Package 'fingertipscharts'

June 5, 2020

Type Package

Version 0.0.11

Title Produce Charts that you See on the Fingertips Website

Description Use Fingertips charts to recreate the visualisations
that are displayed on the Fingertips website (<http://fingertips.phe.org.uk/>).

Depends R (>= 3.4.0)

BugReports <https://github.com/PublicHealthEngland/fingertipscharts/issues>

Imports dplyr (>= 1.0.0), geojsonio, ggplot2 (>= 3.3), httr, leaflet,
purrr, rlang, scales, sf (>= 0.7), stats, stringr, tibble,
tidyr (>= 1.0.0), utils

Suggests gdtools, knitr, rmarkdown, testthat, vdiffr

License GPL-3

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

NeedsCompilation no

Author Sebastian Fox [aut, cre]

Maintainer Sebastian Fox <sebastian.fox@phe.gov.uk>

Repository CRAN

Date/Publication 2020-06-05 08:30:02 UTC

R topics documented:

---

area_profiles   *Plot spine chart*

Description

Returns ggplot of spine chart

Usage

```
area_profiles(
  data,
  value,
  count,
  area_code,
  local_area_code,
  indicator,
  timeperiod,
  trend = NA,
  polarity,
  significance,
  area_type,
  cols = "fingertips",
  median_line_area_code = "E92000001",
  comparator_area_code = NA,
  bar_width = 0.75,
  local_point_shape = 21,
  local_point_outline = "black",
  comparator_point_shape = 23,
  comparator_point_outline = "gray30",
  comparator_point_fill = "gray30",
  relative_point_size = 1,
  relative_text_size = 1,
```

```
    header_positions = c(-1.83, -1.13, -0.53, -0.35, -0.25, -0.15, -0.05, 1.05),
    header_labels = c("Indicator", "Trend", "Time\nperiod", "Local\ncount",
      "Local\nvalue", "England\nvalue", "Worst/\nLowest", "Best/\nHighest"),
    indicator_label_nudgex = -0.075,
    domain = no_domains,
    relative_domain_text_size = 1,
    show_dividers = "none",
    datatable = TRUE,
    datatable_line_height = 0.6,
    dps = 1,
    percent_display = 0.25,
    arrow_length = 1,
    arrow_thickness = 2,
    arrow_head_length = arrow_length/3,
    arrow_head_angle = 25,
    horizontal_arrow_multiplier = 1
  )
```

Arguments

data	a data frame to create the spine chart from. the data frame should contain data for all area types included in the chart (eg, if plotting for County & UA with a comparator of region and a median line for national, the data frame should contain all of these data)
value	unquoted field name containing the values to be plotted
count	unquoted field name where the count (numerator) is stored
area_code	unquoted field name where area codes are stored (local_area_code, median_line_area_code and comparator_area_code, if using, should all exist in this field)
local_area_code	
	string; the code of the area that the spine chart is being drawn for
indicator	unquoted field name of the field containing the indicator labels. Take care as errors will occur where indicator labels are the same but data exist for multiple sub-categories (for example, sex or age)
timeperiod	unquoted field name of the time period field
trend	unquoted field name of the trend field; if the user doesn't want to display trend information then leave this incomplete and amend the header_labels argument to remove the Trend header. Text within this field should contain one of the following words to control the arrows that are displayed; "decreasing", "increasing", "no significant change", "could not be calculated". The text within this field should contain one of the following words to control the colour; "better", "worse", "no significant change". If none of these words appear in the string, the words "increasing" or "decreasing" will be used to colour the arrows in different shades of blue"
polarity	unquoted field name containing the polarity information (currently only handles polarity returned by fingertipsR package)
significance	unquoted field name describing the statistical significance for that indicator (eg, Better, Worse, Similar etc)

area_type   unquoted field name containing area type information. This ensures the vertabra
                     are only plotted for the same area types as the local_area area type (eg, when
                     plotting a spine chart for County & UA areas, regions and national area types
                     will be removed)

cols   named character vector for the cols that will be applied to the significance field.
                     The names should contain all of the levels in the significance field of the data
                     frame. Defaults to the Fingertips colours based on the outputs from the API

median_line_area_code
                     string; area code for the median line. Defaults to "E92000001" (England)

