강원대학교
AI 소프트웨어학과

머신러닝1
- 기초통계(기술통계) -

KNU 강원대학교

01 기초통계

통계란?

통계란 무엇일까?

- 데이터 수집, 기술통계, 추론통계, 확률, 샘플링, 가설검정
- 데이터에서 유효한 결론을 도출하여 실제 문제를 해결하고 삶과 비즈니스의 다양한 측면을 개선하는 데 도움
- 통계의 모델의 적용을 통해 통찰력을 얻고 정보를 전달해 어떠한 문제의 결정을 내리는 것
- 통계는 데이터를 사용하여 결론을 도출하고 정보를 제공하여 문제를 해결하거나 다양한 현상에 대한 통찰력을 얻는 것

01 기초통계

통계란?

데이터란?

- 이론을 세우는 데 기초가 되는 사실, 또는 바탕이 되는 자료
- 관찰이나 실험, 조사로 얻은 사실이나 자료
- 컴퓨터가 처리할 수 있는 문자, 숫자, 소리, 그림 따위의 형태로 된 자료
- 데이터는 신호,기호,숫자,문자 등으로 기록 됨
- 정보를 위한 기초적인 자료를 말함
- 정보는 데이터를 가공하지 않은 경우

01 기초통계

통계란?

정보란 무엇일까?

- 정보란? → 구성, 해석 및 맥락화 과정을 통해 데이터에서 파생

선수들의 수치

PLAYER	DPM	GOLDDIFFAT15	분당 K+A	GPM	분당 골드 차이	분당 데미지 차이
FAKER	375	232.21	0.220	396	11.732	36.575
SHOWMAKER	488	82.86	0.294	404	19.901	31.411
CHOVY	466	352.44	0.223	410	27.059	-30.201
BDD	461	-1.41	0.269	389	4.989	53.180
GORI	459	-400.44	0.239	397	3.552	-29.395
FATE	472	236.74	0.238	406	21.573	-8.689

선수들의 신체 조건에 따른 적성 진단

운동선수-일반인 기초스포츠적성진단 비교

자료: 국민체육진흥공단 체육과학연구원

운동선수 (일반인 평균)	체지방률 (16.1%)	팔굽혀펴기 (60.6회)	농구공던지기 (661.1cm)	윗몸일으키기 (67.7회/2분)	제자리멀리뛰기 (224.7cm)	오래달리기 (465.5초/1600m)
역도	16.9	76.3	1010.7	66.3	285.0	480.9
야구	17.0	86.4	1004.2	88.0	268.2	357.3
축구	13.0	116.7	1025.6	97.2	258.8	311.0
핸드볼	12.2	108.6	1065.0	91.7	263.9	334.6
양궁	14.1	54.3	919.3	71.6	249.3	382.6
농구	18.2	61.7	1346.1	94.4	256.9	354.1
유도	15.2	118.5	926.1	82.7	246.7	347.8
수영	11.2	63.1	942.1	91.2	251.1	436.1
체조	8.8	65.2	745.3	96.1	233.9	459.6

통계란?

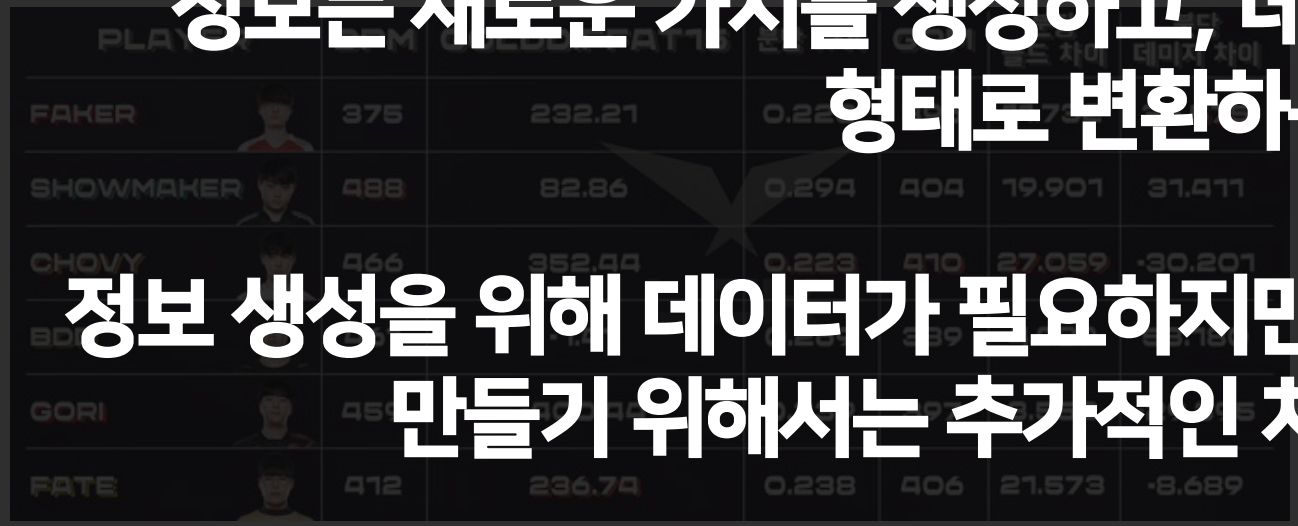
정보란 무엇일까?

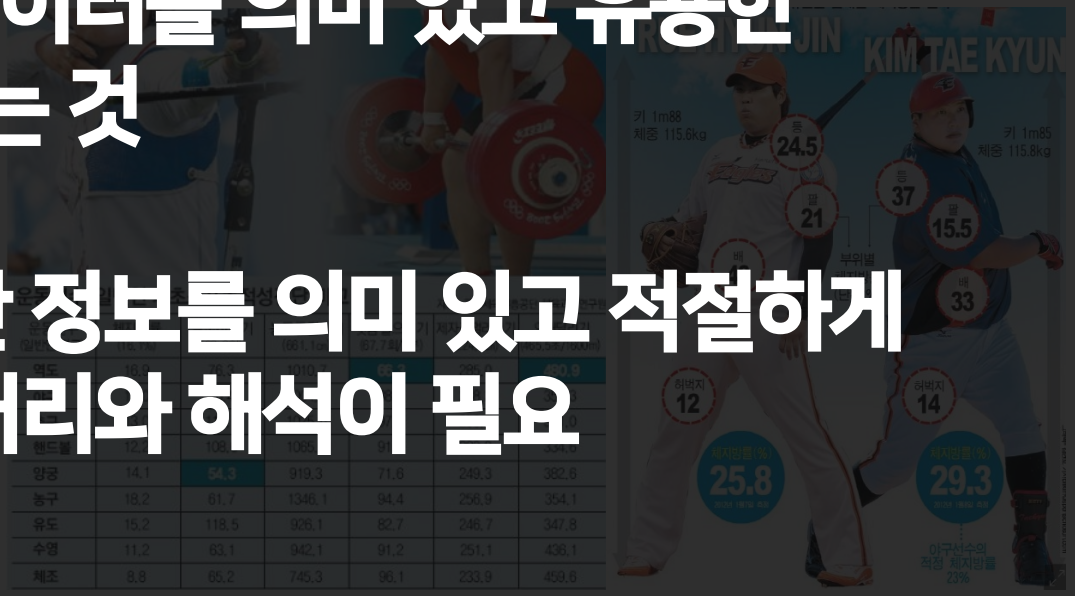
- 정보란? → 구성, 해석 및 [illegible]

데이터는 정보가 생성되는 원재료

정보는 새로운 가치를 생성하고, 데이터를 의미 있고 유용한 형태로 변환하는 것

정보 생성을 위해 데이터가 필요하지만 정보를 의미 있고 적절하게 만들기 위해서는 추가적인 처리와 해석이 필요





01 기초통계

통계란?

통계가 생기게 된 계기

- 경험을 토대로 같거나 비슷한 문제가 발생했을 때, 이를 해결하기 위해 사용됨
- 각자의 상황에 대해 경험하고, 학습해 결과를 도출할 수 있음 → 기억에는 한계가 존재하고, 왜곡이 가능함
- 이러한 문제를 해결하기 위해 기록이란 것이 생겼고, 다양한 방법론들이 나오게 됨 → 수를 추정하기 위해
- 고대 문명: 가장 초기 형태의 통계는 이집트, 메소포타미아, 중국과 같은 고대 문명에서 찾을 수 있으며, 그곳에서 데이터는 세금, 인구 수, 토지 조사와 같은 목적으로 수집

통계란?

통계가 생기게 된 계기

- 통계를 통해 문제를 해결하려면 일반적으로 하나 이상의 가설을 설정해야 함
- 명확하고 검증 가능한 가설을 세우는 것은 분석을 수행하고 데이터에서 의미 있는 결론을 도출하기 위한 프레임워크를 제공
- 가설에는 샘플 정보를 기반으로 모집단에 대한 추론 또는 결론 도출과 관련된 문제를 해결하기 위한 데이터가 필요함

통계란?

