PROGRAMMING LANGUAGE PREFERENCES: A COMPREHENSIVE SURVEY OF DEVELOPER CHOICES AND TRENDS

By

Md. Kofil Uddin

ID: 201-15-3721

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

Shah Md. Tanvir Siddiquee
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2024

APPROVAL

This Project titled "**Programming Language Preferences: A Comprehensive Survey of Developer Choices and Trends**", submitted by Md. Kofil Uddin, ID: 201-15-3721 to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 12.00 p.m. 24 January 2024.

BOARD OF EXAMINERS

Dr. Md. Taimur Ahad (MTA) — **Chairman**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Saiful Islam (SI) — **Internal Examiner 1**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Lamia Rukhsara (LR) — **Internal Examiner 2**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Abu Sayed Md. Mostafizur Rahaman (ASMR) — **External Examiner 1**
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

I hereby declare that, this project has been done by myself under the supervision of **Shah Md. Tanvir Siddiquee, Assistant Professor, Department of CSE** Daffodil International University. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Shah Md. Tanvir Siddiquee
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Kofil Uddin
ID: - 201-15-3721
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, I express my heartiest thanks and gratefulness to almighty God for his divine blessing makes me possible to complete the final year Research project successfully.

I really grateful and wish main profound my indebtedness to **Shah Md. Tanvir Siddiquee, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of my supervisor in the field of "*Research-Based Thesis*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Sheak Rashed Haider Noori, Professor, and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank my entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

This report presents an empirical examination aimed at uncovering the intricate landscape of programming language trends and their implications for software development. By utilizing statistical methods, this study captures historical behaviors and provides a comprehensive understanding of the constantly evolving preferences among developers. This research delves into the intricacies of programming language choices, shedding light on the dynamic nature of trends in software engineering. Potentially, this research paper conducts a thorough examination of the popularity of programming languages using GitHub data, covering aspects like issues, pull requests, and public repositories. The research examines various aspects such as issues, pull requests, and public repositories, and derives insights from data collected from Kaggle. The paper presents experimental results through visualizations that illustrate the popularity of programming languages over the years. Additionally, it compares the findings with other reputable sources of information, such as Stack Overflow, PYPL, TIOBE, and GitHub's Octoverse, to offer a comprehensive perspective. The discussion critically addresses limitations in the research, with a particular focus on primary programming languages, and suggests potential directions for future investigations. Overall, the study provides significant contributions to our understanding of programming language trends, highlighting the crucial role of GitHub in shaping the developer community's preferences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduce the topic

In the realm of software development, the choice of programming languages holds crucial significance, as it has the capacity to mold the industry and sway the preferences of developers. This research embarks on an exhaustive exploration titled "Programming Language Preferences: A Comprehensive Survey of Developer Choices and Trends." In this investigation, we delve into the dynamic realm of programming languages to comprehend the evolving preferences among developers. As technology progresses and the surface of novel languages, decisions undertaken by developers have substantial implications for the effectiveness, scalability, and triumph of software endeavors. The primary objective of this study is to untangle the intricate network of choices made by developers, offer insights into historical patterns, and foresee the trajectory of programming language preferences within the constantly evolving landscape of software development.

1.2 Background Overview

Exploring programming language trends plays a pivotal role in understanding the continuously evolving landscape of software development. Various investigations, exemplified by a survey examining introductory programming courses [1], scrutinize the decisions made by developers and provide insights into the languages integrated into computing courses [2]. Furthermore, large language models (LLMs) are growing, and research has offered an in-depth overview of their architectural and training facets [3]. Valuable perspectives from the Stack Overflow Developer Survey 2023 significantly contribute to the understanding of the developer community and prevalent trends [4]. Broadening the perspective, insights into trends, and trajectories within the software industry further enhances the comprehension of contemporary technological directions [5]. This foundational overview sets the stage for a detailed exploration of the intricate dynamics surrounding programming language preferences and trends in the dynamic realm of software development.

1.3 Research Problem

The world of software development services involves close collaboration between developers and clients, focusing on gathering detailed requirements to guide the entire development process [6]. Once the requirements are established, the design phase comes into play, shaping the overall development trajectory. This dynamic process emphasizes adaptability and responsiveness in the ever-evolving landscape of programming languages. Developers navigate client needs, translating them into effective design strategies that align with the rapidly changing trends and innovations in the software development industry [7]. The success of this collaborative endeavor hinges on the seamless integration of client requirements, design principles, and flexibility to embrace the latest developments in the realm of programming languages.

1.4 Objectives of the Study

i. Observing trends in high-level programming languages:
   Investigate and analyze the trends of high- and low-level programming languages, delving into their evolution and usage patterns for comprehensive insights.

ii. Identifying the optimal programming languages for app development
   Enumerate and discuss the best programming languages for app development considering their advantages and disadvantages. The aim is to guide developers in selecting suitable languages for app projects [8].

iii. Present a brief overview of web development languages:
   Compile a quick snapshot of popular programming languages and frameworks used in web development. This effort is intended to assist in making effective hiring decisions for Web development projects [9].

1.5 Assumption of the Research

This thesis assumes that delving into programming language trends and conducting empirical studies provides crucial insights into the ever-changing realms of software engineering. The study postulates that a focused examination of a concise collection of both high- and low-level programming languages will provide detailed perspectives on their evolutionary paths and usage patterns [6].

Moreover, this assumption extends to the application of empirical research methods, positing that statistical approaches can effectively capture observed behaviors and elucidate historical trends in programming languages. The primary aim is to contribute to the formulation of empirical laws governing the evolution of programming language.

This study also presupposes the value of developing a decision model for programming language ecosystem selection. It foresees that this model will provide comprehensive insights into the intricate process of choosing a programming language ecosystem, enhancing the understanding of the factors influencing this selection [10].

These assumptions collectively form the foundation for exploring programming language trends and their impacts on the broader landscape of software engineering. This study anticipates that the insights gained will play a pivotal role in guiding developers and shaping the future trajectory of programming languages.

1.6 Motivation

The significance of this thesis lies in its exploration of programming language trends, which directly engage with the ever-changing landscape of software development. By integrating empirical studies, employing statistical methods, and introducing a decision model for programming language ecosystems, this study provides a thorough understanding of the ongoing trends. Furthermore, the incorporation of insights from the Stack Overflow Developer Survey 2023 adds a practical dimension, enriching the understanding from a real-world perspective. The paper's focus on large language models (LLMs), a contemporary and vital element in programming language research, underscores their relevance and alignment with current industry dynamics.

1.7 Project Management and Finance

The procedure consists of a set timetable that covers important phases like gathering and preprocessing data, creating the model, training, validating, and assessing it. Project timeline adherence and progress tracking will benefit from well-defined milestones and checkpoints. Furthermore, a diverse group of project managers, medical professionals, and machine learning specialists will collaborate to ensure comprehensive supervision and a smooth rollout. The allocation of resources will be crucial in terms of funding; costs for personnel, computing infrastructure, dataset acquisition, ethical compliance, and expert consultations will all need to be taken into account. The development of effective budgeting and cost management protocols aims to optimize resource allocation while guaranteeing the integrity and quality of project deliverables within budgetary constraints. Project requirements can be adjusted thanks to adaptable allocation procedures and ongoing spending monitoring.

Table 1.1 Project Management Table

Work	**Time**
Data Collection	1 month
Papers and Articles Review	15 days
Experimental Setup	1 month
Implementation	15 days
Report Writing	1 months
Total	4 months

CHAPTER 2

BACKGROUND STUDY

In this section, our goal is to offer valuable insights into pertinent studies, providing a detailed overview of Surveys and Indexes with GitHub. The initial part explores the previous related works with 4 different type of surveys and two web search-based indices, essential for those keen on grasping the platform's capabilities. The subsequent section is vital, offering an in-depth explanation of GitHub, covering repositories, the trending page, and the star system a must-read for a comprehensive understanding.

2.1 Previous Studies

The background investigation for this paper extensively explores prior research, sourcing knowledge from reputable entities in the software development realm. It integrates insights from the Stack Overflow Developer Survey 2023, a thorough examination of developers' preferences and trends. Furthermore, it draws upon the IEEE Spectrum, a distinguished platform providing diverse technological perspectives, enriching the research with valuable insights. The PYPL Popularity of Programming Language Index is employed as a pivotal source, elucidating the popularity dynamics of programming languages through online activities. Additionally, the TIOBE Programming Community Index, a widely acknowledged metric, is employed to scrutinize and compare programming language popularity trends. Through the amalgamation of insights from these esteemed studies and indexes, the background study aspires to construct a robust foundation for the subsequent exploration of programming language preferences and trends in the contemporary software development landscape [11]. 'The Most Popular Programming Languages of GitHub's Trending Repositories' is a research-based thesis that examines and presents the most popular technologies for development among developers [12].

