DRAFT 2.16.18 – NOT FOR DISTRIBUTION

Table of Contents

Overview/Methodology

- This report presents selected findings from the 2017 Node.js User Survey.
- The primary objective of the research was to profile Node.js users and identify potential areas of improvement. The findings will be used for program development, marketing and PR/external communications.
- The study was conducted online from Oct 5, 2017 to January 7, 2018 via a self-administered survey.
- The survey was fielded worldwide in English and Mandarin to encourage maximum response.
- The survey link was distributed by the Linux Foundation through a number of channels including email, Twitter, conferences, blogs and word of mouth (meet-ups).
- A total of **1,626** individuals responded to at least some questions in the survey.*
- To ensure data integrity and unbiased interpretation, data analysis and reporting was conducted by Research Collaborative, an independent market research firm.
- Where appropriate, statistically significant differences are noted as follows:
  - □/○ indicates a score is higher/lower than one or more other subgroup score(s) at 95% confidence level
  - ↑ / ↓ indicates a score is higher/lower than prior wave (2016) score at 95% confidence level (prior wave data not always shown)
- Additionally, significant differences may be noted in callout boxes:
- Numbers may not total to 100% due to rounding

*** Due to skip logic or missed questions, the number of responses for individual questions may be considerably lower**

Major Highlights

1. Node.js is continuing to have a positive impact on users particularly around developer productivity and satisfaction; when asked to describe Node.js, respondents use mostly positive terms like – “fast”, “easy”, “awesome”, “powerful”, “flexible” and even “fun”.
2. But this continued upbeat attitude doesn’t necessarily translate into community involvement. While more want to contribute than have in the past, still only a small minority say they are interested. Time and experience are the main barriers – but some also are intimidated, don’t know how to contribute or feel unwelcome.
3. Users may be more open to mentoring other users than contributing code: currently nearly half say they might be interested in mentoring others, significantly ahead of 2016. Full stack and back-end developers and those in Latin America and APAC are most likely to engage as mentors.
4. The coming year will likely see continued growth for Node, perhaps more outside US/CA than within: most users expect to increase their usage, particularly in Latin America and EMEA. Users expect to increase their use of other languages as well, including Rust, Go and JavaScript.
5. A differentiated marketing strategy by region may be called for given Node’s global presence and the varying needs and approaches by region. Among other things, there are key differences in business profile, deployment locations, language & resource usage, priorities (e.g., around package managers or LTS), educational needs and experiences, all of which may call for different approaches worldwide.

…continued on next page

Major Highlights (cont'd)

5. Although not widely used today, Go and Swift bear watching as they may be stealing the attention of some Node.js users: those who plan to maintain or decrease their Node usage are planning to increase their use of Go and Swift.
6. In the transpiler space, Babel is the leading tool, but use of Typescript seems to be on the rise outside of Latin America and among back-end, full stack and 'other' developers. Among Module Bundlers, Webpack seems to be consolidating its lead across most regions and development areas.
7. It is becoming increasingly important to users to manage different packages for multiple environments – but having access to multiple registries is not particularly important outside Latin America. NPM is by far the most widely used package manager, but Yarn is gaining in popularity in many segments.
8. There may be a need to address the clarity of the LTS schedule/support timeframe for Node.js: it is important to most users to have LTS for Node.js, but many are not clear about the LTS schedule/support timeframe for various versions. New node users are particularly likely to have issue with the clarity of the LTS timeframe – as are front-end developers, those in Asia Pacific region and smaller companies.
9. Recent improvements around education have registered with users as evidenced by improved scores for ease of learning Node.js, and for the availability and quality of resources. But more could be done, particularly in meeting the needs of some segments (APAC, EMEA and new users) and in some topic areas (Managing Node.js in Production and Node.js and Security). Documentation, free online courses and tutorial videos are the preferred channels and should be prominent in any education initiatives moving forward.

SECTION HIGHLIGHTS

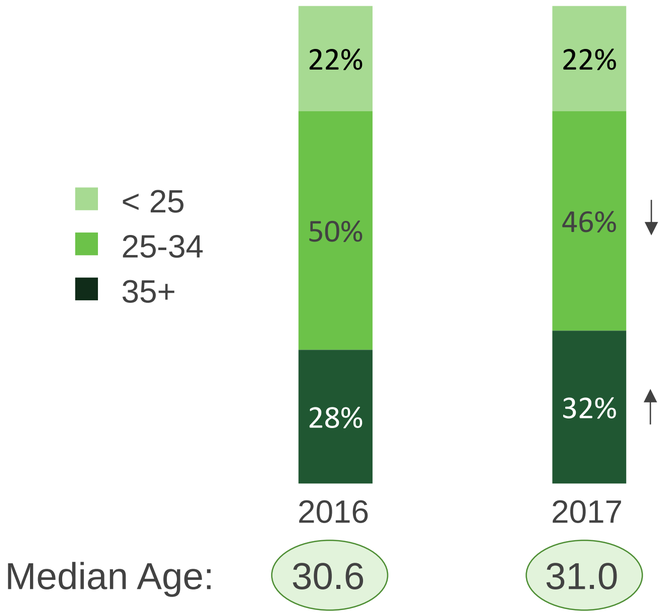
Business/Personal Profile

- As in 2016, the typical node.js user is male, age 31 and college educated.
- A majority are developers, in small (<100 employees) companies, with 5+ years of professional development experience.
- Although many have 10+ years total development experience, this year's respondents are less experienced in terms of total development experience.
- Respondents come from across the globe, but most are in US/CA or EMEA.
- Collectively, respondents speak over 60 languages, but for nearly half English is primary.
- The mix of countries has changed somewhat since 2016 – with fewer from US and China and more from India and Canada.
- There are considerable differences in personal and business characteristics, with those from US/CA older, more experienced, and from larger companies than users elsewhere.

Personal Characteristics

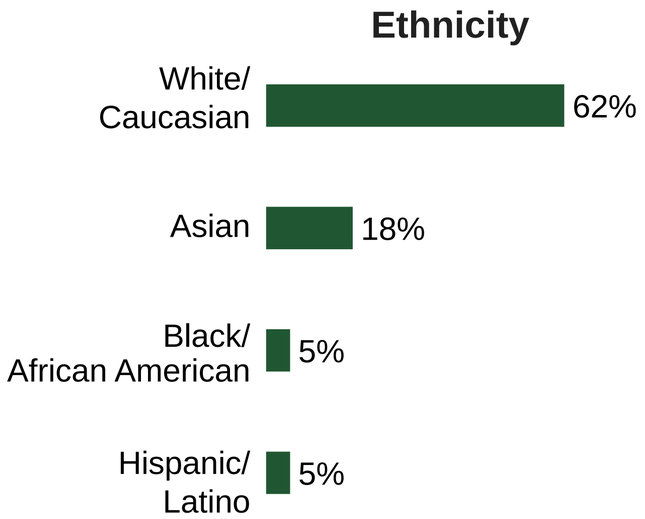
- The typical Node.js user is male, age 31, college educated and white.
- Respondents are slightly older in this wave, but gender and education are unchanged.

Respondent Age



	2016	2017
Gender	**%**	**%**
Male	94	94
Female	5	5
Other	1	1
Education	**%**	**%**
High school or less	15	16
Some college	11	10
College degree	51	51
Grad degree	21	21
Other	2	2

Ethnicity



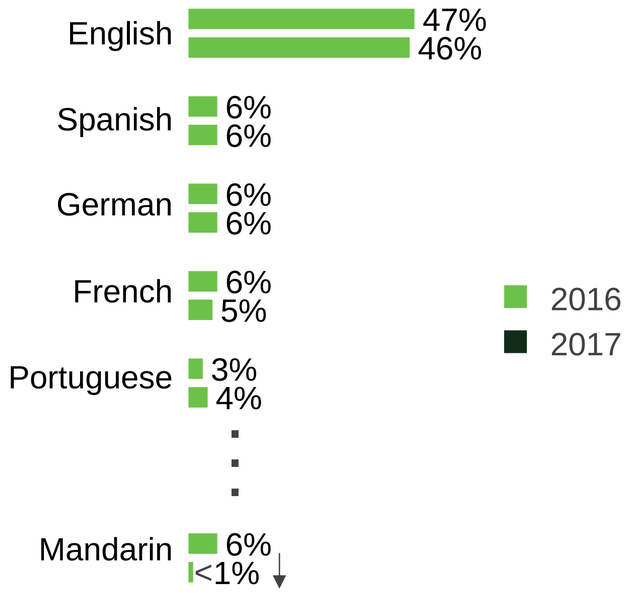
SOURCE: Q9, Q10, Q11, Q14

* Ethnicity asked as open end in 2017, closed end in 2016; not trendable

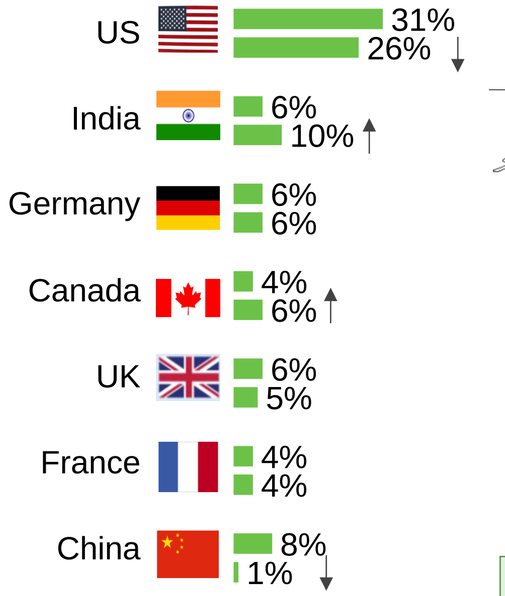
International Presence

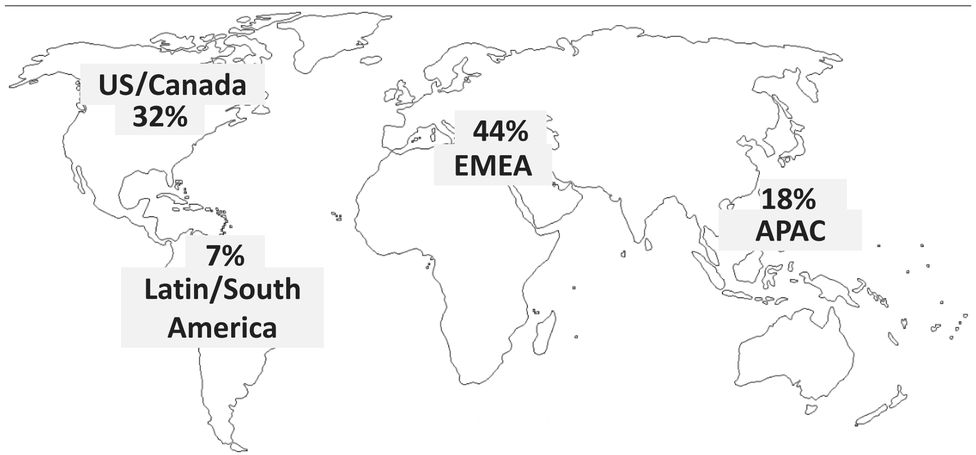
- As in the prior study, respondents come from across the globe, but the mix of countries has changed slightly with fewer respondents from the US and China, and more from India and Canada.
- Fewer than half consider English their primary language.

Primary Language Spoken
(Top Mentions)



Where Reside
(Top Mentions)



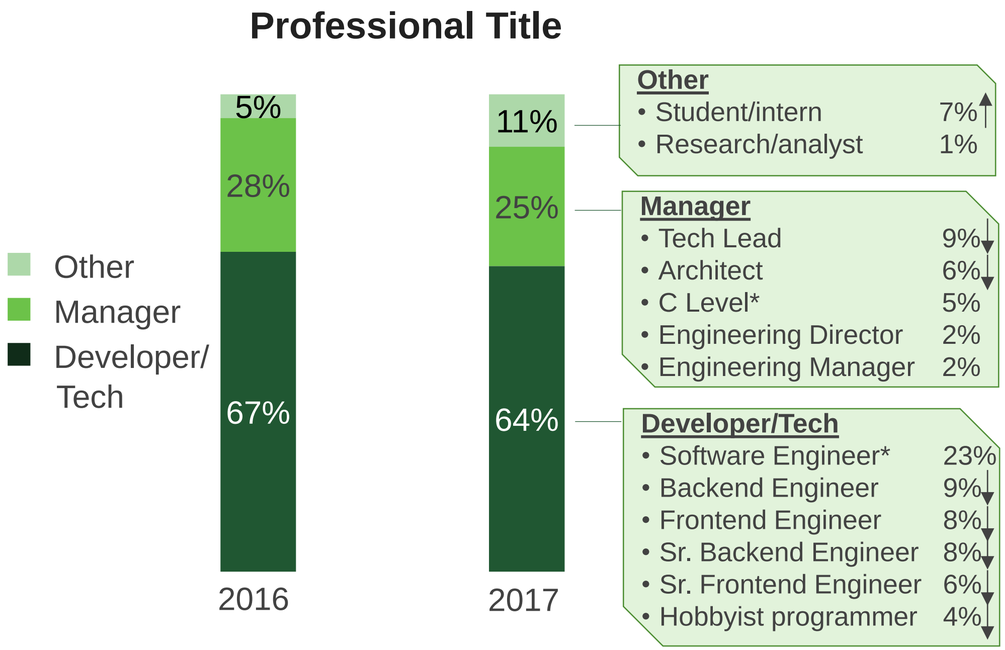


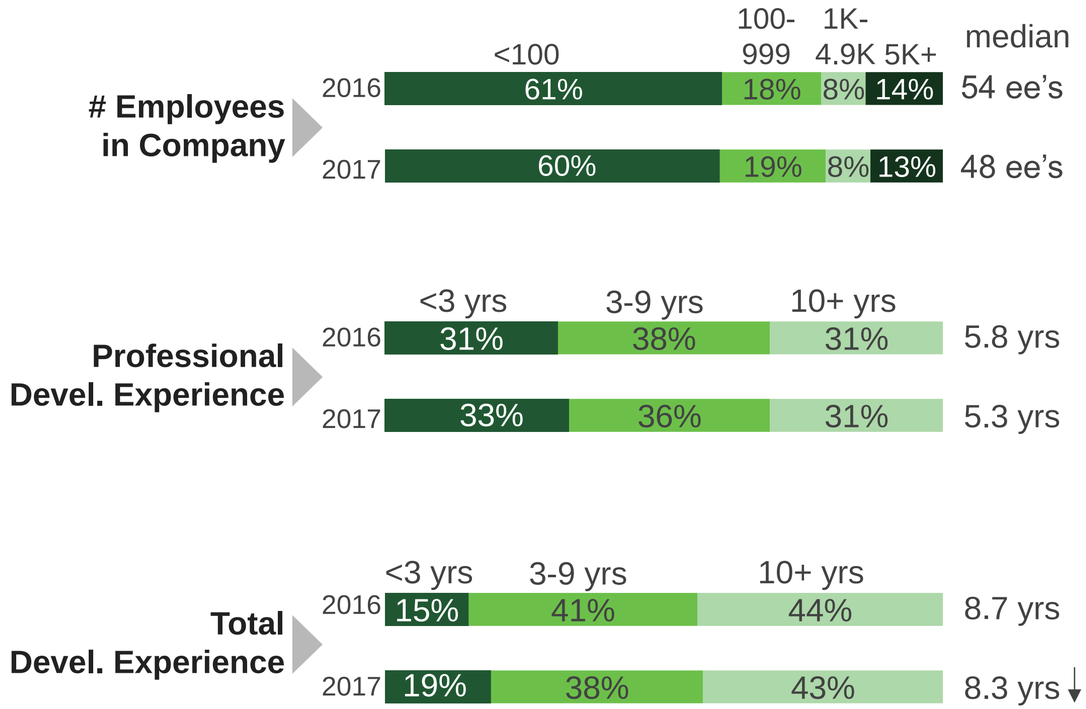
Collectively, respondents span **100+** countries and speak at least **60** languages

SOURCE: Q4, Q7

Professional Profile

- The typical respondent is a developer in a small (<100 employees) firm, with 5 years professional experience,
- Although many have 10+ years total development experience, respondents are somewhat less experienced overall in this wave.





