Package: a11yviz (via r-universe)

May 23, 2026

Title Accessibility Toolkit for 'ggplot2', 'plotly', and 'Quarto'

Version 0.1.8

Description Makes charts and documents accessible across 'ggplot2', 'plotly', and 'Quarto', aligned with the Web Content Accessibility Guidelines (WCAG 2.1, <https://www.w3.org/TR/WCAG21/>). Includes WCAG-tagged palettes, alt-text scaffolds, audits, a document rubric, heading and reading-level checks, 'shiny' ARIA helpers, and a stylesheet for 'DT' and 'DiagrammeR', as employed in Shin et al. (2026) <doi:10.1177/07319487251412879>.

License MIT + file LICENSE

Language en-US

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports grDevices, rlang, stats, utils, yaml

Suggests ggplot2, palmerpenguins, plotly, RColorBrewer, viridisLite, DT, shiny, bslib, knitr, rmarkdown, quarto, testthat (>= 3.0.0), spelling, withr

Depends R (>= 4.1.0)

URL https://mshin77.github.io/a11yviz, https://github.com/mshin77/a11yviz

BugReports https://github.com/mshin77/a11yviz/issues

VignetteBuilder knitr, quarto

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Repository https://mshin77.r-universe.dev

Date/Publication 2026-05-23 17:42:47 UTC

RemoteUrl https://github.com/mshin77/a11yviz

RemoteRef HEAD

RemoteSha 05e4d73911e08009a8af75d0cae5510949ee1eaf

Contents

a11y_alpha_presets *Alpha presets for chart layers*

Description

Returns a named numeric vector of alpha values for common chart roles. Use them when setting `alpha =` in `ggplot2::geom_*()` or plotly traces. Alpha lowers the effective contrast a viewer sees, so verify the composited contrast with `a11y_check_palette()` when the choice matters.

Usage

```
a11y_alpha_presets()
```

Value

Named numeric vector with elements `raw_points`, `overlay_point`, `labels`, `fill`, `ci_ribbon`, `ci_band`.

Examples

```
a11y_alpha_presets()
a11y_alpha_presets()[["overlay_point"]]
```

a11y_alt_template *Generate a deterministic alt-text template for a plot*

Description

Introspects a `ggplot` or `plotly` object and emits a sentence scaffold with chart type, axis labels, ranges, and group counts pre-filled. A bracketed placeholder marks where the substantive trend description belongs. Pass the edited string to `a11y_alt_text()` to attach it.

Usage

```
a11y_alt_template(p)
```

Arguments

p	A `ggplot` or `plotly` object.

Value

Character scalar.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  p <- ggplot(mtcars, aes(wt, mpg, color = factor(cyl))) + geom_point()
  a11y_alt_template(p)
}
```

`a11y_alt_text` *Add alt text to a plot*

Description

For plotly objects, sets the figure-level `aria-label` and a hidden description div. For ggplot, attaches the alt text as an attribute that Quarto renders as `<img alt="...">`.

Usage

```
a11y_alt_text(p, text)
```

Arguments

`p`	A plotly or ggplot object.
`text`	Character. Concise description for screen readers.

Value

The object with alt text attached.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  p <- ggplot(mtcars, aes(mpg, wt)) + geom_point()
  a11y_alt_text(p, "Scatter of car weight against fuel economy.")
}
```

a11y_announce *Announce a status message to assistive technology*

Description

Pair with a CSS rule that defines `.screen-reader-only` (see `a11y_css()`).

Usage

```
a11y_announce(text)
```

Arguments

`text`	Character. Message to announce.

Value

Character scalar containing HTML.

See Also

`a11y_css()`

Examples

```
a11y_announce("Loading results, please wait")
```

a11y_aria_label *Build an ARIA label string*

Description

Generates a semantic `aria-label` value combining an action, element type, and optional context. Intended for Shiny UI elements (buttons, inputs).

