

Package: UpAndDownPlots (via r-universe)

August 31, 2024

Type Package

Title Displays Percentage and Absolute Changes

Version 0.5.0

Description Displays percentage changes by height and absolute changes by area for up to three nested or non-nested levels. The plots visualise changes in indices and markets, showing how the changes for sectors or for individual components contribute to the overall change. Data can be classified by up to three levels of grouping variables in a layered, hierarchical plot. Each level can be ordered in several ways including by baseline, by percentage change, and by absolute change. The vignettes give examples.

Depends R (>= 3.5.0)

Imports tidyverse, ggplot2, dplyr, tidyr, forcats, gridExtra, ggthemes

License GPL (>= 2)

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, colorspace

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Antony Unwin [aut, cre, cph], Bill Venables [ctb]

Maintainer Antony Unwin <unwin@math.uni-augsburg.de>

Date/Publication 2024-04-02 12:42:08 UTC

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

RemoteUrl <https://github.com/cran/UpAndDownPlots>

RemoteRef HEAD

RemoteSha 32034805da33bf79e24d17c9856203c57f076d43

Contents

AutoSalesGermany	2
AutoSalesXGermany	3
ConsumerPriceIndexUK	4
dgroup	5
NIpop	6
sort5	7
ud_colours	8
ud_plot	9
ud_prep	11
UpAndDownPlots	12
Index	13

AutoSalesGermany	<i>Sales of cars and vans in Germany in the years 2017 and 2018.</i>
------------------	--

Description

Sales of cars and vans in Germany for 2017 and 2018. The market is divided into 14 segments and in the two years 2017 and 2018 there were 50 competing manufacturers.

Usage

```
data(AutoSales)
```

Format

A data frame with 393 observations on the following 7 variables.

Sector Car, Van, Other

Segment The KBA splits the market into 14 segments

ModelSeries Full vehicle name reported by KBA

Manufacturer Manufacturer

Model Model name

sales17 Sales in 2017

sales18 Sales in 2018

Details

The Kraftfahrt-Bundesamt (KBA, National Vehicle Authority in Germany) publishes spreadsheets of sales every month including year-to-date figures. The full vehicle name has been split into manufacturer and model names. NAs have been transformed to 0's. The variable names and market segments have been anglicised.

A 'cleaned' and reduced version of the dataset is available as AutoSalesX in this package.

Source

Kraftfahrt-Bundesamt
https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/MonatlicheNeuzulassungen/monatl_neuzulassungen_node.html

Examples

```
yy <- ud_prep(AutoSales, v1="sales17", v2="sales18", levs=c("Segment"), sortLev=c("perc"))
y1 <- ud_plot(yy, labelvar="Segment")
y1$uadl
```

AutoSalesXGermany	<i>Sales of cars and vans in Germany by manufacturer (edited version of AutoSales).</i>
-------------------	---

Description

Sales of cars and vans in Germany for 2017 and 2018. The market is divided into 14 segments and in the two years 2017 and 2018 there were 50 competing manufacturers.

Usage

```
data(AutoSalesX)
```

Format

A data frame with 173 observations on the following 5 variables.

Sector Car, Van, Other

Segment Market segment

Manufacturer Manufacturer

sales17 Sales in 2017

sales18 Sales in 2018

Details

This is a 'cleaned' and reduced version of the dataset AutoSales from this package.

Vehicles selling less than 1000 in both years have been reclassified as 'Other'. The data have been aggregated by manufacturer within segments.

Source

Kraftfahrt-Bundesamt
https://www.kba.de/DE/Statistik/Fahrzeuge/Neuzulassungen/MonatlicheNeuzulassungen/monatl_neuzulassungen_node.html

Examples

```
library(dplyr)
yx <- ud_prep(AutoSalesX %>% filter(Segment=="Compact"), v1="sales17",
              v2="sales18", levs=c("Manufacturer"), sortLev=c("base"))
y5 <- ud_plot(yx, labelvar="Manufacturer")
y5$uadl
```

ConsumerPriceIndexUK *Consumer Price Index data for the UK for August 2017 and August 2018.*

Description

The UK Office for National Statistics publishes updated Consumer Price Index data every month. There are 12 sectors with from 1 to 7 subsectors and the subsectors comprise up to 9 items.

