DATA MANIPULATION IN R WITH DATA.TABLE

Computations by groups

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

The by argument

The `by` argument allows computations for each unique value of the (grouping) columns specified in `by`

```
# How many trips happened from each start_station?
ans <- batrips[, .N, by = "start_station"]
head(ans, 3)
             start_station     N
1: San Francisco City Hall  2145
2:  Embarcadero at Sansome 12879
3:       Steuart at Market 11579
```

The by argument

`by` argument accepts both `character` vector of column names as well as a `list` of variables/expressions

```
# Same as batrips[, .N, by = "start_station"]
ans <- batrips[, .N, by = .(start_station)]
head(ans, 3)
             start_station     N
1: San Francisco City Hall  2145
2:  Embarcadero at Sansome 12879
3:       Steuart at Market 11579
```

The by argument

Allows renaming grouping columns on the fly

```
ans <- batrips[, .(no_trips = .N), by = .(start = start_station)]
head(ans, 3)
                     start no_trips
1: San Francisco City Hall     2145
2:  Embarcadero at Sansome    12879
3:       Steuart at Market    11579
```

Expressions in by

The `list()` or `.()` expression in `by` allows for grouping variables to be computed on the fly

```
# Get number of trips for each start_station for each month
ans <- batrips[ , .N, by = .(start_station, mon = month(start_date))]
head(ans, 3)
             start_station mon   N
1: San Francisco City Hall   1 193
2:  Embarcadero at Sansome   1 985
3:       Steuart at Market   1 813
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!

DATA MANIPULATION IN R WITH DATA.TABLE

Chaining data.table expressions

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

Chaining expressions

data.table expressions can be chained together, i.e., `x[...][...][...]`

```
step_1 <- batrips[duration > 3600]
step_2 <- step_1[duration > 3600][order(duration)]
step_2[1:3]
```

```
# Same as
batrips[duration > 3600]

batrips[duration > 3600][order(duration)]

batrips[duration > 3600][order(duration)][1:3]
   trip_id duration
1:  295912 3601     ...
2:  347471 3602     ...
3:  536050 3602     ...
```

Chaining expressions

```
# Three start stations with the lowest mean duration
step_1 <- batrips[, .(mn_dur = mean(duration)), by = "start_station"]
step_2 <- step_1[order(mn_dur)]
step_2[1:3]
```

```
# Three start stations with the lowest mean duration
batrips[, .(mn_dur = mean(duration)),
        by = "start_station"][order(mn_dur)][1:3]
```

```
                                start_station   mn_dur
1:                                2nd at Folsom 551.0807
2: Temporary Transbay Terminal (Howard at Beale) 655.8563
3:                             2nd at South Park 697.7034
```

uniqueN()

- `uniqueN()` is a helper function that returns an integer value containing the number of unique values in the input object
- It accepts vectors as well as `data.frames` and `data.tables`.

```
id <- c(1, 2, 2, 1)
uniqueN(id)
[1] 2
```

```
x <- data.table(id, val = 1:4)
   id val
1:  1   1
2:  2   2
3:  2   3
4:  1   4

uniqueN(x)
[1] 4

uniqueN(x, by = "id")
[1] 2
```

uniqueN() together with by

Calculate the total number of *unique* bike ids for every month

```
ans <- batrips[, uniqueN(bike_id), by = month(start_date)]
head(ans, 3)
   month  V1   ## <~~ auto naming of cols
1:     1 605
2:     2 608
3:     3 631
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!

DATA MANIPULATION IN R WITH DATA.TABLE

Computations in j using .SD

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

Subset of Data, .SD

- `.SD` is a special symbol which stands for **S**ubset of **D**ata
- Contains subset of data corresponding to each group; which itself is a data.table
- By default, the grouping columns are excluded, for convenience

```
x <- data.table(id = c(1, 1 ,2, 2, 1, 1),
                val1 = 1:6, val2 = letters[6:1])
  id val1 val2
1:  1    1    f
2:  1    2    e
3:  2    3    d
4:  2    4    c
5:  1    5    b
6:  1    6    a
```

Subset of Data, .SD

```
x[, print(.SD), by = id]

   val1 val2
1:    1    f
2:    2    e
3:    5    b
4:    6    a
   val1 val2
1:    3    d
2:    4    c
Empty data.table (0 rows) of 1 col: id
```

Subset of Data, .SD

```
x[, .SD[1], by = id]
   id val1 val2
1:  1    1    f
2:  2    3    d
```

Subset of Data, .SD

```
x[, .SD[.N], by = id]
   id val1 val2
1:  1    6    a
2:  2    4    c
```

.SDcols

`.SDcols` holds the columns that should be included in `.SD`.

```
batrips[, .SD[1], by = start_station]

             start_station trip_id duration            start_date  ...
1: San Francisco City Hall  139545      435   2014-01-01 00:14:00  ...
2:  Embarcadero at Sansome  139547     1523   2014-01-01 00:17:00  ...

# .SDcols controls the columns .SD contains
batrips[, .SD[1], by = start_station, .SDcols = c("trip_id", "duration")]

             start_station trip_id duration
1: San Francisco City Hall  139545      435
2:  Embarcadero at Sansome  139547     1523

batrips[, .SD[1], by = start_station, .SDcols = - c("trip_id", "duration")]

             start_station  start_date           ...
1: San Francisco City Hall  2014-01-01 00:14:00  ...
2:  Embarcadero at Sansome  2014-01-01 00:17:00  ...
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!