Package: imedR (via r-universe)

November 3, 2026

Type Package

Title Guarded Evidence Synthesis for the Indian Microplastics Evidence Database

Version 0.3.3

Description A research-grade, guardrail-first interface to the Indian Microplastics Evidence Database (IMED). It provides release-data access, relational and scientific validation, comparability-aware filtering, descriptive and inferential statistics, evidence-synthesis eligibility, meta-analysis helpers, publication figures, mapping, provenance and citation retrieval, and manuscript-oriented outputs. Unsafe pooling across incompatible matrices, denominators, mass bases and comparability classes is blocked by default.

License MIT + file LICENSE

URL https://github.com/ChanikyaNaiduCN7/imedR

BugReports https://github.com/ChanikyaNaiduCN7/imedR/issues

Encoding UTF-8

Depends R (>= 4.2.0)

Imports dplyr, ggplot2, rlang, tibble, readr

Suggests testthat (>= 3.0.0), sf, metafor, knitr, rmarkdown, scales

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Chanikya Naidu [aut, cre]

Maintainer Chanikya Naidu <thefisherieschanikyaneeti@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-03 21:13:38 UTC

RemoteUrl https://github.com/cran/imedR

RemoteRef HEAD

RemoteSha 3da12a12037725f327d50a196e4efe81c656302f

Contents

imed_analysis_set . . . 2
imed_check_comparability . . . 4
imed_check_size_window . . . 5
imed_cite . . . 5
imed_compare . . . 6
imed_data . . . 7
imed_describe . . . 7
imed_enrich_measurements . . . 8
imed_evidence_profile . . . 9
imed_export_figure . . . 9
imed_filter . . . 10
imed_flag_holds . . . 11
imed_forest . . . 12
imed_funnel . . . 13
imed_leave_one_out . . . 14
imed_manuscript_table . . . 15
imed_map . . . 15
imed_meta_eligibility . . . 16
imed_meta_mean . . . 17
imed_meta_prevalence . . . 18
imed_meta_regression . . . 18
imed_missingness . . . 19
imed_plot . . . 20
imed_plot_composition . . . 21
imed_plot_methods . . . 22
imed_provenance . . . 22
imed_results_table . . . 23
imed_sensitivity . . . 24
imed_stats . . . 25
imed_tables . . . 26
imed_tables_data . . . 26
imed_validate . . . 27
imed_validation_report . . . 27
imed_version . . . 28

Index **29**

`imed_analysis_set` *Create a manuscript/synthesis-ready analysis set*

Description

This is the preferred entry point for quantitative analysis.

Usage

```
imed_analysis_set(
  x,
  comparability_class,
  matrix = NULL,
  mass_basis = NULL,
  metric_type = "Abundance",
  include_holds = FALSE
)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
comparability_class	The single ‘Comparability_Class‘ to analyse (required).
matrix	Optional ‘Matrix_Code‘ value(s) to keep.
mass_basis	Optional ‘Mass_Basis‘ value(s) to keep.
metric_type	‘Metric_Type‘ to keep (default ‘"Abundance"‘).
include_holds	If ‘FALSE‘ (default), records flagged as scientific HOLDs are removed.

Value

A ‘tibble‘ of the selected measurement records that passed the comparability check. It has the additional class ‘imed_analysis_set‘ and the attribute ‘comparability_class‘. The function stops with an informative message if the selected records are not comparable.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
aset <- imed_analysis_set(toy, comparability_class = "CLASS_A")
class(aset)
nrow(aset)
```

`imed_check_comparability` *Check scientific comparability of an analysis subset*

Description

Check scientific comparability of an analysis subset

Usage

```
imed_check_comparability(
  x,
  action = c("error", "warn", "return"),
  require_single_unit = TRUE
)
```

Arguments

`x`	An ‘imed_db‘, or a measurement-level data frame that includes comparability metadata (for example from ‘imed_filter()‘).
`action`	error, warn, or return.
`require_single_unit`	Require one reported unit. Convertible units should first be explicitly standardized outside this function.

