PLATFORM USED:

Programming Language : R

Tool Used for Database: Microsoft Excel 2010

Operating System: Windows

Hardware: Laptop

List of some common supermarket items: Filename – Basket_set1

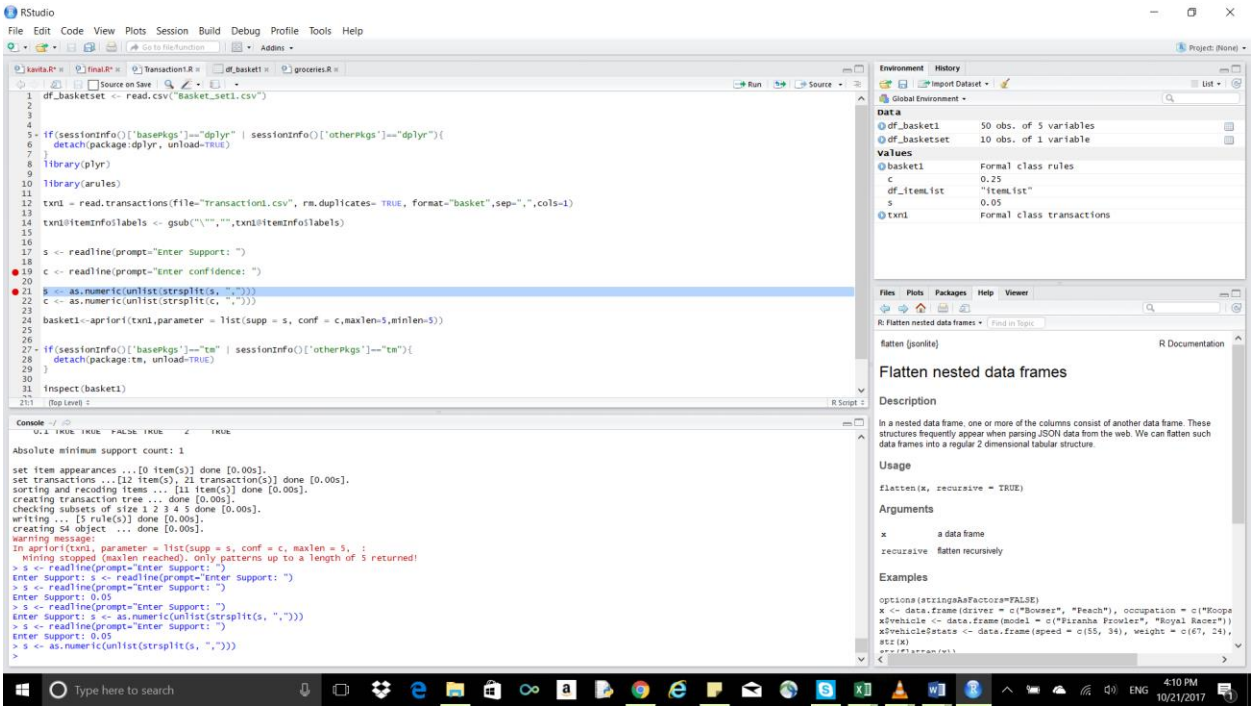
itemDescription
milk
bread
beer
apple
diaper
yogurt
eggs
chocolate
juice
fruits

Database1: Filename- Transaction1

	itemList					
1	milk	bread	beer	apple	diaper	yogurt
2	milk	bread	beer	yogurt	eggs	apple
3	apple	yogurt	beer	chocolate		
4	chocolate	juice	yogurt	fruits		
5	milk	bread	chocolate	beer	yogurt	
6	milk	eggs	bread	beer	fruits	
7	chocolate	eggs	yogurt			
8	milk	chocolate	eggs	fruits		
9	beer	apple	chocolate	fruits		
10	yogurt	milk	fruits	beer		
11	yogurt	juice	bread	milk		
12	beer	fruits	milk	chocolate	juice	

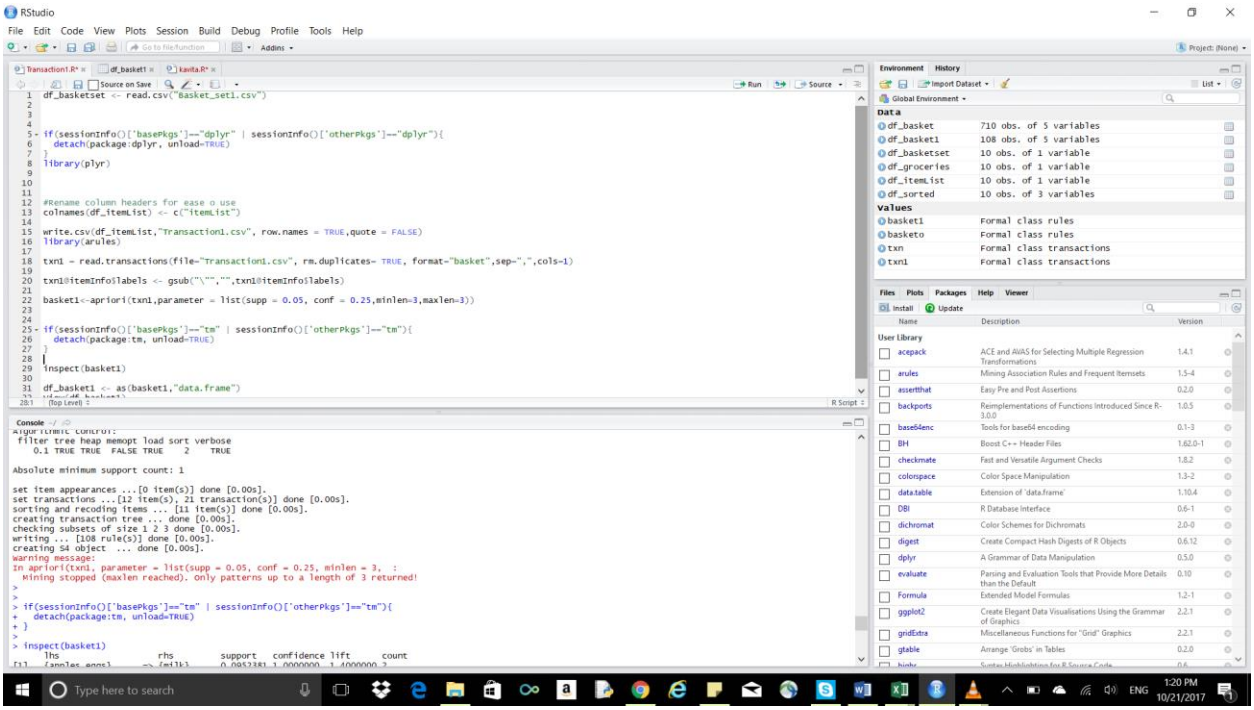
13	milk	bread	beer	diaper		
14	diaper	milk	eggs	apples	chocolate	
15	yogurt	milk	bread	apple	fruits	
16	apple	diaper	chocolate	beer		
17	milk	juice	beer			
18	fruits	juice	eggs	milk	apples	
19	milk	bread	eggs	fruits		
20	milk	eggs	fruits	beer	apple	

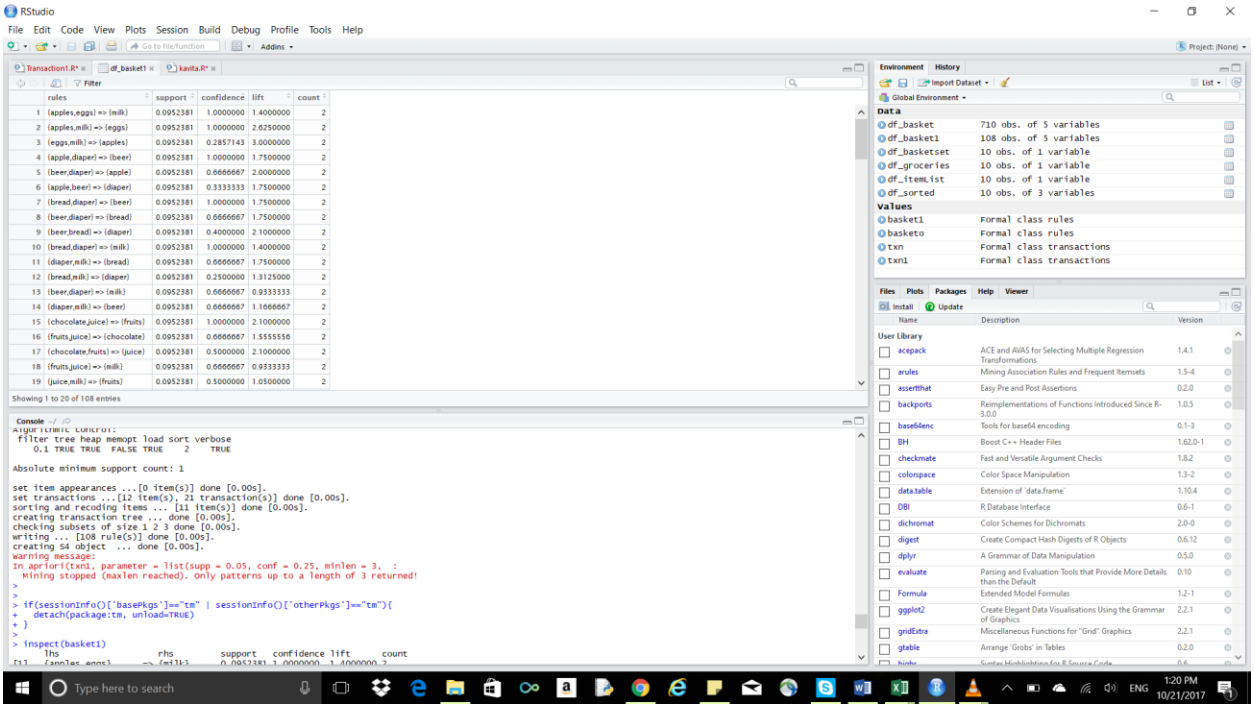
```
df_basketset <- read.csv("Basket_set1.csv")

if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
  detach(package:dplyr, unload=TRUE)
}
library(plyr)

colnames(df_itemList) <- c("itemList")

write.csv(df_itemList,"Transaction1.csv", row.names = TRUE,quote = FALSE)
library(arules)

txn1 = read.transactions(file="Transaction1.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)

txn1@itemInfo$labels <- gsub("\"","",txn1@itemInfo$labels)

s <- readline(prompt="Enter Support: ")
c <- readline(prompt="Enter confidence: ")

s <- as.numeric(unlist(strsplit(s, ",")))
c <- as.numeric(unlist(strsplit(c, ",")))
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
kavita.R* final.R* Transaction1.R df_basket1 groceries.R
Source on Save
Run Source
df_basketset <- read.csv("Basket_set1.csv")
if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
detach(package:dplyr, unload=TRUE)
}
library(plyr)
library(arules)
txn1 = read.transactions(file="Transaction1.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)
txn1@itemInfo$labels <- gsub("\"","",txn1@itemInfo$labels)
s <- readline(prompt="Enter Support: ")
c <- readline(prompt="Enter confidence: ")
s <- as.numeric(unlist(strsplit(s, ",")))
c <- as.numeric(unlist(strsplit(c, ",")))
basket1<-apriori(txn1,parameter = list(supp = s, conf = c,maxlen=5,minlen=5))
if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
detach(package:tm, unload=TRUE)
}
inspect(basket1)
Console
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[12 item(s), 21 transaction(s)] done [0.00s].
sorting and recoding items ... [11 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [5 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
warning message:
In apriori(txn1, parameter = list(supp = s, conf = c, maxlen = 5, :
Mining stopped (maxlen reached). Only patterns up to a length of 5 returned!
> s <- readline(prompt="Enter Support: ")
Enter Support: s <- readline(prompt="Enter Support: ")
> s <- readline(prompt="Enter Support: ")
Enter Support: 0.05
> s <- readline(prompt="Enter Support: ")
Enter Support: s <- as.numeric(unlist(strsplit(s, ",")))
> s <- readline(prompt="Enter Support: ")
Enter Support: 0.05
> s <- as.numeric(unlist(strsplit(s, ",")))
Environment History
Import Dataset
Global Environment
Data
df_basket1 50 obs. of 5 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
c 0.25
df_itemList "ItemList"
s 0.05
txn1 Formal class transactions
Files Plots Packages Help Viewer
R: Flatten nested data frames
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
options(stringsAsFactors=FALSE)
x <- data.frame(driver = c("Bowser", "Peach"), occupation = c("Koopa
x$vehicle <- data.frame(model = c("Piranha Prowler", "Royal Racer"))
x$vehicle$stats <- data.frame(speed = c(55, 34), weight = c(67, 24),
str(x)
Type here to search
ENG
4:10 PM
10/21/2017

Result for Database1 for **frequent -2 itemsets**

```
basket1<-apriori(txn1,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}

inspect(basket1)

df_basket1 <- as(basket1,"data.frame")
View(df_basket1)
```






```
   lhs                    rhs          support    confidence lift      count
[1]   {apples,eggs}       => {milk}       0.0952381 1.0000000  1.4000000 2
[2]   {apples,milk}       => {eggs}       0.0952381 1.0000000  2.6250000 2
[3]   {eggs,milk}         => {apples}     0.0952381 0.2857143  3.0000000 2
[4]   {apple,diaper}      => {beer}       0.0952381 1.0000000  1.7500000 2
[5]   {beer,diaper}       => {apple}      0.0952381 0.6666667  2.0000000 2
[6]   {apple,beer}        => {diaper}     0.0952381 0.3333333  1.7500000 2
[7]   {bread,diaper}      => {beer}       0.0952381 1.0000000  1.7500000 2
[8]   {beer,diaper}       => {bread}      0.0952381 0.6666667  1.7500000 2
[9]   {beer,bread}        => {diaper}     0.0952381 0.4000000  2.1000000 2
[10]  {bread,diaper}      => {milk}       0.0952381 1.0000000  1.4000000 2
[11]  {diaper,milk}       => {bread}      0.0952381 0.6666667  1.7500000 2
[12]  {bread,milk}        => {diaper}     0.0952381 0.2500000  1.3125000 2
[13]  {beer,diaper}       => {milk}       0.0952381 0.6666667  0.9333333 2
[14]  {diaper,milk}       => {beer}       0.0952381 0.6666667  1.1666667 2
[15]  {chocolate,juice}   => {fruits}     0.0952381 1.0000000  2.1000000 2
[16]  {fruits,juice}      => {chocolate}  0.0952381 0.6666667  1.5555556 2
[17]  {chocolate,fruits}  => {juice}      0.0952381 0.5000000  2.1000000 2
[18]  {fruits,juice}      => {milk}       0.0952381 0.6666667  0.9333333 2
[19]  {juice,milk}        => {fruits}     0.0952381 0.5000000  1.0500000 2
[20]  {fruits,milk}       => {juice}      0.0952381 0.2500000  1.0500000 2
[21]  {beer,juice}        => {milk}       0.0952381 1.0000000  1.4000000 2
[22]  {juice,milk}        => {beer}       0.0952381 0.5000000  0.8750000 2
[23]  {apple,eggs}        => {beer}       0.0952381 1.0000000  1.7500000 2
[24]  {apple,beer}        => {eggs}       0.0952381 0.3333333  0.8750000 2
[25]  {beer,eggs}         => {apple}      0.0952381 0.6666667  2.0000000 2
[26]  {apple,eggs}        => {milk}       0.0952381 1.0000000  1.4000000 2
[27]  {apple,milk}        => {eggs}       0.0952381 0.5000000  1.3125000 2
[28]  {eggs,milk}         => {apple}      0.0952381 0.2857143  0.8571429 2
[29]  {apple,chocolate}   => {beer}       0.1428571 1.0000000  1.7500000 3
[30]  {apple,beer}        => {chocolate}  0.1428571 0.5000000  1.1666667 3
[31]  {beer,chocolate}    => {apple}      0.1428571 0.6000000  1.8000000 3
[32]  {apple,bread}       => {yogurt}     0.1428571 1.0000000  2.3333333 3
```

```
[33]  {apple,yogurt}     => {bread}     0.1428571 0.7500000  1.9687500 3
[34]  {bread,yogurt}     => {apple}     0.1428571 0.6000000  1.8000000 3
[35]  {apple,bread}      => {beer}      0.0952381 0.6666667  1.1666667 2
[36]  {apple,beer}       => {bread}     0.0952381 0.3333333  0.8750000 2
[37]  {beer,bread}       => {apple}     0.0952381 0.4000000  1.2000000 2
[38]  {apple,bread}      => {milk}      0.1428571 1.0000000  1.4000000 3
[39]  {apple,milk}       => {bread}     0.1428571 0.7500000  1.9687500 3
[40]  {bread,milk}       => {apple}     0.1428571 0.3750000  1.1250000 3
[41]  {apple,yogurt}     => {beer}      0.1428571 0.7500000  1.3125000 3
[42]  {apple,beer}       => {yogurt}    0.1428571 0.5000000  1.1666667 3
[43]  {beer,yogurt}      => {apple}     0.1428571 0.6000000  1.8000000 3
[44]  {apple,yogurt}     => {milk}      0.1428571 0.7500000  1.0500000 3
[45]  {apple,milk}       => {yogurt}    0.1428571 0.7500000  1.7500000 3
[46]  {milk,yogurt}      => {apple}     0.1428571 0.5000000  1.5000000 3
[47]  {apple,fruits}     => {beer}      0.0952381 0.6666667  1.1666667 2
[48]  {apple,beer}       => {fruits}    0.0952381 0.3333333  0.7000000 2
[49]  {beer,fruits}      => {apple}     0.0952381 0.4000000  1.2000000 2
[50]  {apple,fruits}     => {milk}      0.0952381 0.6666667  0.9333333 2
[51]  {apple,milk}       => {fruits}    0.0952381 0.5000000  1.0500000 2
[52]  {fruits,milk}      => {apple}     0.0952381 0.2500000  0.7500000 2
[53]  {apple,beer}       => {milk}      0.1428571 0.5000000  0.7000000 3
[54]  {apple,milk}       => {beer}      0.1428571 0.7500000  1.3125000 3
[55]  {beer,milk}        => {apple}     0.1428571 0.3333333  1.0000000 3
[56]  {chocolate,eggs}   => {milk}      0.0952381 0.6666667  0.9333333 2
[57]  {eggs,milk}        => {chocolate} 0.0952381 0.2857143  0.6666667 2
[58]  {chocolate,milk}   => {eggs}      0.0952381 0.5000000  1.3125000 2
[59]  {bread,eggs}       => {fruits}    0.0952381 0.6666667  1.4000000 2
[60]  {eggs,fruits}      => {bread}     0.0952381 0.4000000  1.0500000 2
[61]  {bread,fruits}     => {eggs}      0.0952381 0.6666667  1.7500000 2
[62]  {bread,eggs}       => {beer}      0.0952381 0.6666667  1.1666667 2
[63]  {beer,eggs}        => {bread}     0.0952381 0.6666667  1.7500000 2
[64]  {beer,bread}       => {eggs}      0.0952381 0.4000000  1.0500000 2
[65]  {bread,eggs}       => {milk}      0.1428571 1.0000000  1.4000000 3
[66]  {eggs,milk}        => {bread}     0.1428571 0.4285714  1.1250000 3
[67]  {bread,milk}       => {eggs}      0.1428571 0.3750000  0.9843750 3
[68]  {eggs,fruits}      => {beer}      0.0952381 0.4000000  0.7000000 2
[69]  {beer,eggs}        => {fruits}    0.0952381 0.6666667  1.4000000 2
[70]  {beer,fruits}      => {eggs}      0.0952381 0.4000000  1.0500000 2
[71]  {eggs,fruits}      => {milk}      0.2380952 1.0000000  1.4000000 5
[72]  {eggs,milk}        => {fruits}    0.2380952 0.7142857  1.5000000 5
[73]  {fruits,milk}      => {eggs}      0.2380952 0.6250000  1.6406250 5
[74]  {beer,eggs}        => {milk}      0.1428571 1.0000000  1.4000000 3
[75]  {eggs,milk}        => {beer}      0.1428571 0.4285714  0.7500000 3
[76]  {beer,milk}        => {eggs}      0.1428571 0.3333333  0.8750000 3
[77]  {chocolate,yogurt} => {beer}      0.0952381 0.5000000  0.8750000 2
[78]  {beer,chocolate}   => {yogurt}    0.0952381 0.4000000  0.9333333 2
[79]  {beer,yogurt}      => {chocolate} 0.0952381 0.4000000  0.9333333 2
[80]  {chocolate,fruits} => {beer}      0.0952381 0.5000000  0.8750000 2
[81]  {beer,chocolate}   => {fruits}    0.0952381 0.4000000  0.8400000 2
[82]  {beer,fruits}      => {chocolate} 0.0952381 0.4000000  0.9333333 2
[83]  {chocolate,fruits} => {milk}      0.0952381 0.5000000  0.7000000 2
[84]  {chocolate,milk}   => {fruits}    0.0952381 0.5000000  1.0500000 2
[85]  {fruits,milk}      => {chocolate} 0.0952381 0.2500000  0.5833333 2
[86]  {beer,chocolate}   => {milk}      0.0952381 0.4000000  0.5600000 2
[87]  {chocolate,milk}   => {beer}      0.0952381 0.5000000  0.8750000 2
[88]  {bread,yogurt}     => {beer}      0.1428571 0.6000000  1.0500000 3
[89]  {beer,bread}       => {yogurt}    0.1428571 0.6000000  1.4000000 3
```

