SOEN 6611
SOFTWARE MEASUREMENT

Submitted to: Jinqiu Yang

GROUP J

Submitted by:

40084709 : Suruthi Raju

40129435 : Pranoti Mulay

40078258 : Avinash Damodaran

40088004 : Shalin Patel

40080612 : Niralkumar Lad

Outline

- Repositories Used
- Metric 1 , Metric 2 and Metric 4: Statement Coverage, Branch Coverage and McCabe Cyclomatic complexity
- Metric 3: Mutation Score
- Metric 5 and Metric 6: Maintenance Effort Estimation Model and Software Defect Density
- Correlations:
  - Metric 1 and 2 Vs Metric 3
  - Metric 1 and 2 Vs Metric 4
  - Metric 1 and 2 Vs Metric 6
  - Metric 5 Vs Metric 6

Repositories Used

Number	Repository	Version	Lines of Code
1	Apache Commons Logging	1.2	18K
2	Apache Commons Math	3.6.1	387K
3	Apache commons Lang	3.10	154K
4	Apache Commons Net	3.6	62K

Metric 1: Statement Coverage
Metric 2: Branch Coverage
Metric 4: McCabe Cyclomatic complexity

Tool, Process and Challenges

- Tool: JaCoCo
  - Why JaCoCo?
    - Easy integration with eclipse and free as well as Open-Source for Java.
- Process:
  - Imported projects in eclipse.
  - Resolved dependencies error.
  - Ran the tool i.e. JaCoCo to calculate metric 1,2 and 4.
- Challenges:
  - Resolving dependencies error.
  - Ignoring some test cases that were not detecting any coverage.

Results:

Number	Repository	Metric 1: Statement Coverage	Metric 2: Branch Coverage	Metric 4: McCabe Cyclomatic Complexity
1	Apache Commons Logging	0.51	0.3844	51.55%
2	Apache Commons Math	0.919	0.843	81.73%
3	Apache commons Lang	0.943	0.8728	81.34%
4	Apache Commons Net	0.543	0.3278	33.97%

Metric 3: Mutation Score

Tool, Process and Challenges

- Tool: PitClipse
- Process:
  - Install PitClipse and add plugin to POM.xml
  - Resolve the dependencies if any exists
  - Open terminal and write the following command:

    mvn org.pitest:pitest-maven:mutationCoverage
- Challenges:
  - Multiple time outs
  - Ignoring some test cases that were not detecting any mutation.

Results

Number	Repository	Metric 3 Mutation Score
1	Apache Commons Logging	23%
2	Apache Commons Math	79%
3	Apache commons Lang	85%
4	Apache Commons Net	24%

Metric 5: Maintenance Effort Estimation Model
Metric 6: Software Defect Density

Tool and Process

- Tool: CLOC
- Process:
  - Installing CLOC
  - Open terminal and write for the following line with the folder path
    - Metric 5

      cloc –diff folderPath1 folderPath2
    - Metric 6

      Cloc folderPath

Results
Apache Commons Logging

Version	Metric 5		Metric 6		
	DLOC	Effort Required (person hours)	LOC	No. of bugs	Software Defect Density
1.0.4	993	87.93	9564	23	2.404851527
1.1	4157	119.57	16604	9	0.542038063
1.1.1	568	83.68	18898	14	0.740819134
1.1.2	5806	136.06	18155	2	0.11016249
1.1.3	0	78	18150	0	0
1.2	83	78.83	18257	2	0.109547023

Results(Continue)
Apache Commons Math

Version	Metric 5		Metric 6		
	DLOC	Effort Required (person hours)	LOC	No. of bugs	Software Defect Density
1.0	15873	236.73	36846	13	0.3528198
1.1	4696	124.96	47871	24	0.5013474
2.0	85787	935.87	156848	66	0.4207896
2.1	13768	215.68	165423	51	0.3083005
2.2	21411	292.11	193422	68	0.3515629
3.4	135802	1436.02	362017	12	0.0331476
3.6.1	18473	262.73	387777	37	0.0954157

