User Behavior

Mingwei(Show) Wu

1/23/2021

Background

Data randomly extract 8477 customers and 1048575 values from 2014/11/18 to 2014/12/18 in taobao shopping, including customer_id,items_id,behavior_type,location,iten

Goal

Generate the model by analysing User-behavior, and reporting the outcome.

data interpretation

user_id: randomly int of user

item_id: item id

behavior: pv(page view), fav(collection), cart(cart), buy

item_category: type of category of items

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```r
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
ub<-read.csv("UserBehavior.csv",skip = 1,header = TRUE,nrow=200000)
```

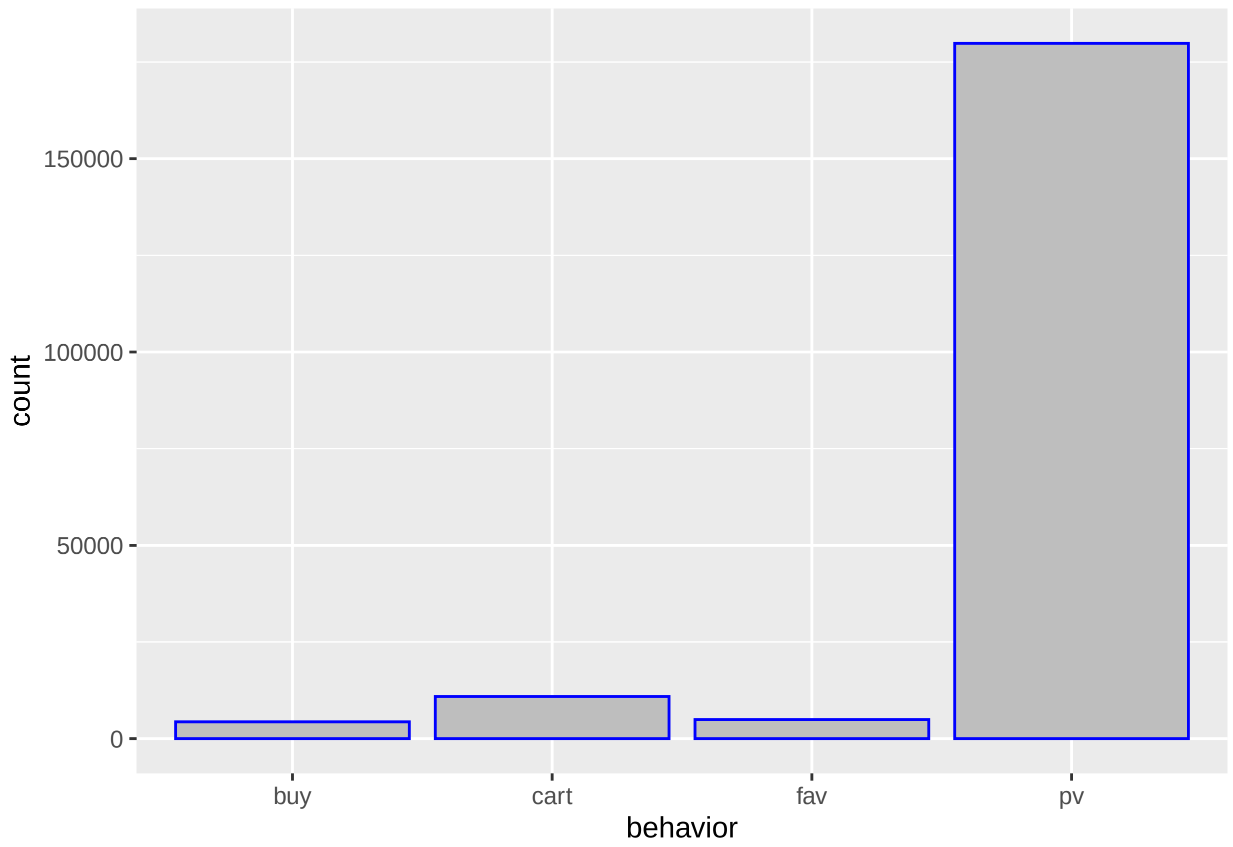
```r
ub<-ub%>%
  rename(id=X1,
         user_id=X2333346,
         item_id=X2520771,
         behavior=pv,
         item_category=X1511561733)
```

