



## ABF 7강: Financial Consumer Analytics

서강대학교, 경영전문대학원/경영학과 교수 이군희

# Agenda

- Financial Consumer Data Analytics
- 사례1: 고객 행동 데이터 분석
- Association Rules
- MDS 기법
- 사례2: 금융상품 관계분석



# Financial Consumer Data Analytics

- 금융회사가 향상된 신용능력을 측정하고 보다 나은 금융 서비스 (금융회사의 이익?)를 제공할 목적으로 기존 금융고객에 대한 과거 이력 데이터를 분석하는 것.
- 주요 행동 데이터
  - 일시불/할부, 이자/원금 상환 등에 대한 이력 (연체 횟수, 단기 연체 여부 등...)
  - 소비자의 구매 자료
  - 금융관련 정보 (계좌의 평균 잔액, 거래 금액, 거래 종류, 거래 횟수, 한도 소진율 등 ...)
- 활용 목적
  - 카드 신규 발급/ 신규 대출/ 대출 연장에 대한 빠른 승인
  - 대출이나 신용카드의 사전 승인
  - 특정 금융 상품에 대한 targeted marketing, Profit Scoring
  - 대출 이자율 설정과 같은 보다 정확한 Risk Based Pricing을 위하여
  - 효율적 Credit Portfolio Management 를 위하여
    - ✓ 고객 세그멘테이션 (Pooling)
    - ✓ IFRS9 에 따른 대손충당금 설정의 정확도를 높이기 위하여



# 사례1: 고객 행동 데이터 분석

*Creditcard.xlsx* 자료 다운  
<http://naver.me/x1X7shU2>

A 카드사의 고객 행동 데이터 (Consumer Behavior Data)

- 학습 목적으로 자료에 수정을 가하여 만들어진 가상 자료.

• 13972 명 고객 자료 196개 변수에 대한 15개월 자료

- GI\_A1~15, (청구금액\*), GI\_B1~15, (지불금액), IL\_C1~15, (일시불\*)

(참고: 가맹점 수수료: 2.5%, 자본조달비용: 연10%)

- HAL\_C1~15, (할부\*), (참고: 가맹점 수수료: 2.5%, 고객 할부 수수료: 3%)

- CS\_C1~15, (현금서비스\*), (참고: 당시 이율: 연29%, 자본 조달비용: 연10%)

- LN\_C1~15, (카드론), IL\_M1~15, (일시불 미수금), HAL\_M1~15, (할부 미수금\*)

(참고: 할부 이율: 15%)

- CS\_M1~15, (현금서비스 미수금), GI\_D1~15, (청구금 연체금),

- GI\_D\_M1~15, (청구금 미수금 연체), LN\_D1~15, (카드론 연체)

- LN\_D\_M1~15, (카드론 미수금 연체), id (고객 id 번호)



# 사례1: 고객 행동 데이터 분석

- 분석 설계

고객 행동 데이터는 시간에 따라 데이터가 지속적으로 발생하기 때문에 양이 크고 한 고객이 여러 데이터를 가지고 있으므로 분석이 쉽지 않음.

- 시간과 관련된 데이터이기 때문에 종단면 분석 (longitudinal analysis)을 진행
- 분석 전문가라도 종단면 분석은 간단한 분석이 아님.
- 분석 설계: 분석가 마다 분석 방법이 다양하게 나타날 수 있음.

나의 분석 설계는 ...

- 1) GI\_A1~15, (청구금액) 을 대상으로
- 2) 한 고객의 자료를 회귀모형으로 평가를 한 후,
- 3) 모델의 회귀계수 값을 이용하여 요약
- 4) 이러한 과정을 13972 명 고객에게 적용
- 5) 고객을 세그멘테이션 한 후 특징을 파악

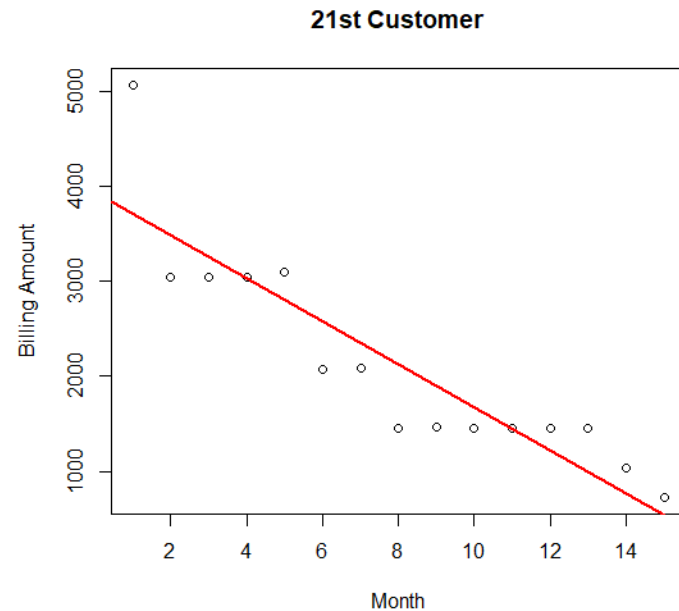


# 사례1: 고객 행동 데이터 분석

21번째 고객 자료를 분석해 보면 ...

```
Cards=read.table("clipboard", h=T)
x=1:15
y=as.numeric(Cards[21,1:15])
M1=lm(y~x)
plot(x,y, xlab="Month", ylab="Billing Amount", main="21st Customer")
abline(M1, lwd=2, col="red")
```

