

# Introduction to data.table

**Bay Area R Users Group**

**12 May 2015**

**Matt Dowle**

# In one slide

```
DT[type=="books", sum(sales)]
```

# filter and do by group

```
DT[type=="books", sum(sales), by=country]
```

# Why inside [...] ?

- **R's lazy evaluation enables optimizations of i, j and by together**
- **Fast development**
- **Flexible - do anything in j**
- **Fast to read and maintain in production**

PRC

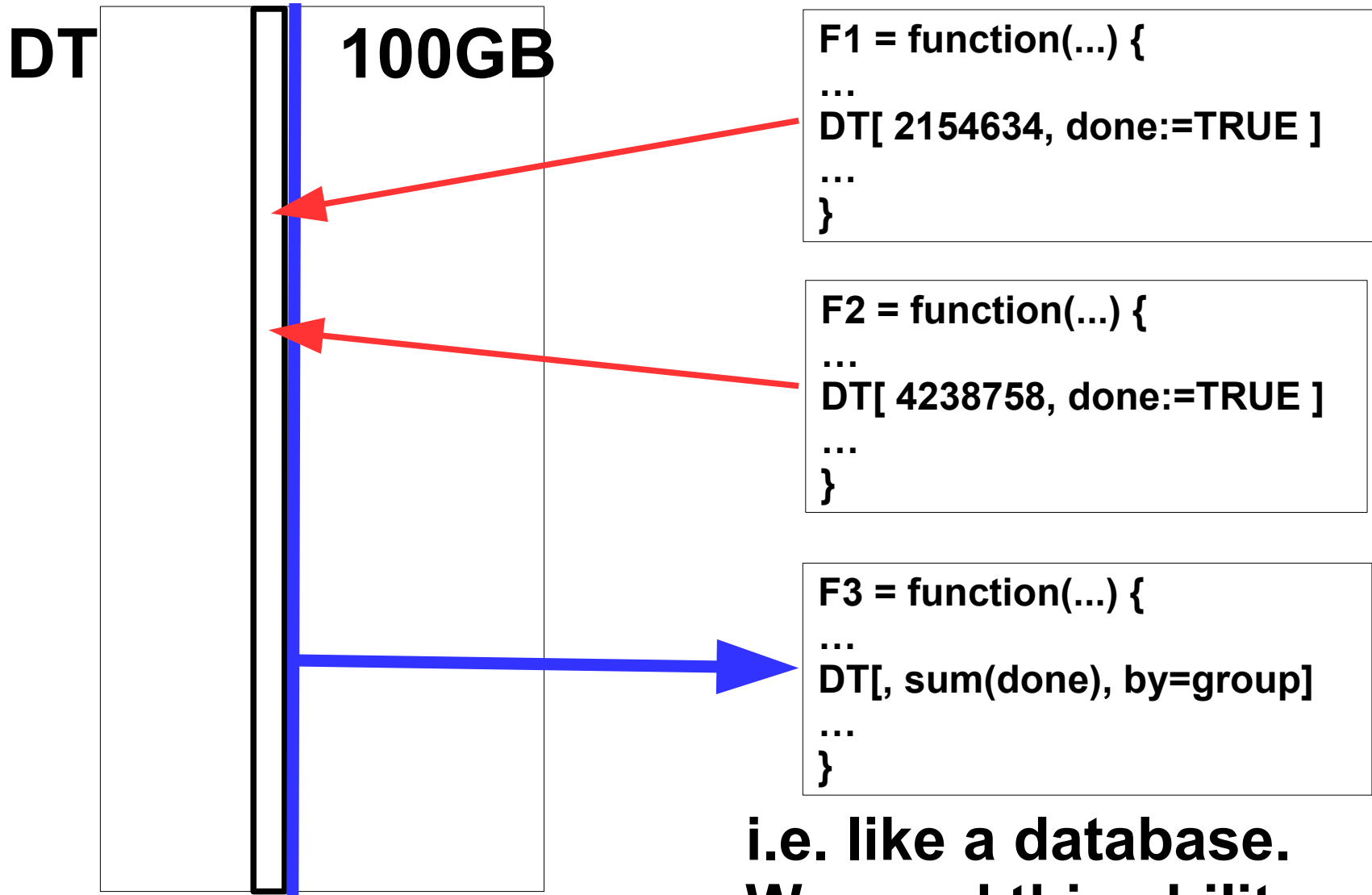
id	date	price
SBRY	20080501	380.50
SBRY	20080502	391.50
SBRY	20080506	389.00
VOD	20080501	159.30
VOD	20080502	163.30
VOD	20080506	160.80

setkey(PRC, id, date)

