Technology Trends & Developer Demographics: Insights from Global Survey Data

Carlo Lopez

07/07/25

Presentation Outline

1. Executive Summary
2. Introduction
3. Methodology
4. Programming Language Trends and Findings
5. Database Trends and Findings
6. Dashboard 1: Current Technology Usage
7. Dashboard 2: Future Technology Trends
8. Dashboard 3: Demographics
9. Insights from Dashboard
10. Overall Findings and Implications
11. Conclusion
12. Appendix

Executive Summary

This project analyzes current and future technological trends as well as developer demographics using interactive dashboards created in IBM Cognos. The findings are based on global developer survey data and highlight both what technologies are widely used today and which ones are gaining momentum for the future.

Key Findings:

- **Programming Languages:** JavaScript, SQL, and HTML/CSS are the most used languages today. However, modern languages like **Go** and **Rust** are rapidly gaining popularity, reflecting a shift toward performance and scalability.
- **Databases:** PostgreSQL and MySQL dominate current use, while platforms like **Supabase** and **Redis** are emerging as future-focused, developer-friendly solutions.
- **Web Frameworks & Platforms:** Node.js and React lead current development stacks, with increasing interest in frameworks like **Next.js**, **FastAPI**, and deployment platforms like **Vercel** and **Netlify**.
- **Demographics:** The majority of respondents are aged **25–34** and hold at least a **bachelor's degree**, with representation from around the world.

Introduction

This project explores key technology trends and developer demographics through data-driven visualizations. By analyzing responses from a global developer survey, the goal is to understand which tools are currently dominant, which ones are gaining popularity, and how factors such as age, education, and geographic location relate to these trends.

The insights are presented through three interactive dashboards built in IBM Cognos:

1. **Current Technology Usage**
2. **Future Technology Preferences**
3. **Demographic Breakdown**

Methodology

Data Source: Global developer survey.

Tool: IBM Cognos dashboards.

Collection Method: Self-reported responses on tech usage and preferences.

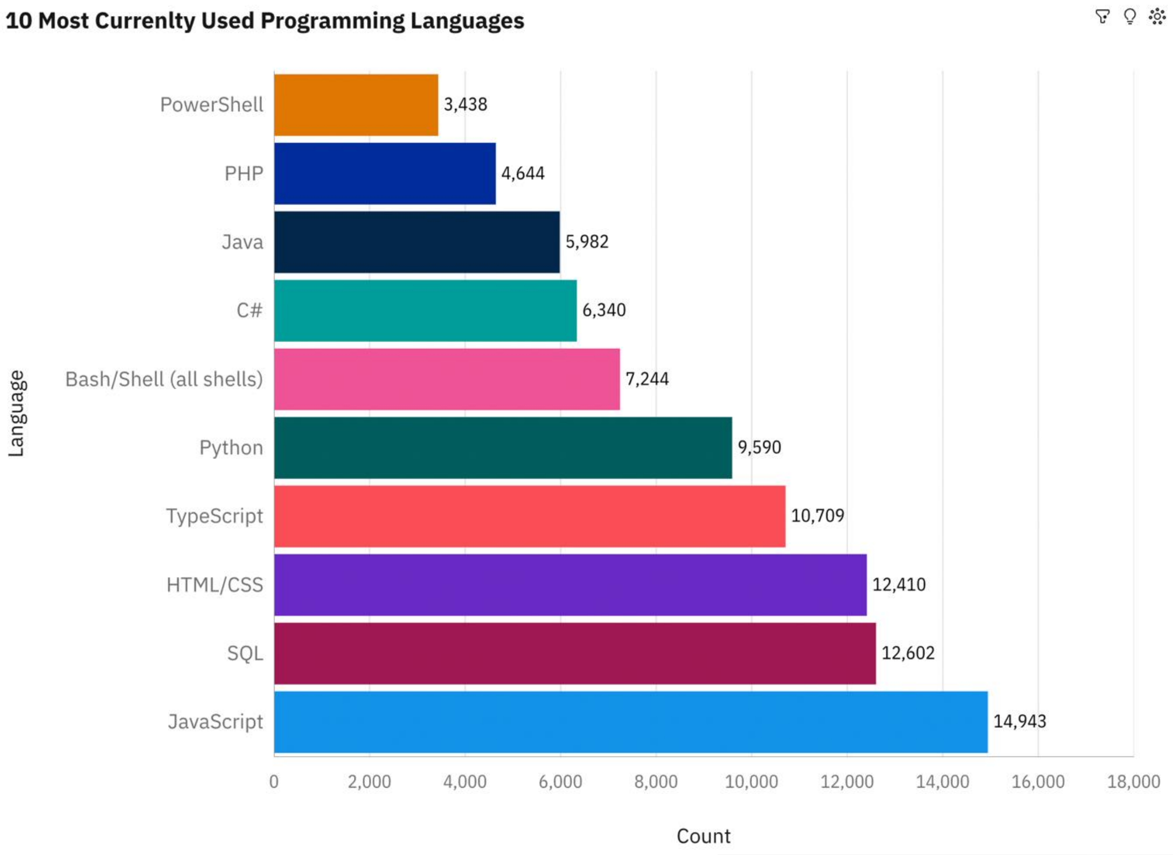
Data Wrangling:

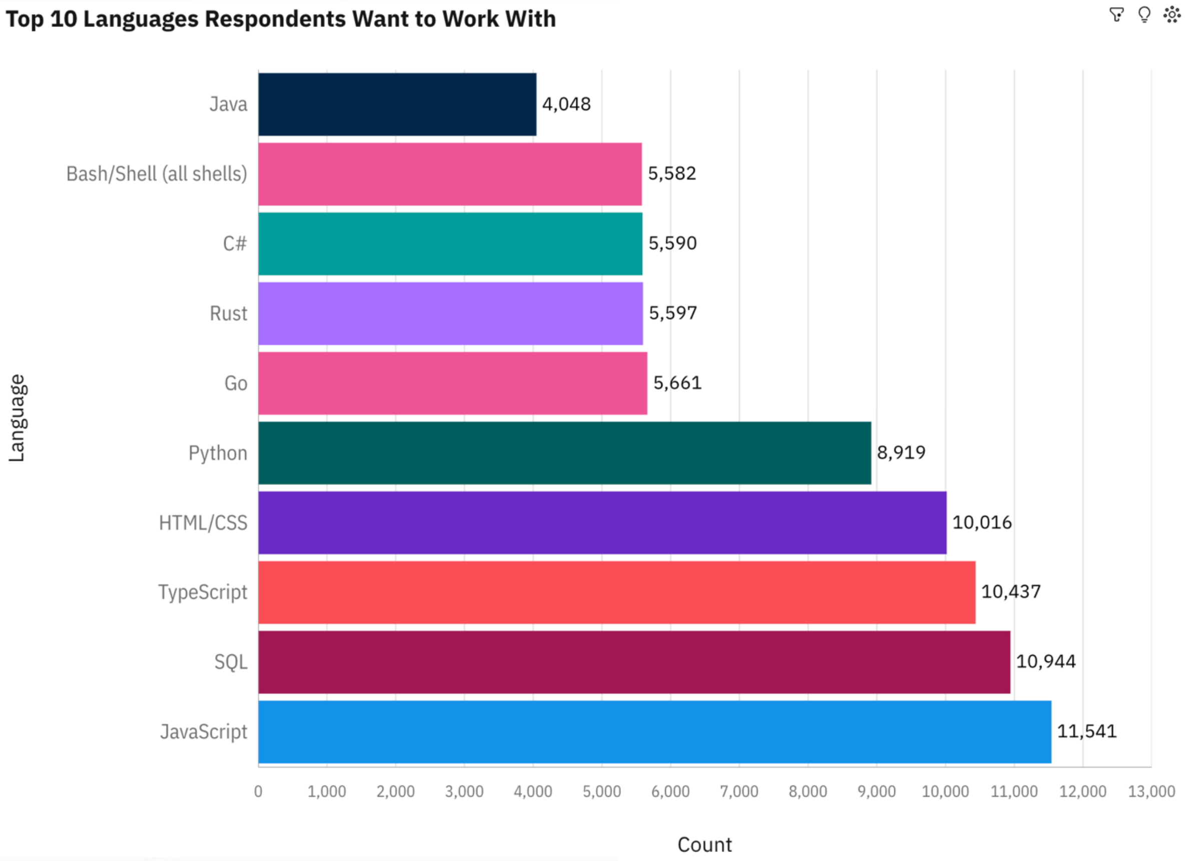
- Cleaned necessary data columns
- Filtered to top 10 items per category.
- Aggregated respondent counts.
- Grouped data by demographics.
- Applied sorting and labels in Cognos for clarity.

Programming Language Trends

*This visualization depicts **current** trends in **programming languages** use:*

*This visualization depicts **future** trends in **programing languages** use:*





Skills Network

IBM

Programming Language Trends – Findings & Implications

Findings:

1. **High Overlap:** JavaScript, SQL, and Python are top in both current use and future interest.
2. **New Interest:** Go and Rust are rising in popularity, despite lower current use.
3. **Shifting Away:** PHP and PowerShell are used now but less desired for future work.

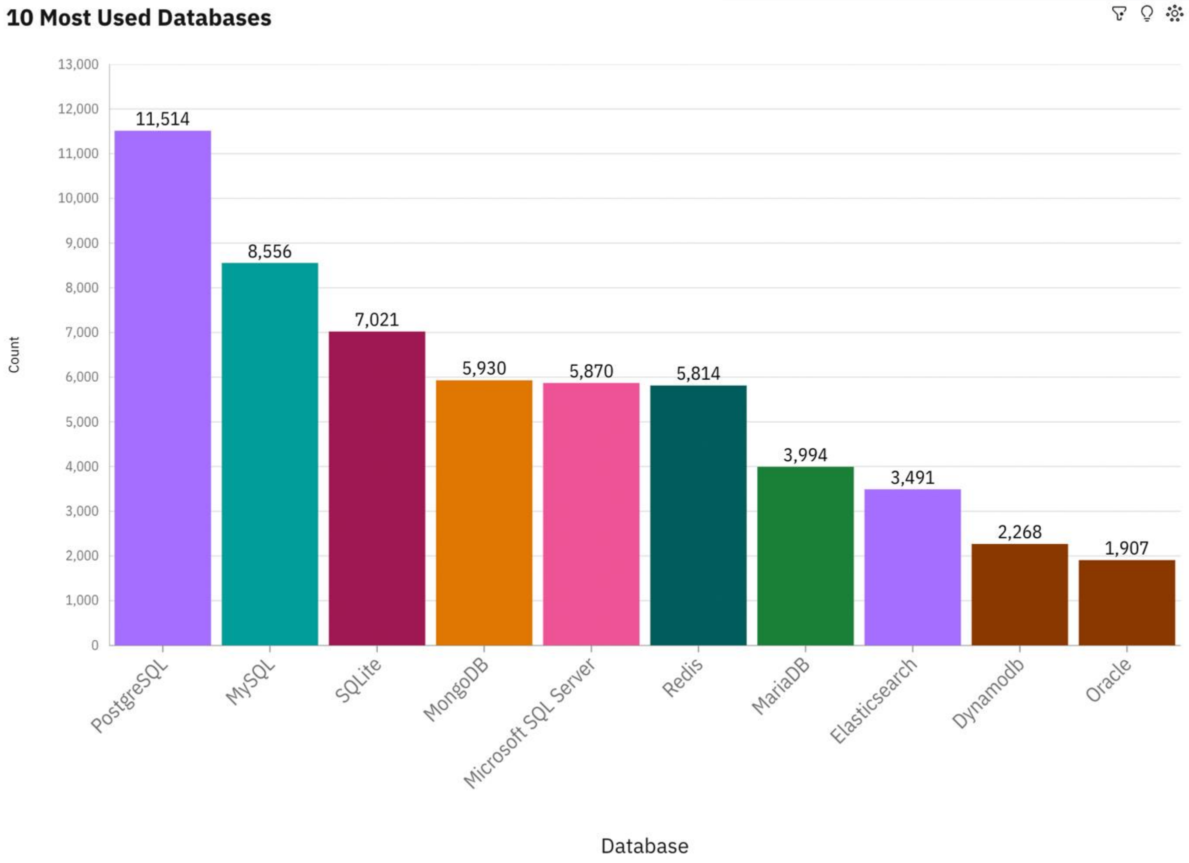
Implications:

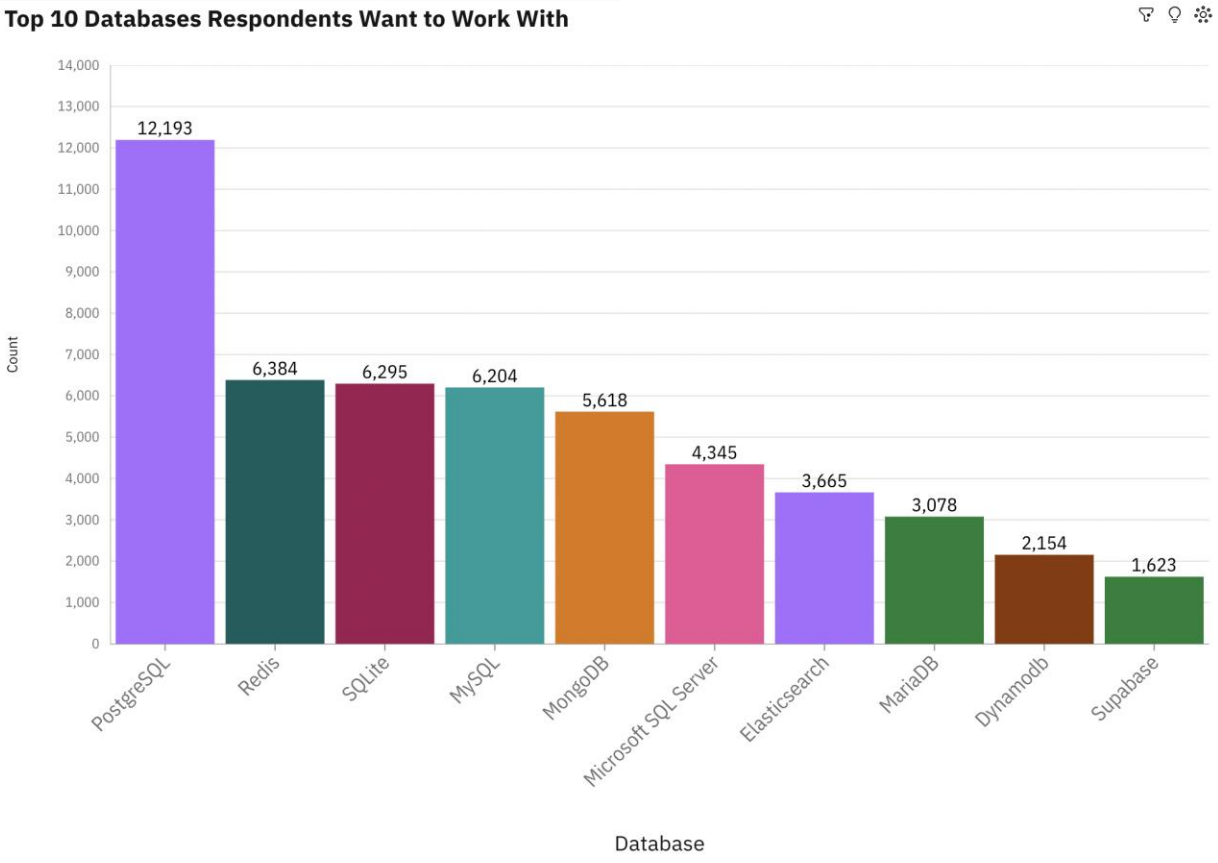
1. **Upskill in Modern Tools:** Organizations should support learning Go, Rust, etc.
2. **Sustain Core Languages:** Continued investment in JavaScript, SQL, and Python is essential.
3. **Plan for Change:** Teams using declining languages should begin planning for transitions.

Database Trends

*This visualization depicts **current** trends in **Database** use:*

*This visualization depicts **future** trends in **Database** use:*


10 Most Used Databases


Top 10 Databases Respondents Want to Work With

Database Trends – Findings & Implications

Findings:

1. **PostgreSQL dominates** both current use and future interest, making it a long-term favorite among developers.
2. **Modern databases like Redis and Supabase** are gaining traction, reflecting a shift toward lightweight, scalable solutions.
3. **Legacy systems such as Oracle and Microsoft SQL Server** rank lower in future interest, signaling a possible decline in relevance.

Implications:

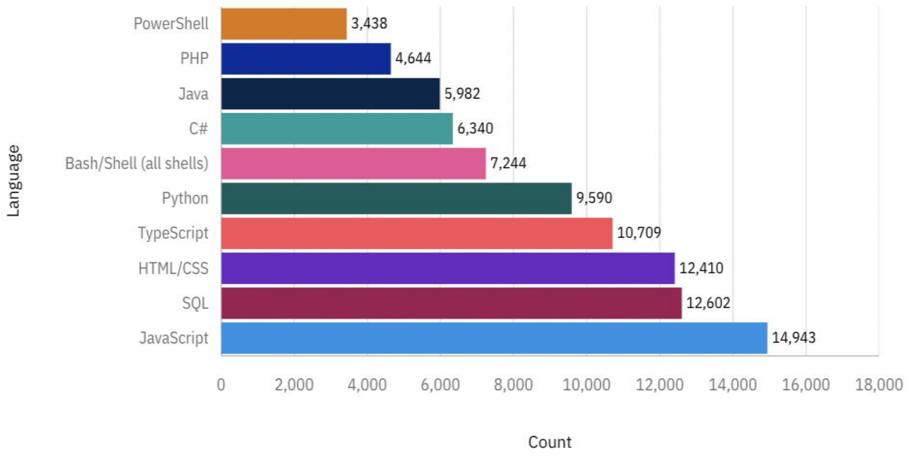
1. **Continue investing in PostgreSQL** as a versatile, developer-approved choice.
2. **Explore and support emerging platforms** like Redis and Supabase to stay aligned with future developer preferences.
3. **Prepare for reduced reliance on legacy databases** by planning gradual transitions where appropriate.

