Data Exploration

1. Jira search terms:
   - [x] ~~Reporter is not equal to assignee~~
   - [x] ~~Both fixed and resolved~~
   - [x] ~~Time duration~~
   - [ ] (Filter test)

   a. `project = HADOOP AND resolved >= 2020-01-01 AND resolved < 2023-01-01 AND resolution = fixed AND type = bug ORDER BY created DESC`

   b. `project = HADOOP AND resolved >= 2020-01-01 AND resolved < 2023-01-01 AND type = bug AND assignee != reporter AND (resolution = fixed) ORDER BY created DESC`

   c. `project = HADOOP AND resolved >= 2020-01-01 AND resolved < 2023-01-01 AND type = bug AND assignee != reporter AND (resolution = fixed OR status = resolved) ORDER BY created DESC`

   d. `project = HADOOP AND resolved >= 2020-01-01 AND resolved < 2023-01-01 AND type = bug AND (resolution = fixed OR status = resolved) ORDER BY created DESC`

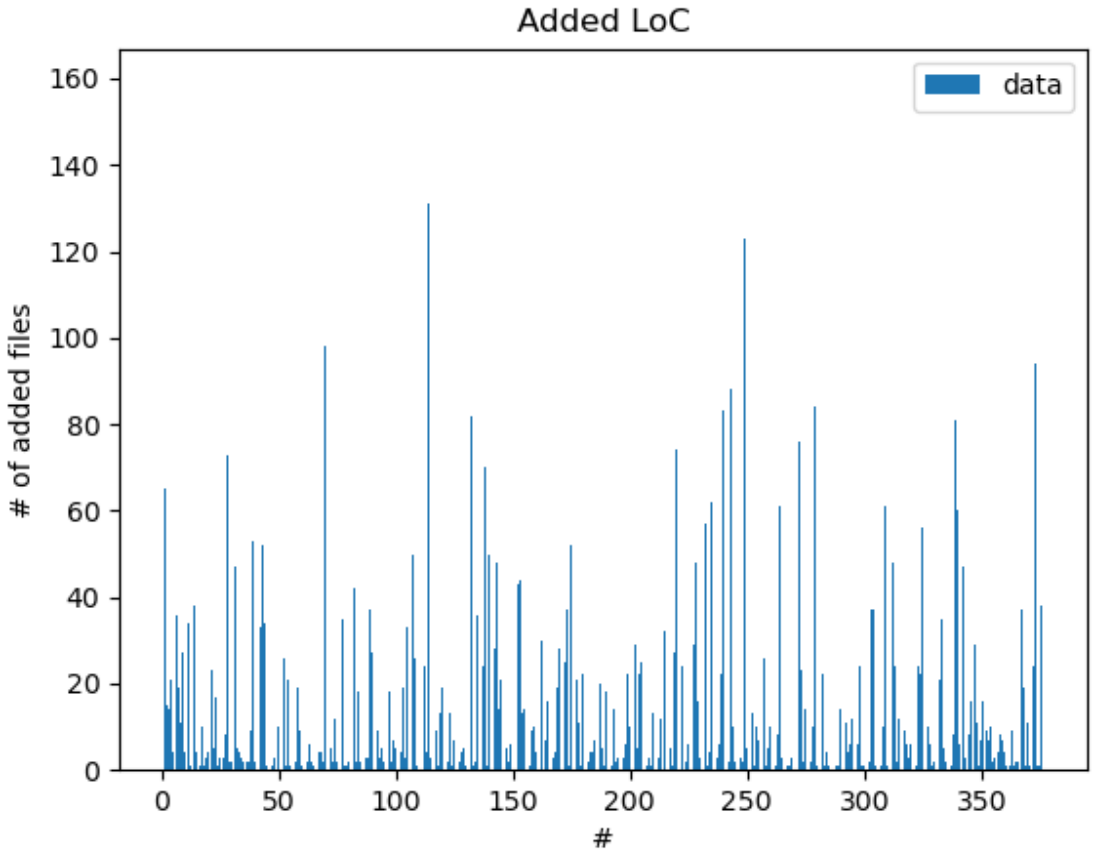
2. Prints the files related to bugfix commits

