

IBM Data Analyst Capstone Project:

Stack Overflow Developer Survey Analysis

Dmitrii Kokhanov
30 August 2025



© IBM Corporation. All rights reserved.

OUTLINE



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

EXECUTIVE SUMMARY



This project analyzes the **Stack Overflow Developer Survey** to uncover insights into technology usage and future adoption trends among developers worldwide.

- **Programming Languages:** JavaScript leads current usage, followed by SQL, HTML/CSS, and Python. Looking forward, Python, Go, and Rust show strong growth in developer interest, signaling a shift toward versatile and performance-oriented languages.
- **Databases:** PostgreSQL is the most widely used and remains the top desired database, while Redis emerges as a rising preference, reflecting increased demand for real-time, in-memory solutions alongside traditional relational systems.
- **Platforms:** Amazon Web Services (AWS) dominates as the leading platform, with Microsoft Azure and Google Cloud also widely used. Future interest expands to Cloudflare, Vercel, and Supabase, highlighting momentum toward serverless and edge-computing platforms.
- **Frameworks:** React and Node.js dominate both current usage and future demand, with growing interest in Next.js and Vue.js. This reflects the continued strength of the JavaScript ecosystem and the evolution toward modern, modular frameworks.

Overall, the analysis reveals a developer community balancing reliance on proven, established technologies with growing enthusiasm for innovative, cloud-native, and high-performance tools. These insights highlight where the industry is today and the direction it is rapidly moving.

INTRODUCTION



This analysis explores global technology adoption and workforce demographics using **Stack Overflow Developer Survey dataset** from respondents worldwide. The study addresses the challenge of understanding how age, education, and professional experience influence technology preferences and career outcomes. By examining key trends such as age distribution, education levels, and platform or language popularity, the analysis seeks to answer several guiding questions:

- Which programming languages are most in demand?
- Which database technologies are currently most sought after?
- Which Web frames are the most popular?

The findings aim to provide insights for educators, employers, and policymakers on shaping training programs, workforce planning, and technology strategies in response to evolving global trends.

METHODOLOGY



- Collect survey data (the latest Stack Overflow Developer Survey)
 - Web scraping
 - APIs
 - Request library
- Data wrangling
 - Data cleaning, finding and removing duplicates, incorrect or irrelevant data
- Exploratory Data Analysis (EDA)
 - Distribution
 - Outlier Handling
 - Correlation (finding key influencers)
- Visualization
 - Distribution of data
 - Relationship
 - Comparison
 - Composition with stacked bar charts
- Dashboards
 - Bar, column, word cloud, tree map, hierarchy bubble, pie, world map, line and stacked bar charts



RESULTS

Current Technology Usage and Future Technology Trends of:

Programming languages
Databases
Platforms
Web Frames

Demographics:

Age
Education
Region

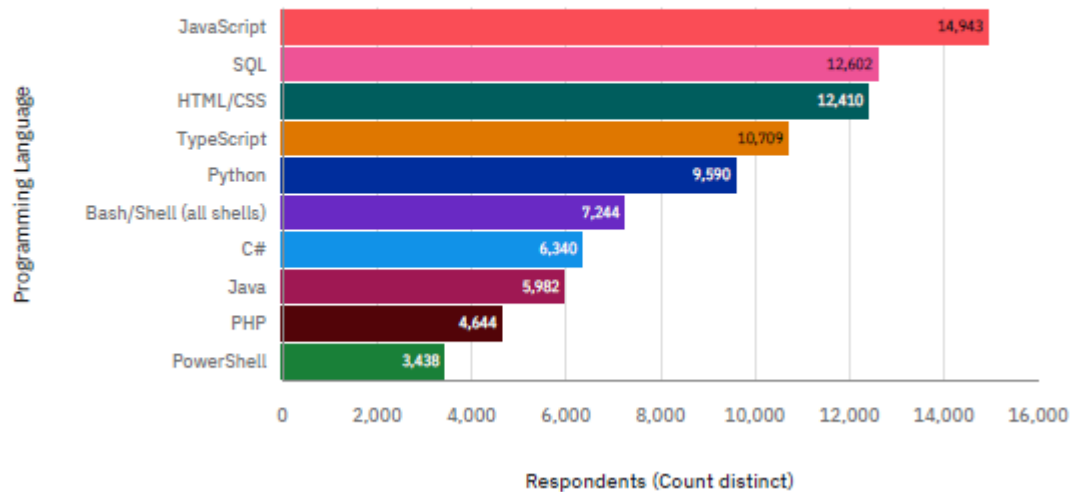


PROGRAMMING LANGUAGE TRENDS

Current Year

Current Technology Usage

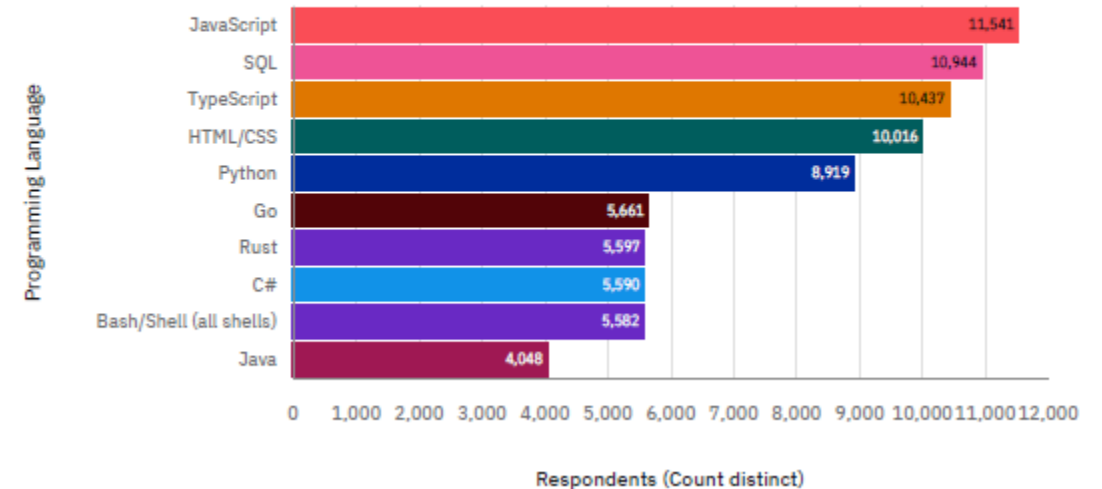
Top 10 Most Popular Languages



Next Year

Future Technology Trend

Top 10 Most Desired Programming Languages



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

- JavaScript is by far the most popular programming language, with SQL close behind.
- TypeScript will take third place, ahead of HTML/CSS.
- While Python is holding steady at fifth place, Go and Rust are quickly moving ahead, ahead of Bash and C#.

Implications

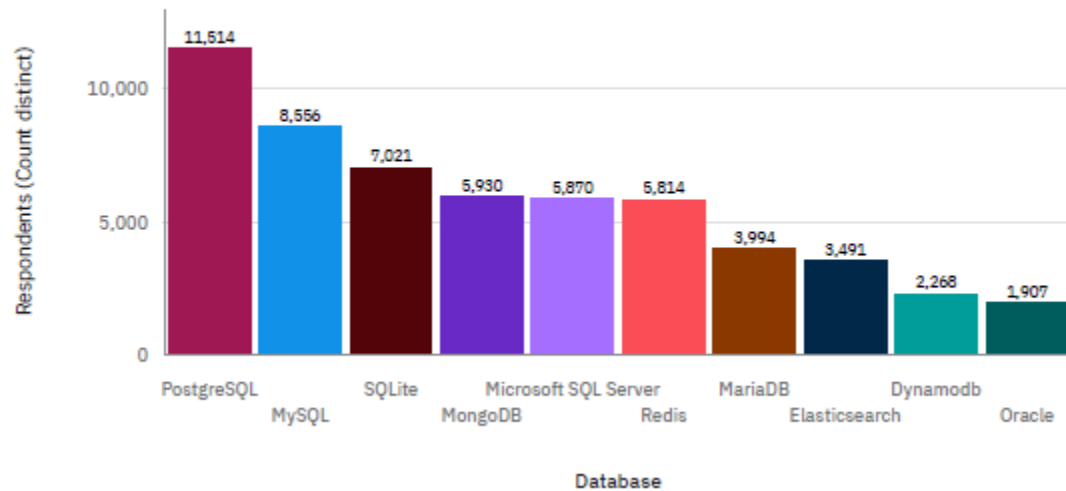
- Dominance of JavaScript and SQL confirms that web development, application development, and even more data management continue to be central in the industry. For organizations, this reinforces the need to maintain strong support for these technologies in hiring and training.
- The rise of TypeScript reflects a move toward safer, large-scale development: As TypeScript overtakes HTML/CSS, it highlights a growing preference for type safety, scalability, and maintainability in enterprise-grade applications. Developers who invest in TypeScript skills are well-positioned for future demand.
- Python remains a top choice for data science, AI, and automation. This indicates stability in its core domains, making it a reliable skill for long-term career development.
- Go and Rust are on the rise: Their advancement over Bash and C# signals a developer community increasingly interested in performance, concurrency, and memory safety. Organizations building cloud-native, systems-level, and high-performance applications should anticipate stronger demand for these languages.



