Lecture 2: Association Rule Mining

Pilsung Kang

School of Industrial Management Engineering

Korea University

AGENDA

01 Association Rules – A Priori algorithm

02 R Exercise

Type of Machine Learning/Data Mining

- According to the existence of target (Y) variable
  - ✓ Supervised learning vs. Unsupervised learning

Supervised Learning

A given dataset **X** & **Y**

	Var. 1	Var. 2	...	Var. d
Ins. 1	..	..	...	..
Ins. 2	..	..	...	..
...	...	...	...	...
Ins. N	..	..	..	..

$y = f(x)$

Y
..
..
...
..

Unsupervised Learning

A given dataset **X**

	Var. 1	Var. 2	...	Var. d
Ins. 1	..	..	...	..
Ins. 2	..	..	...	..
...	...	...	...	...
Ins. N	..	..	..	..

Semi-supervised Learning

A given dataset **X** & **Y**

	Var. 1	Var. 2	...	Var. d
Ins. 1	..	..	...	..
Ins. 2	..	..	...	..
...	...	...	...	...
Ins. N	..	..	..	..
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Ins. M	..	..	..	..

$y = f(x)$

Y
..
..
...
..

Type of Machine Learning/Data Mining

- Unsupervised Learning

$$\mathcal{X} = \{\mathbf{x}_i | i = 1, \ldots, n, \ \mathbf{x}_i \in \mathbb{R}^d\}$$

Unsupervised learning

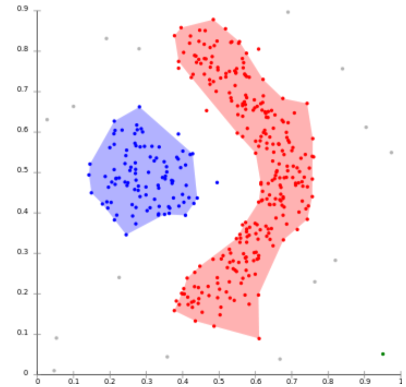
- Explores intrinsic characteristics.
- Estimates underlying distribution.
- Density estimation, clustering, novelty detection, etc.

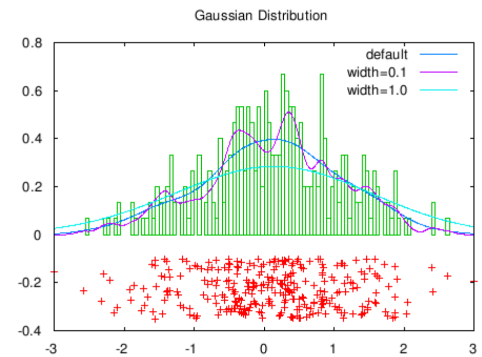
Type of Machine Learning/Data Mining

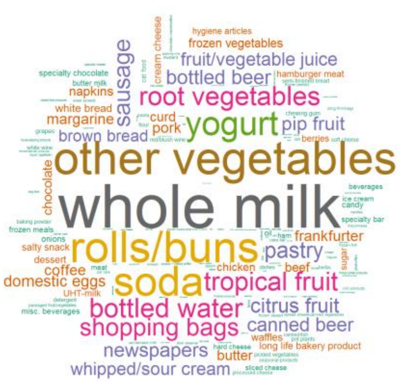
- Unsupervised Learning

A given dataset **X**

	Var. 1	Var. 2	…	Var. d
Ins. 1	..	..	…	..
Ins. 2	..	..	…	..
…	…	…	…	…
Ins. N	..	..	..	..







Type of Machine Learning/Data Mining

- Supervised Learning

$$\mathcal{X} = \{\mathbf{x}_i | i = 1, \ldots, n, \ \mathbf{x}_i \in \mathbb{R}^d\}$$

$y = f(x)$

Supervised learning

- Finds relations between X and Y.
- Estimate the underlying function $y = f(x)$.
- Classification, regression.

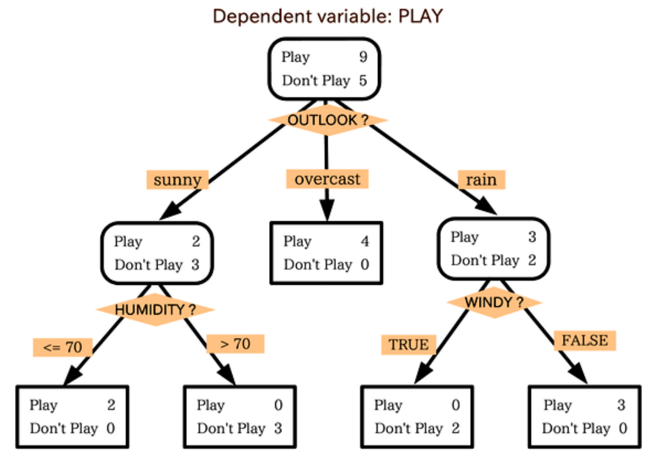
$$\mathcal{Y} = \{y_i | i = 1, \ldots, n, \ y_i = f(\mathbf{x}_i)\}$$

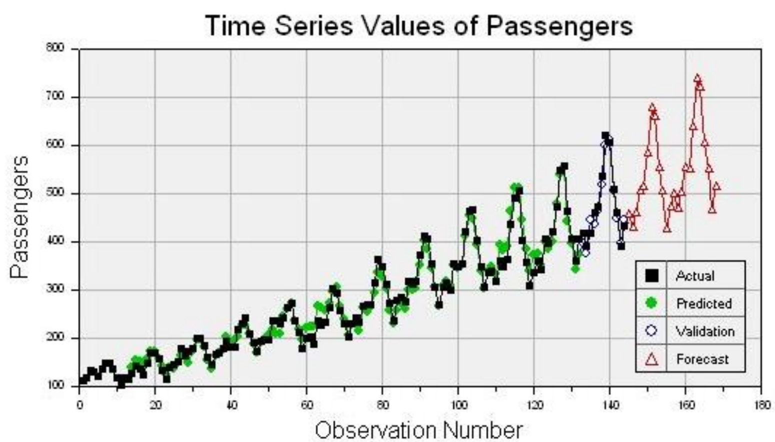
Type of Machine Learning/Data Mining

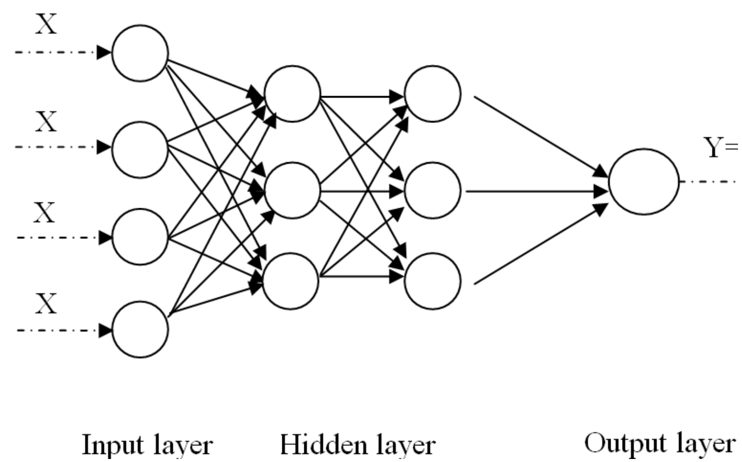
- Supervised Learning

	Var. 1	Var. 2	...	Var. d	Y
Ins. 1	..	..	...	..	..
Ins. 2	..	..	...	..	..
...	...	...	...	...	...
Ins. N	..	..	..	..	..

