Sales data of vending machine

John Wu

2020.08.03

> This is a data set from XXXX company, This company currently has a vending machine business. This data is about the sales information of vending machines in different site in 2020 July. This document will explore some information among them and make some suggestions.

> 这是一个来自XXXX公司的数据集，这个公司目前有自动售货机业务，这个数据是关于不同点位在2020年7月的自动售货机的销售情况，本文档将要探索其中的信息并且作出一些建议。

Attention: This is a real business information. In order to keep it confidential, I have desensitized some of the information. This document is made for practiced and communication. Please do not use this data for other purposes.

注意事项：这是一个真实的商业信息，为了保密，我对一些信息进行了脱敏，本文档仅为了练习和交流，请不要用此数据做其他用途。

0.Prerequisite

In this lab,we use the following packages.

```
# load packages
library(dplyr)    # manipulate data
library(ggplot2)    # visualize
library(readxl)    # import xlsx
library(stringr)    # deal with string
library(forcats)    # deal with factor
library(statsr)    # useful package to statistics
```

and we import this xlsx file.

```
# import xlsx file
df <- read_excel("~/desktop/vending_machine.xlsx")
```

1.Data Cleaning

1.1 Size

This data is vast but contains many information we do not need,or some redundant columns. so first we clean it.it would be mach easier in the following step is we did this part well. let's see the size of this table.

```
dim(df)
```

```
## [1] 15496    34
```

commonly I will use `head()` function to show the glimpse of this data, but it may contain some confidential words, so I just use `dim()` function to show the size, and select columns directly.

1.2 Select Interested Columns

Now select columns we need.

```
# select columns
df1 <- df %>% select("交易日期", "售货机编号", "商品编号", "商品分类", "应结金额", "支付方式",
        "出货状态")
```

let's look into this new table df1.

```
head(df1, n = 10)
```

```
## # A tibble: 10 x 7
##    交易日期             售货机编号 商品编号 商品分类   应结金额 支付方式   出货状态
##    <dttm>               <chr>      <chr>    <chr>        <dbl> <chr>      <chr>
##  1 2020-07-01 00:00:00 `PP6019010 `        `矿泉水       1.19 `微信支付 `已出货
##  2 2020-07-01 00:00:00 `PP6019010 `        `矿泉水       1.19 `微信支付 `已出货
##  3 2020-07-01 00:00:00 `PP6019010 `        `茶饮料       2.98 `微信支付 `已出货
##  4 2020-07-01 00:00:00 `PP6019010 `        `茶饮料       2.98 `微信支付 `已出货
##  5 2020-07-01 00:00:00 `PP6019010 `        `功能饮料     4.97 `微信支付 `已出货
##  6 2020-07-01 00:00:00 `PP6019010 `        `功能饮料     4.97 `微信支付 `已出货
##  7 2020-07-01 00:00:00 `PP6019010 `Y01002  `功能饮料     4.97 `微信支付 `已出货
##  8 2020-07-01 00:00:00 `PP6019010 `Y05001  `矿泉水       1.19 `微信支付 `已出货
##  9 2020-07-01 00:00:00 `PP6019010 `        `矿泉水       1.19 `微信支付 `已出货
## 10 2020-07-01 00:00:00 `PP6019010 `        `矿泉水       1.19 `微信支付 `已出货
```

Form this table, we can get some observations,we would deal with them one by one in the following steps.

- It would be better if we translate the column name to English from Chinese.
- Some " ` " in our table need to be removed.
- “商品编号” column has many empty value, need to be convert to “NULL”.
- Values in columns like “商品分类”, “支付方式”, “出货状态” need to translate to English.

1.3 Rename Columns

Let’s translate column names to English.

```
# create a English column name
new_col_name <- c("date", "machine_id","prod_id", "prod_cat","price", "payment", "dea
        l_status")
# change column name by new_col_name
colnames(df1) <- new_col_name
```

column names explanation：

- date: the date deal completed
- machine_id: the id of vend machine
- prod_id: the id of product
- prod_cat: the category of product
- price: the price of pertaining product
- payment: the method of payment
- deal_status: shows whether the deal is completed,it may intrupted by the machine error

lets show the new dataframe.

```
head(df1,n = 10)
```

```
## # A tibble: 10 x 7
##    date                machine_id prod_id prod_cat  price payment   deal_status
##    <dttm>              <chr>      <chr>   <chr>     <dbl> <chr>     <chr>
##  1 2020-07-01 00:00:00 `PP6019010 `       `矿泉水    1.19 `微信支付 `已出货
##  2 2020-07-01 00:00:00 `PP6019010 `       `矿泉水    1.19 `微信支付 `已出货
##  3 2020-07-01 00:00:00 `PP6019010 `       `茶饮料    2.98 `微信支付 `已出货
##  4 2020-07-01 00:00:00 `PP6019010 `       `茶饮料    2.98 `微信支付 `已出货
##  5 2020-07-01 00:00:00 `PP6019010 `       `功能饮料  4.97 `微信支付 `已出货
##  6 2020-07-01 00:00:00 `PP6019010 `       `功能饮料  4.97 `微信支付 `已出货
##  7 2020-07-01 00:00:00 `PP6019010 `Y01002 `功能饮料  4.97 `微信支付 `已出货
##  8 2020-07-01 00:00:00 `PP6019010 `Y05001 `矿泉水    1.19 `微信支付 `已出货
##  9 2020-07-01 00:00:00 `PP6019010 `       `矿泉水    1.19 `微信支付 `已出货
## 10 2020-07-01 00:00:00 `PP6019010 `       `矿泉水    1.19 `微信支付 `已出货
```

1.4 Delete " `"

After rename the column,it is time to delete " ` " in some column(machine_id,prod_id,prod_cat,payment,deal_status), it is generated by the software we manage the vending machine.

```
# remove "`" in  front of value string
df2 <- df1 %>% mutate(
              machine_id = str_remove_all(machine_id,fixed("`")),
              prod_id = str_remove_all(prod_id,fixed("`")),
              prod_cat = str_remove_all(prod_cat,fixed("`")),
              payment = str_remove_all(payment,fixed("`")),
              deal_status = str_remove_all(deal_status,fixed("`"))
              )
```

show this new dataframe again.

```
head(df2,n = 10)
```

```
## # A tibble: 10 x 7
##    date                machine_id prod_id  prod_cat price payment  deal_status
##    <dttm>              <chr>      <chr>    <chr>    <dbl> <chr>    <chr>
##  1 2020-07-01 00:00:00 PP6019010  ""       矿泉水    1.19 微信支付 已出货
##  2 2020-07-01 00:00:00 PP6019010  ""       矿泉水    1.19 微信支付 已出货
##  3 2020-07-01 00:00:00 PP6019010  ""       茶饮料    2.98 微信支付 已出货
##  4 2020-07-01 00:00:00 PP6019010  ""       茶饮料    2.98 微信支付 已出货
##  5 2020-07-01 00:00:00 PP6019010  ""       功能饮料  4.97 微信支付 已出货
##  6 2020-07-01 00:00:00 PP6019010  ""       功能饮料  4.97 微信支付 已出货
##  7 2020-07-01 00:00:00 PP6019010  "Y01002" 功能饮料  4.97 微信支付 已出货
##  8 2020-07-01 00:00:00 PP6019010  "Y05001" 矿泉水    1.19 微信支付 已出货
##  9 2020-07-01 00:00:00 PP6019010  ""       矿泉水    1.19 微信支付 已出货
## 10 2020-07-01 00:00:00 PP6019010  ""       矿泉水    1.19 微信支付 已出货
```

1.5 Deal With Missing Values

The ***prod_id*** column contains many empty values,it may be a bug of the software we use to manage our machine,we replace it with “NULL”.

```
# turn empty value to "NULL"
df3 <- df2 %>% mutate(prod_id = if_else(prod_id == "", "NULL",prod_id))
```

1.6 Rename Values

There are 3 columns need to translate value.

prod_cat

```
df3 %>% count(prod_cat)
```

```
## # A tibble: 9 x 2
##   prod_cat       n
##   <chr>      <int>
## 1 茶饮料      2510
## 2 功能饮料    4101
## 3 果蔬汁饮料  3079
## 4 咖啡饮料     142
## 5 矿泉水      4705
## 6 膨化          29
## 7 乳饮料        36
## 8 食品          32
## 9 碳酸饮料     862
```

As we can see, there are 9 different kind of values need to be translated.