Value

A ‘tibble‘ with columns ‘comparable‘ (logical) and ‘issue‘ (character; ‘"None"‘ when no problem is found) if ‘action = "return"‘. For ‘action = "error"‘ or ‘"warn"‘ the function returns invisibly a single logical value (‘TRUE‘ if the records are comparable) and stops or warns when they are not.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_check_comparability(toy, action = "return")
```

imed_check_size_window

Guard analytical size-window compatibility

Description

Uses linked SIZE_METHODS records to identify unresolved or heterogeneous analytical/reporting lower bounds. It never substitutes mesh/filter pore size for an analytical lower size limit.

Usage

```
imed_check_size_window(
  x,
  sample_id = NULL,
  action = c("error", "warn", "return")
)
```

Arguments

x	An imed_db.
sample_id	Optional sample IDs.
action	error, warn or return.

Value

A ‘tibble‘ with columns ‘compatible‘ (logical) and ‘issue‘ (character; ‘"None"‘ when no problem is found) if ‘action = "return"‘. For ‘action = "error"‘ or ‘"warn"‘ the function returns invisibly a single logical value (‘TRUE‘ if the size windows are compatible) and stops or warns when they are not.

Examples

```
db <- imed_data(tables = "SIZE_METHODS", validate = FALSE)
imed_check_size_window(db, action = "return")
```

imed_cite

Primary study citations for a subset

Description

Primary study citations for a subset

Usage

```
imed_cite(x, study_id = NULL)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.

Value

A ‘tibble‘ with one row per study and the citation fields that are available (‘Study_ID‘, ‘Authors‘, ‘Year‘, ‘Title‘, ‘Journal‘, ‘DOI‘).

Examples

```
db <- imed_data(tables = "STUDIES", validate = FALSE)
head(imed_cite(db))
```

---

imed_compare *Grouped comparison summary*

Description

Grouped comparison summary

Usage

```
imed_compare(x, group, value = "Mean")
```

Arguments

x	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
group	Grouping field (unquoted or quoted name).
value	Name of the numeric measurement field to summarise (default ‘"Mean"‘).

Value

A ‘tibble‘ of descriptive statistics with one row per group, as returned by ‘imed_describe()‘ with a ‘group‘ argument.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
```

```
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_compare(toy, group = "Group")
```

`imed_data` *Load the bundled frozen IMED v1.0 analytical snapshot*

Description

The package exposes 19 core scientific/schema tables. Internal extraction, workflow and QC-log sheets are intentionally not part of the analytical API.

Usage

```
imed_data(tables = NULL, validate = TRUE)
```

Arguments

`tables`	Optional table names.
`validate`	Run validation after loading.

Value

An object of class ‘c("imed_db", "list")‘: a named list with one ‘tibble‘ for each requested table. The attribute ‘imed_release‘ holds the data release (‘"1.0"‘) and, when ‘validate = TRUE‘, the attribute ‘validation‘ holds the ‘tibble‘ returned by ‘imed_validate()‘.

Examples

```
db <- imed_data(tables = c("STUDIES", "SITES"), validate = FALSE)
class(db)
names(db)
```

`imed_describe` *Descriptive statistics for a guarded analysis set*

Description

Descriptive statistics for a guarded analysis set

Usage

```
imed_describe(x, value = "Mean", group = NULL)
```

Arguments

x	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
value	Name of the numeric measurement field to summarise (default ‘"Mean"‘).
group	Optional grouping field (unquoted or quoted name).

Value

A ‘tibble‘ with the columns ‘n‘ (number of finite values), ‘mean‘, ‘sd‘, ‘median‘, ‘q1‘, ‘q3‘, ‘min‘ and ‘max‘. If ‘group‘ is supplied there is one row per group, with the grouping field as the first column.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_describe(toy)
imed_describe(toy, group = "Group")
```

---

imed_enrich_measurements

Join measurement records to their scientific comparability metadata

---

Description

Join measurement records to their scientific comparability metadata

Usage

```
imed_enrich_measurements(x)
```

Arguments

x	An imed_db.

