Data Science with R and pbdR at ORNL: From the CADES Cloud to the OLCF

Part 1: R and the Cloud

Drew Schmidt and George Ostrouchov
6/18/2018

Outline

(10:00-12:00) Part 1: R and the Cloud

- Basic R information
- Profiling
- Running R services in openstack

(12:00-1:00) Break for Lunch/Q&A

(1:00-3:00) Part 2: pbdR and the OLCF

- Distributed computing with pbdR
- Several applications on OLCF resources

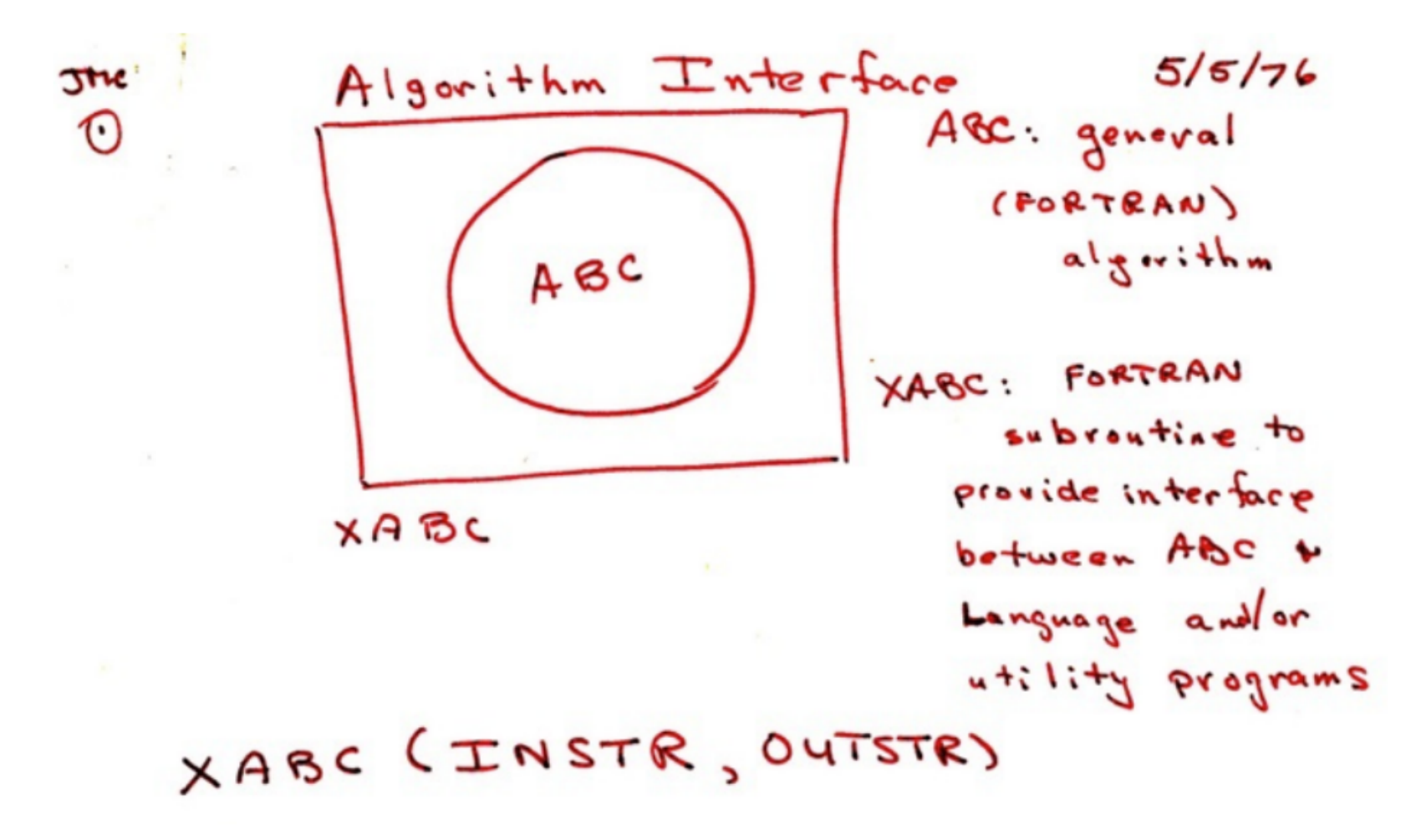
Some R Basics

What is R?

> R is part programming language and part data analysis package.
>
> --me

> R is a shockingly dreadful language for an exceptionally useful data analysis environment.
>
> from aRrgh: a newcomer's (angry) guide to R

What is R?



From http://datascience.la/john-chambers-user-2014-keynote/

What is R?

Variable naming like C

```
x <- 1
_nope <- 2
3_alsono <- 3
```

```
## Error: <text>:2:1: unexpected input
## 1: x <- 1
## 2: _
##    ^
```

What is R?

Variable naming like C

```
x <- 1
_nope <- 2
3_alsono <- 3
```

```
## Error: <text>:2:1: unexpected input
## 1: x <- 1
## 2: _
##    ^
```

lol jk

```
`this variable name has spaces` <- 1
`😺` <- "cat"
ls()
```

```
## [1] "😺"                          "this variable name has spaces"
```

What is R?

