Package ‘mellio’

July 6, 2026

Title Polished, Editable Tables and Statistical Results

Version 1.0.1

Description Sends supported 'R' objects to the 'Mellio' web app and creates polished, editable statistical tables in 'R'. The 'mellio_open' interface handles common hypothesis tests, model objects, model comparisons, descriptive summaries, tabular data, plots, and image files. The 'melliotab' interface formats data frames, model summaries, correlation matrices, and side-by-side comparison tables with APA-style numeric formatting, confidence intervals, table notes, and optional significance markers. Manual table helpers can copy or save 'melliotab' output as 'HTML', 'LaTeX', or 'Markdown' when file-based handoff is needed. Payloads include package-version metadata to support reproducible reporting and software citation.

URL `https://www.mellioapp.com`, `https://github.com/NicoMel1907/mellio-r`

BugReports `https://github.com/NicoMel1907/mellio-r/issues`

Depends R (>= 4.1.0)

Imports gt (>= 0.10.0), jsonlite, methods, broom (>= 1.0.0), rlang (>= 1.0.0), cli (>= 3.6.0)

Suggests ggplot2 (>= 3.4.0), lattice, htmlwidgets, webshot2, ggExtra, plotly, psych, afex, effectsize, emmeans, multcomp, ordinal, MASS, forecast, geepack, randomForest, brms, rstanarm, car, mediation, lavaan (>= 0.6.0), lmerTest, broom.mixed (>= 0.2.9), lme4, mgcv, nnet, nlme, performance, rstatix, survival, rstudioapi (>= 0.14), digest (>= 0.6.0), knitr (>= 1.40), rmarkdown (>= 2.20), testthat (>= 3.0.0), withr (>= 2.5.0)

Encoding UTF-8

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Melih Sahin [aut, cre]

Maintainer Melih Sahin <`nicomelpro@pm.me`>

Repository CRAN

Date/Publication 2026-07-06 13:10:02 UTC

Contents

`mellio-package` *mellio: Polished, Editable Tables and Statistical Results*

Description

mellio provides two main workflows:

Details

- `mellio_open()` sends supported R objects to the Mellio web app. Statistical results open as Result Cards, tabular data opens in the Tables workspace, and supported plots or image files open in Figures.
- `melliotab()` creates polished, editable tables directly in R for data frames, model summaries, correlation matrices, and model comparisons.

Common `mellio_open()` inputs

`mellio_open()` supports common `htest` objects such as `t.test()`, `cor.test()`, and `chisq.test()`; model objects such as `lm`, `glm`, `aov`, `lme4` mixed models, `lavaan` fits, survival models, ordinal models, and selected optional-package result classes; data frames, matrices, and base `table` objects; `ggplot2`, lattice, htmlwidget, recorded plot, and image-file inputs.

Common `melliotab()` workflow

Use `melliotab(x)` to create a table, then apply table modifiers such as `mt_title()`, `mt_note()`, `mt_decimals()`, `mt_format_ci()`, `mt_sig_stars()`, `mt_spanner()`, `mt_section_title()`, and `mt_simplify_headers()`. Pass two or more model objects to `melliotab(m1, m2, ...)` to create a side-by-side model comparison table. For file-based handoff, use `mt_copy()` or `mt_save()` to copy or save a finished table as HTML, LaTeX, or Markdown.

Quick reference

Function or option	What it does	Common values
`melliotab()`	Creates a formatted table in R	`style` = `"apa7"` or `"ieee"`
`mellio_open()`	Opens supported objects in Mellio	Models, tests, tables, data, plots
`style` / `mt_set_style()`	Sets or changes table style	`"apa7"`, `"ieee"`
`title` / `mt_title()`	Sets the table title	Text
`number` / `mt_number()`	Sets the table number	Number or text
`note` / `mt_note()`	Adds a table note	Text
`source` / `mt_source()`	Adds source text	Text
`decimals`, `p_decimals` / `mt_decimals()`	Controls rounding	`decimals` = `2`, `p_decimals` = `3`
`mt_sig_stars()`	Adds significance stars to an existing table	`remove_p` = `TRUE` or `FALSE`
`mt_remove_leading_zeros()`	Controls leading zeros	`TRUE`, `FALSE`
`mt_diagonal()`	Formats correlation matrices	`mode` = `"dash"`; `triangle` = `"lower`
`mt_spanner()`	Adds a spanning column header	Label text and columns
`mt_section_title()`	Adds a section-title row	`before` = or `after` = a row number

