(Continued) Data visualization with R & ggplot2

Introduction — Motivation, Background and Audience

What if data points overlap with each other when we are trying to make the scatterplot? What if we want to demonstrate both relationship and distribution in the same chart? What if we want to generate a map pinpointing target cities? You'll encounter all kinds of different situations when visualizing data, so it is critical to choose the right type of plot for the specific objectives. Continuing from the last time, this post will dig much deeper into ggplot2 and its advanced applications. As we mentioned previsouly, although base graphics in R offers a good set of plots for simple data visualizations, some of them still require a lot of work. By contrast, ggplot2 is extremely flexible and efficient. With more discussion on the advanced application on ggplot2, I hope you will not only get a more comprehensive understanding of the aesthetic power of ggplot2, but also know what type of plot to use for what sort of problem.

As the most elegant and aesthetically pleasing graphics framework available in R, ggplot2 has been used in many places and for different objectives. Therefore, it is critical to dig deeper with ggplot2 to understand what type of visualization to use for a certain type of question and how to implement it in R using ggplot2. This interactive post is divided into three sections:

- **Section 1: Advanced Customization. This section covers advanced techniques and aesthetic features like Hierarchical Dendrogram and clusters that that haven't been emphasized a lot in the lecture. They are unseful in the sense that they will allow us to group qualititative or quantitative data into distinctive groups.**
- **Section 2: A List of applicable ggplot2 Visualizations. Based on the previous section, this section covers different types of ggplots, including population pyramid, diverging bars and even area charts that haven't been discussed in class before.**

Given the level of complexity in section 2 and 3, this post is primarily geared towards Stats 133 students at Berkeley and those who have some knowledge of the R programming language and want to make advanced graphs based on complex data sets.

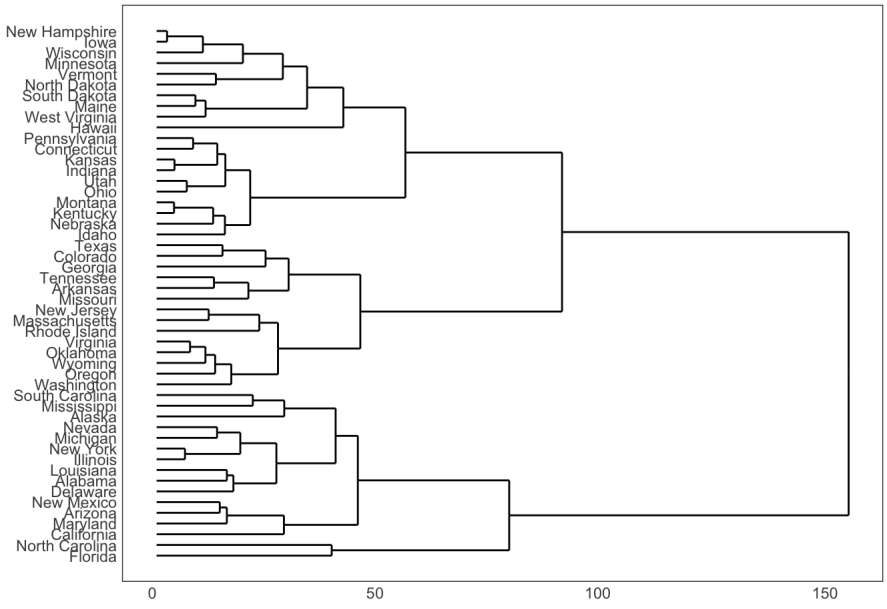
Examples and Discussion — Section 1

1. Hierarchical Dendrogram First to construct a hierarchical dendrogram we need to learn how to interpret this specific data visualization method and know when to use it. The dendrogram is fairly simple to interpret. The **horizontal axis** of the dendrogram represents the dissimilarity between clusters. Specifically, the horizontal position of the split gives the distance (dissimilarity) between the two clusters. The **vertical axis** represents the objects and clusters. Our main interest here is in similarity and clustering. Each fusion of two clusters is represented on the graph by the splitting of a horizontal line into two horizontal lines.

```
# To begin, we need to install the package "ggdendro"
library(ggplot2)
library(ggdendro)
theme_set(theme_bw())

# hierarchical clustering
hc <- hclust(dist(USArrests), "ave")

# plot
ggdendrogram(hc, rotate = TRUE, size = 2)
```