2.1.1 Stack Overflow Developer Survey 2023

In 2023, JavaScript maintained its remarkable streak as the most widely used programming language for the eleventh consecutive year. Python ranks third overall and first for non-professional developers. Bash/Shell, C, Ruby, Perl, and Erlang have ascended [13]. Elixir and Lisp have gained two spots. Lua surged seven positions. The top three technologies among professionals are JavaScript, HTML/CSS, and SQL. However, the landscape shifts for those learning to code, where HTML/CSS and JavaScript are nearly tied as the most popular languages. Student developers favor Python over SQL (59% vs. 37%), whereas professional developers indicate a higher usage of SQL than Python (52% vs. 45%). Notably, those in the learning phase are more inclined towards Java (37% vs. 31%), C++ (32% vs. 20%), and C (32% vs. 17%) compared to their professional counterparts [14].

Figure 2.1 Most popular Technologies

2.1.2 IEEE Spectrum

In the 10th edition of IEEE Spectrum's Top Programming Languages (TPL) *Figure 2.2*, our approach to compiling the rankings has evolved while retaining the fundamental objective: amalgamating various popularity metrics to create a ranking system tailored to diverse reader needs. This year, Python not only maintains its top position in the general "Spectrum" ranking, tailored to typical IEEE members' interests, but also extends its lead. Python's growing dominance comes at the expense of smaller, specialized languages, positioning it as a versatile language, excelling particularly in AI with its robust libraries. Despite the waning impact of Moore's Law on high-end computing, low-end microcontrollers still benefit, enabling Python to thrive in embedded development on a $0.70 CPU. Python is solidifying its long-term position, serving as the entry point for many children and teens into programming, allowing seamless progression to advanced domains and potential employment opportunities [15].

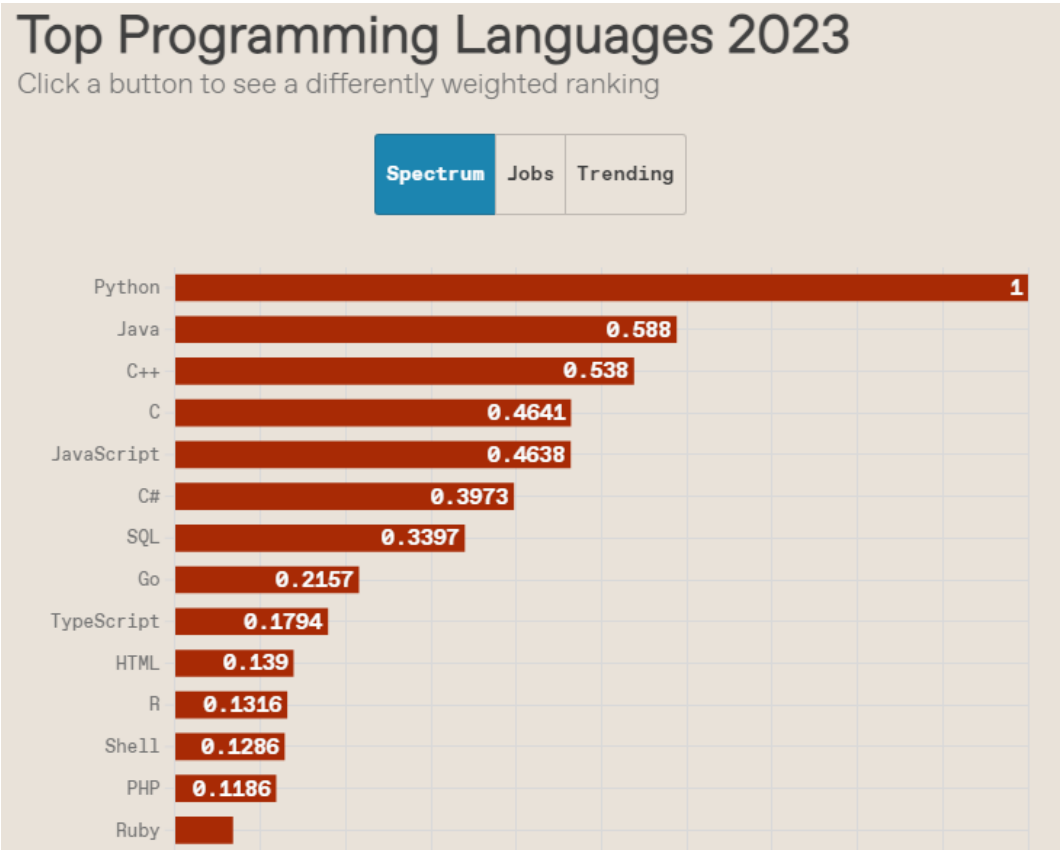

Figure 2.2 Top Programming Languages 2023

2.1.3 The PYPL popularity of Programming Language Index

The PYPL Popularity of Programming Language Index (*Figure 2.3*) evaluates the popularity of programming languages by examining the frequency of searches for language tutorials on Google. It is widely recognized as an important gauge that helps people choose a language for their studies or software projects. At present, Python holds the highest rank worldwide, experiencing significant growth in the past five years, while the popularity of Java has decreased. This index employs a logarithmic scale to offer valuable information on language popularity in different countries. The data source for this index is Google Trends [16].

Worldwide, Jan 2024 :

Rank	Change	Language	Share	1-year trend
1		Python	28.2 %	+0.5 %
2		Java	15.73 %	-0.9 %
3		JavaScript	8.91 %	-0.6 %
4	↑	C/C++	6.8 %	-0.0 %
5	↓	C#	6.67 %	-0.3 %
6	↑	R	4.59 %	+0.6 %
7	↓	PHP	4.54 %	-0.7 %
8		TypeScript	2.92 %	+0.2 %
9		Swift	2.77 %	+0.6 %
10		Objective-C	2.34 %	+0.2 %
11	↑	Rust	2.19 %	+0.3 %
12	↓	Go	2.02 %	+0.1 %
13		Kotlin	1.78 %	-0.0 %

Figure 2.3 PYPL Popularity of Programming Language

2.1.4 TIOBE Programming Community Index

C# secured the title of Programming Language of the Year in the TIOBE index [17], marking a historic achievement. Its two-decade presence in the top 10 has led to substantial growth, with a notable 1.43% increase in a year, surpassing other contenders like Scratch and Fortran. C# is gaining ground on Java, especially in web application backends and games due to its compatibility with Unity. Free to use and continually evolving, C# is poised to potentially surpass Java soon. TIOBE also saw Fortran and Kotlin enter the top 20, replacing R and Perl. Looking forward, Dart (with Flutter) and TypeScript are potential candidates for the TIOBE index's top 20 in 2024, with TypeScript gaining industry traction despite its current TIOBE ranking. The dynamics of the language landscape make 2024 an intriguing year to watch.

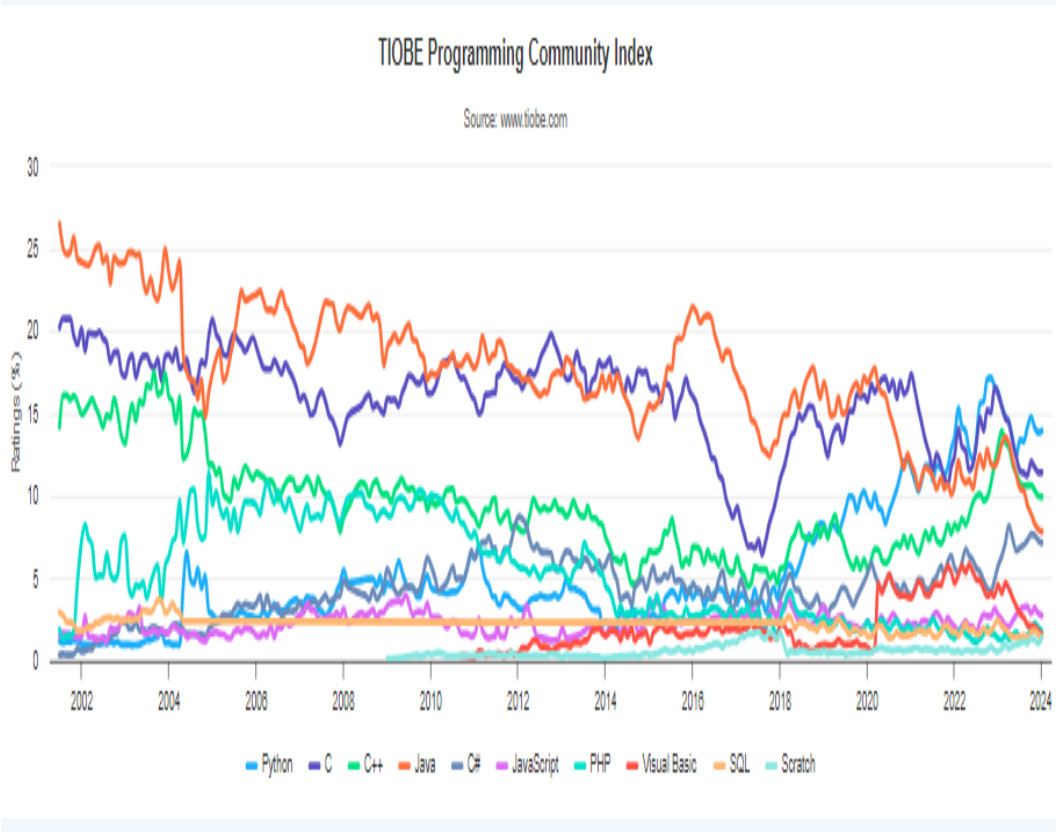

Figure 2.4 TIOBE Index for January 2024

Table 2.1 Others Programming Languages

Position	Programming Language	Ratings
21	D	0.77%
22	F#	0.77%
23	R	0.74%
24	SAS	0.70%
25	(Visual) FoxPro	0.67%
26	Ada	0.62%
27	Classic Visual Basic	0.60%
28	Prolog	0.56%
29	VBScript	0.55%
30	Perl	0.52%
31	Objective-C	0.46%
32	Dart	0.43%
33	Julia	0.40%
34	X++	0.39%

2.2 GitHub The State of the Octoverse

In 2022, a multitude of primary languages, reaching nearly 500 in number, were utilized by developers on GitHub to fashion software. This dynamic linguistic landscape among developers accentuates significant transformations in the methods and categories of software creation.

