Ministry
of Justice

Start using data.table

Coffee and Coding 09/07/19

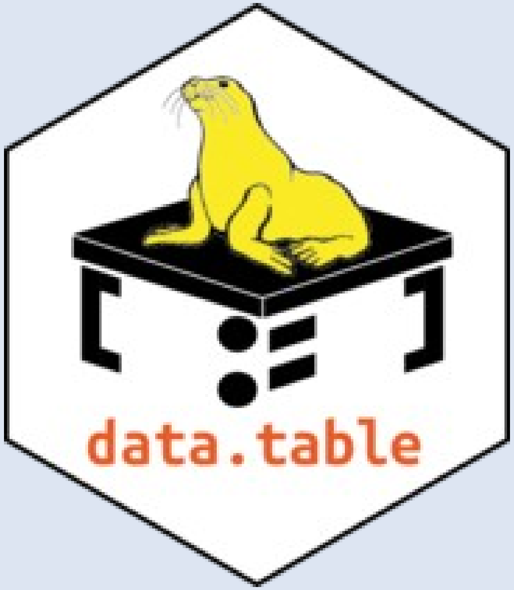
Megan Stodel

Protecting and advancing the principles of justice

About data.table

Defining data.table

- A package in R
  - Originally released in 2006
  - 730k downloads a month
- A data structure
- A way of manipulating data



Why you should use data.table

Speed matters

Speed can be particularly important in some instances:

- Very big datasets
- Apps or tools designed for general use
- Code that you run often

data.table is fast

- One of the most common criticisms of R is that it is slow

- Not data.table!

  - ✓ Things can be modified / altered by reference, so there is in-situ replacement without duplicating the table

  - ✓ The binary search algorithm means it efficiently finds values by searching a small section of the sorted data

  - ✓ You can perform numerous operations in one line, so don’t have to allocate memory for the intermediate result

  - ✓ Speed extends to reading in data using fread()

Benchmarks

-- Aggregation benchmarks here –

- Data.table is consistently substantially faster, not only than dplyr, but also pandas and data structures in other languages.

- Relative performance increases as data size increases.

Few dependencies

DATA.TABLE

- methods (base)

DPLYR

- assertthat
- glue
- magrittr
- pkgconfig
- R6
- Rcpp
- rlang
- tibble
- tidyselect
- utils (base)

Concise and consistent syntax

- Code is often shorter

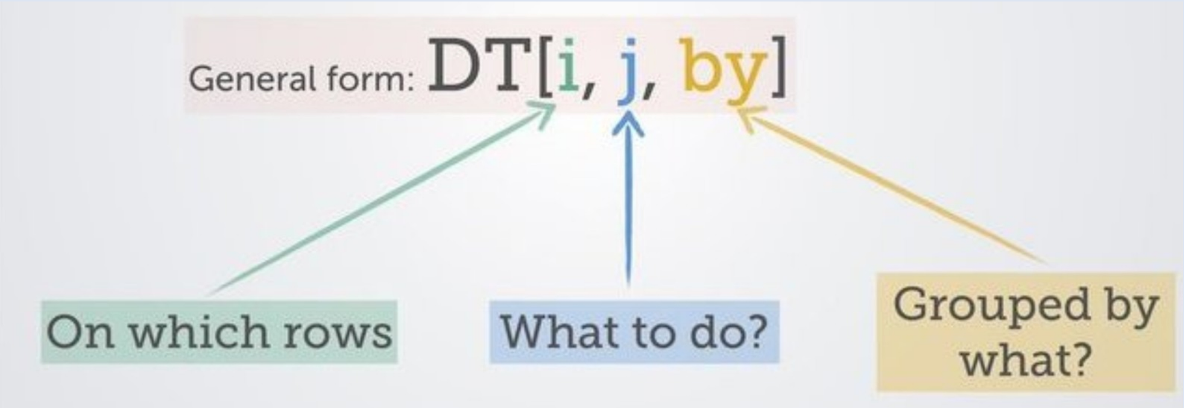
- Consistent logic, once learnt, is straightforward and easy to apply for diverse needs – building blocks rather than individual functions

- Different people will prefer different things

- data.tables are data.frames – so can still use data.frame commands on them

Using data.table

The syntax



(source)

