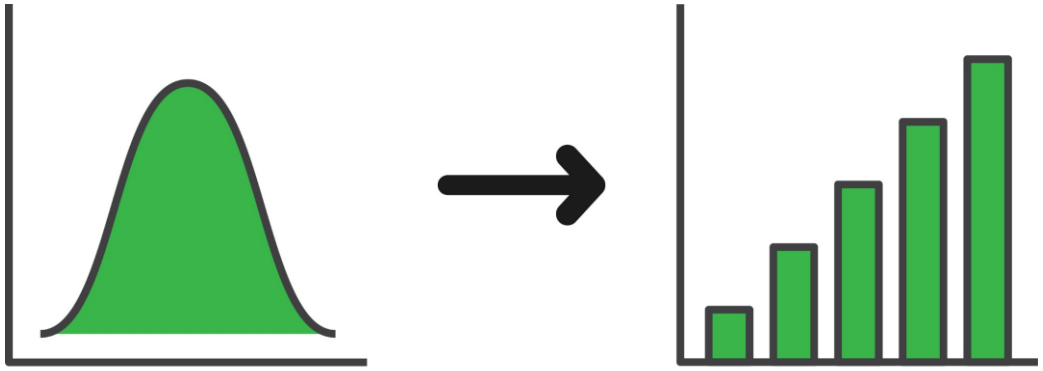
10.2 특징 추출 및 변환 실습

Binning

- **구간화**(binning): 특정 변수를 범주형 변수로 변환
  - 목적: 강건한(robust) 모델 생성, 과적합(overfitting) 방지
  - 수치형/범주형 변수에 모두 적용 가능
    - 수치형 예) 소득 ⇨ 소득 분위, 나이 ⇨ 연령대
    - 범주형 예) 양산동, 세교동 ⇨ 오산시, 진안동, 봉담읍 ⇨ 화성시



Discretization Process

Source: Analytics India Magazine

Binning ex)

- 구간화 예: 3학년 11반 학생들의 키

잘못 입력된 값(이상치)

$$X = \{155, 167, 173, 170, 171, 160, 275\}$$

  - 평균: 181.6cm
  - 이상치가 일정 수준 이상으로 측정될 경우 평균은 데이터를 대표하지 못함
  - 구간화 결과:

$x < 160$: 1명

$160 \leq x < 170$: 2명

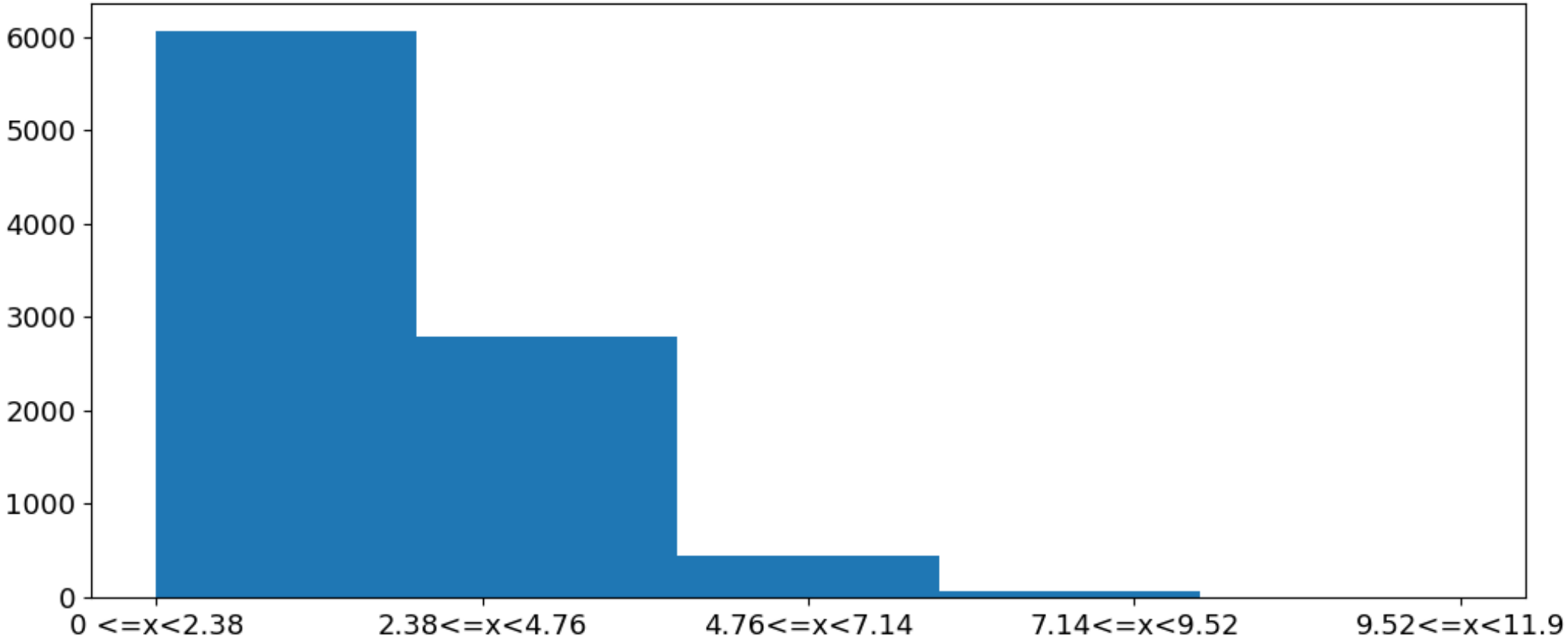
$\underline{170 \leq x < 180}$: **3명**

$180 < x$: 1명

구간화를 통해 얻어진 결과가
평균보다 데이터를 잘 설명함

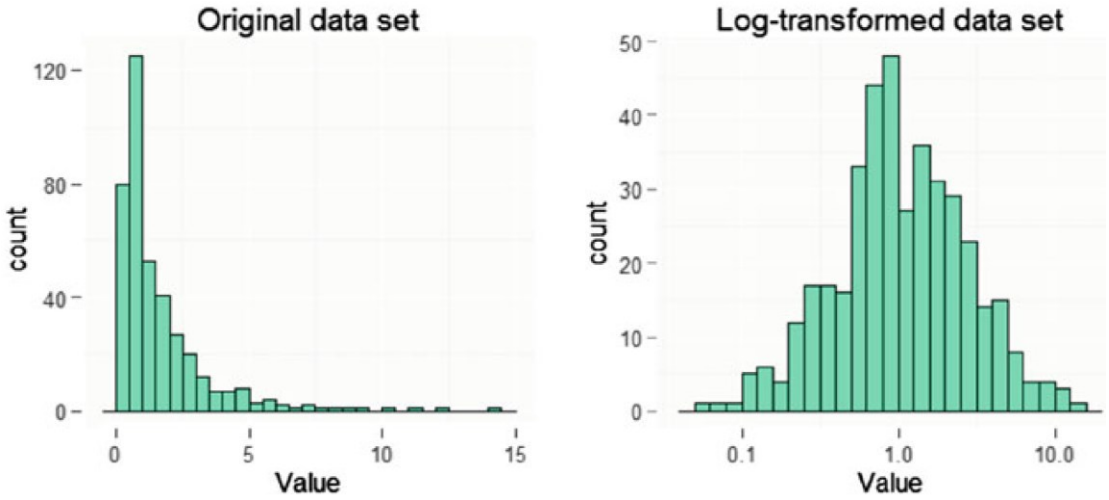
Exercise: Binning

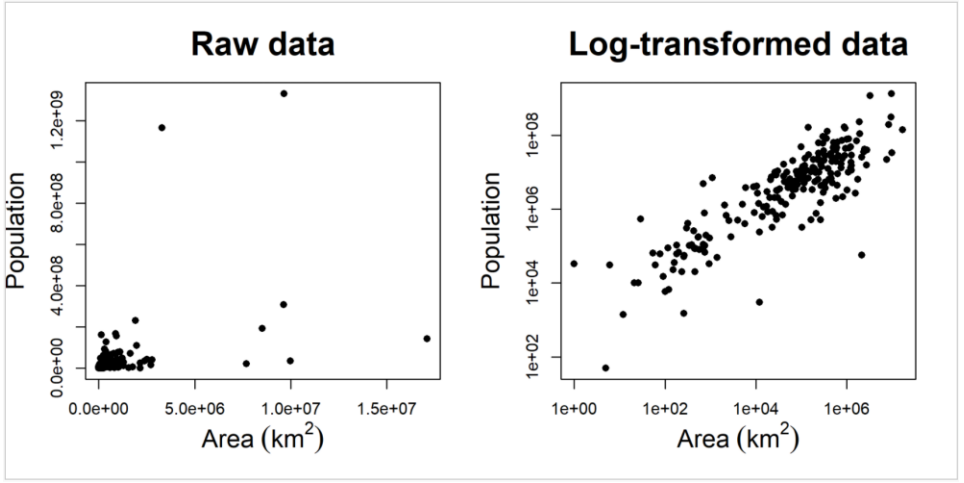
- 예제: Air Quality 일산화탄소(CO)의 구간화



Log Transformation

- **로그 변환**: **우편향된**right-skewed 데이터 분포를 **정규 분포**에 가깝게 변환
  - 이상치에 강건한 모델 생성
  - 정규 분포에 적합한 알고리즘, 모수적 방법 적용이 용이해짐



