DATA MANIPULATION IN R WITH DATA.TABLE

Welcome to the course!

Matt Dowle and Arun Srinivasan
Instructors, DataCamp

What is a data.table?

- Enhanced `data.frame`
    - inherits from and extends `data.frame`
- Columnar data structure
- Every column must be of same length but can be of different type

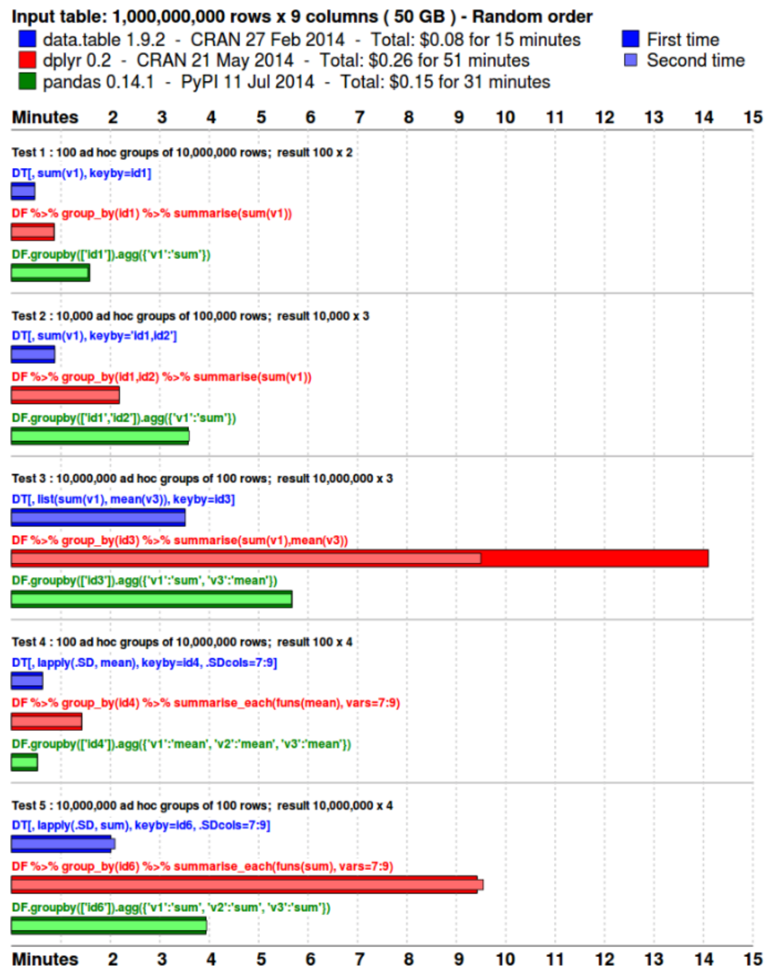
Why use data.table?

- Concise and consistent syntax
  - Think in terms of `rows`, `columns` and `groups`
  - provides a *placeholder* for each

```
# General form of data.table syntax
DT[i, j, by]
   |  |  |
   |  |  --> grouped by what?
   |  -----> what to do?
   --------> on which rows?
```

Why use data.table?

Fast and memory efficient



Why use data.table?

- Feature-rich
    - Parallelisation
    - Fast updates *by reference*
    - Powerful joins (Joining Data in R with data.table)

Creating a data.table (I)

Three ways of creating data tables:

- `data.table()`
- `as.data.table()`
- `fread()`

Creating a data.table (II)

```
library(data.table)

x_df <- data.frame(id = 1:2,
                   name = c("a", "b"))
x_df
  id name
1  1    a
2  2    b

x_dt <- data.table(id = 1:2,
                   name = c("a", "b"))
x_dt
   id name
1:  1    a
2:  2    b
```

Creating a data.table (III)

```
y <- list(id = 1:2, name = c("a", "b"))
y
$id
[1] 1 2

$name
[1] "a" "b"

x <- as.data.table(y)
x
   id name
1:  1    a
2:  2    b
```

data.tables and data.frames (I)

Since a data.table *is* a data.frame ...

```
x <- data.table(id = 1:2,
                name = c("a", "b"))
x
   id name
1:  1    a
2:  2    b

class(x)
[1] "data.table" "data.frame"
```

data.tables and data.frames (II)

