

2021 기계학습 퀴즈 및 중간고사 가이드라인

온라인 시험(웹엑스) 접속안내

	미팅룸 정보	응시자
1반 - A	<p>Meeting link: https://sejong.webex.com/sejong/j.php?MTID=ma8eac9a7d00244b863d832c6bccae85c</p> <p>Meeting number: 184 023 8220</p> <p>Password: 20210412</p>	16011137 17011768 17011771 17011777 17011778 17011787 17011808 17011814 17011815 17011838 17011846 17011848 17011864 17011878 17011881 17013243 18011701 18011703 18011801 18012576 18013320
1반 - B	<p>Meeting link: https://sejong.webex.com/sejong/j.php?MTID=m9e10554b70d5f9ddcded76bb59c68366</p> <p>Meeting number: 184 179 6792</p> <p>Password: NPhZJPrM456</p>	18013321 19011461 19011463 19011494 19011724 19011733 19011750 19011755 19011756 19011757 19011769 19011791 19011798 19011800 19011811 19011822 19011824 19011825 19011827 19011837 19013236

	미팅룸 정보	응시자
2반 – A	<p>Meeting link: https://sejong.webex.com/sejong/j.php?MTID=m12e066f174604790e473fc9f0031392d</p> <p>Meeting number: 184 239 9994</p> <p>Password: mhNMaWSN473</p>	17010387 17011520 17011546 17011571 17011756 17011767 17011774 17011779 17011781 17011788 17011791 17011800 17011801 17011805 17011806 17011817 17011824 17011860 17011873 17011886 17011887
2반 – B	<p>Meeting link: https://sejong.webex.com/sejong/j.php?MTID=m3ded938608c76022c255623183588c22</p> <p>Meeting number: 184 747 4918</p> <p>Password: q4FYdhnNk37</p>	18011660 18011789 18011835 18011860 18013313 19011719 19011728 19011773 19011781 19011801 19011804 19011805 19011808 19011821 19011826 19011832 19011845 19011847 19011854 19011857 19012603

시험 중 자세관련 안내

바른 예시



잘못된 예시



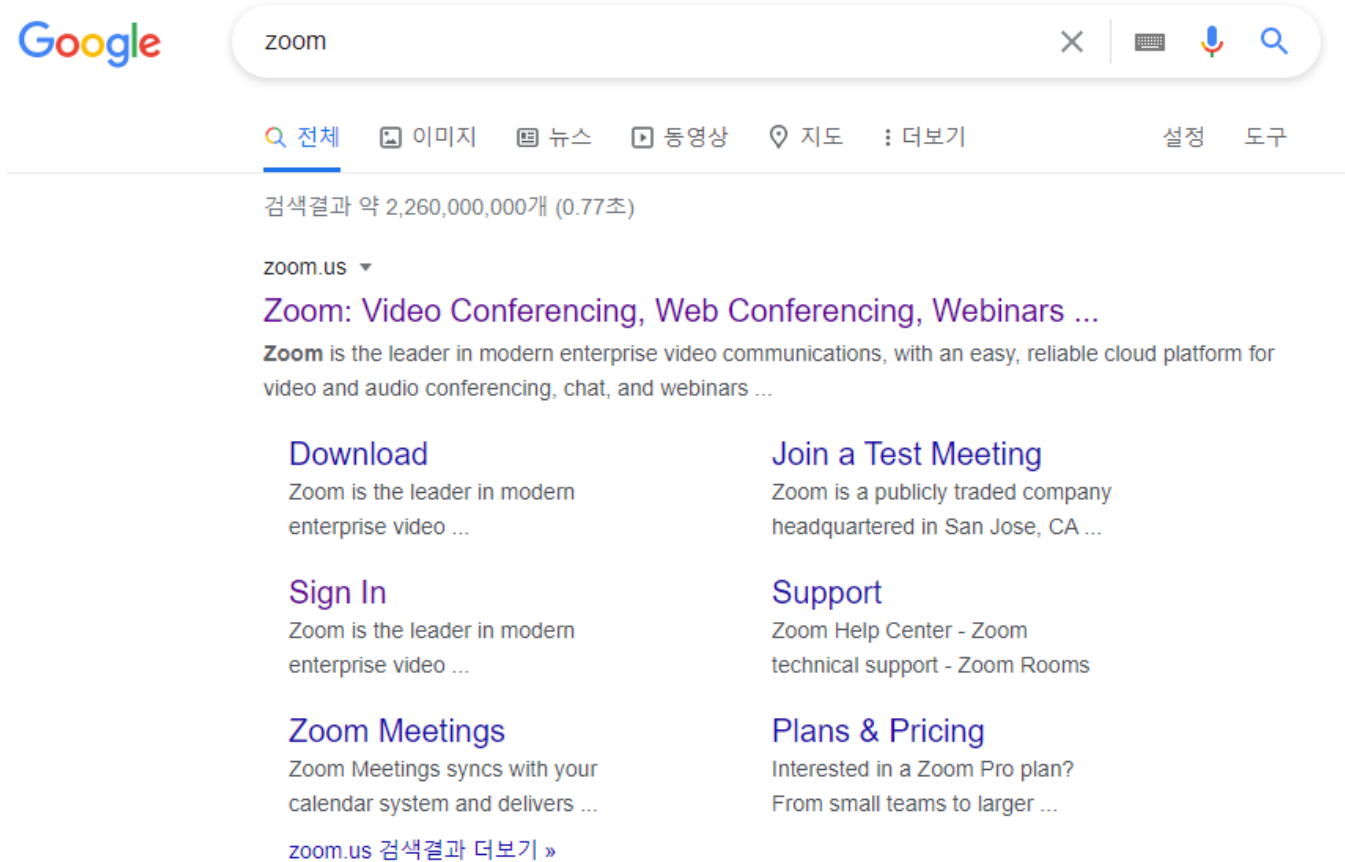
주변환경이 보이지 않음



응시자가 구분이 안됨

모니터 화면은 녹화하여 제출할 예정이므로 손, 옆모습 그리고 주변환경이 나오도록 카메라를 세팅합니다.

자신의 모니터화면 녹화 방법



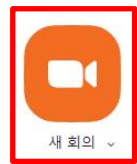
Zoom을 통한 동영상 녹화를 강력히 권장합니다. Zoom을 이용할 경우 동영상 파일의 용량을 줄일 수 있습니다.

