

Package: BFHcharts (via r-universe)

June 4, 2026

Title SPC Visualization for Healthcare Quality Improvement

Version 0.23.0

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 + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

SystemRequirements Quarto CLI ($\geq 1.4.0$) for PDF export via Typst templates

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

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

Depends R ($\geq 4.1.0$)

Imports BFHtheme ($\geq 0.5.1$), ggplot2 ($\geq 3.4.0$), qicharts2 ($\geq 0.7.0$), scales ($\geq 1.2.0$), lubridate ($\geq 1.9.0$), dplyr ($\geq 1.1.0$), purrr ($\geq 1.0.0$), stringr ($\geq 1.5.0$), tibble ($\geq 3.2.0$), yaml, lemon, marquee ($\geq 0.1.0$), grid, commonmark (≥ 1.9), xml2, systemfonts, rlang, svglite

Suggests testthat ($\geq 3.1.7$), vdiff, covr, pdftools ($\geq 3.3.0$), withr, bench, BFHllm, jsonlite, knitr, rmarkdown

VignetteBuilder knitr

Remotes johanreventlow/BFHtheme, johanreventlow/BFHllm

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 libxml2-dev

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

Date/Publication 2026-06-04 13:53:35 UTC

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

RemoteRef HEAD

RemoteSha ea8fce82cc262e8eb621e80fad40cec9147a1bb

Contents

as.list.bfh_spc_analysis	3
AUDIT_EVENT_AI_EGRESS	4
bfh_analyse	4
bfh_build_analysis_context	5
bfh_create_export_session	6
bfh_create_typst_document	8
bfh_export_pdf	9
bfh_export_png	16
bfh_extract_spc_stats	18
bfh_generate_analysis	20
bfh_generate_details	23
bfh_get_plot	24
BFH_MAX_X_LABELS_TEXT	25
bfh_merge_metadata	25
bfh_qic	26
bfh_qic_result	39
bfh_render_analysis	39
bfh_subsample_label_indices	40
BFHCHARTS_OPT_ALLOW_GLOBALENV_INJECT	41
BFHCHARTS_OPT_ANALYSIS_DATE	42
BFHCHARTS_OPT_AUDIT_LOG	42
BFHCHARTS_OPT_DEBUG_LABEL_PLACEMENT	42
BFHCHARTS_OPT_QUARTO_PATH	43
BFHCHARTS_OPT_SUPPRESS_UNIT_AUTO_DETECT	43
close.bfh_export_session	43
DATA_CONSENT_EXPLICIT	44
format.bfh_spc_analysis	44
is_bfh_qic_result	45
is_bfh_spc_analysis	45
LABEL_PLACEMENT_COINCIDENT_THRESHOLD_FACTOR	46
LABEL_PLACEMENT_GAP_REDUCTION_FACTORS	46
LABEL_PLACEMENT_SHELF_CENTER_THRESHOLD	46
LABEL_PLACEMENT_TIGHT_LINES_THRESHOLD_FACTOR	47
LOW_CONFIDENCE_REASONS	47
MAGNITUDE_RATIO_CAP	47
MIN_BASELINE_N	48
N_MIN	48
new_bfh_qic_result	48
plot.bfh_qic_result	50
print.bfh_export_session	51

<code>as.list.bfh_spc_analysis</code>	3
<code>print.bfh_qic_result</code>	51
<code>print.bfh_spc_analysis</code>	52
Index	53

