Discovering what data visualization is and some tools one can use for it

Avanti Mehrotra

10/26/2017

Introduction - Why I chose to research **Data Visualization**

For the last few years, I've heard the term "data visualization" get used a lot, but it simply sounded like fancy and technical terminology to me. Through our class, I've realized that data visualization is actually about creating visuals to represent sets of data. I want to conceptualize this idea more and look at data visualization for data outside of our class data. I also want to delve more into why this topic is important and what other tools people can use to visualize their data, outside of the tools we use in class. I will present my findings in this post.

What exactly *is* **Data Visualization** and why is it important?

In short, data visualization is a way to visualize large sets of data. After taking a large set of data and using tools to visualize the data in the form of graphs, you can make adjustments to your visuals to see how certain changes or effects can alter the patterns shown in your data visualization. Data visualization models can also help you make predictions about how data points you don't have may behave. Moreover, if you visualize your data set by creating graphs, you'll have a much easier time analyzing the data rather than if it were left in a spreadsheet format. (i)

How to visualize data sets based on what we've learned in class

In class, we learned how to do data visualization in R. One such way is to use the package **ggplot2**. While one can use the base graphics, which are simply graphics already installed in R, these are often not aesthetically-pleasing and they require a good amount of work. Some base graphic examples include plot(), barplot(), and hist(). What's nice about R and ggplot2 is that you can quite easily create complex graphs and format them to look nice. For example, take a sample data set (US_births_2000-2014_SSA), from which we want to create a plot of the total number of births each day in January 2000. Right now, the data set gives the number of births per day in the US from 2000-2014. (ii)

In order to read the csv file and use ggplot2, we must first load the packages **readr** and **ggplot2**.

```
# load readr and ggplot2
library(readr)
library(ggplot2)
```

Now, let's actually read the csv file and save it as a data frame:

```
# read csv file using read_csv
us_births_2000to2014 <- read_csv("../data/US_births_2000-2014_SSA.csv", col_types = list("year" = col_integer(), "
month" = col_integer(), "date_of_month" = col_integer(), "day_of_week" = col_integer(), "births" = col_integer()))
us_births_2000to2014
```

```
## # A tibble: 5,479 x 5
##     year month date_of_month day_of_week births
##    <int> <int>         <int>       <int>  <int>
##  1  2000     1             1           6   9083
##  2  2000     1             2           7   8006
##  3  2000     1             3           1  11363
##  4  2000     1             4           2  13032
##  5  2000     1             5           3  12558
##  6  2000     1             6           4  12466
##  7  2000     1             7           5  12516
##  8  2000     1             8           6   8934
##  9  2000     1             9           7   7949
## 10  2000     1            10           1  11668
## # ... with 5,469 more rows
```

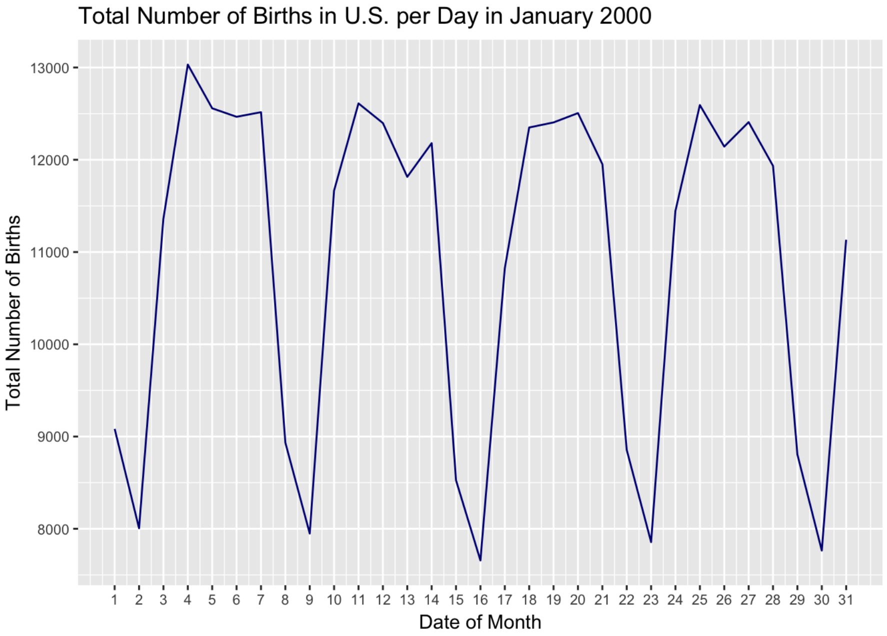
Let's shorten the CSV file so it only includes data for the year 2000, and let's also name the new data frame:

```
# shorten us_births_2000to2014 to only include data for January 2000
us_births_jan_2000 <- us_births_2000to2014[us_births_2000to2014$year == 2000 & us_births_2000to2014$month == 1, ]
us_births_jan_2000
```

```
## # A tibble: 31 x 5
##     year month date_of_month day_of_week births
##    <int> <int>         <int>       <int>  <int>
##  1  2000     1             1           6   9083
##  2  2000     1             2           7   8006
##  3  2000     1             3           1  11363
##  4  2000     1             4           2  13032
##  5  2000     1             5           3  12558
##  6  2000     1             6           4  12466
##  7  2000     1             7           5  12516
##  8  2000     1             8           6   8934
##  9  2000     1             9           7   7949
## 10  2000     1            10           1  11668
## # ... with 21 more rows
```

