# ggmatplot: An R package for data visualization on wide-format data

Xuan Liang[1]**, Francis K. C. Hui**[1]**, Dilinie Seimon**[2]**, and Emi Tanaka**[2]

1 Research School of Finance, Actuarial Studies and Statistics, The Australian National University **2** Department of Econometrics and Business Statistics, Monash University

Summary

The layered grammar of graphics (Wickham, 2010), implemented as the `ggplot2` package (Wickham, 2016) in the statistical language R (R Core Team, 2021), is a powerful and popular tool to create versatile statistical graphics. However, this graphical system requires input data to be organised in a manner that a data column is mapped to an aesthetic element (e.g., x-coordinate, y-coordinate, color, size), which creates friction in constructing plots with an aesthetic element that spans multiple columns in the original data, as it requires the user to reorganise the data.

The `ggmatplot`, built upon `ggplot2`, is an R-package that allows quick plotting across the columns of matrices or data with the result returned as a `ggplot` object. The package is inspired by the function `matplot()` in the core R `graphics` system – as such, `ggmatplot` may be considered as a `ggplot` version of `matplot`, with the benefits of customising the plots as any other `ggplot` objects via `ggplot2` functions, as well as offering several other plotting types that are not immediately available from `matplot` directly e.g., comparative violin plots. The `ggmatplot` package is available on the Comprehensive R Archive Network (CRAN) with the latest developmental source code available at https://github.com/xuan-liang/ggmatplot.

Statement of need

Input data to construct plots with `ggplot2` require data to be organised in a manner that maps data columns to aesthetic elements. This generally works well where data is tidied in a long rectangular form, often referred to as "tidy data" (Wickham, 2014), where each row represents an observational unit, each column represents a variable, and each cell represents a value. However, in some cases what constitutes a variable (or observational unit), and hence a column (or row), in tidy data can be dependent upon interpretation or downstream interest (e.g. Tables 1 and 2 can both be considered as tidy data), but a clear violation of tidy data principles is when the column names contain data values, e.g. Table 3 contains months of the year across a number of column names.

The organisation of the data is largely dependent on the subsequent analysis, and there is no one correct way to do this. Some forms of multivariate data, e.g. Table 3, are prevalent in many scientific fields because it aligns with the input data for a particular modelling software, and/or the format is more convenient for input or view of the data in spreadsheet format (say). Unfortunately, this format is not consistent with the required format for `ggplot2`, and consequently plotting with `ggplot2` interrupts the workflow of a user that is trying to quickly visualise these types of data (as part of their exploratory data analysis, for example). The `ggmatplot` R-package seeks to provide a solution to this common friction.

Table 1: Restaurant rating data in "tidy" form. The first column shows the restaurant ID, and the next four columns show the average ratings (out of 5) for food, service, ambience and overall, respectively.
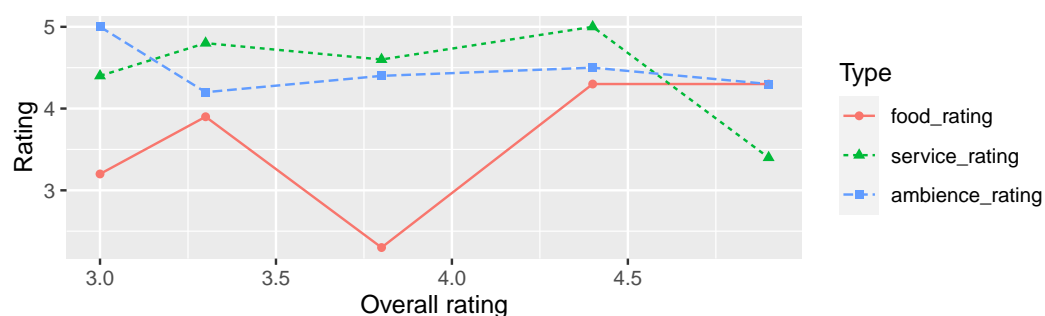
Restaurant	Average rating			
	Food	Service	Ambience	Overall
R1	4.3	3.4	4.3	4.9
R2	4.3	5.0	4.5	4.4
R3	3.2	4.4	5.0	3.0
R4	2.3	4.6	4.4	3.8
R5	3.9	4.8	4.2	3.3

Table 2: Another form for the restaurant rating data in Table 1. In Wickham (2014), this format is called the "molten" data.

Restauant	Rating type	Average rating
R1	food	4.3
R1	service	3.4
R1	ambience	4.3
R1	overall	4.9
R2	food	4.3
R2	service	5.0
R2	ambience	4.5
R2	overall	4.4
R3	food	3.2
R3	service	4.4
R3	ambience	5.0
R3	overall	3.0
R4	food	2.3
R4	service	4.6
R4	ambience	4.4
R4	overall	3.8
R5	food	3.9
R5	service	4.8
R5	ambience	4.2
R5	overall	3.3

Table 3: The first 6 rows and 11 columns of the snowfall data for Grand Rapids, Michigan in the R pacakge `mosaicData` (Prium, Kaplan & Horton, 2021).

SeasonStart	SeasonEnd	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
1893	1894	0	0	0	0.0	8.0	24.9	12.5	6.8	4.8
1894	1895	0	0	0	0.0	7.5	5.3	21.5	8.0	22.5
1895	1896	0	0	0	0.4	23.2	15.0	.	8.5	2.0
1896	1897	0	0	0	0.2	8.0	8.0	4.9	11.2	12.0
1897	1898	0	0	0	0.0	1.4	8.0	15.5	29.5	0.0
1898	1899	0	0	0	0.0	18.5	18.0	20	3.4	16.0

Figure 1: Line plot of the food, service, and ambience ratings versus overall rating, for five restaurants.