```r
head(ub)
```

```
##   id user_id item_id behavior item_category
## 1  1 2576651  149192       pv    1511572885
## 2  1 3830808 4181361       pv    1511593493
## 3  1 4365585 2520377       pv    1511596146
## 4  1 4606018 2735466       pv    1511616481
## 5  1  230380  411153       pv    1511644942
## 6  1 3827899 2920476       pv    1511713473
```

```r
ub%>%
  ggplot(aes(behavior))+geom_histogram(stat = 'count', binwidth = 50, color='blue',fill='gray') #histog
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



## conducted data with SQL to check duplicated observations.

```
sqldf("select *, count(*)
      from ub
      group by id,user_id,item_id,behavior,item_category
      having count(user_id) > 1") # none of ob is duplicated
```

```
## [1] id            user_id       item_id       behavior      item_category
## [6] count(*)
## <0 rows> (or 0-length row.names)
```

```
ub%>%
  group_by(user_id)%>%
  count()%>%
  arrange(desc(n)) # calculates 117081 unique customers buy on the website in 200 thousand observations
```

```
## # A tibble: 117,081 x 2
## # Groups:   user_id [117,081]
##    user_id     n
##      <int> <int>
##  1  812879    66
##  2 2331370    64
##  3  138964    54
##  4 3131062    53
##  5 1223110    46
##  6 2818406    40
##  7 3027414    40
##  8 3845720    39
##  9 4657130    39
## 10 2338453    38
## # ... with 117,071 more rows
```

```
ub1<-ub%>%
  group_by(behavior)%>%
  count() #counting the amount of behavior type
ub1
```

```
## # A tibble: 4 x 2
## # Groups:   behavior [4]
##   behavior      n
##   <chr>     <int>
## 1 buy        4329
## 2 cart      10906
## 3 fav        4934
## 4 pv       179831
```

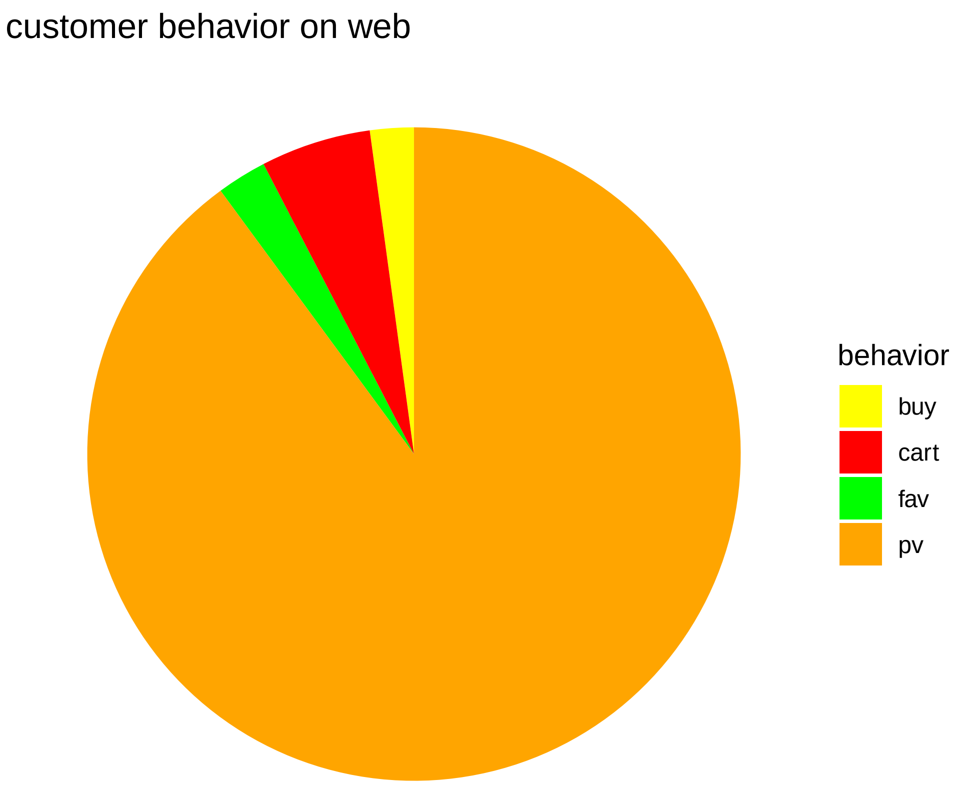
```
ub2<-ub1%>%
  ungroup(behavior)%>%
  arrange(n)
ub2