`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

<code>x</code>	A <code>bfh_spc_analysis</code> object.
<code>...</code>	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)
```

AUDIT_EVENT_AI_EGRESS *Audit event type for AI-egress events*

Description

Used as the event field of audit records produced by `.emit_audit_event()` when AI analysis is invoked.

Usage

AUDIT_EVENT_AI_EGRESS

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 primær canonical-output for struktureret SPC-analyse; rentekst-rendering sker via `bfh_render_analysis()`.

Usage

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

Arguments

<code>x</code>	A <code>bfh_qic_result</code> object from <code>bfh_qic()</code> .
<code>metadata</code>	Optional named list. Kan indeholde: <ul style="list-style-type: none"> <code>data_definition</code>: Description of what the data represents <code>target</code>: Target value (numeric or operator-prefixed character) <code>direction</code>: Optional "higher_better" / "lower_better" / "neutral" override for direction-akse (Slice 4) <code>analysis_date</code>: Date pinned for determinism (Phase 0.4) <code>hospital, department</code>: passes through to context
<code>language</code>	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 resolvede 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-XXX for schema.

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_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

<code>x</code>	A <code>bfh_qic_result</code> object from <code>bfh_qic()</code>
<code>metadata</code>	Optional list with additional context: <ul style="list-style-type: none"> <code>data_definition</code>: Description of what the data represents <code>target</code>: Target value for the metric. Accepts either numeric (backward compatible) or character with optional operator prefix ("<code><= 2,5</code>", "<code>>= 90%</code>"). Operators are parsed to derive <code>target_direction</code>. Fallback chain: when <code>metadata\$target</code> is NULL the function falls back to <code>x\$config\$target_text</code>, then <code>x\$config\$target_value</code>. This means a target supplied to <code>bfh_qic(target_text = ..., target_value = ...)</code> automatically reaches the analysis context without callers duplicating it in <code>metadata</code>. <code>hospital</code>: Hospital name <code>department</code>: 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`

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

<code>chart_image</code>	Path to chart image (SVG or PNG)
<code>output</code>	Path for output .typ file
<code>metadata</code>	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.
<code>spc_stats</code>	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 <code>bfh_qic_result</code> object.
<code>template</code>	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_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 <code>bfh_qic_result</code> object from <code>bfh_qic()</code>
output	Character string specifying the output PDF file path
metadata	List with optional metadata fields: <ul style="list-style-type: none"> • <code>hospital</code>: Hospital name (default: "Bispebjerg og Frederiksberg Hospital") • <code>department</code>: Department/unit name (optional) • <code>analysis</code>: Analysis text with findings (optional) • <code>details</code>: 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\ • <code>author</code>: Author name (optional) • <code>date</code>: Report date (default: <code>Sys.Date()</code>) • <code>data_definition</code>: Data definition text (optional) • <code>target</code>: Target value for analysis context. Must be either <code>NULL</code>, a single finite numeric (length 1, no <code>NA/Inf/NaN</code>), or a single character string (length 1, not <code>NA</code>). Multi-element vectors and non-finite numerics are rejected with an informative error. The target also auto-flows from <code>bfh_qic(target_text=, target_value=)</code> via the analysis-context fallback chain (since v0.12.0). • <code>footer_content</code>: Additional content to display below the chart (optional). Supports markdown formatting (bold, italic, line breaks). • <code>logo_path</code>: Path to hospital logo image rendered in the template's foreground slot (optional). When <code>NULL</code> (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).

<code>template</code>	Character string specifying template name (default: "bfh-diagram")
<code>template_path</code>	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).
<code>restrict_template</code>	<p>Logical. When TRUE (default), any non-NULL <code>template_path</code> is rejected with an error. Use FALSE only in trusted contexts where a custom Typst template is intentionally supplied.</p> <p>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 <code>source()</code> in trust requirements. <code>restrict_template = TRUE</code> prevents a compromised configuration pipeline from injecting a malicious template via <code>template_path</code>. Default: TRUE (production-safe – custom templates require explicit opt-in).</p> <p>Migration from BFHcharts <= 0.15.x: Callers passing <code>template_path</code> without <code>restrict_template</code> now receive a clear validation error. Migration is mechanical:</p> <pre># 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)</pre>
<code>auto_analysis</code>	Logical. If TRUE and <code>metadata\$analysis</code> is not provided, automatically generates analysis text using <code>bfh_generate_analysis()</code> . Default is FALSE for backward compatibility.
<code>use_ai</code>	<p>Logical. Controls AI usage for auto-analysis:</p> <ul style="list-style-type: none"> • FALSE (default): Use standard texts only - no external data sharing • TRUE: Use AI via BFHllm (requires BFHllm installed and <code>data_consent = "explicit"</code>; error if not satisfied) <p>Only used when <code>auto_analysis = TRUE</code>. See <code>bfh_generate_analysis()</code> for security policy details.</p>
<code>data_consent</code>	Character. Required when <code>use_ai = TRUE</code> and <code>auto_analysis = TRUE</code> . Must be "explicit" to acknowledge that clinical data is sent to BFHllm. Passed through to <code>bfh_generate_analysis()</code> . Ignored when <code>use_ai = FALSE</code> . 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 <code>ggplot2::ggsave()</code> . 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 <code>--font-path</code> 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 <code>--ignore-system-fonts</code> to Typst so only fonts from <code>font_path</code> (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., <code><temp_dir>/bfh-template</code>). 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 <code>batch_session</code> (pass <code>inject_assets</code> to <code>bfh_create_export_session()</code> instead).
batch_session	Optional <code>bfh_export_session</code> object from <code>bfh_create_export_session()</code> . When provided, the packaged template assets are reused from the session <code>tmpdir</code> instead of being copied on every call, which eliminates the dominant I/O cost in batch workflows. <ul style="list-style-type: none"> • Cannot be combined with <code>template_path</code> or <code>inject_assets</code>. • <code>font_path</code> here overrides <code>session\$font_path</code>. • Close the session with <code>close(session)</code> after the batch is done.
strict_baseline	Logical. When TRUE (default), the function errors before render if <code>x\$config\$freeze < MIN_BASELINE_N (8)</code> or if any phase in <code>x\$qic_data</code> contains fewer than <code>MIN_BASELINE_N</code> points. When FALSE, the export proceeds and the legacy warning-only behavior of <code>bfh_qic()</code> is preserved. <p>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 <code>bfh_qic()</code> path remains warning-only because the analyst is present.</p> <p>Inheritance: When <code>batch_session</code> is supplied without an explicit per-call value, the session's <code>strict_baseline</code> is used. An explicit per-call value overrides the session.</p>

Details**Requirements:**

- Quarto CLI ($\geq 1.4.0$) 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_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

<code>x</code>	A <code>bfh_qic_result</code> object from <code>bfh_qic()</code>
<code>output</code>	Character string specifying the output file path (e.g., "chart.png")
<code>width_mm</code>	Numeric. Width of the output image in millimeters (default: 200mm)
<code>height_mm</code>	Numeric. Height of the output image in millimeters (default: 120mm)
<code>dpi</code>	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 \approx 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_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_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 = "da",
  texts_loader = NULL
)
```

