Metric 5: Knots

Definition:

The number of crossing lines in a control flow graph.

Measured criterion:

Its used to evaluate the structure of the module. The high number of knots, the less maintainable and understandable the project is.

How knots work:

First, we need to define the line numbers for the module from the start till the end. Any hope from line number x to line number y is represented by (x, y). Let's assume there exist two jumps in the module, (i, k) and (j, l). Without loss of generality, there exist a knot if and only if,

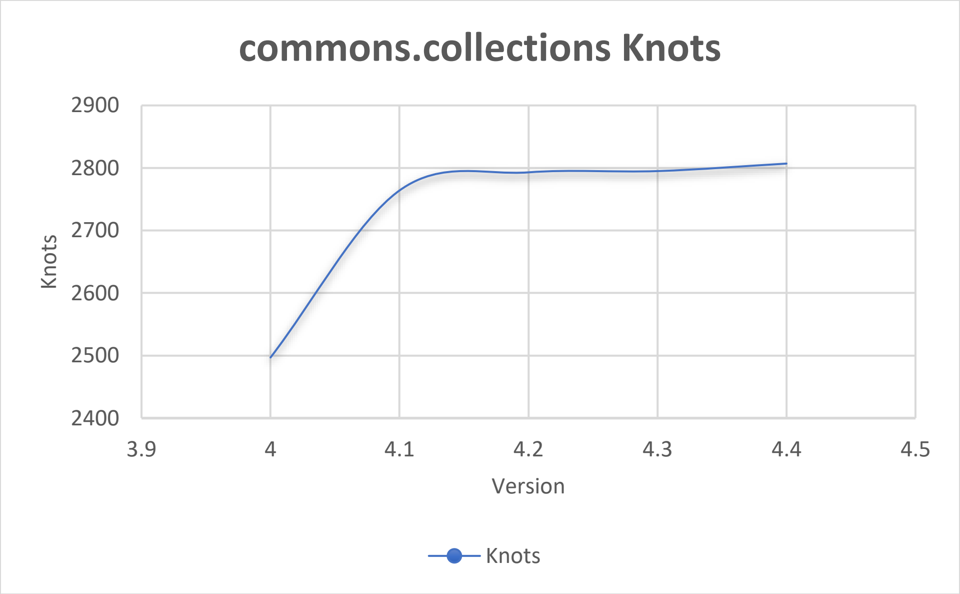
$$\min(i, k) \leq \min(j, l) \leq \max(i, k) \leq \max\,(j, l)$$

In our implementation we have used SciTools Understand to evaluate knots. For a particular version of a particular project we have calculated the total number of knots for simplicity and compared them.

Metric analysis:

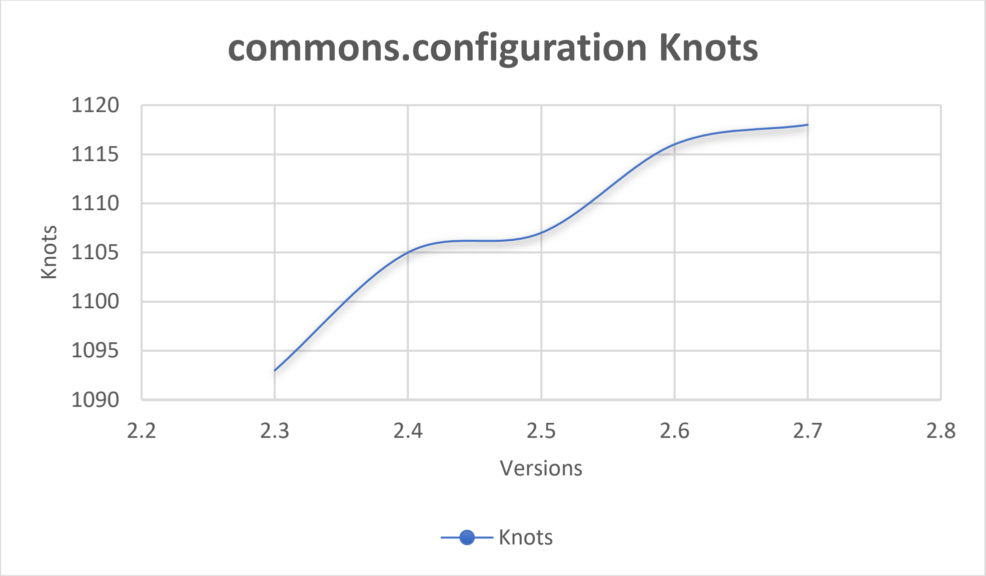
Following are the analysis of the metric for various projects and their various versions,