DATABASE TRENDS

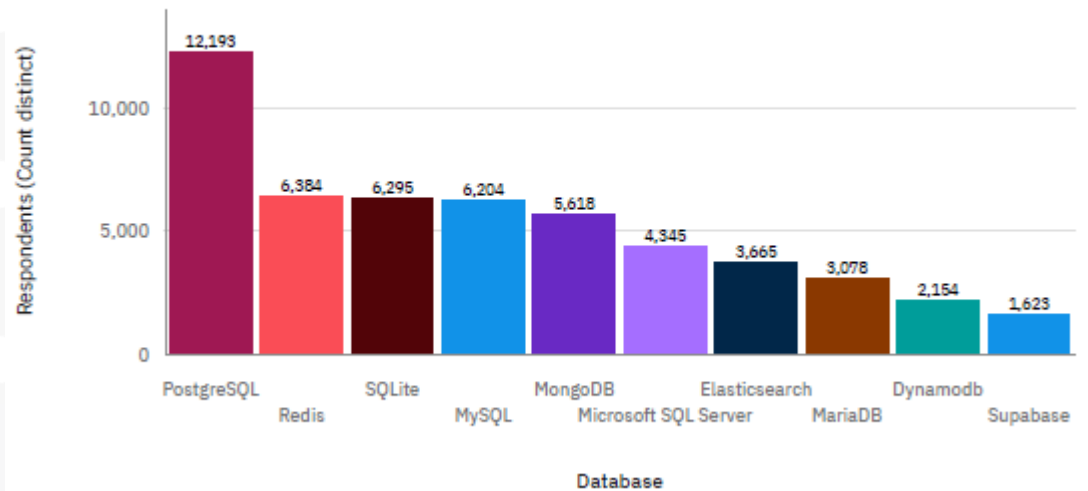
Current Year

Top 10 Most Popular Databases



Next Year

Top 10 Most Desired Databases



DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

- PostgreSQL holds its first place far above others with growing.
- MySQL lost its second place to Redis, which jumped from 6th to 2nd place also moving MongoDB and Microsoft SQL Server one step back.
- SQLite stays on the third place, while Oracle is not in Top 10 anymore.

Implications

- PostgreSQL has become the default choice for many developers and organizations, reinforcing its role as the go-to open-source relational database for scalability and reliability.
- Redis's surge highlights the growing demand for speed and real-time data processing, especially in cloud-native and high-performance applications.
- MongoDB and Microsoft SQL Server shift reflects increasing competition in the database space, with newer technologies reshaping the balance between traditional relational systems and modern NoSQL solutions.
- SQLite's consistency demonstrates the enduring importance of lightweight, embedded databases for mobile, local, and small-scale applications.
- The decline of Oracle signals that proprietary, legacy databases are losing developer mindshare, as the industry continues to favor open-source and cloud-friendly alternatives.



DASHBOARD



The GitHub links for dashboards of:

IBM Cognos Analytics

https://github.com/Witharmsoutstretched/IBM_Data_Analyst_Professional_Certificate/blob/main/Stack_Overflow_survey_Dashboard_Letter_Landscape.pdf

Google Looker Studio

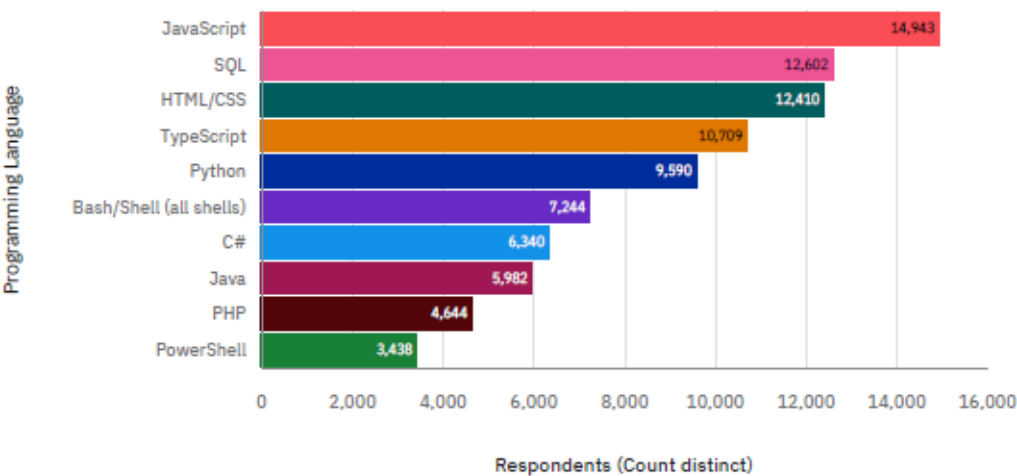
https://github.com/Witharmsoutstretched/IBM_Data_Analyst_Professional_Certificate/blob/main/Stack_Overflow_survey_Dashboard_GLS.pdf



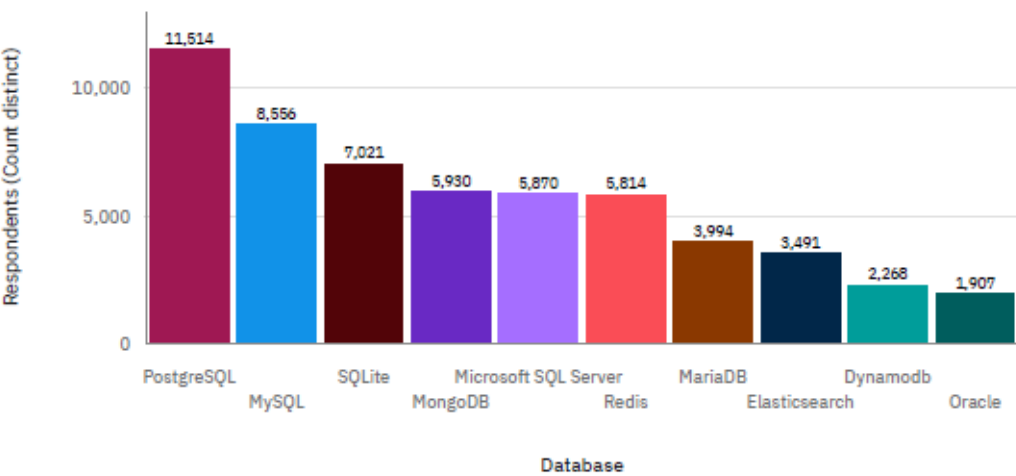
DASHBOARD TAB 1

Current Technology Usage

Top 10 Most Popular Languages



Top 10 Most Popular Databases



Top 10 Most Popular Platforms



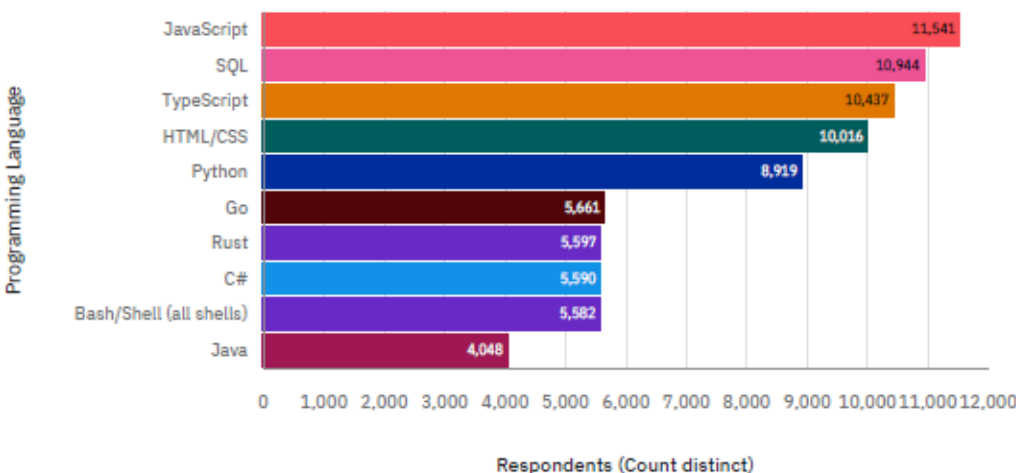
Top 10 Most Popular Web Frames



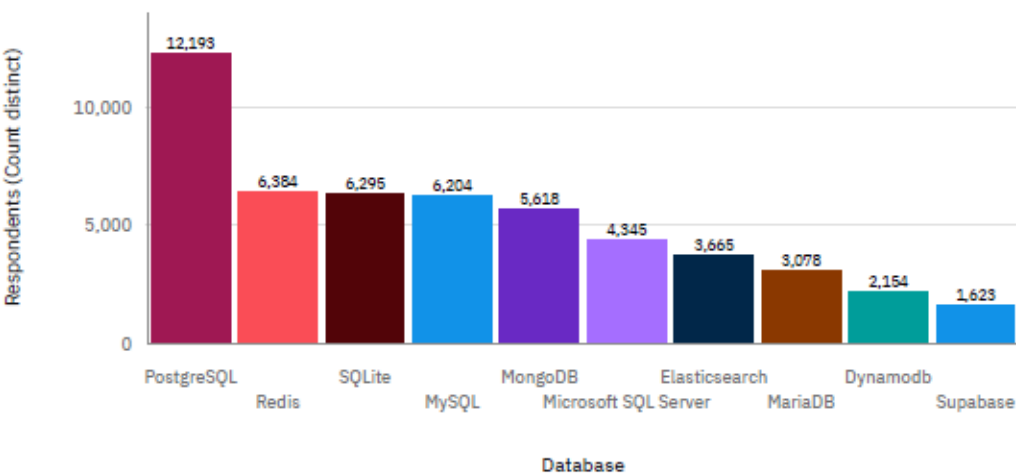
DASHBOARD TAB 2

Future Technology Trend

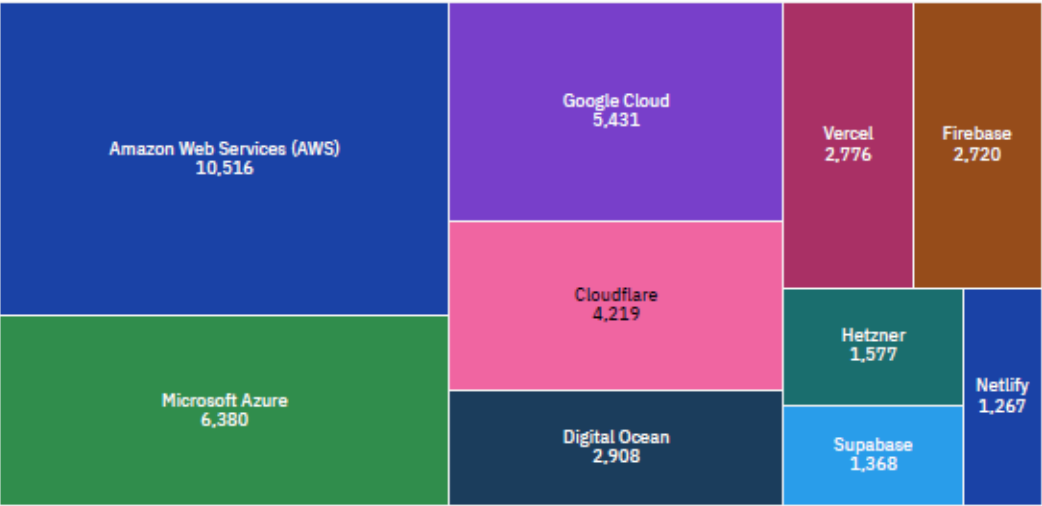
Top 10 Most Desired Programming Languages



