Package: DataCraftR (via r-universe)

November 20, 2026

Type Package

Title Create Datasets Interactively Using 'shiny'

Version 0.5.6

Description Set of 'shiny'-based 'RStudio' add-ins for creating datasets by drawing. Users sketch scatter plots, histograms, density curves, boxplots, bivariate normal ellipses, regression curves with error bands, radar profiles, count bar plots and two-way contingency tables (mosaic plots) in an interactive gadget, and each drawing is returned as a data.frame together with a reproducible specification that can regenerate the same data. Methods are described in: Chang et al. (2024) <https://CRAN.R-project.org/package=shiny>; Bostock et al. (2011) <doi:10.1109/TVCG.2011.185>.

License GPL (>= 3)

URL https://github.com/CarlosRivera1212/DataCraftR

BugReports https://github.com/CarlosRivera1212/DataCraftR/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports bslib, htmltools, jsonlite, rstudioapi, shiny (>= 1.8.1), stats, tools, utils

Suggests testthat (>= 3.0.0), withr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev

Repository https://carlosrivera1212.r-universe.dev

Date/Publication 2026-11-20 05:11:42 UTC

RemoteUrl https://github.com/carlosrivera1212/datacraftr

RemoteRef HEAD

RemoteSha 1c1b5916bf2a6590fa93bb548dbef1f80c51acef

Contents

dcr_bivariate . . . 2
dcr_boxplot . . . 3
dcr_count . . . 4
dcr_delete . . . 4
dcr_density . . . 5
dcr_histogram . . . 6
dcr_list . . . 6
dcr_load . . . 7
dcr_mosaic . . . 8
dcr_radar . . . 9
dcr_regression . . . 10
dcr_sample . . . 11
dcr_scatter . . . 12
dcr_spec . . . 12

Index **14**

`dcr_bivariate` *Draw bivariate normal populations and get a random sample from them*

Description

Opens the Bivariate gadget. Drag the centre, the major handle and the minor handle of each group's ellipse to shape a bivariate normal population (up to ten groups), then press **Done** to get a random sample of size `n` per group.

Usage

```
dcr_bivariate(
  groups = c("A", "B"),
  n = 200,
  xlim = c(0, 10),
  ylim = c(0, 10),
  names = c(x = "x", y = "y", group = "group")
)
```

Arguments

`groups`	Initial group names (1 to 10, unique, non-empty).
`n`	Sample size per group, a whole number from 1 to 1e6 (with `n` times the number of groups at most 1e6).
`xlim`, `ylim`	Initial axis ranges: two finite numbers, increasing.
`names`	Column names for x, y and group: a character vector of length 3, named `x`, `y`, `group` or in that order.

Value

A data.frame with x, y and a group factor whose levels are the groups, carrying its spec (see dcr_spec()); NULL invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_bivariate(c("control", "treated"), n = 300)
}
```

---

dcr_boxplot *Draw boxplots and get a random sample that matches them*

Description

Opens the Boxplot gadget. Drag the first quartile, median and third quartile of one box per variable, then press **Done** to get a random sample of size n from a two-piece distribution with exactly those quartiles.

Usage

```
dcr_boxplot(
  variables = c("V1", "V2", "V3"),
  n = 100,
  family = c("normal", "uniform"),
  ylim = c(0, 10)
)
```

Arguments

variables	Initial variable names (1 to 10, unique, non-empty).
n	Sample size, a whole number from 1 to 1e6.
family	Shape of the distribution on each side of the median: "normal" or "uniform".
ylim	Initial value range: two finite numbers, increasing.

Value

A data.frame with one numeric column per variable, carrying its spec (see dcr_spec()); NULL invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_boxplot(c("control", "treated"), n = 200, ylim = c(0, 100))
}
```

dcr_count *Draw count bars and get exactly those counts*

Description

Opens the Count gadget. Paint the integer count of every (variable, category) combination as grouped bars, then press **Done** to get a data frame with exactly those counts: one row per observation, with factor columns `variable` and `category` (levels in the drawn order).

Usage

```
dcr_count(
  variables = c("V1", "V2"),
  categories = c("C1", "C2", "C3"),
  ymax = 20
)
```

Arguments

`variables`	Initial variable names (1 to 10, unique, non-empty).
`categories`	Initial category names (1 to 10, unique, non-empty).
`ymax`	Initial top of the count axis, a whole number from 1 to 1e6.

