Package 'Giza'

February 19, 2015

Type Package

Title Constructing panels of population pyramid plots based on lattice

Version 1.0

Date 2011-11-08

Author Erich Striessnig

Maintainer Erich Striessnig <erich.striessnig@wu.ac.at>

Description `Giza' offers a simple way of creating multiple pyramid plots in one graphics window, exploiting the power of the lattice package. It is a handy way of visualizing longitudinal grouped (i.e.: age- and education-structured) data.

Depends R (>= 2.10.0), reshape, lattice

Imports lattice

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2012-09-17 14:00:01

NeedsCompilation no

R topics documented:

Giza-package *Construct multiple population pyramids based on lattice*

Description

'Giza' offers a simple way of creating multiple pyramid plots in one graphics window, exploiting the power of the lattice package. It is a handy way of visualizing longitudinal grouped (i.e.: age- and education-structured) data.

Details

Package:	Giza
Type:	Package
Version:	1.0
Date:	2011-11-08
License:	GPL-2
LazyLoad:	yes

Most important functions: pyramidlattice, useOuterStrips2

Author(s)

Erich Striessnig

Maintainer: erich.striessnig@wu.ac.at

References

Neil Klepeis' heR.Misc package:

http://www.exposurescience.org/hosted-projects/inhalation-exposure-simulation-modeling-project/the-her-software-project

See Also

'lattice', 'heR.Misc'

Examples

```
data(EduDat)
data(dictionary)
# select the desired year, country, and education-scenario from EduDat
Years <- c(2010,2030,2050)
Countries <- c("Pakistan","Bangladesh","Indonesia")
Scenarios <- c("GET")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
```

```
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

agegrs <- paste(seq(15,100,5),seq(19,104,5),sep="-")
agegrs[length(agegrs)] <- "100+"

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                            factor(cc,levels=getcode(Countries,dictionary),labels=Countries) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Countries)*2,length(Years)),
         type="l",lwd=1,xlab="Population",ylab="Age",main="Population by Highest Level of Education",
         strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
         scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
         auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                     points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                     title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors

# compare different education-scenarios rather than countries
Countries <- c("Pakistan")
Scenarios <- c("FT","GET","CER")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                                   factor(scen2,levels=Scenarios,labels=Scenarios) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Scenarios)*2,length(Years)),
         type="l",lwd=1,xlab="Population",ylab="Age",main=paste("Population by Highest Level of Education, ",Cou
         strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
         scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
         auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                     points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                     title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors
```

dictionary *UN Country Codes*

Description

Country-names and corresponding country-codes as used by the United Nations Population Division

Usage

```
data(dictionary)
```

Format

A data frame with 228 observations on the following 2 variables.

`area` a factor with levels `Afghanistan` `AFRICA` `Albania` `...`

`ccode` a numeric vector

Source

http://esa.un.org/unpd/wpp/unpp/panel_population.htm

Examples

```
data(dictionary)
head(dictionary)

getname(156,dictionary)
getcode("China",dictionary)
```

EduDat *IIASA/VID population data by age, sex, and education*

Description

Reconstruction of populations by age, sex and highest level of educational attainment for 142 countries for 1970-2000 using demographic back-projection methods together with education forward projections for 2000-2050.

Usage

```
data(EduDat)
```

Format

A data frame with 1009152 observations on the following 7 variables.

`cc` a numeric vector containing the UN's country codes

`yr` a numeric vector indicating the year of the data point

`sex` a character vector

`agegr` a numeric vector

`scen2` a factor with levels `BP` `CEN` `CER` `FT_Singapore_TR` `GET` `UN`

. . . different population projection scenarios - see References.

`variable` a factor with levels `e1` `e2` `e3` `e4`

`value` a numeric vector

. . . the number of people in the respective country, year, by sex, age, and educational attainment level

Source

http://www.iiasa.ac.at/Research/POP/edu07/index.html

References

Lutz, W., A. Goujon, S. KC, and W. Sanderson. 2007. Reconstruction of populations by age, sex and level of educational attainment for 120 countries for 1970-2000. Vienna Yearbook of Population Research 2007:193-235.\

KC, S., B. Barakat, A. Goujon, V. Skirbekk, and W. Lutz. 2010. Projection of populations by level of educational attainment, age, and sex for 120 countries for 2005-2050. Demographic Research 22:383-472.

