Week 1

Martin Aragoneses

July 29, 2015

download.file

```
getwd()
```

```
## [1] "/Users/Martin/Desktop/data/github/Data-Science-R/Coursera/3-Cleaning-Data"
```

```
setwd("~/Desktop/data/github/Data-Science-R/Coursera/3-Cleaning-Data")
if (!file.exists("data")) {# Checks if the directory "data" doesn't exist
   dir.create("data")      # Create a directory "data"
}

str(download.file)
```

```
## function (url, destfile, method, quiet = FALSE, mode = "w", cacheOK = TRUE,
##     extra = getOption("download.file.extra"))
```

```
# .CSV
### Uncomment the following if you're doing it for the first time
fileURL <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.csv?accessType=DOWNLOAD"
##download.file(fileURL, destfile = "./data/cameras.csv", method = "curl")
list.files("./data")
```

```
## [1] "cameras.csv"  "cameras.xlsx"
```

```
# Let's save the date in which the file was downloaded
dateDownloaded <- date()
dateDownloaded
```

```
## [1] "Wed Jul 29 17:22:35 2015"
```

```
str(read.table)
```

```
## function (file, header = FALSE, sep = "", quote = "\"'", dec = ".",
##     row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA",
##     colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip,
##     strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#",
##     allowEscapes = FALSE, flush = FALSE, stringsAsFactors = default.stringsAsFactors(),
##     fileEncoding = "", encoding = "unknown", text)
```

```r
cameraData <- read.table("./data/cameras.csv", sep = ",", header = TRUE)
# same as
cameraData <- read.csv("./data/cameras.csv") # default: sep = ",", header = TRUE)

# .XLSX
fileURL2 <- "https://data.baltimorecity.gov/api/views/dz54-2aru/rows.xlsx?accessType=DOWNLOAD"
##download.file(fileURL2, destfile = "./data/cameras.xlsx", method = "curl")
list.files("./data")
```

```
## [1] "cameras.csv"  "cameras.xlsx"
```

```r
dateDownloaded2 <- date()

## To read xlsx files we need...
## Must do "install.packages("xlsx")" before
library(xlsx)
```

```
## Loading required package: rJava
## Loading required package: xlsxjars
```

```r
cameraData2 <- read.xlsx("./data/cameras.xlsx", sheetIndex = 1, header = TRUE)

## .JSON
## Must run "install.packages("jsonlite")" before
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:utils':
##
##     View
```

```r
jsonData <- fromJSON("https://api.github.com/users/jtleek/repos")
names(jsonData)
```

```
##  [1] "id"                "name"              "full_name"
##  [4] "owner"             "private"           "html_url"
##  [7] "description"       "fork"              "url"
## [10] "forks_url"         "keys_url"          "collaborators_url"
## [13] "teams_url"         "hooks_url"         "issue_events_url"
## [16] "events_url"        "assignees_url"     "branches_url"
## [19] "tags_url"          "blobs_url"         "git_tags_url"
## [22] "git_refs_url"      "trees_url"         "statuses_url"
## [25] "languages_url"     "stargazers_url"    "contributors_url"
## [28] "subscribers_url"   "subscription_url"  "commits_url"
## [31] "git_commits_url"   "comments_url"      "issue_comment_url"
## [34] "contents_url"      "compare_url"       "merges_url"
## [37] "archive_url"       "downloads_url"     "issues_url"
## [40] "pulls_url"         "milestones_url"    "notifications_url"
## [43] "labels_url"        "releases_url"      "created_at"
## [46] "updated_at"        "pushed_at"         "git_url"
## [49] "ssh_url"           "clone_url"         "svn_url"
## [52] "homepage"          "size"              "stargazers_count"
## [55] "watchers_count"    "language"          "has_issues"
## [58] "has_downloads"     "has_wiki"          "has_pages"
## [61] "forks_count"       "mirror_url"        "open_issues_count"
## [64] "forks"             "open_issues"       "watchers"
## [67] "default_branch"
```

```
names(jsonData$owner)
```

```
##  [1] "login"               "id"                  "avatar_url"
##  [4] "gravatar_id"         "url"                 "html_url"
##  [7] "followers_url"       "following_url"       "gists_url"
## [10] "starred_url"         "subscriptions_url"   "organizations_url"
## [13] "repos_url"           "events_url"          "received_events_url"
## [16] "type"                "site_admin"
```

```
names(jsonData$owner$login)
```

```
## NULL
```

```
# To export into JSON

myjson <- toJSON(iris, pretty =TRUE)
```

