Why Ggplot2 is Awesome

Mark Hashimoto

10/29/2017

Introduction/Motivation

As you can clearly see from the title, this blog is going to be focusing on our friendly neighborhood R package **ggplot2**.

Upon hearing that we would write a post about a stat133 topic of our choosing, I decided I would want to write about something that wouldn't get unbearably dry for me over the course of two plus weeks of research. Sorry vector atomicity. I remembered being particularly intrigued by the coding processes that ggplot2 used to construct those pretty-looking data graphs we see everywhere, so eventually I figured digging into data visualization was my best choice.

Thus, it will my goal of this blog to give you a solid idea of why ggplot2 is so cool; covering what it is, how it works, and how people use it in our big big world of data.

Background/ Meet ggplot2

To start off the origin story of our lovely R studio package I will show you this picture.

Hadley Wickham

'Who is this handsome devil?', you may ask. This is R-studio chief scientist Hadley Wickham, who created ggplot2 in 2005 with the intent of making data visualization code simpler. He currently lives in Houston, Texas and is an adjunct professor of Statistics at Stanford, Rice and University of Aukland. He is also responsible for the development of other awesome packages we've seen like 'dplyr' (slicing, filtering, and mutating tables), 'tidyr' (tidying data), 'stringr' (working with strings), and readr (that convenient package we used to import tables using read_csv). Basically, this guy is a gracious data lord who wishes to make data science easier for all in the land, and we are very grateful.

Side note: I was a bit curious as to why ggplot2 had the '2' included; I figured ggplot2 was the sequel to a ggplot/ggplot1, but could not find any information on such a package. Eventually I discovered that ggplot2 was originally called ggplot, but after a major revision during its initial developement the title was changed to ggplot2.

How it works

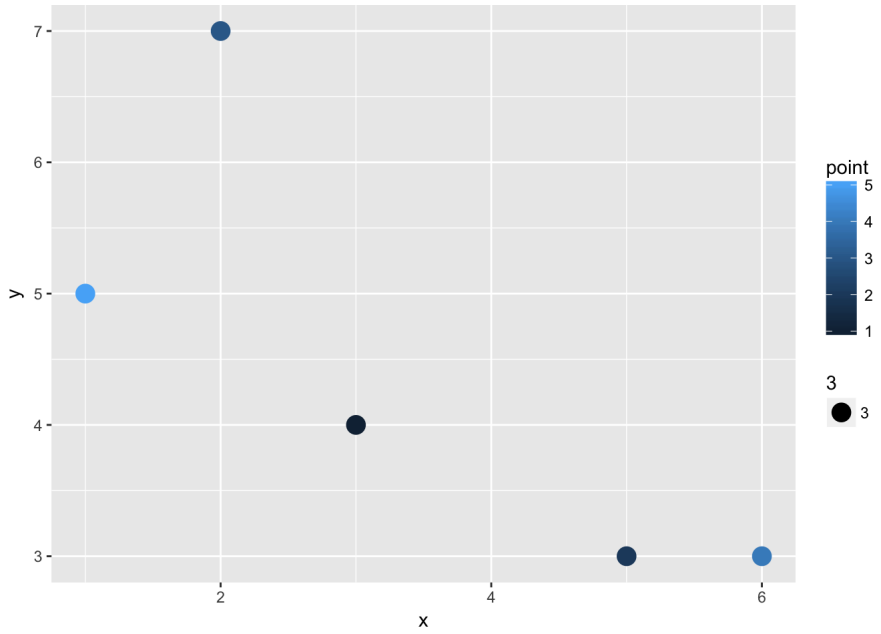
The main principles of ggplot2 are founded on the concepts of statistical visualization coined by computer scientist Leland Wilkinson in his book, "The Grammer of Graphics". I always thought 'ggplot2' was a random coding-type name that made no sense, but it turns out the "gg" comes from Grammer of Graphics. It all makes sense now.

