



Programming Languages

04.06.2023

Feng Yiyang 352042, Zhou Naisong 353331, Tang Xuehan 353567

Introduction

The purpose of this project is to illuminate the impact and distribution of programming languages across the globe. With a dataset grounded on the [Programming Language Dataset \(PLDB\)](#), a rich and expansive resource encompassing 4,304 unique programming languages and their 331 respective attributes, we aim to bring to life the story of programming language evolution. The PLDB contains a series of insightful information, including data on each language's title, country of origin, type, userbase, creators, ranking, as well as metrics from popular platforms such as GitHub, Wikipedia, and Stack Overflow.

This interactive exploration will serve as a powerful tool for understanding the programming landscape, elucidating the rise and fall of languages, their adoption and application in different regions, and the dynamics of their development community.

Whether you're a programming language developer, a software engineer, a computer science student, an educator, or a technology enthusiast, this project will provide a comprehensive, engaging, and intuitive perspective into the programming language universe. With this project, our hope is to deepen understanding and foster a greater appreciation for the progression of the field, and ultimately to inspire further exploration and development in the realm of programming languages.

Our Path for the PLDB Project

Brainstorming

For our project, we set out to identify a data visualization task that would offer rich data to explore and ultimately communicate something compelling to the audience. Initially, we considered two different options: the first was a visualization of character data from Genshin Impact, a well-known game in the anime style, and the second involved an analysis of various data from ChatGPT users. However, we eventually dismissed both of these proposals.

Our reasons were as follows: firstly, the available datasets related to Genshin Impact were limited to character data, offering limited visualization possibilities and perhaps only appealing to a niche audience familiar with the game. Secondly, ChatGPT, being a relatively new platform launched at the end of 2022, had limited user data available. Its primary data, encompassing the distribution of API calls, did not offer a full scope due to privacy issues surrounding user query content.

After careful deliberation, we settled on a theme that is close to our hearts: programming languages. As computer science master students, we possess a deep understanding and passion for this topic. We believe in sharing our knowledge and interest in computer





science, particularly programming, with a wider audience. Whether you're a programming language developer, a software engineer, a computer science student, an educator, or a technology enthusiast, we aim to create a beautifully visualized project that offers a comprehensive, engaging, and intuitive journey through the universe of programming languages. Through this project, we hope to deepen understanding, promote greater appreciation for the progression of the field, and ultimately inspire further exploration and development within the realm of programming languages.

Dataset Selection

Having settled on our theme, we started our hunt for suitable datasets from various sources, including FiveThirtyEight, Google Dataset Search, and Kaggle Datasets. We ultimately discovered the PLDB dataset on Kaggle, an invaluable resource that contains a wealth of insightful information about programming languages. More specifically, this dataset encompasses 4,304 unique programming languages and their 331 respective attributes, detailing each language's title, country of origin, type, userbase, creators, ranking, and Wikipedia summary, in addition to metrics from popular platforms such as GitHub, Wikipedia, and Stack Overflow.

Final Goals

Having identified our dataset, we needed to refine the specific aspects we wished to explore, while also considering the breadth of information the PLDB dataset offered. Eventually, we carefully selected several key aspects to focus on: geographical distribution, evolution over time, relevant terminology, and interconnections among different programming languages. Moreover, we wanted to provide an avenue for users to delve into the specifics of an individual programming language. To effectively convey these insights, we have devised four final visualization components:

-  **Vis 1 - Programming Languages Cloud:** A dynamic programming languages cloud that showcases a random batch of programming languages.
-  **Vis 2 - Geographical distribution:** A map highlighting the global distribution of programming languages.
-  **Vis 3 - Text representation:** A graphical representation of the frequency and prominence of terms associated with different programming languages, and its evolution over time.
-  **Vis 4 - Network visualization:** A diagram illustrating the connections and associations between various programming languages.

In the following sections, we will introduce these four visualizations in detail, discussing our journey towards the final result, and the challenges we encountered along the way.

Vis 1 - Programming Languages Cloud

🦾 Challenges: Our goal is to create a programming languages cloud that provides users with the freedom to explore a multitude of programming languages. However, most of the existing code bases only enable the construction of a static word cloud, which lacks dynamism and interactivity.