`mt_indent()`	Indents selected rows	`rows =`; `level = 1`, `2`, or `3`
`mt_copy()` / `mt_save()`	Copies or saves a table	Clipboard, `.html`, `.tex`, `.md`

Privacy and provenance

By default, Mellio payloads include R/package-version metadata and data fingerprints where available. Local machine details such as user name, host name, working directory, git state, and script path are not included unless you opt in with `options(mellio.provenance = "full")`. Set `options(mellio.provenance = FALSE)` to omit provenance metadata.

Citation

Use `citation("mellio")` to get the package citation.

Author(s)

Maintainer: Melih Sahin `<nicomelpro@pm.me>`

See Also

`mellio_open()`, `melliotab()`, `mt_save()`, `mt_copy()`

`knit_print.melliotab` *Knit print method for R Markdown / Quarto*

Description

Automatically detects the output format and uses the appropriate backend.

Usage

```
knit_print.melliotab(x, ...)
```

Arguments

`x`	A melliotab object
`...`	Additional arguments

Value

knitr output

---

melliotab.mediate *Create a formatted statistical table*

Description

The main entry point for melliotab. Accepts data frames, model objects, or correlation matrices and formats them according to the specified citation style. Passing two or more model objects creates a side-by-side model comparison table.

Usage

```
## S3 method for class 'mediate'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'summary.mediate'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'medSummary'
melliotab(x, ..., section = NULL)

## S3 method for class 'medSummary2'
melliotab(x, ..., section = NULL)

## S3 method for class 'modmedSummary'
melliotab(x, ..., section = NULL)

## S3 method for class 'modmedSummary2'
```

```
melliotab(x, ..., section = NULL)

melliotab(x, ...)

## Default S3 method:
melliotab(x, ..., section = NULL)

## S3 method for class '`NULL`'
melliotab(x, ...)

## S3 method for class 'character'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'summaryDefault'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'table'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)
```

```
## S3 method for class 'data.frame'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'lm'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  conf.int = TRUE,
  conf.level = 0.95,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'glm'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  conf.int = TRUE,
  exponentiate = FALSE,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'aov'
melliotab(
  x,
  style = "apa7",
```

```
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  effect_size = TRUE,
  ...
)

## S3 method for class 'htest'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'matrix'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  decimals = 2L,
  p_decimals = 3L,
  diagonal = "dash",
  triangle = "all",
  ...
)

## S3 method for class 'lavaan'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  what = NULL,
  section = NULL,
```

```
  standardized = TRUE,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'FitDiff'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  section = NULL,
  what = NULL,
  decimals = 2L,
  p_decimals = 3L,
  ...
)

## S3 method for class 'fa'
melliotab(
  x,
  style = "apa7",
  title = NULL,
  number = NULL,
  note = NULL,
  source = NULL,
  section = NULL,
  what = NULL,
  cut = 0,
  sort = FALSE,
  decimals = 2L,
  p_decimals = 3L,
  ...
)
```

Arguments

`x`	A data.frame, model object (lm, glm, aov, htest), or correlation matrix
`style`	Citation style: "apa7" or "ieee"
`title`	Table title
`number`	Table number (integer or character)
`note`	Table note text
`source`	Table source text
`decimals`	Decimal places for estimates/statistics (1-4)

