CRM을 위한 Big Data Analysis
- R을 이용한 데이터 마이닝 및 사례 -
Release 1 version 1.1

김경태(Eric Kim)
eric@the-ecg.com
www.the-ecg.com

May 23, 2012

Chapter 1 5
개요 5
1.1 Big Data Analysis 6
1.2 데이터 마이닝 개요 7
1.3 Data Mart 10
1.4 산업별 데이타 마이닝 적용사례 10
1.5 R의 역사 11
1.6 기초 데이터 작업 12
1.7 Missing Data Handling 14
1.8 Outlier Detection 17
1.9 Data Exploration 17
1.10 Interactive Graph 18
Chapter 2 23
분류 및 예측 모델 23
2.1 Classification 23
2.1.1 활용분야 23
2.1.2 party를 이용한 Decision Tree 25
2.1.3 rpart를 이용한 Decision Tree 28
2.1.3 Random Forrest 30
2.1.4 ROCR package를 이용한 Performance Analysis 32
2.2 Regression 34
Chapter 3 35
Clustering 35
3.1 Clustering 개요 35
3.1.1 기존 세분화 방법의 유형 35
3.1.2 전통적 세분화 방법의 문제점 35

3.1.3 Target-based 세분화 방법 36
3.1.4 Profiling 방법 36
3.1.5 세분화 수행기간 36
3.2 K-means 36
2.3 Hierarchical Clustering 38
2.4 Density-based Clustering 39
2.5 Fuzzy Clustering 42
2.6 Subspace Clustering 44
Chapter 4 45
Association Analysis 45
4.1 Association Analysis 개요 45
4.1.1 용어정의 45
4.1.2 기존 연관성 분석의 이슈 46
4.1.3 최근 연관성 분석 동향 46
4.1.4 연관성 분석 활용방안 46
4.2 Association Analysis 47
4.3 Sequence Pattern 53
4.4 Visualization 54
Chapter 5 61
Time Series Analysis 61
5.1 Time Series Analysis 개요 61
5.1.1 Forecasting이란 61
5.1.2 예측에 대한 의견 61
5.1.3 예측을 위한 데이터 준비 61
5.1.4 TCSI 분석 62
5.1.5 평가지표 62
5.1.6 예측 활용방안 62

5.2 Exercise 62

Chapter 6 68

Text Mining 68

6.1 Text Mining 68

6.2 Corpus 68

6.3 Create Term-Document Matrix 74

6.4 Dictionary 76

6.5 Sentiment Analysis 76

6.6 한글처리 81

6.7 Exercise 82

Chapter 7 86

Social Network Analysis 86

7.1 Social Network 개념 86

7.1.1 용어정의 86

7.1.2 현황 86

7.1.3 솔루션 87

7.1.4 활용방안 87

7.2 문서내 단어간의 연관성을 이용한 Social Network 87

7.3 twitter 검색을 통한 사용자간 Social Network 97

7.4 twitter 검색을 사용자 분포 그래프 100

Chapter 1

개요

"기존 Data Mining 도구는 많은 수작업과 최신 알고리즘 채택이 안되는 문제점이 많았으나, KXEN은 자동화 및 처리속도 향상으로 사용자의 니즈를 만족시켰다. 그러나 고가의 마이닝 도구 도입에 대한 부담으로 대안이 필요하게 되었다."

저자는 1986년도 부터 Statistical Software들을 사용하기 시작했다. Minitab, SPSS, SAS, Systat, RATS 등 수많은 도구들을 사용해 보았지만 모두들 한가지씩은 부족함을 느꼈다. 그리고 1999년 SAS E-Miner를 시작으로 Data Mining에 집중하기 시작했는데 몇번 해보니 이건 왜 이렇게 많은 수작업을 반복해서 마우스를 움직여야 되는지 이해가 안갔다. 그리고 조금은 흥미를 잃게 되었다.

그러다 몇 개월전 부터 R을 알게되어 사용하게 되었다. 첫 느낌은 충격 그 자체였다. 저자가 주로 사용하는 Mac은 물론 Linux 계열에서도 실행되고 명령어도 매우 간단하고 Object Orient개념이 있는 사람들에게 편리함을 주고, 수많은 package들의 기능과 새로운 많은 sample들이 저자를 끌어들이기에 충분하였다. 특히 자동화를 시도할 수 있는 점에서 KXEN에 비교해 가격대비 만족도는 최상이었다.

3000개가 된다는 package들이 혼란스럽기도 하지만 필요에 따라 선택해서 사용할 수 있었고 언제나 update가 지속적으로 이루어지고 있다. 그리고 1개월 정도의 적응기간을 거치면 매우 편리하게 사용할 수 있었다. 단, 한가지 많은 사람들의 불만은 볼만한 도서가 없고 인터넷을 열심히 뒤져야 된다는 것이다. 때로는 script들이 version호환성이 없거나 실행이 안되는 예제들도 은근히 많다. R의 열풍에 너도나도 cut&paste를 하다보니 생긴 일인것 같다. 그래서, 이러한 아쉬움을 해결하고자 이 책을 쓰게 되었다.

"R은 Open Source의 장점이 있지만 관련 자료들이 너무 분산되어 있고 정확성이 떨어지는 사례가 있어서, 이번 자료를 만들게 되었다."

R의 예제나 정보들이 대부분 인터넷에 있는 관계로 대부분의 sample들이 유사하고 help file에서 일부 수정된 경우가 많다. 주로 참조한 것은 Yanchang Zhao의 "R and Data Mining : Examples and Case Studies"이다. 이외의 참조문헌의 내용도 오류가 있거나 내용이 보완되어 추가되어야 할 내용이 있는 경우, 저자가 package관련 help file에서 참고를 하거나 변형하여 작성하였다. 그리고 도중에 데이터가 어떻게 변하는지 알아야 할 사항에 대해서는 일부러 데이터가 변화된 모습을 확인하도록 하였으며 script에 대해 line별로 comment를 달아 문장에 대한 특성과 흐름에 대한 이해를 도왔다. 그리고 script 중에 ">" 표시가 있는 건 한줄한줄 해보기를 바라는 내용이고, prompt가 없는것은 양이 많아서 cut&paste로 실행하라고 ">"표시를 없앴다. 모든 명령어를 모두 알아야 분석을 할 수 있는것은 아니다. 자신의 role에 따라 흐름만을 이해해도 되는 사람, 명령어 수준으로 이해해야 하는 사람, 상세한 option까지 알아야 할 사람 등 다양한 role에 따라 필요한 이해도가 틀리다고 생각한다. 이 내용은 상세한 옵션까지는 다루지 않았다.

내용의 coverage는 데이터 마이닝의 다양한 분야를 다루고자 한다. 단, 너무 깊이있는 통계적인 이야기는 하지 않고 편하게 쓸수 있도록 통계분석에 대해서는 가정과 해석과 활용방안을

일정수준 포함하고자 한다. 데이터 마이닝은 이러한 제약이 거의 없으므로 이책만으로도 충분히 접근할 수 있다. 전체적인 구성은 기초 데이터 핸들링, 마이닝, 비쥬얼라이제이션을 각각 2:7:1로 배분하고자 한다. 그리고 사용되는 데이터는 외부 데이터 보다는 package내에 들어가 있는 기본 데이터를 사용하여 혼란을 없애도록 하였다. 전체적인 내용은 대부분의 데이터 마이닝 책들이 갖는 공통적 구조를 갖고 있으나 다른 데이터 마이닝 도구에서 지원하지 않는 기능들이 50%는 들어가 있다. 이런 점이 R의 매력이라고 할 수 있다.

본 내용은 Open Source 특성상 수많은 정보를 통합하는 작업 자체도 힘든 일이었으며, 각 chapter 별로 보다 심도있게 다루기 위해 다양한 자료를 비교분석하여 작성하였으나 모두 확인 할 수 없었던 점도 있었다. 이점은 양해하기 바란다. 그리고 사용된 script의 모든 library를 선정하기 전에, 예를 들어 party package를 사용하고자 한다면 “library(“party”)라고 하기 전에 “install.packages(“party”)와 같은 형식으로 모두 설치를 하고 library를 설정하기를 바란다.

R1V1에서 추가된 점은 chapter 1에 기초 데이터 처리방안과 Visualization 보강을 하였으며, Social Network Analysis에 예제를 추가하였다.

내용상의 오류를 확인해준 이혜연 차장과 박준용 부장에게 감사를 전하며, 오류나 질문이 있으면 언제든지 eric@the-ecg.com으로 메일을 주면 최대한 답변을 할 수 있도록 하겠다.

1.1 Big Data Analysis

“Big Data Analysis는 Big Data 기술환경하에서 정형 및 비정형의 대용량 데이터를 통해 분석할 수 있는 환경이라고 할수 있다.”

“Data Scientist는 비즈니스에 대한 높은 이해도를 기반으로 Data Mining을 통해 비즈니스 목적을 달성할 수 있도록 해주는 인력으로, 기술기반이 아닌 비즈니스 이해를 기반으로 한 분석가로 전세계적으로 매우 부족한게 현실이다.”

Big Data하면 아직은 정의가 명확하게 자리잡지 못한 것 같다. 과거의 Very Large Data Base에 대한 논의의 확장으로 생각하면 Big한 데이터에 대한 분석이라 할 수 있고, 과거에는 주로 다루지 못했던 비정형 데이터를 포함한 분석이라고 말하는 이들도 있다. 실제로 두가지 모두를 포함해야 적합하다고 생각하며, 최근 시대적 동향도 비정형 데이터의 급증으로 인해 이러한 정보를 활용하고 예측력을 보다 효율적으로 향상시키는데 관심이 주어지고 있다. 세계경제 포럼은 2012년 떠오르는 10대 기술 중 그 첫 번째를 빅 데이터 기술에 선정하였고 국내 지식경제부 R&D 전략기획단은 IT 10대 핵심기술로 선정하는 등 전 세계적인 관심의 대상이 되고 있다.

언제나 그렇듯이 그 정의는 시대적인 비즈니스 요건에서 나온다고 생각한다. 그래서 저자의 생각은 “과거의 TB급 이상의 대용량 정보를 보다 신속하게 분석할 수 있고, 다루지 못했던 비정형 데이터를 추가로 결합하여 분석하는것을 Big Data Analysis다”라고 말하고 싶다.

이들의 특징은 데이터의 양(volume), 데이터의 속도(velocity), 데이터의 다양성(variety) 등 세가지 요소의 복합적인 특징을 갖고 있다.

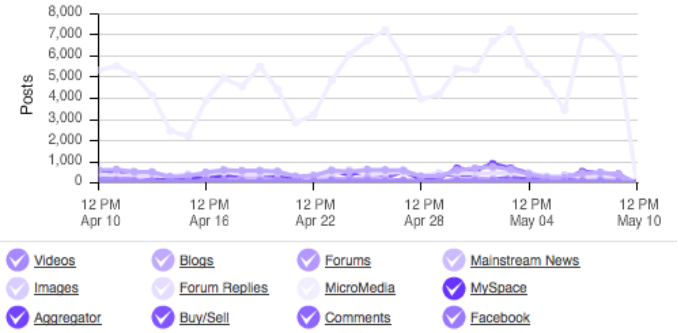
기술적으로 Hadoop, Hive, NoSQL 등을 언급하면 끝이

없을 것 같고 이러한 것들은 Big Data Analysis를 위한 기반일 뿐이다. 그래서 굳이 R을 이용한 Big Data Analysis에 이들과 결합된 내용들을 언급하지는 않겠다. 그리고 다른 Data Mining 도구들에 비해 차별화 되는 것은 R이 이러한 Big Data 기술요소와 잘 결합되어 활용될 수 있고, 그러한 지원을 각 vendor들로 부터 받고 있다는 사실이다.

Big Data Analysis는 크게 분석기술, 표현기술로 분류되며, 분석기술로는 Text Mining, Opinion Mining, Social Network Analysis, Clustering이 많이 언급되며 이를 위한 infra로 hadoop, NoSQL이 기반이 되고, 표현기술로는 R에서의 visualization이 많이 언급되고 있다.

또한 Big Data Analysis에서 가장 중요한 것은 요즘 자주 언급되는 Data Scientist다. 이들은 단순한 R 기술자가 아니고 통계학 박사가 아니라 비즈니스에 대한 이해를 기반으로 정형/비정형 데이터를 잘 활용해서 비즈니스 요건을 충족시킬 수 있는 분석을 지원하는 것이다. 이러한 인력들은 세계적으로 매우 드물다고 할 수 있다. 물론 대학에서 석/박사 과정을 통해 통계학이나 전산학에서 R을 접한 인력들이 많이 나오고 있으나 아직 이들이 비즈니스에 충분히 다가가지는 못했고 언제나 처럼 요원한 사항이기는 하나, 시대적 흐름이 이들을 강하게 요구하고 있다. 따라서 이 분야에 관심을 갖는 분들에게는 R에 대한 학습은 매우 좋은 기회라고 생각된다. 좌측의 word cloud를 보면 big data analysis와 관련된 가장 많이 언급되는 내용이 business와 company이다. 다른 무엇보다도 big data는 비즈니스적인 측면의 니즈에 대한 대응이라는 측면이 강하다. 또한

다양한 social media에서의 R의 언급을 보면 twitter와 같은 micromedia에서 하루 5000번 이상 언급되고, 계속 언급비율이 증가함을 알 수 있다.



1.2 데이터 마이닝 개요[1]

데이터 마이닝에 대한 다양한 정의가 있으나 저자는 “대용량 데이터 속에서 의미있는 데이터 패턴을 파악하거나 예측을 위해 데이터를 자동으로 분석해서 의사결정에 활용하는 방법”이라고 정의하고 싶다.

[1] “Big Data Analysis for CRM using R”, 김경태

"정확도가 높은 모델개발보다 많은 업무에 적용할 수 있는 좋은 데이터 마이닝 모델의 적용이 중요하다. 이를 위해선 자동화가 필요하다."

여기서 데이터 마이닝이 통계분석과 가장 큰 차이는 어떤 가설이나 가정에 따른 분석이나 검증이 아니고, 통계학을 전문으로 하는 사람들이 사용하는 도구가 이나다라고 말하고 싶다. 많이들 데이터 마이닝하면 통계를 생각하는데 물론 통계, 인공지능 등 다양한 분야와 관계가 있지만 이는 비즈니스 의사결정 도구이다.

따라서, 이 분야에 관심을 갖는 사람들은 좀 쉽게 생각하고 접근을 했으면 한다. 깊이 들어가면 알고리즘에 대한 내용이나 기술에 대한 내용 등이 언급될 수 있으나 현업에 있는 사람들 입장에서는 그다지 중요하지 않다고 생각된다. 단지 기본적인 이해와 몇번의 제대로 된 경험이 보다 중요하다고 생각된다. 주요 기법들로는

- Forecasting
- Clustering
- Classification
- Association Analysis
- Text Mining
- Social Network Analysis

이다.

데이터 마이닝 분야를 보면 오랜 역사에 비해 실제적으로 적용되기 시작한 것은 한국의 기준으로 90년대 중반부터라고 생각된다. 이후 2000년대중반에 와서 좀더 비즈니스 적 관점에서 바라보는 시각들이 생겨 CRM의 중요한 요소로 부각되기 시작했다. 많은 기업들이 거액을 투자하여 데이터 마이닝 도구를 구매하였고 이를 활용하려고 많은 시도를 한것 같으나 여전히 통계학 전문가와 대기업 위주의 시장이었다. 특히 CRM 분야에서 필요한 비즈니스적 관점의 시각을 갖춘 인력은 시장에서 더더욱 구하기 힘들었고 이러한 현상은 앞으로도 몇년간 지속될 것으로 생각된다.

2000년대 중반까지는 솔루션 측면에서 보면 SAS, Clementine의 시장이었다. Oracle이 Darwin을 인수합병하면서 데이터베이스에 일부 기능이 포함되기 시작했으며 최근 11g 등의 버전에서는 많은 데이터 마이닝 기능이 포함되었다. 매우 환영할 일이나 여전히 쓰기 힘들고 단순반복적인 작업이 많이 있어서 실무에 다양하게 적용하기에는 늘 상대적 어려움이 있었다. 아마 가장 큰 어려움은 경영진에 대한 설득과 데이터 준비와 단순반복적인 작업이라고 생각된다. 이분야의 내용들이 너무 데이터와 분석에 관련된 전문 내용들이다 보니 경영진과 대화를 하기에는 어려운 내용이다. 간단히 말하면 쉬울일을 어렵게 말하게 만드는 분야다.

그리고 데이터를 기본으로 하기 때문에 데이터 준비를 위한 데이터 추출, 가공 등의 일이 많은 부담이었고 비즈니스 적 관점에서 정의를 하고 활용방안을 적용하는데 시행착오들이 많았다. 대부분 이해의 부족이 많았다. 그러다 보니 대부분 데이터 마이닝을 데이터 가공용 도구로 활용하고 단순 빈도 분석용으로 사용하는 사례가 매우 높았고 현업에서의 신뢰도 낮은 상태였다. 어떤 경우는 마이닝 도구를 도입해서 데이터만 핸들링하는데 사용하는 경우가 있을 정도였고 대부분의 이슈가 데이터 핸들링 과정에서의 에러나 속도를 어떻게 해결하는지가 주 관심사였다.

현업에서는 수십, 수백개의 캠페인을 위해 모델링이 필요했으나 방법론 상의 수많은 수작업으로 년간 20개 정도의 모델을 만드는 수준인 경우도 많았고 더 많은 모델을 말들 필요성을 느끼거나 만들고 싶어하지도 않는 경우가 허다했다. 어떤 경우는 10명 이내의 팀원 중 1명 밖에 데이터 마이닝을 수행할 줄 몰라서 문제였고 어떤 경우는 10명 모두 사용가능해도 초보

수준이 대부분인 경우가 문제고, 석박사 급이 대부분이고 사용능력이 있어도 부서의 목표가 20개만 개발하면 되는 것이기 때문에 더 이상 개발 안하겠다는 곳도 있다.

일반적으로 모델링을 위한 준비에서 개발완료까지 1개를 개발하는데 1개월이 걸린다. 초보의 경우는 2개월이 소요되기도 한다. 전문 컨설턴트의 경우 2주에 1개 개발하는데 무리가 없다. 그러나 10개 이상의 모델을 만들어야 되는 경우 인력과 비용의 제약을 무시하더라도 동시에 10명이 1개월간 모델을 만들려고 서버를 이용하면 서버는 거의 작동 불능상태에 빠진다. 대부분 서버가 응답하기를 기다리며 몇시간에서 하루이상을 대기하기도 한다. 결국 10명이 작업을 하더라도 10개 모델을 만드는데 소요되는 시간은 5개월 이상이 소요되게 된다. 그리고 만들어야 할 보고서도 너무 많아 자료를 가공하여 엑셀이나 파워포인트를 만들려면 그 작업만 1개월 이상이 걸리고 sql로 script를 만들려면 generation된 code를 수정하는데 1개월은 걸린다. 이게 대부분의 경우의 현실이다. 이러한 문제점의 가장 큰 공통점은 솔루션의 수작업에 의존하는 기능과 방법론 상의 문제이다. 모두가 이렇게 작업하지는 않는다.

위에서 언급한 대부분의 문제를 해결한 것이 KXEN이다. 시장에서는 전세계 3위의 솔루션이고 한국에서는 잘 알려져 있지 않다. 현재 1,2위는 잘 알려진 SAS E-Miner, IBM SPSS Clementine이다. 저자가 다년간 사용해본 도구들로 각각 특징과 장단점들을 갖고 있다. 여기서는 너무 자세한 내용을 다루지는 않겠다. 단지 KXEN은 최신 알고리즘을 적용하고 있다는 점과 대부분의 프로세스가 자동화 하여 모델 개발 생산성이 20배 이상 높다는 것이다. 원한다면 몇시간에 1개의 모델을 만들수도 있고 하루에 1개의 모델을 만들 수도 있다. 여러 상황을 고려한다면 1주에 1개를 만드는데 무리가 없다고 생각하면 된다. 이러한 도구 중의 하나가 R이다. 다른점은 언어방식이라는 점과 매우 간단하게 표현이 가능하고, Open Source라는 측면이다. 20줄이면 모델하나 만들어서 적용을 할 수 있다. 또한 자동화도 매우 flexible하게 할 수 있다.

그럼, 구체적으로 데이터 마이닝을 추진하려면 어떻게 해야 할까? 데이터 마이닝을 위해서는 일단 소프트웨어와 데이터가 있어야 될것이다. 두번째로 무엇을 왜 하고자 하는지 명확한 목적이 정의되어 있어야 된다. 아니면 이것을 정의하는 단계부터 시작한다. 목적이 정의가 안되면 뒤에 모델을 개발하는 단계는 전혀 의미가 없다. 목적은 이해 관계자들이 모두 동의할 수 있어야 되고 이해가 될수 있어야 된다.

데이터를 준비해야 한다. 고객정보와 거래정보, 상품 마스터 정보 등이 필요할것이다. 가능하면 웹로그 정보나 소셜 네트웍에서의 정보도 활용할 수 있다. 대부분 용량이 많은 데이터이므로 IT와 사전에 협의하여 데이터에 접근하여 부하가 심한 일을 하는데 문제가 업도록 일정과 도움을 요청한다. 필요하면 데이터를 다른 서버에 저장해 주어서 운영에 지장이 없도록 할 수 있다.

데이터 가공단계에서는 모델링 목적에 따라 목적 변수를 정의하고 필요한 데이터를 적합한 형식에 맞게 가공해야 한다. 나중에 논의할 내용이지만 classification modeling을 위해서는 CRM Data Mart 형식으로 데이터를 만들어야 되므로 시간이 많이 소요된다. 모델 개발 단계에서는 데이터에 대한 읽기와 데이터 마이닝을 실행하는데 부하가 걸리게 되므로 동시에 여러명이 작업하는데는 충분한 CPU, memory, disk 사용경합 등을 고려해야 한다. 때로는 64GB이상의 메모리가 하루종일 1개의 모델링을 위해서도 필요하다. 따라서 모델링 일정계획을 팀원간에 잘 수립해야 한다.

"64bit 환경의 많은 CPU, 64GB 이상의 메모리 활용이 유연하게 지원될 수 있는 환경이 필요하다."

검증단계에서는 테스트 마케팅이나 과거 데이터를 통해 검증할 수 있다. 그러나 테스트 마케팅을 하는것과 모델링의 차이를 잘 이해해야 한다. 예를 들어 휴면고객이 win-back될 가능성을 모델링하는것과 휴면고객 대상으로 win-back캠페인을 실행했을 때 반응할 고객을 모델링하는것은 전혀 다른 이야기다. 검증이 되었으면 자동화 방안을 IT와 협의하여 상시 데이터 마이닝 결과를 적용하여 업무에 적용할 수 있도록 해야 한다. 그리고 보고서를 작성하여 경영진에 기대효과를 년간 추가수익, ROI로 말할 수 있어야 된다. 여기까지가 한단계의 사이클이다.

또한 classification model을 개발할 때에는 언제나 train data와 test data로 구분지어 모델링에 접근한다. 전체 데이터를 7:3 또는 8:2 등으로 나누어 train을 해서 최적의 모델을 확정지은 다음에 test로 검증을 한다. 1만건 이하인 경우 저자는 주로 8:2로 모델을 개발한다. 그리고 Train과 test 데이터 간에는 편차가 없어야 되며 성능은 test가 다소 낮게 나오는 경향이 있다.

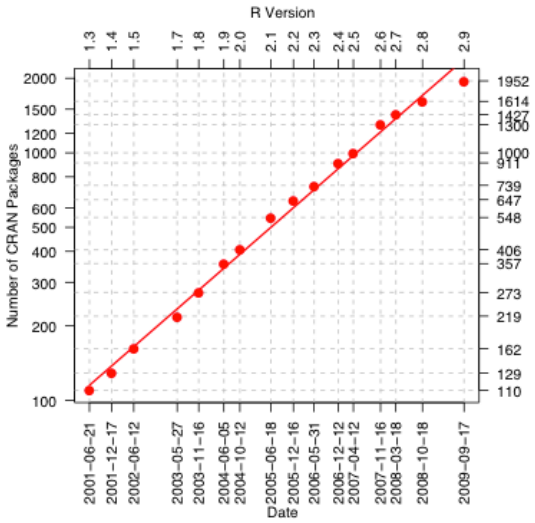
모든 데이터 마이닝 모델은 한번 개발하였다고 계속 사용할 수 있는것이 아니다. 모든 모델링은 revise할 주기가 있는데 classification은 최소 1년에 2번, association rule은 비즈니스 특성에 따라 1주 또는 1개월, forecasting은 일/주/월 단위등 모델링 기준에 따라 달라지고, clustering은 1년에 2회는 최소 실행해야 한다. 이러한 모델 revise주기는 근본적으로 accuracy의 deviation이 급증하는 시점에 자동으로 실행될 수 있는게 가장 초적의 접근이다. 누적 개발된 모델이 1000개가 된 경우 이를 모두 수작업으로 revise한다면 불가능하다.

성공적인 데이터 마이닝을 하는데는 가장 핵심은 비즈니스에 대한 전반적인 프로세스에 대한 이해다. 그리고 각 프로세스에서 어떠한 형태로 데이터가 발생되어 변형되고 축적이 되는지를 이해하고 필요한 데이터를 선별 할 수 있어야 된다. 인터뷰를 통한 접근은 사실 시간이 너무 많이 소요되고 정확성이 떨어진다. 따라서, 간단하고 짧은 기간에 효율적으로 진행하는 것이 적합하다. 주로 몰랐던 사실을, 운영방식을 이해하는데 활용하면 된다. 그리고 데이터에 대한 전반적인 파악이 중요하다. 많은 시간을 쓰지말고 fact와 특이사항을 파악해서 brainstorming을 현업과 할 수 있어야 된다. 그 다음은 결론적으로 mart를 잘 만드는 일이다. 향후 어떠한 분석을 할것인지, 모델링을 할것인지를 고려해서 한번에 잘 만들어야 된다. 보통 마트 만드는 일은 자동화가 안되면 3회 이상의 오류 수정을 거쳐 데이터 입수에서 마트 생성까지를 반복하게 된다. 자동화가 절실한 부분이다. 그리고 모델링이다. 모델링에서 알고리즘이 성능을 결론짓는 경우는 마케팅 분야에서는 없다. 연관성 분석, Social Network Analysis 정도만 처리하는데 소요되는 메모리와 속도가 문제가 될 뿐이다. 모델링은 빨리 신속하게 개발해 보고 정교화를 시도하는게 적합하다. 전체 데이터로 처음부터 접급하지 말고 sampling을 최대한 활용하는게 적합하다.

1.3 Data Mart

1.4 산업별 데이타 마이닝 적용사례

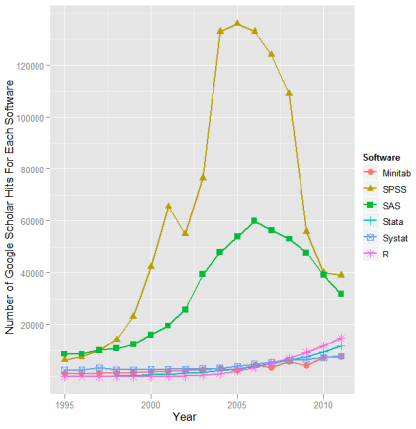
1.5 R의 역사



R은 Open Source 프로그램으로 통계/데이터 마이닝 및 그래프를 위한 언어이다. R은 주로 연구 및 산업별 응용 프로그램으로 많이 사용되고 있으며, 최근에는 기업들이 많이들 사용하기 시작했다. 특히, Big Data Analysis 목적으로 각광을 받고 있으며, 3000개가 넘는 package[2]들이 다양한 기능을 지원하고 있으며 수시로 update되고 있다.

특히, SAS, SPSS, KXEN에 비해 큰 차이점은 다양한 최신 통계분석 및 마이닝 기능을 R이라는 플랫폼하에서 제공한다는데 있다. 사용 패키지들은 새로운 알고리즘을 적용하는데 매우 오랜 시간이 걸린다. 최근 알고리즘을 제공하는 KXEN 보다도 더 다양한 최신 알고리즘을 제공하여 다양한 시도를 할 수 있는 점이 큰 장점이다. 두번째로, 이러한 기능을 자동화 할 수 있다는 것이다. 언어에 가까운 문장형식이므로 자동화가 너무 쉽다. 세번째로, 다양한 사용자들이 다양한 example들을 공유한다는것이다. www.r-project.org에서의 Core 팀에 의한 내용 및 www.r-blogers.com에서의 다양한 예제들은 큰 도움이 된다.

이러한 측면이 저자가 R에 대해 매료될 수 밖에 없고 가격과 효율성을 고려할때 R이 최고라고 생각한다. 그렇다고 정확도가 떨어지는것은 아니다. 일정수준의 데이터 마이닝 도구는 정확도는 문제가 안된다. 모든것은 모델러가 어떻게 모델을 정의하고 데이터를 준비하는지에 달려있다고 생각한다.



참고로 R에 대한 언급이 다른 통계 패키지들과 비교해서 어떻게 변화하였는지를 보여주는 Google Scholar에서의 데이터이다. 이에 대한 데이터로 예측을 해보면 2015년에는 R이 1위로 올라갈것으로 예측된다. 저자의 개인적 판단으로는 SPSS의 급격한 성장은 PC 보급 및 SPSS의 설치 요건이 낮아 학교나 기업에서의 사용이 급격히 상승한점과 SPSS 자체의 대학들 및 연구소들에 대한 집중적인 마케팅 효과가 큰 효과를 보았다고 생각한다. 그러나 200년대 중반부터의 경기불황으로 기업들이 예산을 줄인점과 Big Data에 대한 관심이 Open Source인 R에 대한 관심이 급증되어 상승세를 불러왔다고 본다.

[2] 2011년 5월 기준

1.6 기초 데이터 작업[3]

"데이터 핸들링 작업은 많은 시간을 들여 천천히 필요한 경우에 학습을 해가면서 접근하는것이 효율적이다."

금번 버전에서는 너무 상세한 기초문법이나 데이터 처리방안을 말하지 않고, 기본적으로 알아야 할 데이터 타입 및 읽기, 저장하기, 지우기에 대해서만 언급하겠다.

데이터 타입 및 정의에 관한 내용이다. R에서는 데이터 할당시 "="대신 "<-"를 사용한다. 물론 "="도 작동을 한다. 데이터 종류는 Vectors, Matrices, Arrays, Data Frames, Lists, Factors로 구분된다.

> a <- 1:10
변수를 저장할 때는 변수에 "<-"를 이용한다. 1:10은 1에서 10의 값을 갖는다.

Vector는 numeric, character, logical 등이 되며, 아래와 같이 할당할 수 있다.
a <- 1
b <- c(1,2,3,-1)
c <- c("one", "two", "eric")
d <- c(TRUE, FALSE, TRUE)

그리고 vector의 element를 참조할 때는
c[c(1,3)]
로 지정하면 c vector의 첫번째와 세번째 값을 가져오게 된다.

Matrices는 모든 컬럼은 동일한 모드(number, character, etc)와 길이를 갖게 된다.
> x <- matrix(1:15, nrow=3, ncol=5)
> cell <- c(1,3,5,7)
> rnames <- c('R1', 'R2')
> cnames <- c('C1','C2')
> y <- matrix(cell, nrow=2, ncol=2, byrow=TRUE, dimnames=list(rnames, cnames))

특정값을 참조하기 위해서는
> y[,2]
R1 R2
 3 7
> y[1,]
C1 C2
 1 3

데이터 프레임으로 만드는 방법이다.
> var1 <- (1:5) / 5
> var2 <- c("my name", "data mining", "big data", "R","package")
문자를 숫자와 마찬가지로 동일한 형식으로 할당한다.
> a <- data.frame(var1, var2)
var1과 var2를 결합하여 한개의 데이터 프레임으로 만든다.

List는 순서가 있는 object들의 모임으로 다양한 object를 하나의 이름으로 저장할 수 있다.

---

[3] http://www.statmethods.net/

> l <- list(name='eric', age=45)

Factor는 nominal value를 vector로 저장한 것이다.

```
> gender <- c(rep("male",10),rep("female",10))
> gender
 [1] "male"   "male"   "male"   "male"   "male"   "male"   "male"   "male"
 [9] "male"   "male"   "female" "female" "female" "female" "female" "female"
[17] "female" "female" "female" "female"
> gender <- factor(gender)
> gender
 [1] male   male   male   male   male   male   male   male   male   male   female
[12] female female female female female female female female female
Levels: female male
> summary(gender)
female   male
    10     10
```

데이터를 저장하고 출력하는 방법이다.

> write.csv(a, "test.csv")

csv파일로 저장한다.

> b <- read.csv("test.csv")

csv로 저장된 내용을 b라는 데이터로 읽어들인다.

> save(a,file="abc.Rdata")

로 해당변수를 데이터 파일로 저장한다.

> print(b)

데이터를 읽어들이는 방법이다.

> load("abc.Rdata")
> print(a)

는 default working directory에서 abc파일을 읽어 a를 메모리에 올리고 인쇄한다.

데이터를 삭제하는 방법이다.

> rm(a)

변수 a만 삭제하는 경우

> rm(list=ls(all=TRUE))

변수를 메모리에서 지우고자 하는 경우에 해당 변수명을 지정한다.

이외의 간단한 함수로는 아래와 같은 것들이 있다.

> data(iris)
> describe(iris)
> summary(iris)

이정도만 알고 다양한 데이터 타입과 데이터를 할당하고 변경하는 방법을 배우고, dbms나 sas, spss 파일을 읽고 처리하는 방법을 알게되면 데이터 처리에는 문제가 없어진다.

1.7 Missing Data Handling

"Missing data처리를 위해 시간을 많이 쓰는것은 비효율적이다. 가능하면 missing은 제외하고 처리하는게 적합하다. Missing 자체가 의미가 있는 경우도 있다."

Missing data를 어떻게 처리하느냐는 전체 작업속도에 많은 영향을 준다. 특히 이부분을 자동화 시키는 경우 업무효율성은 매우 향상된다. R에서 관련 package로는 Amelia II, Mice, mistools 등 여러가지가 있으나 본 내용에서는 Amelia를 사용하였다.

