

DATATHON

Ricardo Torres Arevalo, Jack Landers

2023-02-18

Brainstorm

Questions we will be answering

Which items should the store stop selling? Why?

What was the most profitable month in the dataset?

```
library(tidyverse)
library(scales)
library(tinytex)
store <- read_delim("sales_data_2017_2018_for_tableau_with_new_date_columns.csv")

names(store)
```

```
## [1] "receipt_id"      "date"            "hour"
## [4] "quarter"         "year"            "month_number"
## [7] "month_name"      "day_of_week_name" "week_number"
## [10] "is_weekday"      "is_weekend"      "item_code"
## [13] "item_name"       "main_category"   "sub_category"
## [16] "quantity"        "payment_type"    "unit_buying_price"
## [19] "unit_selling_price" "unit_price_margin" "total_buying_price"
## [22] "total_selling_price" "total_profit"
```

These were the variables that we were working with while we attempted to answer the questions.

Profits and Losses of Store

```
store %>%
group_by(main_category) %>%
summarise(main_cc = length(main_category), profit = sum(total_profit)) %>%
arrange(desc(main_cc))
```

```
## # A tibble: 10 x 3
##   main_category    main_cc  profit
##   <chr>          <int>   <dbl>
## 1 Fresh Produce    333206 643988.
## 2 Pantry Staples   21514   36457.
```

##	3	Snacks	6870	8324.
##	4	Dairy, Cheese, and Eggs	5783	5068.
##	5	Breads & Bakery	2074	2448.
##	6	Beverages	1602	2111.
##	7	Bag	967	1085
##	8	Flowers	676	5844
##	9	Beverage	56	157.
##	10	Miscellaneous	9	24

This table shows the the amount of products bought and their profits. This would be organized by having the products be grouped by their main category in the store.

```
store %>%
  group_by(sub_category) %>%
  filter(total_profit < 0, year == 2017) %>%
  summarise(profit_2017 = sum(total_profit)) %>%
  filter(rank(profit_2017) <= 20) %>%
  arrange(profit_2017)
```

```
## # A tibble: 13 x 2
##   sub_category    profit_2017
##   <chr>          <dbl>
## 1 Cabbages        -274.
## 2 Pears           -259.
## 3 Bunch Vegies    -155.
## 4 Avocadoes       -147.
## 5 Bananas         -138.
## 6 Asian Vegies    -105.
## 7 Root Vegies     -44.1
## 8 Grapes          -36.0
## 9 Tropical Fruits -24.9
## 10 Citrus Fruits  -15.3
## 11 Condiments     -5.4
## 12 Deals          -4.69
## 13 Melons         -2.73
```

This table organized the data set by the store's sub categories and would show what sub categories were losing the store money and by how much it was losing it by in the year 2017.

```
store %>%
  group_by(sub_category) %>%
  filter(total_profit < 0, year == 2018) %>%
  summarise(profit_2018 = sum(total_profit)) %>%
  filter(rank(profit_2018) <= 20) %>%
  arrange(profit_2018)
```

```
## # A tibble: 14 x 2
##   sub_category    profit_2018
##   <chr>          <dbl>
## 1 Cabbages       -1836.
## 2 Avocadoes      -969.
## 3 Asian Vegies   -136.
```

```
## 4 Bunch Vegies      -133
## 5 Bananas           -105.
## 6 Condiments        -34.2
## 7 Citrus Fruits     -25.7
## 8 Tropical Fruits   -21.3
## 9 Deals             -20.8
## 10 Vinegar          -7.74
## 11 Melons           -2.02
## 12 Root Vegies      -1.72
## 13 Apples           -1.24
## 14 Tomatoes         -1
```

This table shows the sub categories that were losing the most amount of money and by how much. This would allow us to see what categories need to be revised in order to see where the store is truly losing their money in.

```
store %>%
  group_by(sub_category) %>%
  filter(year == 2017) %>%
  summarise(amount_2017 = sum(quantity)) %>%
  filter(rank(desc(amount_2017)) <= 15) %>%
  arrange(desc(amount_2017))
```

```
## # A tibble: 15 x 2
##   sub_category amount_2017
##   <chr>         <dbl>
## 1 Bananas      13912.
## 2 Melons       12507.
## 3 Other Vegies 12361.
## 4 Potatoes     10340.
## 5 Citrus Fruits 8386.
## 6 Apples       8367.
## 7 Bunch Vegies 8177.
## 8 Tomatoes     7343.
## 9 Stonefruits  6870.
## 10 Herbs       5258.
## 11 Pumpkins    5020.
## 12 Cucumbers   4845.
## 13 Grapes      4560.
## 14 Deals       4541.
## 15 Lettuces    4459.
```