Results(Continue)
Apache Commons Lang

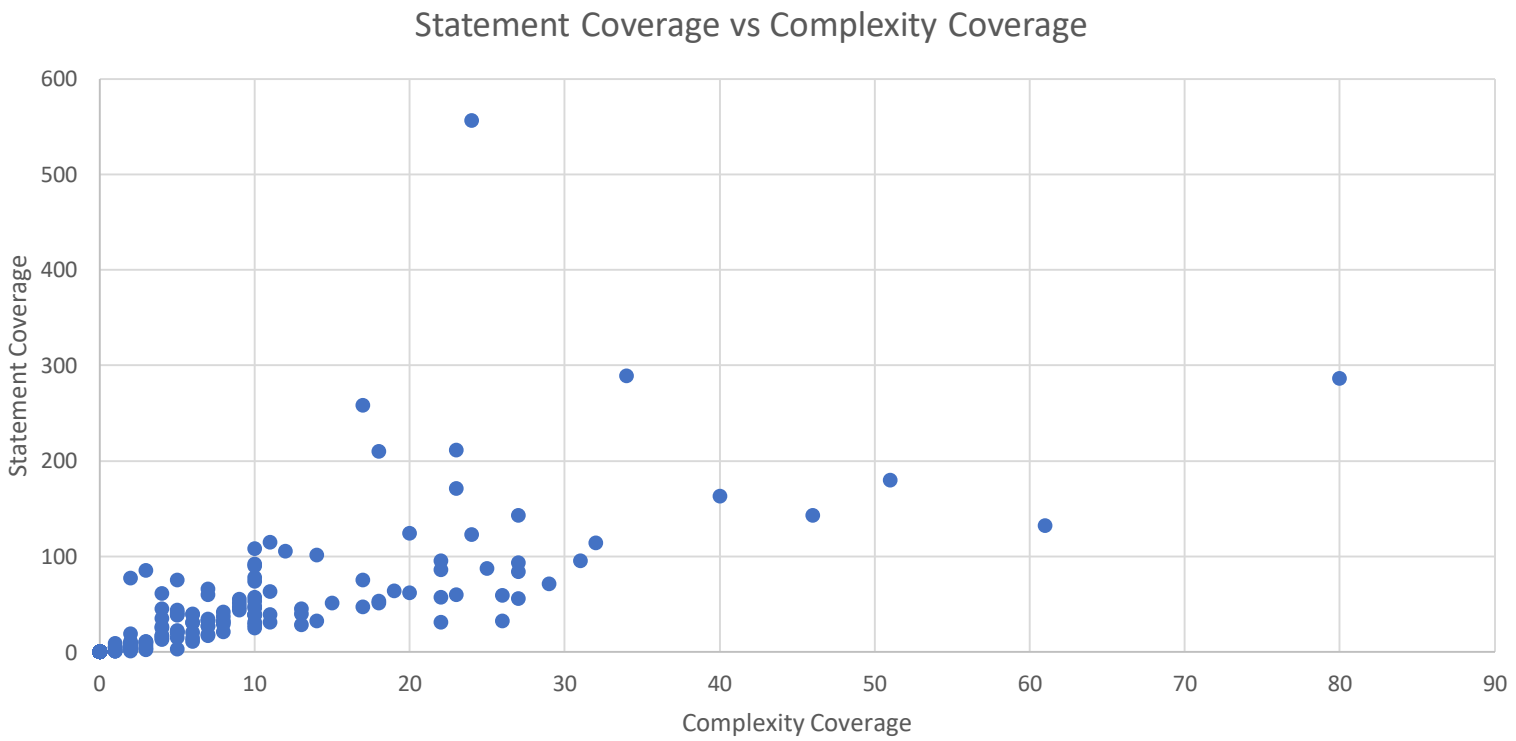
Version	Metric 5		Metric 6		
	DLOC	Effort Required (person hours)	LOC	No. of bugs	Software Defect Density
2.6	3046	108.46	107100	20	0.1867414
3.2	60259	680.59	121483	34	0.2798746
3.3	2687	104.87	125454	6	0.0478263
3.4	4368	121.68	129956	59	0.4539998
3.5	10982	187.82	141758	35	0.2468996
3.10	36569	443.69	154791	2	0.0129206

Results(Continue)
Apache Commons Net

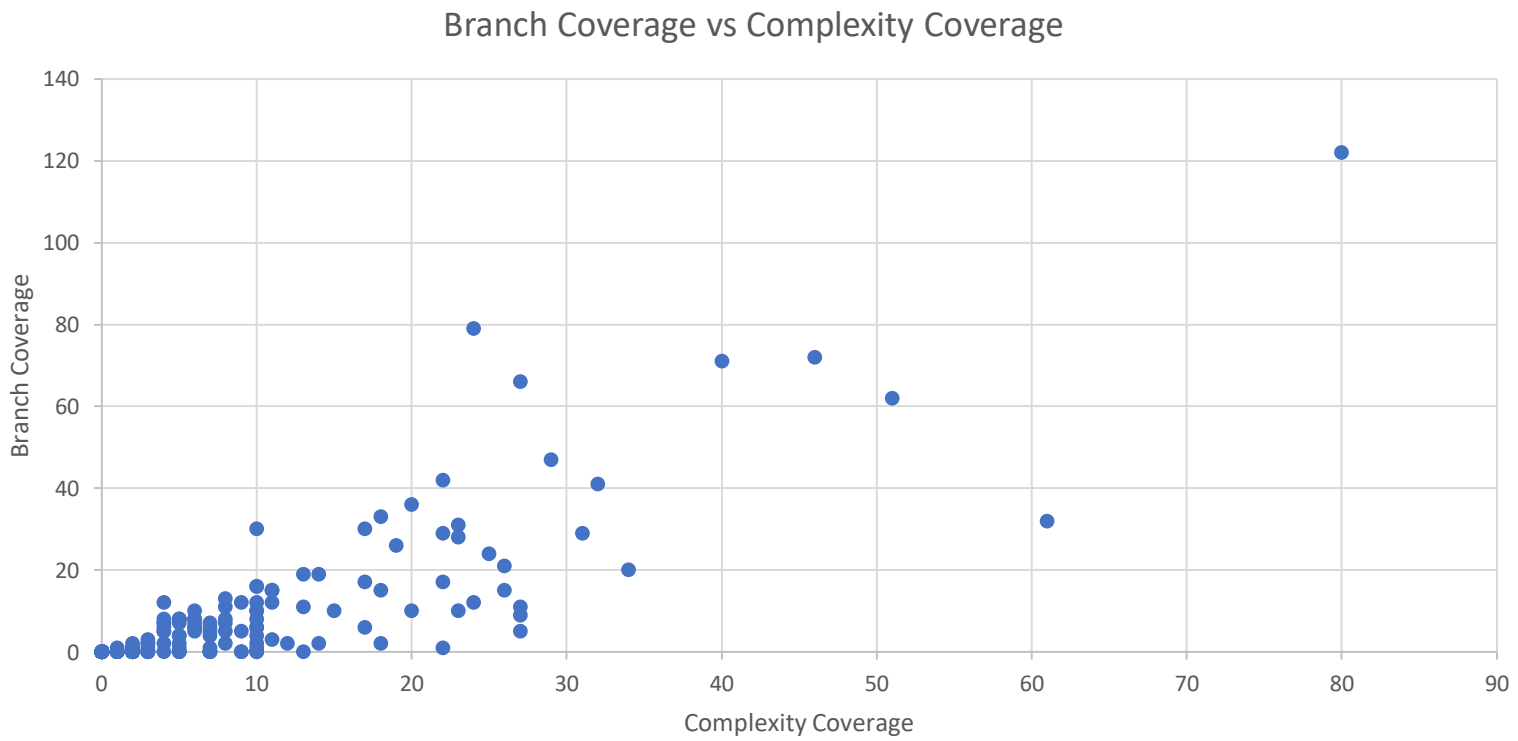
Version	Metric 5		Metric 6		
	DLOC	Effort Required (person hours)	LOC	No. of bugs	Software Defect Density
1.4.1	12	78.12	39741	84	2.1136861
2.2	17591	253.91	46940	29	0.61781
3.3	8017	158.17	57513	44	0.7650444
3.4	3563	113.63	61848	11	0.1778554
3.5	231	80.31	62066	14	0.2255663
3.6	1152	89.52	62865	27	0.4294918

