~~~
```{r}

#Load packages
library(data.table)
library(dplyr)
library(caret)
library(mice)
library(glmnet)
library(caTools)
library(ROCR)

#Read in the data ; I shorten the name as to be more concise
customer<-fread("customer_table.csv")
orders<- fread("order_table.csv")
product<- fread("product_table.csv")
category<- fread("category_table.csv")

#### to change from scientific notation to the actual number
customer[,customer_id := as.character(customer$customer_id)]
orders[,customer_id := as.character(orders$customer_id)]
orders[,order_id := as.character(orders$order_id)]
orders[,product_id := as.character(orders$product_id)]
product[,product_id := as.character(product$product_id)]

```

```{r}

#1.
#Select customers who made only one purchase before 2016/12/22;
onepurchase<-orders %>%
    filter(order_date<20161222,order_amount>0) %>%
    group_by(customer_id) %>%
    mutate(count=n()) %>%
    filter(count==1)
#The head of the results
head(onepurchase)

#2.
#Select users go dormant for 3 months in onepurchase;
is_buyer<-orders %>%
    filter(order_date>=20161222,order_date<20170222,order_amount>0) %>%
    group_by(customer_id) %>%
~~~

```r
        summarise(sum(order_amount))
dormant<-onepurchase[!(onepurchase$customer_id %in% is_buyer$customer_id),]


#3.Flag users who come back
isback<-orders %>%
        filter(order_date>=20170222,order_amount>0) %>%
        group_by(customer_id) %>%
        summarise(n())

dormant$back<-ifelse(dormant$customer_id %in% isback$customer_id,1,0)

#Fill in features
dormant<-dormant[,c(1,3,4,7),with=F]
sample<-inner_join(dormant,customer,by="customer_id")
index<-duplicated(sample$customer_id)
sample<-sample[!index,]

#4.Run a logistic regression using the rest of customer
#features in the customer table, plus anything related to
#their orders before 2016/12/22, including order amount,
#product they purchased, category they purchased.

#a. You have to remove the character type features
#(unless you know how to deal with them), as logistic
#regression could only run a numeric value

#b. You have to deal with missing values (users whose
#features have 'NA'). For now you could just simply
#remove those customers

#Fill in features
sample_features<-inner_join(sample,customer,by="customer_id")

#handle characters
sample_features$country<-as.numeric(as.factor(sample_features$country))
sample_features$gender<-as.numeric(as.factor(sample_features$gender))
sample_features<-sample_features[,-c(1,11,12,16,117)]

#Delete NAs
data<-sample_features[complete.cases(sample_features),]

#Prepare datasets
```

```
set.seed(1234)
split<-sample.split(data$back, SplitRatio = 0.6)
train<-subset(data, split == TRUE)
test<-subset(data, split == FALSE)

#logistic regression
model<-glm(back~.,family=binomial, train)
summary(model)

#Sanity check,deal with characters,dates
summary(sample)
str(sample)
sample$country<-as.numeric(as.factor(sample$country))
sample$gender<-as.numeric(as.factor(sample$gender))
sample<-sample[,-c(1,11,12),with=F]
cor(sample)

#Skewness check example
qqnorm(sample$user_feature1)

#Deal with missing values,dummy & impute
sample$non_latest_device<-ifelse(is.na(sample$latest_device_class),1,0)
sample$age_c<-cut(sample$age,breaks=10,labels=F,include.lowest = T,right=F)

#out of RAM for imputation
##temp<-mice(sample,method="pmm",maxit=5,seed=88)
##data <- complete(temp,1)

#The NA % in rest features is minor, so it's fine to delete some observations here
data<-sample[,-c(10,13),with=F]
data<-data[complete.cases(data),]

#Scale, delete features with all 0
data<-cbind(scale(data[,-c(3,5:7,9,31,61,91,112),with=F]),data[,c(3,5:7,9,112),with=F])

#Set up datasets
set.seed(88)
split<-createDataPartition(data$back,p=0.6,list=F)
train<-data[split,]
test<-data[-split,]

#lasso
train_x <- as.matrix(train[,-105,with=F])
```

```
train_y <- as.factor(ifelse(train$back==1,'YES', 'NO'))
lasso<-cv.glmnet(train_x,train_y,family="binomial",type.measure = "auc")
plot(lasso)

#Evaluation
test_x<-as.matrix(test[,-105,with=F])

predictTest<-predict(lasso,test_x,s="lambda.min",type="response")
ROCRpred<-prediction(predictTest,test$back)

par(mfrow = c(1,2))

ROCRperf<-performance(ROCRpred, "tpr", "fpr")
plot(ROCRperf, colorize=TRUE, print.cutoffs.at=seq(0,0.2,by=0.05),
text.adj=c(-0.2,1.7),main="ROC curve with threshold labels")

PRperf<-performance(ROCRpred, "prec", "rec")
plot(PRperf, colorize=TRUE, print.cutoffs.at=seq(0,0.2,by=0.05), text.adj=c(-0.2,1.7),main="PR curve with threshold labels")

```