데이터 분석 기초

구름

도시공학과 일반대학원

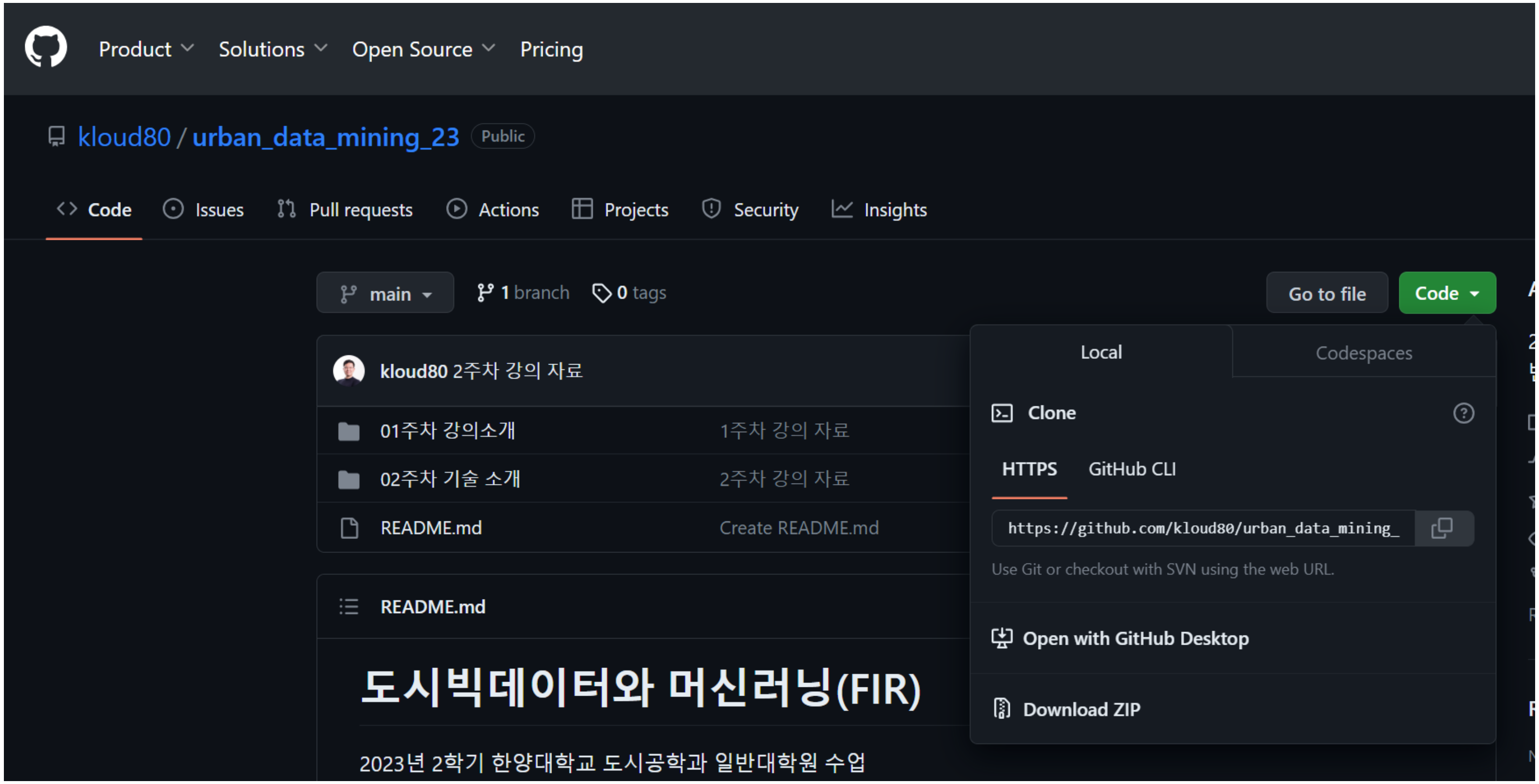
한양대학교

회귀분석

Regression

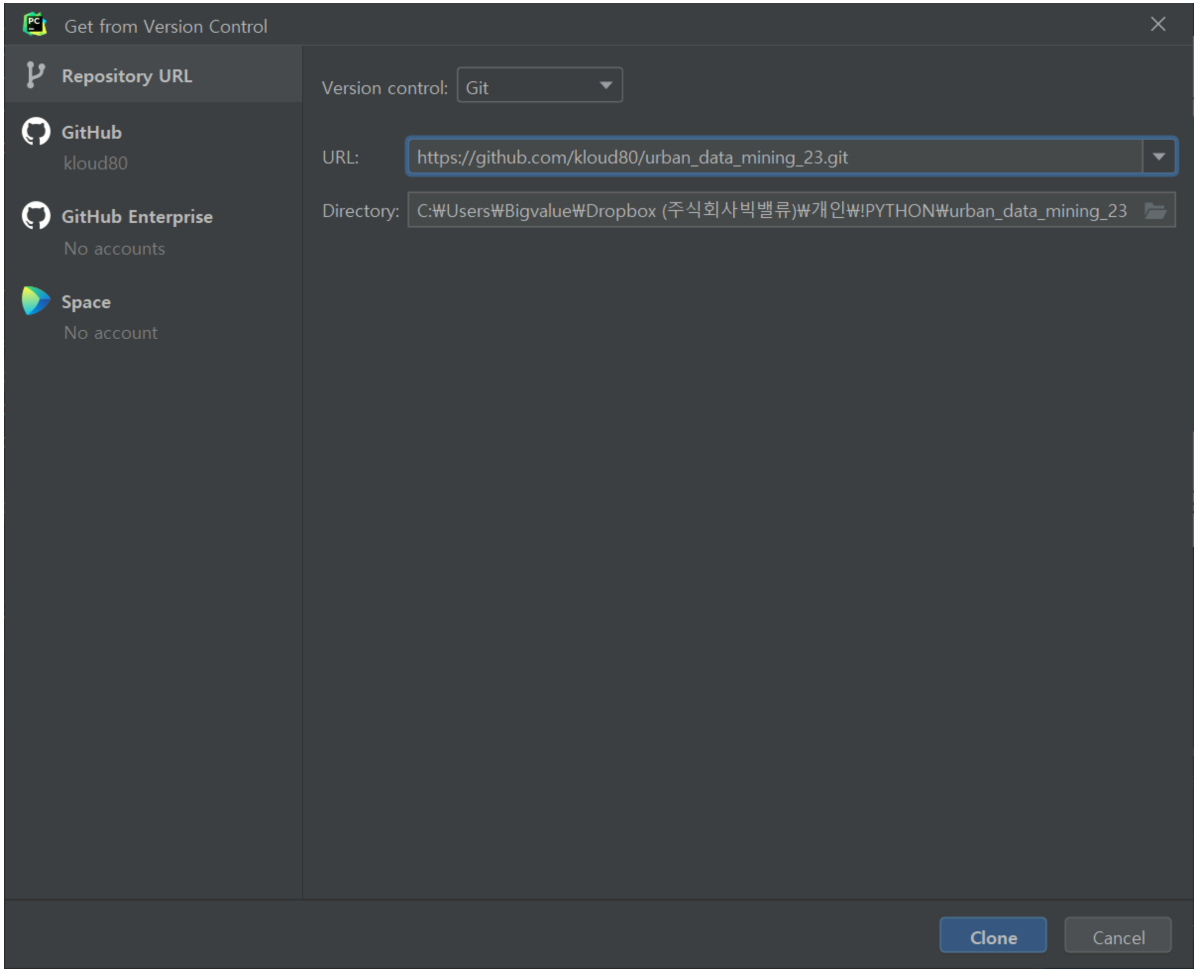
GitHub 데이터 가져오기

https://github.com/kloud80/urban_data_mining_23



Git > clone

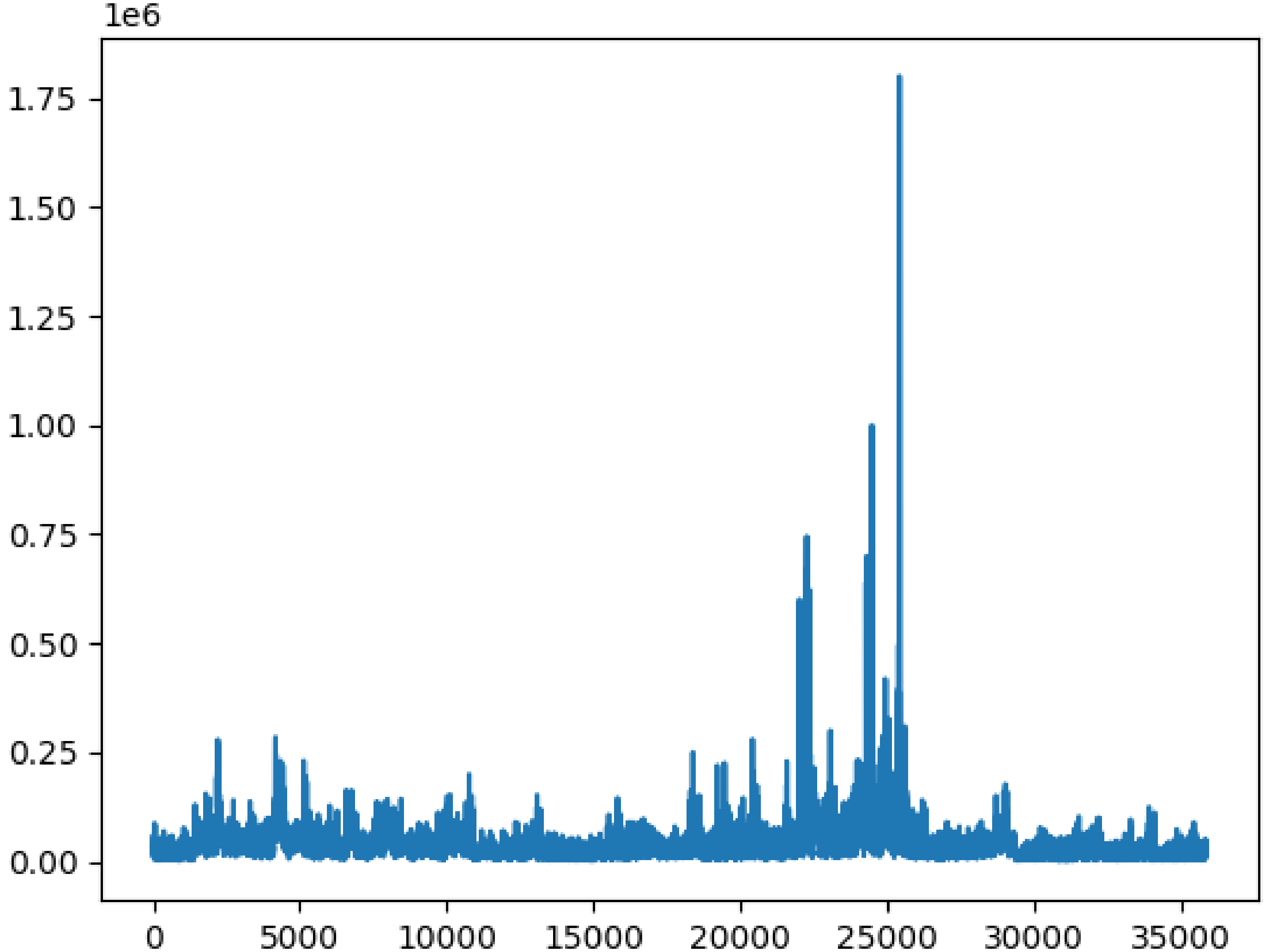
https://github.com/kloud80/urban_data_mining_23.git



