Journal of Statistical Software
MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

ggESDA: An R Package for Exploratory Symbolic Data Analysis using ggplot2

Bo-Syue Jiang
National Taipei University

Han-Ming Wu
National Chengchi University

Abstract

This paper presents the **ggESDA** package, which we designed for exploratory symbolic data analysis in R. The **ggESDA** package developed a general and customized transformation function `classic2sym` implemented for generating a symbolic data table from classical data frame by clustering algorithm using the **RSDA** package Rodriguez (2021) form and generalized **ggESDA** to other prominent SDA packages in R. Based on **ggplot2** Wickham (2016), **ggESDA** package which is familiar programming structure with its parent provides a wide variety of graphical techniques such as histogram, center-range plot and radar plot, etc. With the advanced graphic techniques, the multivariate visualization, exhibition with modal multi-valued and interval-valued, and classification by color mapping are combined for exploration of symbolic data in the present paper.

Keywords: data visualization, symbolic data analysis, exploratory data analysis, **ggplot2** extensions, interval-valued data, R.

1. Introduction

As stated by Diday and Edwin (2018) "In Data Science the aim is to extract new knowledge from Standard, Big, and complex data. Often these data are unstructured with variables defined on different kinds of units. They can also be multi-sources (as mixtures of numerical and textual data, with images and networks)." It indicates that not only conventional data but the unstructured data, also known as symbolic data, is vital for data science. Rather than the classical data represented by a single value, symbolic data with measurements on p random variables can be p-dimensional statistical units such as hypercubes or histograms. The field of symbolic data analysis (SDA) Billard and Diday (2007) is to broaden the application aspects of statistical methodologies, extend traditional cognition of a form of data unit and build a brand-new analysis system of data science. Recent developments in the field of big data analytics have led to a renewed interest in complex structure data such as symbolic data. As shown in Figure 1, the number of researches in SDA represents an increasing trend from 1998 to 2020, which outstands the importance of it during the years.

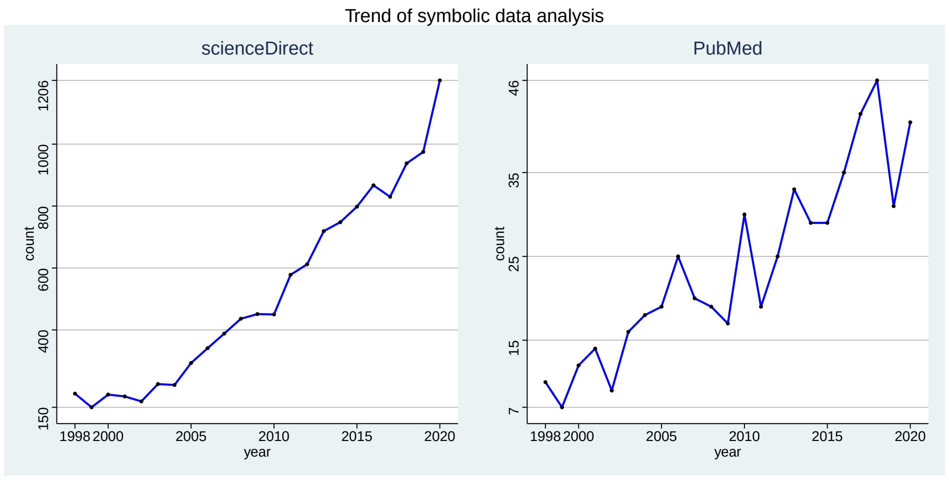


Figure 1: The number of "symbolic data analysis" or "interval-valued data" related articles in researches and applications according to PubMed and ScienceDirect online database over time from 1998 to 2020.

Among ScienceDirect, Engineering and Computer Science lead the subject areas obviously, shown in Figure 2.

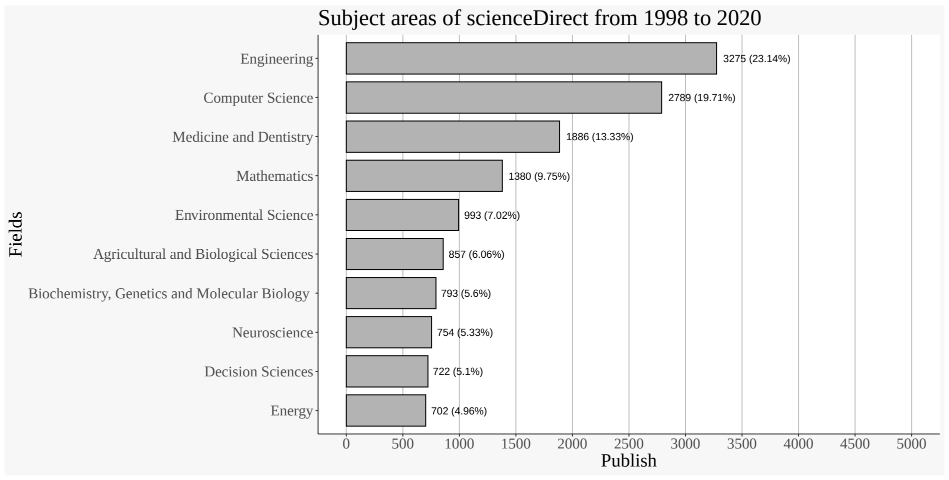


Figure 2: Top 10 researches and applications domains for SDA or interval-valued data (ScienceDirect) from 1998 to 2020

In practice, the symbolic data is often generated by aggregating massive datasets into intervals in order to make the management easy and appropriate. An interval-valued symbolic random variable X, taking values in interval, can be denoted such as $X = [a, b] \subset R^1$, where $a \leq b$, and $a, b \in R^1$. Let the random variable X, for instance, be the weight, then $X = [50, 100]$ represents the interval covering the weight of people. With the advent of big data analytic, interval-valued data is becoming more common and accessible than ever. The researches for interval-valued data such as the sign test for COVID-19 data Sherwani, Shakeel, Saleem, Awan, Aslam, and Farooq (2021), the prediction via regularized artificial neural network Yang, Lin, and Zhang (2019), a bivariate Bayesian method for regression models Xu and Qin (2021), etc.

Exploratory Data Analysis (EDA) Tukey (1977) is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task, provides an overview of raw datasets and obtains a general understanding about the variables and their relationships. In this paper, we describe the design and features of an exploratory symbolic data analysis package, **ggESDA**, which is composed of the interval-valued data, statistics data frame, clustering results, and other components from **R6** Chang (2021) class. Based on it, the developed graphical technology can extend three aspects by its variables, univariate, bivariate, and multivariate. The **ggESDA** aims to convert the traditional data into the **ggESDA** object and visualizes the symbolic data using **ggplot2**, which is shown in Figure 3. Each row (observation) in the conventional data matrix $X_{p\times p}$ in the figure contains a vector of numeric values, $O_i = (x_1, x_2, \cdots, x_n)$, while each row of the interval-valued data matrix $I_{k\times p}$ contains a vector of intervals (ranges), $C_j = ([a_{j1}, b_{j1}], [a_{j2}, b_{j2}], \cdots, [a_{jp}, b_{jp}])$, called a CONCEPT (or UNIT). The CONCEPT describe the behavior of a group of observations. Thus, the aggregation method between them is an essential process in SDA. The **ggESDA** is now available from the Comprehensive R Archive Network (CRAN) at `https://cran.r-project.org/web/packages/ggESDA/index.html`. All reference manual documented by exported functions and introduction vignettes can also be download here. In the following section, we are going to illustrate the details of SDA and the functionalities and syntaxes about **ggESDA**.

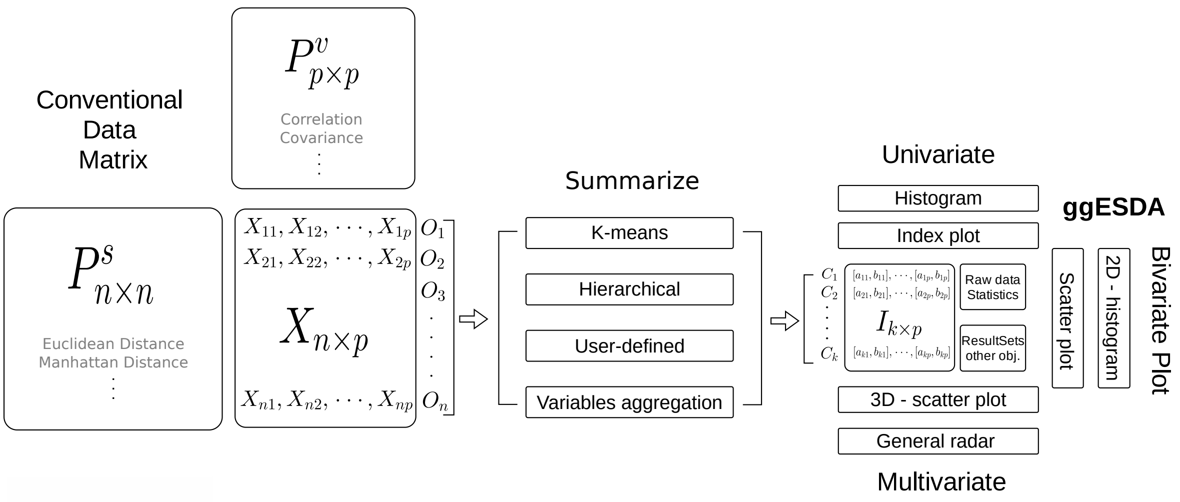


Figure 3: Package Structure and Diagram for the Transformation Flow

2. Advantage of Symbolic Data Analysis

Modern statistical graphical methods for visualizing big data nowadays we encounter large data sets in many areas, including genomics, finance records, environmental research, and internet logs. In these cases, the size of data is getting larger and the structure is more complicated. Aggregation of these huge (volume) of data into symbolic objects such as histograms or distributions is sometimes necessary. As a consequence, the following will discuss the benefit of the symbolic data after aggregating.

2.1. Convenience of Exploration

For the conventional exploratory data analysis, it is always a severe challenge to deal with enormous datasets because conventional displays suffered from overstrikes of data points representing the value (scatterplot type displays) or overstrikes of line segments connecting values of neighboring variables. As a consequence, exploratory symbolic data analysis (SDA) becomes a preliminary yet essential tool for summarizing the main characteristics of a data set before appropriate statistical modeling can be applied. Besides escaping the problem mentioned above, SDA can effectively reduce observations in data, which will make the study focus on what we interesting instead of unnecessary information such as Figure 4.

