WebR

R compiled for WebAssembly and running in the browser

George Stagg

RStudio

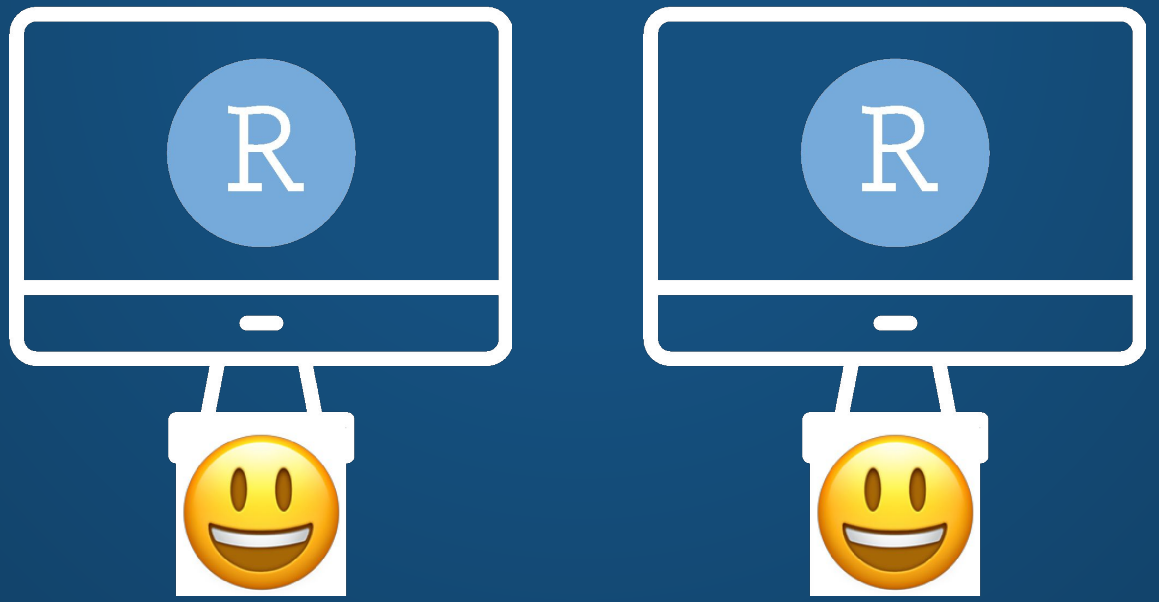
Why run software in a browser?