[솔루션 ▼](#)[요금제 및 가격 책정](#)[영업에 문의](#)[회의 참가](#)[회의 호스팅 ▼](#)[로그인](#)[무료로 가입하세요](#)

Zoom: 가트너 2019 매직 쿼드런트에서 회의 솔루션 분야 리더

[보고서 읽기](#)

Zoom 다운로드 및 설치



새 회의



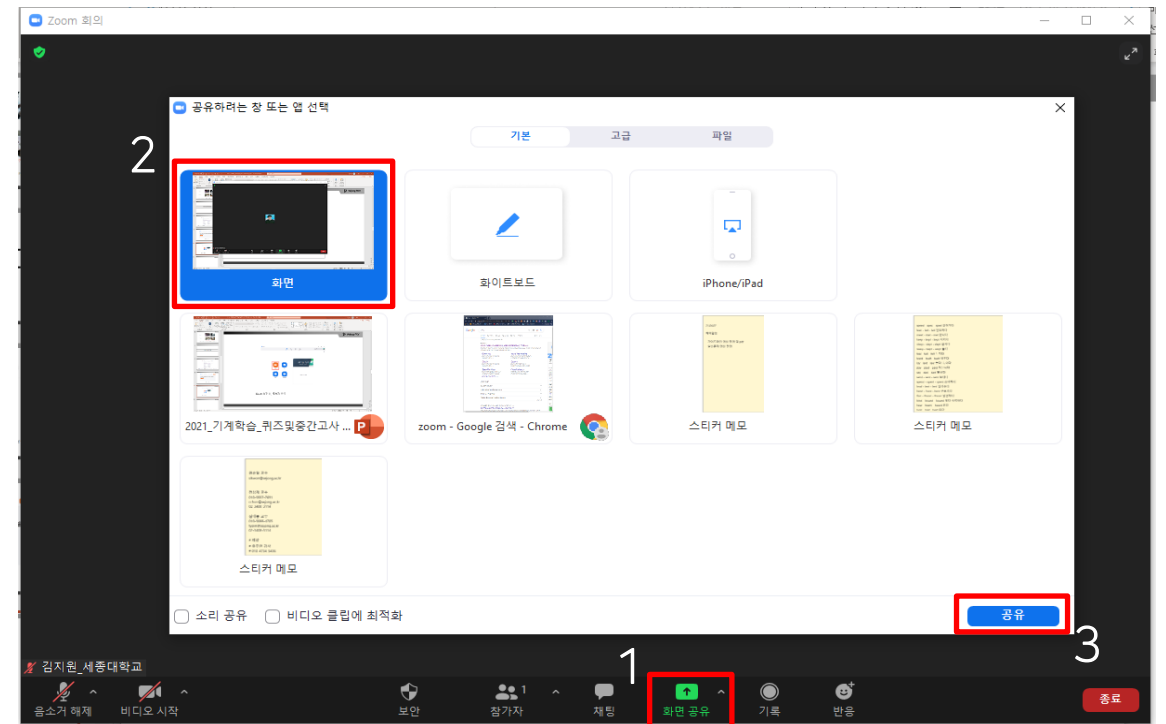
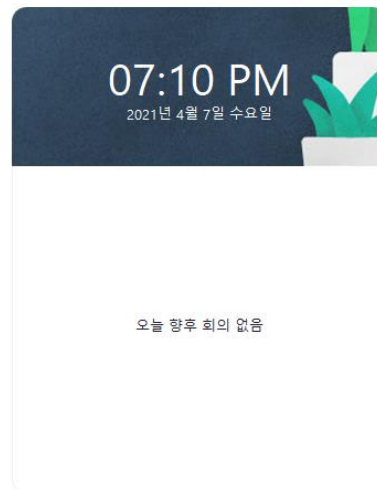
참가