`p_decimals`	Decimal places for p-values (2-4)
`...`	Additional arguments passed to methods
`section`	Optional section selector for multi-section tables. Examples include `"fit"`, `"loadings"`, `"paths"`, `"covariances"`, `"defined"`, `"reliability"`, or `"modification_indices"` for structural model payloads; `"loadings"`, `"variance"`, or `"fit"` for EFA; and `"comparison"`, `"fit"`, or `"diff"` for FitDiff objects.
`conf.int`	Include confidence intervals (default TRUE)
`conf.level`	Confidence level (default 0.95)
`exponentiate`	Show exponentiated coefficients (odds ratios for logistic)
`effect_size`	Include effect size (eta-squared) for ANOVA
`diagonal`	Diagonal display: "dash" (em-dash), "one" (keep 1.00), "blank"
`triangle`	Which triangle to show: "all", "lower", "upper"
`what`	Backward-compatible alias for `section` in methods that previously used `what`. Prefer `section`.
`standardized`	Include standardized estimates (default TRUE)
`cut`	Minimum absolute loading to display (default 0, show all). Loadings below this threshold are shown as blank cells.
`sort`	Sort items by their primary loading (default FALSE)

Value

A melliotab object

Supported inputs

`melliotab()` is intended for table output inside R. It supports plain data frames, matrices, base `table` objects, correlation matrices, common model objects, hypothesis-test objects, and payloads created by `mellio_payload()`. For objects that can produce several tables, call `melliotab(x)` once to see the available `section` choices, then request one explicitly, for example `melliotab(fit, section = "loadings")`.

Common modifiers

The `mt_*()` helpers are optional table modifiers. They follow the same pattern as many R table packages: create a table once, then add formatting only where needed. Common helpers include `mt_title()`, `mt_note()`, `mt_decimals()`, `mt_format_ci()`, `mt_remove_leading_zeros()`, `mt_sig_stars()`, `mt_spanner()`, `mt_indent()`, and `mt_section_title()`.

Significance stars are never added by default. Use `mt_sig_stars()` only when that convention is appropriate for your manuscript, course, or journal.

Examples

```
# From a data frame
df <- data.frame(
  Variable = c("Age", "Gender"),
  B = c(0.45, -1.23),
```

```
  SE = c(0.12, 0.34),
  t = c(3.75, -3.62),
  p = c(0.0003, 0.0004)
)
melliotab(df, style = "apa7", title = "Regression Results")

# From a linear model
model <- lm(Ozone ~ Temp + Wind, data = airquality)
melliotab(model, style = "apa7", title = "Predictors of ozone concentration")

# Compare multiple models side by side
m1 <- lm(Ozone ~ Temp, data = airquality)
m2 <- lm(Ozone ~ Temp + Wind, data = airquality)
m3 <- lm(Ozone ~ Temp + Wind + Solar.R, data = airquality)
melliotab(
  m1, m2, m3,
  labels = c("Step 1", "Step 2", "Step 3"),
  dep.var.labels = "Ozone concentration"
)
```

`mellio_addin_send` *RStudio addin — Send the selected R expression to Mellio*

Description

Reads the active selection in the RStudio editor, evaluates it in the global environment, and opens the result as a Result Card in the Mellio web app. The selected text is preserved as the displayed call so the saved card still says e.g. "t.test(score ~ group, data = df)" rather than the evaluated variable.

Usage

```
mellio_addin_send()
```

Details

Registered via `inst/rstudio/addins.dcf`. Appears in RStudio's **Addins** menu once the package is loaded.

Value

Invisibly `NULL`. Side effect: opens the browser at Mellio.

See Also

Other R bridge: `mellio_capture()`, `mellio_compare()`, `mellio_open()`, `mellio_payload()`, `mellio_to_json()`, `print.mellio_payload()`

Examples

```
if (interactive() &&
    requireNamespace("rstudioapi", quietly = TRUE) &&
    rstudioapi::isAvailable()) {
  # Highlight   t.test(extra ~ group, data = sleep)   in RStudio
  # Addins menu -> Send to Mellio
  mellio_addin_send()
}
```

`mellio_anova_pairwise` *Recompute one-way ANOVA pairwise contrasts under a different correction*

Description

Lets the Lab switch the multiple-comparison adjustment without refitting the model: it reuses an aov object already held in the (WebR) session and runs the contrasts straight through `mellio_payload()`'s emmGrid method.

