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. Among ScienceDirect,

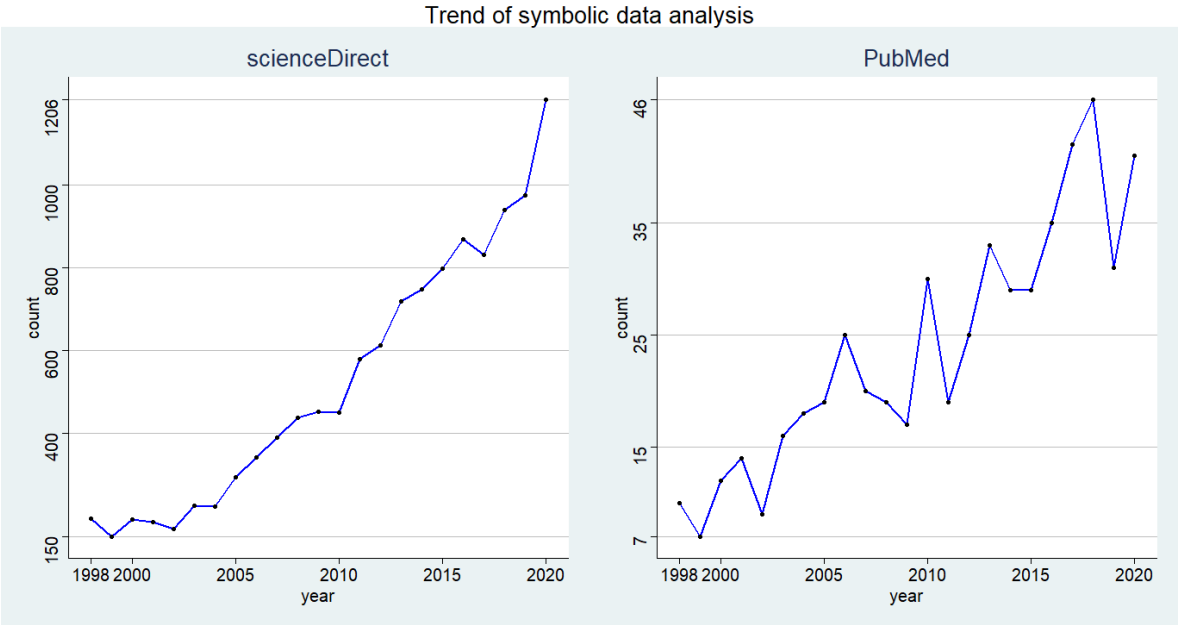

Figure 1: The number of "symbolic data analysis" or "interval-valued data" related articles in researches and applications according to PubMed and ScienceDirect online database over time from 1998 to 2020.

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

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.