Arguments

x	A bfh_qic_result object from bfh_qic()
metadata	Optional list with additional context for AI: <ul style="list-style-type: none"> • 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". <ul style="list-style-type: none"> • 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 ($\text{mean}((\text{UCL} - \text{LCL}) / 6)$ over the last phase, with $\text{sd}(y)$ 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, default) or "en" (English). Default "da" preserves backward compatibility.
texts_loader	Function that returns SPC analysis text templates. Defaults to load_spc_texts(language). Primarily intended for tests/mockings.

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()`.

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 <code>bfh_qic_result</code> object from <code>bfh_qic()</code>
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 \u2013 december") rather than the numeric sequence stored in <code>qic_data\$x</code> . Use this when <code>text-x</code> has been converted to a numeric sequence by upstream pipelines. Length must match <code>nrow(qic_data)</code> . 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-chart, u-chart: Shows numerator/denominator (e.g., "58938/97266")
- 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", "kvarter", "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

[bfh_export_pdf\(\)](#) for PDF export functionality

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

 BFH_MAX_X_LABELS_TEXT *Maximum Visible Text X-Axis Labels*

Description

Default cap on number of categorical x-axis labels rendered horizontally without rotation. Drives `bfh_subsample_label_indices()` when caller does not override `max_visible`. Threshold chosen to fit standard A4 PDF export width (200mm @ 300dpi) with horizontal Roboto Medium labels.

