



7주차: 데이터 시각화

ChulSoo Park

School of Computer Engineering & Information Technology

Korea National University of Transportation



학습목표 (7주차)

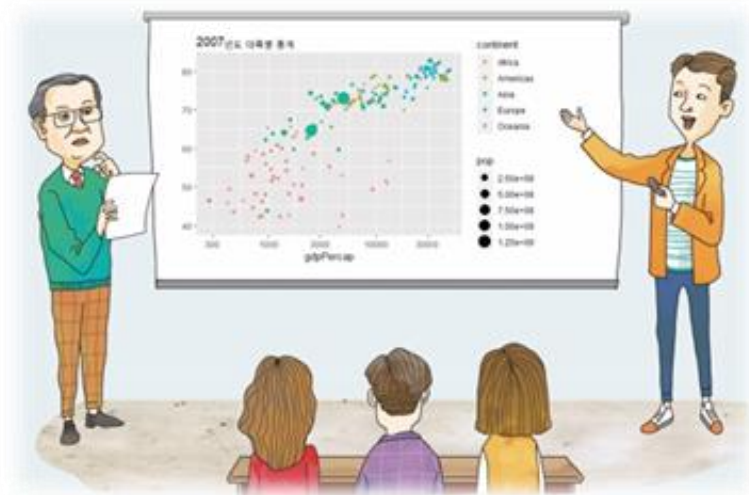
- ❖ 데이터 시각화 Library 사용법 이해
- ❖ 데이터 시각화 기본 기능 숙지
- ❖ 시각화를 이용한 데이터 해석
- ❖ 시각도 도구 숙지