Value

A ‘tibble‘ containing the measurement records joined, by ‘Measurement_ID‘, to their comparability metadata. Columns that occur in both tables carry the suffix ‘.cmp‘ in the comparability copy.

Examples

```
db <- imed_data(tables = c("MEASUREMENTS", "MEASUREMENT_COMPARABILITY"),
                validate = FALSE)
head(imed_enrich_measurements(db))
```

---

imed_evidence_profile *Evidence-quality/reporting profile*

Description

Evidence-quality/reporting profile

Usage

```
imed_evidence_profile(x)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.

Value

A one-row ‘tibble‘ with the number of QA/QC records (‘n_records‘) and the numbers that report a procedural blank (‘procedural_blank_yes‘), a recovery test (‘recovery_test_reported‘) and chemical confirmation (‘chemical_confirmation_reported‘).

Examples

```
db <- imed_data(tables = "METHODS_QAQC", validate = FALSE)
imed_evidence_profile(db)
```

---

imed_export_figure *Export a publication figure*

Description

Export a publication figure

Usage

```
imed_export_figure(
  plot,
  filename,
  width = 180,
  height = 120,
  units = "mm",
  dpi = 600
)
```

Arguments

plot	A ‘ggplot‘ object, for example from ‘imed_plot()‘.
filename	Output file path; the format follows the file extension.
width	Figure width in ‘units‘ (default 180).
height	Figure height in ‘units‘ (default 120).
units	Units for ‘width‘ and ‘height‘ (default ‘"mm"‘).
dpi	Resolution in dots per inch (default 600).

Value

The path ‘filename‘, returned invisibly. The function is called for its side effect of writing the figure to disk.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
p <- imed_plot(toy, type = "histogram")
out <- tempfile(fileext = ".png")
imed_export_figure(p, out, width = 90, height = 60, dpi = 100)
file.exists(out)
unlink(out)
```

---

imed_filter *Filter IMED with scientific guard metadata retained*

Description

Filter IMED with scientific guard metadata retained

Usage

```
imed_filter(
  x,
  matrix = NULL,
  comparability_class = NULL,
  mass_basis = NULL,
  metric_type = NULL,
```

```
  study_id = NULL,
  include_holds = FALSE
)
```

Arguments

x	An 'imed_db' (HOLD flags are added automatically) or a measurement-level data frame.
matrix	Optional 'Matrix_Code' value(s) to keep.
comparability_class	Optional 'Comparability_Class' value(s) to keep.
mass_basis	Optional 'Mass_Basis' value(s) to keep.
metric_type	Optional 'Metric_Type' value(s) to keep.
study_id	Optional character vector of 'Study_ID' values used to restrict the records.
include_holds	If 'FALSE' (default), records flagged as scientific HOLDs are removed.

Value

A 'tibble' of the records that satisfy all supplied filters. If 'x' is an 'imed_db', the result is the measurement-level table with comparability metadata and two added columns: '.imed_hold' (logical) and '.imed_hold_reason' (character).

Examples

```
db <- imed_data(tables = c("MEASUREMENTS", "MEASUREMENT_COMPARABILITY"),
                validate = FALSE)
nrow(imed_filter(db))
nrow(imed_filter(db, include_holds = TRUE))
```

---

imed_flag_holds *Flag measurement-level scientific HOLDs*

Description

Flag measurement-level scientific HOLDs

Usage

```
imed_flag_holds(x)
```

Arguments

x	An 'imed_db', or a measurement-level data frame that includes comparability metadata.

Value

A ‘tibble‘ of measurement records, with comparability metadata when ‘x‘ is an ‘imed_db‘, and two added columns: ‘.imed_hold‘ (logical, ‘TRUE‘ for records that must not enter strict synthesis) and ‘.imed_hold_reason‘ (character, the reason or ‘NA‘).