In his work, Wilkinson takes all kinds of statistical graphics (bar, pie, histogram, line, etc.) and reduces them down to a set of key features, known as the grammar. To describe this grammar in simple terms, basically all statistical graphs can be boiled down to mapping **data values** to **aesthetic features** (colors, shapes, etc.) of **geometric objects** (points, lines, bars, etc). I will go more indepth on this grammar later on, but for now I will use some simple code to give you a better idea of what this "aesthetic mappings" business and "geometric object" mumbo jumbo is all about.

Take for example, we have a set of coordinates

```
point = c(1,2,3,4,5)
x = c(3,5,2,6,1)
y = c(4,3,7,3,5)
coord = cbind.data.frame(point,x,y)

example = ggplot(data = coord, aes(x = x, y= y, color = point, size = 3)) + geom_point()
example
```



Here, our data values are the x and y coordinates of each of our points. Our values include the number of point (1-5), and their respective locations. Here we map these data values (which point) to an aesthetic feature (our color scale from light to dark blue) of a geometric object (points) onto a coordinate system (x and y positions).

Deeper into how Ggplot2 works

The package Ggplot2 consists of two plots

1. qplot() - quick plot
2. ggplot() - grammar of graphics plot

qplot

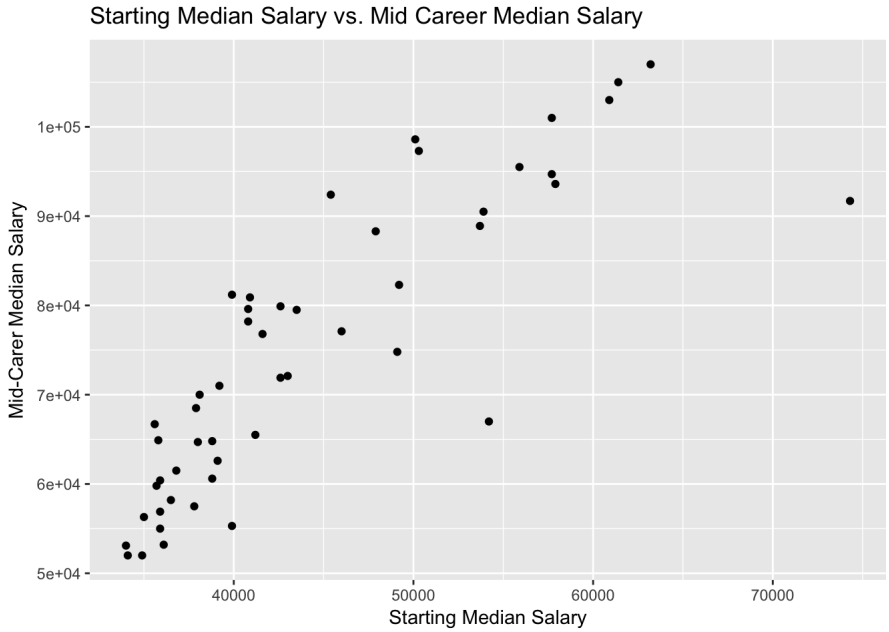
As the name suggests, quick plot is used for simpler plots. The quick plot is used for constructing graphs more quickly than those of ggplot; however this comes at a price: qplot does not have as many features as ggplot.

qplot takes the following arguments:

1. x - vector of x coordinates
2. y - vector of y coordinates
3. data - data containing x and y vectors
4. color/shape/size/fill - used to split data by aesthetic features
5. alpha - used to describe transparency
6. geom - defines type of object
7. method/formula - used to define regression lines
8. facets - used to split data into multiple graphs
9. xlim/ylim - defines x and y axis limits
10. xlab/ylab/main/sub - used to label axes and titles

Lets use some code to give an idea of how we can use qplot and its arguments to constuct a nice looking graph. For our data we will use a set listing college majors and their respective starting median salaries and mid-career median salary. The data was collected from a recent year-long survey from 1.2 million people.

```
q = qplot(data = dat, x = dat$`Starting Median Salary`, y = dat$`Mid-Career Median Salary`, xlab = 'Starting Media
n Salary', ylab = 'Mid-Carer Median Salary', main = 'Starting Median Salary vs. Mid Career Median Salary')
q
```



Here we plot the starting salary versus mid-career salary for all the different types of majors. Here, I've used the arguments x, y, xlab, ylab, and main.

