vlda: An R package for statistical visualization of multidimensional longitudinal data

Bo-Hui Lee^a, Seongwon Ryu^b, Yong-Seok Choi^{1, b}

^aDepartment of Advertising and Public Relations, Silla University, Korea;

^bDepartment of Statistics, Pusan National University, Korea

Abstract

The **vlda** is an R (R Development Core team *et al.*, 2011) package which provides functions for visualization of multidimensional longitudinal data. In particular, the R package **vlda** was developed to assist in producing a plot that more effectively expresses changes over time for two different types (long format and wide format) and uses a consistent calling scheme for longitudinal data. The main features of this package allow us to identify the relationship between categories and objects using an indicator matrix with object information, as well as to cluster objects. The R package **vlda** can be used to understand trends in observations over time in addition to identifying relative relationships at a simple visualization level. It also offers a new interactive implementation to perform additional interpretation, therefore it is useful for longitudinal data visual analysis. Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual aspects of the VLDA plot layout. Furthermore, it allows the projection of supplementary information (supplementary objects and variables) that often occurs in longitudinal data of graphs. In this study, practical examples are provided to highlight the implemented methods of real applications.

Keywords: **vlda**, visualization, longitudinal data, R

1. Introduction

Longitudinal data is measured repeatedly at multiple time points for each object (Jeong and Choi, 2009). This data is present in several fields such as sociology, education, economics, meteorology, and especially in medicine. One of the multiple methods used to visualize longitudinal data for continuous response variables, the box plot was proposed by Everitt and Hothorn (2006). The box plot compares the change according to time points using the box plot of each time point. Another method is the growth curve introduced by Singer and Willett (2003). The growth curve is used for exploratory data analysis. It is useful for identifying patterns by mean or median over time.

For binary or categorical response variables, it is difficult to distinguish patterns by growth curve and cannot be expressed by the box plot. In this case, it can be visualized as a horizontal line plot, a correspondence analysis plot, and other such plots, as well as a dot plot and bar chart. However, there are some limitations in the existing visualization techniques of longitudinal data such as,

- They tend to include simple visualization levels or are only applicable to particular sizes.
- As the number of variables or categories increases, the techniques are complex even if they apply to all contingency table sizes.

¹ Corresponding author: Department of Statistics, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-Gu, Busan 46241, Korea. E-mail: yschoi@pusan.ac.kr

- It is difficult to visually identify the techniques at a glance and interpret the changes in time points.
- The distance between the row and column coordinates is not geometrically meaningful.
- Only variable information is utilized, as opposed to the information of objects.

The VLDA plot method (Lee, 2019) used in this study is a new method for graphical representation of multidimensional longitudinal data. This method is based on the projection method (Greenacre and Hastie, 1987; Lebart *et al.*, 1984), which obtains ideas from multiple correspondence analysis. Specifically, the corresponding analysis plot appropriately explains changes over time, as a characteristic of longitudinal data. In correspondence analysis, there are two methods to simultaneously represent row and column coordinates with different dimensions: a symmetry method of Greenacre (1984) and a projection method of Greenacre and Hastie (1987) and Lebart *et al.* (1984). Multiple correspondence analysis for the indicator matrix is already provided in the package **FactoMineR** of R, but it uses a symmetry method. In the symmetry method, the distance between the row and column coordinates is geometrically meaningless. On the other hand, the projection method attempts to reduce the dimension by projecting it onto the same unit vector on the same dimension. In this case, the projection method has three advantages,

- The distance between the row and column coordinates can be used to identify the relative relationship between the two categories.
- It has the ability to describe data in detail using distance and coordinate directions.
- The multiple correspondence analysis, which uses the indicator matrix, enables us to cluster the objects, as well as to explain the relationships between categories and objects using object information.

The VLDA plot is a visualization technique for multidimensional longitudinal data and can be viewed as an extension of correspondence analysis plots, which are based on projection methods due to these properties. When using the statistical software R language, there is no package available for creating the VLDA plot.

Therefore, to implement the existing VLDA plot method, we implemented a new interactive visualization plot in R language. This implementation is organized according to an R (R Development Core team *et al.*, 2011) package referred to as **vlda**. This package supports visualization techniques that are able to display changes more effectively in multidimensional longitudinal data over time. This is useful for visual analysis as it provides a new interactive implementation that performs additional interpretations to the existing VLDA plot. Interactivity provides a tooltip to display values when hovering over the coordinates and the ability to display different hover effects according to the two types (long format and wide format). Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual aspects of the VLDA plot layout.

This paper is organized as follows: Section 2 provides the theoretical background of this work. Section 3 simply describes the usage of the R function for package **vlda**, while Section 4 explains how the **vlda** package is implemented by using real data for further learning. Finally, Section 5 summarizes the main conclusions.

2. Background

Longitudinal data is measured repeatedly by multiple time points for each object (Jeong and Choi, 2009). This data is present in several fields such as sociology, education, economics, and meteorology, and is particularly easy to observe in medicine. For example, patients can be measured at monthly intervals to ascertain whether a drug treatment is successful. Long format presents p covariates $X_1, \dots, X_p$ representing the patient's characteristics that can affect the outcome of the treatment. Each row is one time point per object and each object has T rows. The response variable obtained over time is Y . The T values for each object are stacked—they all appear in one column. The same applies to covariates. Some covariates that do not change over time have the same value in all the rows. To keep track of which observation occurred at what time point, we add the variable time.

In the special case where the covariates do not change from time point 1 to T , the long format can be converted to the wide format. Here, $Y_1, \dots, Y_T$ are the response variables obtained at time $t (= 1, \dots, T)$. In the wide format, an object's repeated responses are indicated in a single row, with each response in a separate column. Since covariates do not change over time, they have only one value per object. However, response variables assume different values at each time point; therefore, a different variable (column) is required for each time point.

2.1. Algorithm for the VLDA plot

In this section, we provide an overview of the VLDA plot algorithm. For a detailed description of the algorithm, refer to Lee (2019). In general, the indicator matrix Z is represented by 0 and 1 in a multi-way contingency table. The indicator matrix accommodates both the long format and the wide format. Also, the indicator matrix can easily represent multidimensional longitudinal data considering both objects and variables.

Consider a long format consisting of p covariates, one response, and one-time variables for each of the n objects. When the number of categories that appear at least once for the variable $k (= 1, \dots, p+2)$ in the dataset is q_k , it can be expressed as a $nT \times q_k$ matrix Z_k denoted by 0 and 1 for objects, which is referred to as a partition design matrix. Next, consider a wide format consisting of p covariates and T response variables for each of the n objects. Here, it becomes $k = 1, \dots, p + T$ and Z_k is considered a $n \times q_k$ matrix.

