

GenderMag Improves Discoverability in the Field, Especially for Women

Emerson Murphy-Hill, Alberto Elizondo, Ambar Murillo, Marian Harbach, Bogdan Vasilescu, Delphine Carlson, Florian Dessloch



Image credit: ProtoplasmaKid, wikipedia.org

DataWorks Plus helped police misidentify pregnant woman as carjacker

NIST consistently found systems had 2-5x more false negatives for women than men

Making systems work for everyone is hard

Hill, K., 2023. Eight Months Pregnant and Arrested After False Facial Recognition Match. *The New York Times B*, 1.

Grother, P.J., Grother, P.J., Ngan, M. and Hanaoka, K., 2014. *Face recognition vendor test (FRVT)*. US Department of Commerce, National Institute of Standards and Technology.



Development



User Testing



Production

Shift Left



Development



User Testing



Production



Development

Abi (Abigail/Abishek)



- 28 Years Old
- Employed as an Accountant
- Lives in Cardiff, Wales

Abi has always liked music. When she is on her way to work in the morning, she listens to music that spans a wide variety of styles. But when she arrives at work, she turns it off, and begins her day by *scanning all her emails first to get an overall picture before answering any of them.* (This extra pass takes time but seems worth it.) Some nights she exercises or stretches, and sometimes she likes to play computer puzzle games like Sudoku

Background and Skills

Abi works as an accountant. She is comfortable with the technologies she uses regularly, but she just moved to this employer 1 week ago, and their software systems are new to her. Abi says she's a "numbers person", but she has never taken any computer programming or IT systems classes. She likes Math and knows how to think with numbers. She writes and edits spreadsheet formulas in her work. In her free time, she also enjoys working with numbers and logic, she especially likes working out puzzles and puzzle games, either on paper or on the computer.

Motivations and Attitudes

Motivations: Abi uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.

Computer Self-Efficacy: Abi has lower self confidence than her peers about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

Attitude toward Risk: Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

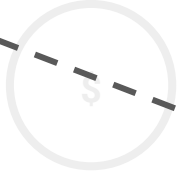
Attitude to Technology

Information Processing Style: Abi tends towards a comprehensive information processing style when she needs to gather more information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is "burst-y"; first she reads a lot, then she acts on it in a batch of activity.

Learning: by Process vs. by Tinkering: When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

¹Abi represents users with motivations/attitudes and information/learning styles similar to hers. For data on men and women similar to and different from Abi, see <http://gendermag.org/foundations.php>

Learning: by Process vs. by Tinkering : When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.



Development

Motivations and Attitudes

Motivations: Abi uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with to keep her focus on the tasks she cares about.

Computer Self-Efficacy: Abi has lower self confidence than her peers about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

Attitude toward Risk: Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

Attitude to Technology

Information Processing Style: Abi leans towards a comprehensive information processing style when she needs to gather more information. So instead of acting upon the first option that seems promising, she gathers information comprehensively to form a complete understanding of the problem before trying to solve it. Thus, her style is "burst-y"; first she reads a lot, then she acts on it in a batch of activity.

Learning: by Process vs. by Tinkering : When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

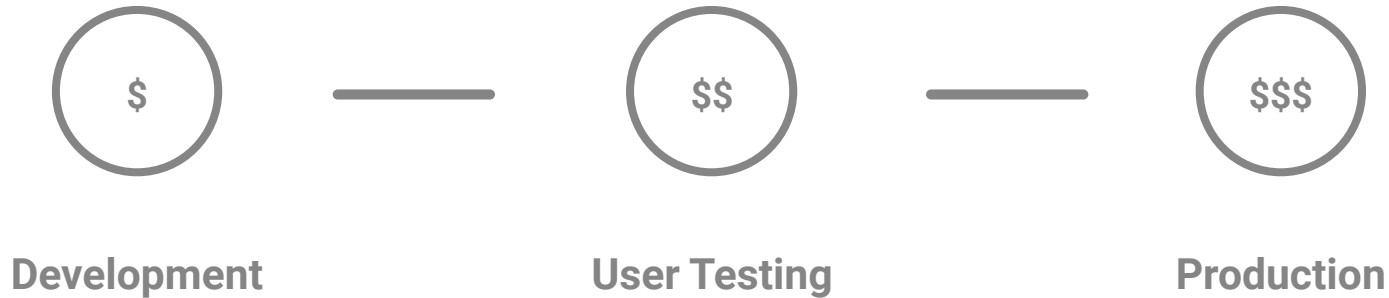
¹Abi represents users with motivations/attitudes and information learning styles similar to hers. For data on men and women similar to Abi, see <http://gendermag.org/foundations.php>.

