

<1. R언어의 데이터 형에 대해서 조사를 하고, 간단한 예를 보이시오.>

동질적 객체: 모두 같은 타입의 성분들로 구성된 데이터 형태를 뜻한다.

벡터(vector)

*벡터의 정의

수학적으로 벡터는 크기와 동시에 방향을 갖는 물리량을 뜻한다.

R 언어에서의 벡터는 동질적 객체로 동일한 유형의 데이터 요소의 순서라고 할 수 있다.

동일한 데이터들 벡터 안의 각각의 값은 구성 요소라고 부른다.

스칼라는 한 개의 값으로 이루어진 변수인 반면에 벡터에서는 여러 개의 값을 담을 수 있는 것이 특징이다.

또 C 언어나 파이썬과 달리 인덱스가 0이 아닌 1부터 시작한다는 것도 특징이라고 할 수 있다.

벡터는 `c(3,4,5,6,7,8)` 이런 식으로 `c()` 함수 안에 요소들을 담을 수 있다.

`a <- c(3,4,5,6,7,8)` a라는 변수 안에 벡터를 지정해줄 수도 있다.

이러한 벡터들은 응용을 한다면 벡터 연산 등으로 확장할 수 있다.

예를 들면

```
> x <- c(1, 2, 3, 4, 5)
```

```
> x + 1
```

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

과 같이 벡터 연산이 가능하다.

행렬(matrix)

*행렬의 정의

- 동일한 스칼라 데이터 타입의 2차원 데이터 구조를 말한다.
- 행렬은 색인 또는 행과 열의 이름을 통해 접근할 수 있다.
- 색인 형태로 음수를 사용해 특정 행이나 열을 제외하거나, 색인에 벡터를 지정해 여러 값을 한 번에 가져올 수 있다.
- `matrix(벡터 값, nrow=행의 수, ncol=열의 수, byrow=T 면 행우선 F면 열 우선으로 데이터를 채움, dimnames=행렬의 각 차원의 이름 저장)`
- `dimnames(행렬)` : 행렬의 각 차원명
- `rownames(행렬)` : 행렬의 행 이름
- `colnames(행렬)` : 행렬의 열 이름

- nrow(행렬) : 행렬의 행의 수
- ncol(행렬) : 행렬의 열의 수
- is.matrix(객체) : 참 거짓 확인

*행렬의 연산

- $A + x$: 행렬 A의 모든 값에 스칼라 x를 더한다. 이외에도 -, *, / 연산자를 사용할 수 있다.
- $A + B$: 행렬 합을 구한다. 행렬 차는 - 연산자를 사용한다.
- $A \%*\% B$: 행렬 곱을 구한다.
- t(행렬이나 데이터프레임) : 전치행렬(모든 열을 대응하는 행으로 바꿈)을 구한다.
- solve(행렬1, 행렬2) : 행렬1 $\%*\% x =$ 행렬2에서 x를 구한다. 행렬2를 지정하지 않으면 행렬1의 역행렬을 구한다.

배열(array)

*배열의 정의

배열은 동질적(homogeneous) 객체에 속합니다.

array는 기본적으로 배열이라는 뜻을 가지고 있고

array가 R에서 함수로의 의미는 동일 데이터 타입을 갖는 다차원 배열입니다.

벡터가 1차원 데이터고 행렬이 2차원 데이터라면 배열은 N 차원 데이터라고 볼 수 있습니다.

ex) 3차원 배열

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

```
dimnames(abc)<-list(c("a1","a2","a3"), c("b1","b2"), c("c1","c2","c3","c4"))
```

dim : 데이터 객체의 차원

dimnames : 차원 이름

이질적 객체: 다른 타입의 성분들로 구성된 경우를 뜻한다.

리스트(list)

*리스트의 정의

벡터처럼 여러 원소를 담을 수 있는 그릇입니다.

*벡터와의 차이점

벡터와 가장 큰 차이점은 여러 가지 자료형 원소도 함께 담을 수 있다는 점이며, 리스트는 배열이 가지고 있는 인덱스라는 장점을 버리고 대신 빈틈없는 데이터의 적재라는 장점을 취한 자료구조입니다.

리스트에서 인덱스는 몇 번째 데이터인가 정도의 의미를 가진다.

*일반적으로 리스트는 아래와 같은 기능(operation)을 가지고 있습니다.

1. 처음, 끝, 중간에 원소를 추가/삭제하는 기능
2. 리스트에 데이터가 있는지를 체크하는 기능
3. 리스트의 모든 데이터에 접근할 수 있는 기능

위와 같은 기능을 가지고 있고, 순서가 있으면서 중복이 허용된다면 그러한 자료구조를 리스트라고 합니다.

빈 원소는 허용하지 않는다.

*만드는 방법은 다음과 같습니다.

```
lst <- list(3.14, 'hello', TRUE)
```

더 할 수 있는 설정은

1. 이름 설정 가능
2. 각 원소에 이름 설정 가능
3. 원소 선택 가능
4. <-를 사용해 빈 list에 원소 추가 가능
5. 원소에 NULL 값을 할당해 list에서 원소 제거 가능
6. mean()을 사용해 원소의 평균 계산 가능

데이터 프레임(data.frame)

*데이터 프레임의 정의

R의 핵심적인 자료구조로서, 엑셀과 같이 숫자, 문자 등 다양한 데이터를 하나의 테이블에 담을 수 있는 자료구조입니다.