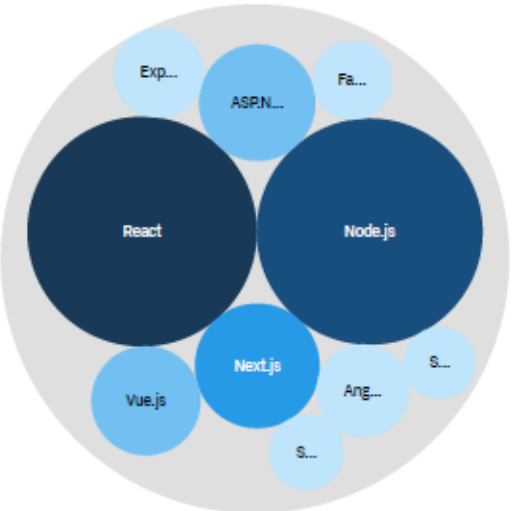
Top 10 Most Desired Databases



Top 10 Most Desired Platforms



Top 10 Most Desired Web Frames

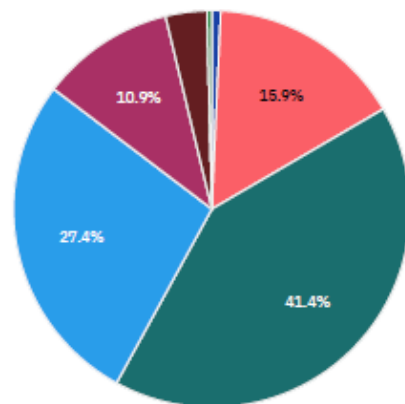


DASHBOARD TAB 3

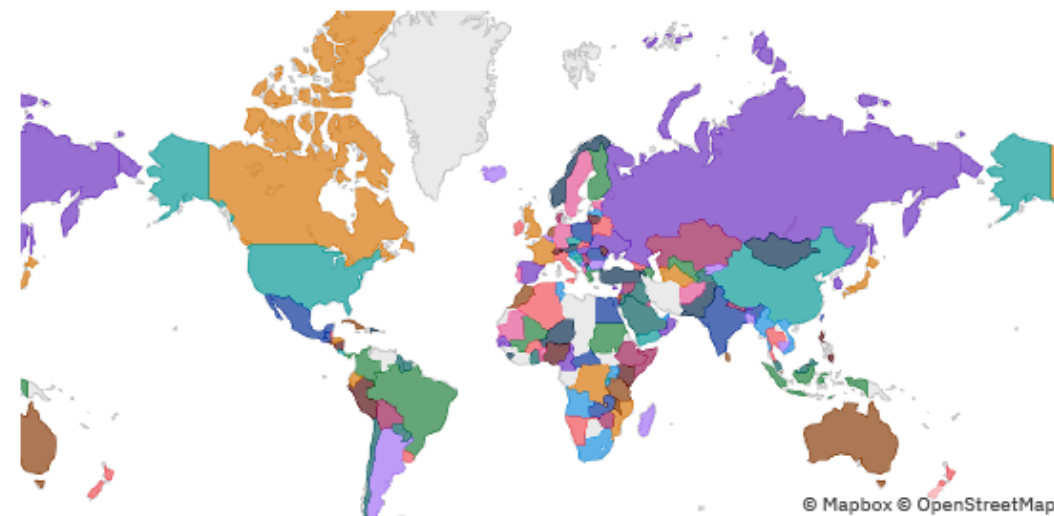
Demographics

Respondent Distribution by Age Groups

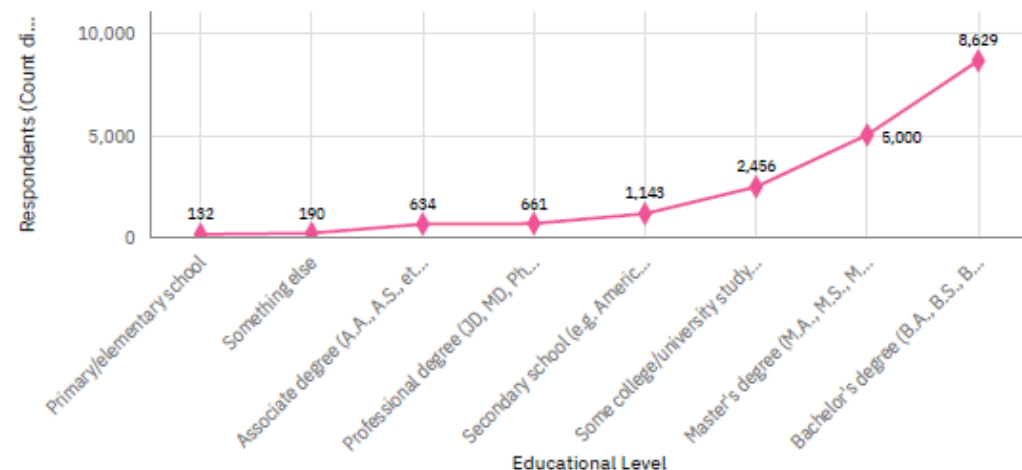
Under 18 years old 18-24 years old 25-34 years old 35-44 years old 45-54 years old
55-64 years old 65 years or older



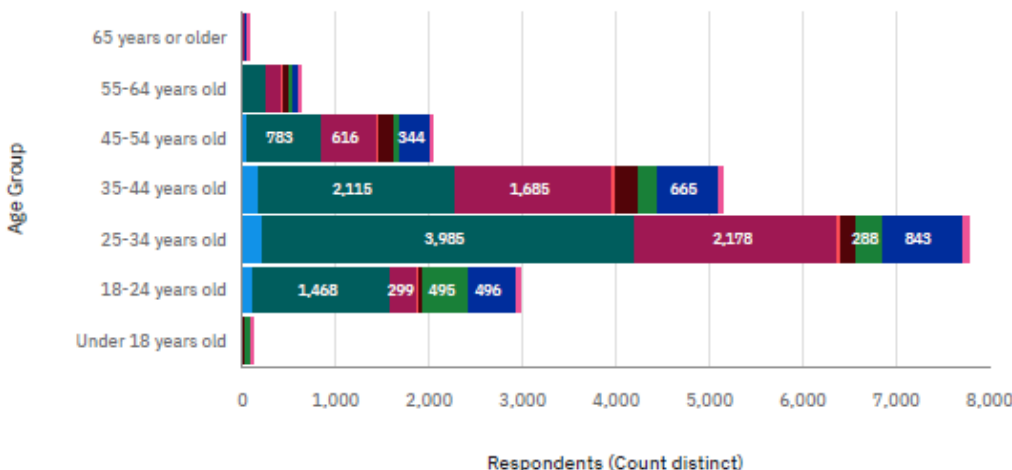
Respondent Count by Country



Respondent Distribution by Formal Education Level



Respondent Count by Age, Classified by Education Level

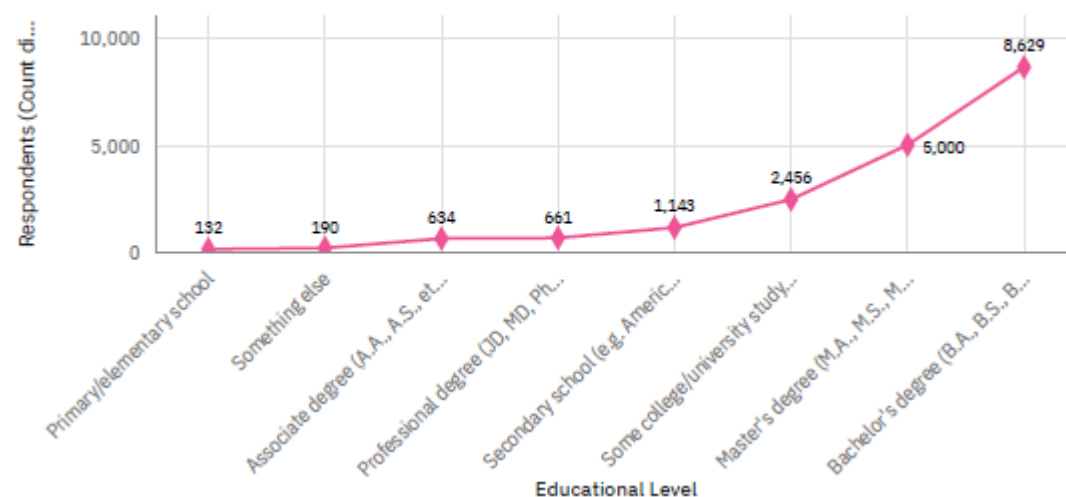


DISCUSSION

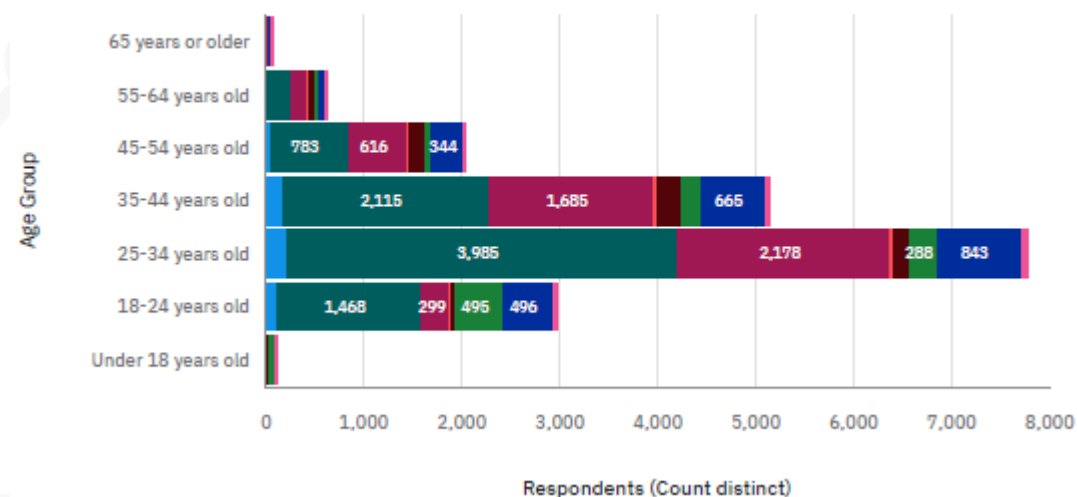


- ❖ Comparing composition of educational levels among developers' age groups shows that demand in master's degree have tendency in growing significantly up to age group of 35-44, although needs further observations and surveys for analysis.

