House Price Prediction through Logistic Regression

발표일 : 2020.01.27

발표자 : 조소영

목 차

1. 데이터 탐색 (EDA)
    1.1 문제 정의
    1.2 변수 설명
    1.3 타겟변수 시각화
    1.4 변수 간 관계 파악 (correlation)
    1.5 각 변수의 분포 시각화
2. 데이터 전처리
    2.1 아웃라이어 제거
    2.2 변수 정규화 및 시각화
3. 변수 수정
4. Logistic Regression

1.1 문제 정의

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data = pd.read_csv("Real estate.csv")
```

In [3]:
```
print(data.shape)
```

```
(414, 8)
```

In [4]:
```
data.isnull().sum() #checking for total null values
```

Out[4]:
```
No                                      0
X1 transaction date                     0
X2 house age                            0
X3 distance to the nearest MRT station  0
X4 number of convenience stores         0
X5 latitude                             0
X6 longitude                            0
Y house price of unit area              0
dtype: int64
```

데이터의 전체 개수는 414개이며, 변수(feature)의 개수는 7개이다. 결측값(null data)은 없다.

1.2 변수 설명

```
In [6]: data.head()
```

Out[6]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	1	2012.917	32.0	84.87882	10	24.98298	121.54024	37.9
1	2	2012.917	19.5	306.59470	9	24.98034	121.53951	42.2
2	3	2013.583	13.3	561.98450	5	24.98746	121.54391	47.3
3	4	2013.500	13.3	561.98450	5	24.98746	121.54391	54.8
4	5	2012.833	5.0	390.56840	5	24.97937	121.54245	43.1

- No : 집을 구분하는 번호 (index)
- X1 transaction date : 집 거래 날짜
- X2 house age : 집이 얼마나 오래 되었는지 (지은 뒤 흐른 시간)
- X3 distance to the nearest MRT station : 가장 가까운 대중교통까지의 거리 (Mass Rapid Transit)
- X4 number of convenience stores : 근처 편의점 개수
- X5 latitude : 경도
- X6 longitude : 위도
- Y house price of unit area : 집의 가격 (Target Variable)

1.3 타겟 변수 시각화

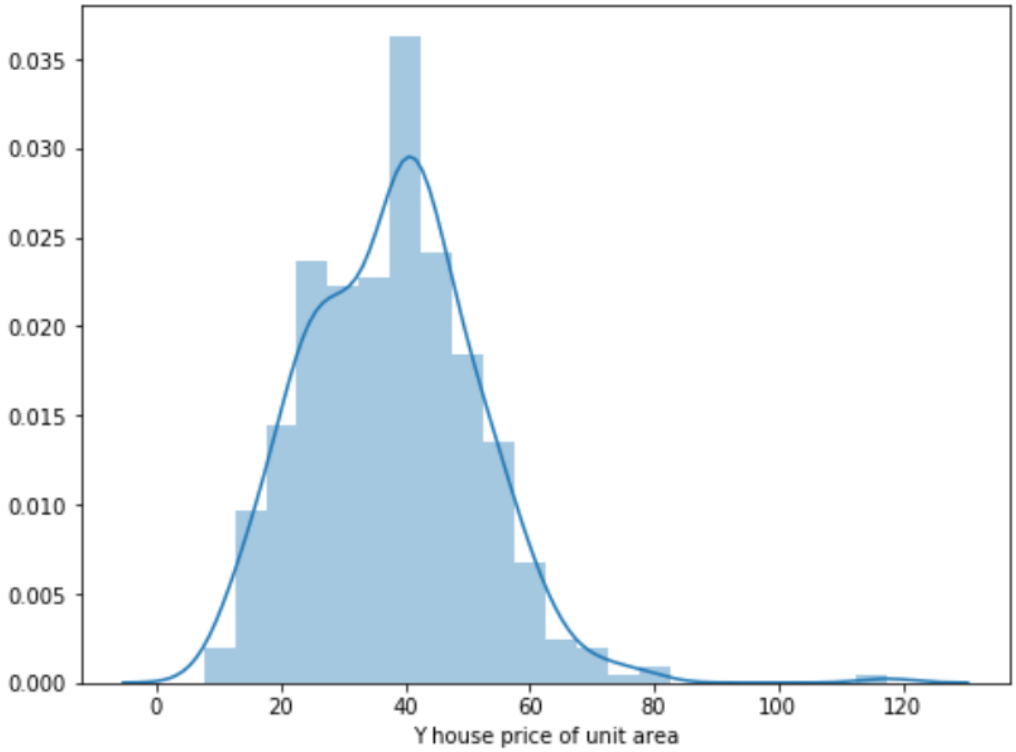
```
In [7]: # descriptive statistics summary (기술적 통계)
        data['Y house price of unit area'].describe()
```