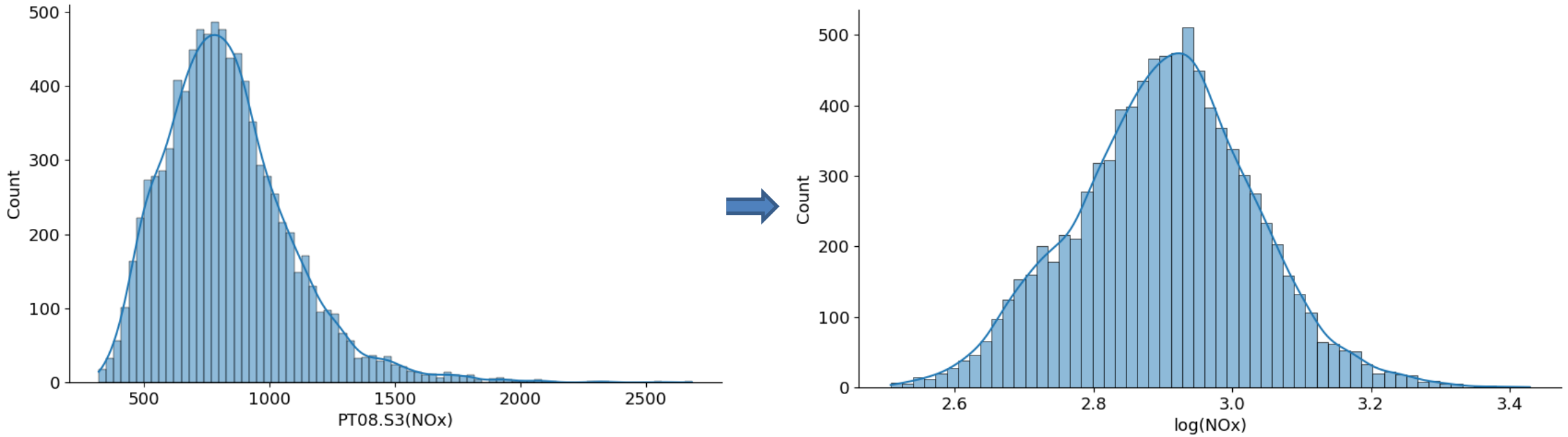


Source: Secondary Analysis of Electronic Health Records, Springer Nature, 2016

Exercise: Log Transformation

- 예제: Air Quality 질소산화물(NOx)의 로그 변환

```
df['log'] = np.log10(df['PT08.S3(NOx)'])
```



