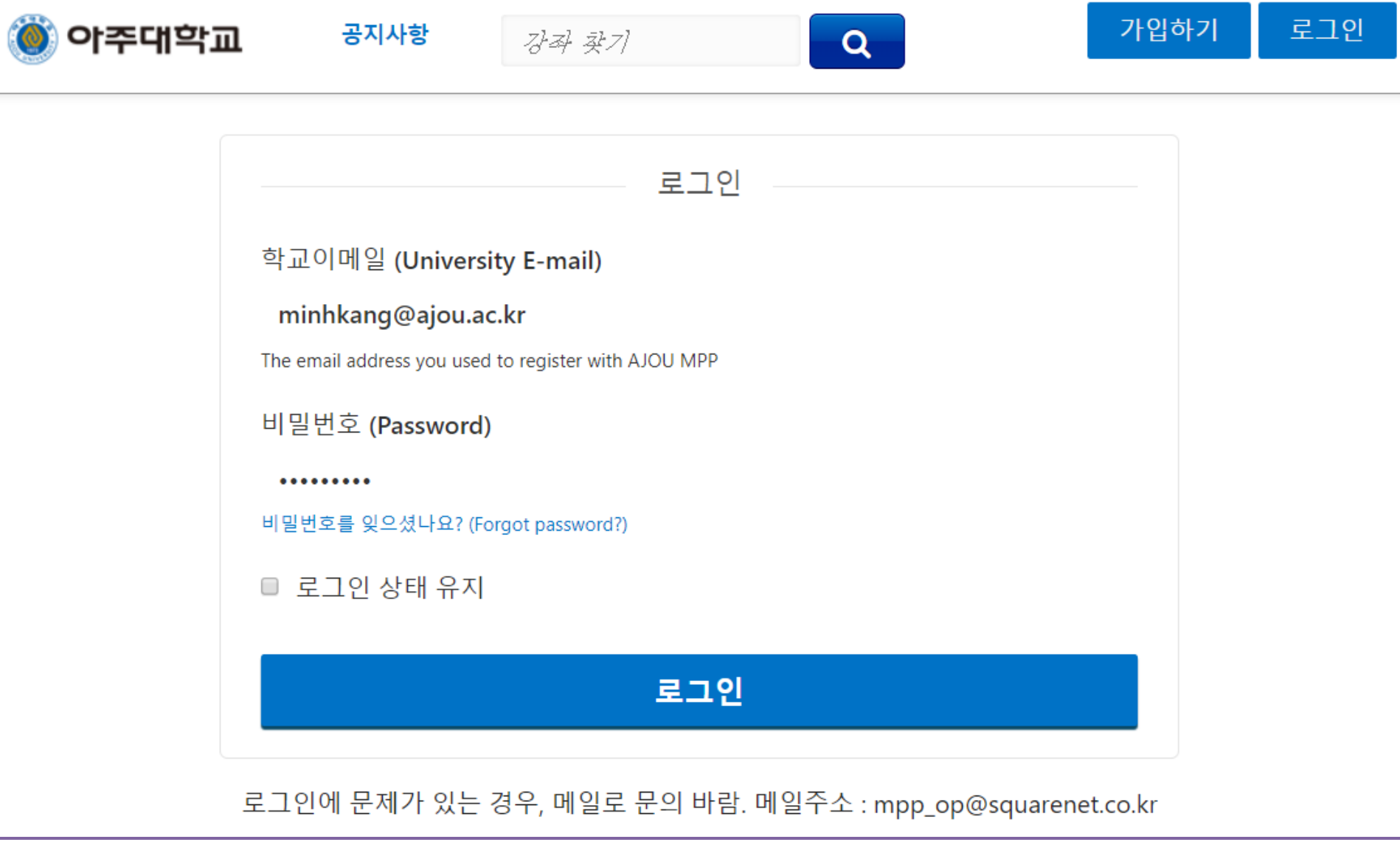
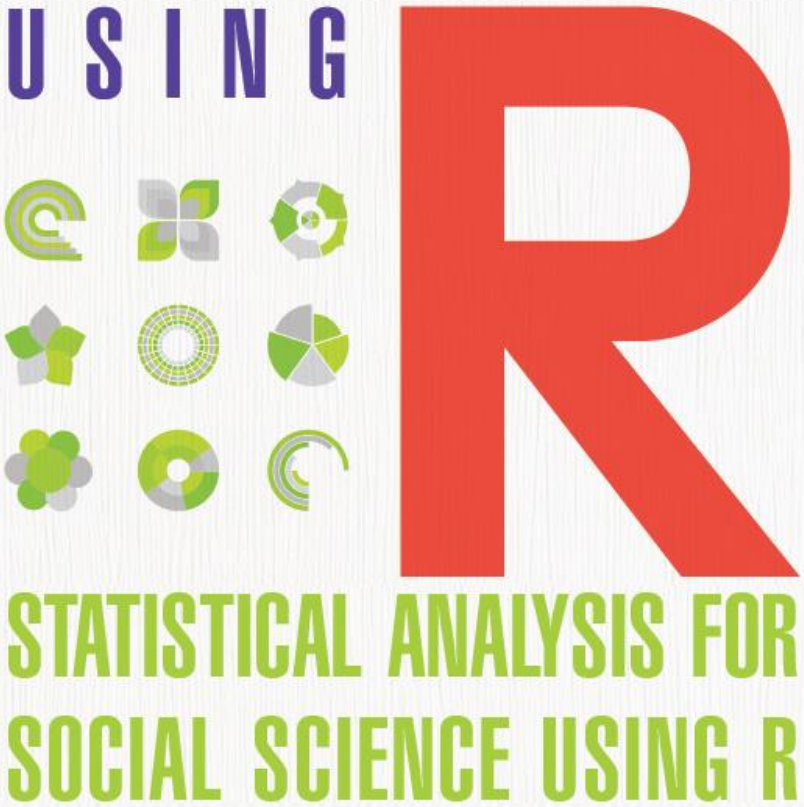
* http://mpp.ajou.ac.kr



2장 예제데이터 만들기와 데이터 변환



1. 예제데이터 만들기
2. 예제데이터에 대한 설명
3. 통계분석을 위한 데이터 변환

1. 예제데이터 만들기

1. SPSS 데이터 파일을 이용한 데이터 만들기

- SPSS 데이터 파일 불러오기
  - ① 'Hmisc' 패키지를 설치하고, 불러온다
  - ② 'Hmisc' 패키지에서 SPSS 데이터 파일을 불러오기 위해 spss.get 함수를 이용한다
  - ③ spss.get 함수를 통해 불러온 SPSS 데이터 파일을 새로운 객체로 저장한다.