In-situ definition and replacement

:=

This combination of characters is how to create or modify a column without needing to create a copy of the dataset.

```
DT[, bigger_number := number_column + 1]
```

Built-in variables

Symbol	Purpose
.N	Number of observations in the group DT[, .N] Counts the number of observations (rows) in the data.table
.SD	**S**ubset of **D**ata. A way of referencing all columns except the grouping columns. So rather than individually calling functions on each column, can use .SD to apply to all of them, used in conjunction with base function lapply(). DT[, lapply(.SD, mean)] Calculates the mean for every column

Lists in data.table

As long as *j* returns a list, each element of the list will become a column in the resulting data.table.

If you only want certain columns in your data.table, you can achieve this using a list, which in its short version is .()

```
twocol_DT <- DT[, .(column_a, column_b)]
```

This is true even if you want a one column data.table

```
onecol_DT <- DT[, .(column_a)]
```

If you don’t have your single column in a list, it will become a vector (which is often useful)

```
vector <- DT[, column_a]
```

Changing column names

Rename by reference with setnames()

```
    setnames(DT, “original_name”, “new_name”)
 setnames(DT, c(“a”, “b”, “c”), c(“A”, “B”, “C”))
```

Or change names as you choose which columns to keep

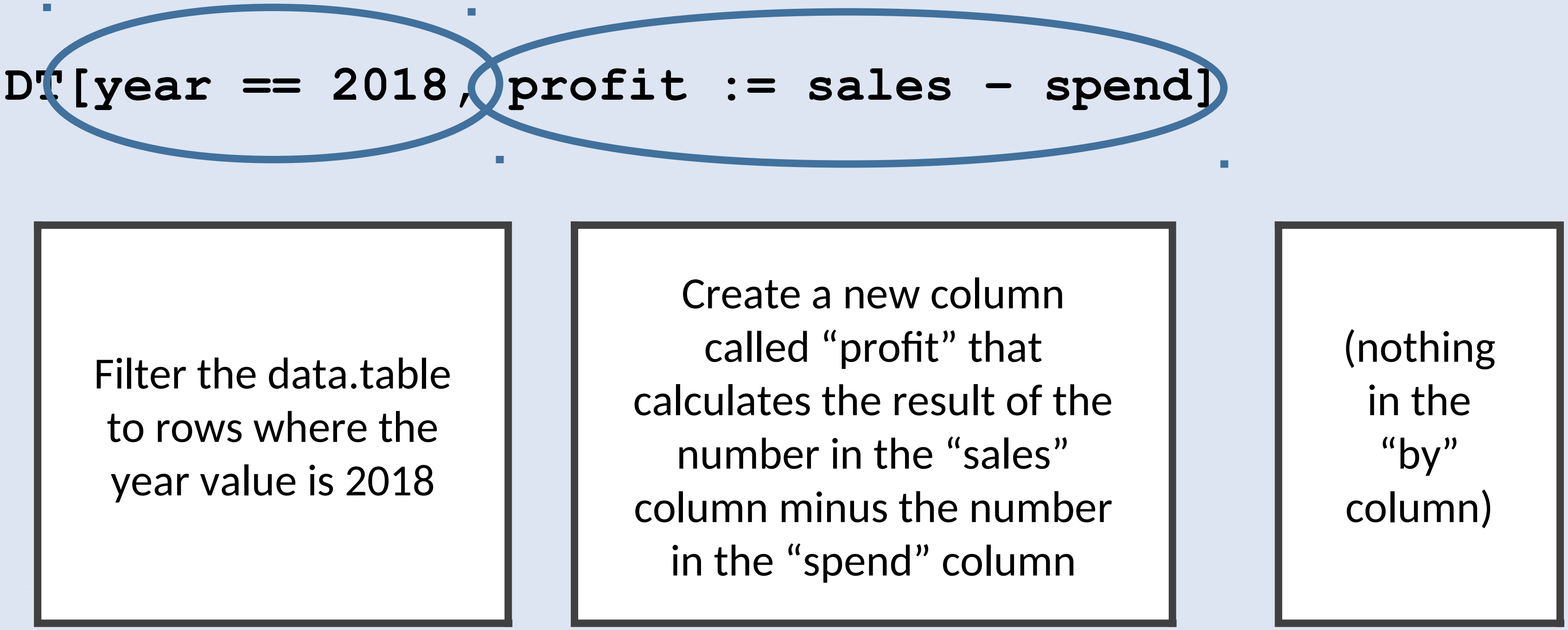
```
new_DT <- DT[, .(new_name = old_name,
                 new_column = old_column)]
```

Chaining

Similar to piping, you can chain data.table commands

```
DT[, .N, by = month][order(-N)][1:3]
```

What is happening here?