```

```
## # A tibble: 4 x 2
##   behavior      n
##   <chr>     <int>
## 1 buy        4329
## 2 fav        4934
## 3 cart      10906
## 4 pv       179831
```

```
ub2<-ub2 %>%
  mutate(behavior = factor(behavior,
                levels = c("buy","cart","fav","pv")),
              cumulative = cumsum(n),
                midpoint = cumulative - n/2,
                labels = paste0(round((n/ sum(n)) * 100, 1), "%"))
ub2
```

```
## # A tibble: 4 x 5
##   behavior      n cumulative midpoint labels
##   <fct>     <int>      <int>    <dbl> <chr>
## 1 buy        4329       4329    2164. 2.2%
## 2 fav        4934       9263    6796  2.5%
## 3 cart      10906      20169   14716  5.5%
## 4 pv       179831     200000  110084. 89.9%
```

```
ub2%>%
  ggplot(aes(x="",y=n,fill=behavior))+geom_bar(width=1,stat = "identity")+
  coord_polar(theta = "y",start=0)+labs(x="",y="", title="customer behavior on web",fill="behavior")+sca
```



```
ub1 #page_view is 179831 in 200k observations
```

```
## # A tibble: 4 x 2
## # Groups:   behavior [4]
##   behavior      n
##   <chr>     <int>
## 1 buy        4329
## 2 cart      10906
## 3 fav        4934
## 4 pv       179831
```

```
unique(ub["user_id"])%>%
  count()   # unique user amount click on website as 117081 in 200k observations
```

```
##        n
## 1 117081
```

```
PV<-c(179831)
UV<-c(117081)
rate_clicked_person<-PV/UV
page_view<-data.frame(PV,UV,rate_clicked_person)
page_view #page_view is 179831, unique customer amount is 117081, mean of clicked is 1.53% as rate
```

```
##       PV     UV rate_clicked_person
## 1 179831 117081            1.535954
```

```
cart<-c(10906); fav<-c(4934);buy<-4329
behavior<-c("PV","fav+cart","buy")
quantity<-c(PV,fav+cart,buy)
rate<-c(PV/PV,(fav+cart)/PV,buy/PV)
```