Dashboards

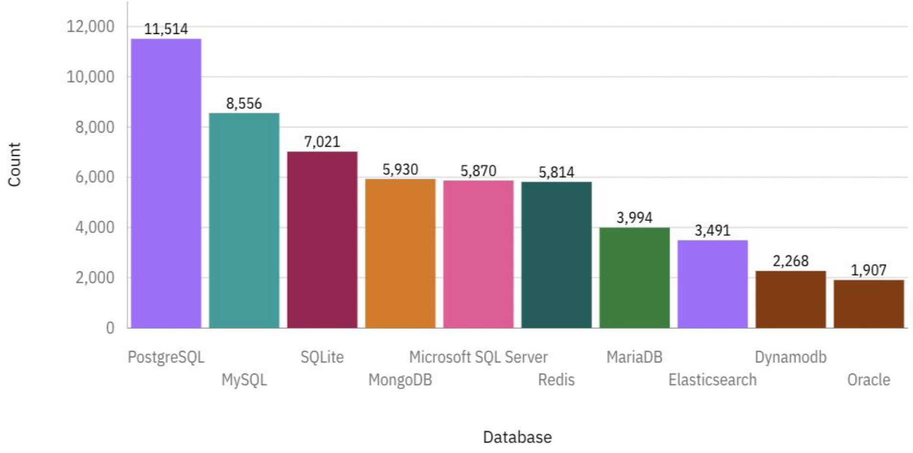
1. Current Technology Trend
2. Future Technology Trend
3. Demographic Trend