```
[90]  {beer,yogurt}      => {bread}     0.1428571 0.6000000  1.5750000 3
[91]  {bread,yogurt}     => {milk}      0.2380952 1.0000000  1.4000000 5
[92]  {bread,milk}       => {yogurt}    0.2380952 0.6250000  1.4583333 5
[93]  {milk,yogurt}      => {bread}     0.2380952 0.8333333  2.1875000 5
[94]  {bread,fruits}     => {milk}      0.1428571 1.0000000  1.4000000 3
[95]  {bread,milk}       => {fruits}    0.1428571 0.3750000  0.7875000 3
[96]  {fruits,milk}      => {bread}     0.1428571 0.3750000  0.9843750 3
[97]  {beer,bread}       => {milk}      0.2380952 1.0000000  1.4000000 5
[98]  {bread,milk}       => {beer}      0.2380952 0.6250000  1.0937500 5
[99]  {beer,milk}        => {bread}     0.2380952 0.5555556  1.4583333 5
[100] {fruits,yogurt}    => {milk}      0.0952381 0.6666667  0.9333333 2
[101] {milk,yogurt}      => {fruits}    0.0952381 0.3333333  0.7000000 2
[102] {fruits,milk}      => {yogurt}    0.0952381 0.2500000  0.5833333 2
[103] {beer,yogurt}      => {milk}      0.1904762 0.8000000  1.1200000 4
[104] {milk,yogurt}      => {beer}      0.1904762 0.6666667  1.1666667 4
[105] {beer,milk}        => {yogurt}    0.1904762 0.4444444  1.0370370 4
[106] {beer,fruits}      => {milk}      0.1904762 0.8000000  1.1200000 4
[107] {fruits,milk}      => {beer}      0.1904762 0.5000000  0.8750000 4
[108] {beer,milk}        => {fruits}    0.1904762 0.4444444  0.9333333 4
```

Result for Database1 for **frequent -3 itemsets**

basket1<-apriori(txn1,parameter = list(supp = s, conf = c,minlen=4,maxlen=4))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

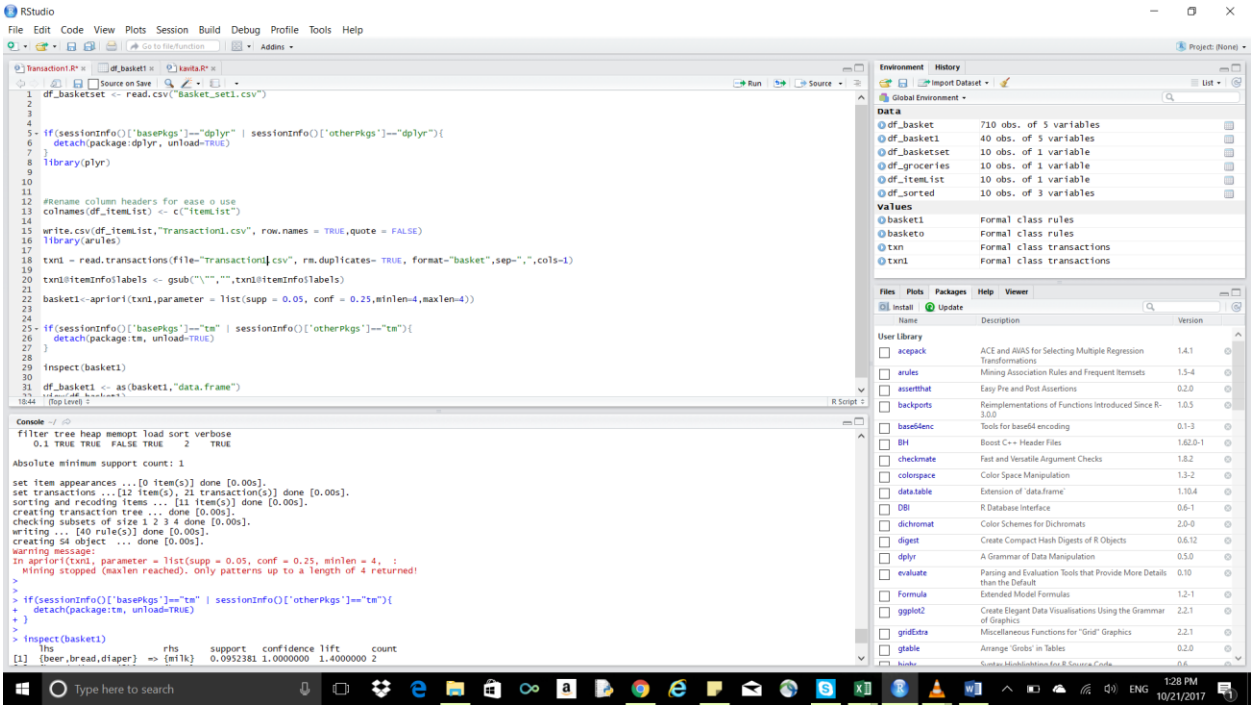
detach(package:tm, unload=TRUE)

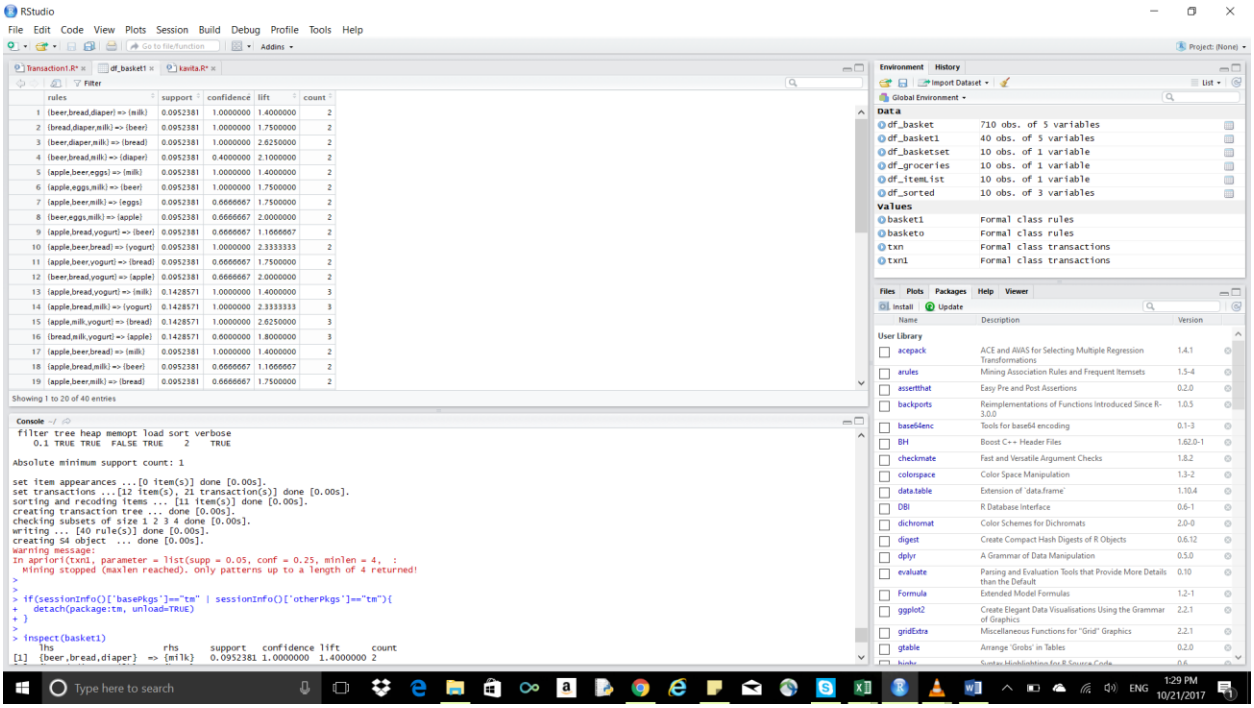
}

inspect(basket1)

df_basket1 <- as(basket1,"data.frame")

View(df_basket1)





	lhs		rhs	support	confidence	lift	count
[1]	{beer,bread,diaper}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[2]	{bread,diaper,milk}	=>	{beer}	0.0952381	1.0000000	1.7500000	2
[3]	{beer,diaper,milk}	=>	{bread}	0.0952381	1.0000000	2.6250000	2
[4]	{beer,bread,milk}	=>	{diaper}	0.0952381	0.4000000	2.1000000	2
[5]	{apple,beer,eggs}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[6]	{apple,eggs,milk}	=>	{beer}	0.0952381	1.0000000	1.7500000	2
[7]	{apple,beer,milk}	=>	{eggs}	0.0952381	0.6666667	1.7500000	2
[8]	{beer,eggs,milk}	=>	{apple}	0.0952381	0.6666667	2.0000000	2
[9]	{apple,bread,yogurt}	=>	{beer}	0.0952381	0.6666667	1.1666667	2
[10]	{apple,beer,bread}	=>	{yogurt}	0.0952381	1.0000000	2.3333333	2
[11]	{apple,beer,yogurt}	=>	{bread}	0.0952381	0.6666667	1.7500000	2
[12]	{beer,bread,yogurt}	=>	{apple}	0.0952381	0.6666667	2.0000000	2
[13]	{apple,bread,yogurt}	=>	{milk}	0.1428571	1.0000000	1.4000000	3
[14]	{apple,bread,milk}	=>	{yogurt}	0.1428571	1.0000000	2.3333333	3
[15]	{apple,milk,yogurt}	=>	{bread}	0.1428571	1.0000000	2.6250000	3
[16]	{bread,milk,yogurt}	=>	{apple}	0.1428571	0.6000000	1.8000000	3
[17]	{apple,beer,bread}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[18]	{apple,bread,milk}	=>	{beer}	0.0952381	0.6666667	1.1666667	2
[19]	{apple,beer,milk}	=>	{bread}	0.0952381	0.6666667	1.7500000	2
[20]	{beer,bread,milk}	=>	{apple}	0.0952381	0.4000000	1.2000000	2
[21]	{apple,beer,yogurt}	=>	{milk}	0.0952381	0.6666667	0.9333333	2
[22]	{apple,milk,yogurt}	=>	{beer}	0.0952381	0.6666667	1.1666667	2
[23]	{apple,beer,milk}	=>	{yogurt}	0.0952381	0.6666667	1.5555556	2
[24]	{beer,milk,yogurt}	=>	{apple}	0.0952381	0.5000000	1.5000000	2
[25]	{bread,eggs,fruits}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[26]	{bread,eggs,milk}	=>	{fruits}	0.0952381	0.6666667	1.4000000	2
[27]	{eggs,fruits,milk}	=>	{bread}	0.0952381	0.4000000	1.0500000	2
[28]	{bread,fruits,milk}	=>	{eggs}	0.0952381	0.6666667	1.7500000	2
[29]	{beer,bread,eggs}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[30]	{bread,eggs,milk}	=>	{beer}	0.0952381	0.6666667	1.1666667	2
[31]	{beer,eggs,milk}	=>	{bread}	0.0952381	0.6666667	1.7500000	2
[32]	{beer,bread,milk}	=>	{eggs}	0.0952381	0.4000000	1.0500000	2

```
[33] {beer,eggs,fruits}   => {milk}   0.0952381 1.0000000  1.4000000 2
[34] {eggs,fruits,milk}   => {beer}   0.0952381 0.4000000  0.7000000 2
[35] {beer,eggs,milk}     => {fruits} 0.0952381 0.6666667  1.4000000 2
[36] {beer,fruits,milk}   => {eggs}   0.0952381 0.5000000  1.3125000 2
[37] {beer,bread,yogurt}  => {milk}   0.1428571 1.0000000  1.4000000 3
[38] {bread,milk,yogurt}  => {beer}   0.1428571 0.6000000  1.0500000 3
[39] {beer,bread,milk}    => {yogurt} 0.1428571 0.6000000  1.4000000 3
[40] {beer,milk,yogurt}   => {bread}  0.1428571 0.7500000  1.9687500 3
```

Result for Database1 for **frequent -4 itemsets**

basket1<-apriori(txn1,parameter = list(supp = s, conf = c,minlen=5,maxlen=5))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

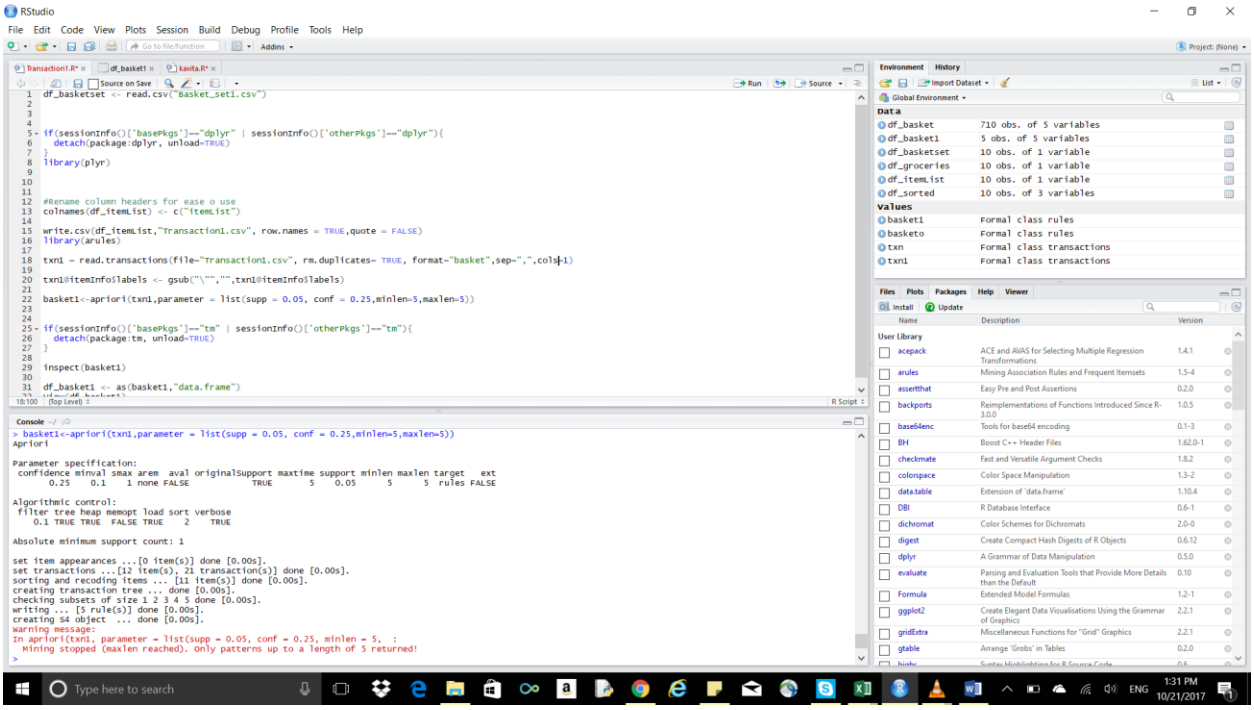
detach(package:tm, unload=TRUE)

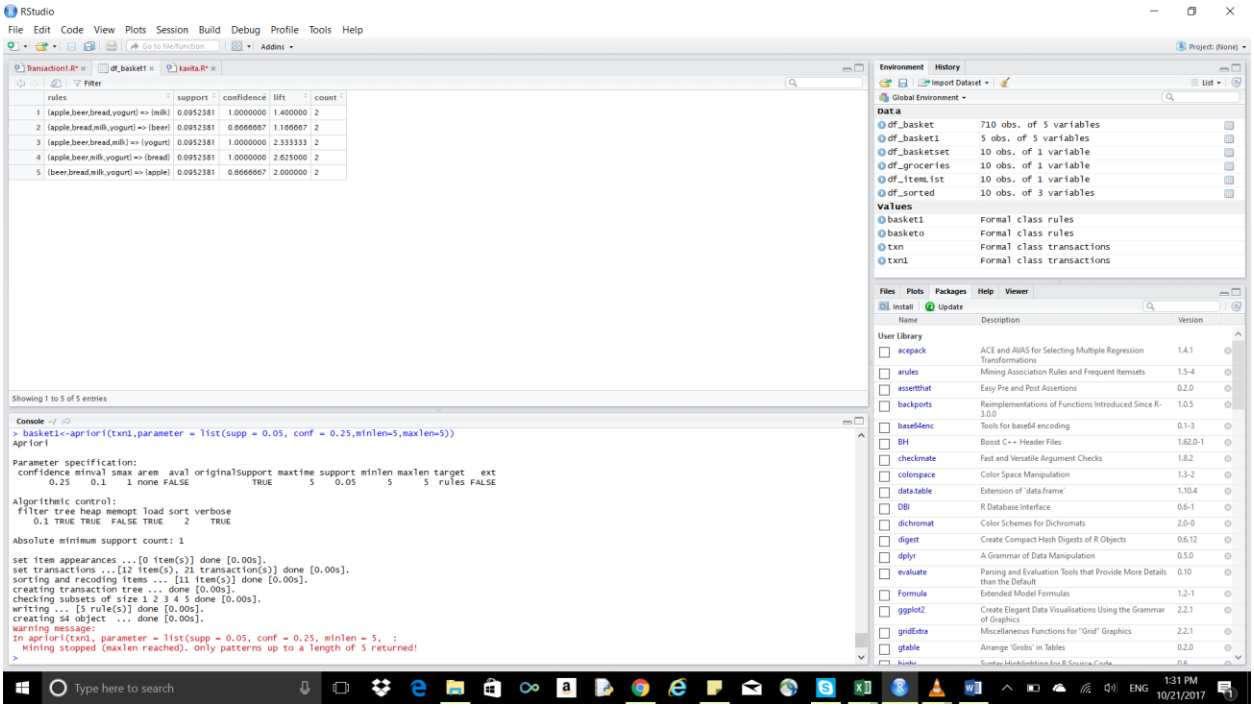
}

inspect(basket1)

df_basket1 <- as(basket1,"data.frame")

View(df_basket1)

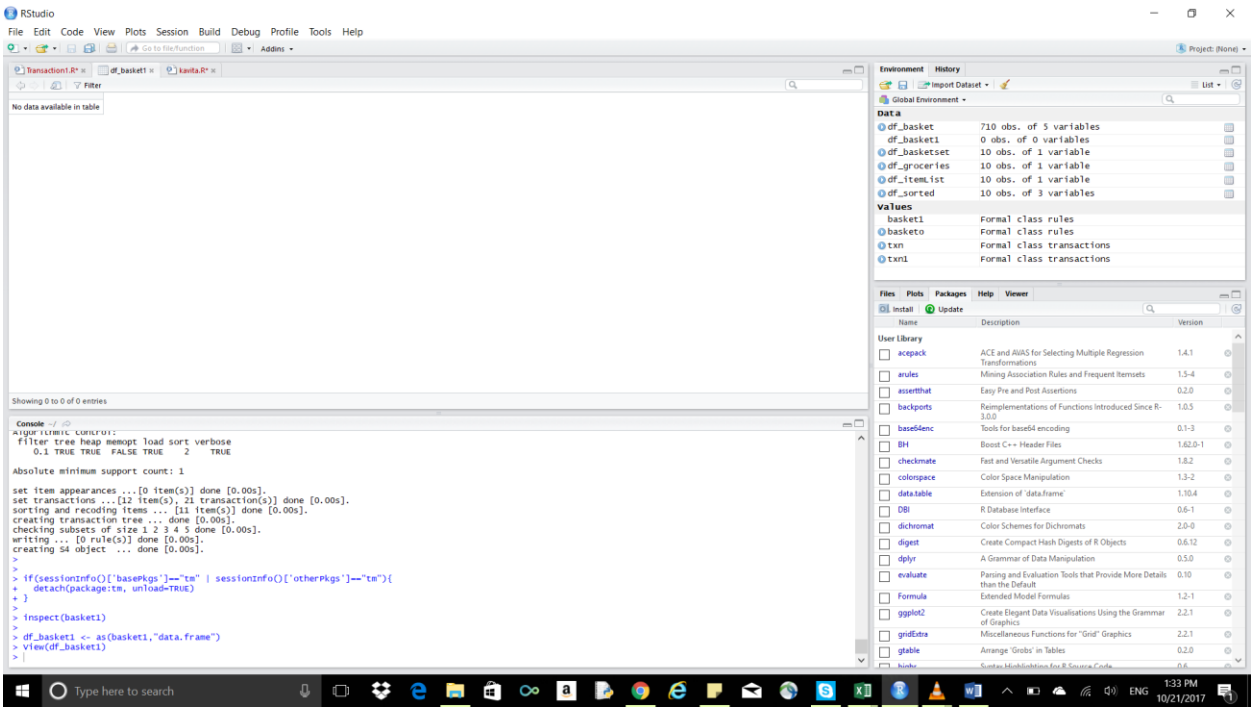





