A Data-Driven Approach to Tech Careers

Introduction

Six months ago, coding was completely an unfamiliar territory for me. The allure of the high-tech sector—offering attractive salaries, flexibility, and networking opportunities—prompted me to consider a career shift. After dedicating nearly a month to making a roadmap, I turned to the internet seeking answers to questions commonly pondered by those navigating similar crossroads. However, the pursuit of quick and easy solutions proved more complicated than expected. Python and Javascript emerged as the primary contenders among countless suggestions. The question of the fastest and easiest is not exclusive to Tech job, we usually see that question wandering around in online forums on every task that is moderately or highly difficult to achieve. That is the fastest and easiest way to lose weight, earn the Guitar, to make money and get rid of bed bugs. This question seemed too flat, some influencers suggested starting to learn today, no matter the language or the Tech field, this only got me more confused. In this decision-making process, I explored various tech careers and had an epiphany while delving into the data section: 'The fastest and easiest route to the tech industry is through using data analysis to find out the fastest and easiest entry to tech.' With this revelation, I chose Python, embarking on a six-month journey that led to the creation of the project I present today.

About the Dataset:

The Stack Overflow Developer Survey is an annual questionnaire conducted by Stack Overflow, a prominent online community for developers. This survey gathers insights from a diverse range of developers globally, covering topics such as programming languages, job satisfaction, and educational backgrounds. It serves as a valuable snapshot of the tech industry, although it's essential to recognize that the results represent a sample of the developer community and individual experiences may vary

Section I: Coding Demographics: The Impact of Education on Coding and Professional Status

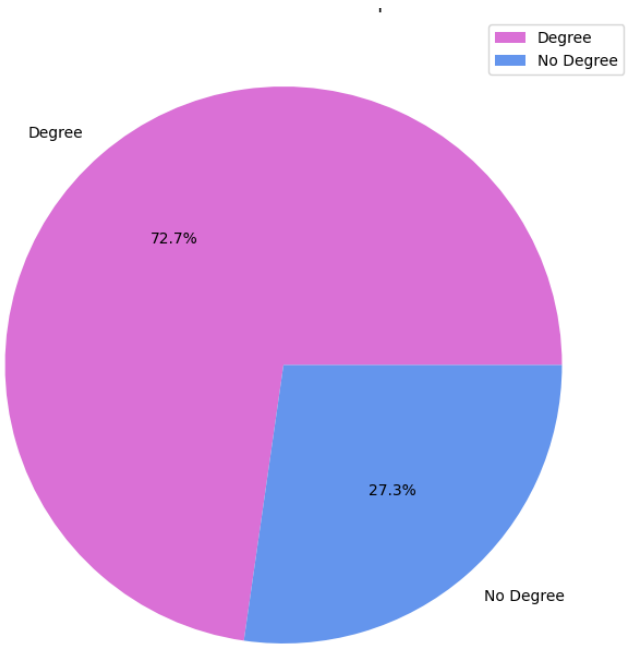
1.1 Degrees or Drive: Assessing the Significance of Formal Education in Tech Careers

In examining the educational background of individuals in the tech industry, I simplified the data by categorizing education levels into two groups: those with a degree and those without. The classifications were as follows:

- **No Degree**: Including individuals with some college/university study without earning a degree, primary/elementary school education, secondary school education, and those with a qualification labeled as "Something else."

- **Degree**: Encompassing individuals with a Bachelor's, Associate, Master's, or Ph.D. degree.

To address the question of whether it's possible to break into the tech industry without a degree, I analyzed the distribution of individuals with and without degrees. The pie chart illustrated that **72.2% of individuals in the high-tech industry held post-secondary degrees**, while **27.8% did not**. This suggests that while entry without a degree is possible, the majority of individuals in the industry have pursued higher education.



This prompts further inquiry: **What factors contribute to individuals without degrees gaining a competitive edge in the market? Does possessing a broader skill set (proficiency in multiple languages), increased proficiency (more years of experience in a specific language), obtaining online certifications, or a combination of these factors play a significant role? Additionally, what is the most influential factor among these considerations?** This analysis seeks to explore these questions and understand the key determinants for success in the tech industry, particularly for those without a traditional degree path.

1.2 Degrees in Tech: How Educational Background Influences Coding and Professional Engagement

In addressing this inquiry, I constructed a horizontal bar chart comparing the main branches of programmers without a degree to those with a degree. The findings uncovered intriguing insights:

- Individuals without a degree who possessed coding skills saw 57% of them pursuing a profession as developers. In contrast, 83.7% of degree holders who code were actively engaged in development roles.

- A notable distinction was observed in coding as a hobby, where 16.5% of individuals without degrees engaged in coding recreationally, while only 9.8% of degree holders pursued coding as a leisure activity.

- Additionally, 13.4% of individuals without degrees identified as learners, compared to a mere 2.7% of degree holders. This suggests that individuals without degrees may exhibit a greater inclination towards learning programming due to the field's perceived benefits. However, it's essential to note that the stark contrast in professional employment percentages (57% vs. 83%) might be influenced by factors such as underage individuals pursuing coding without having completed their education.

To delve deeper into this observation, I narrowed the focus to individuals above 18 years old. Even with this filtering, the proportion of non-degree professionals increased significantly to 66.4%, indicating a notable presence. Yet, it remained relatively lower compared to the professional representation of degree holders, maintaining consistency in other categories.

An intriguing trend emerged, highlighting almost **four times as many hobbyists without degrees compared to those with degrees**. This could suggest that lacking a degree may foster a proclivity for exploring tech as a hobby, emphasizing a more open-minded approach to tech exploration without formal educational constraints.

1. **Does the main branch of those who answered the survey differ between degree and non-degree holders?**

- **Yes**, there are distinctions in the main branches chosen by respondents based on their educational background.

