vdmR: Generating web-based visual data mining tools with R

Tomokazu Fujino

June 3, 2015

1 Outline of usage

1.1 Installation and Sample dataset

The vdmR pakcage can be loaded as following command.

```
library(vdmR)
```

Several packages are required for **vdmR**, which would be installed all together if you didn't install them previously. vdmR provides the sample dataset for demonstration and illustration of the package.

```
data(vsfuk2012)
head(vsfuk2012[,1:5])

##   CityCode CityName Type FertilityRate MortalityRate
## 1    40133   Chuuou Ward           8.9           6.0
## 2    40132   Hakata Ward          10.4           6.9
## 3    40134   Minami Ward           9.9           7.3
## 4    40137   Sawara Ward           9.9           7.0
## 5    40131  Higashi Ward          10.5           7.1
## 6    40135    Nishi Ward          10.2           7.4
```

The `vsfuk2012` data set gives the result of the vital statistics in Fukuoka prefecture of Japan from 2008 to 2012. The data set consists of 72 rows and 17 columns. Each row indicates the result of one municipality of Fukuoka prefecture. Each column shows basic information of the municipalities such as code, name or type (city, town, village) and indices such as population, fertility rate, mortality rate and so on. Information of all variables are given in the help document of the data set.

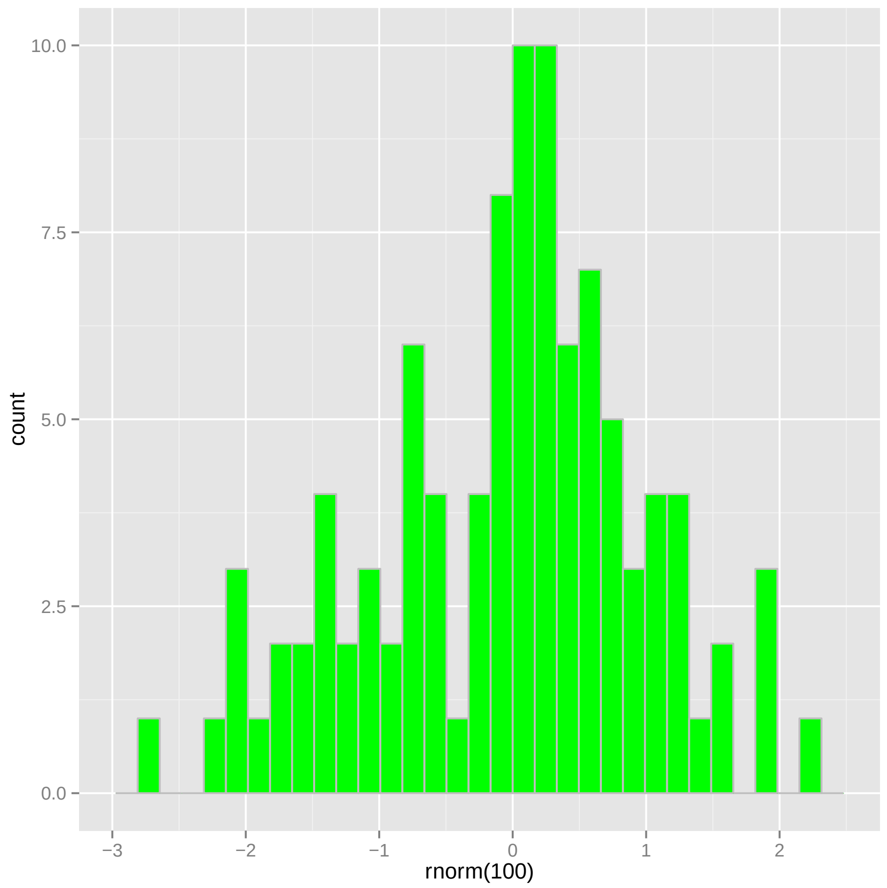
1.2 Creating VDM tools

Basically, to create VDM tools from **vdmR** are achieved in following two steps.