Respondent Distribution by Formal Education Level



Respondent Count by Age, Classified by Education Level



OVERALL FINDINGS & IMPLICATIONS

Findings

- Programming languages:

JavaScript, SQL and Python stay in the same demand. **TypeScript, Go and Rust** become popular quickly.

- Databases:

PostgreSQL and SQLite hold still while **Redis** jumped from the 6-th to the 2-nd place. **Supabase** appeared in the last place of the Top 10 list instead of Oracle.

- Platforms:

Amazon Web Services (AWS) clearly dominates, followed by Microsoft Azure and Google Cloud. **Cloudflare** gaining traction, along with newer players like **Vercel** and **Supabase**.

- Frameworks:

React and Node.js are the most popular frameworks, followed by Express, Angular. **Next.js** and **Vue.js** also attract strong developer interest.

See the **Appendix** for the next findings:

- Most respondents are:

- from developed countries
- have a Bachelors' degree
- of age 24-40 with median age 31

- Master's degree ratio increases up to ages 35-44.

- Compensation are growing steadily with experience up to 35 years, then varies.

- Job satisfaction almost don't correlate with compensation, but tools the respondents are working with.



OVERALL FINDINGS & IMPLICATIONS

Implications

- **Programming Languages**

Demand for JavaScript, SQL, and Python remains stable, but rapid growth of **TypeScript**, **Go**, and **Rust** suggests shifts in hiring priorities and the need for developers to adapt quickly.

- **Databases**

PostgreSQL and SQLite remain strong, but **Redis** jump to 2nd place and **Supabase** entering the Top 10 reflect rising interest in high-performance and developer-friendly solutions, while Oracle's decline shows reduced reliance on legacy systems.

- **Platforms**

AWS dominance continues, yet growth of Azure, Google Cloud, **Cloudflare**, **Vercel**, and **Supabase** highlights intensifying cloud competition and diversification of deployment environments.

- **Frameworks**

React, Node.js, and Express remain core, but strong traction of Angular, **Next.js**, and **Vue.js** indicates teams increasingly value flexibility and modern front-end experiences.

See the **Appendix** for the next implications:

Demographics & Workforce

- The majority of respondents from **developed countries** suggests that survey insights may **over-represent trends from mature tech markets**, which can influence global technology directions.
- The dominance of **Bachelor's degree holders** and the increase in **Master's degrees by mid-career** underscores the value of **formal education in advancing tech careers**, particularly for leadership and specialized roles.
- The **age distribution (24–40, median 31)** highlights that the **developer workforce is concentrated in early- to mid-career stages**, shaping the kinds of technologies in demand and the pace of adoption.
- **Compensation growth up to 35 years of experience** indicates that **career earnings peak mid-career**, after which senior professionals may prioritize flexibility, specialization, or leadership roles.
- The weak correlation between **job satisfaction and compensation** but stronger ties to **tools and technologies used** implies that **engagement, productivity, and retention depend heavily on access to modern, enjoyable tools rather than salary alone**.



CONCLUSION



- Developers and organizations alike must stay adaptable to remain competitive.
- Formal education matters.
- Companies should care more about job satisfaction factors besides money to attract and retain top talent.
- Check why the 18–24 age group appears underrepresented.
- **It is important to remember that survey results may be biased and may not provide statistically valid information. While they can attract attention and offer useful insight, more in-depth research is needed to draw conclusions and make serious decisions.**

APPENDIX



```
> df_age = df.loc[df["Age"] != "Prefer not to say", ["Age"]]

# Copy df and assign a random age within the reported range
def assign_random_age(age_group):
    if age_group == "Under 18 years old":
        return np.random.randint(15, 18)
    elif age_group == "18-24 years old":
        return np.random.randint(18, 25)
    elif age_group == "25-34 years old":
        return np.random.randint(25, 35)
    elif age_group == "35-44 years old":
        return np.random.randint(35, 45)
    elif age_group == "45-54 years old":
        return np.random.randint(45, 55)
    elif age_group == "55-64 years old":
        return np.random.randint(55, 65)
    elif age_group == "65 years or older":
        return np.random.randint(65, 76)
    else:
        return None

df_age["ExactAge"] = df_age["Age"].apply(assign_random_age)
df_age = df_age.sort_values("ExactAge").reset_index(drop=True)
print(df_age.shape)
print(df_age.head())

plt.figure(figsize=(14,3))
plt.boxplot(df_age["ExactAge"], vert=False)
plt.title("Boxplot of Age")
plt.xlabel("Age")
plt.xticks(range(15, 76), fontsize=8)

# Compute statistics
data = df_age["ExactAge"]
min_val = np.min(data)
q1 = np.percentile(data, 25)
median = np.median(data)
q3 = np.percentile(data, 75)
max_val = np.max(data)

# Annotate values
y_pos = 0.9 # center line of your boxplot
offset = 0.2
for val, label in zip([min_val, q1, median, q3, max_val],
                      ["Min", "Q1", "Median", "Q3", "Max"]):
    plt.text(val, y_pos + offset, f"{label}\n{val:.0f}",
             ha="center", va="bottom", fontsize=8, rotation=0)
plt.show()
```

[4]

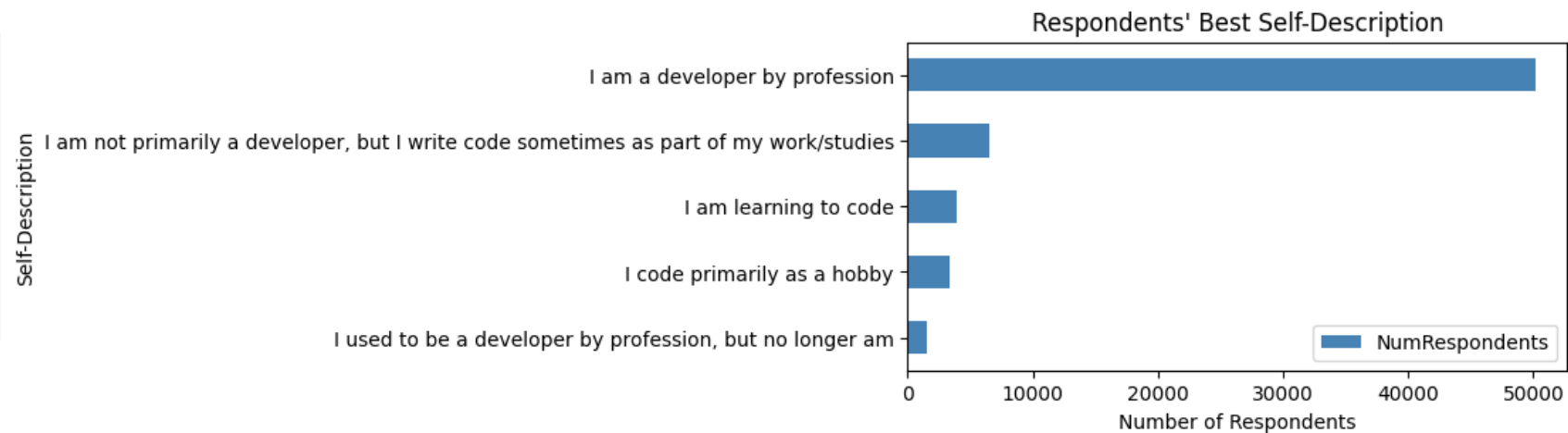
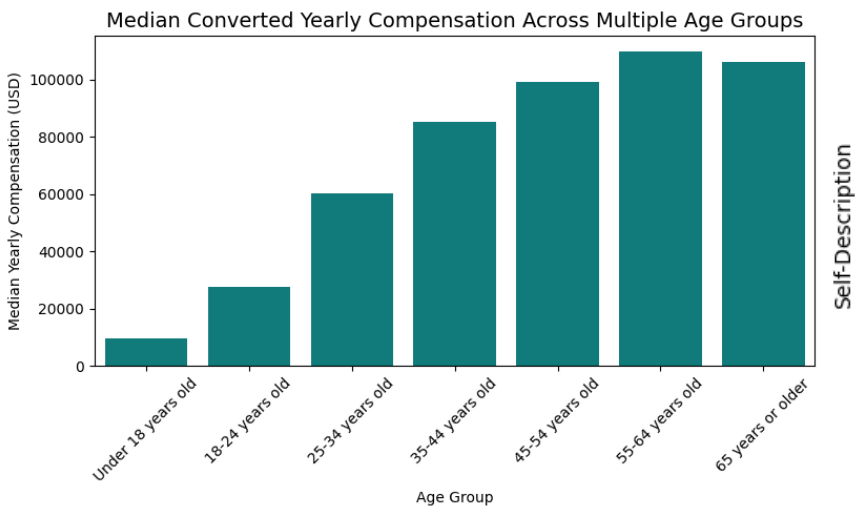
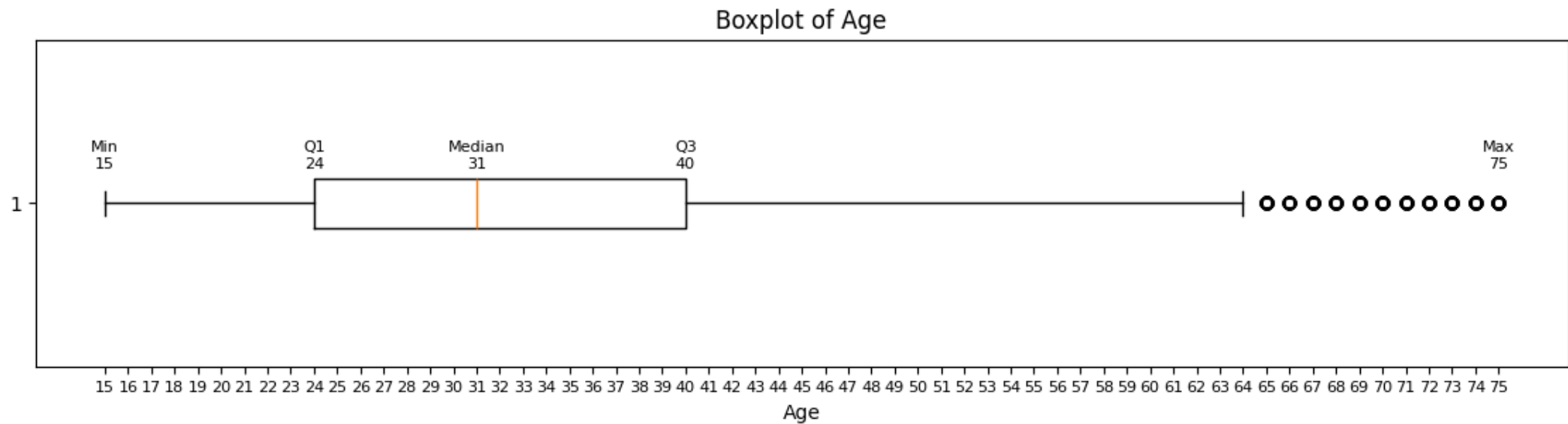
```
... (65115, 2)
```

