Assignment 1_tinytex

Quang Dong Nguyen

2023-05-16

Setting up the working directory and files used in this analysis:

```
setwd("C:/Users/Dell/Documents/WSU RStudio/Semester 1 - Analytics Programming/Assignment 1 (40%)")
a <- read.csv("sales_ug.csv") #daily sales data over seven day period
b <- read.csv("product_hierarchy.csv") #data containing the hierarchy and sizes of product
d <- read.csv("store_cities.csv") #data containing the city, type and size information of the stores
```

Library packages used in the report:

```
library(tinytex)
library(tidyverse)
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.2.3
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 4.2.3
```

Task 1

Viewing the overall information about the dataset a (daily sales record of data over a seven day period)

```
#viewing the dataset
head(a, 10) #head(..., 10) shows the first 10 rows of dataset a
```

```
##    product_id store_id       date sales revenue stock  price promo_type_1
## 1       P0001    S0002 2017-07-03     0       0     1   6.75         PR14
## 2       P0001    S0038 2017-07-03     0       0     1   6.75         PR14
## 3       P0001    S0040 2017-07-03     0       0     2   6.75         PR14
## 4       P0001    S0050 2017-07-03     0       0     1   6.75         PR14
## 5       P0001    S0103 2017-07-03     0       0    10   6.75         PR14
## 6       P0001    S0105 2017-07-03     0       0     5   6.75         PR14
## 7       P0002    S0038 2017-07-03     0       0    24 349.00         PR14
## 8       P0002    S0085 2017-07-03     0       0    25 349.00         PR14
## 9       P0004    S0085 2017-07-03     0       0     7   4.50         PR14
## 10      P0005    S0001 2017-07-03     0       0     3  33.90         PR14
##    promo_bin_1 promo_discount_2 promo_discount_type_2
```

```
## 1                              NA                    NA
## 2                              NA                    NA
## 3                              NA                    NA
## 4                              NA                    NA
## 5                              NA                    NA
## 6                              NA                    NA
## 7                              NA                    NA
## 8                              NA                    NA
## 9                              NA                    NA
## 10                             NA                    NA
```

```
#structure of the dataset
str(a) #show the type of data of the variables
```

```
## 'data.frame':    104000 obs. of  11 variables:
##  $ product_id           : chr  "P0001" "P0001" "P0001" "P0001" ...
##  $ store_id             : chr  "S0002" "S0038" "S0040" "S0050" ...
##  $ date                 : chr  "2017-07-03" "2017-07-03" "2017-07-03" "2017-07-03" ...
##  $ sales                : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ revenue              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ stock                : num  1 1 2 1 10 5 24 25 7 3 ...
##  $ price                : num  6.75 6.75 6.75 6.75 6.75 6.75 349 349 4.5 33.9 ...
##  $ promo_type_1         : chr  "PR14" "PR14" "PR14" "PR14" ...
##  $ promo_bin_1          : chr  "" "" "" "" ...
##  $ promo_discount_2     : logi  NA NA NA NA NA NA ...
##  $ promo_discount_type_2: logi  NA NA NA NA NA NA ...
```

1) Total revenue of each store at the end of each day

To calculate the revenue of each store at the end of each day, using `aggregate()` is the best choice of algorithm, as it can split data into subsets and compute summary statistics for each.
The function below summarise the statistic of revenue based on the store_id and date variables. In this case, it sums the total revenue made based on the store_id and date.

```
revenue_each_day <- aggregate(revenue ~ store_id + date, #calculate revenue based on store_id and date
                              data = a,
                              FUN = sum) #summation is abbreviated to sum
head(revenue_each_day, 10)
```

```
##    store_id       date revenue
## 1     S0001 2017-07-03  767.99
## 2     S0002 2017-07-03  346.82
## 3     S0003 2017-07-03   94.43
## 4     S0004 2017-07-03  461.42
## 5     S0006 2017-07-03   56.45
## 6     S0008 2017-07-03  221.52
## 7     S0009 2017-07-03   19.50
## 8     S0010 2017-07-03  255.77
## 9     S0011 2017-07-03  102.58
## 10    S0012 2017-07-03  216.28
```

The above table demonstrates the total revenue of each store profited by the end of each day, starting from date 3 June to 9 June of 2017.