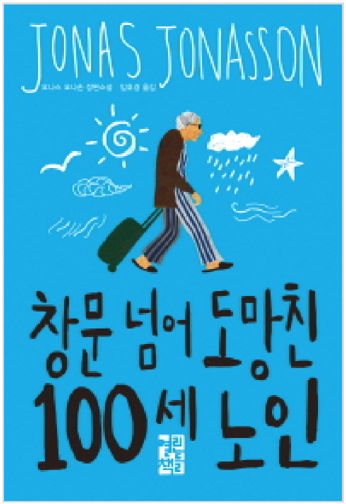
$y = f(x)$







Association Rule Mining



크게보기 | 미리보기

매장 재고 · 위치 >

키워드 Pick 안내

양로원 갱단
트렁크 데뷔작
율리우스 핵폭탄

이 책의 다른 상품 정보

sam : 한달 3권 9,900원 >

eBook : 9,000원 >

원서/번역서 : [보유] The Hundred-Year-Old Man Who Climbed Out of the Window and Disappeared >

오늘의책 무료배송 소득공제

창문 넘어 도망친 100세 노인 요나스 요나손 장편소설

요나스 요나손 지음 | 임호경 옮김 | 열린책들 | 2013년 07월 25일 출간

★★★★★ 리뷰 112개 리뷰쓰기 | 9.0(137)

KBS TV책 -김창완과 책읽기

정가 : 13,800원

판매가 : 12,420원 [10%↓ 1,380원 할인]

통합포인트 : [기본적립] 690원 적립 [5% 적립] 안내
[추가적립] 5만원 이상 구매 시 2천원 추가적립
[회원혜택] 우수회원 5만원 이상 구매 시 2~3% 추가적립

추가혜택 : 카드/포인트 안내 도서소득공제 안내 추가혜택 더보기

배송비 : 무료 배송비 안내

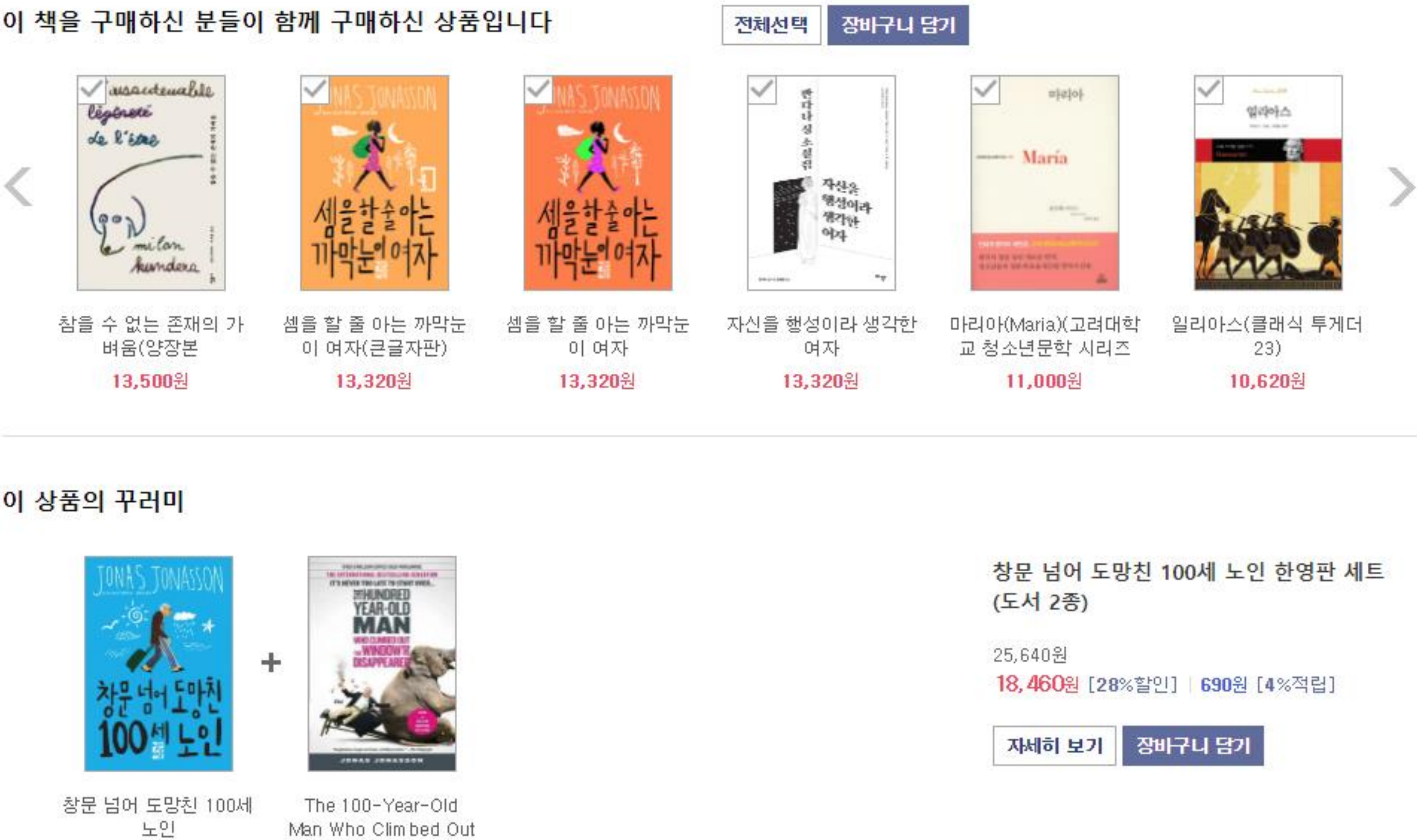
배송일정 : 서울특별시 종로구 세종대로 기준 지역변경
03월 04일 출고 예정 배송일정 안내

