```
#################
# Association Rules for TripAdvisor
# Author: Ravi Makhija
# Version 1
#
# Description:
# We explore the TripAdvisor dataset using association rule mining.
#
# File Dependencies:
#   'data/tripadvisor_data.Rdata'
#
# How to run:
#    Source this script (no need to set wd beforehand if directory structure is
#    maintained as downloaded).
#
# References
#   1) Set working directory to the file path of a script:
#      http://stackoverflow.com/questions/13672720/r-command-for-setting-working-dire
#   2) Tutorial on association rules in R:
#      http://www.rdatamining.com/examples/association-rules
#   3) Renaming levels of a factor:
#      http://www.cookbook-r.com/Manipulating_data/Renaming_levels_of_a_factor/
#   4) Installing package from a source file:
#      https://cran.r-project.org/web/packages/arules/index.html

require("arules")   # version 2.2 is needed, which required installing from source

## Loading required package:  arules
## Warning:  package 'arules' was built under R version 3.2.2
## Loading required package:  Matrix
##
## Attaching package:  'arules'
##
## The following objects are masked from 'package:base':
##
##    %in%, abbreviate, write

require("arulesViz")

## Loading required package:  arulesViz
```

```
## Warning:  package 'arulesViz' was built under R version 3.1.3
## Loading required package:  grid
##
## Attaching package:  'arulesViz'
##
## The following object is masked from 'package:arules':
##
##    abbreviate
##
## The following object is masked from 'package:base':
##
##    abbreviate
```

```
require("Hmisc")
```

```
## Loading required package:  Hmisc
## Warning:  package 'Hmisc' was built under R version 3.1.3
## Loading required package:  lattice
## Loading required package:  survival
## Loading required package:  splines
## Loading required package:  Formula
## Warning:  package 'Formula' was built under R version 3.1.3
## Loading required package:  ggplot2
##
## Attaching package:  'Hmisc'
##
## The following objects are masked from 'package:base':
##
##    format.pval, round.POSIXt, trunc.POSIXt, units
```

```
require("data.table")
```

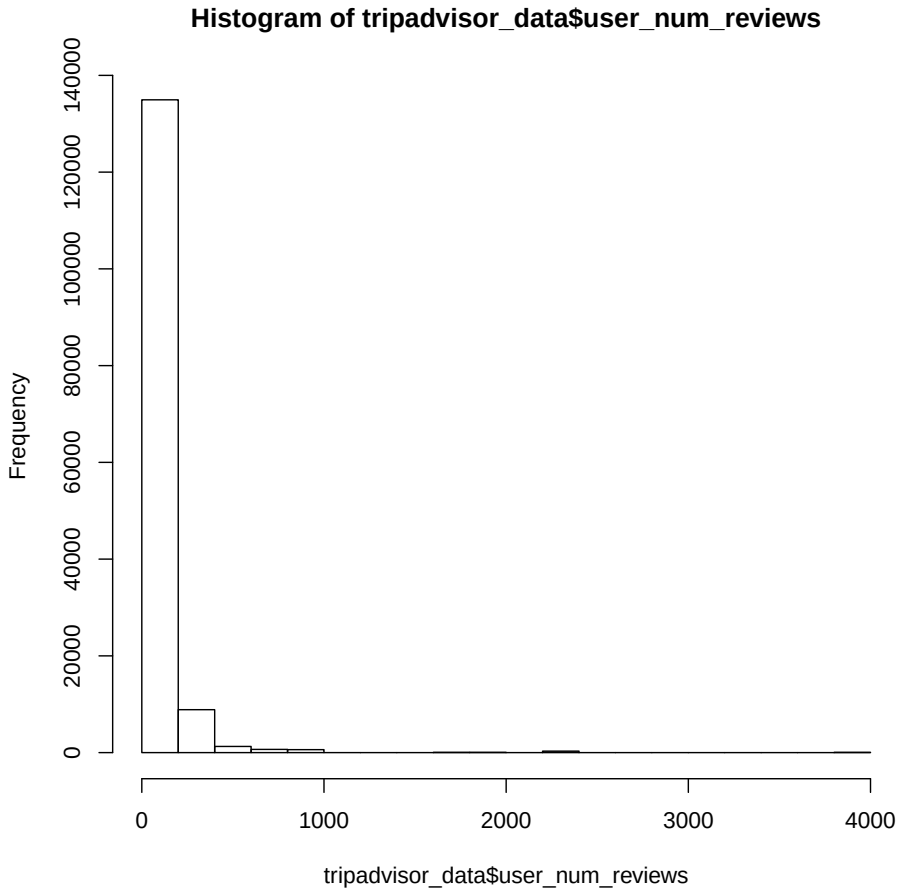
```
## Loading required package:  data.table
## Warning:  package 'data.table' was built under R version 3.1.3
```

```
require("plyr")
```

```
## Loading required package:  plyr
##
## Attaching package:  'plyr'
##
## The following objects are masked from 'package:Hmisc':
```

```
##
##    is.discrete, summarize

#################
# Load in data
#################

load("tripadvisor_data.Rdata")

#################
# Prep data for association rules
#################