우선 missing data를 확인하는 방법과 제외하는 간단한 방법부터 설명하고자 한다. R에서는 missing data를 NA (not available)로 처리한다. 불가능한 값(예를 들면, dividing by zero) 는 NaN (not a number)으로 처리된다. Missing data를 입력하는 방법은 NA를 이용하면 되고, is.na를 이용하여 missing여부를 확인할 수 있다.

```
> y <- c(1,2,3,NA)
> is.na(y)
[1] FALSE FALSE FALSE  TRUE
```

특정값을 missing으로 입력한 경우 이를 변환하여 처리하면 되는데 '99'를 missing으로 처리하는 방법은 해당 값이 있는 위치에 NA를 입력하는 방식이다.

```
mydata[mydata$v1==99,"v1"] <- NA          # 실행하지 마세요
```

평균을 산출하는 등 데이터 처리에서 missing으로 인한 문제를 해결하는 방법으로는 해당값을 제외시키는 방법이 있다.

```
> x <- c(1,2,NA,3)
> mean(x)
[1] NA
> mean(x, na.rm=T)
[1] 2
```

또한, missing이 포함된 record 자체를 삭제하는 방법으로는 complete.cases() 함수를 이용하는데 missing이 넓게 분포된 경우 많은 데이터의 삭제로 문제가 발생할 수 있으니 염두에 두어야 된다.

```
mydata[!complete.cases(mydata),]          # 실행하지 마세요
```

Amelia를 사용하는 방법은 아래와 같다.

```
> install.packages("Amelia")
> library("Amelia")
```

데이터는 freetrade 관련 자료로 무역정책에 대한 자유화에 대한 분석으로 1980년 부터 1993년의 데이터로 구성되어 있다. 변수는 년도, 국가, 관세율, 정치지수(-10~10으로 자유화된 국가는 10의 값을 갖는다), 총인구, 총국민생산, 총국제준비액, IMF 가입년도, 재무적 공개성, US선호지수로 구성되어 있다.

```
> data(freetrade, package="Amelia")
> summary(freetrade)
      year        country              tariff
 Min.   :1981   Length:171         Min.   :  7.10
 1st Qu.:1985   Class :character   1st Qu.: 16.30
 Median :1990   Mode  :character   Median : 25.20
 Mean   :1990                      Mean   : 31.65
 3rd Qu.:1995                      3rd Qu.: 40.80
 Max.   :1999                      Max.   :100.00
                                   NA's   : 58.00
    polity            pop               gdp.pc
 Min.   :-8.000   Min.   : 14105080   Min.   :  149.5
 1st Qu.:-2.000   1st Qu.: 19676715   1st Qu.:  420.1
 Median : 5.000   Median : 52799040   Median :  814.3
 Mean   : 2.905   Mean   :149904501   Mean   : 1867.3
 3rd Qu.: 8.000   3rd Qu.:120888400   3rd Qu.: 2462.9
 Max.   : 9.000   Max.   :997515200   Max.   :12086.2
 NA's   : 2.000
   intresmi          signed          fiveop
 Min.   : 0.9036   Min.   :0.0000   Min.   :12.30
 1st Qu.: 2.2231   1st Qu.:0.0000   1st Qu.:12.50
 Median : 3.1815   Median :0.0000   Median :12.60
 Mean   : 3.3752   Mean   :0.1548   Mean   :12.74
 3rd Qu.: 4.4063   3rd Qu.:0.0000   3rd Qu.:13.20
 Max.   : 7.9346   Max.   :1.0000   Max.   :13.20
 NA's   :13.0000   NA's   :3.0000   NA's   :18.00
    usheg
 Min.   :0.2558
 1st Qu.:0.2623
 Median :0.2756
 Mean   :0.2764
 3rd Qu.:0.2887
 Max.   :0.3083
```

위의 내용을 보면 해당 변수에 NA갯수가 몇개인지를 알수 있다. Tariff의 경우 NA가 58개 있다는 뜻이다.

일반적인 missing 처리 방식은 해당 record를 모두 삭제하는 방법이다. 이러한 경우 missing이 전체적으로 많은 record에 걸쳐 분포하는 경우, 너무 많은 자료가 삭제되어 정보를 획득하기 어려운 경우가 있다. 그러한 경우 imputation을 해당 변수의 대표값으로 대체하는 경우가 있는데 이러한 방식도 많은 문제점이 있다. 그래서 변수들간의 관계를 이용하여 imputation을 실시하는 효율적인 방법이 있는데 금번 예제에서는 그러한 방식을 제공하고자 한다. m은 몇개의 imputation dataset을 만들지를 결정하는 값이며, ts는 시계열에 대한 정보, cs는 cross-sectional 분석에 포함될 정보이다. 따라서 아래 모델에서는 년도와 국가를 고려해서 모든 freetrade정보를 활용한 missing에 대한 imputation이 이루어지게 된다.

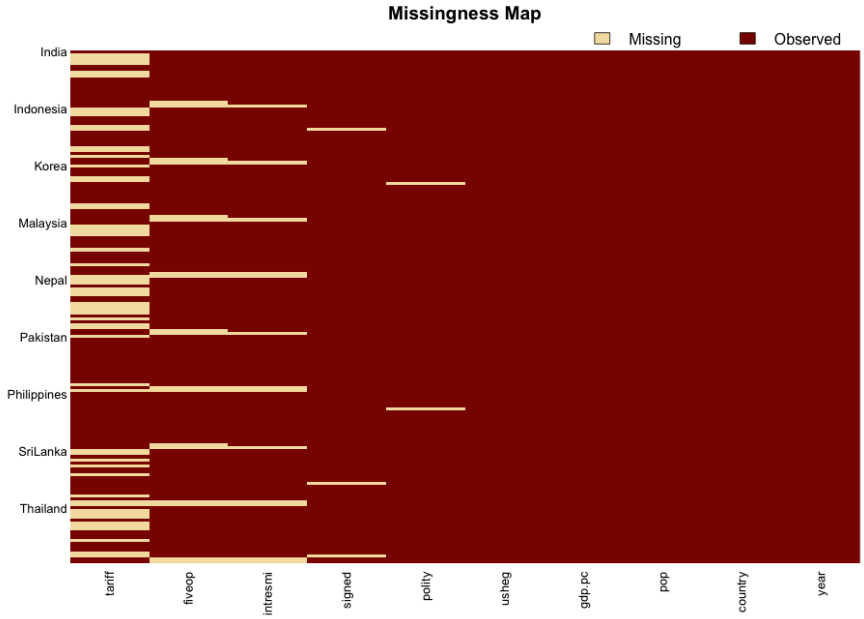
```
a.out <- amelia(freetrade, m = 5, ts = "year", cs = "country")
hist(a.out$imputations[[3]]$tariff, col="grey", border="white")
save(a.out, file = "imputations.RData")
write.amelia(obj=a.out, file.stem = "outdata")
```

This will create one comma-separated value file for each imputed dataset in the following manner:

outdata1.csv
outdata2.csv
outdata3.csv
outdata4.csv
outdata5.csv

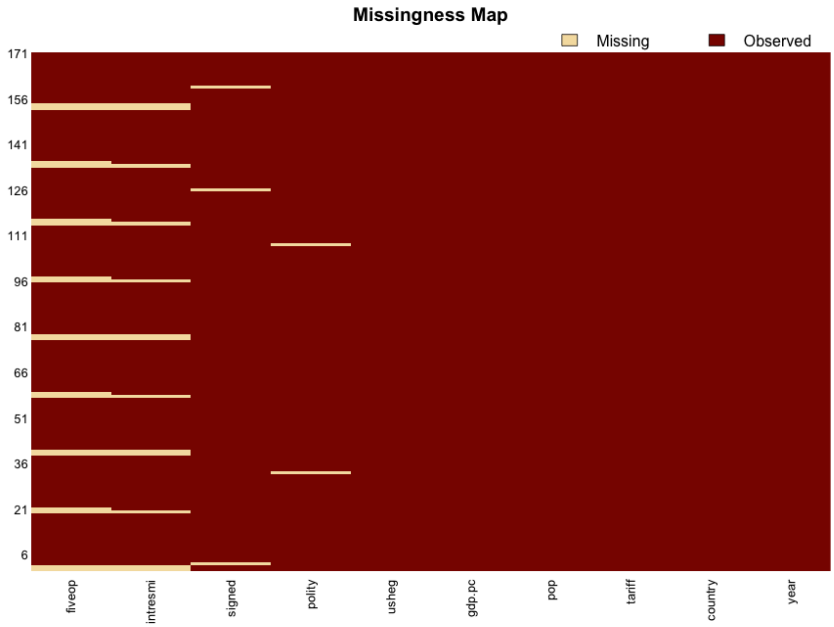
아래 내용을 보면 missing을 처리하기 전과 후의 변화를 알 수 있다.

> missmap(a.out)



imputation을 하기 위해서는 imputation을 위한 값을 dataset에 writting하면 된다.

> freetrade$tariff<-a.out$imputation[[5]]$tariff
> missmap(freetrade)



1.8 Outlier Detection

다음 버전에서 다루고자 함.

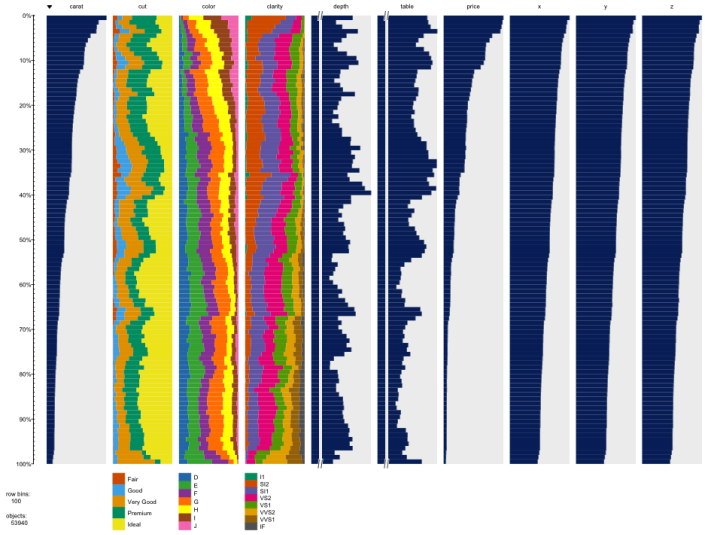
1.9 Data Exploration

> require(ggplot2)
ploting을 위해 호출하며, require나 library는 큰 차이는 없으나 require의 원래 목적은 function안에서 사용하기 위해 쓰는것으로 library가 loading되어 있지 않으면 warning을 주도록 되어 있다.
> data(diamonds)
다이아몬드의 carat, cutting 상태등의 데이터이다.
> summary(diamonds)

```
     carat               cut        color        clarity     
 Min.   :0.2000   Fair      : 1610   D: 6775   SI1     :13065  
 1st Qu.:0.4000   Good      : 4906   E: 9797   VS2     :12258  
 Median :0.7000   Very Good:12082   F: 9542   SI2     : 9194  
 Mean   :0.7979   Premium   :13791   G:11292   VS1     : 8171  
 3rd Qu.:1.0400   Ideal     :21551   H: 8304   VVS2    : 5066  
 Max.   :5.0100                     I: 5422   VVS1    : 3655  
                                    J: 2808   (Other): 2531  
     depth           table           price             x         
 Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000  
 1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710  
 Median :61.80   Median :57.00   Median : 2401   Median : 5.700  
 Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731  
 3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540  
 Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740  
```

```
       y                z          
 Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 4.720   1st Qu.: 2.910  
 Median : 5.710   Median : 3.530  
 Mean   : 5.735   Mean   : 3.539  
 3rd Qu.: 6.540   3rd Qu.: 4.040  
 Max.   :58.900   Max.   :31.800  
```

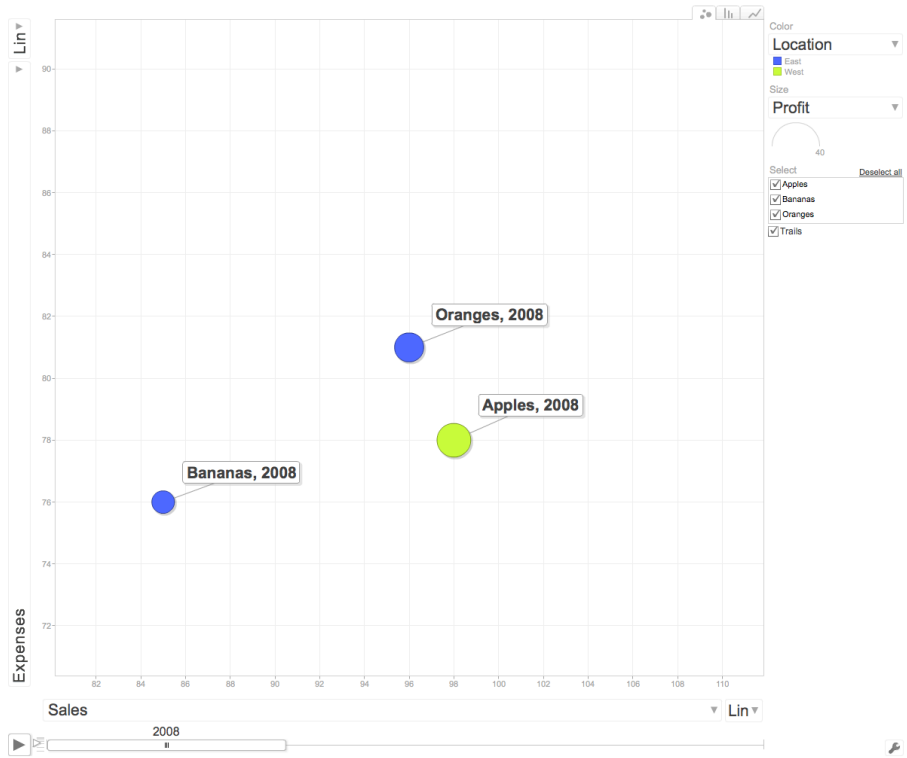
```
> install.packages("tabplot")
> library(tabplot)
# 데이터가 어떻게 입체적으로 분포되어 있는지를 보기 위한 방법이다.
> tableplot(diamonds, cex=1.8)
```



위 그래프 내용을 보면 carat이 클수록 cut 상태가 개선됨을 알 수 있다. 이렇게 값의 변화에 따른 전체적인 다른 변수와의 관계도 파악할 수 있어 시각적인 효과가 도움이 된다.

1.10 Interactive Graph

```
> library(googleVis)
> data(Fruits)
# 년도, 과일종류, 위치, 매출액, 비용, 이익, 일자 관련된 데이터가 있다.

> M1 <- gvisMotionChart(Fruits, idvar="Fruit", timevar="Year")
# 그래프 축에 관련된 시간과  id 변수를 지정한다.
> plot(M1)
# web browser가 뜨면서 그래프가 생성이 되며, interactive하게 그래프가 작동된다.
```




```
> library(MASS)
> data(iris)
> ldaobj <- lda(Species ~ ., data=iris)
> ldapred <- predict(ldaobj)$posterior
$sv
      setosa versicolor virginica
1  0.5761169  0.2119416 0.2119416
2  0.5761169  0.2119416 0.2119416
3  0.5761169  0.2119416 0.2119416
4  0.5761169  0.2119416 0.2119416
5  0.5761169  0.2119416 0.2119416
....
```

klaR package를 이용한 다양한 분석에 대해 설명하고자 한다. 우선 가장 공통적인 모델링 이전 작업으로 데이터가 너무 많은 경우 아무리 시스템 성능이 좋아도 분석자에게는 부담이다. 보다 빨리 전체적인 파악을 하고 상세 모델링을 해야 하기 때문이다. 한번 모델링을 하는데 시스템 처리 시간이 1시간이 걸린다면 업무를 효율적으로 할 수 없다. 결국 우리는 10분 내에 대략적인 파악을 하고 Insight를 얻고 Feasibility에 대해 생각해야 한다. 현재 데이터를 갖고는 모델링 성능이 안나온다면 보다 상세한 분석을 통해 derived variable을 추가하거나, 모델링 절차를 바꾸는 등의 대안을 생각해야 한다. 이러한 경우 observation이 많은 경우는 sampling을 하면 되지만, 이 또한 시간이 걸리고 디스크 공간을 차지한다. 또하나의 대안으로는 변수가 30개 이상이 넘는 경우 예를 들어 300개의 변수가 있다면 일단, 변수를 50개 이하로 줄여봐야 될것이다. 경우에 따라서는 빠른 모델링을 우선 실행하기 위해 두가지 선택을 모두 해야 될 경우도 있다. 이런 경우 모델링 목적에 따른 변수의 선택을 해야 한다. 이를 위한 방법이 greedy.wilks를 사용하는것이다. 모델링을 정의하고 이에 따라 변수를 stepwise하게 투입하여 의미있는 변수를 순서대로 보여준다. 그러면 효율적으로 정확도를 최소한 희생하면서 초기 모델링을 빨리 실행할 수 있게된다. 시스템 성능 만 믿고 무조건 전체를 다 돌리는것 만큼 비효율적인 방법은 없다.

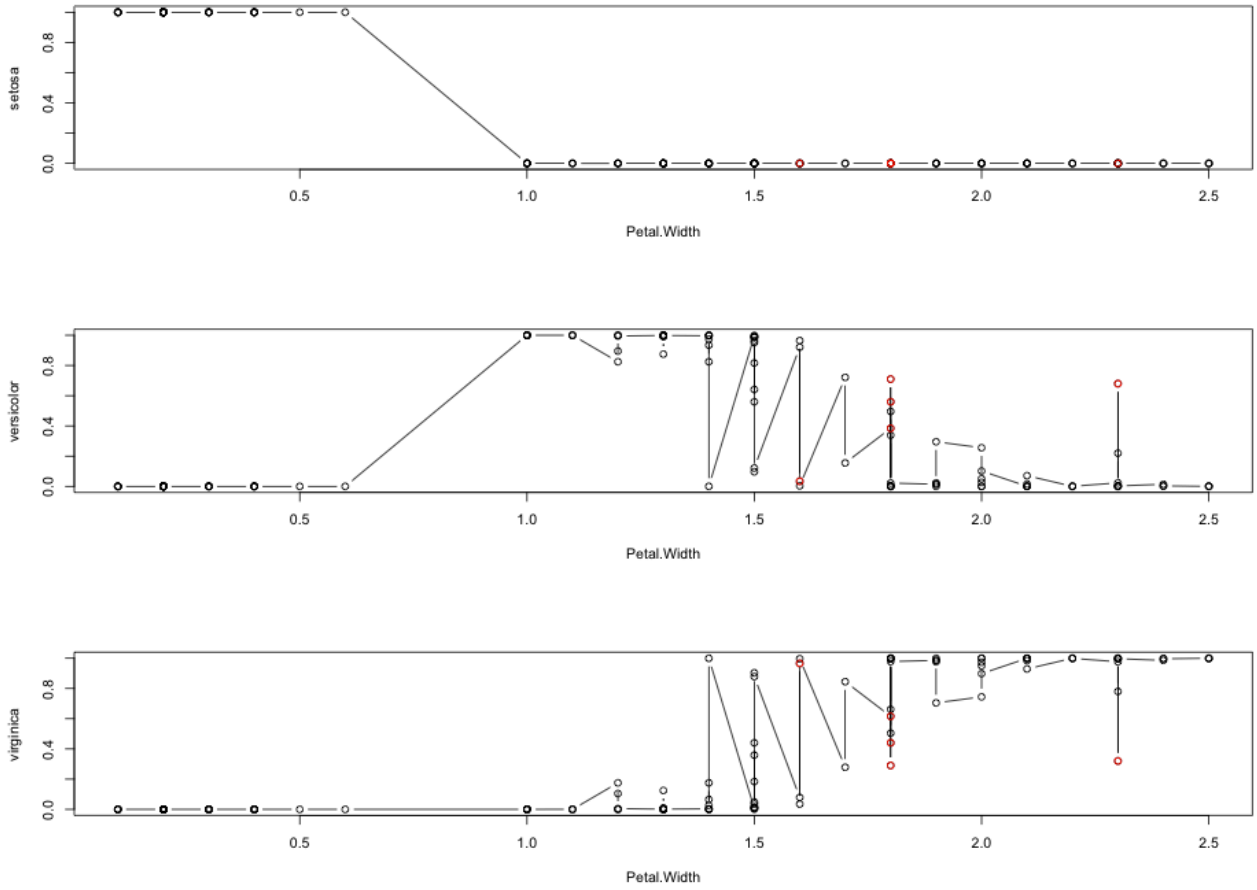
저자는 언제나 simulation modeling을 하건 data mining을 하건 데이터 사이즈를 증대시켜가면서 해당 시스템에서 어느 정도 시간이 소요되는지 performance test를 해본다. 그리고 분석 시나리오를 결정한다. 이런 접근을 하지 않는 경우 초기 분석은 가능하나 시스템 성능문제로 모델링을 완성할 수 없는 경우도 있다.

```
> library(klaR)
> data(B3)
> gw_obj <- greedy.wilks(PHASEN ~ ., data=B3, niveau=0.1)
> gw_obj
Formula containing included variables:

PHASEN ~ EWAJW + LSTKJW + ZINSK + CP91JW + IAU91JW + PBSPJW +
    ZINSLR + PCPJW
<environment: 0x7fb7c4961c78>


Values calculated in each step of the selection procedure:

     vars Wilks.lambda F.statistics.overall p.value.overall F.statistics.diff p.value.diff
1   EWAJW    0.6058201            33.18341    1.405358e-16         33.183411 1.405358e-16
2  LSTKJW    0.4271561            26.85606    1.218146e-25         21.192038 1.554268e-11
3   ZINSK    0.3614525            21.20584    7.607587e-29          9.149422 1.326989e-05
4  CP91JW    0.3002868            19.05337    1.153881e-32         10.184539 3.783582e-06
5 IAU91JW    0.2624925            17.11094    6.597858e-35          7.151127 1.604993e-04
6  PBSPJW    0.2451025            14.99388    3.695840e-35          3.500196 1.708972e-02
7  ZINSLR    0.2205325            13.94619    1.442943e-36          5.459204 1.379166e-03
8   PCPJW    0.1999847            13.10739    9.454573e-38          5.000333 2.486333e-03
```

특정 변수가 주어졌을 때 class가 어떻게 분류되는지에 대해 error rate를 돌려주고, graphical하게 결과를 보여주는 기능을 소개한다.

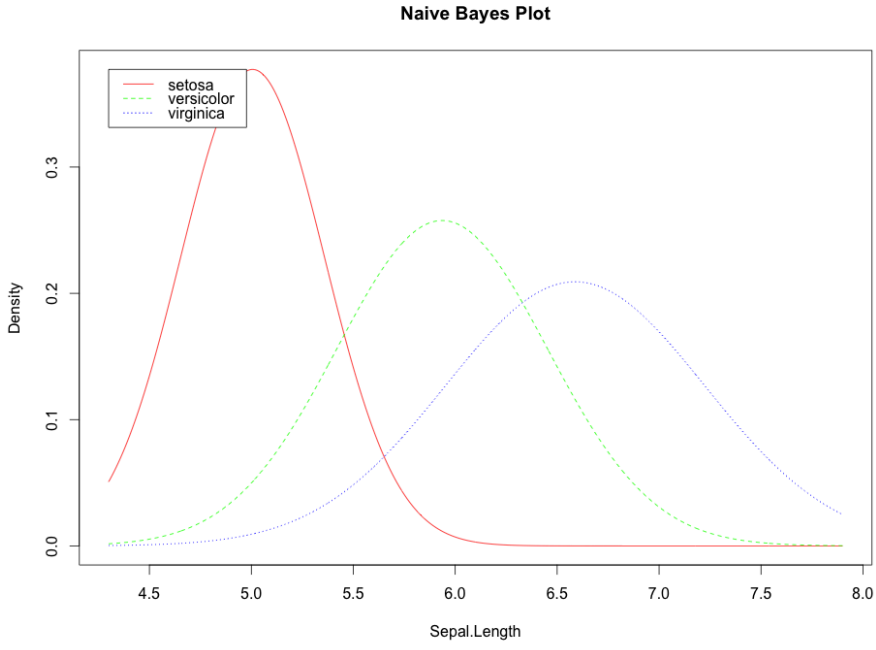
```
> library(klaR)
> data(iris)
> iris2 <- iris[ , c(1,3,5)]
> plineplot(Species ~ ., data = iris2, method = "lda",
+           x = iris[ , 4], xlab = "Petal.Width")
[1] 0.03333333
```

아래 그래프를 보면 Petal.Width에 따라 Species의 분포가 다름을 알 수 있다. 이러한 경우 Petal Width를 grouping해서 categorical variable로 만들어 주면 해석이 용이할 수 있다.




```
> library(klaR)
> data(iris)
> mN <- NaiveBayes(Species ~ ., data = iris)
> plot(mN)
```

변수 별로 그래프를 보여주지만 1개만 예제로 표시했다.



Continuous variable 보다 categorical variable이 이해가 용이할 수 있다. 이런 경우 간단한 방법은 continuous variable을 일정 크기로 binning해서 활용을 한다. 일반적으로 binning의 갯수가 증가하면 정확도는 높아지나 속도가 느려지고 overestimation될 수 있다. 기본적으로 40개 정도를 binning하고 이를 target과 비교하여 유사한 performance를 보이는 인접구간을 merge하는 방식을 적용하는것이 적합하다. 여기서는 그러한 상세 기능까지는 포함되어 있지 않다.

```
> library(klaR)
> library(party)
> data(iris)
> iris$Petal.Width.c <- cut(iris$Petal.Width, 5)
> a<-ctree(Species ~.,data=iris)
> plot(a)
> a

        Conditional inference tree with 5 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, Petal.Width.c
Number of observations:  150

1) Petal.Width.c == {(0.579,1.06], (1.06,1.54], (1.54,2.02], (2.02,2.5]}; criterion = 1, statistic =
251.551
  2) Petal.Width <= 1.7; criterion = 1, statistic = 83.076
    3) Petal.Length <= 4.8; criterion = 1, statistic = 26.855
      4) Petal.Length <= 3.6; criterion = 1, statistic = 20.634
        5)*  weights = 7
      4) Petal.Length > 3.6
        6)*  weights = 40
    3) Petal.Length > 4.8
      7)*  weights = 8
  2) Petal.Width > 1.7
    8)*  weights = 46
1) Petal.Width.c == {(0.0976,0.579]}
  9)*  weights = 49
```

Chapter 2

분류 및 예측 모델

2.1 Classification[4]

"가장 많이 사용하게 되는 기법이다. 그러나 알고리즘 자체에 너무 집착할 필요가 없다. 특히 마케팅 분야에서는 많은 모델을 개발하는 게 더 중요하다"

Classification은 0과 1을 구분하거나 0,1,2,3,4 등의 구분을 분류하는데 목적이 있다. 이는 응용을 하면 CRM에서는 고객의 행동을 예측하는데 또는 고객의 속성을 파악하는데 응용되고 다양한 분야에서 이를 활용할 수 있고, 또 다른 마이닝 결과를 연계하여 분석하는데도 사용할 수 있다.

특히 CRM에서 고객의 행동을 미리 예측할 수 없으면 고객에 대한 니즈를 충족시키기 어려울것이다. 고객은 불만이 쌓이고 결국은 휴면화 되고 이탈되어 고객은 점점 줄어들어 비즈니스는 기울게 될것이다. 따라서 고객 행동을 데이터 마이닝과 같은 기법으로 예측하여 사전에 고객을 만족시킬 수 있어야 된다.

R에서 지원되는 classification 방법으로는 rpart, rpartOrdinal, randomForest, party, tree, marginTree, maptree 등 다양한 방법이 존재한다. 우선 party와 rpart를 갖고 접근하는 방법을 제시하고자 한다.

2.1.1 활용분야

행동예측 유형

행동예측에는 휴면/이탈, 등급변동, 특정상품 구매, 특정금액 이상 구매, 특정 시점의 특정조건에 해당되는 행동 예측등 다양한 경우가 있을 수 있다. 이러한 특정행동 예측능력이 뛰어나면 뛰어날 수록 더욱 정교한 고객관계활동을 전개할 수 있다.

단, 이러한 행동예측이 행동의 결과를 모두 상식적으로 설명할 수 있어야 된다는 생각은 잘못된다고 생각한다. 우리는 일기예보(예측)를 듣고 믿고 대응하지만 왜? 무슨 알고리즘을 써서, 무슨 내용으로 인해 비가 오게될지, 눈이오게 될지 100mm시간당 비가 오는게 타당한것인지 알려고도 하지 않는다. 맞으면 되는것이다. 이해할 수 있는 논리를 제공하면 좋을것이다. 그러나 당연히 논리적으로 이해할 수 있는 결과가 나오리라 생각하거나 요구하지 말기를 바란다. 일부 노회한 현업이나 통계학자들은 설명가능해야 한다고 말할지 모르지만 당신이 왜 통계학교수인지를 통계학적으로 증명할 수 있을지 모르겠다. 어쩌면 당신의 어렸을적 행동은 절대로 통계학 교수가 될수 없다가 맞을지도 모른다. 그리고 당신이 현업의 부장이란것은 대학교때 행동으로는 도저히 말이 안되는 상황일수도 있다.

---

[4] "Big Data Analysis for CRM using R", 김경태

휴면/이탈 예측

휴면/이탈 예측은 이미 고객을 돌이킬 수 없는 상태까지 가기 전에 retention하기 위한 방안이고, 고객의 거래주기를 단축시키기 위한 방안이다. 휴면보다 더 악화된 상태가 이탈이므로 휴면 중심으로 설명하겠다. 휴면은 평균거래주기를 3~4배 이상 초과한 고객이나 다음 달에 거래가 없을 고객으로 정의할 수도 있다. 우선 1개월 휴면인 단기휴면의 경우 예측이 다소 어려우나 매우 도움이 된다. 거래주기에도 연관될 수 있고 어찌했건 간에 예를 들어 최근 3개월간에 거래가 있었으나 다음 달에 거래가 없을 고객이다. 다음달에 어차피 거래가 없을 고객에게 굳이 마케팅을 할 필요는 없다. 그러나 거래주기가 평균 3개월인 경우 이들 집단을 거래를 유도하도록 거래가망성이 높은 고객을 대상으로 캠페인을 해야 한다. 장기 휴면은 거래주기를 한참 벗어나 이제는 1년간 거래가 없을 고객이 될 수 있다. 이런 고객은 미리 파악하여 고객에게 정보를 제공하거나 강한 오퍼를 통해 고객에게 우리의 상품과 서비스에 대한 매력을 전달 할 수 있어야 된다. 장기 휴면고객은 크게 2가지다. 이미 6개월 이상 거래가 없어서 장기휴면가능성이 매우 높은 고객과 최근 3개월간 거래는 있었으나 향후 1년간 거래가 없을 고객으로 분류할 수 있다. 이 기준은 모든 산업에 해당되지는 않으나 분류를 2가지로 나눌 수 있다. 이런 경우 상식적인 장기휴면을 예측하는것은 의미가 없다고 생각할 수 있으나 효율성 측면에서 의미가 있고 최근 3개월 거래가 있었던 일반적인 고객이나 1년간 거래가 없을 고객은 매우 유용한 정보이다. 이러한 고객들은 최근 3개월, 6개월, 1년간의 거래 패턴이 이미 휴면이 되지 않을 고객과 다르다. 구매주기가 길거나 상품을 구매하는 다양성이 낮거나 구매금액이 감소하는 성향이 있다든지, 거래하는 매장이 일정 매장으로 단순하다든지의 형태를 보인다. 결국 자사의 상품과 서비스에 대한 확신이 없는 고객들이다. 따라서 이러한 고객들에게는 적합한 개인화된 상품/서비스에 대한 추천이나 정보제공이 중요하다.

등급변동 예측

등급변동은 어쩌면 자연스러운 현상일지는 모르나 기업의 입장에서는 등급유지율과 등급상승율이 개선되기를 바랄것이다. 아니면 최소한 등급유지율이 높기라도 하면 좋은데 많은 고객이 등급하락을 한다면 문제가 될것이다. 이런 경우 등급하락 고객을 예측하여 재구매나 up-sell을 통해 등급을 유지하고 등급상승 가망고객에 대해서는 보다 더 상승할 수 있도록 동기부여를 함이 필요하다.

상품구매 예측

상품구매 예측은 개인화된 상품추천에 참 중요한 요소이다. 천만명의 고객이 있는데 상품은 100개 정도가 있다고 하자. 이럴때 100가지 상품 중 가장 적합한 상품을 고객에게 개인화 하여 추천하려고 한다면 어떻게 해야 할까? 단순 교차판매의 사례가 아니라 고객이 구매할 만한 상품을 전체적으로 바라봐야 한다. 실제로 이러한 접근방법이 일본에서는 많은 사례가 있다. 고객별로 구매할 상품 best 5가지를 scoring에서 선택하여 추천을 하는것이다. 우리나라는 이러한 점에서는 매우 떨어진다. 특히, 온라인 서적에서의 상품추천을 보면 만족도 가 iTunes 보다 못함을 확실히 알게된다.

캠페인 반응예측

고객이 300만명이 있는데 이번달에 10만명을 대상으로 캠페인할 예산이 있다고 가정하자. 누구를 대상으로 캠페인을 해야 할까? 어떤 이들은 캠페인에 반응할 가능성이 높은 고객을 타겟팅하는것은 당연히 올사람을 타겟팅하는것이므로 추가수익 효과가 없다고 한다. 한

25%는 맞는 말이다. 그러나 반응할 가망성이 높은 고객에게 캠페인을 해야 반응을 더 잘하게 되고 의사결정할 임계치를 넘어서게 되는것이다. 그리고 일부는 캠페인을 안해도 자발적 구매를 할 사람이 섞여 있을것이다. 그렇다고 자발적 구매를 할 사람만 제외해야 할나? 아니면 관계관리를 위해 가끔 해줘야 하나? 일단 캠페인을 해야 반응할 사람과 자발적 구매가능성이 낮은 고객을 대상으로 진행하는것이 적합하다. 이러한 경우 반응율은 2배 정도 개선된다.