Correlation

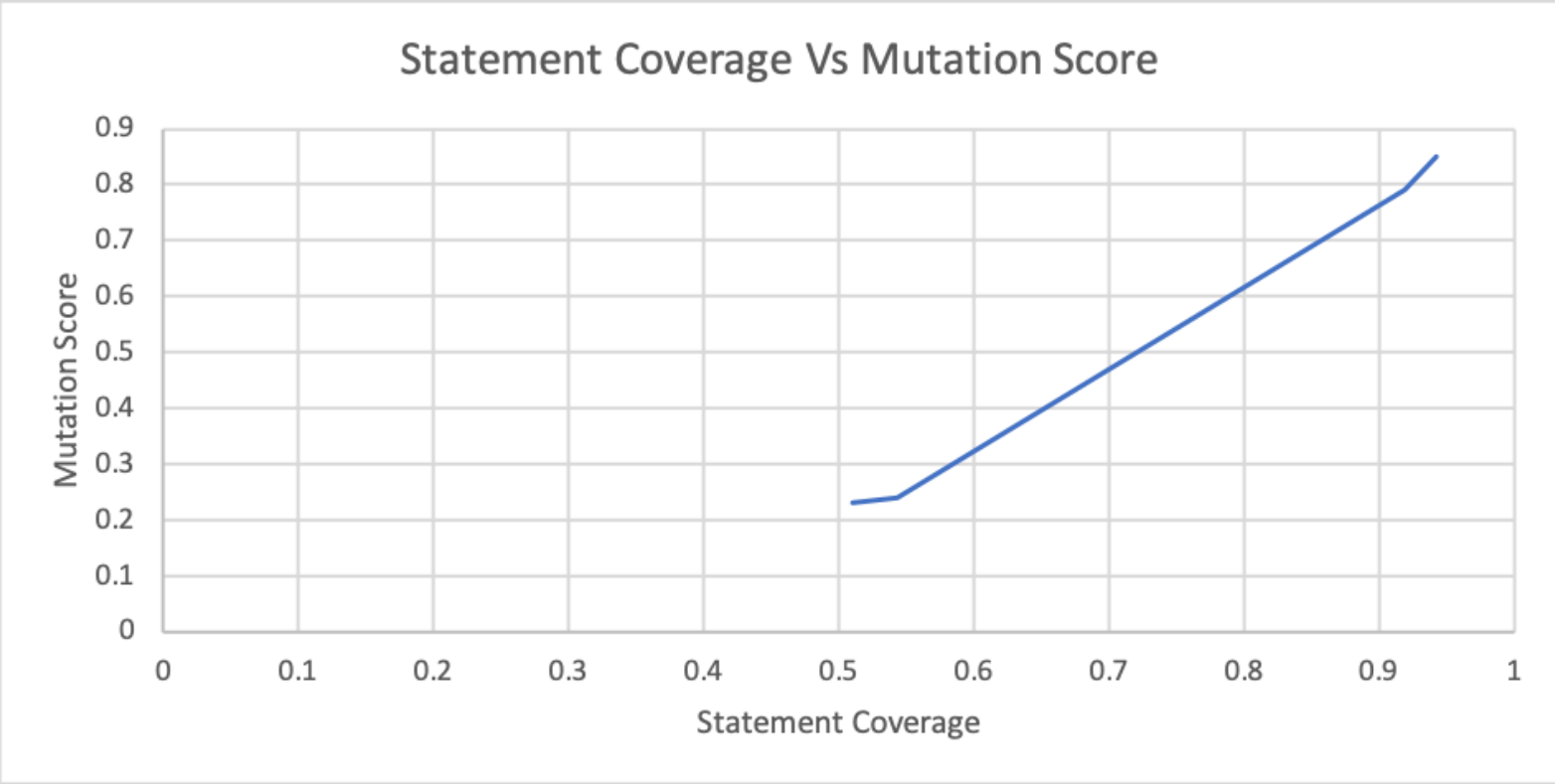
Metric 4 vs Metric 1 (Correlation : 0.7399)