Usage

```
a11y_aria_label(element_type, action, context = NULL)
```

Arguments

`element_type`	Character. Element type, e.g. `"button"`, `"input"`.
`action`	Character. Action verb, e.g. `"analyze"`, `"download"`.
`context`	Character or `NULL`. Optional disambiguating context.

Value

Character scalar suitable for `aria-label`.

Examples

```
a11y_aria_label("button", "analyze", "readability")
a11y_aria_label("input", "search")
```

`a11y_audit` *Chart + document accessibility audit*

Description

Union of `a11y_audit_chart()` and `a11y_audit_doc()`. Prefer the split functions when one scope is enough.

Usage

```
a11y_audit(p, level = "AA")
```

Arguments

`p`	A `plotly` or `ggplot` object.
`level`	`"AA"` or `"AAA"`.

Value

Data frame with columns `criterion`, `check`, `status`, `note`. Join to `a11y_rubric()` for principle, guideline, and threshold.

`a11y_audit_actionable` *Actionable rows from an audit*

Description

Filters an audit data frame to rows with status `todo` or `ok` – the decisions a chart author has to make. Drops rows handled by helpers, rows that belong to the host document, and rows marked `manual` / `n/a`.

Usage

```
a11y_audit_actionable(audit)
```

Arguments

`audit`	Output of `a11y_audit()`, `a11y_audit_chart()`, or `a11y_audit_doc()`.

Value

Data frame with the same columns as `audit`, filtered.

---

a11y_audit_chart *Chart-only accessibility audit*

Description

Returns the chart-relevant rows from the WCAG 2.1 audit: alt text, redundant group encoding, text contrast, text size, non-text contrast, hover/focus, and (at AAA) stronger text contrast.

Usage

```
a11y_audit_chart(p, level = "AA")
```

Arguments

`p`	A `plotly` or `ggplot` object.
`level`	`"AA"` or `"AAA"`.

Value

Data frame with columns `criterion`, `check`, `status`, `note`.

---

a11y_audit_doc *Document-level accessibility audit*

Description

Returns the host-page rows from the WCAG 2.1 audit: heading hierarchy, text resizing, reflow, body text spacing, and visible keyboard focus.

Usage

```
a11y_audit_doc(level = "AA")
```

Arguments

`level`	`"AA"` or `"AAA"`. Doc-level rows are identical for both.

Value

Data frame with columns `criterion`, `check`, `status`, `note`.

a11y_audit_summary *One-line summary of an audit*

Description

Returns a sentence counting how many checks need a decision, how many pass, and how many are already handled.

Usage

```
a11y_audit_summary(audit)
```

Arguments

audit	Output of a11y_audit(), a11y_audit_chart(), or a11y_audit_doc().

Value

Length-1 character vector.

a11y_check_alt_text *Check alt-text presence and length (WCAG 1.1.1)*

Description

Validates a candidate alt-text string for informative content. Decorative elements should pass decorative = TRUE and use alt="" in markup. Companion to a11y_alt_text(), which *attaches* alt text to a plot object; this function *validates* a string before attaching.

Usage

```
a11y_check_alt_text(
  alt_text,
  element_type = "image",
  decorative = FALSE,
  min_length = 10
)
```

Arguments

alt_text	Character or NULL. Candidate alt text.
element_type	Character. Element type for the warning, e.g. "plot".
decorative	Logical. If TRUE, empty alt text is allowed.
min_length	Integer. Minimum length for informative alt text (default 10).

Value

TRUE if valid; FALSE with a warning otherwise.

See Also

a11y_alt_text()

Examples

```
a11y_check_alt_text("Bar chart showing word frequency", "plot")
suppressWarnings(a11y_check_alt_text("", "plot"))
a11y_check_alt_text("", "icon", decorative = TRUE)
```

---

a11y_check_headings *Check Markdown / Quarto / HTML heading hierarchy and labels*

Description

Scans a file for three heading defects: (a) skipped heading levels (e.g., ## followed by ####), (b) empty headings, (c) non-descriptive headings (text shorter than min_chars). These violate WCAG 2.1 Success Criteria 1.3.1 (Info and Relationships), 2.4.6 (Headings and Labels), and 2.4.10 (Section Headings).