2.1.2 party를 이용한 Decision Tree[5]

본격적인 마이닝을 시작해 보기 전에 한가지 언급하고자 하는것은 실행할때 결과가 예시와 다를 수 있다. 이는 random한 효과때문에 그런것으로 random seed number를 지정하고 시작하면 언제나 같은 결과를 얻을 수 있다.

party package의 핵심은 ctree로 사용하기 편한 다양한 classification package 중의 하나이다. 현재 이 package의 문제점으로는 missing value를 잘 처리하지 못하는 사항으로 가끔 왼쪽이나 오른쪽으로 missing이 분류되는 사항이 있다. 그리고 ctree에 투입된 데이터가 표시가 안되는 경우 또는 predict가 실패하는 경우와 categorical variable의 test data가 train과 다르게 처리되는 문제가 있다.

party를 이용한 분석에서는 iris 데이터를 이용하고자 한다. 이 데이터는 Species 별 Sepal(꽃받침 조각)과 Petal(꽃잎)관련된 Sepal.Length, Sepal.Width, Petal.Length, Petal.Width 데이터로 구성되어 있다. Species는 크게 3가지 종류로 구성되어 있다.

```
> library(party)
> data(iris)
> summary(iris)
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
Median :5.800   Median :3.000   Median :4.350   Median :1.300
Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
setosa    :50
versicolor:50
virginica :50
```

Summary를 보면 missing value가 없고 Species가 각각 50개로 구성되고 Species를 분류하는데 사용될 데이터는 연속형으로 4개가 있음을 알 수 있다.

```
> data(iris)
```

사용할 데이터를 지정해서 로딩한다.

```
> ind <- sample(2,nrow(iris),replace=TRUE, prob=c(0.7,0.3))
```

모델 개발을 위해 학습을 할 데이터와 학습을 통해 만든 모델을 검증하는 테스트 데이터를 70%, 30% 할당하기 위해 ind변수에 1과 2값을 할당한다.

```
> trainData <- iris[ind==1,]
```

---

[5] http://www.rdatamining.com/docs

> testData <- iris[ind==2,]
iris 전체 데이터 중에서 ind에 따라 학습 데이터와 테스트 데이터를 구분한다. 일반적으로 학습과 테스트는 7:3으로 하나 데이터가 전체적으로 부족한 경우 8:2로 하면 성능이 향상된다.

> myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
모델링에 타겟변수와 타겟을 분류하는데 사용할 변수들을 공식으로 정의하는데 변수가 많거나, 이미 모델링을 위한 데이터를 선별해서 작성한 경우 "~ ." 처럼 표현해서 모든 변수를 투입한다.
> iris_ctree <- ctree(myFormula, data=trainData)
ctree 기능을 이용해서 학습용 데이터를 해당공식에 맞게 모델을 만들어서 저장한다.
> table(predict(iris_ctree),trainData$Species)
학습용 데이터에 따른 모델의 성과를 확인하기 위해 예측된 데이터가 원래 학습용 데이터에 있는 Species와 어떻게 차이가 있는지 확인한다. 좌측은 예측값이고 상단은 실제값이다. 따라서, vesicolor라고 예측한 것 36개 중에 실제로 vericolor는 34개이고 2개는 virginica이다. 전체적으로 보면 원래값 대로 제대로 예측했음을 알 수 있다.

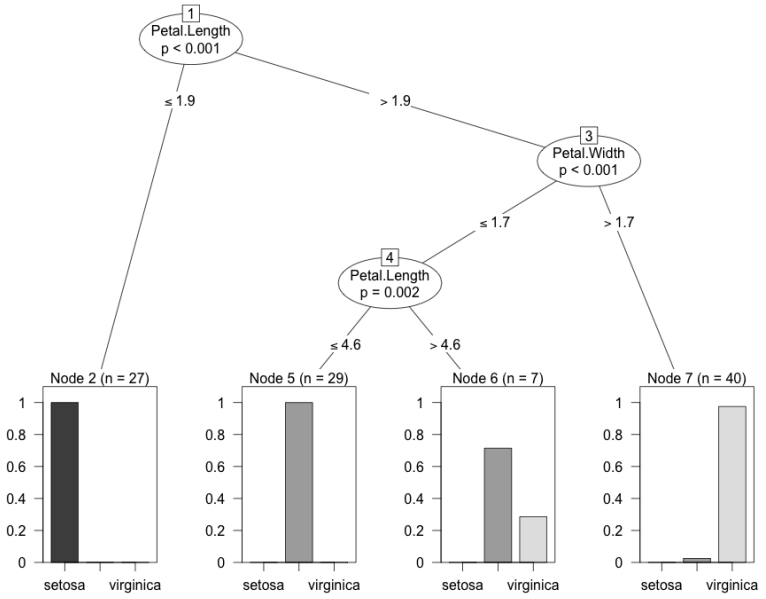
```
            setosa versicolor virginica
setosa          27          0         0
versicolor       0         34         2
virginica        0          1        39
```

> print(iris_ctree)

```
        Conditional inference tree with 4 terminal nodes

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  103

1) Petal.Length <= 1.9; criterion = 1, statistic = 95.084
  2)*  weights = 27
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 52.481
    4) Petal.Length <= 4.6; criterion = 0.998, statistic = 12.375
      5)*  weights = 29
    4) Petal.Length > 4.6
      6)*  weights = 7
  3) Petal.Width > 1.7
    7)*  weights = 40
```

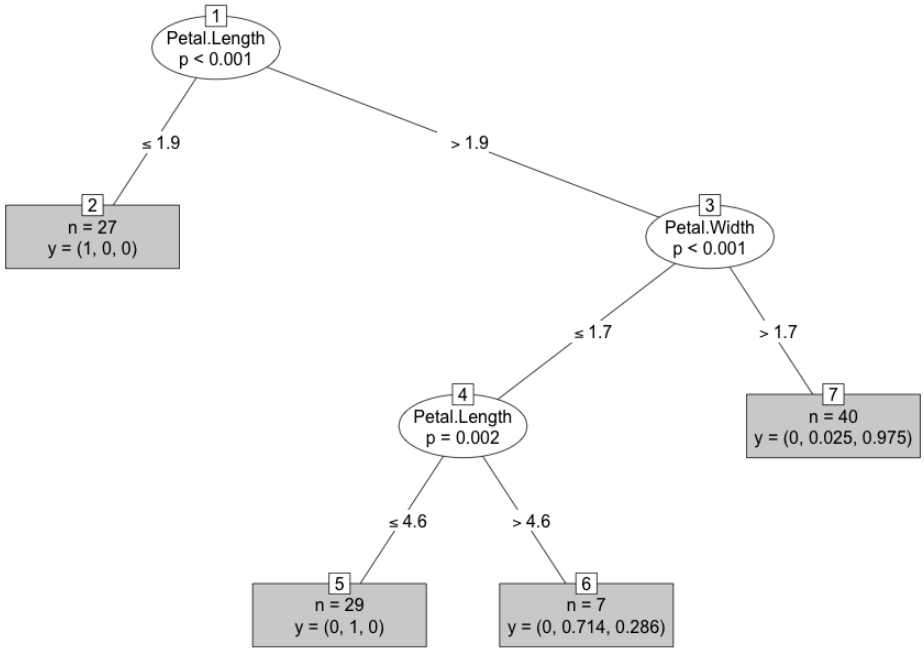
학습용 데이터로 개발된 모델의 산출물을 text형식으로 보면 위와 같은 구조의 조건문으로 표시되며 4개의 끝단 노드를 갖은 tree로 표현이 가능하다.

> plot(iris_ctree)



개발된 모델을 그래프로 표현하면 위와 같고, 이 내용을 보다 단순하게 표시하면 아래와 같다. 이러한 내용을 통해 보다 visual하게 어떤 변수가 크게 분류를 하고, 어떤 조건인 경우에 어떠한 class가 주로 존재하는지를 구조적으로 파악할 수 있다.

> plot(iris_ctree, type="simple")



> testPred <- predict(iris_ctree, newdata=testData)
개발된 모델을 이용해서 테스트 데이터에 적용해 최종 검증을 한다.
> table(testPred, testData$Species)

```
testPred     setosa versicolor virginica
  setosa         23          0         0
  versicolor      0         15         3
  virginica       0          0         6
```

테스트 데이터에서의 accuracy를 분석한다.

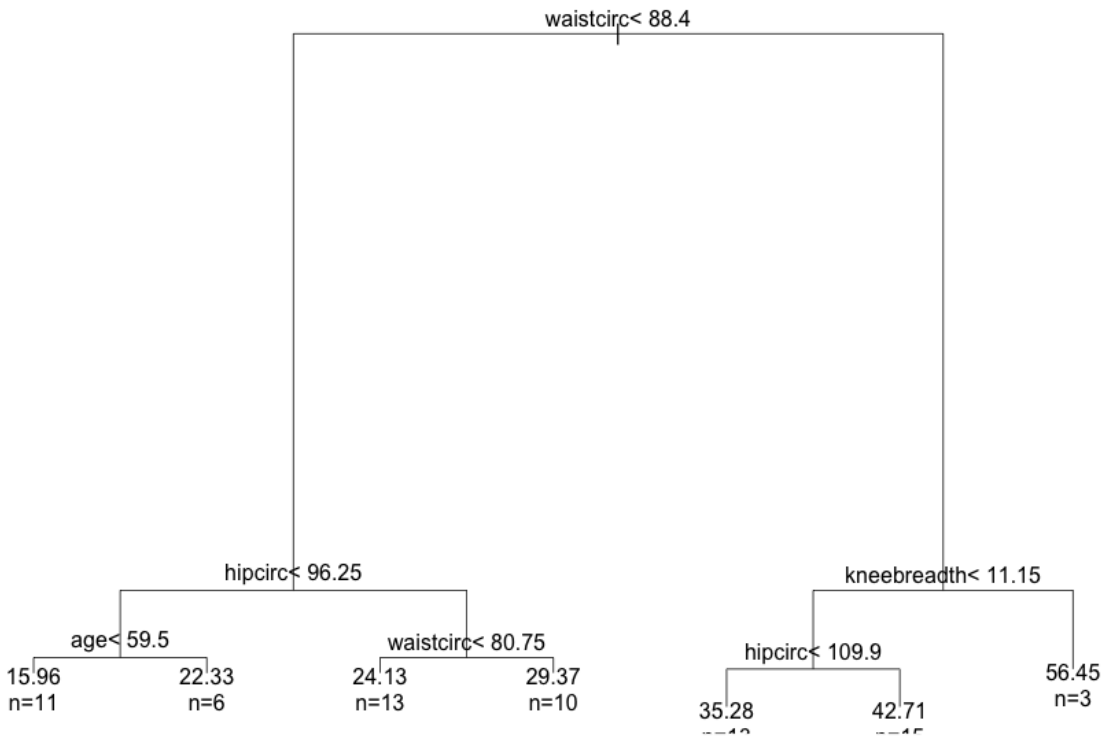
2.1.3 rpart를 이용한 Decision Tree[6]

rpart는 Recursive Partitioning and regression tree로 CART like한 tree를 제공한다. tree의 prediction error를 최소화 하는데 있다.

```
> library(rpart)
> data("bodyfat", package = "mboost")
> dim(bodyfat)
[1] 71 10
> attributes(bodyfat)
$names
 [1] "age"         "DEXfat"      "waistcirc"   "hipcirc"     "elbowbreadth"
 [6] "kneebreadth" "anthro3a"    "anthro3b"    "anthro3c"    "anthro4"

$row.names
 [1] "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
[15] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"
[29] "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"
[43] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"  "100" "101" "102"
[57] "103" "104" "105" "106" "107" "108" "109" "110" "111" "112" "113" "114" "115" "116"
[71] "117"

$class
[1] "data.frame"

> bodyfat[1:5,]
   age DEXfat waistcirc hipcirc elbowbreadth kneebreadth anthro3a anthro3b anthro3c anthro4
47  57  41.68     100.0   112.0         7.1         9.4     4.42     4.95     4.50    6.13
48  65  43.29      99.5   116.5         6.5         8.9     4.63     5.01     4.48    6.37
49  59  35.41      96.0   108.5         6.2         8.9     4.12     4.74     4.60    5.82
50  58  22.79      72.0    96.5         6.1         9.2     4.03     4.48     3.91    5.66
51  60  36.42      89.5   100.5         7.1        10.0     4.24     4.68     4.15    5.91

> myFormula <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth
> bodyfat_rpart <- rpart(myFormula, data = bodyfat,control=rpart.control(minsplit=10))
> attributes(bodyfat_rpart)
$names
 [1] "frame"     "where"     "call"      "terms"     "cptable"   "splits"    "method"
 [8] "parms"     "control"   "functions" "y"         "ordered"
```

[6] http://www.rdatamining.com/docs

```
$class
[1] "rpart"

> print(bodyfat_rpart)
n= 71

node), split, n, deviance, yval
      * denotes terminal node

 1) root 71 8535.98400 30.78282
   2) waistcirc< 88.4 40 1315.35800 22.92375
     4) hipcirc< 96.25 17  285.91370 18.20765
       8) age< 59.5 11   97.00440 15.96000 *
       9) age>=59.5 6   31.45788 22.32833 *
     5) hipcirc>=96.25 23  371.86530 26.40957
      10) waistcirc< 80.75 13  117.60710 24.13077 *
      11) waistcirc>=80.75 10   98.99016 29.37200 *
   3) waistcirc>=88.4 31 1562.16200 40.92355
     6) kneebreadth< 11.15 28  615.52590 39.26036
      12) hipcirc< 109.9 13  136.29600 35.27846 *
      13) hipcirc>=109.9 15   94.46997 42.71133 *
     7) kneebreadth>=11.15 3  146.28030 56.44667 *
> plot(bodyfat_rpart)
> text(bodyfat_rpart, use.n=TRUE)
```



개인적으로는 party의 결과가 보기 편해서 rparty보다 주로 사용한다.

2.1.3 Random Forrest[7]

randomForests package는 random input에 따른 forest of tree를 이용한 분류방법으로, random한 forest에는 많은 tree들이 생성되는데 새로운 object를 분류하기 위해 forest에 있는 tree에 각각 투입하여 각각의 tree들이 voting을 함으로써 분류를 하는 방식이다. 이 방식은 대용량 데이터에서 효율적으로 실행되며, 수천개의 변수를 이용해서 변수제거 없이 실행되어 정확도 측면에서 좋은 성과를 보인다. 특히 unbalanced된 class의 모집단에 대해 잘 대응한다.

이를 사용하는데 제약점은 각 category variable의 value 종류가 32개를 넘을 수 없다는 것이다. 이에 대한 대안으로 party package의 cforest를 사용하면 된다.[8]

```
> library("randomForest")
> data(iris)
> ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7,0.3))
> trainData <- iris[ind==1,]
> testData <- iris[ind==2,]
> rf <- randomForest(Species ~ .,data=trainData, ntree=100, proximity=TRUE)
```

최대 tree갯수를 100으로 지정하고 다양한 tree분할을 시도한다.

```
> table(predict(rf), trainData$Species)

            setosa versicolor virginica
  setosa        37         0         0
  versicolor     0        38         3
  virginica      0         1        31
```

```
> print(rf)

Call:
 randomForest(formula = Species ~ ., data = trainData, ntree = 100,      proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

        OOB estimate of  error rate: 3.64%
Confusion matrix:
           setosa versicolor virginica class.error
setosa         37         0         0  0.00000000
versicolor      0        38         1  0.02564103
virginica       0         3        31  0.08823529
```

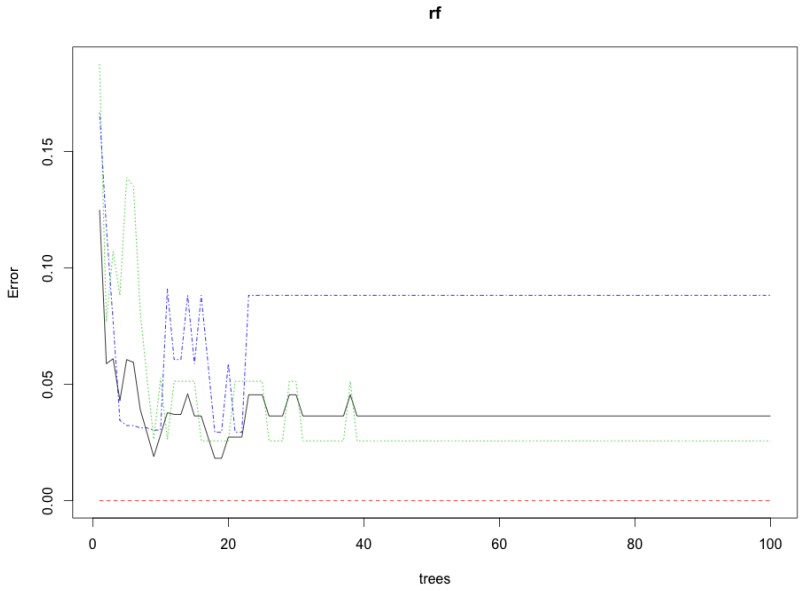
각 class에서의 오차율을 보여준다.

```
> plot(rf)
```

---

[7] http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm

[8] http://www.rdatamining.com/docs



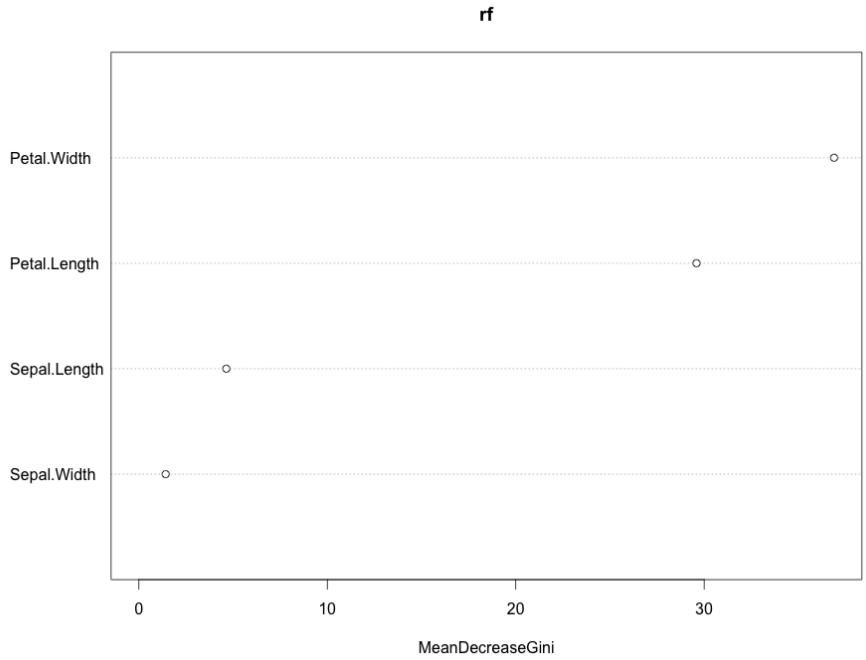
tree갯수에 따른 오차가 개선되는 것을 class별 및 전체 평균으로 보여주며 약 20개 정도에서 오차가 최소화 되고 그 이상인 경우 오차가 개선되지 않음을 알 수 있다.

> importance(rf)

	MeanDecreaseGini
Sepal.Length	4.640239
Sepal.Width	1.420328
Petal.Length	29.593030
Petal.Width	36.892525

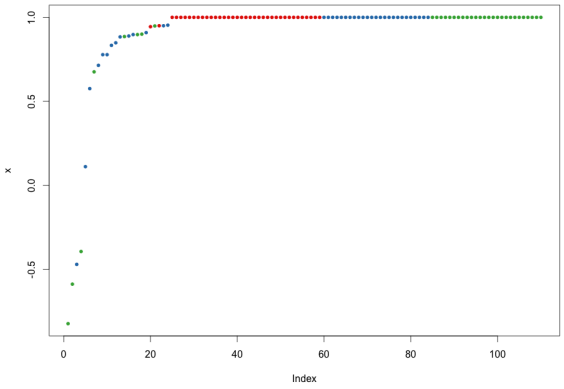
변수별 중요도 값으로 Petal.Width가 class를 분류하는데 가장 큰 영향을 주는것으로 나타났다.

> varImpPlot(rf)



변수 갯수가 많은 경우 그래프 형태로 보면 변수의 상대적 중요도를 쉽게 파악할 수있다.

```
> irisPred <- predict(rf, newdata=testData)
> table(irisPred, testData$Species)
irisPred     setosa versicolor virginica
  setosa         13          0         0
  versicolor      0         10         1
  virginica       0          1        15

> plot(margin(rf,testData$Species))
```

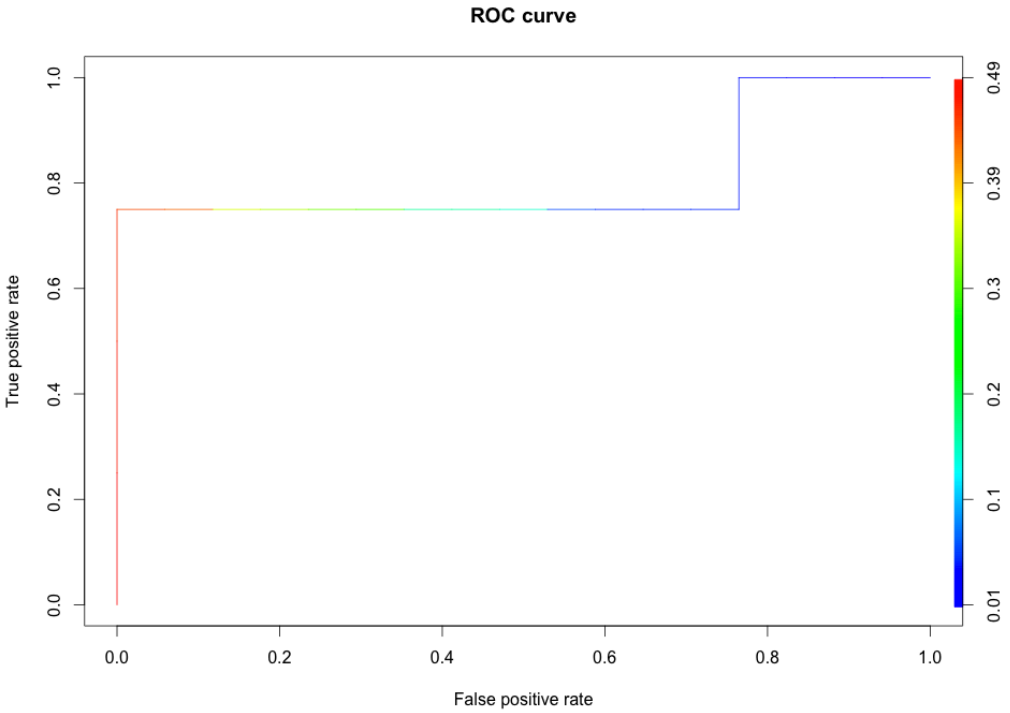


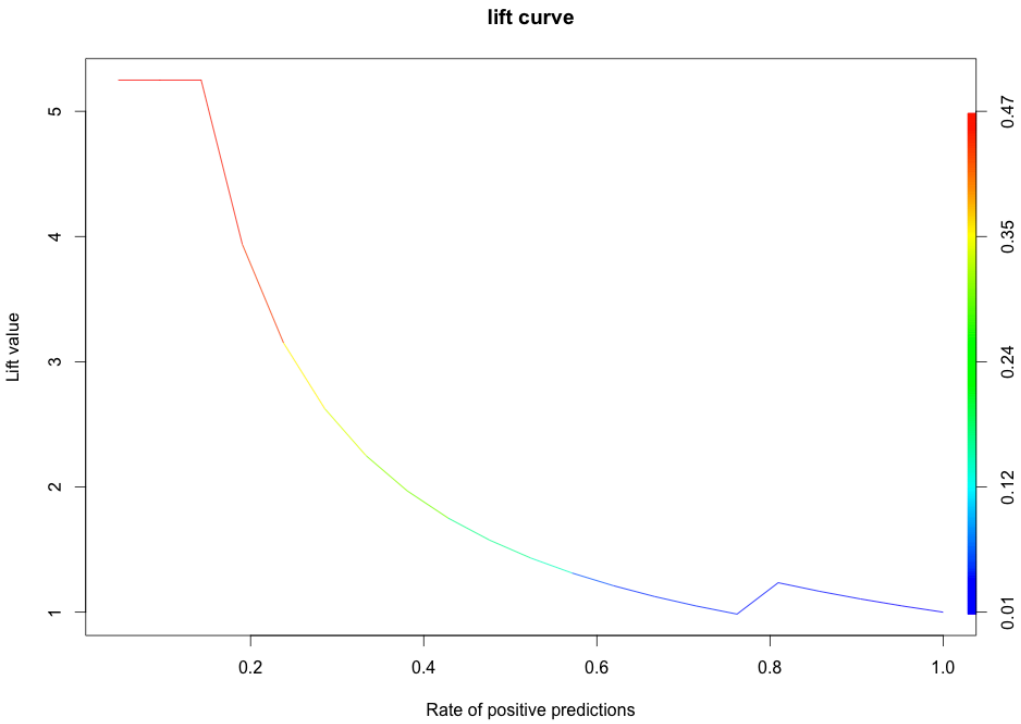
2.1.4 ROCR package를 이용한 Performance Analysis

Performance Analysis를 위해서는 다양한 분석이 이루어지는데 ROC analysis와 Lift analysis에 대해서 알아보겠다. 한가지 고려해야 할점은 ROCR package는 binary classification 만 지원한다는 것이다.

```
> # install.packages('party')
> # install.packages('ROCR')
>
> # Load the kyphosis data set.
> library(rpart)
>
> x <- kyphosis[sample(1:nrow(kyphosis), nrow(kyphosis), replace = F),]
> x.train <- kyphosis[1:floor(nrow(x)*.75), ]
> x.evaluate <- kyphosis[(floor(nrow(x)*.75)+1):nrow(x), ]
>
> library(party)
> x.model <- cforest(Kyphosis ~ Age + Number + Start, data=x.train,
+                    control = cforest_unbiased(mtry = 3))
>
>
> # x.model <- ctree(Kyphosis ~ Age + Number + Start, data=x.train)
> # plot (x.model)
> x.evaluate$prediction <- predict(x.model, newdata=x.evaluate)
> x.evaluate$correct <- x.evaluate$prediction == x.evaluate$Kyphosis
> print(paste("% of predicted classifications correct", mean(x.evaluate$correct)))
[1] "% of predicted classifications correct 0.80952380952381"
>
```

```
> x.evaluate$probabilities <- 1- unlist(treeresponse(x.model,newdata=x.evaluate), use.names=F)
[seq(1,nrow(x.evaluate)*2,2)]
>
> library(ROCR)
> pred <- prediction(x.evaluate$probabilities, x.evaluate$Kyphosis)
> perf <- performance(pred,"tpr","fpr")
> plot(perf, main="ROC curve", colorize=T)
>
> perf <- performance(pred,"lift","rpp")
> plot(perf, main="lift curve", colorize=T)
```





2.2 Regression

다음 버전에서 다루고자 함.

Chapter 3

Clustering

3.1 Clustering 개요[9]

"데이터 분석을 하기 전에 전체적인 특성을 파악하기에 매우 좋은 방법이다. 일단 샘플링을 하여 빠른시간 내에 전체 특성을 파악하면 효율적이다."정확도가 높은 모델개발 보다 많은 업무에 적용할 수 있는 좋은 데이터 마이닝 모델의 적용이 중요하다. 이를 위해선 자동화가 필요하다."

세분화는 고객을 특성에 따라 서로다른 여러개의 배타적인 집단으로 나누는것을 말한다. 세분화는 집단을 나누는 방법에 따라 통계적 기법을 쓰는지, 임의적으로 나누는지에 따라 크게 두가지 방법으로 구분된다. 임의적 방법은 논란의 여지가 많으나 많이 사용되어 왔고, 통계적 기법은 1,2세대 알고리즘을 이용하여 많이들 사용되어 왔으나 실무적용성에 대한 논란이 많았다. 그래서 최신 세분호 방법과 프로파일링 방법 및 활용방안에 대해서 논의해 보겠다.

3.1.1 기존 세분화 방법의 유형

우선 임의로 나누는 방법으로는 고객등급과 고객구분(신규/기존) 두가지 변수로 나누거나 현재가치와 미래가치를 기준으로 4분면이나 9개의 집단으로 나누는 방법 등 다양한 방식이 있을 수 있다. 다음으로 통계적 기법을 이용한 방법은 흔히 clustering, k-means 등으로 말할 수 있다.

3.1.2 전통적 세분화 방법의 문제점

전통적인 세분화 방법은 변수를 선정하고 변수를 구간대를 나눈다음에 이를 기준으로 격자형으로 단순히 나눈 다음에 집단이 적으면 merge하는 방식이 있고 단순 clustering, k-menas가 있다. 문제는 단순 격자형은 작업을 하는데 오랜 시간이 걸리고 후처리로 merge를 할때 원칙이 명확하지 않다는 것이다. 분리된 격자 cell의 profile을 보고 유사한 근처 집단으로 나누어야 하는데 집단간에 profile이 차이가 안나는 경우가 있다. 물론 세분화 변수와 프로파일링 변수는 달라야 된다. 그리고 격자나 cluster, k-means의 공통적인 문제는 변수의 특성으로 인한 변동에 의해 의미없이 고객집단이 이동하게 된다는 것이다. 처음에는 A, B, C 집단이 존재했는데 다음 달에는 B, D, F 집단이 존재하고 특성이 완전히 변하게 되거나 A, B, C 집단이 유지가 되도 분포가 매우 틀리고 A, B, C1으로 약간 프로파일 자체가 변하는 일이 생긴다. 이런 경우 세분화를 안정적으로 관리하면서 전략을 수립하고 액션을 할 수 없고, 자연스런운 변화에 의미가 있는 것처럼 이리저리 끌려다니게 된다. 특히 세분집단의 수가 많은 경우 더 심해 진다.
이러한 문제를 해결하기 위해 나온게 k-menas에 SRM을 결합한 방식으로 세분집단의 변화가 마구 변하지 않는다. 집단은 안정적으로 유지되며 해당 집단에 속한 고객이 변하는 것이다. 이래야 세분화를 통한 고객관리가 가능해진다.

---

9 "Big Data Analysis for CRM using R", 김경태

3.1.3 Target-based 세분화 방법

타겟기반 세분화는 고객가치 또는 특정상품을 구매하는 고객을 타겟으로 해서 세분화를 하는 내용으로 해당 집단이 많이 존재하는 집단과 그렇지 않은 집단으로 구분되며, 이러한 집단들도 다른 변수들에 의해 집단의 특성이 구분된다. 예를 들어 A, B, C 집단으로 고객가치 기준으로 세분화 된 경우 우수고객이 A에는 60%, B는 30%, C는 1% 존재할 수 있다. 또는 A는 40% B는 35% C에는 0% 일수 있는데 A와 B집단의 차이는 우수고객 비중은 유사하나 구매주기가 다를 수 있다.

3.1.4 Profiling 방법

프로파일링은 참 너무나 많은 사기극이 벌어진 분야다. 격자방식의 세분화를 흔히 전략하시는 분들이 많이 사용한다. 이러한 방식은 프로파일링을 한 경우 집단간의 차이가 세분화 기준변수에 의해서 발생하지 프로파일링 변수로 미리 설정한 변수 또는 이후에 프로파일링을 시도한 변수에 의해 집단간에 차이가 나지 않는 경우가 있다. 프로파일링은 집단간에 동일한 변수로 할 수도 있고 서로다른 변수로 집단의 특성이 규명 될수도 있다. 그러나 공통적인것은 동일변수를 기준으로 집단을 비교할때 집단간에 차이가 명확할 수 있다는 것이다. 그러한 변수들이 프로파일링 변수로 선정되게 되고 집단별로 유의미한 변수가 다를 수 있다. 그러나 격자방식에서는 집단간에 차이가 안나거나 프로파일링 변수가 안나올 수가 있다. 때에 따라서는 유의미하지 않은 변수로 집단간에 차이가 있다고 몰아가기도 한다. 이런 우스운 사례가 발생되지 않으려면 누가 해도 동일한 결과가 나온는 프로파일링 기법이 있어야 된다.

따라서 자동화된 방식으로 세분화가 되고 프로파일링이 되어야 동일한 데이터에 대해 일관된 품질의 결과가 나오게 된다. 그리고 세분화가 이루어진 집단의 프로파일링은 세분집단 별로 그 집단을 변별하는 가장 유의미한 변수 순서로 표시가 되어야 한다. 그것도 자동으로 이루어 져야 된다. 그리고 고정된 변수로 다양한 세분집단을 비교할 수 있어야 된다.

3.1.5 세분화 수행기간

일반적으로 세분화 수행하고 보고서를 작성하는데 마트를 준비에서 부터 총 2개월은 걸린다. 매우 낭비적인 방식이고 도중에 여러번 세분화를 다시하곤 한다. 3개월 전략 프로젝트에서 세분화 기법의 적용에 2개월이 걸리면 전략하는 사람은 무엇을 해야 하나? 근본적으로 말이 안되는 경우다. 저자가 언급하는 세분화는 데이터 입수가 된 순간에서 마트 생성에 반나절에서 1일 정도 사용하고 세분화 및 보고서 작성에 1일이면 된다.

