Email thread visualization

Yue Liu	yl3632@nyu.edu	yl3632
Yun Gao	yg1200@nyu.edu	yg1200
Xia Zhao	xz1508@nyu.edu	xz1508
Yining Shan	ys2483@nyu.edu	ys2483

Project page (on Github):
https://github.com/NYU-CS6313-Fall16/Reddit-Threads-18
Video link:
https://youtu.be/MFkdispIXSM
Demo link:
https://clementxia.github.io/InfoProject/

What is the problem you want to solve and who has this problem?

The purpose of our project is to help people understand the activity of email thread in a easier way because most of the unprofessional people are not very familiar with this concept. We want to extract some hidden email thread information in our given raw dataset through our visualization project. Then, people can easily figure out the information about email thread, like the number of threads with 2 people involved. The result of our project can also represent some interesting point about people's habit using email.

What are the driving analytical questions you want to be able to answer with your visualization?
An email thread includes a running list of all the succeeding replies starting with the original email. The replies are arranged visually near the original message, usually from the first reply to the most recent.

1.What is the number of people that most email threads involve? What is the number of threads with only 2 people involved?
→ People may want to know the number of threads with a certain number of people involved. In order to achieve this goal, we need to figure out how the number of threads distributes across a specific number of people.

2.What is the number of emails that most email threads have? What is the number of threads that include 3 emails?
→ People may want to know the number of threads having a certain number of emails. If we figure out how the number of threads distributes across a specific number of people, we will find some interesting information.

3.How long do most email threads last? What is the number of threads that last for 2 days?

→ People may want to know the number of threads that last for a certain number of days. In order to achieve this goal, we need to figure out how the number of threads distribute across a specific time duration. A bar chart could be a good way to represent this information.

4.What is the time that most threads usually update at?
→ This could show people's habit of sending emails. If we know other people's habit of using email, we may send an email at the time that may get a reply in short time.The date attribute in our dataset can be used to answer this question.

What does your data look like? Where does it come from? What real-world phenomena does it capture?
Our dataset is given by our tutor with over 150,000 records. This dataset contains information for email threads over 10 years. We choose the data in February 2016 which contains over 3500 records to do our project. We believe one month's data is enough for us to find some interesting information.

The dataset contains 5 attributes including thread ID, thread index, from-address, to-address and date information. The thread ID is an integer with about 18 digits. The thread index does not contain useful information so we removed this part when processing data. The domain names of from-address and to-address are hidden by long string mixed with various strange numbers and letters.

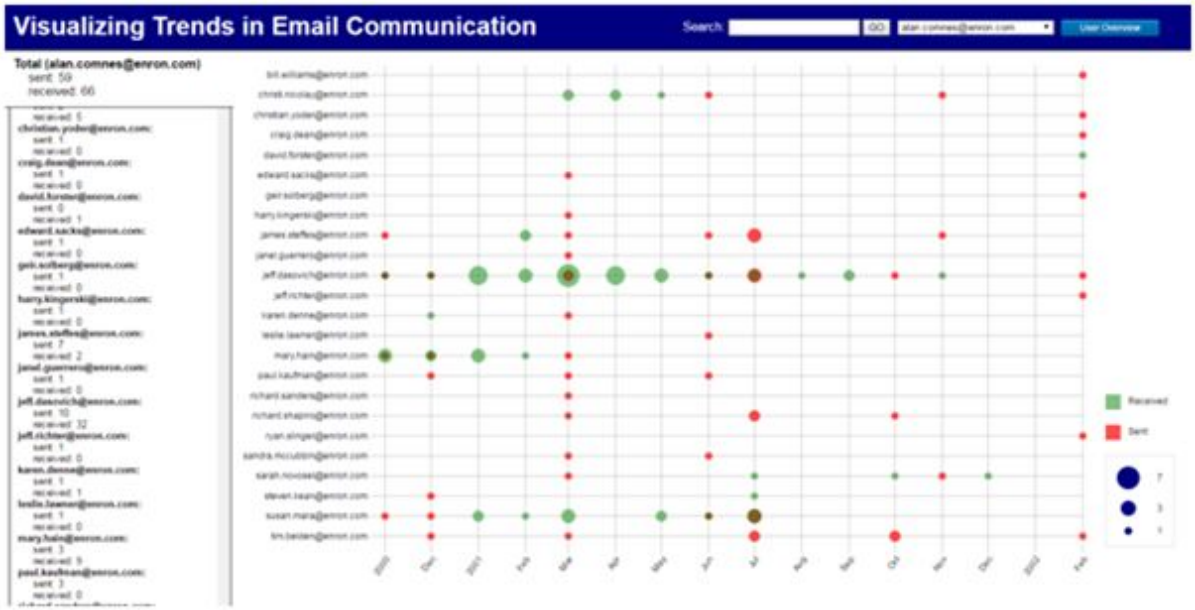
Attribute name	Attribute type	Description	Value range/ Categories	Derived ?
id	categorical	unique id of a thread		N
duration	categorical	number of days a thread last; last day - first day of a thread's updating dates	0 to 25 days	Y
people	categorical	number of people involving in a thread; count the number of distinct email addresses for each thread	1 to 20	Y
email	categorical	number of emails a thread have; count the number of repeating times for each thread id in the datasets	1 to 25	Y
#threads for a specific thread duration	quantitative	#threads for a specific thread duration	0 to 1833	Y
#threads for a specific #people	quantitative	#threads for a specific #people	0 to 1745	Y
#threads for a	quantitative	#threads for a specific #email	0 to 1655	Y

