Stack Overflow Survey 2024

IBM Data Analyst Capstone Project

By. Alviantaa

Issued Monday, 2 February 2025

Skills Network

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Current Technology Usage
  - Top 10 Programming Languages
  - Top 10 Databases
  - Top 10 Frameworks
  - Most Popular Platforms
- Future Technology Trends
  - Top 10 desired Programming Languages
  - Top 10 desired Databases
  - Top 10 desired Frameworks
  - Top 10 desired Platforms
- Developer Demographics
  - Age distributions
  - Country
  - Education Level

INTRODUCTION

The Stack Overflow Survey 2024 provides valuable insights into the preferences and trends among developers worldwide.

This report is designed for:

- **Executives -** seeking insights into emerging technologies to align company’s strategies with industry trends.
- **IT Professionals -** Stay ahead of industry trends and upskill in high-demand technologies.
- **Recruiters** – Understanding talent preferences, in-demand skills, and workforce demographics.
- **Educators** – Updated curriculum to match industry needs.

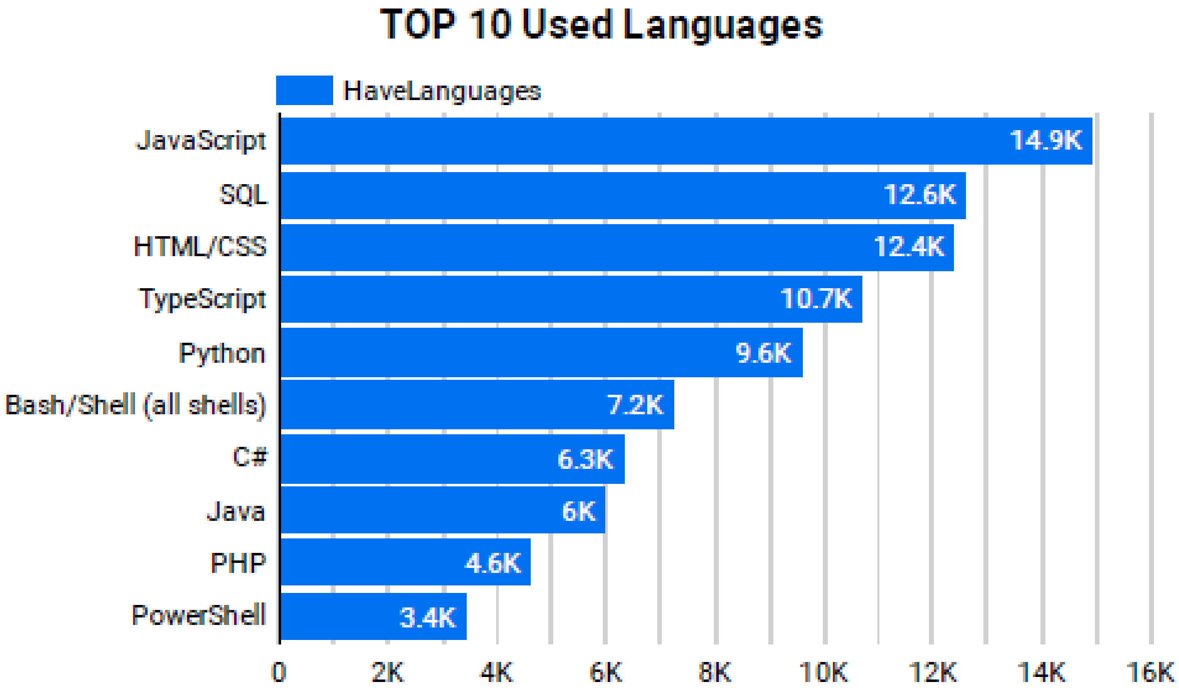
This report highlights key findings in technology usage, desired future trends, and developer demographics.

METHODOLOGY

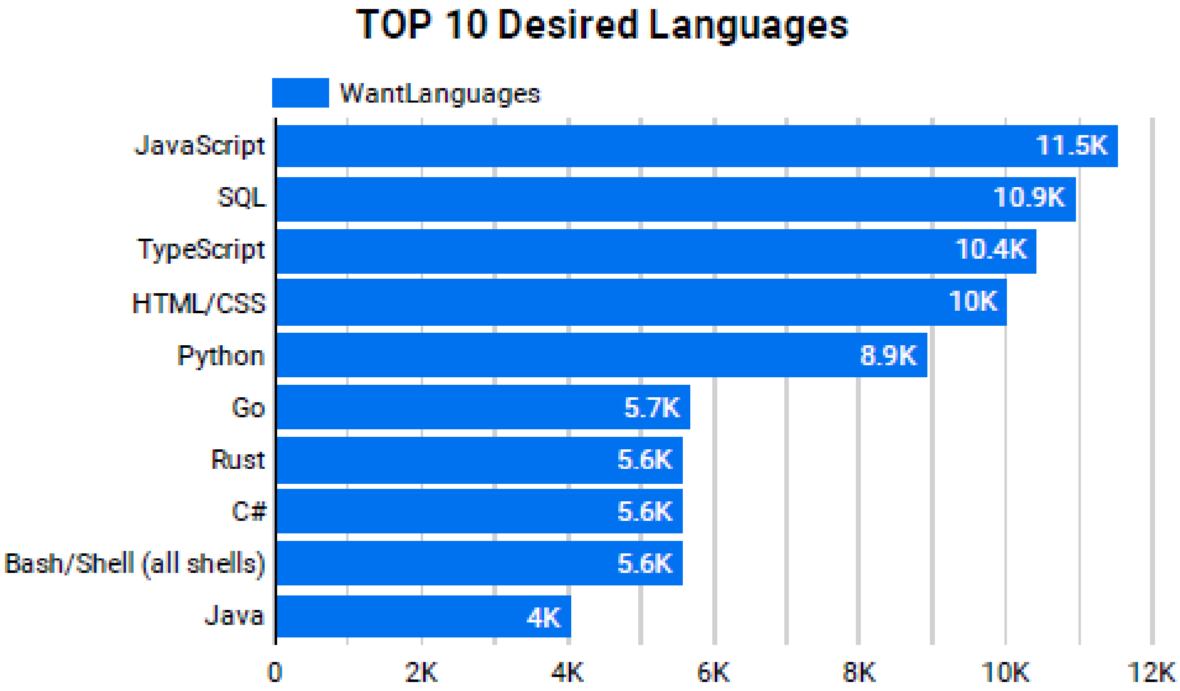
- Data Source
  - A structured dataset containing responses from developers worldwide, capturing their **technology usage, preferences, and demographics**.
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/HLOosvsPgIwt5dgOOh1RSg/survey-data-updated.csv
- **The dataset was downloaded and processed locally using VSCode**, ensuring full control over the data pipeline.
- Data Wrangling
  - Data preparation
  - Normalized Country Names
  - Normalized Current and Future Technology Preferences
- Data Visualizations
- Insights discovery

