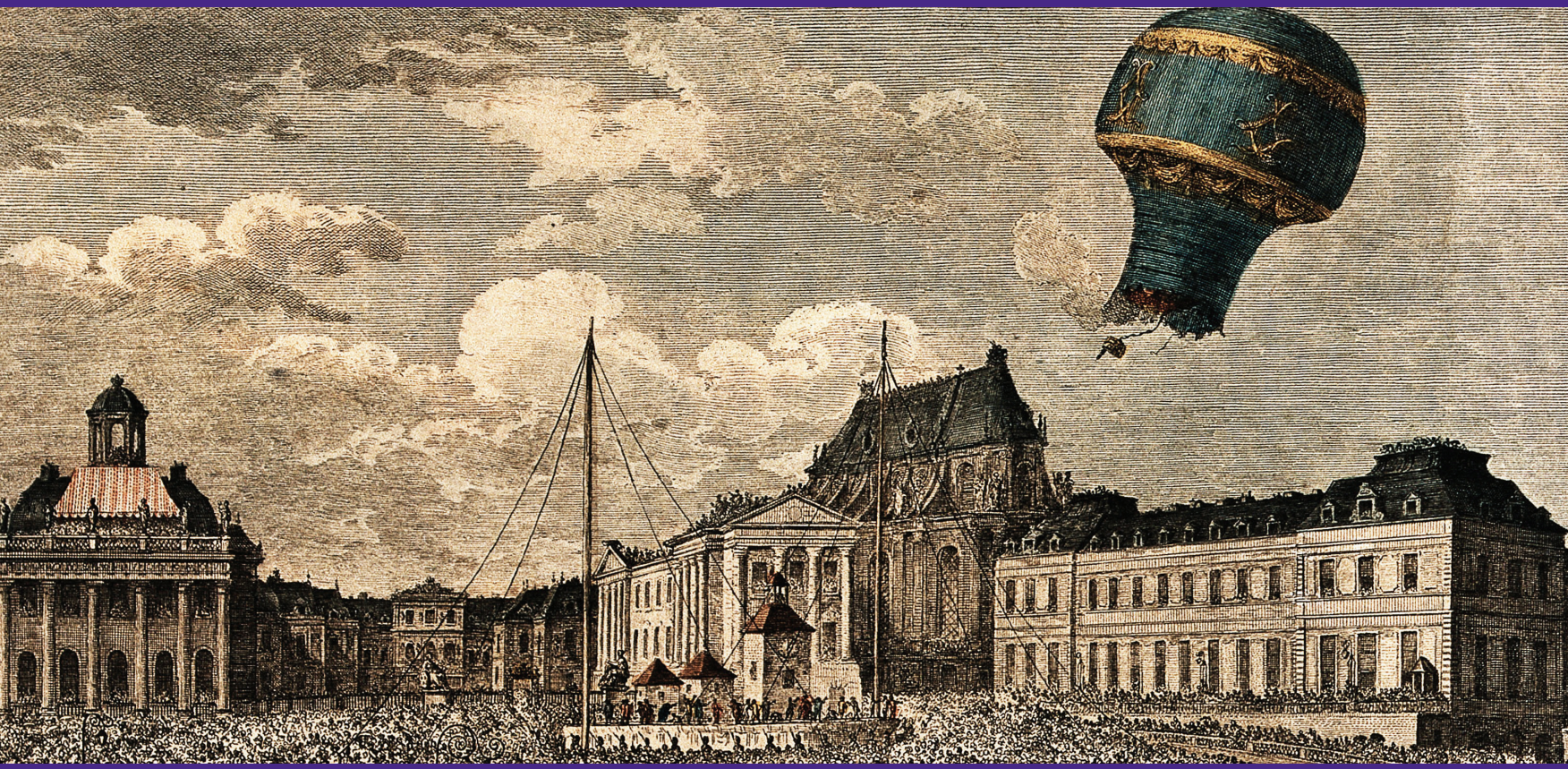
O'REILLY®

2016

European Software Development Salary Survey

Tools, Trends, Titles: What Pays (and What Doesn't) for Programming Professionals in Europe



Andy Oram & John King

O'REILLY®

OF THE MORE THAN 5,000 PARTICIPANTS in O'Reilly Media's 2016 Software Development Salary Survey, 1,353 software engineers, developers, and other programming professionals live and work in Europe. This report includes complete survey results from those respondents, including participants in organizations both large and small, and from a wide variety of industries. You'll learn about the current state of software development—and the careers that propel it—across Europe today.

This report presents European survey results, including:

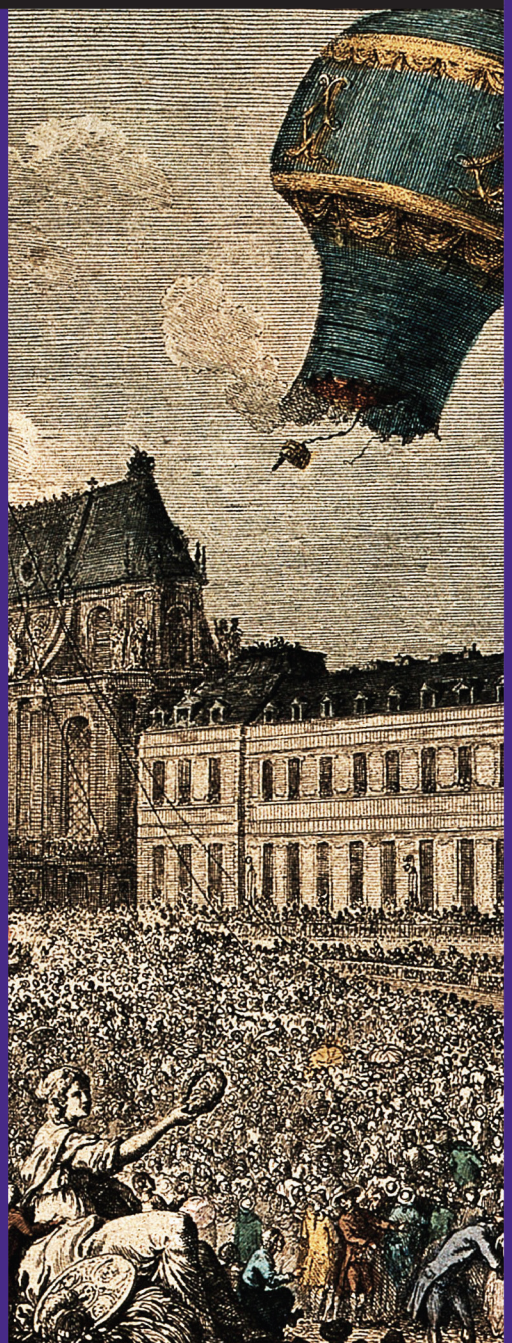
- **The top programming languages that respondents currently use professionally**
- **Where, by country, European programmers make the highest salaries**
- **Salary ranges by industry and by specific programming language**
- **The difference in earnings between programmers who work on very small teams versus those who work on larger teams**
- **The most common programming languages that European respondents no longer use in their work**
- **The most common languages that respondents intend to learn within the next couple of years**

Find out where you stand in the European programming world. We encourage you to plug your own data points into our survey model to see how you compare to other programming professionals in your industry.

Andy Oram is an editor at O'Reilly Media specializing in programming topics.
John King is a data analyst at O'Reilly Media.

ISBN: 978-1-491-96911-3

Take the 2017 survey by visiting oreilly.com/programming/2017-programming-salary-survey.html



Open source software, architecture, frameworks, and tools for today's engineers.

If software is what you do, OSCON is where you want to be.
Get better at what you do and rekindle your love of all things digital.

"This is my opportunity once a year to be completely immersed in All Things Open Source!"

—Mike Wright, Independent

O'REILLY®
oscon
LONDON

oscon.com/uk | #oscon | @oscon

©2015 O'Reilly Media, Inc. O'Reilly is a registered trademark of O'Reilly Media, Inc. D2365

Take the Software Development Salary Survey

SOFTWARE DEVELOPMENT IS A THRIVING FIELD with plenty of opportunities for growth and learning. But because it's moving so quickly, it can be tough to keep pace with rapidly evolving technologies. Choosing the right ones to focus your energy on can lead to bigger paychecks and more career opportunities.

We're setting out to help make more sense of it all by putting a stake in the ground with our annual Software Development Salary Survey. Our goal in producing the survey is to give you a helpful resource for your career, and to keep insights and understanding flowing.

But to provide you with the best possible information we need one thing: participation from you and other members of the programming community. Anonymous and secure, next year's survey will provide more extensive information and insights into the demographics, roles, compensation, work environments, educational requirements, and tools of practitioners in the field.

Take the 2017 O'Reilly Software Development Salary Survey today. (And don't forget to ask your colleagues to take it, too. The more data we collect, the more information we'll be able to share.)

oreilly.com/programming/2017-programming-salary-survey.html

2016 European Software Development Salary Survey

Tools, Trends, Titles: What Pays (and What Doesn't) for Programming Professionals

Andy Oram & John King

O'REILLY

2016 EUROPEAN SOFTWARE DEVELOPMENT SALARY SURVEY

by Andy Oram and John King

Editors: Dawn Schanafelt, Susan Conant
Designer: Ellie Volckhausen
Production Editor: Shiny Kalapurakkel

Copyright © 2016 O'Reilly Media, Inc. All rights reserved.

Printed in Canada.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (http://safaribooksonline.com). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

July 29, 2016. First Edition

ISBN: 978-1-491-96911-3

REVISION HISTORY FOR THE FIRST EDITION

July 29, 2016: First Release

While the publisher and the author(s) have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author(s) disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

Table of Contents

2016 Software Development Salary Survey i
Executive Summary 1
Introduction 2
Geography 5
Company Types 9
Team Structure 14
Individual Background 16
Title, Role, Tasks 18
Tools 24
Programming Languages 38
Work Week, Bargaining, and Ease of Finding Work 47
The Model in Full 51
Conclusion 54

OVER 5,000 RESPONDENTS FROM A VARIETY OF INDUSTRIES COMPLETED THE SURVEY

YOU CAN PRESS ACTUAL BUTTONS (and earn our sincere gratitude) by taking the 2017 survey—it only takes about 5 to 10 minutes, and is essential for us to continue to provide this kind of research.

oreilly.com/programming/2017-programming-salary-survey.html

Executive Summary

IN 2016, O'REILLY MEDIA CONDUCTED A SOFTWARE DEVELOPMENT SALARY SURVEY ONLINE. The survey contained 72 questions about the respondents' roles, tools, compensation, and demographic background. More than 5,000 software engineers, developers, and other professionals involved in programming participated in the survey, 1,353 of them from European countries. This provided us with the opportunity to explore the software-development world—and the careers that propel it—in great detail. Some key findings include:

- Top languages currently used professionally in the sample: JavaScript, HTML, CSS, Java, Bash, and Python.
- Respondents reported using an average of 3.6 languages.
- The highest salaries are in Switzerland, the UK, Ireland, Denmark, and Norway.
- Software development is a social endeavor: people who are on tiny teams and who don't attend meetings tend to earn much less.
- The most common languages that respondents used in the past but no longer use were C/C++, Java, and PHP.
- The most common languages that respondents stated they intend to learn in the next 1–2 years were Go, Swift, Python, and Scala.
- Salary estimates can be obtained from a model based on the survey data whose coefficients are mentioned throughout the report and repeated in full at the end. We hope you will learn something new (and useful!) from this report, and we encourage you to try plugging your own data points into the model.

If you are a developer, you may be wondering, "What should I be earning?" Or at least, "What do other people with work similar to mine earn?" To satisfy this curiosity, at the end of this report, we have provided a way to do a salary estimate. Our model is based on the survey data whose coefficients are mentioned throughout the report. We hope you will learn something new (and useful) from this report, and encourage you to try plugging your own data points into the model.

Introduction

THE FIRST O'REILLY SOFTWARE DEVELOPMENT SALARY SURVEY was conducted through an online survey hosted on Google Forms. More than 5,000 respondents submitted responses between January and May 2016, from 51 countries and all 50 US states, from companies both large and small, and from a wide variety of industries. Respondents were mostly software developers, but other professionals who program also participated in the survey.

Of the responses to the survey, 1,353 came from 27 countries in Europe, and those form the basis of the data in this report. The report on the worldwide findings, with some US-specific statistics, can be downloaded from O'Reilly's web site.

When asking respondents about salaries, we recorded responses in US dollars, and therefore will use dollars throughout this report. The median salary of the entire EU sample was $56,000, with the middle half of all respondents earning between $35k and $80k. The latter statistic is called the interquartile range (IQR)—the middle 50%—and is used to describe the salaries of particular subsets of the sample in this report and its graphs. Imagine the IQR as a bell curve or normal distribution with the left-most 25% and right-most 25% cut off. The IQR is useful for showing the middle of the salary range without the distortion of outliers in the lowest and highest quartiles.

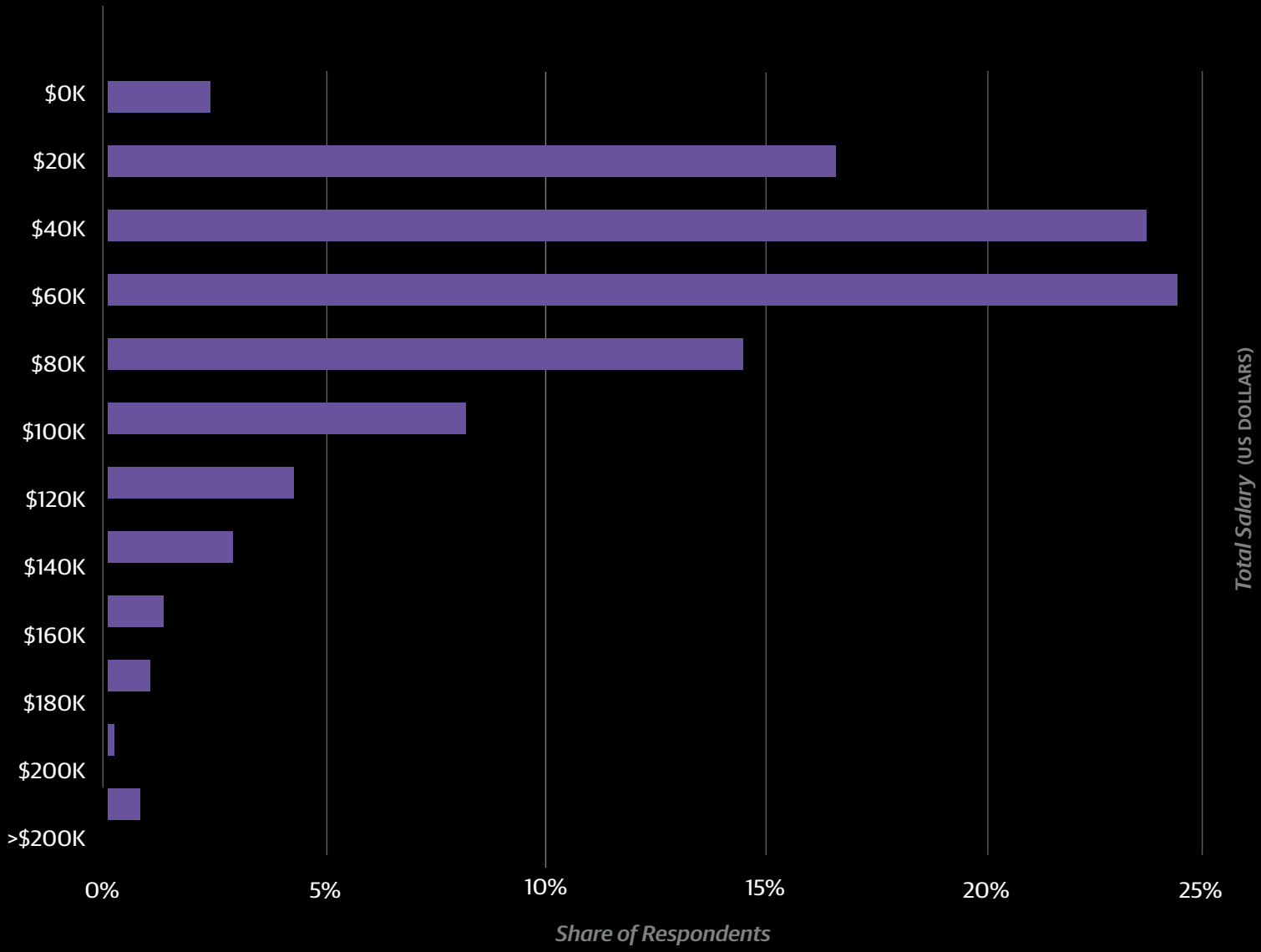
insignificant. In each section we mention the relevant, significant coefficients, and at the end of the report we repeat those coefficients when we show the full model.

In the horizontal bar charts throughout this report, we include the interquartile range (IQR) to show the middle 50% of respondents' answers to questions such as salary. One quarter of the respondents has a salary below the displayed range, and one quarter has a salary above the displayed range.

The IQRs are represented by colored, horizontal bars. On each of these colored bars, the white vertical band represents the median value.

TOTAL SALARY

SHARE OF RESPONDENTS



Much of the variation in salary matches other variables gathered via the survey. We quantify how much each variable seems to contribute to salary. For instance, the country you are in has a major impact on your salary, and the programming language you use has a much smaller (but often important) impact, whereas a person's age has no impact at all. Therefore, in addition to simply reporting the salaries of certain groups of respondents, such as those who work a certain industry or use a certain language, we also estimate how much the differences in salaries are correlated with the variables reported. We have found that we can do this using a simple, linear equation (a + b + c + …), developing the coefficients from the survey data. The coefficients are contribution components: by summing the coefficients corresponding to programming language, job role, or other variables, we obtain an estimate for their salary.

Note that not all variables get included in the model, because the method used to generate the model penalizes complexity to avoid overfitting and thus deems many variables insignificant. In each section we mention the relevant, significant coefficients, and at the end of the report we repeat those coefficients when we show the full model.

A primary motivation for constructing a linear model is to clarify the relationship between salary and demographic or role-related variables when two variables are highly correlated. It is worth remembering that correlation does not imply causation. A classic example involves meetings: just because salary clearly rises with the weekly number of hours spent in meetings, don't expect to get a raise just by maneuvering to add meetings to your schedule! Keep in mind that the survey methodology does not support what may, intuitively, seem like reasonable assumptions of causation from even the strongest correlations—testing for causation is a difficult process at best.

We excluded managers and students from the model because many of the features we think might help determine salary, such as language use, likely work differently (if at all) for these groups. We also exclude those working fewer than 30 hours per week.

Geography

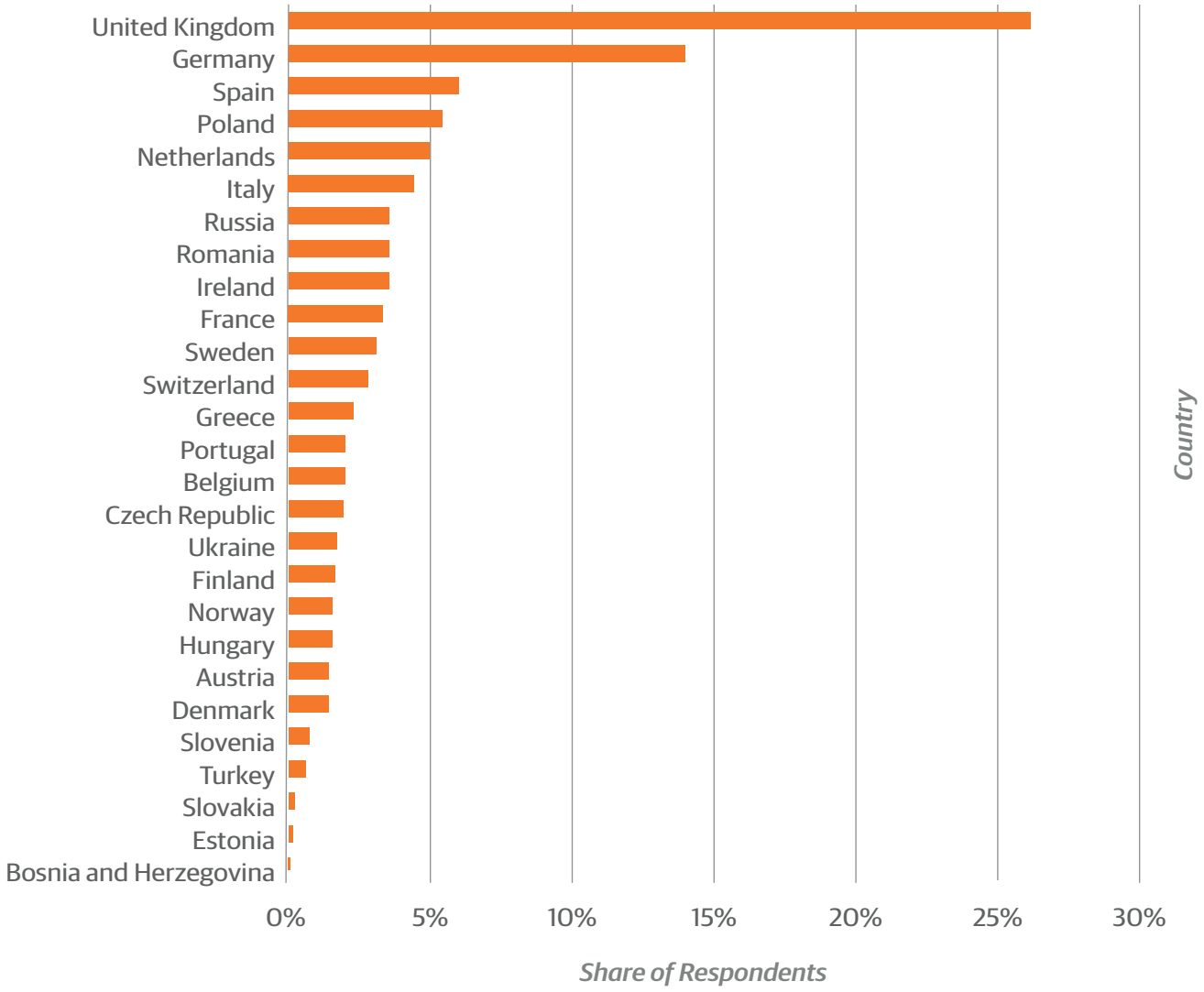
ONE OF THE MOST BASIC PIECES OF INFORMATION with a strong effect on salary is geography. Top countries where respondents were based were the UK (26%), Germany (14%), Spain (6%), Poland (5%), and the Netherlands (5%); 10% were based in countries not currently in the EU.

