

# NA Imputation

# INDEX

---

0. 자료 제작 배경

1. 결측값

2. Imputation

3. 결론

## 0) 자료 제작 배경

- 학회 활동을 하면서, 정말 많은 NA들을 만나봤고 이것들로 인해 많이 고통 받았었습니다!
- 학회에 결측 처리에 대한 자료는 있지만, 그 알고리즘에 대해서 다른 자료가 없는 것 같아 패키지와 함께 해당 자료를 만들게 되었습니다.
- 이 자료를 읽기 전 먼저 선배님들이 만들어 두신 2018년도 1학기 알파팀 PPT을 읽어보시기 바랍니다!
- 이 자료는 정확하지 않을 가능성이 있으니, 이해를 돕기 위한 수준에서만 사용하시고 궁금하신 분은 직접 공부하기를 추천합니다!!!



## 1) 결측값이란?

1

### 결측값의 종류



결측값(Missing Value, NA)



자료 수집과정에서 측정되지 않거나  
기입과정에서 누락된 데이터

**MCAR**  
(Missing  
Completely  
at Random)

**MAR**  
(Missing at  
Random)

**MNAR**  
(Missing Not  
At Random)

(참고 : 2018년도 1학기 알파팀 PPT)

## 2) 결측값을 잘 처리해야 하는 이유는?

### # If ) 그제 삭제?

#### ① 결측값을 그냥 지울 경우 소중한 데이터가 사라진다

> 단순히 row의 개수가 감소하는 것을 떠나, NA가 포함된 이유 하나로 중요 정보들이 삭제될 수 있다!

#### ② 1번의 결과를 통해 편향된 데이터셋을 얻을 수 있다

> 표본이 줄어드는 것도 문제지만, 줄어드는 과정에서 편향된 정보들만 남을 수 있다!

#### ③ 모델의 정확도가 떨어진다

> 1,2번으로 인해 결과적으로 모델의 정확도가 떨어져 올바른 예측이 힘들 수 있다

### # If ) 내버려 두면?

#### ① 사용할 수 있는 모델이 제한되며, 모델의 정확도가 떨어진다

> 자체적으로 NA를 다룰 수 없는 모델도 있고, 자체적으로 다루는 모델도 이를 단순한 방법으로 처리하기에 좋은 정확도를 기대할 수가 없다. (AKA. CatBoost)

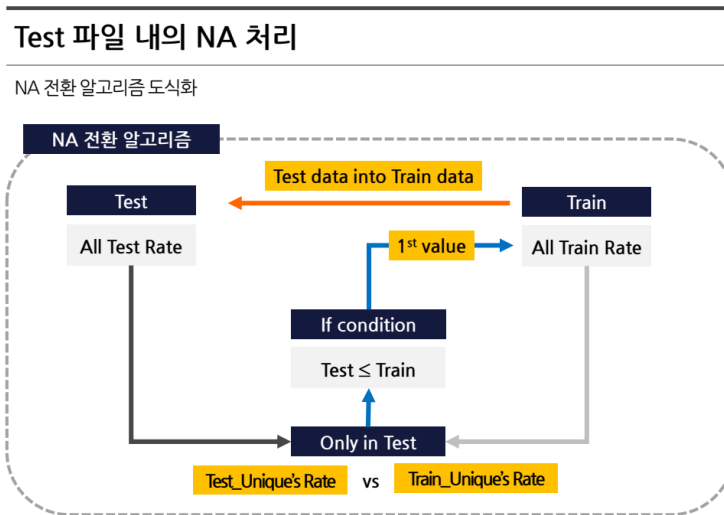
## 2) 결측값을 잘 처리해야 하는 이유는?

# If ) 잘 처리한다면?

결측값을 잘 처리할 경우, 모델의 정확도 향상을 꾀할 수 있다!

**igaworks**  
최종 데이터  
Data\_f4의 문제점  
파생변수 제작  
최종 데이터셋

- 54 -



실제로 성능이 더 좋아짐!

