UWDS2-Wk7-FraudsAndFeatureSelection-jms206

Jim Stearns, NetID=jms206

Due 3 Mar 2015

(Assignment language is in *italics*)

Identifying Fraudulent Retailers

This problem uses real data from a Globys Customer in Latin America (load MobileCarrierHolidayAnalysis2013.csv) and reflects the retailer performance of their customer acquisition campaign for the period Nov 15th 2013 to Jan 15th 2014. The attributes present in the data are a follows:

- *Dealer – the (randomly ordered) name of the retailer.*
- *Sales – the number of sales by that dealer during the period.*
- *SalesWithFirstActivity – the number of sales by that dealer with subsequent activity on the SIM card.*
- *MeanTimeToFirstActivity – the mean time in days from activation of the SIM to first activity on the SIM, for those with activity.*
- *MeanAcctBalance – the average account balance in $ for new customers from that retailer.*
- *PctFirstCDR – the percentage of sales who have had at least one activity on the SIM.*

It is common for some unscrupulous retailers (the names have been randomized!) to claim fees for adding new customers who are not real – SIM cards are activated, but never used.

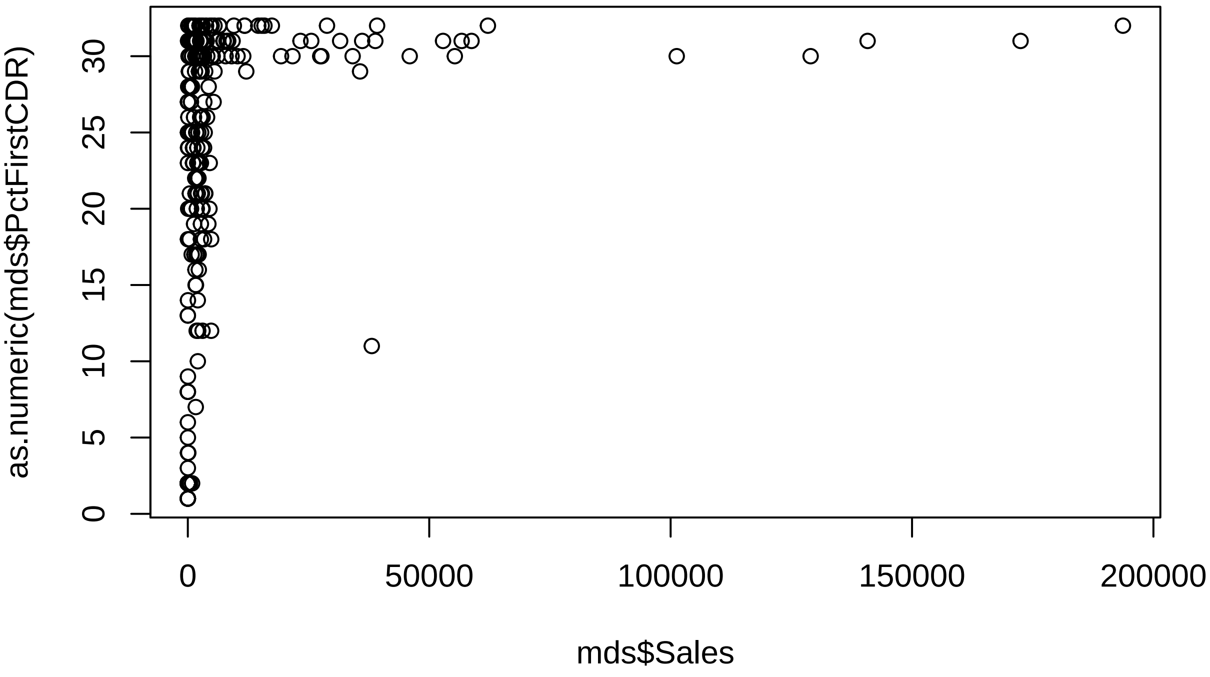
Your task is to explore the data to identify retailers with sales that demonstrate statistically significantly anomalous behavior on 1 or multiple dimensions. Use hypothesis tests and confidence intervals to establish your confidence in retailers that may be fraudulent. State your null and alternative hypotheses clearly and accurately describe conclusion of the tests.

```r
# mds: Mobile Dealer Sales
mds<- read.csv("MobileCarrierHolidayAnalysis2013.csv")
```