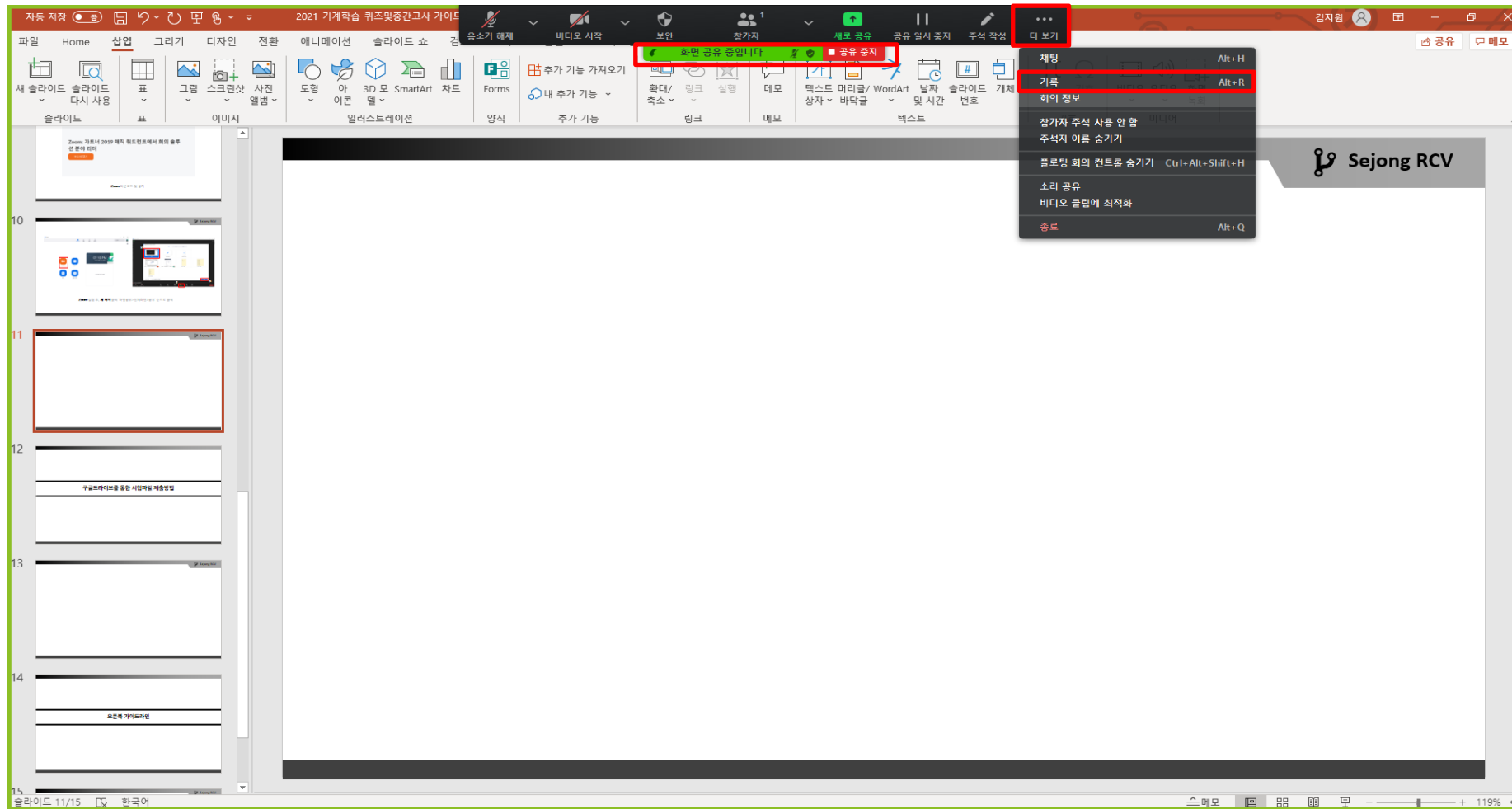
예약



화면 공유

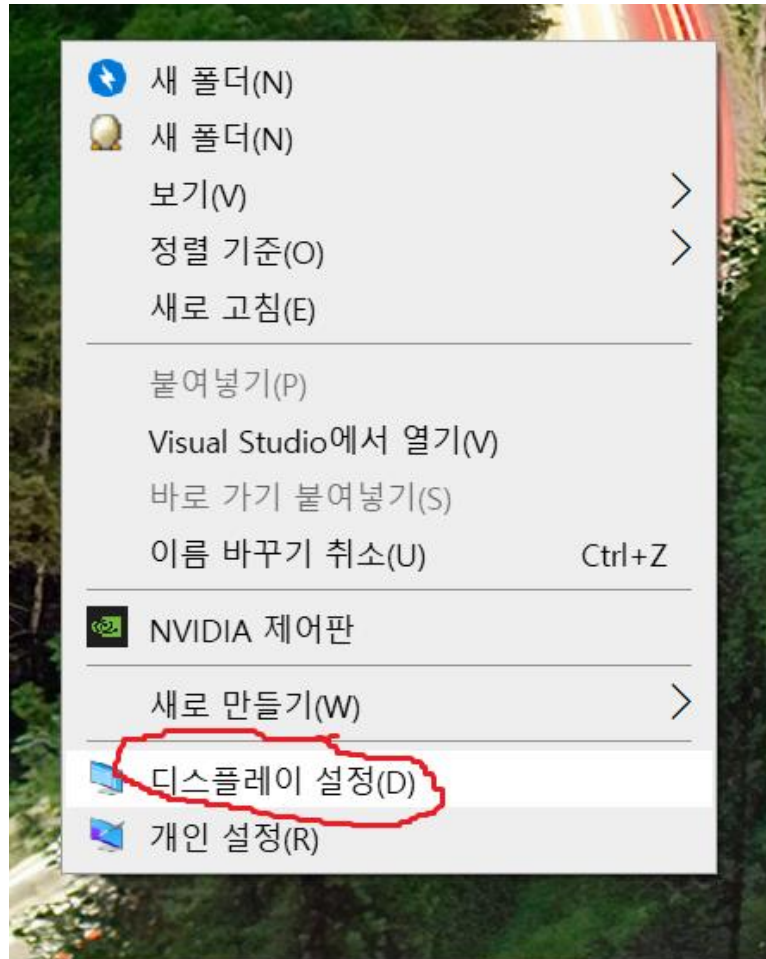


Zoom 실행 후, 새 회의 클릭 '화면공유>전체화면>공유' 순서로 클릭



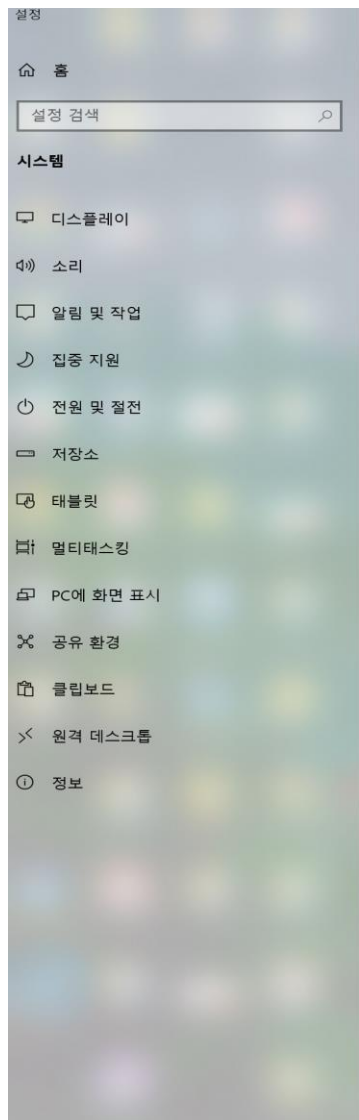
상단 탭에서 '더보기>기록' 클릭, 녹화시작 및 시험응시

듀얼모니터 확인 방법 CHECK (WINDOW)



1. 바탕화면에서 우측 클릭
2. 왼쪽 그림과 같은 창이 뜨면 디스플레이 설정 클릭

- 단일 모니터의 경우 좌측 그림과 같이 모니터 그림이 없음.
- 듀얼 모니터의 경우 우측 그림과 같이 모니터 그림이 존재.



디스플레이

색

야간 모드
☒ 끄

[야간 모드 설정](#)

색 프로필

DELL U2720Q Color Profile, D6500

Windows HD Color

HDR를 지원하는 비디오, 게임 및 앱에서 보다 밝고 생생한 화질을 감상하세요.