comparator_area_code
                     string; area code for the comparator point. Defaults to NA

bar_width   numeric value; the distance between bars (0 to 1)

local_point_shape
                     numeric value; shape type for local area point (defaults to 21, circle). See gg-
                     plot2 shape types for different values

local_point_outline
                     string; control colour of the outline of the local point in the spine chart

comparator_point_shape
                     numeric value; shape type for regional area point (defaults to 23, diamond). See
                     ggplot2 shape types for different values

comparator_point_outline
                     string; control colour of the outline of the regional point in the spine chart

comparator_point_fill
                     string; control the fill colour of the regional point in the spine chart

relative_point_size
                     numeric value; control the size of the points on the spine chart

relative_text_size
                     numeric value; control the size of the text in the accompanying table

header_positions
                     numeric vector; used to adjust columns of data table if they are overlapping. The
                     final value shouldn't be less than 1. Must have a length of 7. Defaults to c(-1.43,
                     -.53, -.35, -.25, -.15, -0.05, 1.05)

header_labels   character vector; labels used for the titles of the columns for a data table. Must
                     have a length of 7. Defaults to c("Indicator", "Time period", "Local count","Local
                     value", "England value", "Worst/Lowest","Best/Highest")

indicator_label_nudgex
                     number; nudge the placement of the indicator label in the x direction. Negative
                     values nudge to the left

domain   unquoted field name describing the grouping of the domains if wishing to split
                     the spine chart into domains

relative_domain_text_size
                     numeric; control the text size for the domain labels (if include.domains = TRUE)
                     relative to 1

show_dividers   string; whether to display horizontal lines between indicators. Values can be
                     "all" or "outer". Any other value will not generate lines

datatable   logical; default = TRUE, display data table alongside spine chart

datatable_line_height

                 number; height of wrapped lines in the data table

dps   number; number of decimal places to be displayed in the data table. The default
                 is 1. Set to NA if this should be the same as the input data

percent_display

                 number between 0 and 1; the percentage of values that needs to exist for a spine
                 to display. Default is 0.25

arrow_length   number to control the length of the trend arrow

arrow_thickness

                 number to control the thickness of the trend arrow

arrow_head_length

                 number to control the length of the arrow head

arrow_head_angle

                 number to control the angle of the arrow head

horizontal_arrow_multiplier

                 number to scale horizontal trend arrows. A value below 1 will shorten the arrows

Details

the function draws a bar chart (which is the spine) and then plots the data table (if datatable = TRUE) using geom_text. The bar chart is always plotted between 0 and 1 on the x scale. The columns in the data table are controlled by the header_positions argument. To adjust the length of the bars in the visualisation, amend the header_positions argument. The more negative the first value of the vector that goes into header_positions, the more condensed the bar part of the visualisation will be.

This function filters for the area type that is the same as your local area type and then calculates the "vertebra" from those data. Therefore, if you are comparing outputs with those seen on the Fingertips website, ensure you perform the same preprocessing. For example, some profiles display spine charts where small areas, such as Isles of Scilly, are removed before the spine is produced.

Value

a ggplot object containing a spine chart

Examples

```
## An example with differing decimal places for individual indicators

library(dplyr)
df <- create_test_data() %>%
mutate(Value = case_when(
        grepl("2$|4$|6$", IndicatorName) ~ round(Value,1),
        TRUE ~ round(Value, 0)))
full_p <- area_profiles(df,
                        value = Value,
                        count = Count,
                        area_code = AreaCode,
                        local_area_code = "AC122",
```

```
                                    indicator = IndicatorName,
                                    timeperiod = Timeperiod,
                                    trend = Trend,
                                    polarity = Polarity,
                                    significance = Significance,
                                    area_type = AreaType,
                                    median_line_area_code = "C001",
                                    comparator_area_code = "PAC12",
                                    datatable = TRUE,
                                    relative_domain_text_size = 0.75,
                                    relative_text_size = 1.2,
                                    bar_width = 0.68,
                                    indicator_label_nudgex = -0.1,
                                    show_dividers = "outer",
                                    header_positions = c(-1, -0.7, -0.44, -0.35, -0.25,
                                                         -0.15, -0.05, 1.08),
                                    dps = NA)
full_p

## An example with domains and non-default indicator ordering

df <- create_test_data()
label_order <- c(1, 2, 4, 3, 6, 5)
df <- df %>%
        mutate(IndicatorName = factor(IndicatorName,
                                      levels = paste("Indicator", label_order)))

p <- area_profiles(df,
                   value = Value,
                   count = Count,
                   area_code = AreaCode,
                   local_area_code = "AC122",
                   indicator = IndicatorName,
                   timeperiod = Timeperiod,
                   trend = Trend,
                   polarity = Polarity,
                   significance = Significance,
                   area_type = AreaType,
                   median_line_area_code = "C001",
                   comparator_area_code = "PAC12",
                   datatable = TRUE,
                   relative_domain_text_size = 0.75,
                   relative_text_size = 1.2,
                   bar_width = 0.68,
                   indicator_label_nudgex = -0.1,
                   show_dividers = "outer",
                   header_positions = c(-1, -0.7, -0.53, -0.35, -0.25,
                                        -0.15, -0.05, 1.05),
                   domain = Domain
)
p
```

