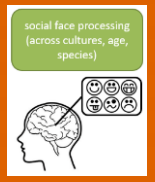
분석 인프라 활용 AI교육(R)
Day 1

Cho Heeseung

hscho9384@korea.ac.kr

소개

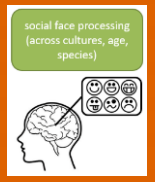


CognitiveSystems

- 서강대학교 수학, 경제학과 졸업 (2011.03~2018.02)
- 서강대학교 경제학과 고급계량경제학 학과목 멘토(2016.09~2016.12)
- LG CNS 재직 – Data Scientist (2018.01~2020.02)
- 고려대학교 인공지능학과 석박통합과정(2020.03~)
- SK 패스트캠퍼스 AI/DT과정 멘토(2021.01)
- 고려대학교 데이터과학과 인공지능 학과목 멘토(2021.03~)

더 궁금하다면? https://heeseung-cho.github.io/

목차



CognitiveSystems

1. R 개요 및 설치방법
2. R 기본 문법
3. 데이터 프레임
4. 시각화
5. 데이터 전처리

1. R 개요 및 설치방법

R이란?

CognitiveSystems

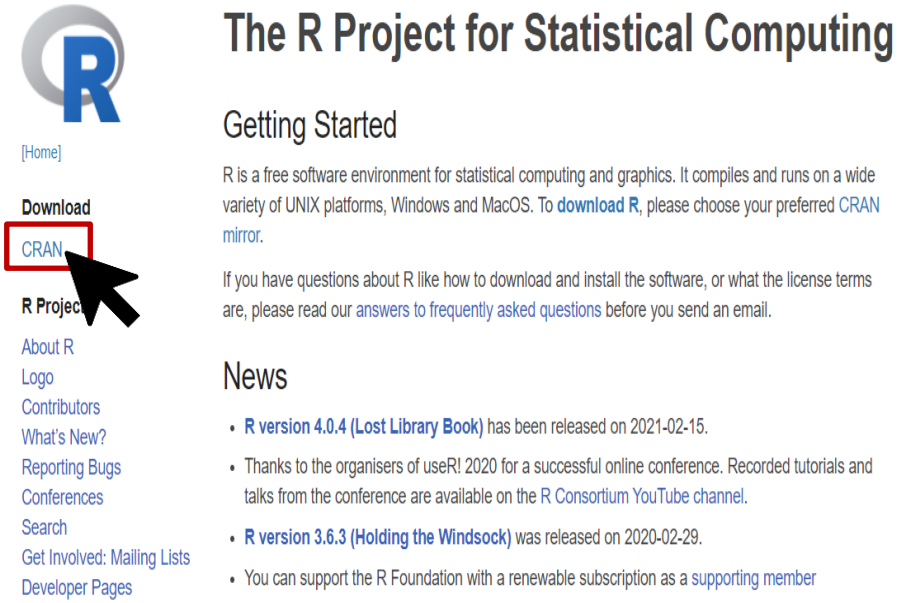
- 통계 컴퓨팅, 그래픽을 위한 언어
- 객체지향 언어 + 함수형 언어
- 오픈소스

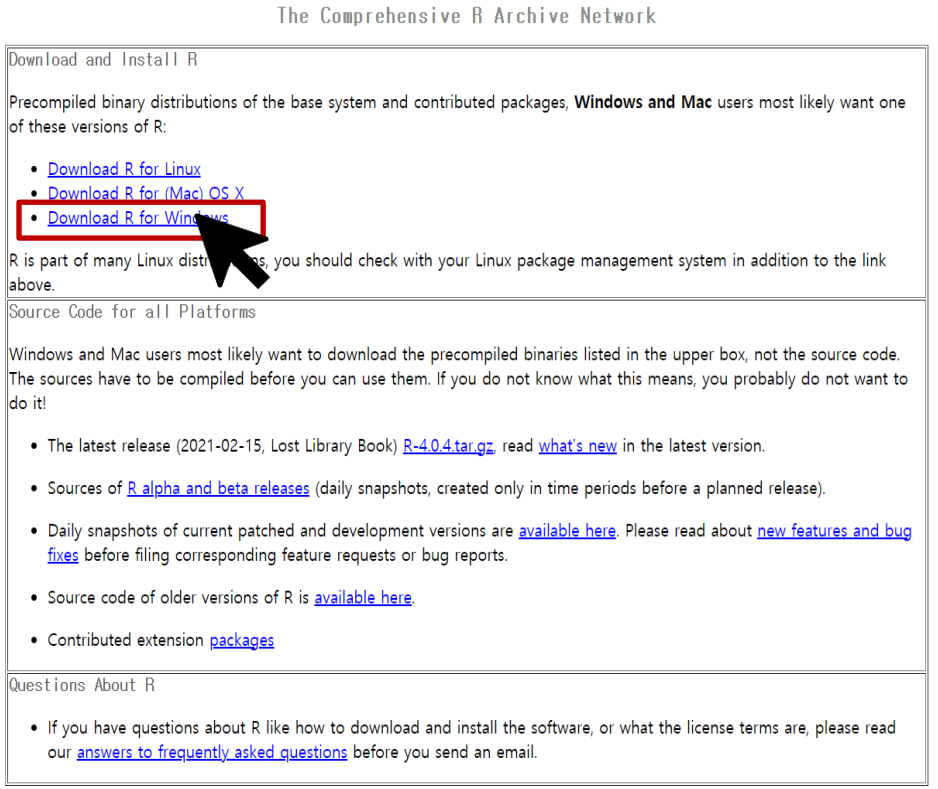
(+): 강력한 통계 분석 툴들을 제공하여 분석 및 시각화에 용이
다른 통계 프로그램(SAS, STATA 등)에 비해 간단하고 습득이 쉬움.
(-): 단일 environment로 여러 작업을 하기 힘듦, 범용성이 파이썬에 비해 부족

원문: https://www.r-project.org/about.html

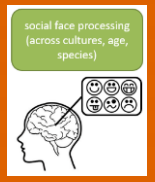
R 설치

R download: https://www.r-project.org/





R 설치



CognitiveSystems

R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

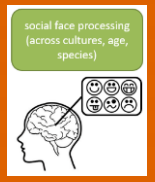
R-4.0.4 for Windows (32/64 bit)

Download R 4.0.4 for Windows (85 megabytes, 32/64 bit)

Installation and other instructions

New features in this version

R Studio란?



CognitiveSystems

- R을 더욱 편리한 환경에서 사용하기 위해 개발된 통합개발환경 툴.

- 자료 분석부터 보고서 작성까지 end-to-end 기능 제공.

- 기본 기능들은 Tool로 제공함으로써, 코딩을 모르는 사람이라도 몇번의 클릭만 가지고 분석이 가능함.

R Studio 설치

R Studio download: https://rstudio.com/products/rstudio/download/

RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Server Pro
Open Source License	Commercial License	Open Source License	Commercial License
Free	**$995** /year	**Free**	**$4,975** /year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY
Learn more	Learn more	Learn more	Evaluation | Learn more

RStudio Desktop 1.4.1106 - Release Notes

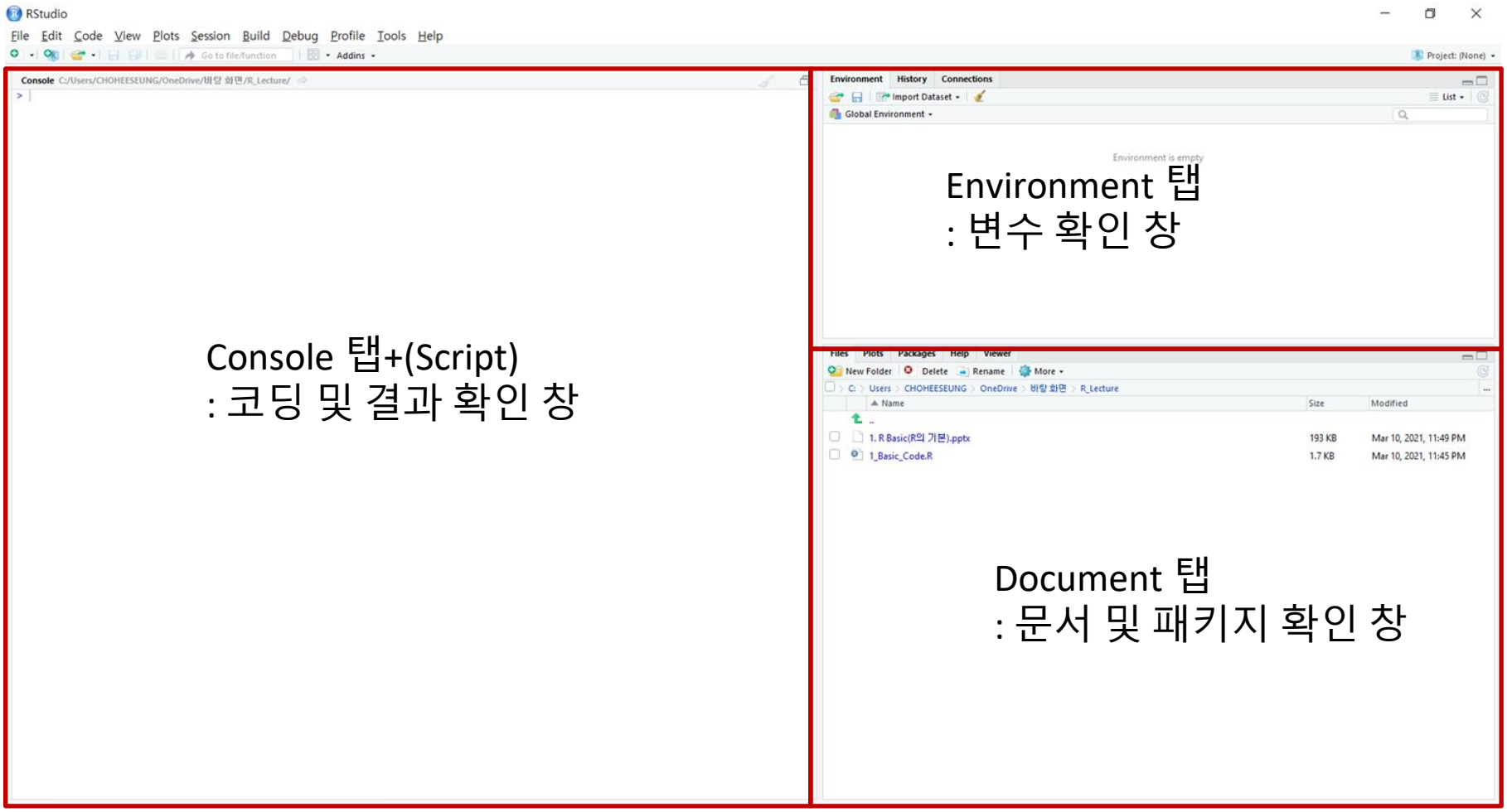
1. Install R. RStudio requires R 3.0.1+.

2. Download RStudio Desktop. Recommended for your system:

DOWNLOAD RSTUDIO FOR WINDOWS
1.4.1106 | 155.97MB

Requires Windows 10/8/7 (64-bit)

Rstudio 실행

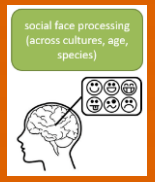


Console 탭+(Script)
: 코딩 및 결과 확인 창

Environment 탭
: 변수 확인 창

Document 탭
: 문서 및 패키지 확인 창

Rstudio - Console



CognitiveSystems

Console: Prompt처럼 실행되는 영역
입력과 동시에 출력이 진행되는 인터랙티브 모드

Console C:/Users/CHOHEESEUNG/OneDrive/바탕 화면/R_Lecture/

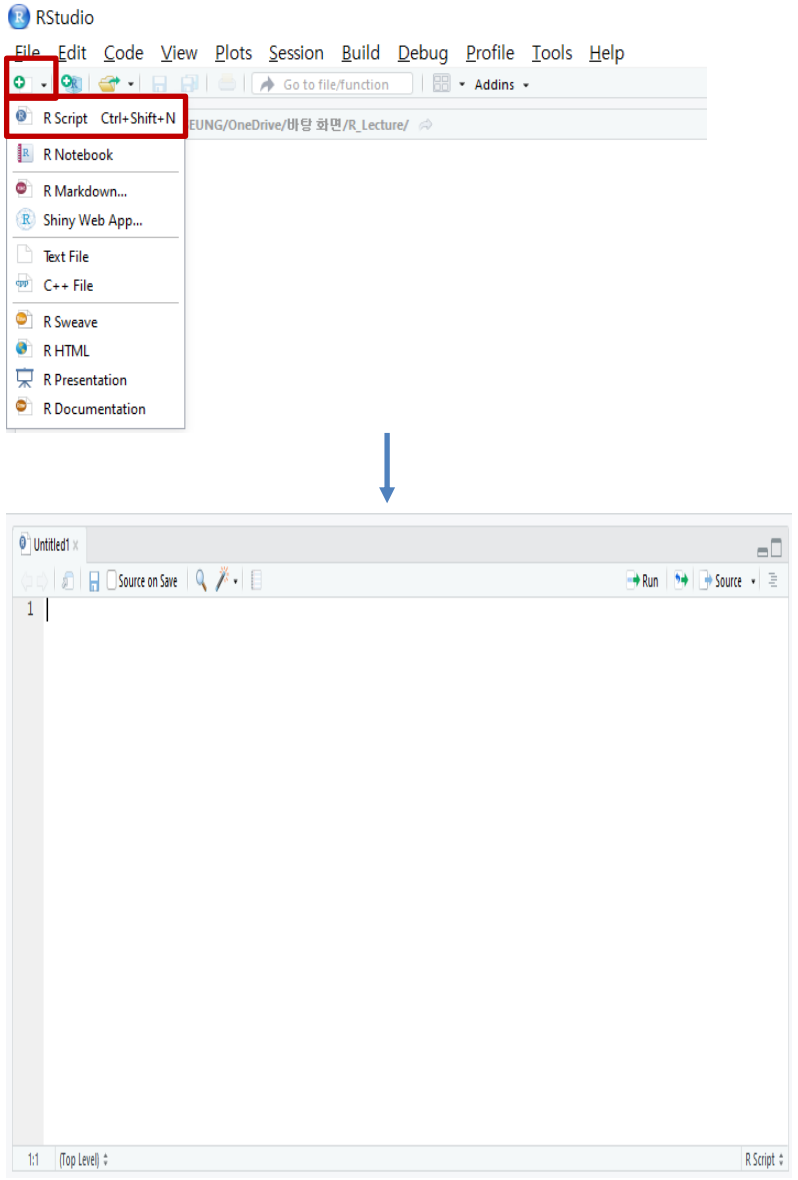
```
> 1+2
[1] 3
> data()
> head(trees, 10)
   Girth Height Volume
1    8.3     70   10.3
2    8.6     65   10.3
3    8.8     63   10.2
4   10.5     72   16.4
5   10.7     81   18.8
6   10.8     83   19.7
7   11.0     66   15.6
8   11.0     75   18.2
9   11.1     80   22.6
10  11.2     75   19.9
>
```

Rstudio - Script

Script: 파이썬의 Jupyter notebook처럼 라인 별, 영역 별 실행이 가능.

