



Ask Analytics
Analytics made easy !



Assembling an army of
Data Gorillas ...



Pages

- Random
- Base SAS
- Advanced SAS
- SAS 'n' Stats
- Fundoo Stuff
- SAS Certification
- Magic with Excel
- R
- Python
- Interview FAQs
- Hot Jobs
- Who we are



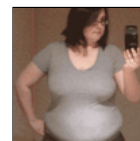
आलसियों के लिए सबसे असरदार चर्बी जलाने का तरीका! केवल अभी 50% छूट.



Enough to go with a belly like a pregnant!



बिना मेहनत के वजन घटाना है? तेरे प्रोडक्ट! 3 हफ्ते में 17 किलो!



Magic transformation in a few weeks!

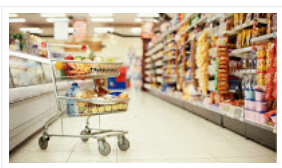


My weight was 85 kg! I was losing 3 kg of day! I just ate a plate of ...

Sponsored by



Market Basket Analysis in R



Market Basket Analysis in R with example

How can we identify the different products which can be bundled together to increase the sales ? The answer of the question is Market Basket Analysis or Apriori Algorithm.

Do you know, how to run the Apriori algorithm in R ?

This article has been written in continuation of the previous article covering [Basic of Market Basket Analysis](#).

We are taking a very common example of grocery store to make you understand the algorithm step by step. Sample snapshot of the data is given right side:

[Please download the data file that would be used in this analysis.](#)

There are only two variables in the datasets

- 1) Customer Id - Unique identity number of customers
- 2) Products - Products bought by the customers

We would now cover all steps to run apriori algorithm in R

1. Let's first import the data.

Code 1:

```
mba_data<-read.csv("C:\\MBA_data_new.csv") # we are
creating a data frame by importing csv file
```

Customer_id	Products
1	bread
1	butter
1	eggs
1	milk
2	beer
2	bread
2	cheese
2	chips
2	Mayo
2	soda
3	banana
3	bread
3	butter
3	cheese
3	oranges

```

> mba_data
  Customer_Id Products
1           1   bread
2           1   butter
3           1    eggs
4           1    milk
5           4    buns
6           4    chips
7           4    beer
8           4 musturd
9           4    soda
10          2    beer
11          2   bread
12          2  cheese
13          2    chips
14          2    mayo
15          2    soda
16          3  banana
17          3   bread
18          3   butter
19          3  cheese
20          3  oranges
21          5    buns
22          5    chips
23          5    beer
24          5 musturd
25          5 pickels
26          5    soda
27          6   bread
28          6   butter
29          6 chocolate
30          6    eggs
  
```

<< Here is the screen shot of data in R.

mba_data(data frame in R) has two variable customer_id and products.

Each customer id has bought some products for example:
customer id 1 has bought

Bread
Butter
Eggs
Milk

2. We cannot directly use imported data to run apriori algorithm. We need to aggregate it first by customer id and transform into different format.

Code 2 :

```
trans <- split(mba_data$Products, mba_data$Customer_Id,"transactions")
```

```
head(trans) # you can check top 6 observation using head() function
```

Screenshot of top transaction :

```

> head(trans)
$`1`
[1] bread butter eggs milk
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda

$`2`
[1] beer bread cheese chips mayo soda
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda

$`3`
[1] banana bread butter cheese oranges
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda

$`4`
[1] buns chips beer musturd soda
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda

$`5`
[1] buns chips beer musturd pickels soda
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda

$`6`
[1] bread butter chocolate eggs milk
Levels: banana beer bread buns butter cheese chips chocolate eggs mayo milk musturd oranges pickels soda
  
```

We have transformed the data into the desired format to run the apriori algorithm. In order to run apriori algorithm, first, we need to install and load [arules](#) library package using below code.

Code 3:

```
install.packages("arules") # install arules library package

library(arules)           # loading arules library
```

Below are the screenshots of R Console :

```
> install.packages("arules")
Loading required package: arules
Installing package into 'C:/users/vinod.pandey/documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.2/arules_1.2-1.zip'
Content type 'application/zip' length 1873859 bytes (1.8 MB)
downloaded 1.8 MB

package 'arules' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\users\vinod.pandey\AppData\Local\Temp\Rtmpc9acwL\downloaded_packages
```

```
> library(arules)
Loading required package: Matrix
Attaching package: 'arules'

The following objects are masked from 'package:base':

    %in%, abbreviate, write

warning message:
package 'arules' was built under R version 3.2.2
> |
```

We have installed the arules library. Now we can run the apriori algorithm using following statement:

Code 4:

```
rules = apriori(trans, parameter=list(support=0.10, confidence=0.5,maxlen=2,minlen=2))
```