- ✓ 카드 사용량이 지속적으로 감소 추세
- ✓ 다른 카드사로 이탈 가능성이 높음
- ✓ 높은 Intercept과 음수의 Slope을 가짐
- ✓ 고객의 15개월 자료를 절편과 기울기  
2개의 변수로 요약할 수 있음



# 사례1: 고객 행동 데이터 분석

```
n=nrow(Cards)
t=data.frame(id=0, Intercept=0, Slope=0)
for(i in 1:n){
  y = as.numeric(Cards[i,1:15])
  M1=lm(y~x)
  t2=data.frame(id=i, Intercept=coef(M1)[1], Slope=coef(M1)[2])
  t=rbind(t,t2)
}
t = t[-1,]
summary(t[,2:3])
```

| Intercept        | Slope            |
|------------------|------------------|
| Min. : -2679.543 | Min. : -547.021  |
| 1st Qu.: 0.876   | 1st Qu.: -38.583 |
| Median : 149.910 | Median : -3.036  |
| Mean : 629.423   | Mean : -29.529   |
| 3rd Qu.: 780.907 | 3rd Qu.: 2.034   |
| Max. : 11885.438 | Max. : 609.518   |



## 사례1: 고객 행동 데이터 분석

```
plot(density(t[,2]), xlim=c(-1000,5000), lwd=2, main="Intercept Density Plot")
d1=density(t[t[,3]>0,2])
d2=density(t[t[,3]<0,2])
lines(d1,col="blue",lwd=2,lty=2)
lines(d2,col="red",lwd=2,lty=3)
legend("topright",legend=c("Density","Density with Positive Slope","Density with Negative Slope"),
      col=c("black","blue","red"),lty=1:3)
```

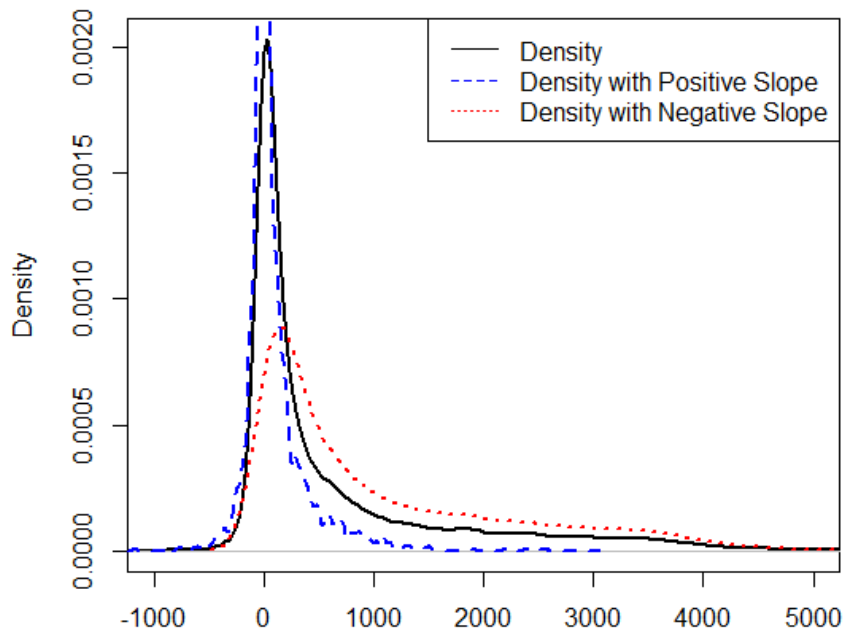
```
plot(density(t[,3]), xlim=c(-400,200), lwd=2, main="Slope Density Plot")
d1=density(t[t[,2]>0,3])
d2=density(t[t[,2]<0,3])
lines(d1,col="blue",lwd=2,lty=2)
lines(d2,col="red",lwd=2,lty=3)
legend("topleft",legend=c("Density","Density with Positive Intercept","Density with Negative Intercept"), col=c("black","blue","red"),lty=1:3)
```





