post01-ruoming-dong.Rmd

Why We Love ggplot2 in R.studio

Ruoming Dong

Octorber.31.2017

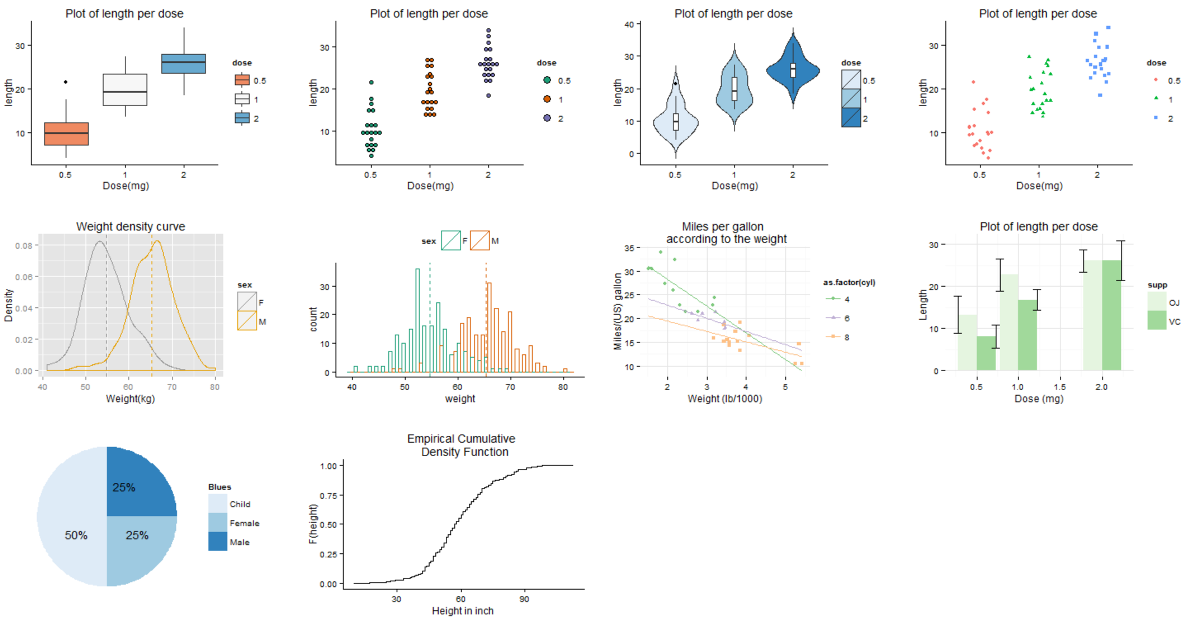
Background

There are a series of important ideas behind ggplot2, which is the syntax of graphics. This kind of thought that makes first user confused and users who always using that praise it. ggplot uses color, area, location and other complex graphics content, but only expressed the simple percentage. This only allows the information recipient's attention to move away from the message itself, even misleading. The first thing to keep in mind, therefore, is that the expression of information is the goal, and the various graphics forms are just tools for its service. You can't let the tools interfere or even drown out your goals. So if your goal is to make more "fancy" or better graphics, ggplot2 may not be your best bet. If your goal is to quickly and easily explore the information contained in the data, then ggplot2 should be used.

> ### Advantage
>
> - The default setting for ggplot2 has already made the graphics nice.
> - The plot () function in the basic drawing system is already powerful, but not convenient.
> - The ggplot code is more reusable and can easily create graphics with your own characteristics. This is helpful for data analysis that has been required to perform repetitive tasks.

