Package: hvtiRtables (via r-universe)

October 23, 2026

Title Manuscript-Compliant Table Construction for HVTI CORR Group

Version 1.0.11

Description Turns 'gtsummary' table objects into 'MS Word' tables that follow the ``Table Construction for Manuscripts'' rules of the HVTI (Heart, Vascular and Thoracic Institute) CORR (Cardiovascular Outcomes, Registries and Research) group: footnotes and abbreviation keys as text below the table, no hidden spacer columns, and the house notation and rounding. Tables are drawn in a style named by its role: a report style, with flat headers, and a manuscript style, which lays a table out to compare groups or to describe cohorts side by side.

License GPL-3

URL https://github.com/ehrlinger/hvtiRtables, https://ehrlinger.github.io/hvtiRtables/

BugReports https://github.com/ehrlinger/hvtiRtables/issues

Depends R (>= 4.4.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Suggests broom (>= 1.0.8), dplyr, haven, knitr, quarto, rmarkdown, smd (>= 0.6.6), testthat (>= 3.1.7)

Config/testthat/edition 3

VignetteBuilder quarto

SystemRequirements Quarto command line tool (<https://github.com/quarto-dev/quarto-cli>)

Imports flextable (>= 0.9.0), gtsummary (>= 2.5.0), officer (>= 0.6.0), xml2 (>= 1.3.0)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libnode-dev

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

Date/Publication 2026-10-23 18:50:59 UTC

RemoteUrl https://github.com/ehrlinger/hvtiRtables

RemoteRef HEAD

RemoteSha fead35f4625ae9a20b5c1264610e485373f3c50a

Contents

hv_check_docx 2
hv_correlation_table 4
hv_man_footnotes 6
hv_man_table 7
hv_man_table_display 10
hv_man_table_jtcvs 12
hv_man_table_save 14
hv_man_table_save_jtcvs 17
hv_table_style 19
hv_tbl_compare 20
hv_tbl_summary 21
hv_test_footnotes_jtcvs 26

Index **28**

`hv_check_docx` *Check a .docx for structural patterns the house rules forbid*

Description

Read-only inspection of any Word document, whether written by this package or received from elsewhere. Reports the three structural failure patterns the canonical "Table Construction for Manuscripts" memo names, a table too wide for its page, and markup left as plain text, and never modifies the file.

Usage

```
hv_check_docx(path, style = NULL)
```

Arguments

`path`	Path to a `.docx` file.
`style`	`NULL`, the default, applies the rule every style shares. A style name or `hv_table_style()` applies that style's rule: the report style merges no cells at all.

Details

Six detectors run:

- `"layer"` (structural, high confidence): paragraphs positioned with `w:framePr`, or floating text boxes (`w:txbxContent`). Table content belongs in the table, not in a floating layer over it.
- `"hidden_column"` (structural, high confidence): a table column that is both entirely empty **and** strictly narrower than 0.1 inch (a `w:gridCol` width below 144 dxa; exactly 144 does not count). Both conditions are required. The house's own JTCVS templates contain all-empty 0.5-inch gutter columns between group pairs, which are deliberate; width alone or emptiness alone would flag them. A cell spanning several columns counts as filling every column it covers, so a merged header never makes the columns beneath it look empty.
- `"wide_table"` (structural, high confidence): a table whose grid is wider than the text block of the page section it sits in (page width less both margins), by more than 1 pt. Word clips what runs past the margin, so the rightmost columns are lost from the page.
- `"embedded_footnote"` (heuristic, lower confidence): a cell in a table's last row whose text starts with a footnote marker (`*`, a dagger or double dagger, a section or pilcrow sign, a lettered `a.`/`b.` marker, `Key:` or `Abbreviation:`) or whose first run is a superscript number, as `gtsummary` writes a footnote, the pattern `flextable::footnote()` produces when misused. This detector is pattern-based rather than a guaranteed structural signature, so treat its findings as prompts to look, not proof.
- `"merged_cell"` (structural, high confidence): a row whose cells are merged (`w:gridSpan` above 1, `w:vMerge`, or the legacy `w:hMerge`) outside the header rows, unless the row is one cell spanning the table. The header rows are the leading run of rows whose `w:tblHeader` flag is on, as Word reads it. With `style = "report"`, any merged cell is a finding.
- `"markup"` (heuristic, lower confidence): a paragraph whose baseline text still holds `^...^` or `~...~` markup, such as `"BSA (m^2^)"`. Raised and lowered runs are skipped, as they are the renderer's own output. `hv_man_table()` and `hv_man_table_save()` render the markup as a superscript or subscript, so markup left in the text reached Word another way, and would print as typed. It is a text pattern, so a URL such as `http://host/~a/~b` also matches. Checked whatever `style` is given.

Value

A `data.frame` with one row per finding and columns `type`, `table` (1-indexed table number, `NA` when document-level), `location` (row/column detail where applicable, `NA` otherwise), and `detail` (human-readable description). Zero rows means clean.

Common mistakes

"`path` must be a single non-empty string." The argument is a file path, not a `flextable` or an `officer` document. The saver does not run this on its own output, so call it on the written `.docx` afterwards.

Reading a `hidden_column` finding as a gutter column. It is not one. The detector requires empty **and** narrower than 0.1 inch, precisely so the deliberate half-inch gutter columns in the house JTCVS templates do not trip it. A finding here is a real spacer.

Treating `embedded_footnote` or `markup` as proof. They are the two heuristic detectors of the six. `embedded_footnote` matches a marker character opening a cell in a table's last row, and a

legitimate last row that happens to start that way matches it too. `markup` matches a pair of ^ or ~ characters, and a URL with two tildes matches it. Read either as a prompt to look, not a verdict.

Expecting a repair. This is read-only and never modifies the file. Fix what it reports in Word, or rebuild the table through `hv_man_table_save()`.

See Also

`hv_man_table_save()`, which writes the files this checks. It does not run the check on its own output, so call this on the path afterwards when you want it.

Examples

```
tbl <- hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = "age"),
  continuous = "age", categorical = character(0), compare = "none"
)
ft <- hv_man_table(tbl, style = "manuscript", layout = "describe")
out <- tempfile(fileext = ".docx")
hv_man_table_save(ft, out, caption = "Table 1. X")
hv_check_docx(out)
unlink(out)
```

---

hv_correlation_table *Pairwise correlations with Fisher confidence intervals*

Description

The table half of the SAS correlation job `proc corr nosimple spearman pearson fisher(biasadj=no alpha=.32)`, with the `SpearmanCorr`, `FisherSpearmanCorr`, `PearsonCorr` and `FisherPearsonCorr` ODS tables folded into one data frame. The scatter-plot matrix (`plots=matrix`) is `hvtiPlotR::hv_correlation_matrix()`.

Usage

```
hv_correlation_table(
  data,
  vars,
  with = NULL,
  by = NULL,
  method = c("spearman", "pearson"),
  conf_level = 0.68,
  digits = 2
)
```

Arguments

`data`	A data frame.
`vars`	Character vector of numeric columns (`VAR` statement).
`with`	Character vector of numeric columns to correlate each of `vars` against (`WITH` statement). `NULL` (default) correlates every unordered pair within `vars`.
`by`	Optional single column name to stratify on (`BY` statement). Each level gets its own rows; the column is carried first in the output.
`method`	One or both of `"spearman"` and `"pearson"`.
`conf_level`	Confidence level for the Fisher interval, strictly between 0 and 1. Default `0.68`.
`digits`	Decimal places in `display`. Default `2`, the SAS `6.2` format.

