Tutorials

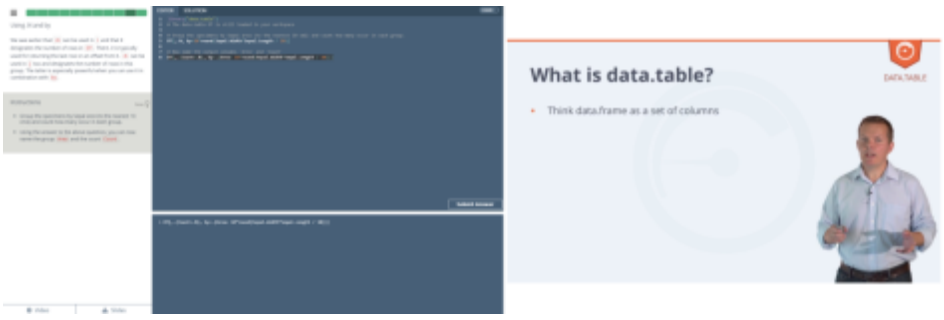
Karlijn Willems
February 3rd, 2015

R PROGRAMMING +1

A data.table R Tutorial: Intro to DT[i, j, by]

The free data.table R tutorial explains the basics and syntax of the data.table package for R. Master the data.table syntax now.

This data.table R tutorial explains the basics of the `DT[i, j, by]` command which is core to the data.table package. If you want to learn more on the data.table package, DataCamp provides an interactive R course on the data.table package. The course has more than 35 interactive R exercises - all taking place in the comfort of your own browser - and several videos with Matt Dowle, main author of the data.table package, and Arun Srinivasan, major contributor. Try if for free.



If you have already worked with large datasets in RAM (1 to more than 100GB), you know that a `data.frame` can be limiting: the time it takes to do

tutorial, the simplicity of doing complicated operations will astonish you. So you will not only be reducing computing time, but programming time as well.

The `DT[i,j,by]` command has three parts: `i`, `j` and `by`. If you think in SQL terminology, the `i` corresponds to WHERE, `j` to SELECT and `by` to GROUP BY. We talk about the command by saying "Take DT, subset the rows using '`i`', then calculate '`j`' grouped by '`by`'". So in a simple example and using the `hflights` dataset (so you can reproduce all the examples) this gives:

```
library(hflights)
library(data.table)
DT &lt;- as.data.table(hflights)
DT[Month==10,mean(na.omit(AirTime)), by=UniqueCarrier]
```



```
UniqueCarrier V1
AA            68.76471
AS            255.29032
B6            176.93548
CO            141.52861
...
```

Where we subsetted the data table to keep only the rows of the 10th Month of the year, calculated the average AirTime of the planes that actually flew (that's why `na.omit()` is used, cancelled flights don't have a value for their AirTime) and then grouped the results by their Carrier. We can see for example that AA (American Airlines) has a very short average AirTime

The i part

The 'i' part is used for subsetting on rows, just like in a data frame.

```
DT[2:5]
```

```
#selects the second to the fifth row of DT
Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum TailNu
2011 1     2          7         1401    1501    AA            428       N557AA
2011 1     3          1         1352    1502    AA            428       N541AA
2011 1     4          2         1403    1513    AA            428       N403AA
2011 1     5          3         1405    1507    AA            428       N492AA

ArrDelay DepDelay Origin Dest Distance TaxiIn TaxiOut Cancelled CancellationCo
-9       1        IAH    DFW  224      6      9       0
-8      -8        IAH    DFW  224      5      17      0
 3       3        IAH    DFW  224      9      22      0
-3       5        IAH    DFW  224      9      9       0
```