[Algorithm 2.1] The algorithm for the VLDA plot based on the projection method using the indicator matrix as proposed by Lee (2019)

Step 1 : Indicator matrix

$$Z = [Z_1 | Z_2 | \dots | Z_K] = (z_{ij}), \quad i = 1, \dots, N; j = 1, \dots, q,$$

where Z_k is $N \times q_k$, $k = 1, \dots, K$, $q = q_1 + q_2 + \dots + q_K$. $N = nT$ and $K = p + 2$ in the long format, and $N = n$ and $K = p + T$ in the wide format.

Step 2 : Relative frequency data matrix

$$S = (s_{ij}), \quad s_{ij} = \frac{1}{NK} \times z_{ij}, \quad i = 1, \dots, N; j = 1, \dots, q,$$

where $NK = \sum_{i=1}^N \sum_{j=1}^q z_{ij}$ is the total sum of elements of Z .

Step 3 : Singular value decomposition

$$G = D_r^{-\frac{1}{2}} S D_c^{-\frac{1}{2}} = P \Lambda Q',$$

where $D_r = \text{diag}(s_{1.}, \dots, s_{N.})$; $s_{i.} = \sum_{j=1}^q s_{ij} = 1/N$ are row marginal proportions, $D_c = \text{diag}(s_{.1}, \dots, s_{.q})$, $s_{.j} = \sum_{i=1}^N s_{ij}$ are column marginal proportions, $\Lambda = \text{diag}(1, \lambda_1, \dots, \lambda_{q-1})$ is a diagonal matrix by singular values, and P and Q satisfy $P'P = Q'Q = I_q$.

Step 4 : Row and column coordinates of the VLDA plot

$$A = D_r^{-\frac{1}{2}} P \Lambda, \quad B = D_c^{-\frac{1}{2}} Q,$$

and the row and column coordinates of the $s(\geq 2)$ -dimensional VLDA plot are

$$A_{(s)} = D_{r(s)}^{-\frac{1}{2}} P_{(s)} \Lambda_{(s)}, \quad B_{(s)} = D_{c(s)}^{-\frac{1}{2}} Q_{(s)}.$$

Step 5 : Goodness-of-fit of the approximation for the $s(\geq 2)$ -dimensional VLDA plot

$$\text{fit} = \frac{\sum_{j=1}^s \lambda_j^2}{\sum_{j=1}^{q-1} \lambda_j^2}.$$

We note that Step 3 in Algorithm 2.1 can be described as follows, the i^{th} row profile vector from the relative frequency data matrix of Step 2 is defined as,

$$\mathbf{r}_i = \frac{1}{s_{i.}} (s_{i1}, \dots, s_{iq})', \quad i = 1, \dots, N,$$

where $s_{i.} = \sum_{j=1}^q s_{ij} = 1/N$ and $\mathbf{r}_i$ have a value of 0 or $1/K$, which contains information about the objects. We can consider $\mathbf{r}_i^*$, which is the projection of the i^{th} row profile vector $\mathbf{r}_i$ onto an arbitrary unit vector $\mathbf{m}$. Then, we minimize the weighted squared distance between $\mathbf{r}_i$ and $\mathbf{r}_i^*$ by,

$$\sum_{i=1}^N \|\mathbf{r}_i - \mathbf{r}_i^*\|_{D_c^{-1}}^2 = \sum_{i=1}^N \|\mathbf{r}_i\|_{D_c^{-1}}^2 - \sum_{i=1}^N \|\mathbf{r}_i^*\|_{D_c^{-1}}^2,$$

where $\|\mathbf{r}_i\|_{D_c^{-1}} = (\mathbf{r}_i' D_c^{-1} \mathbf{r}_i)^{1/2}$. The weighted distance minimized is equivalent to maximizing $\sum_{i=1}^N \|\mathbf{r}_i^*\|_{D_c^{-1}}^2 = (D_r^{-1} S D_c^{-1} \mathbf{m})' (D_r^{-1} S D_c^{-1} \mathbf{m})$. With a constraint of $\mathbf{m}' D_c^{-1} \mathbf{m} = 1$, the objective function with Lagrange function can be defined as,

$$L = (D_r^{-1} S D_c^{-1} \mathbf{m})' D_r (D_r^{-1} S D_c^{-1} \mathbf{m}) - \mathbf{v}(\mathbf{m}' D_c^{-1} \mathbf{m} - 1).$$

The solution is $(D_c^{-1/2} S' D_r^{-1/2}) (D_r^{-1/2} S D_c^{-1/2}) (D_c^{-1/2} \mathbf{m}) = \mathbf{v} (D_c^{-1/2} \mathbf{m})$, let $G = D_r^{-1/2} S D_c^{-1/2}$, $\mathbf{q} = D_c^{-1/2} \mathbf{m}$.

2.2. The algorithm for supplementary objects and variables in the VLDA plot

Longitudinal data inevitably displays the characteristic of added supplementary data as per the following example,

- Outcome variables measured at additional time points, such as $T + 1, T + 2, \dots$ after the last time point T .
- When new objects are added that have not been measured previously.
- Other covariates that indicate the characteristics of objects.

Therefore, in this section, we review an algorithm for coordinates that represent objects and variables added in the VLDA plot already provided. This algorithm is proposed by Lee (2019).

Suppose that the $N \times q$ indicator matrix Z has added n_+ objects (rows) and variables with q_+ categories (columns). Here, Z_R is a supplementary row indicator matrix of size $n_+ \times q$ and Z_C is a supplementary column indicator matrix of size $N \times q_+$.

First, N row profile vectors $\mathbf{r}_1, \dots, \mathbf{r}_N$ and q column profile vectors $\mathbf{c}_1, \dots, \mathbf{c}_q$ consist of row and column profile matrices as follows,

$$R = D_r^{-1}S = (\mathbf{r}_1, \dots, \mathbf{r}_N)', \quad C = D_c^{-1}S' = (\mathbf{c}_1, \dots, \mathbf{c}_q)'. \quad (2.1)$$

Similar to (2.1), supplementary row and column profile matrices are obtained as follows.

$$R_+ = \text{diag}(Z_R \mathbf{1}_q)^{-1} Z_R, \quad C_+ = \text{diag}(Z_C' \mathbf{1}_N)^{-1} Z_C'. \quad (2.2)$$

Here, $\text{diag}(\text{vector})^{-1}$ indicates that a diagonal matrix with a vector should be developed, while locating the inverse matrix.