Value

A data.frame with factor columns `variable` and `category`, carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_count(c("control", "treated"), c("low", "mid", "high"))
  table(df)
}
```

dcr_delete *Delete a saved DataCraftR dataset*

Description

Delete a saved DataCraftR dataset

Usage

```
dcr_delete(id)
```

Arguments

id	Id of a saved spec, as shown by dcr_list().

Value

invisible(TRUE).

Examples

```
if (FALSE) {
  dcr_delete("boxplot_20261117_153012")
}
```

---

dcr_density *Sculpt density curves and get a random sample that matches them*

Description

Opens the Density gadget. Shape a smooth density curve for each variable with a soft brush (**Pull** drags the curve up or down around the point you press, **Paint** moves it towards the cursor height), then press **Done** to get a random sample of size n from that distribution.

Usage

```
dcr_density(variables = "V1", n = 100, xlim = c(0, 10))
```

Arguments

variables	Initial variable names (1 to 10, unique, non-empty).
n	Sample size, a whole number from 1 to 1e6.
xlim	Initial x range: two finite numbers, increasing.

Value

A data.frame with one numeric column per variable (variables with an all-zero curve are dropped with a warning), carrying its spec (see dcr_spec()); NULL invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_density(c("control", "treated"), n = 500, xlim = c(0, 100))
}
```

dcr_histogram *Draw histograms and get a random sample that matches them*

Description

Opens the Histogram gadget. Paint the bar heights (relative weights, i.e. the shape of the population) for one histogram per variable, then press **Done** to get a random sample of size `n` from that population.

Usage

```
dcr_histogram(variables = "V1", n = 100, xlim = c(0, 10), nbins = 10)
```

Arguments

`variables`	Initial variable names (1 to 10, unique, non-empty).
`n`	Sample size, a whole number from 1 to 1e6.
`xlim`	Initial x range: two finite numbers, increasing.
`nbins`	Number of bins, a whole number from 1 to 100.

Value

A data.frame with one numeric column per variable (variables with all-zero bars are dropped with a warning), carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_histogram(c("control", "treated"), n = 500, xlim = c(0, 100))
}
```

dcr_list *List saved DataCraftR datasets*

Description

List saved DataCraftR datasets

Usage

```
dcr_list()
```

Value

A data frame with columns `id`, `tool`, `created` and `n` (the total number of rows), newest first. Files in the store that are not readable specs are ignored.

Examples

```
if (FALSE) {
  dcr_list()
}
```

`dcr_load` *Load a saved DataCraftR dataset*

Description

Tools used from the RStudio Addins menu save their spec in the user data directory (`tools::R_user_dir("DataCraftR", "data")`) and insert `df <- dcr_load("<id>")` into your script. Literal tools (scatter, count, mosaic) return exactly what was drawn. Population tools (boxplot, histogram, density, bivariate, regression, radar) re-sample using the stored seed, so the same data comes back, unless you give a different `seed`.

Usage

```
dcr_load(id, n = NULL, seed = NULL)
```

Arguments

`id`	Id of a saved spec, as shown by `dcr_list()`.
`n`	Number of rows for population tools (per group for bivariate, regression and radar); defaults to the stored `n`. The cap of 1e6 rows in total is checked when the drawing is made, not when `n` is overridden here.
`seed`	Integer seed for population tools; defaults to the stored seed.

Value

A data frame (see `dcr_sample()`).

Examples

```
if (FALSE) {
  dcr_list()
  df <- dcr_load("boxplot_20261117_153012")
}
```

`dcr_mosaic` *Draw a two-way contingency table and get exactly those counts*

Description

Opens the Mosaic gadget. Drag the vertical borders between columns to move observations between two columns (each keeps its row proportions) and drag the horizontal cuts inside a column to shift counts between its blocks. Everything snaps to whole counts and the total stays exactly `n`. Press **Done** to get a data frame with exactly those counts: one row per observation, with two factor columns (levels in the drawn order).

Usage

```
dcr_mosaic(
  cols = c("No", "Yes"),
  rows = c("Healthy", "Sick"),
  n = 200,
  names = c(col = "exposure", row = "outcome")
)
```

Arguments

`cols`, `rows`	Initial category names of the column and row variables (2 to 10 each, unique, non-empty).
`n`	Total number of observations, a whole number from 1 to 1e6. The table starts uniform.
`names`	Names of the two output columns: a character vector of length 2, named `col`, `row` or in that order.

Value

A data.frame with two factor columns named by `names` (column variable first), carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_mosaic(c("No", "Yes"), c("Healthy", "Sick"), n = 300)
  table(df)
}
```

dcr_radar *Draw group profiles on a radar chart and get a random sample*

Description

Opens the Radar gadget. Each group is a profile on a radar chart: drag a vertex along its spoke to set the mean of a variable and a band handle to set its standard deviation, then press **Done** to get a random sample of size `n` per group. Values are normal, truncated to each variable's range.

