Visualizing Email Analysis

Aryadip Sarkar | Visualization and Visual Analytics | 9/14

Motivation

The motivation of the project was to find some trends in the email that we get round the year and through which we can get some logical conclusion about some behavior of people or institutions or myself !

Collection of data

To keep it simple, I took a sample of around 1000 data-sets which comprises of the date, time , day and the inbox in which I am getting the email.
I started collecting the random data-sets from 3 of my personal inboxes:
1)mail.aryadip.sarkar
2)aryadip.sarkar
3)asarka7.uic

The fields of the data-sets are :

1) Type of the inbox
2) Day
3) Time
4) Date
5) Time in 12 hours format

The problem that I faced while collecting the data was , I couldn't get these information about the email from Gmail. Hence, I had to install Mozilla Thunderbird and from there I had to export the data in CSV format. The data that Thunderbird downloaded was raw one. Hence, I had to do a lot of clean-up of the data to bring it to a stable condition, in which I can work upon!

Trends & Conclusions

The whole motivation was doing this exercise was to see some prominent trend by which we can come some logical conclusion. I got some interesting conclusion which I would like to share:

1) On Sunday, I get less number of emails compared to other weekdays and that's quite logical considering the fact that most of the offices/institutions are closed on that day
2) Monday , Tuesday and Wednesday seems to be favorite for sending emails as the number of dots are more on those days
3) From 12Am to 3 /4 Am the density of the emails are quite less, as most of the people will be sleeping during that time
4) Heavy amount of traffic is noticed between 5am to 5pm on weekdays, as that is when most of the of the institutions /offices are open and they trigger the emails

5) From 6pm to 10pm, the traffic gets relatively less that what is compared to the peak time, as most of the offices /institutions close by 5pm /6pm and no one is there to trigger the email or reply to the emails
6) On Sunday, even if I am getting some emails around 6 am to 12 pm , but during the afternoon and evening, it's a strict no !
7) mail.aryadip.sarkar is my official email id where most of the companies reply and it is represented by blue dot. As we can see, the blue dots are more in the weekdays rather than over the weekends as companies tend to be closed over the weekends
8) Over the weekends, most of the dots are green or orange. Green represents my school's official id and Orange represent my general google id. These are constantly active as we may get auto-generated emails into the accounts as they are linked with various advertisement and social network sites!

Decision of choosing the visualization

I wanted to find the gaps in between the hours when plotted against days, so that it can be clearly visible at what point of time I am getting more mails or what time I am not! The conclusions that I have provided above, is an indication that I have chosen a good visualization which actually depicted something worthwhile and I could get some logical conclusion out of it.

The Zipped folder:

The zipped folder contains :
1)Code – which contains the Code for the visualization
2)Images – which has the trial images and the final one
3)Project Synopsis

The "Code" folder contains :

1) bootstrap folder
2) email.csv
3) index.html
4) script.js
5) style.css
6) visualize.html

No need to open the Bootstrap folder.
"index.js" is the landing page. Once you open the landing page, click on the "*Click to Visualize*" button and it will direct you to the visualization page.

The D3 code is there in the "script.js" .
The corresponding CSS code is there in the "style.css"
The HTML code is there in "Visualize.html"
The Data is there in "email.csv"

Reference:

I took some help from Mike's bl.ocks. http://bl.ocks.org/weiglemc/6185069

But here the scatterplot has numbers on both the axes which makes is way easier to plot. I had Day on one axis and Time on the another. To find the Range and Domain of days and time on both sides, has been real challenge. Most of the examples calculated the range /domain from the data that is available (e.g., if the data's range is within Monday to Wednesday, then the axis will have only those days!) But here, I had to break the y-axis into 7 different points and the X-axis in 24 points and the plot the points accordingly. Another challenge was to parse the time format and then plot it on the graph. And the reparse, it so that the viewer can see that in a n understandable format.

If you go through the codebase, you will know that except the legend portion , I had to challenge myself to change almost everything possible to customize it according to my need.