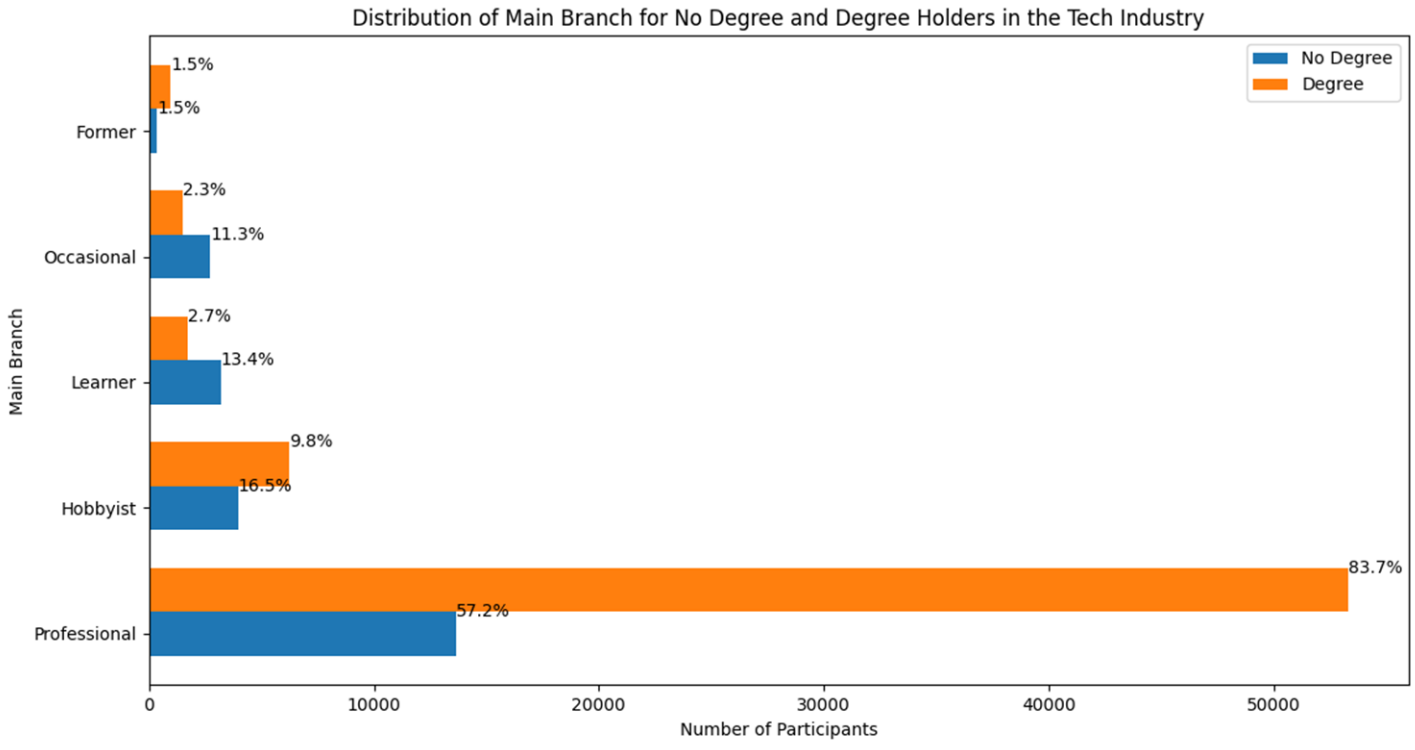
2. Do people choose to get into high tech for different reasons?

- **Yes**, motivations for entering the high-tech field vary among individuals.

3. Is the percentage of all people coding and are professionals affected by having a degree or not?

- **Yes**, the percentage of individuals coding professionally is affected by having a degree.

4. Are there more coders with no degree that don't code professionally?



- **Yes**, there is a notable presence of coders without degrees who engage in coding as a hobby, indicating a potential interest beyond professional endeavors.

1.3 Online Certificates among education groups, facts and numbers

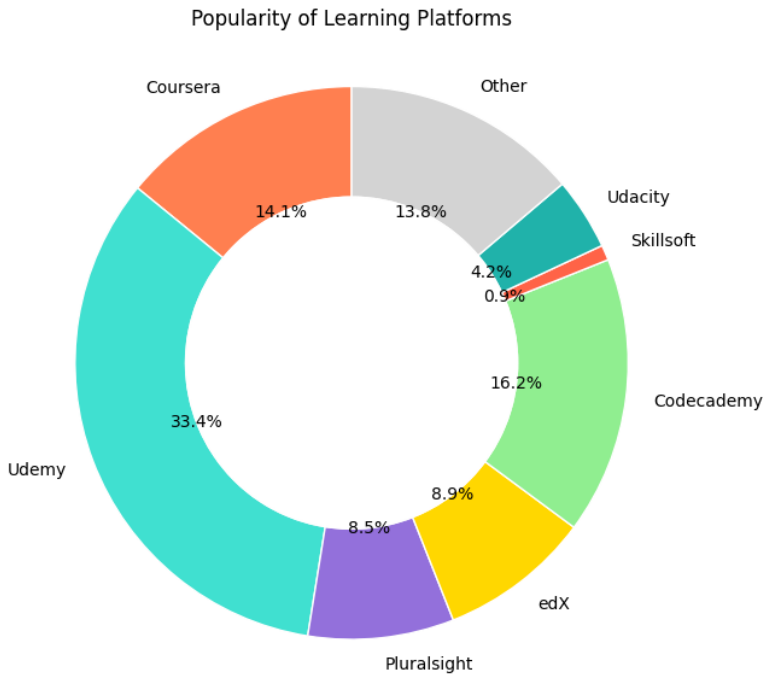
Population Overview:

- For respondents without a degree, the total count was 23,897, with a significant 62% non-response rate (14,886 null responses). Among those with a degree, the total count was 63,728, and the non-response rate was 56% (35,674 null responses).

Online Certificates Usage:

- Despite the high non-response rates, a notable number of both degree and non-degree holders utilized online certificates for learning tech skills.

- Approximately 33.4% of respondents mentioned Udemy, 14% mentioned Coursera, 16.2% mentioned Codecademy, and 8.9% mentioned edX. This indicates the popularity of specific



Learning Methods for Non-Responders:

- Among those who didn't answer the 'LearnCodeCoursesCert' question (indicating no use of online certificates granting platforms), there was a similarity in learning patterns between those with and without a degree.

Insights into Null Responses:

- The analysis reveals that a significant proportion of respondents, both with and without a degree, did not provide an answer to the online certificates question. This highlights a gap in understanding the learning methods of this subgroup.

- While a substantial number of individuals with and without degrees utilize online certificates for learning, a large portion chose not to disclose their preferences. Understanding the reasons behind non-disclosure could provide valuable insights into alternative learning methods or potential barriers to online certificate participation.

Assumption:

Assuming for the sake of this study that null values in the 'LearnCodeCoursesCert' column indicate respondents who didn't use online platforms for learning, the analysis proceeds with this interpretation.

Questions and Answers:

1. How did individuals without a degree attain their tech skills if not from a university?

- **Answer**: Individuals without a degree predominantly acquired tech skills through various means, including documentation, Stack Overflow, written tutorials/books, and video tutorials.

2. Did individuals with a degree use online learning platforms less than those without a degree, potentially due to prior academic education in computer science?

- **Answer:** Both groups showed a similar trend of utilizing online platforms for learning, with a significant proportion indicating no use of online certificates granting platforms.

using platforms such as Udemy, Coursera, Codecademy, and edX, suggesting the popularity of online certificates and open-source platforms.

3. Are online certificates and open-source platforms the most prominent way among coders to learn tech skills, as indicated by the 'LearnCodeCoursesCert' question?

- **Answer**: A substantial number of respondents, both with and without a degree, disclosed

4. **For respondents who didn't use online certificates granting platforms, how did they learn the language?**

- Answer: Those who didn't provide an answer predominantly learned through documentation, Stack Overflow, written tutorials/books, and video tutorials. This underscores the diverse learning methods employed by individuals who may not have utilized online certificates