Next, let’s use ggplot2 to actually graph the data:

```
# plot data of number of births each day in January 2000
ggplot(us_births_jan_2000, aes(date_of_month, births)) + geom_line(color = "Navy") + labs(x = "Date of Month", y =
"Total Number of Births") + ggtitle("Total Number of Births in U.S. per Day in January 2000") + theme(axis.text=el
ement_text(size=8)) +  scale_x_continuous(breaks = 1:31)
```



Description of Graph: This graph is a line plot depicting the total number of births each day in the US in January 2000.

Interpretation of Graph: By taking the data from January 2000 and using ggplot2 to visualize the data, we can analyze the data points more easily than if we had left them in a table. From our graph, we can see that roughly every seven days, the patern of births repeats itself. The highest number of births is, based on our graph, around the fourth day of every week, while the lowest number of births is every seven days starting from January 2nd. Looking up the calendar for January 2000 (iii), we can see that January 2nd is a Sunday, whereas the fourth day of every week is a Tuesday. In January 2000, then, we see that the lowest number of births each week occurred every Sunday, while the highest number of births each week happened around the middle of the week and generally on Tuesdays.

How to visualize data sets using tools outside of this class

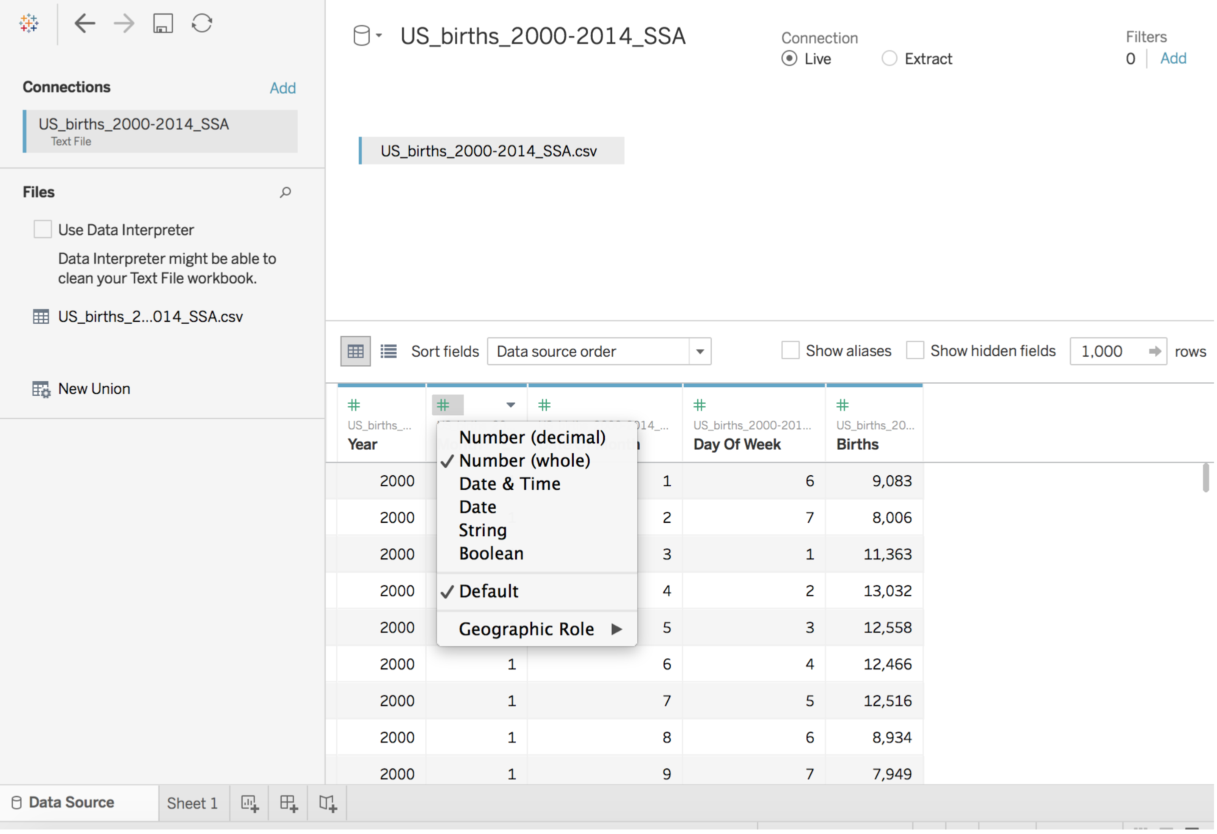
While we have looked at some of the ways to visualize data through R, there are other tools that are popular for data visualization. These include Tableau and its competitor Qlikview, as well as FusionCharts and Highcharts. One tool I want to particularly focus on is **Tableau**, a data analysis software that’s used for especially large sets of volatile data (iv).

