Predicting Red Hat Business Value

Red Hat company collected lots of data to predict which individuals may be their possible customs. This dataset is constitute with individuals and actions.

```r
library(ggplot2)
library(dplyr)
```

Data Overview

Activity

```r
act_train <- read.csv('act_train.csv')
act_test <- read.csv('act_test.csv')
act_train$date <- as.Date(act_train$date)
act_test$date <- as.Date(act_test$date)
str(act_train)
```

```
## 'data.frame':    2197291 obs. of  15 variables:
##  $ people_id        : Factor w/ 151295 levels "ppl_100","ppl_100002",..: 1 1 1 1 1 1 2 2 3 3 ...
##  $ activity_id      : Factor w/ 2197291 levels "act1_100","act1_100001",..: 503692 832760 1289704 14
##  $ date             : Date, format: "2023-08-26" "2022-09-27" ...
##  $ activity_category: Factor w/ 7 levels "type 1","type 2",..: 4 2 2 2 2 4 2 2 2 2 ...
##  $ char_1           : Factor w/ 52 levels "","type 1","type 10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_2           : Factor w/ 33 levels "","type 1","type 10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_3           : Factor w/ 12 levels "","type 1","type 10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_4           : Factor w/ 8 levels "","type 1","type 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_5           : Factor w/ 8 levels "","type 1","type 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_6           : Factor w/ 6 levels "","type 1","type 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_7           : Factor w/ 9 levels "","type 1","type 2",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_8           : Factor w/ 19 levels "","type 1","type 10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_9           : Factor w/ 20 levels "","type 1","type 10",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ char_10          : Factor w/ 6516 levels "","type 1","type 10",..: 5384 2 2 2 2 764 2 2 2 2 ...
##  $ outcome          : int  0 0 0 0 0 0 1 1 1 1 ...
```

```r
summary(act_train)
```

```
##       people_id           activity_id           date
##  ppl_294918:  55103   act1_100   :      1   Min.   :2022-07-17
##  ppl_370270:  53668   act1_100001:      1   1st Qu.:2022-10-14
##  ppl_105739:  45936   act1_100005:      1   Median :2022-12-27
##  ppl_54699 :  23969   act1_100008:      1   Mean   :2023-01-10
##  ppl_64887 :   7052   act1_100011:      1   3rd Qu.:2023-04-01
##  ppl_250020:   4293   act1_100012:      1   Max.   :2023-08-31
##  (Other)   :2007270   (Other)    :2197285
##  activity_category     char_1             char_2             char_3
##  type 1:157615             :2039676           :2039676           :2039676
##  type 2:904683     type 2 :  38030   type 2 :  50524   type 1 :  38224
```

```
##  type 3:429408     type 5 :  34509   type 5 :  31794   type 5 :  35488
##  type 4:207465     type 1 :  14938   type 1 :  21616   type 4 :  20466
##  type 5:490710     type 12:  14917   type 3 :   9810   type 3 :  19637
##  type 6:  4253     type 3 :  12372   type 16:   7551   type 6 :  19631
##  type 7:  3157     (Other):  42849   (Other):  36320   (Other):  24169
##      char_4           char_5           char_6           char_7
##         :2039676         :2039676         :2039676         :2039676
##  type 3 :  98131   type 6 :  67989   type 1:  48658   type 1 :  52548
##  type 1 :  27979   type 1 :  49214   type 2:  61026   type 3 :  42968
##  type 4 :  13730   type 2 :  26982   type 3:  46124   type 2 :  32199
##  type 2 :   9316   type 3 :   6013   type 4:   1241   type 6 :  10604
##  type 5 :   5520   type 5 :   5421   type 5:    566   type 4 :   8751
##  (Other):   2939   (Other):   1996                    (Other):  10545
##      char_8           char_9           char_10          outcome
##         :2039676         :2039676   type 1  :904683   Min.   :0.000
##  type 4 :  77460   type 8 :  31794   type 23 :200408   1st Qu.:0.000
##  type 5 :  12396   type 1 :  24765           :157615   Median :0.000
##  type 1 :  11621   type 2 :  13488   type 2  :116191   Mean   :0.444
##  type 6 :  10322   type 6 :  12824   type 61 : 35417   3rd Qu.:1.000
##  type 7 :   7737   type 5 :  11021   type 452: 23513   Max.   :1.000
##  (Other):  38079   (Other):  63723   (Other) :759464
```

There are over 2190000 observations and 15 variables. 14 of them are category variables except 'outcome'. According to summary, some variables have unbalanced distribution. A specific type often takes up large space. For example, 'ppl__294918' and 'ppl370270' repeated over 50000 times.
On the other hand, the number of NA are same from char1 to char9. RedHat mentioned that type1 activities have char1-char9 and they are exclusive with char10.
Traning data are collected in one year: from 7.11.2022 to 8.31.2023. It's valuable to compare this feature with testset.

