강원대학교
AI 소프트웨어학과

머신러닝1
-데이터의 구조-

KNU
강원대학교

01 데이터의 구조

Dataframe

데이터 프레임(Dataframe 이란?)

- 데이터 프레임은 프로그래밍 및 데이터 분석에 일반적으로 사용되는 표 형식의 데이터 구조
- 행과 열로 구성된 다양한 형태를 가지고 있는 리스트의 집합
- 데이터 프레임에서 각 열은 변수 또는 특정 속성을 나타냄
- 각 행은 개별 관찰 또는 데이터 포인트를 나타냄
- 데이터 프레임은 다목적이며 숫자, 범주 및 텍스트 데이터를 포함하여 다양한 유형의 데이터를 처리할 수 있음

```
city <- c("Seoul", "Busan", "Daegu", "Seoul", "Busan", "Daegu", "Ulsan")
pm25 <- c(18, 21, 21, 17, 8, 11, 25)

df <- data.frame(city = city, pm25 = pm25)
```

데이터 프레임의 변수명

데이터 프레임의 변수 값

Dataframe

데이터 프레임(Dataframe 이란?)

- 데이터 프레임은 프로그래밍 및 데이터 분석에 일반적으로 사용되는 표 형식의 데이터 구조
- 행과 열로 구성된 다양한 형태를 가지고 있는 리스트의 집합
- 데이터 프레임에서 각 열은 변수 또는 특정 속성을 나타냄
- 각 행은 개별 관찰 또는 데이터 포인트를 나타냄
- 데이터 프레임은 다목적이며 숫자, 범주 및 텍스트 데이터를 포함하여 다양한 유형의 데이터를 처리할 수 있음

```
> id<-c(1:5)
> gender<-c("M","F","F","M","M")
> major<-c("Eng","Math","Com","Eng","Busi")
> salary<-c(2500, 2800, 2500, 3000, 2600)
> survey<-data.frame(ID=id, Gender=gender, Major=major, Salary=salary,
+                    stringsAsFactors = FALSE)
> survey
  ID Gender Major Salary
1  1      M   Eng   2500
2  2      F  Math   2800
3  3      F   Com   2500
4  4      M   Eng   3000
5  5      M  Busi   2600
```

Dataframe

데이터 프레임(Dataframe 이란?)

- 리스트와 배열 조작에서 사용하는 방법으로 데이터프레임 조작 가능
- 데이터 프레임의 모든 열은 길이가 같아야 함
- 데이터 프레임 크기가 큰 경우 내용의 일부를 확인하기 위해 head(), tail()함수를 사용 (n의 디폴트 값은 6)

```
> head(survey, n=3)    #survey의 내용을 앞에서 3행만큼 출력
  ID Gender Major Salary
1  1      M   Eng   2500
2  2      F  Math   2800
3  3      F   Com   2500
> tail(survey, n=3)    #survey의 내용을 뒤에서 3행만큼 출력
  ID Gender Major Salary
3  3      F   Com   2500
4  4      M   Eng   3000
5  5      M  Busi   2600
> survey$Salary        #Salary열을 벡터구조로 추출
[1] 2500 2800 2500 3000 2600
> survey[["Salary"]]   #Salary열을 벡터구조로 추출
[1] 2500 2800 2500 3000 2600
> survey["Major"]      #Major열 추출
  Major
1   Eng
2  Math
3   Com
4   Eng
5  Busi
> survey[2]            #2열을 데이터 프레임 구조로 추출
  Gender
1      M
2      F
3      F
4      M
5      M
> survey[[2]]          #2열을 벡터 구조로 추출
[1] "M" "F" "F" "M" "M"
```

Dataframe

데이터 프레임(Dataframe 이란?)

- 리스트와 배열 조작에서 사용하는 방법으로 데이터프레임 조작 가능
- $는 데이터프레임의 특정 변수를 추가하거나 불러올 수 있음

```
> survey
  ID Major Salary
1  1   Eng   2500
2  2  Math   2800
3  3   Com   2500
4  4   Eng   3000
5  5  Busi   2600
> survey$score=c(200,300,400,500,600)
> survey
  ID Major Salary score
1  1   Eng   2500   200
2  2  Math   2800   300
3  3   Com   2500   400
4  4   Eng   3000   500
5  5  Busi   2600   600
```

```
> survey$ss=survey$salary+survey$score
> survey
  ID Major Salary score   ss
1  1   Eng   2500   200 2700
2  2  Math   2800   300 3100
3  3   Com   2500   400 2900
4  4   Eng   3000   500 3500
5  5  Busi   2600   600 3200
```

Dataframe

데이터 프레임(Dataframe 이란?)