3.2 K-means

```
> data(iris)
> newiris<-iris
```

K-means 분석을 위해 데이터를 newiris로 복사한다.

```
> newiris$Species <-NULL
```

복사한 데이터를 세분화 했을때 세분집단이 Species를 잘 분류해 주었는지를 알고자 Species에 NULL값으로 초기화 한다.

```
> kc <- kmeans(newiris,3)
```

Species가 3개가 있으므로 일단 이러한 아이디어를 갖고 최적의 값이 3개라는 사전지식하에 시도해 본다. 실제 업무에서는 unsupervised 분석인 경우 최적의 k값을 선정하는게 필요하다.

```
> kc
K-means clustering with 3 clusters of sizes 50, 62, 38

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.006000    3.428000     1.462000    0.246000
2     5.901613    2.748387     4.393548    1.433871
3     6.850000    3.073684     5.742105    2.071053

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [43] 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2
 [85] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 2 2 3 3 3 2 3 2 3 2 3 3
[127] 2 2 3 3 3 3 3 2 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 15.15100 39.82097 23.87947
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

결과값을 인쇄해 보면 3개의 집단의 size와 각 집단의 변수들의 평균값, 각 observation이 어느 cluster에 속할지에 대한 값을 보여주며, 데이터를 추가적으로 볼 수 있는 변수들을 보여준다.

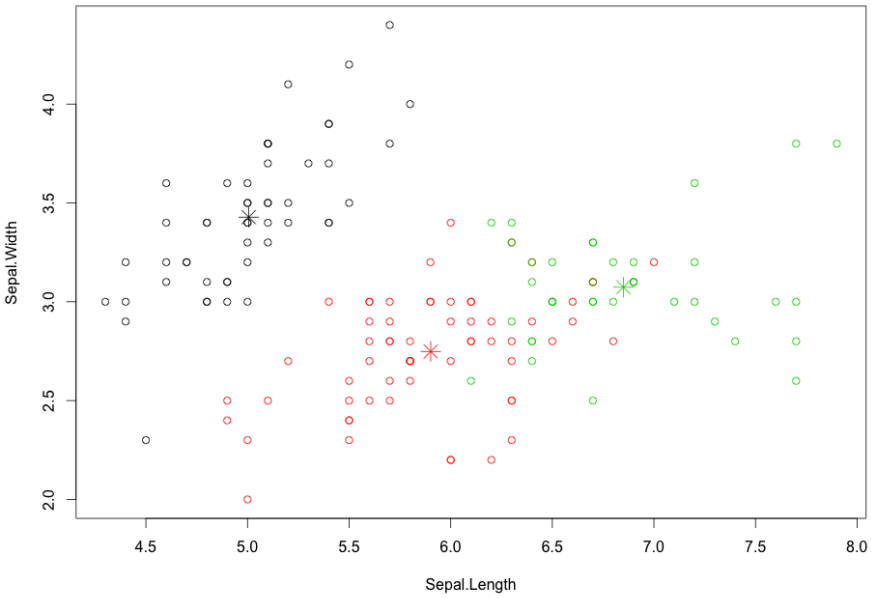
```
> table(iris$Species, kc$cluster)

            1  2  3
  setosa     50  0  0
  versicolor  0 48  2
  virginica   0 14 36
```

우리는 Species를 모르는 데이터를 이용해서 세분화한 결과가 매우 정확하게 Species를 분류하고 있음을 알 수 있다.

```
> plot(newiris[c("Sepal.Length","Sepal.Width")], col=kc$cluster)
```

k-means결과를 color를 다르게 해서 집단을 Sepal의 width와 length로 표시한다.

```
> points(kc$centers[,c("Sepal.Length","Sepal.Width")],col=1:3,pch=8,cex=2)
```



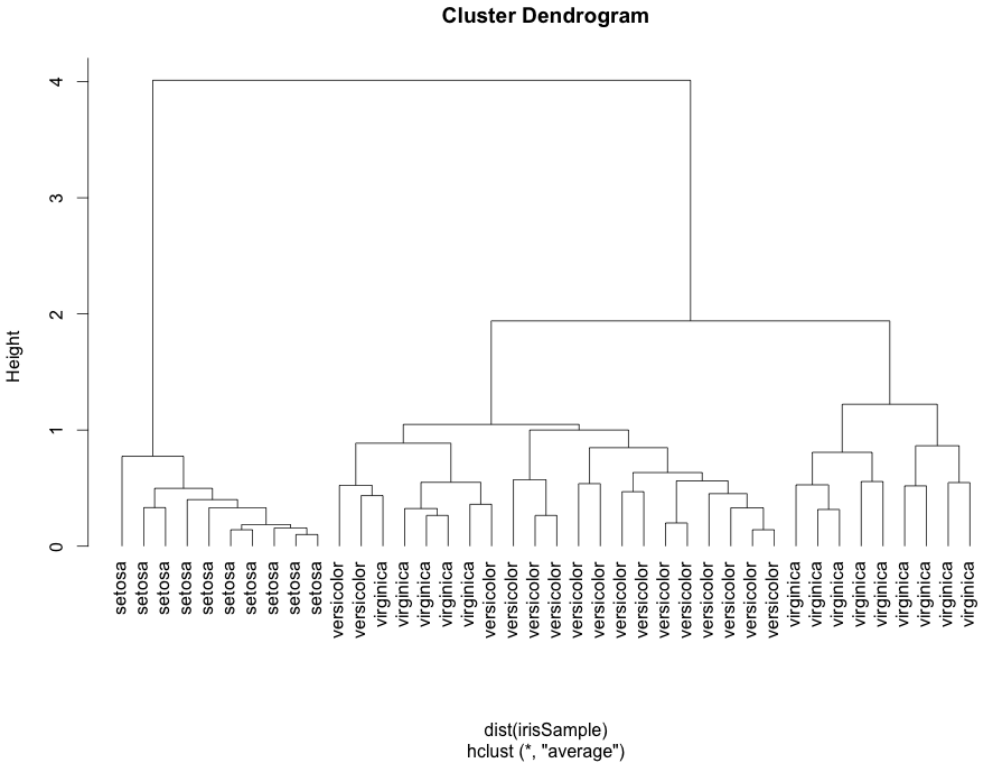
변수가 2개로 제한되어 일부 집단이 겹쳐지는 것으로 나타나지만 전체적으로 집단이 구분됨을 알 수 있다.

```
> kc4 <- kmeans(newiris,4)
> table(iris$Species, kc4$cluster)

            1  2  3  4
setosa      0 22 28  0
versicolor 48  0  0  2
virginica  14  0  0 36
```

일반적으로 k-means를 사용하는 경우 최적의 k값이 몇이냐가 정확도에 많은 영향을 준다. 따라서 이러한 데이터를 학습할때 일정 범위로 변화를 주어서 정확도가 어떻게 변하는지를 보면 된다. 위의 예에서는 k를 4로 한 경우 오류가 커짐을 알 수 있다.

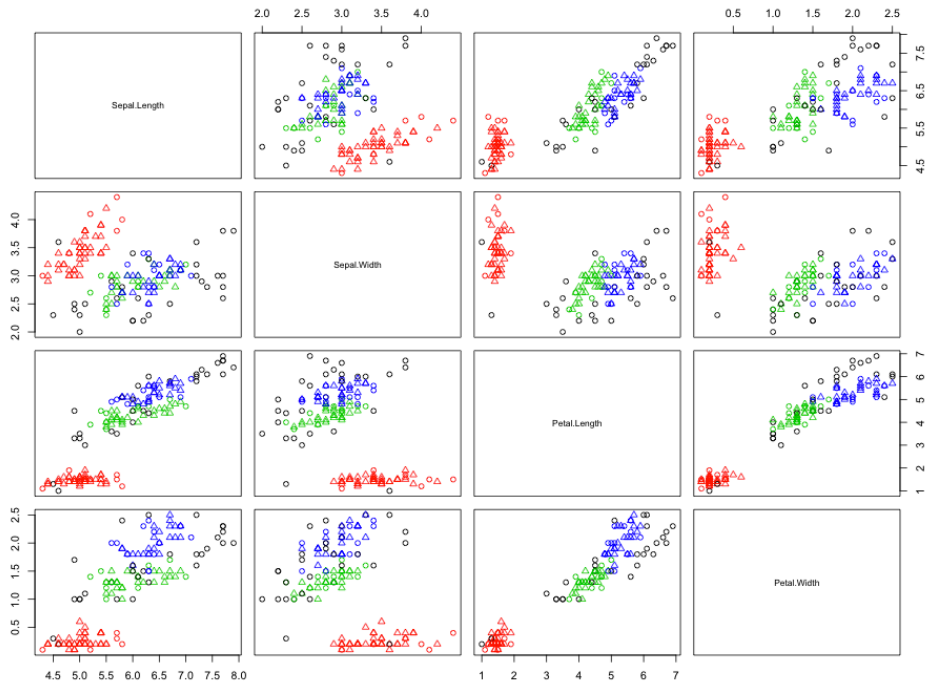
2.3 Hierarchical Clustering

```
> dim(iris)
[1] 150   5
> dim(iris)[1]
[1] 150
> dim(iris)[2]
[1] 5
> idx <- sample(1:dim(iris)[1], 40)
> irisSample <- iris[idx,]
> irisSample$Species<-NULL
> hc <- hclust(dist(irisSample),method="ave")
> plot(hc,hang= -1, labels=iris$Species[idx])
```

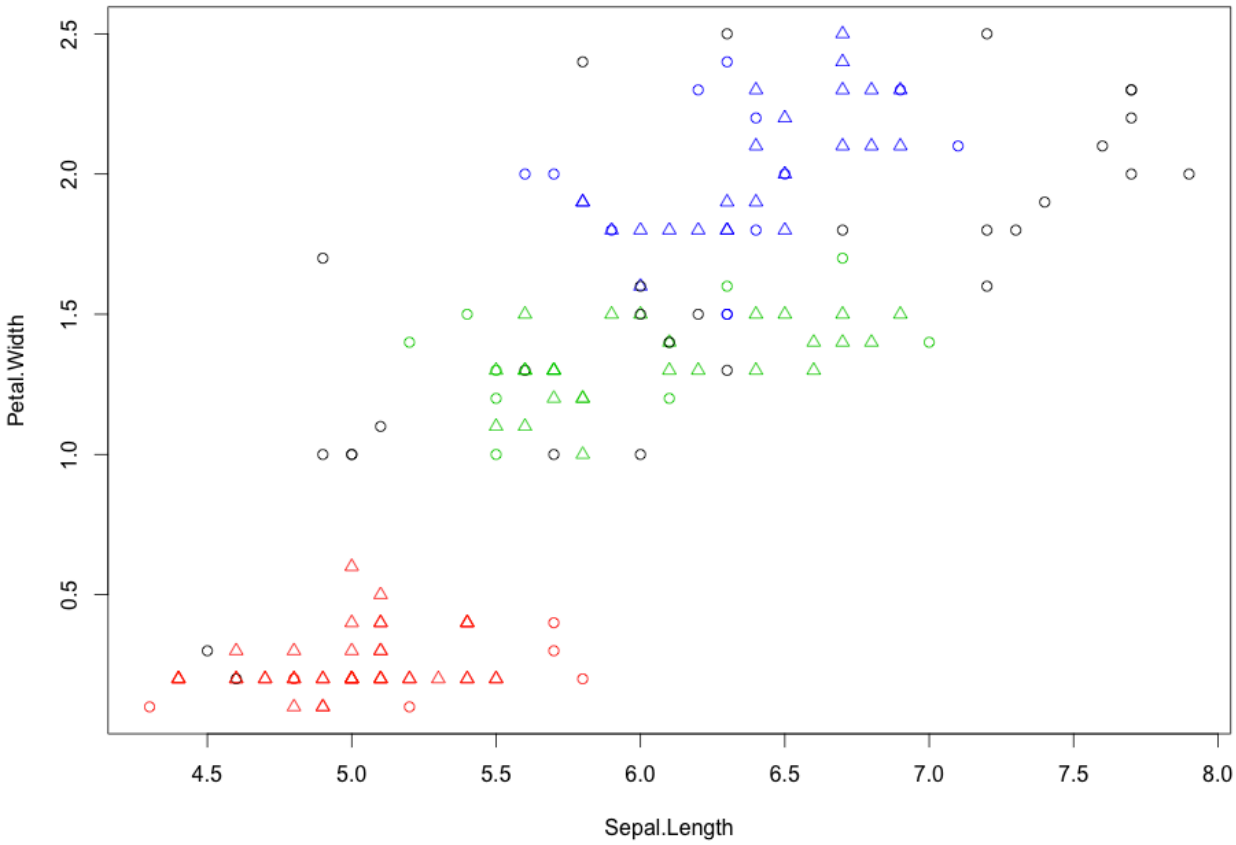


2.4 Density-based Clustering

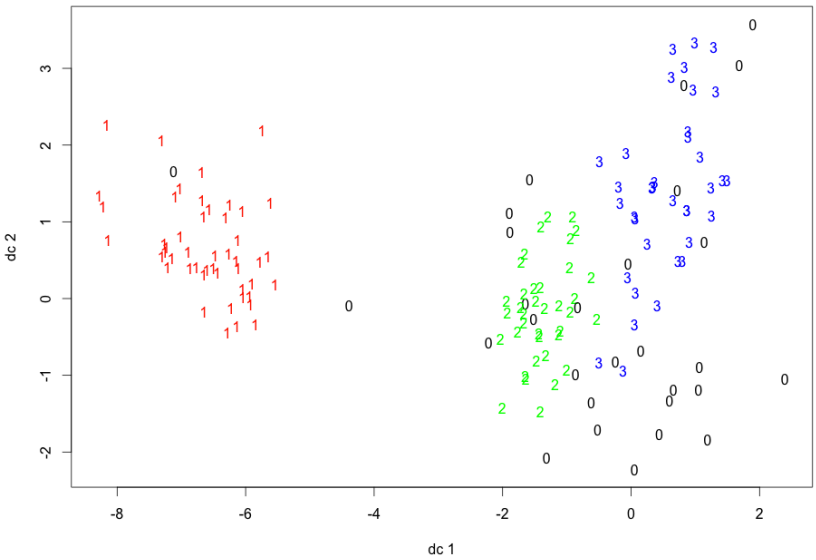
```
> library(fpc)
> newiris <- iris[-5]
> ds <- dbscan(newiris, eps=0.42, MinPts=5)
> table(ds$cluster,iris$Species)

   setosa versicolor virginica
0      2         10        17
1     48          0         0
2      0         37         0
3      0          3        33
> ds1 <- dbscan(newiris, eps=0.42, MinPts=10)
> table(ds1$cluster,iris$Species)

   setosa versicolor virginica
0      6         25        50
1     44          0         0
2      0         10         0
3      0         15         0
> ds2 <- dbscan(newiris, eps=0.5, MinPts=5)
> table(ds2$cluster,iris$Species)

   setosa versicolor virginica
0      1          6        10
1     49          0         0
2      0         44        40
> plot(ds,newiris)
```




```
> newiris[c(1,4)]
    Sepal.Length Petal.Width
1            5.1         0.2
2            4.9         0.2
3            4.7         0.2
4            4.6         0.2
5            5.0         0.2
6            5.4         0.4
7            4.6         0.3
> plot(ds,newiris[c(1,4)])
```



> plotcluster(newiris,ds$cluster)



> set.seed(435)
> idx <- sample(1:nrow(iris), 10)
> newData <- iris[idx,-5]
> newData <- newData + matrix(runif(10*4, min=0, max=0.2), nrow=10, ncol=4)
> # label new data
> myPred <- predict(ds, newiris, newData)
> # check the labels assigned to new data

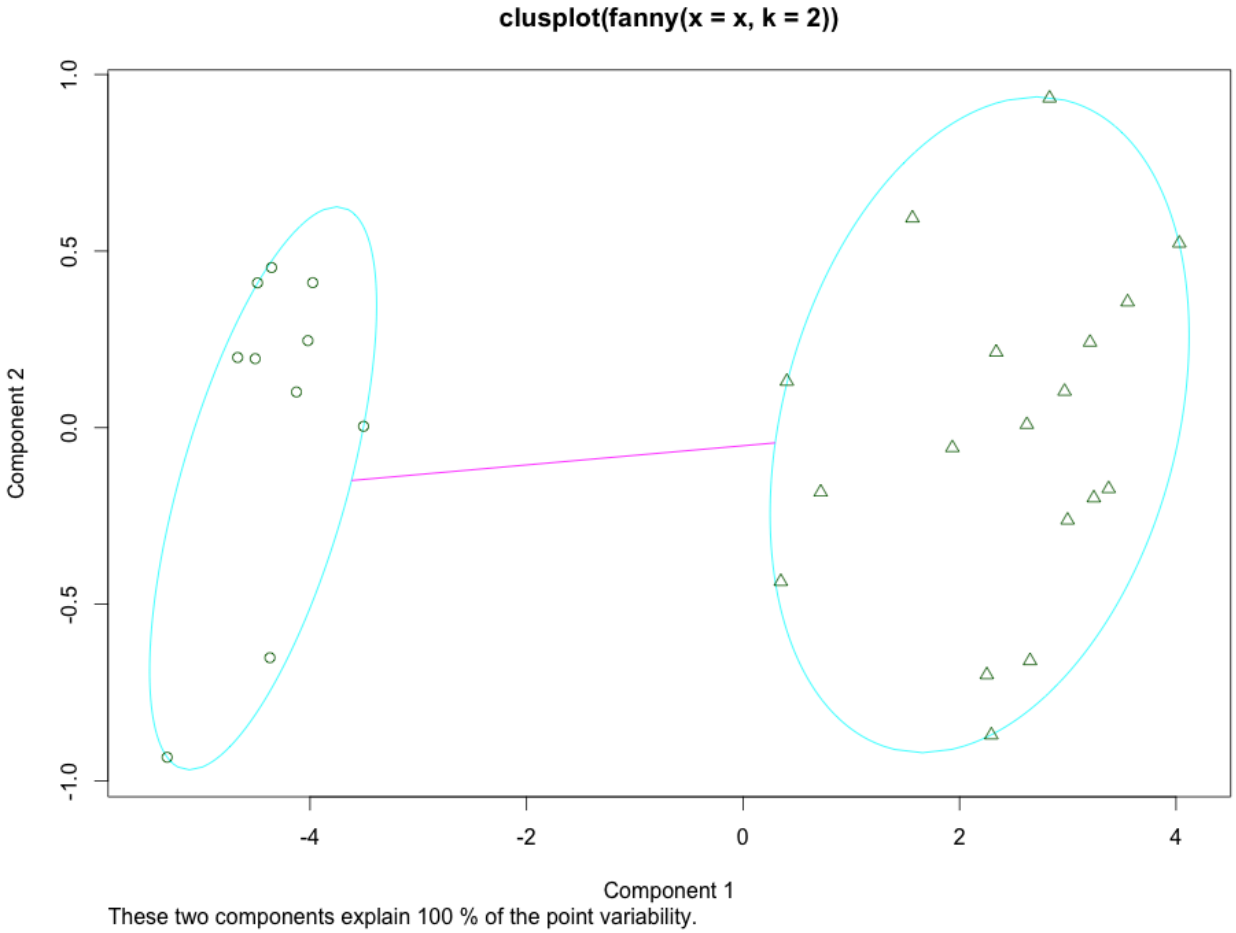
```
> plot(newiris[c(1,4)], col=1+ds$cluster)
> points(newData[c(1,4)], pch="*", col=1+myPred, cex=3)
> # check cluster labels
> table(myPred, iris$Species[idx])
myPred setosa versicolor virginica 0001 1300 2030 3012
```

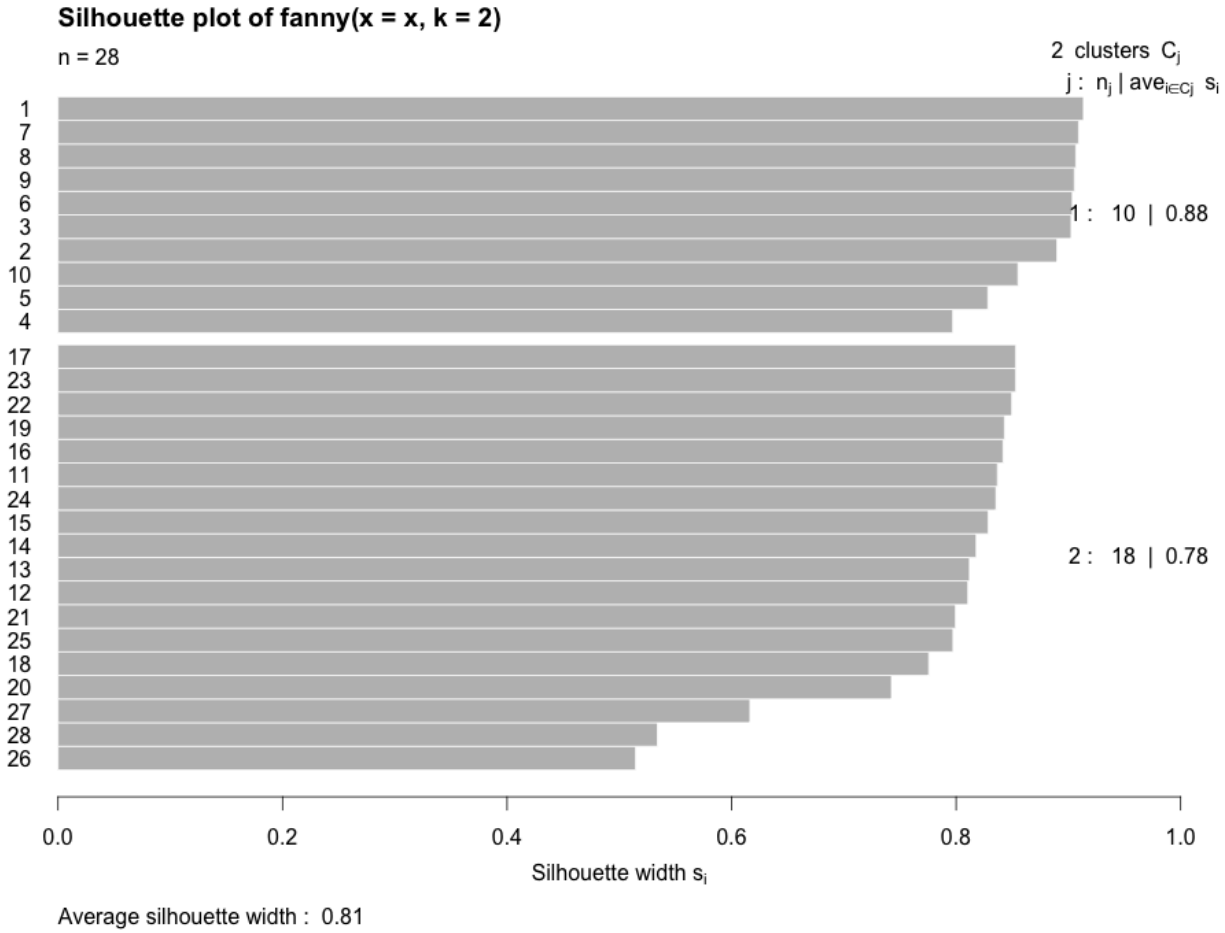
2.5 Fuzzy Clustering

특징으로는 numeric variable만 가능하고 NA가 허락된다는 점이며, k개의 cluster가 생성되는데 observation/2개 까지 가능하다.

```
library(cluster)
## generate 10+15 objects in two clusters, plus 3 objects lying
## between those clusters.
x <- rbind(cbind(rnorm(10, 0, 0.5), rnorm(10, 0, 0.5)),
           cbind(rnorm(15, 5, 0.5), rnorm(15, 5, 0.5)),
           cbind(rnorm( 3,3.2,0.5), rnorm( 3,3.2,0.5)))
fannyx <- fanny(x, 2)
## Note that observations 26:28 are "fuzzy" (closer to # 2):

> fannyx
Fuzzy Clustering object of class 'fanny' :
m.ship.expon.       2
objective     13.78395
tolerance       1e-15
iterations          9
converged           1
maxit             500
n                  28
Membership coefficients (in %, rounded):
      [,1] [,2]
 [1,]   97    3
 [2,]   96    4
 [3,]   97    3
 [4,]   87   13
 [5,]   91    9
 [6,]   96    4
 [7,]   97    3
 [8,]   97    3
 [9,]   97    3
[10,]   92    8
[11,]    5   95
[12,]    8   92
[13,]    8   92
[14,]    8   92
[15,]    6   94
[16,]    5   95
[17,]    4   96
[18,]   12   88
[19,]    5   95
[20,]   14   86
```

```
[21,]   9  91
[22,]   4  96
[23,]   4  96
[24,]   6  94
[25,]  10  90
[26,]  29  71
[27,]  23  77
[28,]  28  72
Fuzzyness coefficients:
dunn_coeff normalized
  0.851417   0.702834
Closest hard clustering:
 [1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

Available components:
 [1] "membership"  "coeff"       "memb.exp"    "clustering"  "k.crisp"     "objective"
 [7] "convergence" "diss"        "call"        "silinfo"     "data"
```





2.6 Subspace Clustering

다음 버전에서 다루고자 함.

Chapter 4

Association Analysis

4.1 Association Analysis 개요[10]

"가장 많이 언급되는 맥주를 사면서 기저귀를 산다는 규칙을 산출하는 방법이다. 그러나 매우 신기한 결과가 나오기를 기대하면 안된다."

연관성 분석은 흔히 Association Analysis, Market Basket Analysis, Sequence Analysis로 불리운다. Association Analysis가 포괄적 개념이고 한 장바구니에 무엇을 같이 사는지에 대한 분석이 Market Basket Analysis고 A를 산 다음에는 B를 산다가 Sequence Analysis이다. 전혀 다른 개념이다. 그러나 대부분 다른 목적으로 분석기법을 선택한 다음에 적용도 대충하는 성향이 있는데 이는 절대적으로 원하는 결과를 얻을 수 없다. 그리고 연관성 분석을 수행한 다음에 규칙을 선택하는 경우에도 산업의 특성에 따라 Support와 Confidence, Lift 값을 잘 보고 규칙을 선택해야 한다. 그리고 Sequence Analysis의 경우 Average Duration을 고려해서 적용하지 않으면 낭패를 보게 된다.

4.1.1 용어정의

용어정의를 공식이나 통계학적으로 설명하면 어려움을 느낄수 있는 경우가 클것 같아 서술적으로 표현하겠다. 정확한 표현이 아닐지라도 이해하는데는 도움이 되리라 생각된다. Support는 전체 거래중에서 거래 상세내용에 A, B 상품이 같이 있을 확률이다. 그만큼 많이 같이 판매되고 있다는 뜻으로 Association Rule이 나왔을때 적용성이 있는지를 판단할 수있고 불필요한 많은 분석을 줄일 수 있다. Confidence는 A를 구매하면 B를 구매한다는 조건부 확률에서 A, B가 같이 있는 건수중에 B가 있을 확률이다. 그만큼 해당 규칙을 적용하면 고객이 반응할 확률이 높아진다. Lift는 random하게 추천했을 때 대비 해당 규칙을 적용했을때 개선되는 비율이다. 1.5면 기존에 1%였을 때 해당 규칙을 적용하면 1.5% 전환율을 얻을 수 있다는 뜻이다. 이러한 용어가 중요한것은 association analysis를 수행할때 모든 경우의 수를 분석하는 것은 매우 불필요한 일이므로 minimum support를 정해서 rule을 도출한다. 처음에는 5% 정도로 임의 설정해서 산출해 보고, 속도와 의미가 현실적인지, 규칙은 충분히 나왔다고 생각하는지에 따라 1%로 조정을 한다든지의 다양한 시도가 필요하다. 처음부터 너무 낮은 support값을 선정해서 도출하는 것은 매우 불필요한 일이다. 예를 들어 d개의 SKU가 있다면 2d의 조합의 경우가 생긴다. SKU가 30개만 되도 엄청난 조합의 경우로 이를 DBMS에서 읽어드리면 매우 많은 시간이 걸린다. 1세대 알고리즘을 사용하면 아마도 100개만 되어도 시스템은 몇일이 지나도 결과를 보여주지 않고 있을것이다.

그리고 A를 구매하면 B를 구매한다에서 A를 antecedent라고 하고 B를 consequent라고 한다. 때로는 antecedent를 기준으로 보아야 될 경우가 있고 때로는 consequent를 기준으로 결과를 봐야할 경우가 있다. 자세한 내용은 활용편에서 설명하겠다.

---

[10] "Big Data Analysis for CRM using R", 김경태

4.1.2 기존 연관성 분석의 이슈

기존 알고리즘의 문제는 대용량의 데이터에 대한 연관성 분석이 불가능하다는 점이었다. 오래된 알고리즘인 Apriori를 사용하였기 때문에 SKU level의 연관성분석은 실행하면 시스템이 먹통이 되는게 일반적이었다. 저자가 2001년 유통사에서 1년간의 SKU 레벨의 데이터를 적용하여 규칙을 도출하려 했다가 서버를 몇일동안 실행한 적이 있었다. 결국 중단을 시키고 상품구조 level 1으로 해도 30개가 된 경우로 생각되는데 하루가 지나도 결과가 나오지 않았다. 결국 level 1을 10개 정도로 grouping하였더니 나온 경우가 있었는데 의미가 전혀 없었고 활용성도 없었다. 예를 들어 정육제품을 사면 채소를 산다는 규칙이다. 너무나 당연할 수 있고 이걸 활용할 실질성이 낮은 것이다. 최소한 소고기 불고기용을 사면 불고기 소스를 같이 산다는게 나오면 다행이다. 그러나 이러한 문제를 해결하기 위해 새로운 알고리즘들이 계속 나오고 이를 지원하는 KXEN과 같은 소프트웨어의 발전으로 많은 문제가 해결되고 있다. R을 이용한 이러한 검증은 아직 수행해 보지 못한 상태이다.

4.1.3 최근 연관성 분석 동향

KXEN에서는 처음부터 1세대 알고리즘인 Apriori나 2세대인 FP Tree가 아닌 3세대의 FPV를 이용하여 메모리를 효율적으로 사용함으로서 SKU level의 연관성 분석을 성공적으로 적용하였다. 가까운 일본에서의 HMV 사례가 대표적인 예이다. 음반은 매우 빨리 변하는 시장으로 매우 많은 고객과 음반의 종류가 계속 변경된다. 따라서 수백만 개의 음반에 대한 연관성 분석을 하루내에 매주 단위로 실행해서 적용해야 되고, 신제품에 대해서는 일단위로 실행하여야 될 상품들도 있다. 그러나 이러한 것들이 기존 솔루션들은 안되는 것을 최신 알고리즘을 적용하여 KXEN에서는 가볍게 처리를 하였고, 이러한 사례들이 알려지자 국내에서는 많은 유통사들과 온라인 기업들이 관심을 가지게 되었다.

4.1.4 연관성 분석 활용방안

장바구니 분석의 경우는 실시간 상품추천을 통한 cross-sell에 응용될 수 있고, 시차분석은 A를 구매한 사람인데 B를 구매안한 경우 B를 추천하는 cross-sell campaign에 사용될 수 있다. 장바구니 분석은 최근 홈페이지, 콜센터, 스마트폰을 통한 식시간 추천이 가능해 짐에 따라 더욱 활용성이 높아지리라 생각된다. 그러나 기업들이 갖고 있는 데이터가 곧장 연관성 분석에 사용될 수 있지는 않다. 예를 들어 여행산업에서 “파격할인 * 푸켓 * 무제한 골프 9월한정”과 같은 데이터를 입력하여 분석한들 무슨의미가 있을까? 아마도 support 값이 매우 낮은 경우가 많을것이다. 따라서 이러한 정보를 가공해서 연관성 규칙을 사용할 수 있는 데이터로 전환해야 한다. 예를 들어 “단체_태국_푸켓_골프_여름”로 변환해야 되지 않을까? 이러한 경우 “단체_중국_해남_골프_겨울”가 연관된 결과로 나올 수 있을 것이다.

이러한 결과가 나오면 검증을 하고 싶을것이다. 그런 경우 테스트 마케팅을 해보면 된다. 기존 방식에 대한 반응율이 얼마인지 정확한 기준으로 평가하고 연관성 규칙을 적용한 테스트 마케팅을 기획한다. 연관성 규칙을 도출하는것과 타겟팅을 하여 캠페인을 기획하는것은 별개의 일이라는 점에서 조심을 해야 한다. 무조건 고객에게 규칙을 적용하여 추천하면 안된다. 예를 들어 10월 여행상품을 추천하고자 9월에 메일을 보내려면 10월에 구매할 상품 즉 가을이라는 단어가 들어간 consequent가 있는 rule들을 선별해서 적용해야 될것이다. 여행은 봄

에가는 사람은 봄에갈 확률이 높고 가을에 간 사람은 가을에 갈 확률이 1.5배 이상 높다. 가을 상품에 대해 과거 거래정보를 기준으로 겨울상품을 추천한다면 10월에 테스트 마케팅을 해도 결과가 안나올것은 당연한 일이다. 또한 채널에 대한 민감도도 고려해야 된다. 주 연령층이 50대 인경우 email이 효과가 없을것은 너무나 당연하다. SMS의 경우 고객별로 response가 다르므로 과거 이력을 기반으로 타겟팅을 해야 한다. 또한 활동고객을 기준으로 해야 한다. 작년에 거래가 있었거나 금년에 웹페이지에 접근한 기록이 있는 사람이 대상이 되어야지 현재 관심이 없고 interaction이 없는 고객은 targeting을 한다고 해도 성과가 안나온다. 즉 연관성 규칙의 효과와 타겟팅의 효과를 결합해서 테스트를 진행하는것이 필요하다. 그러면 최소 40%이상의 증가효과가 나올것으로 예상된다. 그리고 테스트 마케팅 사이즈를 충분히 크게 해야 된다. 충분히 크지 않으면 비율의 증가가 검증되지 않는다.

처음하는 테스트 마케팅은 좀 떨리고 자신감이 없을 것이다. 그러나 꼼꼼하게 설계하고 이런 일을 몇번 해보면 자심감이 생기고 미리 반응율을 충분히 파악할 수 있다. “링에 많이 올라가는 사람이 이긴다”라는 말이 있다. 실수도 해보고 많이 해봐야 한다.

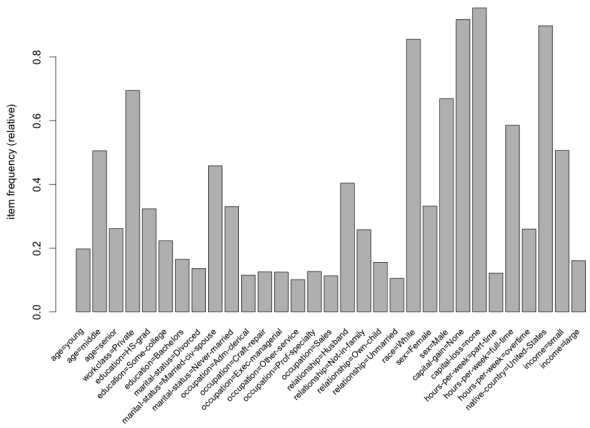
4.2 Association Analysis

```
> library(arules)
> data(Epub)
> Epub
transactions in sparse format with
 15729 transactions (rows) and
 936 items (columns)
> summary(Epub)
transactions as itemMatrix in sparse format with
 15729 rows (elements/itemsets/transactions) and
 936 columns (items) and a density of 0.001758755

most frequent items:
doc_11d doc_813 doc_4c6 doc_955 doc_698 (Other)
    356     329     288     282     245   24393

element (itemset/transaction) length distribution:
sizes
    1     2     3     4     5     6     7     8     9    10    11    12    13
11615  2189   854   409   198   121    93    50    42    34    26    12    10
   14    15    16    17    18    19    20    21    22    23    24    25    26
   10     6     8     6     5     8     2     2     3     2     3     4     5
   27    28    30    34    36    38    41    43    52    58
    1     1     1     2     1     2     1     1     1     1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   1.000   1.646   2.000  58.000

includes extended item information - examples:
   labels
1 doc_11d
2 doc_13d
```

```
3 doc_14c

includes extended transaction information - examples:
     transactionID          TimeStamp
10792  session_4795 2003-01-02 10:59:00
10793  session_4797 2003-01-02 21:46:01
10794  session_479a 2003-01-03 00:50:38


> Year <- strftime(as.POSIXlt(transactionInfo(Epub)[["TimeStamp"]]),"%Y")
> table(Year)
Year
2003 2004 2005 2006 2007 2008 2009
 986 1376 1610 3010 4051 4693    3
> length(Epub)
[1] 15729
> inspect(head(Epub))
  items     transactionID          TimeStamp
1 {doc_154}  session_4795 2003-01-02 10:59:00
2 {doc_3d6}  session_4797 2003-01-02 21:46:01
3 {doc_16f}  session_479a 2003-01-03 00:50:38
4 {doc_11d,
   doc_1a7,
   doc_f4}   session_47b7 2003-01-03 08:55:50
5 {doc_83}   session_47bb 2003-01-03 11:27:44
6 {doc_11d}  session_47c2 2003-01-04 00:18:04
> inspect(Epub[1:3])
  items     transactionID          TimeStamp
1 {doc_154}  session_4795 2003-01-02 10:59:00
2 {doc_3d6}  session_4797 2003-01-02 21:46:01
3 {doc_16f}  session_479a 2003-01-03 00:50:38


> as(Epub[1:5],"list")
$session_4795
[1] "doc_154"

$session_4797
[1] "doc_3d6"

$session_479a
[1] "doc_16f"

$session_47b7
[1] "doc_11d" "doc_1a7" "doc_f4"

$session_47bb
[1] "doc_83"


> transactionInfo(Epub[size(Epub) > 30])
     transactionID          TimeStamp
```

```
11371  session_6308 2003-08-18 07:16:12
13195  session_b391 2005-02-27 00:34:21
15319  session_fe5c 2006-04-16 01:14:28
1785  session_13324 2006-11-14 01:53:17
3867  session_17967 2007-05-22 03:00:10
6869  session_1d082 2008-02-26 03:26:03
8079  session_1f32a 2008-05-18 01:30:06
9236  session_21271 2008-08-18 23:57:26
10405 session_23930 2008-11-23 23:34:13

> data(AdultUCI)
> dim(AdultUCI)
[1] 48842    15
> AdultUCI[1:2,]
  age        workclass fnlwgt education education-num      marital-status
1  39        State-gov  77516 Bachelors            13       Never-married
2  50 Self-emp-not-inc  83311 Bachelors            13 Married-civ-spouse
       occupation  relationship  race  sex capital-gain capital-loss hours-per-week
1    Adm-clerical Not-in-family White Male         2174            0            40
2 Exec-managerial       Husband White Male            0            0            13
  native-country income
1  United-States  small
2  United-States  small


> AdultUCI[["fnlwgt"]] <- NULL
> AdultUCI[["education-num"]]<-NULL
을 하면 column이 없어진다.

