Andrew Downie
0786342

Question 4

The study I've chosen to discuss is available here:

https://research.google.com/pubs/pub41145.html

Context and Research Question

The field I am in is computer science. The study I've chosen was carried out by Google. The title of the paper is: "Does Bug Prediction Support Human Developers? Findings from a Google Case Study". The question this study tries find an answer to is in the title, and in my own words the question that this study focuses on is: can a bug predication algorithm help a developer focus on bug prone areas of a software project?

Description of the study

After a bug prediction algorithm was chosen in the first part of this study, Google then deployed and used this chosen algorithm to flag files based on how error prone the algorithm predicted each file was. When a developer worked on a file, they would be shown the associated flag that indicates how error prone the algorithm thinks that file is.

Three months after deploying this algorithm, Google investigated whether a change in developer behavior had occurred. To do this, they compared the mean time a file spent in code review for the 3 months after the algorithms deployment, versus the 3 months prior to the algorithms deployment. They also looked at the mean number of comments before/after the algorithms deployment on files marked to be error prone by the algorithm.

The Null and Alternative Hypothesis

The null hypothesis in this case is that the bug prediction algorithms had no effect on developers:

$H_0: \mu_1 = \mu_2$ (Adding bug prediction had no effect on the mean of the metrics used to gauge developer behavior)

Google does not explicitly provide the alternative hypothesis in the study, but it implicitly seems it would be:

$H_A: \mu_1 \neq \mu_2$ (Adding bug prediction had an effect on the mean of the metrics used to gauge developer behavior)

These hypothesis aren't provided by Google either, but to tie some hypothesis to a specific metric they used (in this example: mean time spent in code review), I would write the null and alternative hypothesis as:

$H_0: \mu_B = \mu_A$ (The mean time a file flagged as bug prone spent in code review before using the bug prediction algorithm *is the same* as the mean time a file flagged as bug prone spent in code review after using a bug prediction algorithm)

$H_A: \mu_B \neq \mu_A$ (The mean time a file marked as bug-prone spent in code review before using the bug prediction algorithm *is different than* the mean time a file

marked as bug-prone spent in code review after using a bug prediction algorithm)

The Results of This Study

The results of this hypothesis are summarized as: “We then looked at whether the means for each individual file increased or decreased, as displayed in Table III. Student’s t-tests show that neither change was significant, supporting a null hypothesis that the bug prediction deployment had no effect on developers.”

Metric	Increase	Decrease / No Change	Increase Improvement
Mean time to review	464	527	-63
Mean number of comments	524	467	57

TABLE III: A table showing the number of files that saw an increase or decrease in their mean time to review and their number of comments since deployment of the TWR.

This result means that files did not receive a meaningful change in amount of time spent during code review, nor did they have a significant change in number of comments after being flagged as more bug prone, than before they were flagged. Thus, the results of this study do not show that bug prediction can help or change developers behavior.