Package: zhncommandR (via r-universe)

November 3, 2026

Type Package

Title Audit and Analysis Dashboard for Haematological Oncology Cohorts

Version 0.1.0

Description Reads tumour-documentation spreadsheets from haematological tumour centres and computes quality and coverage indicators, OPS-8-544 complex chemotherapy block counts, OPS-1-941 complex diagnostics counts, Kaplan-Meier analyses, oncoprint and cytogenetics summaries, plus an interactive Shiny dashboard for auditor live evaluation. Includes a fully synthetic example cohort for demonstration and testing.

License MIT + file LICENSE

URL https://github.com/CTTIR/zhncommandR, https://cttir.github.io/zhncommandR/

BugReports https://github.com/CTTIR/zhncommandR/issues

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Depends R (>= 4.1)

Imports bslib, bsicons, cli, dplyr, DT, ggplot2, ggsurvfit, janitor, lifecycle, lubridate, ragg, readxl, rlang, scales, shiny, stats, stringr, survival, systemfonts, tidyr, utils

Suggests covr, knitr, openxlsx, pkgdown, rmarkdown, sass, shiny.i18n, shinytest2, testthat (>= 3.0.0), thematic, withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website r-lib/pkgdown, CTTIR/themakR

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev zlib1g-dev

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

Date/Publication 2026-10-28 07:29:24 UTC

RemoteUrl https://github.com/CTTIR/zhncommandR

RemoteRef HEAD

RemoteSha 4333b15ae8b334653579f8f3fbd96e8e45374d58

Contents

new_cohort_df . . . 2
new_diagnostic_blocks . . . 3
new_therapy_blocks . . . 4
zhn_alteration_type . . . 4
zhn_example_path . . . 5
zhn_is_mutation . . . 6
zhn_km_pairwise . . . 6
zhn_normalize_alteration . . . 7
zhn_pal . . . 8
zhn_parse_cytogenetics . . . 8
zhn_parse_oncoprint . . . 9
zhn_plot_cases_by_year . . . 10
zhn_plot_diagnoses . . . 10
zhn_plot_km . . . 11
zhn_prepare_diagnostic_blocks . . . 13
zhn_prepare_therapy_blocks . . . 13
zhn_read_cohort . . . 14
zhn_read_diagnostics . . . 15
zhn_read_therapy . . . 16
zhn_read_tumorboard . . . 16
zhn_register_fonts . . . 17
zhn_run_app . . . 17
zhn_save_plot . . . 18
zhn_theme . . . 19

Index **20**

`new_cohort_df` *Cohort data frame*

Description

A thin S3 wrapper around a data frame returned by `zhn_read_cohort()`. Inherits from `data.frame` so all base subscripting and dplyr verbs work; the class is preserved across `[`-subsetting but stripped by most dplyr verbs (use `tibble::as_tibble()` to opt out cleanly).

Usage

```
new_cohort_df(x = data.frame(), sheet_to_use = NA_character_)
```

Arguments

x	A data frame.
sheet_to_use	Name of the workbook sheet the data came from (carried as an attribute).

Value

A cohort_df (S3 class), inheriting from data.frame.

See Also

Other classes: new_diagnostic_blocks(), new_therapy_blocks()

---

new_diagnostic_blocks *Diagnostic-blocks data frame*

Description

S3 wrapper for the output of zhn_prepare_diagnostic_blocks().

Usage

```
new_diagnostic_blocks(x = data.frame(), component_cols = "")
```

Arguments

x	A data frame.
component_cols	Character; comma-joined list of detected component columns (Morphologie / Immunphaenotypisierung / Zytogenetik / Molekulargenetik). Carried as an attribute.

See Also

Other classes: new_cohort_df(), new_therapy_blocks()

new_therapy_blocks *Therapy-blocks data frame*

Description

S3 wrapper for the output of zhn_prepare_therapy_blocks().

Usage

```
new_therapy_blocks(x = data.frame(), patient_cols_used = "")
```

Arguments

x	A data frame.
patient_cols_used	Character; pipe-joined list of raw patient columns coalesced into patient (carried as an attribute).