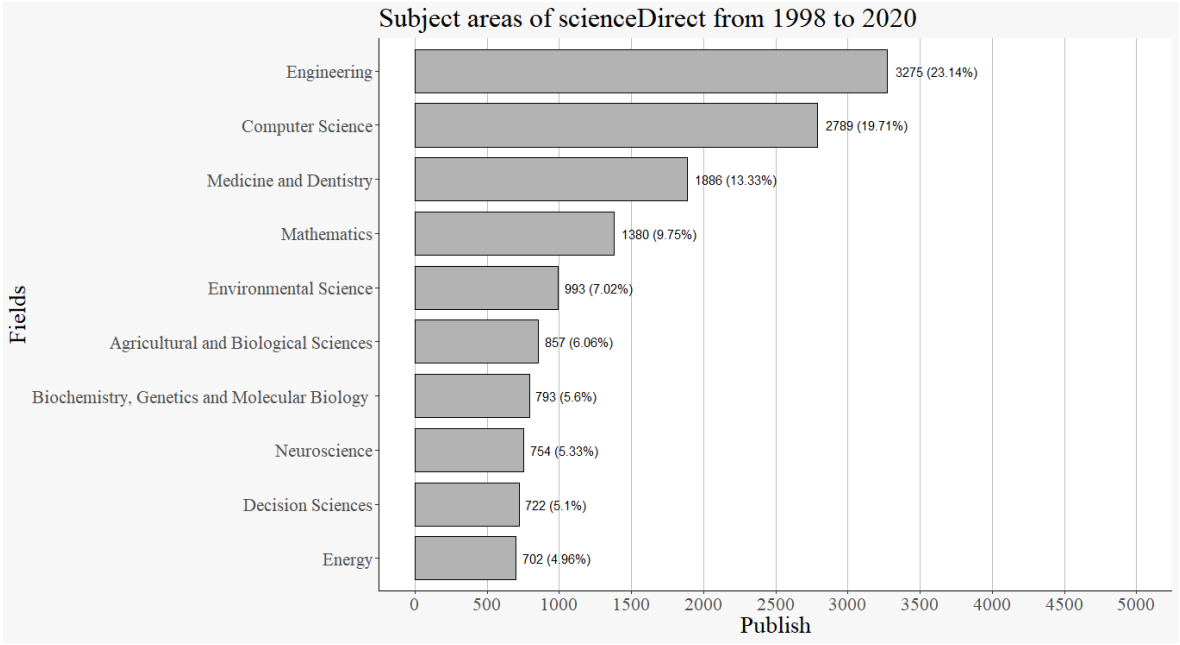

Figure 2: Top 10 researches and applications domains for SDA or interval-valued data (ScienceDirect) from 1998 to 2020

2. Prominent SDA packages

2.1. RSDA

2.2. symbolicDA

2.3. HistDAWass

3. 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**.

3.1.

3.2. General principles

3.3. Multiple plot

3.4. Package dependencies

4. Application to real datasets

4.1. univariate

4.2. bivariate

4.3. multivariate

5. Conclusion

6. why SDA plot (weakness of classical plot)

6.1. 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.

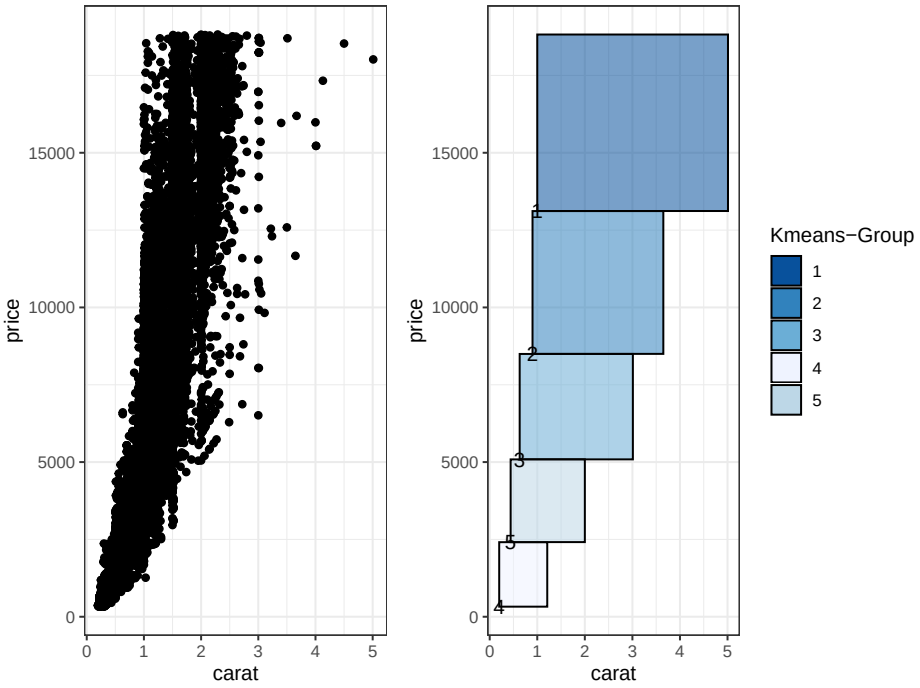

Figure 3: Compare classical data and symbolic data

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 Jin and Han (2010).

6.2. 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.

7. classical data to symbolic data

7.1. 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"
 [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.

7.2. 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`.

7.3. 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`.

7.4. 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.

7.5. 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 this method and let the people who want to analyze symbolic data easier.

References

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.

Jin X, Han J (2010). *K-Means Clustering*, pp. 563–564. Springer US, Boston, MA. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_425. URL https://doi.org/10.1007/978-0-387-30164-8_425.

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.

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/