SOURCE: Q1, Q3, Q15, Q16

*Items pre-listed in 2017, but not in 2016; not trendable

Profile by Region

- There are notable differences in business and personal profile by region.
- Those in the US/CA are older, more experienced and work in larger companies than their peers around the globe.
- EMEA respondents are particularly highly educated.
- The profile of APAC respondents has changed in many respects vs. last wave – perhaps not surprising given the drop in China as a percentage of APAC respondents this year. APAC respondents are relatively new to development.

Business Characteristics
(By Region)

	US/CA	EMEA	APAC	LatAm
Co size (median # ee's)	99	24	24	40
Prof'l dev experience (median yrs)	7.5	5.3	3.7	4.4
Prof'l dev experience (10+ yrs)	42%	29%	18%	28%
Developers	60%	67%	64%	70%
Managers	29%	23%	24%	24%

Personal Characteristics
(By Region)

	US/CA	EMEA	APAC	LatAm
English primary language	92%	21%	42%	1%
Age (median)	33	31	29	30
Age (% 35+)	41%	31%	20%	24%
Male	91%	95%	97%	96%
Have grad degree	15%	28%	20%	8%

SOURCE: Q1, Q3, Q7, Q9, Q14. Q15

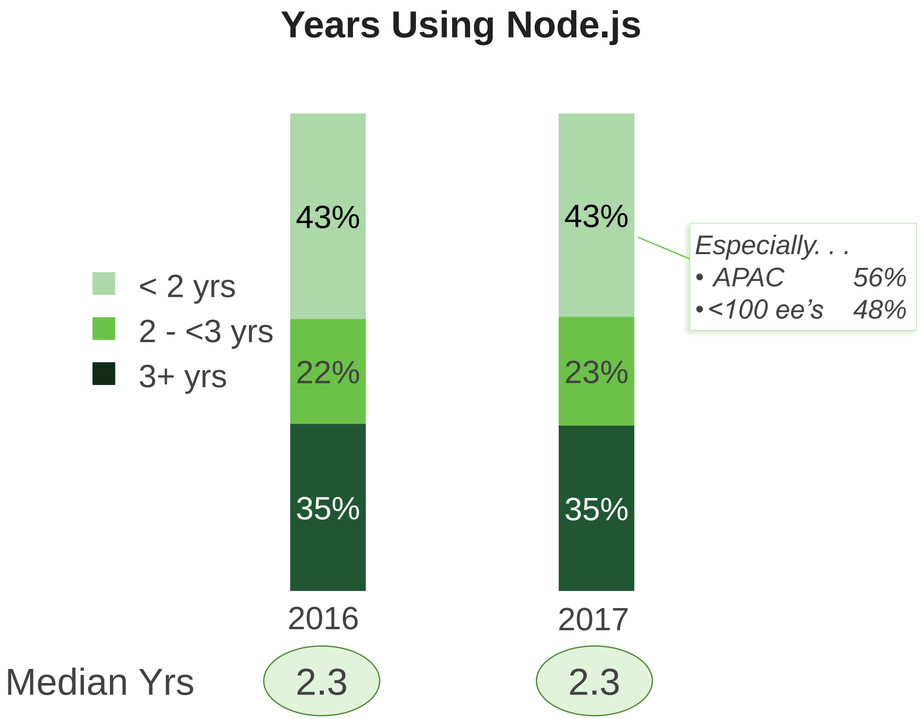
Node.js Usage Profile

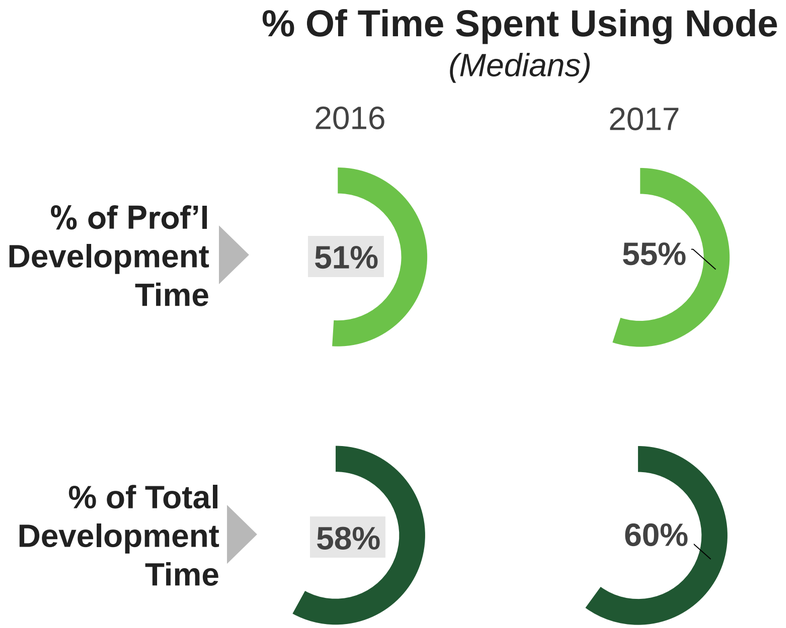
- As in 2016, the typical user has been working with Node over 2 years, and spends more than half of his development time using it. The vast majority are developing web apps.
- Most respondents are primarily back-end or full stack development-focused, and they use Node.js more regularly at work than others do.
- Relatively few users are primarily focused on “other” non-traditional development areas*, but those who are tend to be older and more highly educated than others.
- Users are deploying through a range of channels, but AWS is most widely used for production and On-premise or AWS for development. Heroku seems to be growing in popularity in APAC and Latin America.
- More than 4 in 5 back-end and full stack developers are using node.js frameworks; Express is tops, but Graph QL is increasingly prevalent this wave.
- Most are using a transpiler and module bundler (especially full stack and front-end developers). Babel is the preferred transpiler, but Typescript is growing. Webpack continues to dominate the module bundler space.
- Ubuntu is the primary OS/Distro used in production, and MAC OS in development – but Windows seems to be growing in popularity for both (especially in US/CA and EMEA).

*Including DevOps, Desktop Apps, Mobile, IoT and Security

Experience with Node.js

- There has been no change in the Node.js usage profile since last wave.
- The typical respondent has been using Node for just over 2 years, and spends more than half of his development time with Node.





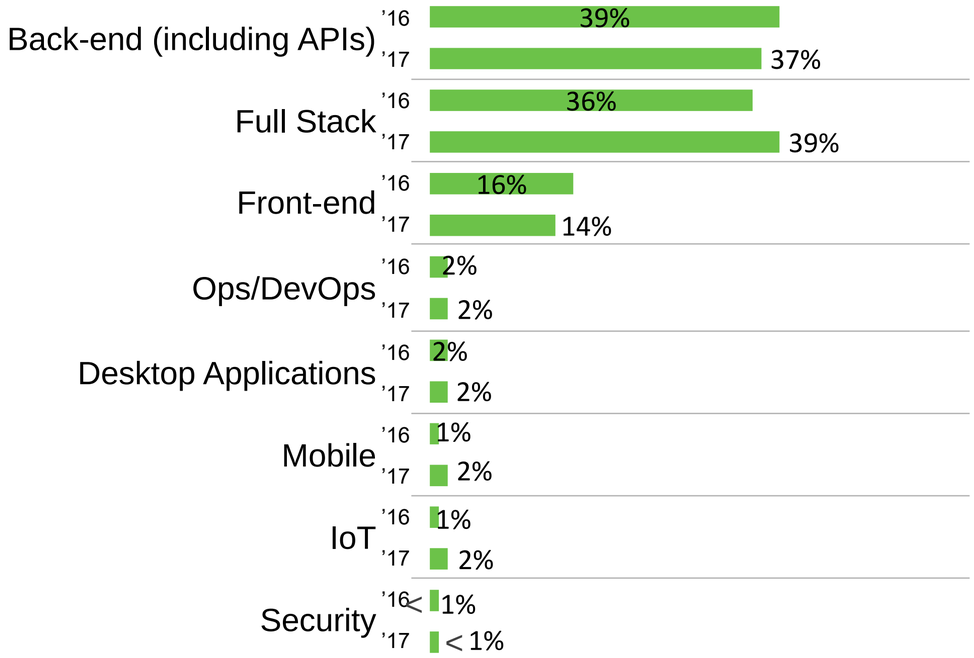
SOURCE: Q17, Q20, Q21

Development Focus

Only primary, not net

- Three in four Node.js users are focused primarily on back-end or full stack development.
- There has been a slight drop in this wave in those who have any focus on back-end development.
- US/CA respondents are more focused on full stack than back-end.
- Ops/DevOps is really only a focus in US/CA – although even here, it is not widely focused on.

Primary Development Focus



Primary Development Focus by Region

	US/CA	EMEA	APAC	LatAm
Back-end	32%	38%	43%	42%
Full Stack	42	39	38	41
Front-end	15	14	13	11
Ops/DevOps	3	1	<1	1
Desktop Apps	2	2	2	1
Mobile	2	2	2	2
IoT	1	2	1	2
Security	<1	1	<1	0

Profile by Development Area

- There are some differences in users' business and personal characteristics based on development focus.
- Full stack developers have been using Node the longest, and along with back-end developers, spend the mos
- Those outside of the three traditional development areas tend to be older and more highly educated than othe

Consider including meaning of the down arrows, up arrows, circles and squares as mouse overs

Business Characteristics
(By Primary Development Focus)

	Back-end	Full Stack	Front-end	Other
Co size (median # ee's)	54	37	79	44
Prof'l dev experience (median yrs)	4.7 ↓	6.0	5.6	6.1
Prof'l dev experience (10+ yrs)	31	32	29	38
Developers	69%	61%	71%	50% ↓
Managers	25% ↓	28%	19%	32%
Years using Node.js (median)	2.2	2.5	2.3	2.2
% Prof Dev time Use Node.js (median)	61% ↑	62%	27%	39%

Personal Character
(By Primary Development Focus)

	Back-end	Full Stack	Front-end	Other
English primary language	43%	48%	44%	48%
Age (median)	31	31	31	35 ↑
Male	94%	95%	95%	91%
Have grad degree	22%	19%	18%	29%
US/CA	27% ↓	34%	35%	39%
EMEA	46%	43%	45%	45%
APAC	19%	16%	15%	10%
Latin America	8%	8%	6%	4%

"Other" includes: Ops/Dev Ops, Desktop Applications, Mobile, IoT and Security

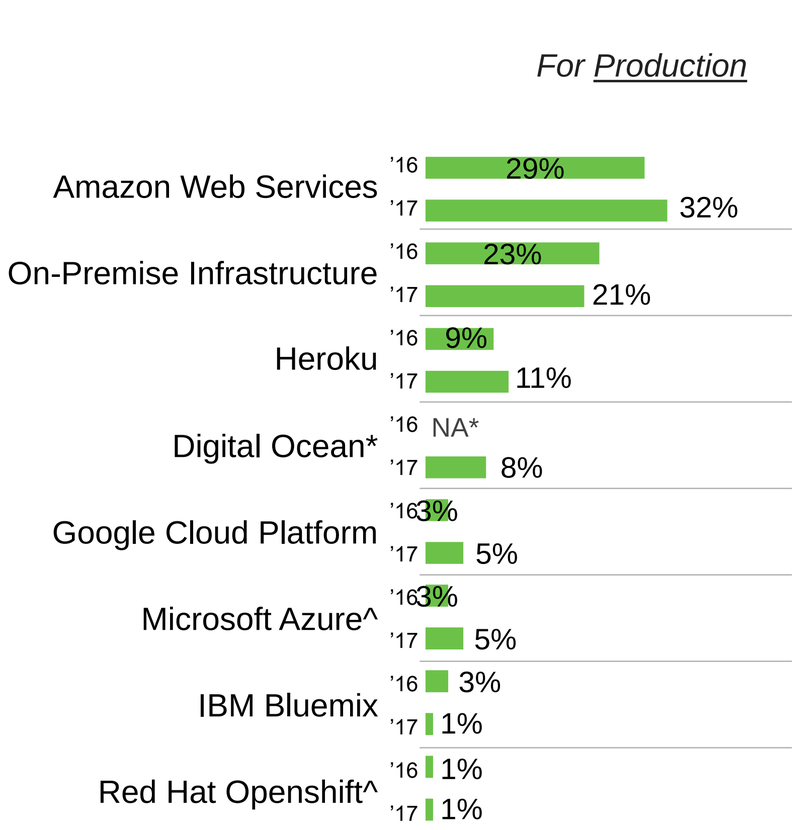
SOURCE: Q1, Q3, Q4, Q7, Q9, Q10, Q15, Q17, Q20

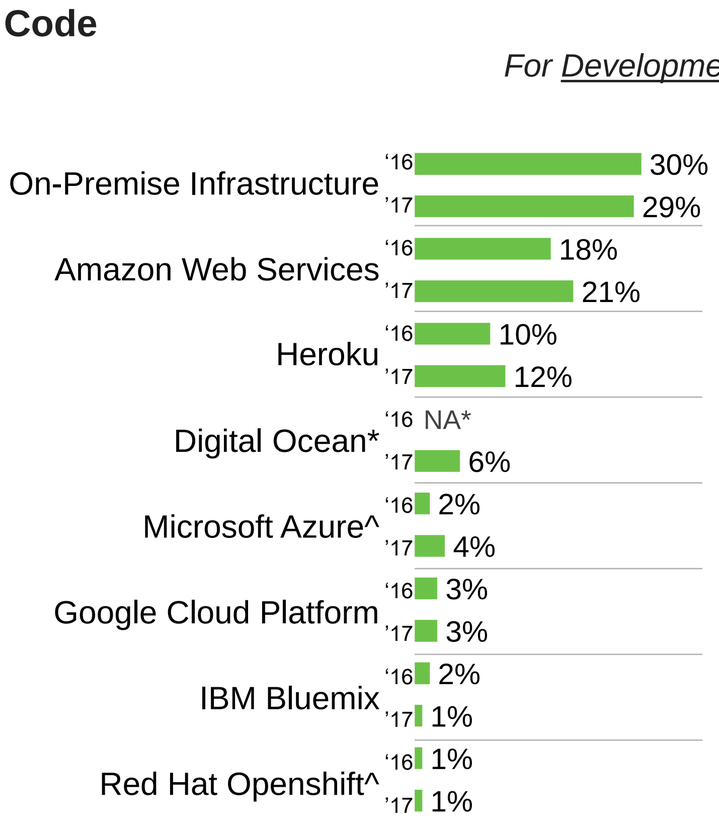
Where Deploy Code

Only Primary, include callouts as mouse overs

- AWS is the primary place where respondents deploy code for production, and it seems to be growing for use in develo
- On-Premise infrastructure is also widely used, but has dropped for use in production since 2016.

Where Deploy Node.js Code





SOURCE: Q28, Q29, Q30

^ Attribute wording change; interpret trend cautiously

* Items added to pre-list in 2017; not trended

Where Deploy Code

- EMEA respondents are less likely than others to use AWS, preferring on-premise infrastructure. US/CA respo also likely to be deploying via on-premise infrastructure.
- Heroku is growing in both APAC and Latin America, and is one of the top choices for deployment for developm America.

Consider including meaning of the down arrows, up arrows, circles and squares as mouse overs

Where Primarily Deploy Node.js Code

(By Region)

For Production

	US/CA	EMEA	APAC	LatAm
Amazon Web Services	37%	23%	42%↑	40%
On-Premise Infrastructure	20	27	11↓	12
Heroku	11	11	12↑	16↑
Digital Ocean*	7	9	7	7
Google Cloud	4	5	5	6
Microsoft Azure^	5	4	7↑	5
IBM Bluemix	2	1	1	3
Red Hat Openshift^	1	1	2	1
Depl not req'd	5	7	4↓	4

For Development

	US/CA	EMEA	APAC	LatAm
On-Premise Infrastructure	32%	32%	20%	22%
Amazon Web Services	24	17	27↑	20
Heroku	9	12	16↑	20↑
Digital Ocean*	5	6	7	5
Google Cloud	3	4↑	4	1
Microsoft Azure^	3	4↑	7↑	4
IBM Bluemix	1↓	1	1	4
Red Hat Openshift^	1	1	1	1
Depl not req'd	16	16↓	14↓	13

SOURCE: Q28, Q30 Trended to 2016, but 2016 data not shown ^ Attribute wording change; interpret trend cautiously * Items added to pre-list in 2017; not trended

Where Deploy Code

- Deployment also varies somewhat based on development focus.
- AWS is widely used by back-end, full stack and front-end developers, but less so for others.
- Heroku is relatively popular among full-stack developers.