Examples

```
library(reshape)

data(EduDat)
data(dictionary)

icountry <- "China"
ipop <- subset(EduDat,cc==getcode(icountry,dictionary) & scen2 != "BP")
ipop <- cast(ipop,yr~scen2,sum)
row.names(ipop) <- ipop[,1]
matplot(as.data.frame(ipop[,-1]),type="l",lwd=3,main=paste("Projected Population 15+ by Education Scenario, ",ic
      xlab="Year",ylab="Total Population in 1000s",xaxt="n",cex=1.5,cex.axis=1.5,cex.lab=1.5)
axis(1,at=1:nrow(ipop),labels=row.names(ipop),cex.axis=1.5)
grid(lwd=2,lty=2)
legend(legend=names(ipop[,-1]),title="Education Scenario: ",
       col=1:ncol(ipop[,-1]),lwd=3,lty=1:ncol(ipop[,-1]),
       "topleft",inset=c(0.02,0.02),bty="o",bg="grey",cex=1.5)
```

getcode *getcode Function*

Description

Extract the UN country code for a given country

Usage

```
getcode(cname, dictionary)
```

Arguments

cname	... any country name in dictionary
dictionary	... see data(dictionary)

Details

"inverse" of getname function

Value

the UN country-code for the country in question

Author(s)

Erich Striessnig <erich.striessnig@wu.ac.at>

See Also

getname

Examples

```
data(dictionary)
getcode("China",dictionary)
```

getname *getname Function*

Description

Extract the UN country name for a given country code

Usage

```
getname(ccode, dictionary)
```

Arguments

ccode	any country code in dictionary
dictionary	see ?dictionary

Details

This is the "inverse" of the 'getcode' function.

Value

the name of the country corresponding to the ccode value in question

Author(s)

Erich Striessnig <erich.striessnig@wu.ac.at>

See Also

'getcode'

Examples

```
data(dictionary)
getname(156,dictionary)
```

panel.pyramid *Default Panel Function for 'pyramidlattice'*

Description

Default panel function for 'pyramidlattice'. Essentially the same as 'panel.barchart' in lattice package, but defaults to stacked bars. See ?panel.barchart in lattice-package.

See Also

'barchart'

prepanel.default.bwplot2 *Default Prepanel Function for 'pyramidlattice'*

Description

Default prepanel function for 'pyramidlattice'. Essentially the same as 'prepanel.default.bwplot' in lattice package, but defaults to stacked bars.

Author(s)

Erich Striessnig <erich.striessnig@wu.ac.at>, modified from package lattice.

pyramid2 *High level lattice function producing population pyramids.*

Description

This is the function used inside 'pyramidlattice' to produce stacked, horizontal barcharts for grouped data with varying x-scale limits to allow for distinction of i.e. male/female data points in population pyramid graphs. This is essentially a modification of 'barchart2' in package heR.Misc (see References). For details, see ?barchart2 (heR.Misc) and ?xyplot (lattice).

Author(s)

Erich Striessnig, adapted from the 'barchart2' function included in Neil Klepeis' heR.Misc package.

References

heR.Misc package source:

http://www.exposurescience.org/heR.doc/library/heR.Misc/html/barchart2.html

Examples

```
data(EduDat)
data(dictionary)
# select the desired year, country, and education-scenario from EduDat
Years <- c(2010,2030,2050)
Countries <- c("Pakistan","Bangladesh","Indonesia")
Scenarios <- c("GET")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

agegrs <- paste(seq(15,100,5),seq(19,104,5),sep="-")
```

```
agegrs[length(agegrs)] <- "100+"

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                            factor(cc,levels=getcode(Countries,dictionary),labels=Countries) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Countries)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main="Population by Highest Level of Education",
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                      points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                      title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors

# compare different education-scenarios rather than countries
Countries <- c("Pakistan")
Scenarios <- c("FT","GET","CER")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                                   factor(scen2,levels=Scenarios,labels=Scenarios) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Scenarios)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main=paste("Population by Highest Level of Education, ",Cou
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                      points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                      title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors
```

`pyramid2.formula` *High level lattice function used inside 'pyramidlattice' to produce population pyramids*

Description

Produce stacked, horizontal barcharts for grouped data with varying x-scale limits to allow for distinction of i.e. male/female data points in population pyramid graphs. This is essentially a modification of the barchart2 function in Neil Klepeis' package heR.Misc (see References).

