Aquisição de Arquivos e Datasets

Lendo arquivos JSON

Delermando Branquinho Filho

Tabela de dados

- Herdado de data.frame
- Todas as funções que aceitam data.frame trabalho em data.table
- Escrito em C por isso é muito mais rápido
- Muito, muito mais rápido em subconjunto, grupo e atualização

---

Criando tabelas como data frame

```
library(data.table)
DF = data.frame(x=rnorm(9),y=rep(c("a","b","c"),each=3),z=rnorm(9))
head(DF,3)
```

```
##            x y           z
## 1  0.30203919 a -0.07060748
## 2 -0.06565024 a  0.15054630
## 3  0.15461701 a  0.62923766
```

```
DT = data.table(x=rnorm(9),y=rep(c("a","b","c"),each=3),z=rnorm(9))
head(DT,3)
```

```
##            x y          z
## 1:  0.5259934 a -1.1960890
## 2:  1.1947834 a -1.2669223
## 3: -0.3637918 a -0.9590022
```

Ver todas as tabelas de dados na memória

```
tables()
```

```
##      NAME NROW NCOL MB COLS  KEY
## [1,] DT      9    3  1 x,y,z
## Total: 1MB
```

Subsetting em Linhas

```
DT[2,]
```

```
##           x y         z
## 1: 1.194783 a -1.266922
```

```
DT[DT$y=="a",]
```

```
##             x y          z
## 1:  0.5259934 a -1.1960890
## 2:  1.1947834 a -1.2669223
## 3: -0.3637918 a -0.9590022
```

```
DT[c(2,3)]
```

```
##             x y          z
## 1:  1.1947834 a -1.2669223
## 2: -0.3637918 a -0.9590022
```

Subsettingem colunas!?

```
DT[,c(2,3)]
```

```
##    y           z
## 1: a -1.196089018
## 2: a -1.266922251
## 3: a -0.959002182
## 4: b -0.103856306
## 5: b  0.248349845
## 6: b -0.374149834
## 7: c -0.090469100
## 8: c -0.001735052
## 9: c  1.692447396
```

Subconjunto de coluna na tabela de dados

- A função de subconjunto é modificada para data.table
- O argumento que você passar após a vírgula é chamado de "expressão"
- Em R uma expressão é uma coleção de declarações encerradas em curley parênteses

```
{
  x = 1
  y = 2
}
k = {print(10); 5}
```

```
## [1] 10
```

```
print(k)
```

```
## [1] 5
```

Cálculo de valores para variáveis com expressões

```
DT[,list(mean(x),sum(z))]
```

```
##          V1        V2
## 1: -0.33994 -2.051427
```

```
DT[,table(y)]
```

```
## y
## a b c
## 3 3 3
```

Adicionar novas colunas

```
DT[,w:=z^2]
```

```
##             x y           z           w
## 1:  0.5259934 a -1.196089018 1.430629e+00
## 2:  1.1947834 a -1.266922251 1.605092e+00
## 3: -0.3637918 a -0.959002182 9.196852e-01
## 4: -1.1743749 b -0.103856306 1.078613e-02
## 5: -2.1894528 b  0.248349845 6.167765e-02
## 6: -1.4653443 b -0.374149834 1.399881e-01
## 7: -1.1904403 c -0.090469100 8.184658e-03
## 8:  1.0557090 c -0.001735052 3.010406e-06
## 9:  0.5474583 c  1.692447396 2.864378e+00
```

```
DT2 <- DT
DT[, y:= 2]
```

```
## Warning in `[.data.table`(DT, , `:=`(y, 2)): Coerced 'double' RHS to
## 'character' to match the column's type; may have truncated precision.
## Either change the target column to 'double' first (by creating a new
## 'double' vector length 9 (nrows of entire table) and assign that; i.e.
## 'replace' column), or coerce RHS to 'character' (e.g. 1L, NA_[real|
## integer]_, as.*, etc) to make your intent clear and for speed. Or, set the
## column type correctly up front when you create the table and stick to it,
## please.
```

```
##             x y           z           w
## 1:  0.5259934 2 -1.196089018 1.430629e+00
## 2:  1.1947834 2 -1.266922251 1.605092e+00
## 3: -0.3637918 2 -0.959002182 9.196852e-01
## 4: -1.1743749 2 -0.103856306 1.078613e-02
## 5: -2.1894528 2  0.248349845 6.167765e-02
## 6: -1.4653443 2 -0.374149834 1.399881e-01
## 7: -1.1904403 2 -0.090469100 8.184658e-03
## 8:  1.0557090 2 -0.001735052 3.010406e-06
## 9:  0.5474583 2  1.692447396 2.864378e+00
```