[Windows HD Color 설정](#)

HDR 사용

☒ 끄

배율 및 레이아웃

텍스트, 앱 및 기타 항목의 크기 변경

150%(권장)

[고급 배율 설정](#)

디스플레이 해상도

3840 × 2160(권장)

디스플레이 방향

가로

여러 디스플레이

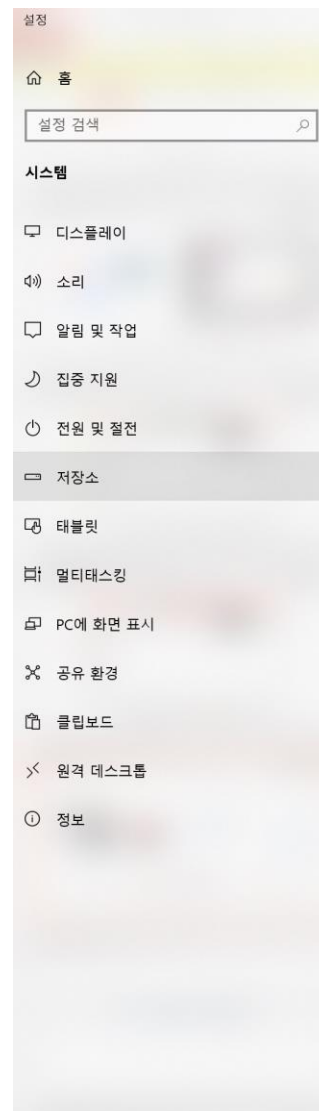
[무선 디스플레이에 연결](#)

오래된 디스플레이는 자동으로 연결되지 않을 수도 있습니다. 연결을 시도하려면 검색을 선택하세요.

검색

[고급 디스플레이 설정](#)

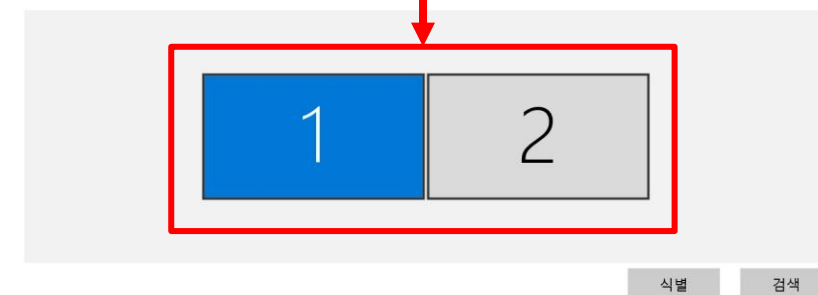
[그래픽 설정](#)



디스플레이

디스플레이 다시 정렬

설정을 변경하려면 아래에서 디스플레이를 선택하세요. 디스플레이를 길게 누르거나 선택한 다음 드래그하여 다시 정렬하세요.



색

야간 모드
☒ 끄

[야간 모드 설정](#)

색 프로필

DELL U2720Q Color Profile, D6500

Windows HD Color

HDR를 지원하는 비디오, 게임 및 앱에서 보다 밝고 생생한 화질을 감상하세요.

[Windows HD Color 설정](#)

배율 및 레이아웃

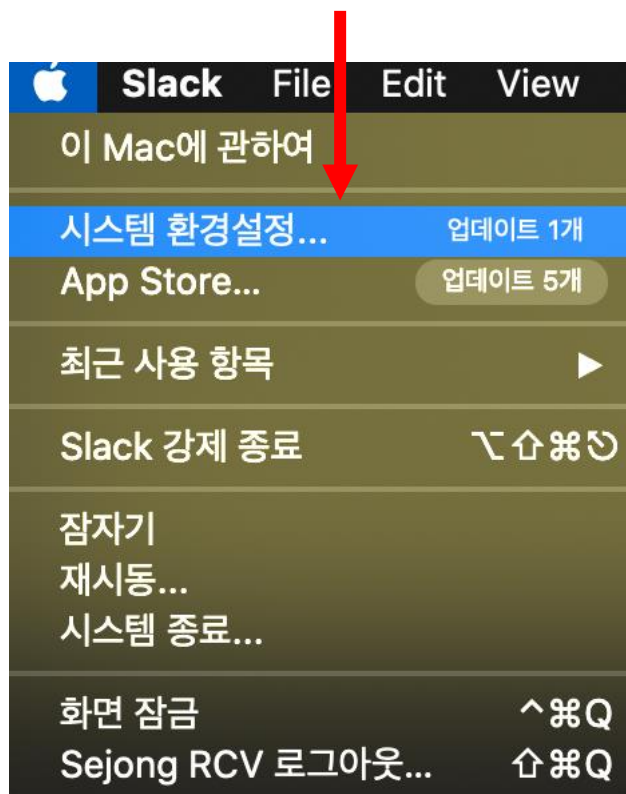
텍스트, 앱 및 기타 항목의 크기 변경

150%(권장)

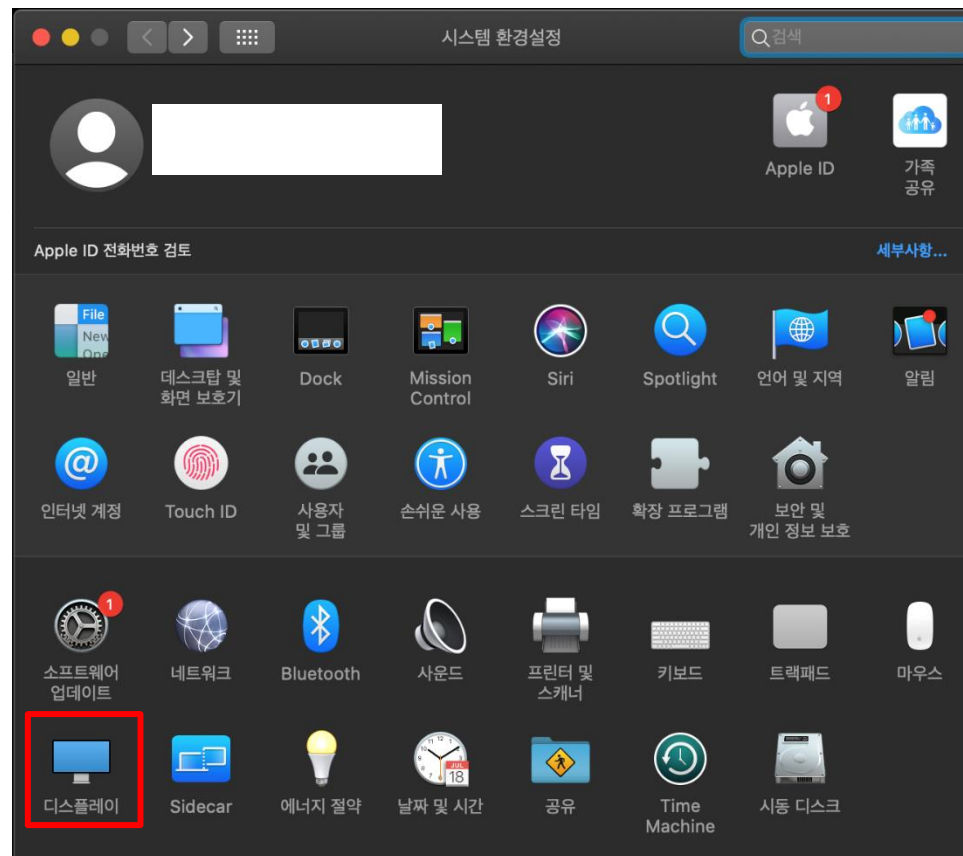
[고급 배율 설정](#)