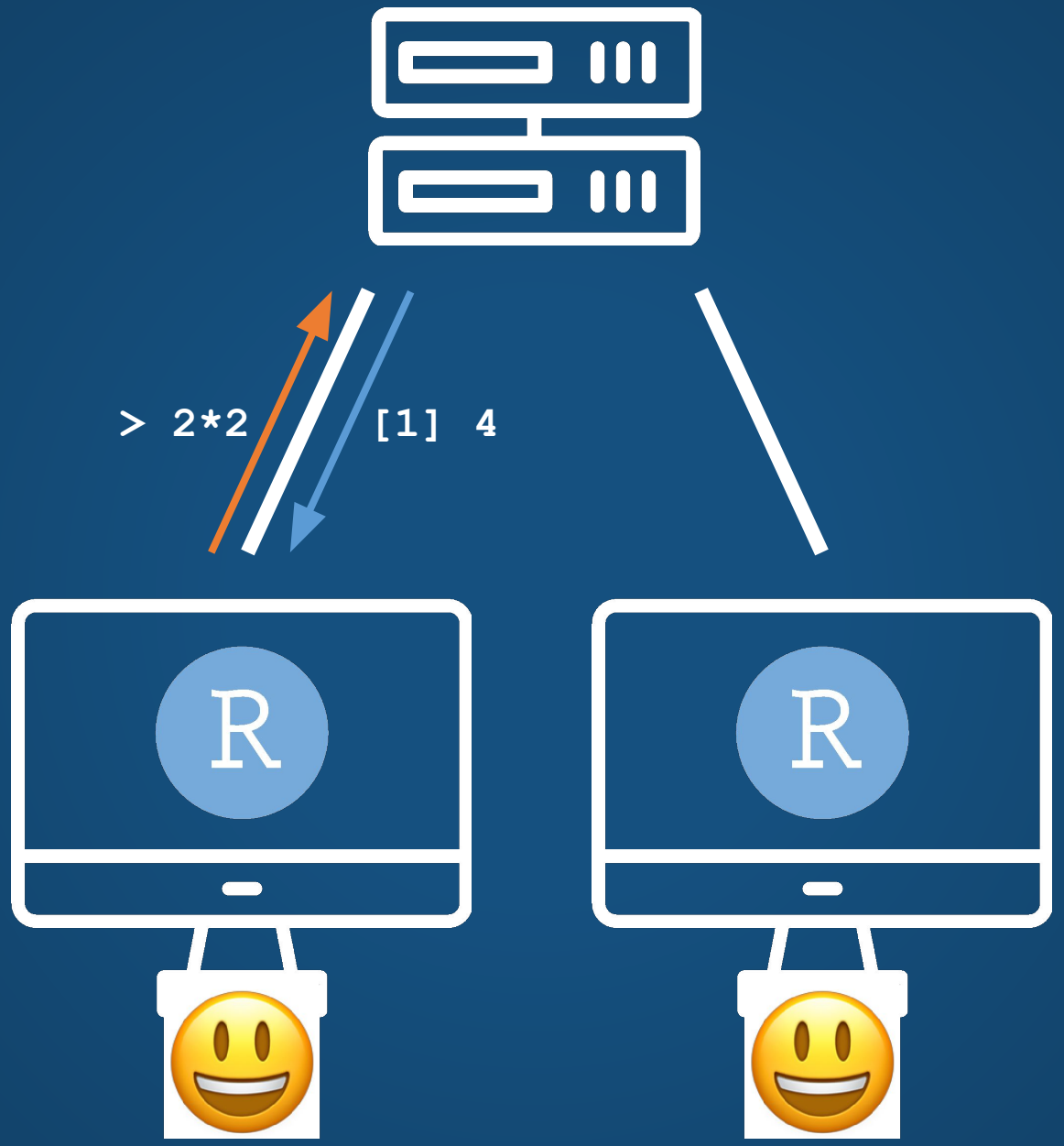
Lots of potential reasons!