Examples

```
db <- imed_data(tables = c("MEASUREMENTS", "MEASUREMENT_COMPARABILITY"),
                validate = FALSE)
h <- imed_flag_holds(db)
table(h$.imed_hold)
```

`imed_forest` *Forest plot for an imedR meta-analysis*

Description

Forest plot for an imedR meta-analysis

Usage

```
imed_forest(fit, ...)
```

Arguments

`fit`	A fitted ‘metafor‘ ‘rma‘ object, for example from ‘imed_meta_mean()‘.
`...`	Further arguments passed to ‘metafor::forest()‘.

Value

The fitted model ‘fit‘, returned invisibly. The function is called for its side effect of drawing a forest plot on the active graphics device.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  toy <- data.frame(
    Study_ID = paste0("S", 1:12),
    Comparability_Class = "CLASS_A",
    Reported_Unit = "items/L",
    Metric_Type = "Abundance",
    Strict_Synthesis_Eligibility = "YES",
    Group = rep(c("g1", "g2"), 6),
    Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
    SD = rep(c(0.6, 0.9), 6),
    N = rep(c(10, 12, 8), 4)
  )
  fit <- imed_meta_mean(toy)
  imed_forest(fit)
}
```

`imed_funnel` *Funnel plot for an imedR meta-analysis*

Description

A funnel plot is diagnostic only and should not be interpreted as a direct test of publication bias, especially with small or heterogeneous evidence.

Usage

```
imed_funnel(fit, ...)
```

Arguments

`fit`	A fitted ‘metafor‘ ‘rma‘ object, for example from ‘imed_meta_mean()‘.
`...`	Further arguments passed to ‘metafor::funnel()‘.

Value

The fitted model ‘fit‘, returned invisibly. The function is called for its side effect of drawing a funnel plot on the active graphics device.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  toy <- data.frame(
    Study_ID = paste0("S", 1:12),
    Comparability_Class = "CLASS_A",
    Reported_Unit = "items/L",
    Metric_Type = "Abundance",
    Strict_Synthesis_Eligibility = "YES",
    Group = rep(c("g1", "g2"), 6),
    Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
    SD = rep(c(0.6, 0.9), 6),
    N = rep(c(10, 12, 8), 4)
  )
  fit <- imed_meta_mean(toy)
  imed_funnel(fit)
}
```

`imed_leave_one_out` *Leave-one-study-out influence analysis for a guarded mean meta-analysis*

Description

Leave-one-study-out influence analysis for a guarded mean meta-analysis

Usage

```
imed_leave_one_out(x, method = "REML")
```

Arguments

`x`	A guarded analysis set that passes ‘imed_meta_eligibility()‘, as for ‘imed_meta_mean()‘.
`method`	Between-study variance estimator passed to ‘metafor‘ (default ‘"REML"‘).

Value

A ‘tibble‘ with one row per omitted study. The first column, ‘omitted‘, identifies the study that was left out; the remaining columns are the pooled estimate and heterogeneity statistics returned by ‘metafor::leave1out()‘ after that study is removed.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  toy <- data.frame(
    Study_ID = paste0("S", 1:12),
    Comparability_Class = "CLASS_A",
    Reported_Unit = "items/L",
    Metric_Type = "Abundance",
    Strict_Synthesis_Eligibility = "YES",
    Group = rep(c("g1", "g2"), 6),
    Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
    SD = rep(c(0.6, 0.9), 6),
    N = rep(c(10, 12, 8), 4)
  )
  imed_leave_one_out(toy)
}
```

imed_manuscript_table *Study-characteristics table for manuscripts*

Description

Study-characteristics table for manuscripts

Usage

```
imed_manuscript_table(x, study_id = NULL)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.

Value

A ‘tibble‘ with one row per study and the descriptive fields that are available (‘Study_ID‘, ‘Authors‘, ‘Year‘, ‘States_UTs‘, ‘Matrices_Studied‘, ‘Journal‘, ‘DOI‘, ‘Entry_Status‘).

