

# Package: ggghost (via r-universe)

August 27, 2024

**Title** Capture the Spirit of Your 'ggplot2' Calls

**Version** 0.2.1

**Maintainer** Jonathan Carroll <rpkg@jcarroll.com.au>

**Description** Creates a reproducible 'ggplot2' object by storing the data and calls.

**Depends** R (>= 3.2.0), ggplot2, animation

**License** GPL(>=3)

**Encoding** UTF-8

**LazyData** true

**URL** <https://github.com/jonocarroll/ggghost>

**BugReports** <https://github.com/jonocarroll/ggghost/issues>

**RoxygenNote** 5.0.1

**Suggests** testthat

**Repository** <https://jonocarroll.r-universe.dev>

**RemoteUrl** <https://github.com/jonocarroll/ggghost>

**RemoteRef** HEAD

**RemoteSha** 888ca70248bc501d3af4ca318e61a46d9c159286

## Contents

+.gg	2
-.gg	2
ggghost	3
is.ggghost	4
print.ggghost	4
reanimate	5
recover_data	6
subset.ggghost	6
summary.ggghost	7
supp_data	8
supp_data<-	9
%g<%	9



## Arguments

e1	An object of class ggghost
e2	A component to remove from e1

## Details

For example, subtracting `geom_line()` will remove all calls matching `geom_line` regardless of their arguments.

'`labs()`' has been identified as a special case, as it requires an argument in order to be recognised as a valid function. Thus, trying to remove it with an empty argument will fail. That said, the argument doesn't need to match, so it can be populated with a dummy string or anything that evaluates in scope. See examples.

## Value

A ggghost structure with calls matching e2 removed, otherwise the same as e1

## Examples

```
## create a ggghost object
tmpdata <- data.frame(x = 1:100, y = rnorm(100))

z %g<% ggplot(tmpdata, aes(x,y))
z <- z + geom_point(col = "steelblue")
z <- z + theme_bw()
z <- z + labs(title = "My cool ggplot")
z <- z + labs(x = "x axis", y = "y axis")
z <- z + geom_smooth()

## remove the geom_smooth
z - geom_smooth()

## remove the labels
## NOTE: argument must be present and able to be
## evaluated in scope
z - labs(TRUE) # works
z - labs(title) # works because of title(), but removes all labs()
```

---

ggghost

*ggghost: Capture the spirit of your ggplot calls*

---

## Description

Creates a reproducible container for ggplot, storing the data and calls required to produce a plot.

**Details**

'ggplot' stores the information needed to build the graph as a 'grob', but that's what the *computer* needs to know about in order to build the graph. As humans, we're more interested in what commands were issued in order to build the graph. For good reproducibility, the calls need to be applied to the relevant data. While this is somewhat available by deconstructing the 'grob', it's not the simplest approach.

Here is one option that solves that problem.

'ggghost' stores the data used in a 'ggplot()' call, and collects 'ggplot' commands (usually separated by '+') as they are applied, in effect lazily collecting the calls. Once the object is requested, the 'print' method combines the individual calls back into the total plotting command and executes it. This is where the call would usually be discarded. Instead, a "ghost" of the commands lingers in the object for further investigation, subsetting, adding to, or subtracting from.

---

is.ggghost	<i>Reports whether x is a ggghost object</i>
------------	--

---

**Description**

Reports whether x is a ggghost object

**Usage**

```
is.ggghost(x)
```

**Arguments**

x	An object to test
---	-------------------

**Value**

logical; TRUE if x inherits class ggghost

---

print.ggghost	<i>Collect ggghost calls and produce the ggplot output</i>
---------------	--

---

**Description**

Collect ggghost calls and produce the ggplot output

**Usage**

```
## S3 method for class 'ggghost'
print(x, ...)
```

**Arguments**

x                    A ggghost object to be made into a ggplot grob  
 ...                  Not used, provided for `print.default` generic consistency.

**Value**

The ggplot plot data (invisibly). Used for the side-effect of producing a ggplot plot.

---

reanimate	<i>Bring a ggplot to life (re-animate)</i>
-----------	--

---

**Description**

Creates an animation showing the stepwise process of building up a ggplot. Successively adds calls from a ggghost object and then combines these into an animated GIF.

**Usage**

```
reanimate(object, gifname = "ggghost.gif", interval = 1, ani.width = 600,
           ani.height = 600)
```

```
lazarus(object, gifname = "ggghost.gif", interval = 1, ani.width = 600,
         ani.height = 600)
```

**Arguments**

object              A ggghost object to animate  
 gifname            Output filename to save the .gif to (not including any path, will be saved to current directory)  
 interval           A positive number to set the time interval of the animation (unit in seconds); see `animation::ani.options`  
 ani.width          width of image frames (unit in px); see `animation::ani.options`  
 ani.height        height of image frames (unit in px); see `animation::ani.options`

**Value**

TRUE if it gets that far

**Examples**

```
## Not run:
## create an animation showing the process of building up a plot
reanimate(z, "mycoolplot.gif")

## End(Not run)
```

---

recover_data	<i>Recover data Stored in a ggghost object</i>
--------------	--

---

### Description

The data used to generate a plot is an essential requirement for a reproducible graphic. This is somewhat available from a ggplot grob (in raw form) but it is not easily accessible, and isn't named the same way as the original call.

