Niklas Kolster
niklas.kolster@gmail.com

Analyzing location data from inside a building

Did two users meet in the building?

The data for this analysis is provided by Teralytics and can be obtained here:
https://drive.google.com/file/d/0B3FWxDUx-_qYSGl0OWkwaE91SDQ/view?usp=sharing

1. Introduction, the data and exploratory analysis

The task is to create a program that checks if two users have met in a building.
We have a dataset in csv-format that is ca. 135MB in size.

For the task we use R-programming language in RStudio IDEfor this assigment, also added a scala wrapper for the program.

We assume the data can be found in our current working directory.
We read in the data and take a look at it.

```
> data <- read.table("reduced.csv", header=T, stringsAsFactors=F, strip.white = T, sep=",")
> head(data)
                 timestamp         x          y floor      uid
1 2014-07-19T16:00:06.071Z 103.7921  71.50419     1 600dfbe2
2 2014-07-19T16:00:06.074Z 110.3361 100.68284     1 5e7b40e1
3 2014-07-19T16:00:06.076Z 110.0663  86.48874     1 285d22e4
4 2014-07-19T16:00:06.076Z 103.7850  71.45633     1 74d917a1
5 2014-07-19T16:00:06.076Z 109.0949  92.82449     1 3c3649fb
6 2014-07-19T16:00:06.563Z 105.9810  67.84456     1 d6f0300c

> nrow(data)
[1] 2228820
```

The data is quite self-explanatory and contains ca. 2.2M rows.

The co-ordinates have values roughly between 42 and 115, We have also know that the building is 100 to 200 metres, so for this excercise we assume the X and Y measuring unit is metres.

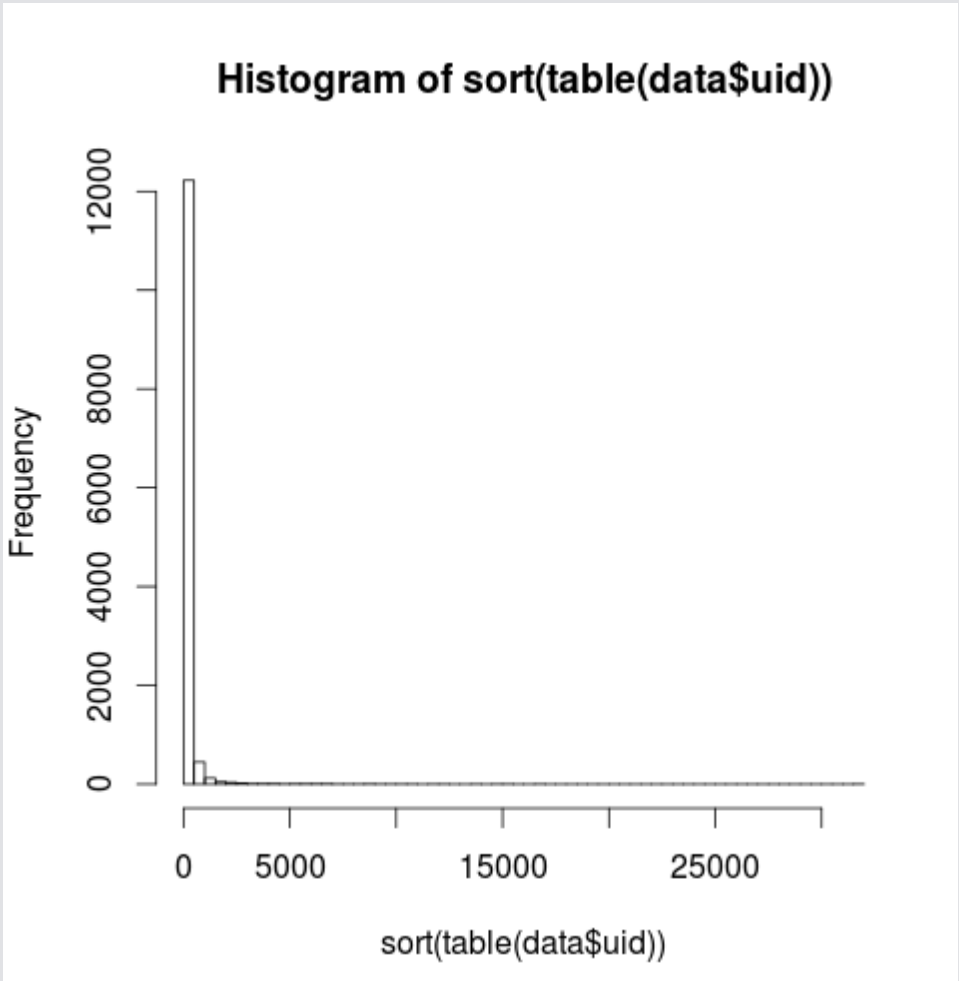
```
> summary(data$x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  43.70   81.23  104.20   94.08  108.00  115.10
> summary(data$y)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  42.49   66.22   71.15   75.10   86.43  102.80
```

