

# 강의교안 이용 안내

- 본 강의교안의 저작권은 이윤환과 한빛아카데미(주)에 있습니다.
- 이 자료를 무단으로 전제하거나 배포할 경우 저작권법 136조에 의거하여 벌금에 처할 수 있고 이를 병과(併科)할 수도 있습니다.





제대로 알고 쓰는  
**R 통계분석**

## CHAPTER 07

# 여러 모집단의 평균비교 검정

# Contents


## 7.1 모집단이 두 개인 경우

- 두 모집단의 종류
- 서로 독립인 두 집단에서의 평균 차이 검정
- 서로 대응인 두 집단에서의 평균 차이 검정

## 7.2 모집단이 세 개 이상

- 분석방법 : 일원분산분석
- 분산분석표

## 8장을 위한 준비

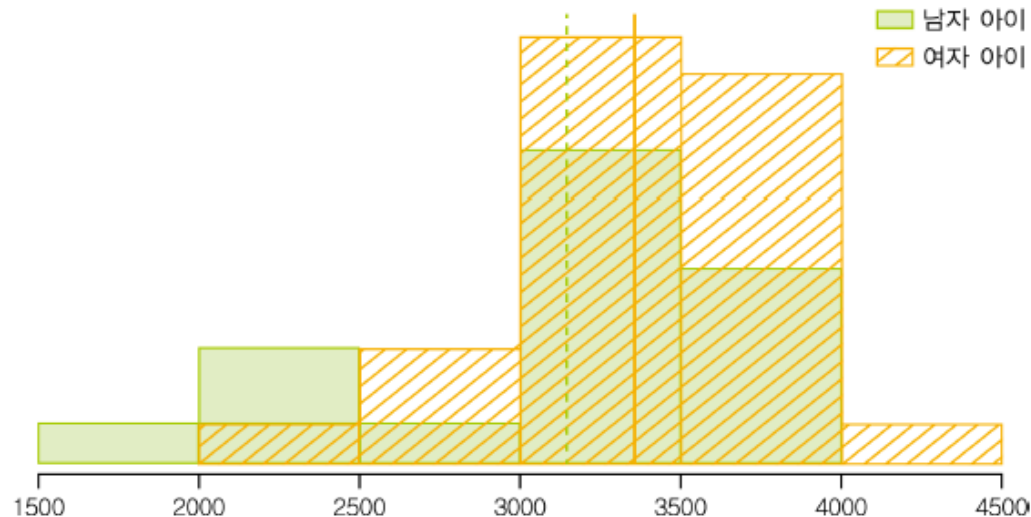


# 01. 모집단이 두 개인 경우

1. 독립표본과 대응표본에 대해 학습한다.
2. 서로 독립인 두 집단에서의 평균 차이 검정에 대해 학습한다.
3. 서로 대응인 두 집단에서의 평균 차이 검정에 대해 학습한다.

## 두 집단의 종류

- 모집단이 두 개인 경우는 ‘서로 독립인 두 집단’과 ‘대응을 이루는 두 집단’이 있습니다.
- 서로 독립인 두 집단 : 독립표본
  - 각 집단을 변수에 의해 두 개로 구분할 때 서로 영향을 끼치지 않는 집단입니다.
  - 예) 성별 변수에 의해 나뉜 남자 아이와 여자 아이의 몸무게

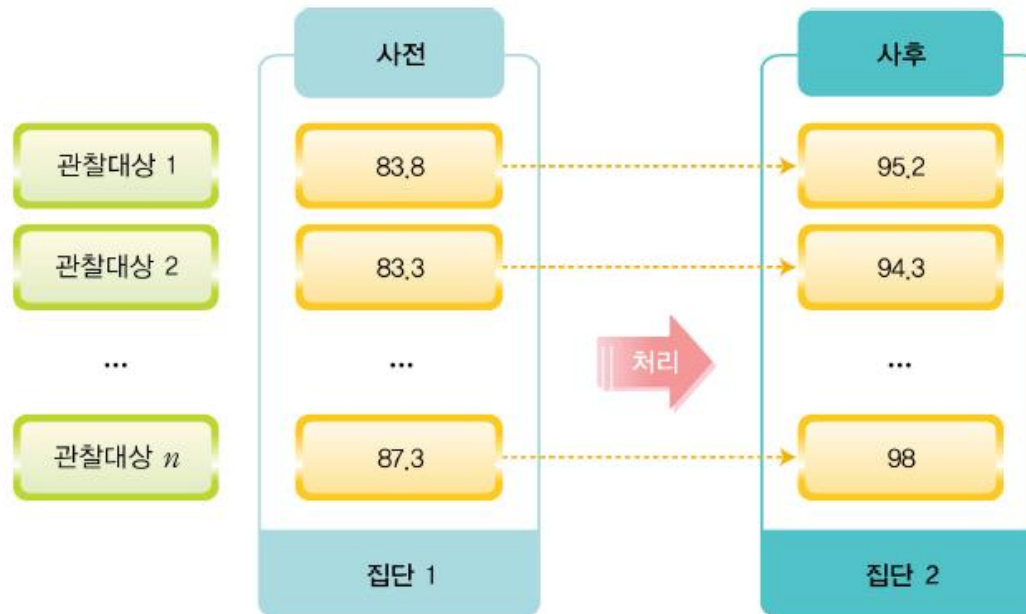


[그림 7-1] 남자 아이와 여자 아이의 체중(독립인 두 집단)

# 두 집단의 종류

## • 서로 대응인 두 집단 : 대응표본

- 주로 ‘처리’ 효과를 알기 위해 사용합니다.
- 동일한 관찰 대상으로 부터 특정 ‘처리’를 실시하기 이전에 관찰되는 모집단과 ‘처리’ 이후에 관찰되는 모집단의 두 모집단입니다.
- 예) 신경성 식욕부진증 치료제 투약 이전과 이후의 체중



[그림 7-2] 대응표본

## 서로 독립인 두 모집단 : 평균 차이검정

- 먼저 모집단의 분산을 모르는 경우가 많으므로, 모집단의 분산을 모를 때로 하겠습니다. (t-분포 이용)
- 다음으로 모집단이 다음의 가정을 만족하는지 확인합니다.
  - 서로 독립인 두 모집단은 정규분포를 이룬다.
    - ‘정규성’이라 하며, 이를 만족하는지 검정해야 합니다. 본 책에서는 ‘정규성’은 만족하는 것으로 가정하겠습니다.
  - 두 집단의 분산은 서로 동일하다
    - ‘등분산성’이라 하며, R의 분산 비교 검정함수를 이용하여 분산이 서로 동일한지 검정해 봅시다.

# 서로 독립인 두 모집단 : 평균 차이검정

## • R을 이용한 분산의 동일성 검정

### 예제 1 남아 신생아와 여아 신생아의 몸무게

다음은 6장의 [7장을 위한 준비]에서 만든 'chapter7.txt' 자료로 여아 신생아 18명의 몸무게와 남아 신생아 26명의 몸무게가 기록된 자료입니다.

여아	3837	3334	2208	1745	2576	3208	3746	3523	3430	3480
	3116	3428	2184	2383	3500	3866	3542	3278		
남아	3554	3838	3625	2846	3166	3520	3380	3294	3521	2902
	2635	3920	3690	3783	3345	3034	3300	3428	4162	3630
	3406	3402	3736	3370	2121	3150				

이로부터 유의수준은 0.05로 하여 여아와 남아의 분산이 서로 동일한지 검정합니다.



# 서로 독립인 두 모집단 : 평균 차이검정

- ▣ 가설수립 : 두 집단의 분산이 동일하면 그 비가 1
  - 영가설 : 두 집단의 분산은 서로 동일하다. ( $H_0: \frac{\sigma^2_{\text{여아몸무게}}}{\sigma^2_{\text{남아몸무게}}} = 1$ )
  - 대안가설 : 두 집단의 분산은 동일하지 않다. ( $H_1: \frac{\sigma^2_{\text{여아몸무게}}}{\sigma^2_{\text{남아몸무게}}} \neq 1$ )
- ▣ 검정 통계량 ( $H_0$ 가 참이라는 가정)
  - 4장에서 알아본 표본분포 중 F-분포를 사용합니다.

$$F = \frac{V_1/(n-1)}{V_2/(m-1)} = \frac{\frac{(n-1)S_1^2}{\sigma_1^2}/(n-1)}{\frac{(m-1)S_2^2}{\sigma_2^2}/(m-1)} = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F(n-1, m-1)$$

에서 영가설이 참이면,  $\sigma_1^2 = \sigma_2^2$  이므로 검정통계량  $F = \frac{S_1^2}{S_2^2} \sim F(n-1, m-1)$

# 서로 독립인 두 모집단 : 평균 차이검정

## ▣ R을 이용한 검정

- 분산의 동일성을 검정 R 함수는 `var.test()` 입니다.
- 성별(gender)로 나뉜 두 집단의 몸무게(weight)는 R의 표현식으로 다음과 같습니다.

**‘data\$weight ~ data\$gender’**

- 이 표현식을 `var.test()`에 전달인자로 전달합니다.

```
1. data <- read.table("./data/chapter7.txt", header=T)
2. var.test(data$weight ~ data$gender)
```

- 1줄 : data 폴더에 있는 chapter7.txt를 읽어 첫 줄을 변수명으로 하는 데이터 프레임 변수 data에 저장
- 2줄 : 성별로 몸무게 분산의 동일성 검정

# 서로 독립인 두 모집단 : 평균 차이검정

```
> var.test(data$weight ~ data$gender)
```

F test to compare two variances

data: data\$weight by data\$gender

F = **2.1771**, num df = 17, denom df = 25, **p-value = 0.07526**

alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:

0.9225552 5.5481739

sample estimates:

ratio of variances  
2.177104

## 판정

- 기각역을 이용한 판정 : 검정통계량이 채택역 구간인 (0.39, 2.36) 사이에 있어 영가설을 채택합니다.
- 유의확률을 이용한 판정 : 유의확률은 0.07526으로 유의수준 0.05보다 크므로 영가설을 채택합니다.