# Create a new data.frame for the data set we want to use association rules on.
# We want to create categorical variables for this purpose.

tripadvisor_data_categorical <- data.frame(user_is_local = as.factor(tripadvisor_data

# Now, we bin some continuous variables and add to this new data set.

###################
# user_review_length
# We omit this for TripAdvisor, since the data is incomplete with some of the
# reviews being cut off (e.g. they end with the word "More").

###################
# user_rating

# Explore data
table(tripadvisor_data$user_rating)




# Bin and add to new data set:
# For the time being, we keep all five categories:
tripadvisor_data_categorical$user_rating <- as.factor(tripadvisor_data$user_rating)

###################
```

```r
# user_num_reviews

# Explore data
summary(tripadvisor_data$user_num_reviews)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   16.00   42.00   82.48   93.00 3834.00

hist(tripadvisor_data$user_num_reviews)
```




```r
# Bin and add to new data set:
# low: [1 to 16)
```

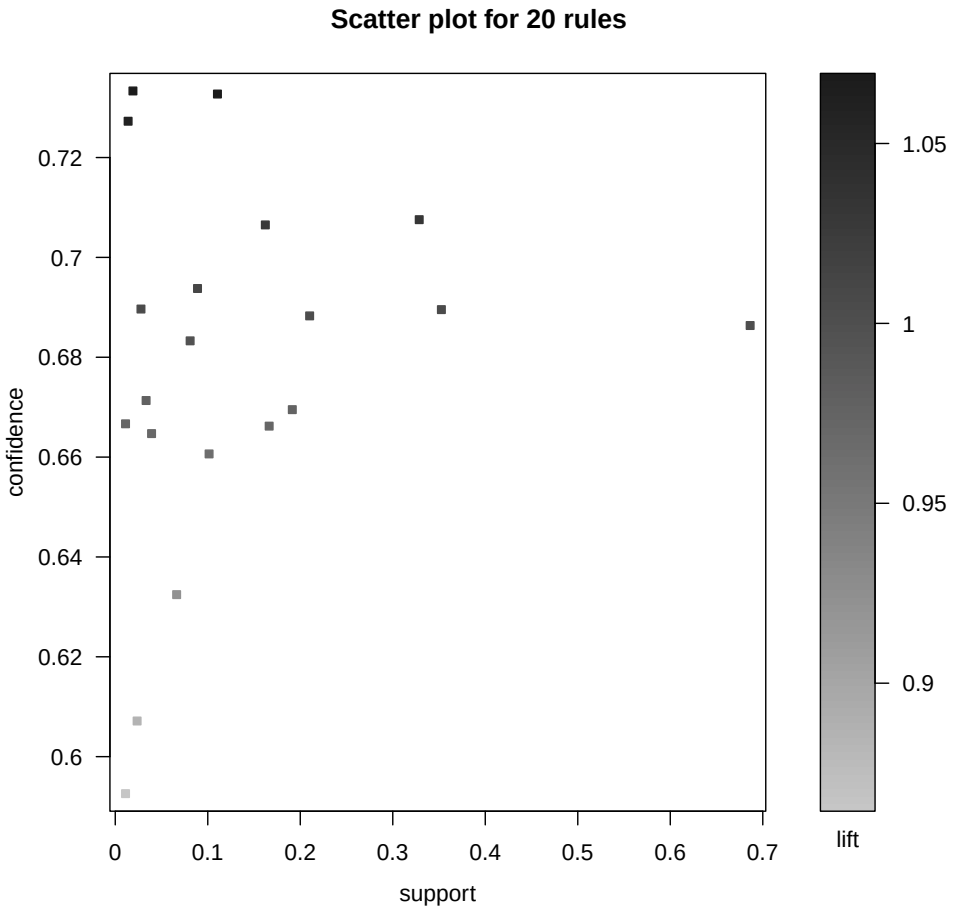
```
# medium: [16 to 93)
# high: [83 and up)
tripadvisor_data_categorical$user_num_reviews <- cut2(x=tripadvisor_data$user_num_rev
                                                      cut=c(1, 16, 83))

###################
# Check out the new data set
head(tripadvisor_data_categorical)

##   user_is_local user_rating user_num_reviews
## 1         FALSE           4      [   1,  16)
## 2         FALSE           4      [   1,  16)
## 3          TRUE           2      [   1,  16)
## 4         FALSE           5      [  83,3834]
## 5         FALSE           4      [   1,  16)
## 6         FALSE           5      [  83,3834]

#################