Exploration

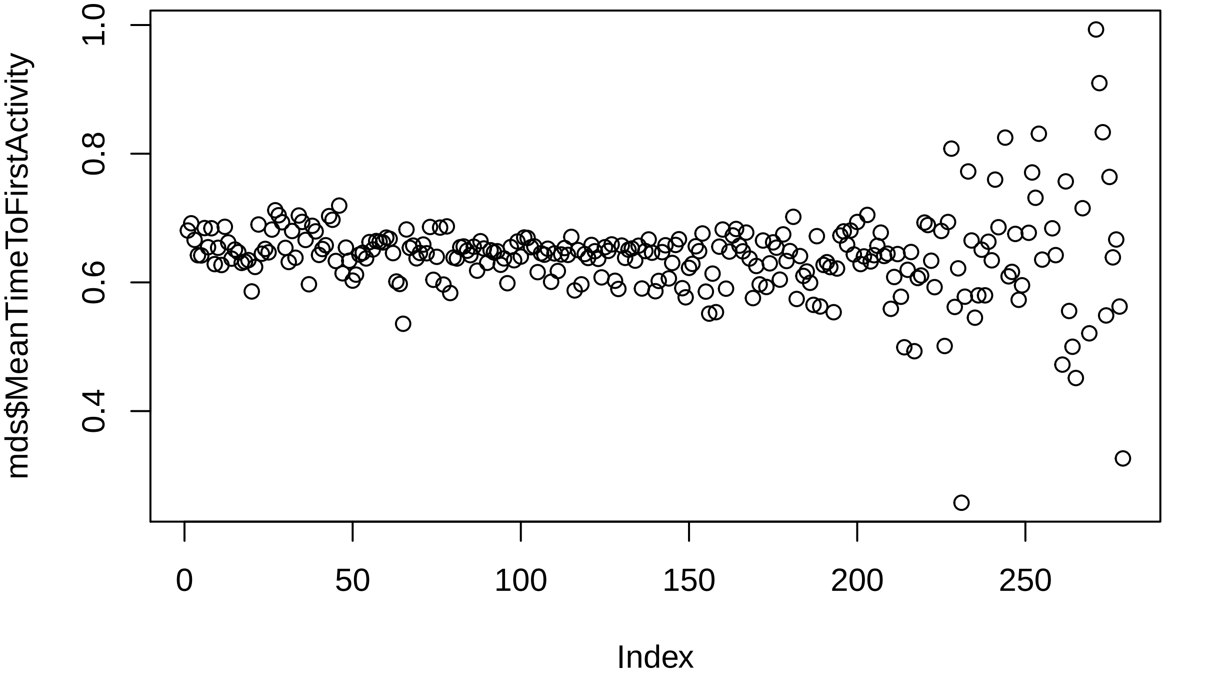
It’s interesting that all of the dealers with the highest sales also have a high percentage of first CDR usage:

```r
plot(mds$Sales, as.numeric(mds$PctFirstCDR))
```



