STAT 580: Statistical Computing

Homework Assignment

Spring 2015, Dr. Raymond Wong

Due Date: April 27 2015

I am using the following packages:

```
library(ggplot2)
library(lubridate)
library(xtable)
library(foreach)
library(rCharts)
library(magrittr)
library(tidyr)
library(dplyr)
library(reshape2)
library(gtools)
library(sqldf)
```

I am also using the simple renaming function:

```
renm = function(dsn,colnum=NULL,newname=NULL){
   if(is.null(newname)) newname=colnum; colnum=NULL
   if (is.null(colnum)) colnum = 1:ncol(dsn)
   names(dsn)[colnum] = newname
   return(dsn)
}
```

and my working directory is set to `dmc2015/ian`.

0.1 Reading the Data

I am working from the current feature matrix:

```
featMat = readRDS("~/dmc2015/data/featureMatrix/featMat_v1.1.rds")
trn = featMat$train
cls = featMat$class
```

In case I need to reference the raw data, I will read that too:

```
raw.trn = read.csv("~/dmc2015/data/clean_data/train_simple_name.csv")
raw.cls = read.csv("~/dmc2015/data/clean_data/test_simple_name.csv")
```

0.2 Working with the data

Since we are looking at coupon by batch information, I am going to melt the data using my coupon melt function:

```
   source("./R/stackCoupons.R")
   stack.res = stackCoupons(trn[,1:49],cls[,1:49])

## using the following as id:
## orderID,
## orderTime,
## userID,
## couponsReceived,
## basketValue,
## couponsReceivedDate,
## couponsReceivedTime,
## couponsReceivedDoW,
## couponsReceivedWeekend,
## couponsReceivedFriSat,
## orderTimeDate,
## orderTimeTime,
## orderTimeDoW,
## orderTimeWeekend,
## orderTimeFriSat,
## batchID,
## couponsExpire,
## couponsSent,
## TimeBtwnSentRec,
## TimeBtwnRecExpire,
## TimeBtwnRecOrder,
## TimeBtwnOrderExpire
##
## using the following as measure columns:
## couponID1,
## price1,
## basePrice1,
## reward1,
## premiumProduct1,
## brand1,
## productGroup1,
## categoryIDs1,
## coupon1Used,
## couponID2,
## price2,
## basePrice2,
## reward2,
## premiumProduct2,
## brand2,
## productGroup2,
## categoryIDs2,
## coupon2Used,
## couponID3,
## price3,
## basePrice3,
## reward3,
## premiumProduct3,
## brand3,
## productGroup3,
## categoryIDs3,
## coupon3Used

   bvalues = stack.res$train %>%
      select(couponID,basketValue) %>%
      arrange(basketValue) %$%
      basketValue %>%
      unique %>%
      data.frame %>%
      renm("basketValue")

   bvalues$bValRank = 1:nrow(bvalues)

   stack.res$train = stack.res$train %>% left_join(bvalues,by=’basketValue’)
```

0.3 Basic Summary Stats

We can get the 5-number summary stats quickly using `dplyr` and this function:

```
   sum_stats = function(dsn){
      dsn %>% summarise(
         min_bValXcpn = min(basketValue),
         q05_bValXcpn = quantile(basketValue,.05),
         q25_bValXcpn = quantile(basketValue,.25),
         mean_bValXcpn = mean(basketValue),
         med_bValXcpn = median(basketValue),
         max_bValXcpn = max(basketValue),
         q75_bValXcpn = quantile(basketValue,.75),
         q95_bValXcpn = quantile(basketValue,.95),

         min_bValrankXcpn = min(bValRank),
         mean_bValrankXcpn = mean(bValRank),
         max_bValrankXcpn = max(bValRank),

         minOrderTimeXcpn = min(orderTimeTime),
         meanOrderTimeXcpn = mean(orderTimeTime),
         medOrderTimeXcpn = median(orderTimeTime),
         maxOrderTimeXcpn = max(orderTimeTime),

         minSentRecTimeXcpn = min(TimeBtwnSentRec),
         meanSentRecTimeXcpn = mean(TimeBtwnSentRec),
         medSentRecTimeXcpn = median(TimeBtwnSentRec),
         maxSentRecTimeXcpn = max(TimeBtwnSentRec),

         minRecExpTimeXcpn = min(TimeBtwnRecExpire),
         meanRecExpTimeXcpn = mean(TimeBtwnRecExpire),
         medRecExpTimeXcpn = median(TimeBtwnRecExpire),
         maxRecExpTimeXcpn = max(TimeBtwnRecExpire),

         minRecOrderTimeXcpn = min(TimeBtwnRecOrder),
         meanRecOrderTimeXcpn = mean(TimeBtwnRecOrder),
         medRecOrderTimeXcpn = median(TimeBtwnRecOrder),
         maxRecOrderTimeXcpn = max(TimeBtwnRecOrder),
         minOrderExpTimeXcpn = min(TimeBtwnOrderExpire),
         meanOrderExpTimeXcpn = mean(TimeBtwnOrderExpire),
         medOrderExpTimeXcpn = median(TimeBtwnOrderExpire),
         maxOrderExpTimeXcpn = max(TimeBtwnOrderExpire))
   }
```

I can store stats in statXcoupon

```
statXcoupon = stack.res$train %>%
   group_by(couponID) %>%
   sum_stats
```

And we can get the same statistics for coupons being used and coupons not being used:

```
statXcoupon = stack.res$train %>%
     group_by(couponID,couponUsed) %>%
     sum_stats %>%
     gather(couponID,couponUsed) %>%
     renm(c("couponID","couponUsed","var","value")) %>%
     group_by(couponID,value) %>%
     summarise(varname = paste(var,couponUsed,sep=".used")) %>%
     select(-couponID) %>%
     spread(varname,value) %>%
     left_join(statXcoupon,by="couponID") %>%
     arrange(couponID)
```

We can save these results:

```
trn = stack.res$train %>% left_join(statXcoupon, by="couponID") %>% as.data.frame
trn = trn[,c("orderID",names(statXcoupon))]
saveRDS(trn,file="../features/feature_files/coupon_basket_stats_train.rds")
write.csv(trn,file="../features/feature_files/coupon_basket_stats_train.csv",quote=FALSE,na="",row.na

cls = stack.res$test %>% left_join(statXcoupon, by="couponID") %>% as.data.frame
cls = cls[,c("orderID",names(statXcoupon))]
saveRDS(cls,file="../features/feature_files/coupon_basket_stats_class.rds")
write.csv(cls,file="../features/feature_files/coupon_basket_stats_class.csv",quote=FALSE,na="",row.na
```