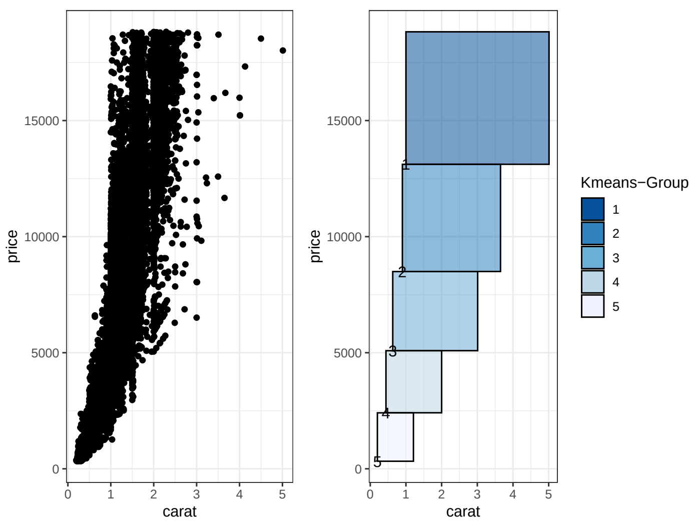


Figure 4: Compare classical data and symbolic data

In Figure 4, we can clearly visualize the scatter plot in the right hands, which is represented by symbolic data and aggregated by K-means MacQueen *et al.* (1967), and effectively reduce the observations from 53940 to 5 groups.

2.2. Completeness of Information

To lead researchers to explore more details in what they are interesting such as a particular part of data, aggregation methods play a vital role to merge the data we interesting. However, the conventional data after merging will usually be represented by a single value (especially mode or mean), which will be unmeaningful to visualize and cause the loss of information that may become larger when the data or the number of factors in that category is growing on. For the category data, SDA will build a histogram by calculating each factor of the category of frequency as bins to solve this kind of problem as a result. In the way, a categorical variable will never be shown as a single value at all, instead, a complete information histogram will be substituted.

2.3. Prominent SDA Packages

The most prominent packages on CRAN are commonly used for statistical or machine learning analyze. It can be briefly classified into two parts, one is focused on statistic analysis, and the other is general SDA packages including both analysis method and some graphical technology. Nevertheless, most of their graphical technology tends to use the basic graphics in R rather than **ggplot2**, or only visualizes univariate distribution which is difficult to present the relationship between variables.

On the contrary, **ggESDA** uses a high-level graphic system by **ggplot2** to solve the problem mentioned above and provides a variety of EDA methods in all kinds of the variate, which can be summarized as the table 1. The number in table 1 shows how many methods are provided in its field.

Table 1: Compare with R packages

		Available	**Summarize**		**EDA**		**Statistic**	**Machine learning**	
Package	Author	Version	Function	Univariate	Bivariate	Multivariate	Stat. Method	Supervised	Unsupervised
RSDA	Rojas *et al.* (2015)	R(4.1.0)	`classic.to.sym`	-	1	-	3	**2**	**1**
symbolicDA	Dudek *et al.* (2013)	R(4.1.0)	-	-	-	**2**	2	1	**1**
HistDAWass	Irpino (2015)	R(4.1.0)	`data2hist`	4	-	-	3	1	**1**
MAINT.Data	Silva & Brito (2011)	R(4.1.0)	-	-	-	-	**7**	-	**1**
iRegression	Neto *et al.* (2011)	R(4.1.0)	-	-	-	-	1	-	-
intReg	Toomet (2012)	R(3.6.0)	-	-	-	-	1	-	-
ISDA.R	Filho & Fagundes (2012)	R(2.15.2)	-	1	-	1	1	-	-
GPCSIV	Brahim (2013)	R(3.0.2)	-	-	-	-	1	-	-
GraphPCA	Brahim & Kallyth (2014)	R(4.1.0)	-	-	1	1	1	-	-
ggESDA	Jiang (2021)	R(4.1.0)	`classic2sym`	**8**	**4**	**2**	1	-	-

In python, we can also find the SDA package such as **iardacil** Umbleja, Ichino, and Yaguchi (2020) which is available from the Github at `https://github.com/iardacil/SDA`. The zoomstart software in **iardacil** is provided with SODAS software project Diday and Noirhomme-Fraiture (2008). It is a basic thinking for general radar plot, improved for distinct groups visualization by **ggESDA**, and implemented in R using **ggplot2**.

3. Basic numerical summaries

The main content of EDA relates to the basic numerical summaries of data (e.g., the central tendency measures, and variation or variability measures) and the basic graphical summaries of data. For example, the five-number summary of numerical data (minimum, 25% quartiles, median, 70% quartiles, and maximum) is used to construct a boxplot. In the field of SDA, there are many algorithms to calculate descriptive statistics and frequency for interval-valued data, and we will illustrate the univariate and bivariate summaries respectively.

3.1. Univariate summaries

To build a statistic chart or analysis, descriptive statistics are necessary to be constructed, as well as the frequency occurring in each bin in a histogram chart. For a histogram chart, subdivisions of it into equidistant and non-equidistant will also be consider in this section.

Descriptive statistics

For the quantile in interval-valued data, summarizing it may seem to be obvious to separate data into a minimum and maximum data table, then calculate quantiles of both data tables to build a new interval-valued quantile data table.

In statistics, it may be more interesting to discuss mean and variance in a particular random variable Z; see Bertrand and Goupil (2000). The realization of Z for the observation W_u is the interval $Z(W_u) = [a_u, b_u]$, where $u = 1, 2, \cdots, m$ and m is the number of concepts.

First of all, assume that each object is equally likely to be observed with probability $\frac{1}{m}$, and the empirical density function of Z is defined as :

$$f(\xi) = \frac{1}{m} \sum_{u:\xi \in Z(W_u)} (\frac{1}{b_u - a_u}) \tag{1}$$

where ξ is the individual descriptions.

The Equation (1) is also equivalently to :

$$f(\xi) = \frac{1}{m} \sum_{u \in E} \frac{I_u(\xi)}{\|Z(u)\|}, \xi \in \mathbb{R} \tag{2}$$

where $I_u(.)$ is the indicator function that ξ is or is not in the interval $Z(u)$, $\|Z(u)\|$ is the length of that interval, and $E = \{w_1, w_2, \cdots, w_m\}$.

Further, the symbolic sample mean from definition for Z is $\bar{Z} = \int_{-\infty}^{\infty} \xi f(\xi)\, d\xi$, which can be reduced as :

$$\bar{Z} = \frac{1}{m} \sum_{u \in E} \frac{a_u + b_u}{2} \tag{3}$$

Finally, after getting the sample mean, the symbolic sample variance can be defined as follow:

$$\begin{aligned} S^2 &= \int_{-\infty}^{\infty} (\xi - \bar{z})^2 f(\xi)\, d\xi \\ &= \int_{-\infty}^{\infty} \xi^2 f(\xi)\, d\xi - \bar{z} \end{aligned} \tag{4}$$

and substituting for $f(\xi)$ from Equation (2), we have

$$\begin{aligned} \int_{-\infty}^{\infty} \xi^2 f(\xi)\, d\xi &= \frac{1}{m} \sum_{u \in E} \int_{-\infty}^{\infty} \xi^2 \frac{I_u(\xi)}{\|Z(u)\|}\, d\xi \\ &= \frac{1}{m} \sum_{u \in E} \int_{a_u}^{b_u} \frac{\xi^2}{(b_u - a_u)}\, d\xi \\ &= \frac{1}{3m} \sum_{u \in E} (\frac{b_u^3 - a_u^3}{b_u - a_u}) \end{aligned} \tag{5}$$

Hence,

$$S^2 = \frac{1}{3m}\sum_{u\in E}(a_u^2 + a_u b_u + b_u^2) - \frac{1}{4m^2}\left[\sum_{u\in E}(a_u + b_u)\right]^2 \tag{6}$$

Histogram Frequency

For the univariate histogram frequency, assume that we partition the interval $I = [\min_{u\in E} a_u, \max_{u\in E} b_u]$ into r subintervals, and all of them in the histogram are equal distance. That is, $I_g = [\zeta_{g-1}, \zeta_g), g = 1, 2, \cdots, r$, then $\|I_j\| = \|I_k\|, j, k = 1, 2, \cdots, r$. As a consequence, the observed frequency of the interval-valued variate Z for the histogram subinterval I_g from the definitions is

$$f_g = \sum_{u\in E}\frac{\|Z(u)\cap I_g\|}{\|Z(u)\|} \tag{7}$$

Moreover, for the interval-valued variate Z, we can pool the a_u and b_u from the interval of all observations, and sort it as a new vector $(x^{(1)}, x^{(2)}, \cdots, x^{(2m)})$ to represent the cut of a histogram. The subinterval from the cut is then defined as $I'_g = [x^{(j)}, x^{(j+1)})$, where $j = 1, 2, \cdots, 2m-1$, and apply the Equation (7) to get frequency. In most cases, $\|I'_g\|$ will not be equal to another, so we can get another histogram type, called non-equidistant-bin histogram.

3.2. Bivariate Summaries

Many of the principles developed for the univariate case can be expanded to a general p-variate case, $p > 1$. We shall focus attention on obtaining joint histograms for $p = 2$. The following will reveal the statistics and bivariate histogram.

Descriptive Statistics

For bivariates interval-valued variables, Z_1 and Z_2, the observations u, where $u \in E$, on the rectangle $Z(u) = Z_1(u) \times Z_2(u)$ is $([a_{1u}, b_{1u}], [a_{2u}, b_{2u}])$. Assume the individual description vectors ξ are each uniformly distributed over the respective intervals $Z_1(u)$ and $Z_2(u)$. Define the empirical joint density function for (Z_1, Z_2):

$$f(\xi_1, \xi_2) = \frac{1}{m}\sum_{u\in E}\frac{I_u(\xi_1, \xi_2)}{\|Z(u)\|} \tag{8}$$

where $I_u(\xi_1, \xi_2)$ is the indicator function that (ξ_1, ξ_2) is or is not in the rectangle Z_u and where $\|Z(u)\|$ is the area of this rectangle.

There is three methods to calculate bivariate interval-valued variables covariance function. For the first one, it can be derived by Billard and Diday (2003) as following:

$$cov(Z_1, Z_2) = \frac{1}{4m}\sum_{u\in E}(b_{1u} + a_{1u})(b_{2u} + a_{2u}) - \frac{1}{4m^2}\left[\sum_{u\in E}(b_{1u} + a_{1u})\right]\left[\sum_{u\in E}(b_{2u} + a_{2u})\right] \tag{9}$$

Second, an alternative expression of the symbolic sample variance for Z in Equation (6) can be expressed as

$$S^2 = \frac{1}{3m} \sum_{u \in E} \left[(a_u - \bar{Z})^2 + (a_u - \bar{Z})(b_u - \bar{Z}) + (b_u - \bar{Z})^2 \right]$$

The above equation can be generalized by Billard and Diday (2007) to formulate the form of the symbolic sample covariance for Z_j and $Z_{j'}$ as

$$cov(Z_j, Z_{j'}) = \frac{1}{3m} \sum_{i=1}^{m} G_j G_{j'} \left[Q_j Q_{j'} \right]^{1/2}, \quad j, j' = 1, 2, \cdots, p \tag{10}$$

where for $J = j, j'$,

$$Q_J = (a_{iJ} - \bar{Z}_J)^2 + (a_{iJ} - \bar{Z}_J)(b_{iJ} - \bar{Z}_J) + (b_{iJ} - \bar{Z}_J)^2,$$
$$G_J = \begin{cases} -1, & \text{if } \xi_{iJ}^c \le \bar{Z}_J \\ 1, & \text{if } \xi_{iJ}^c > \bar{Z}_J \end{cases}$$

and ξ_{iJ}^c is the midpoint of the interval $[a_{iJ}, b_{iJ}]$.

