project4.643

Dieudonne

July 14, 2016

This the fourth mini Project for my course 643 at CUNY

In general ,businesses have customers database where you can find their past purchase's history but usually there is no actual ratings associated to those purchases .How can we build some kind of ratings based on the quantities purchased and generate a recommender system that could be profitable to a company ?Answering this question is the goal in this project .For this particular assignment I use viavi solutions Quaterly Sales dataset .But this can be generalize to many other cases where we can identify customers,items purchased(or service provided) and the amount or quantity purchased

http://www.viavisolutions.com/en-us

```
library(recommenderlab)
library(reshape2)
library(ggplot2)
# Read training file along with header
library(arules)
library(recosystem)
#library(SlopeOne)
library(SVDApproximation)
library(knitr)
library(data.table)
library(RColorBrewer)
library(ggplot2)
df<- read.csv("~/Downloads/QuarterlySalesProject4.csv")
library(psych)
#describe(tr)
head(df)
```

```
##                     Customer         Item Quantity
## 1                        JAS     PathTrak        3
## 2                        JAS       Repair        2
## 3 3 RIVERS COMMUNICATIONS  Accessories        4
## 4 3 RIVERS COMMUNICATIONS   Probe Tips        2
## 5 3 RIVERS COMMUNICATIONS Test Devices        4
## 6      A + COMMUNICATIONS     SDA-5000       10
```

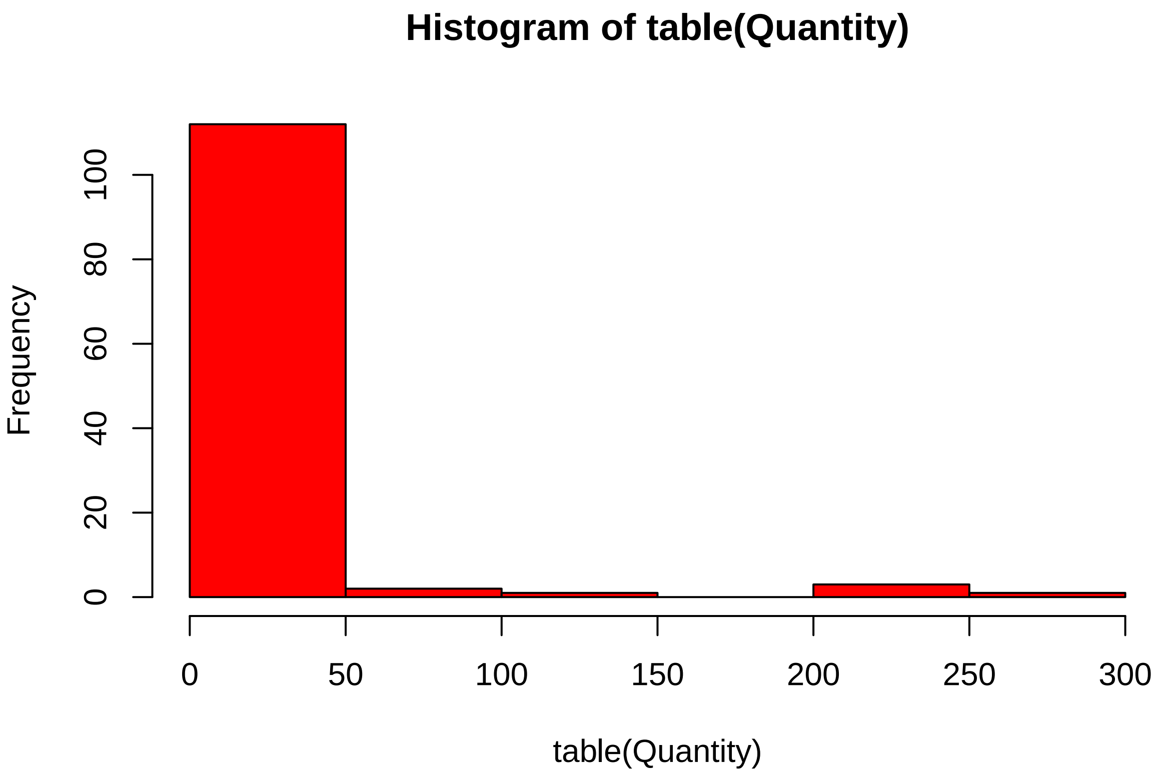
```r
attach(df)
table(Quantity)
```

```
## Quantity
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
## 242 267 225 230  92 111  48  68  38  34  27  47  22  18  13  21  11  16
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   9  13   9   7   3   8   8   5   4   4   7   4   5   5   3   4   4   7
##  37  38  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
##   4   3   5   2   3   1   4   6   2   3   4   2   1   1   2   1   1   1
##  56  57  58  59  60  61  63  64  66  67  69  70  72  75  77  78  81  84
##   1   2   2   1   2   1   1   3   1   1   2   1   1   1   1   1   2   1
##  92  96  98  99 100 108 109 110 115 120 122 123 125 126 135 136 137 141
##   2   1   2   1   1   2   1   1   1   1   1   1   2   1   1   1   1   1
## 146 168 170 176 178 180 183 192 213 216 230 233 234 242 251 271 286 292
##   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1   1   1
## 329 345 349 352 390 403 456 470 496 897 965
##   1   1   1   1   1   1   1   1   2   1   1
```