06

CHAPTER

데이터 시각화



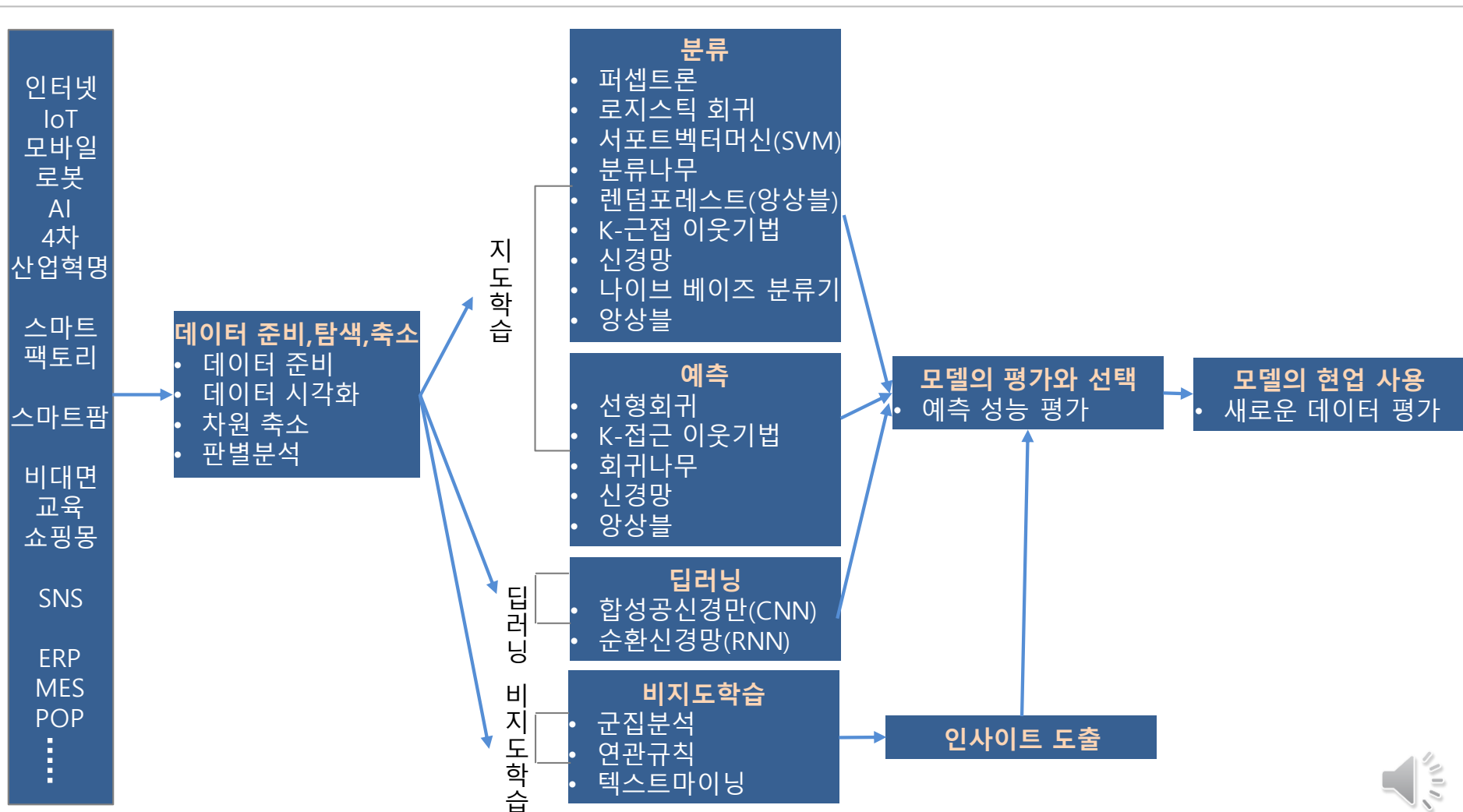
CONTENTS

- 6.1 데이터 시각화란?
- 6.2 시각화의 기본 기능
- 6.3 시각화 도구
- 6.4 시각화를 이용한 데이터 탐색

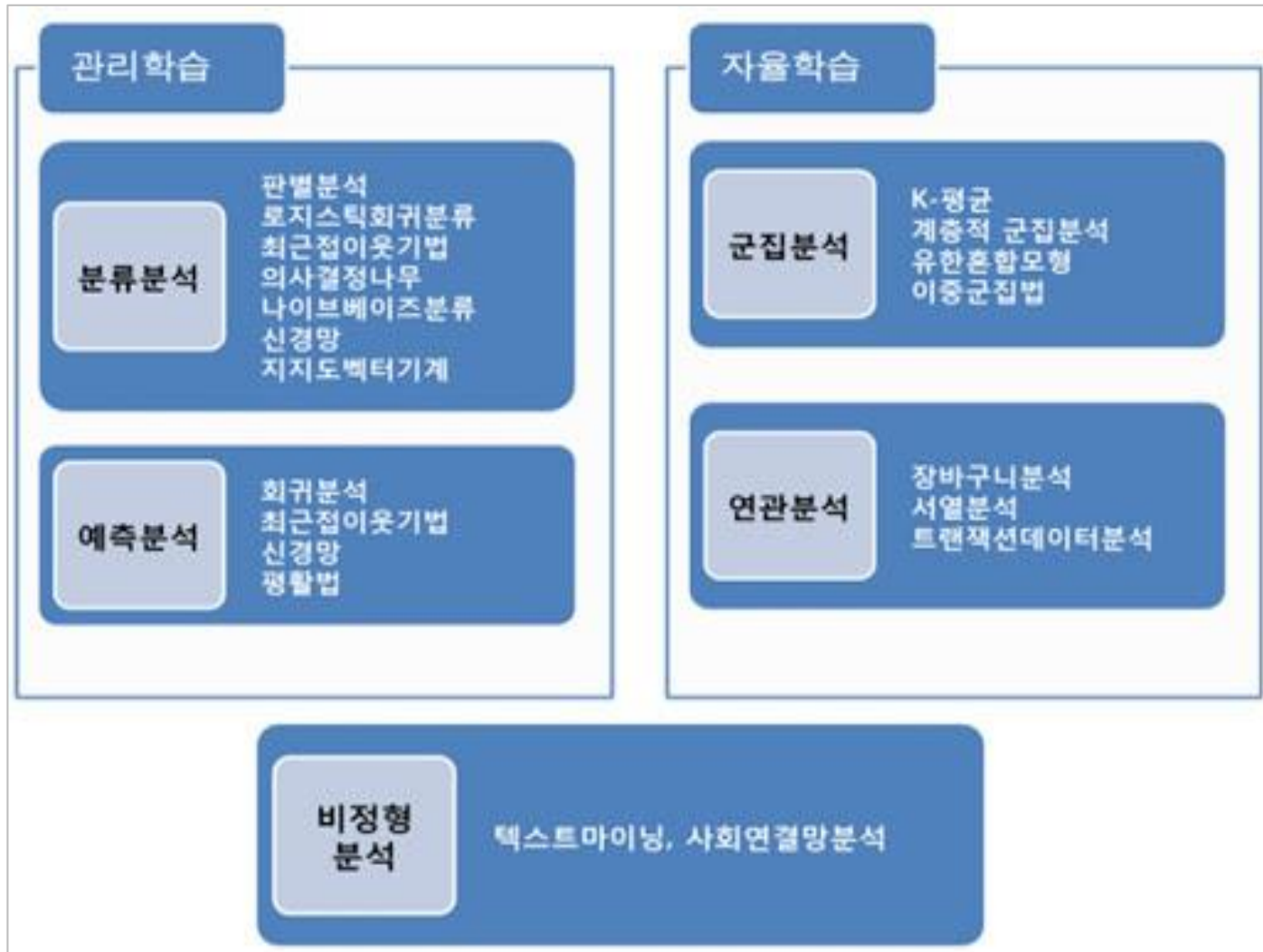
요약



■ 개론 7장 데이터 사이언스 방법 및 분석 기법



■ 입문 7장 데이터 마이닝

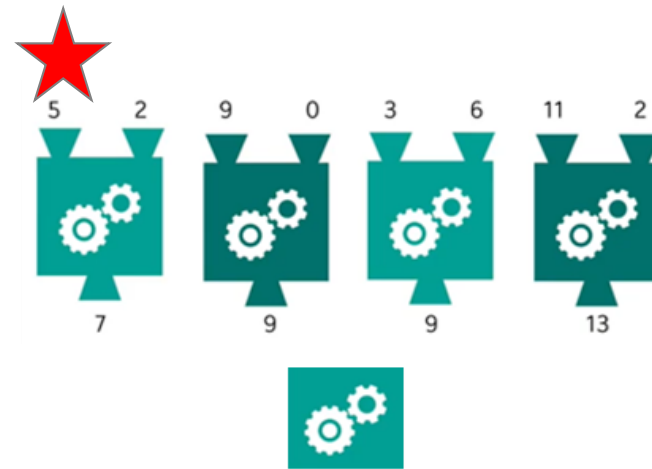


■ 기존의 컴퓨터 사이언스 와 인공지능(AI)

기존 컴퓨터
사이언스



인공지능(AI)