This shows the most bought sub categories in the store. Items that could be dropped would be Asian Veggies since they are not being sold a lot compared to its other sub categories and the store is paying money to import them which is causing loss in profit in that sub category. Something else we can get rid of are condiments since they are also not selling as much as the other categories and their loss in profit has increased more in 2018 than it has in 2017. Therefore, if the store does not get rid of it, then the loss in profit will only increase over the next couple of years.

```
store %>%
  group_by(sub_category) %>%
  filter(year == 2018) %>%
  summarise(amount_2018 = sum(quantity)) %>%
```

```
filter(rank(desc(amount_2018)) <= 15) %>%
  arrange(desc(amount_2018))
```

```
## # A tibble: 15 x 2
##   sub_category amount_2018
##   <chr>         <dbl>
## 1 Other Vegies    11730.
## 2 Melons          11155.
## 3 Bananas         10540.
## 4 Potatoes        9099.
## 5 Bunch Vegies    8159
## 6 Citrus Fruits   7443.
## 7 Apples          7126.
## 8 Tomatoes        6026.
## 9 Stonefruits     5357.
## 10 Herbs          4951.
## 11 Eggs           4604
## 12 Pumpkins       4185.
## 13 Onions          4041.
## 14 Lettuces        3963.
## 15 Deals           3687
```

This part of the data shows what sub categories customers were buying the most of. This would allow us to see if the sub categories that were losing money were being bought a lot or not. That way we can determine if the products in that sub category are worth selling the store anymore. The data was set to be for 2018.

```
store %>%
  filter(total_profit < 0) %>%
  filter(sub_category == "Cabbages") %>%
  mutate(loss = sum(total_profit), amount = sum(quantity)) %>%
  select(item_name, loss, amount) %>%
  head(1)
```

```
## # A tibble: 1 x 3
##   item_name      loss amount
##   <chr>         <dbl> <dbl>
## 1 Cabbage Wombok -2110.   938.
```

By looking at this data, we determined that the store should stop selling Wombok Cabbages because they are the only item that is causing loss in profit. Meanwhile the rest of the cabbages gain profit after each purchase.