Usage

```
data(CPIuk)
```

Format

A data frame with 85 observations on the following 10 variables.

ID ID number made up of Sector, Subsector, Item numbers

SectorID Sector number

SubsectorID Subsector number within sector

Item Item number within Sector

Sector Sector label

Subsector Subsector label

Component Item label

Weight Item weight in 2017

Aug2017 Item index value in August 2017

Aug2018 Item index value in August 2018

Source

Office for National Statistics (ONS) UK

<https://www.ons.gov.uk/economy/inflationandpriceindices/datasets/consumerpriceinflation/>

Examples

```
yw <- ud_prep(CPIuk, weight="Weight", v1="Aug2017", v2="Aug2018",
              levs=c("Sector"), sortLev=c("perc"))
y2 <- ud_plot(yw, labelvar="Sector")
y2$uadl
```

dgroup *Draws UpAndDown plots for different grouping variables*

Description

This function draws plots to compare the effects of different grouping variables.

Usage

```
dgroup(data, byvars, weight=1, v1, v2)
```

Arguments

data	the dataset to be analysed
byvars	the grouping variables to be used
weight	a variable for weighting the individual items, the default is for every item to be weighted equally
v1	the variable specifying the start value in the comparison
v2	the variable specifying the end value in the comparison

Details

This function prepares UpAndDown plots showing percentage and absolute changes for different grouping variables. All unlabelled plots have the same vertical scale to ease comparisons. Scales of labelled plots are affected by label lengths.

To draw an unlabelled or labelled set of plots, apply `grid.arrange` to the appropriate output of `dgroup`.

Value

uad	unlabelled horizontal UpAndDown plots
uad1	vertical UpAndDown plots with labelling

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

Examples

```
library(gridExtra)
zs <- dgroup(NIpop, byvars=c("age", "gender", "LGD2014_name", "area_name"), v1="y2011", v2="y2017")
grid.arrange(zs$uadg)
grid.arrange(zs$uadg1)
```

NIpop

Northern Ireland population estimates 2011-2017

Description

The Northern Ireland Statistics & Research Agency (NISRA) publishes population estimates on their webpage for 80 District Electoral Areas (DEA). Each of these DEAs is in one of the 11 Local Government Districts (LGD). The data are further broken down by gender and age groups. This dataset includes estimates for the years 2011 to 2017.

Usage

```
data(NIpop)
```

Format

A data frame with 640 observations on the following 11 variables.

area_name District Electoral Area

LGD2014_name Local Government District

gender Males or Females

age four age groups: 00-15, 16-39, 40-64, 65+

y2011 population estimates for the year 2011

y2012 population estimates for the year 2012

y2013 population estimates for the year 2013

y2014 population estimates for the year 2014

y2015 population estimates for the year 2015

y2016 population estimates for the year 2016

y2017 population estimates for the year 2017

Source

Northern Ireland Statistics & Research Agency

https://www.nisra.gov.uk/sites/nisra.gov.uk/files/publications/MYE17_DEA_1117.xlsx

Examples

```
km <- ud_prep(NIpop, v1="y2011", v2="y2017", levs=c("age", "gender"), sortLev=c("orig", "perc"))
k2 <- ud_plot(km, labelvar="age")
k2$uad1
```

sort5	<i>Draws five UpAndDown plots for one grouping variable with the five different sorting methods provided</i>
-------	--

Description

This function draws plots to compare the results of different sorting methods for a single grouping variable

Usage

```
sort5(data, gvar, weight=1, v1, v2)
```

Arguments

data	the dataset to be analysed
gvar	the grouping variable to be used in all five plots
weight	a variable for weighting the individual items, the default is for every item to be weighted equally
v1	the variable specifying the start value in the comparison
v2	the variable specifying the end value in the comparison

Details

This function prepares UpAndDown plots showing percentage and absolute changes for one grouping variable with five different sortings.

To draw an unlabelled or labelled set of plots, apply `grid.arrange` to the appropriate output of `sort5`.

Value

uad	five unlabelled horizontal UpAndDown plots
uadl	five vertical UpAndDown plots with labelling of groups

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

Examples

```
library(gridExtra)
xs <- sort5(NIpop, gvar="LGD2014_name", v1="y2011", v2="y2017")
grid.arrange(xs$uad)
grid.arrange(xs$uadl)
```

`ud_colours`*Set colours for UpAndDown plots*

Description

Provides a colour scheme for UpAndDown plots.

