Grocery Data Analysis (Part 2)

Stephen Bernardo

2023-08-27

Importing the necessary packages

Reading the dataset

```
df <- read.csv('ities.csv')
```

Counting the rows and columns

```
nrow(df)
```

```
## [1] 438151
```

```
ncol(df)
```

```
## [1] 13
```

Number of Rows: **438151**

Number of Columns: **13**

Displaying the dataframe structure

```
str(df)
```

```
## 'data.frame':    438151 obs. of  13 variables:
##  $ Date             : chr  "7/18/2016" "7/18/2016" "7/18/2016" "7/18/2016" ...
##  $ OperationType    : chr  "SALE" "SALE" "SALE" "SALE" ...
##  $ CashierName      : chr  "Wallace Kuiper" "Wallace Kuiper" "Wallace Kuiper" "Wallace Kuiper" ...
##  $ LineItem         : chr  "Salmon and Wheat Bran Salad" "Fountain Drink" "Beef and Squash Kabob" "Sa
##  $ Department       : chr  "Entrees" "Beverage" "Kabobs" "Salad" ...
##  $ Category         : chr  "Salmon and Wheat Bran Salad" "Fountain" "Beef" "general" ...
##  $ RegisterName     : chr  "RT149" "RT149" "RT149" "RT149" ...
##  $ StoreNumber      : chr  "AZ23501305" "AZ23501289" "AZ23501367" "AZ23501633" ...
##  $ TransactionNumber: chr  "002XIIC146121" "002XIIC146121" "00PG9FL135736" "00Z3B4R37335" ...
##  $ CustomerCode     : chr  "CWM11331L8O" "CWM11331L8O" "CWM11331L8O" "CWM11331L8O" ...
##  $ Price            : num  66.22 2.88 12.02 18.43 18.43 ...
##  $ Quantity         : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ TotalDue         : num  66.22 2.88 24.04 18.43 18.43 ...
```

Two Main Points:

1. There are **438151 observations** (which corresponds to the number of rows in task 2) and **13 variables** (which corresponds to the number of columns in task 2) in the dataset.

2. Out of the 13 variables, 10 are **characters**, 2 are **numerical**, and 1 is an **integer**. With the data types given, it would be helpful if some of these are converted (such as Date and Category) for data to be summarized in a more meaningful manner.

Displaying a summary of the variables

```
summary(df)
```

```
##      Date           OperationType      CashierName          LineItem
##  Length:438151      Length:438151      Length:438151      Length:438151
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Department          Category         RegisterName       StoreNumber
##  Length:438151      Length:438151      Length:438151      Length:438151
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  TransactionNumber  CustomerCode           Price             Quantity
##  Length:438151      Length:438151      Min.   :-5740.51   Min.   :  1.000
##  Class :character   Class :character   1st Qu.:    4.50   1st Qu.:  1.000
##  Mode  :character   Mode  :character   Median :   11.29   Median :  1.000
##                                        Mean   :   14.36   Mean   :  1.177
##                                        3rd Qu.:   14.68   3rd Qu.:  1.000
##                                        Max.   :21449.97   Max.   :815.000
##                                        NA's   :12
##     TotalDue
##  Min.   :-5740.51
##  1st Qu.:    4.50
##  Median :   11.80
##  Mean   :   15.26
##  3rd Qu.:   15.04
##  Max.   :21449.97
##  NA's   :12
```

Sample two columns that have data type that is not useful:

1. **Category** - This can be converted to categorical data using the factor function in order to be helpful in summarizing how much was sold for a given category. This can also be helpful when the company owning the data wants to determine any trends in the amount of returns for an item, such as which category sees the highest amount of sales and returns and why.
2. **Date** - Date can be converted to a date data type so that it is possible to summarize data frequency by date. This can be helpful when the company owning the data wants to determine the daily (or weekkly) trends in the amount in a sales of certain products or overall sales.
3. **OperationType** - This can be converted to categorical data using the factor function in order to be helpful in summarizing how many items are sold and how much in US dollars are sold. This can also be helpful when the company owning the data wants to determine the total amount of returns and which category sees the highest returns (if Category is also converted to a factor).

Converting to lower case and displaying first 5 rows

Converting Department and LineItem columns to lower case

```
df$Department_lower <- tolower(df$Department)
df$LineItem_lower <- tolower(df$LineItem)
```

Showing the first five rows

```
head(df[,c("Department","Department_lower","LineItem","LineItem_lower")],5)
```

```
##   Department Department_lower                    LineItem
## 1    Entrees          entrees Salmon and Wheat Bran Salad
## 2   Beverage         beverage              Fountain Drink
## 3     Kabobs           kabobs       Beef and Squash Kabob
## 4      Salad            salad Salmon and Wheat Bran Salad
## 5      Salad            salad Salmon and Wheat Bran Salad
##                LineItem_lower
## 1 salmon and wheat bran salad
## 2              fountain drink
## 3       beef and squash kabob
## 4 salmon and wheat bran salad
## 5 salmon and wheat bran salad
```

Explaying why there is an error

Use the "plot" function on Department_lower, and then run that code chunk. You will get an error. .

```
#plot(df$Department_lower)
```

The reason why we have this error is that the plot does not accept character values. Hence, Department_lower needs to be converted to categorical data, which is a finite value.