People

```r
people <- read.csv("people.csv")
str(people)
```

```
## 'data.frame':    189118 obs. of  41 variables:
##  $ people_id: Factor w/ 189118 levels "ppl_100","ppl_100002",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ char_1   : Factor w/ 2 levels "type 1","type 2": 2 2 2 2 2 2 2 2 2 2 ...
##  $ group_1  : Factor w/ 34224 levels "group 1","group 10",..: 5275 33211 17984 9731 31507 11977 5275
##  $ char_2   : Factor w/ 3 levels "type 1","type 2",..: 2 3 3 3 3 3 2 3 3 3 ...
##  $ date     : Factor w/ 1196 levels "2020-05-18","2020-05-19",..: 406 232 752 792 799 878 835 980 10
##  $ char_3   : Factor w/ 43 levels "type 1","type 10",..: 39 21 34 35 35 40 42 34 35 6 ...
##  $ char_4   : Factor w/ 25 levels "type 1","type 10",..: 21 25 24 18 18 22 23 24 18 22 ...
##  $ char_5   : Factor w/ 9 levels "type 1","type 2",..: 5 5 5 9 9 4 8 4 9 8 ...
##  $ char_6   : Factor w/ 7 levels "type 1","type 2",..: 3 3 2 4 3 1 1 1 3 3 ...
##  $ char_7   : Factor w/ 25 levels "type 1","type 10",..: 3 3 21 8 24 1 23 23 25 25 ...
##  $ char_8   : Factor w/ 8 levels "type 1","type 2",..: 2 2 2 2 2 2 1 2 3 6 ...
##  $ char_9   : Factor w/ 9 levels "type 1","type 2",..: 2 4 2 2 2 2 1 3 3 6 ...
##  $ char_10  : Factor w/ 2 levels "False","True": 2 1 2 2 1 2 1 2 1 1 ...
##  $ char_11  : Factor w/ 2 levels "False","True": 1 1 2 2 1 2 1 1 1 1 ...
##  $ char_12  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 2 1 1 ...
##  $ char_13  : Factor w/ 2 levels "False","True": 2 2 2 2 1 2 1 2 1 1 ...
```

```
##  $ char_14  : Factor w/ 2 levels "False","True": 2 1 2 2 1 2 1 2 1 1 ...
##  $ char_15  : Factor w/ 2 levels "False","True": 1 1 2 1 1 2 1 2 1 1 ...
##  $ char_16  : Factor w/ 2 levels "False","True": 2 1 1 2 1 2 1 2 1 1 ...
##  $ char_17  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 2 1 1 ...
##  $ char_18  : Factor w/ 2 levels "False","True": 1 1 1 2 1 1 1 2 1 1 ...
##  $ char_19  : Factor w/ 2 levels "False","True": 1 1 2 2 1 2 1 2 1 1 ...
##  $ char_20  : Factor w/ 2 levels "False","True": 1 1 1 2 1 2 1 2 1 1 ...
##  $ char_21  : Factor w/ 2 levels "False","True": 2 1 2 2 1 2 1 2 1 1 ...
##  $ char_22  : Factor w/ 2 levels "False","True": 1 1 2 2 1 2 1 2 1 1 ...
##  $ char_23  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 2 1 1 ...
##  $ char_24  : Factor w/ 2 levels "False","True": 1 1 2 1 1 2 1 2 1 1 ...
##  $ char_25  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 1 1 1 ...
##  $ char_26  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 1 1 1 ...
##  $ char_27  : Factor w/ 2 levels "False","True": 2 2 2 2 1 2 1 2 1 1 ...
##  $ char_28  : Factor w/ 2 levels "False","True": 2 1 2 2 1 2 1 2 1 1 ...
##  $ char_29  : Factor w/ 2 levels "False","True": 1 1 1 2 1 1 1 1 1 1 ...
##  $ char_30  : Factor w/ 2 levels "False","True": 2 2 1 2 1 2 1 1 1 1 ...
##  $ char_31  : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 1 1 ...
##  $ char_32  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 2 1 1 ...
##  $ char_33  : Factor w/ 2 levels "False","True": 1 2 2 2 1 2 1 2 1 1 ...
##  $ char_34  : Factor w/ 2 levels "False","True": 2 2 2 2 1 2 1 2 1 1 ...
##  $ char_35  : Factor w/ 2 levels "False","True": 2 2 1 2 2 2 1 1 1 1 ...
##  $ char_36  : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 1 1 ...
##  $ char_37  : Factor w/ 2 levels "False","True": 1 1 2 2 1 2 1 2 1 1 ...
##  $ char_38  : int  36 76 99 76 84 90 2 91 84 76 ...
```

```r
people$date <- as.Date(people$date)
```

By observation, we can see information contained in this dataset can divided into 5 parts basically.
1. "group_1": This name is very different from others. It's also a category variable containing over 34000 levels. 2. "char1"-"char9": Those variables are very similar to variables in act dataset. 3. "char10"-"char37": They are logical variables.
4. "char38": The exclusive continuous feature in whole dataset.
5. "date": date variables.

Single feature investigation

Act–"people_id"

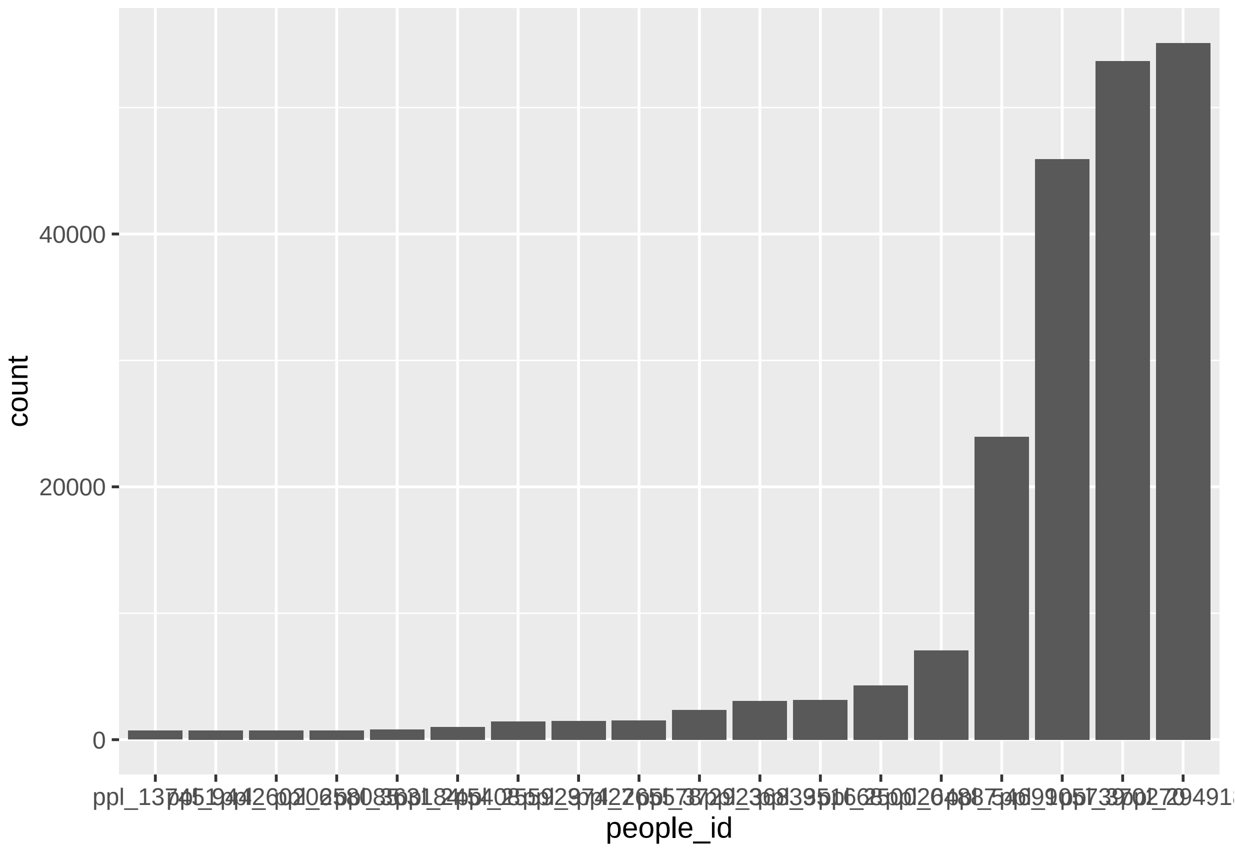
```r
#how many unique people id in training set?
length(unique(act_train$people_id))
```

```
## [1] 151295
```

