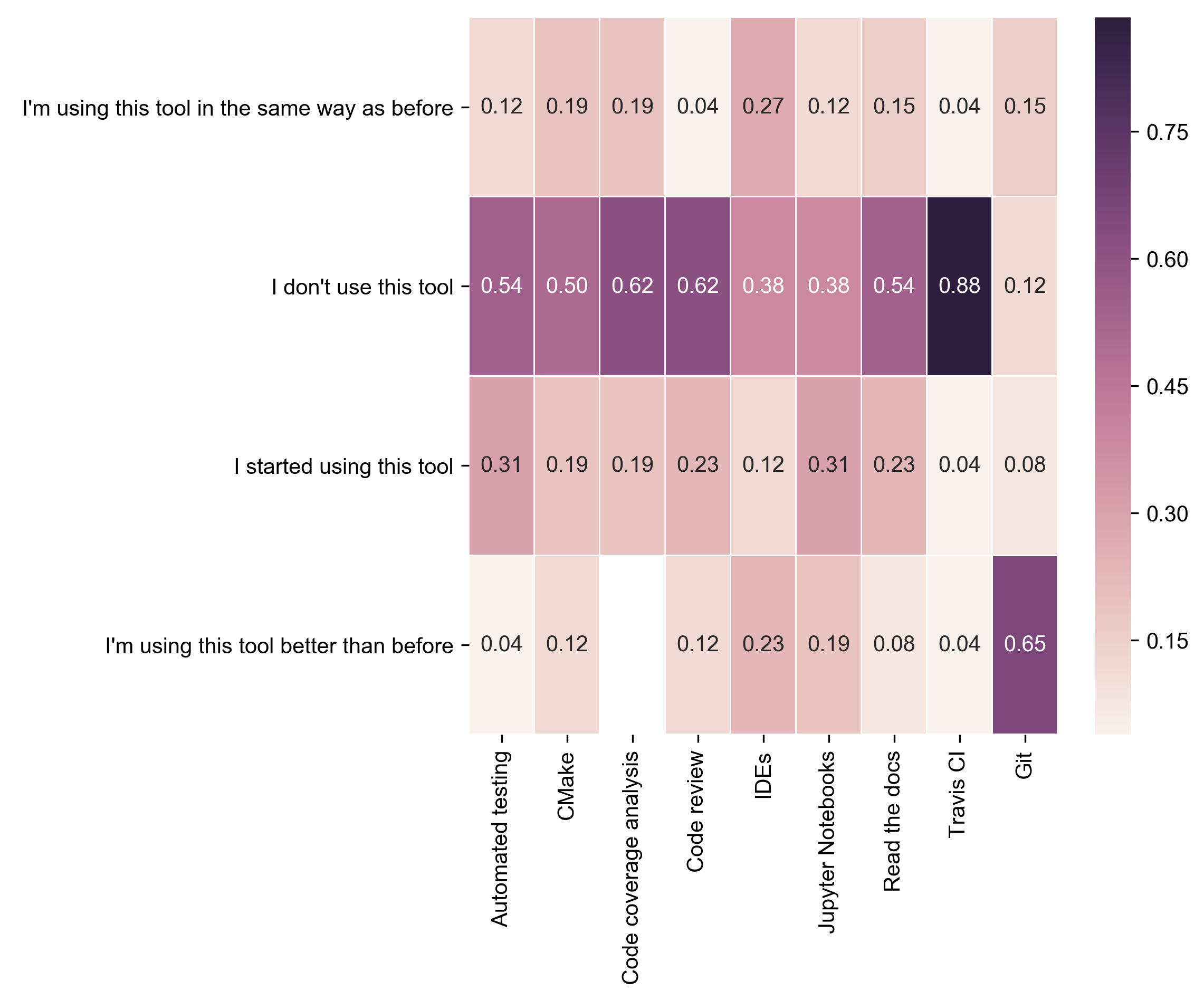
## First year of CodeRefinery

Is there room for improvement in how scientific code in developed in academia? Nowadays, developing software forms a cornerstone of research projects conducted across a wide range of scientific disciplines, but researchers often lack training in modern methods and tools for software development. The CodeRefinery project ([coderefinery.org](http://coderefinery.org)), launched last autumn under the umbrella of NeIC (Nordic e-Infrastructure Collaboration), is founded on the principle that better software leads to better research, and that it is never too late for a researcher, nor too early, to adopt better practices in day-to-day code development. A core aspect of the project is to organize and teach concise three-day workshops on collaborative, sustainable scientific software development. The aim of these workshops is not only to bring researchers up to date with modern tools and practices, but also to encourage researchers to share with colleagues what they have learnt when returning to their home institutions, and in the long run contribute to an improved culture in research software development. In addition to delivering workshops, CodeRefinery has set up a free platform for distributed version control of public and private code. This repository hosting service, based on a GitLab instance, is open to researchers based in the Nordic countries and is available at [source.coderefinery.org](http://source.coderefinery.org).

Now in its second year, CodeRefinery leaves behind an eventful first year packed with workshops, course material development, infrastructure building for source code hosting and community engagement. Workshops have so far been held in Helsinki, Stockholm, Copenhagen, Tromsø, Aarhus and Linköping, and in addition shorter workshops and discussion meetings have been held in Oslo, Umeå and Stockholm, along with a sister event in Manchester, UK. The majority of workshop attendants have been PhD students and postdocs who need to write code or scripts for research, but many participants have been either more junior or more senior. A key question for the project, as it now enters its second year, is whether the workshops have been successful in bringing about a positive change in how research code is developed. To shed light on this question a post-workshop survey was conducted, and the results have been encouraging: over 80% of former participants report that their code is both more reusable and more reproducible, 78% report that it has become easier to collaborate