post01-Thanh-La

Data Visualization

```
library(ggplot2)
library(dplyr)
library(plotly)
setwd("/Users/thanh/Desktop/Stat133/stat133-hws-fall17/post01")
```

We have seen in our Stat 133 class, potentially in other classes as well, the importance of data visualization. Often times numerical summaries are not informative or sufficient to confirm our suspicion about the data we have. The goal of this post is to discuss current topics and thems of data visualization that I found interesting. At the end of the post I will go through a tutorial with the plotly package.

Take for example

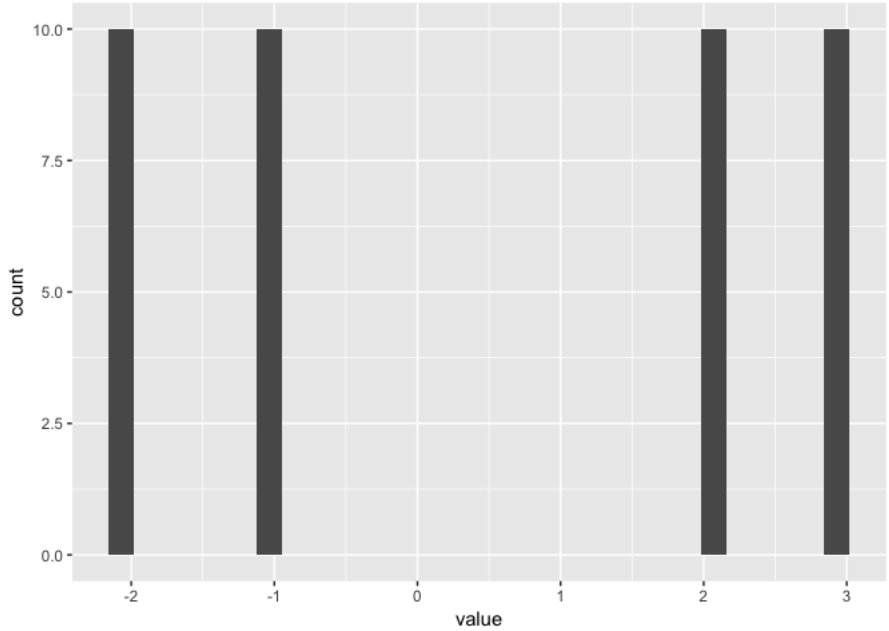
```
#summarizing our data
vec_temp <- rep(c(-1, -2, 2, 3), 10)
vec_temp <- as.data.frame( vec_temp)
summary(vec_temp)
```

```
##     vec_temp
##  Min.   :-2.00
##  1st Qu.:-1.25
##  Median : 0.50
##  Mean   : 0.50
##  3rd Qu.: 2.25
##  Max.   : 3.00
```