---

box_plots   *Plot a series of boxplots*

Description

Plot a series of boxplots

Usage

```
box_plots(
  data,
  timeperiod,
  value,
  title = "",
  subtitle = "",
  xlab = "",
  ylab = ""
)
```

Arguments

data	data.frame object to plot using ggplot2 functions
timeperiod	field containing the time period (unquoted)
value	field containing variable to be plotted on x axis (unquoted)
title	string; title of chart
subtitle	string; text to use as subtitle to graph
xlab	string; x-axis title
ylab	string; y-axis title

Value

a ggplot of boxplots for many areas over time

See Also

Other quick charts: compare_areas(), compare_indicators(), map(), overview(), population(), trends()

Examples

```
library(dplyr)
df <- create_test_data()

df_box <- df %>%
        filter(AreaType == "Local") %>%
        arrange(IndicatorName) %>%
```

```
        mutate(Timeperiod = rep(c("2011", "2012", "2013", "2014", "2015", "2016"),
                                each = 100))
p <- box_plots(df_box,
               timeperiod = Timeperiod,
               value = Value,
               title = "Title of chart",
               subtitle = "Boxplot over time",
               ylab = "Proportion (%)")
```

---

compare_areas   *Plot compare areas chart*

Description

Returns ggplot of compare areas chart

Usage

```
compare_areas(
  data,
  area,
  value,
  lowerci,
  upperci,
  fill,
  order = "desc",
  top_areas,
  title = "",
  xlab = "",
  ylab = "",
  legend.position = "bottom",
  display.values = FALSE,
  dps = 1
)
```

Arguments

data	data.frame object to plot using ggplot2 functions
area	field containing variable to be plotted on y axis (unquoted)
value	field containing variable to be plotted on x axis (unquoted)
lowerci	field containing variable to be plotted as lower confidence interval (unquoted - not required)
upperci	string; field containing variable to be plotted as upper confidence interval (unquoted - not required)
fill	field to be used to determine the colouring of the bars (unquoted)
order	one of "alphabetical", "asc" or "desc" - to determine how to order the bars

top_areas	character vector; the areas to fix at the top of the chart
title	string; title of chart
xlab	string; x-axis title
ylab	string; y-axis title
legend.position	
	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)
display.values	logical; where or not to display the rounded values next to the bars on the chart
dps	number; number of decimal places to be displayed when display.values = TRUE. The default is 1.

Value

a ggplot of a compare areas chart

See Also

Other quick charts: box_plots(), compare_indicators(), map(), overview(), population(), trends()

Examples

```
library(dplyr)
df <- create_test_data()
parent <- "PAC11"
top_names <- c("C001", parent)
ordered_levels <- c("Better",
                    "Similar",
                    "Worse",
                    "Not compared")
df_ca <- df %>%
      filter(IndicatorName == "Indicator 3",
             (AreaCode %in% top_names |
                    ParentAreaCode == parent))
p <- compare_areas(df_ca, AreaCode, Value,
                  fill = Significance,
                  lowerci = LCI,
                  upperci = UCI,
                  order = "desc",
                  top_areas = top_names,
                  title = "Compare the local areas")
p
```

