Package: BFHcharts (via r-universe)

November 3, 2026

Title SPC Visualization for Healthcare Quality Improvement

Version 0.31.0

Description A modern R package for creating Statistical Process Control (SPC) charts in healthcare settings. Built on ggplot2 and qicharts2, BFHcharts provides beautiful, publication-ready SPC visualizations with configurable themes and multi-organizational branding support. Inspired by BBC's bbplot design philosophy.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

SystemRequirements Quarto CLI (>= 1.4.0) for PDF export via Typst templates

URL https://github.com/johanreventlow/BFHcharts

BugReports https://github.com/johanreventlow/BFHcharts/issues

Depends R (>= 4.1.0)

Imports BFHtheme (>= 0.5.5), ggplot2 (>= 3.5.0), qicharts2 (>= 0.7.0), scales (>= 1.2.0), lubridate (>= 1.9.0), dplyr (>= 1.1.0), purrr (>= 1.0.0), stringr (>= 1.5.0), tibble (>= 3.2.0), yaml, lemon, marquee (>= 0.1.0), grid, systemfonts, rlang, svglite

Suggests commonmark (>= 1.9), xml2, testthat (>= 3.1.7), vdiffr, covr, pdftools (>= 3.3.0), qpdf, withr, bench, BFHllm, jsonlite, knitr, pbcharts, rmarkdown

VignetteBuilder knitr

Remotes johanreventlow/BFHtheme@v0.5.5, johanreventlow/BFHllm@v0.2.0, anhoej/pbcharts@d37a8ec1864528ece1f5cd60ac152d06c013515f

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libfontconfig1-dev libfreetype6-dev libfribidi-dev libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev

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

Date/Publication 2026-11-02 21:34:17 UTC

RemoteUrl https://github.com/johanreventlow/BFHcharts

RemoteRef HEAD

RemoteSha 04b8fa0f85cd43b8d48f3db821f7d19804ba4490

Contents

.BFHTHEME_MIN_VERSION . . . 3
as.list.bfh_spc_analysis . . . 3
bfh_analyse . . . 4
bfh_apply_date_axis . . . 5
bfh_build_analysis_context . . . 6
bfh_create_export_session . . . 7
bfh_create_typst_document . . . 9
bfh_export_batch_pdf . . . 11
bfh_export_figure_pdf . . . 12
bfh_export_pdf . . . 15
bfh_export_png . . . 21
bfh_extract_spc_stats . . . 23
bfh_figure_panel . . . 25
bfh_generate_analysis . . . 28
bfh_generate_details . . . 31
bfh_get_plot . . . 33
bfh_merge_metadata . . . 33
bfh_prune_page_cache . . . 35
bfh_qic . . . 36
bfh_qic_result . . . 49
bfh_qic_stats . . . 50
bfh_render_analysis . . . 52
bfh_stage_figure_page . . . 53
bfh_stage_pdf_page . . . 55
bfh_subsample_label_indices . . . 57
close.bfh_export_session . . . 58
format.bfh_spc_analysis . . . 59
is_bfh_qic_result . . . 59
is_bfh_spc_analysis . . . 60
new_bfh_qic_result . . . 60
plot.bfh_qic_result . . . 62
print.bfh_export_session . . . 63
print.bfh_qic_result . . . 63
print.bfh_qic_stats . . . 64
print.bfh_spc_analysis . . . 64

Index **65**

.BFHTHEME_MIN_VERSION *Minimum required version of BFHtheme*

Description

Single source of truth for the BFHtheme lower-bound used in dep-guards (utils_dep_guards.R), .onAttach (zzz.R), and package-level docs (BFHcharts-package.R). Must match the Imports: lower-bound in DESCRIPTION.

Usage

```
.BFHTHEME_MIN_VERSION
```

as.list.bfh_spc_analysis *Convert bfh_spc_analysis to Plain List*

Description

Strips the S3-class and returns a plain named list suitable for JSON-serialization or downstream-tools that do not understand the class. The structure is preserved verbatim; no field-renaming or filtering occurs.

Usage

```
## S3 method for class 'bfh_spc_analysis'
as.list(x, ...)
```

Arguments

x	A bfh_spc_analysis object.
...	Ignored.

Value

Named list.

Examples

```
## Not run:
analysis <- bfh_analyse(result)
flat <- as.list(analysis)
json <- jsonlite::toJSON(flat, auto_unbox = TRUE)

## End(Not run)
```

`bfh_analyse` *Compose a Structured SPC Analysis Object*

Description

Returnerer struktureret `bfh_spc_analysis`-objekt med features, conclusions (i18n-keys), caveats, suggested_actions, og render-context. Det er primaer canonical-output for struktureret SPC- analyse; rentekst-rendering sker via `bfh_render_analysis()`.

Usage

```
bfh_analyse(x, metadata = list(), language = c("da", "en"))
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`.
`metadata`	Optional named list. Kan indeholde: • `data_definition`: Description of what the data represents • `target`: Target value (numeric or operator-prefixed character) • `direction`: Optional `"higher_better"` / `"lower_better"` / `"neutral"` override for direction-akse (Slice 4) • `analysis_date`: Date pinned for determinism (Phase 0.4) • `hospital`, `department`: passes through to context
`language`	Character: `"da"` (default) or `"en"`. Stored in object for downstream render-defaults; SHALL NOT cause text resolution at this stage.

Details

Key-only model: `conclusions$*_key`, `caveats$*` og `suggested_actions` indeholder i18n-noegler, ej resolverede tekst- strenge. Tekst-resolution sker udelukkende i `bfh_render_analysis()` via `texts_loader`. Dette bevarer language-neutralt JSON-eksport-output + tillader audit-replay paa anden sprog uden re-extraction.

Value

Object of class `bfh_spc_analysis`. See ADR-005 for schema.

See Also

Pipeline position: second of three layers.

- `bfh_qic` produces the `bfh_qic_result` input.
- `bfh_build_analysis_context` is a lower-level helper called internally (via `bfh_extract_spc_features()`) to build the raw context consumed by this function.
- `bfh_render_analysis` takes the returned `bfh_spc_analysis` object and renders it to character output.

- bfh_generate_analysis provides a single-step alternative that combines analysis and rendering in one call.
- bfh_generate_details generates the companion detail row string (period, averages, latest values) from the same bfh_qic_result input.

Examples

```
## Not run:
result <- bfh_qic(data, x = month, y = value, chart_type = "i")
analysis <- bfh_analyse(result, metadata = list(target = ">= 90%"))
print(analysis)

# Render to text:
text <- bfh_render_analysis(analysis, max_chars = 375)

# Audit-trail via JSON:
jsonlite::toJSON(as.list(analysis), auto_unbox = TRUE, Date = "ISO8601")

## End(Not run)
```

---

bfh_apply_date_axis *Date Axis Like the SPC Charts*

Description

Adds the x-axis BFHcharts uses on SPC charts to any ggplot with a date x-axis: month labels (and the year at the first month and at year shifts) below the axis, and ISO week numbers ("UGE 08", then every few weeks) just above it. Breaks, label cadence and locale follow bfh_qic exactly, so figures and SPC charts in the same report read the same way.

Usage

```
bfh_apply_date_axis(plot, x, language = "da")
```

Arguments

plot	A ggplot object.
x	The x values the axis should cover (Date or POSIXct), typically all dates in the plot.
language	Language of the week prefix and month names ("da" or "en").

Details

The plot's x aesthetics must be POSIXct (convert Date with as.POSIXct(x, tz = "UTC")), because a datetime scale is added. Set x-limits with coord_cartesian(xlim =), not on the scale. Add the theme (e.g. BFHtheme::theme_bfh()) before or after - tick styling comes from the theme.

Value

The ggplot with a datetime x-scale and week-number labels.

See Also

Other export-functions: `bfh_export_batch_pdf()`, `bfh_export_figure_pdf()`, `bfh_figure_panel()`, `bfh_prune_page_cache()`, `bfh_stage_figure_page()`, `bfh_stage_pdf_page()`

Examples

```
d <- data.frame(
  x = as.POSIXct(seq(as.Date("2026-01-05"), by = "week", length.out = 30), tz = "UTC"),
  y = rpois(30, 20)
)
p <- ggplot2::ggplot(d, ggplot2::aes(x, y)) + ggplot2::geom_col()
bfh_apply_date_axis(p, d$x)
```

`bfh_build_analysis_context` *Build Analysis Context from bfh_qic_result*

Description

Collects all relevant context from a `bfh_qic_result` object for analysis generation. Used internally by `bfh_generate_analysis()`.

Usage

```
bfh_build_analysis_context(x, metadata = list())
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`
`metadata`	Optional list with additional context: • `data_definition`: Description of what the data represents • `target`: Target value for the metric. Accepts either numeric (backward compatible) or character with optional operator prefix (`"<= 2,5"`, `">= 90%"`). Operators are parsed to derive `target_direction`. **Fallback chain:** when `metadata$target` is NULL the function falls back to `x$config$target_text`, then `x$config$target_value`. This means a target supplied to `bfh_qic(target_text = ..., target_value = ...)` automatically reaches the analysis context without callers duplicating it in `metadata`. • `hospital`: Hospital name • `department`: Department name

Value

Named list with complete context including:

- `chart_title`: Chart title from config
- `chart_type`: Chart type (i, p, c, u, etc.)
- `y_axis_unit`: Y-axis unit label
- `n_points`: Number of data points
- `centerline`: Centerline value
- `spc_stats`: SPC statistics from `bfh_extract_spc_stats()`
- `has_signals`: Logical indicating if signals were detected
- `target_value`: Numeric target value (NA if absent). **Percent-target normalization:** when `y_axis_unit == "percent"` and the parsed target value appears to be on the 0-100 scale (display contains `"%"` or `value > 1`), `target_value` is divided by 100 so downstream comparisons are on the same 0-1 proportion scale as `centerline`. `target_display` is always preserved unchanged for user-facing text.
- `target_direction`: `"higher"`, `"lower"`, or `NULL` derived from operator in `metadata$target` (NULL for numeric input)
- `target_display`: Original target string for display purposes
- User-provided metadata fields

Examples

```
## Not run:
result <- bfh_qic(data, x = date, y = value, chart_type = "i")
ctx <- bfh_build_analysis_context(result, metadata = list(hospital = "BFH"))
str(ctx)

## End(Not run)
```

---

bfh_create_export_session

Create a Batch Export Session

Description

Creates a reusable export session that pre-populates Typst template assets once and shares them across multiple `bfh_export_pdf()` calls, eliminating the repeated recursive template directory copy that dominates I/O cost in batch workflows.

Usage

```
bfh_create_export_session(
  font_path = NULL,
  inject_assets = NULL,
  strict_baseline = TRUE
)
```

Arguments

font_path	Optional path to directory containing additional fonts. Applied to all exports in this session. Can be overridden per export via the font_path argument of bfh_export_pdf().
inject_assets	Optional callback function called once after template assets are staged. Receives one argument: the path to the template directory (<session_tmpdir>/bfh-template). Use this to inject fonts or images not bundled in BFHcharts (e.g., proprietary fonts from a private package). If the injected directory contains a fonts/ subdirectory and font_path is NULL, font_path is set automatically.
strict_baseline	Logical. Default for the session: when TRUE (default), every bfh_export_pdf() call inheriting this session errors before render if the result has freeze < MIN_BASELINE_N or any phase shorter than MIN_BASELINE_N (8) data points. Set FALSE to preserve the legacy warning-only behavior across the batch. Per-call strict_baseline on bfh_export_pdf() overrides the session value.

Details

Usage pattern:

```
session <- bfh_create_export_session()
on.exit(close(session))
for (dept in departments) {
  bfh_export_pdf(results[[dept]], paste0(dept, ".pdf"),
                 batch_session = session)
}
```

Limitations:

- Session is single-threaded sequential only - do not share across parallel workers.
- Not compatible with template_path (custom templates).
- Pass inject_assets here, not to individual bfh_export_pdf() calls.

Value

A bfh_export_session object. Close with close(session) to remove the session tmpdir.

Security

inject_assets **is full code execution.** The supplied function runs with the same privileges as the calling R session, with full file-system and network access. It MUST NOT come from user input (Shiny inputs, REST API parameters, configuration files of unknown provenance). **Treat it as trusted-code-only**: pass only code-reviewed, organizationally controlled callbacks.

A runtime heuristic warns when inject_assets originates from .GlobalEnv or a direct child environment. Suppress with options(BFHcharts.allow_globalenv_inject = TRUE) in development.

See bfh_export_pdf for the full security rationale, acceptable/unacceptable sources, and the parallel note for template_path.

Recommended: companion package for proprietary branding. Organizations that need consistent proprietary branding across batch exports should pass a companion-package callback here rather than hardcoding asset paths. For example: `bfh_create_export_session(inject_assets = MyAssetsPkg::inject_my_assets)`. This keeps proprietary fonts and logos out of public BFHcharts distribution while supporting full branding on Posit Connect Cloud, RStudio Connect, and Docker deployments. The callback is still subject to the trusted-code-only contract above. See `bfh_export_pdf` Security section for full details.

See Also

- `bfh_export_pdf()` for single exports and the full security note covering both `inject_assets` and `template_path`

Examples

```
## Not run:
# Create an export session for batch PDF export
session <- bfh_create_export_session(output_dir = tempdir())
# ... add charts ...
close(session)

## End(Not run)
```

`bfh_create_typst_document`

Create Typst Document for SPC Chart

Description

Generates a Typst document (.typ) using BFH hospital template with chart image and metadata. Downstream packages (e.g. biSPCharts) can use this function directly instead of accessing it via `getFromNamespace()`.