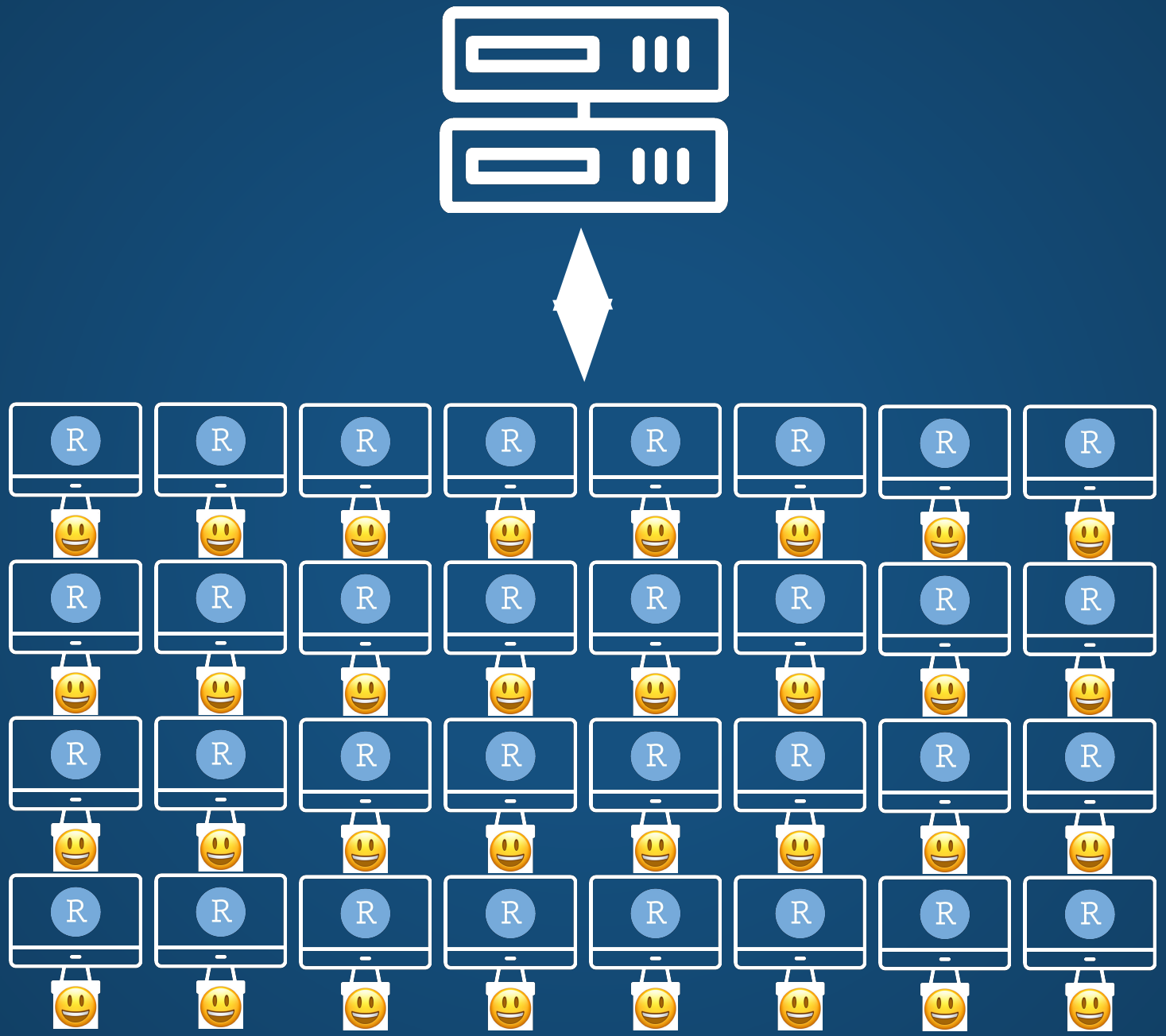
- Containerisation
- Notebook / Jupyter
- Universal binaries
- Sandboxing
- Reproducible output
- Mobile / Tablet development