예제 데이터 다운로드 경로
- 한국청소년정책연구원 데이터아카이브 → NYPI 패널조사 → 한국 청소년패널조사 → 조사표/데이터/코드북 → 중2 패널 데이터 SPSS → [중2패널] 1차년도 ~ 6차년도 데이터(SPSS).zip 다운로드한다.
- 다운 받은 SPSS 데이터 파일의 압축을 푼 다음, 사용할 데이터의 이름은 "04-1 중2 패널 1차년도 데이터(SPSS).sav"이다.

NYPI PANEL SURVEY
NYPI 패널조사

- NYPI 패널조사 소개
- 한국청소년패널조사
- 한국아동·청소년패널조사
- 다문화청소년패널조사

한국청소년패널조사

NYPI 패널조사 입니다.

KYPS 개요 | 조사표/데이터/코드북 | 학술대회 자료집 | 데이터활용 논문/보고서

전체 **12건**, 현재 페이지 1/1

번호	내용	파일	게시일	조회
공지	데이터활용 유의사항(필독)	작성자 시스템관리자	2010-05-12	6904
12	[중 2패널 코드북]		2010-05-12	9789
11	[중2패널 설문지]		2010-05-12	11482
10	[초4패널 코드북]		2010-05-12	3381
9	[초4패널 설문지]		2010-05-12	4616
8	[중2패널 데이터 STATA]		2010-05-12	2771
7	[중2패널 데이터 SPSS]		2010-05-12	8404

1. 예제데이터 만들기

```
# SPSS 데이터 파일 변환하기
install.packages("Hmisc")
library(Hmisc)
test <- spss.get("(파일 경로)/04-1 중2 패널 1차년도 데이터(SPSS).sav",
       use.value.labels=FALSE)
```

- 스크립트 설명
  - ✓ '#' 표시로 시작하면 주석문
  - ✓ use.value.labels 인자를 이용하여 불러온 데이터의 변수값 (variable value)을 변수값 설명(variable value label)으로 대체할지 여부를 결정 (TRUE/FALSE)
    - ex. Replace 1 with "CEO", 2 with "CIO", 3 with "CTO", ...

1. 예제데이터 만들기

- 변수 선택과 사례 선택
  - ✓ 변수 선택
    - **변수명** 혹은 **변수의 위치를** 사용해서 변수를 선택

```
# 변수 선택
# ① 문자형 벡터 만들기
> select_variables <- c("id", "sexw1", "scharew1", "areaw1",
   "q2w1", "q18a1w1", "q18a2w1", "q18a3w1", "q33a01w1",
   "q33a02w1", "q33a03w1", "q33a04w1", "q33a05w1", "q33a06w1",
   "q33a07w1", "q33a08w1", "q33a09w1", "q33a10w1", "q33a12w1",
   "q33a13w1", "q33a14w1", "q33a15w1", "q34a1w1", "q34a2w1",
   "q34a3w1", "q34a4w1", "q34a5w1", "q34a6w1", "q37a01w1",
   "q37a02w1", "q37a03w1", "q37a04w1", "q48a01w1", "q48a02w1",
   "q48a03w1", "q48a04w1", "q48a05w1", "q48a06w1", "q48b1w1",
   "q48b2w1", "q48b3w1", "q48c1w1", "q48c2w1", "q48c3w1",
   "q48c4w1", "q48c5w1", "q48c6w1", "q50w1")
# ② 숫자형 벡터 만들기
> select_variables <- c(1,4,9,10,19,101,102,103,328,329,330,331,
   332,333,334, 335,336,337,339,340,341,342,343,344,345,346,347,
   348,372,375,377, 379,484,485,486,487,488,489,496,497,498,499,
   500,501,502,503,504,532)
# ③ 문자형 혹은 숫자형 벡터에 입력된 변수를 데이터에서 선택
> test1 <- test[select_variables]
# 선택 결과 확인
> length(test1)
[1] 48
```

참고: 괄호의 사용

- 소괄호( )
  - ✓ 함수의 인자(arguments)를 입력할 때 주로 사용
- 중괄호{ }
  - ✓ 함수의 내용 부분 등을 정의하면서 여러 개의 표현식을 일괄처리 단위로 묶을 때 사용
  - ✓ 여러 줄을 한 줄에 쓸 때는 ;로 구분
    - for(i in 1:4){print(i); i=i+1}
- 대괄호[ ]
  - ✓ 벡터나 행렬의 구성요소를 가리키는 색인 위치나 조건을 입력할 때 사용
    - x[1]
  - ✓ 대괄호 안에 관계(비교)연산자와 논리연산자를 사용하여 행렬의 구성요소를 가리키는 조건 입력도 가능
    - x[x>4]