Consider including meaning of the down arrows, up arrows, circles and squares as mouse overs

Where Primarily Deploy Node.js Code

(By Primary Development Focus)

For Production

	Back-end	Full Stack	Front-end	Other
Amazon Web Services	36%	33%	29%	16%
On-Premise Infrastructure	23	20	24	20
Heroku	10	14	9	8
Digital Ocean*	5	11	8	5
Google Cloud	5	5	4	5
Microsoft Azure^	5	4	4	9 ↑
IBM Bluemix	2	1	0	3
Red Hat Openshift^	1	1	1	2
Depl not req'd	4	3	13	13

For Development

	Back-end	Full Stack	Front-end	Other
On-Premise Infrastructure	29%	29%	33% ↑	25%
Amazon Web Services	24	22	17	10
Heroku	12	14 ↑	9	7
Digital Ocean*	3	8	7	4
Google Cloud	3	4	2	4
Microsoft Azure^	5 ↑	3	2	7 ↑
IBM Bluemix	1	1	1	2
Red Hat Openshift^	1	1	1	2
Depl not req'd	15	12 ↓	20 ↓	27

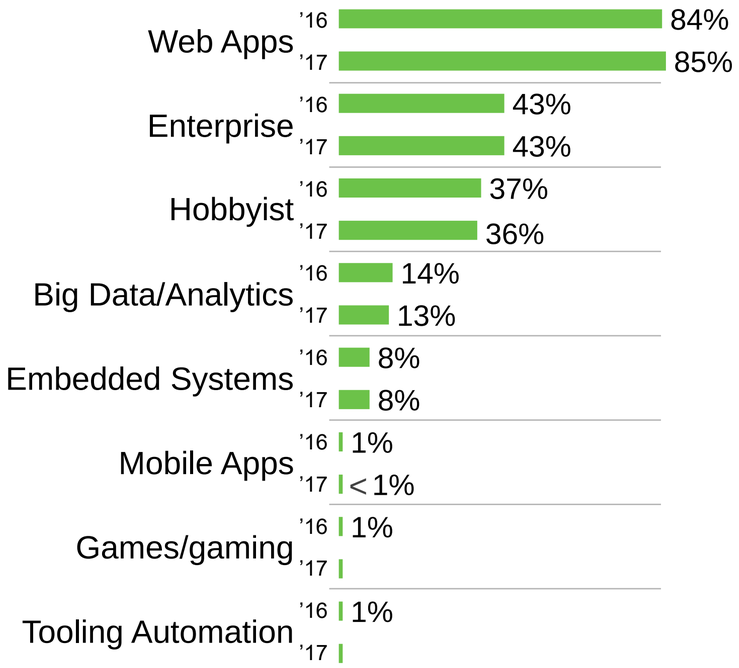
SOURCE: Q28, Q30 Trended to 2016, but 2016 data not shown ^ Attribute wording change; interpret trend cautiously * Items added to pre-list in 2017; not trended

Types of Development Work

- As before, the vast majority of respondents are spending time developing web apps, particularly those in full stack or front end positions.
- A notable minority also engage in enterprise and/or hobbyist work.
- Those outside the traditional development areas are more likely to be working on embedded systems.

Consider including meaning of the down arrows, up arrows, circles and squares as mouse overs

Types of Development Work Spend Time On
(Top Mentions)



Type of Work
By Primary Development Focus

	Back-End	Full Stack	Front-End	Other
Web Apps	82%	92%	89%	66%
Enterprise	47	43	35	40
Hobbyist	32	41	35	37
Big Data/Analytics	13	14	7	13
Embedded Systems	8	7	3	19
Hobbyist ONLY	2	2	3	10

SOURCE: Q24

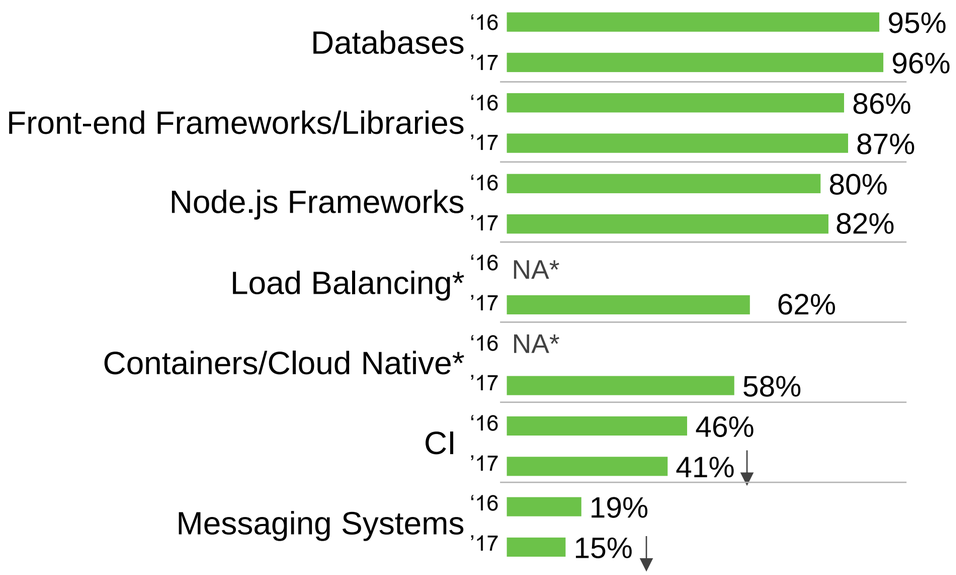
Tools/Technologies Used

- Respondents are using a range of tools with Node.js including, primarily: databases, libraries and Node.js fran
- There is considerable variation based on primary development focus – with back-end and full stack developer to use a range of tools.
- Messaging systems and CI are less commonly used than other tools – and usage has dropped since 2016.

Hold off until I get you the cluster graphs

Types of Tools/Technologies Used with Node.js

(in Past 12 Months)



Type of Tools By Primary Development Focus

	Back-End	Full Stack	Front-End	Other
Databases	98%	97%	89%	92%
Front-end Frameworks	83	93	94	69
Node.js Frameworks	84	90	67	59
Load Balancing*	65	68	48	44
Containers/Cloud*	62	61	48	45
CI	40	45	40	27
Messaging	21	15	4	12

SOURCE: Q27 (net categories)

* Individual technologies within nets changed in 2017; not trended

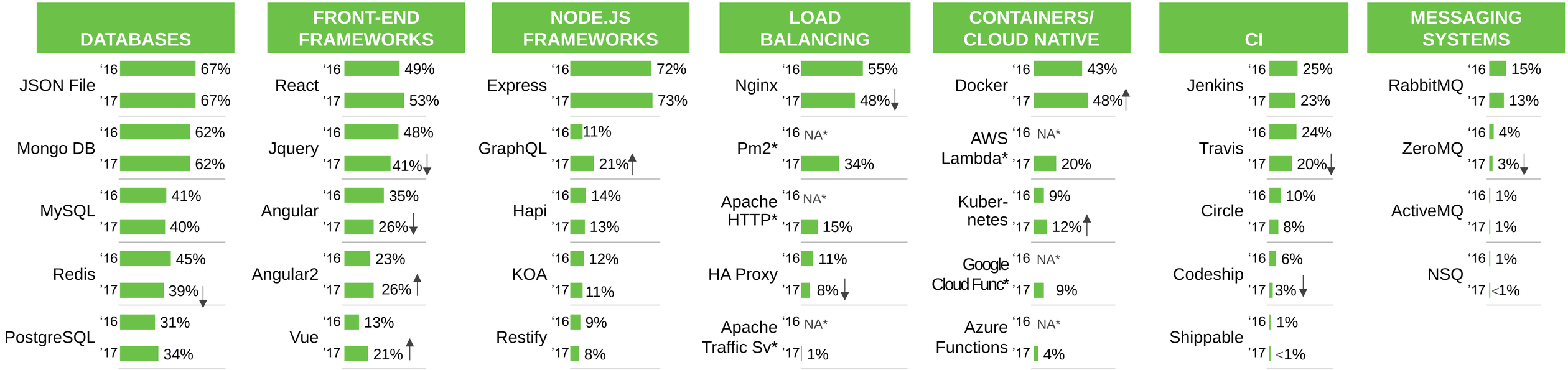
Tools/Technologies Used

- Respondents are using a range of tools within each category, although in most categories, one or two tools do
- There has been a drop in usage for many specific tools – but some have increased since 2016, including Ang GraphQL, Docker and Kubernetes.

Hold off until I get you the cluster graphs

Specific Tools/Technologies Used with Node.js

(in Past 12 Months)



SOURCE: Q27

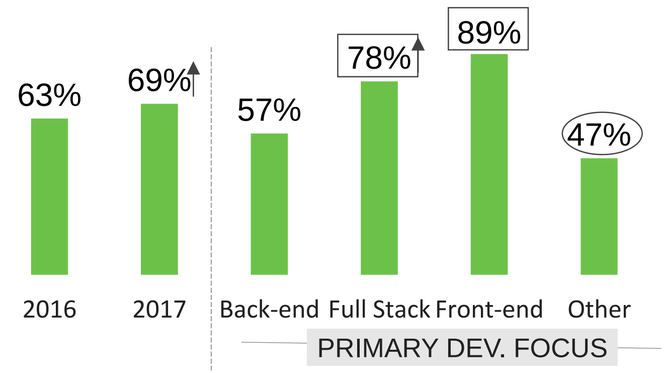
*** New item in 2017; not trended**

Transpilers and Bundlers

- Most respondents use a transpiler and module bundler – particularly front-end and full stack developers.
- Use of transpilers has risen since 2016; Babel is most common, but use of Typescript is on the rise.
- Webpack seems to be solidifying its notable lead among module bundlers.

Consider including meaning of the down arrows, up arrows, circles and squares as mouse overs

Use a Transpiler



***Typescript** growing across most regions . . .*

US/CA	EMEA	APAC	LatAm
18% ↑	24% ↑	22% ↑	14%

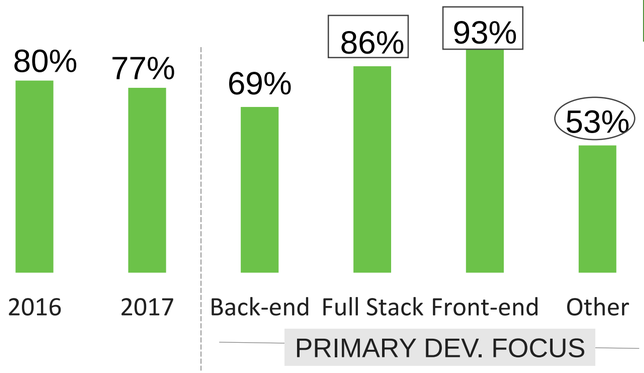
. . .And within most development areas

BckEnd	FullStk	FrEnd	Other
19% ↑	21% ↑	24%	23% ↑

Transpiler Used
(Among all respondents)

	2016	2017
• Babel	49%	46%
• TypeScript	12%	21% ↑
• Traceur	<1%	<1%

Use A Module Bundler



***Webpack** growing across most regions. . .*

US/CA	EMEA	APAC	LatAm
52% ↑	53% ↑	50%	51% ↑

. . .And within most development areas

BckEnd	FullStk	FrEnd	Other
41% ↑	60% ↑	70% ↑	31%

Module Bundler Used
(Among all respondents)

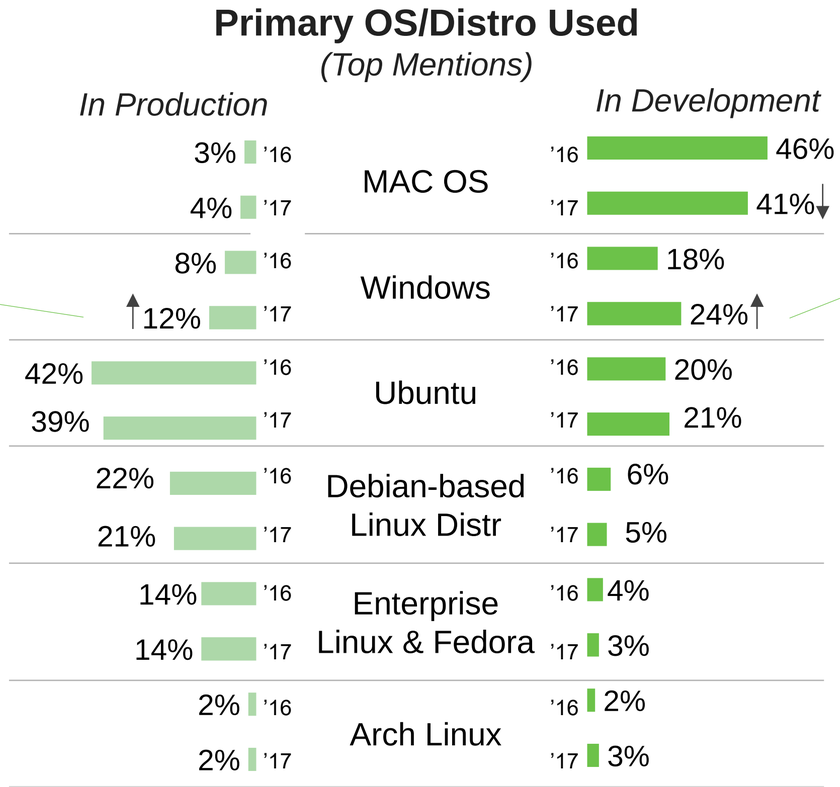
	2016	2017
• Webpack	39%	52% ↑
• Gulp	19%	11% ↓
• Grunt	9%	5% ↓
• Browserify	7%	5% ↓
• Rollup	1%	2% ↑

SOURCE: Q31, Q32

Primary OS/Distro

- Ubuntu is the primary OS/Distro used in production, while MAC OS is primary in development
- Use of Windows – in both production and development – has increased since 2016, particularly in US/CA, EMEA and in smaller companies (among other segments).

The rise in use of Windows is attributable to rises among. . .
- "Other" Dev Focus
- US/CA
- EMEA
- < 100 ee's



The rise in use of Windows is attributable to rises among. . .
- Back End Dev's
- "Other" Dev Focus
- US/CA
- EMEA
- < 100 ee's
- 1000+ ee's

SOURCE: Q41, Q42

Primary OS/Distro

- Distro use varies somewhat by region, with Ubuntu more popular in APAC and Latin America, and MAC OS more popular in US/CA.
- Debian-based Linux, while not widely used anywhere for development, is somewhat popular in EMEA and Latin America in production

Primary OS/Distro Used
(By Region)

In Production					*In Development*			
US/CA	**EMEA**	**APAC**	**LatAm**		**US/CA**	**EMEA**	**APAC**	**LatAm**
6%	3%	3%	3%	MAC OS	50% ↓	35% ↓	40%	33%
12 ↑	12 ↑	14	3	Windows	22 ↑	27 ↑	23	15
37	37	45	41	Ubuntu	15	21	27	37
14	26	16	28	Debian-based Linux	4	6	4	5
16	11	15	17	Ent. Linux & Fedora	4	3	3	3
3	3	2	1	Arch Linux	2	4	3	4

SOURCE: Q41, Q42

Primary OS/Distro

- Primary distro varies somewhat by development focus.
- Ubuntu is most popular among back-end and full stack developers, while Windows is more popular among front-end and “other” developers (where it is the #1 choice for both production and development).