Examples

In this section, we demonstrate the use of the `ggmatplot` package and contrast the specification with `ggplot2` after data wrangling using `dplyr` and `tidyr` (Wickham et al., 2019). We will use the example data in Tables 1 and 3, which are stored in the objects `wide_df` and `SnowGR`, respectively.

Example 1

The code below constructs a line plot (superimposed with a point) of the various types (food, service and ambience) of ratings, contained in columns 2 to 4 of `wide_df`, against the overall rating in column 5 of `wide_df` as shown in Figure 1.

```
library(ggmatplot)
ggmatplot(x = wide_df[, 5], y = wide_df[, 2:4], plot_type = "both",
          xlab = "Overall rating",  ylab = "Rating", legend_title = "Type")
```

In contrast to the above, using `ggplot2` alone to obtain the same result as Figure 1 requires the data be wrangled to a long form first before plotting. This is exemplified in the code below, from which we identify a small, but noticeable, friction to the workflow for the practitioner that is looking to promptly explore their data.

```
library(ggplot2)
library(tidyr) # or library(tidyverse)
wide_df %>%
  select(contains("rating")) %>%
  pivot_longer(-overall_rating,
               names_to = "rating_type",
               values_to = "rating") %>%
  ggplot(aes(x = overall_rating, y = rating, color = rating_type)) +
  geom_point(aes(shape = rating_type)) +
  geom_line(aes(group = rating_type, linetype = rating_type)) +
  labs(x = "Overall rating", y = "Rating",
       color = "Type", linetype = "Type", shape = "Type")
```

Example 2

The example code draws the boxplot of each column of amount of snowfall across months in the `SnowGR` data, as presented in Figure 2. As the resulting object is a `ggplot` object, the user can leverage the `ggplot` functions to modify the output (e.g., addition of a title).

```
library(ggmatplot)
ggmatplot(x = SnowGR[, 3:14], plot_type = "boxplot",
```
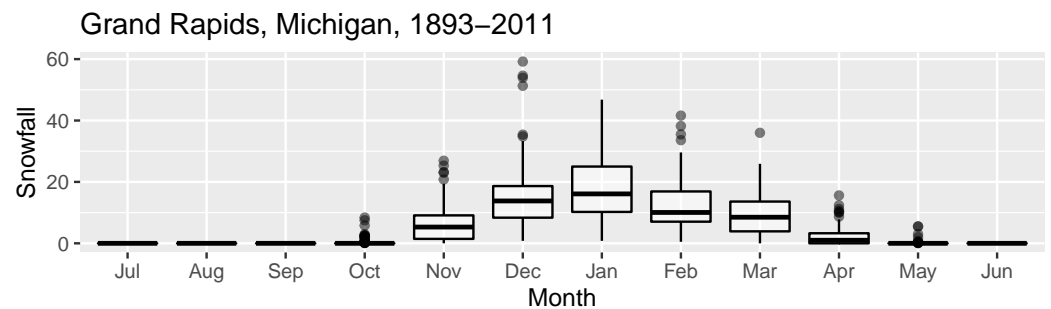
Figure 2: The distribution of the amount of snowfall at Grand Rapids, Michigan, across months from 1893-2011.

```
        xlab = "Month",  ylab = "Snowfall") +
        ggtitle("Grand Rapids, Michigan, 1893-2011")
```

The equivalent code to produce Figure 2 without using `ggmatplot` is given below. Again, we observe a slight but non-negligible friction in putting the data in the right format prior to plotting. The original wide data format as seen in Table 3 is common in the environmental sciences, among other disciplines, and thus an analyst who has to repeat these tasks can benefit from a quick approach as `ggmatplot` offers.

```
library(ggplot2)
library(tidyr)
library(forcats) # or library(tidyverse)
SnowGR %>%
  pivot_longer(Jul:Jun,
               names_to = "Month",
               values_to = "Snowfall") %>%
  mutate(Month = fct_inorder(Month)) %>%
  ggplot(aes(Month, Snowfall)) +
  geom_boxplot() +
  ggtitle("Grand Rapids, Michigan, 1893-2011")
```

Discussion

The `ggmatplot` R-package provides a solution to a common friction encountered when wanting to quickly plot multivariate data, where the primary interest is mapping the column names as an aesthetic element. While an important start, we also acknowledge that solution provided is a recipe-driven approach, where the user can only produce plot types as many there are included in the `plot_type` option. Future developments of the package could benefit from using a grammar approach, like in Wilkinson (2005) and Wickham (2010), where plot types can be extensible. The latest developmental source code can be found at https://github.com/xuan-liang/ggmatplot. Further examples can be found at https://xuan-liang.github.io/ggmatplot/.

Acknowledgements

References

Pruim, R., Kaplan, D., & Horton, N. (2021). *mosaicData: Project MOSAIC data sets*. Retrieved from https://CRAN.R-project.org/package=mosaicData

R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Wickham, H. (2010). A layered grammar of graphics. *Journal of computational and graphical statistics: a joint publication of American Statistical Association, Institute of Mathematical Statistics, Interface Foundation of North America*.

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. doi:10.18637/jss.v059.i10

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., et al. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. doi:10.21105/joss.01686

Wilkinson, L. (2005). *The grammar of graphics*. Springer.