Using the WebAIM Million and User Surveys to Inform Your Inclusion Efforts

Jared Smith
@jared_w_smith
WebAIM.org

There’s a shortage of good, available web accessibility data

“If we have data, let’s look at data.
If all we have is opinion,
let’s go with mine.”

I DIDN'T HAVE ANY ACCURATE NUMBERS SO I JUST MADE UP THIS ONE.
$4,629,873
scottadams@aol.com
www.dilbert.com
STUDIES HAVE SHOWN THAT ACCURATE NUMBERS AREN'T ANY MORE USEFUL THAN THE ONES YOU MAKE UP.
5-8-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.
HOW MANY STUDIES SHOWED THAT?
EIGHTY-SEVEN.

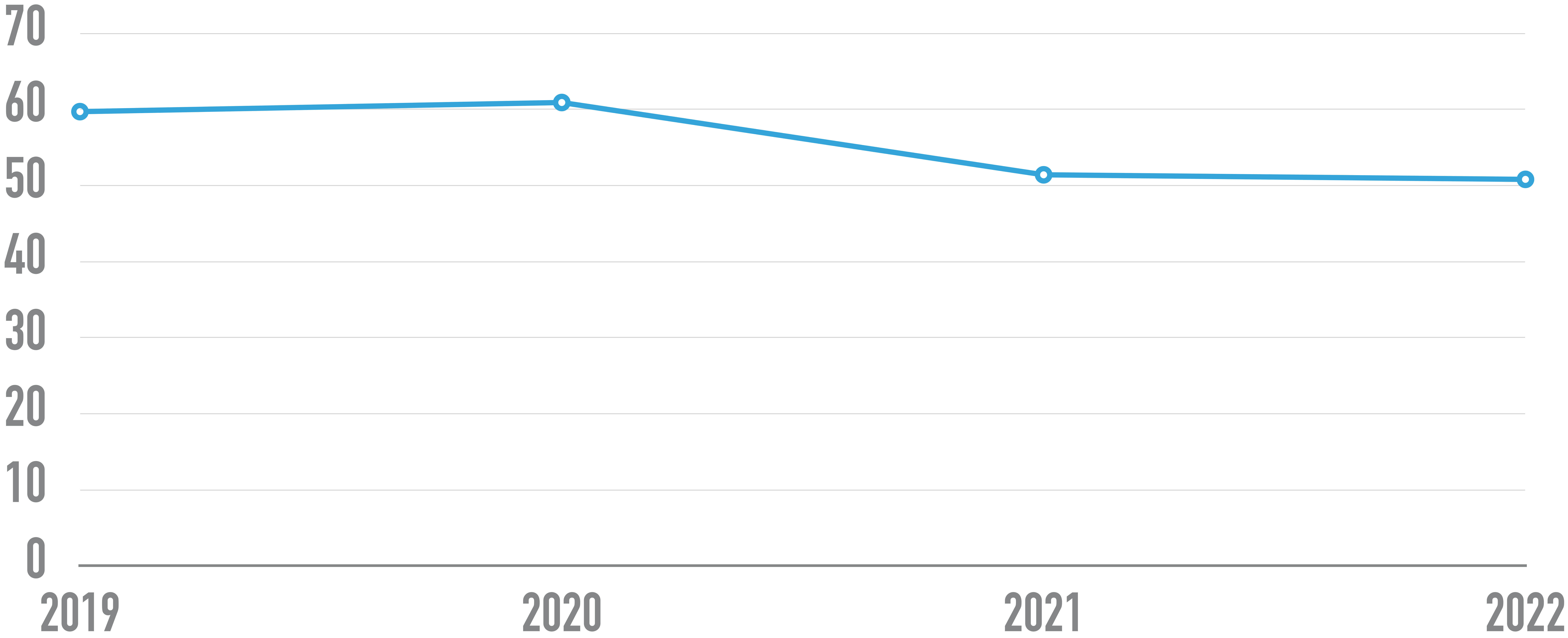
The WebAIM Million

- Accessibility testing on 1,000,000 “top” home pages tested February 2019, 2020, 2021, and 2022.
- Accessibility data collected using the WAVE API
- Detected any of 1,200 web technologies in use
- Site category metadata collected via WebShrinker
- Database stores nearly 3,000,000,000 points of data

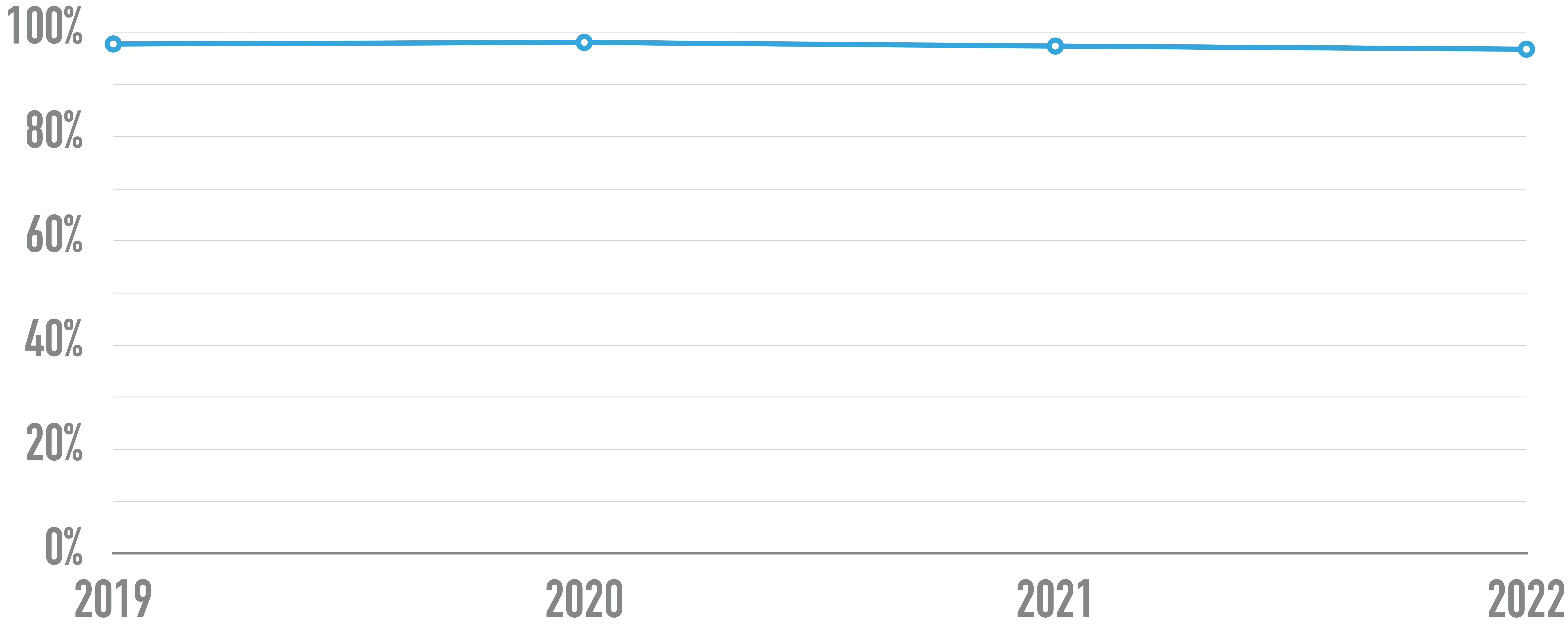
WebAIM Million 2022 Summary

- 50,829,406 errors detected
  - Average of 50.8 detectable errors per page
  - One in 18 home page elements has a detectable error
- 96.8% of pages had detectable WCAG 2 failures
  - Actual WCAG conformance rate is very, VERY low

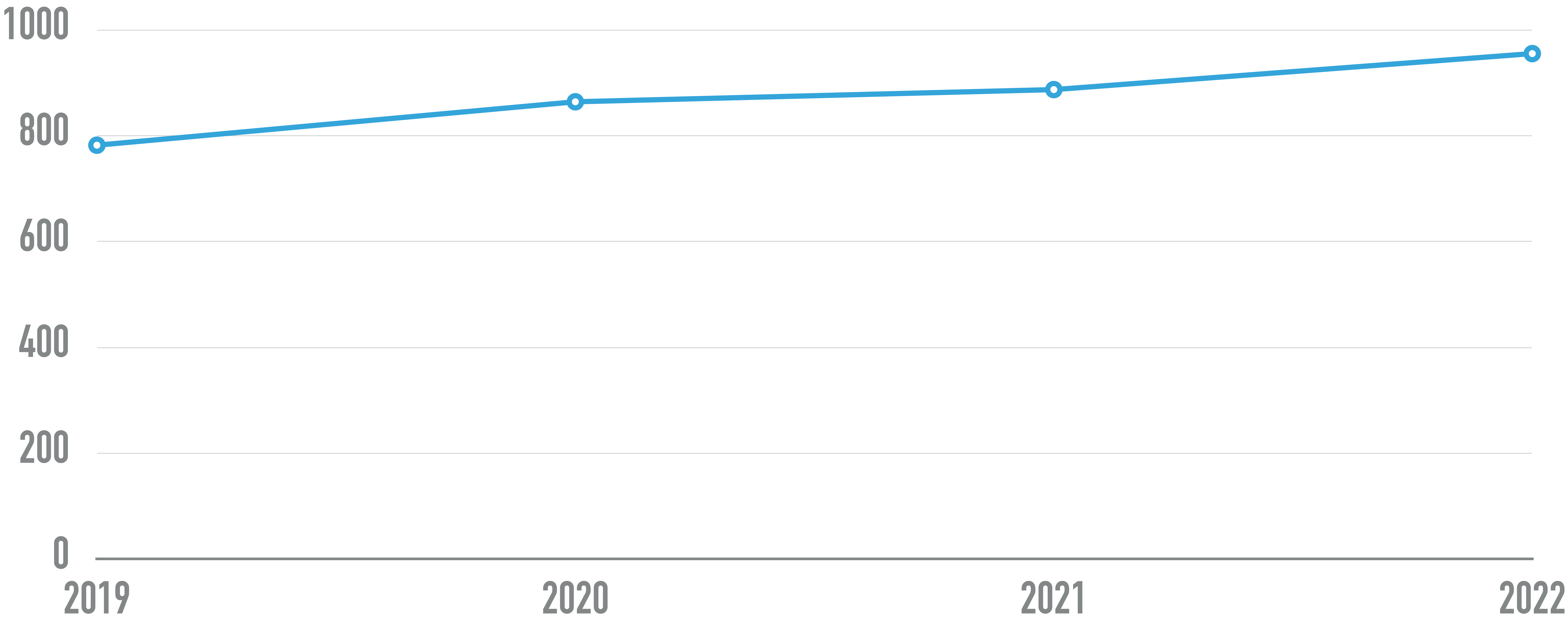
Number of detectable errors on average per page