입력 data 학습을 통한 "+"알고리즘(모델) 도출

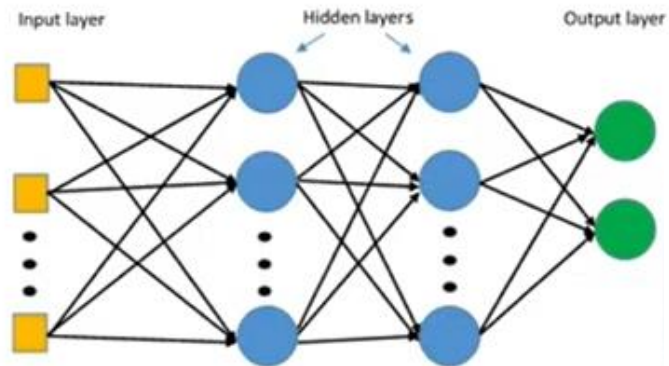


★★★★★
데이터의 질

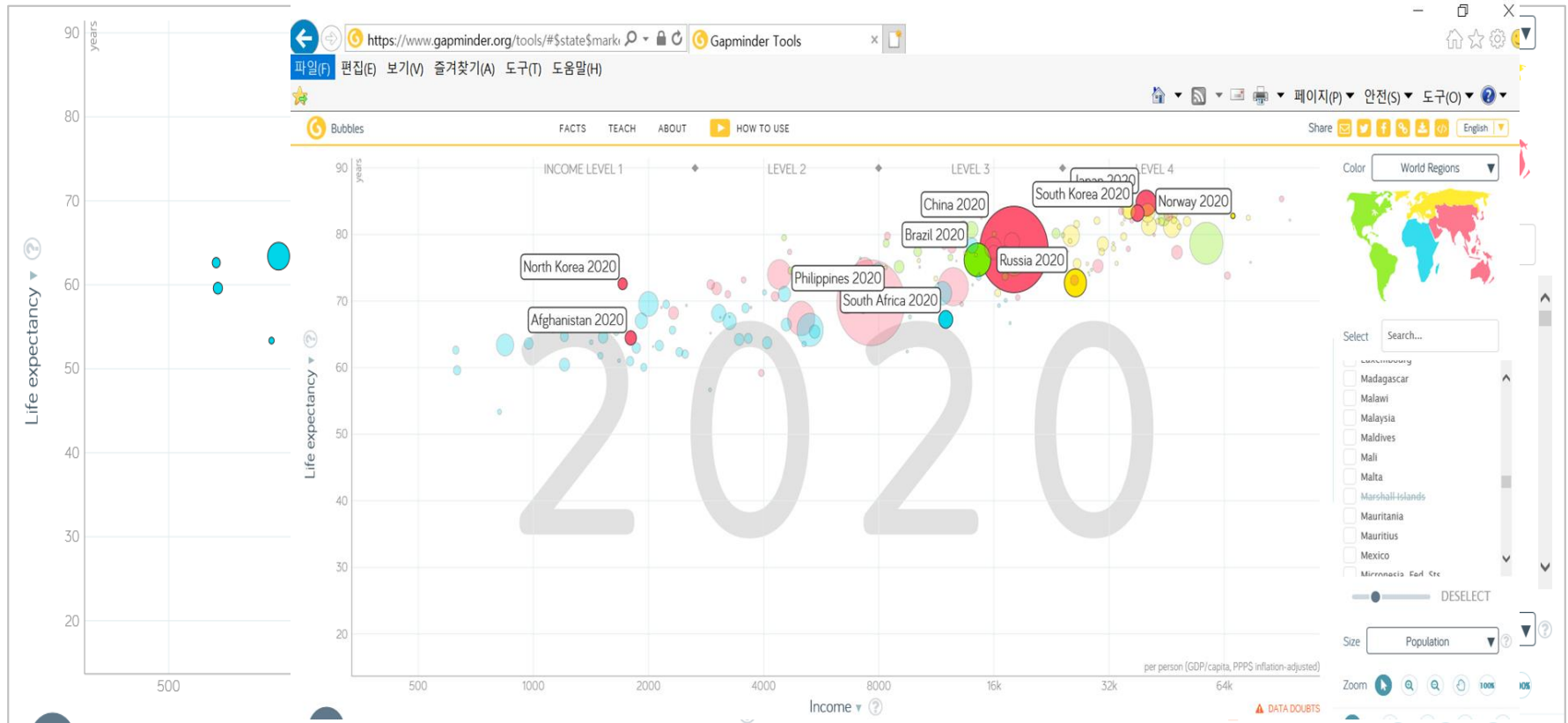
>
데이터의 양
800 PT
100 TB



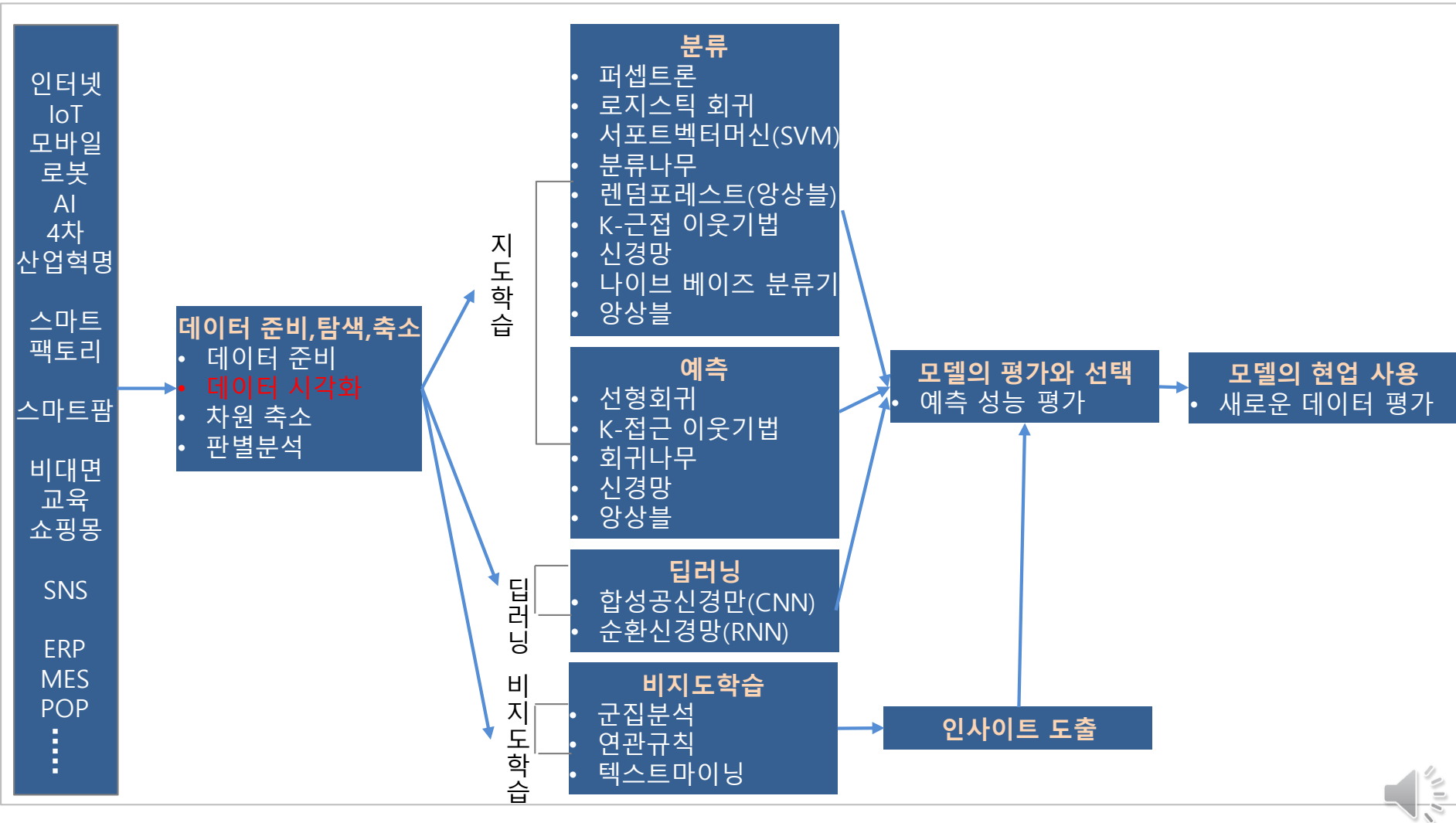
■ 인공지능(AI)과 데이터 사이언스 &.....



- 데이터는 수많은 속성과 샘플로 구성되어 한눈에 의미 파악이 어려움
- 데이터의 의미를 통찰하고 전달하는 가장 좋은 방법은 시각적으로 표현하는 것



6.1 데이터 시각화란?



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

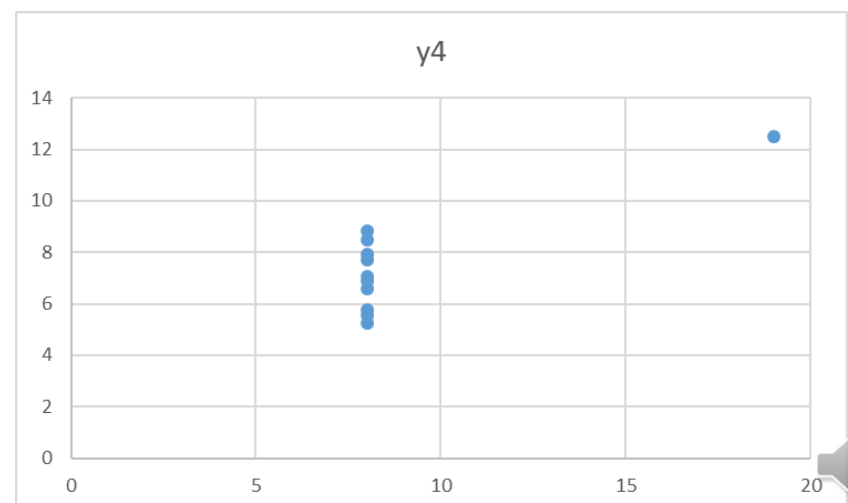
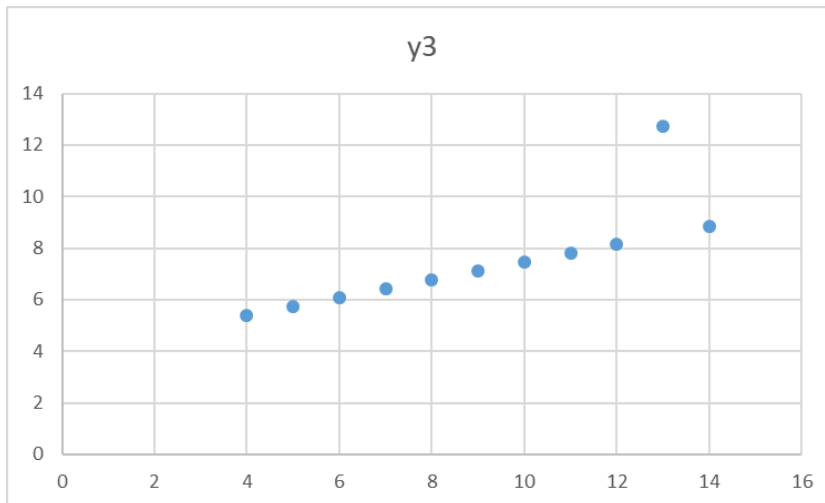
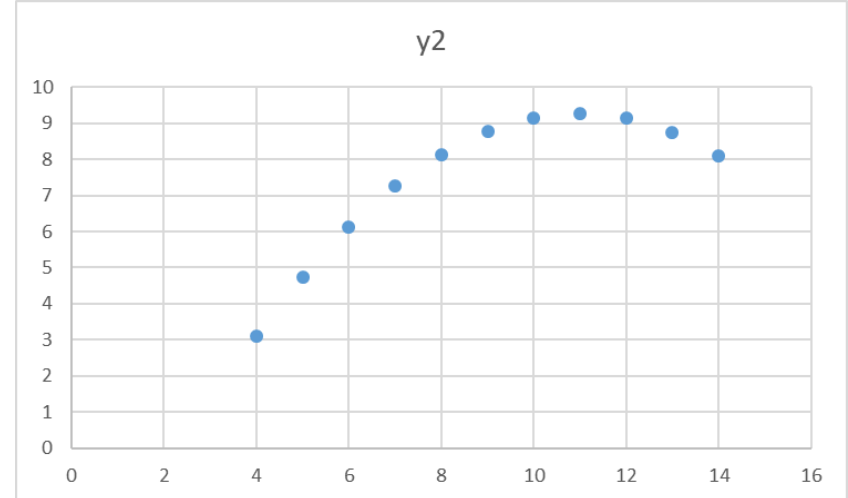
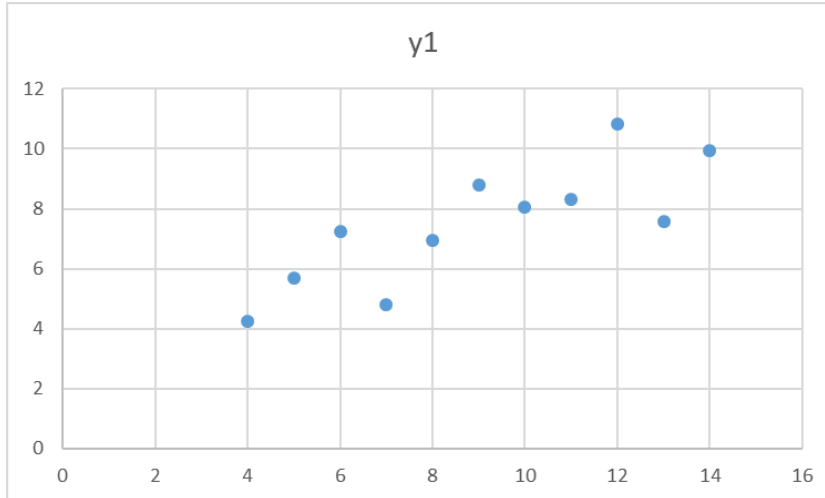
- ✓ 데이터의 시각화는 데이터를 관찰하는 과정에서 선택 사항이 아니라 반드시 거쳐야 하는 필수 과정
- ✓ 엑셀로 Table과 같은 data 준비