A very stupid language

```
T
```

```
## [1] TRUE
```

```
F
```

```
## [1] FALSE
```

```
T <- FALSE
F <- TRUE
T
```

```
## [1] FALSE
```

```
F
```

```
## [1] TRUE
```

IEEE Spectrum

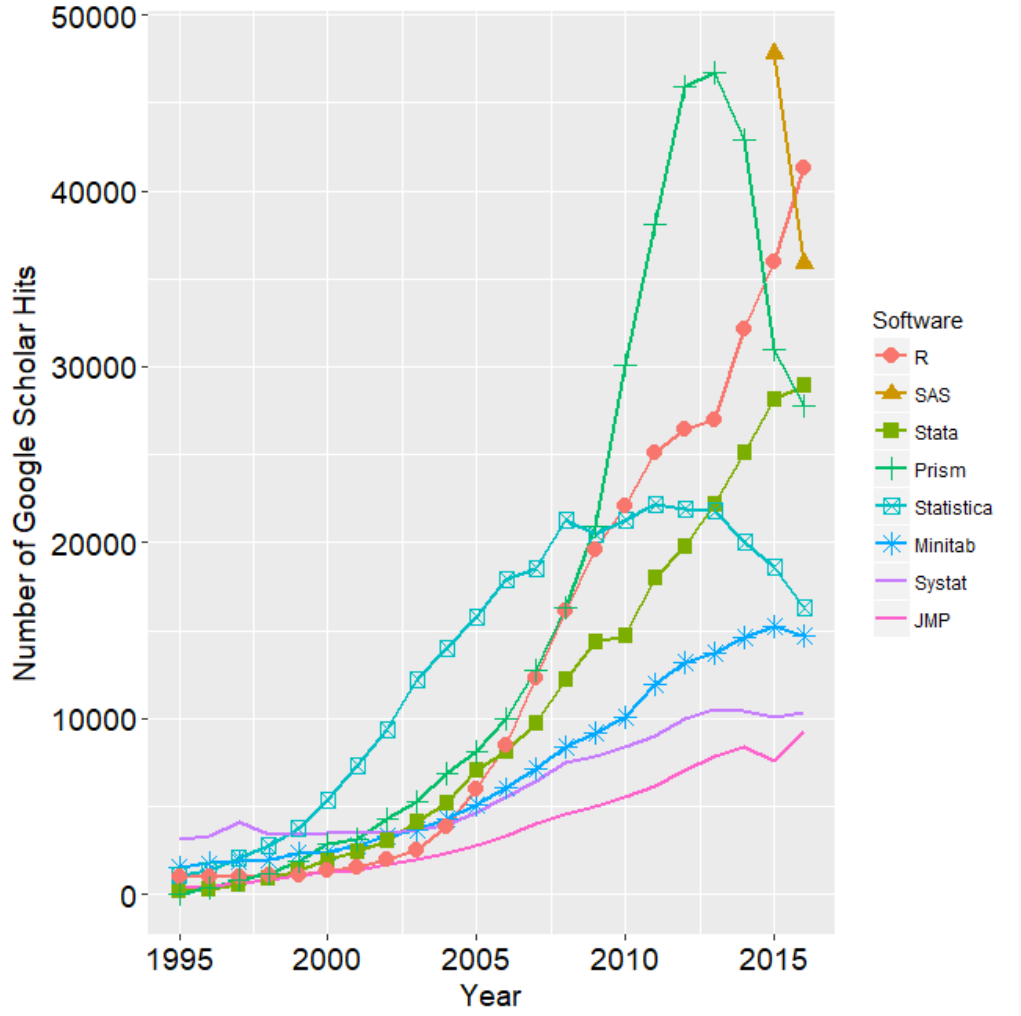
2015

Language Rank	Types	Spectrum Ranking
1. Java		100.0
2. C		99.3
3. C++		95.5
4. Python		93.4
5. C#		92.4
6. PHP		84.7
7. Javascript		84.4
8. Ruby		78.8
9. R		74.2
10. MATLAB		72.9

