Package: ggconsulting (via r-universe)

June 19, 2026

Type Package

Title Executive-Grade Themes and Helpers for 'ggplot2'

Version 0.0.0.9000

Description An opinionated 'ggplot2' extension for executive-grade consulting output. Ships archetype themes, palettes, scale helpers, locale-aware label helpers, and a data-aware polish layer.

License MIT + file LICENSE

URL https://github.com/viniciusoike/ggconsulting

BugReports https://github.com/viniciusoike/ggconsulting/issues

Depends R (>= 3.5)

Encoding UTF-8

LazyData true

Imports cli, ggplot2, lifecycle, rlang, scales, systemfonts

Suggests testthat (>= 3.0.0), vdiffr, withr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev

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

Date/Publication 2026-06-03 19:58:19 UTC

RemoteUrl https://github.com/viniciusoike/ggconsulting

RemoteRef HEAD

RemoteSha 0fb697fee93cb588c283ecd8c6d569619e128531

Contents

`br_macro` *Brazilian macroeconomic indicators (monthly)*

Description

Monthly observations of five headline Brazilian macro series from 2012-03 through 2024-12. Wide format: one row per month, one column per indicator. Snapshot frozen at 2024-12-31. Suitable for `theme_editorial()` demos and `ct_locale("pt-BR")` formatting.

Usage

```
br_macro
```

Format

A data frame with 154 rows and 6 columns:

date First day of the month (Date).

selic Meta Selic, % a.a., end-of-month (numeric).

ipca_12m IPCA accumulated over the trailing 12 months, % (numeric).

ibc_br IBC-Br activity index, seasonally adjusted, 2002=100 (numeric).

usd_brl USD/BRL exchange rate (compra), BRL per USD, end-of-month (numeric).

unemployment PNADC unemployment rate, % (numeric).

Details

Indicators are reported in their native units; daily series (`selic`, `usd_brl`) are sampled at the end of each month.

Source

Banco Central do Brasil SGS, series 432, 13522, 24364, 1, and 24369. Unemployment originates with IBGE PNADC and is redistributed through SGS. Fetched via the `rbcb` package. Snapshot date: 2024-12-31. See `data-raw/br_macro.R`.

Examples

```
head(br_macro)
```

---

bu_quarterly *Quarterly P&L for a fictional Brazilian conglomerate*

Description

Simulated quarterly revenue, COGS, EBITDA, and headcount for five business units of a mid-cap Brazilian conglomerate, spanning 16 quarters (2021-Q1 through 2024-Q4). Suitable for demos of `theme_strategy()`, stacked-bar `ct_finish()` value labels, and `fmt_brl()`. All monetary values are in millions of Brazilian Real (R$ MM).

Usage

```
bu_quarterly
```

Format

A data frame with 80 rows and 6 columns:

quarter First day of the calendar quarter (Date).

business_unit Factor with 5 levels: Industrial, Consumer, Health, Logistics, Digital.

revenue_brl Net revenue, R$ MM (numeric).

cogs_brl Cost of goods sold, R$ MM (numeric).

ebitda_brl EBITDA, R$ MM (numeric); can be negative.

headcount Period-end full-time-equivalent employees (integer).

Source

Simulated. See `data-raw/bu_quarterly.R`.

Examples

```
head(bu_quarterly)
```

client_nps *Quarterly NPS by client segment*

Description

Simulated quarterly Net Promoter Score for three client segments (Enterprise, Mid-Market, SMB) across 12 quarters (2022-Q1 through 2024-Q4). Suitable for line-chart demos with last-point labels and `fmt_delta()` / `fmt_pct()`.

Usage

```
client_nps
```

Format

A data frame with 36 rows and 4 columns:

quarter First day of the calendar quarter (Date).

segment Factor with 3 levels: Enterprise, Mid-Market, SMB.

nps Net Promoter Score, integer in [-100, 100].

responses Number of survey responses in the period (integer).