듀얼모니터 확인 방법 CHECK (Mac)

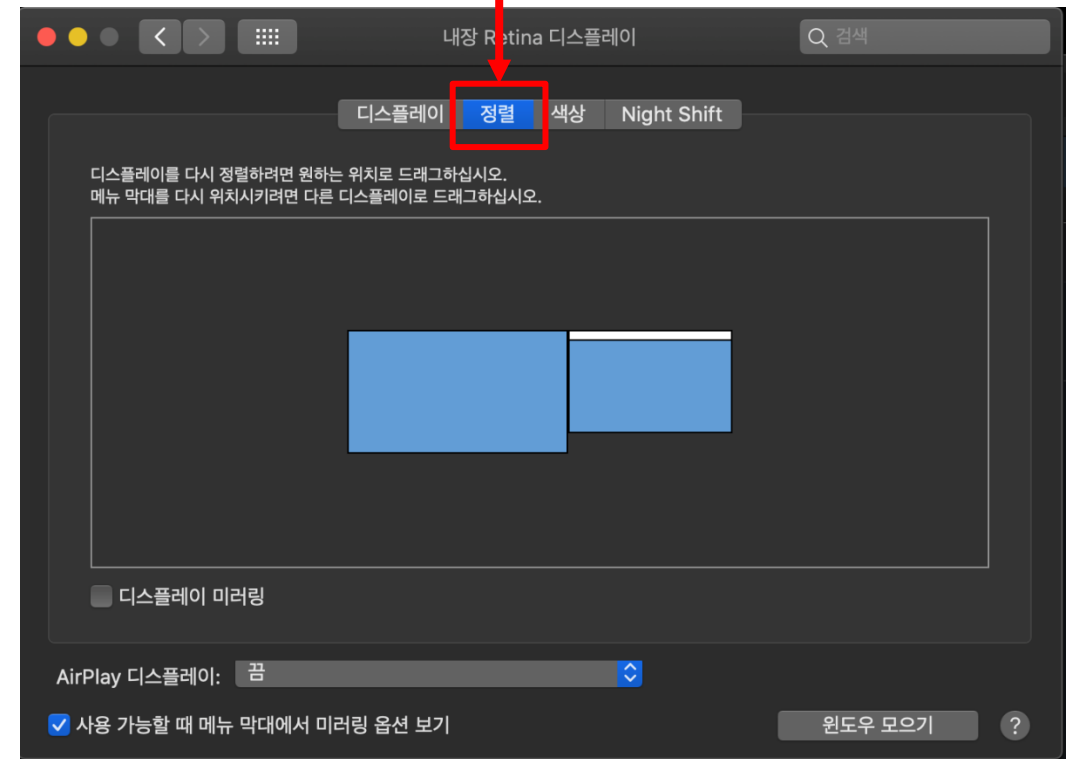
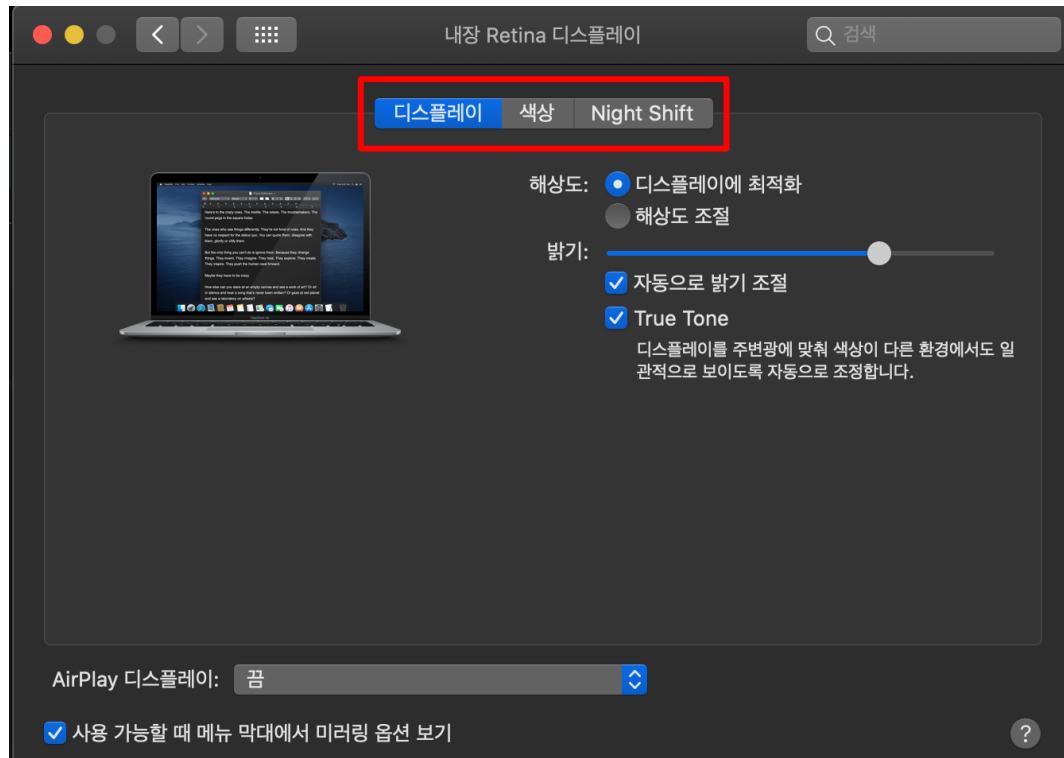
1. 맥북 화면 좌상단에 사과 클릭→ 시스템 환경 설정

