

## KATTA HAJMDAGI MA'LUMOTLARNI QAYTA ISHLASHDA PYTHON, R VA JULIA DASTURLASH TILLARINING ISHLASH TEZLIGI VA SAMARADORLIGINI TAQQOSLASH

**Abdug'opporov Xabibullo Xolmurod o'g'li**

FarDU Axborot tizimlari va texnologiyalari yo'nalishi 3-kurs talabasi  
[habibullo2313@gmail.com](mailto:habibullo2313@gmail.com)

**Bakhtiyor Abdukadirov**

FarDU Axborot texnologiyalari kafedrası dotsenti, t.f.b.f.d. (PhD)  
[bakhtiyor.uz@gmail.com](mailto:bakhtiyor.uz@gmail.com)

<https://doi.org/10.5281/zenodo.19950486>

**Annotatsiya:** Ushbu maqolada katta hajmdagi ma'lumotlarni qayta ishlash jarayonida keng qo'llaniladigan Python, R va Julia dasturlash tillarining ishlash tezligi, xotira samaradorligi va umumiy samaradorligi keng qamrovda tahlil qilinadi. Tadqiqot davomida har bir tilning texnik xususiyatlari, ularning Big Data muhitidagi o'rni hamda real amaliy vazifalarda qo'llanish darajasi o'rganildi. Shuningdek, katta hajmdagi ma'lumotlar bilan ishlashda yuzaga keladigan asosiy muammolar aniqlanib, ularni bartaraf etish uchun zamonaviy yondashuvlar taklif etildi. Tadqiqot natijalari shuni ko'rsatadiki, turli dasturlash tillarining integratsiyasi umumiy samaradorlikni sezilarli darajada oshiradi.

**Kalit so'zlar:** Data Science, Big Data, Python, R, Julia, ishlash tezligi, samaradorlik, parallel hisoblash, ma'lumotlar tahlili, benchmark

**Abstract:** This paper presents a comprehensive analysis of the performance, memory efficiency, and computational speed of Python, R, and Julia programming languages in processing large-scale datasets. The study explores their technical characteristics and evaluates their effectiveness in Big Data environments. Additionally, key challenges associated with large-scale data processing are identified, and modern solutions are proposed. The results demonstrate that integrating multiple programming languages significantly enhances overall efficiency.

**Keywords:** Data Science, Big Data, Python, R, Julia, performance, efficiency, parallel computing, benchmarking

**Аннотация:** В данной статье представлен расширенный анализ производительности, скорости работы и эффективности языков программирования Python, R и Julia при обработке больших объемов данных. Рассмотрены их технические особенности и возможности применения в среде Big Data. Выявлены ключевые проблемы обработки больших данных и предложены современные методы их решения. Результаты показывают, что интеграция языков программирования значительно повышает эффективность обработки данных.

**Ключевые слова:** Data Science, Big Data, Python, R, Julia, производительность, эффективность, параллельные вычисления

### **Kirish**

Axborot texnologiyalari jadal rivojlanayotgan hozirgi davrda ma'lumotlar hajmi geometrik progressiya asosida ortib bormoqda. Har bir foydalanuvchi har kuni turli platformalar orqali ma'lumot yaratadi: ijtimoiy tarmoqlar, elektron tijorat tizimlari, sensor qurilmalar, tibbiy monitoring vositalari va boshqa ko'plab manbalar orqali. Natijada "katta

ma’lumotlar” (Big Data) tushunchasi shakllandi. Katta hajmdagi ma’lumotlar bilan ishlash faqatgina ularni saqlash emas, balki tezkor va samarali qayta ishlashni ham talab qiladi. Bu esa yuqori darajadagi hisoblash resurslari va samarali dasturlash vositalarini talab etadi. Aynan shu nuqtada Data Science sohasi shakllanib, u ma’lumotlarni tahlil qilish, qayta ishlash va ulardan foydali xulosalar chiqarishga xizmat qiladi. Python, R va Julia dasturlash tillari Data Science sohasida eng ko’p qo’llaniladigan vositalar hisoblanadi. Ularning har biri turli vazifalar uchun optimallashtirilgan bo’lib, samaradorlik nuqtai nazaridan sezilarli farqlarga ega. Ushbu maqolada aynan shu farqlar chuqur tahlil qilinadi.

### Metodlar

Mazkur tadqiqotda kompleks yondashuv qo’llanildi. Birinchi bosqichda nazariy tahlil orqali Python, R va Julia tillarining arxitekturasi va ishlash mexanizmlari o’rganildi. Ikkinchi bosqichda esa taqqoslash metodi yordamida ularning asosiy ko’rsatkichlari solishtirildi.

Amaliy tahlil sifatida quyidagi turdagi vazifalar asos qilib olindi:

- Katta hajmdagi datasetlarni yuklash (10 mln+ yozuv)
- Ma’lumotlarni filtrlash va saralash
- Statistik hisob-kitoblar (o’rtacha qiymat, dispersiya)
- Mashinaviy o’rganish modelini ishga tushirish

Har bir vazifa uchta dasturlash tilida bajarilib, quyidagi mezonlar bo’yicha baholandi:

- Bajarilish vaqti (sekundlarda)
- Xotira sarfi (MB/GB)
- CPU yuklanishi
- Parallel ishlash imkoniyati

### Natijalar

Tahlil natijalari quyidagi muhim xulosalarni ko’rsatdi. Birinchidan, Python o’zining moslashuvchanligi va keng ekotizimi bilan ajralib turdi. U katta hajmdagi ma’lumotlarni qayta ishlashda Pandas va NumPy orqali samarali ishladi, lekin sof hisoblash tezligi Julia’dan past bo’ldi. Ikkinchidan, R statistik tahlilda yuqori aniqlik ko’rsatdi. Ayniqsa murakkab matematik modellarda uning natijalari aniq va ishonchli bo’ldi. Biroq katta hajmdagi ma’lumotlarda xotira muammosi yuzaga keldi. Uchinchidan, Julia barcha hisoblash testlarida eng yuqori tezlikni namoyish etdi. Ayniqsa parallel hisoblashda u sezilarli ustunlikka ega bo’ldi.

**Quyidagi jadval umumiy natijalarni ifodalaydi:**

Ko’rsatkich	Python	R	Julia
10 mln yozuvni qayta ishlash vaqti	12 sek	18 sek	5 sek
Xotira sarfi	O’rtacha	Yuqori	Past
Parallel ishlash	O’rtacha	Past	Yuqori
ML model tezligi	Yaxshi	O’rtacha	Juda yaxshi
Umumiy samaradorlik	Yaxshi	O’rtacha	Yuqori

### Muhokama

Natijalar asosida shuni aytish mumkinki, katta hajmdagi ma’lumotlarni qayta ishlashda asosiy muammo — bu vaqt va resurslar o’rtasidagi muvozanatdir. Python va R tillari qulaylik va aniqlikni ta’minlasa-da, tezlik borasida cheklovlarga ega. Julia esa bu muammoni

samarali hal qiladi, ammo uning keng qo'llanilmasligi va kutubxonalar bazasining nisbatan kichikligi uni to'liq ustun qilmaydi.

Shu sababli, optimal yechim sifatida gibril model taklif etiladi:

- Python → ma'lumotlarni tayyorlash va AI modellar
- R → statistik chuqur tahlil
- Julia → tezkor hisoblash

Bu yondashuv tizimning umumiy samaradorligini oshiradi va resurslardan optimal foydalanish imkonini beradi.

### **Xulosa**

Mazkur tadqiqot natijalari shuni ko'rsatdiki, katta hajmdagi ma'lumotlarni qayta ishlash jarayonida Python, R va Julia dasturlash tillari alohida va muhim o'rin egallaydi. Har bir til o'zining ichki arxitekturasi, ishlash mexanizmi va funksional imkoniyatlari bilan ajralib turadi hamda ma'lum turdagi vazifalarni bajarishda yuqori samaradorlikni namoyon etadi. Jumladan, Python dasturlash tili o'zining universalligi, keng kutubxonalar bazasi va qulay sintaksisi orqali ma'lumotlarni dastlabki qayta ishlash va mashinaviy o'rganish modellari yaratishda samarali vosita sifatida namoyon bo'ladi. R dasturlash tili esa statistik tahlil va ilmiy hisob-kitoblarda yuqori aniqlikni ta'minlab, ilmiy tadqiqotlar uchun muhim platforma hisoblanadi. Julia dasturlash tili esa yuqori ishlash tezligi va parallel hisoblash imkoniyatlari orqali katta hajmdagi ma'lumotlar ustida murakkab hisoblashlarni qisqa vaqt ichida amalga oshirish imkonini beradi. Tadqiqot davomida aniqlangan asosiy jihatlardan biri shundan iboratki, ma'lumotlar hajmi ortib borishi bilan dasturlash tillarining ishlash tezligi, xotira samaradorligi va resurslardan foydalanish darajasi muhim omilga aylanadi. Ayniqsa, interpretatsiya qilinadigan tillarda (Python va R) katta hajmdagi ma'lumotlar bilan ishlashda samaradorlikning pasayishi kuzatiladi. Shu sababli yuqori samaradorlikka erishish uchun faqat bitta dasturlash tilidan foydalanish ko'pincha yetarli natija bermaydi. Mazkur muammoning yechimi sifatida integratsiyalashgan, ya'ni gibril yondashuvni qo'llash maqsadga muvofiq deb topildi. Ushbu yondashuvda turli dasturlash tillarining kuchli tomonlari birlashtirilib, yagona tizim doirasida qo'llaniladi. Masalan, Python yordamida ma'lumotlarni yig'ish va tayyorlash, R orqali chuqur statistik tahlilni amalga oshirish, Julia yordamida esa yuqori tezlik talab qilinadigan hisoblashlarni bajarish orqali umumiy tizim samaradorligini sezilarli darajada oshirish mumkin.

### **Adabiyotlar, References, Литературы:**

1. McKinney, W. *Python for Data Analysis*. O'Reilly Media, 2022.
2. Wickham, H. *R for Data Science*. O'Reilly Media, 2023.
3. Bezanson, J. et al. *Julia: A Fresh Approach to Numerical Computing*. SIAM Review.
4. VanderPlas, J. *Python Data Science Handbook*.
5. Provost, F., Fawcett, T. *Data Science for Business*.
6. Zaharia, M. *Apache Spark and Big Data Processing*.
7. Dean, J., Ghemawat, S. *MapReduce Framework*.