통계학이란

- 통계학의 대상인 모집단과 표본에 대해 설명하기 위해
- 표본을 추출하는 개념 및 방법에 대해 설명하기 위해
- 표본에 대한 가중치를 조정해 최적의 모집단을 예측하는 방법에 대해 이해하기 위해

통계란?

통계의 학습 프로세스

- 가설
  - 확률이론, 가설검증, 통계적 추론 등 통계적 방법의 기초가 되는 이론적 개념과 원리

- 데이터
  - 통계 분석을 위한 입력으로 사용되는 정보 또는 관찰의 수집
  - 데이터는 설문 조사, 실험, 관찰 또는 기타 방법을 통해 수집할 수 있음

01 기초통계

통계란?

통계의 학습 프로세스

- 모델
  - 일상속의 현상이나 프로세스의 수학적 또는 통계적 표현
  - 모델은 변수 간의 관계를 설명하고 데이터를 기반으로 예측하는 데 사용됨

- 결과
  - 데이터의 통계 분석에서 얻은 결과 또는 발견
  - 결과는 수행된 데이터 및 분석을 기반으로 통찰력, 패턴 또는 결론을 제공

- 정보제공
  - 통계는 정보에 입각한 결정을 내리고, 결론을 도출하고, 개체군이나 현상에 대한 가설이나 주장을 뒷받침하기 위해 데이터를 분석하고 해석하여 정보를 제공

01 기초통계

통계란?

통계의 학습 프로세스

- 가설을 세우고 이에 대한 데이터를 수집한 뒤 모델을 선정 후, 결과 도출 → 정보 제공

- 가설설정
- 초기 데이터 수집
- 데이터 전처리

- 평균
- 분산
- 표준편차 등등

- T-test
- ANOVA
- 카이제곱 검정 등등
- 가설설정

- 다양한 모델

- 모델에 대한 결과 값
- 결과 값 활용 정보 제공

01 기초통계

통계란?

통계의 학습 프로세스

- 가설을 세우고 이에 대한 데이터를 수집한 뒤 모델을 선정 후, 결과 도출 → 정보 제공

어떠한 문제를 해결하기 위해

분석 : 데이터를 수집하고, 모델을 선정해 결과를 도출하는 과정

가설검증 | 데이터 수집 | 모델선정 | 결과

정보 : 도출된 결과를 요약해 사용자에게 정보를 제공하는 것

- T-test
- ANOVA
- 카이제곱 검정 등등

- 누락된 데이터 정리
- 이상치 처리
- 데이터 정규화 등등

- 다양한 모델

- 모델에 대한 결과 값

기술통계

기술통계 : 보유하고 있는 데이터에 대한 요약 및 설명을 제공함

- **목적: 데이터 세트의 주요 기능을 요약하고 설명함**
- **측정 유형:중심 경향 측정: 평균, 중앙값 및 최빈값, 스프레드 측정: 범위, 분산, 표준 편차, 사분위수 범위.모양 측정: 왜도(비대칭 정도) 및 첨도(꼬리 정도)**
- **그래픽 표현: 히스토그램, 막대 차트, 박스 플롯 및 파이 차트**
- **사용법: 기술 통계는 당면한 데이터 이외의 모집단에 대한 결론을 도출하지 않고, 데이터의 주요 측면에 대한 간단한 개요를 제공함 (예: 기술 통계를 사용하여 설문 응답자의 연령 분포, 평균 소득 요약함)**

기술통계

데이터의 유형

- 변수란? → 데이터(data)를 저장하기 위해 프로그램에 의해 이름을 할당받은 메모리 공간

데이터 종류	변수명	내용	예시
범주형 데이터	명목형 변수	순서나 순위를 암시하지 않고 데이터를 범주화 하는 변수	성별, 머리 색깔, 과일 종류
	순위형 변수	의미 있는 순서가 있는 범주가 있지만 범주 간의 거리가 일정하지 않거나 알려져 있지 않은 변수	교육 수준, 설문 응답, 경제 수준
수치형 데이터	이산형 변수	고유하고 개별적인 값을 갖는 계산가능한 숫자 변수	교통사고 발생 수, 수상 인원의 수
	연속형 변수	주어진 범위 내에서 무한한 수의 값을 가질 수 있는 숫자 변수	몸무게, 키, 성적

기술통계

중심 경향 측정

- 평균 : 데이터 세트에 있는 모든 데이터 포인트의 산술 평균
- 중앙값 : 데이터 세트에서 가장 작은 것부터 큰 순서로 정렬할 때 중간 값
- 최빈값 : 데이터 세트에서 가장 자주 발생하는 값
- 최대값/최소값 : 데이터 세트에서 가장 큰 값/ 데이터 세트에서 가장 작은 값

```
data <- c(10, 15, 20, 25, 30, 15, 20, 25, 25, 10)
mean_value <- mean(data)
median_value <- median(data)
max_value <-max(data)
min_value <-min(data)
mode_value <- find_mode(data)
```

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)

반	학생수	평균
A	50	70
B	50	50
C	40	60
D	60	80

상품	가중치	상품의 크기
A	0.5	40
B	0.7	20
C	0.5	50
D	0.3	70

$$\frac{(50 \times 70) + (50 \times 50) + (40 \times 60) + (60 \times 80)}{50 + 50 + 40 + 60}$$

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)

년도	수익	증가율
2020	542	
2021	674	24.35
2022	841	24.78
2023	966	14.86
2024	1026	6.21
성장률		17.30%

$$CAGR(Compound\ Annual\ Growth\ Rate) = \sqrt[n-1]{\frac{x_n}{x_1}} - 1$$

$$CAGR = \sqrt[4]{\left(\frac{674}{542}\right)\left(\frac{841}{674}\right)\left(\frac{966}{841}\right)\left(\frac{1026}{966}\right)} - 1$$

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)

년도	수익	증가율
2020	542	
2021	674	24.35
2022	841	24.78
2023	966	14.86
2024	1026	6.21
성장률		17.30%

년도	수익	증가율
2020	542	
2021	248	-54.24
2022	841	239.11
2023	966	14.86
2024	1026	6.21
성장률		17.30%

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)
- 조화평균 : 비율, 속도, 효율성과 같은 자료→ 상호작용 또는 역수관계 가질 때 사용(Harmonic Mean)

$$HM = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

서울에서 강원도를(편도 : 300Km) 서울 → 강원(100km/h)
강원 → 서울(50km/h) 일 때, 왕복하는데 걸린 평균 시속은?

$$HM = \frac{300 + 300}{\frac{300}{100} + \frac{300}{50}} = \frac{600}{\frac{900}{100}} = 66.67km/h$$

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)
- 조화평균 : 비율, 속도, 효율성과 같은 자료→ 상호작용 또는 역수관계 가질 때 사용(Harmonic Mean)

$$HM = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

$$HM = \frac{100 + 50 + 50 + 40 + 70}{\frac{100}{10} + \frac{50}{20} + \frac{50}{40} + \frac{40}{30} + \frac{70}{50}} = \frac{310}{16.48} \approx 18.81$$

하나의 종목을 평균 얼마에 구매했는가?

가격	주식수
10$	100
20$	50
40$	50
30$	40
50$	70

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)
- 조화평균 : 비율, 속도, 효율성과 같은 자료→ 상호작용 또는 역수관계 가질 때 사용(Harmonic Mean)

$$HM = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

$$HM = \frac{5}{\frac{1}{0.1} + \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.75} + \frac{1}{0.71}} = 0.303$$

하나의 종목을 평균 얼마에 구매했는가?

가격	주식수
0.1$	1
0.4$	1
0.8$	1
0.75$	1
0.71$	1

기술통계

중심 경향 측정

- 산술평균 : 균등하게 나누고, 수치의 무게중심의 역할(Mean)
- 절단평균 : 최대, 최소값 중 K개를 제외한 평균 → 극단치가 있는 경우(Trimmed Mean)에 활용
- 가중평균 : 각 수치의 중요도에 비례하는 계수를 곱한 다음 산출하는 평균(Weighted Mean)
- 기하평균 : 곱의 형태로 변화하는 자료 → 비율의 평균계산에 많이 사용(Geometric mean)
- 조화평균 : 비율, 속도, 효율성과 같은 자료→ 상호작용 또는 역수관계 가질 때 사용(Harmonic Mean)

$$HM = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

역수의 특수성을 자연스럽게 반영함

단위 시간당 클리어 속도 : 28.15분 → 시간 대비 클리어 효율성(모든 단계가 동일한 작업량을 가짐)

단계별 평균 클리어 시간 : 52분(일반적인 클리어 시간)

하나의 스테이지의 평균 클리어 시간은?

게임 스테이지	클리어 시간(분)
1단계	10
2단계	30
3단계	50
4단계	70
5단계	100

기술통계

1) A반의 학생들이 중간시험에서 75, 88, 91, 68, 82점을 받았고, 기말시험은 60,87,55,47,92이다. 이때 A반의 중간시험과 기말시험의 각각 평균은?