In this case, it is not as obvious that we have a symmetry in the data unless we see the distribution.

```
#generate histogram of our data
ggplot(data=vec_temp, aes(vec_temp)) + geom_histogram() + xlab("value")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



With the help of the ggplot2 package as we have learned how to use in our class, we can see a symmetric distribution of the occurences.

As you can see, data visualization is very important in helping gain more insight about our current data.

We have already explored so many uses of ggplot2. In this post I would like to provide more information about data visualization by providing some methods of creating more graphs. Something more dynamic and interactive.

To check out more existing tools, follow the link ggplot2 to see what is not included in ggplot2.

Package: plotly

I would like to provide examples and information about the plotly package, however, before going through the examples, there are some relevant topics I would like to bring up about data and the role it serves and has served in the past.

According to Stephen Few in his article, there are 8 core principles in the visualization of information:

- **Simplicity**
- **Comparison**
- **Highlights**
- Exploration

- View Diversity
- Why
- Skepticism
- Response

Those bolded have been agreed to be the biggest core principles in data visualization.

Can you think about which go to ggplot2 functions you can use to qualify these 8 principles?

Since the revolution of data visualization, thanks to the computer and advancements of schema designs and methods of highlighting data, we have arrived to a new era of data. With current software tools and packages, we have the capabilities draw out information from data and make them more easily grasped by including visual interpretations. The way we look at data can be reshaped and portrayed in so many ways that the concept of data story-telling has been termed and put practice in recent times. You could think of data visualization as a form of art. As any good art admirer would know, each piece of work tells a story

We can think of the process of data analysis as

1. Collecting and cleaning
2. Analysis
3. Data articulation

Data articulation can be broken into 3 forms of story telling

- Story telling
- Story triggering
- Story listening

As with the lack of insight the summary() operator provided for our generic data, solely analyzing data will not produce strong enough conclusion on its own, even with visualization. The idea behind data stories is that you apply reason and relationship between the data make something more sensible. This may or may not involve intuition and/or guessing.

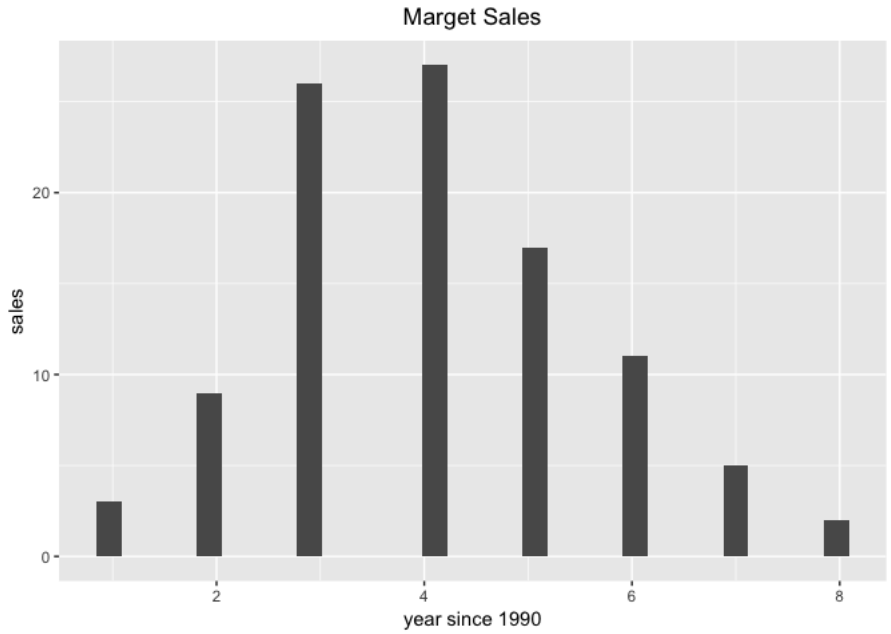
Take for example this hypothetical situation of sales

```
set.seed(100)
df_temp <- as.data.frame(rbinom(100, 20, 0.2))

ggplot(data=df_temp, aes(x=df_temp)) + geom_histogram() + ylab("sales") + xlab("year since 1990") + ggtitle("Marge
t Sales") + theme(plot.title = element_text(hjust = 0.5))
```

```
## Don't know how to automatically pick scale for object of type data.frame. Defaulting to continuous.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



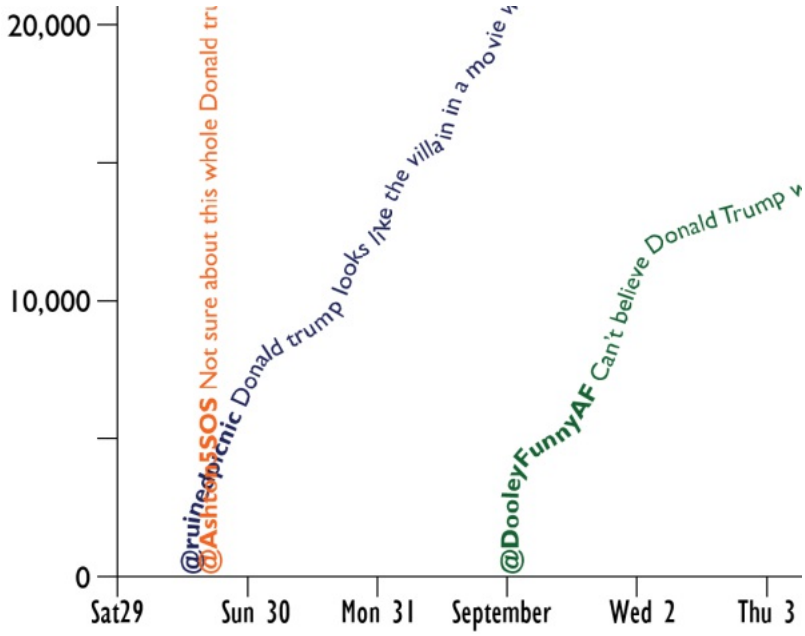
Analytically, you would only see the decrease in sales. However, if you knew that around 1994 there was a competing company on the rise that won over all loyal customers brought in before, you would quickly conclude that the fall in sales were due to competition.