Examples

```
db <- imed_data(tables = "STUDIES", validate = FALSE)
head(imed_manuscript_table(db))
```

imed_map *Coordinate-provenance-aware IMED map*

Description

Coordinate-provenance-aware IMED map

Usage

```
imed_map(x, geo_min = c("GEO-4", "GEO-3", "GEO-2"), study_id = NULL)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
geo_min	Character vector of coordinate classes to retain, chosen from ‘"GEO-4"‘, ‘"GEO-3"‘ and ‘"GEO-2"‘ (default all three). Only sites whose stored class is in this set and that have numeric coordinates are drawn.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.

Value

A ‘ggplot‘ object with one point for each site that has numeric coordinates and a retained coordinate class. The function stops if the suggested package ‘sf‘ is not installed.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  db <- imed_data(tables = c("SITES", "SITE_COORDINATE_CLASS"),
                  validate = FALSE)
  imed_map(db)
}
```

---

imed_meta_eligibility *Evaluate eligibility for quantitative evidence synthesis*

Description

Checks class/unit/basis, usable uncertainty, and study-level independence.

Usage

```
imed_meta_eligibility(x, require_independent_studies = TRUE)
```

Arguments

x	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
require_independent_studies	If TRUE, duplicate Study_ID is flagged.

Value

A ‘tibble‘ with columns ‘eligible‘ (logical) and ‘issue‘ (character). The value ‘"None"‘ in ‘issue‘ means that no problem was found; otherwise each row describes one reason why the records cannot enter a standard meta-analysis.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
```

```
  N = rep(c(10, 12, 8), 4)
)
imed_meta_eligibility(toy)
```

`imed_meta_mean` *Random-effects meta-analysis of compatible raw means*

Description

Intended only for one comparable outcome/unit/basis with independent study effects. Requires metafor.

Usage

```
imed_meta_mean(x, method = "REML")
```

Arguments

`x`	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
`method`	Between-study variance estimator passed to ‘metafor‘ (default ‘"REML"‘).

Value

A fitted random-effects model of class ‘c("rma.uni", "rma")‘ from ‘metafor::rma.uni()‘, using the study means as effects and the squared standard error of each mean as its variance. The pooled estimate is on the original measurement scale.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  toy <- data.frame(
    Study_ID = paste0("S", 1:12),
    Comparability_Class = "CLASS_A",
    Reported_Unit = "items/L",
    Metric_Type = "Abundance",
    Strict_Synthesis_Eligibility = "YES",
    Group = rep(c("g1", "g2"), 6),
    Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
    SD = rep(c(0.6, 0.9), 6),
    N = rep(c(10, 12, 8), 4)
  )
  fit <- imed_meta_mean(toy)
  fit
}
```

imed_meta_prevalence *Random-effects meta-analysis of prevalence*

Description

Uses BIOTA_CONTEXT N_Positive/N_Examined and a logit transformed proportion. Multiple rows per Study_ID are blocked unless the user preselects one independent effect per study.

Usage

```
imed_meta_prevalence(x, study_id = NULL, method = "REML")
```

Arguments

x	An ‘imed_db‘ (required if ‘study_id‘ is used) or a ‘BIOTA_CONTEXT‘-style data frame with ‘N_Positive‘ and ‘N_Examined‘.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.
method	Between-study variance estimator passed to ‘metafor‘ (default ‘"REML"‘).

Value

A fitted random-effects model of class ‘c("rma.uni", "rma")‘ from ‘metafor::rma.uni()‘. Effects are logit-transformed proportions, so the pooled estimate is on the log-odds scale; use ‘plogis()‘ to back-transform it to a proportion.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  biota <- data.frame(
    Sample_ID = paste0("SA", 1:6),
    N_Positive = c(5, 12, 3, 20, 8, 15),
    N_Examined = c(30, 40, 25, 60, 35, 50)
  )
  fit <- imed_meta_prevalence(biota)
  plogis(coef(fit))
}
```

imed_meta_regression *Meta-regression with minimum-information guard*

Description

Requires at least 10 independent effects by default and at least 5 effects per model coefficient. These are guardrails, not guarantees of adequate power.

Usage