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

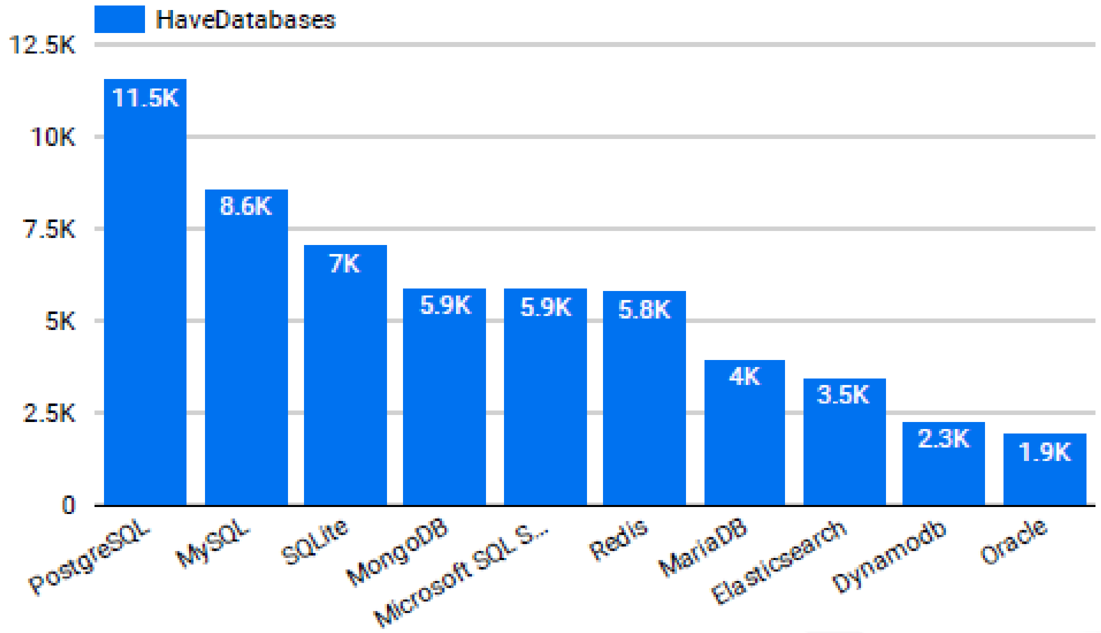
- JavaScript, SQL, and HTML/CSS are the top three.
- On the next year, there is desire to work with JavaScript, SQL, and TypeScript lead
- Python is still on 5th position
- Rust and go are newcomer desired language for the next year

Implications

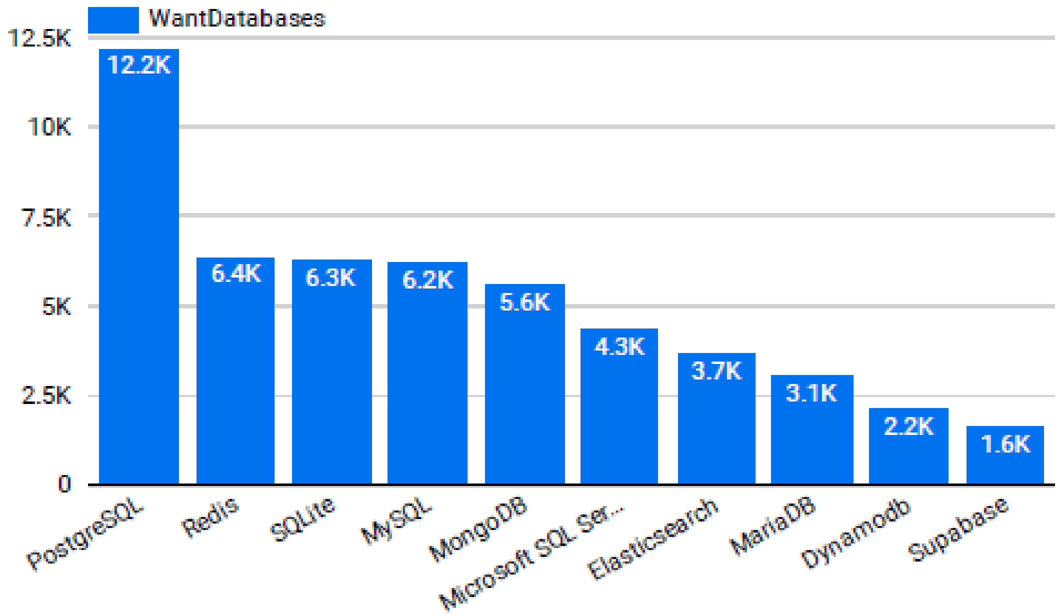
- Strong demand for JavaScript and SQL development.
- Increased desire to TypeScript, indicating a trend towards modern web and database-driven applications.
- Python is still relevant language to work with.
- Go and Rust are increasingly desired, suggesting a shift towards high-performance and cloud-native programming.