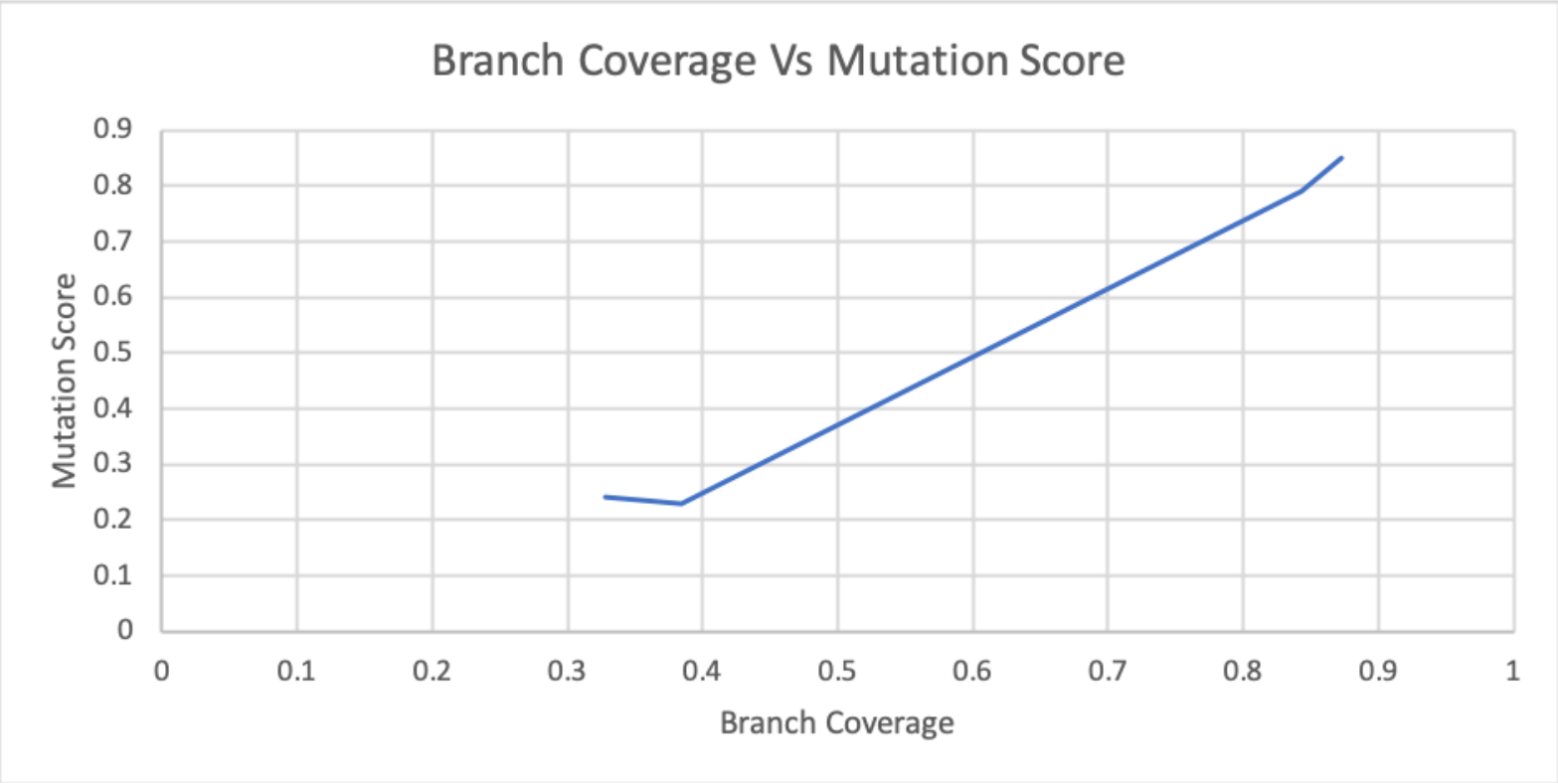
Metric 4 vs Metric 2 (Correlation : 0.8468)



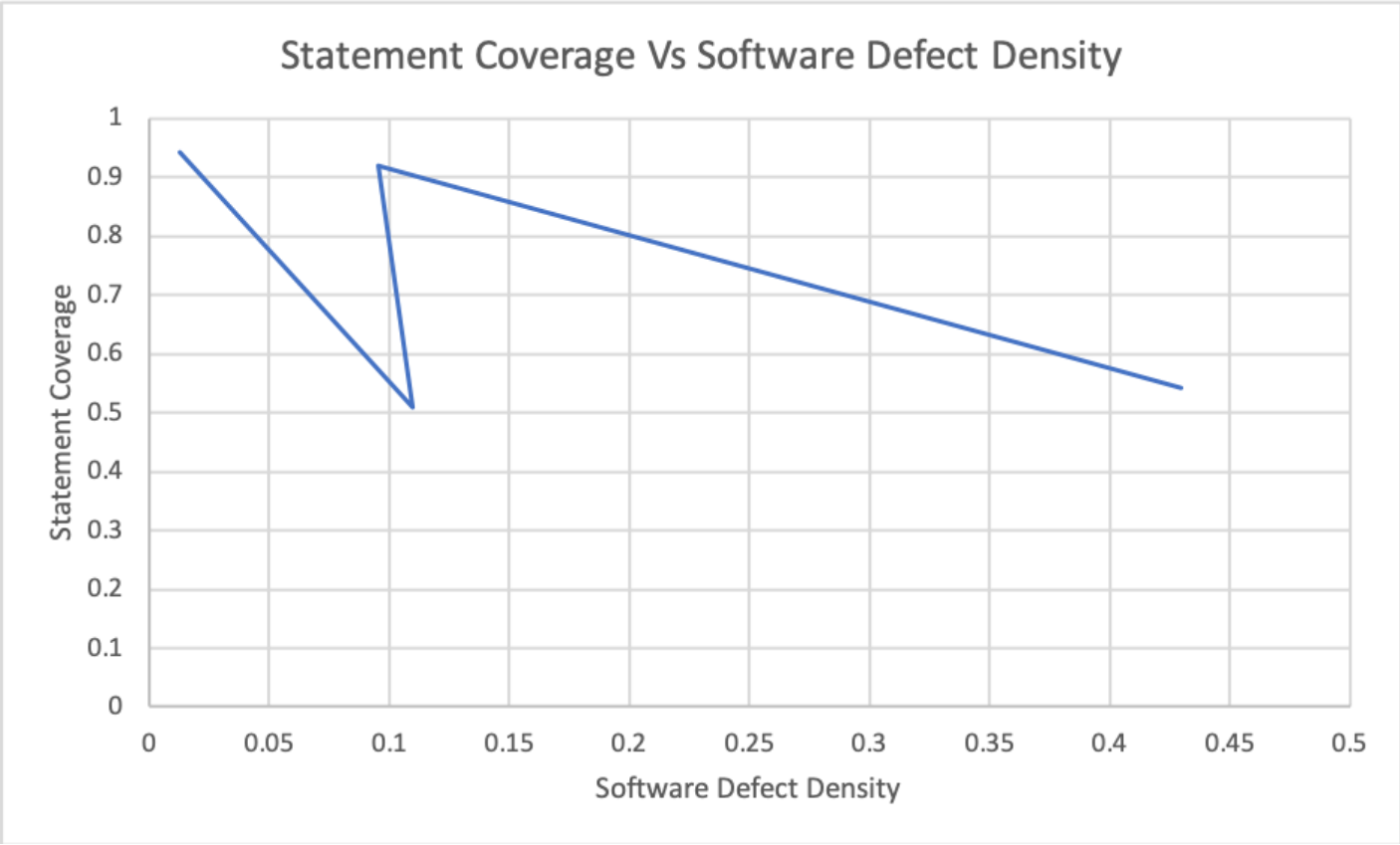
Metric 3 vs Metric 1 (Correlation : 0.9985)