---

compare_indicators *Plot compare indicators plot*

Description

Plot compare indicators plot

Usage

```
compare_indicators(
  data,
  x,
  y,
  xlab = "",
  ylab = "",
  point_size = 4,
  highlight_area,
  area,
  add_R2 = FALSE
)
```

Arguments

data	data.frame object to plot using ggplot2 functions
x	field containing x variable (unquoted)
y	field containing y variable (unquoted)
xlab	string; x-axis title
ylab	string; y-axis title
point_size	number; size of point
highlight_area	character vector; list of areas for highlighting
area	field containing areas - should contain contents of highlight_area. Only required if highlight_area has a value (unquoted)
add_R2	boolean; should R2 be displayed?

Value

a ggplot of compare indicators for 2 indicators

See Also

Other quick charts: box_plots(), compare_areas(), map(), overview(), population(), trends()

Examples

```
library(tidyr)
library(dplyr)
df <- create_test_data()

df_ci <- df %>%
        filter(IndicatorName %in% c("Indicator 1", "Indicator 3")) %>%
        select(IndicatorName, AreaCode, Value) %>%
        pivot_wider(names_from = IndicatorName,
                    values_from = Value) %>%
        rename(Ind1 = `Indicator 1`,
               Ind3 = `Indicator 3`) %>%
        mutate(Ind2 = runif(nrow(.), min = Ind1 * 0.5, max = Ind1 * 1.5))
p <- compare_indicators(df_ci,
                        x = Ind1,
                        y = Ind3,
                        xlab = "Indicator 1 label",
                        ylab = "Indicator 3 label",
                        highlight_area = c("C001", "AC172"),
                        area = AreaCode,
                        add_R2 = TRUE)
p
```

---

create_datatable   *Data table supporting information*

Description

Returns a data frame containing the data that sits next to the spine chart

Usage

```
create_datatable(
  data,
  indicator,
  area_code,
  timeperiod,
  trend,
  count,
  value,
  local_area_code,
  median_line_area_code,
  comparator_area_code,
  dps = 1,
  header_width,
  horizontal_arrow_multiplier
)
```

Arguments

`data`	a data frame to create the spine chart from. the data frame should contain data for all area types included in the chart (eg, if plotting for County & UA with a comparator of region and a median line for national, the data frame should contain all of these data)
`indicator`	unquoted field name of the field containing the indicator labels. Take care as errors will occur where indicator labels are the same but data exist for multiple sub-categories (for example, sex or age)
`area_code`	unquoted field name where area codes are stored (local_area_code, median_line_area_code and comparator_area_code, if using, should all exist in this field)
`timeperiod`	unquoted field name of the time period field
`trend`	unquoted field name of the trend field; if the user doesn't want to display trend information then leave this incomplete and amend the header_labels argument to remove the Trend header. Text within this field should contain one of the following words to control the arrows that are displayed; "decreasing", "increasing", "no significant change", "could not be calculated". The text within this field should contain one of the following words to control the colour; "better", "worse", "no significant change". If none of these words appear in the string, the words "increasing" or "decreasing" will be used to colour the arrows in different shades of blue"
`count`	unquoted field name where the count (numerator) is stored
`value`	unquoted field name containing the values to be plotted
`local_area_code`	
	string; the code of the area that the spine chart is being drawn for
`median_line_area_code`	
	string; area code for the median line. Defaults to "E92000001" (England)
`comparator_area_code`	
	string; area code for the comparator point. Defaults to NA
`dps`	number of decimal places to use in the data table
`header_width`	x dimension of chart to be used for normalising the arrow length when horizonal
`horizontal_arrow_multiplier`	
	number to scale horizontal trend arrows. A value below 1 will shorten the arrows

Value

A data frame containing the information that sits alongside the spine chart

---

create_test_data   *Create test data*

Description

Create dataset to be used in tests (useful for demo purposes too)

Usage

```
create_test_data()
```

Value

a dummy data.frame of data

Examples

```
df <- create_test_data()
```

```
ensure_ons_api_available
```
Check if the given ONS json is available

Description

Check if the given ONS json is available

Usage

```
ensure_ons_api_available(ons_api)
```

Arguments

ons_api   string; GeoJSON address provided from the ONS geography portal

Value

TRUE if the API is available, otherwise stop() is called.

```
fingertipscharts
```
fingertipscharts: A package for creating the visualisations displayed in Fingertips

