

Package ‘repr’

January 4, 2022

Title Serializable Representations

Version 1.1.4

Maintainer Philipp Angerer <phil.angerer@gmail.com>

Description String and binary representations of objects for several formats /
mime types.

URL <https://github.com/IRkernel/repr/>

BugReports <https://github.com/IRkernel/repr/issues/>

Depends R (>= 3.0.1)

Imports utils, grDevices, htmltools, jsonlite, pillar (>= 1.4.0),
base64enc

Suggests methods, highr, Cairo, stringr, testthat (>= 3.0.0), leaflet

Enhances data.table, tibble, htmlwidgets, vegalite, plotly, geojsonio

Config/testthat/edition 3

License GPL-3

Encoding UTF-8

Collate 'generics.r' 'options.r' 'package.r' 'repr_datatable.r'
'repr_datetime.r' 'utils.r' 'repr_list.r' 'repr_vector.r'
'repr_factor.r' 'repr_function.r'
'repr_help_files_with_topic.r' 'repr_htmlwidget.r'
'repr_matrix_df.r' 'repr_packageIQR.r' 'repr_plotly.r'
'repr_recordedplot.r' 'repr_spatial.r' 'repr_ts.r'
'repr_vega.r'

RoxygenNote 7.1.2

NeedsCompilation no

Author Philipp Angerer [aut, cre] (<<https://orcid.org/0000-0002-0369-2888>>),
Thomas Kluyver [aut],
Jan Schulz [aut],
abielr [ctb],
Denilson Figueiredo de Sa [ctb],
Jim Hester [ctb],

karldw [ctb],
 Dave Foster [ctb],
 Carson Sievert [ctb]

Repository CRAN

Date/Publication 2022-01-04 14:20:06 UTC

R topics documented:

repr-package	2
*2repr	3
repr	3
repr-generics	4
repr-options	6
repr_*.data.table	7
repr_*.factor	8
repr_*.function	8
repr_*.help_files_with_topic	9
repr_*.htmlwidget	9
repr_*.list	10
repr_*.matrix/data.frame	11
repr_*.packageIQR	13
repr_*.recordedplot	13
repr_*.ts	15
repr_*.vector	16
repr_geojson.*	17
repr_plotly1.*	19
repr_text	19
repr_vega*	20

Index **21**

repr-package *The repr package*

Description

String and binary representations of objects for several formats / mime types.

Details

The LaTeX repr of vectors needs `\usepackage[inline]{enumitem}`

The LaTeX repr of functions with the `repr.function.highlight` option set to `FALSE` needs `\usepackage{minted}`

See Also

[repr](#), [repr-options](#), [repr-generics](#), [repr_text](#)

*2repr	<i>Lists mapping mime types (mime2repr) or format names (format2repr) to repr functions</i>
--------	---

Description

Lists mapping mime types (mime2repr) or format names (format2repr) to repr functions

Usage

mime2repr

format2repr

Format

Lists mapping mime/name to function

An object of class list of length 18.

Examples

```
names(mime2repr)
names(format2repr)
```

repr	<i>Dynamic representation</i>
------	-------------------------------

Description

Specify an object and a format to represent it in. Will [stop\(\)](#) if no such format is known.

Usage

```
repr(obj, format = "text", ...)
```

Arguments

obj	The object to create a representation for
format	The representation format. repr_<format> is then called. (default: Call repr_text)
...	delegated to the specific repr_<format> function

Value

A character or raw vector of that format or NULL if none is defined. Only the 'text' format is defined for everything (via [print\(\)](#))

See Also

[repr_text](#), [repr-generics](#)