# 서로 독립인 두 모집단 : 평균 차이검정

## □ 결론

- 남아와 여아 몸무게의 분산의 동일성을 검정한 결과 두 집단의 분산이 서로 동일하다는 가정을 만족하는 것으로 판단됩니다.
- 이로부터 두 집단의 분산은 동일함을 확인하고 평균 검정을 실시해 봅시다.

## • 서로 독립인 두 모집단의 평균 차이 검정

- 신생아의 자료를 이용하여, 남아 신생아의 몸무게의 평균이 여아 신생아의 몸무게의 평균보다 큰 지 유의수준 0.05에서 검정해 봅시다.

## □ 가설수립

- 영가설 : 남아와 여아 신생아의 몸무게의 평균은 서로 같다. (차이가 없다)

$$H_0: \mu_{\text{여아 몸무게}} - \mu_{\text{남아 몸무게}} = 0$$

- 대안가설 : 여아 신생아의 몸무게의 평균이 남아 신생아의 몸무게의 평균보다 작다.

$$H_1: \mu_{\text{여아 몸무게}} - \mu_{\text{남아 몸무게}} < 0$$

# 서로 독립인 두 모집단 : 평균 차이검정

- 검정통계량 : 두 집단의 분산이 서로 같을 때
  - 두 집단의 평균 차이에 대한 검정

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\text{Var}(\bar{X}_1 - \bar{X}_2)}} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}}$$

- $\mu_1, \mu_2$  : 두 모집단의 평균,  $\sigma_1^2, \sigma_2^2$  : 두 모집단의 분산
- $n, m$  : 두 표본의 크기,  $\bar{X}_1, \bar{X}_2$  : 두 표본의 평균
- 두 집단의 분산이  $\sigma^2$ 으로 서로 같은 경우

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n} + \frac{\sigma_2^2}{m}}} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\sigma^2(\frac{1}{n} + \frac{1}{m})}} = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sigma\sqrt{\frac{1}{n} + \frac{1}{m}}}$$

- 여기서 모집단의 분산을 알지 못하므로 두 집단이 서로 동일한 분산을 알지 못해 이를 추정하기 위한 합동분산  $S_p^2$  을 추정량으로 사용합니다. ( $S_1^2, S_2^2$  : 두 표본의 분산)

$$S_p^2 = \frac{(n-1)S_1^2 + (m-1)S_2^2}{n+m-2}$$

## 서로 독립인 두 모집단 : 평균 차이검정

- ▣ 앞서 구한 합동분산  $s_p^2$ 에 제곱근을 취해 얻은  $s_p$ 를 두 집단의 동일한 표준편차의 추정량으로 하여 다음과 같은 검정통계량을 사용합니다.
  - 전체의 자유도는 각각의 자유도의 합인  $n - 1 + m - 1 = n + m - 2$ 입니다.

$$T = \frac{(\overline{X}_1 - \overline{X}_2) - (\mu_1 - \mu_2)}{s_p \sqrt{\frac{1}{n} + \frac{1}{m}}} \sim t(n + m - 2)$$

- ▣ 우리의 예에서 여아의 표본수 18, 남아의 표본수 26 이므로 자유도는 42입니다.

# 서로 독립인 두 모집단 : 평균 차이검정

- R을 이용하여 검정통계량을 구하고 검정을 실시해 봅시다.

5. t.test(	data\$weight ~ data\$gender,	①
	mu=0,	②
	alternative="less",	③
	var.equal=TRUE )	④

## ▣ t.test()

- ① 성별로 몸무게가 결정됨을 나타내는 수식 '몸무게 ~ 성별'을 전달합니다.
- ② mu를 통해 전달되는 값은 영가설 상의 두 모집단의 평균의 차이를 나타냅니다.
  - 예제에서는 두 집단의 차이가 없는 ( $\mu_1 - \mu_2 = 0$ ) 것을 검정하는 것으로 0을 전달합니다.
- ③ alternative는 대안가설에 따라 결정됩니다. (양쪽검정, 왼쪽 한쪽 검정, 오른쪽 한쪽 검정)
  - 예에서는 대안가설이 위의 mu 값인 0보다 작은 경우이므로 "less"를 전달합니다.
  - gender의 값이 작은 집단 - 큰 집단으로 R이 결정합니다.
- ④ var.equal을 통해 분산의 동일성 여부를 전달합니다. TRUE이면 동일한 분산, FALSE이면 서로 다른 분산입니다.
  - 예에서는 분산의 동일성을 검정하여 동일한 분산으로 보았으므로 TRUE를 전달합니다.

# 서로 독립인 두 모집단 : 평균 차이검정

## Two Sample t-test

```
data: data$weight by data$gender
```

```
t = -1.5229, df = 42, p-value = 0.06764
```

```
alternative hypothesis: true difference in means is less than 0
```

```
95 percent confidence interval:
```

```
-Inf 25.37242
```

```
sample estimates:
```

```
mean in group 1 mean in group 2
```

```
3132.444
```

```
3375.308
```

### 판정

- 기각역을 이용한 판정
  - 영가설 하에서 검정통계량 -1.5229는 자유도가 42인 t-분포를 따르고
  - 대안가설을 통해 왼쪽 한쪽 검정임을 알 수 있으며, 이 때의 기각역은 ( $\infty \leq T \leq -1.682$ )로, 검정통계량 -1.523은 채택역에 있으므로 영가설을 채택합니다.
- 유의확률을 이용한 판정
  - 유의확률 0.068은 유의수준 0.05보다 크므로 영가설을 채택합니다.



# 서로 독립인 두 모집단 : 평균 차이검정

## ▣ 결론

- 남아 몸무게의 평균이 여아 몸무게의 평균보다 큰지를 알아보기 위해
- 표본 추출을 통해 여아 18명, 남아 26명의 몸무게를 측정한 결과,
  - 여아의 몸무게는  $3132.44 \pm 631.583(g)$ ,
  - 남아의 몸무게는  $3375.31 \pm 428.046(g)$으로 나타났습니다.
- 이를 유의수준 0.05에서 가설검정을 실시한 결과
- 검정통계량과 유의확률이 -1.523(p-value=0.0368)로 나타나,
- 남아 몸무게의 평균이 여아 몸무게의 평균보다 크다는 유의한 결론을 내릴 수 없었습니다.
- 즉, 남아의 몸무게의 평균은 여아의 몸무게의 평균보다 크지 않은 것으로 보입니다.  
(혹은 차이가 없는 것으로 판단됩니다.)

