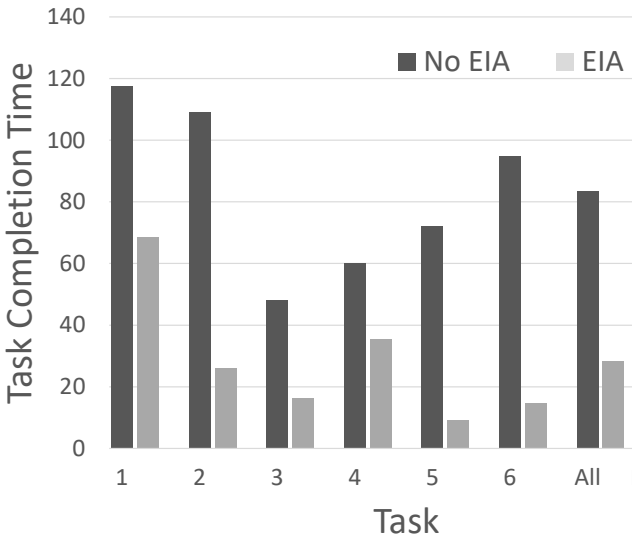

Figure 1: Task completion time to detect inconsistencies in code. EIA: Error Invariant Automata (author's approach).

Same Results: Explaining Inconsistent Code

Summary: The authors developed an approach to automatically detect and highlight inconsistent code via the construction of error invariant automata (EIA). In a small user study, they found that their approach helped participants to detect inconsistencies in code [**?**].

- Independent variable: Approach to display inconsistent code (2 levels, operationalized with visual assistance based on invariant automata, and without visual assistance)
- Tasks: 6 tasks to detect inconsistencies in different code snippets
- Dependent variable: Task completion time [metric scale]
- Null hypotheses:

  No test was conducted because of small sample size (12 participants), so no hypotheses were stated.
- Results: Speed up by a factor of three for the version that created visual assistance based on error invariant automata.

We agree with the authors that with such a small sample size, a statistical test does not make much sense. However, when aggregating the response times for all tasks, we can increase statistical power, so that we can conduct a significance test here. Since the data are not normally distributed, we use a paired Wilcoxon test, showing that the speed up in favor of the visual assistance based on error invariant automata is significant ($W = 75$, $p = 0.002$). In Figure 1, we show the task-wise response times, as well as the aggregated response times over all tasks (right in Figure 1). Thus, in this case, aggregation did not lead to different results, but we can state more determined that using the error invariant automata to visualize inconsistencies in code does significantly help participants to detect inconsistencies.