Data Transformation with data.table : : CHEAT SHEET

Basics

data.table is an extremely fast and memory efficient package for transforming data in R. It works by converting R's native data frame objects into data.tables with new and enhanced functionality. The basics of working with data.tables are:

dt[i, j, by]

Take data.table **dt**,
subset rows using **i**,
and manipulate columns with **j**,
grouped according to **by**.

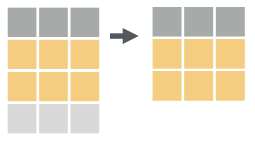
data.tables are also data frames – functions that work with data frames therefore also work with data.tables.

Create a data.table

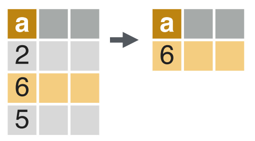
data.table(a = c(1, 2), b = c("a", "b")**)** – create a data.table from scratch. Analogous to data.frame().

setDT(df**)*** or **as.data.table(**df**)** – convert a data frame or a list to a data.table.

Subset rows using i



dt[**1:2**,] – subset rows based on row numbers.



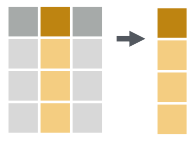
dt[**a > 5**,] – subset rows based on the values in one or more columns.

LOGICAL OPERATORS TO USE IN i

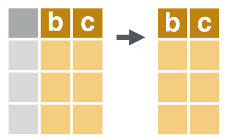
<	<=	is.na()	%in%	|	**%like%**
>	>=	!is.na()	!	&	**%between%**

Manipulate columns with j

EXTRACT

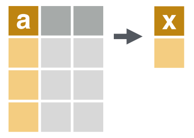


dt[, **c(2)**] – select column(s) by number. Prefix column numbers with "-" to deselect.



dt[, **.(b, c)**] – select column(s) by name.

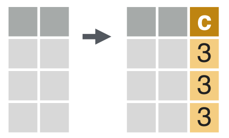
SUMMARIZE



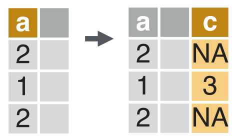
dt[, **.(x = sum(a))**] – create a data.table with new columns based on the summarized values of rows.

Summary functions such as mean(), median(), min(), max(), etc. may be used to summarize rows.

ADD COLUMN

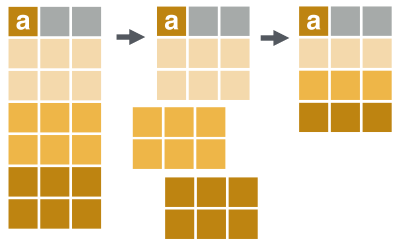


dt[, **c := 1 + 2**] – create a new column based on an expression.



dt[**a == 1**, **c := 1 + 2**] – create a new column based on an expression but only for a subset of rows.

Group according to by



dt[, j, **by = .(a)**] – group rows by values in specified column(s).

dt[, j, **keyby = .(a)**] – group *and simultaneously sort* rows according to values in specified column(s).

COMMON GROUPED OPERATIONS

dt[, **.(c = sum(b))**, **by = .(a)**] – summarize rows within groups.

dt[, **c := sum(b)**, **by = .(a)**] – create a new column and compute rows within groups.

dt[, **.SD[1]**, **by = .(a)**] – extract first row of groups.

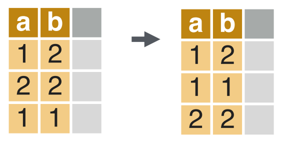
dt[, **.SD[.N]**, **by = .(a)**] – extract last row of groups.

Chaining

dt[…][…] – perform a sequence of data.table operations by *chaining* multiple "[]".

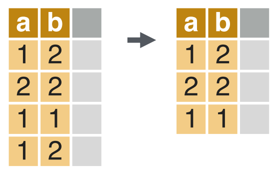
Functions for data.tables

ARRANGE ROWS



setorder(dt, a, -b**)** – arrange the rows of a data.table. Prefix variable names with "-" for descending order.

UNIQUE CASES



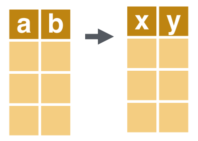
unique(dt, a, b**)** – extract a subset of the data based on a unique combination of values.

uniqueN(dt, by = c("a", "b")**)** – return the number of unique rows, based on columns specified in "by". Leave out "by" to use all columns.

* SET FUNCTIONS

data.table provides a collection of functions beginning with "set". They work without "<-" to alter data.tables in place. For instance, "**setDT(**dt**)**" works like "dt <- **as.data.table(**dt**)**" but without creating any copies in memory.

