Parallel Computing in R

Presentation by Victor Feagins

Introduction

Introduction

- Name: Victor Feagins
- Major: Statistics and Data Science
- Incoming PhD Student at University of Wisconsin-Madison

- Used parallel computing for various work/internship

Assumptions

- Concepts of Coding
- Beginner Statistics
- Basic R syntax will help

What is Parallel Computing?

Series: People in a line

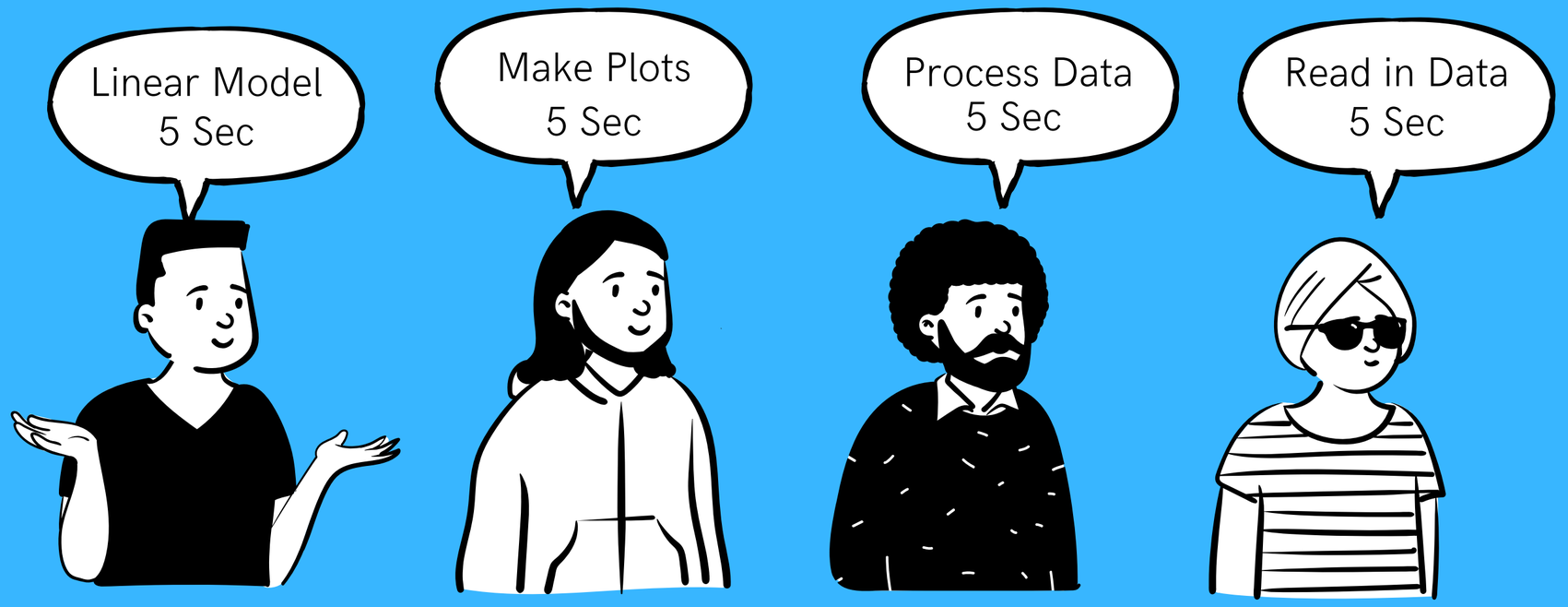
Series: People in a line

Series: People in a line

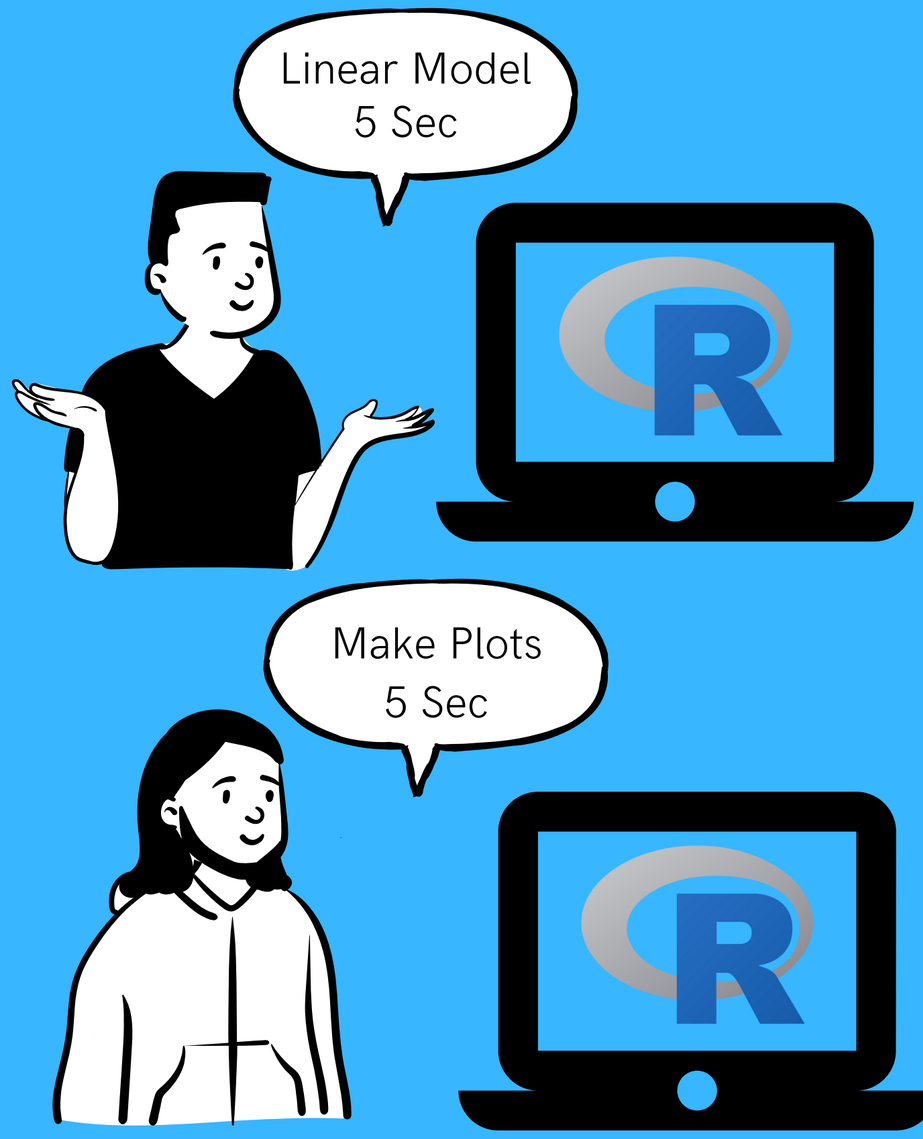
R runs code in series by default

Total Time:

20 seconds

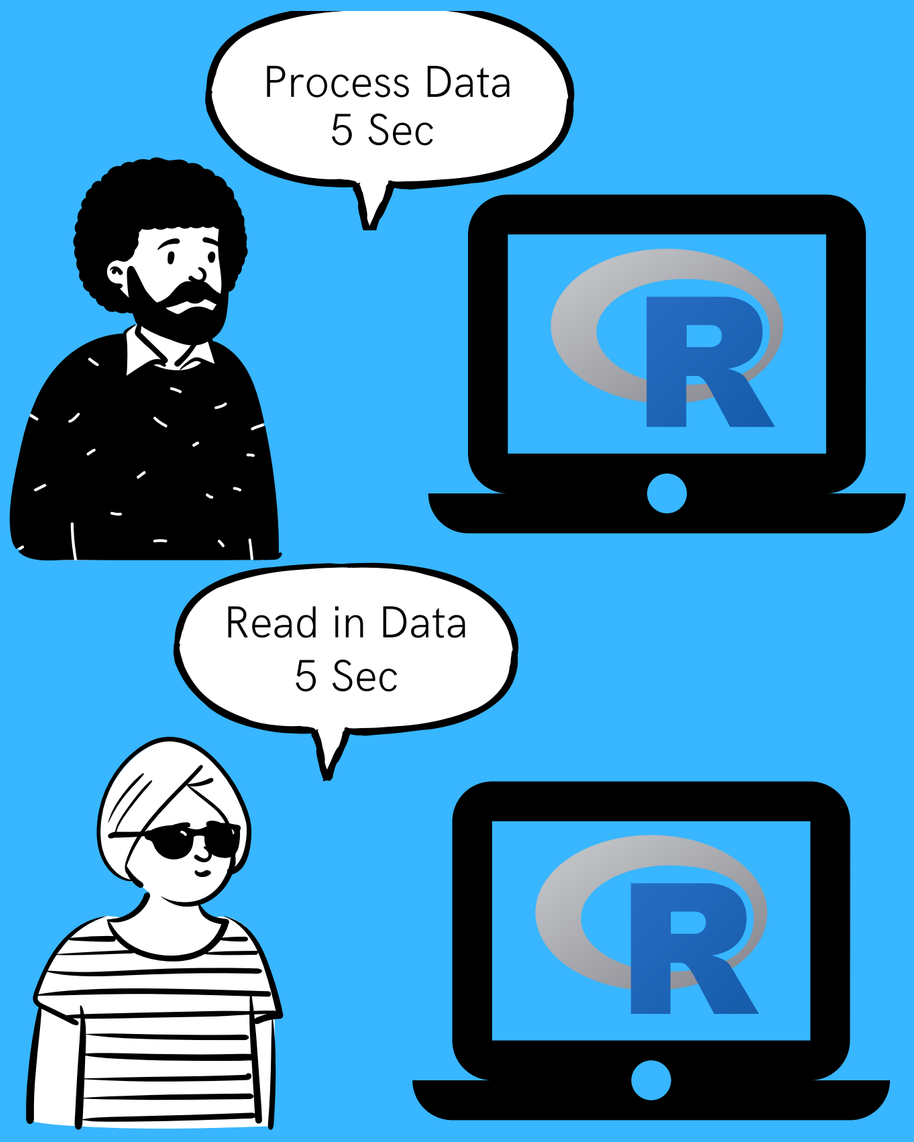


What if? Parallel!