Source

Simulated. See `data-raw/client_nps.R`.

Examples

```
head(client_nps)
```

ct_defaults *Set or restore ggconsulting aesthetic defaults*

Description

`ct_set_defaults()` overrides ggplot2 *aesthetic* defaults via `ggplot2::update_geom_defaults()`. v0.1 sets a single override: `geom_point size = 2.5`. `ct_unset_defaults()` restores whatever values were active the first time `ct_set_defaults()` ran — *not* ggplot2's untouched baseline.

Originals are captured once into a package-private environment, so repeated `ct_set_defaults()` calls are idempotent. Column width and linewidth are handled elsewhere — see `ct_col()` / `ct_line()` for explicit formal overrides, or apply `ct_theme()` / `theme_strategy()` for linewidth via `from_theme()`.

Usage

```
ct_set_defaults()

ct_unset_defaults()
```

Value

Both functions return `invisible(NULL)`.

Examples

```
ct_set_defaults()
ct_unset_defaults()
```

`ct_finish` *Apply data-aware finishing touches to a plot*

Description

[Experimental]

Companion to `ct_theme()` that runs *after* the geom layer is built, so it can inspect the data + active geom to inject value labels, sorting, highlighting, end labels, and scale expansion. Compose with +, after the geoms.

Usage

```
ct_finish(
  values = FALSE,
  sort = NULL,
  label_fmt = NULL,
  highlight = NULL,
  end_labels = FALSE,
  expand = "auto",
  muted_color = "#A8A4A0"
)
```

Arguments

`values`	`TRUE` adds value labels above bars / next to points. `"auto"` adds them only when the first geom is a column or bar.
`sort`	One of `"asc"`, `"desc"`, or `NULL`. When set, reorders the factor levels of the x aesthetic by the y aesthetic.
`label_fmt`	Either a formatter function (anything that maps a numeric vector to a character vector), or one of the shortcut names `"brl"`, `"number"`, `"pct"`, `"delta"` resolved to the matching `fmt_*()` helper.

highlight	Value(s) of the x aesthetic to emphasise. Matching bars use the active palette's main colour; non-matching bars use muted_color. Inserted as a scale_*_manual().
end_labels	For line plots: when TRUE, label the last point of each series with the group identifier.
expand	"auto" picks geom-aware scale expansion (room above column tops, right-side room for line end labels); FALSE disables.
muted_color	Fill / colour used for non-highlighted categories.

Value

A ct_finish object, added to a plot via +. The ggplot_add() method composes the requested layers and scales.

Examples

```
library(ggplot2)
d <- data.frame(g = LETTERS[1:5], v = c(3, 8, 5, 12, 7))
p <- ggplot(d, aes(g, v)) +
  geom_col() +
  ct_finish(values = TRUE, sort = "desc", label_fmt = "brl", highlight = "D") +
  theme_strategy()
```

---

ct_formatters *Locale-aware label formatters*

Description

Factory functions: each returns a function compatible with scales::label_* (callable on a numeric vector). Marks (thousands, decimals) and currency symbols come from the active ggconsulting locale, set via ct_locale().

- fmt_number() — plain numbers with locale separators.
- fmt_brl() — Brazilian Real. Always renders as R$ regardless of active locale; the locale argument lets you override marks.
- fmt_currency() — uses the active locale's currency symbol.
- fmt_pct() — percentages. By default interprets input as a fraction (0.5 → "50%"); pass scale = 1 when the input is already in percent units (50 → "50%").
- fmt_delta() — percentage-point-style signed deltas (+1,2pp / -0,3pp / 0,0pp).
- fmt_month() — Date / POSIXct → localised month string.

Negatives default to a minus prefix; currency helpers also support style = "accounting", which wraps negatives in parentheses. Currency rendering uses a non-breaking space (U+00A0) between symbol and number so the pair never wraps across lines.