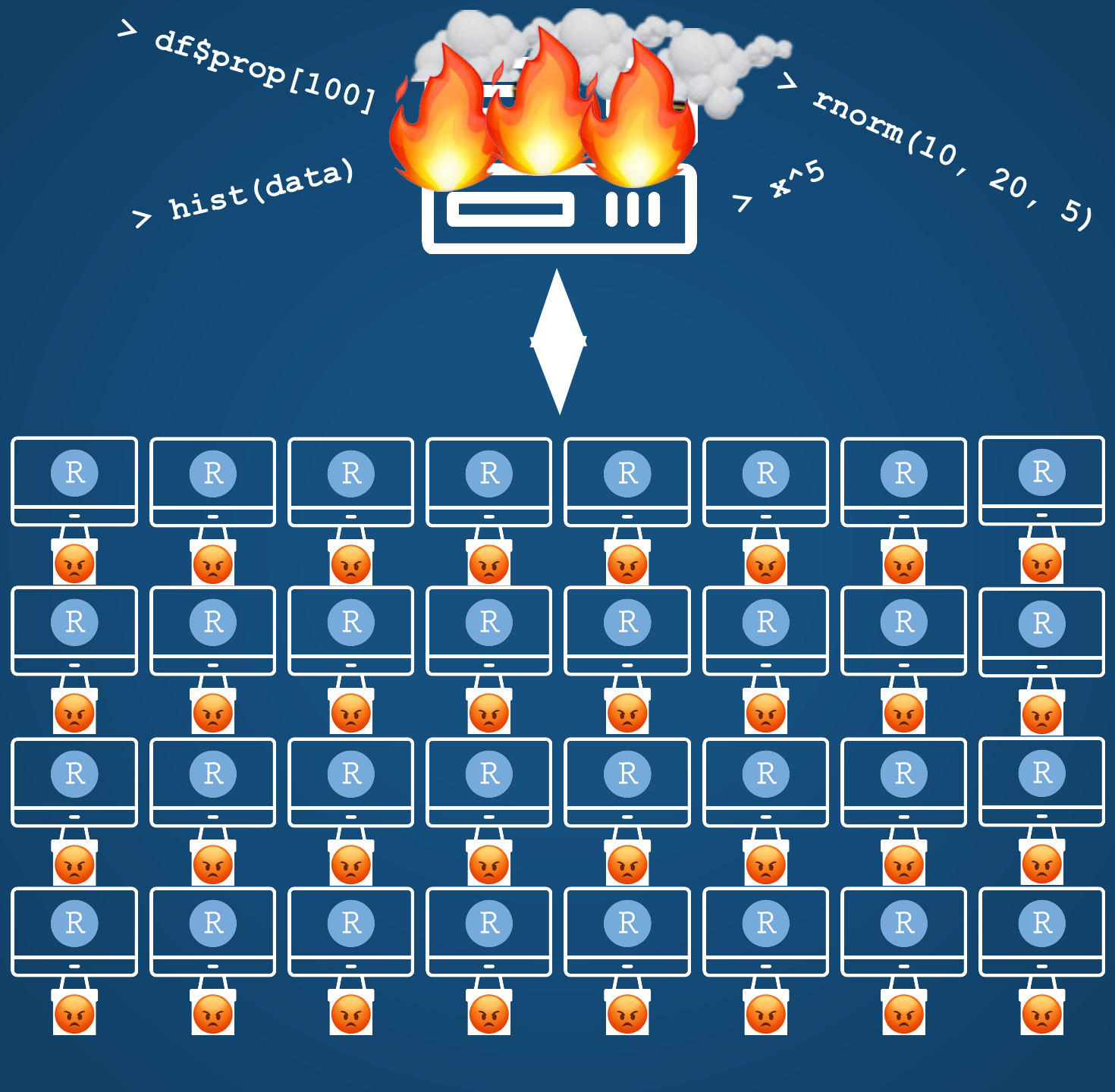
Let me tell you my own reasons for creating webR…

How can we capture R code and evaluate it on devices outside of our control?




> 2*2
[1] 4
R
R


R


> df$prop[100]
> hist(data)
> rnorm(10, 20, 5)
> x^5
R

Solution: Evaluate the R code locally inside a web browser!

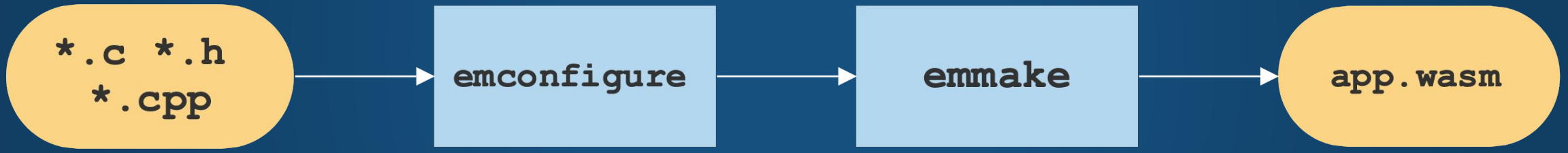
WebAssembly (2017)

- Portable binary code format
- Enables high-performance applications on web pages
- Near-native execution speed
- Supported by most modern browsers

Emscripten (2012)

- C/C++ compiler for WebAssembly
- Based on LLVM/Clang
- Originally designed for ASM.js in web browsers
- DOSBox, SQLite, CPython

How building with Emscripten should work



However… R uses FORTRAN code

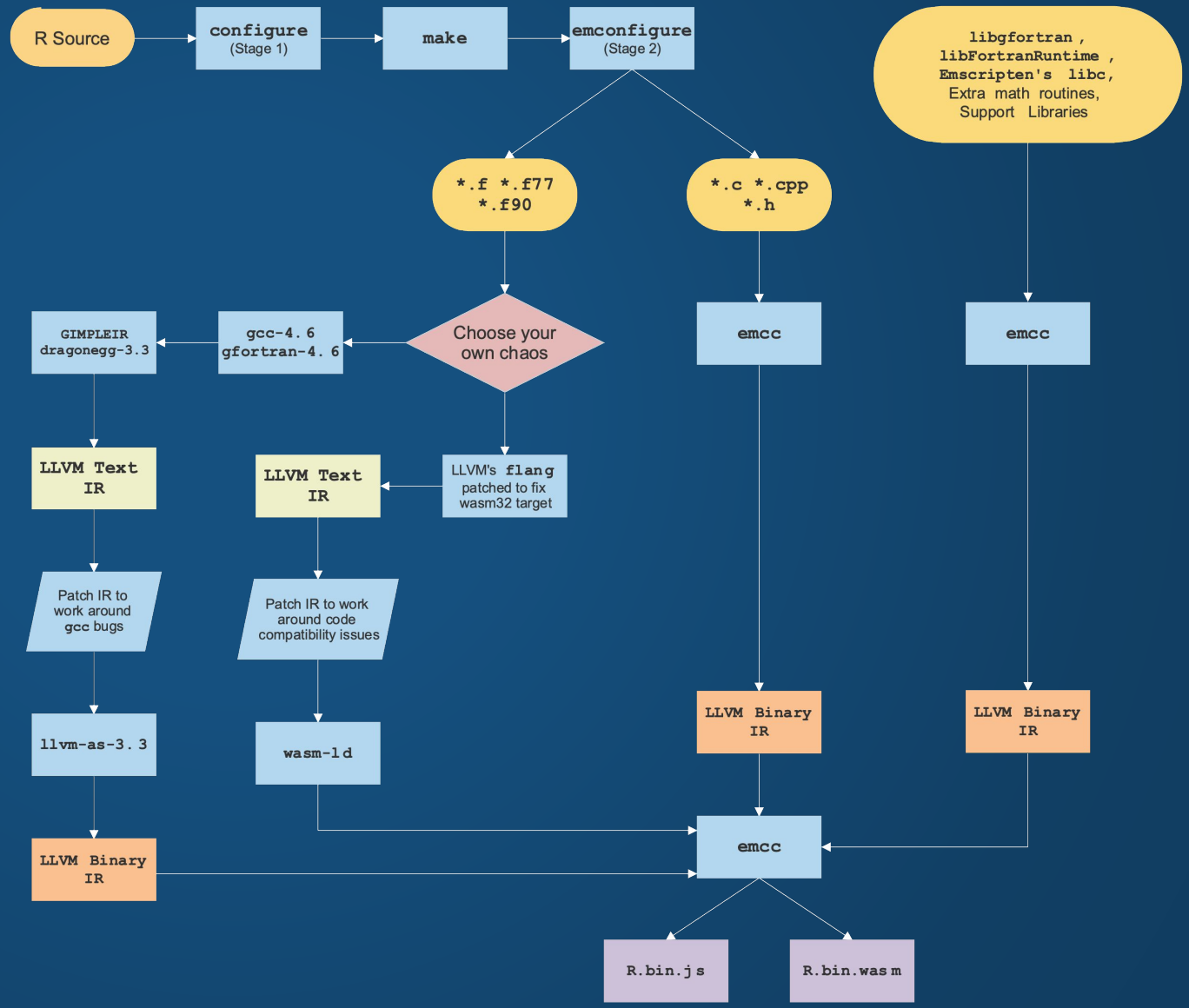
- BLAS (1979)
- LAPACK (1992)
- Various other subroutines…

Not directly supported by Emscripten

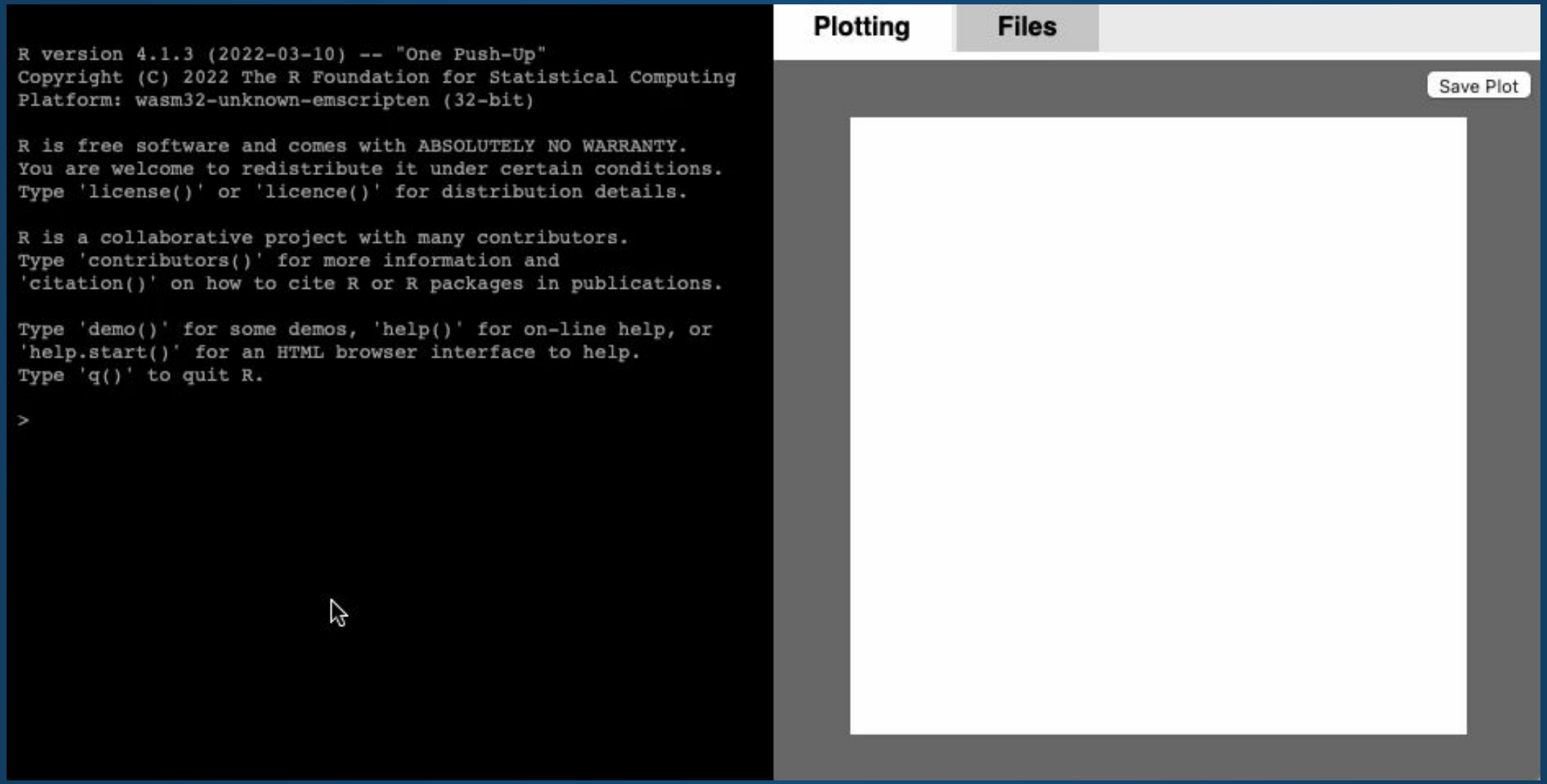
How building R with Emscripten *actually* works:

I won't go into detail, it's a little messy…

Learn more @ https://chrz.de/2020/04/21/fortran-in-the-browser/



…but it works!



Try it for yourself @ https://webr.gwstagg.co.uk

WebR

R compiled for WebAssembly and running in the browser

Try it for yourself @

https://webr.gwstagg.co.uk

Source code @

https://github.com/georgestagg/webR

George Stagg

RStudio