바로드림 : 인터넷으로 주문하고 매장에서 직접 수령 안내

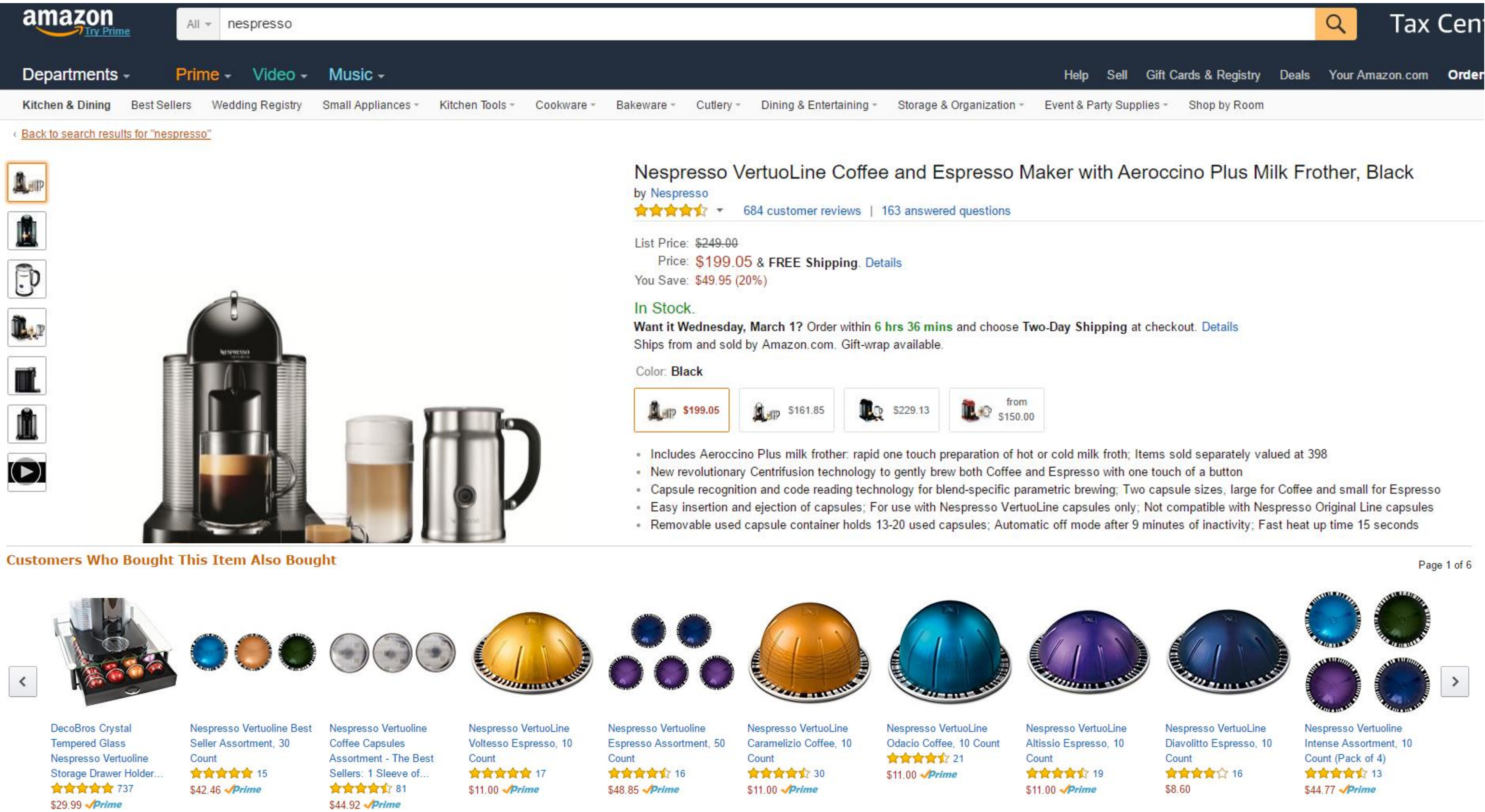
주문수량 1 + −

장바구니 담기 | 바로구매 | 바로드림 주문 | 선물하기 | 보관함 담기

Association Rule Mining



Association Rule Mining



Association Rule Mining

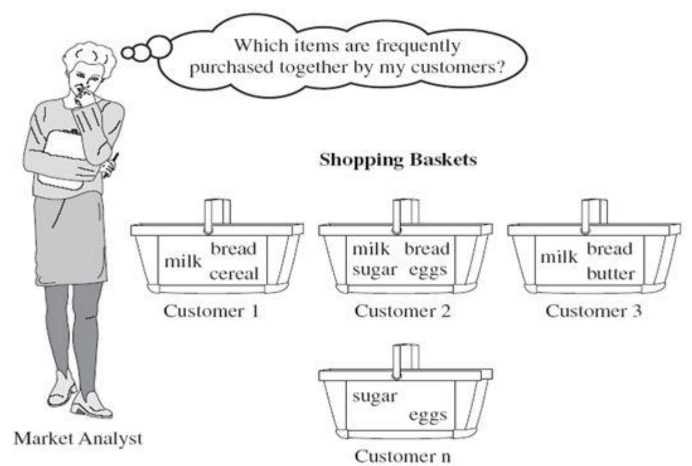
- Also known as "Market Basket Analysis"

Wall Mart (USA)

E-Mart (Korea)

Association Rule Mining

- Goal:
  - ✓ Produce rules that define "what goes with what"
  - ✓ "If X was purchased, then Y was also purchased"
- Features
  - ✓ Rows are transactions
  - ✓ Used in recommendation systems – "Our records show that you bought X, thus you may also like Y"
  - ✓ Also called "affinity analysis" or "market basket analysis"



Association Rule Mining

- Dataset for association rule mining
  - ✓ Each transaction is represented as a record
  - ✓ Two representations are possible: (1) item list and (2) item matrix

TId	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

TId	Bread	Milk	Diapers	Beer	Eggs	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Association Rule Mining

- A toy example: a tiny retail market data

Transaction	Item 1	Item 2	Item 3	Item 4
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Association Rule Mining

- Terminology
  - ✓ Antecedent – "IF" part
  - ✓ Consequent – "THEN" part
  - ✓ Item set – the items comprising the antecedent or consequent
  - ✓ Antecedent and consequent are disjoint (have no items in common)

- Generating rules
  - ✓ Many rules are possible (e.g., for transaction 1)
    - If egg is bought, then noddle is also bought
    - If egg and noddle are bought, then tuna is also bought
    - If tuna is bought, then egg is also bought, etc.

Association Rule Mining

Performance Measures for the rule A → B

- Support

$$\text{support}(A) = P(A)$$

  ✓ Used to find the frequent item sets

- Confidence

$$confidence(A \rightarrow B) = \frac{P(A,B)}{P(A)}$$

  ✓ Used to generate meaningful rules

- Lift

$$lift(A \rightarrow B) = \frac{P(A,B)}{P(A) \cdot P(B)}$$

  ✓ Used to determine the usefulness of generated rules

Association Rule Mining

- How to generate an effective association rules?
  - ✓ Ideally, create all possible combinations of items and see what rules are effective and what rules are not.
  - ✓ Computation time grows exponentially as the number of items increases.

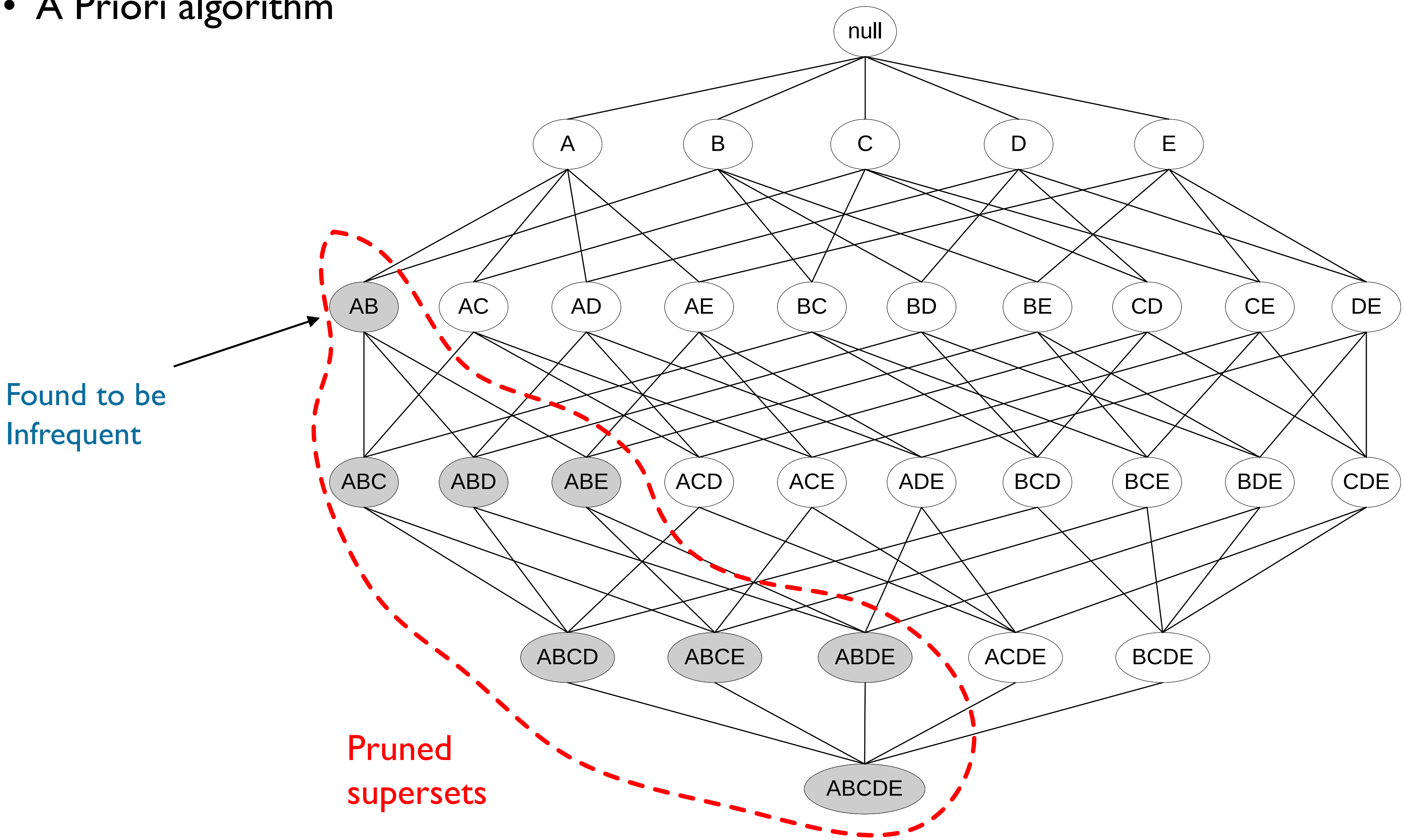
- Brute-force approach
  - ✓ List all possible association rules
  - ✓ Compute the support and confidence for each rule
  - ✓ Prune rules that fail the **minsup** and **minconf** threshold.
  - ✓ Computationally prohibitive!