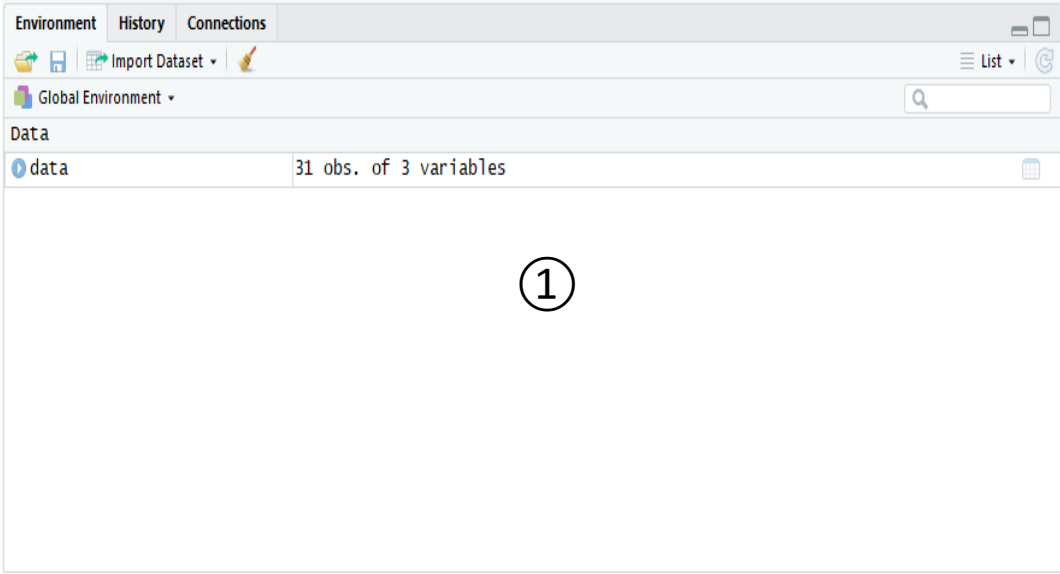
Console의 경우 수정, 재사용이 불편하기에 앞으로의 모든 코딩 및 실습들은 script에서 진행될 예정.

줄 실행: Ctrl+Enter
영역 실행: 드래그 후 Ctrl+Enter
전체 실행: Crtl+A 후 Ctrl+Enter



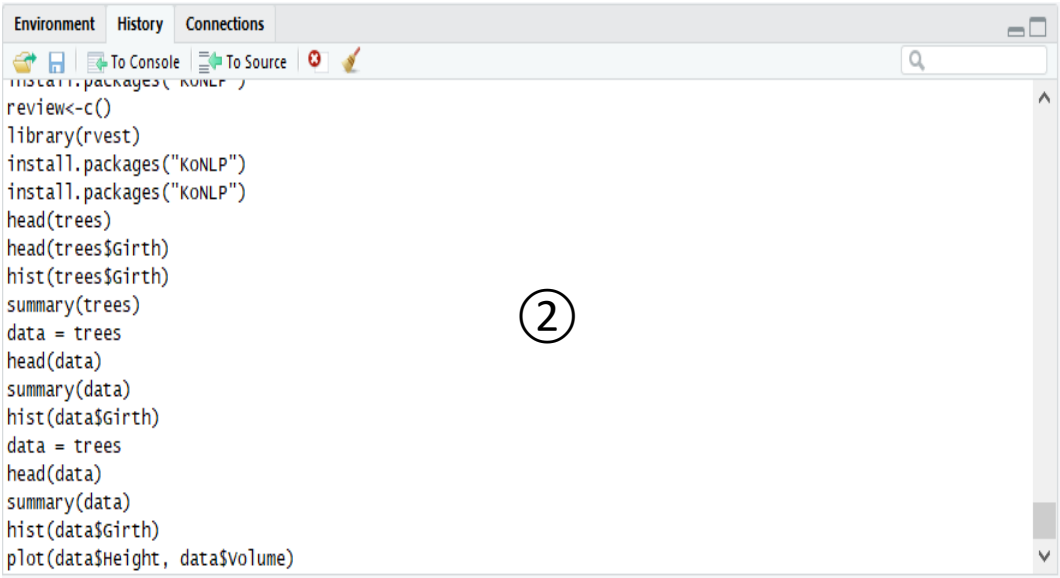
Rstudio - Environment

1. 현재 환경에서 사용하고 있는 변수들 확인.
환경 변수 저장 및 불러오기,
데이터셋 불러오기,
변수 삭제 가능.



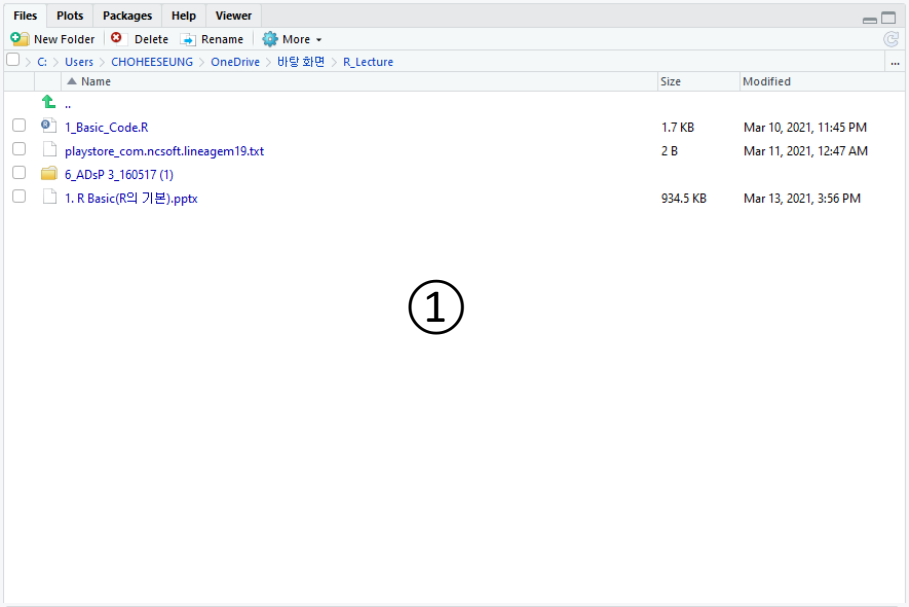
2. History 제공. 이전에 어떤 코드가 돌아갔는지 확인.

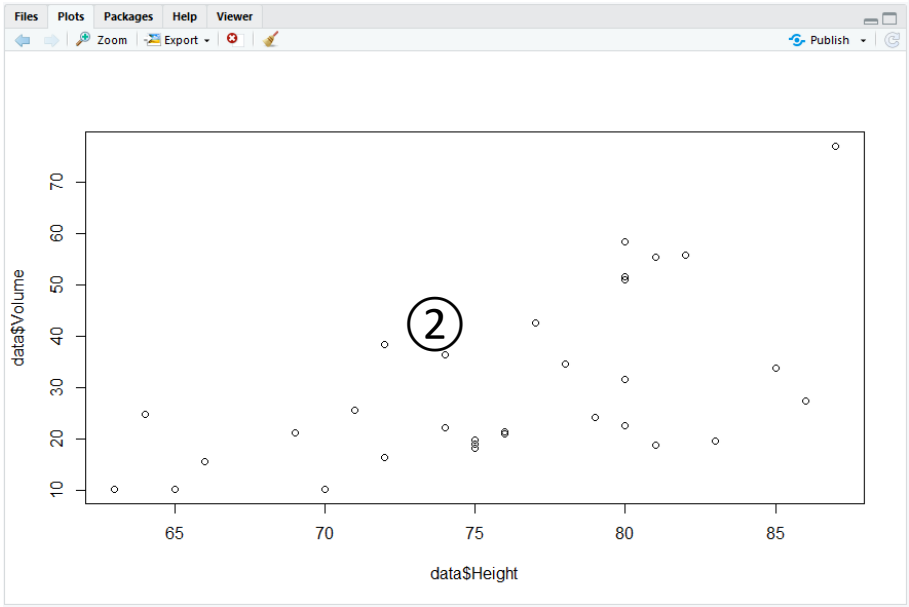
-To Console: 해당 코드를 콘솔에 재사용
-To Source: 해당 코드를 script에 재사용.

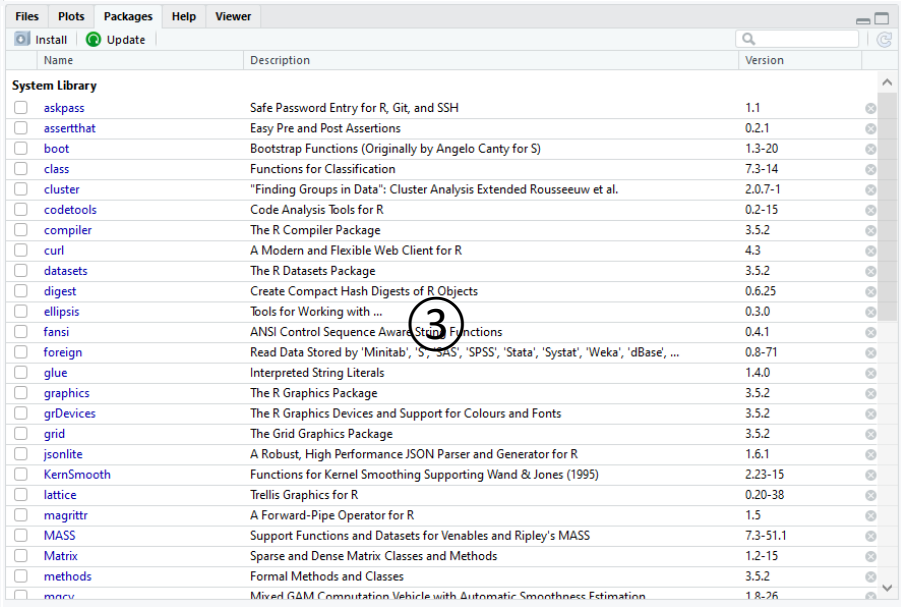


Rstudio - Document

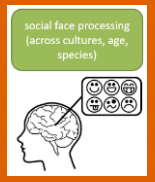
1. 현재 코드가 돌아가고 있는 경로를 확인.
2. 시각화 결과 화면
3. 환경 내 Package 현황 확인 및 현재 불러온 Package 확인 가능.







Library



CognitiveSystems

오픈소스인 R의 장점 중 하나로, 사용자들이 작성한 패키지를 다운받아 사용할 수 있다.

- Install.packages("라이브러리 이름"): 라이브러리를 다운받는다.

*** 경로가 한글로 되어있을 시 다운이 되지 않을 수 있다. 반드시 라이브러리 다운로드 전 경로(내 문서, 바탕화면, 유저이름 등)가 모두 영어로 설정되어 있는지 확인.

- library(라이브러리 이름): 라이브러리를 실행한다.

```
install.packages("MASS")
library(MASS)
```

2. R 기본 문법

R 변수선언

기존 객체 지향 언어(C++, java, C#, …):
변수를 선언한 후(int, String, float, …)
변수를 할당하여 사용

Python, R:
변수 선언 및 할당이 자동으로 이루어짐.
어떤 값을 입력 받는가에 따라
자동으로 변수 선언이 이루어짐.

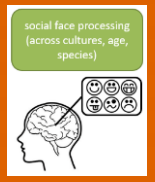
```java
public class MyProgram {
    // 변수 선언
    int num1;
    int num2;
    String msg;

    // Main
    public static void main(String[] args) {

        //변수 할당
        num1 = 1
        num2 = 2
        msg = "Hello World!!!"
        //출력
        System.out.println("Hello World!");
        System.out.println(num1+num2);
        System.out.println(msg);
    }
}
```

Ex) java에서의 변수 선언 및 할당

```r
num1 = 1
num2 = 2
msg = "Hello World!!!"
print("Hello World!")
print(num1+num2)
print(msg)
```

Ex) R에서의 변수 선언과 할당

R 변수선언



CognitiveSystems

변수 선언은 숫자, 문자, 식 등
어느 값이 와도 선언 가능.

```
x = 1
y = 2+3
z = 'Hello World!'
print(x);print(y);print(z)
ls()          #Check variables in environment
ls(pattern = "x")
rm(x) #Remove variable
rm(list = ls()) #Remove all variable
```

- ls(): 현재 환경에서 저장된 변수 확인. (Environment과 동일)
- ls(pattern = “x”): x라는 문자가 들어간 변수들을 모두 출력
- rm(x): x라는 변수 제거
- rm(list = List) : List 내에 있는 변수들을 모두 제거
  - List = ls(pattern = “x”): x라는 문자가 들어간 모든 변수 제거
  - List = ls(): 환경 내 모든 변수 제거

R의 자료 구조 - 숫자

CognitiveSystems

다음 예시와 같이 숫자를 할당해보고, 연산을 해보자.

```r
##1. Constant
x = 5
y = 2
print(x)
print(y)
typeof(x)
is.numeric(x)

#Calculation
x+y
x-y
x*y
x/y
x%%y
x%/%y
x^y
x == y
x <= y
x >= y
```

연산자	결과
x+y	x와 y의 합
x-y	x와 y의 차
x*y	x와 y의 곱
x/y	x와 y의 나눗셈
x%%y	x를 y로 나눈 나머지
x%/%y	x를 y로 나눈 몫
x^y (= x**y)	x의 y거듭제곱
x==y	x와 y의 값이 동일한지. (T or F)
x<=y	x가 y보다 작은지. (T or F)
x>=y	x가 y보다 큰지. (T or F)

R의 자료 구조 - 논리

CognitiveSystems

다음 예시와 같이 논리문을 할당해보고, 연산을 해보자.

