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 developed for exploratory symbolic data analysis in R. Based on **ggplot2** Wickham (2009), the **ggESDA** package which is familiar programming structure with its parent provides a wide variety of graphical techniques such as histogram, 3D-scatterplot and radar plot. In addition, a general and customized transformation function `classic2sym()` is implemented for generating a symbolic data table from classical data frame by clustering algorithm, **RSDA** Rojas (2015) function and user-defined method. wait for edit.....

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

1. Introduction

"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)." Diday and Edwin (2018). The statement 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.

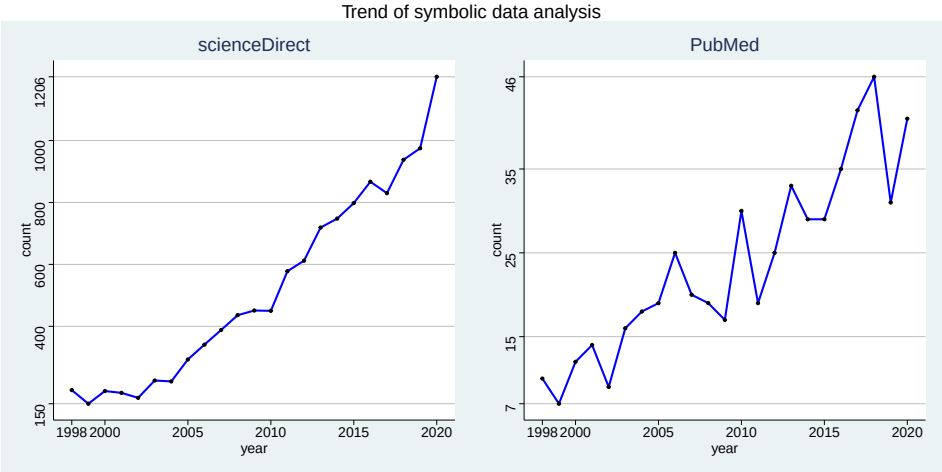

Figure 1: The number of "symbolic data analysis" or "interval-valued data" related articles in researches and applications according to PubMed and ScienceDirect online database over time from 1998 to 2020.

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

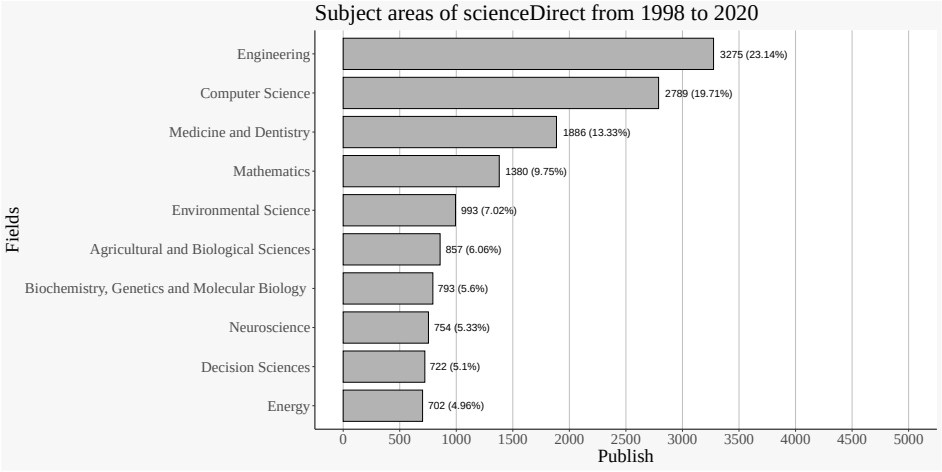

Figure 2: Top 10 researches and applications domains for SDA or interval-valued data (ScienceDirect) from 1998 to 2020

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.

2. Basic usage of ggESDA

ggESDA is now available from the Github at `https://github.com/kiangkiangkiang/ggESDA`. 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 functionalities and syntaxes about **ggESDA**.

3. why SDA plot (weakness of classical plot)

3.1. sol overstike

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 3.