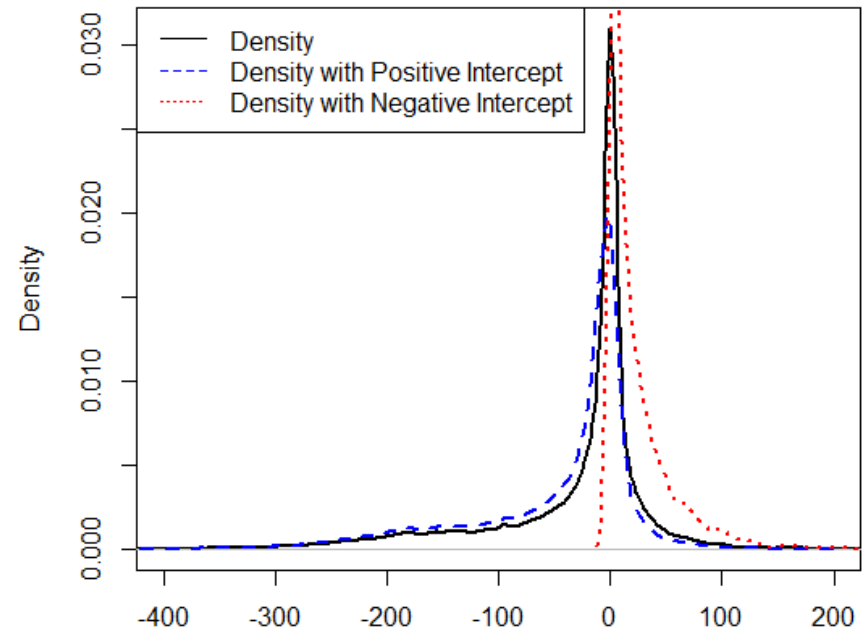
# 사례1: 고객 행동 데이터 분석

Intercept Density Plot



N = 13872 Bandwidth = 77.77

Slope Density Plot



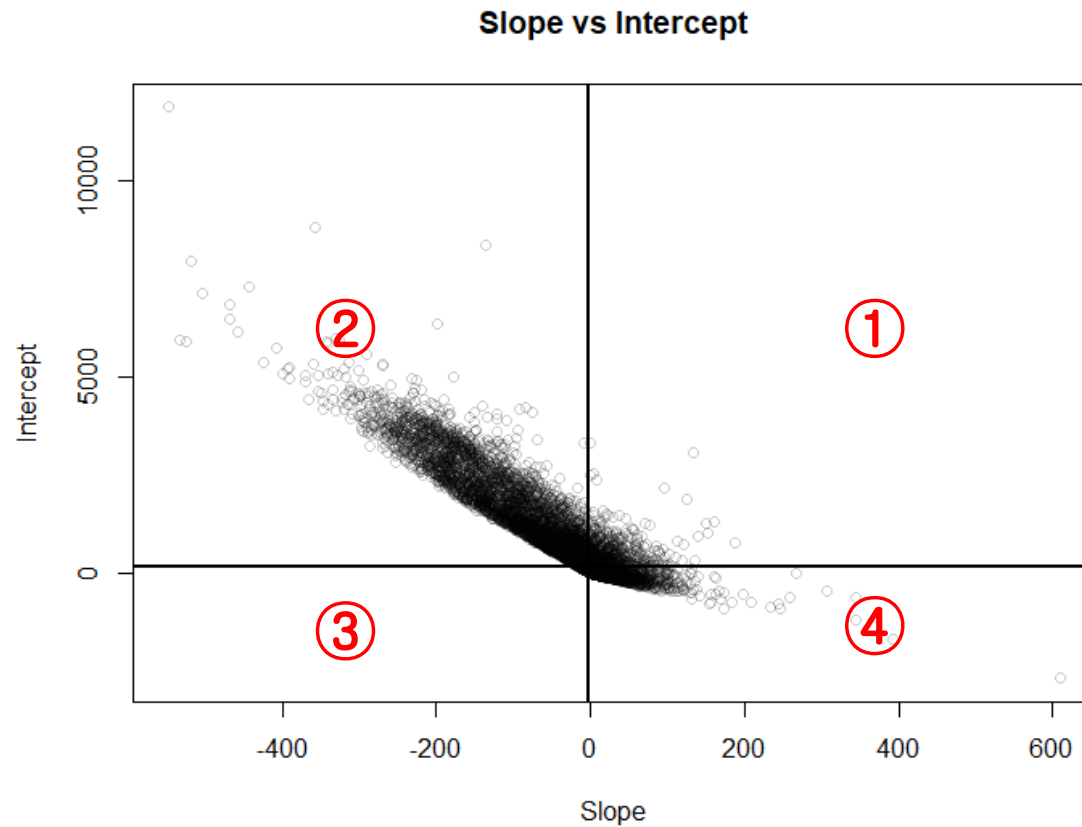
N = 13872 Bandwidth = 4.05

절편이 크면 음수의 기울기를 갖는 성향이 있고,  
절편이 작으면 양수의 기울기를 갖는 성향이 있음.  
어느 경우가 좋은 것인가?



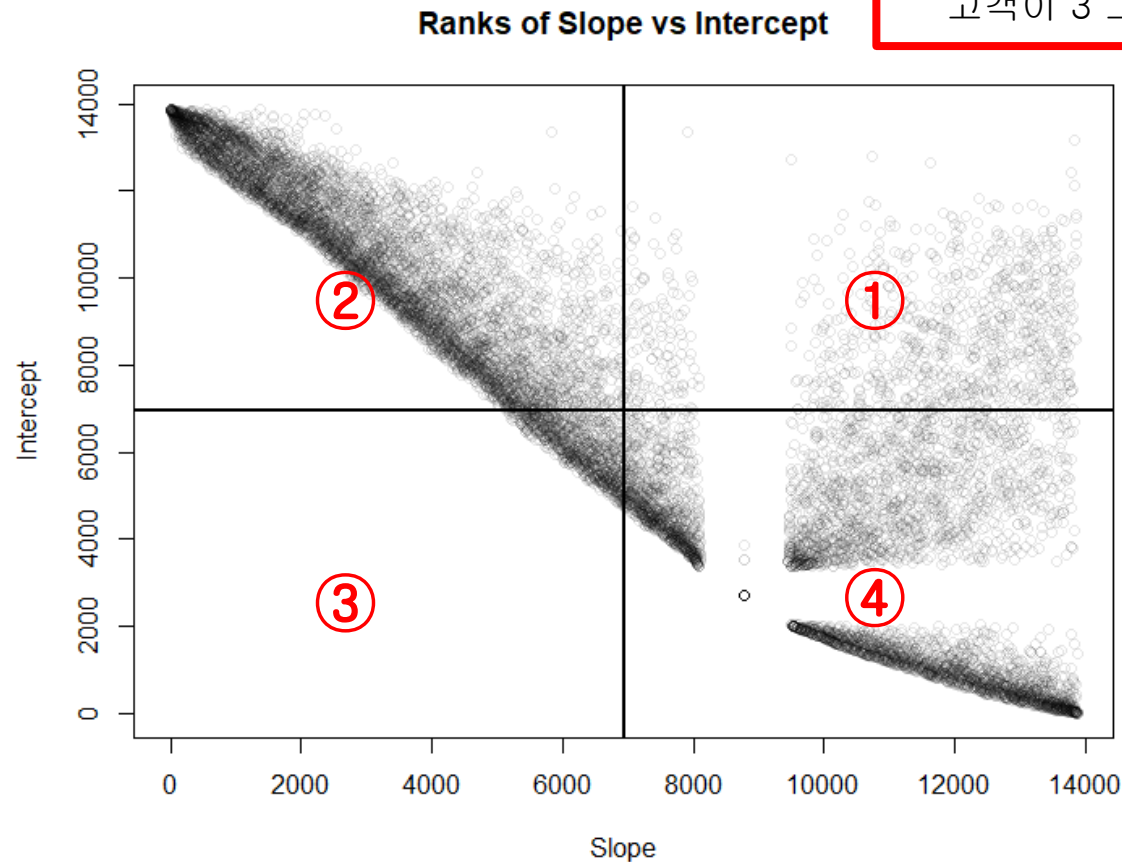
## 사례1: 고객 행동 데이터 분석

```
library(scales)
plot(t[,3], t[,2], col=alpha('black', alpha=0.2), xlab="Slope", ylab="Intercept", main="Slope vs Intercept")
abline(h=150, lwd=2); abline(v=-3, lwd=2)
```