Details

The interval is Fisher's z without bias adjustment, SAS's `biasadj=no`: `z = atanh(r)`, `se = 1 / sqrt(n - 3)`, back-transformed with `tanh()`. The same transform is applied to Spearman's coefficient, as SAS does. The p-value tests rho = 0 on the same z scale, so it is the Fisher p-value SAS reports alongside the interval, not the t-test p-value of `stats::cor.test()`. The interval, by contrast, is identical to `cor.test()`'s for Pearson.

`conf_level` defaults to 0.68, the exemplar's `alpha=.32`: the house one-standard-error convention, the same logic as the 15th and 85th percentiles in `hv_tbl_summary()`. Pass `0.95` for a conventional interval.

Missing values are deleted pairwise, per pair and per stratum, which is `proc corr`'s default. Rows with a missing `by` value are dropped, where SAS's `BY` would keep them as their own group.

Value

A data frame, one row per stratum, pair and method, with columns `by` (when given), `variable`, `label`, `with`, `method`, `n`, `estimate`, `conf.low`, `conf.high`, `p.value` and `display` (`"r (lcl, ucl)"`). `label` describes `variable` only, not the pair. `display` is `NA` whenever the interval is undefined: n <= 3, a zero-variance column, or an empty stratum. The level used is stored as `attr(x, "conf_level")`.

See Also

`hvtiPlotR::hv_correlation_matrix()` for the plot.

Examples

```
hv_correlation_table(
  datasets::mtcars, vars = c("hp", "wt", "qsec"), with = "mpg",
  by = "am"
)
```

hv_man_footnotes *The house-standard manuscript table footnotes*

Description

The house footnote block the macro emitted under PRINTFN=1.

Usage

```
hv_man_footnotes(
  continuous_stat = c("median", "mean", "both"),
  percentiles = c(15, 85),
  style = "report"
)
```

Arguments

`continuous_stat`
: One of "median" (default), "mean", or "both". Controls whether the dagger's continuous half reads as median and percentiles or mean +/- SD. Under "both" it reads as mean +/- SD, because hv_tbl_summary() puts the medians in columns of their own whose headers name them.

`percentiles`
: Numeric vector of length 2, the increasing whole-number percentile pair named by the dagger. Default c(15, 85). Used for continuous_stat = "median" and "both"; validated but not printed for "mean", so the arguments share hv_tbl_summary()'s contract.

`style`
: "report" (default) or "manuscript", or a style from hv_table_style(). The manuscript writes the notes in the words of the submissions: "Patients with non-missing data." and "Values are No. (%) or median [15th, 85th percentiles].".

Details

Two footnotes are universal to the HVTI CORR "Table Construction for Manuscripts" rules, not specific to any one study, so you don't have to re-type them at every hv_man_table_save() call site: a * for the non-missing-value count (house rule 8, exact wording required) and a † explaining the statistic in each group column, e.g. "No. (%) or Median [15th, 85th percentiles]." The categorical half is part of it because the cells print 12 (39) with no % sign (house rules 10/12), and the flat CORR header names only the group.

hv_man_table_save()'s footnotes parameter defaults to calling this function, so every table gets both automatically. A table built by hv_tbl_summary() and hv_man_table() carries its continuous_stat and percentiles choices through to the saver, so its default dagger follows what the table shows. Call this function directly with the same arguments when composing footnotes yourself. Override with ordinary list operations, no special sentinel values needed:

- Suppress both: footnotes = NULL
- Change one: modifyList(hv_man_footnotes(), list(...)) with `†` = "custom text"

- Add a study-specific one alongside: `c(hv_man_footnotes(), list(...))` with `` `||` = "extra note" ``. Avoid ‡, § and ¶ on a table from `hv_tbl_summary()`: ‡ marks Fisher exact p-values there, and § and ¶ mark the minority statistic when a table mixes mean and median.

Value

A named list with elements `` `*` `` and `` `†` ``, in the format `hv_man_table_save()`'s `footnotes` parameter expects.

Common mistakes

Passing the result to the JTCVS saver. These footnotes are CORR-shaped, keyed by symbol, and belong to `hv_man_table_save()`. `hv_man_table_save_jtcvs()` takes an unrelated type keyed by row and column, and rejects this one. Its equivalent, `hv_test_footnotes_jtcvs()`, is deprecated along with that saver: the describe layout of `hv_man_table(style = "manuscript")` names its tests in its own notes, which `hv_man_table_save()` writes.

Calling this helper with settings different from the table. The automatic `hv_tbl_summary()` -> `hv_man_table()` -> `hv_man_table_save()` path stays synchronized. If you call this helper yourself, pass the same `continuous_stat` and `percentiles` used to build the table, or override the dagger explicitly.

See Also

`hv_man_table_save()`

Examples

```
hv_man_footnotes()
hv_man_footnotes(continuous_stat = "mean")
hv_man_footnotes(continuous_stat = "both", percentiles = c(16, 84))
modifyList(hv_man_footnotes(), list(`†` = "custom text"))
```

`hv_man_table` *Convert a gtsummary table into a flextable matching HVTI CORR's table rules*

Description

Stage 2 of the `%summarytable` split: shapes the table. The macro controlled this with `STYLE=` and `PAGE=`. `style =` is the equivalent of `STYLE=`: tables are drawn in a style named by its role, `"report"` (the default, described below) or `"manuscript"`. `PAGE=` has none; pages are US Letter, turning landscape when the table is wide.

Usage

```
hv_man_table(
  tbl,
  font = "Times New Roman",
  font_size = 12,
  digits = 2,
  style = "report",
  layout = "compare",
  ...
)
```

Arguments

`tbl`	A `gtsummary` table object (must support `as_flex_table()`). A table from `hv_tbl_summary()` is split as described above; any other `gtsummary` object renders as-is.
`font`	Font family. Default `"Times New Roman"` (house rule). Any single non-empty string is accepted: `flextable` silently substitutes an unknown font name, so a typo would otherwise pass unnoticed, while a deliberate override is legitimate.
`font_size`	Font size in points. Default `12`; pass `11` for wide tables, per house rule 5. No other values are permitted.
`digits`	Kept for interface symmetry with future callers; currently unused (rounding is controlled upstream via `tbl_summary(digits = ...)`). Reserved so a future version can enforce sig-fig rounding centrally without a breaking signature change.
`style`	A style name, `"report"` (default) or `"manuscript"`, or a style from `hv_table_style()`. `"manuscript"` needs an `hv_tbl_summary()` table, refuses `font` and `font_size`, and in the compare layout refuses `continuous_stat = "both"`.
`layout`	The manuscript style's layout, chosen by what the table does: `"compare"` (default), one overall `n` column for groups it compares, or `"describe"`, a count beside each group, for cohorts or repeated measures it describes side by side. Either shows an Overall column and comparison columns when the table carries them. Only with `style = "manuscript"`.
`...`	Not used. Present so that `%summarytable` parameter names produce an error naming the argument to use instead.

Details

The `"manuscript"` style lays out a comparison table as the journal submissions did: a full grid, a two-row header with the grouping's name spanning its levels, one overall `n` column, shaded section rows that span the table, a bold header, and no font of its own, so the table takes the body font of the document it is saved into. It needs a table from `hv_tbl_summary()`, and `hv_man_table_save()` requires a `caption` for it. See `hv_table_style()` for the properties of each style, and the Table styles article, which draws one table in each style and layout beside the matrix of their properties and says which house rules each style keeps.

