Package 'ggESDA'

August 31, 2021

Type Package

Title Visualizing symbolic data (interval data) with ggplot2

Version 0.1.0

Author Bo-Syue Jiang

Maintainer Bo-Syue Jiang <lucky1515699@gmail.com>

Description This package implements an extension of ggplot2 and visualizes the symbolic data with multiple plot which can be adjusted by more general and flexible input arguments.It also provides a function to transform the classical data to symbolic data by both clustering algorithm and customized method.

Depends ggplot2,
tidyverse

Suggests testthat (>= 2.1.0),
knitr,
rmarkdown

Imports RSDA,
utils,
data.table,
rlang,
R6,
stats,
dplyr,
grid,
gridExtra,
gtools,
stringr,
tibble,
prodlim,
ggforce

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

R topics documented:

`classic2sym` *Convert classical data frame into a symbolic data.*

Description

A function for converting a classical data,which may present as a data frame or a matrix with one entry one value,into a symbolic data,which is shown as a interval or a set in an entry. Object after converting is ggESDA class containing interval data and raw data(if it exist) and typically statistics.

Usage

```
classic2sym(data=NULL,groupby = "kmeans",k=5,minData=NULL,maxData=NULL)
```

Arguments

`data`	A classical data frame that you want to be converted into a interval data
`groupby`	A way to aggregate.It can be either a clustering method or a variable name which exist in input data (necessary factor type) .Default "kmeans".
`k`	A number of group,which is used by clustering.Default k=5.
`minData`	if choose groupby parameter as 'customize',user need to define which data is min data or max data.
`maxData`	if choose groupby parameter as 'customize',user need to define which data is min data or max data.

Value

classic2sym returns an object of class "ggESDA",which have a interval data and others as follows.

- intervalData - The Interval data after converting also known as a RSDA object.
- rawData - Classical data that user input.
- clusterResult - Cluster results .If the groupby method is a clustering method then it will exist.
- statisticsDF - A list contains data frame including some typically statistics in each group.

Examples

```
#classical data to symbolic data
classic2sym(iris)
classic2sym(mtcars,groupby = "kmeans",k=10)
classic2sym(iris, groupby = "hclust",k=7)
classic2sym(iris,groupby=Species)

x1<-runif(10,-30,-10)
y1<-runif(10,-10,30)
x2<-runif(10,-5,5)
y2<-runif(10,10,50)
x3<-runif(10,-50,30)
y3<-runif(10,31,60)

d<-data.frame(min1=x1,max1=y1,min2=x2,max2=y2,min3=x3,max3=y3)
classic2sym(d,groupby="customize",minData=d[,c(1,3,5)],maxData=d[,c(2,4,6)])
classic2sym(d,groupby="customize",minData=d$min1,maxData=d$min2)

#extract the data
symObj<-classic2sym(iris)
symObj$intervalData       #interval data
symObj$rawData            #raw data
symObj$clusterResult      #cluster result
symObj$statisticsDF       #statistics
```

---

ggESDA *A symbolic object by R6 class for interval analysis and ggplot*

Description

This is an object that will be used to make a ggplot object.A ggESDA object contains both classic data that user have and interval data which we transform.More over,some basic statistics from row data will also be recorded in this object,and the interval data which is from RSDA transformation will still contain RSDA properties.

Public fields

`rawData` the data from user.

`statisticsDF` contains min max mean median dataframe for each group of symbolic data

`intervalData` interval data from RSDA type

`clusterResult` clustering result

Methods

Public methods:

- `ggESDA$new()`
- `ggESDA$clone()`

Method `new()`**:** initialize all data, check whether satisfy theirs form

Usage:

```
ggESDA$new(
  rawData = NULL,
  statisticsDF = NULL,
  intervalData = NULL,
  clusterResult = NULL
)
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
ggESDA$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`ggInterval_2Dhist` *Plot a 2-dimension histogram by symbolic data with ggplot package.*

Description

Visualize the two continuous variable distribution by dividing both the x axis and y axis into bins,and calculating the frequency of observation interval in each bin.

Usage

