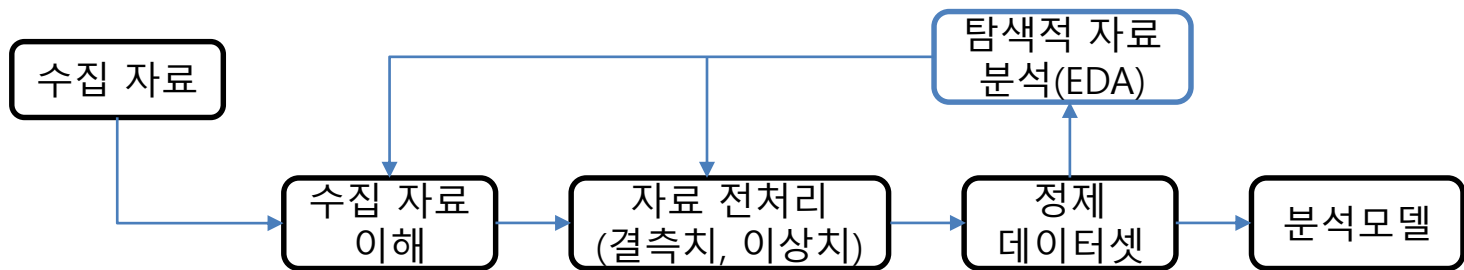


EDA & 데이터 정제

EDA란?

- 탐색적 자료 분석(Exploratory Data Analysis)으로 해석.
- 수집한 자료를 다양한 각도에서 관찰하고 이해하는 과정으로 그래프나 통계적 방법을 이용해서 자료를 직관적으로 파악하는 과정.
- EDA의 필요성
 - 자료의 분포와 통계를 파악하여 자료가 갖고 있는 특성을 이해하고, 잠재적인 문제 발견해서
 - 분석 전에 발견이 어려운 다양한 문제점을 발견하고
 - 이를 바탕으로 기존의 가설을 수정하거나 새로운 방향의 가설을 세울 수 있음



EDA 과정

- 단계1 : 분석의 목적과 변수의 특징 확인
 - 수집된 자료를 대상으로 셀 수 없는 변수(Categorical)와 셀 수 있는 변수(Numerical)를 구분한다.
- 단계2 : 자료 확인 및 전처리
 - 수집된 자료를 확인하고 결측치와 이상치를 확인하고 정제한다.
- 자료의 각 변수 관찰
 - 대표값이나 산포도를 이용한 통계 조사 및 이산 변수와 연속 변수 시각화
- 변수 간의 관계에 초점을 맞춰 패턴 발견
 - 변수와 변수 간의 상관 관계나 고급 시각화 도구를 이용한 변수 간의 패턴 발견

수집 자료 이해

1. 탐색적 데이터 분석

1) 데이터 셋 보기

`print(dataset)`

`View(dataset)` # 별도의 데이터 뷰어창에서 출력 됨, 컬럼 정렬

#데이터의 앞쪽/뒤쪽 조회

`head(dataset)` # 앞부분 데이터 셋 6개

`tail(dataset)` # 끝부분 데이터 셋 6개

`head(dataset, 10)` # 앞부분 10개

Data Mart(dataset.csv)

# resident	gender	job	age	position	price	survey
# 명목	명목	명목	비율	서열	비율	등간
# 1~5	1, 2	1~3	20~69	1~5	2.1~7.9	1~5
# 거주지	성별	직업	나이	직위	구매금액	만족도

