```
# math189 final

setwd("C:\\Users\\hawl1\\Desktop")
library(data.table)
mydata<-fread("translate.csv")
picdata<-fread("imginfo.csv")

L = 1000 # number of samples
l = 914  # number of picdata


######
## deal prob
data.prob = mydata$deal_probability;data.prob
hist(data.prob,labels=T,ylim=c(0,800))
summary(data.prob)
#   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.0000  0.0000  0.0000  0.1390  0.1479  1.0000
hist(data.prob, labels = T, ylim = c(0,710))
hist(data.prob[which(data.prob<0.1)], labels = T, ylim = c(0,640))



#######
## price
data.Price.0 = mydata$price
ind.throw = c(which(data.Price.0 ==0),which(is.na(data.Price.0)));ind.throw
data.Price = data.Price.0[-ind.throw]
data.Price

length(data.Price)                  # 932 informative points
Price.row = seq(1,L,1)[-ind.throw]  # price informative rows

hist(data.Price,breaks=30)
hist(data.Price[which(data.Price<=0.5e07)],breaks=30)
hist(data.Price[which(data.Price<=10000)],breaks=30, labels = T)

summary(data.Price)
# Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
# 1      500     1500    362671    10000  36000000

#may be a GoF test to see this is exponential



######
## date
data.Date = mydata$activation_date
data.Date

length(data.Date)                   # 1000
string.date = numeric(L)
for(i in 1:L){
  date = data.Date[i]
  # split the word and only take the last two numbers as the date of March
  string.date[i] = as.numeric(sapply(strsplit(date, "-"), "[[", 3))
}
string.date
table(string.date)
# Month day  15 16 17 18 19 20 21 22 23 24 25 26 27 28
# Counts    66 82 68 85 68 75 56 77 69 64 68 74 75 73

plot.monthday = barplot(table(string.date), ylim = c(0,90),
                        main = "Barplot of activation date (month day)")
text(plot.monthday, 5, labels = as.matrix(table(string.date)))


week.day = weekdays(as.Date(data.Date))
week.day

length(week.day)                    # 1000
table(week.day)
# Friday    Monday  Saturday    Sunday  Thursday   Tuesday Wednesday
# 132       150       153       142       151       129       143
week.table = data.frame(Mon=150, Tue=129, Wed=143, Thu=151, Fri=132,
                        Sat=153, Sun=142) # reorder the data

plot.weekday = barplot(as.matrix(week.table), ylim = c(0,160),
                       main = "Barplot of activation date (week day)")
text(plot.weekday, 5, labels = as.matrix(week.table))


## list of the activation_date is weekend or not
data.IsWeekend=rep(9,1000)
for(i in 1:L){
  if (week.day[i]=="Saturday" || week.day[i]=="Sunday"){
    data.IsWeekend[i] = 1
  }
  else{
    data.IsWeekend[i] = 0
  }
}
data.IsWeekend
table(data.IsWeekend)
# 705 weekdays(Mon-Fri)
# 295 weekends
isweekend.table = data.frame(weekdays=705, weekends=295)

plot.isweekend = barplot(as.matrix(isweekend.table), ylim = c(0,710),
                         main = "Barplot of activation date (weekdays/weekends)" )
text(plot.isweekend, 40, labels = as.matrix(isweekend.table))



######
## User type
data.user = mydata$user_type
data.user
table(data.user)
# Company Private    Shop
# 271     681        48

plot.usertype = barplot(table(data.user), ylim = c(0,690),
                        main = "Barplot of user type")
text(plot.usertype, 20, labels = as.matrix(table(data.user)))


######
## With or without image
data.image = mydata$image
data.withimage=rep(9,1000)
for(i in 1:L){
  if (data.image[i] == ""){
    data.withimage[i] = 0
  }
  else{
    data.withimage[i] = 1
  }
}
data.withimage
table(data.withimage)

withimage.table = data.frame(with=914, without=86)

plot.withimage = barplot(as.matrix(withimage.table), ylim = c(0,920),
                         main = "Barplot of using of image" )
text(plot.withimage, 40, labels = as.matrix(withimage.table))



######
## text
#~~~ Word count function
wordcount <- function(str) {
  sapply(gregexpr("\\b\\W+\\b", str, perl=TRUE), function(x) sum(x>0) ) + 1
}
#~~~ End


## description
data.description = mydata$en_desc

# word counts
data.des.WordCount = rep(0,L)
for(i in 1:L){
  # data.des.WordCount[i]= lengths(gregexpr("\\W+", data.description[i]))
  if (data.description[i] != "missing"){
    data.des.WordCount[i]= wordcount(data.description[i])
  }
}
data.des.WordCount
summary(data.des.WordCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.00    7.00   17.00   31.97   36.00  384.00
hist(data.des.WordCount,labels=T,ylim=c(0,900))
hist(data.des.WordCount[which(data.des.WordCount<50)],labels=T,ylim=c(0,900))

# Capslock counts
data.des.CapsCount = rep(0,L)
for(i in 1:L){
  data.des.CapsCount[i]= sapply(gregexpr("\\b[A-Z]{2,}\\b", data.description[i]),
                                function(x) length(c(x[x > 0])))
}
data.des.CapsCount
summary(data.des.CapsCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.000   0.000   0.000   0.905   0.000  39.000
hist(data.des.CapsCount, labels = T, ylim = c(0,980))
hist(data.des.CapsCount[which(data.des.CapsCount<=5)], labels = T, ylim = c(0,980))


# Digit count
data.des.DigitCount = rep(0,L)
for(i in 1:L){
  Digits<-gsub("[^0-9 ]","",data.description[i])
  # data.des.DigitCount[i]= lengths(gregexpr("\\W+", Digits))
  if (gsub(" ","",Digits) != ""){
    data.des.DigitCount[i]= wordcount(Digits)
  }
}
data.des.DigitCount
summary(data.des.DigitCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.000   0.000   1.000   2.151   2.000  50.000
hist(data.des.DigitCount, labels = T, ylim = c(0,920))
hist(data.des.DigitCount[which(data.des.DigitCount<=5)], labels = T, ylim = c(0,450))



######
## Title
data.title = mydata$en_title

# word counts
data.ti.WordCount = rep(0,L)
for(i in 1:L){
  # data.ti.WordCount[i]= lengths(gregexpr("\\W+", data.title[i]))
  if (data.title[i] != "missing"){
    data.ti.WordCount[i]= wordcount(data.title[i])
  }
}
data.ti.WordCount
summary(data.ti.WordCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 1       2       3       4       5      12
ti.WordCount.table = table(data.ti.WordCount)
plot.ti.WordCount = barplot(ti.WordCount.table, main = "Barplot of word count for title" )
text(plot.ti.WordCount, 4, labels = ti.WordCount.table)

# Capslock counts
data.ti.CapsCount = rep(0,L)
for(i in 1:L){
  data.ti.CapsCount[i]= sapply(gregexpr("\\b[A-Z]{2,}\\b", data.title[i]),
                               function(x) length(c(x[x > 0])))
}
data.ti.CapsCount
summary(data.ti.CapsCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.000   0.000   0.000   0.091   0.000   3.000
ti.CapsCount.table = table(data.ti.CapsCount)
plot.ti.CapsCount = barplot(ti.CapsCount.table, main = "Barplot of capital count for title" )
text(plot.ti.CapsCount, 40, labels = ti.CapsCount.table)

# Digit count
data.ti.DigitCount = rep(0,L)
for(i in 1:L){
  Digits<-gsub("[^0-9 ]","",data.title[i])
  # data.ti.DigitCount[i]= lengths(gregexpr("\\W+", Digits))
  if (gsub(" ","",Digits) != ""){
    data.ti.DigitCount[i]= wordcount(Digits)
  }
}
data.ti.DigitCount
summary(data.ti.DigitCount)
# Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
# 0.000   0.000   0.000   0.603   1.000   6.000
ti.DigitCount.table = table(data.ti.DigitCount)
plot.ti.DigitCount = barplot(ti.DigitCount.table, main = "Barplot of digit count for title" )
text(plot.ti.DigitCount, 40, labels = ti.DigitCount.table)


######
## region and city          ~~~~ this is ugly !!!! ~~~
data.region = mydata$en_region
```

```
data.region
table(data.region)
barplot(prop.table(table(data.region)), cex.names = 0.8, las =2,
        main = "Barplot of region")
#Altai region                    Bashkortostan
#25                               50
#Belgorod region                 Chelyabinsk region
#18                               57
#Irkutsk region                  Kaliningrad region
#30                               23
#Kemerovo Region Khanty-Mansiysk Autonomous Okrug
#21                               16
#Krasnodar region                Krasnoyarsk region
#104                              39
#Nizhny Novgorod Region          Novosibirsk region
#48                               46
#Omsk Region                     Orenburg region
#37                               25
#Perm Region                     Rostov region
#36                               56
#Samara Region                   Saratov region
#48                               31
#Stavropol region                Sverdlovsk region
#34                               57
#Tatarstan                       Tula region
#45                               16
#Tyumen region                   Udmurtia
#23                               16
#Vladimir region                 Volgograd region
#16                               27
#Voronezh region                 Yaroslavl region
#33                               23


data.city = mydata$en_city
data.city
table(data.city)
barplot(prop.table(table(data.city)), cex.names = 0.4, las =2,
        main = "Barplot of city")



######
## Parent category
data.parent.cat = mydata$en_parent_category_name
data.parent.cat
table(data.parent.cat)
# Animals   Consumer electronics           For business  For home and cottages
# 31                 128                     9                   116
# Hobbies and Recreation         Personal things           The property           The services
# 58                    436                    115                    44
# Transport
# 63
plot.parent = barplot(table(data.parent.cat), las =2, main = "Barplot of parent category")
text(plot.parent, 10, labels = as.matrix(table(data.parent.cat)))



######
## Category
data.cat.1 = mydata$en_category_name
data.cat.1
table(data.cat.1)
barplot(prop.table(table(data.cat.1)), cex.names = 0.6, las = 2, main = "Barplot of category")


data.cat.2 = mydata$en_param_1
data.cat.2
ind.throw = which(data.cat.2 == "missing")
data.cat.2 = data.cat.2[-ind.throw]
barplot(prop.table(table(data.cat.2)), cex.names = 0.4, las = 2, main = "Barplot of parameter 1")


data.cat.3 = mydata$en_param_2
data.cat.3
ind.throw = which(data.cat.3 == "missing")
data.cat.3 = data.cat.3[-ind.throw]
barplot(prop.table(table(data.cat.3)), cex.names = 0.4, las = 2, main = "Barplot of parameter 2")


data.cat.4 = mydata$en_param_3
data.cat.4
ind.throw = which(data.cat.4 == "missing")
data.cat.4 = data.cat.4[-ind.throw]
barplot(prop.table(table(data.cat.4)), cex.names = 0.4, las = 2, main = "Barplot of parameter 3")



######
## image
data.image = mydata$image
pic.index = rep(0,1000)
for(i in 1:1){
  for(j in 1:1){
    if(pic$data$image[j]==paste(data.image[i], ".jpg", sep="")){
      pic.index[i] = j
    }
  }
}
pic.index



mydata2<-fread("newdata.csv")
imgdata<-fread("picdata_sort_1000.csv")
mydata$with.image<-rep(1,1000)
for(i in 1:1000)
{
  if (mydata$image[i]=="")
  {
    mydata$with.image[i]<-0
  }
}
##############################
############################
# Private/ Company/shop
data.user
data.company.index<-which(data.user=="Company");data.company.index
data.company.prob<-data.prob[data.company.index];data.company.prob
length(data.company.index)   #271

data.private.index<-which(data.user=="Private");data.private.index
data.private.prob<-data.prob[data.private.index];data.private.prob
length(data.private.index)    #681

data.shop.index<-which(data.user=="Shop");data.shop.index
data.shop.prob<-data.prob[data.shop.index];data.shop.prob
length(data.shop.index)      #48

summary(data.prob)
length(data.prob[data.prob<0.000001])   #627 deal probs are zeros

CD0<-length(data.company.prob[data.company.prob<0.000001])   #184 zero deal for company
CDN0<-length(data.company.prob[data.company.prob>0.000001])  #87 non-zero deal for company

PD0<-length(data.private.prob[data.private.prob<0.000001])   #420 zero deal for private
PDN0<-length(data.private.prob[data.private.prob>0.000001])  #261 non-zero deal for private

SD0<-length(data.shop.prob[data.shop.prob<0.000001])   #23 zero deal for shop
SDN0<-length(data.shop.prob[data.shop.prob>0.000001])  #25 non-zero deal for shop

matrix1<-matrix(c(CD0,CDN0,PD0,PDN0,SD0,SDN0),ncol=2,byrow=T)
matrix1

T1<-as.data.frame(matrix1)
rownames(T1)<-c("Company","Private","Shop");T1
names(T1)<-c("deal prob is 0", "deal prob not 0");T1
chisq.test(T1)