---

Description

The fingertipscharts package provides one type of function

quick charts functions

Easy to produce Fingertips charts using tidyverse syntax

---

map   *Plot a choropleth map for an indicator*

Description

Plot a choropleth map for an indicator

Usage

```
map(
  data,
  ons_api,
  area_code,
  fill,
  type = "static",
  value,
  name_for_label,
  title = "",
  subtitle = "",
  copyright_size = 4,
  copyright_year = Sys.Date()
)
```

Arguments

`data`	data.frame object to plot using ggplot2 functions
`ons_api`	string; GeoJSON address provided from the ONS geography portal
`area_code`	field containing area codes to join to shape file imported from ONS API
`fill`	field to be used to determine the colouring of the areas (unquoted)
`type`	string; the output map required. Can be "static" or "interactive"
`value`	field containing variable to be plotted on x axis (unquoted)
`name_for_label`	if interactive map, name of field containing area names to be used for label (unquoted) - optional
`title`	string; title of chart
`subtitle`	string; text to use as subtitle to graph
`copyright_size`	number; fix the size of the copyright text
`copyright_year`	number (length 4 characters) or Date class; the copyright year displayed at bottom of the map. Applies to static maps only

Value

a either a static or interactive ggplot choropleth map

See Also

Other quick charts: [box_plots](#)(), [compare_areas](#)(), [compare_indicators](#)(), [overview](#)(), [population](#)(), [trends](#)()

Examples

```
## Not run:
ons_api <- "https://opendata.arcgis.com/datasets/687f346f5023410ba86615655ff33ca9_4.geojson"

p <- map(mapdata,
         ons_api = ons_api,
         area_code = AreaCode,
         fill = Significance,
         title = "Map example",
         subtitle = "An indicator for Upper Tier Local Authorities England",
         copyright_year = 2019)


p

## For an interactive (leaflet) map
p <- map(mapdata,
         ons_api = ons_api,
         area_code = AreaCode,
         fill = Significance,
         type = "interactive",
         value = Value,
         name_for_label = AreaName,
         title = "An indicator for Upper Tier<br>Local Authorities England")
p
## End(Not run)
```

---

mapdata	*Dummy data for Upper Tier Local Authorities so dummy maps can be produced*

Description

A dataset of indicator values for all Upper Tier Local Authorities

Usage

```
mapdata
```

Format

A data frame with 152 records and 4 fields:

AreaCode Office for National Statistics area codes

AreaName Office for National Statistics area names

Significance The statistical significance of the area compared to a benchmark

Value The indicator value for the data

---

overview   *Plot an overview (tartan rug) of multiple indicators*

Description

Plot an overview (tartan rug) of multiple indicators

Usage

```
overview(
  data,
  area,
  indicator,
  value,
  fill,
  timeperiod,
  top_areas,
  wrap_length = 50,
  value_label_size = 1,
  legend_position = "none"
)
```

Arguments

data	data.frame object to plot using ggplot2 functions
area	field containing area names (unquoted)
indicator	field containing indicator names (unquoted)
value	field containing variable to be plotted (unquoted)
fill	field to be used to determine the colouring of the bars (unquoted)
timeperiod	field containing the time period (unquoted)
top_areas	character vector; the areas to fix at the left
wrap_length	number; maximum number of characters in indicator before wrapping it
value_label_size	
	number; amount to scale the size of the value label
legend_position	
	the position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector)

Value

a ggplot of the overview/tartan rug plot

See Also

Other quick charts: [box_plots](), [compare_areas](), [compare_indicators](), [map](), [population](),
[trends]()

Examples

```
library(dplyr)
df <- create_test_data()

parent <- "PAC14"
top_names <- c("C001", parent)
df_over <- df %>%
        filter((AreaCode %in% top_names |
                        ParentAreaCode == parent)) %>%
        mutate(Value = round(Value, 1))
p <- overview(df_over,
              area = AreaCode,
              indicator = IndicatorName,
              value = Value,
              timeperiod = Timeperiod,
              fill = Significance,
              top_areas = top_names,
              wrap_length = 40,
              value_label_size = 0.8)
p
```