```
   lhs                          rhs       support   confidence lift     count
[1] {apple,beer,bread,yogurt} => {milk}   0.0952381 1.0000000  1.400000 2
[2] {apple,bread,milk,yogurt} => {beer}   0.0952381 0.6666667  1.166667 2
[3] {apple,beer,bread,milk}   => {yogurt} 0.0952381 1.0000000  2.333333 2
[4] {apple,beer,milk,yogurt}  => {bread}  0.0952381 1.0000000  2.625000 2
[5] {beer,bread,milk,yogurt}  => {apple}  0.0952381 0.6666667  2.000000 2
```

Result for Database1 for **frequent -5 itemsets**

```
basket1<-apriori(txn1,parameter = list(supp = s, conf = c,minlen=6,maxlen=6))


if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}


inspect(basket1)


df_basket1 <- as(basket1,"data.frame")
View(df_basket1)
```

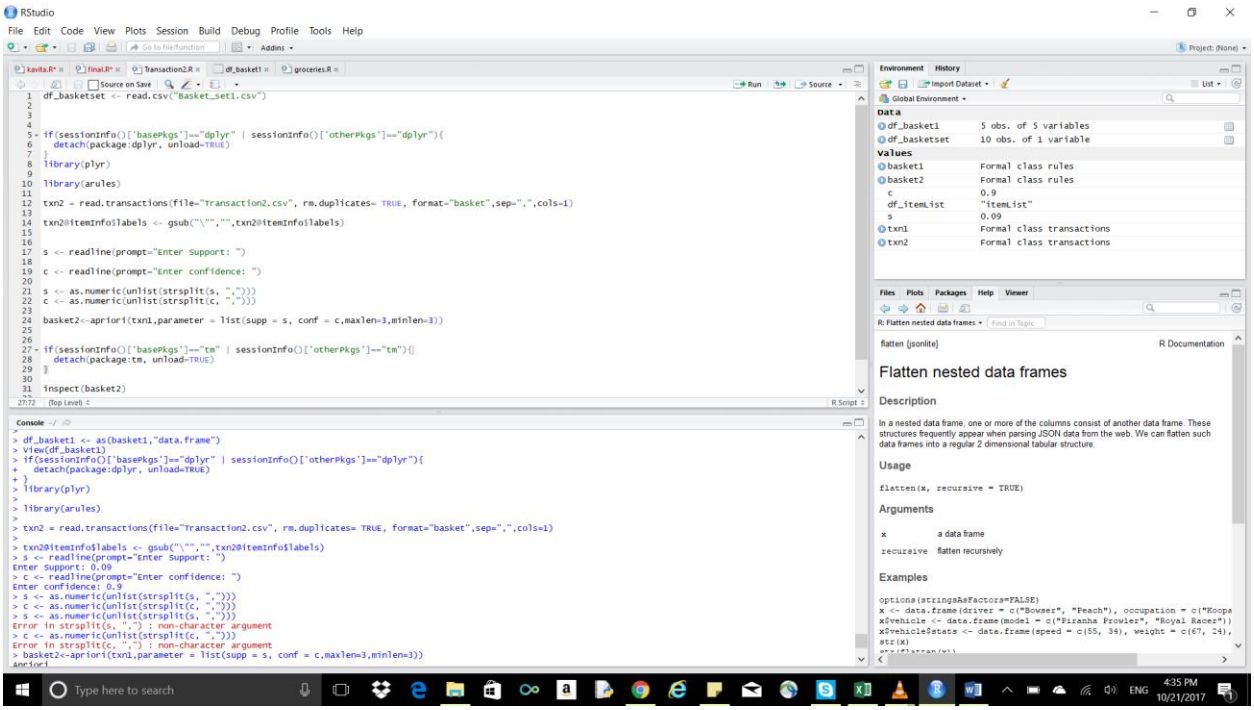


Database2: Filename- **Transaction2**

	itemList					
1	chocolate	milk	juice	apple	diaper	
2	beer	chocolate	yogurt	eggs	apple	
3	apple	eggs	beer	chocolate	diaper	
4	chocolate	juice	yogurt	fruits	milk	
5	milk	bread	chocolate	beer		
6	chocolate	eggs	bread	beer	fruits	
7	chocolate	eggs	yogurt	beer		
8	milk	chocolate	eggs	fruits	juice	
9	beer	apple	chocolate	fruits		
10	chocolate	beer	fruits	diaper		
11	chocolate	juice	bread	milk		
12	beer	fruits	milk	chocolate	juice	
13	milk	bread	beer	diaper		
14	diaper	beer	eggs	apples		
15	diaper	milk	bread	apple	fruits	
16	apple	diaper	chocolate	beer		
17	milk	juice	beer			
18	fruits	juice	beer	milk	apples	
19	milk	bread	diaper	fruits		
20	diaper	eggs	fruits	beer	apple	

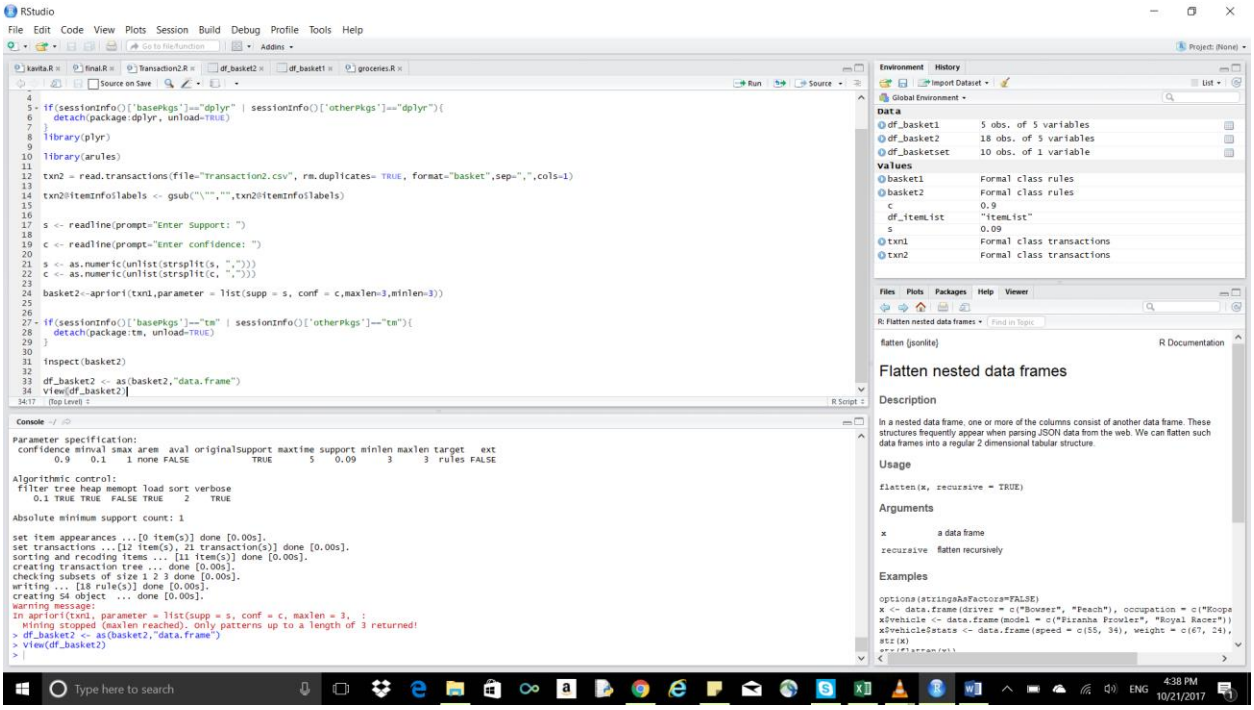
```
df_basketset <- read.csv("Basket_set1.csv")

if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
  detach(package:dplyr, unload=TRUE)
}

library(plyr)


library(arules)


txn2 = read.transactions(file="Transaction2.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)


txn2@itemInfo$labels <- gsub("\"","",txn2@itemInfo$labels)
```

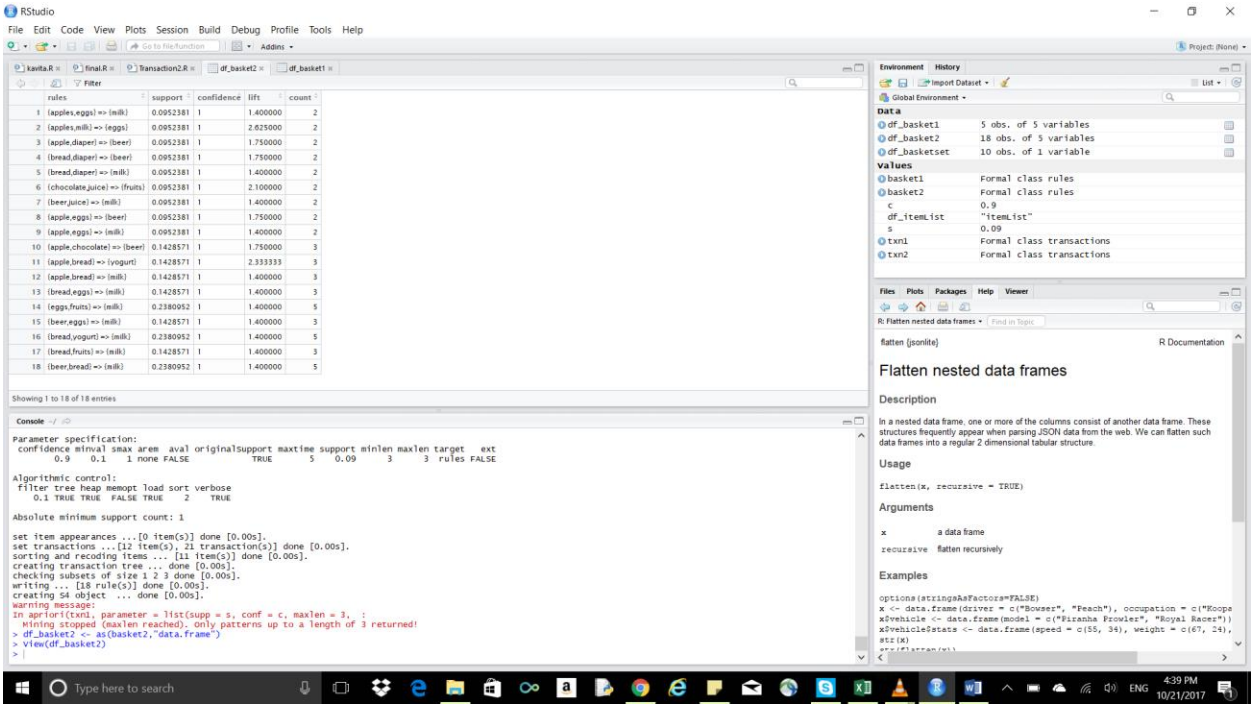
```
s <- readline(prompt="Enter Support: ")

c <- readline(prompt="Enter confidence: ")

s <- as.numeric(unlist(strsplit(s, ",")))

c <- as.numeric(unlist(strsplit(c, ",")))
```

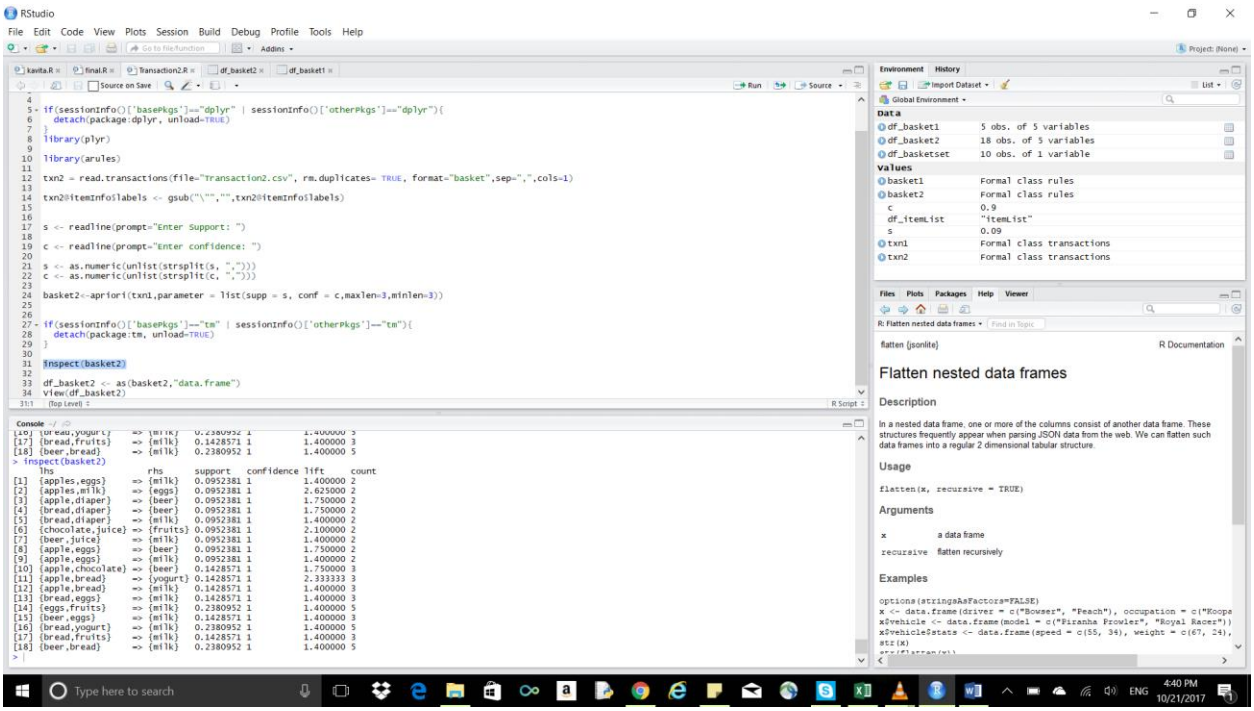


Result for Database2 for **frequent -2 itemsets**

```
basket2<-apriori(txn2,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}


inspect(basket2)

df_basket2 <- as(basket2,"data.frame")

View(df_basket2)
```





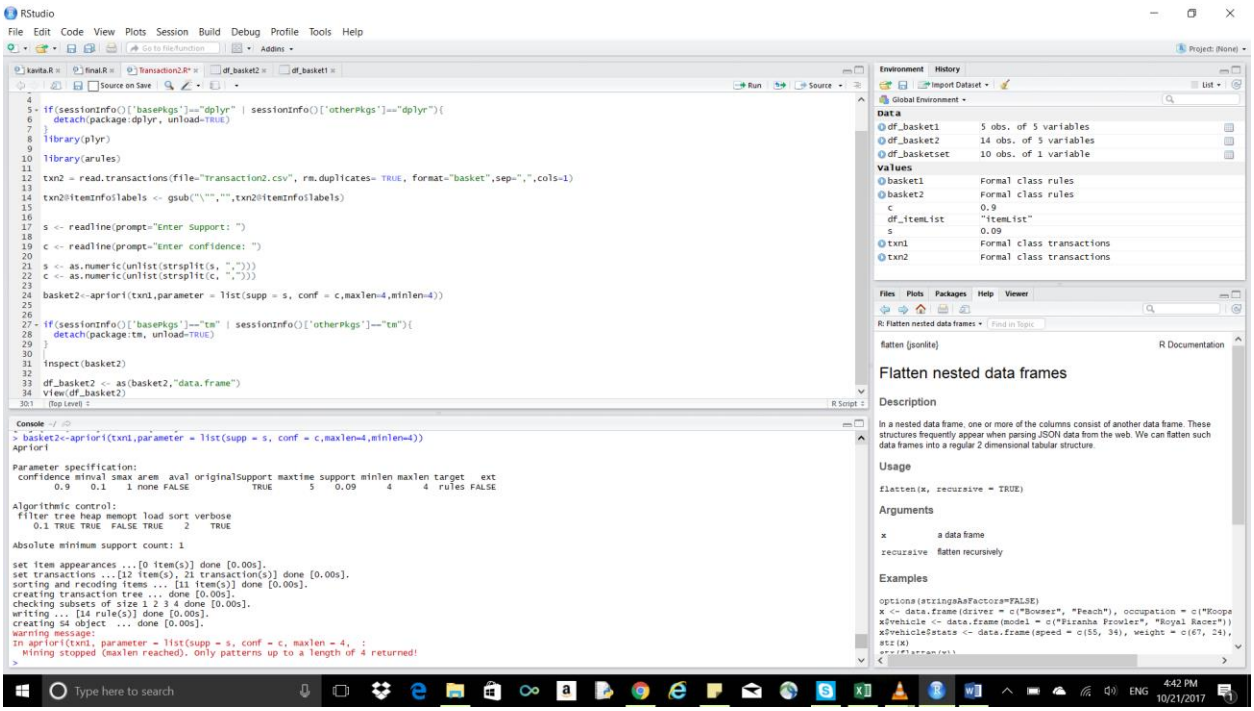


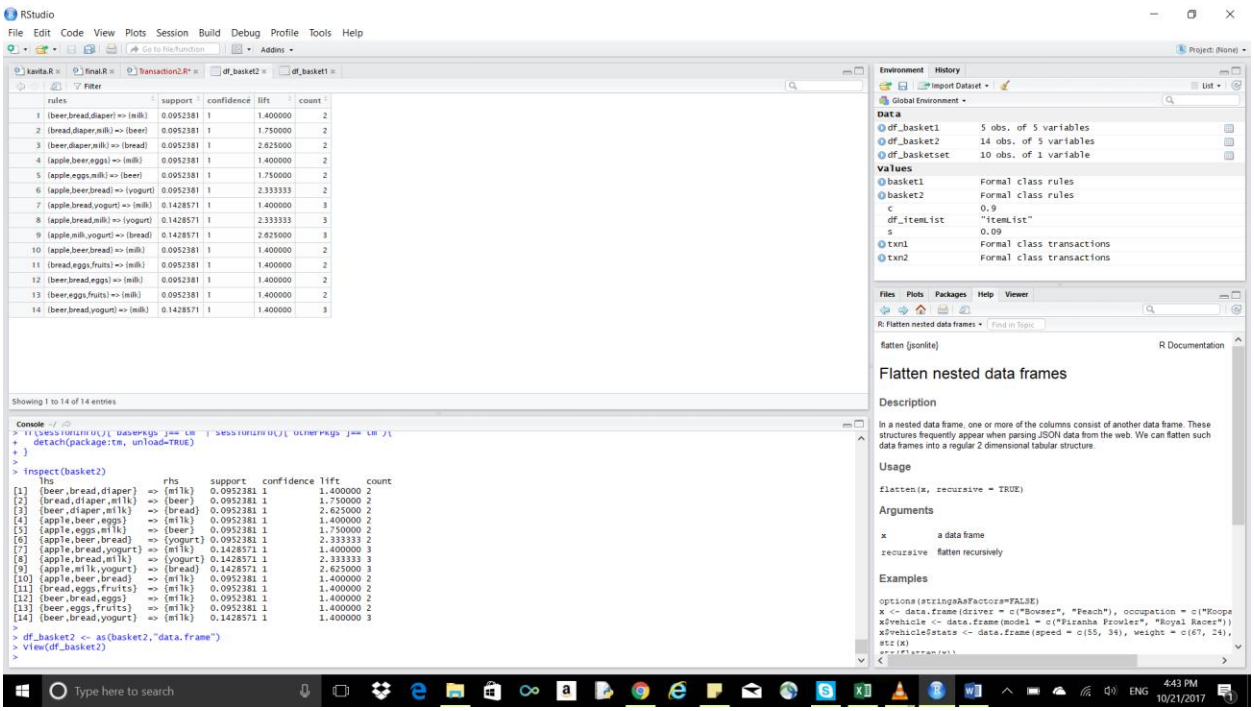
	lhs		rhs	support	confidence	lift	count
[1]	{apples,eggs}	=>	{milk}	0.0952381	1	1.400000	2
[2]	{apples,milk}	=>	{eggs}	0.0952381	1	2.625000	2
[3]	{apple,diaper}	=>	{beer}	0.0952381	1	1.750000	2
[4]	{bread,diaper}	=>	{beer}	0.0952381	1	1.750000	2
[5]	{bread,diaper}	=>	{milk}	0.0952381	1	1.400000	2
[6]	{chocolate,juice}	=>	{fruits}	0.0952381	1	2.100000	2