저희가 수업 시간에 사용하는 Cars93, mtcars, iris 그리고 최근 배운 lavaan 또한 데이터 프레임에 속합니다.

*어떠한 데이터의 구조를 알고 싶으면

str(데이터 프레임의 이름)을 입력하면 확인할 수 있습니다.

*만드는 방법은 다음과 같습니다.

```
example <- data.frame(animal = c("dog","cat"), age = c(4,5), weight=c(12,8))
```

이와 같이 데이터 프레임을 만들 수 있습니다.

<2. 구조방정식 예시와 분석>

#패키지 설치

```
install.packages("lavaan")
```

```
install.packages("semPlot")
```

#설치한 라이브러리 사용하기 위해 선언

```
library(lavaan)
```

```
library(semPlot)
```

#위 CSV를 사용하기 위해 경로 설정 및 변수 test에 Benz.csv 넣어둠

```
setwd("Benz.csv가 저장되어있는 폴더")
```

```
test<-read.csv("Benz.csv")
```

#CSV 구조 확인 / 구조를 확인하면 다양한 값이 존재한다.

```
str(test)
```

#이제 구조 방정식을 이용하기 위해서 상관관계를 설정한다.

```
test_model <- 'img =~ func+design
```

```
brand = ~famous+german
```

```
rePurchase = ~rePurchase1 + rePurchase2
```

```
sat = ~sat1+ sat2
```

```
sat ~ img + brand
```

```
rePurchase ~ sat'
```

위의 식은 모델을 만드는 방식이다. 위 출처를 변형한 방식으로

외부에서 보이는 이미지와 브랜드가 반자동변속기에 어떠한 영향을 끼치는지,

끼치는 영향이 재 구매에 어떠한 영향을 끼치는지 확인하는 모델이다.

이미지는 기능과 디자인 / 브랜드는 유명도와 독일산(?) 인지 / 재 구매는 데이터 1과 2를

/반자동변속기는 그에 관한 데이터 1과 2를 사용한다.

#이제 그림을 그리기 전에 이 데이터가 유의미 한지 확인한다.

```
test_fit<-sem(test_model,data=test)
```

```
summary(test_fit, standardized=TRUE, fit.measures=TRUE)
```

두 가지 분석을 통해서 p-value와 기타 데이터의 값을 확인할 수 있다.

데이터의 분석을 하였다면 이를 이용하여 그림을 그리면 된다.

```
test_diagram<-semPlot::semPaths(test_fit,
```

```
whatLabels="std",intercepts=FALSE,style="lisrel",
```

```
nCharNodes=0,
```

```
nCharEdges=0,
```

```
curveAdjacent=TRUE,title=TRUE,layout="tree2",curvePivot=TRUE)
```

위와 같이 사용하면 결과를 얻을 수 있다.

구조 방정식 사용하기 (스크립트 사용하기)

```
install.packages("lavaan")
```

```
install.packages("semPlot")
```

```
setwd("Benz.csv가 저장되어있는 폴더")
```

```
test<-read.csv("Benz.csv")
```

```
test_model <- 'img =~ func+design
```

```
brand = ~famous+german
```

```
rePurchase = ~rePurchase1 + rePurchase2
```

```
sat = ~sat1+ sat2
```

```
sat ~ img + brand
```

```
rePurchase ~ sat'
```

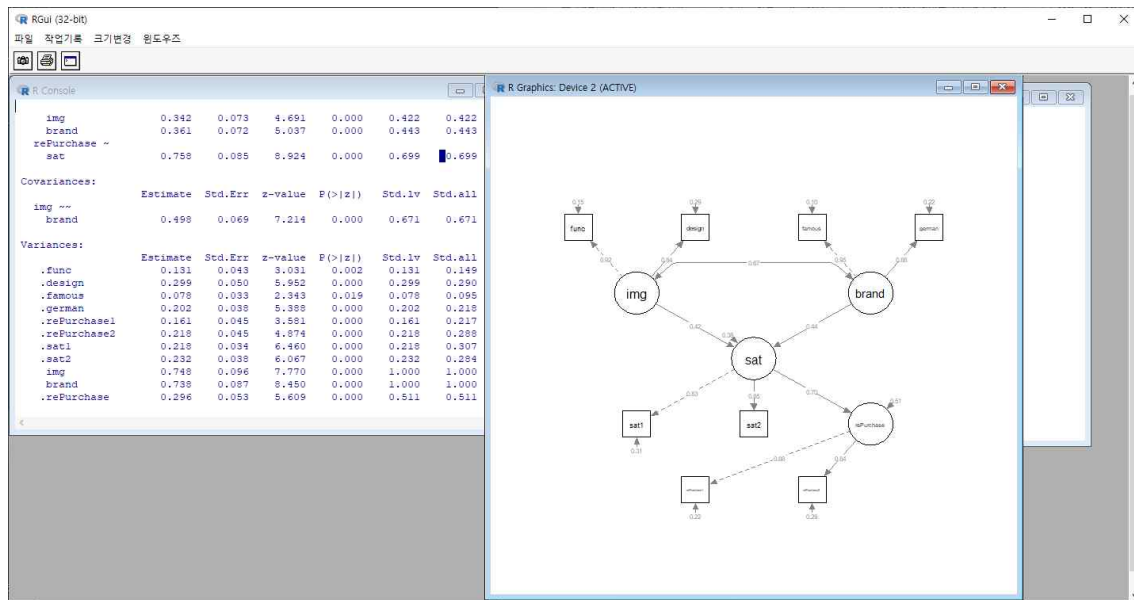
```
test_fit<-sem(test_model,data=test)
```

```
summary(test_fit, standardized=TRUE, fit.measures=TRUE)
```

```
test_diagram<-semPlot::semPaths(test_fit,
```

```
whatLabels="std",intercepts=FALSE,style="lisrel",
```

```
nCharNodes=0,
nCharEdges=0,
curveAdjacent=TRUE,title=TRUE,layout="tree2",curvePivot=TRUE)
```



