

Introdução a Linguagem R

Lendo Dados - continuação ...

Delermundo Branquinho Filho

Formatos Textuais

- `dumping` e `dputing` são úteis porque o formato textual resultante é editável, e no caso de corrupção, potencialmente recuperável.
 - Ao contrário escrevendo uma tabela ou um arquivo csv, `dump` e `dput` preservam o *metadata* (sacrificando alguma legibilidade), para que outro usuário não precise especificar tudo de novo.
 - ‘Textual’ formatos podem trabalhar muito melhor com programas de controle de versão como o `subversion` ou `git` que só pode acompanhar as mudanças de forma significativa em arquivos de texto
 - Os formatos textuais podem ser mais duradouros; Se houver corrupção em algum lugar no arquivo, pode ser mais fácil corrigir o problema
 - Formatos textuais aderem à “filosofia Unix”
 - Desvantagem: O formato não é muito eficiente em termos de espaço
-

dput-ting R Objects

Outra maneira de passar os dados em torno é deparsing o objeto R com `dput` e lê-lo de volta usando `dget`.

```
> y <- data.frame(a = 1, b = "a")
> dput(y)
structure(list(a = 1,
               b = structure(1L, .Label = "a",
                             class = "factor")),
          .Names = c("a", "b"), row.names = c(NA, -1L),
          class = "data.frame")
> dput(y, file = "y.R")
> new.y <- dget("y.R")
> new.y
  a b
1 1 a
```

Dumping R Objects

Vários objetos podem ser separados usando a função `dump` e lidos novamente usando `source`.

```
> x <- "foo"
> y <- data.frame(a = 1, b = "a")
> dump(c("x", "y"), file = "data.R")
> rm(x, y)
> source("data.R")
> y
```

```
a b
1 1 a
> x
[1] "foo"
```

Interfaces para o mundo exterior

Os dados são lidos usando interfaces *connection*. Conexões podem ser feitas para arquivos (mais comum) ou para outras coisas mais exóticas.

- `file`, abre uma conexão com um arquivo
 - `gzfile`, abre uma conexão para um arquivo compactado com gzip
 - `bzfile`, abre uma conexão para um arquivo compactado com bzip2
 - `url`, abre uma ligação a uma página Web
-

File Connections

```
> str(file)
function (description = "", open = "", blocking = TRUE,
          encoding = getOption("encoding"))
```

- `description` é o nome do arquivo
 - `open` é um código que indica
 - `r` somente leitura
 - `w` escrevendo (e inicializando um novo arquivo)
 - `a` adicionando
 - `rb`, `wb`, `ab` leitura, escrita ou anexação em modo binário (Windows)
-

Connections

Em geral, as conexões são poderosas ferramentas que permitem navegar arquivos ou outros objetos externos. Na prática, muitas vezes não precisamos lidar diretamente com a interface de conexão.

```
con <- file("foo.txt", "r")
data <- read.csv(con)
close(con)
```

is the same as

```
data <- read.csv("foo.txt")
```

Leitura de linhas de um arquivo de texto

```

> con <- gzfile("words.gz")
> x <- readLines(con, 10)
> x
[1] "1080"      "10-point" "10th"      "11-point"
[5] "12-point" "16-point" "18-point" "1st"
[9] "2"        "20-point"

```

WriteLines usa um vetor de caracteres e grava cada elemento uma linha de cada vez em um arquivo de texto.

Leitura de linhas de um arquivo de texto

ReadLines pode ser útil para leitura em linhas de páginas da web

```

## This might take time
con <- url("http://www.jhsph.edu", "r")
x <- readLines(con)
> head(x)
[1] "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">"
[2] ""
[3] "<html>"
[4] "<head>"
[5] "\t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8"

```

The Scientist