Converting to factor type without creating new column

```
df$Department_lower <- as.factor(df$Department_lower)
df$LineItem_lower <- as.factor(df$LineItem_lower)

#Checking the new dataframe structure
str(df)
```

```
## 'data.frame':    438151 obs. of  15 variables:
##  $ Date             : chr  "7/18/2016" "7/18/2016" "7/18/2016" "7/18/2016" ...
##  $ OperationType    : chr  "SALE" "SALE" "SALE" "SALE" ...
##  $ CashierName      : chr  "Wallace Kuiper" "Wallace Kuiper" "Wallace Kuiper" "Wallace Kuiper" ...
##  $ LineItem         : chr  "Salmon and Wheat Bran Salad" "Fountain Drink" "Beef and Squash Kabob" "Sa
##  $ Department       : chr  "Entrees" "Beverage" "Kabobs" "Salad" ...
##  $ Category         : chr  "Salmon and Wheat Bran Salad" "Fountain" "Beef" "general" ...
##  $ RegisterName     : chr  "RT149" "RT149" "RT149" "RT149" ...
##  $ StoreNumber      : chr  "AZ23501305" "AZ23501289" "AZ23501367" "AZ23501633" ...
##  $ TransactionNumber: chr  "002XIIC146121" "002XIIC146121" "00PG9FL135736" "00Z3B4R37335" ...
##  $ CustomerCode     : chr  "CWM11331L8O" "CWM11331L8O" "CWM11331L8O" "CWM11331L8O" ...
##  $ Price            : num  66.22 2.88 12.02 18.43 18.43 ...
##  $ Quantity         : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ TotalDue         : num  66.22 2.88 24.04 18.43 18.43 ...
##  $ Department_lower : Factor w/ 9 levels "beverage","catering",..: 3 1 6 7 7 4 6 3 3 1 ...
##  $ LineItem_lower   : Factor w/ 68 levels "aubergine and chickpea vindaloo",..: 53 37 14 53 53 9 22 1
```

There were **9 levels** in the **Department_lower** column

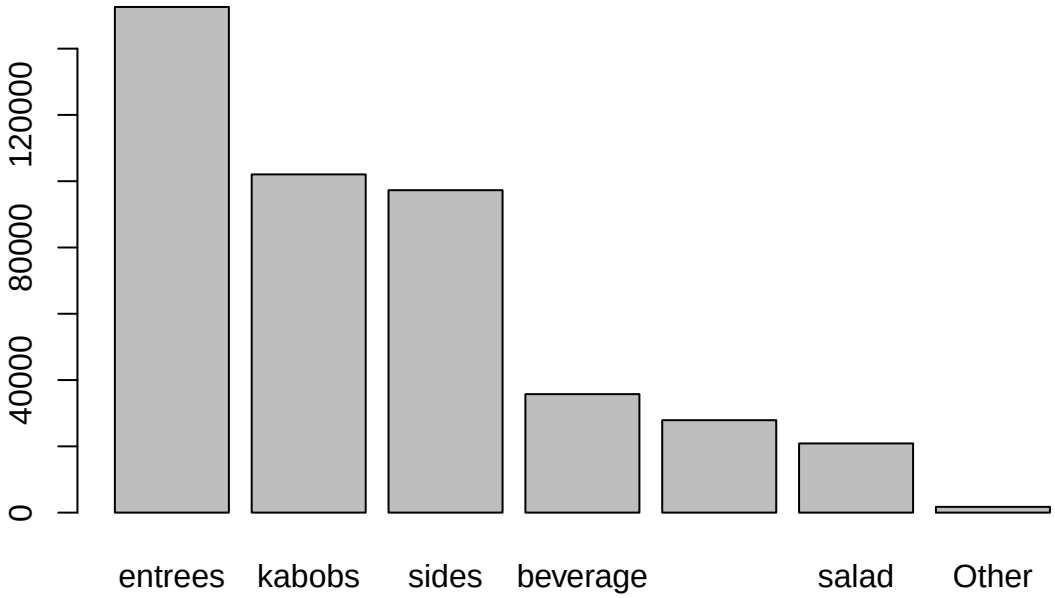
Plotting the Department_lower column

```
#Grouping data according to Department_lower and to sorting to descending order
df$Department_lower_order <- forcats::fct_infreq(fct_lump(df$Department_lower,n=6))

#Displaying summary
summary(df$Department_lower_order)
```

```
##  entrees   kabobs    sides beverage  general    salad    Other
##   152575   102053    97284    35746    27885    20870     1738
```

```
#Display plot
plot(df$Department_lower_order)
```




```
#Grouping data according to Department_lower and summarizing using the count of transactions
Department_lower_order2 <- forcats::fct_infreq(df$Department_lower)

#summarize
summary(Department_lower_order2)
```

```
##    entrees     kabobs      sides   beverage    general      salad gift cards
##     152575     102053      97284      35746      27885      20870        731
##   catering       swag
##        651        356
```

Using the plot above, the **most frequent** Department that occured in the date is **entrees**, with a frequency of **152575**. The **least frequent** Department in terms of occurrence is **swag**, with a frequency of only 356.

Note: The The least frequent Department was found by using the fct_infreq function and then summary function to get the results, which is a summary of categories arranged per frequency.