Discussion:

A word on the data "USArrests"(Violent Crime Rates By US State): This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas. Several points to clarify:

- The y-axis is a measure of closeness of either individual data points or clusters.
- The greater the difference horizontally, the more dissimilarity. If you're interested in learning more about the logic behind hierarchical dendrogram, here is a good video to go over: https://www.youtube.com/watch?v=4OnfJmC-aUI

2. Clusters: When presenting the results, sometimes we would want to encirlce certain special group of points or region in the chart so as to draw the attention to those peculiar cases. This can be conveniently done using the geom_encircle() in ggalt package. How to use geom_encircle()? Within geom_encircle(), we set the data to a new dataframe that contains only the points (rows) or interest. The color and

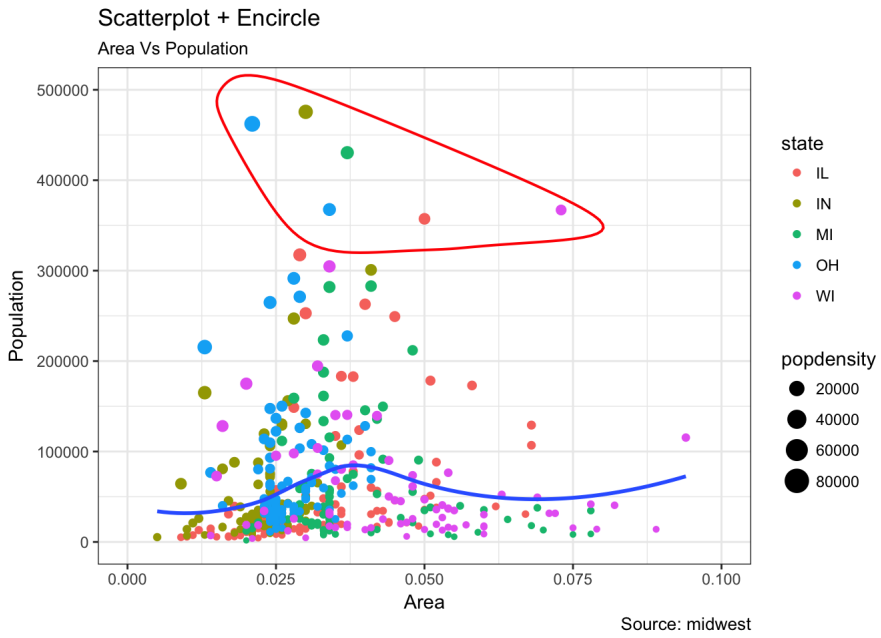
thickness of the curve can be modified as well. For example,

```
# install "ggalt" package
# devtools::install_github("hrbrmstr/ggalt")
options(scipen = 999)
library(ggplot2)
library(ggalt)
midwest_select <- midwest[midwest$poptotal > 350000 & midwest$poptotal <= 500000 & midwest$area > 0.01 & midwest$a
rea < 0.1, ]

# Plot
ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +
# draw points
  geom_smooth(method="loess", se=F) +
  xlim(c(0, 0.1)) +
  ylim(c(0, 500000)) +
# draw smoothing line
  geom_encircle(aes(x=area, y=poptotal),
                data=midwest_select,
                color="red",
                size=2,
                expand=0.08) +
# encircle
  labs(subtitle="Area Vs Population",
       y="Population",
       x="Area",
       title="Scatterplot + Encircle",
       caption="Source: midwest")
```

```
## Warning: Removed 15 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 15 rows containing missing values (geom_point).
```



Although it is possible to show the distinct clusters or groups using geom_encircle(). It is hard for the function to capture weak features. If the dataset has multiple weak features, we can compute the principal components and draw a scatterplot using PC1 and PC2 as X and Y axis.

```
# devtools::install_github("hrbrmstr/ggalt")
library(ggplot2)
library(ggalt)
library(ggfortify)
```