specific #email				
date	ordinal	time when the thread is updated	Feb. 2016	N

We use tools like MapReduce and EXCEL. It is necessary for us to distinguish unique thread ID and calculate the number of emails, the number of people involved and also the duration of each thread in raw data. Before D3 programming, we build a new csv file with processed filtered data.

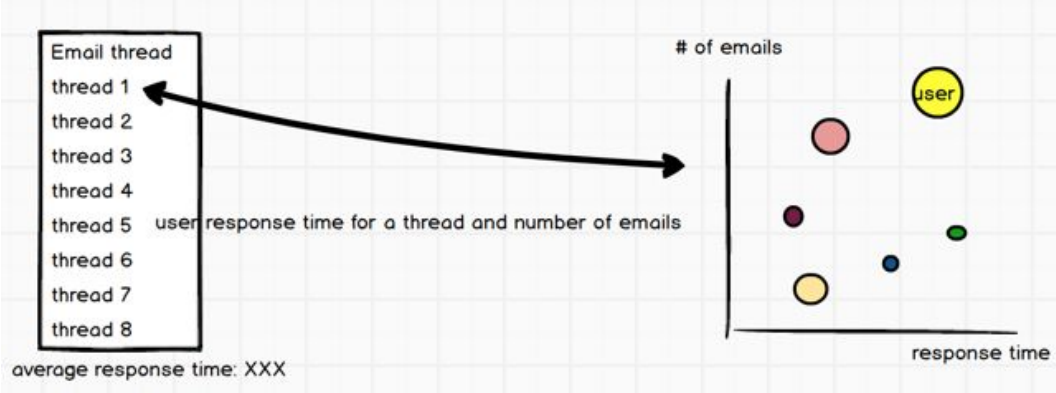
What have others done to solve this or related problems?

http://nyu-cs6313-fall2015.github.io/Group-4/

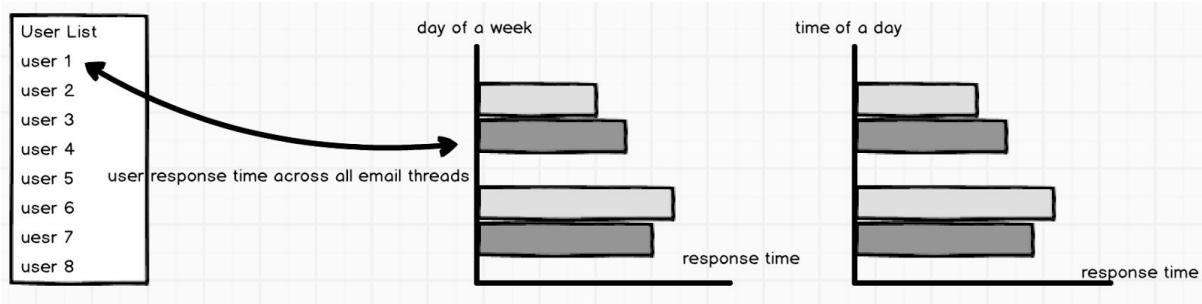


The goal of this project is to visualize given user email address activities over time for an investigator.
This application provides investigators with a timeline of sorts, mapping time (in months) to the x-axis and users (email addresses) to the y-axis. The timeline/chart displayed is always given in the context of a single selected user. The other users on the y-axis are the users that they communicated. On the chart, using area to encode email volume, red nodes indicate emails sent by the selected user and green nodes indicate emails that the selected user receive.
It focuses on the visualizing the email user activities in different time. This is related to our project because the purpose of our project is to visualize email threads which also considers when an email was sent.
Design Iterations

