Data Wrangling Exercise 1

Joe Marco

July 21, 2017

### Step 1: Load Librarys for packages related to data manipulation

* library(devtools)
* library(dplyr)
* library(tidyr)

library(devtools)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

### Step 2:

* Locate the CSV file (view it if you like to see what it looks like) in the below location and save it.
* <https://drive.google.com/file/d/0B9vKjeWdQHa5OE5UQ1k4TWJlelU/view>

### Step 3:

* Load CSV file, and give it the name you wish to proceed with. In my case I am calling this data set "wrangle". You may need to open the excel sheet(xls) and save it as a CSV if needed. I named my file refine.csv and saved it. (ENSURE THE FILE IS SAVED IN THE SAME DIRECTORY AS R CODE).
* After naming the file, you can take an optional step to view the data to see what you want to manipulate

wrangle <- read.csv('refine.csv')  
#optional step below  
# View (wrangle)

### Step 4:

* Clean up the 'company' column so all of the misspellings of the brand names are standardized. For example, you can transform the values in the column to be: philips, akzo, van houten and unilever (all lowercase).
* Take the company code column from refine.csv and place all the values to lower case.
* Before manipulation:

wrangle$company

## [1] Phillips phillips philips phllips phillps phillipS   
## [7] akzo Akzo AKZO akz0 ak zo akzo   
## [13] akzo phillips fillips phlips Van Houten van Houten  
## [19] van houten van houten Van Houten unilver unilever Unilever   
## [25] unilever   
## 19 Levels: ak zo akz0 akzo Akzo AKZO fillips philips phillips ... Van Houten

* Then apply below code:

wrangle$company <- tolower(wrangle$company)

* Result after applying above code (as you can see all values are now in lower case:

wrangle$company

## [1] "phillips" "phillips" "philips" "phllips" "phillps"   
## [6] "phillips" "akzo" "akzo" "akzo" "akz0"   
## [11] "ak zo" "akzo" "akzo" "phillips" "fillips"   
## [16] "phlips" "van houten" "van houten" "van houten" "van houten"  
## [21] "van houten" "unilver" "unilever" "unilever" "unilever"

### Step 5:

* Correct the mispellings (imagine these as user entries and typos - very common) - you will need a mixture of Regex to find values, and replace with your defined common values
* Below code read outloud: Take company code from wrangle, sub out the pattern (uses regex) to find any value that ends with "ps" (.\*\ps$) and replace it with the value "philips" in the company column of wrangle

wrangle$company <- sub(pattern = ".\*\\ps$" , replacement = "philips", x = wrangle$company)

* Below code read outloud: Take company code from wrangle, sub out the pattern (uses regex) to find any value that begins with "ak" ("^ak.\*") and replace it with the value "akzo" in the company column of wrangle

wrangle$company <- sub(pattern = "^ak.\*" , replacement = "akzo", x = wrangle$company)

* Below code read outloud: Take company code from wrangle, sub out the pattern (uses regex) to find any value that begins with "u"" ("^u.\*") and replace it with the value "unilever" in the company column of wrangle

wrangle$company <- sub(pattern = "^u.\*" , replacement = "unilever", x = wrangle$company)

* Below code read outloud: Take company code from wrangle, sub out the pattern (uses regex) to find any value that ends begins with "v" ("^v.\*") and replace it with the value "van houten" in the company column of wrangle

wrangle$company <- sub(pattern = "^v.\*" , replacement = "van houten", x = wrangle$company)

### Step 6:

* Separate the product code and product number into separate columns i.e. add two new columns called product\_code and product\_number, containing the product code and number respectively
* We need to seperate the column labeled "Product.code...number" in the wrangle data set using the below code

wrangle <- separate (wrangle, "Product.code...number", c("product\_code", "product\_number"), sep = "-")

### Step 7:

Add product categories. You learn that the product codes actually represent the following product categories:

* p = Smartphone
* v = TV
* x = Laptop
* q = Tablet

Below code labels a new column "product\_category", then we use a combo of the sub function and regex to replace anything beginning with p with smartphone, replace anything beginning with x with laptop, replace anything beginning with v with TV, and replace anything beginning with q with tablet -- from the product code column in the wrangle data set.

wrangle$product\_category <- sub(pattern = "^p$", replacement = "Smartphone", sub("^x$", "Laptop", sub("^v$", "TV", sub("^q$", "Tablet", wrangle$product\_code))))

### Step 8:

* Add full address for geocoding
* You'd like to view the customer information on a map. In order to do that, the addresses need to be in a form that can be easily geocoded. Create a new column full\_address that concatenates the three address fields (address, city, country), separated by commas.

wrangle <- wrangle %>%   
 mutate(full\_address = paste(address, city, country, sep = ","))

### Step 9:

* Create dummy variables for company and product category
* Both the company name and product category are categorical variables i.e. they take only a fixed set of values. In order to use them in further analysis you need to create dummy variables. Create dummy binary variables for each of them with the prefix company\_ and product\_ i.e.,
* Add four binary (1 or 0) columns for company: company\_philips, company\_akzo, company\_van\_houten and company\_unilever.
* Add four binary (1 or 0) columns for product category: product\_smartphone, product\_tv, product\_laptop and product\_tablet
* Each mutate below creates a new column, a total of 8 new columns. Each column is filled with a a dummy variable either 1 or 0.

wrangle <- mutate(wrangle, company\_philips = ifelse(company == "philips", 1, 0))  
 wrangle <- mutate(wrangle, company\_akzo = ifelse(company == "akzo", 1, 0))  
 wrangle <- mutate(wrangle, company\_van\_houten = ifelse(company == "van houten", 1, 0))  
 wrangle <- mutate(wrangle, company\_unilever = ifelse(company == "unilever", 1, 0))  
 wrangle <- mutate(wrangle, product\_smartphone = ifelse(product\_category == "Smartphone", 1, 0))  
 wrangle <- mutate(wrangle, product\_tv = ifelse(product\_category == "TV", 1, 0))  
 wrangle <- mutate(wrangle, product\_laptop = ifelse(product\_category == "Laptop", 1, 0))  
 wrangle <- mutate(wrangle, product\_tablet = ifelse(product\_category == "Tablet", 1, 0))

### Step 10:

* FINALLY you can write your finished output back to a new file. Below code will allow you to save as csv the manipulated and newly created file!

write.csv(wrangle, "Data\_wrangling\_exercise1.csv")