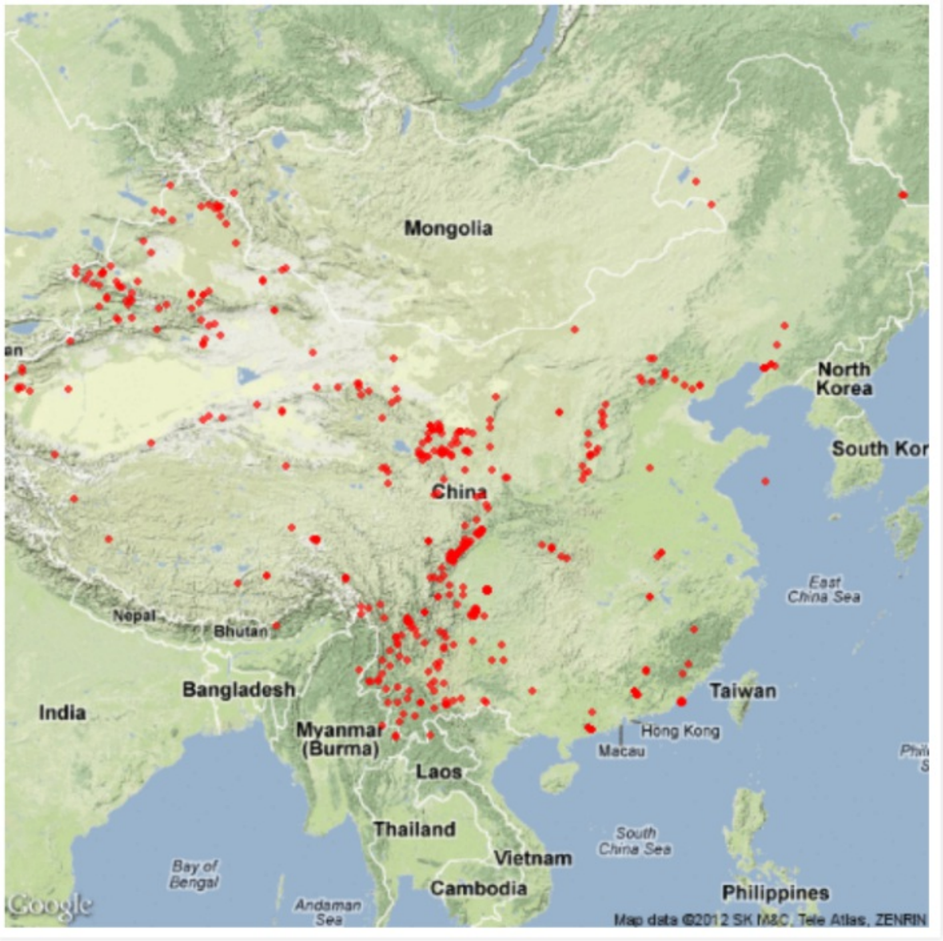
> ### Sample of graphs

- Basic images



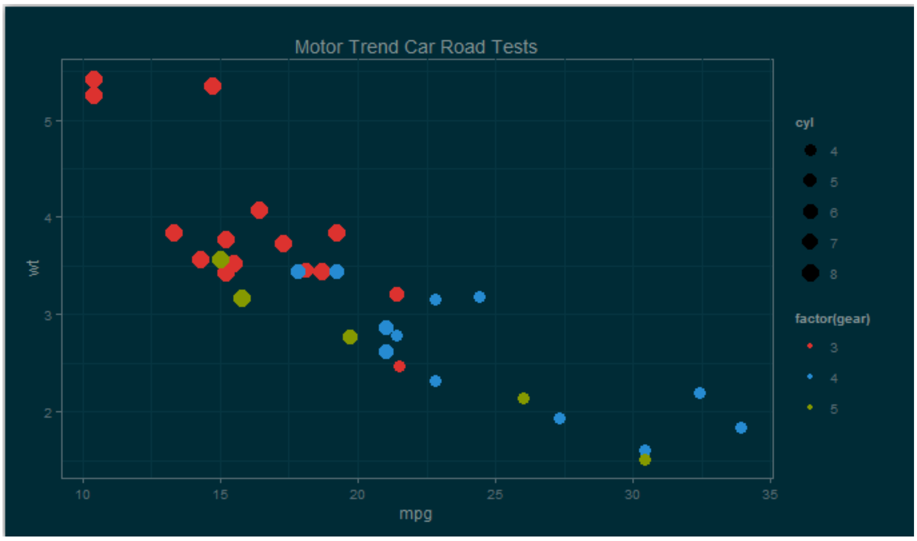
(this picture comes from 'ggplot2_The_Elements_for_Elegant_Data_Visulization_in_R')

- Expanding application examples
    - Mapping (ggmap)



(This picture from ggmap)

- ◦ Mainstream media theme



Process Map

1. data preparation

- If you need to use ggplot, must load package 'dplyr'.
- These are data structures based on data.frame