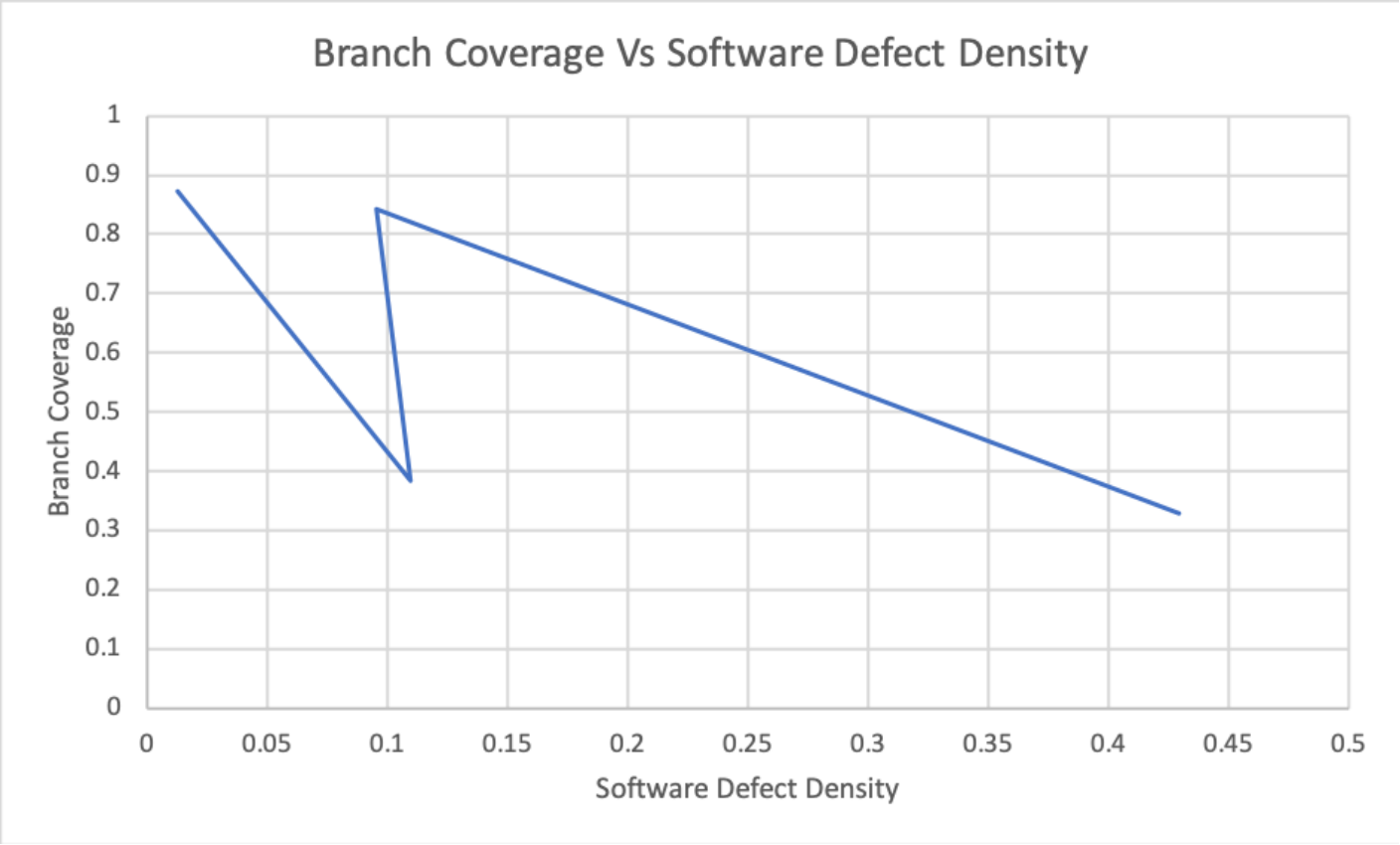
Metric 3 vs Metric 2 (Correlation : 0.9953)



