

# Package ‘ggedit’

April 6, 2017

**Type** Package

**Title** Interactive 'ggplot2' Layer and Theme Aesthetic Editor

**Version** 0.2.1

**Date** 2017-03-31

**Author** Jonathan Sidi [aut, cre]

**Maintainer** Jonathan Sidi <yonis@metrumrg.com>

**Description** Interactively edit 'ggplot2' layer and theme aesthetics definitions.

**Depends** R (>= 2.3.0), shiny, shinyBS, ggplot2 (>= 2.2.0), dplyr (>= 0.5)

**Imports** plyr, reshape2, rstudioapi, utils, colourpicker (>= 0.2), scales, graphics, shinyAce, grid, miniUI

**License** GPL-2 | GPL-3

**URL** <https://github.com/metrumresearchgroup/ggedit>

**BugReports** <https://github.com/metrumresearchgroup/ggedit/issues>

**LazyData** true

**NeedsCompilation** no

**RoxygenNote** 6.0.1

**Repository** CRAN

**Date/Publication** 2017-04-06 21:20:57 UTC

## R topics documented:

aesColourNS . . . . .	2
as.ggedit . . . . .	2
cloneLayer . . . . .	3
compare . . . . .	4
fontDefaults . . . . .	5
ggedit . . . . .	5
gggsave . . . . .	7
is.ggedit . . . . .	8

layersList . . . . .	9
pList . . . . .	9
plot.ggedit . . . . .	10
print.ggedit . . . . .	11
remove_geom . . . . .	11
rgg . . . . .	12
slideDefaults . . . . .	12
ThemeDefaultClass . . . . .	13
themeTips . . . . .	13

<b>Index</b>	<b>14</b>
--------------	-----------

---

aesColourNS	<i>aesColorNS</i>
-------------	-------------------

---

**Description**

ColorInput UI production for discrete variables for module.

**Usage**

aesColourNS(type, session)

**Arguments**

type	character of label and inputId of element
session	shiny session object

**Value**

UI object

---

as.ggedit	<i>Try to coerce a ggplot object into a ggedit object</i>
-----------	---

---

**Description**

Applied to ggplot objects to use the plotting function of ggedit.

**Usage**

as.ggedit(p)

**Arguments**

p	an object
---	-----------

**Value**

an object of class ggedit

**See Also**

[plot.ggedit](#), [ggplot](#)

**Examples**

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p1=p+geom_point(aes(colour=Species))+geom_line()
p2=p+geom_point()+geom_smooth(method='loess')
p3=list(p1,p2)
p4=as.ggedit(p3)
plot(p4)
```

---

cloneLayer

---

*Creates an independent copy of a ggplot layer object*


---

**Description**

Creates copies of ggplot layers from within ggplot objects that are independent of the parent object.

**Usage**

```
cloneLayer(l, verbose = FALSE, showDefaults = TRUE)
```

**Arguments**

l	ggplot2 object layer
verbose	toggle to control if the output is ggproto object (verbose==FALSE,default) or string of layer call (verbose==TRUE)
showDefaults	toggle to control if the verbose output shows all the input arguments passed to the proto object (if verbose==FALSE then ignored)

**Details**

ggplot objects are comprised of layer objects. Once compiled they are part of the plot object environment and if they are changed internally regardless of where they are in the (ie different environment) it will change the original plot. This function allows to create replicates of the plot layers and edit them independent of the original plot. When setting verbose to TRUE function returns the ggplot2 call as a string to paste in regular ggplot script to generate the layer.

**Value**

ggproto or string object (conditional on verbose)

**Examples**

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()
p$layers[[1]]
newLayer=cloneLayer(l=p$layers[[1]])
all.equal(p$layers[[1]],newLayer)
(v=cloneLayer(l=p$layers[[1]],verbose=TRUE))
eval(parse(text=v))
all.equal(p$layers[[1]],eval(parse(text=v)))
```

---

 compare

*compare*


---

**Description**

Compare differences theme object e1 (new theme) to theme object e2 (old theme)

**Usage**

```
compare(e1, e2, verbose = TRUE)
```

**Arguments**

e1	theme object
e2	theme object
verbose	logical to control if the output is a character of script or a theme object (default TRUE)

**Value**

theme object or character depending on verbose

**Examples**

```
compare(theme_bw(), theme_get())
compare(theme_bw(), theme_get(), verbose=FALSE)
```

---

fontDefaults	<i>fontDefaults</i>
--------------	---------------------

---

**Description**

Character of fonts used as options in ggedit.

**Usage**

fontDefaults

**Format**

An object of class "character"

**Examples**

```
data(fontDefaults)
fontDefaults
```

---

ggedit	<i>Interactive shiny gadget for editing ggplot layers and themes.</i>
--------	---

---

**Description**

Shiny gadget that takes an input ggplots and populates a user interface with objects that let the user update aesthetics of layers and theme elements.