Formula explanation

1 2 3 4 5

```
rules = apriori(trans, parameter=list(support=0.10, confidence=0.5,maxlen=2,minlen=2))
```

- 1 Trans : aggregated dataset to run the algorithm
- 2 Support : minimum support level to filter the transactions. Here, minimum support is .10(10%) that simply means if any products has been bought by less than 10% of customer would not be considered in the rules.
- 3 Confidence : minimum confidence level to filter the transactions.
- 4 maxlen : it means that we are looking maximum for two products relation only product a -> product b (how product a is influencing the product b)
- 5 minlen : it means that we are looking minimum for two products relation only product a -> product b (how product a is influencing the product b)

Below are the screen shot of R result.

```

> rules = apriori(trans, parameter=list(support=0.10, confidence=0.5,maxlen=2,minlen=2))

Parameter specification:
confidence minval smax arem aval originalSupport support minlen maxlen target ext
0.5 0.1 1 none FALSE TRUE 0.1 2 2 rules FALSE

Algorithmic control:
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE

warning in apriori(trans, parameter = list(support = 0.1, confidence = 0.5, :
You chose a very low absolute support count of 1. You might run out of memory! Increase minimum support

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09) (c) 1996-2004 Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 15 transaction(s)] done [0.00s].
sorting and recoding items ... [14 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.01s].
writing ... [66 rule(s)] done [0.00s].
creating 54 object ... done [0.00s].

```

We have successfully derived 66 rules(written second last line in the above screenshot).

Now lets have a look on those rules.

Code 5:

```
inspect(rules) # to get the rules
```

Screenshot for R.

```

> inspect(rules)
lhs      rhs      support  confidence lift
1 {buns}    => {mustard} 0.2000000 1.0000000 3.0000000
2 {mustard} => {buns}    0.2000000 0.6000000 3.0000000
3 {buns}    => {chips}   0.2000000 1.0000000 2.5000000
4 {chips}   => {buns}    0.2000000 0.5000000 2.5000000
5 {buns}    => {soda}    0.2000000 1.0000000 2.1428571
6 {buns}    => {beer}    0.2000000 1.0000000 2.1428571
7 {mayo}    => {mustard} 0.1333333 0.6666667 2.0000000
8 {mayo}    => {chips}   0.2000000 1.0000000 2.5000000
9 {chips}   => {mayo}    0.2000000 0.5000000 2.5000000
10 {mayo}   => {cheese}  0.2000000 1.0000000 2.5000000
11 {cheese} => {mayo}    0.2000000 0.5000000 2.5000000
12 {mayo}   => {soda}    0.2000000 1.0000000 2.1428571
13 {mayo}   => {beer}    0.2000000 1.0000000 2.1428571
14 {mayo}   => {bread}   0.2000000 1.0000000 1.3636364
15 {oranges} => {banana}  0.2000000 0.7500000 2.2500000
16 {banana} => {oranges}  0.2000000 0.6000000 2.2500000
17 {oranges} => {eggs}   0.1333333 0.5000000 1.2500000
18 {oranges} => {butter}  0.1333333 0.5000000 1.0714286
19 {oranges} => {milk}   0.2000000 0.7500000 1.6071429
20 {oranges} => {cheese}  0.1333333 0.5000000 1.2500000
21 {oranges} => {bread}   0.2000000 0.7500000 1.0227273
22 {chocolate} => {eggs}  0.2000000 0.7500000 1.8750000
23 {eggs}   => {chocolate} 0.2000000 0.5000000 1.8750000
24 {chocolate} => {butter} 0.1333333 0.5000000 1.0714286
25 {chocolate} => {milk}  0.2000000 0.7500000 1.6071429
26 {chocolate} => {bread}  0.2000000 0.7500000 1.0227273
27 {banana} => {cheese}   0.2000000 0.6000000 1.5000000
28 {cheese} => {banana}   0.2000000 0.5000000 1.5000000
29 {banana} => {bread}    0.2000000 0.6000000 0.8181818

```

Lets manually validate the first two rules (buns => mustard and mustard => buns).

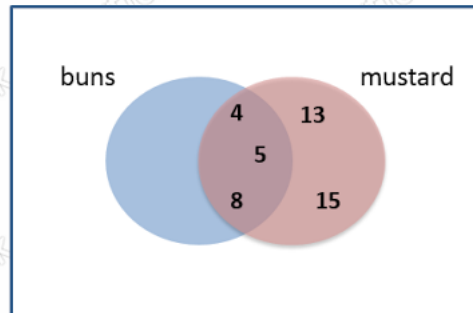
I hope that you would have read our [last blog on Market Basket Analysis](#). In our last article we have explicitly explained [support, confidence and lift](#).