Is GenderMag effective?

The most direct evidence from the literature:

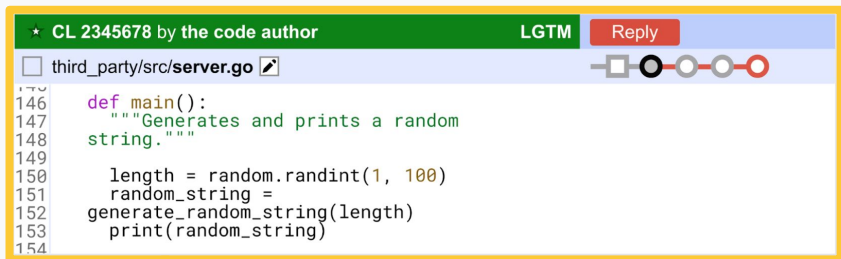
Method: Apply GenderMag to a search engine, count task failures in a laboratory study with 20 faculty members and students

Finding: Women had 2x as many task failures before GenderMag, but the same number in the GenderMag version



Vorvoreanu, M., Zhang, L., Huang, Y.H., Hilderbrand, C., Steine-Hanson, Z. and Burnett, M., 2019, May. From gender biases to gender-inclusive design: An empirical investigation. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1-14).

What We Did



A screenshot of a code review comment. The comment is titled "CL 2345678 by the code author" and has a green "LGTM" badge and a red "Reply" button. The code snippet shows a Go function `main()` that generates a random string. The comment is highlighted with a yellow border.

```
146 def main():
147     """Generates and prints a random
148     string."""
149
150     length = random.randint(1, 100)
151     random_string =
152     generate_random_string(length)
153     print(random_string)
154
```



2014

Suggest Edit originally introduced into Critique



2019

In a sprint, we apply GenderMag to Suggest Edit on Critique, create mockups



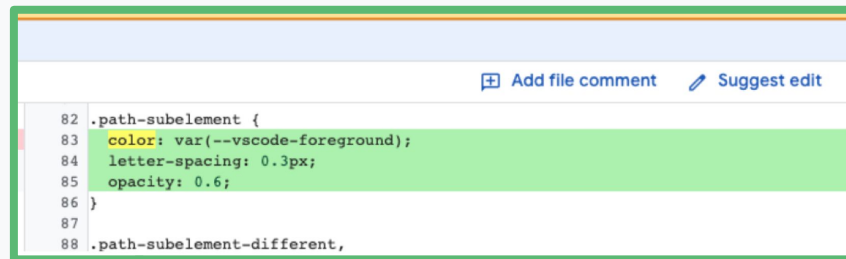
2020

Mockups implemented and deployed



2023

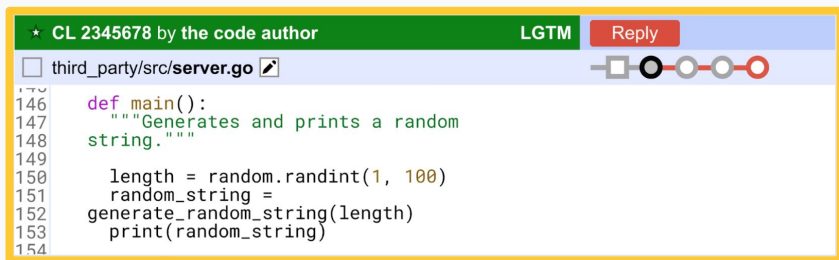
Evaluation



A screenshot of a code review comment showing CSS code. The code snippet is highlighted with a green background. The comment is highlighted with a green border.

```
82 .path-subelement {
83     color: var(--vscode-foreground);
84     letter-spacing: 0.3px;
85     opacity: 0.6;
86 }
87
88 .path-subelement-different,
```