```
Out[7]: count    414.000000
        mean      37.980193
        std       13.606488
        min        7.600000
        25%       27.700000
        50%       38.450000
        75%       46.600000
        max      117.500000
        Name: Y house price of unit area, dtype: float64
```

target variable인 집의 가격에 대한 통계 summary는 위와 같다.

가장 싼 집(min)은 7.6, 가장 비싼 집(max)은 117.50이며 평균가는 약 38이다.

1.3 타겟 변수 시각화



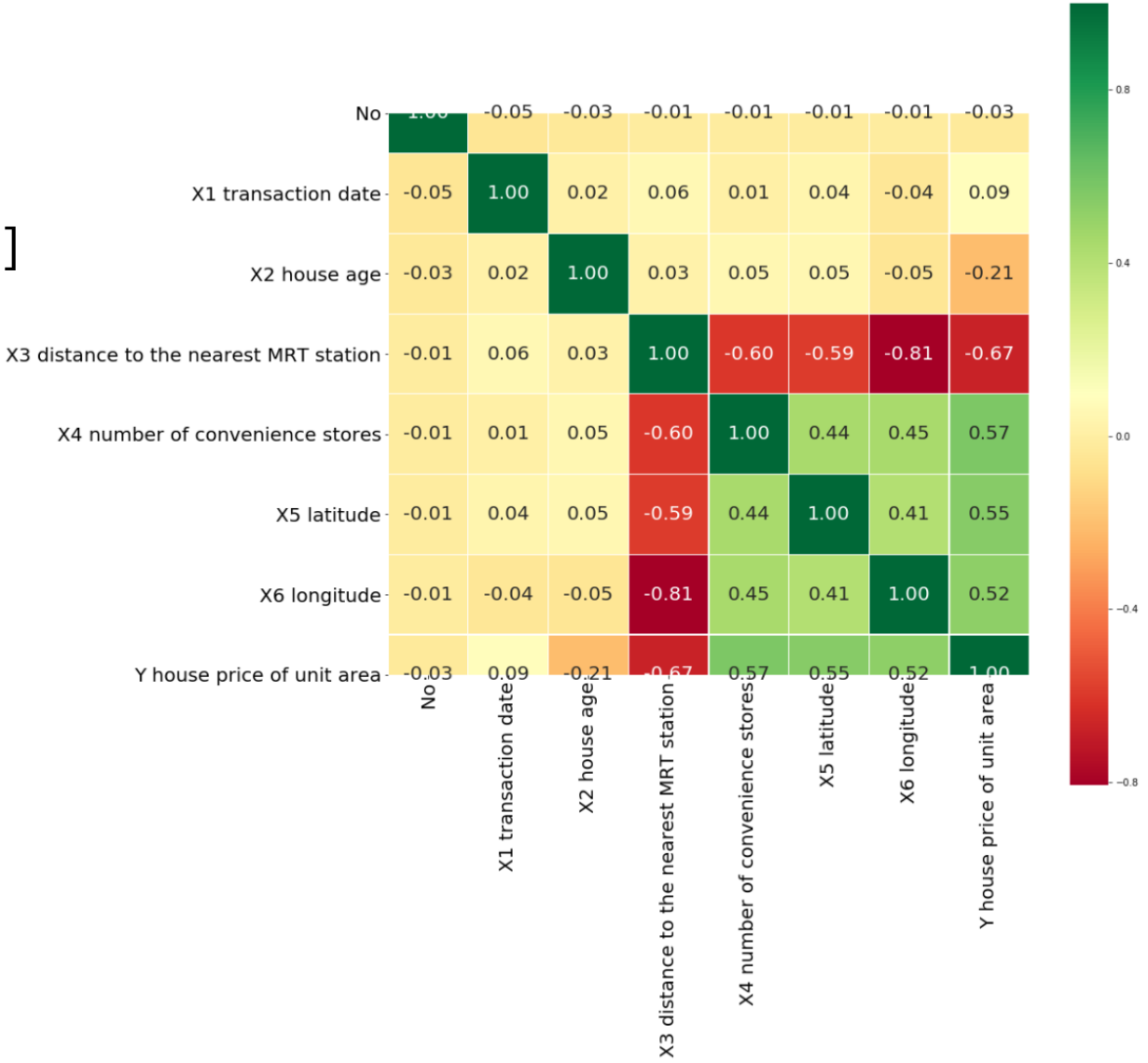
target variable인 집값의 분포를 히스토그램을 통해 확인해보았다.

가장 비싼 집 117.5 는 아웃라이어로 보인다.

1.4 변수 간 관계 파악 (correlation)

[Y와 상관도가 높은 변수]

1. X3 : - 0.67 (음의 상관관계)
2. X4 : 0.57 (양의 상관관계)
3. X5 : 0.55 (양의 상관관계)
4. X6 : 0.52 (양의 상관관계)

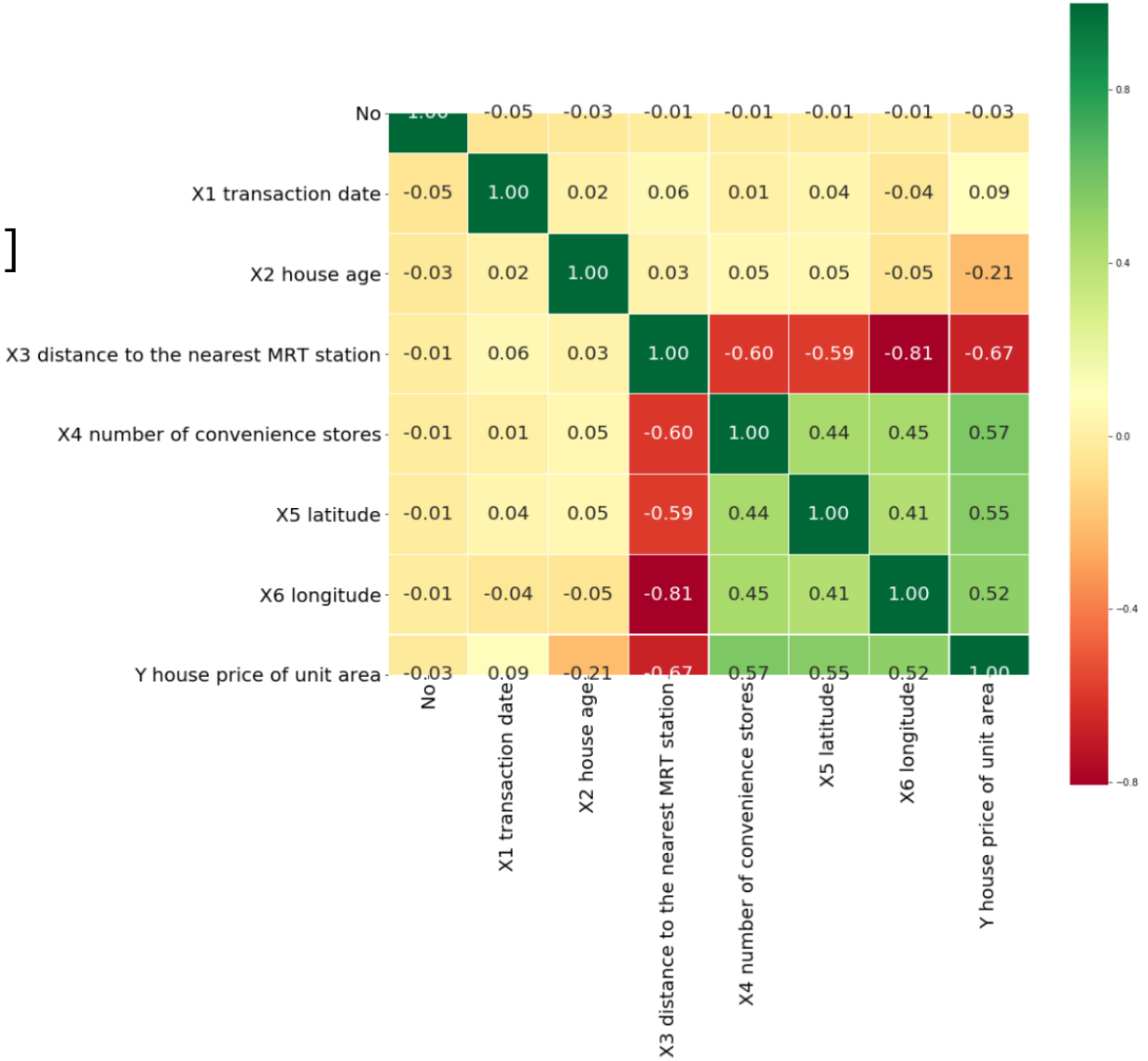