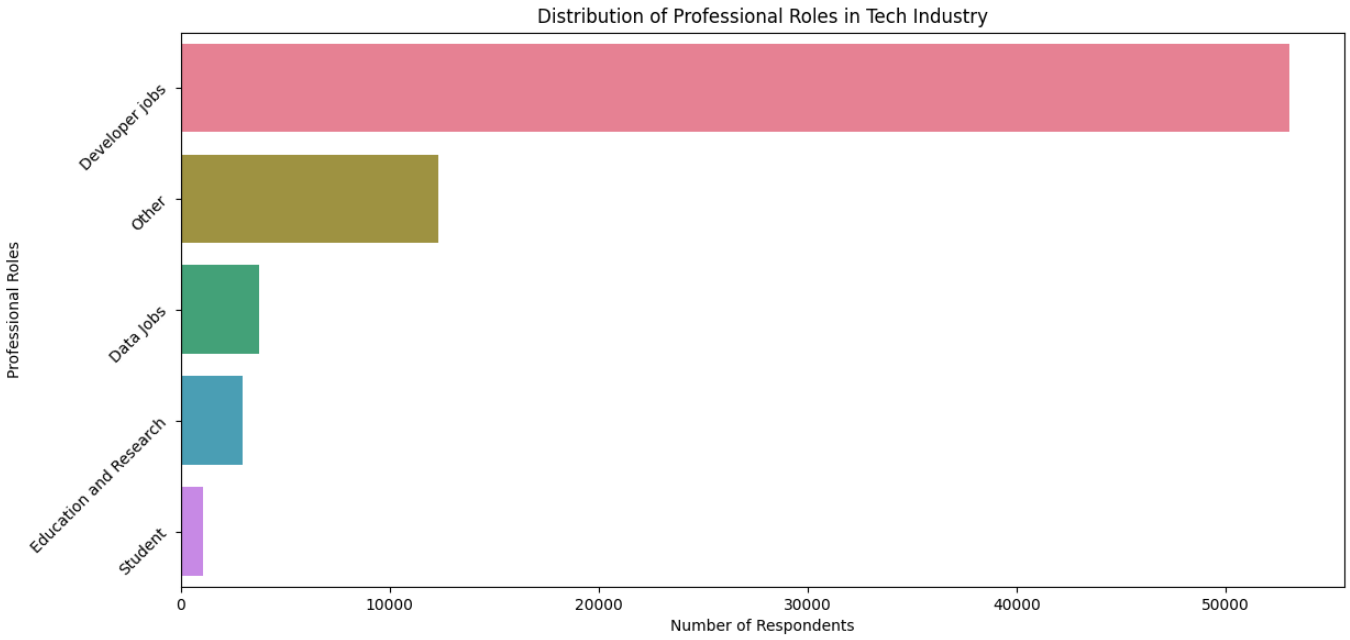
Section II Job Market Analysis

2.1 Tech Roles and Language Preferences: An Analytical Overview

,

1) Which kind of developer jobs were the most prevalent among participants?

For this analysis, various tech job roles were categorized, and the distribution was examined. Developer jobs, particularly those related to back-end, front-end, and full-stack development, were the most prevalent among participants. Data jobs also showed significant representation, while other roles, such as senior executive, system administrator, and designer, were less frequent.