	A	B	C	D	E	F	G	H
1	x1	x2	x3	x4	y1	y2	y3	y4
2	10	10	10	8	8.04	9.14	7.46	6.58
3	8	8	8	8	6.95	8.14	6.77	5.76
4	13	13	13	8	7.58	8.74	12.74	7.71
5	9	9	9	8	8.81	8.77	7.11	8.84
6	11	11	11	8	8.33	9.26	7.81	8.47
7	14	14	14	8	9.96	8.1	8.84	7.04
8	6	6	6	8	7.24	6.13	6.08	5.25
9	4	4	4	19	4.26	3.1	5.39	12.5
10	12	12	12	8	10.84	9.13	8.15	5.56
11	7	7	7	8	4.82	7.26	6.42	7.91
12	5	5	5	8	5.68	4.74	5.73	6.89



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

✓ 엑셀 data R로 불러오기

The screenshot displays the RStudio environment with the following components:

- Environment Panel:** Lists variables:
 - `a`: 11 obs. of 8 variables
 - `avg_data`: 18249 obs. of 7 variables
 - `avg_data1`: 424 obs. of 5 variables
- Console:** Shows the execution of the following R code:


```
library(readxl)
a=read_excel("c:/rdata/6c/anscombe.xlsx",col_names = TRUE)
head(a,11)
```

The output shows a tibble with 11 rows and 8 columns (x1, x2, x3, x4, y1, y2, y3, y4). A red box highlights the first few rows of the output.
- Files Panel:** Shows the file `anscombe.xlsx` in the `c:/rdata/6c/` directory.
- Usage Panel:** Displays the function signature for `read_excel`:


```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
            col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
            n_max = Inf, guess_max = min(1000, n_max),
            progress = readxl_progress(), .name_repair = "unique")
```

6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

- ✓ 엑셀 data R로 불러오기

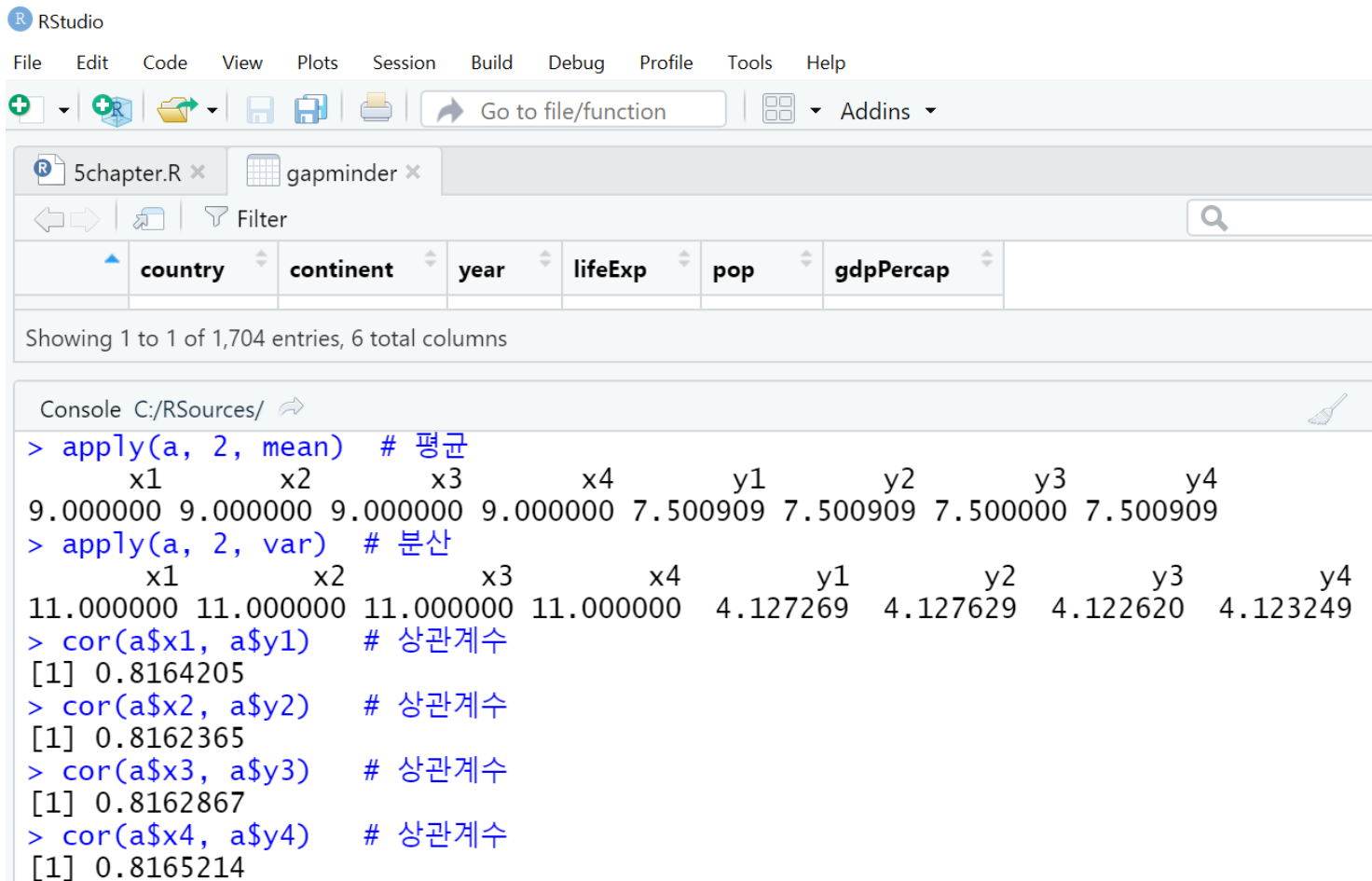
```
Console C:/RSources/ ↗
> library(readxl)
> a=read_excel("c:/rdata/6c/anscombe.xlsx",col_names = TRUE)
> head(a,11)
# A tibble: 11 x 8
   x1    x2    x3    x4    y1    y2    y3    y4
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     10     10     10      8  8.04  9.14  7.46  6.58
2      8      8      8      8  6.95  8.14  6.77  5.76
3     13     13     13      8  7.58  8.74 12.7   7.71
4      9      9      9      8  8.81  8.77  7.11  8.84
5     11     11     11      8  8.33  9.26  7.81  8.47
6     14     14     14      8  9.96  8.1   8.84  7.04
7      6      6      6      8  7.24  6.13  6.08  5.25
8      4      4      4     19  4.26  3.1   5.39 12.5
9     12     12     12      8 10.8   9.13  8.15  5.56
10     7      7      7      8  4.82  7.26  6.42  7.91
11     5      5      5      8  5.68  4.74  5.73  6.89
```