Usage

BFH_MAX_X_LABELS_TEXT

 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

<code>metadata</code>	Named list with user-provided metadata fields. Valid fields: <code>hospital</code> , <code>department</code> , <code>title</code> , <code>analysis</code> , <code>details</code> , <code>author</code> , <code>date</code> , <code>data_definition</code> , <code>footer_content</code> , <code>logo_path</code> . Other fields are ignored.
<code>chart_title</code>	Character string with chart title. Used as default for <code>metadata\$title</code> 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_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"),
  base_size = 14,
  width = NULL,
  height = NULL,
  units = NULL,
  dpi = 96,
  plot_margin = NULL,
  ylab = "",
  xlab = "",
  subtitle = NULL,
  caption = NULL,
  return.data = FALSE,
  language = "da"
)
```

Arguments

<code>data</code>	Data frame with measurements
<code>x</code>	Name of x-axis column (unquoted, NSE). Usually date/time column.
<code>y</code>	Name of y-axis column (unquoted, NSE). The measurement variable.
<code>n</code>	Name of denominator column for ratio charts (optional, unquoted, NSE)
<code>chart_type</code>	Chart type: "run", "i", "mr", "p", "pp", "u", "up", "c", "g", "xbar", "s", "t"
<code>y_axis_unit</code>	Unit type: "count", "percent", "rate", or "time"
<code>chart_title</code>	Plot title (optional)
<code>target_value</code>	Numeric target value (optional)
<code>target_text</code>	Target label text (optional)
<code>notes</code>	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 <code>bfh_qic_result\$summary</code> carries <code>attr(summary, "cl_user_supplied") = TRUE</code> . Downstream consumers (e.g. PDF rendering, biSPCharts UI) can read the attribute via <code>isTRUE(attr(result\$summary, "cl_user_supplied"))</code> . The <code>bfh_export_pdf()</code> PDF surfaces a caveat note below the SPC table when this flag is set. See <code>bfh_extract_spc_stats()</code> for the discoverable surface.
ylim	Y-axis display limits as a numeric vector of length 2, <code>c(min, max)</code> , applied via <code>ggplot2::coord_cartesian()</code> (default: NULL = data-driven). Values are in the plot's data scale: proportion charts (p, pp) use the 0-1 proportion scale (e.g. <code>c(0, 1)</code> shows 0%-100%), all other charts use the absolute data scale. Use NA for a free end: <code>c(0, NA)</code> fixes the bottom at 0 with a data-driven top. Because <code>coord_cartesian</code> zooms rather than clips, data points and control-limit segments outside the range are NOT dropped (unlike <code>scale_y</code> limits) – the chart simply zooms to the requested window. If <code>min >= max</code> (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	Aggregation function for run/I charts with multiple observations per subgroup: "mean" (default), "median", "sum", "sd"
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 <code>c(top, right, bottom, left)</code> in mm, or (2) <code>ggplot2::margin()</code> 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	Logical. If TRUE, return the raw qic data frame instead of bfh_qic_result object. If FALSE (default), return bfh_qic_result S3 object. Legacy parameter maintained for backwards compatibility.
language	Character string specifying output language. One of "da" (Danish, default) or "en" (English). Controls three independent aspects of output (since v0.12.0): <ul style="list-style-type: none"> • 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 normalization
- `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
<code>laengste_loeb</code>	<code>longest.run</code>	<code>max()</code> per phase	per-phase
<code>laengste_loeb_max</code>	<code>longest.run.max</code>	<code>max()</code> global	global
<code>antal_kryds</code>	<code>n.crossings</code>	<code>max()</code> per phase	per-phase
<code>antal_kryds_min</code>	<code>n.crossings.min</code>	<code>max()</code> global	global
<code>anhoej_signal</code>	<code>runs.signal</code>	<code>any()</code> per phase	per-phase
<code>runs_signal</code>	(derived)	<code>laengste_loeb > laengste_loeb_max</code> per phase	per-phase
<code>crossings_signal</code>	(derived)	<code>antal_kryds < antal_kryds_min</code> per phase	per-phase
<code>sigma_signal</code>	<code>sigma.signal</code>	<code>any()</code> per phase	per-phase
<code>centerlinje</code>	<code>cl</code>	first row per phase	per-phase
<code>nedre_kontrolgraense</code>	<code>lcl</code>	first row per phase	per-phase
<code>oevre_kontrolgraense</code>	<code>ucl</code>	first row per phase	per-phase
<code>nedre_kontrolgraense_min/max</code>	<code>lcl</code>	<code>min()/max()</code> per phase	per-phase
<code>oevre_kontrolgraense_min/max</code>	<code>ucl</code>	<code>min()/max()</code> per phase	per-phase
<code>kontrolgraenser_konstante</code>	<code>lcl+ucl</code>	all-equal check per phase	per-phase

antal_observations	n.obs	first row per phase	per-phase
anvendelige_observations	n.useful	first row per phase	per-phase
antal_outliers	sigma.signal	sum() per phase	per-phase
forventede_outliers	(constant 0)	literal 0	per-phase

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)

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 $y \leq n$ (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: $\sqrt{\text{width} \times \text{height}} / 3.5$. 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 $[\emptyset, 1.5]$ (proportion)
- `multiply = 100`: `target_value` must be in $[\emptyset, 150]$ (percent)
- `multiply = m`: `target_value` must be in $[\emptyset, m * 1.5]$

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)"
)

## 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_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- sætter 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/mockings.

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.

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_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 $\text{ceiling}((n_labels - 1) / (\text{max_visible} - 1))$. 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 $\leq n_labels$. 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 $n \leq \text{max_visible}$ 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 $\text{step}=3$ to a $\text{step}=2$ 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 $\leq n_labels$. Length $\leq \text{max_visible}$.

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
```