2. Concept introduction and examples of dplyr

- group_by, summarise, mutate,filter,select,arrange and so on. These functions combine to fit almost all of the data processing jobs
- The operator '% > %' can pass the result of the previous step as the first parameter to the next processing function. This makes the code more readable, and the process of processing data more clearly

3. Data

> - Ggplot2 needs to pass in data in a data.frame format. The data is saved in a graphical image, which means that we can save the graph (ggsave ()) for the next load (load ())
>
> 4. Graph attribute mapping
>
> - aes() is used to map data variables to graphics
> - example: aes(x,y colour = z)
> - example: aes(x,y group = z), divide the data into groups and render each group in the same way. It is worth mentioning that the interaction() function can be grouped when an existing single variable cannot be grouped correctly and the combination of two variables can be grouped correctly.
>
> 5. Geometric objects
>
> - abbreviation: geom. Control the generated image type. Example: geom_line(),geom_point(), geom_bar()
>
> 6. Position adjustment
>
> - For position adjustment, it is commonly seen in discrete data. Example: stack(), fill(), dodge()
> - Also, jitter:Add disturbance to the dot to avoid overlap; identitty: do not make any adjustments

- Syntactic intuition
    - A graphical object is a list of data, maps, layers, scales, coordinates, and points
    - Each element is an object that can be superimposed by overloading the “+”

> ## Example:

```
library(ggplot2)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

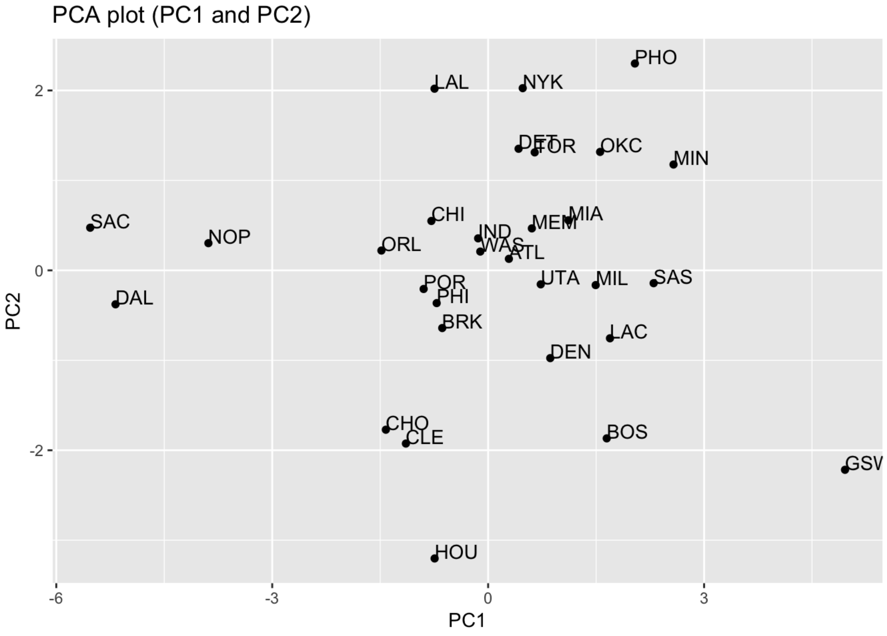
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
teams <- read_csv('/Users/ruomingdong/stat133/stat133-hws-fall17/hw03/data/nba2017-teams.csv')
```

```
## Parsed with column specification:
## cols(
##   team = col_character(),
##   experience = col_integer(),
##   salary = col_double(),
##   points3 = col_integer(),
##   points2 = col_integer(),
##   free_throws = col_integer(),
##   points = col_integer(),
##   off_rebounds = col_integer(),
##   def_rebounds = col_integer(),
##   assists = col_integer(),
##   steals = col_integer(),
##   blocks = col_integer(),
##   turnovers = col_integer(),
##   fouls = col_integer(),
##   efficiency = col_double()
## )
```

```r
teams2 <- teams[ , c('points3', 'points2', 'free_throws', 'off_rebounds', 'def_rebounds', 'assists', 'steals', 'bl
ocks', 'turnovers', 'fouls')]
pca <- prcomp(teams2, scale. = TRUE)
eigs <- data.frame(
  eigenvalue = round(pca$sdev^2, 4),
  proportion = round(((pca$sdev^2)) / (sum(pca$sdev^2)), 4),
  cumprop = cumsum(round((pca$sdev^2) / sum((pca$sdev^2)), 4))
)
pca_table <- as.data.frame(pca$x)
pca_table2 <- mutate(pca_table, team = teams$team)
ggplot(pca_table2, aes(pca_table2$PC1, pca_table2$PC2)) +
  geom_point() +
  labs(x = "PC1", y = "PC2") +
  geom_text(aes(label = team), hjust = 0, vjust = 0) +
  ggtitle("PCA plot (PC1 and PC2)")
```



```
   Each of these '+' is independent of each other. That is to say we can save the variables until the next time we
load them
```