Usage

```
mellio_anova_pairwise(fit, by = NULL, adjust = "tukey")
```

Arguments

`fit`	A fitted one-way `aov`/`lm` object.
`by`	Optional grouping factor name; inferred from the model when `NULL`.
`adjust`	Multiple-comparison adjustment passed to `emmeans::contrast()` (e.g. `"tukey"`, `"bonferroni"`, `"holm"`, `"sidak"`, `"fdr"`, `"none"`).

Value

A pairwise-comparisons payload (table fields + `pairwise_forest`).

`mellio_capture` *Capture the current graphics device and open it in Mellio*

Description

Convenience wrapper for base R plots. Call after producing a base plot with `plot()`, `hist()`, `barplot()`, or another graphics function.

Usage

```
mellio_capture(...)
```

Arguments

`...`	Arguments passed to `mellio_open()`, such as `title`, `number`, `note`, and `browse`.

Value

Invisibly, the Mellio URL string.

See Also

Other R bridge: `mellio_addin_send()`, `mellio_compare()`, `mellio_open()`, `mellio_payload()`, `mellio_to_json()`, `print.mellio_payload()`

Examples

```
if (interactive()) {
  plot(mtcars$wt, mtcars$mpg)
  mellio_capture(
    title = "Fuel efficiency by weight",
    number = 1,
    browse = FALSE
  )
}
```

`mellio_compare` *Compare nested regression models as a Mellio Stats card*

Description

Builds a hierarchical-regression comparison payload from two or more nested model objects. Unlike `mellio_payload(anova(m1, m2))`, this keeps the original models available so Mellio can report model `R^2`, adjusted `R^2`, `Delta R^2`, and the nested-model F-change.

Usage

```
mellio_compare(..., labels = NULL, .call = NULL)
```

Arguments

`...`	Nested model objects, usually `lm` objects ordered from the baseline model to the final model.
`labels`	Optional model labels. Defaults to `"Model 1"`, `"Model 2"`, and so on.
`.call`	Optional call string for provenance / display.

Value

A `mellio_payload` object.

See Also

Other R bridge: `mellio_addin_send()`, `mellio_capture()`, `mellio_open()`, `mellio_payload()`, `mellio_to_json()`, `print.mellio_payload()`

Examples

```
m1 <- lm(mpg ~ wt + hp, data = mtcars)
m2 <- lm(mpg ~ wt + hp + cyl, data = mtcars)
p <- mellio_compare(m1, m2)
p$type
```

`mellio_open` *Open a table, statistical result, or plot in Mellio*

Description

Unified entry point for sending R objects to the Mellio web app. Statistical results and tables open as Stats Result Cards; plots open in the web app's Figures editor.

Usage

```
mellio_open(x, ..., browse = TRUE, .call = NULL)
```

Arguments

`x`	An R object supported by Mellio.
`...`	Additional arguments forwarded to the relevant Mellio constructor or payload method.
`browse`	Open the browser? `TRUE` by default. Set `FALSE` to return the URL without launching anything.
`.call`	Internal captured user call for provenance.

Value

Invisibly, the URL string.

Supported inputs

`mellio_open()` routes objects by class:

- Statistical results such as `htest`, `lm`, `glm`, `aov`, `lavaan`, mixed-model, survival, ordinal, mediation, and selected optional-package result classes open as structured Result Cards.
- Data frames, matrices, base `table` objects, and non-statistical `melliotab` objects open in the Tables workspace.
- `ggplot2`, lattice, htmlwidget, recorded plot, and supported image-file inputs open in Figures.

Optional model and figure integrations use packages listed in `Suggests`. If an optional package is not installed, the corresponding object class is skipped or reported with an informative message.