data.table()

```
library("data.table")
## inherits from data.frame: can use same methods

DT <- data.table(x = rnorm(9), y = rep(c("a", "b", "c"), each = 3), z = rnorm(9))
## same sintax for subsetting

### however: DT[c(2,3)] subsets second AND third ROWS, not an element
DT[c(2,3)]
```

```
##             x y         z
## 1:  0.4983597 a 0.3336832
## 2: -0.1066767 a 0.9749968
```

```
## if x and z are two variables in the data set,
DT[, list(mean(x), sum(z))]
```

```
##            V1      V2
## 1: -0.6680773 0.20469
```

```r
## you can also add a new column w:
DT[, w := z^2]  #makes a change to the original data table
```

```
##               x y            z            w
## 1:  0.06793569 a  1.270483264 1.614128e+00
## 2:  0.49835974 a  0.333683231 1.113445e-01
## 3: -0.10667668 a  0.974996832 9.506188e-01
## 4: -1.03221865 b -1.031782006 1.064574e+00
## 5: -1.87645599 b -0.006930502 4.803186e-05
## 6: -0.40182672 b  0.398933224 1.591477e-01
## 7: -0.19078179 c -0.469824821 2.207354e-01
## 8: -2.61091399 c -0.544633886 2.966261e-01
## 9: -0.36011746 c -0.720235296 5.187389e-01
```

```r
## multiple operations: the last thing that gets returned is what gets created (after ;):

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

```
##               x y            z            w         m
## 1:  0.06793569 a  1.270483264 1.614128e+00 2.6641230
## 2:  0.49835974 a  0.333683231 1.113445e-01 2.5440013
## 3: -0.10667668 a  0.974996832 9.506188e-01 2.5529476
## 4: -1.03221865 b -1.031782006 1.064574e+00 1.5538516
## 5: -1.87645599 b -0.006930502 4.803186e-05 1.6399793
## 6: -0.40182672 b  0.398933224 1.591477e-01 2.3210930
## 7: -0.19078179 c -0.469824821 2.207354e-01 2.1174934
## 8: -2.61091399 c -0.544633886 2.966261e-01 0.8831923
## 9: -0.36011746 c -0.720235296 5.187389e-01 1.9707238
```

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

```
##               x y            z            w         m     a
## 1:  0.06793569 a  1.270483264 1.614128e+00 2.6641230  TRUE
## 2:  0.49835974 a  0.333683231 1.113445e-01 2.5440013  TRUE
## 3: -0.10667668 a  0.974996832 9.506188e-01 2.5529476 FALSE
## 4: -1.03221865 b -1.031782006 1.064574e+00 1.5538516 FALSE
## 5: -1.87645599 b -0.006930502 4.803186e-05 1.6399793 FALSE
## 6: -0.40182672 b  0.398933224 1.591477e-01 2.3210930 FALSE
## 7: -0.19078179 c -0.469824821 2.207354e-01 2.1174934 FALSE
## 8: -2.61091399 c -0.544633886 2.966261e-01 0.8831923 FALSE
## 9: -0.36011746 c -0.720235296 5.187389e-01 1.9707238 FALSE
```

```r
## if we want to summerize a variable by the cases where x>0 vs the cases where x<0
DT[, b := mean(x + w), by = a]
```

```
##               x y            z            w         m     a          b
## 1:  0.06793569 a  1.270483264 1.614128e+00 2.6641230  TRUE  1.1458838
## 2:  0.49835974 a  0.333683231 1.113445e-01 2.5440013  TRUE  1.1458838
## 3: -0.10667668 a  0.974996832 9.506188e-01 2.5529476 FALSE -0.4812146
## 4: -1.03221865 b -1.031782006 1.064574e+00 1.5538516 FALSE -0.4812146
## 5: -1.87645599 b -0.006930502 4.803186e-05 1.6399793 FALSE -0.4812146
## 6: -0.40182672 b  0.398933224 1.591477e-01 2.3210930 FALSE -0.4812146
## 7: -0.19078179 c -0.469824821 2.207354e-01 2.1174934 FALSE -0.4812146
## 8: -2.61091399 c -0.544633886 2.966261e-01 0.8831923 FALSE -0.4812146
## 9: -0.36011746 c -0.720235296 5.187389e-01 1.9707238 FALSE -0.4812146
```

```
####   .N    represents number of times a particular group apears

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

```
##    x     N
## 1: a 33398
## 2: b 33226
## 3: c 33376
```

```
setkey(DT, x) ## set x to be the key

# Now we can subset all the obs in which x == "a" by simply using
DT["a"]
```

```
##        x           y
##     1: a  0.52589082
##     2: a -0.42483070
##     3: a -0.70971553
##     4: a -1.60280587
##     5: a  1.13429376
##    ---
## 33394: a  1.29029347
## 33395: a  0.43763450
## 33396: a -1.24204296
## 33397: a -0.69798392
## 33398: a  0.03938454
```

```
# we can merge two data tables if they have the same key
DT1 <- data.table(x = c("a", "a", "b", "dt1"), y = 1:4)
DT2 <- data.table(x = c("a", "b", "dt2"), y = 5:7)

setkey(DT1, x); setkey(DT2, x)
merge(DT1, DT2)
```

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