##              deal prob is 0    deal prob not 0
# Company          184               87
# Private          420              261
# Shop              23               25

#X-squared = 7.9213, df = 2, p-value = 0.01905

#####################################
#####################################
## Price (seperated into roughly 5 parts)


length(data.Price.0[is.na(data.Price.0)])    #of 1000 advertisement, 66 prices unavailable
index.no.price<-which(is.na(data.Price.0));index.no.price     #914 data with price
data.with.Price<-data.Price.0[-index.no.price];data.with.Price
length(data.with.Price)
data.prob.with.P<-data.prob[-index.no.price];length(data.prob.with.P)


quantile(data.with.Price,c(0.2,0.4,0.6,0.8))        #prices are seperated into 5 parts

# 20%   40%   60%   80%
# 400  1000  2500 18000

index.price.02<-which(data.with.Price<=400);index.price.02
P02.prob<-data.with.Price[index.price.02];P02.prob
DP02<-data.prob.with.P[index.price.02];length(DP02)        #193 number <=400
length(DP02[DP02<0.000001])                                # 137 deal prob=0
length(DP02[DP02>0.000001])                                #56 deal prob>0

index.price.04<-which(data.with.Price<=1000 & data.with.Price>400);index.price.04
P04.prob<-data.with.Price[index.price.04];P04.prob
DP04<-data.prob.with.P[index.price.04];length(DP04)        #220 number >400 and <=1000
length(DP04[DP04<0.000001])                                # 171 deal prob=0
length(DP04[DP04>0.000001])                                #49 deal prob>0

index.price.06<-which(data.with.Price>1000 & data.with.Price<=2500);index.price.06
P06.prob<-data.with.Price[index.price.06];P06.prob
DP06<-data.prob.with.P[index.price.06];length(DP06)        #152 number >1000 and <=2500
length(DP06[DP06<0.000001])                                # 112 deal prob=0
length(DP06[DP06>0.000001])                                #40 deal prob>0

index.price.08<-which(data.with.Price>2500 & data.with.Price<=18000);index.price.08
P08.prob<-data.with.Price[index.price.08];P08.prob
DP08<-data.prob.with.P[index.price.08];length(DP08)        #183 number >2500 and <=18000
length(DP08[DP08<0.000001])                                # 108 deal prob=0
length(DP08[DP08>0.000001])                                #75 deal prob>0

index.price.10<-which( data.with.Price>18000);index.price.10
P10.prob<-data.with.Price[index.price.10];P10.prob
DP10<-data.prob.with.P[index.price.10];length(DP10)        #186 number >18000
length(DP10[DP10<0.000001])                                # 72 deal prob=0
length(DP10[DP10>=0.000001])                               #114 deal prob>0

matrix2<-matrix(c(length(DP02[DP02<0.000001]),length(DP02[DP02>=0.000001]),
                  length(DP04[DP04<0.000001]),length(DP04[DP04>=0.000001]),
                  length(DP06[DP06<0.000001]),length(DP06[DP06>=0.000001]),
                  length(DP08[DP08<0.000001]),length(DP08[DP08>=0.000001]),
                  length(DP10[DP10<0.000001]),length(DP10[DP10>=0.000001])),ncol=2,byrow=T)
T2<-as.data.frame(matrix2);T2
names(T2)<-c("deal prob is 0", "deal prob not 0");T2
rownames(T2)<-c("<=400",">400 and <=1000",">1000 and <=2500",">2500 and <=18000",">18000")
T2
chisq.test(T2)
# X-squared = 82.092, df = 4, p-value < 2.2e-16

#########################################
#########################################
#########################################
# check some properties of image

index.no.image<-which(mydata$image=="");index.no.image
dataSet.with.image<-mydata[-index.no.image,]
imgdata914<-imgdata[-index.no.image,]

data.blurness<-imgdata914$blurriness;data.blurness
deal.prob914<-dataSet.with.image$deal_probability

summary(data.blurness)
#  Min.     1st Qu.   Median   Mean    3rd Qu.     Max.
#  27.69   279.03   517.91   846.03   935.09    13442.24


index.25<-which(data.blurness<=279.03);length(index.25)  #229 blurness <=279.03
blurness.25<-data.blurness[index.25]
p.blurness.25<-deal.prob914[index.25]

length(p.blurness.25[p.blurness.25<0.000001])          #155 deal prob 0
length(p.blurness.25[p.blurness.25>=0.000001])         #74 deal prob not 0


index.50<-which(data.blurness>279.03 & data.blurness<=517.91 );length(index.50)  #228 blurness >279.03 and <=517.91
blurness.50<-data.blurness[index.50]
p.blurness.50<-deal.prob914[index.50]
length(p.blurness.50[p.blurness.50<0.000001])          #155 deal prob 0
length(p.blurness.50[p.blurness.50>=0.000001])         #73 deal prob not 0

index.75<-which(data.blurness>517.91 & data.blurness<=935.09 );length(index.75)  #228 blurness >517.91 and <=935.09
blurness.75<-data.blurness[index.75]
p.blurness.75<-deal.prob914[index.75]
length(p.blurness.75[p.blurness.75<0.000001])          #147 deal prob 0
length(p.blurness.75[p.blurness.75>=0.000001])         #81 deal prob not 0


index.100<-which(data.blurness>935.09 );length(index.100)  #229 >935.09
blurness.100<-data.blurness[index.100]
p.blurness.100<-deal.prob914[index.100]
```

```
length(p.blurness.100[p.blurness.100<0.000001])          #142 deal prob 0
length(p.blurness.100[p.blurness.100>=0.000001])         #87 deal prob not 0


matrix3<-matrix(c(length(p.blurness.25[p.blurness.25<0.000001]),
                  length(p.blurness.25[p.blurness.25>=0.000001]),
                  length(p.blurness.50[p.blurness.50<0.000001]),
                  length(p.blurness.50[p.blurness.50>=0.000001]),
                  length(p.blurness.75[p.blurness.75<0.000001]),
                  length(p.blurness.75[p.blurness.75>=0.000001]),
                  length(p.blurness.100[p.blurness.100<0.000001]),
                  length(p.blurness.100[p.blurness.100>=0.000001])),
                ncol=2,byrow=T)
T3<-as.data.frame(matrix3);T3
names(T3)<-c("deal prob is 0", "deal prob not 0");T3
rownames(T3)<-c("<-279.03",">279.03 and <-517.91",">517.91 and <-935.09",">935.09");T3

#                         deal prob is 0    deal prob not 0
# <=279.03                         155             74
# >279.03 and <=517.91             155             73
# >517.91 and <=935.09             147             81
# >935.09                          142             87

chisq.test(T3)
#  X-squared = 2.448, df = 3, p-value = 0.4848
# FAIL TO REJECT

#########################
#  dullness

data.dullness<-imgdata9145dullness;data.dullness
summary(data.dullness)
length(data.dullness[data.dullness==0])             #601 dullness=0 #313 dullness not 0

index.0.dull<-which(data.dullness==0);index.0.dull
prob.dull.0<-deal.prob914[index.0.dull];length(prob.dull.0)
prob.dull.non.0<-deal.prob914[-index.0.dull];length(prob.dull.non.0)

length(prob.dull.0[prob.dull.0==0])       #among 601 dullness=0, 394 deal prob =0
length(prob.dull.0[prob.dull.0!=0])                  # 207 deal prob not 0

length(prob.dull.non.0[prob.dull.non.0==0])       #among 313 dullness not 0, 205 deal prob =0
length(prob.dull.non.0[prob.dull.non.0!=0])       # 108 deal prob not 0

matrix4<-matrix(c(length(prob.dull.0[prob.dull.0==0]),
                  length(prob.dull.0[prob.dull.0!=0]),
                  length(prob.dull.non.0[prob.dull.non.0==0]),
                  length(prob.dull.non.0[prob.dull.non.0!=0])),
                ncol=2,byrow=T)
matrix4

T4<-as.data.frame(matrix4);T4
names(T4)<-c("deal prob is 0", "deal prob not 0");T4
rownames(T4)<-c("dullness=0","dullness larger than 0");T4

#                          deal prob is 0     deal prob not 0
#      dullness=0                     394             207
#      dullness larger than 0         205             108
chisq.test(T4)

#X-squared = 2.4722e-30, df = 1, p-value = 1



###################################
#########################################
## image size
data.img.size<-imgdata9145image_size;data.img.size
summary(data.img.size)
quantile(data.img.size,seq(0.1,0.9,0.1))


#   10%    20%    30%    40%    50%    60%    70%    80%    90%
# 20016.6 24590.4  28770.8  32937.0  37025.5  40999.4  44986.2  50796.4  59006.3
index.size.10<-which(data.img.size<=20016.6);length(index.size.10)  #92 img size <=20016.6
img.size.10<-data.img.size[index.size.10]
p.img.size.10<-deal.prob914[index.size.10]

length(p.img.size.10[p.img.size.10<0.000001])            #61 deal prob 0
length(p.img.size.10[p.img.size.10>=0.000001])           #31 deal prob not 0

index.size.20<-which(data.img.size>20016.6 & data.img.size<=24590.4);length(index.size.20)  #91 img size >20016.6 & <=24590.4
img.size.20<-data.img.size[index.size.20]
p.img.size.20<-deal.prob914[index.size.20]

length(p.img.size.20[p.img.size.20<0.000001])            #56 deal prob 0
length(p.img.size.20[p.img.size.20>=0.000001])           #35 deal prob not 0

index.size.30<-which(data.img.size>24590.4 & data.img.size<=28770.8);length(index.size.30)  #91 img size >24590.4 & <=28770.8
img.size.30<-data.img.size[index.size.30]
p.img.size.30<-deal.prob914[index.size.30]

length(p.img.size.30[p.img.size.30<0.000001])            #63 deal prob 0
length(p.img.size.30[p.img.size.30>=0.000001])           #28 deal prob not 0

index.size.40<-which(data.img.size>28770.8 & data.img.size<=32937.0);length(index.size.40)  #93 img size >28770.8 & <=32937.0
img.size.40<-data.img.size[index.size.40]
p.img.size.40<-deal.prob914[index.size.40]

length(p.img.size.40[p.img.size.40<0.000001])            #57 deal prob 0
length(p.img.size.40[p.img.size.40>=0.000001])           #36 deal prob not 0

index.size.50<-which(data.img.size>32937.0 & data.img.size<=37025.5);length(index.size.50)  #90 img
img.size.50<-data.img.size[index.size.50]
p.img.size.50<-deal.prob914[index.size.50]

length(p.img.size.50[p.img.size.50<0.000001])            #64 deal prob 0
length(p.img.size.50[p.img.size.50>=0.000001])           #26 deal prob not 0

index.size.60<-which(data.img.size>37025.5 & data.img.size<=40999.4);length(index.size.60)  #91 img
img.size.60<-data.img.size[index.size.60]
p.img.size.60<-deal.prob914[index.size.60]

length(p.img.size.60[p.img.size.60<0.000001])            #61 deal prob 0
length(p.img.size.60[p.img.size.60>=0.000001])           #30 deal prob not 0

index.size.70<-which(data.img.size>40999.4 & data.img.size<=44986.2);length(index.size.70)  #92 img
img.size.70<-data.img.size[index.size.70]
p.img.size.70<-deal.prob914[index.size.70]

length(p.img.size.70[p.img.size.70<0.000001])            #60 deal prob 0
length(p.img.size.70[p.img.size.70>=0.000001])           #32 deal prob not 0

index.size.80<-which(data.img.size>44986.2 & data.img.size<=50796.4);length(index.size.80)  #91 img
img.size.80<-data.img.size[index.size.80]
p.img.size.80<-deal.prob914[index.size.80]

length(p.img.size.80[p.img.size.80<0.000001])            #59 deal prob 0
length(p.img.size.80[p.img.size.80>=0.000001])           #32 deal prob not 0

index.size.90<-which(data.img.size>50796.4 & data.img.size<=59006.3);length(index.size.90)  #91 img
img.size.90<-data.img.size[index.size.90]
p.img.size.90<-deal.prob914[index.size.90]

length(p.img.size.90[p.img.size.90<0.000001])            #62 deal prob 0
length(p.img.size.90[p.img.size.90>=0.000001])           #29 deal prob not 0

index.size.100<-which(data.img.size>59006.3);length(index.size.100)  #92 img
img.size.100<-data.img.size[index.size.100]
p.img.size.100<-deal.prob914[index.size.100]

length(p.img.size.100[p.img.size.100<0.000001])           #56 deal prob 0
length(p.img.size.100[p.img.size.100>=0.000001])          #36 deal prob not 0

matrix5<-matrix(c(61,31,56,35,63,28,57,36,64,26,61,30,60,32,59,32,62,29,56,36),ncol=2,byrow=T)
matrix5
T5<-as.data.frame(matrix5);T5
names(T5)<-c("deal prob is 0", "deal prob not 0");T5
rownames(T5)<-c("0-0.1 quantile","0.1-0.2 quantile","0.2-0.3 quantile",
                "0.3-0.4 quantile","0.4-0.5 quantile","0.5-0.6 quantile",
                "0.6-0.7 quantile","0.7-0.8 quantile","0.8-0.9 quantile",
                "0.9-1 quantile")

T5
chisq.test(T5)
# X-squared = 4.4713, df = 9, p-value = 0.8777

###############################
#############################
#############################
#   image width

data.img.width<-imgdata9145width
summary(data.img.width)
#Min.   1st Qu.  Median   Mean   3rd Qu.  Max.