(참고 : 2020년도 1월 척척학사팀 공모전 자료)

### 3) 결측값 처리 방법

## 2

#### 결측값의 처리방법



#### 결측값의 처리방법

[ 주로 살펴 볼 내용! ]

##### Listwise Deletion

결측값이 있는 행을  
통째로 삭제해  
Complete cases만을  
대상으로 분석

MCAR일 때만

표본 수 감소로 인해  
분석의 정확도, 일반성 ↓  
통계적 편향

##### Pairwise Deletion

결측자료를 짝별로 지우기  
그때 그때 각 쌍의 변수들  
에 대해 누락된 자료 제거

MCAR일 때만

정보의 손실 ↓  
분석 과정별로 표본수가  
달라져  $\widehat{se} \uparrow$

##### Imputation

결측값을 특정 값으로  
대체한 후 분석 진행

## 4) Imputation의 종류

### Single Imputation

Mean / Median / Min / Max

Imputation using model

### Multiple Imputation

KNN

MICE

Joint Modeling (이전 ppt 참고!)



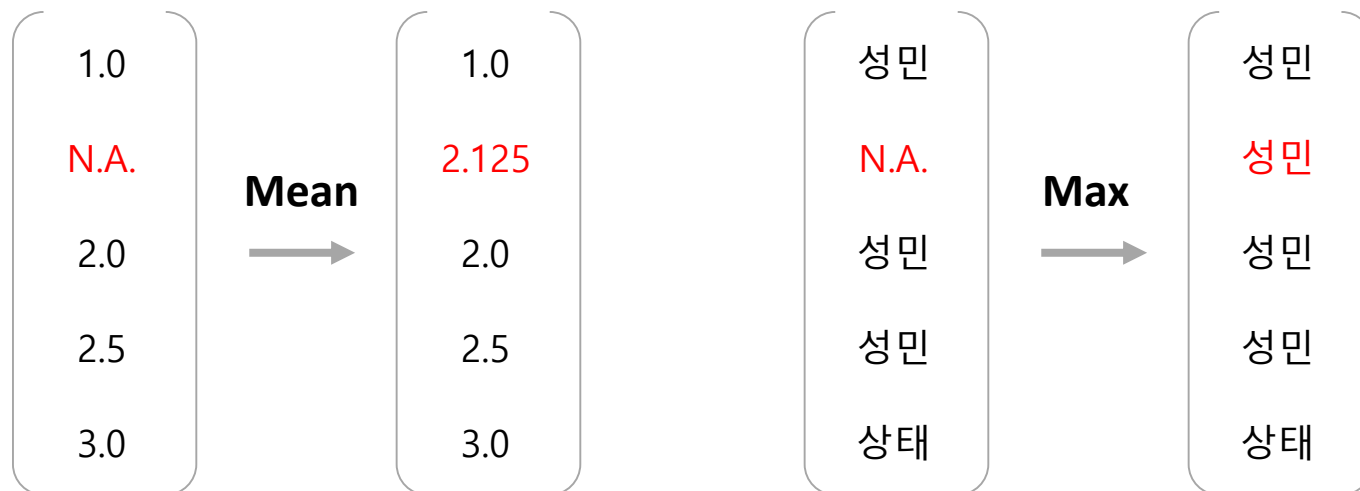
## 1) Single Imputation

# 하나의 변수에 존재하는 결측값만을 대체

### ① 단순 대입법 (Mean / Median / Max / Min)

- > 이름 그대로 단순하게 생각하면 된다! 바로 데이터에 존재하는 요약통계량을 이용하여 NA를 채우는 것!
- > 연속형 변수의 경우 보통 Mean을 사용하고, 범주형 변수는 최빈값과 최소값(?) 등을 사용한다

EX)



## 1) Single Imputation

# 하나의 변수에 존재하는 결측값만을 대체

### ① 단순 대입법의 문제?

> 데이터의 BIAS를 높일 수 있으며, 모델링 결과에 안 좋은 영향을 끼칠 수 있다

이름	키	농구 수업 만족도
신성민	185	6
김상태	178	9
김찬영	195	10
오정민	178	-
⋮		
박정현	151	-
박서영	150	-

키가 작으면 농구를 싫어해서 수업 만족도 입력을 안할 확률이 높다



그러나, 이러한 사실을 무시하고 평균으로 전부 기입하면

박정현	151	7
박서영	150	7



위와 같이 키가 작은데도 농구를 매우 좋아하는 데이터가 생긴다

(※ 농구 수업 만족도 평균이 7이라고 가정)

