전산통계학 실습

R 설치 및 실행

R 프로그래밍

- R 언어는 통계 및 그래프 작업을 위한 프로그래밍 언어
  - R은 언어이면서 동시에 다양한 패키지를 통한 Case Study가 가능
- 수많은 통계 관련 패키지가 이미 개발되어 저장
- 의학, 수학 등 통계가 필요한 다양한 학문에서 사용되는 중

R 설치

- Windows
  - 아래 링크 중 하나로 접속하여 운영체제에 맞도록 다운로드 한다.
    - Official: http://www.r-project.org
    - CRAN: http://cran.nexr.com
    - 위 사이트에서 튜토리얼, 함수의 설명, 추가 패키지 등을 볼 수 있다.
  - 다운로드 받은 툴을 실행시켜 프로그래밍을 시작한다.

- Mac
  - Mac에는 R 프로그래밍이 기본으로 설치되어 있다.
  - 터미널에서 'R' 키워드를 입력하여 프로그래밍을 시작한다.

- 통합 개발 환경 IDE
  - Rstudio: http://www.rstudio.com

R 스크립트 작성

- Windows
  - 위 메뉴에서 파일 > 새 스크립트
  - R 편집기에서 코드 작성 (작성한 코드는 확장자 (.R)로 저장됨)
  - 작성된 코드를 모두 선택(Ctrl+A)하여 F5로 실행
  - R 터미널 내부에서 코드 실행됨

- Mac
  - Rstudio / Vi 에디터 / 다른 에디터 툴 등을 이용하여 코드 작성
  - 작성된 코드를 확장자 (.R)로 저장
  - R 터미널 내부에서 > source("파일경로") 로 실행

R 패키지

```
> # 미리 설치된 패키지(불러오기)
> library(MASS)
>
> # 새로 설치가 필요한 패키지(패키지 설치)
> install.packages("ggplot2")
> library(ggplot2)
```

- R 패키지를 통해 다양한 데이터를 불러오거나, 구현된 함수를 불러올 수 있다.
  - 설치 시 자동으로 다운로드 되는 패키지와 직접 다운로드 해야 하는 패키지가 있다.
- 패키지 설치
  - > install.packages(...)
  - 패키지 직접 다운로드
- 패키지 사용
  - > library(...)
  - 설치된 패키지 사용

R 패키지

```
> library(MASS)
> str(Cars93)
'data.frame':   93 obs. of  27 variables:
 $ Manufacturer      : Factor w/ 32 levels "Acura","Audi",..: 1 1 2 2 3 4 4 4 4 5 ...
 $ Model             : Factor w/ 93 levels "100","190E","240",..: 49 56 9 1 6 24 54 74 73 35 ...
 $ Type              : Factor w/ 6 levels "Compact","Large",..: 4 3 1 3 3 3 2 2 3 2 ...
 $ Min.Price         : num  12.9 29.2 25.9 30.8 23.7 14.2 19.9 22.6 26.3 33 ...
 $ Price             : num  15.9 33.9 29.1 37.7 30 15.7 20.8 23.7 26.3 34.7 ...
 $ Max.Price         : num  18.8 38.7 32.3 44.6 36.2 17.3 21.7 24.9 26.3 36.3 ...
 $ MPG.city          : int  25 18 20 19 22 22 19 16 19 16 ...
 $ MPG.highway       : int  31 25 26 26 30 31 28 25 27 25 ...
 $ AirBags           : Factor w/ 3 levels "Driver & Passenger",..: 3 1 2 1 2 2 2 2 2 2 ...
 $ DriveTrain        : Factor w/ 3 levels "4WD","Front",..: 2 2 2 2 3 2 2 3 2 2 ...
 $ Cylinders         : Factor w/ 6 levels "3","4","5","6",..: 2 4 4 4 2 2 4 4 4 5 ...
 $ EngineSize        : num  1.8 3.2 2.8 2.8 3.5 2.2 3.8 5.7 3.8 4.9 ...
 $ Horsepower        : int  140 200 172 172 208 110 170 180 170 200 ...
 $ RPM               : int  6300 5500 5500 5500 5700 5200 4800 4000 4800 4100 ...
 $ Rev.per.mile      : int  2890 2335 2280 2535 2545 2565 1570 1320 1690 1510 ...
 $ Man.trans.avail   : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 1 1 1 1 1 ...
 $ Fuel.tank.capacity: num  13.2 18 16.9 21.1 21.1 16.4 18 23 18.8 18 ...
 $ Passengers        : int  5 5 5 6 4 6 6 6 5 6 ...
 $ Length            : int  177 195 180 193 186 189 200 216 198 206 ...
 $ Wheelbase         : int  102 115 102 106 109 105 111 116 108 114 ...
 $ Width             : int  68 71 67 70 69 69 74 78 73 73 ...
 $ Turn.circle       : int  37 38 37 37 39 41 42 45 41 43 ...
 $ Rear.seat.room    : num  26.5 30 28 31 27 28 30.5 30.5 26.5 35 ...
 $ Luggage.room      : int  11 15 14 17 13 16 17 21 14 18 ...
 $ Weight            : int  2705 3560 3375 3405 3640 2880 3470 4105 3495 3620 ...
 $ Origin            : Factor w/ 2 levels "USA","non-USA": 2 2 2 2 2 1 1 1 1 1 ...
 $ Make              : Factor w/ 93 levels "Acura Integra",..: 1 2 4 3 5 6 7 9 8 10 ...
> 
```