Last, it was further demonstrated by Billard (2007) and Billard (2008) that the sample variance in Equation (6) is a function of the total sum of squares (SST) and that the SST can be decomposed into the sum of the internal (within) variation and the between variation. The total sum of products (SPT) is the sum of the within sum of products and the between sum of products. The Equation (6) was also extended to the bivariate case to obtain the sample covariance of Z_j and $Z_{j'}$ based on the decomposition of the SPT as

$$\begin{aligned} cov(Z_j, Z_{j'}) = \frac{1}{6m} \sum_{i=1}^{m} \Big[& 2(a_{ij} - \bar{Z}_j)(a_{ij'} - \bar{Z}_{j'}) + (a_{ij} - \bar{Z}_j)(b_{ij'} - \bar{Z}_{j'}) \\ & + (b_{ij} - \bar{Z}_j)(a_{ij'} - \bar{Z}_{j'}) + 2(b_{ij} - \bar{Z}_j)(b_{ij'} - \bar{Z}_{j'}) \Big] \end{aligned} \tag{11}$$

The definitions and calculations of the symbolic sample covariance in Equations (9)-(11) are consistent with the results in the classic data case if $a_{ij} = b_{ij}$ for $i = 1, 2, \cdots, m, \; j = 1, 2, \cdots, p$. If $j = j'$, the Equation (11) reduces to the sample variance of the interval-valued variable as given in Equation (6).

Two-Dimensional Histogram

Analogously with Equation (7), we can find the joint histogram for (Z_1, Z_2) by graphically plotting $\{R_{g_1 g_2}, p_{g_1 g_2}\}$ over the rectangles $R_{g_1 g_2} = \{[\zeta_{1,g_1-1}, \zeta_{1,g_1}) \times [\zeta_{2,g_2-1}, \zeta_{2,g_2})\}, \; g_1 = 1, 2, \cdots, r_1, \; g_2 = 1, 2, \cdots, r_2$, where

$$f_{g_1 g_2} = \sum_{u \in E} \frac{\| Z(u) \cap R_{g_1 g_2} \|}{\| Z(u) \|} \tag{12}$$

i.e., $f_{g_1 g_2}$ is the number of observations that fall in the rectangle $R_{g_1 g_2}$ and is not necessarily an integer value (except in the special case of classical data). The relative frequency that an arbitrary individual description vector lies in the rectangle $R_{g_1 g_2}$ is therefore

$$p_{g_1 g_2} = \frac{f_{g_1 g_2}}{m} \tag{13}$$

4. Application to Real Symbolic Datasets

The typical graphical techniques of EDA include the index plot, barplot, boxplot, and histogram for univariate data; the 2D scatterplot, joint histogram and double boxplot for bivariate data; and the satterplot matrix, and star plot for multivariate data. Of course, there are many other variants of graphical techniques have been proposed. Our aim in this study is to implement good statistical graphics to display symbolic data accurately and clearly.

4.1. Example Datasets

In the following, to illustrate some graphing functions we have implemented so far, we will use the two well-known symbolic datasets. One is face recognition data (Leroy, Chouakria, Herlin, and Diday (1996); Douzal-Chouakria, Billard, and Diday (2011); Le-Rademacher and Billard (2012)). The dataset gives six face measurements of nine men (Figure 5), each with three observations, resulting in a total 27 observations. The measurements for each observation came from a sequence of over 1,000 images. They cover a range of values, hence interval-valued variables.

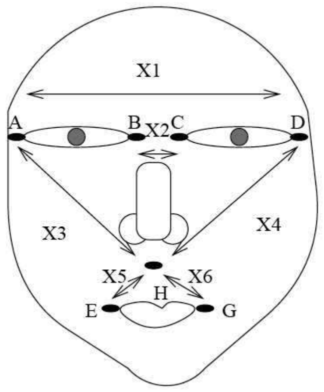


Figure 5: The six face measurements for the face recognition dataset

The other is the Environment questionnaire data (from the SODAS software package Diday and Noirhomme-Fraiture (2008)). Because of the demonstration of the performance dealing with modal multi-valued data, the same datasets were used in the radar plot. This dataset contains 14 objects and 17 variables and 4 of them are modal multi-valued, while the rest are interval-valued variables. Both of the part of example datasets are shown as follow:

```
> facedata[1:3, 1:4]
```

```
# A tibble: 3 x 4
                 AD               BC                 AH
         <symblc_n>       <symblc_n>         <symblc_n>
1 [155.00 : 157.00] [58.00 : 61.01] [100.45 : 103.28]
2 [154.00 : 160.01] [57.00 : 64.00] [101.98 : 105.55]
3 [154.01 : 161.00] [57.00 : 63.00]  [99.36 : 105.65]
# ... with 1 more variable: DH <symblc_n>

> Environment[1:3, 4:6]

# A tibble: 3 x 3
              REGIONDEVELOPME              CONTROL
                   <symblc_m>          <symblc_n>
1 4:0.47 3:0.25 2:0.19 1:0.09 [-723.25 : -339.65]
2 4:0.49 3:0.32 2:0.08 1:0.10  [-243.86 : 203.99]
3 4:0.56 3:0.22 2:0.18 1:0.04   [-195.44 : 90.10]
# ... with 1 more variable: SATISFY <symblc_n>
```

4.2. Interval Presentation Techniques

For interval-valued data, one of the advanced graphics implemented is called the min-max plot, which shows the interval characteristics of the data. As Figure 6 shown, we visualize the Face data for example by the function

```
ggInterval_minmax(data = NULL, mapping = aes(NULL), scaleXY = "local",
  plotAll = FALSE)
```

For the top six of Figure 6 shows the difference between each location of concept in each variable through the 45-degree line, which is a relative position of the variable itself (by setting option `scaleXY = "local"`). And represents the range by the connected line between minimum and maximum. To design the widely-view exploratory graphic function like this instance, we apply the option `plotAll = TRUE` to extend one variable visualization to full variables in the same plot. For examples, the figure constructed by the code

```
> #One case for plotting all variables (Relative Position)
> lapply(1:6, FUN = function(x){
+   plot.var <<- x
+   ggInterval_minmax(facedata, aes(facedata[[plot.var]]))+
+     scale_color_manual(values = c("darkblue", "darkred"))+
+     guides(colour = F) +
+     theme_bw() +
+     coord_fixed(ratio = 1)}) %>%
+   ggarrange(plotlist = ., nrow = 1, ncol = ncol(facedata), labels = "")
> #The other case using 'plotAll = T' (Absolute Position)
> ggInterval_minmax(facedata, plotAll = T, scaleXY = "global") +
+   scale_color_manual(values = c("darkblue", "darkred")) +
+                 guides(colour = F) +
+                 theme_bw() +
+                 coord_fixed(ratio = 1)
```

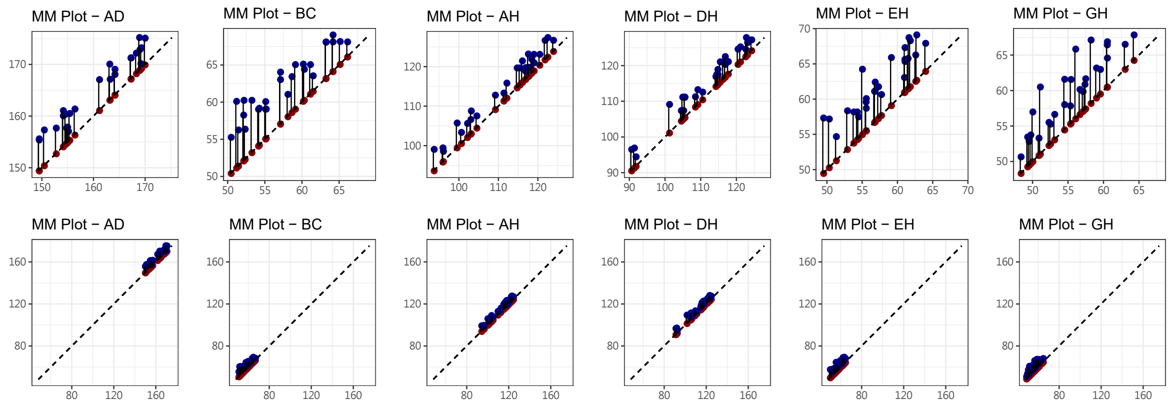


Figure 6: The min-max plot. The figure in the top six conditions the axis in their own variables, whereas the bottom six conditions it in all variables.

However, when the unit is considered, the absolute position in the full sample space will display as the bottom six of Figure 6. It presents a different boundary in the axis from the previous figure, which makes researchers know about the actual location in each variable. In this case, we can clearly discover that the sample space changes along with the variables.

To complete the visualization, it actually needs the command `ggInterval_minmax(facedata, plotAll = T, scaleXY = "local")`, merely. Other aesthetics can be omitted. Because we now focus on the full variables exploration, the `mapping` option for a particular variable is not necessary as well. Thus, in the following, we will pay more attention to the option we changing.

Another graphical technique, called index plot, is famous in conventional data as well. It is composed of minimum and maximum in interval-valued data and exhibits the concepts' behavior. Both of the function

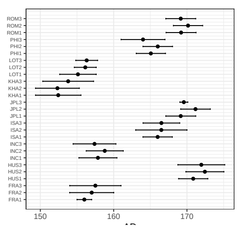
```
ggInterval_index(data = NULL, mapping = aes(NULL))
```

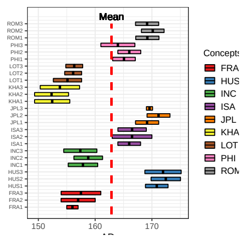
and

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

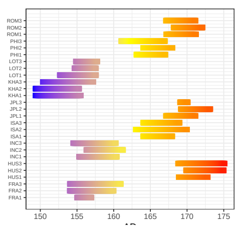
can achieve this objective, but there is still a slight difference for visualizing.

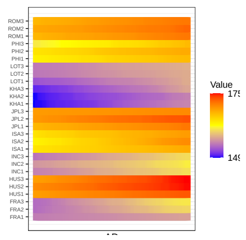
Take the variable AD for example, and set option `mapping = aes(x = AD)`. The former function, shown in Figures 7a and 7b, directly displays the interval which is similar to min-max plot. Among these plots, we classify the concepts by the same person by setting the aesthetics option `fill` and can discover the values of people LOT, KHA, INC, FRA in AD variable are less than others, which easily compares the difference.


