Programming Language Compensation Gap

DA2/C2 - Term Project

Son Nam Nguyen

20 December 2021

Introduction

In this project, I will present my findings on the programming language compensation disparity among full-time developers in the United States. My aim is to show the expected differences in compensation depending on one knowing Python, SQL, Java, etc. controlled with employment, experience, education and demographics confounders.

Source Description

For the analysis, I have used the Stackoverflow Developer Survey conducted in 2021. This report is based on a survey of **83,439 software developers from 181 countries around the world**. Respondents were recruited primarily through channels owned by Stack Overflow. The top sources of respondents were onsite messaging, blog posts, email lists, banner ads, and social media posts.

Data Quality

I consider the sample to be representative for the community. First, because it reaches out to potential respondent though an extensive number of channels. Yes, these mediums all assume that developers all have access to network connection, have their email, and so on, but it extremely rare that people in the IT sector don't have those.

Looking back at earlier polls, the organization has made a number of changes throughout the years to reduce measurement errors in the variables. Instead of allowing respondents to offer arbitrary answers to questions like age, they grouped the variable into bins to eliminate age inconsistencies, at the cost of limiting the resolution with which age coefficients could be estimated.

Another possible window for error could be recognized in the reported level of compensation. People **who believe that they are underpaid are less inclined to reveir their compensation, while those who are paid higher than average are eager to do so**. Also, I need to identify reported compensations which are not plausible given the employment, in other words, drop extreme values which are beyond a certain reliability treshhold.

Like many others, the dataset lacks a variable which lets me determine the coding years spent with the programming languages. Therefore, it is unknown what share of their professional career was spent on working with the listed languages someone has worked with so far. Simply put, I can't estimate the coefficients for what should we expect in terms of renumeration when working an additional year with a language.

Finally, **job titles/fields have always been a challenge to comprehend**. Titles frequently do not reflect what a professional does on a daily basis. Many job titles are not sufficiently differentiated from one another, and as a result, they tend to blur together (data people often call themselves developers). To put it another way, the sets of job tasks are rarely mutually exclusive to each other.

Variables

Let me shortly clarify variables which are not that trivial what they present. Each variable represent an entry in the survey which you can find here.

Salaries are converted from user currencies to USD using the exchange rate on 2021-06-16, and also converted to annual salaries assuming 12 working months and 50 working weeks. As a best practice, I will use the log of converted yearly compensations in thousand USD as my LHS variable.

Control variables which I plan to focus on are **(1) Field of employment, (2) Size of organization, (3) Level of education (4) Gender, (5) Age, (6) Ethnicity (7) Number of known programming languages, (8) Years of coding**. I have also restricted the number of languages under scope to the top five languages according to the survey which are **Javascript, HTML/CSS, Python, Java, SQL**.

Data Munging

To clean the raw data, I performed the following transformations:

1. Dropped of columns not needed in the further analysis
2. Regroup **Organization Size**, **DevType**, **Ethnicity** and **Gender** variables to decrease the number of factor levels.
3. Cast the variables to their appropriate class structure.
4. Flip empty strings, "Prefer not to say" and its synonyms to NAs
5. Filter sample to observations working full-time in the United States

Calculate new dimensions for:

1. Binary flags for working in coding languages (JavaScript, HTML/CSS, Python, Java, SQL)
2. Count of coding languages an individual has worked with
3. Compensation represented in thousand USDs and its log form

Descriptive Statistics

The below descriptive table shows the key statistics of the numeric variables under the discussion. The distribution of **Years of coding is right-skewed with a mean of roughly 17 years**, while the dispersion is relatively high (measured at SD = 10.82). Starters have 1 years of coding, while veterans with the most experience reported 50 years of coding so far.

Regarding yearly compensation, US developers follow an aggressively lognormal distribution in salaries, where the **mean is 265.4 thousand USD**, while the median is 125 thousand USD. 95% of the sample are paif less than 450 thousand USD, and **5% higher than that maxing out at a whooping 22 billion USD a year**. A minimum of 0 thousand USD wage means that there are observations having a single digit yearly compensation. According to Minimum-wage.org, **the federal minimum yearly minimum wage in 2021 was $15,080.00/year**. Using that criterion, I've dropped rows with a yearly compensation below the minimum wage.

At last, **a developer working full-time in the US have experience in working with 4 programming languages in average**. The distribution of the variable is about bell-shaped, with a standard-deviation of 2.9. According to the percentiles, 95% of the sample understands 11 or fewer programming languages, indicating that most two-digit numbers are uncommon in the developer community. **Some of the most experienced developers can code even in total of 38 languages**.