Usage

```
a11y_check_headings(path, min_chars = 3)
```

Arguments

path	Path to a .md, .qmd, .Rmd, or .html file.
min_chars	Minimum heading text length (after trimming) considered descriptive. Default 3.

Value

Data frame with one row per issue, columns line, level, text, issue. Empty data frame if no issues.

Examples

```
## Not run:
  a11y_check_headings("paper.qmd")

## End(Not run)
```

a11y_check_overlap *Scatter overlap check (WCAG Success Criterion 1.3.1)*

Description

Bins point coordinates from `geom_point` layers to a `bins` x `bins` grid in data space and reports the fraction of points that share a cell with another point. Treat as a render-resolution proxy for Success Criterion 1.3.1 (information must remain perceivable). When overlap is non-zero, alpha may be considered, but composited contrast must clear 3:1 per Success Criterion 1.4.11.

Usage

```
a11y_check_overlap(p, bins = 100)
```

Arguments

`p`	A `ggplot` object with at least one `geom_point` layer.
`bins`	Grid resolution per axis (default 100).

Value

List with `total`, `obscured`, `fraction`, `recommendation`.

a11y_check_palette *Check a palette against WCAG contrast thresholds*

Description

Computes the contrast ratio of every color against a reference background and reports whether each meets the WCAG threshold (4.5:1 for AA, 7:1 for AAA). When `alpha < 1`, contrast is computed against the alpha-composited rendered color – the color the viewer actually sees. Works on any palette – `viridis`, `RColorBrewer`, plotly's discrete sequences, or a custom hex vector.

Usage

```
a11y_check_palette(colors, bg = "#ffffff", level = "AA", alpha = 1)
```

Arguments

`colors`	Character vector of hex codes.
`bg`	Reference background hex(es). Pass a single value (e.g. `"#ffffff"`) or a vector (e.g. `c("#ffffff", "#1a1a1a")`) to verify against multiple backgrounds. Returns one row per (color, bg) pair.
`level`	`"AA"` (4.5:1) or `"AAA"` (7:1). Use `"AA-large"` (3:1) for non-text elements such as data marks, axis lines, and focus rings.

alpha — Opacity in [0, 1]. 1 (default) checks the raw color. Values below 1 composite each color over bg first, then check the rendered result – useful when chart geoms use alpha < 1.

Value

Data frame with columns color, bg, alpha, rendered, ratio, status.

Examples

```
a11y_check_palette(c("#000000", "#cccccc", "#ff0000"))
a11y_check_palette(c("#0072B2"), bg = c("#ffffff", "#1a1a1a"), level = "AA-large")
a11y_check_palette(c("#0072B2", "#D55E00"), alpha = 0.7, level = "AA-large")
```

---

a11y_check_palette_size

Flag categorical palettes above the recommended maximum

Description

Distinguishability drops fast above seven categories even in CVD-safe palettes; consider faceting, aggregation, or a sequential / ordinal encoding instead.

Usage

```
a11y_check_palette_size(n, max = 7)
```

Arguments

n — Integer; number of categories.

max — Integer; recommended maximum. Default 7.

Value

Named list with n, max, status ("ok" / "todo"), note.

Examples

```
a11y_check_palette_size(5)
a11y_check_palette_size(12)
```

a11y_check_readability

Estimate reading level of prose

Description

Computes Flesch-Kincaid Grade Level and Flesch Reading Ease for the supplied text. Pure base R with a syllable-count heuristic; no external dependencies. Maps to WCAG 2.1 Success Criterion 3.1.5 (Reading Level, AAA).

Usage

```
a11y_check_readability(text)
```

Arguments

`text`	Character vector, single string, or path to a `.md`, `.qmd`, `.Rmd`, or `.txt` file.

Value

Data frame with one row and columns `sentences`, `words`, `syllables`, `flesch_kincaid_grade`, `flesch_reading_ease`.