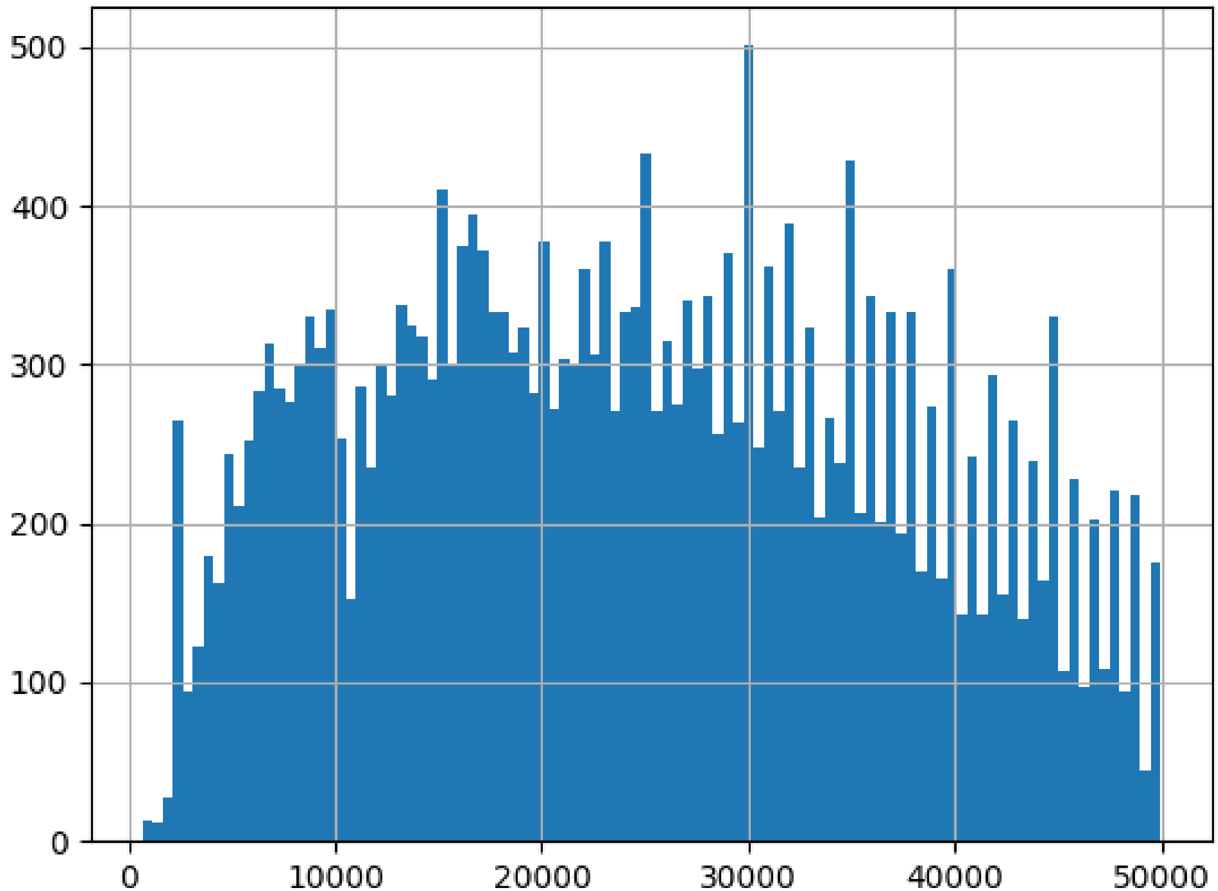
부동산 실거래가 데이터 활용

http://rtdown.molit.go.kr/

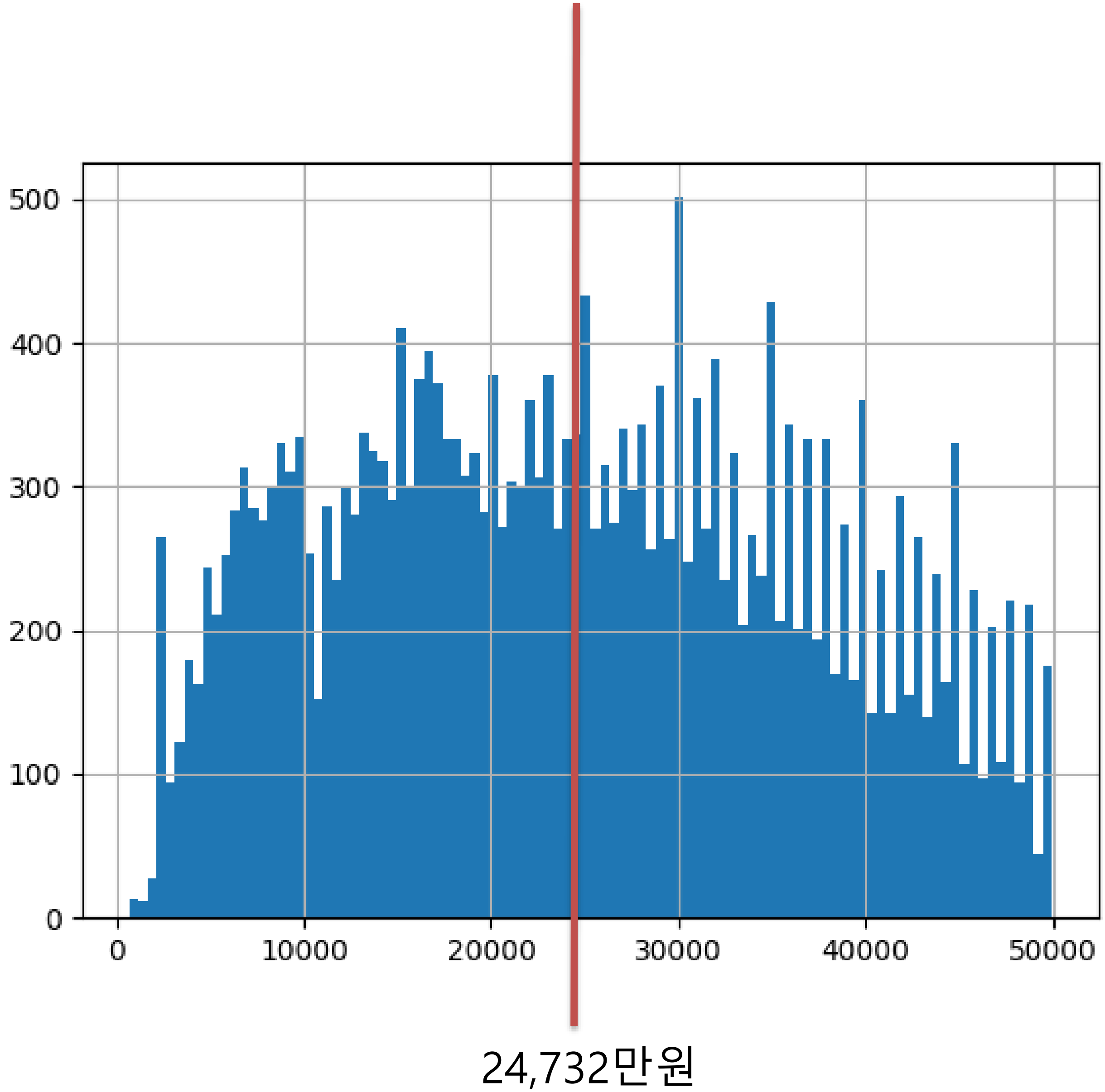
2023년 8월 아파트 실거래가



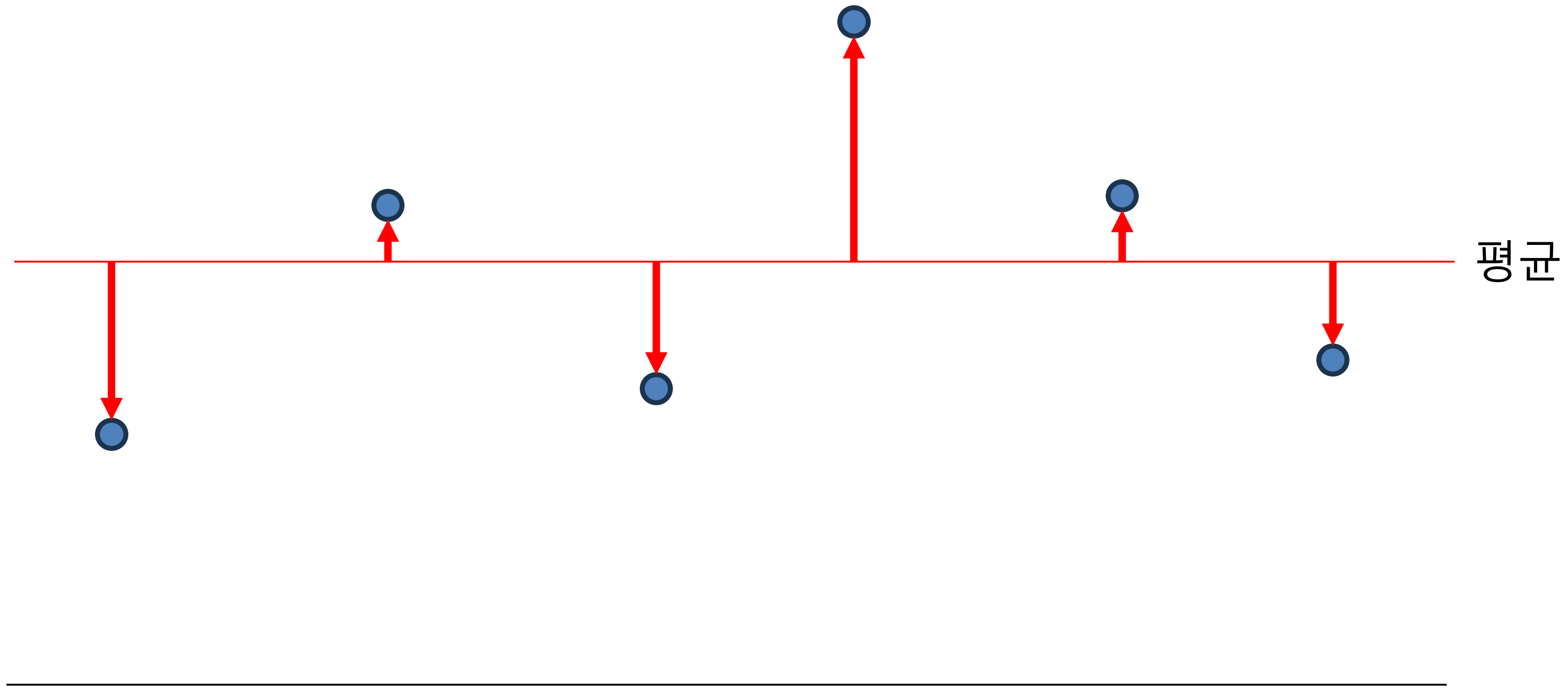