Association Rule Mining

- A priori algorithm
  - ✓ Consider only "frequent item sets"
  - ✓ "support"
    - Criterion for item set frequency P(A)
    - #(%) of transactions that include both the antecedent and the consequent
    - Support for the item set {egg, noodle} is 4 out of transactions, or 40%
  - ✓ Support of an itemset never exceeds the support of its subsets, which is known as anti-monotone property of support.

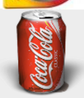
Association Rule Mining

- A Priori algorithm



Association Rule Mining

- Generating frequent item sets
  - ✓ Users set a minimum support criterion: e.g. 2 transactions or 20%

Transaction	Item 1	Item 2	Item 3	Item 4
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Association Rule Mining

- Generating frequent item sets
  - ✓ Generate the list of one-item sets that meets the support criterion
    - Support {noodle} = 8/10 = 80%
    - Support {egg} = 5/10 = 50%
    - Support {cola} = 5/10 = 50%
    - Support {rice} = 3/10 = 30%
    - Support {tuna} = 2/10 = 20%
    - Support {onion} = 1/10 = 10%
  - ✓ Onion is removed because it does not meet the minimum support criterion

Association Rule Mining

- Generating frequent item sets
  - ✓ Use the life of one-item sets to generate list of two-item sets that meet the support criterion

	noodle	egg	cola	rice	tuna
noodle		40%	40%	20%	20%
egg			30%	0%	20%
cola				0%	10%
rice					0%
tuna					

  - ✓ {noodle, egg}, {noodle, cola}, {noodle, rice}, {noodle, tuna}, {egg, cola}, {egg, tuna} are frequent two-item sets

Association Rule Mining

- Generating frequent item sets
  - ✓ Use the list of two-item sets to generate the three-item sets.
  - ✓ Continue up through k-item sets.

Set-size	Item 1	Item 2	Item 3	…	Item 6
1	noodle				
1	egg				
1	cola				
1	rice				
1	tuna				
2	noodle	egg			
2	noodle	cola			
2	noodle	rice			
…	…	…			

Association Rule Mining

- A Priori algorithm
  - ✓ Let k=1
  - ✓ Generate frequent itemsets of length 1
  - ✓ Repeat until no new frequent itemsets are identified
    - Generate length (k+1) candidate itemsets from length k frequent itemsets
    - Prune candidate itemsets containing subsets of length k that are infrequent
    - Count the support of each candidate by scanning the DB
    - Eliminate candidates that are infrequent, leaving only those that are frequent

Association Rule Mining

- Confidence
  - ✓ The % of antecedent transactions that also have the consequent item set
  - ✓ E.g. "if noddle is purchased, then egg is also purchased"

$$\text{support}(noodle) = P(noodle) = \frac{8}{10}, \quad \text{support}(egg) = P(egg) = \frac{5}{10}$$

$$confidence(noodle \to egg) = \frac{P(noodle, egg)}{P(noodle)} = \frac{4/10}{8/10} = 0.5(50\%)$$

  - ✓ Benchmark confidence: transactions with consequent (P(egg), support(egg)) of all transactions
  - ✓ If the confidence of (noodle → egg) is smaller then the support of egg, this rule is useless

Association Rule Mining

- Lift
  - ✓ Confidence/(benchmark confidence)

$$lift(noodle \rightarrow egg)$$

$$= \frac{confidence(noodle \rightarrow egg)}{\text{support}(egg)} = \frac{\frac{P(noodle, egg)}{P(noodle)}}{P(egg)} = \frac{P(noodle, egg)}{P(noodle) \times P(egg)}$$

$$= \frac{\frac{4}{10}}{\frac{8}{10} \times \frac{5}{10}} = 1$$

  - ✓ If lift = 1, then the antecedent and the consequents are statistically independent
  - ✓ If lift > 1, then the rule is useful in finding consequent item sets

Association Rule Mining

- Generated rules
  - ✓ Set the support to 20%.
  - ✓ Set the confidence to 70%.

Rule #	Antecedent (a)	Consequent	Support	Confidence	Lift
1	tuna=>	egg, noodle	2	100	2.5
2	tuna=>	egg	2	100	2
3	noodle, tuna=>	egg	2	100	2
4	rice=>	noodle	3	100	1.25
5	egg, tuna=>	noodle	2	100	1.25
6	tuna=>	noodle	2	100	1.25
7	cola=>	noodle	5	80	1
8	egg=>	noodle	5	80	1

Association Rule Mining

- Summary
  - ✓ Produce rules on associations between items from a database of transactions
  - ✓ Widely used in recommender systems
  - ✓ Most popular method is A-priori algorithm
  - ✓ To reduce computation, consider only "frequent" item sets (=support)
  - ✓ Performance is measured by confidence and lift

AGENDA

01 Association Rules – A Priori algorithm

02 R Exercise

Association Rule Mining: Packages

- Package “arules” & “arulesViz

Package ‘arules’

December 3, 2018

Version 1.6-2

Date 2018-12-02

Title Mining Association Rules and Frequent Itemsets

Description Provides the infrastructure for representing, manipulating and analyzing transaction data and patterns (frequent itemsets and association rules). Also provides C implementations of the association mining algorithms Apriori and Eclat.

Classification/ACM G.4, H.2.8, I.5.1

URL https://github.com/mhahsler/arules, http://lyle.smu.edu/IDA/arules

BugReports https://github.com/mhahsler/arules

Package ‘arulesViz’

December 5, 2018

Version 1.3-2

Date 2018-12-04

Title Visualizing Association Rules and Frequent Itemsets

Depends arules (>= 1.4.1), grid

Imports scatterplot3d, vcd, seriation, igraph (>= 1.0.0), graphics, methods, utils, grDevices, stats, colorspace, DT, plotly, visNetwork

Suggests graph, Rgraphviz, iplots, shiny, htmlwidgets

Description
Extends package 'arules' with various visualization techniques for association rules and itemsets. The package also includes several interactive visualizations for rule exploration.

License GPL-3

URL https://github.com/mhahsler/arulesViz, http://lyle.smu.edu/IDA/arules/

BugReports https://github.com/mhahsler/arulesViz

R Exercise: Load Dataset

- Load dataset

```
# Part 1: Transform a data file into transaction format
# Basket type
tmp_basket <- read.transactions("Transaction_Sample_Basket.csv",
         format = "basket", sep = ",", rm.duplicates=TRUE)
inspect(tmp_basket)
# Single type
tmp_single <- read.transactions("Transaction_Sample_Single.csv",
         format = "single", cols = c(1,2), rm.duplicates=TRUE)
inspect(tmp_single)
```

✓ read.basket( ) function can convert two types of dataset into a transaction format

- basket format: row is associated with transaction id and column is associated with the items in the corresponding id
- single format: each row consists of transaction id and one item in the corresponding id

R Exercise: Load Dataset

- Load dataset
  - ✓ Basket format and after conversion

	A	B	C	D	E
1	A	B	C		
2	A	C	D	E	
3	A	E	B		
4	B	C	D		
5	F	A	B		
6	A	D	F	G	
7	G	F	B	C	E
8	A	B			
9	C	D			
10	C	F	G		

```
> inspect(tmp_basket)
     items
[1]  {A,B,C}
[2]  {A,C,D,E}
[3]  {A,B,E}
[4]  {B,C,D}
[5]  {A,B,F}
[6]  {A,D,F,G}
[7]  {B,C,E,F,G}
[8]  {A,B}
[9]  {C,D}
[10] {C,F,G}
```