> summary(AdultUCI[["age"]])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  17.00   28.00   37.00   38.64   48.00   90.00
> cut(AdultUCI[["age"]],c(15,37,100))
    [1] (37,100] (37,100] (37,100] (37,100] (15,37]  (15,37]  (37,100] (37,100] (15,37]  (37,100]
(15,37]  (15,37]
   [13] (15,37]  (15,37]  (37,100] (15,37]  (15,37]  (15,37]  (37,100] (37,100] (37,100] (37,100]
(15,37]  (37,100]

> table(cut(AdultUCI[["age"]],c(15,37,100)))

 (15,37] (37,100]
   24974    23868

> AdultUCI[[ "age"]] <- ordered(cut(AdultUCI[[ "age"]],
c(15,25,45,65,100)),labels=c("young","middle","senior","old"))
> View(AdultUCI)
> AdultUCI[["hours-per-week"]] <- ordered(cut(AdultUCI[["hours-per-
week"]],c(0,25,40,60,168)),labels=c("part-time","full-time","overtime","workaholic"))
> View(AdultUCI)
```

```
> AdultUCI[[ "capital-gain"]] <- ordered(cut(AdultUCI[[ "capital-gain"]],c(-Inf,
0,median(AdultUCI[[ "capital-gain"]][AdultUCI[[ "capital-gain"]]>0]),Inf)),labels = c("None",
"Low", "High"))
> AdultUCI[[ "capital-loss"]] <- ordered(cut(AdultUCI[[ "capital-loss"]],c(-Inf,0,
median(AdultUCI[[ "capital-loss"]][AdultUCI[[ "capital-loss"]]>0]),Inf)),labels = c("none", "low",
"high"))

> Adult <- as(AdultUCI, "transactions")
> Adult
transactions in sparse format with
 48842 transactions (rows) and
 115 items (columns)
> summary(Adult)
transactions as itemMatrix in sparse format with
 48842 rows (elements/itemsets/transactions) and
 115 columns (items) and a density of 0.1089939

most frequent items:
        capital-loss=none          capital-gain=None native-country=United-States
                    46560                      44807                      43832
             race=White          workclass=Private                    (Other)
                    41762                      33906                     401333

element (itemset/transaction) length distribution:
sizes
    9    10    11    12    13
   19   971  2067 15623 30162

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   9.00   12.00   13.00   12.53   13.00   13.00

includes extended item information - examples:
      labels variables levels
1  age=young       age  young
2 age=middle       age middle
3 age=senior       age senior

includes extended transaction information - examples:
  transactionID
1             1
2             2
3             3
> itemFrequencyPlot(Adult, support = 0.1, cex.names=0.8)
```




```
> rules <- apriori(Adult, parameter=list(support=0.01, confidence=0.6))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.6    0.1    1 none FALSE            TRUE    0.01      1     10  rules FALSE

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

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 ...[115 item(s), 48842 transaction(s)] done [0.04s].
sorting and recoding items ... [67 item(s)] done [0.01s].
creating transaction tree ... done [0.05s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.93s].
writing ... [276443 rule(s)] done [0.05s].
creating S4 object  ... done [0.31s].
> rules
set of 276443 rules

> summary(rules)
set of 276443 rules

rule length distribution (lhs + rhs):sizes
    1     2     3     4     5     6     7     8     9    10
    6   432  4981 22127 52669 75104 67198 38094 13244  2588

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.000   6.000   6.289   7.000  10.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.01001   Min.   :0.6000   Min.   : 0.7171
 1st Qu.:0.01253   1st Qu.:0.7691   1st Qu.: 1.0100
 Median :0.01701   Median :0.9051   Median : 1.0554
 Mean   :0.02679   Mean   :0.8600   Mean   : 1.3109
```

```
 3rd Qu.:0.02741   3rd Qu.:0.9542   3rd Qu.: 1.2980
 Max.   :0.95328   Max.   :1.0000   Max.   :20.6826

mining info:
  data ntransactions support confidence
 Adult        48842    0.01        0.6

> rulesIncomeSmall <- subset(rules, subset = rhs %in% "income=small" & lift > 1.2)
> rulesIncomeLarge <- subset(rules, subset = rhs %in% "income=large" & lift > 1.2)
> length(rulesIncomeSmall)
[1] 10496
> length(rulesIncomeLarge)
[1] 88

> inspect(head(sort(rulesIncomeSmall, by = "confidence"), n = 3))
  lhs                          rhs             support confidence     lift
1 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   hours-per-week=part-time,
   native-country=United-States} => {income=small} 0.01074895  0.7104195 1.403653
2 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   hours-per-week=part-time}     => {income=small} 0.01144507  0.7102922 1.403402
3 {workclass=Private,
   marital-status=Never-married,
   relationship=Own-child,
   sex=Male,
   capital-gain=None,
   hours-per-week=part-time,
   native-country=United-States} => {income=small} 0.01046231  0.7097222 1.402276
> inspect(head(sort(rulesIncomeLarge, by = "confidence"), n = 3))
  lhs                                rhs             support confidence     lift
1 {marital-status=Married-civ-spouse,
   capital-gain=High,
   native-country=United-States}     => {income=large} 0.01562180  0.6849192 4.266398
2 {marital-status=Married-civ-spouse,
   capital-gain=High,
   capital-loss=none,
   native-country=United-States}     => {income=large} 0.01562180  0.6849192 4.266398
3 {relationship=Husband,
   race=White,
   capital-gain=High,
   native-country=United-States}     => {income=large} 0.01302158  0.6846071 4.264454

> WRITE(rulesIncomeSmall, file = "rulesIncomeSmall.txt", sep = "\t", col.names = NA)
```

csv로 “,” separator를 사용하는 경우 format이 문제가 생기는 데이터 형태이므로 tab separator를 이용한다.

```
> library("pmml")
> rules_pmml <- pmml(rulesIncomeSmall)
> saveXML(rules_pmml, file = "data.xml")
```

4.3 Sequence Pattern

```
> library(arulesSequences)
> data(zaki)
> as(zaki,"data.frame")
   transactionID.sequenceID transactionID.eventID transactionID.SIZE     items
1                         1                    10                  2     {C,D}
2                         1                    15                  3   {A,B,C}
3                         1                    20                  3   {A,B,F}
4                         1                    25                  4 {A,C,D,F}
5                         2                    15                  3   {A,B,F}
6                         2                    20                  1       {E}
7                         3                    10                  3   {A,B,F}
8                         4                    10                  3   {D,G,H}
9                         4                    20                  2     {B,F}
10                        4                    25                  3   {A,G,H}
> s1 <- cspade(zaki, parameter = list(support=0.4),control=list(verbose=TRUE))

parameter specification:
support : 0.4
maxsize :  10
maxlen  :  10

algorithmic control:
bfstype : FALSE
verbose :  TRUE
summary : FALSE

preprocessing ... 1 partition(s), 0 MB [0.027s]
mining transactions ... 0 MB [0.009s]
reading sequences ... [0.043s]

total elapsed time: 0.079s
> summary(s1)
set of 18 sequences with

most frequent items:
     A      B      F      D (Other)
    11     10     10      8     28

most frequent elements:
   {A}    {D}    {B}    {F}  {B,F} (Other)
     8      8      4      4      4      3

element (sequence) size distribution:
```

```
sizes
1 2 3
8 7 3

sequence length distribution:
lengths
1 2 3 4
4 8 5 1

summary of quality measures:
    support
 Min.   :0.5000
 1st Qu.:0.5000
 Median :0.5000
 Mean   :0.6528
 3rd Qu.:0.7500
 Max.   :1.0000

mining info:
 data ntransactions nsequences support
 zaki          10         4     0.4
> as(s1,"data.frame")
         sequence support
1           <{A}>    1.00
2           <{B}>    1.00
3           <{D}>    0.50
4           <{F}>    1.00
5         <{A,F}>    0.75
6         <{B,F}>    1.00
7       <{D},{F}>    0.50
8     <{D},{B,F}>    0.50
9       <{A,B,F}>    0.75
10        <{A,B}>    0.75
11      <{D},{B}>    0.50
12      <{B},{A}>    0.50
13      <{D},{A}>    0.50
14      <{F},{A}>    0.50
15  <{D},{F},{A}>    0.50
16    <{B,F},{A}>    0.50
17 <{D},{B,F},{A}>   0.50
18  <{D},{B},{A}>    0.50
```

4.4 Visualization

Association Analysis를 수행한 경우 매우 많은 rule들이 나올 수 있다. 이러한 경우 비즈니스에 대한 이해가 있다해도, 모두 확인해 가면서 흥미로운 규칙을 발견하는 작업은 매우 고된 작업이다. 이러한 일을 좀더 쉽게 접근할 수 있도록 시각화를 편리하게 해주는 package가 aruleViz이다.

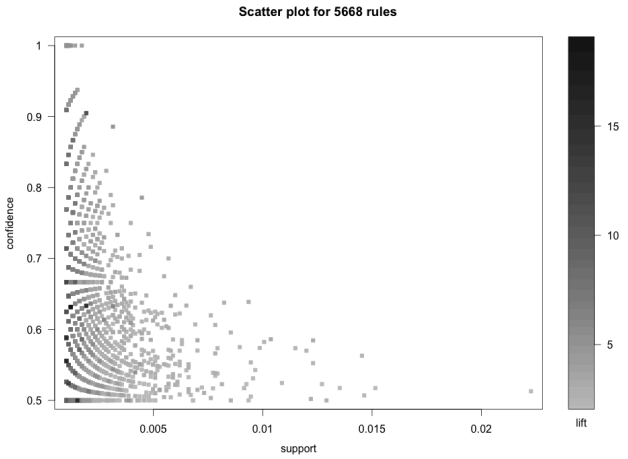
물론 일정 support와 confidence에서 나온 lift가 일정값 이상인 rule을 모두 적용하는것도 하나의 방법이다. 이에 대한 이견이 있을 수도 있으나, 예를 들어 음반관련 온라인 쇼핑몰에서 주단위로 수만개의 SKU에 대해 1,000만 고객들의 거래정보에 따라 association rule을 update해서 적용하는데 수작업이 개입되기는 어렵다.

그러나 분석적 측면에서 보다 의미 있는, 새로운 패턴을 발견한다는 측면에서 visualization은 의미가 있다고 생각된다.

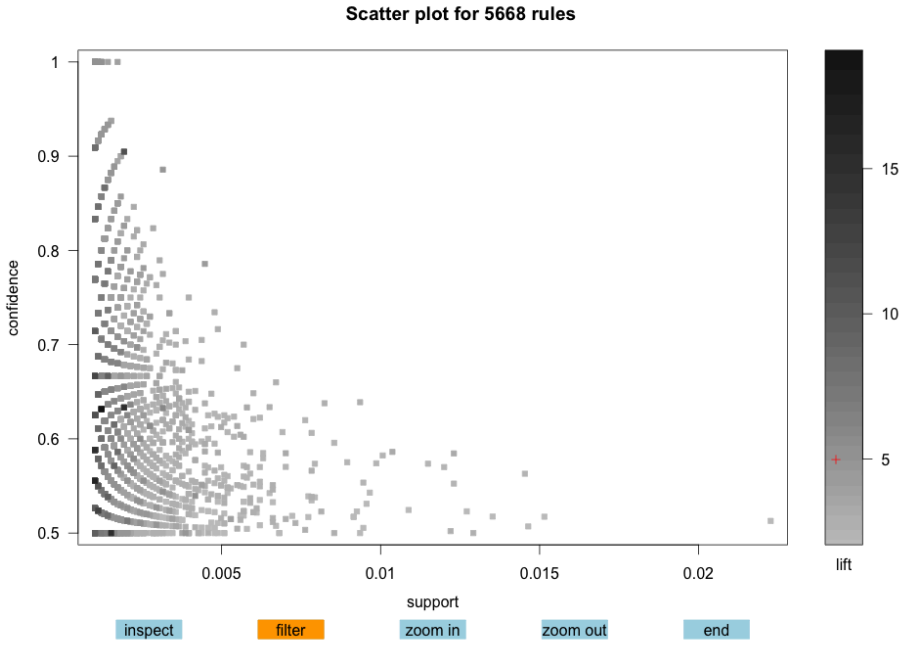
```
> library('arulesViz')
> data('Groceries')
> inspect(head(Groceries,n=5))
  items
1 {citrus fruit,
   semi-finished bread,
   margarine,
   ready soups}
2 {tropical fruit,
   yogurt,
   coffee}
3 {whole milk}
4 {pip fruit,
   yogurt,
   cream cheese ,
   meat spreads}
5 {other vegetables,
   whole milk,
   condensed milk,
   long life bakery product}
> 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
            2513             1903             1809             1715
          yogurt          (Other)
            1372            34055

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

```
> rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target
        0.5    0.1    1 none FALSE            TRUE   0.001      1     10  rules
   ext
 FALSE

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

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 ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.02s].
writing ... [5668 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].

> rules
set of 5668 rules
> inspect(head(sort(rules, by='lift'),3))
  lhs                     rhs                 support confidence     lift
1 {Instant food products,
   soda}                => {hamburger meat} 0.001220132  0.6315789 18.99565
2 {soda,
   popcorn}               => {salty snack}    0.001220132  0.6315789 16.69779
3 {flour,
   baking powder}         => {sugar}          0.001016777  0.5555556 16.40807

> plot(rules)
```



> plot(rules, interactive=TRUE)

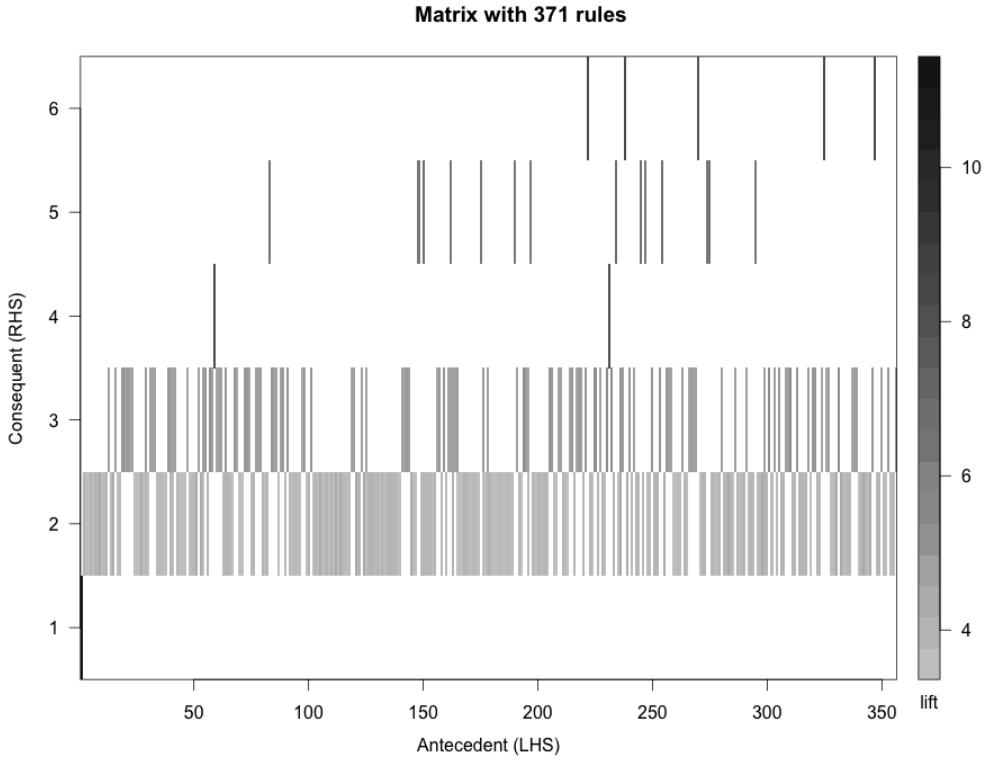


> sel <- plot(rules, measure=c("support", "lift"), shading="confidence", interactive=TRUE)
Interactive mode.
Select a region with two clicks!
Select minimum confidence in colorkey.
Nothing selected!

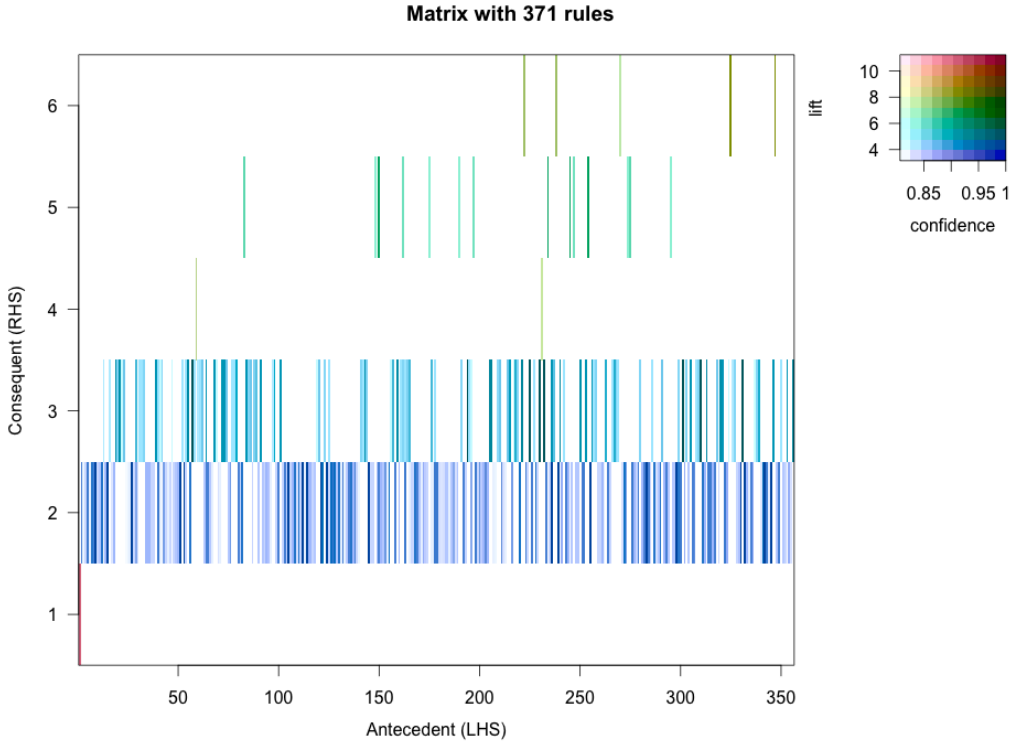
Number of rules selected: 21
lhs rhs support confidence lift
1 {domestic eggs,
sugar} => {whole milk} 0.003558719 0.7142857 2.795464
2 {tropical fruit,
root vegetables,
other vegetables,
yogurt} => {whole milk} 0.003558719 0.7142857 2.795464
3 {pork,
butter} => {whole milk} 0.003863752 0.7037037 2.754049

> subrules <- rules[quality(rules)$confidence > 0.8]
> subrules
set of 371 rules
> plot(subrules, method="matrix", measure="lift")
Itemsets in Antecedent (LHS)
[1] "{liquor,red/blush wine}"
[2] "{curd,cereals}"
[3] "{yogurt,cereals}"
[4] "{butter,jam}"
....
[355] "{citrus fruit,tropical fruit,root vegetables,other vegetables,yogurt}"
[356] "{citrus fruit,tropical fruit,root vegetables,whole milk,yogurt}"
Itemsets in Consequent (RHS)

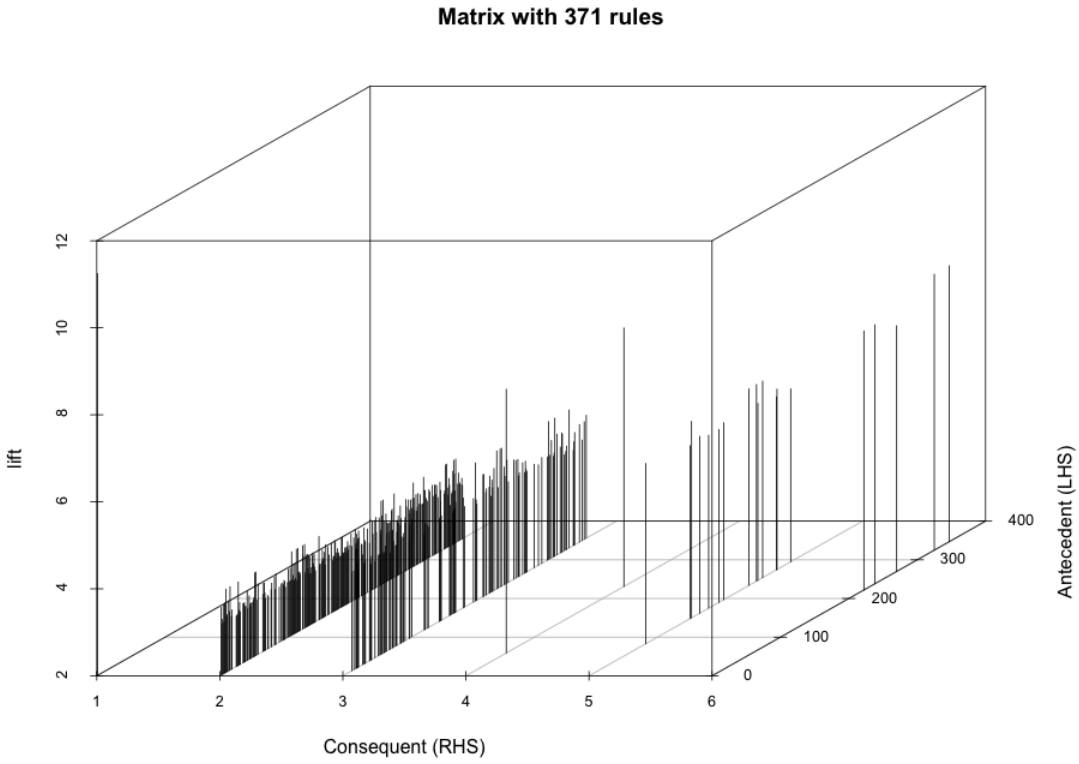
[1] "{bottled beer}" "{whole milk}" "{other vegetables}"
[4] "{tropical fruit}" "{yogurt}" "{root vegetables}"



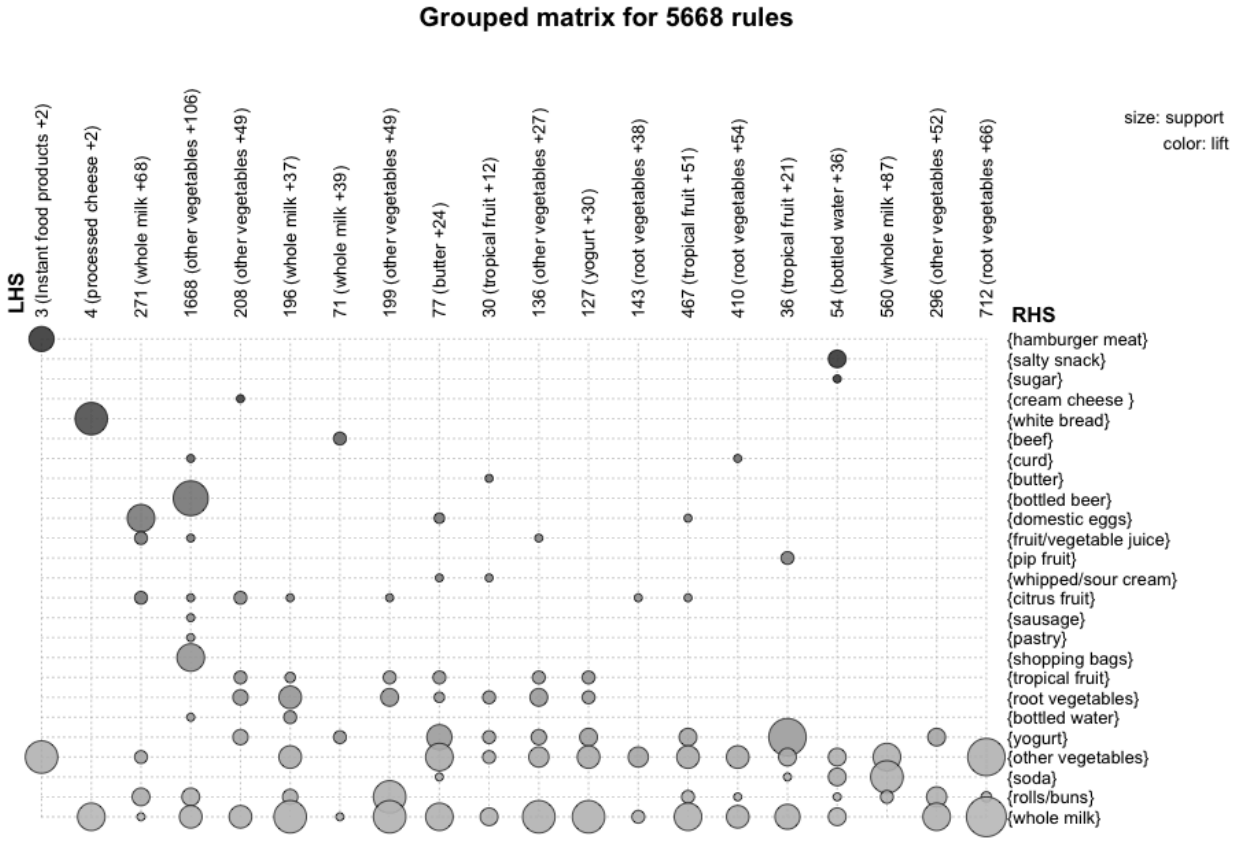
> plot(subrules, method="matrix", measure=c("lift", "confidence"))



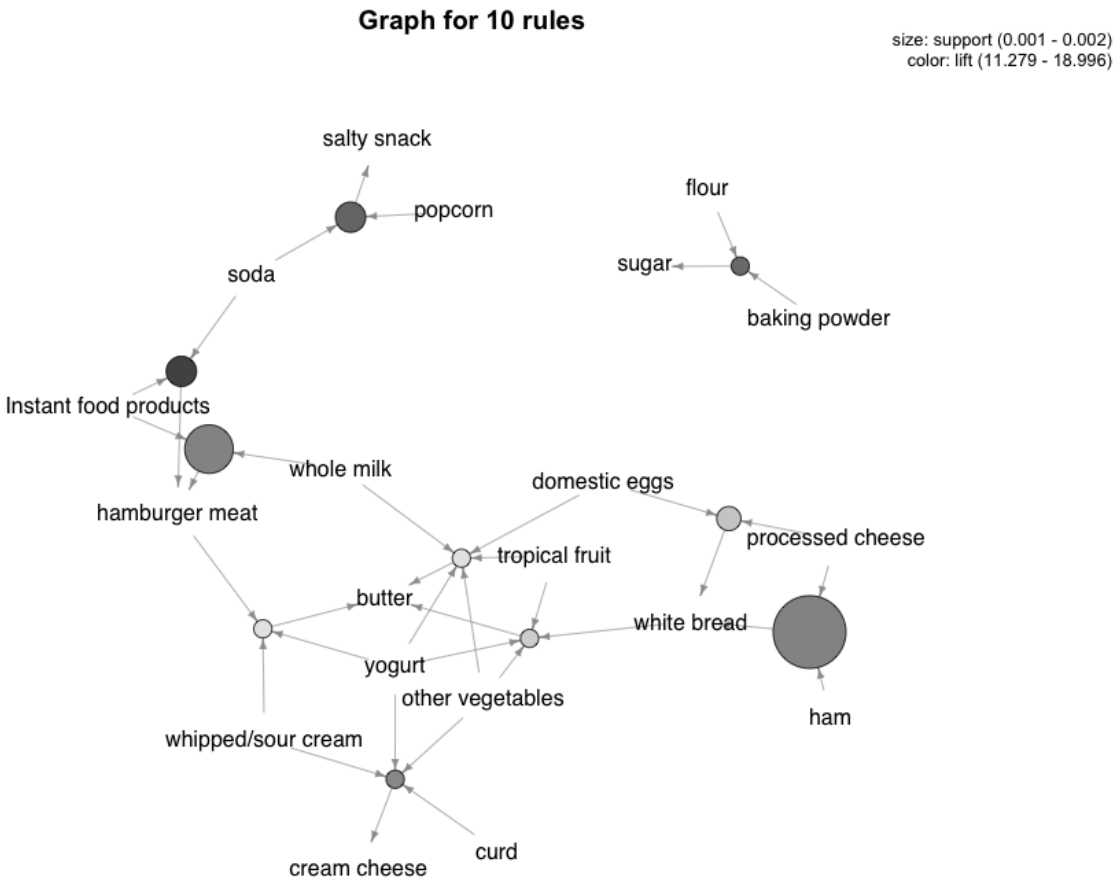
> plot(subrules, method="matrix3D", measure="lift")



> plot(rules, method="grouped")



```
> subrules2 <- head(sort(rules, by="lift"),10)
> plot(subrules2, method="graph", control=list(type="items"))
```



Chapter 5

Time Series Analysis

5.1 Time Series Analysis 개요[11]

5.1.1 Forecasting이란

매출예측에 거부감을 갖는 사람들이 많은 편이고 정확도에 대해 이슈를 제기하곤 한다. 또는 우리는 매출예측 안해도 잘하고 있다는 곳도 있고, 엑셀로도 매출예측 잘해서 쓰고 있는데 무슨 Data Mining에서 매출예측하냐고 하는곳도 있다. 매출예측이 급상승한 가격을 예측할 수 있는지, 갑작스런 외환위기를 예측해 줄 수 있는지 등등의 수많은 논재들이 나오곤 한다. 예측은 과거의 값을 이용해서 또는 미래에 예상되는 기타 정보를 활용해서 미래를 예측하는 기법이다. 정확도가 수작업이나 엑셀로 한것 보다는 높지만 100% 정확할 수는 없다. 저자의 입장은 통계학 박사의 입장이 아니라 현업으로서의 입장에서도 5~10% 오차가 나도 정확도는 높은 편이고 의미있고 활용가능성이 있다고 생각한다. 최근 한전에서 전력수요예측에서 오차가 5%발생하여 전국이 정전될뻔한 사태가 있었다. 전력수요예측에서 전력의 사용량이 일별로 편차가 그렇게 큰지는 모르겠으나 오차도 문제였지만 오차에 대응할 수 없었던 기업도 문제다. 한 예로 통신사의 기업 매출을 예측하는데 오차가 얼마나 클까? 매월 매출이 마구 변할까? 아마 95% 이상 예측을 할 수 있을것이다.

5.1.2 예측에 대한 의견

예측은 마케팅이건 세일즈건 중요한 일이고 예측경영이야 말로 의사결정에 필수적인 내용이다. 예측을 하지 않고도 수작업으로 충분히 더잘한다고 하는 사람들이 있으면 저자가 반드시 시켜보는 일이 있다. 2년 이전 3년간의 자료를 주고 1년전 매출을 월별로 적어보라고 한다. 그리고 데이터 마이닝을 이용해서 예측한 값을 그자리에서 검증해준다. 결과는 수작업이나 엑셀로 만든값의 오차는 15% 수준이다. 마이닝은 2~10% 오차가 나온다. 그걸도 해당 비즈니스에 대한 기본적인 이해도 없는 사람이 수십 초만에 만들어 낼 수 있는 결과이다. 이래도 예측기법을 사용안하겠다고 하면 그냥 하지 말라고 하면 된다. 담당자나 팀장은 의미없다고 하겠지만 경영자나 오너는 가만있지 않을것이다. 그들에겐 너무나 중요한 정보이기 때문이다. 단 1%만 정확도가 개선되어도 수익에 커다란 영향을 미치기 때문이다. 따라서, 이런 중요한 사항에 대한 내용은 현업 담당자나 팀 레벨이 아니라 C-Level이나 오너와 이야기를 해야 한다.