```
[7]  {beer,juice}       => {milk}   0.0952381 1          1.400000 2
[8]  {apple,eggs}       => {beer}   0.0952381 1          1.750000 2
[9]  {apple,eggs}       => {milk}   0.0952381 1          1.400000 2
[10] {apple,chocolate} => {beer}   0.1428571 1          1.750000 3
[11] {apple,bread}      => {yogurt} 0.1428571 1          2.333333 3
[12] {apple,bread}      => {milk}   0.1428571 1          1.400000 3
[13] {bread,eggs}       => {milk}   0.1428571 1          1.400000 3
[14] {eggs,fruits}      => {milk}   0.2380952 1          1.400000 5
[15] {beer,eggs}        => {milk}   0.1428571 1          1.400000 3
[16] {bread,yogurt}     => {milk}   0.2380952 1          1.400000 5
[17] {bread,fruits}     => {milk}   0.1428571 1          1.400000 3
[18] {beer,bread}       => {milk}   0.2380952 1          1.400000 5
```

Result for Database2 for **frequent -3 itemsets**

```
basket2<-apriori(txn2,parameter = list(supp = s, conf = c,maxlen=4,minlen=4))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}


inspect(basket2)

df_basket2 <- as(basket2,"data.frame")

View(df_basket2)
```

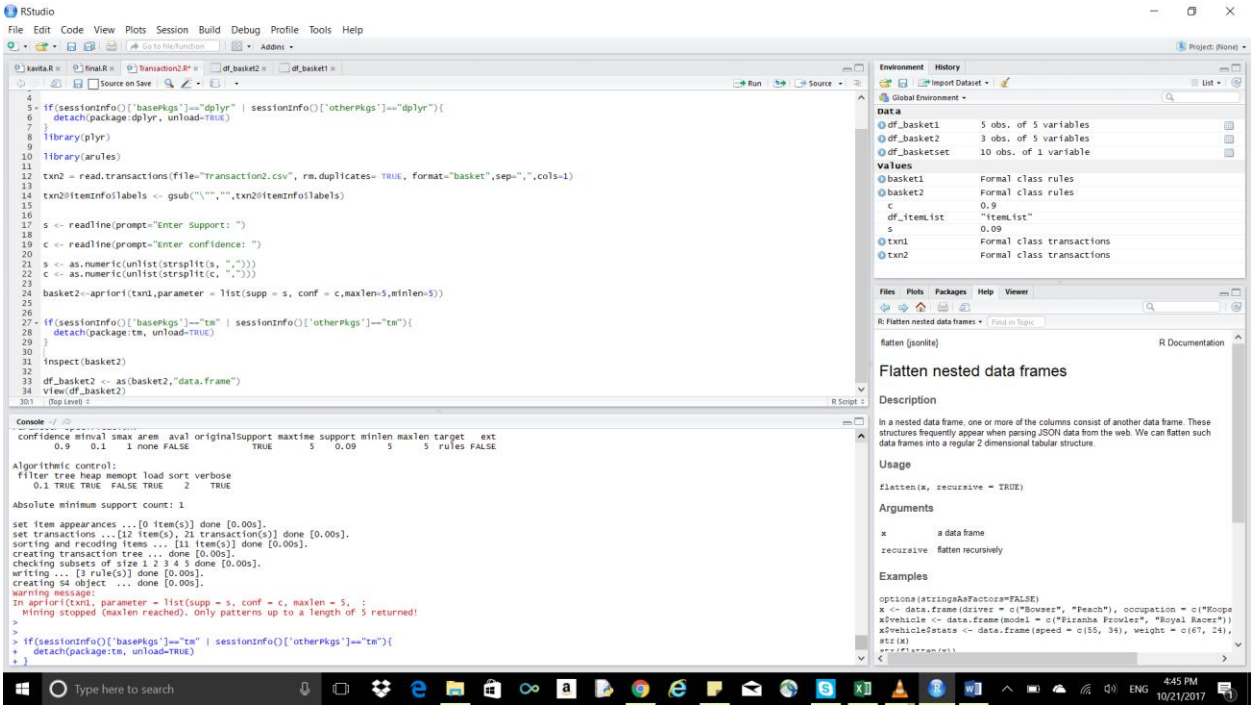


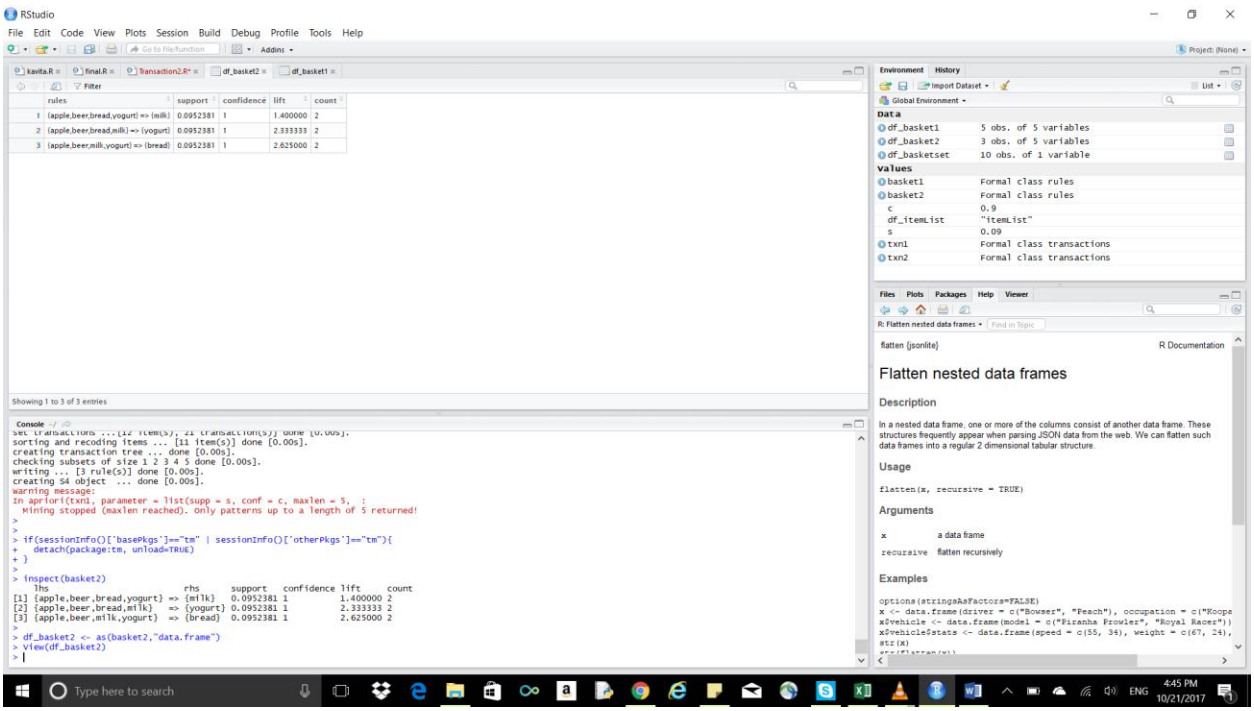



```
   lhs                     rhs      support   confidence lift     count
[1]  {beer,bread,diaper}  => {milk}   0.0952381 1          1.400000 2
[2]  {bread,diaper,milk}  => {beer}   0.0952381 1          1.750000 2
[3]  {beer,diaper,milk}   => {bread}  0.0952381 1          2.625000 2
[4]  {apple,beer,eggs}    => {milk}   0.0952381 1          1.400000 2
[5]  {apple,eggs,milk}    => {beer}   0.0952381 1          1.750000 2
[6]  {apple,beer,bread}   => {yogurt} 0.0952381 1          2.333333 2
[7]  {apple,bread,yogurt} => {milk}   0.1428571 1          1.400000 3
[8]  {apple,bread,milk}   => {yogurt} 0.1428571 1          2.333333 3
[9]  {apple,milk,yogurt}  => {bread}  0.1428571 1          2.625000 3
[10] {apple,beer,bread}   => {milk}   0.0952381 1          1.400000 2
[11] {bread,eggs,fruits}  => {milk}   0.0952381 1          1.400000 2
[12] {beer,bread,eggs}    => {milk}   0.0952381 1          1.400000 2
[13] {beer,eggs,fruits}   => {milk}   0.0952381 1          1.400000 2
[14] {beer,bread,yogurt}  => {milk}   0.1428571 1          1.400000 3
```

Result for Database2 for **frequent -4 itemsets**

```
basket2<-apriori(txn2,parameter = list(supp = s, conf = c,maxlen=5,minlen=5))
if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}

inspect(basket2)
df_basket2 <- as(basket2,"data.frame")
View(df_basket2)
```

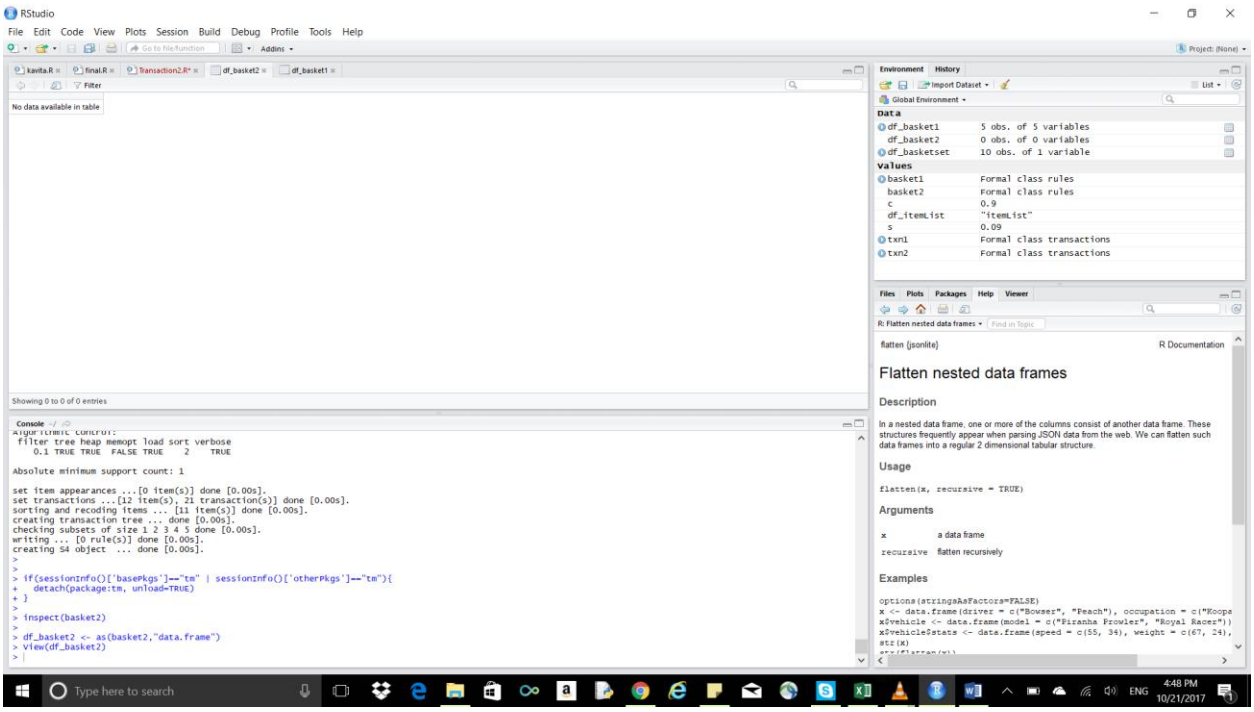




lhs		rhs	support	confidence	lift	count
[1] {apple,beer,bread,yogurt}	=>	{milk}	0.0952381	1	1.400000	2
[2] {apple,beer,bread,milk}	=>	{yogurt}	0.0952381	1	2.333333	2
[3] {apple,beer,milk,yogurt}	=>	{bread}	0.0952381	1	2.625000	2

Result for Database2 for frequent -5 itemsets

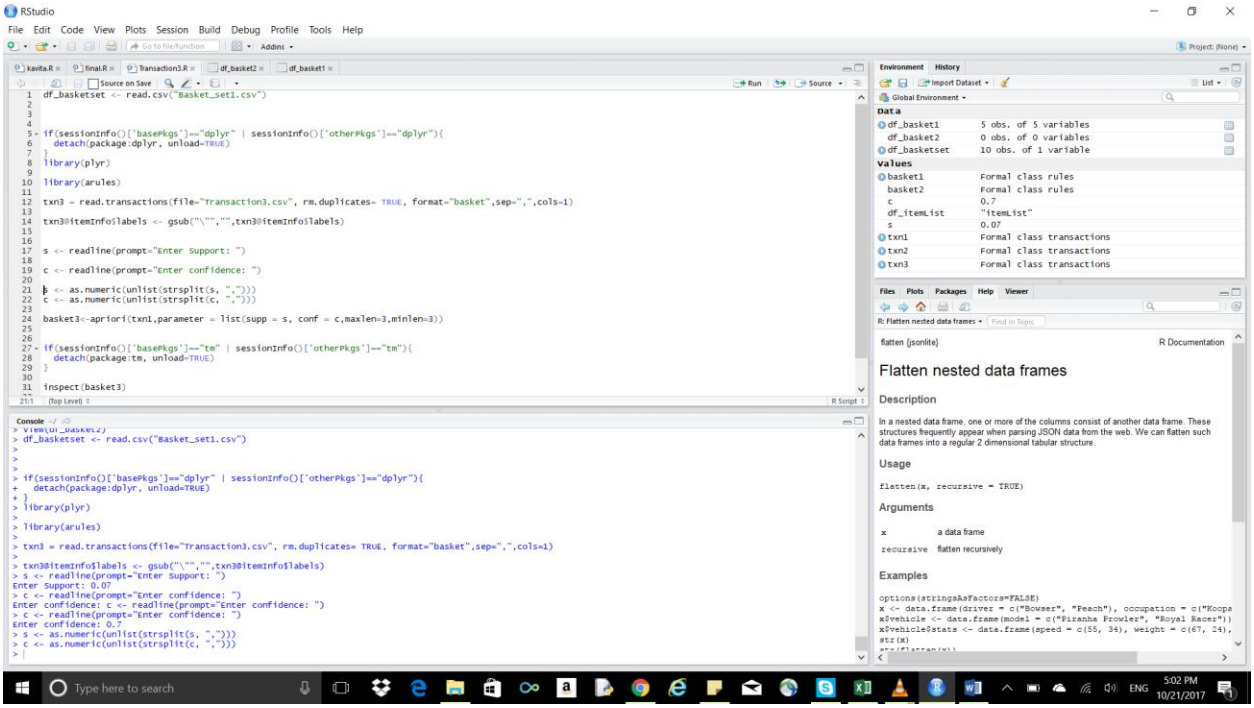
```
basket2<-apriori(txn2,parameter = list(supp = s, conf = c,maxlen=6,minlen=6))
if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}