- 茶饮料: tea beverage (tea)
- 功能饮料: energy drinks like Red Bull (energy_drink)
- 果蔬汁饮料: fruit and vegetable juice drinks (juice)
- 咖啡饮料: coffee drink (coffee)
- 矿泉水: mineral water (mineral_water)
- 膨化: puffed food (puffed_food)
- 乳饮料: milk drink (milk)
- 食品: foods like bread (bread)
- 碳酸饮料: carbonated beverage like Pepsi (carbon_drink)

Now let's translate values.

```
# translate values
df3 <- df3 %>% mutate(
  prod_cat = if_else(prod_cat == "茶饮料", "tea",prod_cat),
  prod_cat = if_else(prod_cat == "功能饮料", "energy_drink",prod_cat),
  prod_cat = if_else(prod_cat == "果蔬汁饮料", "juice",prod_cat),
  prod_cat = if_else(prod_cat == "咖啡饮料", "coffee",prod_cat),
  prod_cat = if_else(prod_cat == "矿泉水", "mineral_water",prod_cat),
  prod_cat = if_else(prod_cat == "膨化", "puffed_food",prod_cat),
  prod_cat = if_else(prod_cat == "乳饮料", "milk",prod_cat),
  prod_cat = if_else(prod_cat == "食品", "bread",prod_cat),
  prod_cat = if_else(prod_cat == "碳酸饮料", "carbon_drink",prod_cat),
  )
```

payment

Our vending machine has only two payment methods,one is 支付宝(Alipay), another is 微信支付(WeChat Pay),paper money and coin are not supported.those two method use some technology like QR code,which are widely used in China now.

- 支付宝:Alipay (alipay)
- 微信支付:WeChat Pay (wechat)

```
# translate values
df3 <- df3 %>% mutate(
  payment = if_else(payment == "支付宝", "alipay",payment),
  payment = if_else(payment == "微信支付", "wechat",payment)
  )
```

deal_status

deal_status has two status, one is success,another is failure,the failure is usually caused by machine error.

- 出货失败: Failure (failure)
- 已出货: Success (success)

```
# translate values
df3 <- df3 %>% mutate(
  deal_status = if_else(deal_status == "出货失败", "failure",deal_status),
  deal_status = if_else(deal_status == "已出货", "success",deal_status)
  )
```

1.7 Hidden Wrong In Price

Up to now, we had cleaned this data a lot, it is time to show the summary,it may help us find something.

```
summary(df3)
```

```
##       date                        machine_id          prod_id
##  Min.   :2020-07-01 00:00:00   Length:15496       Length:15496
##  1st Qu.:2020-07-09 00:00:00   Class :character   Class :character
##  Median :2020-07-17 00:00:00   Mode  :character   Mode  :character
##  Mean   :2020-07-16 08:38:32
##  3rd Qu.:2020-07-24 00:00:00
##  Max.   :2020-07-30 00:00:00
##    prod_cat             price          payment           deal_status
##  Length:15496       Min.   :-6.460   Length:15496       Length:15496
##  Class :character   1st Qu.: 1.990   Class :character   Class :character
##  Mode  :character   Median : 3.480   Mode  :character   Mode  :character
##                     Mean   : 3.339
##                     3rd Qu.: 4.970
##                     Max.   : 6.460
```

From this summary, we find the minimum price is *-6.460*, this is obviously wrong， I consulted the relevent staff, find it is because the deal is not success, so this negative value indicates a refund. So let's check the failure **deal_status** and non-positive **price**.

```
# show total number of failure order
df3 %>% filter(deal_status == "failure") %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   396
```

```
# find the order with price 0 and failure status
df3 %>% filter(deal_status == "failure" & price == 0) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     8
```

```
# show composition of order with negative price
df3 %>% filter(price < 0) %>% count(deal_status)
```

```
## # A tibble: 2 x 2
##   deal_status     n
##   <chr>       <int>
## 1 failure       194
## 2 success         9
```

It seems that we get so many information.

Concerning about deal_status,we find a total failure of 396 orders, in those 396 failure orders, it contains 8 orders with price equals 0,(I consulted relevant staff,the price equals 0 means this is a promotion, means **buy one get one free**), so the negative price failure orders are 388(396 - 8).

Now take price into consideration,when the price is negative(which means it would be a failure order),the total number is 203,with failure 194,and success 9.it is obvious that the observations with negative price but success deal_status are error.so the true failure orders are 194.this is half of 388, because in the table, we set the failure in a pair,the one is the deal we demand,another is also this deal but with a refund.

In general, we need to filter failure orders,and non-positive price is not welcome.

```
# filter observations we need to new dataframe df4
df4 <- df3 %>% filter(deal_status == "success" & price > 0)

# store failure orders in df_failure
df_failure <- df3 %>% filter(deal_status == "failure" & price != 0)
```

the failure cases may be useful in the following part,so I save them to df_failure.

So much for this cleaning, after several steps, we cleaned the data, so it is time to start our journey with new dataframe df4, and don't forget we have df_failure for failure orders.

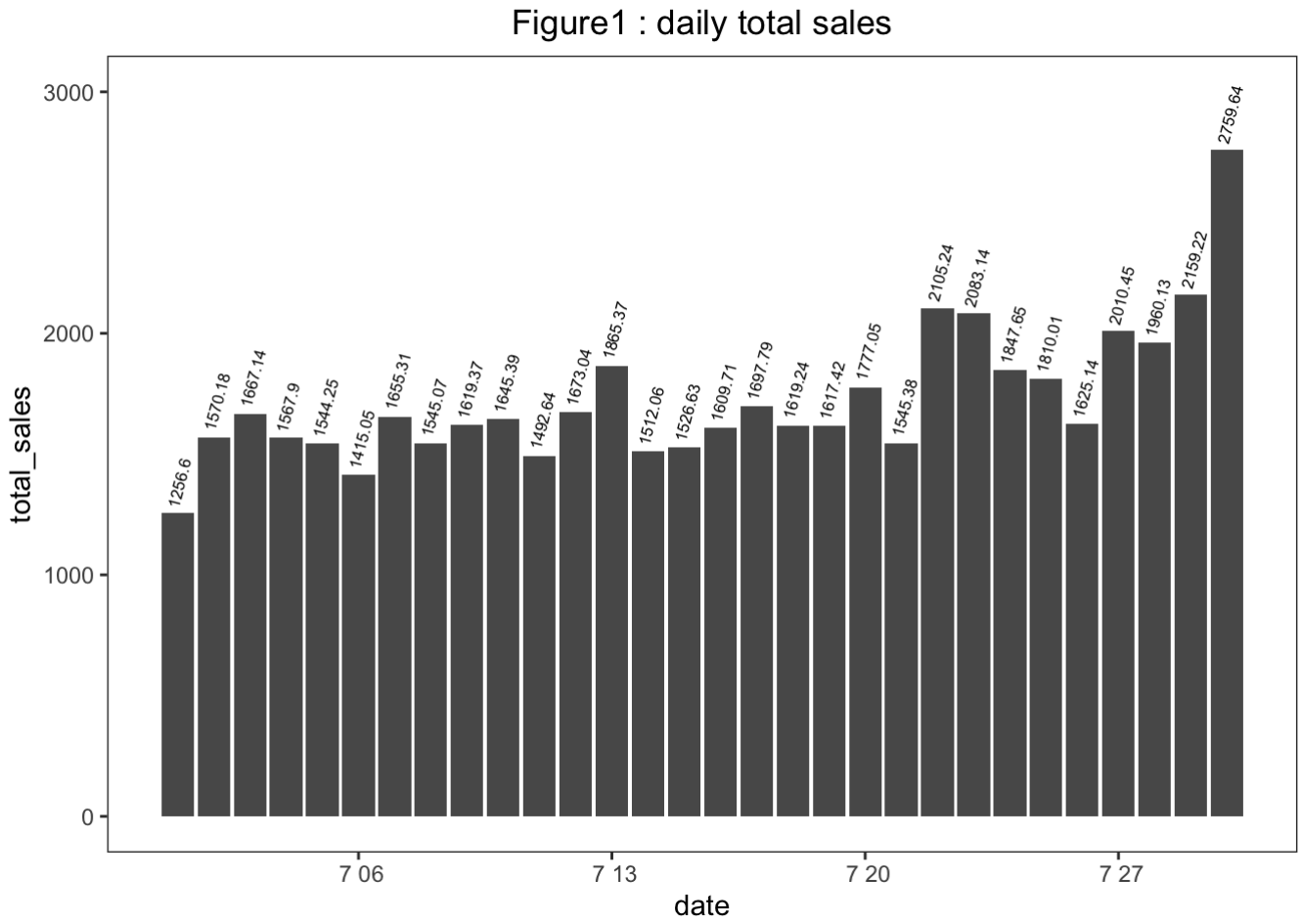
2.Exploratory Data Analysis

In this part, We could freely explore data depend on our interest.

