GitHub Programming Language Evolution and Contributor Dynamics (2017 – 2022)

Author: Jenn Kang
February 2026

1. Project Purpose

The objective of this project is to analyse the use of different programming languages and user engagement on GitHub over a five-year period (November 2017 – November 2022).

2. Information on the Dataset

2.1. Summary & Source

The data is sourced from GitHub, made available through the Google BigQuery Public Datasets (bigquery-public-data.github_repos). It contains a full snapshot of content of more than 2.8 million open-source GitHub repositories including more than 145 million unique commits.

2.2. Dataset Organisation

The dataset is structured across several relational tables. For this project, the primary tables used were:

- **commits**: contains nested records for committers and repeated fields for repository names.
- **languages**: contains information that relates repositories with programming languages used and language size in bytes.

2.3. Dataset Quality and Integrity

I will use **ROCCC** for dataset quality and integrity analysis:

- **R (Reliability)**: high
  - The data is highly reliable as it is sourced directly from GitHub and is generated automatically by the system.
- **O (Original)**: high
  - This is a first-party data source.
  - A significant portion of activity is driven by bots. These were removed to ensure the analysis reflects actions conducted by humans.
- **C(Comprehensive)**: high
  - The dataset is comprehensive with over 2.8 million open-source repositories.
- **C(Current)**: medium
- Initial Exploratory Data Analysis (EDA) showed a data cutoff in November 2022. Data after this date is incomplete, insufficient, or non-existent. This is relatively recent but does not contain data from the current or previous year (current year is 2026).

- **C(Cited)**: high
    - The dataset is highly cited and can be tracked back to its original source.

3. Data Used

Table	Column Used	Purpose
commits	committer.date.seconds	Time-series filtering.
commits	committer.name	Identify unique committers/contributors and bot filtering.
commits	repo_name	To join data to specific repositories.
languages	language.name	Identifying languages used in repositories.
languages	language.bytes	Measuring the size/proportion of a language used in a repository.

4. Exploring the Data

1. The *commits* table in the GitHub dataset (bigquery-public-data.github_repos.commits) shows that some records under column *repo_name* are stored in arrays.
    a. I.e. a unique commit may be linked to multiple repositories.
2. Noticed a record with 'bot' as the *committer.name* in the *commits* table. Further investigation showed there are commit records actioned by non-humans: bots, automation, service-account.
3. The *languages* table in the GitHub dataset (bigquery-public-data.github_repos.languages) shows that some records under column *languages* are stored in arrays.
    a. I.e. a unique repository may have multiple languages used.
4. After the first attempt at visualising the data, a few things were noted:
    a. Google Looker Studio cannot process large amounts of data for a single visualisation chart.
    b. The most recent month containing usable data (e.g. has multiple commits, is an accurate date not set far into the future) was November 2022.
5. BigQuery on the free version has a limit of 1TB of processing data per month.

5. Processing the Data

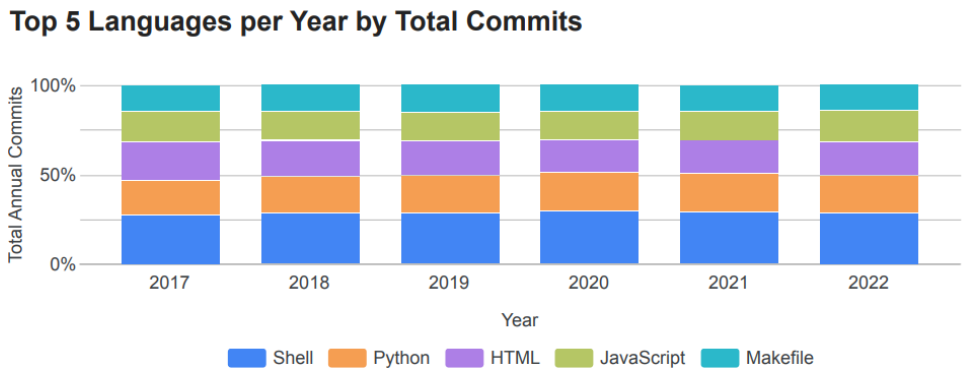
1. Created *summary_stats* table to minimise the amount of data to work with due to BigQuery free user limitations (summary_stats.sql).
    i. Used CROSS JOIN UNNEST() to flatten the records in the *commit* and *language* tables, as the respective values are stored in arrays.
    ii. Only selected relevant columns to use for this project.
    iii. Used REGEXP_CONTAINS on *committer.names* to exclude records of commits by bot/automation/service-accounts.
    iv. Converted the timestamps (in seconds) into BigQuery TIMESTAMP formats, then truncated them to months to provide clean time-series aggregation.

v. Filtered data to only 2017-11-01 to 2022-11-30 (the most recent and usable 5 years of data).

