CSCI946 Assignment

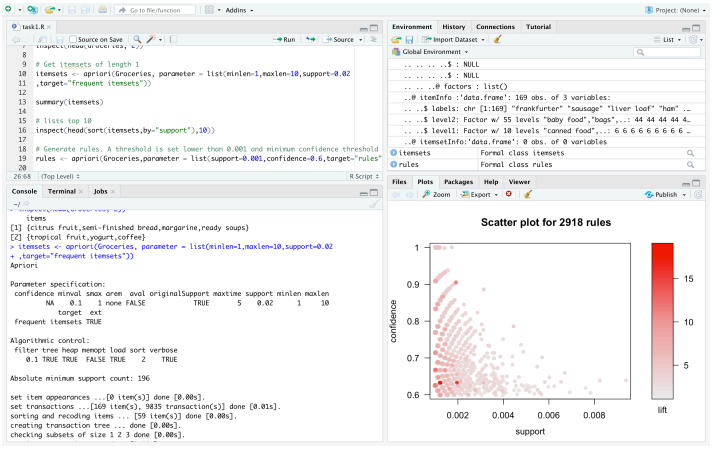
Yao Xiao
SID 2019180015

October 14, 2020

1 Problem 1

```
library(arules)
library(arulesViz)

data("Groceries")

class(Groceries)
inspect(head(Groceries, 2))

# Get itemsets of length 1
itemsets <- apriori(Groceries, parameter = list(minlen=1,maxlen=10,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

itemsets <- apriori(Groceries, parameter = list(minlen=2,maxlen=10,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

itemsets <- apriori(Groceries, parameter = list(minlen=4,maxlen=4,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

# lists top 10
inspect(head(sort(itemsets,by="support"),10))
```

Figure 1: Outputs of the Apriori algorithm



2 Problem 2

```r
library(arules)
library(arulesViz)

data("Groceries")

class(Groceries)
inspect(head(Groceries, 2))

# Get itemsets of length 1
itemsets <- apriori(Groceries, parameter = list(minlen=1,maxlen=10,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

itemsets <- apriori(Groceries, parameter = list(minlen=2,maxlen=10,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

itemsets <- apriori(Groceries, parameter = list(minlen=4,maxlen=4,
    support=0.02
,target="frequent itemsets"))

summary(itemsets)

# lists top 10
inspect(head(sort(itemsets,by="support"),10))

```

```r
# generate rules. A threshold is set lower than 0.001 and minimum confidence threshold is set to 0.6.
rules <- apriori(Groceries, parameter = list(support=0.001, confidence=0.6, target="rules"))

summary(rules)

plot(rules)

# compute the 1/Support(Y) is slope
slope <- sort(round(rules@quality$lift/rules@quality$confidence,2))

# display the number of times each slope appears
unlist(lapply(split(slope, f=slope), length))

inspect(head(sort(rules, by="lift"),10))

confidentRules<-rules[quality(rules)$confidence>0.9]
confidentRules

# plot a matrix-based visualization of the LHS v RHS of rules.
plot(confidentRules, method="matrix", measure=c("lift","confidence"), control=list(reorder="none"))

# visualize the top 5 rules with the highest lift
highLiftRules<-head(sort(rules, by="lift"),5)
plot(highLiftRules, method="graph", control=list(type="items"))
```

Figure 2: Scatter plot of the generated rules

Figure 3: Scatter matrix

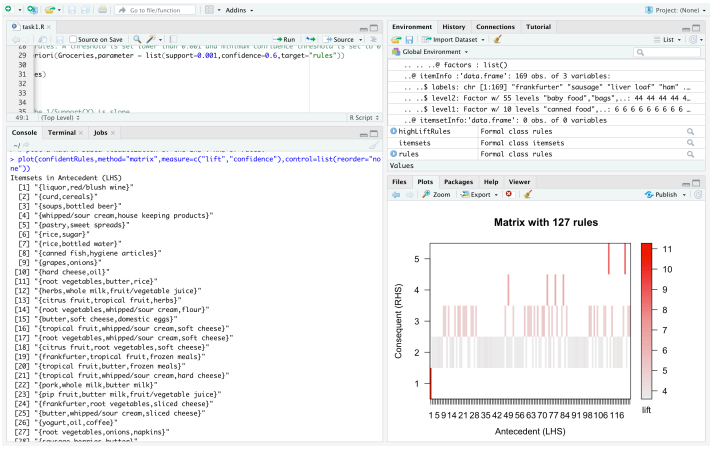


Figure 4: Matrix-based visualization of the LHS and RHS

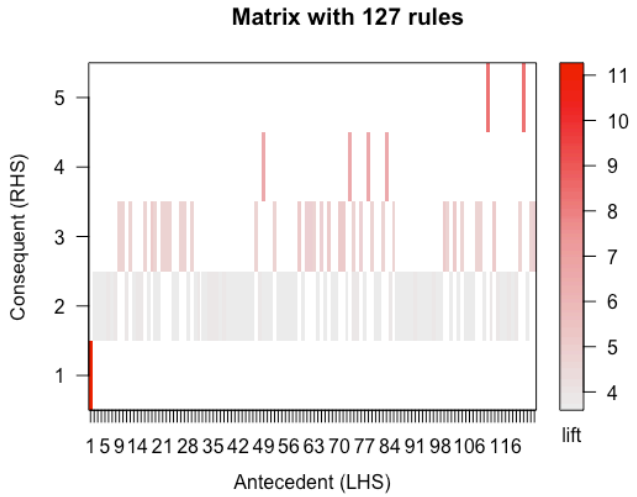


Figure 5: Graph visualization of the top five rules