#108.0   360.0   360.0   415.8   480.0    640.0

quantile(data.img.width,seq(0.1,0.9,0.1))

table(data.img.width)
#108 154 161 182 215 222 225 242 245 250 255 259 266 267 270 271 274 275 279 281 284 287 288
#1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   96  2   1   1   1   1   1   7

#289 295 298 300 301 303 306 313 315 317 318 319 320 322 323 334 343 350 355 356 357 358 359
#1   1   1   1   1   2   1   1   1   2   3  18   1   1   1   1   3   5   1   6   8

#360 380 386 392 394 400 401 402 409 415 427 430 432 438 441 442 443 444 450 451 454 456 458
#325   1   1   1   2   2   1   2   1   1   2   1   1   1   1   1   1   1   1   2   1   1   1

#470 472 477 480 481 482 483 485 486 493 494 498 506 507 521 527 531 532 533 537 538 540 541
#1   1   1  206   4   8   1   1   1   3   1   1   1   2   1   1   1   1   1   1   2   1  15   6

#542 543 549 557 559 563 569 573 580 589 597 600 602 603 608 623 624 638 640
#4   2   1   3   1   2   1   1   1   1   1   3   1   1   1   1   1   1   6  80

index.img.width.first<-which(data.img.width<=300); length(index.img.width.first)  #129 width <=300
index.img.width.second<-which(data.img.width<=400 & data.img.width>300); length(index.img.width.second)  #392 width >300 and <=400
index.img.width.third<-which(data.img.width<=500 & data.img.width>400); length(index.img.width.third)  #249 width >400 and <=500
index.img.width.fourth<-which(data.img.width>500); length(index.img.width.fourth)  #144 width >500

p.img.width.first<-deal.prob914[index.img.width.first]; length(p.img.width.first)
p.img.width.second<-deal.prob914[index.img.width.second];length(p.img.width.second)
p.img.width.third<-deal.prob914[index.img.width.third]; length(p.img.width.third)
p.img.width.fourth<-deal.prob914[index.img.width.fourth]; length(p.img.width.fourth)


length(p.img.width.first[p.img.width.first==0])    #among 129 width<=300, 91 deal prob =0
length(p.img.width.first[p.img.width.first!=0])    # 38 not 0

length(p.img.width.second[p.img.width.second==0])    #among 392, 293 is zero
length(p.img.width.second[p.img.width.second!=0])    # 99 not 0

length(p.img.width.third[p.img.width.third==0])    #among 249, 137 deal prob =0
length(p.img.width.third[p.img.width.third!=0])    # 112 not 0

length(p.img.width.fourth[p.img.width.fourth==0])    #among 144, 78 deal prob =0
length(p.img.width.fourth[p.img.width.fourth!=0])    # 66 not 0

matrix6<-matrix(c(length(p.img.width.first[p.img.width.first==0]),
                  length(p.img.width.first[p.img.width.first!=0]),
                  length(p.img.width.second[p.img.width.second==0]),
                  length(p.img.width.second[p.img.width.second!=0]),
                  length(p.img.width.third[p.img.width.third==0]),
                  length(p.img.width.third[p.img.width.third!=0]),
                  length(p.img.width.fourth[p.img.width.fourth==0]),
                  length(p.img.width.fourth[p.img.width.fourth!=0])),byrow=T,ncol=2)
matrix6
T6<-as.data.frame(matrix6);T6
names(T6)<-c("deal prob is 0", "deal prob not 0");T6
rownames(T6)<-c("width <=300","width >300 and <=400","width >400 and <=500","width >500");T6

#                         deal prob is 0    deal prob not 0
# width <=300                       91             38
# width >300 and <=400             293             99
# width >400 and <=500             137            112
# width >500                        78             66

chisq.test(T6)
# X-squared = 36.582, df = 3, p-value = 5.64e-08

##########################################
#########################################
#########################################


data.img.height<-imgdata9145height
summary(data.img.height)
table(data.img.height)

data.img.height
# 165 179 184 187 206 213 215 225 250 261 268 296 305 312 313 314 319 320 322 340 349 350 359 360 361 362 366
# 1   1   1   1   1   1   1   1   2   1   1   1   1   2   1   1   1   3   1   1   1   2   7  414   1   1   1

# 371 376 377 379 381 393 396 399 400 403 406 408 413 414 417 420 424 428 435 437 441 447 450 452 454 456 460
# 1   2   1   1   1   1   1   1   2   1   1   2   1   2   1   1   1   1   1   1   1   1   1   2   1   1   1

# 461 479 480
# 1   2   428

index.img.height.first<-which(data.img.height<=400); length(index.img.height.first)  #462 height<=400
index.img.height.second<-which(data.img.height>400); length(index.img.height.second) #425 height>400

p.img.height.first<-deal.prob914[index.img.height.first]; length(p.img.height.first)
p.img.height.second<-deal.prob914[index.img.height.second];length(p.img.height.second)

length(p.img.height.first[p.img.height.first==0])    #among 462 height <=400, 266 deal prob =0
```

```
length(p.img.height.first[p.img.height.first!=0])    # 196 not 0

length(p.img.height.second[p.img.height.second==0])    #among 452 height >400, 333 deal prob =0
length(p.img.height.second[p.img.height.second!=0])    # 119 not 0

matrix7<-matrix(c(length(p.img.height.first[p.img.height.first==0]),
                  length(p.img.height.first[p.img.height.first!=0]),
                  length(p.img.height.second[p.img.height.second==0]),
                  length(p.img.height.second[p.img.height.second!=0])),ncol=2,byrow=T)
matrix7

T7<-as.data.frame(matrix7);T7
names(T7)<-c("deal prob is 0", "deal prob not 0");T7
rownames(T7)<-c("height <=400","height>400");T7
chisq.test(T7)

#                deal prob is 0    deal prob not 0
#  height <=400              266             196
#  height>400                333             119

summary(lm(data.img.height~data.img.width))
##################################################
##################################################
# Certain type of dimension of image

index.width.270<-which(data.img.width==270);length(index.width.270)  #96 pic with width 270
width.270.height<-data.img.height[index.width.270];width.270.height
length(width.270.height[width.270.height==480])      #all 96 pics are 270*480

index.width.360<-which(data.img.width==360);length(index.width.360)  #325 pic with width 360
width.360.height<-data.img.height[index.width.360];width.360.height
summary(width.360.height)
length(width.360.height[width.360.height==480])      # 258 pics are 360*480
index.360.480<-which(width.360.height==480);length(index.360.480)

p.360.480<-deal.prob914[index.360.480];length(p.360.480)    # 258 img dim 360*480
p.270.480<-deal.prob914[index.width.270];length(p.270.480)  # 96 img dim  270*480

length(p.360.480[p.360.480==0])     #among 258 img 360*480, 171 deal prob 0
length(p.360.480[p.360.480!=0])     #87 deal prob not 0

length(p.270.480[p.270.480==0])     #among 96 img 270*480, 67 deal prob 0
length(p.270.480[p.270.480!=0])     #29 deal prob not 0

matrix8<-matrix(c(length(p.360.480[p.360.480==0]),
                  length(p.360.480[p.360.480!=0]),
                  length(p.270.480[p.270.480==0]),
                  length(p.270.480[p.270.480!=0])),ncol=2,byrow=T)
matrix8
T8<-as.data.frame(matrix8);T8
names(T8)<-c("deal prob is 0", "deal prob not 0");T8
rownames(T8)<-c("360*480","270*480");T8
#               deal prob is 0    deal prob not 0
#  360*480             171              87
#  270*480              67              29

chisq.test(T8)
#X-squared = 0.24862, df = 1, p-value = 0.618

################################
# width 480

index.width.480<-which(data.img.width==480);length(index.width.480)  #206 pic with width 480
width.480.height<-data.img.height[index.width.480];length(width.480.height==360) # all 206 pics dim 480*360
p.480.360<-deal.prob914[index.width.480]
length(p.480.360[p.480.360==0])      #among 206 480*360 pics, 112 deal prob 0
length(p.480.360[p.480.360!=0])      # 94 not 0

################################
# width 640
index.width.640<-which(data.img.width==640);length(index.width.640)  #80 pic with width 640
width.640.height<-data.img.height[index.width.640]
table(width.640.height)  #mainly 640*360
# 261 296 359 360
#  1   1   7  71

index.640.360<-which(width.640.height==360);length(index.640.360)

p.640.360<-deal.prob914[index.640.360];length(p.640.360)   #71 img dim 640*360
length(p.640.360[p.640.360==0])                 #54 deal prob 0
length(p.640.360[p.640.360!=0])                  # 17 deal prob not 0

matrix9<-matrix(c(length(p.360.480[p.360.480==0]),
                  length(p.360.480[p.360.480!=0]),
                  length(p.270.480[p.270.480==0]),
                  length(p.270.480[p.270.480!=0]),
                  length(p.480.360[p.480.360==0]),
                  length(p.480.360[p.480.360!=0]),
                  length(p.640.360[p.640.360==0]),
                  length(p.640.360[p.640.360!=0])),ncol=2,byrow=T)
matrix9
T9<-as.data.frame(matrix9);T9

names(T9)<-c("deal prob is 0", "deal prob not 0");T9
rownames(T9)<-c("360*480","270*480","480*360","640*360");T9

#               deal prob is 0    deal prob not 0
#      360*480             171              87
#      270*480              67              29
#      480*360             112              94
#      640*360              54              17

chisq.test(T9)
# X-squared = 14.756, df = 3, p-value = 0.002037


##############################################
##############################################
# whiteness
data.whiteness<-imgdata914$whiteness;data.whiteness
summary(data.whiteness)
length(data.whiteness[data.whiteness==0])            #599 whiteness=0 #315 whiteness not 0

index.0.white<-which(data.whiteness==0);index.0.white
prob.white.0<-deal.prob914[index.0.white];length(prob.white.0)
prob.white.non.0<-deal.prob914[-index.0.white];length(prob.white.non.0)

length(prob.white.0[prob.white.0==0])       #among 599 whiteness=0, 415 deal prob =0
length(prob.white.0[prob.white.0!=0])                        # 184 deal prob not 0

length(prob.white.non.0[prob.white.non.0==0])       #among 315 whiteness not 0, 184 deal prob =0
length(prob.white.non.0[prob.white.non.0!=0])       # 131 deal prob not 0

matrix10<-matrix(c(length(prob.white.0[prob.white.0==0]),
                   length(prob.white.0[prob.white.0!=0]),
                   length(prob.white.non.0[prob.white.non.0==0]),
                   length(prob.white.non.0[prob.white.non.0!=0])),
                 ncol=2,byrow=T)
matrix10

T10<-as.data.frame(matrix10);T10
names(T10)<-c("deal prob is 0", "deal prob not 0");T10
rownames(T10)<-c("whiteness=0","whiteness larger than 0");T10

#                          deal prob is 0     deal prob not 0
#     whiteness=0                     415              184
#     whiteness larger than 0         184              131
chisq.test(T10)
#  X-squared = 10.323, df = 1, p-value = 0.001314

#########################################################################
#########################################################################
# region

table(data.region)

