On the Relationship Between Code Verifiability and Understandability

Kobi Feldman\textsuperscript{b}, Martin Kellogg\textsuperscript{a}, Oscar Chaparro\textsuperscript{b}

\textsuperscript{a}New Jersey Institute of Technology \hspace{1cm} \textsuperscript{b}College of William & Mary
Common Wisdom

easier to verify -> easier to understand
Common Wisdom

“rewrite your code to be simpler for the checker to analyze; easier to understand

- Checker Framework manual
Common Wisdom

“rewrite your code to be simpler for the checker to analyze; this is likely to make it easier for people to understand, too”
- Checker Framework manual

easier to understand
“rewrite your code to be simpler for the checker to analyze; this is likely to make it easier for people to understand, too”
- Checker Framework manual

“success in checking the consistency of the specifications and the code will depend on... the complexity and style in which the code and specifications are written”
- OpenJML manual
Common Wisdom

easier to verify -> easier to understand
Common Wisdom

easier to verify -> easier to understand

But how do we know that this is true?
Common Wisdom

easier to verify -> easier to understand

But how do we know that this is true?

Our goal: fill this gap in the literature with an empirical study
Does it matter?

● An empirical study’s results must be **actionable**
Does it matter?

- An empirical study’s results must be actionable
- So, what are the implications if our hypothesis is correct?
Does it matter?

● An empirical study’s results must be actionable
● So, what are the implications if our hypothesis is correct?
● Our hypothesis:
  ○ “There is a correlation between code that is hard to verify and code that is hard for humans to understand.”
Implications

“There is a correlation between code that is hard to verify and code that is hard for humans to understand.”
Implications

- For the **builders** of verification tools:

  "There is a *correlation* between code that is *hard to verify* and code that is *hard for humans to understand.*"
Implications

- For the **builders** of verification tools:
  - we are giving good advice to our users (yay!)

“Our implication is that there is a correlation between code that is hard to verify and code that is hard for humans to understand.”
Implications

- For the **builders** of verification tools:
  - we are giving good advice to our users (yay!)
  - error messages should suggest semantically-equivalent code that would verify (new research direction!)

“There is a correlation between code that is **hard to verify** and code that is **hard for humans to understand**.”
Implications

● For the **builders** of verification tools:
  ○ we are giving good advice to our users (yay!)
  ○ error messages should suggest semantically-equivalent code that would verify (new research direction!)
● For the **users** of verification tools:

“There is a correlation between code that is **hard to verify** and code that is **hard for humans to understand**.”
Implications

● **For the builders** of verification tools:
  ○ we are giving good advice to our users (yay!)
  ○ error messages should suggest semantically-equivalent code that would verify (new research direction!)

● **For the users** of verification tools:
  ○ refactor to avoid warnings

“There is a correlation between code that is hard to verify and code that is hard for humans to understand.”
Implications

- For the **builders** of verification tools:
  ○ we are giving good advice to our users (yay!)
  ○ error messages should suggest semantically-equivalent code that would verify (new research direction!)

- For the **users** of verification tools:
  ○ refactor to avoid warnings

“*There is a correlation between code that is hard to verify and code that is hard for humans to understand.*”

**Auxiliary benefit** of verification: points to hard-to-understand code
Implications

- For the **builders** of verification tools:
  - we are giving good advice to our users (yay!)
  - error messages should suggest semantically-equivalent code that would verify (new research direction!)

- For the **users** of verification tools:
  - refactor to avoid warnings

- For **code understanding researchers**:

  “There is a *correlation* between code that is *hard to verify* and code that is *hard for humans to understand*.”
Implications

- For the **builders** of verification tools:
  - we are giving good advice to our users (yay!)
  - error messages should suggest semantically-equivalent code that would verify (new research direction!)

- For the **users** of verification tools:
  - refactor to avoid warnings

- For **code understanding researchers**:
  - there is a semantic component to human code understanding

“There is a correlation between code that is hard to verify and code that is hard for humans to understand.”
Implications

● For the **builders** of verification tools:
  ○ we are giving good advice to our users (yay!)
  ○ error messages should suggest semantically-equivalent code that would verify (new research direction!)