```
ggInterval_2Dhist(data = NULL,mapping = aes(NULL)
,xBins = 14,yBins=16)
```

Arguments

`data`	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
`mapping`	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.
`xBins`	x axis bins,which mean how many partials x variable will be separate into.
`yBins`	y axis bins.It is the same as xBins.

Value

Return a ggplot2 object.

Examples

```
#a classical data input
ggInterval_2Dhist(mtcars,aes(x=disp,y=wt))
ggInterval_2Dhist(iris,aes(x=iris$Sepal.Length,y=iris[,3]),xBins=30)
ggInterval_2Dhist(mtcars,aes(disp,wt),xBins=23,yBins=35)

#you can add and aesthetic like colour and alpha
p<-ggInterval_2Dhist(mtcars,aes(x=disp,y=wt,col="black",alpha=0.8))
```

```
p
#adjust fill manual you like
p+scale_fill_gradient2(low = "green",mid="grey",high = "red",
    midpoint = (max(p$data$freq)+min(p$data$freq))/2)

#a symbolic data input (ex.RSDA dataset:Cardiological)
#it can also be produce by classic2sym
mydata<-RSDA::Cardiological #generate symbolic data
ggInterval_2Dhist(mydata,aes(Pulse,Syst))
p<-ggInterval_2Dhist(mydata,aes(Pulse,Syst,col="red",lty=2))
p
p+theme_classic()+labs(title="My 2d plot")
p+scale_x_continuous(breaks=c(50,60,80,100),
    labels=c("aa","bb","cc","dd"))+geom_point(aes(x=60,y=150),pch=16,size=3)

#mark a particular interval
p+geom_rect(data=p$data,aes(xmin=p$data[10,"x1"],xmax=p$data[10,"x2"],
    ymin=p$data[10,"y1"],ymax=p$data[10,"y2"]),fill="red")
```

`ggInterval_2DhistMatrix`

2-Dimension histogram matrix

Description

Visualize the all continuous variable distribution by dividing both the x axis and y axis into bins,and calculating the frequency of observation interval in each bin.Eventually show it by a matrix plot.Note:this function will automatically filter out the discrete variables,and plot all continuous in input data,so it can not be necessary that give the particularly variables in aes such like (aes(x=x,y=y)).It isn't also recommended to deal with too many variables because the big O in calculating full matrix will be too large.

Usage

```
ggInterval_2DhistMatrix(data = NULL,mapping = aes(NULL)
,xBins = 14,yBins=16)
```

Arguments

`data`	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
`mapping`	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.
`xBins`	x axis bins,which mean how many bins x variable will be separate into
`yBins`	y axis bins.It is the same as xBins

Value

Return a plot with no longer a ggplot2 object,instead of a marrangeGrob object.

Examples

```
ggInterval_2DhistMatrix(iris,aes(col="black",alpha=0.8))

mydata<-RSDA::Cardiological
ggInterval_2DhistMatrix(mydata)
```

---

ggInterval_3Dscatter *3D scatter plot for interval data*

Description

Visualize the three continuous variable distribution by collecting all vertices in each interval to form a shape of cube.Also show the difference between each group.

Usage

```
ggInterval_3Dscatter(data = NULL,mapping = aes(NULL),scale=FALSE)
```

Arguments

data	A ggSDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggSDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.
scale	A boolean variable,TRUE,standardlize data.FALSE,not standardlize. If variance is too large(or small) or the difference between two variables are too large,it will be distortion or unseeable,which may happen in different units or others.So,a standardlize way is necessary.

Value

Return a ggplot2 object (It will still be 2-Dimension).

Examples

```
p<-ggInterval_3Dscatter(iris,aes(iris$Sepal.Length,iris$Sepal.Width,iris$Petal.Length))
p
p+scale_fill_manual(values = c("red","yellow",
    "green","blue",
    "black"),
    labels=c("group1","group2",
            "group3","group4",
            "group5"),
    name="my fill")

#generate symbolic data
mySymData<-classic2sym(iris,Species)
```

```
mySymData<-mySymData$intervalData
p<-ggInterval_3Dscatter(mySymData,aes(Sepal.Length,
    Sepal.Width,
    Petal.Length,col="red",lty=2))
p
p+scale_fill_manual(values = c("yellow",
    "green",
    "black"),
    labels=rownames(mySymData))
```

---

ggInterval_boxplot *A interval Box plot*

Description

Visualize the one continuous variable distribution by box represented by multiple rectangles.

Usage

```
ggInterval_boxplot(data = NULL,mapping = aes(NULL))
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.