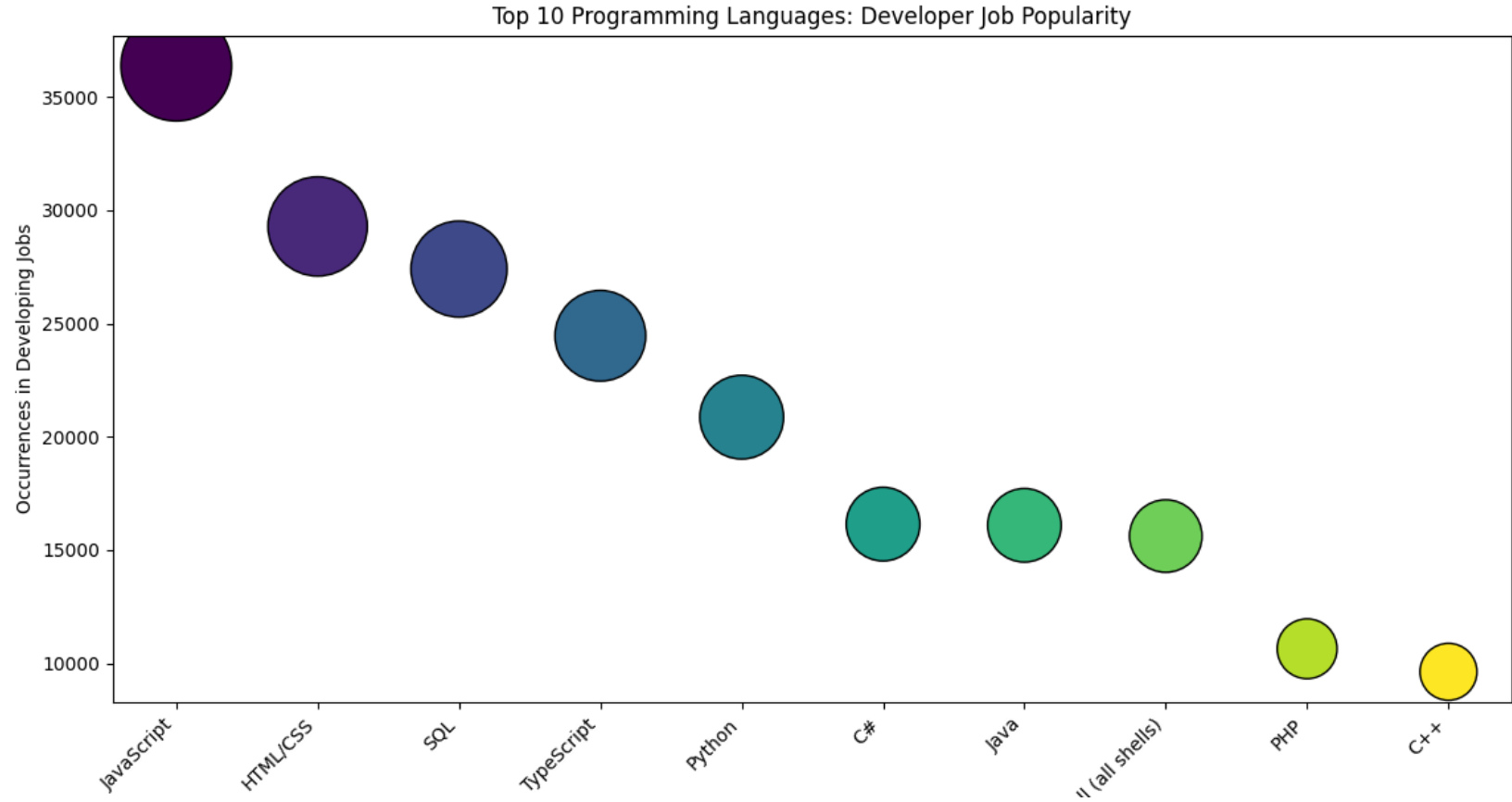


2.2 Language Popularity for different tech roles

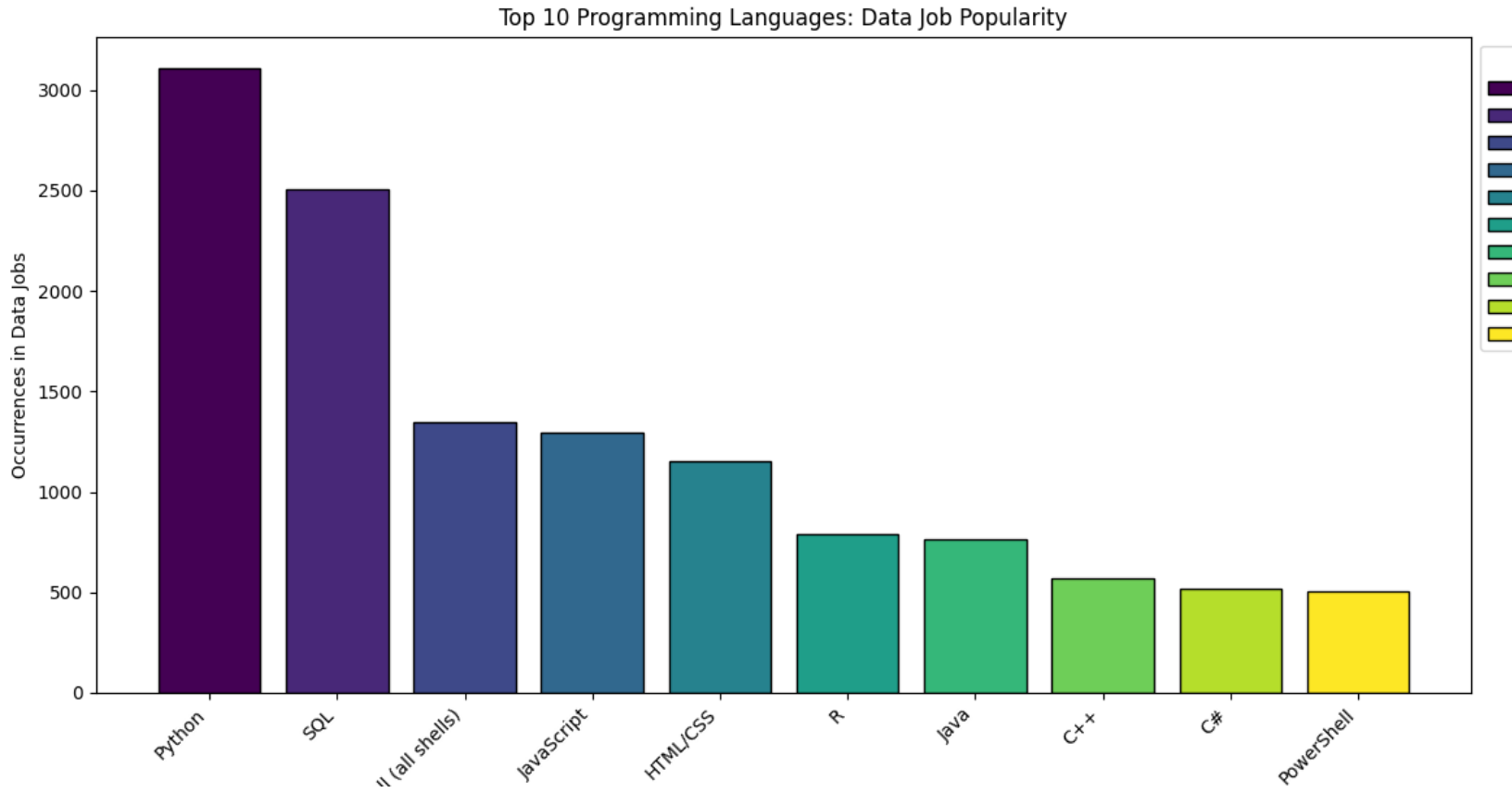
2) Which languages were most popular among different roles in tech?

a) For Developers:

The top languages included **JavaScript, HTML/CSS, SQL, TypeScript, and Python**, as evident from the bubble chart.



) For Data Jobs: **Python, SQL, Shell, JavaScript, and HTML/CSS** were prominent languages.



Considering that HTML/CSS is not considered a standalone programming language, we excluded it from the analysis. Among the most popular tech jobs, including development and data roles, the top languages were **JavaScript, Python, SQL, and Shell**.

In our exploration of popular tech languages among various roles, we observed that JavaScript, Python, SQL, and Shell were prominent across development and data jobs. This challenges conventional wisdom about the ideal starting language, as SQL emerged as a noteworthy contender. This leads us to ponder strategic decisions for a second language—whether to opt for broader applicability, like JavaScript, or embrace specificity with languages such as SQL, Bash, or HTML/CSS. Additionally, our findings highlight the advantageous position that Bash proficiency holds in the job market, irrespective of whether Python or JavaScript is the primary language.

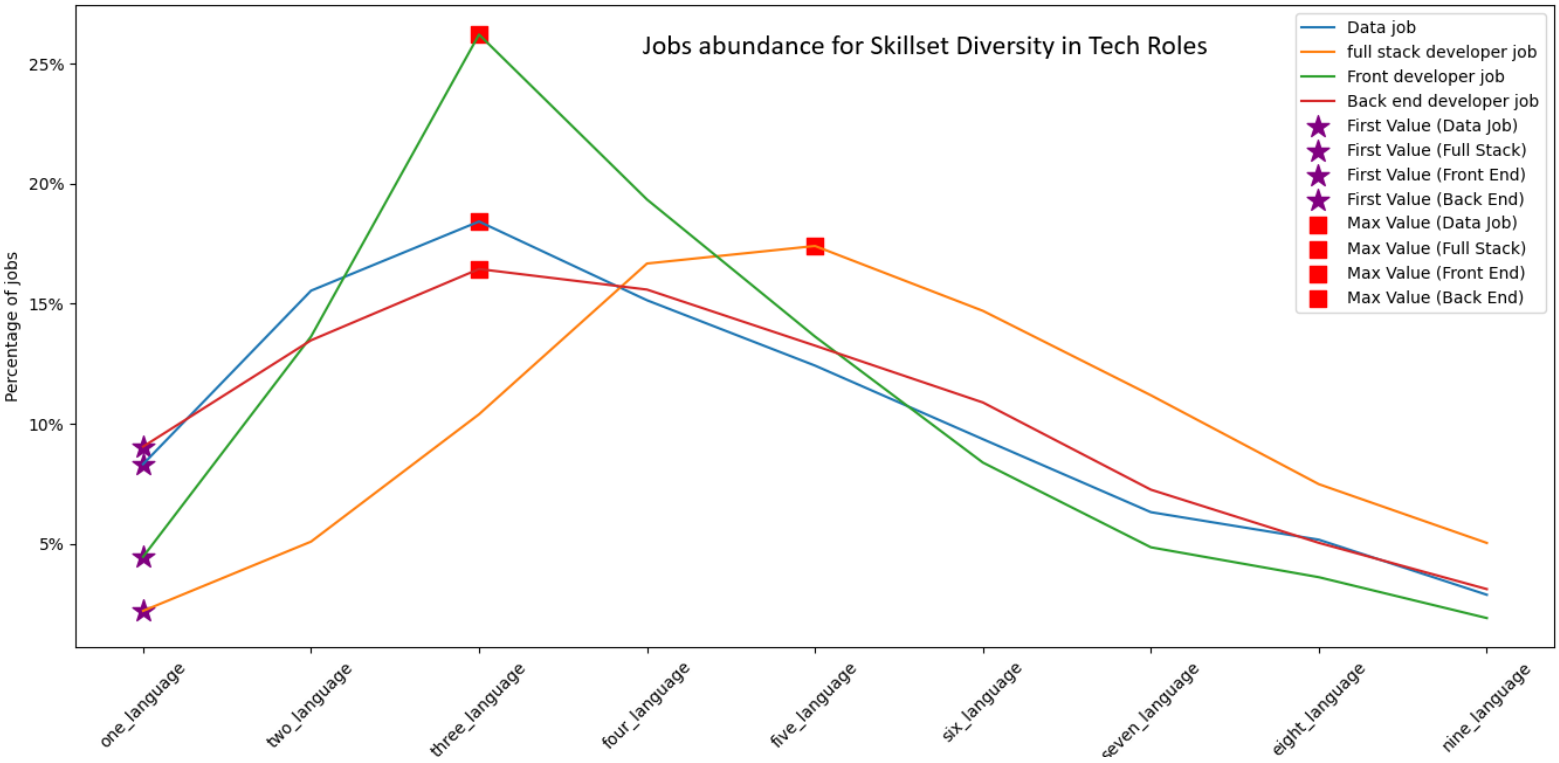
These revelations prompt key questions: **What is the recommended first language according to the data? Should the second language follow the popularity trend or prioritize specificity? When is the opportune moment to diversify one's language repertoire?**

Once you've grasped the basics of your initial language, when should you transition to the next one? The dilemma revolves around the trade-off between deepening proficiency in a single language versus diversifying your skill set. In the next section, we will explore this pivotal question, aiming to provide insights into the optimal approach for individuals navigating the complex landscape of programming languages.