Usage

```
bfh_create_typst_document(
  chart_image,
  output,
  metadata,
  spc_stats,
  template = "bfh-diagram",
  template_path = NULL,
  skip_template_copy = FALSE
)
```

Arguments

chart_image	Path to chart image (SVG or PNG)
output	Path for output .typ file
metadata	List with template parameters (hospital, department, title, analysis, details, author, date, data_definition, footer_content). See bfh_merge_metadata for a convenient way to build this list with defaults filled in.
spc_stats	List with SPC statistics (runs_expected, runs_actual, crossings_expected, crossings_actual, outliers_expected, outliers_actual, is_run_chart). See bfh_extract_spc_stats for how to produce this list from a bfh_qic_result object.
template	Template name (default: "bfh-diagram")
template_path	Optional custom template path. When provided, overrides the packaged template (default: NULL uses packaged template)
skip_template_copy	Logical. If TRUE, skip copying the template directory (assumes it is already present in the output directory). Default: FALSE.

Value

Path to created .typ file (invisibly)

See Also

bfh_extract_spc_stats, bfh_merge_metadata

Other utility-functions: bfh_extract_spc_stats(), bfh_generate_details(), bfh_merge_metadata()

Examples

```
## Not run:
result <- bfh_qic(data, x = date, y = value, chart_type = "i")
metadata <- bfh_merge_metadata(
  list(department = "Quality"),
  chart_title = result$config$chart_title
)
spc_stats <- bfh_extract_spc_stats(result)
tmp_dir <- tempdir()
chart_png <- file.path(tmp_dir, "chart.png")
ggplot2::ggsave(chart_png, result$plot, width = 8, height = 5)
typ_file <- bfh_create_typst_document(
  chart_image = chart_png,
  output      = file.path(tmp_dir, "document.typ"),
  metadata    = metadata,
  spc_stats   = spc_stats
)

## End(Not run)
```

bfh_export_batch_pdf *Compile Staged Page Bundles into One Multi-Page PDF*

Description

Reads page bundles staged by bfh_stage_pdf_page from cache_dir and compiles them into a single multi-page PDF through ONE Typst compilation (or a bounded number of chunked compilations, see pages_per_chunk). Each required font face is embedded at most once per compilation – the reason to prefer this over merging single-chart PDFs with external tools, which duplicates embedded font subsets per page.

Usage

```
bfh_export_batch_pdf(
  cache_dir,
  output,
  ids = NULL,
  order_by = c("staged", "ids"),
  pages_per_chunk = NULL,
  skip_invalid = FALSE,
  font_path = NULL,
  ignore_system_fonts = TRUE,
  inject_assets = NULL,
  restrict_template = TRUE,
  template_path = NULL
)
```

Arguments

cache_dir	Directory holding page bundles staged by bfh_stage_pdf_page. Trusted input; see that function's trust-model section.
output	Output PDF path.
ids	Optional character vector of bundle ids (the manifest). When supplied, the document contains exactly these pages. Default NULL compiles every valid bundle in the cache.
order_by	"staged" (default) sorts pages by their staged order keys (ties broken by id) so a partial re-stage cannot reshuffle the document; "ids" orders pages by manifest position (requires ids).
pages_per_chunk	Optional positive integer. When set and exceeded, compilation runs in chunks of at most this many pages and the chunk PDFs are concatenated (requires the qpdf package). Font subsets in the final document are then bounded by the number of chunks. Default NULL compiles everything in a single Typst run.
skip_invalid	Logical. When TRUE, unreadable/incomplete bundles are skipped with a warning listing each skipped id; when FALSE (default) any invalid bundle aborts the export before a PDF is written.

font_path, ignore_system_fonts, inject_assets, restrict_template, template_path

Same semantics as bfh_export_pdf. Template staging and inject_assets run once per batch compile, not once per page.

Value

Invisibly, the output path.

The manifest is the source of truth

A persistent cache accumulates bundles, including pages for charts that have since been retired. Pass ids – derived programmatically from your production data source, not hand-typed – to make the document contain exactly those pages: an id without a bundle is an error (listing every missing id), and bundles not named are silently excluded. Without ids, all valid bundles in the cache are compiled.

Typical workflow:

```
ids <- production_chart_ids()          # caller's source of truth
for (id in ids_needing_restaging) {    # new or corrected charts only
  bfh_stage_pdf_page(result_for(id), cache_dir, id = id)
}
bfh_export_batch_pdf(cache_dir, "combined.pdf", ids = ids)
```

See Also

bfh_stage_pdf_page(), bfh_prune_page_cache(), bfh_export_pdf()

Other export-functions: bfh_apply_date_axis(), bfh_export_figure_pdf(), bfh_figure_panel(), bfh_prune_page_cache(), bfh_stage_figure_page(), bfh_stage_pdf_page()

Examples

```
## Not run:
bfh_export_batch_pdf("~/reports/spc-cache", "samlet.pdf",
  ids = c("afd-a-infektioner", "afd-b-tryksaar")
)

## End(Not run)
```

---

bfh_export_figure_pdf *Export a ggplot Figure to a Branded PDF Page*

Description

Renders an arbitrary ggplot object – a distribution, bar chart or plain time series that is *not* an SPC chart – to a single-page PDF with the BFH template: blue header (hospital, department, title), analysis row, details line, footer and logo. The figure fills the full width of the chart row; the SPC statistics column of bfh_export_pdf is not rendered.

Usage

```
bfh_export_figure_pdf(
  plot,
  output,
  metadata = list(),
  template = "bfh-diagram",
  template_path = NULL,
  restrict_template = TRUE,
  dpi = 150,
  font_path = NULL,
  ignore_system_fonts = TRUE,
  inject_assets = NULL,
  batch_session = NULL,
  panel = NULL
)
```

Arguments

plot	A single ggplot object. Composite plots (class patchwork) are rejected: title stripping and margins would only reach the last sub-plot.
output	Character string specifying the output PDF file path.
metadata	Named list of template metadata. title is **required** (single non-empty character string): it is rendered in the blue header, and the plot's own title and subtitle are stripped so the title only appears once. Optional fields: hospital, department, analysis, details, author, date, footer_content, logo_path. Unlike bfh_export_pdf, details is not auto-generated. data_definition is rendered in the side panel when panel is given; in full-width mode it is **not rendered** and a non-empty value triggers a warning so the text does not vanish silently.
template	Character string specifying template name (default: "bfh-diagram").
template_path	Optional path to a custom Typst template file. Requires restrict_template = FALSE; see bfh_export_pdf for the trust model.
restrict_template	Logical. When TRUE (default), any non-NULL template_path is rejected.
dpi	Resolution passed to ggplot2::ggsave(). Default 150.
font_path	Optional path to a directory with additional fonts, passed as --font-path to the Typst compiler.
ignore_system_fonts	Logical. If TRUE (default), passes --ignore-system-fonts to Typst so only fonts from font_path (or bundled template fonts) are used.
inject_assets	Optional callback receiving the path to the staged template directory, used to add fonts and images before compilation. Cannot be combined with batch_session. Same trust requirement as in bfh_export_pdf.
batch_session	Optional bfh_export_session object from bfh_create_export_session. Reuses the staged template directory across calls. Cannot be combined with template_path or inject_assets.

panel — Optional side panel from bfh_figure_panel: key figures, colour legend and data definition in a 72.6 mm column to the right of the figure. Default NULL renders the figure full width.

Details

Requires Quarto CLI (>= 1.4.0) for compilation, like bfh_export_pdf. For SPC charts (bfh_qic_result), use bfh_export_pdf instead.

What the function changes about your plot: it removes the plot title and subtitle, removes axis titles that are blank (NULL, "" or whitespace; titles derived from aes() are kept), and sets the plot margins to 0 mm. Nothing else: the caller owns the figure's theme and typography. The chart is rendered 264 mm wide (full width) or 191.4 mm wide next to a panel. Without analysis the page has no analysis row and the chart is 130.8 mm high; with analysis it is 109 mm high, as on SPC pages.

Fonts: text inside the figure is rendered with the font family the plot declares, resolved against the fonts available to the Typst compile (font_path, injected assets, and system fonts only when ignore_system_fonts = FALSE). A default ggplot theme declares "Arial". With the BFHchartsAssets companion injected (it ships both Arial and Mari), the figure text renders in Arial; without any Arial font in the effective font path it falls back to Typst's default font. Add BFHtheme::theme_bfh() to the plot for typography consistent with SPC pages (Mari).

Value

The input plot invisibly, enabling pipe chaining.

Security

Same guards as bfh_export_pdf: output path validation, restrict_template semantics, inject_assets validation and temp-workspace protection. inject_assets is full code execution and MUST NOT come from user input.

See Also

- bfh_export_pdf() for SPC charts
- bfh_stage_figure_page() to stage a figure page for batch reports
- bfh_create_export_session() for batch workflows

Other export-functions: bfh_apply_date_axis(), bfh_export_batch_pdf(), bfh_figure_panel(), bfh_prune_page_cache(), bfh_stage_figure_page(), bfh_stage_pdf_page()

Examples

```
## Not run:
library(ggplot2)

p <- ggplot(mtcars, aes(factor(cyl))) +
  geom_bar() +
  labs(x = "Cylinders", y = "Cars") +
  BFHtheme::theme_bfh()
```

```
bfh_export_figure_pdf(
  p, "cylinders.pdf",
  metadata = list(
    title = "Most cars have four or eight cylinders",
    department = "Kvalitetsafdeling",
    analysis = "Distribution of cylinders in the sample.",
    details = "Source: mtcars"
  )
)

## End(Not run)
```

`bfh_export_pdf` *Export BFH QIC Chart to PDF via Typst*

Description

Exports an SPC chart created by `bfh_qic()` to a PDF document using Typst templates for hospital branding. Requires Quarto CLI for compilation.

Usage

```
bfh_export_pdf(
  x,
  output,
  metadata = list(),
  template = "bfh-diagram",
  template_path = NULL,
  restrict_template = TRUE,
  auto_analysis = FALSE,
  use_ai = FALSE,
  data_consent = NULL,
  use_rag = FALSE,
  analysis_min_chars = 300,
  analysis_max_chars = 375,
  dpi = 150,
  font_path = NULL,
  ignore_system_fonts = TRUE,
  inject_assets = NULL,
  batch_session = NULL,
  strict_baseline
)
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`
`output`	Character string specifying the output PDF file path

`metadata` List with optional metadata fields:

- `hospital`: Hospital name (default: "Bispebjerg og Frederiksberg Hospital")
- `department`: Department/unit name (optional)
- `analysis`: Analysis text with findings (optional)
- `details`: Period info, averages (auto-generated if not provided). Auto-generated format: "Periode: feb. 2019 - mar. 2022 * Gns. maaned: 58938/97266 * Seneste maaned: 60756/88509 * Nuvaerende niveau: 64,5\
- `author`: Author name (optional)
- `date`: Report date (default: Sys.Date())
- `data_definition`: Data definition text (optional)
- `target`: Target value for analysis context. Must be either NULL, a single finite numeric (`length 1`, no NA/Inf/NaN), or a single character string (`length 1`, not NA). Multi-element vectors and non-finite numerics are rejected with an informative error. The target also auto-flows from `bfh_qic(target_text=, target_value=)` via the analysis-context fallback chain (since v0.12.0).
- `footer_content`: Additional content to display below the chart (optional). Supports markdown formatting (bold, italic, line breaks).
- `logo_path`: Path to hospital logo image rendered in the template's foreground slot (optional). When NULL (default), no logo is rendered – the PDF compiles successfully without proprietary branding assets. Companion packages (e.g. `BFHchartsAssets`) inject the logo via `inject_assets` callback, and `compose_typst_document()` auto-detects a staged logo at `<staged-template>/images/Hospital_Maerke_RGB_A1_str.png` – no caller intervention required. Explicit `logo_path` overrides auto-detect. Path must be a non-empty character string; existence is not pre-validated (Typst surfaces the file-not-found error at compile).

`template` Character string specifying template name (default: "bfh-diagram")

`template_path` Optional path to a custom Typst template file. When provided, this overrides the packaged template. The template must exist and be a valid Typst file (.typ). Default is NULL (uses packaged BFH template).

`restrict_template`

Logical. When `TRUE` (default), any non-NULL `template_path` is rejected with an error. Use `FALSE` only in trusted contexts where a custom Typst template is intentionally supplied.

Threat model: A custom Typst template is compiled by the Typst binary and can read or write arbitrary paths during compilation. This is equivalent to `source()` in trust requirements. `restrict_template = TRUE` prevents a compromised configuration pipeline from injecting a malicious template via `template_path`.

Default: `TRUE` (production-safe – custom templates require explicit opt-in).

Migration from BFHcharts <= 0.15.x: Callers passing `template_path` without `restrict_template` now receive a clear validation error. Migration is mechanical:

```
# Before (BFHcharts <= 0.15.x): custom template silently allowed
bfh_export_pdf(result, "out.pdf", template_path = "/my/template.typ")
```