Thirty countries had at least 20 respondents in the sample, allowing for a more detailed view of salary by region. We should note that, even so, not every country is assigned a separate coefficient: coefficients are chosen for world regions (usually continents) or for countries where salaries vary greatly from those in other countries in the region. In this section, therefore, we compare European countries to each other and to other regions of the world. We also note that the positive and negative US dollar amounts quoted as coefficients are only the beginning of a salary estimate: more coefficients will be added later on.

Thirty countries had at least 20 respondents in the sample, allowing for a more detailed view of salary by region.

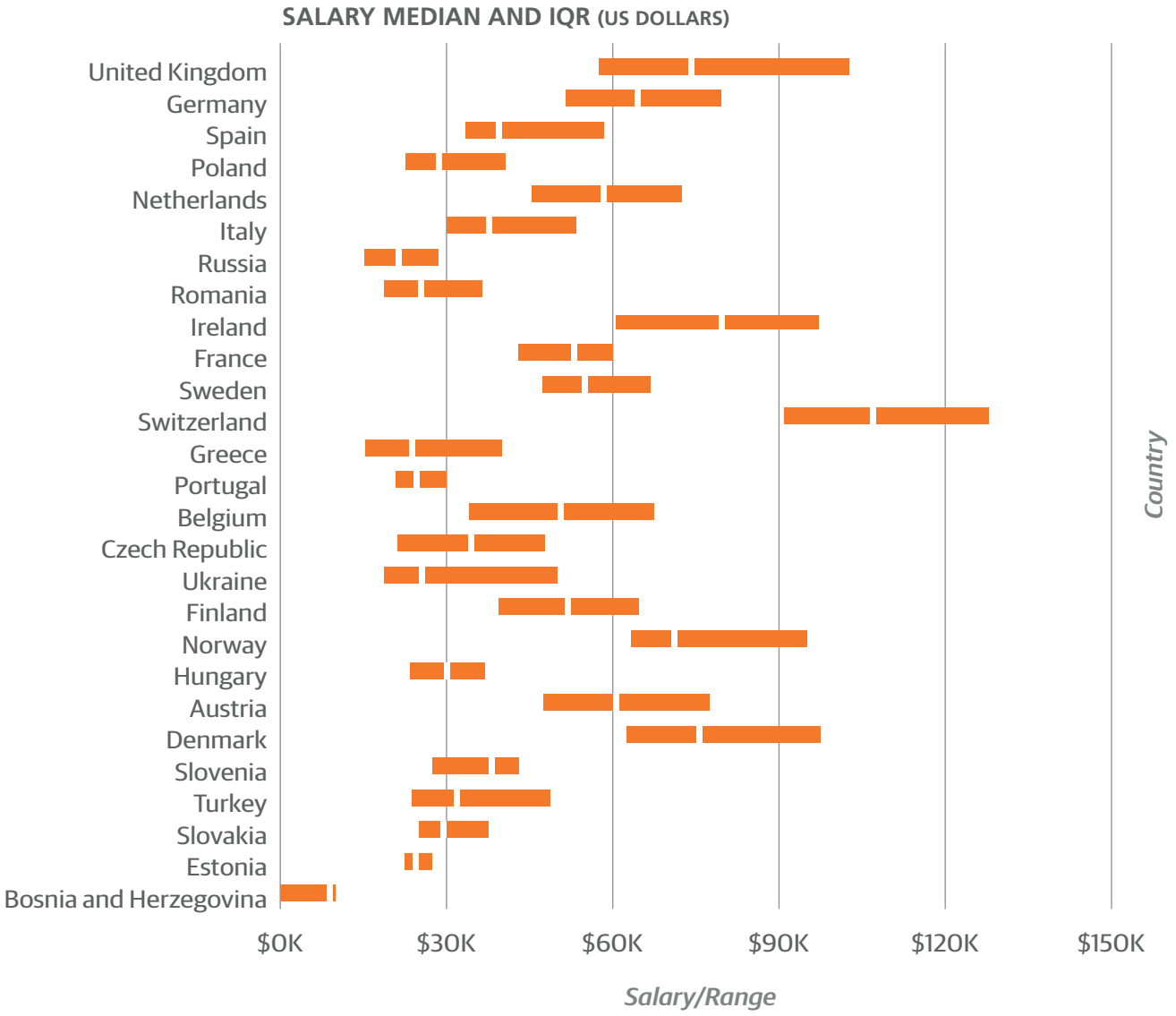
After the US, Switzerland, and Japan, the highest geographical coefficient was Australia's, at +$29,636. New Zealand and Canada were lower (+$17,433 each), while Latin America (chiefly Brazil, Mexico, Argentina, and Colombia) had a coefficient of –$9,057, below Asia but above Eastern Europe. South Africa (the only African country represented in the sample) had a relatively high median salary—$46K (compared to $31K for Asia)—but the South African respondents also tended to be among the most experienced in the sample, so their coefficient was only –$3,766. This is likely just a quirk of the sample and is another good example of why the linear-model coefficients are a better lens to compare features than median salary. *(continued)*

COUNTRIES
SHARE OF RESPONDENTS



COUNTRIES

SALARY MEDIAN AND IQR (US DOLLARS)



Geography (continued)

Switzerland: +$19,161

United Kingdom, Ireland, Norway, Denmark: –$5,513

France, Sweden, Belgium, Finland, Austria: –$22,283

Spain, Italy, Greece, Portugal, Turkey: –$35,911

Poland, Romania, Czech Republic, Ukraine, Hungary, Slovenia, Slovakia, Estonia, Bosnia and Herzegovina: –$42,594

Russia: –$45,224

Salaries in Europe were uneven, with differences among European countries as great as those between world regions. The model assigned numerous coefficients to Europe, grouping countries into six sets. Switzerland was in a class of its own, with a coefficient of +$19,161, and was the only European country with salaries comparable to the US and Japan. Northern/Western Europe tended to have higher salaries, with the UK, Ireland, Norway and Denmark assigned a coefficient of –$5,513, and Germany and the Netherlands a coefficient of –$12,494.

Salaries in Europe were uneven, with differences among European countries as great as those between world regions.

The next group of countries was France, Sweden, Belgium, Finland, and Austria, with a coefficient of –$22,283. Scandinavia was split, Sweden and Finland appearing to have, on average, lower developer salaries than Norway and Sweden. Developer salaries fall as we head into the rest of Western Europe: Spain, Italy, Greece, Portugal, and Turkey had a coefficient of –$35,911. Not far behind, with a coefficient of –$42,594, were countries of Eastern Europe: Poland, Romania, Czech Republic, Ukraine, Hungary, Slovenia, Slovakia, Estonia, and Bosnia and Herzegovina. (Note that a number of countries in the region are not included, since they were not represented in the sample.) Finally, Russia had the lowest salary coefficient in Europe, –$45,224.

It is worth noting that comparing salaries by country can be difficult since currency exchange rates fluctuate; Russia is a good example of this, and had the survey data been collected just a few years ago, the coefficient would have likely been radically different. Many European respondents received substantial raises over the past three years, although a large minority stagnated.

Company Types

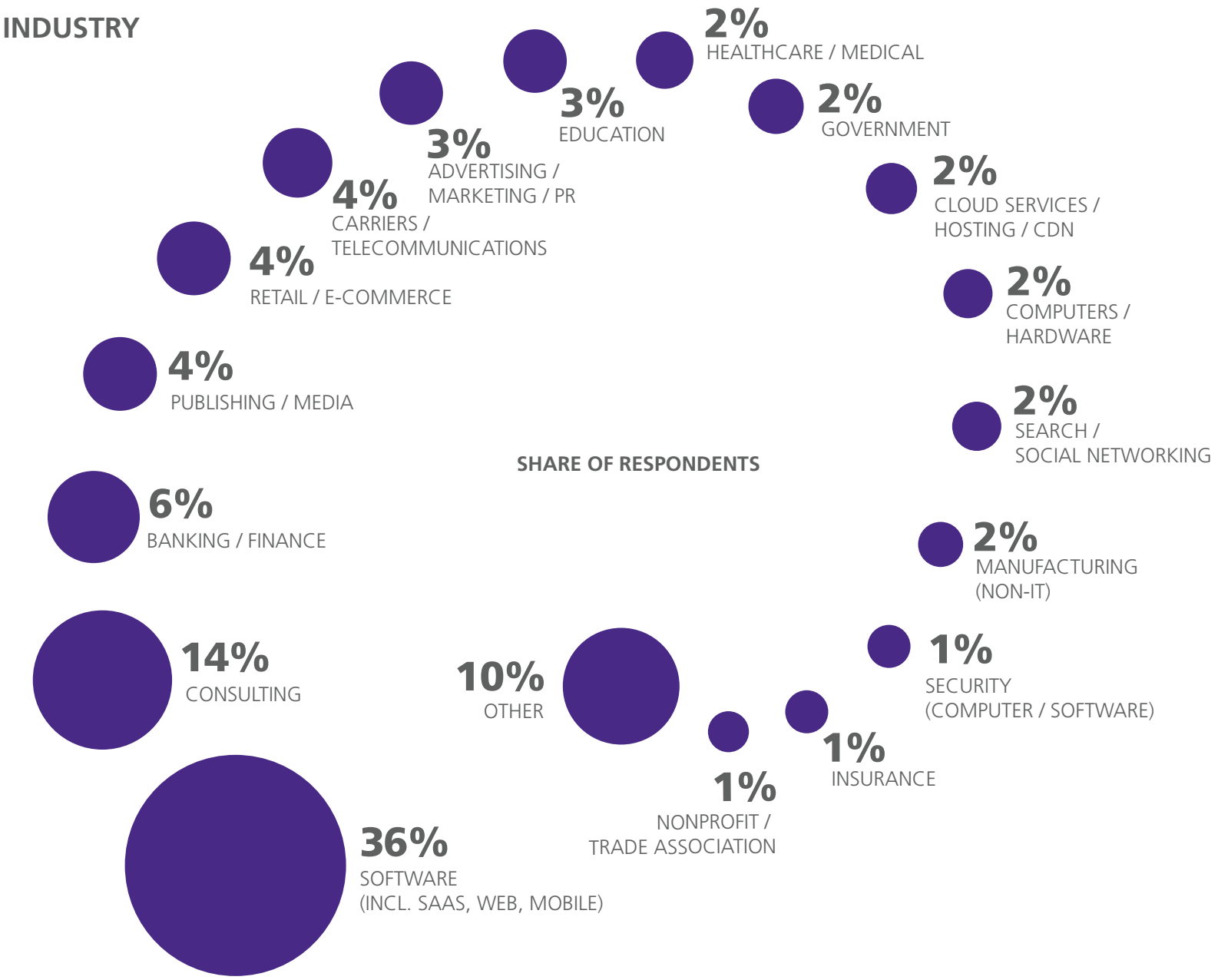
THE SURVEY INCLUDED QUESTIONS ABOUT INDUSTRY, COMPANY SIZE, AND COMPANY AGE. Software was the most well-represented industry (36%, rising to 41% when including cloud services, security, and search/social networking), followed by consulting (14%), and banking/finance (6%). Banking/finance respondents had the highest median salary, $75k, and a model coefficient of +$16,260. The only industry with a negative coefficient was education (–$6,438).

IT consulting (but not non-IT consulting) had a positive coefficient (+$8,419), and combined with the +$8,832 coefficient for self-employment (i.e., company size equals one) paints a favorable picture of solo consulting (2% of the sample were self-employed consultants). But it should be noted that these coefficients may simply be offsetting further coefficients such as the one for team size, which favors larger teams.

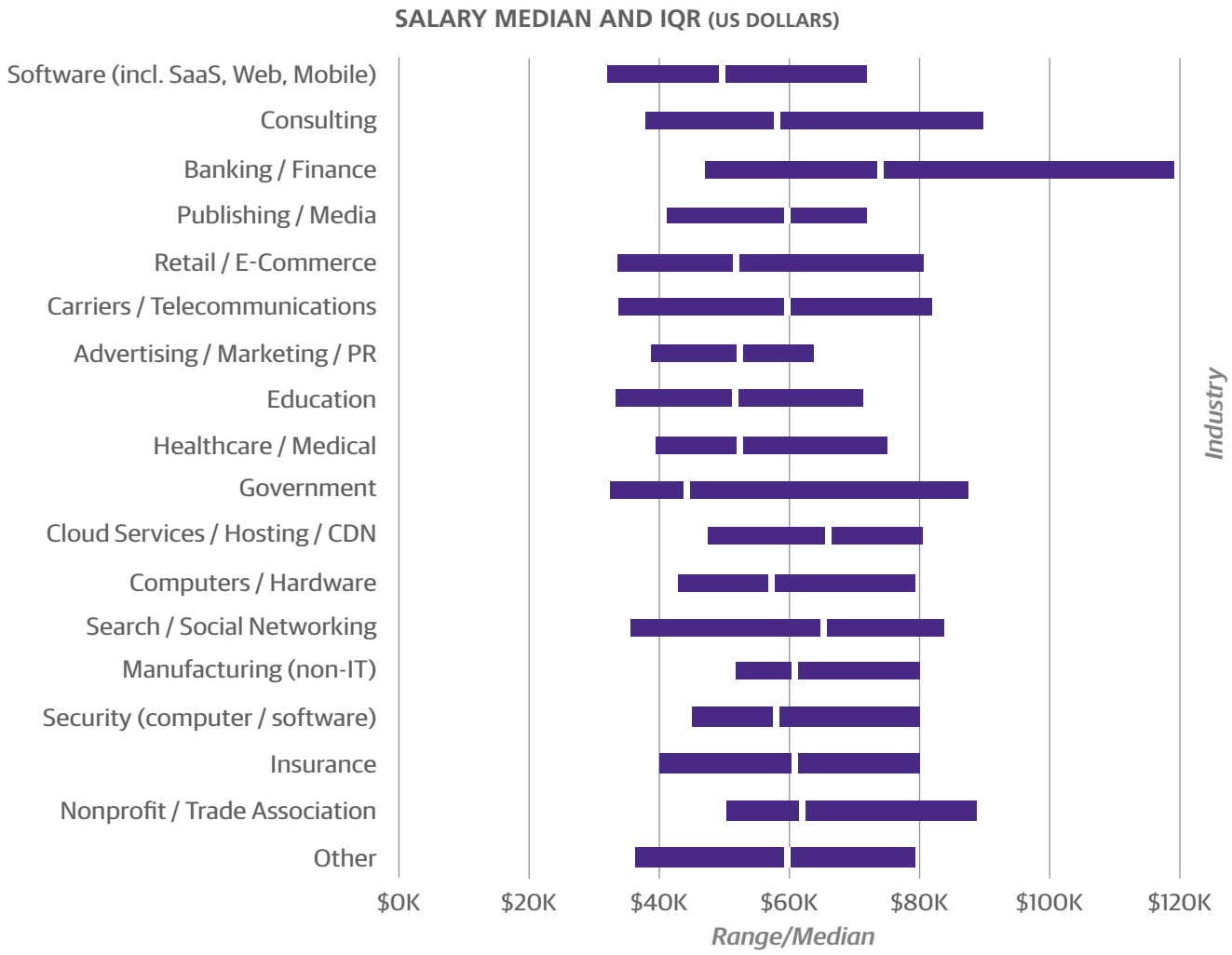
Salary distinctions among companies by age (e.g., start-ups versus mature companies) were subtle enough to be ignored by the model.

Very large companies (over 10,000 employees) made up 12% of the sample and had a median salary of $70 and a coefficient of +$5,156. Old companies (over 20 years old) made up 32% of the sample, and although respondents from these companies had a higher median salary ($63k) than respondents from younger companies, company age over 20 years did not have a coefficient in the model; in other words, the salary discrepancy of this group is likely due to other variables. While company size and age correlate (larger companies tend to be older), the exceptions to this pattern highlight why the previously listed coefficients were chosen: respondents from small, old companies had a median salary of $47k (14% of the sample).

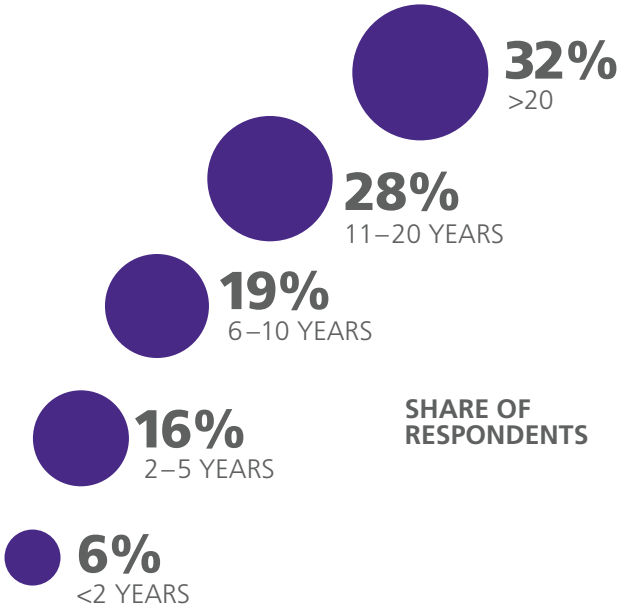
INDUSTRY


2%
HEALTHCARE / MEDICAL
3%
EDUCATION
2%
GOVERNMENT
3%
ADVERTISING / MARKETING / PR
2%
CLOUD SERVICES / HOSTING / CDN
4%
CARRIERS / TELECOMMUNICATIONS
4%
RETAIL / E-COMMERCE
2%
COMPUTERS / HARDWARE
4%
PUBLISHING / MEDIA
2%
SEARCH / SOCIAL NETWORKING
SHARE OF RESPONDENTS
6%
BANKING / FINANCE
2%
MANUFACTURING (NON-IT)
14%
CONSULTING
10%
OTHER
1%
SECURITY (COMPUTER / SOFTWARE)
1%
INSURANCE
1%
NONPROFIT / TRADE ASSOCIATION
36%
SOFTWARE (INCL. SAAS, WEB, MOBILE)

INDUSTRY

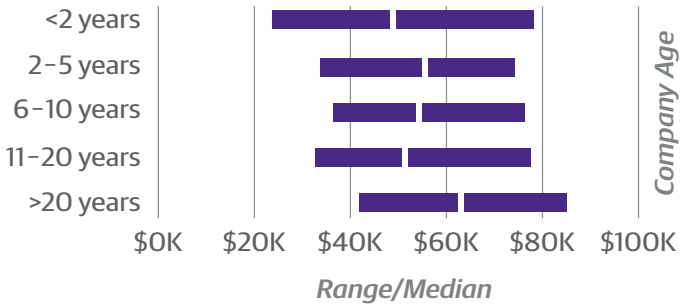

SALARY MEDIAN AND IQR (US DOLLARS)
Software (incl. SaaS, Web, Mobile)
Consulting
Banking / Finance
Publishing / Media
Retail / E-Commerce
Carriers / Telecommunications
Advertising / Marketing / PR
Education
Healthcare / Medical
Government
Cloud Services / Hosting / CDN
Computers / Hardware
Search / Social Networking
Manufacturing (non-IT)
Security (computer / software)
Insurance
Nonprofit / Trade Association
Other
Industry
$0K
$20K
$40K
$60K
$80K
$100K
$120K
Range/Median