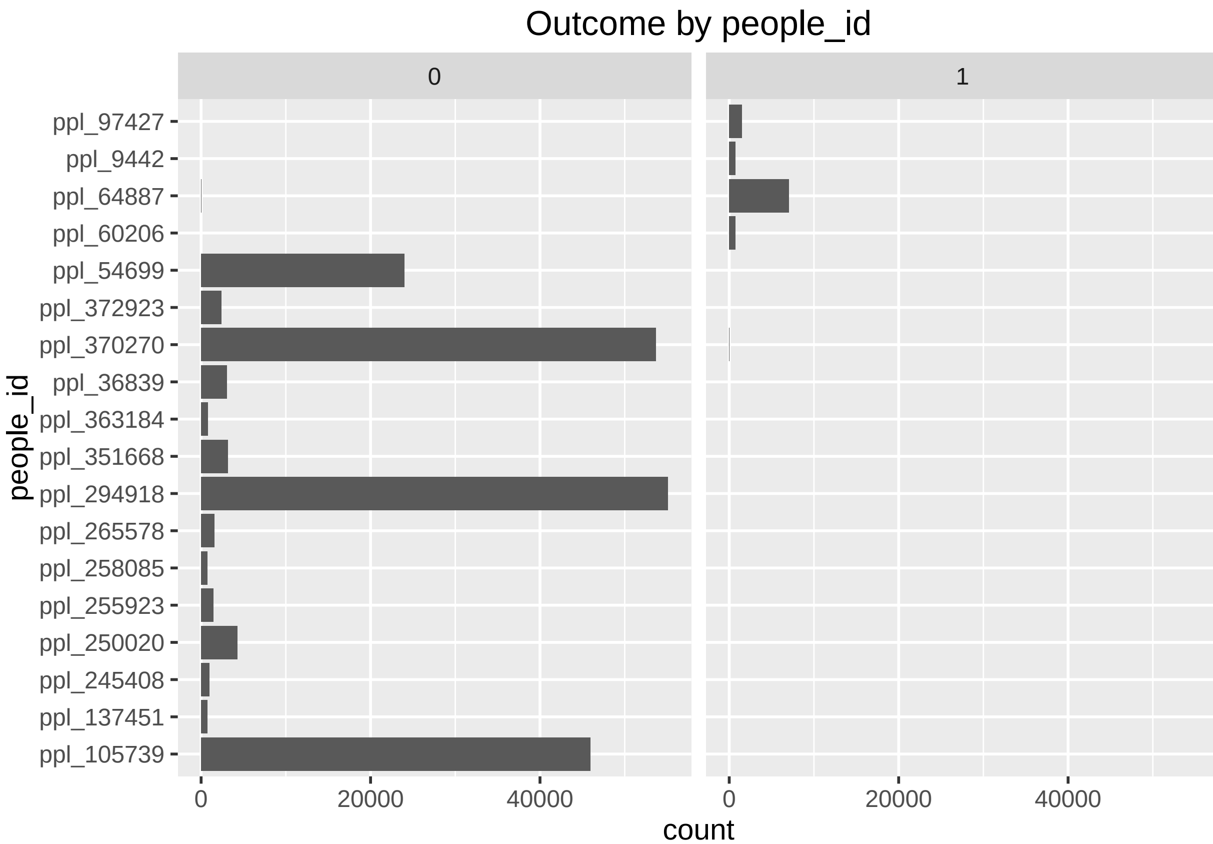
```r
#Do people in training set occured in test set?
sum(unique(act_test$people_id) %in% unique(act_train$people_id))
```

```
## [1] 0
```

```r
#Distribution of people_id
act_train %>%
    count(people_id,sort = TRUE) %>%
    filter(n>700) -> p
p %>%
    ggplot(aes(x= reorder(people_id,n), y= n))+
    geom_bar(stat = "identity")+
    xlab('people_id')+
    ylab("count")
```




```r
act_train %>%
    filter(people_id %in% p$people_id) %>%
    ggplot(aes(x =people_id)) +
    geom_bar() +
    coord_flip() +
    facet_wrap( ~ outcome) +
    ggtitle("Outcome by people_id")
```



151295 people generate 2197291 observations. People in training set are completely different with people in test set.
In this long-tailed bar plot, we knew some people highly repeated. Does same person always has same outcome? If doesn't, I need to figure out which features may drive the difference.

```
#Does a person's outcome change?
sum(filter(act_train, people_id == "ppl_294918")['outcome'])
```

```
## [1] 0
```

```
sum(filter(act_train, people_id == "ppl_370270")['outcome'])
```

```
## [1] 12
```

We choose people occured most to explore this question and it's lucky that we get result on the second person. The outcome will change indeed. This fact brought another question: Which feature drive this difference? Basically, information in act dataset can be divided into two parts: 'date' and 'char'. It's reasonable to make a hypothesis that people may change their decision in different time. Does this difference occured because of date?
Let's make a experiment on "ppl_370270":

```
one <- filter(act_train,people_id == "ppl_370270",outcome == 1)
zero <- filter(act_train,people_id == "ppl_370270",outcome == 0)
#Structure of date of different outcome
min(one$date)
```

```
## [1] "2022-07-21"
```

```r
max(one$date)
```

```
## [1] "2022-10-13"
```

```r
min(zero$date)
```

```
## [1] "2022-10-14"
```

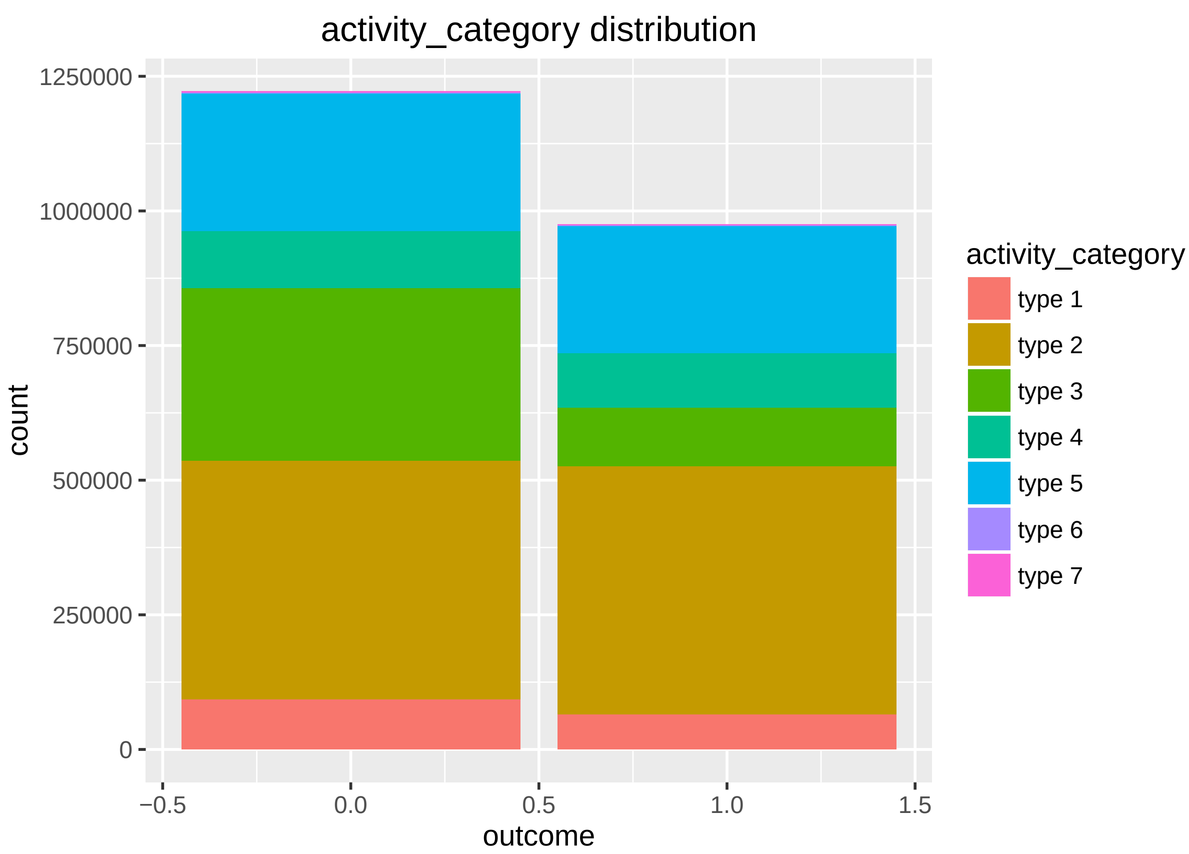
```r
max(zero$date)
```

```
## [1] "2023-06-16"
```