2) A반의 점수에서 최상위와 최하위를 제외한 중간, 기말 평균을 각각 구하시오.

3) A반의 학생들의 전체성적을 계산할 경우 중간시험의 성적은 50%, 기말고사 성적은 30%로 계산해 최종성적을 각각 도출해 각 학생들의 등수를 나열하시오.

4) 한 회사의 수익은 초기 투자금액은 억이다. 이때, 1년차 : 5억, 2년차 : 5억5천, 3년차 7억, 4년차 10억이다. 이때 이 회사의 4년 평균 수익성장률은?

기술통계

5) 투자자가 여러 거래를 통해 다양한 가격으로 회사의 주식을 구매한다고 가정한다. 이때, 투자자가 지불한 종목 조화평균 가격은?

매입 주식 가격	매입 주식수
15$	80
25$	100
35$	60
45$	40
55$	20

기술통계

변동성 측정

- 범위: 데이터 세트의 최대값과 최소값의 차이
- 사분위수 범위(IQR): 데이터의 중간 50%를 나타내는 첫 번째 사분위수(25% 백분위수)와 세 번째 사분위수(75% 백분위수) 사이의 값 범위
- 사분위수(Q1) : 아래쪽 절반에 짝수 개의 관측치가 있는 경우 Q1은 이 절반의 가운데 두 숫자의 평균
- 중앙값(Q2) : 짝수인 경우 중앙값은 가운데 두 숫자의 평균
- 사분위수(Q3) : 위쪽 절반에 짝수 개의 관측치가 있는 경우 Q3은 이 절반의 가운데 두 숫자의 평균

기술통계

변동성 측정

- 범위: 데이터 세트의 최대값과 최소값의 차이
- 사분위수 범위(IQR): 데이터의 중간 50%를 나타내는 첫 번째 사분위수(25% 백분위수)와 세 번째 사분위수(75% 백분위수) 사이의 값 범위

```
data <- c(10, 15, 20, 25, 30, 15, 20, 25, 25, 10)
range_value <- max(data) - min(data)
df <- data.frame(values = c(5, 7, 10, 12, 14, 18, 20, 22, 25, 27, 30))
Q1 <- quantile(data, 0.25)
Q3 <- quantile(data, 0.75)
iqr_value <- Q3 - Q1
```

기술통계

변동성 측정

- 분산: 각 데이터 포인트와 평균 사이의 평균 제곱 차이
- 표준 편차: 데이터가 평균에서 얼마나 퍼져 있는지를 측정함

$$\text{Variance}(\sigma^2) = \frac{\sum(x_i - \mu)^2}{n}$$

$$\textit{Standard Deviation } (\sigma) = \sqrt{\sigma^2}$$

$$Mean(\mu) = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{n}$$

기술통계

변동성 측정

- 분산 또는 표준 편차가 높을수록 데이터 포인트가 더 분산되어 더 큰 변동성 또는 분산을 나타냄
- 낮은 분산 및 표준 편차는 데이터 포인트가 평균에 가깝다는 것을 의미하며 더 일관되고 예측 가능한 데이터 세트를 나타냄
- 반대로 높은 분산 및 표준 편차는 더 많은 변동성과 낮은 일관성을 나타냄
- 높은 분산 및 표준 편차는 데이터에 이상값이 있음을 나타내는 지표로 사용가능 함

```
data <- c(10, 15, 20, 25, 30, 15, 20, 25, 25, 10)
mean_data <- mean(data)
squared_diff <- (data - mean_data)^2
variance <- sum(squared_diff) / length(data)
std_dev <- sqrt(variance)
```

기술통계

변동성 측정

- 분산: 각 데이터 포인트와 평균 사이의 평균 제곱 차이
- 표준 편차: 데이터가 평균에서 얼마나 퍼져 있는지를 측정함

```
data <- c(10, 15, 20, 25, 30, 15, 20, 25, 25, 10)
sd_value <- sd(data)
var_value <- var(data)
```

기술통계

데이터의 활용

- 중심 극한을 이루는 수치형 데이터에 주로 사용됨
- 분산이 너무 크면 결과를 저해할 수 있음

이상값&결측값

- 이상값 : 이상값은 데이터 세트의 다른 관찰에서 크게 벗어나는 데이터 포인트
- 나머지 데이터를 고려할 때 예상할 수 있는 것과 현저하게 다른 값 → 데이터 수집 또는 기록의 잠재적 이상 또는 오류의 결과
- 결측값 : 사용자가 잘못 입력하거나 누락한 값

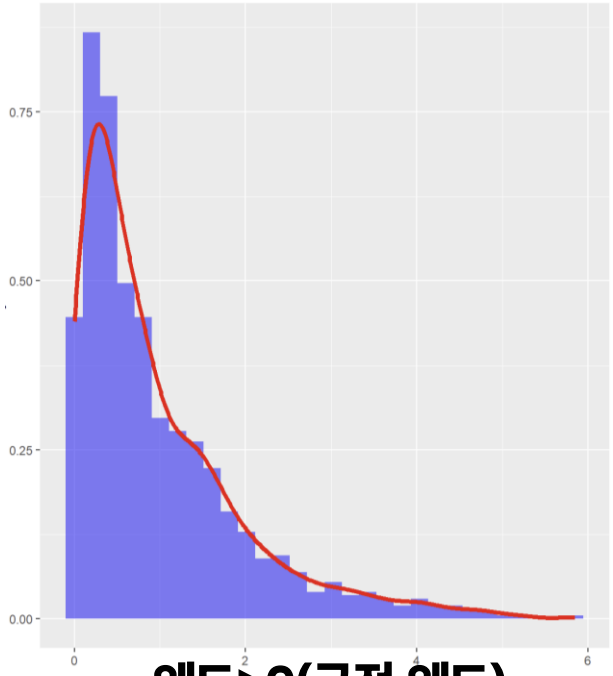
극단값/극한값

- 극단값/극한값 : 데이터 세트의 최소값과 최대값을 나타냄 → 분포의 양쪽 끝에서 가장 극단적인 값
- 일반적으로 오류나 비정상적인 상황으로 인한 결과임을 암시하는 증거가 없는 한 데이터 세트에 유지됨

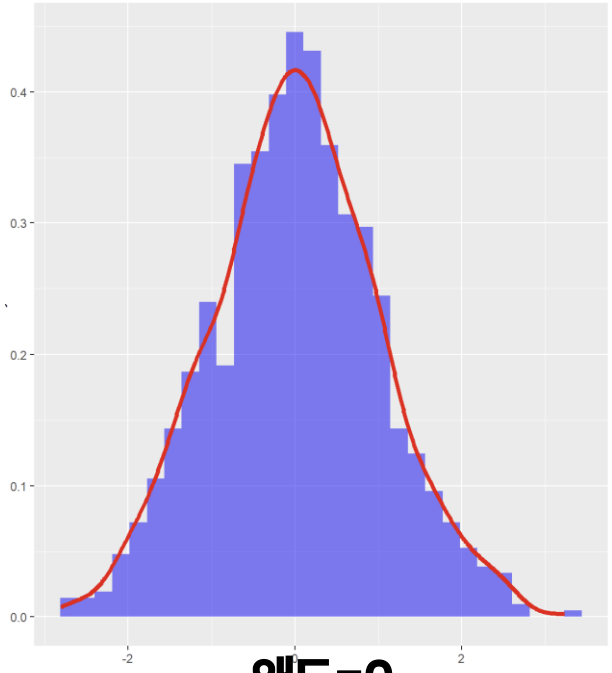
기술통계

분포의 특성

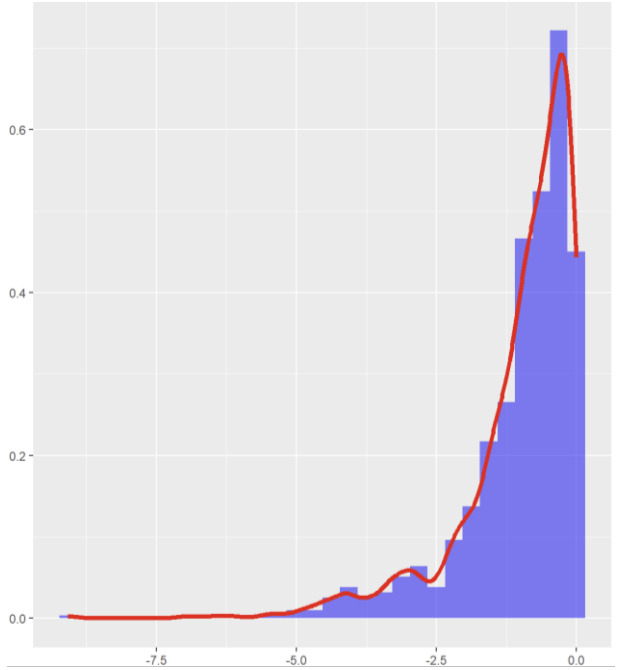
- 왜도(Skewness)는 확률 변수의 평균에 대한 확률 분포의 비대칭성을 측정함(즉, 데이터가 기울어지는 방향을 나타냄)
- 왜도>0 or 왜도<0 : 극단값 또는 이상치가 너무 많음