수집 자료 이해

2) 데이터 셋 구조보기

#dataset에 들어 있는 세부 정보항목 조회

names(dataset) # 변수명(컬럼)

attributes(dataset) # names(열이름), class, row.names(행이름)

str(dataset) # 데이터 구조보기

수집 자료 이해

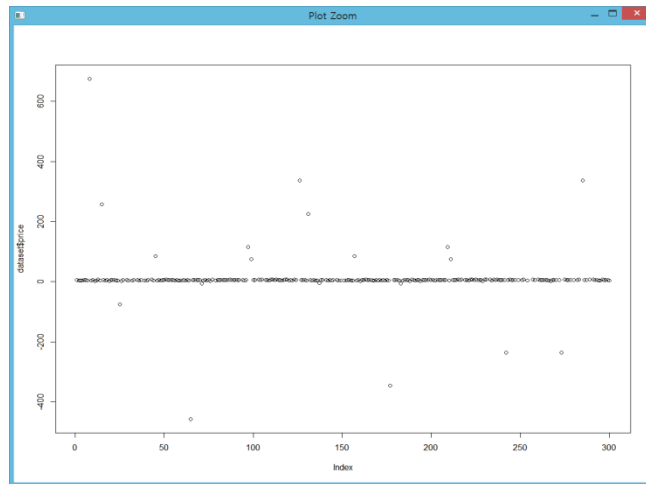
3) 데이터 셋 조회

```
#dataset 데이터 중 특정변수 조회  
dataset$age # [1] [27] [148] age값의 색인  
dataset$resident  
length(dataset$age) # data 수-300개
```

```
x <- dataset$gender # 결과를 변수에 저장  
y <- dataset$price  
x;y
```

```
plot(dataset$price) # 산점도 형태 전반적인 가격분포 보기
```

```
# $기호 대신 [""]기호를 이용한 특정변수 조회  
dataset["gender"]  
dataset["price"]
```



수집 자료 이해

색인(index)으로 변수의 위치값 조회

dataset[2] #두번째 컬럼(gender)

dataset[6] #여섯번째 컬럼(price)

dataset[3,] #3번째 관찰치(행) 전체 -> 열 공통

dataset[,3] # 전체행의 3번째 변수(열) -> 행 공통

수집 자료 이해

dataset 데이터 중 변수를 2개 이상 조회하는 경우

dataset[c("job","price")]

dataset[c(2,6)] # gender, price

dataset[c(1,2,3)] #resident, gender, age

dataset[c(1:3)] #resident, gender, age

dataset[c(2,4:6,3,1)] #gender, age, position, price, job, resident

dataset 데이터 중 특정 행/열을 지정해 조회

dataset[,c(2:4)] #2~4열(gender job age) 전체 -> test[c(2:4)]과 동일

dataset[c(2:4),] #2~4행 전체

dataset[-c(1:100),] # 1~100행 제외

결측치(NA) 처리

2. 결측치(NA) 처리

```
summary(dataset$price) # 결측치 확인 -> NA's - 30개  
sum(dataset$price) # NA 출력
```

결측데이터 제거 방법1

```
sum(dataset$price, na.rm=T) # 2362.9
```

결측데이터 제거 방법2

```
price2 <- na.omit(dataset$price) # price에 있는 모든 NA 제거  
sum(price2) # 2362.9  
length(price2) # 270 -> 30개 제거
```

