

Que NO puede hacer Julia:

Al momento de realizar este trabajo, pudimos ver que Julia es un lenguaje con muchos usos. Al ser este un trabajo de investigación, nos preguntamos entonces: ¿Para qué no sirve Julia?

Primero hablemos de las contras en general. En una investigación respecto a Big Data se hizo una comparación entre los lenguajes R, Python y Julia, dando como resultado una serie de desventajas respecto a nuestro lenguaje:

- No está completamente estabilizado
- Herramientas menos científicas
- Más lento

Las bibliotecas de Julia no se han estabilizado por completo y es probable que rompan la compatibilidad con versiones anteriores.

El conjunto de herramientas científicas existentes todavía es sólo una fracción de lo que está disponible en Python.

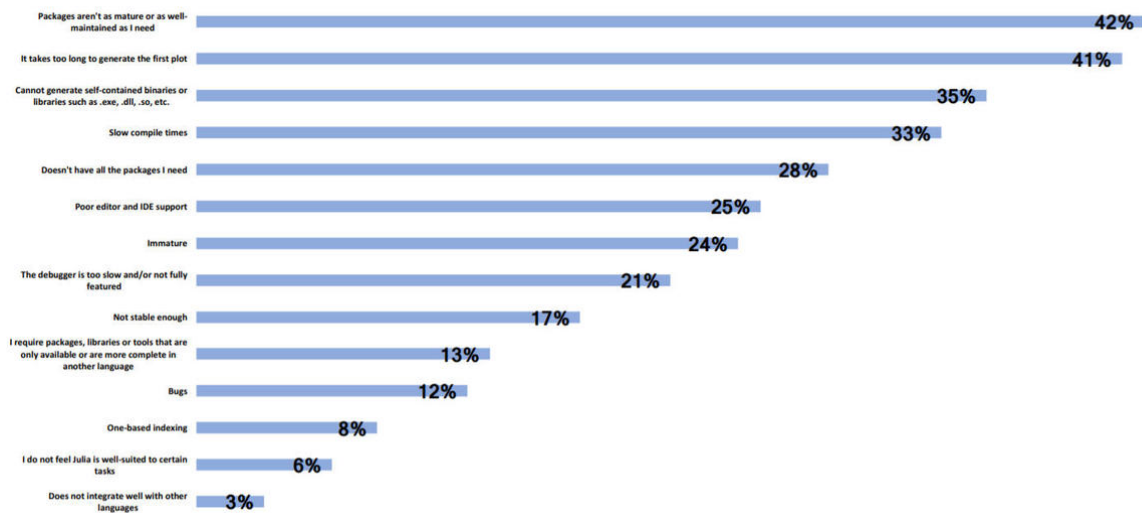
Los diccionarios tienen un hash de forma diferente a los diccionarios de Python, lo que puede hacerlos más lentos en muchos casos.

Luego, respecto a experiencias de usuario, encontramos algo parecido: Los usuarios también informan de sus mayores quejas con el lenguaje. La principal es que las bibliotecas de funciones complementarias no están lo suficientemente maduros o bien mantenidos para satisfacer sus necesidades. Otros problemas importantes son que se tarda demasiado en generar el primer gráfico y que no se puede utilizar para crear bibliotecas o binarios autónomos.

Otro problema es que la comunidad usa otros idiomas y que no hay suficientes usuarios de Julia en su industria o campo. Los usuarios también mencionan la falta de recursos de enseñanza y aprendizaje en línea.

Absolutamente todos estos problemas, personalmente los miembros del grupo los vivimos. Tuvimos muchos problemas con el Google Collab al punto de tener que pasarnos a Pluto. Las bibliotecas tardan demasiado en importarse, los gráficos tardan también bastante en generarse cuando está en frío. Conseguir tutoriales e info del lenguaje es tarea bastante difícil.

Thinking only about the TECHNICAL aspects or features of Julia, what are the TECHNICAL aspects or features you like LEAST about Julia?



El mayor problema técnico de Julia es que los paquetes de funciones complementarias no están lo suficientemente maduros o bien mantenidos para satisfacer las necesidades de los usuarios.

“Pero particularmente molesta es la forma en que los DataFrames se muestran al usuario, y realmente no entiendo las razones para hacer tal cosa de esta manera.”

```
In [1]: 1 # Read in our data!
2 using CSV
3 df = CSV.read("bottle.csv")
4 # That's a lot of observations, which is good, but way too many features.
5 # I really hope DataFrames.JL is improved in the future, because this whole
6 # "omitted printing" thing is probably the biggest downside.
```

Out[1]: 864,863 rows × 74 columns (omitted printing of 68 columns)

	Cst_Cnt	Btl_Cnt	Sta_ID	Depth_ID	Depthm	T_degC
	Int64	Int64	String	String	Int64	Float64
1	1	1	054.0 056.0 19-4903CR-HY-060-0930-05400560-0000A-3		0	10.5
2	1	2	054.0 056.0 19-4903CR-HY-060-0930-05400560-0008A-3		8	10.46
3	1	3	054.0 056.0 19-4903CR-HY-060-0930-05400560-0010A-7		10	10.46
4	1	4	054.0 056.0 19-4903CR-HY-060-0930-05400560-0019A-3		19	10.45
5	1	5	054.0 056.0 19-4903CR-HY-060-0930-05400560-0020A-7		20	10.45

El paquete CSV, vivido en carne propia, es muy débil para hacer ejercicios con dataframes. No es amigable al usuario y si bien Julia sirve para procesamiento de datos, esto retrasa bastante. En la foto podemos ver que se omitieron 68 de 74 columnas (esto también nos pasó a nosotros)

En conclusión, las limitaciones de Julia están determinadas por sus librerías. Esperemos que pronto mejoren, como vimos muchas veces en sus actualizaciones

Fuente:

<https://www.zdnet.com/article/julia-programming-language-users-reveal-what-they-love-and-hate-the-most-about-it/>

<https://towardsdatascience.com/the-serious-downsides-to-the-julia-language-in-1-0-3-e295bc4b4755>

Concurrencia:

Es la división de un problema en subproblemas que se solucionan de forma individual, para crear un programa o aplicación que no se vea afectada en tiempo real.

<https://discourse.julialang.org/t/julia-and-concurrency/25556/2>

<https://docs.julialang.org/en/v1/manual/parallel-computing/>

<https://juliapackages.com/p/actors>