See Also

Other classes: new_cohort_df(), new_diagnostic_blocks()

zhn_alteration_type *Classify a free-text alteration string*

Description

Domain rules for separating true mutations/variants from structural and cytogenetic findings, and from negative/missing entries. The categories returned drive the oncoprint vs. cytogenetics-table split in the dashboard.

Usage

```
zhn_alteration_type(x)
```

Arguments

x	Character vector of free-text alteration strings.

Details

The classification regex is clinical-domain knowledge — do not "improve" it without a corresponding update to the test fixtures.

Value

Character vector the same length as `x`, drawn from: `"Nicht verwertbar/NA"`, `"negativ/kein Nachweis"`, `"Strukturell/Zytogenetik: Deletion/Loss"`, `"Strukturell/Zytogenetik: Zugewinn/Amplifikation"`, `"Strukturell/Zytogenetik: Translokation/Rearrangement/Bruch"`, `"Strukturell/Zytogenetik: Komplexer Karyotyp"`, `"Mutation/Variante"`.

See Also

Other alteration: `zhn_is_mutation()`, `zhn_normalize_alteration()`

Examples

```
zhn_alteration_type(c(
  "TP53 Mutation", "del(17p)", "Trisomie 12",
  "t(11;14)", "komplexer Karyotyp", "negativ", NA
))
```

---

zhn_example_path *Path to the bundled synthetic example workbook*

Description

Returns the on-disk path of the synthetic example `.xlsx` shipped with the package. The example is **100% fake data** — every name, diagnosis and finding is invented for demonstration and testing.

Usage

```
zhn_example_path()
```

Details

Used as the data source when the dashboard's "Beispieldaten laden" button is clicked, and in the documentation examples.

Value

Length-1 character. The path to `zhn_example.xlsx`.

Examples

```
zhn_example_path()
if (interactive()) zhn_read_cohort(zhn_example_path())
```

zhn_is_mutation *Logical: should this alteration class appear in the oncoprint?*

Description

Convenience predicate. Only true mutations/variants populate the oncoprint tile plot; structural/cytogenetic findings are reported in their own tabular and cytogenetics-tab views.

Usage

```
zhn_is_mutation(alteration_class)
```

Arguments

alteration_class	Character vector as returned by zhn_alteration_type().

Value

Logical vector the same length as alteration_class.

See Also

Other alteration: zhn_alteration_type(), zhn_normalize_alteration()

Examples

```
zhn_is_mutation(zhn_alteration_type(c("TP53", "del(17p)")))
```

zhn_km_pairwise *Pairwise log-rank comparisons with Benjamini-Hochberg correction*

Description

For a grouping with more than two levels, computes a log-rank test for every pair of groups and adjusts the raw p-values with the BH method.

Usage

```
zhn_km_pairwise(data, time_col, event_col, group_col)
```

Arguments

data	A data frame / cohort with the time and event columns.
time_col, event_col	Column names (character) for follow-up time and event; the event is parsed to 0/1 via the package's hardened coercion.
group_col	Optional grouping column (character), or NULL for one overall curve.

Value

A data frame with `Gruppe_1`, `Gruppe_2`, `p_roh`, `p_BH`.

See Also

Other figures: `zhn_pal`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_register_fonts()`, `zhn_save_plot()`, `zhn_theme()`

`zhn_normalize_alteration`

Normalise a free-text alteration label

Description

Collapses whitespace and strips trailing "Mutation"/"Mut" suffixes used by clinicians documenting variants in free text.

Usage

```
zhn_normalize_alteration(x)
```

Arguments

`x`	Character vector.

Value

Character vector the same length as `x`.

See Also

Other alteration: `zhn_alteration_type()`, `zhn_is_mutation()`

Examples

```
zhn_normalize_alteration(c("TP53 Mutation", "  NRAS  Mut", "del(17p)"))
```

zhn_pal *zhncommandR figure palette*

Description

Single source of truth for the figure colours. Hugo Coder blue is the data-ink accent (suite purple stays the app chrome accent); revert the figures to suite purple by changing `accent` to `"#5E2C8E"` here.