가격을 가장 잘 대표하는 모델



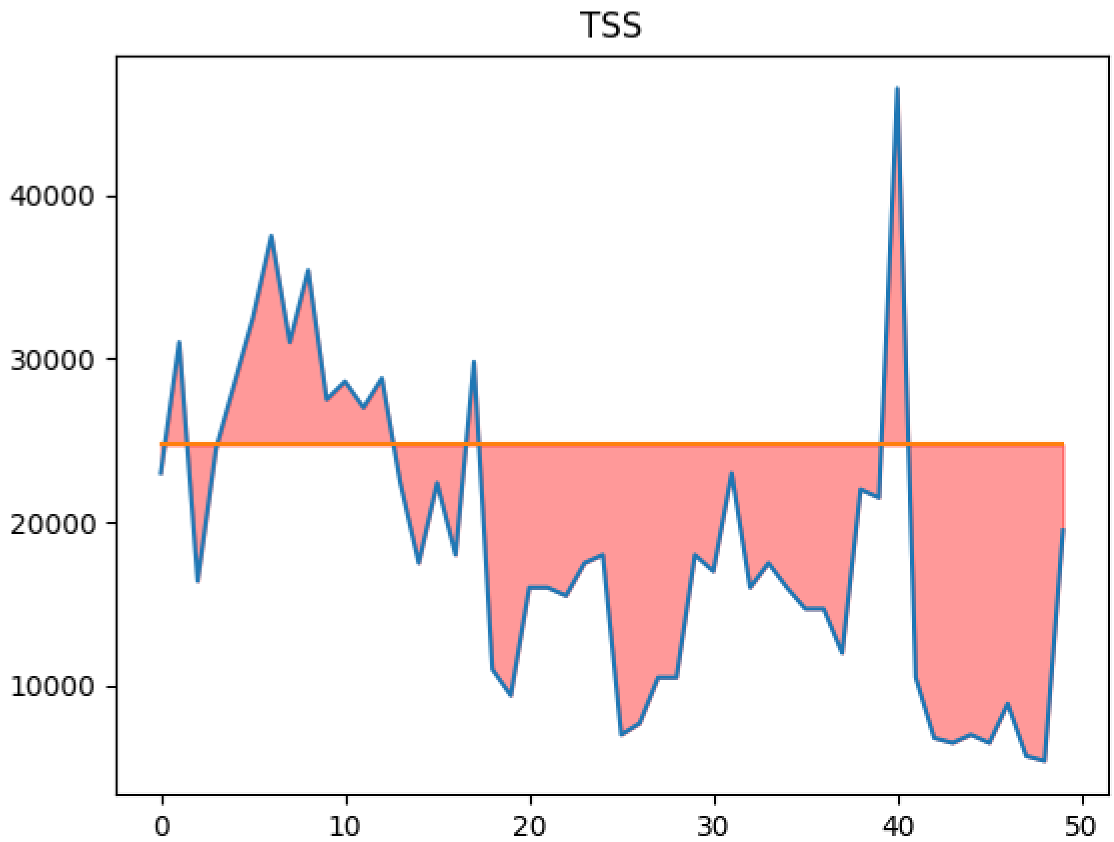
평균이 대표



Total Sum of Squares (TSS)



Total Sum of Squares (TSS)



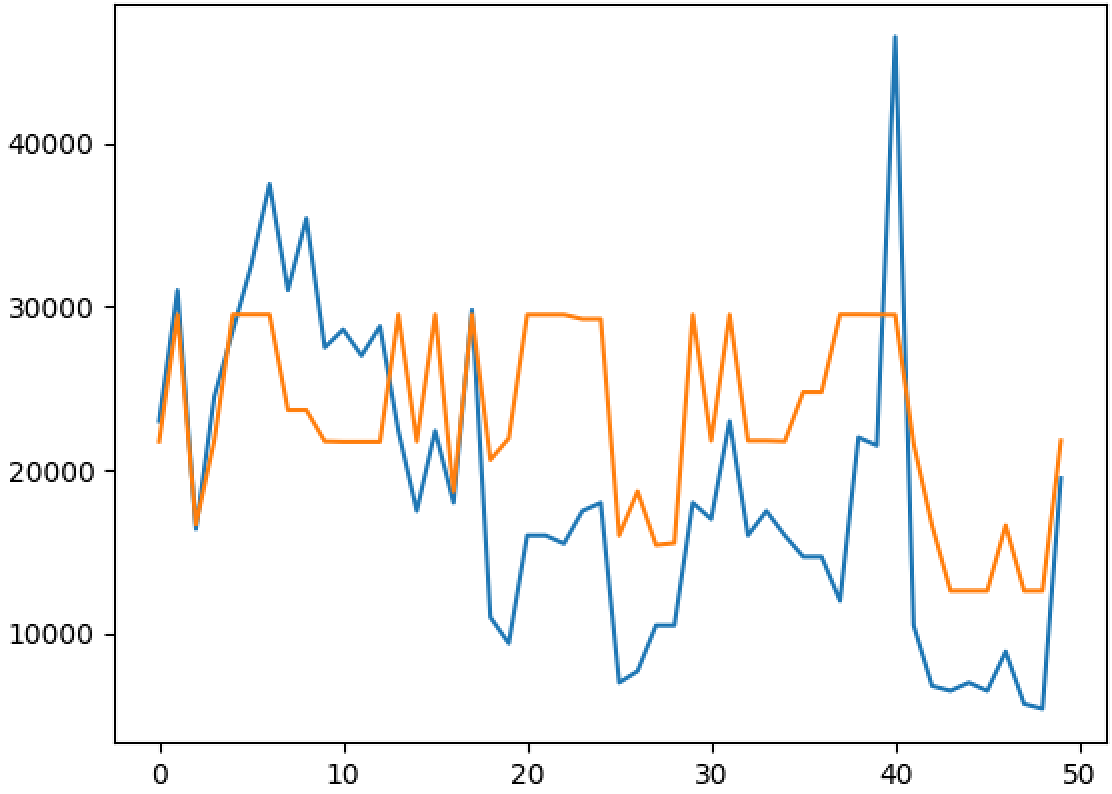
Sum of Squares of Residuals (RSS)