```
# After (BFHcharts >= 0.16.0): explicit opt-out required
bfh_export_pdf(result, "out.pdf",
               template_path = "/my/template.typ",
               restrict_template = FALSE)
```

auto_analysis
: Logical. If TRUE and `metadata$analysis` is not provided, automatically generates analysis text using `bfh_generate_analysis()`. Default is FALSE for backward compatibility.

use_ai
: Logical. Controls AI usage for auto-analysis:

- `FALSE` (default): Use standard texts only - no external data sharing
- `TRUE`: Use AI via BFHllm (requires BFHllm installed and `data_consent = "explicit"`; error if not satisfied)

Only used when `auto_analysis = TRUE`. See `bfh_generate_analysis()` for security policy details.

data_consent
: Character. Required when `use_ai = TRUE` and `auto_analysis = TRUE`. Must be `"explicit"` to acknowledge that clinical data is sent to `BFHllm`. Passed through to `bfh_generate_analysis()`. Ignored when `use_ai = FALSE`. See bfh_generate_analysis for full GDPR/HIPAA context.

use_rag
: Logical. Controls RAG for AI analysis. Default `FALSE` (privacy-preserving). Only used when `auto_analysis = TRUE` and `use_ai = TRUE`. See bfh_generate_analysis.

analysis_min_chars
: Minimum characters for AI-generated analysis. Default 300. Only used when `auto_analysis = TRUE`.

analysis_max_chars
: Maximum characters for AI-generated analysis. Default 375. Only used when `auto_analysis = TRUE`.

dpi
: Resolution passed to `ggplot2::ggsave()`. Default 150. PDF export currently uses SVG as intermediate format, so this is mainly relevant if the plot contains rasterized content.

font_path
: Optional path to directory containing additional fonts. Passed as `--font-path` to the Typst compiler. Useful when fonts (e.g., Mari) are bundled in a downstream package and not installed system-wide on the deployment platform.

ignore_system_fonts
: Logical. If `TRUE` (default), passes `--ignore-system-fonts` to Typst so only fonts from `font_path` (or bundled template fonts) are used. Prevents inconsistent rendering when developers have additional Mari variants (e.g., Mari Heavy) installed system-wide. Passed to the internal Typst compiler.

inject_assets
: Optional callback function called after Typst template structure is created but before compilation. Receives one argument: the path to the template directory (e.g., `<temp_dir>/bfh-template`). Use this to copy fonts, images, or other assets into the template directory when they are not bundled in BFHcharts (e.g., proprietary fonts in a private package). Cannot be combined with `batch_session` (pass `inject_assets` to `bfh_create_export_session()` instead).

batch_session
: Optional bfh_export_session object from bfh_create_export_session(). When provided, the packaged template assets are reused from the session tmpdir instead of being copied on every call, which eliminates the dominant I/O cost in batch workflows.
  - Cannot be combined with template_path or inject_assets.
  - font_path here overrides session$font_path.
  - Close the session with close(session) after the batch is done.

strict_baseline
: Logical. When TRUE (default), the function errors before render if x$config$freeze < MIN_BASELINE_N (8) or if any phase in x$qic_data contains fewer than MIN_BASELINE_N points. When FALSE, the export proceeds and the legacy warning-only behavior of bfh_qic() is preserved.

  Rationale: PDFs from this function typically reach quality-improvement leadership where R warnings never surface. Anhoej & Olesen (2014) recommend >= 8 baseline points for reliable run/crossing detection; charts with shorter baselines have tight but statistically unreliable control limits. Strict-by-default forces explicit acknowledgement of short-baseline output; the interactive bfh_qic() path remains warning-only because the analyst is present.

  Inheritance: When batch_session is supplied without an explicit per-call value, the session's strict_baseline is used. An explicit per-call value overrides the session.

Details

Requirements:

- Quarto CLI (>= 1.4.0) must be installed
- Install from: https://quarto.org

PDF Generation Process:

1. Extract chart title from plot (removed from image for template)
2. Export chart to temporary PNG (without title)
3. Extract SPC statistics (runs, crossings, outliers)
4. Generate Typst document (.typ) with template
5. Compile to PDF via Quarto CLI
6. Clean up temporary files

Title Handling:

- Chart title is extracted and passed to Typst template
- Title appears in PDF header, NOT in chart image
- This differs from PNG export where title is in the image

Plot Optimization for PDF:

- Plot margins are set to 0mm for optimal fit in Typst template

- Blank axis titles (NULL or empty) are removed with element_blank()
- User-defined axis titles are preserved

SPC Statistics:

- Automatically extracted from bfh_qic_result$summary
- Displayed in SPC table on PDF
- Includes: runs (serielaengde), crossings (antal kryds), outliers

Auto-Generated Details:

- If `metadata$details` is not provided, details are auto-generated
- Format: "Periode: start - slut . Gns. interval: values . Seneste interval: values . Nuvaerende niveau: cl"
- p/u-charts show numerator/denominator (e.g., "58938/97266")
- Other chart types show only values (e.g., "127")
- Interval labels adapt to data frequency (month, week, day, etc.)

Value

The input object `x` invisibly, enabling pipe chaining

Security

`inject_assets` **is full code execution.** The supplied function runs with the same privileges as the calling R session, with full file-system and network access. It MUST NOT come from user input (Shiny inputs, REST API parameters, configuration files of unknown provenance).

Acceptable sources:

- A function exported from a controlled companion package installed via `pak::pkg_install("private/repo")`.
- A function defined in your application's source code, version-controlled.

Unacceptable sources:

- `parse(text = input$user_code)`
- A function loaded from an unverified URL
- A function deserialized from an untrusted RDS file

When in doubt, do not pass `inject_assets`.

`template_path` is compiled by the Typst binary. A custom template can read and write arbitrary paths during compilation – treat it with the same trust contract as `source()`.

Default since 0.16.0: `restrict_template = TRUE` – any non-NULL `template_path` is rejected at validation time. Power users supplying a trusted in-process template MUST opt-in with `restrict_template = FALSE`. The default-safe posture eliminates the silent privilege-escalation vector that would otherwise exist if a configuration pipeline forwarded user-controlled input to `template_path`. See ADR-003 for the warning-blind-clinical-reader risk model that drove this default.

Never forward user-supplied input (Shiny inputs, query parameters, untrusted uploads) to either parameter – doing so creates a privilege-escalation vector. If your application surface needs to

expose template customization to end users, validate against a fixed allow-list of approved templates and callbacks before invoking `bfh_export_pdf()`.

A runtime heuristic warns when `inject_assets` originates from `.GlobalEnv` or a direct child environment (typical of accidental Shiny exposure). Suppress with `options(BFHcharts.allow_globalenv_inject = TRUE)` in development.

The same trust requirement applies to `inject_assets` when passed to `bfh_create_export_session()`.

Recommended: companion package for proprietary branding. Organizations deploying BFHcharts-based applications that need consistent proprietary branding (custom fonts, hospital logos) should distribute those assets via a private companion R package. The companion package exposes a single function compatible with `inject_assets`, e.g. `inject_bfh_assets(template_dir)`. Consumer applications then call `bfh_export_pdf(..., inject_assets = MyAssetsPkg::inject_bfh_assets)`. This keeps proprietary assets out of public BFHcharts distribution while preserving full branding in production. The companion package is still subject to the trusted-code-only contract above. For the BFH/Region Hovedstaden reference deployment, the `BFHchartsAssets` private companion package implements this pattern. See `vignette("organizational-deployments")` (when available) or the BFHchartsAssets repository documentation.

See Also

- `bfh_qic()` to create SPC charts
- `bfh_export_png()` to export as PNG
- `bfh_export_figure_pdf()` for graphs without SPC statistics (any `ggplot`, rendered full-width in the same template)
- `bfh_create_export_session()` for batch workflows (same trust requirement applies to its `inject_assets` parameter)

Examples

```
## Not run:
library(BFHcharts)

# Create sample data
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  infections = rpois(24, lambda = 15)
)

# Create and export chart to PDF in one pipeline
bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Hospital-Acquired Infections"
) |>
  bfh_export_pdf(
    "infections_report.pdf",
    metadata = list(
```

```
      hospital = "BFH",
      department = "Kvalitetsafdeling",
      analysis = "Signifikant fald observeret efter intervention",
      data_definition = "Antal hospital-erhvervede infektioner per maaned"
    )
  )

# Multiple exports from same chart
result <- bfh_qic(data, month, infections,
  chart_type = "i",
  chart_title = "Infections"
)

# PNG for email/presentation
bfh_export_png(result, "infections.png")

# PDF for official report
bfh_export_pdf(result, "infections_report.pdf",
  metadata = list(department = "ICU")
)

# Batch export: reuse template assets across multiple exports
departments <- c("ICU", "Medicine", "Surgery")
session <- bfh_create_export_session()
on.exit(close(session))
for (dept in departments) {
  bfh_export_pdf(result, paste0(dept, "_report.pdf"),
    metadata = list(department = dept),
    batch_session = session
  )
}

## End(Not run)
```

`bfh_export_png` *Export BFH QIC Chart to PNG*

Description

Exports an SPC chart created by `bfh_qic()` to a PNG file with configurable dimensions and resolution. Designed for pipe-compatible workflows.

Usage

```
bfh_export_png(x, output, width_mm = 200, height_mm = 120, dpi = 300)
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`
`output`	Character string specifying the output file path (e.g., "chart.png")

`width_mm`	Numeric. Width of the output image in millimeters (default: 200mm)
`height_mm`	Numeric. Height of the output image in millimeters (default: 120mm)
`dpi`	Numeric. Dots per inch resolution for the PNG (default: 300)

Details

Dimension Handling:

- Dimensions are specified in millimeters (Danish/European standard)
- Internally converted to inches for ggplot2::ggsave()
- Default 200mm x 120mm ~= 7.87" x 4.72" (common presentation size)

Resolution (DPI):

- 300 DPI (default): High quality for print and presentations
- 150 DPI: Medium quality, smaller file size
- 96 DPI: Screen resolution, minimal file size

Title Handling:

- The chart title (if present) is rendered in the PNG image
- This differs from PDF export which strips the title for Typst template

Plot Optimization:

- Plot already has 5mm margins from bfh_qic() via apply_spc_theme()
- Blank axis titles are automatically removed when plot is created
- User-defined axis titles are preserved
- No additional processing needed at export time

Pipe Compatibility:

- Returns input object invisibly for chaining
- Example: `bfh_qic(...) |> bfh_export_png("chart.png")`

Value

The input object `x` invisibly, enabling pipe chaining

See Also

- `bfh_qic()` to create SPC charts
- `bfh_export_pdf()` to export as PDF with Typst templates

Examples

```
## Not run:
library(BFHcharts)

# Create sample data
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  infections = rpois(24, lambda = 15)
)

# Create and export chart in one pipeline
bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Monthly Infections"
) |>
  bfh_export_png("monthly_infections.png", width_mm = 250, height_mm = 150, dpi = 300)

# Explicit dimensions for specific use cases
result <- bfh_qic(data, month, infections, chart_type = "i")

# A4 width (210mm) for reports
bfh_export_png(result, "report_chart.png", width_mm = 210, height_mm = 140)

# PowerPoint slide (widescreen 16:9)
bfh_export_png(result, "slide_chart.png", width_mm = 254, height_mm = 143)

# Web display (lower DPI for smaller file size)
bfh_export_png(result, "web_chart.png", width_mm = 200, height_mm = 120, dpi = 96)

## End(Not run)
```

---

bfh_extract_spc_stats *Extract SPC Statistics*

Description

S3 generic that extracts statistical process control metrics. The extraction logic depends on the input type:

Usage

```
bfh_extract_spc_stats(x)

## Default S3 method:
bfh_extract_spc_stats(x)
```

```
## S3 method for class 'data.frame'
bfh_extract_spc_stats(x)

## S3 method for class 'bfh_qic_stats'
bfh_extract_spc_stats(x)

## S3 method for class 'bfh_qic_result'
bfh_extract_spc_stats(x)
```

Arguments

x	Either a data frame (typically `bfh_qic_result$summary`), a `bfh_qic_result` object from `bfh_qic()`, or `NULL`.

Details

- `data.frame` (typically `bfh_qic_result$summary`): Returns runs and crossings from the summary. `outliers_actual` and `outliers_recent_count` remain `NULL` because outlier counts require access to `qic_data`.
- `bfh_qic_result`: Returns runs, crossings, and outlier counts. Outliers are split into two fields so that the PDF table and the analysis text can be driven from consistent - but distinct - numbers.
- `NULL`: Returns an empty stats list (backward compatible).

Downstream packages should prefer the `bfh_qic_result` method so the PDF export and any on-screen preview agree on the outlier count.

Stats are computed on x-sorted observations; input row order is not significant. When `qic_data` contains an `x` column, rows are sorted ascending by `x` before the recency-window slice for `outliers_recent_count` is applied. This ensures that reversed or scrambled input yields identical results to chronologically ordered input.

Value

Named list with SPC statistics:

runs_expected Expected maximum run length (`laengste_loeb_max`)

runs_actual Actual longest run length (`laengste_loeb`)

crossings_expected Expected minimum crossings (`antal_kryds_min`)

crossings_actual Actual number of crossings (`antal_kryds`)

outliers_expected Expected number of outliers (0 for non-run charts, `NULL` otherwise)

outliers_actual Total number of points outside control limits in the latest part (used by the PDF table). `NULL` for `data.frame` input, run charts, or when `sigma.signal` is unavailable.

outliers_recent_count Number of outliers within the last 6 observations of the latest part (used by the analysis text, so stale outliers are not discussed as if they were current). Present only for `bfh_qic_result` input on non-run charts.

is_run_chart Logical indicating run chart. Present only for `bfh_qic_result` input.

cl_user_supplied Logical. TRUE when the caller passed a non-NULL cl argument to bfh_qic(); Anhoej run/crossing signals in this case were computed against the user-supplied centerline, not the data-estimated process mean. Mirrors attr(result$summary, "cl_user_supplied") and is present in both the bfh_qic_result and data.frame (summary) dispatch paths. Always FALSE for NULL input or summaries without the attribute.