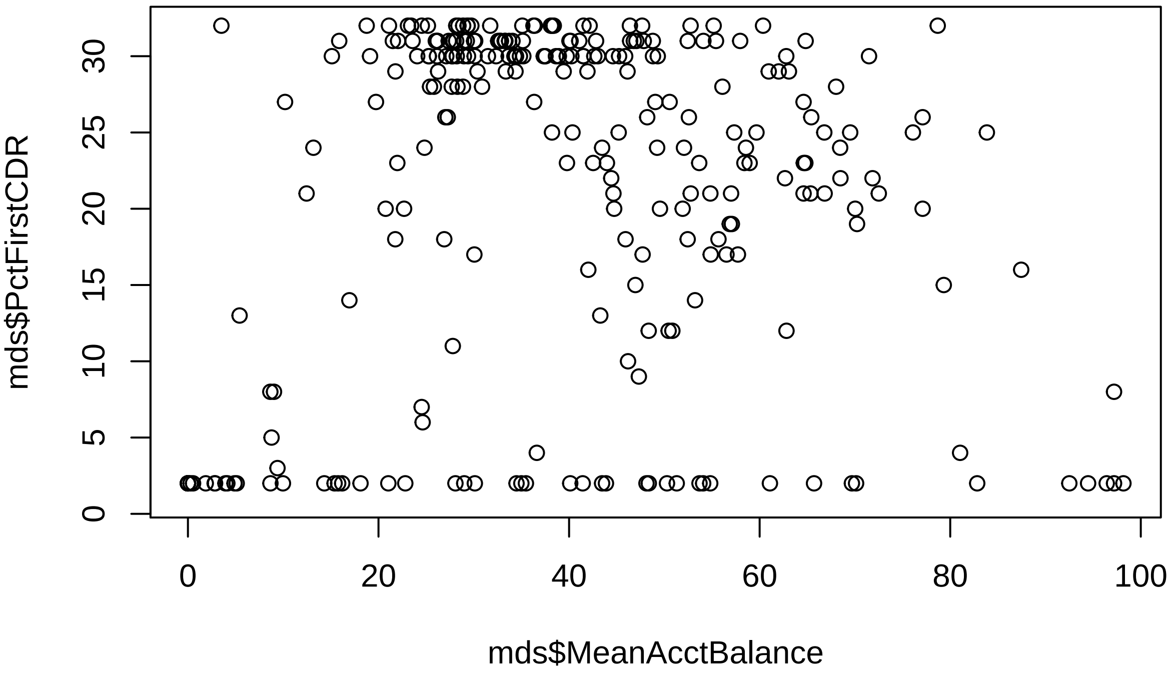
Has the dataset been sorted by some key? Reason for asking: variations in mean to to first activity are clustered around 2/3rds of a day, but the variance starts to increase with Dealer 200 and really spreads out above the 225th dealer:

```
plot(mds$MeanTimeToFirstActivity)
```



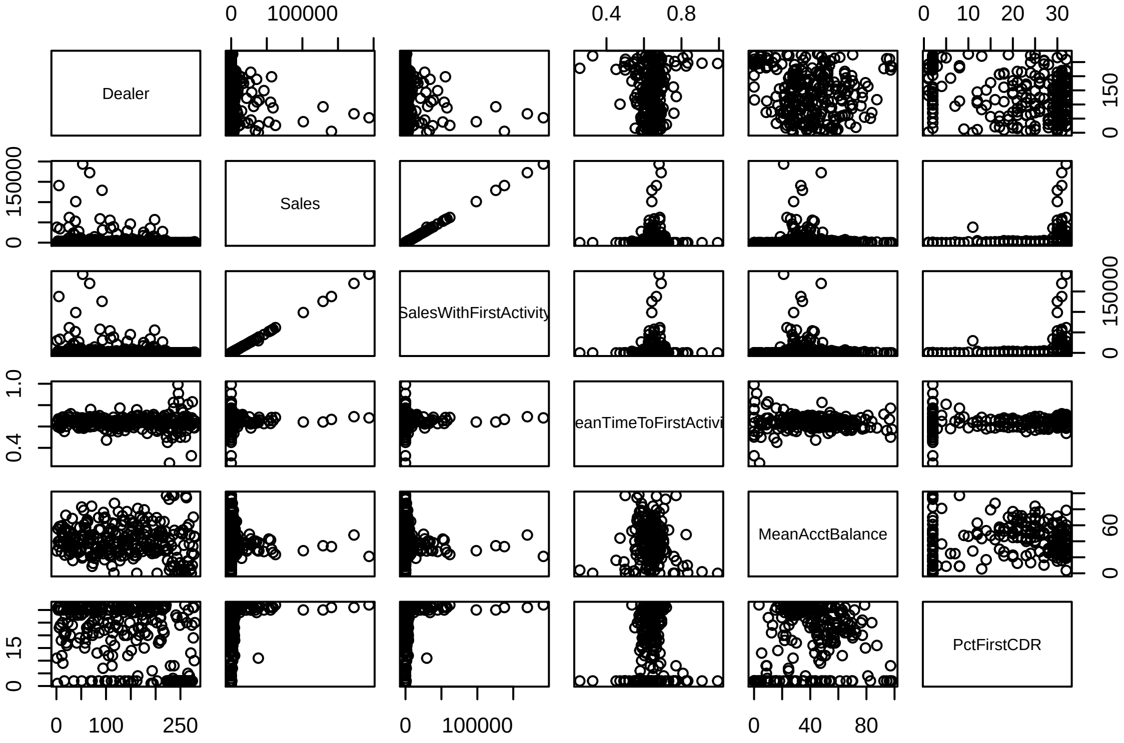
I was expecting the mean account balance for new customers to increase with the percentage of first CDR usage, but it doesn't:

```
plot(mds$MeanAcctBalance, mds$PctFirstCDR)
```



Here’s a 1:1 comparison of all the variables:

```r
plot(mds)
```



# Maximal (or Mutual) Information Coefficient (MIC) and Enron Emails

A subset of the Enron emails (209 of 39,861) have been selected because they are associated with a particular topic. The ID numbers of the emails are contained in doc.list.csv. Use mutual information to identify the words most strongly associated with the emails listed in doc.list.csv (i.e., if each word is a feature, rank them based on mutual information).