Dashboard 1: Current Technology Use

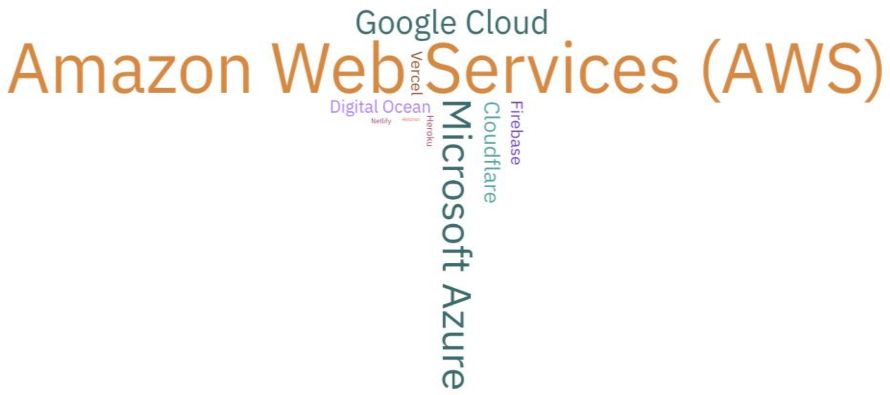
10 Most Used Programming Languages



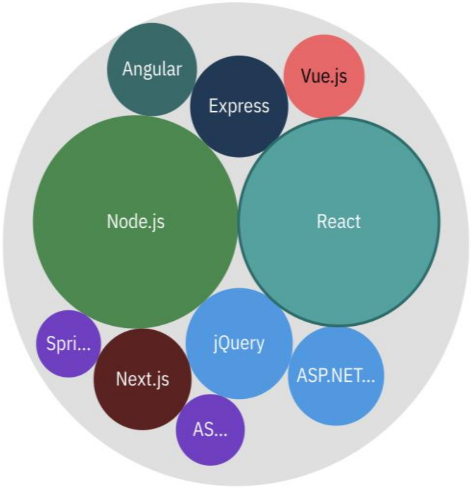
10 Most Used Databases



10 Most Used Data Platforms



10 Most Used Web Frames

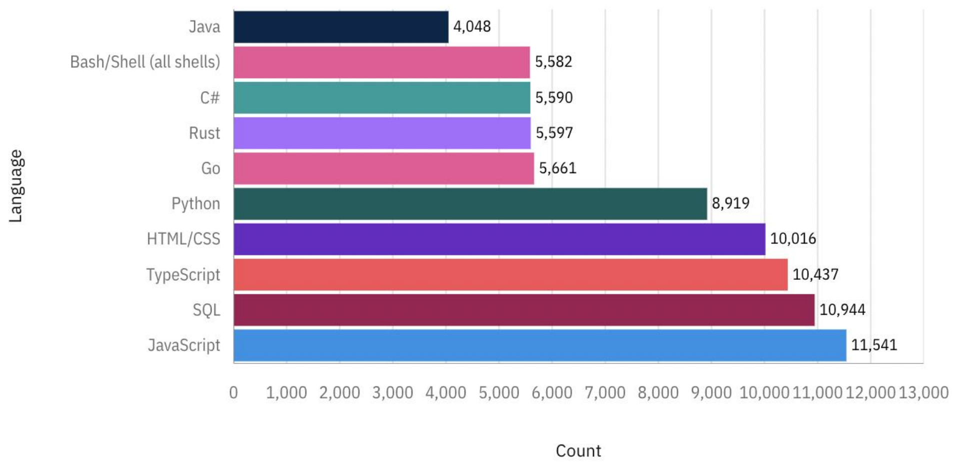


Skills

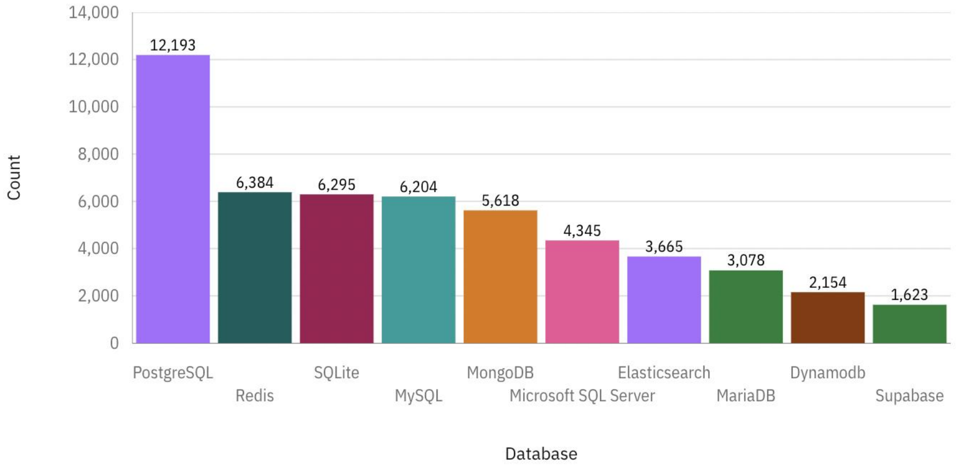
IBM

Dashboard 2: Future Technology Trends

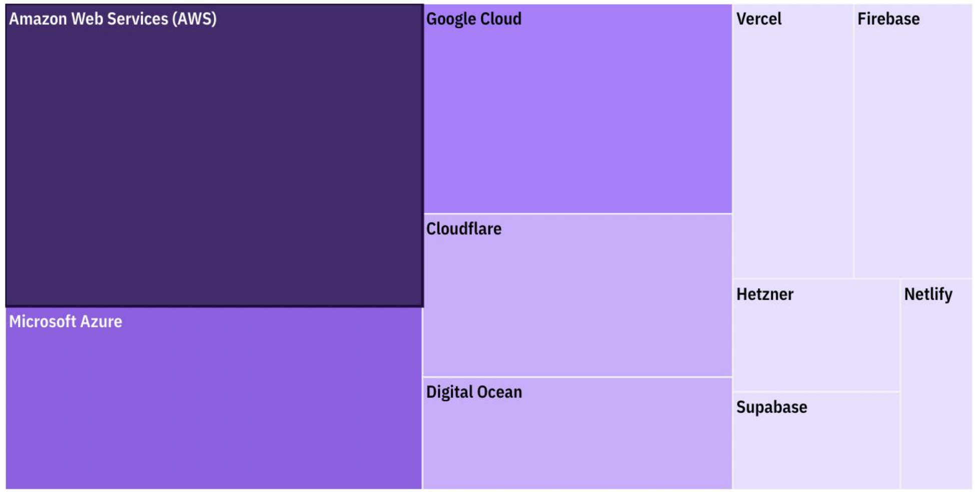
Top 10 Languages Respondents Want to Work With



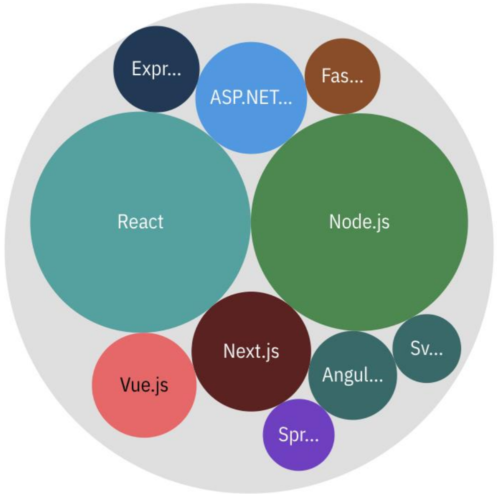
Top 10 Databases Respondents Want to Work With



Top 10 Platforms Respondents Want To Work With.



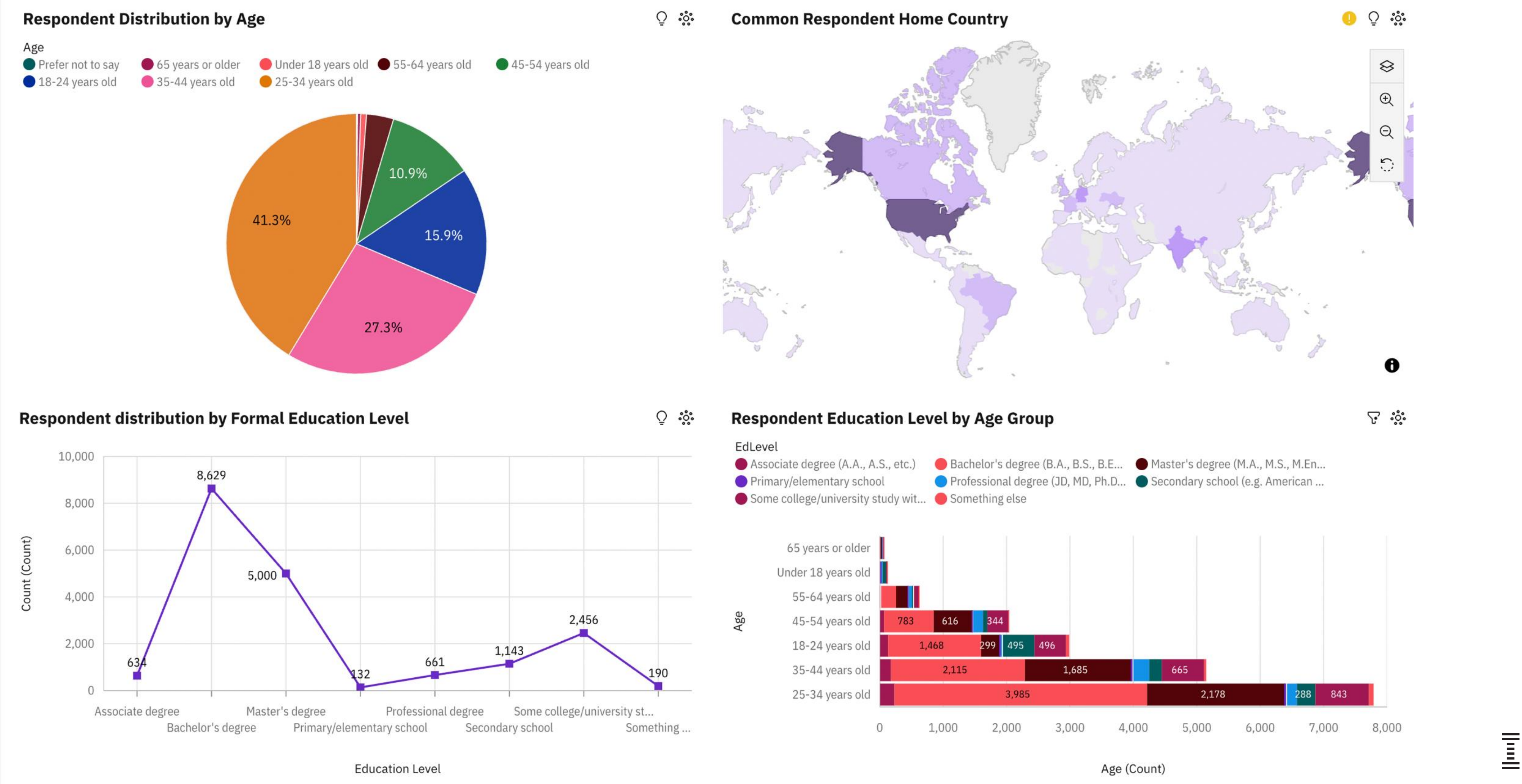
Top 10 Webframes Respondents Want To Work With