로그 변환 전/후의 질소산화물 변수 분포

Encoding

- **인코딩**: **범주형 변수**를 **수치형 변수**로 변환
  - 대부분의 모델/알고리즘은 수치형 데이터를 입력 받음

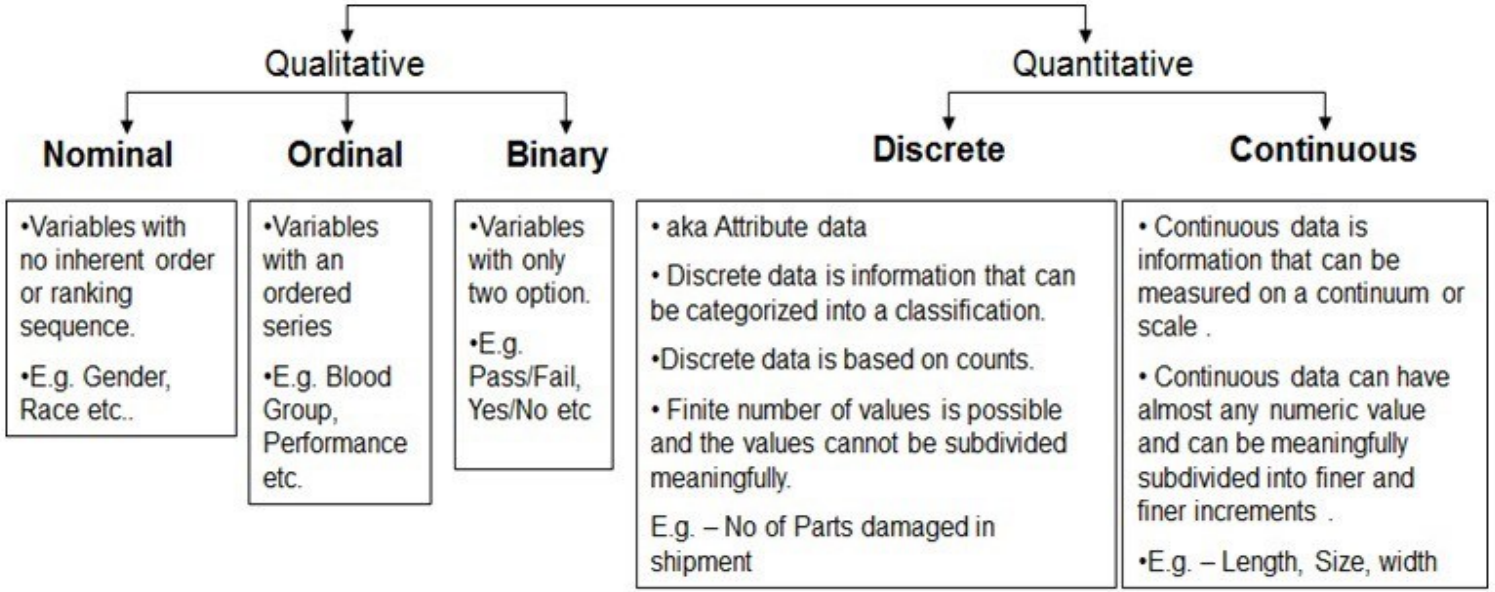


그림. 변수의 종류
(source: towards data science)

Label Encoding

- **레이블 인코딩**: 개별 범주를 특정 숫자 값(레이블)으로 표현
  - 순서형(ordinal) 데이터에 적용할 경우 데이터 순서, 순위 등을 보존 가능

BRIDGE-TYPE (TEXT)	BRIDGE-TYPE (NUMERICAL)
Arch	0
Beam	1
Truss	2
Cantilever	3
Tied Arch	4
Suspension	5
Cable	6

SAFETY-LEVEL (TEXT)	SAFETY-LEVEL (NUMERICAL)
None	0
Low	1
Medium	2
High	3
Very-High	4

그림. 명목형(좌), 순서형(우) 데이터에 대한 레이블 인코딩 예
(source: Medium)

One-hot Encoding

- **원핫 인코딩**: 개별 범주를 특정 이진 벡터(binary vector)로 표현
  - 범주 별로 인덱스를 할당
    - 범주에 해당되는 인덱스의 값: 1
    - 그 외의 나머지 값: 0
  - 명목형 데이터 인코딩에 적합

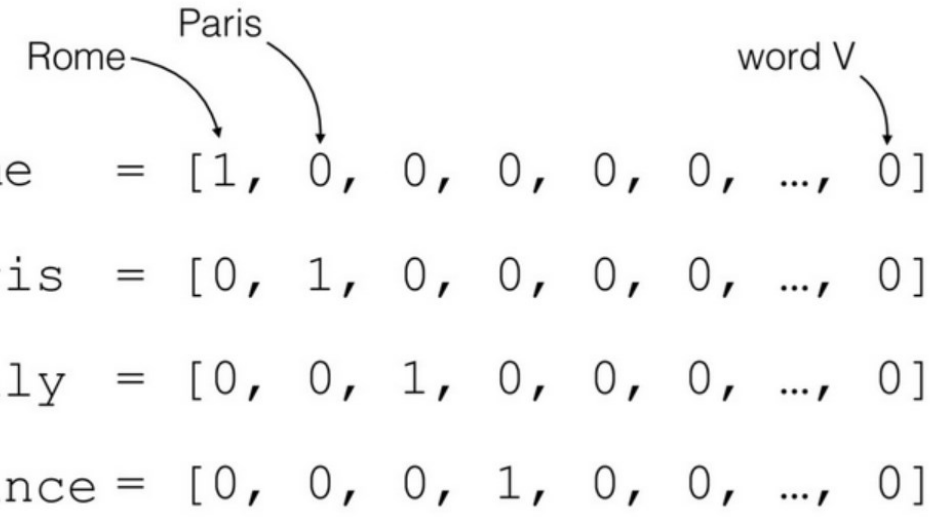


그림. 원핫 인코딩 예: 워드 임베딩
(source: brunch.co.kr)

Label Encoding and One-hot Encoding

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Source: Medium

Exercise: One-Hot Encoding

- 예제: 원핫 인코딩

```
df_emp_encoded = pd.get_dummies(df_emp, columns=['gender', 'remarks'])
```