1. 예제데이터 만들기

✓ 사례 선택

– 특정 조건을 만족시키는 사례만을 선택

– 응답자의 학교 소재지역이 '서울'인 사례만 선택

» 학교 소재지역 변수는 scharew1이고, 변수값은 100 이상 200 미만

```
# 사례 선택
> spssdata <- test1[which(test1$scharew1 >= 100 &
                          test1$scharew1 < 200),]
# 선택 결과 확인
> length(row.names(spssdata))
[1] 595
```

```
spssdata                595 obs. of 48 variables
  id : int 71 72 73 74 75 76 77 78 79 80 ...
  sexw1 :Classes 'labelled', 'integer' atomic [1:595] 1 2 1 1 1 1 2 2 2 1 ...
  .. ..- attr(*, "label")= Named chr "성별"
  .. .. ..- attr(*, "names")= chr "sexw1"
  scharew1:Classes 'labelled', 'integer' atomic [1:595] 156 156 156 156 156 156 156 156 156 156 ...
  .. ..- attr(*, "label")= Named chr "학교 위치"
  .. .. ..- attr(*, "names")= chr "scharew1"
  areaw1 :Classes 'labelled', 'integer' atomic [1:595] 156 156 156 156 156 156 156 156 156 156 ...
  .. ..- attr(*, "label")= Named chr "학생 주소지"
  .. .. ..- attr(*, "names")= chr "areaw1"
  q2w1 :Classes 'labelled', 'integer' atomic [1:595] 3 1 1 3 2 1 2 1 2 2 ...
  .. ..- attr(*, "label")= Named chr "2. 학생은 현재 장래 얻고자 하는 구체적인 직업을 정해 놓으신 상태인가요?"
  .. .. ..- attr(*, "names")= chr "q2w1"
  q18a1w1 :Classes 'labelled', 'integer' atomic [1:595] 3 5 3 3 3 3 3 3 2 3 ...
  .. ..- attr(*, "label")= Named chr "18-a1. (과목별 반 성적 정도) 국어"
```

*which()

- which() 함수는 logical vector에서 TRUE 값을 갖는 element들의 위치(index)를 return

```
> x <- c(1,5,8,4,6)
> x
[1] 1 5 8 4 6
> which(x == 5)
[1] 2
> which(x != 5)
[1] 1 3 4 5
> df
> df[which(df$x <3), ]
> df[c(1,2), ]
> df[c(T, T, F), ]
```

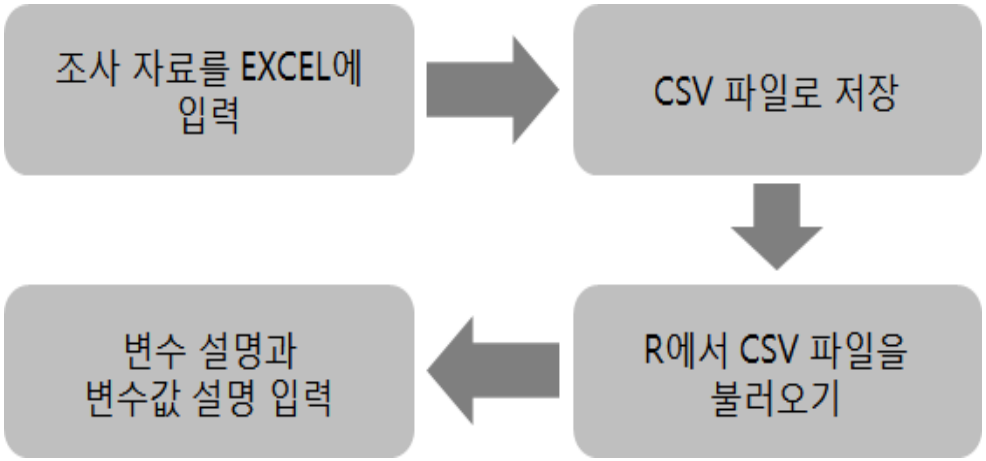
*dplyr 패키지를 활용하여 filtering

```
spssdata <- test1[which(test1$scharew1 >= 100 & test1$scharew1 <200),]
#this operation removes labels for values
str(test1$sexw1)
str(spssdata$sexw1)

spssdata <- test1[(test1$scharew1 >= 100 & test1$scharew1 <200),]
#this operation removes labels for values
str(test1$sexw1)
str(spssdata$sexw1)

library(dplyr)
spssdata <- filter(test1, test1$scharew1 >= 100 & test1$scharew1 <200)
dim(spssdata)
str(test1$sexw1)
str(spssdata$sexw1)
```

1. 예제데이터 만들기

2. CSV 파일을 이용한 데이터 만들기

- 직접 데이터를 만들 경우의 과정



- CSV 파일 불러오기
  - ✓ CSV 파일 데이터를 불러오기 위해 read.table (or read.csv)함수를 이용

```
# ① CSV 파일 불러오기
test <- read.table("(파일 경로)/04-1 중2 패널 1차년도 데이터(SPSS).csv",
         header=TRUE, sep=",")
```

```
names(test)[1] <- "id"
```

* read.csv

- read.csv is a fairly thin wrapper around read.table;

```
> read.csv
function (file, header = TRUE, sep = ",", quote = "\"", dec = ".",
    fill = TRUE, comment.char = "", ...)
read.table(file = file, header = header, sep = sep, quote = quote,
    dec = dec, fill = fill, comment.char = comment.char, ...)
<bytecode: 0x5e3fa88>
<environment: namespace:utils>
> read.csv2
function (file, header = TRUE, sep = ";", quote = "\"", dec = ",",
    fill = TRUE, comment.char = "", ...)
read.table(file = file, header = header, sep = sep, quote = quote,
    dec = dec, fill = fill, comment.char = comment.char, ...)
<bytecode: 0x5c0a330>
<environment: namespace:utils>
```

1. 예제데이터 만들기

- 변수 선택 및 사례 선택

```
# ② 변수 선택
# 문자형 벡터 만들기
select_variables <- c("id", "sexw1", "scharew1", "areaw1", "q2w1",
       "q18a1w1", "q18a2w1", "q18a3w1", "q33a01w1", "q33a02w1",
       "q33a03w1", "q33a04w1", "q33a05w1", "q33a06w1", "q33a07w1",
       "q33a08w1", "q33a09w1", "q33a10w1", "q33a12w1", "q33a13w1",
       "q33a14w1", "q33a15w1", "q34a1w1", "q34a2w1", "q34a3w1",
       "q34a4w1", "q34a5w1", "q34a6w1", "q37a01w1", "q37a02w1",
       "q37a03w1", "q37a04w1", "q48a01w1", "q48a02w1", "q48a03w1",
       "q48a04w1", "q48a05w1", "q48a06w1", "q48b1w1", "q48b2w1",
       "q48b3w1", "q48c1w1", "q48c2w1", "q48c3w1", "q48c4w1",
       "q48c5w1", "q48c6w1", "q50w1")
# 숫자형 벡터 만들기
select_variables <- c(1,4,9,10,19,101,102,103,328,329,330,331,332,
       333,334,335,336,337,339,340,341,342,343,344,345,346,347,
       348,372,375,377, 379,484,485,486,487,488,489,496,497,498,
       499,500,501,502,503,504,532)
# 문자형 혹은 숫자형 벡터에 입력된 변수명과 동일한 변수를 데이터에서 선택
test1 <- test[select_variables]

# ③ 사례 선택
csvdata <- test1[which(test1$scharew1 >= 100 &
                       test1$scharew1 < 200),]
```

