

# Package: apyramid (via r-universe)

September 6, 2024

**Title** Visualize Population Pyramids Aggregated by Age

**Version** 0.1.3

**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 'R4Epi' project <<https://r4epis.netlify.app/>>.

**License** GPL-3

**Depends** R (>= 3.2.0)

**URL** <https://github.com/R4EPI/apyramid>, <https://r4epis.netlify.app/>

**BugReports** <https://github.com/R4EPI/apyramid/issues>

**Imports** ggplot2 (>= 3.0.0), tidyselect, rlang, forcats (>= 1.0.0), dplyr, scales, glue

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

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

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

**RemoteUrl** <https://github.com/r4epi/apyramid>

**RemoteRef** HEAD

**RemoteSha** cb1450315c1ba2c6428ccc4d449b05b79de5c7a0

## Contents

age_pyramid . . . . .	2
us_2018 . . . . .	5

## Index

[7](#)

age\_pyramid

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

## Description

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

## Usage

```
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	<b>for pre-computed data</b> 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.

horizontal_lines	If TRUE (default), horizontal dashed lines will appear behind the bars of the pyramid
pyramid	if TRUE, then binary <code>split_by</code> 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 <code>ggplot2::scale_fill_manual()</code> defaults to the first "qual" palette from <code>ggplot2::scale_fill_brewer()</code> .

### 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

```
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/apyrmaid/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. (us\_2018 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender\\_table1.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender_table1.xls)) (us\_2008 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender\\_table1.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender_table1.xls))

Additional columns:

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

**Health Insurance by Sex and Age** (us\_ins\_2018, us\_ins\_2008): A tibble with 72 rows and 5 columns. (us\_ins\_2018 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender\\_table14.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender_table14.xls)) (us\_ins\_2008 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender\\_table29.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender_table29.xls))

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%

**Generational Distribution of the Population by Sex and Age** (us\_gen\_2018, us\_gen\_2008):

A tibble with 108 rows and 5 columns. (us\_gen\_2018 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender\\_table13.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2018/age-sex-composition/2018gender_table13.xls)) (us\_gen\_2008 source: [https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender\\_table29.xls](https://www2.census.gov/programs-surveys/demo/tables/age-and-sex/2008/age-sex-composition/2008gender_table29.xls))

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

<https://census.gov/data/tables/2018/demo/age-and-sex/2018-age-sex-composition.html>

<https://census.gov/data/tables/2008/demo/age-and-sex/2008-age-sex-composition.html>

# Index

## \* **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](#)