Table 1: Descriptive statistics

	Mean	Median	SD	Min	Max	P05	P95
Years of coding	17.11	15.00	10.82	1.00	50.00	4.00	40.00
Yearly Compensation (K$)	265.36	125.00	843.23	0.00	21 822.25	56.00	450.00
Number of Languages Known	5.52	5.00	2.90	1.00	38.00	2.00	11.00

After clearing, filtering and dropping implausible observations in the sample, I have explored once again the distribution of the years in coding, number of languages a developer have worked with, and the distribution of programming language usage in the community.

The graph in the ***Appendix*** shows that the histogram of years of coding is still relatively right skewed, with a lower mean of 16.78 years.

Then on, I have illustrated the distribution of number of coding languages known among developers in each employment field (see ***Appendix***). **Developers, Managers and SysAdmins are having the highest average language known (with the larger IQRs), wheres people working in data has the lowest number of average languages they have worked with (low IQR)**. It is not surprising at all, considering that oftentimes managers are ex-developers, developers are more likely to work on both the front-end and back-end, while the toolkit of data people are usually reduced to the set of Python, R and SQL. Additionally, the kernel density graph of all within field distributions can be characterized by a **left-peaked shape**.

People still use Java and JavaSript the most (more than 20% of the sample), as these languages have been in the discussion fo decades and people have not yet fully transitioned to the now-booming ones like Python (sitting at roughly 16%). **SQL is also up there close to the leaders of the chart**, and personally, **I expect the SQL to resurface more among developers** because it is in high demand, especially in Data Engineering. The reason for that, new product following the Modern Data Stack mindset have to be SQL agnostic and thrive for supplying the community with a pipeline which can be managed by purely SQL (see ***Appendix***).

Before jumping into the set of models I have estimated, behind the scenes, I checked how numerical variables are correlated, and whether we have to address the possibility of multicollinearity (see ***Appendix***). Fortunately, **none of the pairs in my matrix had a correlation coefficient higher than 0.22**, meaning that we can reject the hypothesis that these are linear combinations of each other.

Model

To present my estimated models, I chose to use an additive logic approach, in which I first estimated the unconditonal link between compensation and (the five) programming languages, then afterwards **added controlling factors relating to employment and demographics**. Variables are chosen on the basis that they are the key compensation mechanism variables in most studies. I could have included experience with a wide range of tools that were indicated in the survey and might be utilized to gain an advantage on the job market, but I refused to do so because it would have thrown off my coefficients/observations balance. The argument about the number of variables vs. the number of observations also relates to my decision not to estimate polynomial and interaction forms.

Unconditional Estimates

The baseline model indicates that all of the languages except Python and JavaScript has a statistically significant impact on compensation levels, at 5%. **Those who know Java tend to have 8% higher yearly compensation in the US. Studying front-end skills like JavaScript and HTML/CSS are on average associated with 6 and 11% lower salaries than those who are not doing it, respectively, holding other covariates constant**. People knowing SQL also tend to earn less by 6% on average compared to those who are not familiar with it. The goodness-of-fit is relatively low with only 1% of the variance in y is explained by the variance of explanatory variables.

Second-best model

Adding employment-specific variables like company size, and working department, we can observe that most of the explanatory variables have lost their significance on the 5% level, except HTML/CSS. The magnitude of coefficients all dropped or stayed the same for the regressors, however, the R^2 has also improved suggesting that included managed to cover a part of the variance in yearly compensation.

First-best model

Estimating my fully extended model where I included demographic control variables, the statistical significance of the explanatory variables changes once again. **Python now is significant and implies a 5% higher yearly compensation for those who have worked with it, all else being equal**. The coefficients of SQL, JavaScript and HTML/CSS are still associated with a lower compensation. **By working with SQL, compensation is expected to be lower on average by 9%, while lower by 11% in the case of HTML/CSS, assuming that other covariates are constant**. The R^2 of roughly 0.91 suggests that there are still a lot of unexplained variances in y, hence, the continuation of this project should incorporate other unknown variables.