Primary OS/Distro Used

(By Primary Development Focus)

In Production					*In Development*			
Back-End	**Full Stack**	**Front-End**	**Other**		**Back-End**	**Full Stack**	**Front-End**	**Other**
3%	3%	8%	6%	MAC OS	36%	47%	48%	26%
8	9	18	29	Windows	21	21	29	39
42	41	30	25	Ubuntu	26	20	14	20
19	24	19	15	Debian-based Linux	5	5	2	6
17	12	14	9	Ent. Linux & Fedora	6	3	1	0
3	2	3	2	Arch Linux	4	2	4	3

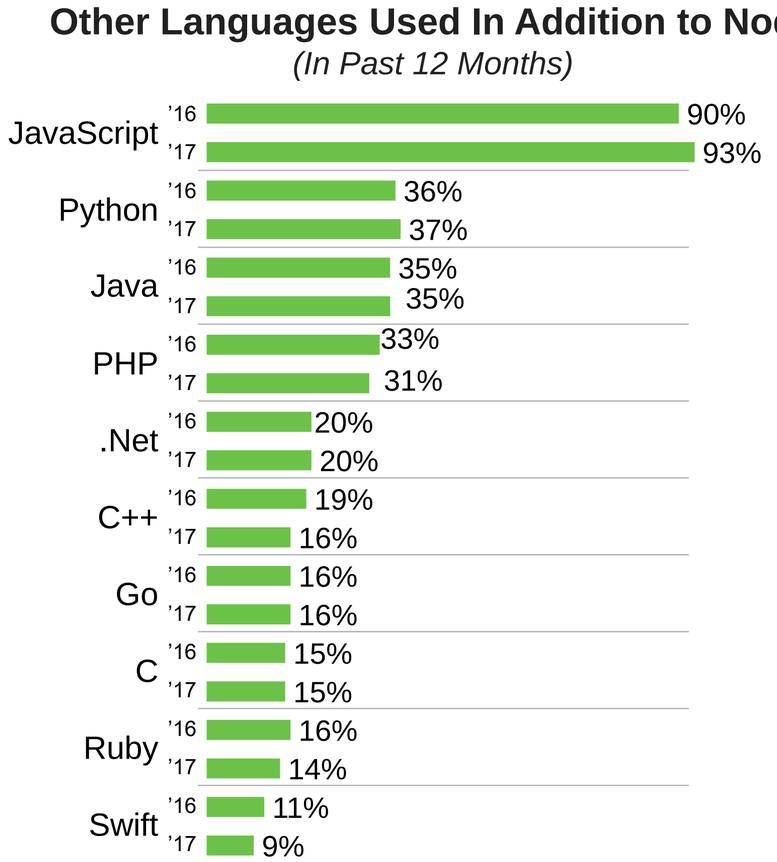
SOURCE: Q41, Q42

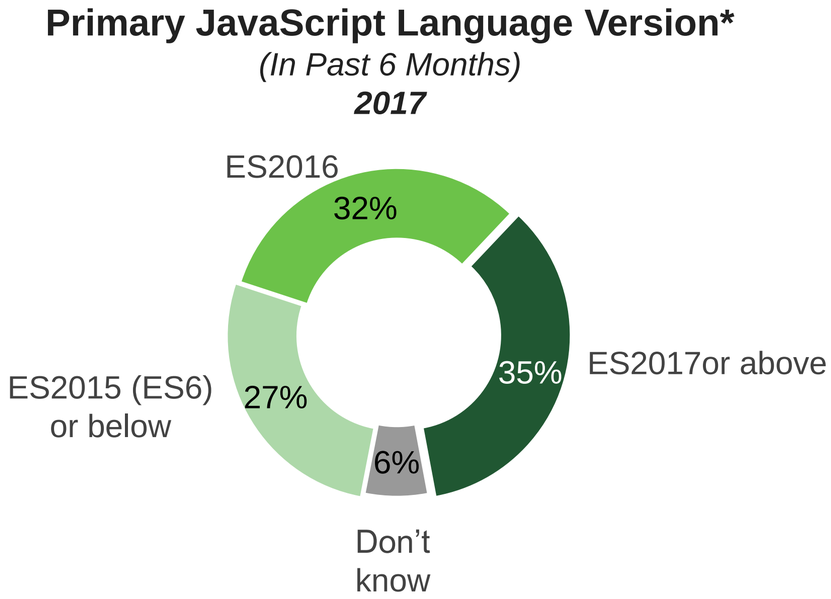
Languages Used

- Node.js users are using a range of other languages besides Node – more than 3 on average, including primarily JavaScript, Python, Java and PHP.
- A third are using ES2017 or above – three times as many as in 2016.
- APAC users use fewer languages on average than others, while those outside traditional development areas use more – particularly C++ and C.
- Most expect to increase their use of Node.js over the next 12 months – and the number is rising. Growth will likely come from outside the US/CA – particularly in Latin America or EMEA.
- Use of other languages is also expected to increase – including Rust, Go and JavaScript.
- Usage of Ruby has dropped, and users are far more likely to say they will "decrease" usage than increase over the next 12 months.
- PHP is less popular in US/CA and among "other" developers; and, many of those who use it say they will decrease usage over the next 12 months.
- Go and Swift may be stealing the attention of Node.js users – many of those who plan to maintain/decrease with Node.js will increase with Go or Swift in the next 12 months.

Languages Used

- Respondents are using at least 3 other languages on average besides Node – typically JavaScript and then Python, Java or PHP.
- More than a third of respondents are using ES2017 or above – up significantly over 2016.





Languages Used

- Language usage varies somewhat by region and development focus.
- Among other differences, PHP is less popular in US/CA and among those outside of traditional programming areas.
- Despite being less likely to use JavaScript and PHP, those “other” developers use more languages on average (closer to 4), including C++ and C.

Other Languages Used In Addition to Node

(In Past 12 Months)

Top Mentions

By Region

	US/CA	EMEA	APAC	LatAm
JavaScript	93%	93%	89%	96%
Python	39	35	35	39
Java	34	36	30	48
PHP	23	37	29	38
.Net	22	20	19 ↑	19
C++	20	16 ↓	13	8 ↓
Go	18	14	15	20
C	16	13	15	15
Ruby	20	12 ↓	8	11
Average #	3.3	3.2	2.8	3.3

By Primary Development Focus

	Back-End	Full Stack	Front-End	Other
JavaScript	92%	95%	94%	84%
Python	36	38	33	41
Java	36	34	35	39 ↑
PHP	31	35	29	21
.Net	21	19	21	26
C++	15 ↓	15 ↓	13	33
Go	17	17	11	15
C	15	13 ↓	10 ↑	28
Ruby	12 ↓	14	14	16
Average #	3.2	3.2	2.9	3.6

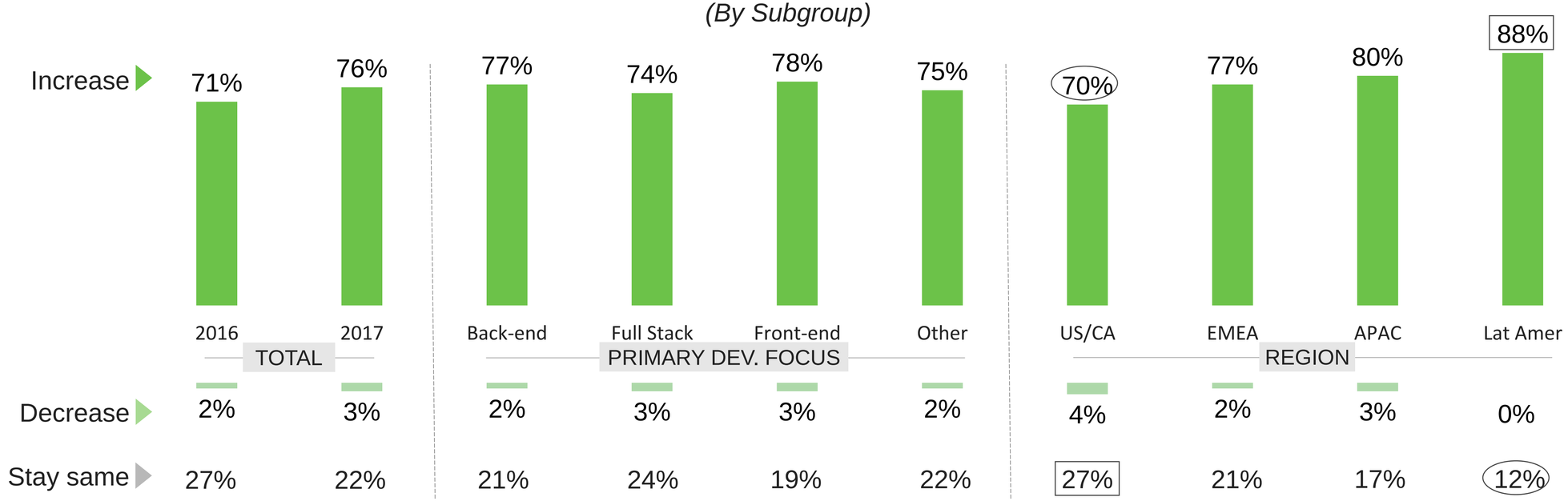
SOURCE: Q25

Expected Change In Node Usage

- Three quarters of respondents say they plan to increase their use of Node over the next 12 months – up from 2016.
- The rise in increased usage is attributable to back-end developers and those in EMEA, although Latin American respondents are most likely to say they will increase their usage.

Expected Change In Use of Node.Js Over Next 12 Months

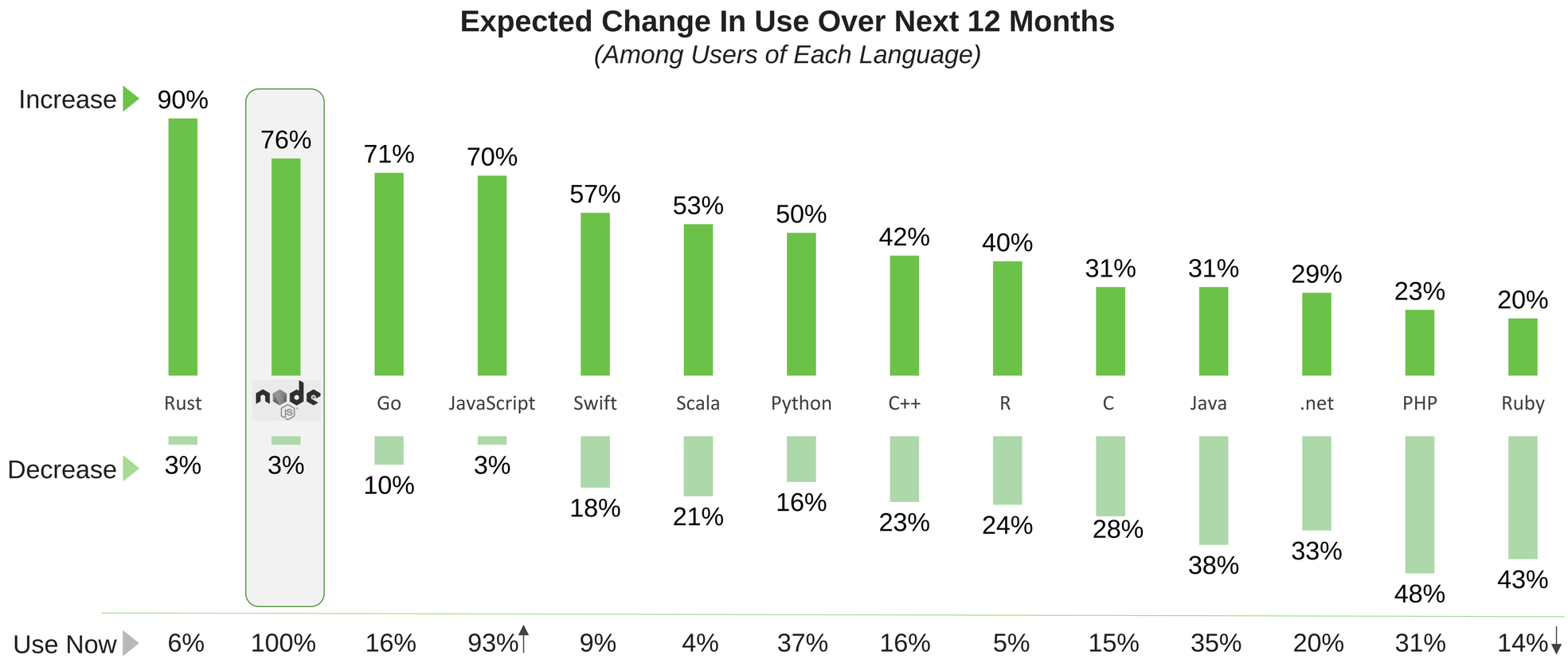
(By Subgroup)



SOURCE: Q26, among Node.js users who provided an answer

Expected Change In Other Languages

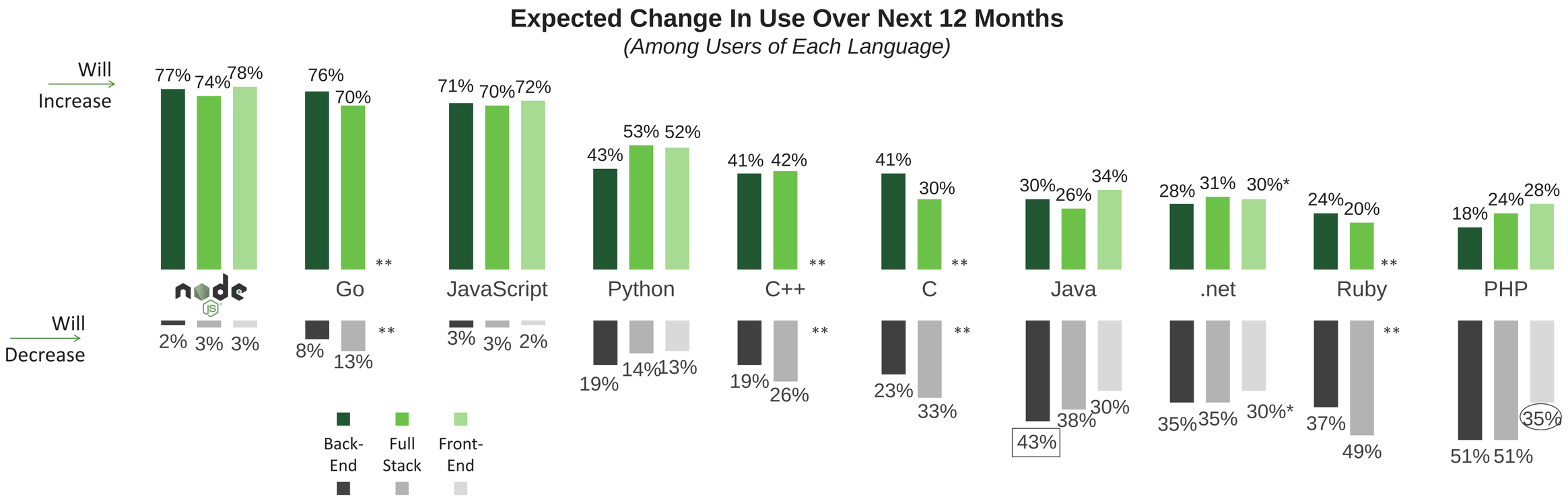
- Use of other languages will also increase among current users – including Rust, Go, and JavaScript.
- Usage of PHP and Ruby appears to be on the decline – although many of those in Asia/Pacific plan to increase their usage of PHP.



SOURCE: Q25, Q26, among those who use respective brand and who provided an answer

Expected Change In Other Languages

- Planned usage does not vary considerably across the three main developer segments – except that back-end developers are particularly likely to decrease their use of Java and front-end developers are less likely to decrease their use of PHP.



