

# Package: cytofan (via r-universe)

August 31, 2024

**Type** Package

**Title** Plot Fan Plots for Cytometry Data using 'ggplot2'

**Version** 0.1.0.9000

**Description** An implementation of Fan plots for cytometry data in 'ggplot2'. For reference see Britton, E.; Fisher, P. & J. Whitley (1998) The Inflation Report Projections: Understanding the Fan Chart  
<<https://www.bankofengland.co.uk/quarterly-bulletin/1998/q1/the-inflation-report-projections-understanding-the-fan-chart>>).

**License** GPL-3

**LazyData** true

**Depends** R (>= 3.0.0), ggplot2 (>= 2.2.0)

**Imports** RColorBrewer

**Suggests** dplyr, reshape2, bodenmiller, knitr, rmarkdown

**URL** <https://github.com/yannabraham/cytofan>

**BugReports** <https://github.com/yannabraham/cytofan/issues>

**Encoding** UTF-8

**RoxygenNote** 6.0.1

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/yannabraham/cytofan>

**RemoteRef** HEAD

**RemoteSha** cb05c58a5fb5bf932c6d7de8f79efcc462057f5c

## Contents

cytofan	2
do_fan	2
geom_fan	3
StatFan	4

**Index****5**


---

cytofan	<i>cytofan: An implementation of Fan plots in ggplot2.</i>
---------	--

---

**Description**

For reference see [Britton, E.; Fisher, P. & J. Whitley \(1998\) The Inflation Report Projections: Understanding the Fan Chart.](#)

---

do_fan	<i>Compute summary statistics for stat_fan</i>
--------	--

---

**Description**

Extracts the limits of the Ntiles of a distribution for use in the stat\_fan function

**Usage**

```
do_fan(x, step = 0.01)
```

**Arguments**

x	the value to summarize
step	the number of bins to break the data into, based on the quantile function

**Value**

a data.frame containing

- ymin : the lower limit of the quantile
- ymax : the upper limit of the quantile
- id : an identifier for the quantile
- percent : the fill color to use in geom\_fan

**Examples**

```
FanEuStockMarkets <- lapply(colnames(EuStockMarkets),function(id) {
  res <- do_fan(EuStockMarkets[,id])
  res$id <- id
  return(res)
})
FanEuStockMarkets <- do.call(rbind,FanEuStockMarkets)
```

geom\_fan

*Fan plots for trend and population visualizations***Description**

Visualise the distribution of continuous variables by dividing each variables into a fixed number of bins and returning the bin limits. In fan plots ('geom\_fan') bins are grouped over all variables and colored after their distance from the center bin, which corresponds to the median. The center bin corresponds to the strongest shade of 'colorbase', while other bins get decreasing shades.

**Usage**

```
geom_fan(mapping = NULL, data = NULL, position = "identity",
         na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, step = 0.01,
         colorbase = "Oranges", ...)
```

```
stat_fan(mapping = NULL, data = NULL, geom = NULL,
         position = "identity", na.rm = FALSE, show.legend = NA,
         inherit.aes = TRUE, step = 0.01, ...)
```

**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	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . 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. 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.
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
na.rm	If <code>FALSE</code> , the default, missing values are removed with a warning. If <code>TRUE</code> , missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? <code>NA</code> , the default, includes if any aesthetics are mapped. <code>FALSE</code> never includes, and <code>TRUE</code> always includes. It can also be a named logical vector to finely select the aesthetics to display.
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> .
step	the number of quantiles to use to compute bins

colorbase	the colors to use to draw the ribbon. defaults to RColorBrewer ‘Oranges’. See <a href="#">brewer.pal</a> for details.
...	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.
geom	The geometric object to use display the data

### Details

‘stat\_fan’ is suitable only for continuous y data. Moreover, if you have less than ‘1/step’ points you might need to adjust the ‘step’ parameter.

### Computed variables

**ymin** the lower limit of the quantile  
**ymax** the upper limit of the quantile  
**id** an identifier for the quantile  
**percent** the fill color to use in geom\_fan

### Examples

```
# reformat dataset from short-wide to tall-skinny
EuStockMarkets_ts <- lapply(colnames(EuStockMarkets),function(id) {
  data.frame(id=id,value=as.numeric(EuStockMarkets[,id]))
})
EuStockMarkets_ts <- do.call('rbind',EuStockMarkets_ts)

# plot the distribution of the different stock markets
ggplot(EuStockMarkets_ts,aes(x=id,y=value))+
  geom_fan()

# Change the step
ggplot(EuStockMarkets_ts,aes(x=id,y=value))+
  geom_fan(step=0.05)

# change the default color
ggplot(EuStockMarkets_ts,aes(x=id,y=value))+
  geom_fan(colorbase='Greens')

# any valid RColorBrewer palette will work
ggplot(EuStockMarkets_ts,aes(x=id,y=value))+
  geom_fan(colorbase='RdYlGn')
```

---

 StatFan

*StatFan*


---

### Description

StatFan

# Index

## \* datasets

StatFan, 4

aes(), 3

aes\_(), 3

borders(), 3

brewer.pal, 4

cytofan, 2

cytofan-package (cytofan), 2

do\_fan, 2

fortify(), 3

geom\_fan, 3

ggplot(), 3

layer(), 4

stat\_fan (geom\_fan), 3

StatFan, 4