### Usage

```
recover_data(x, supp = TRUE)
```

### Arguments

x	A ggghost object from which to extract the data.
supp	(logical) Should the supplementary data be extracted also?

### Details

This function retrieves the data from the ggghost object as it was when it was originally called.

If supplementary data has also been attached using [supp\\_data](#) then this will also be recovered (if requested).

When used interactively, a warning will be produced if the data to be extracted exists in the workspace but not identical to the captured version.

### Value

A data.frame of the original data, named as it was when used in `ggplot(data)`

---

subset.ggghost	<i>Extract a subset of a ggghost object</i>
----------------	---

---

### Description

Alternative to subtracting calls using `'-.gg'`, this method allows one to select the desired components of the available calls and have those evaluated.

### Usage

```
## S3 method for class 'ggghost'
subset(x, ...)
```

**Arguments**

`x` A ggghost object to subset

`...` A logical expression indicating which elements to select. Typically a vector of list numbers, but potentially a vector of logicals or logical expressions.

**Value**

Another ggghost object containing only the calls selected.

**Examples**

```
## create a ggghost object
tmpdata <- data.frame(x = 1:100, y = rnorm(100))

z %g<% ggplot(tmpdata, aes(x,y))
z <- z + geom_point(col = "steelblue")
z <- z + theme_bw()
z <- z + labs(title = "My cool ggplot")
z <- z + labs(x = "x axis", y = "y axis")
z <- z + geom_smooth()

## remove the labels and theme
subset(z, c(1,2,6))
## or
subset(z, c(TRUE,TRUE,FALSE,FALSE,FALSE,TRUE))
```

---

summary.ggghost      *List the calls contained in a ggghost object*

---

**Description**

Summarises a ggghost object by presenting the contained calls in the order they were added. Optionally concatenates these into a single ggplot call.

**Usage**

```
## S3 method for class 'ggghost'
summary(object, ...)
```

**Arguments**

`object` A ggghost object to present

`...` Mainly provided for `summary.default` generic consistency. When `combine` is passed as an argument (arbitrary value) the list of calls is concatenated into a single string as one might write the ggplot call.

**Details**

The data is also included in ggghost objects. If this is also desired in the output, use `str`. See example.

**Value**

Either a list of ggplot calls or a string of such concatenated with " + "

**Examples**

```
## present the ggghost object as a list
tmpdata <- data.frame(x = 1:100, y = rnorm(100))

z %g<% ggplot(tmpdata, aes(x,y))
z <- z + geom_point(col = "steelblue")
summary(z)

## present the ggghost object as a string
summary(z, combine = TRUE) # Note, value of 'combine' is arbitrary

## to inspect the data structure also captured, use str()
str(z)
```

---

`supp_data`*Inspect the supplementary data attached to a ggghost object*

---

**Description**

Inspect the supplementary data attached to a ggghost object

**Usage**

```
supp_data(x)
```

**Arguments**

x                    A ggghost object

**Value**

A list with two elements: the name of the supplementary data, and the supplementary data itself

---

supp\_data<-                      *Attach supplementary data to a ggghost object*

---

**Description**

Attach supplementary data to a ggghost object

**Usage**

```
supp_data(x) <- value
```

**Arguments**

x	A ggghost object to which the supplementary data should be attached
value	Supplementary data to attach to the ggghost object, probably used as an additional data input to a scale_* or geom_* call

**Value**

The original object with suppdata attribute

---

*%g<%                                      Begin constructing a ggghost cache*

---

**Description**

The data and initial ggplot() call are stored as a list (call) with attribute (data).

**Usage**

```
lhs %g<% rhs
```

**Arguments**

lhs	LHS of call
rhs	RHS of call

**Details**

The data must be passed into the ggplot call directly. Passing this in via a magrittr pipe remains as a future improvement. The newly created ggghost object is a list of length 1 containing the ggplot call, with attribute data; another list, containing the data\_name and data itself.

**Value**

Assigns the ggghost structure to the lhs symbol.

**Examples**

```
## create a ggghost object
tmpdata <- data.frame(x = 1:100, y = rnorm(100))

z %g<% ggplot(tmpdata, aes(x,y))
```

# Index

`+.gg`, 2

`-.gg`, 2

`%g<%`, 9

`ggghost`, 3

`ggghost-package (ggghost)`, 3

`is.ggghost`, 4

`lazarus (reanimate)`, 5

`print.ggghost`, 4

`reanimate`, 5

`recover_data`, 6

`subset.ggghost`, 6

`summary.ggghost`, 7

`supp_data`, 6, 8

`supp_data<-`, 9