2.1 About *date*

date is important,Let's look into daily total sales trend.

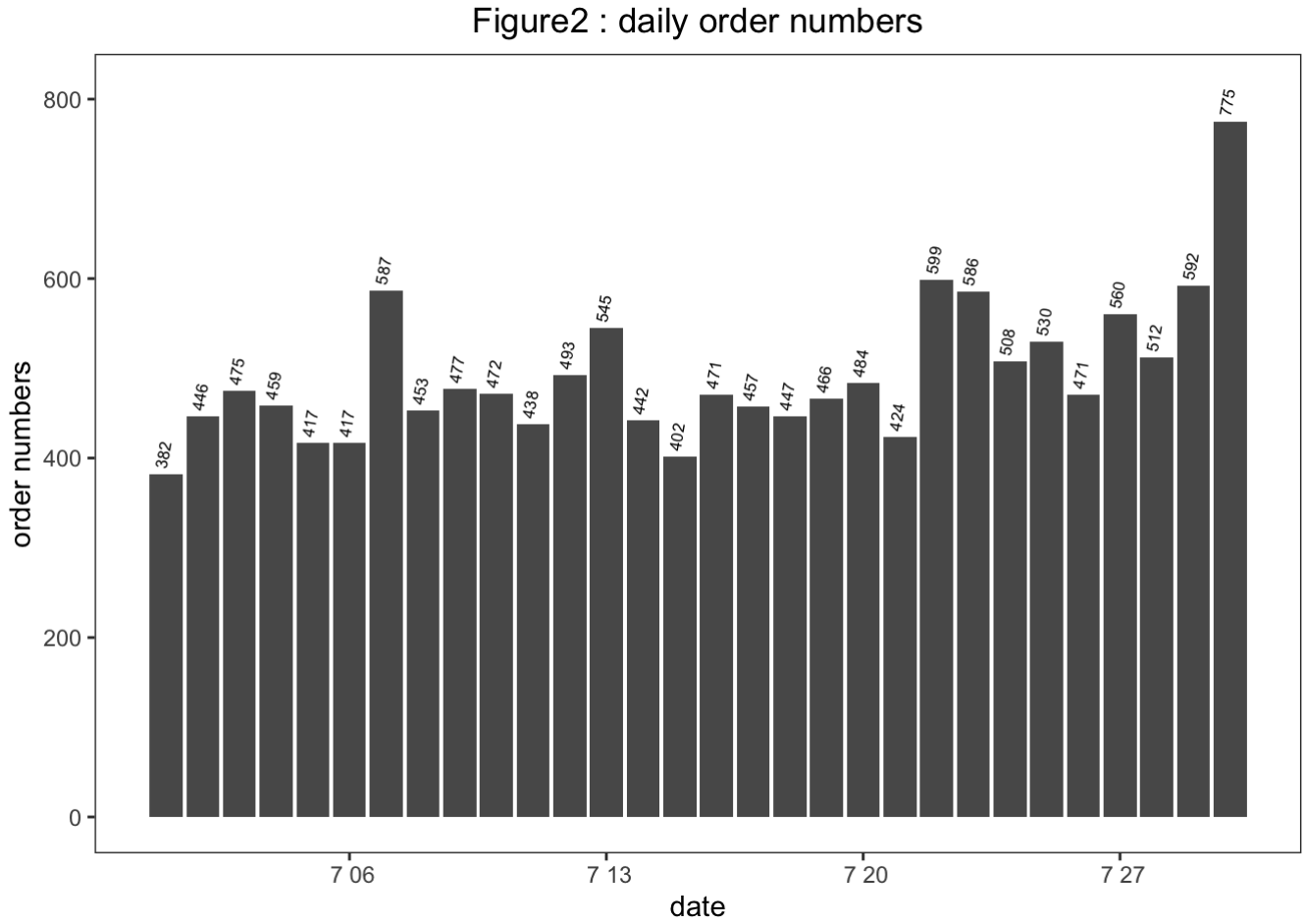
```
# compute total sales group by day
df4 %>% group_by(date) %>%
  summarize(total_sales = sum(price)) %>%
  ggplot(aes(x = date, y = total_sales)) + geom_col() +
  ggtitle("Figure1 : daily total sales") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid = element_blank()
        ) +
  geom_text(aes(label = total_sales), hjust = - 0.1,vjust = 0, size = 2.2,angle = 75)
        +
  ylim(0,3000)
```



observation1: total sales raise up day by day,it seems our promotional events is successful.(Attention:in July, the team added several vending machines, which may be a factor in the increase in total sales.)

Now, let's check daily orders.

```
# show daily orders trend
df4 %>% group_by(date) %>% count() %>%
  ggplot(aes(x = date, y = n)) + geom_col() +
  ggtitle("Figure2 : daily order numbers") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid = element_blank()
        ) +
  geom_text(aes(label = n), hjust = - 0.2,vjust = 0, size = 2.2,angle = 80) +
  ylab("order numbers") +
  ylim(0,810)
```