The rest of this page describes the `"report"` style.

You give this a gtsummary::tbl_summary() (or any gtsummary table object supporting gtsummary::as_flex_table()), and you get back a flextable that already complies with HVTI CORR's "Table Construction for Manuscripts" rules: a single, non-merged header row (no spanning parent cells over grouped columns), no merged row-group section-header cells, a left-justified label column, and Times New Roman at the house font size. Section rows are shaded across every cell, unmerged, in the style's fill (#CAEDFB by default), and the header is bold.

gtsummary::as_flex_table() already emits one header row per group with self-contained labels (e.g. "B\nN = 45"), not a merged spanning header. Any merged region is undone with flextable::merge_none(). That satisfies the "format the table as flat as possible ... simple non-merged column titles" rule.

The sections of an hv_tbl_summary() table (its groups names, such as "Demography") become bold header rows in the label column, each immediately above its first variable, rather than a column of their own.

Rounding, %-free percentage cells, and ±-without-spaces formatting are yours to control, via the statistic/digits arguments to gtsummary::tbl_summary(); see the package README for a worked example using gtsummary::style_sigfig() for 2-significant-figure rounding.

A manuscript table that describes cohorts or repeated measures side by side, rather than comparing groups, takes layout = "describe": each group spans its own count and statistic, so every group has its count. Everything else is the comparison layout's. The JTCVS functions, whose layout this is, are deprecated.

A table from hv_tbl_summary() works in every style and layout. That function writes each cell as "{N_nonmiss} ||| {stat}", and this one splits the two apart into a flat No. column immediately before the statistic it counts. Those are the same two values the describe layout puts under each group's spanning header, without the merge. House rule 8 wants that non-missing count, and hv_man_footnotes()'s * footnote describes it, so it is kept rather than discarded. A plain gtsummary::tbl_summary() table carries no such convention and passes through unchanged.

None of gtsummary's own footnotes reach the table: as_flex_table() writes them as a footer row and a raised digit inside the table, and house rule 13 puts notes below it. hv_tbl_summary() says what they said in the notes hv_man_table_save() writes: the tests behind the p-values in the dagger note, and CI in the key.

Superscripts and subscripts are written into the text as in Pandoc: a label "BSA (m^2^)" raises the 2 and "HbA~1c~" lowers the 1c, in labels, levels, the by levels and section names alike. A pair holds no whitespace, so "~5 years" and "x ^ y" print as typed. The notes, the abbreviation key and the caption hv_man_table_save() writes take the same markup.

Value

A flextable object, ready for hv_man_table_save(). In the report style it has a single header row and no merged cells; in the manuscript style a two-row header. Its hv_style attribute holds the style, which the saver and hv_man_table_display() read. Input from hv_tbl_summary() retains its hv_footnotes attribute so the saver's default dagger follows the table's continuous statistic.

Common mistakes

m^2 **prints with a caret.** A pair needs both markers, so an unclosed m^2 prints as typed and nothing flags it. Write m^2^. A pair runs to the next matching marker, so a^2+b^2 raises 2+b; write a^2^+b^2^.

Cells read `98 ||| 46 (32, 63)`**.** You are on a version before the split; upgrade. `|||` is the internal separator `hv_tbl_summary()` writes between the count and the statistic, never intended to reach a rendered table.

"`tbl` was not built with the `{N_nonmiss} ||| {stat}` **convention."** Only some cells carry the separator, which means `statistic` was set for part of the table. Splitting would leave the rest blank, so it is rejected. Either apply the convention to every variable or to none; `hv_tbl_summary()` does the former for you.

"`font_size` must be 11 or 12." House rule 5. Use `11` only for wide tables.

"`font` and `font_size` set the report style." You passed one with `style = "manuscript"`. That style sets no font, so the table takes the font of the document it is saved into. Drop the argument, or use `style = "report"`.

The compare layout refuses a `continuous_stat = "both"` **table.** That table has a column for the median and another for the mean and SD, which the compare layout's one statistic column per group cannot lay out. Build the table with `"median"` or `"mean"`, or draw it with `layout = "describe"` or the report style, which show both.

The manuscript style refuses a table not built by `hv_tbl_summary()`**.** The layout reads the overall count and the grouping's name that function records. Build the table with it.

"`layout` chooses between the manuscript style's layouts." You named a layout with the report style, which has one. Drop `layout`, or add `style = "manuscript"`.

See Also

`hv_man_table_save()` to write the result to a compliant `.docx` with footnotes and an abbreviation key. `hv_table_style()` for each style's properties.

Examples

```
library(gtsummary)
tbl <- tbl_summary(trial, by = trt, include = c(age, grade))
ft <- hv_man_table(tbl)
ft2 <- hv_man_table(
  hv_tbl_summary(
    trial, by = "trt", groups = list(A = c("age", "grade")),
    continuous = "age", categorical = "grade"
  ),
  style = "manuscript", layout = "describe"
)
```

`hv_man_table_display` *Show a CORR table with its Word notes, for HTML*

Description

Adds to a flextable from hv_man_table() what hv_man_table_save() writes around it in Word, so a report can show the same table in HTML: the footnote marks in the header (an asterisk on each No. column, a dagger on the statistic columns), the footnotes, and the abbreviation key. In Word the notes are paragraphs below the table; here they are footer lines, since HTML has no page to put them on.

Usage

```
hv_man_table_display(ft, footnotes = NULL, abbreviations = NULL)
```

Arguments

ft	A flextable, from hv_man_table().
footnotes	Optional named list, symbol -> footnote text, exactly as for hv_man_table_save(). Omit it for the table's own footnotes; NULL shows none. A note's text may carry ^2^ or ~2~ markup, as a label does.
abbreviations	Optional named character vector, abbreviation -> definition, exactly as for hv_man_table_save(): shown as one "Key:" line, sorted by abbreviation. A table showing a confidence interval adds CI unless you define it. A name or definition may carry ^2^ or ~2~ markup, as a label does.

Details

The notes follow the table's style: a manuscript table's notes begin with the plain symbol and use the submissions' wording, as they do in the saved file. A manuscript table sets no font, so in HTML it takes the font and size of the page around it, as in Word it takes the document's.

The one deliberate difference is the "Variable" column hv_tbl_summary()'s show_variable adds: the display keeps it, as the report view, while the saved table never has it.

The marks, the note text and the key come from the same internal code the saver uses, so the displayed table and the saved one cannot disagree. footnotes resolves the same way too: left out, it is the footnote block the table carries from hv_tbl_summary(), or hv_man_footnotes() for any other flextable. A double dagger hv_man_table() put on a Fisher exact p-value is already on its cell and is footnoted, not marked again in the header.

The result is for display only. Save the flextable you passed in, not the result: its notes are rows inside the table, which the Word layout forbids, and hv_man_table_save() refuses it.

Value

ft with the header marks and the footer lines added, for display.

Common mistakes

"ft is a display copy from hv_man_table_display() ..." You passed the result to hv_man_table_save(). Keep the flextable from hv_man_table() for saving and display a copy: hv_man_table_save(ft, file) and hv_man_table_display(ft).

"The table marks cells with ..." The table has a Fisher exact p-value marked with a double dagger, and `footnotes` leaves it unexplained or files another note under it. Omit `footnotes`, or add the entry the error names.

