Simple Data Manipulation & Visualization - Tidy data

Vangelis Theodorakis, Xueqi Cao

22 June, 2020

Setup

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

Question 1

Read the weather dataset `weather.txt`. It contains the minimal and maximal temperature on a certain city (id) over different dates (year, month, d1-d31). Why is this dataset messy? How would a tidy version of it look like? Create its tidy version.

```
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 d13
## 1: MX000017004 2010     1    TMAX NA  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  NA
## 3: MX000017004 2010     2    TMAX NA 273 241 NA  NA NA NA NA NA  NA 297  NA  NA
## 4: MX000017004 2010     2    TMIN NA 144 144 NA  NA NA NA NA NA  NA 134  NA  NA
## 5: MX000017004 2010     3    TMAX NA  NA  NA NA 321 NA NA NA NA 345  NA  NA  NA
## 6: MX000017004 2010     3    TMIN NA  NA  NA NA 142 NA NA NA NA 168  NA  NA  NA
##    d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24 d25 d26 d27 d28 d29 d30 d31
## 1:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 278  NA
## 2:  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 145  NA
## 3:  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 107  NA  NA  NA  NA  NA  NA  NA
## 5:  NA  NA 311  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 6:  NA  NA 176  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
dim(messy_dt)
```

```
## [1] 22 35
```

```
## 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
```

```
tidy_dt <- messy_dt %>%
  melt(id.vars=c('id', 'year', "month", "element"), na.rm=TRUE) %>%
```

```
  .[, variable := gsub('d', '', variable)] %>%
  unite(col=date, year, month, variable, sep='-') %>%
  dcast(... ~ element) %>%
  .[, date := as.Date(date)]
```

```
## 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))]

# Join all date related columns into one. Use unite or paste
# 1. Using unite():
dt <- unite(dt, "date", c("year", "month", "day"), sep = "-", remove = TRUE)

# 2. Using paste():
# dt[, date := paste(year, month, day, sep = "-")] # convert to date
# dt[, c("year", "month", "day") := NULL] # remove reduntant columns


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
```

```
dim(dt)
```

```
## [1] 33  4
```

```
# An alternative tidy code version
tidy_dt <- messy_dt %>%
  melt(id.vars=c('id', 'year', 'month', 'element'), na.rm=TRUE) %>%
  .[, variable := gsub('d', '', variable)] %>%
  unite(date, year, month, variable, sep='-') %>%
  dcast(... ~ element) %>%
  .[, date := as.Date(date)]
```