Python, exhibiting resolute tenacity, has maintained its prominent position as the second most widely used language, owing largely to its adaptability in diverse domains such as development, education, machine learning, and data science, although JavaScript continues to reign supreme. Noteworthy, TypeScript has also retained its steadfast status as the fourth most utilized language, persisting from year to year. A conspicuous observation is the descent of PHP from the sixth to the seventh-place ranking in 2022 [18].

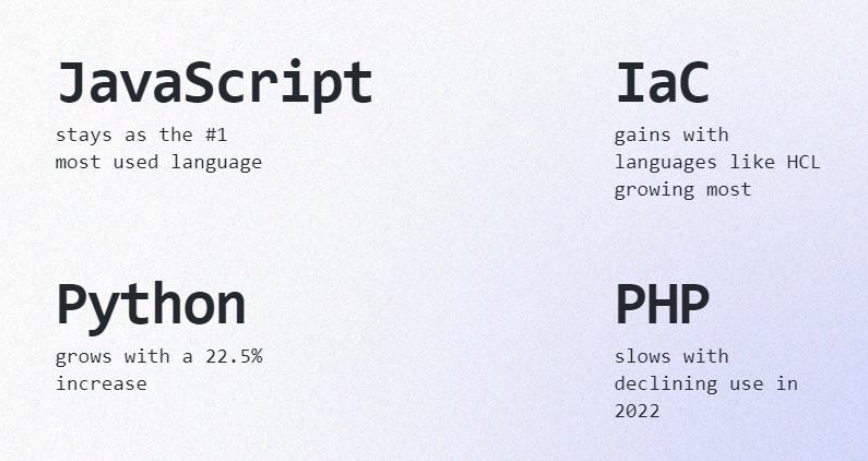

Figure 2.5 The top programming languages

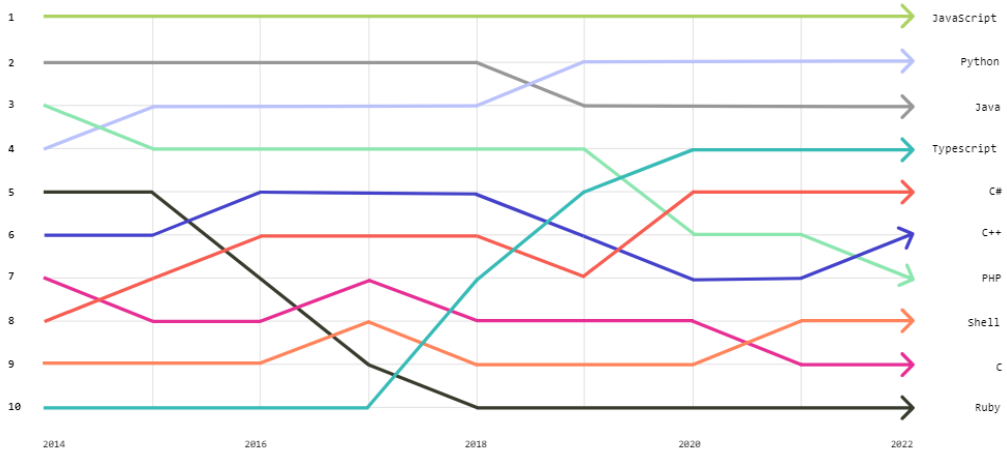

Figure 2.6 Top languages used in 2022

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Collecting Data from Kaggle

We gathered information from predefined data Data Visualization on Github Languages Data (GitHub issues, prs, and repositories) for GitHub's three distinct types of analysis and result determination [19].

3.2 Contents of Data

A good way to measure the popularity of a programming language is to see how many projects and files are created using it. GitHub is a popular platform for sharing and collaborating on code, so looking at the languages used in its repositories, pull requests (PRs), and issues can give us an idea of which ones are popular. This dataset provides statistics on the programming languages used in GitHub repositories, PRs, and issues from 2011 to 2023.

Developers employed almost 500 primary languages in the year 2023 to fabricate software on GitHub. The altering languages utilized by developers highlight notable modifications in the procedures and genres of software being produced. Python has upheld its position as the runner-up throughout the previous year, predominantly owing to its adaptability in various domains, such as development, education, machine learning, and data science, even though JavaScript continues to be the undisputed victor.

TypeScript also remained steady in fourth place, maintaining its position from the previous year. It is worth mentioning that PHP descended from sixth to seventh place in 2023.

3.3 Original Source of Data

Using BigQuery [20], a robust tool for data analysis, we successfully collected and merged significant data from two distinct datasets known as "github_repos" and "githubarchive". The datasets provided comprise a diverse array of openly accessible data obtained from GitHub [21], a commonly utilized internet platform dedicated to software development endeavors. Through the utilization of advanced querying techniques, I compiled the relevant data into an extensive dataset to facilitate subsequent analysis and investigation.

3.4 Limitations of the Data

The general public can only view data specifically related to public GitHub repositories, along with their corresponding pull requests and issues. As a result, this dataset solely relies on information from public repositories, leading to potential limitations in its ability to accurately represent the entirety of repositories hosted on GitHub.

3.5 Analyzing Data

The data-analysis tools of Python were employed to examine the data obtained from the CSV files, which were generated during the process of scraping the newsletters. These tools encompass Pandas, Matplotlib, WordCloud, seaborn, and the Google CoLab environment [22]. The comprehensive analysis and visualization can be located in the APPENDIX. In the majority of the analysis, the repositories falling under the "New Repositories" category were eliminated. This course of action was taken because this category solely consists of repositories that were made accessible to the public on the day the newsletter was issued. To gain a deeper understanding of these repositories, a random sample of distinct repositories was selected and meticulously examined. Each repository in the sample was visited and classified based on its current contents.

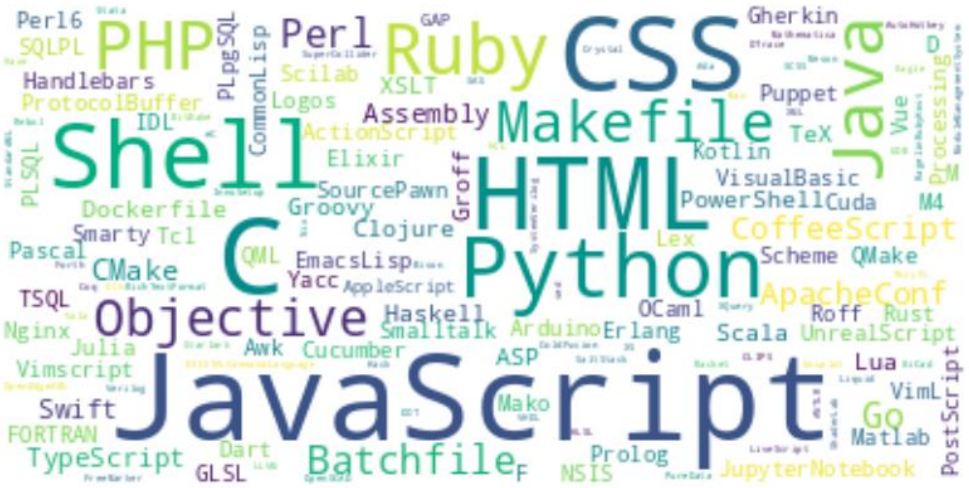

Figure 3.1 Most Popular Programming Languages

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 Issues Data Visualization and Analysis

After analysing the number of Issues in GitHub related to the most popular programming languages, we found that no new language managed to break into the top 5 for any given year, apart from those already ranked in the top 10. While there were a few new languages that entered the top 10 in certain years, they didn't have enough impact to rival the most widely used languages.

