rishabh kansal part 1

rishabh kansal

5/9/2022

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
data = read.csv(file='2019 Winter Data Science Intern Challenge Data Set - Sheet1.csv',header=TRUE)
head(data)
```

```
##   order_id shop_id user_id order_amount total_items payment_method
## 1        1      53     746          224           2           cash
## 2        2      92     925           90           1           cash
## 3        3      44     861          144           1           cash
## 4        4      18     935          156           1    credit_card
## 5        5      18     883          156           1    credit_card
## 6        6      58     882          138           1    credit_card
##            created_at
## 1 2017-03-13 12:36:56
## 2 2017-03-03 17:38:52
## 3  2017-03-14 4:23:56
## 4 2017-03-26 12:43:37
## 5  2017-03-01 4:35:11
## 6 2017-03-14 15:25:01
```

Every store offers only one kind of shoe, which means that the order value just depends on the quantity of shoe ordered from the store. Moreover, each store might be offering a different kind of shoe, some stores may be offering expensive sneakers whereas some might be offering low-end shoes.

```
#What was being done
v1 = data %>%
  group_by(shop_id) %>%
  summarise_at(vars(order_id),
               list(aov = mean))
head(v1)
```

```
## # A tibble: 6 x 2
##   shop_id   aov
##     <int> <dbl>
## 1       1 2515.
## 2       2 2299.
## 3       3 2306.
## 4       4 2748.
## 5       5 2498.
## 6       6 2432.
```

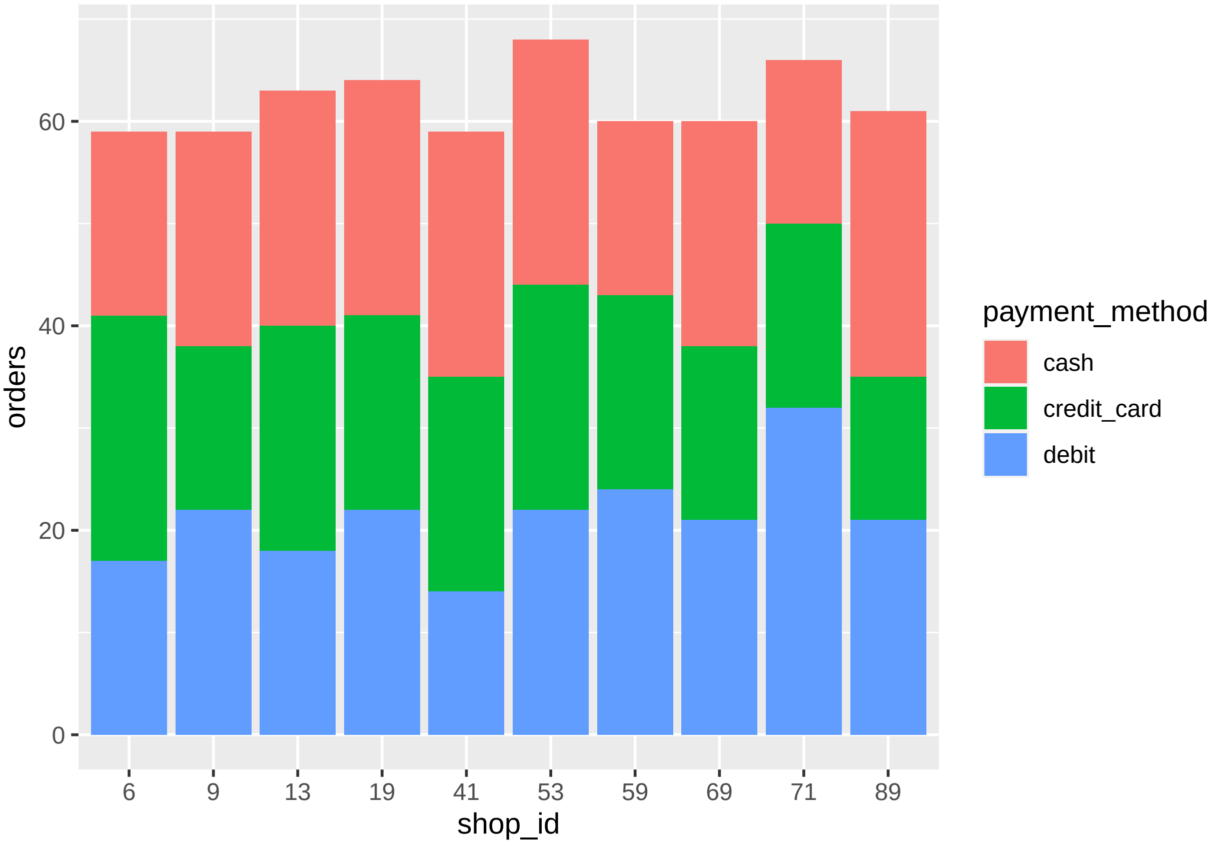
If we really wanted to rank the stores on their performance, we can simply look at the number of orders each store is getting and understand which stores are performing better than the others. We can do that simply by:

```
v2 = data %>%
  group_by(shop_id) %>%
  summarise_at(vars(order_id),
               list(orders = length)) %>%
  arrange(desc(orders))
head(v2)
```

```
## # A tibble: 6 x 2
##   shop_id orders
##     <int>  <int>
## 1      53     68
## 2      71     66
## 3      19     64
## 4      13     63
## 5      89     61
## 6      59     60
```

To compare payment methods and analyze which payment methods are being used more, we can even group by payment_method. We shall be displaying the performance of top 10 stores, visualizing using basic stacked chart.

```
v3 = data %>%
  group_by(shop_id, payment_method) %>%
  summarise_at(vars(order_id),
               list(orders = length))
v3 = v3 %>%
  group_by(shop_id) %>%
  mutate(tot_orders = sum(orders)) %>%
  arrange(desc(tot_orders))
v3$shop_id = factor(v3$shop_id, levels = c(seq(1,100)), ordered = F)
top10 = head(v3, 30)
ggplot(top10, aes(fill=payment_method, y=orders, x=shop_id)) + geom_bar(position="stack", stat="identit
```



2.

The metric that I would report would basically the number of orders for each store, we can then compare it to other stores and even use it to evaluate other metrics such as month-on-month growth etc. I thought this was the best thing to do as each store only sells one kind of shoe.

3. Through this metric we can compare stores, evaluate whcih store is doing better than the others. We can even see which store does better on weekdays vs weekends for added metrics. Moreover since each store only sells one kind of shoe, it's focus should not be on increasing aov, but to increase the number of gross orders as it will ensure more profits.