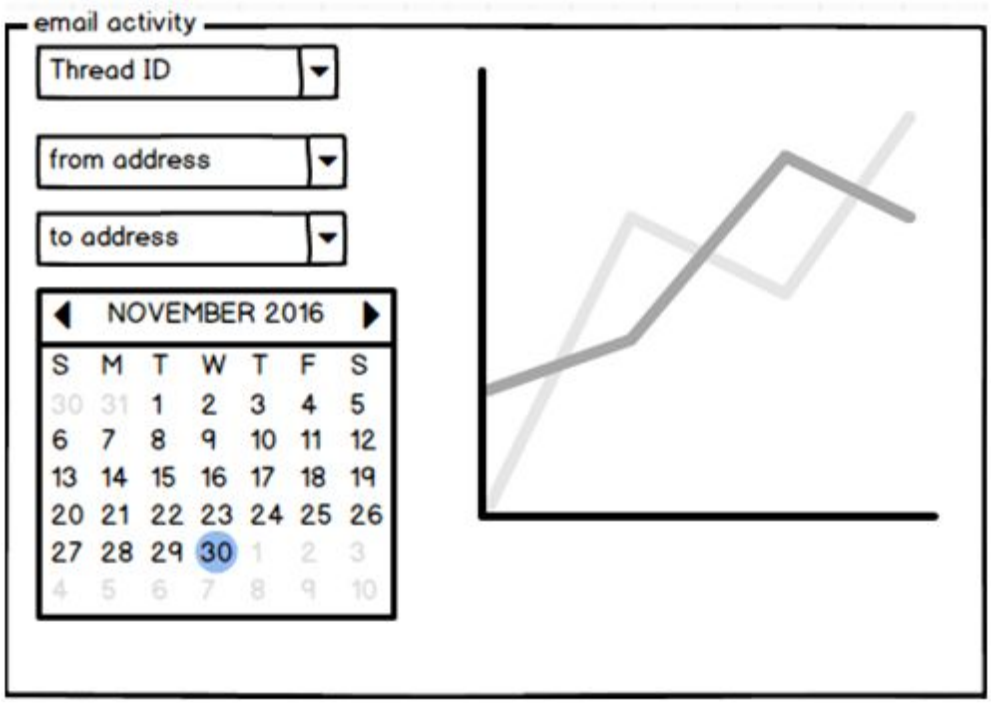
First mockup:
Initially, we want to know the factors that influences response time and the evaluation of the thread id.



So in this mock, the first graph is about the number of user distribution of the response time. To be more precise, the size of the circle means the amount of the users of the specific thread. The x-axis means response time and y-axis is the amount of the emails.



In the second graph, we want to know the response time of the specific user during the week. The response time is visualized by the x-axis and the day of the week is visualized by y-axis. When the user click the user name on the left bar, it will show the response time of this user on the day of the week.