But you can also use column names, as they are evaluated in the scope of DT.

```
DT[UniqueCarrier==&quot;AA&quot;]
```

```
#Returns all those rows where the Carrier is American Airlines
```

```
2011 1     4          2          1403    1513    AA             428       N403AA
2011 1     5          3          1405    1507    AA             428       N492AA
---
2011 12    27         2          1021    1333    AA             2234      N3ETAA
2011 12    28         3          1015    1329    AA             2234      N3FJAA
2011 12    29         4          1023    1335    AA             2234      N3GSAA
2011 12    30         5          1024    1334    AA             2234      N3BAAA
2011 12    31         6          1024    1343    AA             2234      N3HNAA
AirTime ArrDelay DepDelay Origin Dest Distance TaxiIn TaxiOut Cancelled Cancel
40      -10       0       IAH    DFW  224       7     13      0
45       -9       1       IAH    DFW  224       6      9      0
48       -8      -8       IAH    DFW  224       5     17      0
39        3       3       IAH    DFW  224       9     22      0
44       -3       5       IAH    DFW  224       9      9      0
---
112     -12       1       IAH    MIA  964       8     12      0
112     -16      -5       IAH    MIA  964       9     13      0
110     -10       3       IAH    MIA  964      12     10      0
110     -11       4       IAH    MIA  964       9     11      0
119      -2       4       IAH    MIA  964       8     12      0
```

Notice that you don't have to use a comma for subsetting rows in a data table. In a data.frame doing this `DF[2:5]` would give all the rows of the 2nd to 5th column. Instead (as everyone reading this obviously knows), we have to specify `DF[2:5,]`. Also notice that `DT[,2:5]` does not mean anything for data tables, as is explained in the first question of the FAQs of the data.table package.

Quirky and useful: when subsetting rows you can also use the symbol `.N` in the DT[...] command, which is the number of rows or the last row. You can use it for selecting the last row or an offset from it.

```
#Returns the penultimate row of DT
Year Month DayofMonth DayOfWeek DepTime ArrTime UniqueCarrier FlightNum TailNu
2011 12    6          2         656     812     WN            621       N727SW
ArrDelay DepDelay Origin Dest Distance TaxiIn TaxiOut Cancelled CancellationCo
-13      -4       HOU    TUL  453      3      9       0
```

The j part

The 'j' part is used to select columns and do *stuff* with them. And *stuff* can really mean anything. All kinds of functions can be used, which is a strong point of the data.table package.

```
DT[, mean(na.omit(ArrDelay))]
```

```
[1] 7.094334
```

Notice that the 'i' part is left blank, and the first thing in the brackets is a comma. This might seem counterintuitive at first. However, this simply means that we do not subset on any rows, so all rows are selected. In the 'j' part, the average delay on arrival of all flights is calculated. It appears that the average plane of the hflights dataset had more than 7 minutes delay. Be prepared when catching your next flight!

vector, as shown above.

```
DT[, .(mean(na.omit(DepDelay)), mean(na.omit(ArrDelay)))]
```

```
V1       V2
9.444951 7.094334
```

Another useful feature which requires the '`.()`' notation allows you to rename columns inside the DT[...] command.

```
DT[, .(Avg_ArrDelay =
mean(na.omit(ArrDelay)))]
```

```
Avg_ArrDelay
7.094334
```

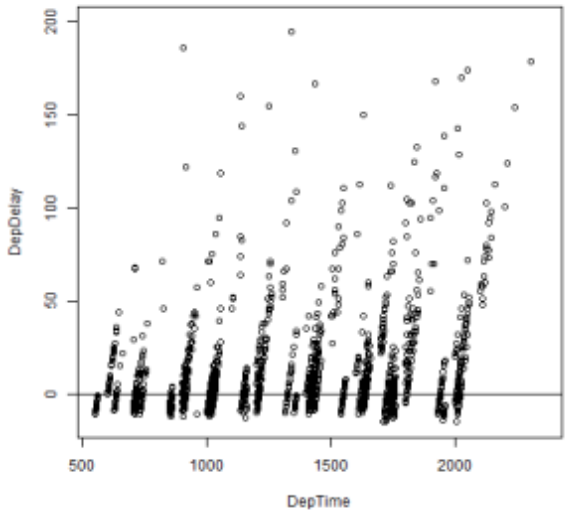
```
DT[, .(Avg_DepDelay = mean(na.omit(DepDelay)),
avg_ArrDelay = mean(na.omit(ArrDelay)))]
```

```
Avg_DepDelay Avg_ArrDelay
9.444951     7.094334
```