2.3 Diversity vs Proficiency

2.3.1 Correlation between diversity of skillset and competition

In this comprehensive exploration, my dual objective was to uncover the correlation –if any– between the diversity of skillsets, measured by the number of languages, and the abundance of job opportunities. Simultaneously, I aimed to identify the profession that presented the highest number of job openings for individuals proficient in just one language. This multifaceted approach seeks to furnish strategic recommendations for aspiring tech professionals, grounded in a nuanced understanding of language requirements and job distribution. Leveraging a line plot, my intention was to establish connections between skillset diversity and job availability.



To make a fair comparison among different tech roles, we normalized the data. This means **we adjusted the numbers by dividing the job count for each language diversity level by the total job count for that specific role.** This step was crucial to even the playing field and provide unbiased insights, considering the variations in job counts across different professions. The goal was to offer accurate recommendations based on the observed connections between language proficiency, job availability, and competition in the tech industry

The analysis yielded a recognizable correlation between the number of programming languages one is proficient in and the abundance of job opportunities. As a start, we know that a higher job count for a specific variable, be it the number of languages, proficiency levels, or educational qualifications, results in lower competition for a particular job. For newcomers, it's crucial to recognize **the negative correlation between competition and Diversity of skillset (the number of languages)**; the more languages one is proficient in, the greater the job availability and the lower the competition. **Notably, data and back-end roles demonstrated the highest percentage of total jobs for individuals proficient in only one language, indicating lower competition for entry-level positions compared to front-end and full-stack roles.**

On the other hand, examining the diversity level (number of languages) associated with the most significant number of jobs (and consequently, the least competition), **data, back-end, and front-end roles peaked at three languages**. In contrast, **full-stack** roles exhibited their peak at **five languages**. Interestingly, **the total number of jobs for individuals proficient in three languages and working in front-end roles surpassed those in data, back-end, and even full-stack roles**. This suggests that the least competition along this spectrum is for individuals proficient in three languages, specifically those popular and in demand in the front-end field, such as Java, JavaScript, HTML, CSS, and C#. The applicable question arising from these findings is the optimal path to secure a tech job. **Does being proficient in three languages take more time than focusing on one language?** Given the established correlation between skillset diversity and competition, the subsequent inquiry delves into the correlation between proficiency (time spent learning, working, contributing to open source) and competition. If such a correlation exists**, which is stronger: more proficiency in one language resulting in less competition or less proficiency in three languages leading to greater competition? This complex question will be explored in the upcoming section.**

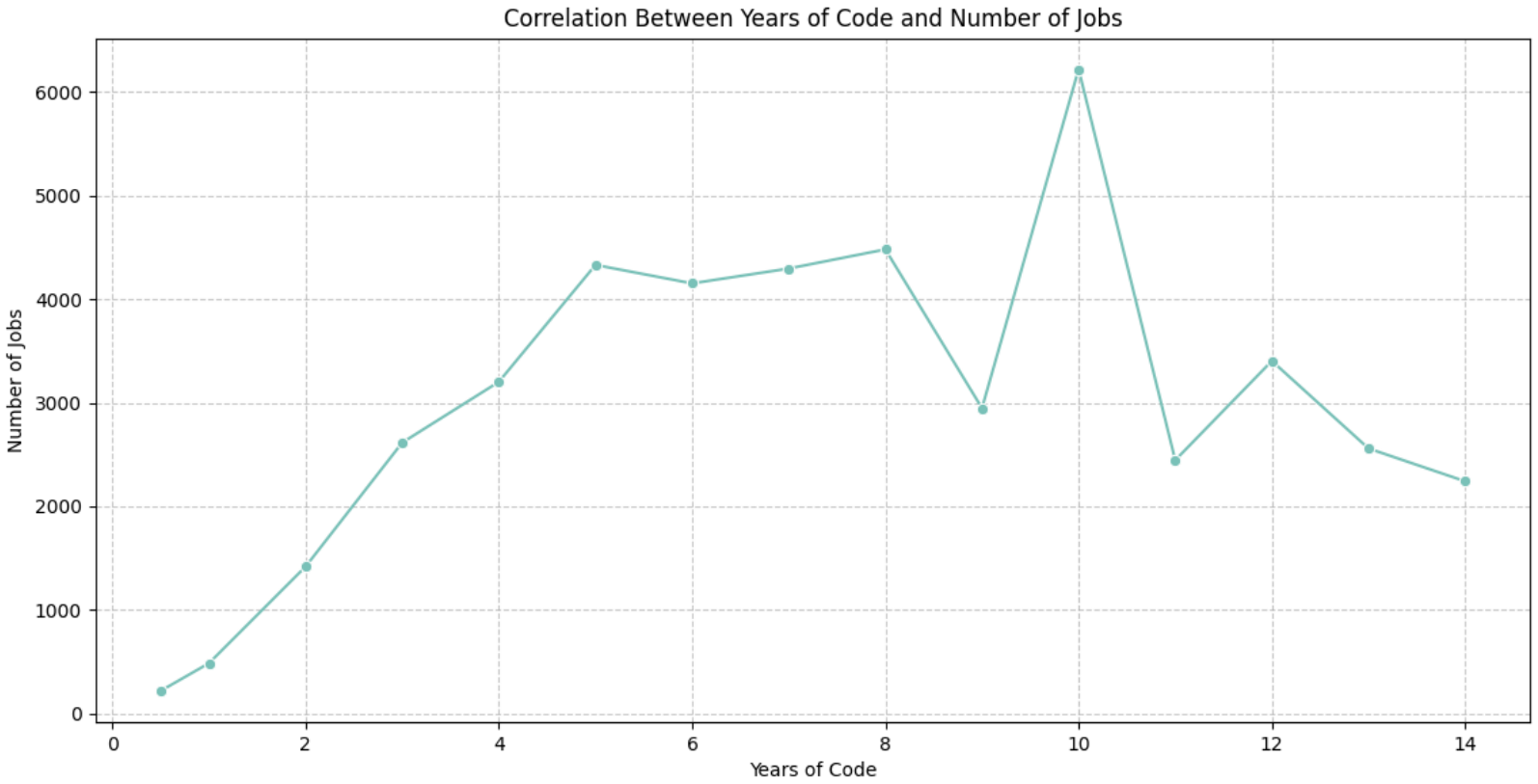
2.3.2 Correlation between Proficiency of a language and Competition

Section I : Total Proficiency

Questions

Is the number of jobs in the tech industry among professionals affected by their number of years of coding? Is the correlation positive or negative?