	name	count	count//1000
0	JavaScript	6500226	6500
1	Python	4192481	4192
2	Java	3536070	3536
3	PHP	2240585	2240
4	C++	1941004	1941
5	TypeScript	1741593	1741
6	Go	1645358	1645
7	C#	1407859	1407
8	Ruby	1292627	1292
9	C	1003056	1003

Figure 4.1 Most used Languages

	year	name	count	total
1	2011	C	1276	0.048938
2	2011	C#	888	0.034057
3	2011	C++	2280	0.087443
10	2011	Java	2228	0.085449
11	2011	JavaScript	4697	0.180141
13	2011	PHP	3401	0.130436
14	2011	Python	3764	0.144358
15	2011	Ruby	5326	0.204265
23	2012	C	16423	0.033993
24	2012	C#	12657	0.026198

Figure 4.2 Absolute count and year wise relative count

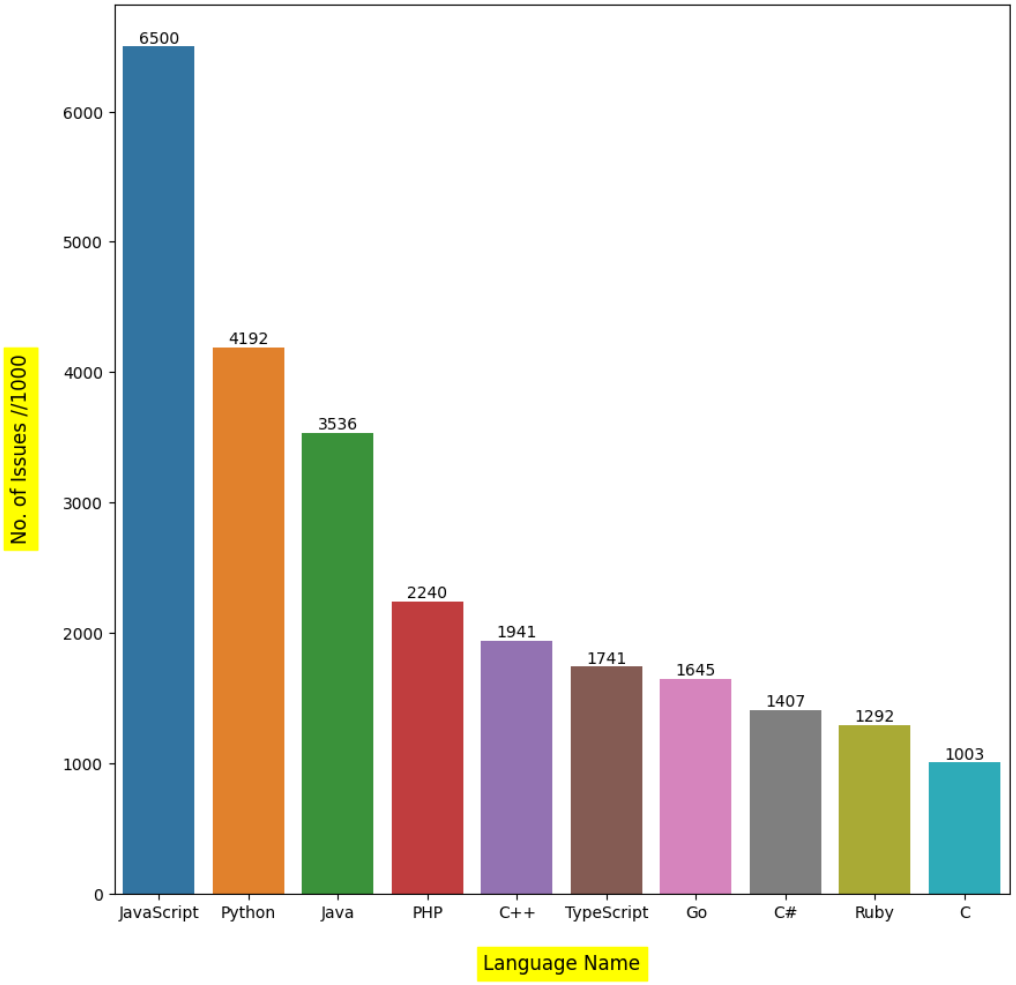

Figure 4.3 The overall trends for popular languages over the past 10 years

4.1.1 An analysis of the popular languages based on their popularity by year

To analyze popular programming languages by year, Popular languages include C, C++, C#, Python, Java, JavaScript, C++, and PHP, but their popularity varies by year and industry. This analysis provides insights into language trends and is useful for developers and businesses.

i. Absolute Value Trends

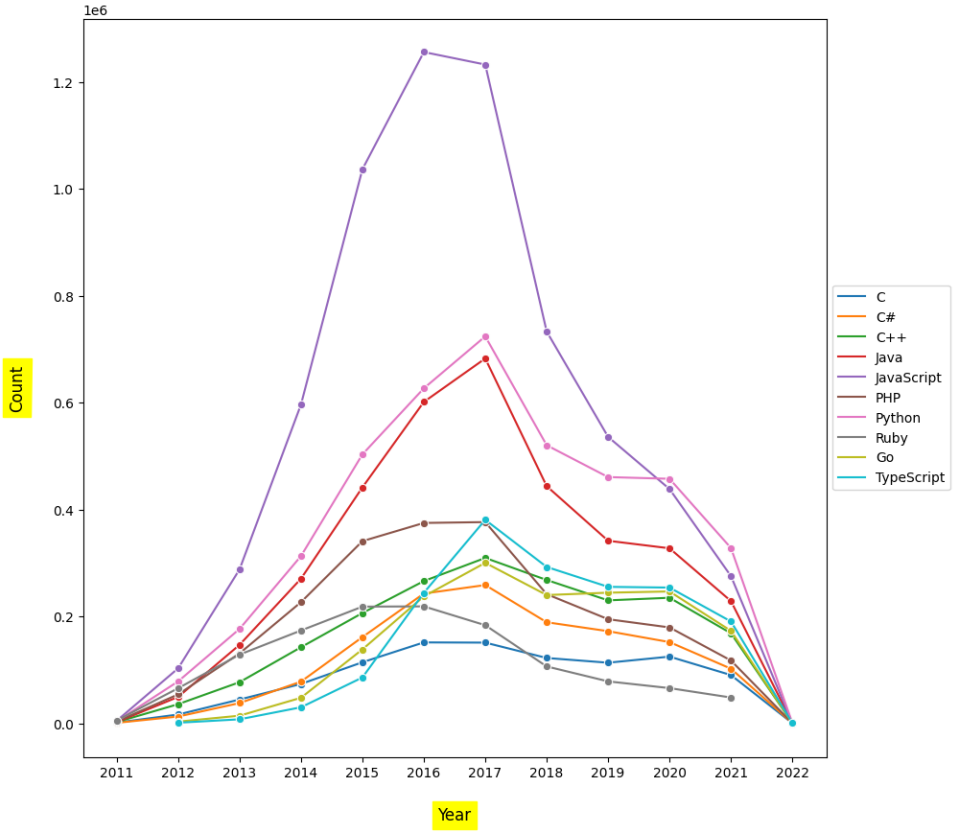

Figure 4.4 Absolute value trends year wise

ii. Relative Value Trends

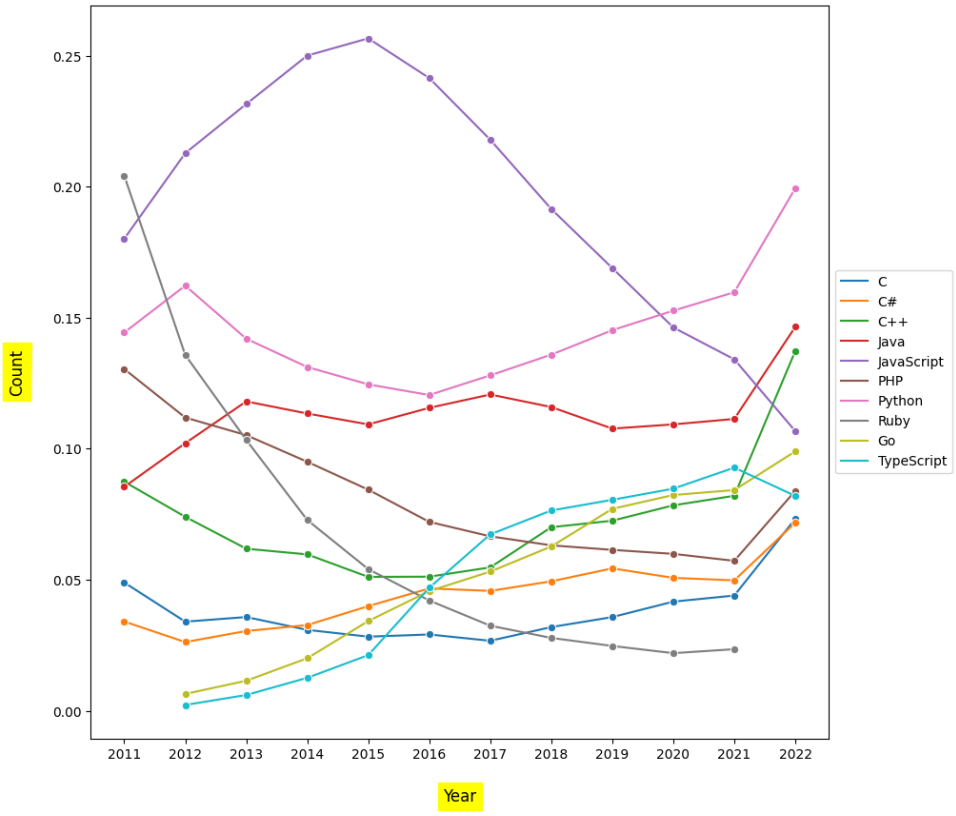

Figure 4.5 Relative value trends year wise

When it comes to the features available on GitHub, JavaScript has consistently been the language favored by developers. However, lately Python has gained more popularity and surpassed JavaScript in this regard. Additionally, TypeScript has been steadily advancing and growing in usage, while Ruby and PHP have experienced a decline in popularity.