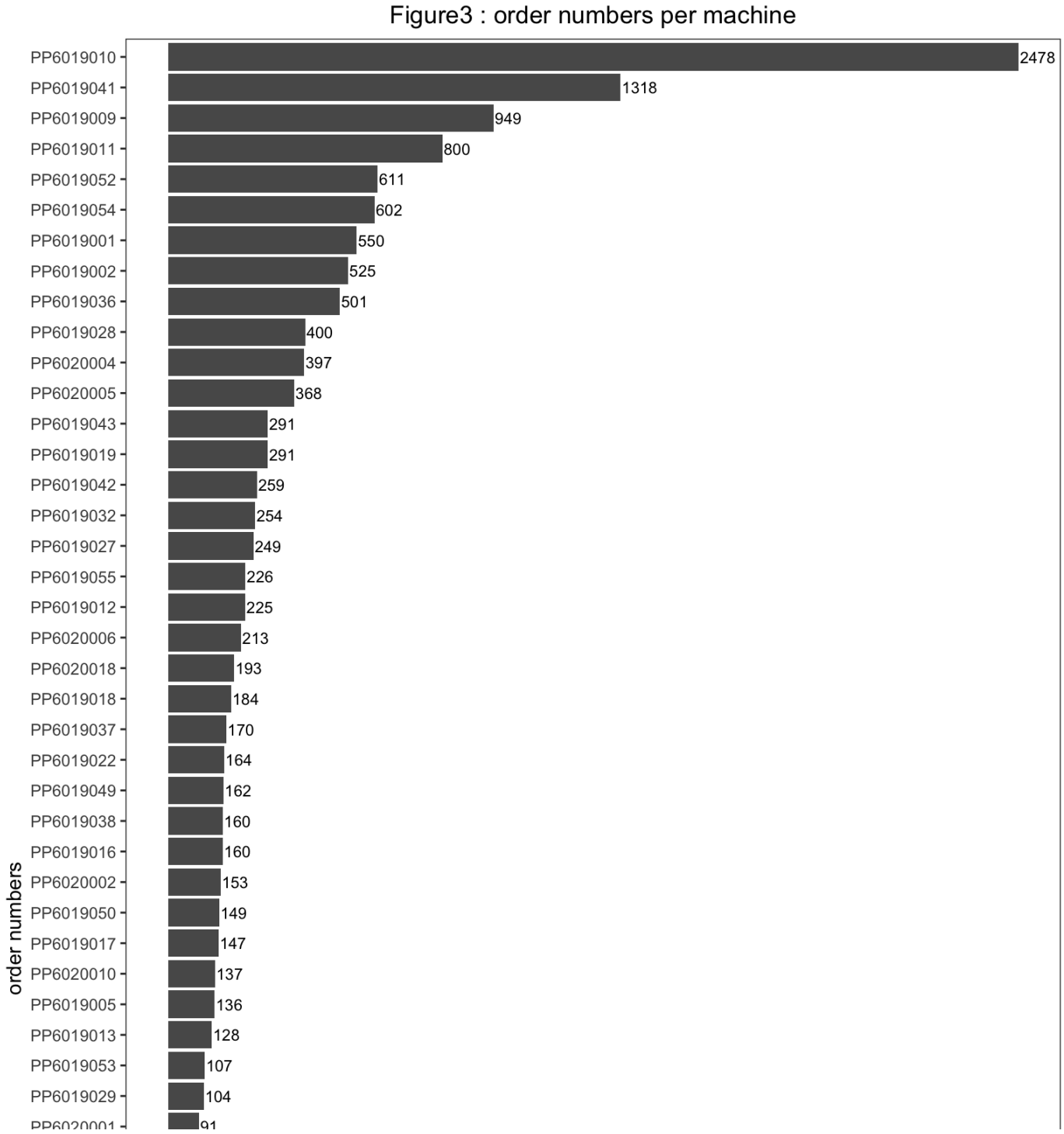


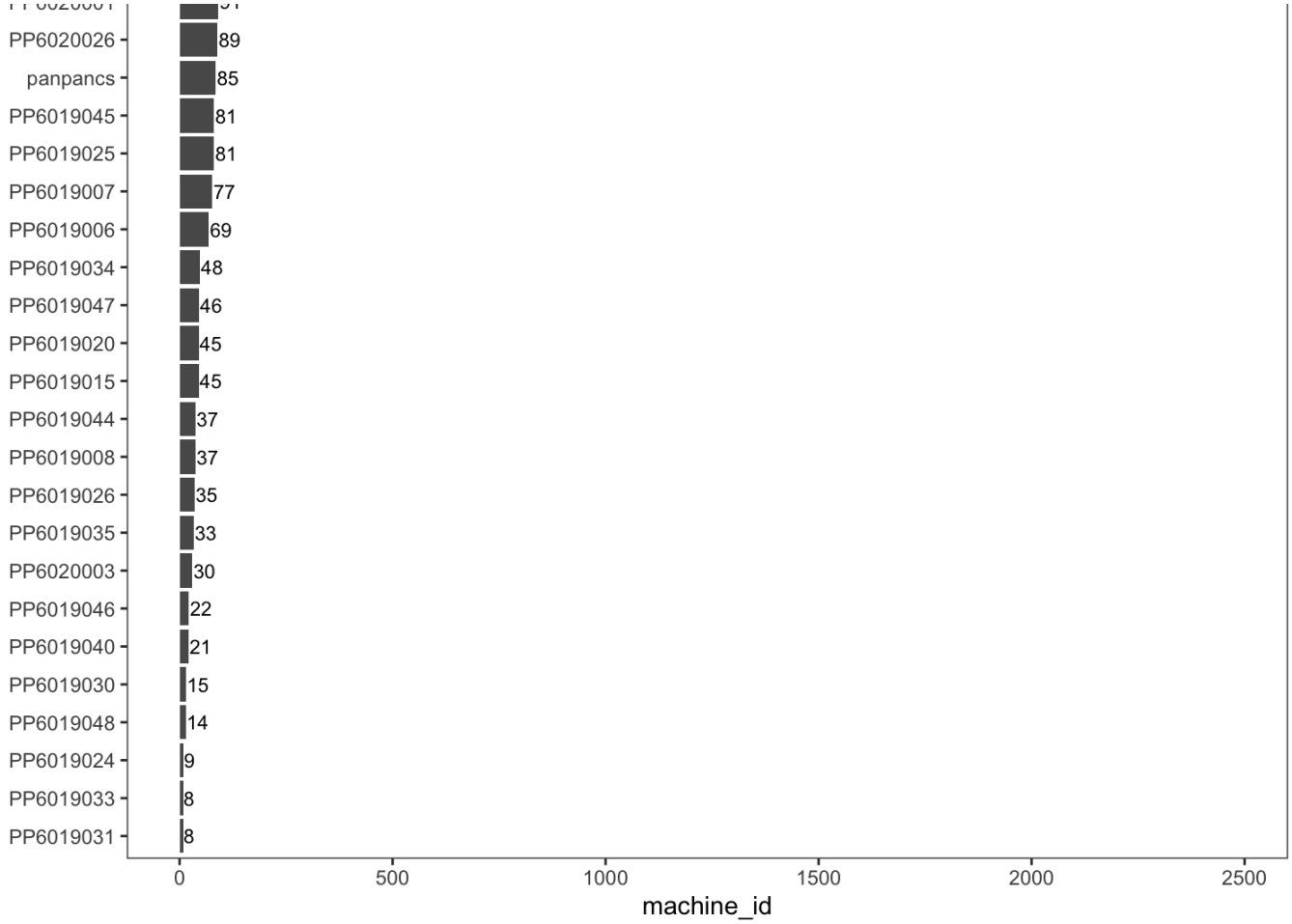
observation2: plot of orders numbers per day looks seem like figure1,it means it is rational.

2.2 About *machine_id*

Now it is time to discuss ***machine_id*** .

```
# turn machine_id from character to factor,and use forcats package to reorder machine
        _id by order number
df4 %>% mutate(machine_id = as.factor(machine_id)) %>%
  group_by(machine_id) %>%
  count() %>%
  ggplot(aes(x = fct_reorder(machine_id, n),y = n)) + geom_col() +
  ggtitle("Figure3 : order numbers per machine") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid = element_blank()
    ) +
  coord_flip() +
  xlab("order numbers") +
  ylab("machine_id") +
  geom_text(aes(label = n),hjust = -0.05,size = 3)
```





observation3: From figure 3, we can find the top 10 machine_id with highest order numbers are:

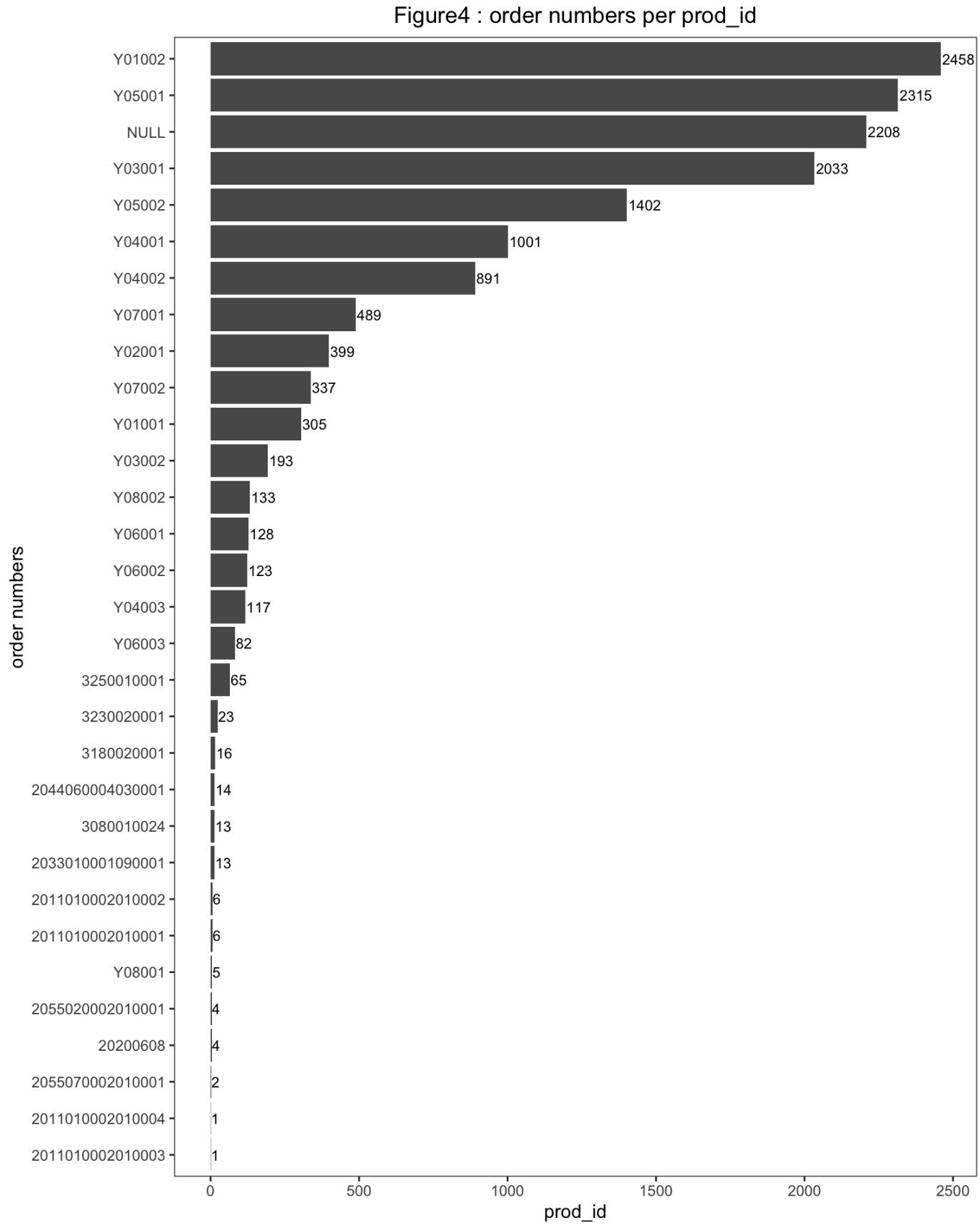
- PP6019010 : This machine is awesome,with 2478 orders, and nearly 2 times of NO2(PP6019041),It seem we need very carefully study the location of this vending machine and find out if there are any factors that could promote sales.
- PP6019041 : This machine is very good,with 1318 orders,We need to carefully study this machine too.
- PP6019009 : pretty good machine with 949 orders,need study.
- PP6019011 : pretty good machine with 800 orders,need study.
- PP6019052 : pretty good machine with 611 orders,need study.
- PP6019054 : pretty good machine with 602 orders,need study.
- PP6019001 : good machine with 550 orders,need study.
- PP6019002 : good machine with 525 orders,need study.
- PP6019036 : good machine with 501 orders,need study.
- PP6019028 : good machine with 400 orders,need study. The top 10 worst machine are as follows:
- PP6019031 : barely 8 orders, need study.
- PP6019033 : barely 8 orders, need study.
- PP6019024 : barely 9 orders, need study.
- PP6019048 : 14 orders, need study.
- PP6019030 : 15 orders, need study.
- PP6019040 : 21 orders, need study.

- PP6019046 : 22 orders, need study.
- PP6019003 : 30 orders, need study.
- PP6019035 : 33 orders, need study.
- PP6019026 : 35 orders, need study. In general,those machine with high orders and low orders should be study.

2.3 About *prod_id*

Now let's turn to ***prod_id***.

```
# turn prod_id to factor and reorder
df4 %>% mutate(prod_id = as.factor(prod_id)) %>%
  group_by(prod_id) %>%
  count() %>%
  ggplot(aes(x = fct_reorder(prod_id, n),y = n)) + geom_col() +
  ggtitle("Figure4 : order numbers per prod_id") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    panel.grid = element_blank()
    ) +
  coord_flip() +
  xlab("order numbers") +
  ylab("prod_id") +
  geom_text(aes(label = n),hjust = -0.05,size = 3)
```



observation4:The difference between different products is significant, product like “Y01002”, “Y05001”, “Y03001” have order numbers over 2000, are most popular among our products,and product like “Y05002”, “Y04001” and “Y04002” have order numbers between 500 and 2000,are also our popular products,however, there are some products have orders less than 10,that is awful.(the prod_id equals “NULL” are caused by machine error, we will omit this one)

Let’s check the popular and unpopular products category.

```
# define popular product with orders higher than 500,and delete "NULL" prod_id
popular_product <- df4 %>% group_by(prod_id) %>% count() %>% filter(n > 500 & prod_id
        != "NULL")
# popular products list
popular_product$prod_id
```

```
## [1] "Y01002" "Y03001" "Y04001" "Y04002" "Y05001" "Y05002"
```

```
# find category of our popular product
df4 %>% filter(prod_id %in% popular_product$prod_id) %>% count(prod_cat)
```

```
## # A tibble: 4 x 2
##   prod_cat          n
##   <chr>         <int>
## 1 energy_drink   2458
## 2 juice          2033
## 3 mineral_water  3717
## 4 tea            1892
```

observation5: Our popular product are composed of “mineral_water”,“energy_drink”, “juice” and “tea”.

Let’s see un-popular product.

```
# define unpopular product with orders less than 10
unpopular_product <- df4 %>% group_by(prod_id) %>% count() %>% filter(n < 10)
# unpopular products list
unpopular_product$prod_id
```

```
## [1] "2011010002010001" "2011010002010002" "2011010002010003" "2011010002010004"
## [5] "20200608"         "2055020002010001" "2055070002010001" "Y08001"
```

```
# find category of our unpopular product
df4 %>% filter(prod_id %in% unpopular_product$prod_id) %>% count(prod_cat)
```

```
## # A tibble: 4 x 2
##   prod_cat         n
##   <chr>        <int>
## 1 bread           12
## 2 coffee           5
## 3 energy_drink     4
## 4 puffed_food      8
```

observation6: It seems that our “coffee”, “puffed_food” and “bread” are unpopular, the “energy_drink” is special, it is so welcomed in our popular group,so we need to find this energy drink.

```
# find the unpopular energy drink product
df4 %>% filter(prod_id %in% unpopular_product$prod_id & prod_cat == "energy_drink")
```

```
## # A tibble: 4 x 7
##   date                machine_id prod_id  prod_cat     price payment deal_status
##   <dttm>              <chr>      <chr>    <chr>        <dbl> <chr>   <chr>
## 1 2020-07-15 00:00:00 PP6019022  20200608 energy_drink  5.96 alipay  success
## 2 2020-07-15 00:00:00 PP6019022  20200608 energy_drink  5.96 alipay  success
## 3 2020-07-19 00:00:00 PP6019026  20200608 energy_drink  5.96 alipay  success
## 4 2020-07-22 00:00:00 PP6019022  20200608 energy_drink  5.96 alipay  success
```