# 사례1: 고객 행동 데이터 분석

```
t$Rint=rank(t[,2])
t$Rslp=rank(t[,3])
plot(t[,5], t[,4], col=alpha('black', alpha=0.1), xlab="Slope", ylab="Intercept", main="Ranks of Slope vs Intercept")
abline(h=13872/2, lwd=2); abline(v=13872/2, lwd=2)
```



고객이 3 그룹으로 구분



# Association Rules

일명 Market Basket Analysis (장바구니 분석) 이라고 부름.

(Example)

$Pr(\text{우유}) = 10\%$  (즉, 100명 방문객 중 10명 꼴로 우유를 구입)

➔ Support (지지도)

$Pr(\text{빵}|\text{우유}) = 71\%$  (조건부 확률, 우유를 구입한 고객 중 빵을 구입한 고객이 71%)

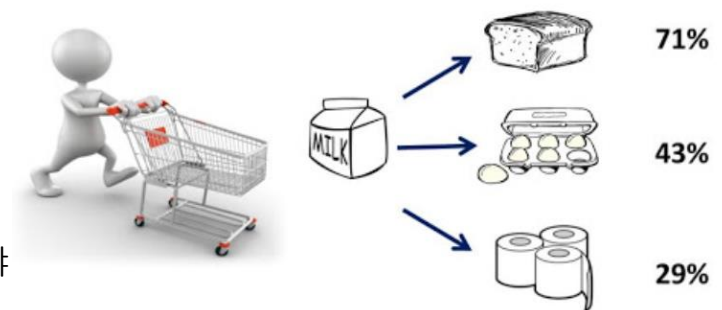
➔ Confidence (신뢰도) 우유는 선행사건으로 LHS (좌변), 빵은 결과로 RHS (우변)

$Pr(\text{빵}|\text{우유}) / Pr(\text{빵}) * 100 = 71\% / 20\% * 100 = 355$  ※  $Pr(\text{빵})$ 이 20%라고 가정

➔ 우유를 구입한 고객은 빵을 살 확률이 3.55배 증가

➔ Lift (리프트)

- ✓ 빵과 우유는 매우 밀접한 관계가 있으며,
- ✓ Cross-Selling 대상
- ✓ Support 와 Confidence가 어느 정도 값을 가져야
- ✓ Lift 값이 의미가 있음.



Of transactions that included milk:

- 71% included bread
- 43% included eggs
- 29% included toilet paper



# Association Rules

```
> library(arules)
> library(arulesViz)
> tran = read.transactions("clipboard", format="basket", sep=",")
> tran
```

transactions in sparse format with

6 transactions (rows) and

5 items (columns)

```
> inspect(tran)
```

items

[1] {라면,맥주,우유}

[2] {고기,라면,우유}

[3] {고기,과일,라면}

[4] {고기,맥주,우유}

[5] {고기,라면,우유}

[6] {과일,우유}

```
> summary(tran)
```

transactions as itemMatrix in sparse format with  
6 rows (elements/itemsets/transactions) and  
5 columns (items) and a density of 0.5666667

most frequent items:

|  | 우유 | 고기 | 라면 | 과일 | 맥주 (Other) |
|--|----|----|----|----|------------|
|  | 5  | 4  | 4  | 2  | 2          |

element (itemset/transaction) length distribution:  
sizes

2 3

1 5

|  | Min.  | 1st Qu. | Median | Mean  | 3rd Qu. | Max.  |
|--|-------|---------|--------|-------|---------|-------|
|  | 2.000 | 3.000   | 3.000  | 2.833 | 3.000   | 3.000 |

includes extended item information – examples:

labels

1 고기

2 과일

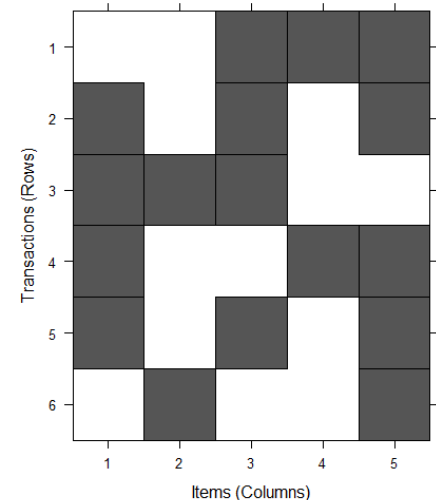
3 라면

```
> image(tran)
```

다음과 같은 transaction  
데이터가 있다고 하자.