DATABASE TRENDS

Current Year


TOP 10 Used Database

Next Year


TOP 10 Desired Database

DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- Current year PostgreSQL, MySQL, and SQLite dominate
- PostgreSQL remains the top choice, while Redis and Supabase are gaining interest.
- MySQL, SQLite, and MongoDB are still taking place for top-5 currently most used and desired database.

Implications

- Reflecting the need for open-source and relational databases.
- PostgreSQL is preferred for RDBMS. Redis and Supabase showing trends toward NoSQL and cloud-native solutions.
- MySQL, SQLite, and MongoDB are still relevant to use, depending on user's preferences.

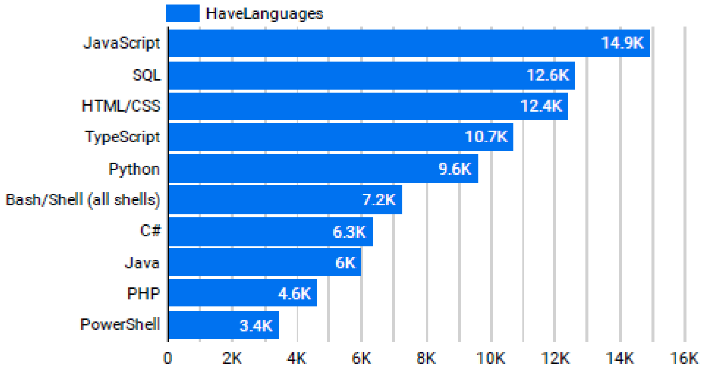
DASHBOARD

Dashboard URL

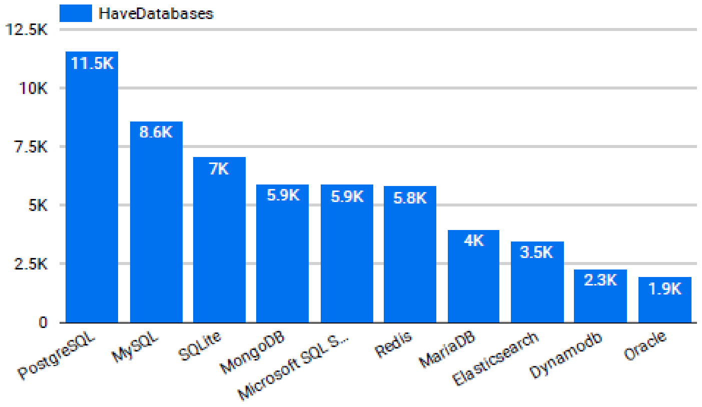
https://lookerstudio.google.com/reporting/c4778164-3495-4b94-88f3-341f5d632ee7