- TRUE는 T, FALSE는 F로 간단하게 할당이 가능.
- 숫자연산도 가능하며, 이때 TRUE는 1, FALSE는 0의 값을 가지게 됨.

```
##2. Bool
x = TRUE
y = FALSE
typeof(x)
is.logical(x)

#Calculate
x+y
!x
x&y
x|y

#Tips
z = F

x&&y&&z
x||y||z
```

연산자	결과
x+y	x와 y의 합
!x	x의 부정
x&y	x 그리고 y (x and y)
x|y	x 또는 y (x or y)
x&&y&&z	x and y and z
x||y||z	x or y or z

R의 자료 구조– 벡터(1)

CognitiveSystems

다음 예시와 같이 벡터를 할당해보자.

```
##3. Vector
X = c(4,7,2,1)
Y = c(2,5,3,7)
is.vector(X)
```

벡터와 그 길이를 출력해보자.
벡터는 indexing을 통해 일부분이 출력 가능하며,
***다른 언어와 달리(python 포함) R의 indexing은 1부터 시작한다.

```
print(X)
print(Y)
print(length(X))
print(length(Y))
print(X[1]);print(X[2]);
print(X[3]);print(X[4]);
print(Y[2:3]); print(Y[c(2,4)]);
```

R의 자료 구조– 벡터(2)

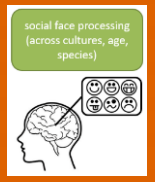
CognitiveSystems

벡터 연산은 벡터 길이가 동일해야 가능하며, 각 연산은 Elementwise하게 진행된다. 또한, 숫자와 연산 시 벡터와 스칼라 간의 연산이 된다.

```
#Calculate,
print(X + Y)
print(X - Y)
print(X * Y)
print(X / Y)
print(X^Y)
print(X %% Y)
print(X %/% Y)

print(X == Y)
identical(X,Y)
print(X >= Y)
```

연산자	결과
X+Y	X와 Y의 합
X-Y	X와 Y의 차
X*Y	X와 Y의 곱
X/Y	X와 Y의 나눗셈
X%%Y	X를 Y로 나눈 나머지
X%/%Y	X를 Y로 나눈 몫
X^Y (= X**Y)	X의 Y거듭제곱
X==Y	X와 Y의 값이 동일한지. (T or F)
X<=Y	X가 Y보다 작은지. (T or F)
X>=Y	X가 Y보다 큰지. (T or F)

R의 자료 구조– 행렬(1)



CognitiveSystems

다음 예시와 같이 행렬을 할당해보자.

```
##4. Matrix
#Generation
X = matrix(data = c(4,7,2,1), nrow = 2, ncol = 2)
Y = matrix(c(2,5,3,7), 2, 2, byrow = TRUE)
I = diag(2)
is.matrix(X)
```

행렬과 차원을 출력해보자.

행렬 역시 indexing을 통해 일부분출력 가능하다.

```
#Print
print(X)
print(Y)
print(dim(X))
print(dim(Y))
print(X[1,1]); print(X[2,2])
print(Y[1:2,1]); print(Y[1,])
print(diag(X))
```

R의 자료 구조– 행렬(2)

CognitiveSystems

행렬 연산 역시 행렬의 차원이 동일해야 가능하며, 각 연산은 Elementwise하게 진행된다.

```
#Calculate,
print(X + Y)
print(X - Y)
print(X * Y)
print(X / Y)
print(X^Y)
print(X %% Y)
print(X %/% Y)
print(X == Y)
identical(X,Y)
print(X >= Y)
```

연산자	결과
X+Y	X와 Y의 합
X-Y	X와 Y의 차
X*Y	X와 Y의 곱
X/Y	X와 Y의 나눗셈
X%%Y	X를 Y로 나눈 나머지
X%/%Y	X를 Y로 나눈 몫
X^Y (= X**Y)	X의 Y거듭제곱
X==Y	X와 Y의 값이 동일한지. (T or F)
X<=Y	X가 Y보다 작은지. (T or F)
X>=Y	X가 Y보다 큰지. (T or F)

R의 자료 구조– 행렬(3)

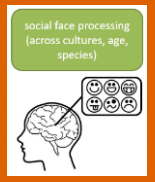
CognitiveSystems

행렬 고유 연산 및 조작을 시도해 보자.

```
#Matrix Calculation
X%*%Y
det(X)
solve(X)
cbind(X,Y)
rbind(X,Y)
```

연산자	결과
X%*%Y	X와 Y의 행렬곱
det(X)	X의 행렬값
solve(X)	X의 역행렬
cbind(X,Y)	X,Y의 열 결합
rbind(X,Y)	X,Y의 행 결합

R의 자료 구조– List



CognitiveSystems

여러가지 형태(숫자, 문자, 행렬 등)를 저장할 수 있는 자료 구조.

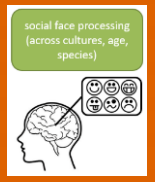
```
## 5. List
List1 = list(c(1,2,3), c("x","y","z"))
List2 = list(vector = c(4,8,6), matrix = matrix(c(1,2,3,4),2), character = c("w","v"))
is.list(List1)
```

List는 index나 List 구성요소의 이름을 이용하여 값을 확인할 수 있다.

List끼리의 연산은 없으며, List 내의 연산은 추후 데이터 핸들링에서 다루어 볼 예정이다.

```
print(List1)
print(List2)
List1[1]          #List1의 첫번째 구성요소 반환
List1[[1]]        #List1의 첫번째 구성요소의 값 반환
List1[[1]][1]     #List1의 첫번째 구성요소값 중 첫번째 값 반환
List2[1]          #List2의 첫번째 구성요소 반환
List2[[2]]        #List2의 두번째 구성요소의 값 반환
List2$matrix      #List2의 matrix이름을 가지는 요소의 값 반환
```

R의 자료 구조– Array



CognitiveSystems

행렬을 다차원으로 확장하여 3차원 이상의 차원을 갖는 자료 구조.

> array(data = X, dim = c(dim1,dim2,…), dimnames = list(이름1,이름2,…))

```
## 6. Array
arr1 = array(data = 1:8, dim = c(2,2,2))
arr2 = array(8:1, c(2,2,2), dimnames = list(c("row1","row2"),c("col1","col2"),c("dim1","dim2")))
is.array(arr1)
```

Array의 차원을 확인하고, 각 원소들을 indexing이나 dimnames을 이용하여 확인할 수 있다.

** attributes: 해당 자료구조의 속성을 확인하는 함수, 2차원 이상(Matrix, List, Array) 자료구조의 속성을 확인하는데 주로 이용.

```
print(arr1)                                  #Array1 전체 출력
print(arr2)                                  #Array2 전체 출력
print(dim(arr1))                             #Array1 차원 출력
print(dim(arr2))                             #Array2 차원 출력
attr(arr1, 'dim')                            #Array1 차원 속성 출력
attr(arr2, 'dimnames')                       #Array2 차원이름 속성 출력
print(arr1[,,1])                             #Array1 첫번째 dime 출력
print(arr1[2,,2])                            #Array1 두번째 dim, 두번째 row 출력
print(arr2[,,'dim2'])                        #Array2 dim2 원소 출력
print(arr2['row1','col2','dim2'])            #Array2 row1, col2, dim2 원소 출력
```

R의 자료 구조– Array(2)

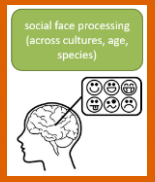
CognitiveSystems

Array의 연산은 array의 차원이 동일해야 가능하며, 각 연산은 Elementwise하게 진행된다.

```
#Calculate,
print(X + Y)
print(X - Y)
print(X * Y)
print(X / Y)
print(X^Y)
print(X %% Y)
print(X %/% Y)
print(X == Y)
identical(X,Y)
print(X >= Y)
```

연산자	결과
arr1+arr2	arr1와 arr2의 합
arr1-arr2	arr1와 arr2의 차
arr1*arr2	arr1와 arr2의 곱
arr1/arr2	arr1와 arr2의 나눗셈
arr1%%arr2	arr1를 arr2로 나눈 나머지
arr1%/%arr2	arr1를 arr2로 나눈 몫
arr1^arr2	arr1의 arr2거듭제곱
arr1==arr2	arr1와 arr2의 값이 동일한지
arr1<=arr2	arr1가 arr2보다 작은 지
arr1>=arr2	arr1가 arr2보다 큰지

R의 자료 구조– Factor



CognitiveSystems

R에만 있는 Categorical을 표현하기 위한 변수.

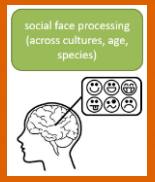
- 명목형 변수의 Dummy화
- 각 명목들의 순서를 설정 가능

> factor(character_vector)

$$Gender = \begin{cases} 1\ if\ male \\ 0\ if\ female \end{cases}$$

```
## 7. Factor
gender = c('Male','Female','Female','Female','Male')
gender_fac = factor(gender)
print(gender)
print(gender_fac)

grade = c('Excellent!!','Great!',"Good","So so","Oops!","Bad..")
grade_fac = factor(grade)
grade_order = factor(grade, ordered = TRUE, levels = grade)
print(grade)
print(grade_fac)
print(grade_order)
```

R 문법 – 조건문(1)



CognitiveSystems

1. If ~ else 문
if 뒤에 따르는 조건에 따라 실행

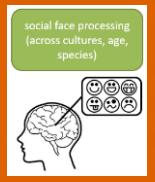
> if(조건식) 참일 때 실행되는 내용

> if(조건식1){
조건식 1이 참일 때 실행되는 내용
} else if(조건식2){
조건식 2가 때 실행되는 내용
} else{
 거짓일 때 실행되는 내용
}

```r
## If
x = 3
y = 2

if(x>y){
  print("x is greater than y")
}

if(x>y) print("x is greater than y")

if(x>y){
  print("x is greater than y")
} else if(x<y){
  print("x is less than y")
} else{
  print("x is equal to y")
}

z=c(1,2,3,4,5)
ifelse(z%%2==0, 'even', 'odd')
```

> ifelse(벡터조건, 참일 때 내용, 거짓일 때 내용)

R 문법 – 조건문(2)



CognitiveSystems

2. Switch

여러가지 조건을 비교해야 할 경우, switch를 이용하여 간소화할 수 있다.

> switch(자연수, 실행1, 실행2, …, 실행n)

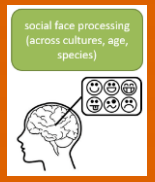
: 자연수에 해당하는 실행문을 실행함.

> switch(문자, 문자1, 문자2, …, 문자n, 문자가 없을 경우)

: 문자에 맞는 곳의 실행문을 실행함. 해당 문자가 없을 경우 마지막 실행문을 실행함.

```
## Switch
x = 3
switch(x,"하나","둘","셋","넷")

str = "R"
switch(str,
       "C++" = print("이 언어는 C++입니다."),
       "R" = print("이 언어는 R입니다."),
       "java" = print("이 언어는 java입니다."),
       print("모르는 언어입니다.")
       )
```

R 문법 – 반복문(1)



CognitiveSystems

1. For loop

같은 실행문을 여러 번 반복

> for(i in vector){반복할 실행문}

2. While loop

조건문이 참이 될 때까지 반복문을 실행

> while(조건문){반복할 실행문}

3. Repeat

조건 없는 반복실행.

> repeat{반복할 실행문}

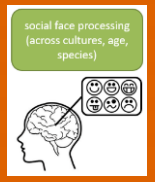
*** 반드시 break를 실행문 안에 설정할 것!

```
## For
x = c(1:10)
sum = 0
for(i in x){
  sum = sum + i
  print(sum)
}
```

```
## while
x = 0
y = 0
while(x<20){
  x = x+2
  y = y+10
  print(x)
}
```

```
## Repeat
x = 0
y = 0
repeat{
  x = x+2
  y = y+10
  print(x)
  if (x>20)
    break
}
```

R 문법 – 반복문(2)



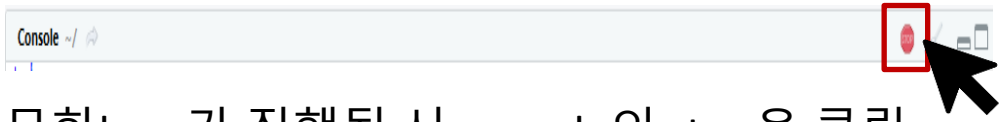
CognitiveSystems

break, next

**break: break를 만나면 loop를 중단하고 나옴.

next: next를 만나면 이후의 실행을 생략하고
반복문 처음으로 다시 돌아감.

조건을 만족하지 않을 시 while과 repeat는
무한히 반복할 수 있음.
따라서 break나 next 등으로 적절히 조절을 하여
무한루프를 피해야 한다.

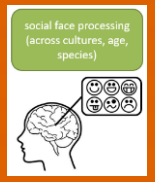


무한loop가 진행될 시 console의 stop을 클릭

```r
## while break
x = 0
y = 0
while(x<20){
  x = x+2
  y = y+10
  print(x)
  if(y>30)
    break
}
```

```r
## while next
x = 0
y = 0
while(x<20){
  x = x+2
  y = y+10
  print('x result')
  print(x)
  if(y>30)
    next
  print('y result')
  print(y)
}
```

R 문법 – 함수



CognitiveSystems

function을 통하여 사용자가 원하는 매서드를 만들 수 있다.

>function(변수){
변수가 들어간 실행문들
return(결과값)
}

return이 있을 경우, 함수의 결과값을 변수로 받을 수 있다.

```
## function
calculator = function(x,y){
  return(x+y)
}
result = calculator(1,2)
print(result)

printer = function(str){
  print(str)
}
printer("Hello World!")
```

3. 데이터 프레임

데이터 프레임이란?

CognitiveSystems

R에서 가장 널리 사용되는 행과 열을 가지는 2차원의 표 형태의 자료구조.

- 행렬: 모든 원소가 같은 자료형
- 데이터 프레임: 각 열마다 다른 자료형을 가질 수 있음

R의 dataset, MASS와 같은 패키지에 다양한 데이터프레임의 예시를 확인할 수 있다.