- create each statistical plots in html files with SVG and javascript
- create main window for launching each plot and open the window

Here, we will show the example for generating two plots.

```r
vscat(MortalityRate, FertilityRate, vsfuk2012, "scat01", "vsfuk2012")
vhist(MarriageRate, vsfuk2012, "hist01", "vsfuk2012")
```



Each command generates some files in the current working directory, which are named in the following rule:

$\{plot\ name\}.\{tag\ name\}.(\texttt{svg}.)\{extension\}$

All html files include `<embed>` tag for the related SVG file which contain the graphical information such as axes and sizes of the shapes in the plot. When opening the html file with a Web browser, the plot will appear. Note that the interactive functions will not enabled yet at this stage. To enable interactive functions, you need to generate "launcher" using following command.

```r
vlaunch(vsfuk2012, "main", "vsfuk2012")
```

This command generate html file named like

{*launcher name*}.{*tag name*}`.html`

with some related files (`.js, .css`), and then launch the default Web browser and display the html file (Figure 1). This VDM tool currently doesn't support Internet Explorer, while supporting latest versions of Google Chrome, Mozilla Firefox and Safari.

1.3 Manipulating VDM tools

When clicking one of the button located on the upper part of the main window, corresponding plot window will be opened (Figure 2, 3). You can see the gray rectangle on the top left of the window. Moving this rectangle selection tool, in the case of scatter plot, the data points in the rectangle region will be highlighted, and then the histogram of the selected data will be drawn on the histogram window by highlighted color. In addition, corresponding rows in the data table of the main window will be also highlighted (Figure 4).

The persistent selection is supported in **vdmR** package, which means that once the objects such as points in the scatter plot are selected, the selected objects continue to be selected even if these objects are out of the region of the selection tool to be moved. In our VDM tools, double-clicking selection tool enables the persistent selection mode. When double-clicking it again, it will return to normal (temporary) selection mode (Figure 5).

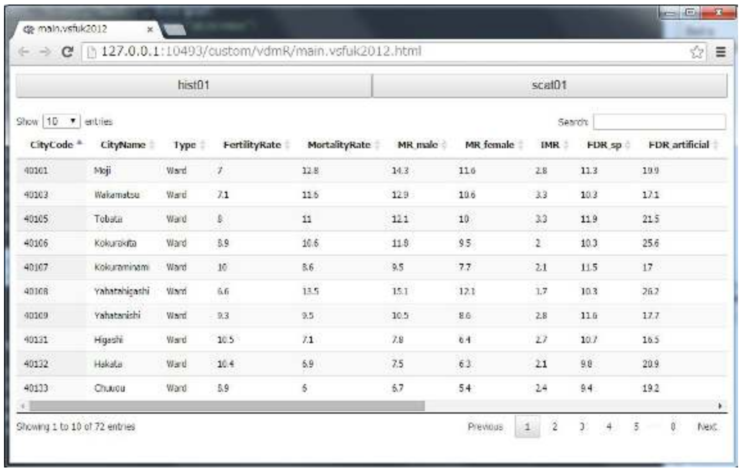
The selection tool is also resizable. When dragging the mouse on the bottom right of the selection tool, the region will be resized.

The sortable table of the data set is displayed on the main window. When clicking one of the variable labels, the rows are sorted based on the descending or ascending order of the corresponding variable.

2 Detail illustration of vdmR

2.1 data table and launcher of other plots: `vlaunch()`

`vlaunch()` function generates a main window which opens each pre-generated windows including a statistical plot with interactivity. At first, the function finds pre-generated html and SVG files in the current working directory by using regular expressions based on the `tag` argument. In the example of section 2.2, `vlaunch()` function finds html files including a SVG file of a statistical plot which have a name like `*.vsfuk2012.svg.html`. Then the function generates buttons for opening these files in the html file. All of the plot window should be opened from main window, because the multiple linked views (linked-brushing) can be implemented by using the cross-document messaging through the main window, which is one of the HTML5 technologies.



main.vsfuk2012

127.0.0.1:10493/custom/vdmR/main.vsfuk2012.html

hist01 | scat01

Show 10 entries Search:

CityCode	CityName	Type	FertilityRate	MortalityRate	MR_male	MR_female	IMR	FDR_sp	FDR_artificial
40101	Moji	Ward	7	12.8	14.3	11.6	2.8	11.3	19.9
40103	Wakamatsu	Ward	7.1	11.6	12.9	10.6	3.3	10.3	17.1
40105	Tobata	Ward	8	11	12.1	10	3.3	11.9	21.5
40106	Kokurakita	Ward	8.9	10.6	11.8	9.5	2	10.3	25.6
40107	Kokuraminami	Ward	10	8.6	9.5	7.7	2.1	11.5	17
40108	Yahatahigashi	Ward	6.6	13.5	15.1	12.1	1.7	10.3	26.2
40109	Yahatanishi	Ward	9.3	9.5	10.5	8.6	2.8	11.6	17.7
40131	Higashi	Ward	10.5	7.1	7.8	6.4	2.7	10.7	16.5
40132	Hakata	Ward	10.4	6.9	7.5	6.3	2.1	9.8	20.9
40133	Chuuou	Ward	8.9	6	6.7	5.4	2.4	9.4	19.2

Showing 1 to 10 of 72 entries Previous 1 2 3 4 5 … 8 Next

Figure 1: Main window generated by `vlaunch()` function

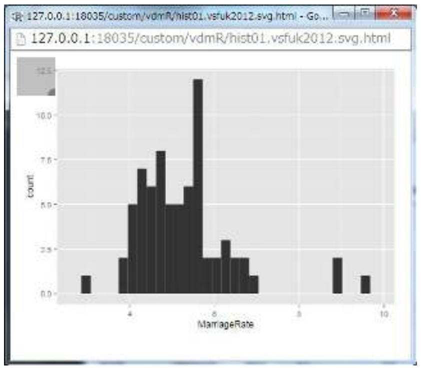


Figure 2: Histogram window generated by `vhist()` function

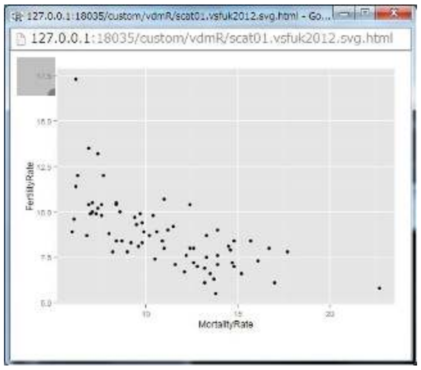
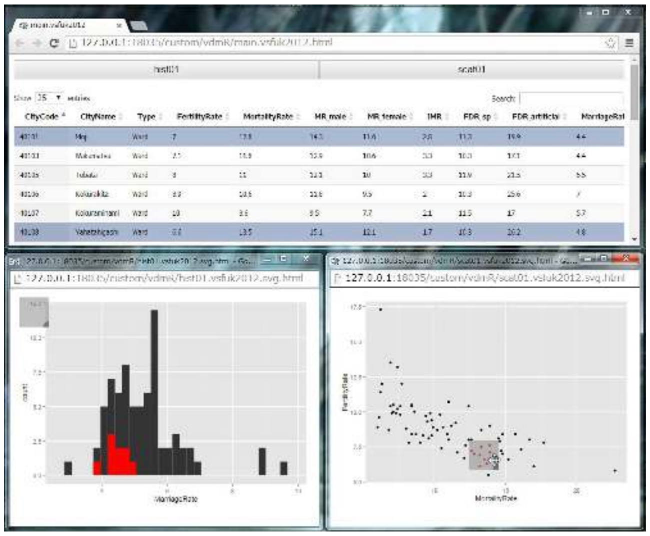


Figure 3: Scatter plot window generated by `vscat()` function



127.0.0.1:18035/custom/vdmR/main.vsfuk2012.html

Show 25 entries

Search:

CityCode	CityName	Type	FertilityRate	MortalityRate	MR male	MR female	IMR	FDR sp	FDR artificial	MarriageRat
40101	Moji	Ward	7	13.8	14.3	11.6	2.6	11.7	19.9	4.4
40103	Wakamatsu	Ward	7.7	11.8	12.9	10.6	3.3	10.1	17.1	4.4
40105	Tobata	Ward	8	11	12.1	10	3.3	11.9	21.5	5.5
40106	Kokurakita	Ward	8.8	10.5	11.6	9.5	2	10.3	25.6	7
40107	Kokuraminami	Ward	10	8.6	9.5	7.7	2.1	11.5	17	5.7
40108	Yahatahigashi	Ward	6.6	13.5	15.1	12.1	1.7	10.3	20.2	4.8

hist01
scat01
127.0.0.1:18035/custom/vdmR/hist01.vsfuk2012.svg.html
127.0.0.1:18035/custom/vdmR/scat01.vsfuk2012.svg.html
count
MarriageRate
FertilityRate
MortalityRate

Figure 4: Example of the multiple linked views

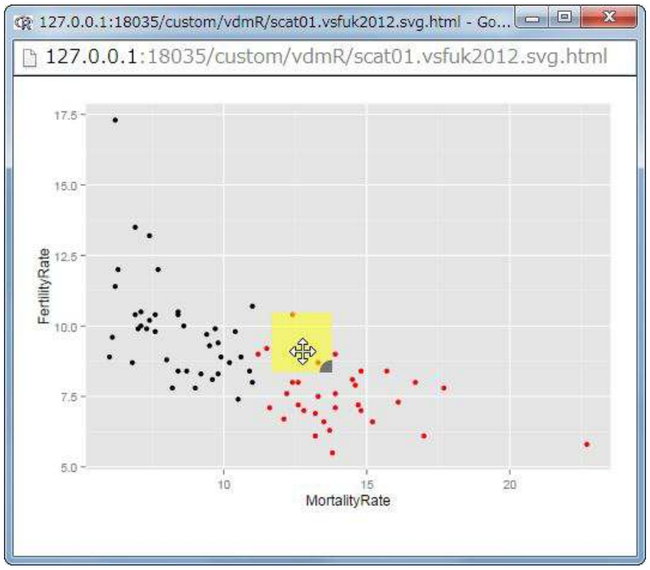

127.0.0.1:18035/custom/vdmR/scat01.vsfuk2012.svg.html - Go...
127.0.0.1:18035/custom/vdmR/scat01.vsfuk2012.svg.html
17.5
15.0
12.5
10.0
7.5
5.0
10
15
20
FertilityRate
MortalityRate

Figure 5: Example of the persistent selection in a scatter plot

The data table in the main window is implemented by DataTables which is a plug-in for the jQuery Javascript library. It provides many useful functions for data tables in web pages such as pagination, instant search, multi-column ordering and so on.

2.2 scatter plot: `vscat()`

`vscat()` function generates a interactive scatter plot in **ggplot2** style. The first two arguments `x,y` are the column names in the dataframe given by third arguments `data`, which are mapped into x-axis and y-axis of the scatter plot respectively. The dots argument (`...`) is passed to `aes()` function of **ggplot2**. Thus, in the `vsfuk2012` dataset, it is possible to relate each city's population of male to its point size on the scatter plot, and each city's type to its color on it by following script:

```
vscat(MortalityRate, FertilityRate, vsfuk2012, "scat01", "vsfuk2012",
      color=Type, size=pop_male)
```

The output of this `vscat()` function is shown in Figure 6. If you need to set (not aesthetic mapping) the color of all of the points to the specific color, you can do it by setting the argument of `I()` function to the character of the color name or color code like following example.

```
vscat(MortalityRate, FertilityRate, vsfuk2012, "scat01", "vsfuk2012",
      color=I("darkgreen"), size=pop_male)
```

This is a same style as `qplot()` function of **ggplot2**.