The stores are shown by `store_id` while the `date` shows the days for which the `revenue` is shown. For example:
1. Store with unique identifier number of S0001 obtained a total revenue of 767.99 on the date 2017-07-03.
2. Store with unique identifier number of S0002 obtained a total revenue of 346.82 on the date 2017-07-03.
3. Store with unique identifier number of S0115 obtained a total revenue of 908.29 on the date 2017-07-03.
And so on.

2) Differences in revenues between the day?

To see the difference in revenues between the day, we can use `tapply()` to provide mathematical function to columns that use the function. In this example, `diff` is a function value that is used to calculated the differences in revenues obtained between each row where store_id is matched with the previous row.

```
tapply(revenue_each_day$revenue,
       revenue_each_day$store_id,
       diff) %>% #each array element represents the difference in revenue between
  head(10)       #the current day and the next day
```

```
## $S0001
## [1]  528.37 -290.51 -112.30  354.33  299.45  -82.10
##
## $S0002
## [1] -120.64  -50.70   87.11 -121.13  444.79 -202.29
##
## $S0003
## [1]  27.28  -9.50 -71.73  55.07 -35.48  19.24
##
## $S0004
## [1] -324.83   -9.83  -14.94   29.68  182.01 -156.84
##
## $S0006
## [1] -29.64  43.70  -1.36 -21.83 -11.78  -6.33
##
## $S0008
## [1] -27.40 -87.07 100.93  57.08 -15.42 -55.36
##
## $S0009
## [1]  -3.02  38.41 -10.17 -19.56  10.57  37.89
##
## $S0010
## [1]   9.11 -87.39 -10.11  74.18 173.72 131.48
##
## $S0011
## [1]  16.62  16.72 -15.13  -7.99 -59.78  34.35
##
## $S0012
## [1] -115.96   39.98    5.28  -44.74  188.43 -150.29
```

In this example, `tapply()` returns values in the form of arrays. It is a poor way to arrange data, however this is the only current available option for my personal choice of algorithm.

```
class(tapply(revenue_each_day$revenue, revenue_each_day$store_id, diff))
```

```
## [1] "array"
```

```
#returns values in the form of arrays.
```

3) Total revenue generated by each store over seven days

```
revenue_seven <- aggregate(revenue ~ store_id,
                           data = a,
                           FUN = sum) #summarise the total revenue made from each store_id over the sev
                                      #function applied to summarise the revenue statistic is sum (summ
head(revenue_seven, 10)
```

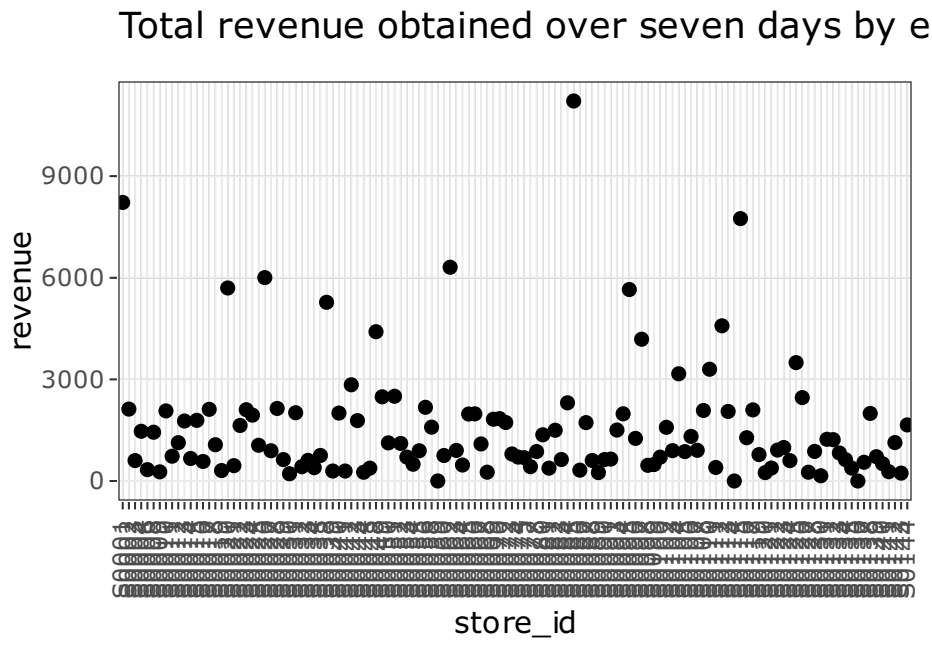
```
##    store_id revenue
## 1     S0001 8224.19
## 2     S0002 2122.74
## 3     S0003  603.76
## 4     S0004 1468.27
## 5     S0006  334.99
## 6     S0008 1439.65
## 7     S0009  270.10
## 8     S0010 2069.12
## 9     S0011  731.68
## 10    S0012 1131.57
```