Examples

```
a11y_check_readability("The cat sat on the mat. The dog ran away.")
```

a11y_check_separability

Flag color pairs below the WCAG 2.1 Success Criterion 1.4.11 contrast threshold

Description

For every pair of colors in the palette, computes the WCAG relative-luminance contrast ratio – the same formula used by `a11y_check_palette()`. Pairs below `min_ratio` (default `3.0`, the WCAG 2.1 Success Criterion 1.4.11 "Non-text Contrast / Graphical Objects" threshold) are flagged. Two data marks of similar colors may fail this – viewers cannot tell them apart.

Usage

```
a11y_check_separability(colors, min_ratio = 3)
```

Arguments

`colors`	Character vector of hex codes.
`min_ratio`	Numeric threshold; default `3.0` per WCAG Success Criterion 1.4.11.

Value

Data frame with columns `from`, `to`, `ratio`, `status`.

Examples

```
a11y_check_separability(c("#1B9E77", "#D95F02", "#7570B3"))
```

---

a11y_check_tabindex *Check that a tabindex value follows WCAG 2.1.1*

Description

Positive tabindex values above the natural document flow create unpredictable keyboard navigation. WCAG 2.1.1 (Keyboard) recommends `tabindex = 0` (natural order) or `tabindex = -1` (focusable but skipped).

Usage

```
a11y_check_tabindex(tabindex = 0)
```

Arguments

`tabindex`	Numeric scalar.

Value

`TRUE` if valid; `FALSE` with a warning if non-numeric or > 100.

Examples

```
a11y_check_tabindex(0)
a11y_check_tabindex(-1)
suppressWarnings(a11y_check_tabindex(999))
```

---

a11y_css *Path to the accessible CSS*

Description

Returns the filesystem path to `a11yviz.css`, an accessible stylesheet that makes plotly, DT, DiagrammeR, and static plot images theme-aware and applies WCAG-aligned defaults for contrast, focus, and text spacing. Use it in Quarto via `css:`, in Shiny via `tags$link()`, or copy it into a project's `www/`.

Usage

```
a11y_css(mode = c("default", "shiny"))
```

Arguments

mode	"default" returns the base CSS path; "shiny" also appends a11yviz-shiny.css (skip-link, screen-reader-only text, reduced-motion, high-contrast rules) in load order.

Value

Character path (default) or character vector of paths (shiny).

Examples

```
basename(a11y_css())
length(a11y_css("shiny"))
```

---

a11y_css_contents *Contents of the accessible CSS*

Description

Reads the bundled CSS file(s) returned by a11y_css() and concatenates them as a single string. Useful for embedding inline in Quarto via <style> tags or in Shiny via tags$style().

Usage

```
a11y_css_contents(mode = c("default", "shiny"))
```

Arguments

mode	"default" returns the base CSS path; "shiny" also appends a11yviz-shiny.css (skip-link, screen-reader-only text, reduced-motion, high-contrast rules) in load order.

Value

Character scalar of CSS source.

Examples

```
nchar(a11y_css_contents())
```

a11y_describe *Generate alt text via a user-supplied LLM backend*

Description

Extracts deterministic plot context (chart type, axes, ranges, group counts) and passes it to the user's backend function, which calls any LLM provider and returns the alt-text string. Result is attached to the plot via `a11y_alt_text()` when `attach = TRUE`.

Usage

```
a11y_describe(p, backend, attach = TRUE)
```

Arguments

p	A `ggplot` or `plotly` object.
backend	A function `function(context) -> character(1)`. Receives a list with fields `chart_type`, `title`, `x`, `y`, `color`, `n_observations`. Returns a single string.
attach	If `TRUE` (default), attach via `a11y_alt_text()`. Otherwise return the string.

Details

The package itself depends on no LLM SDK; the caller supplies the transport. See examples for OpenAI, Gemini, and Ollama backends.

Value

The plot with alt text attached, or the string itself.