극단치(이상치) 처리

3. 극단치 발견과 정제

1) 범주형 변수 극단치 처리

gender 변수 outlier 확인

gender <- dataset\$gender

hist(gender) # 히스토그램으로 outlier 확인

table(gender) # 빈도수로 outlier 확인

pie(table(gender)) # 파이 차트로 outlier 확인

극단치(이상치) 처리

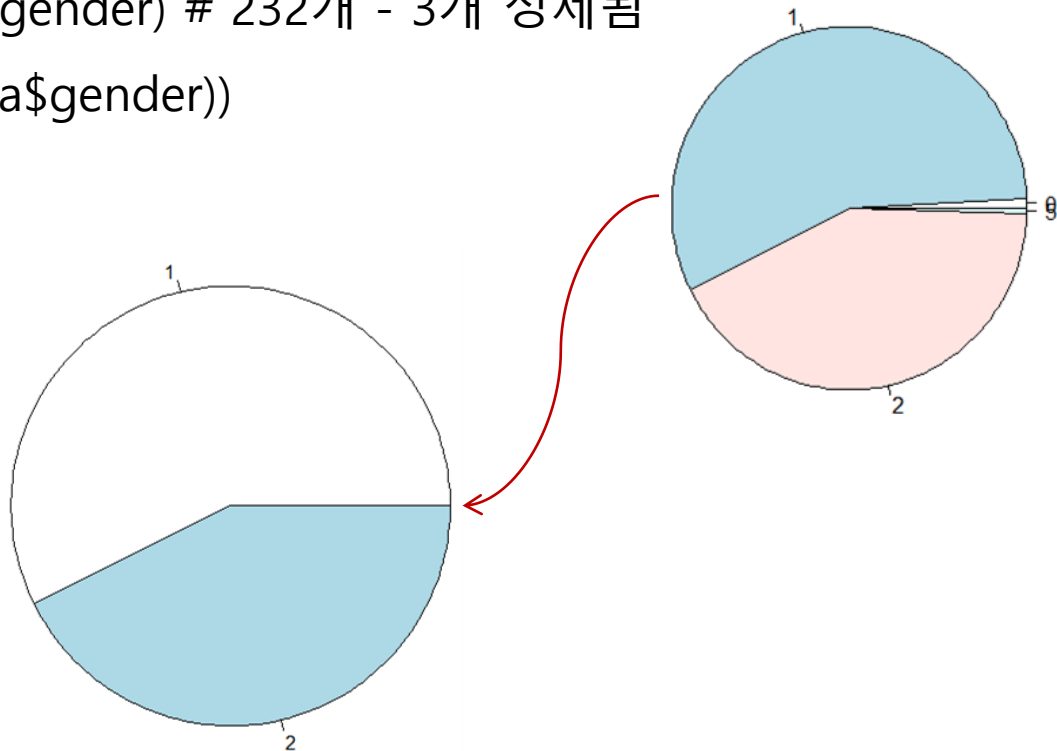
- 성별 데이터 정제 - subset() 함수 이용

```
data <- subset(data, data$gender == 1 | data$gender == 2)
```

```
data # gender변수 데이터 정제
```

```
length(data$gender) # 232개 - 3개 정제됨
```

```
pie(table(data$gender))
```



극단치(이상치) 처리

2) 연속형 변수 극단치 처리

price outlier 확인

dataset\$price # 세부 데이터 보기

length(dataset\$price) #300개(NA포함)

plot(dataset\$price) # 산점도 형태 전반적인 가격분포 보기

summary(dataset\$price) # -457~675 범위확인

극단치(이상치) 처리

```
# price변수 정제(2~8)
```

```
data <- subset(dataset, dataset$price >= 2 & dataset$price <= 8)
```

```
length(data$price) #251개(49개 정제)
```

```
stem(data$price) # 줄기와 잎 도표보기
```

```
The decimal point is at the |
```

```
2 | 133
2 |
3 | 000003344
3 | 555589
4 | 0000000000000001111111122233333444
4 | 5666667777889999
5 | 000000000000000000111111111222222223333344444
5 | 5555555556677777788899
6 | 00000000001111112222222222233333333333333334444444
6 | 5555777777788889999
7 | 000111122
7 | 7799
```

극단치(이상치) 처리

age 변수 NA 발견

```
summary(data$age) # Min(20), Max(69), NA(16)
```

```
length(data$age) # 251
```

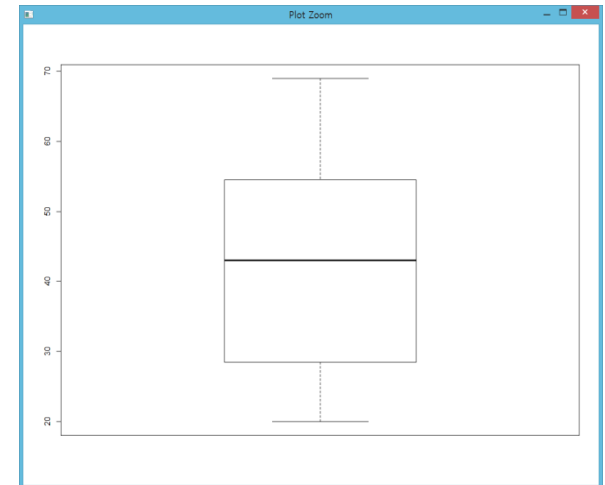
age 변수 정제(20~69)

```
data <- subset(data, data$age >= 20 & data$age <= 69)
```

```
length(data$age) # 235개(16 정제)
```