	Age	ExactAge
0	Under 18 years old	15
1	Under 18 years old	15
2	Under 18 years old	15
3	Under 18 years old	15
4	Under 18 years old	15

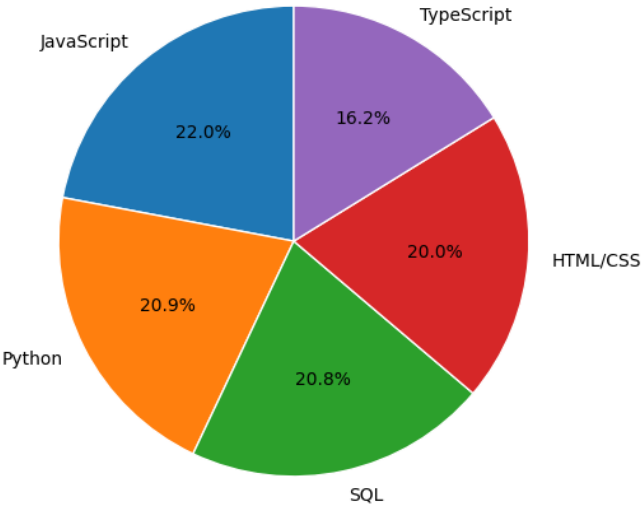
...

Boxplot of Age

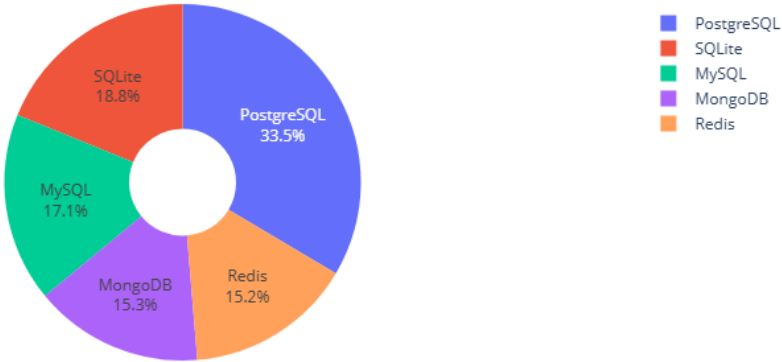




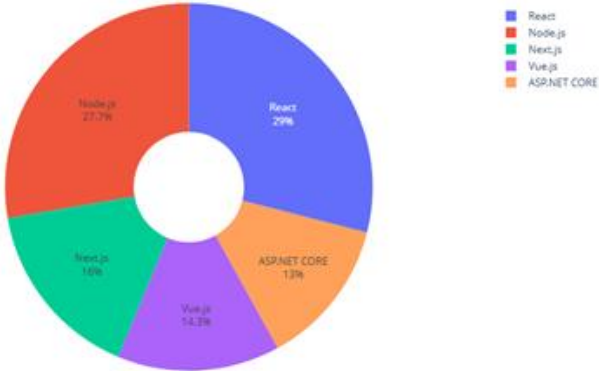
Top 5 Most Admired Programming Languages



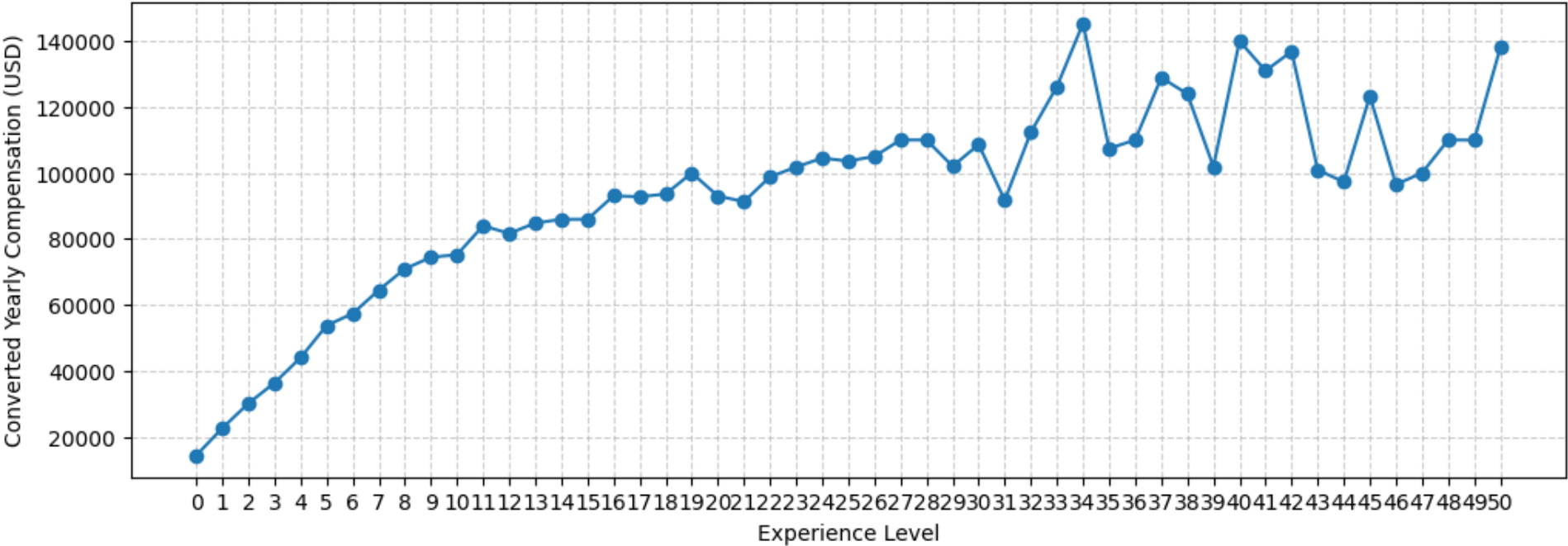
Top 5 Databases Respondents Want to Work With