5.1.3 예측을 위한 데이터 준비

---

[11] “Big Data Analysis for CRM using R”, 김경태

예측을 위한 데이터는 우선 전체 매출, 상품군별 매출, 매장별 매출을 월단위로 3년간 보유하고 있으면 된다. 이런 경우 약 1년간의 월간 매출을 예측하는데는 무리가 없다. 3년 이상의 데이터가 가능한 경우 오래된 데이터의 패턴과 최근 패턴이 매우 크게 차이가 있다면 최근 3년 데이터를 이용하는것이 유리하다. 구

5.1.4 TCSI 분석

예측값을 Trend, Cycle, Seasonality, Irregularity로 구분하여 분석하는것으로 Trend는 매우 중요하고 Seasonality로 인한 변동을 고려해서 예측과 평가를 제대로 해야 한다. 흔히 전월대비 증가하면 전년대비 증가하면 잘했다고 생각하는데 Trend와 Seasonality를 고려하면 도리어 못한 결과일 수 있다.

5.1.5 평가지표

모델 결과에 대한 평가지표로는 MAE, MAPE 등이 있다. MAPE (Mean Absolute Percentage Error)가 광범위하게 쓰이고 있다.

5.1.6 예측 활용방안

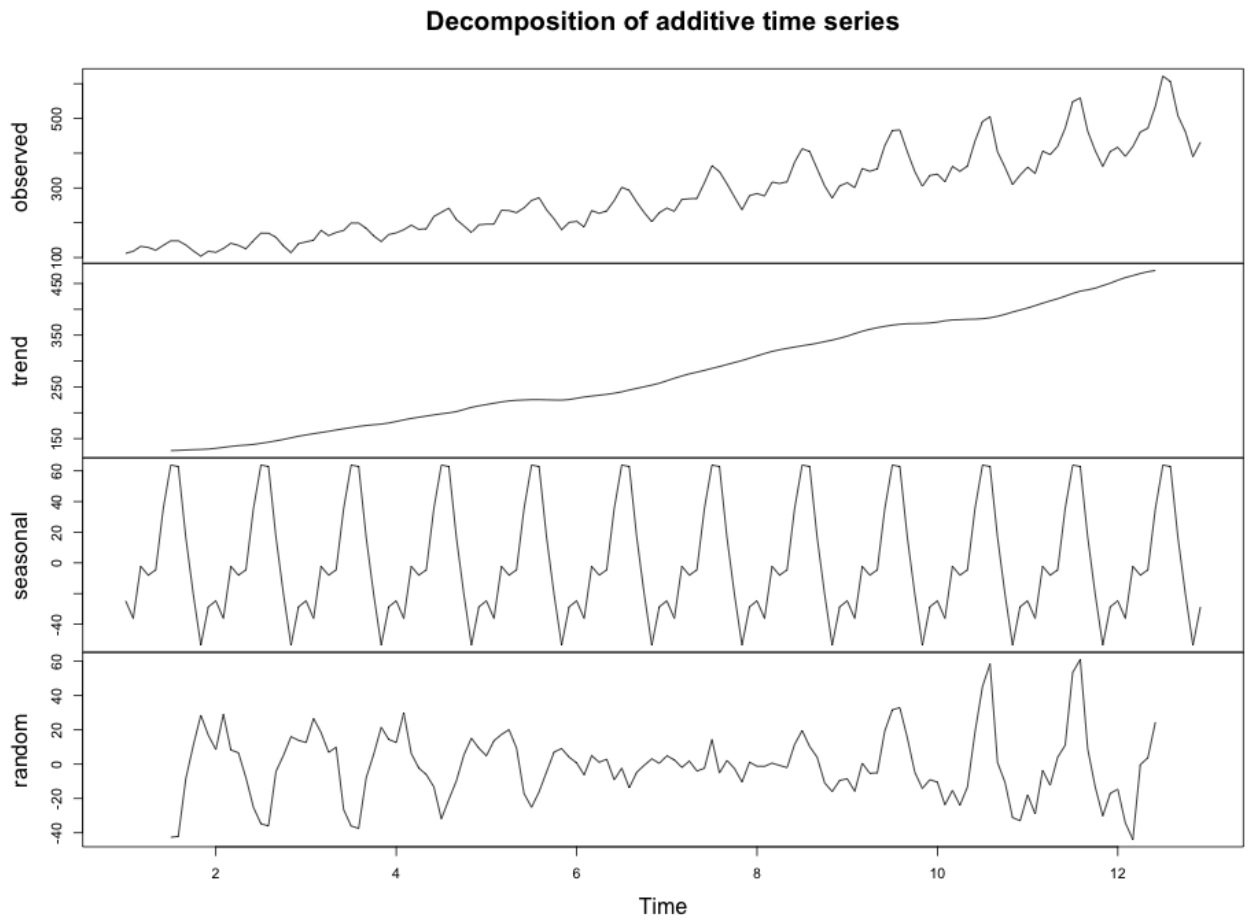
가장 중요한 예측을 어떻게 활용할것인가다. 예측은 예측하는데 자체에는 의미가 없다. 아무리 정확도가 높아도 뭔가에 써야 한다. 그 방안으로 예측경영을 하는것이다. 매출이 감소할 징후가 보이면 대응을 마련하고 예측보다 매출이 적게 나오면 왜 그런지 분석을 하고 이로 인해 향후 매출이 계속 감소할것 같으면 마케팅을 강화해야 한다. 예상보다 매출이 잘 나오면 왜 잘나오는지, 다른 곳의 매출이 줄어들면서 매출이 증가한것은 아닌지 등등을 분석해야 한다. 예를 들어 매출 추이가 이미 하향으로 들어간 상품은 홈쇼핑이나 온라인에서는 곧바로 대체할 상품을 찾아야 한다. 매출이 급상승하는 패턴을 보인 상품에는 마케팅을 집중하여 최대한 초기에 수익을 극대화 하고 또 다른 상품으로 넘어가야 한다. 이것이 예측을 활용하고 수익을 극대화 하는 방안이다.

또 다른 예로는 제조업에서는 예측을 통해 원재료 가격을 예상하여 재고를 얼만큼 가져갈것인지, 원재료를 얼마에 얼만큼 사다두는게 좋을지 예측을 해서 활용할 수 있다. 패션업계에서는 판매될 물건 만큼만 생산한다든지, 수입을 해서 할인이나 소각을 1%만이라도 줄인다면 수익은 엄청난 영향을 받을 것이다.

5.2 Exercise

```
> data(AirPassengers)
> apts <- ts(AirPassengers,frequency=12)
> f <- decompose(apts)
> f$figure
 [1] -24.748737 -36.188131  -2.241162  -8.036616  -4.506313  35.402778  63.830808
```

```
 [8]  62.823232  16.520202 -20.642677 -53.593434 -28.619949
> plot(f)
```



Fit an ARIMA (autoregressive integrated moving average) model to a univariate time series, and use it for forecasting.

```
arima(x, order = c(0, 0, 0),
      seasonal = list(order = c(0, 0, 0), period = NA),
      xreg = NULL, include.mean = TRUE,
      transform.pars = TRUE,
      fixed = NULL, init = NULL,
      method = c("CSS-ML", "ML", "CSS"),
      n.cond, optim.method = "BFGS",
      optim.control = list(), kappa = 1e6)
```

Arguments

x
a univariate time series

order
A specification of the non-seasonal part of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order.

seasonal
A specification of the seasonal part of the ARIMA model, plus the period (which defaults to frequency(x)). This should be a list with componentsorder and period, but a specification of just a numeric vector of length 3 will be turned into a suitable list with the specification as the order.

xreg
Optionally, a vector or matrix of external regressors, which must have the same number of rows as x.

include.mean
Should the ARMA model include a mean/intercept term? The default is TRUE for undifferenced series, and it is ignored for ARIMA models with differencing.

transform.pars
Logical. If true, the AR parameters are transformed to ensure that they remain in the region of stationarity. Not used for method = "CSS".

fixed
optional numeric vector of the same length as the total number of parameters. If supplied, only NA entries in fixed will be varied. transform.pars = TRUE will be overridden (with a warning) if any AR parameters are fixed. It may be wise to set transform.pars = FALSE when fixing MA parameters, especially near non-invertibility.

init
optional numeric vector of initial parameter values. Missing values will be filled in, by zeroes except for regression coefficients. Values already specified in fixed will be ignored.

method
Fitting method: maximum likelihood or minimize conditional sum-of-squares. The default (unless there are missing values) is to use conditional-sum-of-squares to find starting values, then maximum likelihood.

n.cond
Only used if fitting by conditional-sum-of-squares: the number of initial observations to ignore. It will be ignored if less than the maximum lag of an AR term.

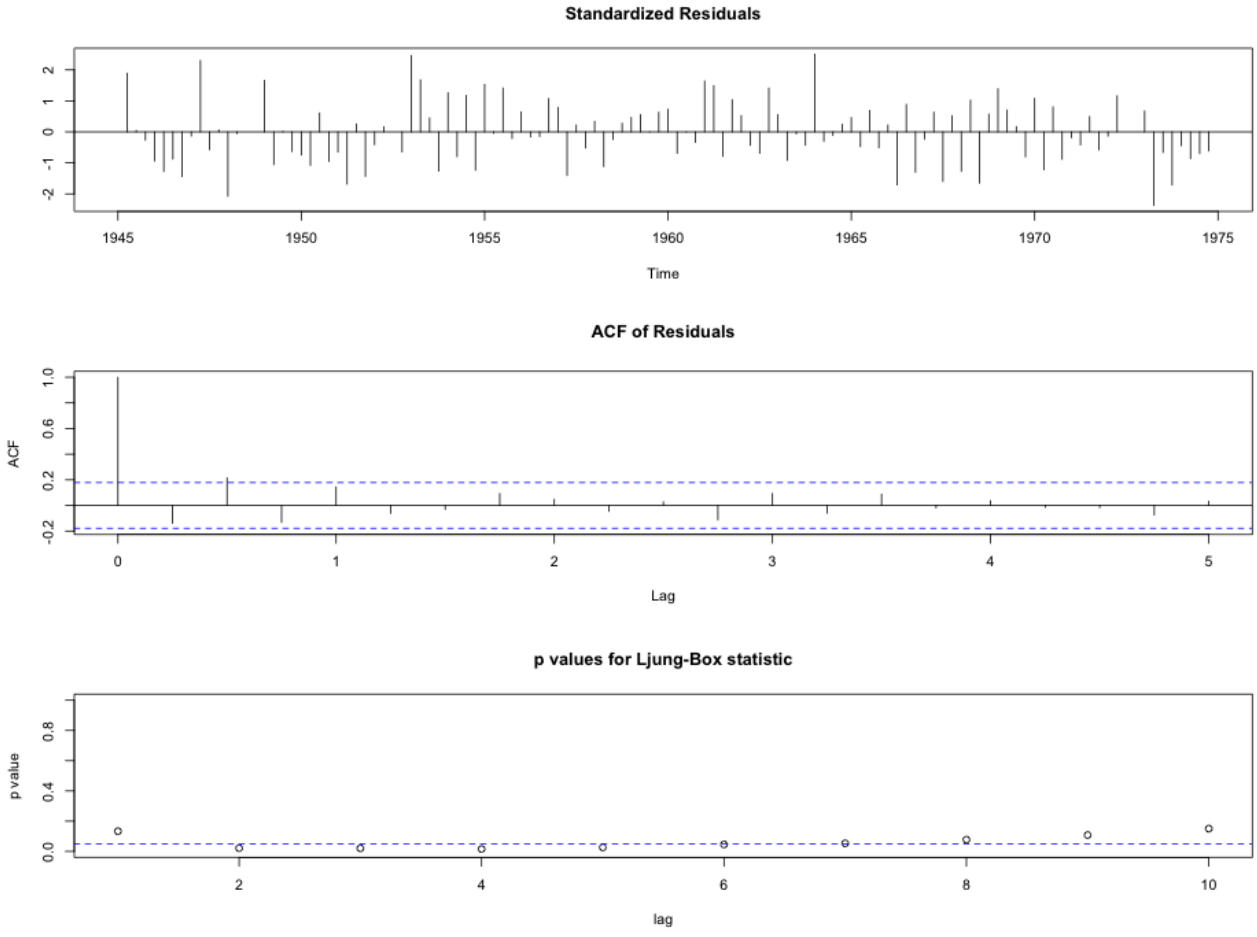
optim.method
The value passed as the method argument to optim.

optim.control
List of control parameters for optim.

kappa
the prior variance (as a multiple of the innovations variance) for the past observations in a differenced model. Do not reduce this.

```
> fit <- arima(AirPassengers, order=c(1,0,0), list(order=c(2,1,0), period=12))
> fore <- predict(fit, n.ahead=24)
> # error bounds at 95% confidence level
> U <- fore$pred + 2*fore$se
> L <- fore$pred - 2*fore$se
> ts.plot(AirPassengers, fore$pred, U, L, col=c(1,2,4,4), lty = c(1,1,2,2))
> legend("topleft", c("Actual", "Forecast", "Error Bounds (95% Confidence)"),
```

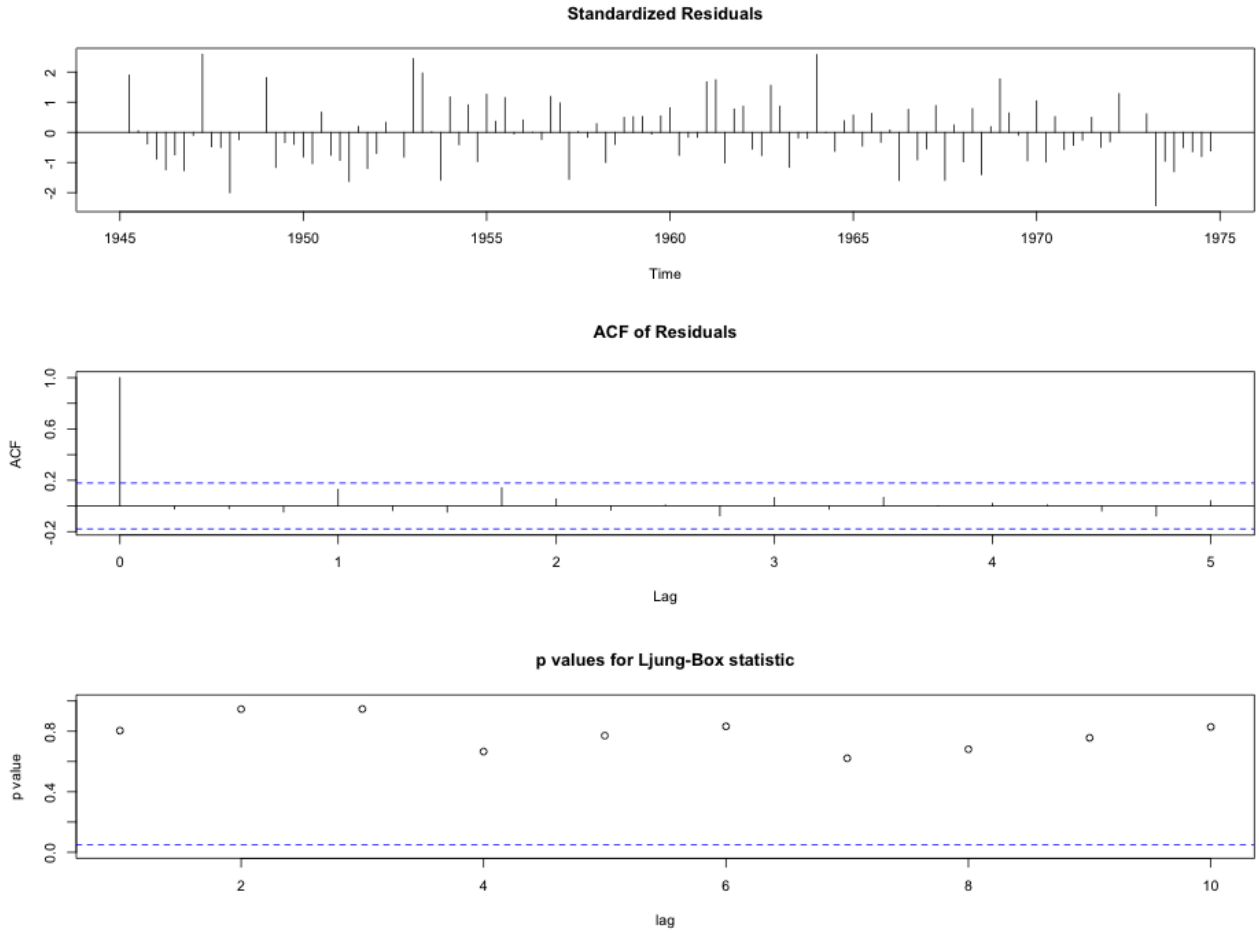
```
+       col=c(1,2,4), lty=c(1,1,2))


> require(graphics)
> (fit1 <- arima(presidents, c(1, 0, 0)))

Call:
arima(x = presidents, order = c(1, 0, 0))

Coefficients:
         ar1  intercept
      0.8242    56.1505
s.e.  0.0555     4.6434


> tsdiag(fit1)
```




```
> (fit3 <- arima(presidents, c(3, 0, 0)))  # smaller AIC

Call:
arima(x = presidents, order = c(3, 0, 0))

Coefficients:
         ar1     ar2      ar3  intercept
```

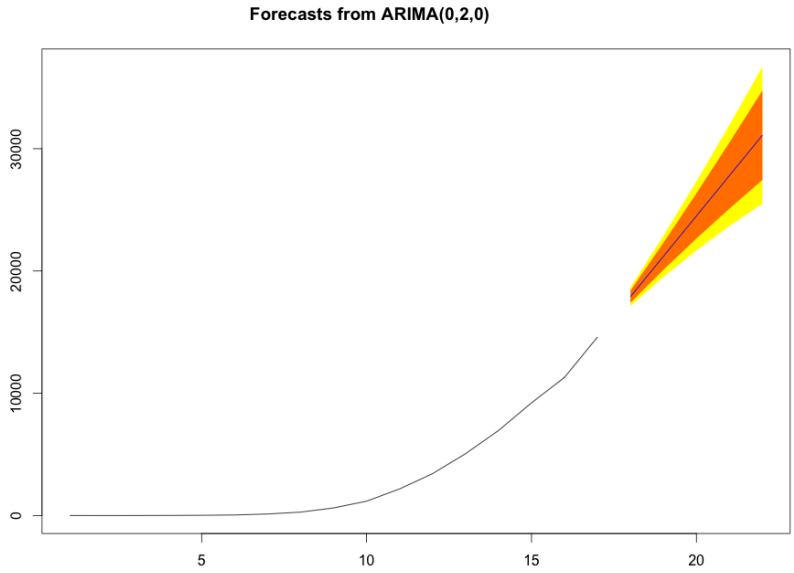
```
      0.7496  0.2523  -0.1890    56.2223
s.e.  0.0936  0.1140   0.0946     4.2845

sigma^2 estimated as 81.12:  log likelihood = -414.08,  aic = 838.16
```

tsdiag(fit3)

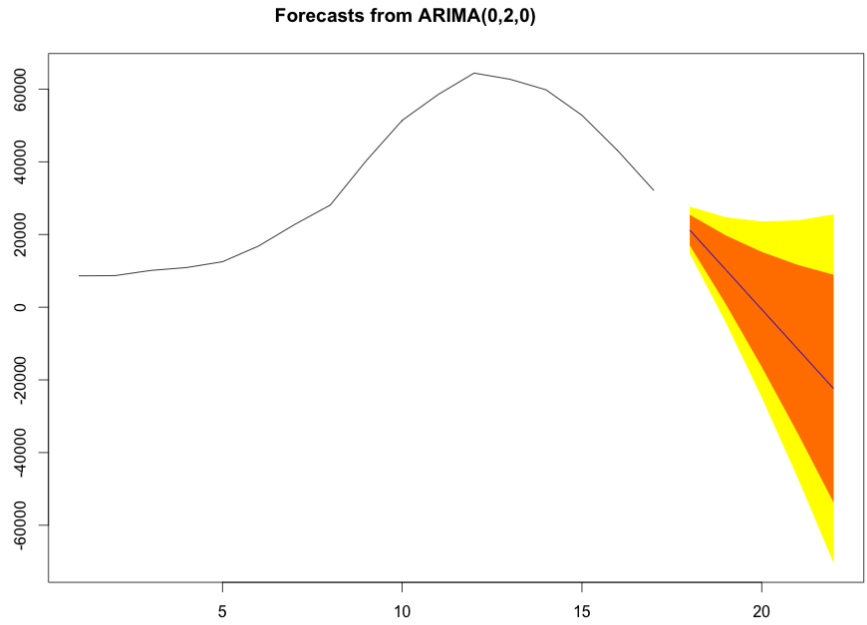



```
> install.packages("forecast")
> library(forecast)
> mining <- read.table("Google Scholar.txt", header=TRUE)

> r <- mining$R
> sas <- mining$SAS

> r.fit <- auto.arima(r)
> r.forecast <- forecast(r.fit,h=5)
> r.forecast
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
18          17900 17407.10 18392.90 17146.18 18653.82
19          21200 20097.85 22302.15 19514.40 22885.60
20          24500 22655.74 26344.26 21679.46 27320.54
```

```
21         27800 25100.29 30499.71 23671.15 31928.85
22         31100 27444.57 34755.43 25509.50 36690.50
> plot(r.forecast)
```




```
> sas.fit <-auto.arima(sas)
> sas.forecast <- forecast(sas.fit, h=5)
> sas.forecast
   Point Forecast      Lo 80    Hi 80      Lo 95    Hi 95
18          21200  16975.5257 25424.47  14739.224 27660.78
19          10300    853.7884 19746.21  -4146.734 24746.73
20           -600 -16406.5353 15206.54 -24774.010 23574.01
21         -11500 -34638.3984 11638.40 -46887.127 23887.13
22         -22400 -53729.5396  8929.54 -70314.396 25514.40
> plot(sas.forecast)
```



Chapter 6

Text Mining

6.1 Text Mining

"최근 많이 각광 받고 있는 분석기법이다. Big Data Analysis에서 가장 많은 기여를 하고 있지 않을까 한다."

Text Mining에 대한 많은 정의가 있겠지만 저자의 해석은, 다양한 문서형태로 부터 데이터를 획득하여 이를 문서 별 단어의 matrix로 만들어 추가적인 분석이나 데이터 마이닝 기법을 적용하여 Insight를 얻거나 의사결정을 지원하는 방법이라고 생각한다.

다양한 문서의 형태는 web에 있는 구조화된 내용이나 pdf 파일, ms office file, open office file, xml, text file 등 다양한 source로 부터 text를 추출해서 이를 하나의 record로 만들고 이에 따른 단어의 구성에 따라 mart를 구성하여 이들간의 관계를 이용하여 sentiment analysis나 word cloud를 수행할 수 있고, 이러한 정보를 clustering이나 classification, social network analysis에 활용할 수 있는 방안이다.

대표적인 예로 twitter에서 자사 브랜드를 언급한 내용이나 자사 twitter계정에서 고객들과 주고 받은 문장을 api를 통해 읽어들여, 사용자/시간 별로 어떠한 keyword로 구성된 내용을 언급하고 있는지, 부정적인 말을 하고 있는지, 긍정적인 말을 하고 있는지, keyword의 변화는 시간의 흐름에 따라 어떻게 되는지, 캠페인을 진행한 전후에 있어서 고객들의 keyword가 변화가 있었는지, 어떤 집단의 고객들이 주로 반응을 보이고 있는지를 통해 insight를 얻고 평판에 대한 관리나 마케팅 활동을 realtime으로 관리할 수 있게된다. 또한 경쟁사 브랜드에 대한 모니터링을 통해 경쟁전략을 수립할 수 있게된다. 이러한 활용이 바로 text mining의 예이다.

좀더 다양한 예를 들면 정치, 환경, 의료 등 다양한 산업과 제조, 설비, 영업 등 다양한 기업의 기능에 관련된 내용을 지원할 수 있다. 공장에서 작업자들이나 설비 유지보수원들이 작성한 정보를 통해 설비고장을 예측할 수 있는 정보를 획득할 수 있기도 하다.

그러나 text mining의 어려운 점은 해당 언어에 대한 깊이 있는 이해와 문화나 습관에 대한 이해가 필요하다. 예를 들어 저자가 영어로 분석을 한다 할지라도 영어에 대한 분석을 많이 한 경우에 한국어에 대한 지식이 아무리 많아도 영어에서의 성과가 더 좋을 수 있다. 실제 영국과 미국, 한국에 대한 분석을 해본 결과 미국에 대한 분석을 더 많이 한 경우 미국에서의 성과는 좋았지만 같은 영어를 쓰는 영국에서는 성과가 좋지 못해 보다 더 영국에 맞는 접근을 시도했고, 이러한 경험들이 한국어에서는 통하지 않았다. 이러한 부분이 text mining의 어려운 점이다.

6.2 Corpus

텍스트 마이닝 package인 tm에서 문서를 관리하는 기본 구조는 Corpus로 불리우고 텍스트 문서들의 집합을 말한다. Corpus는 VCorpus (short for Volatile Corpus)로 메모리에서만 유지되는 Corpus와 PCorpus (Permanent Corpus)로 R외부의 DB나 파일로 관리되는 것이 있으며, 이들은

Corpus는 저장장소를 표시하는 DirSource, VectorSource, or DataframeSource 처럼 directory, 각 vector값 또는 data frame (like CSV files)을 통해 읽어들어져서 생성이 된다.

```
> library(tm)
> txt <- system.file("texts","txt",package="tm")
```

읽어들일 문서의 pass정보를 시스템 폴더의 tm/text/txt 디렉토리에서 가져온다는 것을 정의한다.

```
> txt
[1] "/Library/Frameworks/R.framework/Versions/2.14/Resources/library/tm/texts/txt"
> ovid <- Corpus(DirSource(txt),readerControl=list(langueage="lat"))
```

txt를 통해 읽어들일 디렉토리를 지정하고 latin어로 된 문서임을 지정하여 읽어들인다.

```
> ovid
A corpus with 5 text documents
```

5개의 문서가 읽어졌음을 확인하고 개별 문서에 “[[]]”를 통해 인덱스를 입력하여 데이터를 조회한다.

```
> ovid[[1]]
   Si quis in hoc artem populo non novit amandi,
       hoc legat et lecto carmine doctus amet.
   arte citae veloque rates remoque moventur,
       arte leves currus: arte regendus amor.

   curribus Automedon lentisque erat aptus habenis,
       Tiphys in Haemonia puppe magister erat:
   me Venus artificem tenero praefecit Amori;
       Tiphys et Automedon dicar Amoris ego.
   ille quidem ferus est et qui mihi saepe repugnet:

       sed puer est, aetas mollis et apta regi.
   Phillyrides puerum cithara perfecit Achillem,
       atque animos placida contudit arte feros.
   qui totiens socios, totiens exterruit hostes,
       creditur annosum pertimuisse senem.
> ovid[[2]]
   quas Hector sensurus erat, poscente magistro
       verberibus iussas praebuit ille manus.
   Aeacidae Chiron, ego sum praeceptor Amoris:
       saevus uterque puer, natus uterque dea.
   sed tamen et tauri cervix oneratur aratro,

       frenaque magnanimi dente teruntur equi;
   et mihi cedet Amor, quamvis mea vulneret arcu
       pectora, iactatas excutiatque faces.
   quo me fixit Amor, quo me violentius ussit,
       hoc melior facti vulneris ultor ero:

   non ego, Phoebe, datas a te mihi mentiar artes,
       nec nos aëriae voce monemur avis,
   nec mihi sunt visae Clio Cliusque sorores
       servanti pecudes vallibus, Ascra, tuis:
   usus opus movet hoc: vati parete perito;
```

위에서와 같이 문서가 라틴어로 작성되었음을 추론할 수 있다.

```
> library(tm)
> getReaders()
 [1] "readDOC"                "readGmane"              "readPDF"
 [4] "readReut21578XML"       "readReut21578XMLasPlain" "readPlain"
 [7] "readRCV1"               "readRCV1asPlain"        "readTabular"
[10] "readXML"
```

읽어들일 reader의 종류를 확인해 보면 word, pdf, csv 등의 다양한 문서형식을 읽어들일 수 있음을 확인한다.

```
> reut21578 <- system.file("texts","crude",package="tm")
```

시스템 폴더의 tm/texts/crude 디렉토리에서 읽어들이도록 지정한다.

```
> reuters <- Corpus(DirSource(reut21578),readerControl=list(reader=readReut21578XML))
```

XML 리더를 통해 읽어들인다.

```
> reuters
A corpus with 20 text documents
> reuters[[1]]
$doc
$file
[1] "<buffer>"

$version
[1] "1.0"

$children
$children$REUTERS
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="5670" NEWID="127">
 <DATE>26-FEB-1987 17:00:56.04</DATE>
 <TOPICS>
  <D>crude</D>
 </TOPICS>
 <PLACES>
  <D>usa</D>
 </PLACES>
 <PEOPLE/>
 <ORGS/>
 <EXCHANGES/>
 <COMPANIES/>
 <UNKNOWN>Y
   f0119 reute
u f BC-DIAMOND-SHAMROCK-(DIA   02-26 0097</UNKNOWN>
 <TEXT>
  <TITLE>DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES</TITLE>
  <DATELINE>NEW YORK, FEB 26 -</DATELINE>
  <BODY>Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.
    The reduction brings its posted price for West Texas
Intermediate to 16.00 dlrs a barrel, the copany said.
    &amp;quot;The price reduction today was made in the light of falling
oil product prices and a weak crude oil market,&amp;quot; a company
```

```
spokeswoman said.
    Diamond is the latest in a line of U.S. oil companies that
have cut its contract, or posted, prices over the last two days
citing weak oil markets.
 Reuter</BODY>
 </TEXT>
</REUTERS>


attr(,"class")
[1] "XMLDocumentContent"

$dtd
$external
NULL

$internal
NULL

attr(,"class")
[1] "DTDList"

attr(,"Author")
character(0)
attr(,"DateTimeStamp")
[1] "1987-02-26 17:00:56 GMT"
attr(,"Description")
[1] ""
attr(,"Heading")
[1] "DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES"
attr(,"ID")
[1] "127"
attr(,"Language")
[1] "en"
attr(,"LocalMetaData")
attr(,"LocalMetaData")$TOPICS
[1] "YES"

attr(,"LocalMetaData")$LEWISSPLIT
[1] "TRAIN"

attr(,"LocalMetaData")$CGISPLIT
[1] "TRAINING-SET"

attr(,"LocalMetaData")$OLDID
[1] "5670"

attr(,"LocalMetaData")$Topics
[1] "crude"

attr(,"LocalMetaData")$Places
[1] "usa"
```

```
attr(,"LocalMetaData")$People
character(0)

attr(,"LocalMetaData")$Orgs
character(0)

attr(,"LocalMetaData")$Exchanges
character(0)

attr(,"Origin")
[1] "Reuters-21578 XML"
attr(,"class")
[1] "Reuters21578Document" "TextDocument"         "XMLDocument"
[4] "XMLAbstractDocument"  "oldClass"
```

XML 문서로 다양한 tag 및 meta data가 정의되어 있음을 확인할 수 있다.

> docs<-c("This is a text","This another one.", "My name is Eric")
Vector 소스로 부터 읽어들이는 예제로 문자열을 doc에 지정한다. 즉 3개의 문서를 읽어들이는 것과 같다.
> Corpus(VectorSource(docs))
A corpus with 3 text documents

Vector Source를 통해 Corpus를 생성한다.
> docsCorpus <- Corpus(VectorSource(docs))
> docsCorpus
A corpus with 3 text documents
> docsCorpus[[1]]
This is a text
위에서 지정한것 처럼 첫번째 문서는 “This is a text”임을 알 수 있다.

> writeCorpus(docsCorpus)

Corpus를 다른 object에서 가져온 경우 이를 저장하고자 하면 default working directory에 개별 파일로 저장된다. 예를 들면 사용자 정의로 만든 docs의 경우 1.txt, 2.txt 이런식으로 개별 파일이 생성된다. 문서내용을 보는 방법은 inspect문장을 통해서 array를 지정해서 선택하여 볼 수 있다.

> inspect(docsCorpus[1:3])
A corpus with 3 text documents

The metadata consists of 2 tag-value pairs and a data frame
Available tags are:
  create_date creator
Available variables in the data frame are:
  MetaID

[[1]]
This is a text

[[2]]
This another one.

[[3]]
My name is Eric

Document Handling

> reuters <- tm_map(reuters, as.PlainTextDocument)
앞에서 reuter 예제에서 읽어들인 XML 문서를 tm package의 기능인 tm_map을 이용해서 text로 전환한다.
> reuters
A corpus with 20 text documents
> reuters[[1]]
DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES
NEW YORK, FEB 26 -
Diamond Shamrock Corp said that
effective today it had cut its contract prices for crude oil by
1.50 dlrs a barrel.
    The reduction brings its posted price for West Texas
Intermediate to 16.00 dlrs a barrel, the copany said.
    "The price reduction today was made in the light of falling
oil product prices and a weak crude oil market," a company
spokeswoman said.
    Diamond is the latest in a line of U.S. oil companies that
have cut its contract, or posted, prices over the last two days
citing weak oil markets.
 Reuter
XML tag가 없어지고 일반 text형식으로 변환되었다.

> reuters <- tm_map(reuters, stripWhitespace)
> reuters[[1]]
DIAMOND SHAMROCK (DIA) CUTS CRUDE PRICES
NEW YORK, FEB 26 -
Diamond Shamrock Corp said that effective today it had cut its contract prices for crude oil by 1.50 dlrs a barrel. The reduction brings its posted price for West Texas Intermediate to 16.00 dlrs a barrel, the copany said. "The price reduction today was made in the light of falling oil product prices and a weak crude oil market," a company spokeswoman said. Diamond is the latest in a line of U.S. oil companies that have cut its contract, or posted, prices over the last two days citing weak oil markets. Reuter

중간의 빈칸이 제거되었다.