The motivation of this example is to show that throughout the process of data analysis, each stage can be associated with a story as a guide and tool for insight, which potentially can lead to the conclusion of either proving or disproving suggested hypotheses. This application of contextualizing data helps decision makers better understand the numeric figures when the decision making on behalf of the organization takes place.

You can find more information and examples about data stories by following this link Data Story

The purpose of introducing Data story is to exemplify the power of data visualization and one consequence of the concept.

I would like to highlight that this concept of data story is only one way in which people are utilizing data visualization. On this note, I would like to cite an image from richardbrath's page on wordpress that discusses The Design Space of Typographic Data Visualization.



You may or may not have seen this image before, however, I find it extremely thought provoking in the sense that the bounds of traditional graph plotting are being crossed. Whose to say that that points provide more information than some arbitrary string, however, a non-arbitrary string would surely add more information to the plot. This type of trend-setting or trend-breaking, however you call it, should make you wonder and maybe even come up with new ways to express information much easier.

Something to further alleviate a possible suspicion about how something reshaped but is essentially the same in terms of the data objects itself is sas. SAS stands for software as service, it is a concept and itself an existing company that offers software visualization tools. These tools are sold to a range of customers, from small owners to large enterprises. Their goal is to speed up data interpretation, identify trends and patterns. However, it also mentions the idea of communicating the 'story' of data to others. There are so many tools and many more ideas to be realized than what we have seen with our good friend ggplot2.

I point you to the sas website to see other guidelines and tips for using data visualization. You can follow the link sas. The information accesible on the website should serve as inspiration and a form of reassurance about the language which we should begin adopting to talk about data.

Lastly, before beginning the tutorial, I would like to mention that R already has many power tools and packages. It should be noted that R was based off of older languages like the S language as you may know, additionally with the use of fortran and C, R has become quite popular for data analysis.

Why stop there? This is exactly what the plotly package tries to do. It uses DOM objects for a more dynamic approach at visualizing data. DOM stands for document object model, it treats HTML, XHTML, and XML document as tree structure, such structure containing nodes that in itself contains information.

Why is this important? The plotly package actually utilizes the DOM structure and javascript to create what is called D#.js objects, which is essentially our graphs. Due to the high utility of the DOM structure, we can edit the viewed data quickly and even make edits to our observations.

The D3.js objects in itself deserves attention, but for those who are more familiar with the Javascript language, you can click on D3.js to learn more.

Tutorial: plotly graphs

You can download the package by running install.packages("plotly") in R studio to get start using the added features for free. This package adds on to and interacts with ggplot2 objects to create the D3.js objects aforementioned.

We can create graphs on the nba data that we have been studying.

```
#read data
nba_data <- read.csv('../Lab04/nba2017-players.csv')
#plot with plotly package
plot_ly(data=nba_data, x = ~points, y = ~salary, mode = "markers", color = ~age, type='scatter') %>%
  layout(title='Salary w.r.t. Points NBA players 2017') %>%
  config(showLink=TRUE)
```

In this graph, we see more players under the age of 40 present, and despite the age, the general trend in salary with respect to point appears to to have no influence from the age category.