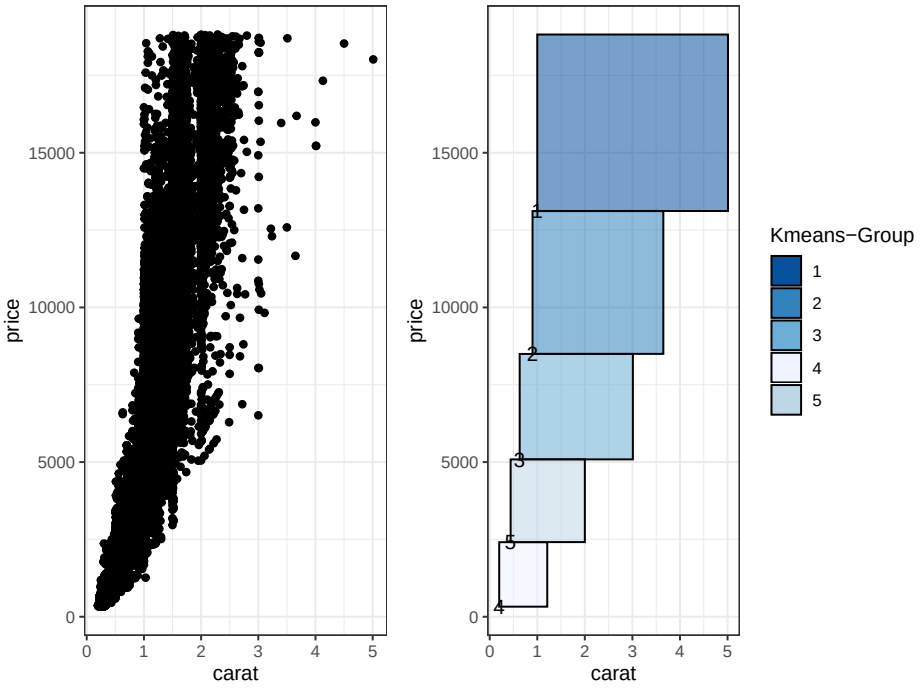

Figure 3: Compare classical data and symbolic data

In Figure 3, 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).

3.2. full information

In the past, we would like to use barplot to visualize the frequency of categorical data, but that was merely represented the distribution of full data in that category. It cannot lead researchers to explore more details in what they are interesting such as a particular part of data, so aggregation methods play a vital role to merge the data we interesting.

However, the conventional categorical data after merging will usually be represented by mode, 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. 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 that way, a categorical variable will never be shown as a single value at all, instead, a complete information histogram will be substituted.

4. classical data to symbolic data

4.1. 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"               "radius_mean"
 [4] "texture_mean"            "perimeter_mean"          "area_mean"
 [7] "smoothness_mean"         "compactness_mean"        "concavity_mean"
[10] "concave points_mean"     "symmetry_mean"           "fractal_dimension_mean"
[13] "radius_se"               "texture_se"              "perimeter_se"
[16] "area_se"                 "smoothness_se"           "compactness_se"
[19] "concavity_se"            "concave points_se"       "symmetry_se"
[22] "fractal_dimension_se"    "radius_worst"            "texture_worst"
[25] "perimeter_worst"         "area_worst"              "smoothness_worst"
[28] "compactness_worst"       "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.

4.2. 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**. 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`.

4.3. 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`.

4.4. particular variable

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           area_mean
B  [6.98 : 17.85]  [9.71 : 33.81] [43.79 : 114.60]   [143.50 : 992.10]
M [10.95 : 28.11] [10.38 : 39.28] [71.90 : 188.50] [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.

4.5. 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 1.

Table 1: Summary for `classic2sym()`

`classic2sym()`					
`groupby` Args.	Transformation	Data Type	Opportune 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

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. The following will discuss these explicitly.

5.1. 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 4.5 plays a important role. With the connection being built, it can be compatible with all other tools in R.

5.2. Example for generalization

For the demonstration, we consider two famous R packages for SDA, **HistDAWass** Irpino and Verde (2015) and **MAINT.Data** Duarte Silva and Brito (2011). Both of these make lots of contributions to the statistics of SDA, so we tend to make some analysis using these and plot the result with **ggESDA**.

HistDAWass

We use the principle in section 5.1 to process `BLOOD` data in **HistDAWass**, first. With the method **HistDAWass** provided, it will be more convenient to get min and max data:

```
> library(HistDAWass)
> # Get min and max data
> 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:

```
> library(MAINT.Data)
> #get data interval-valued data in 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 may facilitate the integration, extend utilize of **ggESDA** and generalize to all SDA studies.

6. 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 2. The number in table 2 shows how many methods are provided in its field.

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. It is a basic thinking for general radar plot, improved for distinct groups visualization by **ggESDA**, and implemented in R using **ggplot2**.