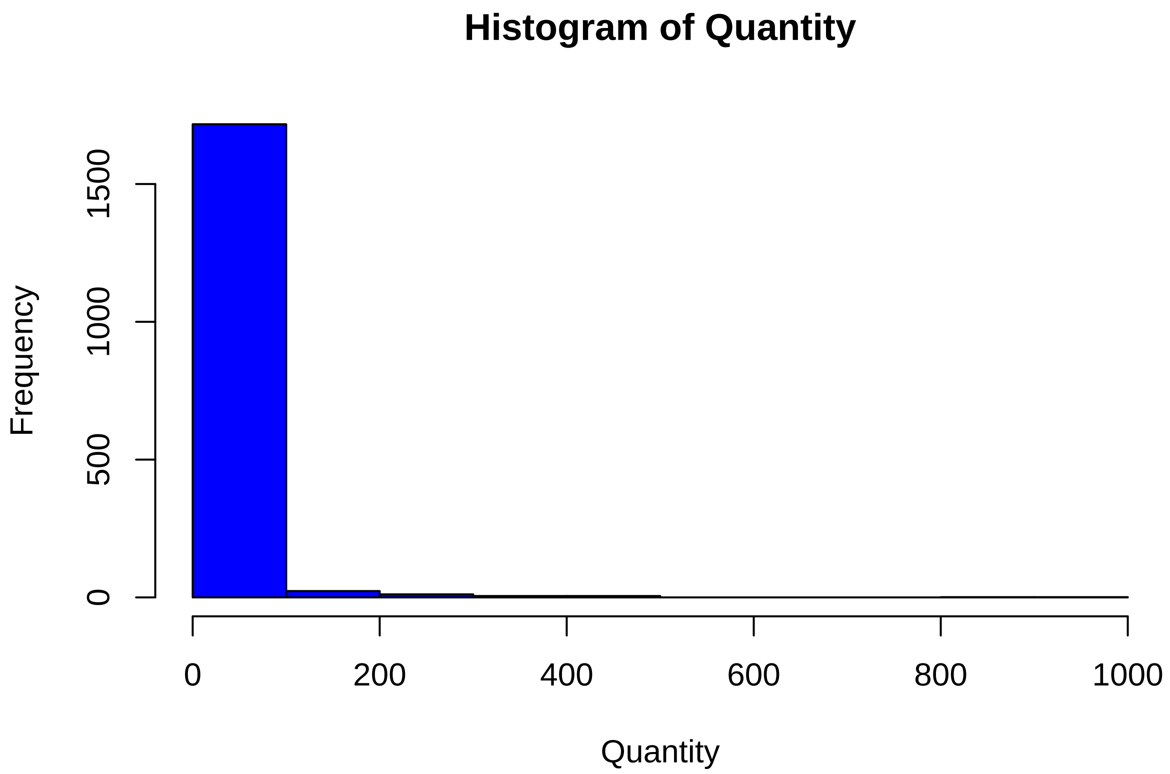
```r
str(Quantity)
```

```
##  int [1:1763] 3 2 4 2 4 10 2 2 2 4 ...
```

```r
hist(table(Quantity),col="red")
```



```r
hist(Quantity,col="blue")
```



```r
names(df)
```

```
## [1] "Customer" "Item"     "Quantity"
```

```r
str(df)
```

```
## 'data.frame':    1763 obs. of  3 variables:
##  $ Customer: Factor w/ 592 levels "3 RIVERS COMMUNICATIONS",..: 288 288 1 1 1 2 3 3 4 4 ...
##  $ Item    : Factor w/ 145 levels "Accessories",..: 81 94 1 84 125 100 16 120 42 43 ...
##  $ Quantity: int  3 2 4 2 4 10 2 2 2 4 ...
```

```r
summary(df)
```

```
##              Customer                            Item
##  COMCAST           :  29   Repair                  : 119
##  TIME WARNER CABLE:  25   Digital Inspection & Test: 100
##  AT&T/NEW HORIZONS:  24   DSAM                     :  98
##  MICROLEASE        :  23   T-Berd/MTS-5800          :  94
##  STOCKING CUSTOMER:  23   Probe Tips               :  87
##  VERIZON WIRELESS :  22   Test Devices             :  87
##  (Other)           :1617   (Other)                  :1178
##     Quantity
##  Min.   :  1.00
##  1st Qu.:  2.00
##   Median :  4.00
##   Mean   : 15.07
##   3rd Qu.: 10.00
##   Max.   :965.00
##
```

```
g<-acast(df, Customer ~ Item)
# Check the class of g
class(g)
```

```
## [1] "matrix"
```