Check full regression table in the **Appendix**

Table 2: Coding Language Compensation Gap in the US

	Baseline	Employment	Demographics
(Intercept)	4.99 (0.02)**	5.13 (0.03)**	4.69 (0.04)**
Python	0.03 (0.02)	0.03 (0.02)	0.05 (0.02)**
SQL	−0.06 (0.02)**	−0.03 (0.02)	−0.04 (0.02)*
Java	0.08 (0.03)*	0.03 (0.03)	0.04 (0.03)
JavaScript	−0.06 (0.03)	−0.01 (0.03)	−0.01 (0.03)
HTML_CSS	−0.11 (0.02)**	−0.11 (0.02)**	−0.08 (0.02)**
Num.Obs.	9205	7186	6976
R2	0.009	0.041	0.091
Std.Errors	Heteroskedasticity-robust	Heteroskedasticity-robust	Heteroskedasticity-robust

Baseline: Estimates unconditionally

Employment: Adds (1) Working Field, (2) Company Size

Demographics: Adds (1) Age, (2) Gender, (3) Level of Education, (4) Years Coding, (5) White dummy

External Validity

To check the external validity of my best model which incorporates variation in employment and demographics. I have created additional subset of dataframes for the other four countries having the most observation in the data (UK and Ireland, France, India, and Germany)

Coefficents do not appear to be steady at all, which could be attributable to regional variances in skill demand. For example, **knowing Python and Java is substantially more favorable in India than in Germany (the former is related with 21% greater remuneration) as compared to the coefficients in the United States**. While Python is higher valued in India, SQL knowledge is considered a no-go, with an average pay cut of 11%. Other notable discrepancies can be found in the projected coeffficients of **France, where workers with Pyhton and JavaScript experience are penalized by 9 percent and 26 percent lower compensation, respectively, whereas Java developers are rewarded with a 22 percent higher salary**.

Table 3: External Validity Coefficients

	Constant	Python	SQL	Java	JavaScript	HTML/CSS
UK and Ireland	3.81	0.05	-0.06	0.07	-0.05	-0.05
India	3.02	0.16	-0.13	0.06	0.04	-0.16
Germany	3.60	0.03	0.02	-0.06	0.12	-0.08
France	3.43	-0.10	-0.05	0.21	-0.25	-0.04

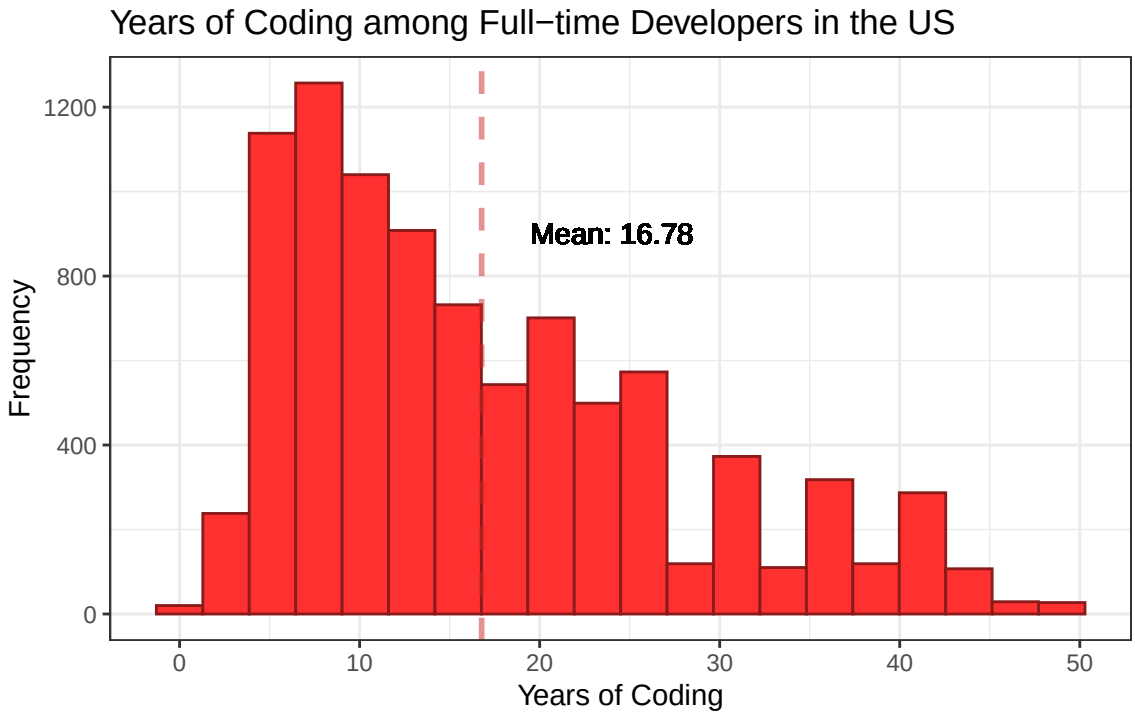
Conclusion

To sum up, There isn't a good language to concentrate on if we want greater pay. It undoubtedly depends on the country's job market, and preferences for specific languages shift regularly. Python has a lot of potential, but it hasn't been widely adopted compared to languages that are on the slide in terms of popularity. It was also surprising to find SQL in the top ten because it is a very basic language but that's what makes it easier for a data team to migrate on new technologies.