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

✓ 데이터 살펴보기(평균, 분산, 상관관계 등)



The image shows the RStudio interface with the 'gapminder' dataset loaded. The Environment pane displays the dataset with columns: country, continent, year, lifeExp, pop, and gdpPercap. The Console pane shows the following R code and output:

```
> apply(a, 2, mean) # 평균
      x1      x2      x3      x4      y1      y2      y3      y4
9.000000 9.000000 9.000000 9.000000 7.500909 7.500909 7.500000 7.500909
> apply(a, 2, var) # 분산
      x1      x2      x3      x4      y1      y2      y3      y4
11.000000 11.000000 11.000000 11.000000 4.127269 4.127629 4.122620 4.123249
> cor(a$x1, a$y1) # 상관계수
[1] 0.8164205
> cor(a$x2, a$y2) # 상관계수
[1] 0.8162365
> cor(a$x3, a$y3) # 상관계수
[1] 0.8162867
> cor(a$x4, a$y4) # 상관계수
[1] 0.8165214
```



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

✓ 선형 회귀식 등을 통한 비교

■ 선형 회귀식도 거의 동일

$$y1 = 0.5001 \times x1 + 3.0001$$

$$y2 = 0.500 \times x2 + 3.001$$

$$y3 = 0.4997 \times x3 + 3.0025$$

$$y4 = 0.4999 \times x4 + 3.0017$$

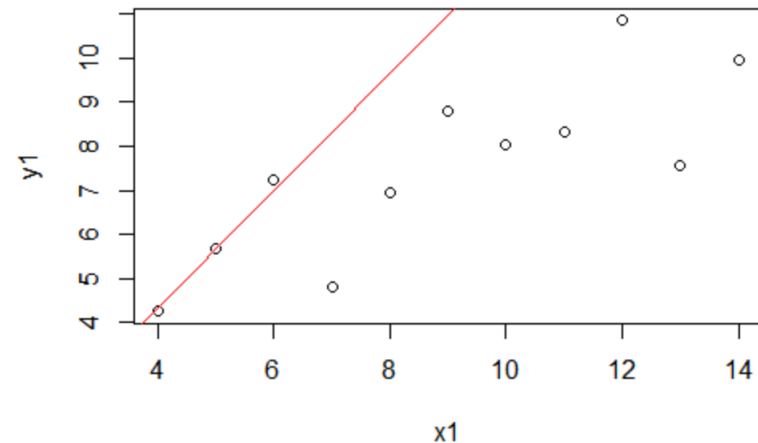
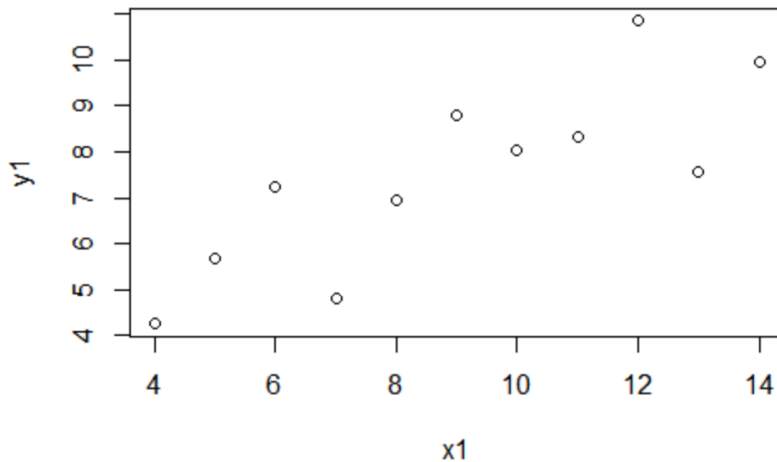
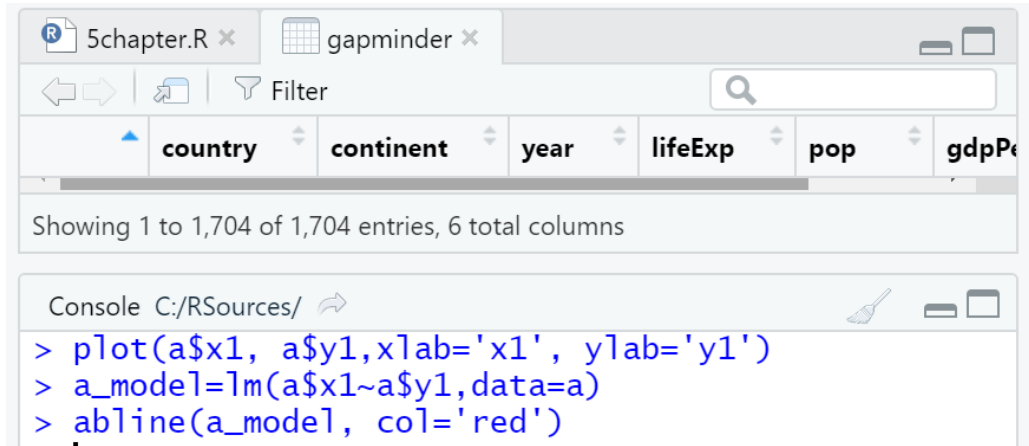
- 통계 지표와 분석 수치만으로 비교해보면, 4개의 데이터 셋은 거의 동일하다고 판단할 수 있음
- 평균, 분산, 상관계수 거의 동일