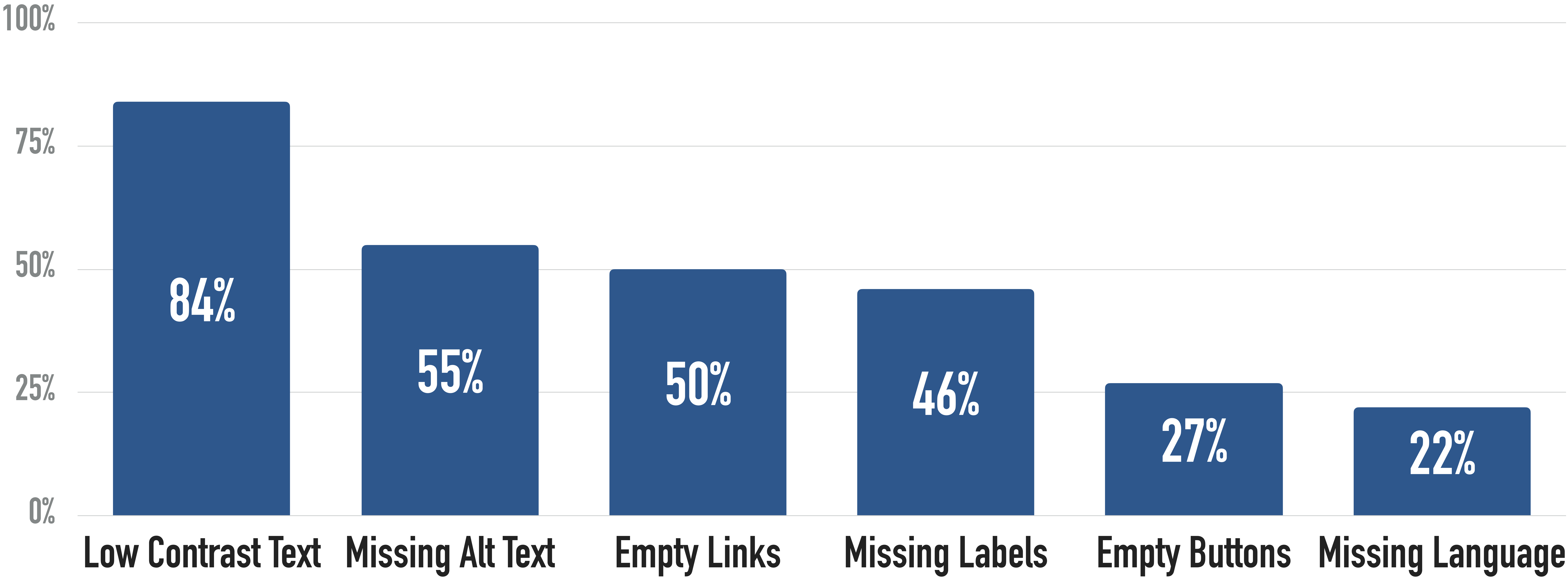
Pages with detectable WCAG failures



Number of home page elements



Percentage of pages with most common errors



96% of all errors detected fall into these six categories.

Addressing only these few types of issues would **significantly** improve accessibility across the web.

Error details

- Average of 32 low contrast text elements per page
- Over 1/3 of images had missing, questionable, or redundant alternative text
  - 50% of images missing alt text were linked images
- 45% of the 4.4M form inputs were not labeled

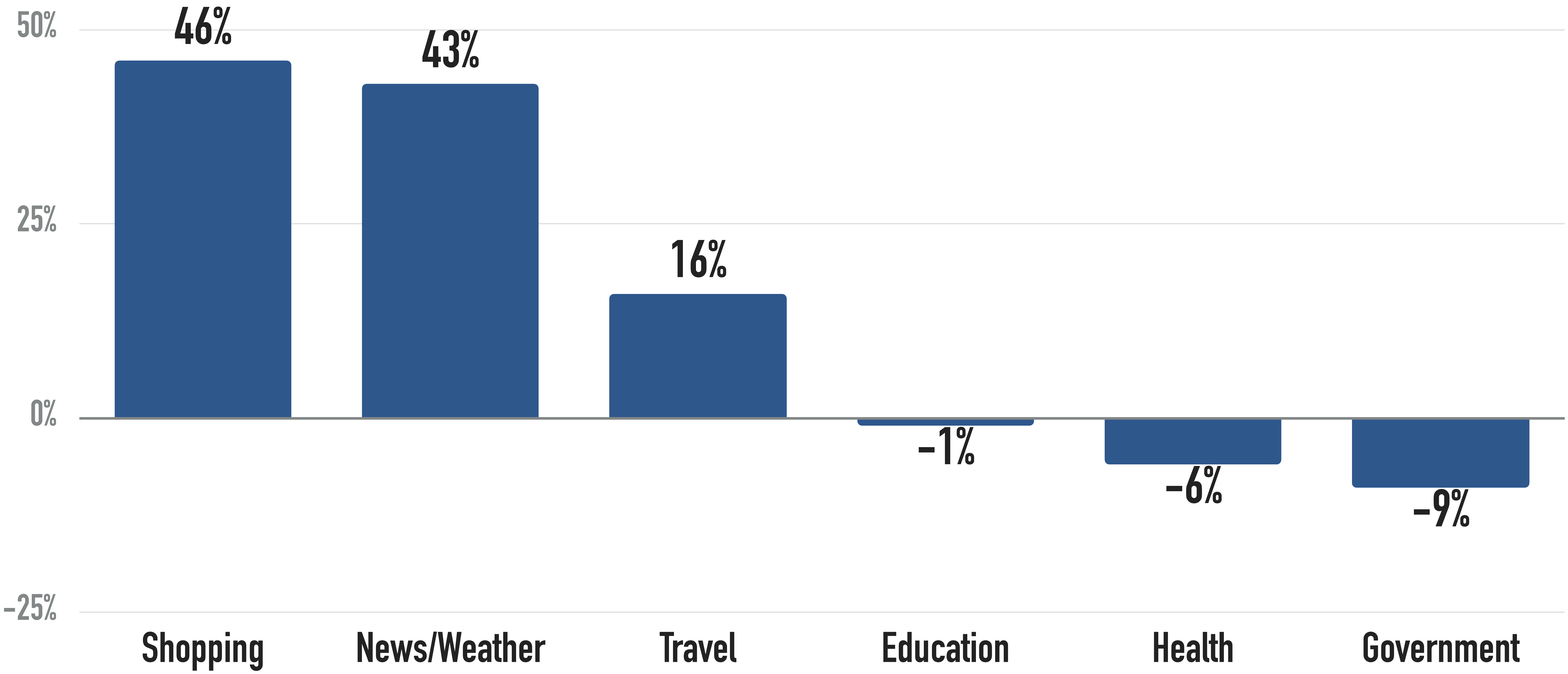
Accessible Rich Internet Applications (ARIA)

- 68% of home pages had ARIA (excluding landmark roles).
- 48 ARIA attributes per home page on average.
- Pages with ARIA present averaged 24 **more** detectable errors than pages without ARIA.
  - More ARIA = More detectable errors
  - 60% of ARIA menus are not properly coded.