Example Finding from GenderMag Session



A screenshot of a code editor interface. At the top, there is a green header with the text '* CL 2345678 by the code author' and 'LGTM'. To the right of the header is a red 'Reply' button. Below the header, the file path 'third_party/src/server.go' is shown. The main area contains Go code:

```
146 def main():
147     """Generates and prints a random
148     string."""
149
150     length = random.randint(1, 100)
151     random_string =
152     generate_random_string(length)
153     print(random_string)
154
```

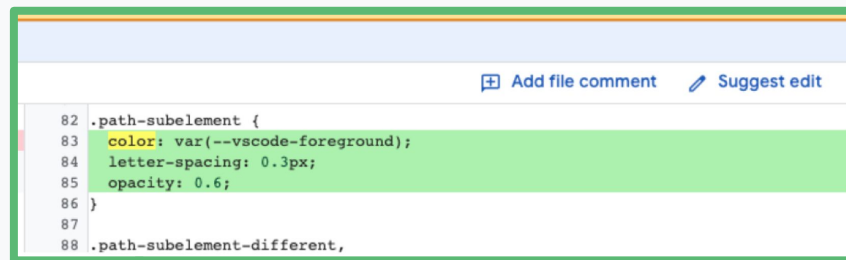
 In the top right corner of the code area, there is a small, faint icon representing the 'Suggest Edit' feature.



Before

Suggest Edit is hidden as a small, cryptic icon. Low self-efficacy/risk averse users may be not try it.

The icon also only appears on a secondary page. Non-tinkering users may not discover it.



A screenshot of a code editor interface. At the top right, there are two buttons: 'Add file comment' and 'Suggest edit'. The 'Suggest edit' button is significantly larger and more prominent than the 'Add file comment' button. Below the buttons, there is a code block:

```
82 .path-subelement {
83     color: var(--vscode-foreground);
84     letter-spacing: 0.3px;
85     opacity: 0.6;
86 }
87
88 .path-subelement-different,
```

 The 'Suggest edit' button is highlighted with a green background.



After

The Suggest Edit button now has equal prominence to the "Add file comment" button, and has text added next to the icon.

It appears in both the file view and the in-place file view

Method

How did our redesign of Suggest Edit affect how quickly Critique users *discover* Suggest Edit? Men vs. women?

Data from

- ~42,000 male and ~11,000 female Critique users
- ~3 years before and after redesign