Interesting, It's clear that outcomes of "ppl_370270" turn to 0 suddenly after 2022-10-13. It's just like a timeline. So we can make a assumption that there may exist timeline for different people where outcome will change.
Based on experience in real life and this assumption, we can assume this change will occur limited times.

Act–"activity_category"

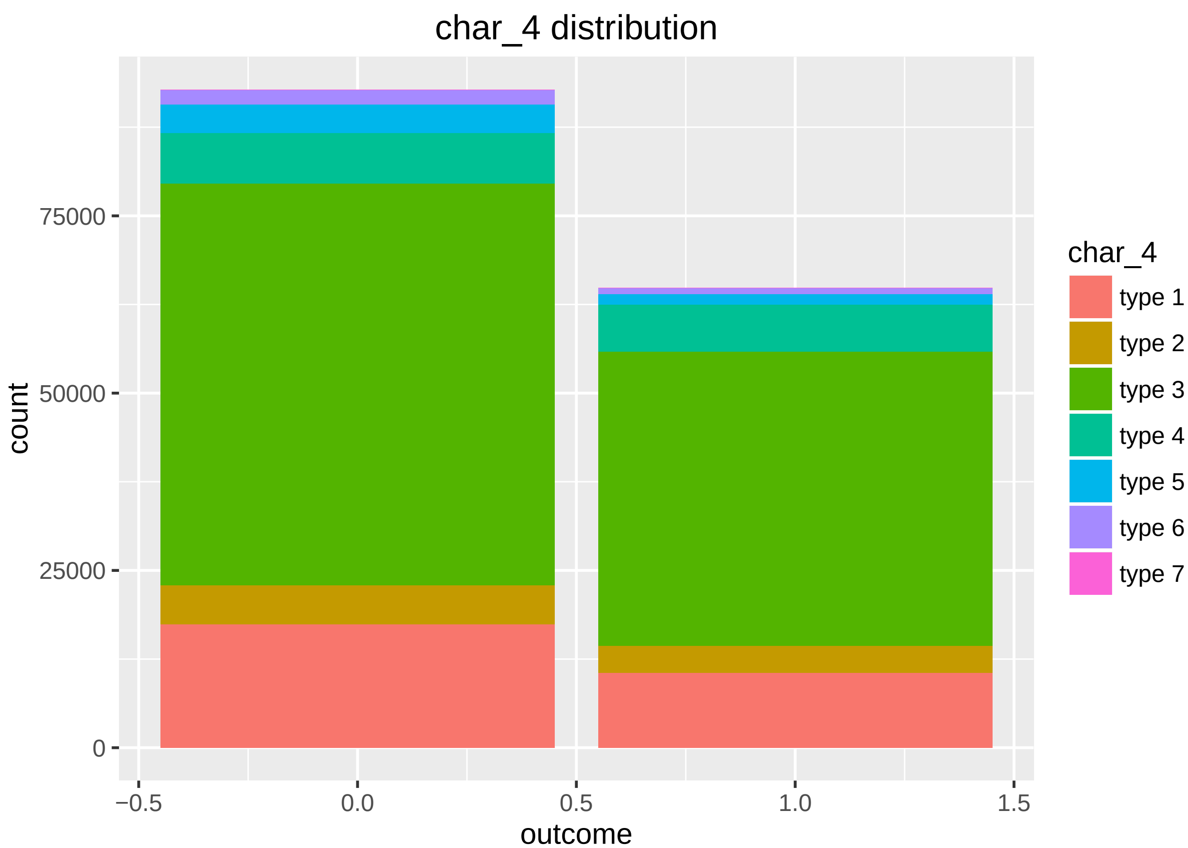
```r
act_train %>%
    ggplot(aes(x = outcome, fill = activity_category))+
    geom_bar()+
    ggtitle("activity_category distribution")
```



It's clear that distributions of activity types are different in two outcome groups especially for "type 2", "type 3" and "type 7".

Act–"char1-char9"

```
act_train %>%
    filter(char_4 != "") %>%
    ggplot(aes(x = outcome, fill = char_4))+
    geom_bar()+
    ggtitle("char_4 distribution")
```



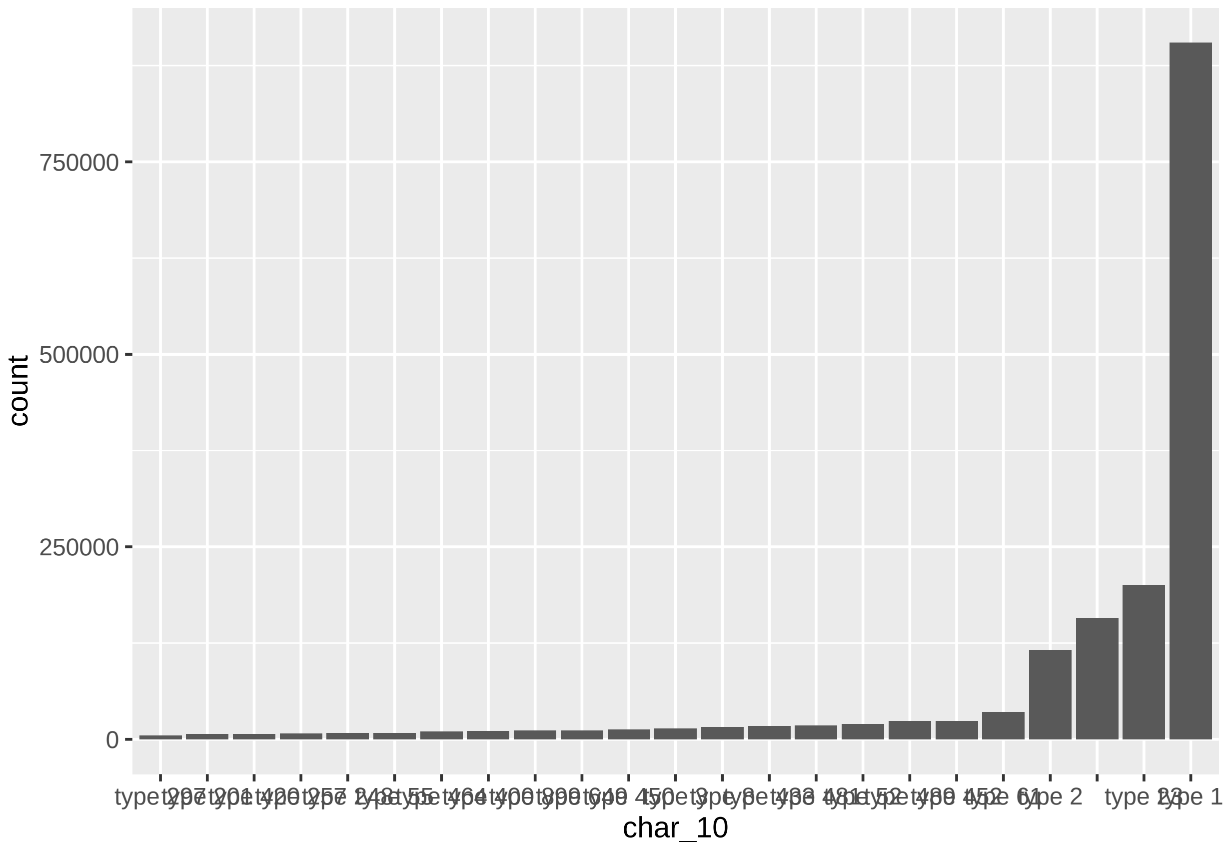
Even though char1-char9 have different levels, but the distributions of them are similar in different groups. We plot barplot of 'char_4' as example. Observing this plot, the right bar is just like the shrinked left bar. Very limited information can be extracted from this plot.