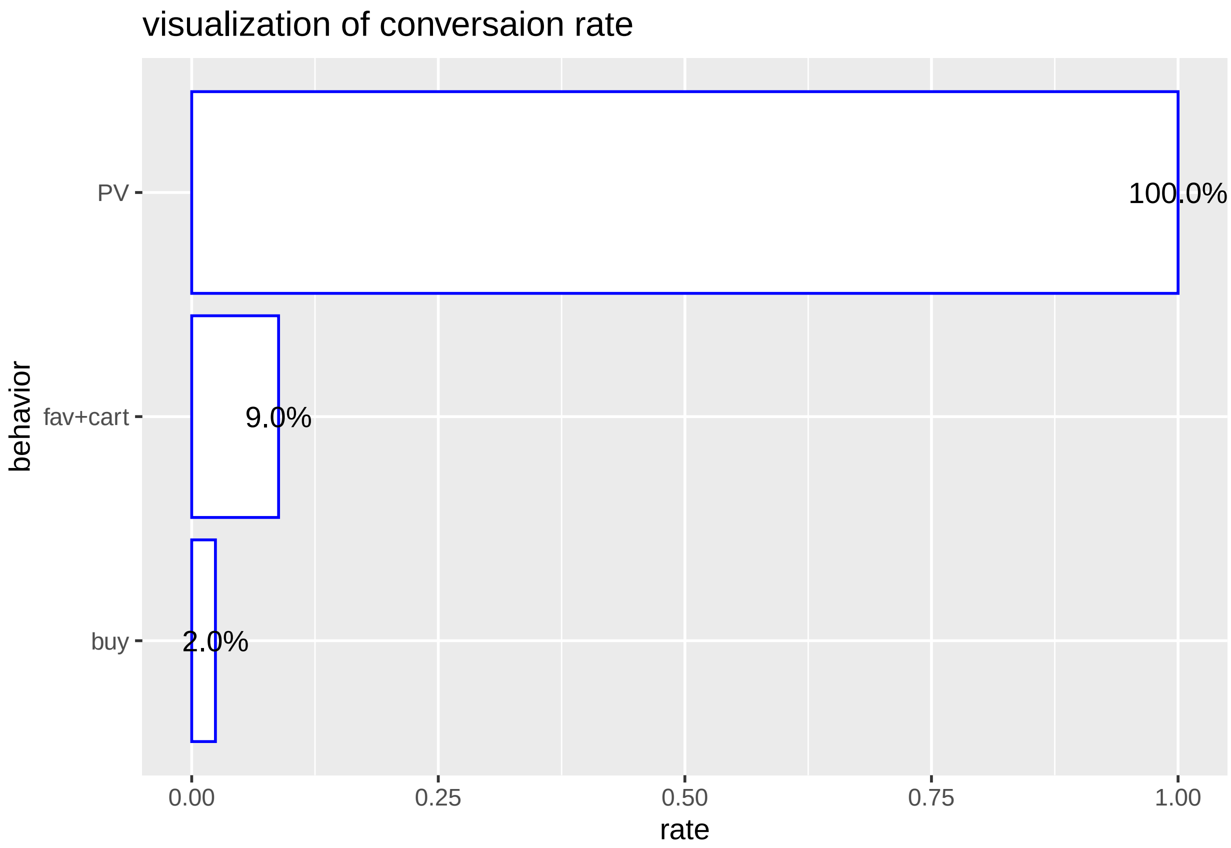
```
rate_shopping<-data.frame(behavior,quantity,rate)
rate_shopping  # only 2.4% of the customer who fished processing of shopping, and 9.0% of page_view is
```

```
##   behavior quantity      rate
## 1       PV   179831 1.0000000
## 2 fav+cart    15840 0.0880827
## 3      buy     4329 0.0240726
```

```
cr<-buy/(fav+cart);percent(cr) #27% as conversation rate between fav+cart and buy
```

```
## [1] "27%"
```

```
rate_shopping%>%
  ggplot(aes(behavior,y=rate))+geom_bar(stat="identity",color="blue",fill="white")+coord_flip()+geom_te
```



```
vnb<-(PV+cart+fav)-buy;vnb #191342 customer only view the page,including add good to chart, but not buy
```

```
## [1] 191342
```

```
sqldf("select count(behavior) as total_buy
      from ub
      where behavior =’buy’") #only 4329 as buy amount in 300k observations
```

```
##   total_buy
## 1      4329
```