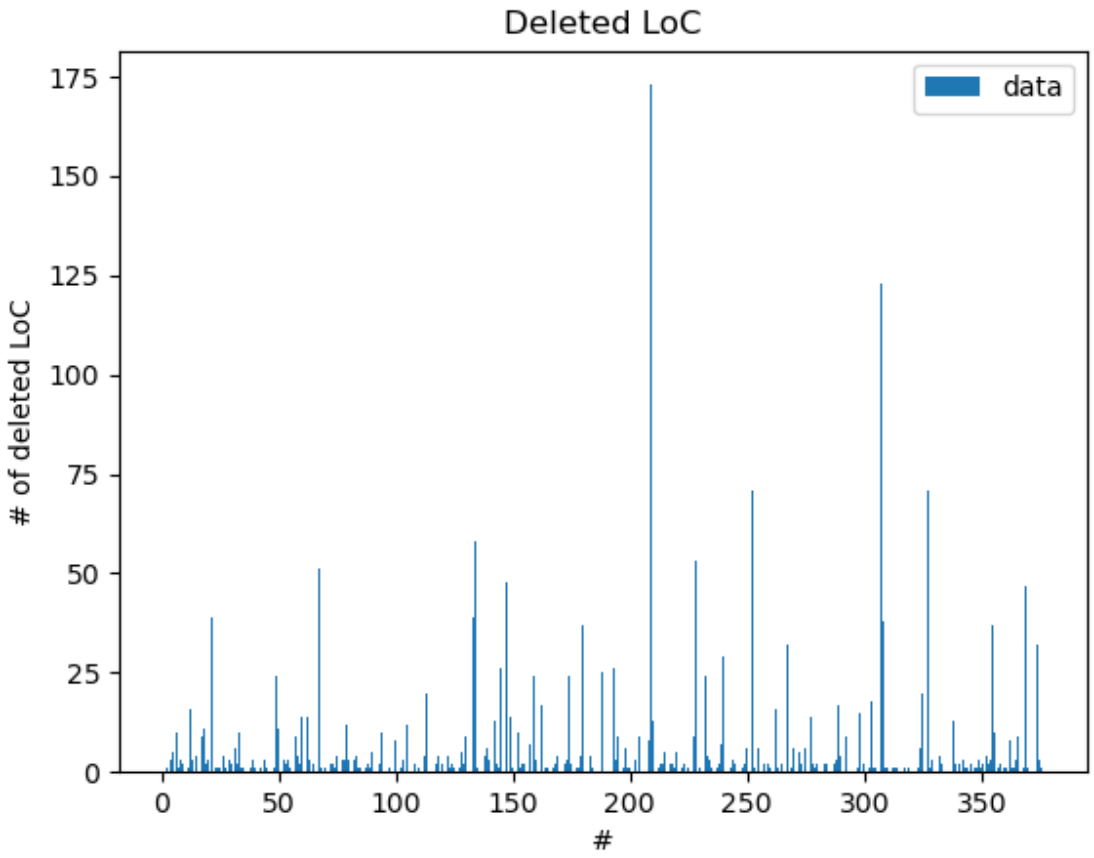
```
/Users/moeachto/.conda/envs/envs/bin/python /Users/moeachto/
[2022-10-25 15:43:40.481052] ---------
[2022-10-25 15:43:40.481240] Hadoop-HDFS
[2022-10-25 15:48:01.612012] Total GitHub Commits: 2878
[2022-10-25 15:48:01.612053] Total Jira Bug Fixes: 355
[2022-10-25 15:48:01.612061] Total GitHub Bug Fixes: 277
Run time: 0:04:21.131665

Process finished with exit code 0
```

3. Data stats

```
added:  [65, 15, 14, 21, 4, 36, 19, 11, 27, 4, 34, 1, 0, 38, 4, 1, 10, 1, 3, 4, 23, 5, 17, 1, 3, 3,
# of added files:  376
max of added LoC:  159
min of added LoC:  0
mean of added LoC:  16.66223404255319
std of added LoC:  23.508039804173077
deleted:  [0, 1, 0, 3, 5, 10, 1, 3, 2, 0, 1, 16, 3, 0, 4, 0, 9, 11, 2, 3, 39, 0, 1, 1, 1, 4, 1, 0,
# of deleted files:  376
max of deleted LoC:  173
min of deleted LoC:  0
mean of deleted LoC:  5.784574468085107
std of deleted LoC:  14.478299106997637
```


Added LoC
data
# of added files
#
0
20
40
60
80
100
120
140
160
0
50
100
150
200
250
300
350



4. How I process the data
    a. Get the bug fix related commits by matching the jira issues with GitHub commits by searching (e.g., HDFS-1)
    b. Get previous code versions and current bug fix versions based on generated bug fix commits
    c. Delete all the comments
    d. Delete lines start with *package, import, @*
    e. Delete the newly added file so that the buggy version and the corresponding fixed version can have the same line number
    f. Make one file in a line to generate buggy.txt and fixed.txt

5. Problems found when applying the existing model to new data
   There are several constraints for using the existing model
   a. Used .java file only (no .xml or others)
   b. Treat each class in the same commit as one piece of data
   c. Deleted all import and @ to fit the example data that the model used
   d. Deleted newly added files that did not show up in the previous commit
   e. Not sure how it deal with nested classes
   f. Can only parse 500 characters

6. Other problems
   a. We only used modified files in selected commits for creating our dataset, what about the relationship and dependencies between them and other unchanged files?
      - Dynamic trees
      - Static code analysis (java parser)
      - source/byte code analysis tool to filter lines of related data/control flow (WALA)
      - Dynamic analysis
      - Heuristics?
      - Dependency graph