

Package: BayesianReasoning (via r-universe)

September 9, 2024

Type Package

Title Plot Positive and Negative Predictive Values for Medical Tests

Version 0.4.2.9000

Date 2023-11-13

Maintainer Gorka Navarrete <gorkang@gmail.com>

Description Functions to plot and help understand positive and negative predictive values (PPV and NPV), and their relationship with sensitivity, specificity, and prevalence. See Akobeng, A.K. (2007) <[doi:10.1111/j.1651-2227.2006.00180.x](https://doi.org/10.1111/j.1651-2227.2006.00180.x)> for a theoretical overview of the technical concepts and Navarrete et al. (2015) for a practical explanation about the importance of their understanding <[doi:10.3389/fpsyg.2015.01327](https://doi.org/10.3389/fpsyg.2015.01327)>.

License CC0

URL <https://github.com/gorkang/BayesianReasoning>

BugReports <https://github.com/gorkang/BayesianReasoning/issues>

Depends R (>= 3.5.0)

Imports cli, dplyr, ggforce (>= 0.4.0), ggplot2, ggtext, gt, magrittr, png, reshape2, scales, stats, tibble, tidy

Suggests curl, httr, knitr, patchwork, purrr, rmarkdown, testthat (>= 3.0.0), vdiff, webshot2

VignetteBuilder knitr

Encoding UTF-8

LazyData FALSE

RoxygenNote 7.3.1

Config/testthat/edition 3

Repository <https://gorkang.r-universe.dev>

RemoteUrl <https://github.com/gorkang/bayesianreasoning>

RemoteRef HEAD

RemoteSha f382c4a004947b5476de4ab18615211dbf674e7c

Contents

min_possible_prevalence	2
plot_cutoff	3
PPV_diagnostic_vs_screening	4
PPV_heatmap	5
remove_layers_cutoff_plot	7
Index	8

min_possible_prevalence

Show minimum possible prevalence given the test characteristics

Description

Given a FP and a desired PPV, what is the Minimum Prevalence of a Condition

Usage

```
min_possible_prevalence(Sensitivity = 95, FP_test = 1, min_PPV_desired = 90)
```

Arguments

Sensitivity	Sensitivity of the test: [0-100]
FP_test	False positive rate (1-Specificity): [0-100]
min_PPV_desired	Which PPV is what you consider the minimum to trust a positive result in the test: [0-100]

Value

A description showing the minimum necessary prevalence.

Examples

```
# Example 1
min_possible_prevalence(Sensitivity = 99.9, FP_test = .1, min_PPV_desired = 70)
"To reach a PPV of 70 when using a test with 99.9 % Sensitivity and 0.1 % False Positive Rate,
you need a prevalence of at least 1 out of 429"

# Example 2
min_possible_prevalence(100, 0.1, 98)
"To reach a PPV of 98 when using a test with 100 % Sensitivity and 0.1 % False Positive Rate,
you need a prevalence of at least 1 out of 21"
```

plot_cutoff	<i>plot_cutoff</i> Create a cutoff plot, showing the healthy and sick distributions, and the consequences of different cutoff points
-------------	--

Description

plot_cutoff Create a cutoff plot, showing the healthy and sick distributions, and the consequences of different cutoff points

Usage

```
plot_cutoff(
  prevalence = 0.1,
  cutoff_point = 30,
  mean_sick = 35,
  mean_healthy = 20,
  sd_sick = 3,
  sd_healthy = 5,
  n_people = 1e+05,
  add_table = FALSE,
  output_filename = NULL
)
```

Arguments

prevalence	prevalence of the disease
cutoff_point	cutoff point to use
mean_sick	mean for the sick people distribution
mean_healthy	mean for the healthy people distribution
sd_sick	sd for the sick people distribution
sd_healthy	sd for the healthy people distribution
n_people	number of people to use
add_table	FALSE/TRUE: add gt table with Sensitivity, Specificity, etc.
output_filename	NULL. If a filename, will save the plot

Value

A list with plots and table

Examples

```
## Not run:
plot_cutoff(prevalence = 0.2)

## End(Not run)
```

 PPV_diagnostic_vs_screening

Plot PPV values for a diagnostic and a screening group

Description

Plot PPV associated to different levels of FP and a specific Sensitivity, for two different Prevalence groups.