>data()

```
Data sets in package ¡®MASS¡¯:

Aids2                           Australian AIDS Survival Data
Animals                         Brain and Body Weights for 28 Species
Boston                          Housing Values in Suburbs of Boston
Cars93                          Data from 93 Cars on Sale in the USA in 1993
Cushings                        Diagnostic Tests on Patients with Cushing's Syndrome
DDT                             DDT in Kale
GAGurine                        Level of GAG in Urine of Children
Insurance                       Numbers of Car Insurance claims
Melanoma                        Survival from Malignant Melanoma
OME                             Tests of Auditory Perception in Children with OME
```

데이터 프레임 - 예시

CognitiveSystems

다음은 R에서 기본으로 제공하는 Edgar Anderson's Iris Data을 출력한 것이다.

```
> head(iris, 10)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
```

View(iris)를 입력하면 script창에서 데이터 프레임을 자세히 확인 할 수 있다.

2_Data_Frame.R × iris ×

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa

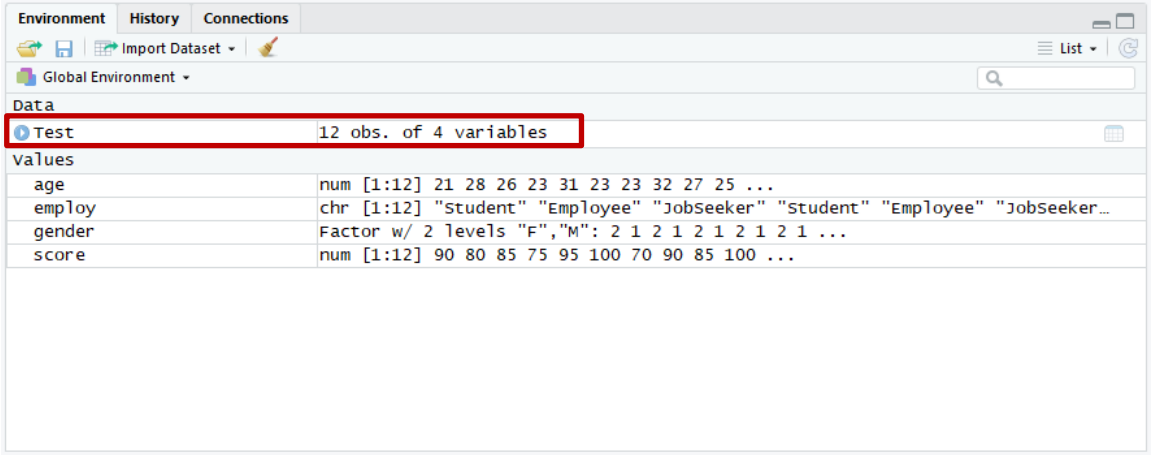
데이터 프레임 – 생성

각 열에 들어갈 벡터들을 생성하고, 이들을 열 이름으로 정의하여 데이터 프레임을 생성

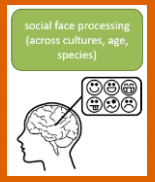
> data.frame(colname1 = vec1, colname2 = vec2, …)

```
## 1. Data Frame Basic
gender = rep(factor(c('M','F')),6)
employ = rep(c('Student', 'Employee','JobSeeker'), 4)
age = c(21,28,26,23,31,23,23,32,27,25,26,25)
score = c(90, 80, 85, 75, 95, 100, 70, 90, 85, 100, 65, 85)

Test = data.frame(GENDER = gender, EMP = employ, AGE = age, SCORE = score)
```

```
> print(Test)
   GENDER        EMP AGE SCORE
1       M    Student  21    90
2       F   Employee  28    80
3       M JobSeeker  26    85
4       F    Student  23    75
5       M   Employee  31    95
6       F JobSeeker  23   100
7       M    Student  23    70
8       F   Employee  32    90
9       M JobSeeker  27    85
10      F    Student  25   100
11      M   Employee  26    65
12      F JobSeeker  25    85
```



데이터 프레임 – 불러오기



CognitiveSystems

외부에서 저장된 csv(comma-separate values) 파일을 불러들여보자.

> read.csv("데이터 이름.csv", header = TRUE, stringAsFactor = TRUE)

```
load_data = read.csv("example_data.csv", header=TRUE)
head(load_data)
```

```
  Rank  Restaurant Sales YOY_Sales Units YOY_Units Headquarters          Segment_Category
1    1  McDonald's 40412     4.90% 13846    -0.50%          <NA>      Quick Service & Burger
2    2   Starbucks 21380     8.60% 15049     3.00%          <NA> Quick Service & Coffee Cafe
3    3 Chick-fil-A 11320    13.00%  2470     5.00%          <NA>     Quick Service & Chicken
4    4   Taco Bell 11293     9.00%  6766     2.70%          <NA>     Quick Service & Mexican
5    5 Burger King 10204     2.70%  7346     0.20%          <NA>      Quick Service & Burger
6    6       Subway 10200    -2.00% 23801    -4.00%          <NA>    Quick Service & Sandwich
```

그 외에도 txt, excel와 같은 파일들도 불러들일 수 있다.

>read.table , read.xlsx(xlsx 라이브러리 필요), …

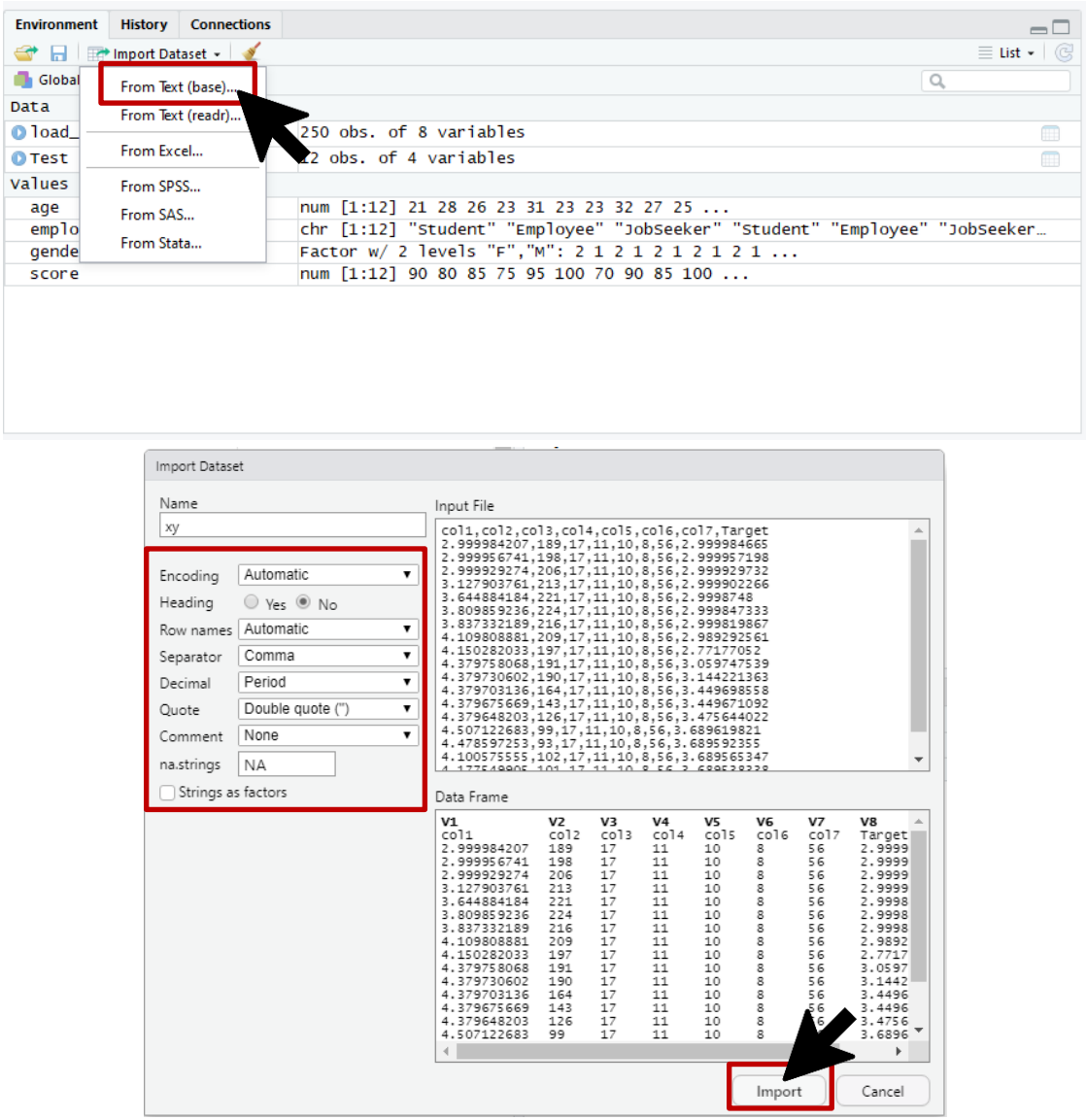
출처: https://www.kaggle.com/michau96/restaurant-business-rankings-2020

참고)데이터 불러오기

다음은 코드 없이 Environment 화면에서 데이터를 불러오는 과정이다

.

불러오고자 하는 데이터를 선택하고, option들을 설정해주면 데이터 프레임 형태로 불러올 수 있다.



데이터 프레임 – 내보내기

CognitiveSystems

write 함수를 이용해 R에서 생성한 data.frame을 외부로 내보낼 수 있다.

> write.csv(data.frame, “경로/데이터 이름.csv”, …)

그 외에도 txt, excel와 같은 파일형태로 내보낼 수 있다.

>write.table , write.xlsx(xlsx 라이브러리 필요), …

```
## Save Data Frame
if(!dir.exists("save_data")){  ## 해당 폴더가 없으면 폴더 생성
  dir.create("save_data")
}
write.csv(Test, 'save_data/mydata.csv')
write.table(Test, 'save_data/mydata.txt', sep = "\t")
```

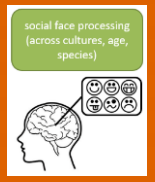
	A	B	C	D	E
1		GENDER	EMP	AGE	SCORE
2	1	M	Student	21	90
3	2	F	Employee	28	80
4	3	M	JobSeeker	26	85
5	4	F	Student	23	75
6	5	M	Employee	31	95
7	6	F	JobSeeker	23	100
8	7	M	Student	23	70
9	8	F	Employee	32	90
10	9	M	JobSeeker	27	85
11	10	F	Student	25	100
12	11	M	Employee	26	65
13	12	F	JobSeeker	25	85

mydata.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

"GENDER"		"EMP"	"AGE"	"SCORE"
"1"	"M"	"Student"	21	90
"2"	"F"	"Employee"	28	80
"3"	"M"	"JobSeeker"	26	85
"4"	"F"	"Student"	23	75
"5"	"M"	"Employee"	31	95
"6"	"F"	"JobSeeker"	23	100
"7"	"M"	"Student"	23	70
"8"	"F"	"Employee"	32	90
"9"	"M"	"JobSeeker"	27	85
"10"	"F"	"Student"	25	100
"11"	"M"	"Employee"	26	65
"12"	"F"	"JobSeeker"	25	85

데이터 프레임 – 조작(1)



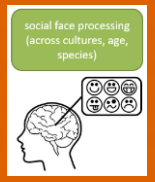
CognitiveSystems

다음 예시들은 데이터 테이블에 대한 정보들을 제공한다.

>summary(data.frame) : 데이터 프레임의 기초 통계량 출력
>str(data.frame): 데이터 테이블의 구조 출력
>attributes(data.frame): 데이터 프레임의 속성 출력
>attr(data.frame, attribute): 데이터 프레임이 가지고 있는 속성값 출력

```
##Print Data Frame
summary(Test)
str(Test)
attributes(Test)
attr(Test,'names')
```

데이터 프레임 – 조작(2)



CognitiveSystems

>head(data.frame, n = 5) : 위에서부터 n개의 행 출력

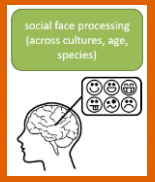
>tail(data.frame, n = 5): 아래서부터 n개의 행 출력

데이터 프레임 역시 list와 array처럼 indexing으로 각 원소들을 출력할 수 있다.

또한, 출력 방식에 따라 data.frame의 형식, 또는 벡터 형식으로 출력 가능하다.

```
head(Test)              #위에서부터 5개의 행
tail(Test,6)            #아래에서부터 6개의 행
Test[1]                 #첫번째 열
Test[c(2,4)]            #2,4번째 열
Test$GENDER             #GENDER 열 백터출력
Test$SCORE              #SCORE열 백터출력
Test[["AGE"]]           #AGE 열 백터출력
Test[1,]                #첫번째 행
Test[3,2]               #세번째 행, 두번째 열
Test[c(5,7),c(1,3)]     #5,7번째 행, 1,3번째열
Test[1:10,2:3]          #1~10행, 2~3 열
```

데이터 프레임 – 통계



CognitiveSystems

Aggregate: 집계 함수

원하는 컬럼을 기준으로 합, 평균, 분산 등의 집계 함수를 이용

> aggregate(x~y, data = data.frame, FUN)

FUN: mean, median, mode, sd, …

```
> aggregate(SCORE~GENDER,data = Test, mean) #성별 별 성적 평균
  GENDER    SCORE
1      F 88.33333
2      M 81.66667
> aggregate(SCORE~EMP,data = Test, max)     #직업 별 성적 최댓값
        EMP SCORE
1  Employee    95
2 JobSeeker   100
3   Student   100
> aggregate(.~EMP,data = Test, mean)        #직업 별 모든 평균
        EMP GENDER   AGE SCORE
1  Employee    1.5 29.25 82.50
2 JobSeeker    1.5 25.25 88.75
3   Student    1.5 23.00 83.75
```

데이터 프레임 – 정렬

CognitiveSystems

> sort: 각 데이터들을 정렬

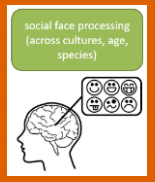
> order: 각 데이터들의 순서(행)을 반환

```
## Sort
sort_gender = sort(Test$GENDER)                    #정렬 내용 확인
ord_gender = order(Test$GENDER)                    #정렬된 값의 원래 위치
Test[ord_gender,]                                  #정렬 결과

order_EMPGEN = order(Test$EMP, Test$GENDER, decreasing = c(TRUE, FALSE)) #Emp, Gender에 대해 정렬
Test[order_EMPGEN,]
```

```
> Test[ord_gender,]
   GENDER        EMP AGE SCORE
2       F   Employee  28    80
4       F    Student  23    75
6       F JobSeeker  23   100
8       F   Employee  32    90
10      F    Student  25   100
12      F JobSeeker  25    85
1       M    Student  21    90
3       M JobSeeker  26    85
5       M   Employee  31    95
7       M    Student  23    70
9       M JobSeeker  27    85
11      M   Employee  26    65
```

```
> Test[order_EMPGEN,]
   GENDER        EMP AGE SCORE
1       M    Student  21    90
7       M    Student  23    70
4       F    Student  23    75
10      F    Student  25   100
3       M JobSeeker  26    85
9       M JobSeeker  27    85
6       F JobSeeker  23   100
12      F JobSeeker  25    85
5       M   Employee  31    95
11      M   Employee  26    65
2       F   Employee  28    80
8       F   Employee  32    90
```

참고)데이터 테이블



CognitiveSystems

데이터 프레임과 유사하지만 보다 빠른 grouping과 ordering, 짧은 문장 지원

하나하나 모든 자료를 비교해서 찾는 vector search를 하는 데이터 프레임과는 달리, 데이터 테이블은 binary search를 하기 때문에 대용량 데이터를 분석할 때 주로 이용.