	<u>Result</u>
1. PRC[.("SBRY")]	rows 1:3
2. PRC[.("SBRY",20080502),price]	391.50
3. PRC[.("SBRY",20080505),price]	NA
4. PRC[.("SBRY",20080505),price,roll=TRUE]	391.50
5. PRC[.("SBRY",20080601),price,roll=TRUE]	389.00
6. PRC[.("SBRY",20080601),price,roll=TRUE,rollends=FALSE]	NA
7. PRC[.("SBRY",20080601),price,roll=20]	NA
8. PRC[.("SBRY",20080601),price,roll=40]	389.00

# Client server demo

# **:= by reference**




```
F1 = function(...) {  
...  
DT[ 2154634, done:=TRUE ]  
...  
}
```


```
F2 = function(...) {  
...  
DT[ 4238758, done:=TRUE ]  
...  
}
```


```
F3 = function(...) {  
...  
DT[, sum(done), by=group]  
...  
}
```

**i.e. like a database.  
We need this ability.**


# Input table: 1,000,000,000 rows x 9 columns ( 50 GB ) - Random order

 data.table 1.9.2 - CRAN 27 Feb 2014 - Total: \$0.08 for 15 minutes

 dplyr 0.2 - CRAN 21 May 2014 - Total: \$0.26 for 51 minutes


 pandas 0.14.1 - PyPI 11 Jul 2014 - Total: \$0.15 for 31 minutes

 First time


 Second time

Minutes    2       3       4       5       6       7       8       9       10      11      12      13      14      15

Test 1 : 100 ad hoc groups of 10,000,000 rows; result 100 x 2


 DT[, sum(v1), keyby=id1]

 DF %>% group\_by(id1) %>% summarise(sum(v1))

 DF.groupby(['id1']).agg({'v1':'sum'})

Test 2 : 10,000 ad hoc groups of 100,000 rows; result 10,000 x 3

 DT[, sum(v1), keyby='id1,id2']

 DF %>% group\_by(id1,id2) %>% summarise(sum(v1))

 DF.groupby(['id1','id2']).agg({'v1':'sum'})

To be updated



Test 3 : 10,000,000 ad hoc groups of 100 rows; result 10,000,000 x 3

DT[, list(sum(v1), mean(v3)), keyby=id3]



DF %>% group\_by(id3) %>% summarise(sum(v1), mean(v3))



DF.groupby(['id3']).agg({'v1':'sum', 'v3':'mean'})



data.table 1.9.2  
dplyr 0.2  
pandas 0.14.1

Test 4 : 100 ad hoc groups of 10,000,000 rows; result 100 x 4

DT[, lapply(.SD, mean), keyby=id4, .SDcols=7:9]



DF %>% group\_by(id4) %>% summarise\_each(funs(mean), vars=7:9)



DF.groupby(['id4']).agg({'v1':'mean', 'v2':'mean', 'v3':'mean'})



Test 5 : 10,000,000 ad hoc groups of 100 rows; result 10,000,000 x 4

DT[, lapply(.SD, sum), keyby=id6, .SDcols=7:9]



DF %>% group\_by(id6) %>% summarise\_each(funs(sum), vars=7:9)