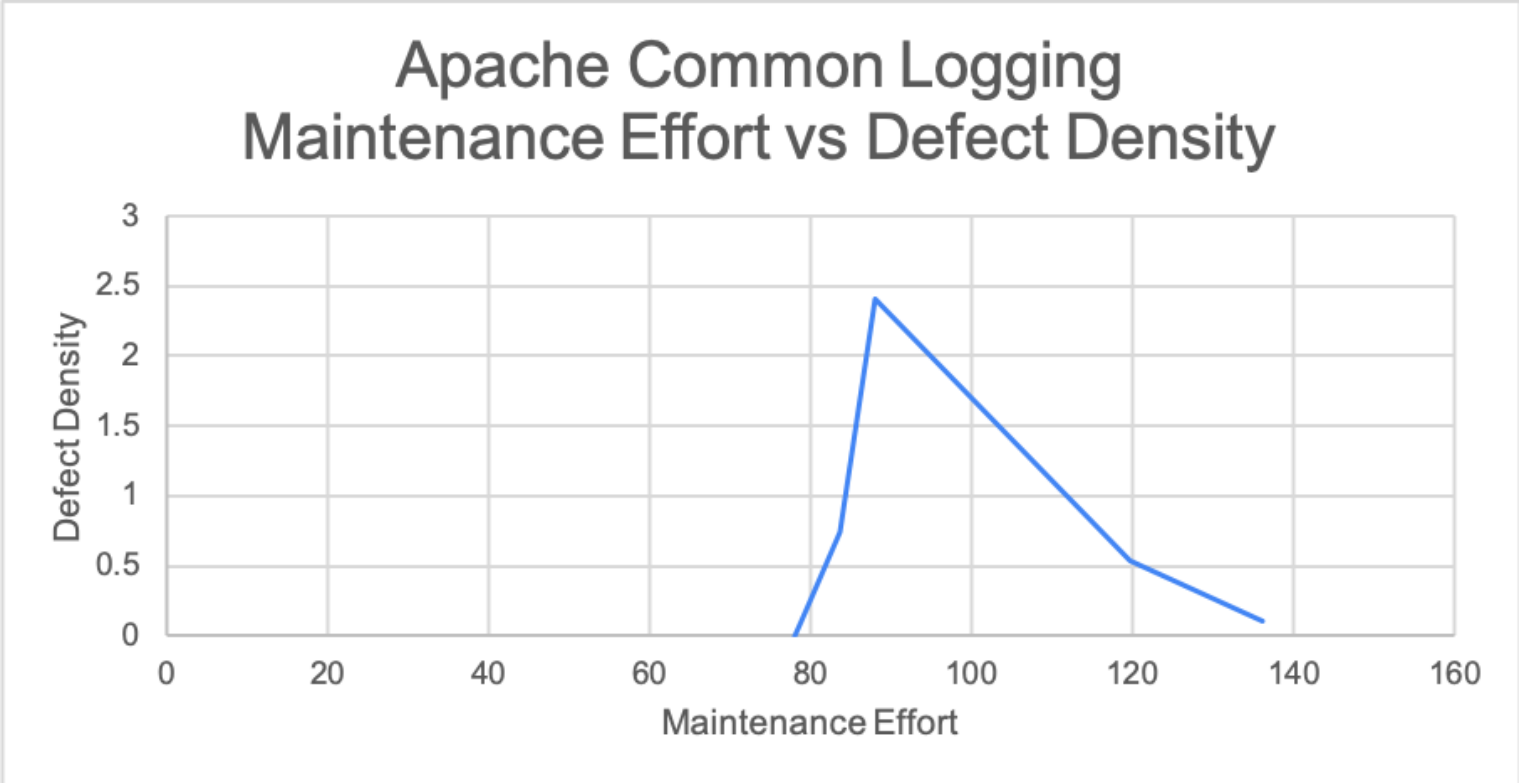
Metric 1 vs Metric 6 (Correlation : -0.6427)