COMPANY AGE

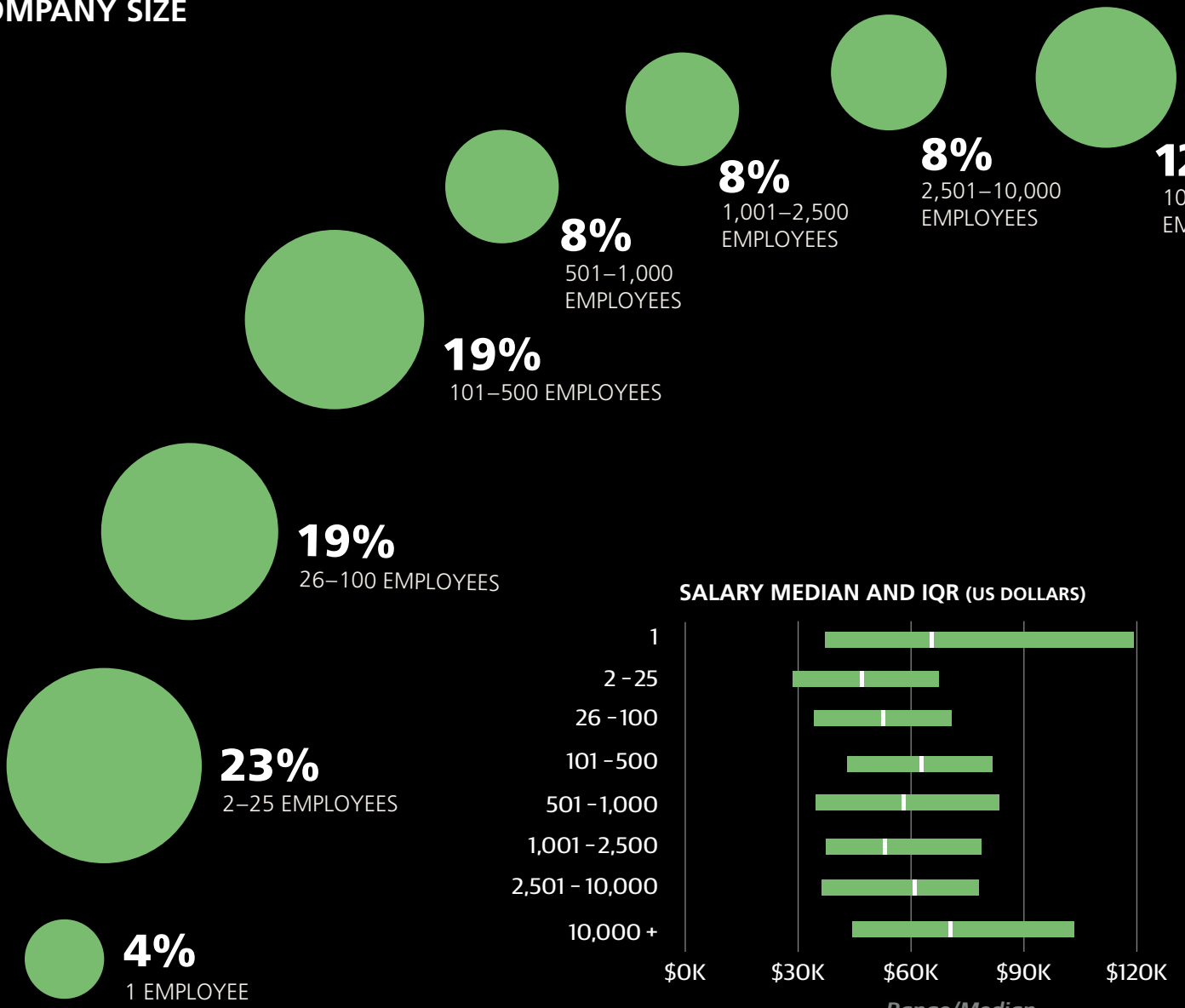


SHARE OF RESPONDENTS

SALARY MEDIAN AND IQR (US DOLLARS)

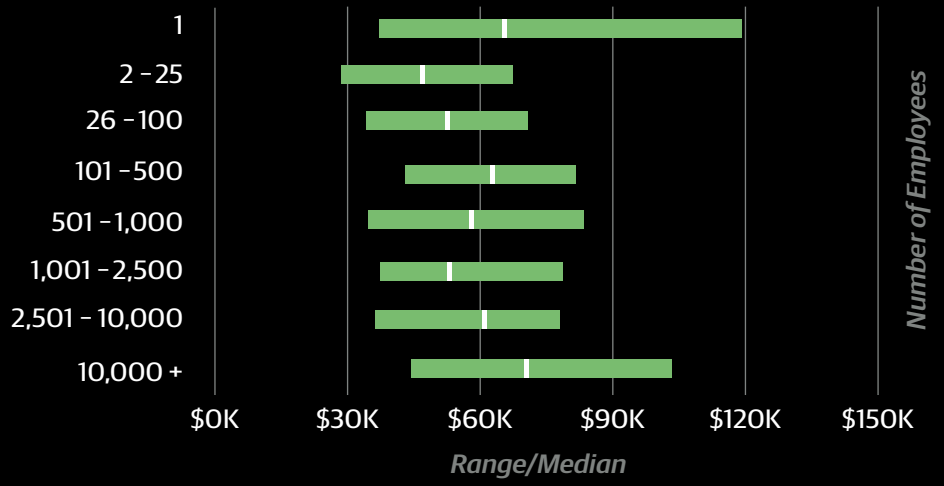


COMPANY SIZE



SHARE OF RESPONDENTS

SALARY MEDIAN AND IQR (US DOLLARS)



Team Structure

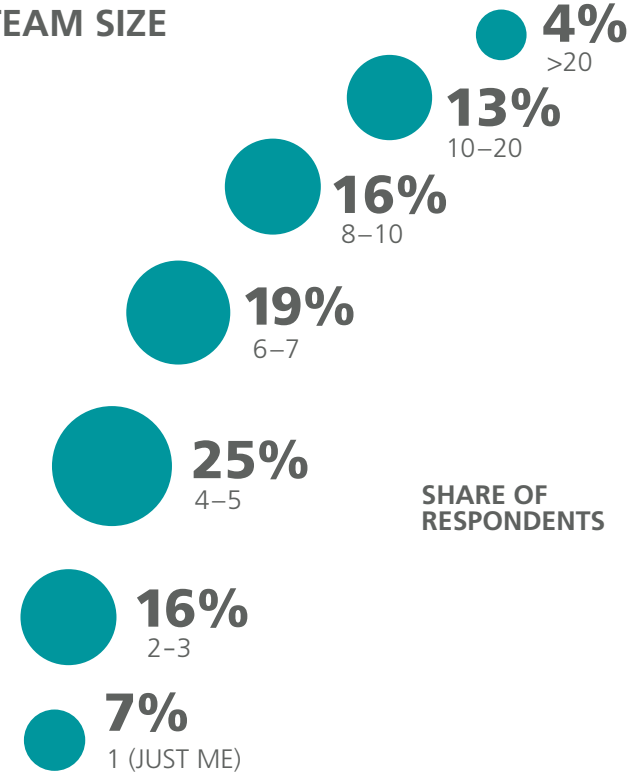
SEVERAL QUESTIONS ON THE SURVEY FOCUSED ON TEAM STRUCTURE, the most basic of which was how many people work on the respondent's team. Salary appears to steadily increase with team size, and with this variable the coefficient is not binary but multiplicative, equal to +$184 times the number of team members.

A slightly different team metric is the size of a team for a typical coding project. The median project team size was 4, with 31% of the sample reporting their typical project team size to be over 5 people. No variables based on answers to this question were significant in the model.

The median project team size was 4, with 31% of the sample reporting their typical project team size to be over 5 people.

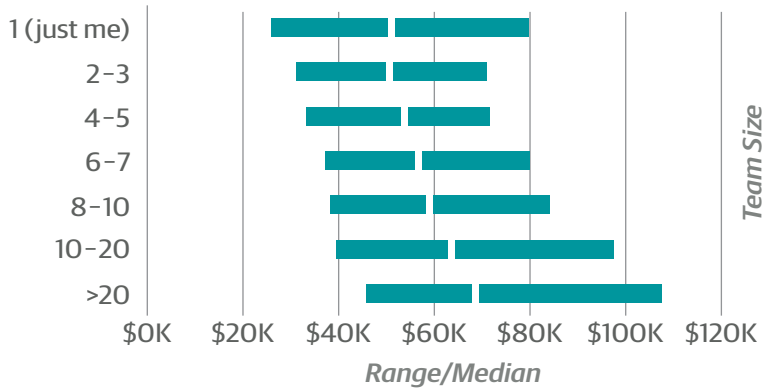
Another question about team structure was whether the respondent worked with people in various roles. Most respondents reported that they work with (other) programmers (89%), product managers (72%), and designers (58%), while 37% said they work with salespeople. The only variable from this question with a positive coefficient was for other programmers, of +$5,332. The small share of respondents (2%) who did not work with people in any of the above roles had a median salary of $37k.

TEAM SIZE



SHARE OF RESPONDENTS

SALARY MEDIAN AND IQR (US DOLLARS)



Individual Background

WE NOW MOVE ON TO DETAILS ABOUT INDIVIDUAL RESPONDENTS.

Gender

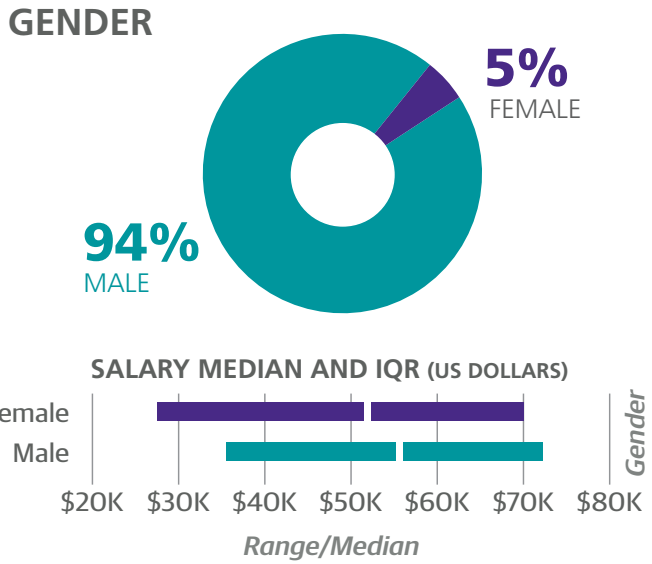
The sample was overwhelmingly male (94%), a breakdown even more skewed than the worldwide results of the survey (where 91% were male). Women in the sample earned less than men, with median salaries of $52k and $56k, respectively, but there was no coefficient for gender included in the model.

Education

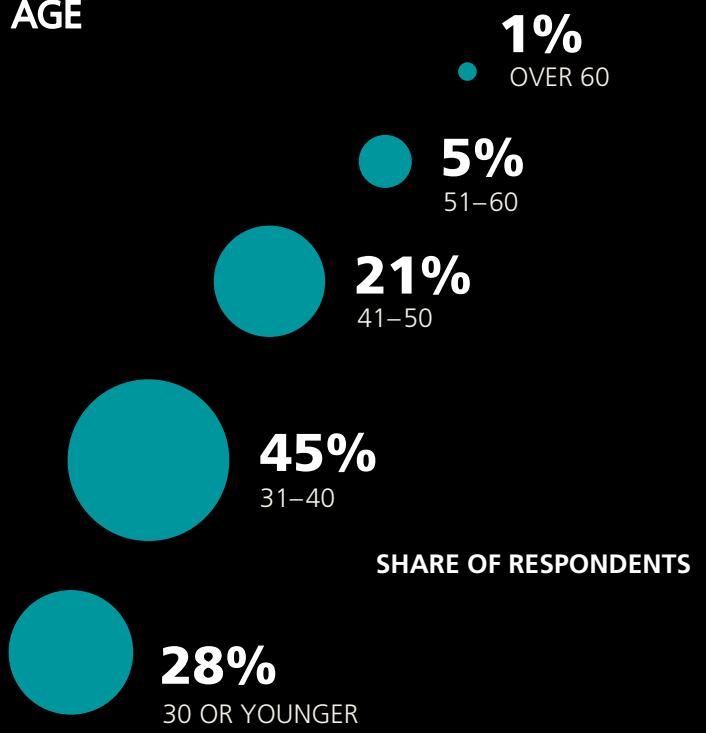
A majority of respondents (56%) had an academic specialization in computer science and 13% had a background in mathematics, statistics, or physics, but no particular specialization was significant in the model. Having a Master's degree (of any discipline, but we assume most were CS or something technical) is also not significant in the model, but a PhD adds +$7,906.

Age and Experience

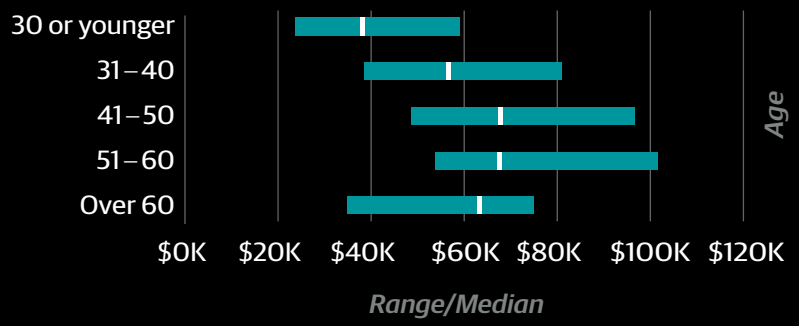
The age range was skewed toward youth: over 60% of the sample was under 40. Salary increased with age, the most well-paid demographic being the 56–60 cohort who earned a median of $71k (followed closely by those aged 41–45). However, we also asked about years of experience, and this appeared to be the actual predictor of salary: given a certain level of experience, age is no longer a factor and thus did not have any associated coefficients. According to the model, developers can expect an additional +$1,257 of pay per year of experience. This is independent of title, role, and tasks, which the model shows affecting salary in different ways (discussed next).



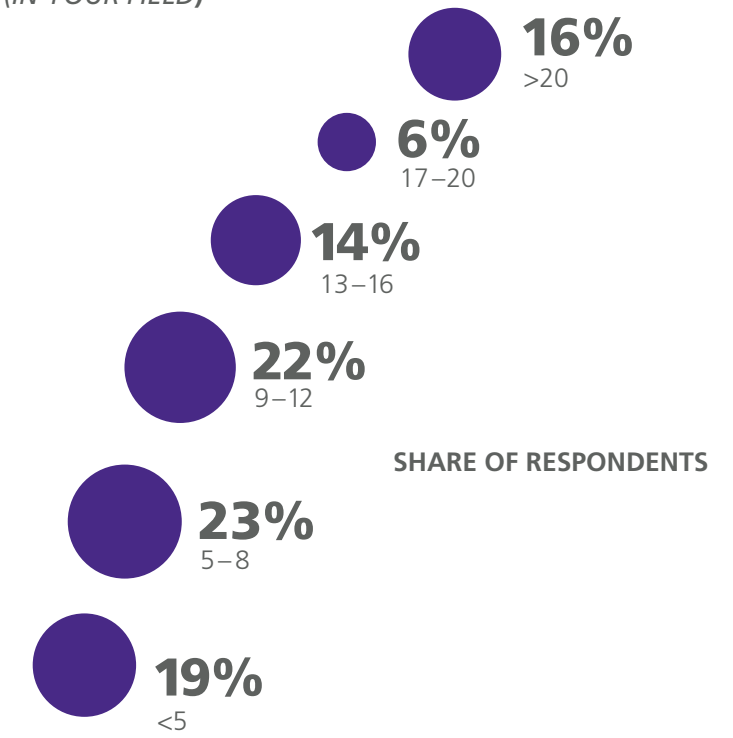
AGE



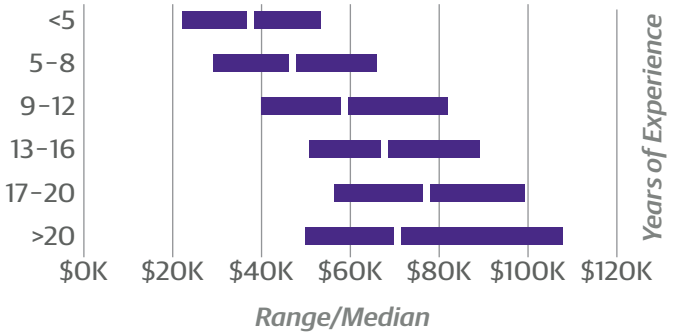
SALARY MEDIAN AND IQR (US DOLLARS)



YEARS OF EXPERIENCE
(IN YOUR FIELD)



SALARY MEDIAN AND IQR (US DOLLARS)



Title, Role, and Tasks

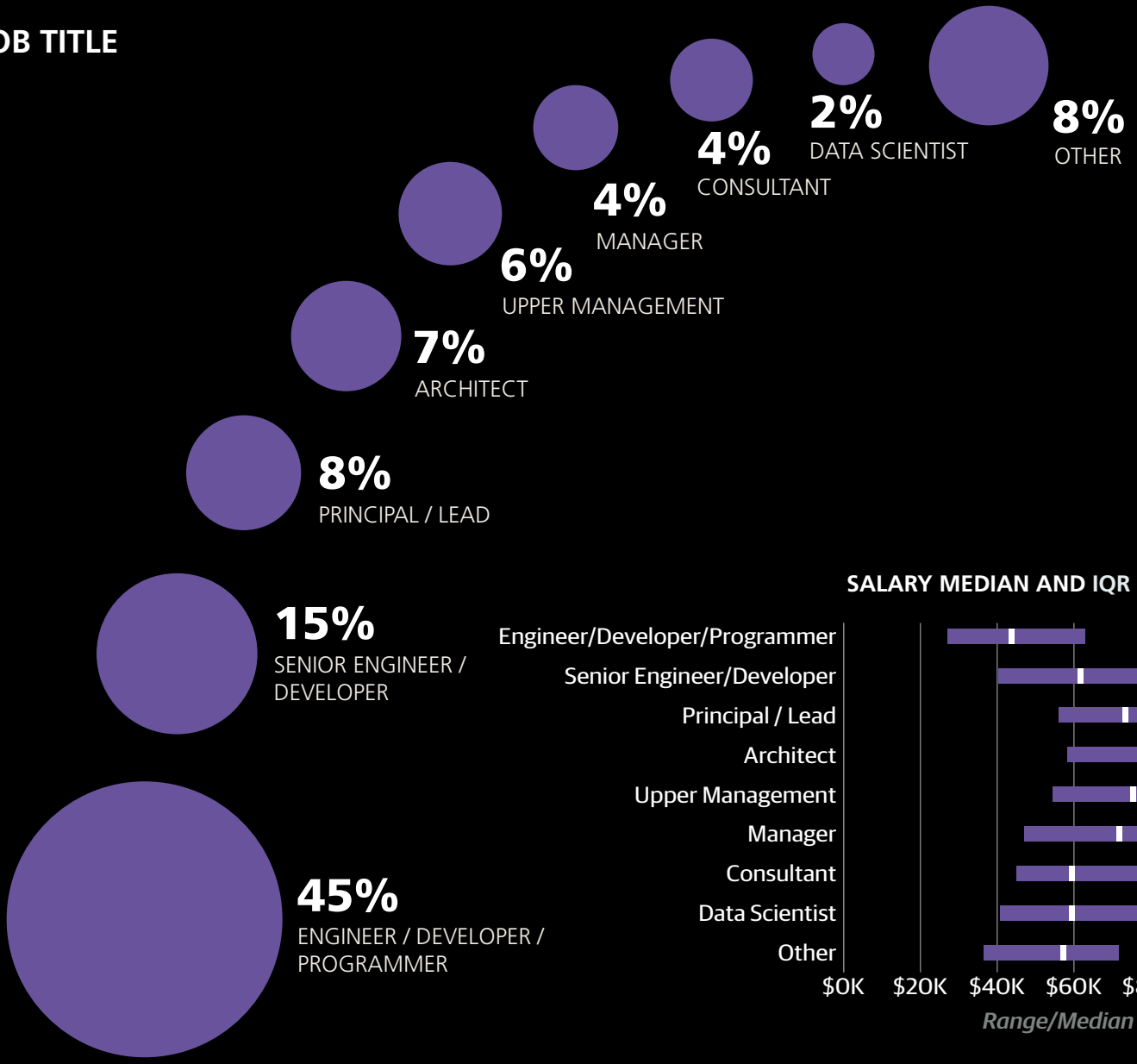
WE TOOK TWO DISTINCT APPROACHES to defining the roles of respondents. The first was a text field for job title, which we parsed to assign respondents to a category. The most common (cleaned) title was Engineer/Developer/Programmer, with 45% of the sample. Engineers or developers with "Senior" in their title made up a further 15% of sample. Two titles were given positive coefficients: Principal/Lead (8% of the sample, for +$6,254) and Architect (7%, for +$10,990). As mentioned at the start of this report, managers and students were excluded from the model, so there were no coefficients associated with them.

The second approach to capturing respondents' roles was to ask whether they engaged in specific tasks. The three possible answers to each of the 16 task questions was "no involvement", "minor involvement", and "major involvement", which was defined as a task that "is essential to most or all of your projects and responsibilities, and that you perform frequently (most days)".

The two tasks with the greatest involvement were writing code for collaborative projects (72% major, 21% minor) and reading/editing code originally written by others (61% major, 32% minor). Even though neither of these tasks had associated coefficients, their high engagement rates highlight the importance of collaboration in software development: it is often a very social activity.

