Package ‘apyramid’

May 8, 2020

Title Visualize Population Pyramids Aggregated by Age

Version 0.1.2

Description Provides a quick method for visualizing non-aggregated line-list or aggregated census data stratified by age and one or two categorical variables (e.g. gender and health status) with any number of values. It returns a ‘ggplot’ object, allowing the user to further customize the output. This package is part of the ‘R4Epis’ project <https://r4epis.netlify.com>.

License GPL-3

Depends R (>= 3.2.0)


BugReports https://github.com/R4EPI/apyramid/issues

Imports ggplot2 (>= 3.0.0), tidyselect, rlang, forcats, dplyr, scales, glue

Suggests testthat (>= 2.1.0), survey, srvyr, vdiffr, covr, outbreaks, knitr, rmarkdown

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

VignetteBuilder knitr

NeedsCompilation no

Author Zhian N. Kamvar [aut, cre] (<https://orcid.org/0000-0003-1458-7108>), Alex Spina [ctb]

Maintainer Zhian N. Kamvar <zkamvar@gmail.com>

Repository CRAN

Date/Publication 2020-05-08 09:50:03 UTC

R topics documented:

  age_pyramid ................................................................. 2
  us_2018 ................................................................. 5
Plot a population pyramid (age-sex) from a dataframe.

**Description**

Plot a population pyramid (age-sex) from a dataframe.

**Usage**

```r
age_pyramid(
  data,
  age_group = "age_group",
  split_by = "sex",
  stack_by = NULL,
  count = NULL,
  proportional = FALSE,
  na.rm = TRUE,
  show_midpoint = TRUE,
  vertical_lines = FALSE,
  horizontal_lines = TRUE,
  pyramid = TRUE,
  pal = NULL
)
```

**Arguments**

- `data` Your dataframe (e.g. linelist)
- `age_group` the name of a column in the data frame that defines the age group categories. Defaults to "age_group"
- `split_by` the name of a column in the data frame that defines the the bivariate column. Defaults to "sex". See NOTE
- `stack_by` the name of the column in the data frame to use for shading the bars. Defaults to NULL which will shade the bars by the split_by variable.
- `count` for pre-computed data the name of the column in the data frame for the values of the bars. If this represents proportions, the values should be within [0, 1].
- `proportional` If TRUE, bars will represent proportions of cases out of the entire population. Otherwise (FALSE, default), bars represent case counts
- `na.rm` If TRUE, this removes NA counts from the age groups. Defaults to TRUE.
- `show_midpoint` When TRUE (default), a dashed vertical line will be added to each of the age bars showing the halfway point for the un-stratified age group. When FALSE, no halfway point is marked.
- `vertical_lines` If you would like to add dashed vertical lines to help visual interpretation of numbers. Default is to not show (FALSE), to turn on write TRUE.
**age_pyramid**

**horizontal_lines**
- If **TRUE** (default), horizontal dashed lines will appear behind the bars of the pyramid.

**pyramid**
- if **TRUE**, then binary split_by variables will result in a population pyramid (non-binary variables cannot form a pyramid). If **FALSE**, a pyramid will not form.

**pal**
- a color palette function or vector of colors to be passed to `ggplot2::scale_fill_manual()`. 
  Defaults to the first "qual" palette from `ggplot2::scale_fill_brewer()`.

**Note**
If the split_by variable is bivariate (e.g. an indicator for a specific symptom), then the result will show up as a pyramid, otherwise, it will be presented as a faceted barplot with empty bars in the background indicating the range of the un-faceted data set. Values of split_by will show up as labels at top of each facet.

**Examples**

```r
library(ggplot2)
old <- theme_set(theme_classic(base_size = 18))

# with pre-computed data ----------------------------------------------------
# 2018/2008 US census data by age and gender
data(us_2018)
data(us_2008)
age_pyramid(us_2018, age_group = age, split_by = gender, count = count)
age_pyramid(us_2008, age_group = age, split_by = gender, count = count)

# 2018 US census data by age, gender, and insurance status
data(us_ins_2018)
age_pyramid(us_ins_2018, 
age_group = age, 
split_by = gender, 
stack_by = insured, 
count = count
)
us_ins_2018$prop <- us_ins_2018$percent/100
age_pyramid(us_ins_2018, 
age_group = age, 
split_by = gender, 
stack_by = insured, 
count = prop, 
proportion = TRUE
)

# from linelist data --------------------------------------------------------
set.seed(2018 - 01 - 15)
ages <- cut(sample(80, 150, replace = TRUE), 
  breaks = c(0, 5, 10, 30, 90), right = FALSE
)
sex <- sample(c("Female", "Male"), 150, replace = TRUE)
```

gender <- sex
gender[sample(5)] <- "NB"
ill <- sample(c("case", "non-case"), 150, replace = TRUE)
dat <- data.frame(
  AGE = ages,
  sex = factor(sex, c("Male", "Female")),
  gender = factor(gender, c("Male", "NB", "Female")),
  ill = ill,
  stringsAsFactors = FALSE
)