왜도>0(긍정 왜도)



왜도=0

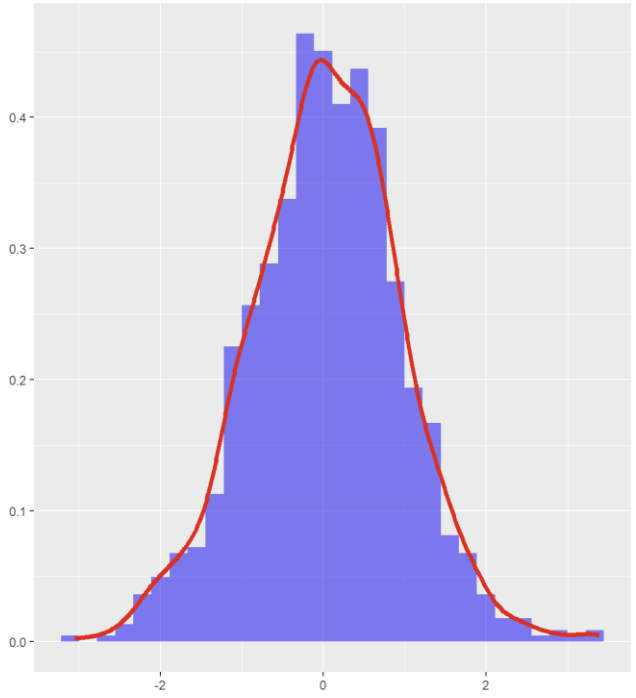


왜도<0(부정 왜도)

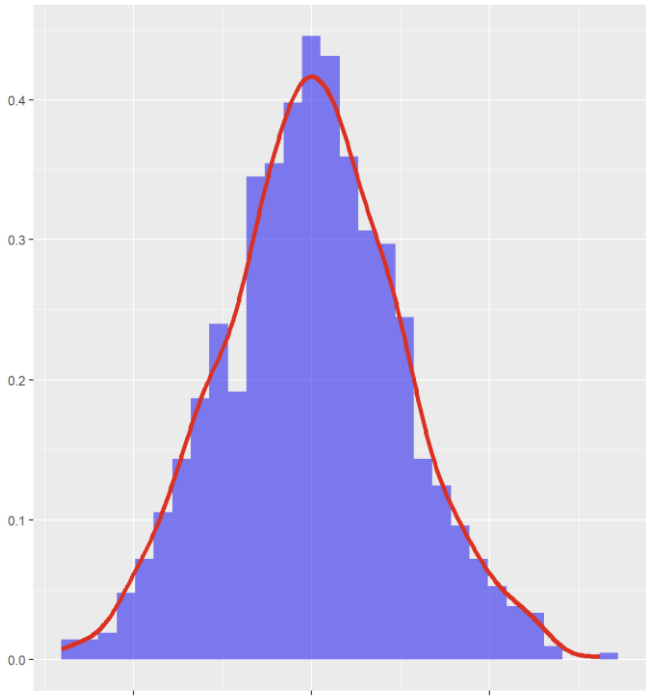
기술통계

분포의 특성

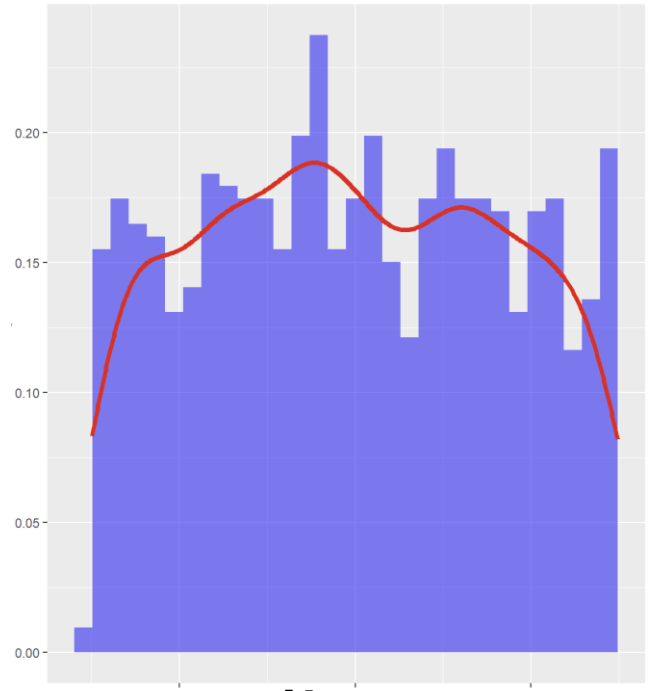
- 첨도(Kurtosis) : 분포의 꼬리 부분의 두께와 중심 부분의 뾰족함을 측정하는 값
- 첨도는 분포의 꼬리와 뾰족함 (즉, 극단 값들의 존재)에 관한 정보를 제공
- 첨도>0 : 이상치나 극단값이 나타날 가능성이 더 높아짐/ 첨도<0 : 데이터에 큰 변동이나 이상치가 덜 있음을 나타냄



첨도>0(꼬리 두꺼움)



첨도=0



첨도<0

리스트

선형구조-리스트

- 데이터 유형을 저장하고, 저장된 데이터들을 그룹화할 수 있는 데이터 구조
- 숫자, 문자, 논리값 … 등등 다양한 데이터 유형의 요소가 포함될 수 있음

```
List <- list(1, 2, 3)
[[1]]
[1] 1
[[2]]
[1] 2
[[3]]
[1] 3
```

```
List <-list(1.6, 2.3, 3.5)
List
[[1]]
[1] 1.6
[[2]]
[1] 2.3
[[3]]
[1] 3.5
```

```
typeof(List)
[1] "list“
```

```
mode(List)
[1] "list"
```

리스트

선형구조-리스트

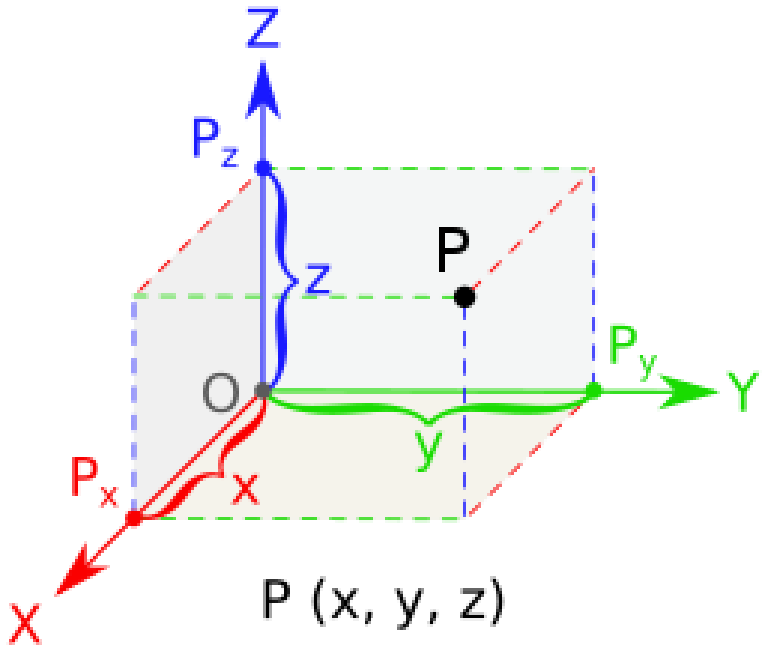
- 각각의 다른 데이터 형태를 모두 묶어서 그룹화할 수 있음

```
List <- list("apple", "banana", "orange", 1, 1.5, TRUE)

List
[[1]]                    [[4]]
[1] "apple"              [1] 1
[[2]]                    [[5]]
[1] "banana"             [1] 1.5
[[3]]                    [[6]]
[1] "orange"             [1] TRUE
```

벡터와 배열

물리에서 벡터란?

- 크기와 방향을 갖는 물리량
- 벡터는 사물의 움직임을 프로그래밍하기 위한 가장 기본적인 구성요소



00 데이터 구조와 함수 이해

벡터와 배열

프로그램에서 벡터란?