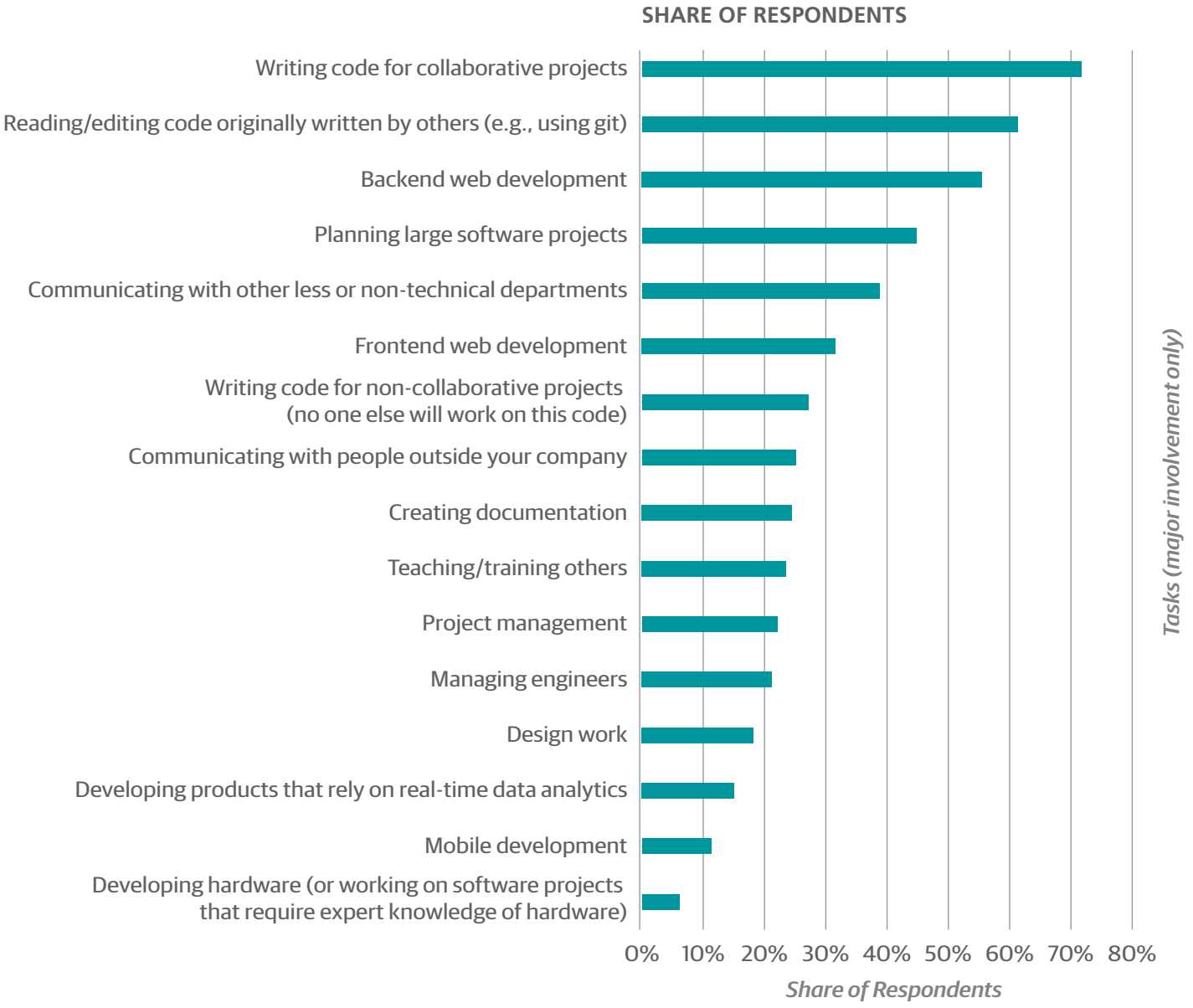
Back-end web development was also very common (56% major, 26% minor), more than front-end web development (32% major, 38% minor) or mobile development (11% major, 26% minor), while only 16% of the sample had no involvement in web or mobile development. The coefficients related to these development distinctions were all penalties: major involvement in mobile development had a coefficient of –$3,593 and lack of involvement in back-end web development had a coefficient of +$3,606.

JOB TITLE

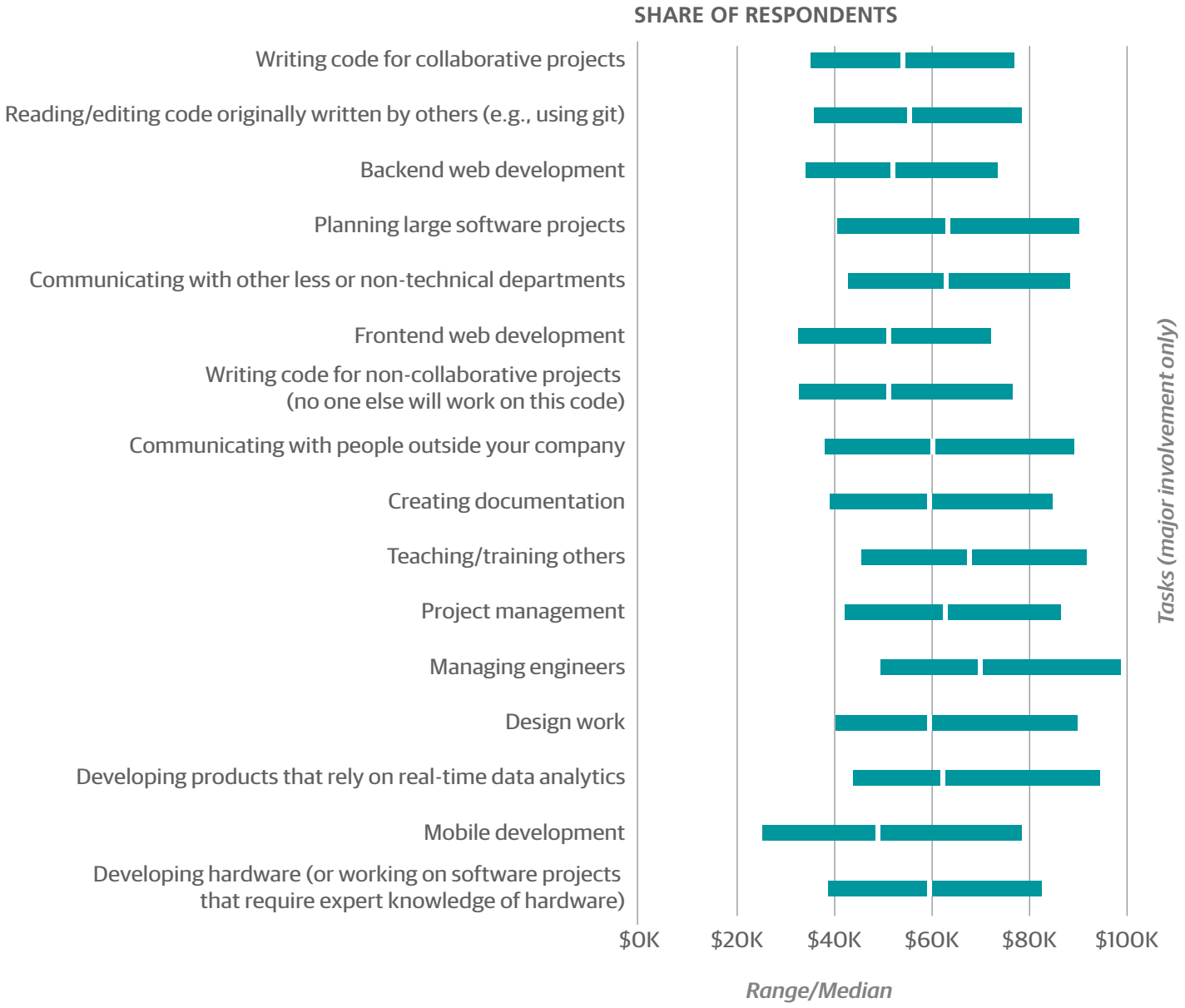


SHARE OF RESPONDENTS

TASKS
(MAJOR INVOLVEMENT ONLY)



TASKS
(MAJOR INVOLVEMENT ONLY)



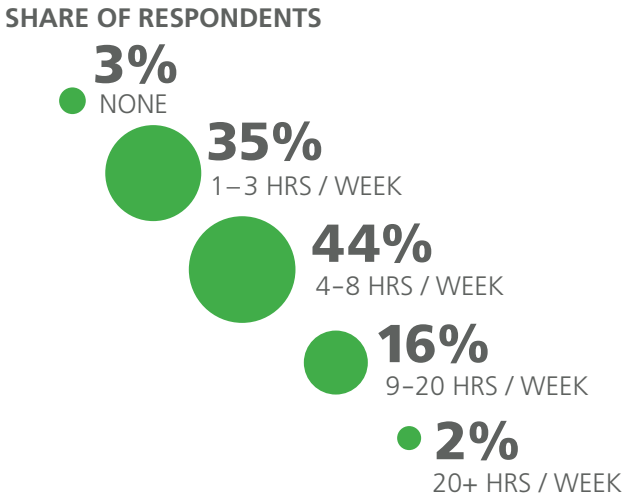
"Planning large software projects" was a task that may seem to be synonymous with architect (a job title category), but some respondents selected more than one task, meaning that the tasks appeared to be interpreted quite broadly. Thus, a full 45% of the sample (most of whom were not architects) reported major involvement in planning large software projects.

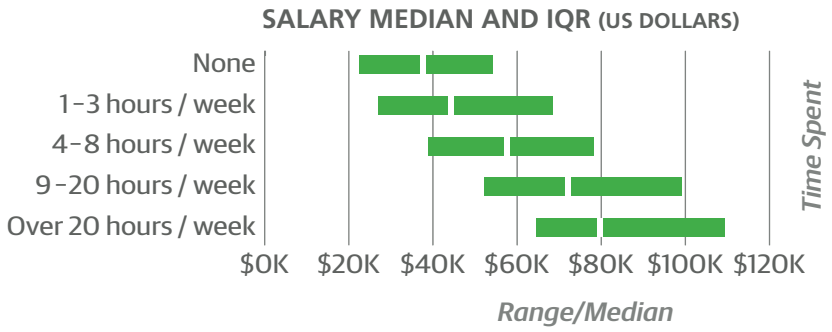
We did not use tasks to determine who was a manager and therefore should be excluded from our model; we used job title for that." A modest coefficient was produced for major involvement in teaching or training others: +$3,499.

Even with questions about management, title, and years of experience, it is difficult to obtain a reliable metric of "level", the track of vertical career advancement that, we assume, plays an integral part in determining salary. Variations in team and management structure, and inconsistencies in title distinctions (e.g., "senior", "staff", "principal") contribute to this fuzziness. One variable that we have found serves as a decent proxy for level is the number of hours spent in meetings. The coefficient of +$150 per weekly meeting hour can be added in addition to any other management- or level-related features. If we consider those professionals who spend somewhere around half of their time in meetings (2% spent over 20 hours/week in meetings), this coefficient can easily dwarf most other coefficients as a contribution to salary estimate. As with the worldwide pattern, salaries in Europe tend to go down with time spent coding, which makes sense because time spent in meetings tends to raise salaries.

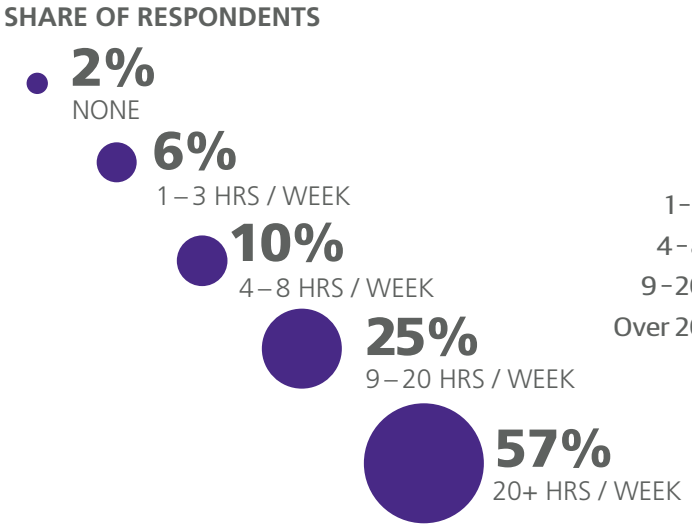
TASK COUNTS

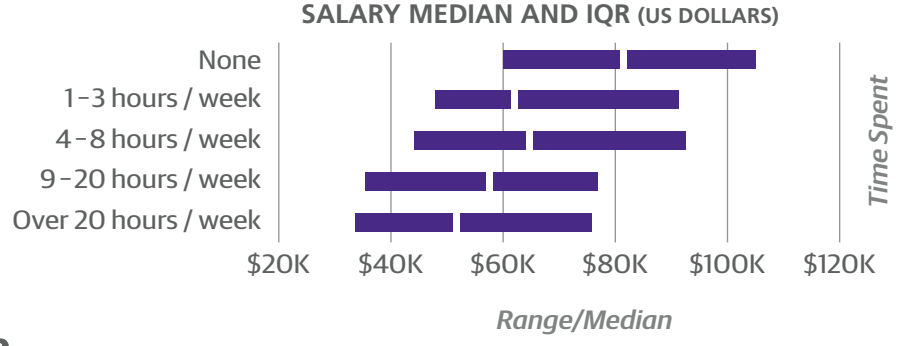
TIME SPENT IN MEETINGS (HOURS PER WEEK)





TIME SPENT CODING (HOURS PER WEEK)





Tools

Eight categories of tools were included as binary questions on the survey; respondents simply marked the ones that they currently use in a professional context. The tool categories were operating systems, programming languages, text editors, IDEs, data tools, cloud/containers, build automation tools, and frameworks.

On average, respondents used 3.6 programming languages and 16 tools of any kind. Less than 3% of the sample used fewer than 6 tools, while 19% used at least 20. Some tools seemed to encourage a larger toolkit: respondents who used Scala, Objective-C, Kubernetes, Google App Engine, Go, Groovy, YAML, Cassandra, Solr, or Spark used 21–23 tools on average.

On average, respondents used 3.6 programming languages and 16 tools of any kind.

It is interesting to note that Vim remains by far the most popular text editor, and IntelliJ is the most popular IDE (a lot higher than Eclipse or Visual Studio). MySQL still rules in databases, with PostgreSQL barely coming out better than Excel in popularity. PostgreSQL pays slightly better than MySQL. But the high salaries tend to be with NoSQL and cloud-related technologies: Hadoop, Spark, MongoDB, Cassandra, etc. These do even better than Oracle.

Instead of feeding individual tools into the model (which would result in a small selection of them being chosen as model coefficients), we instead have first built clusters of the most frequently used tools. The motivation behind this is that tools are often highly correlated with one another. (Operating systems were excluded from the clusters.)

The 18 clusters were formed using the Affinity Propagation algorithm in Python's scikit-learn module, with a transformation

of the correlation coefficients between pairs of tools serving as the similarity metric. The essential idea is simple: for two tools in a cluster, if a respondent used one, she is more likely to use the other as well. It should be noted that this is not necessarily true for every pair of tools in every cluster. Many respondents used tools from multiple clusters, and there were large, positive correlations between a number of tools in different clusters. However, the clusters we produced represent an efficient series of compromises for all of the irregularities in tool co-usage, and provide a decent picture of which tools tend to be used by the same people.

The salary estimate contribution for each cluster is obtained by multiplying a cluster's coefficient by the number of tools you use from the cluster.

The salary estimate contribution for each cluster is obtained by multiplying a cluster's coefficient by the number of tools you use from the cluster. The algorithm produces the following clusters (we've named the clusters—in parentheses—as a mnemonic device):

The first cluster is focused on Javascript, the most commonly used tool (other than operating systems). On average, respondents used 2.3 tools from this cluster, and 31% did not use any tools from the cluster. There was no coefficient associated with Cluster 1 in the model.

The second cluster is a Java stack, although notably languages that run on the JRE—such as Scala and Clojure—are in their own clusters (13 and 15). Cluster 2 does not have a model coefficient, because using more or fewer of these tools does not appear to affect salary after all of the variables are

taken into account. OracleBI seems an outlier to an otherwise list of mostly Java tools, so the correlation of OracleBI to the other tools may indicate more enterprise (i.e., big-organization) use of Java.

Cluster 3 is composed of three tools, Sublime Text, MongoDB, and Grunt. At least two of these are related to web development. They have no impact on salary in the model, and no coefficient.

Cluster 4 is composed of Vim, Bash, Perl, and Go. It has a positive coefficient of +$2,636.

Cluster 5 has a cloud/container theme, containing tools such as AWS and Docker. It has a coefficient of +$698. This cluster shows that the Cloud is used for app development; most of the tools in the cluster are used by web engineers to manage complex, cloud-based web application platforms, and also reflect the move to containers for portable cloud deployment.

Cluster 6 is typified by Python. Cluster 6 has a coefficient of –$1,234. Respondents used Python for a variety of programming tasks, as reflected in the tools in this cluster, including data management, data analysis, and web applications.

Cluster 7 consists primarily of Microsoft products (in particular those in the .NET framework), and has a negative coefficient of –$1,114. The popularity of Excel in our survey suggests that Excel is an integral part of .NET developers' toolbox, for analytics, data query, data investigation, and other uses.

Cluster 8 has the second largest negative coefficient: –$2,206. This is the PHP/MySQL stack, one that may be going somewhat out of style. Still, 44% of the sample used at least one tool from this cluster. This cluster appears to be a silo with no strong correlation with other web development tools—an indication that PHP/MySQL represents a separate web development path for the survey respondents.

Cluster 9, like Cluster 8, consists of tools that mostly support building web apps, in this case Ruby on Rails. This cluster did not have a coefficient.

Cluster 10 is an odd cluster, consisting of the gedit text editor and the Code::Blocks IDE. This cluster had a coefficient of –$3,485. It should be noted that since this is a small cluster, it is not surprising that the coefficient is larger in magnitude than

other coefficients in the model because the model multiplies the coefficients by the number of tools used in the cluster.

Cluster 11 is centered on data processing and analytics, in particular those that process big data: Solr, Hadoop, and Spark. Cluster 11 has a small but positive coefficient of +$173. In the global sample—and in other salary surveys we have conducted—these tools generally have a much bigger (positive) impact on salary.

Cluster 12 consists of three text editors, suggesting that independent of languages and platforms, these editors are often used by the same people. For example, someone who uses TextWrangler is more likely to use jEdit than someone who does not use TextWrangler. It should be mentioned that TextWrangler was much more popular than the other two, jEdit and Komodo Edit, so the positive correlations between these tools, while significant, is caused by a relatively small group of respondents. This cluster had a large, positive coefficient of +$3,149.

Cluster 13 includes Scala and Cassandra. This cluster (somewhat curiously) is separate from Cluster 11, suggesting that in Europe Scala is used frequently in contexts other than big data. (Among US respondents the co-usage between Scala and Spark, for example, is much more significant.) The coefficient associated with this cluster was +$1,112.

Cluster 14 is an Apple/iOS cluster, with tools such as Objective-C, Swift, and Xcode. The positive coefficient of +$2,262 suggests iOS development is relatively well rewarded.

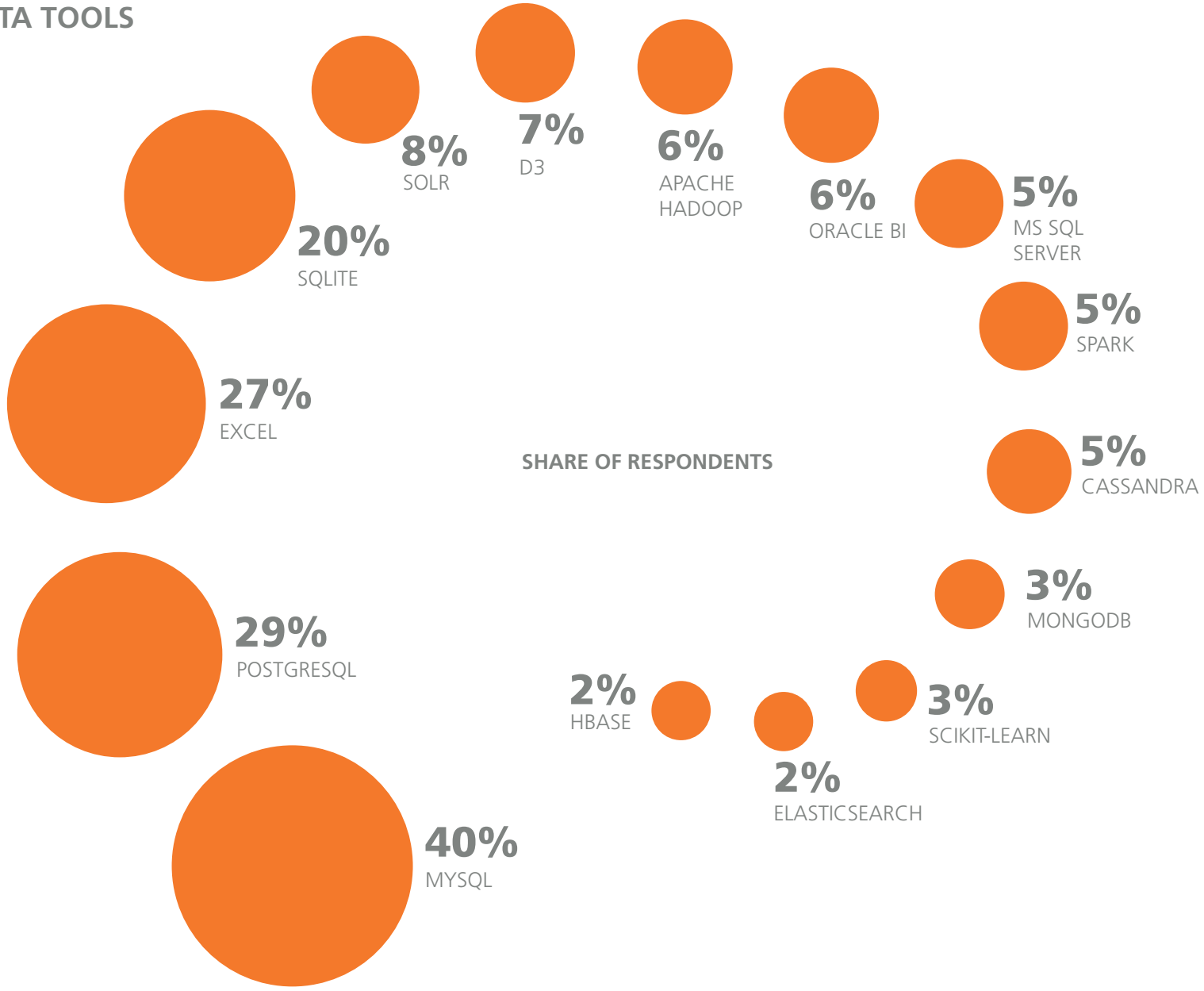
Cluster 15 consists of Emacs, Leiningen, and Clojure, and has a positive coefficient of +$945.