🎨 Visualization Choice and Final Website Implementation: We utilize the TagCloud [1] GitHub repository for generating a dynamic 3D text cloud for programming languages. This tool, which is only 6KB minsize and doesn't depend on other libraries, enables us to calculate the next position of each word using factors like mouse movement, initial speed, and direction. These calculations help transform words in a 3D space, creating an illusion of movement.


Moreover, we've enhanced the original code by adding functional buttons, which allow changing batches, pausing and resuming animations. We've diversified the colors of the programming languages and added tooltips to provide more information about each language. The final visualization is presented below.


- Change to a new Batch: Explore another 30 programming languages
 - Pause Animation
 - Resume Animation



Vis 2 - Geographical Distribution

 **Dataset Preprocessing:** We used the data of the "country" attribute in the dataset. To clean the dataset, we may need to standardize country names in the raw data, as they are currently represented in an informal manner. For example, both "USA" and "United State" refer to the "United States". Since a programming language may be co-developed by multiple countries, we distribute a calculated fraction ($1/\text{total contributing countries}$) to each participating nation.

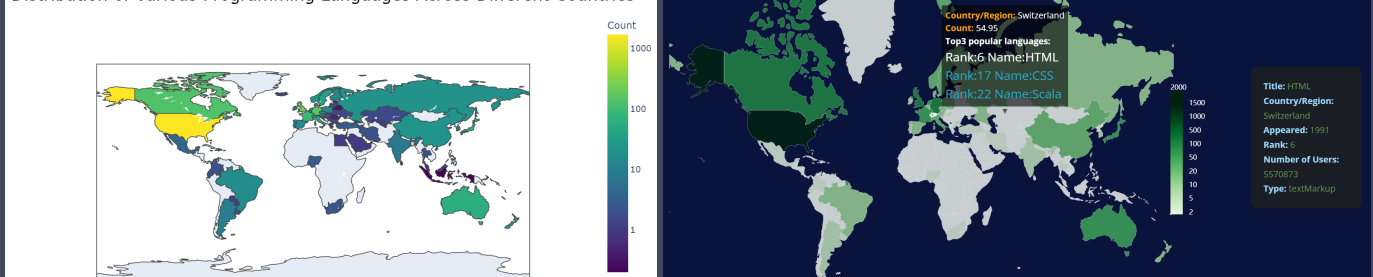
 **First Attempt - Global map:** At first, we created the graph displayed below left. However, this graph does have some limitations. As it is static, users cannot interact with it to glean more information about the distribution. It also doesn't provide an exact count of the programming languages developed in each country, nor does it offer insights into the most popular languages or detailed information about individual languages.

 **Visualization Choice and Final Website Implementation:** Our visualization aims to provide an interactive interface, whereby users can gauge the number of programming languages developed in a specific country through color representation. Clicking on any country reveals detailed information including total languages developed and the top three languages. Users have the option to pin this data for easy reference. These lists also allow users to explore specific details of the top languages, enhancing understanding.

The website implementation uses TopoJSON for rendering the countries' topologies, coupled with the JavaScript-based library D3.js for coloring the map and facilitating interactivity. Guided by valuable inputs from lectures on data visualization, we were able to craft this feature.

In a bid to boost user engagement, we have effectively utilized d3.js to allow the graph to respond to actions such as hovering and clicking. This integration develops an engaging geographical distribution visualization that not only provides an overview of distribution but also lets users explore in-depth information via interactive features. The final product of this effort is our visualization presented below right.