See Also

bfh_qic() for creating SPC charts

Other utility-functions: bfh_create_typst_document(), bfh_generate_details(), bfh_merge_metadata()

Examples

```
## Not run:
result <- bfh_qic(data, x = date, y = value, chart_type = "i")

# Full stats (recommended - populates outliers_actual for the table)
stats <- bfh_extract_spc_stats(result)

# Backward-compatible summary-only dispatch
stats_summary_only <- bfh_extract_spc_stats(result$summary)

## End(Not run)
```

---

bfh_figure_panel *Side Panel for Figure Pages*

Description

Describes the right-hand column of a figure page: large key figures, a colour legend and the data definition. Pass the result as panel to bfh_export_figure_pdf or bfh_stage_figure_page. The panel has the same width (72.6 mm) as the SPC statistics column of bfh_export_pdf, so the figure is rendered 191.4 mm wide.

Usage

```
bfh_figure_panel(
  legend = NULL,
  kpis = NULL,
  legend_title = NULL,
  kpi_title = NULL,
  definition_height_mm = NULL,
  placement = c("side", "bottom"),
  kpi_columns = NULL,
  kpi_size_pt = NULL,
  kpi_labels = TRUE,
  legend_rows = NULL,
```

```
  legend_label_width_mm = NULL,
  definition_first = FALSE,
  kpi_width_mm = NULL,
  kpi_label_size_pt = NULL,
  kpi_label_gap_mm = NULL,
  kpis_end = NULL,
  kpi_end_title = NULL,
  compact = FALSE
)
```

Arguments

legend	Optional data frame with columns `label` and `colour` (hex, e.g. `"#007dbb"`) and optionally `group`, `key` (`"box"`, the default; `"line"` for a line series; `"arrow_up"` or `"arrow_down"` for a thick vertical arrow), `linewidth` (ggplot2 linewidth of a `"line"` key, default 0.5) and `outline` (hex colour of a thin border around a `"box"` key, e.g. a band drawn with an outline).
kpis	Optional data frame with columns `label` and `value` (numbers are formatted as text) and optionally `colour` and `label_colour` (colour of the label below the number in the bottom panel; default grey). A `"\n"` in a label is a line break in the bottom panel.
legend_title	Heading above the legend. Default `NULL` uses the template default ("Tegnforklaring"); `""` shows no heading.
kpi_title	Optional heading above the key figures.
definition_height_mm	Optional height of the data definition block in millimetres (template default 39.6). Text that does not fit is clipped with an ellipsis. Ignored with `placement = "bottom"`.
placement	`"side"` (default): a 72.6 mm column right of the figure. `"bottom"`: a row below a full-width figure.
kpi_columns	Bottom placement only: number of columns in a grid of key figures, filled column by column (e.g. 2 gives a 2 x 2 grid for four key figures). Default `NULL`: all key figures in one row.
kpi_size_pt	Bottom placement only: font size of the key figure numbers in points. Default `NULL` (26 pt).
kpi_labels	Bottom placement only: show the label below each number. Set to `FALSE` when a legend explains the colours.
legend_rows	Bottom placement only: number of legend rows, filled column by column. Default `NULL`: one column up to three entries, otherwise two columns. The legend has the same height as the key figures, with the rows spread evenly.
legend_label_width_mm	Bottom placement only: maximum width of a legend label in millimetres; longer labels wrap onto more lines, which leaves more room for the data definition. Default `NULL`: no limit.
compact	Bottom placement only: legend entries stand at the top with normal spacing (instead of spread over the full height), and the lines of the data definition have normal line spacing (no paragraph gaps). Default `FALSE`.

`definition_first`
: Side placement only: show the data definition at the top of the column in its natural height, followed by the key figures and the legend. Default `FALSE`: the definition is a fixed-height block at the bottom (see `definition_height_mm`).

`kpi_width_mm`, `kpi_label_size_pt`, `kpi_label_gap_mm`
: Bottom placement only: minimum width of each key figure column (mm), font size of the labels (pt, default 7.5) and the space between number and label (mm, default 1.4). With `kpi_width_mm` labels wrap only at `"\n"` and the columns stand closer together.

`kpis_end`, `kpi_end_title`
: Bottom placement only: a second group of key figures (same columns as `kpis`) with its own heading, shown in one column at the far right of the row, after the data definition (as on the old occupancy sheets). Uses the same size and label settings as `kpis`. Ignored with `placement = "side"`.

Details

The legend is drawn by the Typst template, not by ggplot2: remove the plot's own legend (e.g. `guide = "none"`) and list the same colours here. A group heading is shown each time `group` changes, so sort the legend in the order it should be read (typically the stacking order, top first).

With a panel, `metadata$data_definition` is rendered at the bottom of the panel (no warning, unlike full-width mode).

With `placement = "bottom"` the figure uses the full page width and the panel becomes a row below it: key figures side by side (number above, label below), then the legend, then the data definition to the right. Legend group headings are not shown in that row.

Value

An object of class `bfh_figure_panel`.

See Also

`bfh_export_figure_pdf()`, `bfh_stage_figure_page()`

Other export-functions: `bfh_apply_date_axis()`, `bfh_export_batch_pdf()`, `bfh_export_figure_pdf()`, `bfh_prune_page_cache()`, `bfh_stage_figure_page()`, `bfh_stage_pdf_page()`

Examples

```
bfh_figure_panel(
  legend = data.frame(
    label = c("Inden for 30 dage", "Frist overskredet"),
    colour = c("#007dbb", "#c0392b"),
    group = c("Overholdt", "Overskredet")
  ),
  kpis = data.frame(label = "Ikke udredte", value = 37)
)
```

`bfh_generate_analysis` *Generate SPC Analysis Text*

Description

Generates analysis text for PDF export using AI (BFHllm) if available, with automatic fallback to Danish standard texts.

Usage

```
bfh_generate_analysis(
  x,
  metadata = list(),
  use_ai = FALSE,
  data_consent = NULL,
  use_rag = FALSE,
  min_chars = 300,
  max_chars = 375,
  target_tolerance = 0.05,
  language = NULL,
  texts_loader = NULL
)
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`
`metadata`	Optional list with additional context for AI: • `data_definition`: Description of what the data represents • `target`: Target value for the metric • `hospital`: Hospital name • `department`: Department name • `chart_title`: Override chart title • `y_axis_unit`: Override y-axis unit
`use_ai`	Logical. Should AI be used for analysis generation? **Security note:** Default is `FALSE` (explicit opt-in required). AI analysis sends `qic_data`, metadata, baseline, department and hospital data to `BFHllm::bfhllm_spc_suggestion()`. In healthcare contexts, implicit external data processing is unacceptable. Always set `use_ai = TRUE` deliberately, and always supply `data_consent = "explicit"`. • `FALSE` (default): Use standard texts only - no external data sharing • `TRUE`: Use AI (requires BFHllm package and `data_consent = "explicit"`; error if not satisfied)
`data_consent`	Character. Required when `use_ai = TRUE`. Must be `"explicit"` to acknowledge that `qic_data`, metadata, and context are transmitted to `BFHllm::bfhllm_spc_suggestion()`. Any other value (including the default `NULL`) raises an error. Ignored when `use_ai = FALSE`.

GDPR/HIPAA context: qic_data may contain aggregated clinical indicators, department names, and hospital identifiers. In healthcare deployments, processing this data via an external AI service requires documented consent and a data processing agreement. Setting data_consent = "explicit" is the caller's attestation that the appropriate legal basis exists. An audit event is emitted automatically via .emit_audit_event() for traceability.

use_rag
: Logical. Controls whether BFHllm::bfhllm_spc_suggestion() is invoked with retrieval-augmented generation (RAG) enabled. Default is FALSE (privacy-preserving: one-shot LLM call only). Set TRUE to allow broader context from a vector store.

  Privacy implication: When use_rag = TRUE, query data may be stored in or matched against a vector store maintained by BFHllm's backend. This is a separate compliance concern from the one-shot LLM call. Only enable when your data processing agreement explicitly covers vector-store usage. The value is always recorded in the audit event.

min_chars
: Minimum characters in AI-generated output. Default 300.

max_chars
: Maximum characters in AI-generated output. Default 375.

target_tolerance
: Deprecated. Argument is preserved in the signature for backward compatibility but is no longer used. The at_target classification now uses process variation (mean((UCL - LCL) / 6) over the last phase, with sd(y) as fallback for run charts) instead of a relative-to-target tolerance. Passing a non-default value emits a deprecation warning. The parameter will be removed in the next major release.

language
: Character string specifying output language. One of "da" (Danish) or "en" (English). Default NULL inherits the language stored in x$config$language (set via bfh_qic(language=)). Falls back to "da" when the chart carries no language (pre-v0.26 objects).

texts_loader
: Function that returns SPC analysis text templates. Defaults to load_spc_texts(language). Primarily intended for tests/mocking.

Details

Security policy: AI analysis is opt-in only (use_ai = FALSE by default). Setting use_ai = TRUE requires BFHllm to be installed and data_consent = "explicit" to be supplied; an informative error is raised if either condition is not met. This prevents accidental data exposure in environments where BFHllm uses network calls, RAG, or third-party services.

Installing BFHllm: BFHllm is not on CRAN and must be installed manually from GitHub before using use_ai = TRUE:

```
remotes::install_github("johanreventlow/BFHllm")
```

When use_ai = TRUE and all preconditions are met, the function:

1. Validates data_consent = "explicit"
2. Emits a structured audit event via .emit_audit_event()

3. Builds context from the bfh_qic_result and metadata
4. Calls BFHllm::bfhllm_spc_suggestion() for AI-generated analysis
5. Falls back to standard texts if AI call fails

When use_ai = FALSE (default):

- Returns Danish standard texts based on Anhoej SPC rules
- data_consent is not checked

Value

Character string with analysis text suitable for PDF export.

AI audit event

When use_ai = TRUE and BFHllm is installed, a structured audit event is emitted via .emit_audit_event() before calling BFHllm::bfhllm_spc_suggestion(). The event includes: timestamp, event type ("ai_egress"), package, target function, fields transmitted, use_rag value, hostname, and user.

If options(BFHcharts.audit_log = "/path/to/audit.jsonl") is set, the event is appended as a JSON line. Otherwise it is emitted via message() with prefix [BFHcharts/audit].

Rationale: Hospital deployments need an audit trail when patient-context SPC data is sent to an external LLM. The structured event is parseable and cannot be globally suppressed unlike message().

See Also

This function is a single-step convenience wrapper. For the two-step pipeline that gives more control, use:

- bfh_analyse – composes the structured bfh_spc_analysis object (analysis layer).
- bfh_render_analysis – renders that object to character output (render layer).
- bfh_build_analysis_context – lower-level helper used internally to extract chart context.
- bfh_generate_details – generates the companion detail row string from the same bfh_qic_result input.
- bfh_qic for producing the bfh_qic_result input object.

Examples

```
## Not run:
result <- bfh_qic(data, x = date, y = value, chart_type = "i")

# Use standard texts (no AI)
analysis <- bfh_generate_analysis(result, use_ai = FALSE)

# Use AI with explicit data consent
analysis <- bfh_generate_analysis(result,
  metadata = list(
    data_definition = "Antal infektioner pr. 1000 patientdage",
    target = 2.5
  ),
```

```
  use_ai = TRUE,
  data_consent = "explicit"
)

# Use AI with RAG enabled (vector-store context)
analysis <- bfh_generate_analysis(result,
  use_ai = TRUE,
  data_consent = "explicit",
  use_rag = TRUE
)

## End(Not run)
```

`bfh_generate_details` *Generate Details Text for PDF Export*

Description

Automatically generates a details string based on chart data, including period range, averages, latest values, and current level (centerline).

Usage

```
bfh_generate_details(x, language = "da", x_labels = NULL)
```

Arguments

`x`	A `bfh_qic_result` object from `bfh_qic()`
`language`	Character string specifying output language. One of `"da"` (Danish, default) or `"en"` (English). Default `"da"` preserves backward compatibility.
`x_labels`	Optional character vector of original categorical x-axis labels (month names, weekdays, etc.). When supplied, the period range uses the first and last labels (e.g. "januar - december") rather than the numeric sequence stored in `qic_data$x`. Use this when text-x has been converted to a numeric sequence by upstream pipelines. Length must match `nrow(qic_data)`. Default `NULL` uses date-based formatting.

Details

Format:

- Period range with Danish date formatting
- Average values per interval (numerator/denominator for p/u-charts)
- Latest period values
- Current level (centerline value) with appropriate unit formatting

Chart Type Handling:

- p, pp, u, up: Shows numerator/denominator (e.g., "58938/97266") – pp/up (Laney-adjusted) share the same numerator/denominator contract as p/u.
- ip: Shows numerator/denominator only when a real denominator was supplied to `bfh_qic(n = ...)`. Without one, ip degenerates to a plain individuals chart and shows only the value, same as chart type "i".
- Other chart types: Shows only the value (e.g., "127")

Interval Detection:

- Uses detect_date_interval() to determine the interval type
- Labels adapt: "maaned", "uge", "dag", "kvartal", "aar"

Fail-early contract:

- If `qic_data$x` contains no finite/non-NA values (empty, all-NA, or all-Inf for numeric), the function stops with a `bfhcharts_config_error`.
- Calls with valid data are unaffected.

Value

Character string with formatted details, e.g.: "Periode: feb. 2019 - mar. 2022 * Gns. maaned: 58938/97266 * Seneste maaned: 60756/88509 * Nuvaerende niveau: 64,5%"

See Also

Companion to the analysis pipeline; operates on the same `bfh_qic_result` input:

- `bfh_qic` produces the `bfh_qic_result` input.
- `bfh_analyse` composes the structured analysis object (main analysis pipeline entry point).
- `bfh_generate_analysis` generates the main analysis text in a single step.
- `bfh_render_analysis` renders a `bfh_spc_analysis` object to character output.
- `bfh_export_pdf` for PDF export that consumes both the analysis text and the details string.

Other utility-functions: `bfh_create_typst_document()`, `bfh_extract_spc_stats()`, `bfh_merge_metadata()`

Examples

```
## Not run:
result <- bfh_qic(data, x = date, y = value, chart_type = "i")
details <- bfh_generate_details(result)
# "Periode: jan. 2024 - dec. 2024 * Gns. maaned: 50 * ..."