1.4 변수 간 관계 파악 (correlation)

[Y와 상관도가 낮은 변수]

1. X1 : 0.09

2. X2 : -0.21



1.4 변수 간 관계 파악 (correlation)

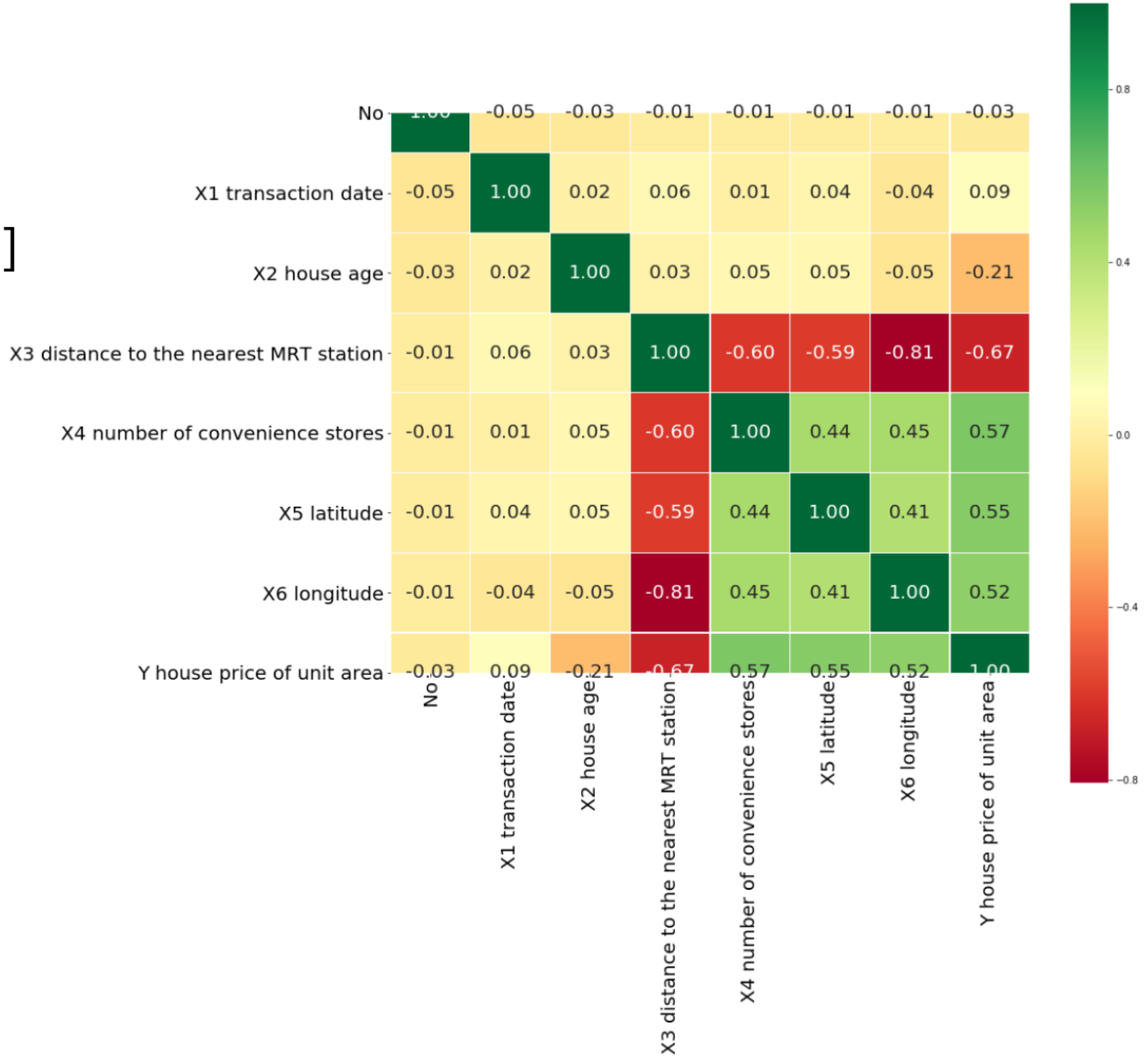
[높은 상관관계의 변수들]

1. X3 & X6 : -0.81

(높은 음의 상관관계)

2. X3 & X4 : -0.60

3. X3 & X5 : -0.59



1.4 변수 간 관계 파악 (correlation)

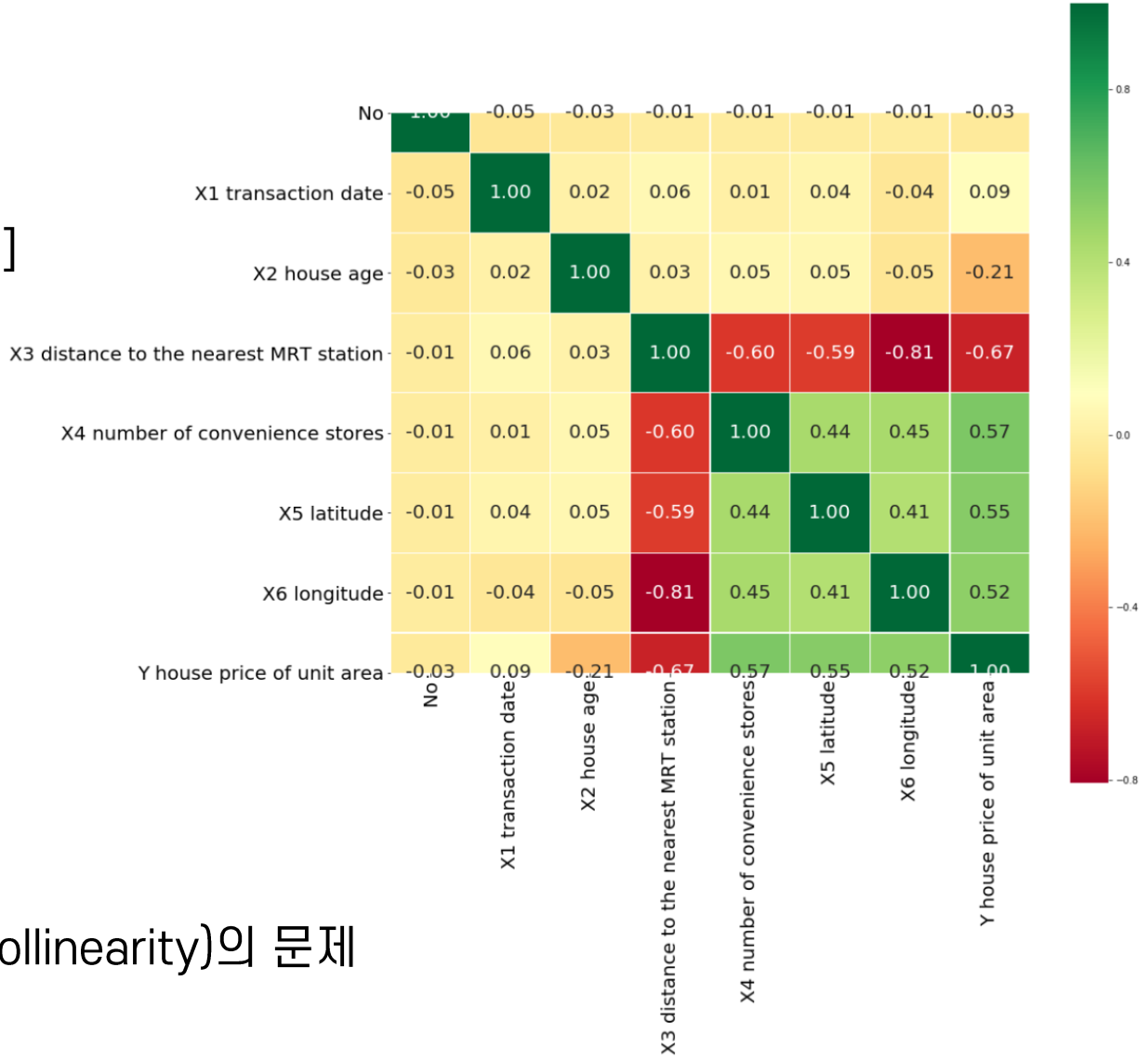
[높은 상관관계의 변수들]

1. X3 & X6 : -0.81

(높은 음의 상관관계)

2. X3 & X4 : -0.60

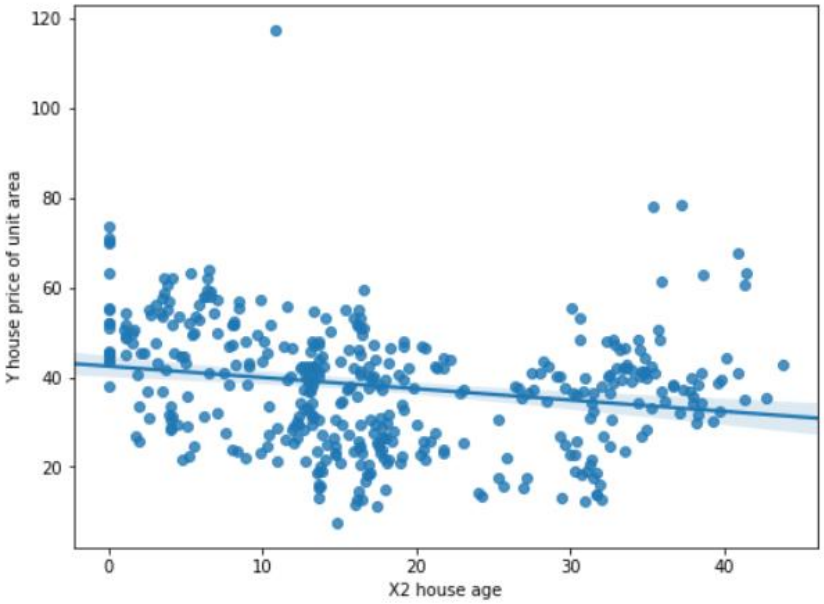
3. X3 & X5 : -0.59

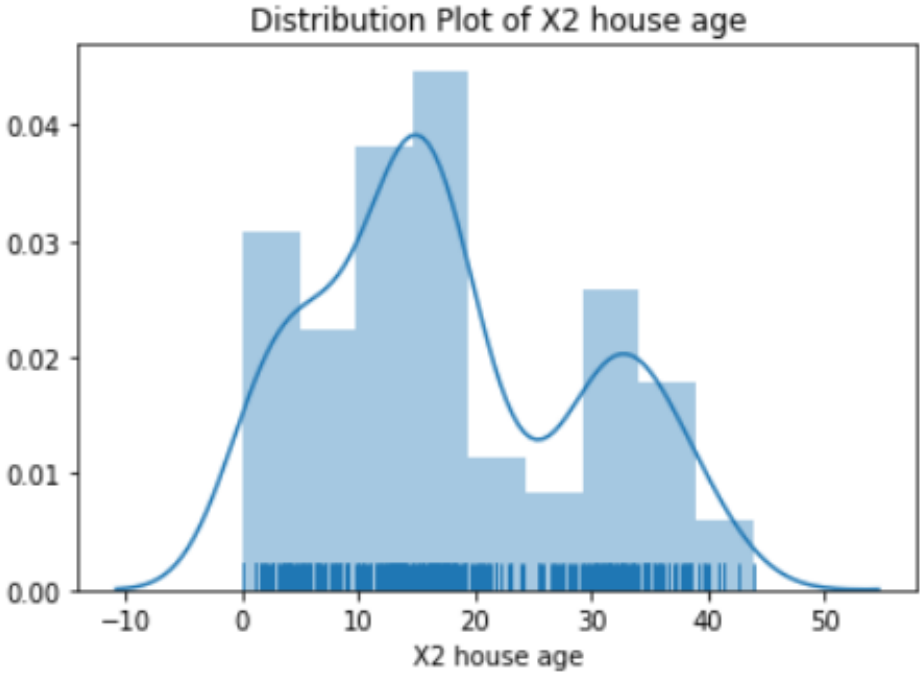


➡ 다중공선성 (multicollinearity)의 문제

1.5 각 변수의 분포 시각화

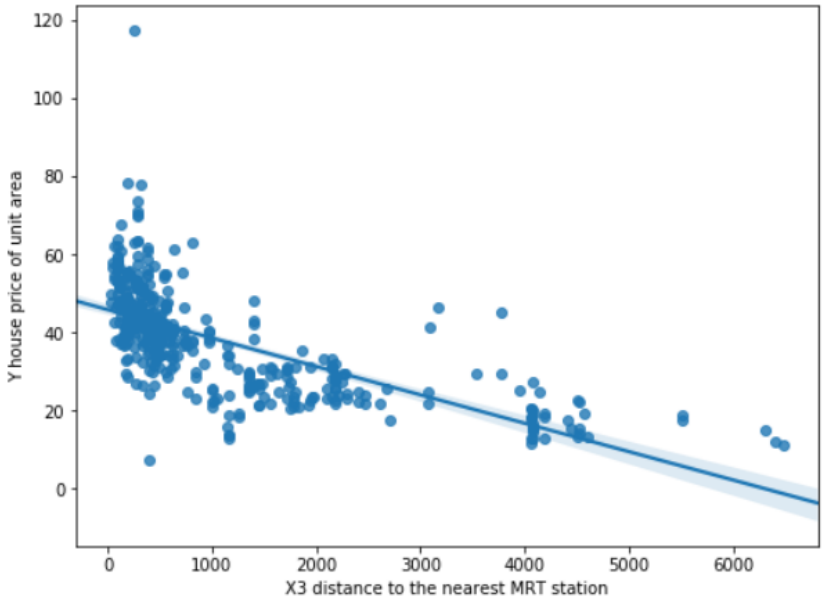
X2 House Age

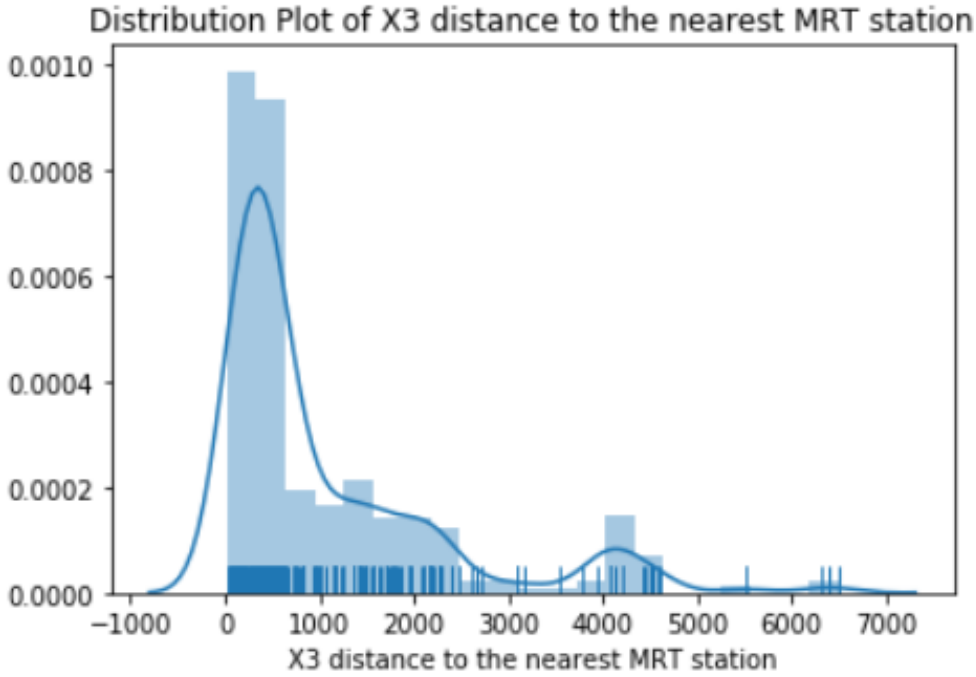




1.5 각 변수의 분포 시각화