Different notes in the report and in Word. Pass the same `footnotes` and `abbreviations` to both calls.

See Also

`hv_man_table_save()`, which writes the same marks and notes to Word. `hv_man_footnotes()` for the default footnotes.

Examples

```
ft <- hv_man_table(hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = c("age", "grade")),
  continuous = "age", categorical = "grade", compare = "none"
))
hv_man_table_display(ft, abbreviations = c(SD = "standard deviation"))
```

---

hv_man_table_jtcvs — *Deprecated: Build a JTCVS-Format Manuscript Table with Merged Headers*

Description

Deprecated. Use `hv_man_table(tbl, style = "manuscript", layout = "describe")`, the manuscript style's describe layout, which this layout became. This function warns on every call and goes in the next release.

Usage

```
hv_man_table_jtcvs(
  tbl,
  groups,
  trailing = NULL,
  stat_label = NULL,
  font = "Times New Roman",
  font_size = 12,
  ...
)
```

Arguments

`tbl`	A `gtsummary` table object whose `statistic` argument used `"{N_nonmiss} ||| {<stat>}"` for every group column. `hv_tbl_summary()` applies this convention for you. A table without it is rejected, since its cells cannot be split into their N and statistic parts.

groups — Named character vector, stat_<k> column name in tbl$table_body -> spanning header label (include the group's N in the label text yourself, e.g. c(stat_1 = "Group A (n=60)")). Every name must be a column of tbl$table_body, and each may appear at most once; unknown names are rejected with the available ones listed, and a repeated name is rejected too.

trailing — Optional named character vector, existing tbl$table_body column name -> header label, one entry per trailing comparison column, laid out in the order given (e.g. c(std_diff = "Std. Diff.") or c(p_value = "P")). Each must already exist in tbl$table_body. A table from hv_tbl_summary(), with any columns hv_tbl_compare() added, records the right value as its hv_trailing attribute.

stat_label — Sub-header text under each group's statistic column. Default NULL takes it from the table: hv_tbl_summary() records the statistic it used (e.g. "No. (%) or Median [15th, 85th percentiles]"), so the header cannot disagree with the cells. A table without that record falls back to "No. (%) or Mean±SD"; pass the right text yourself for any other statistic.

font — Font family. Checked as a single non-empty string, as before, and ignored: the manuscript style sets no font, so the table takes the font of the document it is saved into.

font_size — Font size in points. Checked as before (11 or 12, the rule hv_man_table() enforces) and ignored, for the same reason as font.

... — Not used. Present so that %summarytable parameter names produce an error naming the argument to use instead.

Details

Stage 2 of the %summarytable split: shapes the table. The macro controlled this with STYLE= and PAGE=. This renderer is the journal's one layout and takes neither; the CORR styles are chosen by style = in hv_man_table(). Pages are US Letter.

Until it goes, it passes its arguments through to that layout: groups and trailing choose the same group, Overall and comparison columns, and stat_label names the statistic under each group. The table is drawn in the manuscript style, not the JTCVS look it drew before: a full grid, shaded section rows spanning the table, * and † notes, and no font of its own. font and font_size are checked and set nothing.

Value

A flextable in the manuscript style's describe layout, ready for hv_man_table_save().

Common mistakes

"tbl was not built with the {N_nonmiss} ||| {stat} convention." The table came from a plain gtsummary::tbl_summary() call. Build it with hv_tbl_summary(), which applies the convention automatically, or pass statistic = list(all_continuous() ~ "{N_nonmiss} ||| {mean} ({sd})"). Before this check existed, such a table rendered every cell blank.

"groups names must be columns in tbl$table_body." Group names are the stat_<k> columns gtsummary creates, one per level of the by variable – not the group labels. Two groups give stat_1 and stat_2.

"font_size must be 11 or 12." House rule 5. The value is checked as before and then ignored: the manuscript style sets no font size, so the table takes the size of the document it is saved into.

See Also

hv_man_table(style = "manuscript", layout = "describe"), the layout that replaces it, and hv_man_table(). hv_man_table_save_jtcvs() to write the result to a compliant .docx.

Examples

```
# Instead of hv_man_table_jtcvs() and hv_man_table_save_jtcvs():
tbl <- hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = c("age", "grade")),
  continuous = "age", categorical = "grade", compare = "none"
)
ft <- hv_man_table(tbl, style = "manuscript", layout = "describe")
out <- tempfile(fileext = ".docx")
hv_man_table_save(ft, out, caption = "Table 2. Baseline Characteristics")
unlink(out)
```

---

hv_man_table_save *Write an hv_man_table() table to a compliant .docx*

Description

Stage 3 of the %summarytable split: writes the document. Replaces the macro's RTFFILE=/PDFFILE= output block.

Usage

```
hv_man_table_save(
  ft,
  file,
  footnotes = NULL,
  abbreviations = NULL,
  caption = NULL,
  orientation = c("auto", "portrait", "landscape"),
  ...
)
```

Arguments

ft	A flextable object, typically from hv_man_table().
file	Output .docx path. The output directory (dirname(file)) must already exist; this function does not create it. (%summarytable RTFFILE=/PDFFILE= equivalent; output here is always .docx).

`footnotes` Optional named list, symbol -> footnote text. When omitted, uses the `hv_footnotes` attribute carried from `hv_tbl_summary()` through `hv_man_table()`, so the dagger follows that table's `continuous_stat` and `percentiles`. Other flextables fall back to `hv_man_footnotes()`'s median/15th-85th house defaults. Pass `NULL` to suppress both, or compose with `hv_man_footnotes()` to override or extend. A note's text may carry `^2^` or `~2~` markup, as a label does. Symbols must be drawn from `c("*", "†", "‡", "§", "¶", "||")`. † describes the statistic, so it is appended as a superscript reference mark to every statistic-column header (`stat_<k>`), and to the count column only when the table has no such column. On a table that compares its groups, † also marks the p-value and standardized-difference headers, because the note names the tests behind the p-values and says what the difference is. Every other symbol goes on the table's count-column header cell (a column named `n`, else the first `n_stat_<k>` column `hv_man_table()` creates when it splits an `hv_tbl_summary()` table, else the first column). In a manuscript table the dagger goes on the grouping's spanning cell and `*` on its `n` column. In the describe layout `*` goes on every group's `n` and the dagger on every statistic. Each text is rendered as its own paragraph below the table, in the order given. Every element must be named (unnamed or blank-named entries raise an error) and every text must be a single non-empty string (`NULL`, `NA`, a number, or a multi-element vector raise an error, since each writes a dangling marker); an empty list is a no-op, same as `NULL`. (`%summarytable ADDFN=` equivalent; `PRINTFN=1`'s house block is `hv_man_footnotes()`).

`abbreviations` Optional named character vector, `c(ABBR = "expansion", ...)`. A table with a confidence-interval column (`compare = "both"`) adds `CI` unless you define it yourself. Rendered as one `Key:` paragraph below any footnotes, sorted alphabetically by abbreviation, abbreviation italicized, pairs separated by `"; "` (house rule 14). Every element must be named (unnamed or blank-named entries raise an error); an empty or `NULL` vector is a no-op. A name or definition may carry `^2^` or `~2~` markup, as a label does.

`caption` The table's caption, written in bold on its own line above the table. A manuscript table requires one, so it is written with the table and not typed in later; a report table takes one when given and writes none otherwise. It may carry `^2^` or `~2~` markup, as a label does.