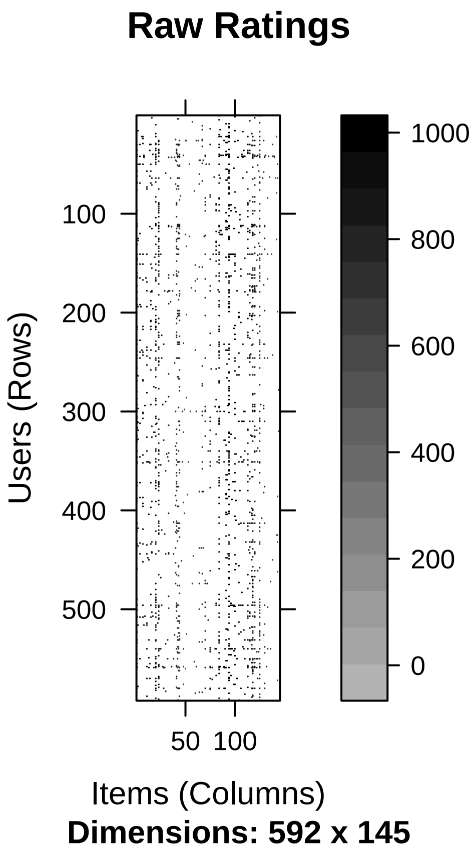
Matrix convertion

```
R<-as.matrix(g)

# Convert R into realRatingMatrix data structure
#   realRatingMatrix is a recommenderlab sparse-matrix like data-structure
r <- as(R, "realRatingMatrix")

# I can turn it into data-frame
kable(head(as(r, "data.frame")))
```

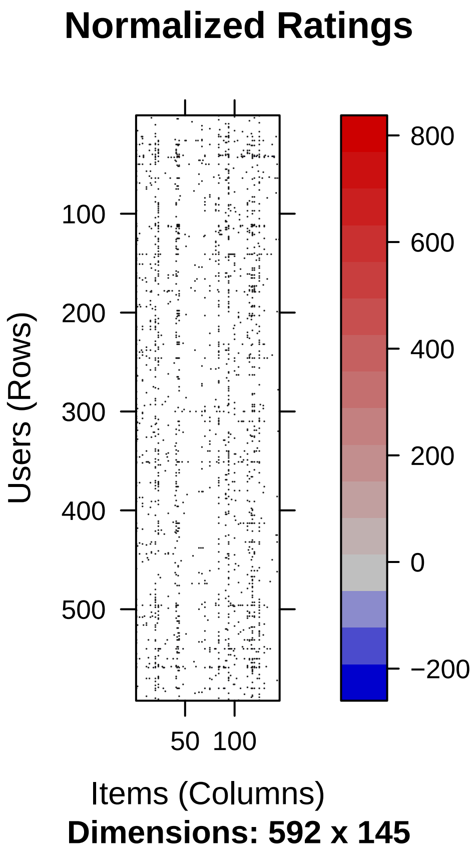
	user	item	rating
1	3 RIVERS COMMUNICATIONS	Accessories	4
878	3 RIVERS COMMUNICATIONS	Probe Tips	2
1610	3 RIVERS COMMUNICATIONS	Test Devices	4
1193	A + COMMUNICATIONS	SDA-5000	10
209	AASKI TECHNOLOGY	Common Product	2
1521	AASKI TECHNOLOGY	TB-6000A Transport Module	2

