Final Project

## R Markdown

# install.packages("arules")  
  
library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

data <- read.csv(file = "AppleStoreClean.csv", head = TRUE)  
tdata <- read.transactions("AppleStoreClean.csv", quote="", sep=",")

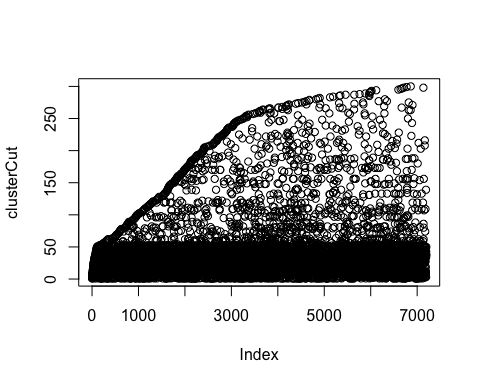
## Warning in asMethod(object): removing duplicated items in transactions

## HCLUST

distance = dist(as.matrix(data), method = "euclidean")

## Warning in dist(as.matrix(data), method = "euclidean"): NAs introduced by  
## coercion

# perform clustering  
hc = hclust(distance)  
  
# cut the tree  
clusterCut <- cutree(hc, 300)  
  
# plot dendrogram  
plot(clusterCut)



#write.csv(clusterCut, file = "C:\\Users\\samue\\Desktop\\hclust.csv")

## Apriori

apprules <- apriori(tdata, parameter = list(support = 0.01, confidence = 0.35, minlen = 2))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.35 0.1 1 none FALSE TRUE 5 0.01 2  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 71   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[18421 item(s), 7198 transaction(s)] done [0.03s].  
## sorting and recoding items ... [67 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 done [0.01s].  
## writing ... [4520 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

summary(apprules)

## set of 4520 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6 7   
## 313 1318 1669 937 262 21   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 3.000 4.000 3.907 5.000 7.000   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.01000 Min. :0.3502 Min. :0.5062 Min. : 72.0   
## 1st Qu.:0.01292 1st Qu.:0.4876 1st Qu.:0.9149 1st Qu.: 93.0   
## Median :0.01848 Median :0.5953 Median :1.0263 Median : 133.0   
## Mean :0.03435 Mean :0.5999 Mean :1.0682 Mean : 247.2   
## 3rd Qu.:0.03418 3rd Qu.:0.6949 3rd Qu.:1.1626 3rd Qu.: 246.0   
## Max. :0.43137 Max. :1.0000 Max. :5.9464 Max. :3105.0   
##   
## mining info:  
## data ntransactions support confidence  
## tdata 7198 0.01 0.35

inspect(sort(apprules, by = "confidence")[1:20])

## lhs rhs support confidence lift   
## [1] {"Shopping"} => {0} 0.01694915 1.0000000 1.468081  
## [2] {"Shopping",37} => {0} 0.01389275 1.0000000 1.468081  
## [3] {"4+","Shopping"} => {0} 0.01139205 1.0000000 1.468081  
## [4] {"9+",6.99} => {"Games"} 0.01500417 1.0000000 1.863801  
## [5] {"Education",3} => {"4+"} 0.01000278 1.0000000 1.623731  
## [6] {"4+","Shopping",37} => {0} 0.01014171 1.0000000 1.468081  
## [7] {"9+",5,6.99} => {"Games"} 0.01472631 1.0000000 1.863801  
## [8] {"9+",0,40,5} => {"Games"} 0.01055849 1.0000000 1.863801  
## [9] {"9+",40,5} => {"Games"} 0.01986663 0.9930556 1.850858  
## [10] {"Education",3.5} => {"4+"} 0.01403168 0.9901961 1.607812  
## [11] {"9+",1,40,5} => {"Games"} 0.01375382 0.9900000 1.845163  
## [12] {"Education",3.5,5} => {"4+"} 0.01236455 0.9888889 1.605690  
## [13] {"Education",40} => {"4+"} 0.01208669 0.9886364 1.605280  
## [14] {"Education",4} => {"4+"} 0.02389553 0.9885057 1.605068  
## [15] {"9+",43,5} => {"Games"} 0.01097527 0.9875000 1.840504  
## [16] {"9+",0,38,4.5,5} => {"Games"} 0.01097527 0.9875000 1.840504  
## [17] {"9+",24} => {5} 0.01055849 0.9870130 1.421472  
## [18] {"9+","Games",24} => {5} 0.01028063 0.9866667 1.420974  
## [19] {"Education",2.99} => {"4+"} 0.02042234 0.9865772 1.601936  
## [20] {"Education",40,5} => {"4+"} 0.01014171 0.9864865 1.601789  
## count  
## [1] 122   
## [2] 100   
## [3] 82   
## [4] 108   
## [5] 72   
## [6] 73   
## [7] 106   
## [8] 76   
## [9] 143   
## [10] 101   
## [11] 99   
## [12] 89   
## [13] 87   
## [14] 172   
## [15] 79   
## [16] 79   
## [17] 76   
## [18] 74   
## [19] 147   
## [20] 73

#itemFrequencyPlot(tdata, support = 0.1) # items with a support of 0.1  
#itemFrequencyPlot(tdata, topN = 20) # 20 most frequent items

## K-Means

## Cut the tree

#clusterCut <- cutree(hc, 10)  
#library(cluster)  
#clusplot(clusterCut, color=TRUE, shade=TRUE, labels=2, lines=0)