라면,맥주,우유  
라면,고기,우유  
라면,과일,고기  
고기,맥주,우유  
라면,고기,우유  
과일,우유

고기 과일 라면 맥주 우유



# Association Rules

```
> itemFrequencyPlot(tran, topN=5)
> Rules=apriori(tran, parameter=list(supp=0.3, conf=0.1, maxlen=3))
> Rules
set of 16 rules
> options(digits=2)
> Rules=sort(Rules, by='lift', decreasing=TRUE)
> inspect(Rules[1:4])
```

|     | lhs  | rhs     | support | confidence | coverage | lift | count |
|-----|------|---------|---------|------------|----------|------|-------|
| [1] | {우유} | => {맥주} | 0.33    | 0.40       | 0.83     | 1.2  | 2     |
| [2] | {맥주} | => {우유} | 0.33    | 1.00       | 0.33     | 1.2  | 2     |
| [3] | {고기} | => {라면} | 0.50    | 0.75       | 0.67     | 1.1  | 3     |
| [4] | {라면} | => {고기} | 0.50    | 0.75       | 0.67     | 1.1  | 3     |

```
> T=crossTable(tran)
```

```
> T
```

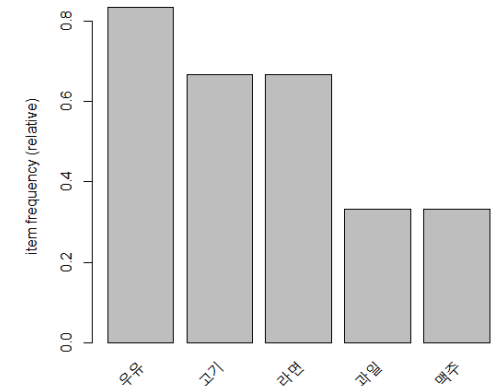
|    | 고기 | 과일 | 라면 | 맥주 | 우유 |
|----|----|----|----|----|----|
| 고기 | 4  | 1  | 3  | 1  | 3  |
| 과일 | 1  | 2  | 1  | 0  | 1  |
| 라면 | 3  | 1  | 4  | 1  | 3  |
| 맥주 | 1  | 0  | 1  | 2  | 2  |
| 우유 | 3  | 1  | 3  | 2  | 5  |

```
> T["라면","라면"]
```

```
[1] 4
```

```
> T["라면","우유"]
```

```
[1] 3
```



```
> Rules1=apriori(tran, parameter=list(supp=0.3, conf=0.1, maxlen=3),
appearance=list(default="lhs", rhs="라면"))
```

```
> Rules1
```

```
set of 4 rules
```

```
> Rules1=sort(Rules1, by='lift', decreasing=TRUE)
```

```
> inspect(Rules1)
```

|     | lhs     | rhs     | support | confidence | coverage | lift | count |
|-----|---------|---------|---------|------------|----------|------|-------|
| [1] | {고기}    | => {라면} | 0.50    | 0.75       | 0.67     | 1.1  | 3     |
| [2] | {}      | => {라면} | 0.67    | 0.67       | 1.00     | 1.0  | 4     |
| [3] | {고기,우유} | => {라면} | 0.33    | 0.67       | 0.50     | 1.0  | 2     |
| [4] | {우유}    | => {라면} | 0.50    | 0.60       | 0.83     | 0.9  | 3     |



