Inflection Points in Open Source Software Evolution

James Walden[1] Kuljit Kaur[2] Noah Burgin[3]

[1]Northern Kentucky University

[2]Guru Nanak Dev University, Amritsar, India

[3]University of Tennessee, Knoxville

Project Goals

Identify inflection points in open source evolution time series

- Number of commits per month
- Number of unique authors per month
- Number of files modified per month

Classify types of inflection points by

- Sign: positive or negative changes
- Magnitude: relative size of changes

Research Questions

1. Is software evolution usually stable (no changepoints)?
2. Are projects with changepoints likely to change once or multiple times?
3. Do changepoints tend to be in the same direction in a project or does sign vary?
4. Are changepoints in one time series correlated with those in other series?
5. Can we develop a classification for changepoints?

WoC Data Flow

1. Select sample of projects meeting criteria (nauthors $>$ 50, ncommits $>$ 5000) from R dataset snapshot in Mongo database.
2. Data cleaning and refinement
   - 2.1 Remove projects whose start timestamps were zero.
   - 2.2 Remove projects with $<$48 months between earliest and latest commits.
3. For each project in the sample
   - 3.1 Use `p2c` map to find commits for the project.
   - 3.2 Use `c2ta` map to obtain timestamp and author for each commit.
   - 3.3 Run a python script to create a CSV file with time series.

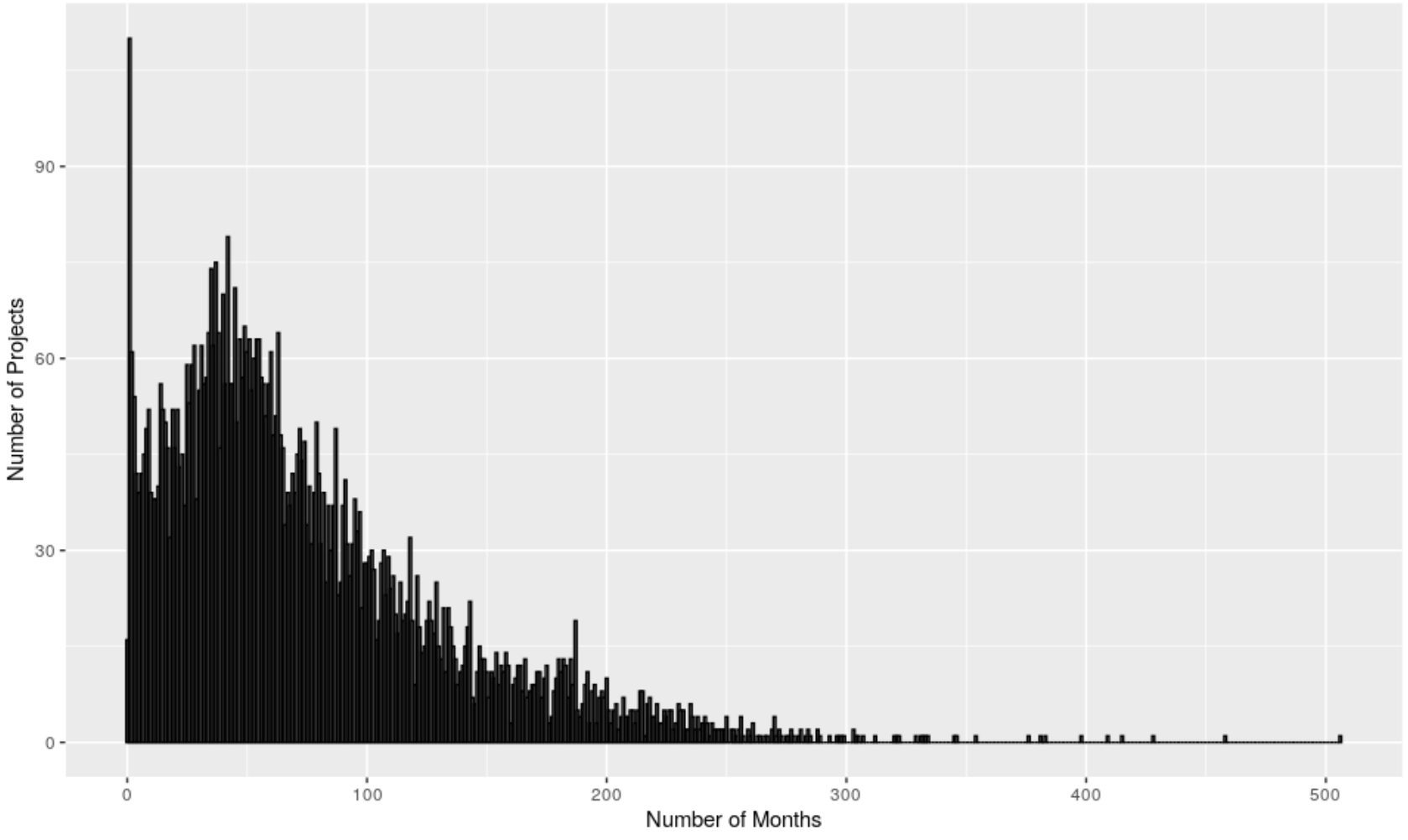
Project Data Set

We found 8311 projects that had sufficient history to meet our criteria after data cleaning, but

- We could only generate time series for 6514 projects.
- 16 of the 6514 time series CSV files contained no data.
- 110 of the 6514 time series CSV files had data for only one month.
- Approximately 2000 of the 6514 time series were shorter than 48 months.