SOURCE: Q25, Q26, among those who use respective brand and who provided an answer

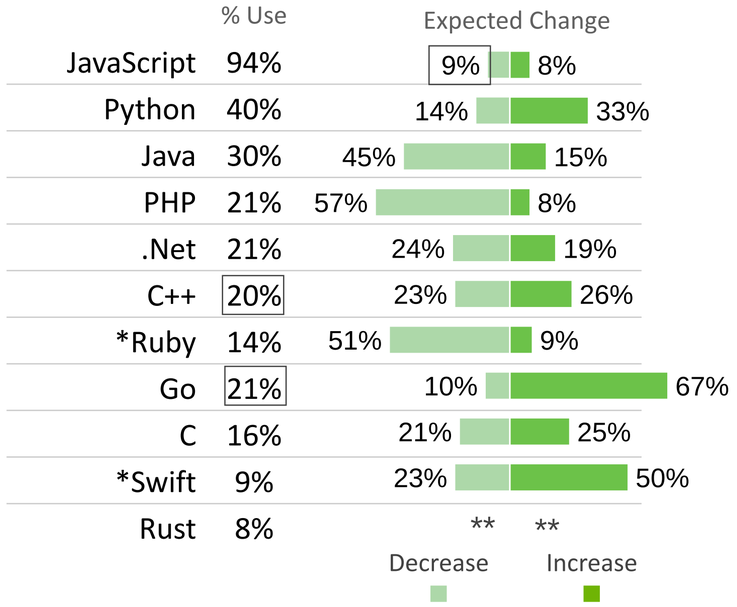
* Sample size small (n<50)

**Sample size too small to report (n<30)

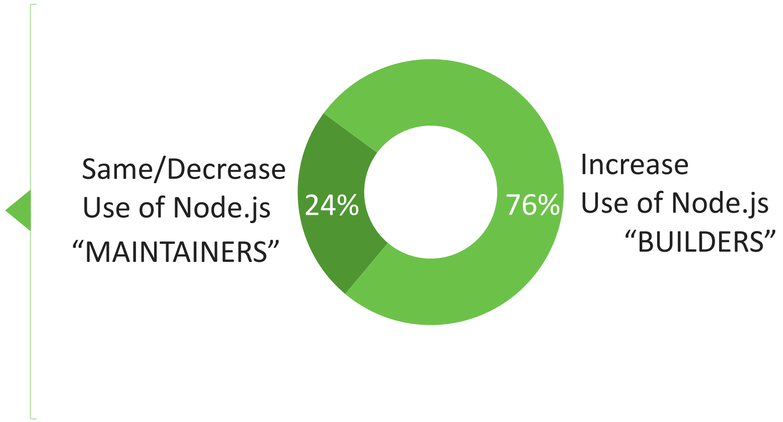
Expected Change In Other Languages

- Those who plan to increase their use of Node.js will be increasing their use of a number of other languages as well.
- Many of those who will just maintain/decrease with Node will be increasing their focus on Go or Swift.

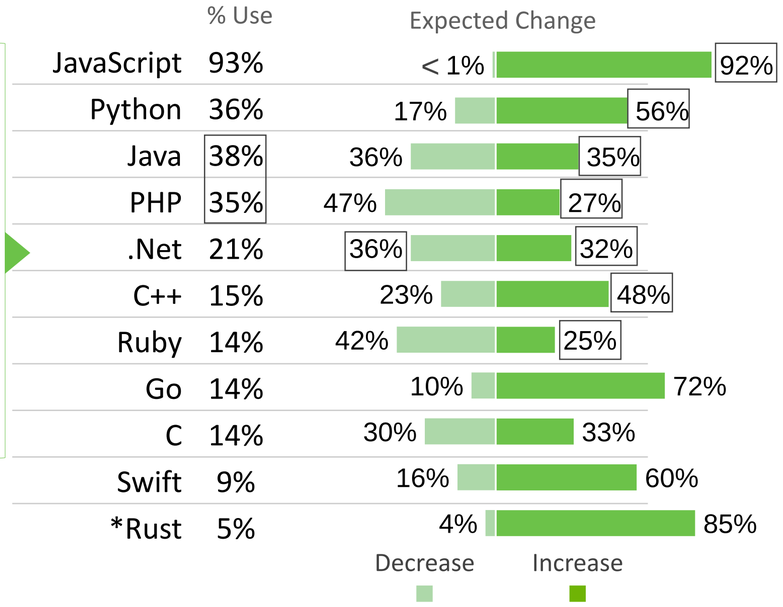
OTHER LANGUAGES: NODE.JS MAINTAINERS



Expected Change In Use of Node.js Over Next 12 Months



OTHER LANGUAGES: NODE.JS BUILDERS



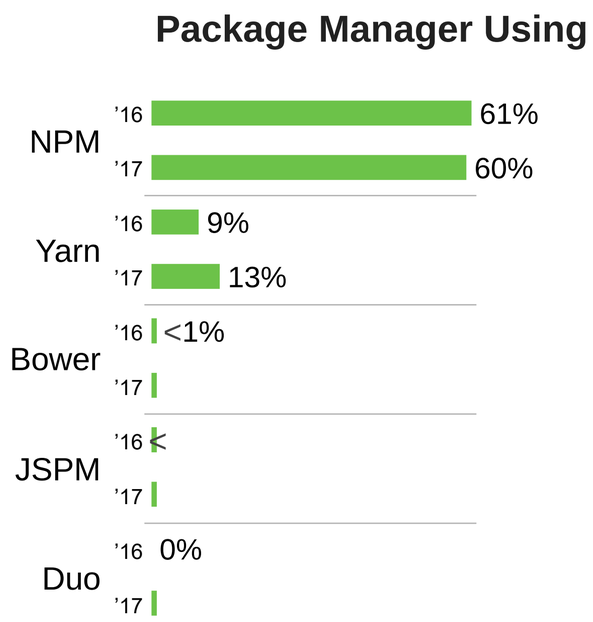
SOURCE: Q25, Q26, among those who use respective brand and who provided an answer

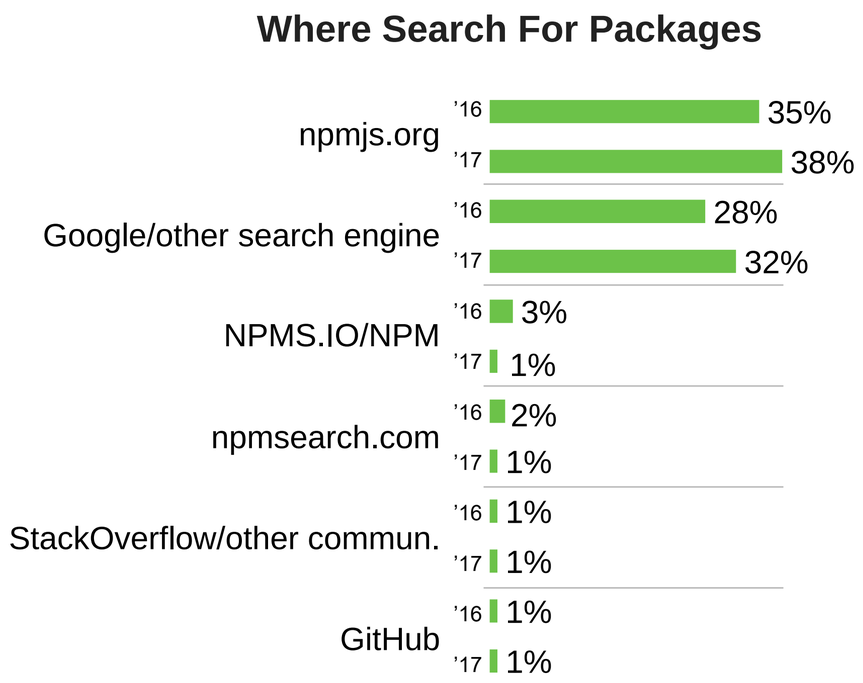
Package Managers

- NPM is by far the most widely used package manager, but Yarn is gaining in popularity.
- Node users search for packages primarily on ptmjs.org or Google/search engines; the use of Google/search engines has increased since 2016.
- It is becoming increasingly important to users to manage different packages for multi environments. Those in APAC and Latin American regions are most likely to see this as a priority.
- Availability of multiple registries is not widely seen as important in certain segments like EMEA, US/CA and small companies.
- Latin America is the only area where having multiple registries *is* an important priority.

Package Manager Usage

- NPM is, by far, the most widely used package manager – but Yarn is gaining in popularity in many subgroups.
- Respondents search for packages almost entirely on npmjs.org or through Google/search engine, which are gaining popularity in APAC and among other developers.

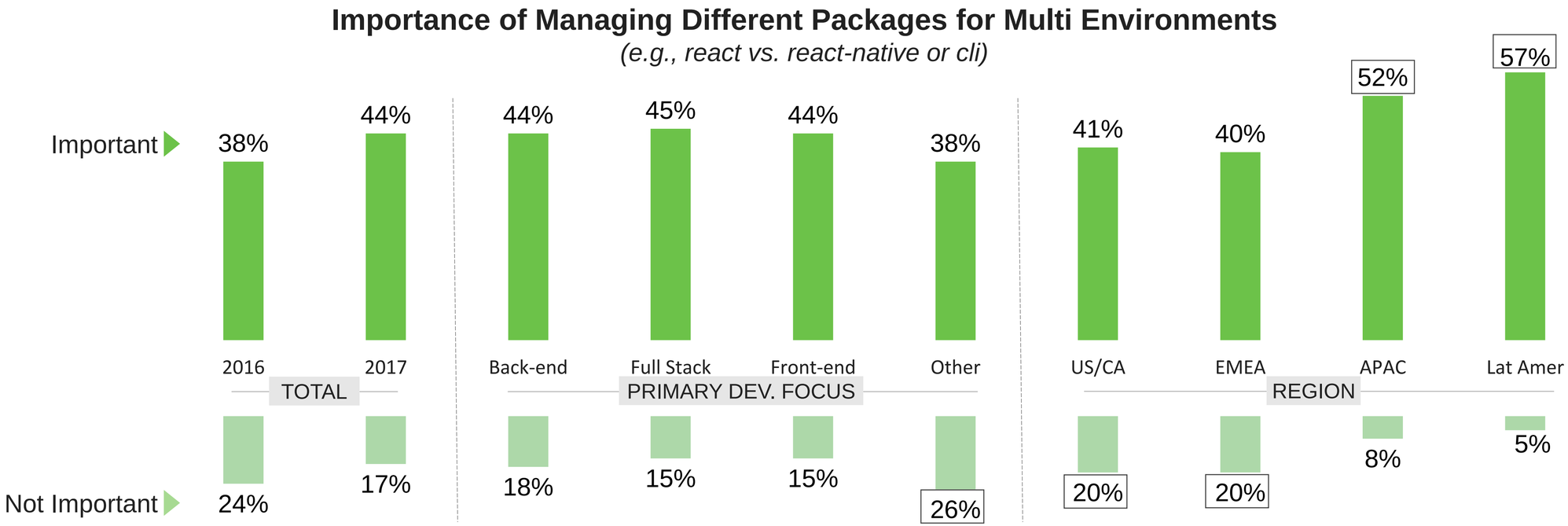




SOURCE: Q47/Q48 (including blank/no answer)

Managing Different Packages

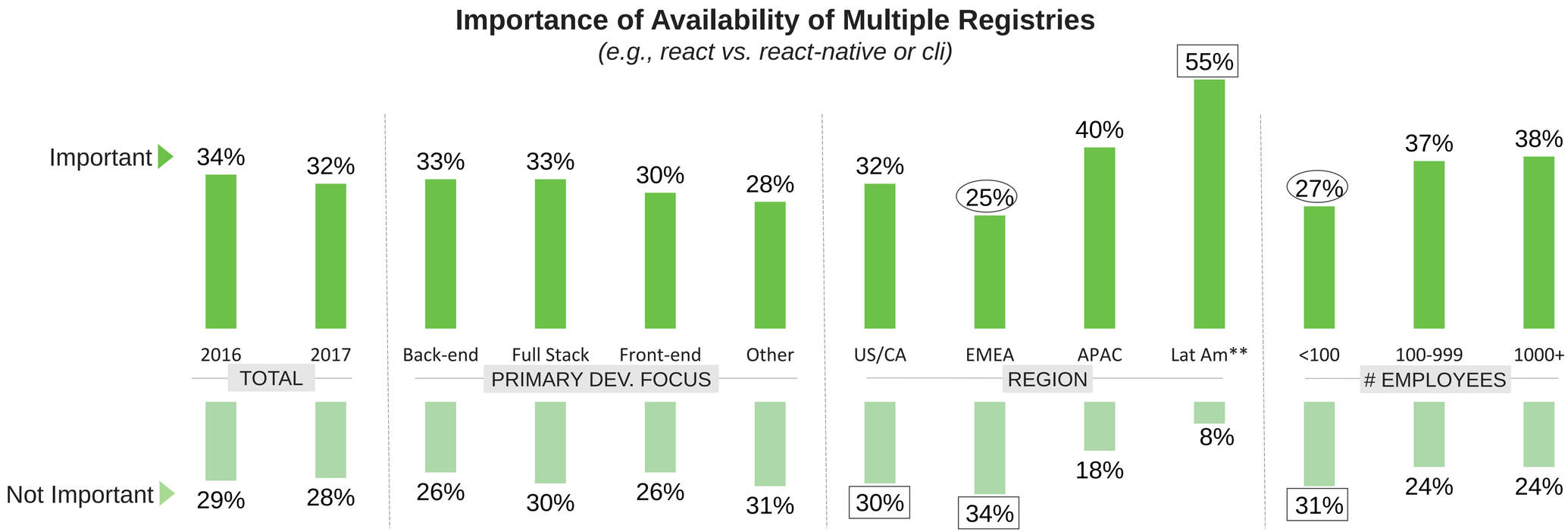
- It is becoming increasingly important to be able to manage different packages for multi environments – the rise most evident among full stack developers and those in US/CA.
- These rises notwithstanding, managing different packages is particularly important to those in APAC and Latin America.



SOURCE: Q49, among those with an opinion; blanks excluded. * Top 3/Bottom 3 box on scale from 1 ‘not at all important to 10 ‘extremely important’.

Availability of Multiple Registries

- The availability of multiple registries is not widely seen as important – at least not outside Latin America.
- EMEA respondents, and those in companies with fewer than 100 employees are least likely to value access to multiple registries.



SOURCE: Q50, among those with an opinion; blanks excluded. * Top 3/Bottom 3 box on scale from 1 ‘not at all important to 10 ‘extremely important’. **Sample size small (n<50)

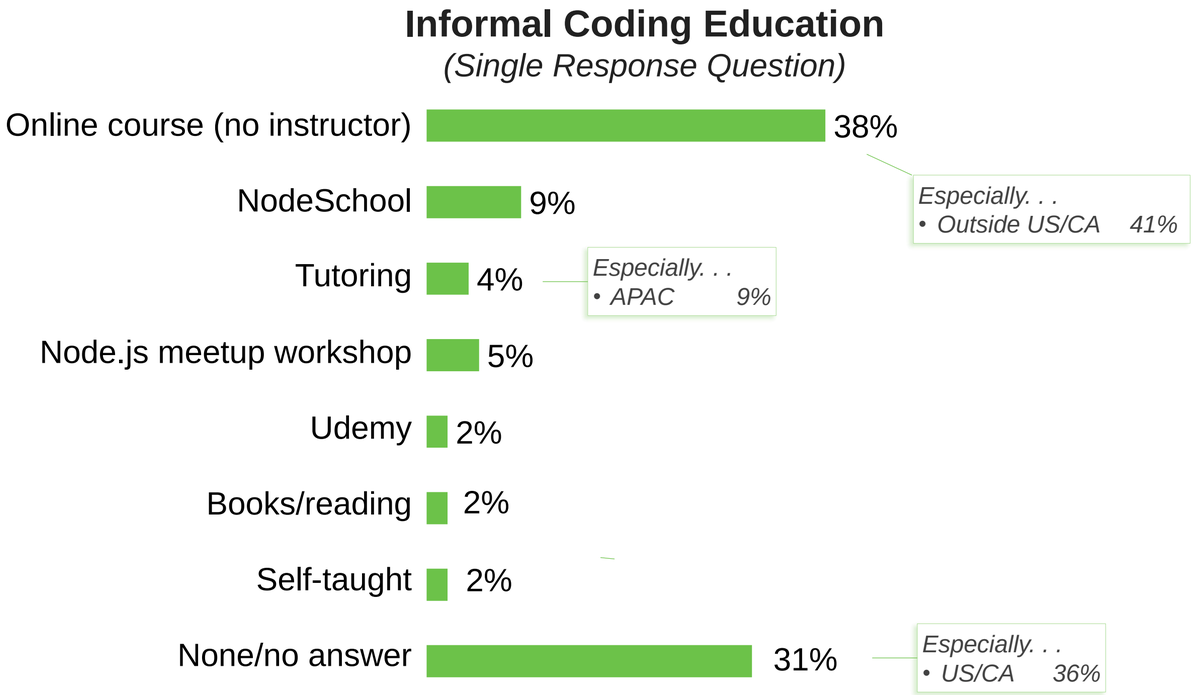
SECTION HIGHLIGHTS

Learning Node.js

- The main way Node users learn a new language is through online courses without an instructor, especially outside the US/CA
- Nearly all users learned Node in English – but for more than half, it was not their native language.
- EMEA and Latin American users are most likely to have learned in a non-native language.
- There has been a rise in those who say it is easy to learn Node.js, and improvement in scores for availability and quality of resources in several topic areas.
- Latin American users, despite having learned in non-native language, give particularly high scores for availability, quality and overall ease of learning.
- There are some differences perceptions by subgroup, with APAC and EMEA having some concerns, and mid-size companies perhaps faring better than others.
- Newer Node users are less enthusiastic about availability and quality of resources than longer term users, although most are still positive.
- Documentation and StackOverflow are the main sources users rely on when learning a new language – but free online courses and tutorial videos are also important and something users would like more of (especially new users and those in Latin America.