$$y_{매매가} = \alpha \times x_{면적} + \beta + \sigma$$

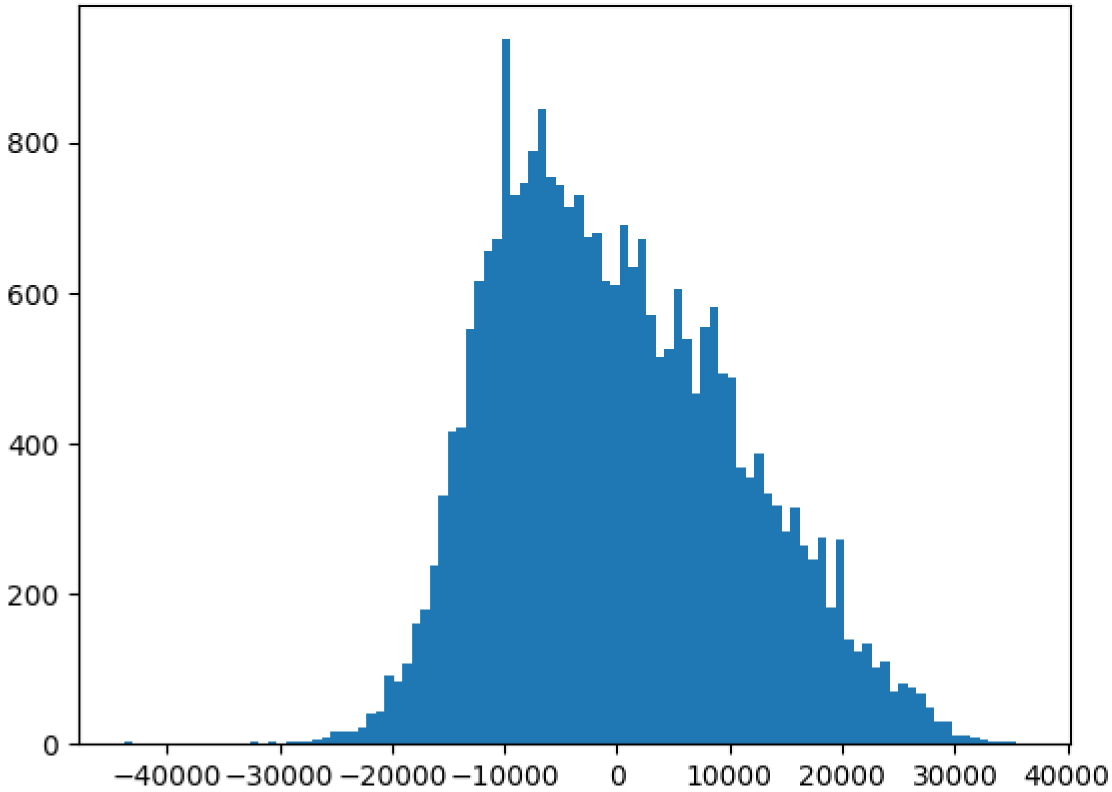
종속변수
Dependent Variable

독립변수
Independent Variable

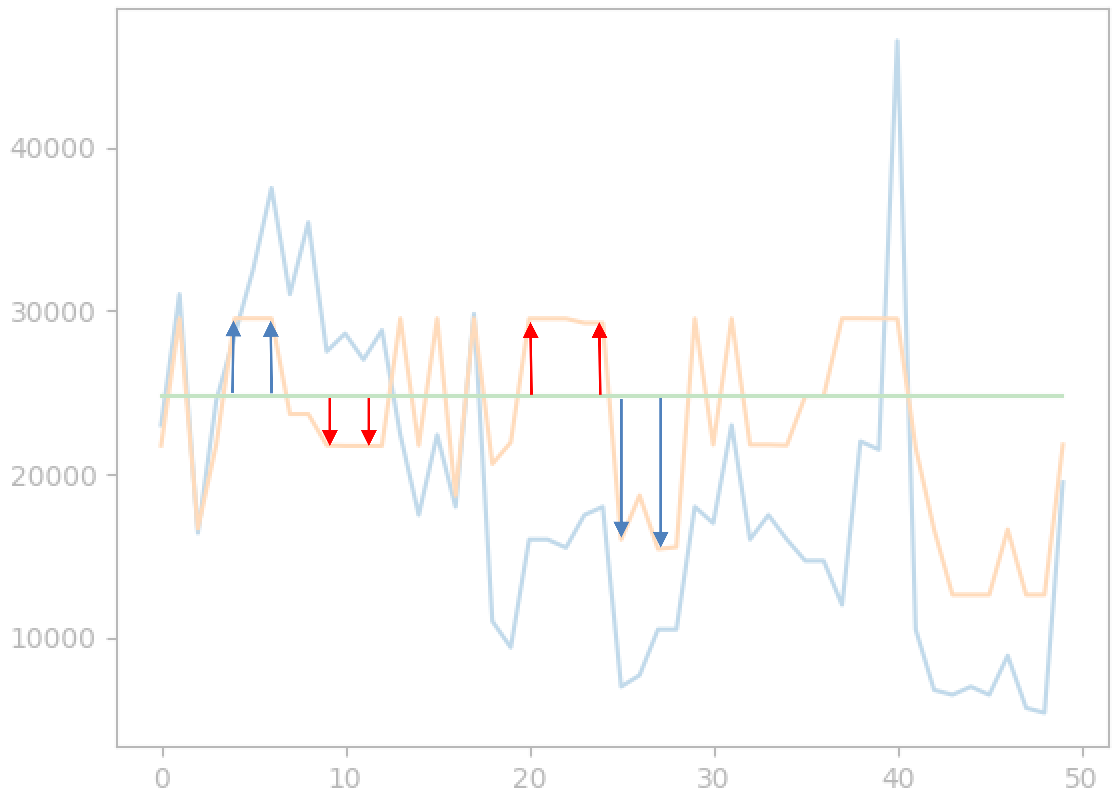
잔차
Residuals



잔차 정규성


800
600
400
200
0
−40000
−30000
−20000
−10000
0
10000
20000
30000
40000

RSS vs TSS



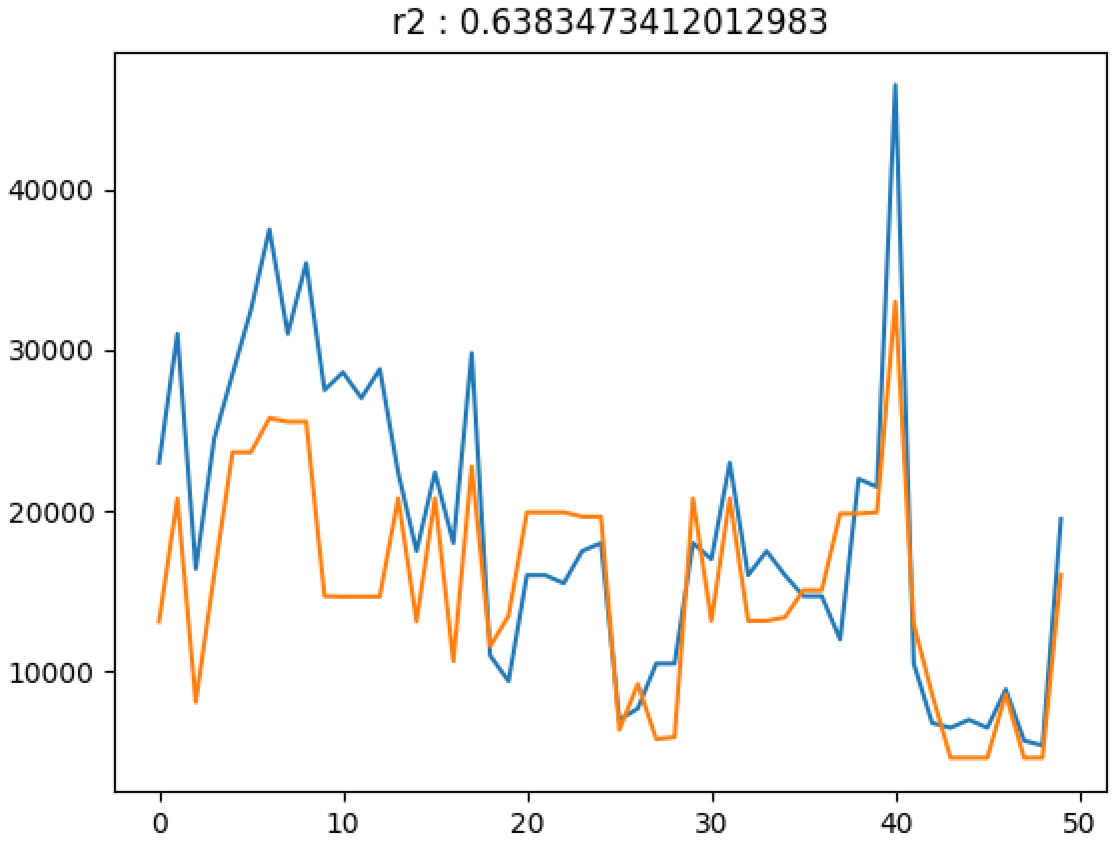
R Sqaure

$$TSS = RSS + ESS$$

$$But, R^2 = 1 - \frac{RSS}{TSS}$$

$$\therefore R^2 = \frac{ESS}{TSS}$$

정보 추가 제공에 따른 모델 설명력 개선



P-Value

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     2440.
Date:                Sat, 16 Sep 2023   Prob (F-statistic):               0.00
Time:                        13:24:50   Log-Likelihood:            -2.7200e+05
No. Observations:               26286   AIC:                         5.440e+05
Df Residuals:                   26266   BIC:                         5.442e+05
Df Model:                          19
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.466e+04    206.899     70.879      0.000    1.43e+04    1.51e+04
x1           305.1522      2.354    129.647      0.000     300.539     309.766
x2          -979.1896     15.413    -63.530      0.000   -1009.400    -948.979
x3            14.1240      0.392     35.998      0.000      13.355      14.893
x4         -3892.8593    204.337    -19.051      0.000   -4293.370   -3492.348
x5          8233.1350    115.739     71.135      0.000    8006.280    8459.990
x6         -3474.4787    159.970    -21.720      0.000   -3788.029   -3160.929
x7         -6965.8511    179.486    -38.810      0.000   -7317.654   -6614.048
x8          1074.0777    222.261      4.832      0.000     638.433    1509.722
x9          1130.1997    184.019      6.142      0.000     769.512    1490.887
x10         3790.6321    235.170     16.119      0.000    3329.687    4251.577
x11         3356.3747    186.337     18.012      0.000    2991.143    3721.606
x12         1.748e+04    319.564     54.696      0.000    1.69e+04    1.81e+04
x13         7401.3005    542.868     13.634      0.000    6337.250    8465.351
x14          156.6670    251.185      0.624      0.533    -335.669     649.003
x15         6609.0067    183.795     35.959      0.000    6248.759    6969.255
x16        -7277.0778    209.252    -34.777      0.000   -7687.223   -6866.933
x17        -5227.6818    203.859    -25.644      0.000   -5627.257   -4828.107
x18         1700.5345    554.715      3.066      0.002     613.263    2787.806
x19        -5285.3587    175.488    -30.118      0.000   -5629.326   -4941.392
x20        -4142.8423    195.837    -21.155      0.000   -4526.693   -3758.992
==============================================================================
Omnibus:                      796.259   Durbin-Watson:                   0.541
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              962.226
Skew:                           0.378   Prob(JB):                    1.14e-209
Kurtosis:                       3.555   Cond. No.                     4.52e+17
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
[2] The smallest eigenvalue is 4.92e-26. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Multicollinearity (다중공선성)