**Usage**

ggedit(p.in, ...)

**Arguments**

p.in	ggplot plot object or list of objects
...	options that are passed to ggedit

## Details

The user can start the gadget using the console `ggedit(plotobj)` or through the Addins menu in Rstudio.

If you are using the the Addin option highlight on the editor window the ggplot object and then click the addin.

### Options to pass to ggedit

**viewer** shiny viewer options. It can be either `paneViewer` (default with `minHeight=1000`), `dialogViewer`, `browserViewer`

**verbose** logical to control if the output includes script for layers and themes calls for parsing to create objects (default, `verbose=TRUE`)

**showDefaults** toggle to control if the verbose output shows all the input arguments passed to the proto object (if `verbose==FALSE` then ignored)

**width,height** dimensions of the `renderPlot` where the active plot is displayed

Once the gadget is running the list of plots are shown in a grid and a number of objects will appear above them.

### Action buttons

Cancel:

Returns a NULL object

Done:

Returns the list decribed below.

### Dropdown list

Navigates through the plots in the input list. If the input list is a named list the names will be in the dropdown. The plot chosen is termed as the "active plot"

### Radio buttons

The options to choose in the radio buttons are the layer names in the active plot.

### Links

Update Plot Layer:

A pop up window will appear and be populated with aesthetic elements found in the layer chosen from the radio buttons. The layer is cloned using `cloneLayer` creating a layer independent of the original plot. If the aesthetic is a factor the values will be shown in dropdown lists. If it is numeric it will be shown in a slider. If it is a factor colour/fill aesthetic the `colourPicker` package will allow to choose from the full pallete of colours. If the continuous colour/fill aesthetic a dropdown list will be shown with different palletes

Update Plot Theme:

A popup modal will appear populated with the theme elements found in the active plot. Each element will appear as having a value or empty depending if it was defined or not. The user can change or fill in any element **with valid values** and any textboxes left empty will use ggplot defaults.

Update Grid Theme:

Copies the theme of the active plot to the other plots in the list

Update Global Theme:

Copies the theme of the active plot to the session theme and all plots created outside of the gadget will have this theme.

View Layer Code:

Opens an ace editor to compare the active layer initial script call and the updated script call.

The ggplot objects returned (layers and themes) can be used on any ggplot object.

## Value

List of elements

**updatedPlots** list containing updated ggplot objects

**updatedLayers** For each plot a list of updated layers (ggproto) objects

**UpdatedLayersElements** For each plot a list elements and their values in each layer

**UpdatedLayerCalls** For each plot a list of scripts that can be run directly from the console to create a layer

**updatedScales** For each plot a list of updated scale objects

**UpdatedScalesCalls** For each plot a list of scripts that can be run directly from the console to create a scale object

**updatedThemes** For each plot a list of updated theme objects

**UpdatedThemeCalls** For each plot a list of scripts that can be run directly from the console to create a theme

## See Also

[cloneLayer](#), [rgg](#), [ggplot](#), [colourPicker](#)

## Examples

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()

pnew=ggedit(p)
pnew
```

---

gggsave

gggsave

---

## Description

Wrapper of ggsave that saves ggplot or list of ggplot objects to image or pdf.

## Usage

```
gggsave(filename = "Rplot.pdf", plot = last_plot(), ...)
```

**Arguments**

filename	a character string giving the name of the file. If it is of the form "lcmd", the output is piped to the command given by cmd. If it is NULL, then no external file is created (effectively, no drawing occurs), but the device may still be queried (e.g., for size of text). For use with onefile = FALSE give a C integer format such as "Rplot%03d.pdf" (the default in that case)
plot	ggplot or list of ggplots to save, defaults to last plot displayed
...	other arguments passed on to graphics device

**Details**

default output is to create one pdf regardless of size of list of plots inputted

**Value**

nothing

**Examples**

```
## Not run: ggglsave(pList)
```

---

is.ggedit

*Is the object of class ggedit*


---

**Description**

Is the object of class ggedit. Very basic for many functions in the package.

**Usage**

```
is.ggedit(p)
```

**Arguments**

p	an object
---	-----------

**Value**

logical - is the object of class ggedit



---

`layersList`*layersList*

---

**Description**

Runs the `ggplot_build` function on the input and converts the output data objects into a nested list with the unique values of each of the aesthetic columns.

**Usage**

```
layersList(obj)
```

**Arguments**

`obj` `ggplot2` plot object or list of plot objects

**Value**

list of aesthetics and their values for each layer in a plot

**Examples**

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()
p
p.list=layersList(p)
p.list
```

---

`pList`*List of plots for ggedit examples*

---

**Description**

List of `ggplot` objects for examples in `ggedit`.