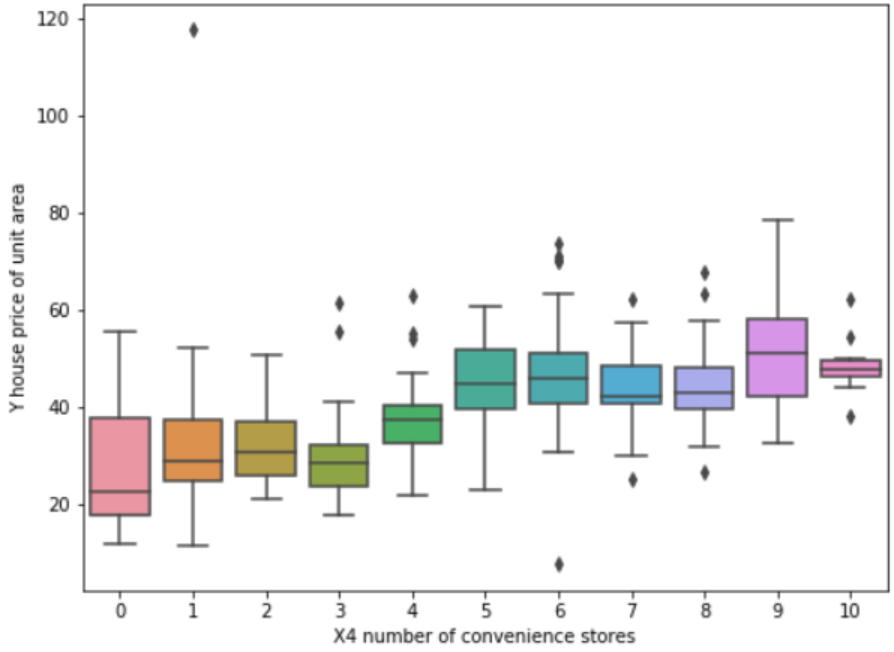
X3 Distance to the nearest MRT Station

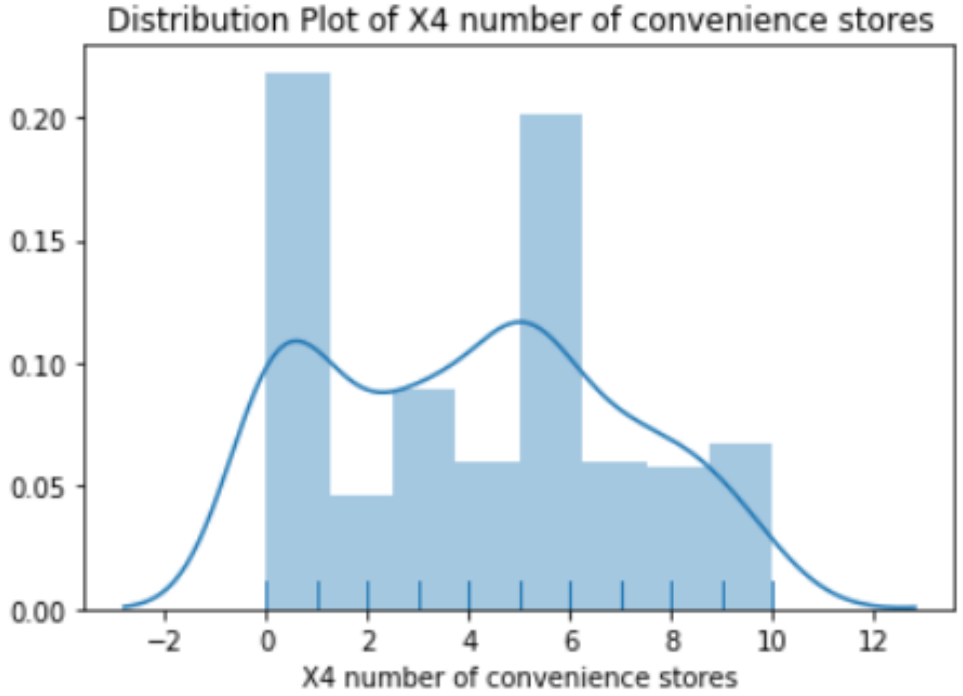




1.5 각 변수의 분포 시각화

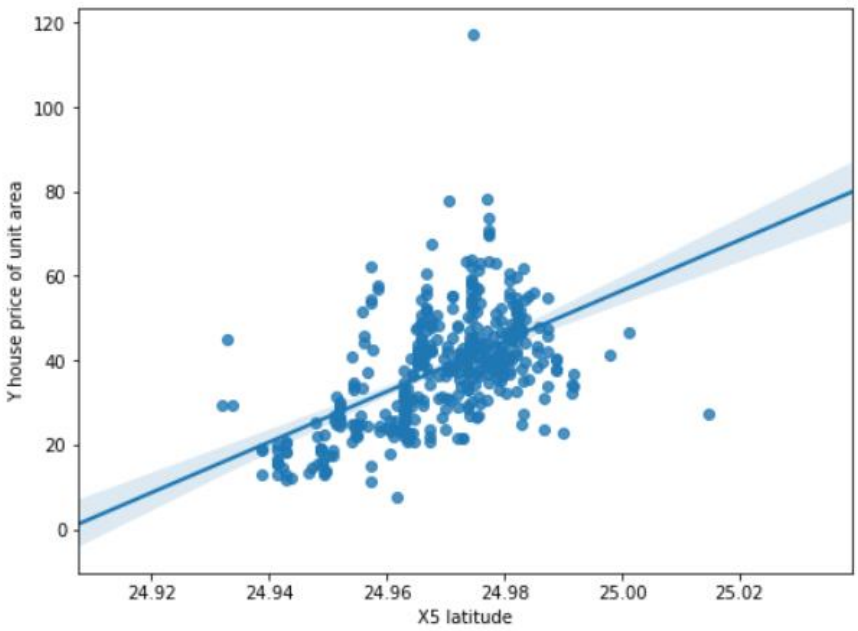
X4 Number of Convenience Stores

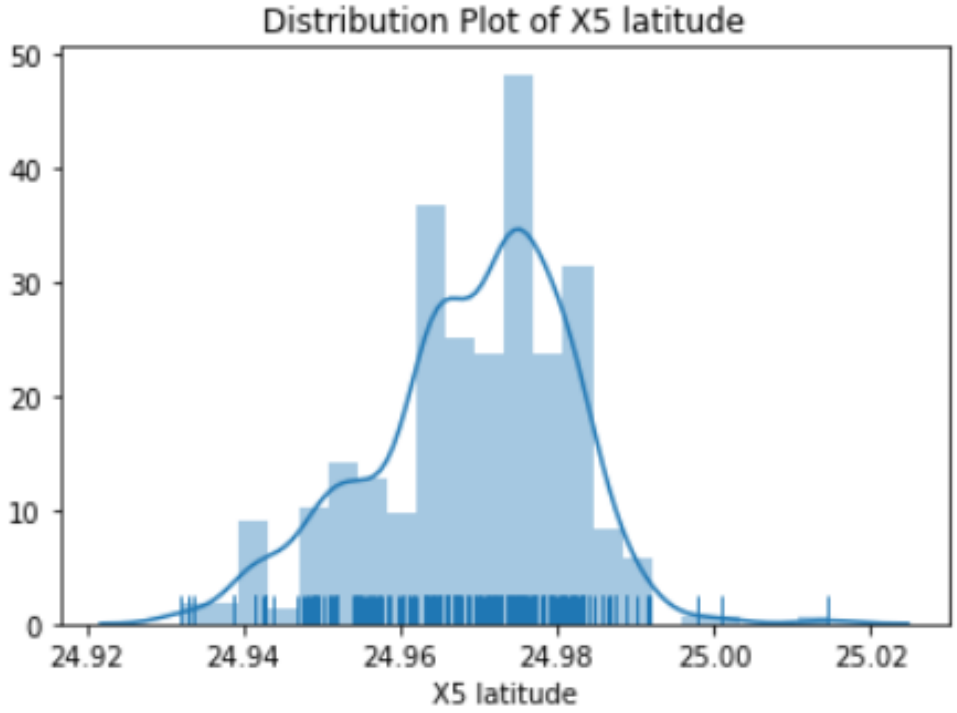




1.5 각 변수의 분포 시각화

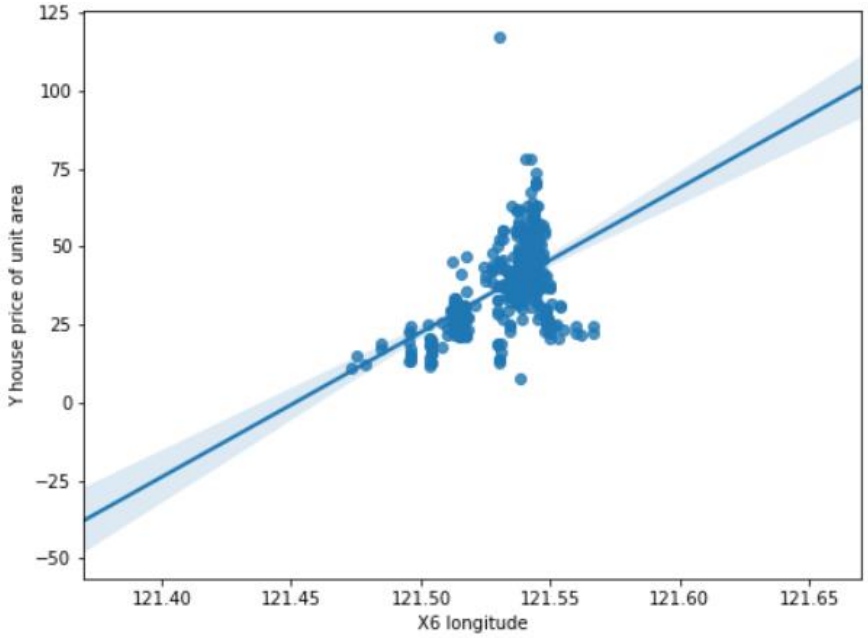
X5 Latitude

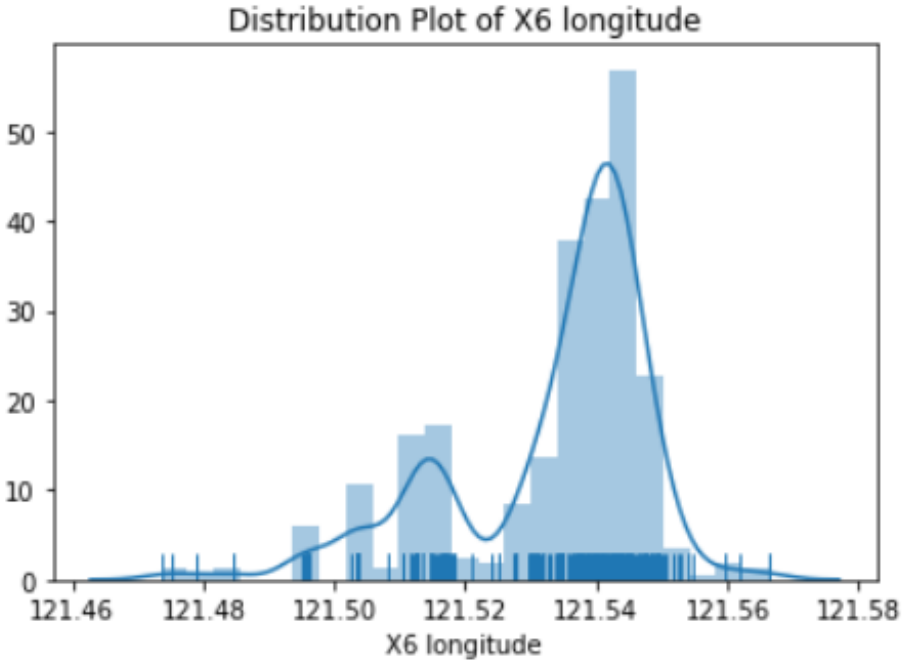




1.5 각 변수의 분포 시각화

X6 Longitude





1.5 각 변수의 분포 시각화

문제점

1. 너무 기울어져 있거나 쏠려 있다.
2. 변수 간 데이터 단위의 차이가 크다.

예) 변수 X3 데이터 범위 : 약 -1000 ~ 7000 (천 단위)
변수 X2 데이터 범위 : 0 ~ 45 (십의 자리 단위)

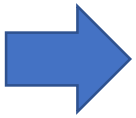
이렇게 변수 간 데이터의 단위 차이가 클 경우,
각 theta값 (파라미터) 학습이 속도가 달라져 원하는 결과가 나오지 못하게 될 수 있다.

1.5 각 변수의 분포 시각화

정규화의 필요성