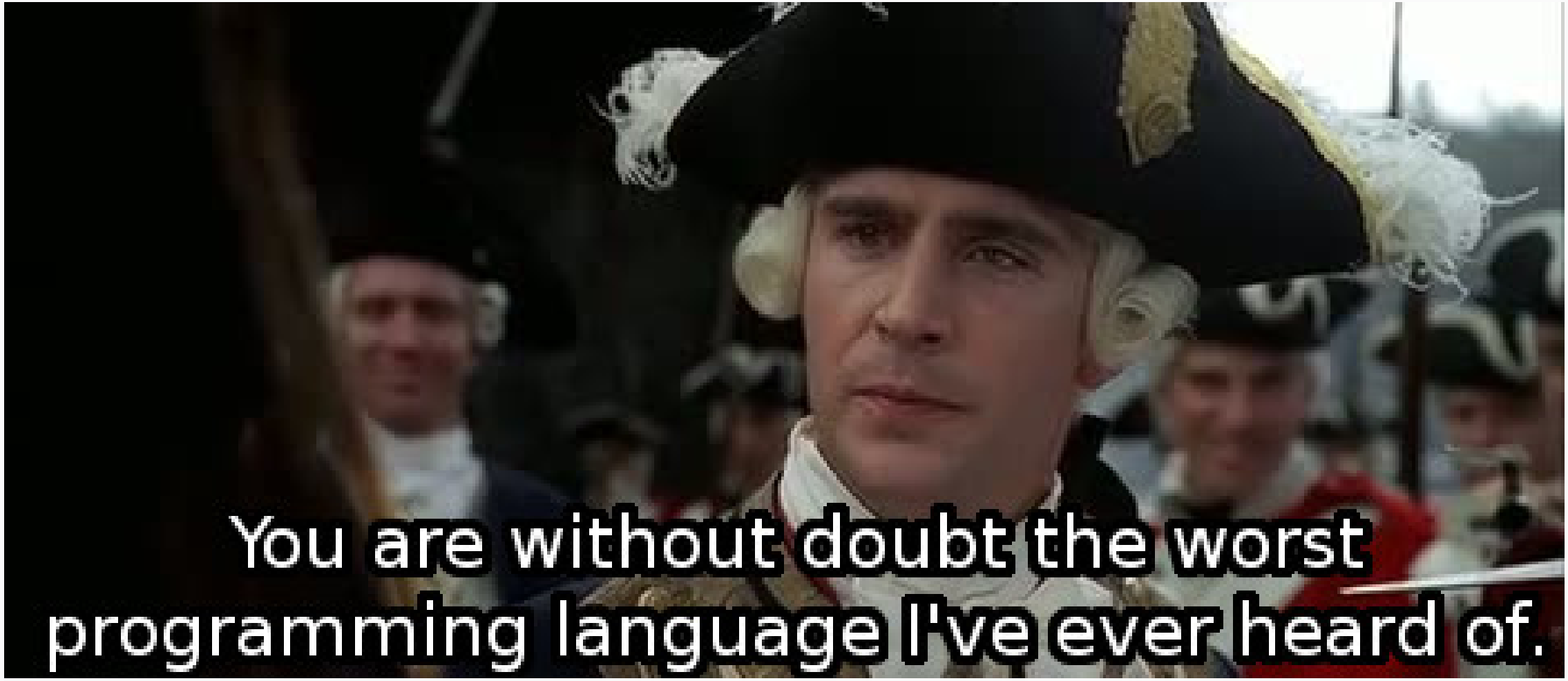
2016

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

Scholarly Impact



From http://r4stats.com/articles/popularity/


You are without doubt the worst
programming language I've ever heard of.

But you HAVE heard of me!

R Resources

- Books
  - Advanced R http://adv-r.had.co.nz/
  - The Art of R Programming http://nostarch.com/artofr.htm
  - An Introduction to R http://cran.r-project.org/doc/manuals/R-intro.pdf
  - The R Inferno http://www.burns-stat.com/pages/Tutor/R_inferno.pdf
- Useful websites
  - Task Views http://cran.at.r-project.org/web/views
  - Mathesaurus: http://mathesaurus.sourceforge.net
  - R language for programmers http://www.johndcook.com/R_language_for_programmers.html
  - aRrgh: a newcomer's (angry) guide to R http://tim-smith.us/arrgh/
- Advanced resources
  - R Installation and Administration http://cran.r-project.org/doc/manuals/R-admin.html
  - Writing R Extensions http://cran.r-project.org/doc/manuals/R-exts.html
  - Mailing list archives: http://tolstoy.newcastle.edu.au/R/
- Getting help
  - The `R` stackoverflow tag
  - The `#rstats` tag on Twitter

Getting Started with R

Interfaces

- Run in the console via `R`
- Windows installs come with RGui, Mac with R.app.
- RStudio
- But more on these later...

Getting Started with R

```
1+1
```

```
## [1] 2
```

```
0:4 + 1
```

```
## [1] 1 2 3 4 5
```

```
runif(5)
```

```
## [1] 0.7253431 0.6689435 0.9777974 0.7955616 0.9900558
```

```
rnorm(5)
```

```
## [1]  1.11592567  0.09732279  0.94398510  1.16211468 -0.22086554
```

Getting Started with R

```
example(lm)
```

```
##
## lm> require(graphics)

## Loading required package: graphics

##
## lm> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
## lm> ## Page 9: Plant Weight Data.
## lm> ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
##
## lm> trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
##
## lm> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
##
## lm> weight <- c(ctl, trt)
##
## lm> lm.D9 <- lm(weight ~ group)
##
## lm> lm.D90 <- lm(weight ~ group - 1) # omitting intercept
```

Getting Started with R

```
nnet::multinom(Species ~ Sepal.Length + Sepal.Width, data=iris, trace=FALSE)
```

```
## Call:
## nnet::multinom(formula = Species ~ Sepal.Length + Sepal.Width,
##     data = iris, trace = FALSE)
##
## Coefficients:
##            (Intercept) Sepal.Length Sepal.Width
## versicolor   -92.09924     40.40326   -40.58755
## virginica   -105.10096     42.30094   -40.18799
##
## Residual Deviance: 110.425
## AIC: 122.425
```