DASHBOARD TAB 1

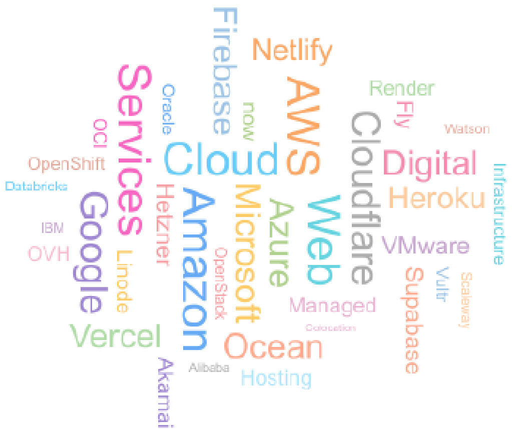
TOP 10 Used Languages



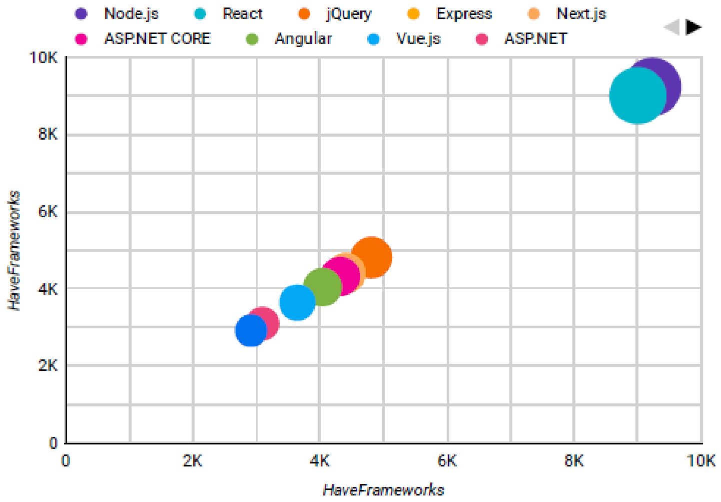
TOP 10 Used Database



Used Platform

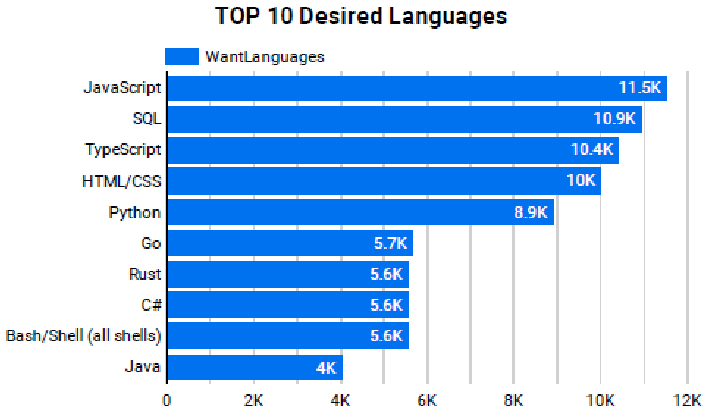


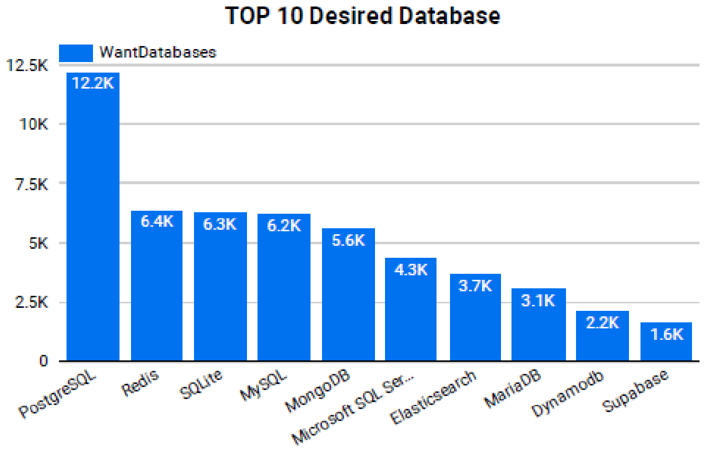
TOP 10 Used Frameworks

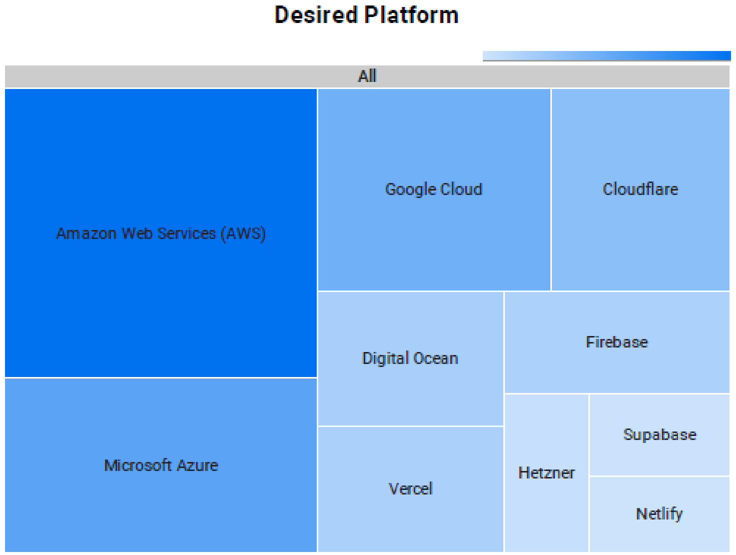


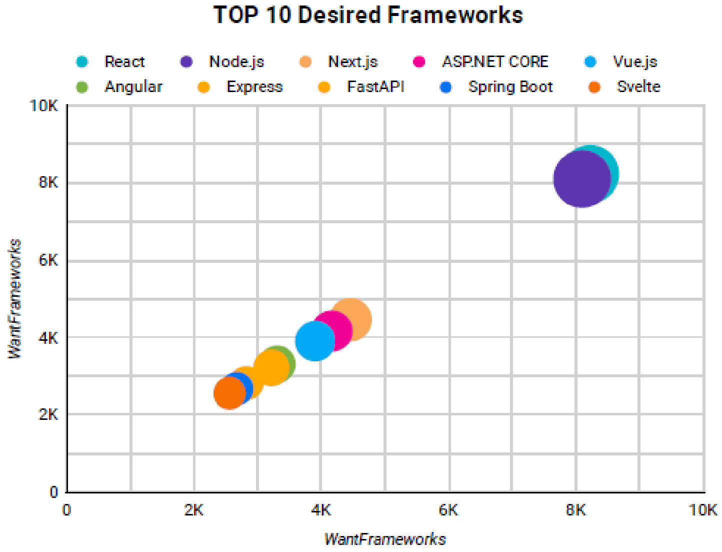
Skills Network

DASHBOARD TAB 2





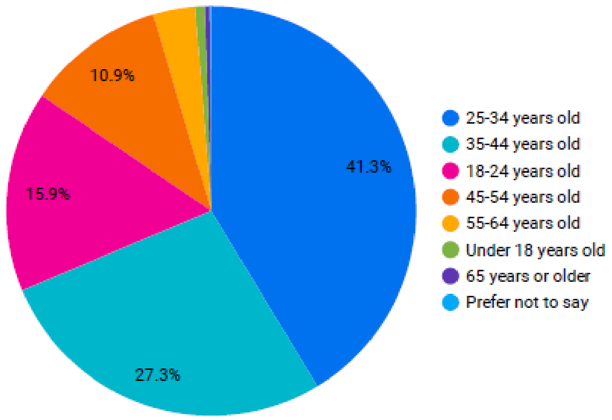




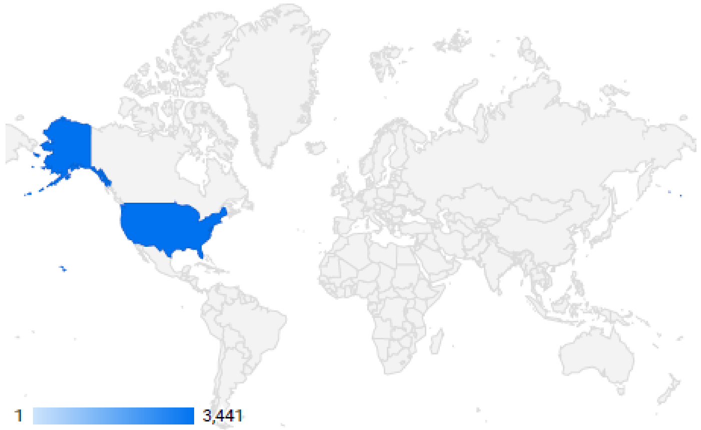
Skills Network

DASHBOARD TAB 3

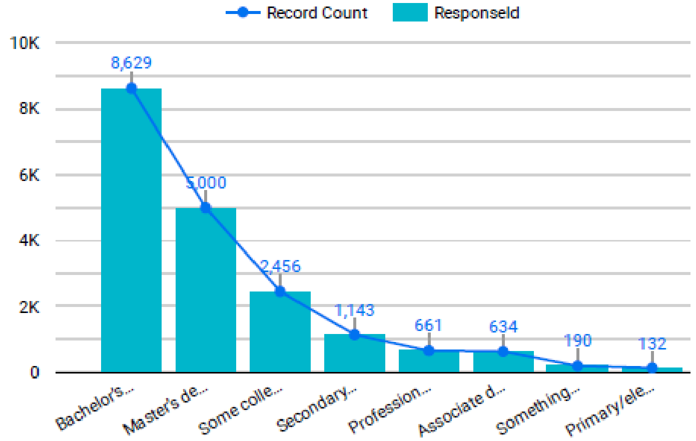
Respondents by Age



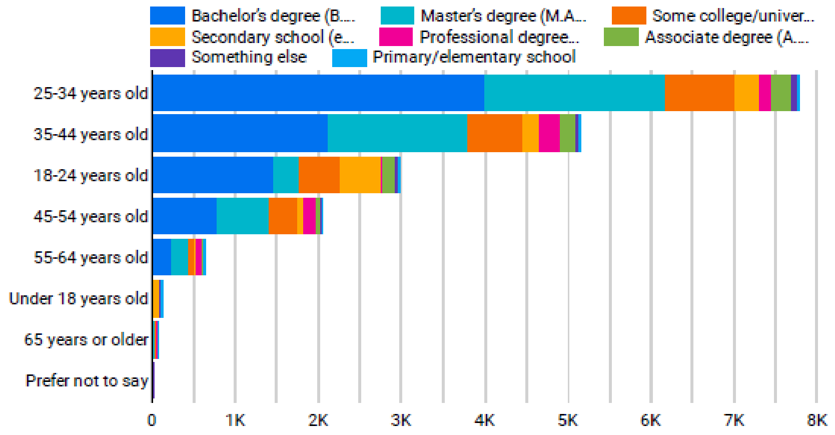
Respondents by Country



Respondents by Education Level



Respondents Age by Education Level



DISCUSSION

- The majority (41.3%) are aged 25-34, followed by 18-24 (27.3%), suggesting a **younger and productive workforce in tech**.
- Most respondents have a **higher education**, primarily a Bachelor’s degree, followed by a Master’s degree. A significant portion has some college or secondary education, highlighting the accessibility of tech careers.
- With slight changes in current technology trends compared to future trends, **high adaptability and continuous learning** are essential to stay competitive.

OVERALL FINDINGS & IMPLICATIONS

Findings

- Current technology trends compared to future trends **are not changing drastically**.
- Web development languages remain in **high demand**.
- Employment in the IT tech industry is primarily held by individuals with **higher education** and within **productive age** groups

