Assignment

Eric Liu

2023-01-17

Reading and preprocessing data

Clean the datasets

```
detroit = read.csv(file = 'detroit_purchases.csv')
newyork = read.csv(file = 'new_york_purchases.csv')
detroit[] = lapply(detroit, gsub, pattern='\\$', replacement='') #Detroit has '$' in the variable 'amou
detroit[, 1] = as.integer(detroit[, 1]) #Since the first column is char, we need int
detroit[, 2] = as.numeric(detroit[, 2]) #Since the second column is char, we need num
detroit[, 3] = as.numeric(detroit[, 3]) #Since the third column is char, we need num
newyork[, 4] = substring(newyork[, 4], 1, 20) # Since there are "+0000" in timestamp for new_york data
str(detroit)
```

```
## 'data.frame':    27 obs. of  5 variables:
##  $ id                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ barcode           : num  1.84e+12 7.76e+12 7.41e+12 3.47e+12 5.58e+12 ...
##  $ amount            : num  1.61 3.86 2.56 2.3 3.67 2.8 2.67 2.9 1.87 3.32 ...
##  $ purchase_timestamp: chr  "2023-01-01 02:58:07 " "2023-01-01 20:34:02 " "2023-01-01 14:01:41 " "202
##  $ type              : chr  "vegetable" "vegetable" "dairy" "vegetable" ...
```

```
str(newyork)
```

```
## 'data.frame':    27 obs. of  5 variables:
##  $ id                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ barcode           : num  7.67e+11 1.17e+12 6.96e+12 5.36e+12 7.91e+12 ...
##  $ amount            : num  3.01 3.48 3.66 3.07 3.74 2.86 2.25 3.81 3.29 2.43 ...
##  $ purchase_timestamp: chr  "2023-01-01 08:33:37 " "2023-01-01 00:41:34 " "2023-01-01 18:22:27 " "202
##  $ type              : chr  "puffs" "cakes" "tomato" "beans" ...
```

Assignment 1

Normalize the type field to a product line (New_York dataset)

```
count_1 = 1
ny_type_new = list() #create a list for output
for (x in newyork[ ,5]){
  if (x %in% c("cakes", "pizzas", "puffs"))
```

```
    ny_type_new[count_1] = print("bakery")
  else if (x %in% c("milk", "cheese"))
    ny_type_new[count_1] = print("dairy")
  else
    ny_type_new[count_1] = print("vegetable")
count_1 = count_1 + 1
}
```

```
## [1] "bakery"
## [1] "bakery"
## [1] "vegetable"
## [1] "vegetable"
## [1] "bakery"
## [1] "vegetable"
## [1] "dairy"
## [1] "bakery"
## [1] "bakery"
## [1] "dairy"
## [1] "bakery"
## [1] "bakery"
## [1] "vegetable"
## [1] "vegetable"
## [1] "vegetable"
## [1] "vegetable"
## [1] "dairy"
## [1] "bakery"
## [1] "bakery"
## [1] "dairy"
## [1] "bakery"
## [1] "bakery"
## [1] "vegetable"
## [1] "dairy"
## [1] "vegetable"
## [1] "vegetable"
## [1] "bakery"
```

```
newyork$type = ny_type_new
```