Informal Education

- The primary way respondents are gaining informal coding education is via online courses without an instructor – particularly outside US/CA.
- While not widely used, tutoring is more popular in APAC than in other regions.



Other Sources
(mentioned by 1% or fewer. .

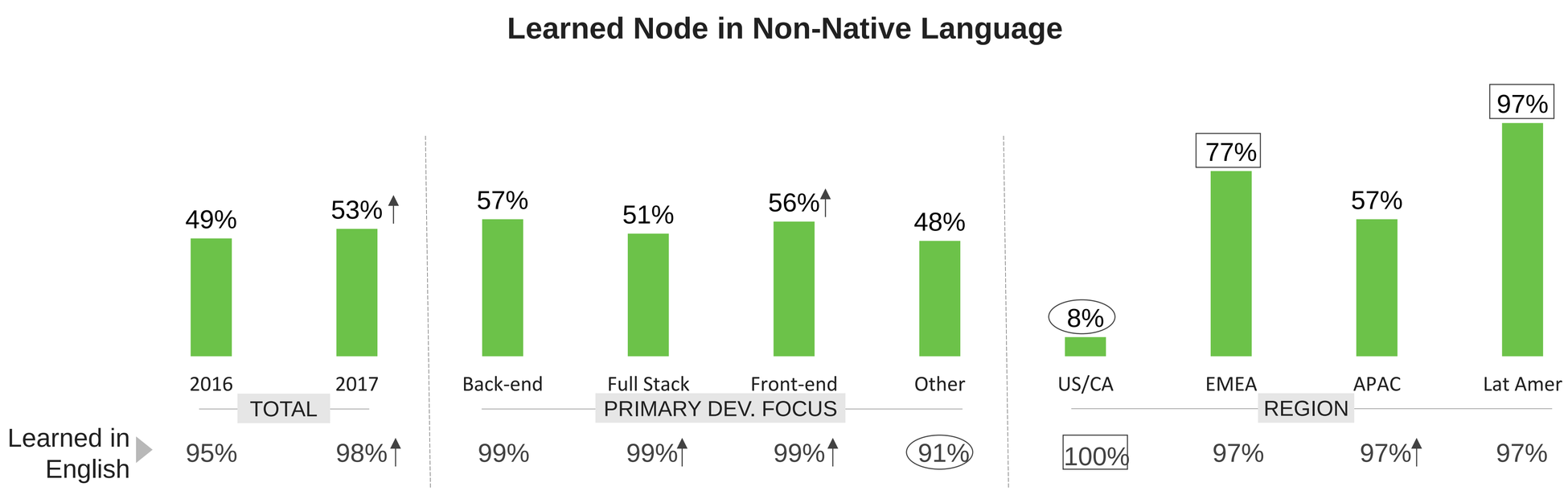
- Google/YouTube
- NodeBots
- Code School
- Code Academy
- Coding Bootcamp
- Online tutorials
- NodeTogether
- NodeGirls
- Node documentation
- PluralSight
- github
- Freecode Camp
- Blogs/articles

SOURCE: Q18

***Single response question, but allowed multiple responses in 'other specify' in 2017; not trendable**

How Learned Node

- Nearly all respondents learned Node in English – which is non-native for more than half of respondents (up since 2016).
- In certain regions – including EMEA and (especially) Latin America – a vast majority of respondents learned Node in a non-native Language.



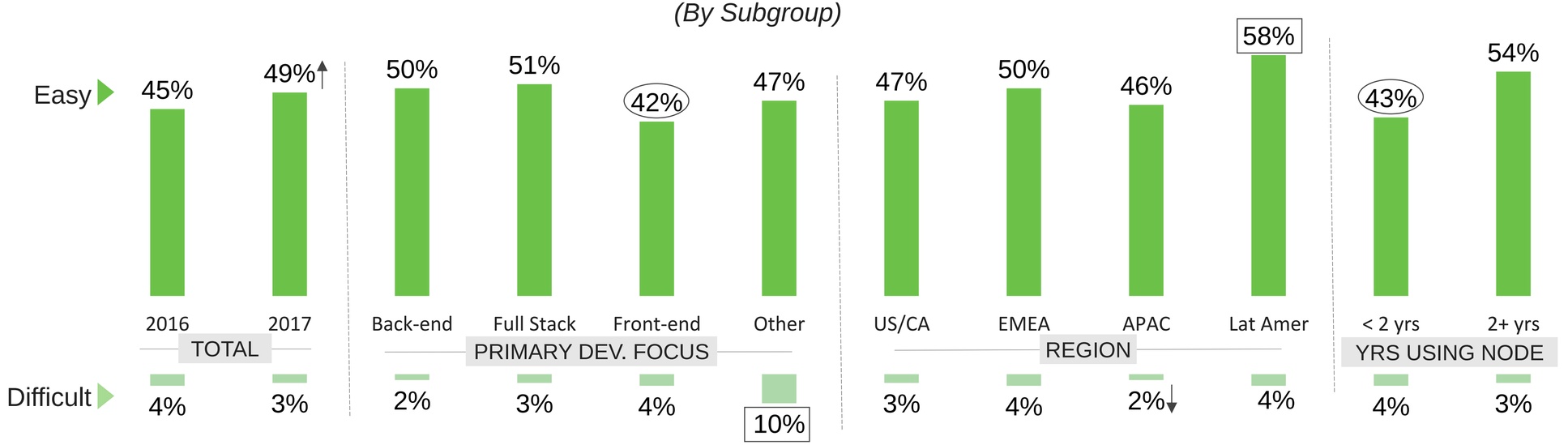
SOURCE: Q7/Q8

Ease of Learning Node

- About half of respondents say it was generally easy to learn Node.js – up slightly from 2016. Very few complain that it is difficult.
- Surprisingly, those in Latin America – who are most likely to have learned in a non-native language – are particularly upbeat about ease of learning.
- Newer Node users are less enthusiastic than those using it 2+ years – suggesting that more could be done to improve the learning experience.

Ease of Learning Node.js

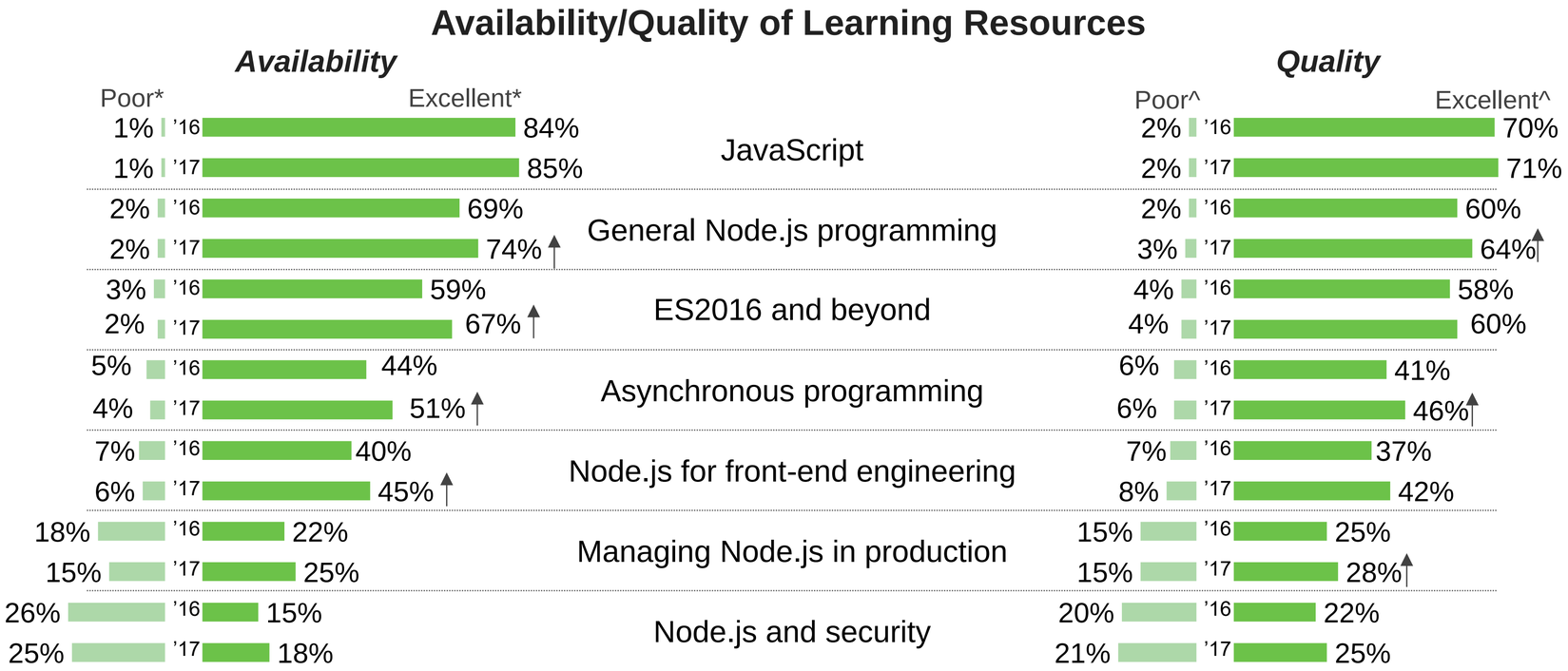
(By Subgroup)



SOURCE: Q34, among those with an opinion; blanks excluded. * Top 3/Bottom 3 box on scale from 1 ‘extremely difficult’ to 10 ‘extremely easy’.

Learning Resources

- There have been notable improvements in access to and/or quality of learning resources – particularly for General Node.js programming and Asynchronous programming.
- Still, more needs to be done to improve ratings for resources around managing node in production and Node.js and security – for both areas, high negative scores are a red flag.



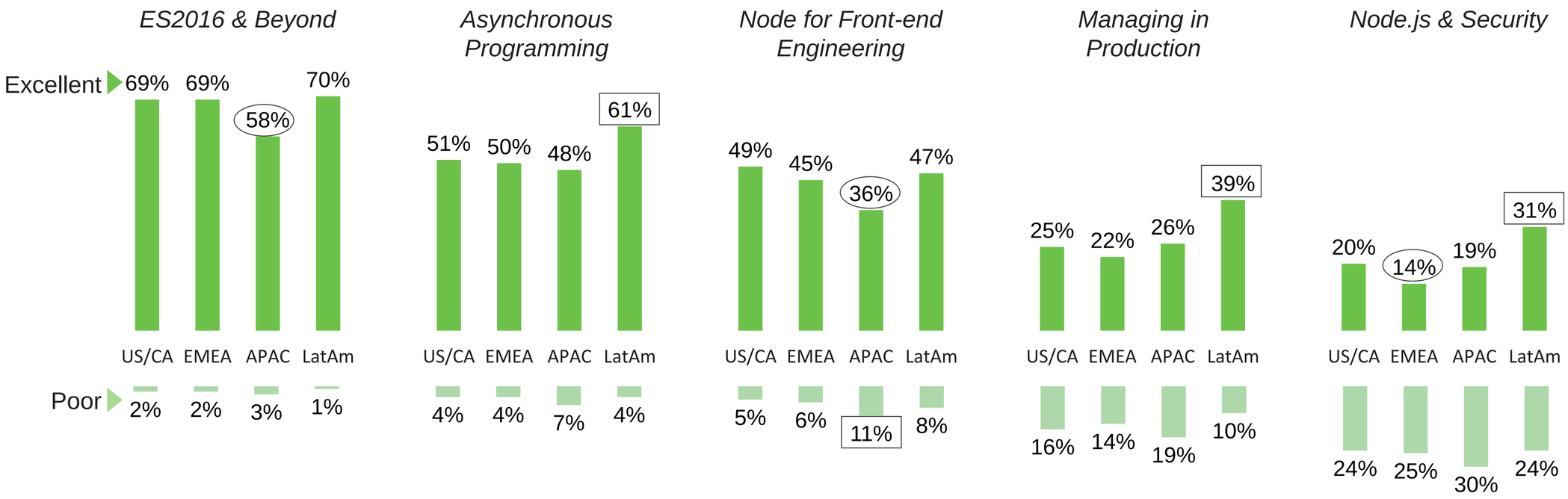
SOURCE: Q35, Q36, among those who provided a rating.

*Top 3/Bottom 3 on scale from 1 'poor – e.g., impossible to find/access' to 10 'excellent – e.g. easy to find/access'
^ Top 3/Bottom 3 on scale from 1 'poor' to 10 'excellent'

Learning Resources

- Perceptions of availability of resources vary considerably by region – with those in Latin America generally more upbeat, but those in APAC and EMEA having concerns in some specific topic areas.

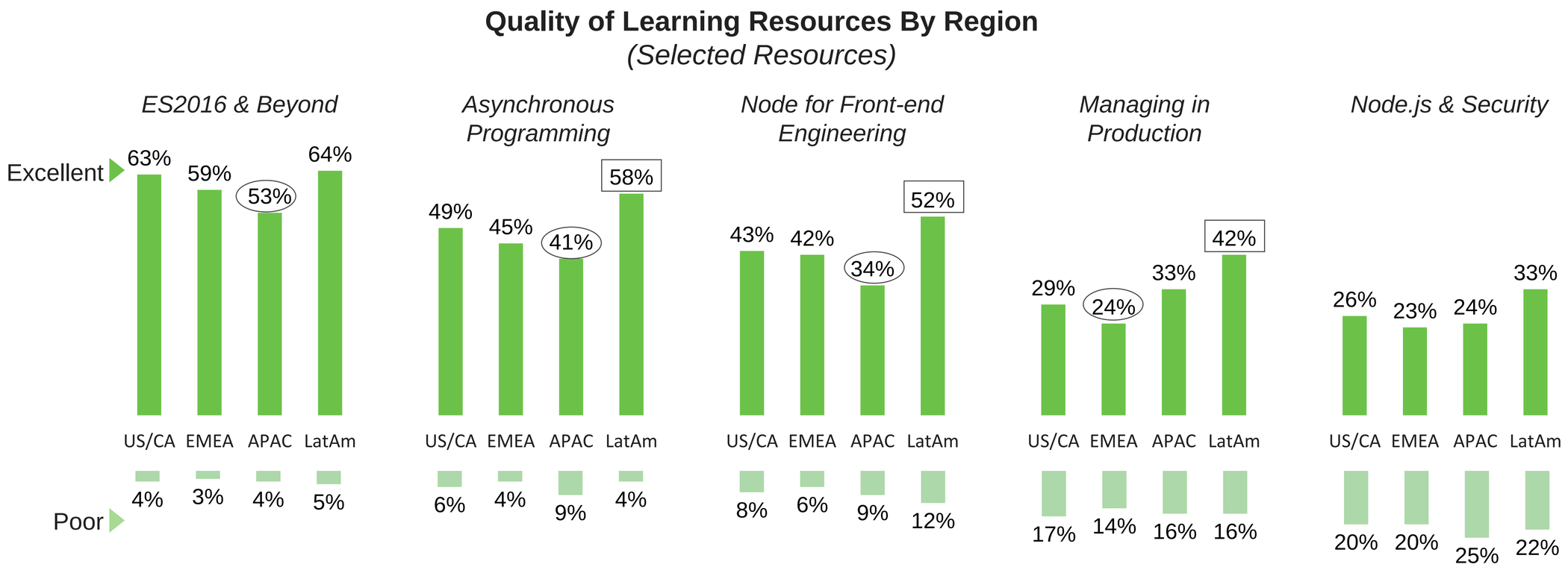
Availability of Learning Resources By Region
(Selected Resources)



SOURCE: Q35 among those who provided a rating. *Top 3/Bottom 3 on scale from 1 'poor – e.g., impossible to find/access' to 10 'excellent – e.g. easy to find/access'

Learning Resources

- A similar pattern exists with regard to quality: Latin America respondents are more pleased with quality while APAC and (less so) EMEA have concerns.



