Data Analysis and Visualization Exercise 5

Vicente Yépez, Žiga Avsec

3 January 2019

Setup

```
library(data.table)
library(magrittr)
library(tidyr)
```

Questions

Q1 Product dataset

The example_product_data file describes the number of times a person bought product "a" and "b"

```
messy_file <- file.path('extdata', 'example_product_data.csv')
messy_dt <- fread(messy_file)
messy_dt
##            name producta productb
## 1:     John Doe       NA       12
## 2:    Marry Doe        3        1
## 3: John Johnson        5        1
```

Why is this data-set messy? Which columns should a tidy version of this table have?

A1

```
## Values are stored as column names.
## Tidy data columns: name, product, n
```

Q2 Product dataset

Tranform `messy_dt` into a tidy from.

A2

```
tidy_dt <- melt(messy_dt, id.vars = "name", value.name = "n", variable.name = "product")
tidy_dt[, product := gsub("product", "", product)]
tidy_dt
##            name product  n
## 1:     John Doe       a NA
## 2:    Marry Doe       a  3
## 3: John Johnson       a  5
## 4:     John Doe       b 12
## 5:    Marry Doe       b  1
## 6: John Johnson       b  1
```

Q3 Weather dataset

Read in the weather dataset `weather.txt`. Why is this dataset messy? How would a tidy version of it look like?

A3

```
messy_dt <- fread("extdata/weather.txt")
messy_dt %>% head
##             id year month element d1  d2  d3 d4  d5 d6 d7 d8 d9 d10 d11 d12
## 1: MX000017004 2010     1    TMAX NA  NA  NA NA  NA NA NA NA NA  NA  NA  NA
## 2: MX000017004 2010     1    TMIN NA  NA  NA NA  NA NA NA NA NA  NA  NA  NA
## 3: MX000017004 2010     2    TMAX NA 273 241 NA  NA NA NA NA NA  NA 297  NA
## 4: MX000017004 2010     2    TMIN NA 144 144 NA  NA NA NA NA NA  NA 134  NA
## 5: MX000017004 2010     3    TMAX NA  NA  NA NA 321 NA NA NA NA 345  NA  NA
## 6: MX000017004 2010     3    TMIN NA  NA  NA NA 142 NA NA NA NA 168  NA  NA
##    d13 d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30
## 1:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 278
## 2:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 145
## 3:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 299  NA  NA  NA  NA  NA  NA  NA
## 4:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 107  NA  NA  NA  NA  NA  NA  NA
## 5:  NA  NA  NA 311  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 6:  NA  NA  NA 176  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
##    d31
## 1:  NA
## 2:  NA
## 3:  NA
## 4:  NA
## 5:  NA
## 6:  NA

## Why is it messy?
## 1. Variables are stored as columns (days)
## 2. A single entity is scattered across many cells (date)
## 3. Element column is not a variable.
```

```
##
## Tidy version: id, date, tmin, tmax
```

Q4 Weather dataset

Create a tidy version of the weather dataset.

A4

```
## wide -> long
dt <- melt(messy_dt, id.vars = c("id", "year", "month", "element"), variable.name = "day")
# you can ignore the warning message
dt[, day := as.integer(gsub("d", "", day))]

dt = unite(dt, "date", c("year", "month", "day"), sep = "-", remove = TRUE)
# dt[, date := paste(year, month, day, sep = "-")] # other option using paste
# dt[, c("year", "month", "day") := NULL] # remove reduntant columns

dt <- dt[!is.na(date)] ## remove NA dates
dt[, element := tolower(element)] # TMAX -> tmax
dt <- dcast(dt, ... ~ element, value.var = "value") # long -> wide

dt <- dt[!(is.na(tmax) & is.na(tmin))] # remove entries with both NA values,
                                       # na.omit(dt) would also do the job

head(dt)
##             id       date tmax tmin
## 1: MX000017004  2010-1-30  278  145
## 2: MX000017004 2010-10-14  295  130
## 3: MX000017004 2010-10-15  287  105
## 4: MX000017004 2010-10-28  312  150
## 5: MX000017004  2010-10-5  270  140
## 6: MX000017004  2010-10-7  281  129
```

Q5 Scattered data across many files

The `baby-names` folder contains 258 csv-files (`1999.girl.csv`, `1999.boy.csv` , . . .) which store name frequencies for a particular year and sex. Read in the data from all files into one table. *Hint*: when you read many files and gather them into one table, be sure to add a column that identifies each file. `rbindlist()`

A5

```
files <- list.files("extdata/baby-names", full.names = TRUE)
read_append <- function(file) {
```

```
  dt <- fread(file)
  dt[, filename := basename(file)]  # Keep the filename as identifier of sex and yob
  # dt[, filename := strsplit(file, "/")[[1]][3]]   # other option
  return(dt)
}

# See one file
read_append(files[1]) %>% head
##       name  percent     filename
## 1:    John 0.081541 1880.boy.csv
## 2: William 0.080511 1880.boy.csv
## 3:   James 0.050057 1880.boy.csv
## 4: Charles 0.045167 1880.boy.csv
## 5:  George 0.043292 1880.boy.csv
## 6:   Frank 0.027380 1880.boy.csv

dt <- lapply(files, read_append) %>%
  rbindlist

head(dt)
##       name  percent     filename
## 1:    John 0.081541 1880.boy.csv
## 2: William 0.080511 1880.boy.csv
## 3:   James 0.050057 1880.boy.csv
## 4: Charles 0.045167 1880.boy.csv
## 5:  George 0.043292 1880.boy.csv
## 6:   Frank 0.027380 1880.boy.csv
```

Q6

Is the data tidy? If not, tidy it up.