The remaining three clusters (16-18) had relatively few users and no significant coefficients. As stated above, respondents generally used tools from multiple clusters: the fact that a tool ended up in a small cluster does not indicate it is in a tiny silo, but rather that it is not specifically tied to any of the larger, more prominent stacks.

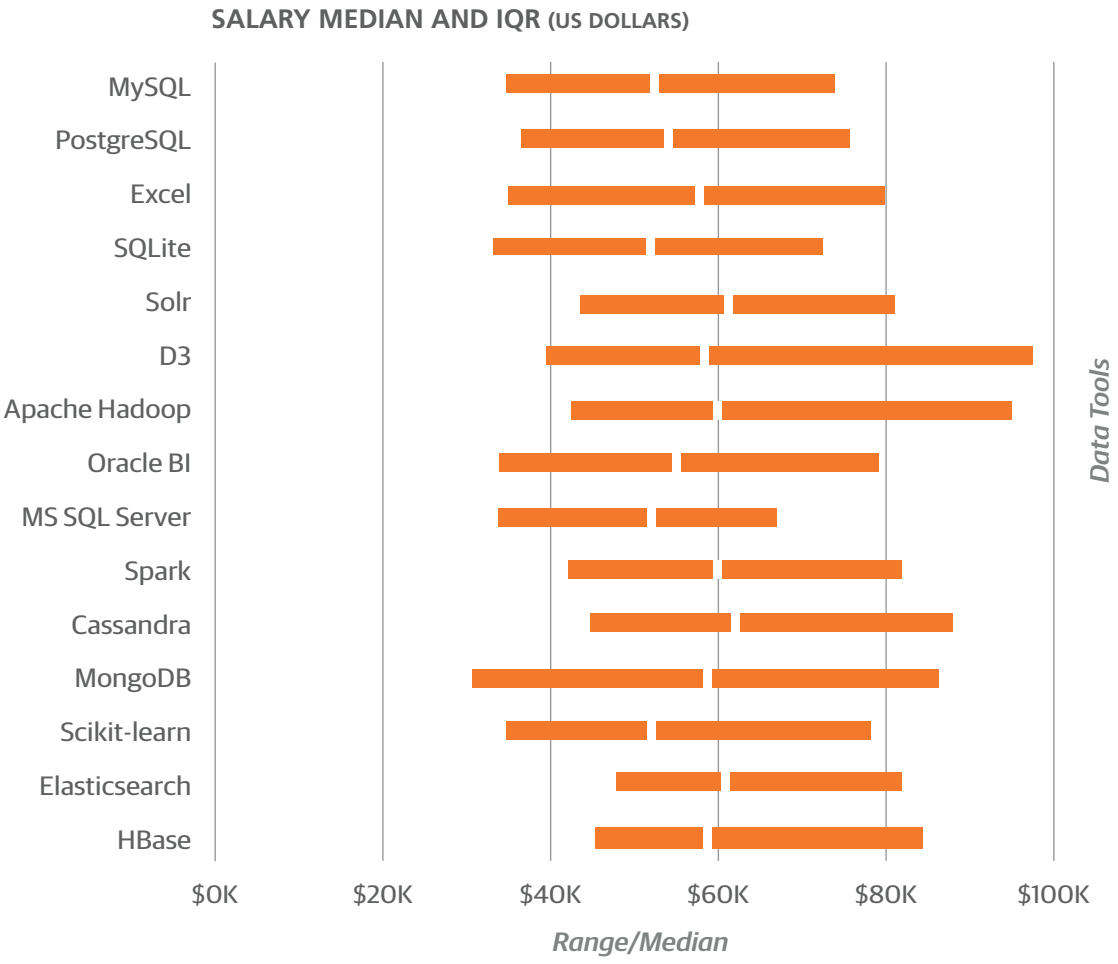
Cluster 1 (JavaScript): not included in model, no coefficient	**Cluster 2 (Java): not included in model, no coefficient**	**Cluster 3 (Web): not included in model, no coefficient**	**Cluster 4 (Unix tools): +$2,636**	**Cluster 5 (Cloud): +$698**	**Cluster 6 (Python): –$1,234**	**Cluster 7 (.NET): –$1,114**	**Cluster 8 (LAMP): –$2,206**	**Cluster 9 (Ruby): not included in model, no coefficient**
JavaScript	Jenkins	Sublime Text	Vim	AWS	Python	Notepad ++	MySQL	Ruby
HTML	Java	MongoDB	Bash	PostgreSQL	SQLite	Excel	PHP	Ruby on Rails
CSS	IntelliJ IDEA	Grunt	Perl 5	Docker	C/C++	Visual Studio	PHPStorm	Heroku
jQuery	Maven		Go	Atom	PyCharm	.NET	Zend Framework	RubyMine
AngularJS	Eclipse		nano	YAML	Django	C#	Symfony	
React	Spring		LISP	Openstack	scikit-learn	Azure	Lua	
D3	Ant		Perl 6	Google App Engine	Haskell	Visual Studio Code	Laravel	
Brackets	Gradle			Kubernetes	IDLE	ASP/ASP.NET	CakePHP	
Gulp	Netbeans			Elasticsearch	Flask	ASP.NET Ajax		
TextMate	Oracle BI			OpenShift	QtCreator	MS SQL Server		
WebStorm	Groovy			Cloud Foundry		TeamCity		
Aptana Studio 3	UltraEdit			Bamboo		VISUAL BASIC .NET		
Geany	Grails			make		F#		
	Vaadin			Processing		MSBUILD		
	Android Studio			Redis		PL/SQL		
	DB2					TFS		
	DigitalOcean					SQL SERVER		

Cluster 10 (Editing 1): –$3,485	**Cluster 11 (Distributed computing): +$173**	**Cluster 12 (Editing 2): +$3,149**	**Cluster 13 (Scala): +$1,112**	**Cluster 14 (Apple): +$2,262**	**Cluster 15 (Clojure): +$945**	**Cluster 16 (Erlang): not included in model, no coefficient**	**Cluster 17 (R): not included in model, no coefficient**	**Cluster 18 (Scite): not included in model, no coefficient**
Gedit	Solr	TextWrangler	Scala	Objective-C	Emacs	Erlang	Kate	Scite
Code::Blocks	Apache Hadoop	jEdit	sbt	Swift	Leiningen	Elixir	R	
	Spark	Komodo Edit	Cassandra	Xcode	Clojure			
	Mesos		Play	Coda				
	HBase			SQL				

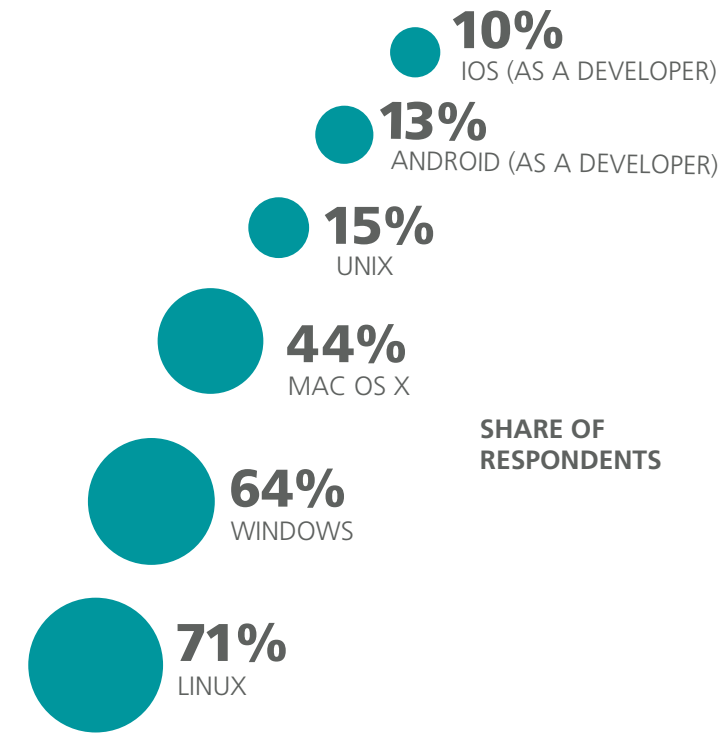
DATA TOOLS


40%
MYSQL
29%
POSTGRESQL
27%
EXCEL
20%
SQLITE
8%
SOLR
7%
D3
6%
APACHE HADOOP
6%
ORACLE BI
5%
MS SQL SERVER
5%
SPARK
5%
CASSANDRA
3%
MONGODB
3%
SCIKIT-LEARN
2%
ELASTICSEARCH
2%
HBASE
SHARE OF RESPONDENTS

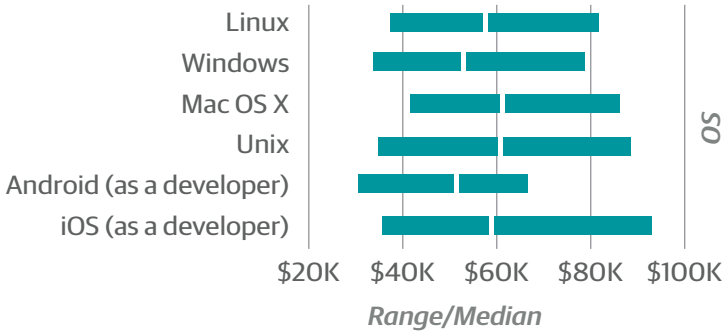
DATA TOOLS


SALARY MEDIAN AND IQR (US DOLLARS)
MySQL
PostgreSQL
Excel
SQLite
Solr
D3
Apache Hadoop
Oracle BI
MS SQL Server
Spark
Cassandra
MongoDB
Scikit-learn
Elasticsearch
HBase
$0K
$20K
$40K
$60K
$80K
$100K
Range/Median
Data Tools

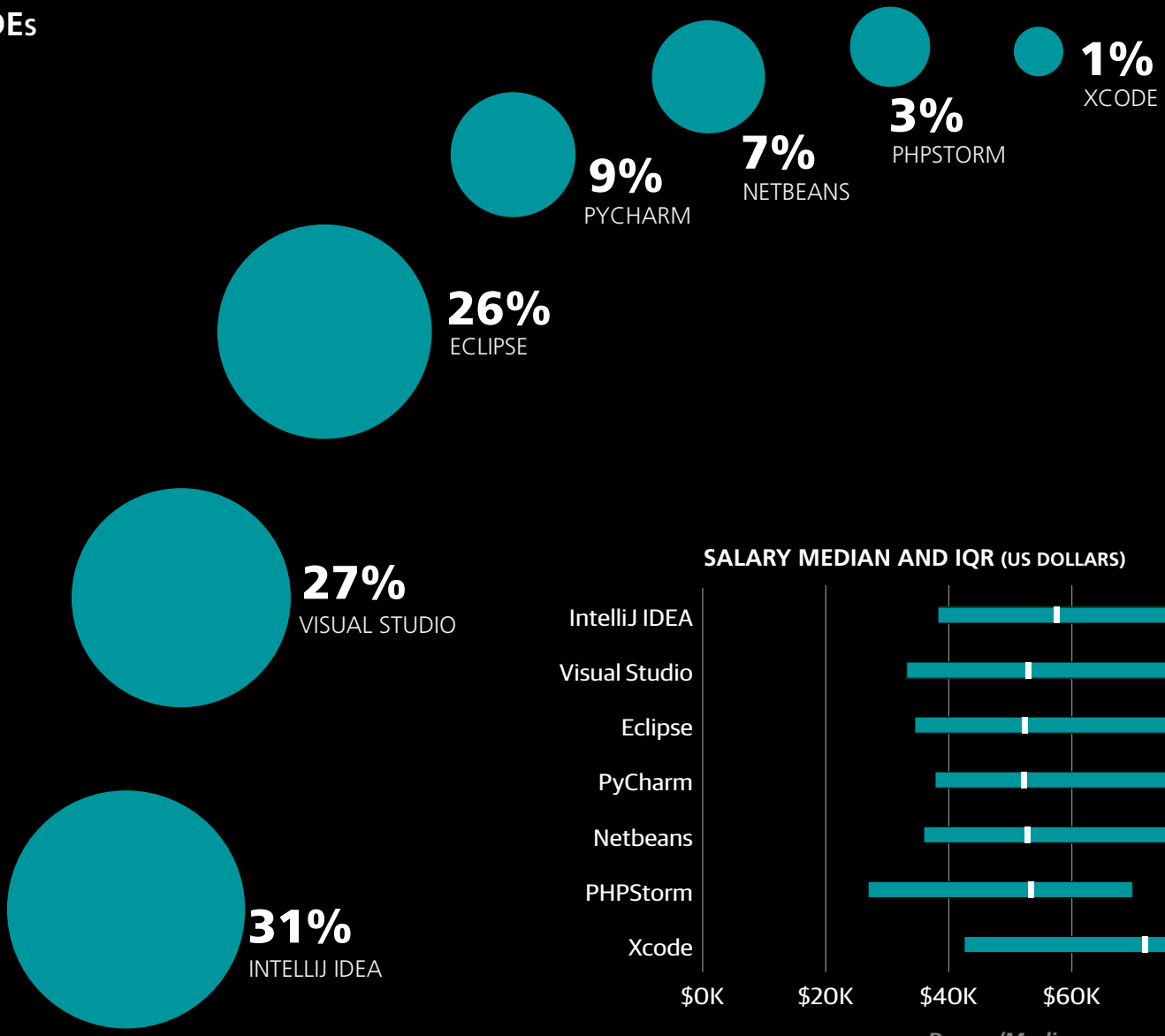
OPERATING SYSTEMS



SALARY MEDIAN AND IQR (US DOLLARS)

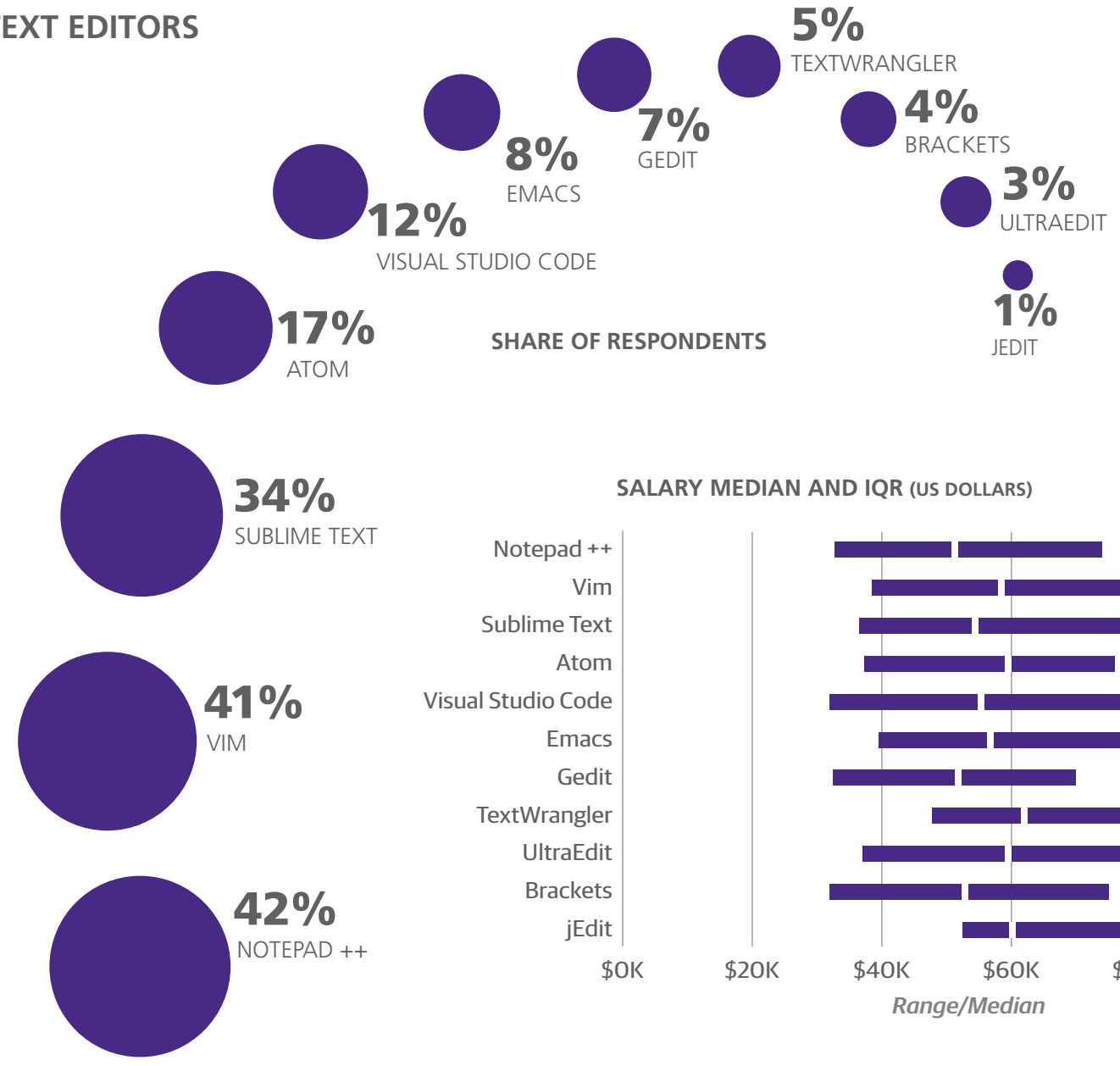


IDEs

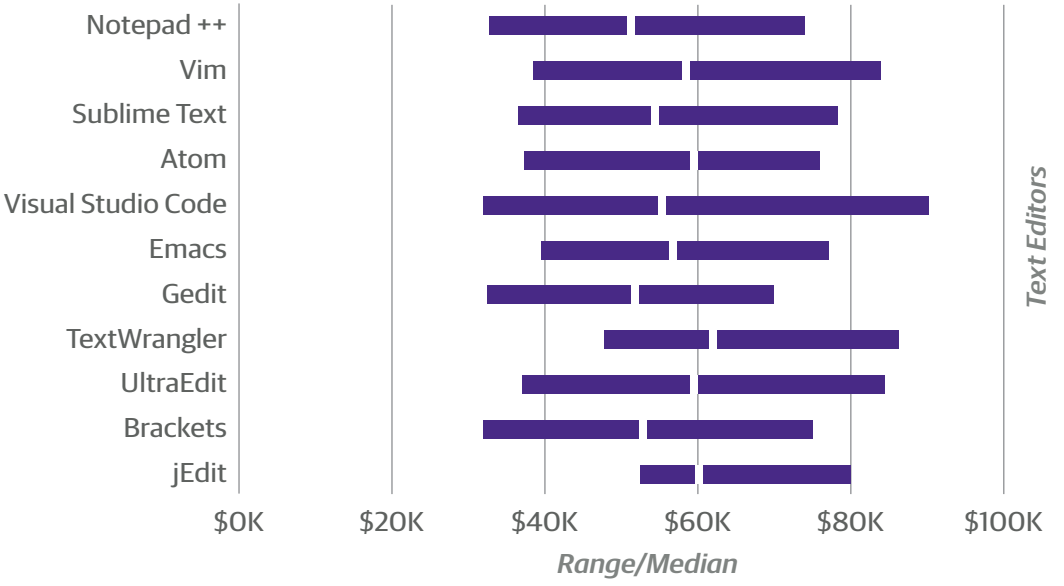


SHARE OF RESPONDENTS

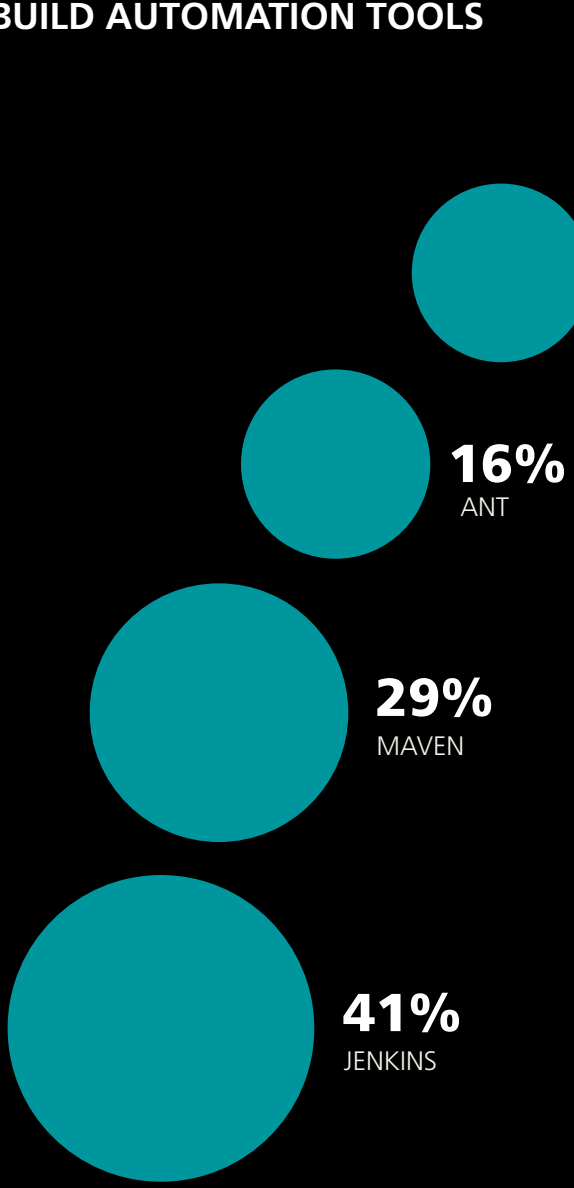
TEXT EDITORS



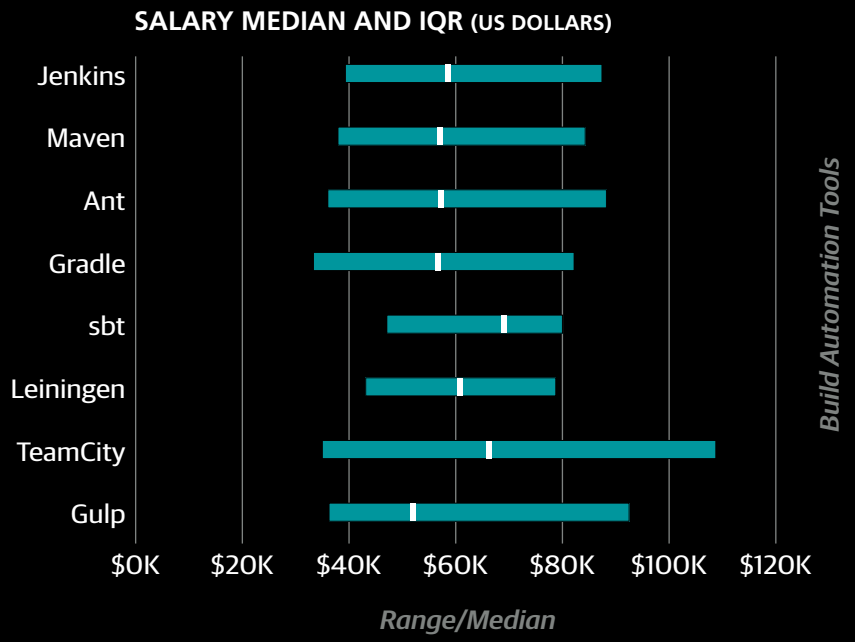
SALARY MEDIAN AND IQR (US DOLLARS)



