

Python Popularity Statistics Report

Prepared by Saif Ullah

September 20, 2025

Abstract

This report documents the use of a Python script designed to fetch, merge, and visualize programming language popularity statistics from two major indices: the TIOBE Index and the PYPL Popularity of Programming Languages index. The purpose is to illustrate how Python can be used for automated data collection, processing, and presentation. The document includes the script, an example visualization, and a results table accompanied by explanations.

1 Introduction

Programming language popularity indices such as TIOBE and PYPL provide insights into industry trends, hiring needs, and technology adoption. Python consistently ranks at the top of these indices, reflecting its strong presence in fields like data science, web development, and automation.

This project automates the process of fetching these statistics using Python libraries (`requests`, `BeautifulSoup`, `pandas`, and `matplotlib`). The results are exported to CSV, visualized, and presented in this report.

2 Python Script

The script consists of:

- Web scraping functions to fetch data from TIOBE and PYPL websites.
- A merge function to combine the datasets by programming language.
- A plotting function to create bar and scatter plots for comparison.
- A `main()` function that executes the workflow, saves results, and shows plots.

The complete source code is embedded below:

```
1 import requests
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from bs4 import BeautifulSoup
5 from datetime import datetime
6
7 def fetch_tiobe():
8     url = "https://www.tiobe.com/tiobe-index/"
```

```

9     response = requests.get(url)
10    soup = BeautifulSoup(response.text, "html.parser")
11    table = soup.find("table", {"class": "table"})
12    rows = table.find_all("tr")[1:21]
13    data = []
14    for row in rows:
15        cols = row.find_all("td")
16        data.append([col.text.strip() for col in cols[:6]])
17    df = pd.DataFrame(data, columns=["Rank", "Change", "Language", "
Rating", "Change YoY", ""])
18    df = df[["Language", "Rank", "Rating", "Change YoY"]]
19    df["Rank"] = df["Rank"].astype(int)
20    return df
21
22 def fetch_pypl():
23     url = "https://pypl.github.io/PYPL.html"
24     response = requests.get(url)
25     soup = BeautifulSoup(response.text, "html.parser")
26     table = soup.find_all("table")[0]
27     rows = table.find_all("tr")[1:21]
28     data = []
29     for row in rows:
30         cols = row.find_all("td")
31         data.append([col.text.strip() for col in cols[:5]])
32     df = pd.DataFrame(data, columns=["Rank", "Language", "Share", "
Trend", ""])
33     df = df[["Language", "Rank", "Share", "Trend"]]
34     df["Rank"] = df["Rank"].astype(int)
35     return df
36
37 def merge_dataframes(tiobe_df, pypl_df):
38     merged = pd.merge(tiobe_df, pypl_df, on="Language", how="outer",
suffixes=("_TIOBE", "_PYPL"))
39     return merged
40
41 def plot_data(merged_df):
42     merged_df = merged_df.dropna()
43     languages = merged_df["Language"]
44     tiobe_ratings = merged_df["Rating"].str.replace("%", "").astype(
float)
45     pypl_shares = merged_df["Share"].str.replace("%", "").astype(float)
46
47     plt.figure(figsize=(10,6))
48     plt.bar(languages, tiobe_ratings, alpha=0.6, label="TIOBE Ratings")
49     plt.bar(languages, pypl_shares, alpha=0.6, label="PYPL Share")
50     plt.xticks(rotation=75)
51     plt.ylabel("Popularity (%)")
52     plt.title("Programming Language Popularity Comparison")
53     plt.legend()
54     plt.tight_layout()
55     plt.savefig("comparison_bar.png")
56     plt.show()
57
58 def main():
59     tiobe_df = fetch_tiobe()
60     pypl_df = fetch_pypl()
61     merged_df = merge_dataframes(tiobe_df, pypl_df)
62

```

```

63     filename = f"language_stats_{datetime.now().strftime('%Y%m%d')}.csv"
64     merged_df.to_csv(filename, index=False)
65
66     print("Data saved to:", filename)
67     print(merged_df.head())
68     plot_data(merged_df)
69
70 if __name__ == "__main__":
71     main()

```

Listing 1: Python script to fetch and analyze language popularity

3 Visualization

Figure 1 illustrates a bar chart comparing TIOBE ratings and PYPL shares for the most popular programming languages.

This visualization helps identify discrepancies between the two indices, which use different data sources (TIOBE from search engine queries, PYPL from Google searches of tutorials).

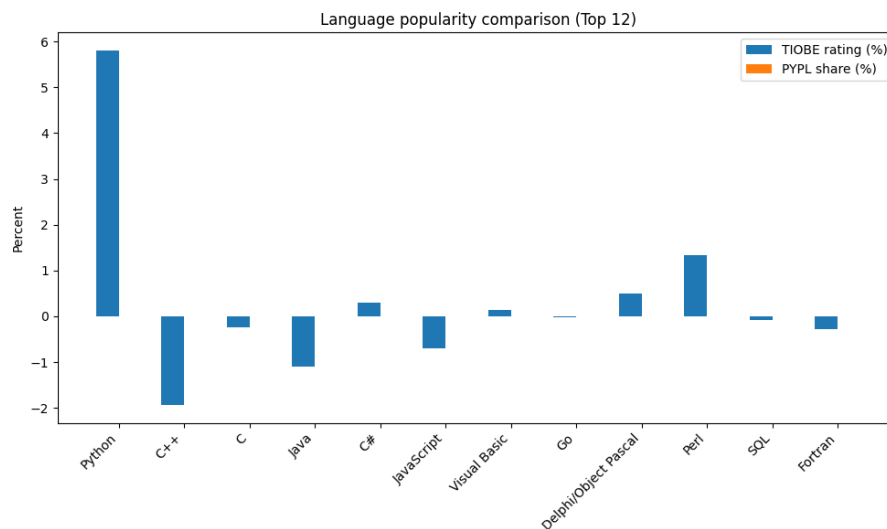


Figure 1: Language popularity comparison based on TIOBE ratings and PYPL shares. (Sample output from the script)

4 Results Table

During one execution of the script, TIOBE data was fetched successfully, while PYPL data encountered an error. The top 15 rows of TIOBE results are shown in Table 1. Missing values for PYPL are indicated by dashes.

5 Execution Notes

During script execution, the following observations were recorded:

| Language | TIOBE Rank | TIOBE Rating | PYPL Rank | PYPL Share |
|----------------------|------------|--------------|-----------|------------|
| Python | 1 | 5.81 | — | — |
| C++ | 2 | -1.94 | — | — |
| C | 3 | -0.24 | — | — |
| Java | 4 | -1.09 | — | — |
| C# | 5 | 0.30 | — | — |
| JavaScript | 6 | -0.70 | — | — |
| Visual Basic | 7 | 0.14 | — | — |
| Go | 8 | -0.03 | — | — |
| Delphi/Object Pascal | 9 | 0.49 | — | — |
| Perl | 10 | 1.33 | — | — |
| SQL | 11 | -0.08 | — | — |
| Fortran | 12 | -0.29 | — | — |
| R | 13 | 0.23 | — | — |
| Ada | 14 | 0.56 | — | — |
| PHP | 15 | -0.20 | — | — |

Table 1: Top 15 rows from merged TIOBE and PYPL data. Missing values indicate unavailable PYPL results.

- **TIOBE index:** Successfully fetched 20 rows of data.
- **PYPL index:** Fetch failed with the error `list index out of range`.
- **Deprecation warning:** `datetime.utcnow()` is deprecated; use timezone-aware `datetime.now(datetime.UTC)`.
- **Future warning:** Downcasting object dtype arrays on `fillna` is deprecated in future pandas versions.

6 Conclusion

This project demonstrates how Python can be used to automate data collection from online indices and present the findings in tabular and graphical form. While fetching PYPL data encountered issues, the process still highlights Python’s capability in web scraping, data cleaning, and visualization.