2.3 histogram: `vhist()`

`vhist()` function generates a interactive histogram in **ggplot2** style. The arguments of the function is almost same as `vscat()` function except that the variable of `y` doesn't exsist. If you need to specify the color of the histogram, you can do it as follows:

```
vhist(MarriageRate, vsfuk2012, "hist01", "vsfuk2012",
      fill=I("darkgreen"), color=I("black"))
```

A histogram visualizes a frequency count table of the specific variable of the dataset. Thus each graphic objects (bars of histogram) of the plot doesn't have one-to-one correspondence with a row of the dataset. This makes a little difficulties to implement linked-brushing facility. We have solved this by embedding the mapping information between each data and a bar of histogram into generated SVG files in JSON (Javascript Object Notation) format using **rjson** package.

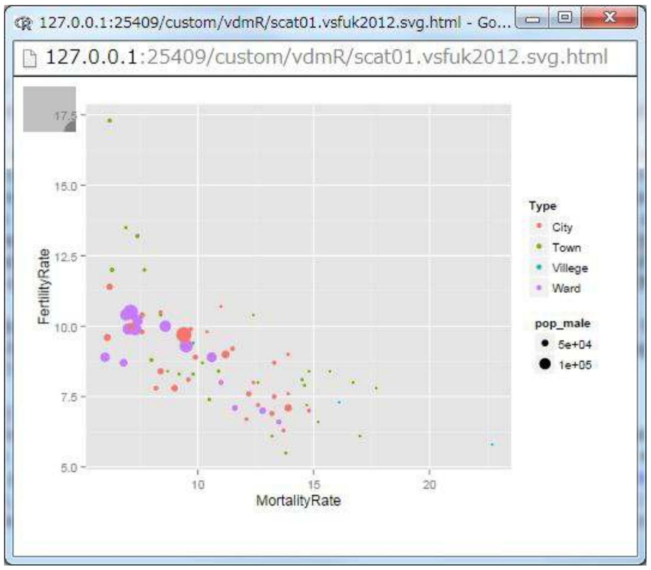


Figure 6: Scatter plot with aesthetic mapping

2.4 parallel coordinate plot: `vpcp()`

A parallel coordinate plot is well-known as a powerful tool for visualizing multivariate data. `vpcp()` function can generate a parallel coordinate plot with interactive facilities in **ggplot2** style. In **ggplot2**, `ggpcp()` function can draw the parallel coordinate plot, however it has been deprecated. Thus, we used `ggparcoord()` function provided by **GGally** package in the `vpcp()` function. So the most of arguments of the `vpcp()` function is same as `ggparcoord()` function. An example is shown in Figure 7 created with the following code.

```
vpcp(vsfuk2012, 4:17, "pcp1", "vsfuk2012",
     groupColumn="Type", scale="uniminmax", missing="min10")
```

Note that the argument of `groupColumn` require the character of the column name. It means that double quotation marks are needed.

2.5 choropleth map: `vcmap()`

The interactive version of a choropleth map in which polygons of the areas are painted in proportion to the value of the specific variable is one of the central feature of **vdmR** package. By using the choropleth map in the multiple linked views, it will be easy to see the relationship between the spatial characteristics and the multivariate characteristics. The interactive linked micromap plot is one of the interactive application of the choropleth map.