Usage

```
fmt_number(decimals = 0, locale = NULL)

fmt_brl(decimals = 2, style = c("minus", "accounting"), locale = NULL)

fmt_currency(decimals = 2, style = c("minus", "accounting"), locale = NULL)

fmt_pct(decimals = 1, scale = 100, locale = NULL, accuracy = NULL)

fmt_delta(decimals = 1, suffix = "pp", locale = NULL)

fmt_month(format = c("abbr", "full"), locale = NULL)
```

Arguments

`decimals`	Number of decimal places.
`locale`	Optional locale name. Defaults to the active locale.
`style`	`"minus"` (default) or `"accounting"` for currency.
`scale`	Multiplier applied to the input before rendering, for `fmt_pct()` only. Defaults to `100` (input is a fraction). Use `scale = 1` when the input is already in percent units.
`accuracy`	Optional explicit accuracy passed to `scales::percent()`; overrides `decimals` if set.
`suffix`	Suffix for `fmt_delta()`. Defaults to `"pp"`.
`format`	`"abbr"` (default) or `"full"` for `fmt_month()`.

Value

A formatter function: `function(x) character`.

Examples

```
fmt_number(decimals = 1)(1234.5)
fmt_brl()(c(1000, -500))
fmt_pct(decimals = 0)(0.5)
fmt_pct(decimals = 1, scale = 1)(50)
fmt_delta()(c(1.2, -0.3, 0))
fmt_month()(as.Date("2026-08-15"))
```

`ct_geoms` *Drop-in replacements for geom_col(), geom_line(), and geom_point()*

Description

Three-line pass-throughs that ship consulting-grade defaults for `ggplot2::geom_col()` `width`, `ggplot2::geom_line()` `linewidth`, and `ggplot2::geom_point()` `size`. Use them when you want the defaults baked into the call site; plain `geom_line()` will also pick up linewidth from `ct_theme()` / `theme_strategy()` via `from_theme()`, and `geom_point()` size is autoloaded by `ct_set_defaults()`.

Usage

```
ct_col(..., width = 0.8)

ct_line(..., linewidth = 0.7)

ct_point(..., size = 2.5)
```

Arguments

`...`	Forwarded to the underlying ggplot2 constructor.
`width`	Column width. Defaults to 0.8 (vs ggplot2's 0.9).
`linewidth`	Line width. Defaults to 0.7 (vs ggplot2's 0.5).
`size`	Point size. Defaults to 2.5 (vs ggplot2's 1.5).

Value

A ggplot2 `ggplot2::layer()` object.

Bar charts with Date x

`ct_col()`'s default `width = 0.8` is in *x-axis units*. On a Date x-axis that is 0.8 *days*, which renders bars as slivers when the data is spaced quarterly or monthly. For bar charts, convert the x variable to either an ordered factor (e.g. `"21Q1"`, `"21Q2"`, ...) or a numeric index before plotting; line charts on Date are fine.

Examples

```
library(ggplot2)
d <- data.frame(g = c("A", "B", "C"), v = c(3, 5, 2))
p_col   <- ggplot(d, aes(g, v)) + ct_col()
p_line  <- ggplot(economics, aes(date, unemploy)) + ct_line()
p_point <- ggplot(mtcars, aes(wt, mpg)) + ct_point()
```

ct_locale *Set the active ggconsulting locale*

Description

Stores the choice in `options(ggconsulting.locale = ...)`. The active locale is read by the formatter helpers (`fmt_number()`, `fmt_currency()`, `fmt_month()`, etc.) when their `locale` argument is `NULL`. Defaults to `"pt-BR"`; `"en-US"` is always available.

Usage

```
ct_locale(locale = c("pt-BR", "en-US"))
```

Arguments

`locale`	One of `"pt-BR"` or `"en-US"`.

Details

Does *not* touch `Sys.setlocale()` — that's OS-dependent and breaks on Windows CI. ggconsulting ships its own month tables and formatting marks instead.