Next, recall the row and column coordinates $A = D_r^{-1/2} P \Lambda$ and $B = D_c^{-1/2} Q$ obtained in Step 4 of Algorithm 2.1. Here, A and B satisfy the singular value decomposition for G in Step 3 of Algorithm 2.1 with the conditions of $P'P = Q'Q = I_q$. According to these conditions and (2.1), the row and column coordinates are,

$$A = D_r^{-\frac{1}{2}} P \Lambda = D_r^{-1} S D_c^{-\frac{1}{2}} Q = R D_c^{-\frac{1}{2}} Q, \quad (2.3)$$

and

$$B = D_c^{-\frac{1}{2}} Q = D_c^{-1} Q \Lambda P' P \Lambda^{-1} = C D_r^{-1} P \Lambda^{-1}. \quad (2.4)$$

Similarly, by using this property we can obtain low-dimensional coordinates of the supplementary row and column profile matrices in (2.2). If these are called A_+ and B_+ respectively, the following is obtained from (2.3) and (2.4),

$$A_+ = R_+ D_c^{-\frac{1}{2}} Q, \quad B_+ = C_+ D_r^{-1} P \Lambda^{-1}. \quad (2.5)$$

Thus, the coordinates in the s -dimensional VLDA plot for supplementary objects and variables are provided by a sub-matrix

$$A_{+(s)} = R_{+(s)} D_{c(s)}^{-\frac{1}{2}} Q_{(s)}, \quad B_{+(s)} = C_{+(s)} D_{r(s)}^{-1} P_{(s)} \Lambda_{(s)}^{-1}, \quad (2.6)$$

consisting of s columns of A_+ and B_+ in (2.5).

3. The **vlda** R package

vlda is an R package that is freely available from the comprehensive R archive network (CRAN) (R Development Core team *et al.*, 2011). It is designed to visualize multidimensional longitudinal data and can be installed on any operating system installed with R. **vlda** is the abbreviation of “visualization of multidimensional longitudinal data”. Users can install the released version from CRAN with: `install.packages("vlda")`. This package has been tested in 32- and 64-bit R versions 3.6.3 on 32- and 64-bit Windows operating systems. The **vlda** has the following characteristics,

- It supports visualization technology that can display changes over time more effectively.
- It is convenient to plot by using a consistent calling method for the two types of longitudinal data.
- Additional analyses can be performed by providing a new interactive implementation of the existing VLDA plot.
- Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual aspects of the VLDA plot layout.
- Two coordinates are used to identify the relative relationship between the two categories using distance.
- It identifies the relationship between categories and objects by using the indicator matrix with the information of objects, and the clustering of objects is also made possible.
- It has the ability to project supplementary information (supplementary objects and variables) onto a graph.

We outline the **vlda** organization below, followed by a detailed example demonstrating some functionalities.

3.1. The **vlda** function

The **vlda** function is used to extract summary variables, such as the row coordinate, column coordinate, eigenvalue, and goodness-of-fit according to Algorithm 2.1. The **vlda** package has a built-in **indicator** function that converts longitudinal data into an indicator matrix. This function corresponds to Step 1 in Algorithm 2.1 within the **vlda** function, which is the most important step to be implemented. The function is typically used to implement algorithms, but can also be used to generate an indicator matrix for factors. The **indicator** function accepts the same conditions as argument **x** of the **vlda** function. The first argument **x** is accepted in a data frame format consisting of categorical data. A more detailed implementation of the **indicator** function can be found in the commands `R> ?indicator` which will display the documentation. Argument **x** must contain individual and time point variables that repeatedly measure the same object via multiple time points. This is accepted in the **time** and **object** arguments and is used to track observations that occur at some point. The **type** argument can adopt a long or wide format depending on the type of data. Refer to Table 1 for a detailed description of the arguments. In the called in the form is as follows,

```
vlda(x, object, time, type = c("long", "wide"))
```

Table 1: Arguments and descriptions in the vlda function

Arguments	Description
x	A data frame consisting of categorical data coded in numbers. The samples (object) should have been measured repeatedly by multiple time points; its variables will be represented as variable coordinates. To keep track of which observation occurred at which time point, the Time variable should be included.
object	A vector of length n samples. The object that would have performed the measure repeatedly by multiple time points; the object is indicated according to the name of the observation coordinate. Accepts a character string that denotes the name of the object variable.
time	A time point of longitudinal data. Accepts a character string that denotes the name of the time variable.
type	A type of longitudinal data. Long format refers to each row that equals one time point per object, so each object has T rows. All T values for each object are stacked-they are all in one column; wide format refers to an object repeating responses in a single row, and each response is in a separate column. So $(Y_1, \dots, Y_T)$ are the response variables obtained at time $t (= 1, \dots, T)$. type = c(long, wide).

3.2. The vlda plot function

The `vlda_plot` function recognizes `vlda` objects as a `fit` argument, and plots them accordingly by interpreting the `obs.coordinate` as row coordinates and the `var.coordinate` as column coordinates. The function based on the packages **ggplot2** (Wickham, 2011) and **ggiraph** (Gohel, 2016), produces two-dimensional interactive graphics of high-quality (Figure 2 and Figure 7). The function is useful for visual analysis as it provides a new interactive implementation to perform interpretations in addition to the existing VLDA plot. Interactivity provides a tooltip (Figure 3) to display values when hovering over the coordinates and the ability to display different hover effects (Figure 4 and Figure 8) according to the two types (long format and wide format). Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual aspects of the VLDA plot layout. In addition, it allows the ability to project supplementary information (supplementary objects and variables) onto a graph (Figure 6 and Figure 9). Also, several plotting options for graphic parameters (see Table 2) have been implemented to create elegant graphics (Figure 9). In the called in the form is as follows,

```
vlda_plot(fit, rename = NULL, interactive = TRUE,
          title = NULL, title.col = NULL, title.size = 15, title.hjust = 0,
          subtitle = NULL, sub.col = NULL, sub.size = 15, sub.hjust = 0,
          labels = NULL, lab.col = NULL, lab.size = NULL, lab.face = NULL,
          legend.position = "bottom", legend.justification = NULL,
          linetype = 2, line.col = "red", font.size = 1.0, var.size = 2.5,
          obs.col = "darkgray", obs.size = 2.5, add.obs.col = "#666666",
          arrow.col = "orange", arrow.size = 0.5, arrow.type = "closed")
```