The important thought of ggplot2

- Take the expression of “information” as the core

  Mapping is the process of mapping the information you explore from data (data) to graph. From data information to pixel and color conversion, in ggplot2, it is referred to as scaling. The first thing to keep in mind is that the expression of information is the goal, and the various graphics forms are just tools for its service. You can’t let the tools interfere or even drown out your goals.

- Structured thinking

  Therefore, the command of drawing is also best able to reflect this structure in our thinking, reflecting the continuous addition of this additional information. As long as you make a small update on the original command, you can generate new shapes that add new information. In this way, command and thought are perfectly consistent. If not using a structured way of thinking, for example, general scatterplot and gender-coded scatterplot need to use very different command to write, then it will upset thinking is a thought process. What is important here is the continuous addition of information, and the change of form should be subject to the input of information, without disturbing the input of information. On the other hand, the same information is best expressed in almost the same order. For example, a bar chart is almost the same as a pie chart, so the command to make a bar graph only needs to add a small change, and it should become a pie chart.

- Free combination of basic graphic elements

  Ggplot2 divides basic graphic elements into different geometric objects (geom), different geometric object attributes (aes), etc. Our graph process is to correspond to the information that we need to express to the above various graphical tools. The graphic tools above should be free combinations to express complex and variable information. We can put a bar graph on the scatter plot, then we can differentiate the gender, then we can use the subwindow to distinguish the region, and so on. If this can

> be the combination of freedom, the researchers is no longer restricted in software provides several limited choice, which can express information need according to oneself, use these basic graphic elements to create an infinite variety of new graphics. These new graphics include graphs on scatter plots, gender, area information, and so on, often without a specific name. This free combination certainly gives researchers a lot of room for freedom.

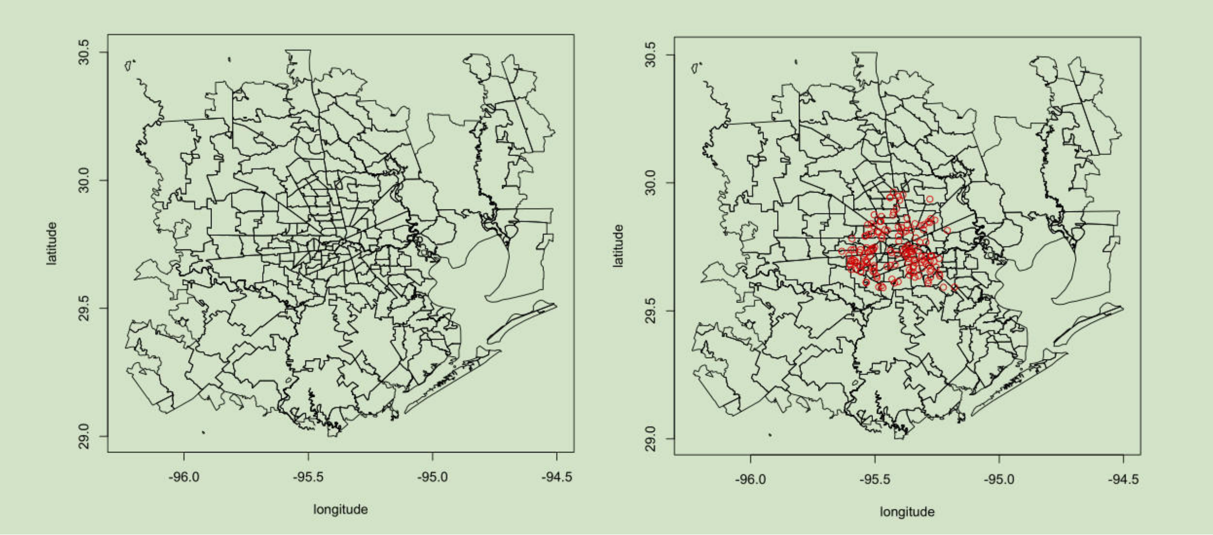
> ## Unique Thing in this post (ggmap)
>
> ggmap took another step in the contextual information of various static maps based on ggplot2. The basic idea of ggmap is to use the downloaded map image, use ggplot2 as the background layer, and then draw data, statistics, or models on the map.