`orientation` The page the table is written on. `"auto"`, the default, turns it landscape when the table is too wide for portrait, as described above. `"portrait"` or `"landscape"` always wins; a table too wide for the chosen page has its columns narrowed in proportion to fit it.

`...` Not used. Present so that `%summarytable` parameter names produce an error naming the argument to use instead.

Details

You hand this a `flextable` (typically from `hv_man_table()`), and it writes a Word document with footnotes and an abbreviation key rendered as text below the table rather than embedded within it (house rules 13-14). `flextable::footnote()` cannot do this for you: it renders footnote text as an extra row inside the table's own `<w:tbl>` block (a "footer" table part), which is exactly the compliance violation this function exists to avoid. Instead, a superscript reference symbol is appended to the target header cell (a normal, compliant reference mark), and the footnote text itself is written

as a genuine document paragraph after the table via `officer::body_add_fpar()`. The table itself is inserted directly (no leading blank paragraph), which keeps vertical cell alignment controllable if you later reformat the surrounding document to 1.5- or double-spacing. House rule 2 concerns the *insertion point* in the destination document, not this function's output; see the package README for the paste-in workflow.

Every style is written on a US Letter page. A report table with three or more statistic columns, such as an Overall column and two groups, is written on a landscape page: with a `No.` column beside each, it does not fit a portrait one. A manuscript table turns only on width, since it has one overall `n` column: Overall, two groups and a P value come to 6.5 inches and stay portrait. Any table wider than a portrait page's text block (6.53 inches on US Letter) is written landscape. A table wider than landscape's (9.0 inches), as a `continuous_stat = "both"` table with an Overall column can be, has its columns narrowed in proportion to fit; the font size is not changed. Pass `orientation` to choose the page yourself: `"landscape"` turns even a narrow table, and `"portrait"` keeps a wide one upright with its columns narrowed to the portrait text block.

In every style the saver drops the "Variable" column `hv_tbl_summary()`'s `show_variable` adds for the report view before the page is fitted, so it never reaches Word and never widens the page. `hv_man_table_display()` keeps it, for the report.

Value

Invisibly, the `file` path.

Common mistakes

"A manuscript table needs `caption` ..." The table is in the manuscript style, which requires a caption so the title is written with the table. Pass `caption = "Table 1. ..."`; a report table takes one optionally.

"`footnotes` must be a named list (symbol -> footnote text) ..." You passed the deprecated JTCVS shape. The two savers' `footnotes` arguments are unrelated types: this one is keyed by symbol, `list(`*` = "text")`, and `hv_man_table_save_jtcvs()`'s is keyed by position, `list(list(row =, col =, text =))`. So `hv_test_footnotes_jtcvs()`'s output does not belong here; `hv_man_footnotes()`'s does.

"Invalid footnote symbol(s): ..." Symbols are drawn from the house set `* † ‡ § ¶ ||`. A letter or a digit is not one of them; lettered markers are the JTCVS renderer's convention.

"Output directory does not exist: ..." The `dirname()` of `file` is not created for you. Run `dir.create(dirname(file), recursive = TRUE)` first.

A hand-built footnote that misstates the table. The automatic `hv_tbl_summary()` -> `hv_man_table()` path keeps the default dagger in sync. When supplying `footnotes` yourself, make its wording match the statistic and percentile pair used to build the table.

See Also

`hv_man_table()` to build a compliant `flextable` first. `hv_man_footnotes()` for details on the default footnotes.

Examples

```
library(gtsummary)
tbl <- tbl_summary(trial, by = trt, include = c(age, grade))
ft <- hv_man_table(tbl)
out <- tempfile(fileext = ".docx")
hv_man_table_save(ft, out, abbreviations = c(N = "sample size"))
unlink(out)
```

---

hv_man_table_save_jtcvs

Deprecated: Write an hv_man_table_jtcvs() table to a compliant .docx

---

Description

Deprecated. Use `hv_man_table_save(ft, file, caption = )` on a table from `hv_man_table(tbl, style = "manuscript", layout = "describe")`. This function warns on every call and goes in the next release.

Usage

```
hv_man_table_save_jtcvs(
  ft,
  file,
  caption,
  footnotes = NULL,
  abbreviations = NULL,
  ...
)
```

Arguments

`ft`	A `flextable`, from `hv_man_table_jtcvs()`.
`file`	Output `.docx` path. The output directory (`dirname(file)`) must already exist; this function does not create it. (`%summarytable RTFFILE=/PDFFILE=` equivalent; output here is always `.docx`).
`caption`	Full caption text, e.g. `"Table 1. Baseline Characteristics"`, rendered bold above the table. This function does not auto-number tables for you; include the number yourself. (`%summarytable TBLTITLE=` equivalent).
`footnotes`	Optional list of `list(row =, col =, text =)`, one per footnote, in the order letters should be assigned. They are lettered from `a`. `col` is a single `col_keys` name. `row` may be a vector, in which case the one letter marks every row named, which is how several rows share a footnote. `row` indexes `ft`'s body rows as shown: for a sectioned table (built with `groupname_col`), that includes the section-header rows `hv_man_table_jtcvs()` interleaves into the body, so count those too. `hv_test_footnotes_jtcvs()` computes these indices for you

when the footnotes mark statistical tests. (`%summarytable ADDFN=` equivalent. `PRINTFN=1`'s house block is `hv_man_footnotes()`, which is CORR-shaped and cannot be passed here: build the equivalent entries in this function's `list(row =, col =, text =)` shape instead.)

`abbreviations` Optional named character vector, same as `hv_man_table_save()`, rendered via the shared `Key:` helper.

`...` Not used. Present so that `%summarytable` parameter names produce an error naming the argument to use instead.

Details

Stage 3 of the `%summarytable` split: writes the document. Replaces the macro's `RTFFILE=/PDFFILE=` output block.

Until it goes, it writes the table as `hv_man_table_save()` does, with `caption` above it and the table's own notes below. A test footnote from `hv_test_footnotes_jtcvs()` is dropped when the table already names that test, as a table from `hv_tbl_summary()` does in its dagger note; every other `footnotes` entry, a plain gtsummary table's test notes included, keeps a lettered superscript on its cells, lettered from `a`, and its line after the table's own notes.

Value

Invisibly, the `file` path.

Common mistakes

"`footnotes\[\[k\]\]$text` must be a single non-empty string." Every footnote needs all three of `row`, `col`, and `text`, and `text` must be one non-empty string. Before this check existed, an entry missing `text` wrote a document with a dangling superscript marker and an empty footnote line.

"`footnotes[[k]]` must be a list of the form list(row =, col =, text =)." `footnotes` is a list *of* footnotes, so a single one is `list(list(row = 1, col = "n_stat_1", text = "..."))`. Dropping the outer `list()` previously failed with `$ operator is invalid for atomic vectors`.

"`footnotes[[k]]$row` must be whole numbers ..." Row indices count the section-header rows `hv_man_table_jtcvs()` interleaves into the body. `hv_test_footnotes_jtcvs()` computes them for you.

"`file` must be a single non-empty file path." Check the argument order: it is `(ft, file, caption)`.

"Output directory does not exist: ..." The directory part of `file` (its `dirname()`) is not created for you; create it first with `dir.create(dirname(file), recursive = TRUE)`.