3.3. The vlda add function

The `vlda_add` function can be used to extract supplementary coordinates, such as those which represent objects and variables added in the VLDA plot already provided. This function is based on the algorithm in Section 2.2. It recognizes the `vlda` object as a `fit` argument and accepts supplemental data in the form of an indicator matrix. The required secondary arguments are `add.col` or `add.row`, and an indicator matrix, which measured additional time points or new objects not previously measured, can be added to the rows or columns. If supplemental data added contain a time variable, the `time.name` argument requires a character string that denotes the name of the time variable. If the user does not input the correct time variable in the `time.name` argument, an error will occur, indicating that is not recognized as a time variable. The variable name of the indicator matrix approved as the

Table 2: Arguments and descriptions in the `vlda_plot` function

Arguments	Description
<code>fit</code>	An object returned by <code>vlda</code> or <code>vlda_add</code> .
<code>rename</code>	Rename a variable.
<code>interactive</code>	Use the interactive graphical elements (default TRUE).
<code>title</code>	Plot title. If NULL, the title is not shown (default NULL).
<code>title.col</code>	Title color (default color is black).
<code>title.size</code>	Title font size (default size = 15).
<code>title.hjust</code>	Alignment of title (Number from 0 [left] to 1 [right]: left-aligned by default).
<code>subtitle</code>	Subtitle for the plot which will be displayed below the title.
<code>sub.col</code>	Subtitle color (default color is black).
<code>sub.size</code>	Subtitle font size (default size = 15).
<code>sub.hjust</code>	Alignment of sub-title (Number from 0 [left] to 1 [right]: left-aligned by default).
<code>labels</code>	Legend labels.
<code>lab.col</code>	Legend labels color.
<code>lab.size</code>	Legend labels size.
<code>lab.face</code>	Legend labels font c("plain", "bold", "italic", "bold.italic") default = "plain".
<code>legend.position</code>	The position of legends ("none", "left", "right", "bottom", "top", or two-element numeric vector) default is "bottom".
<code>legend.justification</code>	Anchor point for positioning the legend inside the plot "center" or two-element numeric vector) or the justification according to the plot area when positioned outside the plot.
<code>linetype</code>	Line types can be specified as: An integer or name: 0 = blank, 1 = solid, 2 = dashed, 3 = dotted, 4 = dotdash, 5 = longdash, 6 = twodash, as shown below:
<code>line.col</code>	Axis line color.
<code>font.size</code>	Font size (left-aligned by default size = 1.0).
<code>var.size</code>	Variable coordinate point size of plot.
<code>obs.col</code>	Observation coordinate point color of plot.
<code>obs.size</code>	Observation coordinate point size on plot.
<code>add.obs.col</code>	Color of added observation coordinate points.
<code>arrow.col</code>	Arrow color (default color = "orange").
<code>arrow.size</code>	Arrow size (default size = 0.5).
<code>arrow.type</code>	One of "open" or "closed" indicating whether the arrowhead should be a closed triangle.

Table 3: Arguments and descriptions in the `vlda_add` function

Arguments	Description
<code>fit</code>	An object returned by <code>vlda</code> .
<code>add.col</code>	A data matrix, the type of indicator matrix. Additional data sets in column format. $p \geq 2$.
<code>add.row</code>	A data matrix, the type of indicator matrix. Additional data sets in row format. Supplemental data should have the same variable name as <code>var.coordinate</code> returned by <code>vlda</code> , and if it is not an indicator matrix, it can be used after generating an indicator matrix using the <code>indicator</code> function built into <code>vlda</code> .
<code>time.name</code>	If the supplemental data to be added contains a time variable, enter the name of the time variable.

`add.row` argument must be the same as the `var.coordinate`. Otherwise, unintended errors may occur. Refer to Table 3 for a description of the arguments.

After the coordinates of the supplementary objects and variables are found, the `vlda_plot` function recognizes `fit` objects and plots the raw coordinates in the color gray and the supplementary coordinates in a dark gray color. In the called in the form is as follows,

```
vlda_add(fit, add.col = NULL, add.row = NULL, time.name = NULL)
```

Figure 1 shows an overview of the basic steps in the data visualization pipeline provided by the **vlda** package for the visualization of longitudinal data. Figure 1 can be summarized as follows,

- Step 1: The `vlda` function accepted longitudinal data and extracts summary variables such as `row coordinates`, `column coordinates`, `eigenvalues`, and `goodness of fit`.

Figure 1: Overview of the basic steps of the data visualization pipeline offered by the **vlda** package for the visualization of multidimensional longitudinal data.

- Step 2: The `vlda_plot` function recognizes the `vlda` object as a `fit` argument and plots them.
- Step 3: `vlda_add` function extract supplementary coordinates, such as coordinates representing objects and variables that are added in the VLDA plot already provided.
- Step 4: The supplementary coordinates found in Step 3, are accepted as a `vlda_plot` function and plotted by projecting supplementary coordinates onto the existing plot.

In Section 4, practical examples are provided to highlight the implemented methods of real applications.

4. Example

4.1. Long format of PTSD data

We illustrate the capabilities of the **vlda** package by using data related to post-traumatic stress disorder (PTSD) collected by Allison (2001). The data was measured on 316 patients who survived a fire at three, six, and twelve months after the incident (i.e., each row contains one time point per object and each patient has three rows). The dataset consists of the outcome variable (PTSD; Yes or No) and four different variables with either real or integer values as follows: control (self-control), problems (the number of life problems), stress (the number of stress-related events), and cohesion (family cohesion). Here, we focus on PTSD levels, which is considered one of the most important levels associated with PTSD progression.

```

R> library(vlda)
R> data("PTSD")
R> str(PTSD)
'data.frame': 948 obs. of 7 variables:
 $ subject : Factor w/ 316 levels "15","18","19",...: 1 1 1 2 2 2 3 3 3 4 ...
 $ control : num 3.22 3.17 3.28 2.56 3.44 ...
 $ problems: num 5.62 5.38 3.75 9.25 4.38 ...
 $ stress : int 1 0 1 0 0 0 1 1 1 0 ...
 $ cohesion: int 8 8 8 8 8 8 7 7 7 8 ...
 $ time : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3 1 2 3 1 ...
 $ ptsd : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 2 2 1 1 ...

```

The primary question in this trial is whether or not PTSD occurs over time according to the conditions of control, problems, stress, and cohesion, which are covariate variables. We start loading **vlda** packages and PTSD data, and transform into status for application to the **vlda** function, namely, categorical variables to continuous variables: problems and stress binarize into upper and lower levels based on 3 and cohesion also binarize based on 6. Let us assume that control (low, high), problems (low, high), stress (low, high), cohesion (low, high), and time (3 months, 6 months, and 12 months) are explanatory variables and PTSD (No, Yes) is a response variable.

