제1절 R기초

1. 분석환경의 이해

1). 데이터 분석 도구의 현황

○ 분석도구 비교

SAS	SPSS	R
고가	고가	오픈소스
대용량	대용량	모듈화로 간단
별도 구매	별도 구매	오픈소스
느림 최고알고리즘개발방향	다소 느림	매우 빠름
유료 도서 위주	유료 도서 위주	공대 논문 및 자료 많음
없 커뮤니티	없	매우 활발

○ R의 특징

- 오픈소스, 무료 : 사용자 커뮤니티 활발, 5000여개 패키지 수시 업데이트
- 그래픽 및 성능이 좋다
- 시스템 데이터 저장방식 : 각 세션 사이마다 시스템에 데이터셋 저장,
  명령어 히스토리 저장 가능
- 모든 운영체제 사용 가능
- S통계 언어 기반(R/S 플랫폼은 전문가들의 표준 플랫폼)
- 객체지향언어이며 함수형 언어

* 꼭 나오는 문제

○ R 스튜디오

- 오픈소스, 무료, 다양한 운영체제 지원
- 메모리에 변수와 타입 확인 가능
- 스크립트 관리와 도큐먼테이션 편리
- 쉽게 자동화

2. 기본 사용법
R기초 포스팅 목록

http://blog.naver.com/liberty264/220984298909

- ctrl+F를 통해 알고싶은 함수 검색해서 보세요
 * 벡터의 입력을 보고 에러가 생성되는 것 찾기
 * 구조 중에서는 데이터프레임에 관한 문제

제2절 데이터 마트

1. R reshape를 활용한 데이터 마트 개발

○ 데이터 마트

- 데이터웨어하우스와 사용자 사이 중간층에 위치
  DW로부터 복제된 data, 자체 수집도 가능 (관계형, 다차원데이터베이스로 구축)
- CRM업무 핵심 : 고객 데이터 마트 구축
- 데이터 마트 구축 방식에 따라 분석 효과 차이

○ 요약변수

- 수집된 정보를 분석에 맞게 종합한 변수
- 데이터마트에서 가장 기본적인 변수(기간·상품별 구매 금액, 횟수, 구매여부 등)
- 재활용성 높음
- 간단한 구조이므로 자동화 가능

○ 파생변수

- 사용자가 만들어 의미를 부여한 변수
- 주관적이므로 논리적 타당성 필요
- 대표성을 나타나게 할 필요
- 근무시간 구매지수, 주 활동 지역 변수, 선호하는 가격대 변수 등

○ reshape 활용

- melt() : 데이터를 DB구조로 녹이는 함수 -> 원데이터 형태로 만드는 함수
- cast() : 새로운 구조로 데이터를 만드는 함수 -> 요약 형태로 만드는 함수

- melt(data, id=c(고정변수1, 고정변수2))
- cast(melt_data, 고정변수1+고정변수2 ~ column이 되게하고 싶은 변수명, 함수)

- 다양한 요약변수와 파생변수를 쉽게 생성하여 데이터마트 구성

```
library(reshape2)

> data("airquality")
> names(airquality) <- tolower(names(airquality)
> head(airquality)

  ozone solar.r wind temp month day
1   41    190  7.4   67     5   1
2   36    118  8.0   72     5   2
3   12    149 12.6   74     5   3
4   18    313 11.5   62     5   4
5   NA     NA 14.3   56     5   5
6   28     NA 14.9   66     5   6

#melt()

aql <-melt(airquality,id.vars = c("month","day"))
head(aql)

  month day variable value
1     5   1    ozone    41
2     5   2    ozone    36
3     5   3    ozone    12
4     5   4    ozone    18
5     5   5    ozone    NA
6     5   6    ozone    28

#cast()

dcast(aql , month ~ variable )
#dcast는 cast를 d ( dataframe ) 으로 반환하겠다는 뜻 reshape2패키지에만 있음
```

```
  month ozone solar.r wind temp
1     5    31      31   31   31
2     6    30      30   30   30
3     7    31      31   31   31
4     8    31      31   31   31
5     9    30      30   30   30
```

식별자가 하나일 때 자동으로 length()를 적용해 같은셀에 모인 행의 개수를 센다.

dcast(sql,month ~ variable, mean , na.rm = TRUE) # sum , mean, range 구할 수 있다.

```
    month    ozone   solar.r      wind       temp
1     5 23.    61538  181.2963  11.622581  65.54839
2     6 29.    44444  190.1667  10.266667  79.10000
3     7 59.    11538  216.4839  8.941935   83.90323
4     8 59.    96154  171.8571  8.793548   83.96774
5     9 31.    44828  167.4333  10.180000  76.90000
```

2. sqldf를 이용한 데이터 분석

○ R에서 sql 명령어를 사용가능하게 해주는 패키지(SAS의 proc sql)

- **head**(df) = sqldf("select * from df **limit 6**")
- **subset**(df, col **%in%** c("BF","HF") = sqldf(" select * from df **where** col **in** ('BF','HR')")
- **merge**(df1, df2) : sqldf("select * **from df1, df2**")
- **rbind** : **union** all select * from df2
- **order (decreasing=T) : order by ... desc**

3. plyr

○ 데이터와 출력변수를 동시에 배열로 치환하여 처리하는 패키지

○ 분리-처리-결합 방식 (split, apply, combine 함수)

○ multicore를 사용해 반복문 없이도 간단하고 빠르게 처리

	array	Data frame	list	nothing
array	apply	adply	alply	a_ply
Data frame	daply	ddply	dlply	d_ply
list	laply	ldply	llply	l_ply
n replicates	raply	rdply	rlply	r_ply
Function arguments	maply	mdply	mlply	m_ply

adply(iris,1,function(row){**row$Sepal.Length>=5.0&row$Species=="setosa"**})