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1.457e+04    244.600     59.579      0.000    1.41e+04    1.51e+04
x1           305.1502      2.354    129.638      0.000     300.536     309.764
x2         -1055.3432    106.391     -9.919      0.000   -1263.875    -846.812
x3            52.5497     80.675      0.651      0.515    -105.577     210.677
x4             3.2739     53.839      0.061      0.952    -102.254     108.802
x5            20.2551     40.472      0.500      0.617     -59.073      99.583
x6            14.1256      0.392     35.999      0.000      13.357      14.895
x7         -3897.2419    204.442    -19.063      0.000   -4297.960   -3496.524
x8          8226.9179    116.070     70.879      0.000    7999.414    8454.422
x9         -3479.2706    160.159    -21.724      0.000   -3793.191   -3165.350
x10        -6970.9894    179.651    -38.803      0.000   -7323.115   -6618.863
x11         1069.9789    222.358      4.812      0.000     634.144    1505.813
x12         1125.0503    184.207      6.108      0.000     763.994    1486.107
x13         3782.9438    235.391     16.071      0.000    3321.565    4244.322
x14         3352.1525    186.508     17.973      0.000    2986.586    3717.719
x15         1.747e+04    319.700     54.652      0.000    1.68e+04    1.81e+04
x16         7389.5387    543.102     13.606      0.000    6325.030    8454.048
x17          151.6712    251.274      0.604      0.546    -340.839     644.182
>>>
```

Stepwise feature selection

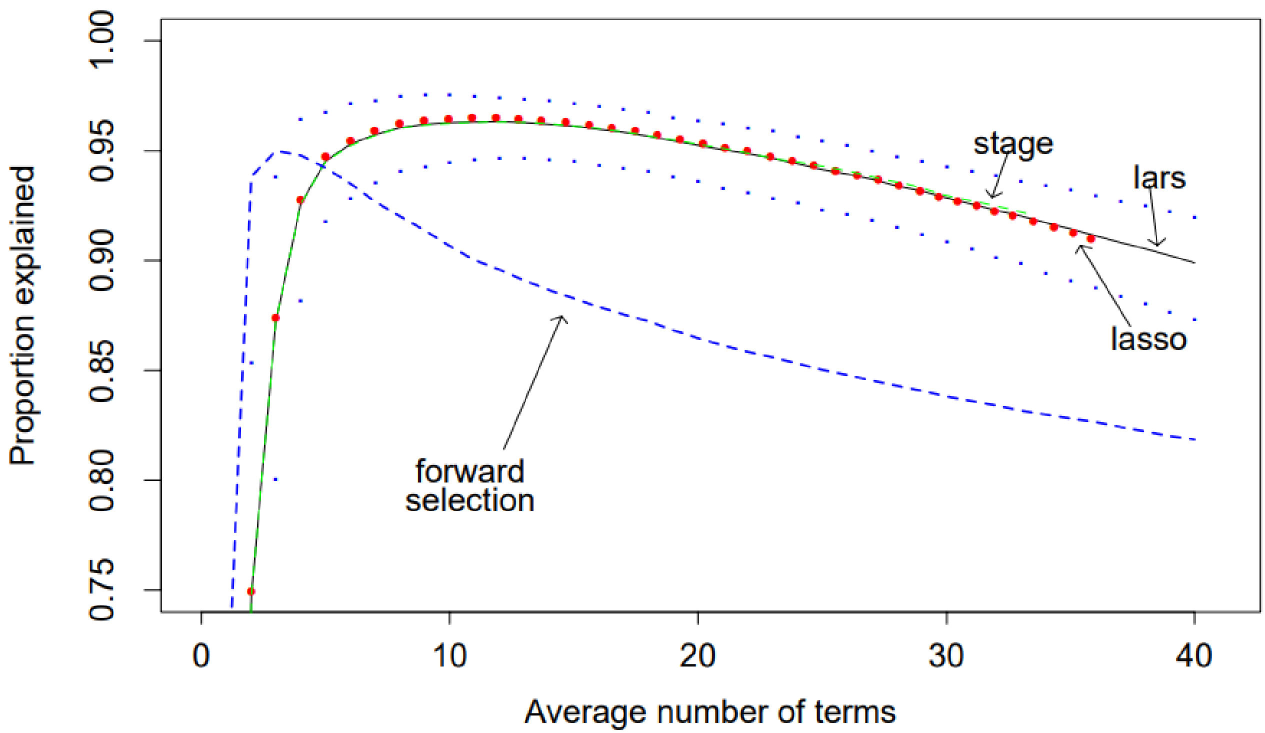


Figure 5. Simulation study comparing LARS, Lasso, and Stagewise algorithms; 100 replications of model (3.15)-(3.16). Solid curve shows average proportion explained, (3.17), for LARS estimates as function of number of steps $k = 1, 2, \ldots, 40$; Lasso and Stagewise give nearly identical results; small dots indicate $\pm$ one standard deviation over the 100 simulations. Classic Forward Selection (heavy dashed curve) rises and falls more abruptly.

https://hastie.su.domains/Papers/LARS/LeastAngle_2002.pdf

주성분분석

Principal Component Analysis

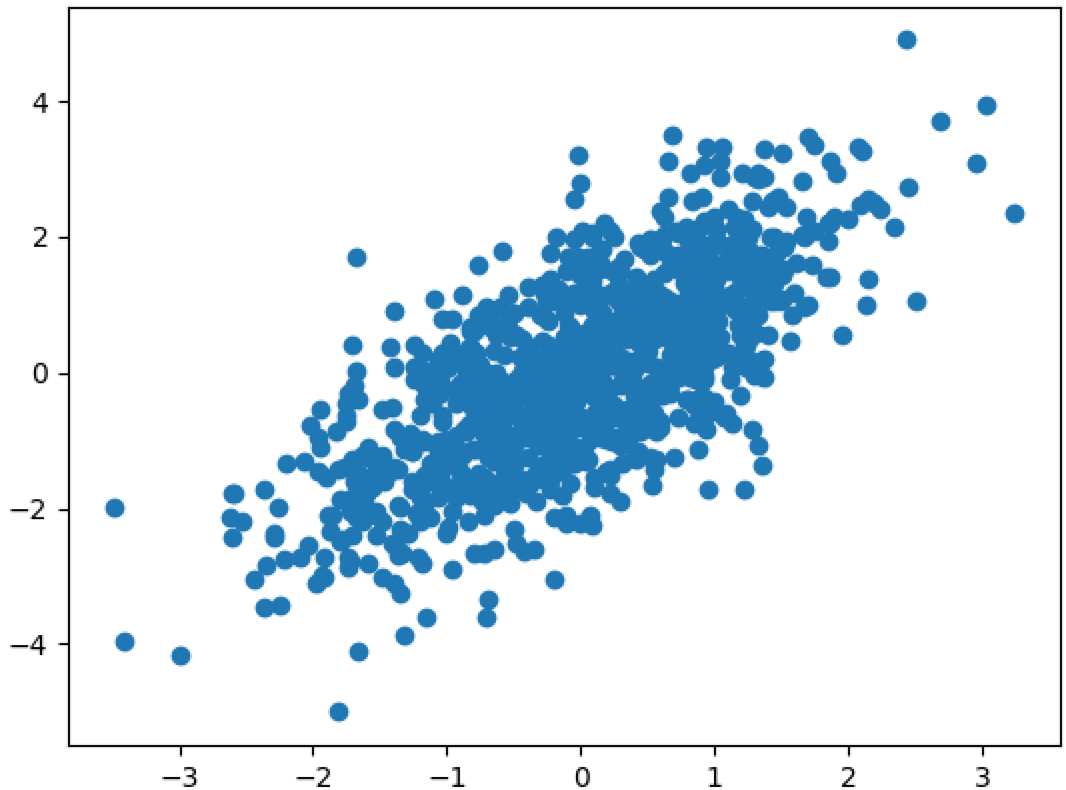
차원이 높다 : 속성 데이터(독립변수) 가 많다

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.466e+04	206.899	70.879	0.000	1.43e+04	1.51e+04
x1	305.1522	2.354	129.647	0.000	300.539	309.766
x2	-979.1896	15.413	-63.530	0.000	-1009.400	-948.979
x3	14.1240	0.392	35.998	0.000	13.355	14.893
x4	-3892.8593	204.337	-19.051	0.000	-4293.370	-3492.348
x5	8233.1350	115.739	71.135	0.000	8006.280	8459.990