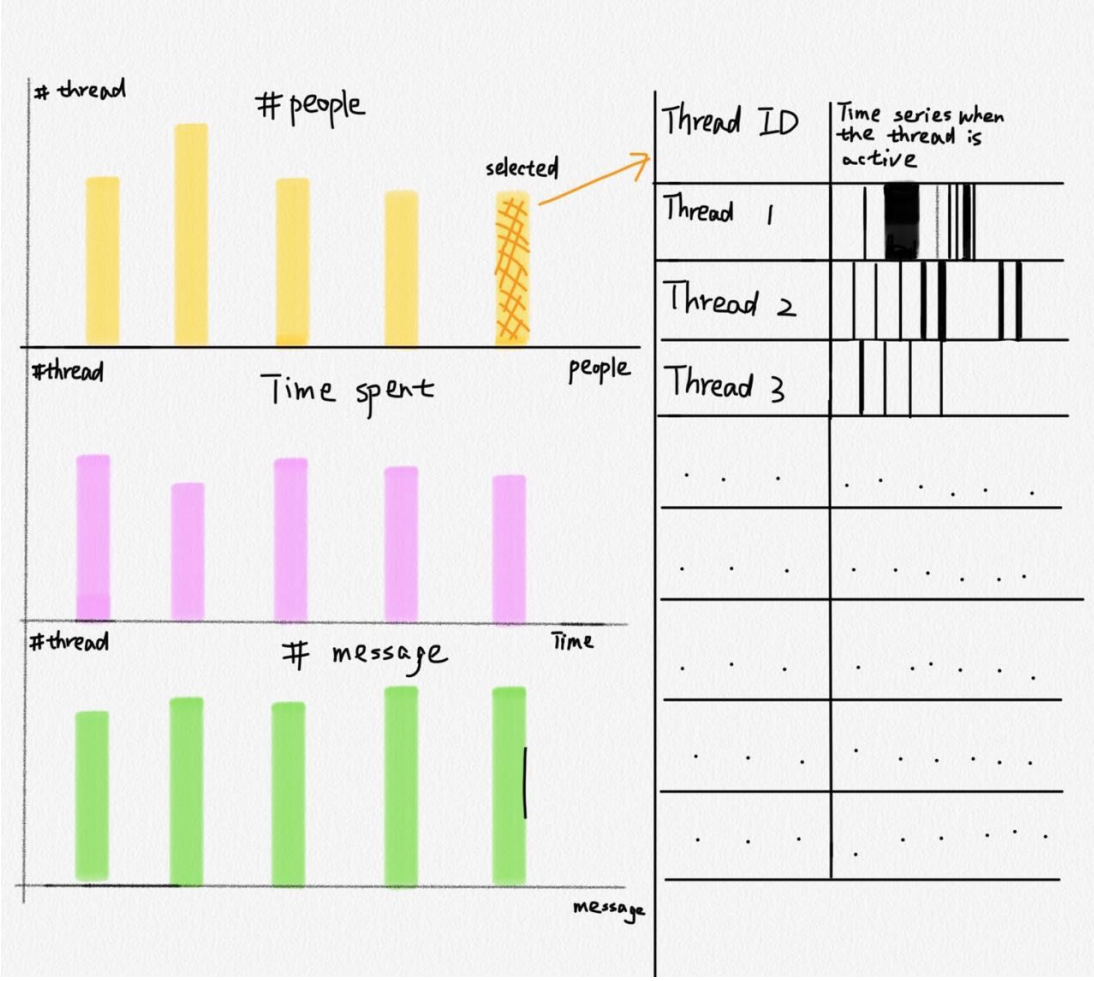


In the last graph, we want to visualize the specific thread by selecting the thread id and the date. Also the user can select the thread by the from and to email address. If the thread has been selected, it will show the amount of email distribution of the time.

Second mockup:

In the first mockup, we focus on both thread and response time. Although the raw dataset contains both information about thread and response time, it is a little mess to present these data in our first mockup. What is more, it is very hard for us to process the dataset since the response time and thread are totally different directions.

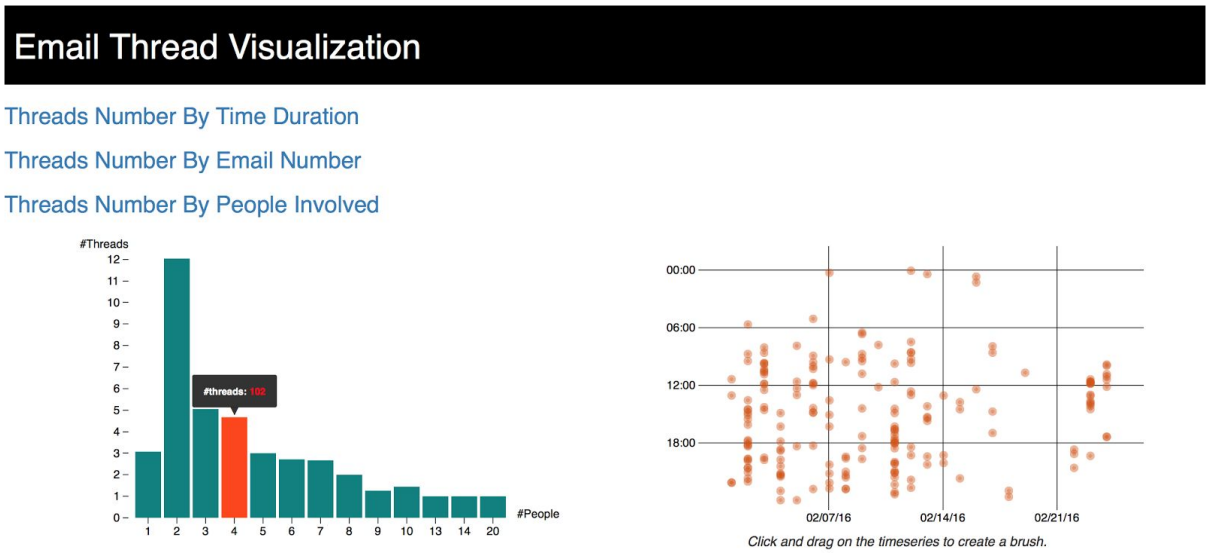
After discussion with our supervisor, we decide to drop the response time and focus on the email thread. The following graph is our second mockup.