```
imed_meta_regression(x, moderators, min_k = 10, method = "REML")
```

Arguments

x	A guarded analysis set that passes ‘imed_meta_eligibility()‘.
moderators	Character vector of field names in the measurement table to use as moderators.
min_k	Minimum number of independent effects required (default 10). At least 5 effects per model coefficient are also required.
method	Between-study variance estimator passed to ‘metafor‘ (default ‘"REML"‘).

Value

A fitted random-effects meta-regression model of class ‘c("rma.uni", "rma")‘ from ‘metafor::rma.uni()‘, with the study identifiers as labels. The coefficients describe the change in the mean outcome associated with each moderator.

Examples

```
if (requireNamespace("metafor", quietly = TRUE)) {
  toy <- data.frame(
    Study_ID = paste0("S", 1:12),
    Comparability_Class = "CLASS_A",
    Reported_Unit = "items/L",
    Metric_Type = "Abundance",
    Strict_Synthesis_Eligibility = "YES",
    Group = rep(c("g1", "g2"), 6),
    Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
    SD = rep(c(0.6, 0.9), 6),
    N = rep(c(10, 12, 8), 4)
  )
  imed_meta_regression(toy, moderators = "Group")
}
```

---

imed_missingness *Missingness profile for a table*

Description

Missingness profile for a table

Usage

```
imed_missingness(x, table = "MEASUREMENTS")
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
table	Name of the table to profile (default ‘"MEASUREMENTS"‘).

Value

A ‘tibble‘ with one row per field and the columns ‘field‘, ‘n‘ (number of rows), ‘missing‘ (number of missing or not-reported values) and ‘missing_pct‘ (the percentage missing).

Examples

```
db <- imed_data(tables = "MEASUREMENTS", validate = FALSE)
head(imed_missingness(db, "MEASUREMENTS"))
```

---

imed_plot *Publication-oriented abundance/distribution figure*

Description

Publication-oriented abundance/distribution figure

Usage

```
imed_plot(
  x,
  xvar = NULL,
  value = "Mean",
  type = c("box", "violin", "scatter", "histogram"),
  guard = TRUE
)
```

Arguments

x	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
xvar	Field to place on the x axis (unquoted or quoted name). Required for the box, violin and scatter types.
value	Name of the numeric measurement field to plot (default ‘"Mean"‘).
type	Plot type: ‘"box"‘, ‘"violin"‘, ‘"scatter"‘ or ‘"histogram"‘.
guard	If ‘TRUE‘ (default), stop when the records fail the comparability check.

Value

A ‘ggplot‘ object that can be printed, modified with further ‘ggplot2‘ layers or saved with ‘imed_export_figure()‘.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_plot(toy, xvar = "Group", type = "box")
imed_plot(toy, type = "histogram")
```

---

imed_plot_composition *Plot polymer or particle-characteristic composition*

Description

Plot polymer or particle-characteristic composition

Usage

```
imed_plot_composition(
  x,
  what = c("polymers", "particles"),
  study_id = NULL,
  characteristic_type = NULL
)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
what	polymers or particles.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.
characteristic_type	Optional character vector of ‘Characteristic_Type‘ values; used only when ‘what = "particles"‘.

Value

A ‘ggplot‘ object showing the reported values by polymer or particle category as horizontal bars.

Examples

```
db <- imed_data(tables = "POLYMERS", validate = FALSE)
imed_plot_composition(db, what = "polymers")
```

imed_plot_methods *Plot method/QAQC reporting profile*

Description

Plot method/QAQC reporting profile

Usage

```
imed_plot_methods(x)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.

Value

A ‘ggplot‘ object showing, for each QA/QC item, the number of records with each reported status (including ‘"Missing/NR"‘).

Examples

```
db <- imed_data(tables = "METHODS_QAQC", validate = FALSE)
imed_plot_methods(db)
```

imed_provenance *Provenance records for studies or target records*

Description

Provenance records for studies or target records

Usage

```
imed_provenance(x, study_id = NULL, record_id = NULL)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.
study_id	Optional character vector of ‘Study_ID‘ values used to restrict the records.
record_id	Optional character vector of ‘Target_Record_ID‘ values to retrieve.