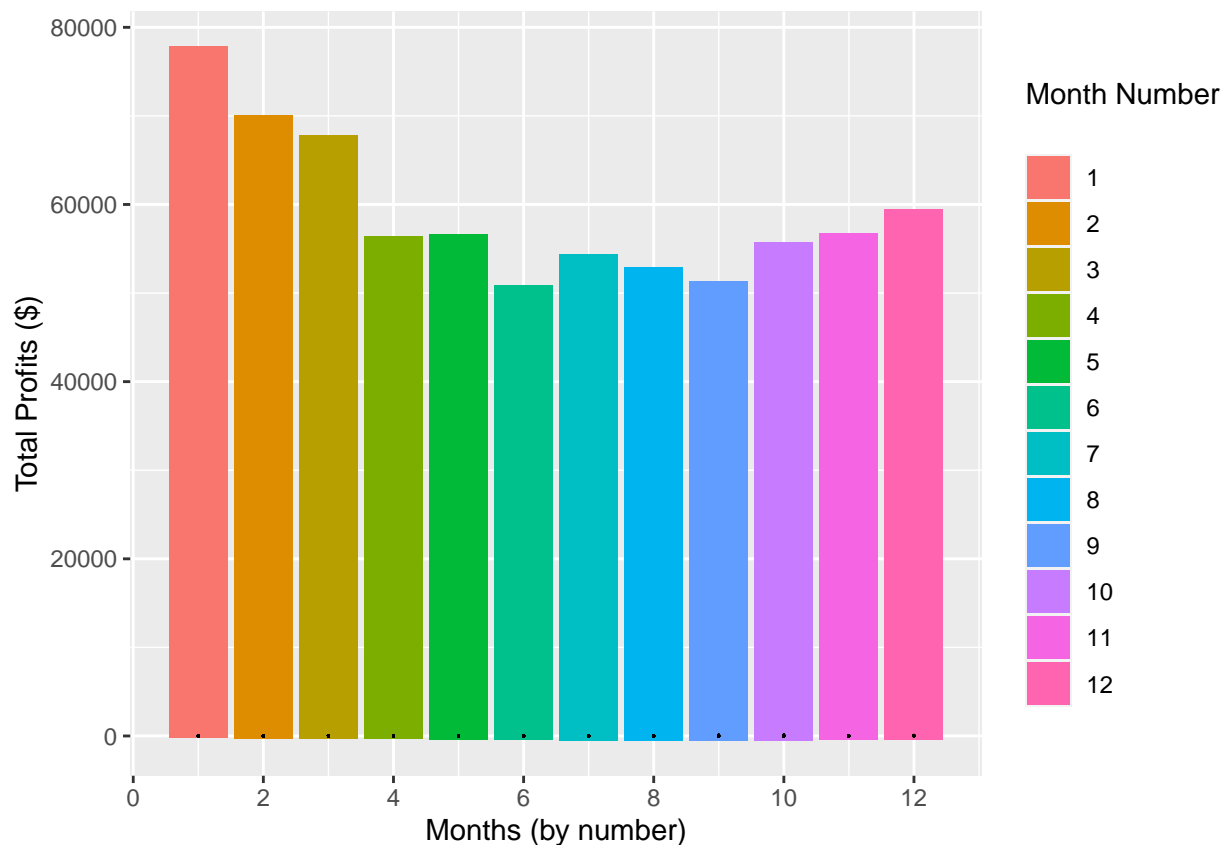
```
store %>%
  filter(total_profit < 0) %>%
  filter(sub_category == "Avocadoes") %>%
  mutate(loss = sum(total_profit), amount = sum(quantity)) %>%
  select(item_name, loss, amount) %>%
  head(1)
```

```
## # A tibble: 1 x 3
##   item_name      loss amount
##   <chr>         <dbl> <dbl>
## 1 Avocado Hass Medium -1116.  1105
```

Another item that the store could get rid of would be Avocado Hass Medium since they are the item that loses the second most amount of profit and it is not sold as much as the rest of the avocados.

Finding the most profitable month by sales

```
store %>%
  ggplot(aes(month_number, total_profit, fill = factor(month_number)))+
  geom_col()+
  geom_line()+
  scale_x_continuous(breaks = pretty_breaks())+
  labs(x = "Months (by number)",
       y = "Total Profits ($)",
       fill = "Month Number\n")
```



Australia has opposite seasons from us, therefore whenever we have our winter season, they have summer. The peak amount of sales for the store occur when Australia is in its summer season. These months are towards the beginning and end of the year.

```
store %>%
  filter(sub_category == "Apples") %>%
  # filter(total_profit > 0) %>%
  arrange(total_profit)
```