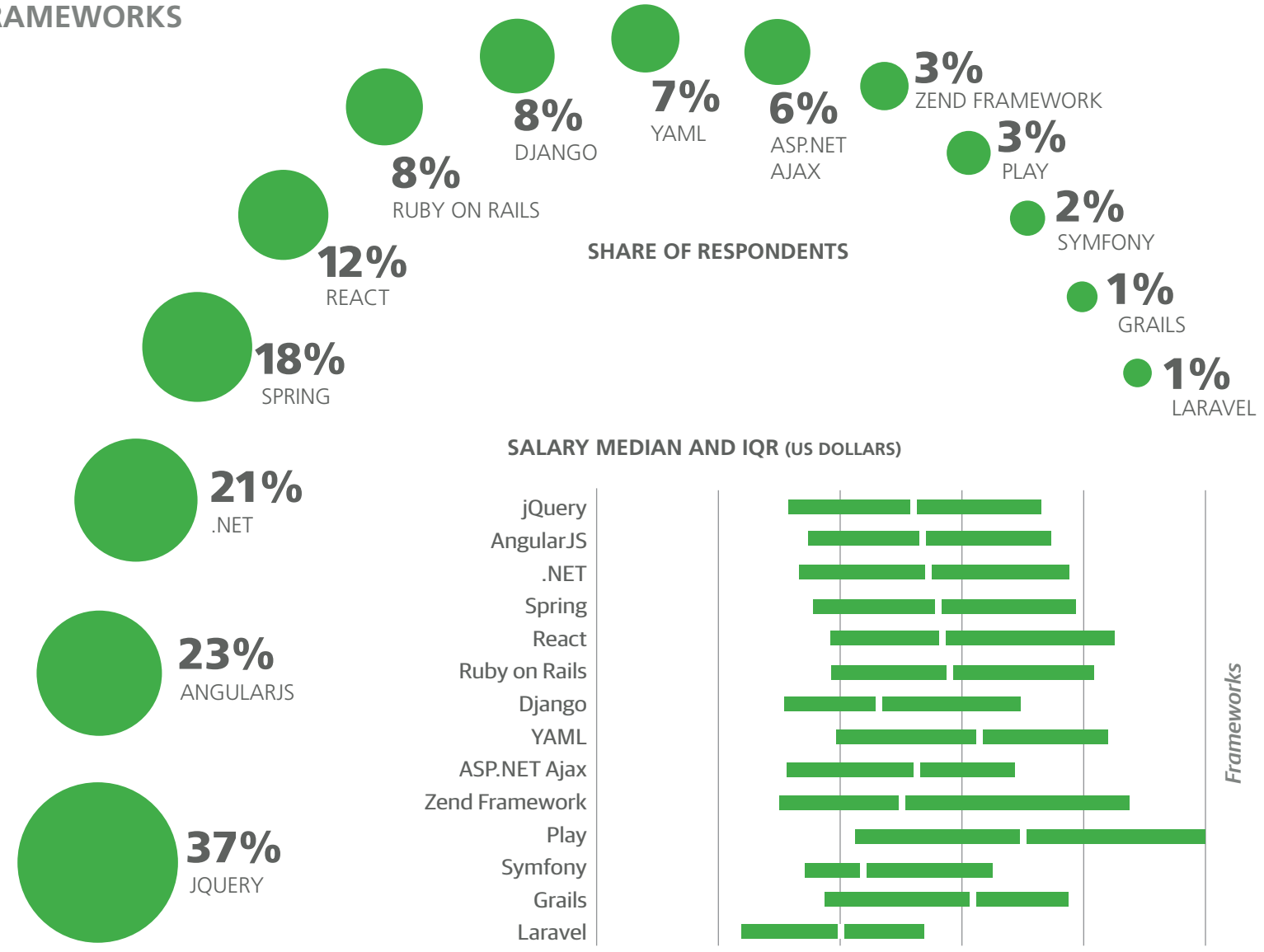
BUILD AUTOMATION TOOLS



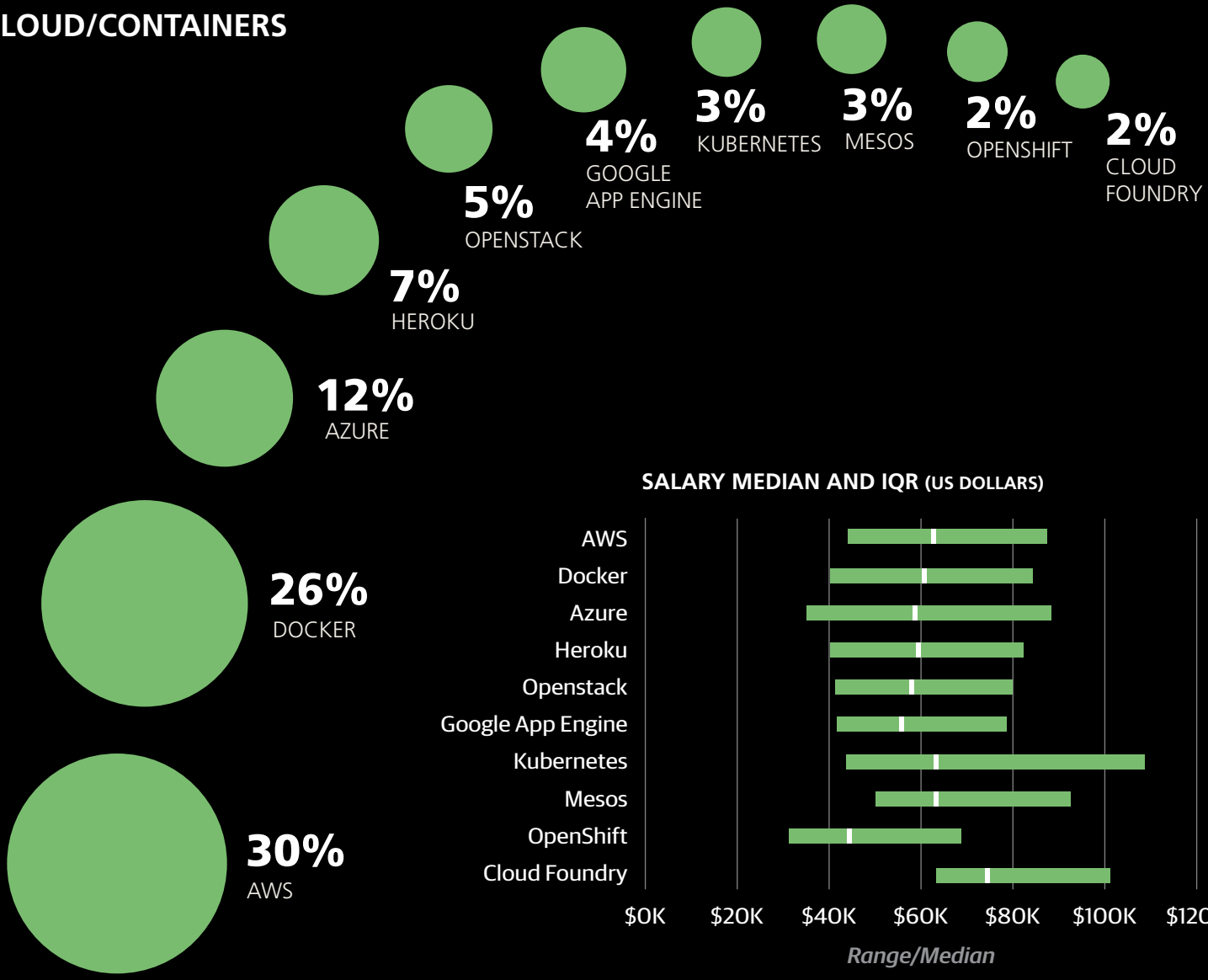
SHARE OF RESPONDENTS



FRAMEWORKS


37%
JQUERY
23%
ANGULARJS
21%
.NET
18%
SPRING
12%
REACT
8%
RUBY ON RAILS
8%
DJANGO
7%
YAML
6%
ASP.NET AJAX
3%
ZEND FRAMEWORK
3%
PLAY
2%
SYMFONY
1%
GRAILS
1%
LARAVEL
SHARE OF RESPONDENTS
SALARY MEDIAN AND IQR (US DOLLARS)
jQuery
AngularJS
.NET
Spring
React
Ruby on Rails
Django
YAML
ASP.NET Ajax
Zend Framework
Play
Symfony
Grails
Laravel
$0K
$20K
$40K
$60K
$80K
$100K
Range/Median
Frameworks

CLOUD/CONTAINERS



SHARE OF RESPONDENTS

Programming Languages

PROGRAMMING LANGUAGES NATURALLY PLAYED A CENTRAL ROLE IN OUR SURVEY, and turned up important coefficients to explain salary differences. They also showed some correlations with other variables.

Correlations

A few differences in programming language by country and other variables are worth pointing out.

Country

- C#/ASP.NET/Visual Basic is more common in the UK (27%) than in Germany (12%); 19% of the rest of the sample (not in the UK or Germany) uses C#.
- PHP is more common in Germany and the Netherlands (25%) vs. 17% in the rest of the sample. It was very rare in Ireland with just 4% (only 2 of 48 Irish respondents reported using PHP).
- Germany had more Bash users (41% vs. 29%) and fewer ASP/ASP.NET users (19% vs. 38%) than the rest of the sample.

Company size

- Smaller companies do more front-end web development: in companies with 2–25 employees, usage was high for CSS (47%), JavaScript (61%), HTML (61%), and PHP (26%). For the rest of the sample (mainly larger companies), the corresponding figures were 34%, 50%, 46% and 16%, respectively. Respondents from small (2–25) companies used Java less (29% vs. 40%).
- The largest companies (>10k employees) used C/C++ more, 26% vs. 15% for smaller companies.

Company age

- Young companies (2 to 5 years) had higher rates of Ruby (18% vs. 11%) and lower rates of ASP/ASP.NET and C# (4% and 9%, respectively, vs. 12% and 23%).
- Old companies (> 20 years) used C/C++ more (24% vs. 13%) with lower rates of JavaScript (45% vs. 56%).

Industry

Compared to the survey results as a whole, certain industries showed variations, including:

- Publishing / media uses C/C++ less (0% vs. 17%) and C# less (6% vs. 21%).
- Cloud Services / Hosting / CDN uses Ruby more (33% vs. 11%).
- Carriers / Telecom uses JavaScript less (33% vs. 53%).
- Manufacturing (non-IT) uses Java less (10% vs. 38%).
- Consulting (IT) uses Java more (51% vs. 36%).
- Computers / Hardware uses C/C++ more (50% vs. 16%).

Education

- Respondents with a master's degree tended to be less likely to use CSS (30% vs. 45%), JavaScript (47% vs. 58%), and HTML (42% vs. 56%). This may demonstrate the widespread assumption that front-end web development doesn't need as much formal education.
- Those with an academic speciality in mathematics, statistics, or physics tend to use Python more (41% vs. 23%) and PHP less (10% vs. 20%).
- Those with an academic speciality in computer science tend to use Java more (44% vs. 30%).
- Those with a PhD use Python more (41% vs. 24%).
- Respondents who are currently students (full or part time) are much more likely to use LISP (13% vs. 3%).

Experience

- Perl 5, Lua, and Visual Basic .NET have, on average, the most experienced respondents (average of 15–17 years of experience in their field).
- Ruby, CSS, and Clojure have, on average, the least experienced respondents (respondents using these languages averaged about 11 years of experience).

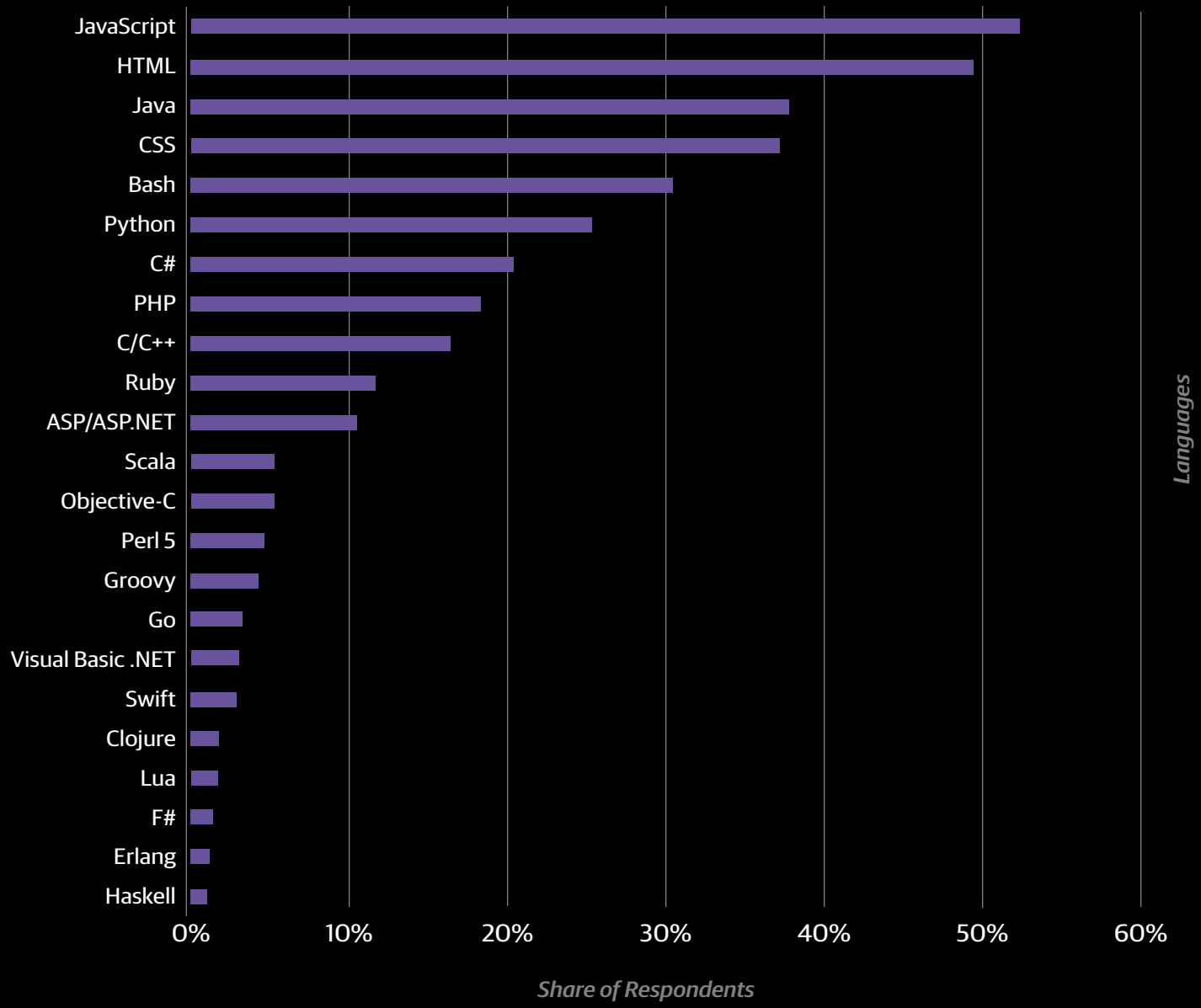
Team Size

- Visual Basic .NET, Clojure, and Ruby were relatively more popular in small teams (average of about 6 people on a team).
- Scala, Java, and Lua were more popular in larger teams (average of 8–9 people).

