Sampling
Introduction to Data Science

Team 15:
Thijs van der Knaap (s2752077)
Hatim Alsayahani (s3183696)
Sebastian Wehkamp (s2589907)
Dimitris Laskaratos (s3463702)

October 2, 2017

1 Introduction: confidence and support

1.1 Importance of lift

Given rule $X \to Y$

- **Confidence** is the number of cases in which the rule is correct relative to the number of cases in which it is applicable.
- **Support** is the percentage of transactions that contain a given itemset
- **Lift** is the ratio of the observed support to that expected if X and Y were independent.

This means that confidence does not take into account the amount of times $X \to Y$ occurs by random. With the lift ratio you can clearly see this since when the lift is one it implies that X and Y are independent of each other and no association rule can be drawn from this. If the lift is higher it means that there is at least some relationship between X and Y.

1.2 Association Rules

In order to conduct an association analysis, the first step is to identify items that are more likely to occur together.
In this case we have the items: name, age, gender, hobbies, favourite colour, income and country and more.
The item 'name' is not suitable to be included in the analysis since it is seldom associated with the rest of the items of the set.

Taking the above into consideration we can find association rules:
- between age and hobbies,
- between age and favourite colour,
- between hobbies and country,
- between income and country.
Etc.. . . . .

1.3 Wine and Cheese

Let wine=X and cheese=Y. The lift of rule

$$wine \Rightarrow cheese$$

is thus

$$Lift(X \to Y) = \frac{Support(X \cup Y)}{Support(X) \times Support(Y)}.$$

Since

$$Lift(X \to Y) = 2$$

and

$$Support(X) = 0.1$$

we get

$$0.2 = \frac{Support(X \cup Y)}{Support(Y)}$$

or

$$\frac{1}{5} = \frac{Support(X \cup Y)}{Support(Y)}.$$

That means cheese alone is 5 times more frequent in transactions than bread and cheese together.

1.4 Itemsets

Item	support
{1}	16.7%
{2}	50.0%
{3}	66.7%
{4}	50.0%
{5}	66.7%
{6}	33.3%
{7}	16.7%
{2,3}	33.3%
{2,4}	33.3%
{2,5}	16.7%
{2,6}	0%
{3,4}	50.0%
{3,5}	50.0%
{3,6}	16.7%
{4,5}	33.3%
{4,6}	16.7%
{5,6}	33.3%
{2,3,4}	33.3%
{3,4,5}	33.3%

From the above table only rules with a minimum Support of 0.3 will be used to calculate their Confidence. So we have the itemsets: {{2,3}, {2,4}, {3,4}, {3,5} {4,5}, {4,5}, {5,6}, {2,3,4}, {3,4,5}}.

By applying these itemsets to the confidence equation,

$$Confidence(X \rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}$$

where X and Y can be rules on their own, we get that the association rules that produce a minimum Support of 0.3 and a minimum Confidence of 0.7 are the following:

$\{2\} \rightarrow \{3\}$
$\{2\} \rightarrow \{4\}$
$\{3\} \rightarrow \{4\}$
$\{3\} \rightarrow \{5\}$
$\{6\} \rightarrow \{5\}$
$\{2,3\} \rightarrow \{4\}$
$\{2,4\} \rightarrow \{3\}$
$\{4,5\} \rightarrow \{3\}$

These rules are found by considering all possible associations in a data set and consequently applying Confidence on each of those. Only the rules with

$$Confidence \geq 0.7$$

are ultimately accepted.

2 Beethoven and Iron Maiden

2.1 Reading the data

We are reading the data by using the code in the listing below.

```
data <- read.csv(file.choose(new = FALSE), header=FALSE, stringsAsFactors=TRUE, sep=
    "\t")
names(data) <- c('user', 'timestamp', 'mbid1', 'artist', 'mbid2', 'song')

keeps <- c('user', 'artist')
data <- data[keeps]
```

Some interesting statistics about the dataset, note that these statistics were also made using the user-profile file.

```
            Country
 United States :204385
 United Kingdom: 98766
 Canada        : 70477
 Peru          : 57442
 Germany       : 46712
 Turkey        : 45335
 (Other)       :313714

          Registered
 Mar 30, 2005: 75877
 Mar 16, 2005: 70447
 Nov 10, 2005: 66609
 Feb 24, 2006: 57439
 May 14, 2006: 51475
 Dec 21, 2005: 45308
```

```
(Other)       :469676

  Gender
  :147480
 f:289362
 m:399989

                   Band
 Kanye West        :  27115
 Radiohead         :   6938
 Nine Inch Nails:   6529
 Muse              :   5459
 ????              :   5394
 T.I.              :   5394
 (Other)           :780002
```