A6

```
# The data is not tidy because one column contains both yob and sex
dt = separate(dt, col = "filename", into = c("year", "sex"), extra = "drop")
head(dt)
##       name  percent year sex
## 1:    John 0.081541 1880 boy
## 2: William 0.080511 1880 boy
## 3:   James 0.050057 1880 boy
## 4: Charles 0.045167 1880 boy
## 5:  George 0.043292 1880 boy
## 6:   Frank 0.027380 1880 boy
```

Small case-study - cleaning up a gene-expression dataset in yeast

Here, we will read and clean up the data from the paper:

- *Bauer et.al., 2007, Coordination of Growth Rate, Cell Cycle, Stress Response, and Metabolic Activity in Yeast, MBoC*, http://www.molbiolcell.org/content/19/1/352.abstract

Read in the data:

```
original_dt <- fread("extdata/gene_expression.tds")
dim(original_dt)
## [1] 5537   40
head(original_dt, n = 2)
##          GID       YORF
## 1: GENE1331X A_06_P5820
## 2: GENE4924X A_06_P5866
##                                                                                NAME
## 1: SFB2       || ER to Golgi transport || molecular function unknown || YNL049C || 1082129
## 2:       || biological process unknown || molecular function unknown || YNL095C || 1086222
##    GWEIGHT G0.05  G0.1 G0.15  G0.2 G0.25  G0.3 N0.05 N0.1 N0.15  N0.2 N0.25
## 1:       1 -0.24 -0.13 -0.21 -0.15 -0.05 -0.05  0.20 0.24 -0.20 -0.42 -0.14
## 2:       1  0.28  0.13 -0.40 -0.48 -0.11  0.17  0.31 0.00 -0.63 -0.44 -0.26
##    N0.3 P0.05  P0.1 P0.15  P0.2 P0.25 P0.3 S0.05  S0.1 S0.15 S0.2 S0.25 S0.3
## 1: 0.09 -0.26 -0.20 -0.22 -0.31  0.04 0.34 -0.51 -0.12  0.09 0.09  0.20 0.08
## 2: 0.21 -0.09 -0.04 -0.10  0.15  0.20 0.63  0.53  0.15 -0.01 0.12 -0.15 0.32
##    L0.05 L0.1 L0.15 L0.2 L0.25 L0.3 U0.05  U0.1 U0.15  U0.2 U0.25 U0.3
## 1:  0.18 0.18  0.13 0.20  0.17 0.11 -0.06 -0.26 -0.05 -0.28 -0.19 0.09
## 2:  0.16 0.09  0.02 0.04  0.03 0.01 -1.02 -0.91 -0.59 -0.61 -0.17 0.18
```

Column description:

- GID - gene ID
- YORF - Some other ID
- NAME - gene description composed of:
  - Gene name
  - Biological process
  - Molecular function
  - Systematic ID
  - Some other ID
- GWEIGHT - some type of weight
- G0.05, .., P0.03 - gene expression values for measured at different nutritient and growth rates:
  - Nutritients (G, N, P, . . .):
    - G = Glucose
    - L = Leucine
    - P = Phosphate
    - S = Sulphate
    - N = Ammonia
    - U = Uracil
  - Growth rate (0.05, 0.3, . . .)

Q6

Why is this dataset not tidy?

A6

```
## - Column headers are values, not variable names.
## - Multiple variables are stored in the column "Name".
```

Q7 - Transorm it into a tidy form

Provide a tidy dataset in the folowing form:

```
##    name           biological_process            molecular_function
## 1: SFB2        ER to Golgi transport    molecular function unknown
## 2:        biological process unknown    molecular function unknown
## 3: QRI7 proteolysis and peptidolysis metalloendopeptidase activity
## 4: CFT2      mRNA polyadenylylation*                   RNA binding
## 5: SSO2              vesicle fusion*              t-SNARE activity
## 6: PSP2   biological process unknown    molecular function unknown
##    systematic_name nutrient rate expression
## 1:         YNL049C  Glucose 0.05      -0.24
## 2:         YNL095C  Glucose 0.05       0.28
## 3:         YDL104C  Glucose 0.05      -0.02
## 4:         YLR115W  Glucose 0.05      -0.33
## 5:         YMR183C  Glucose 0.05       0.05
## 6:         YML017W  Glucose 0.05      -0.69
```

A7

```
## Define some constants:
nutrient_names <- c(G = "Glucose", L = "Leucine", P = "Phosphate",
                    S = "Sulfate", N = "Ammonia", U = "Uracil")

NAME_names <- c("name", "biological_process",
                "molecular_function", "systematic_name", "some_other_id")

# remove unnecessary columns
original_dt[, c("GID", "YORF", "GWEIGHT") := NULL]

## melt the G0.3 .. columns:
melted_dt <- melt(original_dt, id.vars = "NAME", value.name = "expression")

## separate the varaible G0.4.. into multiple columns
melted_dt <- separate(melted_dt, "variable", c("nutrient", "rate"), sep = 1)

## Use full names for nutritient names
```

```
melted_dt[, nutrient := nutrient_names[nutrient]]
# melted_dt[, nutrient := revalue(nutrient, nutrient_names)]  # other option would be mapvalues

## separate the NAME column
melted_dt <- separate(melted_dt, NAME, NAME_names, sep = "\\|\\|")

## remove other id
melted_dt[, some_other_id := NULL]


head(dt)
##       name  percent year sex
## 1:    John 0.081541 1880 boy
## 2: William 0.080511 1880 boy
## 3:   James 0.050057 1880 boy
## 4: Charles 0.045167 1880 boy
## 5:  George 0.043292 1880 boy
## 6:   Frank 0.027380 1880 boy
```