Usage

```
zhn_pal
```

Format

A named list with `accent`, `accent_light`, `accent_dark`, `grid`.

See Also

Other figures: `zhn_km_pairwise()`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_register_fonts()`, `zhn_save_plot()`, `zhn_theme()`

zhn_parse_cytogenetics *Parse the cytogenetics free-text column*

Description

[Experimental]

Usage

```
zhn_parse_cytogenetics(df, remove_negative = TRUE)
```

Arguments

`df`	Cohort data frame from `zhn_read_cohort()`.
`remove_negative`	Logical; drop `"negativ/kein Nachweis"` entries (default `TRUE`).

Details

Same logic as `zhn_parse_oncoprint()` but consumes the separate `zytogenetik/karyotyp/fish` column. Used by the cytogenetics tab.

Value

Data frame with patient_label, diagnose_label, alteration, alteration_class, zytogenetik_raw. Empty input returns a 0-row tibble.

See Also

Other parsers: zhn_parse_oncoprint(), zhn_prepare_diagnostic_blocks(), zhn_prepare_therapy_blocks()

---

zhn_parse_oncoprint *Parse the oncoprint free-text mutation column*

Description

[Experimental]

Usage

```
zhn_parse_oncoprint(df, remove_negative = TRUE)
```

Arguments

df	Cohort data frame from zhn_read_cohort().
remove_negative	Logical; drop "negativ/kein Nachweis" entries (default TRUE).

Details

Splits the free-text mutation column into per-alteration rows, classifies each entry via zhn_alteration_type(), and flags genuine mutations for the oncoprint tile plot.

Unlike the legacy v5 implementation, this function does not call shiny::validate() – packaged functions must work outside Shiny. If the required source column is missing, the function raises a cli::cli_abort so the server layer can wrap the message with validate(need(...)).

Value

A data frame with one row per (patient, alteration) pair: patient_label, diagnose_label, alteration, alteration_class, alteration_raw, oncoprint_mutation. Empty input returns a 0-row tibble with this column set.

See Also

Other parsers: zhn_parse_cytogenetics(), zhn_prepare_diagnostic_blocks(), zhn_prepare_therapy_blocks()

zhn_plot_cases_by_year *Figure: cases per treatment year (vertical bar)*

Description

Figure: cases per treatment year (vertical bar)

Usage

```
zhn_plot_cases_by_year(
  data,
  year_col = "behandlungsjahr",
  transparent = FALSE,
  xlab = "Jahr",
  ylab = "Anzahl"
)
```

Arguments

`data`	A data frame with a treatment-year column.
`year_col`	Name of the year column.
`transparent`	Passed to `zhn_theme()`.
`xlab`, `ylab`	Axis labels.

Value

A ggplot2::ggplot object.

See Also

Other figures: `zhn_km_pairwise()`, `zhn_pal`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_register_fonts()`, `zhn_save_plot()`, `zhn_theme()`

zhn_plot_diagnoses *Figure: top diagnoses / codings (horizontal bar)*

Description

Counts the most frequent values of a diagnosis/coding column and draws a horizontal bar chart in the Hugo Coder look. Title-less by design – the name lives in the card header and the export filename, not in the plot.

Usage

```
zhn_plot_diagnoses(data, col, n = 20, transparent = FALSE, ylab = "Anzahl")
```

Arguments

data	A data frame (typically the filtered cohort).
col	Name of the diagnosis/coding column to count.
n	Number of top categories to show.
transparent	Passed to zhn_theme().
ylab	Count-axis label.

Value

A ggplot2::ggplot object.

See Also

Other figures: zhn_km_pairwise(), zhn_pal, zhn_plot_cases_by_year(), zhn_plot_km(), zhn_register_fonts(), zhn_save_plot(), zhn_theme()

---

zhn_plot_km *Publication-ready Kaplan-Meier plot (ggsurvfit)*

Description

Builds a themed Kaplan-Meier figure with **ggsurvfit**. Every publication element is an argument; the same call drives the live app and the PNG/PDF export. With a risk table the return value is the aligned plot+table composite (ggsurvfit::ggsurvfit_build() output) so zhn_save_plot() writes the full figure. This is the one figure that keeps an (editable) title.