I've learned a lot about the present work market, and I believe I've addressed many of the major stumbling blocks in order to achieve my goal. In terms of constraints, I aggregated levels of factor variables to simplify my model, but I paid the price by being unable to see those coefficients at a finer level. There is still a lot of unexplained variation in compensation, which can be linked to the company's performance, seniority, the effort he or she puts in at work, and other types of compensation (e.g. shares, cafeteria, etc.). Beyond that, there are factors such as family background and social skills that can only help to reduce the variance. I'm looking forward to delving deeper into this link and using other sources to supplement or proxy the latent variables provided.

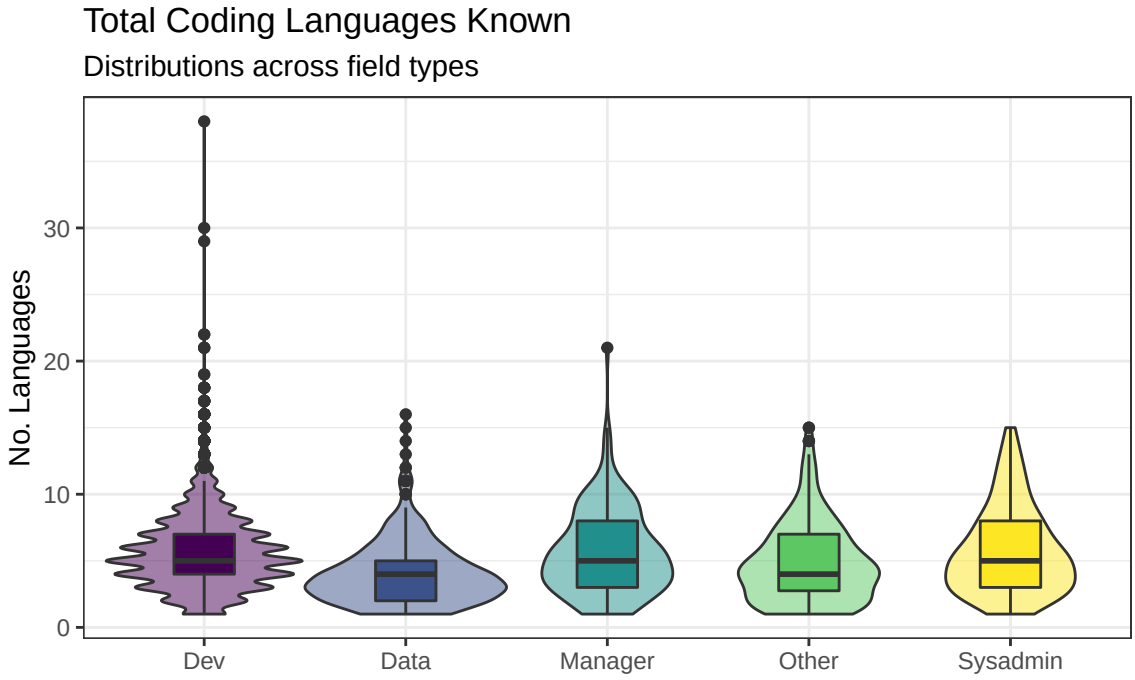
Appendix

Distribution in years of coding



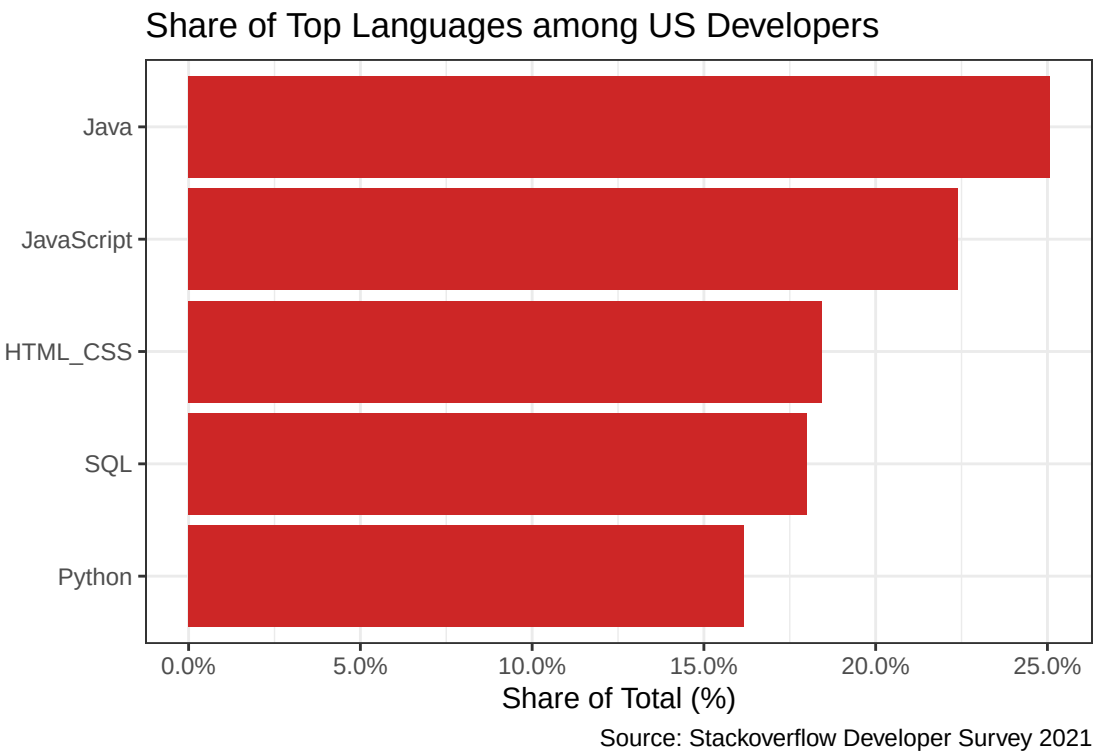
Source: Stackoverflow Developer Survey 2021

Number of languages known



Source: Stackoverflow Developer Survey 2021

Most popular languages distribution



Correlation matrix

Table 4: Correlation Matrix

	YearsCode	ConvertedCompYearlyK	LanguageCount
YearsCode	1.000	0.010	0.022
ConvertedCompYearlyK	0.010	1.000	0.022
LanguageCount	0.022	0.022	1.000

Full regression table