Combining the above about ‘i’ and ‘j’ gives:

```
DT[UniqueCarrier==&quot;AA&quot;, .(Avg_DepDelay =
mean(na.omit(DepDelay)),
Avg_ArrDelay = mean(na.omit(ArrDelay)),
plot(DepTime,DepDelay,ylim=c(-15,200)),
abline(h=0))]
```

```
Avg_DepDelay Avg_ArrDelay V3    V4
6.390144     0.8917558    NULL  NULL
```



Here we took DT, selected all rows where the carrier was AA in the 'i' part, calculated the average delay on departure and on arrival, and plotted the time of departure against the delay on departure in the 'j' part.

This significantly shortens your programming time.

The by part

The final section of this data.table R tutorial focuses on the 'by' part. The 'by' part is used when we want to calculate the 'j' part grouped by a specific variable (or a manipulation of that variable). You will see that the 'j' expression is repeated for each 'by' group. It is simple to use: you just specify the column you want to group by in the 'by' argument.

```
DT[,mean(na.omit(DepDelay)),by=Origin]
```

```
Origin  V1
IAH     8.436951
HOU    12.837873
```

Here, we calculated the average delay before departure, but grouped by where the plane is coming from. It seems that flights departing from HOU have a larger average delay than those leaving from IAH.

Just as with the 'j' part, you can do a lot of *stuff* in the 'by' part. Functions can be used in the 'by' part so that results of the operations done in the 'j' part are grouped by something we specified in the DT[...] command. Using functions inside `DT[…]` makes that one line very powerful. Likewise, the '`.()`' notation needs to be used when using several columns in the 'by' part.

```
Origin Weekdays Avg_DepDelay_byWeekdays
IAH    FALSE     8.286543
IAH    TRUE      8.492484
HOU    FALSE    10.965384
HOU    TRUE     13.433994
```

Here, the average delay before departure of all planes (no subsetting in the 'i' part, so all rows are selected) was calculated first, and grouped secondly, first by origin of the plane and then by weekday. Weekdays is `False` in the weekends. It appears that the average delay before departure was larger when the plane left from HOU than from IAH, and surprisingly the delays were smaller in the weekends.

Putting it all together a typical `DT[i,j,by]` command gives:

```
DT[UniqueCarrier==&quot;DL&quot;, .(Avg_DepDelay =
mean(na.omit(DepDelay)),
Avg_ArrDelay = mean(na.omit(ArrDelay)),
Compensation = mean(na.omit(ArrDelay - DepDelay))), by = .(Origin, Weekdays =
```

```
Origin Weekdays Avg_DepDelay Avg_ArrDelay Compensation
IAH    FALSE     8.979730     4.116751    -4.825719
HOU    FALSE     7.120000     2.656566    -4.555556
IAH    TRUE      9.270948     6.281941    -2.836609
HOU    TRUE     11.631387    10.406593    -1.278388
```

compensated in air was also calculated (in 'j'). It appears that in the weekends, irrespective of the plane was coming from IAH or HOU, the time compensated while in air (thus by flying faster) is bigger.

There is much more to discover in the data table package, but this post illustrated the basic `DT[i,j,by]` command. The DataCamp course explains the whole data table package extensively. You can do the exercises at your own pace in your browser while getting hints and feedback, and review the videos and slides as much as you want. This interactive way of learning allows you to gain profound knowledge and practical experience with data tables. Try it for free.

Hopefully you know understand thanks to this data.table R tutorial the fundamental syntax of data.table, and are you ready to experiment yourself. If you have questions concerning the data.table package, have a look here. Matt and Arun are very active. One of the next blogposts on the data.table package will be more technical, zooming in on the wide possibilities with data tables. Stay tuned!