The CRAN

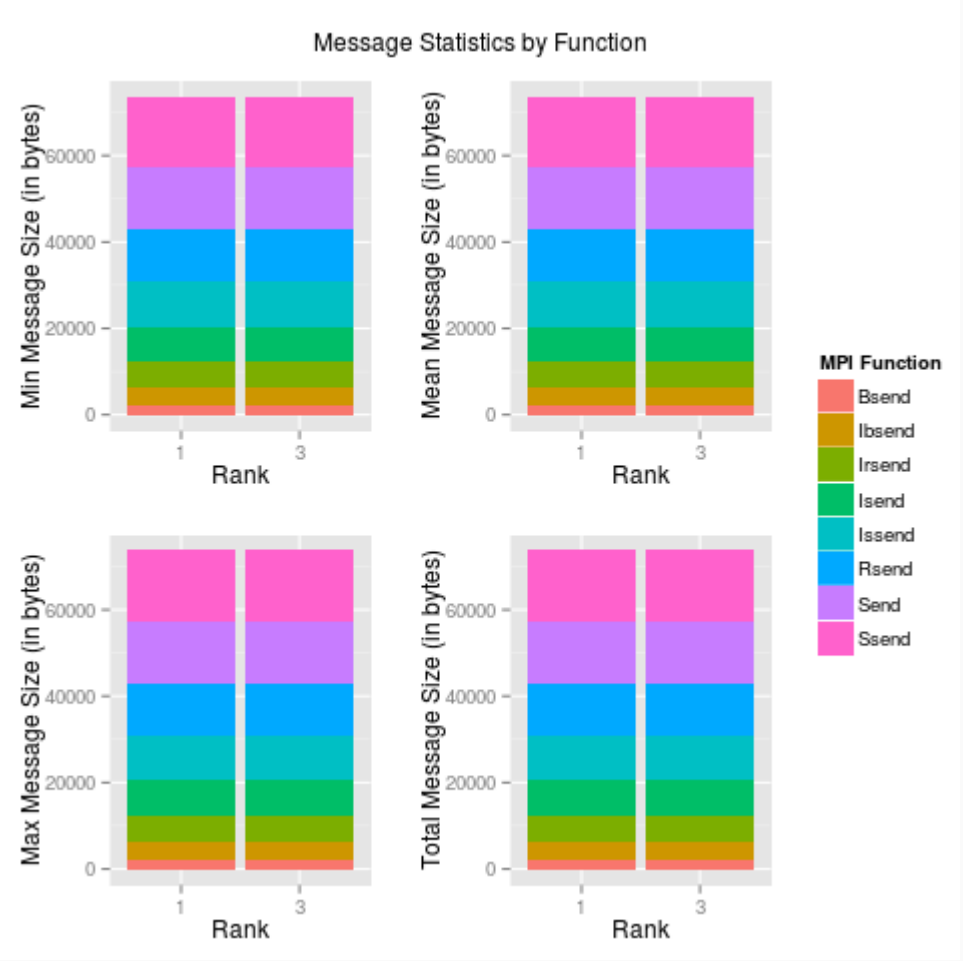
- Comprehensive R Archive Network
- The only good programming language packaging/distribution system.
- Install packages via `install.packages()`:
  - `install.packages("remotes")`
  - `remotes::install_github("wrathematics/openblasctl")`

Performance

"But why should we care about performance???"

Profiling

- Basic profiling
  - `system.time()` : timing blocks of code
  - `Rprof()` : timing all function executions
  - `Rprofmem()` : measuring memory allocations
  - `tracemem()` : tracking data copies
- Other profilers (packages on CRAN/Github)
  - pbdPROF (fpmpi, mpiP)
  - pbdPAPI
  - rbenchmark
  - microbenchmark
  - lineprof



Profiling

```
x <- matrix(rnorm(20000*750), nrow=20000, ncol=750)
str(x)
```

```
##  num [1:20000, 1:750] -1.011 -2.153 -0.263 0.017 1.572 ...
```

```
system.time(t(x) %*% x)
```

```
##    user  system elapsed
##   1.889   0.141   0.609
```

```
system.time(crossprod(x))
```

```
##    user  system elapsed
##   0.994   0.173   0.303
```

```
system.time(cov(x))
```

```
##    user  system elapsed
##   8.945   0.162   8.915
```

Profiling

```
system.time({
  y <- x+1
  z <- y*2
})
```

```
##    user  system elapsed
##   1.234   0.223   1.457
```

Profiling

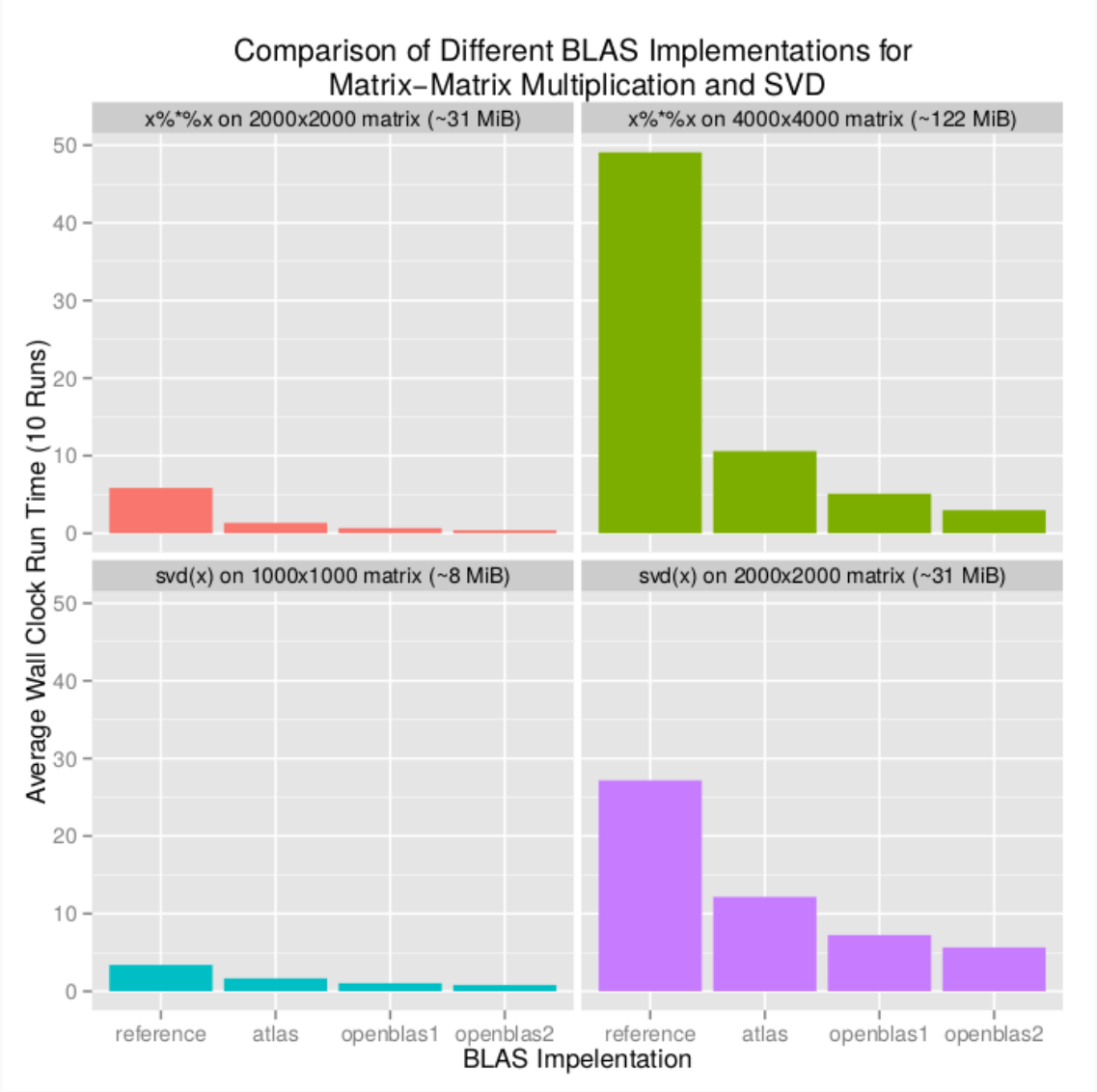
```
Rprof()
invisible(prcomp(x))
Rprof(NULL)

summaryRprof()
```

```
## $by.self
##                 self.time self.pct total.time total.pct
## "La.svd"            13.54    82.56      13.66     83.29
## "%*%"                2.16    13.17       2.16     13.17
## "aperm.default"      0.28     1.71       0.28      1.71
## "is.finite"          0.12     0.73       0.12      0.73
## "array"              0.06     0.37       0.06      0.37
## "matrix"             0.06     0.37       0.06      0.37
## "svd"                0.04     0.24      13.84     84.39
## "any"                0.04     0.24       0.04      0.24
## "t.default"          0.04     0.24       0.04      0.24
## "sweep"              0.02     0.12       0.36      2.20
## "colMeans"           0.02     0.12       0.02      0.12
## "unique"             0.02     0.12       0.02      0.12
##
```