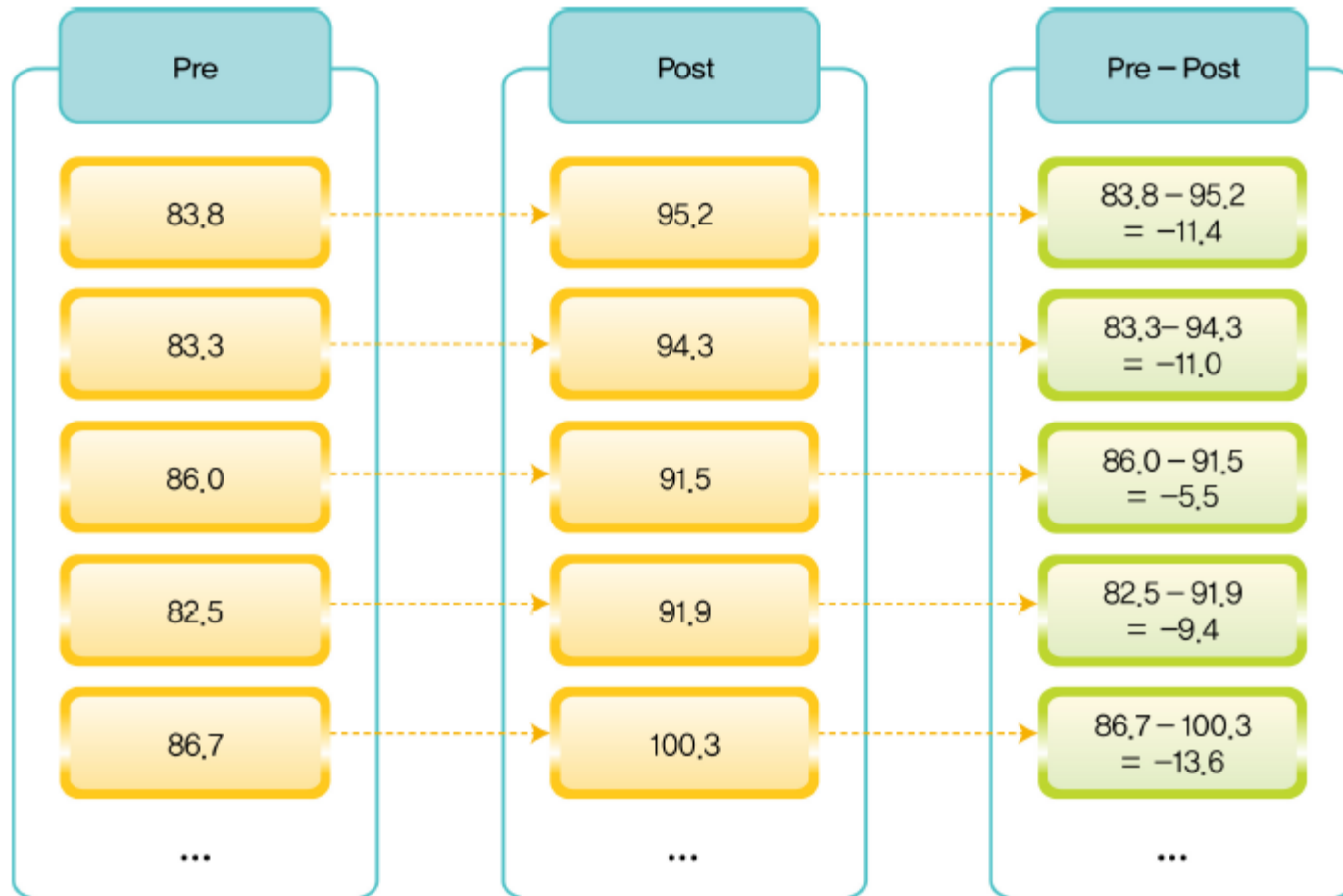
# 서로 대응인 두 모집단 : 평균 차이검정

- 서로 대응인 두 집단의 평균 차이에 대해 가설검정을 실시해봅시다.
  - 대응인 두 집단의 평균 비교는 동일한 관찰대상으로부터 처리 이전의 관찰과 처리 이후 관찰을 통해 처리가 어떠한 영향을 미쳤는지 밝히는 데 많이 사용 됩니다.
  - 예제 데이터) R package 중 하나인 PairedData의 예제 데이터인 anorexia를 이용하여 만든 가상의 자료입니다.
    - 17명의 관찰대상으로부터 식욕부진증 치료제 투여 이전의 사전관찰(Pre) 및 투여 이후 사후관찰(Post)을 변수로 저장한 자료입니다.

```
> str( data )
'data.frame': 17 obs. of 2 variables:
 $ Pre : num 83.8 83.3 86 82.5 86.7 ...
 $ Post : num 95.2 94.3 91.5 91.9 100.3 ...
```

- 자료 중 처음 5개를 예로 들어 어떻게 대응시키는지 확인해봅시다.

# 두 집단의 종류



- 각 관찰대상별로 “사전관찰 – 사후관찰”한 자료를 분석에 사용합니다.
  - 아래에서  $D = \text{사전관찰} - \text{사후관찰}$

# 두 집단의 종류

## 가설 수립

- 서로 대응인 두 집단의 평균 차이 검정에서 사용하는 대안가설은 다음과 같습니다.
  - 양쪽 검정은 차이가 있음을,
  - (왼쪽) 한쪽 검정은 차이의 평균이 0보다 작음을,
    - 즉, 처리로 인해 사후 관찰값이 줄어듦을 나타냅니다.
  - (오른쪽) 한쪽 검정은 차이의 평균이 0보다 큼을 대안가설로 합니다.
    - 즉, 처리로 인해 사후 관찰값이 증가함을 나타냅니다.

검정의 종류	영가설	대안가설
양쪽검정	$H_0 : \mu_D = 0$	$H_1 : \mu_D \neq 0$
(왼쪽) 한쪽검정	$H_0 : \mu_D \geq 0$ $H_0 : \mu_D = 0$	$H_1 : \mu_D < 0$
(오른쪽) 한쪽검정	$H_0 : \mu_D \leq 0$ $H_0 : \mu_D = 0$	$H_1 : \mu_D > 0$

# 두 집단의 종류

## 검정통계량

- 사전관찰과 사후관찰을 각각  $X_{pre}, X_{post}$  라 할 때 각 대응별로 사전관찰에서 사후관찰을 뺀  $D_i = X_{pre,i} - X_{post,i}$  는 평균이  $\mu_{X_{pre}-X_{post}} = \mu_D$  이고 분산이  $\sigma_D^2$  인 정규분포로부터 추출된 확률 표본이라 할 때 ,

n개의 확률표본  $D_1, D_2, D_3, \dots, D_n$ 의 표본평균  $\bar{D}$ 의 분포는  $N(\mu_D, \frac{\sigma_D^2}{n})$ 를 따름을 이용하여 모집단의 분산  $\sigma_D^2$ 의 인지 여부에 따라

① 모집단의 분산  $\sigma_D^2$ 를 알 경우 :  $Z = \frac{\bar{D} - \mu_D}{\sigma_D / \sqrt{n}} \sim N(0, 1^2)$

② 모집단의 분산  $\sigma_D^2$ 를 모를 경우 :  $T = \frac{\bar{D} - \mu_D}{S_D / \sqrt{n}} \sim t(n - 1)$

- $S_D$  는 대응표본의 차이  $D_i = X_{pre,i} - X_{post,i}$ 의 표준편차
- 모집단의 분산을 모르는 경우가 더 일반적인 경우로 모집단의 분산을 모르는 경우 대응표본의 평균 비교 검정을 실시해 봅시다.

# 두 집단의 종류

## 예제 2 식욕부진증 치료요법의 효과 검정

‘./data/01.anorexia.csv’의 자료는 17명의 여학생들로부터 신경성 식욕부진증의 치료 요법 시행 전(Prior)과 시행 후(Post)에 각각의 몸무게를 측정한 자료입니다. 주어진 자료에서 시행 전과 시행 후의 체중은 정규분포를 따른다고 할 때, 유의수준 0.05에서 신경성 식욕부진증의 치료요법이 효과가 있음을 검정해봅시다.

### 가설수립

- 영가설 : 신경성 식욕부진증 치료요법은 효과가 없다.

(효과가 없을 경우 체중의 변화는 없다)

$$H_0 : \mu_D \geq 0 \text{ or } \mu_D = 0$$

- 대안가설 : 신경성 식욕부진증 치료요법은 효과가 있다.

(효과가 있을 경우 체중이 증가하여, 사전-사후는 음수)

$$H_0 : \mu_D < 0$$

## 두 집단의 종류

### 검정통계량

- 검정통계량은  $n$ 개의 관찰대상으로부터 각 대응별 차이의 (표본)평균  $\bar{D}$  을, (표본)표준편차를  $S_D$  로 나타낼 때 영가설 하에서  $\mu_D = 0$  이므로,

$$T = \frac{\bar{D} - \mu_D}{S_D / \sqrt{n}} = \frac{\bar{D}}{S_D / \sqrt{n}} \sim t(n - 1)$$

- R을 이용한 검정통계량 계산 (04.paired.sample.R)

```
4. n <- length(data$Prior - data$Post)
5. m <- mean( data$Prior - data$Post )
6. s <- sd (data$Prior - data$Post)
7. ( t.t <- m / (s / sqrt(n)) )
```

- 4~6째줄 : 대응별 차이의 개수를 구해 변수  $n$ 에, 대응별 차이의 평균을 구해 변수  $m$ 에, 대응별차이의 표준편차를 변수  $s$ 에 저장합니다.
- 7째줄 : 위에서 구한 검정통계량을 변수  $t.t$ 에 저장하고 바로 출력합니다. 검정통계량은 약 -4.185 입니다.

## 두 집단의 종류

- R을 이용한 검정 : t.test()

```
9. t.test(      data$Prior, data$Post,      ①
               paired=TRUE,                ②
               alternative="less")          ③
```

- ① 첫번째 전달인자로 사전관찰값, 두번째 전달인자로 사후관찰값이 저장된 변수를 지정합니다.
- ② paired=TRUE 이면, 전달된 두 변수가 서로 대응인 두 표본으로 인식하여 검정을 실시합니다. (paired의 기본값은 FALSE로 대응인 두 표본의 평균 비교시 반드시 TRUE로 지정합니다.)
- ③ 대안가설이 영가설 하에서의 평균값이 0보다 작을 때 이므로 alternative에 “less”를 전달합니다.



## 두 집단의 종류

### Paired t-test

data: data\$Prior and data\$Post

**t = -4.1849, df = 16, p-value = 0.0003501**

alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:

**-Inf -4.233975**

sample estimates:

mean of the differences  
**-7.264706**

- 유의수준 0.05에서, 자유도가 16인 t-분포에서 (왼쪽) 한쪽검정의 기각역은  $-\infty \leq T \leq -1.746$  (임계값 -1.746)입니다.

#### ▣ 판정

- 기각역을 이용한 판정 : 검정통계량 -4.185는 기각역에 속하므로 영가설을 기각합니다.
- 유의확률을 이용한 판정 : 검정통계량으로부터 유의확률은 0.00035로 유의수준보다 작아 영가설을 기각합니다.

## 두 집단의 종류

### ▣ 결론

- 새롭게 개발한 신경성 식욕부진증 치료요법의 효과가 있는지 알아보기 위해
- 17명의 여학생을 대상으로
- 치료요법 시행 전 몸무게를 측정하고 시행 후 몸무게를 측정하여 차이를 구한 결과
- $7.265 \pm 7.157(\text{lbs})$ 로 나타났습니다.
- 이로부터 유의수준 0.05에서 검정통계량은 -4.185( $p\text{-value}=0.00035$  혹은  $p\text{-value} < 0.000$ )로 나타나
- "신경성 식욕부진증 치료요법은 효과가 있다."는 통계적으로 유의한 결론을 얻을 수 있었습니다.
- 즉 식욕부진증 치료요법은 효과가 있는 것으로 판단됩니다.