To address the question of whether the number of years of coding experience influences job opportunities in the tech industry, a line plot was employed. The analysis revealed a clear positive correlation in the first five years, indicating that individuals with more coding experience within this initial period had access to a higher number of job opportunities. Notably, the plot depicted a substantial increase, with **over 4000 jobs for individuals with five years of coding experience**, in contrast to **less than 1000 jobs for those with less than one year** of experience.



However, the correlation dynamics changed beyond the fifth year, showing fluctuations and inconsistency. After this threshold, the line displayed variability, suggesting an unclear correlation between the number of years of coding experience and job counts. The correlation appeared to be unpredictable, exhibiting both positive and negative trends.

In conclusion, the analysis indicates that coding experience significantly influences job opportunities in the first five years, with a positive correlation. **The negative correlation between competition and Proficiency (the Total experience years)** the more proficient someone was the less competition they had in the job market. Beyond this period, the relationship becomes less discernible, suggesting that the impact of coding experience on job counts becomes less consistent and predictable.

Section II: Professional Proficiency

Questions

1. Among different experience levels (Junior, Mid, Senior), which category had the most job postings in the tech industry?

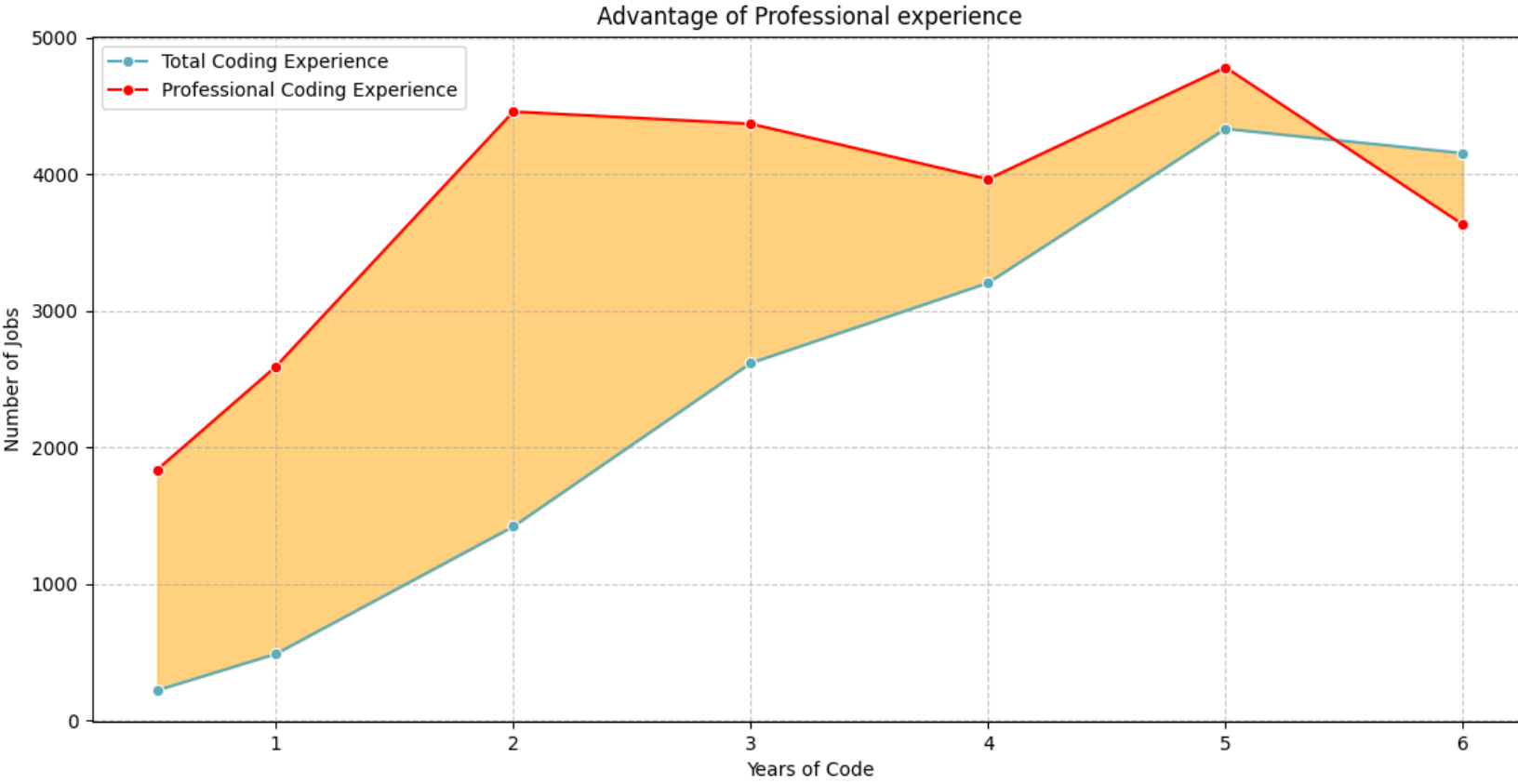
2. How does the competition trend vary in the first 5 years of coding experience?

3. What insight can be gained from comparing total coding experience (YearsCode) and professional coding experience (YearsCodePro) in terms of job opportunities?

Analysis

The analysis focuses on understanding job trends in the tech industry based on coding experience levels, particularly within the first 5 years. Three categories are considered: **Junior (0-3 years), Mid (3-5 years), and Senior (5+ years).** The line plot for total coding experience *(YearsCode)* reveals that, within the first 5 years, there is a negative correlation between competition and total experience. In other words, the more total coding experience one has, the higher the chances of finding job opportunities.

However, the key insight emerges when comparing **total experience with professional experience *(YearsCodePro)*.** By plotting both lines on the same graph and filling the area between them, the analysis showcases the advantage of professional proficiency. Despite the competition trends for those in the early years of coding, individuals with professional experience have a significant edge. For instance, someone with 2 years of professional experience has nearly four times the advantage over someone with 2 years of total coding experience. This underscores the importance of gaining professional proficiency early in one's career.



Beyond the 5-year mark in professional experience **(Senior level),** an intriguing trend emerged. **The number of jobs for individuals with more total experience, including non-professional exposure such as learning additional technologies or contributing to open source, surpassed the count for those with a focus on professional experience**. Which means, at Senior level, continuous learning played a significant role in landing a job. While our study primarily targets aspiring juniors, it's noteworthy to mention this observation, indicating that accumulating diverse skills and experiences can become a valuable asset as one progresses in their career.

For older folks making a career shift against Gen Z, who practically grew up with tech, it might seem like a tough match. But here's the scoop – if you can leverage your network and jump into the professional game as soon as possible, you've got a significant advantage. Think of it as a race where early starters have their perks, but those building a robust professional game can catch up and even lead.

Answers

1. The category with the most job postings in the tech industry varies based on experience levels. Along the graph, there was more amount of Junior jobs. Most of the job count was for people who were Juniors with 5 years of experience. Then Juniors with 3 years of experience, then, interestingly, Juniors with 4 years of total coding experience. The correlation between Total Proficiency, Professional proficiency and job count looks clear along the plot. In Junior jobs, competition is high but decreases with more total and professional coding experience. However, professionals with 3-5 years (Mid) or 5+ years (Senior) of experience face less competition, with the Senior category having fewer job opportunities but lower competition.

2. In the first 5 years of coding experience, the analysis reveals a negative correlation between competition and total coding experience. As total experience increases, the competition decreases, indicating that those with more years of coding are more likely to find job opportunities.