Usage

```
ud_colours(colours = c("grey75", "grey65", "red", "blue", "brown"), colors,  
           gcpal=colorblind_pal()(8))
```

Arguments

<code>colours</code>	A set of colours for UpAndDown plots. The first one is used for filling bars and the second for the borders of bars; the third one is for colouring the dashed line for the overall percentage change; the fourth one is for colouring the borders of rectangles representing the highest grouping level; the last one is for colouring the borders of a second grouping level (if there is one).
<code>colors</code>	To allow users to write 'colors' instead of 'colours'.
<code>gcpal</code>	A colour palette to use if bars are to be coloured by a grouping variable (the parameter <code>barColour</code> in <code>ud_plot</code>). The default is the colorblind palette from <code>ggthemes</code> , which has 8 colours. If more are needed, a palette is generated from the <code>randomcoloR</code> package, always using the same seed for replication purposes. The same procedure applies for a user-defined palette—if there are not enough colours, a <code>randomcoloR</code> palette is used.

Details

`ud_colours` is provided for assigning colours for UpAndDown plots.

Value

A named list of colours.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

[ud_plot](#)

Examples

```
# Using default colours
yw <- ud_prep(CPIuk, weight="Weight", v1="Aug2017", v2="Aug2018",
              levs=c("Sector"), sortLev=c("perc"))
y2 <- ud_plot(yw, labelvar="Sector")
y2$uadl

# Using individually chosen colours
library(ggthemes)
y2a <- ud_plot(yw, ud_control=ud_colours(colours = c("lightblue1", "grey50",
          "red4", "blue", "brown")))
y2a$uadl

# Using barColour and the default palette
# A random palette from colorspace is used as there are more than 8 sectors.
y2b <- ud_plot(yw, barColour="Sector")
y2b$uadl

# Using barColour and a specified palette
y2c <- ud_plot(yw, levelColour="Sector", ud_control=ud_colours(
  gcpal=sample(colorspace::qualitative_hcl(n = 20, l = 80))))
y2c$uadl
```

ud_plot	<i>Draws an UpAndDown plot to display percentage and absolute changes</i>
---------	---

Description

Function for drawing UpAndDown plots for up to 3 levels

Usage

```
ud_plot(outPrep, b=0, totperc="yes", vscale=NULL, labelvar=NULL, drawFrom="BigToSmall",
        levelColour="none", barColour=levelColour, ud_control=ud_colours())
```

Arguments

outPrep	a list of the grouping variables chosen, the sorting methods for them, nesting information, the numbers of categories in them, and the sorted dataset to be plotted
b	the baseline value from which the bars are drawn. The default is 0. It can be informative to set it to the overall rate or to some threshold value.
totperc	whether the dashed line for overall percentage change is to be drawn. The default is "yes". It is useful to set it to "no" if you draw an UpAndDown plot for a subset and do not want to draw the rate for the subset in this way.
vscale	sets the limits for the vertical axis scale. Default limits are calculated from the data.

labelvar	the name of the variable to be used for labelling. If specified, it has to be one of the first two grouping variables. Labelling is only used for vertical UpAndDown plots.
drawFrom	specifies the order in which the levels are drawn, either "BigToSmall" (top level first) or "SmallToBig" (lowest level first).
levelColour	specifies by variable name which level will be coloured (even if there is only one level). The default is not to colour them, other than grey for the top level bars.
barColour	specifies a character or factor variable to colour (fill) the bars. The default is to use the levelColour variable itself.
ud_control	a list of colours for UpAndDown plots. If omitted, ud_colours gives the defaults.

Details

This function takes the output from [ud_prep](#) and draws an UpAndDown plot showing percentage and absolute changes for up to three levels. The levels may be nested or not. There are a number of options for sorting, labelling, and colouring the displays. The plots produced are ggplot2 objects, so you can amend them—to some extent—yourself. Choosing a barColour variable with many different colours is generally not a good idea and takes a while to draw because of the lengthy legend. See [ud_colours](#) for choosing palettes.