Other WebAIM Million data

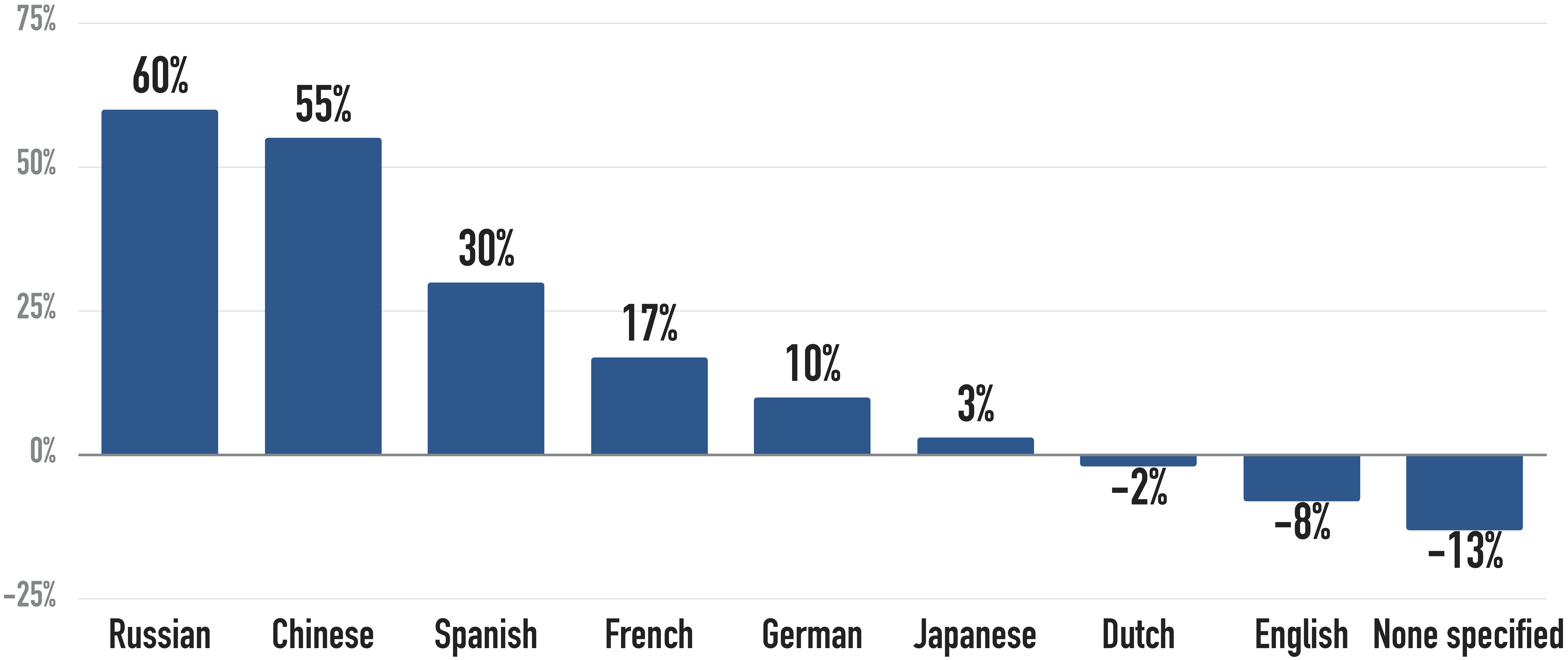
- 22% of pages had links with ambiguous text ("click here", "more", "continue", etc.).
    - 5.9 ambiguous links per page when present.
- 10% of pages had a "skip" link. 11% of them were broken.
- 11% of home pages had no headings. 38% of pages had skipped heading levels.
- Pages with valid HTML5 doctypes had nearly twice as many elements (987 vs. 508) and 35% more errors than other pages.

Difference in errors by site category



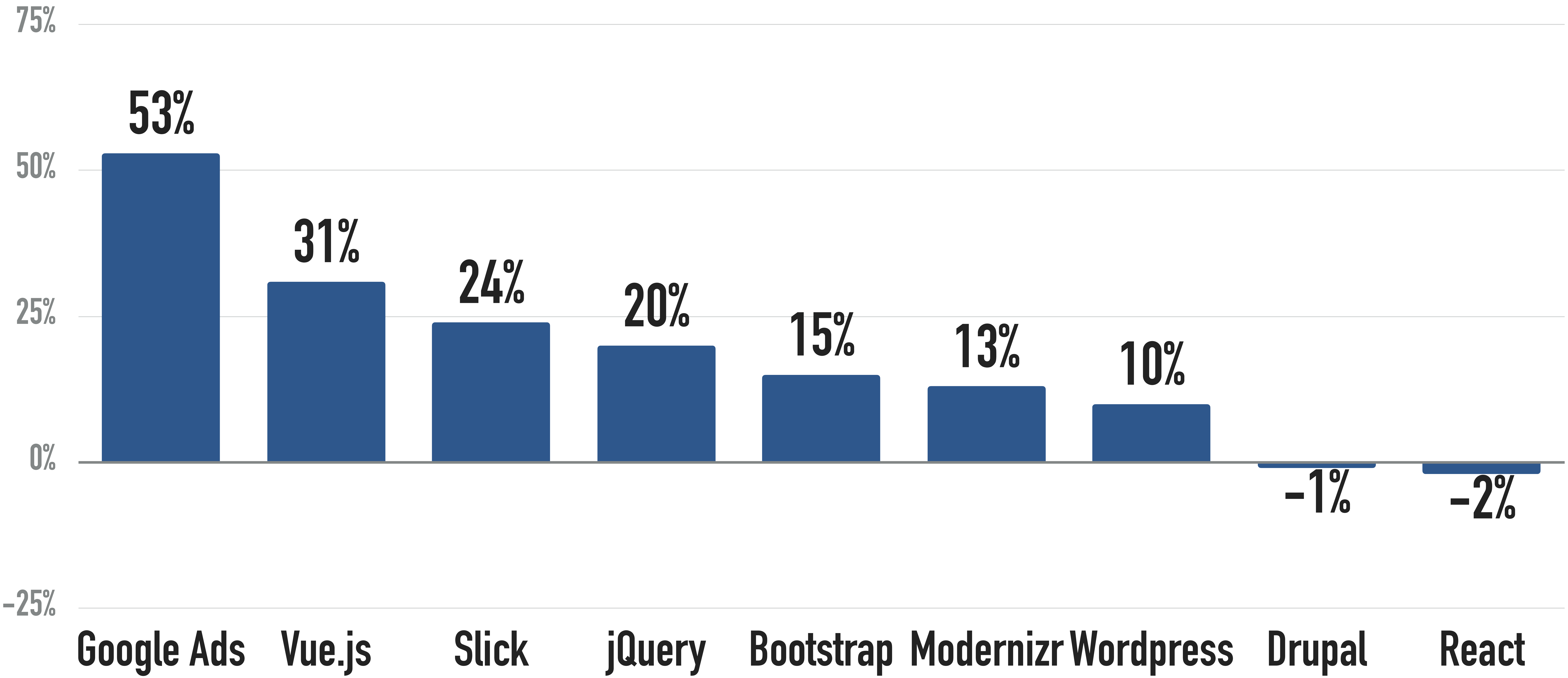
% difference in errors from the overall average (positive means more errors than average)

Difference in errors by document language

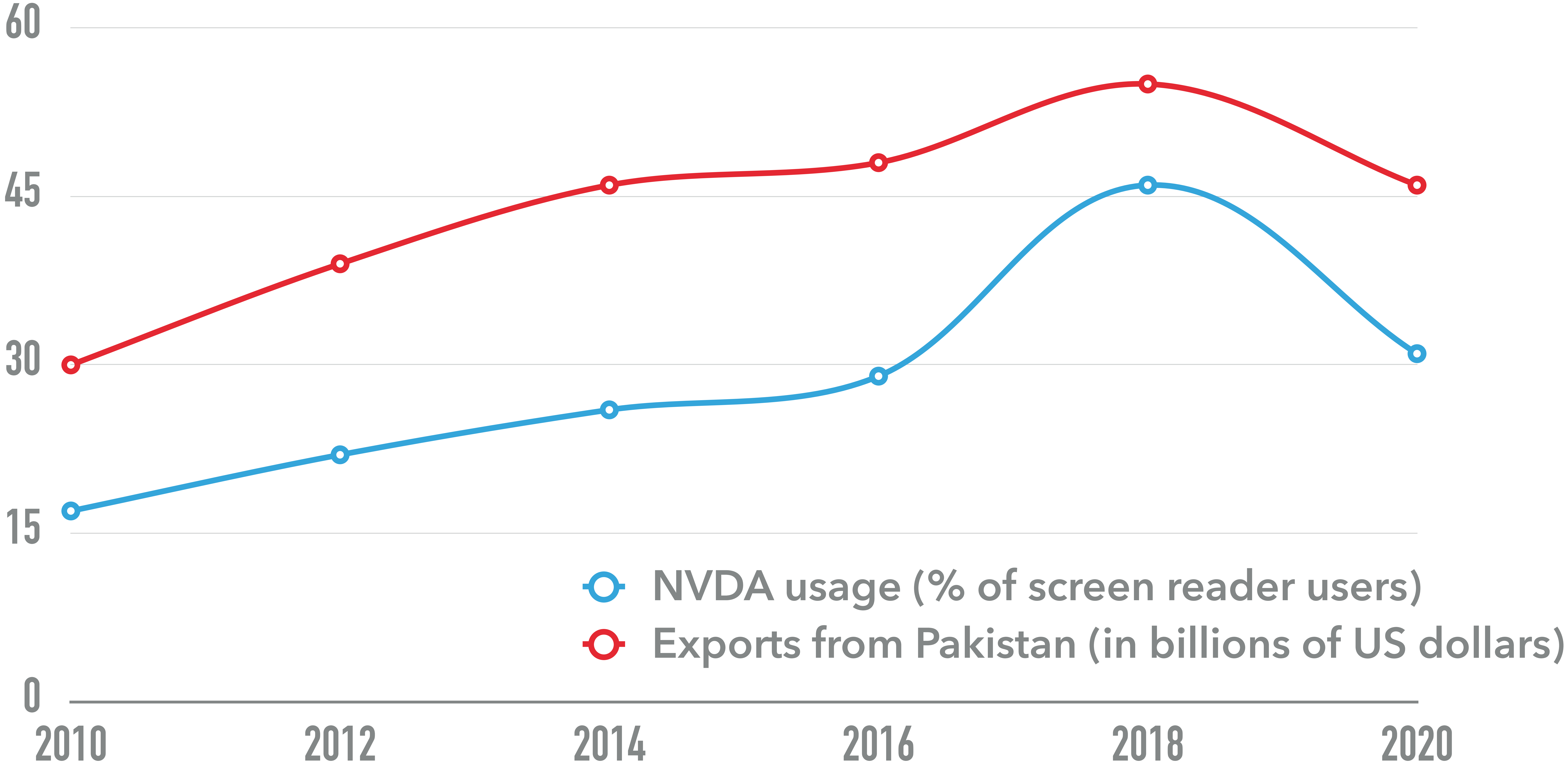


% difference in errors from the overall average (positive means more errors than average)