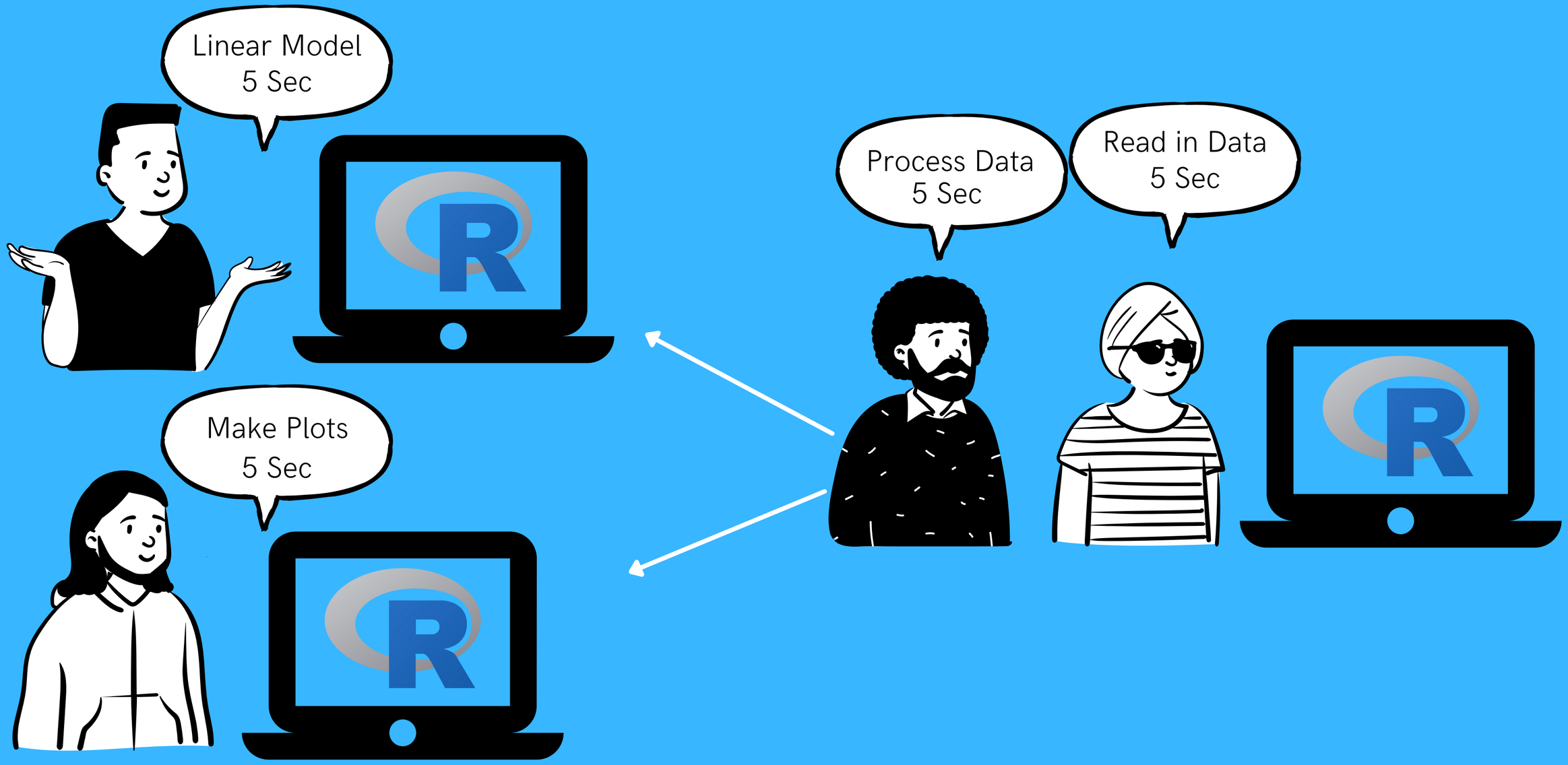


Total Time:

5 seconds!!!



More Realistic but not probable



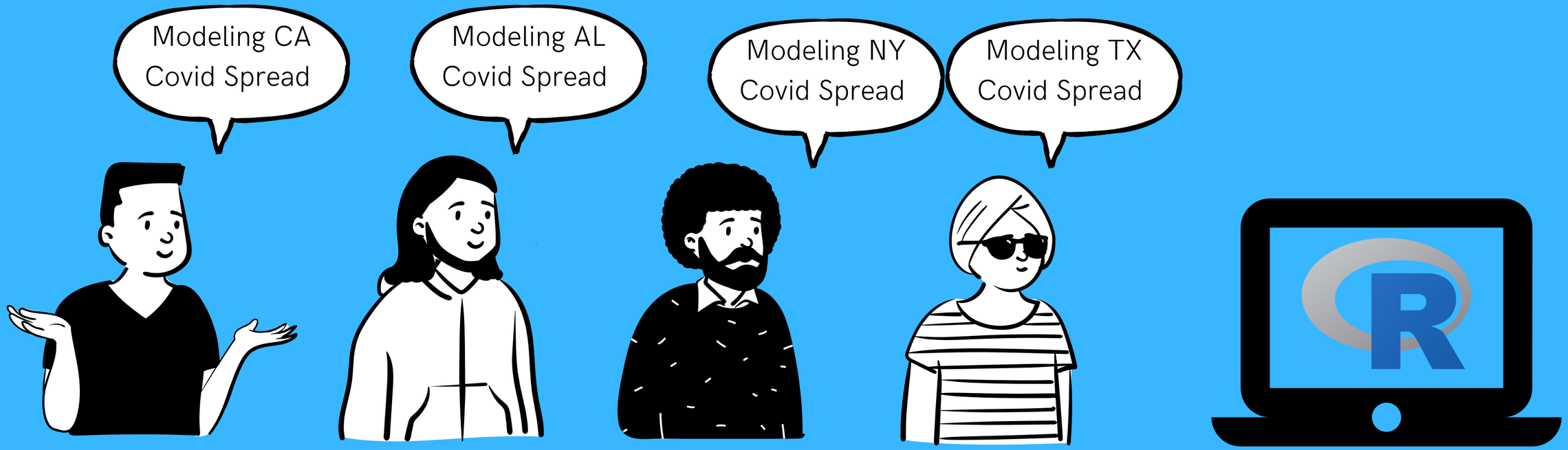
Ideal parallelizable problems

AKA Embarrassingly parallel

Total Time:

2 hours

30 min per



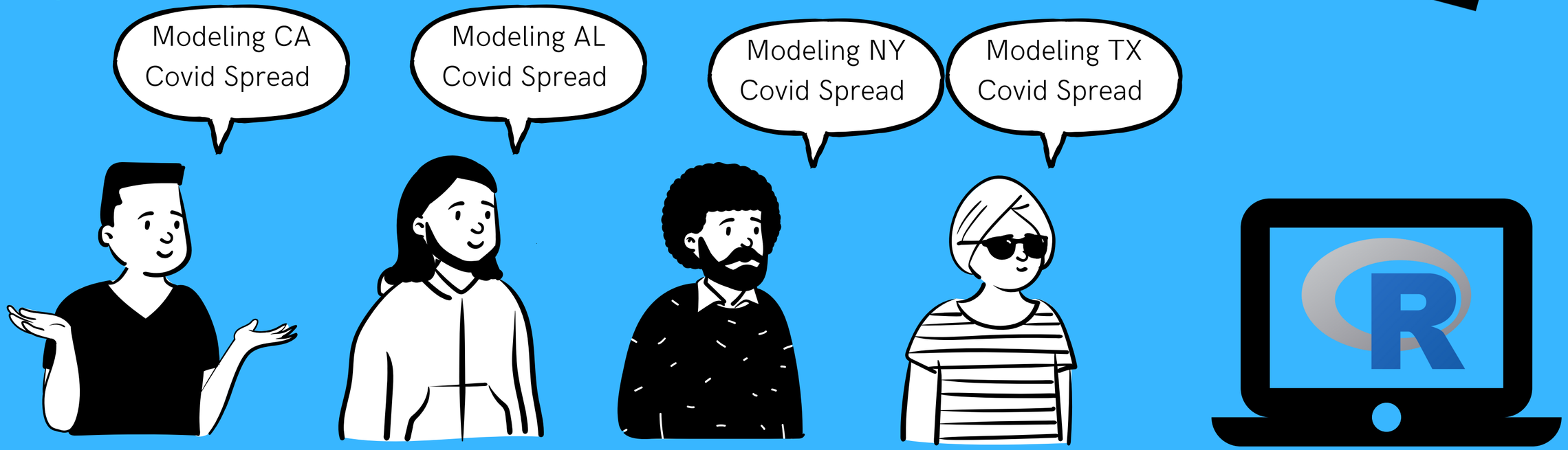
Ideal parallelizable problems

AKA Embarrassingly parallel

Total Time:

2 hours

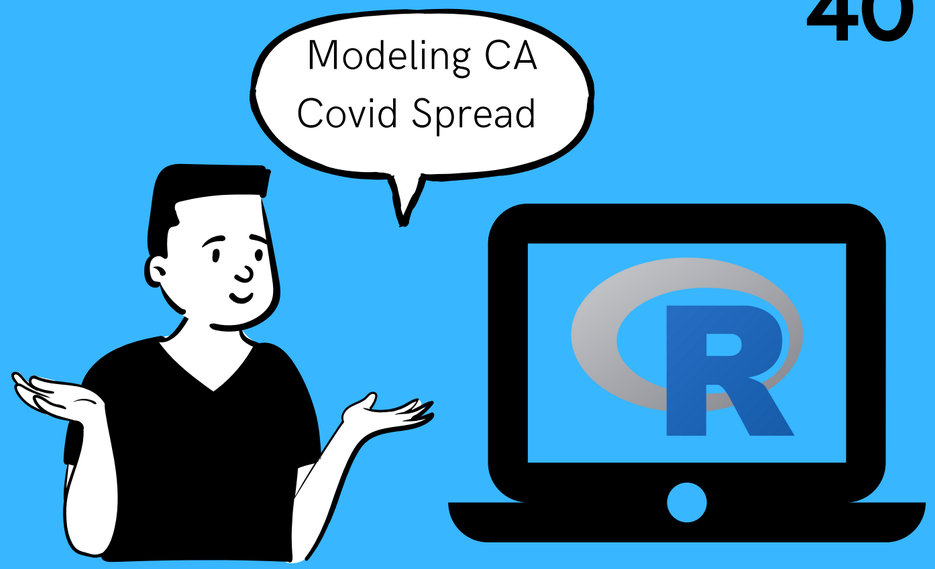
30 min per

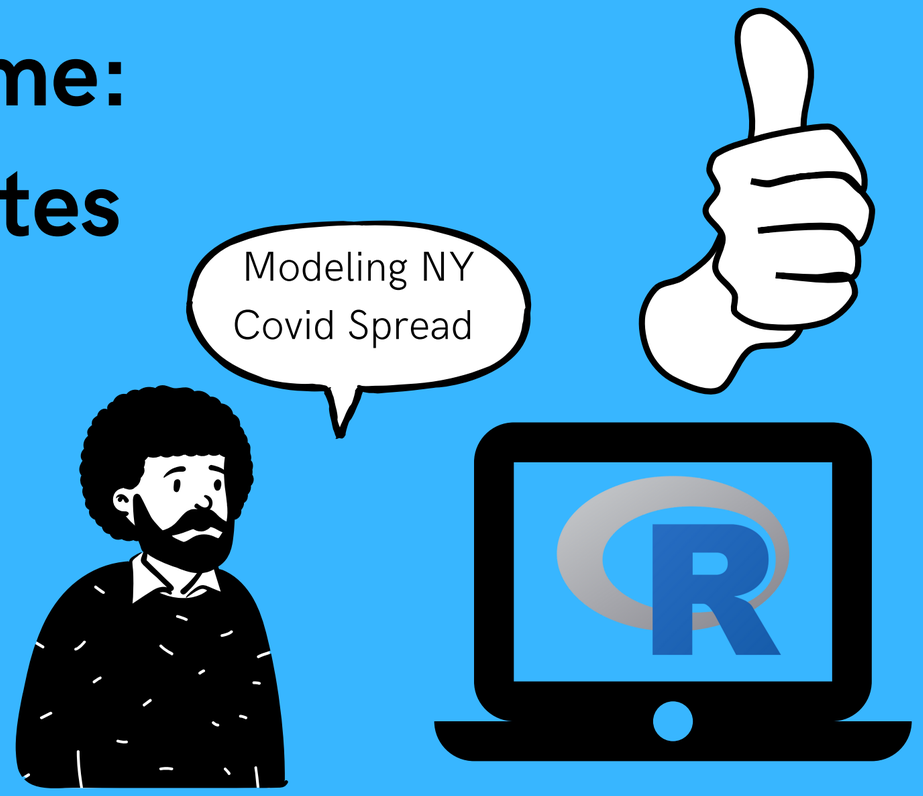