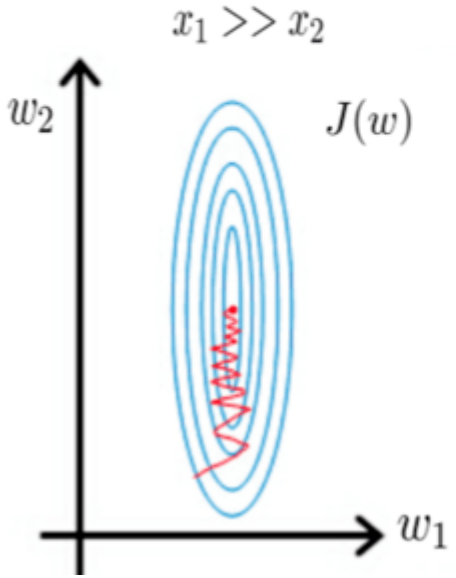
	emp_id	gender	remarks
0	1	Male	Nice
1	2	Female	Good
2	3	Female	Great
3	4	Male	Great
4	5	Female	Nice

	emp_id	gender_Female	gender_Male	remarks_Good	remarks_Great	remarks_Nice
0	1	0	1	0	0	1
1	2	1	0	1	0	0
2	3	1	0	0	1	0
3	4	0	1	0	1	0
4	5	1	0	0	0	1

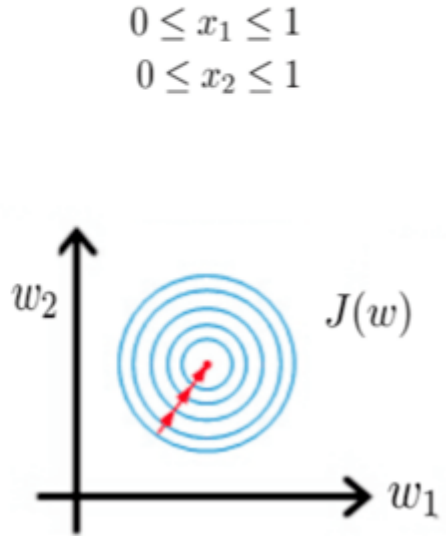
Scaling

- 일반적으로 변수들은 서로 다른 **값 범위**(스케일)를 가짐
  - 예: 나이와 연소득
  - 모델은 기본적으로 변수 별 스케일을 고려하지 않음 ⇨ 모델 학습 수렴 및 과적합 문제