Usage

```
zhn_plot_km(
  data,
  time_col,
  event_col,
  group_col = NULL,
  conf_int = TRUE,
  conf_level = 0.95,
  conf_type = "log-log",
  risktable = TRUE,
  risktable_stats = "n.risk",
  censor_marks = TRUE,
  show_pvalue = TRUE,
  show_hr = FALSE,
  show_median = FALSE,
```

```
  y_scale = c("survival", "percent", "cuminc"),
  x_break = NULL,
  x_max = NULL,
  legend_pos = "top",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  transparent = FALSE
)
```

Arguments

`data`	A data frame / cohort with the time and event columns.
`time_col, event_col`	Column names (character) for follow-up time and event; the event is parsed to 0/1 via the package's hardened coercion.
`group_col`	Optional grouping column (character), or `NULL` for one overall curve.
`conf_int`	Logical; show confidence-interval ribbons.
`conf_level`	CI level (e.g. 0.90/0.95/0.99).
`conf_type`	CI method: `"log-log"` (default), `"log"`, or `"plain"`.
`risktable`	Logical; show the numbers-at-risk table below the plot.
`risktable_stats`	Which rows: any of `"n.risk"`, `"n.censor"`, `"n.event"`.
`censor_marks`	Logical; show censoring tick marks.
`show_pvalue`	Logical; annotate the log-rank p-value (>= 2 groups).
`show_hr`	Logical; annotate the Cox HR with 95% CI (exactly 2 groups), always with a `cox.zph` proportional-hazards check and a caution note when violated.
`show_median`	Logical; draw reference lines at the median survival.
`y_scale`	`"survival"` (0-1), `"percent"` (0-100), or `"cuminc"` (1 - S).
`x_break`	Numeric x-axis break interval, or `NULL` for automatic.
`x_max`	Numeric x-axis truncation, or `NULL` for full range.
`legend_pos`	`"top"`, `"right"`, `"bottom"`, or `"none"`.
`title, subtitle, caption`	Character or `NULL`.
`transparent`	Logical; transparent background for display/export.

Value

A **ggsurvfit**/ggplot object; a patchwork composite when `risktable = TRUE`. Carries an `"n_groups"` attribute.

See Also

Other figures: `zhn_km_pairwise()`, `zhn_pal`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_register_fonts()`, `zhn_save_plot()`, `zhn_theme()`

Examples

```
if (requireNamespace("ggsurvfit", quietly = TRUE)) {
  co <- zhn_read_cohort(zhn_example_path(), verbose = FALSE)
  zhn_plot_km(co, "os", "death_event")                # default single curve
  zhn_plot_km(co, "os", "death_event", group_col = "geschlecht",
              show_pvalue = TRUE)                      # grouped + log-rank
}
```

`zhn_prepare_diagnostic_blocks`

Prepare OPS-1-941 complex-diagnostics blocks

Description

[Experimental]

Usage

```
zhn_prepare_diagnostic_blocks(df)
```

Arguments

`df`	Data frame returned by `zhn_read_diagnostics()`.

Value

A diagnostic_blocks S3 object with `patient`, `diagnose`, `datum`, `primaerfall`, `patientenfall`, `jahr`, `monat_sort`, plus the component columns (Morphologie, Immunphaenotypisierung, Zytogenetik, Molekulargenetik). Empty input returns a 0-row object.

See Also

Other parsers: `zhn_parse_cytogenetics()`, `zhn_parse_oncoprint()`, `zhn_prepare_therapy_blocks()`

`zhn_prepare_therapy_blocks`

Prepare OPS-8-544 therapy blocks for downstream analysis

Description

[Experimental]

Usage

```
zhn_prepare_therapy_blocks(df)
```

Arguments

df	Data frame returned by zhn_read_therapy().