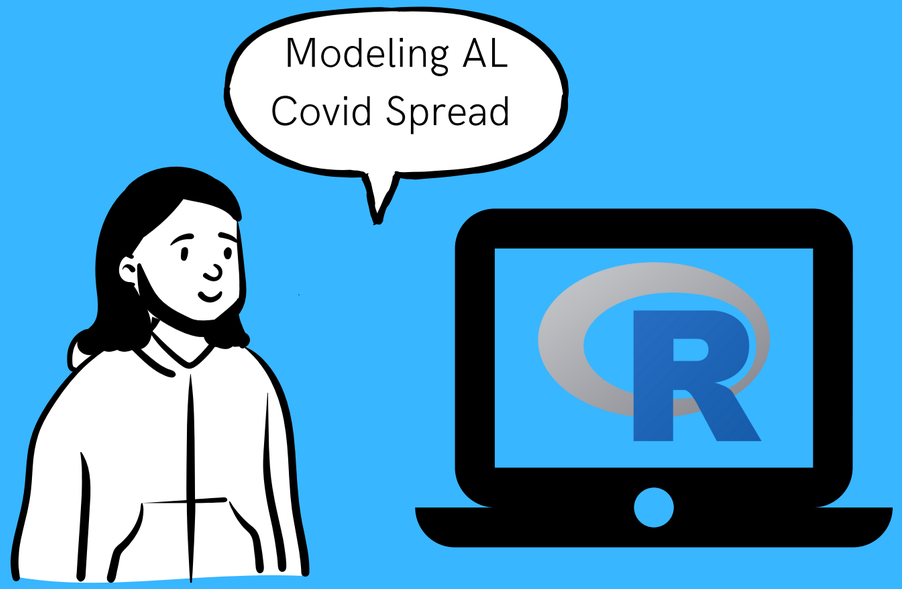
Parallel!

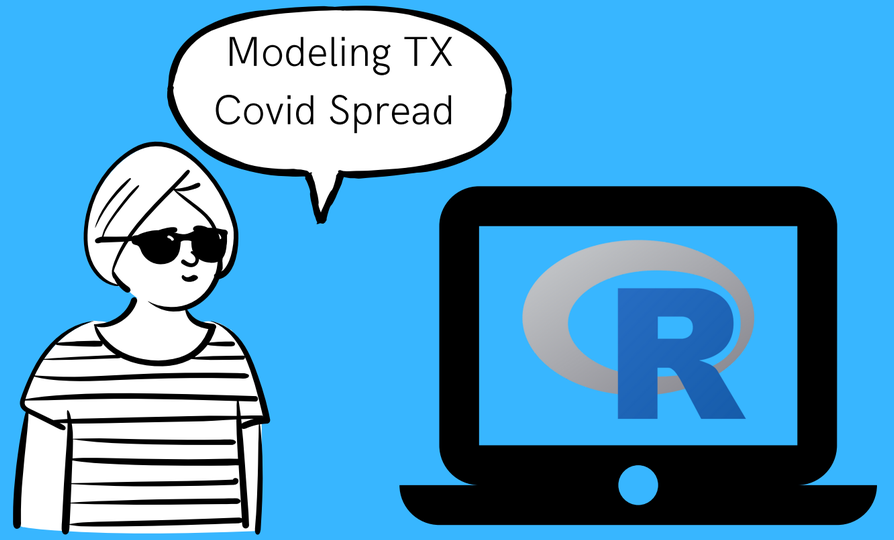
Total Time:

40 minutes

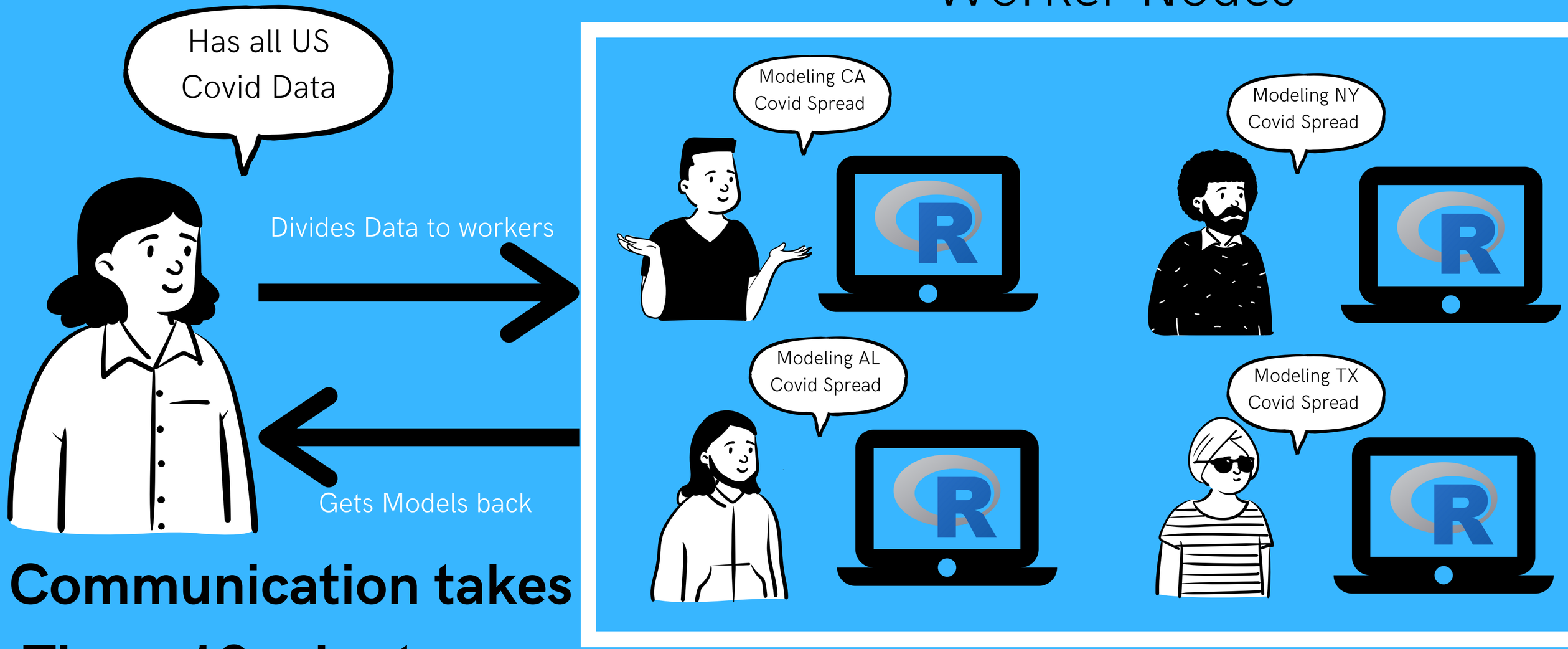








Comunication



Communication takes

Time: 10 minutes

Why do we do this?