1. 예제데이터 만들기

- 결측값 지정
  - ✓ 결측값으로 지정된 값에 대해 분석에서 제외할 수 있도록 'NA'로 지정

```
# ⑥ 결측값 지정
table(csvdata$q33a07w1)
csvdata$q33a07w1[csvdata$q33a07w1==9] <- NA
table(csvdata$q33a07w1, useNA = "ifany")
```

결측값으로 지정한 결과를 저장할 대상

특정 변수값을 결측값으로 지정할 대상 변수와 조건을 입력

지정된 대상 변수의 조건에 해당되는 경우 이 값으로 변경

2. 예제데이터에 대한 설명

1. 성별

- 변수설명(variable label): 학생의 성별 구분
- 변수명(variable name): **sexw1**
- *변수값(variable value): 1, 2*
- *변수값 설명(value label) : 1 = 남자, 2 = 여자*

2. 학교위치

- 변수설명(variable label): 응답자가 다니는 학교위치를 시군구 단위까지 분류하여 제공
- 변수명(variable name): **scharew1**

3. 거주지역

- 변수설명(variable label): 응답자의 주소지를 시군구 단위까지 분류하여 제공
- 변수명(variable name): **areaw1**

2. 예제데이터에 대한 설명

4. 성적

- 일반적인 학업 성취 정도
- 변수명: **q18a**1w1(국어), q18a2w1(영어), q18a3w1(수학)

5. 부모에 대한 애착

- 청소년이 부모에 대해서 가지는 정서적 친밀도
- 변수명: **q33a01**w1부터 q33a06w1

6. 부모 감독

- 부모가 자녀의 일상생활에 대해서 파악하고 있는 정도
- 변수명: **q33a07**w1부터 q33a10w1

2. 예제데이터에 대한 설명

7. 부정적 양육방식

- 부모가 자녀를 양육하면서 부정적인 모습을 보이거나 욕설이나 체벌을 하는 등의 직접적인 영향을 미침으로써 부정적으로 자녀를 양육하는 것
- 변수명: **q33a12**w1부터 q33a15w1

8. 자기통제력

- 순간만족과 충동을 조절할 수 있는지, 스릴과 모험을 추구하기 보다는 분별력과 조심성이 있는지, 근시안적이기 보다는 앞으로의 일을 생각하는지, 쉽게 흥분하는 성격인지의 여부 등
- 변수명: **q34**a1w1부터 q34a6w1

2. 예제데이터에 대한 설명

9. 청소년 비행

- 지위비행(흡연, 음주, 무단결석, 가출)에 대해 조사시점으로부터 지난 1년 동안 해당 항목의 행동을 해본 적이 있는지 경험 유무
- 변수명: **q37a**01w1(흡연), q37a02w1(음주), q37a03w1(무단결석), q37a04w1(가출)

10.자아존중감

- 자신이 스스로를 소중하게 생각하는 정도
- 변수명: **q48a01**w1부터 q48a06w1

2. 예제데이터에 대한 설명

11.자기신뢰감

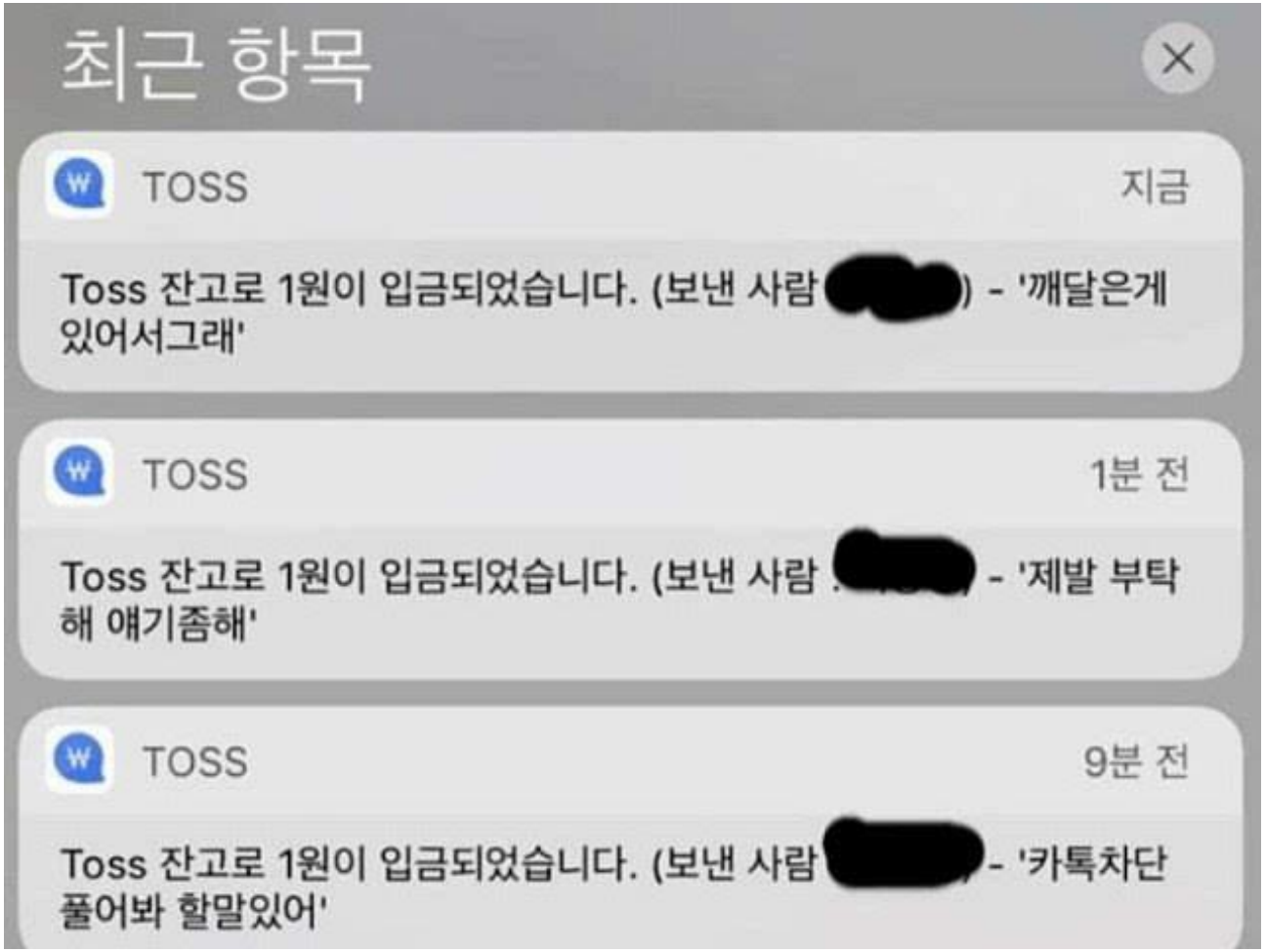
- 자기 자신에 대해서 믿는 정도
- 변수명: **q48b1**w1부터 q48b3w1