**Usage**

```
pList
```

**Format**

An object of class "list"

**Details**

list includes a geom\_point, geom\_point+facet\_wrap, geom\_boxplot+facet\_wrap, geom\_point+geom\_line, geom\_point+geom\_smooth, geom\_point+geom\_line+facet\_wrap, geom\_point+geom\_line+facet\_grid

**Examples**

```
data(pList)
pList
```

---

plot.ggedit

*Plots ggedit plot output objects*


---

**Description**

Plots ggplot2 plots edited by ggedit with layout functionality.

**Usage**

```
## S3 method for class 'ggedit'
plot(x, ...)
```

**Arguments**

x	ggedit output of list of plots from the ggedit output
...	list which defines the location of each plot within the viewport layout. (see details)

**Details**

If ... is NULL then a default layout is used filling in a grid.layout row.wise. Defining plot.layout as a nested list giving the rows and the column indices of each plot will produce a specific layout.

**Examples**

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()
p
x=ggedit(p)
plot(x)
plot(x$UpdatedPlots)
p1=pList[c(1:2)]
x1=ggedit(p1)
plot(x1)
plot(x1,plot.layout = list(list(rows=2,cols=2),list(rows=1,cols=1:2)))
```

---

print.ggedit	<i>Print verbose outputs of objects of class ggedit</i>
--------------	---

---

**Description**

function to tidy the ggedit output to single script calls for each plot

**Usage**

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

**Arguments**

x	ggedit object
...	nothing

---

remove_geom	<i>Remove a layer from a compiled ggplot2 object.</i>
-------------	---

---

**Description**

Removes specified layers from a ggplot object.

**Usage**

```
remove_geom(p, geom, idx)
```

**Arguments**

p	ggplot2 plot object
geom	character string of the name of the layer to remove
idx	numeric of which index of geom to remove

**Examples**

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()
p
pnew=p%>%remove_geom('point',1)
pnew
```

---

 rgg

*Remove and replace ggplot2 layers.*


---

### Description

Removes specified layers from a ggplot object and gives the option to replace them with a new layer. This layer can be either a geom object created from regular ggplot functions or an output from the ggedit gadget. In the latter case the layers are found in the updatedLayers object in the ggedit output.

### Usage

```
rgg(p, oldGeom, oldGeomIdx = 1, newLayer = NULL)
```

### Arguments

p	ggplot2 plot object
oldGeom	character string of the name of the layer to remove
oldGeomIdx	numeric of which index of OldGeom to remove (default is 1)
newLayer	ggplot layer or list of layers

### Examples

```
p=ggplot(iris,aes(x =Sepal.Length,y=Sepal.Width))
p=p+geom_point(aes(colour=Species))+geom_line()
p
p%>%rgg('point',1)

x=ggedit(p)
pnew=p%>%rgg('point',1,x$updatedLayers[[1]])
pnew
```

---

 slideDefaults

*slideDefaults*


---

### Description

List of limits for layer aesthetics.

### Usage

```
slideDefaults
```

**Format**

An object of class "list"

**Examples**

```
data(slideDefaults)
slideDefaults
```

---

ThemeDefaultClass	<i>ThemeDefaultClass</i>
-------------------	--------------------------

---

**Description**

Data.frame of default classes for theme elements.

**Usage**

ThemeDefaultClass

**Format**

An object of class "data.frame"

**Examples**

```
data(ThemeDefaultClass)
ThemeDefaultClass
```

---

themeTips	<i>themeTips</i>
-----------	------------------

---

**Description**

List of HTML tips for the theme BS modal.

**Usage**

data(themeTips)

**Format**

An object of class "list"

**Examples**

```
data(themeTips)
themeTips
```

# Index

## \*Topic **datasets**

- fontDefaults, [5](#)
- pList, [9](#)
- slideDefaults, [12](#)
- ThemeDefaultClass, [13](#)
- themeTips, [13](#)

- aesColourNS, [2](#)
- as.ggedit, [2](#)

- cloneLayer, [3](#), [6](#), [7](#)
- colourPicker, [6](#), [7](#)
- compare, [4](#)

- fontDefaults, [5](#)

- ggedit, [5](#)
- gggsave, [7](#)
- ggplot, [3](#), [7](#)

- is.ggedit, [8](#)

- layersList, [9](#)

- pList, [9](#)
- plot.ggedit, [3](#), [10](#)
- print.ggedit, [11](#)

- remove\_geom, [11](#)
- rgg, [7](#), [12](#)

- slideDefaults, [12](#)

- ThemeDefaultClass, [13](#)
- themeTips, [13](#)