> ### The hierarchical syntax of graphics
>
> - By definition, the hierarchical syntax requires that each diagram be made up of five parts
> - The default data set for aesthetic mappings
> - One or more layers, each of which has a geometric object (“geom”), statistical transformation (“stat”), and a data set with spatial images (which may default)
> - Each spatial image has a scope (which can be generated automatically)
> - A coordinate system
> - A specification

Since ggplot2 can implement the hierarchical syntax of the graph, each of the graphs drawn with ggplot2 has each of these elements. Therefore, the ggmap diagram also has these elements, but some of its elements are fixed to the map component: the X-axis is longitude, the Y-axis is latitude, and the coordinate system is fixed on the Mercator projection. The main theoretical advantage of using layered grammar to map the map is the coordinate scale. In the typical case of a map covering the data range, the latitude and longitude values in the ggmap are limited to the map (by default) and the scale on the axis. Each layer presents the color, filling, alpha mixing and other elements built on top of the map to maintain a scale of consistency. The consistency of grammar is equally important for the faceted diagram, with the aim of making it easier to compare the same elements in several diagrams. , of course, if the user specified spatial data out of the question, then the proportion of size, there are also problems, such as using multiple projection in with a map, and it is difficult to repair these errors.

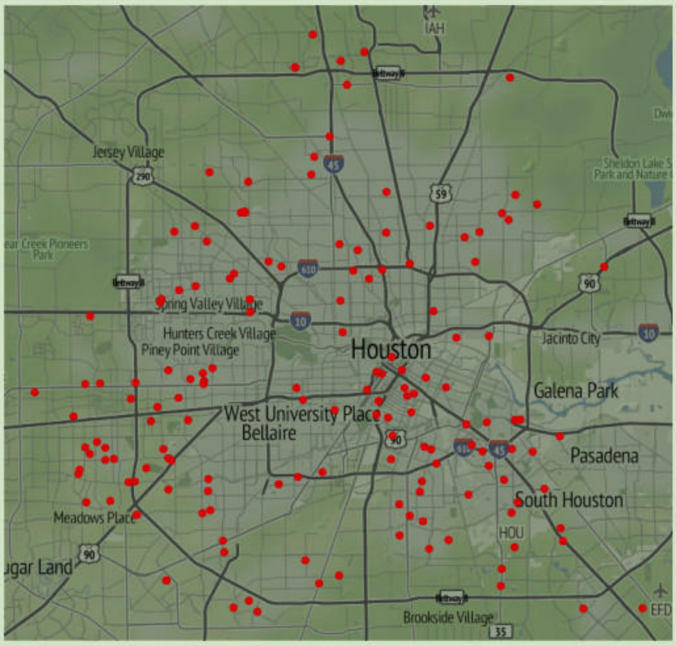
> ### In ggmap, this process is divided into two parts:
>
> - Download the images and format them, using get_map
> - Use the ggmap to complete the drawing

> ### Example



And also

```
murder <- subset(crime, offense == "murder")
qmplot(lon, lat, data = murder, colour = I('red'), size = I(3), darken = .3)
```



Another Unique Thing in this post (geofacet)

1. The main content

- The geofacet package extends the faceted function of ggplot2, which in turn provides a more flexible data visualization scheme based on geographic information. This faceted function is not particularly noted, as the built-in split function (facet_grid, facet_wrap, etc.) is not much different. The only difference is that, in the final graphical layout, a single chart is allowed to depict the location of the corresponding geographical polygon.

