Programming for Big Data

CA 4 Analysing commit data

Name: Beth Craig

Student Number: 10331736

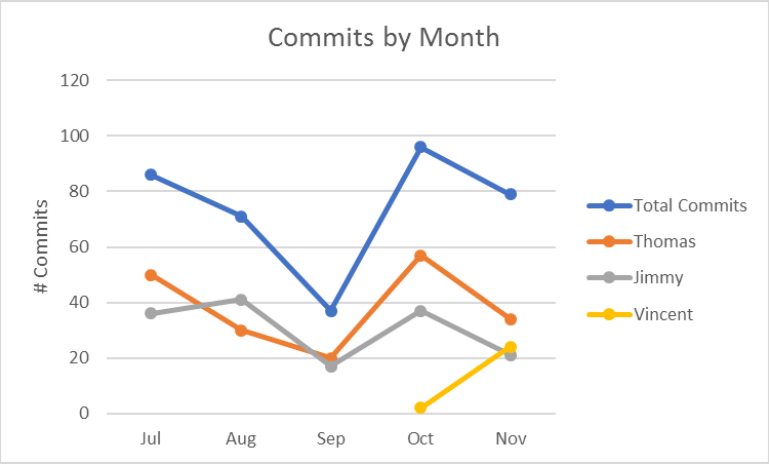
There is debate in the technology industry about the effectiveness of quantifying the output of software developers to appraise performance. Examples of metrics used include lines of codes written and number of commits. It is argued that metrics like these simply incentivise developers to writer longer and not necessarily better code and to commit more than is necessary.

'changes_python' is a text file with the data on 422 commits made between 13 July 2015 and 27 Nov 2015 by 10 software developers. The objective of this assignment was to analyse this data and produce 3 metrics to evaluate the performance of the software developers. The 3 metrics choosen were (1) number of commits per month, (2) number of unique days at least one commit was made and (3) number of files added or modified per author. The first two metrics evaluate consistency while the third is an estimate of productivity.

Two files were committed as 'R' and were ignored during analysis apart from reconciliation calculations. 3 users (Thomas, Jimmy and Vincent) accounted for 87% of the 422 commits and 95% of the 3011 files committed. Therefore, analysis focused on Thomas, Jimmy and Vincent as the % commits and files for the other 7 user names in themselves demonstrate a distinct difference in performance between the two groups.

Metric 1

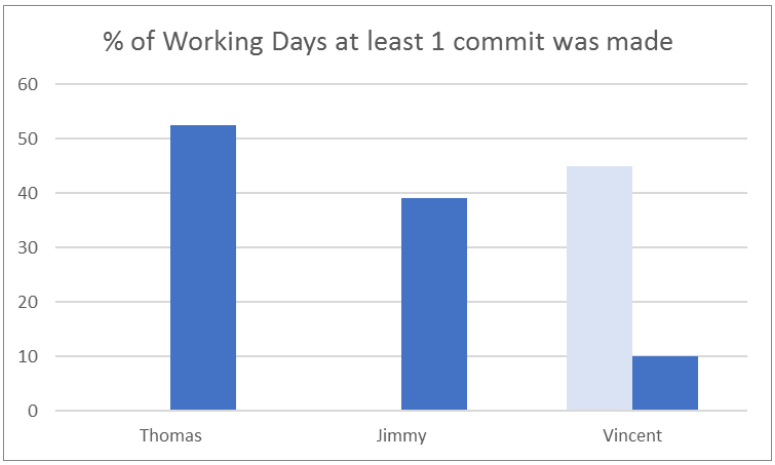
Refer to Graph 1. The total number of commits per month for all users decreases from July (102) to September (44) and then increases again in October (97) and November (96). A possible explanation for this is higher than average annual leave rates during the summer. Vincent did not commit until October, when he committed twice. He committed 24 times in November, comparable to 34 for Thomas and 21 for Jimmy. It is possible that Vincent only started in this role towards the end of October or was unavailable for some other reason.



Graph 1 The total number of commits for the business and number per author, split by month.

Metric 2

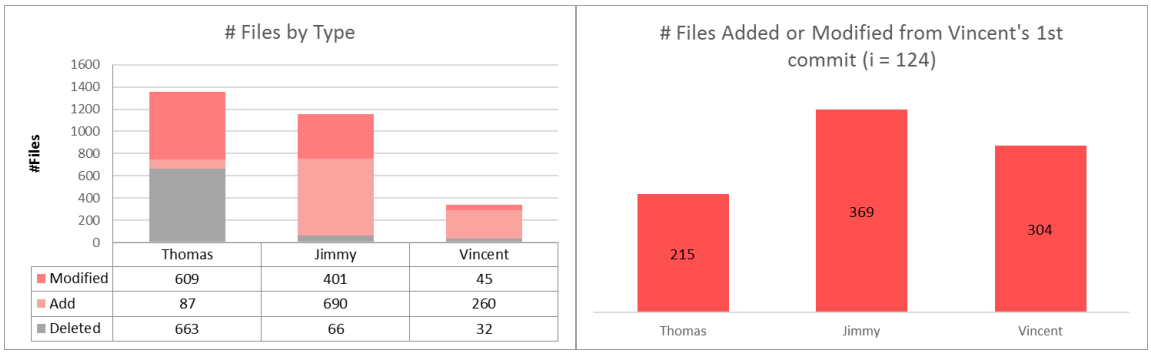
Thomas, Jimmy and Vincent committed at least one file on 51, 38 and 10 days respectively of the 97 working days between 13 Jul and 27 Nov 2015. However, if we assume based on Metric 1 that Vincent was not available for work until the end of October, his number of available working days decreases to somewhere in the region of 22. Normalizing this metric by number of available working days shows that Vincent actually commits at a rate similar to Thomas and Jimmy. Refer to Graph 2.



Graph 2 % of working days at least one commit was made: dark blue #working days = 97; light blue #working days = 22.

Metric 3

On average the time required to create a file (add) or modify (M) is likely to be significantly greater than deleting a file. Thomas, Jimmy and Vincent were responsible for adding or modifying 696, 1091 and 305 files respectively. Since there is evidence to suggest that Vincent was unavailable for work until the end of October this metric was re-calculated for the commits from Vincent's 1st commit (i = 124) to the last commit (i = 0). Over this time, Thomas, Jimmy and Vincent added or modified 215, 369 and 305 files respectively.



Graph 3 (a) Number of files added, modified or deleted per author for all 422 commits. **(b)** The number of files added or modified per author from Vincent's first commit (i = 124) to the last commit (i = 0).

Summary

3 (Thomas, Jimmy and Vincent) of the 10 developers accounted for 95% of files and 84% of commits. Thomas commits more often, but Jimmy adds or modifies more files. This analysis highlights the importance of using metrics that are normalized by an appropriate measure such as number of working days rather than absolute counts to avoid erroneous conclusions. When working days are taken in to account, Vincent's performance comparable Thomas and Jimmy's.

End of Document