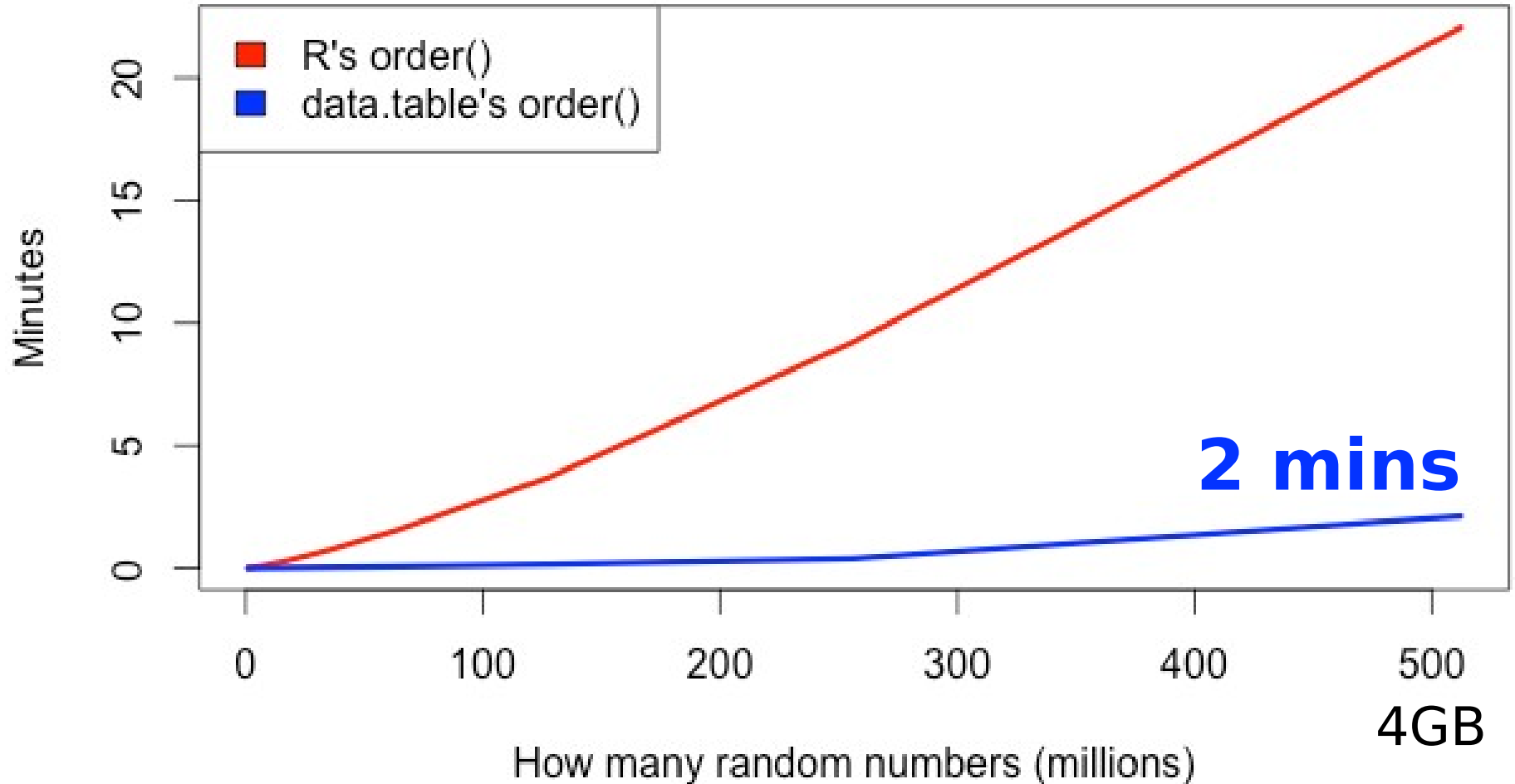


DF.groupby(['id6']).agg({'v1':'sum', 'v2':'sum', 'v3':'sum'})



Minutes 2 3 4 5 6 7 8 9 10 11 12 13 14 15

**22 mins**



MacBook Pro 2.8GHz Intel Core i7 16GB  
R 3.1.3 data.table 1.9.4

# Non-speed reasons

Two S.O. questions

data.table wiki

?data.table examples

# Thank you

<https://github.com/Rdatatable/data.table/wiki>

This presentation was recorded :

<http://livestream.com/h2oai/events/4046257>