(a) Index plot



(b) Index plot with different group and common mean


(c) Index image



(d) Index image with extended color

Figure 7: Four kinds of index plot for the variable AD

The latter function, shown in Figure 7c and 7d, visualizes the interval values using a color mapping. As the legend says, the blue color corresponds to the low values, whereas the red color corresponds to the high. The only difference among these plots is the color bar extension (control by the option `full_strip = TRUE`), which will quickly discover concepts' behavior through the visual sense, as Figure 8 shown.

For exploring data, we extend the index plot into full variables and visualize by color mapping as we did before using the code

```
> #Column Condition
> colCondition <- ggInterval_indexImage(facedata, plotAll = T,
+                                        full_strip = T) +
+   scale_colour_gradient2(low = "green", mid = "gray15", high = "red")
> #Matrix Condition
> #Prepare midpoint for scale_colour_gradient2
> ggESDA.facedata <- RSDA2sym(facedata)
> facedata.m <- mean(c(min(ggESDA.facedata$statisticsDF$min),
+                      max(ggESDA.facedata$statisticsDF$max)))
> matCondition <- ggInterval_indexImage(facedata, plotAll = T,
+                                        full_strip = T,
+                                        column_condition = F) +
+   scale_colour_gradient2(low = "green", mid = "gray15",
+                          high = "red", midpoint = facedata.m)
> ggarrange(colCondition, matCondition, nrow = 1, ncol = 2)
```

Figure 8 shows two kinds of presentation ways for this method.

- *Column condition*: This technique is to standardize data by each variable, which will be able to compare the observations easily without the unit confounder. Figure **??** ,for example, shows the concept JPL has a large value in all variables in contrast with KHA and INC.
- *Matrix condition*: Instead of standardizing variables one by one, this method does it with the full data matrix. That will reveal the different performance between each variable, as Figure **??** shown. The values in variable AD are obviously larger than

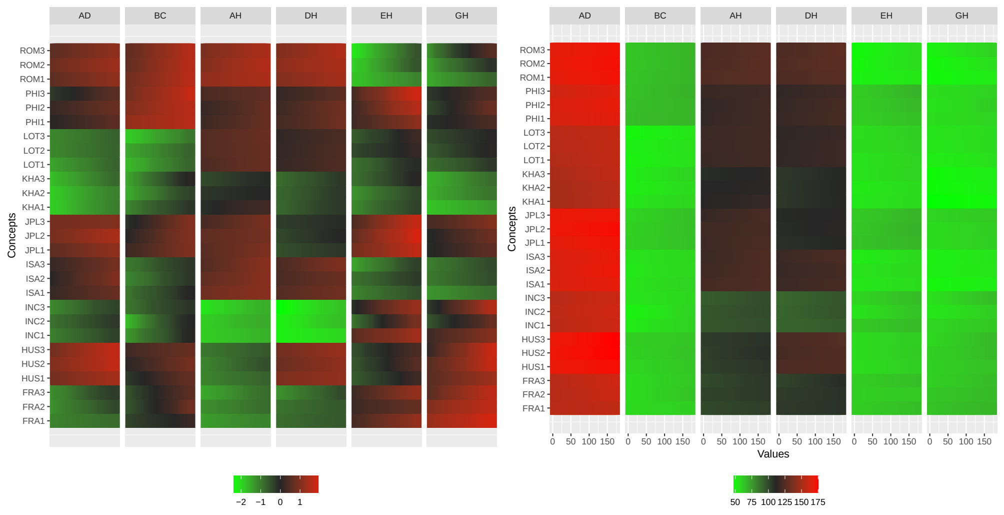


Figure 8: Index image using a heatmap type.

others, which means the interocular distance is the longest than other distances in the face dataset.

4.3. Statistical Graph Implementation

In this field, it reveals the trend, distribution, how large the data spread, etc, for a particular variable. Moreover, another advanced graphics implemented is called center-range plot, which helps researchers to be able to grasp the relationship between center and range.

The side-by-side boxplot, which provides a quick view of data distribution, dispersion, and is particularly useful for comparing distributions between several groups of variables. Use the function

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

By setting option `plotAll`, the top panel of Figure 9 will be easily implemented. In this Figure, the dispersion of six variables is obviously distinct. However, it is difficult to compare with their distribution because of the distinct sample space which leaves no room for a suitable presentation. We standardize the Face data, shown in the bottom panel of the figure, and take a deeper look at it.

Clearly, there is a vacancy in the middle of variable AD, AH, and next to the minimum of the variable DH, which indicates the frequency there may be quite low. Especially for AD and AH, this factor may result in a bimodal distribution.

In addition, the overlap of the variables BC, EH, and GH located in the middle of five quantile-box seem larger than others, which is possible to lead a bell-shaped for its distribution.

For the center-range plot, it is a special technique for the dispersion observing. The code

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

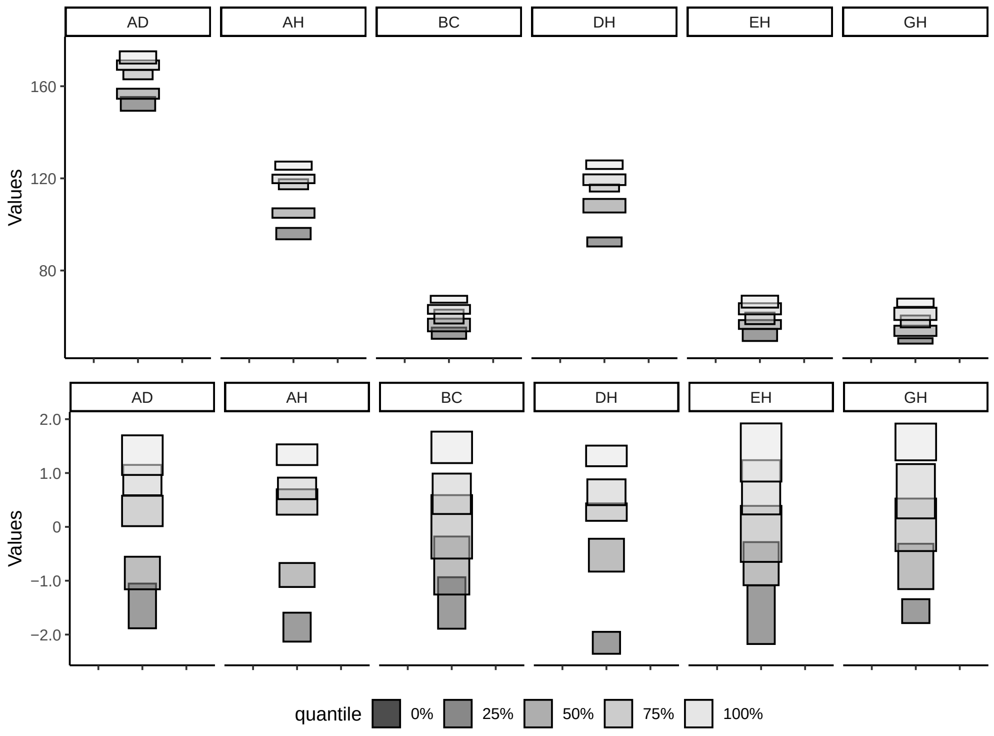


Figure 9: The side-by-side box plot. The figure in the top panel is the raw value of its variables, where the bottom panel is to standardize each variables.

Setting options are the same as the former. With the Figure 9 demonstration, we divide the exploration into two parts by standardization, shown in Figure 10.

With regard to the top panel of Figure 10, there is no significant difference in the range of all variables. But focusing on a particular variable, the dispersion is quite large. For the bottom panel of Figure 10, the relationship between center and range is clearly exposed. Among them, there is an obvious relationship in the variable BC, which presents a negative correlation, where others are not significant.

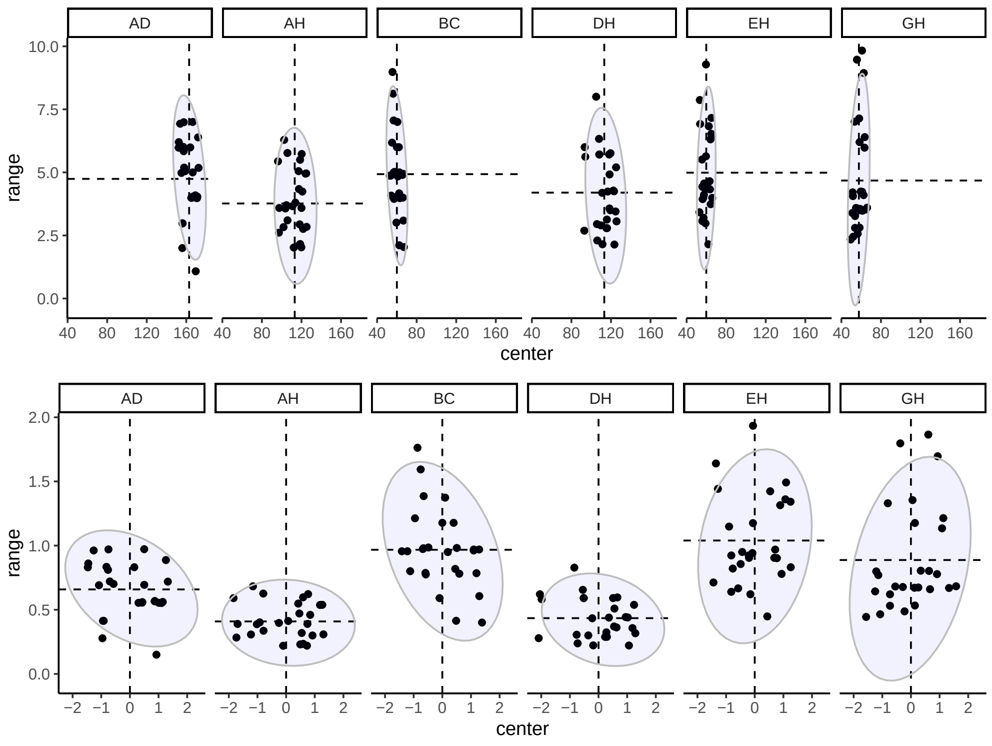
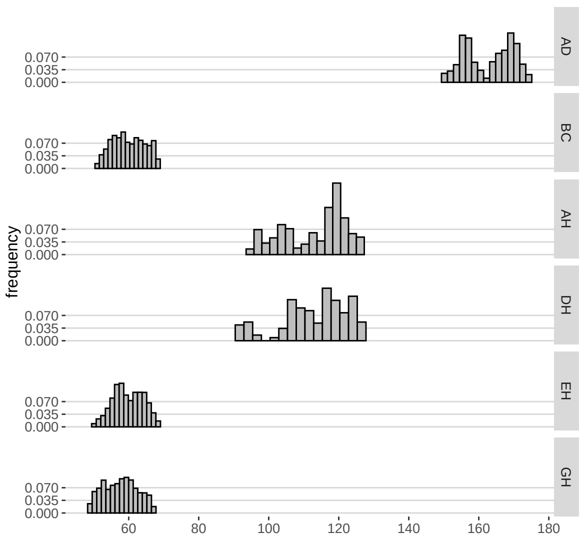