x6	-3474.4787	159.970	-21.720	0.000	-3788.029	-3160.929
x7	-6965.8511	179.486	-38.810	0.000	-7317.654	-6614.048
x8	1074.0777	222.261	4.832	0.000	638.433	1509.722
x9	1130.1997	184.019	6.142	0.000	769.512	1490.887
x10	3790.6321	235.170	16.119	0.000	3329.687	4251.577
x11	3356.3747	186.337	18.012	0.000	2991.143	3721.606
x12	1.748e+04	319.564	54.696	0.000	1.69e+04	1.81e+04
x13	7401.3005	542.868	13.634	0.000	6337.250	8465.351
x14	156.6670	251.185	0.624	0.533	-335.669	649.003
x15	6609.0067	183.795	35.959	0.000	6248.759	6969.255
x16	-7277.0778	209.252	-34.777	0.000	-7687.223	-6866.933
x17	-5227.6818	203.859	-25.644	0.000	-5627.257	-4828.107
x18	1700.5345	554.715	3.066	0.002	613.263	2787.806
x19	-5285.3587	175.488	-30.118	0.000	-5629.326	-4941.392
x20	-4142.8423	195.837	-21.155	0.000	-4526.693	-3758.992

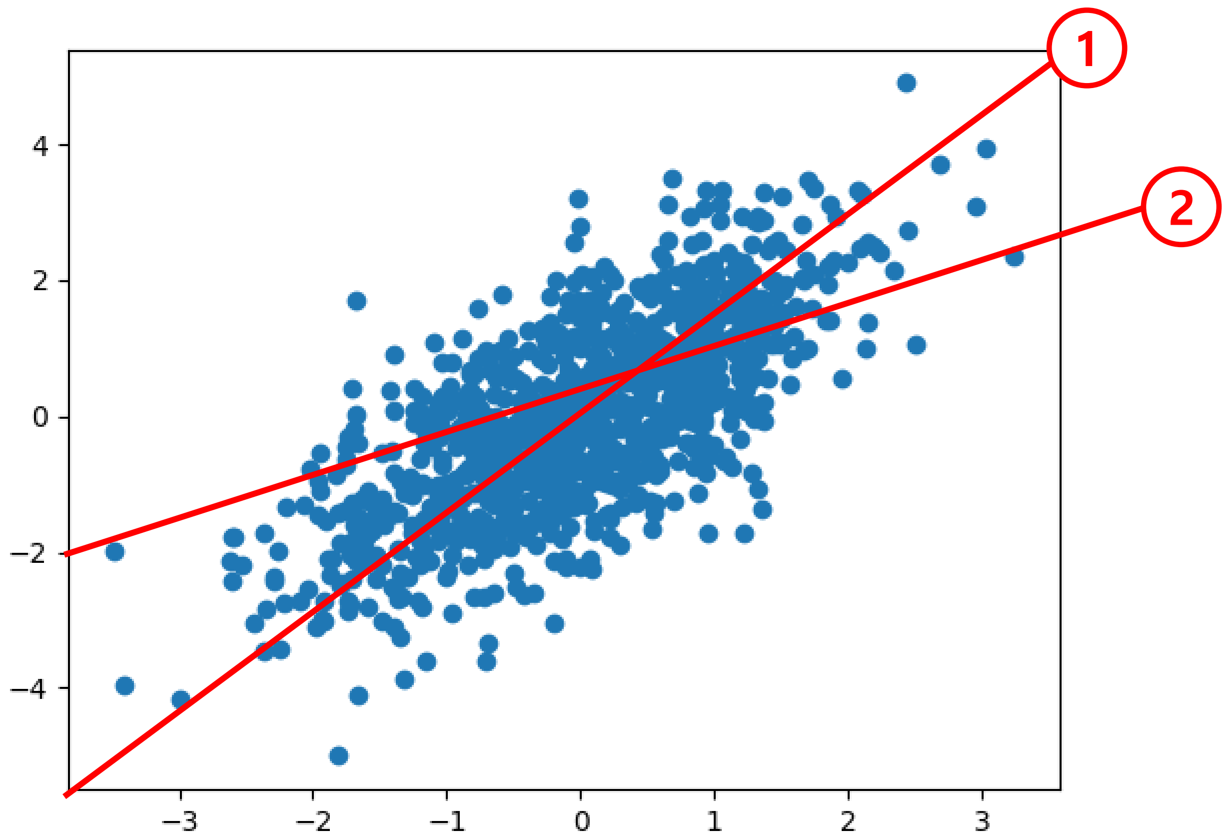
변수의 수를 줄이고 싶다

저차원 평면에 표시

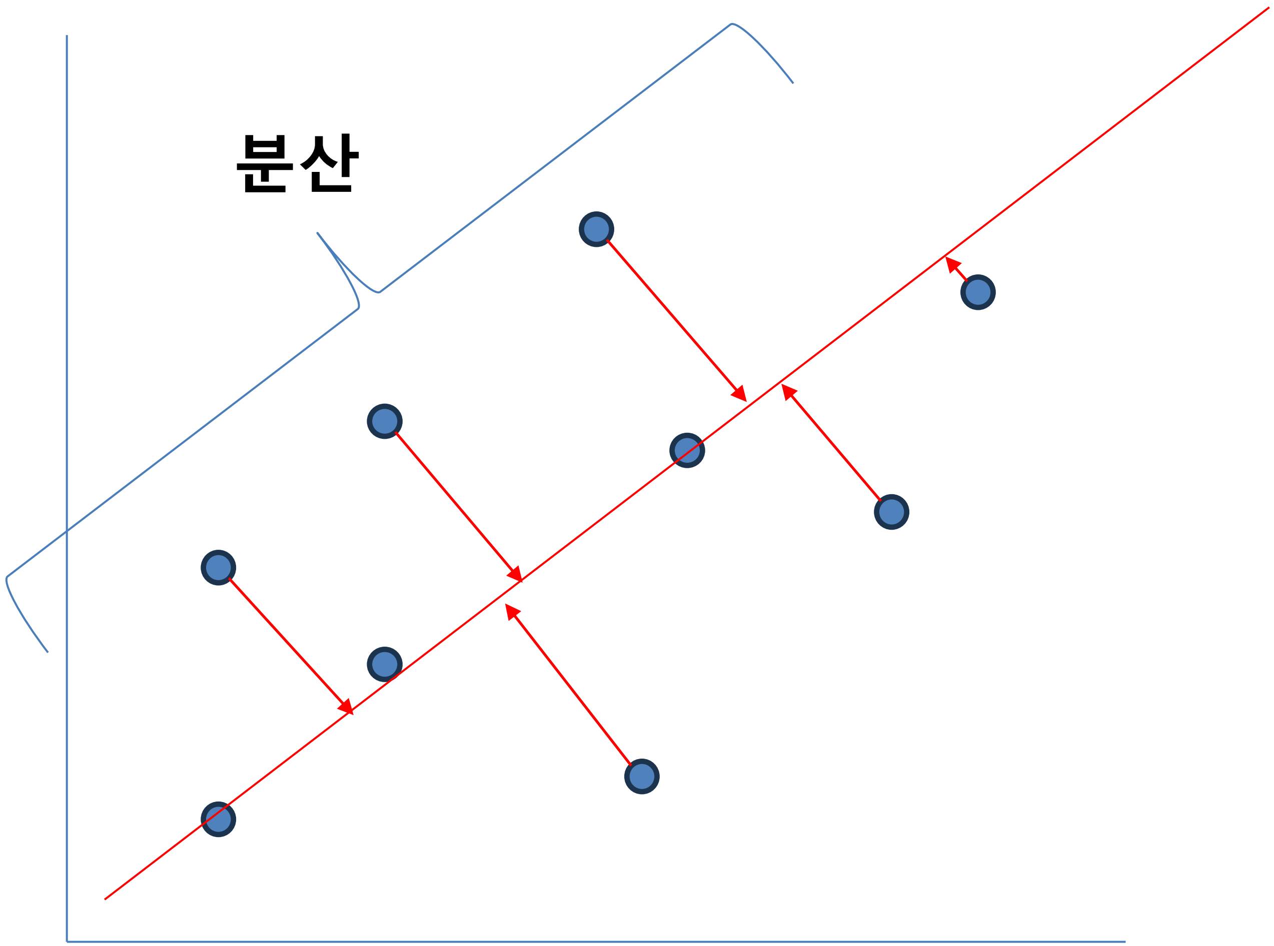
2차원 데이터의 분포



2차원 > 1차원으로 데이터 축소를 하려면 어느 축으로?

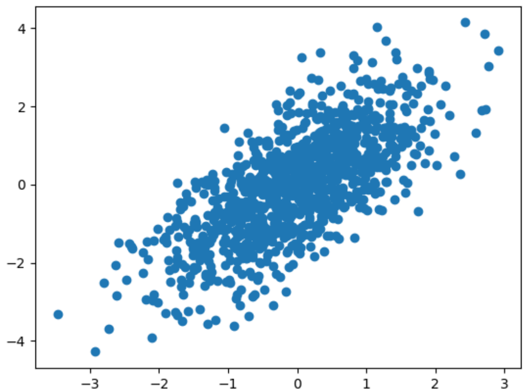


분산이 크면 가장 많은 설명력을 담은 저차원 곡선이다.

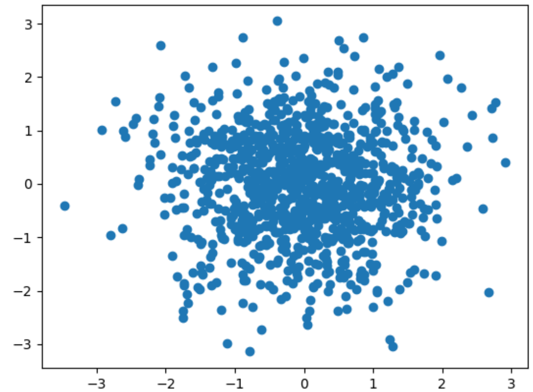


공분산 행렬은 데이터의 산포도를 의미

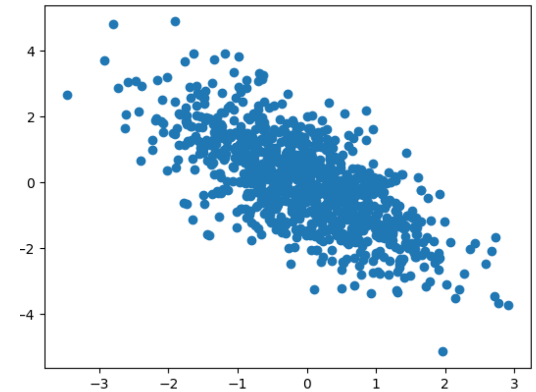
X축 방향 퍼진 정도	x,y축 방향으로 함께 퍼진 정도
x,y축 방향으로 함께 퍼진 정도	y축 방향 퍼진 정도



92.71%	92.05%
92.05%	183.83%



92.71%	-0.32%
-0.32%	100.49%



92.71%	-90.50%