Examples

```
## Not run:
  openai_backend <- function(context) {
    prompt <- paste(
      "Write one-sentence WCAG 1.1.1 alt text.",
      "State chart type, axes, and key trend."
    )
    auth <- paste("Bearer", Sys.getenv("OPENAI_API_KEY"))
    res <- httr::POST(
      "https://api.openai.com/v1/chat/completions",
      httr::add_headers(Authorization = auth),
      body = list(
        model = "gpt-4o-mini",
        messages = list(
          list(role = "system", content = prompt),
          list(role = "user",
               content = jsonlite::toJSON(context, auto_unbox = TRUE))
        )
```

```
    ),
    encode = "json"
  )
  httr::content(res)$choices[[1]]$message$content
}
p |> a11y_describe(backend = openai_backend)

## End(Not run)
```

---

a11y_ggplotly *Convert ggplot to accessible plotly for supplemental online output*

Description

Wraps `plotly::ggplotly()` with `a11y_layout()` and forwards alt text attached via `a11y_alt_text()`. Strips redundant chart titles. Reserve for the interactive supplement; a static ggplot with alt text is the more accessible default.

Usage

```
a11y_ggplotly(
  gg,
  level = "AA",
  palette = NULL,
  alt = NULL,
  tooltip = c("x", "y"),
  strip_title = TRUE,
  ...
)
```

Arguments

`gg`	A `ggplot` object.
`level`	WCAG contrast level: `"AA"` (default) or `"AAA"`.
`palette`	Optional palette name applied as plotly's `colorway`. `NULL` (default) keeps the ggplot's existing scale.
`alt`	Alt-text override. When `NULL`, inherited from `attr(gg, "a11y_alt")`.
`tooltip`	Aesthetic(s) shown in hover. Default `c("x", "y")`.
`strip_title`	Logical; when `TRUE` (default) drops `title` and `subtitle` so the host page heading is authoritative.
`...`	Forwarded to `plotly::ggplotly()`.

Value

A plotly object.

Examples

```
if (requireNamespace("ggplot2",        quietly = TRUE) &&
    requireNamespace("plotly",         quietly = TRUE) &&
    requireNamespace("palmerpenguins", quietly = TRUE)) {
  library(ggplot2)
  p <- ggplot(na.omit(palmerpenguins::penguins),
              aes(flipper_length_mm, body_mass_g,
                  color = species, shape = species)) +
    geom_point() +
    scale_color_a11y("dark2_8") +
    theme_a11y("AA")
  p <- a11y_alt_text(p, "Penguin body mass vs flipper length by species.")
  pl <- a11y_ggplotly(p)
  inherits(pl, "plotly")
}
```

---

a11y_layout *Apply accessible layout to a plotly figure*

Description

Sets fonts, axis styling, hover-label colors, legend position, and an accessible categorical `colorway`. Targets WCAG 2.1 contrast (1.4.3, 1.4.6, 1.4.11) and resizability (1.4.4). Font sizes are package defaults, not WCAG-mandated minimums. Returns the plotly object so it can stay in a `|>` chain.

Usage

```
a11y_layout(p, level = "AA", palette = "dark2_8")
```

Arguments

`p`	A plotly object (from `plotly::plot_ly` or `plotly::ggplotly`).
`level`	`"AA"` or `"AAA"`.
`palette`	Discrete palette name applied as plotly's `colorway`. See `a11y_palette_list()`. Pass `NULL` to leave plotly's default colors unchanged.

Value

Modified plotly object.

a11y_minimum *Layer minimum accessibility onto a chart*

Description

Adds only the non-destructive moves: attaches alt text, and raises the base text size to the WCAG minimum only when the current size is below it. Palette, theme, legend, and geom aesthetics stay intact. Suitable for retrofitting charts with an existing visual; `theme_a11y()` and `scale_a11y()` are the greenfield path. Pair with `a11y_audit_chart()` to surface remaining gaps (color-only encoding, missing alt, hover styling).

