*Evolution – Lehman – Stats Reference*

Project selection -> set selection criteria (Have this already chosen) -> 500 initial projects (50 per language) -> random choice of 100? 200?

Data Structure -> organize into weeks -> remove first year (outliers, look only at mature projects evolution etc.)

|  |  |  |  |
| --- | --- | --- | --- |
| *Law* | *Hypothesis* | *Measure/Metric* | *Test* |
| 1.(1974) "Continuing Change" — an E-type system must be continually adapted or it becomes progressively less satisfactory[[4]](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-laws1-5-4)  6. 1991) "Continuing Growth" — the functional content of an E-type system must be continually increased to maintain user satisfaction over its lifetime | If a projects commits increase, stargazers will also increase and vice versa. (law 1 and 6) | Each projects commits and stargazers are organized into weekly counts/total per week. *Growth?*  P1-C = {20,22,11,40}  P1-S = {12,22,26,32} | 1) Get the mean of each projects stargazers and commits.  P1-C-Mean = 30  P1-S-Mean = 27  ….  2) Perform correlation (pearson/spearman) on the commits/stargazers vectors of means. |
| 2. (1974) "Increasing Complexity" — as an E-type system evolves, its complexity increases unless work is done to maintain or reduce it[[4](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-laws1-5-4) | Total lines of code increases as software evolves (law 2) | time  LOC/*growth*? | Des –  Recommends just doing an average growth rate instead of weekly |
| 3. 1974) "Self Regulation" — E-type system evolution processes are self-regulating with the distribution of product and process measures close to normal[[4]](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-laws1-5-4) | Issues, additions and deletions over time will be normally distributed (law 3) | Issues, additions, deletions counts as weekly intervals | 1) Apply shapiro wilks to each of the 3 vectors for the project.  2) get the p number  3) state how many projects for each of the three metrics are in the 0.05 threshold of significance |
| 4. (1978) "Conservation of Organisational Stability ([invariant work rate](https://en.wikipedia.org/w/index.php?title=Invariant_work_rate&action=edit&redlink=1))" - the average effective global activity rate in an evolving E-type system is invariant over the product's lifetime[[4](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-laws1-5-4)  5.(1978) "Conservation of Familiarity" — as an E-type system evolves, all associated with it, developers, sales personnel and users, for example, must maintain mastery of its content and behaviour to achieve satisfactory evolution. Excessive growth diminishes that mastery. Hence the average [incremental growth](https://en.wikipedia.org/w/index.php?title=Incremental_growth&action=edit&redlink=1) remains invariant as the system evolves.[[4]](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-laws1-5-4) | As software evolves changes to lines of code should not fluctuate (law 4)  Total lines of code in the software increases incrementally at a constant rate (law 5) | LOC (weekly growthRate)– organized into weeks as a vector with the amount of LOC in that interval  Growth rate  5 -Growth rate variance low –fewer issues/stargazers  Variance high – more issues/fewer stargazers | 1) get the variance for each project LOC  2) then do a mean of the variances |
| 7.(1996) "Declining Quality" — the quality of an E-type system will appear to be declining unless it is rigorously maintained and adapted to operational environment changes[[5]](https://en.wikipedia.org/wiki/Lehman%27s_laws_of_software_evolution#cite_note-5) | Project issues will increase as code churn decreases (law ~~6~~7) | Count of issues and LOC as weekly intervals | Same process as the first HP, but looking for no correlation? (ie not positive) |
| 8. (1996) "Feedback System" (first stated 1974, formalised as law 1996) — E-type evolution processes constitute multi-level, multi-loop, multi-agent feedback systems and must be treated as such to achieve significant improvement over any reasonable base | As the number of issue comments increases the number of issues should decrease (law 8) | Count of issues, count of issues comments organized into weeks.  *Stargazers vs comments???* | Same as the above but difference metrics/measures. |