The basics of how Tableau works

In order to understand how Tableau works, I downloaded the software. One thing I immediately noticed is that with this software, you can connect to a data set that’s stored as a PDF, Excel Spreadsheet, or Text file, to name a few. Or, you can connect to a data set stored on a server like MySQL.

I decided to upload the orginal data set we were modified above, which was stored in the CSV file US_births_2000-2014_SSA.csv. In order to manipulate our data in R, we’d have to write code, yet in Tableau we can make some of these manipulations directly with the interactive features in the software (v).

For example, we can simply double click on a column name to change it, and we can click on the hashtag above the column name to change the column type.



Connection ● Live ○ Extract

Filters 0 | Add

□ Use Data Interpreter

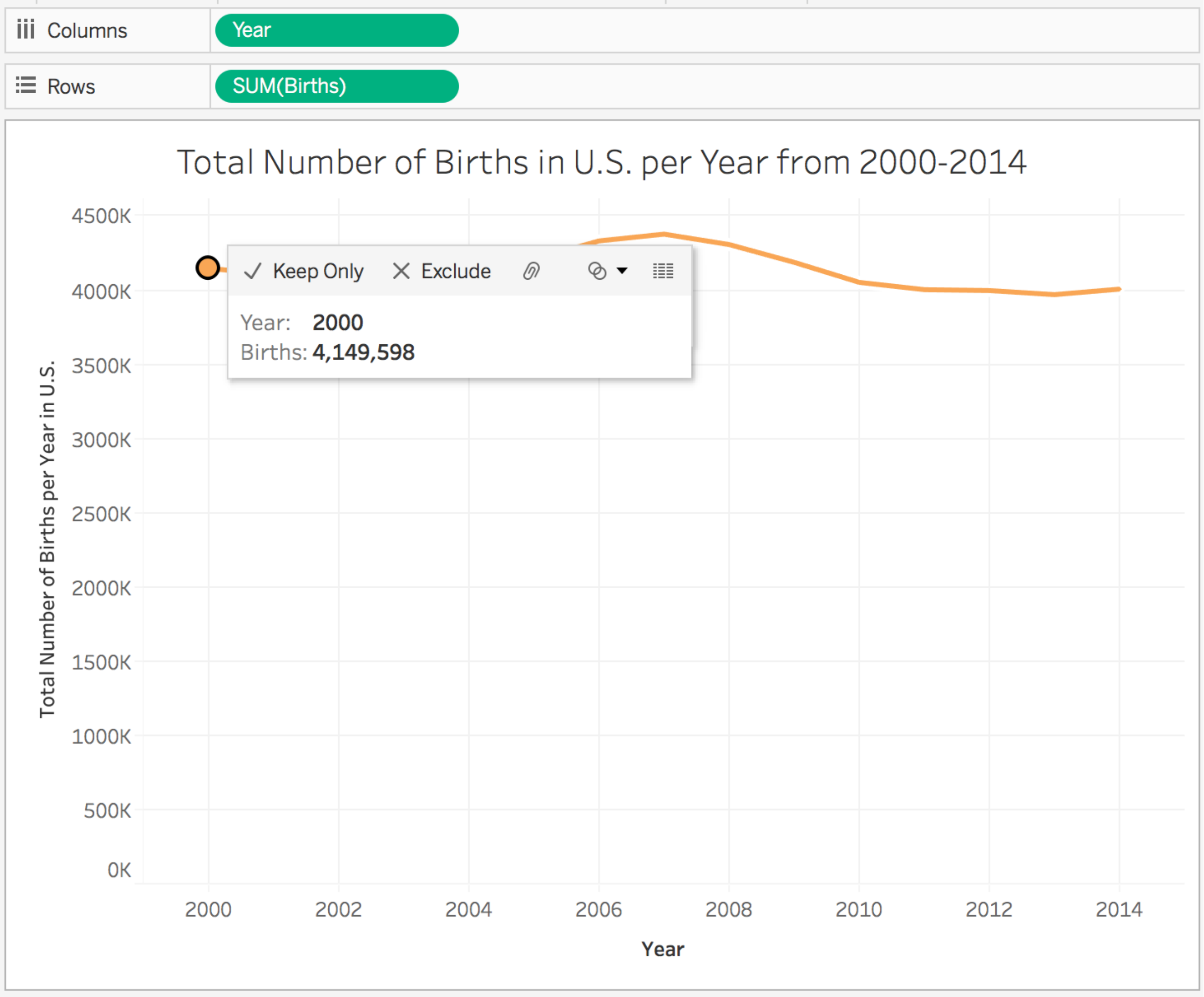
Sort fields: Data source order

□ Show aliases □ Show hidden fields 1,000 rows

US_births_... Year			US_births_2000-201... Day Of Week	US_births_20... Births
2000		1	6	9,083
2000		2	7	8,006
2000		3	1	11,363
2000		4	2	13,032
2000		5	3	12,558
2000	1	6	4	12,466
2000	1	7	5	12,516
2000	1	8	6	8,934
2000	1	9	7	7,949

In order to do visualize the data, one can click on **Sheet 1** in the bottom left hand corner and drag variables listed in **Dimesions** and **Measures** (both of which are on a panel on the left). All one has to do then is drag the chosen variables into the center where it says "Drop Field Here."

You can easily graph the data and scroll over your graph to get specifics about a certain data point. For example, in the below image, scrolling over the graph at the shown point tells us how many total births there were in the US in 2000.



Columns: Year

Rows: SUM(Births)

Year: 2000
Births: 4,149,598

Description of Graph: This graph is a line plot that shows the sum of the number of births in the U.S. per year, from 2000 to 2014.

Why is Tableau useful?

The nice thing about Tableau, as we can see from the above images, is that it's really easy to use. Creating the above graph took me less than five minutes, whereas making a line plot for only January 2000 using R's ggplot2 package took me close to 15 minutes (I had to make sure I was using the correct functions, syntax, and variables.) Tableau is also quite easy to use for people who need to analyze data but don't necessarily have the expertise to code up similar graphs (vi). Moreover, Tableau can handle large sets of data — for example, US_births_2000-2014_SSA.csv has over 5,400 entries — and it's also a very interactive software.

