

# Package ‘FunnelPlotR’

October 9, 2019

**Type** Package

**Title** Funnel Plots for Indirectly-Standardised Ratios

**Version** 0.2.1

**Maintainer** Chris Mainey <chris.mainey@uhb.nhs.uk>

**Description** An implementation of the Spiegelhalter (2005) <doi:10.1002/sim.1970> Funnel plots for reporting standardised ratios, with overdispersion adjustment.

**License** MIT + file LICENSE

**URL** <https://chrismainey.github.io/FunnelPlotR>,  
<https://github.com/chrismainey/FunnelPlotR>

**BugReports** <https://github.com/chrismainey/FunnelPlotR/issues>

**Encoding** UTF-8

**LazyData** true

**Imports** ggrepel, scales

**Depends** ggplot2, dplyr, rlang

**RoxygenNote** 6.1.1

**Suggests** testthat (>= 2.1.0), knitr, rmarkdown, COUNT, tidyr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Chris Mainey [aut, cre] (<<https://orcid.org/0000-0002-3018-6171>>),  
Matthew Bass [ctb]

**Repository** CRAN

**Date/Publication** 2019-10-09 08:50:05 UTC

## R topics documented:

funnel_plot . . . . .	2
<b>Index</b>	<b>5</b>

funnel\_plot

*Funnel Plots for Indirectly-Standardised Ratios***Description**

An implementation of funnel plots for indirectly standardised ratios, as described by Spiegelhalter (2005) <doi:10.1002/sim.1970>. There are several parameters for the input, with the assumption that you will want smooth, overdispersed, funnel control limits. Limits may be inflated for overdispersion based on the DerSimonian Laird  $\tau^2$  additive random effects models, originally described for meta-analysis.

**Usage**

```
funnel_plot(numerator, denominator, group, label_outliers = 95,
  Poisson_limits = FALSE, OD_adjust = TRUE, method = "SHMI",
  Winsorise_by = 0.1, title = "Untitled Funnel Plot", multiplier = 1,
  x_label = "Expected", y_label = "Standardised Ratio",
  xrange = "auto", yrange = "auto", return_elements = c("plot",
  "data", "limits"))
```

**Arguments**

numerator	A vector of the numerator (observed events/counts) values. Used as numerator of the Y-axis
denominator	A vector of denominator (predicted/population etc). Used as denominator of the Y-axis and the scale of the x-axis
group	A vector of group names as character or factor. Used to aggregate and group points on plots
label_outliers	Add group labels to outliers on plot. Accepted values are: 95 or 99 corresponding to 95% or 99.8% quantiles of the distribution. Default=99
Poisson_limits	Draw exact Poisson limits, without overdispersion adjustment. (default=FALSE)
OD_adjust	Draw overdispersed limits using hierarchical model, assuming at group level, as described in Spiegelhalter (2012) <doi:https://doi.org/10.1111/j.1467-985X.2011.01010.x>. It calculates a second variance component ' for the 'between' standard deviation (Tau2), that is added to the 'within' standard deviation (sigma) (default=TRUE)
method	Either "CQC" or "SHMI" (default). There are a few methods for standardisation. "CQC"/Spiegelhalter uses a square-root transformation and Winsorises (rescales the outer most values to a particular percentile). SHMI, instead, uses log-transformation and doesn't Winsorise, but truncates the distribution before assessing overdispersion. Both methods then calculate a dispersion ratio (phi) on this altered dataset. This ratio is then used to scale the full dataset, and the plot is drawn for the full dataset.
Winsorise_by	Proportion of the distribution for winsorisation/truncation. Default is 10 % (0.1). Note, this is applied in a two-sided fashion, e.g. 10% refers to 10% at each end of the distribution (20% winsorised/truncated)

title	Plot title
multiplier	Scale relative risk and funnel by this factor. Default to 1, but 100 sometime used, e.g. in some hospital mortality ratios.
x_label	Title for the funnel plot x-axis. Usually expected deaths, readmissions, incidents etc.
y_label	Title for the funnel plot y-axis. Usually a standardised ratio.
xrange	Manually specify the y-axis min and max, in form c(min, max), e.g. c(0, 200). Default, "auto", allows function to estimate range.
yrange	Manually specify the y-axis min and max, in form c(min, max), e.g. c(0.7, 1.3). Default, "auto", allows function to estimate range.
return_elements	a vector of elements to return, options include "plot" for ggplot2 object, "data" for data after processing, and "limits" for control limit lookup table used in the plot. Default is all three objects.

### Details

Outliers are marked based on the grouping, and controlled by 'label\_outliers'. Overdispersion can be factored in based on the methods in Spiegelhalter et al (2012) <doi:https://doi.org/10.1111/j.1467-985X.2011.01010.x>, set 'OD\_adjust' to FALSE to suppress this.

To use Poisson limits set 'Poisson\_limits=TRUE'. This uses 95 It deliberately avoids red-amber-green colouring, but you could extract this from the ggplot object and change manually if you like.

### Value

A list containing [1] the funnel plot as a ggplot2 object, [2] the base table for the plot, [3] the limits table.

### See Also

Statistical methods for healthcare regulation: rating, screening and surveillance. Spiegelhalter et al (2012) <doi:https://doi.org/10.1111/j.1467-985X.2011.01010.x>

Funnel plots for comparing institutional performance. Spiegelhalter (2005) <doi:10.1002/sim.1970>

Handling over-dispersion of performance indicators. Spiegelhalter (2005) <doi:10.1136/qshc.2005.013755>

### Examples

```
#' # We will use the 'medpar' dataset from the 'COUNT' package.
# Little reformatting needed

library(COUNT)
data(medpar)
medpar$provnum<-factor(medpar$provnum)
medpar$los<-as.numeric(medpar$los)

mod<- glm(los ~ hmo + died + age80 + factor(type)
          , family="poisson", data=medpar)
summary(mod)
```

```
# Get predicted values for building ratio
medpar$prds<- predict(mod, type="response")

# Draw plot, returning just the plot object
fp<-funnel_plot(denominator=medpar$prds,numerator=medpar$los,
group = medpar$provnum, return_elements=c("plot"))
fp
```

# Index

`funnel_plot`, [2](#)