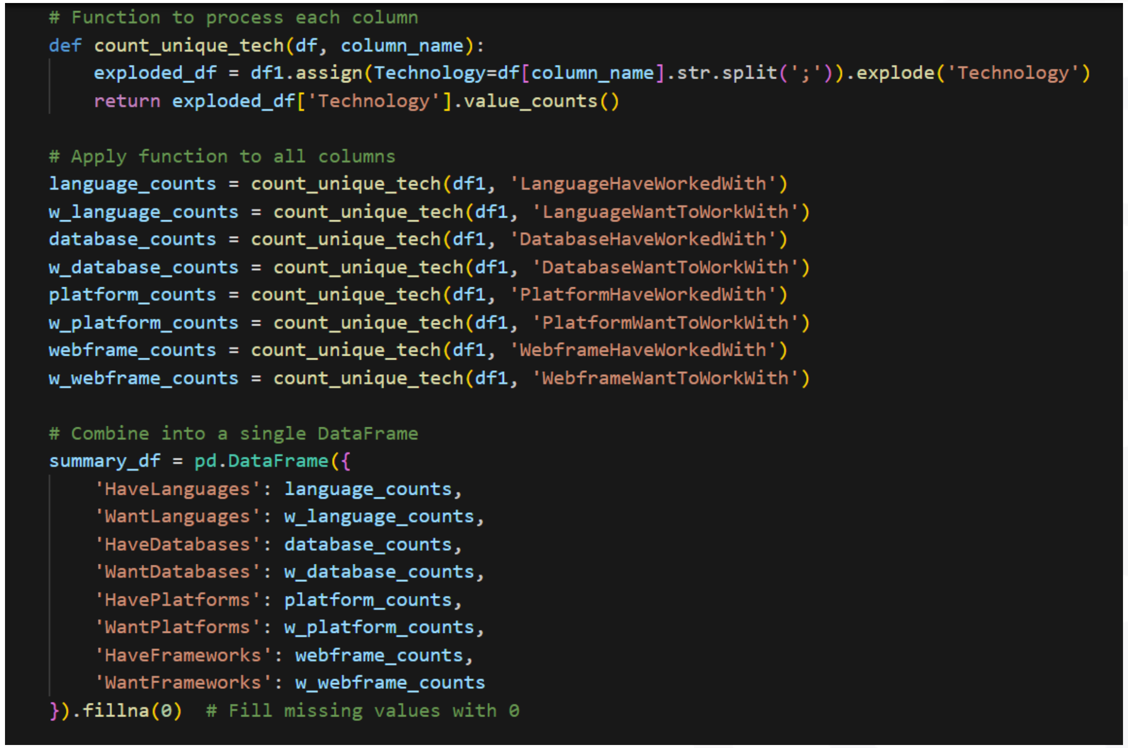
Implications

- Current technology are still relevant to use in the future
- continued growth of web-based applications and the importance of mastering relevant technologies.
- need for continuous learning, skill development, and accessibility of tech education to build credibility, bridge skill gaps, and expand opportunities.

CONCLUSION

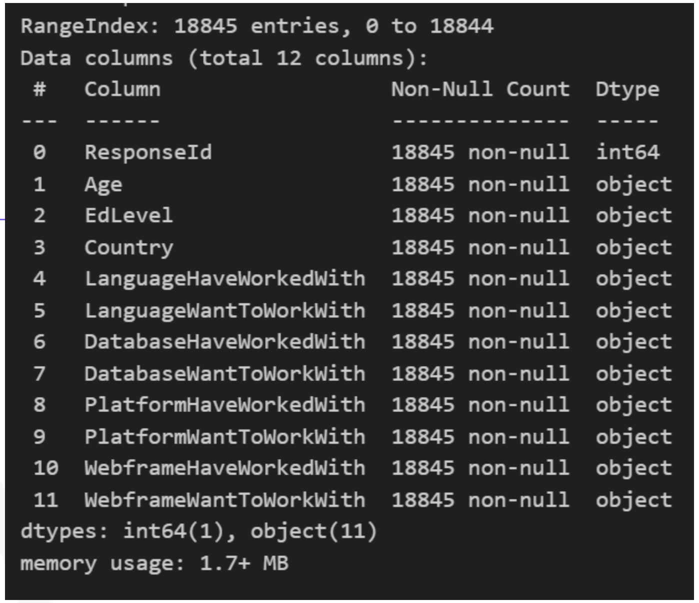
The IT industry continues to evolve, but foundational technologies like **databases, web development, and higher education** remain crucial. To thrive, professionals must **embrace lifelong learning, adapt to gradual changes, and develop versatile skill sets** that align with both **current** and **future industry demands**.

APPENDIX



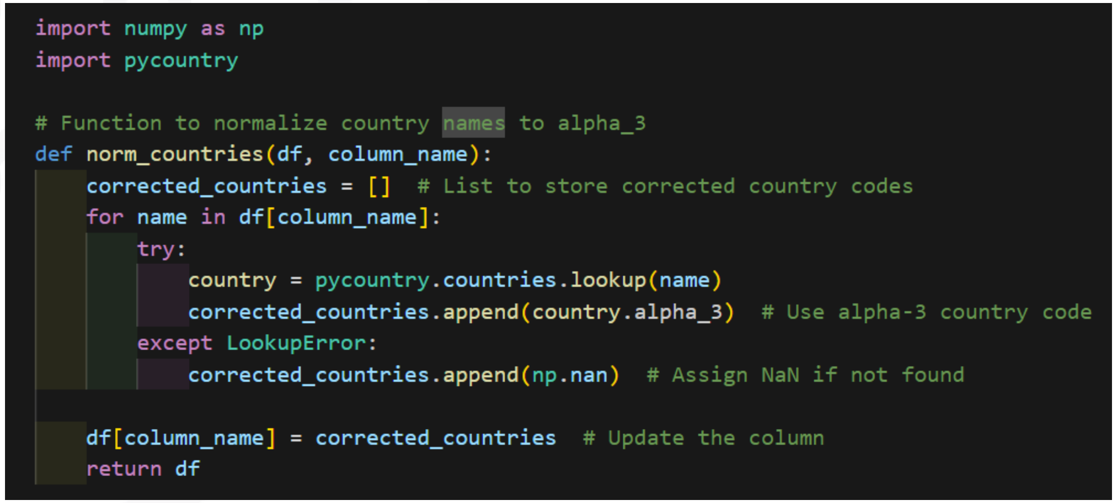
```
# Function to process each column
def count_unique_tech(df, column_name):
    exploded_df = df1.assign(Technology=df[column_name].str.split(';')).explode('Technology')
    return exploded_df['Technology'].value_counts()

# Apply function to all columns
language_counts = count_unique_tech(df1, 'LanguageHaveWorkedWith')
w_language_counts = count_unique_tech(df1, 'LanguageWantToWorkWith')
database_counts = count_unique_tech(df1, 'DatabaseHaveWorkedWith')
w_database_counts = count_unique_tech(df1, 'DatabaseWantToWorkWith')
platform_counts = count_unique_tech(df1, 'PlatformHaveWorkedWith')
w_platform_counts = count_unique_tech(df1, 'PlatformWantToWorkWith')
webframe_counts = count_unique_tech(df1, 'WebframeHaveWorkedWith')
w_webframe_counts = count_unique_tech(df1, 'WebframeWantToWorkWith')

# Combine into a single DataFrame
summary_df = pd.DataFrame({
    'HaveLanguages': language_counts,
    'WantLanguages': w_language_counts,
    'HaveDatabases': database_counts,
    'WantDatabases': w_database_counts,
    'HavePlatforms': platform_counts,
    'WantPlatforms': w_platform_counts,
    'HaveFrameworks': webframe_counts,
    'WantFrameworks': w_webframe_counts
}).fillna(0)  # Fill missing values with 0
```