R Exercise: Load Dataset

- Load dataset
  - ✓ Single format and after conversion

	A	B
1	Tr1 A	
2	Tr1 B	
3	Tr1 C	
4	Tr2 A	
5	Tr2 C	
6	Tr2 D	
7	Tr2 E	
8	Tr3 A	
9	Tr3 E	
10	Tr3 B	
11	Tr4 B	
12	Tr4 C	
13	Tr4 D	
14	Tr5 F	
15	Tr5 A	
16	Tr5 B	
17	Tr6 A	
18	Tr6 D	
19	Tr6 F	
20	Tr6 G	

•••

```
> inspect(tmp_single)
     items       transactionID
[1]  {A,B,C}     Tr1
[2]  {C,G}       Tr10
[3]  {A,C,D,E}   Tr2
[4]  {A,B,E}     Tr3
[5]  {B,C,D}     Tr4
[6]  {A,B,F}     Tr5
[7]  {A,D,F,G}   Tr6
[8]  {B,C,E,F,G} Tr7
[9]  {A,B}       Tr8
[10] {C}         Tr9
```

R Exercise: Market Basket Analysis

- Load the dataset

```r
# Part 2: Association Rule Mining without sequence information
data("Groceries")
summary(Groceries)
str(Groceries)
inspect(Groceries)
```

✓ Use the "Groceries" dataset (it is already installed if you have "arules" package installed

  - Transaction data format, sparse matrix, provide some useful summary information

```
> summary(Groceries)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda           yogurt
            2513             1903             1809             1715             1372
         (Other)
           34055

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46   29
  18   19   20   21   22   23   24   26   27   28   29   32
  14   14    9   11    4    6    1    1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000

includes extended item information - examples:
       labels  level2           level1
1 frankfurter sausage meat and sausage
2     sausage sausage meat and sausage
3  liver loaf sausage meat and sausage
>
```

R Exercise: Market Basket Analysis

- Draw Wordcloud using the items

```
# Item inspection
itemName <- itemLabels(Groceries)
itemCount <- itemFrequency(Groceries)*nrow(Groceries)
col <- brewer.pal(8, "Dark2")
wordcloud(words = itemName, freq = itemCount, min.freq = 1,
         scale = c(3, 0.2), col = col, random.order = FALSE)
```

✓ itemName: item names used in the Wordcloud

✓ itemCount: item occurrence count used in the Wordcloud

✓ brewer.pal( ): color palette (usually choose one from predefined sets)

✓ wordcloud( ): Wordcloud generation function

  - words: used words, freq: item occurrence count, min.freq: minimum number of occurrence to be displayed, scale: relative scale between the most frequently bought item and the least frequently bought item

R Exercise: Market Basket Analysis

- Draw Wordcloud using the items

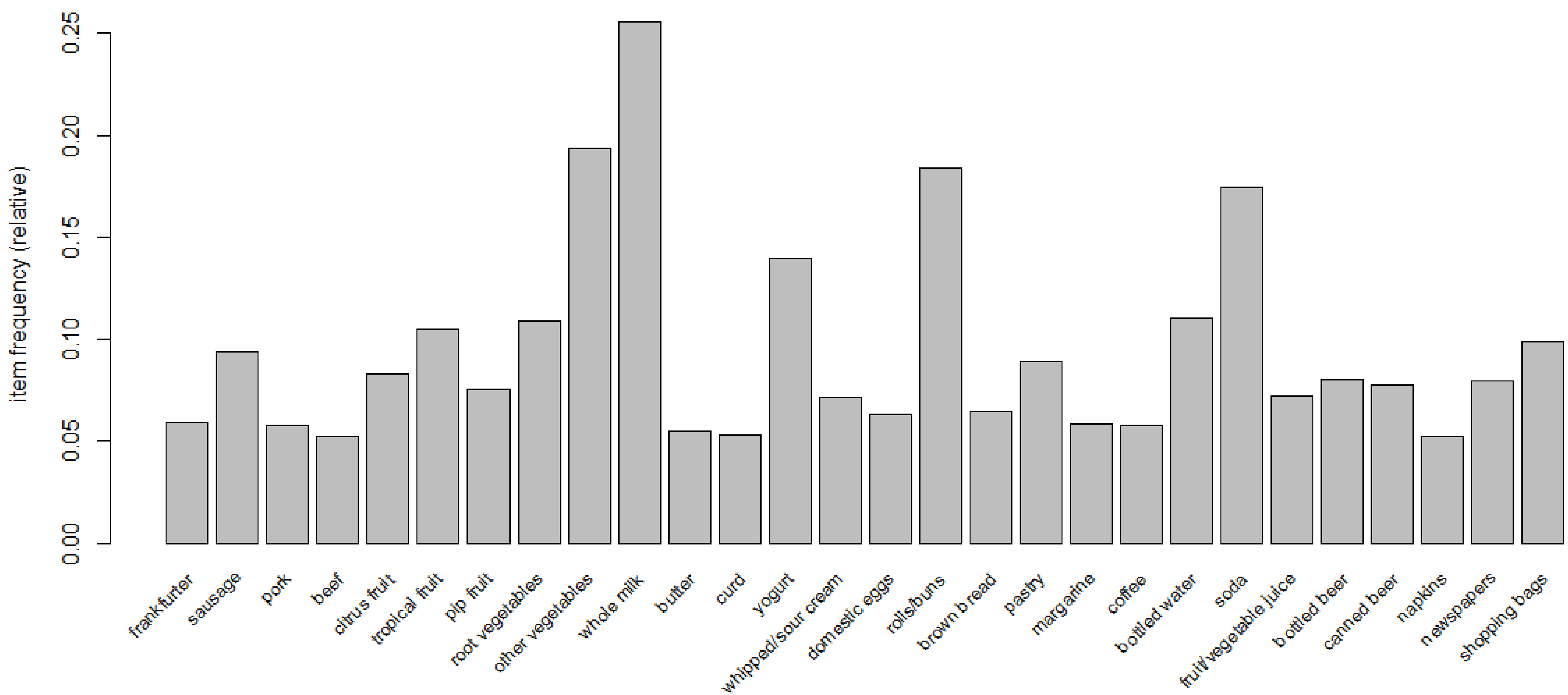
```
# Item inspection
itemName <- itemLabels(Groceries)
itemCount <- itemFrequency(Groceries)*nrow(Groceries)
col <- brewer.pal(8, "Dark2")
wordcloud(words = itemName, freq = itemCount, min.freq = 1,
         scale = c(3, 0.2), col = col, random.order = FALSE)
```

R Exercise: Market Basket Analysis

- Item frequency plot

```
itemFrequencyPlot(Groceries, support = 0.05, cex.names=0.8)
```

✓ Items with the frequency greater than 0.05 are displayed



R Exercise: Market Basket Analysis

- Data Preprocessing
  - ✓ Categorize a numeric variable, remove NA, etc.

```r
# Remove "Name" column and group "Age" column
titanic_ar <- titanic[,2:5]
titanic_ar$Age = as.character(titanic_ar$Age)
c_idx <- which(as.numeric(titanic_ar$Age) < 20)
a_idx <- which(as.numeric(titanic_ar$Age) >= 20)
na_idx <- which(is.na(titanic_ar$Age))

titanic_ar$Age[c_idx] <- "Child"
titanic_ar$Age[a_idx] <- "Adult"
titanic_ar$Age[na_idx] <- "Unknown"

# Convert the attribues to factor
titanic_ar$Age <- as.factor(titanic_ar$Age)
titanic_ar$Survived <- as.factor(titanic_ar$Survived)
```

	PClass	Age	Sex	Survived
1	1st	Adult	female	1
2	1st	Child	female	0
3	1st	Adult	male	0
4	1st	Adult	female	0
5	1st	Child	male	1
6	1st	Adult	male	1
7	1st	Adult	female	1
8	1st	Adult	male	0
9	1st	Adult	female	1
10	1st	Adult	male	0