Value

Return a ggplot2 object.

Examples

```
p<-ggInterval_boxplot(iris,aes(iris$Petal.Length))
p
p+scale_fill_manual(values = c("red","yellow",
    "green","blue","black"),
    labels=c("0%","25%","50%","75%","100%"),
    name="quantile")

mydata<-RSDA::facedata
ggInterval_boxplot(mydata,aes(AD,col="black",alpha=0.5))

myMtcars<-classic2sym(mtcars)
myMtcars<-myMtcars$intervalData
ggInterval_boxplot(myMtcars,aes(disp))
```

ggInterval_centerRange *Figure with x-axis=center y-axis=range*

Description

Visualize the relation between center and range.

Usage

```
ggInterval_centerRange(data = NULL,mapping = aes(NULL))
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.

Value

Return a ggplot2 object.

Examples

```
ggInterval_centerRange(iris,aes(iris$Sepal.Length))

mydata<-RSDA::facedata
ggInterval_centerRange(mydata,aes(AD,col="blue",pch=2))
```

ggInterval_hist *Histogram for symbolic data with equal-bin or inequal-bin.*

Description

Visualize the continuous variable distribution by dividing the x axis into bins,and calculating the frequency of observation interval in each bin.

Usage

```
ggInterval_hist(data = NULL,mapping = aes(NULL),method="equal-bin",bins=10)
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.
method	It can be equal-bin(default) or inequal-bin.Enqual-bin means the width in histogram is equal,which represent all intervals divided have the same range.Inequal-bin means the range of intervals are not the same,and it can be more general on data.Thus,the bins of inequal-bin method depends on the data,and the argument "bins" will be unused.
bins	x axis bins,which mean how many partials the variable will be separate into.

Value

Return a ggplot2 object.

Examples

```
ggInterval_hist(mtcars,aes(x=wt))

ggInterval_hist(iris,aes(iris$Petal.Length,col="blue",alpha=0.2,
   fill="red"),bins=30)


d<-data.frame(x=rnorm(1000,0,1))
p<-ggInterval_hist(d,aes(x=x),bins=40,method="equal-bin")
p

p+scale_fill_manual(values=rainbow(40))+labs(title="myNorm")
```

---

ggInterval_index *Plot the range of each observations*

Description

Visualize the range of the variables of each observations by using a kind of margin bar that indicate the minimal and maximal of observations.

Usage

```
ggInterval_index(data = NULL,mapping = aes(NULL))
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.

Value

Return a ggplot2 object.

Examples

```
#the observations show on the y-axis .values on x-axis
ggInterval_index(iris,aes(x=iris$Sepal.Length))

#change above axis
ggInterval_index(mtcars,aes(y=disp,col="red",fill="grey"))

#symbolic data
mydata <- RSDA::facedata
ggInterval_index(mydata,aes(x=3:13,y=AD))
```

---

ggInterval_indexImage *An index plot presented by color image for interval data.*

Description

Visualize the range of the variables of each observations by using color image.The index image replace margin bar by color,thus it will be more visible for data.

Usage

```
ggInterval_indexImage(data = NULL,mapping = aes(NULL),
column_condition=TRUE,full_strip=FALSE)
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
column_condition	Boolean variables,which mean the color present by column condition (if TRUE) or matrix condition (if FALSE)
full_strip	Boolean variables,which mean the strip present in full figure-width (if TRUE) or only in its variable values(if FALSE).

Value

Return a ggplot2 object.