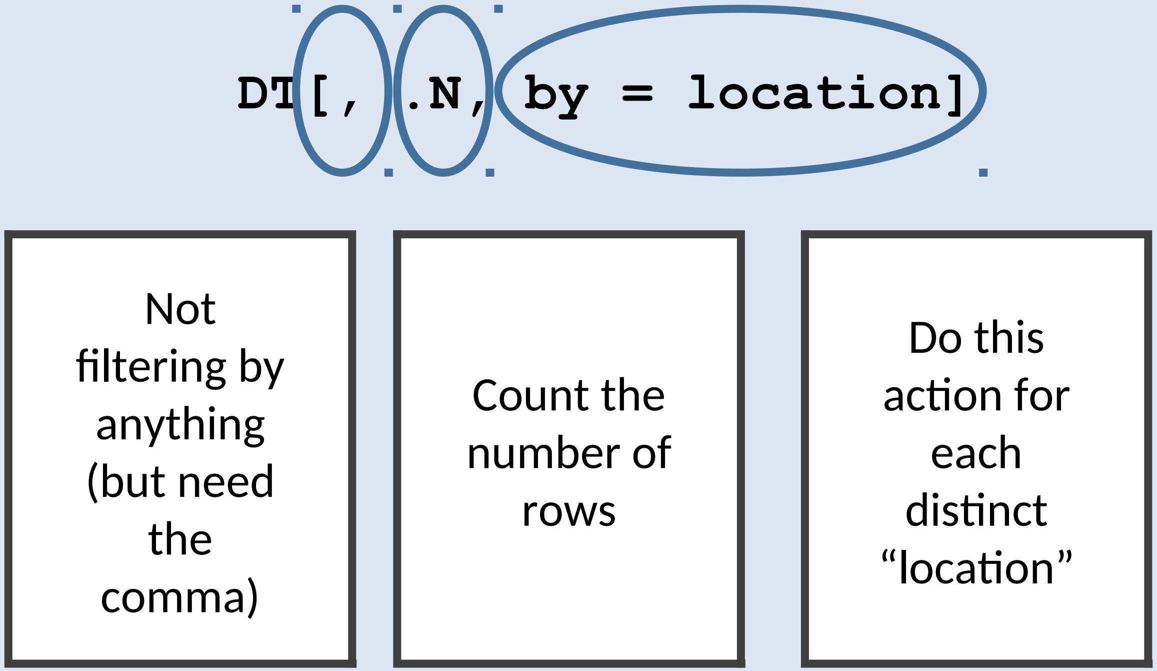
```
DT[year == 2018, profit := sales - spend]
```

Filter the data.table to rows where the year value is 2018

Create a new column called “profit” that calculates the result of the number in the “sales” column minus the number in the “spend” column

(nothing in the “by” column)

What is happening here?



```
DT[, .N, by = location]
```

Not filtering by anything (but need the comma)

Count the number of rows

Do this action for each distinct “location”

Your turn!

Exercises

Set up with the following code:

```
# install.packages(data.table)
library(data.table)
chick_weight <- as.data.table(ChickWeight)
```

1. Which diet is being fed to the most chicks?

2. What is the average (mean) weight of a chick at time 21 for each diet?

3. Add a new column that is TRUE when weight is 100 or more.

4. Make every column a character class.

5. Rename the columns (to anything you like).

If watching this later, find the solutions in the repo:
https://github.com/moj-analytical-services/coffee-and-coding-public

Useful resources

data.table site

data.table FAQ

A data.table and dplyr tour (includes comparison of operations in both packages)

Advanced tips and tricks with data.table (this is so good!)

Questions?

Suggestions for future sessions?