2. The core function

- Each of the split cells can present a dimension of data rather than a single value.
- Each subsurface cell can accommodate any type of ggplot2 built-in chart object
- The split-face system supports any geographic polygons (either built-in or user-defined)

Example

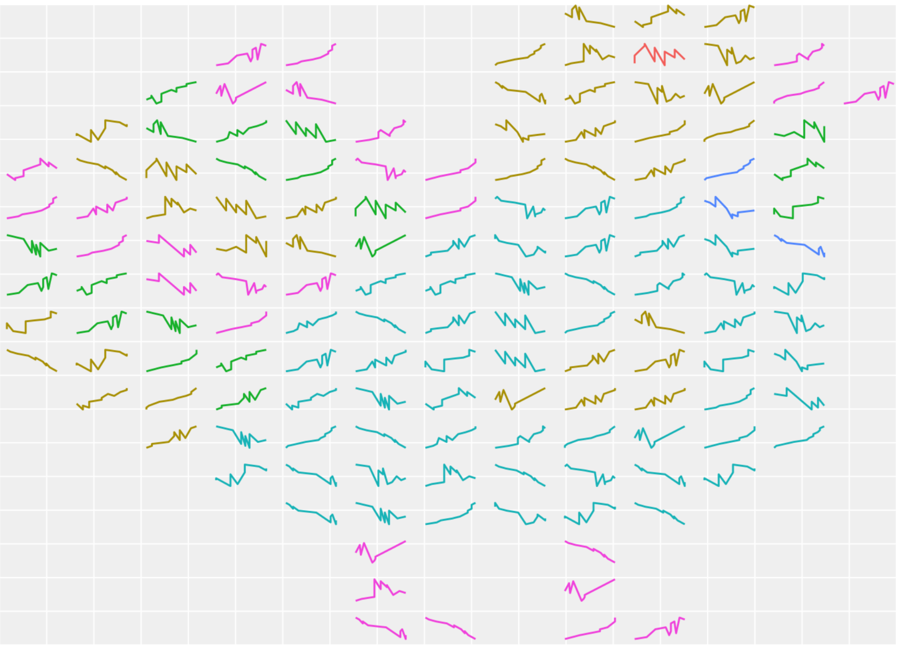
```
Turkey <- read.csv("http://pages.iu.edu/~cdesante/turkey.csv")
colnames(Turkey)[2:1] = c("row", "col")
Turkey$row = max(Turkey$row) - Turkey$row +1
Turkey$name <- Turkey$code <- paste0('turkey', 1:nrow(Turkey))
library(ggplot2)
library(geofacet)
x <- split(eu_gdp, eu_gdp$code)
x <- x[sample.int(length(x), nrow(Turkey), replace=T)]
for (i in 1:length(x)) {
  x[[i]]$code = Turkey$code[i]
}
y <-  do.call(rbind, x)

color = Turkey$Turkey.Colors
names(color) = Turkey$code
y$color = color[y$code]
Turkey = Turkey[, -3]

p1 <- ggplot(y, aes(gdp_pc, year))+ geom_line() +
  facet_geo(~code, grid=Turkey, scales='free')
```

```
## You provided a user-specified grid. If this is a generally-useful
##   grid, please consider submitting it to become a part of the
##   geofacet package. You can do this easily by calling:
##   grid_submit(__grid_df_name__)
```

```
p1 + theme_void() + aes(color=color) + theme(strip.text.x = element_blank(), legend.position='none')
```



From the graph we can see that ggplot2 faceted pixel art!

> ## Conclusion
>
> - Understand and understand the structure of the ggplot code
> - Ggplot2 need a period of time to study, but when you cross the threshold, it can be felt by the concise and elegant, and ggplot2 can through the underlying components structure unprecedented graphics, the limits you just your imagination
> - The core idea of ggplot2 is to separate the drawing from the data, and the graphics-related drawing is separated from the graphics-independent drawing
> - ggplot2 is a layer diagram
> - ggplot2 holds the adjustment function of the imperative graph to make it more flexible
> - ggplot2 integrates common statistical transformations into the drawing.

> ## Reference
>
>
> - https://github.com/tidyverse/ggplot2/wiki/Crime-in-Downtown-Houston,-Texas-:-Combining-ggplot2-and-Google-Maps
> - http://zevross.com/blog/2014/07/16/mapping-in-r-using-the-ggplot2-package/
> - https://dl.acm.org/citation.cfm?id=1795559
> - https://www.rdocumentation.org/packages/geofacet/versions/0.1.5
> - https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
> - http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
> - https://www.statmethods.net/advgraphs/ggplot2.html
>