The ratings matrix need to be normalized

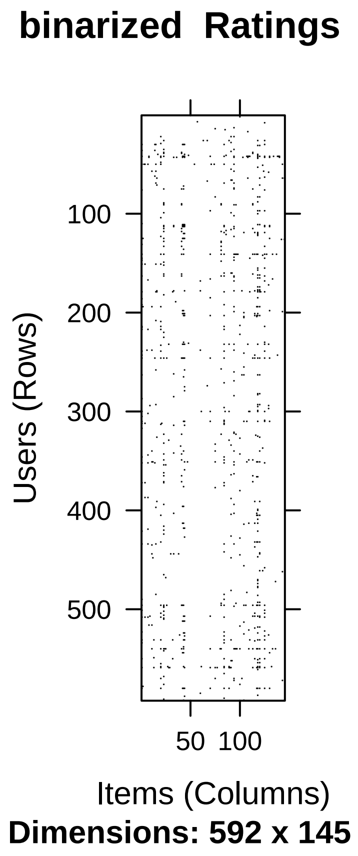
```
r_m <- normalize(r)
#head(r_m)
#head(as(r_m, "list"))

image(r, main = "Raw Ratings")
```



```
image(r_m, main = "Normalized Ratings")
```



```
rb <- binarize(r, minRating=5)
#head(as(rb, "matrix"))
image(rb, main = "binarized  Ratings")
```



Modeling and algorithms and similarity measure

```
#UBCF: User-based collaborative filtering
#
model1=Recommender(r[1:nrow(r)],method="UBCF", param=list(normalize = "Z-score",method="Cosine",nn=5, mi
model2=Recommender(r[1:nrow(r)],method="UBCF", param=list(normalize = "Z-score",method="Jaccard",nn=5, m

#IBCF: Item-based collaborative filtering
#
model3=Recommender(r[1:nrow(r)],method="IBCF", param=list(normalize = "Z-score",method="Jaccard",minRat
# POPULAR
#
model4=Recommender(r[1:nrow(r)],method="POPULAR")


print(model3)
```

```
## Recommender of type 'IBCF' for 'realRatingMatrix'
## learned using 592 users.
```

```
names(getModel(model3))
```

```
##  [1] "description"          "sim"                    "k"
##  [4] "method"               "normalize"            "normalize_sim_matrix"
##  [7] "alpha"                "na_as_zero"           "minRating"
## [10] "verbose"
```

```
getModel(model3)$nn
```

```
## NULL
```

```
print(model1)
```

```
## Recommender of type 'UBCF' for 'realRatingMatrix'
## learned using 592 users.
```

```
names(getModel(model1))
```

```
## [1] "description" "data"        "method"      "nn"          "sample"
## [6] "normalize"   "minRating"   "verbose"
```

```
getModel(model1)$nn
```

```
## [1] 5
```

Predictions and Recommendations to particular customers using differents models

```
# Recommendation to  comcast using model 3
#
Rec.comcast3 <- predict(model3, r["COMCAST",], n=5)
#Top 4  using model4
Rec.comcast4 <- predict(model4, r["COMCAST",], n=10)

Best3comcast <- bestN(Rec.comcast4, n = 3)
Best3comcast
```

```
## Recommendations as 'topNList' with n = 3 for 1 users.
```

```
as(Best3comcast, "list")
```

```
## [[1]]
## [1] "HST-3000C-CE"              "TB-6000A Transport Module"
## [3] "SmartClass HOME"
```

```
#Recommendation to JAS
#
Rec.JAS <- predict(model3, r["JAS",], n=15)
Best5JAS<-bestN(Rec.JAS,n=5)
as(Best5JAS, "list")
```

```
## [[1]]
## [1] "DSAM-6300"                      "Legacy Wireline Services"
## [3] "Location Intelligence Services" "ONX-580"
## [5] "Other - Cable"
```

```
recom <- predict(model3, r[1:nrow(r)], type="ratings")
recom
```

```
## 592 x 145 rating matrix of class 'realRatingMatrix' with 26071 ratings.
```

Models examination

```
head(as(recom, "matrix")[5,3])    # Rating for user 5 for item at index 3
```

```
## [1] NA
```

```
head(as.integer(as(recom, "matrix")[5,3]))# Just get the integer value
```

```
## [1] NA
```

```
head(as.integer(round(as(recom, "matrix")[9,8]))) # Just get the correct integer value
```

```
## [1] 3
```

```
head(as.integer(round(as(recom, "matrix")[368,17])))
```

```
## [1] NA
```

```
# Convert all your recommendations to list structure
rec_list<-as(recom,"list")
head(summary(rec_list))
```

```
##                            Length Class    Mode
## 3 RIVERS COMMUNICATIONS " 96"  "-none-" "numeric"
## A + COMMUNICATIONS      "  0"  "-none-" "numeric"
## AASKI TECHNOLOGY        "  0"  "-none-" "numeric"
## ABB                     " 86"  "-none-" "numeric"
## ACACIA COMMUNICATION    "  0"  "-none-" "numeric"
## ACCELINK TECHNOLOGIES   "  0"  "-none-" "numeric"
```