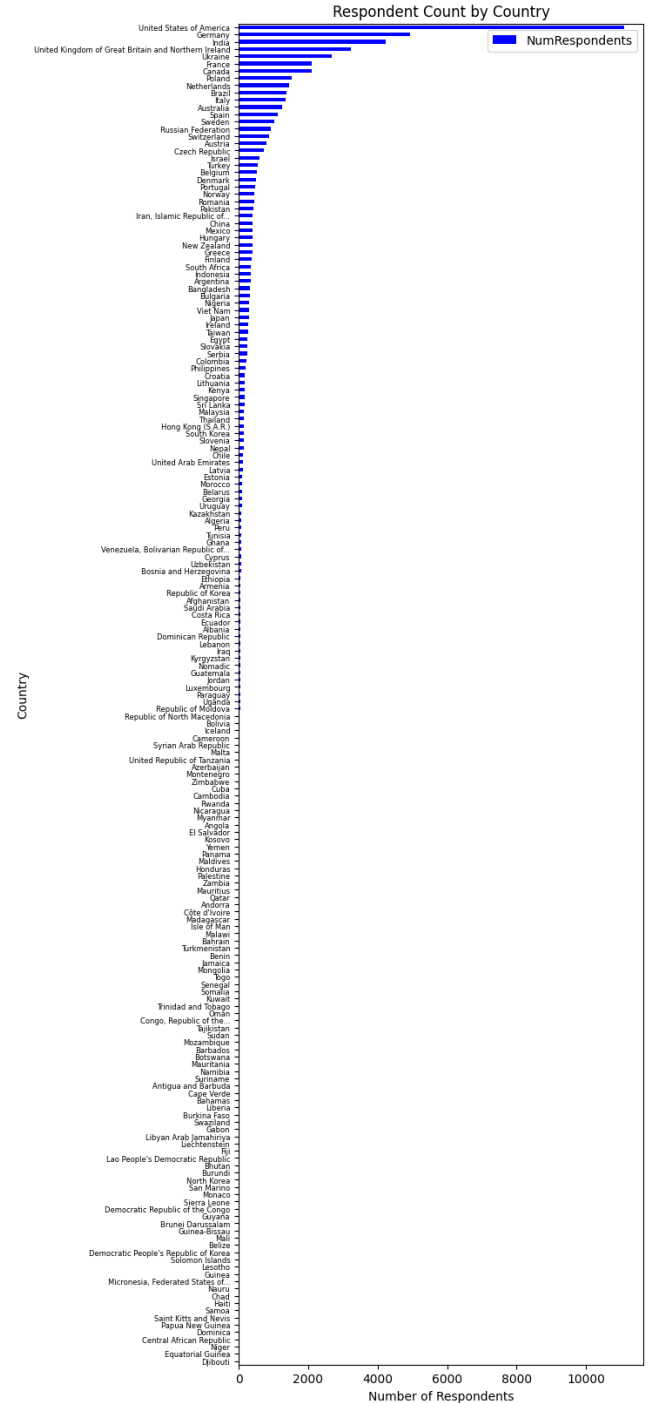
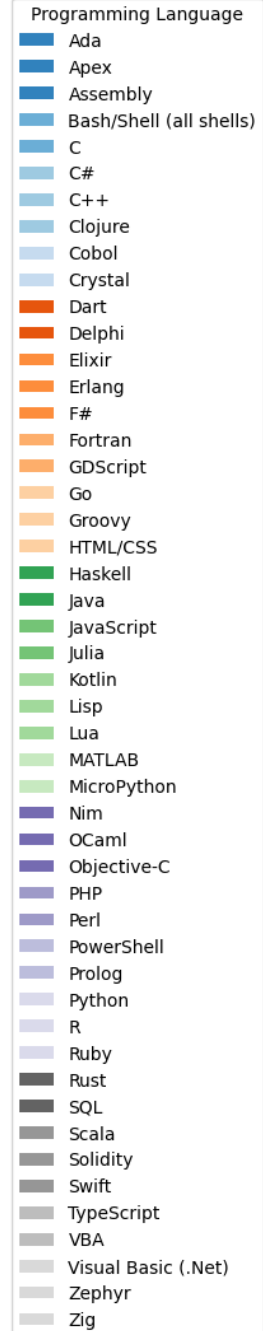
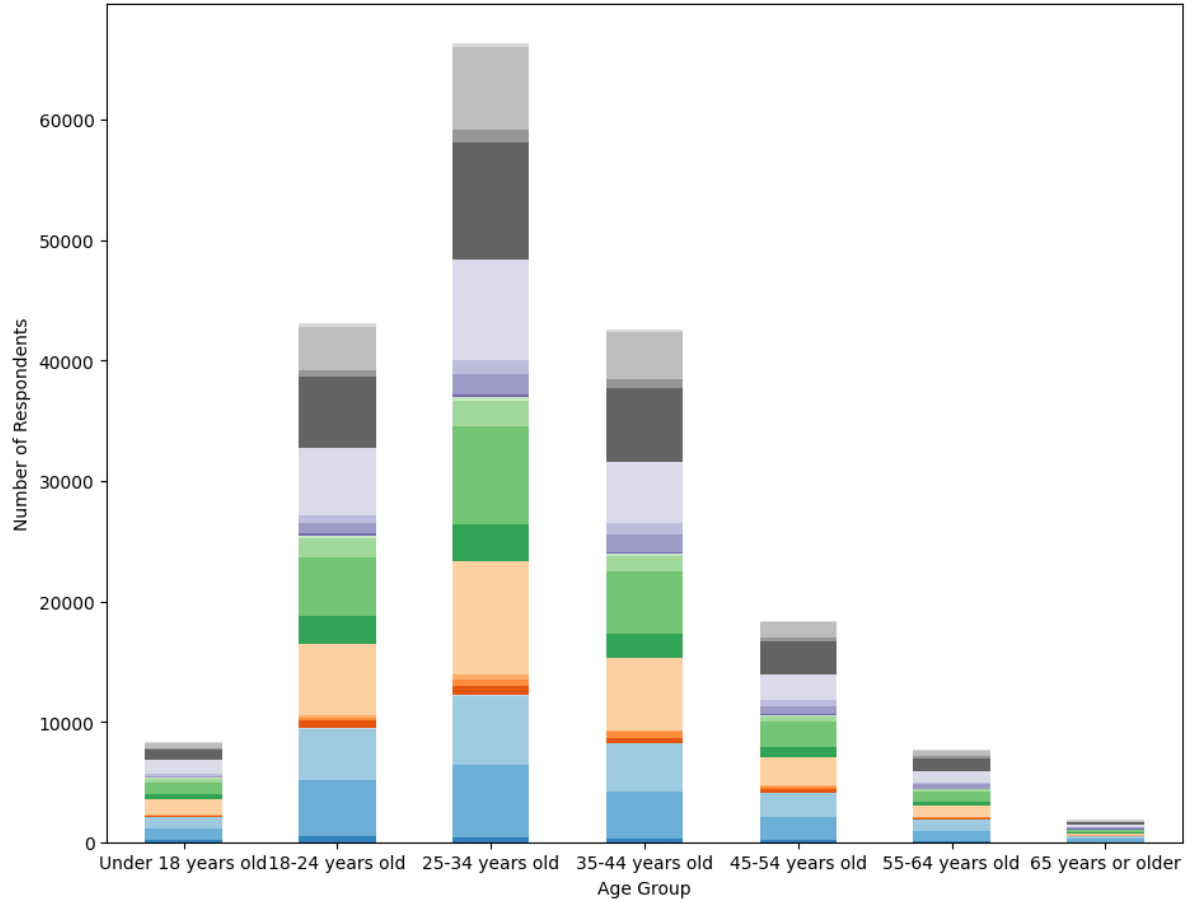
Top 5 Web Frameworks Respondents Are Interested In Working With



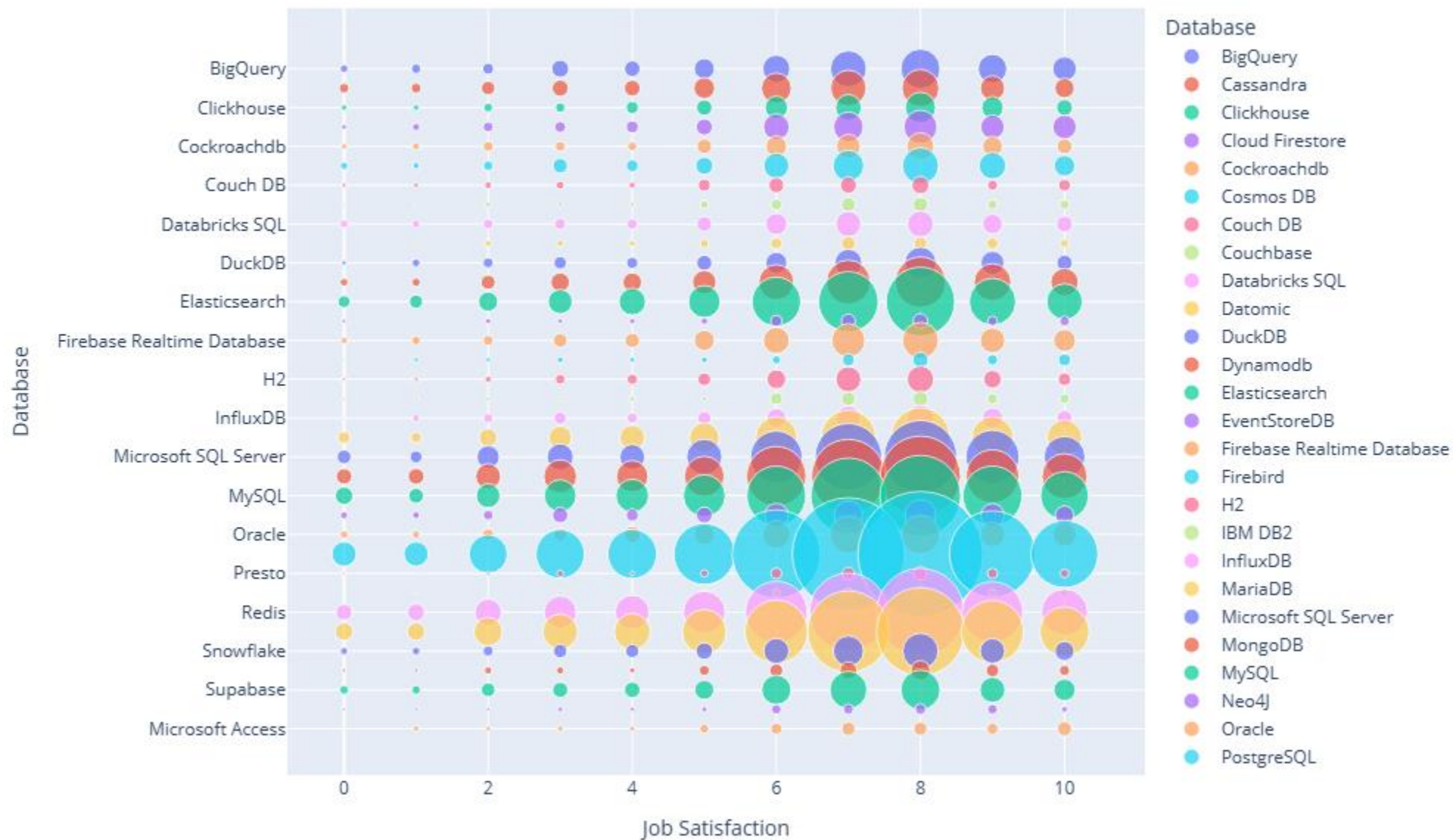
Median Converted Yearly Compensation Over Experience Level



Preferred Programming Languages by Age Groups



Bubble Plot for Preferred Databases vs. Job Satisfaction
(bubble size = number of respondents for each database)




```
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

[3]

Connect to the SQLite database

