

# Package: ggallin (via r-universe)

September 18, 2024

**Type** Package

**Maintainer** Steven E. Pav <shabbychef@gmail.com>

**Version** 0.1.1.1001

**Date** 2019-11-17

**License** LGPL-3

**Title** Grab Bag of 'ggplot2' Functions

**BugReports** <https://github.com/shabbychef/ggallin/issues>

**Description** Extra geoms and scales for 'ggplot2', including `geom_cloud()`, a Normal density cloud replacement for errorbars; transforms `ssqrt_trans` and `pseudolog10_trans`, which are loglike but appropriate for negative data; `interp_trans()` and `warp_trans()` which provide scale transforms based on interpolation; and an infix compose operator for scale transforms.

**Depends** ggplot2 (>= 2.2.1)

**Suggests** knitr, testthat

**Imports** scales, grid

**RoxygenNote** 5.0.1

**URL** <https://github.com/shabbychef/ggallin>

**Collate** 'geom\_cloud.R' 'ggallin.R' 'stat\_subqq.R' 'transforms.R'

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

**RemoteUrl** <https://github.com/shabbychef/ggallin>

**RemoteRef** HEAD

**RemoteSha** b5887be824b5aab7d60c03de1d40d5004af343c2

## Contents

ggallin-package . . . . .	2
geom_cloud . . . . .	2

ggallin-NEWS . . . . .	5
interp_trans . . . . .	5
ssqrt_trans . . . . .	7
%of% . . . . .	8
<b>Index</b>	<b>9</b>

---

ggallin-package	<i>Grab Bag of GGplot2 Functions.</i>
-----------------	---------------------------------------

---

### Description

This package consists of some helper functions for working with ggplot2: geoms, transforms, *etc.*, with no real unifying theme among them.

### Legal Mumbo Jumbo

ggallin is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

### Author(s)

Steven E. Pav <shabbychef@gmail.com>

---

geom_cloud	<i>geom_cloud</i>
------------	-------------------

---

### Description

Draw a normal uncertainty cloud as a ribbon

Draws overlapping ribbons of the same identity to create a cloud of (Gaussian) uncertainty. Similar to an errorbar geom in use, but visually less distracting (sometimes).

### Usage

```
geom_cloud(mapping = NULL, data = NULL, ..., na.rm = TRUE, steps = 7,
           se_mult = 1, max_alpha = 1, inherit.aes = TRUE)
```

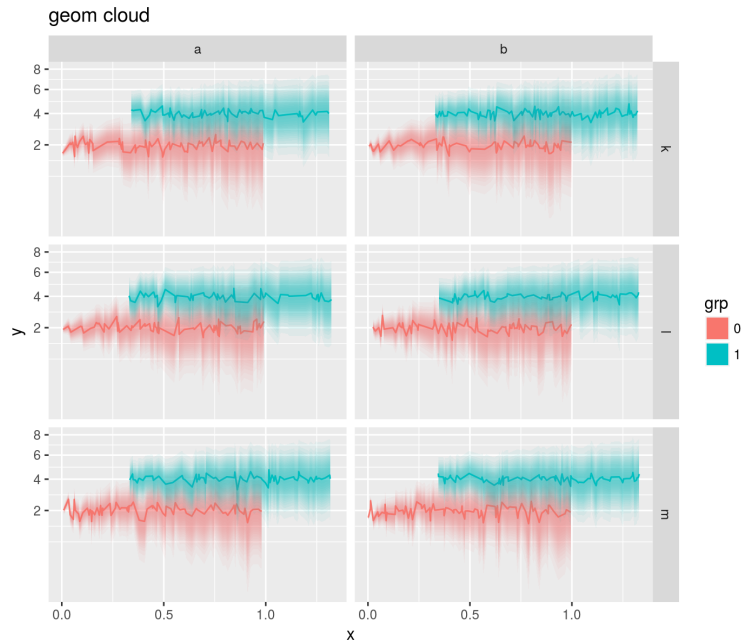
## Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> or <code>aes_()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data.</p>
...	Other arguments passed on to <code>layer()</code> . These are often aesthetics, used to set an aesthetic to a fixed value, like <code>color = "red"</code> or <code>size = 3</code> . They may also be parameters to the paired geom/stat.
na.rm	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
steps	The integer number of steps, or equivalently, the number of overlapping ribbons. A larger number makes a smoother cloud at the possible expense of rendering time. Values larger than around 20 are typically not necessary.
se_mult	The 'multiplier' of standard errors of the given <code>ymin</code> and <code>ymax</code> . If these are at one standard error, then let <code>se_mult</code> take the default value of 1.
max_alpha	The maximum alpha at the maximum density. The cloud will have alpha no greater than this value.
inherit.aes	If <code>FALSE</code> , overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .

## Details

Assumes that `ymin` and `ymax` are plotted at a fixed number of standard errors away from `y`, then computes a Gaussian density with that standard deviation, plotting a cloud (based on `geom_ribbon`) with alpha proportional to the density. This appears as a vertical 'cloud' of uncertainty. In use, this geom should be comparable to `geom_errorbar`.

A sample output from `geom_cloud`:



## Aesthetics

geom\_cloud understands the following aesthetics (required aesthetics are in bold):

- x
- y
- **ymin**
- **ymax**
- fill

Only one of **ymin** and **ymax** is strictly required.

## Note

This is a thin wrapper on the geom\_ribbon geom.

## Author(s)

Steven E. Pav <shabbychef@gmail.com>

## See Also

[geom\\_ribbon](#): The underlying geom

**Examples**

```

set.seed(2134)
nobs <- 200
mydat <- data.frame(grp=sample(c(0,1),nobs,replace=TRUE),
  colfac=sample(letters[1:2],nobs,replace=TRUE),
  rowfac=sample(letters[10 + (1:3)],nobs,replace=TRUE))
mydat$x <- seq(0,1,length.out=nobs) + 0.33 * mydat$grp
mydat$y <- 0.25 * rnorm(nobs) + 2 * mydat$grp
mydat$grp <- factor(mydat$grp)
mydat$se <- sqrt(mydat$x)

ggplot(mydat,aes(x=x,y=y,ymin=y-se,ymax=y+se,color=grp)) +
  facet_grid(rowfac ~ colfac) +
  geom_line() +
  geom_errorbar() +
  labs(title='uncertainty by errorbar')

ggplot(mydat,aes(x=x,y=y,ymin=y-se,ymax=y+se,fill=grp)) +
  facet_grid(rowfac ~ colfac) +
  geom_line() +
  geom_cloud(steps=15,max_alpha=0.85) +
  labs(title='uncertainty by cloudr')

```

---

ggallin-NEWS

*News for package 'ggallin':*

---

**Description**

News for package 'ggallin'

**Version 0.1.1 (2017-10-01)**

- submit to CRAN

---

interp\_trans

*Interpolation based scale transforms.*

---

**Description**

Interpolation based scale transformations. The user supplies  $x$  and  $y$  (which should be monotonic increasing or decreasing in  $x$ ) to create a scale transformation based on linear interpolation.

A 'warp' transformation is also supported wherein the user supplies  $x$  and  $w$  where, after sorting on  $x$ , the cumulative sum of  $w$  are used as the  $y$  in an interpolation transformation. Here  $w$  are the rate of increase, or 'weights'.

**Usage**

```
interp_trans(x=NULL,y=NULL,data=NULL,na.rm=TRUE,breaks=NULL,format=NULL)
```

```
warp_trans(x=NULL,w=NULL,data=NULL,na.rm=TRUE,breaks=NULL,format=NULL)
```

**Arguments**

<code>x</code>	the $x$ coordinates for linear interpolation.
<code>y</code>	the $y$ coordinates for linear interpolation.
<code>data</code>	A <code>data.frame</code> with columns of $x$ and $y$ for <code>interp_trans</code> or $x$ and $w$ for <code>warp_trans</code> . If <code>data</code> is given, it takes precedence over the given $x$ , $y$ , $w$ .
<code>na.rm</code>	If <code>TRUE</code> , then missing $x$ or $y$ will be removed.
<code>breaks</code>	default breaks function for this transformation. The breaks function is applied to the raw data.
<code>format</code>	default format for this transformation. The format is applied to breaks generated to the raw data.
<code>w</code>	the $w$ coordinates for the ‘warp’ interpolation. The cumulative sum of $w$ are computed then fed to the interpolation transform.

**Value**

A scale transformation object.