Privacy and destination

`mellio_open()` opens `https://www.mellioapp.com` by default. Advanced users can set `options(mellio.editor_url = "https://...")`, but should only use a trusted Mellio deployment. The R payload is encoded in the URL fragment. URL fragments are not sent as HTTP requests to the server, but the full URL can still be visible to the browser, the opened web app, browser history, extensions, and anyone the URL is shared with.

By default, Mellio payloads include R/package-version metadata and data fingerprints where available. Local machine details are opt-in: `options(mellio.provenance = "full")`. To omit provenance metadata, use `options(mellio.provenance = FALSE)`.

Plot size limits

Plots are transported through the URL hash. Very large images may exceed browser URL limits; reduce `width`, `height`, or `dpi`, or save the image and upload it manually in Mellio.

See Also

Other R bridge: `mellio_addin_send()`, `mellio_capture()`, `mellio_compare()`, `mellio_payload()`, `mellio_to_json()`, `print.mellio_payload()`

Examples

```
if (interactive()) {
  mellio_open(t.test(extra ~ group, data = sleep))

  m1 <- lm(Ozone ~ Temp, data = airquality)
  m2 <- lm(Ozone ~ Temp + Wind, data = airquality)
  m3 <- lm(Ozone ~ Temp + Wind + Solar.R, data = airquality)
  mellio_open(
    m1, m2, m3,
    labels = c("Step 1", "Step 2", "Step 3"),
    dep.var.labels = "Ozone concentration"
  )
}
```

`mellio_payload` *Build a Mellio Result Card payload from an R object*

Description

Converts a supported R model or test object into a structured JSON payload (a "Result Card") that Mellio renders in the Stats section. Pure and offline — no network calls. Use `mellio_open()` to additionally open the payload in the Mellio web app.

Usage

```
## S3 method for class 'aov'
mellio_payload(x, focal = NULL, controls = NULL, ..., .call = NULL)

## S3 method for class 'summary.aov'
mellio_payload(
  x,
  focal = NULL,
  controls = NULL,
  ...,
  .call = NULL,
  .env = parent.frame()
)

## S3 method for class 'aovlist'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'afex_aov'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'anova'
mellio_payload(x, focal = NULL, controls = NULL, ..., .call = NULL)

## S3 method for class 'brmsfit'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'stanreg'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'glm'
mellio_payload(
  x,
  ...,
  .call = NULL,
  focal = NULL,
  controls = NULL,
  exponentiate = FALSE,
  conf.int = TRUE,
  conf.level = 0.95
)

## S3 method for class 'rlm'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'gam'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'lme'
```

```
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'glmmTMB'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'kmeans'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'hclust'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'randomForest'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'coxph'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'numeric'
mellio_payload(x, name = NULL, ..., .call = NULL)

## S3 method for class 'summaryDefault'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'fa'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'principal'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'factanal'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'prcomp'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'effectsize_table'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'effectsize_difference'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'effectsize_anova'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'see_effectsize_table'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'FitDiff'
```

```
mellio_payload(x, what = NULL, ..., .call = NULL)

## S3 method for class 'geeglm'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'glmerMod'
mellio_payload(
  x,
  ...,
  .call = NULL,
  exponentiate = FALSE,
  conf.int = TRUE,
  conf.level = 0.95
)

## S3 method for class 'htest'
mellio_payload(x, .data = NULL, ..., .call = NULL, .env = parent.frame())

## S3 method for class 'lavaan'
mellio_payload(
  x,
  focal = NULL,
  diagram_omit = NULL,
  standardized = TRUE,
  ...,
  .call = NULL
)

## S3 method for class 'lm'
mellio_payload(x, focal = NULL, controls = NULL, ..., .call = NULL)

## S3 method for class 'lmerMod'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'manova'
mellio_payload(
  x,
  tests = c("Pillai", "Wilks", "Hotelling-Lawley", "Roy"),
  ...,
  .call = NULL
)

## S3 method for class 'summary.manova'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'mediate'
mellio_payload(x, ..., .call = NULL)
```

```
## S3 method for class 'summary.mediate'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'multinom'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'polr'
mellio_payload(x, ..., .call = NULL, conf.int = TRUE, conf.level = 0.95)

## S3 method for class 'clm'
mellio_payload(x, ..., .call = NULL, conf.int = TRUE, conf.level = 0.95)

## S3 method for class 'TukeyHSD'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'pairwise.htest'
mellio_payload(x, ..., .call = NULL, .env = parent.frame())

## S3 method for class 'dunn_test'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'wilcox_test'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 't_test'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'dunnTest'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'dunn.test'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'emmGrid'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'emm_list'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'glht'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'summary.glht'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'medSummary'
mellio_payload(x, ..., .call = NULL)
```

```
## S3 method for class 'medSummary2'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'modmedSummary'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'modmedSummary2'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'alpha'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'corr.test'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'character'
mellio_payload(x, title = NULL, ..., .call = NULL)

## S3 method for class 'data.frame'
mellio_payload(x, title = NULL, ..., .call = NULL)

## S3 method for class 'survfit'
mellio_payload(x, ..., .call = NULL, .env = parent.frame())

## S3 method for class 'Arima'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'forecast_ARIMA'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'ARIMA'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'ets'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'forecast'
mellio_payload(x, ..., .call = NULL)

mellio_payload(x, ...)

## S3 method for class 'mellio_payload'
mellio_payload(x, ..., .call = NULL)

## Default S3 method:
mellio_payload(x, ..., .call = NULL)

## S3 method for class '`NULL`'
```

```
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'table'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'matrix'
mellio_payload(x, ..., .call = NULL)

## S3 method for class 'melliotab'
mellio_payload(x, ..., .call = NULL)
```