Recall that mutual information can be computed by applying mi.plugin() (from the “entropy” package) to a 2x2 contingency table. See Class 7 Examples R file for some help!

```
## Loading required package: entropy
## Loading required package: data.table
```

From Wikipedia MIC:

"The maximal information coefficient uses binning as a means to apply mutual information on continuous random variables. Binning has been used for some time as a way of applying mutual information to continuous distributions; what MIC contributes in addition is a methodology for selecting the number of bins and picking a maximum over many possible grids."

- *Data Files: docword.enron.txt, vocab.enron.txt, doc.list.csv (see (http://archive.ics.uci.edu/ml/datasets/Bag+of+Words) for details of the Enron files)*

```r
datapath = "./"
filename = "docword.enron.txt"
enrondata <- read.table(paste0(datapath, filename), skip=3)
names(enrondata) <- c('docID', 'wordID', 'count')

filename = "vocab.enron.txt"
enronvocab <- read.table(paste0(datapath, filename))
names(enronvocab) = c('wordtext')
nWordsInVocabAllEmail = length(enronvocab$wordtext)

# 209 docIDs of interest.
filename = "doc.list.csv"
enrondoclist <- read.csv(paste0(datapath, filename))
names(enrondoclist) <- c('n', 'docID')
emailSubsetDocIDs <- enrondoclist$docID
nEmailsInSubset = length(emailSubsetDocIDs)

# Sanity checks
stopifnot(nWordsInVocabAllEmail == 28102)
stopifnot(nEmailsInSubset == 209)


# For the 209 emails of interest, get the count of occurrences of words
wcInEmailSubset = enrondata[enrondata$docID %in% emailSubsetDocIDs,]
wcInRestOfEmail = enrondata[! enrondata$docID %in% emailSubsetDocIDs,]
```

Using MIC To Find Words Most Associated with Subset of Email

Compute MIC for each vocabulary word, where the 2x2 contingency table is defined as follows:

_____	Word Present	Word Absent
EmailSubSet (ESS)	nESS_present	nESS_absent
Rest Of Email (RoE)	nRoE_present	nRoE_absent

```r
# Ignore multiple occurrences in a single document (that's the third column).
# Focus on how many different emails each word occurred.
#
# Get word counts in the subset of 209 emails, and in the rest of the emails.
#
# Add zero elements at end (e.g. if subset doesn't have "zycher") by specifying
# the number of bins equal to the number of vocabulary words in all emails.
essWithWord <- data.table(nEmailsWithWord=tabulate(wcInEmailSubset$wordID,
                                                   nbins=nWordsInVocabAllEmail))
roeWithWord <- data.table(nEmailsWithWord=tabulate(wcInRestOfEmail$wordID,
                                                   nbins=nWordsInVocabAllEmail))
nWordsInVocabESS = nrow(essWithWord)
nWordsInVocabRoE = nrow(roeWithWord)
stopifnot(nWordsInVocabESS == nWordsInVocabAllEmail)
stopifnot(nWordsInVocabRoE == nWordsInVocabAllEmail)

calculateMIC <- function(wordIdx, wordCntsESS, wordCntsROE) {
    essT <- wordCntsESS$nEmailsWithWord[wordIdx]
    essF <- nrow(wordCntsESS) - essT
    roeT <- wordCntsROE$nEmailsWithWord[wordIdx]
    roeF <- nrow(wordCntsROE) - roeT
    ctmic <- matrix(c(essT, essF, roeT, roeF), nrow=2, byrow=TRUE)
    if (essT > 0 || roeT > 0) {
        wordIdx
        ctmic
    }
    mi.plugin(ctmic)
}

wordMics = data.table()

for (i in 1:nWordsInVocabAllEmail) {
    mic <- calculateMIC(i, essWithWord, roeWithWord)
    wordtext = as.character(levels(enronvocab$wordtext)[i])
    wordMics = rbind(wordMics, as.list(c(wordtext, mic)), use.names=FALSE)
}
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing value(s)
## in argument freqs2!
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing value(s)
## in argument freqs2!
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing value(s)
## in argument freqs2!
```

```
names(wordMics) = c("word", "mic")
```

```
## Warning in `names<-.data.table`(`*tmp*`, value = c("word", "mic")): The
## names(x)<-value syntax copies the whole table. This is due to <- in R
## itself. Please change to setnames(x,old,new) which does not copy and is
## faster. See help('setnames'). You can safely ignore this warning if it is
## inconvenient to change right now. Setting options(warn=2) turns this
## warning into an error, so you can then use traceback() to find and change
## your names<- calls.
```

```
wordMics$mic = as.numeric(wordMics$mic)
wordMicsSorted <- wordMics[order(wordMics$mic, decreasing=TRUE),]
head(wordMicsSorted, 20)
```

```
##            word        mic
##  1:     meeting 0.09553477
##  2:    attached 0.08639806
##  3:      market 0.08509603
##  4:       think 0.08061907
##  5:       group 0.07928582
##  6:      number 0.07474026
##  7:        date 0.07363933
##  8:    business 0.07327197
##  9:        help 0.07293979
## 10:      energy 0.07177768
## 11:      review 0.06855752
## 12:       going 0.06810367
## 13:        plan 0.06703092
## 14:     forward 0.06671165
## 15:  california 0.06631471
## 16:       power 0.06521491
## 17:     company 0.06449040
## 18:        look 0.06253912
## 19:      office 0.06231592
## 20:    customer 0.06178824
```