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(user_id)%>%
  count()%>%
  ungroup()%>%
  summarise(rate_person_buy=mean(n)) #1.08% as the rate of people in buy behavior
```

```
## # A tibble: 1 x 1
##   rate_person_buy
##             <dbl>
## 1            1.08
```

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(user_id)%>%
  count()%>%
  ungroup()%>%
  summarise(total_person_buy=n())
```

```
## # A tibble: 1 x 1
##   total_person_buy
##              <int>
## 1             4007
```

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(user_id)%>%
  count()%>%
  filter(n > 1)%>%
  ungroup()%>%
  summarise(rate_people_morethan_once=n()) #269 people is buy on the website more than once
```

```
## # A tibble: 1 x 1
##   rate_people_morethan_once
##                       <int>
## 1                       269
```

```
269/4007 # only 6.7% people who buy more than once in the time period.
```

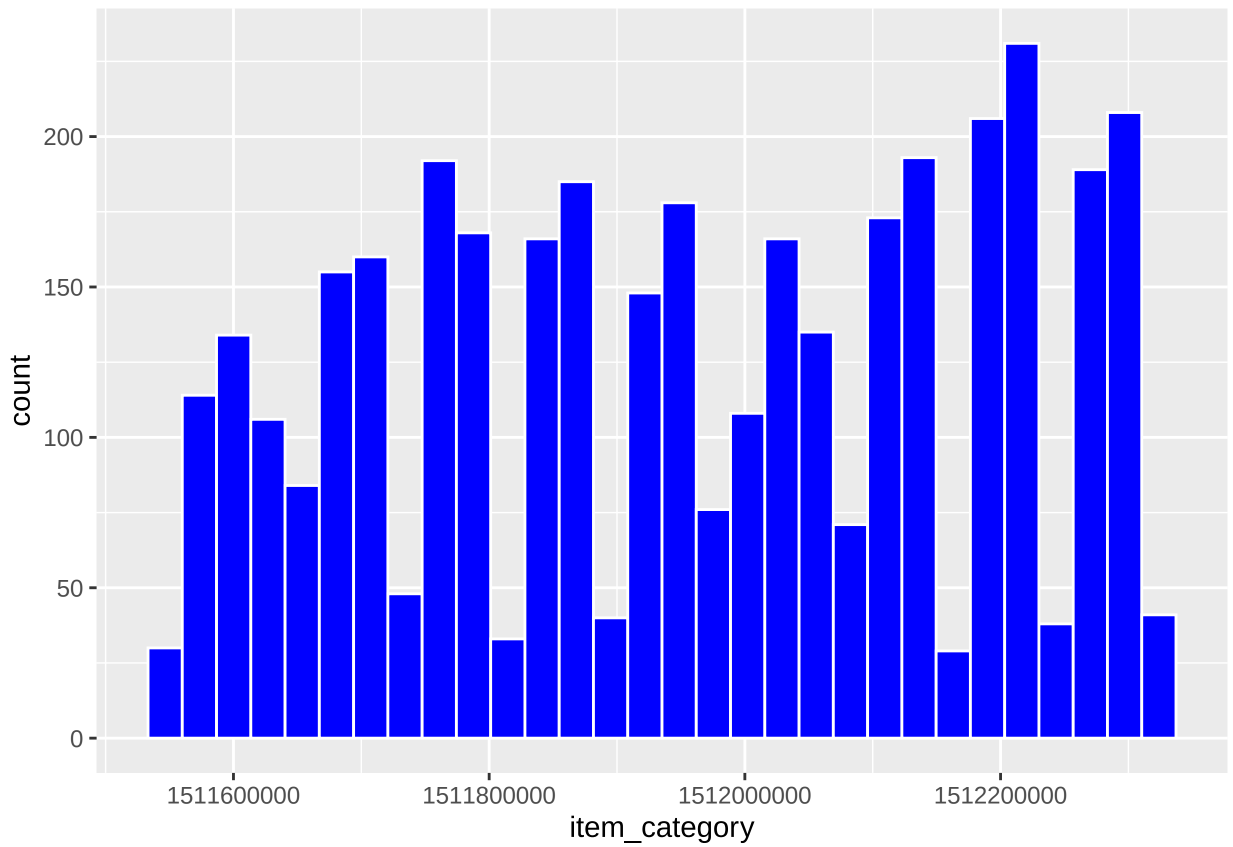
```
## [1] 0.06713252
```

```
sqldf("select item_category, count(behavior) as amout_buy
      from ub
      where behavior == 'buy'
      group by item_category
      order by count(behavior) desc limit 10") # top 10 sales
```

```
##    item_category amout_buy
## 1     1512206464        26
## 2     1511859171        20
## 3     1512041875        14
## 4     1511859722        12
## 5     1511933797        11
## 6     1511886641        11
## 7     1511814264        11
## 8     1511885508        10
## 9     1511710300        10
## 10    1511998190         9
```

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(item_category)%>%
  count()%>%
  arrange(desc(n))%>%
  ggplot(aes(item_category))+geom_histogram(color="white",fill="blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ub%>%
  filter(behavior=="pv")%>%
  group_by(item_id)%>%
  count()%>%
  arrange(desc(n))%>%
  head() #top 6 view of 6 items
```

```
## # A tibble: 6 x 2
## # Groups:   item_id [6]
##   item_id     n
##     <int> <int>
## 1 4756105  8238
## 2 3607361  6955
## 3 4145813  5823
## 4  982926  5685
## 5 2355072  5614
## 6 2520377  3891
```

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(item_id)%>%
  count()%>%
  arrange(desc(n))%>%
  head() # top 6 sales of items
```