12.공격성

- 내면에 있는 분노나 다른 사람들에 대한 공격적 성향의 정도
- 변수명: **q48c1**w1부터 q48c6w1

13.삶의 만족도

- 자신의 삶에 대해서 전반적으로 만족하는 정도의 평가
- 변수명: **q50**w1


최근 항목
TOSS
지금
Toss 잔고로 1원이 입금되었습니다. (보낸 사람 [redacted]) - '깨달은게 있어서그래'
TOSS
1분 전
Toss 잔고로 1원이 입금되었습니다. (보낸 사람 [redacted] - '제발 부탁해 얘기좀해'
TOSS
9분 전
Toss 잔고로 1원이 입금되었습니다. (보낸 사람 [redacted] - '카톡차단 풀어봐 할말있어'

3. 통계분석을 위한 데이터 변환

1. 새 변수 만들기

- 새 변수를 먼저 적어주고 다음에 할당기호(<-)를 적고 그 뒤에 새변수를 만드는 다양한 조건들을 표현하면 됨

```
(새 변수명) <- (다양한 조건)
```

- 데이터프레임명$변수명을 사용하는 방법

```
# 데이터프레임명$변수명을 사용하는 방법
spssdata$attachment <- spssdata$q33a01w1+spssdata$q33a02w1+
     spssdata$q33a03w1+spssdata$q33a04w1+spssdata$q33a05w1+
     spssdata$q33a06w1
```

✓ 모든 변수를 사용할 때 그 앞에 데이터프레임명을 다 적어서 더해주고 그 결과를 새 변수 attachment에 할당

✓ spssdata 데이터 안에 attachment라는 새 변수가 생성

3. 통계분석을 위한 데이터 변환

- attach 함수를 사용하는 방법

```
# attach 함수를 사용하는 방법
attach(spssdata)
spssdata$attachment <- q33a01w1+q33a02w1+q33a03w1+q33a04w1+
q33a05w1+q33a06w1
detach(spssdata)
```

✓ attach(spssdata) 명령어를 실행시켜서 spssdata 데이터를 불러옴

✓ 6개의 변수들을 더해서 새 변수 attachment에 할당

– **새 변수인 attachment 앞에는 데이터프레임명을 붙여야 함**

✓ detach(spssdata) 명령어를 실행시켜 불러온 spssdata를 원위치시킴

- transform 함수를 사용하는 방법

```
# transform 함수를 사용하는 방법
spssdata <- transform(spssdata, attachment = q33a01w1+q33a02w1+
                      q33a03w1+q33a04w1+q33a05w1+q33a06w1)
```

✓ transform 함수는 데이터프레임에 결과를 할당함

3. 통계분석을 위한 데이터 변환

2. 재부호화

- 기존 변수의 변수값에 주어진 조건에 따라서 새로운 변수값을 부여
- 재부호화를 사용하는 대표적인 상황
  - ✓ 연속변수를 재부호화해서 몇 개의 집단으로 구성된 범주형 변수로 만드는 상황
  - ✓ 잘못 입력된 값을 올바른 값으로 수정하는 상황
  - ✓ 변수의 방향을 바꾸기 위해서

```
변수[조건] <- 표현    # 조건이 참일 때만 표현을 할당함
```

3. 통계분석을 위한 데이터 변환

1) 그룹변수 만들기

```
# ① 학교성적 변수 만들기
attach(spssdata)
spssdata$grade <- q18a1w1+q18a2w1+q18a3w1
detach(spssdata)
```

✓ 학교성적이라는 변수를 분석에 사용하기 위해서 집단 변수로 만드는 경우

– 먼저 세 변수를 사용해서 성적변수를 만들고, 이 변수의 분포를 살펴본 후에, 적절하게 구분해서 상, 중, 하라는 세 집단을 가진 범주형 변수로 생성

```
# ② 학교성적 변수(grade)의 분포 살펴보기
table(spssdata$grade)
prop.table(table(spssdata$grade))
```

✓ table 함수와 prop.table 함수를 이용하여 분포를 살펴봄

3. 통계분석을 위한 데이터 변환

```
> table(spssdata$grade)
 3  4  5  6  7  8  9 10 11 12 13 14 15
 6  8 11 31 66 80 98 93 77 67 30 17 11
> prop.table(table(spssdata$grade))
…
> hist(spssdata$grade, label = T, right = T)
> ggplot(data = spssdata, aes(x=spssdata$grade)) +
    geom_histogram(fill="lightgreen",
                   bins = 13,
                   color="grey50") +
    labs(title = "Histogram for Grades", x = "Grade", y = "Count")
> round(prop.table(table(spssdata$grade)), 2)
```