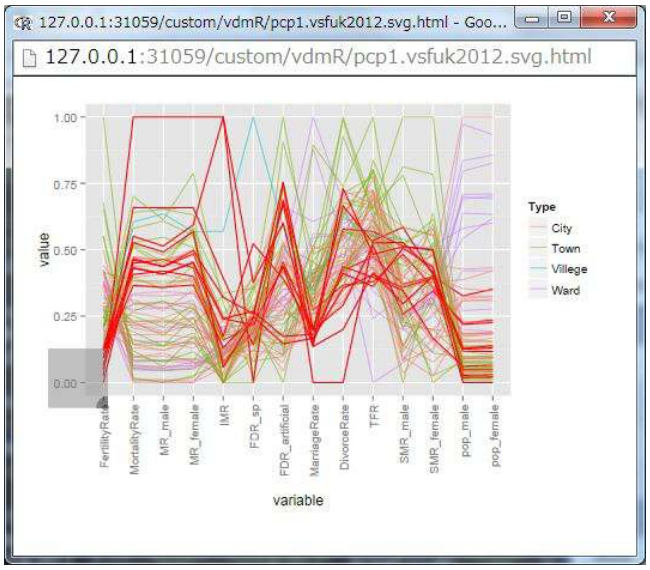


Figure 7: Parallel coordinate plot

To generate the interactive choropleth map in **vdmR**, you have to prepare the ESRI shapefile including the region for the dataset in which attribute table has a common column of the id column of the data frame. The **vdmR** package provides a sample shapefile related to the data `vsfuk2012`. By using **maptools** package it is possible to import the shapefile into R environment as `SpatialPolygonsDataFrame` object as follows:

```
library(maptools)
shp.path <- file.path(system.file(package="vdmR"), "etc/shapes/fukuoka2012.shp")
vsfuk2012.spdf <- readShapeSpatial(shp.path, IDvar="CityCode")
head(vsfuk2012.spdf@data)

##       CityCode      CityName Type
## 40101    40101          Moji Ward
## 40103    40103     Wakamatsu Ward
## 40105    40105        Tobata Ward
## 40106    40106    Kokurakita Ward
## 40107    40107   Kokuraminami Ward
## 40108    40108 Yahatahigashi Ward
```

If the shapefile doesn't have the common column of the id column of the data frame, you need to edit the shapefile by using desktop GIS software such as QGIS or ArcGIS or edit the data frame.

`vcmap()` function provides the interactive choropleth map in **vdmR** package.

For example, following codes will generate the interactive choropleth map of the fertility rate in Fukuoka prefecture:

```r
frcol <- ggplot2::scale_fill_gradient2(low="blue", mid="white", high="red",
                                midpoint=median(vsfuk2012$FertilityRate))
vcmap(shp.path, vsfuk2012, "CityCode", "CityCode", "map1", "vsfuk2012",
      fill=FertilityRate, ggscale=frcol)
```

Figure 8 shows the result of the code. The first and second arguments take the path to the shapefile and the data frame, respectively. The third and fourth argument take column names of the common id for the attribute table and data frame, respectively. The argument `fill` takes a column name assigned to the color of polygons. If you need to use your own color scale, the color scale which is generated by `scale_fill_*()` function has to be passed to the argument `ggscale`.

Thu brushing operation is a little different from other **vdmR** outputs. The select box doesn't appear on the choropleth map, so you have to brush the polygons directly by the mouse pointer. The operation on the choropleth map is always persistent selection. Double-clicking out of the polygons' region will reset the all of the selections.

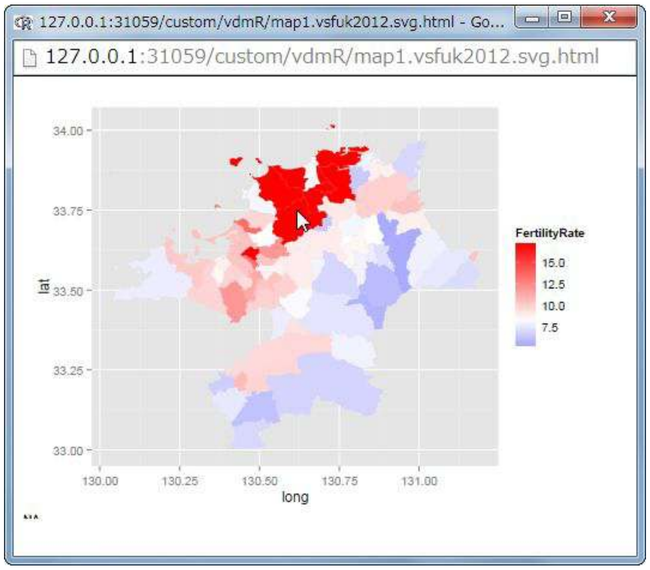

127.0.0.1:31059/custom/vdmR/map1.vsfuk2012.svg.html - Go...
127.0.0.1:31059/custom/vdmR/map1.vsfuk2012.svg.html
34.00
33.75
33.50
33.25
33.00
lat
130.00
130.25
130.50
130.75
131.00
long
FertilityRate
15.0
12.5
10.0
7.5

Figure 8: Choropleth map