data.table: Fast manipulation of large datasets in R

Brooke Anderson

April 25, 2016

data.table

data.table is a package in R that can efficiently read in and manipulate large datasets. It offers a **substantial** speed improvement over the classic data.frame when working with large datasets.

Example: US precipitation

As an example, I have a file with daily precipitation measures for every US county from 1979 through 2011:

- 365 days * 33
- ~3,000 counties

This file has $>$ 37,000,000 lines. The total file size is 2.26 GB.

Reading in a large text file

`fread` is the `data.table` equivalent of the `read.table` family of functions:

```
library(data.table)
system.time(precip <- fread(paste0(precip_dir,
                       "nasa_precip_export_2.txt"),
                      header = TRUE,
                      select = c("county",
                                 "year_month_day",
                                 "precip"),
                      verbose = FALSE))
```

```
##
Read 0.0% of 37496883 rows
Read 4.7% of 37496883 rows
Read 9.5% of 37496883 rows
Read 14.2% of 37496883 rows
Read 18.9% of 37496883 rows
Read 23.6% of 37496883 rows
Read 28.2% of 37496883 rows
Read 33.0% of 37496883 rows
```

Reading in a large text file

`fread` can also read a file directly from http and https URLs, if you'd prefer to not save the flat file locally.

Manipulating a data.table

The data.table class has a series of conventions for summarizing and indexing that runs much, much faster than if you tried to use “classic” R functions.
The general form is:

```
precip[i, j, by]
```

where i filters by row, j selects or calculates on columns, and by groups by some grouping variable when selecting or calculating using columns.

Manipulating a `data.table`

You can use the first element to filter to certain rows. For example, to pull out just values for Larimer County, CO, run:

```
precip[county == 8069 &
         year_month_day %in%
         c(19970727, 19970728), ]
```

```
##    county year_month_day precip
## 1:   8069       19970727    6.1
## 2:   8069       19970728   15.4
```

Manipulating a data.table

You can use the order function in the first element to sort the data:

```
head(precip[order(-precip), ])
```

```
##    county year_month_day precip
## 1:  51133       20100930  251.1
## 2:  24037       20100930  232.8
## 3:  37095       20110827  232.2
## 4:  37013       20110827  232.1
## 5:  22043       20061016  229.2
## 6:  22059       20061016  227.6
```

Manipulating a data.table

You can run calculations on columns using the second element:

```r
precip[ , max(precip)]
```

```
## [1] 251.1
```

```r
precip[ , quantile(precip,
                   probs = c(0.99, 0.999,
                             0.9999))]
```

```
##    99%  99.9% 99.99%
##   32.8   64.5  103.6
```

Manipulating a data.table

You can combine filtering by rows and calculating on columns. For example, to figure out how many counties there were in 2011:

```
precip[year_month_day == 20110101,
       length(precip)]
```

```
## [1] 3111
```

Note: If you want to count rows, you can also use `.N`:

```
precip[year_month_day == 20110101,
       .N]
```

```
## [1] 3111
```

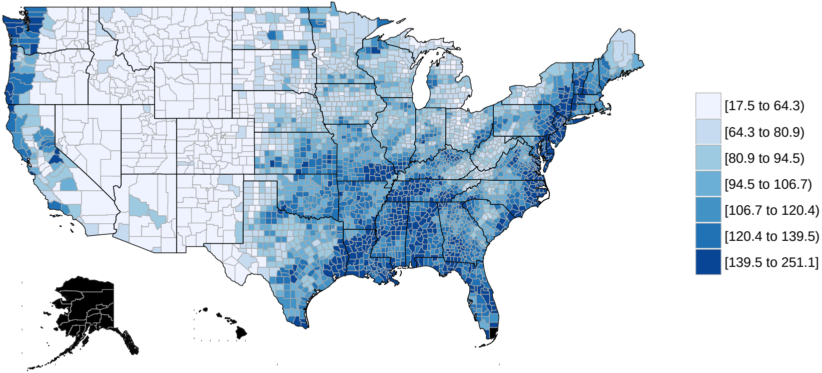
Grouped analysis

You can also group by a variable before you run an analysis. For example, to get the highest recorded precipitation in each county:

```
highest_precip <- precip[ , .(max.precip = max(precip)),
                          by = .(county)]
head(highest_precip, 3)
```

```
##    county max.precip
## 1:  45031       99.7
## 2:  42061       96.8
## 3:   8011       54.4
```

Highest precipitation by county

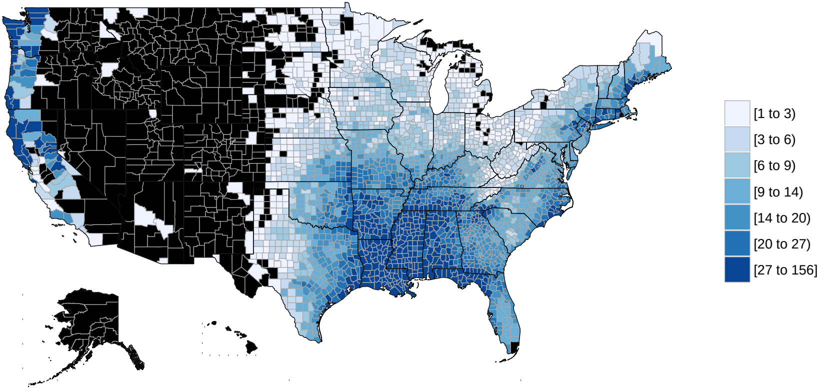


Chaining operation with data.table

If you want to, you can chain together several operations. For example, to determine the number of days over the 99.9th percentile in each county:

```
extreme_precip <- precip[ , .N, .(precip >
                                    quantile(precip,
                                             probs = 0.999),
                                 county)][
                          precip == TRUE,
                        ]
```

Extreme precipitation by county

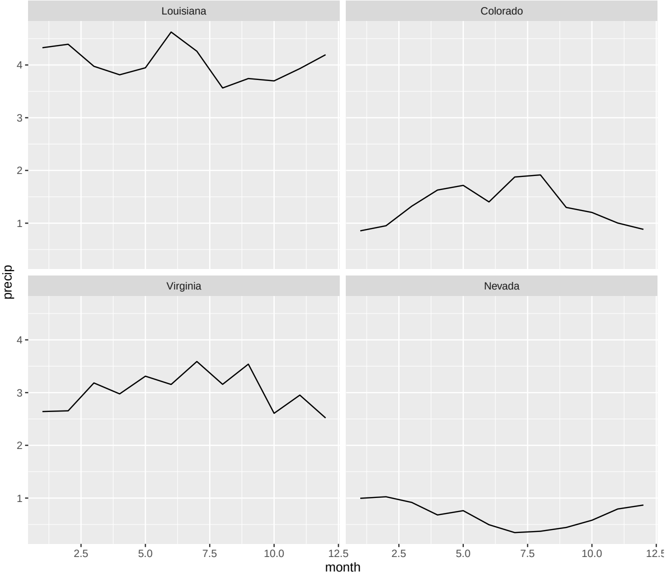


Chaining operation with data.table

To plot trends by month within states:

```
ts_precip <- precip[ , .(precip = precip,
                         state = substring(sprintf("%05d",
                                                   county),
                                           1, 2),
                         month = as.numeric(
                           substring(year_month_day,
                                           5, 6)))][
                     , .(precip = mean(precip)),
                     keyby = .(state, month)
                     ]
```

Precipitation by month and state


Louisiana
Colorado
Virginia
Nevada
precip
month