- 예시) MASS 패키지에 저장된 Cars93 데이터셋 확인
  - MASS 패키지는 이미 설치되어 있으므로 불러오기 가능
  - > str(...) 함수를 사용하여 해당 데이터셋의 요약을 볼 수 있다.
    - 데이터셋 내부 컬럼, 컬럼의 자료형 등

데이터 다루기

- 외부에서 데이터 불러오기
  - > read.csv(“파일경로”)
  - R에서는 CSV, EXCEL, SPSS, SAS 등으로 저장된 외부 데이터들을 불러와 R에서 다루는 데이터 형태로 저장할 수 있다.
  - 테이블 형태의 데이터(행과 열이 존재하는 데이터) 모양으로 저장

- 외부로 데이터 내보내기
  - > write.csv(저장할 데이터이름, “저장경로”)
  - R 내부에서 다루고 있는 데이터를 위에서 서술한 다양한 형태(확장자)의 파일로 저장한다.
  - 또는 R 프로그램에서 사용하는 데이터를 작업공간에 둘 수도 있다.

데이터 다루기

```
> data <- read.csv("C:\\Users\\Teasung\\Desktop\\test.csv")
> data
  x y
1 1 1
2 2 2
3 3 3
4 4 5
5 5 8
> data <- 5:30
> data
 [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
> write.csv(data, "C:\\Users\\Teasung\\Desktop\\data.csv")
>
```

- 예시1) 위 경로에 저장된 데이터를 R 내부로 불러오기
  - 관측된 데이터 5개가 2개의 속성을 나타낸 테이블 형태의 데이터

- 예시2) R 내부에서 다루던 데이터를 외부로 저장하기
  - > num1:num2 는 num1 ~ num2 범위의 숫자 sequence 저장
  - 저장된 데이터를 외부 경로의 데이터로 저장

R의 자료형

- 변수 생성 및 접근
  - R에서는 각 라인 뒤에 세미콜론(;)을 붙이지 않아도 된다.
    - 한 줄에 여러 실행문을 사용하는 경우에는 세미콜론으로 구분 가능
  - 변수에 데이터를 저장(생성)하는 과정에서 자동 선언되므로, 따로 변수를 선언하지 않아도 된다.
    - 변수에 데이터를 저장할 때는 '=' 대신 '<-'를 사용한다.
    - 일시적 동일함을 나타내는 것과 할당하는 것의 구분을 위하여 '='는 함수 매개변수 입력 등에서 사용하고, 실질적인 값의 할당은 '<-'를 이용한다.
  - 다음과 같은 변수명들은 허용되지 않는다.
    - 숫자로 시작하는 변수 이름, ex) 2v
    - .(점), 숫자가 이어지는 변수 이름, ex) .2v
    - 하이픈이 포함된 변수 이름, ex) v-v

