강원대학교
AI소프트웨어학과

재난안전 프로그래밍

Chapter 4. 데이터 전처리

Chapter 4. Data Preprocessing

4.1 데이터 전처리

- 데이터 전처리
  - 데이터를 분석 및 처리에 적합한 형태로 만드는 과정
  - 전처리 결과는 분석 결과에 직접적인 영향을 주기 때문에 매우 중요한 과정
  - 데이터 전처리 80%, 모델을 활용한 분석 10%, 정보 제공 10%
  - 데이터 통합(Data integration), 데이터 클리닝(Data cleaning), 데이터 변환(Data transformation), 데이터 축소(Data reduction) 등

Chapter 4. Data Preprocessing

4.1 데이터 전처리

- **데이터 불러오기(대용량)**

fread('경로/ .csv', fill= TRUE)

```
library(data.table)
hn_2009 <- fread('C:/Users/USER/Desktop/창균/건강조사/a.csv')
hn_2010 <- fread('C:/Users/USER/Desktop/창균/건강조사/b.csv')
hn_2011 <- fread('C:/Users/USER/Desktop/창균/건강조사/c.csv')
```

Chapter 4. Data Preprocessing

4.1 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]
```

Chapter 4. Data Preprocessing

4.1 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**

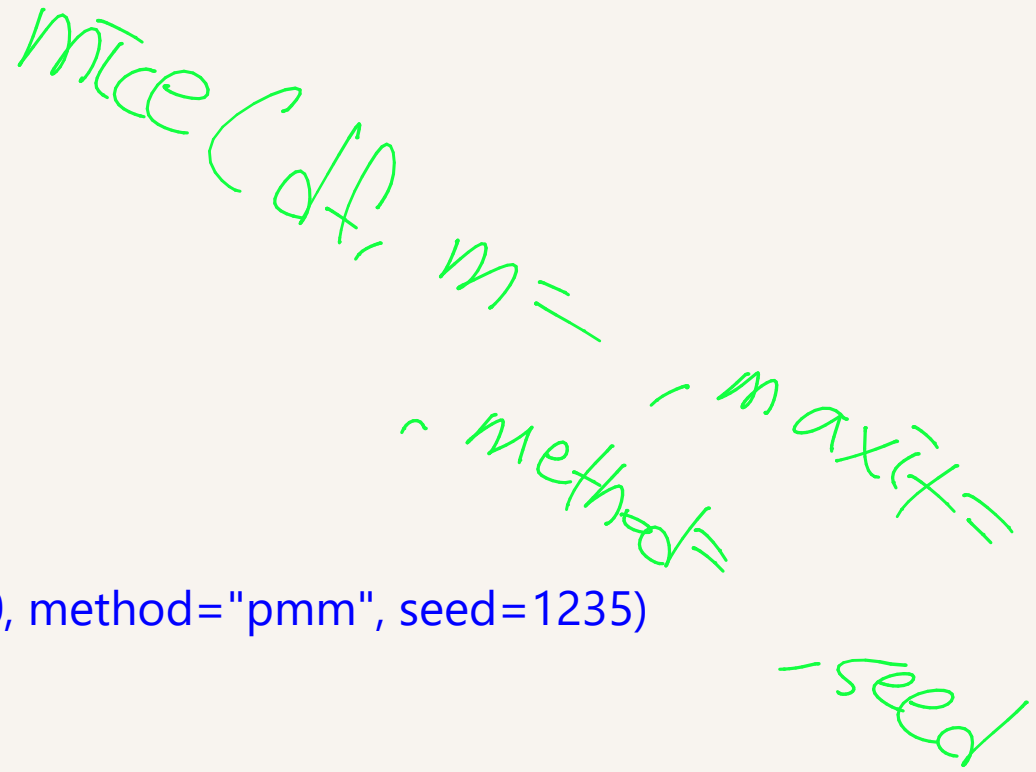
```
df_character <- df[, sapply(df, is.character), with=FALSE]
```
→ **문자형 값을 가지는 변수 분리**

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

Chapter 4. Data Preprocessing

4.1 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)
```

Chapter 4. Data Preprocessing

4.1 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. 베이지안이 아닌 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

Chapter 4. Data Preprocessing

4.1 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')
```

Chapter 4. Data Preprocessing

4.1 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(.))))
```