BFHCHARTS_OPT_ALLOW_GLOBALENV_INJECT

Option name: opt-out for globalenv inject_assets warning

Description

When TRUE, suppresses the warning emitted by `.validate_inject_assets()` for functions defined in `.GlobalEnv` (development convenience).

Usage

```
BFHCHARTS_OPT_ALLOW_GLOBALENV_INJECT
```

BFHCHARTS_OPT_ANALYSIS_DATE

Option name: analysis_date override

Description

When set, `.resolve_analysis_date()` uses this Date as the analysis anchor instead of `Sys.Date()`. Lower priority than per-call metadata\$analysis_date. Intended for test-suites (set in `setup.R`) and audit-replay scenarios where determinism across calendar days matters.

Usage

BFHCHARTS_OPT_ANALYSIS_DATE

BFHCHARTS_OPT_AUDIT_LOG

Option name: path to audit log file

Description

When set, `.emit_audit_event()` appends JSON-line records to this path. Otherwise events are emitted via `message()`.

Usage

BFHCHARTS_OPT_AUDIT_LOG

BFHCHARTS_OPT_DEBUG_LABEL_PLACEMENT

Option name: debug logging for label-placement fallback paths

Description

When TRUE, `place_two_labels_npc()` emits diagnostic messages when the niveau cascade falls back to legacy NPC-based gap calculation.

Usage

BFHCHARTS_OPT_DEBUG_LABEL_PLACEMENT

`BFHCHARTS_OPT_QUARTO_PATH`*Option name: explicit Quarto binary path*

Description

When set, `find_quarto()` uses this path instead of running auto-detection. The path is shell-metachar validated and existence-checked before use.

Usage`BFHCHARTS_OPT_QUARTO_PATH`

`BFHCHARTS_OPT_SUPPRESS_UNIT_AUTO_DETECT`*Option name: suppress unit auto-detection message*

Description

When TRUE, `bfh_export_pdf()` and friends do not emit the informational message "Auto-detected units: ..." when units are inferred from filename.

Usage`BFHCHARTS_OPT_SUPPRESS_UNIT_AUTO_DETECT`

`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.