> reuters <- tm_map(reuters, tolower)
> reuters[[1]]
diamond shamrock (dia) cuts crude prices
new york, feb 26 -
diamond shamrock corp said that effective today it had cut its contract prices for crude oil by 1.50 dlrs a barrel. the reduction brings its posted price for west texas intermediate to 16.00 dlrs a barrel, the copany said. "the price reduction today was made in the light of falling oil product prices and a weak crude oil market," a company spokeswoman said. diamond is the latest in a line of u.s. oil

companies that have cut its contract, or posted, prices over the last two days citing weak oil markets. reuter

대문자를 소문자로 변경하여 사전에 있는 내용과 비교할 수 있도록 표준화 한다. 한글이나 특수문자가 들어 있는 경우 오류를 발생시키므로 이들 데이터에는 별도로 분리하여 처리하거나 삭제해야 한다.

> reuters <- tm_map(reuters, removeWords, stopwords("english"))
> reuters[[1]]
diamond shamrock (dia) cuts crude prices
york, feb 26 -
diamond shamrock corp effective cut contract prices crude oil 1.50 dlrs barrel. reduction brings posted price west texas intermediate 16.00 dlrs barrel, copany . " price reduction light falling oil product prices weak crude oil market," company spokeswoman . diamond line .. oil companies cut contract, posted, prices days citing weak oil markets. reuter

> tm_map(reuters, stemDocument)

텍스트 마이닝에는 언어별 stop word라는게 있다. 한글에서는 조사가 해당되며 띄어쓰기 등을 통해 확인할 수 있고, 영어의 경우도 띄어쓰기와 시제 등의 내용을 제거한다. 예를 들어 update는 updated, updating 등으로 변형된 내용이 있을 수 있는데 이는 updat로 표준화 된다.

6.3 Create Term-Document Matrix

읽어들인 문서를 plain text 전환, space 제거, lowercase로 변환, punctuation제거, stopword 처리 stemming등을 처리한 다음에 문서번호와 단어간의 사용여부 또는 빈도수를 이용하여 matrix를 만드는 작업이 term document matrix 작업이다.

> dtm<-DocumentTermMatrix(reuters)
reuter로 읽어들이고 변환한 문서를 document term matrix로 dtm을 생성한다.
> inspect(dtm[1:5,100:105])
A document-term matrix (5 documents, 6 terms)

Non-/sparse entries: 1/29
Sparsity : 97%
Maximal term length: 10
Weighting : term frequency (tf)

Terms
Docs abdul-aziz ability able abroad, abu accept
127 0 0 0 0 0 0
144 0 2 0 0 0 0
191 0 0 0 0 0 0
194 0 0 0 0 0 0
211 0 0 0 0 0 0

생성된 document term matrix에서 처음 5개 문서의 100번째에서 105번째 단어의 분포를 확인한다. 여기서 “ability”가 144번 문서에서 2번 사용되었음을 확인할 수 있다. 대부분의 단어들이 모든 문서에서 아용되지 않기 때문에 조회한 내용의 5개 문서와 6개 단어에서는 1개 단어가 1번 사용되었고 나머지 29개는 0로 표시된다. 그래서 sparsity가 29/30으로 96.6%에 해당된다.

```
> findFreqTerms(dtm,10)
 [1] "bpd"     "crude"   "dlrs"    "kuwait"  "march"   "mln"     "official" "oil"
 [9] "opec"    "pct"     "price"   "prices"  "reuter"  "saudi"
```

10회 이상 사용된 단어를 찾는 방법이다. findFreqTerms(dtm, 10, 15)로 지정하면 10에서 15회 사이로 사용된 단어가 출력되게 된다.

```
> findAssocs(dtm,"oil",0.65)
      oil    recent    united     (bpd)     lower minister,      opec  ecuador, emirates,
     1.00      0.75      0.72      0.71      0.71      0.71      0.70      0.69      0.69
 estimate     named   planned  pressure    prices   pricing  producer producing published
     0.69      0.69      0.69      0.69      0.69      0.69      0.69      0.69      0.69
   quota.   remarks    review
     0.69      0.69      0.69
```

findAssocs에서 oil과 연관성이 0.65 이상인 단어들이 표시된다. 즉, oil과 같이 사용될 확률로 계산이 된다. 이를 통해 무슨 내용이 많이 언급되는지를 알 수 있다.

```
> data("crude")
> tdm <- TermDocumentMatrix(crude)
> removeSparseTerms(tdm, 0.2)
```

removeSparseTerms(x, sparse)에서 sparse는 최대 sparsity값으로 해당 값까지는 허용된다. 즉 0.2인 경우 0.2를 넘는 경우는 삭제된다.

```
> inspect(removeSparseTerms(dtm,0.4))
A document-term matrix (20 documents, 3 terms)

Non-/sparse entries: 55/5
Sparsity           : 8%
Maximal term length: 6
Weighting          : term frequency (tf)

    Terms
Docs  march oil reuter
  127     0   3      1
  144     0   4      1
  191     0   2      1
  194     0   1      1
  211     0   2      1
  236     2   6      1
  237     1   2      1
  242     1   3      1
  246     1   2      1
  248     1   8      1
  273     1   5      1
  349     1   4      1
  352     1   4      1
  353     1   4      1
  368     1   3      1
  489     1   5      1
```

```
 502    1  5    1
 543    1  3    1
 704    1  1    1
 708    1  2    1
```

6.4 Dictionary

Dictionary는 복수의 문자들의 집합으로 text mining에서 사용되는 단어들의 집합이다. 여기에 단어를 추가할 수 있다.

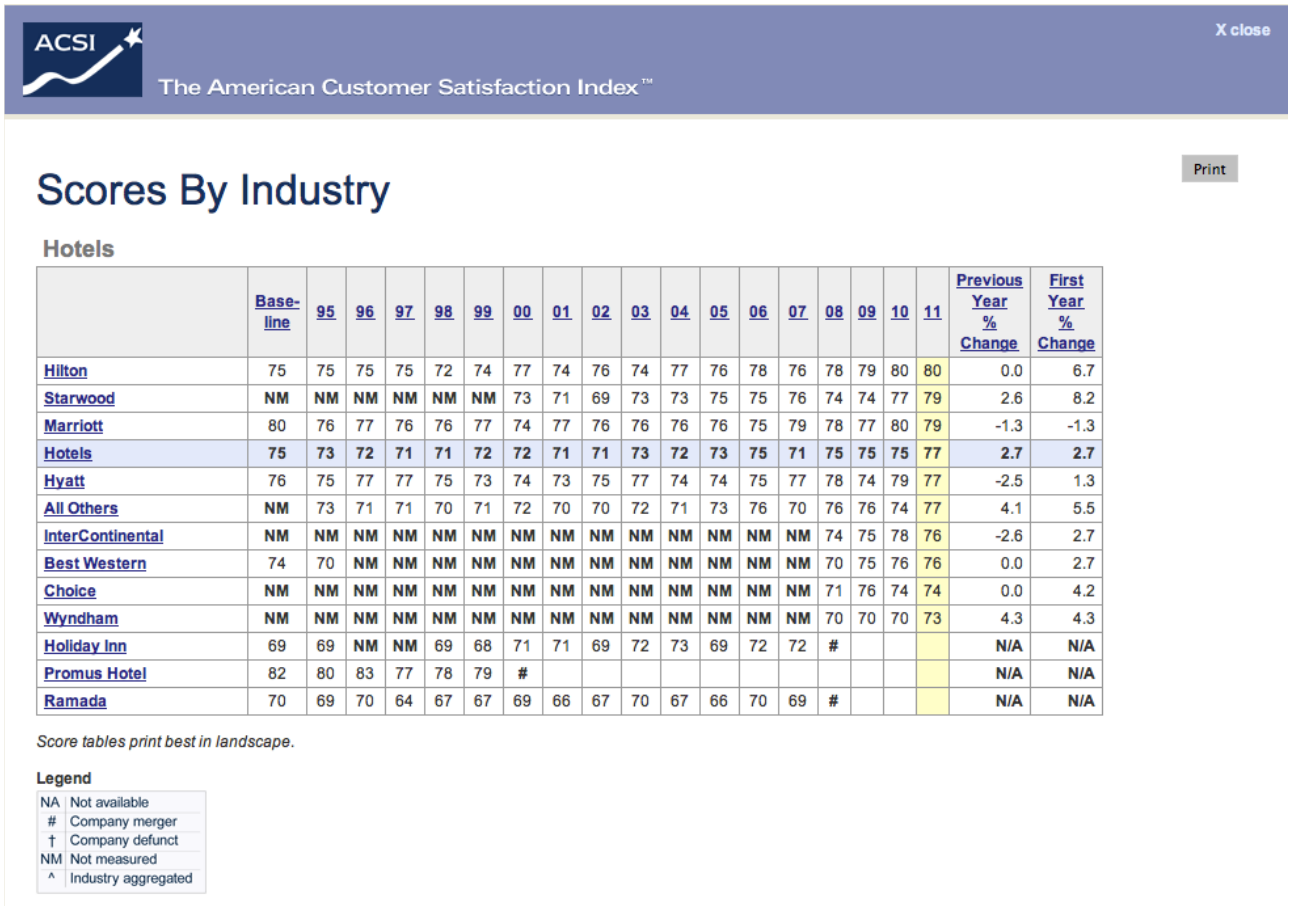
```
> d <- Dictionary(c("prices","crude","oil"))
> inspect(DocumentTermMatrix(reuters, list(dictionary=d)))
A document-term matrix (20 documents, 3 terms)

Non-/sparse entries: 41/19
Sparsity           : 32%
Maximal term length: 6
Weighting          : term frequency (tf)

    Terms
Docs  crude oil prices
 127    2  3    3
 144    0  4    2
 191    3  2    0
 194    4  1    0
 211    0  2    0
 236    1  6    2
 237    0  2    0
 242    0  3    1
 246    0  2    0
 248    0  8    5
 273    6  5    4
 349    2  4    0
 352    0  4    2
 353    2  4    1
 368    0  3    0
 489    0  5    2
 502    0  5    2
 543    3  3    3
 704    0  1    2
 708    1  2    0
```

Dictionary에 price, crude, oil단어를 등록하여 해당 dictionary를 이용해서 document term matrix를 분석하였다.

6.5 Sentiment Analysis

흔히 Sentiment Analysis, Opinion Mining 등으로 언급하는 내용이다. 문장에서 사용된 단어의 긍정과 부정여부에 따라 얼마나 긍정적인 단어가 많이 있는지의 여부로 score를 부여하여 긍정문장인지를 평가한다. 이를 이용해서 브랜드에 대한 평판이 긍정적인 추이가 증가하는지, 감소하고 있는지를 분석할 수 있다. 각 문장의 긍정인지 부정인지는 주체에 따라 다르게 해석할 수 있다. 자사의 브랜드가 apple이고 경쟁사가 samsung인 경우 context상에서 긍정/부정 여부가 달라질 수 있다. 이러한 측면등 복잡한 문장을 분석하는 경우 개별 문장/문서에 대해서는 오류가 생길 수 있다. 그러나 다양한 문서들이나 데이터를 가공하는 경우 그 추이를 보는데는 큰 무리가 없다. 따라서, sentiment analysis에 대해 매우 부정적일 필요는 없다.

예제로는 호텔들에 대한 서베이결과와 twitter 상의 sentiment analysis결과를 비교하는 내용을 사용하고자 한다.



ACSI The American Customer Satisfaction Index™ X close

Print

Scores By Industry

Hotels

	Base-line	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	Previous Year % Change	First Year % Change
Hilton	75	75	75	75	72	74	77	74	76	74	77	76	78	76	78	79	80	80	0.0	6.7
Starwood	NM	NM	NM	NM	NM	NM	73	71	69	73	73	75	75	76	74	74	77	79	2.6	8.2
Marriott	80	76	77	76	76	77	74	77	76	76	76	76	75	79	78	77	80	79	-1.3	-1.3
Hotels	75	73	72	71	71	72	72	71	71	73	72	73	75	71	75	75	75	77	2.7	2.7
Hyatt	76	75	77	77	75	73	74	73	75	77	74	74	75	77	78	74	79	77	-2.5	1.3
All Others	NM	73	71	71	70	71	72	70	70	72	71	73	76	70	76	76	74	77	4.1	5.5
InterContinental	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	74	75	78	76	-2.6	2.7
Best Western	74	70	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	70	75	76	76	0.0	2.7
Choice	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	71	76	74	74	0.0	4.2
Wyndham	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	NM	70	70	70	73	4.3	4.3
Holiday Inn	69	69	NM	NM	69	68	71	71	69	72	73	69	72	72	#				N/A	N/A
Promus Hotel	82	80	83	77	78	79	#												N/A	N/A
Ramada	70	69	70	64	67	67	69	66	67	70	67	66	70	69	#				N/A	N/A

Score tables print best in landscape.

Legend

NA	Not available
#	Company merger
†	Company defunct
NM	Not measured
^	Industry aggregated

> library(twitteR)
twitteR package는 twitter API에 접속해서 정보처리하는 package로 ROauth package와 같이 사용하면 직접 message를 posting하거나 일반적인 데이터 처리 limitation을 초과해서 처리할 수 있게된다.
> hilton.tweets <- searchTwitter("@hilton",n=1500)
@hilton을 언급한 내용을 최대 1500개의 message를 가져와서 hilton.tweet에 저장한다. 그러나 실제로는 대부분 1500개를 가져오지 못한다. 이는 접속한 계정/ip address 등에 의해 일정시간 동안의 사용량에 따라 자동적으로 제한된다.
> length(hilton.tweets)
[1] 267
267건의 메시지를 가져왔음을 알수 있다.
> class(hilton.tweets)
[1] "list"

> hilton.tweets[1:5]
[[1]]
[1] "IsaKjellsdotter: Simon på @Hilton i #Malmö vilken servitör! Fler killar som han och världen skulle se annorlunda ut! Fantastisk (Ping @HiltonStockholm)"

[[2]]
[1] "7amad999: @Hilton"

[[3]]
[1] "BirinciSeval: @hilton dalaman"

[[4]]
[1] "terlanorucov: @Hilton Baku"

[[5]]
[1] "ntake1966: また行きたいなあ@Hilton Times Square http://t.co/iLWLZgor"

5개의 메시지를 표시한 경우이다.

> tweet<-hilton.tweets[[1]]
> tweet$getScreenName()
[1] "IsaKjellsdotter"
해당 tweet 내용을 이용해서 screen name 정보를 가져온다.
> tweet$getText()
[1] "Simon på @Hilton i #Malmö vilken servitör! Fler killar som han och världen skulle se annorlunda ut! Fantastisk (Ping @HiltonStockholm)"
해당 tweet 내용으로 메시지의 텍스트를 가져온다. 이러한 기능으로 해당 tweet관련 사용자의 위경도를 가져올 수 있다.

> library(plyr)
plyr은 splitting, applying and combining data 용도로 사용하는 package이다.

> hilton.text <- laply(hilton.tweets, function(t)t$getText())
가져온 전체 twitt 내용을 이용해서 text만을 추출해서 hilton.txt에 저장한다.
> head(hilton.text,3)
[1] "Simon på @Hilton i #Malmö vilken servitör! Fler killar som han och världen skulle se annorlunda ut! Fantastisk (Ping @HiltonStockholm)"
[2] "@Hilton"
[3] "@hilton dalaman"
text내용 중에서 앞에서 3개만을 가져온다. 끝에서 n개를 가져오려면 tail을 이용하면 된다.

> pos.word=scan("./positive-words.txt", what="character", comment.char=";")
Read 2006 items
> neg.word=scan("./negative-words.txt",what="character",comment.char=";")
Read 4783 items

긍정, 부정에 관련된 영어관련 파일을 읽어서 저장한다. 또는 web에서 직접 가져온다.

>pos.word = scan('/Users/marcinkulakowski/Downloads/r/positive-words.txt',what='character',comment.char=';')

>neg.word = scan('/Users/marcinkulakowski/Downloads/r/negative-words.txt',what='character',comment.char=';')

> pos.words <- c(pos.word,"upgrade")
> neg.words <- c(neg.word, "wtf","wait","waiting","epicfail","mechanical")
기존 긍정과 부정 단어집에 내용에 따른 별도 정의나 추가 단어를 누적한다.

아래 내용은 긍정/부정 단어들을 이용해서 긍정 및 부정에 대한 score를 부여하는 루틴으로 typing을 치지 않고 제공된 파일을 이용해서 직접 loading하여 처리한다.
>score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
score를 부여할 변수와 긍정단어, 부정단어들을 할당하고 ***progress***를 표시할것인지를 결정한다.

require(plyr)
require(stringr)

 # we got a vector of sentences. plyr will handle a list
 # or a vector as an "l" for us
 # we want a simple array ("a") of scores back, so we use
 # "l" + "a" + "ply" = "laply":
scores = laply(sentences, function(sentence, pos.words, neg.words) {

긍정/부정에 따라 ***score***를 ***return***하도록 한다.

 # clean up sentences with R's regex-driven global substitute, gsub():
sentence = gsub('[[:punct:]]', '', sentence)
sentence = gsub('[[:cntrl:]]', '', sentence)
sentence = gsub('\\d+', '', sentence)

구문점이나 특수기호, 숫자문자열을 빈칸으로 대체한다.
 # and convert to lower case:
sentence = tolower(sentence)

단어를 사전과 비교하기 위해 ***lower case***로 변환한다.

 # split into words. str_split is in the stringr package
word.list = str_split(sentence, '\\s+')
 # sometimes a list() is one level of hierarchy too much
words = unlist(word.list)

문장을 단어 리스트로 변환한다.

 # compare our words to the dictionaries of positive & negative terms
pos.matches = match(words, pos.words)
neg.matches = match(words, neg.words)

단어들을 긍정과 부정단어와 비교하여 갯수를 저장한다.

 # match() returns the position of the matched term or NA
 # we just want a TRUE/FALSE:

```
pos.matches = !is.na(pos.matches)
neg.matches = !is.na(neg.matches)

# and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
score = sum(pos.matches) - sum(neg.matches)
```

긍정과 부정단어의 합을 차감하여 점수를 산출하고 돌려준다.

```
return(score)
}, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=sentences)
return(scores.df)
}
```

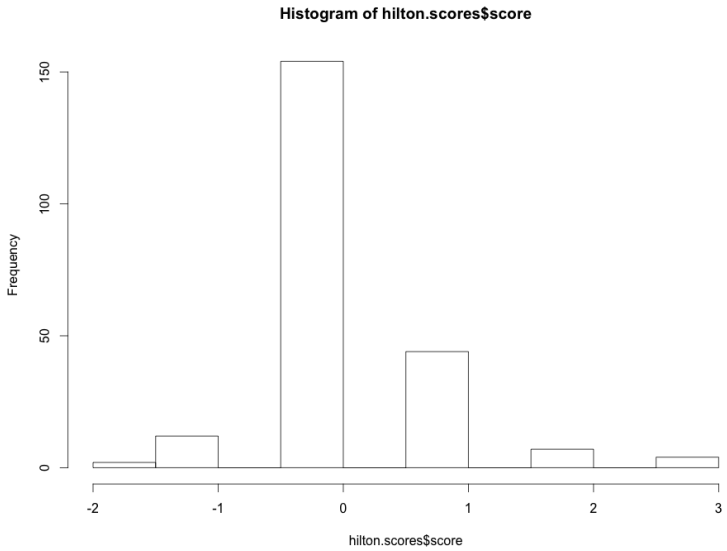
점수와 문장을 data frame 형태로 돌려준다.

```
>
sample=c("You'reawesomeandIloveyou","IhateandhateandhateSoangry.Die!","Impressedandamazed:youarepeerlessinyourachievementofunparalleledmediocrity.","I love you")
> result <- score.sentiment(sample,pos.words,neg.words)
> result$score
[1] 0 0 0 1
```

위의 내용은 별도의 sample로 sample에 저장된 문장들에 대해 sentimental analysis를 수행한것을 보여준다. 단어들이 붙어있는 것들은 구분을 하지 못해 0으로 처리되고 단어가 구분이 명확히 된 4번째 값은 긍정단어인 love가 들어가 있어서 1점이 부여된다. 따라서, 문장에 긍정과 부정단어가 혼재한 경우 긍정단어가 많이 들어가 있으면 긍정으로 처리된다.

```
>hilton.text <- hilton.text[!Encoding(hilton.text)=="UTF-8"]
>hilton.scores=score.sentiment(hilton.text,pos.words,neg.words,.progress='text')
==========================================================================
==========================================================| 100%
> hist(hilton.scores$score)
```

tweet 내용들을 각각 점수를 산출해서 histogram을 산출한 결과 중립적인 내용을 제외하고 보면 긍정이 더 많음을 알수 있다.



```
> hilton.scores$hotel='Hilton'
> hilton.scores$code='HL'
> library("ggplot2")
> qplot(hilton.scores$score)

# Intercontinental
intercontinental.tweets=searchTwitter('@interconhotels',n=1500)
class(tweet)
intercontinental.text=laply(intercontinental.tweets,function(t)t$getText())
intercontinental.text <- intercontinental.text[Encoding(intercontinental.text)=="UTF-8"]
intercontinental.scores=score.sentiment(intercontinental.text,pos.words,neg.words,.progress='text')
intercontinental.scores$hotel='Intercontinental'
intercontinental.scores$code='IC'
```

이렇게 각 호텔의 twitter에서의 평판을 입수하여 서베이 결과와 cross-check을 함으로써 유사한 결과가 나오는 경우 이를 지속적으로 활용하여 주단위 feedback을 받을 수 있다. 인터넷의 정보를 scrap하여 이를 graph로 그리는 내용은 이번에 다루지 않기로 한다.

6.6 한글처리[12]

한글관련 text mining package는 KoNLP가 있다. 예제는 전희원님의 site에서 가져왔고 데이터 파일은 별도로 구해서 작성하였다.

```
library(KoNLP)
library(arules)
library(igraph)
library(combinat)


f <- file("광장_최인훈.txt", encoding="UTF-8")
fl <- readLines(f)
# text line을 f로 부터 읽어와서 fl에 저장한다.
close(f)
# 읽어드린 파일 f를 close한다.

tran <- Map(extractNoun, fl)
# 읽어드린 fl 내용에서 명사를 추출한다.
tran <- unique(tran)
# 라인별 데이터를 unique하게 만든다.
tran <- sapply(tran, unique)
# 라인내의 데이터를 unique하게 만든다.
```

[12] http://freesearch.pe.kr/archives/2726

```
tran <- sapply(tran, function(x) {Filter(function(y) {nchar(y) <= 4 && nchar(y) > 1 && is.hangul(y)},x)} )
# 2자에서 4글자사이의 한글 단어만을 선택한다.

tran <- Filter(function(x){length(x) >= 2}, tran)
names(tran) <- paste("Tr", 1:length(tran), sep="")
# tran list의 name에 “Tr”과 순번을 결합하여 입력한다.

wordtran <- as(tran, "transactions")
# tran 데이터를 arules package의 apriori()를 이용해 association analysis를 하기 위해  transaction type으로 변환한다.

#co-occurance table
wordtab <- crossTable(wordtran)
# 단어들의 cross table을 생성하여 같이 사용되는 단어들을 파악할 수 있도록 한다.

ares <- apriori(wordtran, parameter=list(supp=0.2, conf=0.1))
```

연관성 분석 1세대 알고리즘인 apriori를 이용해서 wordtran 데이터 중에서 빈도수가 20%이상이고 해당 단어가 있을때 다른 단어가 같이 있을 확률이 10% 이상인 규칙을 도출하도록 한다. Support와 confidence를 10%, 10%로 설정한 경우 1754개의 rule이 도출된다. 5%수준으로 모두 설정한 경우 시스템이 꽤 오래동안 실행되며 도중에 중단하였다. 20%와 10%로 설정한 경우 129개 rule이 도출되었고 메모리 부족에 대한 warning이 없었다. 물론 저자가 사용한 환경은 64bit에 16GB RAM 환경이다. 따라서 해당 spec이하의 컴퓨터에서 사용하는 경우 support를 더 높게 돌려본 다음 비즈니스 적으로 의미 있는 수준으로 조절하면 된다. 데이터에 따라 해당 옵션값이 너무 작으면 결과가 도출되지 않을 수 도 있다. 참고로 apriori를 이용한 실무분석은 현실성이 없다. FP-Tree를 64bit 환경에서 64GB 이상의 메모리를 갖고 있어야 이용해야 유통사의 sku level수준의 다양한 상품을 갖고 큰 용량의 데이터를 처리할 수 있다. 여기서는 교육적 측면에서만 다루고자 한다.

```
inspect(ares)
# 규칙들을 보기 위해서는 inspect명령어를 사용한다. 결과는 A인 경우 B이다가 support, confidence, lift가 표시된다. Lift는 random하게 추출한 rule보다 향상된 성능을 보이는 수준이다.
```

6.7 Exercise

```
library(twitteR)
rdmTweets <- userTimeline("rdatamining",n=2000)

# rdatamining 사용자 계정의 내용을 2000건 까지 가져온다. 최대 값은 3200으로 고정되어 있다.

nDocs <- length(rdmTweets)
rdmTweets[1:3]

for (i in 1:5) {
  cat(paste("[[",i,"]]",sep=""))
```

```
  writeLines(strwrap(rdmTweets[[i]]$getText(),width=73))
}

df <- do.call("rbind",lapply(rdmTweets,as.data.frame))
# tweet을 가져온 내용을 data frame형식으로 저장한다.

dim(df)

library(tm)
myCorpus <- Corpus(VectorSource(df$text))
myCorpus <- tm_map(myCorpus, tolower)
myCorpus <- tm_map(myCorpus, removePunctuation)
myCorpus <- tm_map(myCorpus, removeNumbers)

removeURL <- function(x) gsub("http[[:alnum:]]*","",x)
# http로 연결되는 순자와 문자들을 빈칸으로 대체하는 URL 제거 함수를 정의한다.

myCorpus <- tm_map(myCorpus, removeURL)

myStopwords <- c(stopwords('english'), "available","via")
# stop word에 available과 via를 추가한다.

idx <- which(myStopwords %in% c("r","big"))
myStopwords <- myStopwords[-idx]
# r이나 big이란 단어를 stop word에서 제외한다.

myCorpus <- tm_map(myCorpus, removeWords,myStopwords)

myCorpusCopy <- myCorpus
myCorpus <- tm_map(myCorpus, stemDocument)
for (i in 11:15) {
  cat(paste("[[",i,"]]", sep=""))
  writeLines(strwrap(myCorpus[[i]],width=73))
}
myCorpus <- tm_map(myCorpus, stemCompletion, dictionary=myCorpusCopy)
inspect(myCorpus[11:15])

miningCases <- tm_map(myCorpusCopy, grep, pattern="\\<mining")
sum(unlist(miningCases))

minerCases <- tm_map(myCorpusCopy, grep, pattern="\\<miners")
sum(unlist(minerCases))

myCorpus <- tm_map(myCorpus, gsub, pattern="miners",replacement="mining")
# miners를 mining으로 대체한다.

myTdm <- TermDocumentMatrix(myCorpus, control=list(wordLength=c(1,Inf)))
# Term document matrix를 작성하는데 단어의 최소 길이는 1에서 최대는 무한대로 지정한다.

myTdm
```

```
idx <- which(dimnames(myTdm)$Terms=="r")
inspect(myTdm[idx+(0:5),101:110])

findFreqTerms(myTdm,lowfreq=10)
termFrequency <- rowSums(as.matrix(myTdm))
termFrequency <-subset(termFrequency, termFrequency >=10)
# 빈도수가 10개 이상인 단어들의 부분집합을 만든다.

library(ggplot2)
qplot(names(termFrequency),termFrequency,geom="bar") + coord_flip()
barplot(termFrequency, las=2)

findAssocs(myTdm,'data',0.25)
findAssocs(myTdm,'mining',0.25)


library(wordcloud)
# word cloud package는 단어들을 이용해서 사용빈도수에 따라 크기를 다르게 해서 random하
게 위치나 색상을 할당해서 표시해 주는 기능이다.

m <- as.matrix(myTdm)
wordFreq <- sort(rowSums(m),decreasing=TRUE)
grayLevels <-gray((wordFreq+10) / (max(wordFreq)+10))
wordcloud(words=names(wordFreq),freq=wordFreq,min.freq=3, random.order=F,
colors=grayLevels)

myTdm2 <- removeSparseTerms(myTdm,sparse=0.95)
m2 <- as.matrix(myTdm2)
# m2변수는 다음 chapter에서도 사용할것이므로 저장해 놓기 바란다.

distMatrix <- dist(scale(m2))
# scale을 정규화 하고 단어들의 distance를 구한다.

fit <- hclust(distMatrix, method="ward")
# clustering을 실시한다.

plot(fit)
rect.hclust(fit,k=10)
# clustering을 10개 집단으로 묶어 준다.

(groups <- cutree(fit,k=10))

m3 <- t(m2)
k <- 8
kmeansResult <- kmeans(m3,k)
round(kmeansResult$centers,digit=3)

for (i in 1:k) {
  cat(paste("cluster ",i,": ",sep=""))
  s <- sort(kmeansResult$centers[i,],decreasing=T)
  cat(names(s)[1:3],"\n")
```

```
}


library(fpc)
pamResult <- pamk(m3, metric="manhattan")
(k <- pamResult$nc)

pamResult <- pamResult$pamobject
for (i in 1:k) {
  cat(paste("cluster ",i,": " ))
  cat(colnames(pamResult$medoids)[which(pamResult$medoids[i,]==1)],"\n")
}
layout(matrix(c(1,2),2,1))
plot(pamResult, color=F, labels=4, lines=0, cex=.8, col.clus=1,col.p=pamResult$clustering)
layout(matrix(1))
```

Chapter 7

Social Network Analysis

"Social Network Analysis가 R에서 가장 취약한 부분이 아닐까 한다."

이번 chapter에서는 igraph를 이용한 R에서의 social network analysis를 다루고자 한다. R에서 social network을 다루는 package는 매우 다양하다. 그러나 모든 package를 다룰 필요는 없다고 생각하며 너무 지나치게 이론적인 배경에 집중하다 보면 social network analysis를 통한 insight를 얻는데 어려움이 있다고 생각한다. 중요한건 일단 데이터를 입수해서 이를 social network graph로 처리하고 속성에 따라 그래프를 다양하게 처리함으로써 의미 있는 insight를 얻는게 중요하다고 생각한다. 좋은 참고 자료는 Stanford site에 있는 http://www.stanford.edu/~messing/Affiliation%20Data.html 내용을 참조하면 상세하게 파악할 수 있으리라 생각한다. 꼼꼼하게 너무 잘 만들어져 있어서 좋은 내용이나 교육과정에서 예제로 쓰기에는 너무 많은 분량이라 다루지 않았다.

7.1 Social Network 개념[13]

7.1.1 용어정의

소셜 네트웍은 node 또는 vertex와 link 또는 edge로 구성된 그래프로, 노드는 고객, 링크는 고객과 고객간의 관계로 표시할 수 있다. 그리고 링크가 방향성이 있는지 없는지에 따라 directed, undirected로 구분할 수도 있다. 이러한 방식으로 통신사에서 A고객이 B고객한테 전화를 했고 B가 C한테 문자를 보내고 C가 A한테 전화를 하는 등의 행태를 노드와 링크로 구성하면 하나의 그래프가 되는 것이다. 통신사에서의 한달간의 통화 데이터가 수억건에 달하고 문자 및 앱(application)을 이용한 데이터 통신까지를 생각하면 엄청난 규모의 네트웍이 구성될 수 있다.

그리고 유통업체 처럼 고객과 고객의 직접적인 관계가 아닌 고객과 구매상품의 관계를 통해 고객을 연결하는 경우 이를 bi-par-type graph라고 한다. 소셜 네트웍에 대한 이론적 배경과 용어를 하나하나 다루다 보면 책 한권은 나올 내용이고 따분하고 지루할 수 있으므로 이정도에서 생략하겠다.

7.1.2 현황

소셜 네트웍 분석에 대한 이론은 온랜 역사를 가진 분야로 IT기술의 발전에 따라 최근 각광을 받고 있다. 주로 통신 및 온라인 소셜 미디어 등에서 관심을 갖고 있고 게임 및 유통업체에서도 관심을 갖고 있는 분야이다.

---

[13] "Big Data Analysis for CRM using R", 김경태

흔히 일반인이 접할 수 있는 소셜 네트웍 분석은 Facebook 등에서 친구들의 관계를 그래프 형태로 보여주는 등 익숙하게 접할 수 있고 일부 소셜 미디어 검색 및 관리 기능을 갖고 있는 솔루션이나 웹 페이지에서 이러한 기능을 보여주고 있다. 최근 한국에서는 정치권에서의 관심으로 많이 들 활용하려는 노력이 보이고 있다. 그래서 흔히 influencer라는 용어가 매우 익숙하게 일반인들에게도 알려져 있는것 같다.

그러나 가장 많이 필요로 하는 통신사 들에서는 막상 데이터 처리 속도라든지의 기술적인 문제로 어려움을 겪고 있고 활용상에 어려움이 존재하는 것 같다.

7.1.3 솔루션

소셜 네트웍 솔루션으로는 KXEN, SAS, XTRACT, Indiro, Onalytica, Unicet, Pajek, Inflow 등 다수의 솔루션들이 있다. 각각 reference들과 장점들이 있으면서도 데이터 로딩 속도, visualization 기능 등의 제약이 있어서 아직 성숙된 시장은 아닌것으로 판단되나 급성장 하고 있는 분야이다. A사의 제품은 데이터 로딩 속도나 visualization 등에 무리없는 기능을 보이며, community detection, role define 등의 기능을 통해 derived variable을 생성하여 소셜 네트웍 정보를 활용하여 classification이나 segmentation에 활용하여 모델링 성능을 높이는데 활용하는 기능이 뛰어나며, B사의 제품은 visualization 이 매우 취약한 것으로 알려져 있고, C사의 경우 다양한 통계값을 보여주고 있으나 대용량 데이터 활용보다는 학문적 연구에 도움이 되는 수준인 경우등 평가에 사용자들이 어려움이 있을 것으로 예상된다. R로 대용량 데이터를 처리해 본적이 없지만 Big Data 기반 구조에서는 무리가 없으리라 생각된다. 이부분은 다음에 실제 대용량의 데이터를 처리해 보고 평가정보를 올리겠다. 참고로 대용량 처리를 하는데 문제가 없다고 언급하려면 저자의 경험으로 천만 고객의 8억건 정도의 통신사 CDR을 로딩하고 처리해서 그래프를 그려보고 이를 분석한 내용을 write-back하는 정도의 시간이 3시간 정도면 적합하다고 생각한다. 참고로 A사 제품으로 500만 고객의 천만건 정도의 구매정보를 이용한 분석에 Dual Core, 4GB RAM의 mac book pro에서 vmware상의 windows 환경에서 30분 이내에 처리하였다.

