**Handling the 100 project statistics**

100 projects – 100 (growth rate, correlation pairs) for a metric (s) – 100 -> 1 (mean, how?)

What is the best solution? (Average cor value?)

‘How about combining all 100 projects ie summation (deletion-addition) pairs, then correlate’. Should they be appended to one another or…?

**HP – Prep and analysis**

**H1 V1** - If the amount of commits decreases the amount of star gazers will also reduce (law 1 + 6)

I believe this has been ruled out – you gave a good point.

**H1 V2 (Des suggestion)** - If the software does not change stargazers will not increase (law 1 + 6)

‘I suspect, once a stargazer probably you remain one unless there is some motivation to stop. So maybe its growth in stargazers that will change once the commits stop i.e. software doesn’t change new stargazers don’t appear’

Apply growth rate intervals (week to week growth) to stargazers + commits then see if they correlate? (then the 100 -> 1 problem rears its head)

Plot a scatter graph of commits v stargazers – if commits are low then stargazers should also low (visual representation)

Any other better statistical way to measure this?

**H1 V3 (Jordan 2)** – Might be possible just to state software change (commits) and stargazers will correlate (increase together, decrease/stop together) – apply Pearson or spearman

**H2** – Total lines of code increase as software evolves (law 2)

Get growth rate between each week interval for LOC

Get the mean growth rate for each project from ^^ for each

A simple percentage of total positive means?

What % is required to satisfy HP then?

**H3** - Issues, additions and deletions over time for will be normally distributed (law 3)

Shapiro wilks test - > shapiro.test(testset);

This will get a p value and actual test result – so each project will have 3 of this pair for each metric, so 3 x 100 p values & 3 x 100 actual test results

What then? (mean for each metric?)

**H4** - As software evolves changes to lines of code should not fluctuate (law 4)

‘The average effective global activity rate on an evolving software system does not change over time; that is, the amount of work that goes into each release is about the same.’

\*didn’t account for release aspect (blue)

Des input – ‘So new development decreases as maintenance increases – net effect releases the same amount of work, So it’s to do with releases’

<https://api.github.com/repos/jquery/jquery/releases>

<https://api.github.com/repos/ruby/ruby/releases>

<https://api.github.com/repos/michael/github/releases>

<https://api.github.com/repos/michael/github/tags>

<https://api.github.com/repos/jquery/jquery/tags>

<https://api.github.com/repos/ruby/ruby/tags>

‘releases’ don’t seem to be used on github, tags is an alternative.

a) Still need to know the best way to measure this (based on a settled HP)

b) what measure to use to answer a)

**H5** – Total lines of code in the system increases incrementally at an average rate (law 5)

a) At each week period get the amount of increase in LOC – sort all values in array for each project

b) Average rate = mean of the array values for each project?

c) How to determine the increments follow and average rate?

d) Should the first week outlier be discarded? Tends to be a huge dump of code

**H6** - Project issues will increase as code churn decreases (law 7)

Very similar to H1 in terms of applying statistics + summation then correlation?

**H7** - As number of issue comments increase the rate of closing issues should also increase (law 8)

\*need to double check on getting comments

Again it’s a comparison of two metrics and if they correlate, so maybe summation of 100 then correlate