## End(Not run)
```

bfh_get_plot *Extract Plot from bfh_qic_result*

Description

Helper function to extract the ggplot object for further customization. Users can use `result$plot` directly or call this function.

Usage

```
bfh_get_plot(x)
```

Arguments

x	A `bfh_qic_result` object

Value

ggplot2 object

Examples

```
## Not run:
data <- data.frame(
  x = seq(as.Date("2024-01-01"), by = "month", length.out = 12),
  y = rnorm(12, 10, 2)
)
result <- bfh_qic(data, x = x, y = y, chart_type = "run")
p <- bfh_get_plot(result)
print(p)

## End(Not run)
```

bfh_merge_metadata *Merge User Metadata with Defaults*

Description

Merges user-provided metadata with package defaults for PDF generation. This function is useful for downstream packages that need consistent metadata handling without depending on BFHcharts internal functions.

Usage

```
bfh_merge_metadata(metadata, chart_title)
```

Arguments

`metadata`	Named list with user-provided metadata fields. Valid fields: hospital, department, title, analysis, details, author, date, data_definition, footer_content, logo_path. Other fields are ignored.
`chart_title`	Character string with chart title. Used as default for metadata$title if not provided by user.

Value

Named list with merged metadata containing:

hospital Hospital name (default: "Bispebjerg og Frederiksberg Hospital")

department Department name (default: NULL)

title Chart title (from chart_title or metadata)

analysis Analysis description (default: NULL)

details Additional details (default: NULL)

author Author name (default: NULL)

date Report date (default: Sys.Date())

data_definition Data definition (default: NULL)

footer_content Footer content below chart (default: NULL)

logo_path Path to hospital logo image (default: NULL). When NULL, the Typst template renders without a foreground logo. Companion packages (BFHchartsAssets) populate this via inject_assets callback or auto-detection in `compose_typst_document()`.

User-provided values override defaults. Fields not in the default list are silently ignored.

See Also

`bfh_export_pdf()` for PDF export functionality

Other utility-functions: `bfh_create_typst_document()`, `bfh_extract_spc_stats()`, `bfh_generate_details()`

Examples

```
## Not run:
# Basic usage
metadata <- list(
  department = "Kvalitetsafdeling",
  analysis = "Signifikant fald observeret"
)
merged <- bfh_merge_metadata(metadata, chart_title = "Infektioner")

# merged$hospital = "Bispebjerg og Frederiksberg Hospital" (default)
# merged$department = "Kvalitetsafdeling" (user override)
# merged$title = "Infektioner" (from chart_title)

## End(Not run)
```

bfh_prune_page_cache *Prune Page Bundles from a Batch Export Cache*

Description

Removes page bundles staged by bfh_stage_pdf_page from cache_dir, selected either by an explicit keep-list (delete every bundle NOT named – typically the same manifest vector passed to bfh_export_batch_pdf) or by staging age (delete bundles staged before a cutoff). Exactly one selector must be supplied.

Usage

```
bfh_prune_page_cache(
  cache_dir,
  keep = NULL,
  older_than = NULL,
  dry_run = FALSE
)
```

Arguments

cache_dir	The page cache directory.
keep	Character vector of bundle ids to keep; every other bundle is removed. Mutually exclusive with older_than.
older_than	A Date/POSIXct (or string coercible to one): bundles staged before this time are removed. Bundles whose staging time cannot be read are skipped with a warning, never deleted. Mutually exclusive with keep.
dry_run	Logical. When TRUE, nothing is deleted; the return value lists the bundles that WOULD be removed. Default FALSE.

Details

Pruning is deliberately separate from compilation: compiling a subset via the ids manifest is a read operation, while pruning deletes data. Only directories that look like page bundles (containing page.rds) are candidates; other files in the cache directory are never touched.

Value

Character vector of removed (or, under dry_run, would-be removed) bundle ids, sorted.

See Also

bfh_stage_pdf_page(), bfh_export_batch_pdf()

Other export-functions: bfh_apply_date_axis(), bfh_export_batch_pdf(), bfh_export_figure_pdf(), bfh_figure_panel(), bfh_stage_figure_page(), bfh_stage_pdf_page()

Examples

```
## Not run:
ids <- production_chart_ids()
# Inspect first, then delete retired bundles:
bfh_prune_page_cache("~/reports/spc-cache", keep = ids, dry_run = TRUE)
bfh_prune_page_cache("~/reports/spc-cache", keep = ids)

## End(Not run)
```

---

bfh_qic *Create BFH-Styled SPC Chart*

Description

One-function approach to create publication-ready SPC charts. Wraps qicharts2 calculation and BFH visualization in a single call.

Convenience function that combines qicharts2::qic() calculation with BFH-styled visualization and automatic label placement. Handles the entire workflow from raw data to finished plot with intelligent labels.

Usage

```
bfh_qic(
  data,
  x,
  y,
  n = NULL,
  chart_type = "run",
  y_axis_unit = "count",
  chart_title = NULL,
  target_value = NULL,
  target_text = NULL,
  notes = NULL,
  part = NULL,
  freeze = NULL,
  exclude = NULL,
  cl = NULL,
  ylim = NULL,
  multiply = 1,
  agg.fun = c("mean", "median", "sum", "sd"),
  agg_fun = NULL,
  base_size = 14,
  width = NULL,
  height = NULL,
  units = NULL,
  dpi = 96,
```

```
  plot_margin = NULL,
  ylab = "",
  xlab = "",
  subtitle = NULL,
  caption = NULL,
  return.data = FALSE,
  return_data = NULL,
  language = "da"
)
```

Arguments

data	Data frame with measurements
x	Name of x-axis column (unquoted, NSE). Usually date/time column.
y	Name of y-axis column (unquoted, NSE). The measurement variable.
n	Name of denominator column for ratio charts (optional, unquoted, NSE)
chart_type	Chart type: "run", "i", "mr", "p", "pp", "u", "up", "c", "g", "xbar", "s", "t", "ip" **I-prime chart** (`"ip"`): An individuals control chart with varying denominators, computed via the **pbcharts** package (Taylor method). Unlike the standard `"i"` chart – where `y` is plotted directly – the I-prime chart models a ratio: `y` is the NUMERATOR and `n` is the DENOMINATOR; the plotted value is `y / n` with control limits that adjust for the varying subgroup size. This mirrors the P/U-chart denominator contract. If `n` is omitted, the chart degenerates to a classic individuals chart with constant limits. **pbcharts** is an optional dependency; install with `remotes::install_github("anhoej/pbcharts")` if not present.
y_axis_unit	Unit type: "count", "percent", "rate", "time", or "clock" (klokkeslaet tt:mm; y-vaerdier i sekunder siden midnat)
chart_title	Plot title (optional)
target_value	Numeric target value (optional)
target_text	Target label text (optional)
notes	Character vector of annotations for data points (optional, same length as data)
part	Positions for phase splits (optional numeric vector). Must be positive integers, strictly increasing, unique, and within `[2, nrow(data)]`. Non-integer (e.g. `3.5`), duplicated, or unsorted values raise an error before any qicharts2 call.
freeze	Position to freeze baseline (optional single integer). Must be a positive integer in `[1, nrow(data) - 1]`. Non-integer values raise an error.
exclude	Integer vector of data point positions to exclude from calculations (optional). Must be positive integers, unique, and within `[1, nrow(data)]`. Sort order is not required.
cl	Numeric value to set a custom centerline instead of calculating from data (optional). When non-NULL, Anhoej run/crossing signals are computed against the user-supplied centerline rather than the data-estimated process mean – an R warning is emitted, and the returned `bfh_qic_result$summary` carries `attr(summary, "cl_user_supplied") = TRUE`. Downstream consumers (e.g. PDF rendering, biSPCharts UI) can read the attribute via `isTRUE(attr(result$summary, "cl_user_supplied"))`.

The bfh_export_pdf() PDF surfaces a caveat note below the SPC table when this flag is set. See bfh_extract_spc_stats() for the discoverable surface.

ylim — Y-axis display limits as a numeric vector of length 2, c(min, max), applied via ggplot2::coord_cartesian() (default: NULL = data-driven). Values are in the plot's data scale: proportion charts (p, pp) use the 0-1 proportion scale (e.g. c(0, 1) shows 0%-100%), all other charts use the absolute data scale. Use NA for a free end: c(0, NA) fixes the bottom at 0 with a data-driven top. Because coord_cartesian zooms rather than clips, data points and control-limit segments outside the range are NOT dropped (unlike scale_y limits) – the chart simply zooms to the requested window. If min >= max (both non-NA) the limit is ignored with a warning.

multiply — Numeric multiplier for y-axis values, e.g. 100 to convert proportions to percentages (default: 1)

agg.fun — Deprecated. Use agg_fun instead. Kept for backwards compatibility.

agg_fun — Aggregation function for run/I charts with multiple observations per subgroup: "mean" (default), "median", "sum", "sd". Preferred snake_case alias for the legacy agg.fun.

base_size — Base font size in points (default: auto-calculated from width/height if provided, otherwise 14)

width — Plot width (optional). Supports smart unit detection or explicit units parameter. See Details.

height — Plot height (optional). Supports smart unit detection or explicit units parameter. See Details.

units — Unit type for width/height: "cm" (centimeters), "mm" (millimeters), "in" (inches), "px" (pixels), or NULL for smart auto-detection (default)

dpi — Dots per inch for pixel conversion (default: 96). Only used when units = "px"

plot_margin — Plot margins as either: (1) numeric vector c(top, right, bottom, left) in mm, or (2) ggplot2::margin() object. Default NULL uses BFHtheme defaults.

ylab — Y-axis label (default: "" for blank)

xlab — X-axis label (default: "" for blank)

subtitle — Plot subtitle text (default: NULL for no subtitle)

caption — Plot caption text (default: NULL for no caption)

return.data — **Deprecated**. Use return_data instead. Kept for backwards compatibility. Passing TRUE emits a deprecation warning.

return_data — Logical. **Deprecated**: TRUE is type-unstable and will be removed in a future version. Use bfh_extract_spc_stats() to access the underlying SPC data frame. If TRUE, returns the raw qic data frame; if FALSE (default), returns a bfh_qic_result S3 object. Preferred snake_case alias for the legacy return.data.

language — Character string specifying output language. One of "da" (Danish, default) or "en" (English). Controls three independent aspects of output (since v0.12.0):

- Translated labels (control limit text, target prefixes)

- Y-axis number formatting: decimal separator and thousand separator (Danish `1.234,5` vs English `1,234.5`); percent suffix (Danish `12,5 %` vs English `12.5%`)
- X-axis date formatting: month/weekday abbreviations match the requested locale on a best-effort basis (depends on platform locale availability)

Default `"da"` preserves backward compatibility for Danish users.

Details

Helper map (internal orchestration functions):

- `validate_bfh_qic_inputs()` – all input validation (type, bounds, NSE, denominator, target)
- `build_qic_args()` – constructs argument list for `qicharts2::qic()`
- `invoke_qicharts2()` – calls `do.call(qicharts2::qic, ...)` + `add_anhoej_signal()`
- `compute_viewport_base_size()` – unit conversion, responsive `base_size`, label normalisation
- `render_bfh_plot()` – plot_config + viewport + `bfh_spc_plot()` with warning suppression
- `apply_spc_labels_to_export()` – label_size computation + `add_spc_labels()` with warning suppression
- `build_bfh_qic_return()` – return value routing (S3 vs. legacy paths)

SPC Statistics Provenance:

Every field in `result$summary` is derived from `result$qic_data` via `format_qic_summary()`. The aggregation function and scope for each field:

Field	Source column	Aggregation	Scope	C
`laengste_loeb`	`longest.run`	`max()` per phase	per-phase	a
`laengste_loeb_max`	`longest.run.max`	`max()` global	global	a
`antal_kryds`	`n.crossings`	`max()` per phase	per-phase	a
`antal_kryds_min`	`n.crossings.min`	`max()` global	global	a
`anhoej_signal`	`runs.signal`	`any()` per phase	per-phase	a
`runs_signal`	(derived)	`laengste_loeb` > `laengste_loeb_max` per phase	per-phase	a
`crossings_signal`	(derived)	`antal_kryds` < `antal_kryds_min` per phase	per-phase	a
`sigma_signal`	`sigma.signal`	`any()` per phase	per-phase	r
`centerlinje`	`cl`	first row per phase	per-phase	a
`nedre_kontrolgraense`	`lcl`	first row per phase	per-phase	r
`oevre_kontrolgraense`	`ucl`	first row per phase	per-phase	r
`nedre_kontrolgraense_min/max`	`lcl`	`min()`/`max()` per phase	per-phase	r
`oevre_kontrolgraense_min/max`	`ucl`	`min()`/`max()` per phase	per-phase	r
`kontrolgraenser_konstante`	`lcl+ucl`	all-equal check per phase	per-phase	r
`antal_observationer`	`n.obs`	first row per phase	per-phase	a
`anvendelige_observationer`	`n.useful`	first row per phase	per-phase	a
`antal_outliers`	`sigma.signal`	`sum()` per phase	per-phase	r
`forventede_outliers`	(constant 0)	literal 0	per-phase	r

Notes:

- `longest.run` is stored by qicharts2 as a per-phase constant (all rows within a phase hold the same value). `max()` over those rows is equivalent to the value.
- `sigma.signal` is per-row (varies within a phase). `any()` aggregation is correct.
- Limit columns use scalar form when limits are constant within all phases (e.g. I-chart, C-chart). Variable-denominator charts (P, U) use min/max column pairs.
- Run charts (`chart_type = "run"`) produce no `sigma_signal`, `lcl`/`ucl`, or outlier columns in `result$summary`.

Chart Types:

- **run**: Run chart (no control limits)
- **i**: I-chart (individuals)
- **mr**: Moving Range chart – measures point-to-point variability. Typically paired with an I-chart to characterise both process level (I) and short-term variation (MR).
- **p**: P-chart (proportions, requires n)
- **pp**: P-prime chart (Laney-adjusted proportions) – use instead of `p` when denominators are very large (n > 1000 per subgroup) and standard control limits become artificially tight due to over-dispersion. Requires n.
- **u**: U-chart (rates, requires n)
- **up**: U-prime chart (Laney-adjusted rates) – same rationale as `pp`, applied to count rates. Requires n.
- **c**: C-chart (counts)
- **g**: G-chart (geometric)
- **xbar**: X-bar chart
- **s**: S-chart
- **t**: T-chart (time between events)
- **ip**: I-prime chart (individuals with varying denominators, via pbcharts)

Y-Axis Units:

- **count**: Integer counts with K/M notation
- **percent**: Percentage values (0-100%)
- **rate**: Decimal values with comma notation
- **time**: Context-aware minutes/hours/days

Phase Configuration:

- `part`: Vector of positions where phase splits occur (e.g., `c(12, 24)`)
- `freeze`: Position to freeze baseline calculation

Denominator Contract (ratio charts): Ratio chart types (`p`, `pp`, `u`, `up`) require a denominator column supplied via `n`. The content of `n` is validated to prevent silently misleading rate plots:

- `n` must be numeric and finite (no `Inf`/`-Inf`).
- All non-`NA` values of `n` must be `> 0` (zero/negative denominators produce meaningless rates).

- For proportion charts (p, pp): every row with both y and n present must satisfy y <= n (proportion <= 1).
- NA in individual rows of n is allowed (qicharts2 drops them).
- Violations raise an error identifying the offending row number(s) so the source data can be inspected. Pre-filter or correct invalid rows before calling bfh_qic(). Other chart types (run, i, mr, c, g, t, xbar, s) are not subject to denominator validation.

Unit Support (Danish-friendly): Width and height support multiple units for convenience:

- **Smart auto-detection** (default, units = NULL):
  - Values > 100 -> pixels (e.g., width = 800 -> 800px)
  - Values 10-100 -> centimeters (e.g., width = 25 -> 25cm)
  - Values < 10 -> inches (e.g., width = 10 -> 10in, legacy)
- **Explicit units** (units = "cm", "mm", "in", "px"):
  - Centimeters: width = 25, height = 15, units = "cm" (Danish standard)
  - Millimeters: width = 250, height = 150, units = "mm"
  - Inches: width = 10, height = 6, units = "in" (legacy)
  - Pixels: width = 800, height = 600, units = "px", dpi = 96 (web/Shiny)

Responsive Typography: When width and height are provided, base_size is automatically calculated using geometric mean: sqrt(width x height) / 3.5 This ensures fonts scale proportionally with plot size. Override by explicitly setting base_size.

Automatic Label Placement: Labels are automatically added to the plot showing:

- Current level (CL) from the most recent phase
- Target value (if specified via target_value or target_text)
- Intelligent collision avoidance with multi-level fallback strategy
- Provide width and height for optimal label sizing and placement

Arrow Symbol Suppression: If target_text contains arrow symbols (up down or < >), the target line will be suppressed and only the directional indicator shown at the plot edge.

Percent Target Contract: When y_axis_unit = "percent", target_value is validated against the scale implied by multiply:

- multiply = 1 (default): target_value must be in [0, 1.5] (proportion)
- multiply = 100: target_value must be in [0, 150] (percent)
- multiply = m: target_value must be in [0, m * 1.5]

The most common error is passing target_value = 2.0 to mean "2%" when multiply = 1 (proportion scale). Use target_value = 0.02 instead, or set multiply = 100 to pass percent values directly. A 1.5x upper slack permits legitimate stretch targets above 100%.

Value

- Default (return_data = FALSE): `bfh_qic_result` S3 object with components:
  - `$plot`: ggplot2 object with the SPC chart
  - `$summary`: data.frame with SPC statistics
  - `$qic_data`: data.frame with raw qicharts2 calculations
  - `$config`: list with original function parameters
- return_data = TRUE: data.frame with qic calculations (legacy behavior)

Examples