Normalizing technologies
(Languages, platforms, databases, frameworks)
Then dataframe will be exported as CSV



```
RangeIndex: 18845 entries, 0 to 18844
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   ResponseId              18845 non-null  int64 
 1   Age                     18845 non-null  object
 2   EdLevel                 18845 non-null  object
 3   Country                 18845 non-null  object
 4   LanguageHaveWorkedWith  18845 non-null  object
 5   LanguageWantToWorkWith  18845 non-null  object
 6   DatabaseHaveWorkedWith  18845 non-null  object
 7   DatabaseWantToWorkWith  18845 non-null  object
 8   PlatformHaveWorkedWith  18845 non-null  object
 9   PlatformWantToWorkWith  18845 non-null  object
 10  WebframeHaveWorkedWith  18845 non-null  object
 11  WebframeWantToWorkWith  18845 non-null  object
dtypes: int64(1), object(11)
memory usage: 1.7+ MB
```

Dataframe for needed columns only

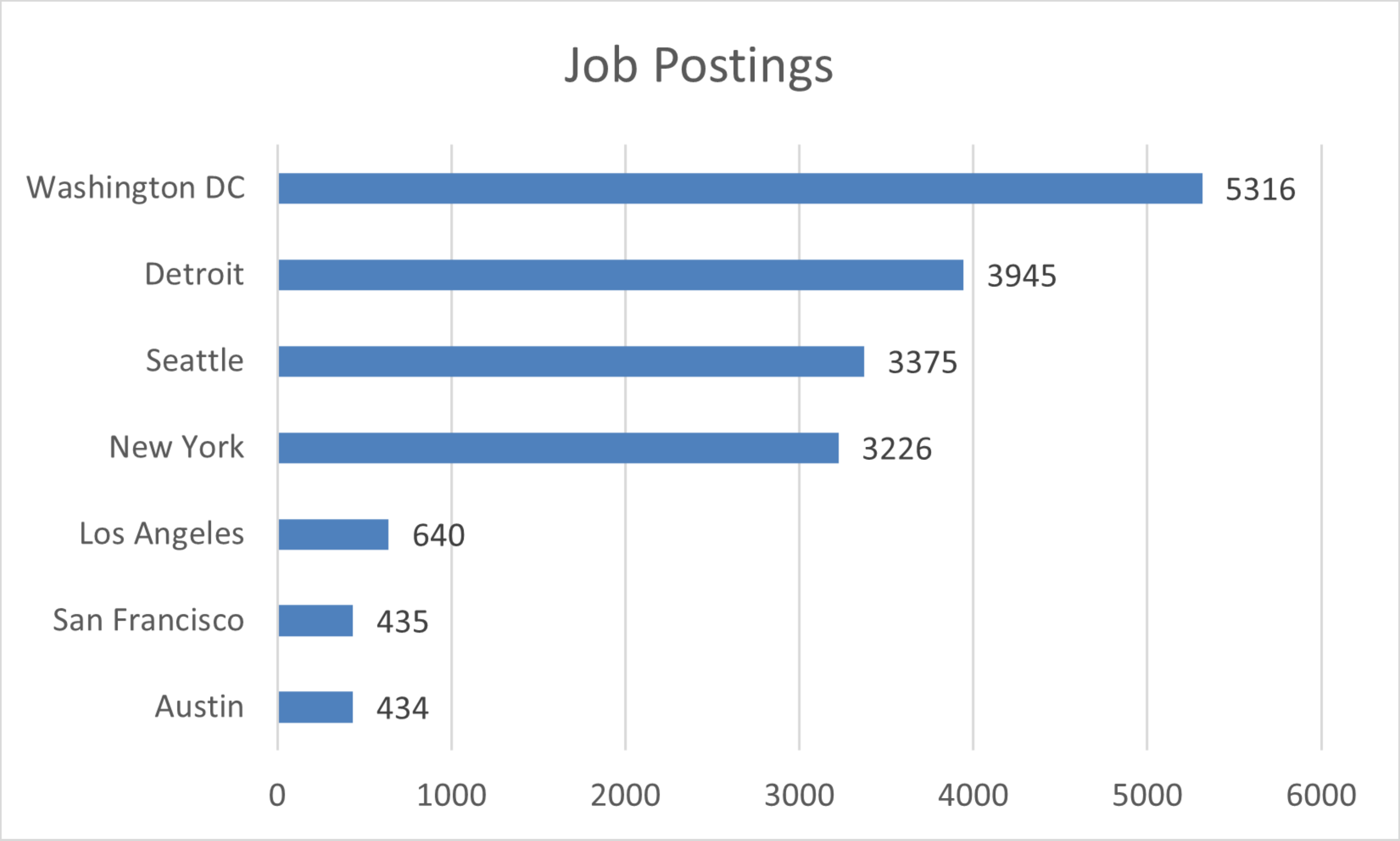


```
import numpy as np
import pycountry

# Function to normalize country names to alpha_3
def norm_countries(df, column_name):
    corrected_countries = []  # List to store corrected country codes
    for name in df[column_name]:
        try:
            country = pycountry.countries.lookup(name)
            corrected_countries.append(country.alpha_3)  # Use alpha-3 country code
        except LookupError:
            corrected_countries.append(np.nan)  # Assign NaN if not found

    df[column_name] = corrected_countries  # Update the column
    return df
```