box 플로팅으로 평균연령 분석

```
boxplot(data$age) # 45대 중반 평균 연령
```



코딩 변경

4. 코딩변경 - 변수변환 : 리코딩 하기

- 데이터의 가독성, 척도 변경, 최초 코딩 내용 변경을 목적으로 수행

문자열로 리코딩(청년층, 중년층, 장년층)

```
data$age2[data$age <= 30] <-"청년층"
```

```
data$age2[data$age > 30 & data$age <=45] <-"중년층"
```

```
data$age2[data$age > 45] <-"장년층"
```

```
head(data) # data 테이블 전체 - age와 age2 비교
```

```
head(data[c("age","age2")]) # 2개만 지정
```

```
# age age2
```

```
# 26 청년층
```

```
head(data) # dataset 테이블 전체 - age2 컬럼 생성
```

```
head(data[c("age","age2")]) # 2개만 지정
```

```
# age age2 age3
```

```
# 26 청년층 1
```

코딩 변경

4. 코딩변경 : 역코딩 : 긍정순서(1 -> 5, 5 -> 1)

```
data$survey
```

```
survey <- data$survey
```

```
csurvey <- 6-survey
```

```
csurvey
```

```
survey # 역코딩 결과와 비교
```

```
data$survey <- csurvey # data set에 survey 변수 수정
```

```
head(data) # survey 결과 확인
```

	resident	gender	job	age	position	price	survey
1	1	1	1	26	2	5.1	1
2	2	1	2	54	5	4.2	2
3	NA	1	2	41	4	4.7	4
4	4	2	NA	45	4	3.5	2
5	5	1	3	62	5	5.0	1
6	3	1	2	57	NA	5.4	2

정제된 데이터 저장

5. 정제된 데이터 저장

```
setwd("C:/workspaces/Rwork/data")
```

1) 정제된 데이터 저장

```
write.csv(dataset2,"cleanData.csv", quote=F, row.names=F)
```

2) 저장된 파일 불러오기/확인

```
new_data <- read.csv("cleanData.csv", header=TRUE)
```

```
new_data
```

```
dim(new_data) # 248 13
```

탐색적 분석을 위한 시각화

6. 탐색적 분석을 위한 시각화

- 1) 명목척도(범주/서열) : 명목척도(범주/서열) - 거주지역 : 성별
- 거주지역과 성별 칼럼 시각화

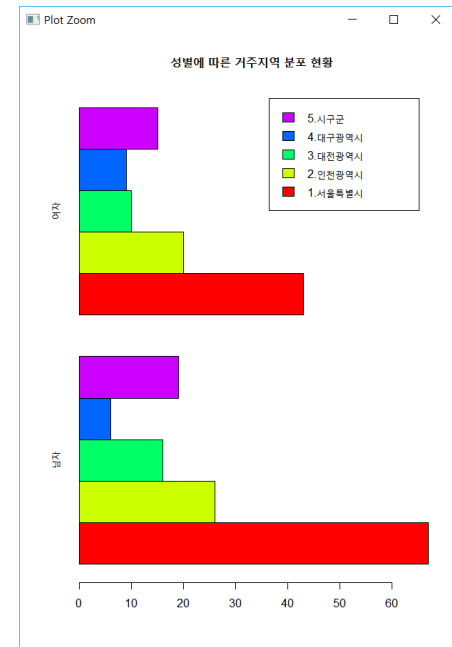
```
resident_gender <- table(new_data$resident2, new_data$gender2)
```

```
gender_resident <- table(new_data$gender2, new_data$resident2)
```