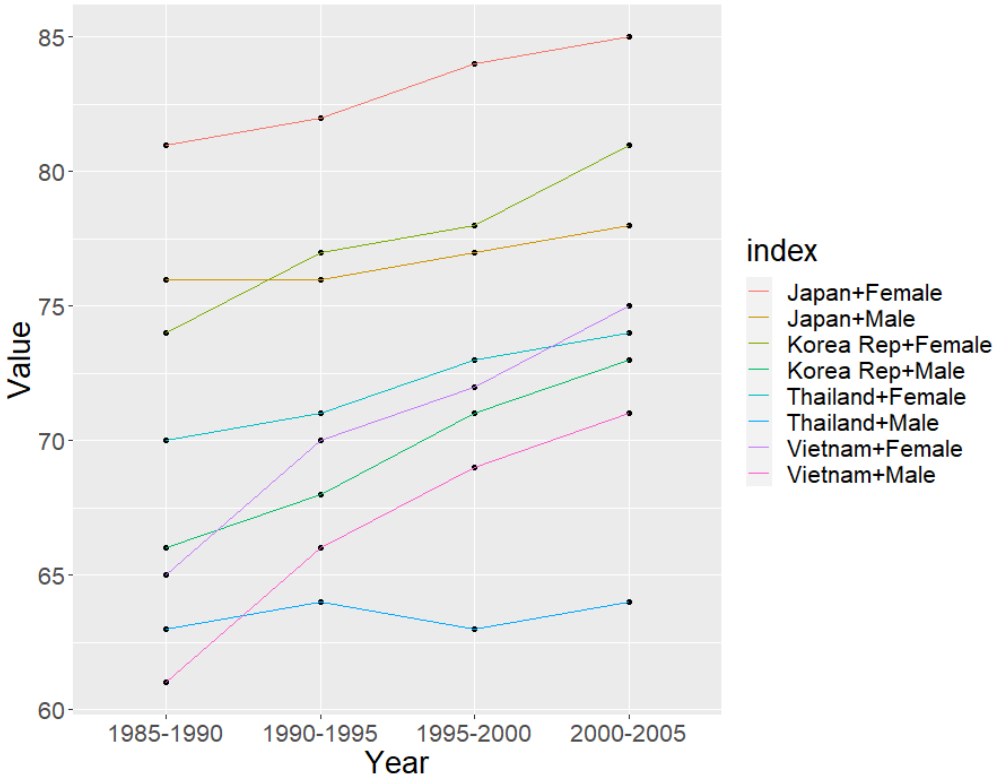
>data.table(colname1 = vec1, colname2 =vec2, …)

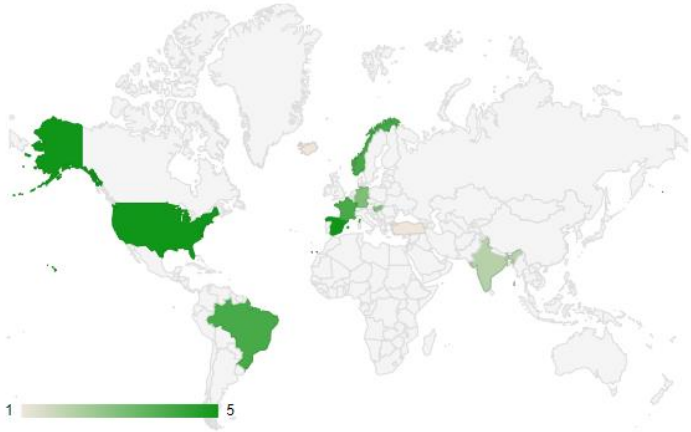
>data.table(data.frame)

4. 시각화

시각화

수치로 된 데이터들을 시각적으로 표현하여 보다 분석의 insight를 효율적으로 얻기 위함.





Plot 함수

CognitiveSystems

Plot 함수는 R에서 제공하는 기본 시각화 함수로, 할당된 데이터를 받아 그래프를 그려준다. 제시되는 그래프들은 받은 데이터의 종류에 따라 다르다.

>plot(x, y, …)

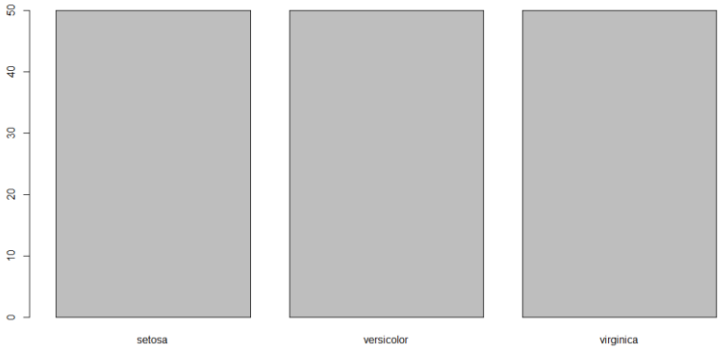
- y가 없을 시 단일 변수에 대한 그래프
- y가 있을 시 두 변수 사이의 그래프

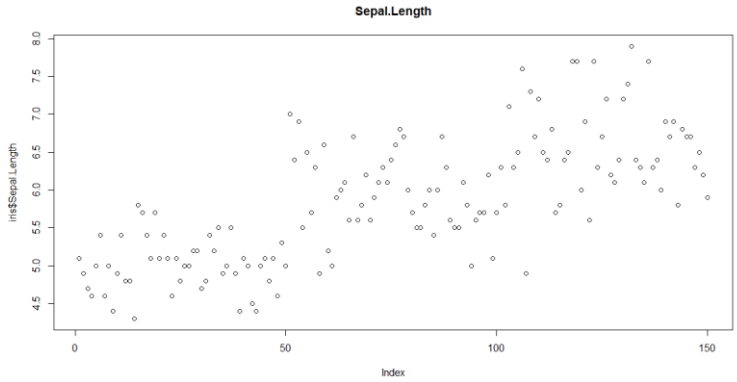
설정변수	변수 내용
main	그래프의 제목
sub	그래프의 부제목
xlab, ylab	X, y축의 라벨
xlim, ylim	X, y축의 상(하)한선
col	선의 색상 ex) col = “red”
type	선의 종류, p: 점, l: 선
lty	선 타입 1: 실선 2: 긴 점선, 3: 짧은 점선

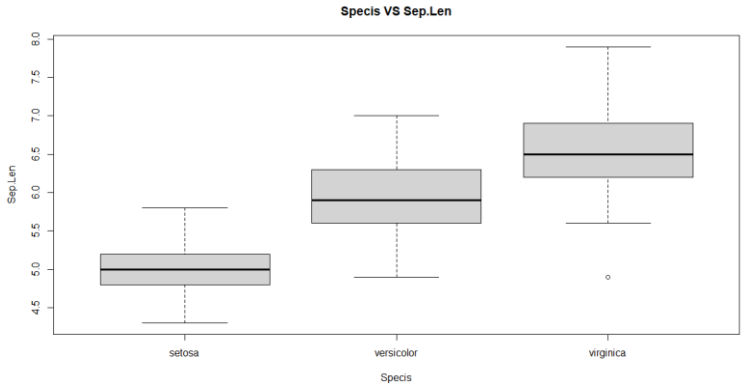
Plot 함수

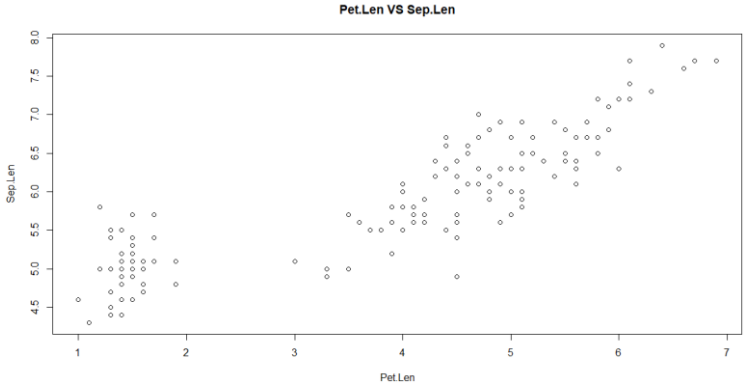
Iris 데이터를 시각화 해보자.

```
#Plot1
plot(iris$Species)
#Plot2
plot(iris$Sepal.Length, main = 'Sepal.Length')
#Plot3
plot(iris$Species, iris$Sepal.Length, main = 'Specis VS Sep.Len', xlab = 'Specis', ylab = 'Sep.Len')
#Plot4
plot(iris$Petal.Length, iris$Sepal.Length, main = 'Pet.Len VS Sep.Len', xlab = 'Pet.Len', ylab = 'Sep.Len')
```







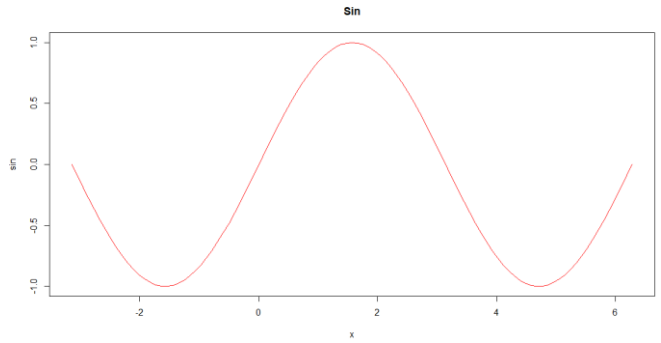


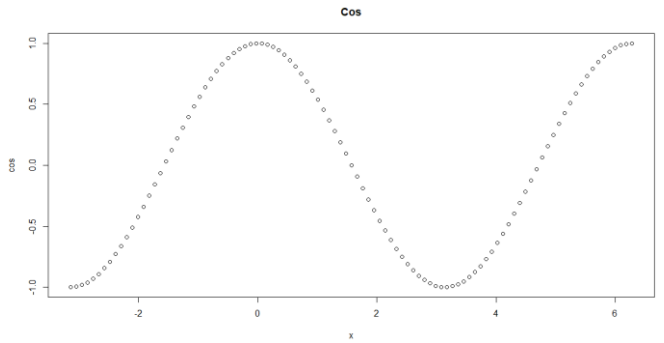
Plot 함수

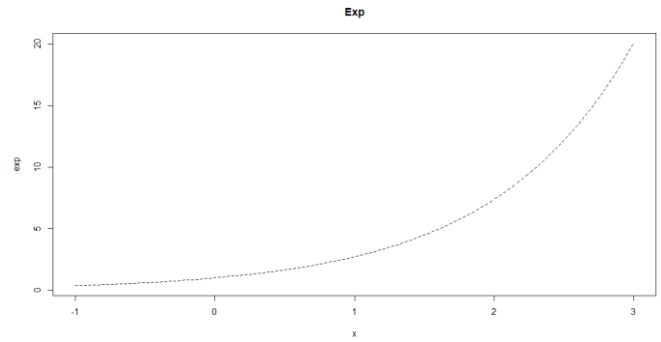
Plot는 수학 함수 또한 출력이 가능하다.

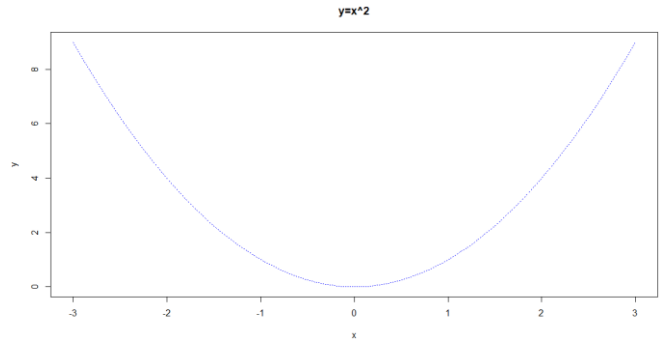
>plot(fun, a, b, …): [a,b] 구간 내에서의 함수 그래프 출력

```
#Plot function
#Sine
plot(sin, -pi, 2*pi, main = "Sin", col = 'red')
#Cosine
plot(cos, -pi, 2*pi, main = "Cos", type = "c")
#Exponential
plot(exp, -1, 3, main = "Exp", lty = 2)
#Custom
curve = function(x){x^2}
plot(curve, -3, 3, main = "y=x^2", col = 'blue', lty = 3)
```







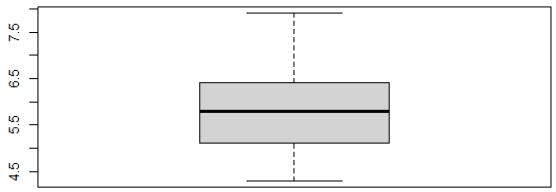


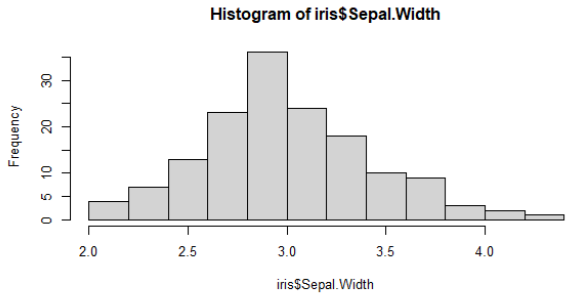
Plot 함수

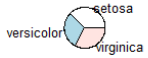
Plot함수는 받는 데이터에 따라 그래프가 변한다.
다음 함수들을 통해 원하는 그래프를 그릴 수 있다.