Normalizing country name function

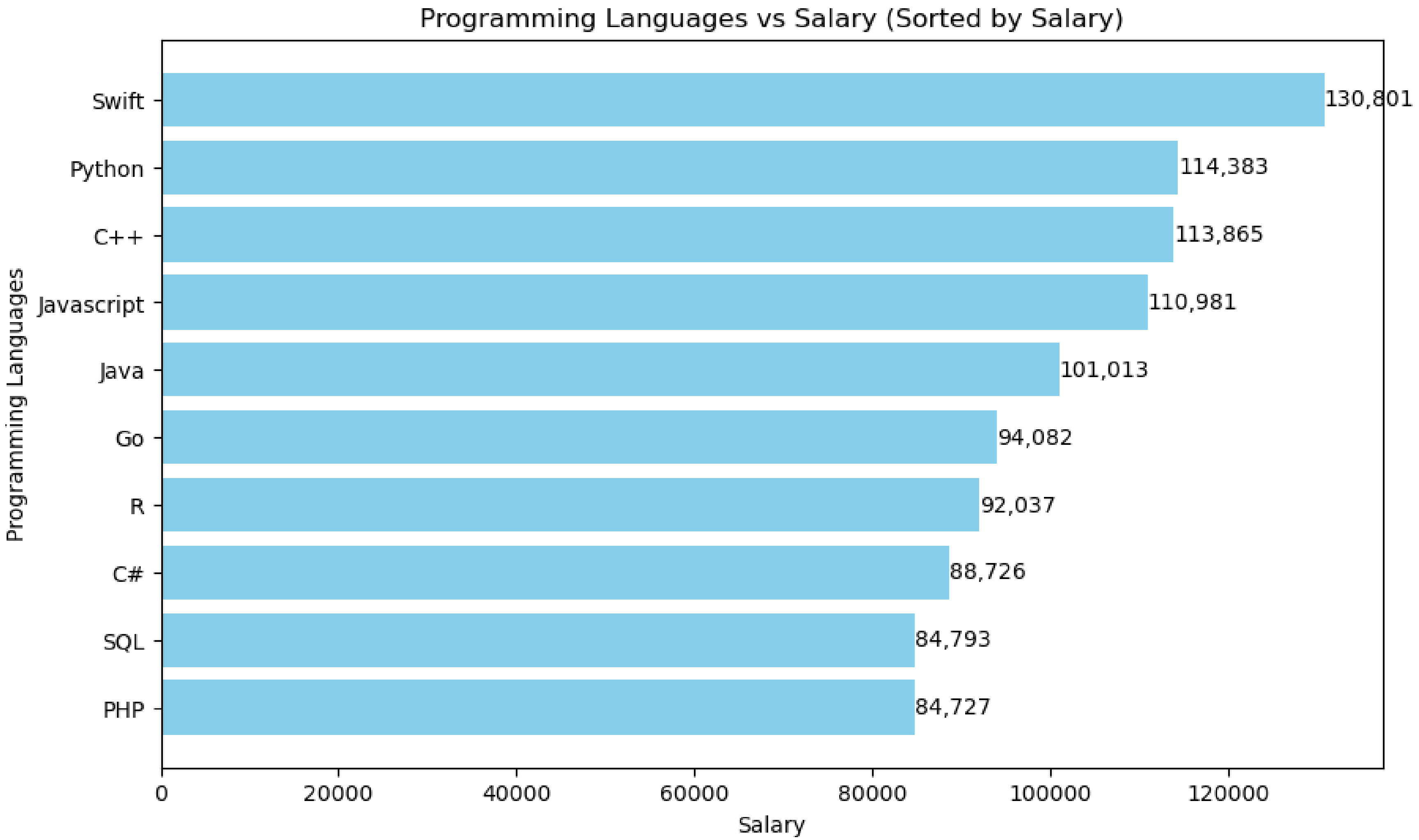
JOB POSTINGS

Module 1 – Job Postings API



POPULAR LANGUAGES

Module 1 - popular-languages.csv


Programming Languages vs Salary (Sorted by Salary)
Programming Languages
Swift
Python
C++
Javascript
Java
Go
R
C#
SQL
PHP
Salary