#Altai region (Siberia)                 Bashkortostan(Volga)
#25                                      50
#Belgorod region(Central)               Chelyabinsk region(Ural)
#18                                      57
#Irkutsk region (Siberia)               Kaliningrad region(Northwest)
#30                                      23
#Kemerovo Region(Siberia)               Khanty-Mansiysk Autonomous Okrug(Ural)
#21                                      16
#Krasnodar region(South)                Krasnoyarsk region(Siberia)
#104                                     39
#Nizhny Novgorod Region(Volga)          Novosibirsk region(Siberia)
#40                                      46
#Omsk Region(Siberia)                   Orenburg region(Volga)
#37                                      25
#Perm Region(Volga)                     Rostov region(South)
#36                                      56
#Samara Region(Volga)                   Saratov region(Volga)
#48                                      31
#Stavropol region(North Caucasus)       Sverdlovsk region(Ural)
#34                                      57
#Tatarstan(Volga)                       Tula region(Central)
#45                                      16
#Tyumen region(Ural)                    Udmurtia(Volga)
#23                                      16
#Vladimir region(Central)                 Volgograd region(South)
#16                                        27
#Voronezh region (Central)                Yaroslavl region(Central)
#33                                        23

### Siberia:Altai region\ Irkutsk region\ Kemerovo Region\ Krasnoyarsk region\
#          Novosibirsk region\ Omsk Region\

### Volga: Bashkortostan\ Nizhny Novgorod Region\ Orenburg region\
#          Perm Region\ Samara Region\ Saratov region\ Tatarstan\ Udmurtia

### Central: Belgorod region\ Tula region\ Vladimir region\ Voronezh region\
#           Yaroslavl region

###  Ural : Chelyabinsk region\ Khanty-Mansiysk Autonomous Okrug\ Sverdlovsk region
#          Tyumen region

###  Northwest: Kaliningrad region\

###  South: Krasnodar region\ Rostov region\ Volgograd region

###  North Caucasus: Stavropol region

length(table(data.region))  #28 regions in 6 Federal District

mydata$Federal.District<-rep(NA,1000)

for (i in 1:1000)
{
  if (data.region[i]=="Altai region" | data.region[i]=="Irkutsk region"
      | data.region[i]=="Kemerovo Region" | data.region[i]=="Krasnoyarsk region"
      | data.region[i]=="Novosibirsk region" | data.region[i]=="Omsk Region")
  {
    mydata$Federal.District[i]<-"Siberia"
  }

  else if (data.region[i]=="Bashkortostan" | data.region[i]=="Nizhny Novgorod Region"
           | data.region[i]=="Orenburg region" | data.region[i]=="Perm Region"
           | data.region[i]=="Samara Region" | data.region[i]=="Saratov region"
           | data.region[i]=="Tatarstan" |data.region[i]=="Udmurtia")
  {
    mydata$Federal.District[i]<-"Volga"
  }

  else if (data.region[i]=="Belgorod region" | data.region[i]=="Tula region"
           | data.region[i]=="Vladimir region" | data.region[i]=="Voronezh region"
           | data.region[i]=="Yaroslavl region")
  {
    mydata$Federal.District[i]<-"Central"
  }

  else if (data.region[i]=="Chelyabinsk region" | data.region[i]=="Khanty-Mansiysk Autonomous Okrug"
           | data.region[i]=="Sverdlovsk region" | data.region[i]=="Tyumen region")
  {
    mydata$Federal.District[i]<-"Ural"
  }

  else if (data.region[i]=="Kaliningrad region" )
  {
    mydata$Federal.District[i]<-"Northwest"
  }

  else if (data.region[i]=="Krasnodar region" | data.region[i]=="Rostov region"
           | data.region[i]=="Volgograd region")
  {
    mydata$Federal.District[i]<-"South"
  }

  else if (data.region[i]=="Stavropol region")
  {
    mydata$Federal.District[i]<-"North Caucasus"
  }
}

Fed.6<-mydata$Federal.District

index.Volga<-which(Fed.6=="Volga");length(index.Volga)   #299 in Volga Federal District
index.Ural <-which(Fed.6=="Ural"); length(index.Ural)    #153 in Ulra Federal District
index.Siberia <-which(Fed.6=="Siberia"); length(index.Siberia)    #198 in Siberia Federal District
index.Central <-which(Fed.6=="Central"); length(index.Central)    #106 in Central Federal District
index.South <-which(Fed.6=="South"); length(index.South)    #187 in South Federal District
index.Northwest <-which(Fed.6=="Northwest"); length(index.Northwest)    #23 in Northwest Federal District
index.North.Caucasus <-which(Fed.6=="North Caucasus"); length(index.North.Caucasus)    #34 in North Caucasus Federal District

deal.prob1000<-mydata$deal_probability
```

```
p.Volga<-deal.prob1000[index.Volga]
p.Ural<-deal.prob1000[index.Ural]
p.Siberia<-deal.prob1000[index.Siberia]
p.Central<-deal.prob1000[index.Central]
p.South<-deal.prob1000[index.South]
p.Northwest<-deal.prob1000[index.Northwest]
p.North.Caucasus<-deal.prob1000[index.North.Caucasus]

LV0<-length(p.Volga[p.Volga==0]);LV0   #Volga 299total, 178 deal prob 0
LV1<-length(p.Volga[p.Volga!=0]);LV1   # 121 not 0
LU0<-length(p.Ural[p.Ural==0]);LU0   #Ural 153 total, 97 deal prob 0
LU1<-length(p.Ural[p.Ural!=0]);LU1   # 56 not 0
LS0<-length(p.Siberia[p.Siberia==0]);LS0   #Siberia 198 total, 130 deal prob 0
LS1<-length(p.Siberia[p.Siberia!=0]);LS1   # 68 not 0
LC0<-length(p.Central[p.Central==0]);LC0   #Central 106 total, 72 deal prob 0
LC1<-length(p.Central[p.Central!=0]);LC1   # 34 not 0
LSouth0<-length(p.South[p.South==0]);LSouth0   #South 187 total, 110 deal prob 0
LSouth1<-length(p.South[p.South!=0]);LSouth1   # 77 not 0
LNW0<-length(p.Northwest[p.Northwest==0]);LNW0   #Northwest 23 total, 17 deal prob 0
LNW1<-length(p.Northwest[p.Northwest!=0]);LNW1   # 6 not 0
LNC0<-length(p.North.Caucasus[p.North.Caucasus==0]);LNC0   #North Caucasus 34 total, 23 deal prob 0
LNC1<-length(p.North.Caucasus[p.North.Caucasus!=0]);LNC1   # 11 not 0

matrix11<-matrix(c(LV0,LV1,LU0,LU1,LS0,LS1,LC0,LC1,LSouth0,LSouth1,LNW0,LNW1,
                   LNC0,LNC1),ncol=2,byrow=T)
matrix11
T11<-as.data.frame(matrix11);T11
names(T11)<-c("deal prob is 0", "deal prob not 0"); T11
rownames(T11)<-c("Volga","Ural","Siberia","Central","South","Northwest","North Caucasus")
T11

chisq.test(T11)


table(mydata2$en_parent_category_name)


deal.prob.with.img<-deal.prob1000[-index.no.image];length(deal.prob.with.img)
deal.prob.no.img<-deal.prob1000[index.no.image];length(deal.prob.no.img)

length(deal.prob.no.img[deal.prob.no.img==0])
length(deal.prob.no.img[deal.prob.no.img!=0])

length(deal.prob.with.img[deal.prob.with.img==0])
length(deal.prob.with.img[deal.prob.with.img!=0])

matrix12<-matrix(c(length(deal.prob.no.img[deal.prob.no.img==0]),
                   length(deal.prob.no.img[deal.prob.no.img!=0]),
                   length(deal.prob.with.img[deal.prob.with.img==0]),
                   length(deal.prob.with.img[deal.prob.with.img!=0])),ncol=2,byrow=T)
matrix12
T12<-as.data.frame(matrix12);T12
names(T12)<-c("deal prob not 0", "deal prob is 0");T12
rownames(T12)<-c("no image","with image");T12
chisq.test(T12)

table(mydata2$en_parent_category_name)

#
#Animals   Consumer electronics           For business  For home and cottages
# 31             128                        9                 116
#Hobbies and Recreation         Personal things           The property           The services
#      58                              436                      115                    44
# Transport
#    63


index.animals<-which(mydata2$en_parent_category_name=="Animals");length(index.animals)
index.consumer.elec<-which(mydata2$en_parent_category_name=="Consumer electronics");length(index.consumer.elec)
index.business<-which(mydata2$en_parent_category_name=="For business");length(index.business)
index.home<-which(mydata2$en_parent_category_name=="For home and cottages");length(index.home)
index.hobby<-which(mydata2$en_parent_category_name=="Hobbies and Recreation");length(index.hobby)
index.personal<-which(mydata2$en_parent_category_name=="Personal things");length(index.personal)
index.property<-which(mydata2$en_parent_category_name=="The property");length(index.property)
index.service<-which(mydata2$en_parent_category_name=="The services");length(index.service)
index.transport<-which(mydata2$en_parent_category_name=="Transport");length(index.transport)

p.animal.0<-length(deal.prob1000[which(deal.prob1000[index.animals]==0)]);p.animal.0    #animal 15 deal prob 0
p.animal.1<-length(deal.prob1000[which(deal.prob1000[index.animals]!=0)]);p.animal.1    #animal 16 deal prob not 0

p.consumer.elec.0<-length(deal.prob1000[which(deal.prob1000[index.consumer.elec]==0)]);p.consumer.elec.0  #consumer 68 deal prob 0
p.consumer.elec.1<-length(deal.prob1000[which(deal.prob1000[index.consumer.elec]!=0)]);p.consumer.elec.1  #consumer 60 deal prob 0


p.home.0<-length(deal.prob1000[which(deal.prob1000[index.home]==0)]);p.home.0  #home 66 deal prob 0
p.home.1<-length(deal.prob1000[which(deal.prob1000[index.home]!=0)]);p.home.1  #home 50 deal prob not 1

p.hobby.0<-length(deal.prob1000[which(deal.prob1000[index.hobby]==0)]);p.hobby.0  #hobby 48 deal prob 0
p.hobby.1<-length(deal.prob1000[which(deal.prob1000[index.hobby]!=0)]);p.hobby.1  #hobby 10 deal prob not 1


p.personal.0<-length(deal.prob1000[which(deal.prob1000[index.personal]==0)]);p.personal.0  #personal 365 deal prob 0
p.personal.1<-length(deal.prob1000[which(deal.prob1000[index.personal]!=0)]);p.personal.1  #personal 71 deal prob not 1

p.property.0<-length(deal.prob1000[which(deal.prob1000[index.property]==0)]);p.property.0  #property 26 deal prob 0
p.property.1<-length(deal.prob1000[which(deal.prob1000[index.property]!=0)]);p.property.1  #property 89 deal prob not 1

p.service.0<-length(deal.prob1000[which(deal.prob1000[index.service]==0)]);p.service.0  #service 7 deal prob 0
p.service.1<-length(deal.prob1000[which(deal.prob1000[index.service]!=0)]);p.service.1  #service 37 deal prob not 1

p.transport.0<-length(deal.prob1000[which(deal.prob1000[index.transport]==0)]);p.transport.0  #transport 28 deal prob 0
p.transport.1<-length(deal.prob1000[which(deal.prob1000[index.transport]!=0)]);p.transport.1  #transport 35 deal prob not 1



matrix13<-matrix(c(p.animal.0,p.animal.1,p.consumer.elec.0,p.consumer.elec.1,
                   p.home.0,p.home.1,p.hobby.0,p.hobby.1,p.personal.0,p.personal.1,p.property.0
                   ,p.property.1,p.service.0,p.service.1,p.transport.0,p.transport.1),ncol=2,byrow=T);matrix13

T13<-as.data.frame(matrix13);T13
names(T13)<-c("deal prob is 0", "deal prob not 0"); T13
rownames(T13)<-c("animal","consumer","home","hobby","personal","property","service","transport");T13

#                deal prob is 0    deal prob not 0
#animal                  15              16
#consumer                68              60