Merge two CSV files into a single dataset

```
data_1 = rbind(detroit, newyork)
data_1$id = c(1:54) #re-range the id order
data_1
```

```
##    id      barcode amount   purchase_timestamp      type
## 1   1 1.835566e+12   1.61 2023-01-01 02:58:07  vegetable
## 2   2 7.758948e+12   3.86 2023-01-01 20:34:02  vegetable
## 3   3 7.410145e+12   2.56 2023-01-01 14:01:41      dairy
## 4   4 3.470283e+12   2.30 2023-01-01 01:50:27  vegetable
## 5   5 5.583888e+12   3.67 2023-01-01 21:49:34      dairy
## 6   6 6.986147e+12   2.80 2023-01-01 17:23:27      dairy
## 7   7 8.765003e+12   2.67 2023-01-01 23:14:17  vegetable
## 8   8 1.463020e+12   2.90 2023-01-01 11:55:07      dairy
## 9   9 8.063514e+12   1.87 2023-01-01 11:59:53      dairy
## 10 10 7.690345e+12   3.32 2023-01-02 16:21:08     bakery
## 11 11 1.643365e+12   2.61 2023-01-02 08:10:58     bakery
## 12 12 7.539630e+12   2.39 2023-01-02 08:55:13     bakery
## 13 13 4.005177e+12   1.69 2023-01-02 10:11:54  vegetable
## 14 14 2.832167e+11   2.59 2023-01-02 10:20:31      dairy
## 15 15 1.204563e+11   2.81 2023-01-02 18:32:20  vegetable
## 16 16 4.203182e+12   2.62 2023-01-02 18:23:10      dairy
## 17 17 9.256742e+12   2.01 2023-01-02 19:41:00  vegetable
## 18 18 9.603244e+12   1.73 2023-01-02 14:16:08  vegetable
## 19 19 4.127156e+12   3.00 2023-01-03 11:29:11      dairy
## 20 20 7.615279e+12   2.25 2023-01-03 22:03:18     bakery
## 21 21 8.440080e+11   2.50 2023-01-03 04:53:51     bakery
## 22 22 5.734283e+12   3.27 2023-01-03 04:00:57      dairy
## 23 23 8.742240e+12   2.27 2023-01-03 03:47:43     bakery
## 24 24 6.048049e+12   2.11 2023-01-03 14:13:58     bakery
## 25 25 8.677556e+11   3.31 2023-01-03 18:23:07     bakery
## 26 26 5.586696e+12   2.09 2023-01-03 21:01:01      dairy
## 27 27 3.341098e+12   3.62 2023-01-03 09:42:21     bakery
## 28 28 7.666359e+11   3.01 2023-01-01 08:33:37     bakery
## 29 29 1.170285e+12   3.48 2023-01-01 00:41:34     bakery
## 30 30 6.963387e+12   3.66 2023-01-01 18:22:27  vegetable
## 31 31 5.357547e+12   3.07 2023-01-01 12:55:49  vegetable
## 32 32 7.907325e+12   3.74 2023-01-01 11:47:16     bakery
## 33 33 3.465971e+11   2.86 2023-01-01 12:38:00  vegetable
## 34 34 1.412567e+12   2.25 2023-01-01 11:33:49      dairy
## 35 35 7.985184e+12   3.81 2023-01-01 03:29:11     bakery
## 36 36 3.841253e+12   3.29 2023-01-01 21:40:04     bakery
## 37 37 2.982705e+12   2.43 2023-01-02 04:00:19      dairy
## 38 38 2.719510e+12   2.61 2023-01-02 19:12:06     bakery
## 39 39 2.671835e+12   2.98 2023-01-02 01:23:00     bakery
## 40 40 4.678310e+12   2.09 2023-01-02 21:17:51  vegetable
## 41 41 8.400929e+12   2.47 2023-01-02 16:45:14  vegetable
## 42 42 5.639689e+12   3.29 2023-01-02 13:05:53  vegetable
## 43 43 9.076814e+12   3.47 2023-01-02 23:26:15  vegetable
## 44 44 2.460943e+12   3.65 2023-01-02 04:44:47      dairy
## 45 45 8.760338e+12   2.52 2023-01-02 16:34:17     bakery
## 46 46 2.204060e+11   1.80 2023-01-03 20:56:43     bakery
## 47 47 6.421009e+12   1.63 2023-01-03 04:04:37      dairy
## 48 48 7.387788e+12   2.72 2023-01-03 06:51:00     bakery
## 49 49 1.112442e+12   3.39 2023-01-03 11:28:08     bakery
## 50 50 5.703557e+12   3.09 2023-01-03 02:44:11  vegetable
## 51 51 6.858847e+12   3.72 2023-01-03 19:33:05      dairy
## 52 52 5.374607e+12   3.65 2023-01-03 16:51:00  vegetable
## 53 53 7.381889e+12   2.06 2023-01-03 18:39:41  vegetable
## 54 54 8.664651e+12   2.53 2023-01-03 22:35:03     bakery
```