Skills

IBM

Dashboard 3: Demographic Trends



Respondent Distribution by Age

Common Respondent Home Country

Respondent distribution by Formal Education Level

Respondent Education Level by Age Group

EdLevel
● Associate degree (A.A., A.S., etc.)
● Bachelor's degree (B.A., B.S., B.E...
● Master's degree (M.A., M.S., M.En...
● Primary/elementary school
● Professional degree (JD, MD, Ph.D...
● Secondary school (e.g. American ...
● Some college/university study wit...
● Something else
Age
65 years or older
Under 18 years old
55-64 years old
45-54 years old
783
616
344
18-24 years old
1,468
299
495
496
35-44 years old
2,115
1,685
665
25-34 years old
3,985
2,178
288
843
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
Age (Count)

Overall Findings & Implications

Overall Findings:

•**Core Technologies Remain Strong:** JavaScript, SQL, and PostgreSQL are consistently top choices in both current use and future preference, indicating their enduring value in the developer ecosystem.

•**Emerging Tools Gaining Momentum:** Languages like Go and Rust, and platforms like Supabase and FastAPI, are gaining interest among developers, suggesting a shift toward lightweight, modern, and cloud-ready solutions.

•**Demographics Reflect a Young, Skilled Workforce:** The majority of respondents are aged 25–34 and hold at least a bachelor's degree, showing that the developer community is largely early-career, educated, and tech-savvy.

Implications:

•**Stabilize Core Stack Support:** Organizations should continue investing in well-established tools that developers trust and want to keep using.

•**Adopt and Integrate Modern Technologies:** Proactively introducing tools like Go and Supabase can improve team productivity and attract forward-thinking talent.

•**Focus on Talent Development and Retention:** With a young, educated developer base, companies should prioritize continuous learning, mentorship, and upskilling programs to stay competitive and retain talent.

Conclusion

This analysis of current and future technology trends, combined with developer demographics, offers valuable insight into the evolving tech landscape.

- **Core technologies** like JavaScript, SQL, and PostgreSQL continue to dominate and should remain a priority in development strategies.

- **Emerging tools** such as Go, Rust, Supabase, and FastAPI represent the next wave of innovation and warrant early adoption and exploration.

- **A young, highly educated developer base** is driving these trends, emphasizing the need for modern tools, flexible workflows, and ongoing learning opportunities.

In summary, balancing investment in foundational technologies with forward-looking innovation will be key to staying competitive and aligned with developer priorities.

Discussion and Extra Visualizations

Slides 16–19 of the Appendix feature 4 additional visualizations developed during the project. These charts were created to highlight key findings from the analysis and provide deeper insights into the data.