Distribution of Various Programming Languages Across Different Countries



Vis 3 - Text Representation

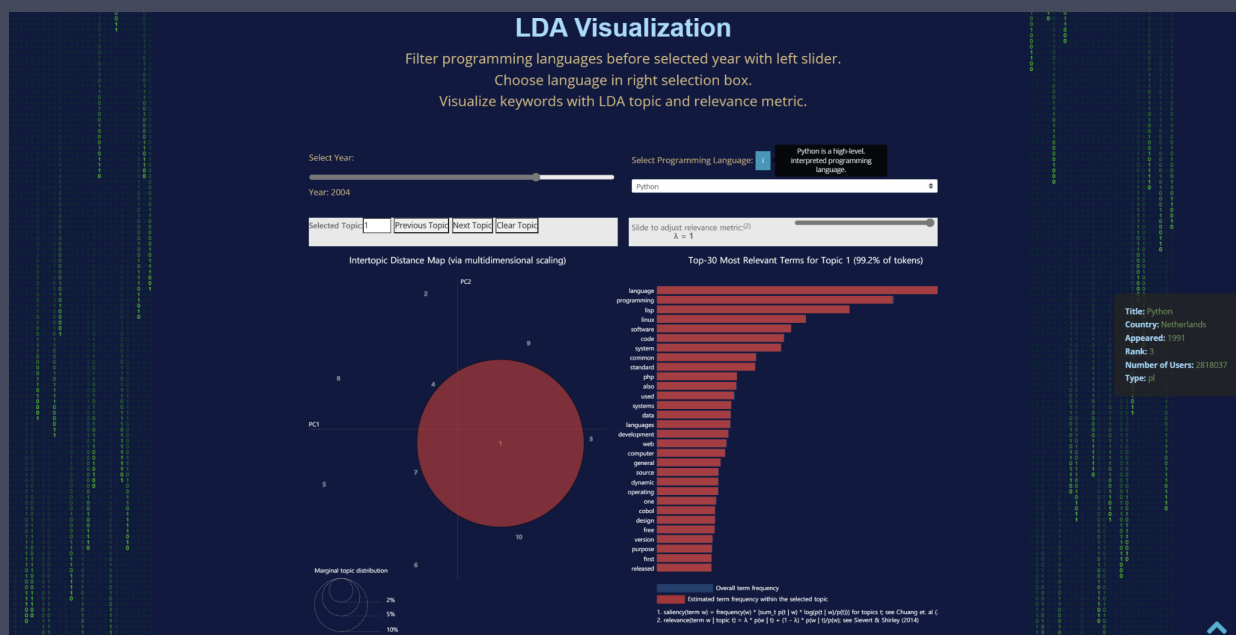
Dataset Preprocessing: We aimed to illustrate crucial programming language terminologies using Wikipedia summaries, focusing on word frequency. However, 66.4% of the languages lacked summaries and existing data contained non-essential content. We cleaned the data, excluding numbers, symbols, and stopwords, and concentrated on the top 100 languages to prevent memory overload.

Visualization Pipeline: We brainstormed how to incorporate the element of time and finally devised a strategy. We incorporated a timeline, allowing viewers to filter programming languages by year and explore specific details and historical progression.


First Attempt - Word Cloud: Initially, we selected word cloud for visualization, revealing the most frequent terms. However, it lacked interactivity, provided limited information, and had inaccurate visual encoding.

Final Decision - LDA: We chose Latent Dirichlet Allocation (LDA), identifying latent topics and presenting topic hierarchy and word distribution. It enabled users to explore topics, view related documents, and customize the visualization.


Website Design: After choosing LDA, we faced challenges in implementing time filtering and language selection. We used JavaScript to update language lists and a CSV table for data access. We used pyLDAvis for LDA visualization, storing LDA representations as JSON. We fixed overlapping visualizations using a debounce function and further enhanced the user experience with a tooltip for additional language information. The final visualization is shown below.