#home                    66              50
#hobby                   48              10
#personal               365              71
#property                26              89
#service                  7              37
#transport               28              35

chisq.test(T13)

index.date.0<-rep(NA,14)
index.date.1<-rep(NA,14)

for (i in 1:14)
{
  index=which(mydata2$data.monthday==(i+14))
  index.date.0[i]<-length(which(mydata2$data.prob[index]==0))
  index.date.1[i]<-length(which(mydata2$data.prob[index]!=0))

}
index.date.0
index.date.1
sum(index.date.0)
sum(index.date.1)

matrix14<-matrix(c(index.date.0,index.date.1),byrow=F,ncol=2);matrix14
T14<-as.data.frame(matrix14);T14
names(T14)<-c("deal prob is 0", "deal prob not 0"); T14
rownames(T14)<-c("2017-03-15","2017-03-16","2017-03-17","2017-03-18","2017-03-19","2017-03-20","2017-03-21",
                 "2017-03-22","2017-03-23","2017-03-24","2017-03-25","2017-03-26","2017-03-27","2017-03-28");T14
#                deal prob is 0  deal prob not 0
#2017-03-15              43              23
#2017-03-16              49              33
#2017-03-17              50              18
#2017-03-18              58              27
#2017-03-19              43              25
#2017-03-20              48              27
#2017-03-21              39              17
#2017-03-22              47              30
#2017-03-23              42              27
#2017-03-24              41              23
#2017-03-25              42              26
#2017-03-26              46              28
#2017-03-27              42              33
#2017-03-28              37              36

chisq.test(T14)
# X-squared = 12.434, df = 13, p-value = 0.4924



length(data.IsWeekend)

index.weekend<-which(data.IsWeekend==1);length(index.weekend)
index.not.weekend<-which(data.IsWeekend==0);length(index.not.weekend)

p.weekend<-deal.prob1000[index.weekend]
p.not.weekend<-deal.prob1000[index.not.weekend]

length(p.weekend[p.weekend==0]) #189 weekend deal prob 0
length(p.weekend[p.weekend!=0]) # 106 weekend deal prob not 0

length(p.weekend[p.not.weekend==0]) #438 not weekend deal prob 0
length(p.weekend[p.not.weekend!=0]) #267 not weekend deal prob not 0
matrix15<-matrix(c(length(p.weekend[p.weekend==0]),
                   length(p.weekend[p.weekend!=0]),
                   length(p.weekend[p.not.weekend==0]),
                   length(p.weekend[p.not.weekend!=0])),ncol=2,byrow=T);matrix15
T15<-as.data.frame(matrix15);T15
names(T15)<-c("deal prob is 0", "deal prob not 0"); T15
rownames(T15)<-c("weekend","weekday");T15

#         deal prob is 0      deal prob not 0
# weekend            189              106
# weekday            438              267

chisq.test(T15)


########################################
########################################
##    description word count
data.des.WordCount<-mydata2$data.des.WordCount
quantile(data.des.WordCount,seq(0.2,0.8,0.2))

index.word.count.20<-which(data.des.WordCount<=6);length(index.word.count.20)
index.word.count.40<-which(data.des.WordCount<=12 & data.des.WordCount>6);length(index.word.count.40)
index.word.count.60<-which(data.des.WordCount<=21 & data.des.WordCount>12);length(index.word.count.60)
index.word.count.80<-which(data.des.WordCount<=45 & data.des.WordCount>21);length(index.word.count.80)
index.word.count.100<-which(data.des.WordCount>45);length(index.word.count.100)

p.word.count.20<-deal.prob1000[index.word.count.20]
p.word.count.40<-deal.prob1000[index.word.count.40]
p.word.count.60<-deal.prob1000[index.word.count.60]
p.word.count.80<-deal.prob1000[index.word.count.80]
p.word.count.100<-deal.prob1000[index.word.count.100]

length(p.word.count.20) ##227
PW20.0<-length(p.word.count.20[p.word.count.20==0]);PW20.0 #170
PW20.1<-length(p.word.count.20[p.word.count.20!=0]);PW20.1 #57

length(p.word.count.40) ##195
PW40.0<-length(p.word.count.40[p.word.count.40==0]);PW40.0 #135
PW40.1<-length(p.word.count.40[p.word.count.40!=0]);PW40.1 #60

length(p.word.count.60) ##185
PW60.0<-length(p.word.count.60[p.word.count.60==0]);PW60.0 #104
PW60.1<-length(p.word.count.60[p.word.count.60!=0]);PW60.1 #81


length(p.word.count.80) ##198
PW80.0<-length(p.word.count.80[p.word.count.80==0]);PW80.0 #112
PW80.1<-length(p.word.count.80[p.word.count.80!=0]);PW80.1 #86

length(p.word.count.100) ##195
PW100.0<-length(p.word.count.100[p.word.count.100==0]);PW100.0 #106
PW100.1<-length(p.word.count.100[p.word.count.100!=0]);PW100.1 #89

matrix16<-matrix(c(PW20.0,PW20.1,PW40.0,PW40.1,PW60.0,PW60.1,PW80.0,PW80.1,PW100.0,PW100.1),
                 ncol=2,byrow=T);matrix16
T16<-as.data.frame(matrix16);T16
names(T16)<-c("deal prob is 0", "deal prob not 0"); T16
rownames(T16)<-c("word count <=6","word count >6 & <=12","word count >12 & <=21","word count >21 & <=45","word count >45");T16

#                          deal prob is 0   deal prob not 0
# word count <=6                  170              57
# word count >6 & <=12            135              60
# word count >12 & <=21           104              81
# word count >21 & <=45           112              86
# word count >45                  106              89

chisq.test(T16)
# X-squared = 30.291, df = 4, p-value = 4.27e-06

##########################################################
##########################################################
#########################################################

#tile word count
```

```
data.title.wordCount<-mydata2$data.ti.WordCount
quantile(data.title.wordCount,seq(0.2,0.8,0.2))

index.title.word.count.20<-which(data.title.wordCount<=2);length(index.title.word.count.20)
index.title.word.count.40<-which(data.title.wordCount<=3 & data.title.wordCount>2);length(index.title.word.count.40)
index.title.word.count.60<-which(data.title.wordCount<=4 & data.title.wordCount>3);length(index.title.word.count.60)
index.title.word.count.80<-which(data.title.wordCount<=6 & data.title.wordCount>4);length(index.title.word.count.80)
index.title.word.count.100<-which(data.title.wordCount>6);length(index.title.word.count.100)

p.title.word.count.20<-deal.prob1000[index.title.word.count.20]
p.title.word.count.40<-deal.prob1000[index.title.word.count.40]
p.title.word.count.60<-deal.prob1000[index.title.word.count.60]
p.title.word.count.80<-deal.prob1000[index.title.word.count.80]
p.title.word.count.100<-deal.prob1000[index.title.word.count.100]

length(p.title.word.count.20) ##333
PWT20.0<-length(p.title.word.count.20[p.title.word.count.20==0]);PWT20.0 #239
PWT20.1<-length(p.title.word.count.20[p.title.word.count.20!=0]);PWT20.1 #94

length(p.title.word.count.40) ##199
PWT40.0<-length(p.title.word.count.40[p.title.word.count.40==0]);PWT40.0 #119
PWT40.1<-length(p.title.word.count.40[p.title.word.count.40!=0]);PWT40.1 #80

length(p.title.word.count.60) ##150
PWT60.0<-length(p.title.word.count.60[p.title.word.count.60==0]);PWT60.0 #98
PWT60.1<-length(p.title.word.count.60[p.title.word.count.60!=0]);PWT60.1 #52


length(p.title.word.count.80) ##147
PWT80.0<-length(p.title.word.count.80[p.title.word.count.80==0]);PWT80.0 #93
PWT80.1<-length(p.title.word.count.80[p.title.word.count.80!=0]);PWT80.1 #54

length(p.title.word.count.100) ##171
PWT100.0<-length(p.title.word.count.100[p.title.word.count.100==0]);PWT100.0 #78
PWT100.1<-length(p.title.word.count.100[p.title.word.count.100!=0]);PWT100.1 #93

matrix17<-matrix(c(PWT20.0,PWT20.1,PWT40.0,PWT40.1,PWT60.0,PWT60.1,PWT80.0,PWT80.1,PWT100.0,PWT100.1),
                 ncol=2,byrow=T);matrix17
T17<-as.data.frame(matrix17);T17
names(T17)<-c("deal prob is 0", "deal prob not 0"); T17
rownames(T17)<-c("title word count <=2","title word count >2 & <=3","title word count >3 & <=4","title word count >4 & <=6","title word count >6");T17

#                          deal prob is 0      deal prob not 0
# title word count <=2              239              94
# title word count >2 & <=3         119              80
# title word count >3 & <=4          98              52
# title word count >4 & <=6          93              54
# title word count >6                78              93

chisq.test(T17)

setwd("C:\\Users\\hawli\\Desktop")

library(data.table)

mydata<-fread("translate.csv")
picdata<-fread("picdata_sort_1000.csv")
newdata<-fread("newdata.csv")

data.prob = newdata$data.prob



Boostarp_CI_Method <- function(data1,data2,n){
  u = mean(data1)
  N = length(data1)
  simulation<-rep(NA,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= mean(btsample)
  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}

Boostarp_Zero_Method <- function(data1,data2,n){
  u = length(which(data1 == 0))
  N = length(data1)
  simulation<-rep(NA,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= length(which(btsample == 0))
  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}


###
### price
###
summary(newdata$data.Price.0)
data.price = newdata$data.Price.0[is.na(newdata$data.Price.0)==F]
{
  category = 10
  quant = quantile(data.price,seq(0,1,by = 1/category))
  list = rep(0,category)
  for (i in 1:category){
    cate.prob = data.prob[which(data.price> quant[i] & data.price <= quant[i+1])]
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # not rejecting



###
### weekend date on deal_pro
###

index.weekend = which(newdata$data.IsWeekend == 1)
weekend_pro = data.prob[index.weekend]

Boostarp_CI_Method(weekend_pro,data.prob,500) #not reject

# mean(weekend_pro)
# mean(data.prob)

index.weekday = which(newdata$data.IsWeekend == 0)
weekday_pro = data.prob[index.weekend]

Boostarp_CI_Method(weekday_pro,data.prob[-index.weekend],500) #not reject


###
###  User Type
###

index.type = which(newdata$data.UserType == 0)
type0_prob = data.prob[index.type]
Boostarp_CI_Method(type0_prob,data.prob,500) #not reject

index.type = which(newdata$data.UserType == 1)
type1_prob = data.prob[index.type]
Boostarp_CI_Method(type1_prob,data.prob,500) #not reject

index.type = which(newdata$data.UserType == 2)
type2_prob = data.prob[index.type]
Boostarp_CI_Method(type2_prob,data.prob,500) #not reject


###
###  description
###

### word count
summary(newdata$data.des.WordCount)
wordcount = newdata$data.des.WordCount
{
  category = 4
  quant = quantile(wordcount,seq(0,1,by = 1/category));quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(wordcount <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(wordcount> quant[i] & wordcount <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # rejecting only on the lower 25 %
index.word = (which(wordcount> quant[1] & wordcount <= quant[2]))
mean(data.prob[index.word])
mean(data.prob)   # the lower 25 % (description word count smaller than 7) has a smaller deal prob


### digit count
summary(newdata$data.des.DigitCount)
digitcount = newdata$data.des.DigitCount

