Data Science Languages

Maria Yli-Heikkilä
maria.yli-heikkila@helsinki.fi
University of Helsinki

ABSTRACT

This paper seeks those focal issues that seem to guide practitioners and scientists in the domain of data science for choosing their main tool, the programming language. None of the languages is a clear cut winner. In practice, an experienced data analyst may become proficient in several languages.

KEYWORDS

data science, programming languages

1 INTRODUCTION

There are myriads of programming languages developed up until now, and new ones are being developed every year. Which are the programming languages mostly used in the domain of data science?

O'Reilly Media is a publisher company of computer technology topics. In their Data Science Salary Survey 2017 around 800 respondents were asked questions about salary, industry, team, programming tools and data technologies. Concerning the popularity of programming languages, SQL showed the biggest share usage among the respondents (>60%), the second Python with >60% share, third R with >50% share, then Bash >30% share, JavaScript, Java, Scala and Visual Basic having all >10% share. [3]

Similarly to O'Reilly's survey, another recent survey conducted by Kaggle in 2018 also showed Python, R and SQL on the top of the lists. Kaggle is an online community of data scientists and machine learners having over a million registered users. Kaggle was founded by Goldboom and Hamner in 2010, and it is probably most famous for its machine learning competitions. In 2017 Kaggle was acquired by Google LLC. Kaggle's Machine Learning and Data Science Survey 2018 aimed to explore qualities of experts working with data and trends in the field of machine learning in different industries. The data from 23,859 usable responses was published and a survey data challenge with prizes was set out. One of the competing data stories by Rochette [4] is referenced here. See Table 1 for the top of the programming languages when the respondents were asked a multiple choice question *what programming languages do you use*. Table 2 shows the top of the list when asked a single choice question *what specific programming language do you use most often*.

DSNS '19, Spring 2019, Helsinki, Finland

Table 1: What programming languages do you use? Number of respondents per the top ten programming language when multiple choices. Source: Rochette [4].

Rank	Language	# respondents
1	Python	15711
2	SQL	8267
3	R	6685
4	C/C++	4383
5	Java	3999
6	JavaScript/Typescript	3249
7	Bash	2708
8	MATLAB	2652
9	C#/.NET	1670
10	Visual Basic/VBA	1274

Table 2: What specific programming language do you use most often? Number of respondents per the top nine programming language when a single choice. Source: Rochette [4].

Rank	Language	# respondents
1	Python	8180
2	R	2046
3	SQL	1211
4	Java	903
5	C/C++	739
6	C#/.NET	432
7	JavaScript/Typescript	408
8	MATLAB	355
9	SAS/STATA	228

2 POPULARITY

Let's focus on the group of data scientists and machine learning practitioners. What makes one choose a specific language? First of all, the nature of the programming language is greatly influenced by its status as a standardized language or de facto language. There is a standardization subcommittee (ISO/IEC JTC 1/SC 22) at the Joint Technical Committee of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) that develops and facilitates standards within the fields of programming languages, their environments and system software interfaces. When it comes to the high standard of robustness needed for critical safety applications, stardardization matters. From the list of programming languages in Tables 1 and 2, SQL,

JavaScript, C, C++ and C# are standardized. Others are so called de facto standards with community driven specifications (R, Python, Java) or proprietary languages (MATLAB, SAS, STATA).

Both Python and R are general-purpose programming languages that have rich choice of application programming interfaces (API) through which other programming enviroments can be accessed interchangeably. Many of the Python or R libraries have parts written for example in C, C++ or Fortran for high performance computation, or in JavaScript for dynamic visualizations, which explains their superior popularity. In contrast to general-purpose, the others are less broadly oriented or even domain-specific languages like Bash at text-processing, organizing files and processes, and general glueing, or SQL at accessing and managing relational databases. This division is higly disputable though.

3 LIBRARY SUPPORT

There is wide library support across computing frameworks and scripting languages, but still there can be found bright clusters of programming language communities in the skies of data science. The clusters are by no means self-sufficient and isolated, but more like polyglots with a leading higher level language.

For example, in the natural language processing (NLP), Apache OpenNLP seems to be quite popular. It is a Java library, and it can be run on Java Virtual Machine. For text mining algorithms Scala is quite efficient and works with Java. Also Apache Spark is Scala native, which makes Scala a prominent pick for NLP. It comes with no supprise that Python community of NLP is also sparkling. There are several libraries for NLP, such as nltk or spyCY.

Tensorflow have become an important library for distributed numerical computation using data flow graphs. It enables to train and run very large neural networks efficiently by distributing the computations across potentially thousands of multi-GPU servers. TensorFlow was created at Google, and was open-sourced in 2015. TensoFlow has APIs to several programming languages, but the APIs in languages other than Python are not yet covered by the API stability promises. Thus, Python has been the first pick entry point to TensorFlow in the early days.

R (earlier S) has been traditionally dominating the academic statistics. It became also the main tool in bioinformatics in the 2000's. In bioinformatics, the gene expression microarrays provided an introduction to large-scale data analysis for many statisticians. In multiple hypothesis testing there were Benjamini and Hochberg's work on false-discovery rate [1] and work from Hastie, Efron and Tibshirani, like LARS algorithm [2], which were publised as R libraries. These gave a reason for bioinformaticians to start using R.

In machine learning algorithms, R and Python have the most extensive libraries. R might even lead in this. Python have had its Scikit-Learn that implements many machine learning algorithms, and enables easy benchmarking, but R's similar Caret library has also become quite popular benchmarking framework. In visualization both Python and R have implementations of JavaScript libraries leaflet and D3. For HTML presentation, Python has its Bokeh and R has Shiny. Vectorization works in both. Parallel processing is not so straighforward, but it works. Geoinformaticians seem to use both Python and R, but Python's visualization libraries of raster data seem to work better. Both have APIs to GDAL and PDAL.

4 COMMUNITY

How does a practitioner become a user of a certain programming language? Tools can be weighed on scales like performance and productivity, but in the end, the community a practitioner belongs to, is quite a decisive factor. The team has settled to some tools, and you need to collaborate. The scientific community discusses through algorithms written in some language, and you need to communicate. Slowly, you assimilate. In fact, if you wish to introduce something new and clearly more productive, organizational inertia may be hard to overcome. Few have succeeded.

5 CONLUSION

In the end of the day, an experienced data analyst may have become proficient in several programming languages. And corollary, technonogies are developing at such a pace, that it is hard to stick to one tool. Knowledge tends to get outdated quite fast in this field. A superhero of today is a polyglot programmer, who can master several programming languages, and thus, leverage their power on different frameworks.

REFERENCES

[1] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B.* 57, 1 (1995), 289–300.

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. 2004. Least angle regression. *Ann. Statist.* 32 (2004), 407–499.

[3] Brian Suda. 2017. *2017 Data Science Salary Survey: Tools, Trends, What Pays (and What Doesn't) for Data Professionals.* O'Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, USA. Retrieved March 31, 2019 from https://www.oreilly.com/data/free/files/2017-data-science-salary-survey.pdf

[4] Rochette Sébastien. 2019. Kaggle survey 2018: Maps of programming languages used and repartition of reproducibility issues. https://statnmap.com/2019-03-20-kaggle-survey-2018-maps-of-languages-used-and-\repartition-of-reproducibility-issues/