```
# Minimal runnable example with deterministic data
df <- data.frame(
  maaned = seq.Date(as.Date("2023-01-01"), by = "month", length.out = 12),
  vaerdi = c(10, 12, 11, 13, 10, 14, 11, 12, 10, 13, 11, 12)
)
result <- bfh_qic(df, x = maaned, y = vaerdi, chart_type = "i")
class(result) # "bfh_qic_result"

## Not run:
library(BFHcharts)

# Example 1: Simple run chart with monthly data
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  infections = rpois(24, lambda = 15),
  surgeries = rpois(24, lambda = 100)
)

plot <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "run",
  y_axis_unit = "count",
  chart_title = "Monthly Hospital-Acquired Infections"
)
plot

# Example 2: P-chart with target line
plot <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  n = surgeries,
  chart_type = "p",
  y_axis_unit = "percent",
  chart_title = "Infection Rate per 100 Surgeries",
  target_value = 0.02,
  target_text = "down Maalet: 2%"
)
plot
```

```
# Example 3: I-chart with phase splits
plot <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Infections with Intervention",
  part = c(12), # Phase split after 12 months
  freeze = 12 # Freeze baseline at month 12
)
plot

# Example 4: Chart with annotations using notes
notes_vec <- rep(NA, 24)
notes_vec[3] <- "Start of intervention"
notes_vec[12] <- "New protocol implemented"
notes_vec[18] <- "Staff training completed"

plot <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Infections with Annotated Events",
  notes = notes_vec
)
plot

# Example 5: Responsive typography with viewport dimensions
# Small plot (6x4 inches) -> base_size ~= 14pt
plot_small <- bfh_qic(
  data = data, x = month, y = infections,
  chart_type = "i", y_axis_unit = "count",
  chart_title = "Small Plot - Auto Scaled Typography",
  width = 6, height = 4 # Auto: base_size ~= 14pt
)

# Medium plot (10x6 inches) -> base_size ~= 22pt
plot_medium <- bfh_qic(
  data = data, x = month, y = infections,
  chart_type = "i", y_axis_unit = "count",
  chart_title = "Medium Plot - Auto Scaled Typography",
  width = 10, height = 6 # Auto: base_size ~= 22pt
)

# Large plot (16x9 inches) -> base_size ~= 34pt
plot_large <- bfh_qic(
  data = data, x = month, y = infections,
  chart_type = "i", y_axis_unit = "count",
  chart_title = "Large Plot - Auto Scaled Typography",
```

```
  width = 16, height = 9 # Auto: base_size ~= 34pt
)

# Override auto-scaling with explicit base_size
plot_custom <- bfh_qic(
  data = data, x = month, y = infections,
  chart_type = "i", y_axis_unit = "count",
  chart_title = "Custom Typography Override",
  width = 10, height = 6,
  base_size = 18 # Explicit override
)

# Example 6: Exclude outliers from calculations
plot_exclude <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "I-Chart with Excluded Outliers",
  exclude = c(3, 15) # Exclude data points 3 and 15
)

# Example 7: Use median instead of mean for aggregation
plot_median <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "I-Chart Using Median",
  agg_fun = "median"
)

# Example 8: Multiply y-values for unit conversion
# Convert proportions (0-1) to percentages (0-100)
data_prop <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  proportion = runif(24, 0.01, 0.05) # Proportions 0.01-0.05
)

plot_multiply <- bfh_qic(
  data = data_prop,
  x = month,
  y = proportion,
  chart_type = "i",
  y_axis_unit = "percent",
  chart_title = "Proportions Converted to Percentages",
  multiply = 100 # Convert 0.01 -> 1%
)

# Example 9: Custom centerline (cl parameter)
# Use a fixed benchmark or standard instead of calculating from data
```

```
plot_cl <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Infections with Custom Centerline",
  cl = 10 # Set centerline to fixed benchmark of 10
)

# Example 10: Custom plot margins (numeric vector in mm)
plot_tight <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Chart with Tight Margins",
  plot_margin = c(2, 2, 2, 2) # 2mm on all sides
)

# Example 11: Custom margins with margin() object
plot_custom_margin <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Chart with Custom Margins",
  plot_margin = ggplot2::margin(t = 5, r = 15, b = 5, l = 10, unit = "mm")
)

# Example 12: Responsive margins using lines (scales with base_size)
plot_responsive <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  base_size = 18,
  plot_margin = ggplot2::margin(t = 0.5, r = 1, b = 0.5, l = 1, unit = "lines")
)

# Example 13: Custom axis labels, subtitle, and caption
plot_labels <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Hospital-Acquired Infections",
  ylab = "Antal infektioner",
  xlab = "Maaned",
```

```
  subtitle = "Kirurgisk afdeling - 2024",
  caption = "Data: EPJ system | Analyse: Kvalitetsafdelingen"
)

# Example 14: Add BFHtheme branding (hospital logo, custom styling)
plot_branded <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Hospital-Acquired Infections - Official Report",
  base_size = 14
) |>
  BFHtheme::add_bfh_logo() # Add hospital branding

# Alternate BFHtheme styles available:
# - BFHtheme::theme_bfh_dark() for dark theme
# - BFHtheme::theme_bfh_print() for print-optimized theme
# - BFHtheme::theme_bfh_presentation() for presentations

# Example 15: Danish-friendly unit support (centimeters)
plot_cm <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Plot in Centimeters (Danish Standard)",
  width = 25, # 25 cm (auto-detected as cm)
  height = 15 # 15 cm
)

# Example 16: Explicit unit specification
plot_explicit <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Explicit Centimeters",
  width = 25, height = 15, units = "cm"
)

# Example 17: Pixel dimensions for web/Shiny
plot_px <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Plot for Web Display",
  width = 800, # 800 px (auto-detected as px)
```

```
  height = 600, # 600 px
  dpi = 96
)

# Example 18: Backward compatibility (inches still work)
plot_inches <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Legacy Inches Format",
  width = 10, # 10 inches (auto-detected as in)
  height = 6 # 6 inches
)

# Example 19: Get raw qic data for further analysis
qic_data <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  return_data = TRUE # Return data.frame instead of plot
)

# Now you can access all qic calculations
head(qic_data)
# Available columns: cl, ucl, lcl, runs.signal, sigma.signal, etc.

# Example 20: Get summary statistics with Danish column names
result <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Infections - With Summary"
)

# Access the plot
result$plot

# Access the summary statistics (Danish column names) via S3 object
print(result$summary)
# Columns: fase, antal_observationer, anvendelige_observationer,
#          centerlinje, nedre_kontrolgraense, oevre_kontrolgraense,
#          laengste_loeb, antal_kryds, anhoej_signal,
#          runs_signal, crossings_signal, sigma_signal

# Example 21: Get both raw qic data and summary via S3 object
result <- bfh_qic(
  data = data,
```

```
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  part = c(12) # Split into phases
)

# Access raw qic data (all qicharts2 calculations)
result$qic_data

# Access summary statistics (one row per phase)
result$summary
# fase 1: baseline period
# fase 2: intervention period

# Example 22: Use summary for reporting
result <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  n = surgeries,
  chart_type = "p",
  y_axis_unit = "percent",
  chart_title = "Infection Rate - Multi-phase Analysis",
  part = c(12)
)

# Extract key metrics for reporting via result$summary
summary_stats <- result$summary
cat("Fase 1 centerlinje:", summary_stats$centerlinje[1], "%\n")
cat("Fase 2 centerlinje:", summary_stats$centerlinje[2], "%\n")
cat("Forbedring:", summary_stats$centerlinje[1] - summary_stats$centerlinje[2], "%-point\n")

if (summary_stats$sigma_signal[2]) {
  cat("VIGTIG: Special cause variation detekteret i fase 2!\n")
}

# Example 23: P-prime chart for large denominators (Laney-adjusted)
data_large_n <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  events = rpois(24, lambda = 50),
  population = rpois(24, lambda = 5000) # Very large denominators
)
plot_pp <- bfh_qic(
  data = data_large_n,
  x = month,
  y = events,
  n = population,
  chart_type = "pp", # Laney-adjusted: prevents artificially tight limits
  y_axis_unit = "percent",
  chart_title = "Event Rate (Laney P-prime)"
)
```