Arguments

`x`	A supported R object.
`focal`	Optional character vector of model terms to mark as the focal predictor(s). Recognised by lm/aov/anova methods.
`controls`	Optional character vector of model terms to mark as controls (vs focal predictors) — recognised by lm/aov/anova methods.
`...`	Additional arguments passed to the dispatched method.
`.call`	Internal — captured user call, used to attribute provenance to the calling script rather than this dispatch.
`.env`	Optional environment used when recovering source variables for supported test objects.
`exponentiate`	Logical. For GLM-style models, exponentiate coefficient estimates (for example, odds ratios in logistic models).
`conf.int`	Logical. Include coefficient confidence intervals when supported by the broom method.
`conf.level`	Confidence level for coefficient intervals.
`name`	Optional name to attach to the payload (used by the numeric/vector descriptive method).
`what`	Optional FitDiff table to extract: `"comparison"`, `"fit"`, or `"diff"`. Defaults to the first available table.
`.data`	Optional data.frame to enrich supported htest results with observed descriptives, effect sizes, and figure data. When omitted, Mellio tries to recover simple source variables from the calling environment before falling back to test-only output.
`diagram_omit`	Optional character vector of parameter keys or variable names to omit from the generated SEM path diagram while retaining them in diagram metadata. Use exact path keys like `"y ~ ideology"` or a variable name like `"ideology"` for hidden covariate-style paths.
`standardized`	Include standardized estimates and standardized-solution metadata when available.
`tests`	Multivariate test statistics to extract (manova method). Defaults to all four classical tests.
`title`	Optional title override (psych/character/data.frame methods).

Details

Supported in v0: objects of class `htest` (`t.test`, `cor.test`, `wilcox.test`, `chisq.test`, `fisher.test`).

Schema: see `docs/STATS-R-BRIDGE-SCHEMA.md` in the repo. The payload is value-only: numeric statistics, p-values, and CIs go through unformatted; the Mellio web app applies citation-style rules (decimals, italics, leading zeros).

Value

A list with class `mellio_payload`, structured according to the v0.1 schema.

See Also

`mellio_open()` to open the result in Mellio. `mellio_to_json()` to serialise the payload offline.