-90.50%	187.95%

행렬과 벡터

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

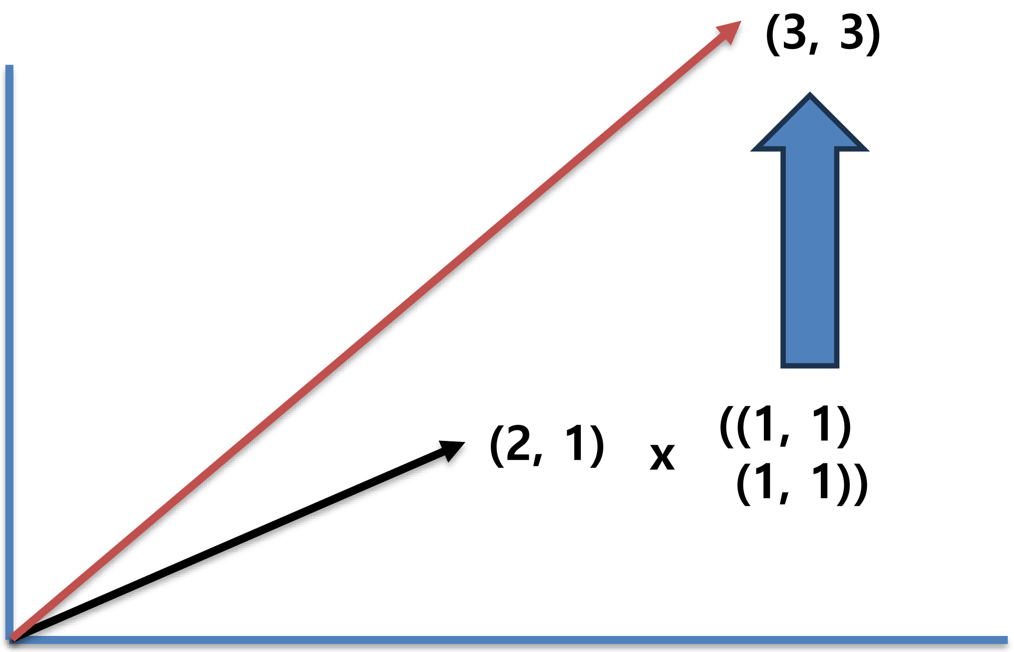
데이터

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

데이터 목록
가중치

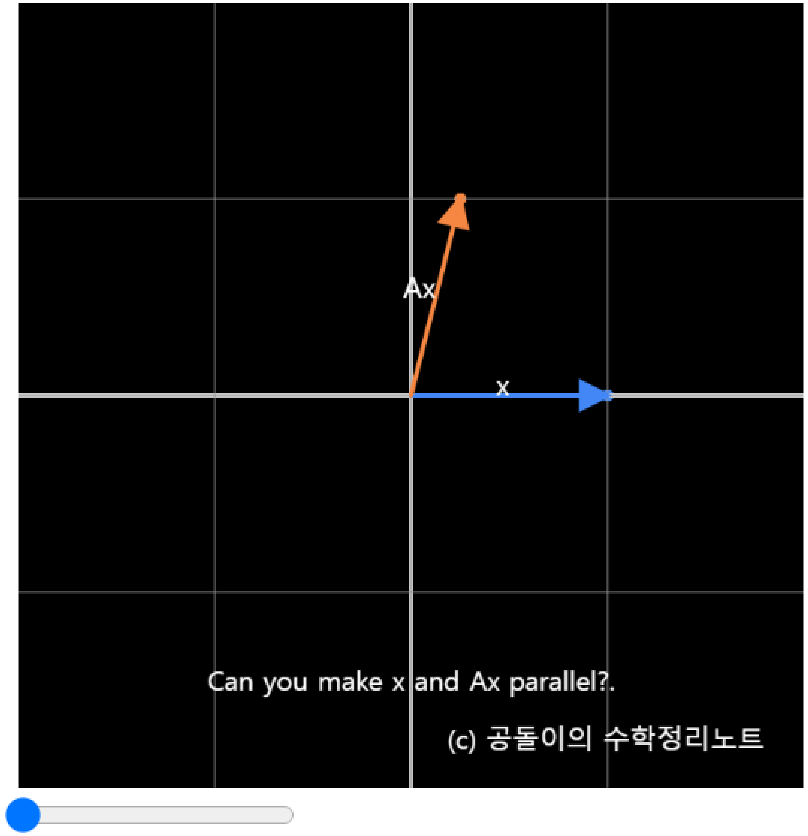
행렬 연산을 통해 벡터가 변함

벡터 * 행렬 = 벡터

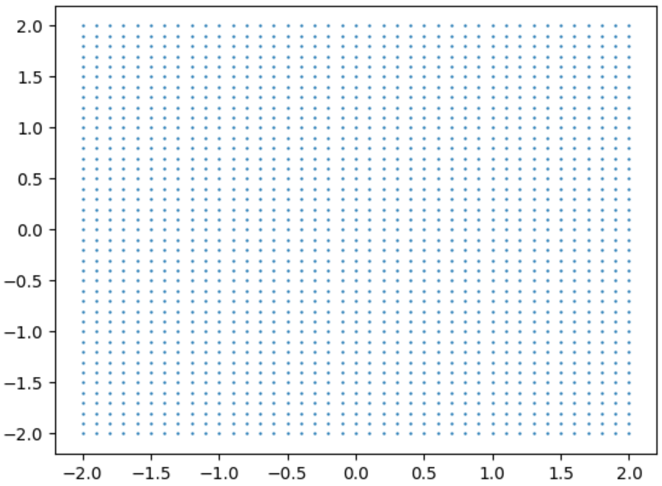


고유벡터(eigenvector)와 고유치(eigenvalue)

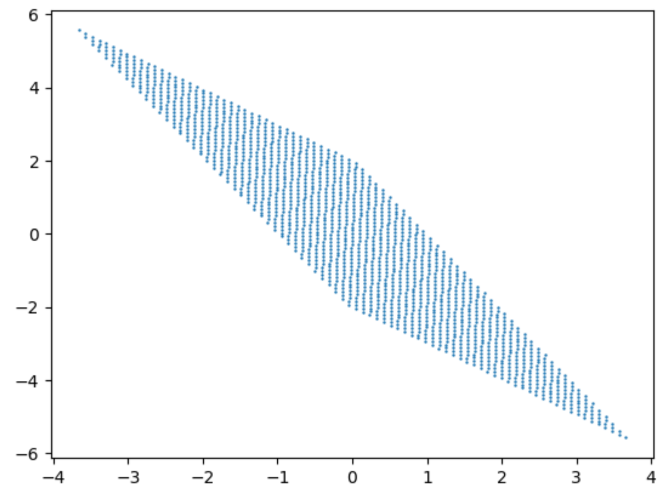
https://angeloyeo.github.io/2019/07/17/eigen_vector.html



공분산 행렬을 곱하면 벡터들의 변형이 일어남

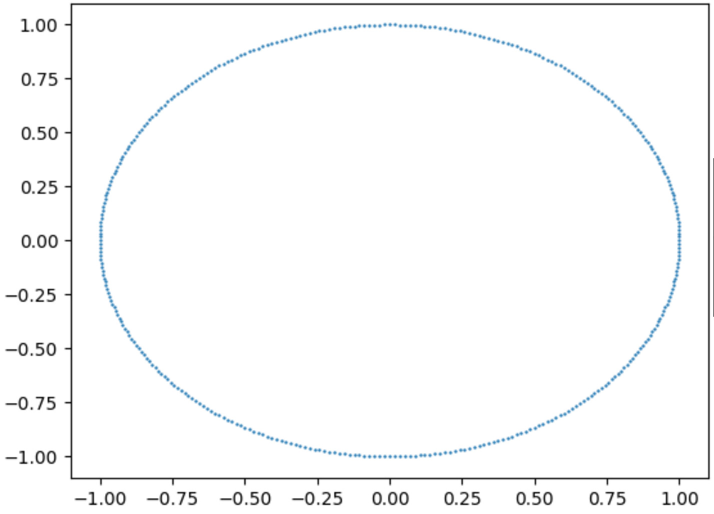


92.71%	-90.50%
-90.50%	187.95%



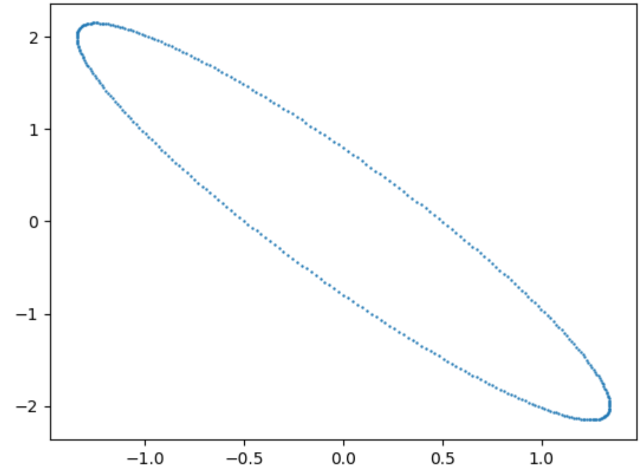
길이가 1인 벡터들을 공분산 행렬로 변환

모든 벡터의 크기가 1인 단위 벡터들

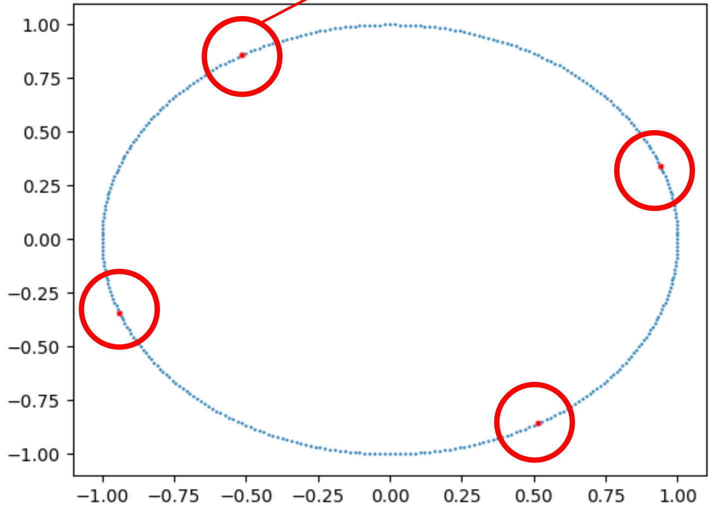


92.71%	-90.50%
-90.50%	187.95%

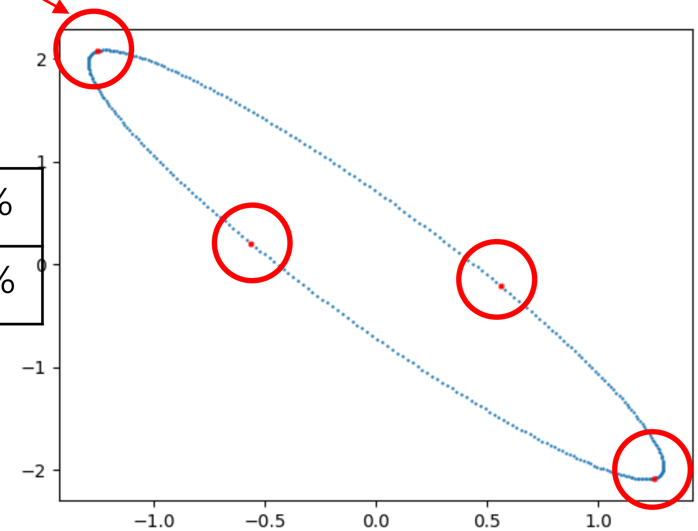
공분산 행렬로 변환한 결과



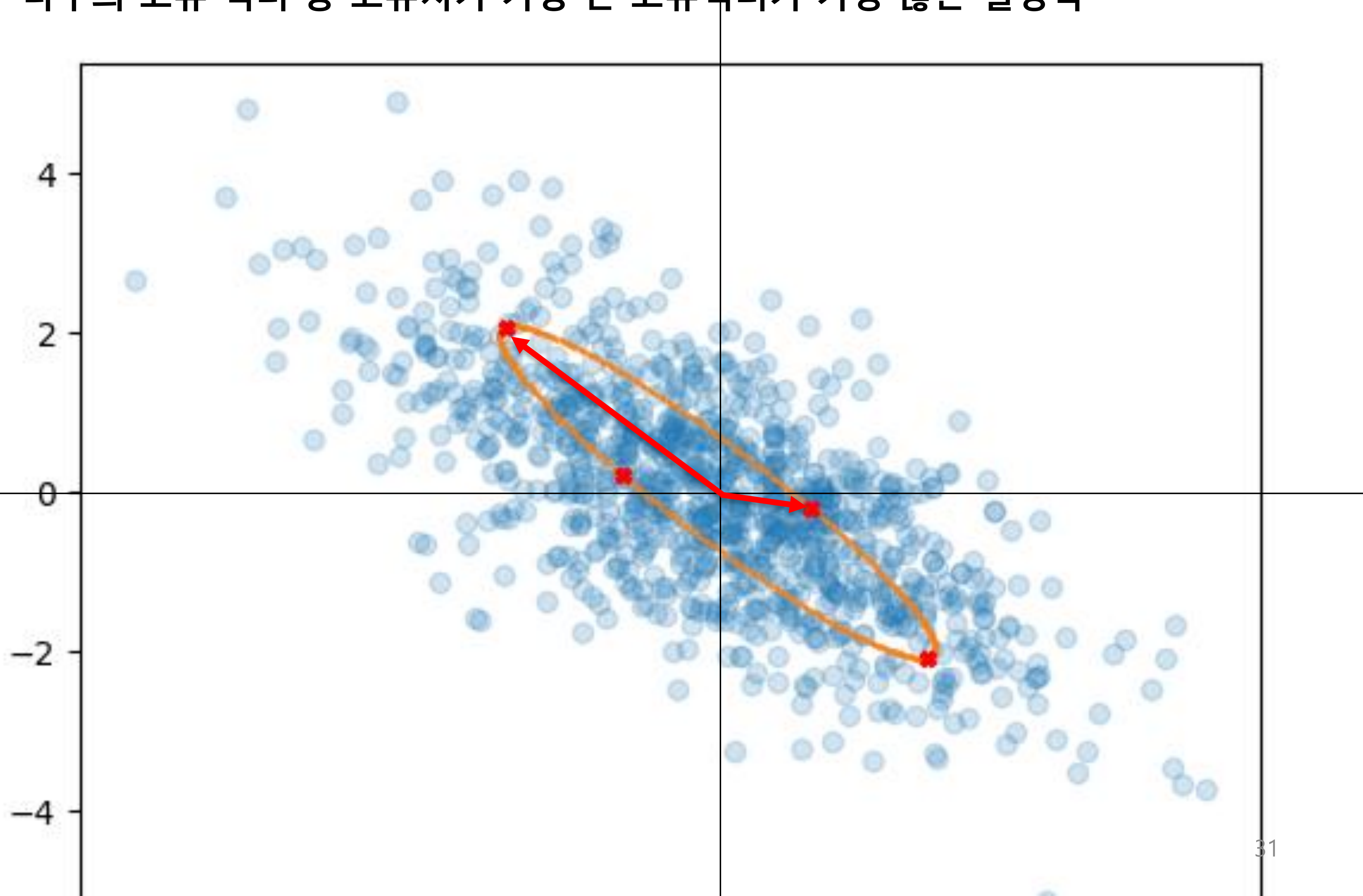
고유치만큼 길이만 변형되고 각도가 그대로인 고유 벡터 존재



92.71%	-90.50%
-90.50%	187.95%

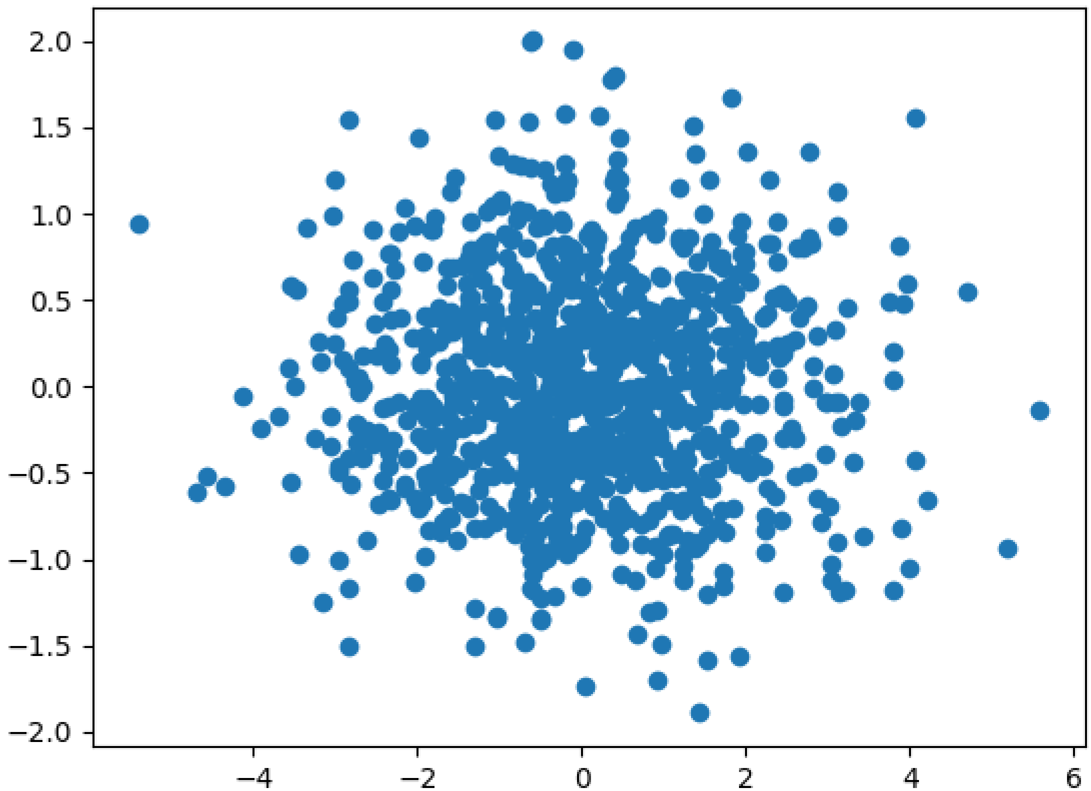


PCA는 공분산의 고유벡터가 가장 데이터를 잘 설명함
다수의 고유 벡터 중 고유치가 가장 큰 고유벡터가 가장 많은 설명력



PCA 분석 결과 각 고유치 변환 행렬 : pca. components_
고유치값(eigenvalue) 크기 비율 : explained_variance_ratio_

주성분2



주성분1

```
>>> pca.components_
array([[-0.51687956,  0.85605813],
       [-0.85605813, -0.51687956]])
>>> pca.explained_variance_ratio_
array([0.86436415, 0.13563585])
```