Cuidado

```
head(DT,n=3)
```

```
##             x y          z         w
## 1:  0.5259934 2 -1.1960890 1.4306289
## 2:  1.1947834 2 -1.2669223 1.6050920
## 3: -0.3637918 2 -0.9590022 0.9196852
```

```
head(DT2,n=3)
```

```
##             x y          z         w
## 1:  0.5259934 2 -1.1960890 1.4306289
## 2:  1.1947834 2 -1.2669223 1.6050920
## 3: -0.3637918 2 -0.9590022 0.9196852
```

Múltiplas operações

```
DT[,m:= {tmp <- (x+z); log2(tmp+5)}]
```

```
##             x y           z            w        m
## 1:  0.5259934 2 -1.196089018 1.430629e+00 2.114335
## 2:  1.1947834 2 -1.266922251 1.605092e+00 2.300962
## 3: -0.3637918 2 -0.959002182 9.196852e-01 1.878610
## 4: -1.1743749 2 -0.103856306 1.078613e-02 1.895988
## 5: -2.1894528 2  0.248349845 6.167765e-02 1.613012
## 6: -1.4653443 2 -0.374149834 1.399881e-01 1.660155
## 7: -1.1904403 2 -0.090469100 8.184658e-03 1.894950
## 8:  1.0557090 2 -0.001735052 3.010406e-06 2.597882
## 9:  0.5474583 2  1.692447396 2.864378e+00 2.855971
```

---

plyr likeComo operações

```
DT[,a:=x>0]
```

```
##             x y           z            w        m     a
## 1:  0.5259934 2 -1.196089018 1.430629e+00 2.114335  TRUE
## 2:  1.1947834 2 -1.266922251 1.605092e+00 2.300962  TRUE
## 3: -0.3637918 2 -0.959002182 9.196852e-01 1.878610 FALSE
## 4: -1.1743749 2 -0.103856306 1.078613e-02 1.895988 FALSE
## 5: -2.1894528 2  0.248349845 6.167765e-02 1.613012 FALSE
## 6: -1.4653443 2 -0.374149834 1.399881e-01 1.660155 FALSE
## 7: -1.1904403 2 -0.090469100 8.184658e-03 1.894950 FALSE
## 8:  1.0557090 2 -0.001735052 3.010406e-06 2.597882  TRUE
## 9:  0.5474583 2  1.692447396 2.864378e+00 2.855971  TRUE
```

```
DT[,b:= mean(x+w),by=a]
```

```
##             x y           z            w        m     a         b
## 1:  0.5259934 2 -1.196089018 1.430629e+00 2.114335  TRUE  2.306012
```

```
## 2:  1.1947834 2 -1.266922251 1.605092e+00 2.300962  TRUE  2.306012
## 3: -0.3637918 2 -0.959002182 9.196852e-01 1.878610 FALSE -1.048616
## 4: -1.1743749 2 -0.103856306 1.078613e-02 1.895988 FALSE -1.048616
## 5: -2.1894528 2  0.248349845 6.167765e-02 1.613012 FALSE -1.048616
## 6: -1.4653443 2 -0.374149834 1.399881e-01 1.660155 FALSE -1.048616
## 7: -1.1904403 2 -0.090469100 8.184658e-03 1.894950 FALSE -1.048616
## 8:  1.0557090 2 -0.001735052 3.010406e-06 2.597882  TRUE  2.306012
## 9:  0.5474583 2  1.692447396 2.864378e+00 2.855971  TRUE  2.306012
```

Variáveis especiais

`.N` An integer, length 1, containing the number of elements of a factor level

```
set.seed(123);
DT <- data.table(x=sample(letters[1:3], 1E5, TRUE))
DT[, .N, by=x]
```

```
##    x     N
## 1: a 33387
## 2: c 33201
## 3: b 33412
```