>hist(x,…): 히스토그램
>barplot(x,…): 막대 그래프
>boxplot(x,y,…): 박스그래프
>pie(x,y, …): 원형 그래프

```
boxplot(iris$Sepal.Length)   #plot1
hist(iris$Sepal.Width)       #plot2
wid_mean = aggregate(Sepal.Width~Species,data = iris, mean)
barplot(Sepal.Width~Species, data = wid_mean) #plot3
pie(wid_mean$Sepal.Width,wid_mean$Species, radius = 2) #plot4
```







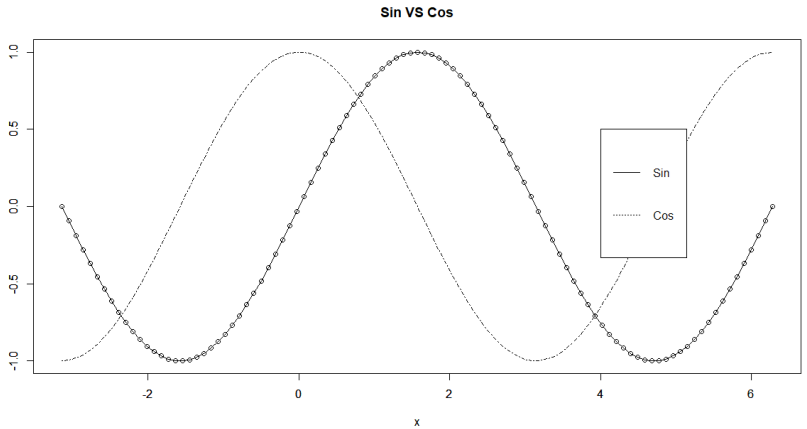
Partition

par(): 동시에 여러 그래프를 그리는 것이 가능.

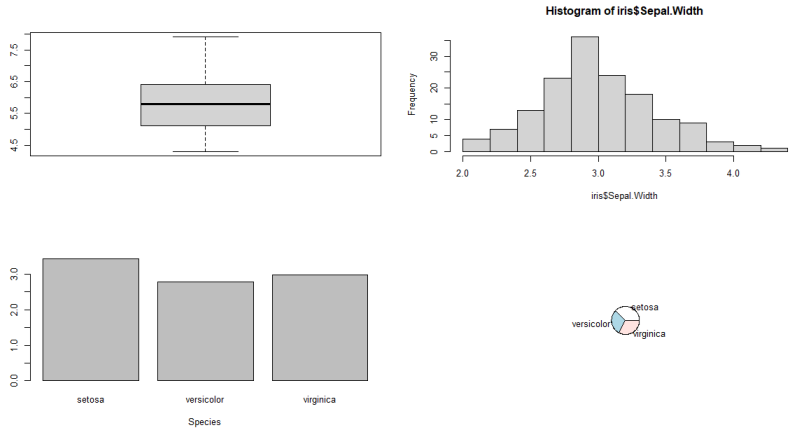
>par(new = TRUE): 기존 그려진 그래프 위에 새로운 그래프를 출력

>par(mfrow = c(m,n)): mxn개의 그래프를 출력

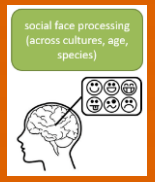
```
plot(sin, -pi, 2*pi, main = "Sin VS Cos", ylab = "", type = "o")
par(new = TRUE)
plot(cos, -pi, 2*pi, ylab = "", lty = 4)
legend(4, 0.5, c("Sin", "Cos"), lty = c(1,3))
```




```
par(mfrow = c(2,2))
boxplot(iris$Sepal.Length)  #plot1
hist(iris$Sepal.Width)      #plot2
wid_mean = aggregate(Sepal.Width~Species,data = iris, mean)
barplot(Sepal.Width~Species, data = wid_mean) #plot3
pie(wid_mean$Sepal.Width,wid_mean$Species, radius = 2) #plot4
```



ggplot2



CognitiveSystems

ggplot2는 “grammar of graphics” 개념을 근거로 하고 있어 문법 내에서 코드를 추가하거나 삭제하야, R에서 제공하는 기본 함수에 비해 그래프 변형이 용이하고, 더 시각적으로 표현할 수 있다.

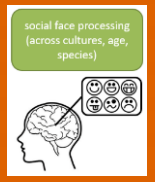
또한, ggplot2를 이용하여 R의 기본 함수에는 없는 새로운 그래프들도 그릴 수 있다.

** ggplot2 reference: https://ggplot2.tidyverse.org/reference/

>install.packages(“ggplot2”)

>library(ggplot2)

ggplot2



CognitiveSystems

ggplot2은 데이터 프레임에서부터 기본 속성을 먼저 정의하고, 원하는 속성들을 +을 이용해 추가하는 방식으로 동작한다.

>ggplot2(data = data.frame, mapping = aes(x =xcol, y =ycol, ...)) *기본 형태

+ geom_*(...) : 그래프의 개형 결정(point, line, histogram 등)

+ stat_* (...) : 통계 및 함수 사용

+ scale_* (...) : 그래프의 시각 속성 부여(크기, 모양, 위치 등)

+ coord_*(...) : 좌표계 설정(cartesian, polar)

...

ggplot2 - 개형

CognitiveSystems

+ geom_*(…) : 그래프의 개형 결정(point, line, histogram 등)

설정변수	결과
geom_point()	산점도(Scatter plot)
geom_line()	선형 그래프(Line plot)
geom_h(v)line	수직(수평) 그래프
geom_histogram()	히스토그램
geom_boxplot()	상자 그래프(Box plot)
geom_bar()	바형 그래프(Bar plot)

ggplot2 – 개형(2)

CognitiveSystems

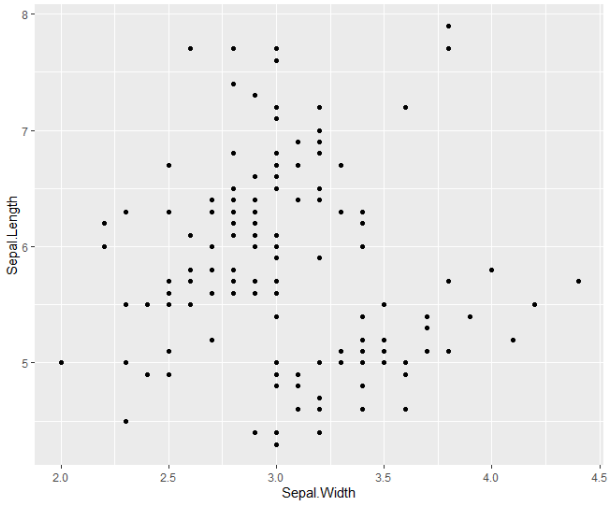
앞서 생성한 base들을 기반으로 그래프를 시각화

```
## ggplot2

# Import library ggplot2
library(ggplot2)

base1 = ggplot(data = iris, mapping = aes(x = Sepal.Width, y = Sepal.Length)) +
  theme(text = element_text(size=25))
#1
base1 + geom_point() + ggtitle("Scatter")            #산점도
#2
base1 + geom_line()  + ggtitle("Line")               #선 그래프
#3
base1 + geom_point() + geom_line() + ggtitle("Both") #동시에
#4
base1 + geom_point() + ggtitle("Scatter with hline") +
  geom_hline(yintercept = mean(iris$Sepal.Length)) #수평선 추가
#5
base2 = ggplot(data = iris, mapping = aes(y = Species, x = Sepal.Width))+
  theme(text = element_text(size=25))
base2 + geom_boxplot(notch = TRUE) + ggtitle("Box")  #박스 그래프
base2 + geom_violin() + ggtitle("Violin")            #바이올린 그래프
#6
base3 = ggplot(data = iris, mapping = aes(x = Petal.Length))+
  theme(text = element_text(size=25))
base3 + geom_histogram(binwidth = 1) +ggtitle("Histogram") #히스토그램
```

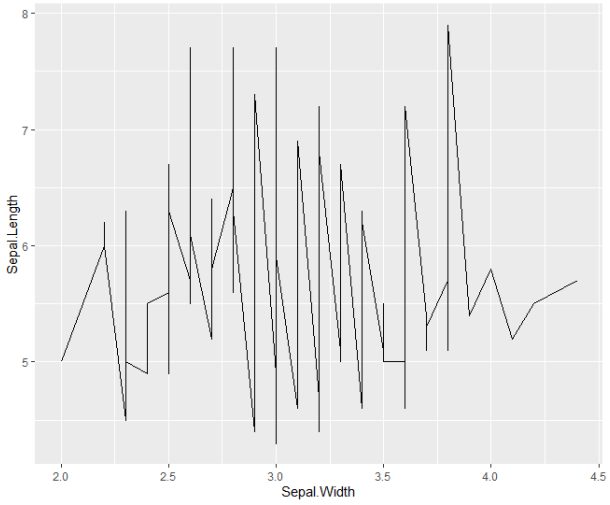
ggplot2 – 개형(3)

CognitiveSystems

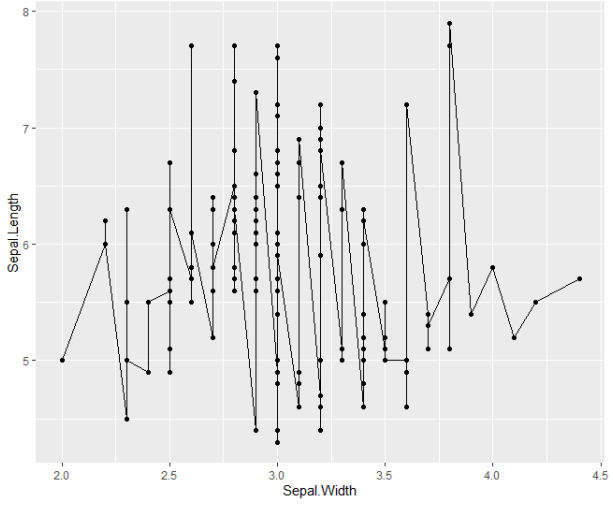
Scatter



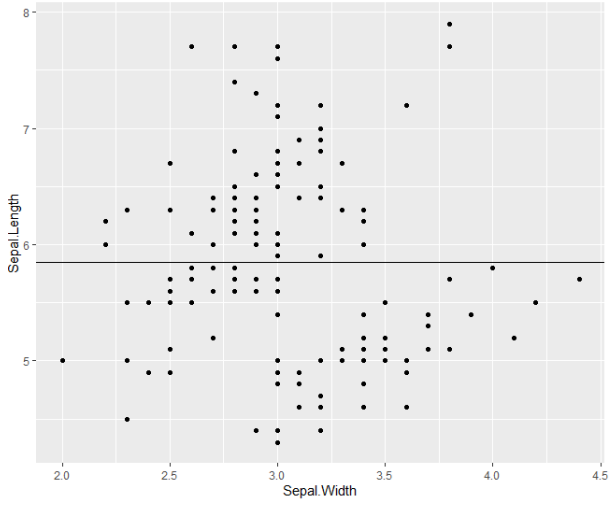
Line



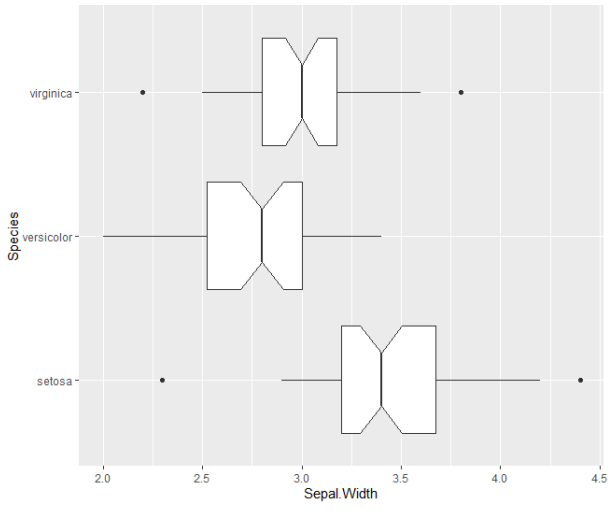
Both



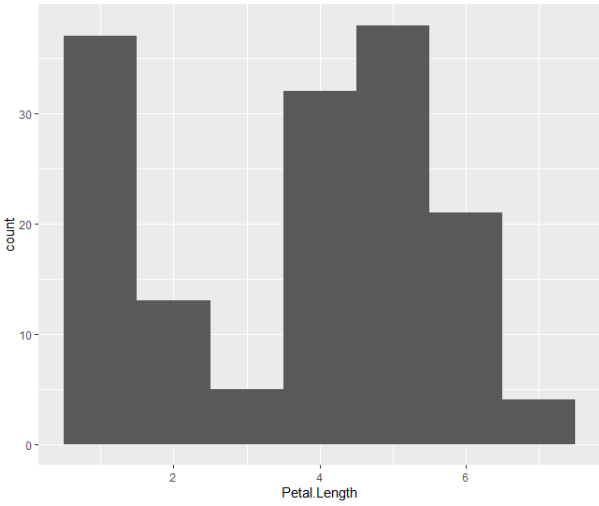
Scatter with hline



Box



Histogram



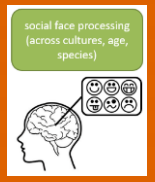
ggplot2 – 통계 및 함수

CognitiveSystems

+ stat_*(...) : 통계 및 함수 사용

설정변수	결과
stat_function()	함수 사용
stat_summary()	통계 요약 사용
stat_summary_2d()	3차원 내용을 2차원으로 요약(Heatmap)
stat_ellipse()	타원형 군집화

ggplot2 – 통계 및 함수(2)