```
# Example 24: Moving Range chart paired with I-chart
# Use MR-chart alongside I-chart for full process variation characterisation
plot_i <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "i",
  y_axis_unit = "count",
  chart_title = "Infections - Process Level (I-chart)"
)
plot_mr <- bfh_qic(
  data = data,
  x = month,
  y = infections,
  chart_type = "mr", # Moving Range: short-term variation
  y_axis_unit = "count",
  chart_title = "Infections - Short-term Variation (MR-chart)"
)

# Example 25: I-prime chart with varying denominator (requires pbcharts)
# y = numerator (event count), n = denominator (varying subgroup size).
# The plotted value is y/n; control limits adjust for each n.
if (requireNamespace("pbcharts", quietly = TRUE)) {
  df_iprime <- data.frame(
    month = seq.Date(as.Date("2023-01-01"), by = "month", length.out = 24),
    events = c(
      2L, 3L, 1L, 4L, 2L, 3L, 5L, 2L, 1L, 3L,
      2L, 4L, 1L, 2L, 3L, 2L, 4L, 1L, 3L, 2L, 4L, 3L, 1L, 2L
    ),
    denom = c(
      80L, 95L, 70L, 105L, 85L, 90L, 110L, 75L, 65L, 100L,
      80L, 95L, 70L, 85L, 90L, 75L, 100L, 70L, 95L, 80L,
      105L, 90L, 65L, 85L
    )
  )
  plot_iprime <- bfh_qic(
    data = df_iprime,
    x = month,
    y = events,
    n = denom,
    chart_type = "ip",
    y_axis_unit = "percent",
    chart_title = "I-prime Chart - Event Rate with Varying Denominator"
  )
  plot(plot_iprime)
}

## End(Not run)
```

---

bfh_qic_result *BFH QIC Result S3 Class*

Description

S3 class for wrapping SPC chart outputs. Enables pipe-compatible export workflows while maintaining backwards-compatible console display.

Details

Objects of class `bfh_qic_result` are returned by `bfh_qic`. Access components with `result$plot`, `result$summary`, `result$qic_data`, and `result$config`.

`bfh_qic_stats` *Compute SPC Statistics Without Building a Plot*

Description

`bfh_qic_stats()` runs the same computation as `bfh_qic()` - input validation, qicharts2 (or pbcharts for `chart_type = "ip"`), auto-mean substitution for run charts and summary formatting - but skips plot construction and label placement. Use it when only the numbers are needed, e.g. when scanning many series for Anhoej signals.

Usage

```
bfh_qic_stats(
  data,
  x,
  y,
  n = NULL,
  chart_type = "run",
  y_axis_unit = "count",
  target_value = NULL,
  notes = NULL,
  part = NULL,
  freeze = NULL,
  exclude = NULL,
  cl = NULL,
  multiply = 1,
  agg_fun = NULL
)
```

Arguments

`data`	Data frame with measurements
`x`	Name of x-axis column (unquoted, NSE). Usually date/time column.
`y`	Name of y-axis column (unquoted, NSE). The measurement variable.
`n`	Name of denominator column for ratio charts (optional, unquoted, NSE)

chart_type	Chart type: "run", "i", "mr", "p", "pp", "u", "up", "c", "g", "xbar", "s", "t", "ip" **I-prime chart** ("ip"): An individuals control chart with varying denominators, computed via the **pbcharts** package (Taylor method). Unlike the standard "i" chart – where y is plotted directly – the I-prime chart models a ratio: y is the NUMERATOR and n is the DENOMINATOR; the plotted value is y / n with control limits that adjust for the varying subgroup size. This mirrors the P/U-chart denominator contract. If n is omitted, the chart degenerates to a classic individuals chart with constant limits. **pbcharts** is an optional dependency; install with remotes::install_github("anhoej/pbcharts") if not present.
y_axis_unit	Unit type: "count", "percent", "rate", "time", or "clock" (klokkeslaet tt:mm; y-vaerdier i sekunder siden midnat)
target_value	Numeric target value (optional)
notes	Character vector of annotations for data points (optional, same length as data)
part	Positions for phase splits (optional numeric vector). Must be positive integers, strictly increasing, unique, and within [2, nrow(data)]. Non-integer (e.g. 3.5), duplicated, or unsorted values raise an error before any qicharts2 call.
freeze	Position to freeze baseline (optional single integer). Must be a positive integer in [1, nrow(data) - 1]. Non-integer values raise an error.
exclude	Integer vector of data point positions to exclude from calculations (optional). Must be positive integers, unique, and within [1, nrow(data)]. Sort order is not required.
cl	Numeric value to set a custom centerline instead of calculating from data (optional). When non-NULL, Anhoej run/crossing signals are computed against the user-supplied centerline rather than the data-estimated process mean – an R warning is emitted, and the returned bfh_qic_result$summary carries attr(summary, "cl_user_supplied") = TRUE. Downstream consumers (e.g. PDF rendering, biSPCharts UI) can read the attribute via isTRUE(attr(result$summary, "cl_user_supplied")). The bfh_export_pdf() PDF surfaces a caveat note below the SPC table when this flag is set. See bfh_extract_spc_stats() for the discoverable surface.
multiply	Numeric multiplier for y-axis values, e.g. 100 to convert proportions to percentages (default: 1)
agg_fun	Aggregation function for run/I charts with multiple observations per subgroup: "mean" (default), "median", "sum", "sd". Preferred snake_case alias for the legacy agg.fun.

Details

For the same computation arguments, $summary (including its cl_user_supplied and cl_auto_mean attributes) and $qic_data are identical to those returned by bfh_qic(). Both functions share one internal computation phase, so they cannot drift apart.

Value

An object of class bfh_qic_stats: a list with

- summary: data.frame with one row per phase, as bfh_qic()$summary

- qic_data: data.frame with the raw qic calculations, as bfh_qic()$qic_data
- config: list with the computation settings (chart_type, y_axis_unit, target_value, part, freeze, exclude, cl, multiply, agg.fun, has_denominator)

The object has no plot and is not a bfh_qic_result; use bfh_qic() when a chart or an export is needed.

See Also

bfh_qic() for the full result with a plot, bfh_extract_spc_stats() for extracting key statistics.

Examples

```
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 24),
  infections = c(15, 18, 14, 16, 19, 13, 17, 20, 15, 18, 16, 14,
                 17, 19, 15, 16, 18, 14, 13, 17, 16, 15, 19, 18)
)

stats <- bfh_qic_stats(data, x = month, y = infections, chart_type = "i")
stats$summary
bfh_extract_spc_stats(stats)
```

---

bfh_render_analysis *Render a bfh_spc_analysis Object to Character Output*

Description

Resolverer i18n-noegler i analysis$conclusions, analysis$caveats og analysis$suggested_actions til tekst via texts_loader, sammen- saetter base + target + action + caveats inden for max_chars-budget.

Usage

```
bfh_render_analysis(analysis, max_chars = 375, texts_loader = NULL)
```

Arguments

analysis	A bfh_spc_analysis object from bfh_analyse().
max_chars	Maximum characters in output. Default 375.
texts_loader	Function that returns SPC analysis text templates. Defaults to load_spc_texts(analysis$language). Primarily intended for tests/mocking.

Details

render_context-vaerdier (target_display, centerline_formatted, operator_unicode, outliers_word_key) bruges verbatim som placeholder-vaerdier. Renderer re-deriverer IKKE disse fra features/aux for at undgaa silent display-drift.

Value

Character of length 1, `nchar(...) <= max_chars`.

See Also

Pipeline position: third (final) of three layers.

- `bfh_analyse` produces the `bfh_spc_analysis` input consumed by this function (analysis layer).
- `bfh_qic` produces the `bfh_qic_result` that feeds into `bfh_analyse()`.
- `bfh_build_analysis_context` is a lower-level helper used internally earlier in the pipeline.
- `bfh_generate_analysis` combines `bfh_analyse()` and rendering in a single convenience call.
- `bfh_generate_details` generates the companion detail row string from the same `bfh_qic_result` input.

Examples

```
## Not run:
result <- bfh_qic(data, x = month, y = value, chart_type = "i")
analysis <- bfh_analyse(result, metadata = list(target = ">= 90%"))
text <- bfh_render_analysis(analysis, max_chars = 375)

## End(Not run)
```

---

bfh_stage_figure_page *Stage a ggplot Figure as a Page Bundle for Batch PDF Export*

Description

Batch counterpart of `bfh_export_figure_pdf`: renders one arbitrary `ggplot` object (not an SPC chart) into a persistent "page bundle" in `cache_dir`, producing no PDF. Figure bundles use the same on-disk format and `format_version` as bundles from `bfh_stage_pdf_page`, so `bfh_export_batch_pdf` compiles SPC pages and figure pages into one document without any change to its interface. Order, manifest selection, chunking and pruning treat figure bundles exactly like SPC bundles.

Usage

```
bfh_stage_figure_page(
  plot,
  cache_dir,
  id = NULL,
  order = NULL,
  metadata = list(),
  template = "bfh-diagram",
```

```
  dpi = 150,
  overwrite = TRUE,
  panel = NULL
)
```

Arguments

plot	A single ggplot object. Composite plots (class patchwork) are rejected.
cache_dir	Existing, writable directory holding the page cache.
id	Bundle identifier (single string matching ^[A-Za-z0-9][A-Za-z0-9_-]*$). Default: a zero-padded sequence id.
order	Numeric ordering key for page sorting at compile time. Default: one past the highest order currently staged in the cache.
metadata	Named list of template metadata; title is **required**. Same fields and data_definition behavior as bfh_export_figure_pdf.
template	Typst template function name (default "bfh-diagram"). All bundles compiled into one document must share the same template.
dpi	Resolution passed to the SVG device (default 150).
overwrite	Logical. Replace an existing bundle with this id (default TRUE). FALSE raises an error on duplicate ids.
panel	Optional side panel from bfh_figure_panel. Same behavior as in bfh_export_figure_pdf.

Details

The bundle carries metadata$spc_panel = FALSE, the optional panel and empty SPC statistics, so the batch compiler renders the figure layout without page-type branching. Chart size and plot handling are identical to bfh_export_figure_pdf: the plot's own title and subtitle are removed, blank axis titles are removed, margins are set to 0 mm, and the caller owns the figure's theme and typography (see the font note there).

Staging is atomic and re-staging an existing id replaces the bundle, exactly as for bfh_stage_pdf_page.

Value

Invisibly, a bfh_staged_page object: a list with id, path (bundle directory), and order.

Trust model

The cache directory is trusted input: bundles are deserialized (RDS) in the calling R session by bfh_export_batch_pdf(). Never populate a cache directory from untrusted sources, and do not point cache_dir at world-writable locations in multi-tenant deployments.

Compatibility

Older BFHcharts versions do not know the full-width layout: reading a figure bundle from a cache shared with such a version renders the page with the SPC layout (empty statistics column, narrower chart). Re-staging does not help on the old installation; upgrade it instead.

See Also

bfh_export_batch_pdf() to compile staged bundles, bfh_stage_pdf_page() for SPC charts, bfh_export_figure_pdf() for single-figure export.

Other export-functions: bfh_apply_date_axis(), bfh_export_batch_pdf(), bfh_export_figure_pdf(), bfh_figure_panel(), bfh_prune_page_cache(), bfh_stage_pdf_page()

Examples

```
## Not run:
cache_dir <- "~/reports/spc-cache"
dir.create(cache_dir, showWarnings = FALSE)
p <- ggplot2::ggplot(mtcars, ggplot2::aes(wt, mpg)) +
  ggplot2::geom_point()
bfh_stage_figure_page(p, cache_dir,
  id = "vaegt-vs-forbrug",
  metadata = list(title = "Tunge biler koerer kortere")
)

## End(Not run)
```

---

bfh_stage_pdf_page *Stage a Chart as a Page Bundle for Batch PDF Export*

Description

Renders one bfh_qic_result into a persistent "page bundle" in cache_dir, performing the same chart preparation as bfh_export_pdf (metadata finalization, title strip, label recalculation, SVG rendering, SPC stats extraction) but producing no PDF. Staged bundles are later assembled into one multi-page PDF by bfh_export_batch_pdf.

Usage

```
bfh_stage_pdf_page(
  x,
  cache_dir,
  id = NULL,
  order = NULL,
  metadata = list(),
  template = "bfh-diagram",
  auto_analysis = FALSE,
  use_ai = FALSE,
  data_consent = NULL,
  use_rag = FALSE,
  analysis_min_chars = 300,
  analysis_max_chars = 375,
  dpi = 150,
```