**Author(s)**

Steven E. Pav <shabbychef@gmail.com>

**See Also**

[trans\\_new](#).

**Examples**

```
set.seed(1234)
ggplot(data.frame(x=rnorm(100),y=runif(100)),aes(x=x,y=y)) +
  geom_point() +
  scale_x_continuous(trans=interp_trans(x=seq(-10,10,by=1),y=cumsum(runif(21))))

set.seed(1234)
ggplot(data.frame(x=rnorm(100),y=runif(100)),aes(x=x,y=y)) +
  geom_point() +
  scale_x_continuous(trans=warp_trans(x=seq(-10,10,by=1),w=runif(21)))

# equivalently:
set.seed(1234)
ggplot(data.frame(x=rnorm(100),y=runif(100)),aes(x=x,y=y)) +
  geom_point() +
  scale_x_continuous(trans=warp_trans(data=data.frame(x=seq(-10,10,by=1),w=runif(21))))
```

```
# this is like trans_sqrt:  
set.seed(1234)  
myx <- seq(0,5,by=0.01)  
ggplot(data.frame(x=rnorm(100),y=runif(100)), aes(x=x,y=y)) +  
  geom_point() +  
  scale_y_continuous(trans=interp_trans(x=myx,y=sqrt(myx)))
```

---

ssqrt\_trans

*Various scale transforms.*

---

## Description

Various scale transformations.

## Usage

ssqrt\_trans

pseudolog10\_trans

## Format

An object of class `trans` of length 7.

## Details

The available transforms:

- `ssqrt_trans` a signed square root transform appropriate for negative or positive numbers.
- `pseudolog10_trans` an asinh transformation, which is like a logarithm, but appropriate for negative or positive numbers. This transformation was taken from the Win Vector blog, <http://www.win-vector.com/blog/2012/03/modeling-trick-the-signed-pseudo-logarithm/>.

## Value

A scale transformation object.

## Author(s)

Steven E. Pav <[shabbychef@gmail.com](mailto:shabbychef@gmail.com)>

## See Also

[trans\\_new](#).

<http://www.win-vector.com/blog/2012/03/modeling-trick-the-signed-pseudo-logarithm/>

**Examples**

```
set.seed(1234)
ggplot(data.frame(x=rnorm(100),y=runif(100)),aes(x=x,y=y)) +
  geom_point() +
  scale_x_continuous(trans=ssqrt_trans)
```

```
set.seed(1234)
ggplot(data.frame(x=rnorm(100),y=runif(100)),aes(x=x,y=y)) +
  geom_point() +
  scale_x_continuous(trans=pseudolog10_trans)
```

---

%of%

*Composition of scale transforms.*

---

**Description**

A binary infix operator that allows one to compose together two scale transformations. We should have that the transformation `a` `%of%` `b` first applies `b`, then applies `a` to the results. This is useful for reversing scales, for example, along with other transformations.

**Usage**

```
a_trans %of% b_trans
```

**Arguments**

<code>a_trans</code>	a transformation object.
<code>b_trans</code>	a transformation object.

**Value**

a transformation object that performs `a` on the output of `b`.

**Author(s)**

Steven E. Pav <shabbychef@gmail.com>

**See Also**

[trans\\_new](#).

**Examples**

```
set.seed(1234)
# compose transformations with %of%:
ggplot(data.frame(x=rnorm(100),y=exp(rnorm(100,mean=-2,sd=4))),aes(x=x,y=y)) +
  geom_point() +
  scale_y_continuous(trans=scales::reverse_trans() %of% scales::log10_trans())
```



# Index

- \* **datasets**
  - ssqrt\_trans, 7
- \* **package**
  - ggallin-package, 2
- \* **plotting**
  - %of%, 8
  - geom\_cloud, 2
  - interp\_trans, 5
  - ssqrt\_trans, 7
- %of%, 8
- aes(), 3
- aes\_(), 3
- borders(), 3
- fortify(), 3
- geom\_cloud, 2
- geom\_ribbon, 4
- ggallin-NEWS, 5
- ggallin-package, 2
- ggplot(), 3
- interp\_trans, 5
- layer(), 3
- pseudolog10\_trans (ssqrt\_trans), 7
- ssqrt\_trans, 7
- trans\_new, 6–8
- warp\_trans (interp\_trans), 5