SOURCE: Q36 among those who provided a rating. ^Top 3/Bottom 3 on scale from 1 'poor' to 10 'excellent'

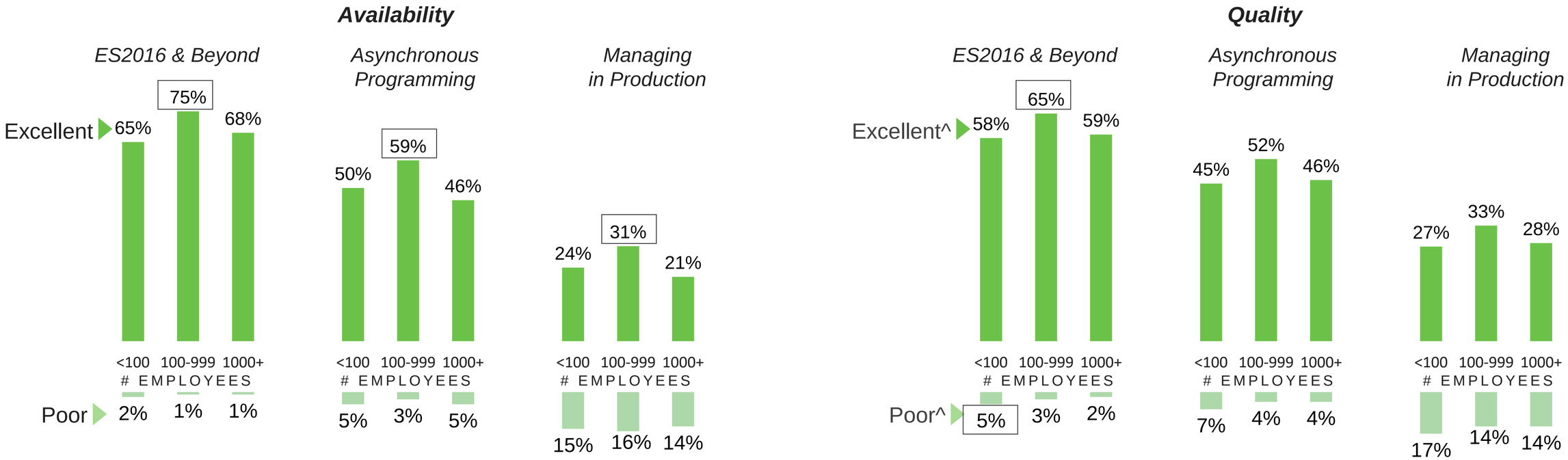
LEARNING NODE

Learning Resources

- Respondents in mid-size companies perceive greater access to some learning resources than those in larger and smaller firms, but perceived quality is only marginally better.
- To the extent there has been an improvement in perceptions of availability of resources, it's coming from midsize and smaller companies.

Availability/Quality of Learning Resources By Co Size

(Selected Resources)



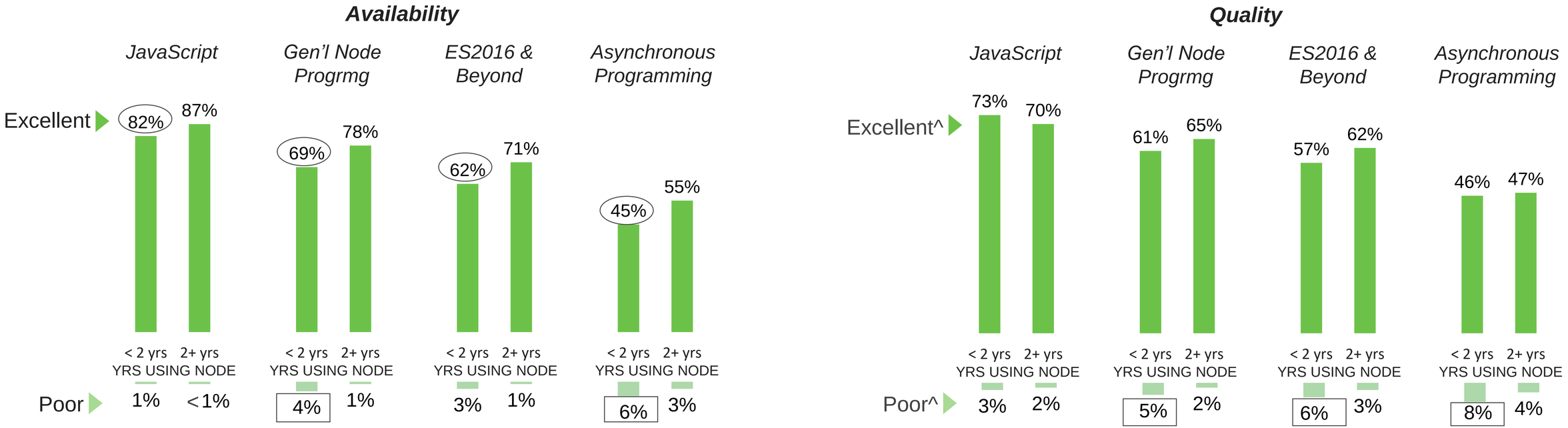
SOURCE: Q35, Q36, among those who provided a rating.

***Top 3/Bottom 3 on scale from 1 'poor – e.g., impossible to find/access' to 10 'excellent – e.g. easy to find/access'**
^ Top 3/Bottom 3 on scale from 1 'poor' to 10 'excellent'

Learning Resources

- Newer Node users give lower scores than others on availability in several topic areas.
- While still low, their higher negative ratings are worthy of note.

Availability/Quality of Learning Resources By Years Using Node
(Selected Resources)



SOURCE: Q35, Q36, among those who provided a rating.
NOT TRENDED BY SUBGROUP

***Top 3/Bottom 3 on scale from 1 'poor – e.g., impossible to find/access' to 10 'excellent – e.g. easy to find/access'**
^ Top 3/Bottom 3 on scale from 1 'poor' to 10 'excellent'

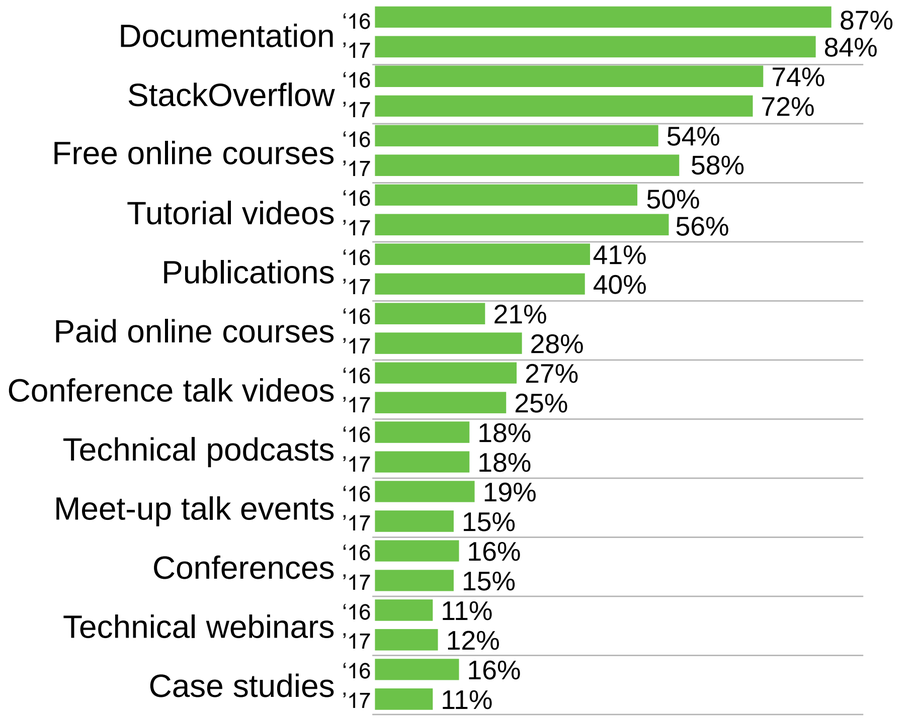
Resources Used

- Respondents use many resources when learning a new language – documentation and StackOverflow chief among them.
- Tutorial videos are also widely used – more than in 2016.
- While documentation and StackOverflow are still top, newer Node users are more likely than others to also use free & paid online courses and tutorial videos.

Other resources (used by 8% or less).

- Meet-up coding events
- Workshops (independent of conf's)
- Enterprise tooling
- Enterprise services
- Biz case study podcasts
- Biz case study webinars

Resources Rely on Most When Learning New Language/Framework/Runtime Environment



Resources Used By Years Using Node.js*

	< 2 yrs	2+ yrs
Documentation	80%	88%
StackOverflow	71	72
Free online courses	66	52
Tutorial videos	62	52
Publications	39	40
Paid online courses	35	24
Conference talk videos	19	29
Technical podcasts	16	19
Meetup talk events	10	17
Conferences	11	18

SOURCE: Q37, among those who provided a response (no answer excluded) *Not trended by subgroup

Resources Used

- To the extent users want more learning resources, documentation, free online courses and tutorial videos top the list.
- Consistent with their current usage, new Node users are particularly likely to want free online courses and tutorial videos.

Other Resources
(used by < 5%).

- Enterprise services
- Biz case study podcasts
- Biz case study webinars

Resources Would Like More Of
By Years Using Node.js*
(Top Mentions)

	< 2 yrs	2+ yrs
Documentation	33%	34%
Free online courses	42	27
Tutorial videos	30	25
Case studies	15	16
Conferences	12	16
Meet-up coding events	14	12
Paid online courses	14	10
Technical podcasts	12	12

SOURCE: Q38, including no answers

***Not trended by subgroup**

Resources Used

- Latin American users are particularly open to new learning resources – particularly around free online courses, tutorial videos, conferences and (increasingly) conference talk videos.
- Front end developers are less likely than others (and than last wave) to want more meet-up events.

Resources Would Like More Of
(Top Mentions)

By Region					*By Primary Development Focus*			
US/CA	EMEA	APAC	LatAm		Back-End	Full Stack	Front-End	Other
34%	34%	34%	36%	Documentation	40%	38%	35%	34% ↓
32	31	36	46	Free online courses	39	38	38	35
27	24	30	39	Tutorial videos	31	35	27	23
14	12	15	29	Conferences	18	18	12 ↓	10 ↓
11	16	22	21	Case studies	19	17	16	15
13	10	17	17	Meet-up coding events	16	15	9 ↓	13
11	9	14	17	Meet-up talk events	13	14	8 ↓	10
15	10	12	17	Technical podcasts	18	14	10	7
12	10	11	22 ↑	Conference talk videos	14	13	11	12

SOURCE: Q38, including no answers

Node Versions & LTS

- Most Node users use a version manager – typically Nvm.
- Just over half use LTS release line, but use of current release line is increasing – particularly among Full stack and “other” developers, and in smaller companies.
- It is important to most users to have LTS for Node.js, although it is somewhat less important to those in small companies or in EMEA.
- There has been a drop in those who say the LTS schedule support timeframe is ‘clear’, down to just half of users. Those least likely to see as clear are front-end developers, APAC and those in smaller companies.
- New users are less likely to use a version manager, and are more likely to use Apt-Get than their more seasoned peers.
- Importantly, only a minority of new users report a good understanding of the LTS schedule/support timeframe.

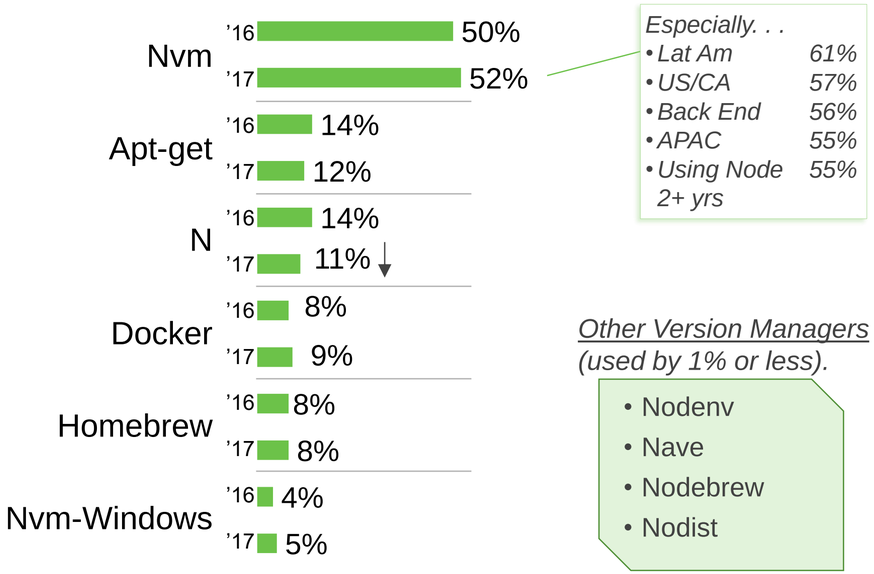
Version Managers

- Three in four Node users say they use a Node.js version manager – typically NVM.

Use a Node.js Version Manager

Node.js Version Manager Used

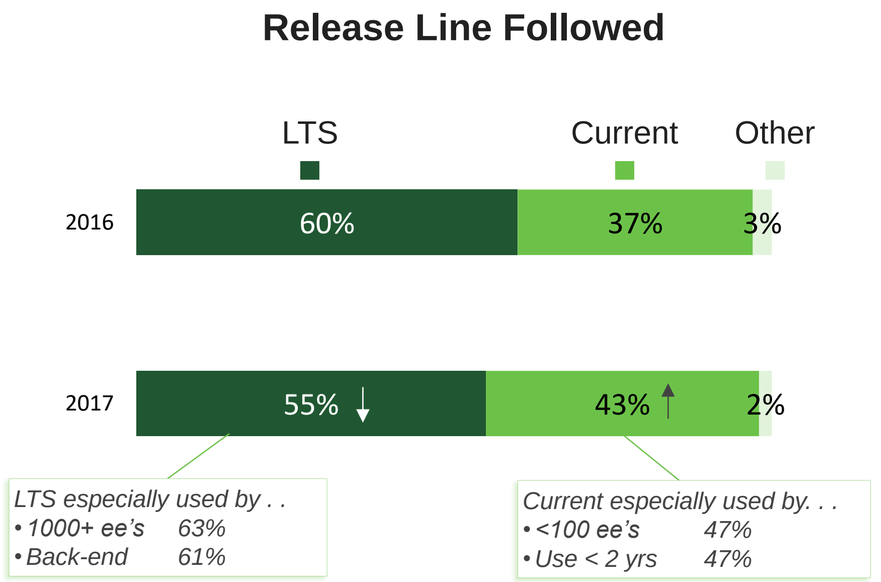
(Top Mentions, Among Those Who Use Any)



SOURCE: Q43

Release Line

- More than half of users rely on LTS release line – but that number is slipping.
- Current is particularly popular among small companies, and newer Node users.



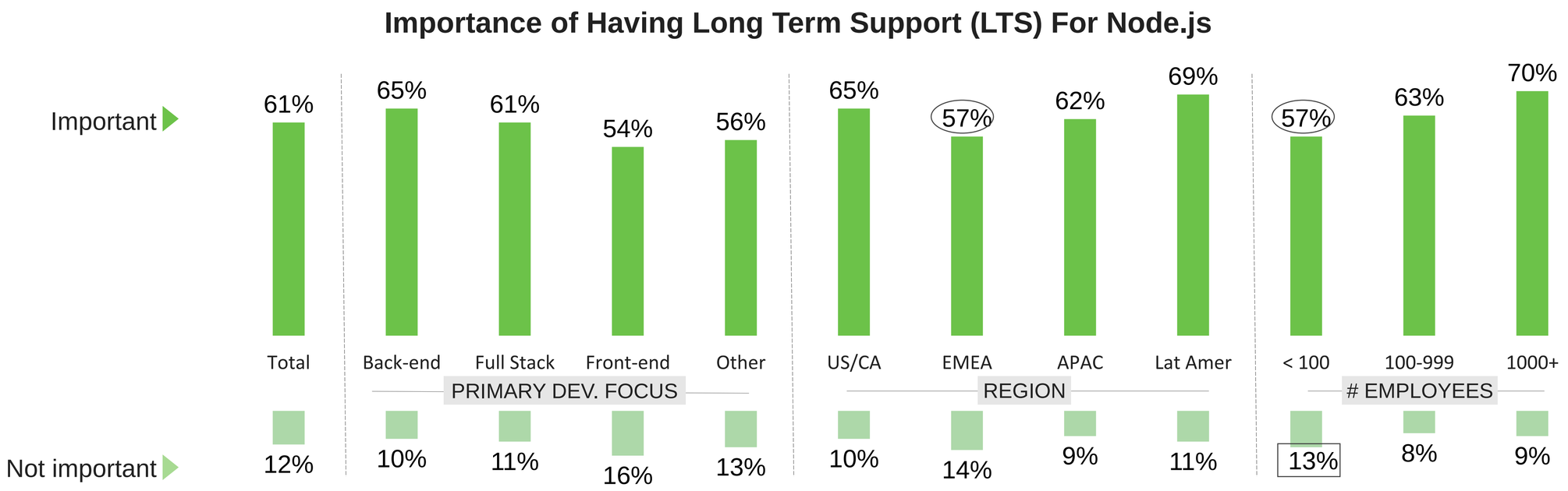
The Rise in Current is attributable to . . .