Improving Performance

- All the usual HLL stuff
  - Vectorize
  - Write C/C++/Fortran kernels
- All the usual HPC stuff
  - Build with a better compiler
  - Use optimized BLAS/LAPACK
  - Go parallel
- Use the bytecode compiler

High Performance BLAS



Basic Parallelism

```
unlist(lapply(1:5, sqrt))
```

```
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068
```

```
n <- 1:1e6
system.time(lapply(n, sqrt))
```

```
##    user  system elapsed
##   1.018   0.003   1.021
```

```
system.time(parallel::mclapply(n, sqrt))
```

```
##    user  system elapsed
##   0.127   0.061   0.501
```

Running R Services in Openstack

Openstack

Ways to Interface with Your VM

- ssh
- remoter
- RStudio server
- Dashboards/webapps (shiny)

ssh

Pros

- Ubiquitous
- Good for running things in batch

Cons

- CLI only
- Have to be comfortable with *nix

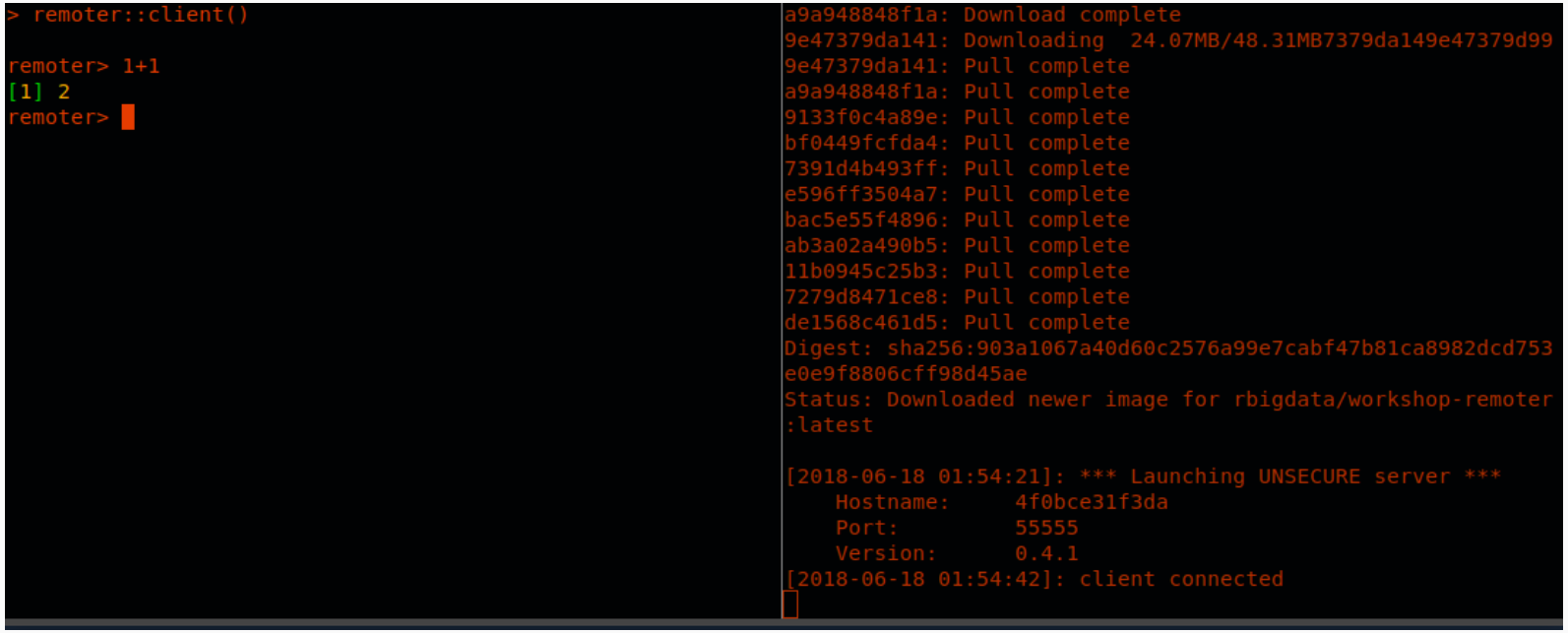
remoter

Pros

- Can use from any local R interface (terminal, R.app, RStudio, ...)
- Can avoid need to use ssh