Value

uad	an unlabelled horizontal UpAndDown plot
uad1	a vertical UpAndDown plot with labelling of groups or items (NULL if no labelvar is set)
TotPerc	the percentage change for all the data used in the plot (the value displayed by the red dashed line)
level1	the category names and percentage changes plotted for the first level specified in the input levs
level2	the category names and percentage changes plotted for the second level specified in the input levs (NULL if there is only one level). These are, of course, conditional on the categories of the first level.
level3	the category names and percentage changes plotted for the third level specified in the input levs (NULL if there are not three levels). These are, of course, conditional on the categories of the first two levels.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

See Also

[ud_colours](#)

Examples

```
yy <- ud_prep(CPIuk, weight="Weight", v1="Aug2017", v2="Aug2018",
  levs=c("Sector", "Subsector", "Component"), sortLev=c("perc", "perc", "perc"))
y1 <- ud_plot(yy, b=0, vscale=NULL, labelvar="Sector")
y1$uad
y1$uadl
```

ud_prep

*Prepares data for drawing an UpAndDown plot***Description**

Checks input parameters and sorts the data

Usage

```
ud_prep(data, weight=1, v1, v2, levs, sortLev, reverse=c(FALSE, FALSE, FALSE))
```

Arguments

data	the dataset to be analysed
weight	a variable for weighting the individual items, the default is for every item to be weighted equally
v1	the variable specifying the start value in the comparison
v2	the variable specifying the end value in the comparison
levs	up to three grouping variables to define the plot levels in order from highest to lowest
sortLev	how the grouping variables named in levs are to be sorted. Each entry must be one of "orig", "base", "final", "perc", "abs" (i.e., no sorting, sorting by initial value, sorting by final value, sorting by percentage change, sorting by absolute change).
reverse	whether a sorting is to be reversed or not. Each entry must be TRUE or FALSE (the default) and is set separately for each level.

Details

This function prepares the data for [ud_plot](#) to draw UpAndDown plots. As well as carrying out some basic checks, any NA values in the variables v1 and v2 are converted to 0. There are a number of options for sorting the data. If a level is nested, then it is sorted within the next highest level. If a level is not nested, then it is sorted as if it were the top level.

Value

levs	the variables specifying the grouping levels for the plot
sortLev	the sorting method used for each plot level
reverse	whether individual sortings should be reversed or not
hx	a summary of the nesting of the grouping variables
lgv	the numbers of categories in the grouping variables
data	an extended version of the dataset comprising the original variables and the derived absolute and percentage changes for the different levels specified in levs. The levels are sorted according to the sortLev settings chosen and the corresponding percentages. The dataset is primarily for providing the necessary input for the Ud_plot function.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

Examples

```
yy <- ud_prep(CPIuk, weight="Weight", v1="Aug2017", v2="Aug2018",
levs=c("Sector", "Subsector", "Component"), sortLev=c("perc", "perc", "perc"))
```

UpAndDownPlots

UpAndDownPlots: display both percentage and absolute changes

Description

UpAndDown plots display percentage changes by height and absolute changes by area for up to three nested or non-nested levels. They can visualise changes in indices, showing how the changes for sectors or for individual components contribute to the overall change.

Details

An UpAndDown plot shows both percentage and absolute changes for a variable recorded at two separate time points.

Author(s)

Antony Unwin unwin@math.uni-augsburg.de

Index

* datasets

- AutoSalesGermany, [2](#)
- AutoSalesXGermany, [3](#)
- ConsumerPriceIndexUK, [4](#)
- NIpop, [6](#)

- AutoSales (AutoSalesGermany), [2](#)
- AutoSalesGermany, [2](#)
- AutoSalesX (AutoSalesXGermany), [3](#)
- AutoSalesXGermany, [3](#)

- ConsumerPriceIndexUK, [4](#)
- CPIuk (ConsumerPriceIndexUK), [4](#)

- dgroup, [5](#)

- NIpop, [6](#)

- sort5, [7](#)

- ud_colours, [8](#), [10](#)

- ud_plot, [8](#), [9](#), [11](#)

- ud_prep, [10](#), [11](#)

- UpAndDownPlots, [12](#)

- UpAndDownPlots-package
 - (UpAndDownPlots), [12](#)