Gradient descent without scaling



Gradient descent after scaling variables

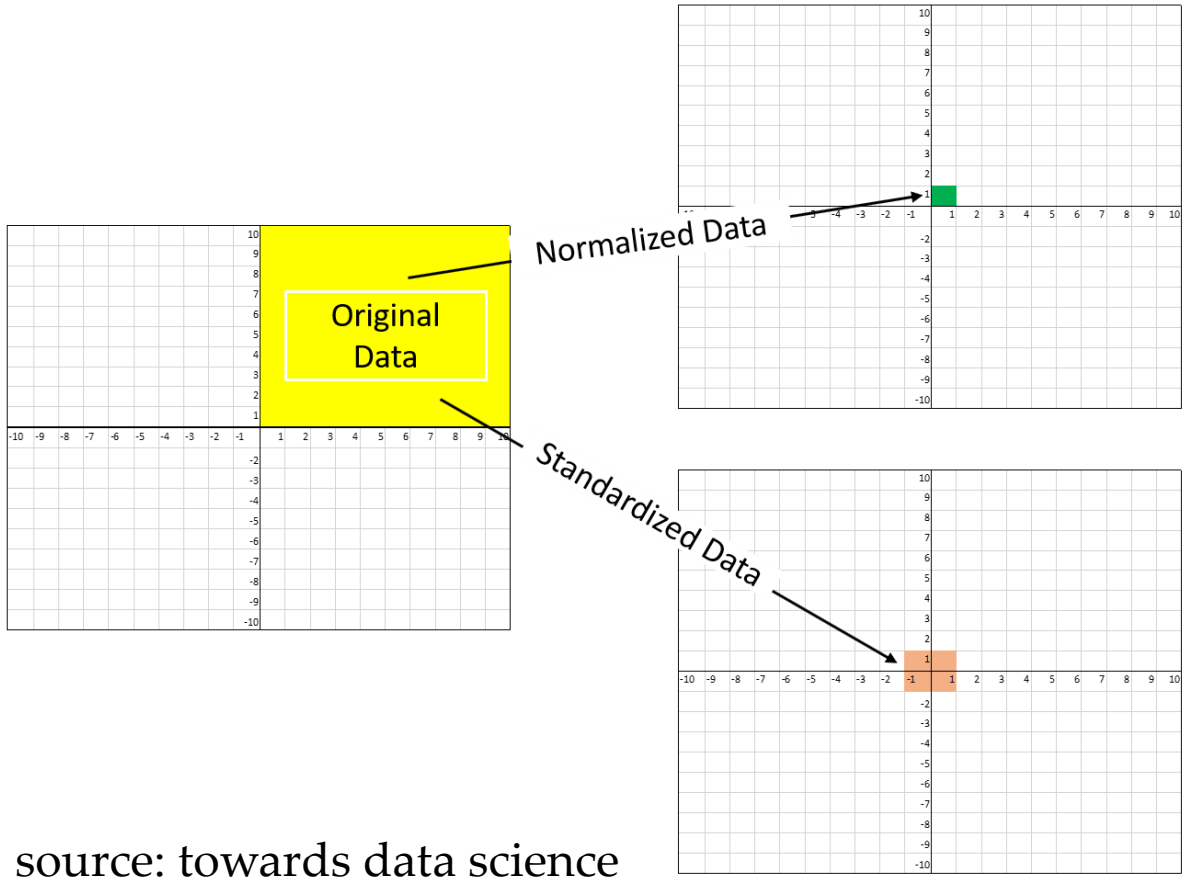


많은 머신러닝 모델/알고리즘들이 스케일링된 데이터에서 더 잘 동작함

source: towards data science

Scaling

- **스케일링**: 모든 변수들이 비슷한 값 범위를 갖도록 변환하는 작업
  - **정규화**(normalization)
  - **표준화**(standardization)



source: towards data science

Scaling: Normalization

- **정규화**: 모든 변수들을 **0~1** 사이의 값으로 스케일링
  - 대표적 방법: Min-Max Normalization(최소-최대 정규화)

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - 변수의 분포에는 변화가 없으므로, 이상치의 영향은 그대로 유지

⇨ 정규화 전에 이상치 처리를 먼저 수행하는 것이 권장됨

1. 이상치 제거 없이 스케일링한 결과

X={0,2,5,10,15,20,22,24,990,1000}

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$x_{max} = 1000$ $x_{min} = 0$

X_{norm}={0, 0.002, 0.005, 0.01, 0.015, 0.02, 0.022, 0.024, 0.99, 1}

2. 이상치 제거 후 스케일링한 결과

X={0,2,5,10,15,20,22,24,990,1000}

$x_{max} = 24$ $x_{min} = 0$

X_{norm}={0, 0.083, 0.208, 0.417, 0.625, 0.833, 0.917, 1}

Scaling: Standardization

- **표준화**(또는 z-score 정규화): **표준편차**를 기반으로 스케일링 수행
  - 데이터가 정규분포를 따른다고 가정하고,

    분포가 0을 중심으로 하고 표준편차가 1이 되도록 스케일링 수행

$$z = \frac{x - \mu}{\sigma}$$

  - 변수마다 표준편차가 다를 경우 스케일링 결과(값 범위)가 다를 수 있음
  - 이상치에 영향을 덜 받는 스케일링 방법

Scaling: Others

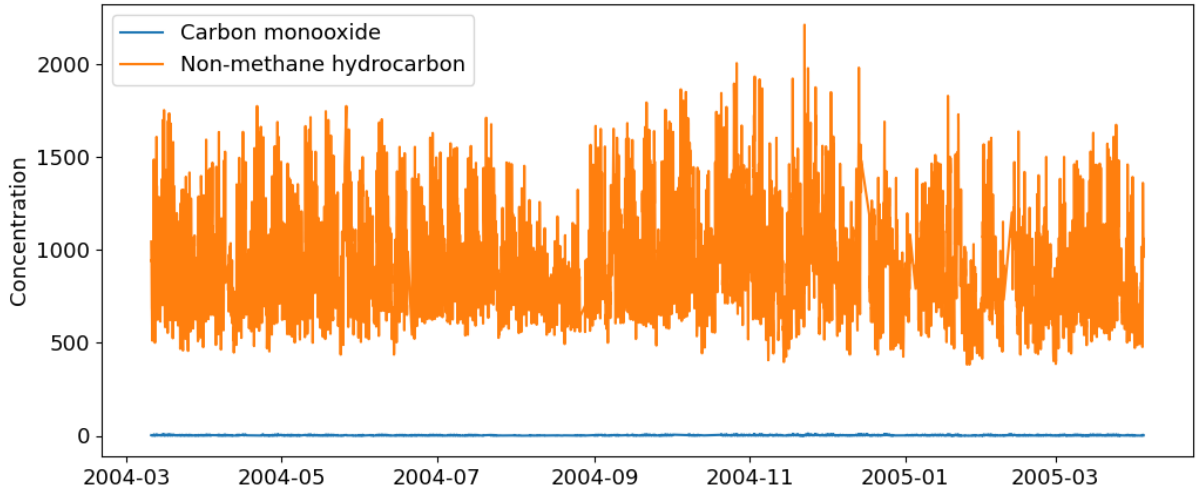
- 그 외의 스케일링 기법 (link)
  - Max Abs Scaler
  - Robust Scaler
  - Quantile Transformer Scaler
  - Power Transformer Scaler
  - Unit Vector Scaler