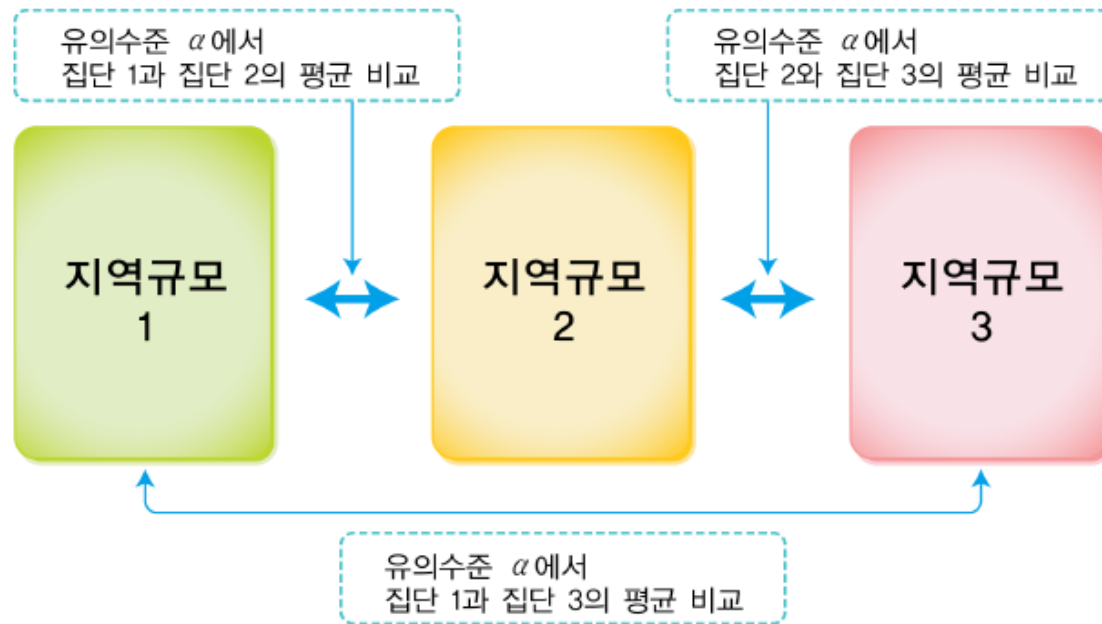
## 02.모집단이 세 개 이상

1. 일원분산분석에 대해 학습한다.
2. 분산분석표에 대해 학습한다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

## • 모집단이 세 개 이상일 경우

- 서로 독립인 두 모집단에서 모집단의 개수가 3개 이상으로 확장한 경우
- 다음 그림과 같이 모집단이 세 개일 때 독립인 두 모집단의 평균 비교를 2개씩 짝을 지어 비교하는 경우를 생각해 봅시다.



# 모집단이 세 개 이상일 경우의 평균 비교 검정

- ▣ 유의수준을 0.05로 하여 각각의 평균비교를 통해 차이가 발생하는 집단을 찾을 수 있을 것입니다.
- ▣ 하지만, 이렇게 사용할 경우 전체의 제 1종 오류를 범할 확률인 유의수준이 증가합니다.
- ▣ 유의수준을 그대로 유지하면서 검정할 다른 방법을 찾아봅시다.

## • 일원분산분석

- ▣ 집단을 구분하는 요인이 하나에 대해
  - 예) 앞선 그림에서 사용된 지역규모 : 집단을 구분하는 요인으로 3개의 수준(혹은 처리)를 가집니다.
  - 자료의 변동(분산)이 발생한 과정을 분석하여
  - 요인에 의한 변동(분산)과 요인을 통해 나누어진 각 집단 내의 변동(분산)을 구해
  - 요인에 의한 변동이 의미있는 크기를 가지는지를 검정합니다.
- ▣ 지역규모별 나이의 차이에 대한 평균 검정을 통해 일원분산분석의 과정을 알아봅시다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 일원분산분석을 위해 수집된 자료들은 다음과 같이 표현해 봅시다.

구분	지역규모=1 ( $y_{1j}$ )	지역규모=2 ( $y_{2j}$ )	지역규모=3 ( $y_{3j}$ )
집단별 관찰자료	$y_{11}$ $y_{12}$ $\vdots$ $y_{1n_1}$	$y_{21}$ $y_{22}$ $\vdots$ $y_{2n_2}$	$y_{31}$ $y_{32}$ $\vdots$ $y_{3n_3}$
평균	$\bar{y}_{1.}$	$\bar{y}_{2.}$	$\bar{y}_{3.}$

- 지역규모라는 요인에 의해 나뉜 각 집단은 처리 집단으로 각각은 정규분포로부터 추출된 표본임을 가정합니다.
  - 처리 1 :  $y_{11}, y_{12}, y_{13}, \dots, y_{1n_1} \sim N(\mu_1, \sigma_1^2)$
  - 처리 2 :  $y_{21}, y_{22}, y_{23}, \dots, y_{2n_2} \sim N(\mu_2, \sigma_2^2)$
  - 처리 3 :  $y_{31}, y_{32}, y_{33}, \dots, y_{3n_3} \sim N(\mu_3, \sigma_3^2)$
 (여기서  $\mu_1, \mu_2, \mu_3$ 는 i번째 처리의 모평균,  $\sigma_1^2, \sigma_2^2, \sigma_3^2$ 는 i번째 처리의 모분산)

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 각 집단의 분산은  $\sigma^2$  으로 동일하다는 가정을 도입하여, 각 처리별로 관찰값과 처리별 평균과의 차이는 서로 독립으로 평균이 0이고, 분산이  $\sigma^2$  인 정규 분포로부터 추출된 확률 표본입니다.

- 처리 1 :  $y_{1j} - \mu_1 \sim N(0, \sigma^2), \quad j = 1, 2, 3, \dots, n_1$

- 처리 2 :  $y_{2j} - \mu_2 \sim N(0, \sigma^2), \quad j = 1, 2, 3, \dots, n_2$

- 처리 3 :  $y_{3j} - \mu_3 \sim N(0, \sigma^2), \quad j = 1, 2, 3, \dots, n_3$

- 이를 정리하여 다음과 같이 나타낼 수 있습니다. (k는 처리의 수)

$$y_{ij} - \mu_i \sim N(0, \sigma^2)$$

$$i = 1, 2, 3, \dots, k, j = 1, 2, 3, \dots, n_i$$

- 위에서  $y_{ij} - \mu_i = \epsilon_{ij}$ 라 하면, (일원분산분석 모형)

$$y_{ij} = \mu_i + \epsilon_{ij}$$

$$i = 1, 2, 3, \dots, k, j = 1, 2, 3, \dots, n_i, \epsilon_{ij} \sim N(0, \sigma^2)$$

# 모집단이 세 개 이상일 경우의 평균 비교 검정

▣ 이로부터 다음을 생각해 봅시다.

- 전체 모집단의 평균  $\mu$ 를 각 처리별 평균들의 평균  $\mu = \frac{1}{k} \sum_{i=1}^k \mu_i$ 로 정의하고,  $i$ 번째 처리의 평균과 전체 평균의 차이  $(\mu_i - \mu)$ 를  $\alpha_i$ 라 하면 일원분산분석모형은 다음과 같이 나타낼 수 있습니다.

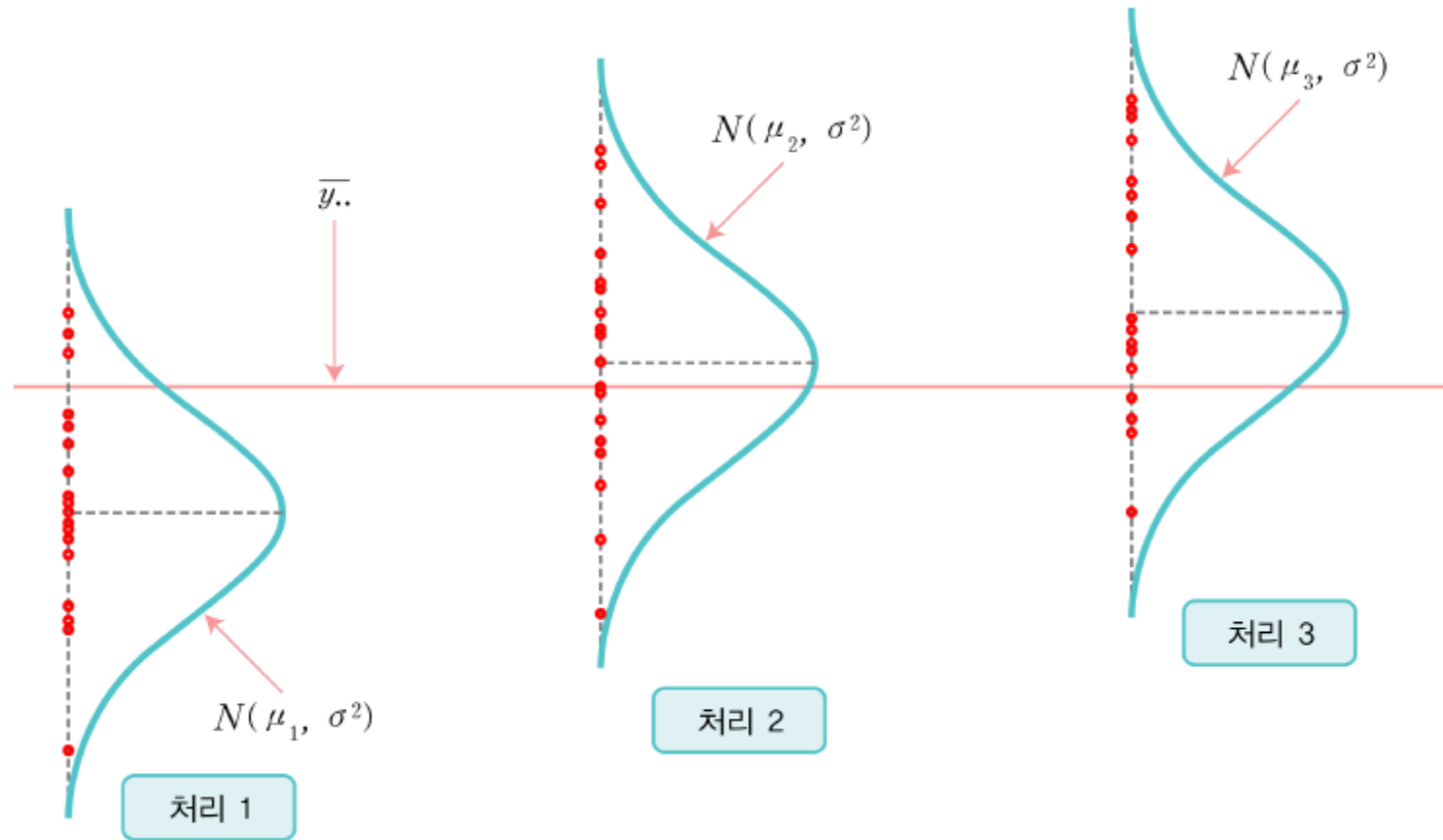
$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

$$i = 1, 2, 3, \dots, k, j = 1, 2, 3, \dots, n_i, \epsilon_{ij} \sim N(0, \sigma^2)$$

- 여기서  $\alpha_i$ 를 ‘ $i$ 번째 처리효과’라고 합니다.
- 만일 모든  $i$ 에 대해  $\alpha_i = 0$ 이면 (모든  $i$ 번째 처리의 평균과 전체 모집단의 평균인  $\mu$ 와 차이가 없을 경우) 처리의 효과가 없음을 의미합니다. (영가설)
- 처리의 효과가 있다면, ‘적어도 하나’의  $i$ 번째 처리에 대해  $\alpha_i \neq 0$ 임을 의미합니다. (대안가설)



# 모집단이 세 개 이상일 경우의 평균 비교 검정



# 모집단이 세 개 이상일 경우의 평균 비교 검정

## 가설 수립

- 영가설 : "모든 처리의 평균이 (전체의 모평균과) 같다." 혹은 "각 처리의 효과는 없다."