Usage

```
dcr_radar(
  variables = c("V1", "V2", "V3", "V4", "V5"),
  groups = c("A", "B"),
  n = 50,
  range = c(0, 10),
  integers = FALSE,
  group_name = "group"
)
```

Arguments

`variables`	Variable names (3 to 12, unique, non-empty).
`groups`	Initial group names (1 to 10, unique, non-empty).
`n`	Sample size per group, a whole number from 1 to 1e6 (with `n` times the number of groups at most 1e6).
`range`	Initial range of every variable: two finite numbers, increasing, or a list with one such pair per variable.
`integers`	If `TRUE`, values are rounded to whole numbers.
`group_name`	Name of the group column; it must differ from the variable names.

Value

A data.frame with a group factor (levels are the groups) and one numeric column per variable, carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_radar(c("speed", "power", "agility"), groups = c("A", "B"))
}
```

`dcr_regression` *Draw a regression curve with its error band and get a random sample*

Description

Opens the Regression gadget. Shape the mean curve `f(x)` and the error spread around it for each group (**Pull** and **Paint** brushes, or type an equation), then press **Done** to get a random sample of size `n` per group.

Usage

```
dcr_regression(
  f = "x",
  groups = "A",
  n = 200,
  xlim = c(0, 10),
  sd = 1,
  names = c(x = "x", y = "y", group = "group"),
  shared_x = TRUE
)
```

Arguments

`f`	The initial curve: one equation (a string in `x`, such as `"2 + 0.5 * x"`, or a function) used for every group, or one per group (a character vector or list). Points where an equation is undefined take the nearest finite value, with a warning.
`groups`	Initial group names (1 to 10, unique, non-empty).
`n`	Sample size per group, a whole number from 1 to 1e6 (with `n` times the number of groups at most 1e6).
`xlim`	Initial x range: two finite numbers, increasing.
`sd`	Initial error standard deviation: one value or one per group, each at least 0.
`names`	Column names for x, y and group: a character vector of length 3, named `x`, `y`, `group` or in that order.
`shared_x`	If `TRUE` all groups share the same x values.

Details

An equation given as a string or function is evaluated in R in your session, like any other R code you run.

Value

A data.frame with x, y and a group factor whose levels are the groups, carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_regression("2 + 0.5 * x", groups = c("control", "treated"))
}
```

`dcr_sample` *Sample a data frame from a DataCraftR spec*

Description

Turns a spec (what was drawn) into a data frame. Population tools draw `n` rows using `seed`; literal tools return exactly what was drawn and ignore `n` and `seed`. The session's random number state (`.Random.seed`) is left as it was.

Usage

```
dcr_sample(spec, n = NULL, seed = NULL)
```

Arguments

`spec`	A `dcr_spec` object, e.g. from `dcr_spec()`.
`n`	Number of rows for population tools (per group for bivariate, regression and radar); defaults to `spec$n`. The cap of 1e6 rows in total is checked when the drawing is made, not when `n` is overridden here.
`seed`	Optional integer seed; defaults to `spec$seed`.

Value

A data frame with attribute `"dcr_spec"` holding the spec used (with `n` and `seed` set to the values used).

Examples

```
if (FALSE) {
  df <- dcr_sample(dcr_spec(my_df), n = 200, seed = 1)
}
```

dcr_scatter *Draw a scatter plot and get its points as a data frame*

Description

Opens the Scatter gadget. Spray, place or erase points for up to ten groups, then press **Done** to get exactly the drawn points.

Usage

```
dcr_scatter(
  xlim = c(0, 10),
  ylim = c(0, 10),
  groups = c("A", "B", "C"),
  names = c(x = "x", y = "y", group = "group")
)
```

Arguments

`xlim`, `ylim`	Initial axis ranges: two finite numbers, increasing.
`groups`	Initial group names (1 to 10, unique, non-empty).
`names`	Column names for x, y and group: a character vector of length 3, named `x`, `y`, `group` or in that order.

Value

A data.frame with the drawn points (x, y and a group factor whose levels are the groups), carrying its spec (see `dcr_spec()`); `NULL` invisibly on Cancel.

Examples

```
if (interactive()) {
  df <- dcr_scatter(xlim = c(-5, 5), groups = c("control", "treated"))
}
```

dcr_spec *Recover the spec behind a DataCraftR data frame*

Description

Recover the spec behind a DataCraftR data frame

Usage

```
dcr_spec(df)
```

Arguments

df	A data frame returned by a DataCraftR tool.

Value

A dcr_spec object.

Index

dcr_bivariate, 2
dcr_boxplot, 3
dcr_count, 4
dcr_delete, 4
dcr_density, 5
dcr_histogram, 6
dcr_list, 6
dcr_list(), *5*, *7*
dcr_load, 7
dcr_mosaic, 8
dcr_radar, 9
dcr_regression, 10
dcr_sample, 11
dcr_sample(), *7*
dcr_scatter, 12
dcr_spec, 12
dcr_spec(), *3–6*, *8–12*