The above table portrays the first 10 values of the total revenue of each store over the seven day period. For example:
1. Store with `store_id` (unique identifier number) of S0001 has gained a total revenue of 8224.19.
2. Store with `store_id` of S0002 has gained a total revenue of 2122.74.
3. Store with `store_id` of S0056 has gained a total revenue of 2175.47. And so on

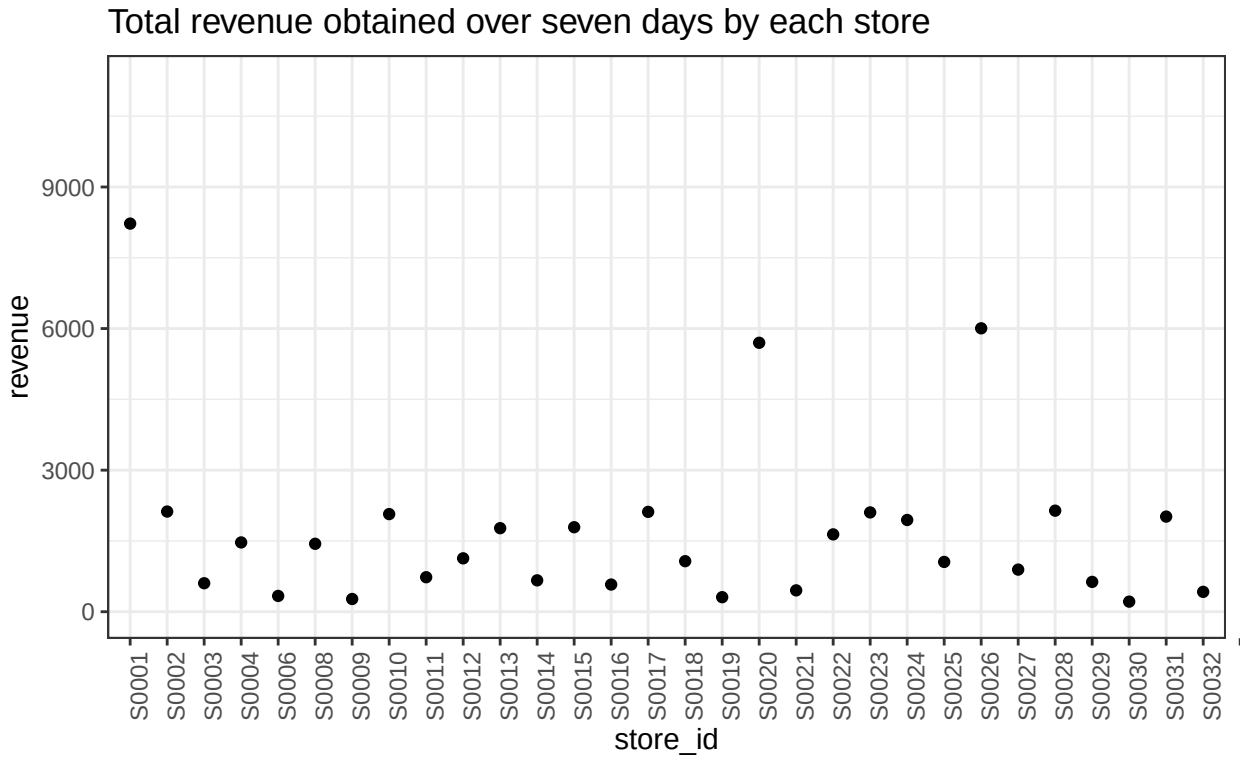
Plotting:

```
#ggplot the whole graph with everystore and its total revenue over seven days
ggstoreid_rev <- ggplot(revenue_seven, aes(store_id, revenue)) + #aesthetic mapping x and y-axis with s
  geom_point() +                                  #create points with x as store_id and y as revenue
  theme_bw() +                                    #change the background theme of the graph to white
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "Total revenue obtained over seven days by each store",
       caption = "*Note: the ggplot shows the ")

ggplotly(ggstoreid_rev)
```




```
#plotting the total revenue over the seven day period
ggplot(revenue_seven, aes(store_id, revenue)) +
  geom_point() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90)) +
  coord_cartesian(xlim = c(1, 30)) + #showing the revenues obtained by the first 30 stores
  labs(title = "Total revenue obtained over seven days by each store",
       caption = "The plot shows only the first 30 stores' revenues due to overloading of data.
       Note: revenue - daily total sales revenue
             store_id - unique identifier of a store")
```



The plot shows only the first 30 stores' revenues due to overloading of data.
Note: revenue − daily total sales revenue
store_id − unique identifier of a store

Task 2:

Viewing information about the dataset b (product_hierarchy data)

```
#viewing the dataset
head(b, 10) #shows the first 10 variables of dataset b
```

```
##    product_id product_length product_depth product_width cluster_id
## 1       P0000            5.0            20          12.0
## 2       P0001           13.5            22          20.0  cluster_5
## 3       P0002           22.0            40          22.0  cluster_0
## 4       P0004            2.0            13           4.0  cluster_3
## 5       P0005           16.0            30          16.0  cluster_9
## 6       P0006            8.5            15          15.0  cluster_0
## 7       P0007            2.0            22           9.5  cluster_4
## 8       P0008            5.0            16           5.0  cluster_0
## 9       P0009            5.0            18          14.0  cluster_6
## 10      P0010            2.0            22           3.0  cluster_0
##    hierarchy1_id hierarchy2_id hierarchy3_id hierarchy4_id hierarchy5_id
## 1            H00         H0004       H000401     H00040105   H0004010534
## 2            H01         H0105       H010501     H01050100   H0105010006
## 3            H03         H0315       H031508     H03150800   H0315080028
## 4            H03         H0314       H031405     H03140500   H0314050003
## 5            H03         H0312       H031211     H03121109   H0312110917
```

```
## 6            H03         H0316       H031608     H03160817   H0316081708
## 7            H03         H0313       H031305     H03130519   H0313051904
## 8            H00         H0000       H000004     H00000400   H0000040017
## 9            H00         H0002       H000201     H00020100   H0002010012
## 10           H01         H0108       H010801     H01080109   H0108010917
```

```
#structure of the dataset
str(b) #shows the structure of b and its datax
```

```
## 'data.frame':    699 obs. of  10 variables:
##  $ product_id    : chr  "P0000" "P0001" "P0002" "P0004" ...
##  $ product_length: num  5 13.5 22 2 16 8.5 2 5 5 2 ...
##  $ product_depth : num  20 22 40 13 30 15 22 16 18 22 ...
##  $ product_width : num  12 20 22 4 16 15 9.5 5 14 3 ...
##  $ cluster_id    : chr  "" "cluster_5" "cluster_0" "cluster_3" ...
##  $ hierarchy1_id : chr  "H00" "H01" "H03" "H03" ...
##  $ hierarchy2_id : chr  "H0004" "H0105" "H0315" "H0314" ...
##  $ hierarchy3_id : chr  "H000401" "H010501" "H031508" "H031405" ...
##  $ hierarchy4_id : chr  "H00040105" "H01050100" "H03150800" "H03140500" ...
##  $ hierarchy5_id : chr  "H0004010534" "H0105010006" "H0315080028" "H0314050003" ...
```

