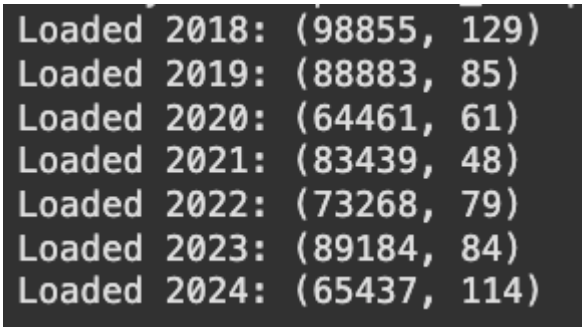
1. Context, Data Curation & Statistical Population

a. **Project goal:** In today's fast-moving tech world, figuring out how your skills affect your salary is crucial for your career. Our project is all about creating a model that predicts a developer's salary based on things like skills, experience, education, and other personal details. We'll also look at how the demand for different skills has changed over time using survey data. By doing this, we hope to help developers understand which skills are worth focusing on now and in the future, so they can stay ahead in the job market.

b. Entities & Relations:

i. Entity: Respondent (primary key respondent_id).
ii. Attributes (1-1): demographics, experience, salary.
iii. Multi-valued relations (1-N): language stacks, DBs, frameworks, etc.

c. Dataset Origins



d. Role & Curation Rules

i. Kept all years to preserve temporal signal.
ii. Dropped columns missing in ≥ 3 years.

e. **Transformation & Linking Pipeline**

i. Alias consolidation → 39 canonical headers.
ii. Row-wise merge of alias columns (first non-NA wins).
iii. Unit normalisation:
    1. Currency → USD via mid-market annual FX table.
    2. CPI → real-2024 USD.
iv. Numeric coercion & ordinal maps (`education_level`, `survey_length`, …).
v. One-hot top-N tokens (N = 15) for 13 multi-select fields; rare tokens dropped.

f. Statistical View: We treat each record as an *iid* draw from the annual developer population, acknowledging survey self-selection bias. Target variable salary. Distribution assumed log-normal conditional on covariates.

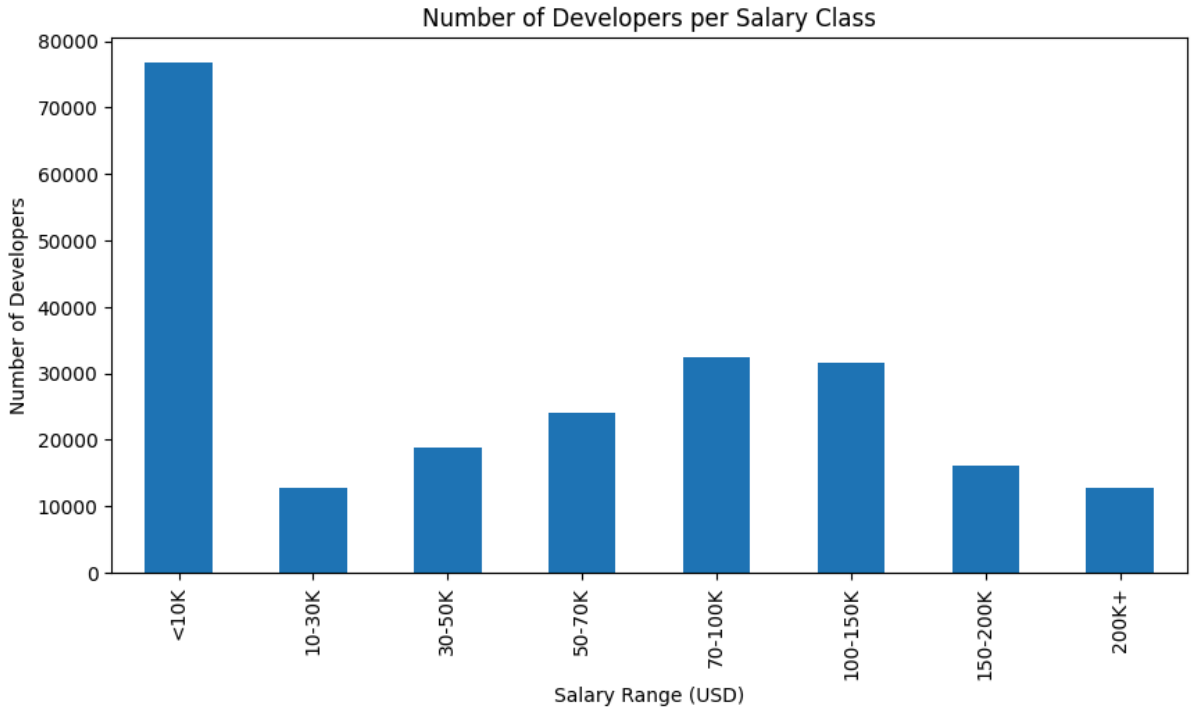
2. Exploratory Data Analysis and Cleaning

Preprocessing steps:

- **Harmonize Column Names:** Mapped inconsistent column names to a common schema across years.
  Ensures we work with the same variable names for all years.
- **Keep 39 Canonical Columns:** Filtered and retained only the 39 most relevant columns.
  This removes irrelevant data and focuses the analysis on key features.
- **Drop Empty Columns:** Removed columns that were completely NaN. They provide no information and clutter the dataset.
- **Drop Low Information Columns** Dropped columns with >95% missing values. Such columns contribute little to analysis or modeling.
- **Convert to Numeric:**Convert numeric-looking data to proper numerical types. Prepares data for modeling and statistical analysis.
- **Numerisation & One-Hot Encoding:** Mapped categorical fields (like education level) and encoded multi-choice fields. Makes the data machine-learning friendly and numerically consistent.
- **Build Token Frequencies & One-Hot Top Tokens:** Created frequency dictionaries and encoded top answers only. Reduces dimensionality and focuses on common patterns.
- **Convert Currency to USD (Nominal):** Standardized compensation using FX rates to USD. Allows fair salary comparisons across countries.
- **Adjust to 2024 USD using CPI:** Adjusted nominal salaries to 2024 values. Accounts for inflation and makes salary data temporally comparable.
- **Drop Empty Columns from Encoded Data:** Removed any remaining all-NaN columns post-encoding. Keeps data tidy and avoids errors in later steps.
- **Country Median Incomes:** Pulled country-level median income from World Bank data. To compare individual salaries with national benchmarks.
- **Drop Extra Columns (Country, Currency, ISO):** Removed country-related columns after ratio calculation. Prevents data leakage and keeps the dataset lean.
- **Merge All Years into df_all:** Concatenated all cleaned yearly datasets into one DataFrame. Allows overall analysis and cross-year comparisons.
- **Filter Salary Outliers:** Removed extreme low/high salary values outside $1K–$800K. Improves model robustness and analysis accuracy.

Exploratory Data Analysis:

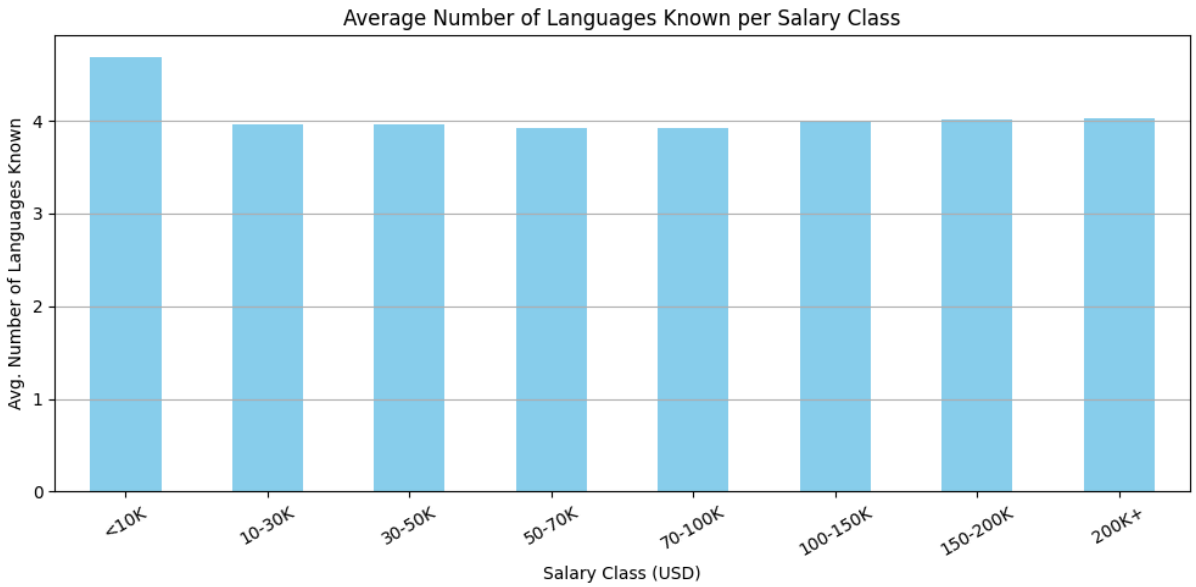
Number of Developers per Salary Class



Why We Need It:
This bar chart helps us understand the overall distribution of developer salaries by grouping them into salary brackets. It gives a high-level overview of where most developers fall in terms of compensation.

