Unsupervised Learning I: Association Rules - Answers

Instructions

Follow the steps below to construct association rules with the arules package and the colleges data set.

1. Read in the data using the `read.delim` function. Then use `ggplot2`, `ggpairs`, and `dplyr` to identify interesting relationships in the data. Write a short description of one interesting pattern you identified.

This is entirely exploratory so investigate any relationships you might be interested in. Below we look at the federal loan rate vs top ten school status among universities that grant graduate degrees.

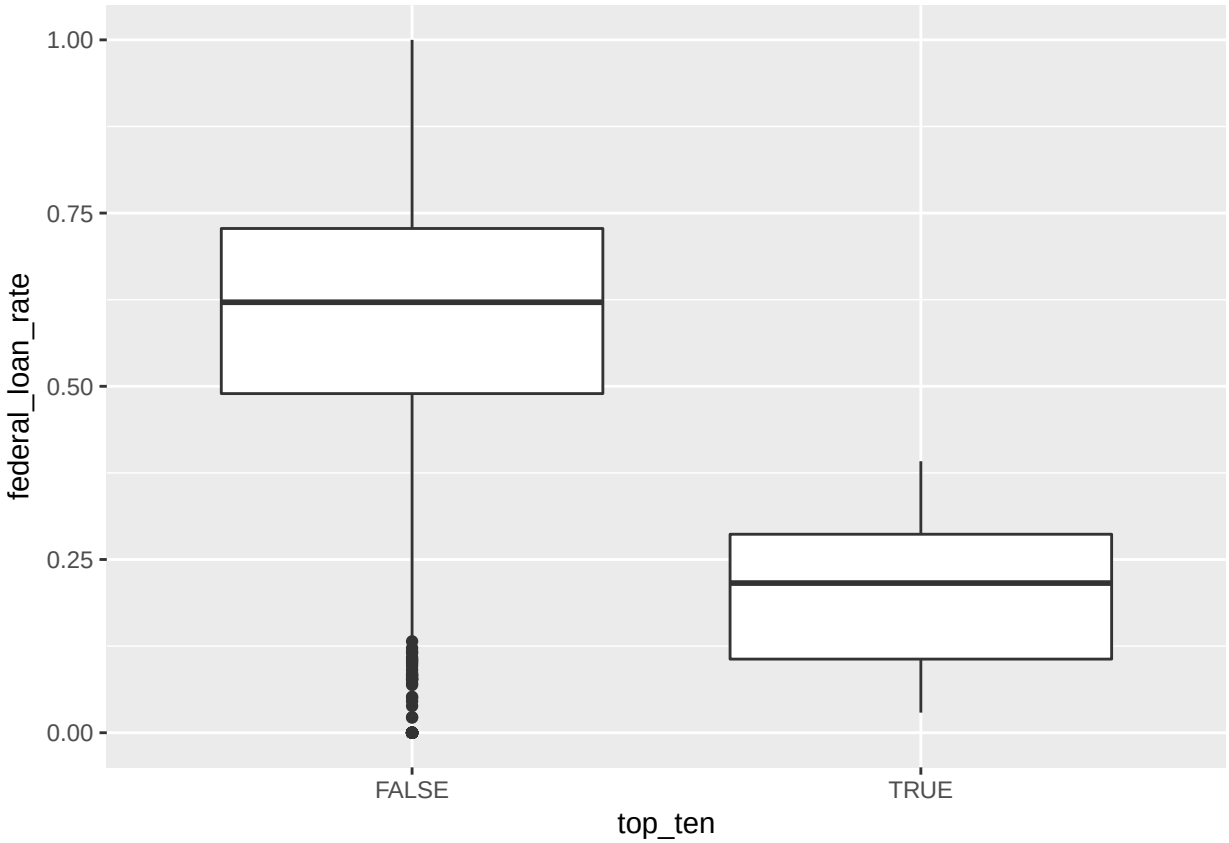
```
library(dplyr)
library(ggplot2)
library(arules)

### Load the colleges datasets on your machine
### colleges = read.delim("colleges.tsv", sep = '\t', header = TRUE)
```

```
graduate_universities = colleges %>%
  filter(highest_degree == "Graduate degree" )

ggplot(graduate_universities, aes(x = top_ten, y = federal_loan_rate)) +
  geom_boxplot()
```

```
## Warning: Removed 283 rows containing non-finite values (stat_boxplot).
```