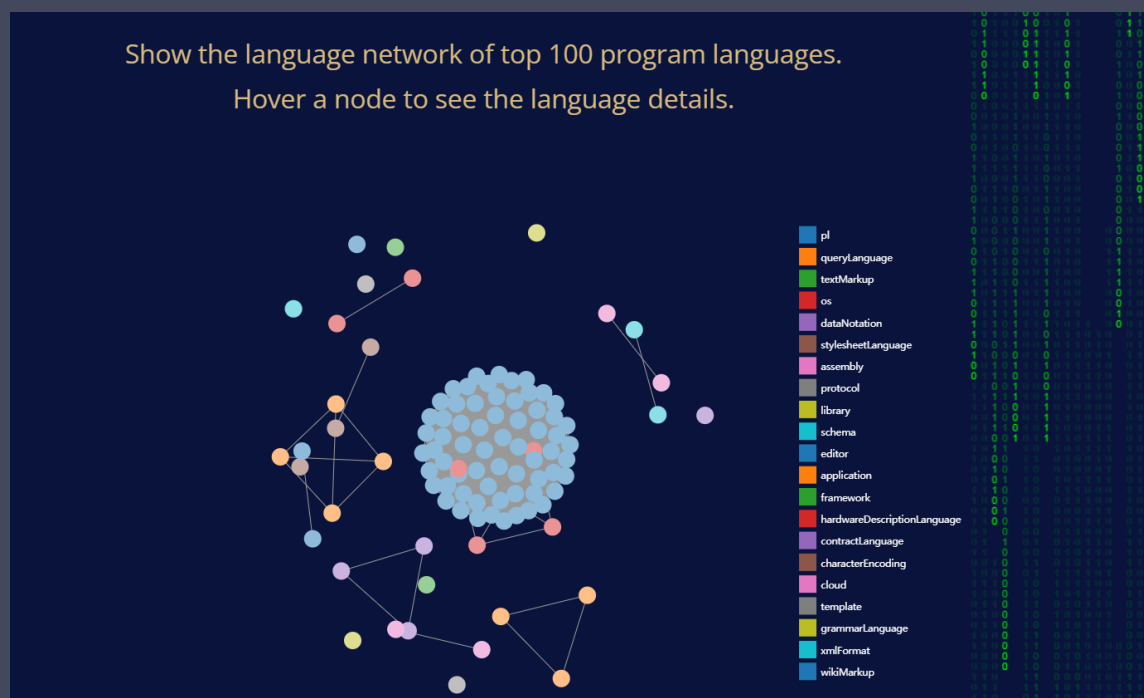


Vis 4 - Network Visualization

 **Dataset Preprocessing:** Facing the challenge of visualizing similarities between programming languages, we decided to build a network to illustrate their closeness. Defining similarity posed the first obstacle due to the predominance of NaN values across the 353 attributes. With 89.2% of attributes containing over 80% NaN values, we opted to utilize the top 100 languages for clarity. Another challenge was defining the edges in the network using relevant attributes. Among 11 attributes with fewer than 20% Null values, one was categorical: language type. Considering the diversity of programming languages, including general-purpose, web, and database types, we decided to use this attribute to define similarity. This meant languages of the same type were linked by an undirected edge.

✓ **Visualization Choice:** Choosing a visualization that could also reveal individual language details proved complex. Drawing from social media community analysis, we conceptualized languages as nodes, with similarity determining their proximity.

 **Website Design:** In the implementation, we used d3.js to create a force-directed graph. We developed an interactive experience with hover tooltips, node dragging, and zoom-in/out functionality. Despite the challenges, we produced an informative visualization to explore the relationships between different programming languages.



Peer assessment

Yiyang: Contributed to discussions regarding the selection of the dataset and aspects of the visualization. Implemented the LDA visualization, including the slider, selection box, and usage of the pyLDAvis library. Integrated the created visualization into the website. Authored the introduction and LDA visualization sections of the process book.

Naisong: Contributed to discussions regarding the selection of the dataset and aspects of the visualization. Implemented network visualization on the website. Authored the introduction and network visualization sections of the process book. Recorded the screencast video.

Xuehan.: Contributed to discussions regarding the selection of the dataset and aspects of the visualization. Implemented visualization of programming languages cloud and geographical distribution, along with corresponding sections in the process book.

All other parts not specifically mentioned have been collaboratively completed by our team

Conclusion

In this project, we explored the various attributes of programming languages using the expansive PLDB dataset. We created four interactive visualizations: a dynamic Programming Languages Cloud, a Geographical Distribution Map, a Text Representation of terminology frequency, and a Network Visualization showing language interconnections.

Our website offers engaging interactions and a wealth of detailed information. Technologies used include JavaScript libraries such as D3.js and TopoJSON, and additional custom components. We aim to provide a fun, engaging, and informative exploration of the world of programming languages.

References

[1] C. Min, H. Wen, F. Mayer, and H. Ahrens, "3d tagcloud," <https://github.com/cong-min/TagCloud>, 2022.