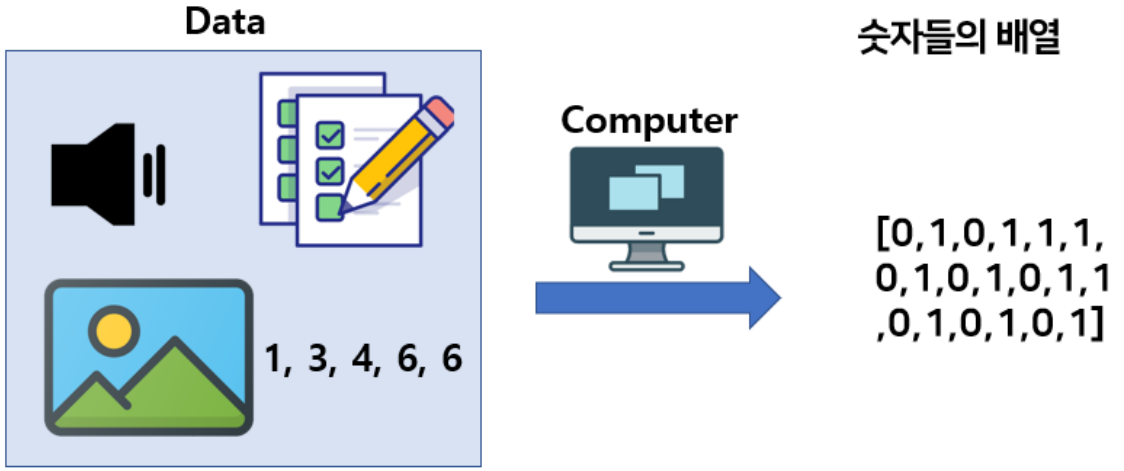
- 값을 저장하고, 조작할 수 있는 기본 데이터 구조
- 숫자, 문자 또는 논리 값과 같은 동일한 데이터 유형의 요소를 보유할 수 있는 1차원 배열
- R의 벡터는 combin을 나타내는 c( ) 함수를 사용하여 만들 수 있음

List(리스트)

- 자료를 순서대로 한 줄로 저장하는 자료구조
- 여러 자료가 일직선으로 서로 연결된 선형 구조(리스트에 있는 데이터는 몇 번째 인지 의미를 가짐)

Array(배열)

- 단일 타입으로 구성되는 자료구조



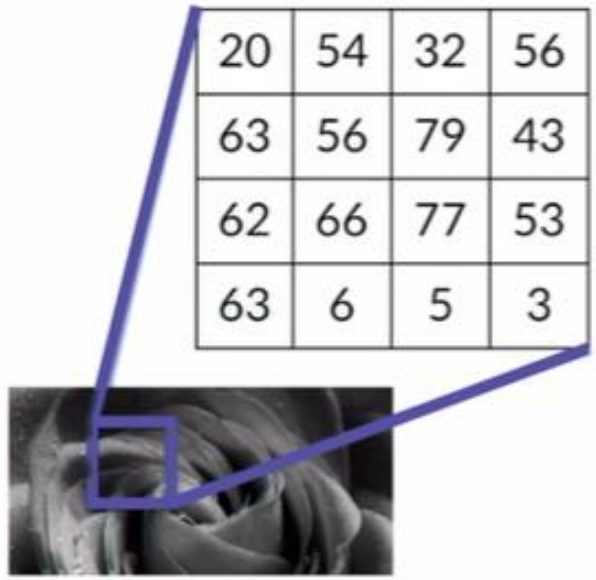
벡터와 배열

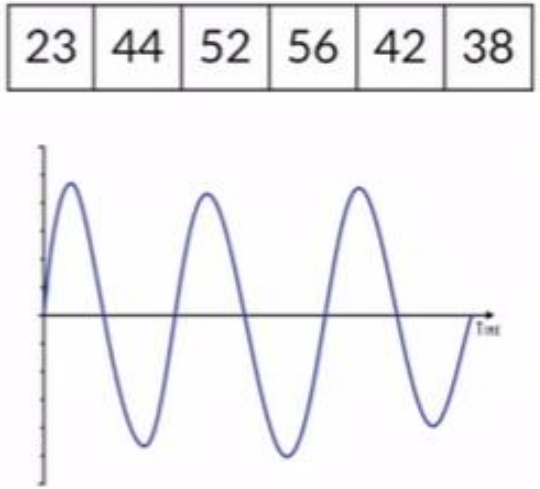
대규모 다차원 배열

- 데이터의 대부분은 숫자 배열로 볼 수 있음
- 흑백 이미지는 픽셀의 밝기와 명암을 2차원 배열로 표현할 수 있고 소리 같은 경우는 1차원 배열로 나타낼 수 있음

List(리스트)와 Array(배열)

- List는 [1,2,"Kim",2.5,True,False]와 같은 실수형, 정수형, 문자열과 같은 다양하게 관계없이 구성이 가능함
- array(배열)는 모두 단일 타입으로 구성됨





00 데이터 구조와 함수 이해

벡터와 배열

프로그램에서 벡터란?

- **값을 저장하고, 조작할 수 있는 기본 데이터 구조**
- **숫자, 문자 또는 논리 값과 같은 동일한 데이터 유형의 요소를 보유할 수 있는 1차원 배열**
- **R의 벡터는 combin을 나타내는 c( ) 함수를 사용하여 만들 수 있음**

```
Vector <- c(1, 2, 3)

Vector
[1] 1 2 3
```

```
logical_vector <- c(TRUE, FALSE, TRUE)

logical_vector
[1] TRUE FALSE TRUE
```

```
char_vector <- c("apple", "banana", "orange")

char_vector
[1] "apple" "banana" "orange"
```

벡터와 배열

프로그램에서 벡터란?

- **값을 저장하고, 조작할 수 있는 기본 데이터 구조**
- **숫자, 문자 또는 논리 값과 같은 동일한 데이터 유형의 요소를 보유할 수 있는 1차원 배열**
- **R의 벡터는 combin을 나타내는 c( ) 함수를 사용하여 만들 수 있음**

```
Vector <- c(1, 2, 3)

Vector
[1] 1 2 3

mode(Vector)
[1] "numeric"
```

```
List <- list(1, 2, 3)

[[1]]
[1] 1
[[2]]
[1] 2
[[3]]
[1] 3

typeof(List)
[1] "list“
```

벡터와 배열

프로그램에서 벡터란?

- **다양한 형태의 값들이 하나의 벡터에 들어갈 수 없음**
- **리스트 끼리는 연산이 불가능, 벡터 끼리는 연산이 가능**

```
Vector <- c("aa", 2, 3)

Vector
[1] "aa" "2" "3“

mode(Vector)
[1] "character"
```

```
List_1 <-list(1.6, 2.3, 3.5)
List_2 <-list(2.6, 5.3, 7.5)
List_1+List_2


Vector_1 <-c(1.6, 2.3, 3.5)
Vector_2 <-c(2.6, 5.3, 7.5) Vector_1+Vector_2
[1] 4.2 7.6 11.0
```

벡터와 배열

배열이란?

- **벡터의 확장된 개념**
- **벡터는 1차원 데이터를 나타내는 데이터의 기본 구조이지만 배열은 다차원 확장으로 표현**
- **행렬은 2차원 데이터만을 표현할 수 있지만 배열은 다차원 표현 가능**

```
array(data = c(1, 2, 3, 4, 5, 6), dim = c(2, 3))
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

```
matrix(data = c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

벡터와 배열

배열이란?

- **벡터의 확장된 개념**
- **벡터는 1차원 데이터를 나타내는 데이터의 기본 구조이지만 배열은 다차원 확장으로 표현**
- **행렬은 2차원 데이터만을 표현할 수 있지만 배열은 다차원 표현 가능**

```
array(data = c(1, 2, 3, 4, 5, 6), dim = c(2, 2, 2))
, , 1
[,1] [,2]
[1,] 1 3
[2,] 2 4

, , 2
[,1] [,2]
[1,] 5 1
[2,] 6 2
```

인덱싱

인덱싱이란?

- **목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정**
- **문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음**

substr(변수, 시작,끝)

abcdefg
1 2 3 4 5 6 7

순서

library(stringr)

a<-"abcdefg“

substr(a, 1,2)
[1] "ab"

인덱싱

리스트 인덱싱이란?