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_k, \text{ 혹은 } \alpha_1 = \alpha_2 = \dots = \alpha_k = 0$$

- 예) 지역규모별로 응답자의 연령의 평균은 동일하다.
- 대안가설 : 평균의 차이가 있는 것으로 "적어도 한 개의 처리의 평균은 다르다." 혹은 "적어도 한 개의 처리는 효과가 있다."

$$H_1 : \text{적어도 하나의 } \mu_i \text{ 는 다르다, 혹은 적어도 하나의 } \alpha_i \neq 0 \text{ 이다.}$$

- 예) 지역규모별로 응답자의 연령의 평균은 차이가 있다. (적어도 한 집단의 평균은 차이가 난다)

## 검정통계량

- 전체의 변동을 처리에 의한 변동과 자연발생적인 변동으로 구분해 봅시다.
- 총편차 :  $y_{ij} - \bar{y}_{..}$  개별 자료( $y_{ij}$ )와 전체 평균( $\bar{y}_{..}$ )과의 차이입니다.
  - 총편차를 다음과 같이 나타낼 수 있습니다.

$$y_{ij} - \bar{y}_{..} = (y_{ij} - \bar{y}_{i.}) - (\bar{y}_{i.} - \bar{y}_{..})$$

- 여기서  $\bar{y}_{i.}$ 는 i번째 처리(집단)의 평균입니다.
- $(y_{ij} - \bar{y}_{i.})$  : 처리집단내의 개별값과 처리평균의 차이를 '처리내 편차'라고 합니다.
- $(\bar{y}_{i.} - \bar{y}_{..})$  : 각 처리의 평균과 전체 평균과의 차이를 '처리간 편차'라고 합니다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

## 총편차의 제곱합

- 총편차의 합을 통해 변동량의 총량을 구하고자 할 때 총편차 역시 편차이므로 그 합은 0이 됩니다.
- 제곱하여 총편차의 제곱합을 다음과 같이 구합니다.

$$\sum_{i=1}^k \sum_j^{n_i} (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^k \sum_j^{n_i} ((y_{ij} - \bar{y}_{i.}) - (\bar{y}_{i.} - \bar{y}_{..}))^2$$

- 위의 식을 풀어 전체의 변동량의 제곱합을 '처리내 편차'와 '처리간 편차'의 제곱합으로 나타낼 수 있습니다.

$$\sum_{i=1}^k \sum_j^{n_i} (y_{ij} - \bar{y}_{..})^2 = \sum_{i=1}^k \sum_j^{n_i} (y_{ij} - \bar{y}_{i.})^2 + \sum_{i=1}^k \sum_j^{n_i} (\bar{y}_{i.} - \bar{y}_{..})^2$$

총제곱합  
(SST)

오차제곱합  
(SSE)

처리제곱합  
(SSt)

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 전체 자료의 변동량을 처리내에서 발생하는 자연발생적인 변동량 (오차제곱합)과 요인의 처리 수준에 의한 변동량(처리제곱합)으로 나누어 보았습니다.
- 여기서 다음의 두가지를 생각해 봅시다.
  - 요인에 의한 효과가 크지 않다면, 전체 변동량 중 오차제곱합이 처리제곱합보다 클 것입니다.
  - 요인에 의한 효과가 크다면, 전체 변동량 중 처리제곱합이 오차제곱합보다 클 것입니다.
- 즉, 두 변동량 오차제곱합과 처리제곱합의 비를 이용하여 검정통계량을 구합니다.
- 두 제곱합을 비교하기 위해 각각의 자유도로 나눈 값을 사용합니다
  - 오차평균제곱합(MSE) :  $SSE/n-k$ 
    - 자유도 계산은 각 처리 집단별 자유도의 합으로  $\sum_{i=1}^k (n_i - 1) = n - k$  입니다.
  - 처리평균제곱합(MSt) :  $SSt/k-1$ 
    - 요인의 처리 집단의 수가 k개인 경우 k-1이 처리의 자유도입니다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 이제 두 변동량의 비를 검정통계량으로 사용합니다.

$$F = \frac{SS_t / (k - 1)}{SSE / (n - k)} = \frac{MSt}{MSE} \sim F(k - 1, n - k)$$

- ▣ R을 이용하여 각 값을 직접 구해 봅시다.
  - 오차평균제곱합, 처리평균제곱합을 직접 구하는 과정을 먼저 수행합니다.
  - 그 다음으로 R이 내장하고 있는 함수를 이용하여 분석을 수행해 봅시다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 직접 구해보기

- Step#1) 오차제곱합 구하기

```
7. y1 <- ad$age[ad$scale=="1"]
8. y2 <- ad$age[ad$scale=="2"]
9. y3 <- ad$age[ad$scale=="3"]

11. y1.mean <- mean( y1 )
12. y2.mean <- mean( y2 )
13. y3.mean <- mean( y3 )

15. sse.1 <- sum( (y1 - y1.mean)^2 )
16. sse.2 <- sum( (y2 - y2.mean)^2 )
17. sse.3 <- sum( (y3 - y3.mean)^2 )

19. (sse <- sse.1 + sse.2 + sse.3)
20. (dfe <- (length(y1)-1) + (length(y2)-1) + (length(y3)-1))
```

## 모집단이 세 개 이상일 경우의 평균 비교 검정

- 7-9 줄 : 각 집단별로 나누어 저장합니다.
- 11-13줄 : 각 집단별 평균을 구합니다.
- 15-17줄 : 각 집단별 편차 제곱합을 구합니다.
- 19줄 : 각 집단의 편차 제곱합을 모두 더해 sse에 저장하고 출력합니다.
- 20줄 : 각 집단의 자유도를 모두 더해 dfe에 저장하고 출력합니다.

```
> (sse <- sse.1 + sse.2 + sse.3)
[1] 30139.38
> (dfe <- (length(y1)-1) + (length(y2)-1) + (length(y3)-1))
[1] 147
```

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- ▣ Step# 2) 처리제곱합 구하기
  - 처리 제곱합은 다음과 같이 구합니다.

$$\sum_{i=1}^k \sum_j n_i (\bar{y}_{i.} - \bar{y}_{..})^2 = \sum_{i=1}^k n_i (\bar{y}_{i.} - \bar{y}_{..})^2$$

```
22. y <- mean(ad$age)
```

```
24. sst.1 <- length(y1) * sum((y1.mean - y)^2)
```

```
25. sst.2 <- length(y2) * sum((y2.mean - y)^2)
```

```
26. sst.3 <- length(y3) * sum((y3.mean - y)^2)
```

```
28. (sst <- sst.1 + sst.2 + sst.3)
```

```
29. (dft <- length( levels( ad$scale ) ) - 1)
```



# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 22줄 : 전체 평균을 구해 변수 y에 저장합니다.
- 24-26줄 : 각 처리별로 처리의 평균과 전체 평균과의 편차제곱합을 구하고 각 처리의 표본의 개수와 곱합니다.
- 28줄 : 각 처리별로 구한 값을 모두 더해 처리제곱합을 구해 변수 sst에 저장하고 출력합니다.
- 29줄 : (처리 집단의 수 - 1)로 처리간 제곱합의 자유도를 구하고 출력합니다.
- levels() 함수는 R에서 범주형 자료의 각 처리를 출력하는 함수로 levels() 함수의 출력 결과의 개수가 처리집단의 수가 됩니다. (8장을 위한 자료준비 참고)

```
> (sst <- sst.1 + sst.2 + sst.3)
[1] 150.0933
> (dft <- length( levels( ad$scale ) ) - 1)
[1] 2
```

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 전체의 편차제곱합을 오차제곱합과 처리제곱합으로 잘 나누었는지 확인해 봅시다.

```
31. ( tsq <- sum( (ad$age - y)^2 ) )
32. ( ss <- sst + sse )
```

- 31줄 : 총 편차제곱합을 구해 tsq에 저장하고 출력합니다.
- 32줄 : 위에서 구한 처리제곱합(sst)와 오차제곱합(sse)의 합을 ss에 저장하고 출력합니다.
  - tsq와 ss 가 동일합니다.

```
> ( tsq <- sum( (ad$age - y)^2 ) )
[1] 30289.47
> ( ss <- sst + sse )
[1] 30289.47
```

# 모집단이 세 개 이상일 경우의 평균 비교 검정

## ▣ Step #3) 검정통계량

```
34. mst <- sst / dft  
35. mse <- sse / dfe  
36. (f.t <- mst / mse)
```

- 34줄 : 처리평균제곱합을 구해 변수 mst에 저장합니다.
- 35줄 : 오차평균제곱합을 구해 변수 mse에 저장합니다.
- 36줄 : mst를 mse로 나눈 값을 f.t에 저장하고 출력합니다.

```
> (f.t <- mst / mse)  
[1] 0.3660281
```