inspect(basket2)
df_basket2 <- as(basket2,"data.frame")
View(df_basket2)
```



Database3: Filename- **Transaction3**

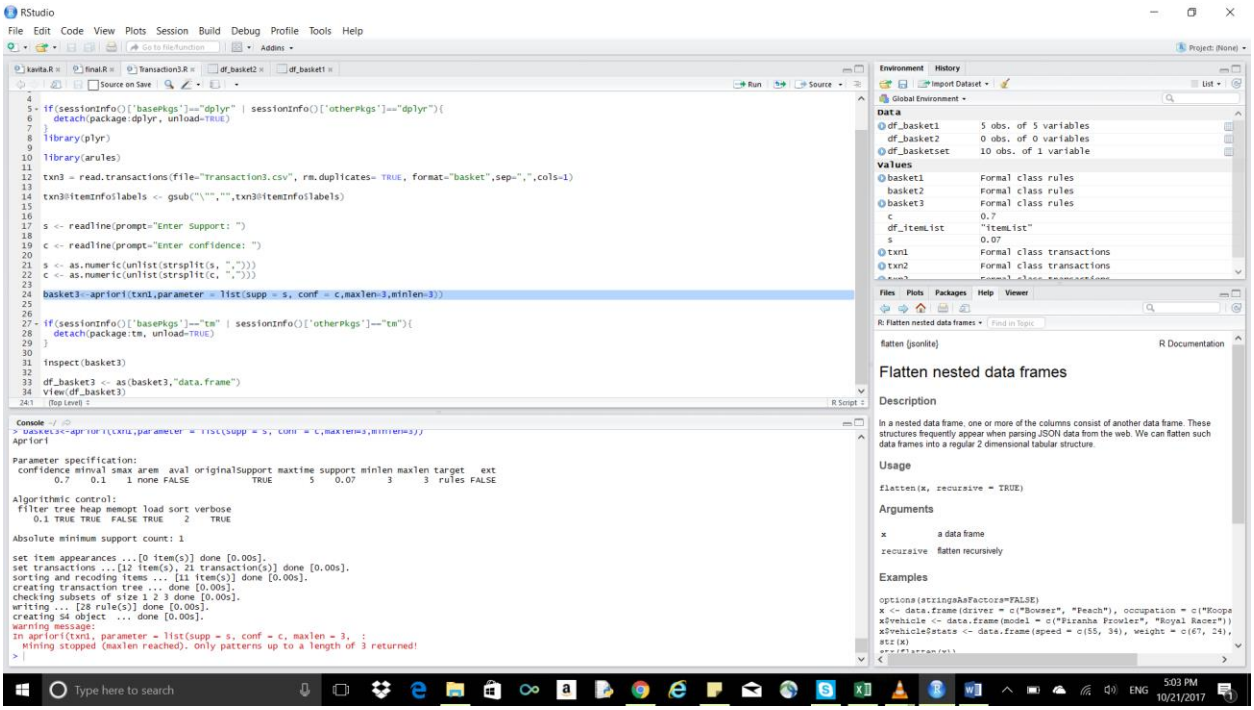
	itemList					
1	chocolate	milk	juice	apple		
2	beer	chocolate	yogurt	eggs		
3	apple	eggs	beer	chocolate		
4	chocolate	juice	yogurt	fruits		
5	milk	bread	chocolate	beer	juice	
6	chocolate	eggs	bread	beer	juice	
7	chocolate	eggs	yogurt	beer	fruits	
8	beer	chocolate	eggs	fruits	yogurt	
9	beer	apple	chocolate	fruits		
10	chocolate	beer	fruits	diaper		
11	chocolate	juice	bread	milk		
12	beer	fruits	milk	chocolate	juice	
13	milk	chocolate	beer	diaper	apple	
14	diaper	beer	chocolate	apple		
15	diaper	milk	bread	apple	juice	
16	apple	diaper	chocolate	beer		
17	milk	juice	chocolate	yogurt	diaper	
18	fruits	juice	beer	milk	eggs	
19	milk	chocolate	diaper	fruits	eggs	
20	diaper	eggs	fruits	beer	juice	

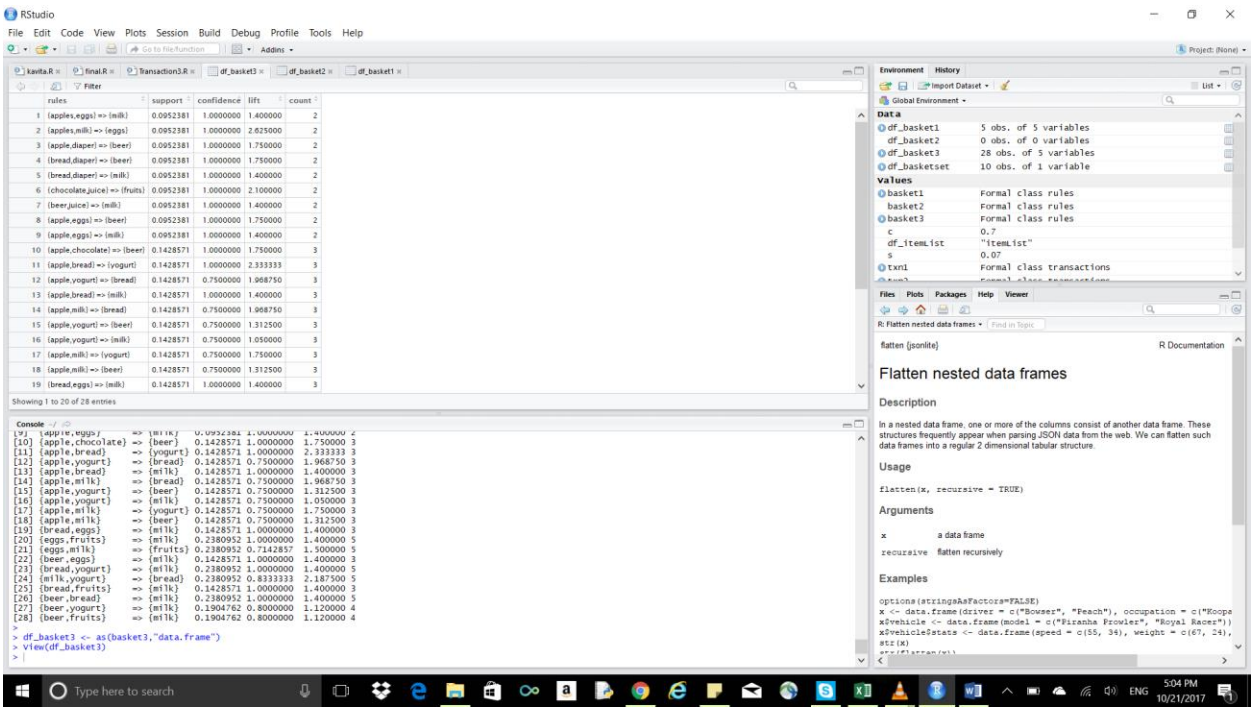
```
df_basketset <- read.csv("Basket_set1.csv")

if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){

  detach(package:dplyr, unload=TRUE)

}

library(plyr)

library(arules)

txn3 = read.transactions(file="Transaction3.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)

txn3@itemInfo$labels <- gsub("\"","",txn3@itemInfo$labels)


s <- readline(prompt="Enter Support: ")

c <- readline(prompt="Enter confidence: ")

s <- as.numeric(unlist(strsplit(s, ",")))

c <- as.numeric(unlist(strsplit(c, ",")))
```


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
kavita.R
final.R
Transaction3.R
df_basket2
df_basket1
1 df_basketset <- read.csv("Basket_set1.csv")
5 if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
6 detach(package:dplyr, unload=TRUE)
7 }
8 library(plyr)
10 library(arules)
12 txn3 = read.transactions(file="Transaction3.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)
14 txn3@itemInfo$labels <- gsub("\"","",txn3@itemInfo$labels)
17 s <- readline(prompt="Enter Support: ")
19 c <- readline(prompt="Enter confidence: ")
21 s <- as.numeric(unlist(strsplit(s, ",")))
22 c <- as.numeric(unlist(strsplit(c, ",")))
24 basket3<-apriori(txn1,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))
27 if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
28 detach(package:tm, unload=TRUE)
29 }
31 inspect(basket3)
Console
> df_basketset <- read.csv("Basket_set1.csv")
> if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
+ detach(package:dplyr, unload=TRUE)
+ }
> library(plyr)
> library(arules)
> txn3 = read.transactions(file="Transaction3.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)
> txn3@itemInfo$labels <- gsub("\"","",txn3@itemInfo$labels)
> s <- readline(prompt="Enter Support: ")
Enter Support: 0.07
> c <- readline(prompt="Enter confidence: ")
Enter confidence: c <- readline(prompt="Enter confidence: ")
> c <- readline(prompt="Enter confidence: ")
Enter confidence: 0.7
> s <- as.numeric(unlist(strsplit(s, ",")))
> c <- as.numeric(unlist(strsplit(c, ",")))
Environment History
Import Dataset
Global Environment
Data
df_basket1 5 obs. of 5 variables
df_basket2 0 obs. of 0 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
basket2 Formal class rules
c 0.7
df_itemList "itemList"
s 0.07
txn1 Formal class transactions
txn2 Formal class transactions
txn3 Formal class transactions
Files Plots Packages Help Viewer
R: Flatten nested data frames
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
options(stringsAsFactors=FALSE)
x <- data.frame(driver = c("Bowser", "Peach"), occupation = c("Koopa
x$vehicle <- data.frame(model = c("Piranha Prowler", "Royal Racer"))
x$vehicle$stats <- data.frame(speed = c(55, 34), weight = c(67, 24),
str(x)
Type here to search
ENG
5:02 PM
10/21/2017

Result for Database3 for **frequent -2 itemsets**

```
basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}

inspect(basket3)

df_basket3 <- as(basket3,"data.frame")

View(df_basket3)
```

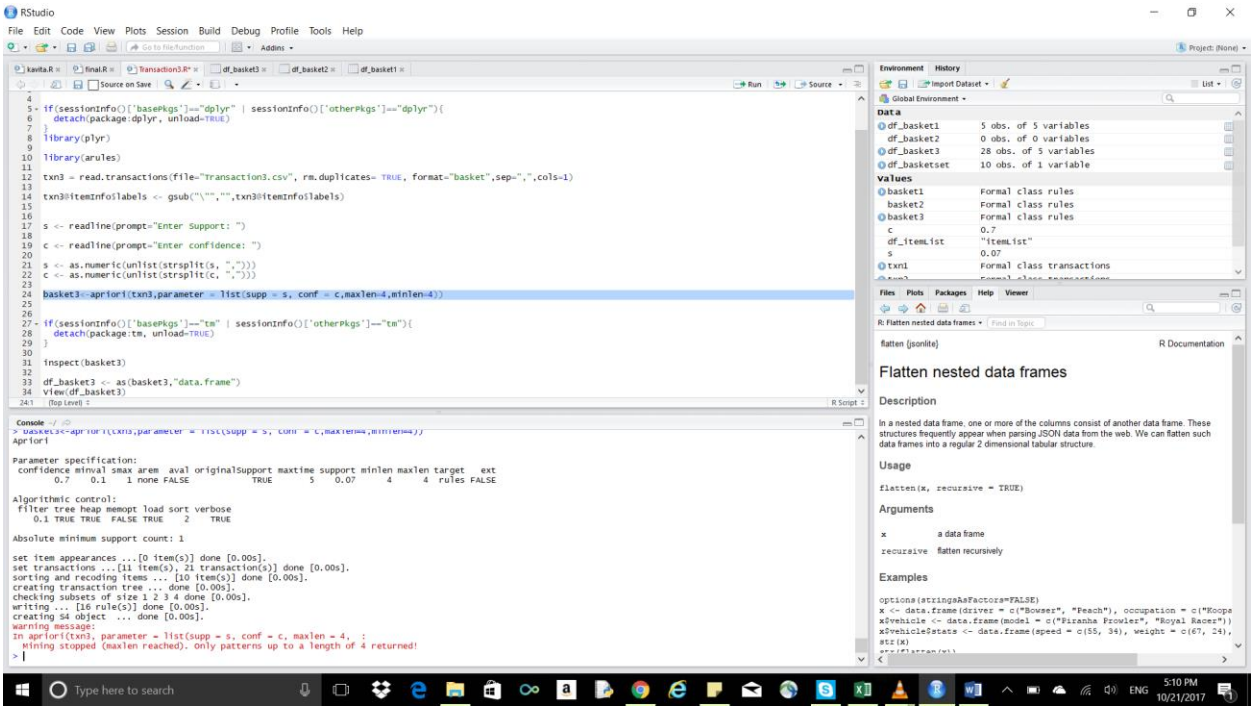


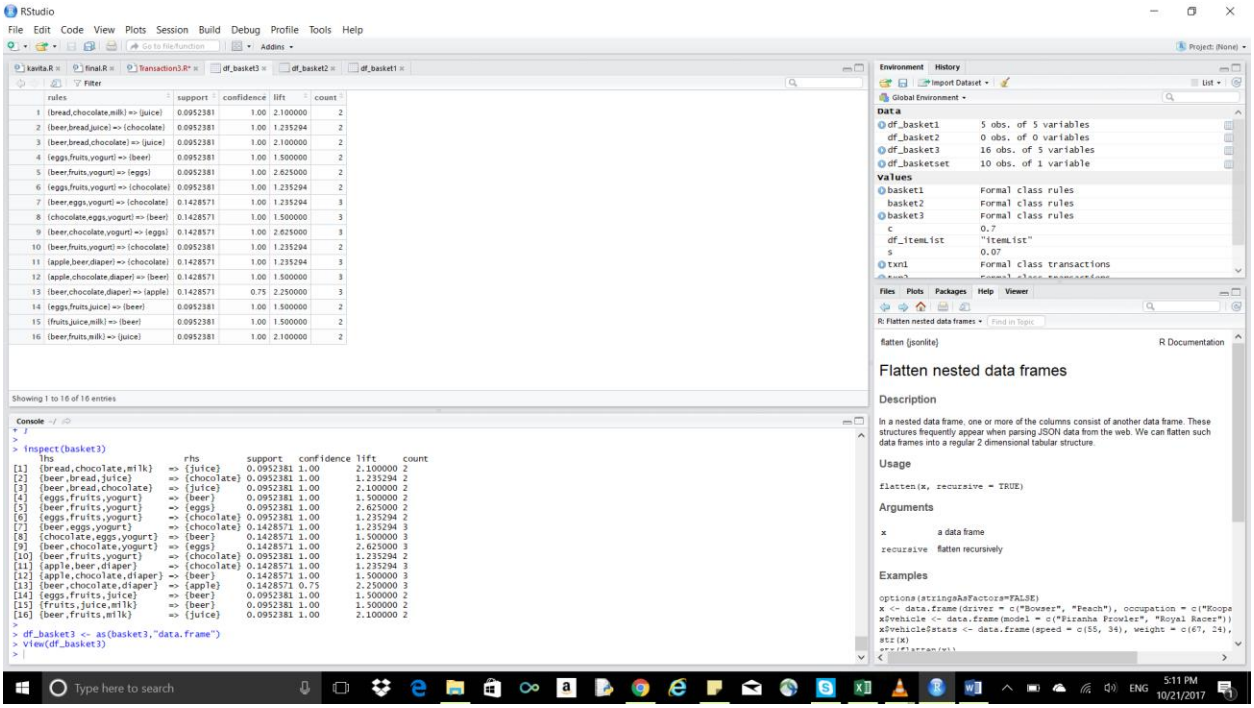


	lhs		rhs	support	confidence	lift	count
[1]	{apples,eggs}	=>	{milk}	0.0952381	1.0000000	1.400000	2
[2]	{apples,milk}	=>	{eggs}	0.0952381	1.0000000	2.625000	2
[3]	{apple,diaper}	=>	{beer}	0.0952381	1.0000000	1.750000	2
[4]	{bread,diaper}	=>	{beer}	0.0952381	1.0000000	1.750000	2
[5]	{bread,diaper}	=>	{milk}	0.0952381	1.0000000	1.400000	2
[6]	{chocolate,juice}	=>	{fruits}	0.0952381	1.0000000	2.100000	2
[7]	{beer,juice}	=>	{milk}	0.0952381	1.0000000	1.400000	2
[8]	{apple,eggs}	=>	{beer}	0.0952381	1.0000000	1.750000	2
[9]	{apple,eggs}	=>	{milk}	0.0952381	1.0000000	1.400000	2
[10]	{apple,chocolate}	=>	{beer}	0.1428571	1.0000000	1.750000	3
[11]	{apple,bread}	=>	{yogurt}	0.1428571	1.0000000	2.333333	3
[12]	{apple,yogurt}	=>	{bread}	0.1428571	0.7500000	1.968750	3
[13]	{apple,bread}	=>	{milk}	0.1428571	1.0000000	1.400000	3
[14]	{apple,milk}	=>	{bread}	0.1428571	0.7500000	1.968750	3
[15]	{apple,yogurt}	=>	{beer}	0.1428571	0.7500000	1.312500	3
[16]	{apple,yogurt}	=>	{milk}	0.1428571	0.7500000	1.050000	3
[17]	{apple,milk}	=>	{yogurt}	0.1428571	0.7500000	1.750000	3
[18]	{apple,milk}	=>	{beer}	0.1428571	0.7500000	1.312500	3
[19]	{bread,eggs}	=>	{milk}	0.1428571	1.0000000	1.400000	3
[20]	{eggs,fruits}	=>	{milk}	0.2380952	1.0000000	1.400000	5
[21]	{eggs,milk}	=>	{fruits}	0.2380952	0.7142857	1.500000	5
[22]	{beer,eggs}	=>	{milk}	0.1428571	1.0000000	1.400000	3
[23]	{bread,yogurt}	=>	{milk}	0.2380952	1.0000000	1.400000	5
[24]	{milk,yogurt}	=>	{bread}	0.2380952	0.8333333	2.187500	5
[25]	{bread,fruits}	=>	{milk}	0.1428571	1.0000000	1.400000	3
[26]	{beer,bread}	=>	{milk}	0.2380952	1.0000000	1.400000	5
[27]	{beer,yogurt}	=>	{milk}	0.1904762	0.8000000	1.120000	4
[28]	{beer,fruits}	=>	{milk}	0.1904762	0.8000000	1.120000	4

Result for Database3 for **frequent -3 itemsets**

```
basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=4,minlen=4))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}

inspect(basket3)

df_basket3 <- as(basket3,"data.frame")

View(df_basket3)
```

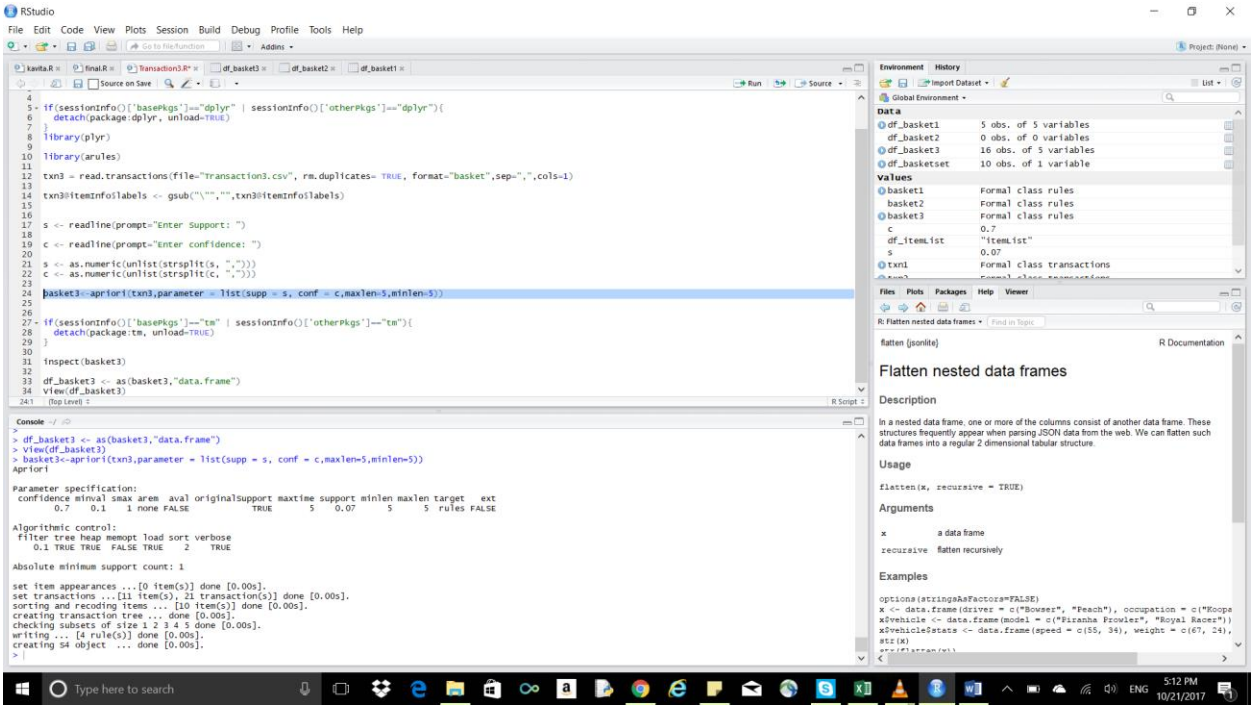


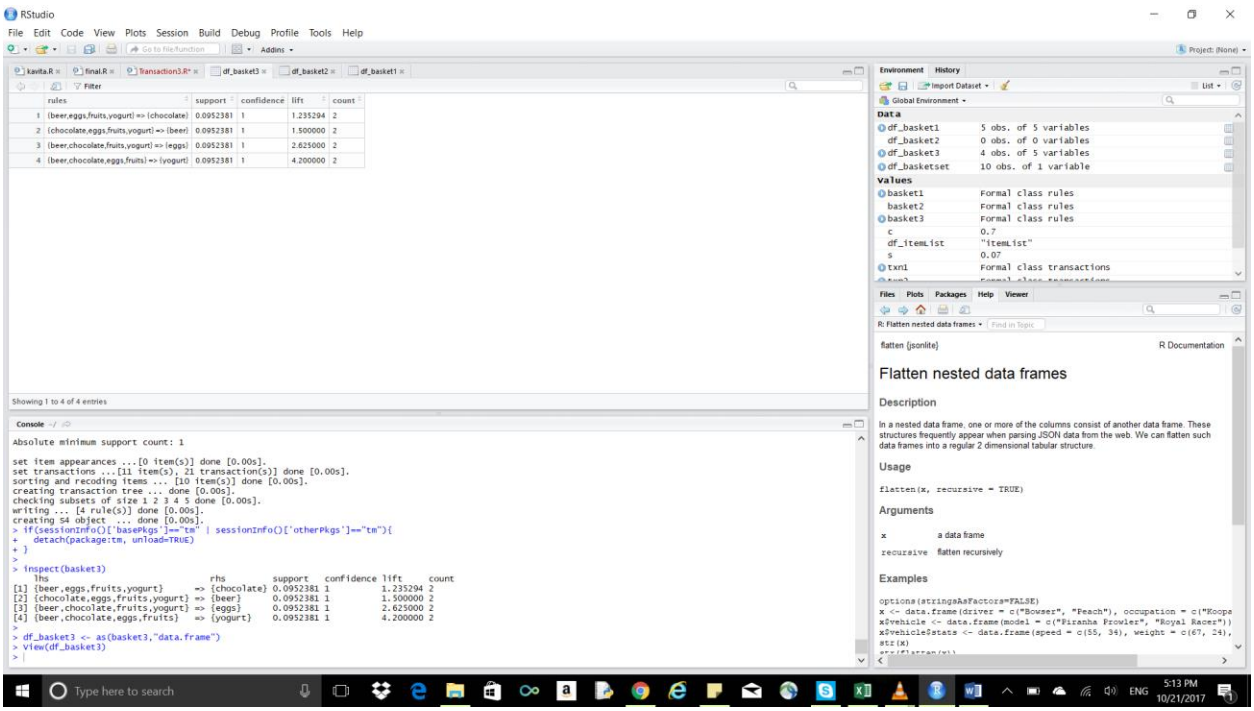


	lhs		rhs	support	confidence	lift	count
[1]	{bread,chocolate,milk}	=>	{juice}	0.0952381	1.00	2.100000	2
[2]	{beer,bread,juice}	=>	{chocolate}	0.0952381	1.00	1.235294	2
[3]	{beer,bread,chocolate}	=>	{juice}	0.0952381	1.00	2.100000	2
[4]	{eggs,fruits,yogurt}	=>	{beer}	0.0952381	1.00	1.500000	2
[5]	{beer,fruits,yogurt}	=>	{eggs}	0.0952381	1.00	2.625000	2
[6]	{eggs,fruits,yogurt}	=>	{chocolate}	0.0952381	1.00	1.235294	2
[7]	{beer,eggs,yogurt}	=>	{chocolate}	0.1428571	1.00	1.235294	3
[8]	{chocolate,eggs,yogurt}	=>	{beer}	0.1428571	1.00	1.500000	3
[9]	{beer,chocolate,yogurt}	=>	{eggs}	0.1428571	1.00	2.625000	3
[10]	{beer,fruits,yogurt}	=>	{chocolate}	0.0952381	1.00	1.235294	2
[11]	{apple,beer,diaper}	=>	{chocolate}	0.1428571	1.00	1.235294	3
[12]	{apple,chocolate,diaper}	=>	{beer}	0.1428571	1.00	1.500000	3
[13]	{beer,chocolate,diaper}	=>	{apple}	0.1428571	0.75	2.250000	3
[14]	{eggs,fruits,juice}	=>	{beer}	0.0952381	1.00	1.500000	2
[15]	{fruits,juice,milk}	=>	{beer}	0.0952381	1.00	1.500000	2
[16]	{beer,fruits,milk}	=>	{juice}	0.0952381	1.00	2.100000	2

Result for Database3 for **frequent -4 itemsets**