In this graph, we just focus on the thread of the email. There are three bar charts and one time series chart in our page. The top bar chart shows the number of threads for the specific number of people involved. The second bar chart is similar to the top bar chart. It shows the the number of threads for the number of days the thread last. The last shows the number of thread distribution on the number of message included in the thread.

If the user click the bar, it will show the time series of the emails in these threads.

Final Visualization



Note: #Thread is the cube root of original numbers because of the display reason.

This is our final visualization, it is derived from the second mock up. The durations with no email thread are not shown in the bar chart. Besides, we slightly modify the time series graph of the original mock up. Now the graph can not only show the time of the email but as the date as well. Every small dots in the right graph represent a email. Some dots seems darker because of overlapping. It is more clear to show the email time distribution than the second mock up.

How to read it

There are two charts above. The left chart shows how the number of threads distributes across a different number of people. The y axis refers to the number of threads and the x axis refers to the duration. The right chart shows the emails' update time. The y axis is the time in a day and the x axis is the date. Each dot in right graph represents one email.

A user can select a bar in the left bar chart, then the right chart will update according to the bar selected. For example, if a user clicks the first bar, the right chart shows emails' update time for all the thread that lasts no more than one day.

A user can also view other distribution bar charts. For example, a user can choose to see how the number of threads distributes across a specific number of people by clicking "Threads Number By People Involved".

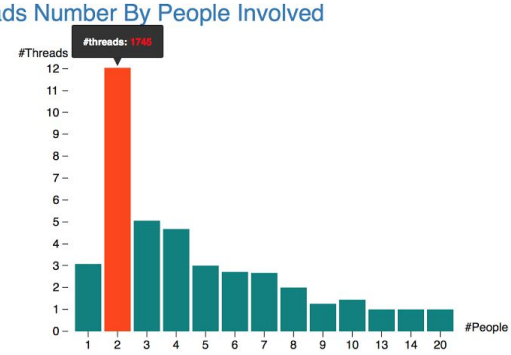
Findings

1.What is the number of people that most email threads involve in? What is the number of threads with only 2 people involved?

Threads Number By Time Duration

Threads Number By Email Number

Threads Number By People Involved

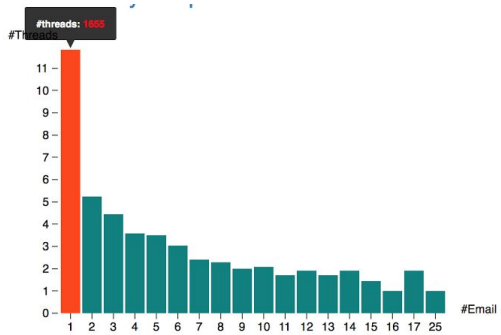


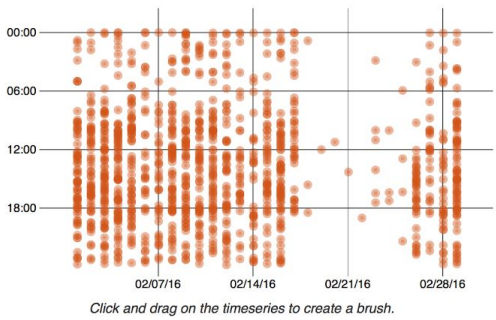
Note: #Thread is the cube root of original numbers because of the display reason.

