

Package ‘ggtibble’

May 14, 2026

Title Create Tibbles and Lists of 'ggplot' Figures for Reporting

Version 1.0.3

Description Create tibbles and lists of 'ggplot' figures that can be modified as easily as regular 'ggplot' figures. Typical use cases are for creating reports or web pages where many figures are needed with different data and similar formatting.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 4.3)

Suggests ggbreak, ggforce, quarto, rmarkdown, spelling, testthat (>= 3.0.0), withr

Config/testthat/edition 3

Imports checkmate, dplyr, ggplot2, glue, knitr, purrr, rlang, tibble, tidyr, vctrs

URL <https://humanpred.github.io/ggtibble/>,
<https://github.com/humanpred/ggtibble>

BugReports <https://github.com/humanpred/ggtibble/issues>

Language en-US

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Bill Denney [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5759-428X>>)

Maintainer Bill Denney <wdenney@humanpredictions.com>

Repository CRAN

Date/Publication 2026-05-14 20:50:10 UTC

Contents

as_gglist	2
extract_glue_expr	3
gglist	3
ggsave	4
ggtibble	6
knit_print.gg	8
knit_print.gglist	9
labs_glue	10
new_gglist	11
new_ggtibble	11
%+%	12

Index	13
--------------	-----------

as_gglist	<i>Convert an object to a gglist</i>
-----------	--------------------------------------

Description

Promotes an input value to a gglist. When the input includes a gg object that uses `ggforce::facet_wrap_paginate()` or `ggforce::facet_grid_paginate()`, the paginated plot is expanded into one gglist element per rendered page so subsequent calls to `print()`, `knit_print()`, or `ggsave()` render every page. Inputs that already use a non-paginated facet (or no facet at all) pass through unchanged.

Usage

```
as_gglist(x, ...)
```

Arguments

x	A gg, gglist, list of gg objects, NULL, or labels object to convert.
...	Not used.

Details

For an input that is already a gglist, the value is returned unchanged so the method is a no-op when nothing needs to be coerced. Call this yourself before rendering when you want paginated facets expanded — the render methods do not call it implicitly because page expansion is not always desired.

Value

A gglist object.

Examples

```
p <- ggplot2::ggplot(mtcars, ggplot2::aes(mpg, wt)) + ggplot2::geom_point()
as_gglist(p)
```

extract_glue_expr *Extract all expressions to be evaluated by glue()*

Description

Extract all expressions to be evaluated by glue()

Usage

```
extract_glue_expr(...)
```

Arguments

... passed to glue()

Value

A character vector of expressions to be evaluated

Examples

```
## Not run:  
extract_glue_expr("foo {character(0)} {bar}")  
  
## End(Not run)
```

gglist *Generate a list of ggplots from a list of data.frames*

Description

Generate a list of ggplots from a list of data.frames

Usage

```
gglist(  
  data = NULL,  
  mapping = ggplot2::aes(),  
  ...,  
  environment = parent.frame()  
)
```

Arguments

data	A list of data.frames (or similar objects)
mapping	Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.
...	Other arguments passed on to methods. Not currently used.
environment	[Deprecated] Used prior to tidy evaluation.

Value

A list of ggplot2 objects

Examples

```
mydata <-  
  list(  
    data.frame(x = 1:3, y = 3:1),  
    data.frame(x = 4:7, y = 7:4)  
  )  
gglist(mydata, ggplot2::aes(x = x, y = y)) +  
  ggplot2::geom_point()
```

ggsave

Save a plot or list of plots

Description

Save a plot or list of plots

Usage

```
ggsave(  
  filename,  
  plot = ggplot2::last_plot(),  
  device = NULL,  
  path = NULL,  
  scale = 1,  
  width = NA,  
  height = NA,  
  units = c("in", "cm", "mm", "px"),  
  dpi = 300,  
  limitsize = TRUE,  
  bg = NULL,  
  create.dir = FALSE,  
  ...  
)
```

```

## S3 method for class 'gglist'
ggsave(
  filename,
  plot,
  device = NULL,
  path = NULL,
  scale = 1,
  width = NA,
  height = NA,
  units = c("in", "cm", "mm", "px"),
  dpi = 300,
  limitsize = TRUE,
  bg = NULL,
  create.dir = FALSE,
  ...
)

## S3 method for class 'ggtibble'
ggsave(
  filename,
  plot,
  device = NULL,
  path = NULL,
  scale = 1,
  width = NA,
  height = NA,
  units = c("in", "cm", "mm", "px"),
  dpi = 300,
  limitsize = TRUE,
  bg = NULL,
  create.dir = FALSE,
  ...
)

