Fast data reading with fread()

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

Blazing FAST!

- Fast and parallel file reader
- Argument `nThread` controls the number of threads to use

User-friendly

- Can import local files, files from the web, and strings
- Intelligent defaults - `colClasses` , `sep` , `nrows` etc.
- *Note: Dates and Datetimes are read as character columns but can be converted later with the excellent* `fasttime` *or* `anytime` *packages*

Fast and friendly file reader

```
# File from URL
DT1<-fread("https://bit.ly/2RkBXhV")
DT1
```

```
a b
1 2
3 4
```

```
# Local file
DT2 <- fread("data.csv")
DT2
```

```
a b
1 2
3 4
```

```
# String
DT3 <- fread("a,b\n1,2\n3,4")
DT3
```

```
a b
1 2
3 4
```

```
# String without col names
DT4 <- fread("1,2\n3,4")
DT4
```

```
V1 V2
1  2
3  4
```

nrows and skip arguments

```r
# Read only first line (after header)
fread("a,b\n1,2\n3,4", nrows = 1)
```

```
a b
1 2
```

```r
# Skip first two lines containing metadata
str <- "# Metadata\nTimestamp: 2018-05-01 19:44:28 GMT\na,b\n1,2\n3,4"
fread(str, skip = 2)
```

```
a b
1 2
3 4
```

More on nrows and skip arguments

```
str <- "# Metadata\nTimestamp: 2018-05-01 19:44:28 GMT\na,b\n1,2\n3,4"
fread(str, skip = "a,b")
```

```
a b
1 2
3 4
```

```
fread(str, skip = "a,b", nrows = 1)
```

```
a b
1 2
```

select and drop arguments

```r
str <- "a,b,c\n1,2,x\n3,4,y"
fread(str, select = c("a", "c"))


# Same as
fread(str, drop = "b")
```

```
a c
1 x
3 y
```

```r
str <- "1,2,x\n3,4,y"
fread(str, select = c(1, 3))


# Same as
fread(str, drop = 2)
```

```
V1 V3
 1  x
 3  y
```

Advanced file reading

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

Reading big integers using integer64 type

- By default, R can only represent numbers less than or equal to 2^31 - 1 = 2147483647
- Large integers are automatically read in as `integer64` type, provided by the `bit64` package

```
ans <- fread("id,name\n1234567890123,Jane\n5284782381811,John\n")
ans
```

```
           id name
1234567890123 Jane
5284782381811 John
```

```
class(ans$id)
```

```
"integer64"
```

Specifying column class types with colClasses

```r
str <- "x1,x2,x3,x4,x5\n1,2,1.5,true,cc\n3,4,2.5,false,ff"
ans <- fread(str, colClasses = c(x5 = "factor"))
str(ans)
```

```
Classes ‘data.table’ and 'data.frame':    2 obs. of  5 variables:
 $ x1: int  1 3
 $ x2: int  2 4
 $ x3: num  1.5 2.5
 $ x4: logi  TRUE FALSE
 $ x5: Factor w/ 2 levels "cc","ff": 1 2
```

Specifying column class types with colClasses

```r
ans <- fread(str, colClasses = c("integer", "integer",
                                 "numeric", "logical", "factor"))
str(ans)
```

```
Classes ‘data.table’ and 'data.frame':    2 obs. of  5 variables:
 $ x1: int  1 3
 $ x2: int  2 4
 $ x3: num  1.5 2.5
 $ x4: logi  TRUE FALSE
 $ x5: Factor w/ 2 levels "cc","ff": 1 2
```

Specifying column class types with colClasses

```r
str <- "x1,x2,x3,x4,x5,x6\n1,2,1.5,2.5,aa,bb\n3,4,5.5,6.5,cc,dd"
ans <- fread(str, colClasses = list(numeric = 1:4, factor = c("x5", "x6")))
str(ans)
```

```
Classes ‘data.table’ and 'data.frame': 2 obs. of 6 variables:
 $ x1: num  1 3
 $ x2: num  2 4
 $ x3: num  1.5 5.5
 $ x4: num  2.5 6.5
 $ x5: Factor w/ 2 levels "aa","cc": 1 2
 $ x6: Factor w/ 2 levels "bb","dd": 1 2
```

The fill argument

```r
str <- "1,2\n3,4,a\n5,6\n7,8,b"
fread(str)
```

```
V1 5 6
 7 8 b
Warning message:
In fread(str) :
  Detected 2 column names but the data has 3 columns (i.e. invalid file).
  Added 1 extra default column name for the first column which is guessed to
  be row names or an index.
  Use setnames() afterwards if this guess is not correct,
  or fix the file write command that created the file to create a valid file.
```

The fill argument

```
fread(str, fill = TRUE)
```

```
V1 V2 V3
 1  2
 3  4  a
 5  6
 7  8  b
```

The na.strings argument

Missing values are commonly encoded as: `"999"` or `"##NA"` or `"N/A"`

```r
str <- "x,y,z\n1,###,3\n2,4,###\n#N/A,7,9"
ans <- fread(str, na.strings = c("###", "#N/A"))
ans
```

```
x   y   z
1  NA   3
2   4  NA
NA  7   9
```

Fast data writing with fwrite()

DATA MANIPULATION WITH DATA.TABLE IN R

Matt Dowle, Arun Srinivasan
Instructors, DataCamp

fwrite

Ability to write `list` columns using secondary separator (`|`)

```
dt <- data.table(id = c("x", "y", "z"), val = list(1:2, 3:4, 5:6))
fwrite(dt, "fwrite.csv")
fread("fwrite.csv")
```

```
id  val
 x  1|2
 y  3|4
 z  5|6
```

date and datetime columns (ISO)

- `fwrite()` provides three additional ways of writing date and datetime format - `ISO` , `squash` and `epoch`
- Encourages the use of ISO standards with `ISO` as default

Date and times

```
now <- Sys.time()
dt  <- data.table(date = as.IDate(now),
                  time = as.ITime(now),
                  datetime = now)
dt
```

```
      date     time            datetime
2018-12-17 19:54:51   2018-12-17 14:54:51
```

date and datetime columns (ISO)

```r
# "ISO" is default
fwrite(dt, "datetime.csv", dateTimeAs = "ISO")


fread("datetime.csv")
```

```
      date       time                    datetime
2018-12-17   19:55:39  2018-12-17T19:55:39.735036Z
```

date and datetime columns (Squash)

- `squash` writes `yyyy-mm-dd hh:mm:ss` as `yyyymmddhhmmss` , for example
- Read in as integer. Very useful to extract month, year etc by simply using modulo arithmetic. e.g., `20160912 %/% 10000 = 2016`
- Also handles milliseconds (ms) resolution
- POSIXct type (17 digits with ms resolution) is automatically read in as `integer64` by `fread`

date and datetime columns (Squash)

```
fwrite(dt, "datetime.csv", dateTimeAs = "squash")

fread("datetime.csv")

       date   time          datetime
1: 20181217 195539 20181217195539735
```

```
20181217 %/% 10000
[1] 2018
```

date and datetime columns (Epoch)

- `epoch` counts the number of `days` (for dates) or seconds (for time and datetime) since relevant epoch
- Relevant epoch is `1970-01-01` , `00:00:00` and `1970-01-01T00:00:00Z` for `date` , `time` and `datetime` , respectively

date and datetime columns (Epoch)

```
fwrite(dt, "datetime.csv", dateTimeAs = "epoch")
fread("datetime.csv")
```

```
 date  time   datetime
17882 71871 1545076672
```