- **목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정**
- **문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음**

List(1,2,3,4,5,6,7)

1 2 3 4 5 6 7

순서

```
a<-list(1,2,3,4,5,6)

a[1]
[[1]]
[1] 1

a[[1]]
[1] 1
```

인덱싱

벡터 인덱싱이란?(오직 1차원의 형태만 가짐)

- **목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정**
- **문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음**

c(1,2,3,4,5,6,7)

1 2 3 4 5 6 7

순서

```
a<-c(1,2,3,4,5,6)
a[1]
[1] 1
```

인덱싱

배열 인덱싱

- 목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정
- 문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음

array(1,2,3,4,5,6,7)

1 2 3 4 5 6 7

a[행, 열]

```
a=array(data = c(1, 2, 3, 4, 5, 6), dim = c(2, 3))

a[2]
[1] 2

a[1,2]
[1] 3
```

인덱싱

배열 인덱싱

- 목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정
- 문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음

```
[,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
```

```
a=array(data = c(1, 2, 3, 4, 5, 6), dim = c(2, 3))

a[2]
[1] 2

a[1,2]
[1] 3

a[1:2]
[1] 1 2

a[1:3]
[1] 1 2 3

a[1:2,2]
[1] 3 4
```

인덱싱

배열 인덱싱

- 목록이나 배열에서 특정 값을 추출하기 위해 위치나 인덱스를 지정하는 과정
- 문자, 리스트, 행렬, 배열 모두 위치나 인덱스를 가지고 있음

```
[,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6
```

```
a[1:2,2:3]
[,1] [,2]
[1,]  3   5
[2,]  4   6
```

인덱싱

인덱싱 비교

```
my_list <- list("apple", 3.14, c(1, 2, 3), TRUE)
```

```
my_list[[3]][2]
[1] 2
```

my_list[3] → **여전히 리스트 형태를 유지하며, 해당 원소가 단독으로 반환**

```
[[1]]
[1] 1 2 3
```

my_list[[3]] → **원소의 값 자체가 반환되는 것이 아니라 값을 나타내는 데이터 타입으로 반환**

```
[1] 1 2 3
```

인덱싱

인덱싱 비교

```
my_vec <- c("apple", 3.14, c(1, 2, 3), TRUE)

my_vec[[3]][2]
[1] NA

my_vec[3]

[1] "1“

my_vec[[3]]
[1] "1"
```

인덱싱

인덱싱 비교

```
my_array <- array(c(5, 3, 1, 5, 7, 8, 10), dim = c(2, 3))
```

my_array[1, 2]: my_array 배열의 첫 번째 행, 두 번째 열에 접근합니다.

my_array[2, 1]: my_array 배열의 두 번째 행, 첫 번째 열에 접근합니다.

my_array[1,]: my_array 배열의 첫 번째 행에 접근합니다.

my_array[, 2]: my_array 배열의 두 번째 열에 접근합니다.

인덱싱

인덱싱 비교

```
my_array <- array(1:24, dim = c(3, 4, 2))

my_array[2, 3, 1]

my_array[3, , 2]

my_array[, 2:3, ]

my_array[2, 3, 1]

my_array[3, , 2]

my_array[, , 2]
```

인덱싱

인덱싱 비교

아래와 같은 배열을 만들고, 5와 16을 각각 인덱싱해 값을 추출하시오

```
, , 1 [,1] [,2] [,3] [,4] [,5]
[1,] 1 3 5 7 9
[2,] 2 4 6 8 10

, , 2 [,1] [,2] [,3] [,4] [,5]
[1,] 11 13 15 17 19
[2,] 12 14 16 18 20
```

데이터 구조 파악

데이터 프레임(Dataframe 이란?)

- 데이터 프레임은 프로그래밍 및 데이터 분석에 일반적으로 사용되는 표 형식의 데이터 구조
- 행과 열로 구성된 다양한 형태를 가지고 있는 리스트의 집합
- 데이터 프레임에서 각 열은 변수 또는 특정 속성을 나타냄
- 각 행은 개별 관찰 또는 데이터 포인트를 나타냄
- 데이터 프레임은 다목적이며 숫자, 범주 및 텍스트 데이터를 포함하여 다양한 유형의 데이터를 처리할 수 있음

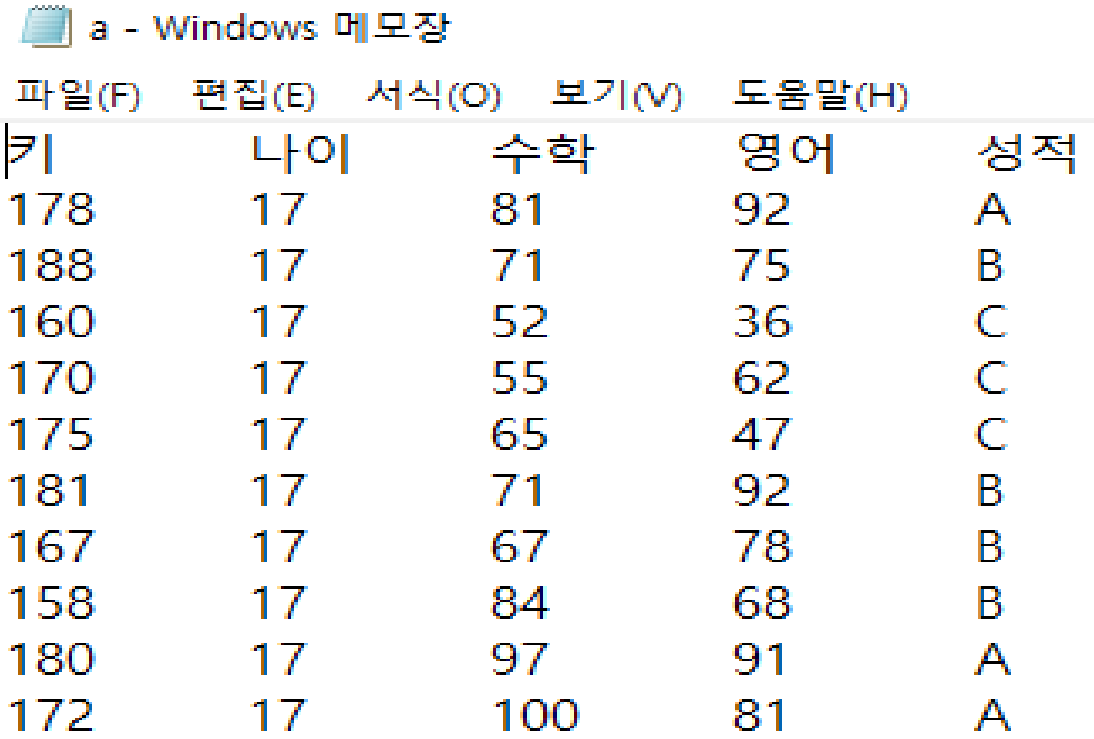
```
city <- c("Seoul", "Busan", "Daegu", "Seoul", "Busan", "Daegu", "Ulsan")
pm25 <- c(18, 21, 21, 17, 8, 11, 25)

df <- data.frame(city = city, pm25 = pm25)

write.csv(df, "저장할 경로", row.names=FALSE/TRUE)
```

00 데이터 구조와 함수 이해

데이터 구조 파악

TEXT

- 텍스트 파일은 데이터를 저장하고 표현하기 위해 간단하고 널리 사용되는 형식
- 텍스트 파일의 데이터는 일반적으로 각 데이터 포인트가 구분 기호(예: 쉼표 또는 탭)로 구분된 일반 텍스트로 저장됨
- 텍스트 파일의 주요 이점은 단순성과 다양한 프로그래밍 언어 및 소프트웨어 응용 프로그램과의 호환성이 좋음
- 복잡한 데이터 구조에 대한 지원 부족
- 데이터 조작 및 분석 기능이 제한됨
- 고급 데이터 작업을 위해 수동 처리가 필요함



데이터 구조 파악

Excel

- 특히 .xlsx 형식의 Excel 파일은 Microsoft Excel에서 만들고 사용하는 스프레드시트 파일
- Excel 파일은 데이터 구성, 조작, 시각화 및 분석을 위한 포괄적인 기능 세트를 제공함
- 복잡한 수식, 조건부 서식, 그래픽 표현 및 다양한 데이터 유형을 지원함
- Excel 파일은 사용자 친화적인 인터페이스와 광범위한 기능을 제공하여 기본 및 고급 데이터 관리 작업에 모두 적합함
- 그러나 Excel 파일은 크기가 상대적으로 큼
- 타사 소프트웨어와의 호환성 문제

	A	B	C	D	E
1	키	나이	수학	영어	성적
2	178	17	81	92	A
3	188	17	71	75	B
4	160	17	52	36	C
5	170	17	55	62	C
6	175	17	65	47	C
7	181	17	71	92	B
8	167	17	67	78	B
9	158	17	84	68	B
10	180	17	97	91	A
11	172	17	100	81	A

00 데이터 구조와 함수 이해

데이터 구조 파악

CSV(Comma-Separated Values)