```
No 의 mean :  207.5
No 의 std :  119.6557562342907


X1 transaction date 의 mean :  2013.1489710144933
X1 transaction date 의 std :  0.281967240262992


X2 house age 의 mean :  17.71256038647343
X2 house age 의 std :  11.392484533242536


X3 distance to the nearest MRT station 의 mean :  1083.8856889130436
X3 distance to the nearest MRT station 의 std :  1262.1095954078514


X4 number of convenience stores 의 mean :  4.094202898550725
X4 number of convenience stores 의 std :  2.945561805663617


X5 latitude 의 mean :  24.969030072463745
X5 latitude 의 std :  0.012410196590450125


X6 longitude 의 mean :  121.53336108695667
X6 longitude 의 std :  0.015347183004590918


Y house price of unit area 의 mean :  37.98019323671498
Y house price of unit area 의 std :  13.606487697735314
```

각 변수의 평균과 표준편차



표준화 공식 $= \frac{X-m}{\sigma}$

2.1 아웃라이어 제거

In [21]:
```
data.loc[data['Y house price of unit area'] > 100]
```

Out[21]:

	No	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
270	271	2013.333	10.8	252.5822	1	24.9746	121.53046	117.5

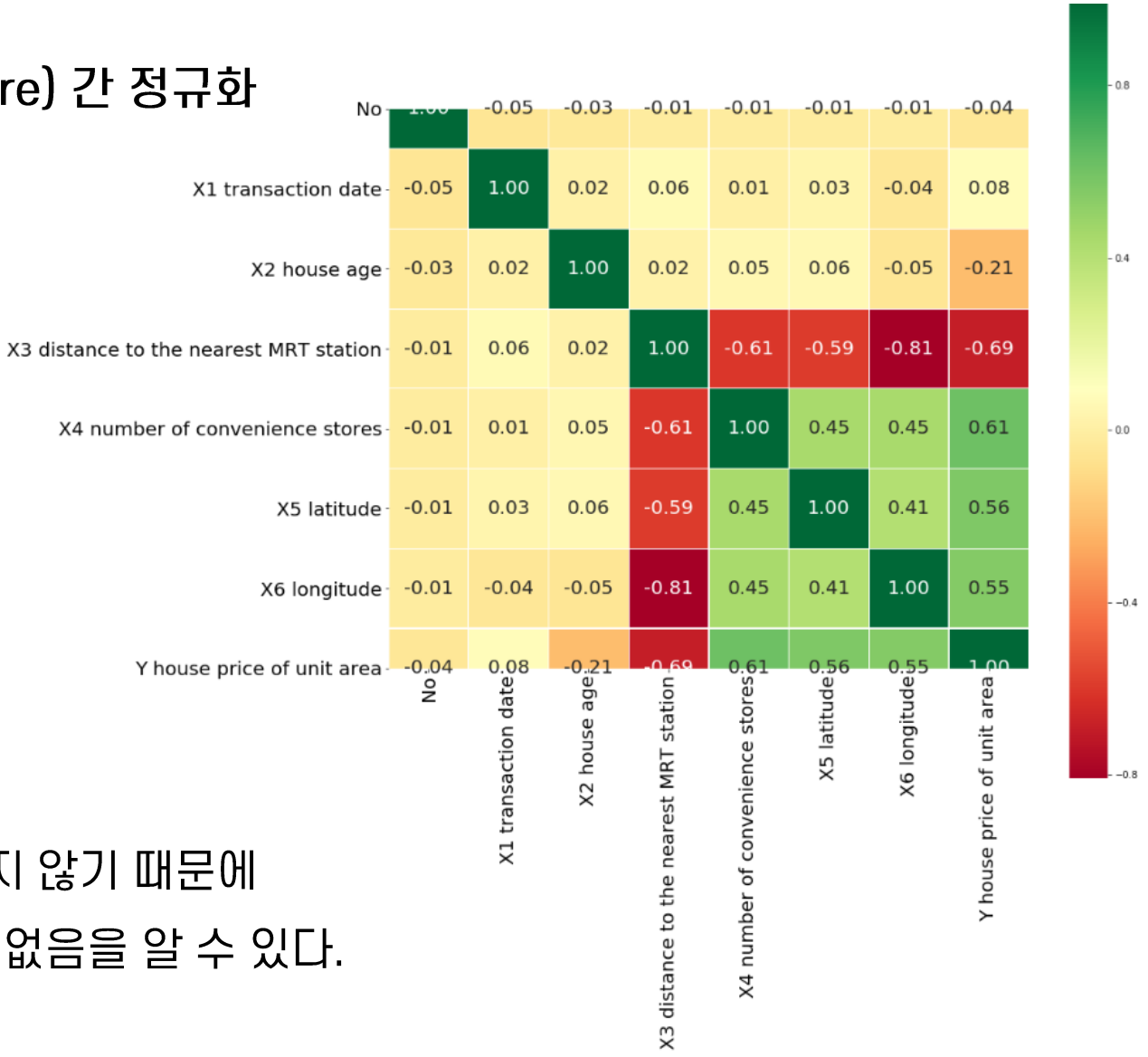
In [22]:
```
data = data.loc[data['No']!=271]
```

각 변수의 분포 그래프를 그려보았을 때,

집 값이 유난히 높았던 outlier를 제거한다.

2.2 변수 정규화

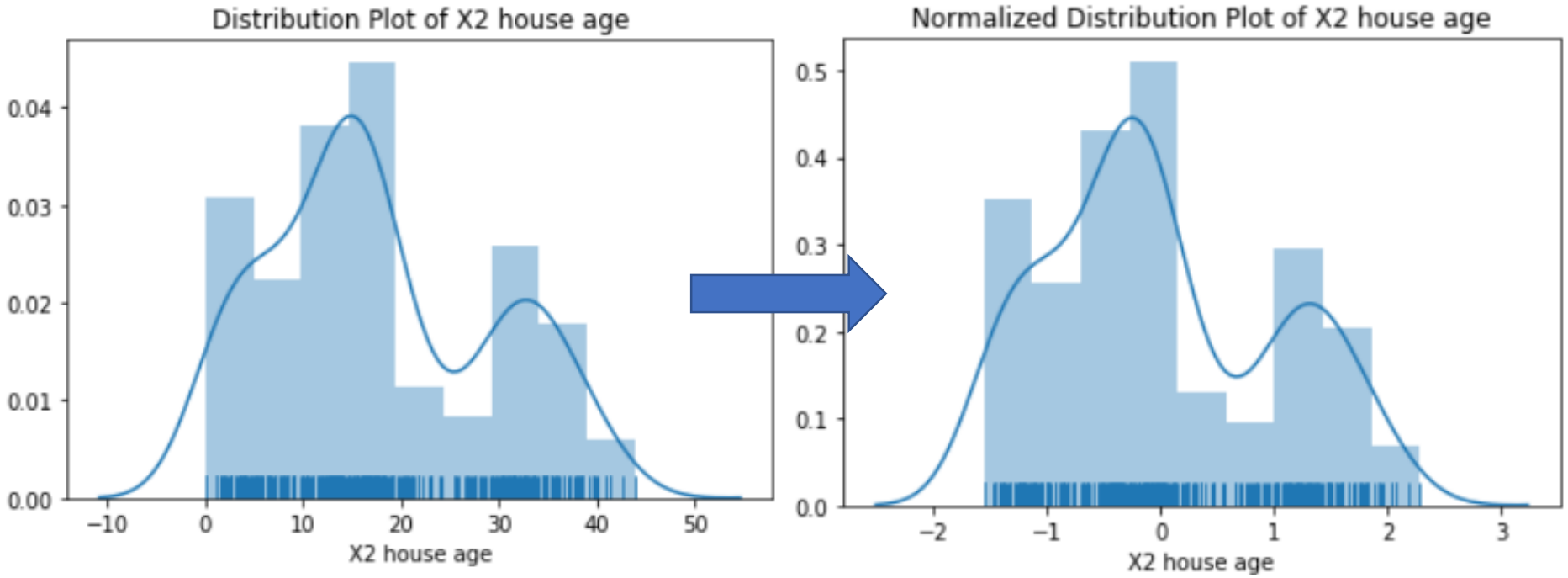
정규화 후 변수(feature) 간 정규화



데이터의 성격은 변하지 않기 때문에
정규화 전후에 거의 차이가 없음을 알 수 있다.

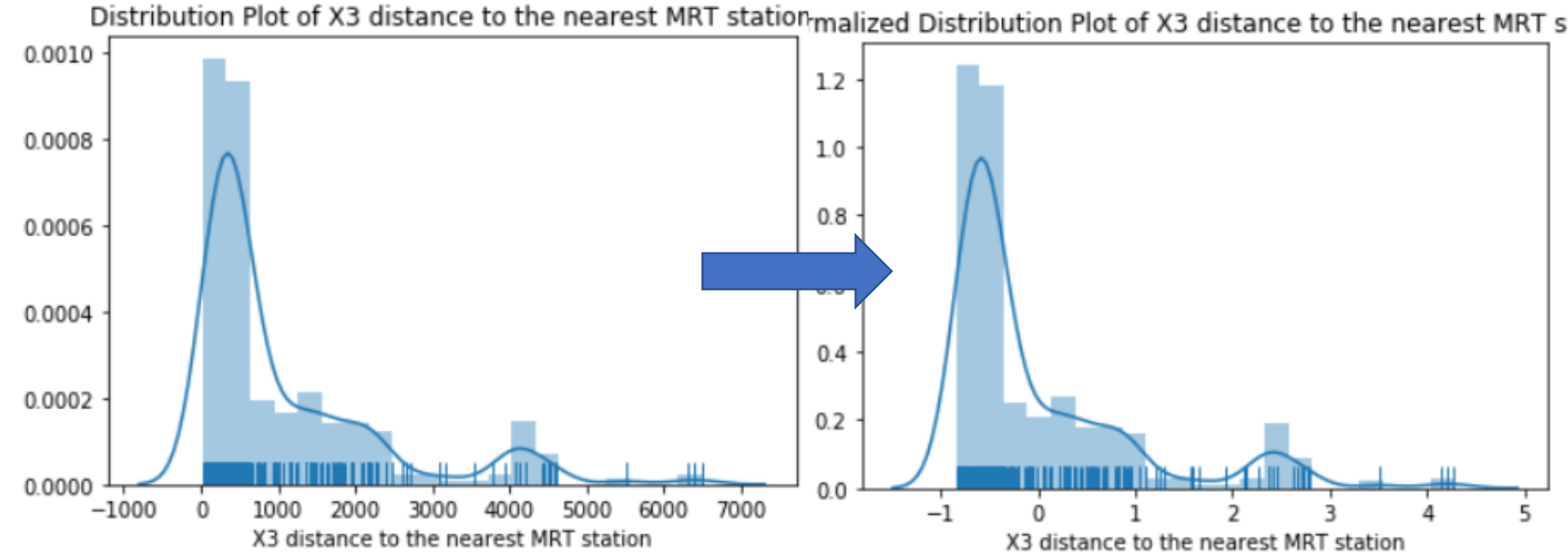
2.2 변수 정규화 후 분포 시각화

Normalized X2 House Age



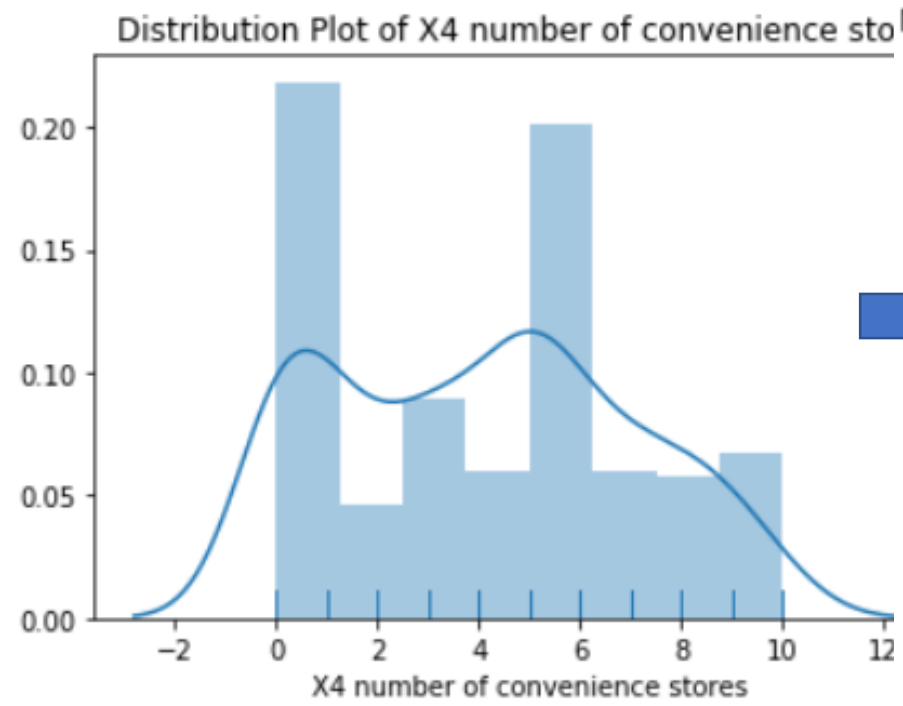
2.2 변수 정규화 후 분포 시각화

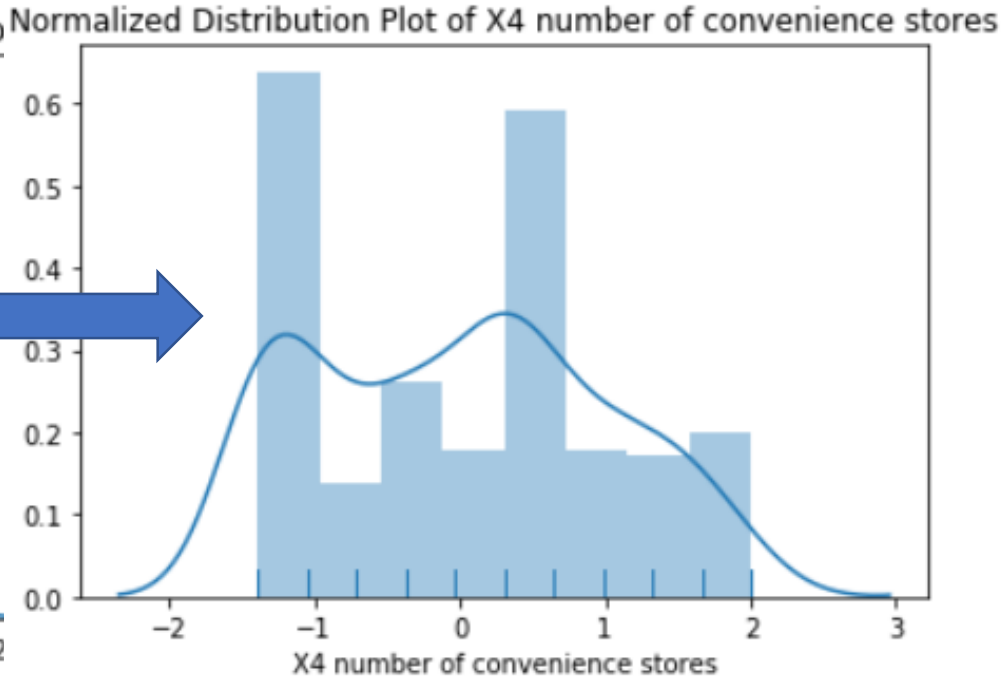
Normalized X3 Distance to the nearest MRT Station



2.2 변수 정규화 후 분포 시각화

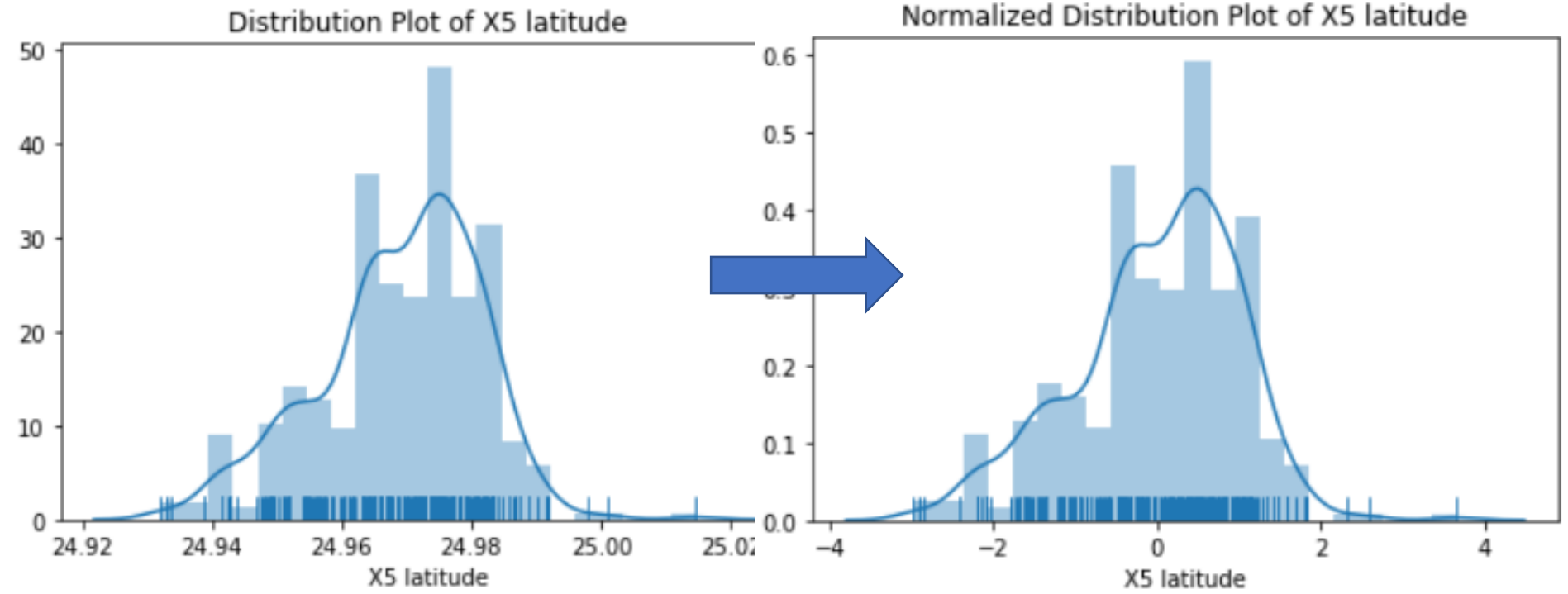
Normalized X4 Number of Convenience Stores





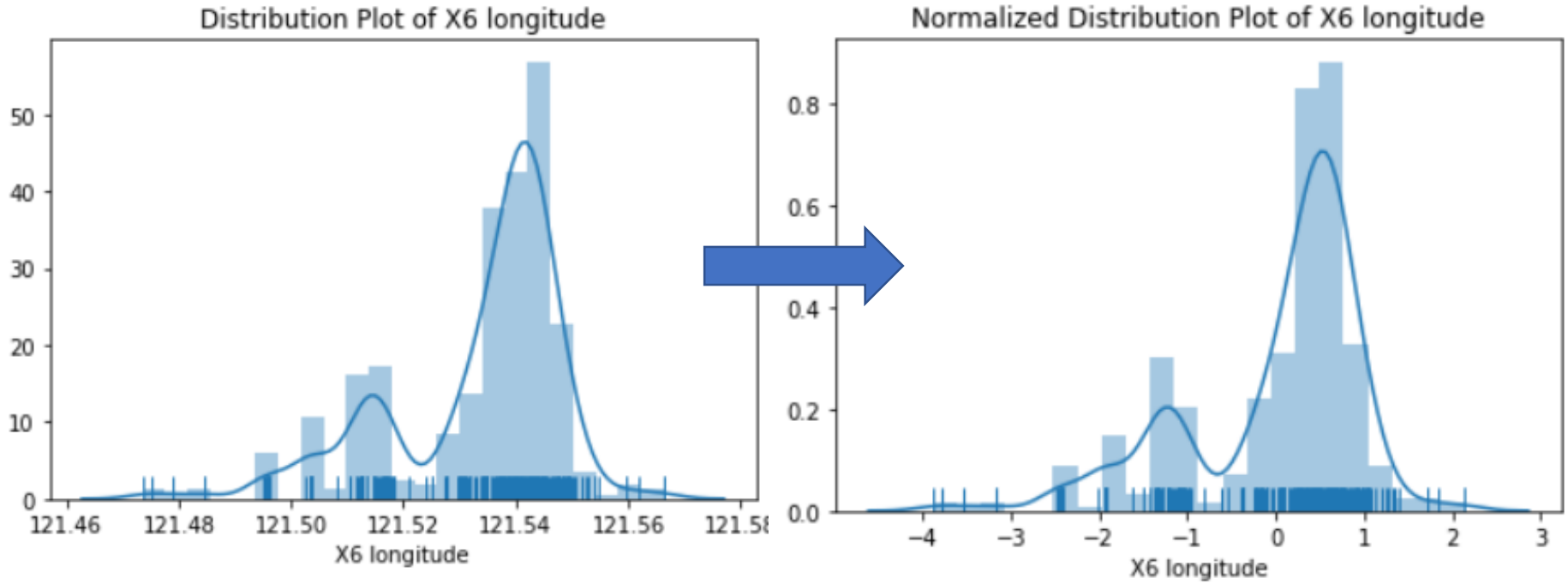
2.2 변수 정규화 후 분포 시각화

Normalized X5 Latitude



2.2 변수 정규화 후 분포 시각화

Normalized X6 Longitude



2.2. 변수 정규화 후 분포 시각화

변수 정규화 결과

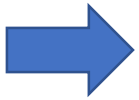
각 변수 데이터의 평균과 표준편차가 맞춰져 데이터의 단위도 일정하게 정규화

➔ 각 변수 별 theta값(파라미터) 학습이 비슷한 속도로 진행될 수 있게 되었다.