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

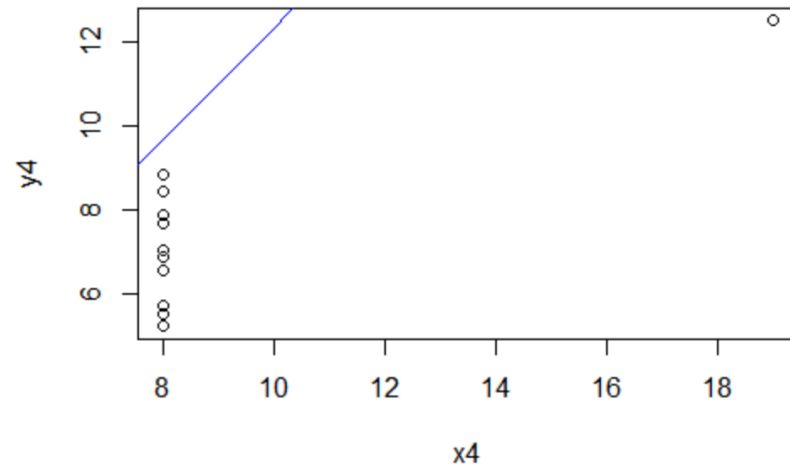
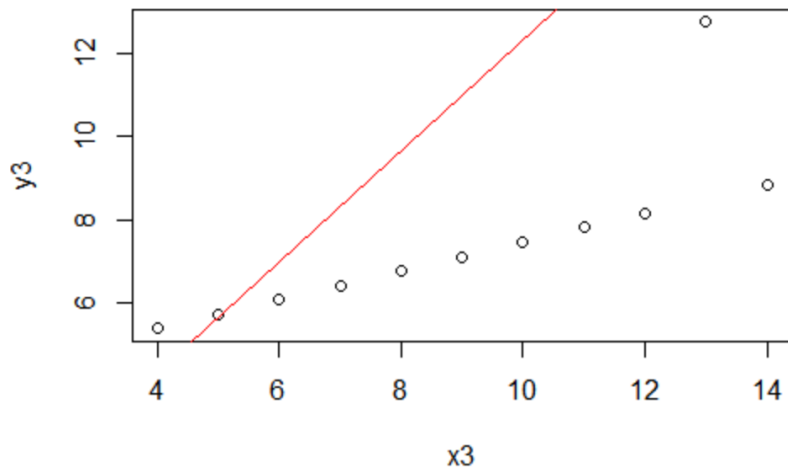
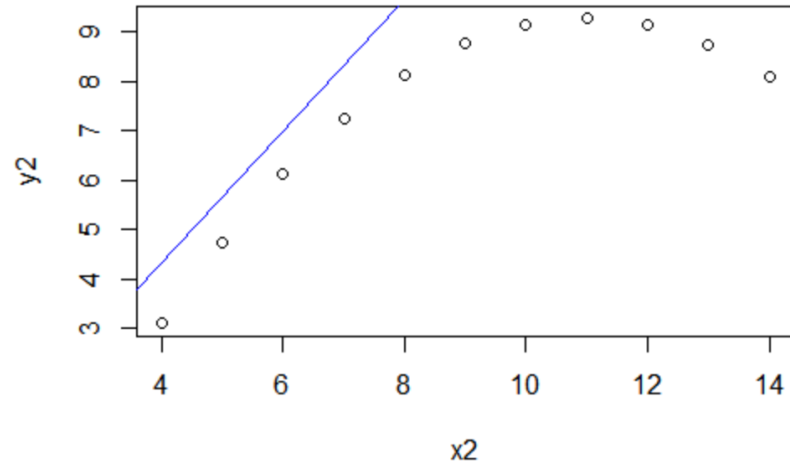
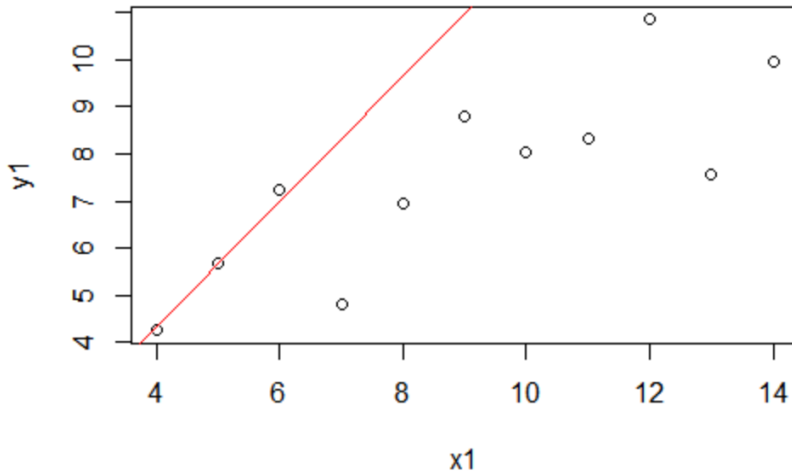
✓ 시각화



6.1 데이터 시각화란?

■ 데이터 시각화의 필요성

✓ 시각화



6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ gapminder 데이터에는 5개 대륙, 총 142개 국가에 대한 1952~2007년의 인구 데이터가 5년 간격으로 저장 되어 있음
- ✓ 먼저 전체를 파악하기 위해 : 인구 변화의 추이를 대륙별로 보고자 함

표 5-1 gapminder 데이터 프레임의 구성 항목

열이름(변수명)	변수형	내용
country	142개 레벨의 범주형	국가명
continent	5개 레벨의 범주형	국가가 속한 대륙
year	int	1952~2007 관측 연도(5년 단위)
lifeExp	num	기대 수명(평균 수명)
pop	int	인구
gdpPercap	num	1인당 국내총생산(물가 상승 반영)



6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ gapminder 데이터에는 5개 대륙, 총 142개 국가에 대한 1952~2007년의 인구 데이터가 5년 간격으로 저장되어 있음
- ✓ 먼저 전체를 파악하기 위해 : 인구 변화의 추이를 대륙별로 보고자 함

	A	B	C	D	E	F	G
1	NO	country	continent	year	lifeExp	pop	gdpPercap
841	840	Korea, Dem. Rep.	Asia	2007	67.297	23,301,725	1593.07
842	841	Korea, Rep.	Asia	1952	47.453	20,947,571	1030.59
843	842	Korea, Rep.	Asia	1957	52.681	22,611,552	1487.59
844	843	Korea, Rep.	Asia	1962	55.292	26,420,307	1536.34
845	844	Korea, Rep.	Asia	1967	57.716	30,131,000	2029.23
846	845	Korea, Rep.	Asia	1972	62.612	33,505,000	3030.88
847	846	Korea, Rep.	Asia	1977	64.766	36,436,000	4657.22
848	847	Korea, Rep.	Asia	1982	67.123	39,326,000	5622.94
849	848	Korea, Rep.	Asia	1987	69.810		
850	849	Korea, Rep.	Asia	1992	72.244		
851	850	Korea, Rep.	Asia	1997	74.647		
852	851	Korea, Rep.	Asia	2002	77.045		
853	852	Korea,	Asia	2007	78.623		
854	853	Kuwait	Asia	1952	55.565		
855	854	Kuwait	Asia	1957	58.033		
856	855	Kuwait	Asia	1962	60.47		
857	856	Kuwait	Asia	1967	64.624	575,003	80894.88





열이름(변수명)	변수형	내용
country	142개 레벨의 범주형	국가명
continent	5개 레벨의 범주형	국가가 속한 대륙
year	int	1952~2007 관측 연도(5년 단위)
lifeExp	num	기대 수명(평균 수명)
pop	int	인구
gdpPercap	num	1인당 국내총생산(물가 상승 반영)



6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ 년도별, 대륙별 인구 합구하기

```
Console C:/Rsources/      
> library(gapminder)  
> library(dplyr)  
> y_c_pop <- gapminder %>% group_by(year, continent) %>% summarize(c_pop=sum  
(pop))  
`summarise()` has grouped output by 'year'. You can override using the `.`groups` argument.  
> head(y_c_pop,10)  
# A tibble: 10 x 3  
# Groups:   year [2]  
   year continent      c_pop  
  <int> <fct>      <dbl>  
1  1952 Africa    237640501  
2  1952 Americas  345152446  
3  1952 Asia     1395357351  
4  1952 Europe   418120846  
5  1952 Oceania   10686006  
6  1957 Africa    264837738  
7  1957 Americas  386953916  
8  1957 Asia     1562780599  
9  1957 Europe   437890351  
10 1957 Oceania   11941976
```