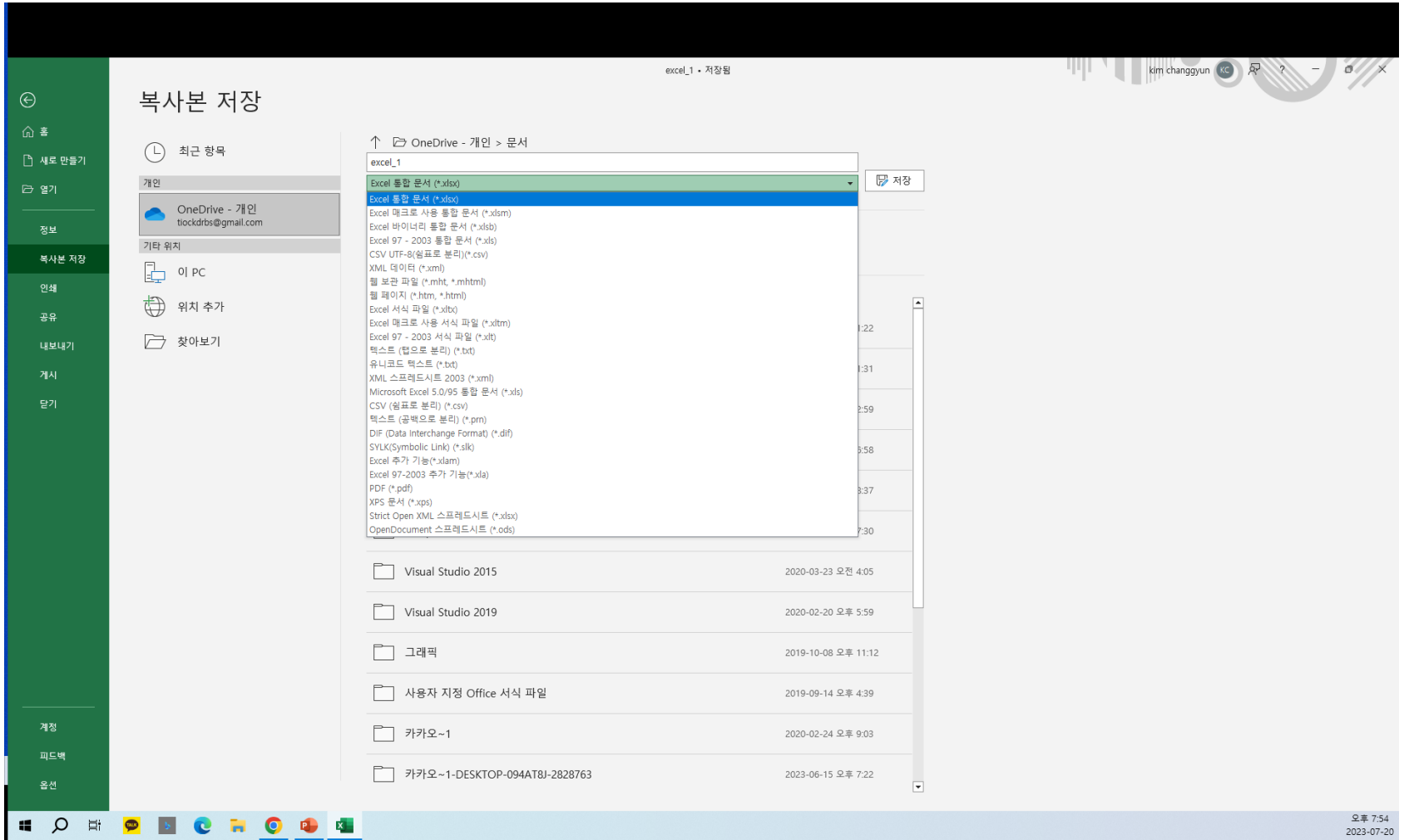
- 파일은 테이블 형식 데이터 저장 및 교환에 일반적으로 사용되는 특정 유형의 텍스트 파일 형식
- CSV 파일에서 각 행은 데이터 레코드를 나타내며 행 내의 각 필드는 쉼표 또는 기타 지정된 구분 기호로 구분됨
- CSV 파일은 스프레드시트 소프트웨어 및 데이터베이스 응용 프로그램에서 광범위하게 지원되므로 데이터 공유 및 상호 운용성을 위해 많이 사용 가능함
- 데이터를 행과 열로 구성하기 위한 기본 구조를 제공하지만 복잡한 수식이나 서식 옵션은 지원하지 않음
- 복잡한 데이터 구조 또는 수식에 대한 제한된 지원
- 고급 서식 옵션이 부족함

	A	B	C	D	E
1	키	나이	수학	영어	성적
2	178	17	81	92	A
3	188	17	71	75	B
4	160	17	52	36	C
5	170	17	55	62	C
6	175	17	65	47	C
7	181	17	71	92	B
8	167	17	67	78	B
9	158	17	84	68	B
10	180	17	97	91	A
11	172	17	100	81	A

데이터 구조 파악

Excel to CSV(Comma-Separated Values)

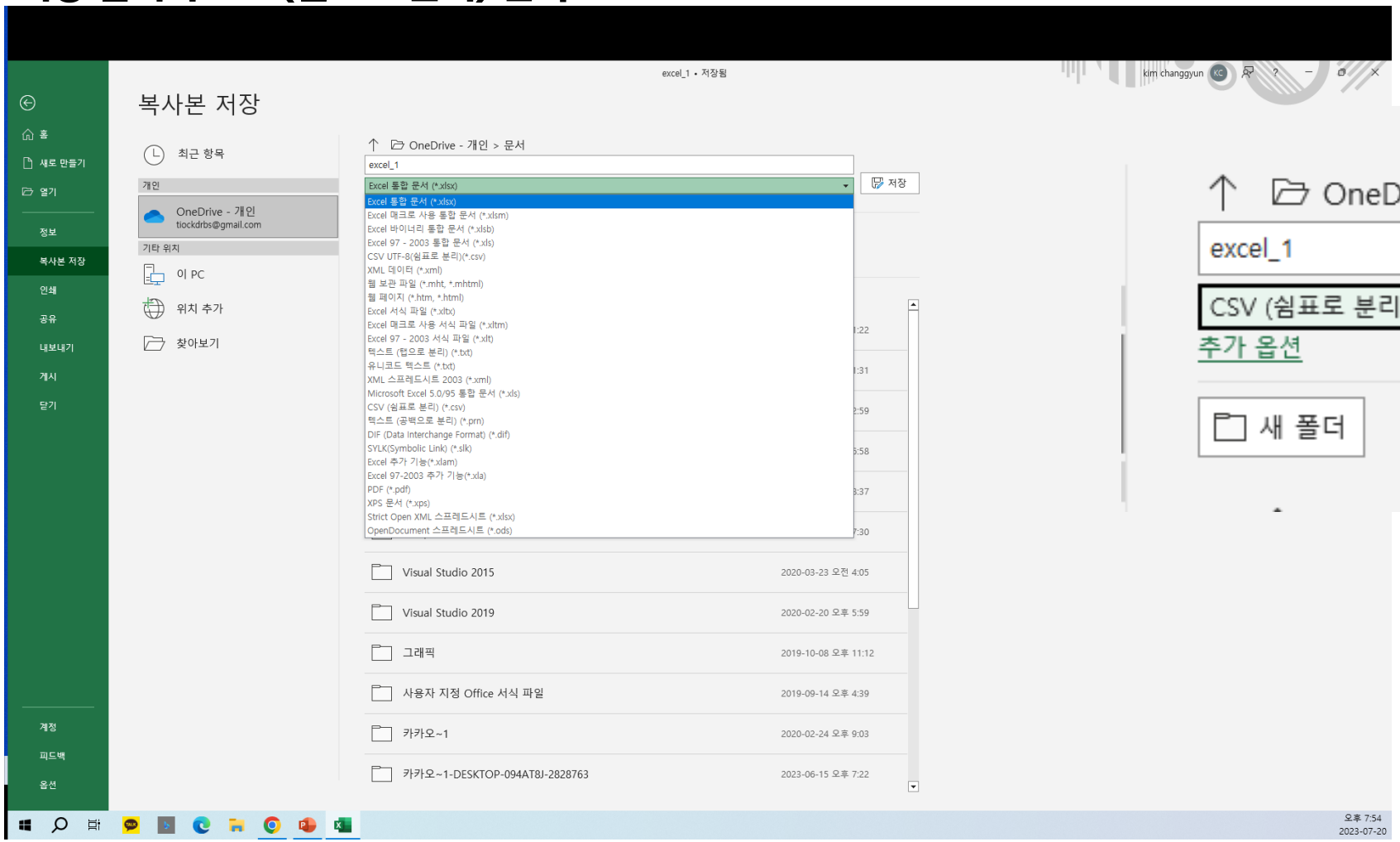
- 다른 이름으로 저장 or 복사본 저장



데이터 구조 파악

Excel to CSV(Comma-Separated Values)