From the bar chart above, we can see the number of threads with only 2 people involved is 1745 far more than other bars. .we can easily find out the bar chart is not uniformly distributed. The number of threads that involves in 2 people takes a big part of the total number of threads.

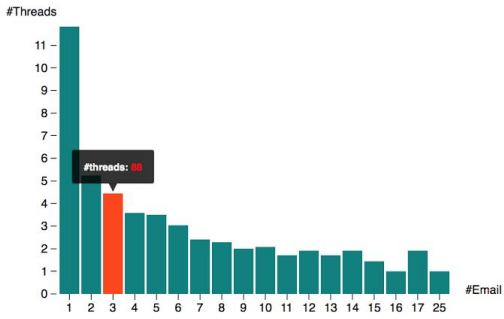
An interesting point is that only one person involved in a number of threads. At first, we expect this to be a rare phenomenon, but it still occurs more frequently than some other situation which involves more people. Probably because some people often send emails to themselves. This situation may happen when someone wants to attach files to store it in the mailbox. Maybe this happens to IT programmer when he needs to transform file, like cloud storage works, between a local host and virtual machine on the computer.

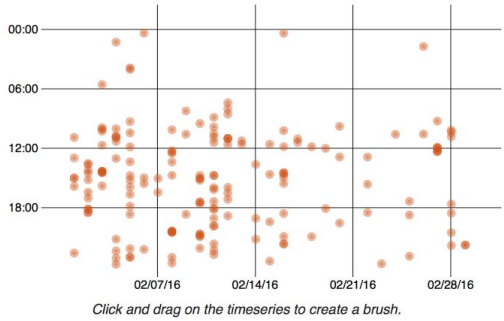
2.What is the number of emails that most email threads have? What is the number of threads that include 3 emails?





Note: #Thread is the cube root of original numbers because of the display reason.

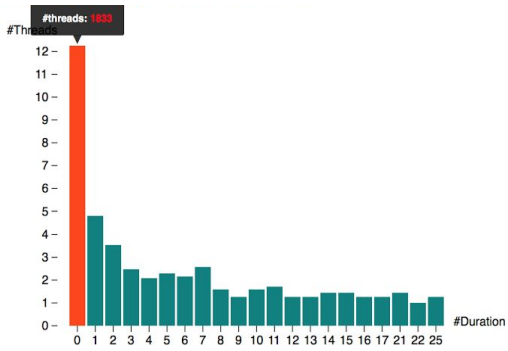


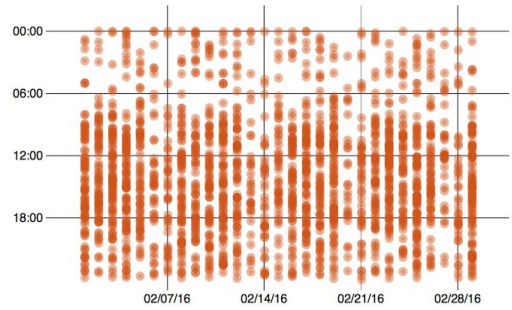


Note: #Thread is the cube root of original numbers because of the display reason.

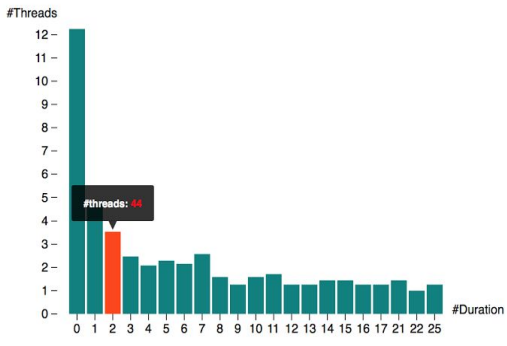
From the bar chart above, we can see that with 1655 threads, which take up most of the whole dataset, only contain 1 email. However, only 88 threads include 3 emails. The difference between them is really huge which can be seen obviously from the graph above. Even the total sum value of other circumstances are far from matched to this case.
I think theses email threads only contain one conversation which means most emails are not replied or do not need to be responded. Maybe because user forgets to reply. Also, it is possible that a large amount of junk mail overwhelmed to people's mailbox and ignored by the user.