index.digit = which(digitcount == 0)
digit0_prob = data.prob[index.digit]
Boostarp_CI_Method(type0_prob,data.prob,500) #not reject

index.digit = which(digitcount > 0 & digitcount<= 1)
digit1_prob = data.prob[index.digit]
Boostarp_CI_Method(digit1_prob,data.prob,500) #not reject

index.digit = which(digitcount >  1)
digit2_prob = data.prob[index.digit]
Boostarp_CI_Method(digit2_prob,data.prob,500) #not reject

### capital count
summary(newdata$data.des.CapsCount)
capscount = newdata$data.des.CapsCount

index.caps = which(capscount == 0)
caps0_prob = data.prob[index.caps]
Boostarp_CI_Method(caps0_prob,data.prob,500) # reject
mean(caps0_prob)
mean(data.prob)    # the description without capslock has a smaller deal prob

index.caps = which(capscount > 0)
caps1_prob = data.prob[index.caps]
Boostarp_CI_Method(caps1_prob,data.prob,500) # reject
mean(caps1_prob)
mean(data.prob)    # the description with capslock has a bigger deal prob


###
###  title
###

### word count
summary(newdata$data.ti.WordCount)
wordcount = newdata$data.ti.WordCount
{
  category = 4
  quant = quantile(wordcount,seq(0,1,by = 1/category));quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(wordcount <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(wordcount> quant[i] & wordcount <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # not rejecting

### digit count
summary(newdata$data.ti.DigitCount)
digitcount = newdata$data.ti.DigitCount

index.digit = which(digitcount == 0)
digit0_prob = data.prob[index.digit]
Boostarp_CI_Method(type0_prob,data.prob,500) #not reject

index.digit = which(digitcount >  0)
digit2_prob = data.prob[index.digit]
Boostarp_CI_Method(digit2_prob,data.prob,500) # reject
mean(digit2_prob)
mean(data.prob)    # the title with digits has a bigger deal prob
```

```
### capital count
summary(newdata$data.ti.CapsCount)
capscount = newdata$data.ti.CapsCount

index.caps = which(capscount == 0)
caps0_prob = data.prob[index.caps]
Boostarp_CI_Method(caps0_prob,data.prob,500) # reject
mean(caps0_prob)
mean(data.prob)    # the title without capslock has a smaller deal prob

index.caps = which(capscount > 0)
caps1_prob = data.prob[index.caps]
Boostarp_CI_Method(caps1_prob,data.prob,500) # reject
mean(caps1_prob)
mean(data.prob)    # the description with capslock has a bigger deal prob


###
### pic.blur on deal_pro
###
summary(picdata$blurrness)
blurrness = picdata$blurrness
{
  category = 5
  quant = quantile(blurrness,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(blurrness <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(blurrness> quant[i] & blurrness <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # not rejecting


###
### dullness
###
summary(picdata$dullness)
dullness = picdata$dullness
{
  category = 2
  quant = quantile(dullness,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(dullness <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(dullness> quant[i] & dullness <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # not rejecting


###
### whiteness
###
summary(picdata$whiteness)
whiteness = picdata$whiteness
{
  category = 2
  quant = quantile(whiteness,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(whiteness <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(whiteness> quant[i] & whiteness <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # not rejecting


###
### size
###

summary(picdata$image_size)
size = picdata$image_size
{
  category = 4
  quant = quantile(size,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(size <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(size> quant[i] & size <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # rejecting on the (size bigger than 26931.5)
mean(data.prob[which(size > quant[4])])
mean(data.prob)    #the bigger pictures(upper 25% bigger than 47860) have a smaller deal prob


###
### width
###

summary(picdata$width)
width = picdata$width
{
  category = 3
  quant = quantile(width,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(width <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(width> quant[i] & width <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  # rejecting on all width


###
### height
###

summary(picdata$height)
height = picdata$height
{
  category = 2
  quant = quantile(height,seq(0,1,by = 1/category),na.rm = T);quant
  list = rep(0,category)
  for (i in 1:category){
    if (i == 1){
      cate.prob = data.prob[which(height <= quant[i+1])]
    }
    else{
      cate.prob = data.prob[which(height> quant[i] & height <= quant[i+1])]
    }
    list[i] = Boostarp_CI_Method(cate.prob,data.prob,500)
  }
  list
}  #  rejecting on all height

###
### temp_size
###

summary(picdata$temp_size)
temp_size=picdata$temp_size

index.size = which(temp_size == "(360, 480)")
size0_prob = data.prob[index.size]
Boostarp_CI_Method(size0_prob,data.prob,500) # not reject

index.size = which(temp_size == "(270, 480)")
size1_prob = data.prob[index.size]
Boostarp_CI_Method(size1_prob,data.prob,500) # not reject

index.size = which(temp_size == "(480, 360)")
size2_prob = data.prob[index.size]
Boostarp_CI_Method(size2_prob,data.prob,500) #  reject
Boostarp_Zero_Method(size2_prob,data.prob,500)
mean(size2_prob)
mean(data.prob)   # higher deal pro mean in this category
length(which(size2_prob != 0))/length(size2_prob)
length(which(data.prob != 0))/length(data.prob) # higher deal pro non zero rate in this category


index.size = which(temp_size == "(640, 360)")
size3_prob = data.prob[index.size]
Boostarp_CI_Method(size3_prob,data.prob,500) # not reject
Boostarp_Zero_Method(size3_prob,data.prob,500)
length(which(size3_prob == 0))/length(size3_prob)

setwd("C:\\Users\\haw1i\\Desktop")

library(data.table)

mydata<-fread("translate.csv")
picdata<-fread("picdata_sort_1000.csv")
newdata<-fread("newdata.csv")

data.prob = newdata$data.prob
data.region = mydata$en_region
data.price = mydata$price

index.throw = which(is.na(data.price)==T | data.price == 0)
data.lprice = log(data.price[-index.throw])

table(mydata$user_id)
table(mydata$user_type)


Boostarp_CI_Method <- function(data1,data2,n){
  u = mean(data1)
  N = length(data1)
  simulation<-rep(0,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= mean(btsample)

  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}

Boostarp_Zero_Method <- function(data1,data2,n){
  u = length(which(data1 == 0))
  N = length(data1)
  simulation<-rep(0,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= length(which(btsample == 0))

  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}

###
### with/ without image
###

index.image = which(is.na(picdata$image)==T)
image0.prob = data.prob[index.image]
image1.prob = data.prob[-index.image]
```

```
Boostarp_CI_Method(image0.prob,data.prob,500)    #not reject
Boostarp_Zero_Method(image0.prob,data.prob,500)  #reject

length(which(image0.prob!=0))/length(image0.prob)  #0.6744186
length(which(data.prob!=0))/length(data.prob)      #0.373

Boostarp_CI_Method(image1.prob,data.prob,500)   #not reject
Boostarp_Zero_Method(image1.prob,data.prob,500)   #reject

length(which(image1.prob!=0))/length(image1.prob)   #0.3446389
length(which(data.prob !=0))/length(data.prob)      #0.373

#cross type
Boostarp_CI_Method(image0.prob,image1.prob,500)   #not reject
Boostarp_Zero_Method(image0.prob,image1.prob,500) #reject

length(which(image0.prob!=0))/length(image0.prob)  #0.6744186
length(which(image1.prob!=0))/length(image1.prob)   #0.3446389


################  log price   ################

###
###   weekday
###

index.day0 = which(newdata5data.IsWeekend == 0 )
index.day1 = which(newdata5data.IsWeekend == 1 )

day0.price = data.price[index.day0]
day1.price = data.price[index.day1]

index.na0 = which(is.na(day0.price)==T | day0.price == 0)
index.na1 = which(is.na(day1.price)==T | day1.price == 0)

day0.lprice = log(day0.price[-index.na0])
day1.lprice = log(day1.price[-index.na1])

Boostarp_CI_Method(day0.lprice,data.lprice,500)    # not reject
Boostarp_CI_Method(day1.lprice,data.lprice,500)    # not reject

#across type
Boostarp_CI_Method(day1.lprice,day0.lprice,500)    # reject

mean(day0.lprice)  #  not weekend mean  8.198072
mean(day1.lprice)  #  weekend mean      7.88289
mean(data.lprice)




###
### description word count
###

summary(newdata5data.des.WordCount)
wordcount = newdata5data.des.WordCount
category = 4
quant = quantile(wordcount,seq(0,1,by = 1/category));quant
# 0%  25%  50%  75% 100%
# 1    2    3    5   12

cate.price0 = data.price[which(wordcount <= quant[2])]
cate.price1 = data.price[which(wordcount> quant[2] & wordcount <= quant[3])]
cate.price2 = data.price[which(wordcount> quant[3] & wordcount <= quant[4])]
cate.price3 = data.price[which(wordcount > quant[4])]

cate.lprice0 = log(cate.price0[-which(is.na(cate.price0)==T | cate.price0 == 0)])
cate.lprice1 = log(cate.price1[-which(is.na(cate.price1)==T | cate.price1 == 0)])
cate.lprice2 = log(cate.price2[-which(is.na(cate.price2)==T | cate.price2 == 0)])
cate.lprice3 = log(cate.price3[-which(is.na(cate.price3)==T | cate.price3 == 0)])

length(cate.lprice0)+length(cate.lprice1)+length(cate.lprice2)+length(cate.lprice3) #932
length(data.lprice) #932

Boostarp_CI_Method(cate.lprice0,data.lprice,500)    #  reject

Boostarp_CI_Method(cate.lprice1,data.lprice,500)    #  reject

Boostarp_CI_Method(cate.lprice2,data.lprice,500)    #  not reject

Boostarp_CI_Method(cate.lprice3,data.lprice,500)    #  reject

mean(cate.lprice0) #6.887436
mean(cate.lprice1) #7.526705
mean(cate.lprice2) #8.2252
mean(cate.lprice3) #9.816935
mean(data.lprice)  #8.104735


###
### title word count
###

summary(newdata5data.ti.WordCount)
wordcount = newdata5data.ti.WordCount
category = 4
quant = quantile(wordcount,seq(0,1,by = 1/category));quant
# 0%  25%  50%  75% 100%
# 1    2    3    5   12

cate.price0 = data.price[which(wordcount <= quant[2])]
cate.price1 = data.price[which(wordcount> quant[2] & wordcount <= quant[3])]
cate.price2 = data.price[which(wordcount> quant[3] & wordcount <= quant[4])]
cate.price3 = data.price[which(wordcount > quant[4])]

cate.lprice0 = log(cate.price0[-which(is.na(cate.price0)==T | cate.price0 == 0)])
cate.lprice1 = log(cate.price1[-which(is.na(cate.price1)==T | cate.price1 == 0)])
cate.lprice2 = log(cate.price2[-which(is.na(cate.price2)==T | cate.price2 == 0)])
cate.lprice3 = log(cate.price3[-which(is.na(cate.price3)==T | cate.price3 == 0)])

length(cate.lprice0)+length(cate.lprice1)+length(cate.lprice2)+length(cate.lprice3) #932
length(data.lprice) #932

Boostarp_CI_Method(cate.lprice0,data.lprice,500)    #  reject

Boostarp_CI_Method(cate.lprice1,data.lprice,500)    #  not reject

Boostarp_CI_Method(cate.lprice2,data.lprice,500)    # reject

Boostarp_CI_Method(cate.lprice3,data.lprice,500)    #  reject