The below code fits **vlda** to the PTSD data, and prints a summary of row coordinates, column coordinates, eigenvalues, and goodness-of-fit. To simplify, it is called in the form **vlda(x, object, time, type = c("long", "wide"))**.

```

R> PTSD[,2:4] <- apply(PTSD[,2:4], 2, function(x) ifelse(x >= 3, 1, 0))
R> PTSD[,5] <- ifelse(PTSD[,5] >= 6, 1, 0)
R> PTSD <- data.frame(lapply(PTSD, function(x) as.factor(x)))
R> fit <- vlda(x = PTSD, object = "subject", time = "time", type = "long")
R> fit

```

```

$obs.coordinate
# A tibble: 57 x 3
# Groups:   x [57]
      x      y obs_list
  <dbl> <dbl> <named list>
1 -1.49 -1.20 <chr [2]>
2 -1.26 -0.283 <chr [21]>
3 -1.03 -1.14 <chr [5]>
4 -1.03 -1.01 <chr [1]>
5 -1.02 -0.379 <chr [13]>
6 -0.92 0.233 <chr [30]>
7 -0.916 -0.784 <chr [2]>
8 -0.858 -0.808 <chr [4]>
9 -0.801 -0.215 <chr [11]>
10 -0.799 -0.088 <chr [7]>
# ... with 47 more rows

```

```

$var.coordinate
      x      y
control.0 -2.215 -0.326
control.1  0.530  0.078
problems.0  2.549 -1.702
problems.1 -0.589  0.393
stress.0    0.081  0.326
stress.1  -1.292 -5.200
cohesion.0 -1.658 -2.528
cohesion.1  0.373  0.568
time.1     -1.270  1.243
time.2      0.132  0.667
time.3      1.138 -1.910
ptsd.0      0.855  0.362
ptsd.1     -1.902 -0.805

$Eigen
  Eigenvalue Percent Cumulative
1      0.289  24.8 %    24.8 %
2      0.184  15.8 %    40.6 %
3      0.169  14.5 %    55.1 %
4      0.165  14.1 %    69.2 %
5      0.133  11.4 %    80.6 %
6      0.116   9.9 %    90.5 %
7      0.110   9.5 %   100 %

$GOF
[1] "Goodness of fit : 40.6 %"

```

As a result of `vlada`, `obs.coordinates`, `var.coordinates`, `Eigen`, and `GOF` are returned. The row and column coordinates are represented by `obs.coordinate` and `var.coordinate` respectively in Step 4 of the VLDA plot algorithm. The unique coordinates for the 948() rows are represented by `obs.coordinate`. This appears as often as the number of combinations of categories for each variable, and the observations corresponding to each row are included in the `obs_list` variable in the form "observation_timepoint". For example, the observations for the first-row coordinates "x = -1.49, y = -1.20" are,

```

R> fit$obs.coordinate$obs_list
$ "x = -1.488 y = -1.204"
[1] "127_1" "350_1"

$ "x = -1.259 y = -0.283"
[1] "57_1" "69_1" "71_1" "115_1" "141_1" "150_1" "155_1" "162_1" "178_1"
[10] "190_1" "244_1" "273_1" "284_1" "319_1" "331_1" "383_1" "393_1" "447_1"
[19] "523_1" "529_1" "566_1"
# ... with 46 more rows

```

Figure 2: VLDA plot of PTSD data by applying the `vlada`.

This means that patients 127 and 350 are located at that coordinate three months after the fire. `Eigen` summarizes the principal inertias as a result of applying the VLDA plot using the indicator matrix. The number of non-zero principal inertias derives from the total number of categories minus the number of categorical variables. GOF appears as goodness-of-fit of the two-dimensional VLDA plot. Subsequently, a two-dimensional VLDA plot using the `vlada_plot` function is created. This function accepts the value returned by `vlada` as the main argument. To do so, the following code is used,

```
R> vlada_plot(fit)
```

- control (control.0 and control.1), problems (problems.0 and problems.1), stress (stress.0 and stress.1), and cohesion (cohesion.0 and cohesion.1), where 0 and 1 denote low and high respectively.
- Time, where time.1, time.2, and time.3 denote three, six, and twelve months respectively.
- PTSD, where ptsd.0 denotes that PTSD does not appear, and ptsd.1 denotes that PTSD appears.

In Figure 2, a total of 948 rows for 316 objects are grayed out at three time points. Gray dots indicate the number of category combinations for each variable. This refers to each row of `obs.coordinate`. The `vlada_plot` function is implemented as an interactive graphic by default. This interactive figure can be displayed in the RStudio's viewer pane. The viewer pane is an interactive version with tooltips and hover effects for performing additional analysis. Figures 3 and 4 illustrate tooltips and hover effects in an interactive graph.

Tooltip allows users to identify the time point and observations associated with the coordinate when the mouse hovers over the coordinates. The function visually displays the above-mentioned

Figure 3: Interactive graphics of PTSD data.

Figure 4: Observation trends of PTSD data over time.

obs_list without any special call in the graph, and distinguishes between observations of specific coordinates. The hover effect, which is another feature, is displayed in yellow alongside the coordinates with the same time point as the coordinates that are hovered over. For example, Figure 4 shows 3 months, 6 months, and 12 months, respectively, in order from left to right, using the hover effect. The yellow arrow that connects each time point represents the trend of change over time. The yellow coordinate points indicate the tendency to move in the same direction as the arrow over time and are placed further to the right of the first axis. This means that the proportion of PTSD decreases over time.

The VLDA plot is used to understand trends in observations over time in addition to identifying relative relationships at a simple visualization level. Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual

Figure 5: Changing the theme and palette.

aspects of the VLDA plot layout.

In the VLDA plot, each variable is separated according to a different color, and each category included in the variable is displayed in the same color. The default passed to the color argument in **vlda** is the **D3** palette, which offers a sequential color palette and a good balance of these properties. Users can set the **interactive** argument to **FALSE** for an ideal palette and theme. The **ggsci** (Xiao, 2018) and **ggplot2** packages, respectively, are helpful divergent palettes and themes that are available in the package.

```
R> G <- vlda_plot(fit, interactive = FALSE)
R> G + scale_color_discrete() + theme_grey()
```

Here, we explain the geometric interpretation of the VLDA plot. Coordinates in opposite directions on each axis can be considered separate groups. If the distance between the coordinates is close, it indicates that the group has a similar tendency. Even if the explanatory variable is not significant, a small tendency can be confirmed because the coordinate is placed in consideration to the relative influence.