1) The most popular product type (hierarchy 1) sold in all stores over a week

To check for the popularity ranking of the product type (hierarchy 1) in terms of selling, we use `sort()` to sort table values. By using `decreasing = TRUE` as additional argument, it sorts table values from the highest to the lowest.

```
sort(table(b$hierarchy1_id), decreasing = TRUE)#product named H03 are most popularly sold
```

```
##
## H03 H00 H01 H02
## 292 215 181  11
```

As it can be seen in the above table, the most sold product type is H03 with 292 items sold over the week. And the second most popular product type sold is H00 with 215 items sold over the week.

2) How much revenue did the stores receive for that product during the week?

Joining two datasets a and b based on their corresponding variables. In this case the corresponding keys are product_id, and the joined variables are hierarchy1_id and hierarchy2_id

```
merged_ab_tab <- b %>%
  select("product_id", "hierarchy1_id", "hierarchy2_id") %>%
  right_join(a)
```

```
## Joining, by = "product_id"
```

```
head(merged_ab_tab, 10)
```

```
##    product_id hierarchy1_id hierarchy2_id store_id       date sales revenue
## 1       P0001           H01         H0105    S0002 2017-07-03     0       0
## 2       P0001           H01         H0105    S0038 2017-07-03     0       0
## 3       P0001           H01         H0105    S0040 2017-07-03     0       0
## 4       P0001           H01         H0105    S0050 2017-07-03     0       0
## 5       P0001           H01         H0105    S0103 2017-07-03     0       0
## 6       P0001           H01         H0105    S0105 2017-07-03     0       0
## 7       P0001           H01         H0105    S0002 2017-07-04     0       0
## 8       P0001           H01         H0105    S0038 2017-07-04     0       0
## 9       P0001           H01         H0105    S0040 2017-07-04     0       0
## 10      P0001           H01         H0105    S0050 2017-07-04     0       0
##    stock price promo_type_1 promo_bin_1 promo_discount_2 promo_discount_type_2
## 1      1  6.75         PR14                         NA                    NA
## 2      1  6.75         PR14                         NA                    NA
## 3      2  6.75         PR14                         NA                    NA
## 4      1  6.75         PR14                         NA                    NA
## 5     10  6.75         PR14                         NA                    NA
## 6      5  6.75         PR14                         NA                    NA
## 7      1  6.75         PR14                         NA                    NA
## 8      1  6.75         PR14                         NA                    NA
## 9      2  6.75         PR14                         NA                    NA
## 10     1  6.75         PR14                         NA                    NA
```

Revenue received from that product during the week:

```
#revenue made
stores_rev_made <- merged_ab_tab[which(merged_ab_tab$hierarchy1_id == "H03"),]
aggregate(revenue ~ store_id + date, data = stores_rev_made, sum) %>%
  head(10) #shows the first 10 values of revenues made from products with hierarchy1_id of "H03"
```

```
##    store_id       date revenue
## 1     S0001 2017-07-03  268.05
## 2     S0002 2017-07-03   70.87
## 3     S0003 2017-07-03    9.25
## 4     S0004 2017-07-03   21.98
## 5     S0006 2017-07-03   38.54
## 6     S0008 2017-07-03   27.82
## 7     S0009 2017-07-03    0.00
## 8     S0010 2017-07-03    5.50
## 9     S0011 2017-07-03    9.21
## 10    S0012 2017-07-03   22.53
```

As shown in the table above, Each store has received a various amount of revenue over each days. For instance, Store with the store_id of S0001 has made a total of 268.05 on the date of 3/7/2017. While store with the store_id of S0003 has only made a total of 9.25 on the date of 3/7/2017 on the same product as the store with store_id of S0001.

Furthermore, there are also stores that made zero revenue on some days, for example, store with the store_id of S0009 has made zero revenue on that product on the date of 3/7/2017.

Therefore, the revenues generated from each store are unique.

3) How does that compare with the second most popular product?