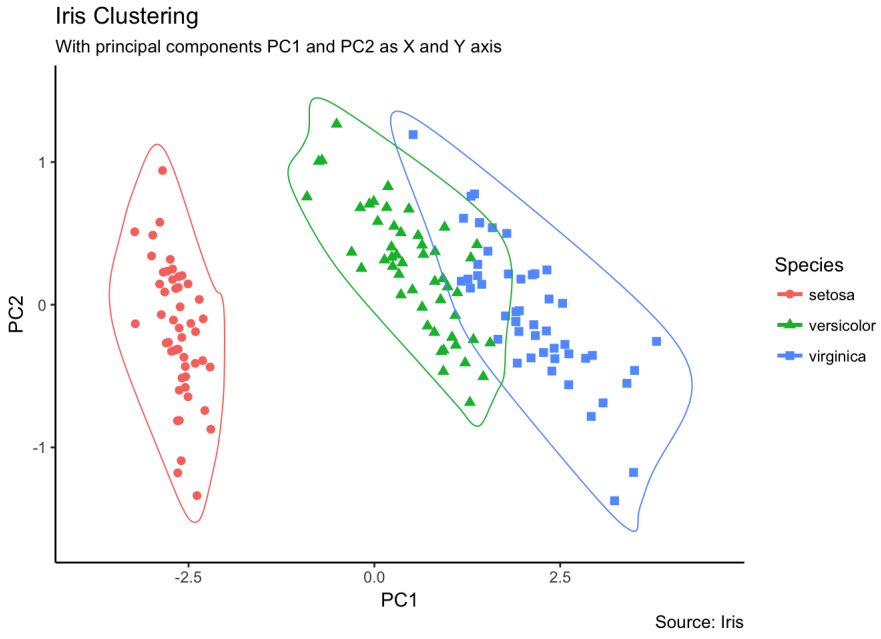
```
## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'quiet'

## Warning: namespace 'DBI' is not available and has been replaced
## by .GlobalEnv when processing object 'quiet'
```

```
theme_set(theme_classic())

# Compute data with principal components
df <- iris[c(1, 2, 3, 4)]
# compute principal components
pca_mod <- prcomp(df)
# Data frame of principal components
# dataframe of principal components
df_pc <- data.frame(pca_mod$x, Species=iris$Species)
# df for 'virginica'
df_pc_vir <- df_pc[df_pc$Species == "virginica", ]
# df for 'setosa'
df_pc_set <- df_pc[df_pc$Species == "setosa", ]
# df for 'versicolor'
df_pc_ver <- df_pc[df_pc$Species == "versicolor", ]

# Plot
ggplot(df_pc, aes(PC1, PC2, col=Species)) +
  geom_point(aes(shape=Species), size=2) +
# draw points
  labs(title="Iris Clustering",
       subtitle="With principal components PC1 and PC2 as X and Y axis",
       caption="Source: Iris") +
# change axis limits
  coord_cartesian(xlim = 1.2 * c(min(df_pc$PC1), max(df_pc$PC1)),
                  ylim = 1.2 * c(min(df_pc$PC2), max(df_pc$PC2))) +
# draw circles
  geom_encircle(data = df_pc_vir, aes(x=PC1, y=PC2)) +
  geom_encircle(data = df_pc_set, aes(x=PC1, y=PC2)) +
  geom_encircle(data = df_pc_ver, aes(x=PC1, y=PC2))
```