DATA_CONSENT_EXPLICIT *Required value for data_consent when use_ai = TRUE*

Description

Caller must pass exactly this string to acknowledge AI-egress of clinical data. Used by `bfh_generate_analysis()`.

Usage

DATA_CONSENT_EXPLICIT

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.

LABEL_PLACEMENT_COINCIDENT_THRESHOLD_FACTOR
Coincident-lines threshold factor

Description

When $\text{abs}(yA_{\text{npc}} - yB_{\text{npc}}) < \text{label_height_npc} * \text{THIS_FACTOR}$, lines are treated as effectively coincident; labels are placed one above and one below the same line position.

Usage

LABEL_PLACEMENT_COINCIDENT_THRESHOLD_FACTOR

LABEL_PLACEMENT_GAP_REDUCTION_FACTORS
Gap-reduction factors for NIVEAU 1 collision resolution

Description

When the initial label placement creates a line-gap collision, NIVEAU 1 incrementally shrinks the inter-label gap by these factors (50%, then 30%, then 15% of the configured gap_labels). The first factor that resolves the collision is used.

Usage

LABEL_PLACEMENT_GAP_REDUCTION_FACTORS

LABEL_PLACEMENT_SHELF_CENTER_THRESHOLD
Shelf-center threshold for NIVEAU 3 placement

Description

During NIVEAU 3 (last-resort shelf placement), the non-priority label is pushed to the opposite shelf (top vs bottom of panel) based on whether the priority label center is below this NPC threshold.

Usage

LABEL_PLACEMENT_SHELF_CENTER_THRESHOLD

 LABEL_PLACEMENT_TIGHT_LINES_THRESHOLD_FACTOR

Tight-lines threshold factor for early flip-strategy

Description

When $\text{abs}(yA_{\text{npc}} - yB_{\text{npc}}) < \text{min_center_gap} * \text{THIS_FACTOR}$, lines are considered too close for both labels to share a side; `pref_pos` is rewritten so one label sits above and the other below.

Usage

LABEL_PLACEMENT_TIGHT_LINES_THRESHOLD_FACTOR

LOW_CONFIDENCE_REASONS

Low-confidence reason enum

Description

Set af gyldige `low_confidence_reason`-vaerdier ved `confidence_tier == "low"`. Bruges af `feature-extraction(.compute_low_confidence_reason)`, `render-dispatch(.render_stability -> texts$base$not_evaluable[` og `schema-validator`. `NA_character_` er gyldig vaerdi naar `tier != "low"`.

Usage

LOW_CONFIDENCE_REASONS

 MAGNITUDE_RATIO_CAP *Magnitude-modifier ratio-cap*

Description

Safety-cap for `.compute_magnitude()` `baseline-delta-sigma-ratio`. Microscopic `sigma` ($\sim 1e-12$) fra near-constant data combineret med float-noise `baseline-delta` kan producere astronomical ratios and falsk "large"-klassifikation. `Ratio > cap` returnerer NA i stedet for `magnitude-bucket`. Default 100 svarer til 100 `sigma-shifts` – empirisk usandsynligt klinisk-meaningful + klart-ufysisk for kontrolgraense-baseret SPC.

Usage

MAGNITUDE_RATIO_CAP

Details

Cycle 04 H2 fix (2026-05-18). Foreloebigt safety-net; permanent loesning kraever `sigma-floor` med `scale/unit-awareness`.

MIN_BASELINE_N	<i>Minimum recommended baseline observations for stable SPC control limits</i>
----------------	--

Description

The Anhoej rules and SPC literature (Anhoej & Olesen 2014) require approximately 8+ points for meaningful control limits and reliable signal detection. Below this threshold, the control limits are statistically unreliable.

Usage

MIN_BASELINE_N

N_MIN	<i>Minimum observation count for full Anhoej-evaluability</i>
-------	---

Description

Below this threshold, confidence_tier collapses to "low" and the renderer substitutes the not_evaluable-base. Default value follows Anhoej & Olesen (2014) detection-power analysis: run-detection-power drops sharply below n = 12.

