DATA.TABLE

Olivia Johnson

Adelaide Code Club

11 August 2022

OUTLINE FOR TODAY

- Recap of last week
- Why use data.table
- Reading in data
- Syntax
- Applying functions
- Exercise!

RECAP

- For loops and nested for loops
  - ```
    for (i in sequence){
        statement
    }
    ```
  - ```
    for (x in x_vals){
        for (y in y_vals){
            print(paste("x =", x, ", y =", y))
        }
    }
    ```

RECAP

- If else statements

```
if (condition){
    statement
} else {
    statement
}
```

RECAP

- apply functions
  - **`apply(x, MARGIN (1 for rows, 2 for columns), function)`**
    - For data frame or matrix
  - **`lapply(x, function)`**
    - Output is a **l**ist
  - **`sapply(x, function)`**
    - **S**implified, output is a vector.

WHY DATA.TABLE

- Considered the fasted R package for data manipulation
- R generally thought not suitable for big data (>10 GB)
  - Not memory efficient.
- Benchmarked against dplyr and pandas (python), data.table was best.

READING IN DATA

- **fread**(path)
  - Will read data or webpage in as data.table
- Can also convert pre-existing R objects
  - **setDT**(data frames and lists)
  - **as.data.table**(for other structures)
- Or create using **data.table**(values)

STRUCTURE OF A DATA.TABLE

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	fast	cars	carname
1:	4.582576	6	160.0	110	3.90	2.620	16.46	0	1	4	4	1	Mazda RX4	Mazda RX4
2:	4.582576	6	160.0	110	3.90	2.875	17.02	0	1	4	4	1	Mazda RX4 Wag	Mazda RX4 Wag
3:	4.774935	4	108.0	93	3.85	2.320	18.61	1	1	4	1	1	Datsun 710	Datsun 710
4:	4.626013	6	258.0	110	3.08	3.215	19.44	1	0	3	1	1	Hornet 4 Drive	Hornet 4 Drive
5:	4.324350	8	360.0	175	3.15	3.440	17.02	0	0	3	2	1	Hornet Sportabout	Hornet Sportabout
6:	4.254409	6	225.0	105	2.76	3.460	20.22	1	0	3	1	1	Valiant	Valiant
7:	3.781534	8	360.0	245	3.21	3.570	15.84	0	0	3	4	0	Duster 360	Duster 360
8:	4.939636	4	146.7	62	3.69	3.190	20.00	1	0	4	2	1	Merc 240D	Merc 240D
9:	4.774935	4	140.8	95	3.92	3.150	22.90	1	0	4	2	1	Merc 230	Merc 230
10:	4.381780	6	167.6	123	3.92	3.440	18.30	1	0	4	4	1	Merc 280	Merc 280
11:	4.219005	6	167.6	123	3.92	3.440	18.90	1	0	4	4	1	Merc 280C	Merc 280C
12:	4.049691	8	275.8	180	3.07	4.070	17.40	0	0	3	3	1	Merc 450SE	Merc 450SE
13:	4.159327	8	275.8	180	3.07	3.730	17.60	0	0	3	3	1	Merc 450SL	Merc 450SL
14:	3.898718	8	275.8	180	3.07	3.780	18.00	0	0	3	3	0	Merc 450SLC	Merc 450SLC
15:	3.224903	8	472.0	205	2.93	5.250	17.98	0	0	3	4	0	Cadillac Fleetwood	Cadillac Fleetwood

BASIC SYNTAX

DT[i, j, by]

- Use square brackets
- **i** subsets rows
- **j** subsets columns
- Group by **by**

SUBSETTING

- To subset rows
  - `DT[2:3]`
  - `DT[column==x]`
- To subset columns
  - `DT[, col_name]` (will return as vector)
  - `DT[,.(col1, col2)]` (will return as data.table)

GROUPING BY

- DT[i, j, by=col_name]
- Apply j to groups of values of col_name
- Can use multiple columns names, just have to be in a vector.
  - i.e. c(“cyl”, ”fast”)

USING FUNCTIONS

- **.N** - returns the number of rows, goes in j
- DT[, .N, by="cyl")

```
> mtcars[, .N, by="cyl"]
   cyl  N
1:   6  7
2:   4 11
3:   8 14
```

USING FUNCTIONS

- Can also use regular functions
- DT[, **sum**(column)]
- DT[, **mean**(col2),by="col1"]
  - This will not change the table, just output the result

EXERCISE TIME!