성별에 따른 거주지역 분포 현황

```
barplot(resident_gender, beside=T, horiz=T,  
        col = rainbow(5),  
        legend = row.names(resident_gender),  
        main = '성별에 따른 거주지역 분포 현황')
```

row.names(resident_gender) # 행 이름



탐색적 분석을 위한 시각화

2) 비율척도(연속) : 명목척도(범주/서열)

- 나이와 직업유형에 따른 시각화

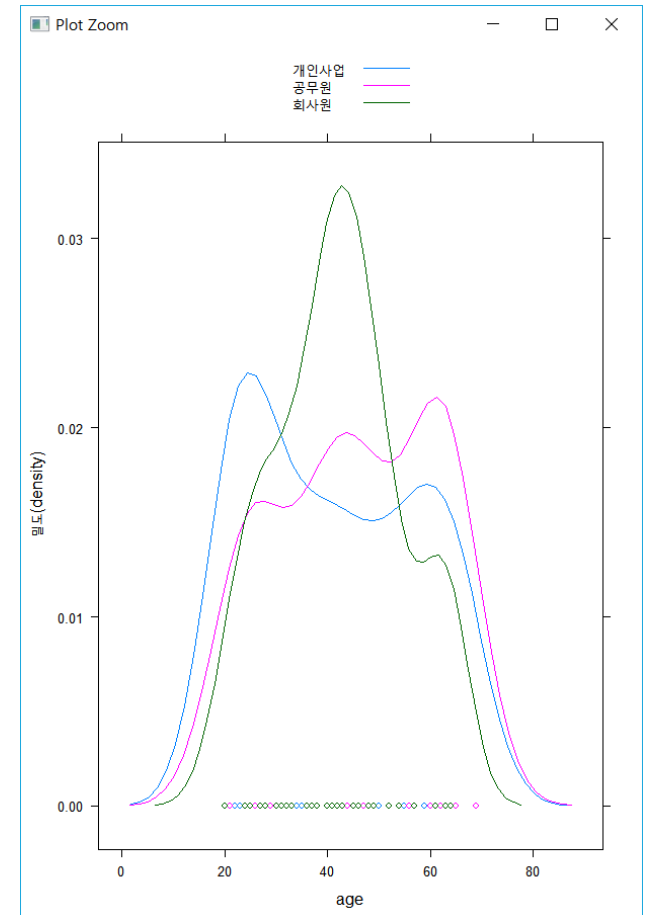
```
install.packages("lattice")
```

```
library(lattice)
```

```
# 나이와 직업유형 데이터 분포
```

```
densityplot( ~ age, data=new_data, groups = job2,  
            plot.points=T, auto.key = T)
```

```
# plot.points=T : 밀도, auto.key = T : 범례
```