● For the **users** of verification tools:
  ○ refactor to avoid warnings

● For **code understanding researchers**:
  ○ there is a semantic component to human code understanding
  ○ explains ineffectiveness of traditional, syntactic metrics like cyclomatic complexity

“There is a correlation between code that is hard to verify and code that is hard for humans to understand.”
There is a correlation between code that is hard to verify and code that is hard for humans to understand.

Problem: neither of these are easy to measure directly
Empirical study design

“There is a correlation between code that is hard to verify and code that is hard for humans to understand.”

- **Problem**: neither of these are easy to measure directly
  - must use proxies
Proxy for verifiability
Proxy for verifiability

Warnings on unannotated, correct code snippets
Proxy for verifiability

**Warnings** on **unannotated, correct** code snippets

- “unannotated” = “no specifications”
Proxy for verifiability

**Warnings** on **unannotated**, **correct** code snippets

- “unannotated” = “no specifications”
  - but still trying to prove e.g., absence of buffer overflows
Proxy for verifiability

Warnings on unannotated, correct code snippets

- “unannotated” = “no specifications”
  - but still trying to prove e.g., absence of buffer overflows
- “correct” so that no warnings correspond to real bugs
Proxy for verifiability

Warnings on unannotated, correct code snippets

● “unannotated” = “no specifications”
  ○ but still trying to prove e.g., absence of buffer overflows
● “correct” so that no warnings correspond to real bugs
  ○ that is, all warnings are false positives
Choosing verifiers

- We selected four “verifiers”:
  - Checker framework
  - Infer
  - OpenJML
  - Java Typestate Checker
Choosing verifiers

- We selected four “verifiers”:

  All tools have **sound cores**: internally, they try to construct a proof (= “do verification”).
Proxy for understandability
Proxy for understandability

Metrics for understandability from prior work
Proxy for understandability

Metrics for understandability from prior work

● this is a pragmatic decision: don’t run another human study!
Proxy for understandability

Metrics for understandability from *prior work*

- this is a *pragmatic* decision: don’t run another human study!
  - but studies in the literature don’t use the same set of metrics
Prior studies

- we used 6 prior studies
Prior studies: descriptive stats

Table 1: Datasets (DSs) of code snippets and understandability measurements/metrics used in our study. The metrics types are “C” for correctness, “R” for ratings, “T” for time, and “P” for physiological.

<table>
<thead>
<tr>
<th>DS</th>
<th>Snippets</th>
<th>NCLOC</th>
<th>Participants</th>
<th>Understandability Task</th>
<th>Understandability Metrics</th>
<th>Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23 CS algorithms</td>
<td>6 - 20</td>
<td>41 students</td>
<td>Determine prog. output</td>
<td>C: correct_output_rating (3-level correctness score for program output) R: output_difficulty (5-level difficulty score for determining program output) T: time_to_give_output (seconds to read program and answer a question)</td>
<td>2,829</td>
</tr>
<tr>
<td>2</td>
<td>12 CS algorithms</td>
<td>7 - 15</td>
<td>16 students</td>
<td>Determine prog. output</td>
<td>P: brain_deact_31ant (deactivation of brain area BA31ant) P: brain_deact_31post (deactivation of brain area BA31post) P: brain_deact_32 (deactivation of brain area BA32) T: time_to_understand (seconds to understand program within 60 secs.)</td>
<td>228</td>
</tr>
<tr>
<td>3</td>
<td>100 OSS methods</td>
<td>5 - 13</td>
<td>121 students</td>
<td>Rate prog. readability</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td>6</td>
<td>50 OSS methods</td>
<td>18 - 75</td>
<td>50 students and 13 developers</td>
<td>Rate underst./answer Qs</td>
<td>C: correctness (0/1 program understandability score) R: readability_level_ba (5-level avg. score for readability before code completion) T: time_to_read_complete (avg. seconds to read and complete code)</td>
<td>1,197</td>
</tr>
<tr>
<td>9</td>
<td>10 OSS methods</td>
<td>10 - 34</td>
<td>104 students</td>
<td>Rate read./complete prog.</td>
<td>C: gap_accuracy (0/1 accuracy score for filling in program blanks) R: readability_level_before (5-level score for readability before code completion) T: time_to_read_complete (avg. seconds to read and complete code)</td>
<td>2,600</td>
</tr>
<tr>
<td>F</td>
<td>16 CS algorithms</td>
<td>7 - 19</td>
<td>19 students</td>
<td>Determine prog. output</td>
<td>P: brain_deact_31 (deactivation of brain area BA31) P: brain_deact_32 (deactivation of brain area BA32) R: complexity_level (score for program complexity) C: perc_correct_output (percentage of subjects who correctly gave program output) T: time_to_understand (seconds to understand program within 60 seconds)</td>
<td>631</td>
</tr>
</tbody>
</table>
Prior studies: descriptive stats