Conclusion from Output:
A large portion of developers report earning less than $10K, possibly due to unpaid internships, part-time work, or survey misreporting. It may also reflect respondents from lower-income countries like India, where average salaries are significantly lower. Aside from that, there's a more balanced distribution from $30K–$150K, with a gradual decline above $150K. This indicates that while high salaries exist, the majority of developers earn within a mid-range bracket.

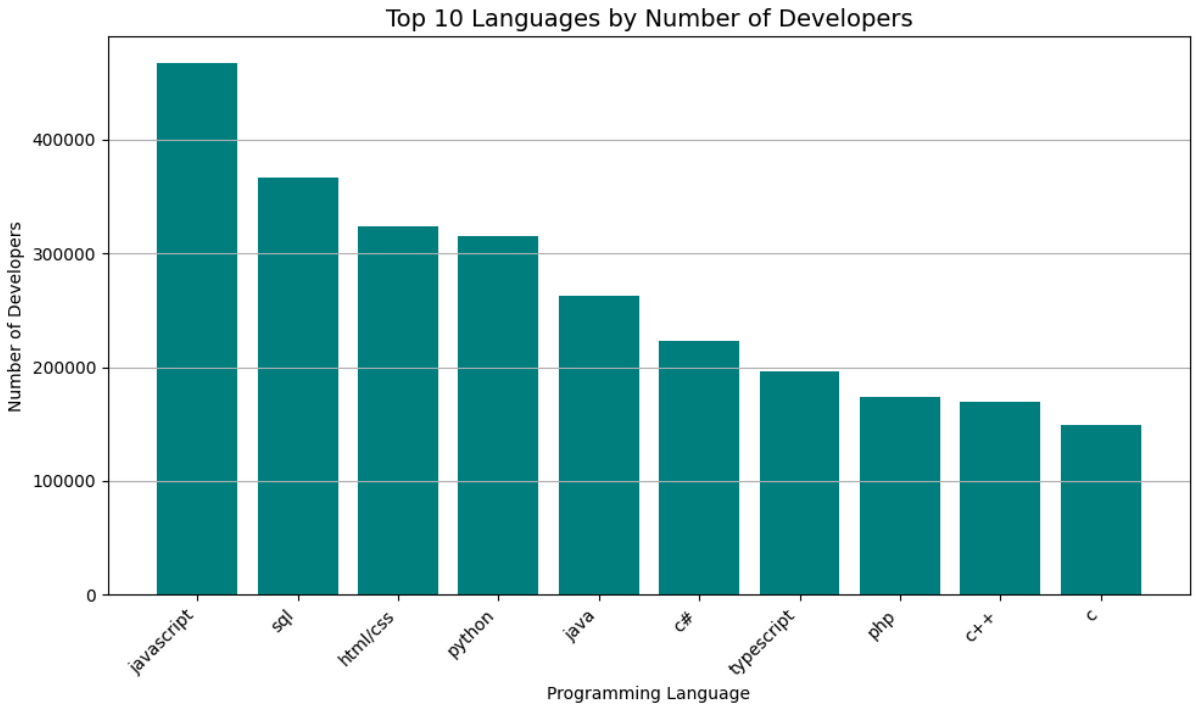
Average Number of Languages Known per Salary Class



Why We Need It:
This chart examines the relationship between a developer's salary and the average number of programming languages they know. It helps assess whether knowing more languages is associated with higher pay.

Conclusion from Output:
Surprisingly, the lowest salary group (<10K) knows the most languages on average. This could suggest students or junior developers exploring many languages, while higher earners might focus on fewer but more in-demand languages. Overall, language count alone doesn't guarantee a higher salary.