We need to debug our process to determine why we're not finding all of the commits via the maps that we were expecting to find based on MongoDB.

Projects by Duration



World of Code Problems

- On two separate days, the sampling web application returned no results for any search. While we were able to get searches working on another day, we decided it was too unreliable to depend on.
- It wasn't clear what was the best source of project data to select our sample of projects from.
- It took considerable time to determine that the Clickhouse database could not be used to get the commits per month time series data we needed.
- The commits data from the maps we have obtained is not consistent with the MongoDB data about each project. We're not sure yet if that's a problem with our scripts or an actual data inconsistency.

Changepoint Detection

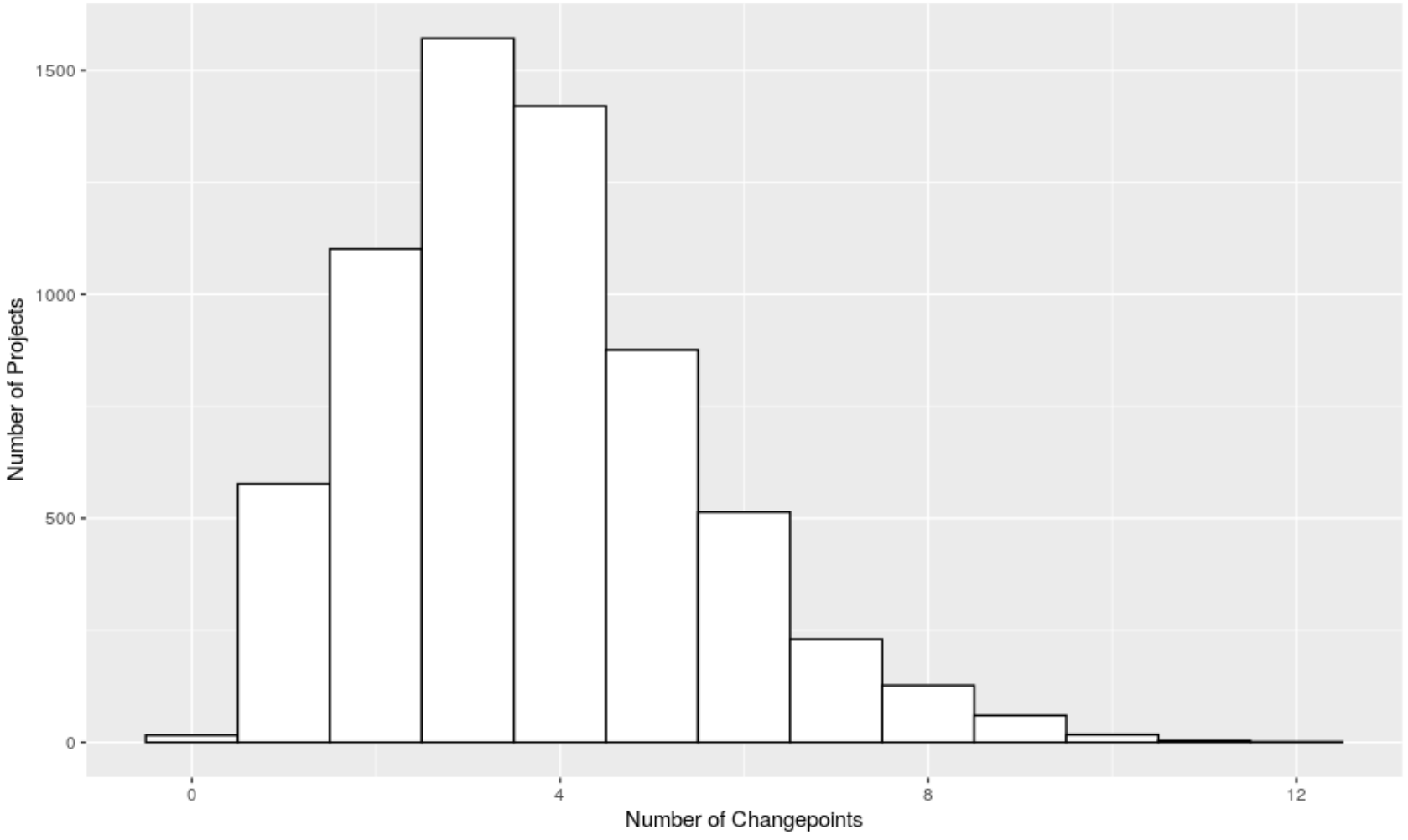
We tested three techniques for changepoint detection on

- Synthetic time series constructed with known changepoints.
- Commits time series for OpenSSL, where we know were changepoints exist from external events and prior analysis.

The three techniques were

- Segmented regression, which always finds a changepoint.
- Changepoint detection (PELT).
- Nonparametric changepoint detection.

Projects by Number of Changepoints



Future Plans

1. Identify and fix problems in data collection process.
2. Tune algorithms and parameters for changepoint detection.
3. Classify changepoints by size and magnitude.
4. Collect unique authors and files changed time series data.
5. Classify changepoints of new time series.
6. Identify correlations in changepoints across different time series for the same project.