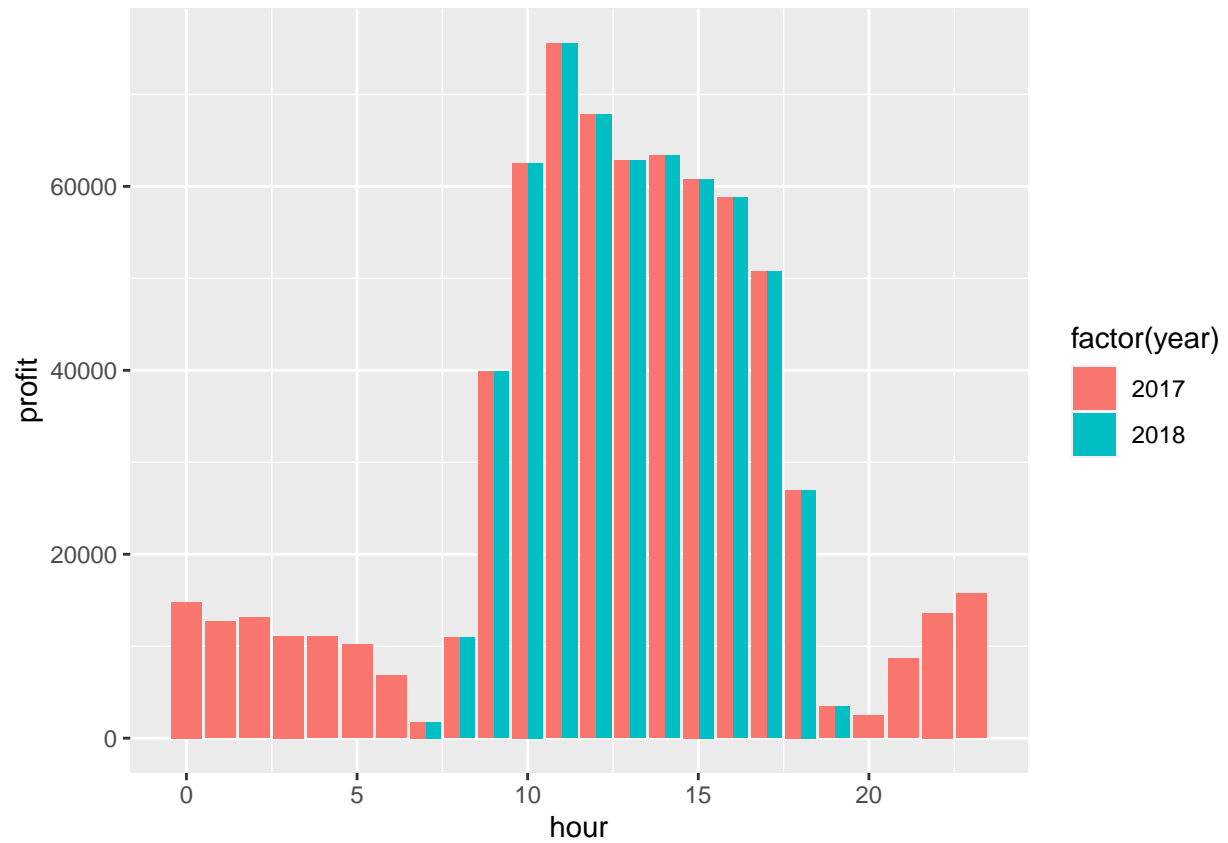
```
## # A tibble: 17,046 x 23
```

```
## receipt_id date hour quarter year month_number month_name day_of_week_name
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1 e07045cf~ 4/9/~ 17 2 2018 4 April Monday
## 2 32410b44~ 4/9/~ 17 2 2018 4 April Monday
## 3 76577560~ 4/7/~ 17 2 2018 4 April Saturday
## 4 f60aa4ff~ 4/22~ 12 2 2017 4 April Saturday
## 5 8cf74f57~ 5/27~ 11 2 2017 5 May Saturday
## 6 92bf4549~ 4/26~ 9 2 2017 4 April Wednesday
## 7 78878cf1~ 4/26~ 10 2 2017 4 April Wednesday
## 8 b1803785~ 5/26~ 11 2 2017 5 May Friday
## 9 85666ed1~ 5/12~ 11 2 2018 5 May Saturday
## 10 99bee43e~ 4/30~ 15 2 2018 4 April Monday
## # i 17,036 more rows
## # i 15 more variables: week_number <dbl>, is_weekday <dbl>, is_weekend <dbl>,
## # item_code <dbl>, item_name <chr>, main_category <chr>, sub_category <chr>,
## # quantity <dbl>, payment_type <chr>, unit_buying_price <dbl>,
## # unit_selling_price <dbl>, unit_price_margin <dbl>,
## # total_buying_price <dbl>, total_selling_price <dbl>, total_profit <dbl>
```

```
# summarise(apple_profit = mean(total_profit))
```

Another item that the store should get rid of is Apple Granny Smith 1kg bags because they barely make any sales and are the only apple product that is making the store lose money. The store sells a lot of apples.

```
store %>%
  group_by(hour) %>%
  mutate(profit = sum(total_profit)) %>%
  ggplot(aes(hour, profit, fill = factor(year)))+
  geom_col(position = "dodge")+
  scale_x_continuous(breaks = pretty_breaks())
```



According to the data, the total profits for the hour in 2017 match the total profits for 2018. However, since 2017 was run 24/7 they generate the greater profit then 2018 because they are not missing out on the money. The only change they can make to improve there margain for profits would be to change the prices on what they are selling, increase costs for customers to buy.