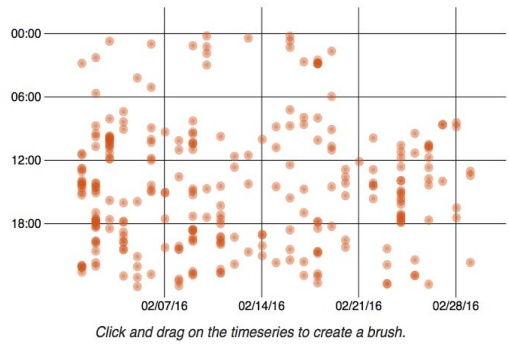
3.How long do most email threads last? What is the number of threads that last for 2 days?





Note: #Thread is the cube root of original numbers because of the display reason.





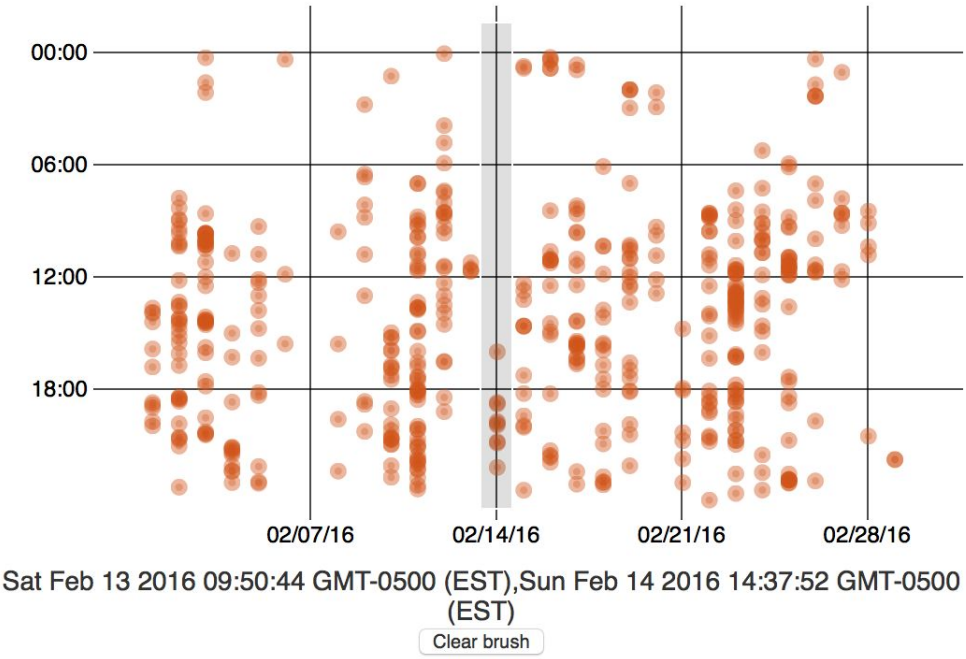
Note: #Thread is the cube root of original numbers because of the display reason.

1833 threads' lifespan is within one day which takes up a large part of the total threads which means the email conversation is finished in one day. The number of threads that last for 2 days is 111 which is a great decrease compared to 1833 threads last within one day. We can conclude that most email conversation lasts no more than two days.

What surprise us is that a small part of threads is really active which lasts over 10 days, even longer than 20 days.

4.What is the time that most threads usually update at?

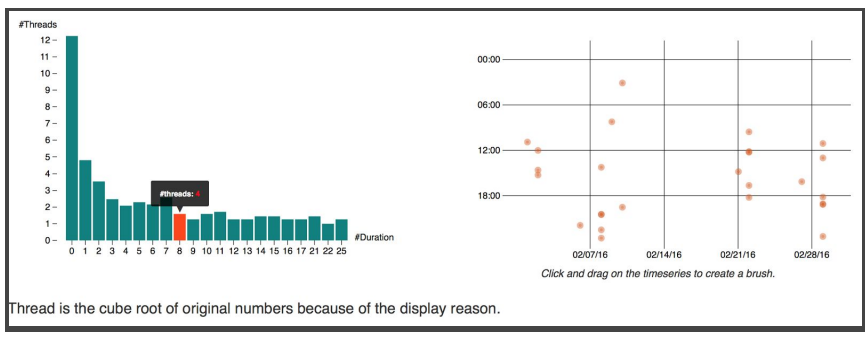
From the right charts of those graphs on last page, we can see that the dots appears densely between about 8:00 to 20:00 on vertical axis. On the other hand, the amount of dot is obviously less in night and early morning compared to other time periods. So, we can conclude that the email communication usually happens on day time.