Usage

```
PPV_diagnostic_vs_screening(
  max_FP = 10,
  Sensitivity = 100,
  prevalence_screening_group = 100,
  prevalence_diagnostic_group = 2,
  labels_prevalence = c("Screening", "Diagnostic"),
  folder = ""
)
```

Arguments

max_FP	False positive rate (1-Specificity) [0-100].
Sensitivity	Sensitivity of the test [0-100].
prevalence_screening_group	Prevalence of the screening group, 1 out of x [1-Inf].
prevalence_diagnostic_group	Prevalence of the diagnostic group, 1 out of x [1-Inf].
labels_prevalence	Labels to use for both groups.
folder	Where to save the plot (the filename would be automatically created using the plot parameters)

Value

Shows a plot or, if given a folder argument, saves a .png version of the plot

Examples

```
# Example 1
PPV_diagnostic_vs_screening(
  max_FP = 10, Sensitivity = 100,
  prevalence_screening_group = 1500,
  prevalence_diagnostic_group = 3
)

# Example 2. QWith custom labels
```

```

PPV_diagnostic_vs_screening(
  max_FP = 10, Sensitivity = 100,
  prevalence_screening_group = 1667,
  prevalence_diagnostic_group = 44,
  labels_prevalence = c("20 y.o.", "50 y.o.")
)

```

 PPV_heatmap

Plot PPV and NPV heatmaps

Description

Plot heatmaps showing the PPV for a given Sensitivity and a range of Prevalences and False Positive values or NPV values for a given Specificity and a range of Prevalences and True Positive values

Usage

```

PPV_heatmap(
  min_Prevalence = 1,
  max_Prevalence = 1000,
  Sensitivity = NULL,
  Specificity = NULL,
  limits_Sensitivity = NULL,
  limits_Specificity = NULL,
  one_out_of = FALSE,
  overlay = "no",
  overlay_labels = "",
  overlay_extra_info = FALSE,
  overlay_position_FP = NULL,
  overlay_position_FN = NULL,
  overlay_prevalence_1 = NULL,
  overlay_prevalence_2 = NULL,
  uncertainty_prevalence = "high",
  label_title = "",
  label_subtitle = "",
  Language = "en",
  folder = "",
  PPV_NPV = "PPV",
  steps_matrix = 100,
  DEBUG = FALSE,
  ...
)

```

Arguments

min_Prevalence [x] out of y prevalence of disease: [1-Inf]
 max_Prevalence x out of [y] prevalence of disease: [1-Inf]

Sensitivity	Sensitivity of test: [0-100]
Specificity	Specificity of test: [0-100]
limits_Sensitivity	c(min Sensitivity, max Sensitivity)
limits_Specificity	c(min Specificity, max Specificity)
one_out_of	Show y scale as 1 out of x [TRUE, FALSE] FALSE by default
overlay	Type of overlay: ["line", "area"]
overlay_labels	Labels for each point in the overlay. For example: c("80", "70", "60", "50", "40", "30", "20 y.o.")
overlay_extra_info	show extra info in overlay? [TRUE/FALSE]
overlay_position_FP	FP value (position in the x-axis) for each point in the overlay. For example: c(7, 8, 9, 12, 14, 14)
overlay_position_FN	FN value (position in the x-axis) for each point in the overlay. For example: c(7, 8, 9, 12, 14, 14)
overlay_prevalence_1	Prevalence value (position in the y-axis) for each point in the overlay. For example: c(1, 1, 1, 2, 1, 1)
overlay_prevalence_2	Prevalence value (position in the y-axis) for each point in the overlay. For example: c(26, 29, 44, 69, 227, 1667)
uncertainty_prevalence	How much certainty we have about the prevalence ["high"/"low"]
label_title	Title for the plot
label_subtitle	Subtitle for the plot
Language	Language for the plot labels: ["sp", "en"]
folder	Where to save the plot (the filename would be automatically created using the plot parameters)
PPV_NPV	Should show PPV or NPV ["PPV", "NPV"]
steps_matrix	width of PPV/NPV matrix. 100 by default
DEBUG	Shows debug warnings [TRUE/FALSE]
...	Other parameters. Now used to pass dpi, height and width in the Show and Save plot section

Value

Shows a plot or, if given a folder argument, saves a .png version of the plot

Examples

```
PPV_heatmap(  
  min_Prevalence = 1,  
  max_Prevalence = 1000,  
  Sensitivity = 100,  
  Specificity = 98,  
  Language = "en"  
)
```

remove_layers_cutoff_plot

remove_layers_cutoff_plot Remove layers from a cutoff plot. This is useful to show how different things are calculated (e.g. Sensitivity)

Description

remove_layers_cutoff_plot Remove layers from a cutoff plot. This is useful to show how different things are calculated (e.g. Sensitivity)

Usage

```
remove_layers_cutoff_plot(cutoff_plot, delete_what, silent = TRUE)
```

Arguments

cutoff_plot	A plot_cutoff() plot
delete_what	Elements to delete (i.e. FP, FN, TP, TN)
silent	TRUE do not show debug info

Value

a cutoff plot without the elements deleted

Examples

```
## Not run:  
PLOT = plot_cutoff(prevalence = 0.2)  
remove_layers_cutoff_plot(PLOT$final_plot, delete_what = c("FN", "TP")) +  
ggplot2::labs(subtitle = "Specificity = TN/(TN+FP)")  
  
## End(Not run)
```

Index

`min_possible_prevalence`, [2](#)

`plot_cutoff`, [3](#)

`PPV_diagnostic_vs_screening`, [4](#)

`PPV_heatmap`, [5](#)

`remove_layers_cutoff_plot`, [7](#)