If you hover over the image, you can find many functionailities for analyzing and altering the plot. You can even play with the data or recreate another graphical object by clicking `Edit chart`. This is just like the shiny apps we learned about in class. Just as dynamic, and less coding!!!

note I used %>% operator to pipe the objects

Let's see a couple more examples

In this next example we will show how we can use existing ggplot2 objects to create plotly objects

```
#data clean: filter 3 groups for data analysis
nba_small <- filter(nba_data, team =='GSW' | team == 'LAL' | team == 'BOS')
nba_small <- as.data.frame(nba_small)
nba_small$points <- nba_small$points - mean(nba_small$points)
nba_small$salary <- nba_small$salary - mean(nba_small$salary)
gg_plot <- ggplot(data=nba_small, aes(x=points, y=salary)) +
  geom_point() +
  ggtitle('points centered vs. salary centered nba2017') +
  geom_smooth(aes(colour = team, fill = age)) + facet_wrap(~ team)
#cast into plotly object
ggplotly(gg_plot) %>%
  config(showLink=TRUE)
```

This graph displays higher pay for the Golden State warriors in comparison to Boston and LA Lakers. This model does not display a linear trend, which leads us to believe that salary is determined by more than just the points that the players score in the respective teams.

The interaction between ggplot2 and plotly objects are very natural and can be deemed very useful, as these graphs are also interactive and dynamic.

Lastly, we will see a 3 dimensional object

```
#clean data
nba_data <- select(nba_data, points, age, salary)
nba_data$salary <- nba_data$salary - mean(nba_data$salary)
nba_data$points <- nba_data$points - mean(nba_data$points)
nba_data <- slice(nba_data, 1:20)
#form matrix for input argument
salary_matr <- as.matrix(nba_data$salary)
points_matr <- t(as.matrix(nba_data$points))
#generate wave values for input argument
waves <- matrix(c(cos(salary_matr %*% points_matr) +
                sin(salary_matr %*% points_matr)),
                nrow = 20, ncol = 20)
#graph plotly object
plot_ly(x = nba_data$points,
        y = nba_data$salary,
        z = waves,
        type='surface') %>%
          config(showLink=TRUE)
```

```
cor(nba_data$salary, nba_data$points)
```

```
## [1] 0.4493748
```

This map is a nice looking graph However, the height of the graph shows no immediately obvious trend. This could be reflected in the somewhat lower correlation among the two data vectors. After linear transformations then sinusoidal transformations, the values could be less obviously correlated.

There are so many more graphical objecsts you can use. These are but a few. To explore more, check out the link plotly

Summary

Data visualization is a very powerful tool and is beginning to pick up in trend. In addition to providing insight and helping better understand data, it contributes to a new culture of story telling with data to better characterize data for informative decisions. Tools such as plotly is one but many. The pro to using plotly is that it interacts well with ggplot2 objects, so it is only natural to explore packages that work with ones we already use often and really enjoy. Package ggplot2 in itself is already very powerful, however, some graphics are still very mediocre when displaying data with ggplot2. Package plotly also adds useful functionalities at an immediate hover and click. You could download the picture, edit the data, zoom, rescale, change axis and many more.

I hope you enjoyed my post, thank you for your time.

References: Unmasked Links

- https://www.statmethods.net/advgraphs/ggplot2.html
- https://www.tableau.com/blog/stephen-few-data-visualization
- https://www.linkedin.com/pulse/visualization-redefining-how-we-view-data-analysis-done-saket-kumar
- http://www.anecdote.com/2016/08/stories-data-storytelling/
- https://richardbrath.wordpress.com/2016/08/13/the-design-space-of-typographic-data-visualization/
- https://medium.freecodecamp.org/which-languages-should-you-learn-for-data-science-e806ba55a81f
- https://www.sas.com/en_us/insights/big-data/data-visualization.html
- https://plot.ly/
- https://d3js.org/
- https://moderndata.plot.ly/interactive-r-visualizations-with-d3-ggplot2-rstudio/
- https://medium.com/datavisualization/the-10-best-data-visualization-articles-of-2016-and-why-they-were-awesome-ce30618ea06a