Value

The previous locale, invisibly. Use it to round-trip: `old <- ct_locale("en-US"); ...; ct_locale(old)`.

Examples

```
old <- ct_locale("en-US")
ct_locale(old)
```

ct_palette *Get palette colours*

Description

[Experimental]

Returns the hex colour vector for a named palette, optionally subsetting or interpolating to `n` colours. Called with no arguments, returns the names of all available palettes.

Usage

```
ct_palette(palette = NULL, n = NULL, reverse = FALSE)
```

Arguments

palette	Palette name (e.g. "strategy_navy") or NULL (default) to list available palette names.
n	Number of colours to return. When n is smaller than the palette, the first n colours are returned. When n is larger, colours are interpolated via grDevices::colorRampPalette() (with a warning). Defaults to NULL (return the full palette).
reverse	Reverse palette order before subsetting. Defaults to FALSE.

Value

A character vector of hex colours, or (when palette is NULL) a character vector of palette names.

Examples

```
# List available palettes
ct_palette()

# Full palette
ct_palette("strategy_navy")

# First 3 colours
ct_palette("strategy_navy", n = 3)

# Interpolate to 9 colours
ct_palette("strategy_navy", n = 9)
```

---

ct_palette_show *Preview ggconsulting palettes as a swatch*

Description

Returns a ggplot showing palettes as colour tiles with the hex value overlaid in monospace. With palette = NULL, every palette shipped in .ct_palettes is shown faceted.

Usage

```
ct_palette_show(palette = NULL)
```

Arguments

palette	Palette name (e.g. "strategy_emerald"), character vector of colours, or NULL (default) to show every shipped palette.

Value

A ggplot object.

Examples

```
p_all  <- ct_palette_show()
p_one  <- ct_palette_show("strategy_emerald")
p_vec  <- ct_palette_show(c("#0F4D38", "#177B57", "#3DA876"))
```

`ct_scales` *Consulting colour and fill scales*

Description

[Experimental]

Discrete and continuous scales backed by ggconsulting palettes. Discrete variants interpolate (with a warning) when the data has more levels than the palette can hold. Continuous variants gradient across all palette colours via `ggplot2::scale_color_gradientn()` / `ggplot2::scale_fill_gradientn()`.

British spellings (`scale_colour_ct()`, `scale_colour_ct_c()`) are provided as aliases.

Usage

```
scale_color_ct(palette = "strategy_navy", reverse = FALSE, ...)

scale_colour_ct(palette = "strategy_navy", reverse = FALSE, ...)

scale_fill_ct(palette = "strategy_navy", reverse = FALSE, ...)

scale_color_ct_c(palette = "strategy_navy", direction = 1, ...)

scale_colour_ct_c(palette = "strategy_navy", direction = 1, ...)

scale_fill_ct_c(palette = "strategy_navy", direction = 1, ...)
```

Arguments

`palette`	Palette name (e.g. `"strategy_navy"`) or character vector of colours.
`reverse`	Reverse palette order before mapping. Defaults to `FALSE`.
`...`	Forwarded to the underlying ggplot2 scale constructor.
`direction`	`1` (default) or `-1` to reverse the continuous gradient.

Value

A ggplot2 scale.

Examples

```
library(ggplot2)
p_d <- ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
  geom_point() +
  scale_color_ct("strategy_azure")
p_c <- ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_raster() +
  scale_fill_ct_c("strategy_emerald")
```

`ct_theme` *Build a consulting-grade ggplot2 theme*

Description

[Experimental]

Flexible builder behind the archetype presets (`theme_strategy()` and friends). Composes `ggplot2::theme_minimal()` with executive-output overrides via `theme_sub_*()` helpers, sets the theme `geom` element so `from_theme()`-aware layers (e.g. `ggplot2::geom_line()`) inherit the resolved linewidth and main colour, and applies the resolved font with a fallback chain.