The second most popular product is "H00" according to the sorted table above.

```
stores_rev_made <- merged_ab_tab[which(merged_ab_tab$hierarchy1_id == "H00"),]
aggregate(revenue ~ store_id + date, data = stores_rev_made, sum) %>%
  head(10) #total revenue made in each store from the products with hierarchy1_id "H00" during the week
```

```
##    store_id       date revenue
## 1     S0001 2017-07-03  315.09
## 2     S0002 2017-07-03  210.99
## 3     S0003 2017-07-03   85.18
## 4     S0004 2017-07-03  397.83
## 5     S0006 2017-07-03   17.91
## 6     S0008 2017-07-03  117.56
## 7     S0009 2017-07-03   19.50
## 8     S0010 2017-07-03   85.05
## 9     S0011 2017-07-03   74.53
## 10    S0012 2017-07-03  110.24
```

needs to be fixed

4) Provide a table showing the product type ranked from most to least popular

```
sort(table(b$hierarchy1_id), decreasing = TRUE)
```

```
##
## H03 H00 H01 H02
## 292 215 181  11
```

The table above shows the ranking of product type from most to least, where the most and least popular product types are H03 and H02.

5) For each product: how many subtypes products are there?

```
matx_1 <- table(b$hierarchy1_id, b$hierarchy2_id)
matx_1
```

```
##
##       H0000 H0001 H0002 H0003 H0004 H0105 H0106 H0107 H0108 H0209 H0210 H0311
##   H00    32    38    54    53    38     0     0     0     0     0     0     0
##   H01     0     0     0     0     0    17    28    40    96     0     0     0
##   H02     0     0     0     0     0     0     0     0     0     4     7     0
##   H03     0     0     0     0     0     0     0     0     0     0     0    51
##
##       H0312 H0313 H0314 H0315 H0316 H0317
##   H00     0     0     0     0     0     0
##   H01     0     0     0     0     0     0
##   H02     0     0     0     0     0     0
##   H03    61   101    28    40     5     6
```

As described in the description of variables, each product has subtype products corresponded to and is categorised into levels of hierarchy. According to the hierarchy table shown above:
- There are 5 subtype products of H00: H0000, H0001, H0002, H0003, H0004. - There are 4 subtype products of H01: H0105, H0106, H0107, H0108.
- There are 2 subtype products of H02: H0209, H0311.
- There are 7 subtype products of H03: H0311, H0312, H0313, H0314, H0315, H0316, H0317.

6) How many products are in this product type?

As shown in the matrix table `matx_1` above:
- There are 32 items in H0000 (subset of H00).
- There are 38 items in H0001 (subset of H00). - And so on.

7) Sales quantity:

```
#hierarchy1_id:
aggregate(sales ~ hierarchy1_id, data = merged_ab_tab, sum)
```

```
##   hierarchy1_id     sales
## 1           H00 40256.818
## 2           H01  5797.000
## 3           H02  1141.983
## 4           H03  4266.000
```

There are four product types, and each made a unique number of sales over the seven days:
- H00 has made a total sale of H01.
- H01 has made a total sale of 5797.
- H02 has made a total sale of . - H03 has made a total sale of .

```
#hierarchy2_id:
aggregate(sales ~ hierarchy1_id + hierarchy2_id, data = merged_ab_tab, sum) %>%
  head(10) #shows the first 10 values of sale obtained
```

```
##    hierarchy1_id hierarchy2_id     sales
## 1            H00         H0000 13093.000
## 2            H00         H0001  2481.000
## 3            H00         H0002  2955.000
## 4            H00         H0003 17920.000
## 5            H00         H0004  3807.818
## 6            H01         H0105   787.000
## 7            H01         H0106  1888.000
## 8            H01         H0107  1438.000
## 9            H01         H0108  1684.000
## 10           H02         H0209  1133.513
```

Total sale made based on the second level of hierarchy (hierarchy2_id). For instance:
- In a week, the total sale produced by selling products where the first level of hierarchy is H00 and the second level of hierarchy is H0000 was 13093.000.
- Meanwhile, the total sale produced by selling products where the first hierarchy level is H00 and the second hierarchy level is H0001 was 2481.000.

