Why you should swear more!!

Exploring the Correlation between the Use of Swearwords and Code Quality in Open Source Code
Disclaimer: Naughty Language Ahead!!
Contents

- Idea and Approach
- Methods
  - Data Gathering
  - Data Evaluation
  - Data Analysis
- Results
- Conclusion
Idea and Approach
Idea

Occurrences of words in the Linux kernel source code over time

Source

Source
Approach

Initial Hypothesis:
- There is no difference in code quality with regards to the use of swearwords in open-source code

4 Questions
- How do we gather our data?
- How do we identify Swearwords?
- How do we measure Code Quality?
- How do we compare the two samples?
Data Gathering
Definitions

Star-repos: repositories $\geq 4$ stars

Swear-repos: repositories $\geq 1$ swearword
Git API

1. Identify Parameters
2. Programming Language
3. Search term
4. URL Construction
5. Send URL and receive data
Why we chose the Git-API?

Pro
- Easy to learn and use
- Already existing code search functionality
- Fast

Contra
- “Only” 1000 results per search-query
- Primary and secondary rate limit
- Timeout
Restrictions

Repository
- Size ≤ 625MB
- Execution time of SoftWipe < 1h

Swearwords
- No swearwords that can be misinterpreted e.g.:
  - Ass and Asses
  - |swearword| > 3
Crawling flowchart

1. Swearword
2. URL-Construction
3. Running URL through requests library
4. Relevant data
5. Pagiation
6. Ratelimit hit
Data Evaluation
SoftWipe

- Benchmark for scientific software in C / C++
- Uses static and dynamic code analysers
  - Number of compiler warnings / assertions / tests
  - Code style violations
  - Modularity of the software
- returns a score between 0 (low adherence) and 10 (good adherence)
Counting Swearwords

How?
- NLP vs. regex
  - → regex due to time constraints

Regex:
- \bS*([-_][0-9]) swearword ((-__)[A-Z][0-9])S*?\b
  - what_the FUCK
  - fuck10
  - fuckThis
- \bS*([0-9])? swearword_first_cap ((-__)[A-Z][0-9])S*?\b
  - whatTheFUCK
  - Fuck
  - this_Fuck-ingOddity
- \b\S*([-_][0-9])? swearword_caps ((-__)[0-9])S*?\b
  - WHAT_THE FUCK
  - FUCK
  - FUCK_MY-badExamples
Evaluation flowchart

List of repositories to evaluate

multiprocessing pool

...
Runtime Bottlenecks and their Optimisation

Execution time of SoftWipe → Parallelisation
- Using multiprocessing library
- Creating a process pool
- 6 times faster due to 6 cores being utilised

Swearword counting → re2 library
- guarantees execution in linear time
- NFA → DFA
- 579s re → 8s re2
Data Analysis
Data Analysis Goals

Defining our goals:

- Find inferences of sample → underlying population
- Find a relationship between the two samples → relationship of the target and the general population

- To determine if swear-repos do have a higher/lower code quality than the general population.
How accurate is the sample mean $\bar{X}$?

Instead of a point estimator $\rightarrow$ confidence interval = interval of plausible values.

Accuracy can be determined by its width.

Requires:
- The population has to be normally distributed
- The true value of the population standard deviation is known.

Given a large enough sample the requirements can be assumed to be true $\rightarrow$ Central limit theorem.
Bootstrapping

- re-sampling method that returns measures of accuracy for a given sample statistic
  - confidence interval, standard error
- does not assume any underlying distributions

The basic idea behind bootstrapping:
- It is generally done by re-sampling the original sample with replacement
- calculate a point estimate of that newly generated sample
- repeat x amount of times (x=9999 usually)
Analysis Methods

- **Kolmogorov-Smirnov test**
  - Determines whether two samples are from different distributions

- **Welch’s t-test**
  - Approximates whether the means of two population are different without assuming equal variances
Results
Scatterplots

Scatter plot of the star-repos

Scatter plot of the swear-repos
Star-repos

Histogram of the SoftWipe score of star-repos

Q-Q plot for the distribution of the SoftWipe score of star-repos

$R^2 = 0.9958$
Swear-repos

Histogram of the SoftWipe score of swear-repos

Q-Q plot for the distribution of the SoftWipe score of swear-repos

$R^2 = 0.9836$
Test results

- KS-test
  - statistic ≈ 0.20 and p-value ≈ 3.17 * 10^{-89}
- Welch’s t-test.
  - statistic ≈ 16.71 and p-value ≈ 2.04 * 10^{-61}

→ correlation between swearing and an improvement in code quality

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-repos</td>
<td>5.41</td>
<td>[5.38 - 5.45]</td>
</tr>
<tr>
<td>swear-repos</td>
<td>5.87</td>
<td>[5.81 - 5.93]</td>
</tr>
</tbody>
</table>
Conclusion?

- Initial Hypothesis:
  - There is no difference in code quality with regards to the of swearwords in open-source code

- Swear-repos exhibit a statistically significant higher average code-quality
  - 5.87 compared to 5.41

- But what about the clusters??
Cluster Analysis

Manual look at repositories in Cluster 1 and 2 to identify common denominator

- Cluster 1 (PintOS):
  - Introduction to OS at Stanford [2]

- Cluster 2 (OS/161):
  - Teaching OS used by University of Toronto and others
Swear-repos

Histogram of the SoftWipe score of swear-repos

Q-Q plot for the distribution of the SoftWipe score of swear-repos

$R^2 = 0.9975$
Test results

- KS-test
  - statistic ≈ 0.042 and p-value ≈ 0.0006
- Welch’s t-test.
  - statistic ≈ -0.54 and p-value ≈ 0.59

→ NO correlation between swearing and an improvement in code quality

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>star-repos</td>
<td>5.41</td>
<td>[5.38 - 5.45]</td>
</tr>
<tr>
<td>swear-repos</td>
<td>5.40</td>
<td>[5.34 - 5.46]</td>
</tr>
</tbody>
</table>
Conclusion
Conclusion and Outlook

- Swear-repos exhibit do not exhibit a statistically significant higher average code-quality

- Preferably:
  - It does not matter whether you swear or not so you might as well do it

- Look at the Code Quality of repositories with a lot of swearwords

- Publish a Paper
Questions?
Sources:
