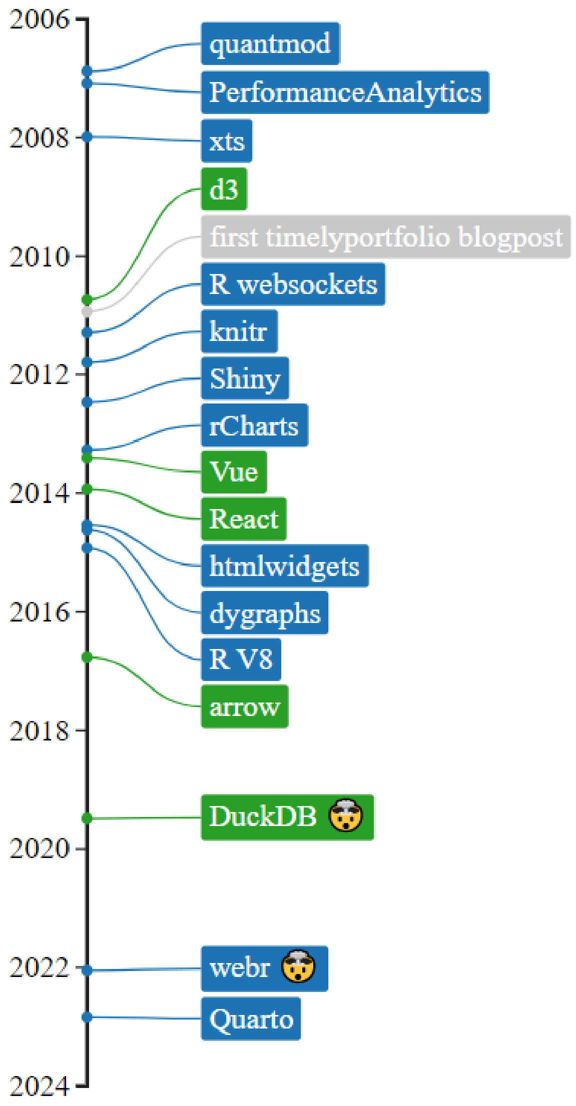
R ∪ Finance ∪ Web

Kent Russell @timelyportfolio

Short History of R ∪ Finance ∪ Web



- Important events defining my personal journey combining the best of R and JavaScript.
- Thanks to everyone here who has contributed to the ecosystem and my journey!

Features of R and JavaScript

R

- amazing, robust built-in statistical functionality
- brilliant ecosystem covering nearly all topic areas
- fantastic community
- data.frames, `xts`
- extensible with other programming languages

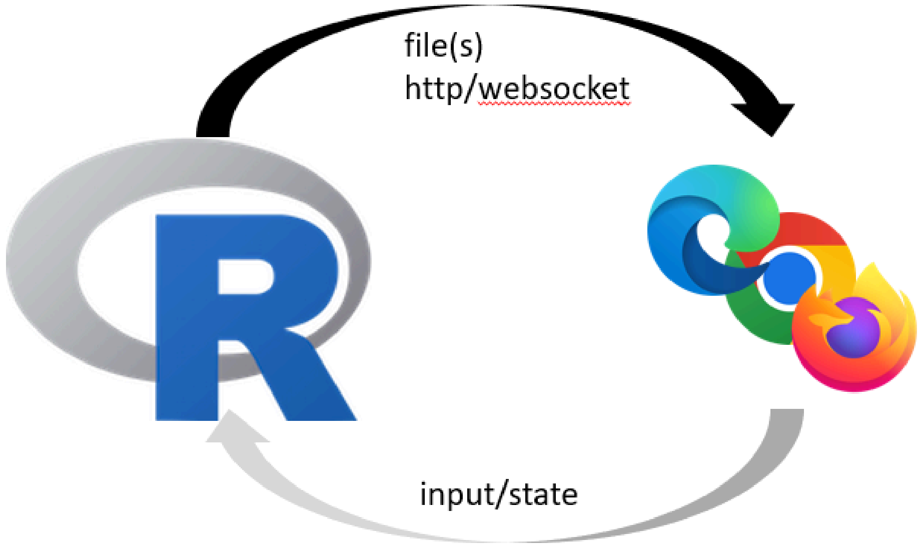
JavaScript

- omnipresent
- interactivity and input
- low/no barrier to entry
- active community
- server or client side

Initial Approach to Combination

R does work and then sends to web for distribution, communication, interactivity, and input as file or through `http/websocket`.

JavaScript is more like an accessory.

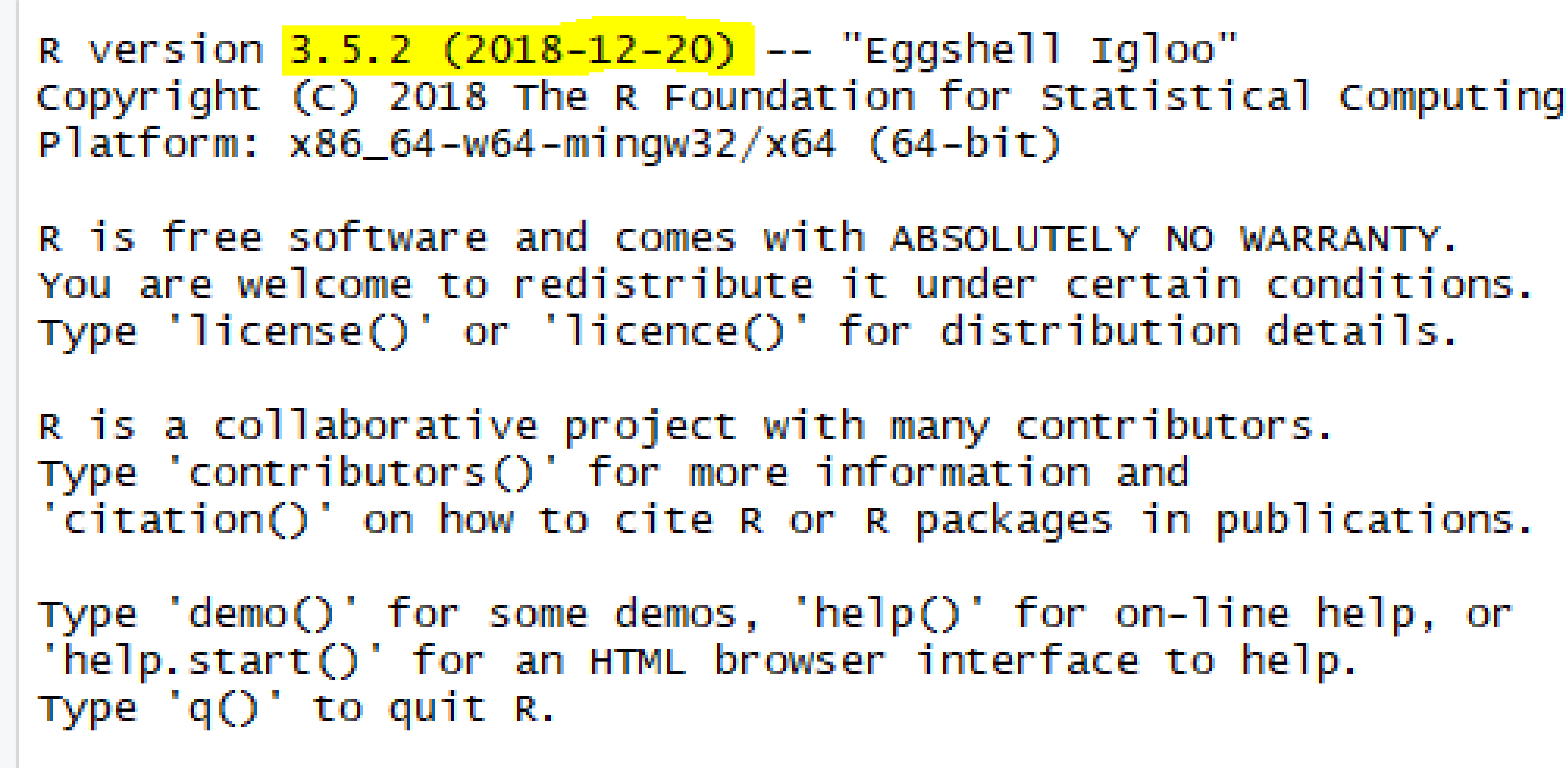


V8 - JS Running inside R

Title:	Embedded JavaScript and WebAssembly Engine for R
Description:	An R interface to V8 <https://v8.dev>: Google's open source JavaScript and WebAssembly engine. This package can be compiled either with V8 version 6 and up or NodeJS when built as a shared library.
Authors:	Jeroen Ooms [aut, cre], Jan Marvin Garbuszus [ctb]
Maintainer:	Jeroen Ooms <jeroen@berkeley.edu>
License:	MIT + file LICENSE
Version:	4.4.2
Built:	2024-04-29 16:38:48 UTC
Source:	https://github.com/jeroen/v8

Free and Open ≠ Available

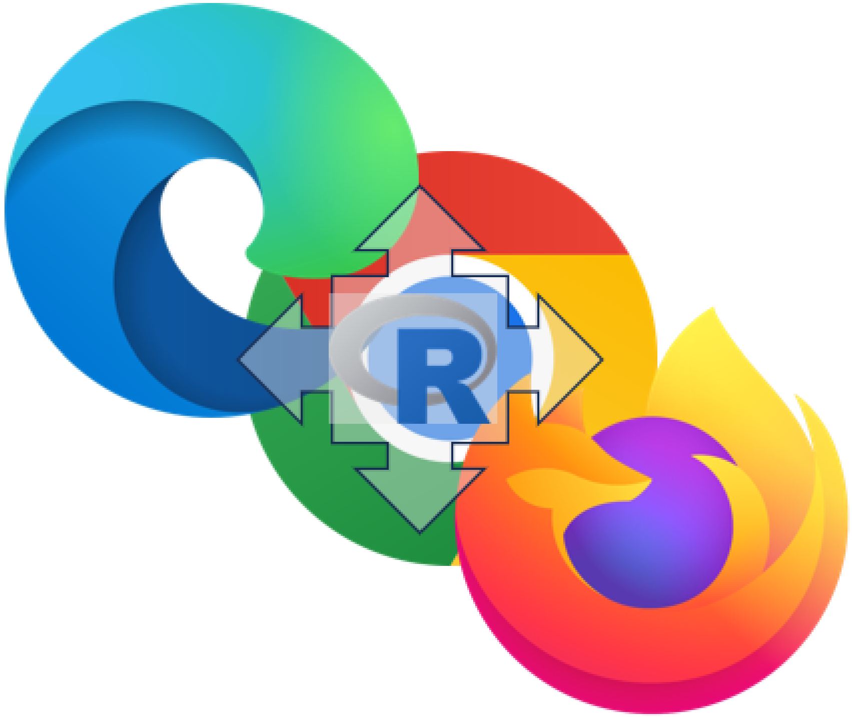
Version Info from Real Job Computer



```
R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

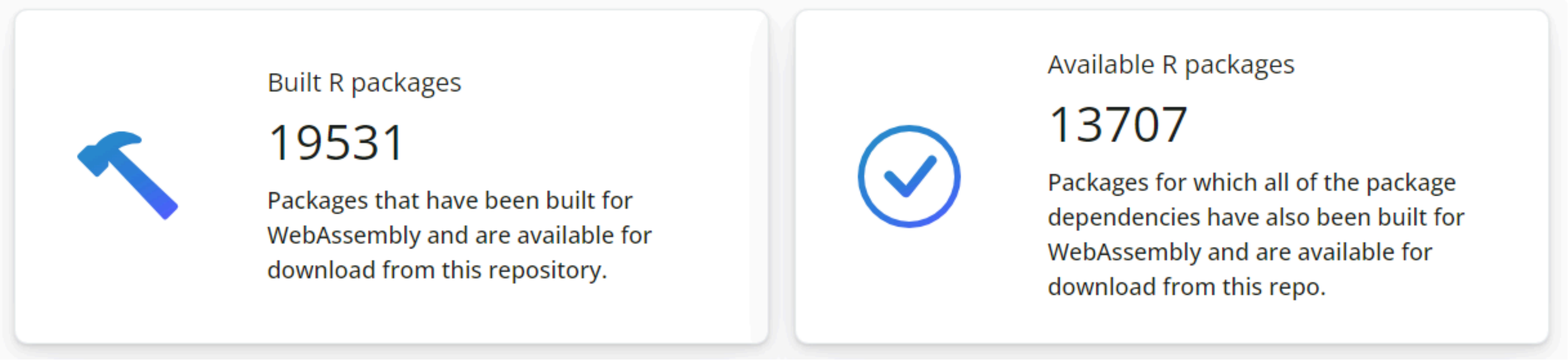
WebR for Omnipresence

R running inside of the browser or Node.js through WebAssembly (WASM)



Widespread Package Availability

- WebR Binary Package repo



Built R packages

19531

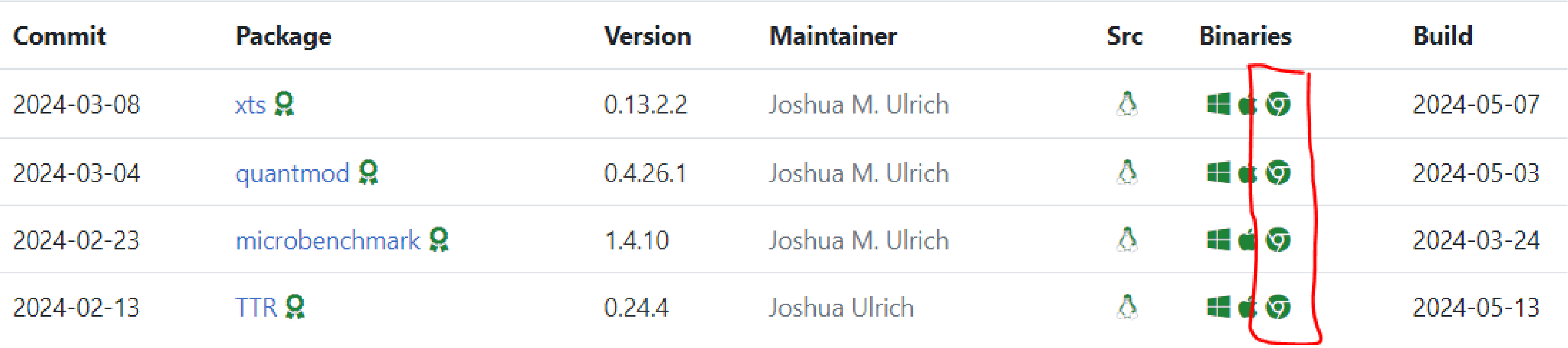
Packages that have been built for WebAssembly and are available for download from this repository.

Available R packages

13707

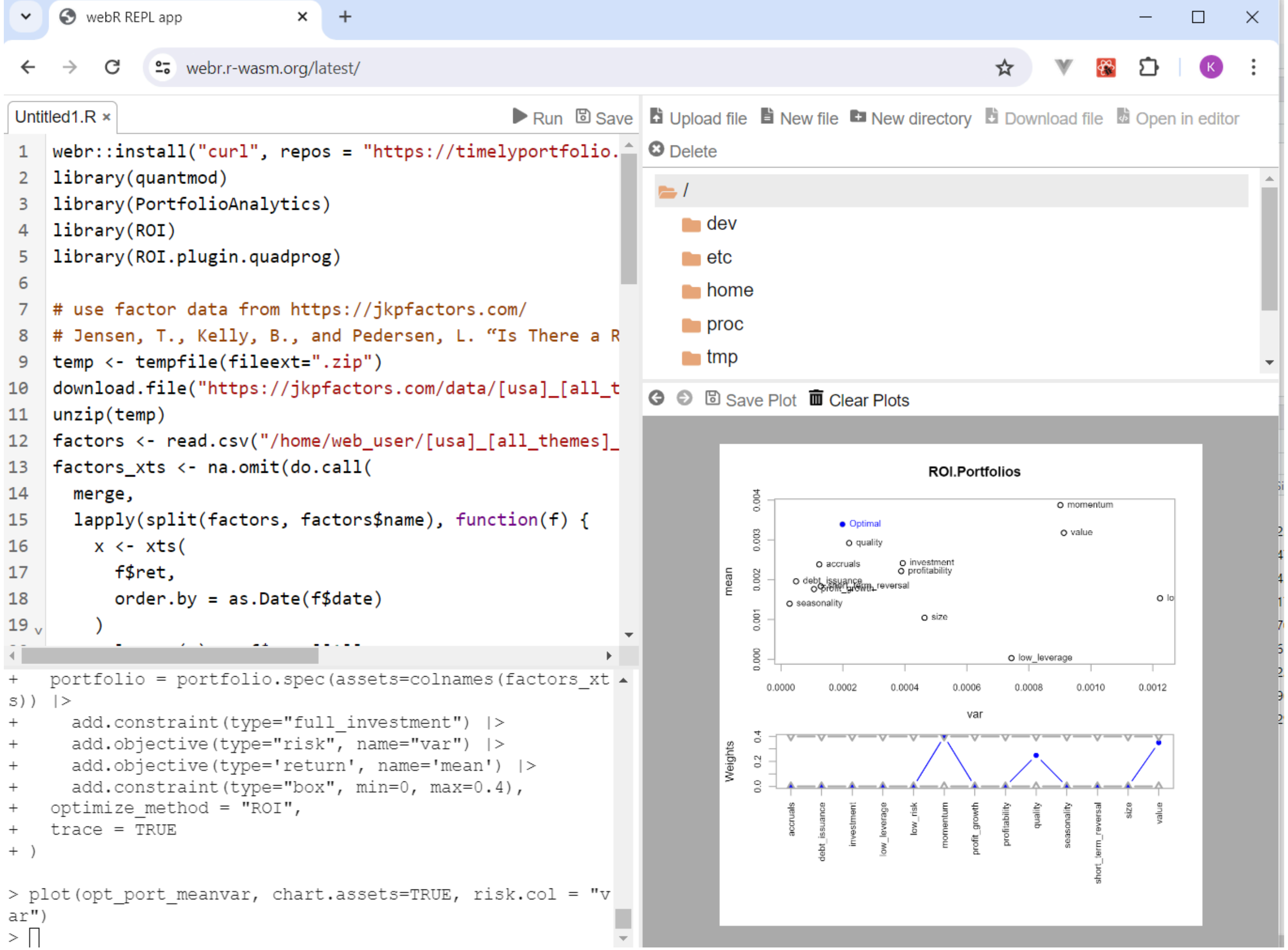
Packages for which all of the package dependencies have also been built for WebAssembly and are available for download from this repo.

- r-universe / how to use



Commit	Package	Version	Maintainer	Src	Binaries	Build
2024-03-08	xts	0.13.2.2	Joshua M. Ulrich			2024-05-07
2024-03-04	quantmod	0.4.26.1	Joshua M. Ulrich			2024-05-03
2024-02-23	microbenchmark	1.4.10	Joshua M. Ulrich			2024-03-24
2024-02-13	TTR	0.24.4	Joshua Ulrich			2024-05-13

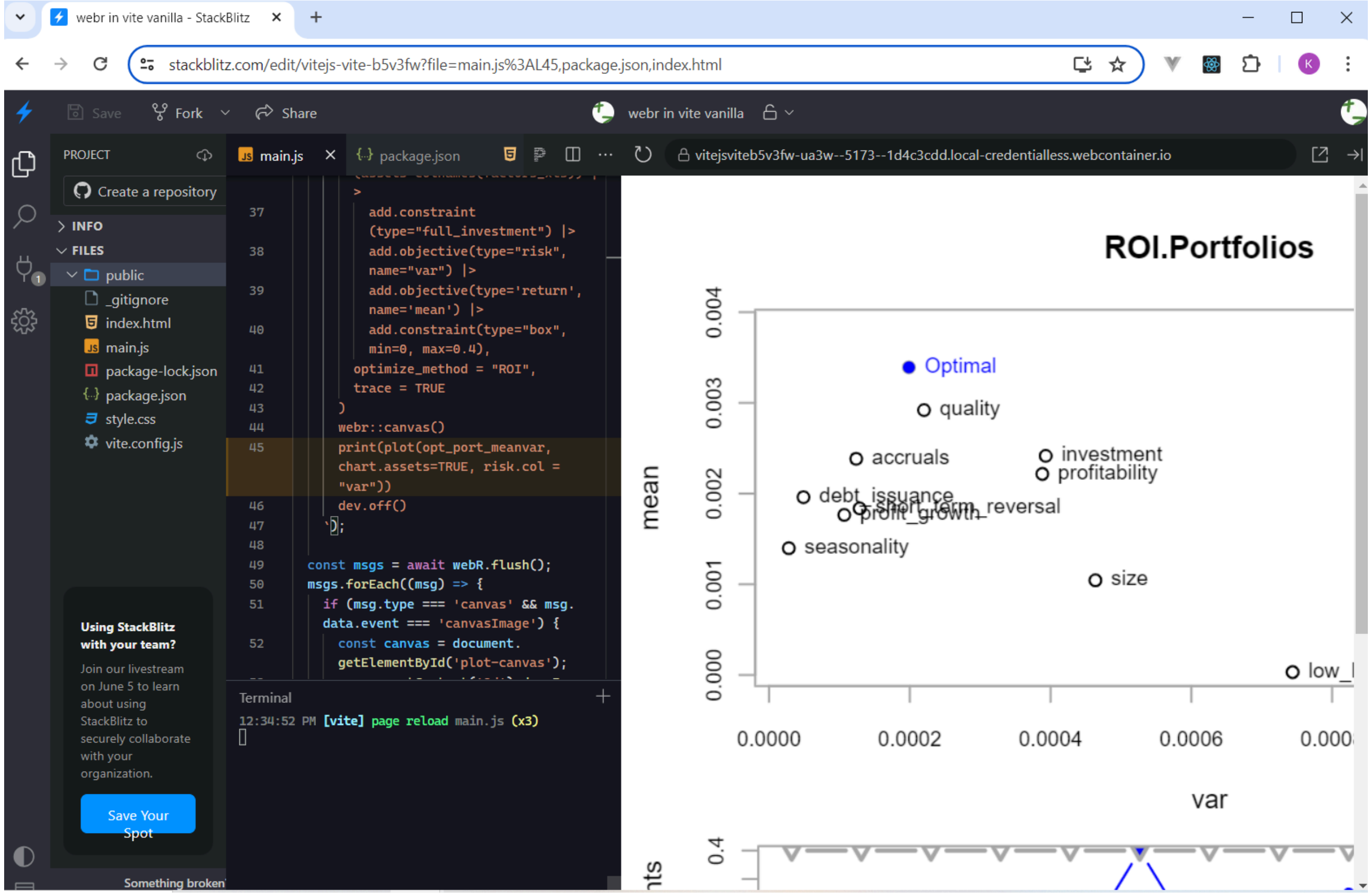
Portfolio Workflow in WebR REPL



```
webr::install("curl", repos = "https://timelyportfolio.
library(quantmod)
library(PortfolioAnalytics)
library(ROI)
library(ROI.plugin.quadprog)

# use factor data from https://jkpfactors.com/
# Jensen, T., Kelly, B., and Pedersen, L. “Is There a R
temp <- tempfile(fileext=".zip")
download.file("https://jkpfactors.com/data/[usa]_[all_t
unzip(temp)
factors <- read.csv("/home/web_user/[usa]_[all_themes]_
factors_xts <- na.omit(do.call(
  merge,
  lapply(split(factors, factors$name), function(f) {
    x <- xts(
      f$ret,
      order.by = as.Date(f$date)
    )
```

```
+   portfolio = portfolio.spec(assets=colnames(factors_xt
s)) |>
+     add.constraint(type="full_investment") |>
+     add.objective(type="risk", name="var") |>
+     add.objective(type='return', name='mean') |>
+     add.constraint(type="box", min=0, max=0.4),
+   optimize_method = "ROI",
+   trace = TRUE
+ )

> plot(opt_port_meanvar, chart.assets=TRUE, risk.col = "v
ar")
> 
```

Portfolio Workflow in Node.js



```
          >
            add.constraint
            (type="full_investment") |>
            add.objective(type="risk",
            name="var") |>
            add.objective(type='return',
            name='mean') |>
            add.constraint(type="box",
            min=0, max=0.4),
          optimize_method = "ROI",
          trace = TRUE
        )
        webr::canvas()
        print(plot(opt_port_meanvar,
        chart.assets=TRUE, risk.col =
        "var"))
        dev.off()
      `);

    const msgs = await webR.flush();
    msgs.forEach((msg) => {
      if (msg.type === 'canvas' && msg.
      data.event === 'canvasImage') {
        const canvas = document.
        getElementById('plot-canvas');
```

Terminal

```
12:34:52 PM [vite] page reload main.js (x3)
```

Using StackBlitz with your team?

Join our livestream on June 5 to learn about using StackBlitz to securely collaborate with your organization.

Save Your Spot

Stackblitz example

Portfolio Workflow Code | Step 1

Load Libraries

```r
# stubbed non-functioning version of curl so quantmod will load
webr::install(
  "curl",
  repos = "https://timelyportfolio.github.io/webr_repo/"
)
webr::install(c(
  "quantmod", "PortfolioAnalytics",
  "ROI", "ROI.plugin.quadprog"
))

library(quantmod)
library(PortfolioAnalytics)
library(ROI)
library(ROI.plugin.quadprog)
```

Portfolio Workflow Code | Step 2

Get JKP Factor Data and Convert to xts

```r
# use factor data from https://jkpfactors.com/
# Jensen, T., Kelly, B., and Pedersen, L.
# “Is There a Replication Crisis in Finance?” Journal of Finance (2023)
temp <- tempfile(fileext=".zip")
download.file(
  "https://jkpfactors.com/data/[usa]_[all_themes]_[monthly]_[vw_cap].zip",
  temp
)
unzip(temp)
factors <- read.csv("/home/web_user/[usa]_[all_themes]_[monthly]_[vw_cap].csv")
factors_xts <- na.omit(do.call(
  merge,
  lapply(split(factors, factors$name), function(f) {
    x <- xts(f$ret, order.by = as.Date(f$date))
    colnames(x) <- f$name[[1]]
    x
  })
))
```

Portfolio Workflow Code | Step 3

Optimize Mean-Variance Portfolio

```r
# mean variance portfolio
opt_port_meanvar <- optimize.portfolio(
  R = factors_xts,
  portfolio = portfolio.spec(assets=colnames(factors_xts)) |>
    add.constraint(type="full_investment") |>
    add.objective(type="risk", name="var") |>
    add.objective(type='return', name='mean') |>
    add.constraint(type="box", min=0, max=0.4),
  optimize_method = "ROI",
  trace = TRUE
)
plot(opt_port_meanvar, chart.assets=TRUE, risk.col = "var")
```

Things To Consider

- Slower browser load time (still < 30 seconds) and more bandwidth consumption
- Be careful refresh in the browser means starting over and everything disappears (but this can also be a good thing)
- Some packages are not yet available. `Rglpk` is not working (but I think can be fixed). Many depend on `curl` but often do not require it for most of the functionality (sort of solved with a stubbed package).
- CORS can make loading data in the browser from remote sources difficult or impossible.

WebR Resources

WebR Team George Stagg and Lionel Henry

- documentation
- github
- talk at posit::conf(2023)

James Balamuta @coatless

- quarto-webr | amazing tool and fantastic, thorough guide to all things WebR
- talk at Stanford

Bob Rudis @hrbrmstr

- talk at NYR
- webr book

Colin Fay @colinfay

- blog