repr-generics

Representations for specific formats

Description

Representations for specific formats

Usage

```
repr_html(obj, ...)
```

```
## Default S3 method:
```

```
repr_html(obj, ...)
```

```
repr_markdown(obj, ...)
```

```
## Default S3 method:
```

```
repr_markdown(obj, ...)
```

```
repr_latex(obj, ...)
```

```
## Default S3 method:
```

```
repr_latex(obj, ...)
```

```
repr_json(obj, ...)
```

```
## Default S3 method:
```

```
repr_json(obj, ...)
```

```
repr_javascript(obj, ...)
```

```
## Default S3 method:
```

```
repr_javascript(obj, ...)
```

```
repr_pdf(obj, ...)
```

```
## Default S3 method:
```

```
repr_pdf(obj, ...)
```

```
repr_png(obj, ...)
```

```
## Default S3 method:
```

```
repr_png(obj, ...)
```

```
repr_jpg(obj, ...)

## Default S3 method:
repr_jpg(obj, ...)

repr_svg(obj, ...)

## Default S3 method:
repr_svg(obj, ...)

repr_geojson(obj, ...)

## Default S3 method:
repr_geojson(obj, ...)

repr_vdom1(obj, ...)

## Default S3 method:
repr_vdom1(obj, ...)

repr_plotly1(obj, ...)

## Default S3 method:
repr_plotly1(obj, ...)

repr_vegalite2(obj, ...)

## Default S3 method:
repr_vegalite2(obj, ...)

repr_vegalite3(obj, ...)

## Default S3 method:
repr_vegalite3(obj, ...)

repr_vegalite4(obj, ...)

## Default S3 method:
repr_vegalite4(obj, ...)

repr_vega4(obj, ...)

## Default S3 method:
repr_vega4(obj, ...)

repr_vega5(obj, ...)
```

```
## Default S3 method:
repr_vega5(obj, ...)
```

Arguments

obj The object to create a repr for
 ... parameters of the specific repr_* functions

See Also

[repr_text](#) for the only repr that is always defined

repr-options

repr options

Description

These options are used to control the behavior of repr when not calling it directly. Use [options\(repr.* = ...\)](#) and [getOption\('repr.*'\)](#) to set and get them, respectively.

Usage

```
repr_option_defaults
```

Format

An object of class list of length 15.

Details

Once this package is loaded, all options are set to defaults which weren't set beforehand.

Setting all options set to NULL are reset to defaults when reloading the package (or calling `repr:::load()`).

Options

repr.plot.* Those are for representations of recordedplot instances:

repr.plot.width Plotting area width in inches (default: 7)

repr.plot.height Plotting area height in inches (default: 7)

repr.plot.pointsize Text height in pt (default: 12)

repr.plot.bg Background color (default: white)

repr.plot.antialias Which kind of antialiasing to use for for lines and text? 'gray', 'sub-pixel' or 'none'? (default: gray)

repr.plot.res PPI for rasterization (default: 120)

repr.plot.quality Quality of JPEG format in % (default: 90)

repr.plot.family Vector font family. 'sans', 'serif', 'mono' or a specific one (default: sans)

repr.vector.quote Output quotation marks for character vectors? (default: TRUE)

- repr.vector.max.items How many items to display at max. Will insert an item with a horizontal ellipsis to show elision. (default: 400)
- repr.matrix.max.rows How many rows to display at max. Will insert a row with vertical ellipses to show elision. (default: 60)
- repr.matrix.max.cols How many cols to display at max. Will insert a column with horizontal ellipses to show elision. (default: 20)
- repr.matrix.latex.colspec How to layout LaTeX tables when representing matrices or data.frames. List of row.head, other.col, and end strings. end mainly exists for when you want a vertical line there (default: 'r', 'l', and '')
- repr.function.highlight Use the highr package to insert highlighting instructions into the code? Needs that package to be installed. (default: FALSE)
- repr.html.deduplicate Use the [html_dependencies](#) manager to only include dependencies once? This can greatly reduce notebook size, but fails if e.g. iframes are used (default: FALSE)

repr_*.data.table *Representation of data.table objects*

Description

Representation of data.table objects

Usage

```
## S3 method for class 'data.table'
repr_html(obj, ...)

## S3 method for class 'data.table'
repr_text(obj, ...)

## S3 method for class 'data.table'
repr_latex(obj, ...)
```

Arguments

obj The list to create a representation for

... ignored

```
repr_*.factor      Representations of factors
```

Description

Representations of factors

Usage

```
## S3 method for class 'factor'
repr_html(obj, ...)
```

```
## S3 method for class 'factor'
repr_markdown(obj, ...)
```

```
## S3 method for class 'factor'
repr_latex(obj, ...)
```

Arguments

obj	The factor to create a representation for
...	ignored

```
repr_*.function   Representations of functions
```

Description

Representations of functions

Usage

```
## S3 method for class '`function`'
repr_html(obj, highlight = getOption("repr.function.highlight"), ...)
```

```
## S3 method for class '`function`'
repr_latex(obj, highlight = getOption("repr.function.highlight"), ...)
```

```
## S3 method for class '`function`'
repr_markdown(obj, fenced = TRUE, ...)
```