We take a look at how many data-points we have per user:

```
> tail(sort(table(data$uid)))
51f21bd3 d05c03a0 17c725e9 22533c61 8eefdd01 f8c4ee12
   14640    15080    15430    15488    16522    31801
> head(sort(table(data$uid)))
00c787a9 01628a84 01772ca3 01b0a30d 035523b2 035ffe77
       1        1        1        1        1        1
```

For some users we have only 1 datapoint and for some we have over 10 000 data-points.

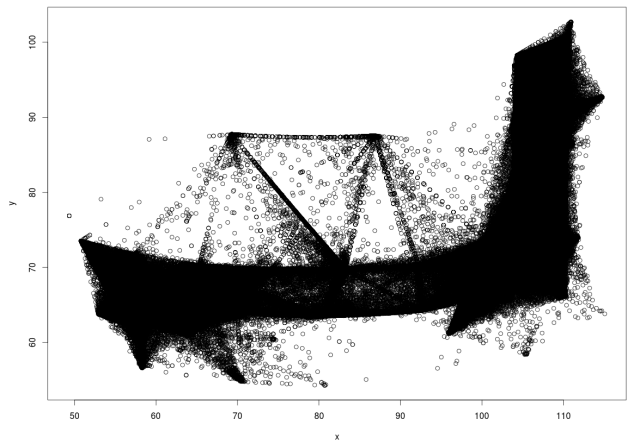
```
hist(sort(table(data$uid)), breaks=50
```



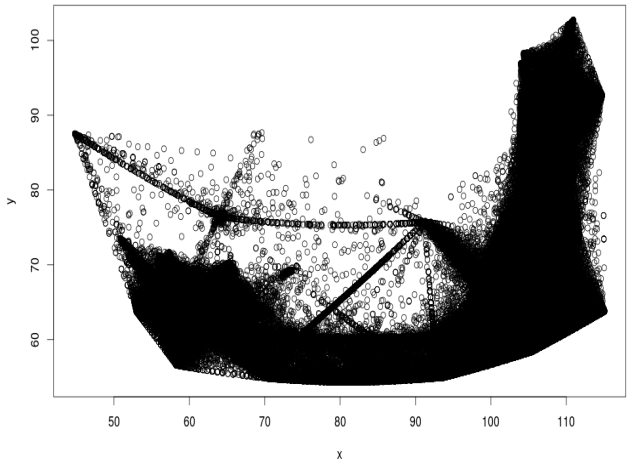
For majority of users we have relatively few data-points.

Looking at how the data-points are distributed in the building:

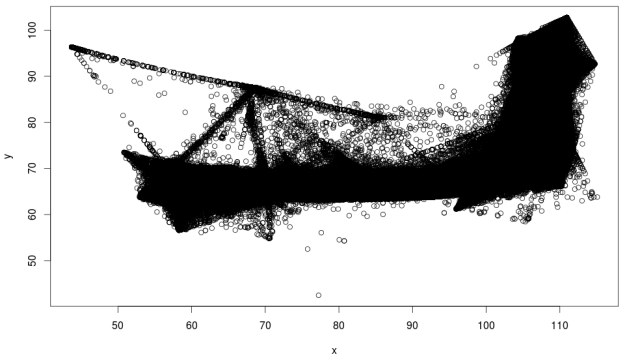
Floor 1:



Floor 2:



Floor 3:



Based on these one could start making guesses about the shape of the building etc. but without knowing more about how the data is gathered and how reliable the data-points are it is not good to engage in guessing. First floor could be a baseball stadium but guessing without more context is not really efficient use of time and might lead astray, so not drawing any conclusions here.