Usage

```
a11y_minimum(p, alt = NULL, level = "AA")
```

Arguments

`p`	A `ggplot` or `plotly` object.
`alt`	Optional alt text attached via `a11y_alt_text()`.
`level`	`"AA"` (12 pt minimum) or `"AAA"` (14 pt minimum).

Value

The chart with alt text attached (if supplied) and base text size raised to the level threshold when it was below.

a11y_palette *Discrete color palette (categorical)*

Description

Returns hex codes for a color-vision-aware categorical palette. Most palettes are runtime wrappers around `RColorBrewer::brewer.pal()` so the authoritative colors stay in their source package.

Usage

```
a11y_palette(name = "dark2_8", n = NULL, bg = NULL)
```

Arguments

`name`	Discrete palette name. Built-in: `"dark2_8"` (default, RColorBrewer Dark2), `"set2_8"` (RColorBrewer Set2), `"paired_12"` (RColorBrewer Paired), `"aaa_5"` (custom AAA-on-white set).
`n`	Optional number of colors. Defaults to the palette's full size (truncates from the start when smaller).
`bg`	Plot background context. One of `NULL` (default – no check), `"white"`, or `"dark"`. When set, the function warns if the palette's `safe_on` tag does not match.

Value

Character vector of hex codes (e.g., "#1B9E77"). For sequential gradients, see `a11y_palette_seq()`.

Examples

```
a11y_palette("dark2_8")
a11y_palette("aaa_5")
a11y_palette("dark2_8", n = 4)
```

`a11y_palette_div` *Diverging palette*

Description

Returns the low/mid/high anchor colors for a diverging gradient. Most palettes resolve at runtime from `RColorBrewer`. `*_dual` variants pick mid-saturation Brewer positions whose endpoints clear non-text 3:1 on both white and `#1a1a1a` dark backgrounds. `coolwarm_aaa` is a custom built-in whose endpoints both clear AAA on white only.

Usage

```
a11y_palette_div(name = "rdbu")
```

Arguments

`name`	One of `"rdbu"` (default), `"puor"`, `"brbg"`, `"rdbu_dual"`, `"puor_dual"`, `"brbg_dual"`, `"coolwarm_aaa"`.

Value

Named list with elements `low`, `mid`, `high` (hex codes for the diverging anchors). To materialize an N-step gradient, pass the list to a color interpolator (e.g., `grDevices::colorRampPalette()`).

Examples

```
a11y_palette_div("rdbu")
a11y_palette_div("rdbu_dual")
```

a11y_palette_info *Discrete palette metadata*

Description

Returns the source spec, resolved colors, and WCAG metadata for a discrete palette. For continuous palettes, see a11y_palette_div() and a11y_palette_seq().

Usage

```
a11y_palette_info(name = "dark2_8")
```

Arguments

name	Discrete palette name. Built-in: "dark2_8" (default, RColorBrewer Dark2), "set2_8" (RColorBrewer Set2), "paired_12" (RColorBrewer Paired), "aaa_5" (custom AAA-on-white set).

Value

Named list with name, colors, safe_on, purpose, notes, plus the source spec fields.

Examples

```
a11y_palette_info("aaa_5")
```

a11y_palette_list *List available palettes*

Description

Lists discrete, diverging, and sequential palettes in one data frame.

Usage

```
a11y_palette_list(type = NULL)
```

Arguments

type	Optional filter: "discrete", "diverging", or "sequential". NULL (default) returns all.

Value

Data frame with columns name, type, source, n, safe_on, purpose. n is NA for continuous palettes. For the notes field of a single palette, call a11y_palette_info().

Examples

```
a11y_palette_list()
a11y_palette_list(type = "diverging")
```

`a11y_palette_seq` *Sequential continuous palette*

Description

Returns a viridisLite spec (option, begin, end, direction) for a sequential gradient. Use directly with `ggplot2::scale_*_viridis_c()` or via `scale_fill_a11y_seq()`. Pass `n =` to materialize hex codes.