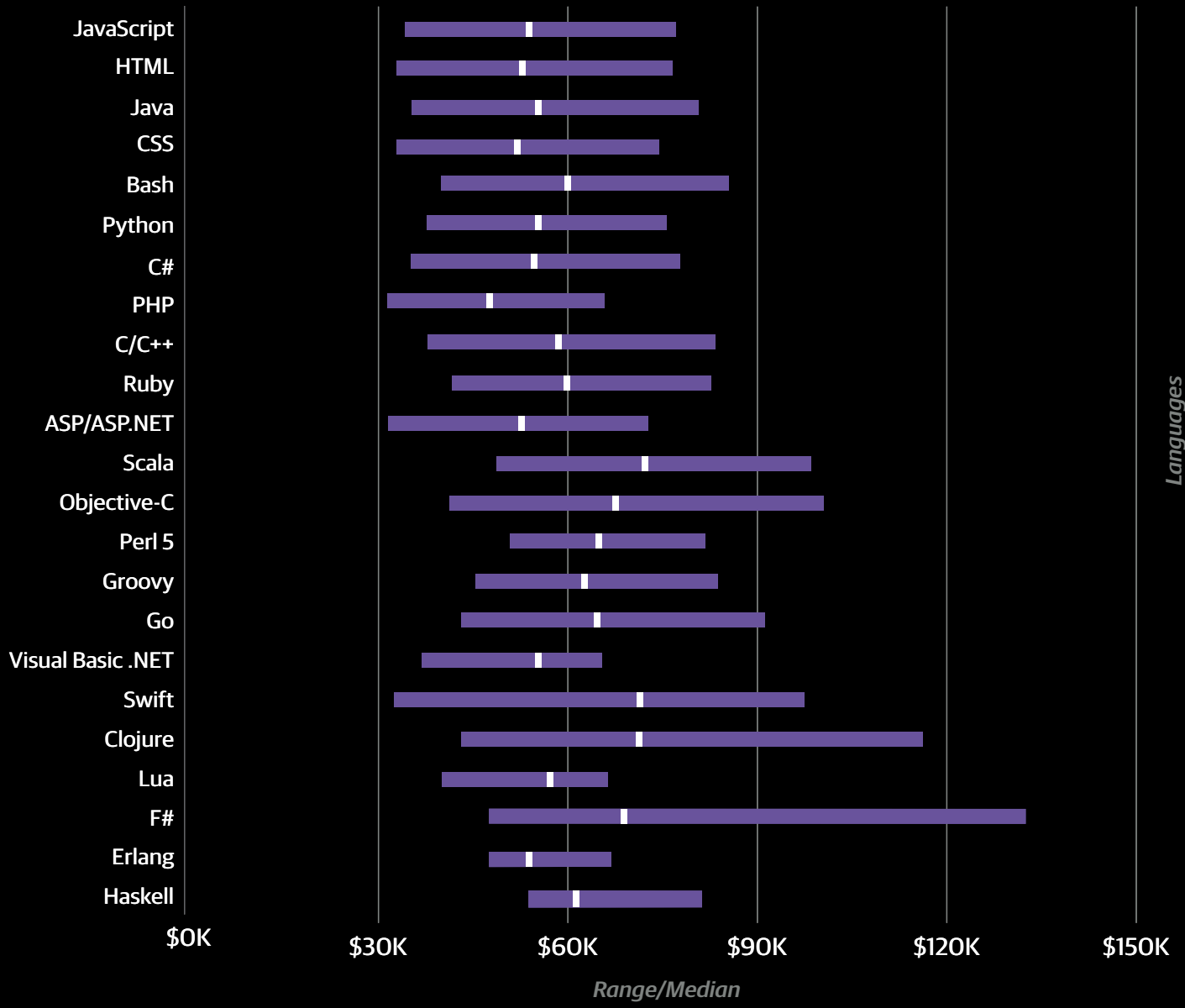
Past/Future Languages

Respondents were asked not only which languages they currently use, but also which ones they have previously used (but no longer do) and which ones they are planning to learn within the next 1–2 years. Most respondents had at least one "past" and "future" language: only 18% of the sample had never used a language professionally that they no longer use, and 17% did not plan on learning any new languages. *(continued)*

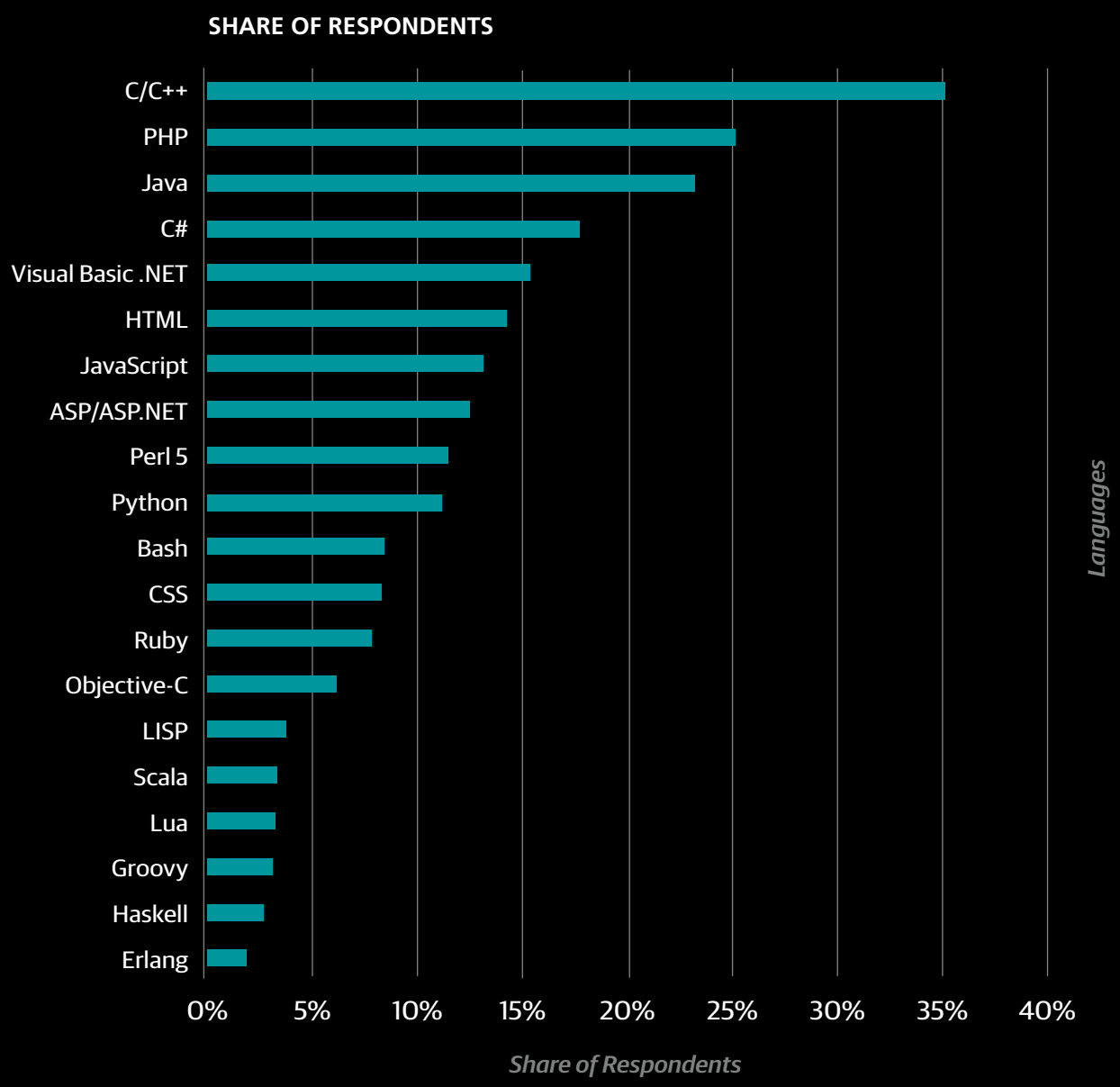
PROGRAMMING LANGUAGES

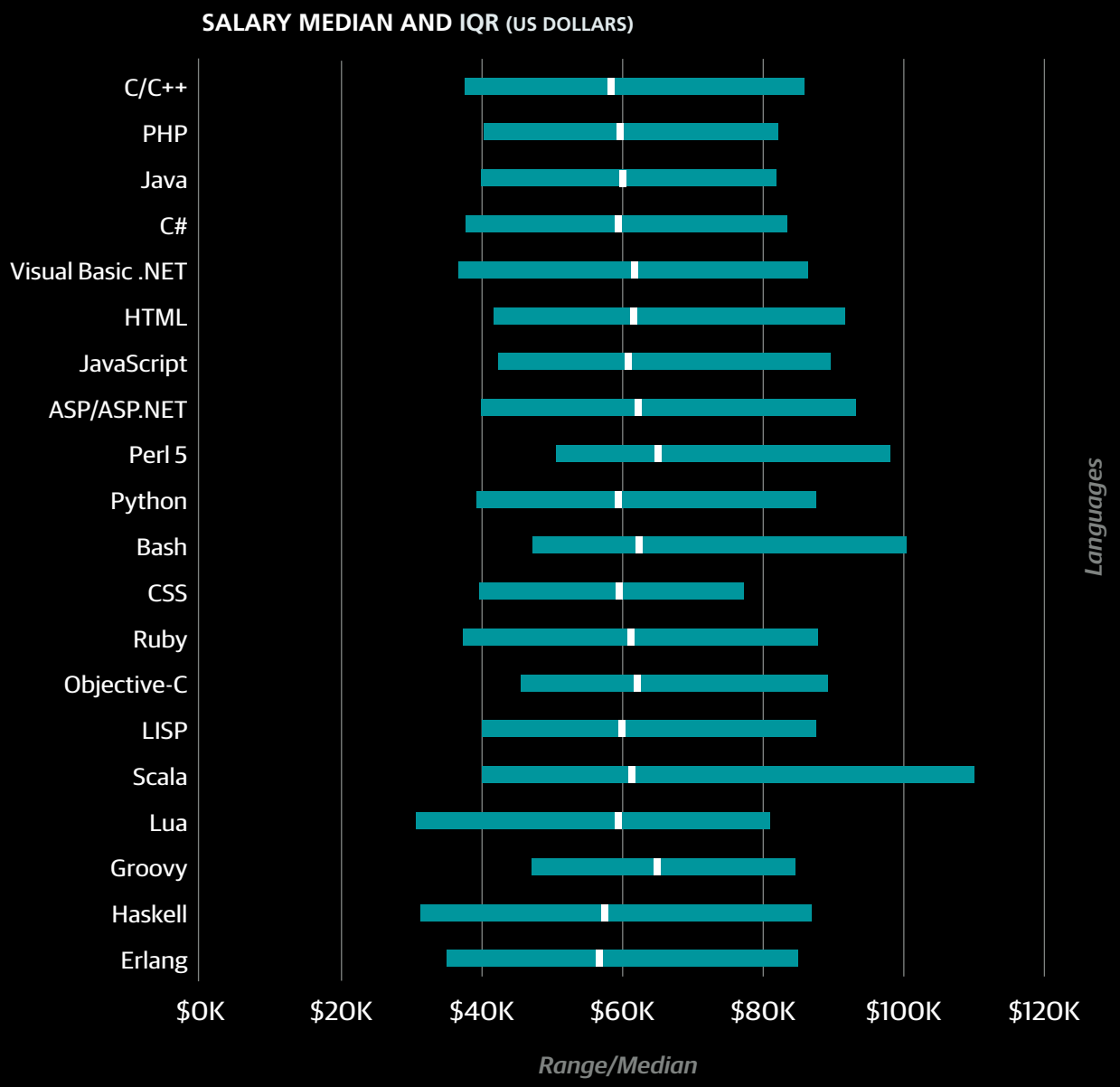


PROGRAMMING LANGUAGES

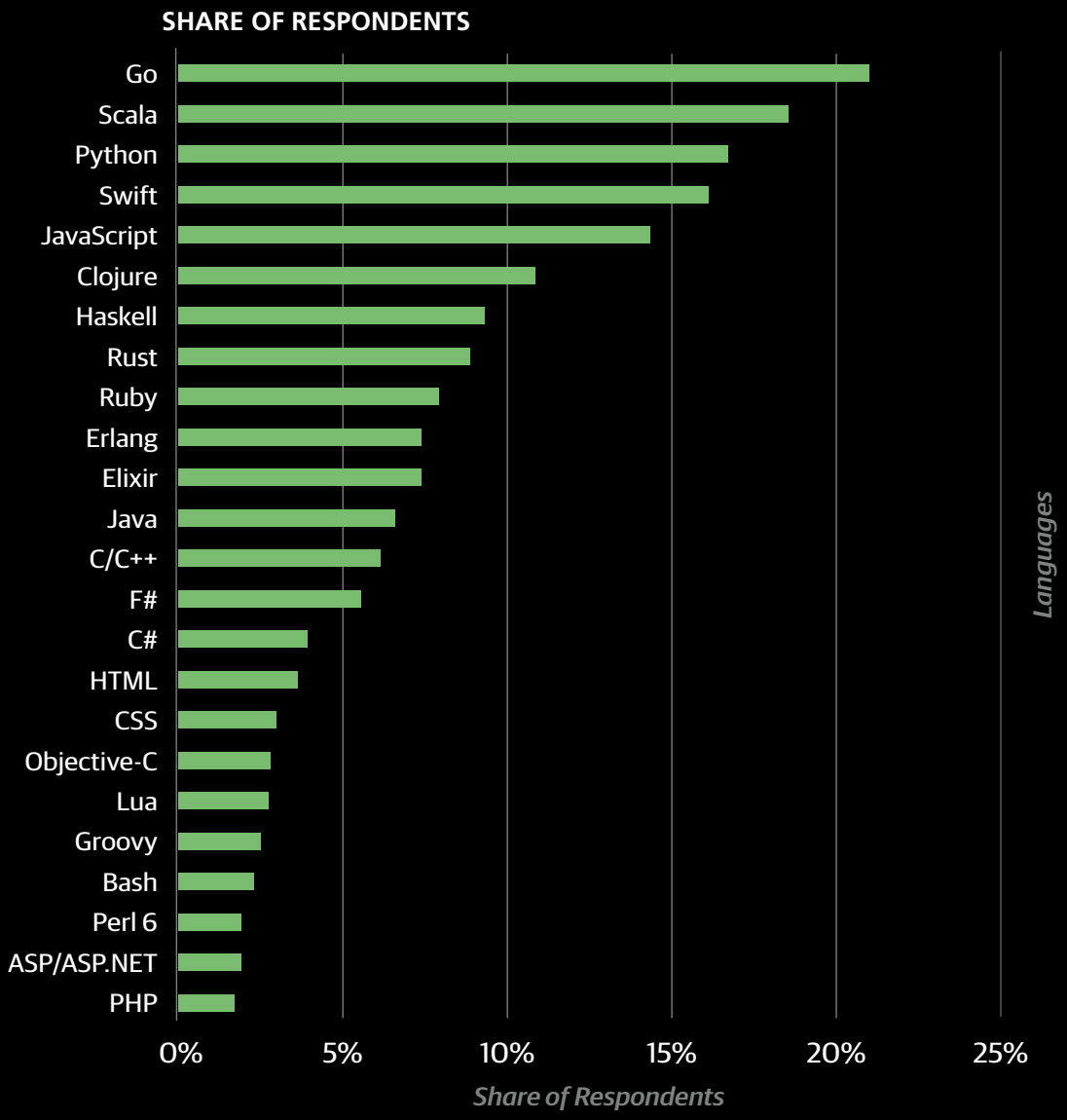


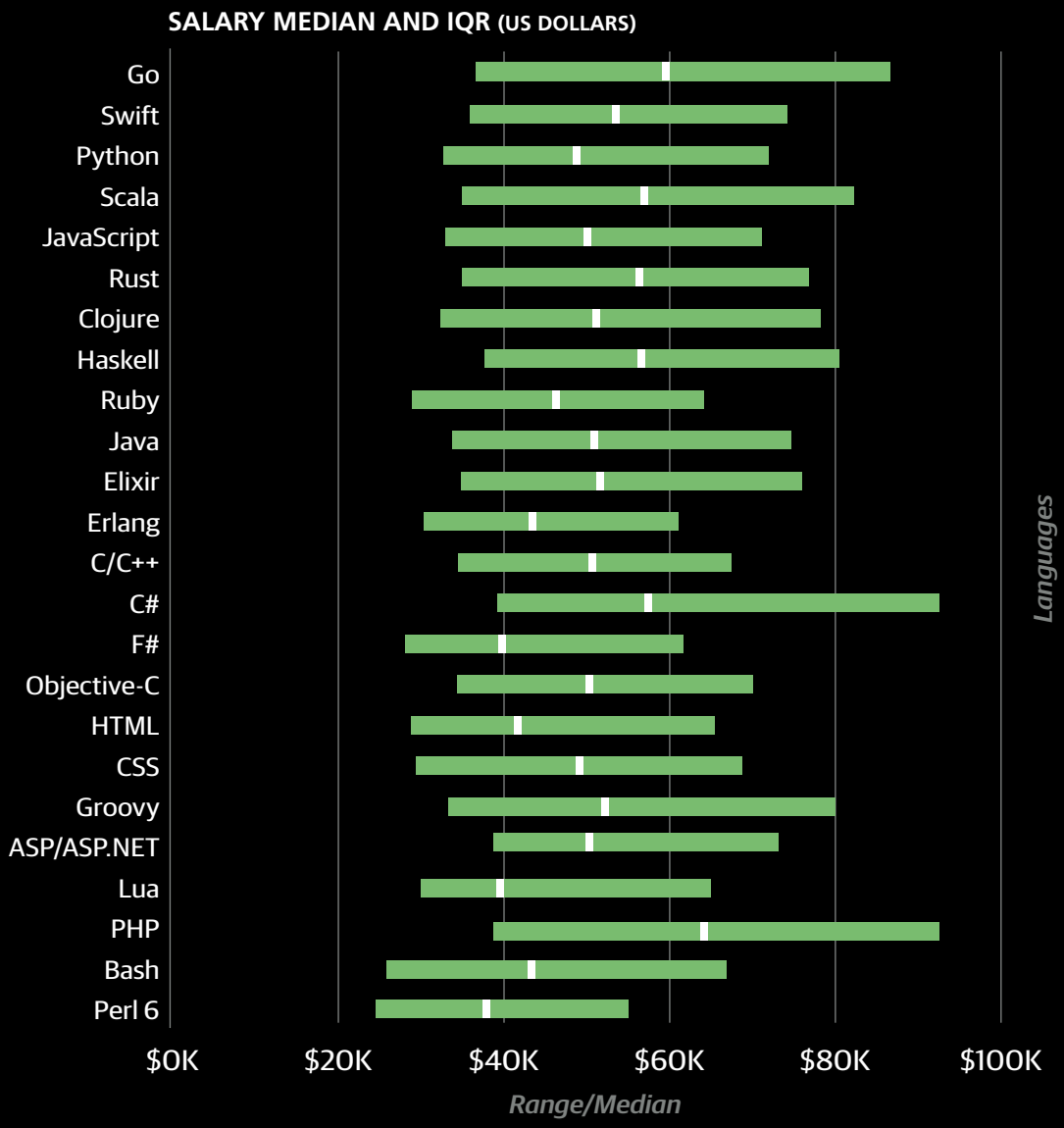
PAST PROGRAMMING LANGUAGES (PREVIOUSLY USED PROFESSIONALLY)




SALARY MEDIAN AND IQR (US DOLLARS)
C/C++
PHP
Java
C#
Visual Basic .NET
HTML
JavaScript
ASP/ASP.NET
Perl 5
Python
Bash
CSS
Ruby
Objective-C
LISP
Scala
Lua
Groovy
Haskell
Erlang
$0K
$20K
$40K
$60K
$80K
$100K
$120K
Range/Median
Languages

FUTURE PROGRAMMING LANGUAGES (INTENT TO LEARN)




SALARY MEDIAN AND IQR (US DOLLARS)
Go
Swift
Python
Scala
JavaScript
Rust
Clojure
Haskell
Ruby
Java
Elixir
Erlang
C/C++
C#
F#
Objective-C
HTML
CSS
Groovy
ASP/ASP.NET
Lua
PHP
Bash
Perl 6
$0K
$20K
$40K
$60K
$80K
$100K
Range/Median
Languages

Programming Languages (continued)

The most common "past" languages were C/C++ (35% of the sample), PHP (25%), Java (23%), C# (18%), Visual Basic .NET (15%), and HTML (14%). This does not necessarily mean these languages are dead or dying: Java and C#, for example, were reported more frequently as currently used languages than as past languages. Visual Basic, on the other hand, was reported more frequently as a past language than a currently used one: 15% versus 3%.

Four coefficients are associated with past languages: Bash (+$6,322), Clojure (+$12,549), Objective-C (–$3,869) and Perl 6 (+$14,743). Interpreting these bonuses is not straightforward. A positive coefficient can be spun negatively since the respondents got a big boost in their expected salary when they stopped using the language, but it could also be taken in a positive light: these languages may have served as important career steps or learning paths.

Most respondents (54%) selected one or two languages that they would like to learn (most of the rest selected three or more). The top choices were Go (21% of respondents), Scala (19%), Python (17%), Swift (16%), JavaScript (14%), and Clojure (11%). Preferences in Europe were similar to the rest of the world.

There is a clear distinction on the to-learn list defined by the ratio between those that currently use the language and those that want to learn the language. In the case of Go, Swift, Scala, Rust, Clojure, Haskell, Elixir, Erlang, and F#, there are far more people who want or plan to learn them than currently use them. In contrast, Python, JavaScript, Ruby, and Java are used by more respondents currently than they were chosen as learning goals.

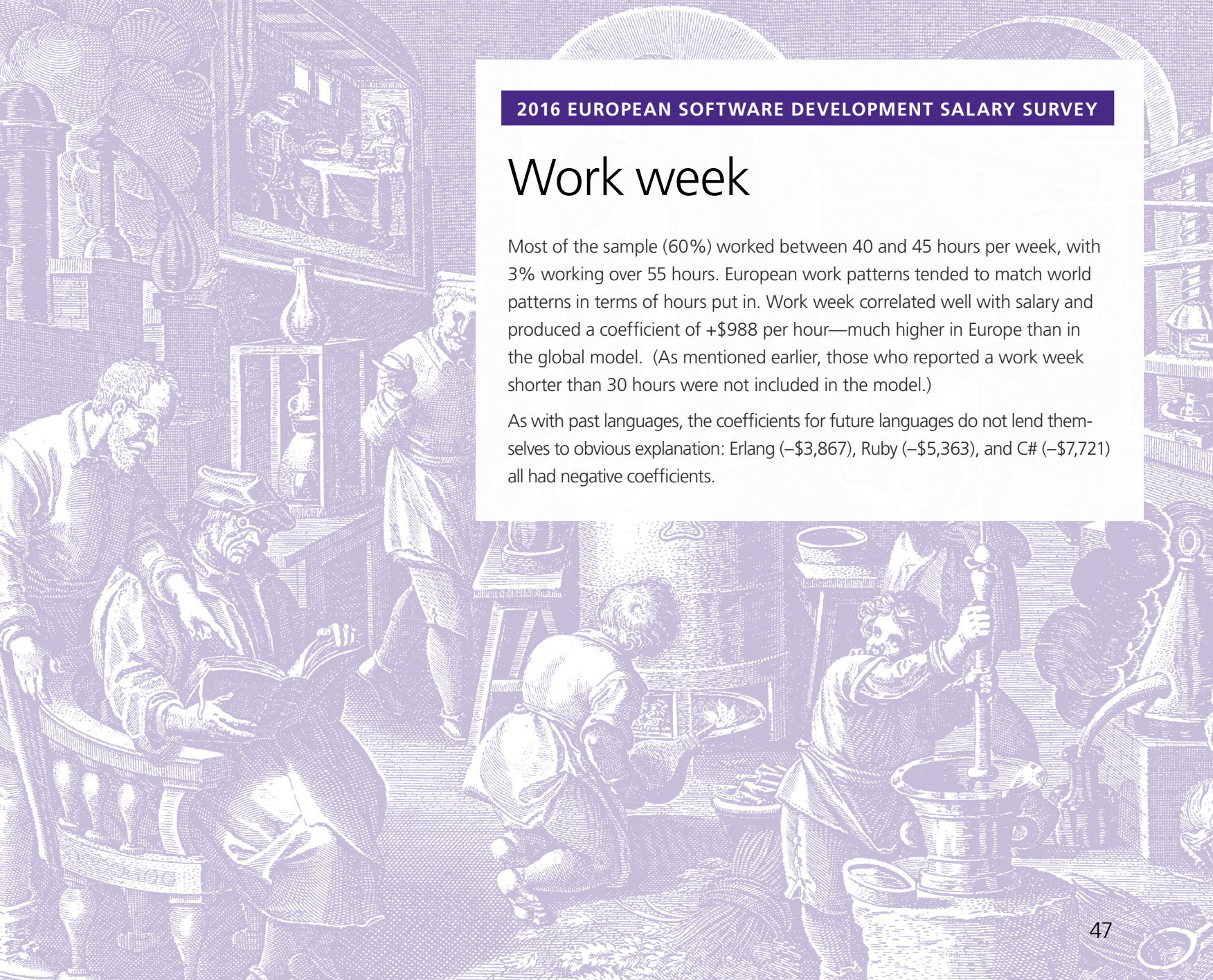
The common paths taken by the most well paid respondents was C# > Bash > Go (median salary $60K).