Parallel Projects

- Compressing Large Amounts Of Data
  - Python with Message Passing Interface
  - Physics Simulation Data
  - Compressed thousands of CSV's and into HDF5

Parallel Projects

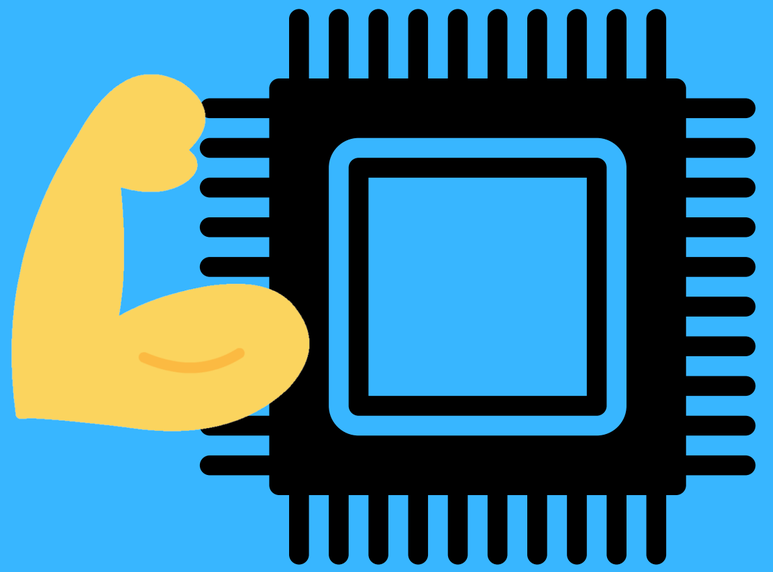
- Compressing Large Amounts Of Data
  - Python with Message Passing Interface
  - Physics Simulation Data
  - Compressed thousands of CSV's and into HDF5

- Modeling Time Series Satellite Data
  - R with Computing Clusters
  - ~5 min, 3 Satellite Images
  - Cleaned, Processed, Modeled
  - Daily

How do we do this?

Multi Core Processors

- CPU on computers today have multiple cores on them
- The first multicore processer invented 2001
- R language developed in 1993



Parallel Packages in R

- Parallel [R Baseline]
- Rmpi
- future
- Foreach
- Many more

More info:
https://cran.r-project.org/web/views/HighPerformanceComputing.html

The Parallel package

- Support for parallel computation
- Sockets(taken from package multicore)
- Forking (taken from package snow)（Windows can't do)
- Random-number generation.

Starting and stopping clusters

```
library(parallel)
detectCores() #How many cores do you have
```

```
[1] 8
```

```
cl <- makeCluster(2) #starting a cluster
#cl <- makeForkCluster(8) #fork clusters don't
work on windows
#stopCluster(cl) #Stopping cluster
```

Sending libraries to Clusters

```
clusterEvalQ(cl, {
  library(tidyverse)
  }) #clusterEvalQ make all the clusters run this
code as is
```

```
[[1]]
 [1] "forcats"   "stringr"   "dplyr"     "purrr"
"readr"     "tidyr"
 [7] "tibble"    "ggplot2"   "tidyverse" "stats"
"graphics"  "grDevices"
[13] "utils"     "datasets"  "methods"   "base"

[[2]]
 [1] "forcats"   "stringr"   "dplyr"     "purrr"
"readr"     "tidyr"
 [7] "tibble"    "ggplot2"   "tidyverse" "stats"
"graphics"  "grDevices"
[13] "utils"     "datasets"  "methods"   "base"
```

Sending functions and variables to Clusters

```
a <-  2
square <-  function (num){num**2}
clusterExport(cl,c("a","square")) #sends
variables and functions to clusters

clusterEvalQ(cl, {
  print(c(a,square(a)))
})
```

```
[[1]]
[1] 2 4

[[2]]
[1] 2 4
```

In series Time

```
ptm <- proc.time()
for (i in 1:5){
  Sys.sleep(3)
}
print(proc.time() - ptm)
```

```
   user  system elapsed
   0.02    0.00   15.01
```

In parallel Time

```
ptm <- proc.time()

invisible(parSapply(cl, rep(3,5),Sys.sleep))
#invisible here just hides the null list from
Sapply

print(proc.time() - ptm)
```

```
   user  system elapsed
   0.00    0.00    9.01
```

```
stopCluster(cl)
```

Let's go see an example!

Imagine

- We work for the Department of State Health Services
- Our task is to map every county's risk levels using the Census's Community Resilience Estimates

Workshop Folder: https://tinyurl.com/4jutmdyc

Wrap Up

- Introduced the Concepts of Parallel Computing
- Saw an Example in Action
- We just scratched the surface
- To Learn More:
  - Check out Future Package
    - :https://github.com/HenrikBengtsson/future
  - Youtube Videos

Contact Email: victor.feagins@my.utsa.edu

Practical

Workshop Folder: https://tinyurl.com/4jutmdyc