Package 'canvasXpressDashboards'

November 11, 2026

Type Package

Title Linked Interactive Dashboards from 'CanvasXpress'

Version 0.1.1

Description Renders 'canvasxpress-dashboards' specs - multi-panel, linked 'CanvasXpress' dashboards described by one JSON document - as an 'htmlwidgets' widget for 'R Markdown', 'Quarto', the 'RStudio' viewer and 'shiny'. A data frame can be turned into a whole dashboard in one call: a patient cohort with survival columns, or a per-sample sequencing run-QC table, gets filters, linked charts and a click-to-load case card. See <https://www.canvasxpress.org> for more information.

License GPL-3

Encoding UTF-8

Language en-US

URL https://github.com/neuhausi/canvasxpress-dashboards

BugReports https://github.com/neuhausi/canvasxpress-dashboards/issues

Depends R (>= 3.6)

Imports htmlwidgets (>= 1.5), htmltools, jsonlite

Suggests shiny (>= 1.5), canvasXpress, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Isaac Neuhaus [aut, cre]

Maintainer Isaac Neuhaus <imnphd@gmail.com>

Repository CRAN

Date/Publication 2026-11-11 20:30:09 UTC

Contents

cxdashboard 2
cxdashboard-proxy 3
cxdashboard-shiny 4
cxd_data 5
cxd_read_spec 5

Index **7**

`cxdashboard` *Render a canvasxpress-dashboards dashboard*

Description

Renders a multi-panel, linked dashboard described by one JSON spec - or builds a whole dashboard from a data frame. Works in R Markdown, Quarto, the RStudio viewer, ‘htmlwidgets::saveWidget()‘ and Shiny (see [cxdashboardOutput()]).

Usage

```
cxdashboard(
  spec = NULL,
  data = NULL,
  id = NULL,
  title = NULL,
  theme = NULL,
  engine = c("cdn", "canvasXpress"),
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

`spec`	A dashboard spec: a list, a JSON string, or the path of a ‘.json‘ file.
`data`	A data frame (see [cxd_data()]) or a CanvasXpress data object, to suggest a dashboard for.
`id`	With a data frame: the column holding the record names (default: the row names).
`title`	A dashboard title (overrides the spec’s, or the template’s).
`theme`	A theme to apply (‘"light"‘, ‘"dark"‘, ‘"auto"‘, ‘"editorial"‘, ...), overriding the spec’s.
`engine`	Where the CanvasXpress engine comes from: ‘"cdn"‘ (the current release from canvasxpress.org; the default, as dashboards use recent engine features) or ‘"canvasXpress"‘ (the copy bundled with the installed canvasXpress R package, for offline use; it may be older).

width, height	Widget size; the height defaults to '"auto"' (the dashboard's own layout height).
elementId	An explicit element id.

Details

Give **either** a 'spec' (rendered as written) **or** 'data' (a dashboard is suggested for it).

With 'data', the dashboard is assembled in the browser from the same templates as the dashboards wizard: a patient cohort (a numeric time column such as 'os_months' or 'time' plus a numeric 0/1 event column) gets filters, a Kaplan-Meier curve, a measure by group and against time, and a case card; a per-sample sequencing run-QC table (three or more of reads, mapping, quality, duplication, GC, insert size, coverage, adapter, contamination) gets filters, depth per sample, mapping against duplication and quality by lane or batch. Clicking a patient or sample loads its record into the case card. If no template fits, the widget says so instead of guessing; pass a 'spec'.

Arrays in a hand-built 'spec' list must stay arrays: write 'xAxis = I("age")', not 'xAxis = "age"' (a length-one vector would be sent as a plain string). A spec read from JSON (a string or a file) needs no care.

The insights strip ('insights' in a spec) is computed by the dashboards server and is not shown by this widget.

Value

An 'htmlwidget'.

Examples