on software development with colleagues, and 62% report that they have introduced one or more colleagues to new tools or practices. The graph below shows how survey respondents evaluate the impact of attending a CodeRefinery workshop on their usage of various software development tools. Version control with Git is a core topic in CodeRefinery workshops, and the graph below shows that this makes a large impact on the work of workshop participants: 65% report that their usage of Git has improved after the workshop. Even other topics, which due to time constraints take up less than 2 hours of the 3-day workshop schedule, have significant impact. Automated testing and Jupyter Notebooks have been adopted by 31% of former participants, and 23% report that they have started using code review and the Read the Docs platform in their projects.



The second year of the CodeRefinery project will also be its last, at least in its current form, but a follow-up project is in the pipeline. However, a major challenge for a CodeRefinery-2.0 project will be how to scale up. Funding is restricted to a handful of instructors, and based on experience up to now this is not enough to meet demand. An interesting future prospect is for the CodeRefinery workshops to grow into a community initiative based on voluntary efforts by motivated individuals in the academic world who want to make a positive impact on the quality of research software. That this is not an unattainable goal is demonstrated by the success of the Software Carpentry project (https://software-carpentry.org/about/), which has been teaching basic computing skills to researchers since 1998, and whose instructors are, without exception, “regular” academics who enjoy teaching and find it meaningful to share their expertise to fellow researchers (and in doing so pick up new skills themselves!).

Before this becomes an urgent question, CodeRefinery members have much to look forward to.

Workshops are being planned for Helsinki in December 2017, and in 2018 the project will visit Trondheim, Turku, Odense, Uppsala, Oslo and, finally, Reykjavik in the late summer. If you want to attend a workshop, follow the project website (you can also sign up to be notified of future workshops in your area, see coderefinery.org/workshops). If you’re interested in helping out and/or getting involved in the project, get in touch via the support line, [support@coderefinery.org](mailto:support@coderefinery.org), or via the mailing list, <https://groups.google.com/group/coderefinery>.

Until then, good luck with your software projects!