Naive method: use data on time it takes users to discover Suggest Edit

- But that ignores data from users who *never* discovered Suggest Edit

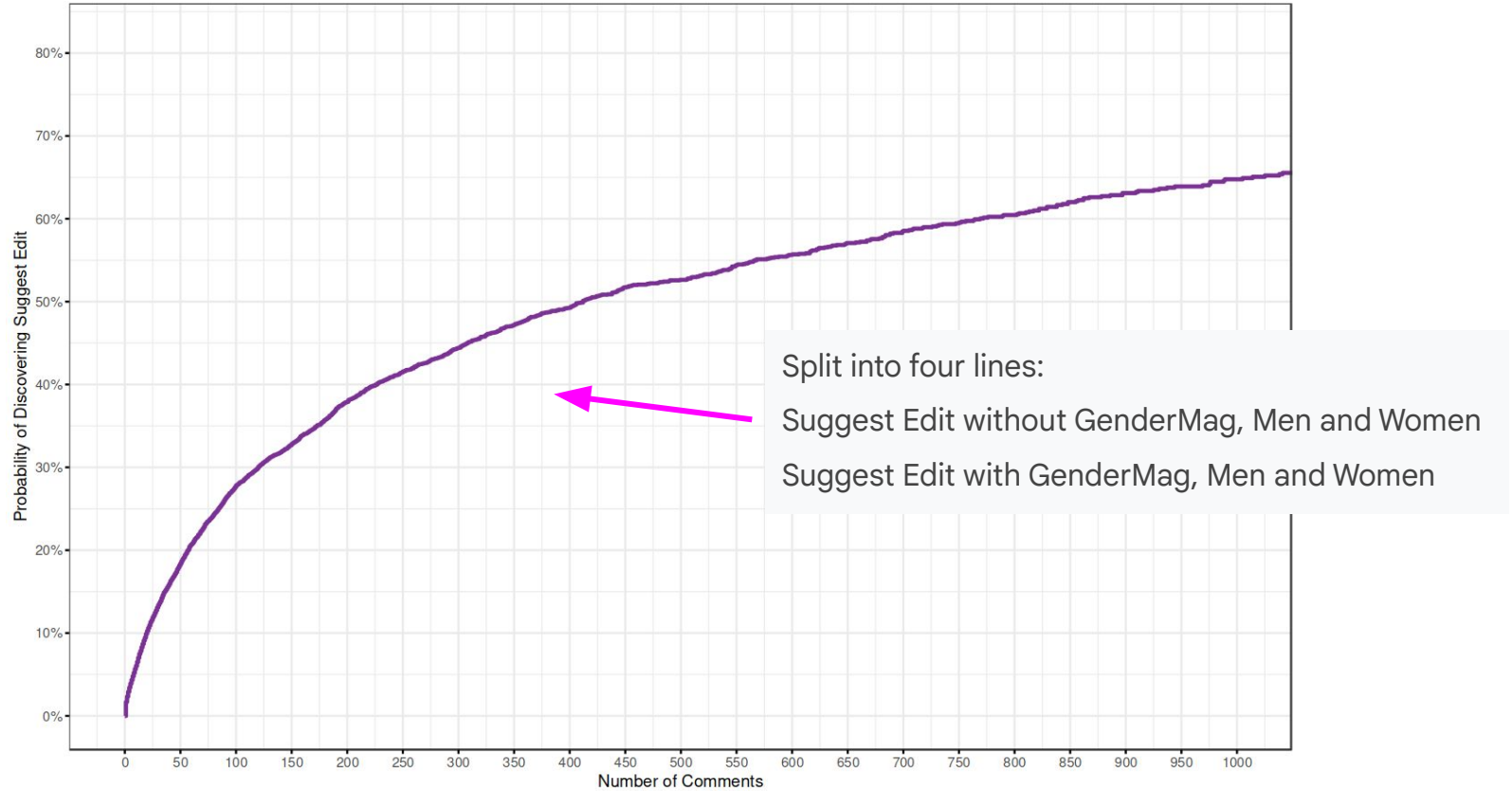
The key idea of our evaluation is that we can repurpose *survival analysis*, integrating both types of data

- Survival duration: number of comments until first Suggest Edit use
- Right censor: number of comments until end of data