Table 1: Datasets (DSs) of code snippets and understandability measurements/metrics used in our study. The metrics types are “C” for correctness, “R” for ratings, “T” for time, and “P” for physiological.

<table>
<thead>
<tr>
<th>DS</th>
<th>Snippets/Study</th>
<th>Snippet/Study Type</th>
<th>Understandability Task</th>
<th>Understandability Metrics</th>
<th>Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 [81]</td>
<td>23 CS algorithms</td>
<td>7 - 15</td>
<td>Determine prog. output</td>
<td>C: correct_output_rating (3-level correctness score for program output) R: output_difficulty (5-level difficulty score for determining program output) T: time_to_give_output (seconds to read program and answer a question)</td>
<td>2,829</td>
</tr>
<tr>
<td>2 [70]</td>
<td>12 CS algorithms</td>
<td>7 - 15</td>
<td>Determine prog. output</td>
<td>P: brain_deact_31ant (deactivation of brain area BA31ant) P: brain_deact_31post (deactivation of brain area BA31post) P: brain_deact_32 (deactivation of brain area BA32) T: time_to_understand (seconds to understand program within 60 secs.)</td>
<td>228</td>
</tr>
<tr>
<td>3 [16]</td>
<td>100 OSS methods</td>
<td>5 - 13</td>
<td>Rate prog. readability</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td>4 [77]</td>
<td>50 OSS methods</td>
<td>7 - 15</td>
<td>Rate underst./answer Qs</td>
<td>C: correct_verif_questions (% of correct answers to verification questions) T: time_to_understand (seconds to understand program)</td>
<td>1,197</td>
</tr>
<tr>
<td>5 [14]</td>
<td>10 OSS methods</td>
<td>10 - 34</td>
<td>Rate read./complete prog</td>
<td>C: gap_accuracy (0/1 accuracy score for filling in program blanks) R: readability_level_before (5-level score for readability before code completion) T: time_to_read_complete (avg. seconds to rate readability and complete code)</td>
<td>2,600</td>
</tr>
<tr>
<td>F [68]</td>
<td>16 CS algorithms</td>
<td>7 - 19</td>
<td>Determine prog. output</td>
<td>P: brain_deact_31 (deactivation of brain area BA31) P: brain_deact_32 (deactivation of brain area BA32) R: complexity_level (score for program complexity) C: perc_correct_output (% of subjects who correctly gave program output) T: time_to_understand (seconds to understand program within 60 seconds)</td>
<td>631</td>
</tr>
</tbody>
</table>
Prior studies: descriptive stats

Table 1: Datasets (DSs) of code snippets and understandability measurements/metrics used in our study. The metrics types are “C” for correctness, “R” for ratings, “T” for time, and “P” for physiological.