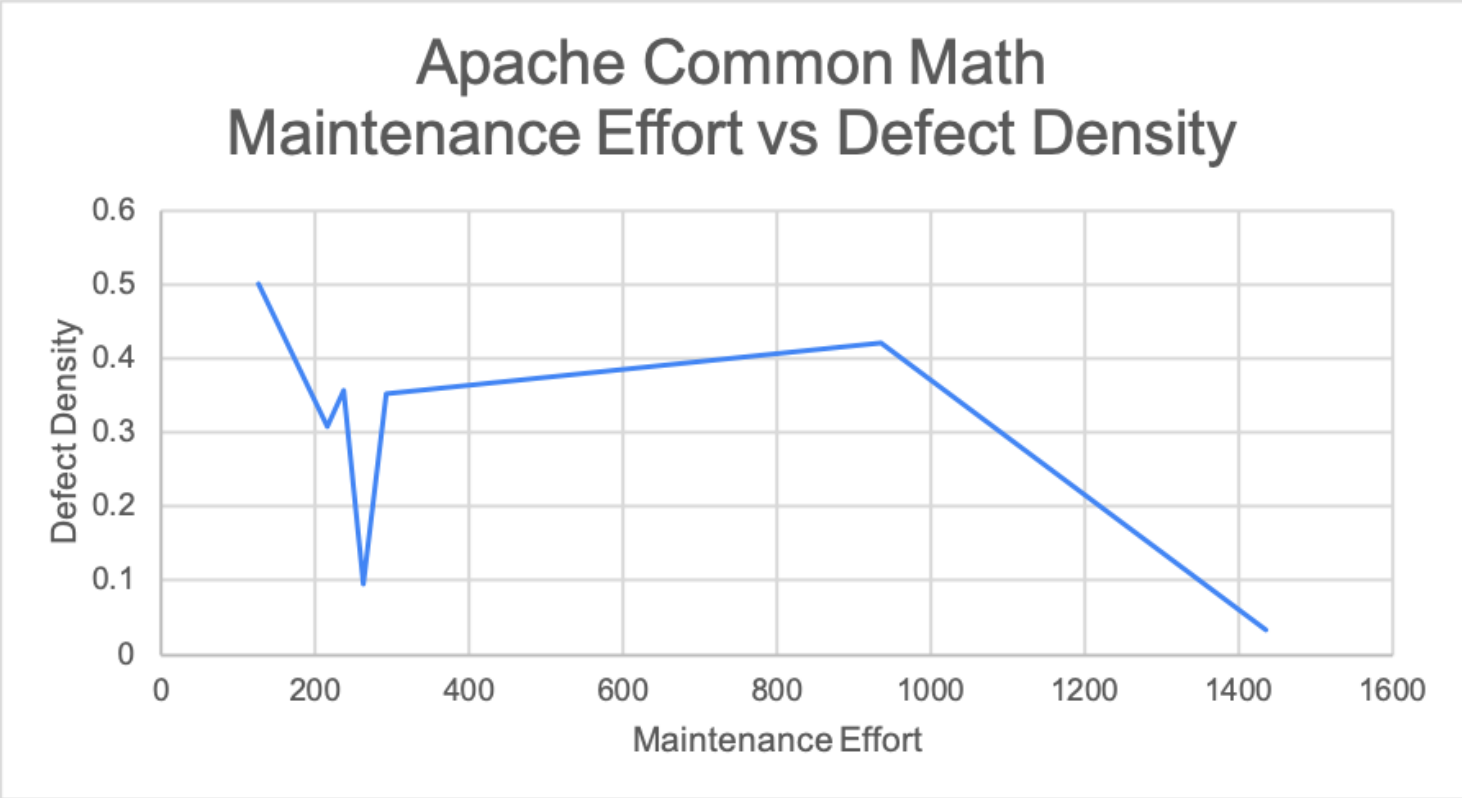
Metric 2 vs Metric 6 (Correlation : -0.7392)



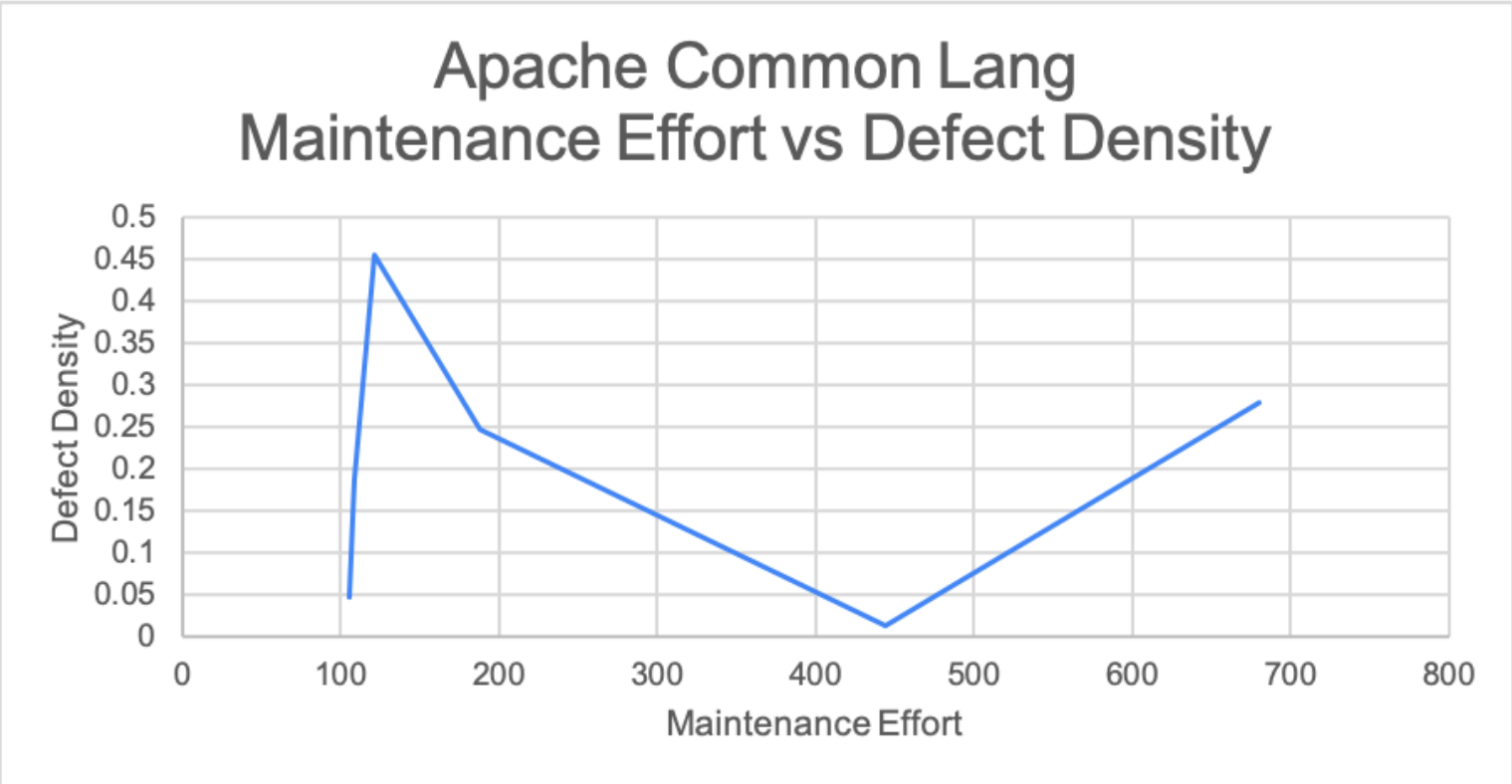
Metric 5 vs Metric 6
Apache Commons Logging (Correlation : -0.1669)