2. Prepare your data for association rule mining by transforming it into a set of transactions. Use the inspect and summary functions to view the transactions.

```
colleges$cost_quartiles = discretize(colleges$cost,
                                     method = "frequency", categories = 4,
                                     labels = c("cost_Q1", "cost_Q2", "cost_Q3", "cost_Q4"))

colleges$earnings_quartiles = discretize(colleges$median_earnings,
                                         method = "frequency", categories = 4,
                                         labels = c("earnings_Q1", "earnings_Q2", "earnings_Q3", "earni

colleges$debt_quartiles = discretize(colleges$median_debt,
                                     method = "frequency", categories = 4,
                                     labels = c("debt_Q1","debt_Q2", "debt_Q3", "debt_Q4"))

colleges = colleges %>%
  mutate(stem_perc = architecture_major_perc + comm_tech_major_perc +
           computer_science_major_perc + engineering_major_perc + eng_tech_major_perc +
           bio_science_major_perc + math_stats_major_perc,
         high_stem = ifelse(stem_perc >= 0.3, TRUE, FALSE))

college_features = colleges %>%
  select(locale, control, pred_deg, historically_black, men_only,
         women_only, religious, online_only, earnings_quartiles,
         debt_quartiles, cost_quartiles, high_stem, top_ten)

college_trans = as(college_features, "transactions")

inspect(college_trans[1:3])
```

```
##     items                                                transactionID
## [1] {locale=City: Midsize,
##      control=Public,
##      pred_deg=Predominantly bachelor's-degree granting,
##      historically_black,
##      earnings_quartiles=earnings_Q2,
##      debt_quartiles=debt_Q4,
##      cost_quartiles=cost_Q2,
##      high_stem}                                                  1
## [2] {locale=City: Midsize,
##      control=Public,
##      pred_deg=Predominantly bachelor's-degree granting,
##      earnings_quartiles=earnings_Q4,
##      debt_quartiles=debt_Q3,
##      cost_quartiles=cost_Q2}                                     2
## [3] {locale=City: Midsize,
##      control=Private nonprofit,
##      pred_deg=Predominantly bachelor's-degree granting,
##      religious,
##      earnings_quartiles=earnings_Q3,
##      cost_quartiles=cost_Q1}                                     3
```

```
summary(college_trans)
```

```
## transactions as itemMatrix in sparse format with
```

```
##  7308 rows (elements/itemsets/transactions) and
##  39 columns (items) and a density of 0.1426607
##
## most frequent items:
##                          control=Private for-profit
##                                                3365
## pred_deg=Predominantly certificate-degree granting
##                                                3025
##  pred_deg=Predominantly bachelor's-degree granting
##                                                2078
##                                       control=Public
##                                                2044
##                           control=Private nonprofit
##                                                1899
##                                             (Other)
##                                               28249
##
## element (itemset/transaction) length distribution:
## sizes
##    2    3    4    5    6    7    8    9
##   13  347  800 1574 3567  938   66    3
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   6.000   5.564   6.000   9.000
##
## includes extended item information - examples:
##                 labels variables        levels
## 1   locale=City: Large    locale   City: Large
## 2 locale=City: Midsize    locale City: Midsize
## 3   locale=City: Small    locale   City: Small
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

3. Generate rules with the apriori function with a support of 0.01 and a confidence of 0.60.

```
rules = apriori(college_trans, parameter = list(sup = 0.01, conf = 0.6, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.01      1
##  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: 73
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[39 item(s), 7308 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [889 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

4. Try the following combinations of support and confidence: [0.10, 0.60], [0.01, 0.10]. What happens to the number of rules as the support increases? (**Hint:** use the summary function to see the number of rules).

