Dirty Spreadsheet Data

Patrick Kelly

2/13/2020

Click Here"Excel in the era of big data"

Spreadsheets

Most people with computers use them. But few use them well. Introduced in 1980, VisiCalc was the first "killer" application for personal computers. Today there are many,many choices. Excel captures about 85% of the market. OPenOffice Calc and LibreOffice Calc are two popular platforms.

Question: At what grade level are spreadsheeet introduced in school? Probably not early enough.

Data scientists spend up to 80% of their time wrangling dirty data into a usable format. This drudgery could be greatly reduced if people who present data in spreadsheets are taught that humans and computers have different requirements for interpretability.

Here is a data format that humans can understand.

Sales 2018					
Fruits	Apple	5000	**Total : 12500**		
	Orange	7000			
	Banana	500			
Vegetables	Carrot	3000	**Total : 8500**		
	Cucumber	4500			
	Eggplant	1000		**Total 2018**	**21000**
Sales 2019					
Fruits	Apple	3500	**Total : 13000**		
	Orange	8000			
	Banana	1500			
Vegetables	Carrot	4000	**Total : 9500**		
	Cucumber	4200			
	Eggplant	1300		**Total 2019**	**22500**

Figure 1: Human Version

And here is a format of the same data that computers can understand.

Thequestion is: How do we get from Dirty to Tidy?

year	type	product	quantity
2018	fruit	apple	5000
2018	fruit	orange	7000
2018	fruit	banana	500
2018	vegetables	carrot	3000
2018	vegetables	cucumber	4500
2018	vegetables	eggplant	1000
2019	fruit	apple	3500
2019	fruit	orange	8000
2019	fruit	banana	1500
2019	vegetables	carrot	4000
2019	vegetables	cucumber	4200
2019	vegetables	eggplant	1300

Figure 2: Computer Version

The first challenge is scraping the data from the URL.

Here is how I did it.

1. Copy the image of the data with the screencapture function (Command+Shift+4) on the Mac.
2. Save to clipboard.
3. Load the image into Photoshop.
4. Save to a jpg file.

Click Here"Online OCR"

5. Transform the file from jpg to xlsx.
    a. Online Optical Character Recognition software
    b. Upload the jpg file and download the xlsx file.
    c. Load the xlsx filel into Excel.
    d. Save it to a csv file without any tweaking.
6. Now load the cv file into R.

```
dirty <- read.csv("Dirty_Excel.csv",
        stringsAsFactors = FALSE)
dirty
```

```
##     Sales.2018    X           X.1        X.2   X.3
## 1        Apple 5000                            NA
## 2       Orange 7000 Total : 12500              NA
## 3       Banana  500                            NA
## 4       Carrot 3000                            NA
## 5     Cucumber 4500  Total : 8500              NA
## 6     Eggplant 1000               Total 2018 21000
## 7                NA                            NA
## 8   Sales 2019   NA                            NA
## 9        Apple 3500                            NA
## 10      Orange 8000 Total : 13000              NA
## 11      Banana 1500                            NA
## 12      Carrot 4000                            NA
## 13    Cucumber 4200  Total : 9500              NA
## 14    Eggplant 1300               Total 2019 22500
## 15               NA                            NA
## 16      Fruits   NA                            NA
## 17 Vegetable:"   NA                            NA
## 18           I   NA                            NA
## 19               NA                            NA
## 20      Fruits   NA                            NA
## 21           1   NA                            NA
## 22  Vegetables   NA                            NA
## 23           I   NA                            NA
```

OUCH!. This is even dirtier than the original.

Why didn't I clean the file in Excel. I want my wrangling to be reproducible by anyone. This is easy if all the transformations are done with R code.

So now let's do some tidying.

```
tidy <- data.frame(year = c(rep(2018,6),
        rep(2019,6)))
tidy$type <-c(rep(dirty$Sales.2018[20],3),
   rep(dirty$Sales.2018[22],3),
   rep(dirty$Sales.2018[20],3),
  rep(dirty$Sales.2018[22],3))
tidy$product <- c(dirty$Sales.2018[1:6],
    dirty$Sales.2018[9:14])
tidy$quantity <- c(dirty$X[1:6],
    dirty$X[9:14])
tidy
```

```
##    year       type  product quantity
## 1  2018     Fruits    Apple     5000
## 2  2018     Fruits   Orange     7000
## 3  2018     Fruits   Banana      500
## 4  2018 Vegetables   Carrot     3000
## 5  2018 Vegetables Cucumber     4500
## 6  2018 Vegetables Eggplant     1000
## 7  2019     Fruits    Apple     3500
## 8  2019     Fruits   Orange     8000
## 9  2019     Fruits   Banana     1500
## 10 2019 Vegetables   Carrot     4000
## 11 2019 Vegetables Cucumber     4200
## 12 2019 Vegetables Eggplant     1300
```

Exploratory Data Anaalysis (EDA)

```
dim(tidy)
```

```
## [1] 12  4
```

```
names(tidy)
```

```
## [1] "year"     "type"     "product"  "quantity"
```

```
anyNA(tidy)
```

```
## [1] FALSE
```

```
str(tidy)
```

```
## 'data.frame':    12 obs. of  4 variables:
##  $ year    : num  2018 2018 2018 2018 2018 ...
##  $ type    : chr  "Fruits" "Fruits" "Fruits" "Vegetables" ...
##  $ product : chr  "Apple" "Orange" "Banana" "Carrot" ...
##  $ quantity: int  5000 7000 500 3000 4500 1000 3500 8000 1500 4000 ...
```

```
summary(tidy)
```

```
##       year          type              product             quantity
##  Min.   :2018   Length:12          Length:12          Min.   : 500
##  1st Qu.:2018   Class :character   Class :character   1st Qu.:1450
##  Median :2018   Mode  :character   Mode  :character   Median :3750
##  Mean   :2018                                         Mean   :3625
##  3rd Qu.:2019                                         3rd Qu.:4625
##  Max.   :2019                                         Max.   :8000
```

Now for some analysis

```
suppressMessages(library(dplyr))
suppressMessages(library(formattable))
by_group <- tidy %>% group_by(year,type)
table_1 <- by_group %>% summarize(total = sum(quantity))
formattable(table_1, align =c("c","c","c"))
```

year

type

total

2018

Fruits

12500

2018

Vegetables

8500

2019

Fruits

13000

2019

Vegetables

9500

```
by_year <- tidy %>% group_by(year)
table_2 <- by_year %>% summarize(total = sum(quantity))
formattable(table_2, align =c("c","c"))
```

year

total

2018

21000

2019

22500

Click Here" Using Formattable"

```
library(linguisticsdown)
library(htmlwidgets)
voila <- "voila.gif"
include_graphics2(voila)
```

View gif at voila.gif

Chose dite. . . chose faite.