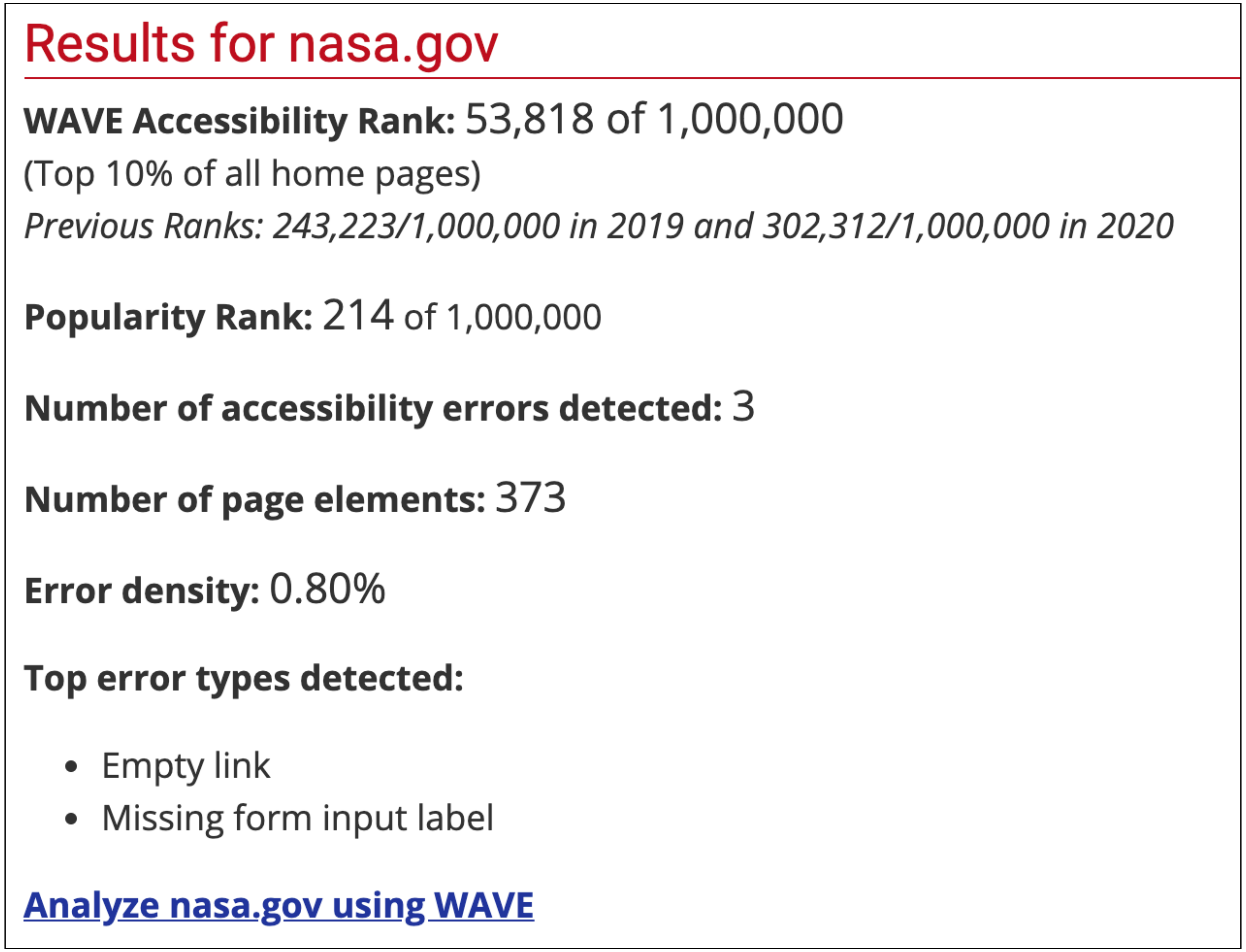
Difference in errors by technology



NVDA Usage
correlates strongly with
Exports from Pakistan



WebAIM Million report at webaim.org/projects/million/



Results for nasa.gov

WAVE Accessibility Rank: 53,818 of 1,000,000
(Top 10% of all home pages)
Previous Ranks: 243,223/1,000,000 in 2019 and 302,312/1,000,000 in 2020

Popularity Rank: 214 of 1,000,000

Number of accessibility errors detected: 3

Number of page elements: 373

Error density: 0.80%

Top error types detected:

- Empty link
- Missing form input label

Analyze nasa.gov using WAVE

webaim.org/projects/million/lookup

The "Error Density" problem

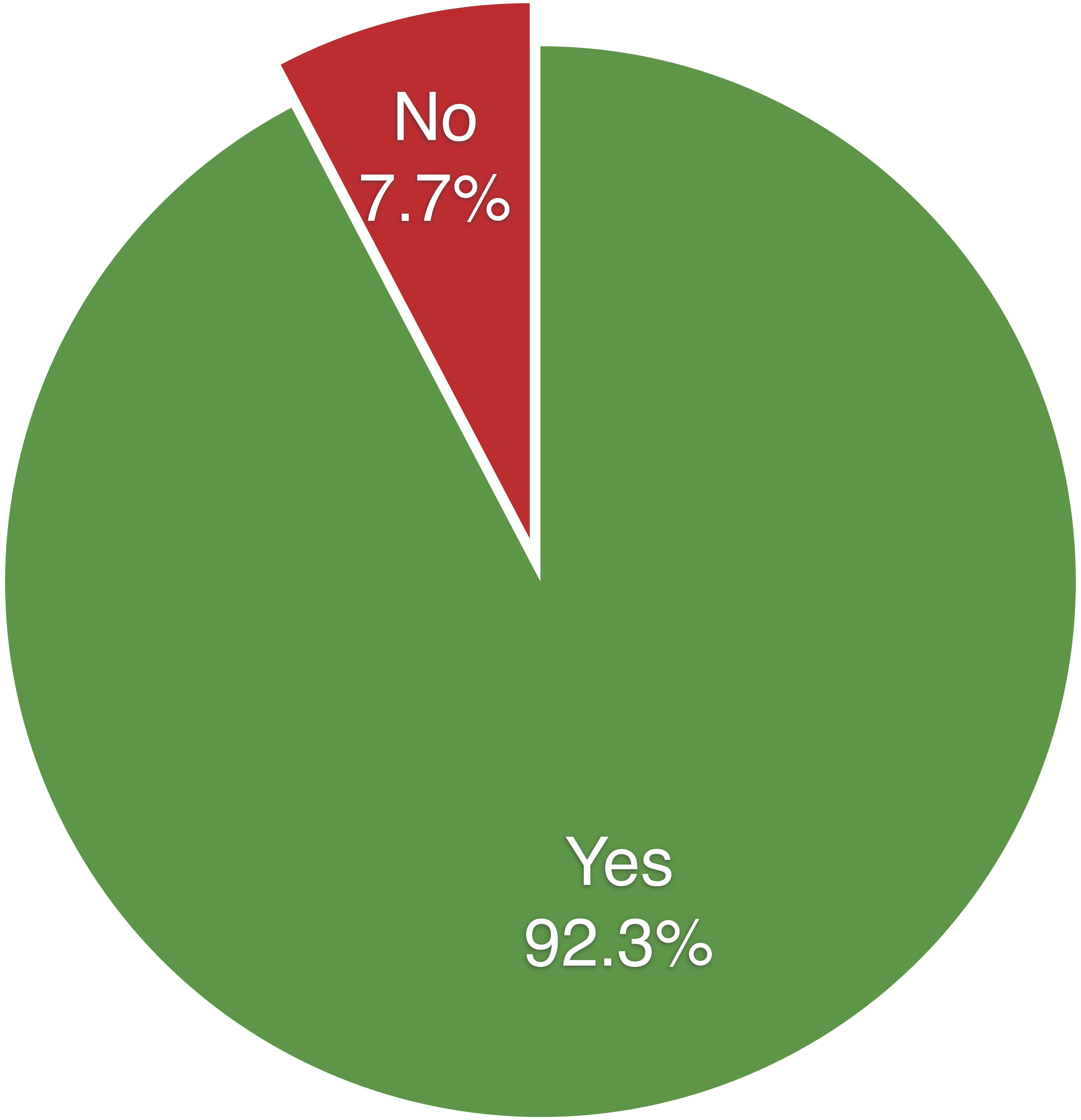
- Error density = # of detectable errors / # of page elements
- Number of errors ranged from 0 to 68826
- Error density ranged of 0% to 99.9%
- Pages without Vue.js:
  - 51.1 errors and 879 elements = 5.8% error density
- Page with Vue.js:
  - 67.3 errors and 1283 elements = 5.2% error density
- 32% more errors, but 46% more elements, so 10% lower error density
- Does Vue.js correspond with better or worse accessibility???

