Big Data Systems R

Steven Nydick

Ben Wiseman

Jeff Jones

4/15/2021

Setup

1. Requires 
    - *https://www.r-project.org/*
2. Suggests 
    - *https://www.rstudio.com/products/rstudio/download/*
3. Materials
    - *https://bit.ly/32ag86B*

Package Installation

This demonstration requires the following packages:

1. Base R Section
    - *NOTHING*
2. Data Table Section
    - *data.table*
3. Tidyverse Section ...
    - *tidyverse ... which includes*
        - ggplot2
        - tibble
        - tidyr
        - readr
        - purrr
        - dplyr
        - stringr
        - forcats

Package Installation

You can install the packages with (modifying) the following line of code

```
install.packages("data.table")
install.packages("tidyverse")
```

And then load them with the following line of code

```
library(data.table)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.3     ✓ purrr   0.3.4
## ✓ tibble  3.1.0     ✓ dplyr   1.0.4
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::between()    masks data.table::between()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks data.table::first()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## x dplyr::last()       masks data.table::last()
## x purrr::transpose()  masks data.table::transpose()
```

What we want from you:

1. This session assumes little familiarity with `R`. If you have questions, please email the presenters. We have used `R` for a very long time and sometimes forget what we had to learn.
2. Try to run all of the code in `RStudio`. The setup of the demonstrations naturally works in `RStudio`. If you do not have `RStudio`, you can certainly run all of the code in `R` or a different IDE.
3. Have fun!

Overview of Data Systems

There are three major data systems in `R`, each with different purposes:

- Base `R`: Backwards compatibility. This is what comes with `R`, and the developers want to make sure that code that works now will work in the future. They are very hesitant at writing breaking code, so some odd and bad ideas either took a long to get rid of or are still there.
- Tidyverse: Maintainability. This is mostly developed by people at `RStudio`, and they have very strong beliefs at how code should run. They have modularized and separated a lot of their code so that it's relatively clear and easy to maintain as well as encompasing an environment of packages with similar standards.
- Data Table: Speed. This is developed to make read, process, and write speed as fast as possible.

Basic Functions

Each of the three data systems have functions that perform similar purposes.

```
x <- data.frame(a = 1:3, b = 4:6)
y <- data.frame(a = c(1, 2, 4), c = letters[1:3])

# three ways of joining
merge(x, y)      # base R
```

```
##   a b c
## 1 1 4 a
## 2 2 5 b
```

```
inner_join(x, y) # tidyverse
```

```
## Joining, by = "a"
```

```
##   a b c
## 1 1 4 a
## 2 2 5 b
```

```
as.data.table(x)[y, on = "a", nomatch = 0]
```

```
##    a b c
## 1: 1 4 a
## 2: 2 5 b
```

Keep in Mind

There are a few differences between the data systems.

1. Base `R`. These functions are often very slow and have a lot of inconsistent arguments.
2. Tidyverse. There are usually lots of functions with very specific purposes from lots of different packages that need to be all used to do something complex. They like to retire/obsolete functions/packages.
3. Data Table. The most efficient functions use a similar process: `dt[stuff][stuff]` rather than `g(f(stuff))` (from base `R`) or `f(stuff) %>% g(stuff)` from the tidyverse.

People who use the tidyverse seem to have been hatched in a pond where the tidyverse is the only mother duck around. They often don't know that other functions exist!

Function Dictionary

Here's a dictionary of similar operations across all three languages

Type	Base R	Tidyverse	data.table
Read CSV	`read.csv`	`readr::read_csv`	`data.table::fread`
Merging	`merge`	`dplyr::inner_join`	`dt[dt, on]`
		`dplyr::left_join`	`data.table::merge`
		`dplyr::right_join`	
		`dplyr::full_join`	
Combining	`rbind`	`dplyr::bind_rows`	`data.table::rbindlist`
Reshaping	`reshape`	`tidyr::pivot_wider`	`data.table::dcast`
		`tidyr::pivot_longer`	`data.table::melt`
Aggregating	`aggregate`	`dplyr::group_by`	`dt[i, j, by, .SDcols]`
		`dplyr::summarize`	

Exercises