3. 변수 수정

불필요한 변수가 지나치게 많을 경우

-> 오버피팅



집 값(Y)과 상관관계가 낮은 변수

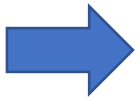
X1 transaction date
X2 house date

변수들 간 상관관계가 높을 경우,

타겟 변수를 독립적으로 예측하지

못해 모델의 유의성을 떨어뜨리는

다중공선성의 문제



다른 변수들 간 상관관계가 높은 변수

X3

➔ 더 높은 집값 예측 정확도와 높은 모델의 신뢰성을 위해
No, X1, X2, X3 변수 삭제

3. 변수 수정

In [39]:
```
data.head()
```

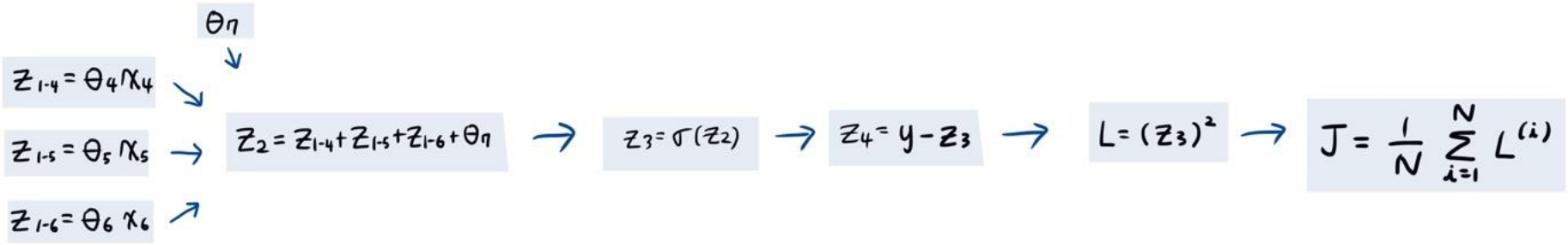
Out[39]:

	X4 number of convenience stores	X5 latitude	X6 longitude	Y house price of unit area
0	2.002696	1.124068	0.447239	0.008612
1	1.663159	0.911546	0.399729	0.338212
2	0.305008	1.484713	0.686092	0.729134
3	0.305008	1.484713	0.686092	1.304018
4	0.305008	0.833460	0.591071	0.407198

➔ 총 3개의 독립변수(X), 1개의 종속변수(Y)

4. Logistic Regression

$$y = \theta_4 x_4 + \theta_5 x_5 + \theta_6 x_6 + \theta_7$$



독립변수(X)에 해당하는 파라미터 theta 3개와 bias에 해당하는 theta

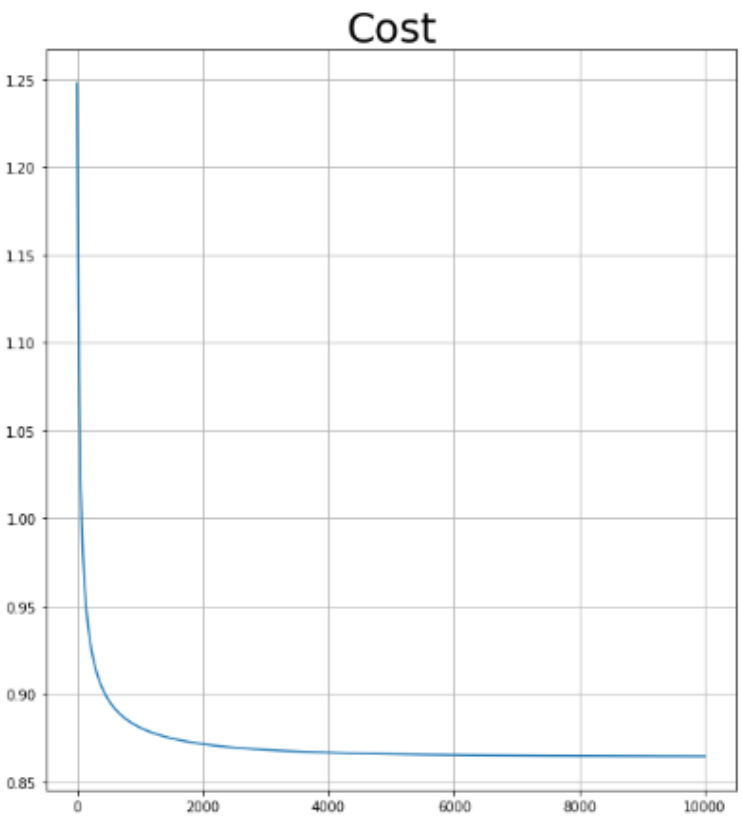
➔ 총 4개의 파라미터를 업데이트

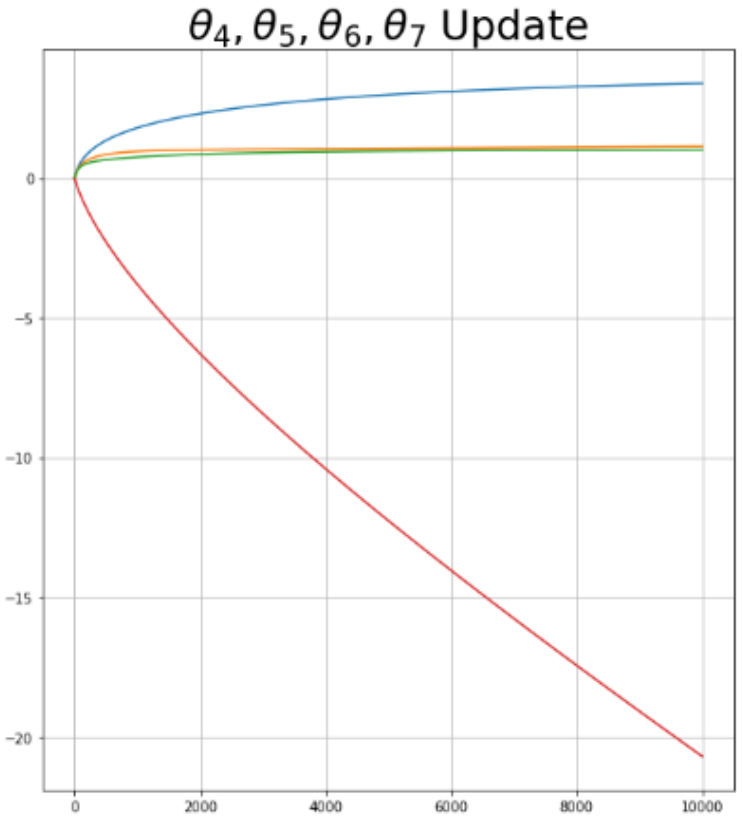
4. Logistic Regression