Arguments

obj	Function to create a representation for
highlight	Should code highlighting be performed
...	ignored
fenced	Should a fenced code block instead of an indented one be used?

```
repr_*.help_files_with_topic
    Representations of help
```

Description

Representations of help

Usage

```
## S3 method for class 'help_files_with_topic'
repr_text(obj, ...)

## S3 method for class 'help_files_with_topic'
repr_html(obj, ...)

## S3 method for class 'help_files_with_topic'
repr_latex(obj, ...)
```

Arguments

obj	Help topic to create a representation for
...	ignored

```
repr_*.htmlwidget    HTML widget representations
```

Description

Standalone HTML representation and dummy text representation.

Usage

```
html_dependencies

## S3 method for class 'htmlwidget'
repr_text(obj, ...)

## S3 method for class 'htmlwidget'
repr_html(obj, ...)

## S3 method for class 'shiny.tag'
repr_text(obj, ...)

## S3 method for class 'shiny.tag'
```

```
repr_html(obj, ...)

## S3 method for class 'shiny.tag.list'
repr_text(obj, ...)

## S3 method for class 'shiny.tag.list'
repr_html(obj, ...)
```

Arguments

obj The htmlwidget, shiny.tag, or shiny.tag.list to create a representation for
 ... ignored

Format

An object of class environment of length 4.

Details

html_dependencies is an [environment](#) containing the following functions. `getOption('repr.html.deduplicate')`

`get()` Get the list of added dependencies

`add(dep)` Marks a dependency as added. Call this e.g. after appending a script tag with the dependency.

`clear()` Clear the list as seen dependencies. Now everything will be added again when encountered.

`dir()` Returns the directory in which the dependencies reside.

repr_*.list

Representations of lists

Description

Representations of lists

Usage

```
## S3 method for class 'list'
repr_html(obj, ...)

## S3 method for class 'list'
repr_markdown(obj, ...)

## S3 method for class 'list'
repr_latex(obj, ...)
```

Arguments

obj	The list to create a representation for
...	ignored

repr_*.matrix/data.frame

Tabular data representations

Description

HTML, LaTeX, and Markdown representations of Matrix-like objects

Usage

```
## S3 method for class 'matrix'
repr_html(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

## S3 method for class 'data.frame'
repr_html(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

## S3 method for class 'matrix'
repr_latex(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols"),
  colspec = getOption("repr.matrix.latex.colspec")
)

## S3 method for class 'data.frame'
repr_latex(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols"),
  colspec = getOption("repr.matrix.latex.colspec")
)
```

```

)

## S3 method for class 'matrix'
repr_markdown(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

## S3 method for class 'data.frame'
repr_markdown(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

## S3 method for class 'matrix'
repr_text(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

## S3 method for class 'data.frame'
repr_text(
  obj,
  ...,
  rows = getOption("repr.matrix.max.rows"),
  cols = getOption("repr.matrix.max.cols")
)

```

Arguments

<code>obj</code>	The matrix or data.frame to create a representation for
<code>...</code>	ignored
<code>rows</code>	The maximum number of rows displayed. The default is given by the option <code>repr.matrix.max.rows</code>
<code>cols</code>	The maximum number of columns displayed. The default is given by the option <code>repr.matrix.max.cols</code>
<code>colspec</code>	The colspec for the LaTeX table. The default is given by the option <code>repr.matrix.latex.colspec</code>

See Also

[repr-options](#) for `repr.matrix.latex.colspec`

```
repr_*.packageIQR      packageIQR representations
```

Description

Text representations of packageIQR objects like the list of available example data or vignettes

Usage

```
## S3 method for class 'packageIQR'
repr_text(obj, ...)

## S3 method for class 'packageIQR'
repr_html(obj, ...)
```

Arguments

```
obj          The packageIQR obj to create a representation for
...          ignored
```

Examples

```
repr_html(data(package = 'datasets'))
repr_text(vignette(package = 'highr'))
```

```
repr_*.recordedplot    Plot representations
```

Description

repr_text.recordedplot only returns a small info string containing the title (if any) while the others return a character vector (SVG) or a raw vector (the rest) containing the image data.