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- Step #4) 판정을 위한 기각역과 유의확률 (유의수준 0.05)
  - 처리의 변동량이 오차의 변동량보다 큰지를 알아보는 검정으로 (오른쪽) 한쪽 검정입니다.
  - 다음과 같이 임계값을 구할 수 있습니다.

```
38. alpha <- 0.05  
39. (tol <- qf(1-alpha, 2, 147))
```

- 유의수준이 0.05일 때 qf() 함수를 이용해 임계값을 구한 후 변수 tol에 저장하고 출력합니다

```
> (tol <- qf(1-alpha, 2, 147))  
[1] 3.057621
```

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- 검정통계량으로부터 유의확률을 구해봅시다.

```
41. (p.value <- 1 - pf(f.t, 2, 147))
```

- 검정통계량이 저장된 f.t값 보다 클 확률을 구해 p.value에 저장하고 출력합니다.
  - 다음과 같이 약 0.694임을 알 수 있습니다.

```
> (p.value <- 1 - pf(f.t, 2, 147))
[1] 0.6941136
```

## 판정

- 기각역을 이용한 판정
  - 검정통계량 0.366은 기각역에 포함되지 않아 영가설을 채택합니다.
- 유의확률과 유의수준을 비교한 판정
  - 검정통계량으로부터 구한 유의확률은 0.694로 유의수준 0.05보다 크므로 영가설을 채택합니다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

- ▣ R 함수를 이용한 검정 : 분산분석표 구하기

```
56. ow <- lm(age~scale, data=ad)
57. anova(ow)
```

- 56줄 : `lm()` 함수를 이용하여 일원분산분석을 위한 모형을 구축합니다.
  - R에서 모형 구축은 '종속변수~ 독립변수'의 형태로 구축합니다.
  - 우리 자료에서는 지역규모에 따라 나이가 영향을 받는지를 확인하는 것으로 지역규모를 설명변수(독립변수), 나이를 반응변수(종속변수)로 합니다.
  - 각 변수가 데이터 프레임 `ad`에 있음을 알려주었기에(`data=ad`) '`age ~ scale`'로 표현합니다.
- 57줄 : 위에서 생성한 모형의 분산분석을 실시합니다.
  - 분산분석은 앞서 실시한 제곱합의 분해 과정입니다.
  - `anova()` 함수는 R에서 구축한 모형을 이용하여 '**분산분석표**'를 제시합니다.
  - 분산분석표는 제곱합 분해 시 구한 각 과정을 기록한 것으로 결론을 작성할 때 근거가 됩니다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

```
> anova(ow)
```

Analysis of Variance Table

Response: age

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
scale	2	150.1	75.047	0.366	0.6941
Residuals	147	30139.4	205.030		

검정통계량

유의확률

- 위에서 작성한 분산분석표를 조금 더 알아보시다.

# 모집단이 세 개 이상일 경우의 평균 비교 검정

구분	자유도	제곱합	평균 제곱합	$F$ 값	유의확률
처리	2	150.1	75.047	0.366	0.6941
오차	147	30139.4	205.030		

처리의 제곱합을 자유도로 나누면 처리의 평균제곱합이 됩니다.

처리의 평균제곱합을 오차의 평균제곱합으로 나누면  $F$  값이 됩니다.



# 모집단이 세 개 이상일 경우의 평균 비교 검정

## ▣ 결론

- 지역의 규모에 따라 나이의 평균에 차이가 있는지 확인해 보기 위해 규모별로 50명, 총 150명의 표본추출을 통해 확인한 결과,
  - 지역규모 1의 나이의 평균과 표준편차는  $45.94 \pm 14.46$ 세,
  - 지역규모 2의 나이의 평균과 표준편차는  $45.68 \pm 13.59$ 세,
  - 지역규모 3의 나이의 평균과 표준편차는  $47.92 \pm 14.88$ 세로 나타났습니다.
- 일원분산분석을 통해 검정한 결과,
  - 검정통계량 0.366, 유의확률 0.694로
  - 유의수준 0.05에서 통계적으로 유의한 차이를 보이지 않았습니다.
- 즉, 지역규모에 따라 나이의 평균은 차이가 나지 않는 것으로 볼 수 있습니다.

## 모집단이 세 개 이상일 경우의 평균 비교 검정

## 모집단이 세 개 이상일 경우의 평균 비교 검정



# 8장을 위한 준비

: 범주형 자료와 테이블

# 범주형 자료

- 범주형 자료

- 자료 값이 크기를 나타내기 위한 것이 아니라, (범주를 분류 또는 구분하는) 의미를 나타내는 자료입니다. (질적 자료)
  - 1장에서 통계에서 다루는 자료의 종류 중 명목형 자료와 순서형 자료가 여기에 해당합니다.
- `factor()` 함수
  - R에서 범주형 자료를 지정하는데 사용합니다.
  - 범주를 구성하는 각 하위요소를 수준(level)이라 부르고 수준으로 정한 문자열만 저장합니다.
  - 각 수준을 구분하는 이름은 label을 이용하여 변경하여 사용할 수 있습니다.
  - 순위형 자료로 저장하기 위해서는 `ordered = TRUE`로 지정합니다.

# 범주형 자료

- 문자열 자료와 R에서 범주형 자료

- factor() 함수는 문자열 값을 이용하여 각 범주를 구별하는 자료를 만드는 함수로 문자열 자료와 비슷하게 사용하지만,
- 범주를 구성하는 수준(level)이 정해져 있어 일반 문자열 자료처럼 어떤 문자열이나 사용하는 것이 아닌 수준에 맞는 문자열만 추가될 수 있습니다.
- 다음의 예로 이를 확인해 봅시다.

- names 에는 등장인물들의 이름을 저장합니다.
- gender는 성별로 '여자'일 경우 '1'로 '남자'일 경우 '2'로 저장합니다.

```
> names <- c("고길동", "둘리", "영희")  
> gender <- c("2", "2", "1")
```

- names와 gender는 각각 이름과 성별을 저장하는 문자열 자료입니다.

```
> gender [1] "2" "2" "1" "남자" > str(gender) chr [1:4] "2" "2" "1" "남자" > gender[5] <- "여자"
```

## 범주형 자료

- 새롭게 남자인 희동이를 위의 자료에 추가해 봅시다.
- 입력을 하는 시점에 gender 에 남자를 '2'로 입력한 것을 잊은 채 '남자'로 입력한 경우입니다.

```
> ( names <- c(names, "희동이") )  
[1] "고길동" "둘리" "영희" "희동이"  
> ( gender <- c(gender, "남자") )  
[1] "2" "2" "1" "남자"
```

- 이름은 어떠한 값이 와도 되지만,
- 성별은 남자를 나타내는 문자열 "2" 혹은 여자를 나타내는 문자열 "1" 중에 하나가 되어야 합니다.
  - 하지만 gender는 문자열로 구성된 벡터를 나타내는 변수이므로 어떠한 값이 들어와도 실행됩니다.
  - 단순 문자열 벡터로 구성된 자료는 어떤 문자열이 들어와도 되지만, 성별이라는 변수에는 어울리지 않습니다.

## 범주형 자료

- 일반 문자열 벡터에는 얼마든지 값에 제약을 받지 않고 넣을 수 있습니다.
- 위의 예에서는 5번째 값으로 “여자”라는 문자열을 아무 이상없이 넣었습니다.

```
> gender
[1] "2"      "2"      "1"      "남자"
> str(gender)
chr [1:4] "2" "2" "1" "남자"
> gender[5] <- "여자"
> gender
[1] "2"      "2"      "1"      "남자" "여자"
```

- 성별을 저장하는 문자열로 구성된 gender를 factor() 함수를 이용하여 ‘남자’와 ‘여자’의 두 범주를 갖는 범주형 자료로 구성해봅시다.



# 범주형 자료

- factor() 함수 사용

```

> f.gender <- factor(gender) ①
> f.gender ②
[1] 2      2      1      남자 여자
Levels: 1 2 남자 여자
> str(f.gender) ③
Factor w/ 4 levels "1","2","남자",...: 2 2 1 3 4
> levels(f.gender) ④
[1] "1"      "2"      "남자"   "여자"
> f.gender[6] <- "여" ⑤
Warning message:
In `[<- .factor`(`*tmp*`, 6, value = "여") :
  invalid factor level, NA generated
> f.gender ⑥
[1] 2      2      1      남자 여자 <NA>
Levels: 1 2 남자 여자

```

## 범주형 자료

- ① `factor()` 함수를 이용하여 기존 벡터로부터 factor형 자료(이후 범주형 자료는 R의 입장에서 factor형 자료라고 하겠습니다.)를 만듭니다.
- ② 기존 자료로부터 factor형 자료를 만들 시 기존 값으로부터 범주를 생성하여, 기존에 있던 네 개의 문자열 "1", "2", "남자", "여자"가 나뉘는 하나의 범주에 해당하는 수준(level)이 되도록 합니다.
- ③ `str()` 함수의 결과로 나온 'Factor w/ 4 levels'는 factor형 자료로 4개의 수준을 갖고 있음을 의미합니다.
- ④ 어떤 수준을 갖고 있는지 확인하기 위해 **`levels()` 함수**를 써서 factor형 자료의 수준을 확인합니다.
- ⑤ factor형 변수의 수준이 정해진 다음에는 임의의 값을 넣을 수 없습니다.
  - 일단 값을 입력하되 경고(warning message)를 보여줍니다.
  - 새롭게 추가한 값은 수준(level)으로 지정되지 않은 값임을 나타내고 있습니다.
- ⑥ 수준 외의 값이 추가되면 해당 값은 NA로 대체됩니다.

