**CA 4 - Perform Analysis on a 5000 line dataset**

**Assignment 4** is based on transforming a large dataset in text format - over 5000 lines of text.

You will need to scrub (clean) the data and place it into the relevant holder/container objects.

Once in these objects you will see that there are 422 different sets of commit objects.

So your task will be to analyse these 422 objects that are in a list and come up with 3 interesting statistical pieces of information for this dataset with supporting evidence of "interestingness'

You code for calculating the analysis should be documented and tested.

Test should be in a separate file runnable from the command line.

Your statistical analytics conclusions should be in a word document explaining in approximately 500 words the information that you have gleamed from the dataset.

You will be required to submit your code via github along with all documentation and tests.

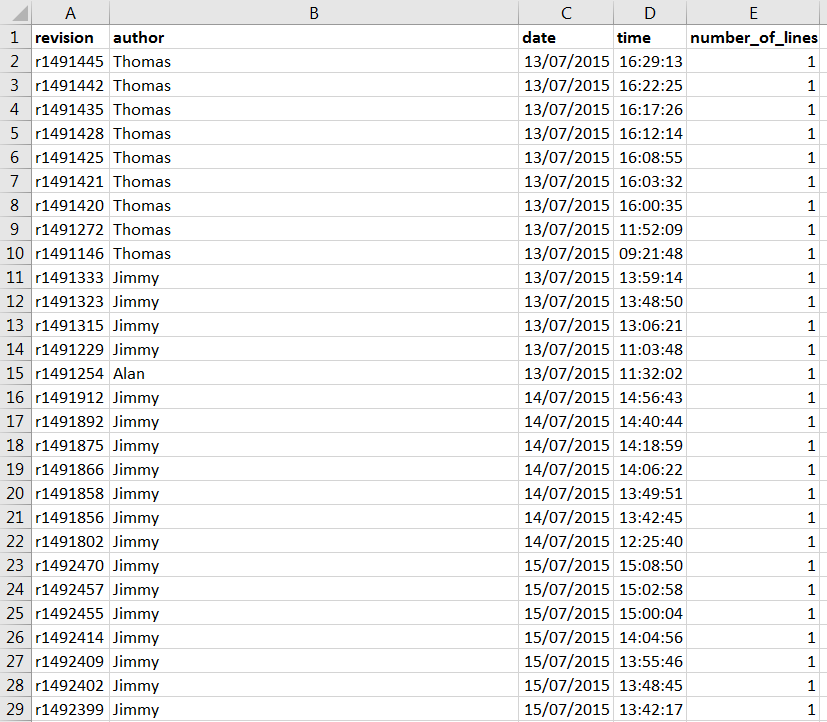
**Solution:**

**Step1: Python Processing**

Stripping the file and identifying the important attributes:

* revision
* author
* date
* time
* number\_of\_lines

Sample CSV as generated from running python scripts.



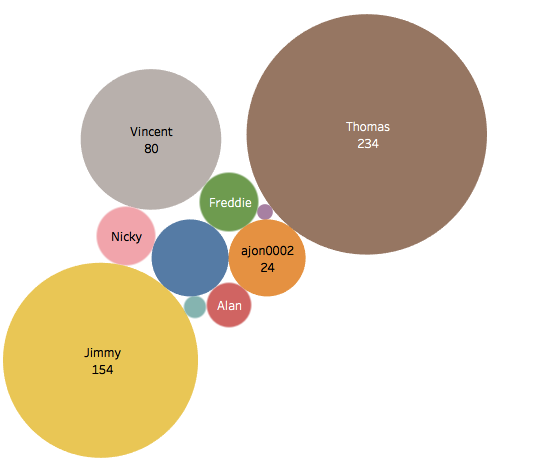
**Analysis**

**Step2**:

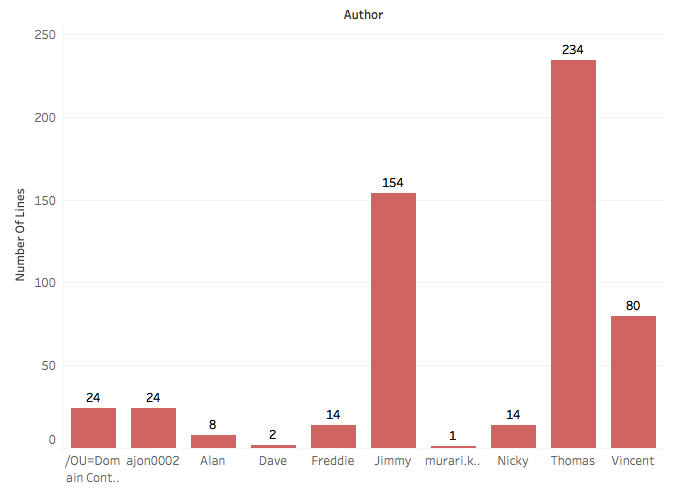
I used Tableau to slide and dice the CSV to generate insights about the data.

**Most Frequent User**

The author with the greatest number to comments is Thomas with 234 comments. Jimmy is second highest with 154 comments in the given time frame.

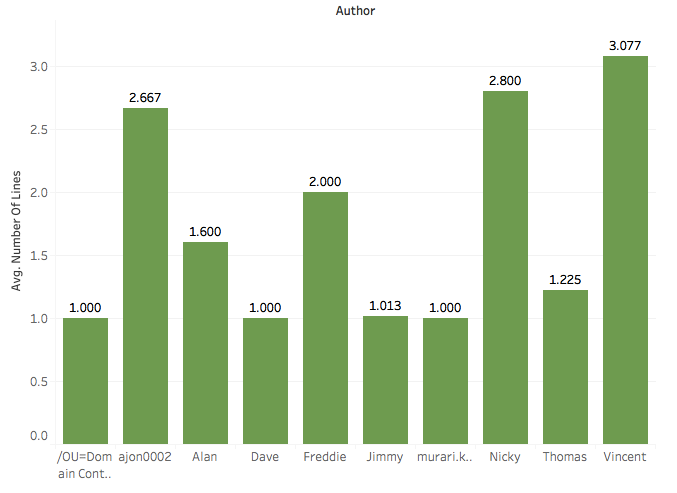


Another way of depicting Most Frequent User based on number of comments (Bar Chart)



**Step3: Average lines per commit**

Evaluating Average number of lines of each user during commit. Though Thomas has maximum number of commit but highest average number of lines per commit is Vincent with 3.077 lines per commit whereas Thomas has 1.225 lines per commit.



**Step 4: Peak Periods**

The below chart shows the trend line when commits were made to the database. You can see the peak was in July with 102 commits and minimum was in September with 44 commits.



**Step 5: Time of Commits**

The below charts shows the time of commits for various users. Jimmy and Thomas were mostly busy during business from 8am to 5pm with a little lull period from 12noon to 1300pm which would be their lunch time. The others users were actively making commits, busy with other responsibilities. We can infer assume from the graph that Jimmy and Thomas had main responsibilities to update and make updates to Database.