- 데이터 프레임의 형태를 파악하고, 잘못된 형태일 경우 변환 가능

변환할 변수=as.변환 값(변환할 변수)

summary(name_age_df) → 데이터 프레임의 변수 요약

```
name_age_df$Age=as.integer(name_age_df$Age)
name_age_df$Age=as.numeric(name_age_df$Age)
name_age_df$Age=as.factor(name_age_df$Age)
name_age_df$Age=as.logical(name_age_df$Age)
name_age_df$Age=as.character(name_age_df$Age)
```

```
> name_age_df
    name Age
1   John  28
2   Jane  32
3 Smith  45
4    Doe  22
```

Dataframe

데이터 프레임(Dataframe 이란?)

- 변수의 이름 변경

```
names(df)
```

```
names(df)[names(df)==“데이터프레임의 변수명"]=“변환할 변수명”
```

```
names(name_age_df)[names(name_age_df)=="name"]="Name"
name_age_df
```

```
  Name Age
1 John  28
2 Jane  32
3 Smith 45
4  Doe  22
```

Dataframe 인덱싱

데이터 프레임(Dataframe 이란?)

- 특정 조건에 맞게 인덱싱 할 수 있음

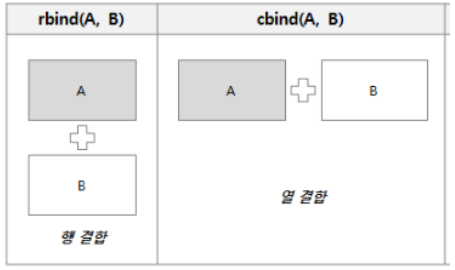
```
> survey[c(1,2)]        #1,2열 추출
  ID Gender
1  1      M
2  2      F
3  3      F
4  4      M
5  5      M
> survey[c(-1,-2)]      #1,2열을 제외한 나머지 열 추출
  Major Salary
1   Eng   2500
2  Math   2800
3   Com   2500
4   Eng   3000
5  Busi   2600
> survey[survey$Gender=="F",] #Gender=F인 행만 추출
  ID Gender Major Salary
2  2      F  Math   2800
3  3      F   Com   2500
> survey[survey$Major=="Eng"&survey$Salary>2600,]  #Major이 Eng이고 Salary>2600인 행추출
  ID Gender Major Salary
4  4      M   Eng   3000
```

Dataframe 인덱싱

데이터 프레임(Dataframe 이란?)

- nrow(), ncol(), names(), rownames(), colnames(), dim()을 이용하여 행과 열의 정보를 얻을 수 있음
- sort(), order()을 이용하여 자료를 정렬할 수 있음
- $를 통해 데이터 프레임에 새로운 변수를 추가하거나, 기존에 있는 변수의 값을 변환하고, rbind(), cbind()를 이용해 두 개의 데이터 프레임을 결합하거나 병합할 수 있음

```
> id<-c(1:5)
> gender<-c("M","F","F","M","M")
> major<-c("Eng","Math","Com","Eng","Busi")
> salary<-c(2500,2800,2500,3000,2600)
> survey<-data.frame(ID=id,Gender=gender,Major=major,Salary=salary,
+                    stringsAsFactors = FALSE)
> survey
  ID Gender Major Salary
1  1      M   Eng   2500
2  2      F  Math   2800
3  3      F   Com   2500
4  4      M   Eng   3000
5  5      M  Busi   2600
> nrow(survey)                    #행의 수
[1] 5
> ncol(survey)                    #열의 수
[1] 4
> dim(survey)                     #데이터 프레임 차원
[1] 5 4
> names(survey)                   #데이터 프레임 변수 이름
[1] "ID"     "Gender" "Major"  "Salary"
> colnames(survey)                #데이터 프레임 열 이름
[1] "ID"     "Gender" "Major"  "Salary"
> rownames(survey)                #데이터 프레임 행 이름
[1] "1" "2" "3" "4" "5"
> sort(survey$Salary)             #Salary를 크기 순서대로 정렬
[1] 2500 2500 2600 2800 3000
> order(survey$Salary)            #정렬된 Salary 값의 원래 위치
[1] 1 3 5 2 4
> survey[c(order(survey$Salary)),] #1,3,5,2,4행 순서대로 나열
  ID Gender Major Salary
1  1      M   Eng   2500
3  3      F   Com   2500
5  5      M  Busi   2600
2  2      F  Math   2800
4  4      M   Eng   3000
```




```
> survey$grade<-c(3.5,3.7,4.2,3.3,2.9)   #새로운 변수 추가
> survey
  ID Gender Major Salary grade
1  1      M   Eng   2500   3.5
2  2      F  Math   2800   3.7
3  3      F   Com   2500   4.2
4  4      M   Eng   3000   3.3
5  5      M  Busi   2600   2.9
> survey1<-data.frame(ID=id[1:3],Gender=gender[1:3],Major=major[1:3],
+                     Salary=salary[1:3],stringsAsFactors = FALSE)
> survey1
  ID Gender Major Salary
1  1      M   Eng   2500
2  2      F  Math   2800
3  3      F   Com   2500
> survey2<-data.frame(ID=id[4:5],Gender=gender[4:5],Major=major[4:5],
+                     Salary=salary[4:5],stringsAsFactors = FALSE)
> survey2
  ID Gender Major Salary
1  4      M   Eng   3000
2  5      M  Busi   2600
> survey3<-rbind(survey1,survey2)        #rbind()를 통해 행 결합
> survey3
  ID Gender Major Salary
1  1      M   Eng   2500
2  2      F  Math   2800
3  3      F   Com   2500
4  4      M   Eng   3000
5  5      M  Busi   2600
> Job<-c("office","profession","technician")
> survey4<-cbind(survey1, Job)           #cbind()를 통해 열 결합
> survey4
  ID Gender Major Salary        Job
1  1      M   Eng   2500     office
2  2      F  Math   2800 profession
3  3      F   Com   2500 technician
```