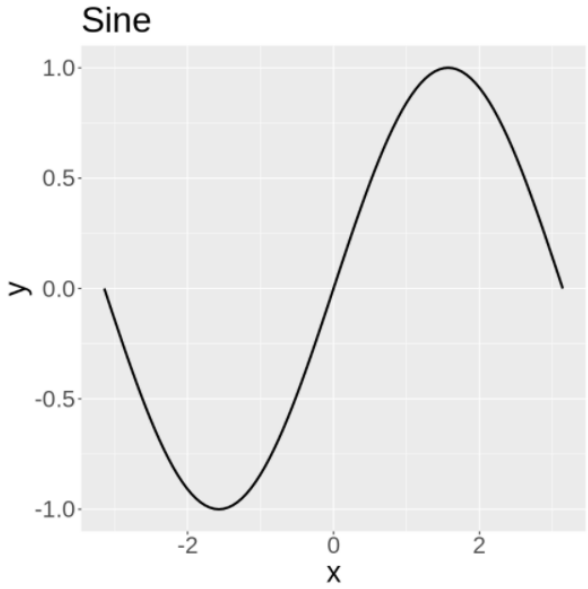


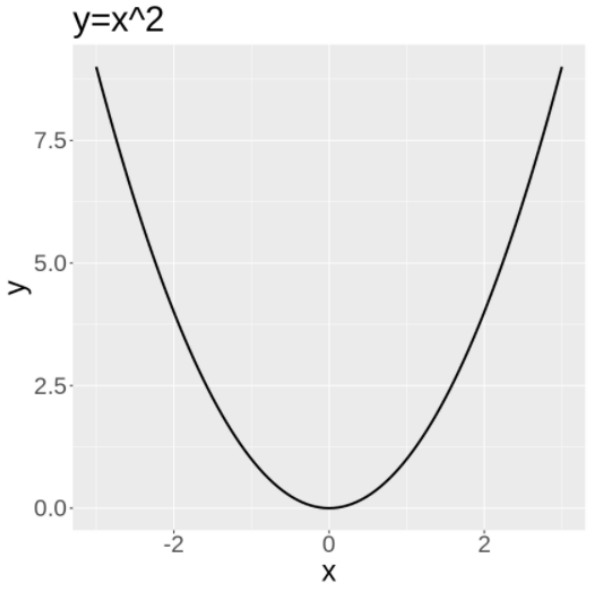
CognitiveSystems

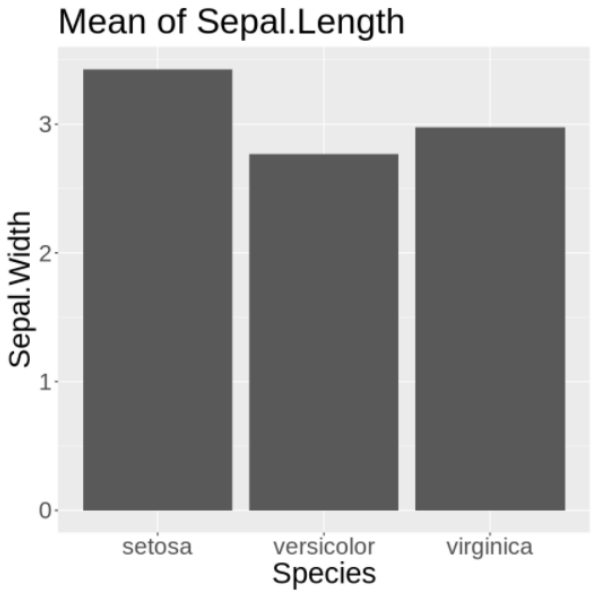
앞서 생성한 base들을 기반으로 그래프를 시각화

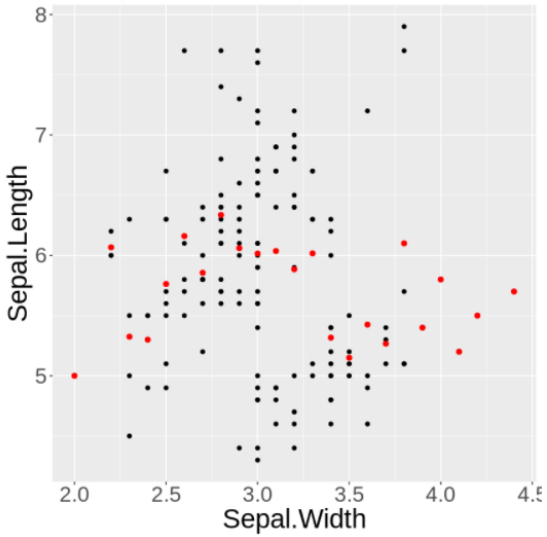
```
## Stat
#1
ggplot(data.frame(x = c(-pi,pi)), aes(x=x)) +
  stat_function(fun = sin, size = 1) + ggtitle("Sine") +
  theme(text = element_text(size=25))                          #Sine Function
#2
curve = function(x){return(x^2)}
ggplot(data.frame(x = c(-3,3)), aes(x=x)) +
  stat_function(fun = curve, size = 1) + ggtitle("y=x^2") +
  theme(text = element_text(size=25))                          #y=x^2 function
#3
base1 + geom_point() + stat_summary(fun = "mean", colour = "red", size = 2, geom = "point")
#4
base2 + stat_summary(aes(y = Sepal.Width), fun = "mean", geom = "bar") +
  ggtitle("Mean of Sepal.Length")
#5
base4 = ggplot(data = iris, mapping = aes(x = Sepal.Width, y = Sepal.Length , z= Petal.Width)) +
  theme(text = element_text(size=25))
base4 + stat_summary_2d() + ggtitle("Heatmap")                 #Heatmap
#6
base5 = ggplot(data = iris, mapping = aes(x = Sepal.Width, y = Sepal.Length, color = Species)) +
  theme(text = element_text(size=25))
base5 + geom_point() + stat_ellipse() + ggtitle("Cluster by Species")
```

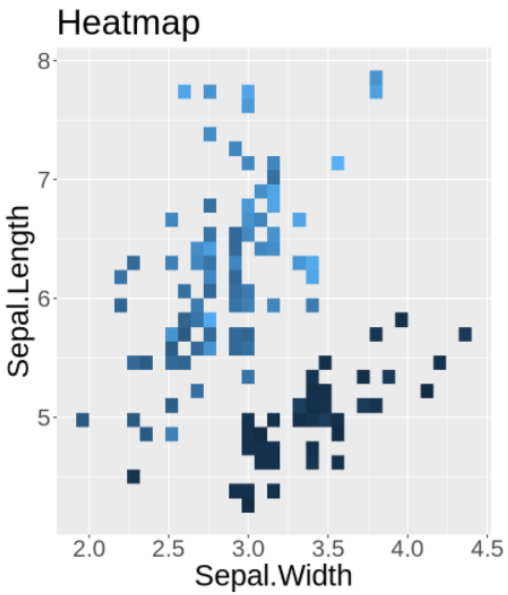
ggplot2 – 통계 및 함수(3)

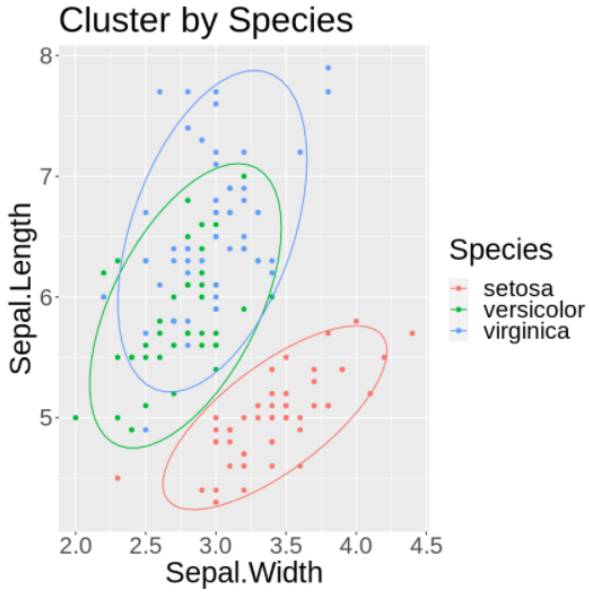












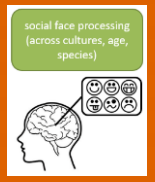
ggplot2 - 스케일

CognitiveSystems

+ scale_* (…) : 그래프의 시각 속성 부여(크기, 모양, 위치 등)

설정변수	결과
labs(), xlab(), ylabs=()	Label 설정(메인, x축, y축)
xlim(), ylim()	x축 y축의 범위
ggtitle()	Plot 메인 제목
scale_alpha()	그래프의 투명도
scale_colour	그래프의 색깔 변경 _hue(): 진하기, _grey(): 흑백, …
scale_x(y)_bin()	x(y)축 값의 너비 조정
scale_x(y)_continuous()	x(y)축 값을 연속형으로 변환
scale_x(y)_time()	x(y)축 값을 시간으로 변환

ggplot2 – 스케일(2)

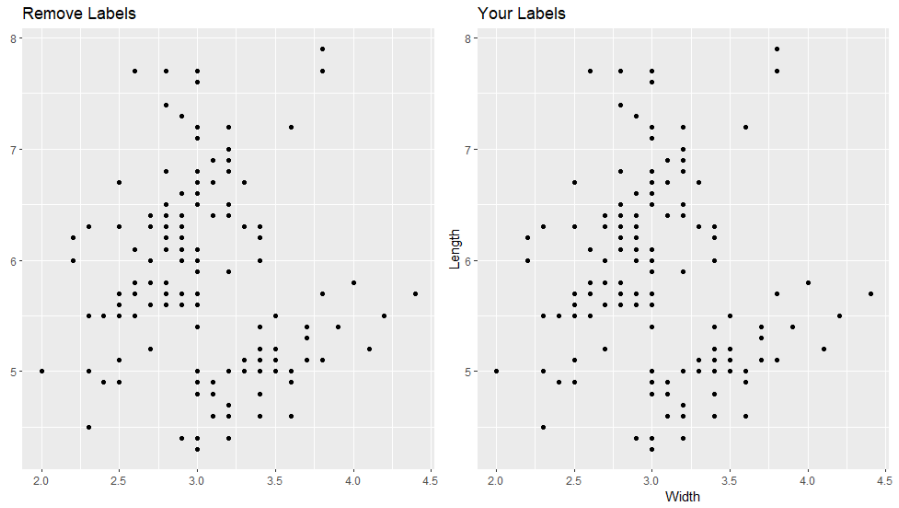


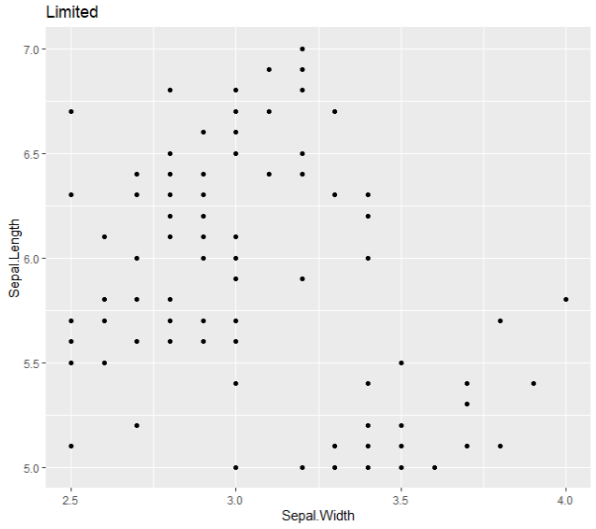
CognitiveSystems

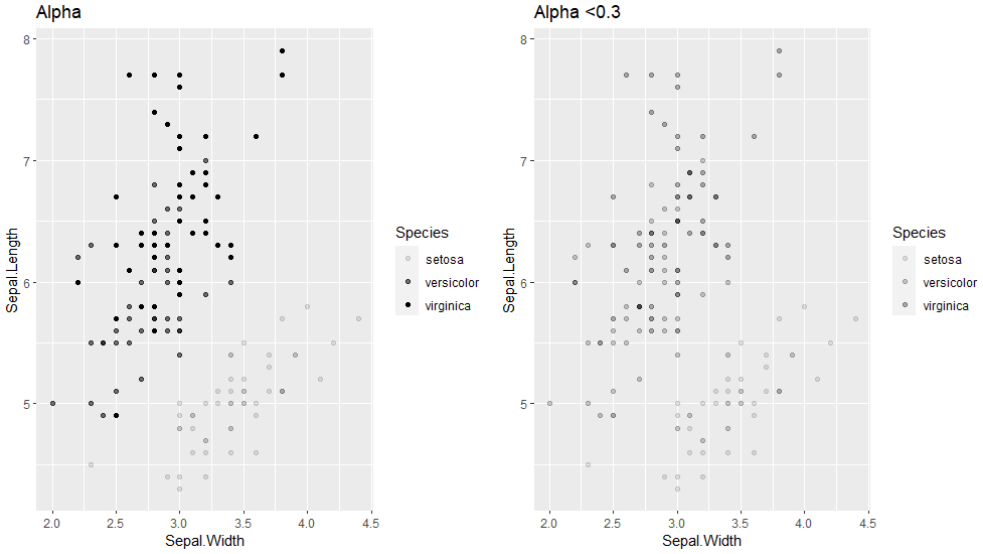
앞서 생성한 base들을 기반으로 그래프의 속성 변경

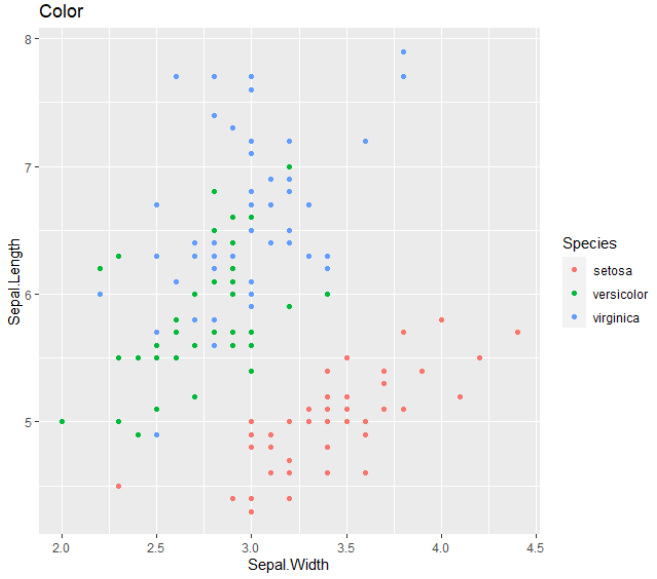
```
## Scale
install.packages("patchwork")
library(patchwork)

base1 + geom_point() + ggtitle("Remove Labels") +  #Label 제거
  xlab("") + ylab("")
base1 + geom_point() + ggtitle("Your Labels") +    #Labe 변경
  xlab("Width") + ylab("Length")
base1 + geom_point() + ggtitle("Limited") +        #범위 변경
  xlim(c(2.5,4.0)) + ylim(c(5,7))
base1 + geom_point(aes(alpha = Species)) + ggtitle("Alpha")+ #Species마다 진하기 변경
base1 + geom_point(aes(alpha = Species)) + ggtitle("Alpha")+ #색상 진하기 변경
  scale_alpha_discrete(range = c(0.1,0.3))
base1 + geom_point(aes(colour = Species)) + ggtitle("Alpha")+
  scale_colour_hue()
```

ggplot2 – 스케일(3)