Figure 10: The center-range plot

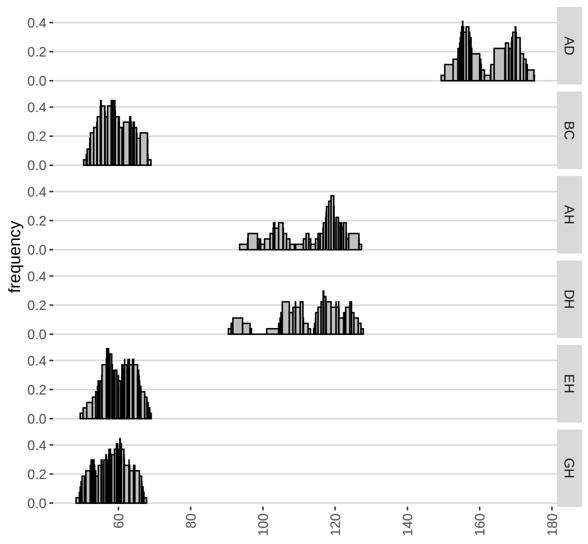
In the same way, demonstrating the distribution of data by histogram is also one of the most common ways. Construct the frequency of data by Section 3.1.2 for the first, which will be processed by the function

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

Option `method` is the key for setting whether the histogram bins are equal width, detail in Section 3.1.2, which can be set with `"equal-bin"` or `"unequal-bin"`. In Figure 11 depicts the histograms with equidistant and non-equidistant breaks for all six variables. Both of them present the location and scale of distribution of each variable, but the non-equidistant-bin histogram whose breaks are obtained by the boundaries of the intervals shows more details about the characteristic of data. As we did before, set `plotAll` and `method`, and it can be directly completed with one line command. In the figure, we may obtain some similar distribution information conclusion with Figure 9.



(a) Equidistant-bin histogram



(b) Non-equidistant-bin histogram

Figure 11: Histogram

4.4. Bivariate Relationship Plot

The bivariate functions reveal relationships between variables, trends of the regression line, and joint frequency across the sample space. In this field, we design the scatter plot, two-dimension histogram, and also implement the most well-known approach, the vertices method of principal component analysis (V-PCA) Cazes, Chouakria, Diday, and Schektman (1997). Additionally, these graphic techniques can visualize multiple variables at the same time by matrix type. For instance, the code of scatter matrix of Figure 12:

```
> ggInterval_scaMatrix(data = facedata)+
+   theme_bw()
```

In Figure 12, we use grayscale to represent the concepts and the rectangle to display the intersection of two interval-valued observations. The relationship among this graph is clearly depicted. The variables AH and DH are an obvious positive correlation, whereas AD and AH seems to be an exponential relation. Of course, there is an easy way for scatter plot of two particular variables using the function `ggInterval_scatter`. With the technique, we combine the dimension reduction methods that **RSDA** provides, and implement the maximum covering area rectangle (MCAR) Cazes *et al.* (1997) using the code:

```
> pca.results <- RSDA::sym.pca(facedata, method = "top")
> ggInterval_scatter(pca.results$Sym.Components, aes(Dim.1, Dim.2)) +
+   scale_fill_manual(values = Concepts)
```

which is widely used for graphical representation of interval objects on a DR (dimension reduction) subspace, owing to its simplicity, shown as Figure 13a.

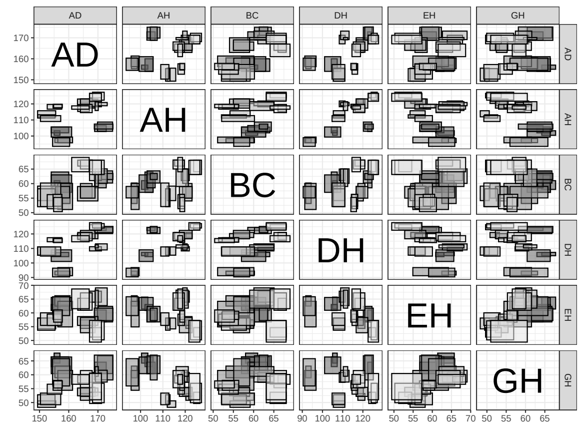
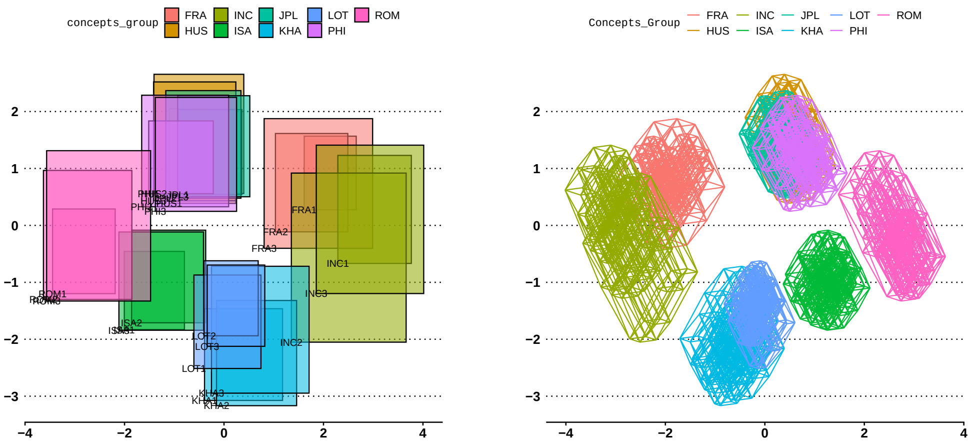


Figure 12: Scatter matrix



(a) Vertice-PCA (MCAR) (b) Vertice-PCA (Polytope)

Figure 13: Two kinks of PCA for interval-valued data, where the XY-axis are represent the first principal component and the second principal component, respectively.

Nevertheless, the main drawback of MCAR is that the interval objects are oversized in the DR subspace with respect to the real objects in $\mathbb{R}^p$, hence, the polytopes representation Le-Rademacher and Billard (2012) for interval data was proposed. In **ggESDA**, we have also developed the graphic function for them using

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

The control option of PCA visualization method mentioned above is `poly`, which can be set with `TRUE` for the polytopes representation as shown in Figure 13b. By the dimension reduction and color mapping on each person, Figure 13 shows most of the concepts are clearly separated.

Another bivariate relationship plot is two-dimension histogram, which is to calculate joint histogram frequency (details in Section 3.2.2) and using matrix visualization (Chen (2002); Chen, Hwu, Jang, Kao, Tien, Tzeng, and Wu (2004)) to plot. The code

```
> ggInterval_2Dhist(facedata, aes(x = AD, y = BC, col = "grey30"),
+                  xBins = 25, yBins = 25) +
+   theme(legend.position = "bottom") +
+   scale_fill_gradient(low = "gray95", high = "red", limits = c(0, 0.15),
+                       na.value = "red")
```

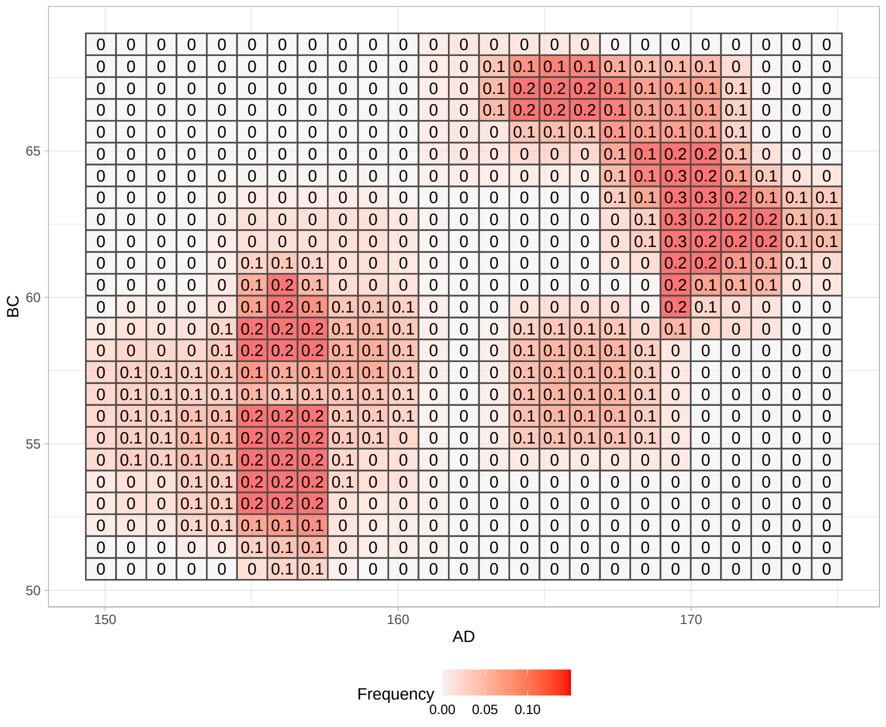


Figure 14: 2D histogram

uses the joint frequency between the variables AD and BC and makes the XY-axis bins partial into 25 subintervals. As Figure 14 shown, we can observe that the concepts in these joint sample spaces concentrate on the top-right and bottom-left areas that seem like a slightly positive correlation.

