An illustrative *examination of 101repo*

Ralf Lämmel
on behalf of 101companies.org

How does the LOC metric compare for Java versus Haskell?

Go to 101repo @ github.

https://github.com/101companies/101repo

Select "Downloads".
Download as *zip* or *tar.gz* and unpack.

https://github.com/101companies/101repo/downloads

... or you git clone / fork,
if you are familiar with git/github.

Determine LOC for all ***Java*** and ***Haskell*** files in 101repo

$ ./analysis.py ../101repo/contributions **.java**

573 files found.

Mininum LOC: 3

Maximum LOC: 5917

Median LOC: 41

$ ./analysis.py ../101repo/contributions **.hs**

91 files found.

Mininum LOC: 6

Maximum LOC: 207

Median LOC: 24

Rushed conclusion:

Haskell modules are more concise than Java modules.

Disclaimer: *This is nowhere close to a scientifically relevant statement!*

- What is our hypothesis anyway?
- NCLOC may be better than LOC.
- How comparable are the file sets really?
- ...

See these conferences for proper science:
CSMR, ICPC, ICSM, MSR, SLE, ...

Summary:
What's the point of this demo?

- To illustrate cheaply the value of examining 101repo.
- To show how easily one gets started with examination.
- To motivate more profound experiments.

Thanks!

- *Contact 101companies*
  - *Email*: 101companies@gmail.com
  - *Twitter*: @101companies
- Code for this presentation:
  - https://github.com/rlaemmel/loc101demo