Computations by groups

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

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
San Francisco City Hall  2145
 Embarcadero at Sansome 12879
      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
San Francisco City Hall  2145
 Embarcadero at Sansome 12879
      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
San Francisco City Hall       2145
 Embarcadero at Sansome      12879
      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
San Francisco City Hall   1  193
 Embarcadero at Sansome   1  985
      Steuart at Market   1  813
```

Let's practice!

DATA MANIPULATION WITH DATA.TABLE IN R

Chaining data.table expressions

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

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
 295912 3601
 347471 3602
 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
                              2nd at Folsom 551.0807
 Temporary Transbay Terminal (Howard at Beale) 655.8563
                             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)
```

```
2
```

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

```
id val
 1   1
 2   2
 2   3
 1   4
```

```
uniqueN(x)
```

```
4
```

```
uniqueN(x, by = "id")
```

```
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   605
     2   608
     3   631
```

Let's practice!

DATA MANIPULATION WITH DATA.TABLE IN R

Computations in j using .SD

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

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    f
 1    2    e
 2    3    d
 2    4    c
 1    5    b
 1    6    a
```

Subset of Data, .SD

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

```
val1 val2
   1    f
   2    e
   5    b
   6    a
val1 val2
   3    d
   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    f
 2    3    d
```

Subset of Data, .SD

```
x[, .SD[.N], by = id]
```

```
id val1 val2
 1    6    a
 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
 San Francisco City Hall    139545        435   2014-01-01 00:14:00
  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
San Francisco City Hall    139545        435
 Embarcadero at Sansome    139547       1523
```

.SDcols

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

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

Let's practice!

DATA MANIPULATION WITH DATA.TABLE IN R