Act–"char10"

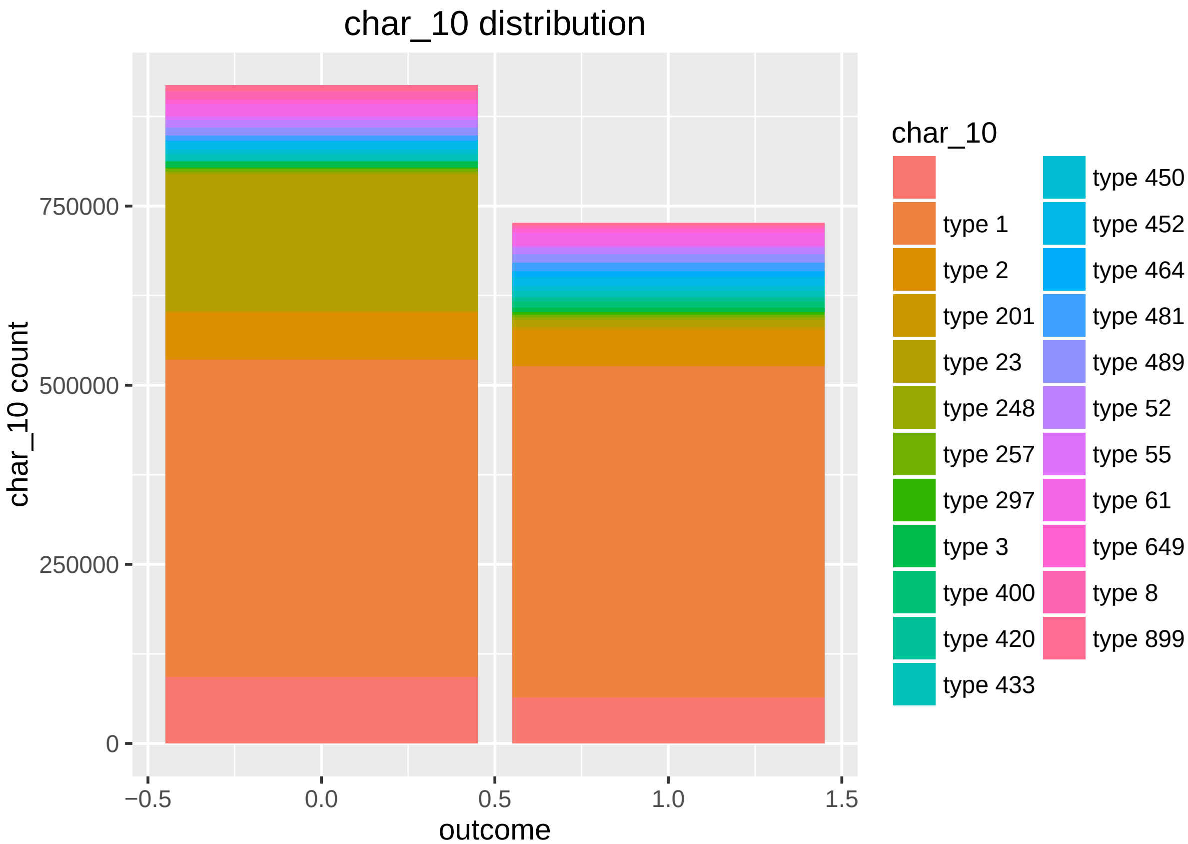
```
#unique number of char_10
length(unique(act_train$char_10))
```

```
## [1] 6516
```

```r
act_train %>%
    count(char_10,sort = TRUE) %>%
    filter(n>5000) -> p
p %>%
    ggplot(aes(x= reorder(char_10,n), y= n))+
    geom_bar(stat = "identity")+
    xlab('char_10')+
    ylab("count")
```




```r
act_train %>% filter(char_10 %in% p$char_10) %>%
    ggplot(aes(x = outcome, fill = char_10))+
    geom_bar()+
    ylab("char_10 count")+
    ggtitle("char_10 distribution")
```



There are two many type for plotting. We only plot type which occured over 5000 times. It's still a long-tailed data. We can find most type 23 belong to group 0.

Act&people–date

```
#merge act and people
act_train %>%
    merge(people, all.x = TRUE, by = "people_id") -> data

#count the row act_date less than people_date
sum(data$date.x < data$date.y)
```