# Start association rules mining for tripadvisor
#################

attach(tripadvisor_data_categorical)

# Since a central question we are asking is whether or not local or non_local
# ratings are higher, we start association rule mining with the binary
# user_is_local on the right, to see if we can find any implications. We
# adjust the minimum support and confidence levels to obtain the most
# meaningful rule set. Just as we did for Yelp.

# A first look shows that generally speaking, confidence levels for TripAdvisor
# are much lower than for Yelp. This seems to be in line with the fact that
# the difference in mean local/non-local user ratings for TripAdvisor was
# much smaller than for Yelp. E.g. without large difference, we are not seeing
# much confidence in implying a local or non-local user.

tripadvisor_rules_1 <- apriori(tripadvisor_data_categorical,
                               parameter = list(minlen=1, supp=.01, conf=.5),
                               appearance = list(rhs=c("user_is_local=FALSE", "user_i
                               control = list(verbose=F))
```

```
## Warning in match(x, table, nomatch = 0):  bytecode version mismatch;
using eval
```

```
inspect(tripadvisor_rules_1)
```

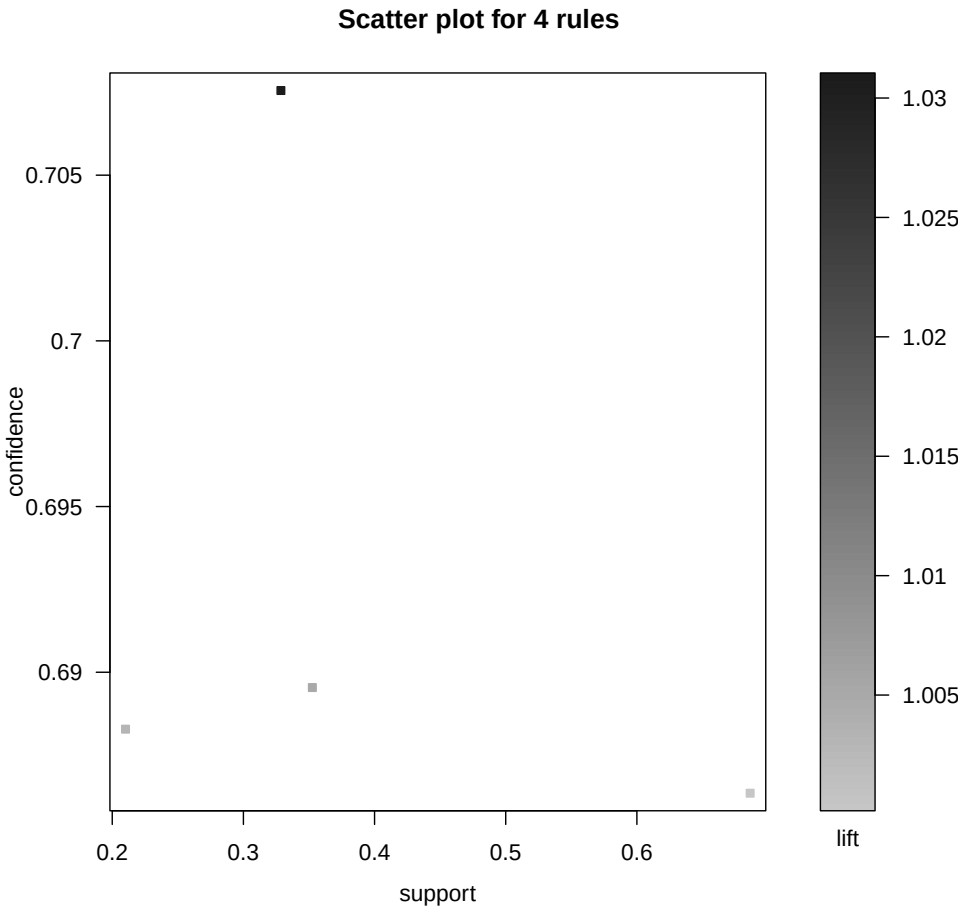
```
##    lhs                                rhs                       support confidence
## 1  {}                              => {user_is_local=FALSE} 0.68634943  0.6863494 1
## 2  {user_rating=1}                 => {user_is_local=FALSE} 0.01910386  0.7333333 1
## 3  {user_rating=2}                 => {user_is_local=FALSE} 0.02361931  0.6071429 0
## 4  {user_rating=3}                 => {user_is_local=FALSE} 