```
## # A tibble: 6 x 2
## # Groups:   item_id [6]
##   item_id     n
##     <int> <int>
## 1 1464116    71
## 2 2735466    64
## 3 2885642    63
## 4 4145813    62
## 5  901282    47
## 6 4801426    47
```

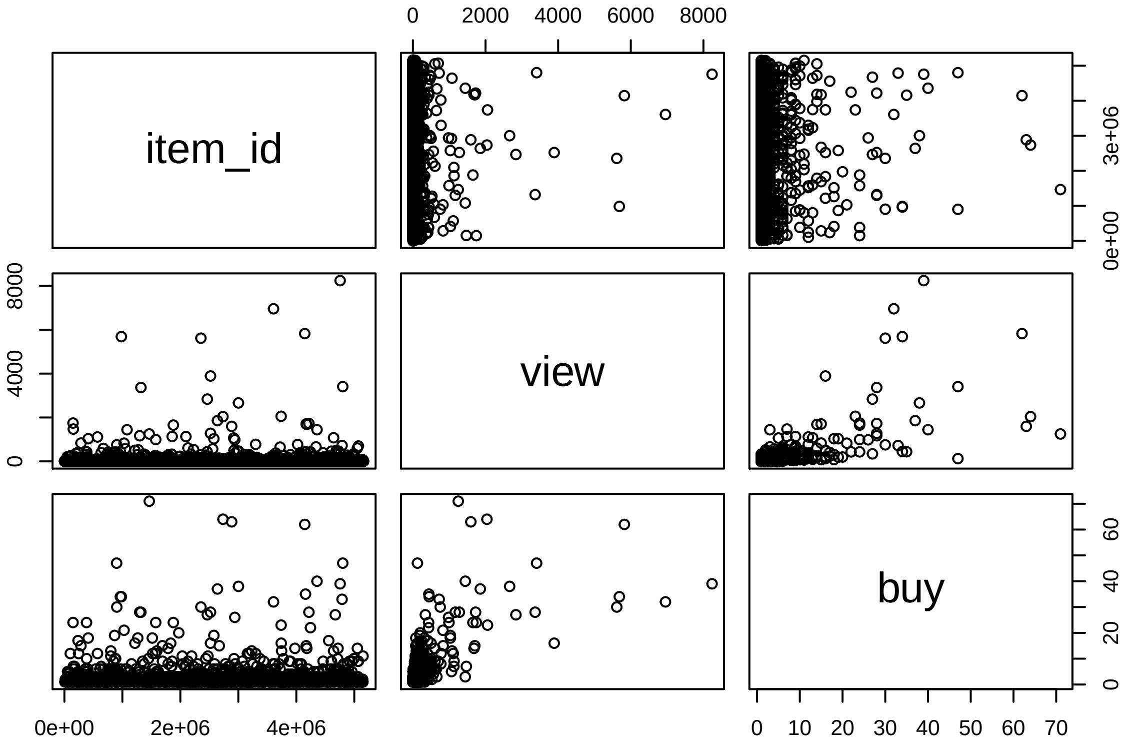
```
sqldf(" select a.item_id, a.view, b.buy
      from (
      select item_id, count(*) as view
      from ub
      where behavior =’pv’
      group by item_id
      order by count(*) desc
      limit 6
      ) as a
      join (
      select item_id, count(*) as buy
      from ub
      where behavior =’buy’
      group by item_id
      order by count(*) desc
      limit 6
      ) as b
      on a.item_id=b.item_id") # using SQL to query the relationship between the view and buy. only 1 i
```

