‘WTBL’, ‘PARAMN’), which the app’s variable mapping resolves to canonical NONMEM names.

Usage

```
adppk_example
```

Format

A data frame with one row per event record and CDISC ADPPK columns including 'USUBJIDN', 'STUDYIDN', 'AFRLT', 'NFRLT', 'EVID', 'MDV', 'DV', 'AMT', 'CMT', 'II', 'SS', 'PARAMN', and baseline covariates.

Source

```
pharmaverseadam::adppk
```

check_catalogue	<i>Catalogue of available checks</i>
-----------------	--------------------------------------

Description

Lists every check the package runs, with the rule it applies and what to do when it flags.

Usage

```
check_catalogue()
```

Value

A tibble with one row per check: 'check_id', 'title', 'domain', 'severity', 'study_types', 'rule', 'guidance'.

Examples

```
check_catalogue()
```

default_thresholds	<i>Default check thresholds</i>
--------------------	---------------------------------

Description

Conservative, data-driven defaults for checks that take tunable cutoffs. Passed into every check by `[run_nmpk_checks()]`; override individual elements to retune without touching check code.

Usage

```
default_thresholds()
```

Value

A named list of thresholds.

Examples

```
th <- default_thresholds()
th$min_quantifiable_n <- 3
```

infer_study_type	<i>Infer likely study type(s) from an NMPK dataset</i>
------------------	--

Description

Data-driven inference for ‘CORE-STUDYTYPE-001’, following the *Study Type Map* sheet of the QC checklist. Operates on a dataset already mapped to canonical variable names. The result is informational: the app presents it for user confirmation before checks are selected.

Usage

```
infer_study_type(data)
```

Arguments

data A data frame with canonical variable names.

Value

A tibble with columns ‘study_type’, ‘applicable’ (logical), ‘reason’.

Examples

```
df <- data.frame(
  ID = c(1, 1, 2, 2), TIME = c(0, 1, 0, 1), EVID = c(1, 0, 1, 0),
  AMT = c(100, 0, 100, 0), ADDL = c(5, NA, 5, NA), II = c(24, NA, 24, NA)
)
infer_study_type(df)
```

run_app	<i>Run the Shiny Application</i>
---------	----------------------------------

Description

Run the Shiny Application

Usage

```
run_app(  
  onStart = NULL,  
  options = list(),  
  enableBookmarking = NULL,  
  uiPattern = "/",  
  ...  
)
```

Arguments

- | | |
|-------------------|---|
| onStart | A function that will be called before the app is actually run. This is only needed for shinyAppObj, since in the shinyAppDir case, a global.R file can be used for this purpose. |
| options | Named options that should be passed to the runApp call (these can be any of the following: "port", "launch.browser", "host", "quiet", "display.mode" and "test.mode"). You can also specify width and height parameters which provide a hint to the embedding environment about the ideal height/width for the app. |
| enableBookmarking | Can be one of "url", "server", or "disable". The default value, NULL, will respect the setting from any previous calls to enableBookmarking() . See enableBookmarking() for more information on bookmarking your app. |
| uiPattern | A regular expression that will be applied to each GET request to determine whether the ui should be used to handle the request. Note that the entire request path must match the regular expression in order for the match to be considered successful. |
| ... | arguments to pass to golem_opts. See <code>“?golem::get_golem_options”</code> for more details. |

run_nmpk_checks

*Run NONMEM dataset quality checks***Description**

Headless entry point and the same path the Shiny app uses. Applies the variable mapping, orders checks by their prerequisites, evaluates skip conditions, runs each check, and returns a tidy findings tibble.

Usage

```
run_nmpk_checks(
  data,
  mapping = NULL,
  thresholds = default_thresholds(),
  study_type = "All"
)
```

Arguments

data	A data frame of a NONMEM-style (NMPK) dataset.
mapping	Optional named character vector mapping canonical variable names to user column names (‘c(ID = "USUBJID", TIME = "TAFD")’). When ‘NULL’, columns are assumed to already use canonical names.
thresholds	A list of thresholds; see [default_thresholds()]. Elements not supplied fall back to their defaults.
study_type	Character vector of confirmed study types used to gate conditional checks. Defaults to “All”.

Value

A tibble with one row per check: ‘check_id’, ‘title’, ‘domain’, ‘severity’, ‘status’, ‘n_flagged’, ‘message’. The full [new_check_result()] objects are attached as the “results” attribute for the app and reporting layers.

Examples

```
df <- data.frame(
  ID = c(1, 1), TIME = c(0, 1), EVID = c(1, 0),
  MDV = c(1, 0), DV = c(0, 5), AMT = c(100, 0)
)
run_nmpk_checks(df)
```

Index

* **datasets**

adppk_corrupted, [2](#)

adppk_example, [2](#)

adppk_corrupted, [2](#)

adppk_example, [2](#)

check_catalogue, [3](#)

default_thresholds, [3](#)

enableBookmarking(), [5](#)

infer_study_type, [4](#)

run_app, [5](#)

run_nmpk_checks, [6](#)