Examples and Discussion — Section 2

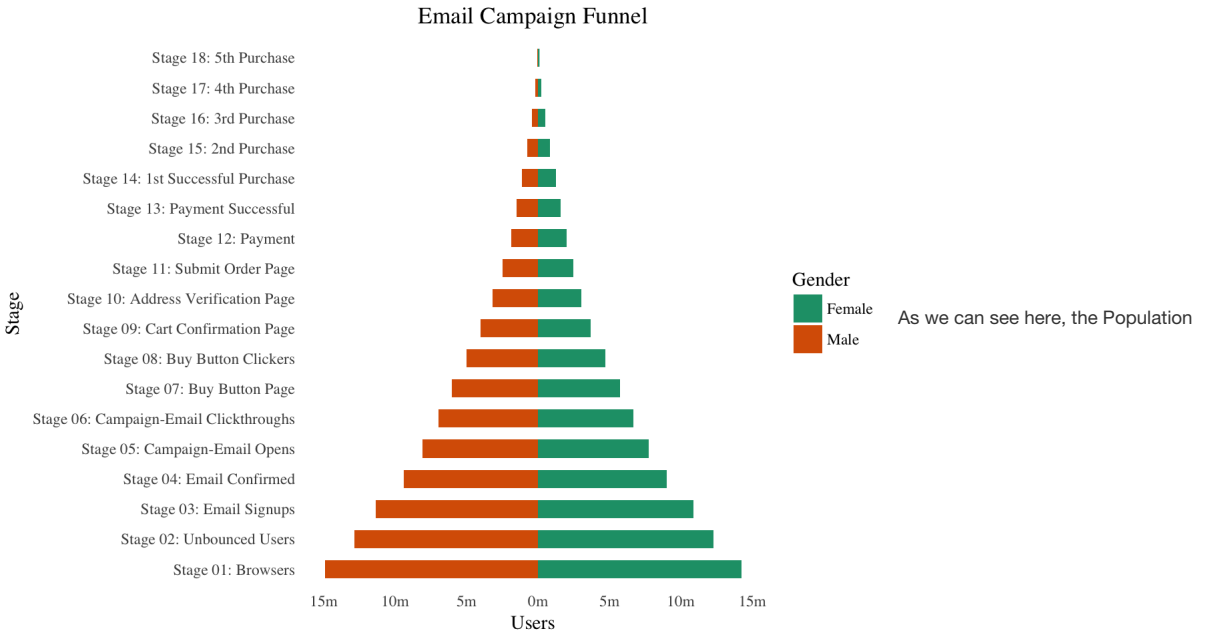
1. More real life application in **marketing**: Population Pyramid Population pyramids offer a unique way of visualizing how much population or what percentage of population fall under a certain category. The below pyramid is an excellent example of how many users are retained at each stage of a email marketing campaign. The marketing data can be easily accessed here: https://raw.githubusercontent.com/selva86/datasets/master/email_campaign_funnel.csv

```
library(ggplot2)
library(ggthemes)
# turns of scientific notations like 1e+40
options(scipen = 999)

# Read data
email_campaign_funnel <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/email_campaign_funnel
.csv")

# X Axis Breaks and Labels
brks <- seq(-15000000, 15000000, 5000000)
lbls = paste0(as.character(c(seq(15, 0, -5), seq(5, 15, 5))), "m")

# Plot
ggplot(email_campaign_funnel,
# Fill column
       aes(x = Stage, y = Users, fill = Gender)) +
# draw the bars
                              geom_bar(stat = "identity", width = .6) +
# Breaks
                              scale_y_continuous(breaks = brks,
# Labels
                                                 labels = lbls) +
# Flip axes
                              coord_flip() +
                              labs(title="Email Campaign Funnel") +
# Tufte theme from ggfortify
                              theme_tufte() +
                              theme(plot.title = element_text(hjust = .5),
# Centre plot title
                                    axis.ticks = element_blank()) +
# Color palette
                              scale_fill_brewer(palette = "Dark2")
```



As we can see here, the Population pyramids are useful in the sense that it can be used real business and marketing research dynamically. If you want to learn more about pipulation pyramid, here is a more interactive website for you to visit: https://rpubs.com/walkerke/pyramids_ggplot2

