Homework 1

Ethen Liu, Sai Haran, Sophia Hoffman, Annie Didier, Arindam Bhattacharya

1/10/2017

Homework 1

```
library(lubridate)
library(data.table)

# change the working directory to where the dataset lies
setwd('/Users/ethen/Desktop/northwestern/winter/MSIA 421 Data Mining/hw1')
items <- fread('items.csv')
orders <- fread('orders.csv')
```

Question 1

How many unique customers.

```
length( unique(orders$id) )
```

```
## [1] 9856
```

Question 2 - 4

- 2. Suppose that the current date is 01JUL2013, compute the recency (most recent purchase)
- 3. Find the monetary value (total amount spent, item price times quantity) per customer
- 4. Find frequency (number of orders per customer)

Report the mean and standard deviation of recency, frequency, monetary value.

```
orders[ , orddate := dmy(orddate) ]
rfm <- orders[ , .( recency = max(orddate),
                    frequency = length(unique(ordnum)),
                    monetary = sum(price * qty) ), by = id ]
rfm[ , recency := as.numeric( dmy('01JUL2013') - recency ) ]

compute_mean_and_sd <- function(value) {
    # pass in value to compute the mean and standard deviation,
    # the outputed list will contain the passed in variable
    # as an indication of what value is being computed
    value_name <- deparse( substitute(value) )
    mean_name <- paste('mean', value_name, sep = '_')
    sd_name <- paste('sd', value_name, sep = '_')
    info <- list( mean(value), sd(value) )
    names(info) <- c(mean_name, sd_name)
    return(info)
}
```

```
recency  <- rfm[['recency']]
compute_mean_and_sd(recency)
```

```
## $mean_recency
## [1] 630.4676
##
## $sd_recency
## [1] 598.3836
```

```
monetary <- rfm[['monetary']]
compute_mean_and_sd(monetary)
```

```
## $mean_monetary
## [1] 254.3924
##
## $sd_monetary
## [1] 3878.903
```

```
freqency <- rfm[['frequency']]
compute_mean_and_sd(freqency)
```

```
## $mean_freqency
## [1] 5.942269
##
## $sd_freqency
## [1] 7.506697
```

Question 5

Compute the number of unique items that each customer purchased in each category and report the mean and standard deviation for each category.

```
merged <- merge(items, orders, by = 'sku')
category_counts <- merged[ , .( counts = length( unique(name) ) ), by = .(id, category) ]
counts <- category_counts[['counts']]
category_counts[ , compute_mean_and_sd(counts), by = category ]
```

```
##     category mean_counts  sd_counts
##  1:       19    4.689930  7.5948807
##  2:       35    4.622373  7.8118278
##  3:       12    2.147554  2.5431455
##  4:        7    1.911365  2.3970967
##  5:       31    3.757533  6.0899000
##  6:       20    5.370561 10.3418807
##  7:        6    1.392387  1.0006159
##  8:       17    1.353116  1.0924242
##  9:       30    1.208401  0.5398614
## 10:        8    3.086596  5.3570305
## 11:        9    1.147222  0.4814012
## 12:       37    2.139274  2.2780641
## 13:        3    1.664395  1.4285399
## 14:        1    2.026077  1.9416670
## 15:       41    1.719454  1.4020549
## 16:       23    2.187889  2.9925938
## 17:       10    1.452816  0.9739855
```

```
## 18:       44    1.724293  1.5105435
## 19:       21    1.585824  1.3756505
## 20:        5    1.735678  1.5636274
## 21:       27    2.159436  2.7532555
## 22:       40    2.156250  2.2673252
## 23:       14    4.969829  9.5563244
## 24:       36    2.229498  2.7448039
## 25:       38    1.061047  0.2516304
## 26:       26    1.528879  1.2821881
## 27:       99    1.267287  0.8004339
## 28:       22    1.099567  0.3142275
## 29:       50    2.195183  2.5502061
## 30:       39    1.000000  0.0000000
##     category mean_counts  sd_counts
```

Question 6

Compute the entropy and examine if those with higher entropy have more diversity in their reading interest.

```
# compute the entropy and sort them in decreasing order
entropy <- category_counts[ , {
    p <- counts / sum(counts)
    entropy <- -sum( p * log10(p) )
    list(entropy = entropy)
}, by = id ][order(-entropy), ]
```

```
# the top and bottom entropy's customer information
category_counts[ id == 4335961, ]
```

```
##          id category counts
##  1: 4335961        3      5
##  2: 4335961       36      2
##  3: 4335961        6      1
##  4: 4335961       35      2
##  5: 4335961        5      4
##  6: 4335961       19     15
##  7: 4335961       40      3
##  8: 4335961       31      4
##  9: 4335961        1      6
## 10: 4335961       14      9
## 11: 4335961       10      3
## 12: 4335961       41      1
## 13: 4335961       27      2
## 14: 4335961        8      5
## 15: 4335961       17      1
## 16: 4335961       23      7
## 17: 4335961       20      3
## 18: 4335961       12      4
## 19: 4335961       99      3
## 20: 4335961       30      1
## 21: 4335961       37      4
## 22: 4335961       38      1
##          id category counts
```

```
category_counts[ id == 4313828, ]
```

```
##         id category counts
## 1: 4313828       99      1
```

Base on the printed result, the customer that has the highest entropy does in fact have a more diverse taste than the customer that has the lowest entropy.