Fundamentals of Applied Data Science with R	
	Association Rules
	Individual Assignment 3 Report
Name	Ahmed Shehata Mahmoud AboMoustafa
E-Mail	aabom018@uOttawa.ca

Part A.1: Association Rules

a) Find all frequent item sets in database X

Iteration 1

C1

Item	frequency	Support (0.25)
A	5	0.625
B	4	0.5
C	5	0.625
D	6	0.75
E	1	0.125
F	4	0.5
G	5	0.625

F1

Item	Frequency	Support (0.25)
A	5	0.625
B	4	0.5
C	5	0.625
D	6	0.75
F	4	0.5
G	5	0.625

Iteration 2

C2

Item	Frequency	Support (0.25)
A, B	3	0.375
A, C	3	0.375
A, D	4	0.5
A, F	2	0.25
A, G	2	0.25
B, C	2	0.25
B, D	2	0.25
B, F	1	0.125
B, G	2	0.25
C, D	4	0.5
C, F	2	0.25
C, G	3	0.375
D, F	4	0.5
D, G	3	0.375
F, G	2	0.25

F2

Item	Frequency	Support (0.25)
A, B	3	0.375
A, C	3	0.375
A, D	4	0.5
A, F	2	0.25
A, G	2	0.25
B, C	2	0.25
B, D	2	0.25
B, G	2	0.25
C, D	4	0.5
C, F	2	0.25
C, G	3	0.375
D, F	4	0.5
D, G	3	0.375
F, G	2	0.25

Iteration 3

C3

Item	Frequency	Support (0.25)
A, B, C	1	0.125
A, B, D	2	0.25
A, B, F	1	0.125
A, B, G	1	0.125
A, C, D	3	0.375
A, C, F	1	0.125
A, C, G	1	0.125
A, D, F	2	0.25
A, D, G	1	0.125
A, F, G	0	0
B, C, D	1	0.125
B, C, F	0	0
B, C, G	1	0.125
C, D, F	2	0.25
C, D, G	2	0.25
D, F, G	2	0.25

F3

Item	Frequency	Support
A, B, D	2	0.25
A, C, D	3	0.375
A, D, F	2	0.25
C, D, F	2	0.25
C, D, G	2	0.25
D, F, G	2	0.25

B) Find strong association rules for database X

Item set 1	**Confidence**	**>= 0.6**
	A, B, D	
{A, B} → D	2/3 = 0.67	Accepted
{A, D} → B	2/4 = 0.5	Rejected
{B, D} → A	2/2 = 1	Accepted
A → {B, D}	2/5 = 0.4	Rejected
B → {A, D}	2 /4 = 0.5	Rejected
D → {B, A}	2/6 = 0.33	Rejected

Item set 2	**Confidence**	**>= 0.6**
	A, C, D	
{A, C} → D	3/3 = 1	Accepted
{A, D} → C	3/4 = 0.75	Accepted
{C, D} → A	3/4 = 0.75	Accepted
A → {C, D}	3/5 = 0.6	Accepted
C → {A, D}	3/5 = 0.6	Accepted
D → {C, A}	3/6 = 0.5	Rejected

Item set 3	**Confidence**	**>= 0.6**
	A, D, F	
{A, F} → D	2/2 = 1	Accepted
{A, D} → F	2/4 = 0.5	Rejected
{F, D} → A	2/4 = 0.5	Rejected
A → {F, D}	2/5 = 0.4	Rejected
F → {A, D}	2/4 = 0.5	Rejected
D → {F, A}	2/6 = 033	Rejected

Item set 4	**Confidence**	**>= 0.6**
	C, D, F	
{C, D} → F	2/4 = 0.5	Rejected
{C, F} → D	2/2 = 1	Accepted
{F, D} → C	2/4 = 0.5	Rejected
F → {C, D}	2/4 = 0.5	Rejected
C → {D, F}	2/5 = 0.4	Rejected
D → {C, F}	2/6 = 033	Rejected

Item set 5	**Confidence**	**>= 0.6**
	C, D, G	
{C, D} → G	2/4 = 0.5	Rejected
{C, G} → D	2/3 = 0.67	Accepted
{D, G} → C	2/3 = 0.67	Accepted
G → {C, D}	2/5 = 0.4	Rejected
C → {G, D}	2/5 = 0.4	Rejected
D → {G, C}	2/6 = 033	Rejected