Usage

```
ct_theme(
  palette = "strategy_navy",
  font = "Inter",
  font_fallback = c("Helvetica Neue", "Arial", "sans"),
  density = c("normal", "tight", "loose"),
  context = c("presentation", "report", "screen"),
  base_size = NULL,
  main_color = NULL
)
```

Arguments

`palette`	Palette name (e.g. `"strategy_navy"`) or character vector of colours.
`font`	Preferred font family.
`font_fallback`	Character vector of fallback families to try if `font` is unavailable. The generic R families `"sans"`, `"serif"`, and `"mono"` are always recognised as terminal fallbacks.
`density`	One of `"normal"`, `"tight"`, `"loose"`. Controls `panel.spacing` and axis-text margins.
`context`	One of `"presentation"`, `"report"`, `"screen"`. Drives default `base_size` and `plot.margin`.
`base_size`	Base font size. If `NULL`, derived from `context`.

`main_color` Routed into the theme `geom` element's `ink` slot, so unmapped geoms pick it up via `from_theme()`. `NULL` falls back to the palette's first colour. Title colour is a fixed neutral near-black (`#1A1A1A`); override with a follow-on `theme()` call if you want it palette-tinted.

Value

A `ggplot2::theme()` object.

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  ct_theme()
```

---

ebitda_bridge *FY23 to FY24 EBITDA bridge*

Description

Eight-row ordered breakdown decomposing the year-over-year EBITDA change for the fictional conglomerate behind bu_quarterly into volume, price, cost, mix, FX, and one-off effects. The first and last rows are absolute levels; the six intermediate rows are signed deltas that reconcile the endpoints. Designed for waterfall charts.

Usage

```
ebitda_bridge
```

Format

A data frame with 8 rows and 3 columns:

component Ordered factor naming the bridge step.

value_brl R$ MM (numeric). Absolute level for endpoint rows; signed delta for intermediate rows.

type Factor with 3 levels: `"total"` for endpoint rows, `"increase"` or `"decrease"` for deltas.

Source

Simulated. See `data-raw/ebitda_bridge.R`.

Examples

```
ebitda_bridge
```

has_font *Test whether a font family is installed*

Description

Wrapper around `systemfonts::system_fonts()` used to gate font-dependent code paths and tests.

Usage

```
has_font(name)
```

Arguments

`name`	Font family name as it appears in `systemfonts::system_fonts()$family`.

Value

`TRUE` if `name` matches an installed family, `FALSE` otherwise.

Examples

```
has_font("Arial")
```

ibov_sectors *Monthly B3 sector indices*

Description

Monthly closing values and total monthly return for the Ibovespa (IBOV) and six B3 sector indices, covering 2020-01 through 2024-12. Snapshot frozen at 2024-12-31. Long format: one row per (`sector_index`, month). Suitable for `theme_finance()` demos and small-multiples.

Usage

```
ibov_sectors
```

Format

A data frame with ~420 rows and 4 columns:

date First day of the month (Date).

sector_index Factor with 7 levels: IBOV, IFNC, INDX, IMAT, IEEX, ICON, IMOB.

close End-of-month index level (numeric).

return_m Monthly arithmetic return, fraction (numeric); `NA` for the first month of each series.

Source

B3 historical index series, fetched via the rb3 package. Snapshot date: 2024-12-31. See data-raw/ibov_sectors.R.

Examples

```
head(ibov_sectors)
```

`install_consulting_fonts` *Install consulting fonts from Google Fonts*

Description

Downloads and installs the font families used by the three archetype themes (theme_strategy(), theme_finance(), theme_editorial()). All fonts are OFL or Apache-2.0 licensed.

Usage