0.08093088  0.6832845 0
## 5  {user_num_reviews=[   1,  16)} => {user_is_local=FALSE} 0.16637721  0.6662031 0
## 6  {user_num_reviews=[  83,3834]} => {user_is_local=FALSE} 0.19138590  0.6695018 0
## 7  {user_rating=4}                 => {user_is_local=FALSE} 0.21014241  0.6882821 1
## 8  {user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.32858631  0.7075542 1
## 9  {user_rating=5}                 => {user_is_local=FALSE} 0.35255297  0.6895380 1
## 10 {user_rating=1,
##     user_num_reviews=[   1,  16)} => {user_is_local=FALSE} 0.01111497  0.6666667 0
## 11 {user_rating=2,
##     user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.01111497  0.5925926 0
## 12 {user_rating=3,
##     user_num_reviews=[   1,  16)} => {user_is_local=FALSE} 0.01389371  0.7272727 1
## 13 {user_rating=3,
##     user_num_reviews=[  83,3834]} => {user_is_local=FALSE} 0.02778743  0.6896552 1
## 14 {user_rating=3,
##     user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.03924974  0.6647059 0
## 15 {user_rating=4,
##     user_num_reviews=[   1,  16)} => {user_is_local=FALSE} 0.03334491  0.6713287 0
## 16 {user_rating=5,
##     user_num_reviews=[   1,  16)} => {user_is_local=FALSE} 0.10142411  0.6606335 0
## 17 {user_rating=4,
##     user_num_reviews=[  83,3834]} => {user_is_local=FALSE} 0.06634248  0.6324503 0
## 18 {user_rating=5,
##     user_num_reviews=[  83,3834]} => {user_is_local=FALSE} 0.08891976  0.6937669 1
## 19 {user_rating=4,
##     user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.11045502  0.7327189 1
## 20 {user_rating=5,
##     user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.16220910  0.7065053 1
```

```
plot(tripadvisor_rules_1)
```