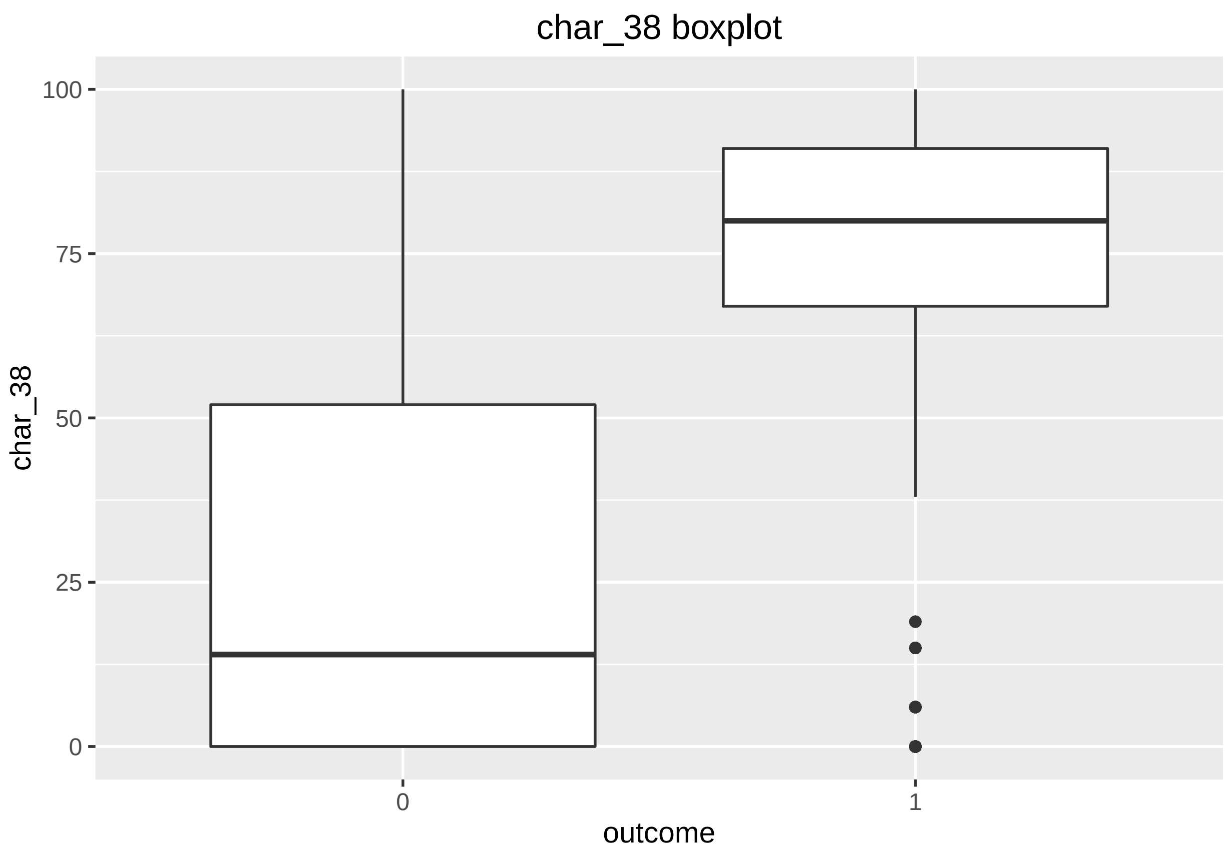
```
## [1] 0
```

All activity occured after people date. It provide a idea that the difference between people date and activity date may provide important information when we process data later.

People–char_38

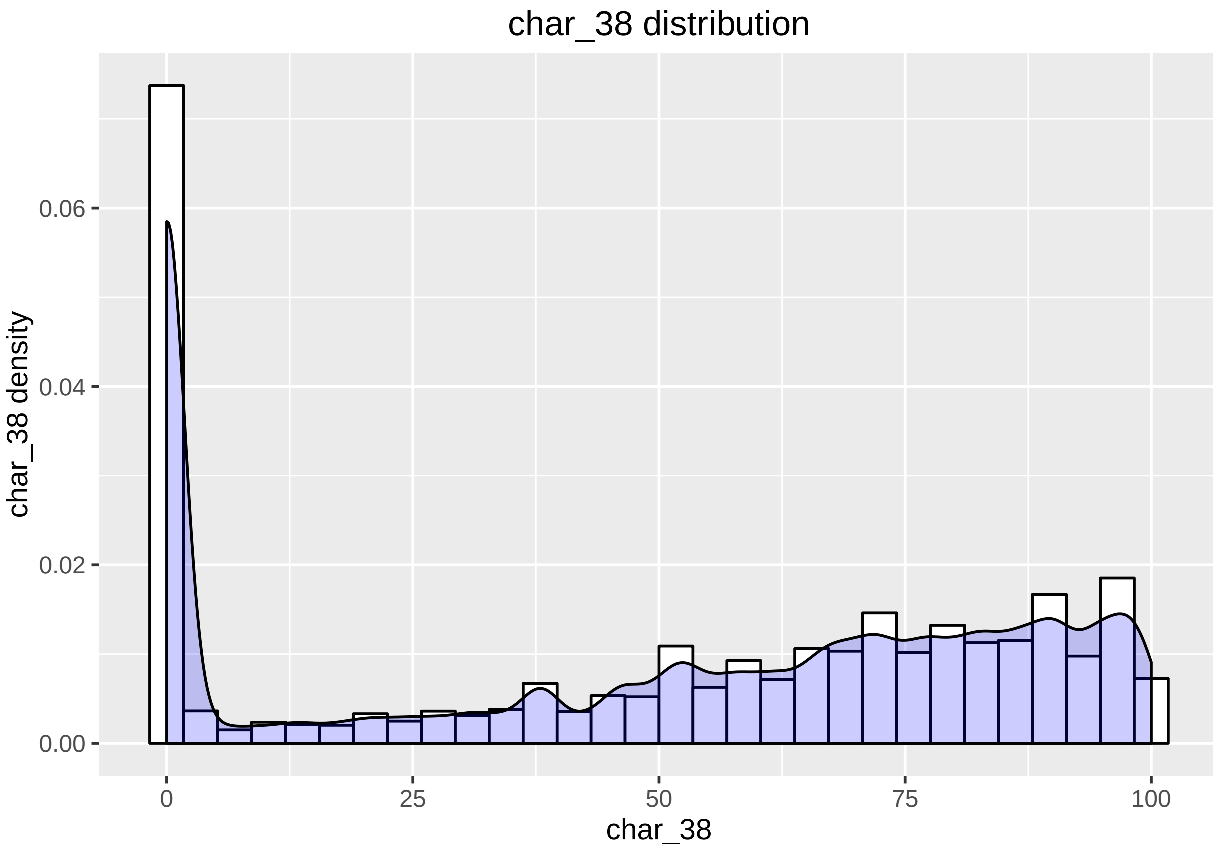
```
data %>%
    ggplot(aes(y = char_38, x = as.factor(outcome) ))+
    geom_boxplot()+
```

```
    ggtitle("char_38 boxplot")+
    xlab("outcome")
```