# 1) Single Imputation

# 하나의 변수에 존재하는 결측값만을 대체


## ② 모델 기반 NA imputation

- > 결측값이 존재하는 변수를 반응변수로 나머지를 독립변수로 두고 결측값을 예측하여 채워 넣는 방식
- > 연속형 변수의 경우 회귀모형을, 범주형 변수의 경우 의사결정나무를 이용한다

EX)

$$Y = x_1 + 2x_2 + \varepsilon$$

5.1	1.0	2.0
N.A.	1.5	2.2
6.2	2.0	2.0
7.1	3.0	2.0
9.3	2.5	3.5



5.1	1.0	2.0
5.9	1.5	2.2
6.2	2.0	2.0
7.1	3.0	2.0
9.3	2.5	3.5

## 1) Single Imputation

# 하나의 변수에 존재하는 결측값만을 대체

### ② 모델 기반 NA imputation

모델을 사용할 경우 각 데이터에 알맞은 모델을 선택해 사용하면 된다! 각자 맞는 모형 선택하자

EX)

솔로	169	72
N.A.	180	68
솔로	158	48
커플	165	56
커플	185	80



솔로	169	72
커플	180	68
솔로	158	48
커플	165	56
커플	185	80

```
> library(rpart)
> m <- rpart(
+   survived ~ pclass + sex + age + sibsp + parch + fare + embarked,
+   data=titanic.train)
> p <- predict(m, newdata=titanic.train, type="class")
> head(p)
      1      3      4      5      6      7
survived survived  dead survived  dead survived
Levels: dead survived
```

(※이런 감성이다)

# 1) Single Imputation

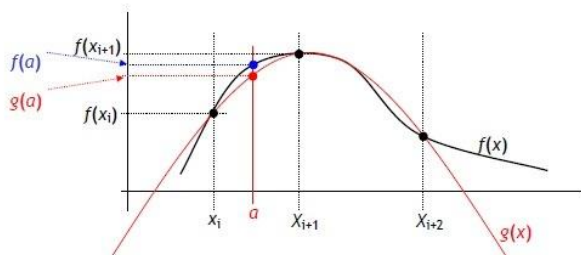
# 하나의 변수에 존재하는 결측값만을 대체

## ② 모델 기반 NA imputation (Time Series)

- > 하지만 시계열 변수일 경우 기존의 회귀 방법을 통해 접근하면, 올바른 계산이 안될 가능성이 높다
- > 그러니 시계열 변수일 경우 다른 방법을 사용하면 된다

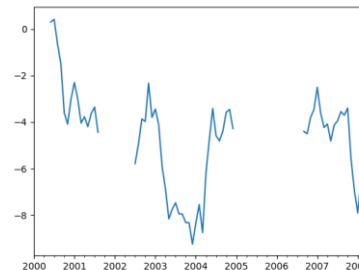
# interpolation (보간법)

- > 두 점 (n 차원의 데이터)가 주어졌을 때 그 사이에 위치한 값을 알려진 값으로 추정하는 방법
- > Interpolation을 구하는 식도 여러가지가 있으니 (linear, poly ..) 각자 상황에 맞게 사용하면 된다

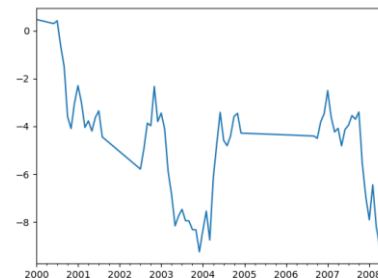


(※통수에서 봤던 그림)

전 )



후 )



(참고 : linear interpolation from pandas documentation)

## 1) Single Imputation

# 하나의 변수에 존재하는 결측값만을 대체

### ② 모델 기반 NA imputation (Time Series)

- > Interpolation 방법 외에도 시계열 모델을 활용한 아래와 같은 방법들 또한 있다!
- > 적극적으로 패키지를 활용해서 알맞게 사용해보자

# If ) 시계열 모델?

Function	Option	Description
na_interpolation	linear	Imputation by Linear Interpolation
	spline	Imputation by Spline Interpolation
	stine	Imputation by Stineman Interpolation
na_kalman	StructTS	Imputation by Structural Model & Kalman Smoothing
	auto.arima	Imputation by ARIMA State Space Representation & Kalman Sm.
na_locf	locf	Imputation by Last Observation Carried Forward
	nocb	Imputation by Next Observation Carried Backward
na_ma	simple	Missing Value Imputation by Simple Moving Average
	linear	Missing Value Imputation by Linear Weighted Moving Average
	exponential	Missing Value Imputation by Exponential Weighted Moving Average