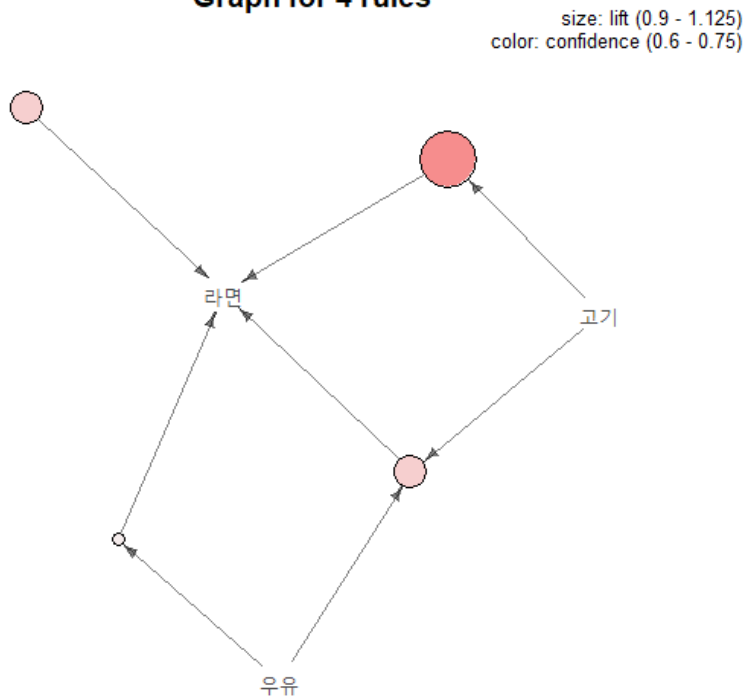
# Association Rules

```

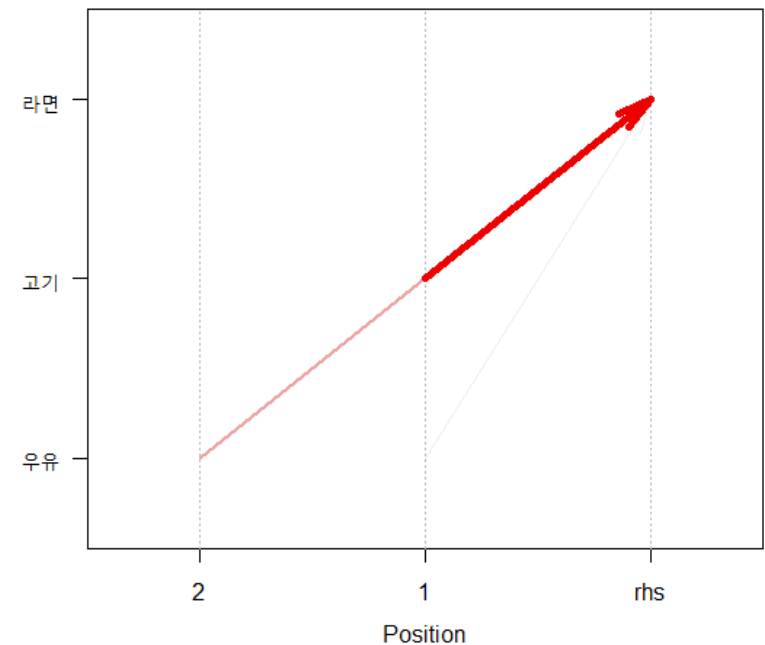
set.seed(123457)
plot(Rules1, method="graph", measure="lift", shading="confidence", edgeCol="black")
plot(Rules1, method="scatterplot", measure="lift", shading="confidence")
plot(Rules1, measure=c("support", "lift"), shading="confidence")
plot(Rules1, method="paracoord", measure="lift", shading="confidence")
plot(Rules1, method="grouped", measure="lift", shading="confidence")
plot(Rules1, method="two-key plot", measure="lift", shading="confidence")

```

**Graph for 4 rules**



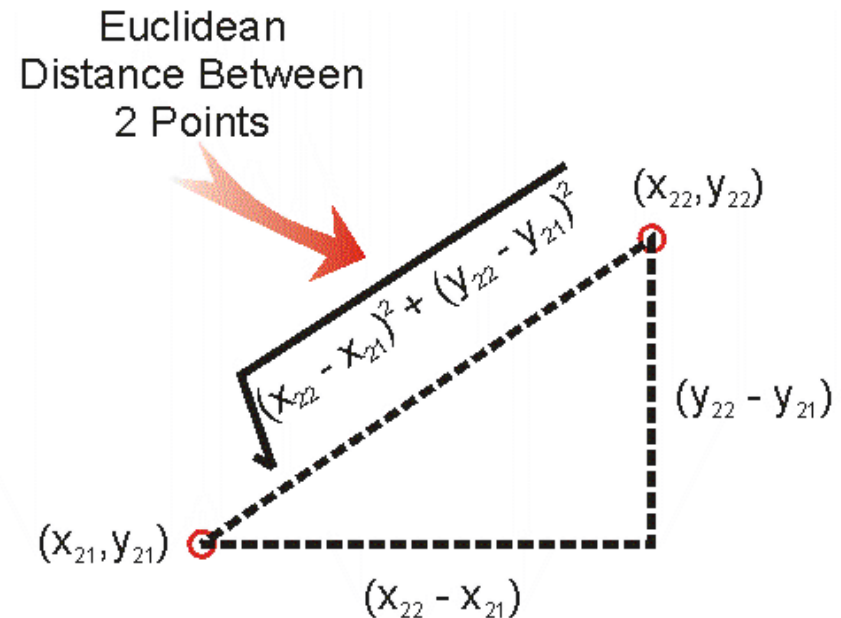
**Parallel coordinates plot for 3 rules**



# MDS 기법

MDS: (다차원 척도법, Multi-Dimensional Scaling)

- 자료들 사이의 거리를 측정하여 2차원 또는 3차원 공간상에 점으로 표현하는 방법.
- 즉, 자료들간의 근접성(proximity)을 시각화하는 기법으로
- 이를 통하여 데이터 속에 잠재해 있는 패턴이나 구조를 찾을 수 있다.
- 자료들간의 거리 계산은 일반적으로 유클리드 거리 행렬을 주로 사용한다.
- 상대적 거리의 정확도는 스트레스 값(stress value)으로 표현된다.