```

Arguments

filename	A character string passed to <code>glue::glue_data()</code> to generate file names for each row in plot.
plot	Plot to save, defaults to last plot displayed.
device	Device to use. Can either be a device function (e.g. png), or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only). If NULL (default), the device is guessed based on the filename extension.
path	Path of the directory to save plot to: path and filename are combined to create the fully qualified file name. Defaults to the working directory.
scale	Multiplicative scaling factor.
width, height	Plot size in units expressed by the <code>units</code> argument. If not supplied, uses the size of the current graphics device.

units	One of the following units in which the width and height arguments are expressed: "in", "cm", "mm" or "px".
dpi	Plot resolution. Also accepts a string input: "retina" (320), "print" (300), or "screen" (72). Only applies when converting pixel units, as is typical for raster output types.
limitsize	When TRUE (the default), ggsave() will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.
bg	Background colour. If NULL, uses the plot.background fill value from the plot theme.
create.dir	Whether to create new directories if a non-existing directory is specified in the filename or path (TRUE) or return an error (FALSE, default). If FALSE and run in an interactive session, a prompt will appear asking to create a new directory when necessary.
...	Other arguments passed on to the graphics device function, as specified by device.

Methods (by class)

- ggsave(gglist): Save the figures in a gglist object
- ggsave(ggtibble): Save the figures in a ggtibble object

ggtibble	<i>Make a tibble where one column is the data to plot, one is the gglist, and one is the caption</i>
----------	------------------------------------------------------------------------------------------------------

Description

Make a tibble where one column is the data to plot, one is the gglist, and one is the caption

Usage

```
ggtibble(data, ...)

## S3 method for class 'data.frame'
ggtibble(
  data,
  mapping = ggplot2::aes(),
  ...,
  outercols = group_vars(data),
  labs = list(),
  caption = ""
)
```

Arguments

data	The data.frame to plot
...	Passed to subsequent methods (usually passed to gglist())
mapping	Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.
outercols	The columns to have outside the nesting
labs	Labels to add via labs_glue()
caption	The glue specification for creating the caption

Value

A data.frame with a column named "data_plot" with the data to plot, "figure" with the gglist, and "caption" with the captions

A ggtibble object which is a tibble with columns named "figure" which is a gglist object (a list of ggplots), "data_plot" which is the a list of data.frames making up the source data used for each individual plot, "caption" which is the text to use for the plot caption, and all of the outercols used for nesting.

Methods (by class)

- ggtibble(data.frame): The default method for a data.frame or tibble

Examples

```
d_plot <-
  data.frame(
    A = rep(c("foo", "bar"), each = 4),
    B = 1:8,
    C = 11:18,
    Bunit = "mg",
    Cunit = "km"
  )
all_plots <-
  ggtibble(
    d_plot,
    ggplot2::aes(x = B, y = C),
    outercols = c("A", "Bunit", "Cunit"),
    caption = "All the {A}",
    labs = list(x = "B ({Bunit})", y = "C ({Cunit})")
  ) +
  ggplot2::geom_point() +
  ggplot2::geom_line()
knit_print(all_plots)
```

knit_print.gg

Print a ggplot (usually within knit_print.gglist)

Description

Print a ggplot (usually within knit_print.gglist)

Usage

```
## S3 method for class 'gg'
knit_print(
  x,
  ...,
  fig_prefix,
  fig_suffix,
  filename = NULL,
  width = 6,
  height = 4,
  units = "in"
)
```

Arguments

x	The gg object (i.e. a ggplot)
...	Ignored
fig_prefix	Text to cat() before the figure is printed
fig_suffix	Any text to add after the figure. Defaults to NULL, which means "auto-select": "\n\n\\FloatBarrier\n\n" for LaTeX output when length(x) > float_barrier_after, otherwise "\n\n".
filename	A filename saving the plot
width, height	Plot size in units expressed by the units argument. If not supplied, uses the size of the current graphics device.
units	One of the following units in which the width and height arguments are expressed: "in", "cm", "mm" or "px".

Value

The gg object, invisibly

See Also

Other knitters: [knit_print.gglist\(\)](#)

knit_print.gglist *Print a list of plots made by gglist*

Description

The filename argument may be given with an `sprintf()` format including `"%d"` to allow automatic numbering of the output filenames. Specifically, the pattern of `"%d"` with an optional non-negative integer between the `"%"` and `"d"` is searched for and if found, then the filename will be generated using that `sprintf()` format. Note that also means that other requirements for `sprintf()` must be met; for example, if you want a percent sign (`"%"`) in the filename, it must be doubled so that `sprintf` returns what is desired.

