

Package ‘audiometry’

May 7, 2026

Type Package

Title Standard Conform Pure Tone Audiometry (PTA) Plots

Version 0.3.0

Author Bernhard Lehnert

Maintainer Bernhard Lehnert <bernhard.lehnert@med.uni-greifswald.de>

Description Facilitates plotting audiometric data (mostly) by preparing the coordinate system according to standards, given e. g. in American Speech-Language-Hearing Association (2005), <doi:10.1044/policy.GL2005-00014>.

Imports ggplot2

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, ggbeeswarm, ggthemes

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-02 11:00:02 UTC

Contents

audiometry	2
boltzmann	2
gg_freiburg	3
gg_pta	4
Index	7

 audiometry

audiometry

Description

a package for standard confirm pur tone audiometry data visualisation using the versatile ggplot2 package.

Details

Right now this is almost only about the gg_pta function to start a ggplot with pure tone audiometry data so that the reference frame looks familiar to audiologists and ent doctors.

 boltzmann

Boltzmann's function

Description

s-shaped curve , originally used as discrimination function to draw the "normal" curves in the Freiburger Sprachtest before I could find the official norm values. Could still be usefull for someone seeking to add something like that to her plots be it as example or for simulation. The function is given as $y = (\exp(-4*(L-L_{50}))\backslash s_{50})^{-1}$ this is similar to a logistic regression result but with parameterization that is expecially usefull here:

Usage

```
boltzmann(L, L_50 = 18.4, s_50 = 0.08)
```

Arguments

L	sound pressure level for which the intelligibility is to be computed
L_50	sound pressure level at 50% intelligibility
s_50	intelligibility at L_50, happens to be 8% in Freiburger Zahleentest and 5% in Freiburger Einsilbertest (values taken from S. Hoth, Der Freiburger Sprachtest, HNO 2016, 64:540-48).

Value

predicted intelligibility

Examples

```
# Freiburger Einsilbertest has L_50 = 29.3 dB and s_50 at 5 %/dB.
# Compute the expected intelligibility at 20, 30 and 40 dB SPL
```

```
boltzmann(L = c(20, 30, 40), L_50 = 29.3, s_50 = .05)
```

Description

Draws the most influential speech intelligibility test in German speaking countries. This function serves as a starting point for plotting data in way that reflects the usual representation of Freiburger Sprachtest results.

Usage

```
gg_freiburg(
  data = data.frame(),
  mapping = aes(),
  horizontal = FALSE,
  xlab = "Sprachschallpegel [dB]",
  ylab = "Sprachverstehen [%]",
  x_ticks_at = seq(0, 110, 10),
  y_ticks_at = seq(0, 100, 20),
  plot_reference = TRUE,
  plot_reference_lwd_1 = 0.8,
  plot_reference_lwd_2 = 0.8,
  plot_reference_color_1 = "darkgrey",
  plot_reference_color_2 = "darkgrey",
  plot_discr_loss_scale = TRUE,
  plot_discr_loss_scale_values = seq(0, 90, 10),
  plot_discr_loss_scale_color = "darkgrey",
  NC_alpha = NULL,
  HV_color = NULL
)
```

Arguments

<code>data</code>	a data.frame that is given to ggplot for initialization
<code>mapping</code>	same as mapping in ggplot2::ggplot
<code>horizontal</code>	logical whether to orient the plot horizontally
<code>xlab</code>	label on the x axis
<code>ylab</code>	label on the y axis
<code>x_ticks_at</code>	vector of x values where numbers on x axis should appear. This is seq(0, 110, 10) in the DIN but c(0, seq(5, 120, 15)) in Muster 13.
<code>y_ticks_at</code>	corresponding to x_ticks_at for the y axis. Set to seq(0,100,10) to mimick the DIN, seq(0, 100, 20) to mimick Muster 13.
<code>plot_reference</code>	logical whether to plot the normal hearing curves for numbers and syllables
<code>plot_reference_lwd_1</code>	line width for reference line 1

```

plot_reference_lwd_2      line width for reference line 2
plot_reference_color_1   line color for reference line 1
plot_reference_color_2   line color for reference line 2
plot_discr_loss_scale    logical whether to print discrimination loss values in the middle of the plot
                          (likely to change in later versions)
plot_discr_loss_scale_values
                          numeric vector of discrimination loss values (likely to change in later versions)
plot_discr_loss_scale_color
                          color value of discrimination loss values (likely to change in later versions)
NC_alpha                  deprecated parameter that was used only in version 0.2.0
HV_color                  deprecated parameter that was used only in version 0.2.0