In the same way, extend the plot to full variables as matrix type by the code

```
> ggInterval_2DhistMatrix(facedata, aes(col = "grey50"), xBins = 10,
+                          yBins = 10, removeZero = T, addFreq = F) +
+   theme_light() +
+   theme(legend.position = "bottom") +
+   scale_fill_gradient(low = "gray95",
+                       high = "red",
+                       limits = c(0, 1),
+                       na.value = "red")
```

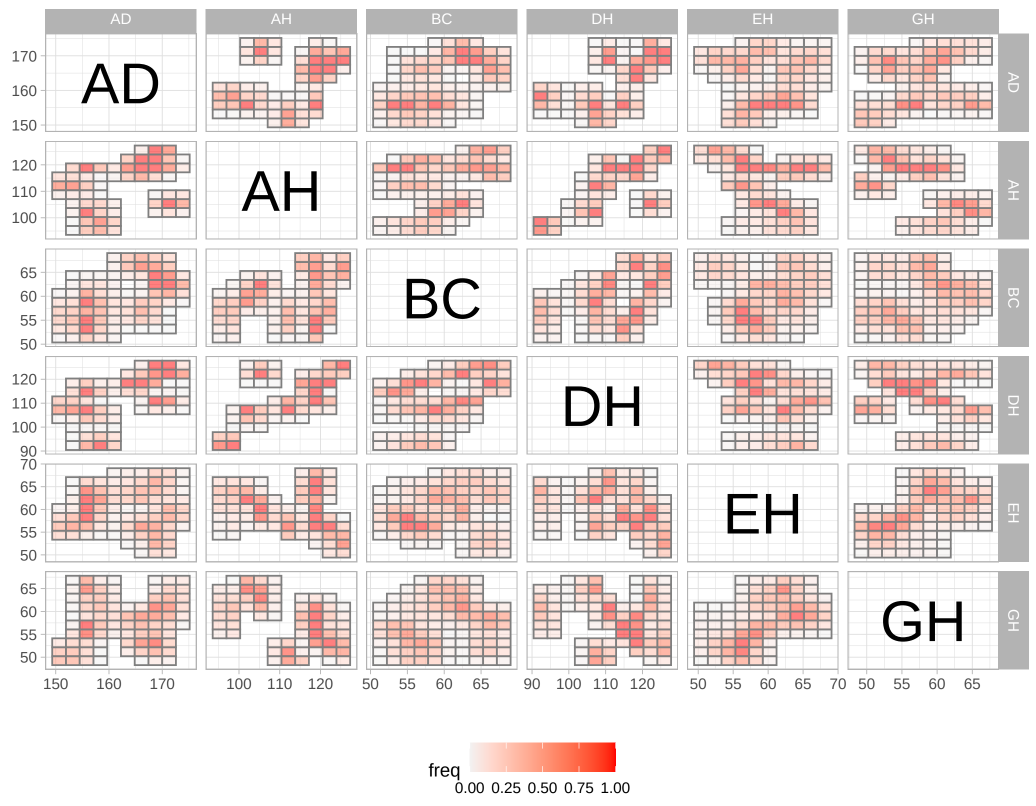


Figure 15: 2D histogram matrix

The addition options `removeZero` and `addFreq` make zero value and frequency displaying remove, shown as Figure 15. Not only the relationship presentation, the graphic technique also depicts the details frequency. Among these, we might get a similar conclusion between variables relation, but for the calculation, this technique may consume a larger time-complexity than scatter plot.

4.5. Multivariate Visualization

In the field of multivariate visualization, the most common graphic technique is radar plot, also known as star plot. In **ggESDA**, we provide the function

```
ggInterval_radar(data = NULL, layerNumber = 3, inOneFig = TRUE, Drift = 0.5,
  showLegend = TRUE, showXYLabs = FALSE, plotPartial = NULL, alpha = 0.5,
  base_circle = TRUE, base_lty = 2, addText = TRUE, type = "default",
  quantileNum = 4, addText_modal = TRUE, addText_modal.p = FALSE)
```

Except for the aesthetic options `layerNumber`, `showLegend`, `showXYLabs`, `alpha`, `base_circle`, `base_lty`, and `addText`, the visualizing key is to determine which concepts are presented through setting `plotPartial` with the rows of concepts. In addition, it will be decided to use area (`"default"`), rectangle (`"rect"`), or quantile (`"quantile"`) to represent the interval by option `type`.

Above all, take the Environmental questionnaire dataset for example, and the variety of typical multivariate radar plots for a specific concept is shown in Figure 16 by the code

```
> #Left fig : Remove the modal multi-valued variables for demo
> ggInterval_radar(Environment[, 5:17], plotPartial = 2, showLegend = F,
+                  addText = F)
> #Right fig : Remain the modal multi-valued variables
> ggInterval_radar(Environment, plotPartial = 2, showLegend = F,
+                  base_circle = F, base_lty = 1) +
+   scale_fill_manual(values = "gray50") +
+   scale_color_manual(values = "gray50")
```

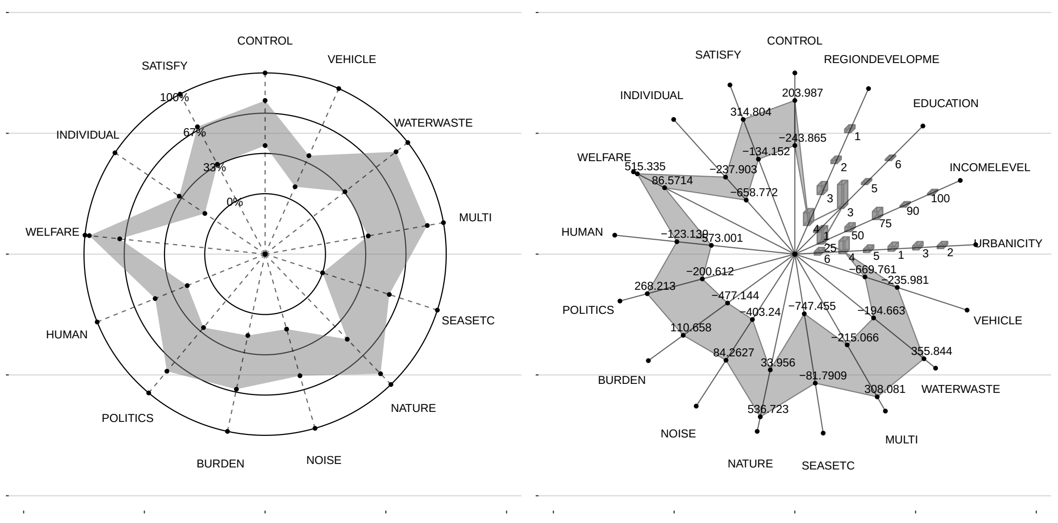


Figure 16: The variety of typical radar plot which is able to contain interval-valued and modal multi-valued variables.

Use the color to represent the interval area, and it can clearly present the behavior of the particular concept in each variable. The figure in the left panel shows the relative position of the interval by the percentage circle, whereas the right one removes it and adds the frequency in the plot. Additionally, the kind of radar plot is generalized to visualize not only the interval-valued data but the modal multi-valued, which we design a cuboid to represent, and the height will correspond to the probability of the factor in the variable, as the right panel in the figure. The technique not just merges the distinct variable's type in one plot but even helps the researcher to compare with concepts easily under multiple variables. Figure 17 shows the three observations presented by the different plot.

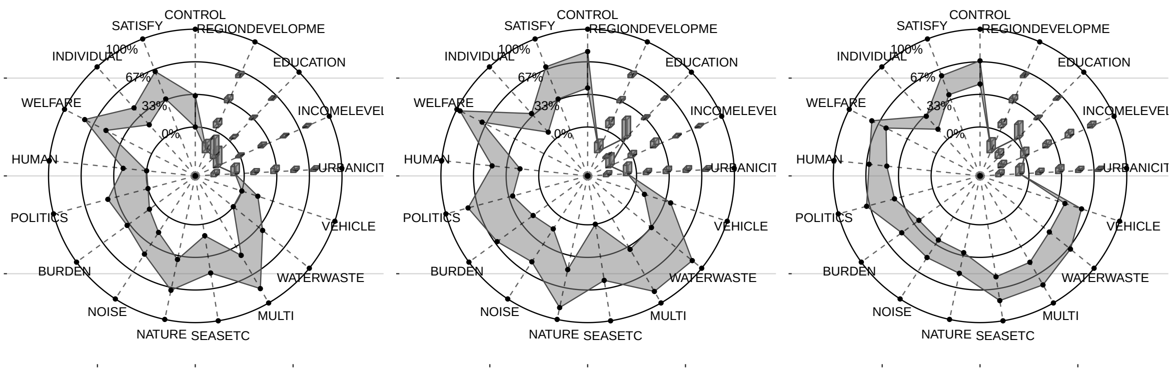


Figure 17: Compare three distinct observations by multiple variables.

Moreover, another advanced implemented graphic technique of radar plot is classification by different concepts using a color mapping. By the code:

```
> area.fig <- ggInterval_radar(Environment, plotPartial = c(4, 6),
+                              showLegend = F, base_circle = F,
+                              base_lty = 1, addText = F,
+                              addText_modal = F) +
+             scale_fill_manual(values = c("darkred", "darkblue")) +
+             scale_color_manual(values = c("darkred", "darkblue")) +
+             labs(title = "") +
+             theme_hc()
> rect.fig <- ggInterval_radar(Environment, plotPartial = c(4, 6),
+                              showLegend = F, base_circle = F,
+                              base_lty = 1, addText = F, type = "rect",
+                              addText_modal = F) +
+             scale_fill_manual(values = c("darkred", "darkblue")) +
+             scale_color_manual(values = c("darkred", "darkblue")) +
+             labs(title = "") +
+             theme_hc()
> gridExtra::marrangeGrob(list(area.fig, rect.fig),
+                         nrow = 1, ncol = 2, top = "")
```

we can get the Figure 18. It's worth mentioning that whenever the users determine the option `plotPartial` with multiple observations and set `inOneFig = TRUE` which means if pool observations into the same figure, the mechanism in the function will automatically classify by color. In this case, the two visualized concepts are represented by blue and red color, and the modal multi-valued variables are stacked by the same factor. On the contrast, the graph displays the interval-valued variables in two ways. The left panel of the figure shows a typical area-presenting way, while the right one use rectangle to show the interval. These implement can be completed by setting the option `type` with `"default"` and `"rect"` respectively.

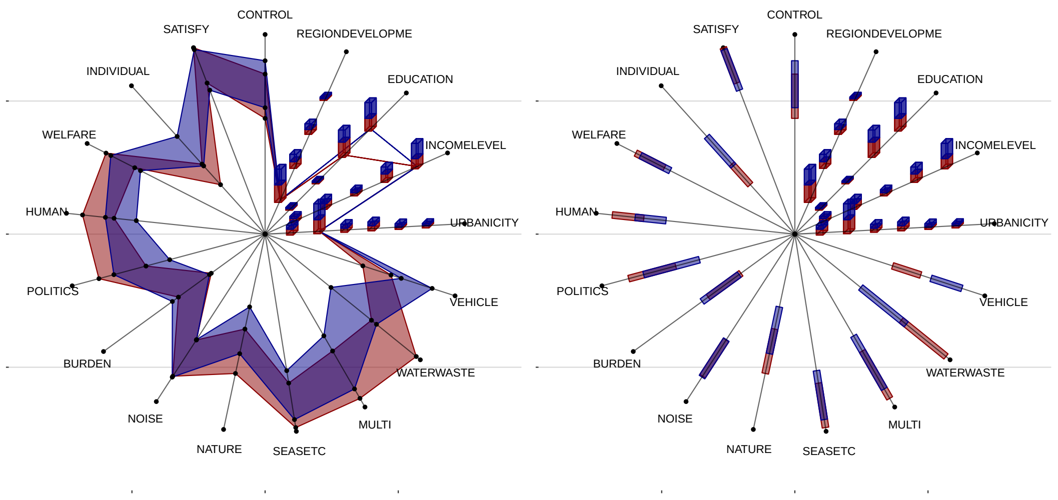


Figure 18: Compare the distinct observations in the same figure by color mapping.