(참고 : R package - ImputeTS)

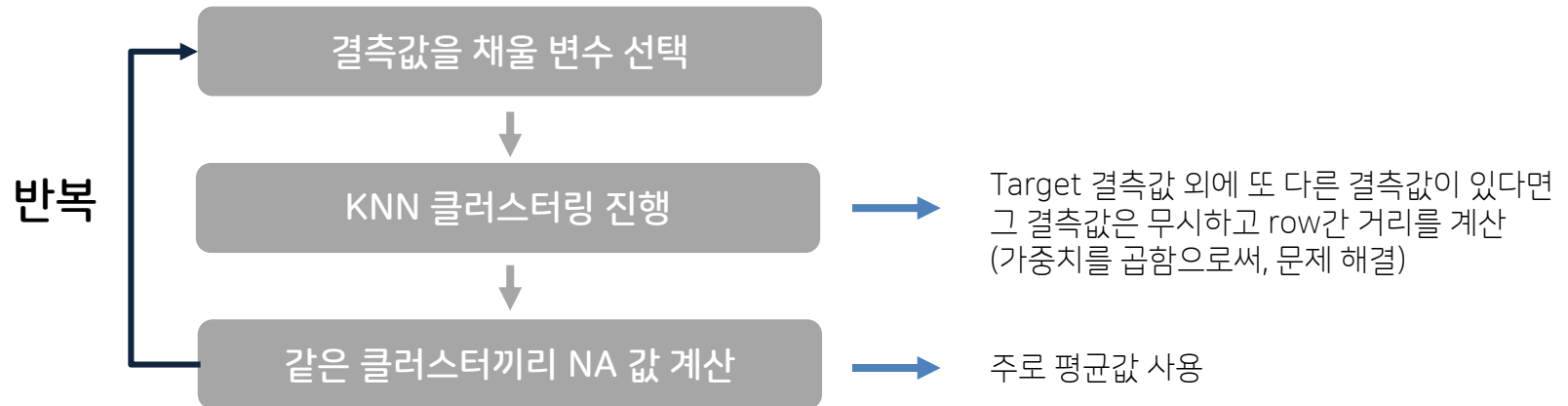
## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation

- > 데마 발표를 열심히 들었다면, 어떤 것인지 잘 알 수 있는 KNN을 활용하면 데이터를 채울 수 있다
- > 쉽게 이해하자면, 같은 클러스터에 속한 데이터들을 바탕으로 NA를 채운다고 생각하면 된다

EX)



※ 사이트마다 각자 다른 알고리즘을 소개하고 있기에 이해를 돕기 위해 scikit-learn 것을 준비했습니다.

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation (step by step)

Step 1 : 결측값을 채워 넣을 변수 선정

	Friends	HIMYM	GOT	Suits	BreakinBad
0	80	30	7	14	27
1	44	-	10	0	29
2	-	85	25	5	88
3	50	70	74	9	49
4	29	54	49	20	-

- 'Friends' 변수를 NA 값을 채우기 위한 첫 번째 변수로 선정



## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation (step by step)

Step 2 : 해당 row와 다른 row간의 거리 계산 (target 변수 제외)

	Friends	HIMYM	GOT	Suits	BreakinBad
0	80	30	7	14	27
1	44	-	10	0	29
2	-	85	25	5	88
3	50	70	74	9	49
4	29	54	49	20	-



	Distance
0	$\frac{4}{4} * \text{distance}(0,2)$
1	$\frac{4}{3} * \text{distance}(1,2)$
3	$\frac{4}{4} * \text{distance}(3,2)$
4	$\frac{4}{3} * \text{distance}(4,2)$

- Index 2번과 0,1,3,4번의 거리를 각각 구함 이때 거리는 **nan\_euclidian** 활용

※ nan\_Euclidia

- Na가 있을 경우 — Na가 있는 열을 제외하고 계산 \* **가중치**
- Na가 없는 경우 — 기존의 Euclidian distance

$\frac{\text{전체 변수}}{\text{NA 제외 변수}}$

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation (step by step)

