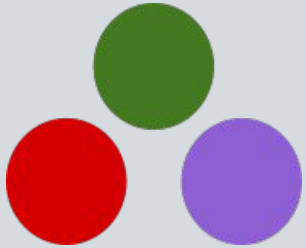
CURSO-TALLER
Introducción a Julia



Eficiencia de alto nivel
Un acercamiento al lenguaje de programación Julia

Easy-Fast

El problema de los dos lenguajes

“You’re led to believe that it’s an immutable law of nature that you can either have a programming language that’s easy to use, or fast. We don’t agree with that.”

Viral Shah, CEO of Julia Computing and co-creator of the Julia programming language

Un proyecto de IA, ML considera...

Matlab: Cálculos numéricos y operaciones de álgebra lineal

R: Cálculos estadísticos

C: Aceleración de funciones y procedimientos

Ruby: Integración todo el código

Dos tipos de lenguajes

Bajo nivel **“system languages”**	Alto nivel **“scripting languages”**
Estáticos	Dinámicos
Compilados	Interpretados
Tipos de datos definidos por usuario	Tipos de datos predefinidos
Autónomos	Integradores

Stefan Karpinski - Solving the "Two Language" Problem in Data Science

Unificación con Julia

Julia

Dinámicos

Compilado

Tipos de datos
predefinidos y creados por el usuario

Autónomo e integrador

Stefan Karpinski - Solving the "Two Language" Problem in Data Science

Compromiso de dos niveles

Conveniencia: Usar un lenguaje de scripting (Python, Matlab, R) para agilizar el desarrollo

Ejecución: Acelerar la ejecución del código con algún lenguaje de bajo nivel (C, C++, Fortran)

Inconvenientes...

Acelerar la ejecución de la **carga computacional** (hard stuff) no es problema de los lenguajes de scripting

Se obliga a la **vectorización** para optimizar la ejecución

Se crea una **barrera social** entre desarrolladores y usuarios

---

Stefan Karpinski - Solving the "Two Language" Problem in Data Science