Details

See the examples below and the help file for ?xyplot in the lattice-package for further details.

Author(s)

Erich Striessnig, adapted from the 'barchart2' function in Neil Klepeis' heR.Misc package.

References

heR.Misc package source:

http://exposurescience.org/?q=hosted-projects/human-exposure-research-software-package http://www.exposurescience.org/

Examples

```
data(EduDat)
data(dictionary)
# select the desired year, country, and education-scenario from EduDat
Years <- c(2010,2030,2050)
Countries <- c("Pakistan","Bangladesh","Indonesia")
Scenarios <- c("GET")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

agegrs <- paste(seq(15,100,5),seq(19,104,5),sep="-")
agegrs[length(agegrs)] <- "100+"

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                            factor(cc,levels=getcode(Countries,dictionary),labels=Countries) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Countries)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main="Population by Highest Level of Education",
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                      points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                      title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
```

```
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors

# compare different education-scenarios rather than countries
Countries <- c("Pakistan")
Scenarios <- c("FT","GET","CER")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                                   factor(scen2,levels=Scenarios,labels=Scenarios) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Scenarios)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main=paste("Population by Highest Level of Education, ",Cou
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                    points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                    title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors
```

---

pyramidlattice *High level lattice-function producing population pyramids*

Description

This is the function used to produce stacked, horizontal barcharts for grouped data with varying x-scale limits to allow for distinction of i.e. male/female data points in population pyramid graphs. This is essentially a modification of barchart2 in package heR.Misc (see References).

By calculating the x-scale limits properly and by allowing them to vary for male- and female-panels, 'pyramidlattice' produces population pyramids to compare population age-structures of countries in different years, as well as different scenarios for one and the same country in different years.

In addition to that, useOuterStrips2() provides a possibility to draw parsimonious outer-strips for three factor variables which helps to save space in the plot window.

Note that the values for the 'male'-column have to be flipped by hand before passing the data to 'pyramidlattice'.

See examples below and ?barchart2 (heR.Misc) as well as ?xyplot (lattice) for details.

Author(s)

Erich Striessnig, adapted from the 'barchart2' function included in Neil Klepeis' heR.Misc package.

References

heR.Misc package source:

http://www.exposurescience.org/heR.doc/library/heR.Misc/html/barchart2.html

See Also

'barchart' is the original lattice function for plotting (univariate) barcharts in each panel.

'barchart2' from heR.Misc package is Neil Klepeis' modification of barchart, dealing with grouped data by plotting stacked or side-by-side bars just like the bar plotting function in base R graphics.

'panel.pyramid' is the panel function used to plot group data as stacked bars.

'prepanel.default.bwplot2' is the prepanel function used to specify default (relation = same) horizontal and vertical limits for each panel for stacked bars. Yet, in order to get the reverse limits for the male and the female panels, the limits have to be passed on to pyramidlattice using the limits-argument in scale (see ?xyplot).