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	V1
1	5.1	3.5	1.4	0.2	setosa	TRUE
2	4.9	3.0	1.4	0.2	setosa	FALSE
3	4.7	3.2	1.3	0.2	setosa	FALSE

adply함수 **데이터프레임으로 결과 출력**하고, species=setosa 확인하고 그 결과를 v1나타냄. 즉 데이터프레임 다른 데이터 타입이 가능하여 문자형으로 표시가 안됨

4. 데이터 테이블

○ 큰 데이터를 탐색, 연산, 병합하는데 유용한 데이터 핸들링 패키지
○ 기존 data.frame보다 빠르다
○ 특정 column을 key값으로 색인 지정 가능
○ 빠른 그룹핑, ordering, 짧은 문장 지원

○ setkey(DT, y); x<-DT[J("C"),]

제3절 데이터 가공

1. 데이터 탐색 data exploration

○ 데이터 탐색

- 변수 상태 파악 : head(), summary()

수치형변수 : 최대값, 최소값, 평균, 등...
명목형변수 : 명목값, 데이터개수

변수의 중요도

모형을 생성하여 사용된 변수의 중요도를 정리

- 변수 선택법과 유사한 개념
- **klaR 패키지**는 특정 변수가 주어졌을 때 클래스가 어떻게 분류되는지에 대한 에러율을 계산해주고, 그래픽으로 결과를 보여주는 기능을 함
- **greedy.wilks( )**는 종속변수에 가장 영향력을 미치는 변수에 가장 영향력을 미치는 변수를 **wilks lambda**를 활용해 세분화를 위한 **stepwise forward** 변수를 선택함으로써 변수의 중요도를 정리

→ 전진선택법 (가장영향력있는것부터선택)

※ Wilk's Lambda = 집단 내 분산 / 총 분산
ㄴ> 값이 작을 수록 설명력이 높다

변수의 구간화

분석 목적에 활용하기 위해 연속형 변수를 구간화

- 일반적으로 10진수 단위로 구간화
- 보통 구간을 5개로 나누며 7개 이상의 구간은 잘 만들지 않음
- 신용평가모델, 고객 세분화와 같은 시스템에서 모형에 활용하는 각 변수들을 구간화해서 구간별 점수를 산정 -> **스코어링방식**

*** 변수의 구간화 방법**

① binning

- 신용평가모형의 개발에서 연속형 변수를 범주형 변수로 구간화하는데 자주 활용되는 방법
- bin은 '쓰레기통'이라는 뜻으로 연속형 변수를 정렬한 후 **각각의 bin에 나눠 담아 범주형 변수로 구간화**

② 의사결정나무

- 세분화 또는 예측에 활용되는 의사결정나무 모형을 사용해 입력변수들을 구간화
- 의사결정나무를 사용하면 **동일한 변수를 여러 개의 분리기준으로 사용 가능**
- 연속변수가 반복적으로 선택될 경우, 각각의 분리 기준값으로 연속형 변수를 구간화 가능

제4절 결측값 처리와 이상값 검색

1. 데이터 EDA (탐색적 자료분석)

분석에 앞서 전체적인 데이터의 특징 파악 , 데이터를 다양한 각도로 접근
summary 를 이용하여 데이터의 기초통계량 확인

2. 결측값 처리

○ 표현 : NA
○ 자체가 의미있을 수 있다
- 가입자 중 특정 거래 없을 경우, 부정사용방지 시스템이나 부도예측시스템)
○ 결측값 처리가 전체 작업속도에 영향
○ 결측치 20% 이상인 변수는 제거하는 것이 바람직
○ 관측치가 있는 것은 default로 기록되었더라도 결측치로 처리하면 안됨

○ 결측값 처리 방법

- 단순 대치법

· complete analysis - 결측값 존재하는 행 삭제

· 평균 대치법 - 비조건부 평균대치법 : 관측데이터 평균으로

- 조건부 평균대치법 : 회귀분석을 통해 유추하여 대치

· 단순확률 대치법 - 추정량 표준오차의 과소 추정문제 보완

Hot deck, nearest nieghbor 방법 등

- 다중 대치법

· 단순 대치법을 여러번 / 대치 -> 분석 -> 결합 ㅍ

· boostrapping based algorithm

○ R에서 결측값 처리

결측값 있으면 FALSE

- 탐색 : complete.cases(), is.na() → 결측값 있으면 TRUE

- 단순 대치 : DMwR::centralImputation(), DMwR::knnImputation()

중위수, 최빈값 k개 주변이웃의 가중평균

- 다중 대치 : Amelia::amelia()

-> 랜덤포레스트 모델에서 사용 / rmImpute() 함수로 NA결측값 대치

3. 이상값 검색

○ 이상값

- bad data : 잘못 입력, 분석 목적에 부합하지 않는 경우 삭제

- **이상값** : 의도하지 않은 현상으로 입력, 의도된 극단값 경우 활용

○ 이상값 인식 방법

주관식

- **ESD : 평균으로부터 3 표준편차 떨어진 값**

- 기하평균보다 2.5 표준편차 떨어진 값

- 1사분위와 3사분위 사이 범위보다 2.5배이상 떨어진 값

○ 이상값 처리

- 절단 : 이상값 포함 행 삭제

(기하평균 이용, 하단 상단 5%씩 총 10% 제거)

- 조정 : 이상값을 상한 또는 하한값으로 조정(데이터 손실률이 낮아진다)

* 의미, 인식방법, 처리의 전반적인 내용을 묻는 문제