Last but not least, we propose a graphic method based on quantiles. The benefit of this proposed method is that distributional details are visible for the graph. Under the option `type = "quantile"`, for example, we divide the quantile percentage of 25% to 75% into the equal subintervals represented as grayscale and shown in Figure 19. The variable WELFARE of the left panel in the figure depicts the rapid increase close to the 3rd quartiles, which implies it exists a left-skewed distribution. In this way, it becomes more straightforward to extend the comparison of variables to datasets.

5. Generalization and Extension

As far as generalization and extension are concerned, the package provides a simple way for making connections with other useful packages, so that the result of common statistical or machine learning methods on other packages may be visualized as well using **ggESDA** if it is interval-valued. In addition, we may get the classical data most of the time, so summarization between classical to interval-valued data becomes important. The following will discuss these explicitly.

5.1. Generalization with Prominent SDA Packages

For the demonstration, we consider two famous R packages for SDA, **HistDAWass** Irpino

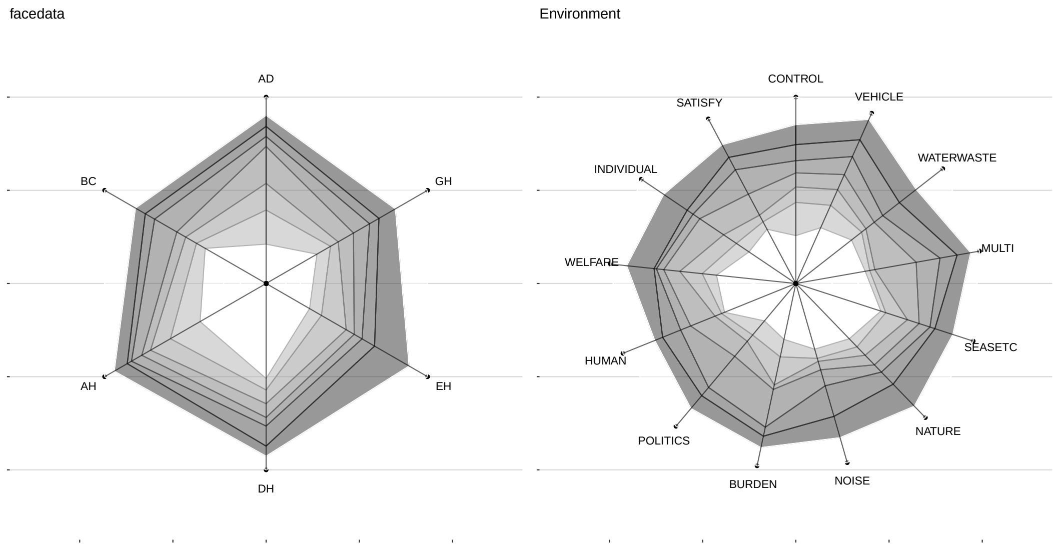


Figure 19: Quantile for two symbolic datasets.

(2021) and **MAINT.Data** Silva and Brito (2021). Both of these make lots of contributions to the statistics of SDA, so we tend to generalize these data into the **ggESDA** type, which may help us to visualize.

General Principle

Generally, it is not merely the well-known packages in R that can make a plot using it. As long as keeping some principle, it will be easily performed:

1. Understand the data structure clearly if it is an object from other packages.
2. Extract the min data and max data from it or make some necessary transformation.
3. Classify the data you extract belong.
4. Reorganized it and use `classic2sym` for the final transformation.
5. Visualize the front result using **ggESDA**.

Because of the interval-valued data, all SDA packages studying in the same field will store and deal with the min and max data. Hence, the transformation method in section 5.2.5 plays a important role. With the connection being built, it can be compatible with all other tools in R.

HistDAWass

We further use the principle to process `BLOOD` data in **HistDAWass**, first. With the method **HistDAWass** provided, it will be more convenient to get min and max data:

```
> if (!require("HistDAWass")) install.packages("HistDAWass")
> library(HistDAWass)
> # Get min and max data
> BLOOD <- HistDAWass::BLOOD
> blood.min <- get.MatH.stats(BLOOD, stat = "min")
> blood.max <- get.MatH.stats(BLOOD, stat = "max")
> blood <- data.frame(blood.min, blood.max)
> # Reorganized and Build ggESDA obj.
> blood.sym <- classic2sym(blood, groupby = "customize",
+                         minData = blood[, 2:4],
+                         maxData = blood[, 6:8])
> # Make names
> blood.names <- get.MatH.main.info(BLOOD)$varnames
> blood.i <- blood.sym$intervalData
> colnames(blood.i) <- blood.names
> head(as.data.frame(blood.i), 5)

        Cholesterol      Hemoglobin      Hematocrit
1  [80.00 : 240.00] [12.00 : 15.00] [35.00 : 47.00]
2  [80.00 : 240.00] [10.50 : 14.00] [31.00 : 44.00]
3  [95.00 : 245.00] [10.50 : 14.00] [31.00 : 43.50]
4 [105.00 : 260.00] [10.50 : 14.00] [31.00 : 42.50]
5 [115.00 : 260.00] [10.80 : 13.60] [31.00 : 42.50]
```

After getting the necessary data, classifying data belonging is vital for reorganization, which means that differentiating the min data and max data. For instance, `minData = blood[, 2:4]` represents the min data are the columns of $2, 3, 4$ in this case.

MAINT.Data

However, it is also a common way to store interval-valued data by median and range. In **MAINT.Data**, the data will exist in this form. Fortunately, a median-range form is not difficult to deal with. We can do the necessary conversion directly to get the data we expect:

```
> if (!require("MAINT.Data")) install.packages("MAINT.Data")
> library(MAINT.Data)
> #get data interval-valued data in AbaloneIdt
> AbaloneIdt <- MAINT.Data::AbaloneIdt
> Aba.range <- AbaloneIdt@LogR
> Aba.mid <- AbaloneIdt@MidP
> #make a necessary transformation for build min max data
> Aba <- data.frame(Aba.min = Aba.mid - exp(Aba.range) / 2,
+                   Aba.max = Aba.mid + exp(Aba.range) / 2)
> # Reorganized and Build ggESDA obj.
> Aba.sym<- classic2sym(Aba, groupby = "customize",
+                         minData = Aba[, 1:7],
+                         maxData = Aba[, 8:14])
> # Make names
> colnames(Aba.sym$intervalData) <- AbaloneIdt@VarNames
> Aba.i <- Aba.sym$intervalData %>%
+   cbind(Aba.obs = AbaloneIdt@ObsNames) %>%
+   column_to_rownames(var = "Aba.obs")
> head(Aba.i[, 1:4], 5)
```

```
               Length      Diameter        Height  Whole_weight
F-10-12 [0.34 : 0.78] [0.26 : 0.63] [0.06 : 0.23] [0.21 : 2.66]
F-13-15 [0.39 : 0.82] [0.30 : 0.65] [0.10 : 0.25] [0.27 : 2.51]
F-16-18 [0.40 : 0.74] [0.32 : 0.60] [0.10 : 0.24] [0.35 : 2.20]
F-19-21 [0.49 : 0.72] [0.36 : 0.58] [0.12 : 0.21] [0.68 : 2.12]
F-23-24 [0.45 : 0.80] [0.38 : 0.63] [0.14 : 0.22] [0.64 : 2.53]
```

In brief, following the general principle in section 5.1.1 may facilitate the integration, extend utilize of **ggESDA** and generalize to all SDA studies.

5.2. Interval-valued Data Summarization

Another possibility of the data structure would be a classical data, we provide multiple method for dealing with this kind of transformation. The following would discuss the functionality of `classic2sym`.

Classical Datasets

We will apply the breast mass dataset, which is computed from a digitized image of a fine needle aspirate (FNA), to demonstrate how does a classical dataset transforms into a symbolic dataset. The breast mass dataset describe characteristics of the cell nuclei present in the image. It can be downloaded from the kaggle at `https://www.kaggle.com/uciml/breast-cancer-wisconsin-data?select=data.csv`. There are 569 observations and 32 variables in the dataset. We are going to store this dataset in `breastData` as data frame type in R, and the variables will be shown as follows:

```
> colnames(breastData)
```

```
 [1] "id"                      "diagnosis"
 [3] "radius_mean"             "texture_mean"
 [5] "perimeter_mean"          "area_mean"
 [7] "smoothness_mean"         "compactness_mean"
 [9] "concavity_mean"          "concave points_mean"
[11] "symmetry_mean"           "fractal_dimension_mean"
[13] "radius_se"               "texture_se"
[15] "perimeter_se"            "area_se"
[17] "smoothness_se"           "compactness_se"
[19] "concavity_se"            "concave points_se"
[21] "symmetry_se"             "fractal_dimension_se"
[23] "radius_worst"            "texture_worst"
[25] "perimeter_worst"         "area_worst"
[27] "smoothness_worst"        "compactness_worst"
```

```
[29] "concavity_worst"         "concave points_worst"
[31] "symmetry_worst"          "fractal_dimension_worst"
```

Except for the first two variables, they are all composed of mean, standard error, and "worst" in their own field respectively.

K-means

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. In **ggESDA**, the algorithm will be based on the **stats** package, and the number of k is a parameter that user can define themself:

```
> breastData <- dplyr::select(breastData, -id)
> breastData.sym <- classic2sym(breastData, groupby = "kmeans", k = 5)
> breastData.sym.i <- breastData.sym$intervalData
> as.data.frame(head(breastData.sym.i[, 1:4], 5))
```

```
      diagnosis     radius_mean    texture_mean    perimeter_mean
1 B:0.04 M:0.96 [13.81 : 19.59] [11.89 : 39.28]  [91.56 : 132.40]
2 B:0.00 M:1.00 [15.50 : 24.25] [10.38 : 32.47] [102.90 : 166.20]
3 B:0.00 M:1.00 [20.73 : 28.11] [17.25 : 31.12] [135.70 : 188.50]
4 B:0.68 M:0.32 [11.84 : 16.30] [10.89 : 30.72]  [77.93 : 109.80]
5 B:0.98 M:0.02  [6.98 : 13.05]  [9.71 : 33.81]   [43.79 : 85.09]
```

The id is unused in this case, so we remove it by **dplyr** Wickham, Francois, Henry, and Muller (2021). Then using classic2sym to aggregate breastData. It will return several result sets include clustering result and interval-valued data, etc. The interval-valued data can be extracted by $intervalData, and it will be presented by the package of **RSDA** type.

The groupby is a parameter that determine what kind of aggregation methods will be used. Whenever the K-means method is applied, the consequent k will become meaningful, whereas the other situation is not. It is also a default method when users have no input arguments in groupby.