Other R bridge: `mellio_addin_send()`, `mellio_capture()`, `mellio_compare()`, `mellio_open()`, `mellio_to_json()`, `print.mellio_payload()`

Examples

```
tt <- t.test(extra ~ group, data = sleep)
p <- mellio_payload(tt)
p$type
p$fields$statistic
```

`mellio_to_json` *Serialise a Mellio payload to JSON*

Description

Serialise a Mellio payload to JSON

Usage

```
mellio_to_json(payload, pretty = FALSE)
```

Arguments

`payload`	A `mellio_payload` object from `mellio_payload()`.
`pretty`	Pretty-print with line breaks (default `FALSE`).

Value

A length-1 character vector containing the JSON.

See Also

`mellio_payload()` to build the payload, `mellio_open()` to send it directly to the Mellio web app.

Other R bridge: `mellio_addin_send()`, `mellio_capture()`, `mellio_compare()`, `mellio_open()`, `mellio_payload()`, `print.mellio_payload()`

Examples

```
p <- mellio_payload(t.test(extra ~ group, data = sleep))
cat(mellio_to_json(p, pretty = TRUE))
```

`mt_as_gt` *Convert a melliotab object to a gt table*

Description

Convert a melliotab object to a gt table

Usage

```
mt_as_gt(x)
```

Arguments

`x`	A melliotab object

Value

A gt table object

`mt_as_html` *Get HTML string from a melliotab object*

Description

Produces a standalone HTML fragment with the title rendered above the table (outside the gt borders), matching the viewer layout.

Usage

```
mt_as_html(x)
```

Arguments

`x`	A melliotab object

Value

Character string of HTML

`mt_as_latex` *Convert a melliotab object to LaTeX*

Description

Convert a melliotab object to LaTeX

Usage

```
mt_as_latex(x)
```

Arguments

`x`	A melliotab object

Value

Character string of LaTeX code

`mt_as_markdown` *Convert a melliotab object to Markdown*

Description

Convert a melliotab object to Markdown

Usage

```
mt_as_markdown(x)
```

Arguments

`x`	A melliotab object

Value

Character string of Markdown table

`mt_copy` *Copy a melliotab table to the system clipboard*

Description

Copies the formatted table to the clipboard as rich HTML so it can be pasted into Word with formatting preserved.

Usage

```
mt_copy(x)
```

Arguments

`x`	A melliotab object

Value

Invisible melliotab object (for piping)

Examples

```
if (interactive()) {
  model <- lm(mpg ~ wt + hp, data = mtcars)
  tab <- melliotab(model, title = "Regression Results")
  mt_copy(tab)
  # Now paste into Word
}
```

`mt_decimals` *Set decimal places*

Description

Set decimal places

Usage

```
mt_decimals(x, decimals = 2L, p_decimals = 3L)
```

Arguments

`x`	A melliotab object
`decimals`	Decimal places for estimates/statistics
`p_decimals`	Decimal places for p-values

Value

Modified melliotab object

`mt_diagonal` *Set correlation matrix display options*

Description

Set correlation matrix display options

Usage

```
mt_diagonal(
  x,
  mode = c("dash", "one", "blank"),
  triangle = c("all", "lower", "upper")
)
```

Arguments

`x`	A melliotab object
`mode`	Diagonal display: "dash", "one", "blank"
`triangle`	Triangle to show: "all", "lower", "upper"

Value

Modified melliotab object

`mt_format_ci` *Format confidence intervals*

Description

Format confidence intervals

Usage

```
mt_format_ci(x, decimals = NULL, bracket = TRUE)
```

Arguments

`x`	A melliotab object
`decimals`	Decimal places for CI values (NULL uses table default)
`bracket`	Whether to use bracket notation

Value

Modified melliotab object

`mt_indent` *Indent row labels*

Description

Indent row labels

Usage

```
mt_indent(x, rows, level = 1L)
```

Arguments

`x`	A melliotab object
`rows`	Row indices to indent
`level`	Indentation level (1-3)

Value

Modified melliotab object

`mt_note` *Set table note*

Description

Set table note

Usage

```
mt_note(x, note)
```

Arguments

`x`	A melliotab object
`note`	Note text

Value

Modified melliotab object

`mt_number` *Set table number*

Description

Set table number

Usage

```
mt_number(x, number)
```

Arguments

`x`	A melliotab object
`number`	Table number

Value

Modified melliotab object

`mt_remove_leading_zeros` *Control leading zero removal*

Description

Control leading zero removal

Usage

```
mt_remove_leading_zeros(x, enabled = TRUE)
```

Arguments

`x`	A melliotab object
`enabled`	Whether to remove leading zeros

Value

Modified melliotab object

`mt_save` *Save a melliotab table to file*

Description

Auto-detects the format from the file extension.