<table>
<thead>
<tr>
<th>DS</th>
<th>Snippets</th>
<th>NCLOC</th>
<th>Participants</th>
<th>Understandability Task</th>
<th>Understandability Metrics</th>
<th>Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS algorithms 6</td>
<td>6</td>
<td></td>
<td>Rate prog. readability</td>
<td>MEAS1 (3-level correctness score for program output)</td>
<td>2,829</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MEAS2 (5-level difficulty score for determining program output)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MEAS3 (seconds to read program and answer a question)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MEAS4 (deactivation of brain area BA31)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MEAS5 (deactivation of brain area BA32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program within 60 secs.)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OSS methods 7</td>
<td>7</td>
<td></td>
<td>Rate underst./answer Qs</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: binary_understandability (0/1 program understandability score)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: correct_verif_questions (% of correct answers to verification questions)</td>
<td>1,197</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OSS methods 110</td>
<td>5-13</td>
<td>121 students</td>
<td>Rate prog. readability</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: binary_understandability (0/1 program understandability score)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: gap_accuracy (0/1 accuracy score for filling in program blanks)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_read_complete (avg. seconds to rate readability and complete code)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OSS methods 10</td>
<td>10-34</td>
<td>104 students</td>
<td>Rate read./complete prog.</td>
<td>R: readability_level_traffic (5-level avg. score for readability w/ a code completion)</td>
<td>2,600</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: readability_level_before (5-level score for readability before code completion)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_read_complete (avg. seconds to rate readability and complete code)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>OSS methods 14</td>
<td>7-19</td>
<td>19 students</td>
<td>Determine prog. output</td>
<td>R: complexity_level (score for program complexity)</td>
<td>631</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: perc_correct_output (% of subjects who correctly gave program output)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program within 60 seconds)</td>
<td></td>
</tr>
</tbody>
</table>
Prior studies: descriptive stats

Table 1: Datasets (DSs) of code snippets and understandability measurements/metrics used in our study. The metrics types are “C” for correctness, “R” for ratings, “T” for time, and “P” for physiological.

<table>
<thead>
<tr>
<th>DS</th>
<th>Snippets</th>
<th>NCLOC</th>
<th>Par</th>
<th>Understandability Metrics</th>
<th>Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 [81]</td>
<td>23 CS algorithms</td>
<td>6 - 20</td>
<td>41</td>
<td>P: brain_deact_31ant (deactivation of brain area BA31ant)</td>
<td>2,829</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P: brain_deact_31post (deactivation of brain area BA31post)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>P: brain_deact_32 (deactivation of brain area BA32)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program within 60 secs.)</td>
<td></td>
</tr>
<tr>
<td>2 [70]</td>
<td>12 CS algorithms</td>
<td>7 - 15</td>
<td>16 students</td>
<td>Determine prog. output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: correct_output (3-level correctness score for program output)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: correct_output_dif (5-level difficulty score for determining program output)</td>
<td></td>
</tr>
<tr>
<td>3 [16]</td>
<td>100 OSS methods</td>
<td>5 - 13</td>
<td>121 students</td>
<td>Rate prog. readability</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: binary_understandability (0/1 program understandability score)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: correct_verify_questions (% of correct answers to verification questions)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program)</td>
<td></td>
</tr>
<tr>
<td>6 [77]</td>
<td>50 OSS methods</td>
<td>18 - 75</td>
<td>50 students and 13 developers</td>
<td>Rate underst./answer Qs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: gap_accuracy (0/1 accuracy score for filling in program blanks)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_read_complete (avg. seconds to rate readability and complete code)</td>
<td></td>
</tr>
<tr>
<td>9 [14]</td>
<td>10 OSS methods</td>
<td>10 - 34</td>
<td>104 students</td>
<td>Rate read./complete prog.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: complexity (score for program complexity)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program within 60 secs)</td>
<td></td>
</tr>
<tr>
<td>F [68]</td>
<td>16 CS algorithms</td>
<td>7 - 19</td>
<td>19 students</td>
<td>Determine prog. output</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R: complexity_level (score for program complexity)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C: perc_correct_output (% of subjects who correctly gave program output)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>T: time_to_understand (seconds to understand program within 60 secs)</td>
<td></td>
</tr>
</tbody>
</table>
Prior studies: descriptive stats