mean(cate.lprice0) #6.944468
mean(cate.lprice1) #8.397511
mean(cate.lprice2) #7.687226
mean(cate.lprice3) #9.960613
mean(data.lprice)  #8.104735



mydata<-fread("translate.csv")
picdata<-fread("picdata_sort_1000.csv")
newdata<-fread("newdata.csv")



data.prob
hist(data.prob)
mean(data.prob)
mean.prob = mean(data.prob)

nonzero.prob = data.prob[which(data.prob!= 0)]
hist(nonzero.prob)

# by MLE
theta = 1/ mean.prob
sim.exp = rexp(1000, rate = theta)
hist(sim.exp)
hist(data.prob)

expquantile<-rep(NA,19)
for (i in 1:19)
{
  expquantile[i]<-qexp(0.05*i,rate=theta)
}
expquantile

obsnumber<-rep(NA,20)
obsnumber[1]<-length(data.prob[data.prob<=expquantile[1]])
for (i in 2:19)
{
  obsnumber[i]<-length(data.prob[data.prob>expquantile[i-1] & data.prob<=expquantile[i]])
}
obsnumber[20]<-length(data.prob[data.prob>expquantile[19]])
obsnumber
chisq.test(obsnumber, p = rep(0.05,20)) # reject

## non zero probs
nonzero.prob
hist(nonzero.prob)
mean(nonzero.prob)

uniquantile<-rep(NA,19)
for (i in 1:19)
{
  uniquantile[i]<-qunif(0.05*i,0,1)
}
uniquantile

obsnumber<-rep(NA,20)
obsnumber[1]<-length(nonzero.prob[nonzero.prob<=uniquantile[1]])
for (i in 2:19)
{
  obsnumber[i]<-length(nonzero.prob[nonzero.prob>uniquantile[i-1] & nonzero.prob<=uniquantile[i]])
}
obsnumber[20]<-length(nonzero.prob[nonzero.prob>uniquantile[19]])
obsnumber

chisq.test(obsnumber, p = rep(0.05,20)) # reject


zim.dis <- function(u1,sd1,u2,sd2,p,n){

  value= rep(0,n)
  for(i in 1:n){
    luck = runif(1,0,1)
    if ( luck > p ){
      value [i] = rnorm(1,u2,sd2)
    }
    else{
      value [i] = rnorm(1,u1,sd1)
    }
    if ( value [i] < 0 ){
      value [i] = 0
    }
    else if (value [i] > 1){
      value [i] = 1
    }

  }
  return(value)
}

hist(zim.dis(0.18,0.1,0.8,0.08,0.65,1000))
hist(nonzero.prob)


### comparison in 0 dealprob, small dealprob and large dealprob
# price1% user type1% whiteness1% image width

Boostarp_Zero_Method <- function(data1,data2,n){
  u = length(which(data1 == 0))
  N = length(data1)
  simulation<-rep(0,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= length(which(btsample == 0))

  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}

Boostarp_Small_Method <- function(data1,data2,n){
  u = length(which(data1 > 0 & data1 < 0.5))
  N = length(data1)
  simulation<-rep(0,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= length(which(btsample > 0 & btsample < 0.5))

  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
```

```
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}

Boostarp_Big_Method <- function(data1,data2,n){
  u = length(which(data1 >=0.5))
  N = length(data1)
  simulation<-rep(0,n)
  for (i in 1:n)
  {
    btsample = sample(data2,N,replace = F)
    simulation[i]= length(which(btsample >= 0.5))

  }
  lower = quantile(simulation,0.05,na.rm = T)
  upper = quantile(simulation,0.95,na.rm = T)
  inside = NA
  if(u >=lower && u <= upper){
    inside = T
  }
  else{
    inside = F
  }
  return(inside)
}



#~~~~ dealprob   zero, samll and big deal probs
# price1% user type1% whiteness1% image width
data.prob
prob.0 = data.prob[which(data.prob==0)]
length(prob.0)          #627

prob.small = data.prob[which(data.prob > 0 & data.prob < 0.5)]
length(prob.small)   #257

prob.big = data.prob[which(data.prob >=0.5)]
length(prob.big)     #116

percentrate <-function(data){
  a=length(data[which(data == 0)])/length(data)
  b=length(data[which(data >0 & data <0.5)])/length(data)
  c=length(data[which(data >=0.5)])/length(data)
  return(c(a,b,c,"zero,small,big = 0.627, 0.257,0.116"))
}

###
### price
###

summary(data.price)

category = 4
quant = quantile(data.price,seq(0,1,by = 1/category));quant
# 0%  25%  50%  75% 100%
# 1    2    3    5   12

prob.price0 = data.prob[which(data.price <= quant[2])]
prob.price1 = data.prob[which(data.price> quant[2] & data.price <= quant[3])]
prob.price2 = data.prob[which(data.price> quant[3] & data.price <= quant[4])]
prob.price3 = data.prob[which(data.price > quant[4])]

percentrate(prob.price0)   #0.6366782 0.2629758 0.1003460
percentrate(prob.price1)   #0.6391753 0.2577320 0.1030928
percentrate(prob.price2)   #0.6053812 0.2511211 0.1434978
percentrate(prob.price3)   #0.6184211 0.2675439 0.1140351

Boostarp_Zero_Method(prob.price0,data.prob,500)  # not reject
Boostarp_Small_Method(prob.price0,data.prob,500) # not reject
Boostarp_Big_Method(prob.price0,data.prob,500)   # not reject

Boostarp_Zero_Method(prob.price1,data.prob,500)  # not reject
Boostarp_Small_Method(prob.price1,data.prob,500) # not reject
Boostarp_Big_Method(prob.price1,data.prob,500)   # not reject

Boostarp_Zero_Method(prob.price2,data.prob,500)  # not reject
Boostarp_Small_Method(prob.price2,data.prob,500) # not reject
Boostarp_Big_Method(prob.price2,data.prob,500)   # not reject

Boostarp_Zero_Method(prob.price3,data.prob,500)  # not reject
Boostarp_Small_Method(prob.price3,data.prob,500) # not reject
Boostarp_Big_Method(prob.price3,data.prob,500)   # not reject


###
### user type
###
index.type = which(newdata$data.UserType == 0)
type0_prob = data.prob[index.type]
Boostarp_Zero_Method(type0_prob,data.prob,500)    #not reject
Boostarp_Small_Method(type0_prob,data.prob,500)   #not reject
Boostarp_Big_Method(type0_prob,data.prob,500)     #not reject


index.type = which(newdata$data.UserType == 1)
type1_prob = data.prob[index.type]
Boostarp_Zero_Method(type1_prob,data.prob,500)  #reject
Boostarp_Small_Method(type1_prob,data.prob,500) #reject
Boostarp_Big_Method(type1_prob,data.prob,500)   #not rejct


index.type = which(newdata$data.UserType == 2)
type2_prob = data.prob[index.type]
Boostarp_Zero_Method(type2_prob,data.prob,500)  # reject
Boostarp_Small_Method(type2_prob,data.prob,500) # reject
Boostarp_Big_Method(type2_prob,data.prob,500)   # reject

percentrate(type2_prob)
percentrate(type1_prob)
percentrate(type0_prob)

###
###  whiteness
###
index.type = which(picdata$whiteness == 0)
white0.prob = data.prob[index.type]
percentrate(white0.prob)
Boostarp_Zero_Method(white0.prob,data.prob,500)  # reject
Boostarp_Small_Method(white0.prob,data.prob,500) # reject
Boostarp_Big_Method(white0.prob,data.prob,500)   # not reject

index.type = which(picdata$whiteness > 0)
white1.prob = data.prob[index.type]
percentrate(white1.prob)
Boostarp_Zero_Method(white1.prob,data.prob,500)  # reject
Boostarp_Small_Method(white1.prob,data.prob,500) # reject
Boostarp_Big_Method(white1.prob,data.prob,500)   # not reject

###
### tempsize
###
summary(picdata$temp_size)
temp_size=picdata$temp_size

index.size = which(temp_size == "(360, 480)")
size0.prob = data.prob[index.size]
percentrate(size0.prob)
Boostarp_Zero_Method(size0.prob,data.prob,500)  # reject
Boostarp_Small_Method(size0.prob,data.prob,500) # reject
Boostarp_Big_Method(size0.prob,data.prob,500)   # not reject

index.size = which(temp_size == "(270, 480)")
size1.prob = data.prob[index.size]
percentrate(size1.prob)
Boostarp_Zero_Method(size1.prob,data.prob,500)  # not reject
Boostarp_Small_Method(size1.prob,data.prob,500) # not reject
Boostarp_Big_Method(size1.prob,data.prob,500)   # not reject

index.size = which(temp_size == "(480, 360)")
size2.prob = data.prob[index.size]
percentrate(size2.prob)
Boostarp_Zero_Method(size2.prob,data.prob,500)  # reject
Boostarp_Small_Method(size2.prob,data.prob,500) # not reject
Boostarp_Big_Method(size2.prob,data.prob,500)   # reject

index.size = which(temp_size == "(640, 360)")
size3.prob = data.prob[index.size]
percentrate(size3.prob)
Boostarp_Zero_Method(size3.prob,data.prob,500)  # reject
Boostarp_Small_Method(size3.prob,data.prob,500) # reject
Boostarp_Big_Method(size3.prob,data.prob,500)   # reject


setwd("C:\\Users\\hawli\\Desktop")
newdata<-fread("newdata.csv")

newdata914<-newdata[-index.no.image,]

mydata$with.image<-rep(1,1000)
for(i in 1:1000)
{
  if (mydata$image[i]=="")
  {
    mydata$with.image[i]<-0
  }
}
length(index.no.image)   #86 of 1000 ad no image
mean(mydata$deal_probability[index.no.image])#mean(mydata$deal_probability[-index.no.image])
length(which(mydata$deal_probability[index.no.image]==0)) #28/86 no img deal prob 0
length(which(mydata$deal_probability[-index.no.image]==0)) #599/914 with img deal prob 0

length(mydata$with.image[mydata$with.image==0])

mydata$with.deal<-rep(0,1000)
for (i in 1:1000)
{
  if(mydata$deal_probability[i]>0)
  {
    mydata$with.deal[i]<-1
  }
}
length(mydata$with.deal[mydata$with.deal==1])

mydata$is.private<-rep(0,1000)
for (i in 1:1000)
{
  if (mydata$user_type[i]=="Private")
  {
    mydata$is.private[i]<-1
  }
}

mydata$is.company<-rep(0,1000)
for (i in 1:1000)
{
  if (mydata$user_type[i]=="Company")
  {
    mydata$is.company[i]<-1
  }
}

mydata$is.360x480<-rep(0,1000)
for (i in 1:914)
{
  if (picdata$temp_size[i]=="(360, 480)")
  {
    mydata$is.360x480[i]<-1
  }
}

mydata$is.270x480<-rep(0,1000)
for (i in 1:914)
{
  if (picdata$temp_size[i]=="(270, 480)")
  {
    mydata$is.270x480[i]<-1
  }
}

mydata$is.480x360<-rep(0,1000)
for (i in 1:914)
{
  if (picdata$temp_size[i]=="(480, 360)")
  {
    mydata$is.480x360[i]<-1
  }
}



glm.1<-glm(mydata$with.deal~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private,family = binomial(link=logit))
summary(glm.1)

# get rid of newdata$data.ti.WordCount   -3.388e-03  3.917e-02  -0.086   0.9311

glm.2<-glm(mydata$with.deal~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +mydata$is.company+mydata$is.private, family = binomial(link=logit))
```

```
summary(glm.2)

# get rid of mydata$is.private            1.205e-01  3.716e-01   0.324   0.7457

glm.3<-glm(mydata$with.deal~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.3)

#get rid of newdata$data.IsWeekend       6.404e-02  1.588e-01   0.403   0.6868

glm.4<-glm(mydata$with.deal~mydata$price+mydata$with.image+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.4)

#get rid of newdata$data.ti.CapsCount    1.698e-01  2.217e-01   0.766   0.4436

glm.5<-glm(mydata$with.deal~mydata$price+mydata$with.image+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.5)

#get rid of newdata$data.des.WordCount   1.910e-03  2.358e-03   0.810   0.4179

glm.6<-glm(mydata$with.deal~mydata$price+mydata$with.image
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.6)

#get rid of newdata$data.des.DigitCount  3.144e-02  2.020e-02   1.556   0.1197

glm.7<-glm(mydata$with.deal~mydata$price+mydata$with.image
           +newdata$data.des.CapsCount+newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.7)

anova(glm.6,glm.7,test="Chisq")

#delete newdata$data.des.CapsCount -4.408e-02  2.452e-02  -1.798   0.0721

glm.8<-glm(mydata$with.deal~mydata$price+mydata$with.image
           +newdata$data.ti.DigitCount
           +mydata$is.company,family = binomial(link=logit))
summary(glm.8)