Dataframe 인덱싱

데이터 프레임(Dataframe 이란?)

- 특정 데이터만 사용하고 싶을 경우
- 조건에 맞는 데이터를 선택할 수 있음

subset(데이터 프레임, 조건, select= c(도출하고 싶은 변수 명1, 도출하고 싶은 변수 명2)

```
> age_gt_24 <- subset(name_age_df, name_age_df$Age > 24)
> age_gt_24
  LastName FirstName Age
5  Kim Min       jun  35
6 Park Min       jun  40
7   Kim Ji     young  34
8  Park Ji     young  35

> age_gt_24_name_age_only <- subset(name_age_df, name_age_df$Age > 24, select = c("LastName", "Age"))
> print(age_gt_24_name_age_only)
  LastName Age
5  Kim Min  35
6 Park Min  40
7   Kim Ji  34
8  Park Ji  35
```

Dataframe 인덱싱

데이터 프레임(Dataframe 이란?)

- 특정 데이터만 사용하고 싶을 경우
- 조건에 맞는 데이터를 선택할 수 있음

subset(데이터 프레임, 조건, select= c(도출하고 싶은 변수 명1, 도출하고 싶은 변수 명2)

```
> subset(survey, Major=="Com",c(Gender,Salary))    #Gender와 Salary에 대해 Major=Com인 행 추출
  Gender Salary
3      F   2500
> survey$Gender<-NULL  #Gender열 삭제
> survey
  ID Major Salary
1  1   Eng   2500
2  2  Math   2800
3  3   Com   2500
4  4   Eng   3000
5  5  Busi   2600
```

Dataframe 인덱싱

데이터 프레임(Dataframe 이란?)

- 데이터 프레임의 변수에 특정 기호가 존재할 때, 이를 구분해 두개의 변수로 나눌 수 있음
- 이때 separate 함수 사용

library(tidyr) # 사용 패키지
separate(데이터 프레임, col = " 지정한 변수 이름 " , into = c("생성변수1", " 생성변수2"), sep = " 나눌 기준기호")

```
> name_age_df <- data.frame(
+   Name = c("Kim Cheol-soo", "Lee Cheol-soo", "Kim Young-hee", "Lee Young-hee",
+            "Kim Min-jun", "Park Min-jun", "Kim Ji-young", "Park Ji-young"),
+   Age = c(20, 24, 21, 24, 35, 40, 34, 35),
+   stringsAsFactors = FALSE
+ )
> name_age_df <- separate(name_age_df, col = "Name", into = c("LastName", "FirstName"), sep = "-")
> print(name_age_df)
   LastName FirstName Age
1 Kim Cheol       soo  20
2 Lee Cheol       soo  24
3 Kim Young       hee  21
4 Lee Young       hee  24
5   Kim Min       jun  35
6  Park Min       jun  40
7    Kim Ji     young  34
8   Park Ji     young  35
```

데이터 전처리

- **데이터 불러오기(대용량)**

```
library(data.table)
hn_2009 <- fread('C:/Users/USER/Desktop/창균/건강조사/HN_2019.csv')
hn_2010 <- fread('C:/Users/USER/Desktop/창균/건강조사/HN_2020.csv')
hn_2011 <- fread('C:/Users/USER/Desktop/창균/건강조사/HN_2021.csv')
```