2. More real life application in **finance**: Area Chart Area charts are typically used to visualize how a particular metric (such as % returns from a stock) performed compared to a certain baseline. Other types of %returns or %change data are also commonly used. The function geom_area() will useful here.

```
library(ggplot2)
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 3.4.2
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

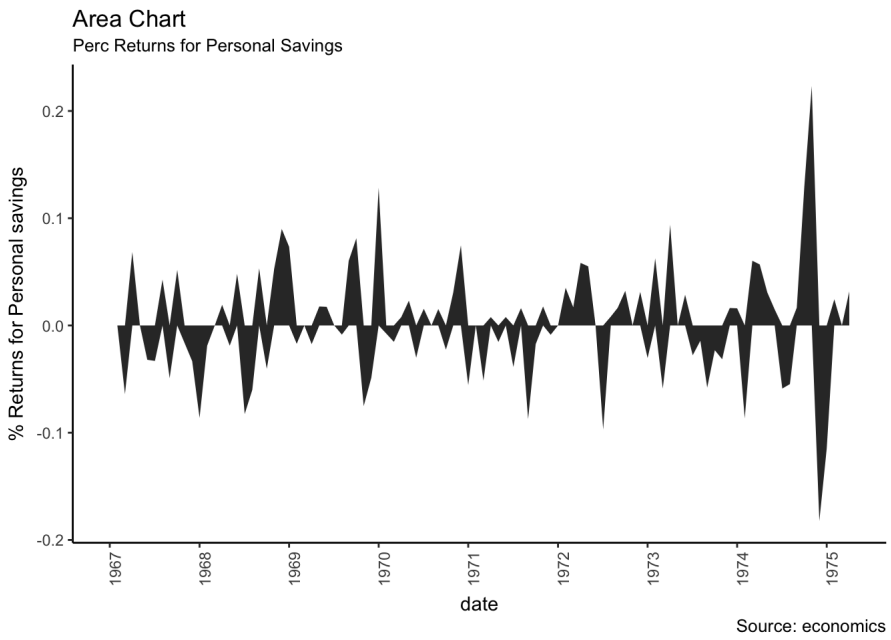
```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
data("economics", package = "ggplot2")

# Compute % Returns
economics$returns_perc <- c(0, diff(economics$psavert)/economics$psavert[-length(economics$psavert)])

# Create break points and labels for axis ticks
brks <- economics$date[seq(1, length(economics$date), 12)]
lbls <- lubridate::year(economics$date[seq(1, length(economics$date), 12)])

# Plot
ggplot(economics[1:100, ], aes(date, returns_perc)) +
  geom_area() +
  scale_x_date(breaks=brks, labels=lbls) +
  theme(axis.text.x = element_text(angle=90)) +
  labs(title="Area Chart",
       subtitle = "Perc Returns for Personal Savings",
       y="% Returns for Personal savings",
       caption="Source: economics")
```



3. More real life application (when we have both positive and negative values): diverging bars Diverging Bars is a bar chart that can handle both negative and positive values, which makes it extremely useful in real life application when we want to compare each individual data point with the average of the population. This can be implemented by a smart tweak with geom_bar(). Yet the usage of geom_bar() can be quite confusing because it can be used to make a bar chart as well as a histogram.

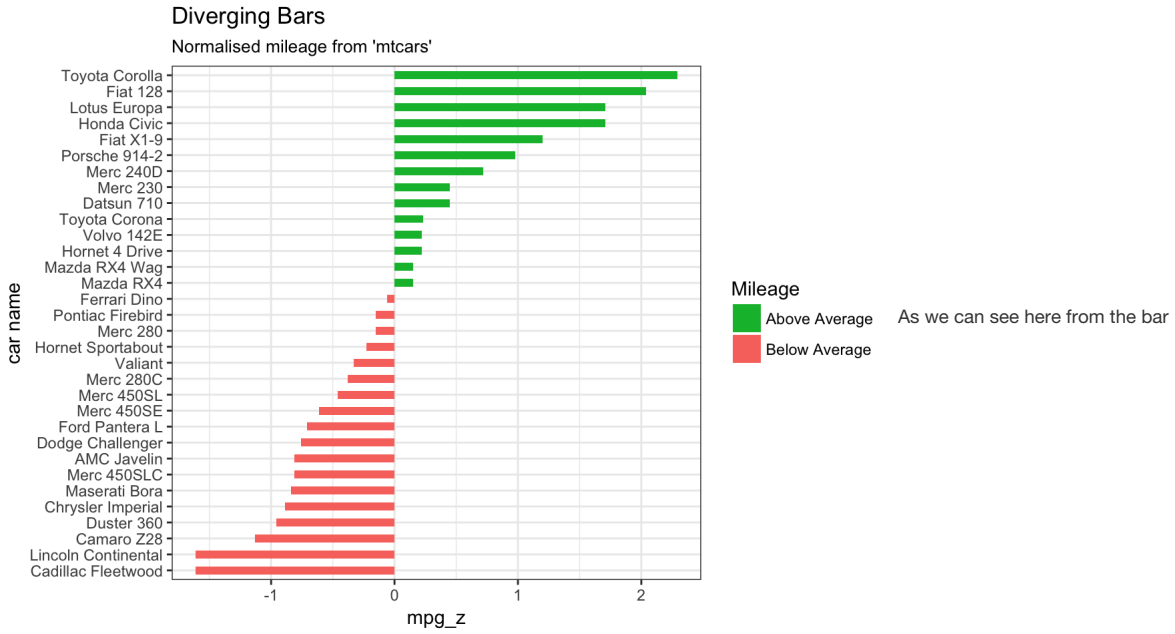
By default, geom_bar() has the stat set to count. That means, when you provide just a continuous X variable (and no Y variable), it tries to make a histogram out of the data. Therefore, in order to make a bar chart create bars instead of histogram, we need to do two things. Set stat=identity. Provide both x and y inside aes() where, x is either character or factor and y is numeric. In order to make sure we get diverging bars instead of just bars, we also need to make sure that our categorical variable has 2 categories that changes values at a certain threshold of the continuous variable.

In the example below, the mpg from mtcars dataset is normalised by computing the z score. Those vehicles with mpg above zero are marked green and those below are marked red.