---

population   *Plot population pyramid*

Description

Plot population pyramid

Usage

```
population(
  data,
  value,
  sex,
  age,
  area,
  area_name,
  comparator_1,
  comparator_2,
  title,
  subtitle,
  xlab
)
```

Arguments

data	data.frame object to plot using ggplot2 functions
value	field containing variable to be plotted on x axis (unquoted)
sex	field containing sex variable (unquoted)
age	field containing age variable (unquoted)
area	field containing variable to be plotted on y axis (unquoted)
area_name	string; name of the local area (this should exist in the field described by the area parameter)
comparator_1	string; name of comparator area (this should exist in the field described by the area parameter)
comparator_2	string; name of comparator area (this should exist in the field described by the area parameter)
title	string; title of chart
subtitle	string; text to use as subtitle to graph
xlab	string; x-axis title

Value

a ggplot of a population pyramid against 2 optional comparators

See Also

Other quick charts: `box_plots()`, `compare_areas()`, `compare_indicators()`, `map()`, `overview()`, `trends()`

Examples

```
library(dplyr)
agelevels <- c("0-4", "5-9","10-14","15-19",
               "20-24","25-29","30-34",
               "35-39","40-44","45-49",
               "50-54","55-59","60-64",
               "65-69","70-74","75-79",
               "80-84","85-89","90+")
areas <- c("Area 1", "Area 2", "Area 3")
pops <- data.frame(Age = factor(rep(agelevels, length(areas) * 2),
                                levels = agelevels),
                   Value = rep(sample(1000:3000, length(agelevels), replace = TRUE),
                               length(areas) * 2),
                   Sex = rep(rep(c("Male", "Female"),
                                 each = length(agelevels)), length(areas)),
                   AreaName = rep(areas, each = length(agelevels) * 2))

p <- population(pops,
                value = Value,
                sex = Sex,
                age = Age,
```

```
                  area = AreaName,
                  area_name = "Area 1",
                  comparator_1 = "Area 3",
                  comparator_2 = "Area 2",
                  title = "Age Profile",
                  subtitle = "2015/16",
                  xlab = "% of total population")
    p
```

---

round2 *Proper rounding of values*

Description

Proper rounding of values

Usage

```
round2(val, dps)
```

Arguments

val	numeric value to round
dps	numeric, number of decimal places

Details

function taken from this link (<https://stackoverflow.com/questions/12688717/round-up-from-5>)

---

scale_fill_phe *Fingertips scale fill for ggplot2*

Description

Fingertips scale fill

Usage

```
scale_fill_phe(theme = "fingertips", ...)
```

Arguments

theme	string; current options are fingertips only for discrete scales
...	inputs to the scale_manual (for discrete values)

---

spine_data_check   *Check function for multiple values for an area in an indicator for spine chart*

Description

Check function for multiple values for an area in an indicator for spine chart

Usage

```
spine_data_check(data, indicator, area_code)
```

Arguments

data	a data frame to create the spine chart from. the data frame should contain data for all area types included in the chart (eg, if plotting for County & UA with a comparator of region and a median line for national, the data frame should contain all of these data)
indicator	unquoted field name of the field containing the indicator labels. Take care as errors will occur where indicator labels are the same but data exist for multiple sub-categories (for example, sex or age)
area_code	unquoted field name where area codes are stored (local_area_code, median_line_area_code and comparator_area_code, if using, should all exist in this field)

---

spine_preprocess   *Preprocess data for spine chart*

Description

Returns a data frame with the latest time period of data for each indicator name.

Usage

```
spine_preprocess(data, indicator, timeperiod_sortable)
```

Arguments

data	a data frame to create the spine chart from. the data frame should contain data for all area types included in the chart (eg, if plotting for County & UA with a comparator of region and a median line for national, the data frame should contain all of these data)
indicator	unquoted field name for indicators. This should be what is presented as the label for the final spine chart, hence should be unique for each vertabra. Be careful the indicator doesn't have sub-categories based on other fields, such as sex (male, female, persons) or age group

timeperiod_sortable

> unquoted field name containing the time period that is numeric and sortable, such that higher values are a later time period

Details

This processing only takes place on the indicator field and the time period field provided. If the data contains multiple sexes or age groups for an indicator, make sure the indicator field reflects this.