Usage

N_MIN

Details

Anhoej J, Olesen AV (2014). Run charts revisited: a simulation study of run chart rules for detection of non-random variation in health care processes. PLoS One. 9(11):e113825.

new_bfh_qic_result	<i>Create a bfh_qic_result Object</i>
--------------------	---------------------------------------

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

<code>plot</code>	ggplot2 object containing the SPC chart
<code>summary</code>	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.
<code>qic_data</code>	data.frame with raw qicharts2 calculation results
<code>config</code>	list with original function parameters

Value

An object of class `bfh_qic_result` containing:

<code>plot</code>	ggplot2 object with the SPC chart
<code>summary</code>	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.
<code>qic_data</code>	data.frame with qicharts2 calculations
<code>config</code>	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` $\geq 0.7.0$):

- 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 $\geq 0.7.0$. 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_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 <code>bfh_spc_analysis</code> object.
...	Ignored.

Value

x (invisibly).

Index

* utility-functions

- bfh_create_typst_document, 8
 - bfh_extract_spc_stats, 18
 - bfh_generate_details, 23
 - bfh_merge_metadata, 25
- as.list.bfh_spc_analysis, 3
- AUDIT_EVENT_AI_EGRESS, 4
- bfh_analyse, 4
- bfh_build_analysis_context, 5
- bfh_create_export_session, 6, 14, 44, 51
- bfh_create_export_session(), 15
- bfh_create_typst_document, 8
- bfh_create_typst_document(), 19, 24, 26
- bfh_export_pdf, 7, 8, 9
- bfh_export_pdf(), 6, 8, 17, 24, 26
- bfh_export_png, 16
- bfh_export_png(), 14
- bfh_extract_spc_stats, 8, 9, 18
- bfh_extract_spc_stats(), 9, 24, 26
- bfh_generate_analysis, 11, 12, 20
- bfh_generate_details, 23
- bfh_generate_details(), 9, 19, 26
- bfh_get_plot, 24
- BFH_MAX_X_LABELS_TEXT, 25, 40
- bfh_merge_metadata, 8, 9, 25
- bfh_merge_metadata(), 9, 19, 24
- bfh_qic, 26, 39
- bfh_qic(), 14, 17–19
- bfh_qic_result, 39
- bfh_render_analysis, 39
- bfh_subsample_label_indices, 40
- bfh_subsample_label_indices(), 25
- BFHCHARTS_OPT_ALLOW_GLOBALENV_INJECT, 41
- BFHCHARTS_OPT_ANALYSIS_DATE, 42
- BFHCHARTS_OPT_AUDIT_LOG, 42
- BFHCHARTS_OPT_DEBUG_LABEL_PLACEMENT, 42
- BFHCHARTS_OPT_QUARTO_PATH, 43
- BFHCHARTS_OPT_SUPPRESS_UNIT_AUTO_DETECT, 43
- close.bfh_export_session, 43
- DATA_CONSENT_EXPLICIT, 44
- format.bfh_spc_analysis, 44
- ggplot2::coord_cartesian(), 28
- is_bfh_qic_result, 45
- is_bfh_spc_analysis, 45
- LABEL_PLACEMENT_COINCIDENT_THRESHOLD_FACTOR, 46
- LABEL_PLACEMENT_GAP_REDUCTION_FACTORS, 46
- LABEL_PLACEMENT_SHELF_CENTER_THRESHOLD, 46
- LABEL_PLACEMENT_TIGHT_LINES_THRESHOLD_FACTOR, 47
- LOW_CONFIDENCE_REASONS, 47
- MAGNITUDE_RATIO_CAP, 47
- MIN_BASELINE_N, 48
- N_MIN, 48
- new_bfh_qic_result, 48
- plot.bfh_qic_result, 50
- print.bfh_export_session, 51
- print.bfh_qic_result, 51
- print.bfh_spc_analysis, 52