A wide table on a landscape page. Expected: a table wider than a portrait page's text block (6.5 inches on US Letter), as a `"both"` table or several trailing columns make it, is saved on a landscape page. One wider than landscape's (9.0 inches) has its columns narrowed in proportion to fit; the font size is not changed.

See Also

`hv_man_table_jtcvs()` to build a compliant `flextable` first.

Examples

```
# Instead of hv_man_table_jtcvs() and hv_man_table_save_jtcvs():
tbl <- hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = c("age", "grade")),
  continuous = "age", categorical = "grade", compare = "none"
)
ft <- hv_man_table(tbl, style = "manuscript", layout = "describe")
out <- tempfile(fileext = ".docx")
hv_man_table_save(ft, out, caption = "Table 2. Baseline Characteristics")
unlink(out)
```

---

hv_table_style *A table style, named by its role*

Description

The properties a table is drawn with, from the style matrix of the table-styles design: section rows, fill, borders, header, count column, note markers and wording, font, caption, row height, page and merged cells. A style is named for what the table is for, not for a journal. `"report"` is the working table of an analysis report, as `hv_man_table()` has always drawn it. `"manuscript"` is the table a manuscript submits: a full grid, a two-row header with the grouping spanning its levels, one overall `n` column, shaded section rows spanning the table, a caption, and no font of its own, so it takes the font of the document it is saved into.

Usage

```
hv_table_style(name = c("report", "manuscript"), fill = "#CAEDFB")
```

Arguments

`name`	`"report"` (default) or `"manuscript"`.
`fill`	The section rows' shading, one color written `"#RRGGBB"`. Default `"#CAEDFB"`. Journals differ here (one submission used `"#A5C9EB"`), so the fill is the property a study changes.

Details

Both styles follow the same house notation (medians, P values, marks), which `hv_tbl_summary()` writes, so a report reads like the manuscript and only the layout differs.

Value

A list of class `"hv_table_style"`, one element per property, for `hv_man_table()`'s `style` argument.

See Also

hv_man_table(), which draws a table in a style. The Table styles article prints every style's properties, read from this object.

Examples

```
hv_table_style("manuscript", fill = "#A5C9EB")$fill
```

---

hv_tbl_compare *Join a comparison column computed elsewhere onto a summary table*

Description

Adds one column to a table from hv_tbl_summary(), holding a value you computed yourself for each variable: typically the standardized difference %summarytable printed, which comes from the %stddiff macro and is not the number hv_tbl_summary()'s compare = "smd" gives. hvtiRpropensity::ps_stddiff() is the R port of %stddiff; this function is how its output reaches the table, without hvtiRtables depending on hvtiRpropensity.

Usage

```
hv_tbl_compare(tbl, values, label)
```

Arguments

tbl	A gtsummary object, normally from hv_tbl_summary().
values	Named character vector: variable name -> the text to show on that variable's first row.
label	The column header, e.g. "Std. Diff. (%)".

Details

Each value goes on its variable's first row, as %summarytable laid it out: beside the label of a categorical variable, not on every level. The column is placed rightmost, after any compare column, so compare = "pvalue" followed by this function puts the p-value and the standardized difference side by side. Call it again for a further column. hv_man_table() shows the column directly; The manuscript style shows it too, from the table's hv_trailing attribute, which this function extends.

The values are shown exactly as given, so format them first, for example sprintf("%.1f", 100 * d) for a difference as a percentage. A variable missing from values is left blank.

Value

tbl with the column added and its hv_trailing attribute extended by the new column.

Common mistakes

"`values` must be a character vector named by variable ..." Numbers are not formatted for you. Convert them with `sprintf()` or `format()`, and name each element by the variable's column name, the name in `groups`, not its label.

"`values` names variable(s) not in the table: ..." A name that matches no variable would silently show nothing. Usually a typo, or a variable dropped from `groups`.

See Also

`hv_tbl_summary()`, whose `compare` argument computes a p-value and gtsummary's generic standardized difference.

Examples

```
tbl <- hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = c("age", "grade")),
  continuous = "age", categorical = "grade",
  compare = "pvalue", show_variable = FALSE
)
# Standardized differences computed elsewhere, e.g. by
# hvtiRpropensity::ps_stddiff(), as proportions.
d <- c(age = 0.031, grade = 0.120)
tbl <- hv_tbl_compare(
  tbl, stats::setNames(sprintf("%.1f", 100 * d), names(d)),
  label = "Std. Diff. (%)"
)
hv_man_table(tbl)
```

---

hv_tbl_summary — *Build a gtsummary table from a SAS %summarytable-style grouped variable list*

Description

Thin wrapper over `gtsummary::tbl_summary()` modeled on the interface biostats team members already know from the `%summarytable` SAS macro: a grouped, ordered variable list (`groups`, the macro's `LIST=` equivalent) and variable-type buckets (`continuous`/`binary`/ `categorical`, the `CON3=`/`CAT1=`/`CAT2=` equivalents), rather than gtsummary's own tidyselect-based interface. Returns a plain `gtsummary` object, ready for `hv_man_table()`.

Usage

```
hv_tbl_summary(
  data,
  by = NULL,
  groups,
```

```
  continuous = character(0),
  binary = character(0),
  categorical = character(0),
  compare = c("pvalue", "smd", "both", "none"),
  percentiles = c(15, 85),
  overall = TRUE,
  continuous_stat = c("median", "mean", "both"),
  normal = character(0),
  mean_sd = character(0),
  show_variable = TRUE,
  cutexact = "any",
  ...
)
```

Arguments

`data` A data frame.

`by` Grouping variable name as a string (`%summarytable CLASS=` equivalent), or `NULL` for a single ungrouped "Overall" column.

`groups` Named list, section label -> variable names in display order (`%summarytable LIST=` equivalent), e.g. `list(Demography = c("age", "female"), Symptoms = c("nyha"))`. Each section must name at least one variable; a section holding `character(0)` is an error, not an empty section. Every variable named here must appear in exactly one of `continuous`, `binary`, or `categorical`, and every classified variable must appear in `groups`.

`continuous` Character vector of continuous variable names (`%summarytable CON3=` equivalent), summarized as set by `continuous_stat`. Variables the macro classified as `CON1=` (mean +/- SD, one-way ANOVA) or `CON2=` (median with min and max) belong here too; list `CON1=` variables in `normal` as well to keep the macro's mean +/- SD and ANOVA. Each named column must be numeric.

`binary` Character vector of 0/1 variable names (`%summarytable CAT1=` equivalent), summarized as `n (%)` on a single row. Each named column must have at most 2 distinct non-`NA` values, and must be logical, `0`/`1`, or `Yes`/`No` data, so that the "event" the single row counts is unambiguous. Anything else belongs in `categorical`, which shows every level. The event is the `TRUE`, `1`, or `Yes` side, and is stated to `gtsummary` explicitly rather than inferred, so columns read from SAS with `haven::read_sas()` summarize the same as their plain-vector equivalents.

`categorical` Character vector of multi-level variable names (`%summarytable CAT2=` equivalent), summarized as `n (%)` per level. No type rule applies: factors, characters, and small-integer codes are all accepted. Ordinal variables belong here too; this function does not run a trend test (`%summarytable`'s `ORD1=` distinction is not preserved).