```
# Narrowing down by increasing the minimum support, however, shows that
# ratings of 4 or 5, e.g. the higher ratings, are associated with non-local
# users. This is also what we found with the parametric test! E.g. the t-test
# showed that, while the difference was small, indeed there was a statistically
# significant difference in mean user rating between non-local and local, in
# favor of non-local reviews.

tripadvisor_rules_2 <- apriori(tripadvisor_data_categorical,
                               parameter = list(minlen=1, supp=.2, conf=.5),
                               appearance = list(rhs=c("user_is_local=FALSE", "user_i
                               control = list(verbose=F))
inspect(tripadvisor_rules_2)
```

```
##   lhs                                 rhs                       support confidence
## 1 {}                                => {user_is_local=FALSE} 0.6863494  0.6863494 1.0
## 2 {user_rating=4}                   => {user_is_local=FALSE} 0.2101424  0.6882821 1.0
## 3 {user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.3285863  0.7075542 1.0
## 4 {user_rating=5}                   => {user_is_local=FALSE} 0.3525530  0.6895380 1.0

plot(tripadvisor_rules_2)
```

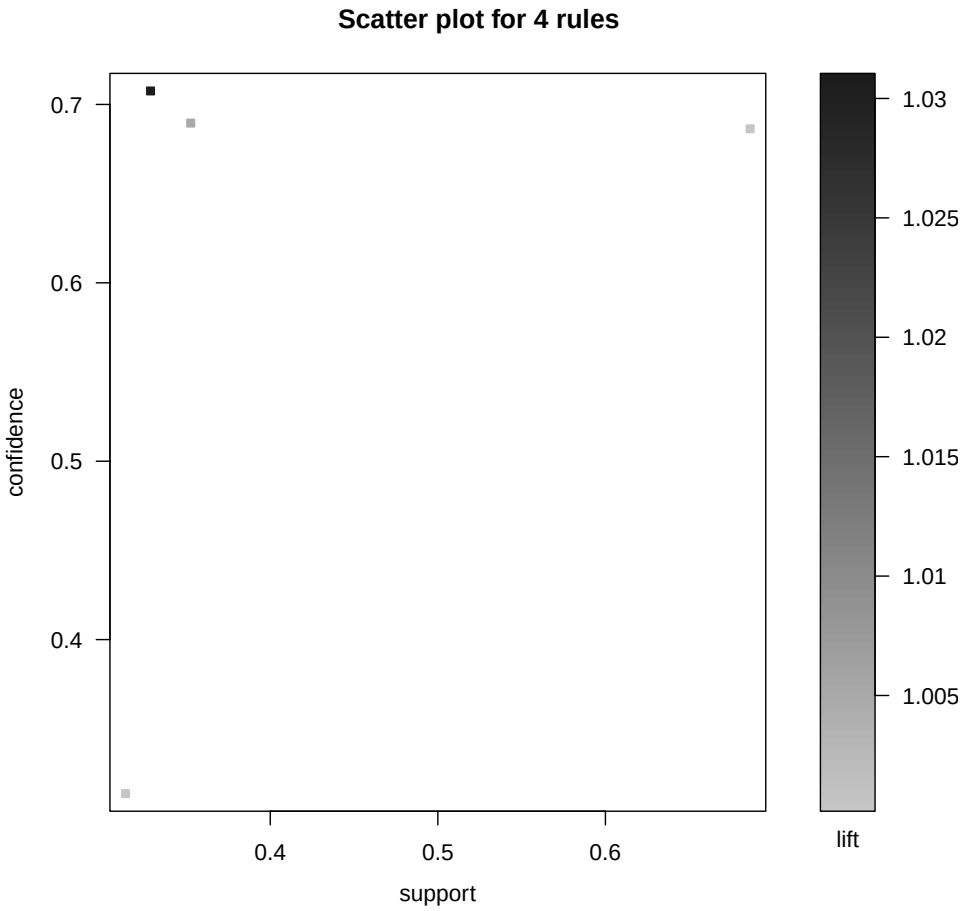



```
# We use one more level of support to mine the most frequent item sets. This
# however does not bring up any new information when compared to the last
# rule set.
```

```
tripadvisor_rules_3 <- apriori(tripadvisor_data_categorical,
                               parameter = list(minlen=1, supp=.3, conf=.1),
                               appearance = list(rhs=c("user_is_local=FALSE", "user_i
                               control = list(verbose=F))
inspect(tripadvisor_rules_3)
```

```
##   lhs                                rhs                     support confidence
## 1 {}                              => {user_is_local=TRUE}  0.3136506  0.3136506 1.0
## 2 {}                              => {user_is_local=FALSE} 0.6863494  0.6863494 1.0
## 3 {user_num_reviews=[  16,  83)} => {user_is_local=FALSE} 0.3285863  0.7075542 1.0
## 4 {user_rating=5}                 => {user_is_local=FALSE} 0.3525530  0.6895380 1.0
```

```
plot(tripadvisor_rules_3)
```


Scatter plot for 4 rules
confidence
0.7
0.6
0.5
0.4
0.4
0.5
0.6
support
lift
1.03
1.025
1.02
1.015
1.01
1.005