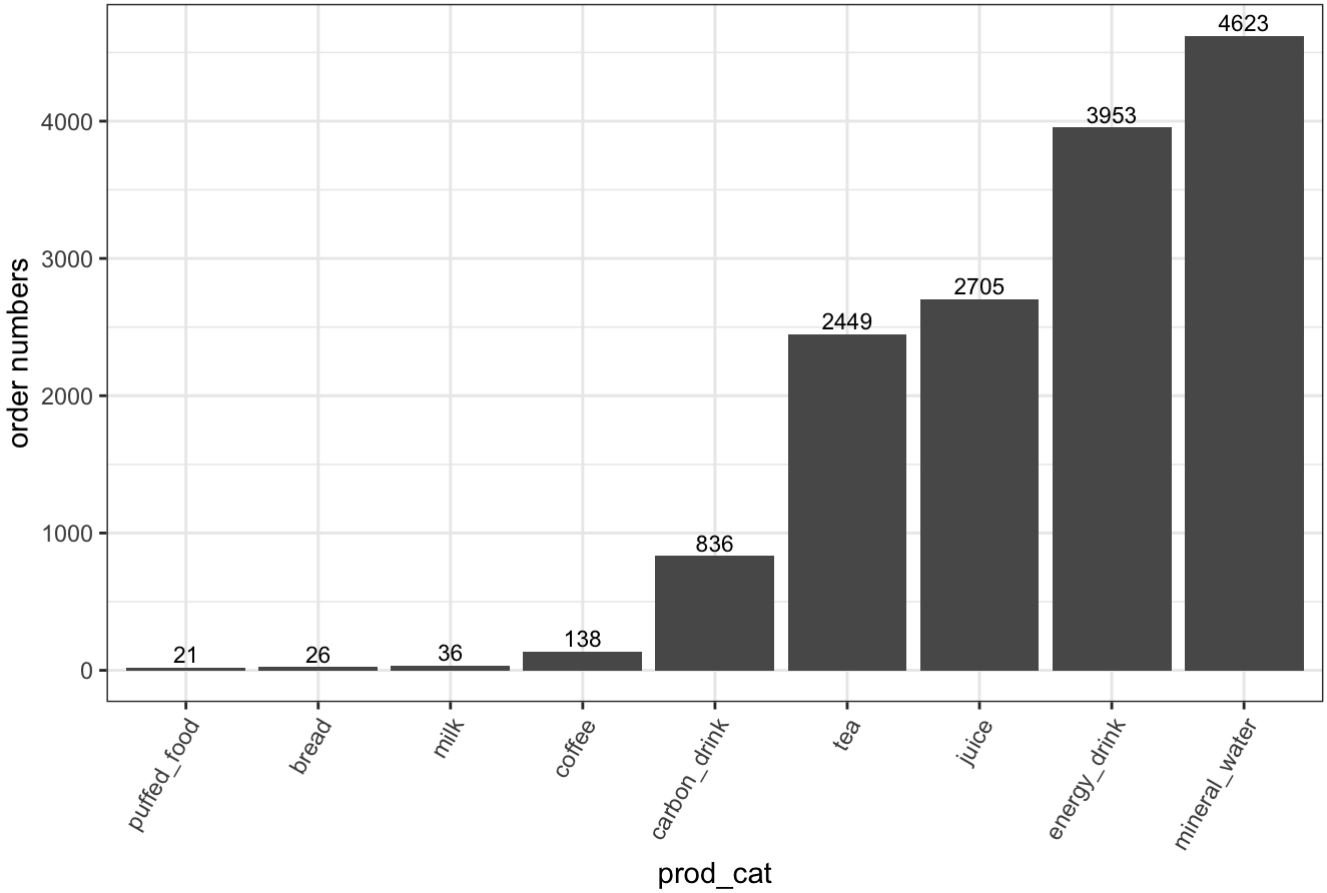
observation7: The product “20200608” should be study,why other energy_drink sales good,however this one so bad.

2.4 About *prod_cat*

We have study the category before, but now it is time to explore it more thoroughly.

```
# orders per catgory
df4 %>% mutate(prod_cat = as.factor(prod_cat)) %>%
  group_by(prod_cat) %>%
  count() %>%
  ggplot(aes(x = fct_reorder(prod_cat, n),y = n)) + geom_col() +
  ggtitle("Figure5 : order numbers per prod_cat") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.text.x=element_text(angle=60,hjust = 1)
    ) +
  ylab("order numbers") +
  xlab("prod_cat") +
  geom_text(aes(label = n),size = 3,vjust = - 0.3)
```

Figure5 : order numbers per prod_cat



observation7: Obviously “puffed_food”, “bread” and “milk” are unpopular, while “mineral_water”, “energy_drink” are very popular, “juice” and “tea” are popular too.

2.5 About *price*

Let’s talk about ***price***.

```
df4 %>% summarize(mean_price = mean(price),
                  median_price = median(price),
                  min_price = min(price),
                  max_price = max(price),
                  price_q1 = quantile(price, 0.25),
                  price_q3 = quantile(price, 0.75)
                  )
```

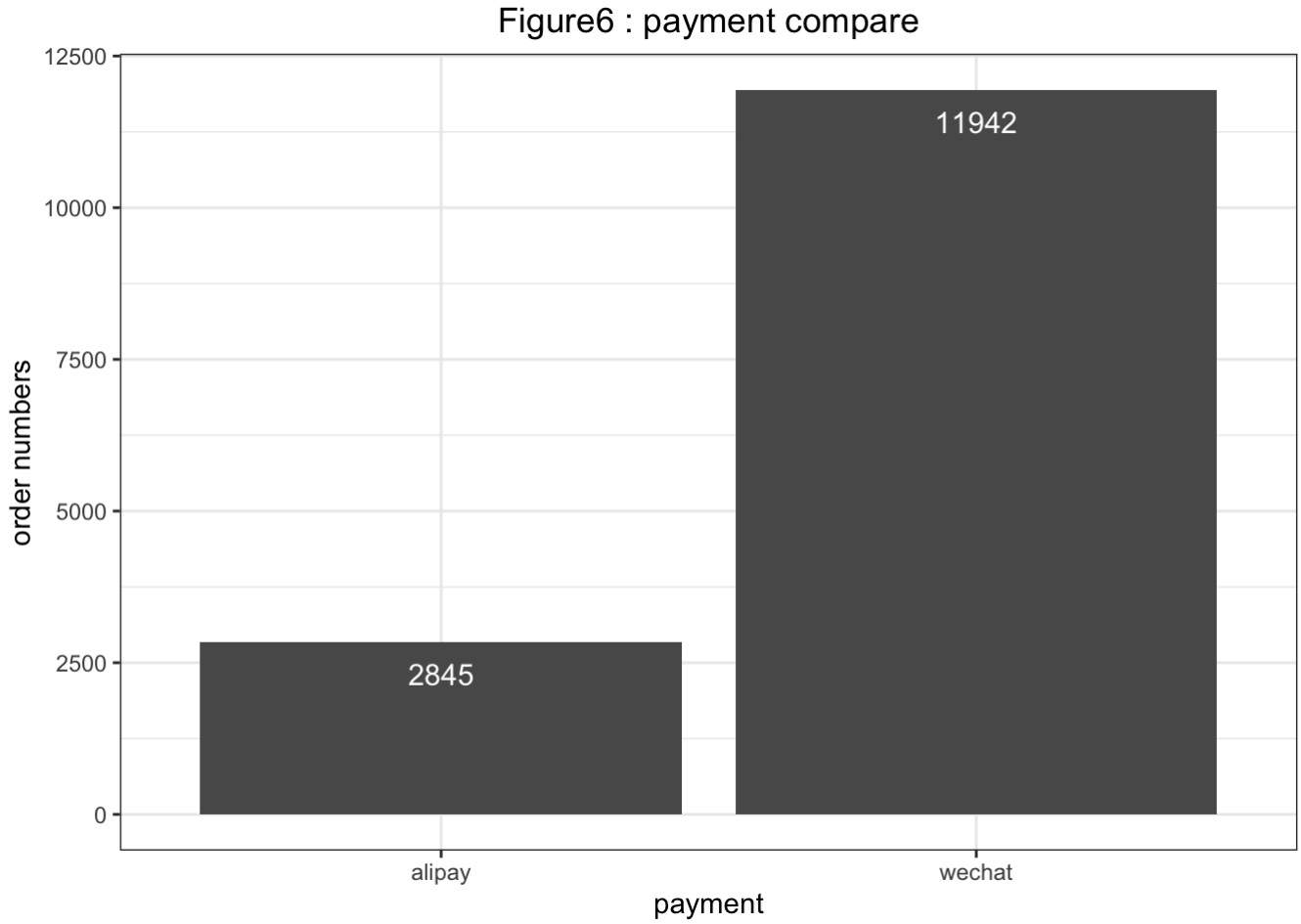
```
## # A tibble: 1 x 6
##   mean_price median_price min_price max_price price_q1 price_q3
##        <dbl>        <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
## 1       3.50         3.48       0.5      6.46     1.99     4.97
```

observation8: the product price range from 0.5 to 6.46, with mean price 3.48, in general, the price is cheap.