Assignment 2

Filter the data such that it only contains transactions for 1/2/2023

```
a = which(startsWith(data_1$purchase_timestamp, '2023-01-02'))
data_new = data_1[a, ]
data_new
```

```
##    id      barcode amount   purchase_timestamp      type
## 10 10 7.690345e+12   3.32 2023-01-02 16:21:08    bakery
## 11 11 1.643365e+12   2.61 2023-01-02 08:10:58    bakery
## 12 12 7.539630e+12   2.39 2023-01-02 08:55:13    bakery
## 13 13 4.005177e+12   1.69 2023-01-02 10:11:54 vegetable
## 14 14 2.832167e+11   2.59 2023-01-02 10:20:31     dairy
## 15 15 1.204563e+11   2.81 2023-01-02 18:32:20 vegetable
## 16 16 4.203182e+12   2.62 2023-01-02 18:23:10     dairy
## 17 17 9.256742e+12   2.01 2023-01-02 19:41:00 vegetable
## 18 18 9.603244e+12   1.73 2023-01-02 14:16:08 vegetable
## 37 37 2.982705e+12   2.43 2023-01-02 04:00:19     dairy
## 38 38 2.719510e+12   2.61 2023-01-02 19:12:06    bakery
## 39 39 2.671835e+12   2.98 2023-01-02 01:23:00    bakery
## 40 40 4.678310e+12   2.09 2023-01-02 21:17:51 vegetable
## 41 41 8.400929e+12   2.47 2023-01-02 16:45:14 vegetable
## 42 42 5.639689e+12   3.29 2023-01-02 13:05:53 vegetable
## 43 43 9.076814e+12   3.47 2023-01-02 23:26:15 vegetable
## 44 44 2.460943e+12   3.65 2023-01-02 04:44:47     dairy
## 45 45 8.760338e+12   2.52 2023-01-02 16:34:17    bakery
```