2. Queried to calculate the percentiles determine category cutoffs for user engagement based on total commits made in the 5-year period (2017 – 2022).
    - 25th percentile: 2 total commits
    - 50th percentile: 5 total commits
    - 75th percentile: 25 total commits
3. Created separate tables of data for each of the visualisations to minimise the amount of data processed on Looker Studio.
    a) top_5_language_yearly.sql:
        i. Stores the top 5 languages used (determined by total commit count that year) for years 2017 – 2022.
    b) top_5_growth.sql:
        i. Stores the top 5 languages that have the highest growth rate over the period (2017 – 2022).
        ii. $Growth\ Rate = \frac{(Total\ Commits\ in\ Nov.2022) - (Total\ Commits\ in\ Nov.2017)}{Total\ Commits\ in\ Nov.2017}$
    c) language_correlation.sql:
        i. Cross-tabulation/self-join of languages, counting the total number of repositories that use the same language, to determine language to language correlation.
    d) top_5_repos_2022.sql:
        i. Stores the top 5 repositories in 2022 (engagement) and the top 3 languages used within each (size of the language (bytes) used in the repository in percentage).
        ii. $Engagement = SUM(commits) + SUM(Unique\ Committers)$
        iii. $Percent\ Language\ in\ a\ Repository = \frac{Average\ Language\ Size\ (Bytes)}{SUM(Average\ Language\ Size\ (Bytes)}$
    e) contributer_segments.sql:
        i. Grouped users (*committer_name*) by how many total number of commits they made in the 5 years (2017 – 2022).
            - **Occasional Users**: 0-2 total commits
            - **Light Users**: 3-5 total commits
            - **Medium Users**: 6-25 total commits
            - **Heavy Users**: 26+ total commits

6. Data Analysis

6.1. Language Popularity

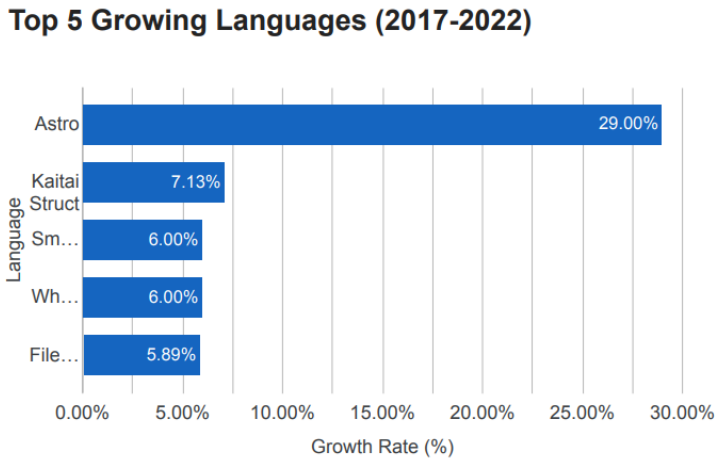


Observation:

- The top 5 languages used (based on total annual commits) is consistent over 2017 – 2022 (Shell, Python, HTML, JavaScript, Makefile).

Insight: The stability of the top 5 languages may suggest these are the core languages used globally. Software developers would highly benefit from being knowledgeable in these languages as they are not expected to drop off in the near future.

6.2. Language Growth



Observation:

- Astro is the top growing language at a 29% increased usage over the 5-year period.
- Kaitai Struct, Smithy, Whiley, and Filebench WML have similar growth rates (6-7%).

Insight: Learning Astro would open new opportunities for software developers due to being skilled in a niche language.

6.3. Language Correlation Matrix

Correlation between Languages (2017-2022)