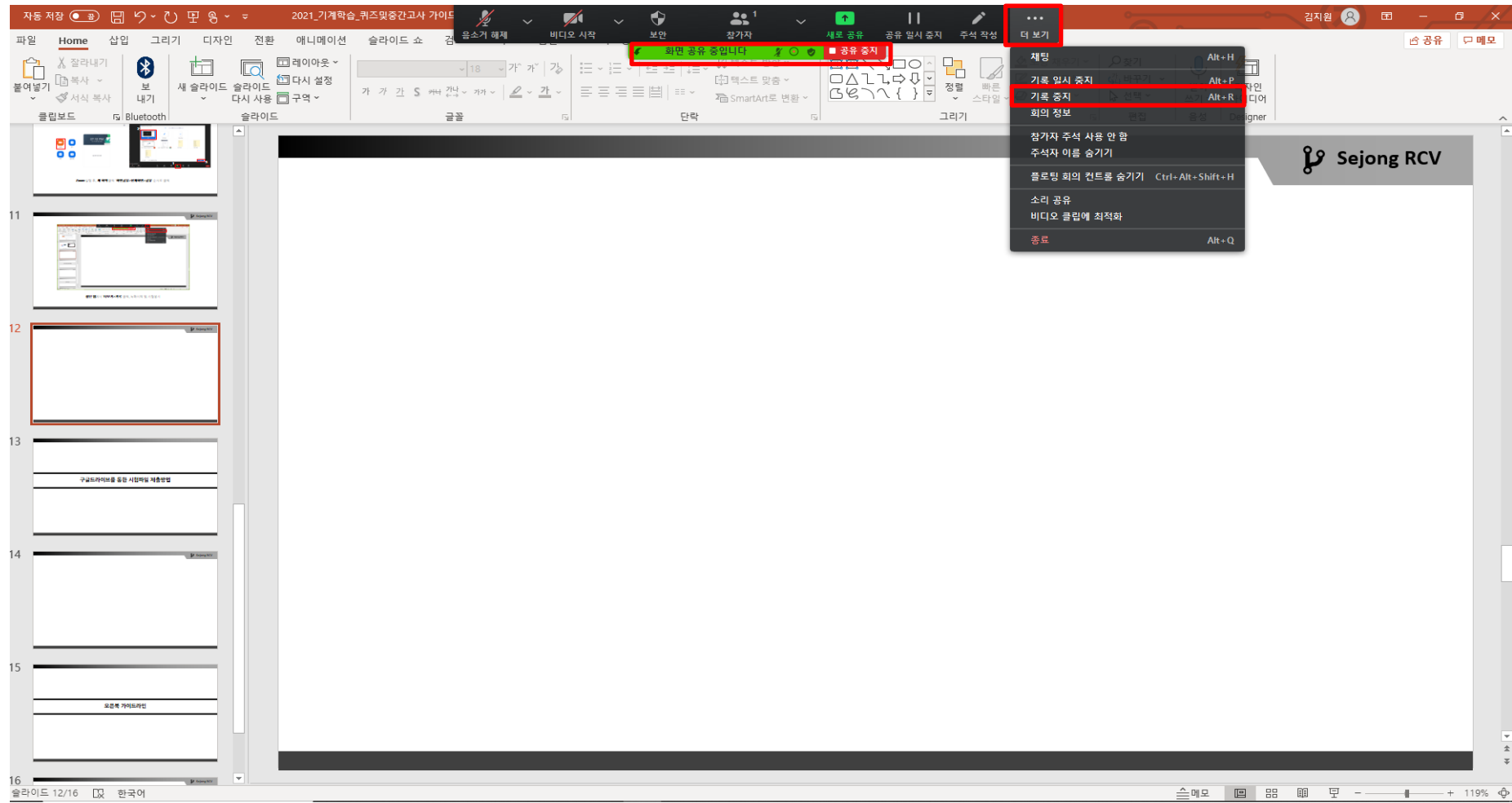


2. 시스템 환경 설정 중 디스플레이 클릭

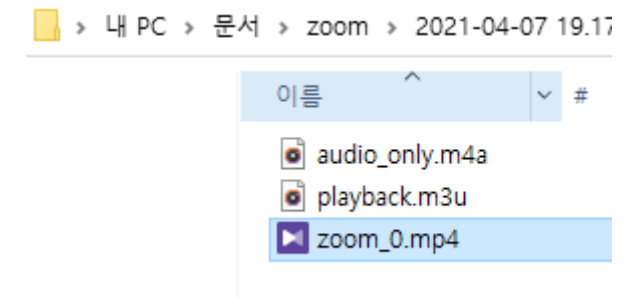
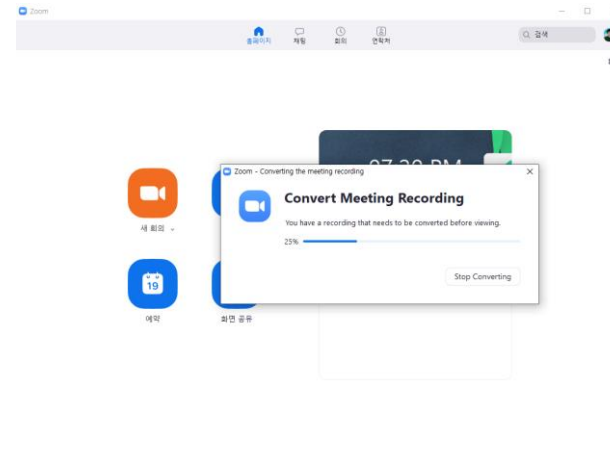
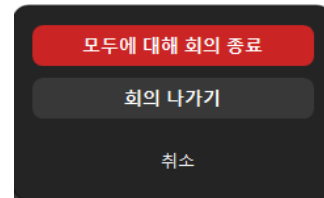
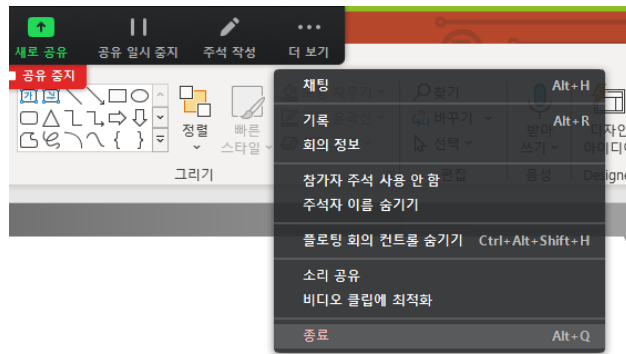


- 단일 모니터의 경우 좌측 그림과 같이 정렬창이 없음.
- 듀얼 모니터의 경우 우측 그림과 같이 정렬창이 존재.