Details

Reshapes the raw therapy sheet (as returned by zhn_read_therapy()) into the per-block tibble the dashboard counts. Counts every numeric OPS-8-544 entry > 0 as a positive block; falls back to deduplicating Patient/Datum/Zyklus/Protokoll combinations when no OPS column exists.

Value

A therapy_blocks S3 object (inheriting from data.frame) with at minimum: therapieprotokoll, diagnose, patient, datum, zyklus, jahr, monat_sort. Empty input returns a 0-row object with this column set; the patient_cols_used attribute records which raw columns were coalesced into patient.

See Also

Other parsers: zhn_parse_cytogenetics(), zhn_parse_oncoprint(), zhn_prepare_diagnostic_blocks()

---

zhn_read_cohort *Read the cohort sheet (Basisdaten / Faelle)*

Description

[Experimental]

Usage

```
zhn_read_cohort(path, sheet = NULL, verbose = TRUE)
```

Arguments

path	Path to the .xlsx file (typically a fileInput() temp path or the bundled example, obtained via zhn_example_path()).
sheet	Optional sheet name to override the canonical/regex resolver.
verbose	Logical; emit a cli_inform when duplicate columns are detected.

Details

Reads the primary cohort sheet from a tumour-documentation workbook, cleans column names via janitor::clean_names(), drops trailing empty none* columns, and derives a behandlungsjahr (treatment year) column from the first available date column.

Replaces the legacy read_main_data() from v5 which scanned ~/Desktop for Daten.xlsx. The package never reads ~/Desktop.

Value

A cohort_df S3 object (inheriting from `data.frame`) with cleaned names, an added integer `behandlungsjahr` column, and a `"sheet_to_use"` attribute. Empty input workbooks return a 0-row object.

See Also

Other readers: `zhn_read_diagnostics()`, `zhn_read_therapy()`, `zhn_read_tumorboard()`

Examples

```
if (interactive()) {
  df <- zhn_read_cohort(zhn_example_path())
  head(df)
}
```

---

zhn_read_diagnostics *Read the OPS-1-941 (complex diagnostics) sheet*

Description

Reads the diagnostics sheet — canonical name `Komplexe Diagnostik`. If missing, returns a 0-row data frame with a `"source_label"` attribute.

Usage

```
zhn_read_diagnostics(path, sheet = NULL, verbose = TRUE)
```

Arguments

`path`	Path to the `.xlsx` file (typically a `fileInput()` temp path or the bundled example, obtained via `zhn_example_path()`).
`sheet`	Optional sheet name to override the canonical/regex resolver.
`verbose`	Logical; emit a `cli_inform` when duplicate columns are detected.

Value

A data frame with cleaned names and a `"source_label"` attribute. 0-row if no diagnostics sheet exists.

See Also

Other readers: `zhn_read_cohort()`, `zhn_read_therapy()`, `zhn_read_tumorboard()`

zhn_read_therapy *Read the OPS-8-544 (complex chemotherapy) sheet*

Description

Reads the therapy-block sheet — typically the canonical name `Komplexe Chemotherapie` — from the workbook. If the sheet is genuinely missing, returns a 0-row data frame with a `"source_label"` attribute describing the situation; the dashboard surfaces this in the therapy tab.

Usage

```
zhn_read_therapy(path, sheet = NULL, verbose = TRUE)
```

Arguments

`path`	Path to the `.xlsx` file (typically a `fileInput()` temp path or the bundled example, obtained via `zhn_example_path()`).
`sheet`	Optional sheet name to override the canonical/regex resolver.
`verbose`	Logical; emit a `cli_inform` when duplicate columns are detected.

Value

A data frame with cleaned names and a `"source_label"` attribute. 0-row if no therapy sheet exists.

See Also

Other readers: `zhn_read_cohort()`, `zhn_read_diagnostics()`, `zhn_read_tumorboard()`

zhn_read_tumorboard *Read previously exported tumour-board decisions*

Description

Reads a CSV file written by the dashboard's tumour-board download. Returns an empty, typed data frame when the file is missing or empty — the contract is the same in both cases so downstream `bind_rows` is safe.