1. Apache.commons.collections



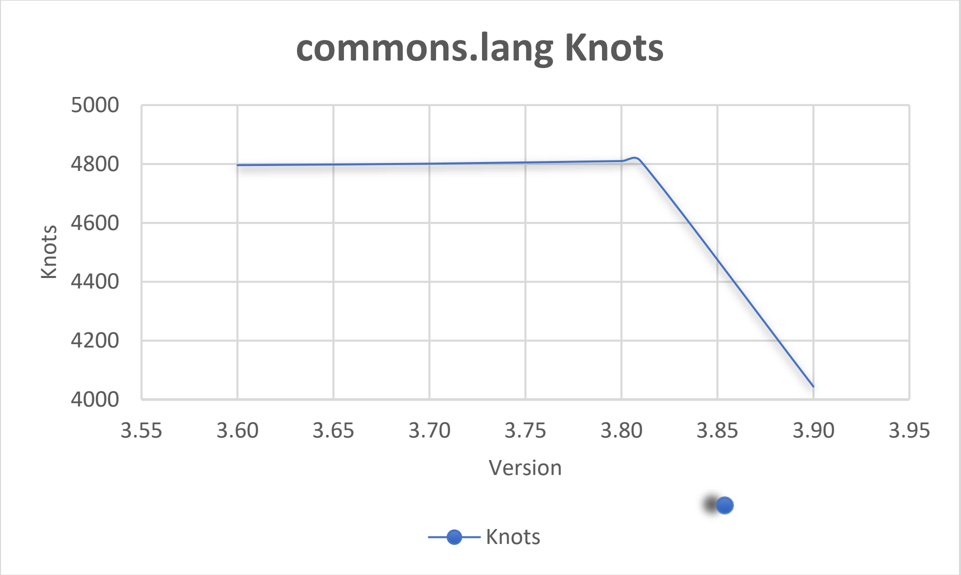
Here we can see that in early versions of the commons.collections the number of knots is less but as the version is updated to its higher version, the complexity increases and due to which the overall complexity to understand the project implementation increases. One would not be able to make any further modification if the complexity is high and therefore the maintainability is less. We can notice that after a certain stable version the increase of knots reduces to minimal.

2. Apache.commons.configuration



In the above graph we can see that commons.configuration shares the similar growth of increase in knots with commons.collections as the version upgrades. But the only difference is that the increment of the knots is quite less compared to commons.collections.

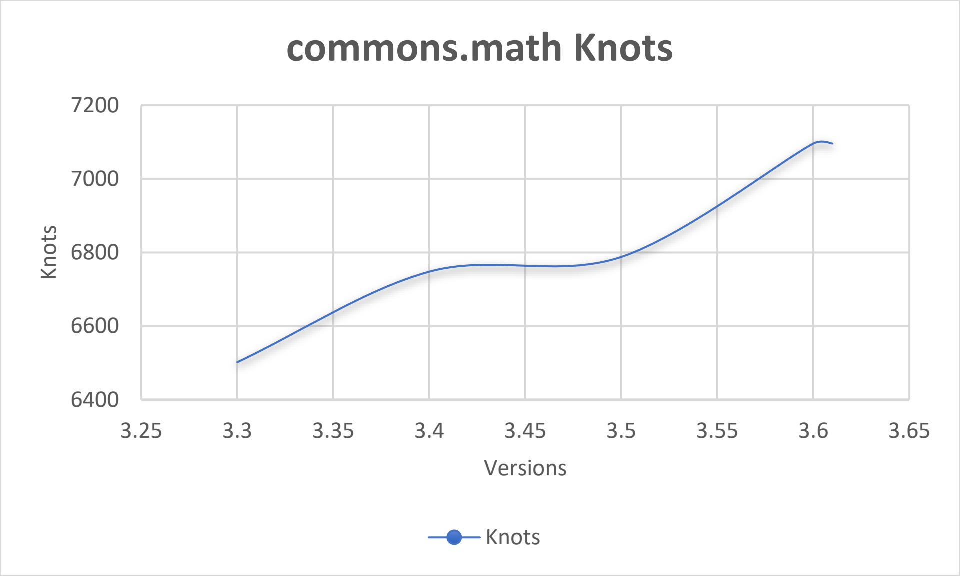
3. Apache.commons.lang



We can observe that commons.lang do not share similar fashion as commons.collections has. In early stages the project had a constant growth in number of knots but as to the latest version, the number of knots reduces dramatically. The sudden drop in the knots implies that the latest version is more maintainable compared to previous versions.

Metric 5: Knots

4. Apache.commons.math



In the above line graph, we can see that it is also growing the number knots as the version upgrades but as the version upgrades, the complexity of the project is getting stable. Out of all the four selected projects we can see that commons.math has the high number of knots and we can conclude that commons.math is highly complex and less maintainable compared to all above mentioned projects.

References:

Woodward, Martin R., Michael A. Hennell, and David Hedley. "A measure of control flow complexity in program text." *IEEE Transactions on Software Engineering* 1 (1979): 45-50.

Conte, Samuel Daniel, Hubert E. Dunsmore, and Y. E. Shen. *Software engineering metrics and models*. Benjamin-Cummings Publishing Co., Inc., 1986.

http://www.dmi.usherb.ca/~frappier/Papers/tm2.pdf