Please read that article before getting into below calculation to understand it better.

Data

Customer_Id	Products
4	mustard
5	mustard
8	mustard
13	mustard
15	mustard

Customer_Id	Products
4	buns
5	buns
8	buns



Support, Confidence and Lift

Total number of customers : 15

Support

Number of customers who bought mustard: 5 $\Rightarrow \text{support}(\text{mustard}) = 5/15 = .33 \text{ (33\%)}$

Number of customer who bought buns : 3 $\Rightarrow \text{support}(\text{buns}) = 3/15 = .2 \text{ (20\%)}$

Number of customers who bought buns and mustard both : 3

$\Rightarrow \text{support}(\text{buns and mustard}) = 3/15 = .2 \text{ (20\%)}$

Confidence

$\{\text{buns}\} \Rightarrow \{\text{mustard}\}$ = means who buys buns also buys mustard = $3/3 = 100\%$

$\{\text{mustard}\} \Rightarrow \{\text{buns}\}$ = means who buys mustard also buys buns = $3/5 = 60\%$

Lift

$\text{Lift}(\text{mustard}) = \text{confidence}(\text{buns, mustard}) / \text{support}(\text{mustard}) = 100\% / 33.33\% = 3$

$\text{Lift}(\text{buns}) = \text{confidence}(\text{mustard, buns}) / \text{support}(\text{buns}) = 60\% / 20\% = 3$

If you want to export the rule file into csv file, you can get it by below mentioned code.

Code 6:

```
write(rules,file="mba_rules1.csv",sep="," ,row.names = FALSE)
```

I hope, the article is useful in understanding the market basket analysis

Enjoy reading our other articles and stay tuned with ...

Kindly do provide your feedback in the 'Comments' Section and share as much as possible.

Posted by [Vinod Pandey](#)

[Email This](#)[BlogThis!](#)[Share to Twitter](#)[Share to Facebook](#)[Share to Pinterest](#)

 +5 Recommend this on Google

[Newer Post](#) [Older Post](#) [Home](#)
 Subscribe to: [Post Comments \(Atom\)](#)
Finally it happened !



On your plate

[statistics R](#) [Advanced SAS](#) [Base SAS](#) [Linear Regression](#) [interview](#)
[Text Mining](#) [Logistic Regression](#) [cluster analysis](#) [Magic of Excel](#)
[Python](#) [Base SAS](#) [certification](#) [time-series](#) [Decision Science](#)
[forecasting](#) [Macro](#) [ARIMA](#) [Market Basket Analysis](#) [NLP](#) [R](#)
[Visualization](#) [SAS](#) [Gems](#) [Sentiment Analysis](#) [automation](#) [Cool](#)
[Dashboards](#) [Factor Analysis](#) [Principal Component Analysis](#) [SAS](#)
[Projctcs](#) [Conjoint Analysis](#) [X Commands](#) [guesstimate](#)



[Ask Analytics @ Facebook](#)



Search This Blog

Loading...



Popular Posts



[Excel](#)

[Create your own Google in Excel](#)

Learn to create G o o g l e in Excel Excited ? Confused ?
 ... Don't be, as you are one click away from learning how
 to create y...



[SQL in SAS](#)

SQL and Excel are omnipresent ! I am stating from my
 experience in Analytics industry. Every analyst in
 his/her life time, sometime would n...



[SAS Macros - Part 4 - A finishing touch to Macro](#)

We have already covered most about Macros our
 previous articles. This article has been written in order
 to give it a finishing touch. ...



[Text Mining in R - Part 7](#)

Sentiment Analysis in R - Coolest Method So Far So far
 we have discussed all the basics, a rudimentary
 method, an evolved method and a c...



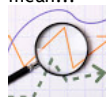
[Fun with Beer and of course Excel](#)

Ever got High over the Beer 'n' Excel punch?? It's been
 long, may be since our school days, that we all are using
 Excel as a...



Understanding p-value

A tale about p-value You keep seeing the term 'p-value' every now and then, but don't understand what it really mean...



Time Series Forecasting - Part 5

ARIMA using SAS We have covered basics about time series and also the basic methods of forecasting. It is time to learn the most importa...



Descriptive Statistics With Proc Univariate

Feel your data ! Before going to a battle, a warrior better know what he is fighting against and so a data analyst ! It is advised to ...



Logistic Regression - Part 1 - Theory

In the Movie Matrix, while sentiment machines captured the world with a "Matrix" designed with a binary logic (1 and 0)...



Market Basket Analysis in R

Market Basket Analysis in R with example How can we identify the different products which can be bundled together to increase the sal...



Follow by Email



loading...



Impress with your



Solution of previous



Fall in love with excel



Copyright 2015: Ask Analytics. Simple template. Powered by [Blogger](#).