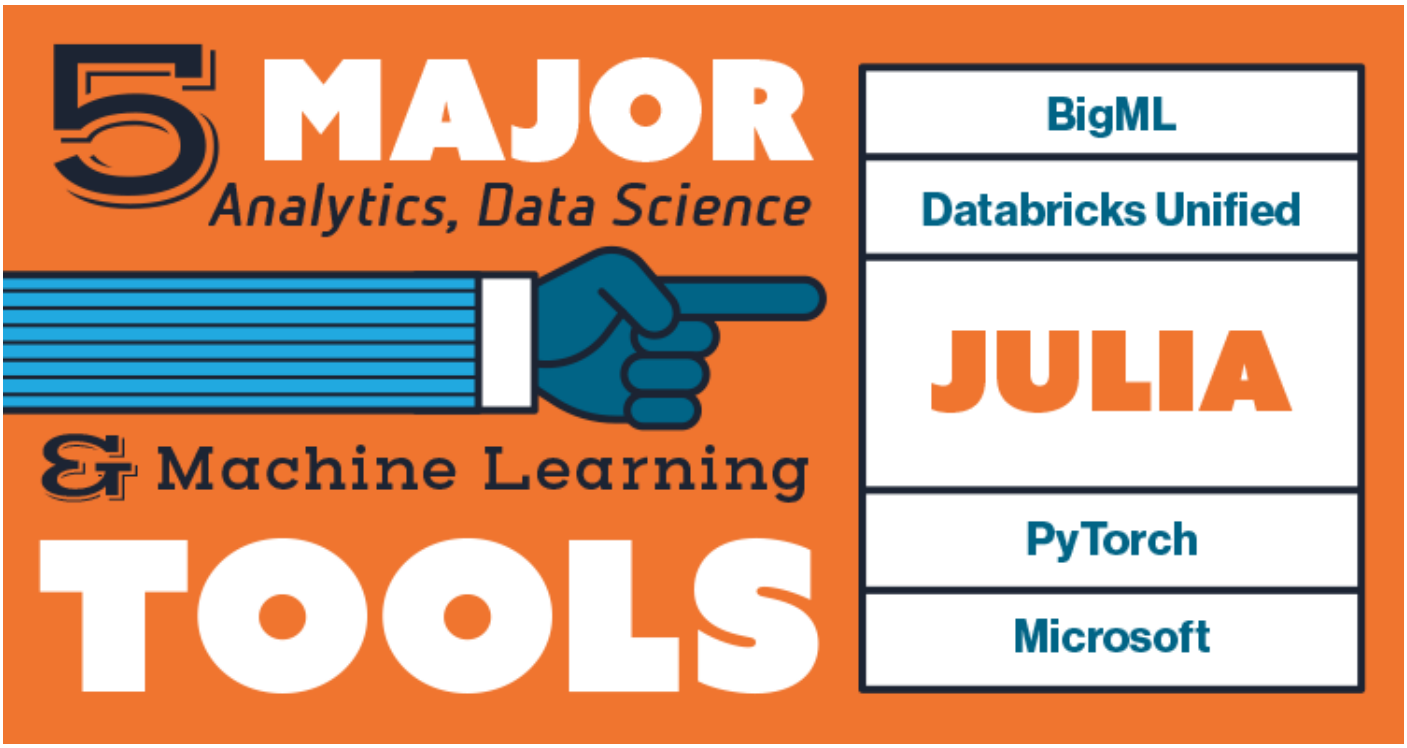
What is all this buzz about the Julia Programming Language?

Janet Wagner

Zylotech™

Customer Data & Analytics Blog

Janet Wagner, on August 01, 2019 | 3 minute read



There seems to be a lot of buzz around the Julia programming language lately. After nearly ten years of development, Julia 1.0 was released in August 2018. And according to the KDnuggets software poll published in May 2019, Julia is one of five major analytics, data science, and machine learning tools with a significant increase in usage compared to the 2018

software poll. This post takes a look at the Julia programming language and highlights a few of the advantages and disadvantages of Julia compared to Python.

What is Julia?

Julia is an open-source, dynamically-typed, programming language designed to excel at scientific and numerical computing. The Julia programming language can be used for specialized domains such as machine learning, visualization, and data science. Julia can also be used for general-purpose programming. The language supports parallelism out of the box, offering three main levels of parallelism which are categorized as Julia coroutines (green threading), multi-threading (currently experimental), and multi-core or distributed processing.

Comparing Julia to Python

The most apparent disadvantage of Julia compared to Python is that it is a relatively new language, so the developer community is small. A smaller developer community means that there are far fewer debugging tools for Julia than Python. And the number of libraries and packages available for Julia is significantly lower than Python. However, when it comes to packages, Julia has an advantage over Python in that packages can be written entirely in Julia. Many of the data science packages available in Python are based, in part, on other programming languages such as C/C++ and Java. Data scientists must often solve problems where there are no corresponding Python packages available. Writing a package in one language is easier than writing a package that requires several languages. And in some cases, creating a data science package in Python would require C/C++, which is a complicated language that some data scientists

may not know well or at all.

Another advantage of Julia over Python is that it solves the “two-language problem” in data science, where data scientists are writing programs twice: one time for development and a second time for speed and performance. Data scientists typically develop algorithms and analytics prototypes in Python or R first and then rewrite the code for production in a higher-performing, faster language such as C/C++ or Java. Julia is designed so that the same source code can be used for the development prototype and production application.

When it comes to performance, Julia has an advantage over Python because Python is an interpreted language, and Julia is a compiled language. Unlike Julia code, Python code must be interpreted before it is executed on the processor. Julia code is sent directly to the processor as executable code. Also, there are performance optimizations that can be done in Julia that cannot be done in Python. Julia is designed for high performance, so it is possible to write Julia code that provides a level of performance and speed that is close to C (see the recommended Julia performance tips).

Should you learn Julia?

When it comes to data science, machine learning, and analytics, the demand for data scientists (and marketers) to know Python is high. Python holds the #1 spot on the May 2019 KDnuggets software poll, and will likely remain in the top place on the poll for years to come. If you're new to data science and machine learning, it would be prudent to learn Python first. With that said, Julia has some advantages over Python that make the language worth checking out. If you're already fluent in Python, learning Julia shouldn't be difficult. Depending on your upcoming data science projects, it may be worth adding the Julia programming language to your repertoire.

Janet Wagner is a Zylotech contributing writer.

If you liked this post, check out our recent blog post: Common algorithms in marketing: Decision Trees.

Subscribe to our blog newsletter for all things customer data and analytics, AI and marketing, sent monthly.