Step 3 : Neighbor에 따라 클러스터링 진행

	Distance
0	$\frac{4}{4} * \text{distance}(0,2)$
1	$\frac{4}{3} * \text{distance}(1,2)$
3	$\frac{4}{4} * \text{distance}(3,2)$
4	$\frac{4}{3} * \text{distance}(4,2)$

	Friends	HIMYM	GOT	Suits	BreakinBad
2	-	85	25	5	88
3	50	70	74	9	49
4	29	54	49	20	-

- n=2라는 가정하에, 3번과 4번이 2번과 가장 가까웠기에 (2,3,4)로 묶임

- 초기에 n의 값을 어떻게 정하냐에 따라 달라짐

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation (step by step)

Step 4 : 클러스터군의 값들을 이용하여 imputation 진행하기

	Distance
0	$\frac{4}{4} * \text{distance}(0,2)$
1	$\frac{4}{3} * \text{distance}(1,2)$
3	$\frac{4}{4} * \text{distance}(3,2)$
4	$\frac{4}{3} * \text{distance}(4,2)$

	Friends	HIMYM	GOT	Suits	BreakinBad
2	39.5	85	25	5	88
3	50	70	74	9	49
4	29	54	49	20	-

- 같은 군집에 속한 3,4번의 friends 변수가 50, 29임을 이용해 계산
- 평균을 이용하여  $(50+29)/2 = 39.5$  라는 새로운 값을 채울 수 있게 됨 (평균/최빈 등은 분석자가 선택)
- 이 과정을 계속해서 반복

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ① KNN Imputation의 문제

- > 간단하다! 범주형 변수가 있으면 사용하기가 힘들다
- > 물론 범주형 변수를 연속형으로 인코딩하는 등의 방법을 통해 해결할 수 있다곤 하지만 정확하지 않다
- > 고차원 데이터에서 (column의 개수가 많아지면) 올바른 계산이 되지 않을 수 있다
- > 기존의 knn 방식이 가지고 있는 한계와 동일하다고 생각하면 된다!

### ② KNN Imputation의 장점

- > 사전에 가지고 있는 데이터를 바탕으로 NA를 추정해서 단순대입법보다는 더 좋은 성능을 보인다
- > 기존의 knn 방식이 가지고 있는 장점과 동일하다고 생각하면 된다!

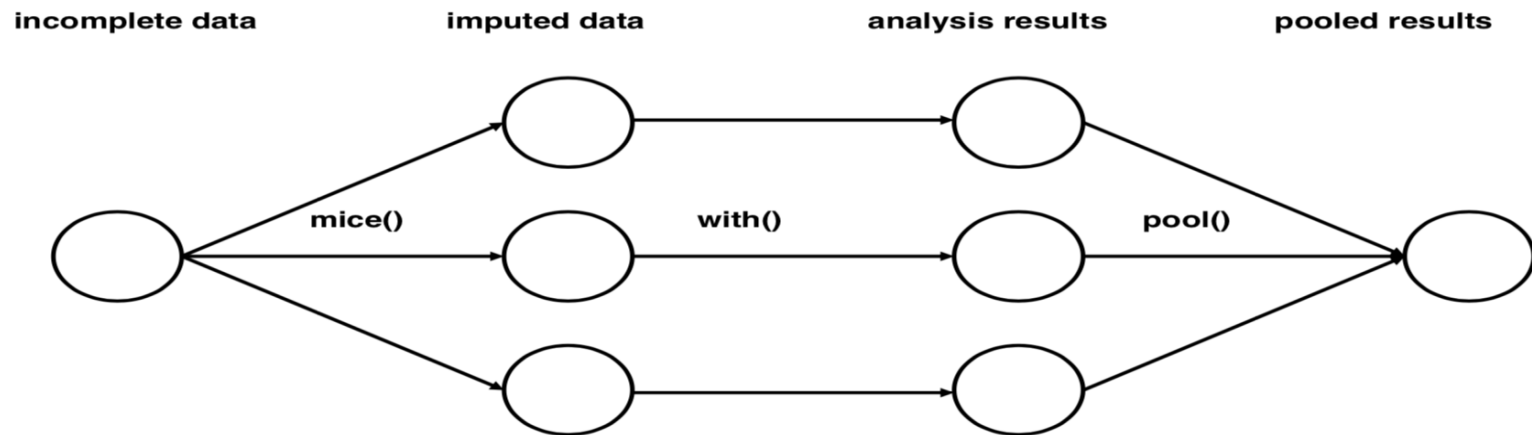
“ 그럼 연속/범주형 변수 둘 다 NA가 존재하는 데이터는 어떻게 해야하죠? ”