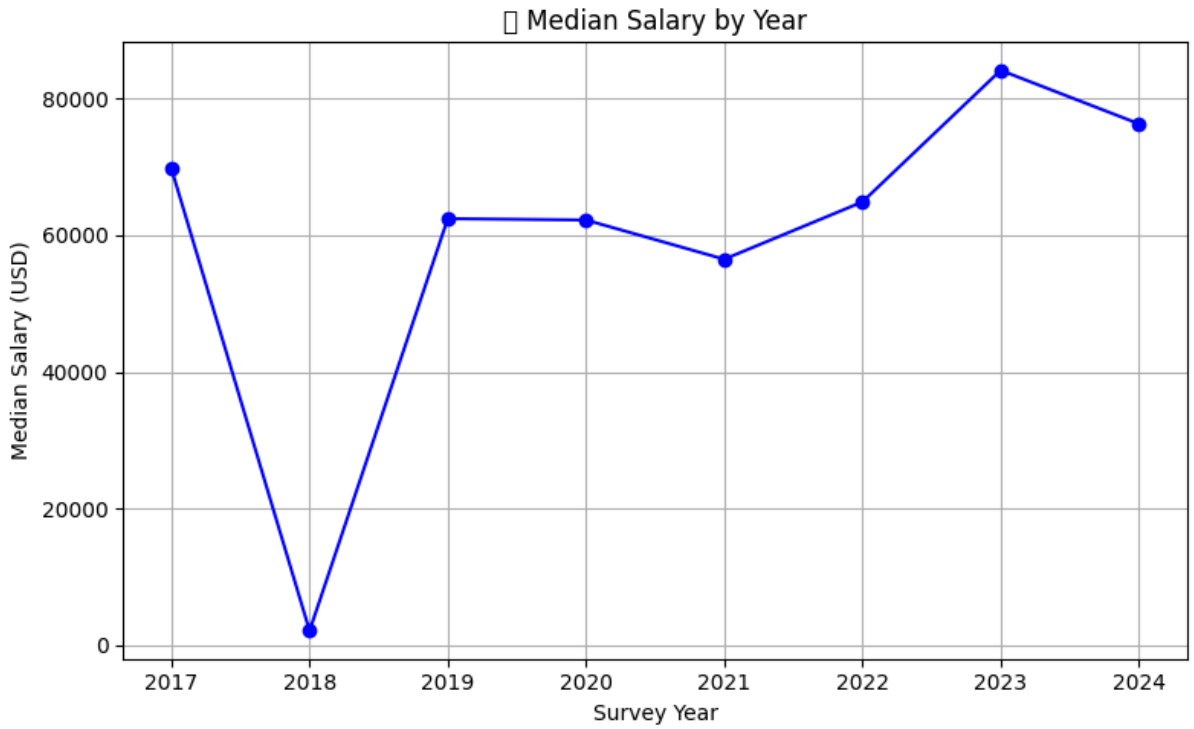
Top 10 Languages by Number of Developers



Why We Need It:
This chart identifies the most commonly used programming languages among developers. It gives insight into language popularity, which can influence hiring, learning priorities, and ecosystem support.

Conclusion from Output:
JavaScript leads by a wide margin, followed by SQL and HTML/CSS, showing the dominance of web-related technologies. Python also ranks high, highlighting its popularity across domains. These trends can guide developers in choosing widely adopted languages for career growth.