2.6 About *payment*

Let’s check the payment method.

```
# df4 %>% ggplot(aes(x = payment)) + geom_bar()
df4 %>% group_by(payment) %>% count() %>%
  ggplot(aes(x = payment, y = n)) + geom_col() +
  ggtitle("Figure6 : payment compare") +
  theme_bw() +
  theme(
    plot.title = element_text(hjust = 0.5)
    ) +
  xlab("payment") +
  ylab("order numbers") +
  geom_text(aes(label = n),size = 4,vjust = 2, color = "white")
```



observation9: The result shocked me,I thought they would be half and half before, but it seems WeChat Pay is more popular among our customers.(WeChat is also the most popular communication app, so people may let this app active in there smart phone background, so it may be more convenient to pay with WeChat Pay)

2.7 About Failure

Let's talk about failure orders,the reason are various,software error,network error may cause failure.I want to see if some payment method have higher failure rate.

```
df_failure %>% group_by(payment) %>% count()
```

```
## # A tibble: 2 x 2
## # Groups:   payment [2]
##   payment     n
##   <chr>   <int>
## 1 alipay     98
## 2 wechat    290
```

observation10: Note that the number should be divided by 2 because an order has double observation.In failure data,Alipay has proportion of 0.25(49/194), while in success data the proportion is 0.19(2845/14787). it is hard to say whether payment method proportion are the same in different deal status.we need an inference.

3 Inference

In the last but one section，I want to dive into the relationship of payment method and deal status.in other words,Regarding the two payment methods, is the order success rate the same?

3.1 Inference Preparation

Remember our cleaned data frame df3? we will use this data frame to build up a new one,by filter orders with positive price,then change **payment** and **deal_status** to factor.

```
# filter positive price orders
df_inference <- df3 %>% filter(price > 0)
# convert **payment** and **deal_status** to factor
df_inference <- df_inference %>% mutate(payment = as.factor(payment), deal_status = a
        s.factor(deal_status))
```

3.2 Hypothesis Test

In this part,Let's do a hypothesis test,the question is in different payment method(Alipay and WeChat Pay), the success order rate are the same or not.And we set $\alpha = 0.05$.

Set Hypothesis:

Let's set our hypothesis.

$H_0 : P_{alipay} - P_{wechat} = 0;$

$H_A : P_{alipay} - P_{wechat} \neq 0$

Conditions Confirmation:

Before we perform our test,we need to confirm whether the conditions are met.

before that we need compute $\widehat{p}_{pool}$, $\widehat{p}_{pool} = \frac{2845+11942}{2894+12087} = 0.987.$

- Independence:

1. Within groups: each orders are independent for both payment, and customers less than 10% of population for both payment too.
2. Between groups: different orders are independent of each other.

- Sample size:

1. Alipay success : $n_{alipay} \times \widehat{p}_{pool} = 2894 \times 0.987 \approx 2856 > 10$
2. Alipay failure : $n_{alipay} \times (1 - \widehat{p}_{pool}) = 2894 \times (1 - 0.987) \approx 37 > 10$
3. WeChat Pay success : $n_{wechat} \times \widehat{p}_{pool} = 12807 \times 0.987 \approx 12640 > 10$

4. WeChat Pay failure : $n_{wechat} \times (1 - \hat{p}_{pool}) = 12807 \times (1 - 0.987) \approx 166 > 10$

- Skew:

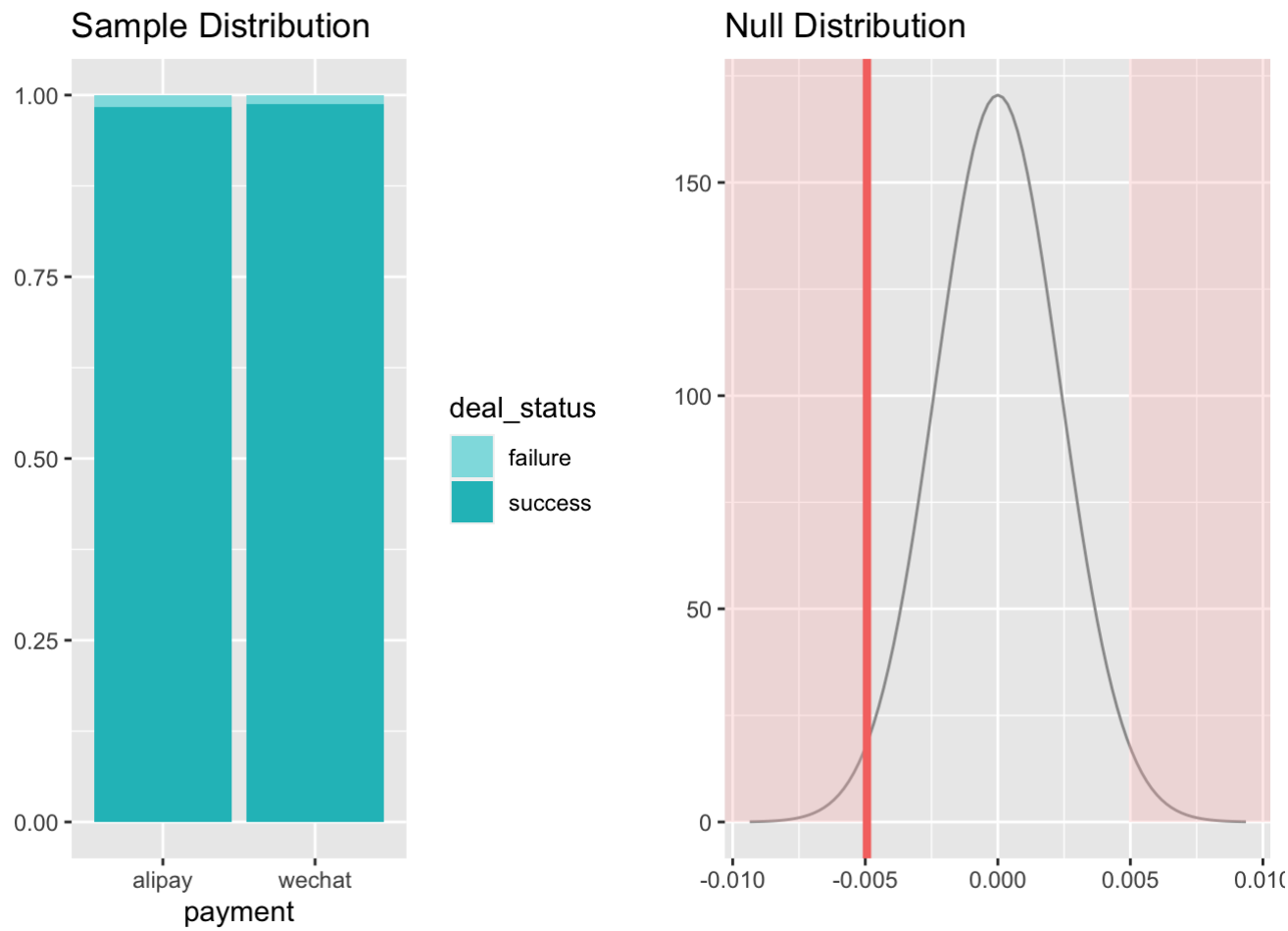
  We can assume that the sampling distribution of the difference between proportions is nearly normal.

As all requirements met well, we can proceed.

Hypothesis Test

```
# inference is a function in *statsr* package
inference(y = deal_status, x = payment, data = df_inference, statistic = "proportion"
        , type = "ht", null = 0, alternative = "twosided", method = "theoretical", su
        ccess = "success")
```

```
## Response variable: categorical (2 levels, success: success)
## Explanatory variable: categorical (2 levels)
## n_alipay = 2894, p_hat_alipay = 0.9831
## n_wechat = 12087, p_hat_wechat = 0.988
## H0: p_alipay =  p_wechat
## HA: p_alipay != p_wechat
## z = -2.1093
## p_value = 0.0349
```



In the result, We can see the p_value is 0.0349, it is small than our α, so we reject the H_0, that is means the success rate of different payment method is different.

3.3 Confidence Interval

Also we examine the confidence interval of success rate of different payment method in 95% confidence level.

Also we need comfirm conditions.