Chaves

```
DT <- data.table(x=rep(c("a","b","c"),each=100), y=rnorm(300))
setkey(DT, x)
DT['a']
```

```
##      x           y
##   1: a  0.25958973
##   2: a  0.91751072
##   3: a -0.72231834
##   4: a -0.80828402
##   5: a -0.14135202
##   6: a  2.25701345
##   7: a -2.37955015
##   8: a -0.45425393
##   9: a -0.06007418
##  10: a  0.86090061
##  11: a -1.78466393
##  12: a -0.13074225
##  13: a -0.36983749
##  14: a -0.18065990
##  15: a -1.04973030
##  16: a  0.37831550
##  17: a -1.37079353
##  18: a -0.31611578
##  19: a  0.39435003
##  20: a -1.68987831
##  21: a -1.46233527
```

```
##  22: a  2.55837664
##  23: a  0.08788697
##  24: a  1.73141492
##  25: a  1.21512638
##  26: a  0.29954390
##  27: a -0.17245754
##  28: a  1.13249663
##  29: a  0.02319828
##  30: a  1.33587399
##  31: a -1.09879007
##  32: a -0.58176064
##  33: a  0.03892452
##  34: a  1.07315441
##  35: a  1.34969593
##  36: a  1.19527937
##  37: a -0.02217912
##  38: a  0.69849448
##  39: a  0.67240626
##  40: a -0.79164585
##  41: a -0.21790545
##  42: a  0.02307037
##  43: a  0.11539395
##  44: a -0.27708029
##  45: a  0.03688377
##  46: a  0.47520014
##  47: a  1.70748924
##  48: a  1.07600560
##  49: a -1.34571320
##  50: a -1.44024891
##  51: a -0.39392783
##  52: a  0.58106297
##  53: a -0.17078819
##  54: a -0.90585446
##  55: a  0.15621346
##  56: a -0.37322530
##  57: a -0.34587104
##  58: a -0.35828720
##  59: a -0.13306601
##  60: a -0.08959642
##  61: a  0.62793032
##  62: a -1.42882873
##  63: a  0.17255399
##  64: a -0.79115025
##  65: a  1.26204078
##  66: a -0.26940548
##  67: a  0.15698296
##  68: a -0.76059823
##  69: a  1.37060069
##  70: a  0.03758155
##  71: a  0.44949417
##  72: a  2.78868764
##  73: a -0.46848614
##  74: a  1.01260608
##  75: a -0.04374086
```

```
##  76: a  1.40669725
##  77: a  0.41992874
##  78: a  0.31008615
##  79: a  1.11904687
##  80: a -1.29814018
##  81: a -1.28248182
##  82: a  1.65942788
##  83: a  0.78374544
##  84: a  0.57771022
##  85: a -0.26724640
##  86: a -0.64569141
##  87: a -0.44952912
##  88: a -0.82619821
##  89: a  1.05503854
##  90: a -0.87926983
##  91: a -1.27712832
##  92: a -0.63412243
##  93: a  0.66470047
##  94: a -0.50958183
##  95: a  0.40736335
##  96: a  1.67774776
##  97: a -1.05205570
##  98: a -0.63690737
##  99: a  0.56539163
## 100: a  0.38015779
##      x           y
```

Junta tudo

```
DT1 <- data.table(x=c('a', 'a', 'b', 'dt1'), y=1:4)
DT2 <- data.table(x=c('a', 'b', 'dt2'), z=5:7)
setkey(DT1, x); setkey(DT2, x)
merge(DT1, DT2)
```

```
##    x y z
## 1: a 1 5
## 2: a 2 5
## 3: b 3 6
```

Leitura rápida

```
big_df <- data.frame(x=rnorm(1E6), y=rnorm(1E6))
file <- tempfile()
write.table(big_df, file=file, row.names=FALSE, col.names=TRUE, sep="\t", quote=FALSE)
system.time(fread(file))
```

```
##
Read 94.0% of 1000000 rows
Read 1000000 rows and 2 (of 2) columns from 0.035 GB file in 00:00:03
```

```
##    user  system elapsed
##    2.07    0.08    2.30
```

```
system.time(read.table(file, header=TRUE, sep="\t"))
```

```
##    user  system elapsed
##    6.86    0.22    7.17
```