Item set 6	**Confidence**	**>= 0.6**
	D, F, G	
{D, F} → G	2/4 = 0.5	Rejected
{D, G} → F	2/3 = 0.67	Accepted
{F, G} → D	2/2 = 1	Accepted
D → {F, G}	2/6 = 0.33	Rejected
F → {D, G}	2/4 = 0.5	Rejected
G → {D, F}	2/5 = 04	Rejected

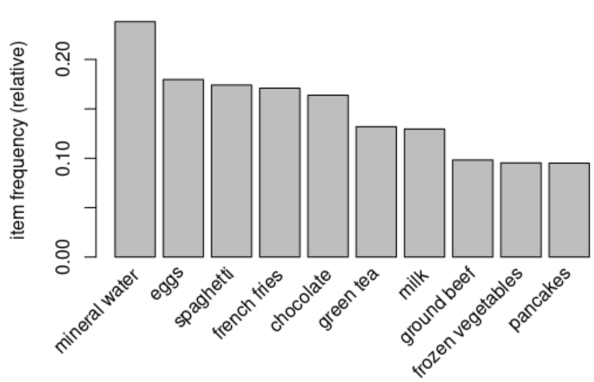
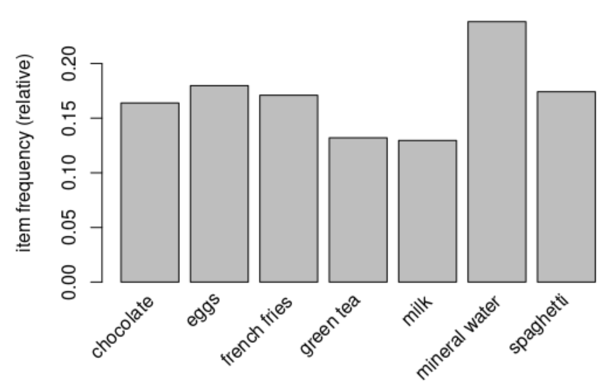
C) Analyze misleading associations for the rule set obtained in (b):

Accepted Rule	Lift	Situation
{A, B} → D	0.25 / (0.375 * 0.75) = 0.89	Negative
{B, D} → A	0.25 / (0.25 * 0.625) = 1.6	Positive
{A, C} → D	0.375 / (0.375 * 0.75) = 1.3	Positive
{A, D} → C	0.375 / (0.5 * 0.625) = 1.2	Positive
{C, D} → A	0.375 / (0.5 * 0.625) = 1.2	Positive
A → {C, D}	0.375 / (0.5 * 0.625) = 1.2	Positive
C → {A, D}	0.375 / (0.5 * 0.625) = 1.2	Positive
{A, F} → D	0.25 / (0.25 * 0.75) = 1.3	Positive
{C, F} → D	0.25 / (0.25 * 0.75) = 1.3	Positive
{D, G} → C	0.25 / (0.375 * 0.625) =1.067	Positive
{D, G} → F	0.25 / (0.375 * 0.5) = 1.3	Positive
{C, G} → D	0.25 / (0.375 * 0.75) = 0.89	Negative
{F, G} → D	0.25 / (0.25 * 0.75) = 1.3	Positive

Part A.2: Association Rules

a) Generate a plot of the top 10 transactions

```
# Generate a plot of the top 10 transactions
itemFrequencyPlot(trans_df, support = 0.1)
itemFrequencyPlot(trans_df, topN = 10)
```

Top 10	Support
	

b) Generate association rules using minimum support of 0.002, minimum confidence of 0.20, and maximum length of 3. Display the rules, sorted by descending lift value.

```
# set parameters support, confidence, and maxlen levels.
trans_rules_1 <- apriori(trans_df, parameter = list(support = 0.002,
                                                    confidence =0.20,
                                                    maxlen = 3))

trans_rules_1
# Display the rules, sorted by descending lift value
l <- inspect(sort(trans_rules_1, by = "lift"))
# Display the rules, sorted by descending support value
s <- inspect(sort(trans_rules_1, by = "support"))
```