R Exercise: Market Basket Analysis

- Association rule generation

```r
# Rule generation by Apriori
rules <- apriori(Groceries, parameter=list(support=0.01, confidence=0.35))

# Check the generated rules
inspect(rules)

# List the first three rules with the highest lift values
inspect(sort(rules, by="lift"))
```

✓ Directions insider the inspect( ) function means that the rules are displayed in an descending order

```
> inspect(sort(rules, by="lift"))
     lhs                                     rhs                support    confidence lift     count
[1]  {citrus fruit,other vegetables}      => {root vegetables}  0.01037112 0.3591549  3.295045 102
[2]  {citrus fruit,root vegetables}       => {other vegetables} 0.01037112 0.5862069  3.029608 102
[3]  {tropical fruit,root vegetables}     => {other vegetables} 0.01230300 0.5845411  3.020999 121
[4]  {whole milk,curd}                    => {yogurt}           0.01006609 0.3852140  2.761356  99
[5]  {root vegetables,rolls/buns}         => {other vegetables} 0.01220132 0.5020921  2.594890 120
[6]  {root vegetables,yogurt}             => {other vegetables} 0.01291307 0.5000000  2.584078 127
[7]  {tropical fruit,whole milk}          => {yogurt}           0.01514997 0.3581731  2.567516 149
[8]  {yogurt,whipped/sour cream}          => {other vegetables} 0.01016777 0.4901961  2.533410 100
[9]  {other vegetables,whipped/sour cream} => {yogurt}          0.01016777 0.3521127  2.524073 100
[10] {root vegetables,whole milk}         => {other vegetables} 0.02318251 0.4740125  2.449770 228
```

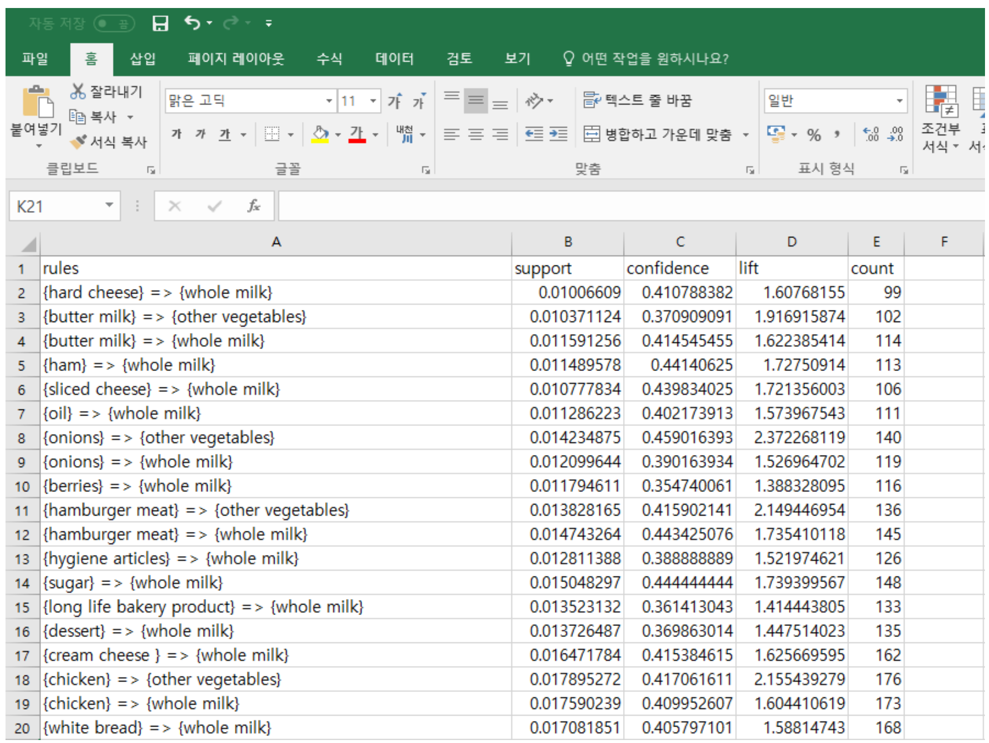
R Exercise: Market Basket Analysis

- Export the generated rules

```r
# Save the rules in a text file
write.csv(as(rules, "data.frame"), "Groceries_rules.csv", row.names = FALSE)
```

✓ Convert the generated rules to data.frame format and export it as a csv file

▪ MS Excel file format (ex: xlsx) is not recommended (too slow!)



	A	B	C	D	E
1	rules	support	confidence	lift	count
2	{hard cheese} => {whole milk}	0.01006609	0.410788382	1.60768155	99
3	{butter milk} => {other vegetables}	0.010371124	0.370909091	1.916915874	102
4	{butter milk} => {whole milk}	0.011591256	0.414545455	1.622385414	114
5	{ham} => {whole milk}	0.011489578	0.44140625	1.72750914	113
6	{sliced cheese} => {whole milk}	0.010777834	0.439834025	1.721356003	106
7	{oil} => {whole milk}	0.011286223	0.402173913	1.573967543	111
8	{onions} => {other vegetables}	0.014234875	0.459016393	2.372268119	140
9	{onions} => {whole milk}	0.012099644	0.390163934	1.526964702	119
10	{berries} => {whole milk}	0.011794611	0.354740061	1.388328095	116
11	{hamburger meat} => {other vegetables}	0.013828165	0.415902141	2.149446954	136
12	{hamburger meat} => {whole milk}	0.014743264	0.443425076	1.735410118	145
13	{hygiene articles} => {whole milk}	0.012811388	0.388888889	1.521974621	126
14	{sugar} => {whole milk}	0.015048297	0.444444444	1.739399567	148
15	{long life bakery product} => {whole milk}	0.013523132	0.361413043	1.414443805	133
16	{dessert} => {whole milk}	0.013726487	0.369863014	1.447514023	135
17	{cream cheese } => {whole milk}	0.016471784	0.415384615	1.625669595	162
18	{chicken} => {other vegetables}	0.017895272	0.417061611	2.155439279	176
19	{chicken} => {whole milk}	0.017590239	0.409952607	1.604410619	173
20	{white bread} => {whole milk}	0.017081851	0.405797101	1.58814743	168

R Exercise: Market Basket Analysis

- Draw plots for the generated rules

```
# Plot the rules
plot(rules, method = "scatterplot")
plotly_arules(rules, method = "scatterplot", measure = c("support", "confidence"),
          shading = "lift")
```