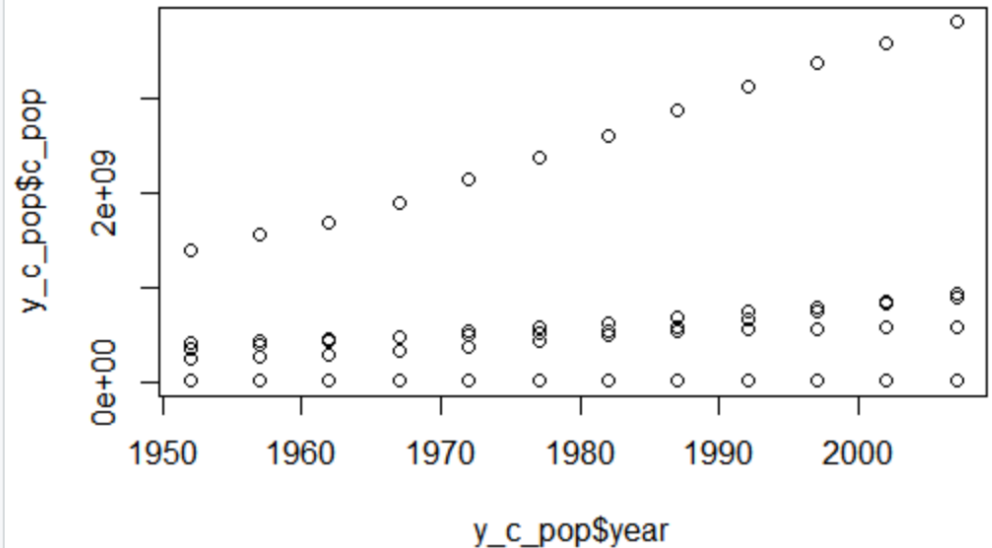


6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ 년도별, 대륙별 인구 합구하기 → 1차 기본 시각화(년도 가로축, 인구 세로축)

```
1:1 (Top Level) R Script
Console C:/RSources/
> head(y_c_pop,10)
# A tibble: 10 x 3
# Groups:   year [2]
  year continent    c_pop
  <int> <fct>      <dbl>
1  1952 Africa    237640501
2  1952 Americas  345152446
3  1952 Asia     1395357351
4  1952 Europe   418120846
5  1952 Oceania   10686006
6  1957 Africa    264837738
7  1957 Americas  386953916
8  1957 Asia     1562780599
9  1957 Europe   437890351
10 1957 Oceania   11941976
> plot(y_c_pop$year, y_c_pop$c_pop)
```



6.1 데이터 시각화란?

■ >?plot

R: The Default Scatterplot Function ▾ Find in Topic

Usage

```
## Default S3 method:  
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL,  
      log = "", main = NULL, sub = NULL, xlab = NULL,  
      ann = par("ann"), axes = TRUE, frame.plot = axes,  
      panel.first = NULL, panel.last = NULL, asp = NA,  
      xgap.axis = NA, ygap.axis = NA,  
      ...)
```

Arguments

<code>x, y</code>	the <code>x</code> and <code>y</code> arguments provide the <code>x</code> and <code>y</code> coordinates for the plot. Any reasonable way of defining the coordinates is acceptable. See the function xy.coords for details. If supplied separately, they must be of the same length.
<code>type</code>	1-character string giving the type of plot desired

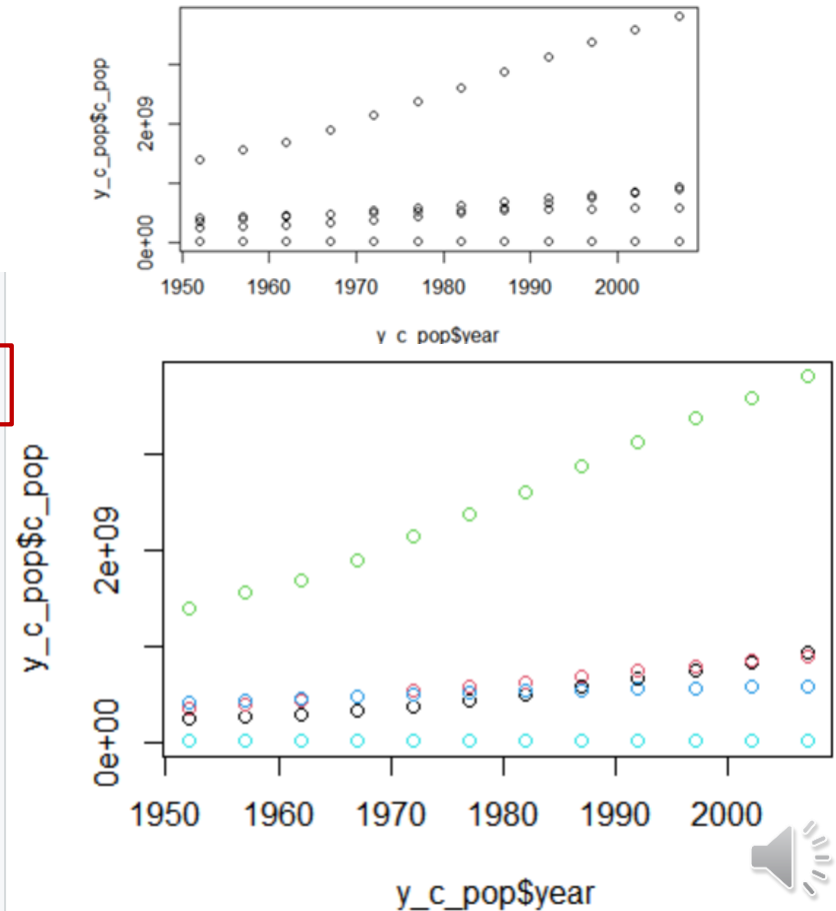


6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ 년도 별, 대륙 별 인구 합 구하기 → 1차 기본 시각화(년도 가로축, 인구 세로축)
→ 2차 대륙 별 구분 색상 부여하기 (col option 사용)

```
Console C:/RSources/
> plot(y_c_pop$year, y_c_pop$c_pop)
> plot(y_c_pop$year, y_c_pop$c_pop, col=y_c_pop$continent)
> |
```

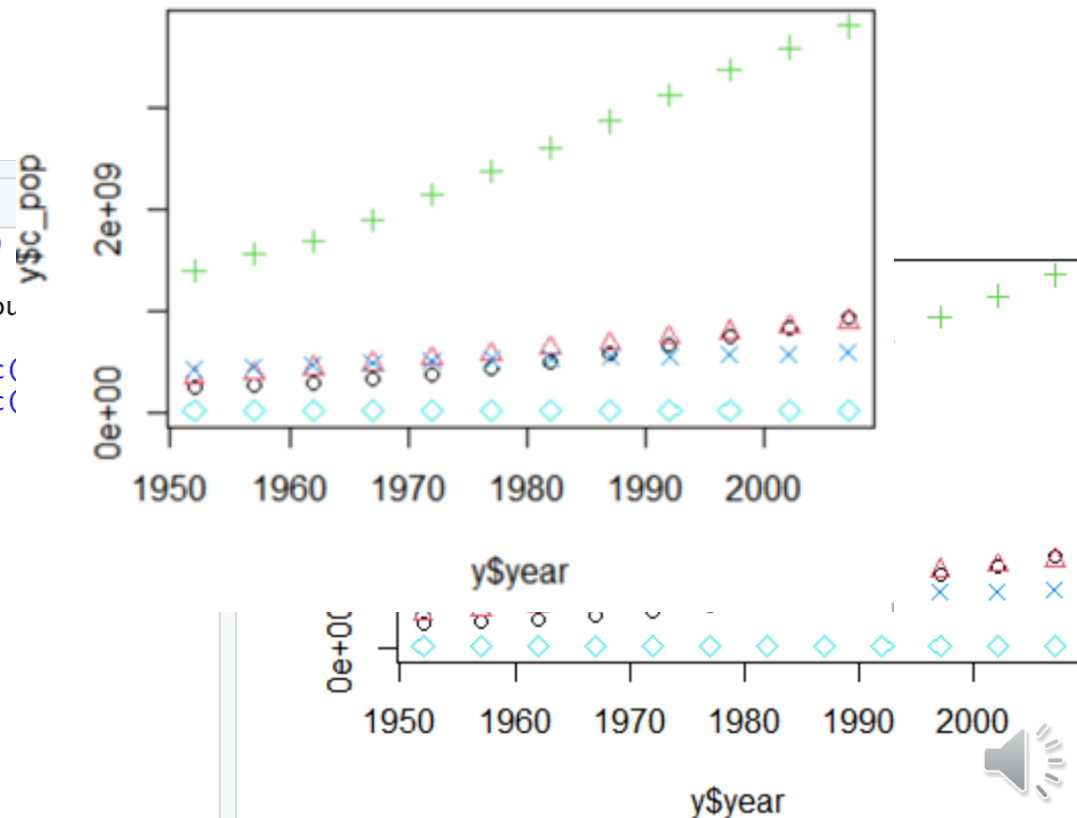


6.1 데이터 시각화란?