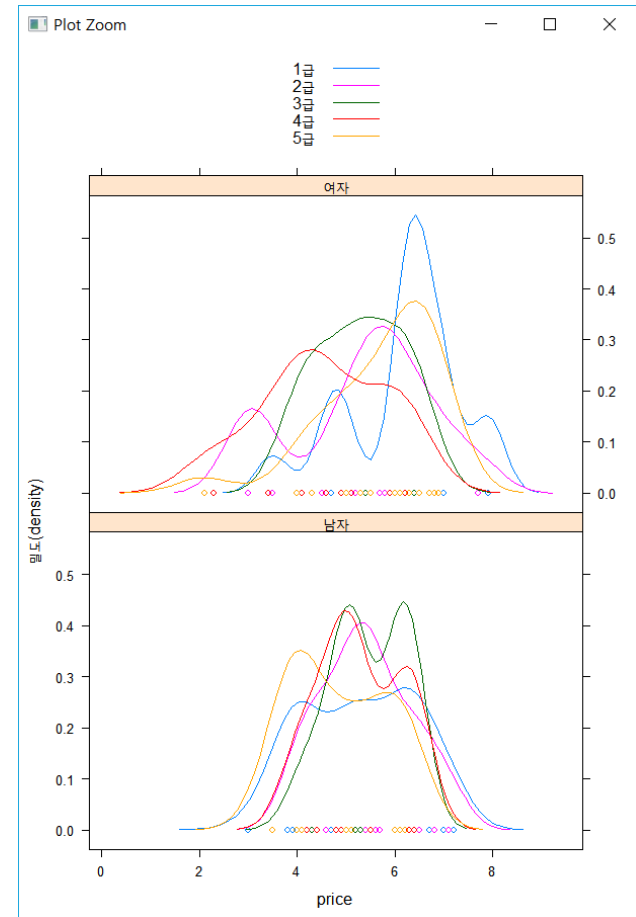


탐색적 분석을 위한 시각화

3) 비율(연속), 명목(범주/서열) : 명목(범주/서열)

- 구매비용(연속) : x칼럼 , 직급(서열):조건, 성별(범주):그룹

```
densityplot(~ price | factor(gender2), data=new_data,  
            groups = position2, plot.points=T, auto.key = T)  
# groups = gender2 : 한 격자에 나타날 그룹 칼럼 지정
```