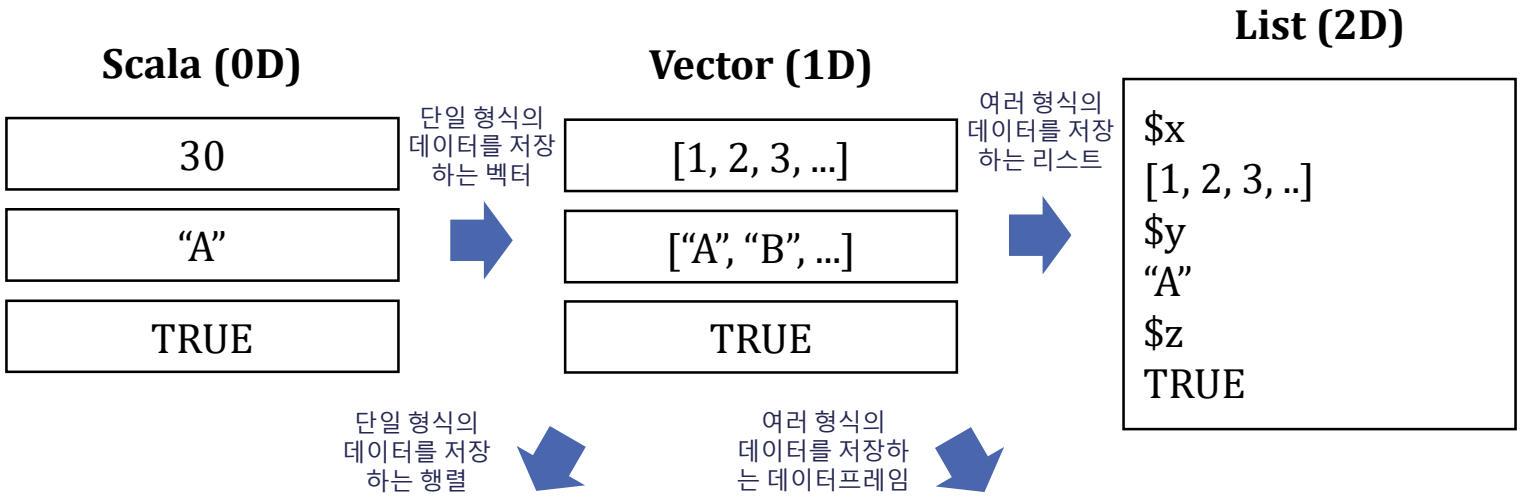
R의 자료형

```
> x1 <- 30
> y1 <- 1:100
> x1
[1] 30
> y1
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  $
 [42]  42  43  44  45  46  47  48  49  50  51  52  53  54  55  $
 [83]  83  84  85  86  87  88  89  90  91  92  93  94  95  96  $
> x2
에러: 객체 'x2'를 찾을 수 없습니다
> sum(1, x2=2)
[1] 3
> x2
에러: 객체 'x2'를 찾을 수 없습니다
> sum(1, x2<-2)
[1] 3
> x2
[1] 2
```

- 예시) x1, y1에 각각 값을 할당하고 출력
  - 출력에서 [1], [42], [83] 등의 표시는 현재 배열(데이터의 행)에서 가장 앞에 있는 값의 전체 index를 표현한다. 즉, 가장 앞에 있는 값이 전체 데이터 배열 내에서 1, 42, 83번째 값이라는 뜻을 나타낸다.
  - '=' 를 이용하면 일시적인 할당을 수행하고 실제 값의 변화를 수행하지 않는다. 반면, '<-'는 실질적으로 할당 또한 포함하고 있다.

R의 자료형

- 다른 프로그래밍 언어들과 같이 '값'과 '여러 값'들을 저장할 수 있는 자료구조 형태를 가진다.



Matrix (2D)

	1	2	3
1	10	20	30
2	40	50	60
3	70	80	90

DataFrame (2D)

	x	y	z
1	"A"	1	TRUE
2	"B"	2	TRUE
3	"C"	3	FALSE

R의 자료형

- Scala
  - 자료형의 기본 단위로써, 단일 자료형을 가지는 하나의 변수
  - R에서는 단일 수치 값으로 존재하지 않고, 사실 한 개의 원소를 가지는 벡터로 표현된다.

```
> x1 <- 30
```

  - 위 코드의 'x1'은 사실 하나의 원소 Scala가 아니라, '30' 하나 만을 가진 벡터이다. 즉, x1 == x1[1] && x1[1] == 30 이다.

  - Scala의 자료형 타입들
    - Numeric: 숫자 자료형
    - Character: 문자열 자료형
    - Bool: 미리 선언된 TRUE, FALSE 값 (T, F로도 접근 및 사용 가능)
    - Factor: 범주 데이터 자료형 (데이터셋 내부 값들의 목록 = Levels)
    - NULL: 실제로 존재하지 않는 Undefined 값
    - NA: 관측 내에서 결측된 값 (통계에서 '취합되지 않는 값'을 표현할 때 사용)

R의 자료형

```
> library(MASS)
> class(Cars93)
[1] "data.frame"
> cars93_manuf <- Cars93$Manufacturer
> nlevels(cars93_manuf)
[1] 32
> levels(cars93_manuf)
 [1] "Acura"         "Audi"          "BMW"           "Buick"         "Cadillac"      "Chevrolet"     "Chrylser"      "Chrysler"
[11] "Ford"          "Geo"           "Honda"         "Hyundai"       "Infiniti"      "Lexus"         "Lincoln"       "Mazda"
[21] "Mitsubishi"    "Nissan"        "Oldsmobile"    "Plymouth"      "Pontiac"       "Saab"          "Saturn"        "Subaru"
[31] "Volkswagen"    "Volvo"
> levels(cars93_manuf)[1]
[1] "Acura"
>
```

- Factor
  - 예시) MASS패키지 내부 Cars93 데이터셋 중 Manufacturer 변수컬럼
    - DataFrame 자료형은 '$'를 이용하여 변수명으로 자료에 접근할 수 있다.
    - 주어진 데이터 목록(범주) 내의 데이터가 저장되어 있다.
  - > levels(...): 변수의 범주 출력
  - > nlevels(...): 변수의 범주 개수 출력