



# 비즈니스프로그래밍

- Python과 R 기초 -

# ○○○ 1. R 데이터 분석 환경 만들기 ○○○

- ▶ R과 R스튜디오 설치하기
- ▶ R스튜디오와 친숙해지기
- ▶ 프로젝트 만들기
- ▶ 유용한 환경 설정



# R스튜디오와 친숙해지기



- ▶ 명령어를 실행하는 콘솔 창
- ▶ 명령어를 기록하는 소스 창
- ▶ 생성한 데이터를 보여 주는 환경 창
- ▶ 폴더에 있는 파일을 보여주는 파일 창



# 프로젝트 만들기



- ▶ 여러 가지 작업들을 프로젝트 폴더별로 관리하면 편리
- ▶ 오른쪽위 육각형 모양 버튼
- ▶ 프로젝트 활용 – 스크립트 관리하기
  - ▶ 저장하기
  - ▶ 새 스크립트 만들기
- ▶ 워킹 디렉터리



# 유용한 환경 설정



- ▶ 글로벌 옵션 vs. 프로젝트 옵션
- ▶ Sort-wrap 자동 줄바꿈 옵션 설정하기
- ▶ 인코딩 방식 설정하기



## 2. 데이터의 구조



- ▶ 2.1 데이터의 종류와 변수
  - ▶ 변수
  - ▶ 데이터 연산
  - ▶ 데이터 타입
  - ▶ 데이터 형태



# 변수



- ▶ 변수 variable : 데이터를 저장하는 공간
  - ▶ (변수 이름) <- (값)
- ▶ 변수명은 문자로 시작해야 한다!
  - ▶ 문자, 숫자, 언더바(\_) 조합
  - ▶ 대소문자 구분 주의!
- ▶ 참고: 주석(메모, 노트, 설명 넣을 때 활용)은 샵(#) 기호
- ▶ 문자 데이터 저장할 때는 큰따옴표(" ")나 작은따옴표(' ') 이용



# 데이터 연산



산술연산자	기능
+	더하기
-	빼기
*	곱하기
/	몫
%%	몫(정수)
%%	나머지
^	제곱





# 데이터 연산



비교연산자	기능
>	크다
>=	크거나 같다
<	작다
<=	작거나 같다
==	같다
!=	같지 않다

- ▶ 두 데이터의 크기를 비교할 때 사용
- ▶ 결과는 조건을 만족할 때는 TRUE, 만족하지 않을 때는 FALSE를 반환



# 데이터 연산



논리연산자	기능
&	그리고 AND
	또는 OR

- ▶ 조건이 두 개 이상일 때 사용
- ▶ & 연산자는 두 조건을 둘 다 만족하는 경우에 TRUE 반환
- ▶ | 연산자는 두 조건 중 하나만 만족해도 TRUE 반환



# 데이터 타입



데이터 타입	의미	값
numeric	실수형	10, 15.2
character	문자형	"apple", "banana"
logical	논리형	TRUE, FALSE, T, F
factor	범주형	1, 2, 3

- ▶ 변수는 데이터값에 따라 데이터 타입이 달라진다.
- ▶ 정수, 실수와 같이 숫자로 이루어진 데이터 → 실수형 타입
- ▶ 따옴표로 묶어서 표현한 문자 데이터 → 문자형 타입
- ▶ TRUE, FALSE로 이루어진 데이터 → 논리형 타입
  - ▶ 값이 참인지 거짓인지 나타낼 때 사용
  - ▶ 줄여서 T, F로도 표기
  - ▶ 따옴표를 쓰지 않으며 모두 대문자로 작성
- ▶ 범주형 타입은 분류를 의미한다.
  - ▶ 1, 2, 3이라고 한다면 숫자를 의미하는 것이 아니라 각 분류번호를 나타낸다.