시험이 종료된 이후 상단 탭에서 '더보기>기록중지' 클릭



기록중지를 클릭한 이후, '더보기>종료>모두에 대해 회의 종료' 클릭 후 녹화된 영상을 컨버팅한다.
(자동으로 진행) 이후 컨버팅된 .mp4파일 제출

오픈북 가이드라인

1. 기존 코드를 참고하는 행위 절대 안됨

- 화면 녹화상으로 확인하여 문제시 * F F F F F F F *

2. 오픈 북 아님

- 수업 자료 참고 불가

3. 오픈 매뉴얼 검색은 가능

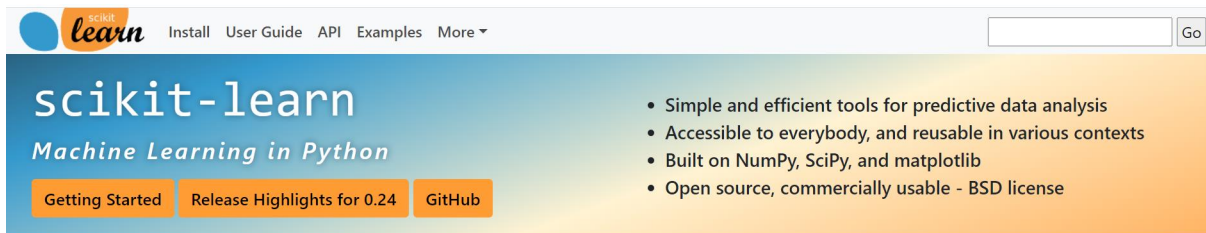
- 공식 라이브러리 사이트에서 사용방법 *검색 가능*

* 공식 라이브러리에서 검색 가능*

- 공식 라이브러리 사이트 외의 사이트에서 검색 발견 시 F+ 학사경고

<https://scikit-learn.org/>

Scikit-learn



scikit-learn
Machine Learning in Python

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

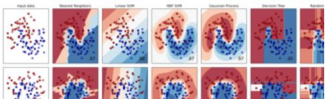
Getting Started | Release Highlights for 0.24 | GitHub

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

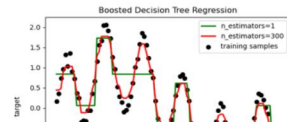


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...

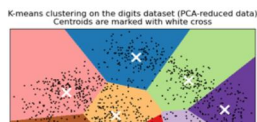


Clustering

Automatic grouping of similar objects into sets.

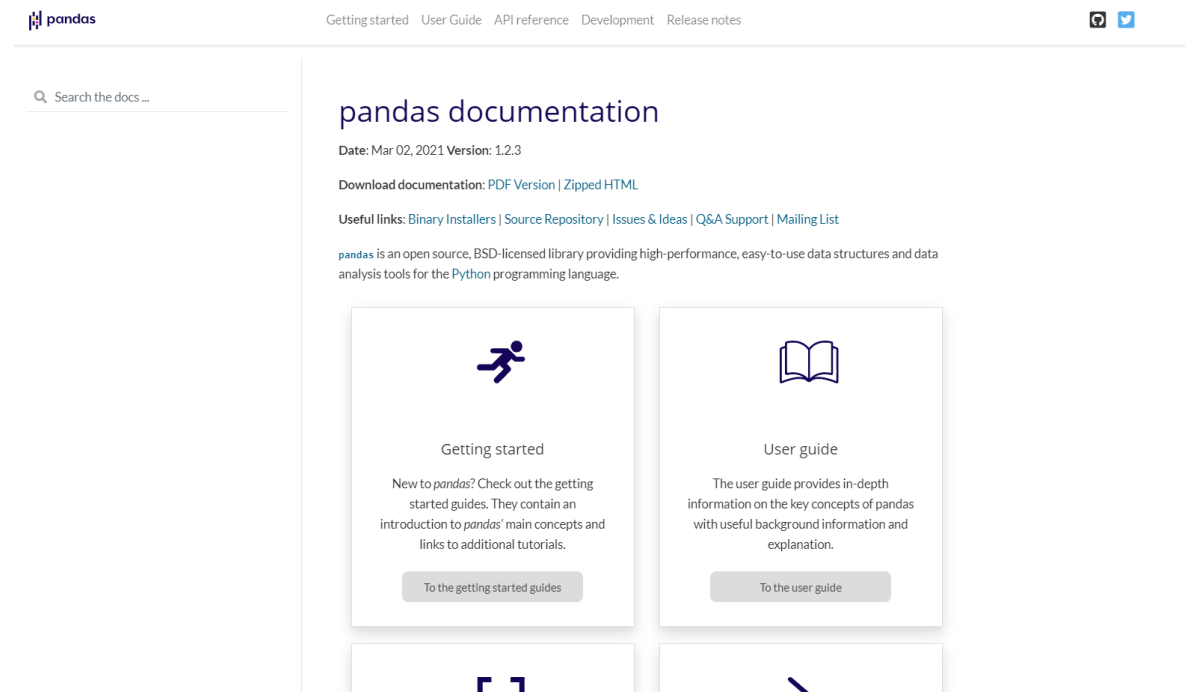
Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



<https://pandas.pydata.org/docs/>

Pandas



pandas documentation

Date: Mar 02, 2021 Version: 1.2.3

Download documentation: [PDF Version](#) | [Zipped HTML](#)

Useful links: [Binary Installers](#) | [Source Repository](#) | [Issues & Ideas](#) | [Q&A Support](#) | [Mailing List](#)

pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

Getting started

New to pandas? Check out the getting started guides. They contain an introduction to pandas' main concepts and links to additional tutorials.