Insight:
The most popular subtype of H00 sold in all stores is H0003 with a total sale of 17,920.000 made over the seven days. And the second most popular subtype of H00 sold in all stores is H0000 with a total sale of 13,093.000 made over the seven days.

8) Revenue generated by each product type:

```
#hierarchy1_id
aggregate(revenue ~ hierarchy1_id, data = merged_ab_tab, sum)
```

```
##   hierarchy1_id   revenue
## 1           H00 100165.44
## 2           H01  61773.15
## 3           H02  12221.22
## 4           H03  25377.66
```

The total revenue obtained by each product type over the seven day period shows that:
- The top ranked product type is H00,which has obtained a total revenue of $100,165.44 over seven days. - Meanwhile, the second-ranked product type is H01,which has obtained a total revenue of $61,773.15.
- And, the last ranked product type is H02,which has obtained a total revenue of $12,221.22.

```
#hierarchy2_id:
aggregate(revenue ~ hierarchy1_id + hierarchy2_id, data = merged_ab_tab, sum) %>%
  head(10)
```

```
##    hierarchy1_id hierarchy2_id  revenue
## 1            H00         H0000 35413.54
## 2            H00         H0001  9207.45
## 3            H00         H0002 11134.93
## 4            H00         H0003 24249.76
## 5            H00         H0004 20159.76
## 6            H01         H0105  7698.96
## 7            H01         H0106 21503.25
## 8            H01         H0107 16386.22
## 9            H01         H0108 16184.72
## 10           H02         H0209 12180.40
```

Total revenue made asaed on the second level of hierarchy (hierarchy2_id).
- The most sold item in H00 is H0000 with a total of $35,413.54 made over the week.
- And the least sold item in H00 is H0001, with a total of $9,207.45 made over the week.

Task 3:

View information about the dataset d (store_cities data)

```
#Viewing the first 10 values of the dataset
head(d, 10)
```

```
##    store_id storetype_id store_size city_id
## 1     S0091         ST04         19    C013
## 2     S0012         ST04         28    C005
## 3     S0045         ST04         17    C008
## 4     S0032         ST03         14    C019
## 5     S0027         ST04         24    C022
## 6     S0088         ST04         20    C009
## 7     S0095         ST02         44    C014
## 8     S0055         ST04         24    C014
## 9     S0099         ST03         14    C014
## 10    S0078         ST04         19    C036
```

```
#structure of the dataset
str(d)
```

```
## 'data.frame':    144 obs. of  4 variables:
##  $ store_id    : chr  "S0091" "S0012" "S0045" "S0032" ...
##  $ storetype_id: chr  "ST04" "ST04" "ST04" "ST03" ...
##  $ store_size  : int  19 28 17 14 24 20 44 24 14 19 ...
##  $ city_id     : chr  "C013" "C005" "C008" "C019" ...
```

Compare the Sales volumes between the two most common store types in the data set.

Sorting store types accross the stores cities data set:

```
sort(table(d$storetype_id), decreasing = TRUE)
```

```
##
## ST04 ST03 ST01 ST02
##   83   53    4    4
```

Ranking from most to least, there are:
- ST04 is the most common storetype with over 83 stores accross cities. - ST02 and ST01 are the least common storetypes accross cities, with only 4 stores for each.