```
conn = sqlite3.connect('survey-data.sqlite')
```

[4]

Demo: Basic SQL queries

Demo 1: Count the number of rows in the table

```
QUERY = "SELECT COUNT(*) FROM main"
df = pd.read_sql_query(QUERY, conn)
print(df)
```

[5]

```
...    COUNT(*)
0         65437
```

1.1 Histogram of `CompTotal` (Total Compensation)

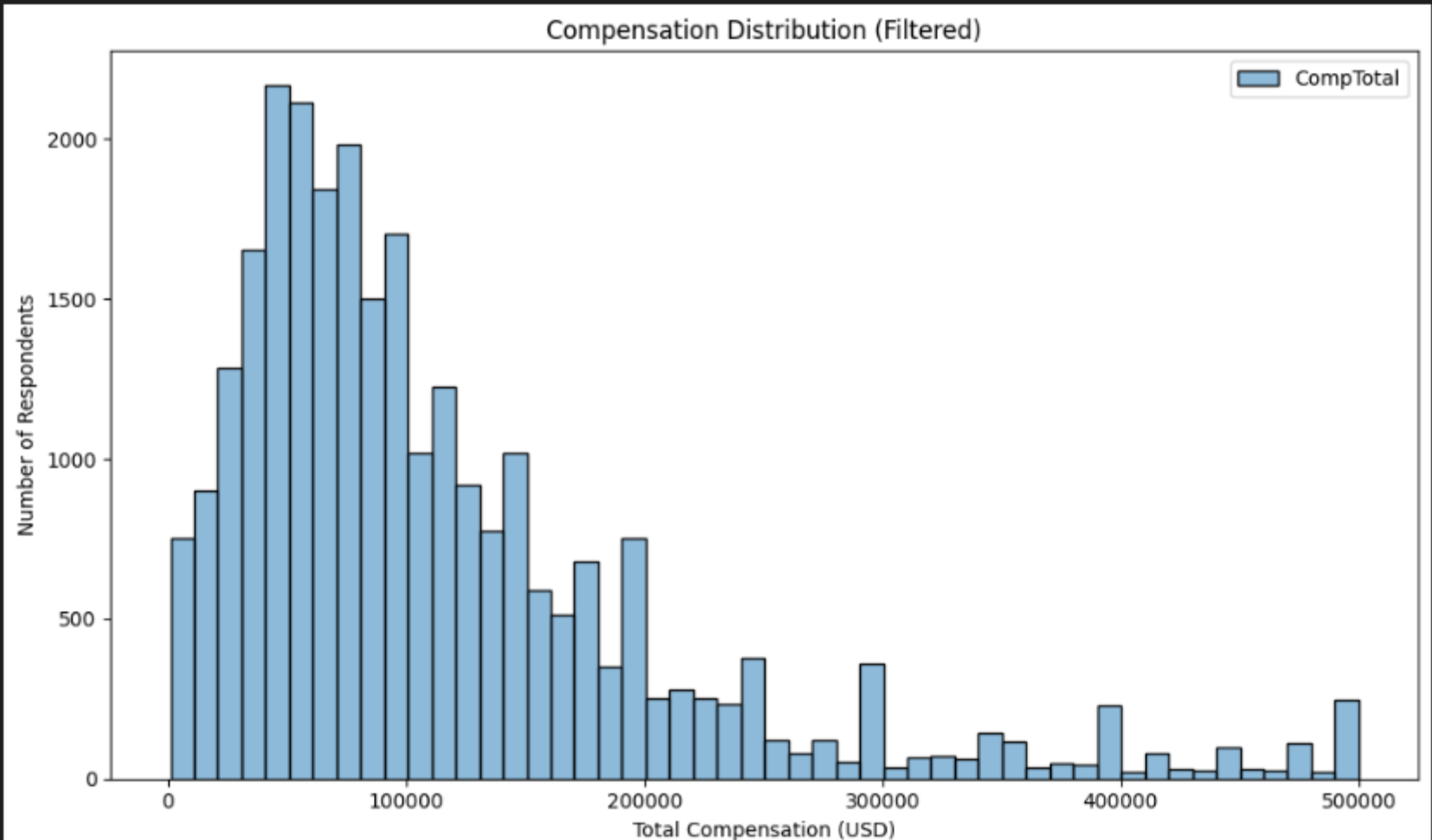
Objective: Plot a histogram of `CompTotal` to visualize the d

```
## Write your code here
QUERY = """
SELECT CompTotal
FROM main
WHERE CompTotal IS NOT NULL AND CompTotal >= 1000
"""
df_comptotal = pd.read_sql_query(QUERY, conn)
```