```r
library(ggplot2)
theme_set(theme_bw())

# Data Prep
data("mtcars")  # load data
mtcars$`car name` <- rownames(mtcars)  # create new column for car names
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg), 2)  # compute normalized mpg
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")  # above / below avg flag
mtcars <- mtcars[order(mtcars$mpg_z), ]  # sort
mtcars$`car name` <- factor(mtcars$`car name`, levels = mtcars$`car name`)  # convert to factor to retain sorted o
rder in plot.

# Diverging Barcharts
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_bar(stat='identity', aes(fill=mpg_type), width=.5)  +
  scale_fill_manual(name="Mileage",
                    labels = c("Above Average", "Below Average"),
                    values = c("above"="#00ba38", "below"="#f8766d")) +
  labs(subtitle="Normalised mileage from 'mtcars'",
       title= "Diverging Bars") +
  coord_flip()
```



chart, the use of diverging bar chart can offer a clear insight into the distribution of individual data point with respect to the average of the population and thus helpful in the data analysis stage in the end. Here is a step-by-step tutorial for you in case you're lost: https://www.youtube.com/watch?v=ynhRqI3_iwU

Conclusion — Take-home Message

> Although base graphics in R offers a good set of plots for simple data visualizations, some of them still require a lot of work. By contrast, ggplot2 is extremely flexible and efficient. With this two-part post on ggplot2, I hope you not only got a more comprehensive understanding of the aesthetic power of ggplot2, but also started to see the great power of ggplot2 in real life application including marketing, finance and other research projects. You'll encounter all kinds of different situations when visualizing data, so it is critical to choose the right type of plot for the specific objectives.

References

- https://stats.stackexchange.com/questions/82326/how-to-interpret-the-dendrogram-of-a-hierarchical-cluster-analysis
- http://www.statisticshowto.com/hierarchical-clustering/
- https://www.rdocumentation.org/packages/datasets/versions/3.4.1/topics/USArrests
- https://www.youtube.com/watch?v=4OnfJmC-aUI
- https://raw.githubusercontent.com/selva86/datasets/master/email_campaign_funnel.csv
- https://rpubs.com/walkerke/pyramids_ggplot2
- https://www.youtube.com/watch?v=ynhRqI3_iwU