```
install_consulting_fonts(fonts = NULL, dest = NULL, quiet = FALSE)
```

Arguments

fonts	Character vector of family names to install, or NULL (default) for the full set: Inter, Source Sans 3, Lato, Source Serif 4, IBM Plex Sans. Validated against the known catalog.
dest	Destination directory. NULL (default) uses a platform-appropriate user font directory: ~/Library/Fonts on macOS, ~/.local/share/fonts on Linux, or a session tempdir on Windows (with systemfonts::register_font() for the active session).
quiet	Suppress informational messages. Errors are always emitted. Defaults to FALSE.

Details

Variable-weight families (Inter, Source Sans 3, Source Serif 4, IBM Plex Sans) are downloaded as variable .ttf files containing all weights (Light through Bold and beyond). Lato is downloaded as four static .ttf files (Regular, Bold, Italic, Light).

Value

Invisibly, a character vector of installed file paths.

Examples

```
## Not run:
install_consulting_fonts()
install_consulting_fonts("Inter")

## End(Not run)
```

market_share *Annual market share for a fictional sector*

Description

Simulated annual market share for six players (five named + Others) over ten years (2015 through 2024). Shares sum to 1 within each year. Story arc: incumbent erodes, challenger gains. Designed for slope and bump-chart demos.

Usage

```
market_share
```

Format

A data frame with 60 rows and 3 columns:

year Calendar year (integer).

company Factor with 6 levels: Player A, Player B, Player C, Player D, Player E, Others.

share Market share, fraction in [0, 1] (numeric).

Source

Simulated. See `data-raw/market_share.R`.

Examples

```
head(market_share)
```

theme_editorial *Editorial archetype theme*

Description

Preset path through `ct_theme()` tuned for analyst notes and market commentary: serif typography (`"Source Serif 4"` with a Georgia / Times / serif fallback chain), a slightly larger title with tighter leading, and italic subtitles for typographic personality.

Usage

```
theme_editorial(main_color = NULL, ...)
```

Arguments

main_color	Routed into the theme geom ink slot for from_theme() linkage. NULL (default) falls back to editorial_warm[1]. Title colour is a fixed neutral near-black.
...	Forwarded to ct_theme() — e.g. density, context, base_size, or an explicit palette override.

Value

A ggplot2::theme() object.

Examples

```
library(ggplot2)
p <- ggplot(economics, aes(date, unemploy)) +
  geom_line() +
  theme_editorial()
```

---

theme_finance *Finance archetype theme*

Description

Preset path through ct_theme() tuned for finance reports and pitch books: serif typography ("Source Serif 4" with a Georgia / Times / serif fallback chain), regular-weight title (restrained — finance reports avoid bold), a lighter major gridline than the strategy archetype, and denser defaults (density = "tight", context = "report") so plots read closer to a printed page than a slide.

Usage

```
theme_finance(main_color = NULL, density = "tight", context = "report", ...)
```

Arguments

main_color	Routed into the theme geom ink slot for from_theme() linkage. NULL (default) falls back to finance_classic[1]. Title colour is a fixed neutral near-black.
density	Passed to ct_theme(). Defaults to "tight" — finance reports favour denser layouts than presentation slides.
context	Passed to ct_theme(). Defaults to "report" — drives a smaller base_size and tighter plot.margin.
...	Forwarded to ct_theme() — e.g. base_size, or an explicit palette override.

Value

A ggplot2::theme() object.

Examples

```
library(ggplot2)
p <- ggplot(economics, aes(date, unemploy)) +
  geom_line() +
  theme_finance()
```

`theme_strategy` *Strategy archetype theme*

Description

Preset path through `ct_theme()` tuned for a minimal, generous-whitespace, navy-default look inspired by top-tier global strategy consultancies.

Usage

```
theme_strategy(main_color = NULL, ...)
```

Arguments

`main_color`	Routed into the theme geom `ink` slot so unmapped geoms pick it up via `from_theme()`. Defaults to `strategy_navy[1]`. Pass any value `grDevices::col2rgb()` accepts to override. Does not affect title colour (a fixed neutral near-black).
`...`	Forwarded to `ct_theme()` — e.g. `density`, `context`, `base_size`.

Value

A `ggplot2::theme()` object.

Examples

```
library(ggplot2)
p <- ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  theme_strategy()
```

Index