[To the getting started guides](#)

User guide


The user guide provides in-depth information on the key concepts of pandas with useful background information and explanation.

[To the user guide](#)

... etc

Example

원하는 내용 검색



scikit-learn

Machine Learning in Python

Getting Started Release Highlights for 0.24 GitHub

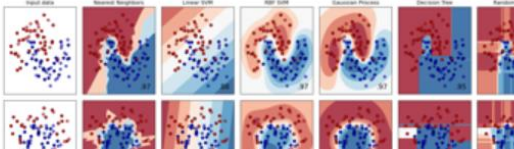
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

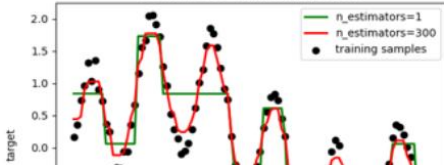


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...




Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



learn

Install User Guide API Examples More ▾

Prev Up Next

scikit-learn 0.24.1
Other versions


Please cite us if you use the software.

Search

Searching for multiple words only shows matches that contain all words

Searching...

- [sklearn.preprocessing.StandardScaler](#)
- [Compare the effect of different scalers on data with outliers](#)
...e to download the full example code or to run this example in your browser via Binder Compare the effect of different scalers on data with outliers Feature 0 (median income in a block) and feature 5 (number of households) of the Californi...
- [sklearn.linear_model.Ridge](#)
...d before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use StandardScaler before calling fit on an estimator with normalize=False. copy_X=bool, default=True If True, X will be copied; else...
- [sklearn.linear_model.ridge_regression](#)
...' fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing. All last five solvers support both dense and sparse data. However, only 'sag' and 's...'
- [sklearn.linear_model.RidgeClassifier](#)
...d before regression by subtracting the mean and dividing by the l2-norm. If you wish to standardize, please use



scikit-learn 0.24.1
Other versions

Please [cite us](#) if you use the software.

sklearn.preprocessing.StandardScaler

Examples using
sklearn.preprocessing.StandardScaler

파라미터 정보확인

Parameters:

copy : *bool, default=True*

If False, try to avoid a copy and do inplace scaling instead. This is not guaranteed to always work inplace; e.g. if the data is not a NumPy array or scipy.sparse CSR matrix, a copy may still be returned.

with_mean : *bool, default=True*

If True, center the data before scaling. This does not work (and will raise an exception) when attempted on sparse matrices, because centering them entails building a dense matrix which in common use cases is likely to be too large to fit in memory.

with_std : *bool, default=True*

If True, scale the data to unit variance (or equivalently, unit standard deviation).

Attributes:

scale_ : *ndarray of shape (n_features,) or None*

Per feature relative scaling of the data to achieve zero mean and unit variance. Generally this is calculated using `np.sqrt(var_)`. If a variance is zero, we can't achieve unit variance, and the data is left as-is, giving a scaling factor of 1. `scale_` is equal to `None` when `with_std=False`.

New in version 0.17: scale_

mean_ : *ndarray of shape (n_features,) or None*

The mean value for each feature in the training set. Equal to `None` when `with_mean=False`.

var_ : *ndarray of shape (n_features,) or None*

The variance for each feature in the training set. Used to compute `scale_`. Equal to `None` when `with_std=False`.

n_samples_seen_ : *int or ndarray of shape (n_features,)*

The number of samples processed by the estimator for each feature. If there are no missing samples, the `n_samples_seen` will be an integer, otherwise it will be an array of dtype int. If `sample_weights` are used it will be a float (if no missing data) or an array of dtype float that sums the weights seen so far. Will be reset on new calls to fit, but increments across `partial_fit` calls.

Toggle Menu

scikit-learn

Install User Guide API Examples More ▾

Prev Up Next

scikit-learn 0.24.1
Other versions

Please cite us if you use the software.

sklearn.preprocessing.StandardScaler

Examples using sklearn.preprocessing.StandardScaler

sklearn.preprocessing.StandardScaler(*, copy=True, with_mean=True, with_std=True)

[source]

Standardize features by removing the mean and scaling to unit variance

The standard score of a sample x is calculated as:

$$z = (x - u) / s$$

where u is the mean of the training samples or zero if `with_mean=False`, and s is the standard deviation of the training samples or one if `with_std=False`.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using `transform`.

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

For instance many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered around 0 and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

This scaler can also be applied to sparse CSR or CSC matrices by passing `with_mean=False` to avoid breaking the sparsity structure

scikit-learn

Install User Guide API Examples More ▾

Prev Up Next

scikit-learn 0.24.1
Other versions

Please cite us if you use the software.

sklearn.preprocessing.StandardScaler

Examples using sklearn.preprocessing.StandardScaler

NaNs are treated as missing values: disregarded in fit, and maintained in transform.

We use a biased estimator for the standard deviation, equivalent to `numpy.std(x, ddof=0)`. Note that the choice of `ddof` is unlikely to affect model performance.

For a comparison of the different scalers, transformers, and normalizers, see [examples/preprocessing/plot_all_scaling.py](#).

Examples

```
>>> from sklearn.preprocessing import StandardScaler
>>> data = [[0, 0], [0, 0], [1, 1], [1, 1]]
>>> scaler = StandardScaler()
>>> print(scaler.fit(data))
StandardScaler()
>>> print(scaler.mean_)
[0.5 0.5]
>>> print(scaler.transform(data))
[[-1. -1.]
 [-1. -1.]
 [ 1.  1.]
 [ 1.  1.]]
>>> print(scaler.transform([[2, 2]]))
[[3.  3.]]
```

Methods

<code>fit(X[, y, sample_weight])</code>	Compute the mean and std to be used for later scaling.
<code>fit_transform(X[, y])</code>	Fit to data, then transform it.
<code>get_params([deep])</code>	Get parameters for this estimator.

아래로 스크롤

Example 확인

라이브러리 공식 메뉴얼에서 나오는 내용 전부 확인 **가능**

공식 메뉴얼 외 사이트에서 검색 **불가**