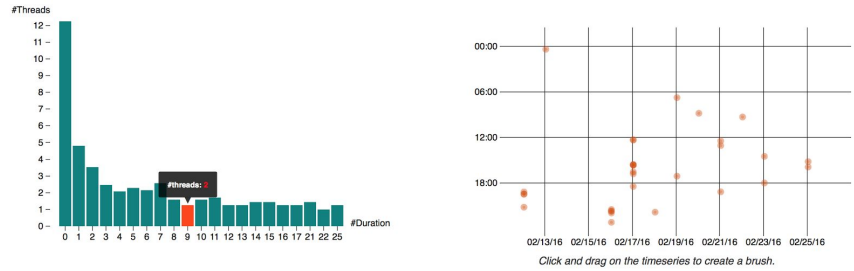


Also, we notice that the black vertical line which intersects with the horizontal axis indicated the weekend day after we create a time brush by dragging on the time series showing on the graph above. The dot around weekend is less dense which means the email contact activity is decreased.
According to the two foundings in last two paragraphs, we conclude that most threads usually updates at weekday and working time. The amount of thread is reduced maybe because people need to relax and have rest after busy working.

Some other interesting findings:

1.

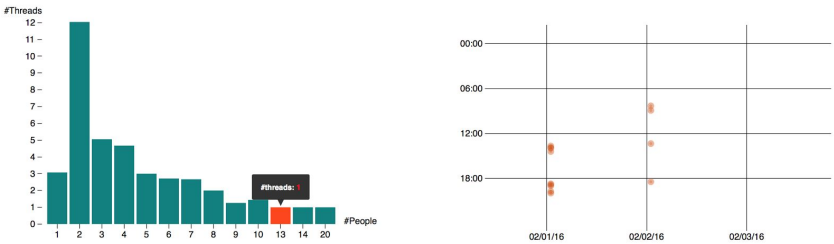


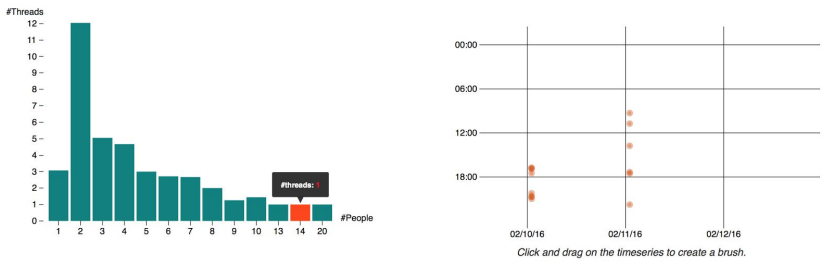


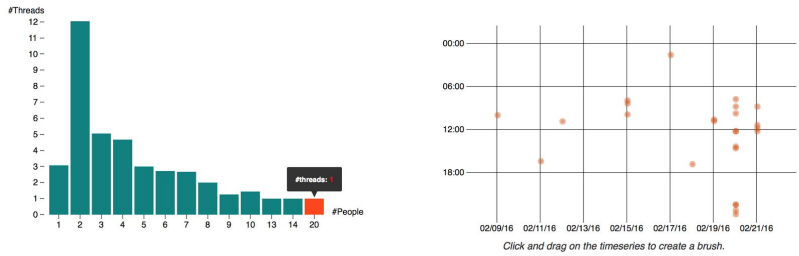
Thread is the cube root of original numbers because of the display reason.

Even though most threads updates at weekday, some threads seem to update around weekends when thread duration is long. I make some guesses. Maybe this is because when threads last long, this means the email are not in urgent need to be responded, such as emails among friends. This kind of email can usually be sent during weekends.

2.







From three graphs above, we can see that people usually send email on weekends periodically when people involved are more than 10 in a thread. And these kind of threads usually last more than a week and emails under these threads are not be responded or sent very often.

Limitation and future work

The limitation of our project is that we greatly reduced the scale of large dataset. We only processed the dataset with a month's record but the dataset contain records with over ten years. Even a year's dataset is overwhelmed for our project.
One problem is that some threads are intersected over two months which means the thread start at the end of previous month but end at the following month. So, a small fraction of results may not 100% accurate but does not influence much.

In the future, we will try to make the dot in right chart show date and time when the mouse moves in a dot. And try to add more functionality to our application.