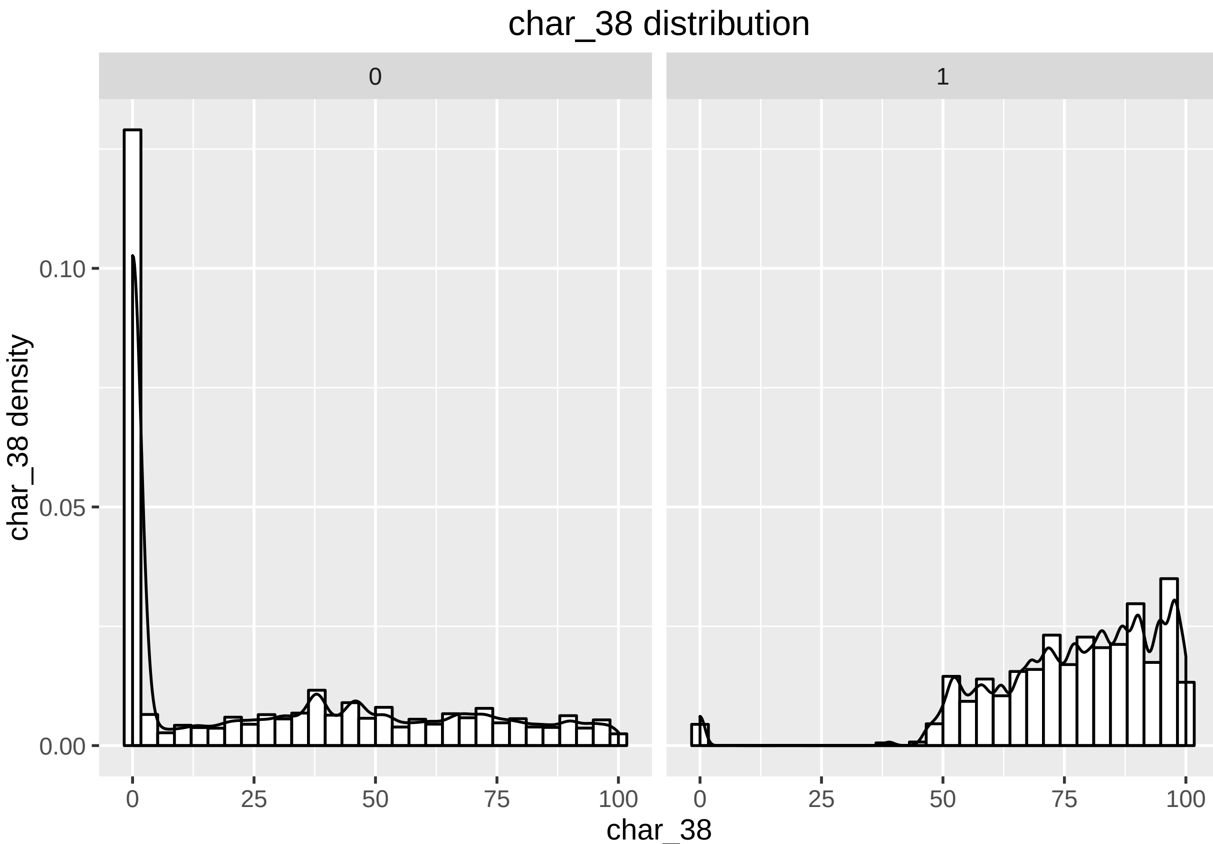

```
data %>%
    ggplot(aes(x = char_38))+
    geom_histogram(aes(y=..density..), fill = "white", color = "black")+
    geom_density(fill = "blue", alpha = 0.2)+
    ylab("char_38 density")+
    xlab("char_38")+
    ggtitle("char_38 distribution")
```

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




```
data %>%
    ggplot(aes(x = char_38))+
    geom_histogram(aes(y=..density..), fill = "white",  color = "black")+
    geom_density()+
    facet_wrap(~outcome)+
    ylab("char_38 density")+
    xlab("char_38")+
    ggtitle("char_38 distribution")
```

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



From boxplot, we can find the distribution of char_38 are highly skewed. We need to process it before applying classifier.In group 1, there are some outliers.

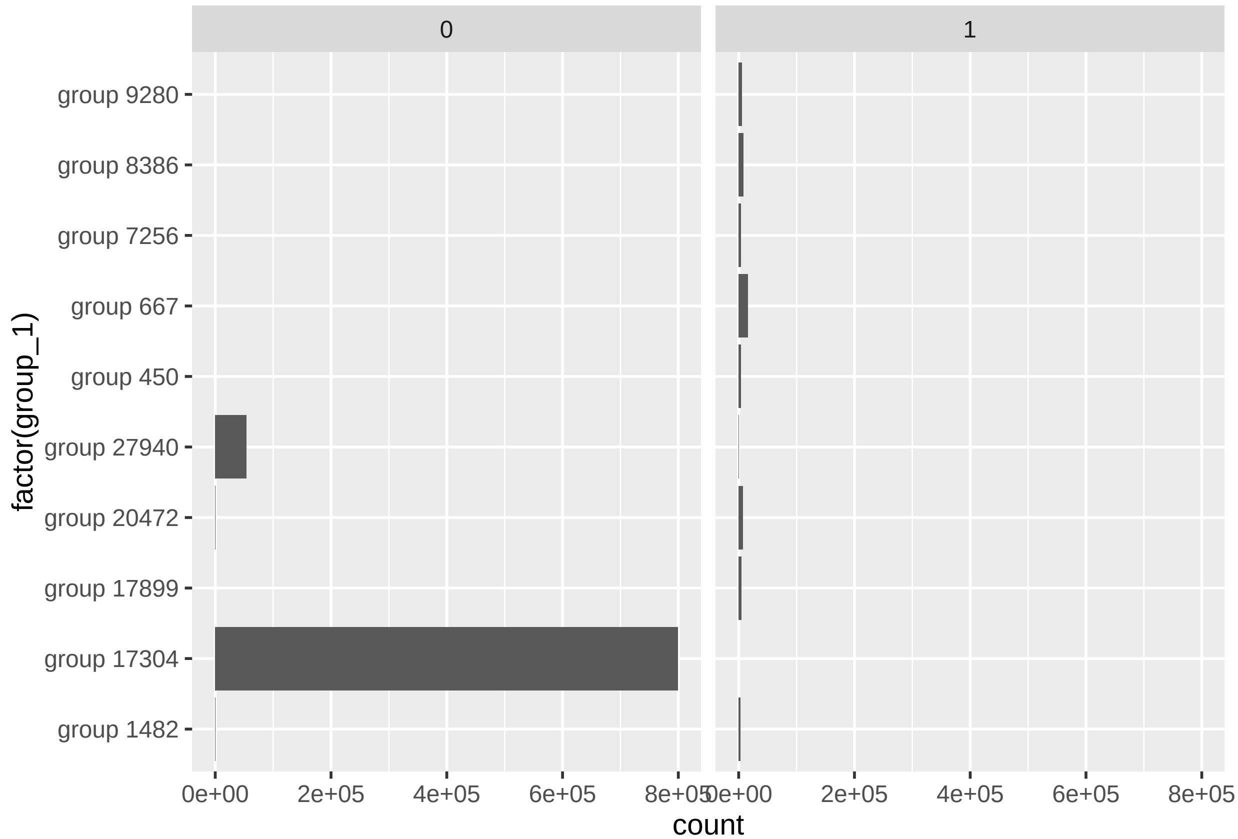
People–group_1

```
#the number of group 1
length(unique(people$group_1))
```

```
## [1] 34224
```