파생변수 생성

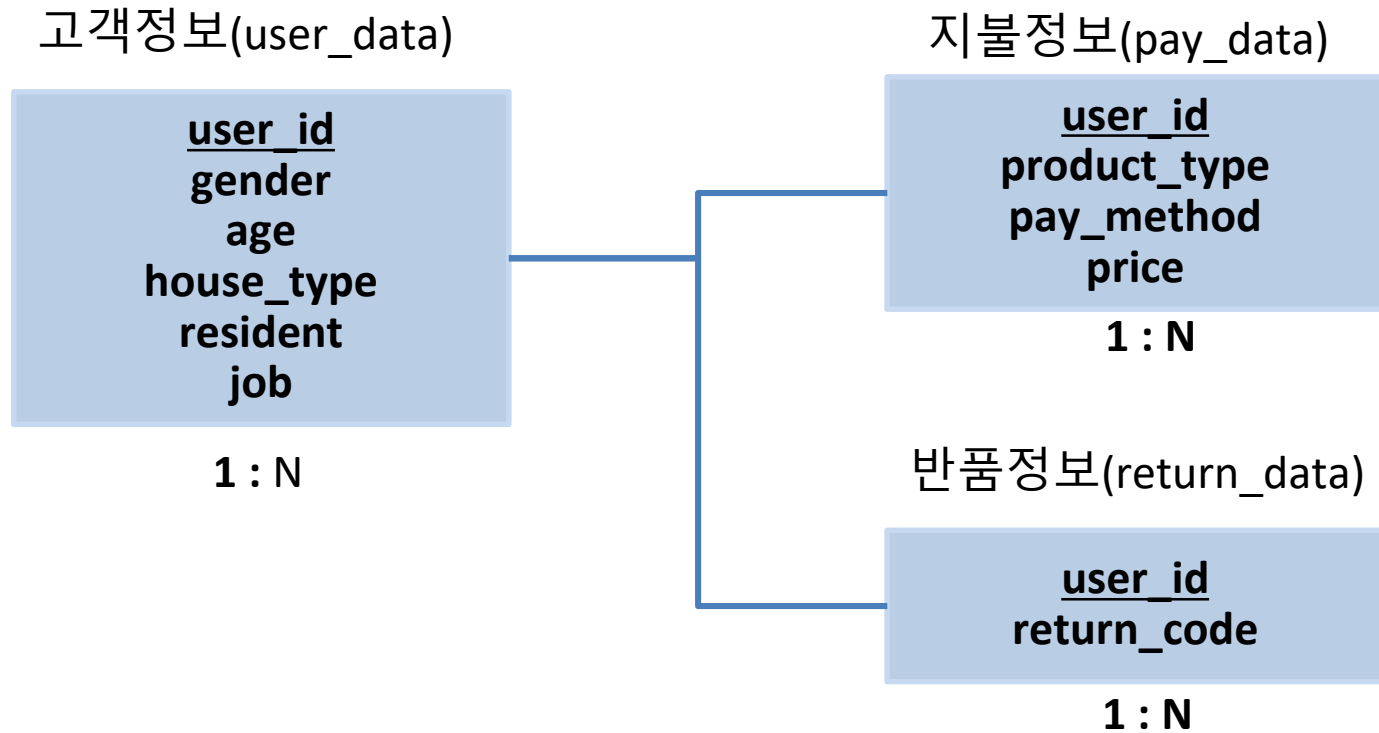
7. 파생변수 생성

- 기존 변수로 새로운 변수 생성
- 방법) 사칙연산 적용, 1:N 관계 -> 1:1 관계 변수 생성

id : 주거환경(주택,빌라,아파트,오피스텔) -> 4개의 칼럼 추가

파생변수 생성

- 파생변수에 생성을 위한 테이블 구조



파생변수 생성

➤ 고객정보(user_data)

	A	B	C	D	E
1	No	변수영문명	변수타입	변수명	변수 설명
2	1	user_id	N	고객ID	고객 식별번호
3	2	gender	N	성별	남성 : 1, 여성 : 2
4	3	age	N	연령	고객 연령
5	4	house_type	N	거주 Type	고객 거주지 유형 : 단독주택(1), 다가구주택(2), 아파트(3), 오피스텔(4)
6	5	resident	C	고객 거주 지역	고객 거주 지역(시/도)
7	6	job	N	직업유Type	총 4개 직업군으로 분류 : 자영업(1),사무직(2),서비스(3),전문직(4),서비스(5),기타(6)
8					

파생변수 생성

➤ 지불정보(pay_data)

	A	B	C	D	E
1	No	변수영문명	변수타입	변수명	변수 설명
2	1	user_id	N	고객ID	고객 식별번호
3	2	product_type	N	상품타입	고객이 구매한 상품 유형 : 식료품(1), 생필품(2), 의류(3), 잡화(4), 기타(5)
4	3	pay_method	C	지불방법	상품구매 지불 방법 : 1.현금, 2.직불카드, 3.신용카드, 4.상품권
5	4	price	N	구매금액	고객이 구매한 상품 구매 금액

➤ 반품정보(return_data)

	A	B	C	D	E
1	No	변수영문명	변수타입	변수명	변수 설명
2	1	user_id	N	고객ID	고객을 구분하는 고유번호
3	2	return_code	N	반품사유코드	상품에 대한 반품의 원인이 되는 사유코드 - 제품이상(1), 변심(2), 원인불명(3), 기타(4)

표본 추출

8. 표본 추출

1) 정제된 데이터 파일 저장(cleanData.csv)

```
print(data) # 정제 데이터 확인
```

```
getwd() # 작업 디렉터리 확인
```

```
setwd("c:/workspaces/Rwork/data") # 저장 디렉터리 지정
```

```
# 따옴표와 행 이름 제거하여 저장
```

```
write.csv(data,"cleanData.csv", quote=F, row.names=F)
```

```
# 저장된 파일 불러오기/확인
```

```
data <- read.csv("cleanData.csv", header=TRUE)
```

```
data # 저장된 파일 불러오기/확인
```

```
length(data$age)# 235개 정제 데이터 확인
```

표본 추출

2) 정제 데이터 대상 샘플링하기

nrow(data) # 235개 : 행수 구하기 -> Number of Rows

235개 중 30개 무작위 추출
choice1 <- sample(nrow(data), 30)
choice1 # 추출된 행 번호 출력

50~235 사이에서 30개 무작위 추출
choice2 <- sample(50:nrow(data), 30)
choice2

50~100 사이에서 30개 무작위 추출
choice3 <- sample(c(50:100), 30)
choice3

Sampling 결과는 데이터 셋에서 선택된 레코드 번호를 의미