Popularity of Programming languages over the years

Data analysis on most popular programming languages from 2004 to 2022

Aryan Singh
Department of Computer Science and Engineering
Meghnad Saha Institute of Technology
Calcutta, India
aryansingh_cse2021@msit.edu.in

Neeraj P Pillai
Department of Computer Science and Engineering
Meghnad Saha Institute of Technology
Calcutta, India
neeraj_p.cse2021@msit.edu.in

Abdul Basit
Department of Computer Science and Engineering
Meghnad Saha Institute of Technology
Calcutta, India
abdul_b.cse2021@msit.edu.in

Kuntal Barik
Department of Computer Science and Engineering
Meghnad Saha Institute of Technology
Calcutta, India
kuntal_b.cse2021@msit.edu.in

Harshil D Sheth
Department of Computer Science and Engineering
Meghnad Saha Institute of Technology
Calcutta, India
hdsheth.cse2021@msit.edu.in

***Abstract*—We begin with the age-old and divisive question ”Which is the best programming language?”. But with a willingness to keep the same number of friends, when we started out with this project!**

I. Introduction

This project is based on data analysis of popular programming languages over the decades We have used the functionality provided by the Python programming language and libraries built for the same. The libraries used are Pandas, Numpy, and Seaborn. The data collected has been categorically sorted into months leading to January 2022. In this project, we have discussed how languages have held up over the years.

A. Primary Categorisation

We have settled into four primary categorisations :

- **[PP]** The languages which have remained popular over the years.
- **[PF]** The languages which are no longer popular.
- **[FP]** The languages which have become popular.
- **[FF]** Languages that never gained market dominance.

B. Secondary Categorisation

The secondary category are designed to include languages that show trends that do not necessarily fit into what we have established in the primary categories, secondary categories :

- ”Mid for Life”, this category has been named after a popular gen-z slang ”mid” meaning below average.
- ”Gaining Popularity”, this category includes languages that are early in development and are gaining market rapidly.

II. Definitions

Popular: for a language to be considered popular a minimum of 8

Mid: is a language that never crossed the upper boundary of 7

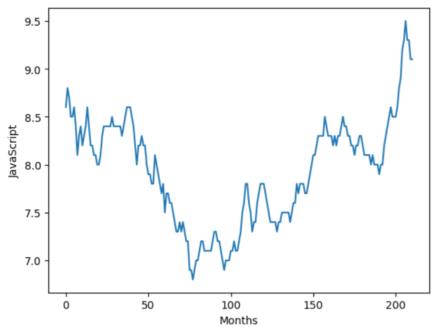
Bust: is a language that never gained more than 3

III. Primary Classification

A. PP

The languages which have remained popular over the year. The only language in our data-set that fits with this description of ours is JavaScript.

- Javacript

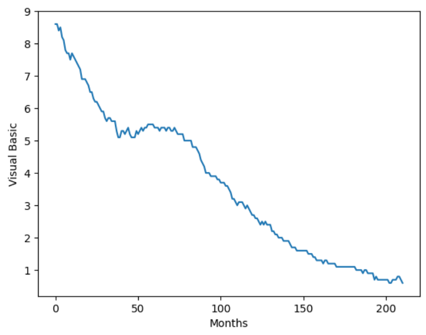


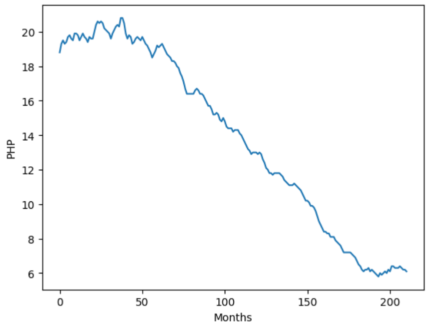
B. PF

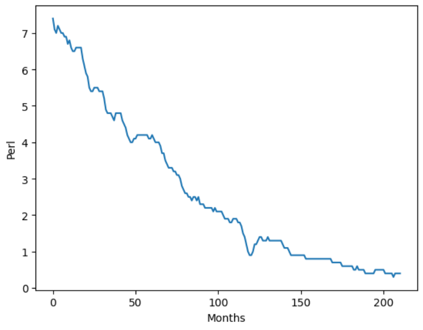
The languages which are no longer popular.

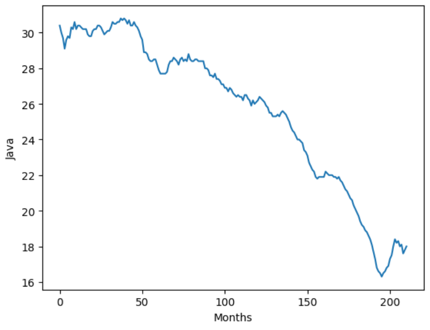
- Visual Basic
- PHP

- Perl:Although PHP never reached a market dominant position, reaching a maximum position of market share we can however correlate its downfall to Python with a correlation coefficient of -0.943219
- Java





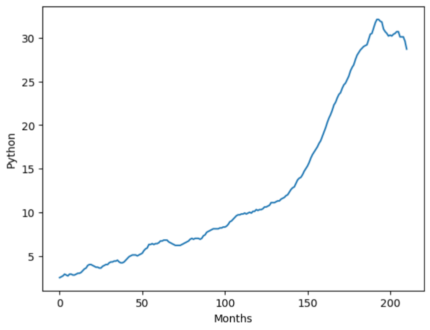




C. FP

The languages which have become popular. The only language fitting this description of ours is Python.

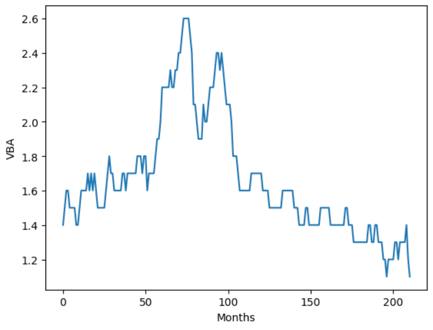
- Python

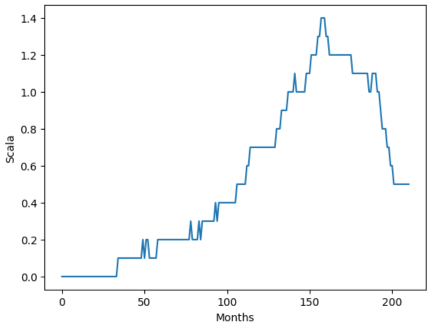


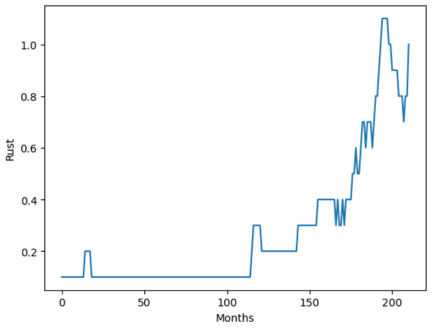
D. FF

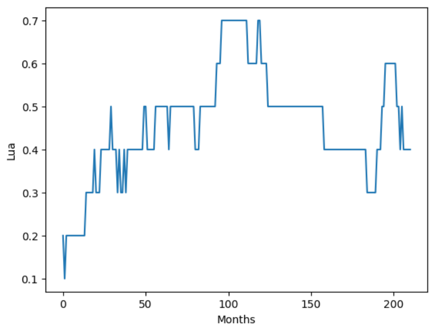
The languages that never gained market dominance

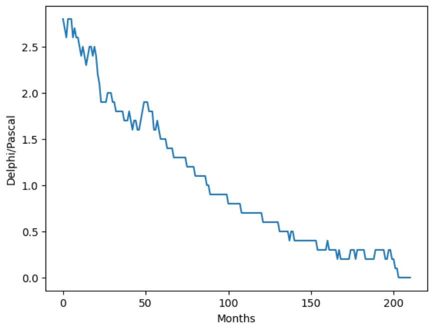
- VBA
- Scala
- Rust
- Lua
- Pascal
- Cobol
- Ada
- Abap

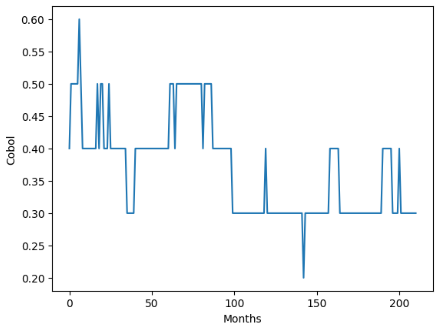


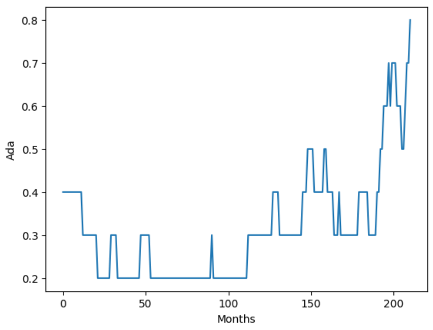


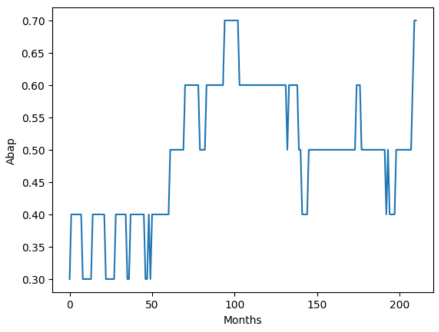










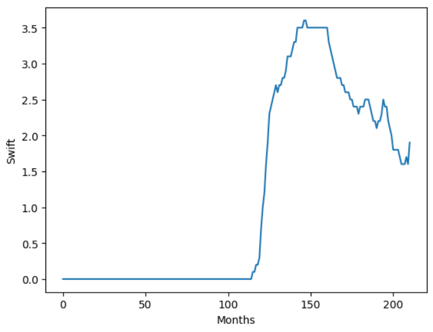


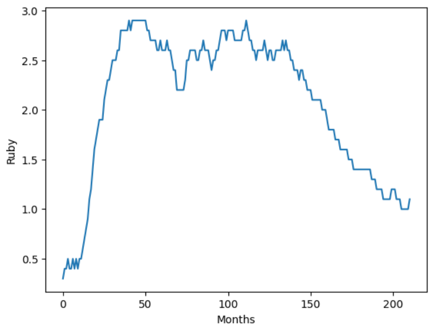
IV. Secondary Classification

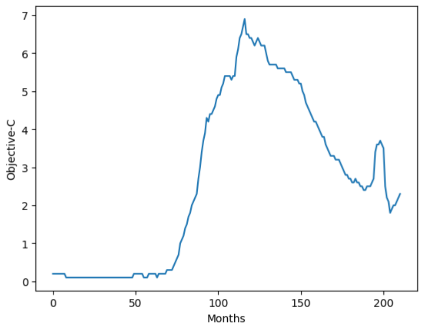
A. Mid for life

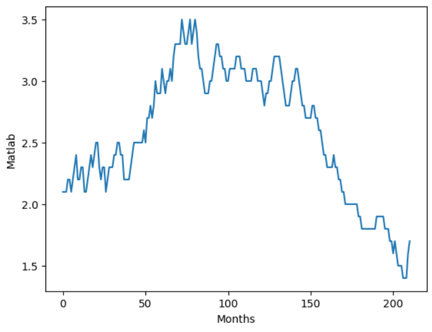
This category has been named after a popular gen-z slang ”mid” meaning below average.

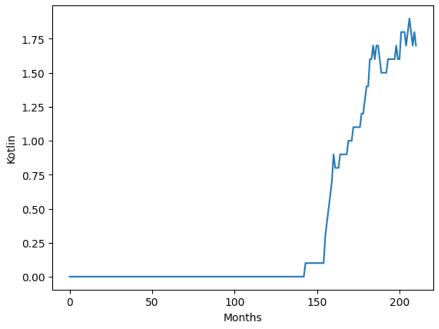
- Swift
- Ruby
- Objective-C
- Matlab
- Kotlin
- Julia
- Haskell
- C sharp
- C/C++

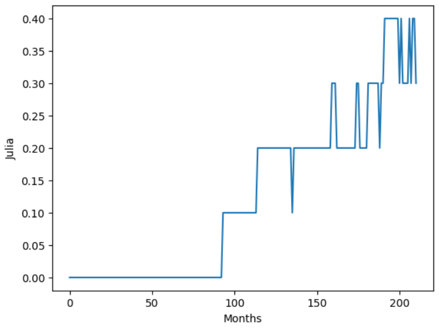


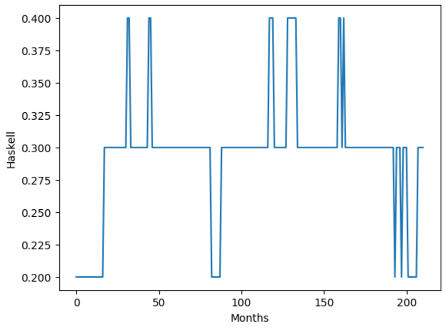


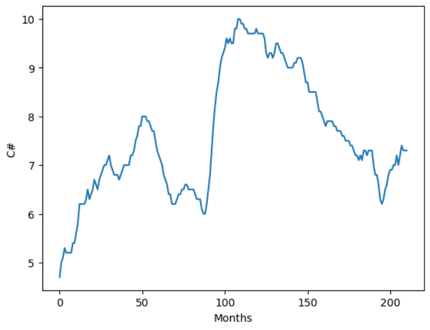


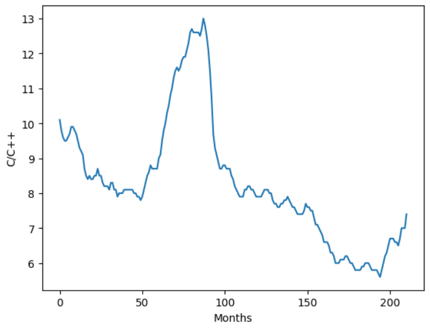








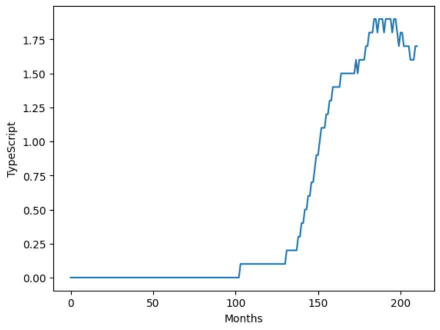


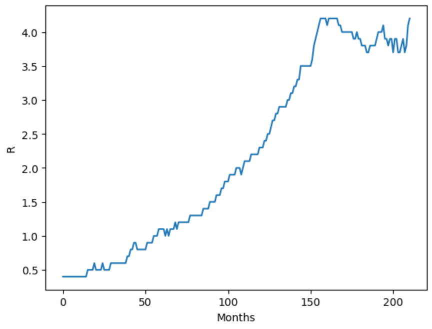


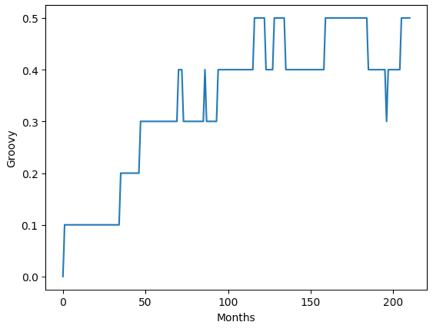
B. Gaining Popularity

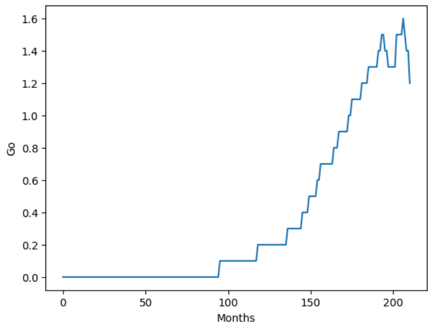
This category includes languages that are early in development and are gaining market rapidly.

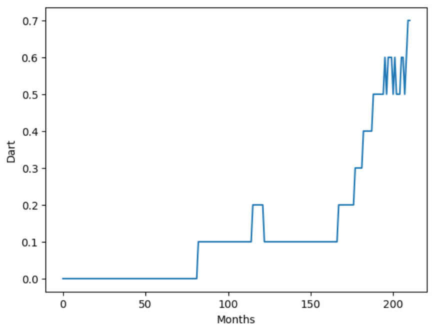
- Typescript
- R
- Groovy
- Go
- Dart











C. Correlation

In statistics, correlation or dependence is any statistical relationship, whether causal or not, between two random variables or bivariate data. Although in the broadest sense, ”correlation” may indicate any type of association, in statistics it usually refers to the degree to which a pair of variables are linearly related. Familiar examples of dependent phenomena include the correlation between the height of parents and their offspring, and the correlation between the price of a good and the quantity the consumers are willing to purchase, as it is depicted in the so-called demand curve.

Correlations are useful because they can indicate a predictive relationship that can be exploited in practice. For example, an electrical utility may produce less power on a mild day based on the correlation between electricity demand and weather. In this example, there is a causal relationship, because extreme weather causes people to use more electricity for heating or cooling. However, in general, the presence of a correlation is not sufficient to infer the presence of a causal relationship (i.e., correlation does not imply causation).

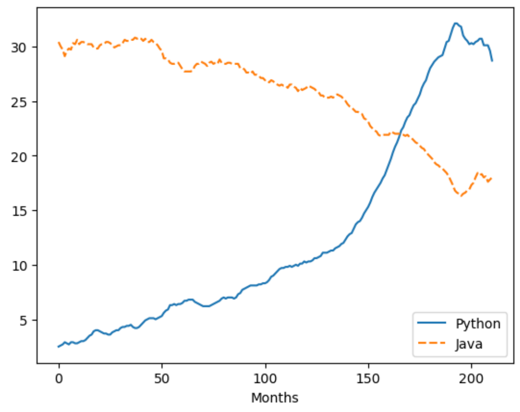
In our analysis we have found that Java was the top language till April 2018, starting from July 2004. Then the top spot has been reserved by Python, till the end of our sample size. Seeing this we decided to run a comprehensive analysis of the correlation between python and Java. We have also included their correlation to other languages.

D. Correlation between other languages and Python and Java

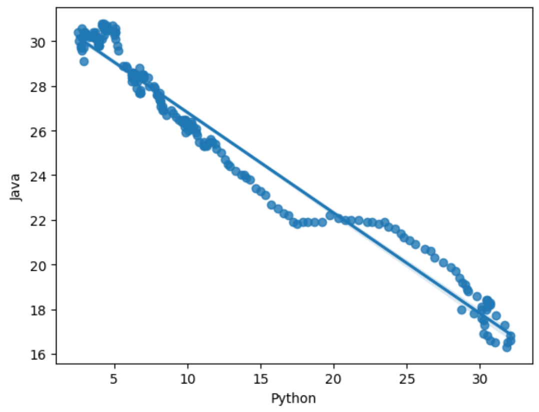
Languages	Python	Java
Abap	0.210543	-0.275354
Ada	0.677052	-0.711266
C/C++	-0.702704	0.681450
C-Sharp	0.085039	-0.169606
Cobol	-0.514747	0.534975
Dart	0.901937	-0.891201
Delphi/Pascal	-0.803821	0.842192
Go	0.985374	-0.957954
Groovy	0.676539	-0.719915
Haskell	0.028710	-0.049515
Java	-0.983492	1.000000
Javascript	0.385331	-0.322988
Julia	0.905960	-0.939636
Kotlin	0.945595	-0.892701
Lua	0.069608	-0.154700
Matlab	-0.617304	0.529346
Objective-C	0.325187	-0.431106
Perl	-0.783993	0.826646
PHP	-0.943219	0.966258
Python	1.000000	-0.983492
R	0.900140	-0.933396
Ruby	-0.441982	0.411585
Rust	0.925995	-0.916171
Scala	0.733003	-0.777016
Swift	0.695197	-0.745004
Typescript	0.970837	-0.944515
VBA	-0.602485	0.588156
Visual Basic	-0.870000	0.898426

E. Correlation between the rise and fall of Python and Java

```
import numpy as np
import pandas as pd
import seaborn as sb

data = pd.read_csv('data/original_data.csv');
data.to_csv('data/ad3.csv',
    float_format='%.3f')

sb.regplot(x='Python', y='Java', data = data)
data2 = data.drop(columns = ['Months','Date'])

corelation = data2.corr()
corelation[['Python', 'Java']]

data3 = data[['Months', 'Python', 'Java']]
data3 = data3.set_index('Months')

sb.lineplot(data = data3)
```



A visual representation of the co-relation between Java and Python's fall and rise



A visual representation of the co-relation between Java and Python's fall and rise

V. Data

A. Data and derivatives

- Original Data CSV file
- Barplot graphs of all langauages
- Lineplot graphs of all langauages

B. Jupyter Notebook

- Main
- Correlation
- Visualisation

VI. Acknowledgment

The whole team would like to attribute the successful completion of this project to our friends and the internet at large.

References

[1] Correlation (2023) Wikipedia. Available at: https://en.wikipedia.org/wiki/Correlation (Accessed: 20 July 2023).

[2] @articleWaskom2021, doi = 10.21105/joss.03021, url = https://doi.org/10.21105/joss.03021, year = 2021, publisher = The Open Journal, volume = 6, number = 60, pages = 3021, author = Michael L. Waskom, title = seaborn: statistical data visualization, journal = Journal of Open Source Software

[3] @ArticleHunter:2007, Author = Hunter, J. D., Title = Matplotlib: A 2D graphics environment, Journal = Computing in Science & Engineering, Volume = 9, Number = 3, Pages = 90–95, abstract = Matplotlib is a 2D graphics package used for Python for application development, interactive scripting, and publication-quality image generation across user interfaces and operating systems., publisher = IEEE COMPUTER SOC, doi = 10.1109/MCSE.2007.55, year = 2007

[4] @softwarereback2020pandas, author = The pandas development team, title = pandas-dev/pandas: Pandas, month = feb, year = 2020, publisher = Zenodo, version = latest, doi = 10.5281/zenodo.3509134, url = https://doi.org/10.5281/zenodo.3509134

[5] @InProceedings mckinney-proc-scipy-2010, author = Wes McKinney , title = Data Structures for Statistical Computing in Python , booktitle = Proceedings of the 9th Python in Science Conference , pages = 56 - 61 , year = 2010 , editor = Stéfan van der Walt and Jarrod Millman , doi = 10.25080/Majora-92bf1922-00a

[6] @conferenceKluyver2016jupyter, Title = Jupyter Notebooks – a publishing format for reproducible computational workflows, Author = Thomas Kluyver and Benjamin Ragan-Kelley and Fernando Pérez and Brian Granger and Matthias Bussonnier and Jonathan Frederic and Kyle Kelley and Jessica Hamrick and Jason Grout and Sylvain Corlay and Paul Ivanov and Damián Avila and Safia Abdalla and Carol Willing, Booktitle = Positioning and Power in Academic Publishing: Players, Agents and Agendas, Editor = F. Loizides and B. Schmidt, Organization = IOS Press, Pages = 87 - 90, Year = 2016

[7] @bookstatistical, title=Statistical Methods (Combined), isbn=9781259081071, url=https://books.google.co.in/books?id=PU0oNBfhn5gC, publisher=Tata McGraw-Hill Education