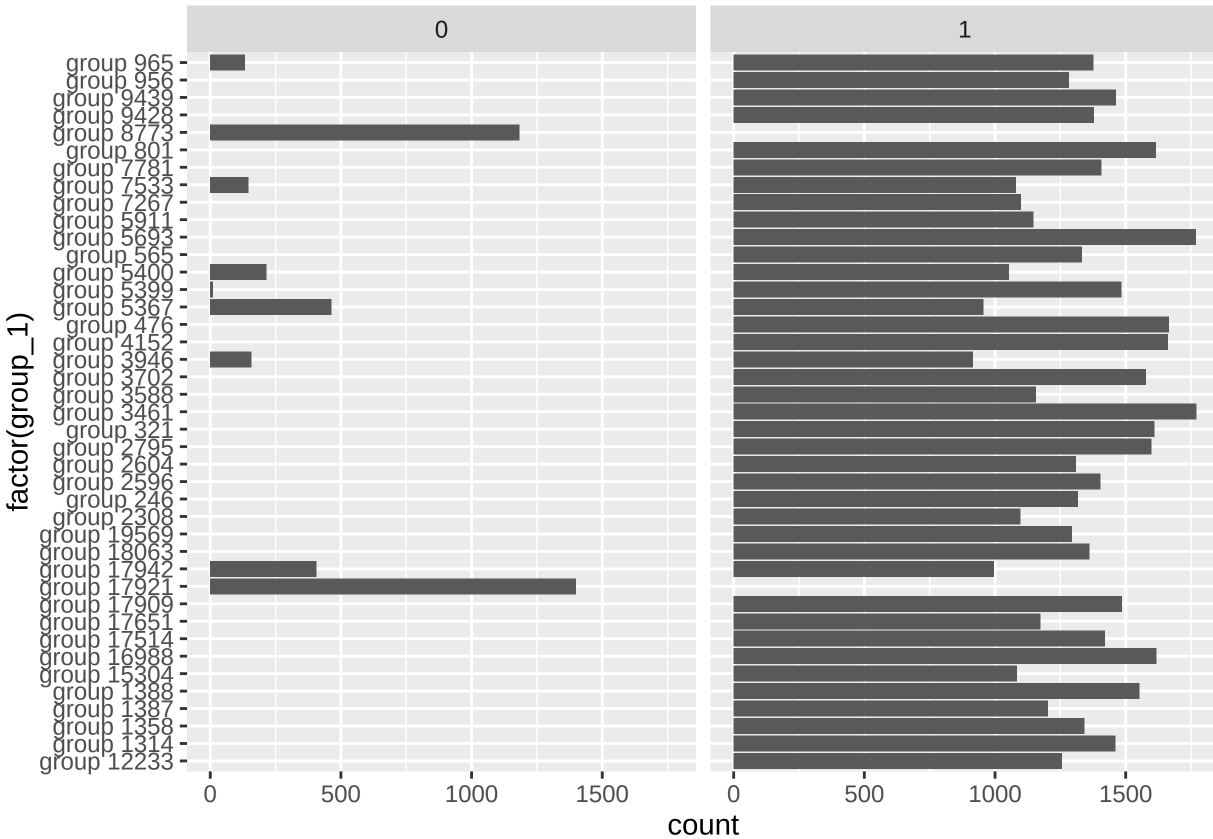
```
data %>%
    count(group_1, sort = TRUE) -> p

data %>%
    filter(group_1 %in% p$group_1[1:10]) %>%
    ggplot(aes(factor(group_1) ))+
    geom_bar()+
    coord_flip()+
    facet_wrap(~outcome)+
    ylab("count")
```



Wow, it's interesting that group 17304 and group 27940 only contain outcome 0 and a lot of data come from them. We also notice that some group only belong to group 1.

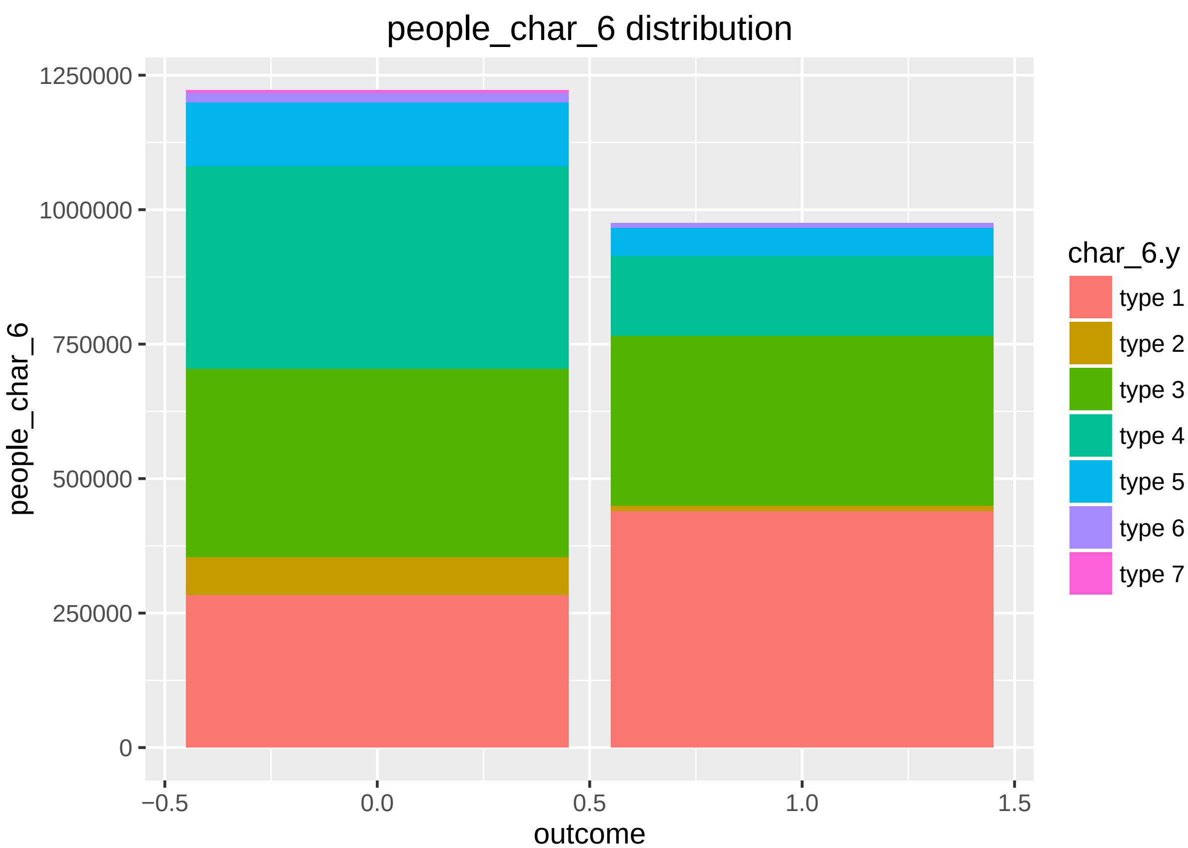
```
data %>%
    filter(group_1 %in% p$group_1[30:70]) %>%
    ggplot(aes(factor(group_1) ))+
    geom_bar()+
    coord_flip()+
    facet_wrap(~outcome)+
    ylab("count")
```



We checked other group which prove our assumption. We can divide group_1 into 3 groups:
1. Only has result 0.
2. Only has result 1.
3. Has mixed result.
We believe that this is a important finding which will bring great help to build our model later.

People–rest of features

```
data %>%
    ggplot(aes(x = outcome, fill = char_6.y))+
    geom_bar()+
    ylab("people_char_6")+
    ggtitle("people_char_6 distribution")
```



For people, char1 to char9 havesimilar distribution, the value can affect outcome but we can see they're not as important as group1.