Examples

```
data(EduDat)
data(dictionary)
# select the desired year, country, and education-scenario from EduDat
Years <- c(2010,2030,2050)
Countries <- c("Pakistan","Bangladesh","Indonesia")
Scenarios <- c("GET")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

agegrs <- paste(seq(15,100,5),seq(19,104,5),sep="-")
agegrs[length(agegrs)] <- "100+"

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                            factor(cc,levels=getcode(Countries,dictionary),labels=Countries) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Countries)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main="Population by Highest Level of Education",
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                      points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
```

```
                  title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
            prepanel=prepanel.default.bwplot2,panel=function(...){
                     panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                     panel.pyramid(...)
                    })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors

# compare different education-scenarios rather than countries
Countries <- c("Pakistan")
Scenarios <- c("FT","GET","CER")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                                   factor(scen2,levels=Scenarios,labels=Scenarios) *
                                   factor(yr,levels=Years,labels=Years),
      groups=variable,data=iEduDat,layout=c(length(Scenarios)*2,length(Years)),
      type="l",lwd=1,xlab="Population",ylab="Age",main=paste("Population by Highest Level of Education, ",Cou
      strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
      scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
      auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                    points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                    title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
      prepanel=prepanel.default.bwplot2,panel=function(...){
                     panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                     panel.pyramid(...)
                    })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors
```

`useOuterStrips2` *Use outer strips in a lattice display instead of multiple inner strips*

Description

useOuterStrips2 helps to save space in a lattice display by getting rid of the strips above those panels inside the trellis figure while keeping the strips on top of the entire figure. It is a modification of useOuterStrips in the latticeExtra package, but it is able to deal with three factors instead of two.

Used in combination with 'pyramidlattice'.

Usage

```
useOuterStrips2(x, strip.lines = 1, strip.left.lines = 1)
```

Arguments

x	An object of class 'trellis'.
strip.lines	height of strips in number of lines.
strip.left.lines	height of left strip in number of lines.

Details

See useOuterStrips in package latticeExtra.

Value

An object of class 'trellis'; essentially the same as 'x', but with strips modified.

Author(s)

Erich Striessnig, adapted from useOuterStrips in latticeExtra by Deepayan Sarkar.

See Also

pyramidlattice, latticeExtra, useOuterStrips

Examples

```
data(EduDat)
data(dictionary)
# select the desired year, country, and education-scenario from EduDat
Years <- c(2010,2030,2050)
Countries <- c("Pakistan","Bangladesh","Indonesia")
Scenarios <- c("GET")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

agegrs <- paste(seq(15,100,5),seq(19,104,5),sep="-")
agegrs[length(agegrs)] <- "100+"

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                            factor(cc,levels=getcode(Countries,dictionary),labels=Countries) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Countries)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main="Population by Highest Level of Education",
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                      points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                      title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
```

```
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors

# compare different education-scenarios rather than countries
Countries <- c("Pakistan")
Scenarios <- c("FT","GET","CER")
# the male-column needs to be flipped
iEduDat <- subset(EduDat,match(cc,getcode(Countries,dictionary)) & match(yr,Years) & match(scen2,Scenarios))
iEduDat$value[iEduDat$sex == "Male"] <- (-1) * iEduDat$value[iEduDat$sex == "Male"]

lattice.options(axis.padding = list(numeric=0))
x <- pyramidlattice(agegr ~ value| factor(sex,levels=c("Male","Female")) *
                                   factor(scen2,levels=Scenarios,labels=Scenarios) *
                                   factor(yr,levels=Years,labels=Years),
          groups=variable,data=iEduDat,layout=c(length(Scenarios)*2,length(Years)),
        type="l",lwd=1,xlab="Population",ylab="Age",main=paste("Population by Highest Level of Education, ",Cou
        strip=TRUE,par.settings = simpleTheme(lwd=3,col=colors()[c(35,76,613,28)]),box.width=1,
        scales=list(alternating=3,tick.number=5,relation="same",y=list(at=1:length(4:21),labels=agegrs)),
        auto.key=list(text=c("No-edu","Primary","Secondary","Tertiary"),reverse.row=TRUE,
                    points=FALSE,rectangles=TRUE,space="right",columns=1,border=FALSE,
                    title="ED-Level",cex.title=1.1,lines.title=2.5,padding.text=1,background="white"),
          prepanel=prepanel.default.bwplot2,panel=function(...){
                    panel.grid(h=length(agegrs),v=5,col="lightgrey",lty=3)
                    panel.pyramid(...)
                   })

x # with strips for every factor over each panel
# useOuterStrips(x) # with outer strips, but only in case of two factors
useOuterStrips2(x) # with outer strips in case of three factors
```

Index