```
basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=5,minlen=5))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}

inspect(basket3)

df_basket3 <- as(basket3,"data.frame")

View(df_basket3)
```






```
    lhs                                rhs          support   confidence lift
count
[1] {beer,eggs,fruits,yogurt}       => {chocolate} 0.0952381 1           1.2352
94 2
[2] {chocolate,eggs,fruits,yogurt} => {beer}       0.0952381 1           1.5000
00 2
[3] {beer,chocolate,fruits,yogurt} => {eggs}       0.0952381 1           2.6250
00 2
[4] {beer,chocolate,eggs,fruits}    => {yogurt}    0.0952381 1           4.2000
00 2
```

Result for Database3 for frequent -5 itemsets

basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=6,minlen=6))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

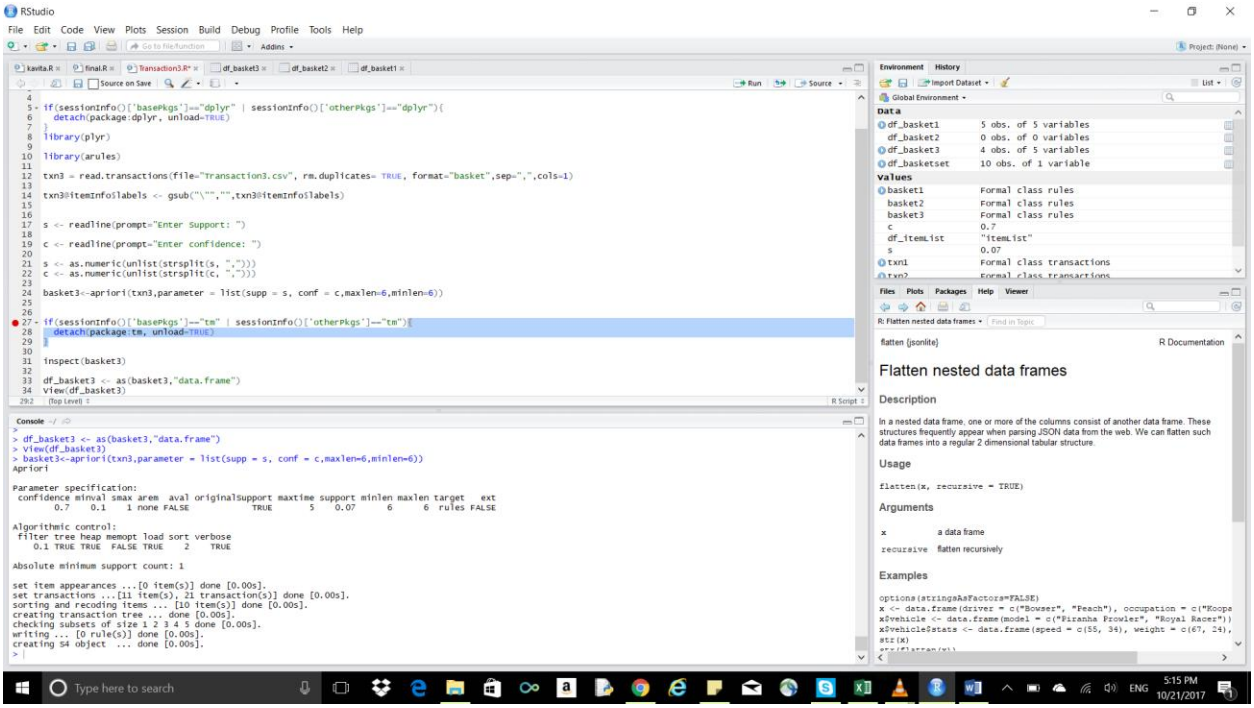
detach(package:tm, unload=TRUE)

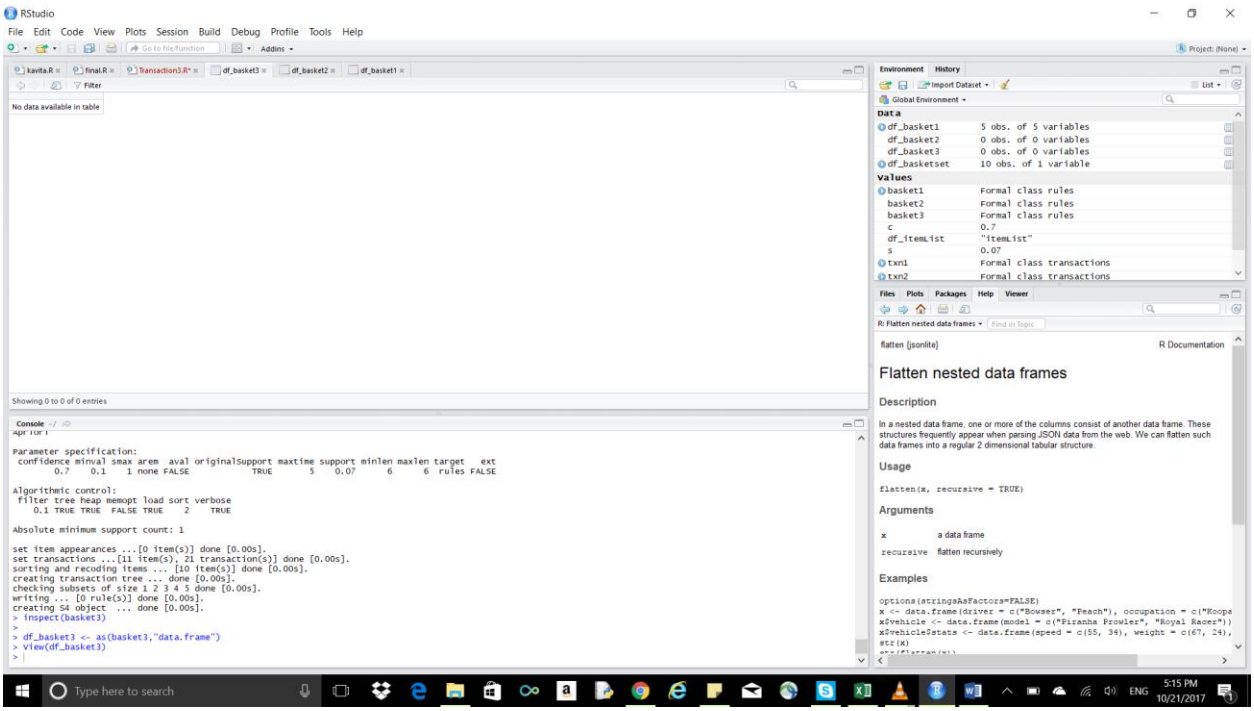
}

inspect(basket3)

df_basket3 <- as(basket3,"data.frame")

View(df_basket3)

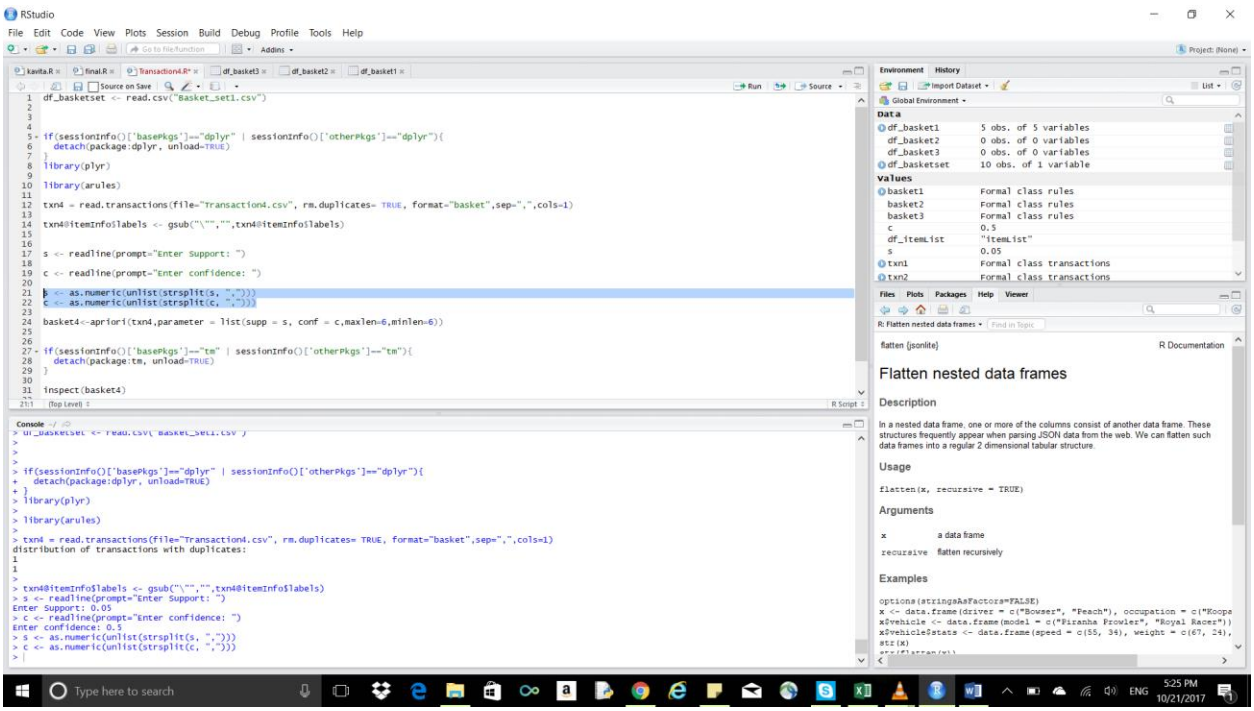

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
kavita.R
final.R
Transaction3.R*
df_basket3
df_basket2
df_basket1
Source on Save
Run
Source
if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
detach(package:dplyr, unload=TRUE)
}
library(plyr)
library(arules)
txn3 = read.transactions(file="Transaction3.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)
txn3@itemInfo$labels <- gsub("\"","",txn3@itemInfo$labels)
s <- readline(prompt="Enter Support: ")
c <- readline(prompt="Enter confidence: ")
s <- as.numeric(unlist(strsplit(s, ",")))
c <- as.numeric(unlist(strsplit(c, ",")))
basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=6,minlen=6))
if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
detach(package:tm, unload=TRUE)
}
inspect(basket3)
df_basket3 <- as(basket3,"data.frame")
View(df_basket3)
Console
> df_basket3 <- as(basket3,"data.frame")
> View(df_basket3)
> basket3<-apriori(txn3,parameter = list(supp = s, conf = c,maxlen=6,minlen=6))
Apriori
Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext
0.7 0.1 1 none FALSE TRUE 5 0.07 6 6 rules FALSE
Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 21 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
Environment History
Import Dataset
Global Environment
Data
df_basket1 5 obs. of 5 variables
df_basket2 0 obs. of 0 variables
df_basket3 4 obs. of 5 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
basket2 Formal class rules
basket3 Formal class rules
c 0.7
df_itemList "itemList"
s 0.07
txn1 Formal class transactions
txn2 Formal class transactions
Files Plots Packages Help Viewer
R: Flatten nested data frames
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
options(stringsAsFactors=FALSE)
x <- data.frame(driver = c("Bowser", "Peach"), occupation = c("Koopa
x$vehicle <- data.frame(model = c("Piranha Prowler", "Royal Racer"))
x$vehicle$stats <- data.frame(speed = c(55, 34), weight = c(67, 24),
str(x)
Type here to search
ENG
5:15 PM
10/21/2017


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
No data available in table
Showing 0 to 0 of 0 entries
Apriori
Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target ext
0.7 0.1 1 none FALSE TRUE 5 0.07 6 6 rules FALSE
Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 21 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> inspect(basket3)
> df_basket3 <- as(basket3,"data.frame")
> View(df_basket3)
Environment History
Import Dataset
Global Environment
Data
df_basket1 5 obs. of 5 variables
df_basket2 0 obs. of 0 variables
df_basket3 0 obs. of 0 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
basket2 Formal class rules
basket3 Formal class rules
c 0.7
df_itemList "itemList"
s 0.07
txn1 Formal class transactions
txn2 Formal class transactions
Files Plots Packages Help Viewer
R: Flatten nested data frames
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
options(stringsAsFactors=FALSE)
x <- data.frame(driver = c("Bowser", "Peach"), occupation = c("Koopa
x$vehicle <- data.frame(model = c("Piranha Prowler", "Royal Racer"))
x$vehicle$stats <- data.frame(speed = c(55, 34), weight = c(67, 24),
str(x)
Type here to search
ENG
5:15 PM
10/21/2017

Database4: Filename- **Transaction4**

	itemList				
1	chocolate	milk	juice	apple	bread
2	beer	bread	yogurt	eggs	milk
3	apple	milk	beer	chocolate	juice
4	chocolate	milk	yogurt	fruits	
5	milk	diaper	chocolate	beer	
6	chocolate	diaper	bread	beer	
7	chocolate	diaper	yogurt	beer	
8	beer	milk	eggs	fruits	
9	beer	milk	chocolate	fruits	
10	chocolate	diaper	fruits	diaper	
11	chocolate	diaper	bread	milk	
12	beer	fruits	milk	chocolate	
13	milk	fruits	beer	diaper	
14	diaper	milk	chocolate	apple	
15	diaper	beer	bread	apple	
16	apple	fruits	chocolate	beer	
17	milk	beer	chocolate	yogurt	
18	fruits	diaper	beer	milk	
19	milk	beer	diaper	fruits	
20	diaper	milk	fruits	beer	