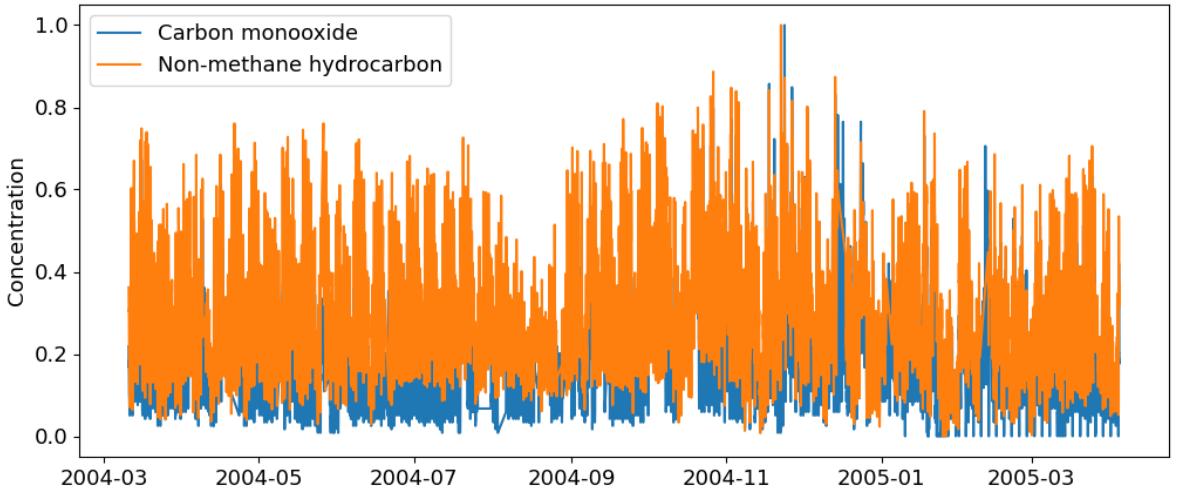
Exercise: Min-Max Normalization

- 예제: Air Quality 일산화탄소/비메탄 탄화수소의 최소-최대 정규화

```
co_max = co.max()
co_min = co.min()

df['CO_Norm'] = (co - co_min) / (co_max - co_min)
df['CO_Norm']
```

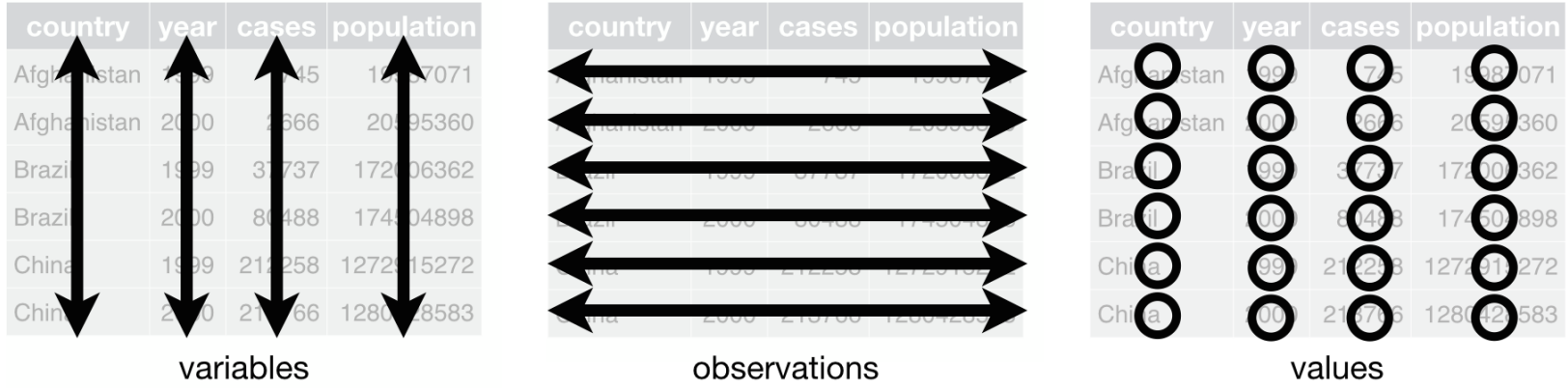
```
nmhc_max = nmhc.max()
nmhc_min = nmhc.min()

df['NMHC_Norm'] = (nmhc - nmhc_min) / (nmhc_max - nmhc_min)
df['NMHC_Norm']
```





Grouping Operations

- **깔끔한 데이터**(tidy data): 데이터 분석 / 머신러닝에 적합한 데이터 형태



source: biocorecrg.github.io

day	month	year	weight	height
12	4	2020	3.5	48
23	8	2019	2.9	50
9	11	2020	3.8	50

tidy data 예:

day	month,year	weight	height
12	4,2020	3.5kg	48
23	8,2019	2.9kg	50
9	11,2020	3.8kg	50

untidy data 예:

Grouping Operations: Pivot Table

- **피벗 테이블**: 개별 데이터 항목들의 집계 및 테이블 재구조화를 통해 데이터의 요약된(그룹화된) 결과를 나타내는 테이블
  - 트랜잭션 데이터(예: 계좌이체 내역)를 tidy 형태의 데이터로 변환할 때 활용

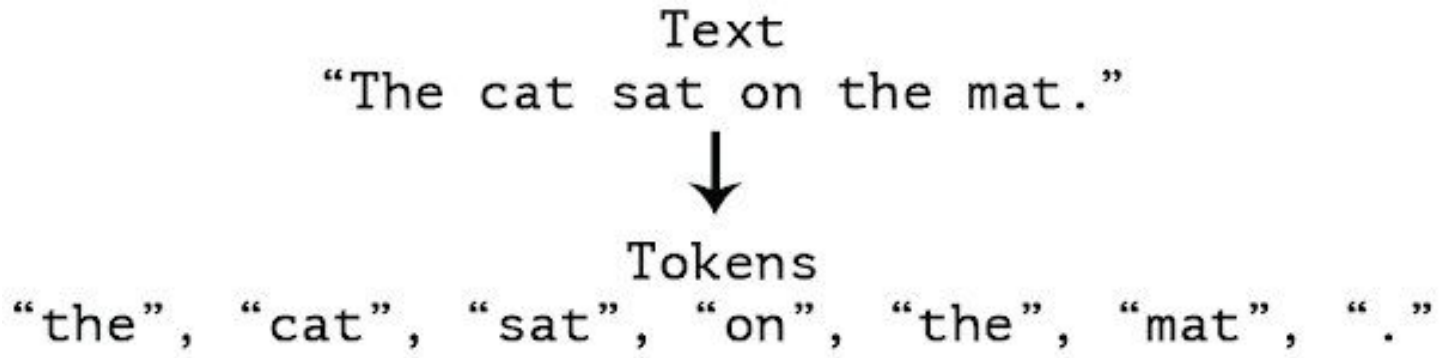
User	City	Visit Days
1	Roma	1
2	Madrid	2
1	Madrid	1
3	Istanbul	1
2	Istanbul	4
1	Istanbul	3
1	Roma	3

User	Istanbul	Madrid	Roma
1	3	1	4
2	4	2	0
3	1	0	0

source: towards data science

Feature Split

- **특징 분할**: 복합적인 값으로 구성된 특징 값을 여러 개의 값으로 분할
  - 예: 정제되지 않은 문자열의 토크나이징(tokenizing)



Source: Manning

Exercise: Feature Split

- 예제: `split()` 를 이용한 영화 데이터의 특징 분할

```
"The Godfather, 1972, Francis Ford Coppola“
"Contact, 1997, Robert Zemeckis“
"Parasite, 2019, Joon-ho Bong"
```

```
title, year, director = val.split(',')
```

	title	year	director
0	The Godfather	1972	Francis Ford Coppola
1	Contact	1997	Robert Zemeckis
2	Parasite	2019	Joon-ho Bong

References

- 꿈쟁이, **구간화란?**, 네이버블로그 (link)
- 오일석, **패턴인식**, 교보문고
- 오일석, **패턴인식: 8장. 특징 추출**, slideplayer (link)
- Roy, B., **All about Feature Scaling**, towards data science (link)
- Dey, V., **Common Feature Engineering Techniques To Tackle Real-World Data**, Analytics India Magazine (link)
- Komorowski, M., et al., "**Exploratory Data Analysis**," Secondary Analysis of Electronic Health Records, Springer Nature, 2016
- Grabiński, P., **Feature Engineering for Machine Learning: 10 Examples**, KD nuggets (link)
- Karbhari, V., **Feature engineering in machine learning**, Medium (link)
- Rençberoğlu, E., **Fundamental Techniques of Feature Engineering for Machine Learning**, towards data science (link)
- Desarda, A., **Getting Data ready for modelling: Feature engineering, Feature Selection, Dimension Reduction (Part 1)**, towards data science (link)
- Desarda, A., **Getting Data ready for modelling: Feature engineering, Feature Selection, Dimension Reduction (Part 2)**, towards data science (link)
- YAĞCI, H. E., **Label Encoding vs One Hot Encoding**, Medium (link)
- Yuan, J., et al., "**Machine Learning Applications on Neuroimaging for Diagnosis and Prognosis of Epilepsy: A Review**," arXiv.org (link)
- Moffitt, C., **Pandas Pivot Table Explained**, Practical Business Python (link)
- **Representation: Feature Engineering**, developers.google.com (link)
- biocorecrg, **What is "tidy" data?**, GitHub (link)

수고하셨습니다.