One final analysis we can make of the past-present-future languages is to associate them in language paths. For example, respondents who previously used C/C++ tend to use Java now, and those who use Java now tend to want to learn Scala. The most common paths are shown in the following graph (**or name of graph, etc.**). While these pathways were not included as additional features to the model, we can look at the median salaries of respondents who took a particular path. The common paths taken by the most well paid respondents was C# > Bash > Go (median salary $60k), while respondents who took the C/C++ > HTML > Python path had a median salary of only $31k.

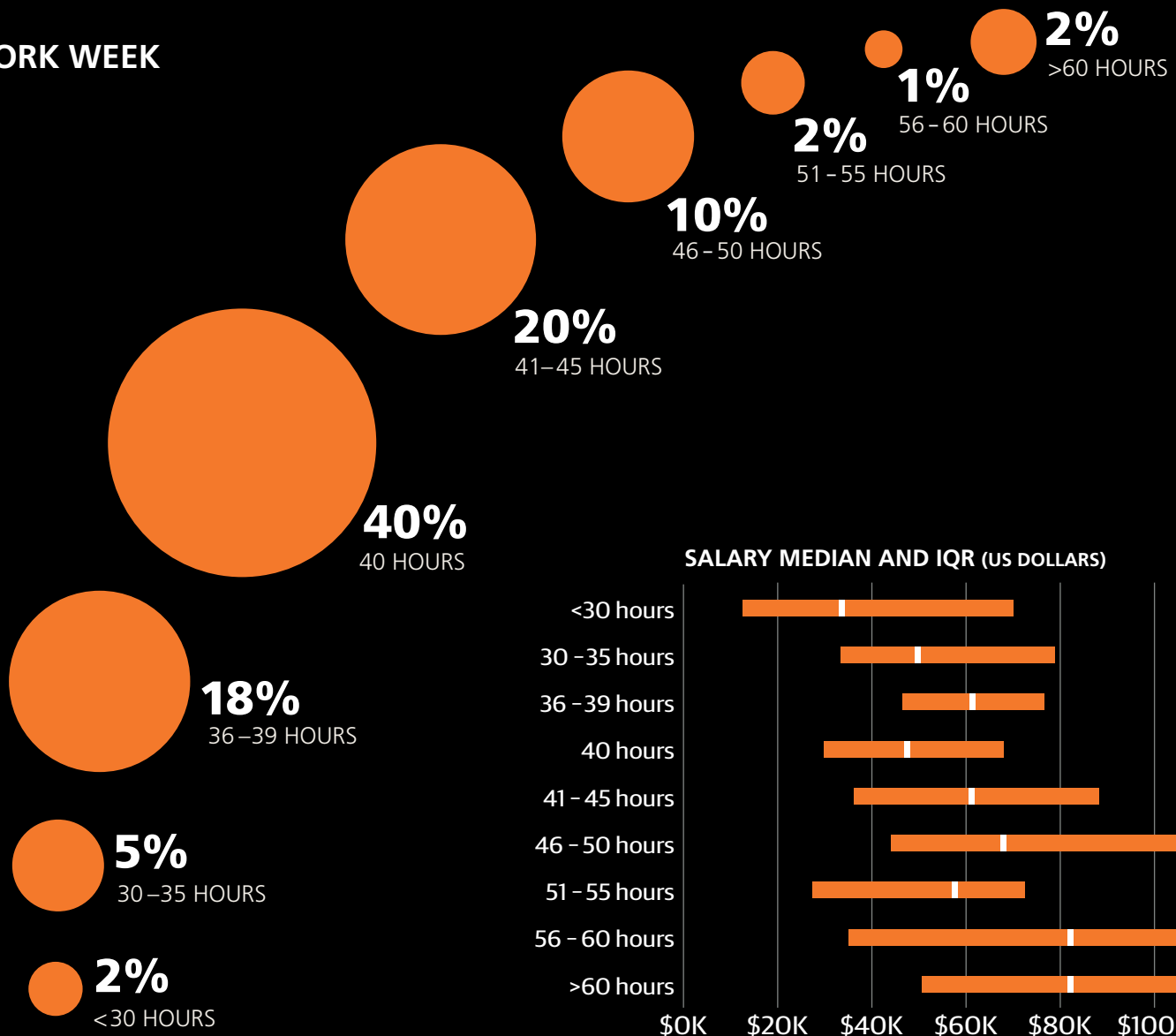
Work week

Most of the sample (60%) worked between 40 and 45 hours per week, with 3% working over 55 hours. European work patterns tended to match world patterns in terms of hours put in. Work week correlated well with salary and produced a coefficient of +$988 per hour—much higher in Europe than in the global model. (As mentioned earlier, those who reported a work week shorter than 30 hours were not included in the model.)

As with past languages, the coefficients for future languages do not lend themselves to obvious explanation: Erlang (–$3,867), Ruby (–$5,363), and C# (–$7,721) all had negative coefficients.

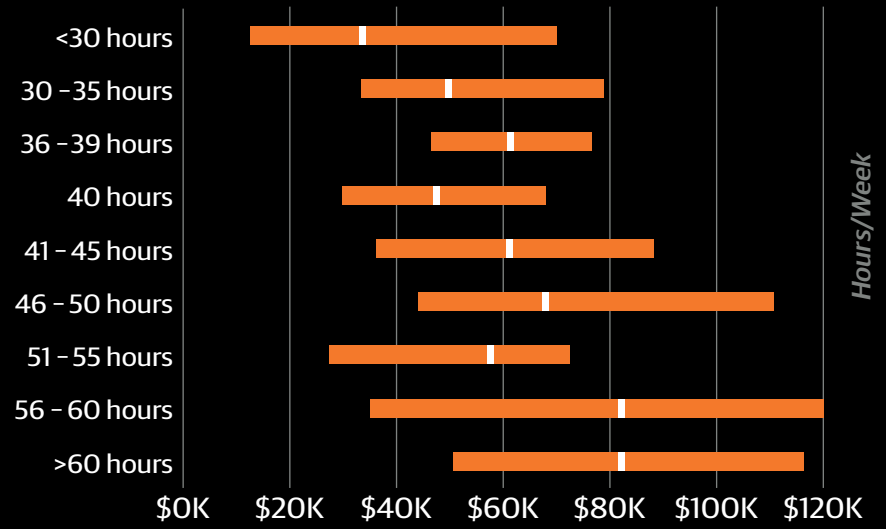


WORK WEEK



SHARE OF RESPONDENTS

SALARY MEDIAN AND IQR (US DOLLARS)



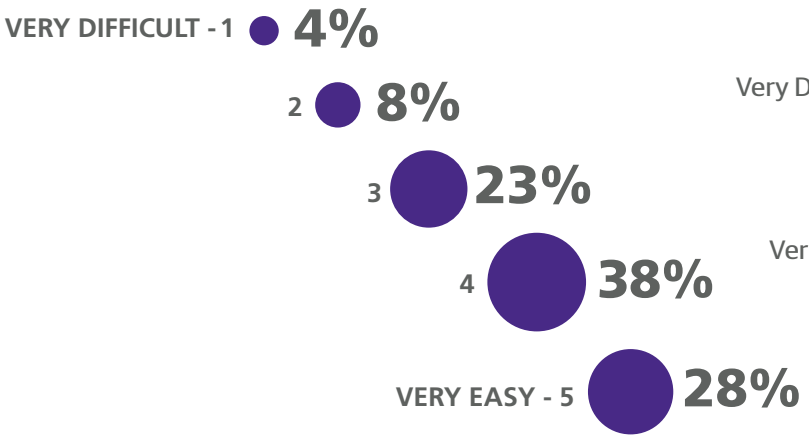
Bargaining and Ease of Finding Work

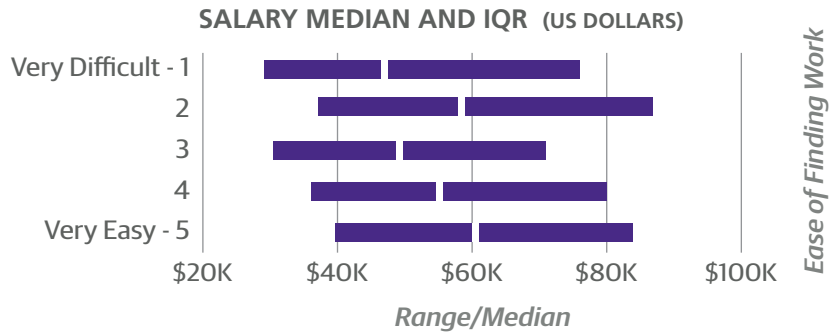
RESPONDENTS WERE ASKED TO RATE HOW WELL THEY CAN BARGAIN on a scale of 1 to 5. Most respondents gave themselves a 3 or 4. Although this is a highly subjective rating, we asked the question because we find it necessary to have some proxy for bargaining skills, since this can have a huge effect on your eventual compensation. Bargaining points correlated highly with salary, and the model predicts a salary boost of +$4,290 for each point on the scale.

A related question, similarly opinion-based, was the ease of finding new work. In some cases this may be obvious (for example, if the respondent has job offers on the table, or had recently been looking for work unsuccessfully), but in most cases it is probably just as rough a measure as bargaining skills. The average score, also on a five-point scale, was 3.8. Since this is not a variable that we can change (that is, in the same way that we can move to new states or countries, learn new tools, or shift careers to new roles), this variable was not included in the model.

EASE OF FINDING A NEW ROLE *ON A SCALE FROM 1-5* (1 being very difficult, 5 being very easy)

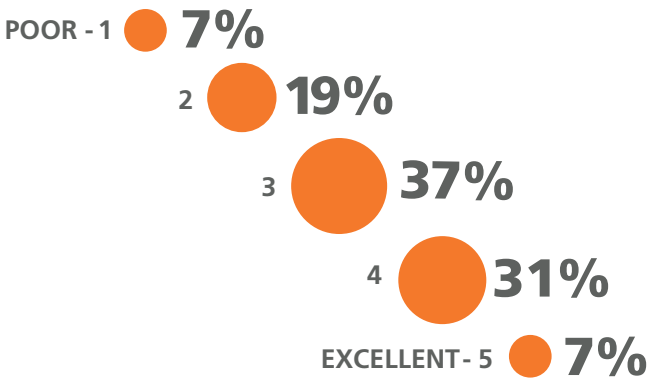
SHARE OF RESPONDENTS

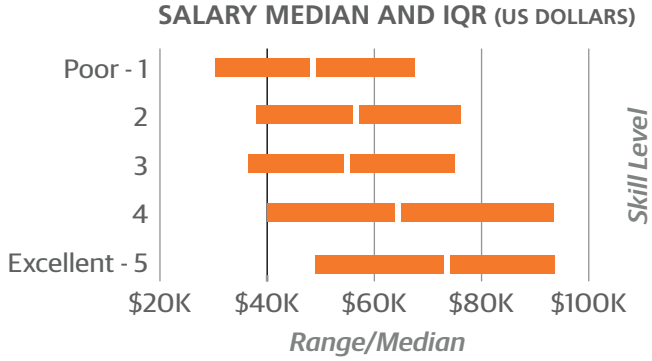




SELF-ASSESSED BARGAINING SKILLS *ON A SCALE FROM 1-5* (1 being poor, 5 being excellent)

SHARE OF RESPONDENTS





The Model in Full

THIS SECTION PRESENTS EVERYTHING WE FOUND IN OUR STUDY that has a predictive effect on salary—with the warning, as stated earlier, that the variables may not control or cause salary changes. The model has an R-squared of 0.475, which means that the model explains approximately 48% of the variation in the sample salaries.

To use the model, proceed through the coefficients and add or subtract the ones associated with a feature that applies to you. Remember that some coefficients are multiplied by a factor: number of hours in your work week, years of experience, numbers of tools in a cluster. Once you sum up the coefficients, you will obtain an estimate for your annual total salary in US dollars.

Work week, per hour: +$988

Experience, per year: +$1,257

Bargaining skills, per point (scale of 1 to 5): +$4,290

Switzerland: +$19,161

United Kingdom, Ireland, Norway, Denmark: –$5,513

Germany, the Netherlands: –$12,495

France, Sweden, Belgium, Finland, Austria: –$22,283

Spain, Italy, Greece, Portugal, Turkey: –$35,911

Poland, Romania, Czech Republic, Ukraine, Hungary, Slovenia, Slovakia, Estonia, Bosnia and Herzegovina: –$42,594

Russia: –$45,224

Industry = Banking/Finance: +$16,260

Industry = Consulting (IT): +$8,419

Industry = Education: –$6,438

Company size = 1: +$8,832

Company size = 10,000 or more: +$5,156

Team size, per team member: +$184

Works with (other) programmers: +$5,332

PhD: +$7,906

Title = Architect: +$10,990

Title = Principal/Lead: +$6,254

No involvement in back-end web development: +$3,606

Major involvement in mobile development: –$3,593

Minor involvement in hardware development: –$4,595

Major involvement in teaching/training others: +$3,499

No communication with people outside of the company: +$4,115

Hours spent in meetings, per hour/week: +$151

Cluster 4 (Unix tools): +$2,636

Cluster 5 (Cloud): +$698

Cluster 6 (Python): –$1,234

Cluster 7 (.NET): –$1,114

Cluster 8 (LAMP): –$2,206

Cluster 10 (Editing 1): –$3,485

Cluster 11 (Distributed computing): +$173

Cluster 12 (Editing 2): +$3,149

Cluster 13 (Scala): +$1,112

Cluster 14 (Apple): +$2,262

Cluster 15 (Clojure): +$945

Past language, Bash: +$6,422

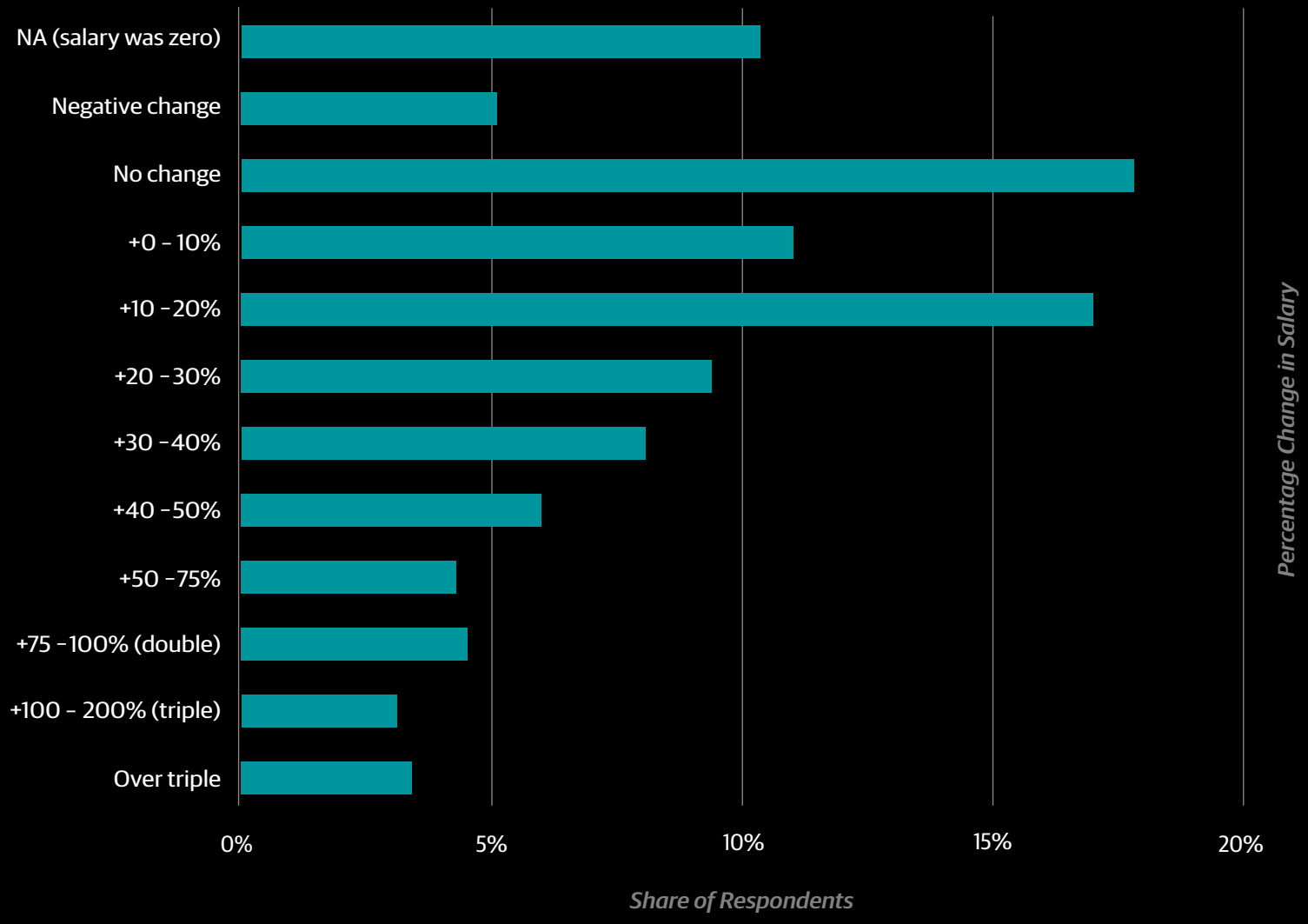
Past language, Clojure: +$12,549

Future language, Objective-C: –$3,869

Future language, Perl 6: –$13,743

PERCENTAGE CHANGE IN SALARY OVER LAST THREE YEARS

SHARE OF RESPONDENTS



Conclusion

IN ANY INDUSTRY, IT IS WISE TO KEEP IN TOUCH WITH RELEVANT TECHNOLOGICAL CHANGES that could affect your career. In a field such as software development where the tools change so rapidly, this becomes an increasingly challenging task. This report is intended to give a quantitative look at the various careers and profiles of software professionals, and shed some light on what your next step might be, whether it is to learn a new language or to shift roles.

Surveys certainly have drawbacks, especially when the sample is self-selected. In our conclusions here, we rely on the assumption that the people who took the survey are reasonably representative of the entire software development world, or at least some important subset of it. The O'Reilly programming audience—from which the respondents generally come—has a proclivity toward choosing open source and emerging technology, and this will affect the results: the percentages of respondents who used certain tools, for example, are probably not good estimates of the global development rates.

The correlation-causation distinction should certainly be kept in mind when reading this report, but it is worth noting that, while correlation does not imply causation, correlation does not deny it, either. If you were planning on learning a new tool or language anyway, it's not a bad idea to choose one that correlates positively with salary, if it suits your professional needs. Generally speaking, a broader skillset is respected in the software world and learning more tools always opens new doors. Taking into account the information in this report might increase the chances of opening a door with a bigger paycheck somewhere on the other side.

We need *your* data.

To stay up to date on this research, your participation is critical. The survey is now open for the 2017 report, and if you can spare just 10 minutes of your time, we encourage you to take the survey.

oreilly.com/programming/2017-programming-salary-survey.html

Take the Survey

O'REILLY®

Wait. There's more.

4 easy ways to stay ahead of the game.

Programming technologies don't stand still—neither should you. Sharpen your skills and advance your career potential with these resources, most of which are free.

1. **Sign up for the O'Reilly Programming Newsletter** (*oreilly.com/programming/newsletter*) to get fresh news each week, including ideas, insights, interviews, and advice from industry leaders, and even a couple of laughs. Receive advance notice of O'Reilly programming books, reports, and events, plus exclusive offers and discounts for subscribers.

2. **Bookmark** *oreilly.com/topics/software-engineering*, and make it part of your essential reading. You'll find timely, in-depth interviews and podcasts with industry leaders, excerpts from forthcoming books, and special reports on software-related issues that will keep you on top of your game.

3. **Participate in free webcasts at webcasts.oreilly.com.** Learn programming skills and tools online from some of the top minds practicing today in a casual, interactive forum.

4. **Immerse yourself in learning** at an upcoming O'Reilly conference. Check out *conferences.oreilly.com*.

O'Reilly Media, Inc.
O'Reilly is a registered trademark of O'Reilly Media, Inc.