<소감>

우리 조원들은 이번 중간고사를 치고 난 뒤 각자 개인의 부족한 점을 깨닫게 되었고 이번 고난도 과제를 열심히 하기 위해 수업을 복습하고 개념들을 학습하면서 다시 한 번 복습의 필요성을 깨닫게 되었습니다.

또한 평소에는 연이 닿지 못한 조원들과 함께 조별 과제를 해결하고 공부하면서 혼자보다는 여럿이 함께 학습하는 것이 학습과 정보의 양과 질의 측면에서 큰 차이를 보인다는 것 또한 느낄 수 있었습니다.

또한 R 언어에 사용되는 데이터형과 함수들을 학습하면서 알게 알고 있던 지식들이 세분화되고 구체화되어

R 언어가 다양한 통계에 사용된다는 것을 알게 된 것 등

각각의 용도와 필요성의 지식을 확장할 수 있는 좋은 계기가 되었던 것 같습니다.

또한 R 언어의 데이터형과 함수들의 사용 방법이 간단하고 적시적으로 인터넷에 표현되어 있어서

R 언어는 누구나 쉽고 편하게 사용 가능하다는 것을 다시 한 번 느끼게 되었습니다.

그리고 팀원 모두가 자신의 역할을 충실히 수행하여 정말 팀 프로젝트처럼 과제를 진행한 점에서

서로에게 고맙고 서로가 뿌듯함을 느낄 수 있었던 좋은 계기였던 것 같습니다.

만약 이 과제가 개인 과제였다면 구조 방정식의 예시를 찾고 분석하는 것에서 큰 어려움을 느꼈을 것이라고 생각합니다. 하지만 팀원 모두가 머리를 맞대고 공부하면서 구조방정식에 대한 두려움과 어려움이 타파되는 좋은 계기였던 것 같습니다.

이번 팀 프로젝트를 열심히 수행한 것을 계기로 앞으로의 R 언어 수업과 공부에 자신감이 붙게 되었고

더 열심히 학습해야겠다는 생각이 정말 많이 들었고 복습을 습관화하자라는 큰 뜻을 얻을 수 있었습니다.

어려웠지만 큰 깨달음을 얻게 해주신 심재창 교수님께 정말 감사하다고 느꼈습니다.

앞으로 우리 조원들은 심재창 교수님의 열정에 보답할 수 있게

R 언어에 더 관심을 갖고 학습할 수 있는 자세를 가질 수 있도록 노력하겠습니다.

<출처>

벡터 출처:

<https://sjh836.tistory.com/114>

<https://dsmoon.tistory.com/entry/%EB%B2%A1%ED%84%B0>

<https://jeongchul.tistory.com/479>

행렬 출처:

<https://sjh836.tistory.com/112?category=719115>

배열 출처:

<https://m.blog.naver.com/dongji0529/220705747220>

<https://sjh836.tistory.com/112>

<https://dic1224.blog.me/80206351755>

리스트 출처:

<https://insightteller.tistory.com/entry/R-%EA%B8%B0%EC%B4%88-%EC%8B%A4%EC%8A%B5-%EC%9E%90%EB%A3%8C-2-%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EB%A9%94%ED%8A%B8%EB%A6%AD%EC%8A%A4-%EB%B0%B0%EC%97%B4?category=628138>

<https://wayhome25.github.io/cs/2017/04/17/cs-18-1/>

데이터 프레임 출처:

<https://insightteller.tistory.com/entry/R-%EA%B8%B0%EC%B4%88-%EC%8B%A4%EC%8A%B5-3-dataframe>

<https://statkclee.github.io/R-ecology-lesson/kr/03-data-frames.html>

csv 출처:

<https://m.blog.naver.com/PostView.nhn?blogId=swkim4610&logNo=220844239597&proxyReferer=https%3A%2F%2Fwww.google.com%2F>