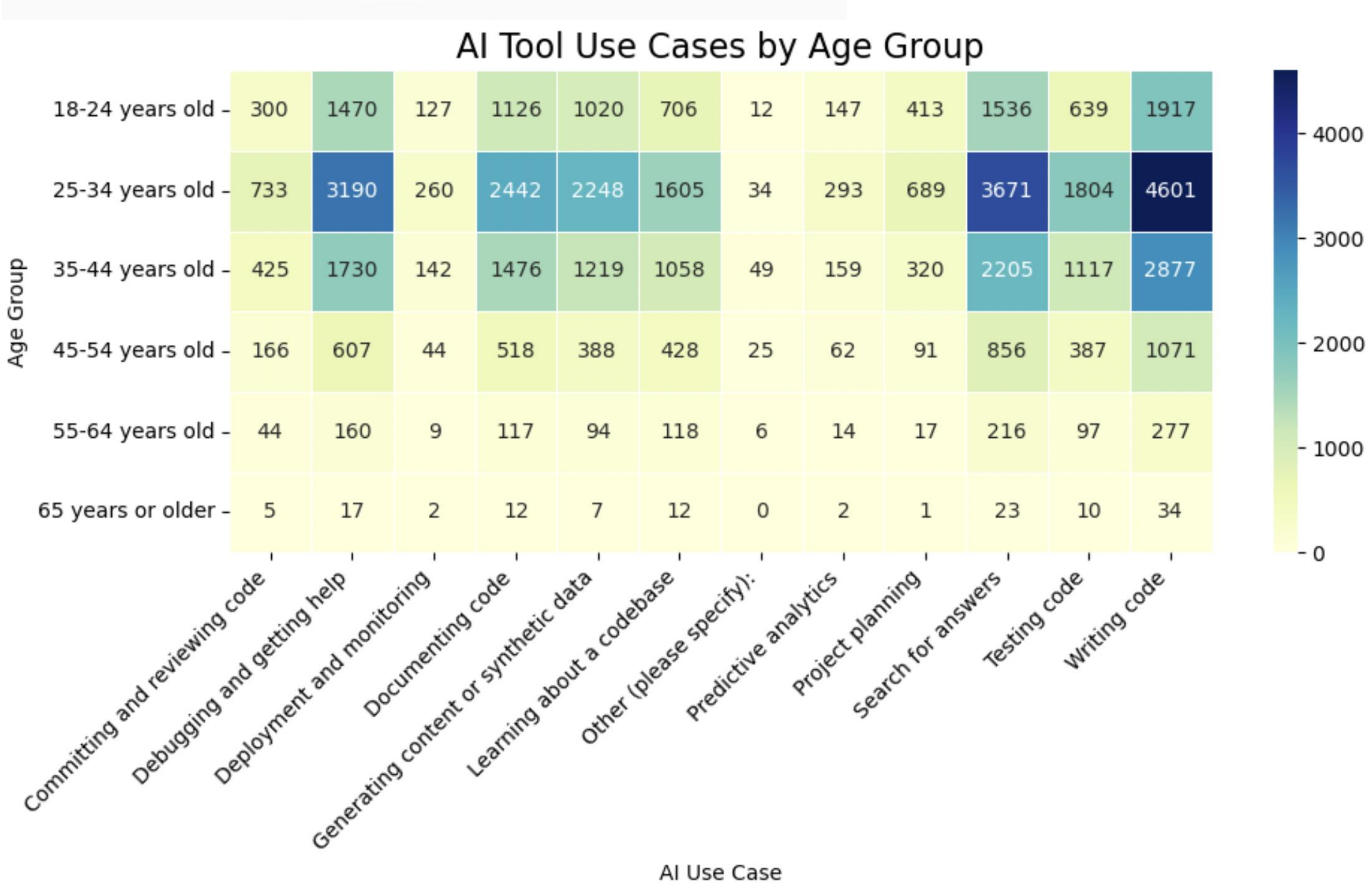
Appendix I: AI Use Cases By Age Group

The heatmap below shows common AI use cases across age groups. **The 24–35 age range stands out as the most active demographic**, with key use cases including:

1. Debugging code
2. Searching for answers
3. Writing code

This supports our analysis by showing:

- **Younger professionals are leading in AI adoption**
- AI is used primarily for technical, skill-based tasks
- Productivity and problem-solving are major drivers of usage



AI Tool Use Cases by Age Group

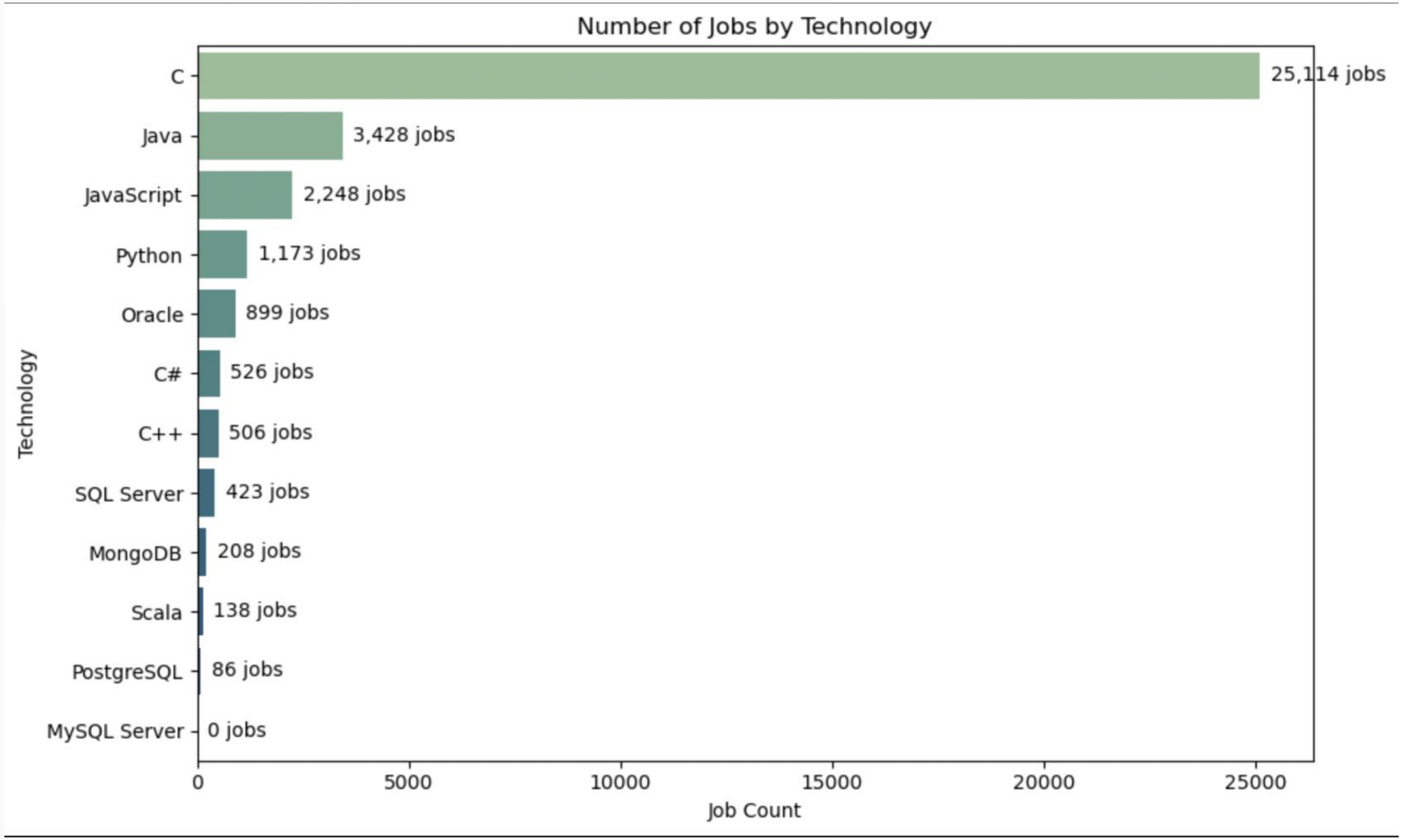
Age Group	Committing and reviewing code	Debugging and getting help	Deployment and monitoring	Documenting code	Generating content or synthetic data	Learning about a codebase	Other (please specify):	Predictive analytics	Project planning	Search for answers	Testing code	Writing code
18-24 years old	300	1470	127	1126	1020	706	12	147	413	1536	639	1917
25-34 years old	733	3190	260	2442	2248	1605	34	293	689	3671	1804	4601
35-44 years old	425	1730	142	1476	1219	1058	49	159	320	2205	1117	2877
45-54 years old	166	607	44	518	388	428	25	62	91	856	387	1071
55-64 years old	44	160	9	117	94	118	6	14	17	216	97	277
65 years or older	5	17	2	12	7	12	0	2	1	23	10	34