<table>
<thead>
<tr>
<th>DS</th>
<th>Snippets</th>
<th>NCLOC</th>
<th>Participants</th>
<th>Understandability Task</th>
<th>Understandability Metrics</th>
<th>Meas.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[81]</td>
<td>6 - 20</td>
<td>41 students</td>
<td>Determine program output</td>
<td>C: correct_output_rating (3-level correctness score for program output)</td>
<td>2,829</td>
</tr>
<tr>
<td>2</td>
<td>[70]</td>
<td>7 - 15</td>
<td>16 students</td>
<td>Determine program output</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>12,100</td>
</tr>
<tr>
<td>3</td>
<td>[16]</td>
<td>5 - 13</td>
<td>121 students</td>
<td>Rate prog. readability</td>
<td>R: binary_understandability (0/1 program understandability score)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>[77]</td>
<td>18 - 75</td>
<td>50 students and 13 developers</td>
<td>Rate underst./answer Qs</td>
<td>C: correct_verif_questions (% of correct answers to verification questions)</td>
<td>1,197</td>
</tr>
<tr>
<td>9</td>
<td>[14]</td>
<td>10 - 34</td>
<td>104 students</td>
<td>Rate read./complete prog.</td>
<td>R: readability_level (5-level score for readability/ease to understand)</td>
<td>2,600</td>
</tr>
<tr>
<td>F</td>
<td>[68]</td>
<td>7 - 19</td>
<td>19 students</td>
<td>Determine prog. output</td>
<td>R: complexity_level (score for program complexity)</td>
<td>631</td>
</tr>
</tbody>
</table>

almost all students; # of participants varies
Prior studies: metrics

- we used 6 prior studies
- 20 metrics:
Prior studies: metrics

- we used 6 prior studies
- 20 metrics:
  - 4 correctness (e.g., “% answering a question correctly”)
Prior studies: metrics

- we used 6 prior studies
- 20 metrics:
  - 4 correctness (e.g., “% answering a question correctly”)
  - 6 rating (e.g., “readability level”)
Prior studies: metrics

- we used 6 prior studies
- 20 metrics:
  - 4 correctness (e.g., “% answering a question correctly”)
  - 6 rating (e.g., “readability level”)
  - 5 time (e.g., “time to read program and answer a question”)

Prior studies: metrics
Prior studies: metrics

- we used 6 prior studies
- 20 metrics:
  - 4 correctness (e.g., “% answering a question correctly”)
  - 6 rating (e.g., “readability level”)
  - 5 time (e.g., “time to read program and answer a question”)
  - 5 physiological (e.g., brain area deactivation via fMRI)
Meta-analysis

- it is not obvious how to combine these metrics
Meta-analysis

- it is **not obvious** how to combine these metrics
- **tempting but wrong** idea: measure correlation for each metric independently, then count correlations
Meta-analysis

- it is **not obvious** how to combine these metrics
- **tempting but wrong** idea: measure correlation for each metric independently, then count correlations
  - a statistical error! (“**vote counting**”):
Meta-analysis

● it is not obvious how to combine these metrics
● tempting but wrong idea: measure correlation for each metric independently, then count correlations
  ○ a statistical error! (“vote counting”):
    ■ overweights studies with more metrics
    ■ doesn’t take into account effect sizes
Meta-analysis

- It is **not obvious** how to combine these metrics
- **Tempting but wrong** idea: measure correlation for each metric independently, then count correlations
  - a statistical error! ("vote counting"):  
    - overweights studies with more metrics  
    - doesn’t take into account effect sizes
- Instead, use **random-effects meta-analysis**
Meta-analysis

- it is **not obvious** how to combine these metrics
- **tempting but wrong** idea: measure correlation for each metric independently, then count correlations
  - a statistical error! (“**vote counting**”):
    - overweights studies with more metrics
    - doesn’t take into account effect sizes
- instead, use **random-effects meta-analysis**
  - technique for combining medical studies on different populations and proxies
Unit-of-analysis problem