WebAIM Surveys

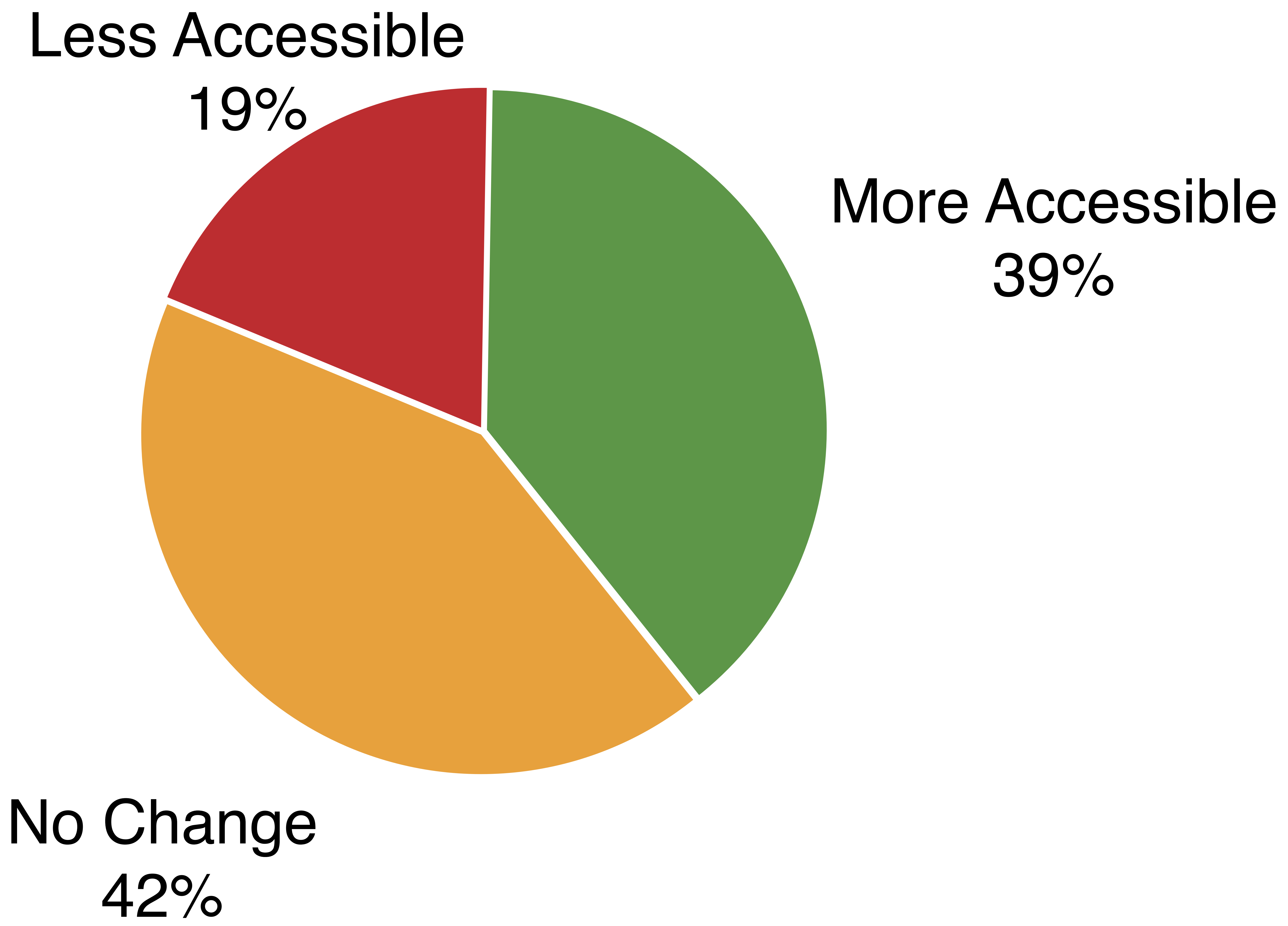
- 9 X Screen Reader User Surveys
  - Conducted every ~18 months
- 2 X Surveys of Users with Low Vision
- 3 X Surveys of Web Accessibility Practitioners
- Survey of Users with Motor Disabilities

webaim.org/projects/

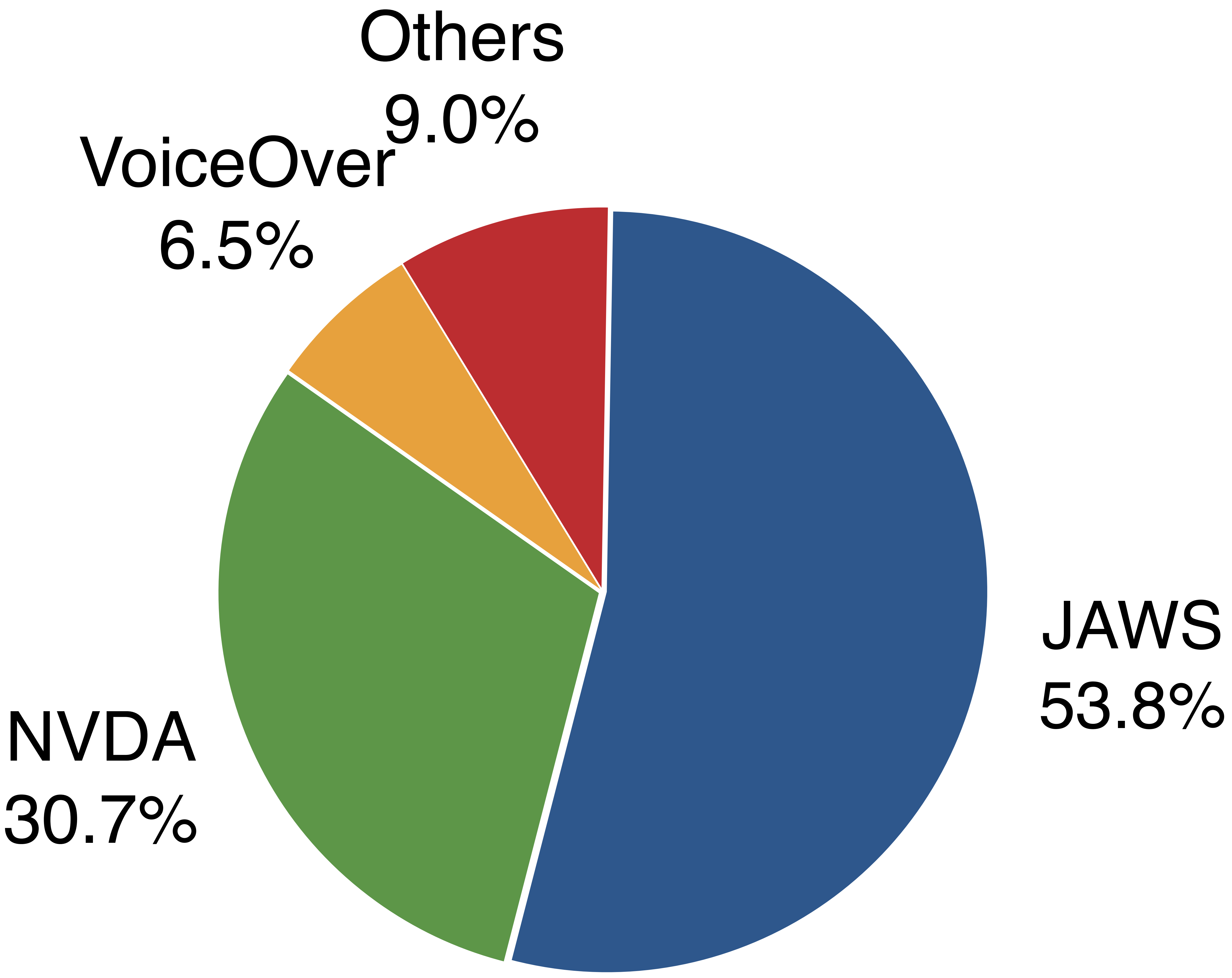
Do you use a screen reader due to a disability?