Usage

```
## S3 method for class 'recordedplot'
repr_text(obj, ...)

## S3 method for class 'recordedplot'
repr_png(
  obj,
  width = getOption("repr.plot.width"),
  height = getOption("repr.plot.height"),
  bg = getOption("repr.plot.bg"),
```

```

    pointsize = getOption("repr.plot.pointsize"),
    antialias = getOption("repr.plot.antialias"),
    res = getOption("repr.plot.res"),
    ...
)

## S3 method for class 'recordedplot'
repr_jpg(
  obj,
  width = getOption("repr.plot.width"),
  height = getOption("repr.plot.height"),
  bg = getOption("repr.plot.bg"),
  pointsize = getOption("repr.plot.pointsize"),
  antialias = getOption("repr.plot.antialias"),
  res = getOption("repr.plot.res"),
  quality = getOption("repr.plot.quality"),
  ...
)

## S3 method for class 'recordedplot'
repr_svg(
  obj,
  width = getOption("repr.plot.width"),
  height = getOption("repr.plot.height"),
  bg = getOption("repr.plot.bg"),
  pointsize = getOption("repr.plot.pointsize"),
  antialias = getOption("repr.plot.antialias"),
  family = getOption("repr.plot.family"),
  ...
)

## S3 method for class 'recordedplot'
repr_pdf(
  obj,
  width = getOption("repr.plot.width"),
  height = getOption("repr.plot.height"),
  bg = getOption("repr.plot.bg"),
  pointsize = getOption("repr.plot.pointsize"),
  antialias = getOption("repr.plot.antialias"),
  family = getOption("repr.plot.family"),
  ...
)

```

Arguments

<code>obj</code>	The plot to create a representation for
<code>...</code>	ignored
<code>width</code>	Plot area width in inches (default: 7)

height	Plot area height in inches (default: 7)
bg	Background color (default: white)
pointsize	Text height in pt (default: 12)
antialias	Which kind of antialiasing to use for for lines and text? 'gray', 'subpixel' or 'none'? (default: gray)
res	For PNG and JPEG, specifies the PPI for rasterization (default: 120)
quality	For JPEG, determines the compression quality in % (default: 90)
family	Font family for SVG and PDF. 'sans', 'serif', 'mono' or a specific one (default: sans)

Details

All parameters can also be specified using the eponymous `repr.plot.*` [repr-options](#).

Examples

```
dev.new()
dev.control(displaylist = 'enable')
plot(sqrt, main = 'Square root')
p <- recordPlot()
dev.off()

repr_text(p)
```

```
repr_*.ts
```

Time series representations

Description

HTML, LaTeX, and Markdown representations of `ts` objects.

Usage

```
## S3 method for class 'ts'
repr_html(obj, ...)

## S3 method for class 'ts'
repr_latex(obj, ..., colspec = getOption("repr.matrix.latex.colspec"))

## S3 method for class 'ts'
repr_markdown(obj, ...)

## S3 method for class 'ts'
repr_text(obj, ...)
```

Arguments

obj	The <code>ts</code> object to create a representation for
...	ignored
colspec	The colspec for the LaTeX table. The default is given by the option <code>repr.matrix.latex.colspec</code>

See Also

[repr-options](#) for `repr.matrix.latex.colspec`

repr_*.vector	<i>Representations of vectors</i>
---------------	-----------------------------------

Description

Representations of vectors

Usage

```
## S3 method for class 'logical'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'integer'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'complex'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'numeric'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'character'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'Date'
repr_html(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'logical'
repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'integer'
repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'complex'
repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'numeric'
```



```

repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'character'
repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'Date'
repr_markdown(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'logical'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'integer'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'complex'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'numeric'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'character'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

## S3 method for class 'Date'
repr_latex(obj, ..., items = getOption("repr.vector.max.items"))

```

Arguments

obj	The vector to create a representation for
...	ignored
items	The maximum number of items displayed. The default is given by the option <code>repr.vector.max.items</code>

repr_geojson.*	<i>Representations of spatial objects: See geojson_list for supported classes.</i>
----------------	--

Description

Representations of spatial objects: See [geojson_list](#) for supported classes.