✓ plot( ) function generates a fixed plot

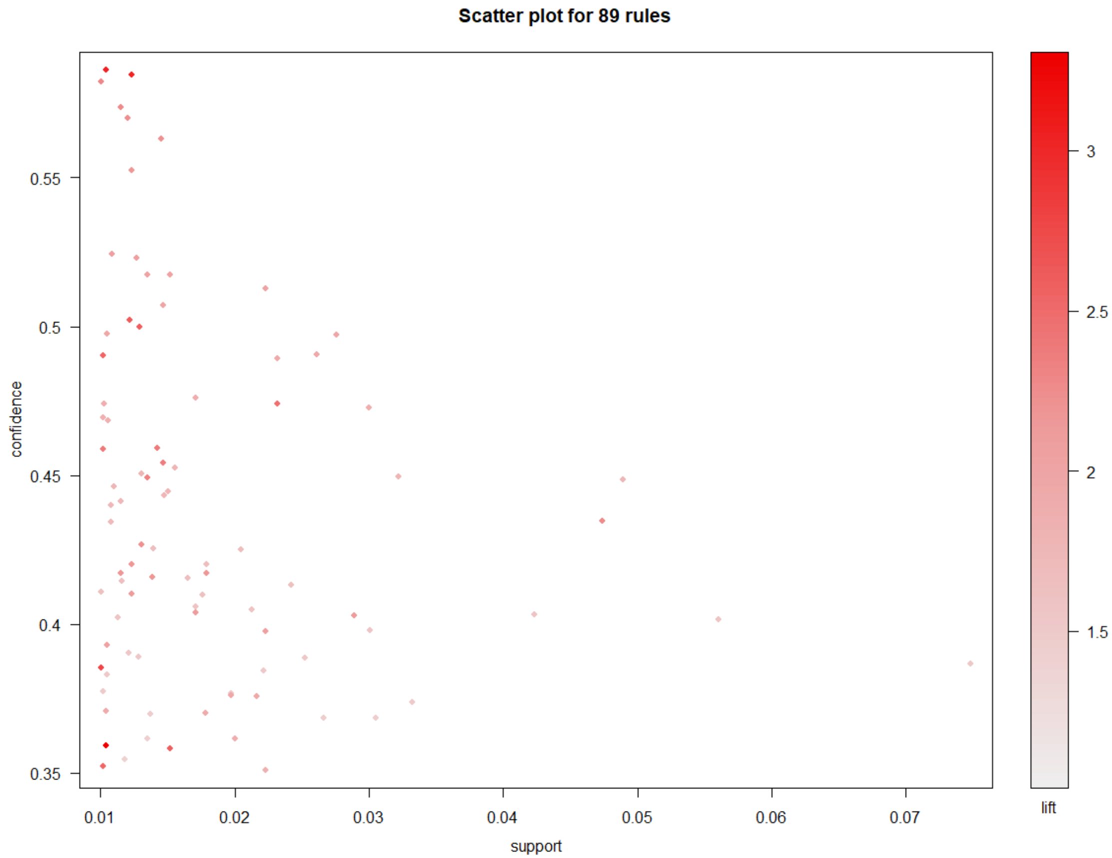
  - Different formats are available (ex: scatterplot, matrix, graph) using the "method" option

✓ ploty_arules( ) function generates an interactive plot

  - Users can adjust the axis, zoom in/out, etc.
  - This function is now deprecated but still can be used

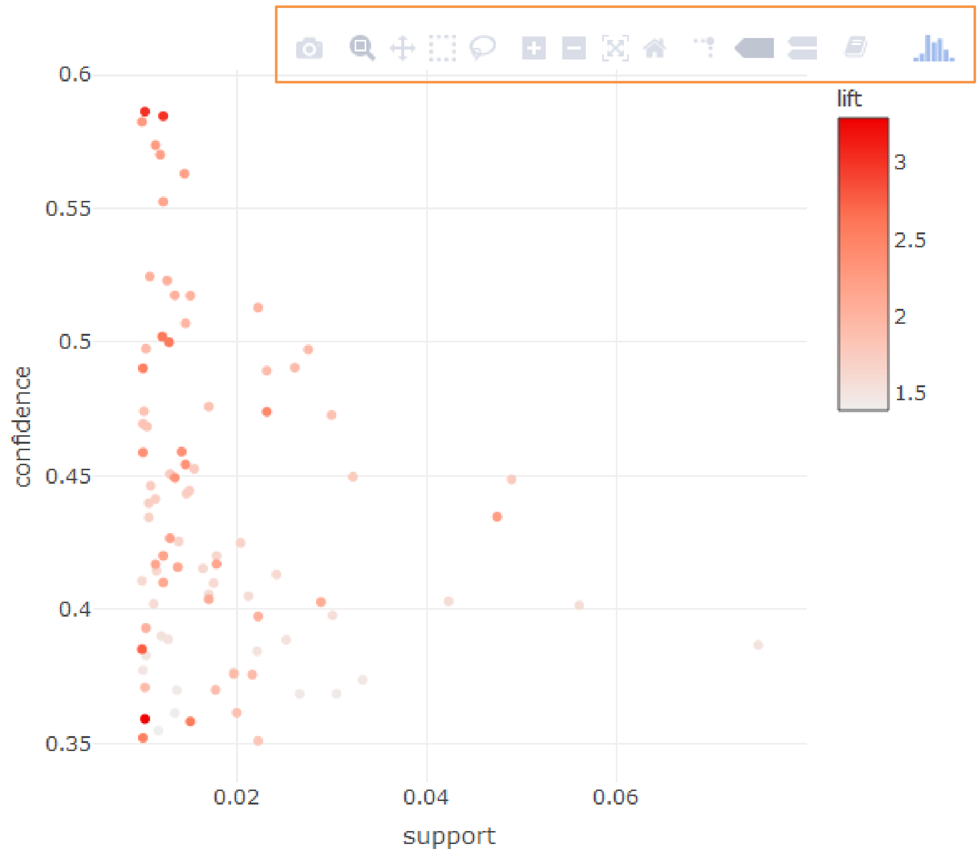
R Exercise: Market Basket Analysis

- Draw plots for the generated rules
  - ✓ plot( ) function (method = "scatterplot")
  - ✓ Used to understand the distribution of generated rules (not for interpreting individual rules)



R Exercise: Market Basket Analysis

- Draw plots for the generated rules
  - ✓ plotly_arules( ) function (method = “scatterplot”)
  - ✓ You can adjusts some options

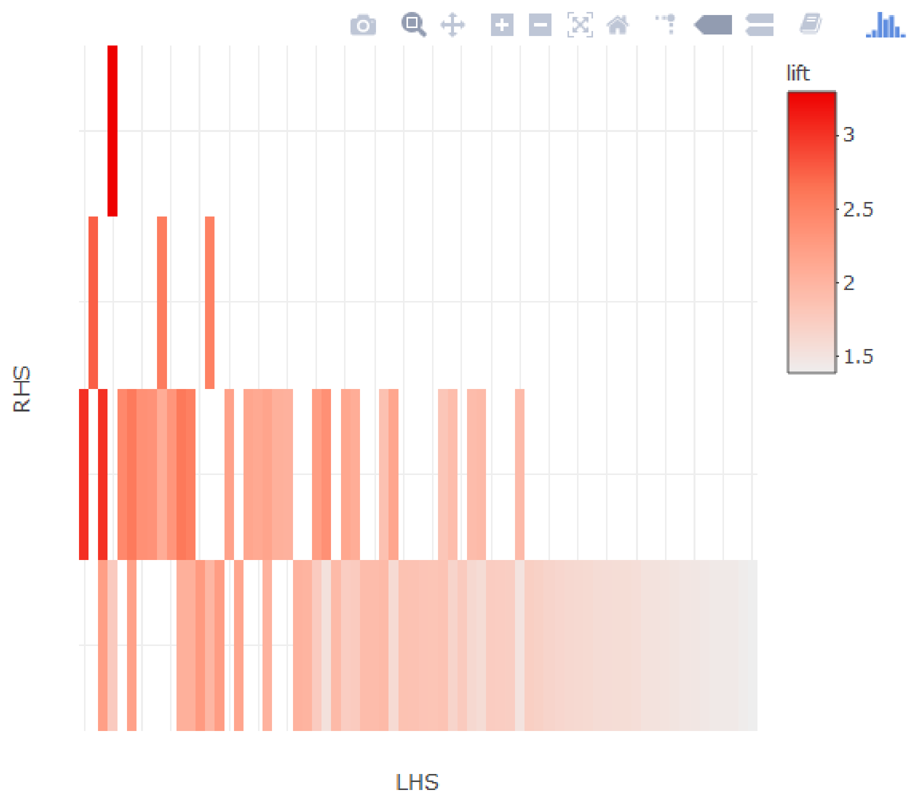


R Exercise: Market Basket Analysis

- Draw plots for the generated rules

```
plot(rules, method="matrix")
plotly_arules(rules, method = "matrix", measure = c("support", "confidence"),
          shading = "lift")
```

✓ method = “matrix”



R Exercise: Market Basket Analysis

- Change options to generate fewer rules

```
# Rule generation by Apriori with another parameters
rules <- apriori(Groceries, parameter=list(support=0.01, confidence=0.5))
plot(rules, method="graph")
plot(rules, method="paracoord")
```