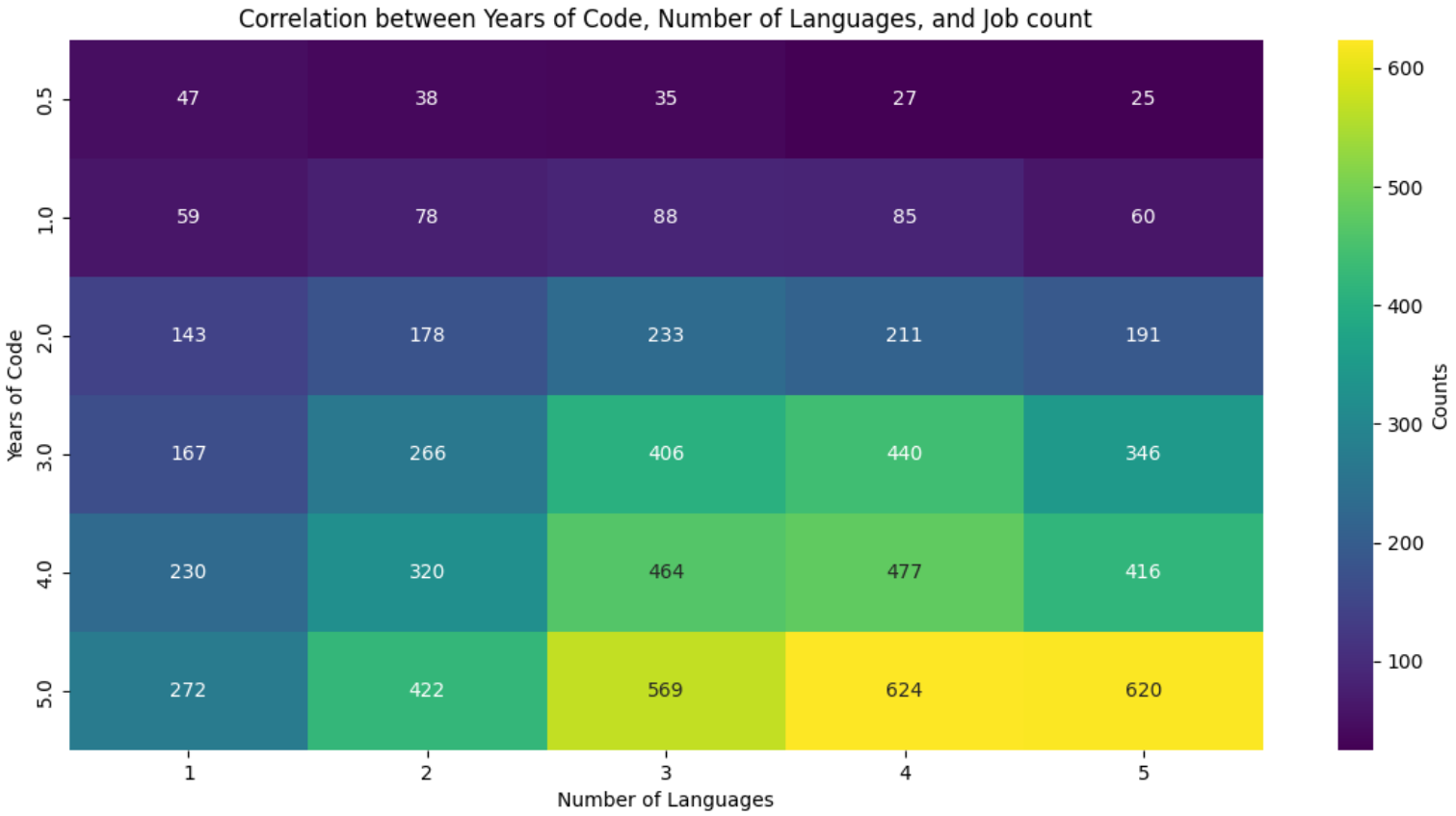
3. The comparison between total coding experience (YearsCode) and professional coding experience (YearsCodePro) highlights **a significant advantage** for individuals with professional proficiency. The advantage is particularly pronounced in the first 5 years, where having professional experience provides a substantial edge in competing for job opportunities in the tech industry.

2.3.3 The correlation between Diversity, Proficiency, and Competition

Analysis

After uncovering the correlation between Diversity(number of languages), Proficiency(years of total experience). and job count. This section delves into piecing the puzzle. we delved into unraveling the intricate correlation between job count, diversity (number of languages), and proficiency (years of total experience).

In this section we are trying to understand **what is the more strategic choice between dedicating more time to mastering a single language or diversifying skillsets to navigate the competitive tech job landscape**. Leveraging a heatmap with the number of languages and years of coding as axes and job count represented by the colormap, we uncovered nuanced insights dependent on the individual's position in their tech job journey.



- For those with less than 1 year of total coding experience

- Tracing the most amount of jobs on the heatmap for “Less than 1 year” category shows that the market is **favoring a focused approach on a single language**. This insight could guide aspiring tech professionals in their initial stages.

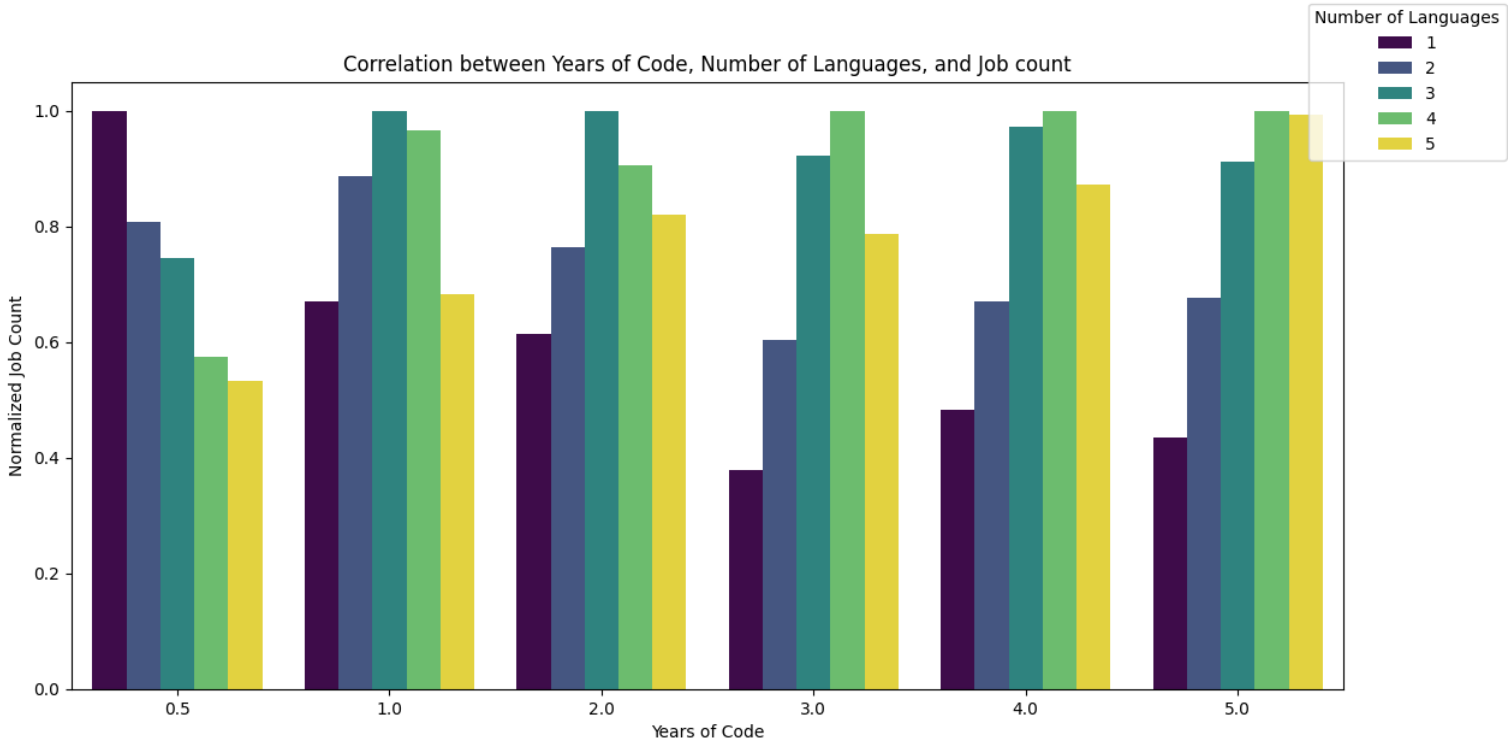
- Upon reaching one year of total coding experience