```
##   item_id view buy
## 1 4145813 5823  62
```

```
view_buy<-sqldf(" select a.item_id, a.view, b.buy
      from (
      select item_id, count(*) as view
      from ub
      where behavior =’pv’
      group by item_id
      order by count(*) desc
      ) as a
      left join (
      select item_id, count(*) as buy
      from ub
      where behavior =’buy’
      group by item_id
      order by count(*) desc
      ) as b
      on a.item_id=b.item_id") #create a data frame between view and buy
head(view_buy)
```

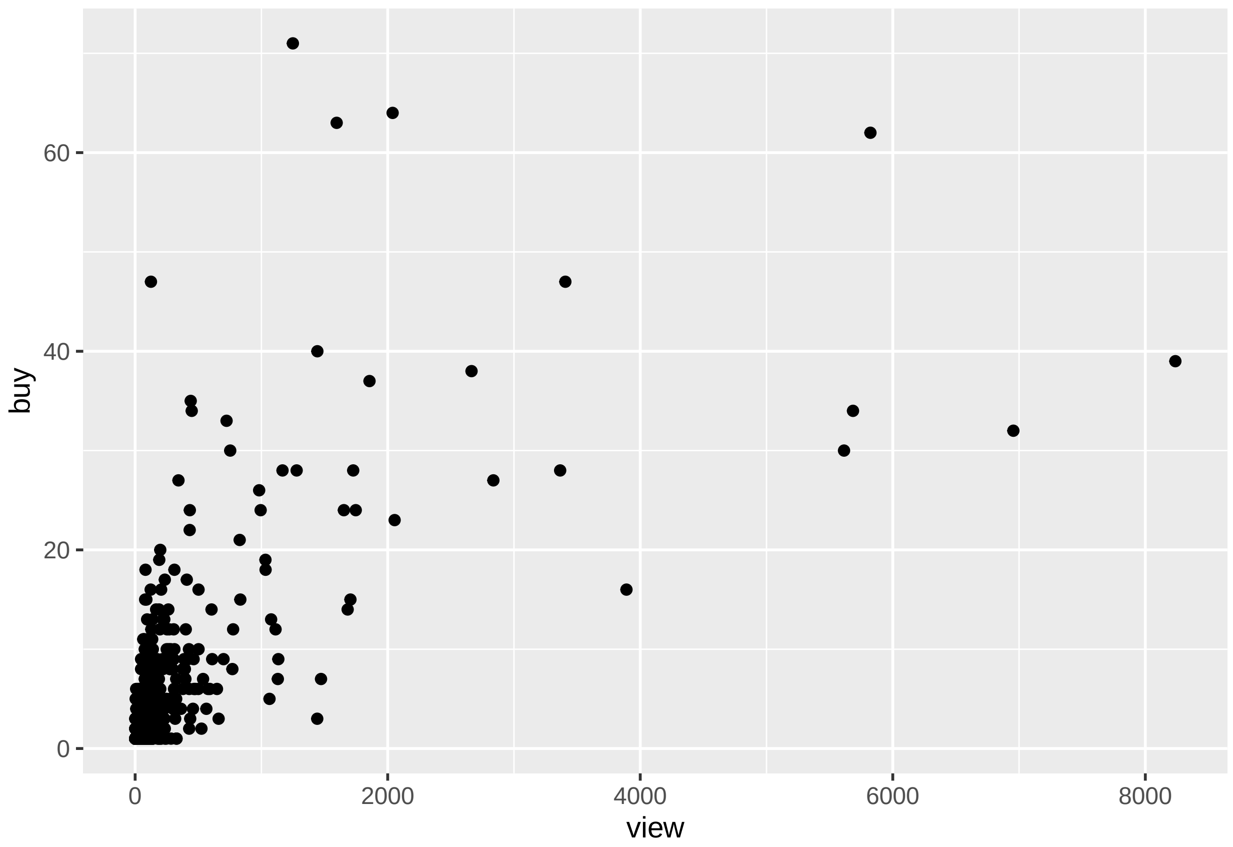
```
##   item_id view buy
## 1 4756105 8238  39
## 2 3607361 6955  32
## 3 4145813 5823  62
## 4  982926 5685  34
## 5 2355072 5614  30
## 6 2520377 3891  16
```