# MDS 기법

(예제) 우리나라 주요 도시간 거리 (Km)

*DistInKorea.xlsx* 파일 다운  
<http://naver.me/x6ibFA4E>

```
Distance=read.table("clipboard", header=TRUE, row.names=1)
```

```
mds1 = cmdscale(Distance, k = 2)
```

```
plot(mds1[,1], mds1[,2], type = "n", xlab = "", ylab = "", asp=1, main = "Cities in Korea")
```

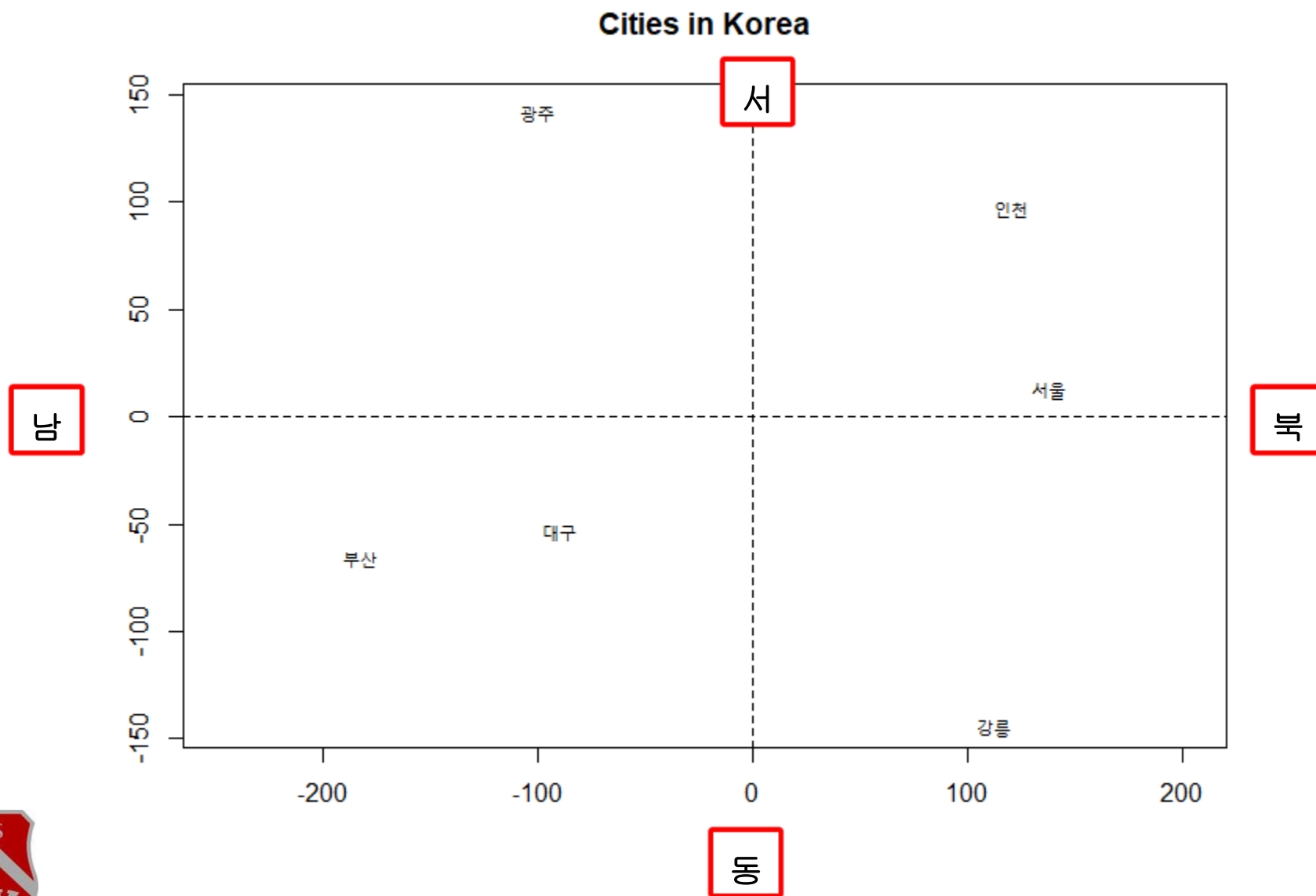
```
text(mds1[,1], mds1[,2], rownames(Distance), cex = 0.8)
```

```
abline(v = 0, h = 0, lty = 2, lwd = 1)
```

|    | 서울  | 부산  | 광주  | 대구  | 인천  | 강릉  |
|----|-----|-----|-----|-----|-----|-----|
| 서울 | 0   | 325 | 270 | 242 | 40  | 170 |
| 부산 | 325 | 0   | 195 | 95  | 344 | 296 |
| 광주 | 270 | 195 | 0   | 170 | 165 | 355 |
| 대구 | 242 | 95  | 170 | 0   | 250 | 210 |
| 인천 | 40  | 344 | 165 | 250 | 0   | 207 |
| 강릉 | 170 | 296 | 355 | 210 | 207 | 0   |



# MDS 기법



## 사례2: 금융상품 관계분석

*Bank1.csv 자료 다운*  
<http://naver.me/G5uqY5uf>

A 은행의 금융상품 자료 (1230명 고객자료)

- Ordinary: 보통예금 계좌 고객
- Loans: 대출상품 사용 고객
- Credit Cards: 신용카드 고객
- MMF: MMF 계좌 고객
- Installment: 정기적금 보유 고객

목적: 신용카드를 가장 잘 사용할 것 같은 고객 확인

- ➔ Credit Cards와 가장 연관성이 높은 금융상품을 확인하여, 해당 상품에 가입되어 있으면서 신용카드를 보유하지 않고 있는 고객에게 신용카드에 대한 프로모션 진행

분석방법: Association Rule (Market Basket Analysis)



## 사례2: 금융상품 관계분석

```
> Bank1 = read.transactions("clipboard", format="basket")
```

```
> summary(Bank1)
```

transactions as itemMatrix in sparse format with  
1230 rows (elements/itemsets/transactions) and  
5 columns (items) and a density of 0.26

most frequent items:

| Installment | Ordinary | Loans | CreditCard | MMF | (Other) |
|-------------|----------|-------|------------|-----|---------|
| 396         | 392      | 358   | 242        | 188 | 0       |

...