Appendix II: Job Postings

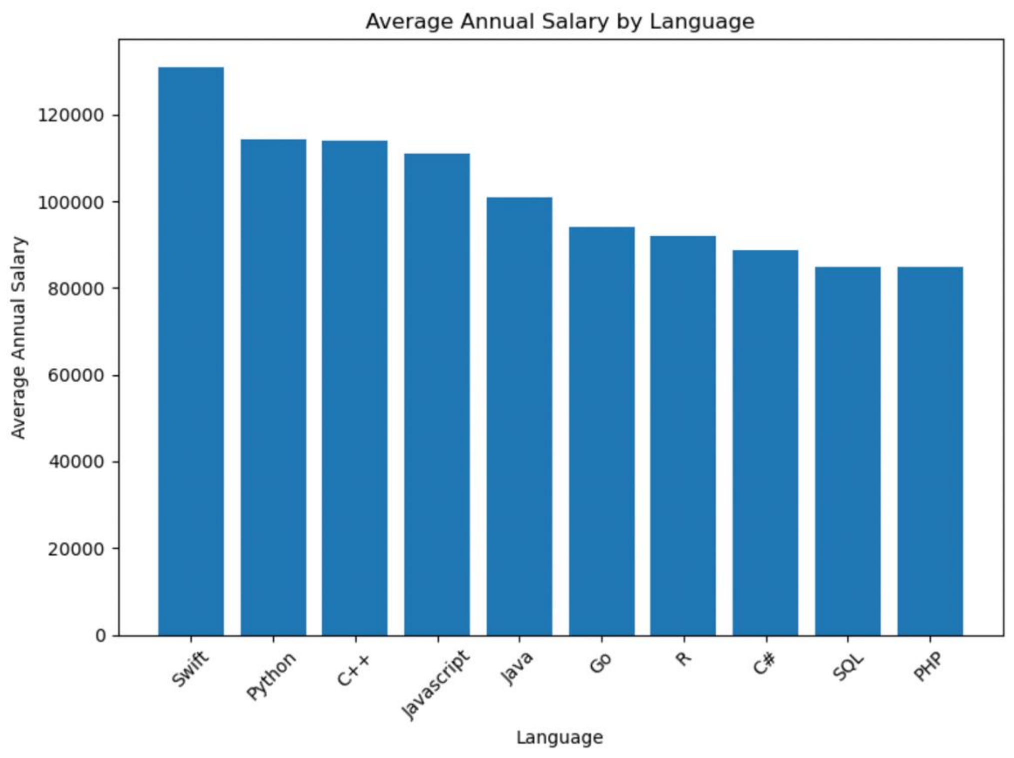
This chart visualizes the number of job postings for various programming languages and technologies.

- **C leads** with over **25,000 listings**,
- followed by **Java** and **JavaScript**, indicating high demand for foundational or legacy languages.
- Modern technologies like **Python** also show strong presence
- tools such as **PostgreSQL**, **Scala**, and **MySQL Server** reflect **lower demand** in the dataset.

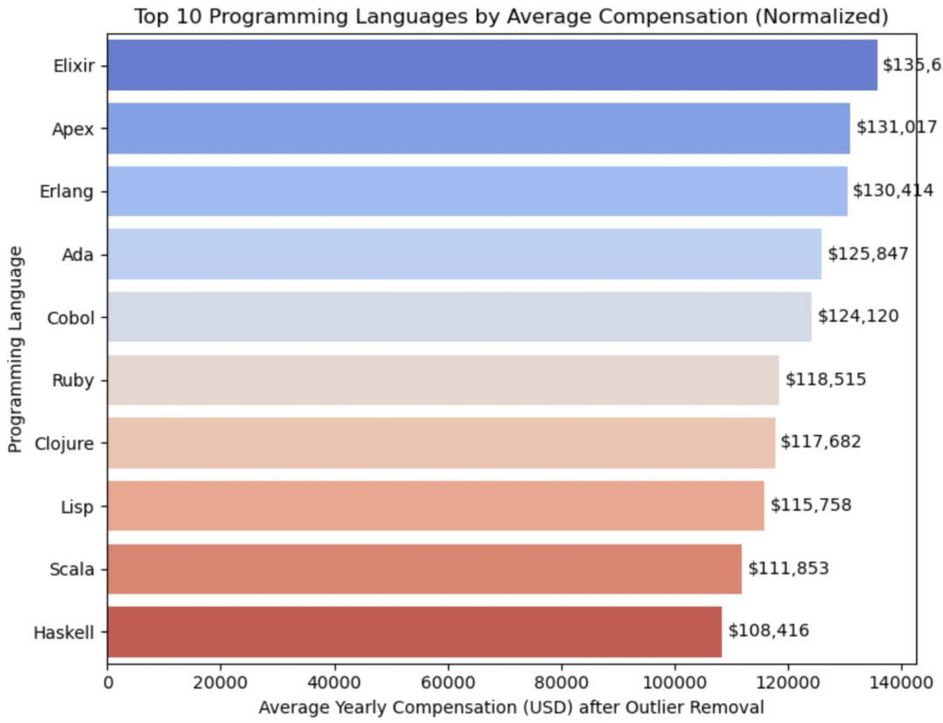
This reinforces our broader project insight that **legacy technologies still dominate job availability**, but this does not always correlate with the **highest salaries** (See analysis on slide 19), which often favor newer, niche languages. It also highlights a potential **mismatch between market saturation and compensation trends**.



Appendix III: Popular Languages Compensation



Building on Appendix II (slide 18), where we analyzed job postings by data language, these visuals illustrate the **average annual compensation** for each language. Graph 1 highlights that mainstream languages such as **Python, JavaScript, and C++ offer strong compensation alongside broad job availability**.



Visual 2 takes the analysis further by presenting **normalized annual compensation** data, allowing for a direct comparison of salaries across different languages and countries. This normalized view reveals that **niche languages like Elixir, Erlang, and Apex command the highest average salaries**. These findings support our conclusion that <u>while traditional core technologies continue to dominate the market, emerging niche languages with higher compensation are creating new opportunities and shaping the future landscape.</u>

Thank You!

Skills Network

IBM