# 데이터 타입



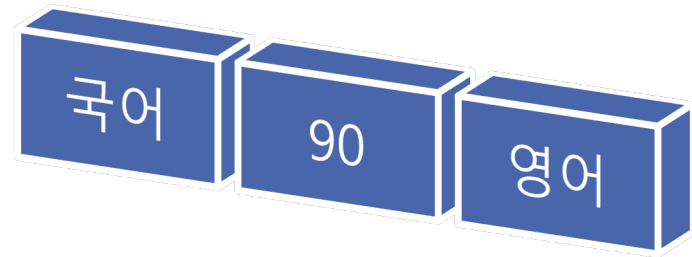
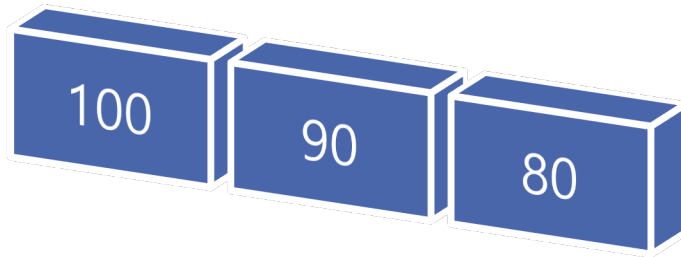
c( )	데이터들을 하나로 결합하는 함수
class( )	데이터 타입을 확인하는 함수
factor( )	범주형 변수로 만드는 함수



# 데이터 형태



- ▶ 데이터는 형태에 따라 단일형과 다중형으로 나뉜다.
- ▶ 단일형은 한 가지 데이터 타입으로 이루어진 형태
- ▶ 다중형은 다양한 데이터 타입으로 구성



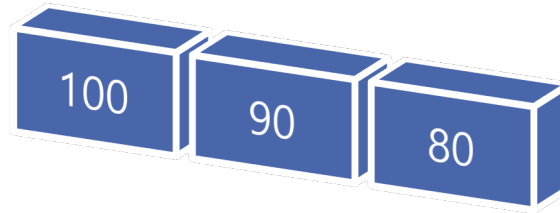
- ▶ 왼쪽은 숫자 데이터만 담아서 있으므로 단일형
- ▶ 오른쪽은 숫자와 문자 데이터가 담아서 있으므로 다중형



# 벡터



- ▶ 벡터(Vector)는 한 개나 여러 개의 값으로 구성된 데이터 구조
- ▶ 단일형 데이터 형태의 값들을 1차원으로 저장할 수 있는 공간



- ▶ `c()` 함수는 Combine(결합하다)라는 뜻 ➔ 벡터 데이터를 만들 수 있다.
  - ▶ 여러 개의 숫자를 쉼표(,)로 구분하여 `c()` 함수에 입력하면 숫자 벡터가 된다.
  - ▶ 숫자와 문자를 `c()` 함수를 이용하여 저장하고 출력하면 모두 문자로 저장된다.
  - ▶ ➔ 왜냐하면 벡터 데이터 구조는 단일형만 저장할 수 있으므로 숫자를 같이 저장해도 문자 데이터로 자동 변환되어 한가지 데이터 타입으로만 저장
  - ▶ 콜론(:)을 이용해 시작 숫자와 마지막 숫자를 지정하면 마지막 숫자까지 1씩 증가되어 저장



# 벡터



- ▶ `seq( )` 함수는 연속적인(Sequence)의 뜻 → 연속된 값이나 일정한 간격의 숫자를 저장할 때

<code>seq(from, to, by)</code>
from: 시작 값
to: 끝 값
by: 증감값

- ▶ `by` 옵션값을 지정하지 않으면 기본값은 1
  - ▶ 단, `by` 옵션값에 음수값을 쓰는 경우는 시작 값이 끝값보다 반드시 커야 한다.



# 벡터



- ▶ `rep()` 함수는 반복하다(Repeat)의 뜻 → 특정한 값을 반복할 때

`rep(x, times | each)`

x: 반복할 자료

times: x 반복 횟수(기본값)

each: x의 원소 각각의 반복 횟수

- ▶ 반복 횟수 옵션명을 생략하면 기본값은 times로 설정된다.