includes extended item information – examples:

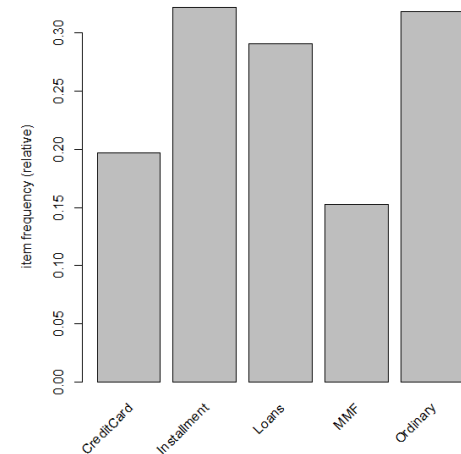
labels

1 CreditCard

2 Installment

3 Loans

```
> itemFrequencyPlot(Bank1)
```



```
> Rules=apriori(Bank1, parameter=list(supp=0.01, conf=0.05, minlen=2, maxlen=3))
```

```
> Rules
```

set of 23 rules

```
> options(digits=2)
```

```
> Rules=sort(Rules, by='lift', decreasing=TRUE)
```

```
> inspect(Rules[1:5])
```

|     | lhs                   | rhs             | support | confidence | coverage | lift | count |
|-----|-----------------------|-----------------|---------|------------|----------|------|-------|
| [1] | {CreditCard,Ordinary} | => {MMF}        | 0.039   | 0.56       | 0.070    | 3.7  | 48    |
| [2] | {MMF,Ordinary}        | => {CreditCard} | 0.039   | 0.67       | 0.059    | 3.4  | 48    |
| [3] | {CreditCard,MMF}      | => {Ordinary}   | 0.039   | 0.83       | 0.047    | 2.6  | 48    |
| [4] | {Loans,Ordinary}      | => {CreditCard} | 0.015   | 0.35       | 0.042    | 1.8  | 18    |
| [5] | {MMF}                 | => {CreditCard} | 0.047   | 0.31       | 0.153    | 1.6  | 58    |



## 사례2: 금융상품 관계분석

```
> Rules1=apriori(Bank1, parameter=list(supp=0.01, conf=0.01, minlen=2, maxlen=3),
  appearance=list(default="lhs", rhs="CreditCard"))
```

```
> Rules1
```

```
set of 6 rules
```

```
> Rules1=sort(Rules1, by='lift', decreasing=TRUE)
```

```
> inspect(Rules1)
```

|     | lhs              | rhs             | support | confidence | coverage | lift | count |
|-----|------------------|-----------------|---------|------------|----------|------|-------|
| [1] | {MMF,Ordinary}   | => {CreditCard} | 0.039   | 0.67       | 0.059    | 3.39 | 48    |
| [2] | {Loans,Ordinary} | => {CreditCard} | 0.015   | 0.35       | 0.042    | 1.76 | 18    |
| [3] | {MMF}            | => {CreditCard} | 0.047   | 0.31       | 0.153    | 1.57 | 58    |
| [4] | {Ordinary}       | => {CreditCard} | 0.070   | 0.22       | 0.319    | 1.12 | 86    |
| [5] | {Loans}          | => {CreditCard} | 0.033   | 0.11       | 0.291    | 0.57 | 40    |
| [6] | {Installment}    | => {CreditCard} | 0.013   | 0.04       | 0.322    | 0.21 | 16    |

```
set.seed(123457)
```

```
plot(Rules1, method="graph", measure="lift", shading="confidence", edgeCol="black")
```

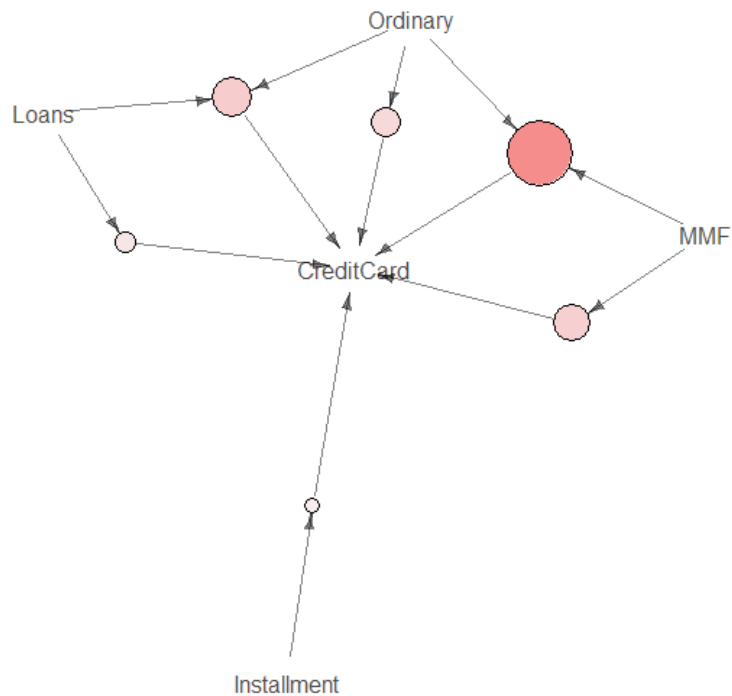
```
plot(Rules1, method="paracoord", measure="lift", shading="confidence")
```



## 사례2: 금융상품 관계분석

### Graph for 6 rules

size: lift (0.205 - 3.388)  
color: confidence (0.04 - 0.667)



### Parallel coordinates plot for 6 rules