```

Value

a ggplot suitable for adding Freiburger Sprachtest data as geom_*

Examples

```

library(ggplot2)
data.frame(loud = c(20, 35, 50, 65), perc = c(0,10,65,100)) |>
gg_freiburg(aes(x = loud, y = perc)) +
  geom_point() +
  geom_line()

id = gl(25,4)
gender=gl(2,25, label =c("Frauen", "M\u00e4nner"))
x = rep(c(35, 50, 65, 80), 25)
y = 100*boltzmann(jitter(x,3), 45, .03)
example <- data.frame(Patient=id, Geschlecht = gender, x=x, y=y)
p <- gg_freiburg() +
  geom_boxplot(aes(x = x, y = y, group = x), example) +
  geom_line(aes(x = x, y = y, color = Geschlecht, group = id), example)
print(p)

gg_freiburg(plot_reference_lwd_1 = 2.5, plot_reference_lwd_2 = 3,
            plot_reference_color_1 = "green", plot_reference_color_2 = "pink")

```

Description

Call this to start building a plot based on pure tone audiometry.

Usage

```
gg_pta(
  data = data.frame(),
  theme = theme_light,
  lettermark = NULL,
  lettermarksize = 30,
  xlab = "Frequency in Hertz (Hz)",
  ylab = "Hearing Levels in Decibels (dB)",
  xlim = c(125, 8000),
  xbreaks = c(125, 250, 500, 1000, 2000, 4000, 8000),
  minor_xbreaks = c(750, 1500, 3000),
  x_base_lwd = 1,
  xlabels = c("125", "250", "500", "1000", "2000", "4000", "8000"),
  ylim = c(120, -10),
  yposition = "left"
)
```

Arguments

<code>data</code>	data.frame that contains the data, later to be added to the plot. If no such data.frame is available, can be <code>data = data.frame()</code>
<code>theme</code>	theme for plotting in ggplot2. Can be set to NULL. A different theme can always be added later
<code>lettermark</code>	either "R" or "L" or <code>c("R", "L")</code> to add a letter describing the left or right side (see <code>lettermarksize</code>)
<code>lettermarksize</code>	size of letter for lettermark
<code>xlab</code>	string containing the x axis label
<code>ylab</code>	string containing the y axis label
<code>xlim</code>	limits of the frequencies displayed at the x axis.
<code>xbreaks</code>	frequencies at which major line breaks should be drawn. Must be of same length as <code>xlabels</code>
<code>minor_xbreaks</code>	frequencies at which minor line breaks should be drawn
<code>x_base_lwd</code>	if positive, a line to mark the 0 dB threshold level is drawn, the line width of which is given by <code>x_base_lwd</code> . Set to -1 to turn the line of
<code>xlabels</code>	vector of strings as frequency axis labels. Must be of same length as <code>xbreaks</code> .
<code>ylim</code>	limits of the decibels on the y axis
<code>yposition</code>	side on which to label the y axis: either "right" or "left"

Details

This function is called instead of `ggplot2::ggplot` with a `data.frame` and will return a `ggplot` with fixed axes, fixed axis ratio, ...

Value

a `ggplot` with standard axis ratio, given axis etc. to add geoms to

Author(s)

Bernhard Lehnert

Examples

```
library(ggplot2)
fig1 <- gg_pta(data.frame())
print(fig1)

fig2 <- gg_pta(data.frame(), xlab="Frequency [Hz]", xlim=c(125,12000),
               xbreaks = c(125, 250, 500, 1000, 2000, 4000, 8000, 12000),
               xlabels = c("125", "250", "500", "1k", "2k", "4k", "8k", "12k"))
print(fig2)

expl <- data.frame(x=rep(c(500, 1000, 2000, 4000), 200),
                  y=5 + 70*rbeta(200,1,5))
fig3 <- gg_pta(expl, lettermark = "R",
               xlab="frecuencia", ylab="volumen") +
  geom_boxplot(aes(x=x, y=y, group=x)) +
  theme_grey()
print(fig3)
```

Index

[audiometry](#), 2

[audiometry-package \(audiometry\)](#), 2

[boltzmann](#), 2

[gg_freiburg](#), 3

[gg_pta](#), 4