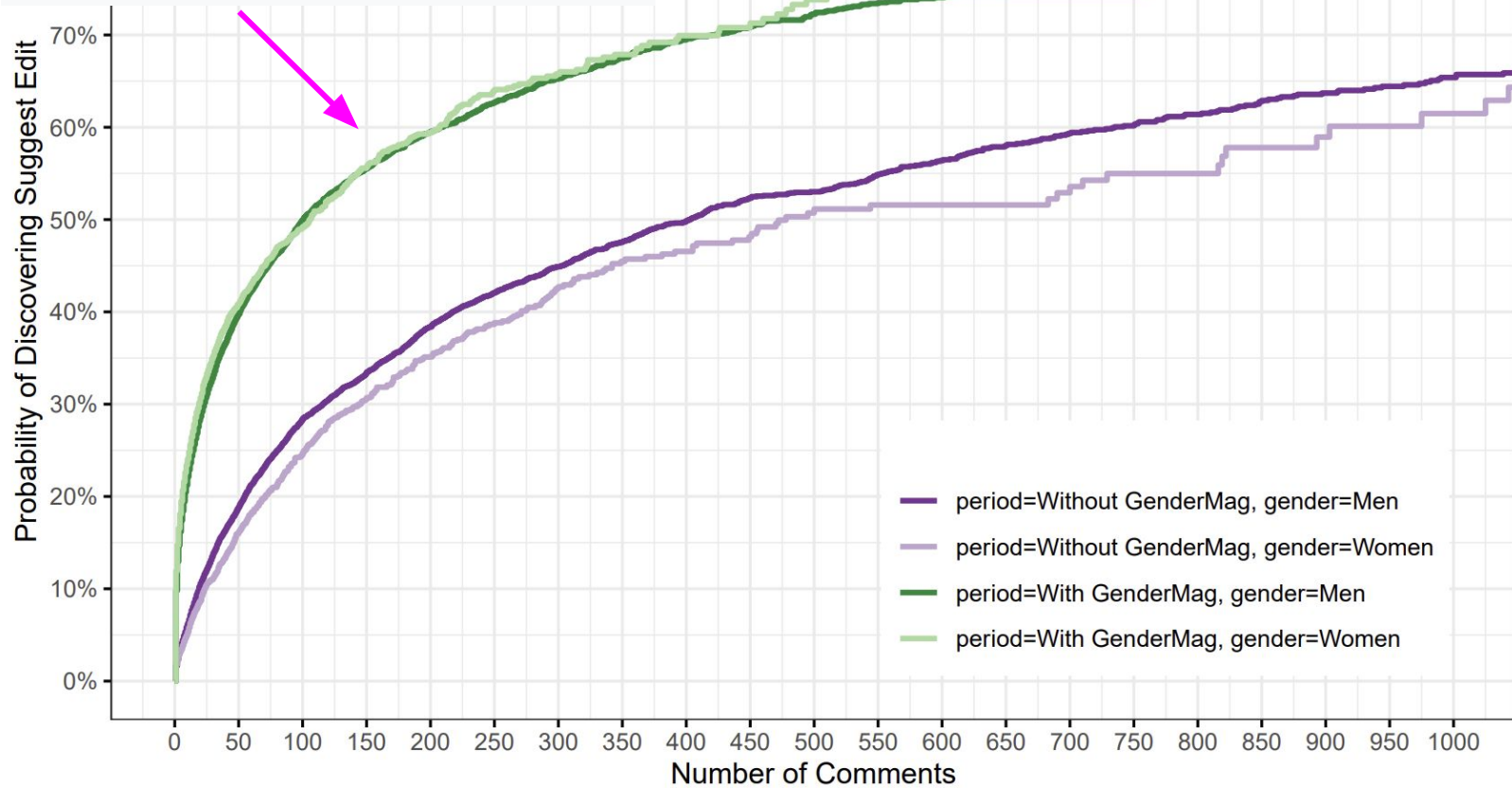
Estimate differences

- Statistically, using a Cox regression, controlling for covariates
- Intuitively, using a Kaplan-Meier plots

Method



After applying GenderMag, Suggest Edit discoverability increased 2.4x and closed the gender gap



How did we get a Distinguished Paper for changing a button?

```
* CL 2345678 by the code author LGTM Reply
third_party/src/server.go
146 def main():
147     """Generates and prints a random
148     string."""
149
150     length = random.randint(1, 100)
151     random_string =
152     generate_random_string(length)
153     print(random_string)
154
```

```
Add file comment Suggest edit
82 .path-subelement {
83     color: var(--vscode-foreground);
84     letter-spacing: 0.3px;
85     opacity: 0.6;
86 }
87
88 .path-subelement-different,
```

Write Preview

⌨ H B I ↵ <> 🔗 ⋮ ⋮ ⋮ @ 🗨 ⏪ 🗑

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 📎

Cancel Add single comment Start a review

GitHub pull request reviews have a similar feature with a near-identical discoverability problem

▲ makerofthings 4 months ago | prev | next [-]

One thing I like about it is that reviewers can suggest changes and you can accept them inline. Makes it really easy to deal with nits.

▲ eftychis 4 months ago | parent | next [-]

You can do that in Github, but for some reason a lot of reviewers are not familiar or bother doing that. Fixing nits that way or giving a suggestion improves turnaround speed greatly and builds a relationship between reviewer and proposer and the final product/commit(s).

▲ yohannparis 4 months ago | root | parent | next [-]

Yes, I have been "suggesting" a lot of one liners, typo, rephrasing, or just simple clean up of code with it. Make it a breeze, it's like playing tidy-up without having to branch out or bother much the author.

▲ froh 4 months ago | root | parent | next [-]

I feel stupid. How do I do this?

▲ mplanchard 4 months ago | root | parent | next [-]

When writing an inline comment, there's a button to make a suggestion. It inserts a markdown code fence with the existing code on the line(s), which you can edit. When you submit the comment, it shows up with diff highlighting and can be applied by the author with one click.

