refine

milti leonard

9/18/2017

Choosing my weapons

I used the **tidyverse**, **dplyr** and **stringr** packages.

```
library(dplyr)
library(tidyverse)
library(stringr)
```

0: Load the data in RStudio

Save the data set as a CSV file called **refine_original.csv** and load it in RStudio into a data frame.

```
refine <- read_csv("~/R/github/springboard exercises/springboard exercises/refine_original.csv")
```

1: Clean up brand names

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).

```
akzo <- which(tolower(refine$company) %>% str_detect("ak"))
vanhouten <- which(tolower(refine$company) %>% str_detect("hou"))
unilever <- which(tolower(refine$company) %>% str_detect("uni"))
phillips <- which(tolower(refine$company) %>% str_detect("ps"))
refine$company[phillips] = "philips"
refine$company[akzo] = "akzo"
refine$company[vanhouten] = "van houten"
refine$company[unilever] = "unilever"
```

2: Separate product code and number

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

```
refine <- add_column(refine, product_code = "1", .before = 2)
refine <- add_column(refine, product_number = "1", .after = 2)
refine$product_code = str_sub(as.character(refine$`Product code / number`),
    1, 1)
refine$product_number <- str_split(refine$`Product code / number`,
    "-") %>% sapply("[", 2)
refine <- refine[-4]
```

3: Add product categories

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

```
p = Smartphone
```

v = TV

x = Laptop

q = Tablet

In order to make the data more readable, add a column with the product category for each record.

```
refine$product_code <- refine$product_code %>% str_replace_all(c(p = "Smartphone",
    v = "TV", x = "Laptop", q = "Tablet"))
```

4: 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.

```
refine <- add_column(refine, full_address = , .after = 4)
refine <- unite(refine, full_address, c(address, city, country),
    sep = ", ", remove = TRUE)
# refine <- refine[-c(5:7)]
```

5: 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**

Add four binary (1 or 0) columns for product category: **product_smartphone**, **product_tv**, **product_l

```
for (t in unique(refine$company)) {
    refine[paste("company", t, sep = "_")] <- ifelse(refine$company ==
        t, 1, 0)
}
for (p in unique(refine$product_code)) {
    refine[paste("product", tolower(p), sep = "_")] <- ifelse(refine$product_code ==
        p, 1, 0)
}

# add_column(refine, product_smartphone, product_tv,
# product_laptop, product_tablet, .after = 2)
# add_column(refine, company_philips, company_akzo,
# company_van_houten, company_unilever, .after = 1)
```