As shown in Figure 2, **control.0**, **problem.1**, **stress.1**, **cohesion.0**, **time.1**, and **ptsd.1** are placed on the left side of the first axis, indicating homogeneity. This indicates that patients have low self-control and family cohesion, and have experienced many life problems and stress-related events, which caused PTSD after 3 months. In contrast, **control.1**, **problem.0**, **stress.0**, **cohesion.1**, **time.2**, **time.3**, and **ptsd.0** are placed on the right side of the first axis. Therefore, there was no sign of PTSD after 6–12 months because patients had greater self-control and family cohesion, fewer life problems, and fewer stress-related events.

Subsequently, we can consider supplementary data as per the characteristic of longitudinal data. **PTSD_row** refers to control, problems, stress, and cohesion; PTSD would add rows for 316 patients after 18 months. **PTSD_column** is the degree of alcohol consumed (low, high) that may affect PTSD, which can be added to the columns that correspond from the first to the third time point for 316 patients.

```

R> data(PTSD_row)
R> data(PTSD_column)
R> str(PTSD_row)
'data.frame': 316 obs. of 13 variables:
 $ control.0 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ control.1 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ problems.0: int  1 1 1 1 1 1 1 1 0 1 ...
 $ problems.1: int  0 0 0 0 0 0 0 0 1 0 ...
 $ stress.0  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stress.1  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ cohesion.0: int  0 0 0 0 0 0 0 0 0 0 ...
 $ cohesion.1: int  1 1 1 1 1 1 1 1 1 1 ...
 $ time.1    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ time.2    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ time.3    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ptsd.0    : int  1 1 1 1 1 1 1 1 1 1 ...
 $ ptsd.1    : int  0 0 0 0 0 0 0 0 0 0 ...

R> str(PTSD_column)
'data.frame': 948 obs. of 2 variables:
 $ Drinking.0: int  1 1 1 0 1 1 0 0 0 1 ...
 $ Drinking.1: int  0 0 0 1 0 0 1 1 1 0 ...

R> fit2 <- vlada_add(fit, add.row = PTSD_row, add.col = PTSD_column)
R> vlada_plot(fit2))

```

- control (control.0 and control.1), problems (problems.0 and problems.1), stress (stress.0 and stress.1), cohesion (cohesion.0 and cohesion.1), and Drinking (Drinking.0 and Drinking.1), where 0 and 1 denote low and high respectively.
- Time, where time.1, time.2, time.3, and time.4 denote three, six, twelve, and eighteen months respectively.
- PTSD, where ptsd.0 denotes that PTSD does not appear, and ptsd.1 denotes that PTSD appears.

Figure 6 shows an added drinking column parameter and supplementary coordinates for 316 patients of the fourth time point in the VLDA plot. In the supplementary data, the number of categorical combinations of each variable has a total of 15 possibilities for the added fourth time point of the existing 316 patients. Therefore, Figure 6 shows 15 supplementary coordinates in dark grey. The center of the supplementary coordinates is placed on the right side of the first axis. Also, we observe that it draws nearer to ptsd.0 at the fourth time point than at the third time point.

That is, after 18 months, we know that the ratio of PTSD decreases over time, and that the same interpretation can be made after 6–12 months. Drinking.1 is located on the left side of the first axis, indicating homogeneity with control.0, problems.1, stress.1, cohesion.0, time.1, and ptsd.1. This shows that patients have low self-control and family cohesion, and experience many life problems and stress-related events, as well as heavy drinking, which caused an increase in PTSD after 3 months. On the contrary, Drinking.0 is located on the right side, which displays similarities with control.1,

Figure 6: Coordinates for supplementary objects and variables added to Figure 2.

problems.0, stress.0, cohesion.1, time.2, time.3, and ptsd.0. In other words, after 6–12 months, self-control and family cohesion increased, and the number of life problems, stress-related events, and drinking decreased, and thus, showed no sign of PTSD.

4.2. Wide format of depression data

In this section, we illustrate the capability of how to apply the package **vlda** in other types of wide formats, and how to plot a more complete graph fit suitable to specifying graphic element arguments of the `vlda_plot` function.

In depression data, two drugs that treat patients suffering from depression are compared. This data has modified some of the data collected by Koch *et al.* (1977). According to the initial severity of depression, a total of 800 patients were assigned to two groups: mild or severe. In each group, patients were randomly assigned to either standard or new drugs. After one, two, and four weeks of treatment, the degree of depression in each patient was classified as normal or abnormal.

```
R> data(Depression)
R> str(Depression)
'data.frame': 800 obs. of 6 variables:
 $ Case      : Factor w/ 800 levels "1","10","100",...: 1 112 223 334 ...
 $ Diagnosis : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
 $ Drug      : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ 1week     : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ 2weeks    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
 $ 4weeks    : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...
```

The focus is on drug levels, which is considered one of the most important treatments associated

Figure 7: VLDA plot of depression data.

with depression. The key question in this trial is whether the new drug treatment group indicates a significantly more effective treatment of depression than the standard drug treatment group. Therefore, we consider whether the initial depression and drug levels affect drug treatment over time. In the call to `vlda` argument, `time` accepts 1week, 2weeks, and 4weeks in a character string that denotes the name of the time variable.

```
R> wide.fit <- vlda(x = Depression, object = "Case",
+   time = c("1week", "2weeks", "4weeks"), type = "wide")
R> vlda_plot(wide.fit)
```

According to the depression data, the number of categorical combinations for each variable of 800 objects has a total of 32 unique possible combinations, and 32 observed coordinate points are grayed out. The yellow arrows that are connected between the same categories denote the trend of changes over time. In Figure 7, `Drug.1`, `Diagnosis.1`, and Normal (`1week.2`, `2week.2`, and `4weeks.2`) are placed on the left side of the first axis, indicating homogeneity. For example, patient 201 has the values of 1 in mild (`Diagnosis.1`), new drug (`Drug.1`), and normal (`1week.2`, `2weeks.2`, and `4weeks.2`); therefore, it is placed in the second quadrant.

```
R> Depression[201,]
      Case Diagnosis Drug 1week 2weeks 4weeks
201  201         1     1     2     2     2
```

In contrast, `Drug.2`, `Diagnosis.2`, and abnormal (`1week.1`, `2weeks.1`, and `4weeks.1`) are placed on the right side of the first axis. For example, patient 600 has the value of 1 in severe (`Diagnosis.2`),

Figure 8: Comparison of the observation coordinate points according to the variable combination of the covariate in depression data.