Usage

```
## S3 method for class 'gglist'
knit_print(
  x,
  ...,
  filename = NULL,
  fig_suffix = NULL,
  float_barrier_after = 10
)

## S3 method for class 'ggtibble'
knit_print(x, ...)
```

Arguments

<code>x</code>	The gglist object
<code>...</code>	extra arguments to <code>knit_print()</code>
<code>filename</code>	A filename with an optional <code>"%d"</code> <code>sprintf</code> pattern for saving the plots
<code>fig_suffix</code>	Any text to add after the figure. Defaults to <code>NULL</code> , which means "auto-select": " <code>\n\n\\FloatBarrier\n\n</code> " for LaTeX output when <code>length(x) > float_barrier_after</code> , otherwise " <code>\n\n</code> ".
<code>float_barrier_after</code>	Numeric threshold for emitting <code>\FloatBarrier</code> between figures in LaTeX output. When <code>length(x) > float_barrier_after</code> and <code>knitr::is_latex_output()</code> is <code>TRUE</code> and the user did not supply <code>fig_suffix</code> , <code>fig_suffix</code> defaults to " <code>\n\n\\FloatBarrier\n\n</code> ". Has no effect on non-LaTeX output. Set to <code>Inf</code> to disable. Defaults to <code>10</code> .

Details

When `length(x)` exceeds `float_barrier_after` and the output format is LaTeX (as detected by `knitr::is_latex_output()`), `fig_suffix` defaults to "`\n\n\\FloatBarrier\n\n`" instead of the usual "`\n\n`". This avoids the LaTeX "Output loop—100 consecutive dead cycles" error that occurs when the float queue (default capacity ~18) overflows. `\FloatBarrier` is provided

by the `placeins` LaTeX package, which is *not* loaded by default in `rmarkdown::pdf_document`; add `\usepackage{placeins}` to the document preamble (e.g. via `header-includes` in the YAML) when relying on the auto-suffix. Pass `fig_suffix` explicitly to override, or set `float_barrier_after = Inf` to disable the auto-suffix entirely.

Value

The list, invisibly

Functions

- `knit_print(ggtibble)`: Print the plots in a `ggtibble` object

See Also

Other knitters: [knit_print.gg\(\)](#)

Examples

```
# Ensure that each figure is within its own float area
mydata <-
  list(
    data.frame(x = 1:3, y = 3:1),
    data.frame(x = 4:7, y = 7:4)
  )
p <- gglist(mydata, ggplot2::aes(x = x, y = y)) +
  ggplot2::geom_point()
knit_print(p, fig_suffix = "\n\n\\FloatBarrier\n\n")
```

labs_glue

Generate ggplot2 labels based on data in a ggtibble

Description

Generate `ggplot2` labels based on data in a `ggtibble`

Usage

```
labs_glue(p, ...)
```

Arguments

<code>p</code>	The <code>ggtibble</code> object
<code>...</code>	Named arguments to be used as <code>ggplot2::labs()</code> labels where the value is a glue specification

Value

`p` with the labels modified

new_gglist	<i>Create a new gglist object</i>
------------	-----------------------------------

Description

Create a new gglist object

Usage

```
new_gglist(x = list())
```

Arguments

x A list of ggplot2 objects to convert into a gglist

Value

The list verified to be a gglist and with the gglist class

See Also

Other New ggtibble objects: [new_ggtibble\(\)](#)

Examples

```
new_gglist(list(NULL, ggplot2::ggplot(data = data.frame())))
```

new_ggtibble	<i>Create a new ggtibble object</i>
--------------	-------------------------------------

Description

Create a new ggtibble object

Usage

```
new_ggtibble(x)
```

Arguments

x A data.frame with a column named "figure" and "caption", and where the "figure" column is a ggtibble.

Value

The object with a ggtibble class

See Also

Other New ggtibble objects: [new_gglist\(\)](#)

Examples

```
new_ggtibble(tibble::tibble(figure = list(ggplot2::ggplot()), caption = ""))
```

%+%

Use the %+% operator from ggplot2 for ggtibble and gglist objects

Description

Use the %+% operator from ggplot2 for ggtibble and gglist objects

Usage

```
e1 %+% e2
```

Arguments

e1	Either a ggtibble or gglist object or an object that can use the default ggplot2::%+% function
e2	A plot component (see ?ggplot2::%+%)

Index

* New ggtribble objects

new_gglist, 11

new_ggtribble, 11

* knitters

knit_print.gg, 8

knit_print.gglist, 9

%+%, 12

as_gglist, 2

extract_glue_expr, 3

gglist, 3

ggsave, 4

ggtribble, 6

knit_print.gg, 8

knit_print.gg(), 10

knit_print.gglist, 9

knit_print.gglist(), 8

knit_print.ggtribble

(knit_print.gglist), 9

labs_glue, 10

new_gglist, 11

new_gglist(), 12

new_ggtribble, 11

new_ggtribble(), 11

png, 5