4.2 Pull Requests Data Visualization and Analysis

Pull request analysis the data with name-count pairs, languages considered in the top 10 did not appear in the top 5 any year except for the languages considered in the top 5. While there were some new entries in the top 10 each year, the number wasn't significant when considering the most popular languages. There are more than 750k pull requests for JavaScript, and there are more than 690k requests for Python, which is the second most popular. Java and PHP are next on the list with 373k+ and 247k+ pull requests respectively. This is followed by C++ with 214k+ pull requests.

	name	count	count//1000
0	JavaScript	7512246	7512
1	Python	5904273	5904
2	Java	3731712	3731
3	Ruby	2887567	2887
4	PHP	2474738	2474
5	Go	2401902	2401
6	C++	2145920	2145
7	TypeScript	1994032	1994
8	C#	1181788	1181
9	HTML	1180280	1180

Figure 4.6 Languages vs Pull Requests

	year	name	count	total
1	2011	C#	291	0.018740
2	2011	C++	1012	0.065173
4	2011	HTML	149	0.009596
6	2011	Java	1200	0.077280
7	2011	JavaScript	2184	0.140649
9	2011	PHP	2745	0.176777
10	2011	Python	2543	0.163769
11	2011	Ruby	3679	0.236927
15	2012	C#	7650	0.017776
16	2012	C++	20120	0.046753