- *Full stack developers*
- *"Other" developers*
- *< 100 ee's*

SOURCE: Q39

LTS Support

- It is important to most users to have LTS for Node.js.
- Users in EMEA and smaller companies are less likely to see it as a priority, but even here, more than half say it is important.



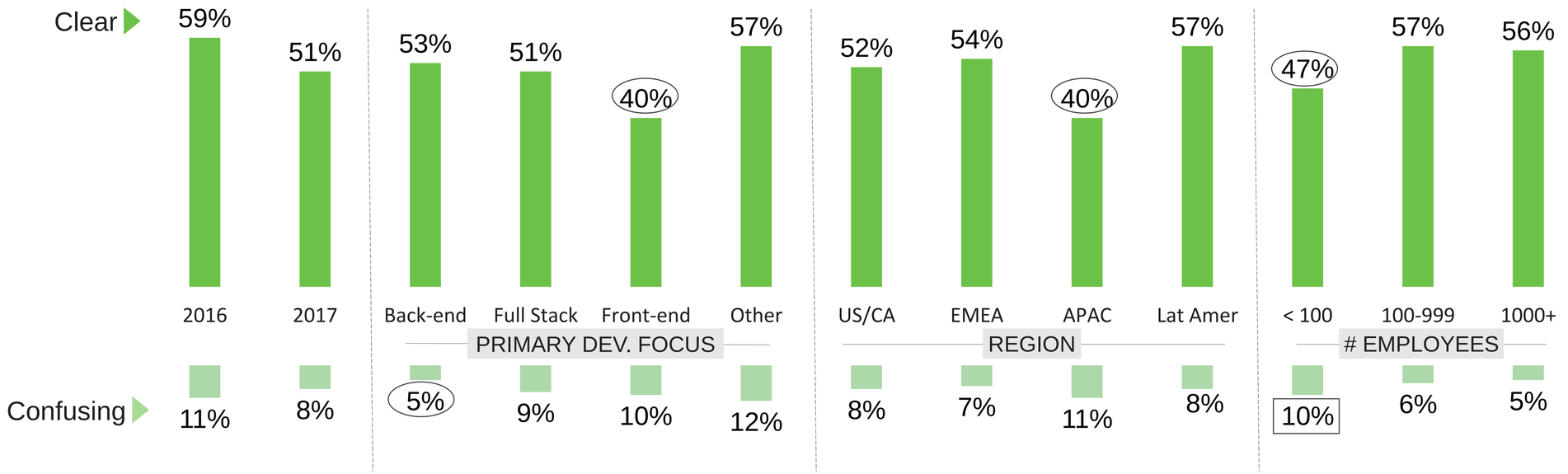
SOURCE: Q40, among those with an opinion; blanks excluded. * Top 3/Bottom 3 box on scale from 1 ‘not at all important’ to 10 ‘extremely important’ Question new in 2017/not trended

LTS Support

- While many say the LTS Schedule/Support timeframe is clear, that number has dropped significantly since 2016 – overall and across multiple segments.
- APAC users, front-end developers and those in small companies are least likely to see LTS schedule/support timeframe as clear.

Understanding of LTS Schedule/Support Timeframe for Various Versions

(By Subgroup)

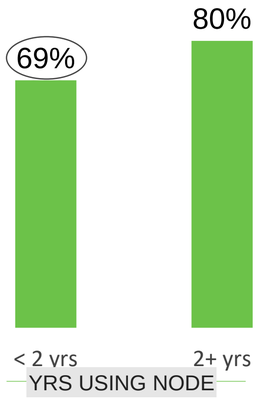


SOURCE: Q45, Among those with an opinion (Blanks excluded) *Top 3/Bottom 3 box on scale from 1 ‘very confused’ to 10 ‘perfectly clear’

Understanding Newer Node Users

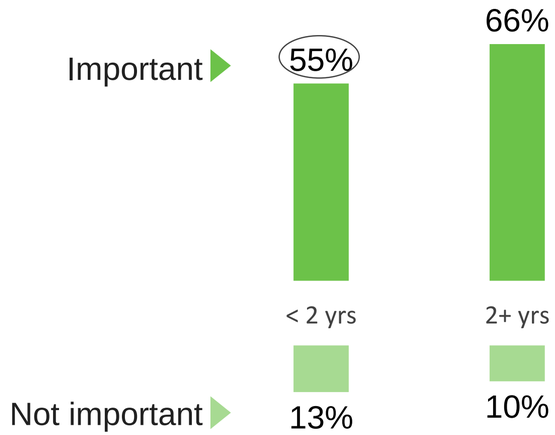
- Newer Node users are less likely than their more seasoned peers in their use of version manager, and are more likely to use Apt-Get.
- While it is not as important to them to have LTS support, most novice users still do want it, and their understanding of the schedule/support timeframe is weak.

Use Version Manager

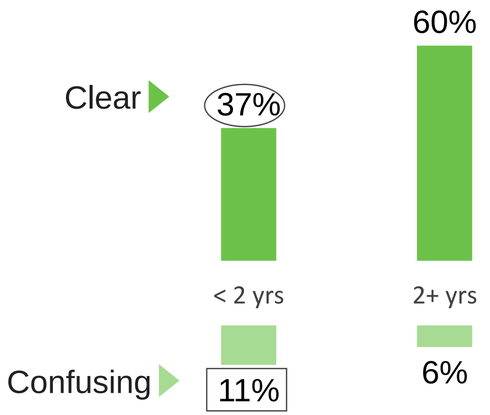


	< 2 yrs	2+ yrs
NVM	48%	55%
Apt-Get	19%	8%
N	9%	12%

Importance of Having LTS For Node.js



Understanding of LTS Schedule/Support Timeframe



SOURCE: Q40, Q43, Q45
NOT TRENDED BY SUBGROUP

* Top 3/Bottom 3 box on scale from 1 ‘not at all important’ to 10 ‘extremely important’
^ Top 3/Bottom 3 box on scale from 1 ‘very confused’ to 10 ‘perfectly clear’

SECTION HIGHLIGHTS

Node.js Impact & Getting Involved

- Users are very upbeat and excited about Node.js – with words like “fast” “easy” “awesome” “simple” “powerful” and “fun” widely used to describe Node.js
- Node.js is continuing to have a positive impact on many users – primarily through increased productivity and satisfaction, reduced development costs and increased app performance.
- The impacts may not be immediately clear however: new users are less likely to report positive impacts in many areas.
- While it’s not the most widely felt benefit, users in US/CA are more likely than others to say node.js has helped with recruiting.
- Despite their positive perceptions, few have been contributing to open source projects for Node.js.
- There is growing interest in getting involved, however: nearly a third say they are interested in contributing and nearly half say they might be open to mentoring others (both up from 2016).
- Those most interested in being involved include users in Latin America, APAC, and back-end and full stack developers.
- The main barriers are time and inexperience – but some Node users don’t know how to contribute, or feel the community is not welcoming.

Words to Describe Node.js

- In their own words, respondents used mostly positive adjectives to describe Node.js
- They particularly like that it is fast, simple, easy, yet powerful and flexible.

This is important – should be brought up earlier

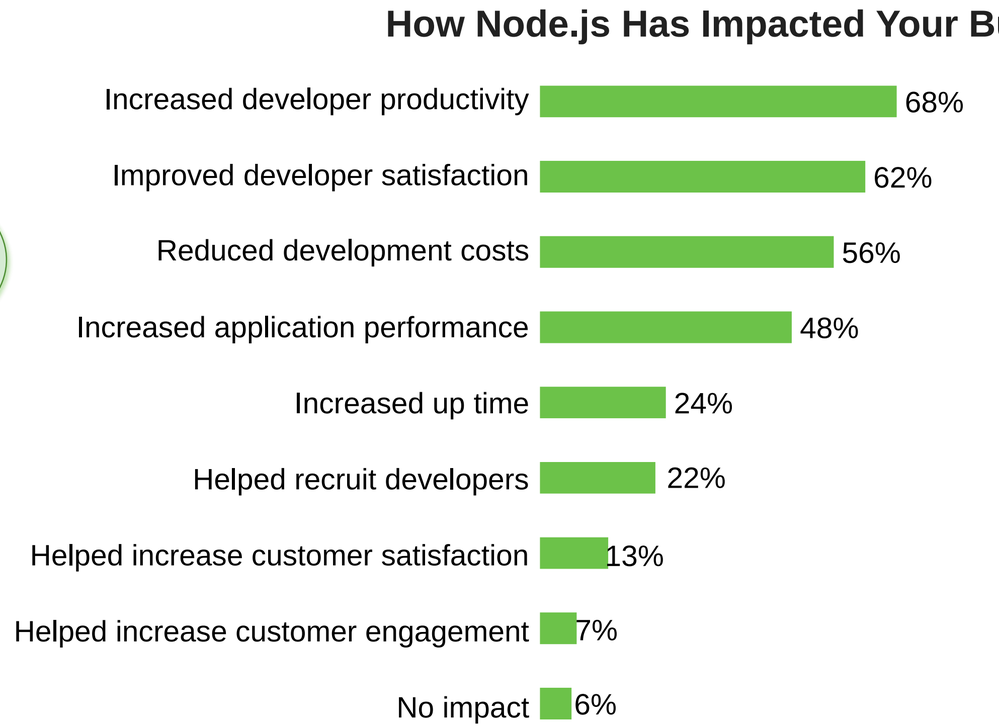
SOURCE: Q53

This is important – should be brought up earlier

Business Impact

- As in 2016, many users say that Node.js has had a positive impact on their business – chiefly through increased developer productivity and satisfaction.
- Reduced development costs and increased application performance are also important outcomes tied to Node.js.
- Perhaps not surprisingly, longer-tenured users are far more likely than novices to report an impact from Node.js.

There has been no change in business impact since 2016



Impact by Years Using Node
(Top Impacts)

	< 2 yrs	2+ yrs
Increased developer productivity	58%	74%
Improved developer satisfaction	51	68
Reduced development costs	45	62
Increased application performance	42	51
Increased uptime	19	27
Helped recruit developers	15	27
No impact	9	4

SOURCE: Q54

Business Impact

- Users in Latin America are particularly likely to note positive impacts – particularly around productivity, application performance and uptime.
- Node has helped with recruiting in the US/CA more than other regions.

This is important – should be brought up earlier

How Node.js Has Impacted Your Business

(Top Impacts)

	US/CA	EMEA	APAC	LAT AM
Increased developer productivity	68%	66%	69%	78% ↑
Improved developer satisfaction	64	60 ↓	60	65
Reduced development costs	60	51	61	56
Increased application performance	43	47 ↓	53	63
Increased uptime	22	23	24	40
Helped recruit developers	34	18	18	12
No impact	7	6	3	3

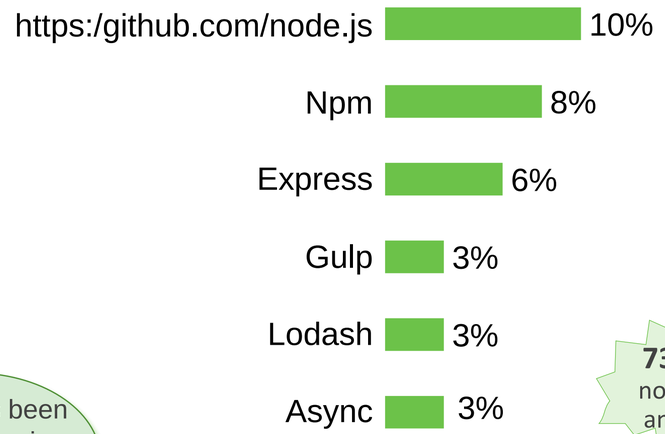
SOURCE: Q54

Groups Working With

- Consistent with 2016, relatively few Node users reports contributing to open source repositories.
- Github, NPM and Express are most widely contributed to.

Don’t need to include this or after

Open Source Repositories/Groups/Projects in Node.js Contribute To



Most Widely Mentioned:

• ...nodejs/nodejs.org	3%
• ...nodejs/LTS	2%
• ...nodejs/v8	2%
• ...nodejs/http2	1%
• ...nodejs/build	1%
• ...nodejs/node-gyp	1%
• ...nodejs/TSC	1%

There has been no change in projects contributed to since 2016

73% did not check anything

2% or fewer mention. . .

- Request
- NodeSchool
- Browserify
- I18n
- Libuv
- Streams
- Nodebots
- Node-serialport

SOURCE: Q19

Groups Working With

- Users in Asia/Pacific region are more likely to contribute to many groups .

Open Source Repositories/Groups/Projects in Node.js Contribute To

By Region					*By Primary Development Focus*			
US/CA	**EMEA**	**APAC**	**LatAm**		**Back-End**	**Full Stack**	**Front-End**	**Other**
12%	7%	12%	9%	https:/github.com/node.js	10%	10%	5%	12%
5	7	14	9	Npm	6	7	8	12
4	5	11	6	Express	6	7	4	7
2	3	5	4	Gulp	3	3	4	4
2	3	6	4	Lodash	2	3	3	3
1	3	6	4	Async	3	3	1	4
1	1	4	3	Browserify	2	1	2	2

SOURCE: Q19

Interest in Getting Involved

- Encouragingly, there has been a rise in the number of Node users who are interested in contributing to the project and/or mentoring others.
- Those most interested in being involved include users in Latin America and APAC, and back-end and full stack developers.
- While time and lack of skill are, by far, the primary barriers to contributing, some are held back because they don't know HOW to contribute.

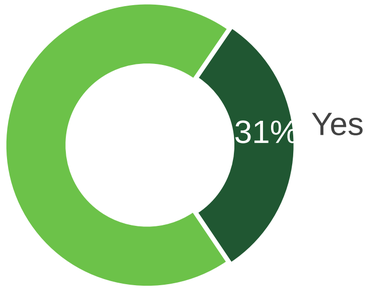
Barriers to Contributing

Mostly lack of time, or lack of experience. . .
- "I don't feel I have enough programming knowledge to contribute, but I will once I feel confident with my skills."
- "Lack of time and fear of incompetency."

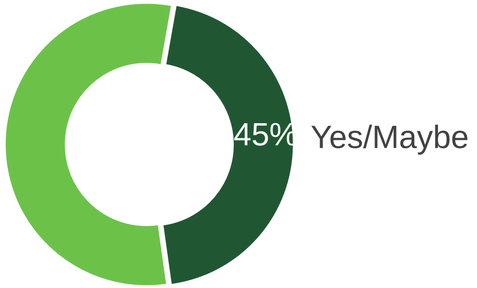
. . . .But some are unsure how to contribute, or feel intimidated
- "I'm not really sure how to get started or how large of a commitment I have to make to make a difference
- "It is kind of intimidating to know where to start, plus many of the PRs look very unfriendly."

. . . A few respondents describe a community that is not welcoming
- "Node makes it super clear contributions from those not in the cool camp are not welcomed. The core is not a friendly, welcoming, or well-adjusted place."
- "It's a closed, clique-ish community."
- "Toxic, male dominant community'.

Interest in Contributing to Open Source Node.js Project



Interest in Mentoring Others



SOURCE: Q58, Q59, Q60

Thank you

gwallace@linuxfoundation.org