2. The objective and assumptions

Our objective is to find if 2 UID's have met.
We are not looking to find out how many times they have met or if they have been moving together etc. This means we only need to search if they have been close enough to each other at least once in both space and time with sufficient reliability.

To proceed we make some assumptions and set some requirements:

1. The X and Y measuring unit is metres
2. We look only at the known and last known positions of each person, we disregard blocking effects of possible walls because we do not know the walls locations.
3. Last known positions for the other party with more than 120 seconds between them are disregarded as one person can already move some distance in 2 minutes.
4. For a meeting to occur the following conditions have to be satisfied:
    a. The persons have been on the same floor
    b. The persons have been within 2 metres of each other
    c. There is not more than 120 seconds since the last known position of the other party.

We proceed and create an R script to analyze the potential meetings between two users. The maximum-distance and maximum acceptable age of timestamps for last known location are parameters that are easy to adjust in the script.

The script is saved as analyzeMeetings.R it requires the readr, xts and quantmod packages in addition to base R. In R they can easily be installed with the command: install.packages(c('readr', 'xts', 'quantmod'))

The program can be run in three ways, from R/RStudio, from command-prompt or using the scala-wrapper.

1. Running from R-console or RStudio

Load the function and read in the data, on R-command prompt or RStudio:

```
> source("analyzeMeetings.R")
```

Reading the csv-file takes 5-20seconds depending on the HDD/SSD of the computer, but once its in RAM the program is fast to run.

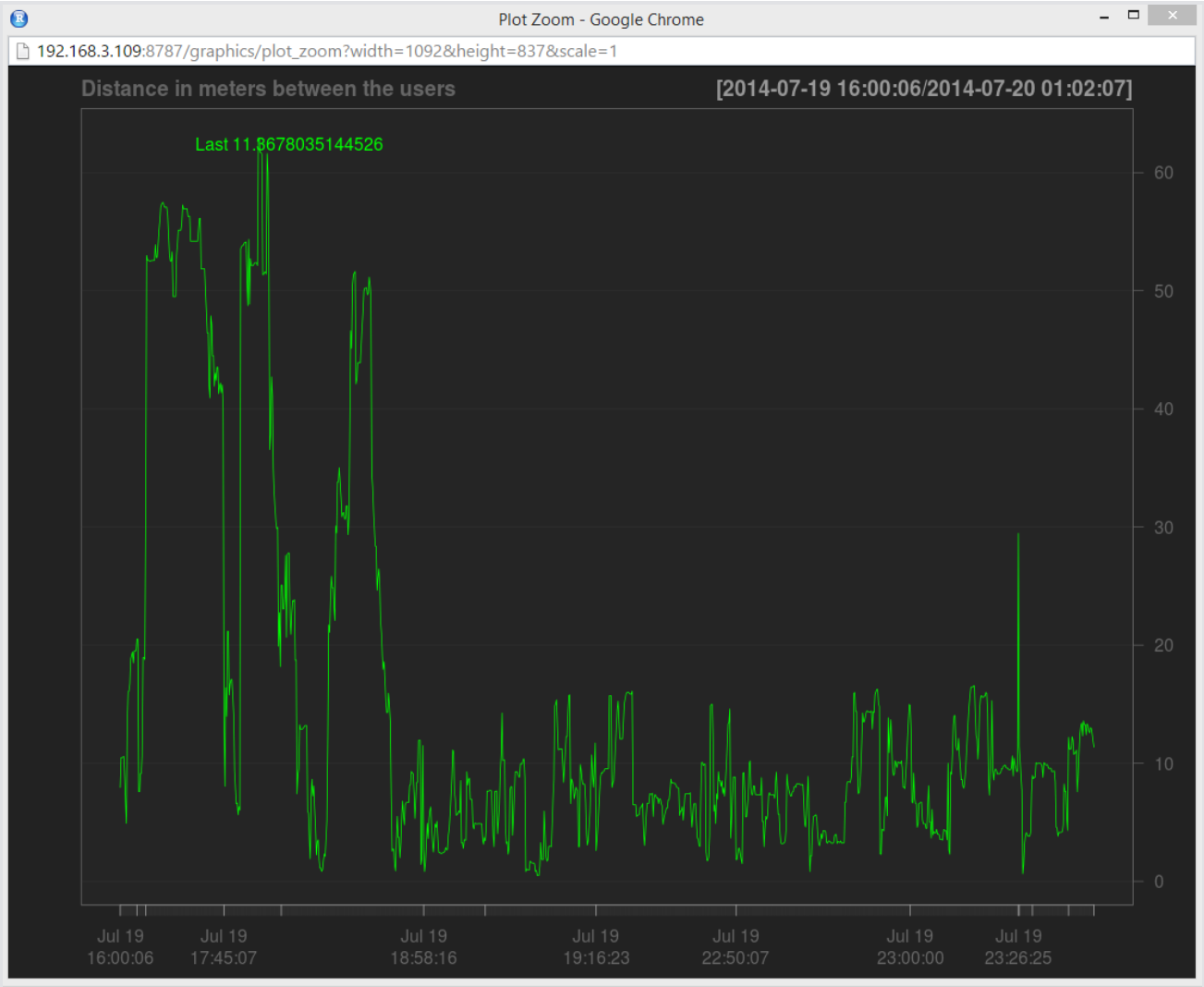
The program can then be run with the following command:

```
> findMeetings("3c3649fb", "5e7b40e1", data)
 Based on the data the two users can have met the first time at 2014-07-19 18:06:09
 Location: Floor:  3  X: 109.06043  Y:  97.14313
 Distance between the users:  1.926  Seconds between sightings: 1.093
```