A support of 0.1 and confidence of 0.6 is very restrictive compared to the other a priori settings, with only 20 rules meeting the requirements.

```
rules4a = apriori(college_trans, parameter = list(sup = 0.1, conf = 0.6, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5     0.1      1
##  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: 730
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[39 item(s), 7308 transaction(s)] done [0.00s].
## sorting and recoding items ... [23 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [20 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules4b = apriori(college_trans, parameter = list(sup = 0.01, conf = 0.1, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  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: 73
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[39 item(s), 7308 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [3205 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules)
```

```
## set of 889 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4   5   6
##  19 249 429 181  11
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   4.000   3.906   4.000   6.000
##
## summary of quality measures:
##     support          confidence          lift
##  Min.   :0.01013   Min.   :0.6012   Min.   : 1.317
##  1st Qu.:0.01368   1st Qu.:0.7186   1st Qu.: 2.172
##  Median :0.01779   Median :0.8378   Median : 3.057
##  Mean   :0.02581   Mean   :0.8241   Mean   : 3.145
##  3rd Qu.:0.02942   3rd Qu.:0.9333   3rd Qu.: 3.649
##  Max.   :0.30624   Max.   :1.0000   Max.   :16.350
##
## mining info:
##           data ntransactions support confidence
##  college_trans          7308    0.01        0.6
```

```
summary(rules4a)
```

```
## set of 20 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3
## 12  8
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     2.0     2.0     2.4     3.0     3.0
##
## summary of quality measures:
##     support         confidence          lift
##  Min.   :0.1006   Min.   :0.6072   Min.   :1.319
##  1st Qu.:0.1136   1st Qu.:0.6715   1st Qu.:1.700
##  Median :0.1367   Median :0.7641   Median :1.839
##  Mean   :0.1466   Mean   :0.7708   Mean   :2.013
##  3rd Qu.:0.1453   3rd Qu.:0.8207   3rd Qu.:2.276
##  Max.   :0.3062   Max.   :1.0000   Max.   :3.848
##
## mining info:
##           data ntransactions support confidence
##  college_trans          7308     0.1        0.6
```

```
summary(rules4b)
```

```
## set of 3205 rules
##
## rule length distribution (lhs + rhs):sizes
##    1    2    3    4    5    6
##   23  449 1401 1020  300   12
```

```
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   3.362   4.000   6.000
##
## summary of quality measures:
##     support          confidence          lift
##  Min.   :0.01013   Min.   :0.1006   Min.   : 0.2195
##  1st Qu.:0.01300   1st Qu.:0.2654   1st Qu.: 1.2145
##  Median :0.01779   Median :0.4000   Median : 1.8180
##  Mean   :0.02685   Mean   :0.4674   Mean   : 2.0731
##  3rd Qu.:0.02969   3rd Qu.:0.6392   3rd Qu.: 2.6224
##  Max.   :0.46045   Max.   :1.0000   Max.   :16.3499
##
## mining info:
##           data ntransactions support confidence
##  college_trans          7308    0.01        0.1
```

```
rules3 = apriori(college_trans, parameter = list(sup = 0.01, conf = 0.6, target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.01      1
##  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: 73
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[39 item(s), 7308 transaction(s)] done [0.00s].
## sorting and recoding items ... [34 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [889 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

5. In the text we constructed earnings quartiles and explored the associations in top earners by filtering the rules for the top quartile of earners. Now, re-filter the rules to explore the bottom 25% of earners (Q1). Report at least 1 interesting finding. **Hint:** Use the subset and inspect functions to filter the left-hand side (lhs) for `earnings_quartiles=earnings_Q1`. When using filter here, do not add spaces to the categories, for example "earnings_quartiles = Q1" will not work.

There are not rhs rules with this subset using the support and confidence restrictions in the tutorial. There were 85 rules with this low earnings quartile on the lhs. I noticed that "control=Private for-profit" was common on the rhs when inspecting these rules. After subsetting on "control=Private for-profit" on the rhs too, there were 42 rules, about 50% of the total rules with low earning quartile on the lhs.

```
low_earners = subset(rules, subset = lhs %in% "earnings_quartiles=earnings_Q1")
low_earners
```

```
## set of 85 rules
```

```
inspect(head(low_earners))
```

```
##     lhs                                  rhs                                                      supp
## [1] {earnings_quartiles=earnings_Q1} => {pred_deg=Predominantly certificate-degree granting} 0.13943
## [2] {earnings_quartiles=earnings_Q1} => {control=Private for-profit}                          0.13533
## [3] {locale=City: Midsize,
##      earnings_quartiles=earnings_Q1} => {pred_deg=Predominantly certificate-degree granting} 0.01860
## [4] {locale=City: Midsize,
##      earnings_quartiles=earnings_Q1} => {control=Private for-profit}                          0.01778
## [5] {locale=City: Small,
##      earnings_quartiles=earnings_Q1} => {pred_deg=Predominantly certificate-degree granting} 0.01847
## [6] {locale=City: Small,
##      earnings_quartiles=earnings_Q1} => {control=Private for-profit}                          0.02038
```

```
low_earners_for_profit =  subset(rules, subset = lhs %in% "earnings_quartiles=earnings_Q1" &
                                 rhs %in% "control=Private for-profit")
low_earners_for_profit
```

```
## set of 42 rules
```