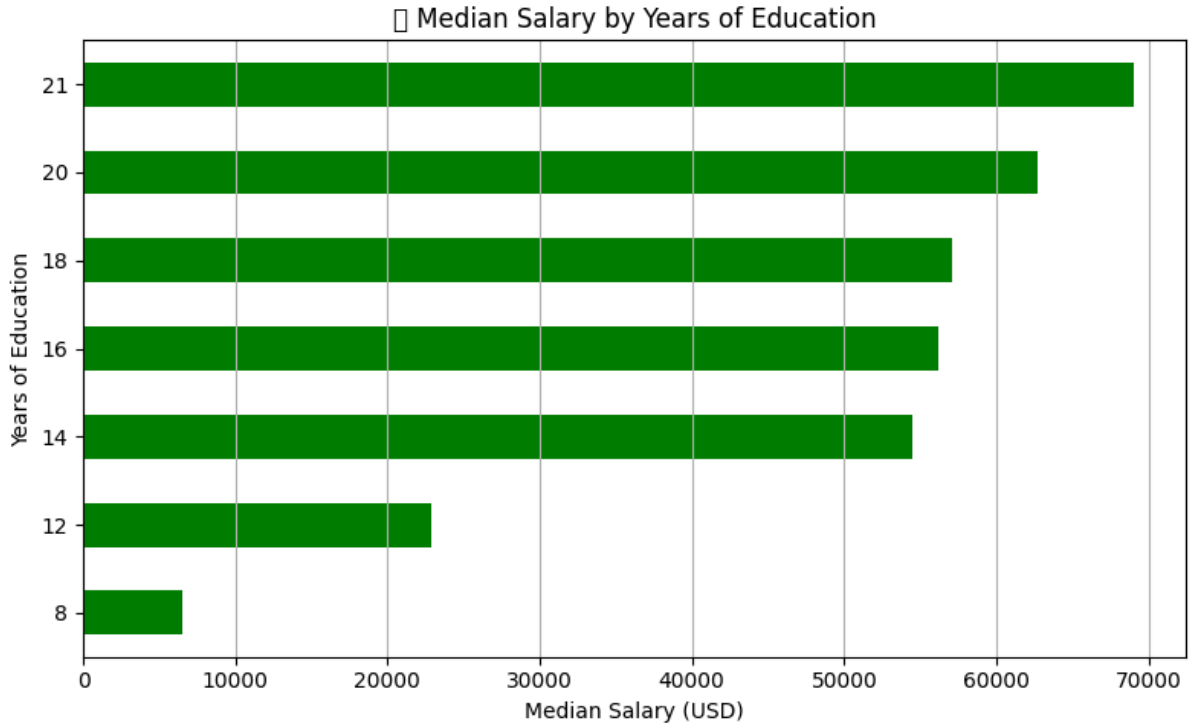
Median Salary by Year



Why We Need It:
This line chart helps us observe how developers' median salaries changed over time. It reveals trends, outliers, and potential influences like economic shifts or data quality issues.

Conclusion from Output:
Salaries generally increased from 2017 to 2023, peaking in 2023. The sharp drop in 2018 is likely a data anomaly or missing values. The decline in 2024 could reflect market corrections or survey composition changes.

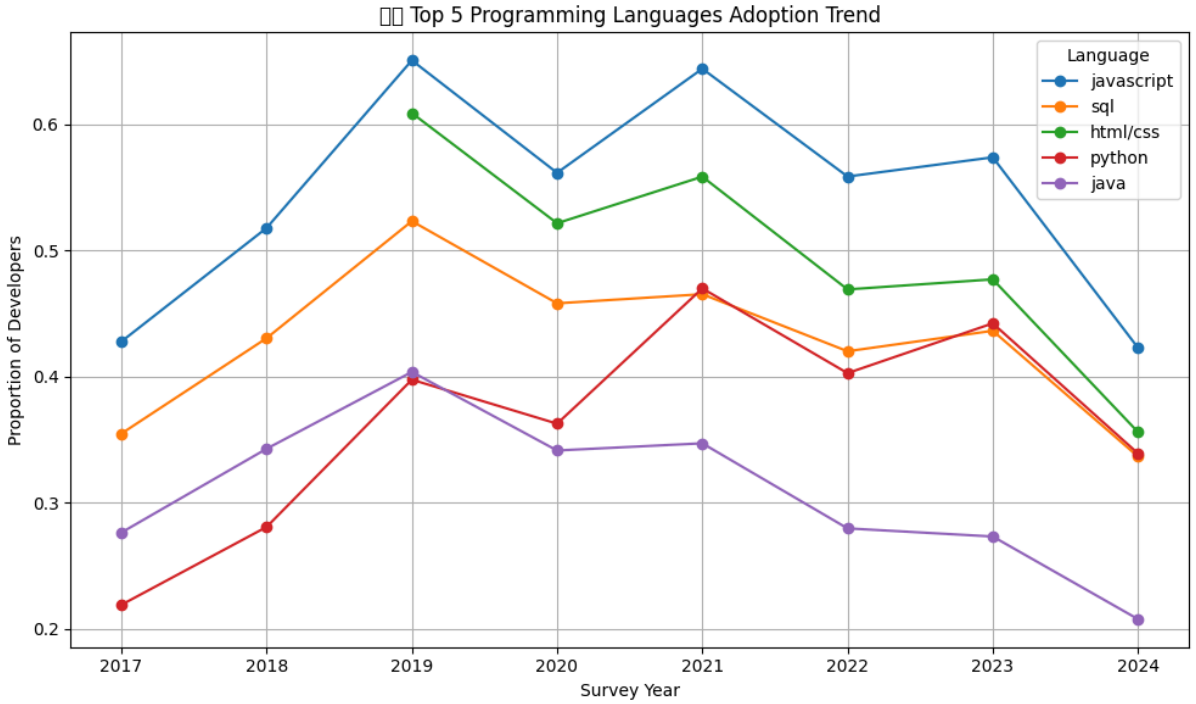
Median Salary by Years of Education



Why We Need It:
This chart shows how the median salary changes based on years of education. It helps analyze the value of educational attainment in relation to developer compensation.

Conclusion from Output:
Salaries increase consistently with education level—developers with 21 years of education (e.g., PhDs and professional degrees) earn the most. This suggests a strong correlation between higher education and higher earning potential.

Top 5 Programming Languages Adoption Trend



Why We Need It:

This line chart tracks how the usage of the most popular programming languages changed over the years. It helps us understand shifting developer preferences and technology trends.