Cons

- Setting up the server is somewhat DIY
- Lots of ssh tunneling depending on firewall



RStudio

Pros

- Ubiquitous among R users
- Well-supported

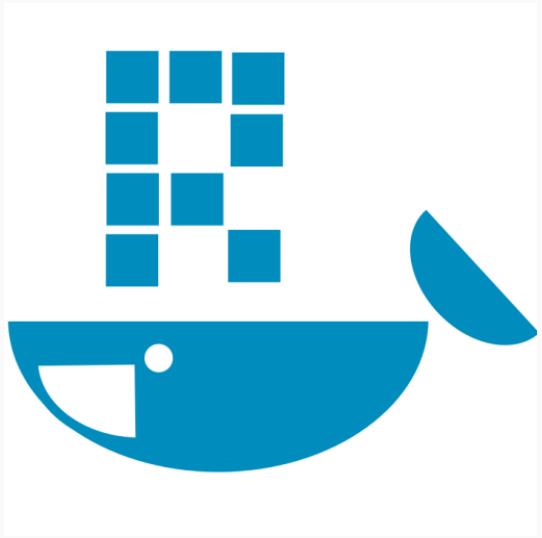
Cons

- Ubiquitous among R users
- Sever variant has same ssh tunnel issue

Docker

For ease of distributing things, we'll be using Docker.

- Container platform for Linux
- ***NOT A VM***
- ...except on Windows and Mac
- rocker project maintains helpful R distributions:
  - rocker/r-base
  - rocker/rstudio
  - rocker/shiny
  - rocker/tidyverse



Installing Docker on Your Laptop

Windows

- Windows 10 or later
- Install the Docker Community Edition for Windows

Mac

- OS X El Capitan 10.11 or later
- Install the Docker Community Edition for Mac.

Linux

- deb (Debian, Ubuntu): `apt-get install docker.io`
- rpm (Fedora, Centos): `yum install docker-io`

Openstack and Docker Resources

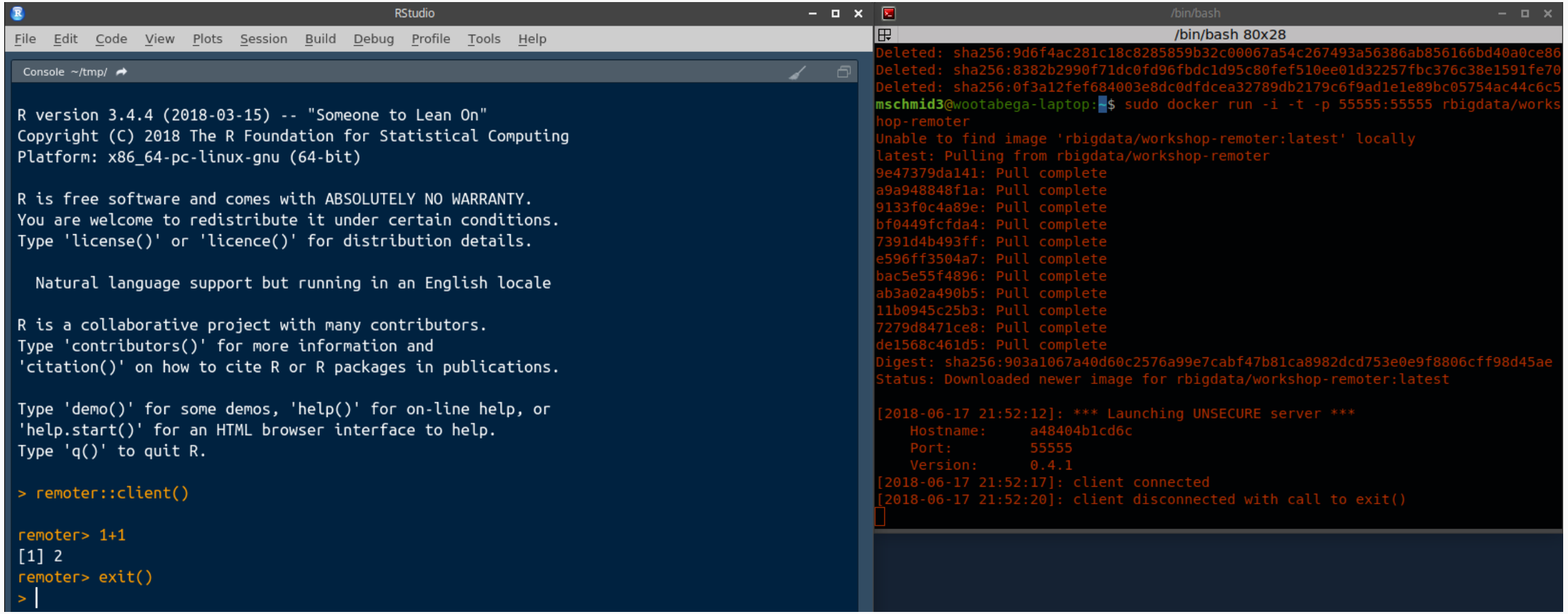
- Request birthright cloud access
- Birthright cloud login (domain: ornl)
- R Docker tutorial