Examples

```
d<-data.frame(qq=rnorm(1000,0,1))
ggInterval_indexImage(d,aes(qq))

mydata<-RSDA::facedata
p<-ggInterval_indexImage(mydata,aes(AD),full_strip=TRUE,column_condition = TRUE)
#Recommend to add coord_flip() to make the plot more visible
p+coord_flip()

myIris<-classic2sym(iris,groupby=Species)
myIris<-myIris$intervalData
p<-ggInterval_indexImage(myIris,aes(myIris$Petal.Length),full_strip=FALSE,column_condition=TRUE)
p

ggInterval_indexImage(mtcars,aes(disp))+labs(x="anything")
```

---

ggInterval_minmax *A min-max plot for interval data*

Description

Visualize the range of the variables of each observations by marking minimal and maximal point.

Usage

```
ggInterval_minmax(data = NULL,mapping = aes(NULL),sort=TRUE)
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
sort	if FALSE, it will not be sort by min data,default TRUE.

Value

Return a ggplot2 object.

Examples

```
ggInterval_minmax(mtcars,aes(disp))

mydata2<-RSDA::Cardiological
ggInterval_minmax(mydata2,aes(mydata2$Pulse,size=3))

d<-mapply(c(10,20,40,80,160),c(20,40,80,160,320),FUN=runif,n=1000)
d<-data.frame(qq=matrix(d,ncol=1))
ggInterval_minmax(d,aes(qq))
```

```
myIris<-classic2sym(iris,groupby=Species)
myIris<-myIris$intervalData
ggInterval_minmax(myIris,aes(myIris$Petal.Length))+
   theme_classic()
```

`ggInterval_PCA` *Vertice-PCA for interval data*

Description

ggInterval_PCA performs a principal components analysis on the given numeric interval data and returns the results like princomp , ggplot object and a interval scores.

Usage

```
ggInterval_PCA(data = NULL,mapping = aes(NULL),plot=TRUE)
```

Arguments

`data`	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
`mapping`	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping. It is the same as the mapping of ggplot2.
`plot`	Boolean variable,Auto plot (if TRUE).It can also plot by its inner object

Value

A ggplot object for PC1,PC2,and a interval scores and others.

- scores_interval - The interval scores after PCA.
- ggplotPCA - a ggplot object with x-axis and y-axis are PC1 and PC2.
- others - others are the returns values of princomp.

Examples

```
ggInterval_PCA(iris)

mydata2<-RSDA::Cardiological
ggInterval_PCA(mydata2,aes(col="red",alpha=0.2))

d<-mapply(c(10,20,40,80,160),c(20,40,80,160,320),FUN=runif,n=1000)
d<-data.frame(qq=matrix(d,ncol=4))
ggInterval_PCA(d)

myIris<-classic2sym(iris,Species)
p<-ggInterval_PCA(myIris,plot=FALSE)
p$ggplotPCA
p$scores_interval
```

ggInterval_radar *A interval Radar plot*

Description

Using ggplot2 package to make a radar plot with multiple variables.Each variables contains min values and max values as a symbolic data.

Usage

```
ggInterval_radar(data=NULL,layerNumber=4,
inOneFig=FALSE,showLegend=TRUE,showXYLabs=FALSE,
plotPartial=NULL,fillBetween=TRUE)
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame(not recommended),which will be automatically convert to ggESDA data.
layerNumber	number of layer of a concentric circle,usually to visuallize the reach of a observation in particularly variable.
inOneFig	whether plot all observations in one figure.if not, it will generate a new windows containing distinct observations.
showLegend	whether show the legend.
showXYLabs	whether show the x,y axis labels.
plotPartial	a numeric vector,which is the row index from the data.if it is not null,it will extract the row user deciding to draw a radar plot from original data.Notes : the data must be an interval data if the plotPartial is not null.
fillBetween	default TRUE,it will fill color between interval. Else,it will draw two radar plot to show min value and max value.