- meta-analysis combines independent correlations into a single, aggregate correlation
Unit-of-analysis problem

- meta-analysis combines independent correlations into a single, aggregate correlation
  - however, our correlations are not independent!
Unit-of-analysis problem

- meta-analysis combines independent correlations into a single, aggregate correlation
  - however, our correlations are not independent!
    - each study has same subjects, same snippets
Unit-of-analysis problem

- meta-analysis **combines** independent correlations into a single, aggregate correlation
  - however, our correlations are **not independent**!
    - each study has **same** subjects, **same** snippets
- in meta-analysis, this is the “**unit-of-analysis problem**”
Unit-of-analysis problem

- meta-analysis **combines** independent correlations into a single, aggregate correlation
  - however, our correlations are **not independent**!
    - each study has **same** subjects, **same** snippets
- in meta-analysis, this is the “**unit-of-analysis problem**”
  - an **open problem** (!) in statistical methods research
Unit-of-analysis problem

- meta-analysis *combines* independent correlations into a single, aggregate correlation
  - however, our correlations are *not independent*!
    - each study has *same* subjects, *same* snippets
- in meta-analysis, this is the “*unit-of-analysis problem*”
  - an *open problem* (!) in statistical methods research
    - we tried some cutting-edge statistical techniques, but their (strong) assumptions weren’t satisfied
Unit-of-analysis problem

- meta-analysis combines independent correlations into a single, aggregate correlation
  - however, our correlations are not independent!
    - each study has same subjects, same snippets
- in meta-analysis, this is the “unit-of-analysis problem”
  - an open problem (!) in statistical methods research
    - we tried some cutting-edge statistical techniques, but their (strong) assumptions weren’t satisfied
    - instead, use brute force: combine all metrics for each study into one correlation
Unit-of-analysis problem

- meta-analysis **combines** independent correlations into a single, aggregate correlation
  - however, our correlations are **not independent**!
    - each study has the **same** subjects, snippets
- in meta-analysis, this is the "**unit-of-analysis problem**"
  - an **open problem** (!) in statistical methods research
    - we tried some cutting-edge statistical techniques, but their (strong) assumptions weren’t satisfied
    - instead, use **brute force**: combine all metrics for each study into one correlation

Brute force is **safe**, but **throws away** the benefit of multiple metrics per study
Results
Results: overall
Results: overall

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Snippets</th>
<th>Weights</th>
<th>Estimate [95% CI]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>14.84%</td>
<td>-0.52 [-0.77, -0.14]</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>7.88%</td>
<td>-0.43 [-0.80, 0.20]</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>34.87%</td>
<td>-0.22 [-0.40, -0.03]</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>25.40%</td>
<td>0.03 [-0.25, 0.30]</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6.33%</td>
<td>0.04 [-0.60, 0.66]</td>
</tr>
<tr>
<td>f</td>
<td>16</td>
<td>10.68%</td>
<td>-0.36 [-0.73, 0.16]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00%</td>
<td>-0.23 [-0.46, 0.03]</td>
</tr>
</tbody>
</table>

Test for Heterogeneity: $Q = 6.80$, df = 5, $p = 0.24$

Pearson's $r$ (negative correlation supports our hypothesis)
Results: overall

overall correlation of $r=0.23$ (small effect size)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Snippets</th>
<th>Weights</th>
<th>Estimate [95% CI]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>14.84%</td>
<td>$-0.52 [-0.77, -0.14]$</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>7.88%</td>
<td>$-0.43 [-0.80, 0.20]$</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>34.87%</td>
<td>$-0.22 [-0.40, -0.03]$</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>25.40%</td>
<td>0.03 [-0.25, 0.30]</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6.33%</td>
<td>0.04 [-0.60, 0.66]</td>
</tr>
<tr>
<td>f</td>
<td>16</td>
<td>10.68%</td>
<td>$-0.36 [-0.73, 0.16]$</td>
</tr>
</tbody>
</table>