Usage

```

## S3 method for class 'geo_list'
repr_geojson(obj, ...)

## S3 method for class 'SpatialCollections'

```

```
repr_geojson(obj, ...)

## S3 method for class 'SpatialPolygons'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPolygons'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPolygonsDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPoints'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPointsDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'SpatialLines'
repr_geojson(obj, ...)

## S3 method for class 'SpatialLinesDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'SpatialGrid'
repr_geojson(obj, ...)

## S3 method for class 'SpatialGridDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPixels'
repr_geojson(obj, ...)

## S3 method for class 'SpatialPixelsDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'SpatialRings'
repr_geojson(obj, ...)

## S3 method for class 'SpatialRingsDataFrame'
repr_geojson(obj, ...)

## S3 method for class 'sf'
repr_geojson(obj, ...)

## S3 method for class 'sfg'
repr_geojson(obj, ...)

## S3 method for class 'sfc'
```

```
repr_geojson(obj, ...)
```

Arguments

obj	The spatial object to create a representation for
...	ignored

repr_plotly.*	<i>Representation as Plotly JSON.</i>
---------------	---

Description

Representation as [Plotly JSON](#).

Usage

```
## S3 method for class 'plotly'
repr_plotly1(obj, ...)
```

```
## S3 method for class 'ggplot'
repr_plotly1(obj, ...)
```

Arguments

obj	The plot_ly plot or ggplot to create a representation for
...	ignored

repr_text	<i>Text representation</i>
-----------	----------------------------

Description

The only representation defined per default for everthing (via [print\(\)](#))

Usage

```
repr_text(obj, ...)
```

```
## Default S3 method:
repr_text(obj, ...)
```

Arguments

obj	The object to print and then return the output
...	ignored

See Also

[repr-generics](#) for other generics

repr_vega*	<i>Representation as vegalitev2 or vega4 JSON.</i>
------------	--

Description

Representation as [vegalitev2](#) or vega4 JSON.

Usage

```
## S3 method for class 'vegalite'  
repr_vegalite2(obj, ...)
```

Arguments

obj	The vegalite plot to create a representation for
...	ignored

Index

- * **datasets**
 - *2repr, 3
 - repr-options, 6
 - repr_*.htmlwidget, 9
- *2repr, 3
- 'repr.html.deduplicate', 10
- environment, 10
- format2repr (*2repr), 3
- geojson_list, 17
- getOption, 6
- ggplot, 19
- html_dependencies, 7
- html_dependencies (repr_*.htmlwidget), 9
- mime2repr (*2repr), 3
- options, 6
- plot_ly, 19
- Plotly JSON, 19
- print, 3, 19
- repr, 2, 3
- repr-generics, 2, 4, 4, 20
- repr-options, 2, 6, 12, 15, 16
- repr-package, 2
- repr_*.data.table, 7
- repr_*.factor, 8
- repr_*.function, 8
- repr_*.help_files_with_topic, 9
- repr_*.htmlwidget, 9
- repr_*.list, 10
- repr_*.matrix/data.frame, 11
- repr_*.packageIQR, 13
- repr_*.recordedplot, 13
- repr_*.shiny.tag (repr_*.htmlwidget), 9
- repr_*.ts, 15
- repr_*.vector, 16
- repr_geojson (repr-generics), 4
- repr_geojson.*, 17
- repr_geojson.geo_list (repr_geojson.*), 17
- repr_geojson.sf (repr_geojson.*), 17
- repr_geojson.sfc (repr_geojson.*), 17
- repr_geojson.sfg (repr_geojson.*), 17
- repr_geojson.SpatialCollections (repr_geojson.*), 17
- repr_geojson.SpatialGrid (repr_geojson.*), 17
- repr_geojson.SpatialGridDataFrame (repr_geojson.*), 17
- repr_geojson.SpatialLines (repr_geojson.*), 17
- repr_geojson.SpatialLinesDataFrame (repr_geojson.*), 17
- repr_geojson.SpatialPixels (repr_geojson.*), 17
- repr_geojson.SpatialPixelsDataFrame (repr_geojson.*), 17
- repr_geojson.SpatialPoints (repr_geojson.*), 17
- repr_geojson.SpatialPointsDataFrame (repr_geojson.*), 17
- repr_geojson.SpatialPolygons (repr_geojson.*), 17
- repr_geojson.SpatialPolygonsDataFrame (repr_geojson.*), 17
- repr_geojson.SpatialRings (repr_geojson.*), 17
- repr_geojson.SpatialRingsDataFrame (repr_geojson.*), 17
- repr_html (repr-generics), 4
- repr_html.character (repr_*.vector), 16
- repr_html.complex (repr_*.vector), 16
- repr_html.data.frame (repr_*.matrix/data.frame), 11