Usage

```
zhn_read_tumorboard(path = NULL)
```

Arguments

`path`	Path to a CSV (typically from a `fileInput()` upload).

Value

A data frame with columns `Patient`, `Board_Datum`, `Tumorboardbeschluss`, `Verantwortlich`, `Erfasst_am`. Always returns this column set, even with 0 rows.

See Also

Other readers: `zhn_read_cohort()`, `zhn_read_diagnostics()`, `zhn_read_therapy()`

---

zhn_register_fonts *Register the bundled Inter font for live display and exports*

Description

Registers the OFL Inter weights shipped in `inst/fonts/` with **systemfonts** so the same `family = "Inter"` renders in the live **ragg** Shiny device, in 600-dpi PNG exports (`ragg::agg_png`) and in vector PDF exports (`grDevices::cairo_pdf`). Safe to call repeatedly. If the fonts or **systemfonts** are unavailable it does nothing and figures fall back to the default device font (see `zhn_theme()`); it never errors.

Usage

```
zhn_register_fonts()
```

Value

Invisibly, `TRUE` if Inter was registered, otherwise `FALSE`.

See Also

Other figures: `zhn_km_pairwise()`, `zhn_pal`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_save_plot()`, `zhn_theme()`

---

zhn_run_app *Launch the zhncommandR Shiny dashboard*

Description

Opens the auditor live-evaluation dashboard. With no data file uploaded the app starts empty; the "Beispieldaten laden" button loads the synthetic example cohort.

Usage

```
zhn_run_app(...)
```

Arguments

...	Passed to shiny::runApp().

Details

The app is copied to a session temporary directory and run from there. shiny::runApp() sets the working directory to the app folder, so running it directly from the package library would keep that folder open and block updating or reinstalling the package on Windows ("device or resource busy") while the app is running. Running from a temp copy avoids that lock; the package's own files (translations, example data) are still read from the installed library via system.file().

Value

Invisible NULL; called for the side effect of starting the Shiny app.

Examples

```
zhn_run_app()
```

---

zhn_save_plot *Save a zhncommandR figure to PNG (600 dpi) or PDF (vector)*

Description

One save helper for every figure so resolution, device and font rules live in one place. PNG uses ragg::agg_png (sharp text, true 600 dpi, reliable transparency and font rendering); PDF uses grDevices::cairo_pdf (vector output that embeds Inter — base pdf() would not). The bundled Inter font is registered first so exported text matches the screen.

Usage

```
zhn_save_plot(
  plot,
  file,
  format = c("png", "pdf"),
  width = 8,
  height = 5,
  transparent = FALSE
)
```

Arguments

plot	A ggplot2::ggplot (or other printable plot) object.
file	Output path; written as-is.
format	One of "png" or "pdf".
width, height	Dimensions in inches.
transparent	Logical; transparent background if TRUE, else white.

Value

The `file` path, invisibly.

See Also

Other figures: `zhn_km_pairwise()`, `zhn_pal`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_register_fonts()`, `zhn_theme()`

Examples

```
p <- ggplot2::ggplot(mtcars, ggplot2::aes(factor(cyl))) +
  ggplot2::geom_bar(fill = zhn_pal$accent) + zhn_theme()
tmp <- tempfile(fileext = ".png")
zhn_save_plot(p, tmp, format = "png", width = 6, height = 4)
```

---

zhn_theme *zhncommandR figure theme (Hugo Coder look)*

Description

A single ggplot2 theme used by every `zhn_plot_*()` function so the figure look is identical everywhere and a restyle is a one-line edit. Minimal panel (light major gridlines only, no minor grid, no panel border), Inter type, and generous margins.

Usage

```
zhn_theme(base_size = 11, transparent = FALSE)
```

Arguments

`base_size`	Base font size in points.
`transparent`	If `TRUE`, the panel and plot backgrounds are fully transparent (`fill = NA`); otherwise white. Driven by the app's transparency toggle so live display and exports match.

Value

A ggplot2::theme object (add it to a plot with +).