Joining two datasets a and d together

```
#right join dataset d and a according to the corresponding id:
merged_da_tab <- d %>%
  select("store_id", "storetype_id", "store_size") %>%
  right_join(a)
```

```
## Joining, by = "store_id"
```

```
head(merged_da_tab, 10)
```

```
##    store_id storetype_id store_size product_id       date sales revenue stock
## 1     S0091         ST04         19      P0015 2017-07-03     0       0     6
## 2     S0091         ST04         19      P0017 2017-07-03     0       0    20
## 3     S0091         ST04         19      P0035 2017-07-03     0       0     3
## 4     S0091         ST04         19      P0042 2017-07-03     0       0     5
```

```
## 5     S0091         ST04         19      P0046 2017-07-03     0       0     7
## 6     S0091         ST04         19      P0051 2017-07-03     0       0    22
## 7     S0091         ST04         19      P0054 2017-07-03     0       0     6
## 8     S0091         ST04         19      P0055 2017-07-03     0       0    12
## 9     S0091         ST04         19      P0057 2017-07-03     0       0     6
## 10    S0091         ST04         19      P0067 2017-07-03     0       0     4
##    price promo_type_1 promo_bin_1 promo_discount_2 promo_discount_type_2
## 1   2.85         PR14                          NA                    NA
## 2   1.49         PR12    veryhigh              NA                    NA
## 3   4.25         PR14                          NA                    NA
## 4   5.50         PR14                          NA                    NA
## 5  34.50         PR14                          NA                    NA
## 6   0.70         PR14                          NA                    NA
## 7   3.95         PR14                          NA                    NA
## 8   3.50         PR14                          NA                    NA
## 9  14.90         PR14                          NA                    NA
## 10 16.90         PR14                          NA                    NA
```

```
#sales volume of ST03 and ST04
aggregate(sales ~ storetype_id, data = merged_da_tab, sum)[c(3,4),]
```

```
##   storetype_id     sales
## 3         ST03  7980.007
## 4         ST04 35566.554
```

In terms of sales, Stores with storetype_id ST03 has gained a total of 7980 in sale volume while stores with the store_id ST04 has gained a total of 35,556 in sale volume over the seven days. This means that stores with the storetype_id ST04 is more potential than the other, since the difference in the volume of sale made over a week is at least 5 times over the other.

```
#difference in sales volume between ST04 and ST03
```

How do they compare in terms of total revenue?

```
#Total revenue of ST03 and ST04
aggregate(revenue ~ storetype_id, data = merged_da_tab, sum)[c(3,4),]
```

```
##   storetype_id   revenue
## 3         ST03  21776.75
## 4         ST04 144628.73
```

Is there a relationship betwen a store's size and its revenue?

```
summary(lm(revenue~store_size, data = merged_da_tab))
```

```
##
## Call:
## lm(formula = revenue ~ store_size, data = merged_da_tab)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5.13  -2.27  -1.52  -0.98 838.99
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.042201   0.070331   -0.60    0.548
## store_size   0.067889   0.002224   30.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.244 on 103998 degrees of freedom
## Multiple R-squared:  0.008883,   Adjusted R-squared:  0.008874
## F-statistic: 932.1 on 1 and 103998 DF,  p-value: < 2.2e-16
```

Task 4:

For each promotion type, display the different levels of promotion during the period

```
#Different levels of promotion
table(a$promo_type_1, a$promo_bin_1)
```

```
##
##              high   low moderate veryhigh verylow
##   PR03     0    0     0        0        0     286
##   PR05     0  123   744       14        0     240
##   PR06     0    0   175        0        0     481
##   PR08     0    0     0        0      126       0
##   PR09     0  190  1638        0        0       0
##   PR10     0    0     0        0        0      58
##   PR12     0    0     0        0     3196    1804
##   PR13     0    0     0        0        0      26
##   PR14 94899    0     0        0        0       0
```

Each promotion type has a unique level of ranking, from very high to very low. Except PR14, it has one level of promotion and is not categorised to any level of ranking (high-to-low).

```
#Uses of promotion accross the seven day period
table(a$date, a$promo_type_1)
```

```
##
##              PR03  PR05  PR06  PR08  PR09  PR10  PR12  PR13  PR14
##   2017-07-03   52   236    93     0   263     9   704     0 13422
##   2017-07-04   52    85    93     0   262     9   710     0 13616
##   2017-07-05   52    86    95     0   260     8   715     0 13605
##   2017-07-06   52   103    94     0   262     8   716     0 13652
##   2017-07-07   52   104    93     0   260     8   716     0 13668
##   2017-07-08   13   252    94    66   259     8   720    12 13476
##   2017-07-09   13   255    94    60   262     8   719    14 13460
```

However, as it can be seen, the most commonly used promotion accross the seven days was PR14, with more over 13400 promotions were used on each day in every stores accross cities.

Analyse the effectiveness of the promotion on the sales of the products