RE Model

Test for Heterogeneity: $Q = 6.80$, df = 6, $p = 0.24$

Pearson's $r$ (negative correlation supports our hypothesis)
Results: overall

95% confidence interval is wide [-0.46, 0.03], but most of it supports our hypothesis.
Results: overall

meta-analysis weights these two datasets (with 50 and 100 snippets) much higher than the others

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Snippets</th>
<th>Weights</th>
<th>Estimate [95% CI]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>14.84%</td>
<td>-0.52 [-0.77, -0.14]</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>7.88%</td>
<td>-0.43 [-0.80, 0.20]</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>34.87%</td>
<td>-0.22 [-0.40, -0.03]</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>25.40%</td>
<td>0.03 [-0.25, 0.30]</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>6.33%</td>
<td>0.04 [-0.60, 0.66]</td>
</tr>
<tr>
<td>f</td>
<td>16</td>
<td>10.68%</td>
<td>-0.36 [-0.73, 0.16]</td>
</tr>
</tbody>
</table>

RE Model
Test for Heterogeneity: $Q = 6.80, \ df = 5, \ p = 0.24$

Pearson's $r$ (negative correlation supports our hypothesis)
Results: interpretation

- Our results give **mild but suggestive** support for our hypothesis
Results: interpretation

- Our results give **mild but suggestive** support for our hypothesis
  - especially given our **relatively conservative** statistical methods
Results: interpretation

- Our results give **mild but suggestive** support for our hypothesis
  - especially given our **relatively conservative** statistical methods
- The main limitation preventing us from making stronger conclusions is **the small number of snippets** in prior work
Results: interpretation

- Our results give **mild but suggestive** support for our hypothesis
  - especially given our **relatively conservative** statistical methods
- The main limitation preventing us from making stronger conclusions is **the small number of snippets** in prior work
  - future work: new study with a **larger number of snippets**
Results: secondary analyses
Results: secondary analyses: per-tool

- per-tool analysis:
  - same meta-analysis using one tool’s warnings
Results: secondary analyses: per-tool

- **per-tool** analysis:
  - same meta-analysis using one tool’s warnings
  - results were **similar**:
    - all tools have same pattern of correlations
    - gives us **a bit more confidence**
Results: secondary analyses: ablation

- leave-one-out ablation analysis:
  - same meta-analysis without the warnings from each tool
Results: secondary analyses: ablation

- **leave-one-out ablation** analysis:
  - same meta-analysis without the warnings from each tool
  - results *nearly identical*, implying no one tool dominates
Results: secondary analyses: categories

- **per-metric-category** analysis:
  - same meta-analysis, but with only metrics from one category
  - correctness, rating, time, and physiological categories
Results: secondary analyses: categories

- **per-metric-category** analysis:
  - same meta-analysis, but with only metrics from one category
  - correctness, rating, time, and physiological categories
  - similar results; **too-wide** confidence intervals (except rating)
Contributions
Contributions

- The first **empirical evidence** of a correlation between verifiability and understandability
  - supports the **common wisdom** of verification experts
Contributions

● The first **empirical evidence** of a correlation between verifiability and understandability
  ○ supports the **common wisdom** of verification experts

● **Implications** for verification tool builders, verification tool users, and comprehensibility researchers
Contributions

- The first **empirical evidence** of a correlation between verifiability and understandability
  - supports the **common wisdom** of verification experts
- **Implications** for verification tool builders, verification tool users, and comprehensibility researchers
- A **replication package** with our scripts and data, so that others can repeat or extend our experiments
  - [https://tinyurl.com/34hv45bm](https://tinyurl.com/34hv45bm)
Contributions

- The first **empirical evidence** of a correlation between verifiability and understandability
  - supports the **common wisdom** of verification experts
- **Implications** for verification tool builders, verification tool users, and comprehensibility researchers
- A **replication package** with our scripts and data, so that others can repeat or extend our experiments
  - [https://tinyurl.com/34hv45bm](https://tinyurl.com/34hv45bm)

Thanks to my fabulous collaborators!