```
df_basketset <- read.csv("Basket_set1.csv")
if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
  detach(package:dplyr, unload=TRUE)
}
library(plyr)
library(arules)
txn4 = read.transactions(file="Transaction4.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)
txn4@itemInfo$labels <- gsub("\"","",txn4@itemInfo$labels)
s <- readline(prompt="Enter Support: ")
c <- readline(prompt="Enter confidence: ")
s <- as.numeric(unlist(strsplit(s, ",")))
c <- as.numeric(unlist(strsplit(c, ",")))
```



Result for Database4 for **frequent -2 itemsets**

basket4<-apriori(txn4,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

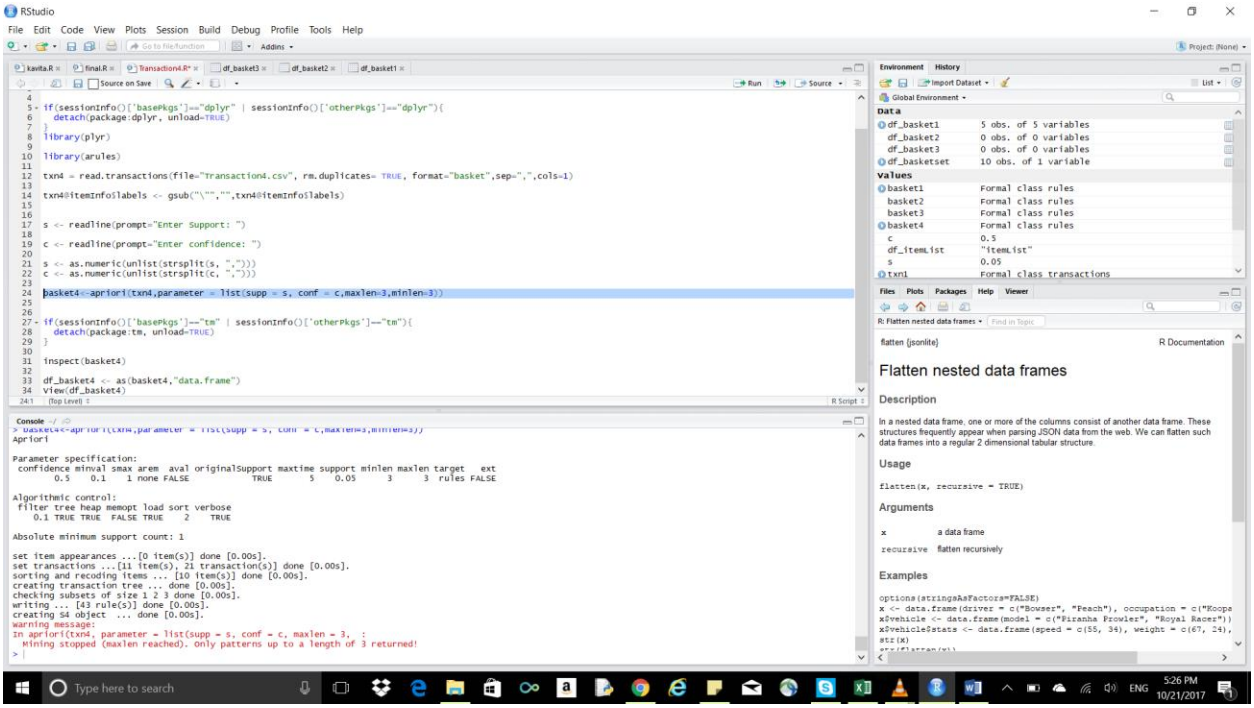
detach(package:tm, unload=TRUE)

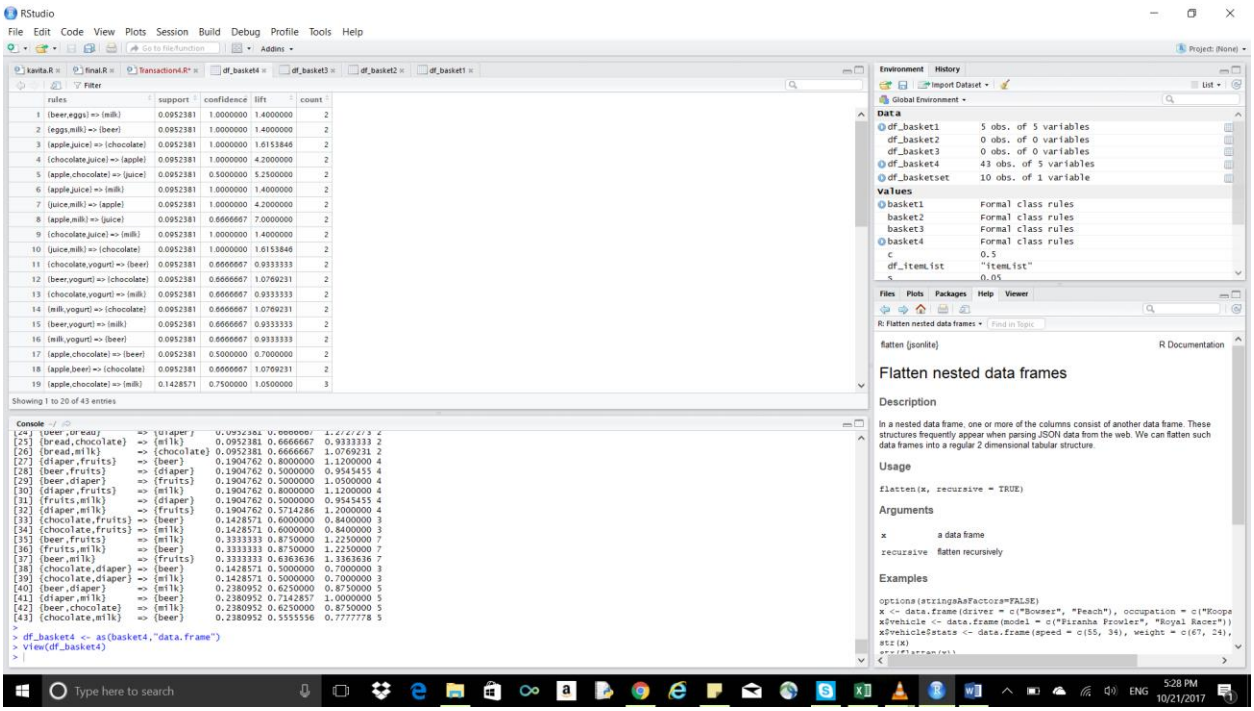
}

inspect(basket4)

df_basket4 <- as(basket4,"data.frame")

View(df_basket4)



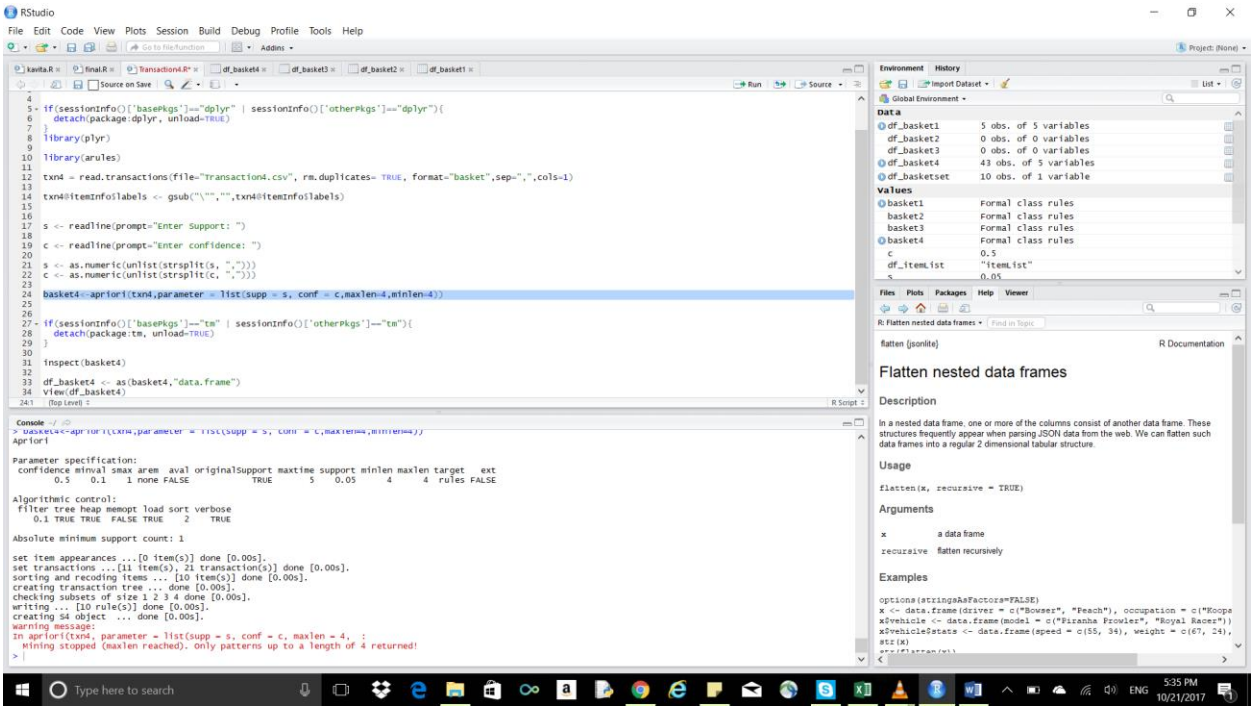


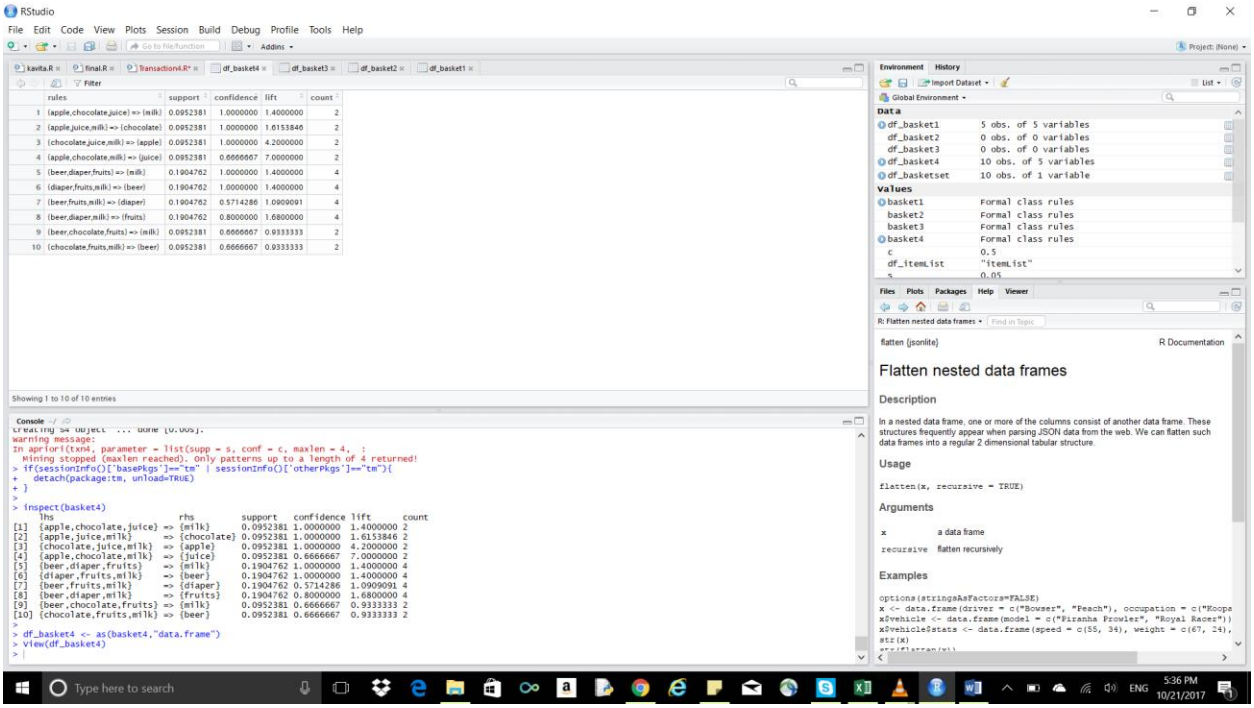
	lhs		rhs	support	confidence	lift	count
[1]	{beer,eggs}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[2]	{eggs,milk}	=>	{beer}	0.0952381	1.0000000	1.4000000	2
[3]	{apple,juice}	=>	{chocolate}	0.0952381	1.0000000	1.6153846	2
[4]	{chocolate,juice}	=>	{apple}	0.0952381	1.0000000	4.2000000	2
[5]	{apple,chocolate}	=>	{juice}	0.0952381	0.5000000	5.2500000	2
[6]	{apple,juice}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[7]	{juice,milk}	=>	{apple}	0.0952381	1.0000000	4.2000000	2

```
[8]  {apple,milk}       => {juice}     0.0952381 0.6666667  7.0000000 2
[9]  {chocolate,juice}  => {milk}      0.0952381 1.0000000  1.4000000 2
[10] {juice,milk}       => {chocolate} 0.0952381 1.0000000  1.6153846 2
[11] {chocolate,yogurt} => {beer}      0.0952381 0.6666667  0.9333333 2
[12] {beer,yogurt}      => {chocolate} 0.0952381 0.6666667  1.0769231 2
[13] {chocolate,yogurt} => {milk}      0.0952381 0.6666667  0.9333333 2
[14] {milk,yogurt}      => {chocolate} 0.0952381 0.6666667  1.0769231 2
[15] {beer,yogurt}      => {milk}      0.0952381 0.6666667  0.9333333 2
[16] {milk,yogurt}      => {beer}      0.0952381 0.6666667  0.9333333 2
[17] {apple,chocolate}  => {beer}      0.0952381 0.5000000  0.7000000 2
[18] {apple,beer}       => {chocolate} 0.0952381 0.6666667  1.0769231 2
[19] {apple,chocolate}  => {milk}      0.1428571 0.7500000  1.0500000 3
[20] {apple,milk}       => {chocolate} 0.1428571 1.0000000  1.6153846 3
[21] {bread,diaper}     => {chocolate} 0.0952381 0.6666667  1.0769231 2
[22] {bread,chocolate}  => {diaper}    0.0952381 0.6666667  1.2727273 2
[23] {bread,diaper}     => {beer}      0.0952381 0.6666667  0.9333333 2
[24] {beer,bread}       => {diaper}    0.0952381 0.6666667  1.2727273 2
[25] {bread,chocolate}  => {milk}      0.0952381 0.6666667  0.9333333 2
[26] {bread,milk}       => {chocolate} 0.0952381 0.6666667  1.0769231 2
[27] {diaper,fruits}    => {beer}      0.1904762 0.8000000  1.1200000 4
[28] {beer,fruits}      => {diaper}    0.1904762 0.5000000  0.9545455 4
[29] {beer,diaper}      => {fruits}    0.1904762 0.5000000  1.0500000 4
[30] {diaper,fruits}    => {milk}      0.1904762 0.8000000  1.1200000 4
[31] {fruits,milk}      => {diaper}    0.1904762 0.5000000  0.9545455 4
[32] {diaper,milk}      => {fruits}    0.1904762 0.5714286  1.2000000 4
[33] {chocolate,fruits} => {beer}      0.1428571 0.6000000  0.8400000 3
[34] {chocolate,fruits} => {milk}      0.1428571 0.6000000  0.8400000 3
[35] {beer,fruits}      => {milk}      0.3333333 0.8750000  1.2250000 7
[36] {fruits,milk}      => {beer}      0.3333333 0.8750000  1.2250000 7
[37] {beer,milk}        => {fruits}    0.3333333 0.6363636  1.3363636 7
[38] {chocolate,diaper} => {beer}      0.1428571 0.5000000  0.7000000 3
[39] {chocolate,diaper} => {milk}      0.1428571 0.5000000  0.7000000 3
[40] {beer,diaper}      => {milk}      0.2380952 0.6250000  0.8750000 5
[41] {diaper,milk}      => {beer}      0.2380952 0.7142857  1.0000000 5
[42] {beer,chocolate}   => {milk}      0.2380952 0.6250000  0.8750000 5
[43] {chocolate,milk}   => {beer}      0.2380952 0.5555556  0.7777778 5
```

Result for Database4 for **frequent -3 itemsets**

```
basket4<-apriori(txn4,parameter = list(supp = s, conf = c,maxlen=4,minlen=4))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}

inspect(basket4)

df_basket4 <- as(basket4,"data.frame")

View(df_basket4)
```

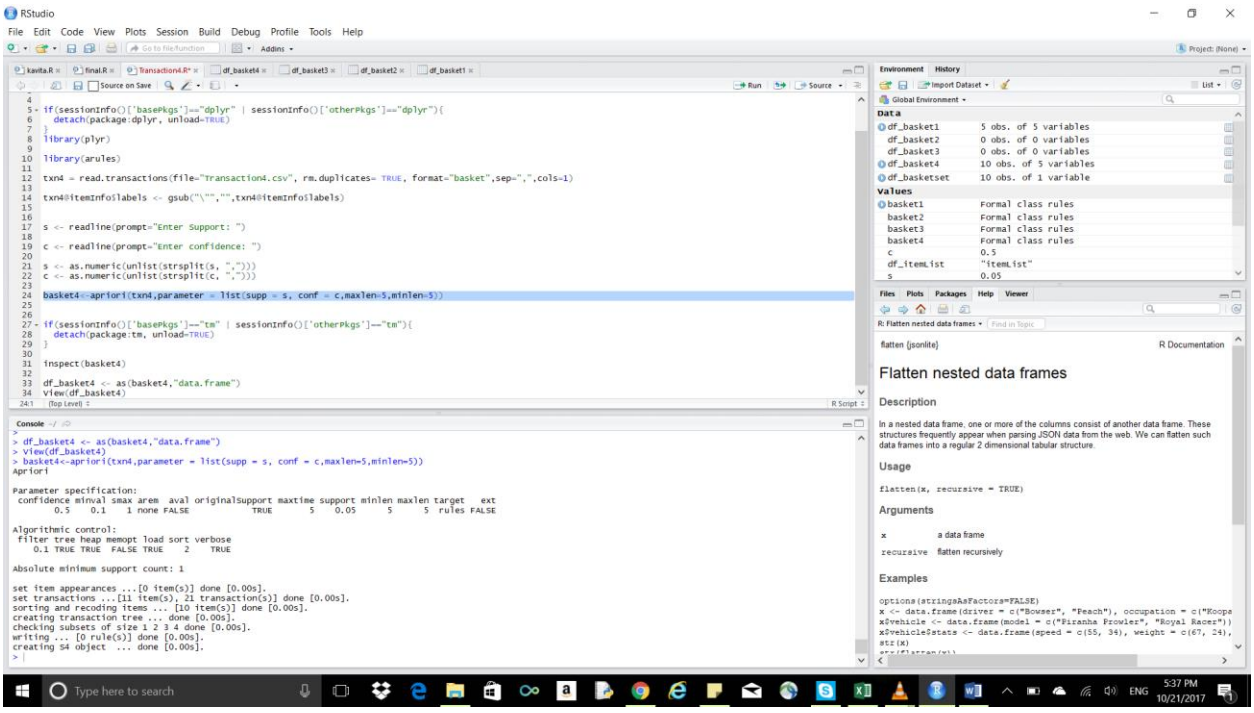


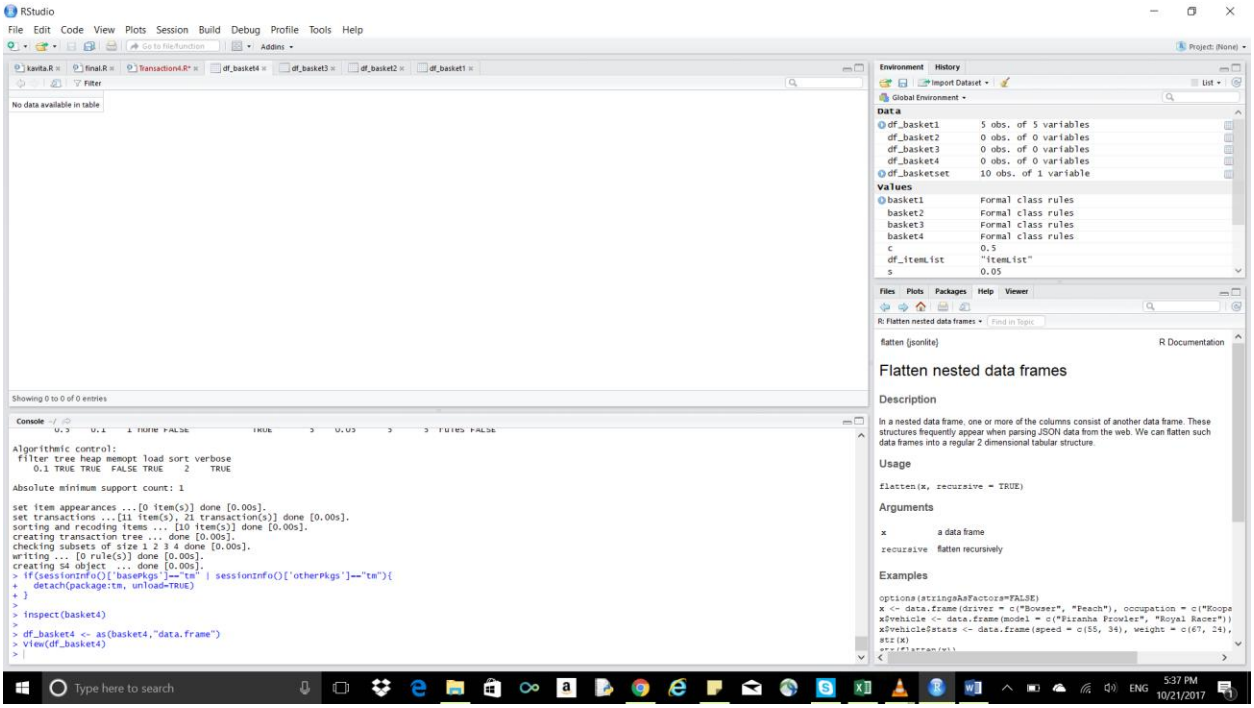


	lhs		rhs	support	confidence	lift	count
[1]	{apple,chocolate,juice}	=>	{milk}	0.0952381	1.0000000	1.4000000	2
[2]	{apple,juice,milk}	=>	{chocolate}	0.0952381	1.0000000	1.6153846	2
[3]	{chocolate,juice,milk}	=>	{apple}	0.0952381	1.0000000	4.2000000	2
[4]	{apple,chocolate,milk}	=>	{juice}	0.0952381	0.6666667	7.0000000	2
[5]	{beer,diaper,fruits}	=>	{milk}	0.1904762	1.0000000	1.4000000	4
[6]	{diaper,fruits,milk}	=>	{beer}	0.1904762	1.0000000	1.4000000	4
[7]	{beer,fruits,milk}	=>	{diaper}	0.1904762	0.5714286	1.0909091	4
[8]	{beer,diaper,milk}	=>	{fruits}	0.1904762	0.8000000	1.6800000	4
[9]	{beer,chocolate,fruits}	=>	{milk}	0.0952381	0.6666667	0.9333333	2
[10]	{chocolate,fruits,milk}	=>	{beer}	0.0952381	0.6666667	0.9333333	2

Result for Database4 for **frequent -4 itemsets**

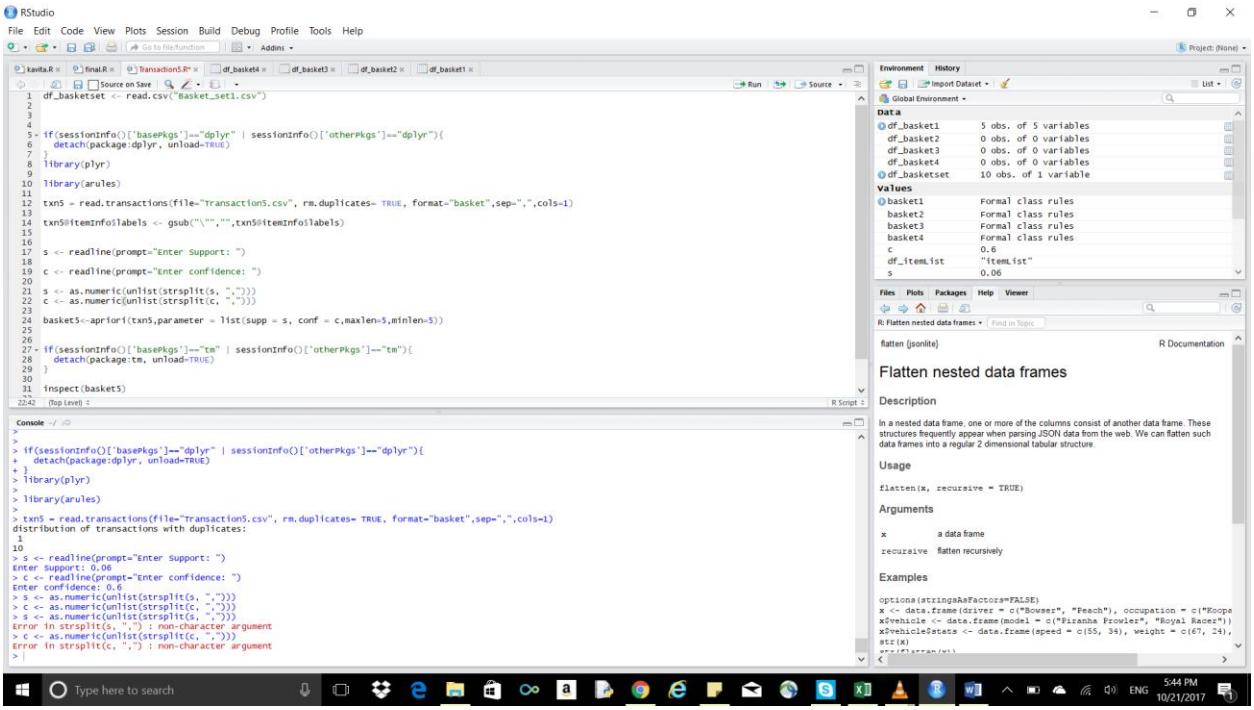
```
basket4<-apriori(txn4,parameter = list(supp = s, conf = c,maxlen=5,minlen=5))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}