Value

A processed data frame for latest time periods of given indicators

---

spine_rescaler   *Rescale spine data*

Description

Rescales data so it can be plotted on a spine chart

Usage

```
spine_rescaler(
  data,
  area_code,
  indicator,
  significance,
  polarity,
  area_type,
  value,
  timeperiod,
  local_area_code,
  median_line_area_code,
  comparator_area_code = NA,
  percent_display,
  dps = 1
)
```

Arguments

data
> a data frame to create the spine chart from. the data frame should contain data for all area types included in the chart (eg, if plotting for County & UA with a comparator of region and a median line for national, the data frame should contain all of these data)

area_code
> unquoted field name where area codes are stored (local_area_code, median_line_area_code and comparator_area_code, if using, should all exist in this field)

indicator	unquoted field name of the field containing the indicator labels. Take care as errors will occur where indicator labels are the same but data exist for multiple sub-categories (for example, sex or age)
significance	unquoted field name describing the statistical significance for that indicator (eg, Better, Worse, Similar etc)
polarity	unquoted field name containing the polarity information (currently only handles polarity returned by fingertipsR package)
area_type	unquoted field name containing area type information. This ensures the vertabra are only plotted for the same area types as the local_area area type (eg, when plotting a spine chart for County & UA areas, regions and national area types will be removed)
value	unquoted field name containing the values to be plotted
timeperiod	unquoted field name of the time period field
local_area_code	
	string; the code of the area that the spine chart is being drawn for
median_line_area_code	
	string; area code for the median line. Defaults to "E92000001" (England)
comparator_area_code	
	string; area code for the comparator point. Defaults to NA
percent_display	
	number between 0 and 1; the percentage of values that needs to exist for a spine to display. Default is 0.25
dps	number; number of decimal places to be displayed in the data table. The default is 1. Set to NA if this should be the same as the input data

Value

A list containing "bars" and "points" which contains data that can be passed to the phe_spine_chart function

---

theme_phe   *Fingertips theme for ggplot2*

Description

fingertips theme

Usage

```
theme_phe(
  theme = "fingertips",
  base_size = 11,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22
)
```

Arguments

`theme`	string; theme of chart, current are fingertips only
`base_size`	base font size
`base_family`	base font family
`base_line_size`	base size for line elements
`base_rect_size`	base size for rect elements

`trends`	*Plot trend chart*

Description

Plot trend chart

Usage

```
trends(
  data,
  timeperiod,
  value,
  area,
  comparator,
  area_name,
  fill,
  lowerci,
  upperci,
  title = "",
  subtitle = "",
  xlab = "",
  ylab = "",
  point_size = 4
)
```

Arguments

`data`	data.frame object to plot using ggplot2 functions
`timeperiod`	field containing the time period (unquoted)
`value`	field containing variable to be plotted on x axis (unquoted)
`area`	field containing variable to be plotted on y axis (unquoted)
`comparator`	string; name of comparator area (this should exist in the field described by the area parameter)
`area_name`	string; name of the local area (this should exist in the field described by the area parameter)
`fill`	field to be used to determine the colouring of the bars (unquoted)

lowerci	field containing variable to be plotted as lower confidence interval (unquoted - not required)
upperci	string; field containing variable to be plotted as upper confidence interval (unquoted - not required)
title	string; title of chart
subtitle	string; text to use as subtitle to graph
xlab	string; x-axis title
ylab	string; y-axis title
point_size	number; size of point

Value

a ggplot of trends for an indicator alongside a comparator

See Also

Other quick charts: box_plots(), compare_areas(), compare_indicators(), map(), overview(), population()

Examples

```
library(dplyr)
df <- create_test_data()

df_trend <- df %>%
        arrange(IndicatorName) %>%
        mutate(Timeperiod = rep(c("2011", "2012", "2013", "2014", "2015", "2016"),
                                each = 111))
p <- trends(df_trend,
            timeperiod = Timeperiod,
            value = Value,
            area = AreaCode,
            comparator = "C001",
            area_name = "AC142",
            fill = Significance,
            lowerci = LCI,
            upperci = UCI,
            title = "Trend compared to country",
            subtitle = "For area AC142",
            xlab = "Year",
            ylab = "Value (%)")
p
```

Index