

# Package ‘repr’

January 26, 2023

**Title** Serializable Representations

**Version** 1.1.6

**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' 'zzz\_onload.r'

**RoxygenNote** 7.2.3

**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** 2023-01-26 22:30:02 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

<b>Index</b>	<b>21</b>
--------------	-----------

---

repr-package	<i>The repr package</i>
--------------	-------------------------

---

### 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 <a href="#">repr_text</a> )
...	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:::.onload()`).

### 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 <a href="#">geojson_list</a> 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_plotly1.*	<i>Representation as <a href="#">Plotly JSON</a>.</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 <a href="#">plot_ly</a> plot or <a href="#">ggplot</a> 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 <a href="#">print</a> and then return the output
...	ignored

**See Also**

[repr-generics](#) for other generics

---

repr_vega*	<i>Representation as <a href="#">vegalitev2</a> or vega4 JSON.</i>
------------	--

---

**Description**

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

**Usage**

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

**Arguments**

obj	The <a href="#">vegalite</a> 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