데이터 전처리

- **데이터 통합(같은 이름을 가지는 변수들 합치기)**

데이터프레임의 모든 변수 합치기

```
combined_df <- rbindlist(list(hn_2009, hn_2010, hn_2011), fill=TRUE)
fill=TRUE #누락된 변수에 대해 처리함
```

데이터의 차원이 어떻게 이루어져 있는지 판단하는 함수

```
dim(combined_df)
```

hn_2009에 저장되어 있는 변수들만 가지고 오고 싶을 경우

```
col=colnames(hn_2009)
col_2009 <- combined_df[, col, with = FALSE]
```

데이터 전처리

- **결측값 처리**

데이터 불러오기

```
df <- fread('C:/Users/USER/Desktop/창균/건강조사/HN_19~21.csv')
```

모든 결측치 제거

```
df_clean <- na.omit(df)
```

sapply(변수 및 데이터, 모두 적용하거나 실행하고 싶은 함수)

```
df_numeric <- df[, sapply(df, is.numeric), with=FALSE]
```
→ **수치형 값을 가지는 변수 분리**

```
df_character <- df[, sapply(df, is.character), with=FALSE]
```
→ **문자형 값을 가지는 변수 분리**

```
dim(df_numeric)
dim(df_character)
```

데이터 전처리

- **결측값 처리**

```
library(mice)
```

mice(데이터, m=, maxit=, method="pmm", seed=)

- **m : 몇 개의 후보를 추출할 것인가?**
- **maxit : 해당 작업을 몇 번 반복할 것인가?**
- **seed : 랜덤으로 일어나는 시행에서 그 값을 고정하는 것**

```
imputed_data <- mice(sub_df_numeric, m=5, maxit=10, method="pmm", seed=1235)
```

데이터 전처리

- **결측값 처리**

```
imputed_data <- mice(sub_df_numeric, m=5, maxit=10, method="pmm", seed=1235)
```

- "pmm": Predictive Mean Matching. 누락된 값을 예측한 후, 예측값에 가장 가까운 실제 관측값으로 대체함
- "norm": Bayesian Linear Regression. 연속형 변수에 대한 베이지안 선형 회귀를 사용하여 누락된 값을 예측함
- "norm.nob": Non-Bayesian Linear Regression. 선형 회귀를 사용하여 누락된 값을 예측함
- "logreg": Logistic Regression. 이진 범주형 변수에 대한 누락된 값을 예측함
- "polyreg": Polytomous Regression. 다범주형 변수에 대한 누락된 값을 예측함
- "cart": Classification and Regression Trees. CART 알고리즘을 사용하여 누락된 값을 예측함
- "rf": Random Forest. 랜덤 포레스트 알고리즘을 사용하여 누락된 값을 예측함
- "mean": 평균 대체, 변수의 평균값으로 누락된 값을 대체함
- "midastouch": Weighted Predictive Mean Matching. 가중치를 적용한 Predictive Mean Matching 방법
- "sample": 무작위 추출, 누락되지 않은 값 중에서 무작위로 값을 선택하여 누락된 값을 대체함

데이터 전처리

결측값 처리

```
imputed_data <- mice(sub_df_numeric, m=5, maxit=10, method="pmm", seed=1235)
```

처리된 데이터 선택

```
completed_data <- complete(imputed_data, 1) #첫번째 데이터를 사용
```

csv파일 저장

```
fwrite(completed_data, 'C:/Users/USER/Desktop/창균/건강조사/test.csv')
```

데이터 전처리

- **데이터 변환**

- **~ifelse : ~ifelse(범위값, 범위에 해당되는 값에 대한 변환 값, 변환하고 싶은 대상) → 하나의 조건에 대해서만 처리**

```
library(dplyr)
mutate(데이터, across(c("변수명1","변수명2","변수명3"), ~ifelse(범위값, 변환값, 변환대상)))
mutate(a, across(c("X20s","X30s","X40s","X50s","X60s"), ~ifelse(. > 6.0, "High", .)))
```

- **~case_when : ~case_when(범위값 ~ 변환값, TRUE ~ 변환하고 싶은 대상)**

```
mutate(a, across(c("X30s","X40s","X50s","X60s"), ~case_when(. >= 6.0 ~ 1, . >= 2.5 & . < 6.0 ~ 2, . < 2.5 ~ 3,  TRUE ~ .)))
mutate(a, across(c("X20s","X30s","X40s","X50s","X60s"), ~case_when(. >= 6.0 ~ "High", . >= 2.5 & . < 6.0 ~ "Medium", . < 2.5 ~ "Low", TRUE ~ as.character(.)
```