standard drug (Drug.2), and abnormal (1week.1, 2weeks.1, and 4weeks.1), so it is placed in the fourth quadrant.

```
R> Depression[600,]
      Case Diagnosis Drug 1week 2weeks 4weeks
600    600         2      2      1         1      1
```

Therefore, patients with mild initial depression (Diagnosis.1), who take a new drug (Drug.1), are more likely to be normal, and patients with severe initial depression (Diagnosis.2), who take a standard drug (Drug.2), are more likely to be abnormal. Also, regardless of the drug taken after one week, mild (Diagnosis.1) is close to normal (1week.2) and severe (Diagnosis.2) is close to abnormal (1week.1). This means that the drug has no effect after one week. However, as time passes, normal (1week.2 → 2weeks.2 → 4weeks.2) draws nearer to the new drug (Drug.1) and abnormal (1week.1 → 2week.1 → 4week.1) draws nearer to the standard drug (Drug.2). That is, the effect of the drugs is visible over time. In this way, it is possible to relate variables and objects, and to cluster objects by utilizing object information. Also, the VLDA plot shows that it is more suitable to the visualization of longitudinal data by dynamically illustrating the trend of change over time, unlike the existing visualization techniques of categorical data.

In the case of long format, coordinates with the same time points are illustrated on the graph in yellow to distinguish between patients with similar time points. However, in the case of wide format, when the mouse hovers over the observation coordinate points, the graph shows the hover effect in which the coordinate points of observations with the same covariate are displayed in yellow points on the graph. Figure 8 shows four graphs with yellow observation coordinate points according to the variable combinations of the covariates. For example, observations with a covariate of Diagnosis.1

and Drug.1 appear in the upper left of the graph. In this graph, the observations on the right side of the first axis appear abnormal for all three time points, and the more they move to the left, the more normal they appear. Conversely, observations with covariates Diagnosis.2 and Drug.2 appear in the lower right of the graph, and the observations on the left side of the first axis appear normal for all three time points, however, the more they move to the right, the more abnormal they appear. It is possible to visually examine the number of observation points, according to the covariate, which are close to the response variables.

Next, we consider applying the additional measured objects and variables to the depression data. In `Depression_row` data, supplementary objects represent 100 patients in each mild and severe group that took placebos instead of standard and new drugs. These supplementary objects are added as rows to the depression data. `Depression_column` data consists of the response of 800 patients at the fourth time point (after 6 weeks) and how gender affects depression. The data is added to the depression data in the form of columns.

```
R> data(Depression_row)
R> data(Depression_column)
R> str(Depression_row)
'data.frame': 100 obs. of 10 variables:
 $ Diagnosis.1: int  1 1 1 1 1 1 1 1 1 1 ...
 $ Diagnosis.2: int  0 0 0 0 0 0 0 0 0 0 ...
 $ Drug.1      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Drug.2      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ 1week.1     : int  0 0 0 0 0 0 0 0 0 0 ...
 $ 1week.2     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ 2weeks.1    : int  0 0 0 0 0 0 0 0 1 1 ...
 $ 2weeks.2    : int  1 1 1 1 1 1 1 1 0 0 ...
 $ 4weeks.1    : int  0 0 0 1 1 1 1 1 0 0 ...
 $ 4weeks.2    : int  1 1 1 0 0 0 0 0 1 1 ...

R> str(Depression_column)
'data.frame': 800 obs. of 4 variables:
 $ 6weeks.1: int  0 0 0 0 0 0 0 0 0 0 ...
 $ 6weeks.2: int  1 1 1 1 1 1 1 1 1 1 ...
 $ sex.1   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ sex.2   : int  0 0 0 0 0 0 0 0 0 0 ...
```

The first example was already shown in Figure 6 in Section 4.1 by using the default command `vlda_add(fit, add.row = PTSD_row, add.col = PTSD_column)`. Note that `vlda_add` contains the argument `time.name`. The argument `time.name` is a character string that specifies the name of the time variable, if the supplemental data to be added contains a time variable. `Depression_column` data contains the time variable, `6weeks`, therefore, argument `time.name` specifies the character string `6weeks`.

The `vlda_plot` function can draw a more complete graph by suitably specifying the graphic element arguments. If the variable name is unsatisfactory or complex, defining the argument `rename` can change the variable name. The argument `rename` is input in the order in which the variables will join the plot. A more detailed implementation of graphic element arguments can be found in the

displayed documentation using the commands `R> ?vlda_plot`. The function call which was used to create the graphic in Figure 9, is,

```
R> Depression_row <- as.matrix(Depression_row)
R> Depression_column <- as.matrix(Depression_column)
R> wide.fit2 <-
+ vlda_add(
+     wide.fit,
+     time.name = "6weeks",
+     add.row = Depression_row,
+     add.col = Depression_column
+ )

R> rownames(wide.fit2$var.coordinate)
[1] "1week.1" "1week.2" "2weeks.1" "2weeks.2" "4weeks.1"
[6] "4weeks.2" "6weeks.1" "6weeks.2" "Diagnosis.1"
[10] "Diagnosis.2" "Drug.1" "Drug.2" "Drug.3" "sex.1" "sex.2"

R> vlda_plot(
+     wide.fit2,
+     rename = c("1week.Ab", "1week.N",
+                 "2weeks.Ab", "2weeks.N",
+                 "4weeks.Ab", "4weeks.N",
+                 "6weeks.Ab", "6weeks.N",
+                 "Mild", "Severe",
+                 "New", "Standard", "Placebo",
+                 "Male", "Female" ),
+     title = "Depression data",
+     title.col = "#555555", title.size = 25,
+     subtitle = "Supplementary objects and variables added",
+     sub.size = 15, sub.col = "darkgrey",
+     legend.position = c(0.15, 0.15)
+ )
```

Figure 9 shows an additional representation of coordinates to the VLDA plot already provided in Figure 7. It illustrates sex and the fourth time point (after 6weeks) as columns parameters for 800 patients added by the `Depression_row` data. The supplementary coordinates of the 100 patients that took placebos were added by the supplementary row indicator matrix. The number of categorical combinations for each variable had a total of 16 possible combinations for the 100 patients that were taking placebos. Therefore, Figure 9 shows 16 supplementary coordinates in dark grey. The `add.obs.col` argument allows the user to specify the color of the supplementary coordinates (default = "darkgray"). If `add.obs.col = "equal"`, the output is in the same color as the color specified by the user in the `obs.col` argument. The color argument used in the package can be specified as a hex code (e.g., `title.col = "#555555"`) or a string (e.g., `sub.col = "darkgray"`). The `legend.postition` argument is used to adjust the position of the legends. The position of the legends can be specified by the character string "none", "left", "right", "bottom", "top", or a numeric vector of both elements (default = "bottom"). The placebo, a supplementary overall

Figure 9: Coordinates for supplementary objects and variables added to Figure 7.