- **Resulted rules sorted descending by lift value**

```
> trans_rules_1
set of 2186 rules
> # Display the rules, sorted by descending lift value
> l <- inspect(sort(trans_rules_1, by = "lift"))
     lhs                                  rhs                   support     confidence coverage    lift      count
[1]  {escalope,mushroom cream sauce}     => {pasta}              0.002533333 0.4418605  0.005733333 28.084352  19
[2]  {escalope,pasta}                    => {mushroom cream sauce} 0.002533333 0.4318182  0.005866667 22.647807  19
[3]  {mushroom cream sauce,pasta}        => {escalope}           0.002533333 0.9500000  0.002666667 11.974790  19
[4]  {parmesan cheese,tomatoes}          => {frozen vegetables}  0.002133333 0.6666667  0.003200000  6.993007  16
[5]  {mineral water,whole wheat pasta}   => {olive oil}          0.003866667 0.4027778  0.009600000  6.127451  29
[6]  {frozen vegetables,parmesan cheese} => {tomatoes}           0.002133333 0.3902439  0.005466667  5.705320  16
[7]  {burgers,herb & pepper}             => {ground beef}        0.002266667 0.5483871  0.004133333  5.580601  17
[8]  {light cream,mineral water}         => {chicken}            0.002400000 0.3272727  0.007333333  5.454545  18
[9]  {french fries,mushroom cream sauce} => {escalope}           0.002000000 0.4285714  0.004666667  5.402161  15
[10] {fromage blanc}                     => {honey}              0.003333333 0.2450980  0.013600000  5.178128  25
```

C)

Lift with max length: 3

```
     lhs                                 rhs        support     confidence coverage    lift      count
[1]  {escalope,mushroom cream sauce}  => {pasta}    0.002533333 0.4418605  0.005733333 28.084352  19
```

Lift with max length: 2

```
     lhs                 rhs        support     confidence coverage    lift     count
[1]  {fromage blanc}  => {honey}    0.003333333 0.2450980  0.013600000 5.178128  25
```

From this table:

- Rule based on lift max length of 3 has the highest value so it's considered the best rule according to the lift

  {escalope, mushroom cream sauce} → {Pasta}

- According to support, the rule based on max length of 2 is the best

  {fromage blanc} → {honey}

- If I were a marketing manager, I will go through the rule based on max length of 3 because it has the highest lift and confidence. and this will lead to improving the company profit as we sure that lots of people buy those items together.

Part B

1. **First consider a user-based collaborative filter. This requires computing correlations between all student pairs. For which students is it possible to compute correlations with E.N.? Compute them.**

   a. requires computing correlations between all student pairs
      Compute average ratings for all students

User	LN	MH	JH	EN	DU	FL	GL	AH	SA	RW	BA	MG	AF	KG	DS
Avg	3	3.66	2	3.75	4	4	4	3	4	3	4	4	4	3	3.3

   b. For which students is it possible to compute correlations with E.N.? Compute them

$$\textbf{LN} \longrightarrow \text{Corr(LN, EN)} = \frac{(4-3)(4-3.75)+(4-3)(4-3.75)+(2-3)(3-3.75)}{\sqrt{(4-3)^2+(4-3)^2+(2-3)^2} * \sqrt{(4-3.75)^2+(4-3.75)^2+(3-3.75)^2}} = \mathbf{0.87}$$

$$\textbf{MH} \longrightarrow \text{Corr(MH, EN)} = \frac{(3-3.66)(4-3.75)}{\sqrt{(3-3.66)^2}\sqrt{(4-3.75)^2}} = \mathbf{-1}$$

$$\textbf{JH} \longrightarrow \text{Corr(JH, EN)} = \frac{(2-2)(4-3.75)}{\sqrt{(2-2)^2} * \sqrt{(4-3.75)^2}} = \mathbf{0}$$

$$\textbf{DU} \longrightarrow \text{Corr(DU, EN)} = \frac{(4-4)(4-3.75)}{\sqrt{(4-4)^2 * \sqrt{(4-3.75)^2}}} = \mathbf{0}$$

$$\textbf{DS} \longrightarrow \text{Corr(DS, EN)} = \frac{(4-3.3)(4-3.75)+(2-3.3)(4-3.75)+(4-3.3)(4-3.75)}{\sqrt{(4-3.3)^2+(2-3.3)^2+(4-3.3)^2} * \sqrt{(4-3.75)^2+(4-3.75)^2+(4-3.75)^2}} = \mathbf{0.003553}$$

From calculation, we can conclude that, **LN** student has the highest correlation with EN as they share 3 ratings for the shared courses