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (Multiple imputation by chained equations)

- > KNN에서 다루지 못한 범주형 변수의 NA처리를 해결하는 방법 중 하나이다
- > 이름부터 굉장히 무서운데 직관적으로 생각하면 된다. Multiple하게 imputation한다고만 이해해보자
- > 개괄적으로 설명을 하면 mice는 크게 3단계로 나뉜다. 데이터를 여러 개로 만들어, **계산**하고, **분석**해서, **합친다**!



※ MICE에 대한 설명 중 제일 깔끔하다고 생각되는 자료 :

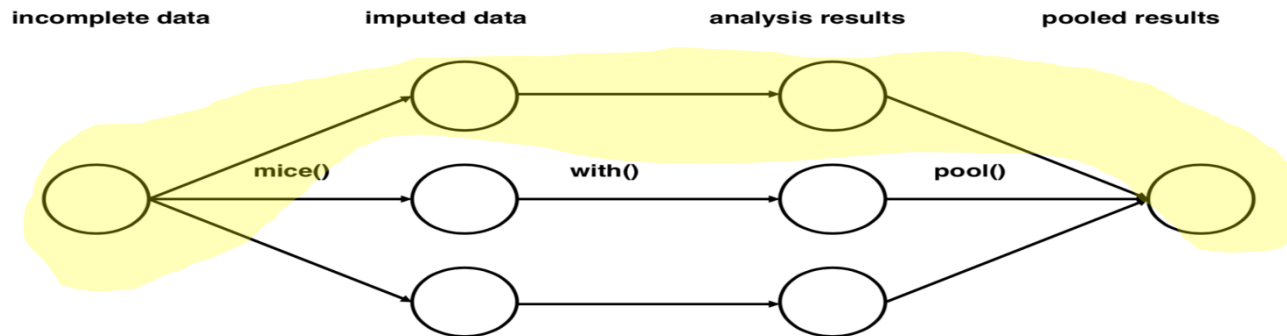
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/>

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (Multiple imputation by chained equations)

> 앞서 보여준 그림을 조금 나눠보자면 다음처럼 설명할 수 있다



(주스를 흘린 것 같지만, 작성자의 손이 문제이니 넘어가자..)

- > 한 번의 시행 동안 single imputation을 연속적으로 수행해 모든 변수의 NA를 채운다! (노란색 부분)
- > 모든 NA가 채워지면 또 다른 데이터셋을 가지고 위의 과정을 수행한다 (노란색과 동일하게 생긴 나머지 부분)
- > 각기 다른 데이터셋에서 구해진 NA들이 수렴값을 가질 때 멈추고 데이터를 합친다! (마지막 pooled 부분)

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 1 : 결측값들을 하나의 단순대입법으로 채워 하나의 데이터셋 형성

# Original Data

A	B	C
5.1	1.0	N.A.
N.A.	1.5	2.2
6.2	N.A.	2.0
7.1	3.0	2.0
9.3	2.5	N.A.

Mean을 사용



# 후보 데이터 1

A	B	C
5.1	1.0	2.06
6.925	1.5	2.2
6.2	2	2.0
7.1	3.0	2.0
9.3	2.5	2.06

## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 2 : 결측값이 있는 변수들마다 돌아가면서 알맞은 모델로 적합한 후, 예측값을 NA 위치에 다시 대입

# 후보 데이터 1 : 회귀 사용

$$\hat{A} = -17.21 + 1.8B + 10C$$

$$\hat{B} = 12.17 + 0.365A - 6.168C$$

$$\hat{C} = 1.98 + 0.04A - 0.0996B$$

A	B	C
5.1	1.0	2.06
7.382	1.5	2.2
6.2	2	2.0
7.1	3.0	2.0
9.3	2.5	2.06

6.925 → 7.382



A	B	C
5.1	1.0	2.06
7.382	1.5	2.2
6.2	2.096	2.0
7.1	3.0	2.0
9.3	2.5	2.06

2 → 2.096



A	B	C
5.1	1.0	2.08
7.382	1.5	2.2
6.2	2.096	2.0
7.1	3.0	2.0
9.3	2.5	2.10

2.06 → 2.08  
2.10



## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 2 : 결측값이 있는 변수들마다 돌아가면서 알맞은 모델로 적합한 후, 예측값을 NA 위치에 다시 대입