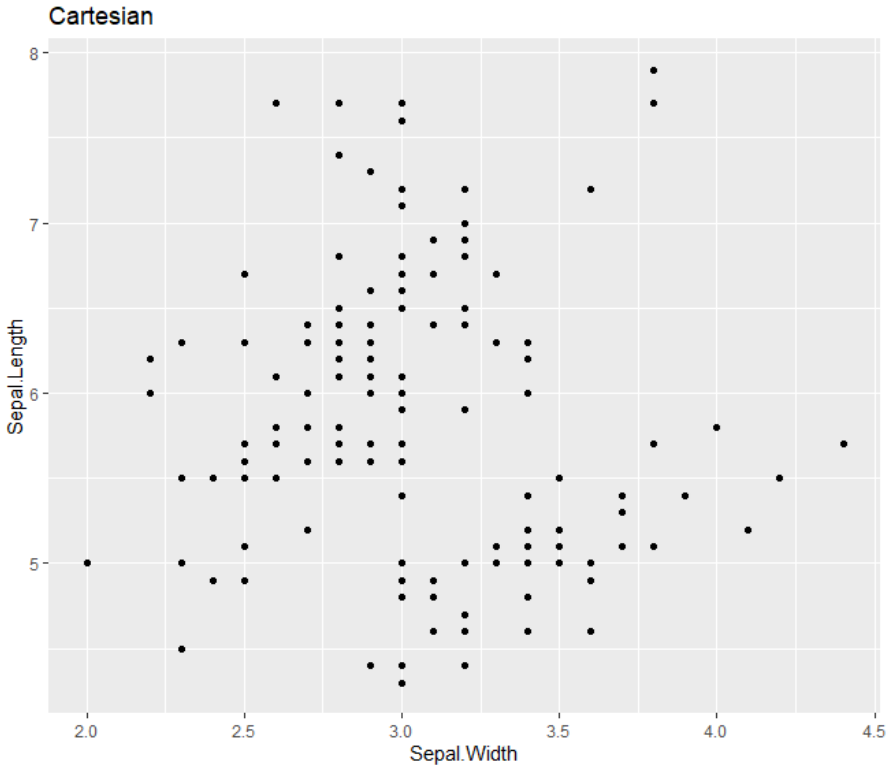


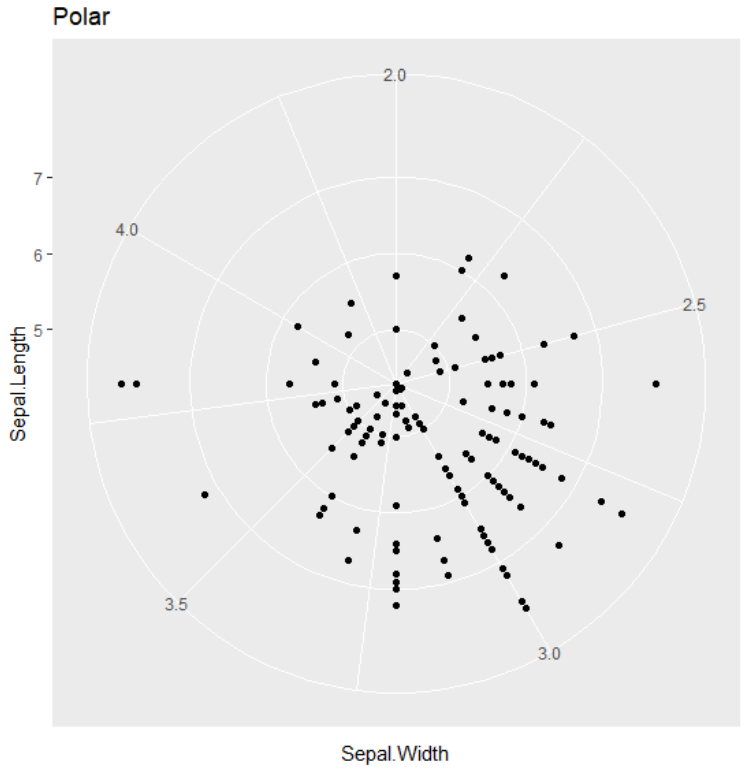


ggplot2 - 좌표계

+ coord_*(…) : 좌표계 설정(cartesian, polar)

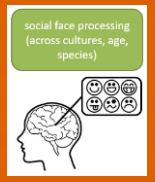
```
## Coordinate
base1 + geom_point() + coord_cartesian() + ggtitle("Cartesian") #직교좌표계
base1 + geom_point() + coord_polar() + ggtitle("Polar")         #극좌표계
```





5. 데이터 전처리

전처리(Preprocessing)



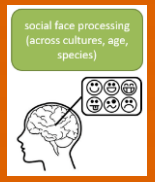
CognitiveSystems

데이터를 분석 가능한 데이터로 만드는 과정

- NA를 제거하거나 다른 값으로 변경
- 파생변수 생성
- 이상치 탐지
- 여러 개의 Dummy변수 생성

전처리는 raw 데이터를 변환하여 모델에 적용할 최종 데이터를 만드는 데에 의의가 있다.

데이터 필터



CognitiveSystems

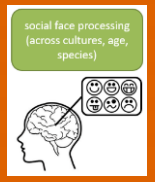
앞서 index를 이용하여 데이터를 필터하는 과정을 익혔다면, 조건문을 통해 해당 조건을 만족하는 데이터를 필터하도록 한다.

df[행 조건, 열 조건] : 행 조건을 만족하는 행과, 열 조건을 만족하는 열을 추출

조건이 True이면 표시, False면 표시되지 않음

```
ind = rep(c(T,F),75)
iris[ind,]                              #홀수 행만 추출
iris[iris$Species == 'setosa',]         #setosa종 iris만 추출
iris[iris$Sepal.Length <= 5.0,]         #Sepal.Length가 5.0보다 작은 iris 추출
iris[iris$Species == 'setosa' & iris$Sepal.Length <= 5.0,]  #위 두 조건을 모두 만족
```

결측치



CognitiveSystems

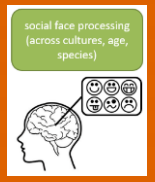
어떤 데이터 프레임에는 NA, NULL과 같이 값이 비어 있거나, 사용할 수 없는 값이 할당되어 있는 경우가 있다. 이들을 결측치라 하며, 이를 확인하기 위해 summary(), is.na(), anyNA() 등으로 데이터 프레임 내의 결측치를 확인해야 한다.

** NULL: 비어있는 값, NA: Not Available, NaN: Not a Number

```
#Cars93 Dataset
library(MASS)
anyNA(Cars93)              #해당 데이터프레임에 NA가 있는지?: TRUE
is.na(Cars93)              #각 값들이 NA인지 확인
sum(is.na(Cars93))         #NA의 개수 파악 : 13

Cars93[is.na(Cars93$Luggage.room),c(1,2,3,24,25)]
```

```
> Cars93[is.na(Cars93$Luggage.room),c(1,2,3,24,25)]
   Manufacturer      Model    Type Luggage.room Weight
16    Chevrolet Lumina_APV     Van           NA   3715
17    Chevrolet      Astro     Van           NA   4025
19    Chevrolet   Corvette Sporty           NA   3380
26        Dodge    Caravan     Van           NA   3705
36         Ford   Aerostar     Van           NA   3735
56        Mazda        MPV     Van           NA   3735
57        Mazda       RX-7 Sporty           NA   2895
66       Nissan      Quest     Van           NA   4100
70   Oldsmobile Silhouette     Van           NA   3715
87       Toyota     Previa     Van           NA   3785
89   Volkswagen    Eurovan     Van           NA   3960
```

결측치 처리



CognitiveSystems

1. na.omit()함수를 이용하여 결측치를 제거한다.

```
na.omit(Cars93)
```

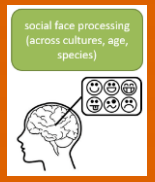
1. 결측치를 다른 값으로 바꾼다.

```
ind_na = which(Cars93$Luggage.room %in% c(NA))    ##NA인 index
Cars93[ind_na,]$Luggage.room = 0                  ##NA 값을 0으로 변환
Cars93[ind_na,c(1,2,3,24,25)]
```

```
> Cars93[is.na(Cars93$Luggage.room),c(1,2,3,24,25)]
   Manufacturer      Model   Type Luggage.room Weight
16    Chevrolet Lumina_APV    Van           NA   3715
17    Chevrolet      Astro    Van           NA   4025
19    Chevrolet   Corvette Sporty           NA   3380
26        Dodge    Caravan    Van           NA   3705
36         Ford   Aerostar    Van           NA   3735
56        Mazda        MPV    Van           NA   3735
57        Mazda       RX-7 Sporty           NA   2895
66       Nissan      Quest    Van           NA   4100
70   Oldsmobile Silhouette    Van           NA   3715
87       Toyota     Previa    Van           NA   3785
89   Volkswagen     Eurovan    Van           NA   3960
```

```
> Cars93[ind_na,c(1,2,3,24,25)]
   Manufacturer      Model   Type Luggage.room Weight
16    Chevrolet Lumina_APV    Van           0   3715
17    Chevrolet      Astro    Van           0   4025
19    Chevrolet   Corvette Sporty           0   3380
26        Dodge    Caravan    Van           0   3705
36         Ford   Aerostar    Van           0   3735
56        Mazda        MPV    Van           0   3735
57        Mazda       RX-7 Sporty           0   2895
66       Nissan      Quest    Van           0   4100
70   Oldsmobile Silhouette    Van           0   3715
87       Toyota     Previa    Van           0   3785
89   Volkswagen     Eurovan    Van           0   3960
```

데이터 필터(2)



CognitiveSystems

어떤 데이터 프레임에는 NA, NULL과 같이 값이 비어 있거나, 사용할 수 없는 값이 할당되어 있는 경우가 있다. 이들을 결측치라 하며, 이를 확인하기 위해 summary(), is.na(), anyNA() 등으로 데이터 프레임 내의 결측치를 확인해야 한다.

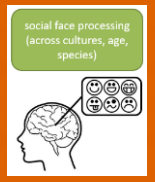
** NULL: 비어있는 값, NA: Not Available, NaN: Not a Number

```
#Cars93 Dataset
library(MASS)
anyNA(Cars93)              #해당 데이터프레임에 NA가 있는지?: TRUE
is.na(Cars93)              #각 값들이 NA인지 확인
sum(is.na(Cars93))         #NA의 개수 파악 : 13

Cars93[is.na(Cars93$Luggage.room),c(1,2,3,24,25)]
```

NA를 다른 값으로 변경하거나, na.omit()함수를 이용하여 결측치를 제거한다.

```
na.omit(Cars93)
```

데이터 생성



CognitiveSystems

데이터 프레임에 새로운 행이나 열을 생성할 수 있다.
이를 통해 분석에 새로운 파생변수들을 생성할 수 있다.

```
> ## Add Column
> new_iris = iris
> new_iris$Lentgh.mean = (iris$Sepal.Length + iris$Petal.Length)/2
> head(new_iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Lentgh.mean
1          5.1         3.5          1.4         0.2  setosa        3.25
2          4.9         3.0          1.4         0.2  setosa        3.15
3          4.7         3.2          1.3         0.2  setosa        3.00
4          4.6         3.1          1.5         0.2  setosa        3.05
5          5.0         3.6          1.4         0.2  setosa        3.20
6          5.4         3.9          1.7         0.4  setosa        3.55

> new_iris['Width.mean'] = (iris$Sepal.Width + iris$Petal.Length)/2
> head(new_iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Lentgh.mean Width.mean
1          5.1         3.5          1.4         0.2  setosa        3.25       2.45
2          4.9         3.0          1.4         0.2  setosa        3.15       2.20
3          4.7         3.2          1.3         0.2  setosa        3.00       2.25
4          4.6         3.1          1.5         0.2  setosa        3.05       2.30
5          5.0         3.6          1.4         0.2  setosa        3.20       2.50
6          5.4         3.9          1.7         0.4  setosa        3.55       2.80
```

데이터 결합

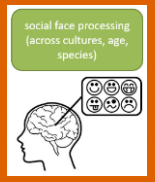
CognitiveSystems

여러 변수를 추가해야 할 경우, 해당 변수를 데이터프레임 형식으로 만들고 cbind, 혹은 rbind를 이용하여 결합한다.

```
new_iris = iris    ##초기화
new_data = data.frame(Sepal.Mean = Sepal.mean,
                      Petal.Mean = Petal.mean) ##Data.Frame화
new_iris = cbind(new_iris, new_data)
head(new_iris)
```

```
> head(new_iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Sepal.Mean Petal.Mean
1          5.1         3.5          1.4         0.2  setosa       4.30        0.80
2          4.9         3.0          1.4         0.2  setosa       3.95        0.80
3          4.7         3.2          1.3         0.2  setosa       3.95        0.75
4          4.6         3.1          1.5         0.2  setosa       3.85        0.85
5          5.0         3.6          1.4         0.2  setosa       4.30        0.80
6          5.4         3.9          1.7         0.4  setosa       4.65        1.05
```

더미변수 생성



CognitiveSystems

Categorical 변수를 dummy로 만드는 factor가 있으나, factor 내의 level이 3개 이상일 시에는 데이터 분석에 적합하지 않다.

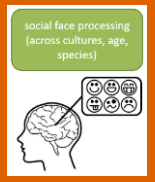
이를 여러 개의 dummy로 만드는 과정이다.

```
## Dummy Columns
install.packages("fastDummies")
library(fastDummies)
dummy_cols(iris, select_columns = 'Species')
```

또한, Continuous 데이터(실수형 데이터)를 필요에 따라 Categorical(범주형) 데이터로 변환할 수 있다.

```
# Dummy from continuous
bin = seq(min(iris$Sepal.Length)-0.1,max(iris$Sepal.Length)+0.1, length.out = 4)
iris$interval = cut(iris$Sepal.Length, bin)
dummy_cols(iris, select_columns = 'interval')
```

Dplyr 패키지



CognitiveSystems

Dplyr 패키지를 이용하여 위의 전처리를 쉽게 할 수 있다.

```
## dplyr package
install.packages("dplyr")
library(dplyr)
# Filter
iris %>% filter(Species == 'setosa')
# Select
iris %>% select(Sepal.Length, Petal.Length, Species)
# New column
iris %>% mutate(Mean.Length = (Sepal.Length + Petal.Length)/2)
# Summarize
iris %>% summarize(Sepal = Sepal.Length+Sepal.Width)
# Group by
iris %>% group_by(Species) %>% summarize(Mean = mean(Sepal.Length))
```

Q&A