2.2 Recommendations based on Beethoven

We are going to make some suggestions based on someone liking Beethoven alot. To do this we will see which user really likes Beethoven a lot using the code below. After which we will see which other bands that user likes as well. Those bands will be our recommendations.

```
trans <- as(data, "transactions")

rules <- apriori(trans,
                 parameter = list(minlen=2, supp = 0.00000000001, conf = 0.0000000000000001),
                 appearance = list(lhs = c("artist=Ludwig Van Beethoven"), default = "rhs")
                 )
inspect(rules)
```

This results in the listing below

```
     lhs                                rhs                  support      confidence  lift       count
[1]  {artist=Ludwig Van Beethoven} => {user=user_000020} 1.425033e-05 0.13333333 12.0263639  6
[2]  {artist=Ludwig Van Beethoven} => {user=user_000013} 2.375054e-06 0.02222222 1.7801581   1
[3]  {artist=Ludwig Van Beethoven} => {user=user_000028} 2.375054e-06 0.02222222 1.2611553   1
[4]  {artist=Ludwig Van Beethoven} => {user=user_000003} 1.187527e-05 0.11111111 5.9195945   5
[5]  {artist=Ludwig Van Beethoven} => {user=user_000006} 2.375054e-06 0.02222222 1.0494068   1
[6]  {artist=Ludwig Van Beethoven} => {user=user_000025} 1.187527e-05 0.11111111 3.2699067   5
[7]  {artist=Ludwig Van Beethoven} => {user=user_000001} 4.750109e-06 0.04444444 1.1252569   2
[8]  {artist=Ludwig Van Beethoven} => {user=user_000017} 9.500217e-06 0.08888889 2.0686516   4
```

```
[9]  {artist=Ludwig Van Beethoven} => {user=user_000031} 4.750109e-06 0.04444444
    0.9080024  2
[10] {artist=Ludwig Van Beethoven} => {user=user_000019} 4.750109e-06 0.04444444
    0.6798308  2
[11] {artist=Ludwig Van Beethoven} => {user=user_000012} 2.375054e-05 0.22222222
    2.3487577 10
[12] {artist=Ludwig Van Beethoven} => {user=user_000026} 1.425033e-05 0.13333333
    1.2390542  6
```

User 000020 is the most interesting user since it has the highest lift value. Now we will use the apriori algorithm again to determine which artists "User 00020" also likes.

```
    rules <- apriori(trans,
                     parameter = list(minlen=2, supp = 0.0001, conf = 0.0000001),
                     appearance = list(lhs = c("user=user_000020"), default = "rhs")
                     )
```

resulting in

```
          lhs                              rhs                                    support
              confidence   lift        count
[1]  {user=user_000020} => {artist=Nicky Wire}                         0.0001068774
    0.009640103 90.1977292  45
[2]  {user=user_000020} => {artist=The Blood Brothers}                 0.0001235028
    0.011139674 76.8898675  52
[3]  {user=user_000020} => {artist=L'Arc~En~Ciel}                      0.0001068774
    0.009640103 42.2801856  45
[4]  {user=user_000020} => {artist=Iamx}                               0.0002042547
    0.018423308 68.0439010  86
[5]  {user=user_000020} => {artist=Moneybrother}                       0.0002707562
    0.024421594 87.8849669 114
[6]  {user=user_000020} => {artist=The Dandy Warhols}                  0.0001638787
    0.014781491 36.1839728  69
[7]  {user=user_000020} => {artist=The Cooper Temple Clause}           0.0001686289
    0.015209940 32.6736672  71
[8]  {user=user_000020} => {artist=Patrick Wolf}                       0.0002280052
    0.020565553 43.5124724  96
[9]  {user=user_000020} => {artist=Kasabian}                           0.0002945067
    0.026563839 43.8608566 124
[10] {user=user_000020} => {artist=Manic Street Preachers}             0.0001401282
    0.012639246 18.7382606  59
[11] {user=user_000020} => {artist=Black Rebel Motorcycle Club}        0.0004702608
    0.042416452 33.3815895 198
[12] {user=user_000020} => {artist=The Mars Volta}                     0.0001828792
    0.016495287 10.7345056  77
[13] {user=user_000020} => {artist=Dirty Pretty Things}                0.0002398805
    0.021636675 13.4167462 101
[14] {user=user_000020} => {artist=Mando Diao}                         0.0004797610
    0.043273350 24.1643784 202
[15] {user=user_000020} => {artist=Queens Of The Stone Age}            0.0001971295
    0.017780634  7.2966974  83
[16] {user=user_000020} => {artist=The Killers}                        0.0001472534
    0.013281919  5.2312995  62
[17] {user=user_000020} => {artist=Dredg}                              0.0002660061
    0.023993145  8.6713697 112
```

```
[18] {user=user_000020} => {artist=The Strokes}                    0.0002873816
     0.025921165   8.5265041  121
[19] {user=user_000020} => {artist=Babyshambles}                   0.0001021273
     0.009211654   2.6366433   43
[20] {user=user_000020} => {artist=The Libertines}                 0.0001805041
     0.016281063   3.5684682   76
[21] {user=user_000020} => {artist=Muse}                           0.0019522947
     0.176092545  20.2963409  822
[22] {user=user_000020} => {artist=Radiohead}                      0.0001021273
     0.009211654   0.8743242   43
```