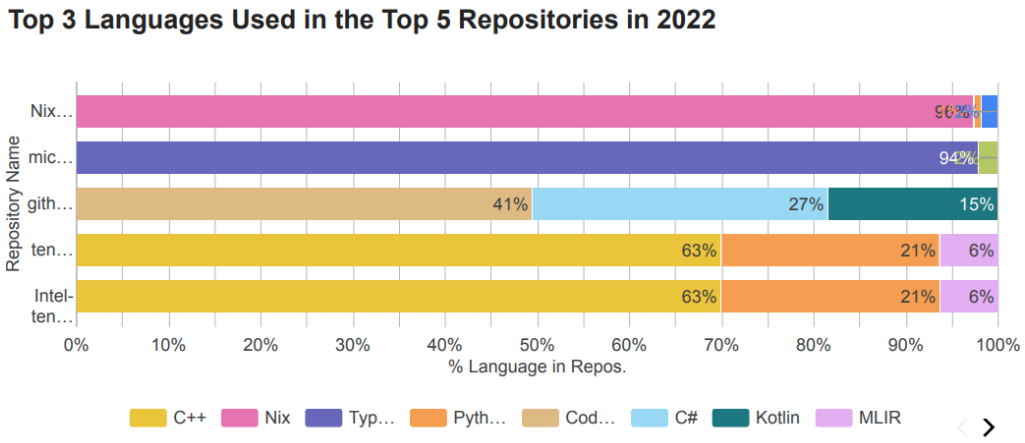
						Language / Shared Repos. Count		
Language	Shell	JavaSc…	HTML	Python	PHP	C++	Ruby	Makefile
CSS	39.3k	119.7k	123.3k	25.1k	25.7k	-	-	-
HTML	50.3k	127.7k	-	34.6k	-	-	31.2k	-
JavaScript	46.3k	-	-	27.2k	26.3k	-	-	-
C	29.9k	-	-	-	-	33.4k	-	29.3k
Makefile	39.6k	-	-	29.6k	-	-	-	-
Python	53.6k	-	-	-	-	-	-	-
C++	25.6k	-	-	-	-	-	-	-

Observation:

- JavaScript is heavily correlated with CSS and HTML (120k and 128k shared repositories, respectively).
- HTML is heavily correlated with CSS with 123k shared repositories respectively.
- Shell is somewhat correlated with HTML, JavaScript, and Python (50k, 46k, and 54k shared repositories, respectively).

Insight: There is an opportunity for growth if you are knowledgeable in JavaScript, but are lacking in CSS and HTML, as these appear to be heavily correlated with each other.

6.4. Languages used in the Top 5 Repositories in 2022

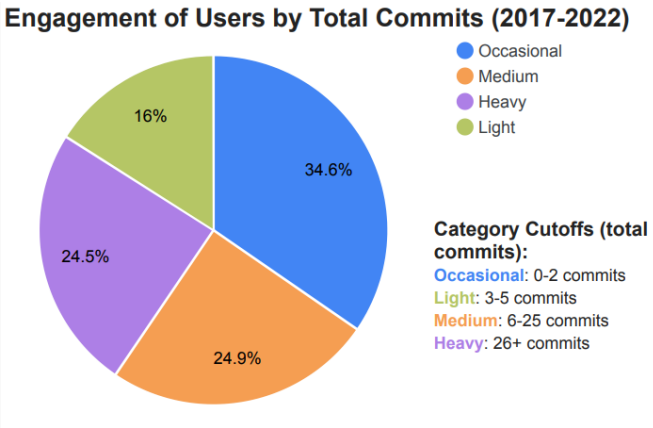


Observation:

- The top repository (NixOS/nixpkgs) predominantly uses Nix (96% of the whole repository).
- The 2nd top repository (microsoft/vscode) predominantly uses TypeScript (94% of the whole repository).
- The 4th and 5th top repository use the same proportions of C++, Python, and MLIR languages (63%, 21%, and 6% of the whole repository, respectively).

Insight: If you want to work on elite projects, you will need to also be specialised in the field related to it. Alternatively, building knowledge in Python (which is also a top language used consistently across 5 years) will be extremely useful.

6.5. Contributor Engagement



Observation:

- Users on GitHub are majority occasional users (commits 0-2 times within 5 years) at 34.6%.
- Light Users (commits 3-5 times in 5 years) are the smallest group at 16%.
- Medium and Heavy Users make up similar proportions at 24.9% and 24.5%, respectively.

Insight: As 34% of users are 'Occasional' you have an opportunity to stand out by consistently contributing over time (e.g. by updating your repositories), compared to creating one project that does not get updated.

7. Conclusion

- Shell, Python, HTML, JavaScript, and Makefile are the core languages in software development as they remain unchanged in ranking, over a 5-year period.
    - As a software developer, it is crucial to be specialised in at least one of these languages.
- The high growth-rate of Astro may indicate that the industry is moving towards newer and optimised languages.
    - As a software developer, it may be beneficial to become knowledgeable on modern languages (such as Astro) to stand out amongst other developers.
- Software developers are recommended to be knowledgeable in related languages to what they primarily use (e.g. if you primarily use JavaScript, you should be knowledgeable in CSS and HTML also).
- Elite repositories focus on their primary language and use it for over 90% of their code.
    - Software developers would benefit from mastering a single language (like C++ or Nix) compared to having a surface-level understanding of many tools.

- The largest portion of the GitHub community are occasional users, but Medium and Heavy Users make up about 50% of all users. Developers are recommended to focus on consistency in making commits, to build a solid profile.