Assignment 3

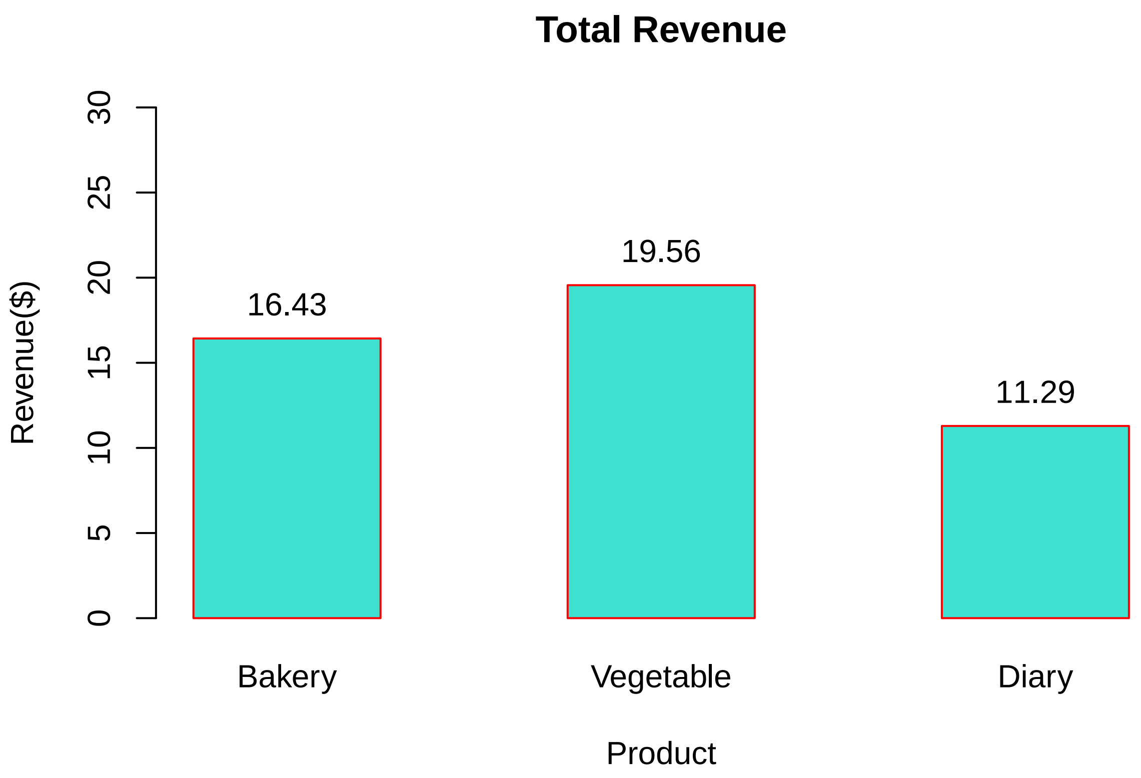
3.1) Bar chart for total revenue in each product line on 1/2

```
# install.packages("tidyverse")
library("tidyverse")
data_new %>% group_by(type) %>% summarise(total_revenue = sum(amount))
```

```
## # A tibble: 3 x 2
##   type      total_revenue
##   <list>            <dbl>
## 1 <chr [1]>          16.4
## 2 <chr [1]>          19.6
## 3 <chr [1]>          11.3
```

```
x1 = c("Bakery", "Vegetable", "Diary")
y1 = c(16.43, 19.56, 11.29)
total_revenue = data.frame(x1, y1)

bar = barplot(height = total_revenue$y1, names = total_revenue$x1, xlab = "Product", ylab = "Revenue($)
      main = "Total Revenue", border = "red", ylim=c(0,30), space = 1, width = 1)
text(bar, total_revenue$y1 + 2, paste(total_revenue$y1, sep=""), cex = 1)
```



3.2) Histogram for the number of items purchased for each hour on 1/2

```
# install.packages("dplyr")
# install.packages("ggplot2")
library(dplyr)
hour = format(as.POSIXct(data_new$purchase_timestamp), format = "%H") #filter out the hour first
data_new$hour = hour
number_of_purchase = data_new %>% group_by(hour) %>% summarise(number_of_purchase = n_distinct(id))
number_of_purchase$hour = sub("^0+", "", number_of_purchase$hour) # get rid of situation like "01", "04
number_of_purchase
```

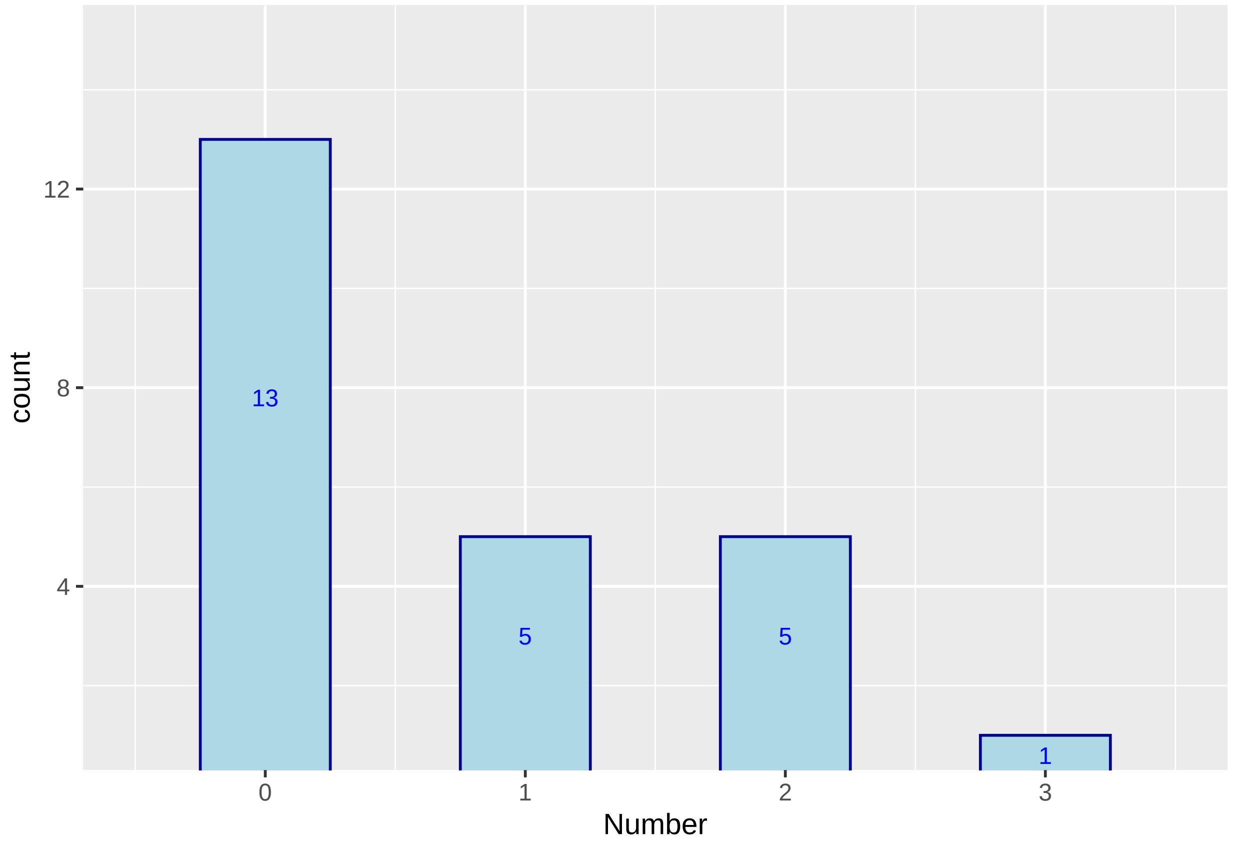
```
## # A tibble: 11 x 2
##    hour  number_of_purchase
##    <chr>              <int>
##  1 1                      1
##  2 4                      2
##  3 8                      2
##  4 10                     2
##  5 13                     1
##  6 14                     1
##  7 16                     3
##  8 18                     2
##  9 19                     2
## 10 21                     1
## 11 23                    1
```

```
count_2 = 1
time1 = list() #create a list for output
x2 = c(0:23)
for (y in x2){
  if (y %in% number_of_purchase$hour)
    time1[count_2] = print(number_of_purchase[which(number_of_purchase$hour == y), 2])
  else
    time1[count_2] = print(0)
count_2 = count_2 + 1
}
```

```
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  1
## [1] 0
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  2
## [1] 0
## [1] 0
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  2
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  2
## [1] 0
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  1
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  1
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                <int>
## 1                  3
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                  <int>
## 1                    2
## # A tibble: 1 x 1
##   number_of_purchase
##                  <int>
## 1                    2
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                  <int>
## 1                    1
## [1] 0
## # A tibble: 1 x 1
##   number_of_purchase
##                  <int>
## 1                    1
```

```
data_2 = data.frame(x2, unlist(time1))
colnames(data_2) <- c('Hour','Number') #clean new data
data_2[, 2] = as.integer(data_2[, 2])

library(ggplot2)
ggplot(data_2, aes(x = Number)) + geom_histogram(binwidth = 0.5, color="darkblue", fill="lightblue") +
  stat_bin(binwidth = 1, geom = 'text', color = 'blue', size = 3, aes(label = ..count..), position = pos
```


count
Number