We run it again and give the optional argument plotDistance=TRUE:

```
> findMeetings("3c3649fb", "5e7b40e1", data, plotDistance=TRUE)
 Based on the data the two users can have met the first time at 2014-07-19 18:06:09
 Location: Floor:  3  X: 109.06043  Y:  97.14313
 Distance between the users:  1.926  Seconds between sightings: 1.093
```

If the program is given the additional parameter of plotDistance=TRUE it will produce a graph with the distance of the users as a function of time:



Execution time:

```
> system.time(findMeetings(uid1, uid2, data))
 Based on the data the two users can have met the first time at 2014-07-19 18:06:09
 Location: Floor:  3  X: 109.06043  Y:  97.14313
 Distance between the users:  1.926  Seconds between sightings: 1.093
   user  system elapsed
  0.908   0.020   0.928
```

On a i5-2500 it executes in less than 1 second.

2. Running from command-line

The command can also be run from the command line:

```
$ Rscript analyzeMeetings.R 3c3649fb 5e7b40e1

 Based on the data the two users can have met the first time at 2014-07-19 18:06:09
 Location: Floor:  3  X: 109.06043  Y:  97.14313
 Distance between the users:  1.926  Seconds between sightings: 1.093
 In total there are 44 timestamps with the users in proximity of each other.
```

3. Using scala

Compiling:

```
$ scalac analyzeMeetings.scala
```

Running:

```
$ scala analyzeMeetings 3c3649fb 5e7b40e1
 Analyzing if the two userId's specified as arguments have met in the building.

 Based on the data the two users can have met the first time at 2014-07-19 18:06:09
 Location: Floor:  3  X: 109.06043  Y:  97.14313
 Distance between the users:  1.926  Seconds between sightings: 1.093
 In total there are 44 timestamps with the users in proximity of each other.
```

This versions execution time was also below 1 second after the speed optimizations and on a laptop with SSD-drive.

The additional displayed information

With these additional measures we can get quite a good approximation for the reliability of the meeting to occur:

- Distance between the users
- Seconds between sightings (We accept up to 120 seconds old timestamps, this can easily be changed)
- Number of timestamps with the users seen together

This is in addition to the first timestamp when the users have been seen together.

To improve in later versions:

- Fill in the paths for assumed movements based on speed, trajectories etc.
- Error and exception-handling and more informative messages etc.
- More advanced logic for detecting if users move together etc. (Would need to know more about the location and how the data is collected to do this well)
- Add the plotting feature to the scala version