

EKOIST Journal of Econometrics and Statistics

REVIEW ARTICLE / DERLEME MAKALASI

Veri Bilimi İçin Yeni Bir Programlama Dili: Julia

A New Programming Language for Data Science: Julia

Münever Turanlı* , Ünal Halit Özden** 

Öz

Bilimsel programlama ve veri bilimi ile ilgili olarak kullanıcılar; yüksek performanslı, hızlı, kod yazım ve protipleme kolaylığına sahip, esnek ve dinamik programlama diline ihtiyaç duymaktadır. Veri bilimi dünyasında kullanılan, bir çok programlama dili vardır. Bu dillerden bazıları çok hızlı ama öğrenmesi ve kod yazması zordur. Bazılarında ise kod yazmak çok kolaydır, ancak yazılan kodların çalışma hızları çok yavaştır. Diğer dillerle karşılaştırıldığında, nispeten daha yeni bir dil olan Julia programlama dili, hem yüksek performanslı, hem hızlı, hem de kod yazması kolay olduğu için bu sorunları aşmayı amaçlayan programlama dillerinden biridir. Bu makalenin amacı da, diğer programlama dillerine nazaran daha yeni olan Julia'yı tanıtmak, istatistik ve veri biliminde kullanılan diğer diller ile karşılaştırmak ve araştırmacıların ve özellikle istatistik ve veri bilimi konuları ile ilgilenen kişilerin, Julia dili ile ilgili bilgi edinmesini sağlayıp, kendileri için en uygun dili seçmelerine yardımcı olmaktır.

Anahtar Kelimeler

Julia Programlama Dili, Programlama Dilleri, Python, R, İstatistik, Veri Bilimi

Abstract

Users working in scientific programming and data science need a fast, flexible, and dynamic high-performance programming language with easy code writing and prototyping. Many programming languages exist that are used in the data science world. Some of these languages are very fast but difficult to learn and code, while others are very easy to write code for but have a very slow running speed. In comparison to other programming languages, the relatively new Julia is a high performance programming language that aims to overcome these problems by being both fast and easy to code. Therefore, the purpose of this article is to introduce Julia and to compare it to the other programming languages used in statistics and data science. In addition, this article also aims to help researchers, especially those interested in statistics and data science, learn about the Julia programming language and to choose the language best suited for them.

Keywords

Julia Programming Language, Programlama Languages, Python, R, Statistics, Data Science

* Münevver Turanlı (Prof. Dr.) İstanbul Ticaret Üniversitesi, İnsan ve Toplum Bilimleri Fakültesi, İstatistik Bölümü, İstanbul, Türkiye.
E-Posta: mturanli@ticaret.edu.tr ORCID: 0000-0002-9535-4527

** Sorumlu Yazar: Ünal Halit Özden (Prof. Dr.) İstanbul Ticaret Üniversitesi, İnsan ve Toplum Bilimleri Fakültesi, İstatistik Bölümü, İstanbul, Türkiye. E-Posta: uozen@ticaret.edu.tr ORCID: 0000-0003-0924-4848

Atıf: Turanlı, M., & Ozden, U.H. (2023). Veri bilimi için yeni bir programlama dili: Julia. *EKOIST Journal of Econometrics and Statistics*, 38,

Extended Summary

Many programming languages are used in the world of data science. Some of these languages are very fast but difficult to learn and code, while others are very easy to write code for but have a very slow code speed. This is known in the literature as the *two-language problem*, and Julia is one programming language that aims to overcome this problem.

With regard to the field of data science, Python is mostly used in the business world, while R is more used in the academic world. Despite being a general programming language, Julia is also one of the strongest languages used in the field of data science and is even considered to be the future of scientific computing and scientific data analysis.

Julia (Bezanson et al., 2017) is a relatively new programming language that was first released in 2012 and aims to be both easy and fast compared to other programming languages. Julia has been described as working like C while reading like Python (Perkel, 2019). Julia is a dynamically typed language, which means that the type of variable is not specified when declared but rather determined while in operation. Julia can easily process large amounts of data and calculations, and manipulating, creating, and prototyping code are very easy in Julia. Julia is known for its speed and ability to easily interface with other programming languages. This makes it a popular choice for data science and machine learning applications. Julia is also one of the rare programming languages grouped with C, C++, and Fortran in the in the Petaflop Club, these languages having been able to achieve the highest performances in excess of one petaflop per second.

Julia is an MIT-certified, open-source programming language. While it operates as a general-purpose language for use in any application development, many of its features are customized for numerical analysis and data science. In other words, Julia is a high-level programming language designed for use in numerical and scientific computing. Therefore, Julia offers a high success rate for numerical computational operations. In addition to these, Julia's syntax has a distributed (parallel) processing facility and large library of mathematical functions, similar to programming languages such as Python and Ruby. Julia is also at the forefront of many other computational fields such as data science, machine learning, artificial intelligence, and statistics.

Julia is newer compared to other programming languages. This suggests that learning about Julia and its programming language ecosystem will be difficult due to the small amount of knowledge and number of users involved in the language. On the other hand, difficulties may exist in terms of where to start or how to install and use different packages in relation to the programming language. For this reason, this article aims to introduce Julia as the newest programming language and to compare it with other programming languages used in statistics and data science. This article

additionally aims to help researchers learn about Julia, especially those interested in statistics and data science, and choose the most suitable language for them.

Julia's most important features are its speed, performance, and ease of use. It has been designed to perform like C++, have the general programming ability of Python, and the statistical power of R. One of Julia's outstanding is its ability to use code written in other programming languages such as Python, R, Java, and C with the help of packages such as PyCall, RCall, and JavaCall. In this way, codes written in these languages can be referenced and executed in Julia. Meanwhile, for those who want to learn Julia, many printed and online resources exist that are being rapidly developed for teaching the basic features of this language, although this number is still not as high as for other languages. However, due to being a new programming language, Julia can also be said to have technical disadvantages: Similar to more established high-level languages such as Python, Julia requires a little more development to become more stable.

Ultimately, coding languages are simply tools, and as tools change and develop, they can be widely used for a time before eventually losing popularity and becoming obsolete. Therefore, focusing on the fundamentals of the field of science (i.e., the domain knowledge for statistics and data science) and most importantly how to apply them would be more important than giving too much focus to the tools used. Whether one knows how to code in Python, R, or Julia, if they lack proficiency in data science or statistics and don't know how to identify the problem or how to ask the right questions, then one's knowledge of coding simply won't do much good. In contrast, having a good foundation in statistics and data science will facilitate one's ability to learn any programming language.

Giriş

Her meslek, verilerin artan erişilebilirliğinden ve verilerin artan öneminden etkileniyor ve artarak etkilenmeye devam edecek (Chen ve diğerleri, 2014 ; Khan ve diğerleri, 2014). Genel olarak veriler; iletişim kurmak ve bilgi oluşturmak için, karar verme sürecinde girdi olarak kullanılır. Bu nedenle, yaşamın içinde veriler çok önemlidir. Ancak daha da önemlisi var olan bu verilerin işlenmesi, analiz edilmesi ve bu verilerden bilgi üretilmesidir. Bunun için de önce istatistik ve sonrasında da programlama dili bilmek edinilmesi gereken en önemli temel yetkinliklerden biridir.

Veri bilimi dünyasında kullanılan, bir çok programlama dili vardır. Bu dillerden bazıları çok hızlı ama öğrenmesi ve kod yazması zordur. Bazılarında ise kod yazmak çok kolaydır, ancak yazılan kodların çalışma hızları çok yavaştır. Literatürde bu durum, iki dil sorunu olarak bilinir ve Julia bu sorunu aşmayı amaçlayan programlama dillerinden biridir.

Programlamada, ikili dil tipik olarak tek bir proje veya uygulamada iki programlama dilinin kullanılmasını ifade eder. Bu, her dilin güçlü yönlerinden yararlanmak, bir dilde yazılmış mevcut kodu kullanmak veya yalnızca bir dilde mevcut olan bir kütüphaneyi veya çerçeveyi kullanmak gibi çeşitli nedenlerle yapılabilir. Örneğin, bir programcı, bir uygulamanın ana mantığı için bir dil ve kullanıcı arabirimi için başka bir dil kullanabilir. Ya da verileri işlemek için bir dil ve onu görselleştirmek için başka bir dil kullanabilir. İkili dil kullanan bir programcının her iki dile de aşina olması ve birlikte nasıl etkili bir şekilde çalışabileceklerini anlaması gerekmektedir.

Veri bilimi alanına ilişkin olarak İş dünyasında, çoğunlukla Python ve akademik dünyada ise R daha çok kullanılmaktadır. Genel bir programlama dili olmasına karşın Julia da veri bilimi alanında kullanılan güçlü dillerden biridir. Hatta Julia'nın bilimsel hesaplama ve bilimsel veri analizinin geleceği olduğu düşünülmektedir.

Julia programlama dili (Bezanson vd., 2017), ilk olarak 2012'de piyasaya sürülen nispeten yeni bir dildir ve diğer programlama dillerine göre hem kolay, hem de hızlı olmayı amaçlamaktadır. "C gibi çalışır ama Python gibi okur" (Perkel, 2019). Dinamik olarak yazılmış bir dildir, yani bir değişkenin türü, bildirildiğinde belirtilmez, bunun yerine çalışma zamanında belirlenir. Büyük miktarda veriyi ve hesaplamayı kolaylıkla işleyebilir, kodları manipüle etmek, oluşturmak ve kodların prototipini oluşturmak Julia'da oldukça kolaydır. Julia, hızı ve diğer programlama dilleriyle kolayca arayüz oluşturabilme yeteneği ile tanınır. Bu da onu veri bilimi ve makine öğrenimi uygulamaları için popüler bir seçim haline getirmektedir. Diğer taraftan Julia, 'Petaflop Kulübü' olarak adlandırılan, saniyede bir petaflop'u aşan en yüksek performansa ulaşan nadir programlama (bu diller C, C++, Fortran ve Julia) dillerinden birisidir.

Julia programlama dili, MIT sertifikalı, açık kaynak kodlu bir programlama dilidir. Genel amaçlı bir dil olsa da (herhangi bir uygulama geliştirilmesinde kullanılabilse

de) özelliklerinin çoğu sayısal analiz ve veri bilimi için özelleştirilmiştir. Diğer bir ifade ile Julia, sayısal ve bilimsel hesaplama için kullanılmak üzere tasarlanmış üst düzey bir programlama dilidir. Dolayısıyla sayısal hesaplamalı işlemler için yüksek başarı oranı sunmaktadır. Bunların haricinde dağıtık (paralel) işlem olanağına ve geniş matematiksel işlev kütüphanesine sahip olan Julia'nın söz dizimi Python, Ruby gibi programlama dilleriyle benzerlik göstermektedir. Diğer taraftan Julia programlama dili veri bilimi, makine öğrenimi, yapay zeka, istatistik ve daha bir çok hesaplamalı alanda ön plana çıkmaktadır.

Julia programlama dilinin geliştiricileri, Julia'nın temel amacına yönelik olarak 2012'deki bir blog yazısında aşağıda yer alan açıklamaları yapmışlardır (Bezanson, vd. 2012).

“Açgözlüyüz, daha fazlasını istiyoruz. Liberal bir lisansa sahip açık kaynaklı bir dil istiyoruz. Ruby'nin dinamikliği ile C'nin hızını istiyoruz. Lisp gibi gerçek makrolara sahip, ancak Matlab gibi açık, tanıdık matematiksel notasyona sahip homoikonik bir dil istiyoruz. Genel programlama için Python kadar kullanılabilir, istatistik için R kadar kolay, dizi işleme için Perl kadar doğal, doğrusal cebir için Matlab kadar güçlü, programları birbirine yapıştırmada kabuk kadar iyi bir şey istiyoruz. Öğrenmesi çok basit ama en ciddi bilgisayar korsanlarını bile mutlu eden bir şey. İnteraktif olmasını ve derlenmesini istiyoruz.”

Diğer dillerle karşılaştırıldığında, Julia diğer programlama dillerinden daha yeni bir dildir. Bu durum, dille ilgili bilgi birikiminin ve kullanıcı sayısının az olması nedeniyle (programlama dili ekosistemi) Julia'ya ilişkin bilgi edinmenin zor olacağını düşündürmektedir. Diğer taraftan, programlama diline ilişkin olarak nereden başlanacağı, farklı paketlerin nasıl kurulacağı ve kullanılacağı konusunda da zorluklar yaşanabileceğini akla getirmektedir. Bu nedenle bu makalenin amacı diğer programlama dillerine nazaran daha yeni bir programlama dili olan Julia'yı tanıtmak, istatistik ve veri biliminde kullanılan diğer diller ile karşılaştırmaktır. Buna ek olarak, araştırmacıların ve özellikle istatistik ve veri bilimi konuları ile ilgilenen kişilerin, Julia dili ile ilgili bilgi edinmesini sağlayıp, kendileri için en uygun dili seçmelerine yardımcı olmaktır.

2. Julia Programlama Dili

Julia, teknik bilgi işlem disiplinindeki mevcut yazılım araç setindeki prototipleme, performans ve hızdaki boşlukları kapatmak için tasarlanmıştır. Bu konu ile ilgili çalışmalar 2009 yılında, üst düzey hızlı ve de ücretsiz bir dil oluşturmak için yola çıkan Jeff Bezanson, Stefan Karpinski, Viral B. Shah ve Alan Edelman tarafından başlatılmıştır (Lauwens & Allen, 2019).

2012 yılındaki ilk lansmanından günümüze kadar, Julia birçok değişiklikten geçmiş ve birçok yeni sürümü piyasaya sürülmüştür. Günlük olarak halen üzerinde yapılan

çalışmalarla geliştirilmeye devam edilmektedir. Julia, veri analitiği, algoritmik ticaret, makine öğrenimi ve yapay zeka için en hızlı modern yüksek performanslı bilgi işlem dilidir ve Python, R, Matlab, Perl, SAS ve Stata'nın işlevselliğini ve kullanım kolaylığını, C++ ve Java hızıyla birleştirmektedir. Paketleri, etkileyici ve sade söz dizimi nedeniyle gittikçe popüleritesini artırmaktadır. Ayrıca Julia betikleri içerisinde kolaylıkla Python, C, Fortran ve R paketleri kullanılabilir ve kodları çalıştırılabilir. Julia GitHub'da geliştirilen ilk 10 programlama dilinden biridir ve finans, sigorta, enerji, robotik, genomik, havacılık ve diğer birçok alanda kullanılmaktadır.

Ekim 2022'ye kadar, Julia 40 milyondan fazla indirildi ve bu programlama dili için 8000'den fazla paketin kayıtlı olduğu bir topluluk oluştu. Yine Aralık 2022 itibarıyla Julia, PYPL Endeksi'nde 25. (PYPL, Çevrimiçi: <https://pypl.github.io/PYPL.html>) ve TIOBE Endeksi'nde 27. sırada yer almaktadır (TIOBE, Çevrimiçi: <https://www.tiobe.com/tiobe-index/>). Julia'nın en önemli hedef kitlesi, veri bilimi ve bilimsel hesaplamalar için kullanılan Python, MATLAB, R, Mathematica, SAS, Scala ve Octave gibi programlama dilleri ve ortamlarının kullanıcılarıdır. Julia henüz adı Python, R veya MATLAB kadar popüler olmasa da SAS veya Scala ile benzer kullanım oranlarına ulaşmıştır. Julia'nın destekçileri arasında Amazon, Apple, BlackRock, Capital One, Comcast, Disney, Facebook, Ford, Google, Grindr, IBM, Intel, KPMG, Microsoft, NASA, Oracle, PwC, Raytheon ve Uber yer almaktadır.

2.1. Julia Programlama Dilinin Özellikleri

Python ve R, veri biliminde en yaygın kullanılan programlama dilleri olarak üstünlüğünü sürdürürmesine karşın Julia bu alanda kendine özgü gelişmiş özelliklere sahiptir. Julia'nın popüleritesinin artmasının arkasında birçok faydalı özelliğinin olması yer almaktadır. Bu özellikleri en önemlileri; prototipleme kolaylığı, performans, hız, sözdizimi, çoklu gönderim, dinamik programlama dili olması, proje çözüm yönetimi, paket yönetimi, unicode desteği ve kod paylaşımı kolaylığı ve parametrik polimorfizm olmasıdır. Adı geçen özellikler aşağıda açıklanmıştır (Storopoli vd., 2022, s.9).

Prototipleme: Genellikle araştırmacılar çok kolay ama yavaş bir dil kullanarak ve prototip oluşturarak kod geliştirirler. Kodun doğru çalıştığından ve amacına ulaşacağından emin olduktan sonra, yazdıkları kodu hızlı ama kodlaması zor bir programlama diline dönüştürürler. Buna programlamada “iki dil problemi” denir. “İki Dil Problemi”, bilimsel hesaplamada bir problemi çözmek için karşılaşılan tipik bir durumdur. Problemin çözümü, kodlaması kolay bir dilde (Python veya R gibi) prototiplenir. Prototip çalışırsa, daha hızlı ve performanslı olan ancak prototiplemesi kolay olmayan bir dilde (C++ veya FORTRAN) kodları tekrar yazılır. Böyle bir durumda, bir çözüm geliştirme sürecinde yer alan iki programlama diline ihtiyaç duyulmaktadır. Birincisi prototiplemesi kolay ancak uygulamaya uygun olmayan (çoğunlukla yavaş olması nedeniyle) programlama dili, ikincisi kodlaması zor ve

sonuç olarak prototiplemesi kolay olmayan, ancak hızlı olduğu için uygulamaya uygun olan programlama dilidir. Julia, hem hızlı hem de protiplemesi kolay bir dildir. Prototipini oluşturduğunuz (kullanım kolaylığı) ve çözümü (hız) uyguladığımız dilin aynı olması nedeniyle Julia bu tür durumlarda çok fayda sağlamaktadır.

Performans: Programlamada, CPU ve bellek gibi bilgisayar kaynaklarını C veya Fortran kadar verimli kullanan yüksek performanslı programlara ihtiyaç duyulmaktadır. Bu nedenle, bir programlama dilinin üretim versiyonu genellikle “Fortran” veya “C ” programlama dilinde yazılır. Julia sahip olduğu özellikler ve performans sayesinde, kullanıcının basit bir sözdizimi ile hızlı ve güçlü kod geliştirmesini olanak sağlar. Julia programlama dili, CPU ve bellek gibi bilgisayar kaynaklarını C veya Fortran kadar verimli kullanan yüksek performanslı bir programlama dilidir. Bu anlamda Julia, düşük seviyeli bir programlama diline olan ihtiyacı azaltır. Julia, Level Virtual Machine1 (Düşük Seviyeli Sanal Makine-LLVM) ve Just in Time (Tam Zamanında-JIT) gibi derleyici teknolojisindeki son gelişmeleri, söz dizimi kolaylığını ve hızı tek bir ortamda kullanma imkanı sunmaktadır.

Hız: Teknik alandaki bir diğer önemli konu da hızdır. Kullanıcılar genel olarak programlama dillerinin kod çalıştırma hızlarının yüksek olmasını isterler. Julia, bu konuda C ve Fortran’a yaklaşmaktadır. Julia’nın ortaya çıkış ilkelerinden biri, Python kadar genel, R kadar istatistik dostu, ancak C kadar hızlı olması gerektiğidir. Bu, özellikle bilimsel hesaplama için daha hızlı veri iş akışlarının sağlanmasına olanak tanımaktadır.

Paket ve proje çözüm yönetimi: Julia, kontrol edilen, yönetilebilen ve tekrarlanabilen bağımlılıklar ve sürüm kontrolü ile, çok iyi proje ve paket yönetimi sunmaktadır. Daha açık bir ifade ile Julia’nın çok iyi bir proje yönetimi çözümü ve paket yöneticisi vardır. Tek bir küresel paket kümesi kuran ve yöneten geleneksel paket yöneticilerinin aksine, Julia’nın paket yöneticisi “ortamlar” etrafında tasarlanmıştır: Bağımsız bir proje için yerel olabilen veya projeler arasında paylaşılabilen bağımsız paket kümeleri bulunur. Her proje kendi bağımsız paket sürümleri kümesini kullanmaktadır. Ayrıca Julia, kullanıcıların farklı paketlerdeki kodları ve veri yapılarını kolayca paylaşmasına ve yeniden kullanmasına olanak tanımaktadır.

Unicode desteği: Bir bilgisayarın, insanların anlayabileceği metin ve sayıları depolayabilmesi için, karakterleri sayılara dönüştüren bir kod olması gerekir. Unicode standardı, böyle bir kodu karakter kodlaması kullanarak tanımlar. Julia unicode karakterlerini, değişkenler veya parametreler olarak kullanılmasına izin vermektedir. Bir algoritmadaki matematiksel denklem veya ifade, hemen hemen aynı görüldüğü gibi kodlanabilir. Güçlü bir özellik olan bu özellik “Bire Bir Kod ve Matematik İlişkisi” olarak adlandırılmaktadır.

1 *Low Level Virtual Machine:* Düşük Seviyeli Sanal Makine- elle yazılmış ve okunması kolay kodu, insanların okuması çok zor ama bilgisayarların okuması kolay olan LLVM makine koduna dönüştürme

Kod paylaşımını kolaylığı: Paketlerin Julia ekosistemi programcılara benzersiz bir ortam sağlar. Yalnızca kod paylaşımını sağlamakla kalmaz, aynı zamanda kullanıcı tarafından oluşturulan türlerin paylaşılmasına da izin verir.

Geliştirme ortamları: Julia kodlarının çalıştırılması için bir çok seçenek vardır. Bunlardan en kolayı, Julia REPL’i kullanmaktır. Bu, Julia yürütülebilir dosyasının (julia veya julia.exe) başlatılması ve orada kodun çalıştırılması anlamına gelir. Julia kodunu içeren dosyaların uzantı adı “.jl”dir, bu dosyalara kodlar yazılabilir ve bunlar Julia’ya yüklenebilir. REPL net bir şekilde belgelenmemiş bazı manuel işlemler gerektirmektedir. Bu sorunu aşmak için Pluto.jl kullanılabilir. Pluto.jl bağımlılıkları otomatik olarak yönetir, kodları çalıştırır ve değişikliklere tepki verir. Julia programlama diline yeni başlayan kişiler için “Pluto.jl”, en kolay yoldur. Bu paketin en temel dezavantajı, daha büyük projeler için daha az uygun olmasıdır. Diğer seçenekler, “Visual Studio Code”, “Vim” ve “Emacs”ı kullanmaktır. Büyük projeleri yönetmek için “Visual Studio Code”, ileri düzey kullanıcılar için “Vim”, “Emacs” ve “RPL” en iyi alternatiflerdir.

Sözdizimi: Julia’nın sözdizimi, hızına rağmen dinamik olarak yazılmıştır, bu da onu daha karmaşık programlama dillerinden çok R ve Python gibi kodlaması kolay olan dillere yakın kılmaktadır. Dolayısıyla Julia, diğer yüksek performanslı programlama dillerinden çok daha erişilebilir ve öğrenmesi kolay bir programlama dilidir.

Dinamik tipte bir dil: Julia, tam zamanında derleyiciye sahip dinamik tipte bir dildir. Dinamik tipte diller; statik programlama dillerinin derleme zamanında gösterdikleri davranışları, çalışma zamanında da gösterebilen programlama dilleridir. Dinamik kodlanan bir dil olması nedeniyle Julia’da da tanımlanan değişkenlerin türlerinin başlangıçta belirtilmesi gerekmektedir. Tanımlanan değişkenlerin tür kontrolleri programın çalışma anına bırakılmaktadır. Bu, C++ veya FORTRAN’da olduğu gibi programı çalıştırmadan önce kodları derlemek gerekmediği anlamına gelir. Bunun yerine, Julia kodu alır, gerektiğinde türleri tahmin eder ve çalıştırmadan hemen önce kod bölümlerini derler. Ayrıca, Julia çalışırken türleri tahmin ettiğinden, her türü açıkça belirtmek gerekmemektedir. Julia ile R ve Python gibi diğer dinamik diller arasındaki temel fark şudur: Julia, kullanıcının tür bildirimleri belirtmesine izin verir. Bununla birlikte, değişkenlerin veya işlevlerin türü belirtilmek istenmezse, Julia bunları anlar ve tahmin eder.

Çoklu gönderim: Julia’nın bir başka özelliği de çoklu gönderimdir. Bu özellik, Julia fonksiyonlarının kendisine verilen bağımsız değişken türlerine göre farklı davranma becerisine olanak tanır ve Julia’nın daha esnek bir programlama dili olmasını sağlar. Çoklu gönderim, mevcut işlevlerin genişletilmesine veya yeni türler için özel ve karmaşık davranışlar tanımlanmasına olanak tanıyan güçlü bir özelliktir. Ayrıca, Julia, kullanıcıların çoklu gönderim yoluyla birçok bağımsız

değişken türü kombinasyonunda fonksiyon davranışının tanımlanmasına olanak tanır. Aynı fonksiyon adını kullanırken argüman türleri için yeni fonksiyonlar tanımlanarak farklı bir tür davranışı belirtilebilir.

Parametrik polimorfizm (Parametric polymorphism): Parametrik polimorfizm, jenerik programlama dillerinin ve yapılarının arkasındaki temel prensiptir. Bu prensip veri türünden bağımsız olarak değerler üzerinde çalışan genel işlevlerin ve veri türlerinin oluşturulmasını sağlar. Örneğin, bir programlama fonksiyonu iki farklı değer üzerinde çalışıyorsa, aynı veri türlerine sahip olmasalar bile yeni değerler eklenebilir.

2.2. Julia'nın Avantajları ve Dezavantajları

Yukarıda belirtilen özellikler nedeniyle Julia, 2022 StackOverflow geliştirici anketinde en sevilen diller arasında ilk 5'te yer almıştır. Aşağıda, Julia'nın bazı önemli avantaj ve dezavantajları verilmiştir (Julia 1.8 Documentation, çevrimiçi: <https://docs.julialang.org/en/v1/>).

- Daha hızlı çalışma süresi performansı için Julia, LLVM derleyici çerçevesi kullanmaktadır.
- JIT ile derlenmektedir.
- Hız konusunda Julia, çok iyi performans göstermektedir.
- Julia paralellik ve dağıtılmış hesaplama için çok uygundur. Bu bilimsel hesaplama, paralel bilgi işlem ve minimum çaba ile sınırsız ölçeklenebilirlik sağlar. Bu sayede, Julia uygulamaları, bir düğmeye tıklanarak büyük kümelere dağıtılabilir ve on binlerce düğümde hızlı ve kolay bir şekilde paralel ve dağıtılmış işlem gerçekleştirilebilmektedir.
- Julia'yı öğrenmek kolaydır.
- Julia, REPL (read-eval-print loop) veya etkileşimli komut satırı içermektedir.
- Julia iki dil problemini çözer. Bu, zamandan tasarruf sağlar, hata ve maliyeti azaltmaktadır.
- Etkileyici, güçlü ve dinamik söz dizimine sahiptir.
- Julia, işlemlerin kolayca yönetilmesini sağlayan güçlü bir kabuğa sahiptir.
- Julia dinamik yazmanın ve statik yazmanın avantajlarını bir araya getirmektedir.
- Birçok programlama dilinin kütüphanelerini kolaylıkla kullanabilir.
- Lisp'de desteklenen makroların benzerleri Julia'da da yer almaktadır.

- Julia metaprogramlamayı desteklemektedir.
- Çok başarılı bir hata ayıklama paketine sahiptir.
- Julia'nın matematik hesaplamalar için kullanılan söz dizimi daha çok matematik formüllerinin yazılma biçimine benzemektedir. Bu da programcı olmayanların Julia'da kod yazmaya alışmasını kolaylaştırmaktadır.
- Julia'da kullanıcı performans için bellek yönetimi gibi ayrıntılarla uğraşmak zorunda değildir.
- Julia farklı argüman türleri için otomatik, verimli ve özel kod oluşturma imkanı sağlar.
- Açık bir sözdizimine sahiptir.
- Çoklu gönderim yardımıyla kullanıcı, birçok argüman kombinasyonunda fonksiyon davranışını tanımlayabilmektedir.
- Çoklu gönderim ve buna benzer bir dizi özelliği, onu birçok alanda kullanılabilir kılmaktadır.
- Çok yönlü genel bir programlama dili olması nedeniyle Julia'nın pek çok kullanım alanı var.
- Julia, Unicode için verimli bir destek sağlamaktadır.
- Julia'da performans için kodun vektörleştirilmesine gerek yoktur. Bu da Julia'nın hızlı olmasına katkı sağlamaktadır.
- Julia sıfırdan matematiksel, bilimsel ve istatistiksel işlemler için üretilmiştir. Programlamayı basit ve hızlı hale getiren ve gerekli kod satırı sayısını önemli ölçüde azaltan (bazı durumlarda %90'lara kadar) gelişmiş bir çok kütüphaneye sahiptir.

Julia'nın Dezavantajları

- Nispeten yeni olmasının bir sonucu olarak, topluluk ve ekosistem hala küçüktür (giderek büyüyor).
- Yeni bir dil olması sebebiyle teknik kusurlar barındırmaktadır.
- Pek çok benzersiz çözüm paketi olmasına karşın, diğer dillerle karşılaştırıldığında, bu sayı oldukça azdır.
- Bazı derleme sorunları vardır. Tam zamanında (JIT) derleyicisi kullanılarak kod derlerken gecikmeler yaşanabilmektedir.

2.3. Julia Programlama Dilinin Kullanım Alanları

Julia'nın geliştiricilerinin temel amaçları, matematiksel formülleri andıran açık, yüksek seviyeli, genel ve soyut kod yazmaya olanak tanıyan, genellikle hızlı, düşük seviyeli makine kodu üretebilen bir dil üretmektir. Bu amaç çerçevesinde zaman içinde oldukça aşama kaydedilmiş ve Julia, aşağıda yer alan alanlarda sıklıkla kullanılmaya başlamıştır (Crabtree, 2022).

a. Sayısal Hesaplama

Julia'da sayısal işlem görevlerini gerçekleştirmeye olanak tanıyan bir dizi paket vardır. Örneğin lineer cebir ile ilgili görevler için "LinearAlgebra.jl", kök bulma için "NonlinearSolve.jl" modülü vardır.

b. Makine Öğrenimi

Julia'da makine öğrenmesi, veri kümelerinden gelen zorlukları ve içgörülerini dikkate alarak inşa edilmiştir. Paralellik için tasarlanmış ve derlenmiş bir dil olarak Julia, makine öğrenme süreçlerini hızla gerçekleştirebilmektedir. Örneğin Julia, makine öğrenme modellerinin eğitim sürelerini önemli ölçüde iyileştirerek, para ve zaman maliyetlerini azaltmaktadır. "MLJ" makine öğrenimi paketiyle Julia, bu alan için önemli bir seçenek. Örneğin ülkedeki elektrik hatları bakımını yapan Avustralyalı bir şirket, 3B harita oluşturup, Julia makine öğrenimini kullanarak, olası arızaları eskisinden 100 kat daha hızlı tespit etmektedir.

c. İstatistikler

Julia'nın ana işlevlerinden bir diğeri de istatistik ve istatistiksel analizlerdir. Örneğin, "StatsKit" meta paketi, temel istatistik işlevlerinin yanı sıra "DataFrame" tablo verileri, olasılık dağılımları, çok değişkenli istatistiksel analiz, hipotez testleri ve daha fazlasına ilişkin araçlar sunmaktadır.

d. Genel Programlama

Genel amaçlı bir programlama dili olarak Julia, çok çeşitli yazılım ve işlevler oluşturma yeteneğine sahiptir. Örneğin, "Blink" paketi Electron GUI uygulamaları oluşturulmasına izin vermektedir. "GameZero", yeni başlayanlar için oyun geliştirme imkanı sunmaktadır. "Miletus" ise, Julia'da finansal sözleşmeler oluşturulmasına olanak tanımaktadır.

e. Web Geliştirme

Julia'da web geliştirme için "Genie.jl" ve "Dash.jl" gibi kullanıcılara sunulan araçlar da bulunmaktadır. "Genie", kullanıcıların Julia'da web üzerinden interaktif olarak verilerle çalışmasına izin vermektedir. Kullanıcılara akıcı ve verimli bir iş akışıyla, web uygulamaları geliştirme yolu sağlayan bir web çerçevesidir. "Dash.jl" ise, Julia'da web uygulamaları oluşturmanıza olanak tanıyan, Dash ekosistemine yönelik bir Julia arayüzüdür.

f. Farmakoloji Modelleme ve Geliştirme

Julia tıp alanında yaygın olarak kullanılmaktadır. Araştırmacılar Julia'yı yeni ilaçları belirlemek, ilaçları test etmek, hastalıkları teşhis etmek ve tedavileri optimize etmek için kullanmaktadır. Bu alanlar için Julia'da, "BioJulia" (bir biyoinformatik altyapısı) ve "Pumas" (farmakometri modellemesi için, nitelikli ilaç deneme bilgisi bilimi) gibi paketler yer almaktadır. Örnek olarak, ilaç devi Pfizer, Julia'yı yeni tedavilerin simülasyonlarını hızlandırmak için kullanmaktadır.

g. Paralel Süper Hesaplama

Julia'nın hızının en iyi örneklerinden biri, astronomik ve astrofiziksel rutinlerden oluşan "AstroLib.jl" paketinin sıklıkla kullanıldığı astronomi alanıdır. Julia'nın astronomide kullanımına bir örnek, NASA'da astronomlar, fizikçiler, bilgisayar mühendisleri ve istatistikçilerden oluşan bir araştırma ekibinin Julia'yı kullanarak yeni bir paralel hesaplama yöntemi geliştirdiği "Celeste" projesidir. Araştırmacılar üç yıllık geliştirmenin ardından, bu proje sayesinde 188 milyon astronomik nesnenin en doğru kataloğunu 1.000 kattan fazla performans artışı sağlayarak yalnızca 14,6 dakikada üretme başarısı göstermişlerdir.

h. Finansal Modelleme ve Yönetim

Julia'nın performansı ve esnekliği, onu finans dünyasında da güçlü bir araç haline getirmektedir. Profesyoneller, finansal kararlar almak, modeller ve tahminler geliştirmek için Julia programlama dilini sıklıkla kullanmaktadırlar. Bu alanda önde gelen paketlerden biri, varlık yönetimi, risk yönetimi, algoritmik ticaret, geriye dönük test ve finansal sözleşmelerin modellenmesi de dahil olmak üzere hesaplamalı finansın diğer alanlarında uzmanlaşmış "JuliaFin"dir. Örneğin, Brezilya'nın ulusal kalkınma bankası (BNDES) ücretli bir çözümden vazgeçip açık kaynaklı Julia'yı tercih ederek 10 kattan fazla hız kazanmıştır.

i. Robotik

MIT'deki robotistler esnekliği nedeniyle, robotları programlamak için Julia'yı kullanmaktadırlar. Bu alandaki popüler paket koleksiyonlarından biri, hareket yakalama verilerinden (yürüme, merdiven çıkma ve zorlu ve tehlikeli arazileri aşma) robotlar için kinematik kalibrasyon sağlayan "MotionCaptureJointCalibration.jl" gibi Julia paketlerini içeren "JuliaRobotics" paket koleksiyonudur.

j. Ağ Güvenliği

Cisco'daki ağ uzmanları, ağ güvenliğini iyileştirmek ve makine öğrenimi için "Flux.jl", Julia paketini kullanmaktadırlar. Tek bir dilde verimli kod yazabilme yeteneği, Julia'yı bu alanda kullanmanın ana faydalarından biri olarak gösterilmektedir.

k. İklim

NASA Jet Propulsion Laboratuvarı ve Donanma Yüksek Lisans Okulu ile işbirliği içinde başlatılan CliMA (İklim Modelleme Birliği), kuraklıkları, sıcak dalgaları ve yağışları, büyük bir hassasiyetle tahmin edebilen bütüncül bir sistem modeli geliştirmek için Julia'yı kullanmaktadır.

Bütün bunların yanı sıra;

- ABD Federal Havacılık İdaresi (FAA), Julia'yı kullanarak bir Havadan Çarpışma Önleme Sistemi (ACAS-X) geliştirmek,

- Brezilya uydusu Amazonia-1'in Tutum ve Yörünge Kontrol Sistemi'nin (AOCS) tamamı,

- Voxel8, 3 boyutlu baskı,

- Zipline, dünya çapında hayat kurtaran acil tıbbi malzemeleri drone aracılığıyla ulaştırmak için uçak simülasyonu,

- United Therapeutics, akciğerleri etkileyen hastalıklar da dahil olmak üzere nadir hastalıklara yönelik tedaviler geliştirmek,

- Timeline, Web tabanlı finansal planlama sağlamak,

- Tangent Works, prototip oluşturma ve üretim arasındaki engeli ortadan kaldıran kapsamlı bir analitik çözüm üretmek,

- Araştırmacılar, Julia kullanarak küresel genetik çeşitliliğin haritasını çıkarmak, için Julia'yı kullanmaktadır (JuliaHub, Çevrimiçi: <https://juliahub.com/case-studies/>).

3. Julia ve Veri Biliminde Kullanılan Diğer Programlama Dilleri

Veri bilimi projeleri üzerinde çalışırken, genel ve özel bir amaçlı olarak kullanılabilecek Python, R ve SQL, MATLAB, C/C++ ve Go gibi çok çeşitli programlama dili seçeneği vardır. Ancak veri bilimi alanında en çok kullanılan ilk 2 programlama dili Python ve R'dır. Bu sıralamada Julia ilk 5'te yer almaktadır. Julia'yı kullanan geliştiriciler arasında kullanılan ikinci en popüler dilin, anketlerde %61 gibi bir oranla Python olduğu görülmüştür. Katılımcıların yaklaşık %25'inin beğendiklerini bildirdikleri diller arasında R bulunmaktadır. Dolayısı ile bu alanda Julia'nın rakipleri olarak Python ve R düşünülebilir. Bu nedenle çalışmanın amacına uygun olarak burada veri bilimi alanında en çok kullanılan bu iki programlama dili (Python ve R) ile Julia karşılaştırılacaktır. Bu üç programlama dilinin ortak noktası 3'ünün de açık kaynak kodlu olması (yani ticari amaçlar için bile kullanmak ve

dağıtmak ücretsizdir), hemen hemen tüm işletim sistemlerine ilişkin sürümlerinin bulunması ve özellikle veri bilimi alanına odaklanmasıdır.

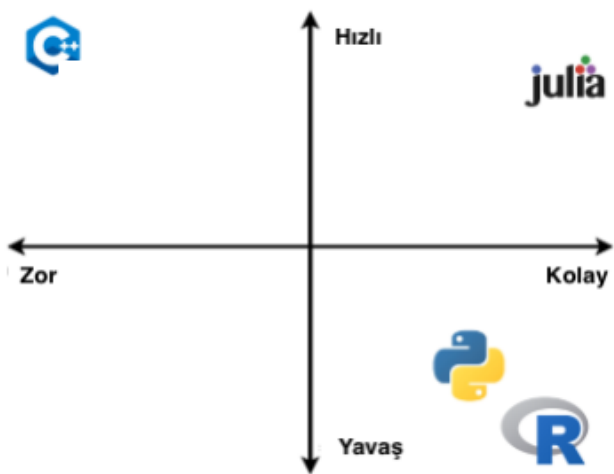
Veri bilimcilerin hedeflerinden biri, programlama alanında en hızlı ve performanslı dillerden biri olan “C++” programlama dilini yakalamaktır. Bu nedenle programlama dilleri hız ve performans açısından genellikle C++ ile karşılaştırılırlar. Bunun yanı sıra bir hız ve performansın yanı sıra bir programlama dilinden beklenen diğer unsur da kod yazımının kolay olmasıdır.

Aşağıda veri bilimi alanında kullanılan açık kaynak kodlu dillerin, C++ ile karşılaştırması yer almaktadır. Bu diller, eksenlerden birinde hızın diğerinde kullanım kolaylığının yer aldığı 2 boyutlu uzayda gösterilmiştir (Storopoli vd., 2022, s.11).

C++ III’üncü bölgede yani zor ve hızlı olan diller arasında yer almaktadır. Bu dillerin derlenmesi, tür denetimi ve diğer profesyonel işlemler için dikkat gerektiren statik diller sınıfında yer almaktadırlar. Bu nedenle öğrenilmesi ve prototip oluşturulması zor olan dillerdir. Bu dillerin avantajı, gerçekten hızlı diller olmalarıdır.

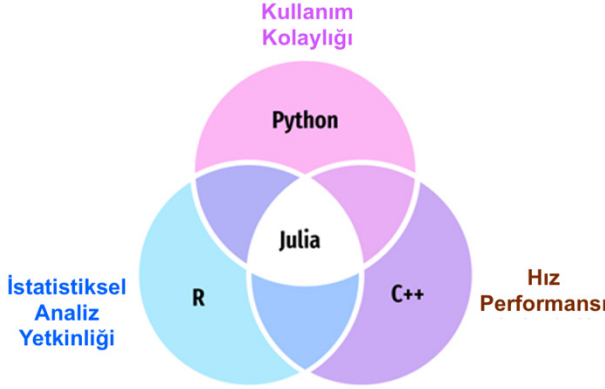
R ile Python kolay ve yavaş diller sınıfında yani IV’cü bölgede yer almaktadır. Derlenmemiş dinamik dillerdir ve çalışma zamanında yürütülürler. Bu nedenle, bu diller öğrenmesi çok kolay ve hızlı bir şekilde prototip oluşturulabilen dillerdir. Ancak bu diller de çok yavaş dillerdir.

Julia, kolay ve hızlı olan I’inci bölgedeki tek dildir. Veri bilimi alanında öğrenmesi zor ve hızlı yavaş olan herhangi bir dil bulunmamaktadır. Böyle bir dil olsa idi, bu dilin kullanıcı sayısının çok az olacağı düşünülmektedir.



Şekil 1. Bilimsel Bilgi İşlem Dillerinin Karşılaştırmaları

Bu diller arasında Julia'nın yerini göstermek için aşağıdaki görsel oldukça faydalı olacaktır (Selvaraj, 2022). Görselde görüldüğü gibi, kullanımı en kolay olan programlama dili Python, istatistiksel analize en yatkın olan dil R ve en hızlı olan dil ise C++'dır. Julia bu iyi ve önemli özelliklerin hepsini tek başına sunmaktadır. Diğer bir ifade ile, Julia, Python'un basitliğine, R'nin istatistiksel yeteneklerine ve C++ hızına sahip bir programlama dilidir.



Şekil 2. C++, Python, R ve Julia'nın hız, istatistiksel analiz yetkinliği ve kullanım kolaylığı açısından karşılaştırılması

Julia, Python benzeri kodu, programcıya C++ programlama dili ile aynı performansı veren makine kodunda derler. Julia ve Python'un performansı karşılaştırıldığında, Julia Python'a göre 10 ila 30 kat daha hızlıdır. Julia, sayısal hesaplama için python programlama dilinden daha basit bir sözdizimine sahiptir. Bu nedenle Julia, python dilinden daha kolay ve basit bir şekilde kodlanabilmektedir.

İstatistiksel analiz alanında, R en iyi programlama dillerinden biridir. Ancak Julia, R'dan 10 ila 1000 kat daha hızlıdır ve istatistiksel analiz için en az R kadar uygundur. Bunun yanı sıra R lineer cebir yapmak için uygun değildir, ancak Julia lineer cebir için de çok uygundur. Öte yandan, Julia'nın söz dizimi R ile karşılaştırıldığında, çok daha zengindir.

Julia'da; Python, R, C, ve Java gibi diğer programlama dilleriyle verimli bir şekilde etkileşim kurabilmektedir. Kullanıcı herhangi bir özel API (Application Programming Interface/Uygulama Programlama Arabirimi) olmadan C fonksiyonlarını çağırabilir. Ancak diğer dillerde (Python ve R'de), bu çok kolay değildir. Örneğin, Julia'da, Python kodları PyCall paketi, R kodları RCall paketi ve Java kodları da JavaCall paketi ile kullanılabilir. Diğer bir ifade ile, Julia, mevcut kod ve platformlarla iyi bir şekilde bütünleşir. Diğer bir ifade ile C, C++, Python, R ve diğer dillerin kullanıcıları, mevcut kodlarını Julia'ya kolayca entegre edebilirler.

Aşağıda, Julia'nın en çok kullanılan veri bilimi programlama dilleri olan R ve Python ile karşılaştırması yer almaktadır. Üç dil arasında bir çok teknik fark olmasına karşın, çalışmada aralarındaki en önemli farklar belirtilmiştir (Ratmelia, 2022).

Tablo 1
Python, R ve Julia'nın önemli özelliklerinin karşılaştırılması

Özellikler	Python	R	Julia
Yayın tarihi	1991	1993	2012
Sözdizimi okunabilirliği	Doğal insan diline yakın olduğundan okunabilirliği kolaydır.	Python ve Julia'a göre daha zordur.	Python ve R'dan daha kolaydır.
Veri Bilimi paketleri	R ve Julia'dan daha fazla yerleşik paketi vardır.	Julia'ya göre daha çok yerleşik paketi vardır.	Python ve R'ye kıyasla daha az yerleşik paketi vardır, ancak sürekli olarak geliştiriliyor. R ve Python paketlerini kullanabiliyor.
Öne çıkan avantajlar	-Yaygın kullanım -Makine öğrenimi -Yapay zeka -Güçlü destek -Bilgi güvenliği -Oyun -Web geliştirme	-Veri madenciliği, -İstatistik ve sayısal analiz -Güçlü destek -Akademide yaygın kullanım	-Veri bilimi -İstatistik ve sayısal analiz -Makine öğrenimi -Kolay kodlama -Hız -Performans -Diğer dillerin paketlerini kullanabilme
Yeni Başlayanlar İçin Öğrenme Desteği	Python, güçlü bir topluluk desteğine sahiptir ve onlarca yıldır kullanılmaktadır, bu nedenle pek çok öğretici kaynak mevcuttur.	R, güçlü bir topluluk desteğine sahiptir ve onlarca yıldır kullanılmaktadır, bu nedenle pek çok öğretici kaynak mevcuttur.	Julia, Python ve R'dan nispeten yenidir, bu nedenle öğreticiler kaynaklar mevcut olsa da, Python veya R kadar çok değildir.
Tip	Python yorumlanmış bir dildir.	R yorumlanmış bir dildir.	Julia derlenmiş bir dildir.
İnternet sitesi	Web sitesi bağlantısı: https://www.python.org/	https://cran.r-project.org/	Web sitesi linki: https://julialang.org/
Kod dönüştürme	Python'un diğer dil kodlarından kod dönüştürmesi zordur.	R'da diğer dil kodlarından kod dönüştürmesi zordur.	Julia, kodu diğer dil kodlarından kolayca dönüştürür.

Bütün bu bilgiler çerçevesinde istatistik ve veri bilimi için hangi programlama dilini öğrenmek ve/veya kullanmak daha iyi olacaktır? Bu sorunun cevabı şüphesiz duruma göre değişir. Herhangi bir programlama dili bilinmiyorsa, Julia'yı ilk programlama dili olarak öğrenmek faydalı bir tercih olabilir. Bunun yanı sıra, veri bilimi ile uğraşanların başka bir programlama dili bilse de Julia öğrenmesinde yarar vardır. Çünkü Julia'nın veri biliminde kullanılan diğer programlama dillerine göre yukarıda belirtilen (hız, sözdizimi kolaylığı, entegre edilebilirlik ve performans gibi) üstünlükleri söz konusudur.

Diğer taraftan seçim yaparken aşağıda belirtilen unsurları da dikkate almak fayda sağlayacaktır.

Trend ne yönde? Pragmatik yaklaşım, mevcut çalışma/araştırma alanınızda hangi dil daha popüler ise onunla devam etmektir. Bu konuda arkadaşlarınızdan ve araştırmacılardan bilgi alabilirsiniz. Diğer taraftan alanınızdaki araştırma makalelerine bakarak bu araştırmalarda hangi programlama dillerinin kullanıldığını incelenerek programlama dili trendlerini takip edebilirsiniz.

Gerçekleştirilmek istenen işlemler neler? Gerçekleştirmek istenilen temel unsurların neler olduğu belirlenmeli ve ona göre karar verilmelidir. Çalışmaların çoğu karmaşık istatistiksel analiz içeriyorsa, önce R'yi öğrenmek muhtemelen daha iyi bir seçim olacaktır. Makine öğrenme ve metin madenciliği yapılmak isteniyorsa, o zaman belki de önce Python veya Julia öğrenmek daha faydalı olacaktır. Popüler olan programlama dilini öğrenmek daha güvenli bir tercih gibi görünebilir, ancak bazen popüler olan dil, çözülmek istenen problemi çözmek için en iyi araç olmayabilir.

Yeterince öğrenme kaynağı var mıdır? Programlamaya yeni başlanmışsa, mevcut destek göz önünde bulundurulmalıdır. Python ve R daha eski ve yerleşik dillerdir. Julia'dan daha fazla desteğe ve daha fazla öğrenme kaynaklarına sahiptirler. Ancak giderek Julia'nın da öğrenme kaynakları artmaktadır.

Sonuç

Bilimsel programlama ve veri bilimi ile ilgili olarak kullanıcılar; yüksek performanslı, hızlı, kod yazım ve protipleme kolaylığına sahip esnek ve dinamik programlama diline ihtiyaç duymaktadırlar. Ancak Julia dışındaki programlama dilleri bu ihtiyaçların hepsine birden cevap verememektedir.

Julia, programcılara üst düzey bir programlama dilinde kütüphaneler geliştirebilecekleri yeni bir teknik bilgi işlem ortamı sunmaktadır. Bu dil, herhangi bir uygulamayı yazmak için kullanılabilen genel amaçlı, yüksek performanslı, esnek, öğrenmesi kolay ve güçlü bir programlama dilidir. Bilimsel ve sayısal hesaplama için çok uygundur. Julia, bilimsel bilgi işlem, makine öğrenimi ve veri madenciliği için ideal dil olmak üzere tasarlanmış nispeten yeni bir programlama dilidir. 2012 yılında piyasaya sürülen Julia'nın kullanımı, deneyimli veri bilimcileri arasında hızla artmaktadır.

Julia'nın en önemli özellikleri hızı, performansı ve kullanım kolaylığıdır. C++ performansına, Python'un genel programlama yeteneğine ve R'nin istatistiksel gücüne sahip olacak şekilde tasarlanmıştır. Julia'nın öne çıkan özelliklerinden biri de Python, R, Java ve C gibi diğer programlama dillerinde yazılmış kodları PyCall, RCall, JavaCall gibi paketler yardımıyla kullanabilmesidir. Bu sayede bu dillerde yazılmış kodlar Julia'da çağrılarak çalıştırılabilmektedir. Diğer taraftan Julia'yı öğrenmek isteyenler için, diğer diller kadar olmasa da hızla gelişen ve bu dilin temel özelliklerini öğreten referans niteliğinde basılı ve online bir çok kaynak

vardır. Ancak yeni sayılabilecek bir programlama dili olması sebebiyle teknik olarak dezavantajlarının da olduğu söylenebilir. Python gibi daha köklü yüksek seviye diller gibi Julia'nın da daha stabil olması için biraz daha gelişme göstermesi gerekmektedir.

Bilindiği gibi, veri bilimi alanında en çok kullanılan programlama dilleri, Python veya R'dır. Yeni bir dil olması nedeniyle Julia'nın hemen bu iki dilin yerini alması beklenmese de, veri bilimi alanında bunlardan çok daha fazla üstünlüklere (hız/performans/kullanım kolaylığı/iki dil sorununu ortadan kaldırması gibi) ve gelişme potansiyeline sahip olması nedeniyle Julia'daki gelişmelerin yakından takip edilmesinde ve hatta Julia'da kodlamanın öğrenilmesinde büyük yarar olacaktır.

Diğer taraftan unutulmamalıdır ki, kodlama dili yalnızca bir araçtır. Araçlar değişip gelişir, belirli bir süre yaygın şekilde kullanılabilir ve bir müddet sonra popüleritesini kaybederek kullanılmamaya başlar. Bu nedenle araçlara çok fazla odaklanmaktansa, bilim alanı temellerine (yani istatistik/veri bilimi için alan bilgisine) ve en önemlisi bunların nasıl uygulanacağına odaklanmak daha etkili olacaktır. Python, R veya Julia'da nasıl kod yazılacağını bilerseniz de, veri bilimi veya istatistik konusunda yetkin değilseniz, sorunun nasıl belirleneceği veya doğru soruların nasıl sorulacağı bilinmiyorsa, tek başına kodlama bilgisinin pek bir faydası olmayacaktır. Buna karşın, istatistik ve veri bilimi alanında iyi temellere sahip olmak, herhangi bir programlama dilini öğrenmeyi daha kolay hale getirecektir.

Hakem Değerlendirmesi: Dış bağımsız.

Çıkar Çatışması: Yazarlar çıkar çatışması bildirmemiştir.

Finansal Destek: Yazarlar bu çalışma için finansal destek almadığını beyan etmiştir.

Yazar Katkısı: Çalışma Konsepti/Tasarımı: M.T., Ü.H.Ö.; Veri Toplama: M.T., Ü.H.Ö.; Veri Analizi /Yorumlama: M.T., Ü.H.Ö.; Yazı Taslağı: M.T., Ü.H.Ö.; İçeriğin Eleştirel İncelemesi: M.T., Ü.H.Ö.; Son Onay ve Sorumluluk: M.T., Ü.H.Ö.

Peer-review: Externally peer-reviewed.

Conflict of Interest: The authors has no conflict of interest to declare.

Grant Support: The authors declared that this study has received no financial support.

Author Contributions: Conception/Design of study: M.T., Ü.H.Ö.; Data Acquisition: M.T., Ü.H.Ö.; Data Analysis/ Interpretation: M.T., Ü.H.Ö.; Drafting Manuscript: M.T., Ü.H.Ö.; Critical Revision of Manuscript: M.T., Ü.H.Ö.; Final Approval and Accountability: M.T., Ü.H.Ö.

Kaynakça/References

- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2012), Why We Created Julia. Çevrimiçi: <https://julialang.org/blog/2012/02/why-we-created-julia/>, Erişim tarihi 23.12.2022.
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98.
- Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209.
- Crabtree, M. (2022). What is Julia Used For? 10 Applications of Julia Programming. Çevrimiçi: <https://www.datacamp.com/blog/what-is-julia-used-for>, Erişim tarihi: 22.12.2022.

- JuliaHub, (2022). Case Studies. Çevrimiçi: <https://juliahub.com/case-studies/>, Erişim tarihi: 22.12.2022.
- JuliaLang, (2022). Julia 1.8 Documentation, Çevrimiçi: <https://docs.julialang.org/en/v1/>, Erişim tarihi: 20.12.2022.
- Khan, N., Yaqoob, I., Hashem, I. A. T., Inayat, Z., Mahmoud Ali, W. K., Alam, M., Shiraz, M., & Gani, A. (2014). Big data: Survey, technologies, opportunities, and challenges. *The Scientific World Journal*, 2014.
- Lauwens, B. & Allen, D. (2019). *Think Julia: How to Think Like a Computer Scientist*. O'Reilly Media ISBN: 9781492044987.
- Perkel, J. M. (2019). Julia: Come for the syntax, stay for the speed. *Nature*, 572(7767), 141–142. <https://doi.org/10.1038/d41586-019-02310-3>
- PYPL, Çevrimiçi: <https://pypl.github.io/PYPL.html>, Erişim tarihi: 23.12.2022.
- Ratmelia, B. , (2022). Julia, Python, and R: What's the difference?. Çevrimiçi: <https://library.smu.edu.sg/topics-insights/julia-python-and-r-whats-difference>, Erişim tarihi: 20.12.2022).
- Selvaraj, N. (2022). Meet Julia: The Future of Data Science. Çevrimiçi: <https://towardsdatascience.com/meet-julia-the-future-of-data-science-52414b29ebb>, Erişim tarihi: 22.12.2022.
- Storopoli, J., Huijzer, J. & Alonso, L. (2022). *Julia Data Science*. ISBN: 9798489859165. Published <https://juliadatascience.io>.
- TIOBE, Çevrimiçi: <https://www.tiobe.com/tiobe-index/>, Erişim tarihi: 23.12.2022.