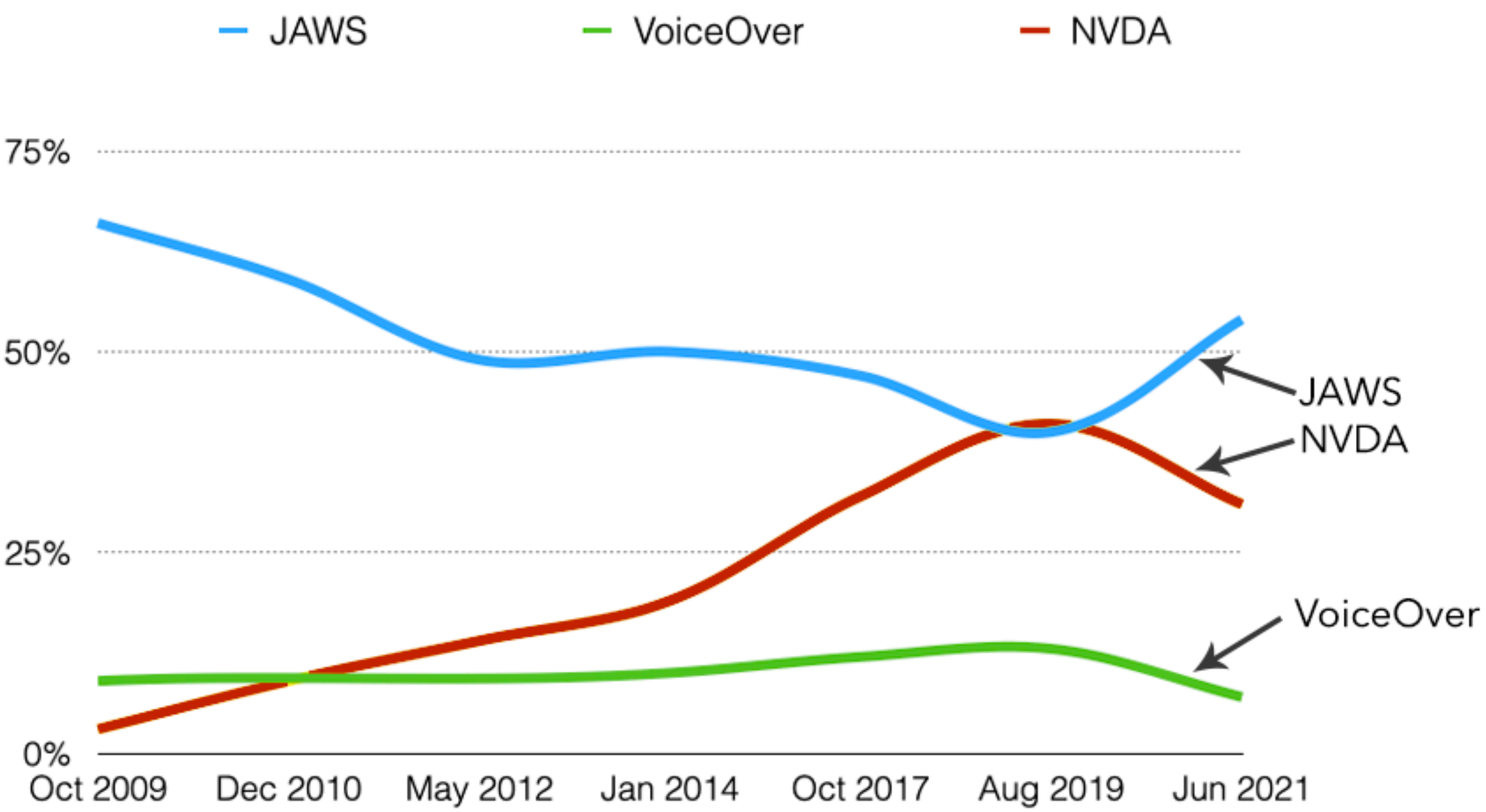
Feelings regarding the accessibility of web content over the previous year?