# 다른 모델 사용

Method	Description	Scale type	Default
pmm	Predictive mean matching	numeric	Y
norm	Bayesian linear regression	numeric	
norm.nob	Linear regression, non-Bayesian	numeric	
mean	Unconditional mean imputation	numeric	
2l.norm	Two-level linear model	numeric	
logreg	Logistic regression	factor, 2 levels	Y
polyreg	Polytomous (unordered) regression	factor, >2 levels	Y
lda	Linear discriminant analysis	factor	
sample	Random sample from the observed data	any	

(※ 이렇게 상황에 맞게 각자의 모델로 계산을 하게 된다)

## 2) Multiple Imputation


# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 3 : step 2에서 한 과정을 N번 반복해서 NA값을 계속해서 업데이트

# 10번의 업데이트가 완료되었다는 가정 하의 데이터셋

A	B	C
5.1	1.0	<b>2.08</b>
<b>7.382</b>	1.5	2.2
6.2	<b>2.096</b>	2.0
7.1	3.0	2.0
9.3	2.5	<b>2.10</b>



A	B	C
5.1	1.0	<b>1.95</b>
<b>7.52</b>	1.5	2.2
6.2	<b>2.4</b>	2.0
7.1	3.0	2.0
9.3	2.5	<b>2.3</b>

(※ 우측의 결과는 이해를 돕기 위한 것이며, 실제 계산한 값은 아닙니다)

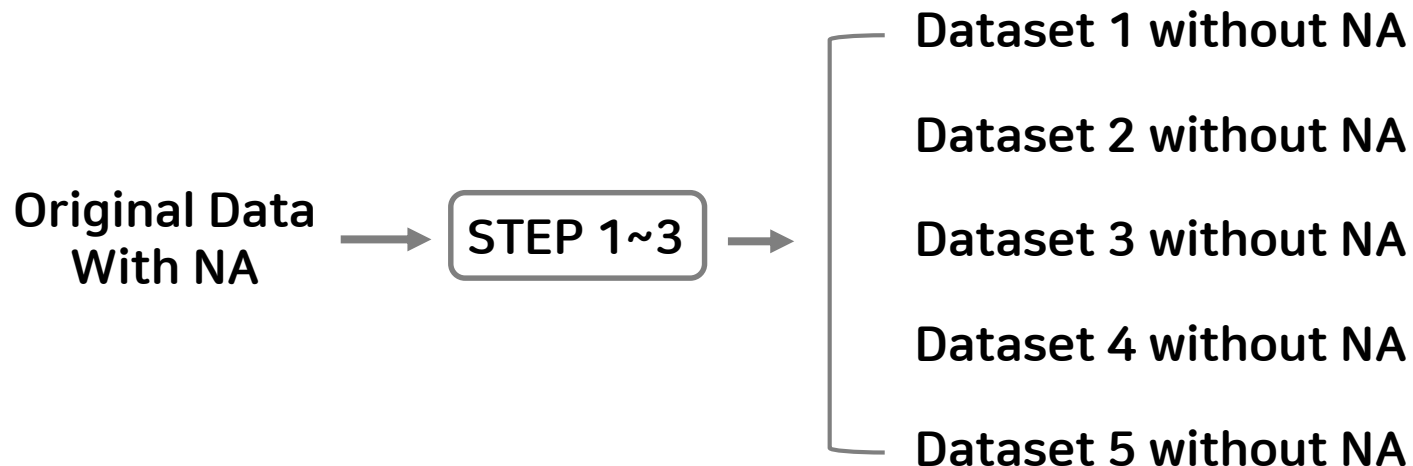
## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 4 : 여러 데이터셋을 만들어 step 1~3에서 한 과정을 반복 (default = 5)