```
for i in range(epochs):
    Z14 = Z14_node.forward(theta4, data['X4 number of convenience stores'])
    Z15 = Z15_node.forward(theta5, data['X5 latitude'])
    Z16 = Z16_node.forward(theta6, data['X6 longitude'])
    Z2 = Z2_node.forward(Z14, Z15, Z16, theta0)
    Z3 = Z3_node.forward(Z2)
    Z4 = Z4_node.forward(data['Y house price of unit area'], Z3)
    L = L_node.forward(Z4)
    J = J_node.forward(L)

    dL = J_node.backward()
    dZ4 = L_node.backward(dL)
    dY, dZ3 = Z4_node.backward(dZ4)
    dZ2 = Z3_node.backward(dZ3)
    dZ4, dZ5, dZ6, dTheta7 = Z2_node.backward(dZ2)
    dTheta4, dX4 = Z14_node.backward(dZ4)
    dTheta5, dX5 = Z15_node.backward(dZ5)
    dTheta6, dX6 = Z16_node.backward(dZ6)

    theta4 = theta4 - lr*np.sum(dTheta4)
    theta5 = theta5 - lr*np.sum(dTheta5)
    theta6 = theta6 - lr*np.sum(dTheta6)
    theta7 = theta7 - lr*np.sum(dTheta7)

    cost_list.append(J)
    theta4_list.append(theta4)
    theta5_list.append(theta5)
    theta6_list.append(theta6)
    theta7_list.append(theta7)
```

| r = 0.03
Epochs = 10000

4. Logistic Regression





| r = 0.03
Epochs = 10000