So we can recommend for example "Nicky Wire" and "Moneybrother" since those lift values are very high. These are sort of suprising recommendations since most of the bands listed above are alternative rockbands which is quite different from Beethoven.

2.3 Make someone like Eminem

For this assignment we concatenated for every user the music that it likes into one colum.

```
    merged <- dcast(setDT(data), user ~ rowid(user), value.var = c('artist'), fill
       = '')
```

We planned on running the apriori on this merged dataset, but due to limited stack size this was not possible. Each time we ran it the size surpassed 4.2 Gb, which is max available RAM at the lab computers. If it would have been possible to run this command we would search for a chain of artists that would persuade a Beethoven fan to like Eminem music. From Beethoven we would search for a relation to an artist that either directly or indirectly(by repeating this step) has a high chance of also liking Eminem.

2.4 Merge with user profiles

We merged both files using a python dataframe and export it again to a csv file named "merged.csv". This is done using the code from the listing below.

```
import pandas as pd

lastFM = pd.read_csv('lastFM.tsv', sep='\t')
lastFM.columns = ['User', 'Timestamp', 'MBIDBand', 'Band', 'MBIDSong', 'Song']
lastFM = lastFM[['User', 'Timestamp', 'Band', 'Song']]

profiles = pd.read_csv('userid-profile.tsv', sep='\t')
profiles.columns = ['User', 'Gender', 'Age', 'Country', 'Registered']

merged = pd.merge(lastFM, profiles, how='right', on='User')

merged.to_csv('merged.csv')
```

The generated CSV file could be used to for example check which bands are popular in a specific country, gender, or age. Below is example code which shows what is popular in the "United States".

```r
library(arules)
library(data.table)

data <- read.csv(file.choose(new = FALSE), header=TRUE, stringsAsFactors=TRUE)

keeps <- c('User', 'Timestamp', 'Band', 'Song', 'Gender', 'Country', 'Registered')
data <- data[keeps]

rules <- apriori(data,
                 parameter = list(minlen=2, supp = 0.0001, conf = 0.005),
                 appearance = list(lhs = c("Country=United States"), default = "rhs")
                )
```

Resulting the sample below. In this result you can see that for example the band "Sasha & John Digweed" is popular in the "United States".

```
     lhs                          rhs                          support     confidence  lift      count
[1]  {Country=United States} => {Band=Sasha & John Digweed} 0.001276243 0.005225432 4.0867325 1068
[2]  {Country=United States} => {Band=Matthew Good}         0.001384987 0.005670671 4.0943856 1159
[3]  {Country=United States} => {Band=Gomez}                0.001350332 0.005528781 3.9816314 1130
[4]  {Country=United States} => {Band=Spoon}                0.001371842 0.005616851 3.7245283 1148
[5]  {Country=United States} => {Band=They Might Be Giants} 0.001645494 0.006737285 4.0825264 1377
[6]  {Country=United States} => {Band=Broken Social Scene}  0.001471026 0.006022947 3.0921403 1231
[7]  {Country=United States} => {Band=Elbow}                0.001714803 0.007021063 3.5565638 1435
[8]  {Country=United States} => {Band=Band Of Horses}       0.001910780 0.007823470 3.9391833 1599
[9]  {Country=United States} => {Band=Cut Copy}             0.002069713 0.008474203 3.5316115 1732
[10] {Country=United States} => {Band=The Verve}            0.001960969 0.008028965 3.0609962 1641
[11] {Country=United States} => {Band=Pixies}               0.001479391 0.006057196 2.3050702 1238
[12] {Country=United States} => {Band=Boards Of Canada}     0.002356509 0.009648458 3.5089650 1972
[13] {Country=United States} => {Band=Blur}                 0.002202356 0.009017296 3.1612705 1843
[14] {Country=United States} => {Band=Sasha}                0.002842868 0.011639797 4.0704318 2379
```

2.5 Give Iron Maiden CD to your friends

You could create a profile using Data Mining of a person who is very likely to like it. You could use data mining to determine artists who users like that also like Iron Maiden and some artists which they are sure not to like. You can ask your friends to grade these bands and see who is a perfect match for Iron Maiden. You can also combine this with factual data like country, gender, and age to do some extra matching.