# 과정은 같지만, 결과값만 다르게 나오는 5개의 데이터셋 형성



## 2) Multiple Imputation

# 여러 변수에 걸쳐 존재하는 결측값을 대체

### ② MICE (step by step)

Step 5 : 만들어진 데이터셋을 분석과 풀링을 진행하여 최종 데이터셋 선정

# 5개의 데이터가 있다고 가정

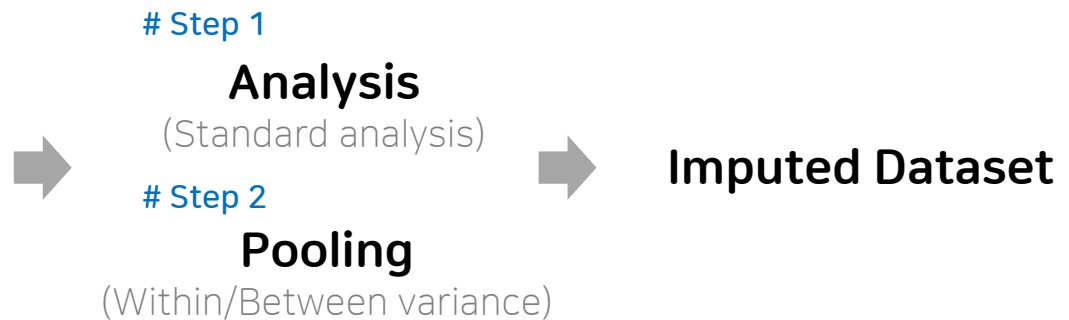
Dataset 1 without NA

Dataset 2 without NA

Dataset 3 without NA

Dataset 4 without NA

Dataset 5 without NA



## 2) Multiple Imputation

### # 여러 변수에 걸쳐 존재하는 결측값을 대체

#### ① MICE 기본 가정과 문제

- > MICE는 MAR, 즉 모든 NA가 랜덤하게 생겨났다는 가정하에 프로세스가 이루어진다
- > 만약 앞에서 보여줬던 농구 예시처럼, 랜덤으로 생겨난 NA가 아닐 경우 올바른 imputation이 안될 수 있다

#### ② MICE 사용시 유의사항

- > 앞의 알고리즘에서 보았듯이, MICE는 주어진 데이터를 활용하여 NA값을 '예측'하는 것이다
- > 때문에 MICE를 실행할 때는 변수들이 모든 관계가 잘 반영된 데이터셋을 이용해야 한다.
- > 예를 들자면, 내가 연애허수를 맞추는 모델을 만들고 싶고 X 변수에는 키, 근육량, 옷 수량, 신발 개수 등이 있다
- > 분명 옷이 많은 사람은 신발도 많을 것이고, 키가 큰 사람은 근육량도 상대적으로 더 많을 것이다.
- > 이때 최종적으로 우리가 모델에 키, 옷 수량만 사용한다고 해서 이 둘 만 가지고 MICE를 돌리면 안된다는 것!
- > 앞서 제시한 자료에서 작성자는 'general'한 데이터를 만든다고 말했으니 딱 이해가 될거라 생각합니다!

## 드디어 피피티가 끝났다

# 다음주 패키지에 이거 낼거예요 :)

- NA IMPUTATION을 검색할 때 가장 기본적으로 나오는 것들에 대해 다뤄봤습니다!
- 이거 말고도 여러가지가 있지만, 기본적으로 이 정도만 알아도 충분히 문제를 해결하실 수 있다고 생각합니다!
- 또한, 이렇게 알고리즘에 대해서 이해를 하면 각자 필요한 상황에 맞게 변화해서 적용할 수 있을거라 기대합니다!
- 더 찾아보고 싶은 분들은, 제가 일단 참고한 사이트들을 올려둘테니 확인해보시면 됩니다!
- P-SAT 파이팅.



A background image showing several salmon swimming in a river with white water rapids. The fish are silvery with hints of pink and orange, typical of salmon during spawning season. They are positioned at various points in the frame, swimming towards the left. The water is turbulent and white with foam.

**감사합니다.**

## # 결측값

<https://m.blog.naver.com/tjdudwo93/220976082118>

## # pandas interpolation

<https://rfriend.tistory.com/264>

## # MICE

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/>

## # impute TS

<https://cran.r-project.org/web/packages/imputeTS/vignettes/imputeTS-Time-Series-Missing-Value-Imputation-in-R.pdf>

## # knn

<https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html>

<https://towardsdatascience.com/the-use-of-knn-for-missing-values-cf33d935c637>

<https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/>

<https://www.mathworks.com/help/bioinfo/ref/knnimpute.html>

<https://core.ac.uk/download/pdf/231139578.pdf>





**THANK YOU**