Value

A ‘tibble‘ with the provenance records (the PROVENANCE table), restricted to the requested studies or target records when these are supplied.

Examples

```
db <- imed_data(tables = "PROVENANCE", validate = FALSE)
head(imed_provenance(db))
```

`imed_results_table` *Tidy a statistical result object for manuscript reporting*

Description

Tidy a statistical result object for manuscript reporting

Usage

```
imed_results_table(x)
```

Arguments

x	A result object: an ‘imed_stats‘ result or a fitted ‘metafor‘ ‘rma‘ object.

Value

A ‘tibble‘. For an ‘imed_stats‘ object, the data-frame components of the result are stacked and labelled in a ‘section‘ column. For a ‘metafor‘ ‘rma‘ object, a single row with the pooled ‘estimate‘, its standard error ‘se‘, the confidence limits ‘ci_lb‘ and ‘ci_ub‘, the test statistic ‘z‘, ‘p_value‘, the heterogeneity measures ‘tau2‘, ‘I2‘ and ‘H2‘, and the number of effects ‘k‘. Other objects cause an error.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
res <- imed_stats(toy, group = "Group", test = "welch")
imed_results_table(res)
```

`imed_sensitivity` *Sensitivity summary by evidence restriction*

Description

Compares all eligible records with a stricter subset when evidence fields are available. This is descriptive sensitivity analysis, not causal proof.

Usage

```
imed_sensitivity(x, value = "Mean")
```

Arguments

`x`	An ‘imed_db‘, or a measurement-level data frame such as the output of ‘imed_analysis_set()‘. The comparability guard is applied, so the records must belong to one compatible comparability class.
`value`	Name of the numeric measurement field to summarise (default ‘"Mean"‘).

Value

A ‘tibble‘ with columns ‘stratum‘, ‘n‘ (number of finite values), ‘mean‘ and ‘median‘. The first row summarises all compatible records; a second row for strict-synthesis-eligible records is added when the field ‘Strict_Synthesis_Eligibility‘ is present.

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_sensitivity(toy)
```

imed_stats *Conservative inferential statistics*

Description

Tests are explicit, never auto-selected from p-values or normality alone.

Usage

```
imed_stats(
  x,
  value = "Mean",
  group = NULL,
  test = c("none", "welch", "wilcoxon", "kruskal")
)
```

Arguments

x	An 'imed_db', or a measurement-level data frame such as the output of 'imed_analysis_set()'. The comparability guard is applied, so the records must belong to one compatible comparability class.
value	Name of the numeric measurement field to analyse (default '"Mean"').
group	Grouping field (unquoted or quoted name); required when 'test' is not '"none"'.
test	Test to run: '"none"' (default), '"welch"', '"wilcoxon"' or '"kruskal"'. The test is never chosen automatically.

Value

An object of class 'c("imed_stats", "list")' with the elements 'descriptive' (a 'tibble' from 'imed_describe()'), 'normality' (a 'tibble' with the Shapiro-Wilk test, present when 3 to 5000 finite values are available) and 'test' (a 'tibble' with 'method', 'statistic' and 'p_value', present when 'test' is not '"none"').

Examples

```
toy <- data.frame(
  Study_ID = paste0("S", 1:12),
  Comparability_Class = "CLASS_A",
  Reported_Unit = "items/L",
  Metric_Type = "Abundance",
  Strict_Synthesis_Eligibility = "YES",
  Group = rep(c("g1", "g2"), 6),
  Mean = c(2.1, 3.4, 1.8, 4.0, 2.9, 3.1, 2.4, 3.8, 1.9, 3.3, 2.7, 4.2),
  SD = rep(c(0.6, 0.9), 6),
  N = rep(c(10, 12, 8), 4)
)
imed_stats(toy, group = "Group", test = "welch")
```

imed_tables *List loaded IMED tables*

Description

List loaded IMED tables

Usage

```
imed_tables(x = imed_data(validate = FALSE))
```

Arguments

x	An ‘imed_db‘ object. Defaults to the bundled snapshot loaded without validation.