# 범주형 자료

- factor() 함수는 기존 값을 참고하여 수준을 지정합니다.
  - 기존에 값을 갖고 있지 않다면 수준으로 인식하지 못합니다..
- 예제) factor() 함수의 수준 판별
  - 만족도에 대해 1부터 5까지 각각 "매우 불만족", "불만족", "보통이다", "만족", "매우 만족"을 나타낼 경우입니다.
  - 사용자의 응답을 보니 다음과 같이 2, 3, 4로만 구성되어 있다고 할 때, 이때의 factor() 함수를 적용하면 어떻게 되는지 확인해 봅시다.

```
> answer <- c(2, 2, 3, 2, 3, 4, 4, 4, 3, 4)
> f.answer <- factor(answer)
> str( f.answer )
Factor w/ 3 levels "2","3","4": 1 1 2 1 2 3 3 3 2 3
```

- 설문지 상에는 수준이 다섯 개가 있으나, 실제 응답에서 수준 중 일부가 빠져있어 factor() 함수를 바로 적용하면 위와 같이 자료에 있는 값들만 수준으로 처리합니다.

# 범주형 자료

- 수준을 직접 지정하여 만들기
  - factor() 함수 적용시 levels에 전달인자로 수준을 직접 지정할 수 있습니다.

```
> f.answer <- factor(answer, levels=c(1, 2, 3, 4, 5))
> str(f.answer)
Factor w/ 5 levels "1","2","3","4",...: 2 2 3 2 3 4 4 4 3 4
```

- levels에 위와 같이 우리가 사용하고자 하는 수준이 들어있는 vector를 전달합니다.
- 순서있는 범주형 자료 만들기
  - 순서형 자료일 경우에는 levels에 전달하는 벡터를 그 순서대로 정의하고
  - ordered=TRUE를 전달하면 순서가 있는 factor형 자료가 됩니다.

```
> o.f.answer <- factor(answer, levels=c(1, 2, 3, 4, 5), ordered=TRUE)
> str(o.f.answer)
Ord.factor w/ 5 levels "1"<"2"<"3"<"4"<...: 2 2 3 2 3 4 4 4 3 4
> o.f.answer
[1] 2 2 3 2 3 4 4 4 3 4
Levels: 1 < 2 < 3 < 4 < 5
```

## 범주형 자료

- 수준의 이름을 변경하기
  - 앞서 수준을 나타내는 1, 2, 3, 4, 5는 각각 "매우 불만족", "불만족", "보통이다", "만족", "매우만족"을 나타내는 범주의 구별 기호입니다.
  - 1, 2, 3, 4, 5 대신 원래의 의미대로 값을 부여해봅시다.
  - 다음과 같이 labels 전달인자를 사용합니다.

```
> o.f.answer <- factor(answer, levels=c(1, 2, 3, 4, 5), ordered=TRUE,
+   labels=c("매우 불만족", "불만족", "보통이다", "만족", "매우만족"))
> str(o.f.answer)
Ord.factor w/ 5 levels "매우 불만족"<...: 2 2 3 2 3 4 4 4 3 4
> o.f.answer
[1] 불만족    불만족    보통이다 불만족    보통이다 만족      만족
[8] 만족      보통이다 만족
Levels: 매우 불만족 < 불만족 < 보통이다 < 만족 < 매우만족
```

- labels를 통해 수준(level)별로 보여지는 값을 변경하기 위해서는 다음처럼 levels에 전달되는 벡터의 위치를 서로 일치시켜 값을 지정한 벡터를 사용합니다.

## 범주형 자료

	levels=c(		labels=c(
1번째	1, .....	→	"매우 불만족",
2번째	2, .....	→	"불만족",
3번째	3, .....	→	"보통이다",
4번째	4, .....	→	"만족",
5번째	5, .....	→	"매우 만족"
	)		)

# 범주형 자료

## • 데이터 프레임과 factor

- 데이터 프레임 생성을 위해 함수 `data.frame()`을 사용하면 문자열 벡터의 경우 factor형 자료로 구성된 벡터로 변환하는 것을 기본으로 합니다.
- 다음 예를 봅시다.

```
> names <- c("고길동", "둘리", "영희")
> gender <- c("2", "2", "1")
> characters <- data.frame(name=names, gender=gender)
> str(characters)
'data.frame': 3 obs. of 2 variables:
 $ name : Factor w/ 3 levels "고길동","둘리",...: 1 2 3
 $ gender: Factor w/ 2 levels "1","2": 2 2 1
```

- 문제점을 찾아봅시다.
  - 이름은 범주형 자료가 아닌 일반 문자열로 구성된 자료가 되어야 하는데, factor가 되어버려 문자열이 갖고 있는 의미는 사라지고 범주를 구분하는 기호로만 사용됩니다
  - 성별을 저장하는 gender 역시 범주형 자료로 변환되는 것은 맞지만, 앞서 살펴본 것처럼 올바르게 범주가 설정될지는 모를 일입니다.

# 범주형 자료

- ▣ data.frame 생성시 문자열의 자동 변환 방지
  - 데이터 프레임 생성 시 자동으로 변환하지 않도록 stringsAsFactors 전달인자에 FALSE를 전달하여 문자열을 변환하지 않게 합니다.
  - 만일 factor형 자료로 사용할 경우라면, 앞서 살펴본 것처럼 읽어온 원본 자료를 바탕으로 추후에 새로운 factor형 열로 생성하는 것을 추천합니다.
  - 변환과정 이전 원본값은 유지하는 것을 추천하며, 처음부터 원본 데이터 프레임은 별도로 두고 원본의 복사본으로 작업을 하는 것도 좋은 방법입니다.

```
> characters <- data.frame(name=names, gender=gender,
+                           stringsAsFactors=FALSE)
> str(characters)
'data.frame':  3 obs. of  2 variables:
 $ name   : chr  "고길동" "둘리" "영희"
 $ gender: chr  "2" "2" "1"
```

①

②

- ① stringsAsFactors는 문자열을 factor형 자료로 변환할지를 정하는 역할을 합니다. 기본값으로 TRUE로 생략 시 TRUE가 됩니다. 자동변환을 방지하려면 FALSE를 전달합니다.
- ② stringsAsFactors를 FALSE로 할 경우 문자열 그대로 생성됩니다.



# 범주형 자료

```

> characters <- transform(characters,
+                           f.gender=factor(gender,
+                           levels=c("1", "2"), labels=c("여자", "남자"))) ①
> str(characters) ②
'data.frame': 3 obs. of 3 variables:
 $ name      : chr  "고길동" "둘리" "영희"
 $ gender    : chr  "2" "2" "1"
 $ f.gender: Factor w/ 2 levels "여자","남자": 2 2 1

```

- ① transform() 함수를 이용하여 기존에 문자열로 되어 있는 gender 열에서 "1"은 "여자"로, "2"는 "남자"로 표기하는 factor형 자료를 만들어 새로운 열 f.gender로 저장했습니다.
- ② 새롭게 생성된 f.gender 열은 factor형 자료로 "여자"와 "남자"의 두 수준을 갖습니다.

# 범주형 자료

## • 외부 파일 읽기와 factor

- data.frame() 함수를 이용하여 데이터 프레임 생성 시 factor형 자료 처리는 외부로부터 파일을 읽어오는 read.table(), read.csv() 함수에도 동일하게 적용됩니다. 다음 예제를 통해 외부 파일을 읽어온 결과를 살펴봅시다.

```

> sns <- read.csv("./data/snsbyage.csv", header=T) ①
> str( sns ) ②
'data.frame': 1439 obs. of  2 variables:
 $ age      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ service: Factor w/ 5 levels "C","E","F","K",...: 3 3 3 ...
> sns.c <- read.csv("./data/snsbyage.csv", header=T, ③
+                  stringsAsFactors=FALSE) ③
> str( sns.c ) ④
'data.frame': 1439 obs. of  2 variables:
 $ age      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ service: chr  "F" "F" "F" "F" ...

```

## 범주형 자료

- ① 위의 변수 `sns`는 `data` 디렉토리 아래에 있는 `snsbyage.csv`를 읽어온 파일로 두 개의
- ② `sns` 변수는 데이터 프레임으로 열 `age`와 `service`로 구성되어 있습니다.
  - `age`가 갖는 값 중 1은 20대, 2는 30대, 3은 40대를 가리키는 범주형 자료이지만, 자료를 불러올 때 숫자형 자료로 판단하여 정수로 읽어왔습니다. 추후 `factor`형 변수로 변환할 것입니다.
  - 또한 `service`는 "F", "T", "K", "C", "E" 다섯 개의 범주를 갖는 값으로 R이 문자열 자료를 바로 `factor`형으로 변환했습니다.
- ③ `sns.c`는 `sns`와 동일하나 `read.csv()` 함수를 이용하여 파일을 읽어 읽어올 때 앞서 `data.frame()` 함수에서 사용한 `stringsAsFactors`의 전달인자를 `FALSE`로 하여 문자열을 R이 알아서 `factor`형으로 변환하는 것을 못하게 하였습니다.
- ④ `sns.c` 변수는 데이터 프레임으로 열 `age`와 `service`로 구성되어 있습니다.
  - `service`는 앞서 `sns`로 저장할 때와 다르게 `factor`로 변환되지 않고 문자열로 읽어왔습니다.