`compare` One of `"pvalue"` (default), `"smd"`, `"both"`, or `"none"`. Ignored (treated as `"none"`) when `by` is `NULL`, since there is nothing to compare. `"smd"` and `"both"` require `by` to have exactly two groups, because a standardized mean difference is defined between two of them; they error otherwise. `"pvalue"` and `"none"` work at any number of groups. `%summarytable PVALUES=` equivalent, and `ASD=` only

in part: the "smd" number is gtsummary's generic standardized mean difference (gtsummary::add_difference()), a proportion computed alike for every continuous variable. %summarytable's column came from the %stddiff macro, which treats non-Gaussian and ordinal variables differently and prints a percentage, so the two disagree for those variables (by 30% on one skewed test variable). To show %stddiff's number, compute it with hvtiRpropensity::ps_stddiff() and add it with hv_tbl_compare(), beside compare = "pvalue" if wanted.

percentiles
: Numeric vector of length 2, the low/high percentile pair for continuous summaries, as increasing whole numbers between 0 and 100. Default c(15, 85), the hv_man_footnotes() house convention (%summarytable PP= equivalent).

overall
: Single TRUE/FALSE. When TRUE (default), prepends an Overall column across all groups (%summarytable TOTALCOL=1, the macro's default). Ignored when by is NULL, since the single column already is the overall one.

continuous_stat
: One of "median" (default), "mean", or "both": how continuous variables not listed in normal are summarized. "median" gives median [P<low>, P<high>]; "mean" gives mean +/- SD, with no spaces around the plus-minus sign and EHB/Blackstone paired rounding; "both" puts the two side by side: each group's statistic column shows mean +/- SD (or n (%) for a categorical row), and a median column beside it shows median [P<low>, P<high>], blank on rows with no median. Both renderers lay it out the same way. Choosing one for the manuscript is then a matter of dropping a column. The test does not change with the statistic: only normal selects ANOVA. The CORR footnote carried through hv_man_table() to hv_man_table_save() follows this choice automatically. Placed after overall so calls passing overall by position keep working.

normal
: Character vector, a subset of continuous: the variables to treat as normally distributed (%summarytable CON1= equivalent). Each is summarized as mean +/- SD whatever continuous_stat says, and, under compare = "pvalue" or "both", tested with classical (pooled-variance) one-way ANOVA, the test the macro ran. Default none. Each name must also appear in continuous. Placed after continuous_stat so positional calls keep working. To show mean +/- SD without the ANOVA, use mean_sd.

mean_sd
: Character vector, a subset of continuous: the variables to *display* as mean +/- SD while keeping the nonparametric test (Wilcoxon rank-sum, or Kruskal-Wallis for 3+ groups). Use it for variables that are roughly symmetric in a table whose every continuous test should stay nonparametric. The display is the same as for normal, cell mark included when the table mixes statistics; only normal changes the test. A variable may appear in both, and is then tested with ANOVA. Default none. Last in the signature so positional calls keep working.

show_variable
: Single TRUE/FALSE. When TRUE (default), adds a "Variable" column beside the label holding each variable's column name in data, on its first row, as %summarytable's report did. It belongs to the report view: hv_man_table() and hv_man_table_display() show it, while hv_man_table_save() always drops it and the manuscript style never shows it, so a saved table is a manuscript table whatever this says. Set FALSE to leave it out of the report view too.

cutexact
: When a binary or categorical variable is tested with Fisher's exact test rather than Pearson's chi-square (no continuity correction). The default, "any", uses

Fisher whenever any one group-by-level cell has an expected count below 5 (gtsummary's rule, and what the CORR team asked for). A number opts in to `%summarytable`'s `CUTEXACT=` rule, to reproduce a SAS table: Fisher when at least that percentage of cells expect fewer than 5. The macro's default was `50`; `0` makes every categorical test exact and a value above 100 makes none exact. In the CORR table each Fisher p-value is marked with a superscript double dagger, footnoted "Fisher exact test.", in every style. Used only when `compare` produces p-values. The other tests that ran are named in the dagger note instead, as "P values from the Wilcoxon rank sum test and Pearson's chi-square test.", and the CORR table puts the dagger on the p-value header too. `compare = "smd"` or `"both"` names the difference column "Std. Diff." and the same note says what it is.

`...` Not used. Present so that `%summarytable` parameter names produce an error naming the argument to use instead.

Details

Continuous variables are summarized as `median [P<low>, P<high>]` by default, or as mean +/- SD, or both (`continuous_stat`), and tested non-parametrically (Wilcoxon rank-sum for 2 groups, Kruskal-Wallis for 3+), the macro's `CON3=`. A variable you also list in `normal` is the macro's `CON1=` instead: mean +/- SD, tested with one-way ANOVA. `percentiles` defaults to the house convention documented in `hv_man_footnotes()` (15th/85th), overridable per study (`%summarytable` equivalent: `PP=`).

The house notation: each median and percentile is shown to 2 significant digits, and a value of 10 or more to the units place (`102 [63, 170]`, `0.31 [0.21, 0.39]`), and a P value has no leading zero (`.013`, `<.001`, `>.9`).

When rows of one table show different statistics, the header and dagger name the statistic most rows use, and the cells of the other are marked instead of the row labels (the macro's `DESCLABEL=1` suffixes, such as `"Age, Mean±SD"`, are gone). When more rows show mean +/- SD than a median, the mean is the default and a section sign (§) marks the median cells; otherwise a pilcrow (¶) marks the mean cells. Each mark has its own footnote. A tie leaves the median as the default.

Mean/SD output follows Eugene H. Blackstone's (EHB's) paired reporting rule: the mean is rounded to the first-significant-digit place of the SD, and the SD is rounded one place finer. When the SD begins with 1, both retain one additional place. Exact ties round to even; values beyond a tie round up. This applies to every group and Overall column when `continuous_stat = "mean"`, and to the mean row under `"both"`. Medians and percentiles follow the house notation above instead, and their exact ties also round to even, as for mean +/- SD: 62.5 shows as 62, where gtsummary and SAS showed 63. When an SD is zero or unavailable, no paired precision can be inferred, so gtsummary's existing display is kept.

The returned object carries these renderer attributes: `hv_stat_label`, the sub-header text naming the statistics shown (`"No. (%) or Median [<low>, <high> percentiles]"` by default, with ordinal suffixes; it follows `continuous_stat`); `hv_trailing`, a named character vector ready to pass as naming the comparison columns the manuscript style shows, when `compare` produced one (`NULL` when `compare = "none"`); and `hv_footnotes`, the CORR footnote block that `hv_man_table()` carries into `hv_man_table_save()` so the default dagger follows `continuous_stat` and `percentiles` automatically. A mixed table also carries `hv_cell_marks`, the symbol and variables whose cells `hv_man_table()` marks, and a table built with `continuous_stat = "both"` carries `hv_median_label`.

Value

A gtsummary object, ready for hv_man_table(). See Details for the hv_stat_label, hv_trailing, and hv_footnotes renderer attributes. A table whose rows mix mean and median also carries hv_cell_marks, a list with symbol ("§" or "¶") and variables (the names whose cells carry it); the attribute is NULL otherwise. The manuscript style also reads hv_by_label, the grouping's name (the variable's label when the data carry one), and the hidden hv_n column of table_body, each variable's overall non-missing count.

Common mistakes

"<var> appears in more than one of continuous, binary, and categorical." Each variable is classified exactly once. A 0/1 variable is binary; a multi-level factor is categorical.

"continuous lists <var>, but <var> is factor data ..." The bucket has to suit the data, not just be free of overlaps. A non-numeric variable in continuous used to reach gtsummary, which only *messages* about it, and produced a complete, correctly styled table whose every statistic was NA.