Functions used to query data.frames also work on data.tables

```
nrow(x)
[1] 2

ncol(x)
[1] 2

dim(x)
[1] 2 2
```

data.tables and data.frames (III)

data table never automatically converts character columns to factors

```
x_df <- data.frame(id = 1:2,
                   name = c("a", "b"))
class(x_df$name)
[1] "factor"

x_dt <- data.table(id = 1:2,
                   name = c("a", "b"))
class(x_dt$name)
[1] "character"
```

data.tables and data.frames (IV)

Never sets, needs or uses *row names*

```
rownames(x_dt) <- c("R1", "R2")
x_dt

   id name
1:  1    a
2:  2    b
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!

DATA MANIPULATION IN R WITH DATA.TABLE

Filtering rows in a data.table

Matt Dowle and Arun Srinivasan
Instructors, DataCamp

General form of data.table syntax

First argument `i` is used to *subset* or *filter* rows

```
# General form of data.table syntax
DT[i, j, by]
   |  |  |
   |  |  --> grouped by what?
   |  -----> what to do?
   --------> on which rows?
```

Row numbers

```
# Subset 3rd and 4th rows from batrips
batrips[3:4]

# Same as
batrips[3:4, ]
```

```
# Subset everything except first five rows
batrips[-(1:5)]

# Same as
batrips[!(1:5)]
```

Special symbol .N

- `.N` is an integer value that contains the number of rows in the data.table
- Particularly useful alternative to `nrow(x)` in `i`

```
nrow(batrips)
[1] 326339

batrips[326339]
   trip_id duration ...
1:  588914      364 ...

# Returns the last row
batrips[.N]
   trip_id duration ...
1:  588914      364 ...

# Return all but the last 10 rows
ans <- batrips[1:(.N-10)]
nrow(ans)
[1] 326329
```

Logical expressions (I)

```
# Subset rows where subscription_type is "Subscriber"
batrips[subscription_type == "Subscriber"]

# If batrips was only a data frame
batrips[batrips$subscription_type == "Subscriber", ]
```

Logical expressions (II)

```
# Subset rows where start_terminal = 58 and end_terminal is not 65
batrips[start_terminal == 58 & end_terminal != 65]

# If batrips was only a data frame
batrips[batrips$start_terminal == 58 & batrips$end_terminal != 65]
```

Logical expressions (III)

Optimized using secondary indices for speed automatically.

```
set.seed(1)
dt <- data.table(x = sample(10000, 10e6, TRUE),
                 y = sample(letters, 1e6, TRUE))
indices(dt)
NULL

# 0.207s on first run (time to create index + subset)
system.time(dt[x == 900])
user  system elapsed
0.207   0.015   0.226

indices(dt)
[1] "x"

# 0.002s on subsequent runs (instant subset using index)
system.time(dt[x == 900])
user  system elapsed
0.002   0.000   0.002
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!

DATA MANIPULATION IN R WITH DATA.TABLE

Helpers for filtering

Matt Dowle and Arun Srinivasan
Instructors, DataCamp

%like%

- `%like%` allows you to search for a *pattern* in a *character* or a *factor* vector
  - Usage: `col %like% pattern`

```
# Subset all rows where start_station starts with San Francisco
batrips[start_station %like% "^San Francisco"]

# Instead of
batrips[grepl("^San Francisco", start_station)]
```

%between%

- `%between%` allows you to search for values in the closed interval `[val1, val2]`
  - Usage: `numeric_col %between% c(val1, val2)`

```
# Subset all rows where duration is between 2000 and 3000
batrips[duration %between% c(2000, 3000)]

# Instead of
batrips[duration >= 2000 & duration <= 3000]
```

%chin%

- `%chin%` is similar to `%in%`, but it is *much* faster and only for character vectors
  - Usage: `character_col %chin% c("val1", "val2", "val3")`

```
# Subset all rows where start_station is
# "Japantown", "Mezes Park" or "MLK Library"
batrips[start_station %chin% c("Japantown", "Mezes Park", "MLK Library")]

# much faster than
batrips[start_station %in% c("Japantown", "Mezes Park", "MLK Library")]
```

DATA MANIPULATION IN R WITH DATA.TABLE

Let's practice!