Examples

```
mydata<-ggESDA::classic2sym(mtcars,k=4)$intervalData
ggInterval_radar(data=mydata[,c("mpg","disp",'drat')])
ggInterval_radar(data=mydata[,c("mpg","disp",'drat')],inOneFig = TRUE,plotPartial = c(2,3))


mydata<-ggESDA::classic2sym(iris,groupby = Species)$intervalData
ggInterval_radar(mydata,inOneFig = TRUE)+geom_text(aes(x=0.6,0.6),label="Add anything you want")
```

`ggInterval_scaMatrix` *scatter plot for all variable by interval data.*

Description

Visualize the all continuous variable distribution by rectangle for both x-axis and y-axis with a matrix grid. Note:this function will automatically filter out the discrete variables,and plot all continuous in input data,so it can not be necessary that give the particularly variables in aes such like (aes(x=x,y=y)).It isn't also recommended to deal with too many variables because the big O in calculating full matrix will be too large.

Usage

```
ggInterval_scaMatrix(data = NULL,mapping = aes(NULL))
```

Arguments

`data`	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
`mapping`	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

Value

Return a plot with no longer a ggplot2 object,instead of a marrangeGrob object.

Examples

```
a<-rnorm(1000,0,5)
b<-runif(1000,-20,-10)
c<-rgamma(1000,10,5)
d<-as.data.frame(cbind(norm=a,unif=b,gamma_10_5=c))
ggInterval_scaMatrix(d)


ggInterval_scaMatrix(mtcars[,c("mpg","wt","qsec")],
    aes(col="red",lty=2,fill="blue",alpha=0.3))


myIris <- classic2sym(iris,groupby = Species)$intervalData
ggInterval_scaMatrix(myIris[,1:3])


mydata <- RSDA::Cardiological
ggInterval_scaMatrix(mydata[,1:3],aes(fill="black",alpha=0.2))
```

ggInterval_scatter *scatter plot for two continuous interval data*

Description

Visualize the twwo continuous variable distribution by rectangle and each of its width and heigth represents a interval of the data.

Usage

```
ggInterval_scatter(data = NULL,mapping = aes(NULL))
```

Arguments

data	A ggESDA object.It can also be either RSDA object or classical data frame,which will be automatically convert to ggESDA data.
mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

Value

Return a ggplot2 object.

Examples

```
a<-rnorm(1000,0,5)
b<-runif(1000,-20,-10)
d<-as.data.frame(cbind(norm=a,unif=b))
ggInterval_scatter(d,aes(a,b))


ggInterval_scatter(mtcars[,c("mpg","wt","qsec")],
    aes(x=mpg,y=wt,
    col="red",lty=2,fill="blue",alpha=0.3))


myIris <- classic2sym(iris,groupby = Species)$intervalData
p<-ggInterval_scatter(myIris,aes(myIris$Petal.Length,myIris$Petal.Width))
p
p+scale_fill_manual(labels=rownames(myIris),
                    values=c("red","blue","green"),
                    name="Group")


mydata <- RSDA::facedata
p<-ggInterval_scatter(mydata[1:10,],aes(AD,BC,alpha=0.2))
p+scale_fill_manual(labels=rownames(mydata)[1:10],
                    values=rainbow(10),
                    name="Group")
```

RSDA2sym *RSDA object to symbolic object for ggplot*

Description

It will be a good way to unify all symbolic data object in R that collects all useful symbolic analysis tools such like RSDA into the same class for management.In this way, user who wants to do some study in symbolic data will be more convenient for searching packages.Thus,RSDA2sym collecting RSDA object into ggESDA object will do for plot(ggplot) and RSDA's analysis.

Usage

```
RSDA2sym(data=NULL,rawData=NULL)
```

Arguments

data	an interval data ,which may transfrom by RSDA::classic.to.sym .Note:data is a necessary parameter,and must have symbolic_tbl class.
rawData	rawData,which can be transformed to interval data, must be a data frame and match to data.

Value

Return an object of class "ggESDA",which have a interval data and others as follows.

- intervalData - The Interval data after converting also known as a RSDA object.
- rawData - Classical data that user input.
- clusterResult - Cluster results .If the groupby method is a clustering method then it will exist.
- statisticsDF - A list contains data frame including some typically statistics in each group.

#'

Examples

```
r<-RSDA::Cardiological
mySym<-RSDA2sym(r)
mySym$intervalData
```

Index