Tableau, however, as expalined in source (vii), does not have some of the helpful algorithms that R has. One way to alleviate this problem is to download the R package **RServe**, which allows you to use R in Tableau (vii). There are also other cons to Tableau. For example, it's expensive software and doesn't allow you to actually create data tables (viii).

Conclusion/Takeaways from this post

While the visuals we can create with R are pretty cool, especially those we've learned about in class so far, there are other tools and softwares we can use. Tableau is one software — which according to a course on Coursera, is the "most popopular visualization program in the business world" (ix) — that can create similar, or even more extensive, graphs through it's interactive features.

I really enjoyed writing this post because I got to learn about terms I've heard used a lot but that I never really understood what they meant. While intuitively the terms "data visualization" mean exactly what one would think they'd mean — making visuals of data sets — I now know exactly what it means, and I can also provide examples of different types of data visualization/ways to do data visualization. In class, we've learned some forms of data visualization, which include using packages like **ggplot2**. Through this post, I learned about Tableau, a popular software I had heard about but didn't know what it was, and I also learned how to use some of Tableau's simple features.

Resources Used

- https://www.sas.com/en_us/insights/big-data/data-visualization.html (i)
- https://github.com/fivethirtyeight/data (ii)
- https://www.timeanddate.com/calendar/monthly.html?year=2000&month=1&country=1 (iii)
- https://www.forbes.com/sites/bernardmarr/2017/07/20/the-7-best-data-visualization-tools-in-2017/#147bdfea6c30 (iv)
- https://www.tableau.com/learn/tutorials/on-demand/getting-started-data?reg-delay=true (v)
- http://www.affecto.com/insights/blog/9-reasons-why-tableau-rocks/ (vi)
- http://www.simafore.com/blog/bid/120209/Integrating-Tableau-and-R-for-data-analytics-in-four-simple-steps (vii)
- https://www.sam-solutions.com/blog/tableau-software-review-pros-and-cons-of-a-bi-solution-for-data-visualization/ (viii)
- https://www.coursera.org/learn/analytics-tableau (ix)