Value

A ‘tibble‘ with one row per table and the columns ‘table‘ (table name), ‘rows‘ (number of rows) and ‘columns‘ (number of columns).

Examples

```
db <- imed_data(tables = c("STUDIES", "SITES"), validate = FALSE)
imed_tables(db)
```

imed_tables_data *IMED v1.0 bundled analytical tables*

Description

The package snapshot contains the core scientific tables from the frozen IMED v1.0 release. Use ‘imed_data()‘ rather than accessing CSV files directly. The relational backbone is STUDIES -> SITES -> SAMPLES -> MEASUREMENTS. Supporting tables preserve size-method hierarchy, particle characteristics, polymers, biota, processing, QA/QC, derived metrics, risk assessments, source discrepancies, provenance, comparability and coordinate classes.

imed_validate *Validate IMED core scientific and relational integrity*

Description

Validate IMED core scientific and relational integrity

Usage

```
imed_validate(x, strict = FALSE)
```

Arguments

x	An imed_db.
strict	Stop on ERROR-level failures.

Value

A ‘tibble‘ with one row per check and the columns ‘check‘, ‘severity‘ (‘"ERROR"‘ or ‘"HOLD"‘), ‘passed‘ (logical), ‘n_issues‘ (integer) and ‘detail‘. With ‘strict = TRUE‘ the function stops if any ERROR-level check fails.

Examples

```
db <- imed_data(validate = FALSE)
v <- imed_validate(db)
table(v$severity, v$passed)
```

imed_validation_report *Validation report for manuscript/release audit*

Description

Returns database integrity, HOLD burden, missingness and comparability counts.

Usage

```
imed_validation_report(x)
```

Arguments

x	An ‘imed_db‘ object, as returned by ‘imed_data()‘.

Value

A named list with five elements: ‘integrity‘ (the ‘tibble‘ returned by ‘imed_validate()‘), ‘cardinalities‘ (table, row and column counts from ‘imed_tables()‘), ‘hold_summary‘ (a one-row ‘tibble‘ with the number of measurements, the number flagged as HOLD and the number eligible), ‘comparability‘ (counts of records by ‘Comparability_Class‘ and ‘Strict_Synthesis_Eligibility‘) and ‘measurement_missingness‘ (the ‘tibble‘ returned by ‘imed_missingness()‘ for the measurements table).

Examples

```
db <- imed_data(validate = FALSE)
rep <- imed_validation_report(db)
names(rep)
rep$hold_summary
```

`imed_version` *Package and targeted data-release version*

Description

Package and targeted data-release version

Usage

```
imed_version()
```

Value

A named character vector with two elements: ‘package‘ (the version of imedR) and ‘imed_release‘ (the version of the bundled IMED data release).

Examples

```
imed_version()
```

Index

imed_analysis_set, 2
imed_check_comparability, 4
imed_check_size_window, 5
imed_cite, 5
imed_compare, 6
imed_data, 7
imed_describe, 7
imed_enrich_measurements, 8
imed_evidence_profile, 9
imed_export_figure, 9
imed_filter, 10
imed_flag_holds, 11
imed_forest, 12
imed_funnel, 13
imed_leave_one_out, 14
imed_manuscript_table, 15
imed_map, 15
imed_meta_eligibility, 16
imed_meta_mean, 17
imed_meta_prevalence, 18
imed_meta_regression, 18
imed_missingness, 19
imed_plot, 20
imed_plot_composition, 21
imed_plot_methods, 22
imed_provenance, 22
imed_results_table, 23
imed_sensitivity, 24
imed_stats, 25
imed_tables, 26
imed_tables_data, 26
imed_validate, 27
imed_validation_report, 27
imed_version, 28