Essentially the same thing exists in both GitHub and GitLab

▲ froh 4 months ago | root | parent | next [-]

ah yes! duh. and thanks. I need better glasses..

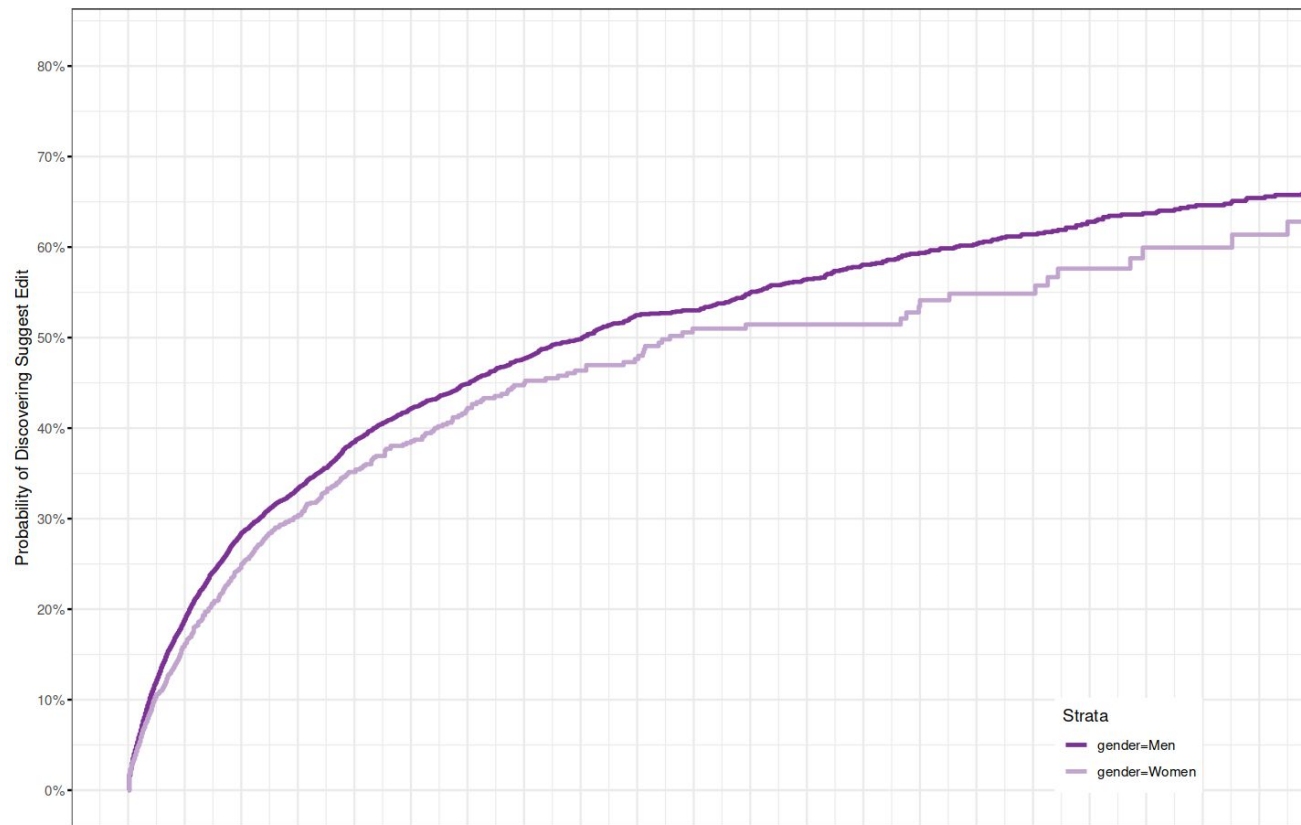
▲ bbu 4 months ago | root | parent | prev | next [-]

It's not integrated well into the UI and thus a big hassle to use. Much easier to just checkout the branch and create a commit (or just write a comment...)

▲ Tempest1981 4 months ago | root | parent | prev | next [-]

A lot of folks didn't know it was there. It's just 1 of many icons, and tbh, I'm not sure when the feature got added.

Discussion: Can we shift left further than GenderMag?



Conclusion