Table 5: Full table: Coding Language Compensation Gap in the US

	Baseline	Employment	Demographics
(Intercept)	4.99 (0.02)**	5.13 (0.03)**	4.69 (0.04)**
Python	0.03 (0.02)	0.03 (0.02)	0.05 (0.02)**
SQL	−0.06 (0.02)**	−0.03 (0.02)	−0.04 (0.02)*
Java	0.08 (0.03)*	0.03 (0.03)	0.04 (0.03)
JavaScript	−0.06 (0.03)	−0.01 (0.03)	−0.01 (0.03)
HTML_CSS	−0.11 (0.02)**	−0.11 (0.02)**	−0.08 (0.02)**
Data		−0.15 (0.04)**	−0.11 (0.04)**
Manager		0.42 (0.07)**	0.34 (0.07)**
Other Field		−0.11 (0.06)	−0.12 (0.06)*
SysAdmin		0.00 (0.11)	−0.07 (0.10)
Medium Company		−0.14 (0.02)**	−0.13 (0.02)**
Small Company		−0.39 (0.03)**	−0.36 (0.03)**
Associate degree			−0.19 (0.05)**
Master's degree			0.03 (0.02)
Doctoral degree			−0.03 (0.04)
Elementary School			0.08 (0.17)
Professional degree			0.02 (0.13)
Secondary School			−0.20 (0.05)**
University w/out degree			−0.07 (0.03)*
Other Education Level			−0.37 (0.14)*
Non-binary			−0.10 (0.06)
Women			−0.10 (0.04)**
25-34			0.21 (0.04)**
35-44			0.27 (0.04)**
45-54			0.09 (0.05)
55-64			−0.06 (0.06)
65+			−0.21 (0.09)*
18-			−0.55 (0.20)**
People of color			0.05 (0.02)*
Years Coded			0.01 (0.00)**
Num.Obs.	9205	7186	6976
R2	0.009	0.041	0.091
Std.Errors	Heteroskedasticity-robust	Heteroskedasticity-robust	Heteroskedasticity-robust