2. **Based on the single nearest student to E.N., which single course should we recommend to E.N.? Explain why**

   a. Based on the nearest student to EN which is LN, he can recommend two courses for LN which are **Python and forecast**.
   b. And based on EN ratings for those two courses, the recommended courses for LN is **Python** as it has higher ratings form forecast according to LN

3.

```
library(lsa)    # Latent Semantic Analysis

ratings = read.csv("/media/shehata/Data/DEBI/R_Assignment_03/Assignment_03/Dataset/ratings.csv")
ratings_transpose = read.csv("/media/shehata/Data/DEBI/R_Assignment_03/Assignment_03/Dataset/ratings_transposed.csv")

# View(ratings)

# convert NA to 0
# ratings[is.na(ratings)] <- 0
ratings_transpose[is.na(ratings_transpose)] <- 0

# convert dataframe to matrix and drop traget col
ratings_mx <- as.matrix(ratings[, -1])
ratings_tra_mx <- as.matrix(ratings_transpose[, -1])

# Use R to compute the cosine similarity between users.
corr <- cosine(ratings_tra_mx)

# correlation with EN
corr[,4]
```

Cosine similarity between users:

```
          LN        MH        JH        EN        DU        FL        GL        AH        SA        RW        BA        MG        AF        KG        DS
LN 1.0000000 0.5354529 0.4040610 0.7190319 0.4040610 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.2020305 0.0000000 0.0000000 0.7619048
MH 0.5354529 1.0000000 0.7730207 0.2482286 0.7730207 0.6246950 0.6246950 0.6246950 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.3123475
JH 0.4040610 0.7730207 1.0000000 0.3746343 1.0000000 0.7071068 0.7071068 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.4714045
EN 0.7190319 0.2482286 0.3746343 1.0000000 0.3746343 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.8830216
DU 0.4040610 0.7730207 1.0000000 0.3746343 1.0000000 0.7071068 0.7071068 0.7071068 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.4714045
FL 0.0000000 0.6246950 0.7071068 0.0000000 0.7071068 1.0000000 1.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
GL 0.0000000 0.6246950 0.7071068 0.0000000 0.7071068 1.0000000 1.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
AH 0.0000000 0.6246950 0.7071068 0.0000000 0.7071068 1.0000000 1.0000000 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
SA 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.4472136 1.0000000 0.7071068 1.0000000 1.0000000 0.0000000
RW 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.4472136 1.0000000 0.4472136 0.3162278 0.4472136 0.4472136 0.0000000
BA 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.4472136 1.0000000 0.7071068 1.0000000 1.0000000 0.0000000
MG 0.2020305 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.7071068 0.3162278 0.7071068 1.0000000 0.7071068 0.7071068 0.0000000
AF 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.4472136 1.0000000 0.7071068 1.0000000 1.0000000 0.0000000
KG 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000 0.4472136 1.0000000 0.7071068 1.0000000 1.0000000 0.0000000
DS 0.7619048 0.3123475 0.4714045 0.8830216 0.4714045 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 1.0000000
```

4.

```
       LN        MH        JH        EN        DU        FL        GL        AH        SA        RW        BA        MG        AF        KG        DS
0.7190319 0.2482286 0.3746343 1.0000000 0.3746343 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.8830216
```

Based on Cosine similarity between all users and EN, we found that the highest correlation is found between **EN and DS,** but DS has no new courses to recommend to EN as all ratings courses according to DS is shared also with EN, So the next highest correlation is between **LN and EN**, and according to that, LN can recommend two courses to EN which are Python and Forecast, but the highest rating one according to LN is Python, So **Python** is the recommended one to EN.

5.

```
library(recommenderlab)
library(dplyr)

# Convert ratings matrix to real rating matrix which makes it dense
ratings_tra_mx = as(ratings_mx, "realRatingMatrix")

# Create Recommender Model. The parameters are UBCF and Cosine similarity.
# We take 1 nearest neighbours
rec_mod = Recommender(ratings_tra_mx, method = "IBCF", param=list(method="Cosine"))

# Obtain top 3 recommendations for 1st user entry in dataset
Top_3_pred = predict(rec_mod, ratings_tra_mx[4], n=3)

#Convert the recommendations to a list
Top_3_List = as(Top_3_pred, "list")
Top_3_List
```

- After Applying item-based collaborative filtering to the dataset
  The function predicts that the top courses recommended to EN are
  Python and Spatial and Forecast

```
[[1]]
[1] "Forecast" "Spatial"  "Python"
```