Conditions Confirmation:

- Independence:

  Within groups: each orders are independent for both payment, and customers less than 10% of population for both payment too.

- Sample size:

1. Alipay success : 2845 > 10
2. Alipay failure : 49 > 10
3. WeChat Pay success : 11942 > 10
4. WeChat Pay failure : 142 > 10

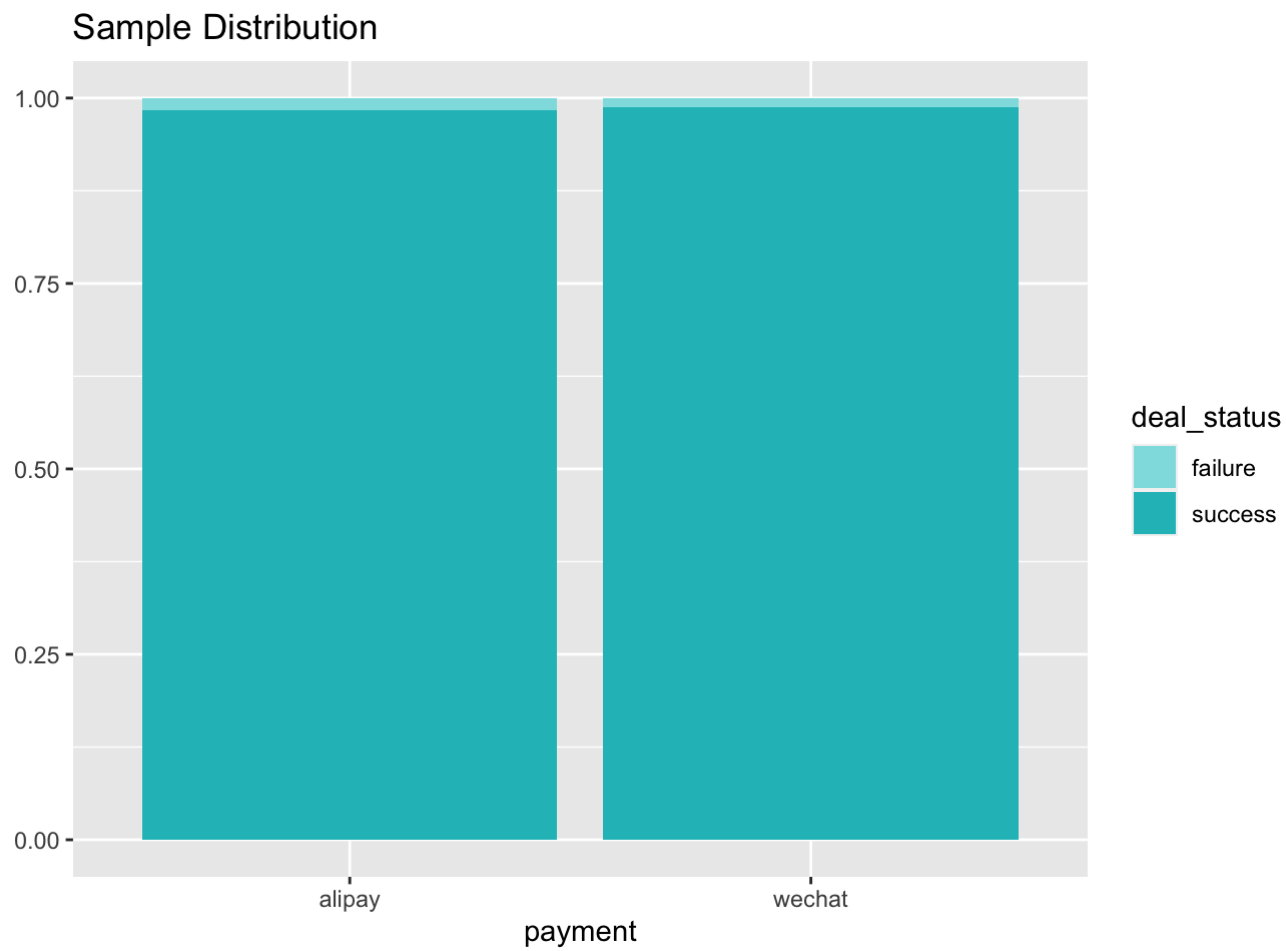
- Skew:

  We can assume that the sampling distribution of the difference between proportions is nearly normal.

Confidence Interval

```
inference(y = deal_status, x = payment, data = df_inference, statistic = "proportion"
        , type = "ci", method = "theoretical", success = "success")
```

```
## Response variable: categorical (2 levels, success: success)
## Explanatory variable: categorical (2 levels)
## n_alipay = 2894, p_hat_alipay = 0.9831
## n_wechat = 12087, p_hat_wechat = 0.988
## 95% CI (alipay - wechat): (-0.01 , 2e-04)
```



In the result, we can find the confidence interval of P_{alipay} - P_{wechat} are (-0.01, 0.0002), it is hard to say two different payment method has no difference in success rate based on this interval(because 0 is in it),There are many reasons why this result is inconsistent with the hypothesis test conclusion，

- The result of the confidence interval and the conclusion of the hypothesis test are not absolutely consistent.
- The condition of hypothesis test and confidence interval may not meet completely， and the success rate is too high in both method.
- the upper limit of confidence interval is 0.0002, it is so close to 0.

Based on the above facts,and my personal judgment,I think the success rate of 2 different payment method are different, and the WeChat Pay has a higher success rate.

4. Conclusion

Our research on this dataset is over, it is time to draw a conclusion.

1. Overall, sales in July have steadily increased, indicating that we are generally in the right direction, but we need to make some adjustments based on the following results.
2. Vending machine “PP6019010”, “PP6019041” are our heroes,We need to carefully study the location of these vending machines and the surrounding environment and other factors to find out why the sales at these sites are so good. And put it to other vending machines, the specific machine list can refer to figure 3
3. The sales of vending machine “PP6019031”, “PP6019033” and “PP6019024” are bad,We also need to carefully study the location of these vending machines and the surrounding environment and other factors to find out why these sites are not selling well. And to prevent it from appearing on other vending machines, please refer to figure 3 for the specific machine list
4. Some products sell well, such as “Y01002”,“Y05001” and “Y03001”. These competitive products can be further promoted, and some products are not sold well, such as “2011010002010003”, “2011010002010003”, “2011010002010001”, etc., These products can be considered for special promotion activities, price reduction or withdrawal from the counter, etc. , See figure 4 for specific product sales
5. Regarding the categories of products, we found that mineral_water, energy_drink and juice sell well, while puffed_food, bread and milk are not selling well. We need to make other plans for these popular and unpopular categories.
6. In the payment method, WeChat Pay greatly beats Alipay, which is about 4:1. This result, which is different from general perception, may require further research
7. The overall order success rate is 0.987($\widehat{p}_{pool}$), which is pretty high, but when our sales increase, the number of lost orders is also considerable, and the failure of the order may make customers distrust our vending machine. In addition, the data exported by the machine management system has some logical errors. And null value, related departments need to optimize the software of our vending machine
8. In the last part, we did a hypothesis test on whether the success rates of different payment methods are different. The result is that within the criterion of $\alpha = 0.05$, we believe that the success rates of the two payment methods are significantly different. Then we calculated the confidence interval of the success rate of Alipay and WeChat Pay payment methods is (-0.01, 0.0002) under the 95% confidence level.We are 95% confident that the success rate difference between Alipay compare to WeChat Pay is (-0.01, 0.0002) every order in our vending machine.

5. Postscripts

In this report, I found some areas for improvement.

- Because this report is not for commercial use, it is just for communication and practice, so the words and sentences I choose is not that commercial.
- And there are still some unresolved questions, but I don’t have the corresponding resources to solve them, just raise my questions.like WeChat Pay is much more popular than Alipay in our orders,is there some error in our system? like QR code of ALipay shows very slow or even can not shown,because Alipay is a more professional payment method,it owns more than half of the market share.
- In fact, the original data set also has a user identification column. I am not sure whether this column is accurate and how it is implemented. If this represents the unique identification of the user, then we can check whether someone will proceed a new order after the order fails.if he continues to purchase, whether he will change the payment method.