The top words aren't very distinctive: "meeting", "attached", "market", etc.

Clearly, I'm not using MIC correctly. So I'll take another tack.

Counting Occurrences in Subset of Email Without Reference to Balance of Emails

```
# Focus on the 209 emails in the subset.
# Ignore multiple occurrences in a single document (that's the third column).
# Focus on how many different emails each word occurred.
# For each word, get the count of docs in the 209 that include the word at least once.
# Add zero elements at end (e.g. if subset doesn't have "zycher") by specifying
# the number of bins equal to the number of vocabulary words in all emails.
docCntsByWord <- data.table(nDocsWithWord=tabulate(wcInEmailSubset$wordID,
                                                   nbins=nWordsInVocabAllEmail))
nWordsInVocabEmailSubset = nrow(docCntsByWord)
stopifnot(nWordsInVocabEmailSubset == nWordsInVocabAllEmail)

# Add the word text as a column in the data table.
docCntsByWord$wordtext = as.character(levels(enronvocab$wordtext))
str(docCntsByWord)
```

```
## Classes 'data.table' and 'data.frame':   28102 obs. of  2 variables:
##  $ nDocsWithWord: int  0 0 0 0 0 0 0 25 0 0 ...
##  $ wordtext     : chr  "aaa" "aaas" "aactive" "aadvantage" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

- *What are the top 10 words that describe the topic common to these emails?*

```
docCntsSorted <-docCntsByWord[order(nDocsWithWord, decreasing=TRUE),]
top10words <- head(docCntsSorted$wordtext, 10)
top10words
```

```
##  [1] "sportslinecom" "report"        "fantasy"       "football"
##  [5] "game"          "against"       "customize"     "season"
##  [9] "team"          "receiving"
```

- *What is the topic that these emails share? Examine some of the top ranked words and give a short description of the topic that ties the 209 emails together.*

Fantasy football league.

A sampling of reinforcing words found in the top 40: player, injury, sunday, games, knee, sunday_game, nfl, league, coach.

Even five non-obvious keywords in the Top 100, all starting with ‘coords’, ended up being related to fantasy football: coords{0,166, 249, 332, 83}. Googling ‘coords166’ yielded a link to CBS SportsLine.com Fantasy Football email. It turns out to be a stripped reference in an HTML entry (‘<AREA coords=“166, 1, 249, 25”>’).

Sportsline.com is owned by CBS, yet “cbs” is not even a vocabulary word. This omission can be explained by historical context. The Enron emails date to 2001, before CBS acquired SportsLine.com in December 2004 (Reference: Wikipedia, search for “CBS Purchase”)