- ✓ 성적 변수의 분포를 살펴보아 성적의 정도에 따라 학교성적이 높은 집단, 학교성적이 중간인 집단, 그리고 학교성적이 낮은 집단으로 구분
- ✓ 학교성적 정도에 따른 집단은 가급적 비슷한 사례수가 되도록 하기 위해 누적 비율이 33.3%와 66.7%를 기준으로 이 기준에 가장 가까운 값으로 집단을 나누도록 함

3. 통계분석을 위한 데이터 변환

```
# ③ 학교성적 변수를 3집단으로 분류
# 학교성적 변수를 정도에 따라 3집단으로 분류 및 변수와 변수값 설명 입력
attach(spssdata)
spssdata$grp.grade[grade<=8] <- 1
spssdata$grp.grade[grade>=9 & grade<=10] <- 2
spssdata$grp.grade[grade>=11] <- 3
detach(spssdata)
table(spssdata$grp.grade)
```

- ✓ 학교성적 변수(grade)가 가장 낮은 점수부터 8점까지는 낮은 학교성적 집단을 의미하는 숫자로 '1'을 부여하여 새로운 변수인 학교성적 정도에 따른 집단 변수(grp.grade)에 저장
- ✓ 학교성적 변수가 9점과 10점인 경우에는 중간 학교성적 집단을 의미하는 숫자로 '2'를 부여하여 학교성적 정도에 따른 집단 변수에 저장
- ✓ 학교성적 변수가 11점 이상부터 최대값까지는 높은 학교성적 집단을 의미하는 숫자로 '3'을 부여하여 학교성적 정도에 따른 집단 변수에 저장

3. 통계분석을 위한 데이터 변환

```
> table(spssdata$grp.grade)
  1   2   3
202 191 202
```

✓ 새로 만든 집단 변수인 grp.grade의 빈도분포를 구해보면 성적이 낮은 집단(1)이 202명, 중간인 집단(2)이 191명, 높은 집단(3)이 202명으로 나타남

3. 통계분석을 위한 데이터 변환

2) 새 변수로 재부호화하기

```
# q50w1의 속성을 '부정', '중립', 그리고 '긍정' 응답으로 재부호화
attach(spssdata)
spssdata$satisfaction[q50w1<=2] <- 1  # 만족하지 못하는 편
spssdata$satisfaction[q50w1==3] <- 2            # 보통
spssdata$satisfaction[q50w1>=4] <- 3  # 만족하는 편
detach(spssdata)
```

- ✓ attach 함수를 이용하여 데이터(spssdata)를 지정
- ✓ spssdata 데이터 내의 전반적인 생활만족도 변수인 q50w1을 재부호화
- ✓ 변수 q50w1의 변수값에서 1 또는 2는 '만족하지 못하는 집단'을 의미하는 1로 재부호화하고([q50w1<=2] <- 1), 재부호화한 결과는 spssdata 내의 satisfaction 변수(spssdata$satisfaction)에 저장
- ✓ 같은 방법으로 변수 q50w1의 변수값에서 3은 '보통인 집단'을 의미하는 2로 재부호화
- ✓ 다음으로 변수 q50w1의 변수값에서 4 또는 5는 '만족하는 집단'을 의미하는 3로 재부호화

3. 통계분석을 위한 데이터 변환

3) 변수의 방향을 역으로 재부호화하기

```
# 반대로 측정된 변수의 값을 재부호화
attach(spssdata)
spssdata$rq48a04w1[q48a04w1==1] <- 5
spssdata$rq48a04w1[q48a04w1==2] <- 4
spssdata$rq48a04w1[q48a04w1==3] <- 3
spssdata$rq48a04w1[q48a04w1==4] <- 2
spssdata$rq48a04w1[q48a04w1==5] <- 1

spssdata$rq48a05w1[q48a05w1==1] <- 5
spssdata$rq48a05w1[q48a05w1==2] <- 4
spssdata$rq48a05w1[q48a05w1==3] <- 3
spssdata$rq48a05w1[q48a05w1==4] <- 2
spssdata$rq48a05w1[q48a05w1==5] <- 1

spssdata$rq48a06w1[q48a06w1==1] <- 5
spssdata$rq48a06w1[q48a06w1==2] <- 4
spssdata$rq48a06w1[q48a06w1==3] <- 3
spssdata$rq48a06w1[q48a06w1==4] <- 2
spssdata$rq48a06w1[q48a06w1==5] <- 1
detach(spssdata)
```

3. 통계분석을 위한 데이터 변환

- ✓ 자아존중감을 측정한 6개의 문항 중에서 처음 3문항과 나중의 3문항은 서로 다른 방향으로 측정됨
  - 첫 3문항은 점수가 높을수록 자아존중감이 높은 것으로 해석되지만, 나중의 3문항은 점수가 높으면 자아존중감이 낮아지는 것으로 해석될 수 있음
  - 이 변수들을 묶어서 하나의 변수로 만들어 사용하기 위해서는 3개의 문항의 변수값의 방향을 바꾸어주어야 함
- ✓ 재부호화 방법을 사용하여 나중의 **세 변수(q48a04w1, q48a05w1, q48a06w1)에서** 1에는 5, 2에는 4, 3에는 3, 4에는 2, 5에는 1로 재부호화하여 이를 새 변수인 rq48a04w1에서 rq48a06w1에 할당

3. 통계분석을 위한 데이터 변환

4) 두 독립변수의 변수값을 교차시켜 하나의 집단 변수 만들기

```
# 두 독립변수의 변수값을 교차시켜 하나의 집단변수 만들기
attach(spssdata)
spssdata$grp.sex.grade[sexw1==1 & grp.grade==1] <- 11
spssdata$grp.sex.grade[sexw1==1 & grp.grade==2] <- 12
spssdata$grp.sex.grade[sexw1==1 & grp.grade==3] <- 13
spssdata$grp.sex.grade[sexw1==2 & grp.grade==1] <- 21
spssdata$grp.sex.grade[sexw1==2 & grp.grade==2] <- 22
spssdata$grp.sex.grade[sexw1==2 & grp.grade==3] <- 23
detach(spssdata)
```