Tunneling

- If running a docker service in openstack, you need to tunnel.
- If you are on the ORNL network, you need 1 tunnel:
  - If your VM's IP is `1.2.3.4`:
  - `ssh -L 8787:localhost:8787 -N cades@1.2.3.4`
- If you're off the ORNL network, you need 2 tunnels:
  - If your XCAMS/UCAMS ID is `abc`:
  - `ssh -L 8787:localhost:8787 abc@cades-extlogin01.ornl.gov ssh -L 8787:localhost:8787 -N cades@1.2.3.4`

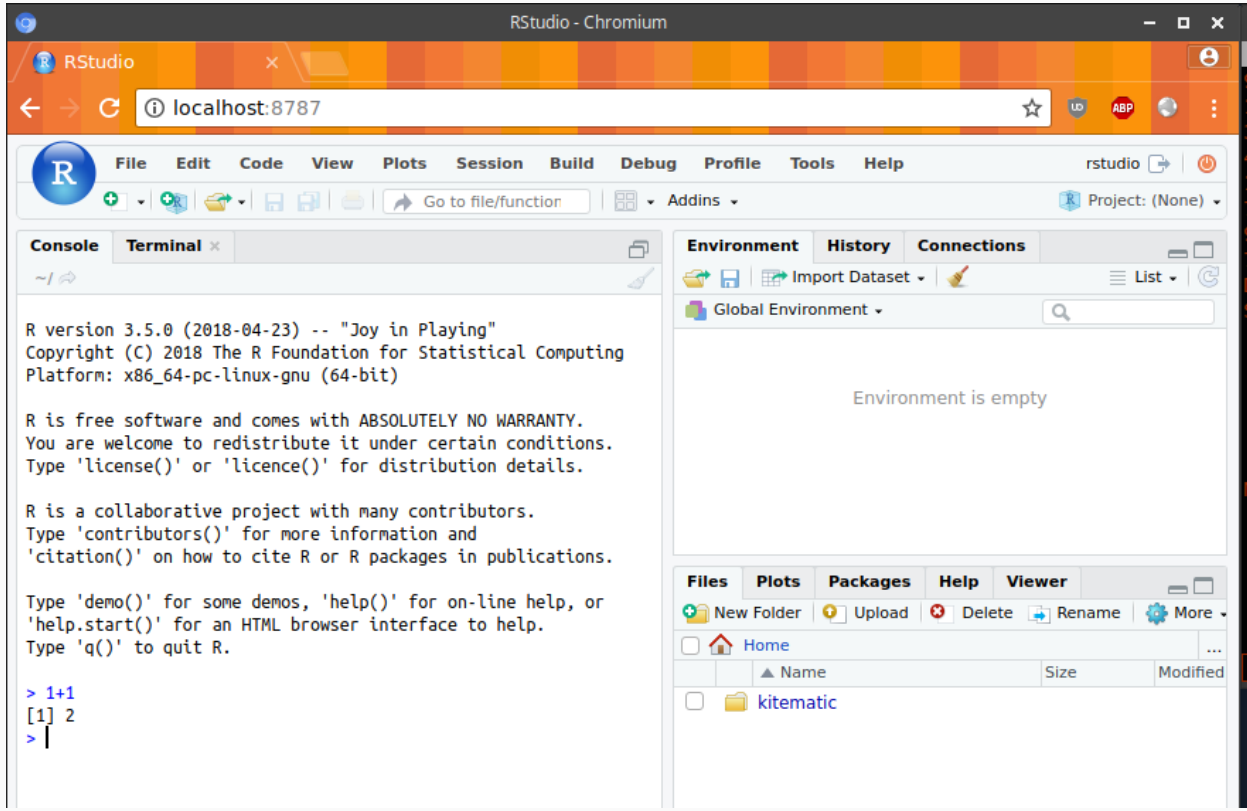
remoter

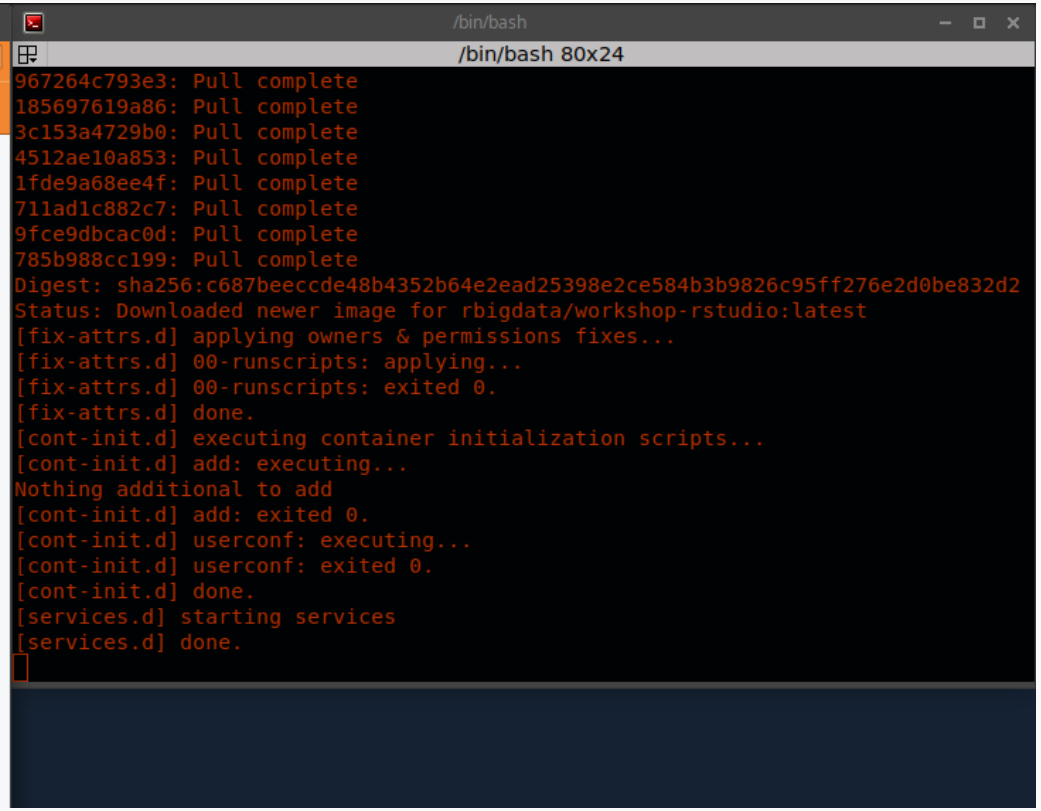
```
sudo docker run -i -t -p 55555:55555 rbigdata/workshop-remoter
```