Usage

```
mt_save(x, filename, ...)
```

Arguments

`x`	A melliotab object
`filename`	Output file path. Supported extensions: .html, .tex, .md
`...`	Additional arguments passed to format-specific functions

Value

Invisible file path

Examples

```
model <- lm(mpg ~ wt + hp, data = mtcars)
tbl <- melliotab(model, style = "apa7", title = "Results")

html_file <- tempfile(fileext = ".html")
tex_file <- tempfile(fileext = ".tex")
md_file <- tempfile(fileext = ".md")

mt_save(tbl, html_file)
mt_save(tbl, tex_file)
mt_save(tbl, md_file)
```

`mt_section_title` *Add a section title row*

Description

Add a section title row

Usage

```
mt_section_title(x, label, before = NULL, after = NULL)
```

Arguments

`x`	A melliotab object
`label`	Section title text
`before`	Row index to insert before
`after`	Row index to insert after

Value

Modified melliotab object

`mt_set_style` *Change the citation style*

Description

Change the citation style

Usage

```
mt_set_style(x, style = c("apa7", "ieee"))
```

Arguments

`x`	A melliotab object
`style`	Style name: "apa7" or "ieee"

Value

Modified melliotab object

`mt_sig_stars` *Add significance stars*

Description

Add significance stars

Usage

```
mt_sig_stars(
  x,
  target = "auto",
  remove_p = TRUE,
  levels = c(`*` = 0.05, `**` = 0.01, `***` = 0.001)
)
```

Arguments

`x`	A melliotab object
`target`	Column to add stars to ("auto" to detect)
`remove_p`	Whether to remove the p-value column
`levels`	Named vector of significance thresholds

Value

Modified melliotab object

`mt_simplify_headers` *Simplify SPSS verbose headers*

Description

Simplify SPSS verbose headers

Usage

```
mt_simplify_headers(x)
```

Arguments

`x`	A melliotab object

Value

Modified melliotab object

`mt_source` *Set table source*

Description

Set table source

Usage

```
mt_source(x, source)
```

Arguments

`x`	A melliotab object
`source`	Source text

Value

Modified melliotab object

`mt_spanner` *Add a spanning column header*

Description

Add a spanning column header

Usage

```
mt_spanner(x, label, columns, level = 1L)
```

Arguments

`x`	A melliotab object
`label`	Spanner label text
`columns`	Column indices or names covered by the spanner
`level`	Spanner level (1 = closest to data)

Value

Modified melliotab object

`mt_title` *Set table title*

Description

Set table title

Usage

```
mt_title(x, title)
```

Arguments

`x`	A melliotab object
`title`	Title text

Value

Modified melliotab object

`print.melliotab` *Print a melliotab object*

Description

Renders the table in the RStudio Viewer pane (or browser).

Usage

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

Arguments

`x`	A melliotab object
`...`	Ignored

Value

Invisible melliotab object

`print.mellio_payload` *Print method for Mellio payloads*

Description

Compact summary of a `mellio_payload` object. Use `mellio_to_json()` to see the full structure or `str()` to inspect the field tree.

Usage

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

Arguments

`x`	A `mellio_payload` object.
`...`	Unused.

Value

Invisibly returns `x`.

See Also

Other R bridge: `mellio_addin_send()`, `mellio_capture()`, `mellio_compare()`, `mellio_open()`, `mellio_payload()`, `mellio_to_json()`

Index