- ✓ 성별 변수와 학교성적 변수를 교차시켜 하나의 변수로 만들기 위해 '남자 청소년(sexw1==1)'이면서 '낮은 학교성적 집단(grp.grade==1)'인 경우에는 새로운 부호('11')를 지정하여 새로운 변수(grp.sex.grade)에 할당
- ✓ '여자 청소년(sexw1==2)'이면서 '낮은 학교성적 집단(grp.grade==1)'인 경우에는 새로운 부호('21')를 지정하여 새로운 변수(grp.sex.grade)에 할당하는 식으로 두 독립변수를 교차시킨 모든 경우에 새로운 부호를 지정하여 새로운 변수에 저장하는 방법을 사용

3. 통계분석을 위한 데이터 변환

3. 결측값 지정하기 *(csv파일일 경우)*

```
# 결측값 지정
spssdata$q33a07w1[spssdata$q33a07w1==9] <- NA
```

결측값으로 지정한 결과를 저장할 대상

특정 변수값을 결측값으로 지정할 대상 변수와 조건을 입력

지정된 대상 변수의 조건에 해당되는 경우 이 값으로 변경

- ✓ 특정 값을 결측값으로 지정하는 것은 앞에서 설명한 재부호화의 특수한 경우에 해당
  - R에서 결측값을 의미하는 기호는 'NA(not available)'
  - 특정 조건에 해당하는 값을 'NA'라고 지정

3. 통계분석을 위한 데이터 변환

4. 날짜변수 사용하기

```
# 날짜관련 변수의 사용
# 관련 데이터프레임 만들기
id <- c(1, 2, 3, 4, 5)
born <- c("1989-02-13", "1990-05-25", "1992-11-30", "1993-07-01",
          "1991-09-22")
first.crime <- c("2007-05-17", "2009-02-21", "2006-09-01",
                 "2009-08-19", "2010-01-02")
second.crime <- c("2010-03-10", "2011-10-01", "2007-12-21",
                  "2012-01-05", "2015-10-12")
sampledata <- data.frame(id, born, first.crime, second.crime)
sampledata
str(sampledata$born)
```

- ✓ 날짜변수를 사용하기 위해서 5사례에 대한 가상 데이터를 만드는데, id는 사례를 구분하는 숫자이며, born은 생년월일, first.crime은 첫 번째 범죄를 저지른 날짜, second.crime은 두 번째 범죄, 즉 재범을 저지른 날짜를 의미함
- ✓ 각각의 벡터를 만든 후에 data.frame 함수를 사용하여 데이터를 만들고 sampledata라는 이름으로 저장한다

3. 통계분석을 위한 데이터 변환

```
> sampledata
  id       born first.crime second.crime
1  1 1989-02-13  2007-05-17   2010-03-10
2  2 1990-05-25  2009-02-21   2011-10-01
3  3 1992-11-30  2006-09-01   2007-12-21
4  4 1993-07-01  2009-08-19   2012-01-05
5  5 1991-09-22  2010-01-02   2015-10-12
> str(sampledata$born)
 Factor w/ 5 levels "1989-02-13","1990-05-25",..: 1 2 4 5 3
```

- ✓ sampledata라는 데이터프레임 확인
- ✓ 날짜변수의 속성을 알아보기 위해 생년월일 변수(born)의 구조를 str 함수를 사용해서 살펴봄
  - 생년월일 변수는 **Factor**라고 제시됨
  - 입력된 날짜를 숫자가 아니라 **문자로** 인식하고 있음
- ✓ 이 변수를 숫자형식의 **날짜변수로 사용하기 위해서는** 데이터 변환을 시도해야 함
  - 이 경우에 사용할 수 있는 함수가 **as.Date** 함수임

3. 통계분석을 위한 데이터 변환

```
# 문자변수를 날짜변수로 바꾸기
sampledata$born.date <- as.Date(sampledata$born)
sampledata$first.crime.date <- as.Date(sampledata$first.crime)
sampledata$second.crime.date <- as.Date(sampledata$second.crime)
str(sampledata$born.date)
```

✓ as.Date 함수를 사용해서 born, first.crime, second.crime 변수를 날짜변수로 변환하여, 각각 born.date, first.crime.date, second.crime.date에 할당

```
> str(sampledata$born.date)
 Date[1:5], format: "1989-02-13" "1990-05-25" "1992-11-30"
"1993-07-01" "1991-09-22"
```

✓ 결과를 확인하면 새로 생성된 born.date 변수는 날짜 형식

3. 통계분석을 위한 데이터 변환

```
# 나이계산하기
today <- Sys.Date()
difftime(today, sampledata$born.date, units="days")

# 재범기간 계산하기
difftime(sampledata$second.crime.date, sampledata$first.crime.date,
         units="days")
```

- ✓ 두 시점 간의 시간을 계산
  - 의료계에서 질병의 재발기간이나 범죄경력연구에서 재범기간 등의 계산에 유용하게 사용될 수 있음
- ✓ 현재 날짜를 보여주는 Sys.Date 함수를 사용하여 오늘 날짜를 today에 저장
- ✓ difftime 함수를 사용하여 today와 출생일자(sampedata$born.date) 간의 시점 차이를 계산
  - units 인자를 사용하여 계산한 결과를 초(seconds), 분(minutes), 시간(hours), 일(days), 주(weeks) 단위로 표시할 수 있음
- ✓ 두 번째로 동일한 방법으로 재범기간, 즉 첫 번째 범죄와 두 번째 범죄 간의 기간을 계산

3. 통계분석을 위한 데이터 변환

```
> # 나이계산하기
> today <- Sys.Date()
> today
[1] "2016-03-04"
> difftime(today, sampledata$born.date, units="days")
Time differences in days
[1] 9881 9415 8495 8282 8930
> # 재범기간 계산하기
> difftime(sampledata$second.crime.date, sampledata$first.crime.date,
+ units="days")
Time differences in days
[1] 1028  952  476  869 2109
```