## 범주형 자료

- 두 개의 자료 중에 R이 자동으로 변환하지 않은 sns.c를 사용하여 다음 장에서 사용할 자료를 준비합니다.
  - 이로부터 age는 20대, 30대, 40대를 수준으로 하는 factor형 자료로,
  - service는 "F", "T", "K", "C", "E"의 순서로 순위를 갖는 factor형 자료로 변환해봅시다.

```
> sns.c <- transform(sns.c,
+                    age.c = factor(age, levels=c(1, 2, 3),
+                                   labels=c("20대", "30대", "40대")))
> sns.c <- transform(sns.c,
+                    service.c = factor(service,
+                                       levels=c("F", "T", "K", "C", "E"),
+                                       ordered=TRUE))
> str(sns.c)
'data.frame': 1439 obs. of 4 variables:
 $ age      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ service  : chr  "F" "F" "F" "F" ...
 $ age.c    : Factor w/ 3 levels "20대","30대",...:....
 $ service.c: Ord.factor w/ 5 levels "F"<"T"<"K"<"C"<...:....
```

# 범주형 자료를 요약하는 table

- **table() 함수를 이용하여 범주형 자료를 요약해 봅시다.**
  - table() 함수는 범주의 각 수준별로 몇 개의 자료가 있는지를 요약합니다.
  - 앞에서 읽어온 sns.c 데이터 프레임에서 age.c는 "20대", "30대", "40대"의 세 개 수준으로 이뤄진 범주형 자료입니다.

```

> age.c.tab <- table(sns.c$age.c) ①
> str(age.c.tab) ②
'table' int [1:3(1d)] 532 571 336
- attr(*, "dimnames")=List of 1
..$ : chr [1:3] "20대" "30대" "40대"
> age.c.tab ③
20대 30대 40대
532 571 336

```

- ① table() 함수로 전달되는 첫 번째 전달인자는 표로 구할 factor형 자료입니다.
  - 문자열 자료나 숫자형 자료도 개별 값별로 숫자를 구해줍니다.
- ② table() 함수를 사용하면 'table'형의 자료를 만들어주고, 각 수준의 이름이 각 셀의 이름처럼 사용됩니다.
- ③ 각 수준별 응답 수를 나타냅니다.

# 범주형 자료를 요약하는 table

- table 자료를 이용한 함수들

```
> margin.table(age.c.tab) ①
```

```
[1] 1439
```

```
> addmargins(age.c.tab) ②
```

```
20대 30대 40대 Sum
```

```
532 571 336 1439
```

```
> prop.table(age.c.tab) ③
```

```
20대 30대 40대
```

```
0.3697012 0.3968033 0.2334955
```

- margin.table() 함수는 전달되는 table형 자료의 수준별 응답 수의 합을 구합니다.
- addmargins() 함수는 전달되는 table형 자료에 margin.table()로 구한 합을 붙인 table을 생성해줍니다.
- prop.table() 함수는 전달인자로 table형 자료로부터, 비율표를 만들어줍니다.

# 범주형 자료를 요약하는 table

## • 두 범주형 자료의 요약

- factor형 변수 두 개를 이용하여 하나의 변수의 수준으로는 행을, 또 다른 변수의 수준으로는 열을 구성된 테이블을 작성해봅시다.

```
> c.tab <- table(sns.c$age.c, sns.c$service.c) ①
```

```
> str(c.tab) ②
```

```
'table' int [1:3, 1:5] 207 107 78 117 104 76 111 ...
```

```
- attr(*, "dimnames")=List of 2
```

```
..$ : chr [1:3] "20대" "30대" "40대"
```

```
..$ : chr [1:5] "F" "T" "K" "C" ...
```

```
> c.tab ③
```

	F	T	K	C	E
20대	207	117	111	81	16
30대	107	104	236	109	15
40대	78	76	133	32	17

# 범주형 자료를 요약하는 table

- ① 두 factor형 변수로 행과 열을 구성하는 table을 만들기 위해
  - table() 함수에 첫 번째 전달인자로는 행으로 수준을 구분할 변수를,
  - 두 번째 전달인자로는 열로 수준을 구분할 변수를 전달합니다.
- ② table형으로 구성되었으며, 행 이름으로는 첫 번째 전달인자의 수준이, 열 이름으로는 두 번째 전달인자의 수준이 사용되었음을 알 수 있습니다.
- ③ 만들어진 표입니다.

▣ margin() / addmargin() / prop.table() 을 구해 봅시다.

```
> margin.table(c.tab) ①
```

```
[1] 1439
```

```
> margin.table(c.tab, margin=1) ②
```

```
20대 30대 40대
```

```
532 571 336
```

```
> margin.table(c.tab, margin=2) ③
```

```
F T K C E
```

```
392 297 480 222 48
```



# 범주형 자료를 요약하는 table

> addmargins(c.tab)

④

	F	T	K	C	E	Sum
20대	207	117	111	81	16	532
30대	107	104	236	109	15	571
40대	78	76	133	32	17	336
Sum	392	297	480	222	48	1439

> addmargins(c.tab, margin=1)

⑤

	F	T	K	C	E
20대	207	117	111	81	16
30대	107	104	236	109	15
40대	78	76	133	32	17
Sum	392	297	480	222	48

> addmargins(c.tab, margin=2)

⑥

	F	T	K	C	E	Sum
20대	207	117	111	81	16	532
30대	107	104	236	109	15	571
40대	78	76	133	32	17	336

# 범주형 자료를 요약하는 table

```
> prop.table(c.tab)
```

	F	T	K	C	E
20대	0.14384990	0.08130646	0.07713690	0.05628909	0.01111883
30대	0.07435719	0.07227241	0.16400278	0.07574705	0.01042391
40대	0.05420431	0.05281445	0.09242530	0.02223767	0.01181376

```
> prop.table(c.tab, margin=1)
```

	F	T	K	C	E
20대	0.38909774	0.21992481	0.20864662	0.15225564	0.03007519
30대	0.18739054	0.18213660	0.41330998	0.19089317	0.02626970
40대	0.23214286	0.22619048	0.39583333	0.09523810	0.05059524

```
> prop.table(c.tab, margin=2)
```

	F	T	K	C	E
20대	0.5280612	0.3939394	0.2312500	0.3648649	0.3333333
30대	0.2729592	0.3501684	0.4916667	0.4909910	0.3125000
40대	0.1989796	0.2558923	0.2770833	0.1441441	0.3541667

## 범주형 자료를 요약하는 table

- 공통으로 사용되는 margin은 전달되는 값은 행 및 열별로 table 내의 연산을 합니다.
  - margin.table은 margin 값을 주지 않으면(margin=NULL) 전체의 합을 기준으로 처리하고, 1이면 행 방향, 2이면 열 방향으로 연산을 실시합니다.
  - addmargins 는 행 및 열별 합을 구해 table에 추가하는데 margin.table과 margin의 값에 따라 다른 합을 구하는데, 1일 경우 새로운 행을 생성해 그 합을 표현하는 것으로 행 방향으로 처리하는 것이라고 생각하면 좋을 것 같습니다.
  - prop.table의 경우 margin이 1이면, 행 내의 비율, 2이면, 열 내의 비율을 나타내고, margin 값을 지정하지 않으면 전체에서의 비율을 나타냅니다.

# 범주형 자료를 요약하는 table

- 조금 더 편리하게 table형을 만드는 xtabs()

- xtabs()는 변수 간 관계를 식으로 표현하여 table을 만듭니다.
- 먼저 단일 변수로 만들어 봅시다.

```

> xt.age <- xtabs(~age.c, data=sns.c) ①
> str(xt.age) ②
int [1:3(1d)] 532 571 336
- attr(*, "dimnames")=List of 1
..$ age.c: chr [1:3] "20대" "30대" "40대"
- attr(*, "class")= chr [1:2] "xtabs" "table"
- attr(*, "call")= language xtabs(formula = ~age.c, data =
sns.c)
> xt.age ③
age.c
20대 30대 40대
532 571 336

```

# 범주형 자료를 요약하는 table

① xtabs()에서 행과 열 등 차원을 지정하는 표현식은 '~ 변수명 1+변수명 2+...변수명 n'입니다.

- 예에서는 sns.c 데이터 프레임(data=sns.c)의 age.c 변수 하나에 대한 테이블을 작성하고 xt.age에 저장합니다.

② table()을 통해 만든 구조보다 몇몇 정보들이 들어가 있으며, 자료의 형태는 table형 이면서 xtabs() 함수로 만들어진 자료형임을 나타내고 있습니다.

③ xtabs()로 만들어진 자료는 행과 열을 설명하는 변수명(열 이름)도 함께 출력합니다.

▫ 다음으로 두 변수로 만들어 봅시다.

```
> xt.sns <- xtabs( ~ age.c+service.c, data=sns.c)
```

①

```
> xt.sns
```

②

	service.c				
age.c	F	T	K	C	E
20대	207	117	111	81	16
30대	107	104	236	109	15
40대	78	76	133	32	17

# 범주형 자료를 요약하는 table

① 두 개의 변수로 행과 열을 구성하는 table을 만듭니다.

- 틸드(~) 이 후 처음 나오는 변수가 행에 위치합니다.

② 변수 하나만 사용했을 때와 마찬가지로 차원을 설명하는 변수명(행과 열의 이름)도 함께 출력합니다.

## ▣ 요약된 자료의 테이블 구성

- 다음의 “./data/xtab.count.csv”는 이미 각 범주형 변수별로 몇 개가 해당하는지 요약된 자료입니다. 이 자료를 읽어 xtabs() 를 이용해 표를 만들어 봅시다.

- 자료에서 count 변수에 group별, result 별 개수가 들어가 있습니다.

```
> s.data <- read.csv("./data/xtab.count.csv", header=T)
> s.data
```

	group	result	count
1	treat	1	14
2	treat	0	16
3	test	1	20
4	test	0	10


# 범주형 자료를 요약하는 table

```

> xt.s.data <- xtabs(count ~ group+result, data=s.data) ①
> xt.s.data ②
      result
group    0   1
test   10  20
treat  16  14

```

- ① xtabs() 함수의 수식 표현에서 ~(틸드) 앞부분이 관찰수가 기록된 변수가 들어갑니다.
- 수식 'count~group+result'는 행에는 group별 수준이, 열에는 result별 수준이 들어가고, 각각 교차하는 셀의 관찰수가 변수 count에 있음을 의미합니다.
- ② 데이터를 읽어 만든 table입니다.



# Q & A



수고하셨습니다.