rstudio

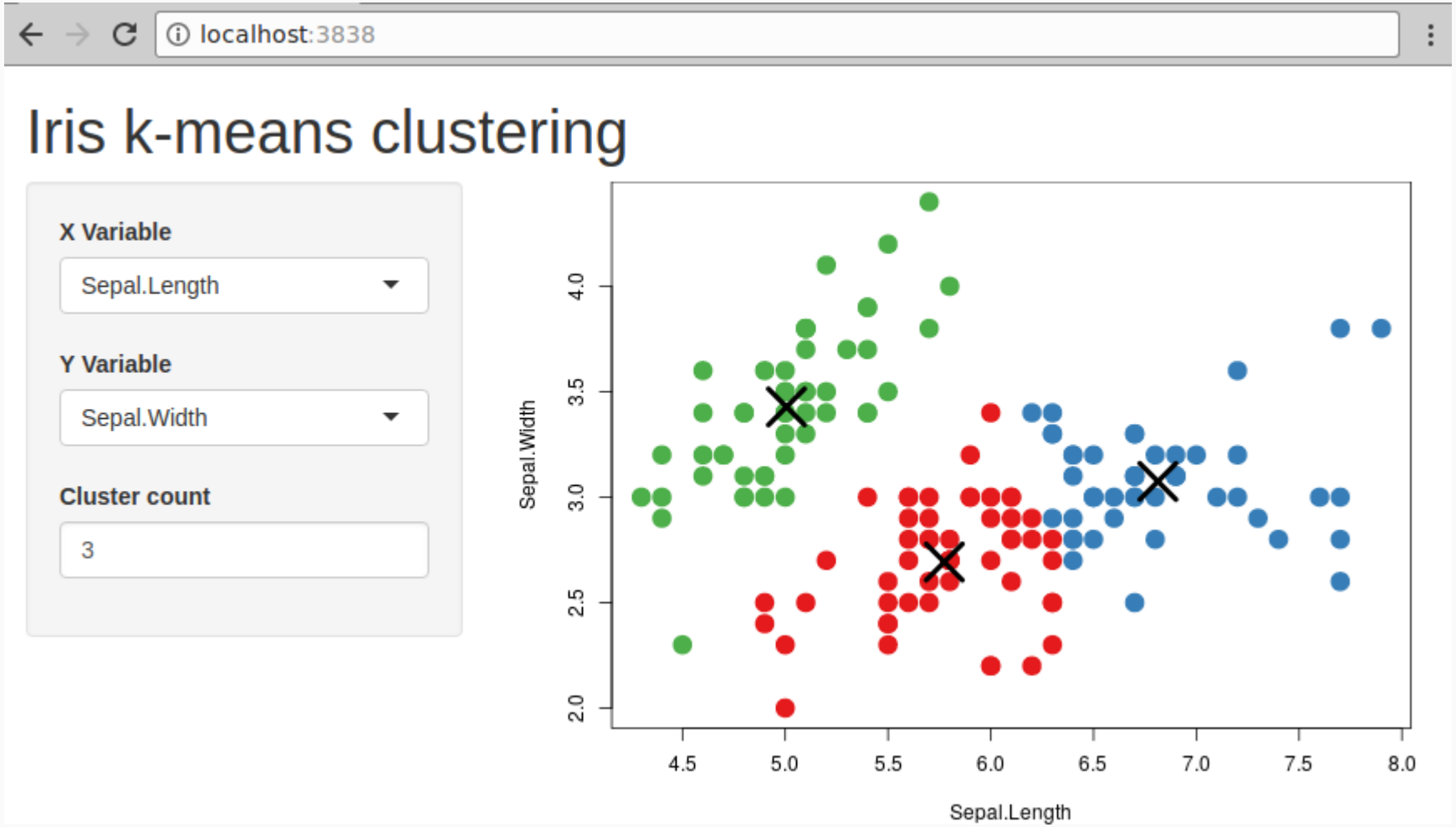
```
sudo docker run -i -t -p 8787:8787 rbigdata/workshop-rstudio
```





Shiny

```
sudo docker run -i -t -p 3838:3838 rbigdata/workshop-shiny-kmeans
```



Shiny

```
sudo docker run -i -t -p 3838:3838 rbigdata/workshop-shiny-movieexplorer
```

Shiny

```
sudo docker run -i -t -p 3838:3838 rbigdata/workshop-shiny-ggplotwithyourdata
```

Thanks!