- repr_html.data.table
 - (repr_*.data.table), 7
- repr_html.Date (repr_*.vector), 16
- repr_html.factor (repr_*.factor), 8
- repr_html.function (repr_*.function), 8
- repr_html.help_files_with_topic
 - (repr_*.help_files_with_topic), 9
- repr_html.htmlwidget
 - (repr_*.htmlwidget), 9
- repr_html.integer (repr_*.vector), 16
- repr_html.list (repr_*.list), 10
- repr_html.logical (repr_*.vector), 16
- repr_html.matrix
 - (repr_*.matrix/data.frame), 11
- repr_html.numeric (repr_*.vector), 16
- repr_html.packageIQR
 - (repr_*.packageIQR), 13
- repr_html.shiny.tag
 - (repr_*.htmlwidget), 9
- repr_html.ts (repr_*.ts), 15
- repr_javascript (repr-generics), 4
- repr_jpg (repr-generics), 4
- repr_jpg.recordedplot
 - (repr_*.recordedplot), 13
- repr_json (repr-generics), 4
- repr_latex (repr-generics), 4
- repr_latex.character (repr_*.vector), 16
- repr_latex.complex (repr_*.vector), 16
- repr_latex.data.frame
 - (repr_*.matrix/data.frame), 11
- repr_latex.data.table
 - (repr_*.data.table), 7
- repr_latex.Date (repr_*.vector), 16
- repr_latex.factor (repr_*.factor), 8
- repr_latex.function (repr_*.function), 8
- repr_latex.help_files_with_topic
 - (repr_*.help_files_with_topic), 9
- repr_latex.integer (repr_*.vector), 16
- repr_latex.list (repr_*.list), 10
- repr_latex.logical (repr_*.vector), 16
- repr_latex.matrix
 - (repr_*.matrix/data.frame), 11
- repr_latex.numeric (repr_*.vector), 16
- repr_latex.ts (repr_*.ts), 15
- repr_markdown (repr-generics), 4
- repr_markdown.character
 - (repr_*.vector), 16
- repr_markdown.complex (repr_*.vector), 16
- repr_markdown.data.frame
 - (repr_*.matrix/data.frame), 11
- repr_markdown.Date (repr_*.vector), 16
- repr_markdown.factor (repr_*.factor), 8
- repr_markdown.function
 - (repr_*.function), 8
- repr_markdown.integer (repr_*.vector), 16
- repr_markdown.list (repr_*.list), 10
- repr_markdown.logical (repr_*.vector), 16
- repr_markdown.matrix
 - (repr_*.matrix/data.frame), 11
- repr_markdown.numeric (repr_*.vector), 16
- repr_markdown.ts (repr_*.ts), 15
- repr_option_defaults (repr-options), 6
- repr_pdf (repr-generics), 4
- repr_pdf.recordedplot
 - (repr_*.recordedplot), 13
- repr_plotly1 (repr-generics), 4
- repr_plotly1.*, 19
- repr_plotly1.ggplot (repr_plotly1.*), 19
- repr_plotly1.plotly (repr_plotly1.*), 19
- repr_png (repr-generics), 4
- repr_png.recordedplot
 - (repr_*.recordedplot), 13
- repr_svg (repr-generics), 4
- repr_svg.recordedplot
 - (repr_*.recordedplot), 13
- repr_text, 2–4, 6, 19
- repr_text.data.frame
 - (repr_*.matrix/data.frame), 11
- repr_text.data.table
 - (repr_*.data.table), 7
- repr_text.help_files_with_topic
 - (repr_*.help_files_with_topic), 9
- repr_text.htmlwidget
 - (repr_*.htmlwidget), 9
- repr_text.matrix
 - (repr_*.matrix/data.frame), 11
- repr_text.packageIQR
 - (repr_*.packageIQR), 13
- repr_text.recordedplot

- (repr_*.recordedplot), 13
- repr_text.shiny.tag
 - (repr_*.htmlwidget), 9
- repr_text.ts (repr_*.ts), 15
- repr_vdom1 (repr-generics), 4
- repr_vega*, 20
- repr_vega4 (repr-generics), 4
- repr_vega5 (repr-generics), 4
- repr_vegalite2 (repr-generics), 4
- repr_vegalite2.vegalite (repr_vega*), 20
- repr_vegalite3 (repr-generics), 4
- repr_vegalite4 (repr-generics), 4
- stop, 3
- ts, 15, 16
- vegalite, 20