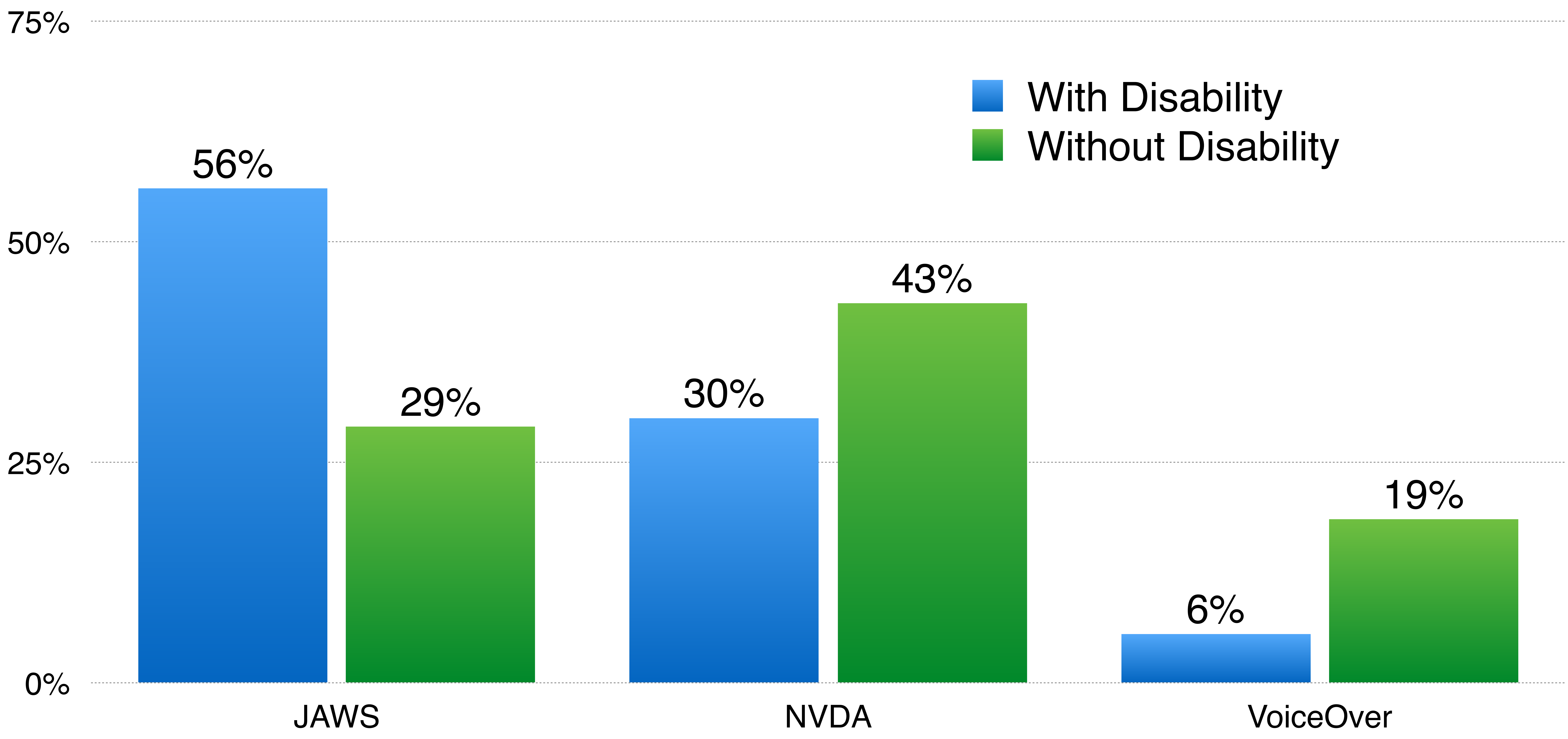
Primary Desktop/Laptop Screen Reader



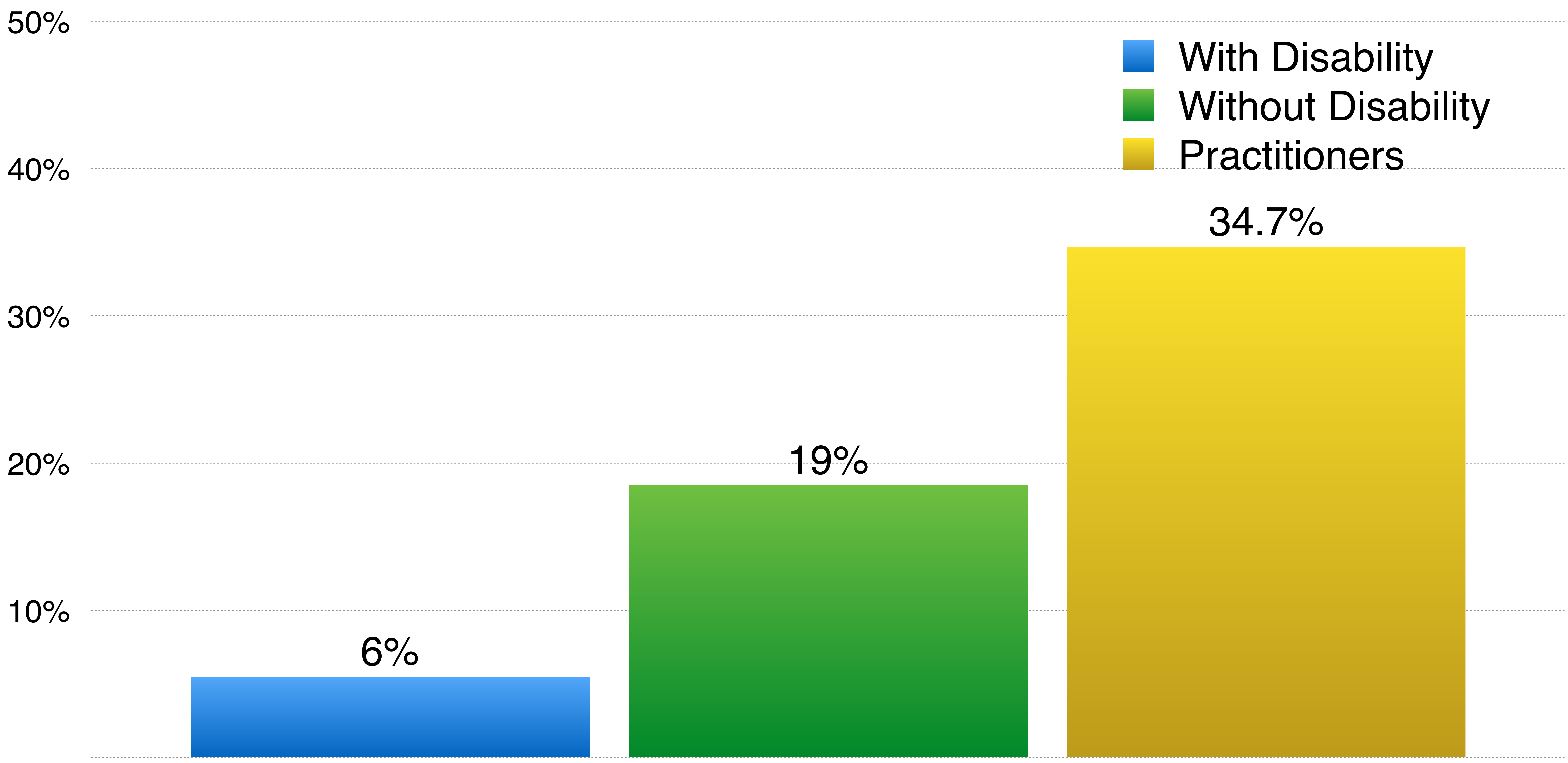
Primary Desktop/Laptop Screen Reader



Primary Desktop/Laptop Screen Reader



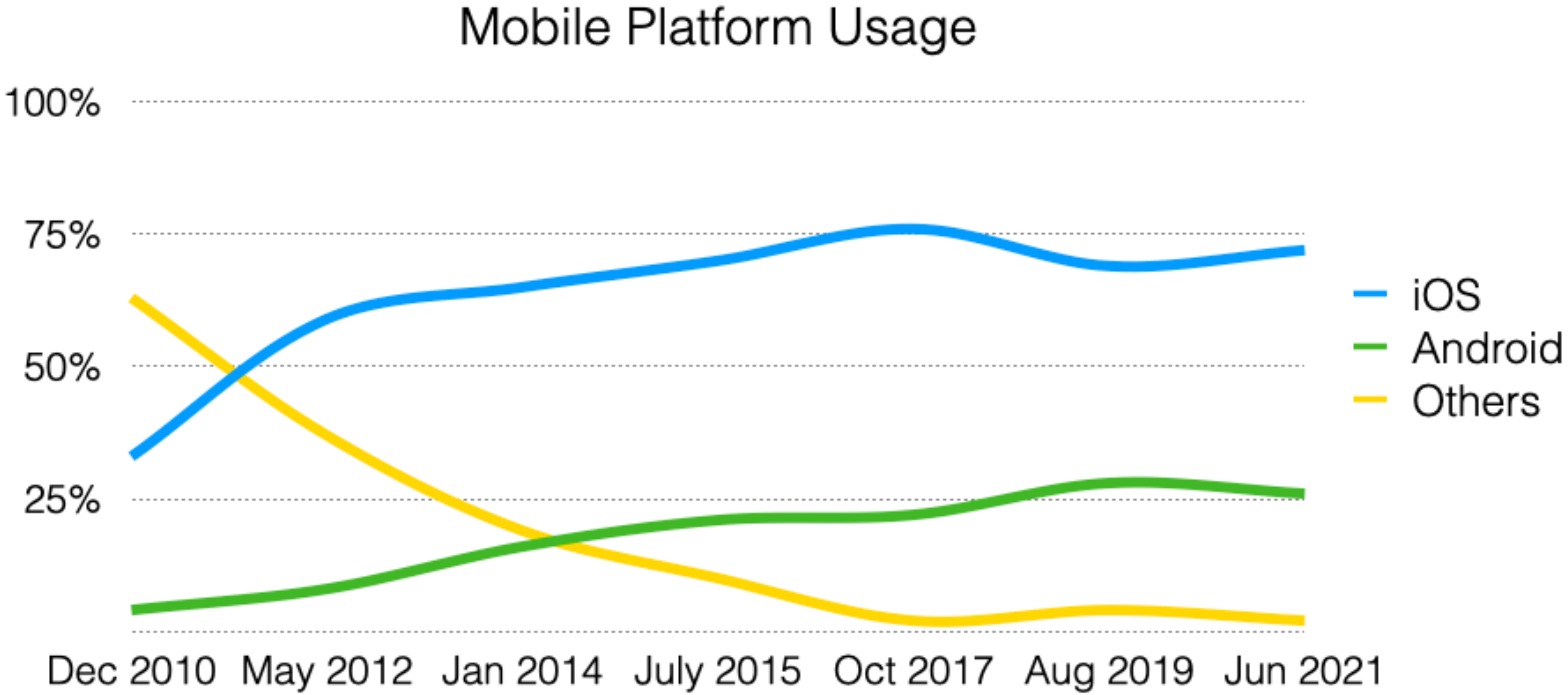
VoiceOver Usage



Screen Reader / Browser Combinations

Screen Reader & Browser	% of Respondents
JAWS with Chrome	33%
NVDA with Chrome	13%
JAWS with Edge	13%
NVDA with Firefox	10%
JAWS with Firefox	5%
VoiceOver with Safari	5%
Other combinations	21%

Mobile

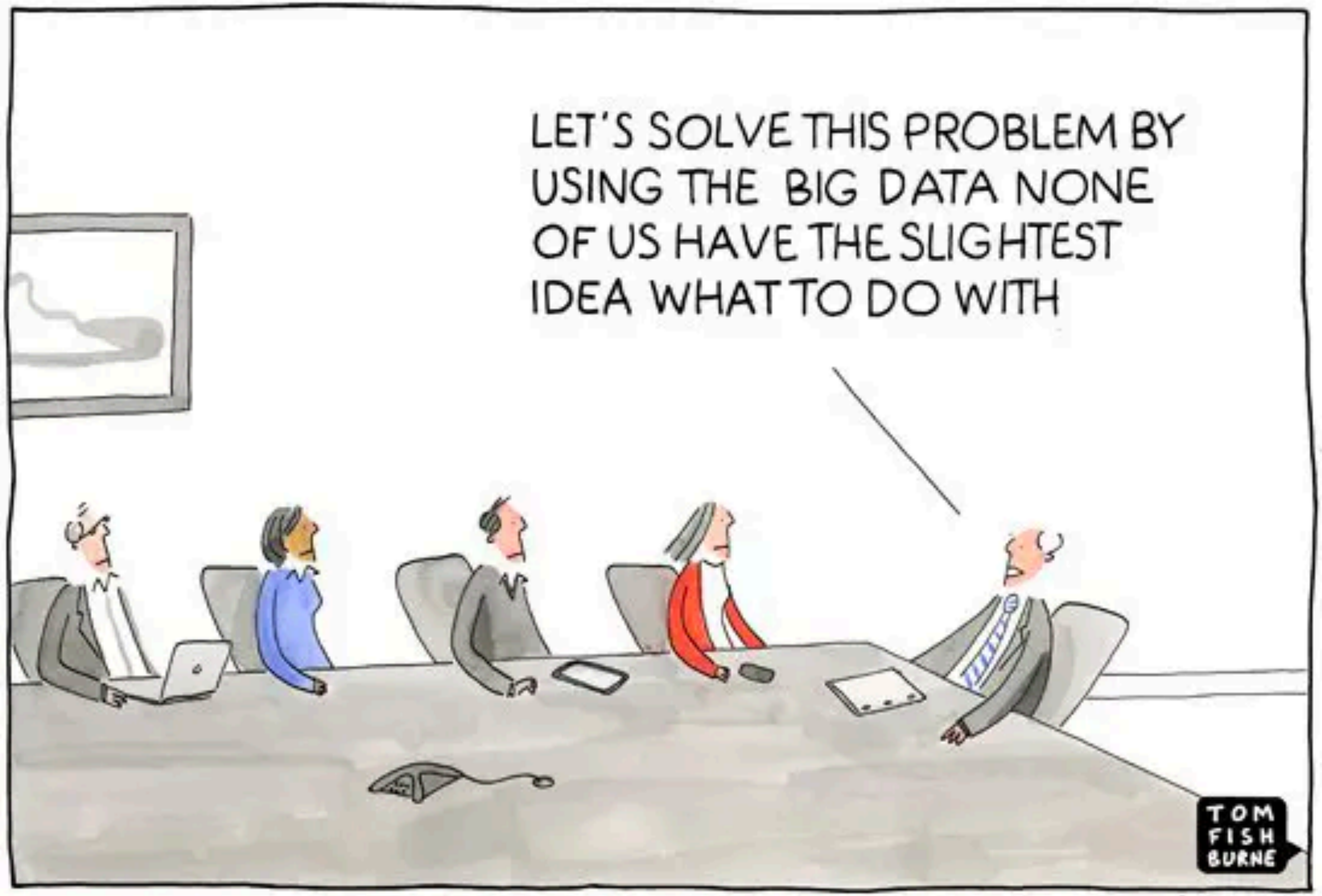

Mobile Platform Usage
100%
75%
50%
25%
Dec 2010
May 2012
Jan 2014
July 2015
Oct 2017
Aug 2019
Jun 2021
iOS
Android
Others

Be careful to avoid **false inferences**

We must be very careful with survey data due to **loss aversion**

Do you prefer that images in a web page be identified even if this results in redundancy?

Do you prefer that alternative text be repetitively duplicated or do you prefer that it be presented efficiently?