■ 시각화의 기본 요소































- ✓ 년도 별, 대륙 별 인구 합 구하기 → 1차 기본 시각화(년도 가로축, 인구 세로축)
 - 2차 대륙 별 구분 색상 부여하기 (col option 사용)
 - 3차 서로 다른 마커(Marker)로 표현하기(pch option 사용)

```
Console C:/RSources/ ↗  
> y <- gapminder %>% group_by(year, continent)  
  summarise(c_pop=sum(pop))  
`summarise()` has grouped output by 'year'. You  
  can use the `.groups` argument.  
> plot(y$year, y$c_pop, col=y$continent, pch=c(  
  > plot(y$year, y$c_pop, col=y$continent, pch=c(  
    els(y$continent))))  
> |
```



6.1 데이터 시각화란?

■ pch option 사용

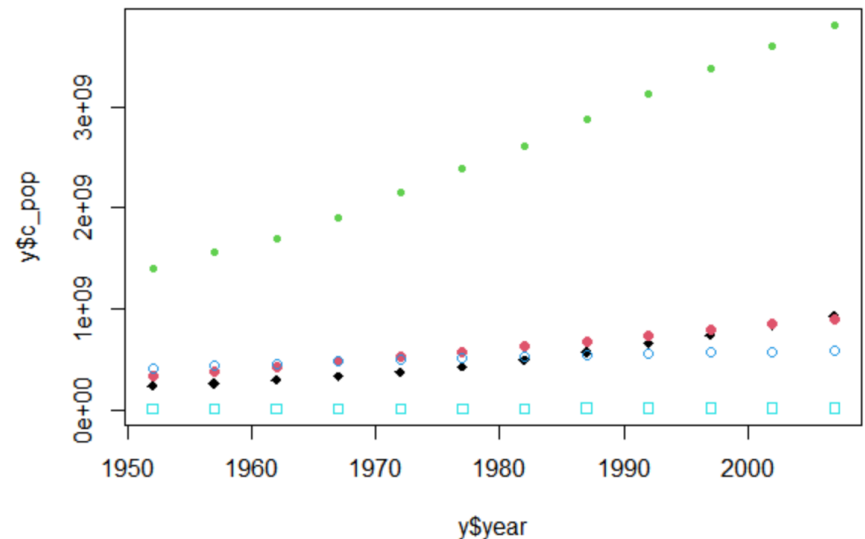
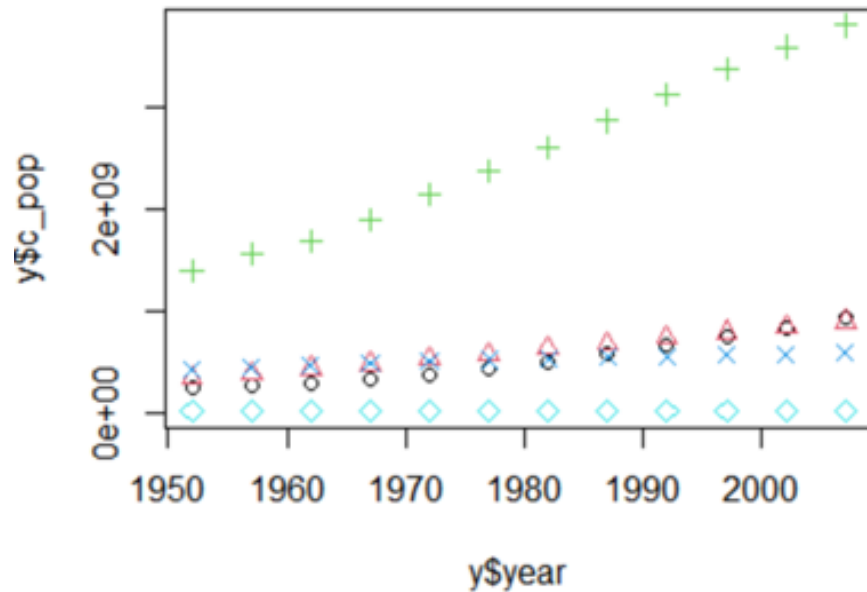
					
pch=0	pch=5	pch=10	pch=15	pch=20	pch=25
					
pch=1	pch=6	pch=11	pch=16	pch=21	
					
pch=2	pch=7	pch=12	pch=17	pch=22	
					
pch=3	pch=8	pch=13	pch=18	pch=23	
					
pch=4	pch=9	pch=14	pch=19	pch=24	



6.1 데이터 시각화란?

■ pch option 사용

```
>plot (y$year, y$c_pop, col=y$continent,pch=c(1:5))
>plot (y$year, y$c_pop, col=y$continent,pch=c(18:22))
```



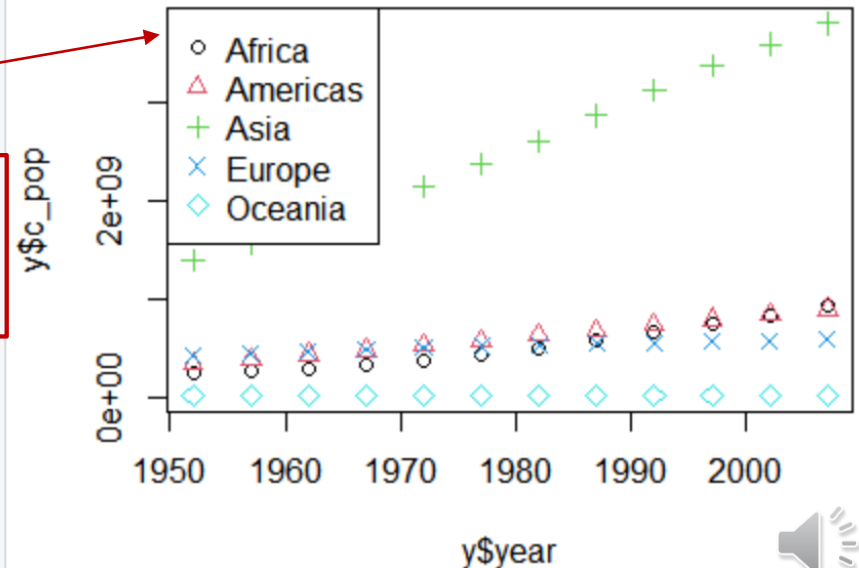
6.1 데이터 시각화란?

■ 시각화의 기본 요소

- ✓ 년도 별, 대륙 별 인구 합 구하기 → 1차 기본 시각화(년도 가로축, 인구 세로축)
 - 2차 대륙 별 구분 색상 부여하기 (col option 사용)
 - 3차 서로 다른 마커(Marker)로 표현하기(pch option 사용)
 - 4차 범례 표시하기(legend)

Console C:/RSources/ ↗

```
> y <- gapminder %>% group_by(year, continent) %>% summarize(c_pop=sum(pop))
`summarise()` has grouped output by 'year'. You can override using the `.groups` argument.
> plot(y$year, y$c_pop, col=y$continent, pch=c(1:5))
> # 범례 개수를 상수로 지정
> legend("topleft", legend=5, col=c(1:5), pch=c(1:5))
> legend("topleft", legend=levels(y$continent), pch=c(1:length(levels(y$continent))), col=c(1:length(levels(y$continent))))
>
```



Thank you