anova(glm.7,glm.8,test="Chisq")
#STOP HERE
#Coefficients:
#                                Estimate Std. Error  z value    Pr(>|z|)
#  (Intercept)                 2.459e-01    2.866e-01    0.858      0.3909
#  mydata$price                1.198e-07    6.102e-08    1.963      0.0496 *
#  mydata$with.image           -1.192e+00    2.880e-01   -4.137     3.52e-05 ***
#  newdata$data.ti.DigitCount  5.010e-01    8.173e-02    6.130     8.80e-10 ***
#  mydata$is.company           -3.597e-01    1.671e-01   -2.152      0.0314 *



#model2<-step(object = glm.1,trace = 0)
#summary(model2)
#anova(object = model2,test = "Chisq")

df<-data.frame(Price=mydata$price,with.image=mydata$with.image,data.ti.DigitCount=newdata$data.ti.DigitCount,is.company=mydata$is.company)
df
predict(glm.8,df,type="response")
prediction1<-as.vector(predict(glm.8,df,type="response"))
length(prediction1[prediction1>=0.5])

test.1<-rep(0,1000)
for (i in 1:1000)
{
  if (!is.na(prediction1[i]) & prediction1[i]>0.5 & mydata$with.deal[i]==1)
  {
    test.1[i]<-1
  }
}
sum(test.1)
############################################################
#################################################################
############################################################

lm.1<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
         +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.1)

# delete newdata$data.IsWeekend       6.551e-03  1.861e-02   0.352  0.72498
lm.2<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.des.WordCount
         +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.2)

# delete newdata$data.des.DigitCount  1.717e-03  2.773e-03   0.619  0.53587
lm.3<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.des.WordCount
         +newdata$data.des.CapsCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.3)

#delete newdata$data.ti.CapsCount   2.003e-02  2.671e-02   0.750  0.45347
lm.4<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.des.WordCount
         +newdata$data.des.CapsCount+newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.4)

# delete mydata$with.image          -2.983e-02  3.400e-02  -0.877  0.38051
lm.5<-lm(mydata$deal_probability~mydata$price+newdata$data.des.WordCount
         +newdata$data.des.CapsCount+newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.5)

#delete newdata$data.des.CapsCount -3.193e-03  2.838e-03  -1.125 0.260913
lm.6<-lm(mydata$deal_probability~mydata$price+newdata$data.des.WordCount
         +newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.6)

#delete newdata$data.des.WordCount  1.948e-04  2.060e-04   0.946 0.344405
lm.7<-lm(mydata$deal_probability~mydata$price
         +newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.7)

#delete mydata$price               -4.994e-09  4.402e-09  -1.134 0.256949
lm.8<-lm(mydata$deal_probability~
           newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lm.8)

#delete mydata$is.company          0.065799   0.041268   1.594  0.11116

lm.9<-lm(mydata$deal_probability~
           newdata$data.ti.DigitCount
         +newdata$data.ti.WordCount+mydata$is.private)
summary(lm.9)

#delete mydata$is.private          0.028764   0.017823   1.614  0.10688
lm.10<-lm(mydata$deal_probability~
            newdata$data.ti.DigitCount
          +newdata$data.ti.WordCount)
summary(lm.10)


##########################################
# backward adjusted R^2(not recommended, too much....)
############################################################
lmb1.0<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.0) #Adjusted R-squared:  0.007457

lmb1.1<-lm(mydata$deal_probability~mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.1) #Adjusted R-squared:  0.0062

lmb1.2<-lm(mydata$deal_probability~mydata$price+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.2) #Adjusted R-squared:  0.007514

lmb1.3<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.3) #Adjusted R-squared:  0.008399

lmb1.4<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.4) #Adjusted R-squared:  0.007804

lmb1.5<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.5) #Adjusted R-squared:  0.006611

lmb1.6<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.6) #Adjusted R-squared:  0.00812

lmb1.7<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.DigitCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.7) #Adjusted R-squared:  0.007977

lmb1.8<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount
           +newdata$data.ti.WordCount+mydata$is.company+mydata$is.private)
summary(lmb1.8) #Adjusted R-squared:  -0.001685

lmb1.9<-lm(mydata$deal_probability~mydata$price+mydata$with.image+newdata$data.IsWeekend+newdata$data.des.WordCount
           +newdata$data.des.CapsCount+newdata$data.des.DigitCount+newdata$data.ti.CapsCount+newdata$data.ti.DigitCount
           +mydata$is.company+mydata$is.private)
summary(lmb1.9) #Adjusted R-squared:  0.003428




library(data.table)

mydata<-fread("translate.csv")
picdata<-fread("picdata_sort_1000.csv")
newdata<-fread("newdata.csv")

del = which(mydata$deal_probability!=0)
del
tukey.mydata = mydata[del]
tukey.picdata = picdata[del]
tukey.newdata = newdata[del]
data.prob = tukey.newdata$data.prob

small = which(tukey.mydata$deal_probability<0.5)
small
big = which(tukey.mydata$deal_probability>0.5)
big
small.tukey.mydata = tukey.mydata[small]
small.tukey.picdata = tukey.picdata[small]
small.tukey.newdata = tukey.newdata[small]
big.tukey.mydata = tukey.mydata[big]
big.tukey.picdata = tukey.picdata[big]
big.tukey.newdata = tukey.newdata[big]
small.data.prob = small.tukey.newdata$data.prob
big.data.prob = big.tukey.newdata$data.prob
hist(small.data.prob)



###
### funtion of getting groups for numerical variables
###
cut = function(variable, n){
  category = n
  quant = quantile(variable, seq(0,1,by = 1/category))
  cate = numeric(category)
  for (i in 1:category){
    cate[i] = paste(quant[i], "~", quant[i+1])
  }
  return(cate)
}

###
### function of splitting deal probability into groups for numerical variables
###
###
anovacat = function(variable){
  var.fac = as.factor(variable)
  category = length(levels(var.fac))
  cate.prob = list()
  j = 1
  for (i in levels(var.fac)){
    cate.prob[j] = list(data.prob[var.fac == i])
    j = j+1
  }
  names(cate.prob) = levels(var.fac)
  aov.out = aov(values~ind, stack(cate.prob))
```

```
  return(aov.out)
}


###
### function of splitting deal probability into groups for categorical variables
###
###
anovanum = function(variable, n){
  category = n
  quant = quantile(variable, seq(0,1,by = 1/category))
  cate.prob = list()
  for (i in 1:category){
    cate.prob[i] = list(data.prob[which(variable> quant[i] & variable <= quant[i+1])])
  }
  names(cate.prob) = 1:category
  aov.out = aov(values~ind, stack(cate.prob))
  return(aov.out)
}


###
### function of splitting deal probability into groups for numerical variables
### small
###
small.cutcat = function(variable){
  var.fac = as.factor(variable)
  category = length(levels(var.fac))
  cate.prob = list()
  j = 1
  for (i in levels(var.fac)){
    cate.prob[j] = list(small.data.prob[var.fac == i])
    j = j+1
  }
  names(cate.prob) = levels(var.fac)
  aov.out = aov(values~ind, stack(cate.prob))
  return(aov.out)
}


###
### function of splitting deal probability into groups for numerical variables
### big
###
big.cutcat = function(variable){
  var.fac = as.factor(variable)
  category = length(levels(var.fac))
  cate.prob = list()
  j = 1
  for (i in levels(var.fac)){
    cate.prob[j] = list(small.data.prob[var.fac == i])
    j = j+1
  }
  names(cate.prob) = levels(var.fac)
  aov.out = aov(values~ind, stack(cate.prob))
  return(aov.out)
}


###
### function of splitting deal probability into groups for categorical variables
### small
###
small.cutnum = function(variable, n){
  category = n
  quant = quantile(variable, seq(0,1,by = 1/category))
  cate.prob = list()
  for (i in 1:category){
    cate.prob[i] = list(small.data.prob[which(variable> quant[i] & variable <= quant[i+1])])
  }
  names(cate.prob) = 1:category
  aov.out = aov(values~ind, stack(cate.prob))
  return(aov.out)
}


###
### function of splitting deal probability into groups for categorical variables
### big
###
big.cutnum = function(variable, n){
  category = n
  quant = quantile(variable, seq(0,1,by = 1/category))
  cate.prob = list()
  for (i in 1:category){
    cate.prob[i] = list(big.data.prob[which(variable> quant[i] & variable <= quant[i+1])])
  }
  names(cate.prob) = 1:category
  aov.out = aov(values~ind, stack(cate.prob))
  return(aov.out)
}


###
### price
###
# check data
summary(newdata$data.Price.0)
data.price = newdata$data.Price.0[is.na(newdata$data.Price.0)==F]
cut(data.price, 5)  # split into groups
price.aov = anovanum(data.price, 5)
summary(price.aov) #ANOVA
TukeyHSD(price.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(price.aov), las = 2) #plot
# none of these reject

summary(small.tukey.newdata$data.Price.0)
small.data.price = small.tukey.newdata$data.Price.0[is.na(small.tukey.newdata$data.Price.0)==F]
cut(small.data.price, 5)  # split into groups
small.price.aov = anovanum(small.data.price, 5)
summary(small.price.aov) #ANOVA
TukeyHSD(small.price.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(small.price.aov), las = 2) #plot
# none of these reject


###
### month day on deal_pro
###
# check data
data.monthday = newdata$data.monthday[is.na(newdata$data.monthday)==F]
cut(data.monthday, 7)
monthday.aov = anovanum(data.monthday, 7) # convert into ANOVA table
summary(monthday.aov) #ANOVA
TukeyHSD(monthday.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(monthday.aov), las = 2) #plot
# none of these reject


###
### week day on deal_pro
###
# check data
week.day = substr(weekdays(as.Date(mydata$activation_date)),1,3)
data.week.day = week.day[is.na(week.day)==F]
weekday.aov = anovacat(data.week.day) # convert into ANOVA table
summary(weekday.aov) #ANOVA
TukeyHSD(weekday.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(weekday.aov), las = 2) #plot
# none of these reject


###
### weekend date on deal_pro
###
# check data
data.isweekend = newdata$data.IsWeekend[is.na(newdata$data.IsWeekend)==F]
isweekend.aov = anovacat(data.isweekend) # convert into ANOVA table
summary(isweekend.aov) #ANOVA
TukeyHSD(isweekend.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(isweekend.aov), las = 2) #plot
# not reject


###
### user type
###
# check data
data.usertype = newdata$data.UserType[is.na(newdata$data.UserType)==F]
usertype.aov = anovacat(data.usertype) # convert into ANOVA table
summary(usertype.aov) #ANOVA
TukeyHSD(usertype.aov) #Tukey's Honest Significance Differences
plot(TukeyHSD(usertype.aov), las = 2) #plot
# none of these reject
```