```
  overwrite = TRUE,
  strict_baseline = TRUE
)
```

Arguments

x	A bfh_qic_result object from bfh_qic.
cache_dir	Existing, writable directory holding the page cache. Persistent and caller-owned: bundles survive R sessions and processes.
id	Bundle identifier (single string matching ^[A-Za-z0-9][A-Za-z0-9_-]*$; it becomes a directory name). Default: a zero-padded sequence id ("page-0001", ...) one past the highest existing default id in the cache.
order	Numeric ordering key for page sorting at compile time (ties broken by id). Default: one past the highest order currently staged in the cache (staging sequence).
metadata	Named list of template metadata, same fields as bfh_export_pdf (hospital, department, analysis, ...).
template	Typst template function name (default "bfh-diagram"). All bundles compiled into one document must share the same template.
auto_analysis, use_ai, data_consent, use_rag, analysis_min_chars, analysis_max_chars	Auto-generated analysis controls, identical to bfh_export_pdf. AI/RAG analysis runs at staging time (the chart data is available here), never at batch compile time.
dpi	Resolution passed to the SVG device (default 150).
overwrite	Logical. Replace an existing bundle with this id (default TRUE). FALSE raises an error on duplicate ids.
strict_baseline	Logical. Same baseline validation as bfh_export_pdf; default TRUE.

Details

A bundle consists of the rendered chart SVG plus a serialized data file (page.rds) holding finalized template metadata, extracted SPC statistics, the template selection, an ordering key, and a cache format version. No SPC statistics are recomputed and no ggplot objects are serialized, so bundles remain readable across ggplot2/BFHtheme upgrades.

Staging is atomic: the bundle is written to a hidden staging directory and renamed into place, so a failed or interrupted staging never leaves a half-written bundle visible to bfh_export_batch_pdf(). Re-staging an existing id (with overwrite = TRUE, the default) replaces the bundle atomically – the supported way to correct individual pages before recompiling a combined report.

Value

Invisibly, a bfh_staged_page object: a list with id, path (bundle directory), and order.

Trust model

The cache directory is trusted input: bundles are deserialized (RDS) in the calling R session by `bfh_export_batch_pdf()`. Never populate a cache directory from untrusted sources, and do not point `cache_dir` at world-writable locations in multi-tenant deployments.

See Also

`bfh_export_batch_pdf()` to compile staged bundles, `bfh_prune_page_cache()` to remove stale bundles, `bfh_export_pdf()` for single-chart export.

Other export-functions: `bfh_apply_date_axis()`, `bfh_export_batch_pdf()`, `bfh_export_figure_pdf()`, `bfh_figure_panel()`, `bfh_prune_page_cache()`, `bfh_stage_figure_page()`

Examples

```
## Not run:
cache_dir <- "~/reports/spc-cache"
dir.create(cache_dir, showWarnings = FALSE)
result <- bfh_qic(data, x = date, y = value, chart_type = "p")
bfh_stage_pdf_page(result, cache_dir,
  id = "afd-a-infektioner",
  metadata = list(department = "Afdeling A")
)

## End(Not run)
```

---

bfh_subsample_label_indices

Subsample Text X-Axis Label Indices

Description

Returns deterministic, evenly-spaced indices for selecting which categorical x-axis labels to display on a chart. First index is always 1. Intermediate indices follow integer step `ceiling((n_labels - 1) / (max_visible - 1))`. The last index `n_labels` is included only when it lands naturally on the step grid (i.e. when `(n_labels - 1)` is divisible by `step`); otherwise the sequence ends at the highest step-aligned position `<= n_labels`. Designed for tekst-x-data (month names, weekdays, observation IDs) where showing all labels would overlap or require rotation.

Usage

```
bfh_subsample_label_indices(n_labels, max_visible = BFH_MAX_X_LABELS_TEXT)
```

Arguments

`n_labels`	Integer. Total number of labels available.
`max_visible`	Integer. Maximum number of labels to display. Default BFH_MAX_X_LABELS_TEXT (currently 12).

Details

Algorithm scales smoothly: for n <= max_visible all indices returned; otherwise step-based thinning preserves a constant label-density and avoids the asymmetric-rounding gap-bug that `round(seq(..., length.out))` produces for moderate n (issue #396).

Note: prior versions (0.22.0) appended `n_labels` as a force-last anchor, producing a shorter tail-gap mid-rhythm. This was removed because the rhythm-break was visually jarring (e.g. n=24 jumped from a constant step=3 to a step=2 tail). Callers requiring the last label as anchor must append it explicitly.

Value

Integer vector of indices into the label sequence. First index is always 1; last index is `n_labels` only when it falls on the step grid, otherwise the highest step-aligned position <= `n_labels`. Length <= `max_visible`.

Examples

```
# All 12 months visible when n <= max
bfh_subsample_label_indices(12)
# [1] 1 2 3 4 5 6 7 8 9 10 11 12

# 24 months: step=3, no force-last (sidste = 22, ej 24)
bfh_subsample_label_indices(24)
# [1]  1  4  7 10 13 16 19 22

# 100 obs: step=9, n=100 lands naturally on grid -> included
bfh_subsample_label_indices(100)
# [1]   1  10  19  28  37  46  55  64  73  82  91 100

# Custom max
bfh_subsample_label_indices(24, max_visible = 6)
# [1]  1  6 11 16 21
```

`close.bfh_export_session`

Close a BFH Export Session

Description

Releases the session tmpdir and marks the session as closed. Subsequent calls are no-ops. Called automatically by the session finalizer at garbage collection or R session exit, but explicit `close()` is recommended for prompt cleanup.

Usage

```
## S3 method for class 'bfh_export_session'
close(con, ...)
```

Arguments

con	A bfh_export_session object created with bfh_create_export_session.
...	Additional arguments (ignored).

Value

Invisibly returns NULL.

---

format.bfh_spc_analysis *Format Method for bfh_spc_analysis*

Description

Returns a single-line character summary. Used by paste(), format- contexts, and debugging-helpers.

Usage

```
## S3 method for class 'bfh_spc_analysis'
format(x, ...)
```

Arguments

x	A bfh_spc_analysis object.
...	Ignored.

Value

Character scalar.

---

is_bfh_qic_result *Check if Object is bfh_qic_result*

Description

Check if Object is bfh_qic_result

Usage

```
is_bfh_qic_result(x)
```

Arguments

x	Object to test

Value

Logical indicating whether x is a bfh_qic_result object

Examples

```
## Not run:
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 12),
  infections = rpois(12, lambda = 15)
)
result <- bfh_qic(data, x = month, y = infections, chart_type = "run")
is_bfh_qic_result(result) # TRUE
is_bfh_qic_result(result$plot) # FALSE

## End(Not run)
```

---

is_bfh_spc_analysis *Check if Object is bfh_spc_analysis*

Description

Check if Object is bfh_spc_analysis

Usage

```
is_bfh_spc_analysis(x)
```

Arguments

x	Object to test.

Value

Logical indicating whether x inherits from `bfh_spc_analysis`.

---

new_bfh_qic_result *Create a bfh_qic_result Object*

Description

Constructor for the bfh_qic_result S3 class. This class wraps SPC chart outputs to enable pipe-compatible export functions while preserving backwards-compatible console behavior.

Usage

```
new_bfh_qic_result(plot, summary, qic_data, config)
```

Arguments

plot	ggplot2 object containing the SPC chart
summary	data.frame with SPC statistics (runs, crossings, control limits). Plain data.frame – not a tibble; downstream consumers should not rely on tibble-specific subscripting behavior.
qic_data	data.frame with raw qicharts2 calculation results
config	list with original function parameters

Value

An object of class bfh_qic_result containing:

plot	ggplot2 object with the SPC chart
summary	data.frame with summary statistics (Danish column names). Returned as plain data.frame – the previous documentation incorrectly described it as a tibble; cycle 01 review aligned the doc with the implementation. The class is part of the public contract and will not silently flip to tibble without a deprecation cycle.
qic_data	data.frame with qicharts2 calculations
config	list with original parameters

Stability

new_bfh_qic_result is a stable, exported constructor. The structure of bfh_qic_result objects has been stable since v0.10.0. Field names ($plot, $summary, $qic_data, $config) will not be removed without a deprecation cycle.

qic_data columns

The $qic_data field is a data.frame with the following canonical columns (supplied by qicharts2 >= 0.7.0):

x Original x-axis input values

y Original y-axis input values (per-period)

n Denominator counts (for proportion/rate charts)

cl Numeric center line

ucl Numeric upper control limit

lcl Numeric lower control limit

ucl.95, lcl.95 95-percent warning limits (Shewhart sigma = 2)

sigma.signal Logical: point outside 3-sigma control limits

runs.signal Logical: Anhoej runs-rule signal

longest.run, longest.run.max Observed and threshold run lengths

n.crossings, n.crossings.min Observed and minimum crossing counts

anhoej.signal Logical combined Anhoej-rule signal (runs OR crossings). May be FALSE for series < 10 points where crossing criterion is not applied.

cl.lab, ucl.lab, lcl.lab Character labels for chart annotations

part, baseline, include Phase/freeze/filter columns

notes Free-text annotation column

The qicharts2 column contract is stable across minor versions >= 0.7.0. Additional columns may be present but should not be relied upon.

Examples

```
## Not run:
# bfh_qic() returns a bfh_qic_result via this constructor
data <- data.frame(
  month = seq(as.Date("2024-01-01"), by = "month", length.out = 12),
  infections = rpois(12, lambda = 15)
)
result <- bfh_qic(data, x = month, y = infections, chart_type = "run")
inherits(result, "bfh_qic_result")

## End(Not run)
```

---

plot.bfh_qic_result *Plot Method for bfh_qic_result*

Description

Extracts and displays the ggplot object from a bfh_qic_result. Enables use of generic plot() function.

Usage

```
## S3 method for class 'bfh_qic_result'
plot(x, ...)
```

Arguments

x	A bfh_qic_result object
...	Additional arguments passed to print.ggplot

Value

The ggplot object invisibly

print.bfh_export_session *Print a BFH Export Session*

Description

Displays a one-line summary of the session: open/closed status, tmpdir location, and font path when set.

Usage

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

Arguments

x	A bfh_export_session object created with bfh_create_export_session.
...	Additional arguments (ignored).

Value

The session object invisibly, for pipe chaining.

print.bfh_qic_result *Print Method for bfh_qic_result*

Description

Displays the plot in the console/viewer, maintaining backwards-compatible behavior with previous versions that returned ggplot objects directly.

Usage

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

Arguments

x	A bfh_qic_result object
...	Additional arguments (ignored)

Value

The object invisibly for pipe chaining

print.bfh_qic_stats *Print Method for bfh_qic_stats*

Description

Prints a one-line header and the per-phase summary. No plot is drawn.

Usage

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

Arguments

x	A bfh_qic_stats object
...	Passed to print() for the summary data.frame

Value

x, invisibly

print.bfh_spc_analysis *Print Method for bfh_spc_analysis*

Description

Displays a compact summary of features, conclusions, confidence, and active caveats. Use format() for a single-line summary or as.list() to inspect raw structure.

Usage

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

Arguments

x	A bfh_spc_analysis object.
...	Ignored.

Value

x (invisibly).

Index

∗ **export-functions**
  bfh_apply_date_axis, 5
  bfh_export_batch_pdf, 11
  bfh_export_figure_pdf, 12
  bfh_figure_panel, 25
  bfh_prune_page_cache, 35
  bfh_stage_figure_page, 53
  bfh_stage_pdf_page, 55
∗ **utility-functions**
  bfh_create_typst_document, 9
  bfh_extract_spc_stats, 23
  bfh_generate_details, 31
  bfh_merge_metadata, 33
.BFHTHEME_MIN_VERSION, 3

as.list.bfh_spc_analysis, 3

bfh_analyse, 4, *30*, *32*, *53*
bfh_apply_date_axis, 5
bfh_apply_date_axis(), *12*, *14*, *27*, *35*, *55*, *57*
bfh_build_analysis_context, *4*, 6, *30*, *53*
bfh_create_export_session, 7, *13*, *20*, *59*, *63*
bfh_create_export_session(), *14*, *20*
bfh_create_typst_document, 9
bfh_create_typst_document(), *25*, *32*, *34*
bfh_export_batch_pdf, 11, *35*, *53*, *55*
bfh_export_batch_pdf(), *6*, *14*, *27*, *35*, *55*, *57*
bfh_export_figure_pdf, 12, *25*, *53, 54*
bfh_export_figure_pdf(), *6*, *12*, *20*, *27*, *35*, *55*, *57*
bfh_export_pdf, *8, 9*, *12–14*, 15, *25*, *32*, *55, 56*
bfh_export_pdf(), *7*, *9*, *12*, *14*, *22*, *34*, *57*
bfh_export_png, 21
bfh_export_png(), *20*
bfh_extract_spc_stats, *10*, 23
bfh_extract_spc_stats(), *10*, *32*, *34*, *38*, *52*
bfh_figure_panel, *14*, 25, *54*
bfh_figure_panel(), *6*, *12*, *14*, *35*, *55*, *57*
bfh_generate_analysis, *5*, *17*, 28, *32*, *53*
bfh_generate_details, *5*, *30*, 31, *53*
bfh_generate_details(), *10*, *25*, *34*
bfh_get_plot, 33
BFH_MAX_X_LABELS_TEXT, *57*
bfh_merge_metadata, *10*, 33
bfh_merge_metadata(), *10*, *25*, *32*
bfh_prune_page_cache, 35
bfh_prune_page_cache(), *6*, *12*, *14*, *27*, *55*, *57*
bfh_qic, *4, 5*, *30*, *32*, 36, *50*, *53*, *56*
bfh_qic(), *20*, *22*, *24, 25*, *50–52*
bfh_qic_result, 49
bfh_qic_stats, 50
bfh_render_analysis, *4*, *30*, *32*, 52
bfh_stage_figure_page, *25*, 53
bfh_stage_figure_page(), *6*, *12*, *14*, *27*, *35*, *57*
bfh_stage_pdf_page, *11*, *35*, *53, 54*, 55
bfh_stage_pdf_page(), *6*, *12*, *14*, *27*, *35*, *55*
bfh_subsample_label_indices, 57

close.bfh_export_session, 58

format.bfh_spc_analysis, 59

ggplot2::coord_cartesian(), *38*

is_bfh_qic_result, 59
is_bfh_spc_analysis, 60

new_bfh_qic_result, 60

plot.bfh_qic_result, 62
print.bfh_export_session, 63
print.bfh_qic_result, 63
print.bfh_qic_stats, 64
print.bfh_spc_analysis, 64