Usage

```
a11y_palette_seq(name = "cividis", n = NULL)
```

Arguments

`name`	One of `"cividis"` (default), `"viridis"`, `"plasma"`.
`n`	Optional integer. If supplied, returns `n` hex codes from the gradient instead of the spec.

Value

Named list with elements `option`, `begin`, `end`, `direction` – a viridisLite specification, NOT a color vector. Pass `n =` to materialize hex codes.

Examples

```
a11y_palette_seq("cividis")
a11y_palette_seq("viridis", n = 7)
```

`a11y_plotly_sequences` *Audit plotly's built-in discrete color sequences*

Description

Plotly ships several discrete categorical sequences (from plotly.js). This helper computes WCAG contrast statistics for each one against a reference background so users can pick a sequence that passes their contrast bar.

Usage

```
a11y_plotly_sequences(bg = "#ffffff", level = "AA")
```

Arguments

bg	Reference background hex (default "#ffffff").
level	"AA" (4.5:1) or "AAA" (7:1).

Value

Data frame with one row per sequence: name, n, min_ratio, median_ratio, n_pass, pct_pass.

Examples

```
a11y_plotly_sequences()
a11y_plotly_sequences(level = "AAA")
```

---

a11y_rubric *WCAG 2.1 rubric for the success criteria a11yviz addresses*

Description

Returns the chart-relevant subset of WCAG 2.1 success criteria, with the AA / AAA threshold and the a11yviz function that addresses it. Joins to a11y_audit() on the criterion column.

Usage

```
a11y_rubric(level = NULL)
```

Arguments

level	Optional filter: "AA" or "AAA". NULL (default) returns all rows.

Value

Data frame with columns criterion, name, level, threshold_aa, threshold_aaa, a11yviz_function. Pass any criterion value to a11y_wcag_url() for the spec link.

Examples

```
a11y_rubric()
a11y_rubric(level = "AAA")
```

a11y_show_palette *Visualize a palette with WCAG contrast overlay*

Description

Renders the swatches of a built-in palette and overlays each color's contrast ratio against bg, plus a pass/fail label for `level`.

Usage

```
a11y_show_palette(name = "dark2_8", bg = "#ffffff", level = "AA")
```

Arguments

`name`	Discrete palette name. See `a11y_palette_list()`.
`bg`	Reference background hex (default `"#ffffff"`).
`level`	`"AA"` or `"AAA"`.

Value

A ggplot object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  a11y_show_palette("dark2_8")
}
```

a11y_text_spacing_ratios *WCAG 1.4.12 text-spacing ratios (reference data)*

Description

Returns the spacing ratios specified in WCAG Success Criterion 1.4.12 (Text Spacing, Level AA). All values are multiples of the font size.

Usage

```
a11y_text_spacing_ratios()
```

Value

Named numeric vector with `line_height`, `paragraph`, `letter`, `word`.

Examples

```
a11y_text_spacing_ratios()
a11y_text_spacing_ratios()[["line_height"]]
```

`a11y_wcag_url` *WCAG 2.1 specification URL for a success criterion*

Description

Returns a deep link to the W3C WCAG 2.1 specification entry for one or more success criteria.

Usage

```
a11y_wcag_url(criterion)
```

Arguments

`criterion`	Character vector of success-criterion numbers (e.g., `"1.4.3"`). Recognised values are the chart-relevant subset returned by `a11y_rubric()`.

Value

Character vector of URLs the same length as `criterion`. Returns the spec root URL for unrecognised values.

Examples

```
a11y_wcag_url("1.4.3")
a11y_wcag_url(c("1.1.1", "2.4.7", "4.1.3"))
```

`make_a11y` *One-shot accessibility wrapper*

Description

Applies the most common transforms in one call: theme plus color and fill palettes for ggplot, or layout plus palette for plotly.

Usage

```
make_a11y(p, level = "AA", palette = "dark2_8", alt = NULL)
```

Arguments

p	A ggplot or plotly object.
level	"AA" or "AAA".
palette	Categorical palette name passed to a11y_palette().
alt	Optional alt-text string.

Value

The transformed object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  p <- ggplot(mpg, aes(class, fill = drv)) + geom_bar()
  make_a11y(p, palette = "dark2_8", alt = "Vehicle classes by drivetrain.")
}
```

---

run_app *Launch the local accessibility playground*

Description

Opens a Shiny app comparing default ggplot2 output against the accessible theme, palette, and alt-text helpers, with a per-criterion WCAG audit. Requires shiny, bslib, ggplot2, and DT; missing packages are flagged at launch.