Note: I tried to figure out if there was a way to label each point by its respective major name to no avail, so I will leave that as an inability part of the qplot arguments. I also tried to graph a line (y=x) to show if in general, jobs had a a higher or lower mid-career salary compared to starting salary, but I found this was also not an option within the given arguments of qplot.

ggplot

To give an idea of how ggplot and its arguments works, I will use ggplot on the same salary by major data set from our qplot example.

Ggplot consists of both a 'ggplot()' object and additional 'geom' layers. The ggplot objects defines the data and x,y variables we will be visualizing, and each geom layer defines which geometric shapes we assign this data to.

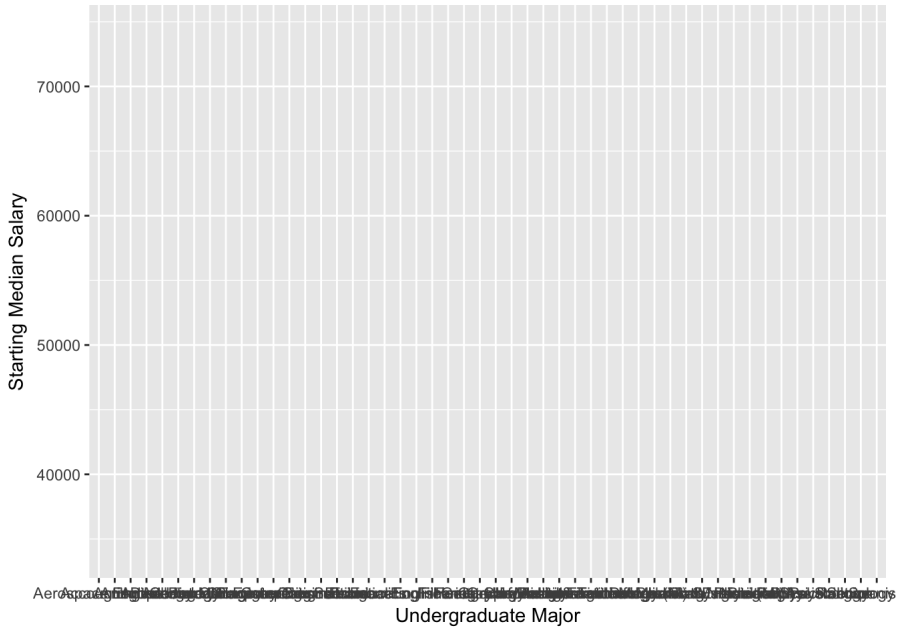
ggplot differs from qplot mainly in that instead of including all graphic parameters into one object, you can add different graphical parameters with additional layers using the + operator.

ggplot() Arguments:

aes: what data values are mapped aesthetically, includes x and y vectors

data: data fram with x and y vectors

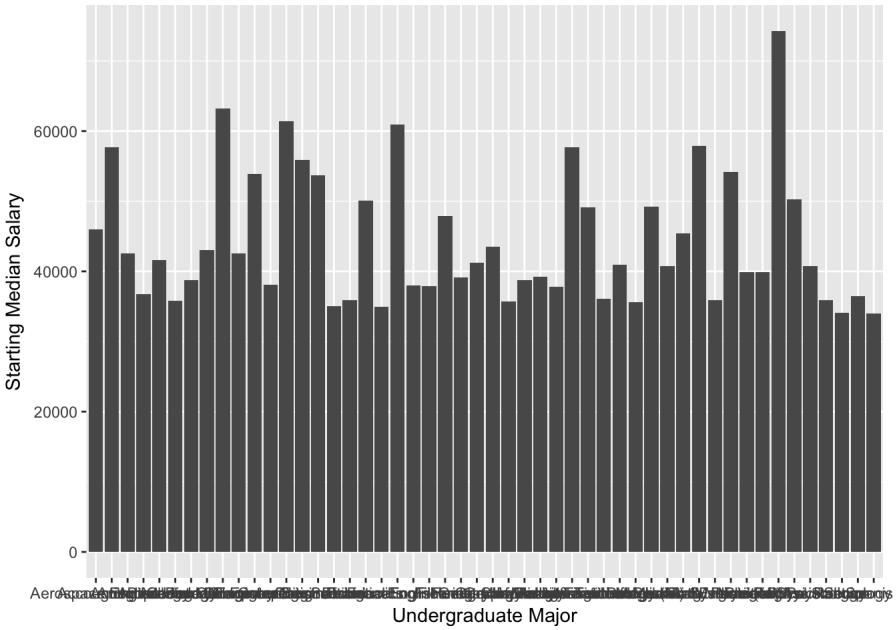
```
gg = ggplot(data = dat, aes(x = `Undergraduate Major`, y = `Starting Median Salary` ))
gg
```



Looking at this graph, we notice there is **alot** wrong with it. Lets see if we can fix it up and make it look nice and pretty. To start, our graph is looking pretty empty.Thats because we have only defined what data we will be using, but haven't defined how we will map our data onto the coordinate system.

So lets add a 'geom' layer to make beef this plot up a little. By using the '+' operator, we can connect an aesthetic value (the x and y values in our ggplot object) to a geometric object. We will use 'geom_bar' to make it a bar graph.

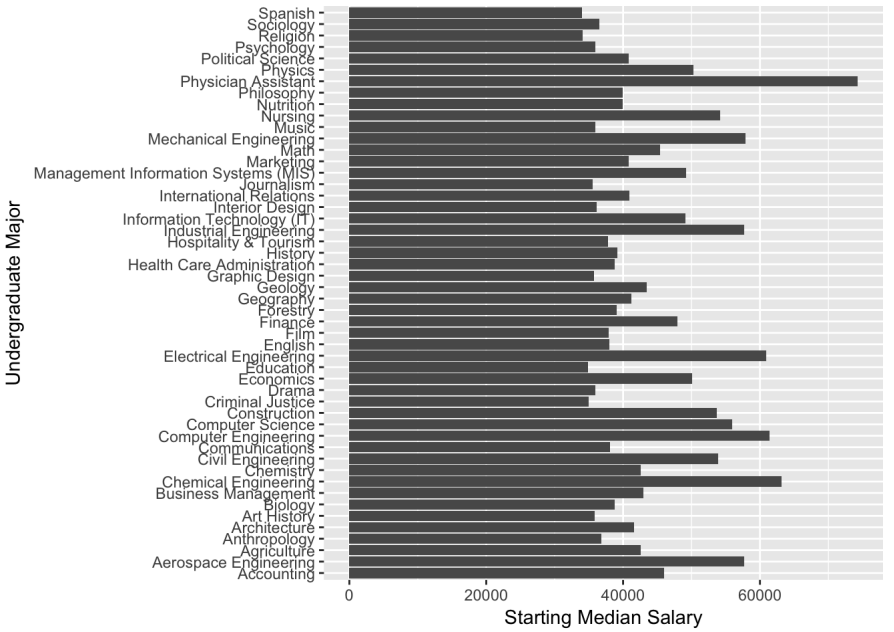
```
gg1 = gg + geom_bar(stat = 'identity')
gg1
```



Ok so now we actually have something to look at here. Success. However, our x axis item titles are a big jumbled mess, so lets try to fix that up. Lets make this vertical bar plot into a horizontal plot so there's some space for the undergrad major names.

Here we will use the coord_flip function to switch the x and y axis.