LET'S SOLVE THIS PROBLEM BY USING THE BIG DATA NONE OF US HAVE THE SLIGHTEST IDEA WHAT TO DO WITH
TOM FISH BURNE

There are two kinds of people in the world…

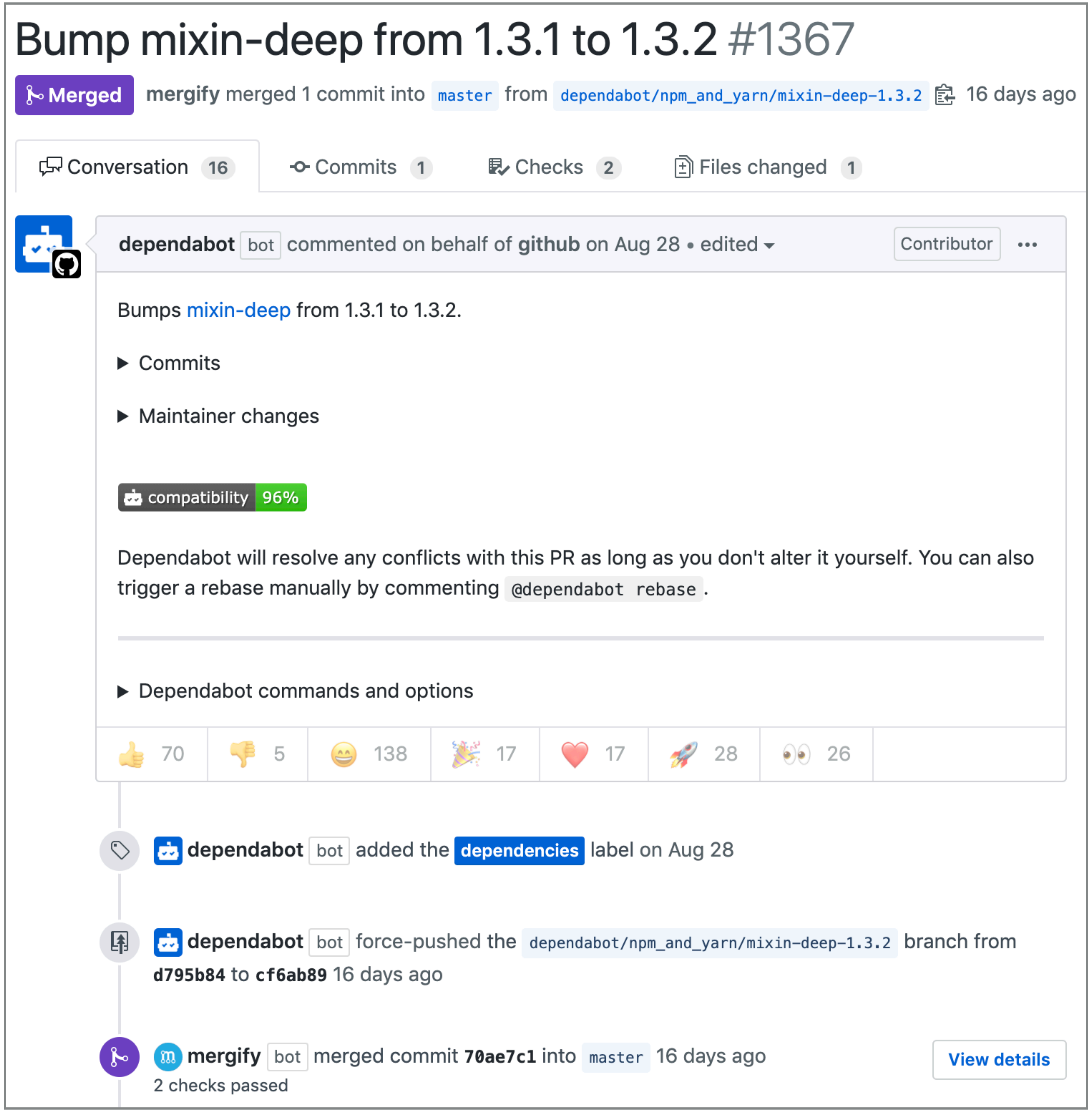
1. Those who can extrapolate from incomplete data.

Future possibilities

- Annual WebAIM Million+ and regular surveys provide longitudinal data and new insights.
- Define metrics for weighting errors to better determine user impact.
- Deeper data analytics
  - Patterns, correlations, and regression testing
  - Add manual testing data to identify auto-to-manual correlations

Machine Learning and Artificial Intelligence

- What are the trends? Why are errors occurring? Tools? Frameworks? 3rd party code?
- Where are successes occurring?
- We have millions of bad models, we need good models
- Create models of end user interactions to better detect end user barriers.
- Shifting accessibility **impact** measures to the end user device.



Bump mixin-deep from 1.3.1 to 1.3.2 #1367

Merged mergify merged 1 commit into `master` from `dependabot/npm_and_yarn/mixin-deep-1.3.2` 16 days ago

dependabot bot commented on behalf of **github** on Aug 28 • edited

Bumps mixin-deep from 1.3.1 to 1.3.2.

▶ Commits

▶ Maintainer changes

compatibility 96%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

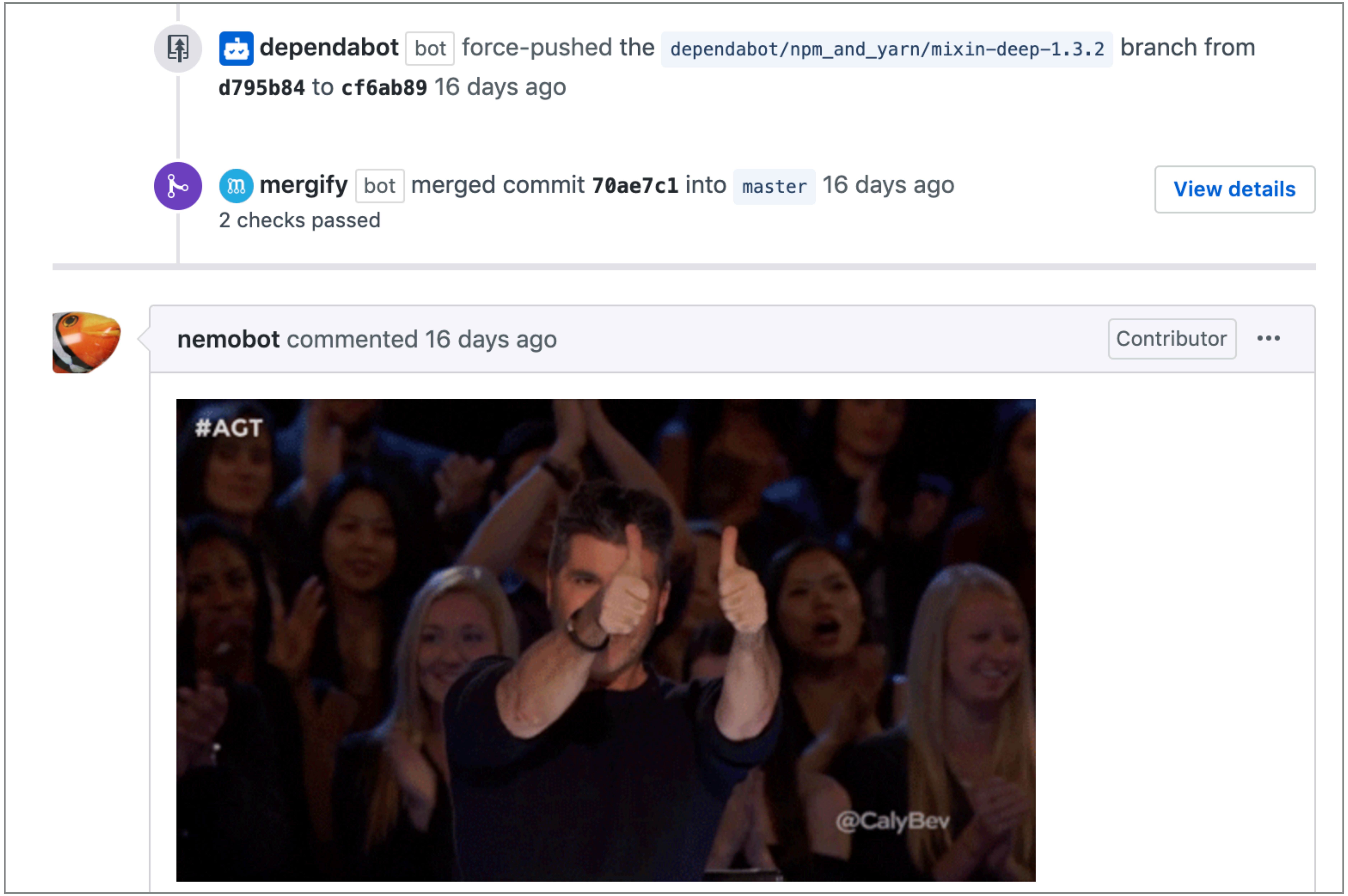
👍 70 👎 5 😄 138 🎉 17 ❤️ 17 🚀 28 👀 26

dependabot bot added the **dependencies** label on Aug 28

dependabot bot force-pushed the `dependabot/npm_and_yarn/mixin-deep-1.3.2` branch from **`d795b84`** to **`cf6ab89`** 16 days ago

mergify bot merged commit **`70ae7c1`** into `master` 16 days ago

2 checks passed



dependabot bot force-pushed the `dependabot/npm_and_yarn/mixin-deep-1.3.2` branch from `d795b84` to `cf6ab89` 16 days ago

mergify bot merged commit `70ae7c1` into `master` 16 days ago
2 checks passed

nemobot commented 16 days ago

Transition from:

what → **why**

Transition from:

what → **why**

descriptive → **prescriptive**

Transition from:

what → **why**

descriptive → **prescriptive**

detection → **prevention**

So…

what are you going to do about it?

Thank you!!!

Jared Smith
@jared_w_smith
WebAIM.org

WebAIM
web accessibility in mind