"binary lists <var>, but <var> has ..." binary renders one n (%) row, so it takes at most two distinct values and needs an unambiguous event value: logical, 0/1, or Yes/No. A three-level factor, or a two-level one like F/M, belongs in categorical.

"Variable(s) in groups not classified ..." Every variable listed in groups also needs a type bucket, and every classified variable needs to appear in groups. The two lists must match.

"compare = "smd" requires exactly two groups." A standardized mean difference is defined between two groups. Use compare = "pvalue" for three or more. If by is a factor with an unused level, droplevels() is usually what you want.

"by must not also be listed in groups." by is the grouping variable being compared across, not a row to summarize. Before this check existed, the combination reached gtsummary and failed several calls later with "names must be NULL or a character vector, not an empty integer vector.", a message that never mentioned by. Remove the variable from groups.

See Also

hv_man_table() to render the result. hv_man_footnotes() for the percentile-footnote house convention.

Examples

```
# A baseline-characteristics table of the kind a study manuscript
# actually carries: demography and disease sections, compared
# across treatment arms.
hv_tbl_summary(
  gtsummary::trial,
  by = "trt",
  groups = list(
    Demography = c("age", "marker"),
    Disease = c("stage", "grade")
  ),
  continuous = c("age", "marker"),
  categorical = c("stage", "grade")
)
```

hv_test_footnotes_jtcvs

Deprecated: Build lettered test footnotes from a gtsummary table

Description

Deprecated. A table from `hv_man_table(tbl, style = "manuscript", layout = "describe")` names its tests in its dagger note and marks Fisher's exact test on its P values, so it needs no test footnotes. This function warns on every call and goes in the next release.

Usage

```
hv_test_footnotes_jtcvs(tbl)
```

Arguments

`tbl`	A `gtsummary` object, normally from `hv_tbl_summary()`. Must be the same object passed to `hv_man_table_jtcvs()`, since the row indices are computed from its `table_body`.

Details

Reproduces the macro's lettered test footnotes. The letters differ by design: the macro lettered by test (`a=ANOVA`, `b=Kruskal-Wallis`, ...), while here letters follow the order the tests appear. One-way ANOVA appears only for variables listed in `hv_tbl_summary()`'s `normal`.

The SAS `%summarytable` macro marks every p-value with a superscript letter naming the test behind it. This builds those footnotes from the `test_name` column `gtsummary::add_p()` records, in the format `hv_man_table_save_jtcvs()`'s `footnotes` argument expects, so you do not have to map test identifiers or count body rows by hand. `hv_man_table_save_jtcvs()` drops the ones a describe table already names itself.

Takes the `gtsummary` object rather than the `flextable`, because `hv_man_table_jtcvs()` discards `test_name` when it reshapes the body. The returned `row` indices are rendered body rows, counting the section-header rows `hv_man_table_jtcvs()` interleaves, and are therefore valid only against that renderer's output.

Letters follow a fixed order (Wilcoxon, Kruskal-Wallis, chi-square, Fisher), filtered to the tests actually used, so two tables in the same manuscript that use the same tests get the same letters regardless of variable order. Every row sharing a test collapses onto one letter.

Returns an empty list when there is nothing to mark: `by = NULL`, `compare = "none"`, or `compare = "smd"`, since a standardized difference is not a test. An empty list concatenates and renders harmlessly, so no conditional is needed at the call site.

With `compare = "both"` the cell reads e.g. `0.7 (-0.03)` and the letter is appended to the cell, after the SMD. Note also that `gtsummary::add_difference()` requires exactly two groups, so `compare = "smd"` and `compare = "both"` are unavailable for a 3-group table; `compare = "pvalue"` is.

An unrecognized `test_name` (only reachable from a hand-built object that passed `add_p(test = ...)`, since `hv_tbl_summary()` never sets `test`) is kept, using the raw identifier as its footnote text, and warned about. A marked cell whose letter has no definition would be worse than an unpolished label.

Value

A list of `list(row =, col =, text =)` entries, one per distinct test, ready to pass as `hv_man_table_save_jtcvs()`'s `footnotes` argument. `list()` when there is nothing to mark.

Common mistakes

"`tbl` must be a gtsummary table object." You passed the `flextable`. This needs the `gtsummary` object, because `hv_man_table_jtcvs()` drops the `test_name` column when it reshapes the body – by render time the test identities are gone. Pass the same object you passed to the renderer.

An empty list back, and no footnotes in the document. Expected, not a failure: nothing is marked when `by = NULL`, `compare = "none"`, or `compare = "smd"`, since a standardized difference names no test. An empty list concatenates and renders harmlessly, so the call site needs no conditional.

Row indices that land on the wrong rows. They are computed from `tbl$table_body` at the moment you call this, and they count the section-header rows `hv_man_table_jtcvs()` interleaves, so they are valid against that renderer's output and nothing else. Finish the table first and call this last.

Passing the result to `hv_man_table_save()`. This is the JTCVS shape. The CORR saver takes a symbol-keyed list and rejects it; its equivalent is `hv_man_footnotes()`.

See Also

`hv_man_table_save_jtcvs()`, which renders the result. `hv_tbl_summary()`, which produces a suitable `tbl`. `hv_man_footnotes()` for the house-universal footnotes, which are CORR-shaped and belong to `hv_man_table_save()`, not to this function's JTCVS output.

Examples

```
# Instead of hv_test_footnotes_jtcvs():
tbl <- hv_tbl_summary(
  gtsummary::trial, by = "trt",
  groups = list(Demography = c("age", "grade")),
  continuous = "age", categorical = "grade", compare = "pvalue"
)
ft <- hv_man_table(tbl, style = "manuscript", layout = "describe")
attr(ft, "hv_footnotes")[["\u2020"]]
```

Index

flextable::footnote(), *15*
flextable::merge_none(), *9*
format(), *21*

gtsummary::add_difference(), *23*, *26*
gtsummary::add_p(), *26*
gtsummary::as_flex_table(), *9*
gtsummary::style_sigfig(), *9*
gtsummary::tbl_summary(), *9*, *13*, *21*

hv_check_docx, 2
hv_correlation_table, 4
hv_man_footnotes, 6
hv_man_footnotes(), *9*, *11*, *12*, *15*, *16*, *18*, *23–25*, *27*
hv_man_table, 7
hv_man_table(), *3*, *6*, *7*, *11*, *13–16*, *19–21*, *23–25*
hv_man_table_display, 10
hv_man_table_display(), *9*, *16*, *23*
hv_man_table_jtcvs, 12
hv_man_table_jtcvs(), *17*, *18*, *26*, *27*
hv_man_table_save, 14
hv_man_table_save(), *3*, *4*, *6–13*, *18*, *23*, *24*, *27*
hv_man_table_save_jtcvs, 17
hv_man_table_save_jtcvs(), *7*, *14*, *16*, *26*, *27*
hv_table_style, 19
hv_table_style(), *2*, *6*, *8*, *10*
hv_tbl_compare, 20
hv_tbl_compare(), *13*, *23*
hv_tbl_summary, 21
hv_tbl_summary(), *5–13*, *15*, *16*, *18–21*, *26*, *27*
hv_test_footnotes_jtcvs, 26
hv_test_footnotes_jtcvs(), *7*, *16–18*

officer::body_add_fpar(), *16*

sprintf(), *21*

stats::cor.test(), *5*