RENAME COLUMNS



setnames(dt, c(“a”, “b”), c(“x”, “y”)**)** – rename multiple columns.

SET KEYS

setkey(dt, a, b**)** – set keys in a data.table to enable faster repeated lookups in specified columns using “dt[.(value),]” or for merging without specifying merging columns “dt_a[dt_b]”.

Combine data.tables

JOIN

a	b
1	c
2	a
3	b

+

x	y
3	b
2	c
1	a

=

a	b	x
3	b	3
1	c	2
2	a	1

dt_a[dt_b, **on = .(b = y)**] – join two data.tables based on rows with equal values. setkey() can be used in stead of “.on”.

a	b	c
1	c	7
2	a	5
3	b	6

+

x	y	z
3	b	4
2	c	5
1	a	8

=

a	b	c	x
3	b	4	3
1	c	5	2
NA	a	8	1

dt_a[dt_b, **on = .(b = y, c > z)**] – join two data.tables based on rows with equal and unequal values

ROLLING JOIN

By default, a rolling join matches rows, defined by an id variable, but only keeps the most recent preceding match with the left table, defined by a date variable.

a	id	date
1	A	01-01-2010
2	A	01-01-2012
3	A	01-01-2014
1	B	01-01-2010
2	B	01-01-2012

+

b	id	date
1	A	01-01-2013
1	B	01-01-2013

=

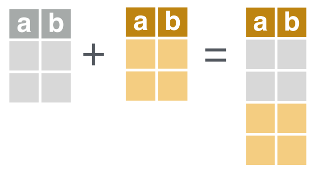
a	id	date	b
2	A	01-01-2013	1
2	B	01-01-2013	1

```
# first set keys
setkey(dt_a, id, date)
setkey(dt_b, id, date)
```

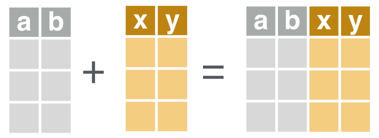
```
# then roll
dt_a[dt_b, roll = TRUE]
```

dt_a[dt_b, **roll = -Inf]** – reverse the direction of the rolling join.

BIND



rbind(dt_a, dt_b**)** – combine rows of two data.tables



cbind(dt_a, dt_b**)** – combine columns of two data.tables

Reshape a data.table

RESHAPE TO WIDE FORMAT

id	y	a	b
A	X	1	3
A	Z	2	4
B	X	1	3
B	Z	2	4

→

id	a_X	a_Z	b_X	b_Z
A	1	2	3	4
B	1	2	3	4

```
dcast(dt,
      id ~ y,
      value.var = c(“a”, “b”))
```

Reshape a data.table from long to wide format.

dt	A data.table.
id ~ y	Formula with a LHS: id column(s) containing id(s) for multiple entries. And a RHS: column(s) with value(s) to spread in column headers.
value.var	Column(s) containing values to fill into cells.

RESHAPE TO LONG FORMAT

id	a_X	a_Z	b_X	b_Z
A	1	2	3	4
B	1	2	3	4

→

id	y	a	b
A	1	1	3
B	1	1	3
A	2	2	4
B	2	2	4

```
melt(dt,
     id.vars = c("id"),
     measure = patterns("^a", "^b"),
     variable.name = "y",
     value.name = c("a", "b"))
```

Reshape a data.table from wide to long format.

dt	A data.table.
id.vars	Id column(s) with id(s) for multiple entries.
measure	Column(s) containing values to fill into cells (often in pattern form).
variable.name, value.name	Name(s) of new column(s) for variables and values derived from old headers.

.SD

Refer to a **S**ubset of the **D**ata within a data.table with **.SD**.

MULTIPLE COLUMN TYPE CONVERSION

dt[, **lapply(.SD, as.character), .SDcols = c(“a”, “b”)**] – convert designated columns to character

GROUP OPTIMA

dt[, **.SD[which.max(a)], by = b**] – select the row with the highest value of within a column grouped according to b. Also works with which.min() and which(). Similar to .SD[.N] and .SD[1] on page 1.

fread & fwrite

fread & fwrite are data.table’s fast and multithreaded functions for importing from and exporting to flat files – such as csv and tsv.

IMPORT

fread(“file.csv”**)** – read a flat file into R.

fread(“file.csv”, cols = c(“a”, “b”)**)** – read two columns named “a” and “b” from a file named “file.csv” in the working directory.

EXPORT

fwrite(dt, file =“”**)** – write a flat file from R.

MULTITHREADING

setDTthreads() – set the number of threads that fread may use. Default is all available and appropriate for the task at hand.