Usage

```
run_app(host = "127.0.0.1", port = 8000, launch_browser = TRUE, ...)
```

Arguments

host	Network host (default "127.0.0.1").
port	Port (default 8000).
launch_browser	Open the default browser (default TRUE).
...	Passed to shiny::runApp().

Value

Invisibly, the return value of shiny::runApp().

Examples

```
if (interactive()) run_app()
```

scale_a11y *Accessible discrete color and fill scales*

Description

Categorical palettes that meet WCAG 1.4.1 (Use of Color) by being distinguishable to viewers with color vision differences.

Usage

```
scale_color_a11y(palette = NULL, level = "AA", ...)

scale_fill_a11y(palette = NULL, level = "AA", ...)
```

Arguments

palette	One of "dark2_8" (default), "set2_8", "paired_12", "aaa_5". See a11y_palette_list().
level	WCAG contrast level. "AAA" switches the default palette to "aaa_5" (deep, AAA-on-white set). Ignored if palette is set explicitly.
...	Passed to ggplot2::discrete_scale.

Value

A ggplot2 scale.

Examples

```
if (requireNamespace("ggplot2",        quietly = TRUE) &&
    requireNamespace("palmerpenguins", quietly = TRUE)) {
  library(ggplot2)
  pg <- na.omit(palmerpenguins::penguins)
  ggplot(pg, aes(flipper_length_mm, body_mass_g, color = species)) +
    geom_point() + scale_color_a11y()
  ggplot(pg, aes(species, fill = island)) +
    geom_bar() + scale_fill_a11y("set2_8")
}
```

scale_a11y_div *Accessible diverging color and fill scales*

Description

ggplot2 scales for diverging gradients with anchor colors sourced from a11y_palette_div().

Usage

```
scale_color_a11y_div(palette = "rdbu", ...)

scale_fill_a11y_div(palette = "rdbu", ...)
```

Arguments

palette	One of "rdbu" (default), "puor", "brbg", "coolwarm_aaa".
...	Passed to ggplot2::scale_*_gradient2().

Value

A ggplot2 scale.

---

scale_a11y_seq *Accessible sequential continuous color and fill scales*

Description

ggplot2 scales for sequential viridisLite gradients sourced from a11y_palette_seq(). The default "cividis" remains readable in greyscale and spans the full lightness range.

Usage

```
scale_color_a11y_seq(palette = "cividis", ...)

scale_fill_a11y_seq(palette = "cividis", ...)
```

Arguments

palette	One of "cividis" (default), "viridis", "plasma".
...	Passed to ggplot2::scale_*_viridis_c().

Value

A ggplot2 scale.

theme_a11y *Accessible ggplot2 theme*

Description

Adds a ggplot2::theme that applies WCAG 2.1 contrast settings plus the package's recommended font sizes. Compose with + like any other theme. Title, axis title, and legend sit at the body floor (12 pt AA / 14 pt AAA); axis tick text drops 2 pt below.

Usage

```
theme_a11y(level = "AA", base_family = "", dark = FALSE)
```

Arguments

level	WCAG contrast level: "AA" (default) or "AAA". The level controls contrast targets and the package's default font sizes; only the contrast targets are WCAG-defined.
base_family	Font family. Defaults to system sans.
dark	Logical; if TRUE, use a dark-mode palette appropriate for darkly-style themes.

Value

A theme object.

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(mtcars, aes(mpg, wt)) + geom_point() + theme_a11y()
}
```

Index