Table 2: Compare with R packages

		R(4.1.0)	Transformation	EDA			Statistic	Machine learning	
Package	Author	Available	Transform to sym	Univariate	Bivariate	Multivariate	Stat. Method	Supervised	Unsupervised
RSDA	Rojas (2015)	V	V	0	1	0	3	2*	1*
symbolicDA	Dudek (2013)	V		0	0	2*	2	1	1*
HistDAWass	Irpino (2015)	V	V	4	0	0	3	1	1*
MAINT.Data	Silva (2011)	V		0	0	0	7*	0	1*
iRegression	Neto (2011)	V		0	0	0	1	0	0
intReg	Toomet (2012)			0	0	0	1	0	0
ISDA.R	Filho (2012)			1	0	1	1	0	0
GPCSIV	Brahim (2013)			0	0	0	1	0	0
GraphPCA	Brahim (2014)			0	0	0	1	0	0
ggESDA	Jiang (2021)	V	V	8*	4*	2*	1	0	0

7. General design

The **ggESDA** object is composed of the interval-valued data, statistics data frame, clustering results, and other components from **R6** 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 4.

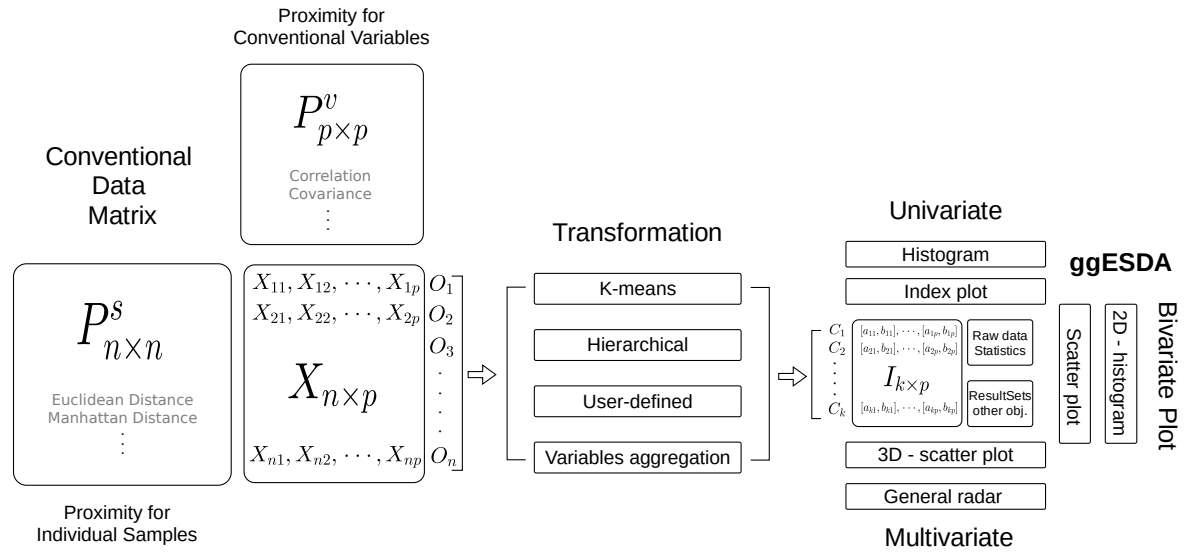

Figure 4: Package Structure and Diagram for the Transformation Flow

As illustrated in Figure 4, each row (observation) in the conventional data matrix $X_{p\times p}$ 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.

8. 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.

8.1. 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 data table 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. The realization of Z for the observation W_u is the interval $Z(W_u) = [a_u, b_u]$.

First, assume that each object is equally likely to be observed with probability $\frac{1}{m}$(m is the number of concepts), and the empirical density function of Z Bertrand and Goupil (2000) 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, and $u = 1, 2, \cdots, m$.

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 :

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

Non-equidistant Histogram

8.2. Bivariate summaries

Descriptive statistics

2D-Histogram

References

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

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

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.

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

Duarte Silva AP, Brito P (2011). "MAINT. DATA: Modeling and Analysing Interval Data in R."

Irpino A, Verde R (2015). "Basic statistics for distributional symbolic variables: a new metric-based approach." *Advances in Data Analysis and Classification*, **9**(2), 143–175.

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.

Rojas OR (2015). "R to Symbolic Data Analysis." URL https://www.imsbio.co.jp/RGM/R_rdfile?f=RSDA/man/RSDA-package.Rd&d=R_CC.

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.

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 (2009). "**ggplot2**: Elegant Graphics for Data Analysis." *Media*, **35**(211), 10–1007. doi:10.1007/978-0-387-98141-3.

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

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