```
spec <- list(
  id = "demo", title = "Demo",
  layout = list(cols = 12, rowHeight = 30,
    items = list(list(panel = "bars", x = 0, y = 0, w = 12, h = 10))),
  data = list(d = list(kind = "inline", value = cxd_data(
    data.frame(region = c("N", "S", "E"), revenue = c(10, 20, 15)), id = NULL))),
 panels = list(bars = list(dataRef = "d", config = list(graphType = "Bar", xAxis = I("revenue"))))
)
if (interactive()) cxdashboard(spec)
```

---

cxdashboard-proxy *Drive a rendered dashboard from the Shiny server*

Description

Drive a rendered dashboard from the Shiny server

Usage

```
cxdashboardProxy(outputId, session = shiny::getDefaultReactiveDomain())

cxd_set_param(proxy, name, value)

cxd_focus_panel(proxy, panel)
```

Arguments

outputId	The output id the dashboard is rendered in.
session	The Shiny session (default: the current one).
proxy	A [cxdashboardProxy()].
name	The dashboard parameter (a key of the spec's ‘params‘).
value	Its new value (‘NULL‘ clears it).
panel	A panel id of the spec.

Value

‘cxdashboardProxy()‘ a proxy object; the ‘cxd_*‘ functions return it invisibly.

Examples

```
## Not run:
observeEvent(input$load_p018, cxd_set_param(cxdashboardProxy("dash"), "case", "P018"))

## End(Not run)
```

---

cxdashboard-shiny *Shiny bindings for cxdashboard*

Description

Output and render functions, plus a proxy to drive a rendered dashboard from the server.

Usage

```
cxdashboardOutput(outputId, width = "100%", height = "auto")

renderCxdashboard(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	Output variable to read from.
width, height	CSS sizes; the height defaults to ‘"auto"‘.
expr	An expression that returns a [cxdashboard()].
env	The environment in which to evaluate ‘expr‘.
quoted	Is ‘expr‘ a quoted expression?

Details

Besides the usual widget behaviour, a dashboard reports back to the app through two inputs named after its output id: ‘input$<id>_params‘ (a named list of the dashboard's current parameters; it updates whenever a click action, control or [cxd_set_param()] changes one - for the case-card dashboards, ‘input$<id>_params$case‘ is the patient or sample last clicked) and, for a dashboard suggested from ‘data‘, ‘input$<id>_template‘ (‘"cohort"‘ or ‘"ngs-qc"‘).

Value

'cxdashboardOutput()' an output element; 'renderCxdashboard()' a render function.

`cxd_data` *Turn a data frame into a CanvasXpress data object*

Description

Dashboards bind their panels to data shaped as a CanvasXpress object: the numeric columns are the *measures* ('y', one row per column of the frame) and every other column is an *annotation* of the records ('x'). Rows are the records. This is the orientation the dashboard templates expect (more rows than numeric columns).

Usage

```
cxd_data(df, id = NULL)
```

Arguments

`df`	A data frame (or matrix).
`id`	The column holding the record names (default: the row names). The names must be unique; the column is not also used as an annotation.

Value

A list with 'y' ('vars', 'smps', 'data') and, when there are annotation columns, 'x'. Missing values are kept as 'NA' and become JSON 'null'.

Examples

```
cxd_data(data.frame(id = c("a", "b"), age = c(40, 50), arm = c("Drug", "Placebo")), id = "id")
```

`cxd_read_spec` *Read a dashboard spec*

Description

Accepts what is convenient: a list, a JSON string, or the path of a '.json' file (for example one saved from the dashboards builder or returned by the 'suggest_canvasxpress_dashboard' MCP tool).

Usage

```
cxd_read_spec(spec)
```

Arguments

`spec`	A list, a JSON string, or a file path.

Value

The spec as a list.

Index

cxd_data, 5
cxd_focus_panel *(cxdashboard-proxy)*, 3
cxd_read_spec, 5
cxd_set_param *(cxdashboard-proxy)*, 3
cxdashboard, 2
cxdashboard-proxy, 3
cxdashboard-shiny, 4
cxdashboardOutput *(cxdashboard-shiny)*, 4
cxdashboardProxy *(cxdashboard-proxy)*, 3

renderCxdashboard *(cxdashboard-shiny)*, 4