```
df_comptotal = df_comptotal.loc[df_comptotal["CompTotal"]<=500000]
plt.figure(figsize=(10, 6))
sns.histplot(df_comptotal, bins=50, kde=False)
# plt.xlim(0, 500000)
plt.xlabel("Total Compensation (USD)")
plt.ylabel("Number of Respondents")
plt.title("Compensation Distribution (Filtered)")
plt.tight_layout()
plt.show()
```

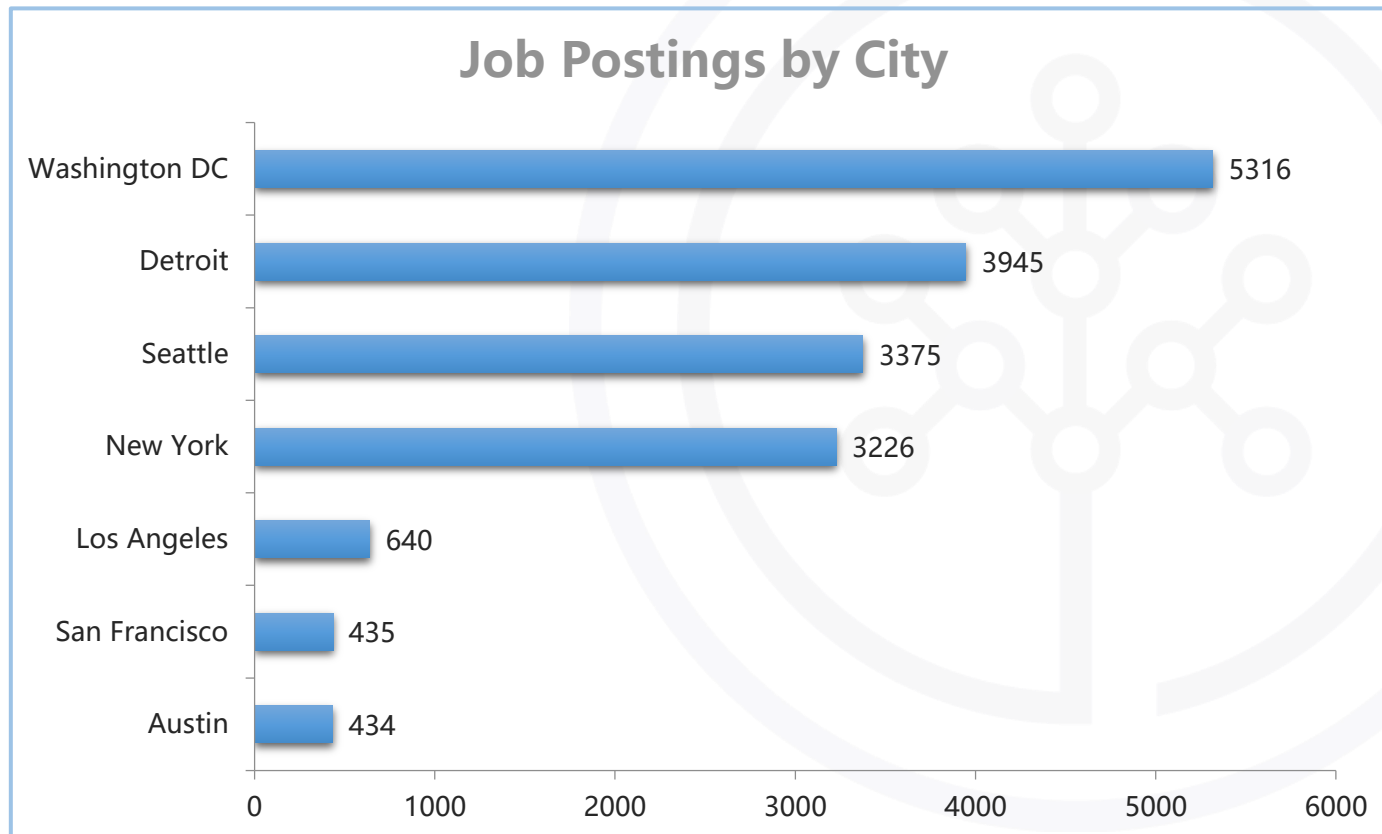
[9]

...



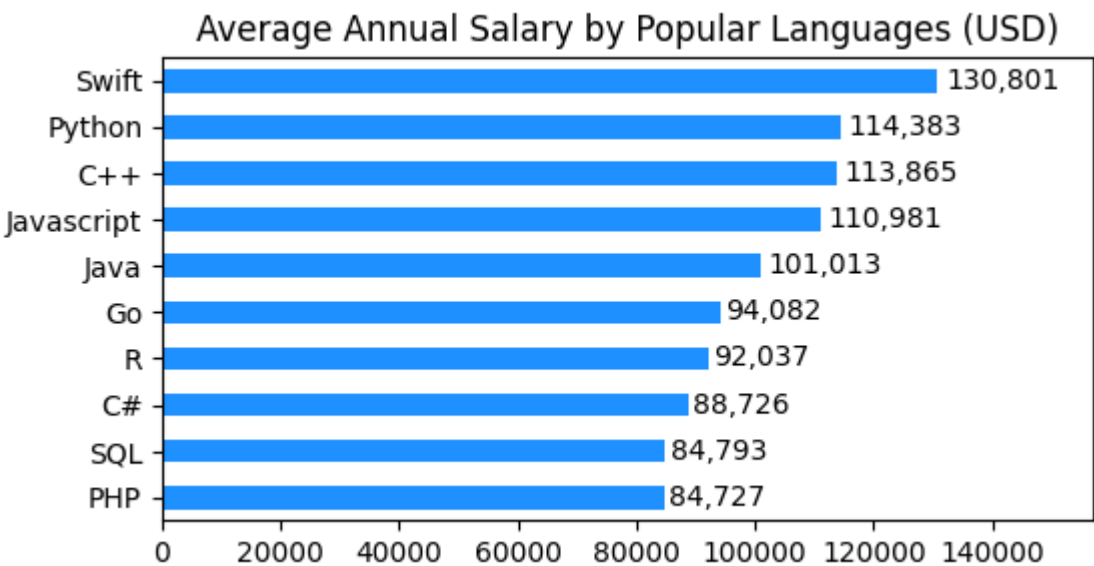
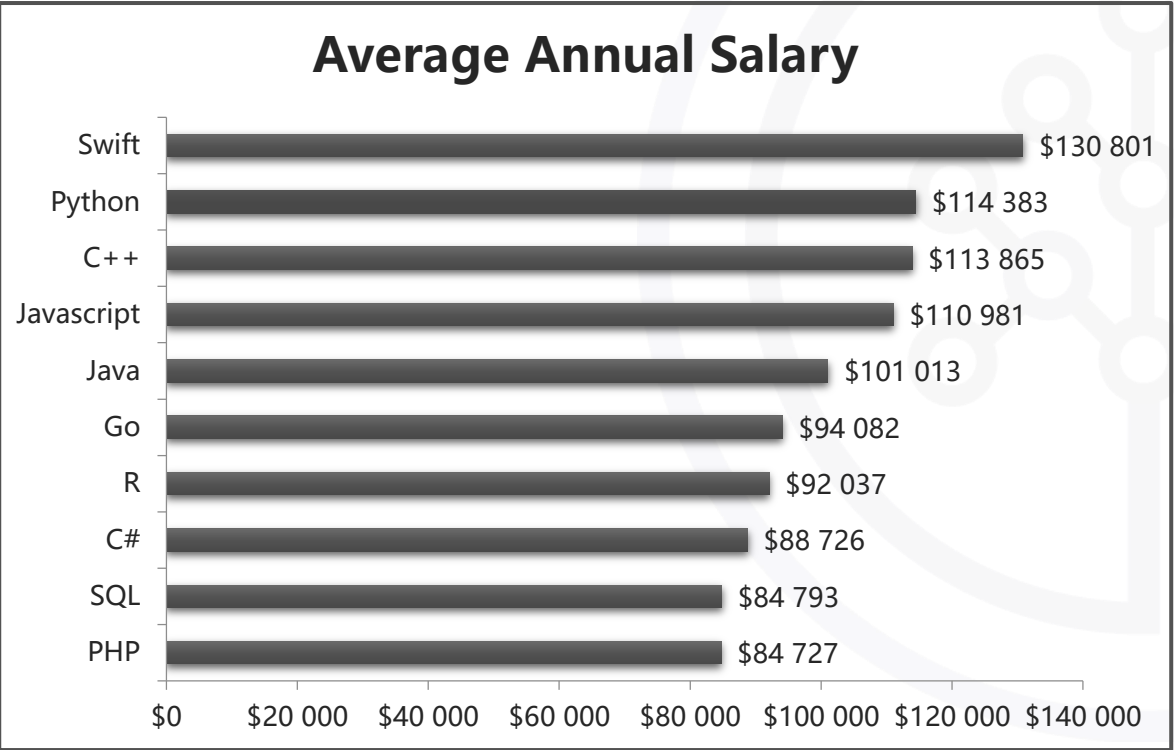
JOB POSTINGS

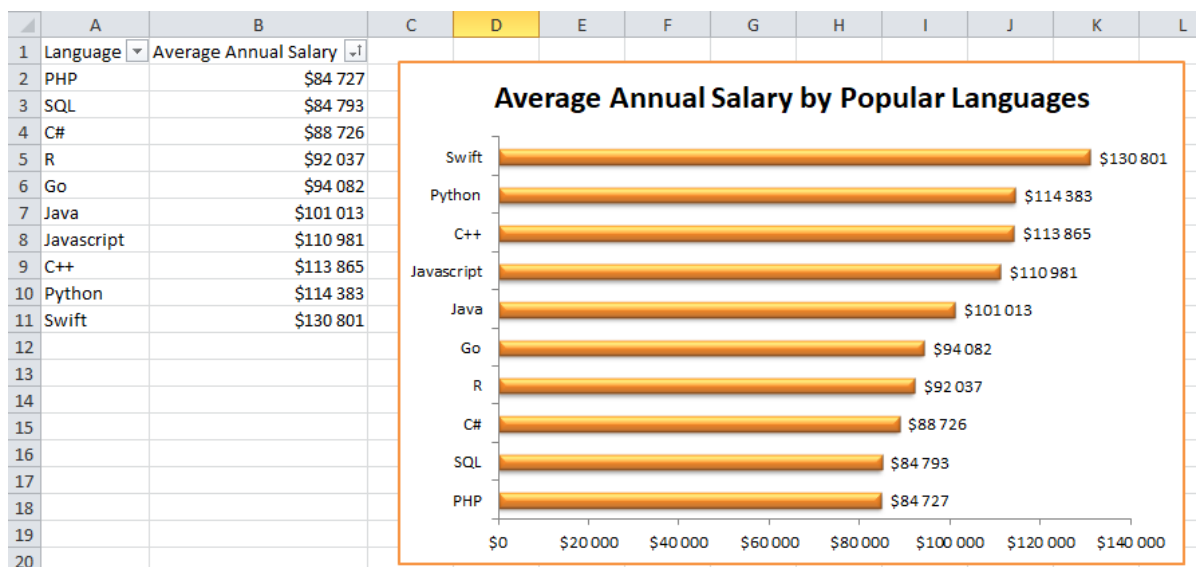
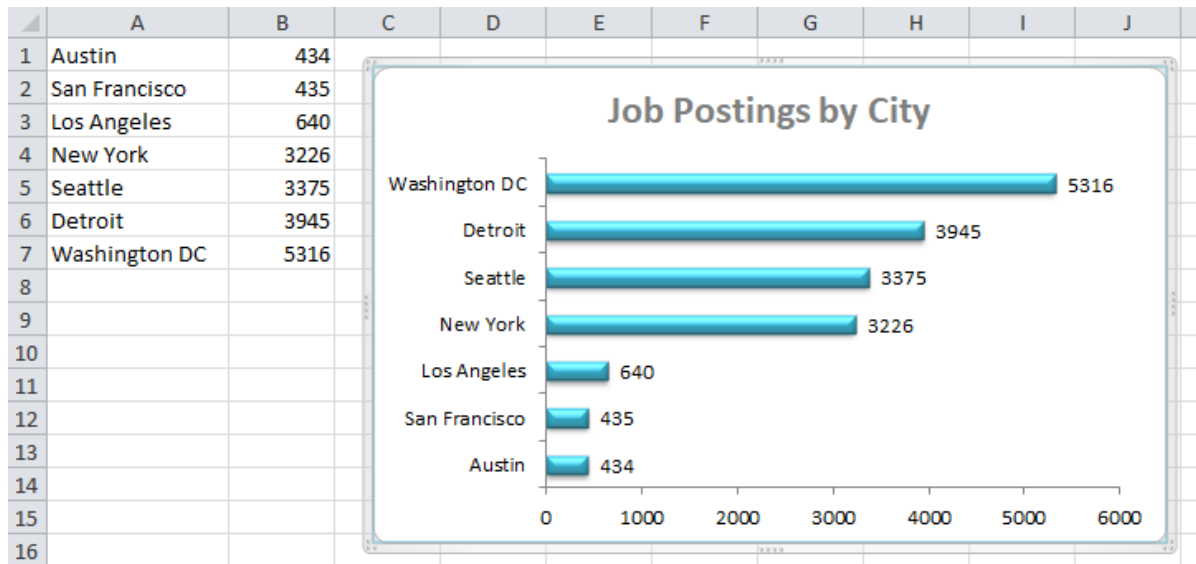
Bar chart presenting the job posting data collected using GitHub Job API and saved to Excel file named "job-postings.xlsx" (visualized in Excel).



POPULAR LANGUAGES

Bar chart displaying popular languages and their average annual salary. The data was collected through web scraping the GitHub jobs data and saved in a "popular-languages.csv". Visualized using Excel and Python Matplotlib.pyplot library.





```

import os
import pandas as pd
import matplotlib.pyplot as plt

# Get current working directory
cwd = os.getcwd()
csv_filename = os.getcwd() + "\\popular-languages.csv"
# print(csv_filename)
df_pl = pd.read_csv(csv_filename)
# print(df_pl.head(20))

# Remove the symbols "$" and "," and convert the column to a numeric type
df_pl['Average Annual Salary'] = df_pl['Average Annual Salary'].str.replace('[,$]', '', regex=True)

# Convert the cleaned column to a numeric data type
df_pl['Average Annual Salary'] = pd.to_numeric(df_pl['Average Annual Salary'])
df_pl = df_pl.sort_values("Average Annual Salary").reset_index(drop=True)

# Print the updated DataFrame to verify the result
# print(df_pl.info())

# Create the bar chart
ax = df_pl.set_index("Language").plot(
    kind='barh', figsize=(6, 3), color='dodgerblue', legend=False, ylabel = ""
)

plt.title("Average Annual Salary by Popular Languages (USD)")
# Add labels to the right of bars
for container in ax.containers:
    for bar in container:
        width = bar.get_width()
        ax.text(
            width + (0.01 * width), # a bit to the right of bar
            bar.get_y() + bar.get_height()/2,
            f"{int(width):,}",
            va='center'
        )

# Extend x-axis so labels fit
ax.set_xlim(0, df_pl["Average Annual Salary"].max() * 1.2)

plt.show()

```

[19] ✓ 0.9s