Hierarchical

The second well-known clustering algorithm is called Hierarchical clustering Cecil C. Bridges (1966), also called hierarchical cluster analysis or HCA. It can be performed with a distance matrix calculated by raw data and used to present the distance of each cluster. In basic R package, it is also realized by **stats**, which the **ggESDA** is based on for implementing HCA:

```
> breastData.sym <- classic2sym(breastData, groupby = "hclust")
> breastData.sym.i <- breastData.sym$intervalData
```

Remark that the k parameter is not meaningful in the case without K-means clustering. In classic2sym, the keywords of HCA is called hclust.

Variables Aggregation

Using a particular variable to merge different data is a common way for data analysis, too. **ggESDA** provides such as this concept in `classic2sym` to analyze different factors of category variables, and merge the same factor into the symbolic data type:

```
> breastData.sym <- classic2sym(breastData, groupby = "diagnosis")
> breastData.sym.i <- breastData.sym$intervalData
> head(breastData.sym.i[, 1:4], 5)

      radius_mean    texture_mean   perimeter_mean
B  [6.98 : 17.85]  [9.71 : 33.81] [43.79 : 114.60]
M [10.95 : 28.11] [10.38 : 39.28] [71.90 : 188.50]
            area_mean
B   [143.50 : 992.10]
M [361.60 : 2,501.00]
```

In `breastData`, the only category variable is `diagnosis`, which means the diagnosis of breast tissues (M = malignant, B = benign). We put it as an input argument in `groupby` for merging different diagnosis results, and the interval-valued data of result sets will display its factor levels in row names.

User-defined

In general, users may not always use the aggregation methods we provide, thus, besides generating a particular variable for the group, **ggESDA** facilitates the process through the min data and max data that user-defined.

For the demonstration, we will build both min data and max data using `runif`. Generate a uniform random variable to make sure that all min data are smaller than max data:

```
> minData <- runif(100, -100, -50)
> maxData <- runif(100, 50, 100)
> demoData <- data.frame(min = minData, max = maxData)
> demoData.sym <- classic2sym(demoData, groupby = "customize",
+                            minData = demoData$min,
+                            maxData = demoData$max)
> demoData.sym.i <- demoData.sym$intervalData
> as.data.frame(head(demoData.sym.i, 5))

                V1
1 [-75.85 : 63.98]
2 [-93.71 : 85.33]
3 [-64.99 : 94.69]
4 [-57.34 : 66.03]
5 [-66.02 : 50.95]
```

Then choose the `customize` argument in `groupby`, input which data are `minData` or `maxData`, and the transformation will be simply completed.

In order to simplify the process and make the preprocessing friendly, we develop these methods and let the people who want to analyze symbolic data easier. Overall, the conversion and essential concepts can be summarized in table 2.

Table 2: Summary for `classic2sym` function

`classic2sym`					
`groupby` Args.	Transformation	Data Type	Moment	Cluster Result	Require Other Args.
`kmeans`	K-means	Numeric/Category	Smaller data	V	TRUE
`hclust`	Hierarchical	Numeric/Category	Smaller data	V	FALSE
variable name	Variables aggregation	Numeric/Category	Own the pre-clustering group	-	FALSE
`customize`	User-defined	Numeric	Connect with other packages	-	TRUE

6. Conclusion and Future Development

We developed **ggESDA** to complementary the **ggESDA** package for symbolic data analysis, especially for interval-valued data. Being an extension of **ggESDA**, we inherit most of its plot-construction syntax and strict adherence to the rules governed by *The Grammar of Graphics* Wilkinson (2012). So far, we have demonstrated various newly implementations of traditional graphics to symbolic data. As an answer to the challenges of big data and complex data that the traditional data encounter, **ggESDA** reduces and summarizes by classes into a structured data table with paired symbolic-valued variables, and visualizes symbolic data which are becoming increasingly important in this area.

We are committed to maintaining and developing the **ggESDA** package in the future. Future versions of the package will contain more multivariate exploratory graphical techniques and methodological improvement, including the revisions of 3D scatter plots, interactive tools, quantile representation, and so on. In order to overcome the constraints of the coordinate system of **ggESDA** in three dimensions, we will offer new geometries as well. The availability of implementation of these would be desirable for applications.

Table 3: Fields of the built-in datasets of **ggESDA**

Built-In interval-valued data set	
Data	Reference: methods
facedata	Douzal-Chouakria *et al.* (2011); Le-Rademacher and Billard (2012): iPCA
oils	Lauro and Palumbo (2000); Lauro, Verde, and Irpino (2008): iPCA
mushroom	Neto, Cordeiro, and de Carvalho (2011); Domingues, de Souza, and Cysneiros (2010): iRegression
Cardiological (blood pressure)	Billard and Diday (2000); Xu (2010): iRegression
AbaloneIdt	Malerba, Esposito, Gioviale, and Tamma (2001): Dissimilarity measure
Environment	Umbleja *et al.* (2020): EDA

References

Bertrand P, Goupil F (2000). "Descriptive statistics for symbolic data." In *Analysis of symbolic data*, pp. 106–124. Springer.

Billard L (2007). "Dependencies and variation components of symbolic interval-valued data." In *Selected contributions in data analysis and classification*, pp. 3–12. Springer.

Billard L (2008). "Sample covariance functions for complex quantitative data." In *Proceedings of World IASC Conference, Yokohama, Japan*, pp. 157–163.

Billard L, Diday E (2000). "Regression analysis for interval-valued data." In *Data Analysis, Classification, and Related Methods*, pp. 369–374. Springer.

Billard L, Diday E (2003). "From the statistics of data to the statistics of knowledge: symbolic data analysis." *Journal of the American Statistical Association*, **98**(462), 470–487.

Billard L, Diday E (2007). *Symbolic Data Analysis: Conceptual Statistics and Data Mining.* Wiley, New Jersey.

Cazes P, Chouakria A, Diday E, Schektman Y (1997). "Extension de l'analyse en composantes principales à des données de type intervalle." *Revue de Statistique appliquée*, **45**(3), 5–24.

Cecil C Bridges J (1966). "Hierarchical Cluster Analysis." *Psychological Reports*, **18**(3), 851–854. doi:10.2466/pr0.1966.18.3.851. URL https://doi.org/10.2466/pr0.1966.18.3.851.

Chang W (2021). *R6: Encapsulated Classes with Reference Semantics.* R package version 2.5.1, URL https://CRAN.R-project.org/package=R6.

Chen CH (2002). "Generalized association plots: Information visualization via iteratively generated correlation matrices." *Statistica Sinica*, pp. 7–29.

Chen CH, Hwu HG, Jang WJ, Kao CH, Tien YJ, Tzeng S, Wu HM (2004). "Matrix visualization and information mining." In *COMPSTAT 2004—Proceedings in Computational Statistics*, pp. 85–100. Springer.

Diday, Edwin (2018). "New Advances on the Symbolic Data Analysis Framework: Basic Theory, Explanatory Criteria, Improving Machine Learning, New Directions of Research." p. 17.

Diday E, Noirhomme-Fraiture M (2008). *Symbolic data analysis and the SODAS software.* John Wiley & Sons.

Domingues MA, de Souza RM, Cysneiros FJA (2010). "A robust method for linear regression of symbolic interval data." *Pattern Recognition Letters*, **31**(13), 1991–1996.

Douzal-Chouakria A, Billard L, Diday E (2011). "Principal component analysis for interval-valued observations." *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **4**(2), 229–246.

Irpino A (2021). *HistDAWass: Histogram-Valued Data Analysis.* R package version 1.0.6, URL https://CRAN.R-project.org/package=HistDAWass.

Lauro CN, Palumbo F (2000). "Principal component analysis of interval data: a symbolic data analysis approach." *Computational statistics*, **15**(1), 73–87.

Lauro NC, Verde R, Irpino A (2008). "Principal component analysis of symbolic data described by intervals." *Symbolic Data Analysis and the SODAS Software, ed. E. Diday and M. Noirhomme-Fraiture*, pp. 279–312.

Le-Rademacher J, Billard L (2012). "Symbolic covariance principal component analysis and visualization for interval-valued data." *Journal of Computational and Graphical Statistics*, **21**(2), 413–432.

Leroy B, Chouakria A, Herlin I, Diday E (1996). "Approche géométrique et classification pour la reconnaissance de visage." In *Congrès de Reconnaissance des Formes et Intelligence Artificielle*, pp. 548–557.

MacQueen J, *et al.* (1967). "Some methods for classification and analysis of multivariate observations." In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA.

Malerba D, Esposito F, Gioviale V, Tamma V (2001). "Comparing dissimilarity measures for symbolic data analysis." *Proceedings of Exchange of Technology and Know-how and New Techniques and Technologies for Statistics*, **1**, 473–481.

Neto EdAL, Cordeiro GM, de Carvalho FdA (2011). "Bivariate symbolic regression models for interval-valued variables." *Journal of Statistical Computation and Simulation*, **81**(11), 1727–1744.

Rodriguez O (2021). *RSDA: R to Symbolic Data Analysis.* R package version 3.0.9, URL `https://CRAN.R-project.org/package=RSDA`.

Sherwani RAK, Shakeel H, Saleem M, Awan WB, Aslam M, Farooq M (2021). "A new neutrosophic sign test: An application to COVID-19 data." *Plos one*, **16**(8), e0255671.

Silva PD, Brito P (2021). *MAINT.Data: Model and Analyse Interval Data.* R package version 2.6.1, URL `https://CRAN.R-project.org/package=MAINT.Data`.

Tukey JW (1977). *Exploratory data analysis.* Addison-Wesley series in behavioral science : quantitative methods. Addison-Wesley. ISBN 0201076160. URL `https://www.worldcat.org/oclc/03058187`.

Umbleja K, Ichino M, Yaguchi H (2020). "Improving symbolic data visualization for pattern recognition and knowledge discovery." *Visual Informatics*, **4**(1), 23–31.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Wickham H, Francois R, Henry L, Muller K (2021). *dplyr: A Grammar of Data Manipulation.* R package version 1.0.7, URL `https://CRAN.R-project.org/package=dplyr`.

Wilkinson L (2012). "The grammar of graphics." In *Handbook of computational statistics*, pp. 375–414. Springer.

Xu M, Qin Z (2021). "A bivariate Bayesian method for interval-valued regression models." *Knowledge-Based Systems*, p. 107396.

Xu W (2010). *Symbolic data analysis: interval-valued data regression*. Ph.D. thesis, University of Georgia Athens, GA.

Yang Z, Lin DK, Zhang A (2019). "Interval-valued data prediction via regularized artificial neural network." *Neurocomputing*, **331**, 336–345.

Affiliation:

Firstname Lastname
Affiliation
Address, Country
E-mail: name@address
URL: http://link/to/webpage/

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd