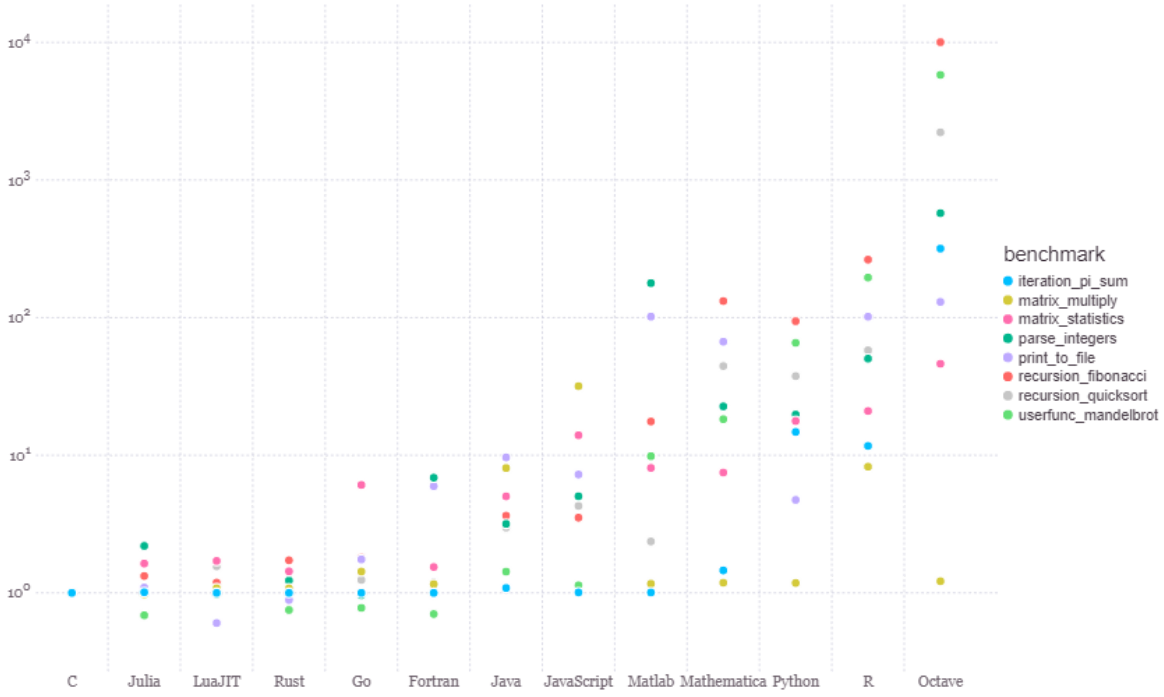
Benchmarks

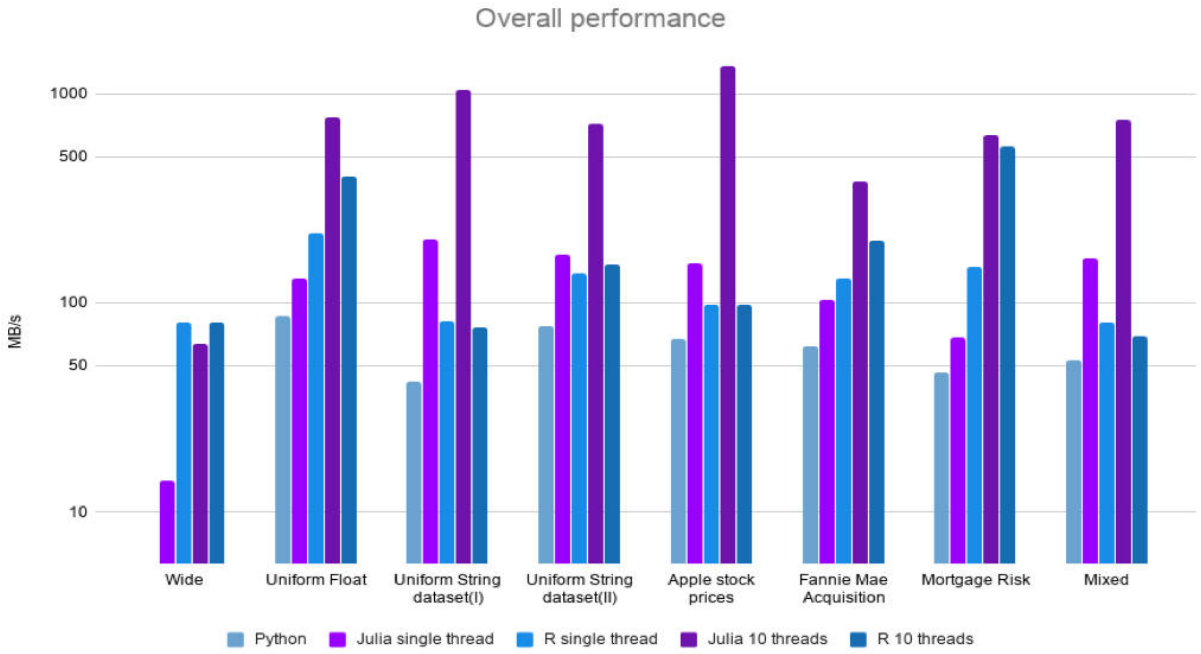
Llega la hora de comparar un poco nuestro lenguaje con otros. Hemos visto que esta caracterizado por su velocidad. Así que veamos unos ejemplos.

Estos micro-benchmarks, aunque no son exhaustivos, prueban el rendimiento del compilador en una variedad de patrones de código comunes, como llamadas a funciones, análisis de cadenas, clasificación, bucles numéricos, generación de números aleatorios, recursividad y operaciones de matriz.
Es importante tener en cuenta que los códigos de referencia no están escritos para un rendimiento máximo absoluto.



El eje vertical muestra cada tiempo de referencia normalizado frente a la implementación de C. Está hecho en un Intel® Core™ i7-3960X 3.30GHz CPU con 64GB of 1600MHz DDR3.

Data Science:
Una actualización reciente de Julia ha mejorado el subproceso múltiple para ofrecer más mejoras de velocidad, y eso es lo que los desarrolladores de Julia argumentan que le da una ventaja considerable sobre Python y el lenguaje de programación estadística R en la tarea de analizar archivos CSV para el análisis de datos.
Deepak Suresh, ingeniero::
"El archivo CSV.jl de Julia es de 1,5 a 5 veces más rápido que Pandas incluso en un solo núcleo; con el subproceso múltiple habilitado, es tan rápido o más rápido que el read_csv de R", señala.



Julia Computing dice que, en los ocho conjuntos de datos, el archivo CSV.jl de Julia es siempre más rápido que Pandas y, con el subproceso múltiple, es competitivo con el archivo data.table de R.

Ultimo ejemplo
“Hice un pequeño punto de referencia comparándolo con C, Python optimizado y Scala. Algunos resultados bastante interesantes, con Julia solo un 3% por detrás de la mejor implementación de C del código. ¡Estoy impresionado!”