inspect(basket4)

df_basket4 <- as(basket4,"data.frame")

View(df_basket4)
```




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
No data available in table
Showing 0 to 0 of 0 entries
Environment History
Import Dataset
Global Environment
Data
df_basket1 5 obs. of 5 variables
df_basket2 0 obs. of 0 variables
df_basket3 0 obs. of 0 variables
df_basket4 0 obs. of 0 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
basket2 Formal class rules
basket3 Formal class rules
basket4 Formal class rules
c 0.5
df_itemList "itemList"
s 0.05
Files Plots Packages Help Viewer
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 21 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
+ detach(package:tm, unload=TRUE)
+ }
> inspect(basket4)
> df_basket4 <- as(basket4,"data.frame")
> View(df_basket4)
Type here to search
ENG
5:37 PM
10/21/2017

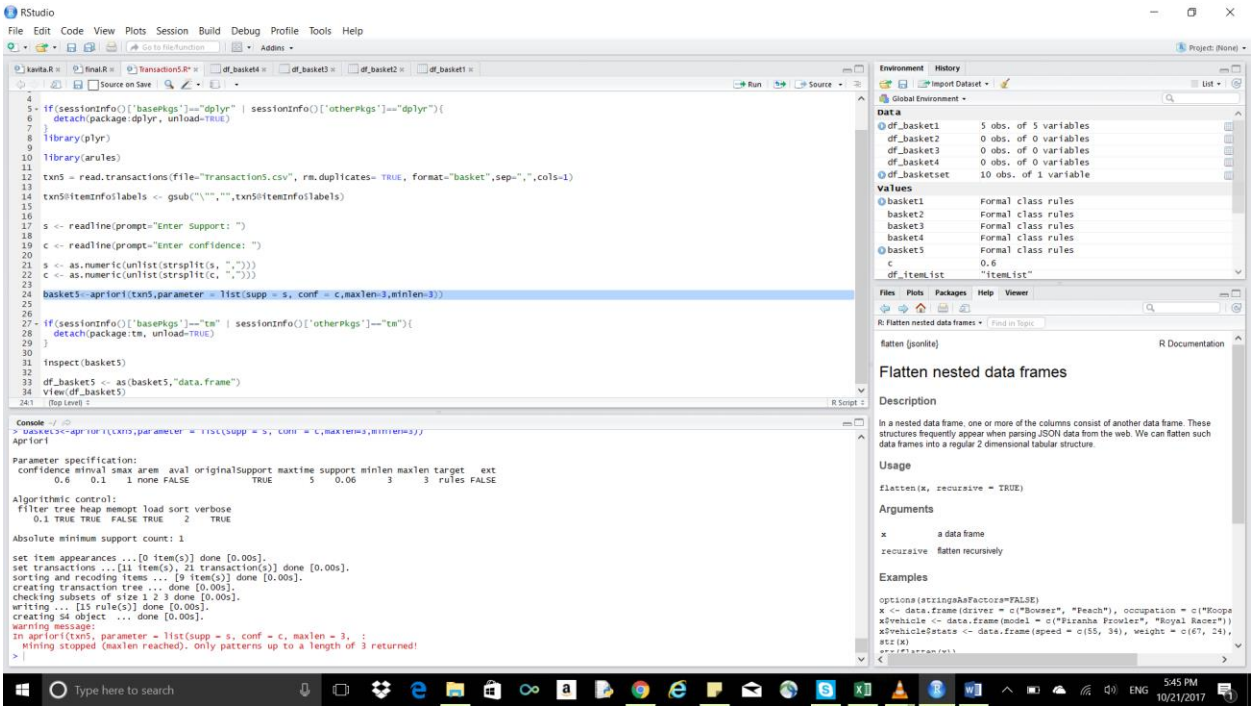
Database5: Filename- **Transaction5**

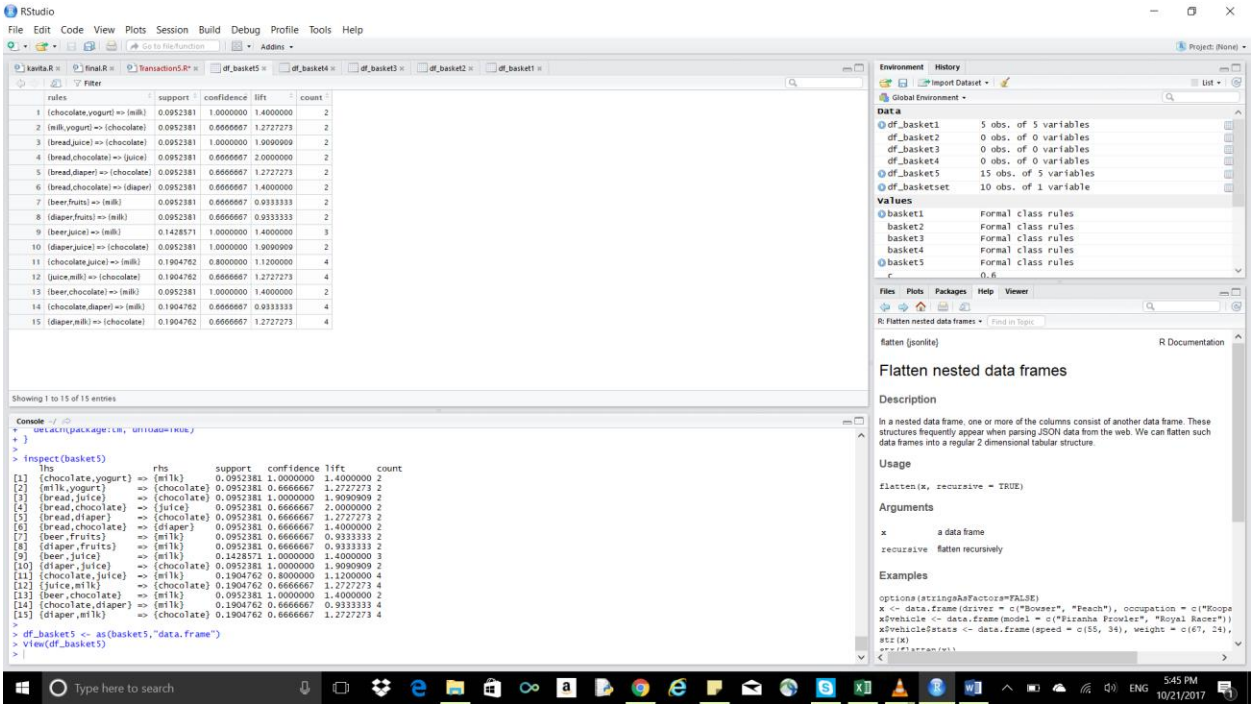
```
df_basketset <- read.csv("Basket_set1.csv")

if(sessionInfo()['basePkgs']=="dplyr" | sessionInfo()['otherPkgs']=="dplyr"){
  detach(package:dplyr, unload=TRUE)
}

library(plyr)

library(arules)

txn5 = read.transactions(file="Transaction5.csv", rm.duplicates= TRUE, format="basket",sep=",",cols=1)

txn5@itemInfo$labels <- gsub("\"","",txn5@itemInfo$labels)

s <- readline(prompt="Enter Support: ")

c <- readline(prompt="Enter confidence: ")

s <- as.numeric(unlist(strsplit(s, ",")))

c <- as.numeric(unlist(strsplit(c, ",")))
```



Result for Database5 for **frequent -2 itemsets**

```
basket5<-apriori(txn5,parameter = list(supp = s, conf = c,maxlen=3,minlen=3))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
  detach(package:tm, unload=TRUE)
}

inspect(basket5)

df_basket5 <- as(basket5,"data.frame")

View(df_basket5)
```

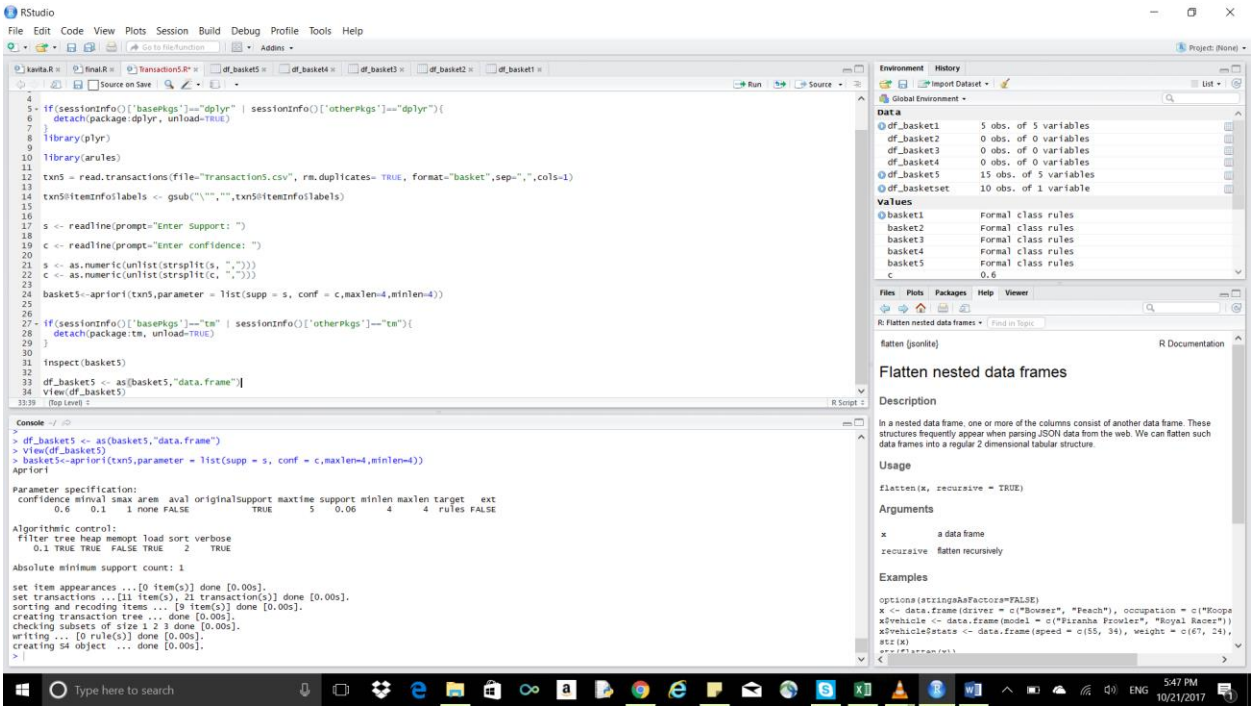


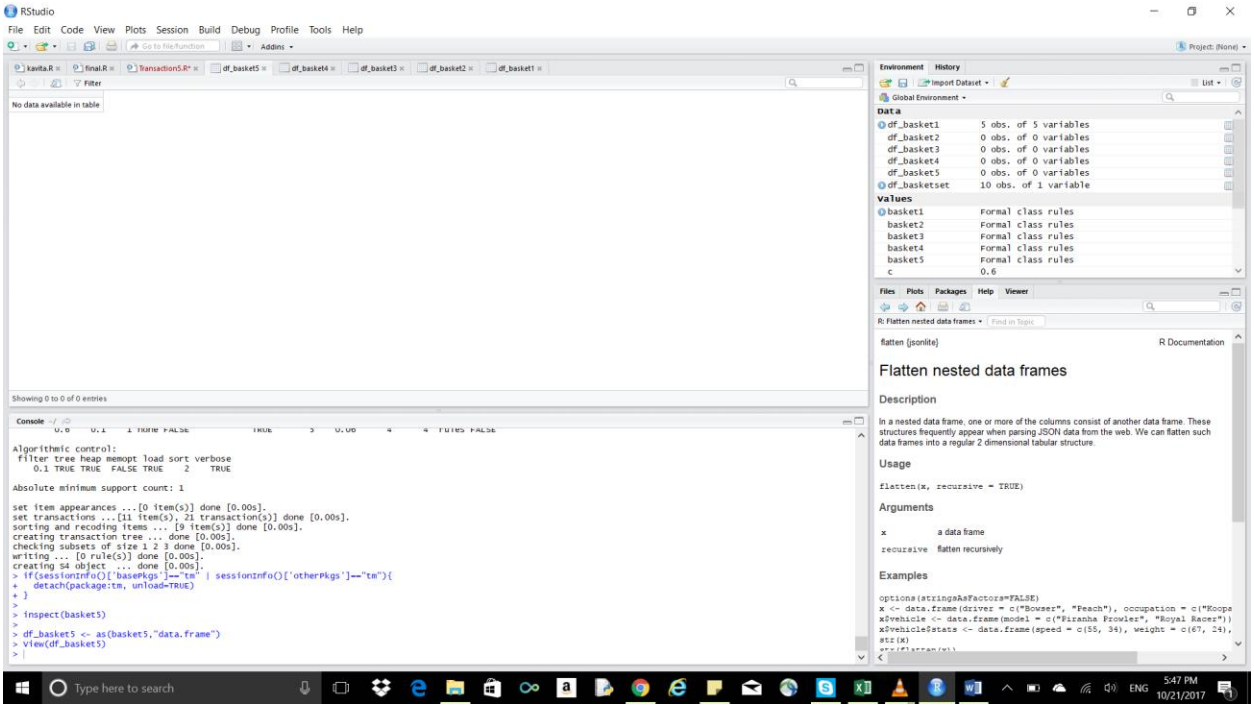



```
    lhs                     rhs          support    confidence lift      count
[1]  {chocolate,yogurt} => {milk}      0.0952381 1.0000000  1.4000000 2
[2]  {milk,yogurt}      => {chocolate} 0.0952381 0.6666667  1.2727273 2
[3]  {bread,juice}      => {chocolate} 0.0952381 1.0000000  1.9090909 2
[4]  {bread,chocolate}  => {juice}     0.0952381 0.6666667  2.0000000 2
[5]  {bread,diaper}     => {chocolate} 0.0952381 0.6666667  1.2727273 2
[6]  {bread,chocolate}  => {diaper}    0.0952381 0.6666667  1.4000000 2
[7]  {beer,fruits}      => {milk}      0.0952381 0.6666667  0.9333333 2
[8]  {diaper,fruits}    => {milk}      0.0952381 0.6666667  0.9333333 2
[9]  {beer,juice}       => {milk}      0.1428571 1.0000000  1.4000000 3
[10] {diaper,juice}     => {chocolate} 0.0952381 1.0000000  1.9090909 2
[11] {chocolate,juice}  => {milk}      0.1904762 0.8000000  1.1200000 4
[12] {juice,milk}       => {chocolate} 0.1904762 0.6666667  1.2727273 4
[13] {beer,chocolate}   => {milk}      0.0952381 1.0000000  1.4000000 2
[14] {chocolate,diaper} => {milk}      0.1904762 0.6666667  0.9333333 4
[15] {diaper,milk}      => {chocolate} 0.1904762 0.6666667  1.2727273 4
```

Result for Database5 for **frequent -3 itemsets**

```
basket5<-apriori(txn5,parameter = list(supp = s, conf = c,maxlen=4,minlen=4))

if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){

  detach(package:tm, unload=TRUE)

}

inspect(basket5)

df_basket5 <- as(basket5,"data.frame")

View(df_basket5)
```




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
No data available in table
Showing 0 to 0 of 0 entries
Environment History
Import Dataset
Global Environment
Data
df_basket1 5 obs. of 5 variables
df_basket2 0 obs. of 0 variables
df_basket3 0 obs. of 0 variables
df_basket4 0 obs. of 0 variables
df_basket5 0 obs. of 0 variables
df_basketset 10 obs. of 1 variable
Values
basket1 Formal class rules
basket2 Formal class rules
basket3 Formal class rules
basket4 Formal class rules
basket5 Formal class rules
c 0.6
Files Plots Packages Help Viewer
flatten {jsonlite} R Documentation
Flatten nested data frames
Description
In a nested data frame, one or more of the columns consist of another data frame. These structures frequently appear when parsing JSON data from the web. We can flatten such data frames into a regular 2 dimensional tabular structure.
Usage
flatten(x, recursive = TRUE)
Arguments
x a data frame
recursive flatten recursively
Examples
Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
Absolute minimum support count: 1
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[11 item(s), 21 transaction(s)] done [0.00s].
sorting and recoding items ... [9 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> if(sessionInfo()['basePkgs']=="tm" | sessionInfo()['otherPkgs']=="tm"){
+ detach(package:tm, unload=TRUE)
+ }
> inspect(basket5)
> df_basket5 <- as(basket5,"data.frame")
> View(df_basket5)
Type here to search
ENG
5:47 PM
10/21/2017