# Create the age pyramid, stratifying by sex
print(ap <- age_pyramid(dat, age_group = AGE))

# Create the age pyramid, stratifying by gender, which can include non-binary
print(apg <- age_pyramid(dat, age_group = AGE, split_by = gender))

# Remove NA categories with na.rm = TRUE
dat2 <- dat
dat2[1, 1] <- NA
dat2[2, 2] <- NA
dat2[3, 3] <- NA
print(ap <- age_pyramid(dat2, age_group = AGE))
print(ap <- age_pyramid(dat2, age_group = AGE, na.rm = TRUE))

# Stratify by case definition and customize with ggplot2
ap <- age_pyramid(dat, age_group = AGE, split_by = ill) +
  theme_bw(base_size = 16) +
  labs(title = "Age groups by case definition")
print(ap)

# Stratify by multiple factors
ap <- age_pyramid(dat,
  age_group = AGE,
  split_by = sex,
  stack_by = ill,
  vertical_lines = TRUE
) +
  labs(title = "Age groups by case definition and sex")
print(ap)

# Display proportions
ap <- age_pyramid(dat,
  age_group = AGE,
  split_by = sex,
  stack_by = ill,
  proportional = TRUE,
  vertical_lines = TRUE
) +
  labs(title = "Age groups by case definition and sex")
print(ap)

# empty group levels will still be displayed
dat3 <- dat2
dat3[, dat$AGE == "[0,5)", "sex"] <- NA
age_pyramid(dat3, age_group = AGE)
theme_set(old)

us_2018

US Census data for population, age, and gender

Description

All of these tables were read directly from the excel sources via custom script located at https://github.com/R4EPI/apyramid/blob/master/scripts/read-us-pyramid.R.

Usage

us_2018
us_2008
us_ins_2018
us_ins_2008
us_gen_2018
us_gen_2008

Format

All tables are in long tibble format. There are three columns common to all of the tables:

- **age** [factor] 18 ordered age groups in increments of five years from "<5" to "85+"
- **gender** [factor] 2 reported genders (male, female).
- **count** [integer] Numbers in thousands. Civilian noninstitutionalized and military population.

Below are specifics of each table beyond the stated three columns with names as reported on the US census website

**Population by Age and Sex (us_2018, us_2008):** A tibble with 36 rows and 4 columns.

Additional columns:

- **percent** [numeric] percent of the total US population rounded to the nearest 0.1%

Additional columns:
- **insured** [factor] Either “Insured” or “Not insured” indicating insured status
- **percent** [numeric] percent of each age and gender category insured rounded to the nearest 0.1%


Additional columns:
- **generation** [factor] Three categories of generations in the US: First, Second, Third and higher (see note)
- **percent** [numeric] percent of the total US population rounded to the nearest 0.1%

Note: from the US Census Bureau: The foreign born are considered first generation. Natives with at least one foreign-born parent are considered second generation. Natives with two native parents are considered third-and-higher generation.

An object of class tbl_df (inherits from tbl.data.frame) with 36 rows and 4 columns. An object of class tbl_df (inherits from tbl.data.frame) with 72 rows and 5 columns. An object of class tbl_df (inherits from tbl.data.frame) with 72 rows and 5 columns. An object of class tbl_df (inherits from tbl.data.frame) with 108 rows and 5 columns. An object of class tbl_df (inherits from tbl.data.frame) with 108 rows and 5 columns.

**Source**

Index

*Topic datasets
  us_2018, 5

age_pyramid, 2

ggplot2::scale_fill_brewer(), 3
ggplot2::scale_fill_manual(), 3

tibble, 5

us_2008 (us_2018), 5
us_2018, 5
us_gen_2008 (us_2018), 5
us_gen_2018 (us_2018), 5
us_ins_2008 (us_2018), 5
us_ins_2018 (us_2018), 5