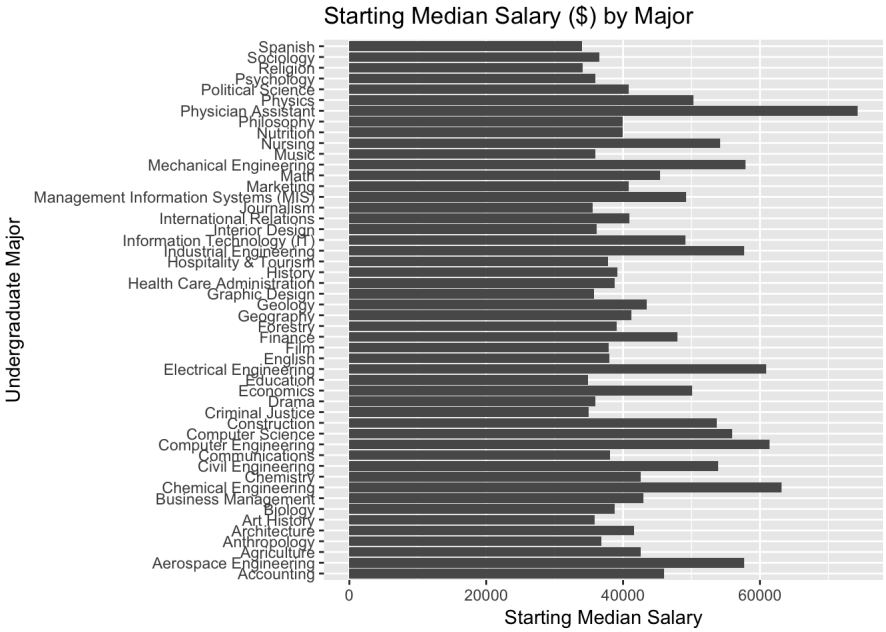
```
gg2 = gg1 + coord_flip()
gg2
```



Looking pretty good so far. Now lets slap a title on this guy and I'd say were good to go.

Lets use the ggtitle function to do this

```
gg3 = gg2 + ggtitle('Starting Median Salary ($) by Major')
gg3
```



All done. From this example we see how simple it is to manipulate different graphic parameters in ggplot by starting with a ggplot object and adding several 'layers' of functions. In this example we only used a few of the layers ggplot has to offer, but there are many many more out there.

Conclusion

From my explanation of qplot and ggplot within the ggplot2 package I hope to make 2 things clear about the ggplot2 package.

1. ggplot2 is kind

ggplot2 makes data visualization simple and easy. In qplot, we can easily and quickly construct graphs in a single line of code by providing a number of graphic arguments. In ggplot, we can manipulate a large number of graphical parameters with the addition of geom layers and functions.

2. ggplot2 is beautiful

By using ggplot2 we can easily attribute a number of aesthetic qualities to our data visualizations to make our plots both simple and pleasing to the eye. In a world of data science where visually communicating statistical information is key, the ability to make data pleasing to the eye is not only preferable but essential.

I hope from reading this post you have gained a substantial amount of insight into the mechanisms and applications of ggplot2.

References

1. 'http://www.r-graph-gallery.com/portfolio/ggplot2-package/'
2. 'https://www.aridhia.com/technical-tutorials/the-fundamentals-of-ggplot-explained/'
3. 'http://ramnathv.github.io/swc-nw-dataviz/visualize/ggplot2.html'
4. 'http://ggplot2.org/book/appendices.pdf'
5. 'http://ggplot2.org/resources/2007-past-present-future.pdf'
6. 'https://www.statmethods.net/advgraphs/ggplot2.html'
7. 'http://moderngraphics11.pbworks.com/f/ggplot2-Book09hWickham.pdf'
8. 'http://www.rmanchester.org/presentations/2013/05/ManchesterR_-_Introduction_to_ggplot2_-_Dennis_Prangle_-_20130502.pdf'

Data set:

9. 'https://www.kaggle.com/wsj/college-salaries'

Info on Hadley Wickham:

10. 'http://hadley.nz/'