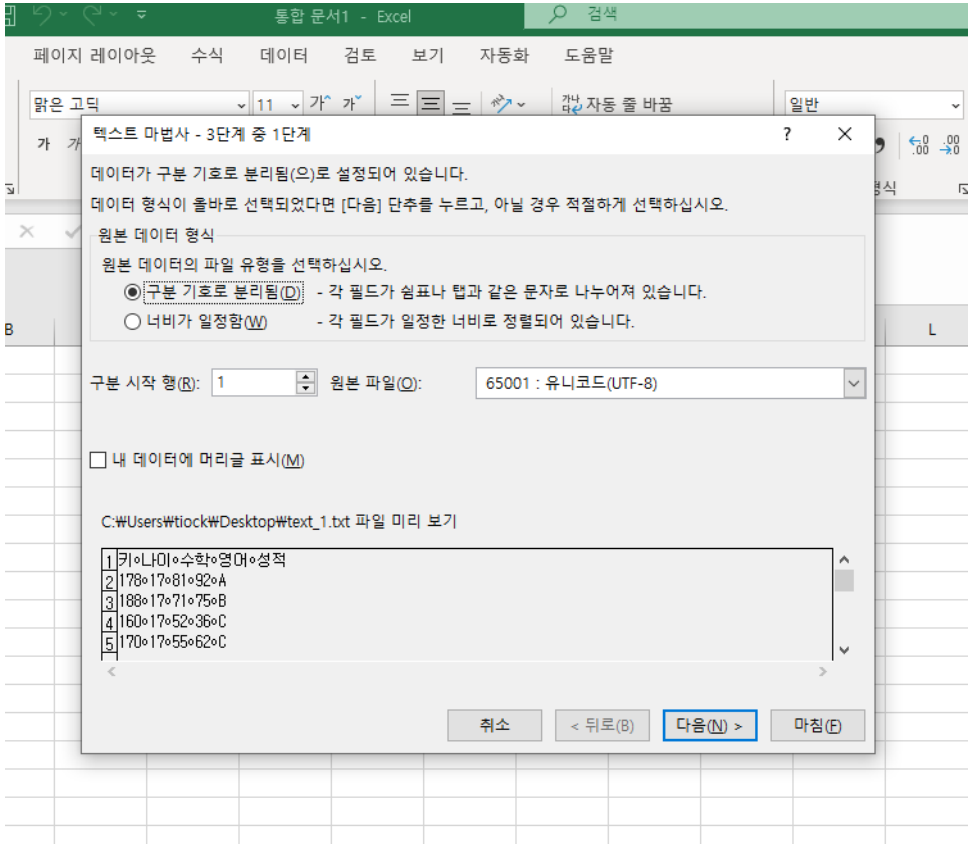
- **저장 탭에서 csv (쉼표로 분리) 선택**

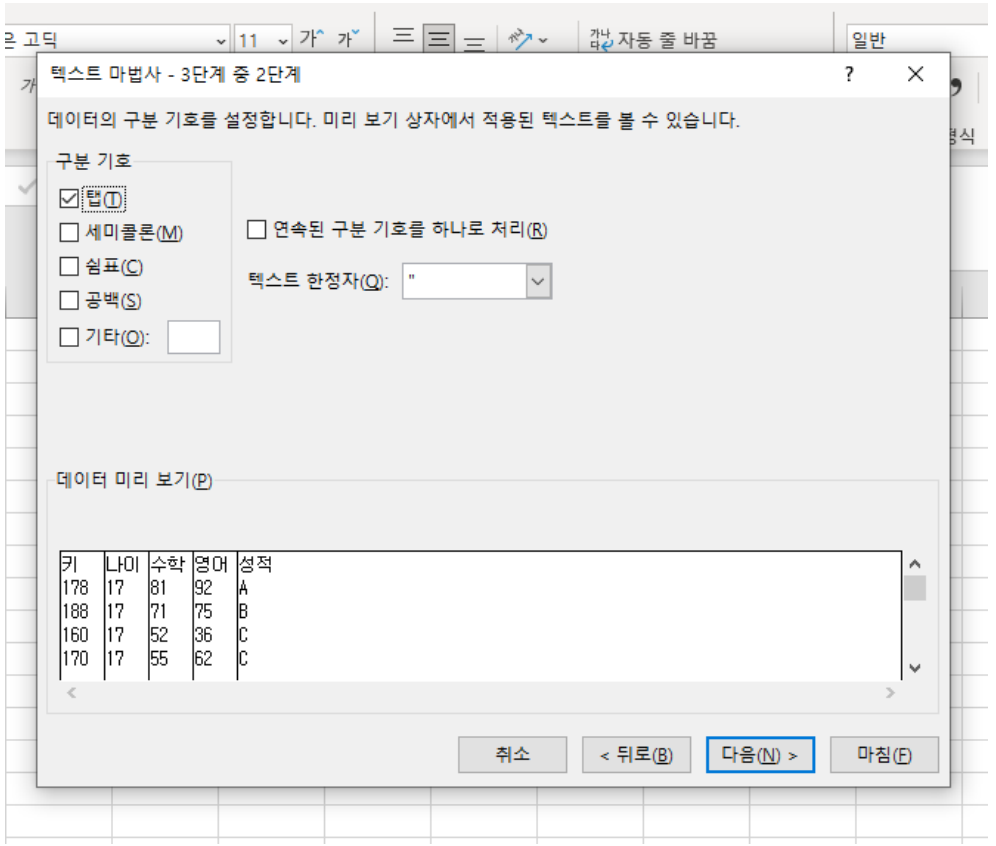


데이터 구조 파악

TEXT to CSV(Comma-Separated Values)

- **Excel → 열기 → Text파일 선택**

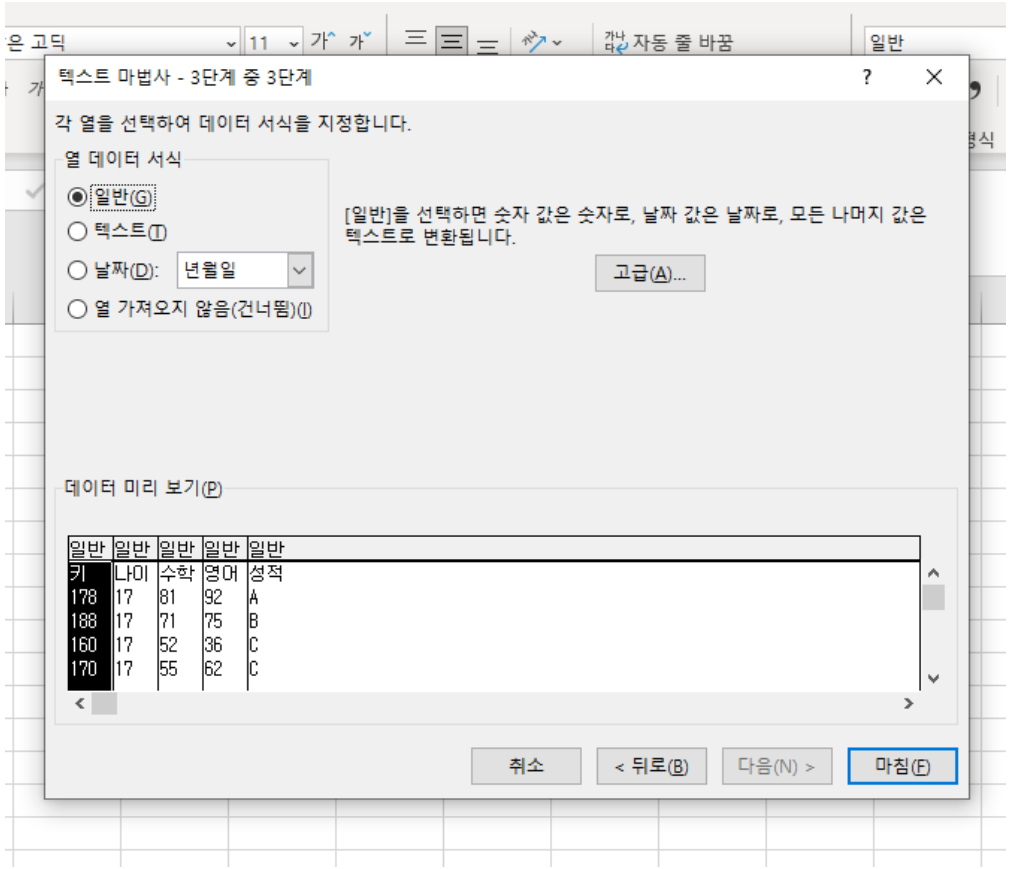




데이터 구조 파악

TEXT to CSV(Comma-Separated Values)

- **Excel → 열기 → Text파일 선택**



데이터 구조 파악

CSV(Comma-Separated Values)

```
data <- read.csv("data.csv", header =?, stringAsFactors =?, fileEncoding =?)
```

- header = 데이터 프레임의 첫 행에 변수명이 있는지 없는지에 따라 TRUE와 FALSE로 구분됨
- stringAsFactors = 문자열 변수를 요소로 변환할지 여부를 설정함 TRUE와 FALSE로 구분됨
- fileEncoding = R은 영어 기반으로 영어 이외의 단어들이 들어 갔을 때, 오류가 발생할 수 있으므로 영어 이외의 단어를 인식할 수 있도록 인코딩 하는 방식(주로 한국어는 cp949, euc-kr, utf-8)세 가지로 저장됨

데이터 구조 파악

- **Kaggle** : https://www.kaggle.com/
- **Google Dataset Search** : https://toolbox.google.com/datasetsearch
- **UCI Machine Learning Repository** : https://archive.ics.uci.edu/
- **Data.gov** : https://data.gov/
- **Naver DataLab** : https://datalab.naver.com/
- **Huggingface** : https://huggingface.co/spaces