Figure 4.7 Absolute count and year wise relative count

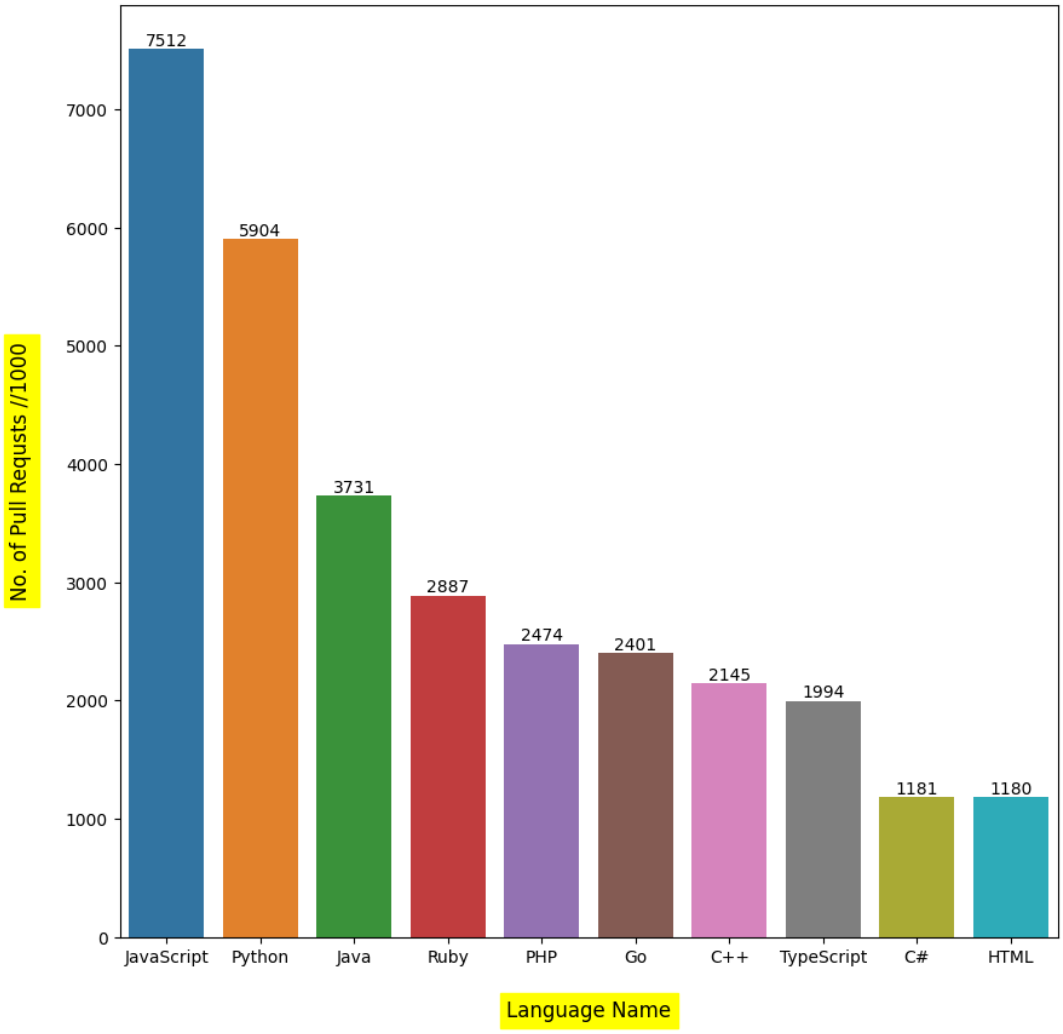

Figure 4.8 Absolute value trends overall

i. Relative Value Trends

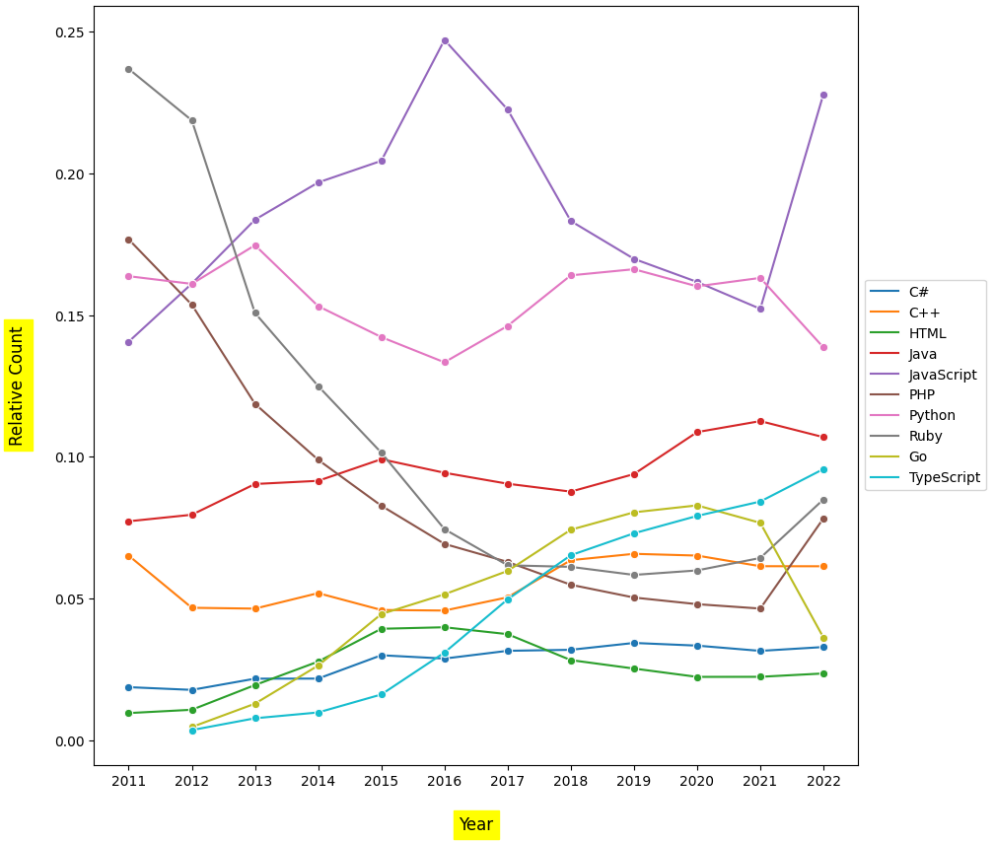

Figure 4.9 Relative Value Trends

ii. Absolute Value Trends

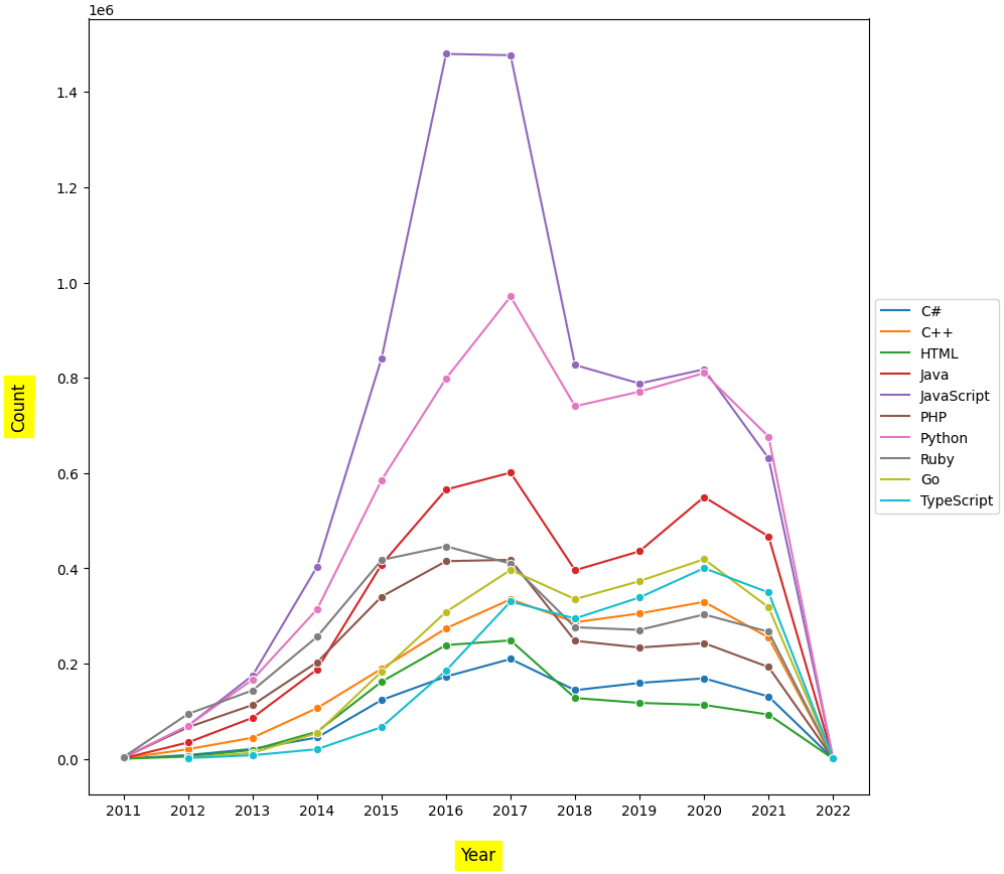

Figure 4.10 Absolute Value Trends

JavaScript has been the most popular language when it comes to pull request feature in GitHub, but recently Python has falling in GitHub. TypeScript and Ruby (maybe even PHP can be considered) have also been steadily improving while GO and Java have been declining

4.3 Public Repositories Data Visualization and Analysis

GitHub has witnessed the creation of numerous repositories dedicated to novel programming languages. The vast array includes an impressive 110k repositories for JavaScript, over 81k for CSS, and more than 77k for HTML files. Python and C++ have also gained popularity on this platform, boasting 54k+ and 27k+ repositories respectively. Additionally, GitHub is home to a substantial number of repositories for other languages such as Ruby, Perl, and Scala.

	language	num_repos	num_repos//1000
0	JavaScript	1100421	1100
1	CSS	813443	813
2	HTML	779549	779
3	Shell	638068	638
4	Python	548870	548
5	Ruby	374802	374
6	Java	369282	369
7	PHP	339901	339
8	C	292000	292
9	C++	278066	278

Figure 4.11 Public Repositories

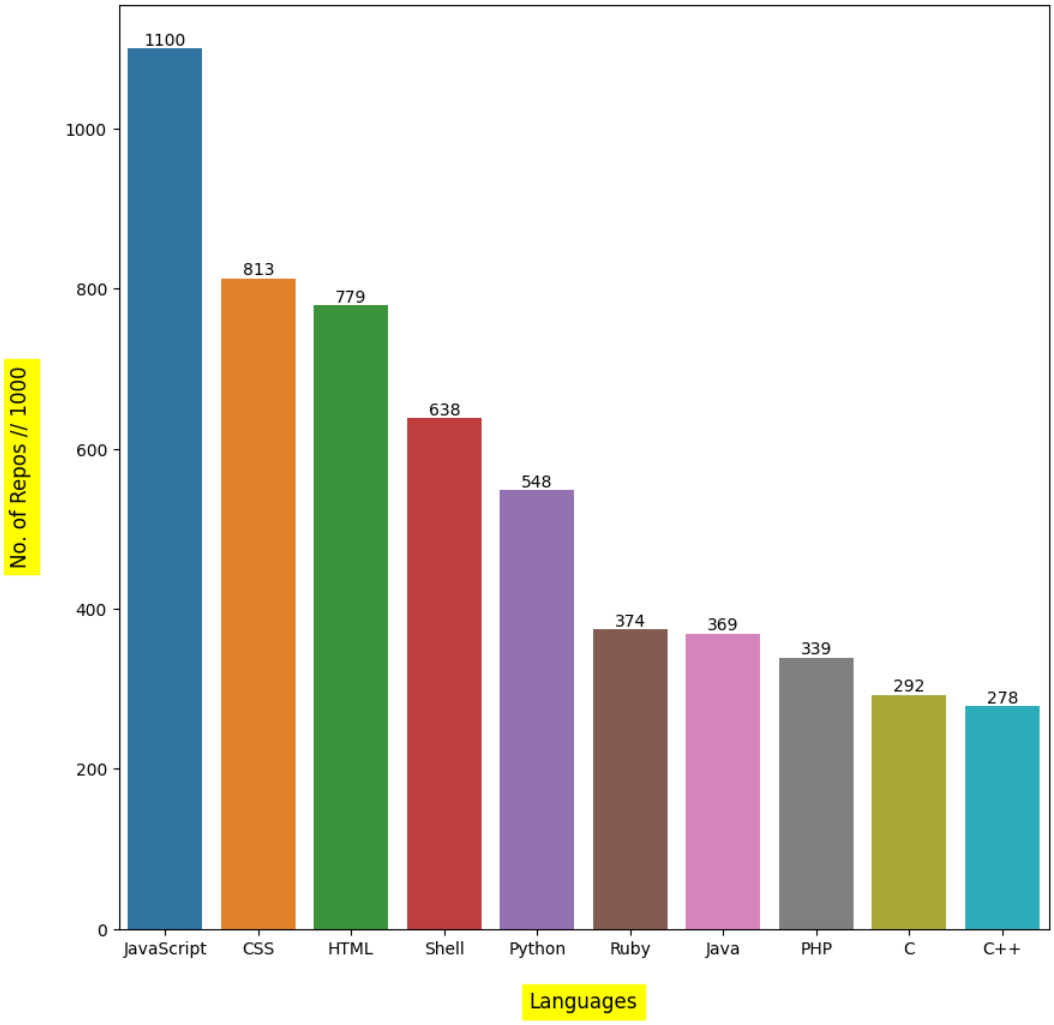

Figure 4.12 Absolute Value trend overall

GitHub Public Repositories data only shows us the language count versus the overall count, so we cannot find out what trends have occurred year over year. As a result, JavaScript, CSS, and HTML are the top 3 languages for Public Responsibilities.

CHAPTER 5

DISCUSSION

5.1 Comparison With Other Indexes

5.1.1 Stack Overflow

Stack Overflow's Developer Survey is a valuable resource for understanding programming language trends. The 2023 Developer Survey (Figure 2.1) collected global responses, providing insights into the preferences of Stack Overflow users. Based on the survey findings, Python has secured the top position as the most favored programming language. Following closely are Java, JavaScript, and C#.

This aligns with the PYPL index, which also highlights the dominance of Python in recent programming language preferences. The survey further compares technologies across different groups, including all respondents, professional developers, and specific demographics, offering a nuanced view of language popularity among diverse developer communities.

While Stack Overflow's survey provides valuable insights, it's essential to consider other indices, such as TIOBE and PYPL, to gain a comprehensive understanding of programming language trends. Each index employs distinct methodologies, offering unique perspectives on language popularity and usage.

Combining insights from multiple sources ensures a more robust evaluation of programming language preferences in the developer community.

5.1.2 PYPL Index

The *Figure 2.3 PYPL Popularity of Programming Language* is a useful tool for determining popular programming languages, deciding which ones to learn, and selecting those to apply to a new software project. However, it's important to note that their results may differ from ours. According to PYPL, Java was the top programming language in 2015, but our data shows that JavaScript is the most widely used language. It's also worth mentioning that Python overtook JavaScript in mid-2018 and had a share of around 30% by the start of 2024. JavaScript, C/C++, and PHP appear to be relatively stable in their popularity, at around ten percent each. However, PYPL suggests that both PHP and C/C++ are declining in popularity.

5.1.3 TIOBE Index

To acquire a more comprehensive comprehension, let us turn to the TIOBE index (Figure 2.*4*) within the relevant timeframe of roughly 2015 to 2020. This vantage point diverges somewhat from PYPL's depiction. TIOBE employs a distinct methodology, evaluating the number of inquiries that various search engines receive for the term "[language] + programming" rather than simply tallying searches. While PYPL assesses interest through language tutorial searches, TIOBE concentrates on the presence of this query on websites.

The outcomes reveal that Java reached a zenith of approximately 22% around 2016 but declined to about 13% in 2018. Although its popularity demonstrated a gradual recovery thereafter, it never fully rebounded. Significantly, C closely mirrors Java's patterns in both decline and growth. C++ and Python maintained a consistent presence at around a five percent margin until 2018 when both observed an upswing in popularity. However, in 2019, there was a notable decline in C++ popularity, while Python witnessed a surge akin to C++'s descent. By 2020, the foremost rankings according to TIOBE will be C, Java, Python, and C++.

5.1.4 GitHub - The State of the Octoverse

An insightful reference for comparison lies in GitHub's annual report, "The State of the Octoverse"(Figure 2.6). A comprehensive study, conducted annually by GitHub, delves into statistics about their platform. Our study finds resonance in their evaluation of the most utilized programming languages from 2014 to 2019. Notably, their ranking methodology differs, as they assess languages based on the number of unique collaborators across both public and private repositories. This distinction is crucial, considering our focus on public repos only. The alignment between their findings and ours is noteworthy, with JavaScript consistently holding the top position throughout the entire duration. Subsequently, the sequence of Python, Java, and PHP follows, demonstrating relative stability over time, albeit with occasional fluctuations in their respective rankings.

5.1.5 Overall Remarks

When contrasting our study with other indexes, it becomes evident that PHP is notably absent from our top results, despite its popularity in alternative indices. This divergence could be attributed to PHP's historical association as an older language for web development. In contemporary scenarios, JavaScript frequently supersedes PHP in new projects. The prevalence of existing codebases in PHP may contribute to its continued popularity, contrasting with our focus on currently trending and likely newer projects.

Our findings showcase projects currently in vogue, potentially reflecting a shift towards newer technologies. In contrast, other indexes incorporate programming languages prevalent in older projects, contributing to the varied results observed.

An interesting trend emerges when examining programming languages for web applications, including JavaScript, HTML, CSS, and TypeScript. These languages appear more prominently in our results compared to broader indexes. This aligns with findings from prior studies on popular repositories, suggesting a potential inclination for web applications to garner attention on GitHub. Alternatively, it raises the possibility that a substantial portion of GitHub projects pertains to web applications.

5.2 Research Limitations

5.2.1 Primary Programming Languages

Our findings are contingent on the identification of the primary programming language within each repository, excluding consideration of all languages utilized in a project. This methodology could yield divergent outcomes if a comprehensive assessment of all programming languages used were undertaken. GitHub's classification criteria for determining the primary programming language exhibit a broad scope. Consequently, our results encompass classifications that may not conventionally be acknowledged as programming languages. Noteworthy examples include HTML, and CSS, which, according to GitHub's classification, contribute to our results. This broader perspective emphasizes the varied nature of projects hosted on GitHub, ranging from traditional programming languages to platforms and tools integral to diverse development endeavors.

5.2.2 Future Research

The dataset we have collected can be used to explore questions that go beyond what we have studied so far. It would be really interesting to analyze how long repositories tend to remain popular. Unfortunately, we weren't able to delve deeper into the repositories by using GitHub's API to gather more detailed information [23].

CHAPTER 6

CONCLUSION

Our study aligns with established research on prevalent programming languages, showcasing the dominance of Java, Python, JavaScript, and C++. This consistency is affirmed across comparable studies, while discrepancies with the TIOBE index are explained by differing measurement methods.

GitHub's trending page reflects broader language trends, emphasizing the significance of Java, Python, and JavaScript. Notably, our findings spotlight additional web development languages like HTML, CSS, and TypeScript, indicating a distinctive focus in GitHub's trending projects. JavaScript emerges as the prominent language in web development in our study and maintains consistent popularity across various research, signifying its prevalence in GitHub's trending projects, particularly in web development. Despite GitHub's primary role in hosting software projects, a noteworthy percentage (nearly 18% over the past five years) of the most popular repositories lack any code. This proportion has steadily increased since 2016, reaching around 25% in 2020. This shift suggests a broader utilization of GitHub beyond software development, with a marked presence of computer science and technology-related projects.

The culmination of this study marks a significant contribution to understanding the dynamics of programming language popularity and its impact on software development. This conclusion encapsulates the key findings and implications derived from the extensive exploration undertaken.

6.1 Recapitulation of Study Goals

This research significantly advances our comprehension of the dynamic landscape of programming language popularity and its implications for software development. The concluding section consolidates pivotal discoveries and ramifications obtained from the extensive exploration conducted.

6.2 Insights from Methodology

Thorough scrutiny and analysis have successfully met the outlined study objectives. The investigation navigated the multifaceted terrain of programming languages, utilizing diverse indices, such as Stack Overflow Developer Survey, IEEE Spectrum, PYPL, TIOBE, and GitHub's Octoverse, to glean comprehensive insights.

6.3 Methodological Effectiveness

The research methodology, centered on data acquisition from Kaggle, has demonstrated efficacy. Rigorous consideration of the data's original sources, coupled with an acknowledgment of limitations, was complemented by a meticulous analytical approach.

6.4 Unveiling Trends through Results and Visualizations

The presentation of experimental results in Sections 4, alongside their corresponding visualizations, has bestowed a nuanced comprehension of popular languages, issues, pull requests, and public repositories. Noteworthy is the illumination of evolving trends in language popularity over the years.

6.5 Valuable Comparative Analysis

DISCUSSION provides a valuable comparative analysis, contrasting findings with varied indexes such as Stack Overflow, PYPL, TIOBE, and GitHub's Octoverse. Candid discussions on research limitations, particularly concerning primary programming languages, pave the way for future investigations.

6.6 Charting Future Trajectories

The identified limitations act as a catalyst for future research, with Section 5.2 outlining potential avenues for exploring primary programming languages and suggesting directions to refine and expand upon this study.

In summary, this research delivers a comprehensive panorama of programming language popularity, drawing insights from diverse sources and methodologies. The outcomes enhance our understanding of language trends, establishing a solid foundation for sustained exploration in this ever-evolving domain.

APPENDIX

By using Google Colaboratory:

#Importing Libraries

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/Thesis/Appendices /Data Set'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import matplotlib.pyplot as plt
from wordcloud import WordCloud
import matplotlib.ticker as ticker
import seaborn as sns
%matplotlib inline
```

Most Popular Programming Languages

```
text = ""
for i, lan in enumerate(repos.language):
    text = "".join([text,("".join(lan.split()) + " ")*repos.num_repos[i]])
word_cloud = WordCloud(collocations = False,max_font_size=50, background_color="white").generate(text)
plt.figure(figsize=(10,5))
plt.imshow(word_cloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

```
issues.head()
```

```
prs.head()
```

```
repos.head()
```

Issues Data Visualization and Analysis

```
#Language vs Issues Count
df = pd.DataFrame(issues.groupby('name')['count'].sum().nlargest(10))
df['count//1000'] = df['count']//1000
df.reset_index(level=0, inplace=True)
df
```

```
# # Verifying whether there has been a sharp increase in the use of any particular language (we are only looking at five per year to keep things simple).
merged = pd.DataFrame()
for i in [2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023]:
    rr = pd.DataFrame(issues[issues['year']==i].groupby(['name','year'])['count'].sum().nlargest(5))
    rr.reset_index(level=0, inplace=True)
    merged = pd.concat([merged,rr],axis = 'index')
```

```
# Year vs Count
df_year = pd.DataFrame(issues.groupby('year')['count'].sum())
df_year.reset_index(level=0, inplace=True)
df_year.head()
```

```
# Absolute count and yearwise relative count( count for the language/total count for that year) have been mentioned for top 10 languages
df1 = pd.DataFrame(issues.groupby(['year','name'])['count'].sum())
```

```
df1.reset_index(level=[0,1], inplace=True)
lst = ['JavaScript','Java','Python','C++','PHP','Go','TypeScript','C#','Ruby','C']
total = []
for i in range(len(df_year['year'])):
    for j in df1.loc[df1['year']== df_year.loc[i,'year']]['count']:
        total.append(j/df_year.loc[i,'count'])
df1['total'] = total
df2 = df1.loc[df1['name'].isin(lst)]

df2.head(10)
```

```
#Absolute Value trends for overall case
plt.figure(figsize = (10,10))
ax = sns.barplot(x = 'name', y = 'count//1000',data = df)
ax.bar_label(ax.containers[0])
ax.set_ylabel('No. of Issues //1000', size = 'large', backgroundcolor = 'yellow',labelpad = 20)
ax.set_xlabel('Language Name', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

An analysis of the popular languages based on their popularity by year

Absolute Value Trends

```
# Absolute value trends yearwise
plt.figure(figsize = (10,10))
ax = plt.subplot(111)
sns.lineplot(x = 'year', y = 'count', hue = 'name',data = df2,marker = 'o')
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
tick_spacing = 1
ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
ax.set_ylabel('Count', size = 'large', backgroundcolor = 'yellow',labelpad = 20)
ax.set_xlabel('Year', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

Relative Value Trends

```
# Relative value trends yearwise
plt.figure(figsize = (10,10))
ax = plt.subplot(111)
sns.lineplot(x = 'year', y = 'total', hue = 'name',data = df2,marker = 'o')
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
tick_spacing = 1
ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
ax.set_ylabel('Count', size = 'large', backgroundcolor = 'yellow',labelpad = 20)
ax.set_xlabel('Year', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

When it comes to Issues features in GitHub, JavaScript has been the most popular language, but Python has recently overtaken it. Similarly, TypeScript has improved steadily, while Ruby and PHP have declined.

Pull Requests Data Visualization and Analysis

```
# Languages vs Pull Requests data
df3 = pd.DataFrame(prs.groupby('name')['count'].sum().nlargest(10))
df3.reset_index(level=0, inplace=True)
df3['count//1000'] = df3['count']//1000
df3
```

```
#Year vs Pull Requests count data
df_year_1 = pd.DataFrame(prs.groupby('year')['count'].sum())
df_year_1.reset_index(level=0, inplace=True)
df_year_1.head()
```

```
# Checking to see if there was a sudden rise in popularity of any one language (only considering 5 every year here to not make it too complicated)
merged = pd.DataFrame()
for i in [2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021]:
    rr = pd.DataFrame(issues[prs['year']==i].groupby(['name','year'])['count'].sum().nlargest(5))
    rr.reset_index(level=0, inplace=True)
    merged = pd.concat([merged,rr],axis = 'index')
#merged (commented out to improve readability)
```


```
# Checking to see if there was a sudden rise in popularity of any one language (only considering 5 every year here to not make it too complicated)
merged = pd.DataFrame()
for i in [2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021]:
    rr = pd.DataFrame(issues[prs['year']==i].groupby(['name','year'])['count'].sum().nlargest(5))
    rr.reset_index(level=0, inplace=True)
    merged = pd.concat([merged,rr],axis = 'index')
#merged (commented out to improve readability)
```


```
# Absolute count and yearwise relative count( count for the language/total count for that year) have been mentioned for top 10 languages
df4 = pd.DataFrame(prs.groupby(['year','name'])['count'].sum())
df4.reset_index(level=[0,1], inplace=True)
lst1 = ['JavaScript','Java','Python','Ruby','PHP','Go','C++','TypeScript','C#','HTML']
total = []
for i in range(len(df_year['year'])):
    for j in df4.loc[df4['year']== df_year_1.loc[i,'year']]['count']:
        total.append(j/df_year_1.loc[i,'count'])
df4['total'] = total
df5 = df4.loc[df4['name'].isin(lst1)]
df5.head(10)
```

```
# Absolute value trends overall
plt.figure(figsize = (10,10))
ax = sns.barplot(x = 'name', y = 'count//1000',data = df3)
ax.bar_label(ax.containers[0])
ax.set_ylabel('No. of Pull Requsts //1000', size = 'large', backgroundcolor = 'yellow',labelpad = 20)
ax.set_xlabel('Language Name', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

Observing the annual trend for the most popular languages

Absolute Value Trends

```
# Absolute value trends yearwise
plt.figure(figsize = (10,10))
ax = plt.subplot(111)
sns.lineplot(x = 'year', y = 'count', hue = 'name',data = df5,marker= 'o')
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
tick_spacing = 1
ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
ax.set_ylabel('Count', size = 'large', backgroundcolor = 'yellow',labelpad = 20)
ax.set_xlabel('Year', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

Relative Value Trends

```
# Relative value trends yearwise
plt.figure(figsize = (10,10))
ax = plt.subplot(111)
sns.lineplot(x = 'year', y = 'total', hue = 'name',data = df5,marker = 'o')
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
tick_spacing = 1
ax.xaxis.set_major_locator(ticker.MultipleLocator(tick_spacing))
ax.set_ylabel('Relative Count', size = 'large', backgroundcolor = 'yellow',labelpad
= 20)
ax.set_xlabel('Year', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

Key Takeaways: JavaScript has been the most popular language when it comes to pull request feature in GitHub, but recently Python has falling in GitHub. TypeScript and Ruby (maybe even PHP can be considered) have also been steadily improving while GO and Java have been declining.

Public Repositories Data Visualization and Analysis

```
df9 = pd.DataFrame(repos.head(10))
df9['num_repos//1000'] = df9['num_repos']//1000
df9
```

```
# Absolute Value trend overall
plt.figure(figsize = (10,10))
ax = sns.barplot(x = 'language', y = 'num_repos//1000',data = df9)
ax.bar_label(ax.containers[0])
ax.set_ylabel('No. of Repos // 1000', size = 'large', backgroundcolor =
'yellow',labelpad = 20)
ax.set_xlabel('Languages', size = 'large',backgroundcolor = 'yellow',labelpad = 20)
```

Finally, based on the provided data, it is important to draw the following important conclusion about the two most popular languages on GitHub: Python has recently surpassed JavaScript in terms of popularity among issues and pull requests on the platform. Previously, JavaScript was highly popular.

REFERENCES

[1] R. M. Siegfried, K. Herbert, K. Leune and J. P. Siegfried, "Trends Of Commonly Used Programming Languages in CS1 And CS2 Learning," in *2021 16th International Conference on Computer Science & Education (ICCSE)*, August 2021.

[2] E. G. Jr, "CS1 programming language options," *ResearchGate*, 2007.

[3] H. Naveed, . A. U. Khan, S. Qiu, M. Saqib and S. Anwa, "A Comprehensive Overview of Large Language Models," 27 December 2023.

[4] "Stackoverflow," 2023. [Online]. Available: https://survey.stackoverflow.co/2023/.

[5] M. M. A. K. N. I. &. T. B. Samuli Laato, "Trends and Trajectories in the Software Industry: implications for the future of work," 20 04 2022.

[6] R. D. A. M. L. W. K. W. Yaofei Chen, "An Empirical Study of Programming Language Trends," *Research Gate*, p. 8, 2010.

[7] S. K. VERMA, "Choosing the Right Programming Language," DevOps.com, 03 01 2024. [Online]. Available: https://devops.com/choosing-the-right-programming-language/.

[8] K. Shah, "Top Programming Languages to Consider for App Development in 2023," *IEEE Computer Society*, 2023.

[9] The Upwork Team, "Web Development Languages 101," 2022.

[10] S. Farshidi, S. Jansen and M. Deldar, "A decision model for programming language ecosystem selection: Seven industry case studies," vol. 139, p. 106640, 2021.

[11] L. S. Vailshery, "Most used programming languages among developers worldwide as of 2023," 2024.

[12] K. GUNNARSSON and O. HERBER, "The Most Popular Programming," *School of Electrical Engineering and Computer Science (EECS)*, vol. 6, p. 50, 2020.

[13] M. El-Qassas, "Top 50 Programming languages in 2023," 2023.

[14] "Stack Overflow Developer Survey 2023," 08 January 2024. [Online]. Available: https://survey.stackoverflow.co/2023/#section-most-popular-technologies-programming-scripting-and-markup-languages.

[15] "The Top Programming Languages 2023," 29 Aug 2023. [Online]. Available: https://spectrum.ieee.org/the-top-programming-languages-2023.

[16] "PYPL Popularity of Programming Language index," 2023. [Online]. Available: https://pypl.github.io/PYPL.html.

[17] "TIOBE Index for January 2024," 2024. [Online]. Available: https://www.tiobe.com/tiobe-index/.

[18] GitHub, "The top programming languages | The State of the Octoverse," 2022. [Online]. Available: https://octoverse.github.com/2022/top-programming-languages.

[19] V. NAGPAL, "Kaggle," [Online]. Available: https://www.kaggle.com/code/varunnagpalspyz/data-visualization-on-github-languages-data.

[20] "Google Big Query," [Online]. Available: https://cloud.google.com/bigquery?hl=en. [Accessed 02 12 2023].

[21] V. NAGPAL, "Kaggle," [Online]. Available: https://www.kaggle.com/code/varunnagpalspyz/data-visualization-on-github-languages-data/notebook. [Accessed 18 10 2023].

[22] N. Duggal, "20 Must-Have Python Libraries for Data Science in 2024," 2023.

[23] "GitHub Docs," 2024. [Online]. Available: https://docs.github.com/en/rest?apiVersion=2022-11-28.