```
pairs(view_buy) #view the plot between view and buy
```



```
view_buy%>%
  ggplot(aes(view,buy))+geom_point()
```

```
## Warning: Removed 2616 rows containing missing values (geom_point).
```

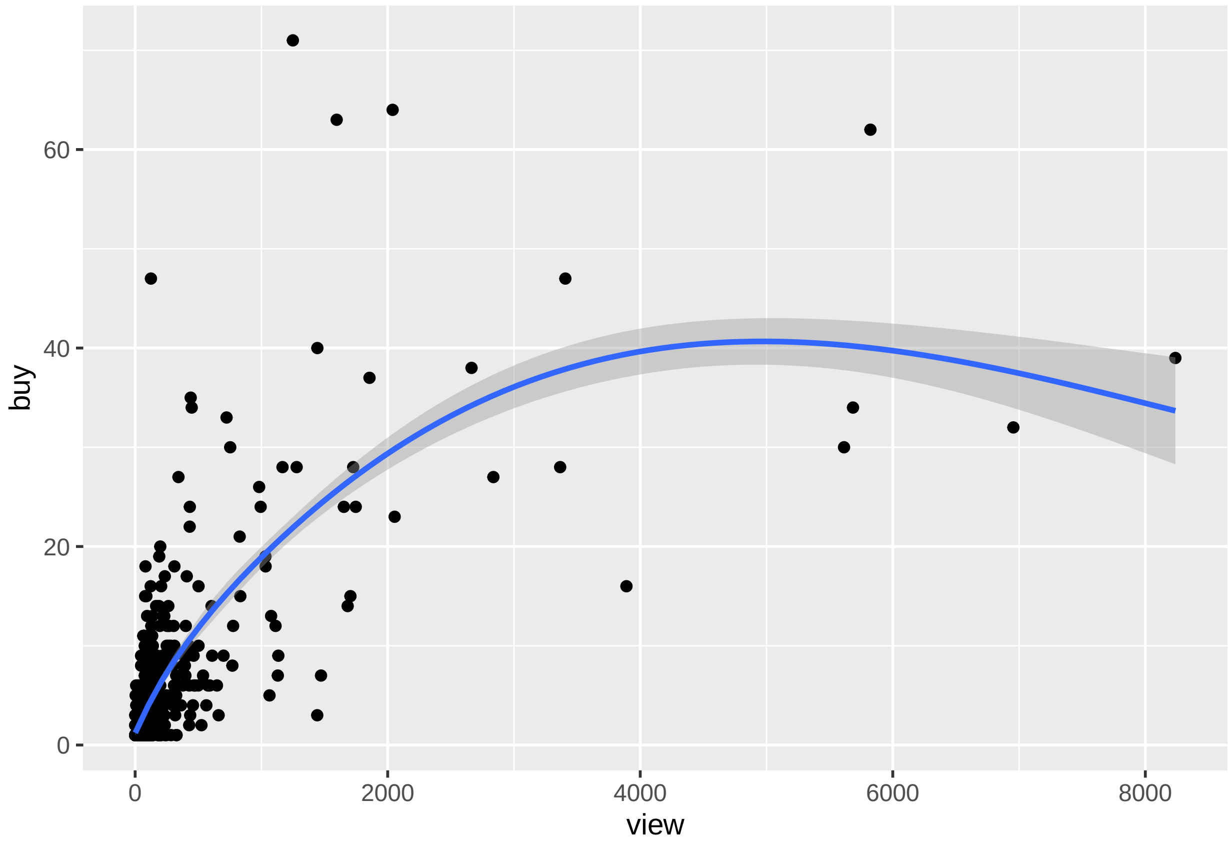


```
view_buy%>%
  ggplot(aes(view,buy))+geom_point()+geom_smooth() # graph does not show the linear relationship betwee
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

## Warning: Removed 2616 rows containing non-finite values (stat_smooth).

## Warning: Removed 2616 rows containing missing values (geom_point).
```



RFM model

Since the database miss the date values. we cannot count the intervel of the time for the recent buyer. However, we are counting the frequency of the buyer and set up the standard as 3. The further research, we can add the date values, then combine with the frequecy data frame to build up RFM model. Group our customers into 4 major types and provide the sales strategy.

```
ub%>%
  filter(behavior=="buy")%>%
  group_by(user_id)%>%
  count()%>%
  arrange(desc(n)) # amount 4007 as buy, the most frequency of buyer is 6, then minimum is 1. so we can
```

```
## # A tibble: 4,007 x 2
## # Groups:   user_id [4,007]
##    user_id     n
##      <int> <int>
##  1 4157341     6
##  2 1542908     5
##  3  667682     4
##  4  855191     4
##  5 1095113     4
##   6 1910706     4
##   7 4395247     4
##   8     322     3
##   9   62002     3
## 10  166219     3
## # ... with 3,997 more rows
```

Long Tail Theory

```
item_fre<-ub%>%
  filter(behavior=="buy")%>%
  group_by(item_id)%>%
  count()%>%
  arrange(desc(n))
head(item_fre) #after counting we can see the most sales of item only 71 times. the total buy was 4007.
```

```
## # A tibble: 6 x 2
## # Groups:   item_id [6]
##   item_id     n
##     <int> <int>
## 1 1464116    71
## 2 2735466    64
## 3 2885642    63
## 4 4145813    62
## 5  901282    47
## 6 4801426    47
```