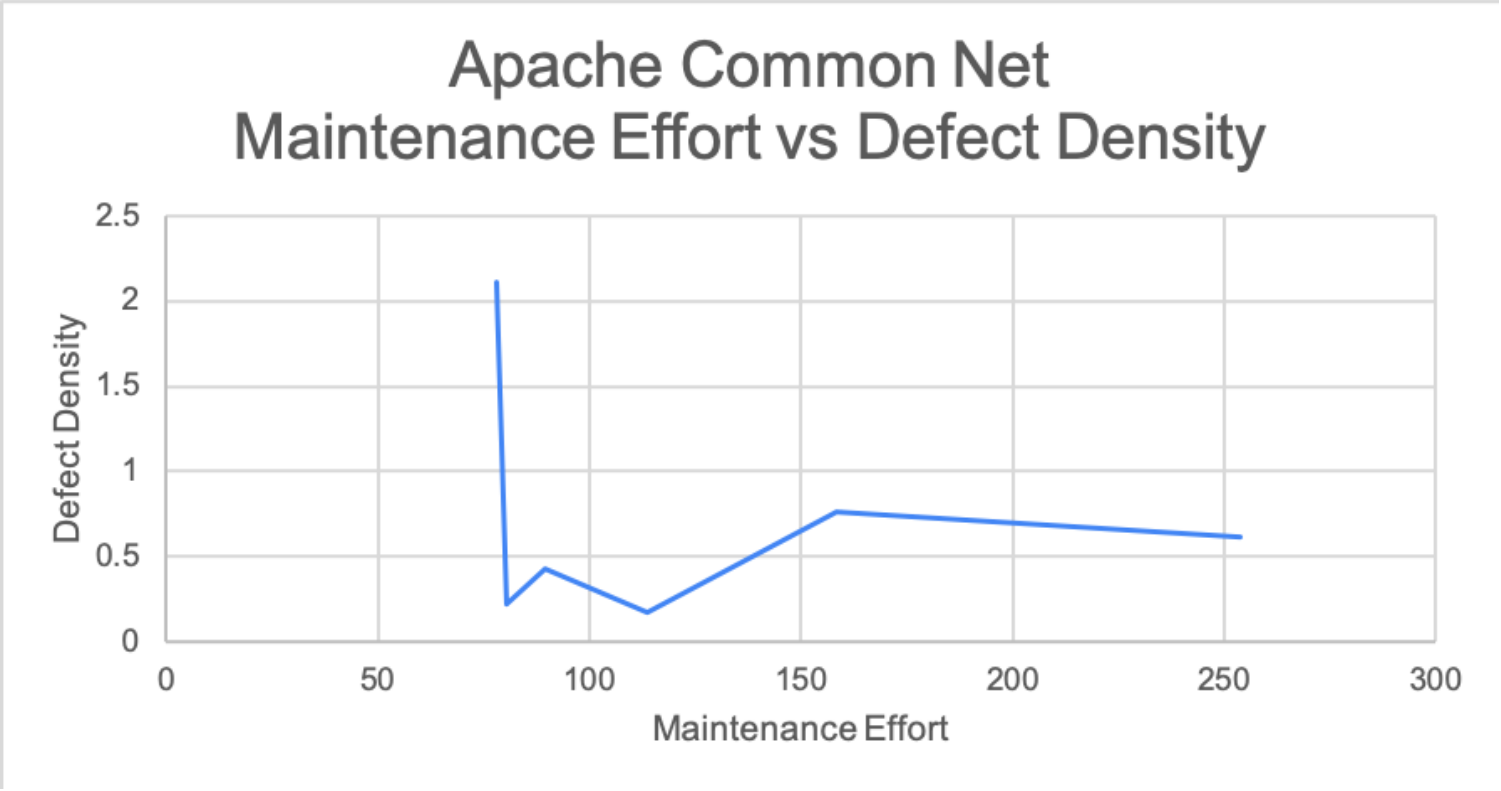
Metric 5 vs Metric 6(Continue)
Apache Commons Math (Correlation : -0.5)



Metric 5 vs Metric 6(Continue)
Apache Commons Lang (Correlation : -0.07)



Metric 5 vs Metric 6(Continue)
Apache Commons Net (Correlation : -0.1573)



References

- Cyclomatic complexity [Online]. Available: https://en.wikipedia.org/wiki/Cyclomatic_complexity [Accessed: 15-Feb-2020].
- Mccabe's Cyclomatic Complexity: Calculate with Flow Graph (Example) [Online]. Available: https://www.guru99.com/cyclomatic-complexity.html [Accessed: 15-Feb-2020].
- M. M. S. Sarwar, S. Shahzad and I. Ahmad, "Cyclomatic complexity: The nesting problem," *IEEE,* 2014.
- Mutation Testing in Software Testing: Mutant Score & Analysis Example [Online]. Available: https://www.guru99.com/mutation-testing.html [Accessed: 15-Feb-2020].
- ThinkSys, “34 Software Testing Metrics & KPIs: Complete Guide,” *ThinkSys Inc*, 11-Oct-2019. [Online]. Available: https://www.thinksys.com/qa-testing/software-testing-metrics-kpis/. [Accessed: 16-Feb-2020].
- J. Arnold, “64 Essential Software Quality Testing Metrics For Measuring Success,” *QASymphony*, 07-Jan-2019. [Online]. Available: https://www.qasymphony.com/blog/64-test-metrics/. [Accessed: 16-Feb-2020].