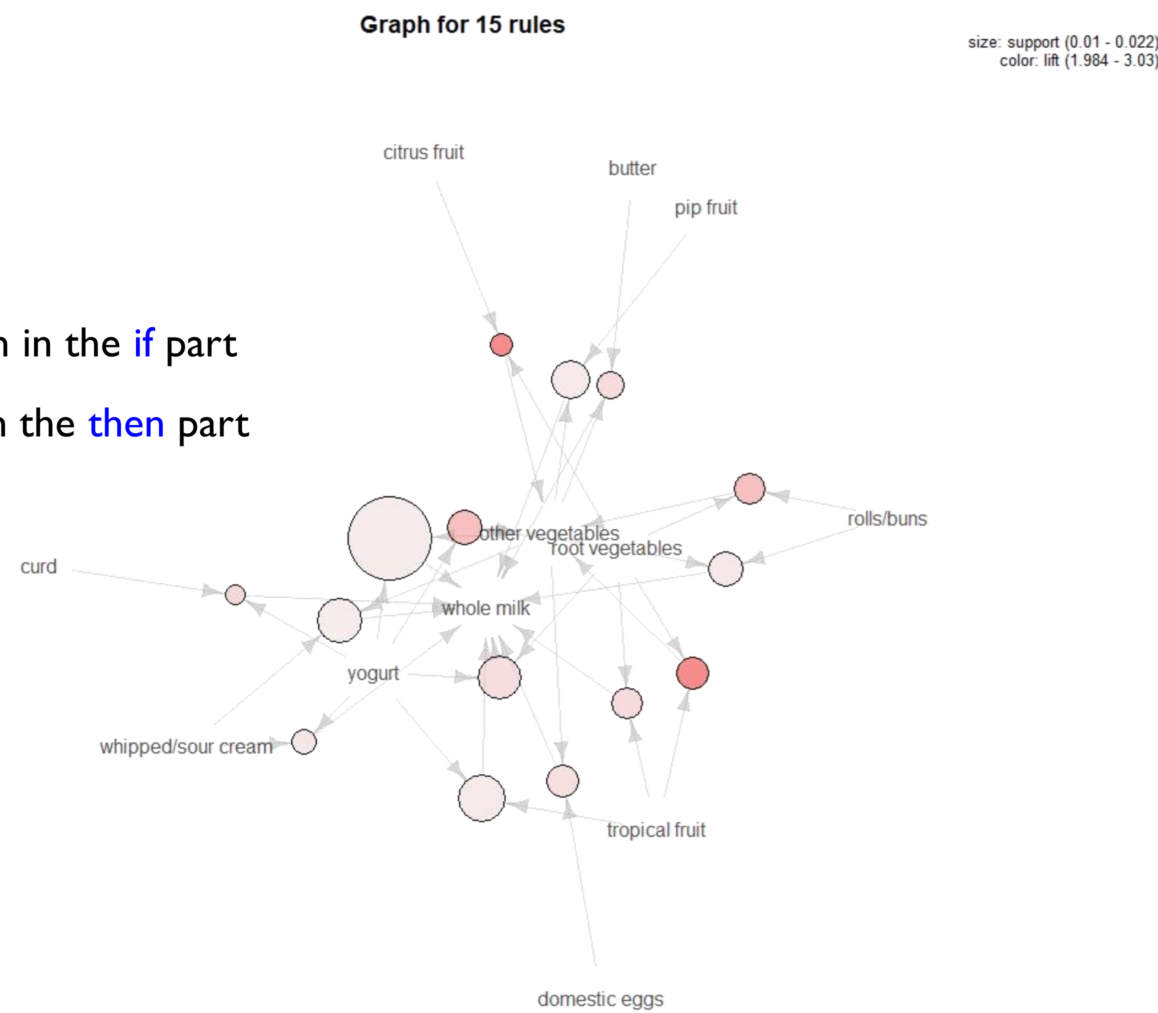
✓ Increase the confidence cut-off from 0.35 to 0.5

- 89 rules are reduced to 15 rules

✓ "graph" method and "paracoord" method can display the rules focusing on the relationship between the items in the generated rules

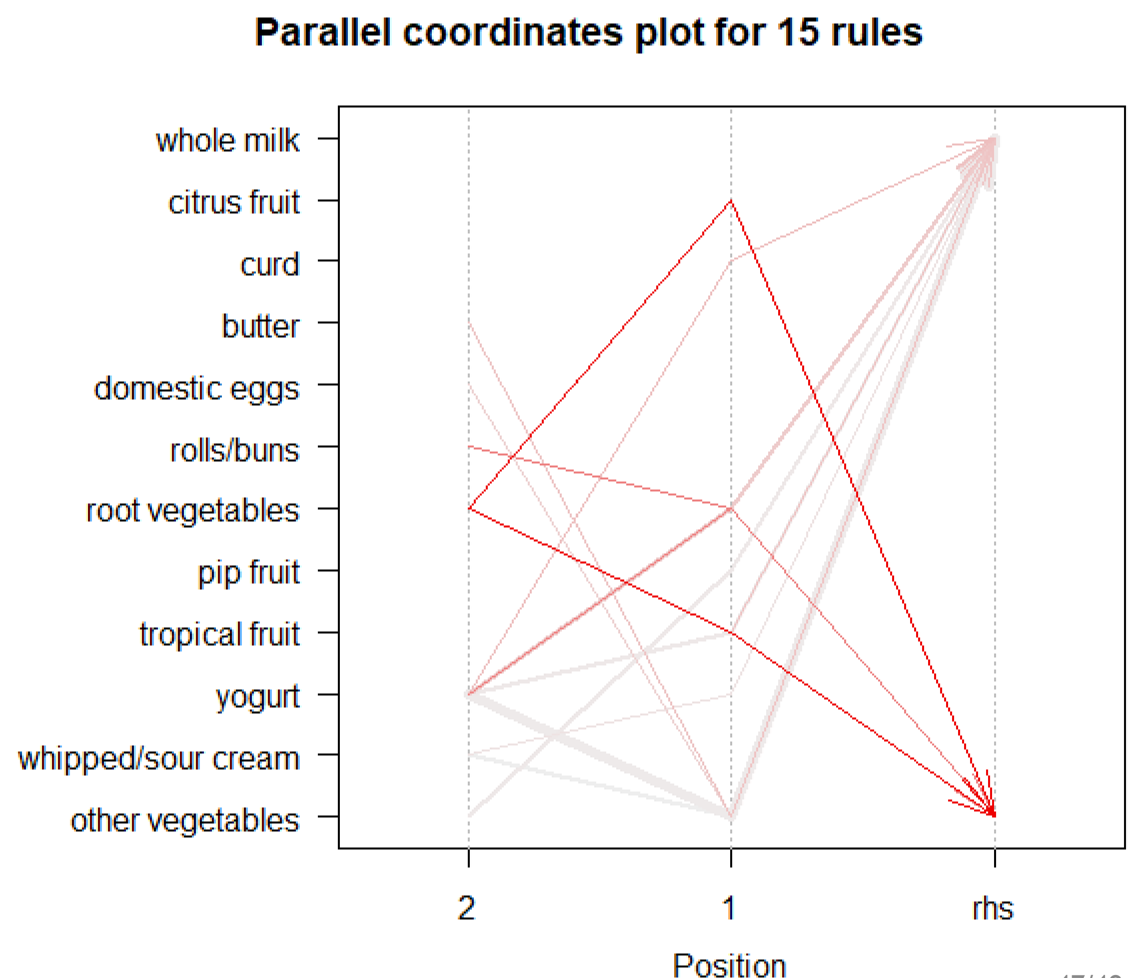
R Exercise: Market Basket Analysis

- Change options to generate fewer rules
  - ✓ "graph" method
    - Circle: rule
    - Circle size: support
    - Circle color: lift
    - Arrow from the circle: Item in the if part
    - Arrow to the circle: Item in the then part



R Exercise: Market Basket Analysis

- Change options to generate fewer rules
  - ✓ "paracoord" method
    - Line: rule
    - x-axis: item sequence
    - y-axsis: item name




ANY
questions?