# Package: clitable (via r-universe)

May 19, 2026

**Type** Package

**Title** Render Tables in Text for the Terminal

**Version** 0.1.0

**Description** Render tables in text format in the terminal using ANSI strings thanks to the 'cli' and 'crayon' packages.

**License** GPL (>= 3)

**URL** https://github.com/kforner/clitable

**BugReports** https://github.com/kforner/clitable/issues

**Imports** cli, crayon, grDevices, utils

**Suggests** devtools, testthat

**Config/testthat/edition** 3

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Repository** https://kforner.r-universe.dev

**Date/Publication** 2025-10-20 08:07:34 UTC

**RemoteUrl** https://github.com/kforner/clitable

**RemoteRef** HEAD

**RemoteSha** ac4adb2d3db894a07c569f9848ee0053b55b8bdd

## Contents

clitable-package *clitable: Render Tables in Text for the Terminal*

## Description

Render tables in text format in the terminal using ANSI strings thanks to the 'cli' and 'crayon' packages.

## Features

- can display any ansi string (without end of lines) content
- multiple table border styles: single, double, single-double, double-single, classic
- can display heatmaps
- can highlight rows
- can display NAs with custom style
- few dependencies: only `crayon` and `cli`

## Author(s)

**Maintainer**: Karl Forner <karl.forner@gmail.com> [copyright holder]

## See Also

Useful links:

- https://github.com/kforner/clitable
- Report bugs at https://github.com/kforner/clitable/issues

---

cli_table *generates a text table*

## Description

generates a text table

## Usage

```
cli_table(
  mat,
  header = TRUE,
  header_style = NULL,
  border_style = "single",
  heatmap_columns = NULL,
  heatmap_colorspace = c("green", "red"),
```

```
  hilite_rows = NULL,
  hilite_style = "bgRed",
  NA_style = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| mat | the table content to print, can be a data.frame or a matrix |
| header | whether to use the row names as table headers |
| header_style | the (crayon) style to use to print the headers (cf crayon::style() |
| border_style | the style to use for the table borders, one of single, double, single-double, double-single, classic |
| heatmap_columns | the columns that should be displayed as heatmaps, as a vector of column indices, names or logicals |
| heatmap_colorspace | the colorspace to use for the heatmaps, to be passed to grDevices::colorRamp() |
| hilite_rows | the rows to highlight, as a vector of column indices, names or logicals |
| hilite_style | the (crayon) style to use to highlight the rows (cf crayon::style() |
| NA_style | the (crayon) style to use to highlight the NA values (cf crayon::style() |
| ... | Arguments passed on to scale_numeric<br>x the numeric vector to scale<br>xmin the minimum value used for the scaling. all all x < xmin are set to 0<br>xmax the maximum value used for the scaling. all x > xmax set to 1 |

## Value

the lines of the text table as an ansi_string vector

## Examples

```
df <- head(datasets::penguins, 20)
ct <- cli_table(df, header_style = "bold",
  NA_style = "strikethrough",
  heatmap_columns = list("flipper_len"), xmin = 180, xmax = 200,
  hilite_rows = !is.na(df$sex) & df$sex == "female" & df$bill_dep >= 19,
  hilite_style = "bgGreen"
)
cat(ct, sep = "\n")
```

## demo *a function to demo the clitable package*

### Description

a function to demo the clitable package

### Usage

```
demo()
```

### Value

nothing

### Examples

```
demo()
```

## scale_numeric *scales a numeric vector*

### Description

scales a numeric vector

### Usage

```
scale_numeric(x, xmin = min(x, na.rm = TRUE), xmax = max(x, na.rm = TRUE))
```

### Arguments

| | |
|---|---|
| x | the numeric vector to scale |
| xmin | the minimum value used for the scaling. all all x < xmin are set to 0 |
| xmax | the maximum value used for the scaling. all x > xmax set to 1 |

### Value

a numeric vector of the same length as x, with all values between 0 and 1, except for NAs which are unchanged

### Examples

```
x <- c(0.1, 100, -2.5, 20, 78.2, NA)
scaled <- scale_numeric(x)
all(is.na(scaled) | (scaled >= 0 & scaled <= 1))
```

# Index