7.1.4 활용방안

소셜 네트웍의 가장 큰 활용방안은 네트웍을 구성하여 몇개의 집단으로 구성되는지, 집단간의 특징은 무엇이고 해당집단에서 영향력있는 고객은 누구인지, 시간의 흐름에 따른 고객상태의 변화에 따라 다음에 누가 영향을 받을 지를 이용해서 churn/acquisition prediction, fraud, product recommendation 등에 활용할 수 있다.

7.2 문서내 단어간의 연관성을 이용한 **Social Network**

이전 chapter에서 사용하던 데이터 중에서 m2를 이용하여 계속 진행을 하고자 한다.

```
> m2[5:10,1:20]
            Docs
Terms        1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
 examples    1 0 0 1 1 0 0 0 0  0  0  0  0  0  0  1  0  0  0  0
 introduction 0 0 0 0 0 0 0 0 0  0  0  0  0  0  1  0  0  0  0  0
 mining      0 1 0 1 0 0 0 0 1  1  1  1  1  1  0  0  0  0  0  0
 network     1 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0  0  0  0  0
```

```
  package     1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
  parallel    0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1
```

> m2[m2>=1] <-1
단어가 문서에서 사용된 횟수를 사용 여부로 0, 1값을 변환한다.
> termMatrix <- m2 %*% t(m2)
위의 matrix 공식을 이용하면 단어와 단어간의 구조로 변경된다. 그리고 t() 함수는 transpose 함수이다.

```
> termMatrix[5:10,5:10]
             Terms
Terms         examples introduction mining network package parallel
  examples          21           2      6       3       3        1
  introduction       2          11      2       2       0        0
  mining             6           2     55       1       2        1
  network            3           2      1      16       2        0
  package            3           0      2       2      17        2
  parallel           1           0      1       0       2        9
```
변환된 matrix의 모습을 보면 어떤 단어가 다른 단어와 얼만큼 사용되었는지를 알 수 있다.

```
> library(igraph)
> g <- graph.adjacency(termMatrix, weight=T, mode="undirected")
```
Matrix 데이터를 이용해서 그래프를 생성한다. 여기서 그래프의 방향성은 단어의 연관성과 상관이 없으므로 unidirected로 변환하도록 조건을 준다.

```
> g <- simplify(g)
```
단순화 시키기 위해 사용하는 함수이다.

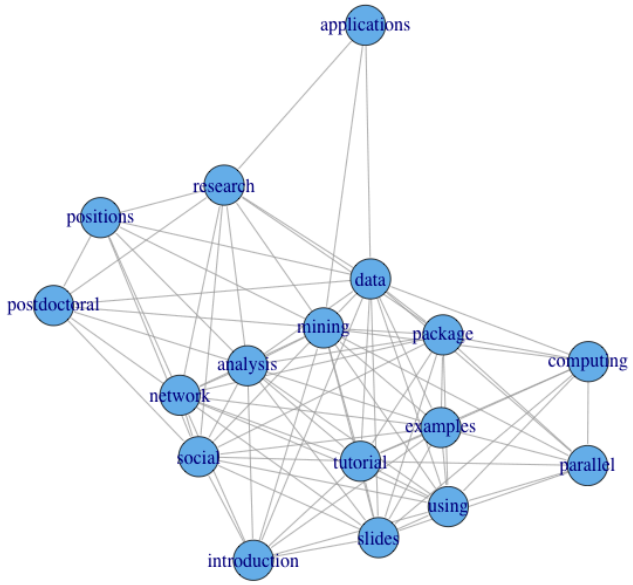
```
> V(g)$label<-V(g)$name
> V(g)$degree <-degree(g)
> V(g)$degree
 [1] 13  3  8 14 12  9 16 12 12  8  7  7  9 12 12 12 12
```
Social network에서 처음 나오는 measure이다. 해당 node 또는 vertex가 몇개와 연결되어 있는지에 대한 값이다. 통신사에서의 CDR에서는 나와 통화나 문자등을 주고 받은 unique한 전화번호 수라고 할 수 있다.

```
> layout1 <- layout.fruchterman.reingold(g)
```
Graph drawing을 위해 verticies의 위치를 결정하는 단순한 함수이다. layout 관련 함수에는 layout.kamada.kawai 등 다른 내용도 있으나 간단히 말해서 의미 있게 배치를 해주는 내용이다. 이러한 layout을 처리하지 않고 그냥 plot을 하면 random seed를 지정하지 않으면 매번 random하게 graph를 보여준다. 얼핏보면 의미 있어 보일수도 있으나 의미 없는 내용이다.

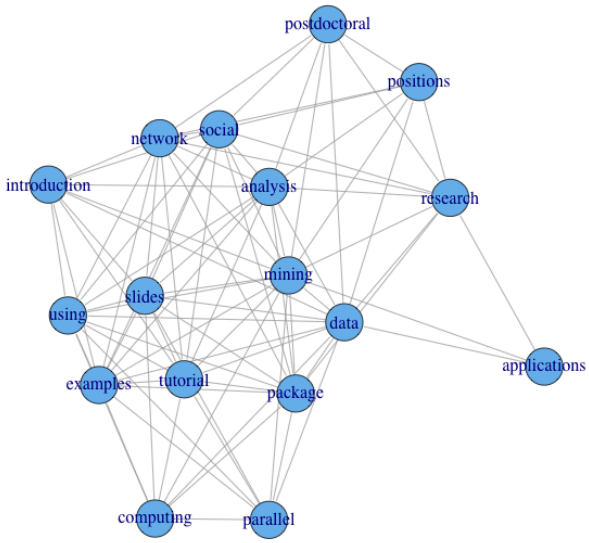
```
> plot(g,layout=layout1)
```

아래 그래프는 이러한 layout을 적용하여 drawing한 사례인데 parallel과 computing이 관련이 많다고 볼수 있으며, research와 postdoctoral이 관련이 높음을 보인다. 이러한 내용은 하나의 유사집단으로 볼 수 있으며 이러한 집단의 profile은 의미가 있게 된다. 이러한 social network graph의 경우 이러한 집단을 community라고 하며, 각 노드는 community에서 role이 정의가 된다. 예를들어 자신의 community와 다른 community에 모두 연결이 높은 것은 influencer, 집단 내에서 내부에서 여러개 노드와 연결된 것은 해당 community에서 leader position이고,

community와 community를 연결하는데 사용되는 노드는 bridge역할을 하는 노드고, community에서 끝단에 위치한 노드로 다양한 노드와 연결되지 않은것은 passive하다고 정의할 수 있다. 단어가 아니라 고객으로 생각한다면 우리는 social marketing을 위해 influencer와 local leader 또는 bridge 순으로 marketing을 해야 지 효율적일것이다.



plot(g, layout=layout.kamada.kawai)

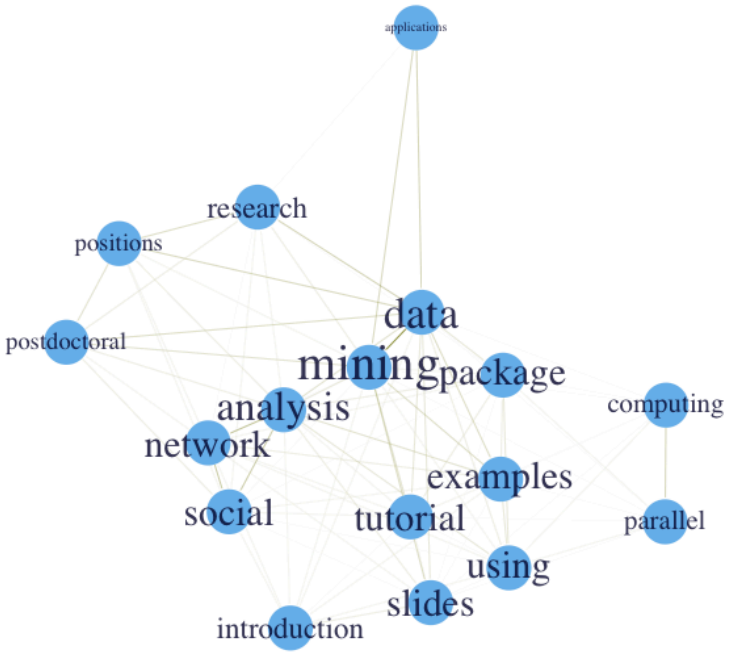


> V(g)$label.cex <- 2.2 * V(g)$degree / max(V(g)$degree) + .2

Vertecies의 label 크기를 전체 vertices에서 최대 degree와의 비율로 크기를 차별화 해서 시각적 효과를 극대화 한 경우이다.

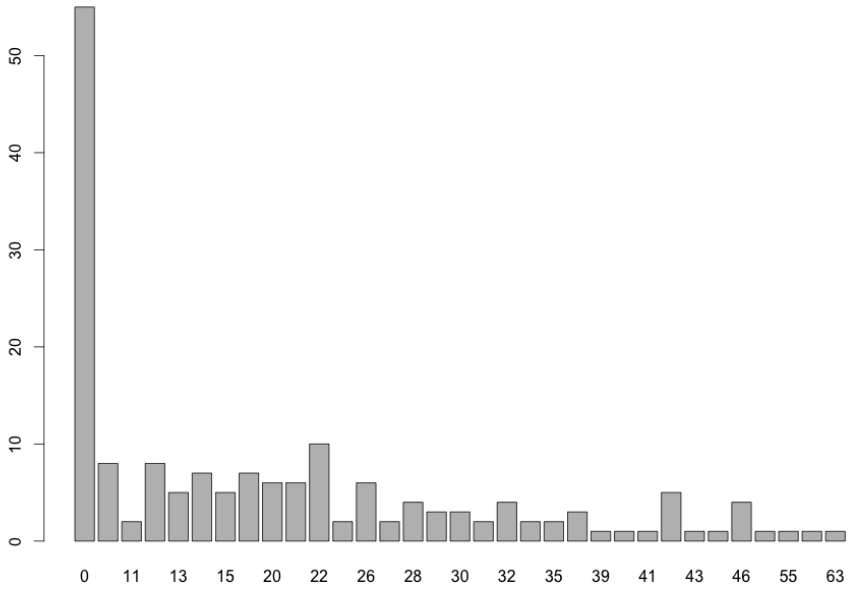
```
> V(g)$label.color <- rgb(0,0,.2,.8)
> V(g)$frame.color <- NA
> egam <- (log(E(g)$weight)+.4) / max(log(E(g)$weight)+.4)
```
연결하는 link 또는 edge의 굵기를 차별화한 경우다. 이는 목적별로 다르게 활용할 수 있다.

```
> E(g)$color <- rgb(.5,.5,0,egam)
> E(g)$width <- egam
> plot(g,layout=layout1)
```



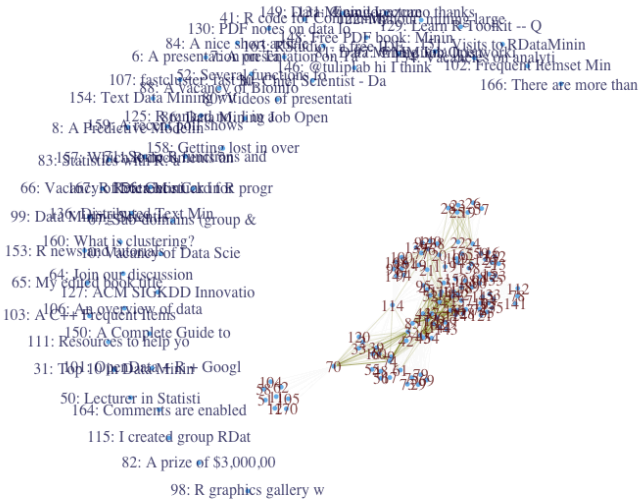

```
idx<-which(dimnames(termDocMatrix)$Terms %in% c("r", "data", "mining"))
# r과 data 또는 mining 단어가 어디에 있는지 index 값을 idx에 저장한다.
M <- termDocMatrix[-idx,]
# 해당 인덱스의 단어를 matrix에서 제거한다.

tweetMatrix <- t(M) %*% M
# 해당 단어들의 matrix를 구성한다.

library(igraph)
g <- graph.adjacency(tweetMatrix, weight=T, mode="undirected")
V(g)$degree <- degree(g)
```

```
g <- simplify(g)
V(g)$label <- V(g)$name
V(g)$label.cex <- 1
V(g)$label.color <- rgb(.4,0,0,.7)
V(g)$size <-2
V(g)$frame.color <- NA
barplot(table(V(g)$degree))
```



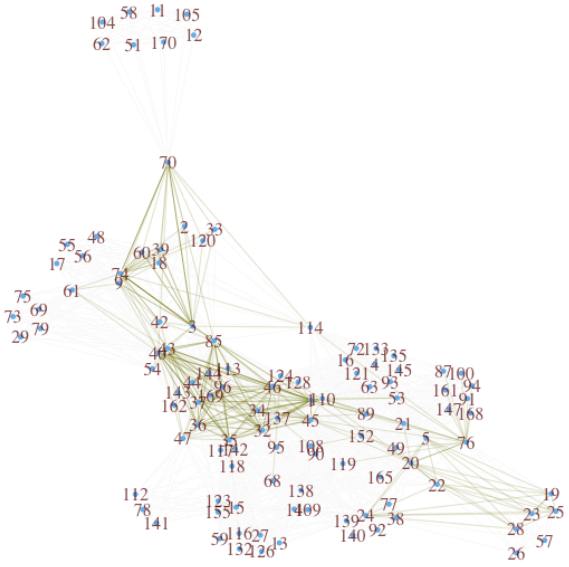
Simplify함수는 loop나 multiple edge를 제거해서 단순하게 해준다. 0인 값이 제일 많은 이유는 sparcity에 의해 삭제된 경우이기 때문이다.

```
idx <- V(g)$degree==0
# Degree가 0인 vertices의 index를 저장한다.
V(g)$label.color[idx] <- rgb(0,0,.3,.7)
# Degree가 0인 경우 label 색상을 할당한다. rgb()함수는 RGB 색상을 0에서 1사이의 값으로 할당한다.
V(g)$label[idx] <- paste(V(g)$name[idx], substr(df$text[idx],1,20),sep=": ")
# Degree가 0인 경우에 label에 번호와 내용으로 표시하도록 한다.
egam <- (log(E(g)$weight)+.2) / max(log(E(g)$weight) +.2)
# Edge weight 비중에 따라 값을 할당한다.
E(g)$color <- rgb(.5,.5,0,egam)
E(g)$width <- egam
layout2 <- layout.fruchterman.reingold(g)
plot(g,layout=layout2)
```



Degree가 0으로 처리된 것들은 번호와 text가 표시되고 나머지는 관계에 따라 network이 그려진다.

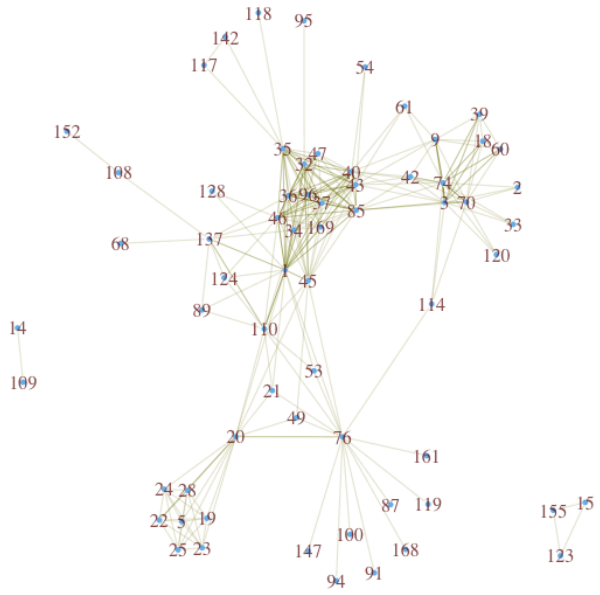
g2 <- delete.vertices(g,V(g)[degree(g)==0])
# degree가 0인 vertices가 삭제된다.
plot(g2, layout=layout.fruchterman.reingold)



구조가 보다 잘 드러나게 drawing된다.

g3 <- delete.edges(g, E(g)[E(g)$weight <=1])
g3 <- delete.vertices(g3, V(g3)[degree(g3)==0])

# 더 단순화 하기 위해서 edge weight가 1보다 작은 경우와 vertices의 degree가 0인 경우 삭제한다.
plot(g3, layout=layout.fruchterman.reingold)



위의 그래프를 보면 하단 좌측에 하나의 그룹과 중앙의 그룹들을 볼 수 있다. 이 집단의 내용을 보려면 아래와 같이 확인할 수 있다. 하나는 parallel computing 중앙은 social network analysis에 대해 언급한 community임을 알 수 있다.

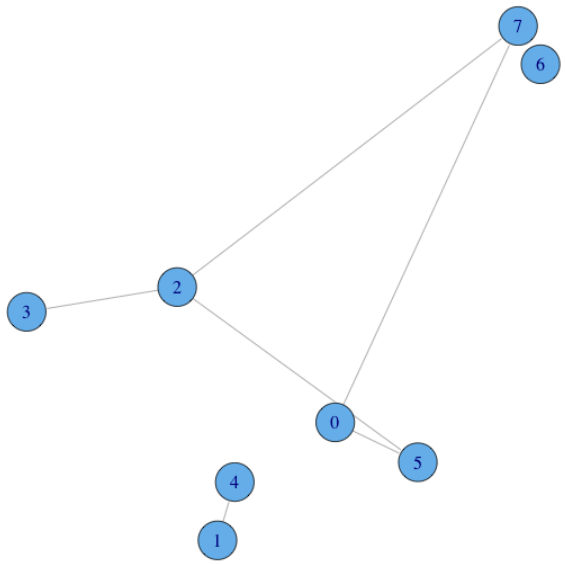
> df$text[c(5,19,22,23,24,28)] # parallel computing
[1] "A Simple Example of Parallel Computing on a Windows (and also Mac) Machine http://t.co/T9Tcw2Fi"
[2] "Easier Parallel Computing in R with snowfall and sfCluster http://t.co/BPcinvzK"
[3] "Parallel Computing with R using snow and snowfall http://t.co/nxp8EZpv"
[4] "State of the Art in Parallel Computing with R http://t.co/zmClglqi"
[5] "Slides on Parallel Computing in R http://t.co/AdDVxbOY"
[6] "The R Reference Card for Data Mining is updated with functions & packages for handling big data & parallel computing. http://t.co/FHoVZCyk"
> df$text[c(32,34,35,40,43,46,47,85)] # social network analysis
[1] "A statnet Tutorial for social network analysis http://t.co/dg3SJqEV"
[2] "Tutorials on using statnet for network analysis http://t.co/v7bsEkJx"
[3] "Slides on Social network analysis with R http://t.co/QFC6Y6C3"
[4] "Post-doc positions at NTU Singapore on distributed systems and social network analysis http://t.co/KFsAydKd"
[5] "Postdoc Position in Social Network Analysis, Tartu, Estonia http://t.co/Gx0NnxNv"
[6] "Examples on R for Social Network Analysis: http://t.co/19dcdvzp"
[7] "An online textbook on Introduction to social network: http://t.co/WRMXV19y"
[8] "Vacancy of Research Scientist - Natural Language Processing & Social Network Analysis, CSIRO, Australia http://t.co/FumQEQSy"

graph.incidence는 아래와 같은 형식으로 bipartite graph를 생성할 수 있다.
> inc <- matrix(sample(0:1, 15, repl=TRUE), 3, 5)
0,1의 값으로 3 by 5의 matrix를 random하게 생성한다.
> colnames(inc) <- letters[1:5]
소문자 a, b, c, d, e가 생성된다.

> rownames(inc) <- LETTERS[1:3]
대문자 A, B, C가 생성된다.

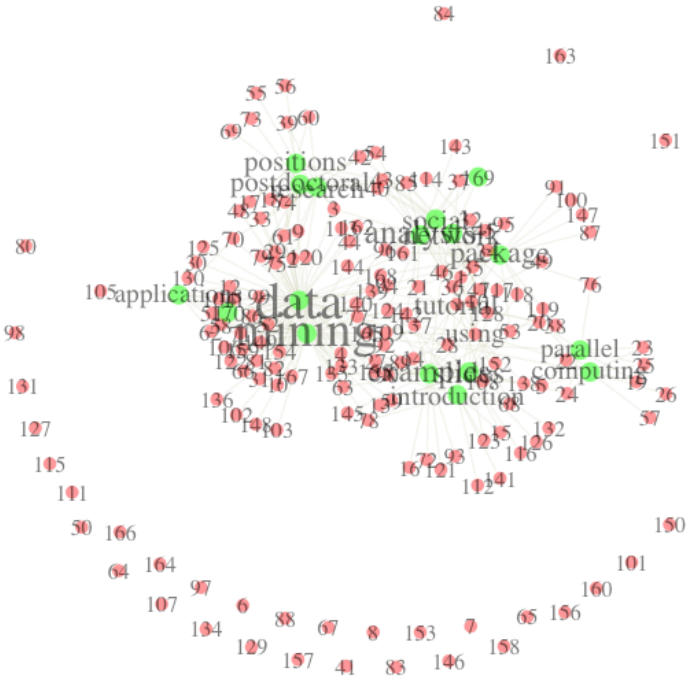
> eric <- graph.incidence(inc)
그래프를 생성한다.
> plot(eric)
그래프를 그린다.

> inc
  a b c d e
A 0 0 1 0 1
B 0 1 0 0 0
C 1 0 1 0 1



g <- graph.incidence(termDocMatrix, mode=c("all"))
nTerms <- nrow(M)
nDocs <- ncol(M)
idx.terms <- 1:nTerms
idx.docs <- (nTerms+1):(nTerms+nDocs)
V(g)$degree <- degree(g)
V(g)$color[idx.terms] <- rgb(0, 1, 0, .5)
V(g)$size[idx.terms] <- 6
V(g)$color[idx.docs] <- rgb(1, 0, 0, .4)

```
V(g)$size[idx.docs] <- 4
V(g)$frame.color <- NA
V(g)$label <- V(g)$name
V(g)$label.color <- rgb(0, 0, 0, .5)
V(g)$label.cex <- 1.4*V(g)$degree / max(V(g)$degree)+1
E(g)$width <- .3
E(g)$color <- rgb(.5, .5, 0, .3)
plot(g, layout=layout.fruchterman.reingold)
```




```
> V(g)[nei("mining")]
Vertex sequence:
 [1] "2"   "4"   "9"   "10"  "11"  "12"  "13"  "14"  "28"  "29"  "31"  "51"  "52"  "58"  "59"
[16] "61"  "62"  "66"  "71"  "75"  "77"  "78"  "79"  "81"  "82"  "86"  "89"  "92"  "94"  "96"
[31] "99"  "102" "103" "104" "106" "109" "120" "122" "124" "133" "135" "136" "140" "142" "144"
[46] "145" "148" "149" "154" "155" "159" "165" "167" "168" "170"
```

mining관련된 모든 neighbor를 표시하는 방법이다.

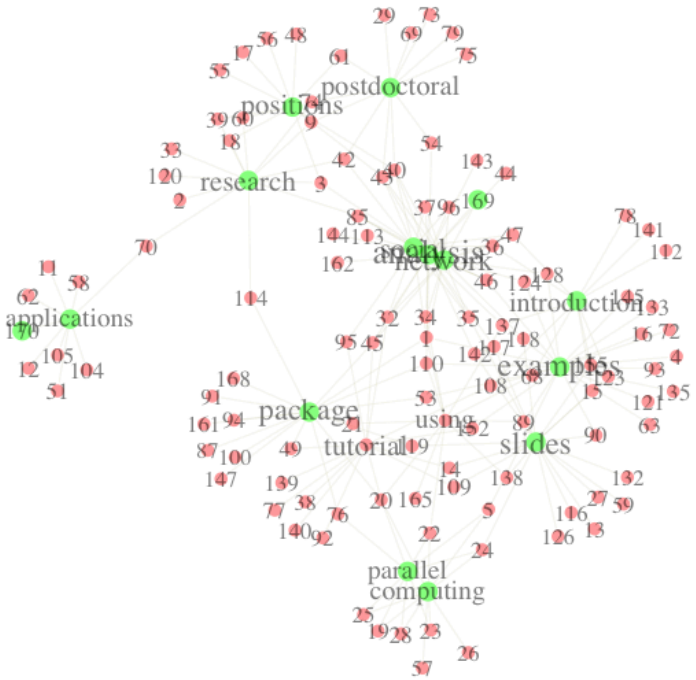
```
> (rdmVertices <- V(g)[nei("mining") & nei("data")])
Vertex sequence:
 [1] "2"   "4"   "9"   "10"  "11"  "12"  "28"  "29"  "31"  "51"  "52"  "58"  "61"  "62"  "66"
[16] "71"  "75"  "77"  "79"  "81"  "82"  "86"  "94"  "99"  "104" "106" "109" "120" "122" "133"
[31] "135" "140" "149" "154" "155" "159" "165" "167" "168" "170"
```

이 방식은 mining과 data관련 내용을 포함하는 모든 내용이다.

```
> df$text[as.numeric(rdmVertices$label)]
 [1] "Job on Artificial Intelligence and Big Data Mining at new research lab of Huawei, Hong Kong http://t.co/hOLsPo7B"
 [2] "My book in draft titled “R and Data Mining: Examples and Case Studies” is now on CRAN. Check it out at http://t.co/wOItXnHI"
```

위에서 data와 mining을 포함하는 내용을 일부 조회한 내용이다.

```
idx <- which(V(g)$name %in% c("data","mining"))
g2 <- delete.vertices(g, V(g)[idx-1])
g2 <- delete.vertices(g2,V(g2)[degree(g2)==0])
plot(g2, layout=layout.fruchterman.reingold)
```



7.3 twitter 검색을 통한 사용자간 Social Network

새로운 예제로 특정 #관련 social network을 그려보는 내용이다.[14] 원저에 일부 오류가 있어서 follower를 가져오는 내용을 수정해 보았다.

```
> search.term <- '#Rstats'
```
twitter API를 통해 검색해 올 내용을 정의한다.

```
> search.size <- 50
```
최대 검색할 내용을 제한한다.

```
> library(igraph)
> library(twitteR)
> search.results <- searchTwitter(search.term,n=search.size)
> vl <- vector()
```
초기화를 한다.
```
> for (twt in search.results)
+   vl <- c(vl,screenName(twt))
```
검색된 결과의 사용자 이름을 가져온다. 여기서도 인터넷 접속이 필요하다.

```
> vl
 [1] "DGleebits"      "zentree"        "cranatic"       "cranatic"
 [5] "gvegayon"       "almostMike"     "jingle"         "MRMacAskill"
 [9] "siah"           "rmitchell"      "gawbul"         "aadhyadi"
[13] "DataJunkie"     "Rbloggers"      "zentree"        "mesg_n"
[17] "OmerNadirler"   "revodavid"      "hadleywickham"  "brechtverduyn"
[21] "gsantosgo"      "amichalek"      "CosimoAccoto"   "amichalek"
[25] "GilPress"       "therealprotonk" "ElectricEskimo" "revodavid"
[29] "timelyportfolio" "kdnuggets"     "Rbloggers"      "andrewxhill"
[33] "neilkod"        "timelyportfolio" "zimmeee"       "Gagan_S"
[37] "tomjwebb"       "measurefuture"  "siah"           "weecology"
[41] "ethanwhite"     "rOpenSci"       "hadleywickham"  "fredbenenson"
[45] "josvandongen"   "jzb14"          "mmparker"       "RLangTip"
[49] "Biff_Bruise"    "noticiasSobreR"
> vl <- as.data.frame(table(vl))
> colnames(vl) <- c('user','tweets')
> # build the network of relations between contributors
> g <- graph.empty(directed=TRUE)
> g <- add.vertices(g,nrow(vl),name=as.character(vl$user),
+                   tweets=vl$tweets)
```
노드를 추가한다.
```
> V(g)$followers <- 0 # default to zero
> V(g)$tweets
 [1] 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1
[40] 1 2 1
> getUser('Biff_Bruise')$getFollowers(n=10)
$`76922581`
[1] "VitaNetLLC"
```

---

[14] http://www.babelgraph.org/wp/?p=120

```
$`78789634`
[1] "TwitCleaner"

$`24200201`
[1] "lamclay"

$`284477923`
[1] "iidesu_"

$`18005444`
[1] "kurtstat"

$`400782899`
[1] "carmen_tsang"

$`24985189`
[1] "Adaptv"

$`120212144`
[1] "HoagLevins"

$`466611881`
[1] "DenverListBiz1"

$`425881490`
[1] "Ash_Brisebois"
```

특정 사용자의 follower들을 가져오는 script이다.

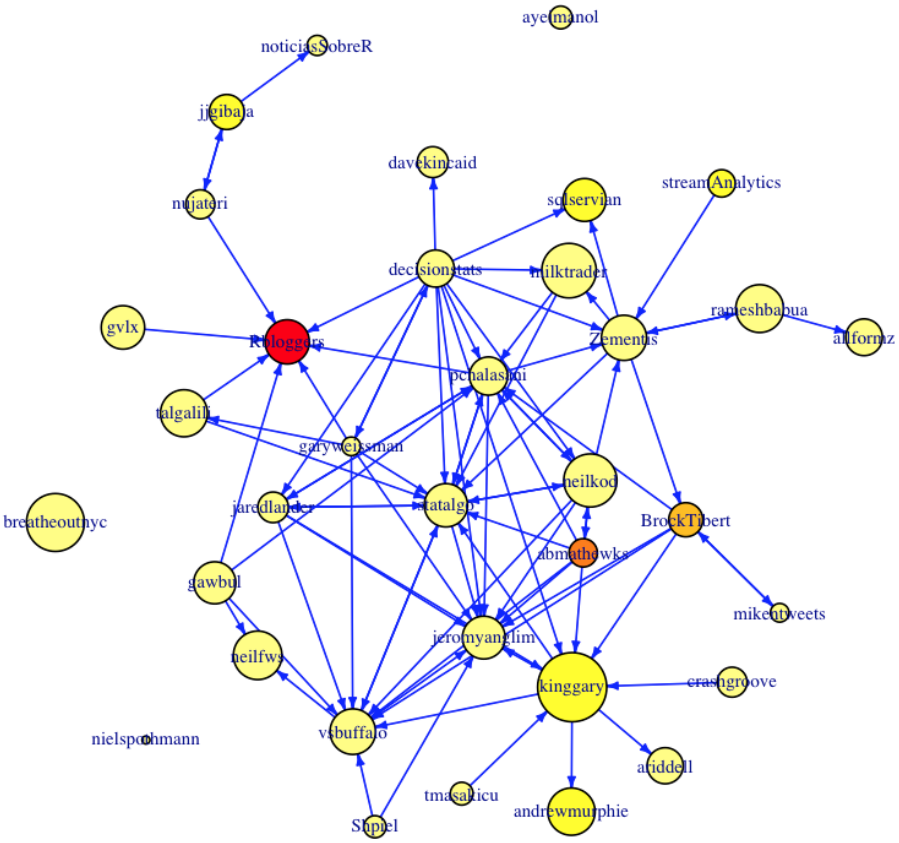
```
> # count total number of followers in the larger twitterverse and
> # add relationships based on who follows whom within the conversation
> for (usr in V(g)) {
+
+   # get the user info by name
+   tuser <- getUser(V(g)$name[usr+1])
+   print(paste("Getting info on",screenName(tuser)))
+
+   # count total followers in larger twitterverse
+   V(g)$followers[usr+1] <- followersCount(tuser)
+
+   # access as many followers as we can to see if any
+   # appeared in the search results
+   followers.list <- getUser(V(g)$name[usr+1])$getFollowers()
+   # userFollowers(tuser,n=1200)
+   for (tflwr in followers.list) {
+     if (screenName(tflwr) %in% V(g)$name)
+       g <- add.edges(g,c(as.vector(V(g)[ name == screenName(tflwr) ]),usr))
+   }
+   print('Sleeping 10 min...')
+  # Sys.sleep(600); # don't exceed request limit
+ }
```

```
[1] "Getting info on Biff_Bruise"
[1] "Sleeping 10 min..."
[1] "Getting info on CosimoAccoto"
[1] "Sleeping 10 min..."
[1] "Getting info on DGleebits"
[1] "Sleeping 10 min..."
[1] "Getting info on DataJunkie"
[1] "Sleeping 10 min..."
[1] "Getting info on ElectricEskimo"
[1] "Sleeping 10 min..."
[1] "Getting info on Gagan_S"
이하에 에러.self$twFromJSON(out) :
  Error: Rate limit exceeded. Clients may not make more than 150 requests per hour.
```

위의 소스에서 sleeping을 comment처리해서 데이터를 가져오는 twitter상의 제약에 걸려 오류가 발생하였다.

```
>
> # layout the graph
> g$layout <- layout.fruchterman.reingold(g)
> # adjust size based on number of total followers
> # scale accordingly to keep readable
> V(g)$size = log( V(g)$followers ) * 1.8
> # set label name and size
> V(g)$label=V(g)$name
> V(g)$label.cex = 0.6
> # do something with color...
> tcolors <- rev(heat.colors(max(V(g)$tweets)))
> V(g)$color <- tcolors[ V(g)$tweets ]
> # make edge arrows less intrusive
> E(g)$arrow.size <- 0.3
> # make symmetric connections a little easier to read
> E(g)$curved <- FALSE # fun to play with
> E(g)$color <- 'blue'
> # now plot...
> plot(g)
```

위의 실행에서는 오류가 발생했지만 sleeping time을 충분히 준 경우 정상적으로 종료가 된다. 성공적인 실행의 경우의 결과를 보면 다음과 같다. 아래 내용을 보면 자사 브랜드나 특정 경쟁사 관련 특정내용에 대한 고객집단이 어떻게 연결되어 있고, 각 집단의 특성이 무엇인지, 누가 influencer인지를 추가적으로 분석할 수 있다.



7.4 twitter 검색을 사용자 분포 그래프