Table 4: Normal response proportion at each time point for the placebo

Diagnosis	Drug	Normal response proportion		
		1 week	2 weeks	4 weeks
Mild	Placebo	0.36	0.36	0.24
Severe	Placebo	0.20	0.18	0.18

Table 5: Normal response proportion at the fourth time point (after 6 weeks) according to combinations of diagnosis and drug

Diagnosis	Drug	Normal response proportion
		6 weeks
Mild	Standard	0.57
	New	1.00
Severe	Standard	0.35
	New	0.94

characteristic of the patient, is placed on the right side of the first axis. In other words, the placebo is not effective at all, indicating homogeneity with abnormal. This is also shown in Table 4, in which the normal response proportion at each time point for placebo is indicated. Within the placebo, we know that the normal proportion decreases over time, and that mild has a slightly higher proportion than severe. In other words, new drugs are more effective than standard drugs, and mild is more effective than severe. This suggests that the placebo has no effect at all, as described earlier.

In the supplemental data added by the column indicator matrix, the fourth time point (after 6 weeks) refers to the time that passes from 4 weeks to 6 weeks, normal (4weeks.N → 6weeks.N) draws closer to the new drug, and abnormal (4weeks.Ab → 6weeks.Ab) draws closer to the standard drug. That is, the effect of the drugs still increases over time. This can also be confirmed in Table 5 which shows the normal response proportion at the fourth time point (after 6 weeks) according to the combinations of diagnosis and drug. The drug effect according to diagnosis shows that 6 weeks

Table 6: Normal response proportion at each time point by combinations of diagnosis, drug, and gender

Diagnosis	Drug	Gender	Normal response proportion		
			1 week	2 weeks	4 weeks
Mild	Standard	Male	0.40	0.54	0.68
		Female	0.42	0.42	0.42
	New	Male	0.38	0.62	0.86
		Female	0.36	0.44	0.54
Severe	Standard	Male	0.22	0.26	0.46
		Female	0.20	0.24	0.16
	New	Male	0.22	0.30	0.46
		Female	0.14	0.20	0.36

increased normal proportion compared to after 4 weeks. Also, we observe that the new drug is more effective than the standard drug, and that mild is more effective than severe.

Next, we consider the combination of sex and the fourth time point added by the supplementary column indicator matrix. Normal at the fourth time point (6weeks.N) and male on the left side of the first axis shows homogeneity with new drug, mild, and normal (1week.2, 2weeks.2, and 4weeks.2). On the contrary, abnormal at the fourth time point (6weeks.Ab) and female on the right side of the first axis shows homogeneity with standard, severe, and abnormal (1week.Ab, 2weeks.Ab, and 4weeks.Ab, respectively). Therefore, if patients who take a new drug have mild initial depression and are a male, they are more likely to be normal. Contrarily, if patients who take a standard drug have severe initial depression and are female, they are more likely to be abnormal. Table 6 shows the normal response proportion at three time points by combining diagnosis, drug, and gender. In males, the proportion of all combinations increased, and the proportion of new drug was higher than standard drug, and the proportion of mild was higher than severe. Conversely, in females, the patients who took the standard drug showed no change or decrease, and the patients who took the new drug had a higher proportion of mild than severe.

5. Conclusion

This paper presents the R package **vlda**, which can be used to implement the algorithm for the analysis and visualization of longitudinal data. In particular, the R (R Development Core team *et al.*, 2011) package **vlda** was developed to assist in producing a plot that more effectively expresses changes over time for two different types (long format and wide format) and uses a consistent calling scheme for longitudinal data. The R package **vlda** is useful for visual analysis as it provides a new interactive implementation to perform interpretations and is an addition to the existing VLDA plot. Interactivity provides a tooltip to display values when hovering over the coordinates and the ability to display different hover effects according to the two types (long format and wide format). Due to the synergistic relationship between the existing VLDA plot and interactive features, the user is empowered by a refined observe the visual aspects of the VLDA plot layout.

The generalized estimation equation (GEE) as introduced by Liang and Zeger (1986) is commonly used to analyze longitudinal data. It is possible to identically interpret the VLDA plot and the GEE result. In fact, the VLDA plot showed a relatively strong performance compared to the GEE results based on the real longitudinal data analysis from Lee (2019). As a result, the same interpretation as the GEE results is possible. However, **vlda** does not implement a result table for GEE. Therefore, extensions in a future version of this package should implement the standard GEE approach for longitudinal data.

References

- Allison PD (2001). *Logistic Regression Using the SAS System, Theory and Application*, SAS Institute Inc.
- Everitt BS and Hothorn T (2006). *A Handbook of Statistical Analyses Using R*(2nd ed), Chapman & Hall, CRC Press, Florida, USA.
- Gohel D (2016). *ggiraph: Make 'ggplot2' Graphics Interactive*, R package version 0.3.2.
- Greenacre MJ (1984). *Theory and Applications of Correspondence Analysis*, Academic Press, New York.
- Greenacre M and Hastie T (1987). The geometric interpretation of correspondence analysis, *Journal of the American Statistical Association*, **82**, 437–447.
- Jeong KM and Choi YS (2009). *Introduction to Categorical Data Analysis - Application and Interpretation of the SAS*(2nd ed), Free Academy, Seoul.
- Koch GG, Landis JR, Freeman JL, Freeman DH, and Lehnen RC (1977). A general methodology for the analysis of experiments with repeated measurement of categorical data, *Biometrics*, **33**, 133–158.
- Lebart L, Morineau A, and Warwick K (1984). *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*, John Wiley & Sons, New York.
- Lee B (2019). *Visualization of Multidimensional Longitudinal Data* (Ph.D. thesis), Pusan National University, Pusan.
- Liang KY and Zeger SL (1986). Longitudinal data analysis using generalized linear models, *Biometrika*, **73**, 13–22.
- R Development Core team A *et al.* (2011). R: A language and environment for statistical computing, *R Foundation for Statistical Computing*, Vienna, URL: <http://www.R-project.org>.
- Singer JD and Willett JB (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*, Oxford University Press, Oxford.
- Wickham H (2011). *ggplot2*, *Wiley Interdisciplinary Reviews: Computational Statistics*, **3**, 180–185.
- Xiao N (2018). *ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'*, R package version 2.9.

Received January 07, 2021; Revised May 03, 2021; Accepted May 26, 2021