- A noticeable shift occurred, revealing that proficiency in 3 languages emerged as an optimal strategy for accessing job opportunities. This trend persisted for those with 2 years of total coding experience.

- As total coding experience increased

- The heatmap showcased a pronounced peak in job opportunities for individuals with 4-5 years of total coding experience, proficient in 4 or 5 languages. This observation highlighted the dynamic and evolving correlation between proficiency, diversity, and competition at different seniority stages. This nuanced understanding empowers individuals to make strategic decisions aligned with their specific tech journey stage. It provides a detailed compass for navigating the dynamic terrain of the tech job market, catering to individual career goals and industry dynamics.

This interesting finding prompts further exploration. It suggests that the connection between skill level, diversity, and competition changes at different career stages. To investigate this, I used a stacked bar graph. Normalizing the data helped us see a clearer picture by reducing the influence of the total job counts. Now, let's dive into the details of this graph in the next part.



Examining the stacked bar chart reveals intriguing insights. In the category of less than one year, **a substantial number of jobs were secured by those proficient in a single language**, exhibiting considerable variance across different language counts. Notably, for this category**, job count decreased as the number of languages increased**. Moving on to the 1-2 years category, we observe that **the optimal scenario for job acquisition is linked to proficiency in three languages**. Calculating standard deviation highlights less variance compared to the less-than-one-year category.

For individuals with 3, 4, or 5 years of coding experience, **the highest job count is associated with proficiency in four languages**, accompanied by substantial variance in job counts. This leads to significant conclusions:

- Questions pertaining to the duration of learning a language and when to transition to another are crucial for aspiring juniors.

- Initial language proficiency significantly influences future opportunities, emphasizing the importance of a focused approach during the first year of language learning.

- In the "Less than 1 year" category, Proficiency in one language was the most important factor in the chances of getting a junior job.

- In the 1-2 years coding journey, considerations of diversity or proficiency take a back seat, with other factors such as professional and total experience coming into play (e.g., contributing to open source and non-paid internships).

- For individuals with 3, 4, or 5 years of coding experience, diversifying the skillset becomes more critical than proficiency in a specific language.

This leads to the conclusion that aspiring juniors are advised to spend the initial six months mastering a general-purpose language and the subsequent six months:

- Learning a more specific and less versatile language (e.g., SQL, Bash, HTML: CSS).

- Gaining hands-on experience through online projects, contributions to open source, and internships.

- Building a portfolio promptly to commence applications for tech jobs and gain professional experience

2.4 Choosing the Language That Opens Doors

In my recommendation to focus on back-end development or the data field, I checked the popularity of programming languages in these domains by examining the occurrence of each language in the "Languages have worked with" field. Python (240 occurrences), SQL (228 occurrences), R (not specified), and JavaScript (not specified) emerged as the most popular languages.

As we saw previously , the peak amount of jobs for both data and back-end happened at knowing 3 language. Analyzing most number of occurrences for any language combination, however, showed an interesting point**: while the total sum of all jobs for both data and back-end happened at 3 languages, the maximum amount of jobs for any language combination happened at 2 language combination (Python & SQL 240 occurrences) for data jobs and only one language (Java 317 occurrences)**. As we previously concluded that the amount of professional years of coding experience (Professional Proficiency) put coders at a remarkable advantage. Putting your foot in the industry as soon as possible is significantly important. Thus, tracing the most occurrences for any combination of language, and generally speaking, the question of what language should aspiring Junior learn first is consequential.

To add versatility to our recommendation, I explored the most frequent language combinations for both data and back-end jobs, focusing on the top 5 occurrences in each category:

Back-end :

-Java (317 occurrences),

-Python (197 occurrences)

-C#(193 occurrences)

- Java/SQL combination (157 occurrences)

- C#:SQL combination(74 occurrences)

Data :

-Python:SQL combination (229 occurrences)

- Python (217 occurrences)

- Bash/Shell (all shells);Python;SQL combination (111 occurrences)

- Python;R;SQL combination (83 occurrences)

- Bash/Shell (all shells);Python(51 occurrences)

Emphasizing the correlation between years of coding and job abundance, I highlighted that **for those certain about pursuing back-end development, focusing on Java for at least the first year is recommended**. However, considering this study's focus on individuals potentially making a career shift for the quickest entry into high-tech jobs, I reiterated that the specific field is not the main question. To address the primary question, **given that the second most jobs were for people who knew Python and SQL appeared in significant language combinations for back-end roles (Java:SQL, C#:SQL)**, I maintained the recommendation to start with learning Python. This is

because **proficiency in Python provides versatility, allowing individuals to explore both data libraries like pandas and numpy and gain a brief introduction to backend libraries like Django and Flask**. Ultimately, this approach enables them to decide whether to pursue data jobs or back-end development. Knowing Python on its own also provides a significant advantage for those who opt to become back-end developers.

Conclusion:

In summary, our exploration of the high-tech industry trends, using data from the Stack Overflow sample, has provided valuable insights. Initially, we observed a preference for individuals with post-secondary degrees and debunked the exclusivity of online certificates to university graduates. Subsequently, we delved into the realm of online certifications, highlighting the most popular platforms for obtaining them. Transitioning into the job market analysis, we explored the correlations between skillset diversity (number of languages), language proficiency (years of coding), and professional versus total experience. By intertwining these variables, we derived meaningful insights to guide aspiring tech professionals. Further, we delved into the frequency of language combinations to make informed recommendations for starting with a first language. In the concluding section, we present our final recommendations based on these comprehensive findings

Recommendations:

- Dedicate the first six months to mastering a general-purpose language and the subsequent six months to a more specific and less versatile language.

- Gain hands-on experience through online projects, contributions to open source, and internships.

- Build a portfolio promptly for tech job applications and gaining professional experience.

- Consider the nuanced correlation between years of coding and job abundance.

- For those inclined towards back-end development, focus on Java for the first year.

- However, the primary question is not confined to a specific field but revolves around career shifters seeking swift entry into high-tech jobs.

- Python, with 197 occurrences and a significant role in back-end language combinations, emerges as a strong starting point.

- Proficiency in Python provides versatility for exploring both data and back-end development.

- Knowing Python on its own offers a substantial advantage for aspiring back-end developers.