See Also

Other figures: `zhn_km_pairwise()`, `zhn_pal`, `zhn_plot_cases_by_year()`, `zhn_plot_diagnoses()`, `zhn_plot_km()`, `zhn_register_fonts()`, `zhn_save_plot()`

Examples

```
# Returns a ggplot2 theme object; add it to any plot with `+`.
th <- zhn_theme()
th_transparent <- zhn_theme(base_size = 13, transparent = TRUE)
```

Index

∗ **alteration**
  zhn_alteration_type, 4
  zhn_is_mutation, 6
  zhn_normalize_alteration, 7
∗ **app**
  zhn_run_app, 17
∗ **classes**
  new_cohort_df, 2
  new_diagnostic_blocks, 3
  new_therapy_blocks, 4
∗ **data**
  zhn_example_path, 5
∗ **figures**
  zhn_km_pairwise, 6
  zhn_pal, 8
  zhn_plot_cases_by_year, 10
  zhn_plot_diagnoses, 10
  zhn_plot_km, 11
  zhn_register_fonts, 17
  zhn_save_plot, 18
  zhn_theme, 19
∗ **parsers**
  zhn_parse_cytogenetics, 8
  zhn_parse_oncoprint, 9
  zhn_prepare_diagnostic_blocks, 13
  zhn_prepare_therapy_blocks, 13
∗ **readers**
  zhn_read_cohort, 14
  zhn_read_diagnostics, 15
  zhn_read_therapy, 16
  zhn_read_tumorboard, 16

cohort_df, *15*

diagnostic_blocks, *13*

ggplot2::ggplot, *10, 11*, *18*
ggplot2::theme, *19*
ggsurvfit::ggsurvfit_build(), *11*

new_cohort_df, 2
new_cohort_df(), *3, 4*
new_diagnostic_blocks, 3
new_diagnostic_blocks(), *3, 4*
new_therapy_blocks, 4
new_therapy_blocks(), *3*

shiny::runApp(), *18*
shiny::validate(), *9*
system.file(), *18*

therapy_blocks, *14*
tibble::as_tibble(), *2*

zhn_alteration_type, 4
zhn_alteration_type(), *6, 7*, *9*
zhn_example_path, 5
zhn_example_path(), *14–16*
zhn_is_mutation, 6
zhn_is_mutation(), *5*, *7*
zhn_km_pairwise, 6
zhn_km_pairwise(), *8*, *10–12*, *17*, *19*
zhn_normalize_alteration, 7
zhn_normalize_alteration(), *5, 6*
zhn_pal, *7*, 8, *10–12*, *17*, *19*
zhn_parse_cytogenetics, 8
zhn_parse_cytogenetics(), *9*, *13, 14*
zhn_parse_oncoprint, 9
zhn_parse_oncoprint(), *8, 9*, *13, 14*
zhn_plot_cases_by_year, 10
zhn_plot_cases_by_year(), *7, 8*, *11, 12*, *17*, *19*
zhn_plot_diagnoses, 10
zhn_plot_diagnoses(), *7, 8*, *10*, *12*, *17*, *19*
zhn_plot_km, 11
zhn_plot_km(), *7, 8*, *10, 11*, *17*, *19*
zhn_prepare_diagnostic_blocks, 13
zhn_prepare_diagnostic_blocks(), *3*, *9*, *14*
zhn_prepare_therapy_blocks, 13
zhn_prepare_therapy_blocks(), *4*, *9*, *13*

zhn_read_cohort, 14
zhn_read_cohort(), *2*, *8, 9*, *15–17*
zhn_read_diagnostics, 15
zhn_read_diagnostics(), *13*, *15–17*
zhn_read_therapy, 16
zhn_read_therapy(), *14, 15*, *17*
zhn_read_tumorboard, 16
zhn_read_tumorboard(), *15, 16*
zhn_register_fonts, 17
zhn_register_fonts(), *7, 8*, *10–12*, *19*
zhn_run_app, 17
zhn_save_plot, 18
zhn_save_plot(), *7, 8*, *10–12*, *17*, *19*
zhn_theme, 19
zhn_theme(), *7, 8*, *10–12*, *17*, *19*