- ✓ 첫 번째 분석결과로 출생 이후 오늘까지의 기간이 일(days) 단위로 출력
- ✓ 다음으로 재범기간이 일 단위로 계산되어 출력
  - 재범기간이 가장 짧은 것은 3번(476일)이며, 가장 긴 사람은 5번(2,109일)

3. 통계분석을 위한 데이터 변환

- R에서의 시간표현 형식

기호	의미	예
%d	일을 숫자로 나타낸다.	01-31
%a	요일을 간략하게 표현한다.	월, 화, 수 등
%A	요일을 모두 표현한다.	월요일, 화요일 등
%m	달을 숫자로 표현한다.	01-12
%b	달을 간략하게 표현한다.	1-12
%B	달을 모두 표현한다.	1월-12월
%y	연도를 2자리로 표현한다	16
%Y	연도를 4자리로 표현한다.	2016

```
> format(sampledata$born.date, format="%B %d일 %Y")
> print(format(sampledata$born.date, format="%B %d일 %Y"))
> print(sampledata$born.date)
```

3. 통계분석을 위한 데이터 변환

5. 데이터 병합하기

1) 변수를 병합하는 경우

```
# 변수를 병합하는 경우
# 2차년도 데이터 불러오기
second <- read.table("04-2 중2 패널 2차년도 데이터(SPSS).csv",
                     header=TRUE, sep=",")
names(second)[1] <- "id"
# 데이터 합치기
mergedata <- merge(spssdata, second, by="id")
```

✓ merge 함수를 이용하여 변수를 병합

- merge 함수의 인자로 병합할 데이터프레임의 이름을 차례로 적어주고, 다음으로 by를 통해서 두 데이터프레임에 공통적으로 존재하고 **병합의 기준이 되는 주변수(key variable)를 지정**
- 두 데이터가 병합할 때에는 1차년도의 응답자와 2차년도의 응답자가 동일한 응답자로 지정되어야 하므로 각 데이터에서 응답자의 고유번호인 'id' 변수를 기준으로 병합

3. 통계분석을 위한 데이터 변환

2) 사례를 병합하는 경우

```
# 사례수를 병합하는 경우
total <- rbind(dataA, dataB)
```

✓ 동일한 변수를 가진 두 개의 데이터프레임을 수직으로 병합할 때는 rbind 함수를 사용

– rbind 함수의 인자로, 병합하는 두 데이터(여기에서는 가상으로 dataA, dataB)의 이름을 적어주고, 그 결과를 total이라는 데이터 프레임으로 저장한다

6. 데이터 분할하기

1) 일부 변수를 추출하는 경우

```
# spssdata의 첫 세 변수만 추출하는 경우 1
newdata <- spssdata[,c(1:3)]
```

✓ 쉼표 앞이 **빈칸이기 때문에 모든 사례를 선택하고**, 변수는 첫 번째부터 세 번째 변수를 선택한다는 의미

```
# spssdata의 첫 세 변수만 추출하는 경우 2
var <- c("id", "sexw1", "scharew1")
newdata <- spssdata[var]
```

✓ **변수명을** 사용하는 경우에는 변수명 이름을 지정한 **객체를 먼저 만들고** 데이터에 그것을 지정

3. 통계분석을 위한 데이터 변환

2) 일부 사례만 선택하는 경우

```
# spssdata의 처음 5개 사례만 추출하는 경우
newdata <- spssdata[1:5,]
```

- ✓ dataframe[행, 열]의 표현을 사용하는 방법
  - 앞에서 일부 변수를 선택할 때와 반대의 방법

```
# 남자만 선택하는 경우
newdata <- spssdata[which(spssdata$sexw1==1),]
# 남자 중에서 성적이 낮은 경우만 선택하는 경우
newdata <- spssdata[which(spssdata$sexw1==1 & grp.grade==1),]
```

- ✓ 특정 조건을 지정하게 **그것을 충족하는 사례만을 선택하도록** 하는 경우
  - which 함수를 사용
  - 두 가지의 조건을 모두 충족시키는 사례만 선택할 수도 있음

3. 통계분석을 위한 데이터 변환

3) subset 함수를 사용한 경우

```
# subset 함수를 사용하여 사례와 변수를 추출하는 경우
newdata <- subset(test, scharew1 >= 100 & scharew1 < 200, select=
c("id", "sexw1", "scharew1", "areaw1", "q2w1",
  "q18a1w1", "q18a2w1", "q18a3w1", "q33a01w1", "q33a02w1", "q33a03w1",
  "q33a04w1", "q33a05w1", "q33a06w1", "q33a07w1", "q33a08w1",
  "q33a09w1", "q33a10w1", "q33a12w1", "q33a13w1", "q33a14w1",
  "q33a15w1", "q34a1w1", "q34a2w1", "q34a3w1", "q34a4w1", "q34a5w1",
  "q34a6w1", "q37a01w1", "q37a02w1", "q37a03w1", "q37a04w1",
  "q48a01w1", "q48a02w1", "q48a03w1", "q48a04w1", "q48a05w1",
  "q48a06w1", "q48b1w1", "q48b2w1", "q48b3w1", "q48c1w1", "q48c2w1",
  "q48c3w1", "q48c4w1", "q48c5w1", "q48c6w1", "q50w1"))
```

- ✓ subset 함수를 사용할 때는 인자로서 먼저 데이터셋의 이름을 지정하고(test), 다음으로 **사례를 선택할 조건을 지정하며**(scharew1 >= 100 & scharew1 < 200), 마지막으로 select 인자를 사용하여 **남겨두고자 하는 변수를 지정**
- ✓ 데이터셋에서 따로 떨어진 변수들을 추출하고자 할 때는 c("변수명", "변수명")의 표현을 사용하여 지정

```
save(spssdata, file="spssdata.Rda")  #작업 데이터 저장
rm(spssdata) #spssdata 삭제
load("spssdata.Rda") #저장 데이터 불러들이기
```

Weekly Assignment #2

1. dplyr 패키지 설명 (with 예제 데이터)
2. tidyr 패키지 설명 (with 예제 데이터)
3. ggplot2 패키지 설명 (with 예제 데이터)

*상위 3팀만 만점 부여