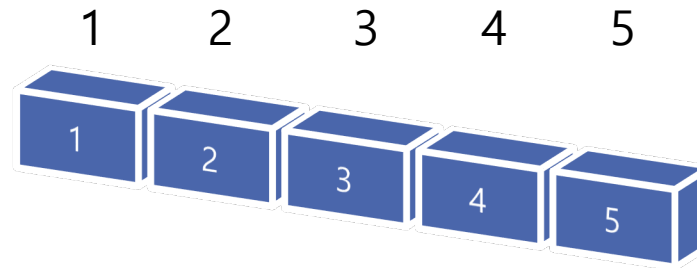
# 벡터



## ▶ 벡터의 각 원소에 접근하는 방법

벡터명[선택하고자하는원소번호]	벡터명[시작원소번호:끝원소번호]
해당 원소의 값을 추출	여러 개의 원소들 추출

원소번호

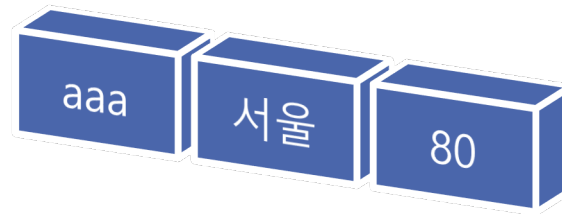




# 리스트



- ▶ 리스트(List)는 모든 데이터 구조를 포함
- ▶ 다중형 데이터를 다차원으로 저장할 수 있는 공간



- ▶ 리스트와 벡터의 차이점은 벡터는 한 가지 데이터 타입만 저장하는 단일형, 리스트는 여러 데이터 타입을 포함하는 다중형
- ▶ 리스트 구조는 `list( )` 함수를 이용하여 생성
- ▶ 참고: `str( )`은 변수의 속성을 자세히 알려주는 함수



# 리스트



## ▶ 리스트의 각 원소에 접근하는 방법

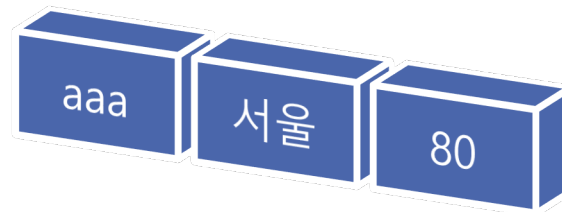
리스트명[원소번호] 리스트명["원소명"]	리스트명[["원소명"]] 리스트명\$원소명
원소명과 원소값 출력	원소값만 출력

원소번호

1

2

3



원소명

이름

지역

점수



# 데이터 프레임



- ▶ 데이터 프레임(Data Frame)은 R에서 많이 사용되는 데이터 구조
- ▶ 표와 같이 행과 열로 구성
- ▶ 숫자형, 문자형 등 다중형 형태로 데이터 저장 가능

4개 열

이름	학과	학년	학점
aaa	경영학과	4	4.2
bbb	물리학과	3	3.7
ccc	영문학과	2	3.5

3개 행



# 데이터 프레임



## ▶ 데이터 프레임에서 원소를 추출하는 방법

데이터프레임명[열번호]	지정된 열의 모든 값 출력
데이터프레임명[행번호,]	지정된 행의 모든 값 출력
데이터프레임명[행번호,열번호]	지정된 행과 열이 교차되는 지점의 값 출력
데이터프레임명\$변수명	지정하면 변수명의 열 값이 출력
데이터프레임명[행번호,열번호] <- 값	해당위치의 원소값 변경



## 2. 데이터의 구조



### ▶ 2.2 파일 입출력

- ▶ 파일로 저장하기
- ▶ 외부 파일 가져오기



# 파일로 저장하기



- ▶ 데이터 프레임으로 작성된 데이터는 파일로 저장할 수 있다.
- ▶ csv파일로 저장하기 위해서는 `write.csv( )` 함수를 이용
- ▶ `write.csv( )` 함수에 저장할 데이터 프레임명과 저장할 위치 및 파일명 전달



# 외부 파일 가져오기



- ▶ 데이터분석을 위해 외부에 있는 데이터를 불러와 사용하는 경우가 많다.
- ▶ xlsx파일은 엑셀로 만들어진 파일, csv파일은 쉼표로 구분된 데이터 파일
- ▶ csv파일을 불러오기 위해 `read.csv()` 함수 이용
- ▶ 파일을 불러올 때는 파일의 경로, 파일명, 파일확장자가 오타없이 정확하게 작성되었는지 확인한다.
- ▶ 엑셀파일은 `readxl` 패키지 설치 필요 ➔ 자세한 설명은 패키지 소개에서
- ▶ `readxl` 패키지에서 제공하는 `read_excel()` 함수 이용