

구분	함수	메소드 설명
1day	.tail()	DataFrame 또는 Series 객체의 끝에서부터 지정된 수의 행을 반환합니다. 최신데이터셋을 확인할때 유용한데 데이터 크기가 큰 경우에 전체 데이터를 로드하지 않고도 빠른 데이터 검토를 할수 있도록 도와 줍니다.
1day	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
1day	.isna()	메소드는 DataFrame 또는 Series 객체에서 결측치(누락된 값)를 찾는 데 사용됩니다. 이 메소드는 데이터에서 각 요소가 결측치인지 아닌지를 나타내는 조건이 참이면 True, 거짓이면 False를 반환 값으로 이루어진 객체를 반환합니다.
1day	.dropna()	DataFrame 또는 Series에서 결측치(누락된 데이터)를 제거하는 데 사용됩니다.
1day	.fillna()	DataFrame 또는 Series 객체 내의 결측치(누락된 데이터)를 채우는 데 사용됩니다.
1day	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
1day	.sort_values()	DataFrame 또는 Series 내의 값에 따라 데이터를 정렬하는 데 사용됩니다. 데이터를 오름차순 또는 내림차순으로 정렬할 수 있으며, 복수의 열을 기준으로 정렬하는 것도 가능합니다. -by: 정렬 기준이 되는 열의 이름(또는 열의 리스트). -axis: 정렬을 수행할 축. 0은 행을, 1은 열을 의미합니다. -ascending: 오름차순 여부. True는 오름차순, False는 내림차순 정렬을 의미합니다. -inplace: True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다. -kind: 정렬 알고리즘 선택 (예: 'quicksort', 'mergesort', 'heapsort'). -na_position: 결측치의 위치. 'first'는 결측치를 앞에 배치, 'last'는 뒤에 배치합니다.
1day	.sort_values().tail()	먼저 데이터프레임을 특정 기준에 따라 정렬한 후, 정렬된 데이터프레임의 마지막 몇 개의 행을 반환합니다
1day	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
1day	.duplicated()	DataFrame 또는 Series에서 중복된 행을 찾습니다
1day	.drop_duplicates()	DataFrame 또는 Series에서 중복된 행을 제거합니다.
1day	.replace()	특정 값을 다른 값으로 대체 <예시> 데이터프레임의 'A' 열에서 'apple'을 'orange'로 바꿉니다.
1day	.map()	시리즈의 각 요소를 지정된 함수나 매핑에 따라 변환 데이터프레임의 'A' 열에서 'apple'을 'fruit'로, 'carrot'을 'vegetable'로 매핑합니다.
1day	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
1day	.lower()	문자열을 소문자로 변경 합니다.
1day	.upper()	문자열을 대문자로 변경 합니다.
1day	str.split(expand=true)	pandas의 Series나 DataFrame 객체에 있는 문자열을 분할할 때 사용됩니다. expand=True 옵션은 분할된 각 요소를 별도의 열로 확장하여 DataFrame 형태로 반환하는데 사용됩니다. 기본적으로 expand=False로 설정되어 있으며, 이 경우 결과는 Series의 리스트 형태로 반환됩니다.
1day	.astype()	메서드를 사용하여 특정 열(시리즈)의 데이터 타입을 변경할 수 있습니다.

1day	.isdigit()	문자열에 있는 모든 문자가 숫자인지 확인하고, 그 결과를 참(True) 또는 거짓(False) 값으로 반환합니다.
1day	pd.to_numeric()	문자열이나 다른 형식의 데이터를 숫자형 데이터로 변환하는 데 사용됩니다.
1day 프로젝트	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
1day 프로젝트	.duplicated()	DataFrame 또는 Series에서 중복된 행을 찾습니다
1day 프로젝트	.drop_duplicates()	DataFrame 또는 Series에서 중복된 행을 제거합니다.
1day 프로젝트	.isna()	메소드는 DataFrame 또는 Series 객체에서 결측치(누락된 값)를 찾는 데 사용됩니다. 이 메소드는 데이터에서 각 요소가 결측치인지 아닌지를 나타내는 조건이 참이면 True, 거짓이면 False를 반환 값으로 이루어진 객체를 반환합니다.
1day 프로젝트	isna().sum()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 총 개수 를 계산합니다.
1day 프로젝트	isna().mean()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 비율을 계산합니다.
1day 프로젝트	.dropna()	DataFrame 또는 Series에서 결측치(누락된 데이터)를 제거하는 데 사용됩니다.
1day 프로젝트	.sort_values()	DataFrame 또는 Series 내의 값에 따라 데이터를 정렬하는 데 사용됩니다. 데이터를 오름차순 또는 내림차순으로 정렬할 수 있으며, 복수의 열을 기준으로 정렬하는 것도 가능합니다. -by: 정렬 기준이 되는 열의 이름(또는 열의 리스트). -axis: 정렬을 수행할 축. 0은 행을, 1은 열을 의미합니다. -ascending: 오름차순 여부. True는 오름차순, False는 내림차순 정렬을 의미합니다. -inplace: True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다. -kind: 정렬 알고리즘 선택 (예: 'quicksort', 'mergesort', 'heapsort'). -na_position: 결측치의 위치. 'first'는 결측치를 앞에 배치, 'last'는 뒤에 배치합니다.
1day 프로젝트	len()	주어진 컨테이너 내의 요소들의 총 개수를 알려주는 데 사용됩니다.
1day 프로젝트	.iloc[]	"integer location"(정수위치)의 약자로, 데이터프레임이나 시리즈에서 정수형 위치 인덱스를 사용하여 데이터를 선택합니다
1day 프로젝트	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
1day 프로젝트	.unique()	데이터셋에서 중복을 제거하고 유니크한(고유한) 값들의 배열을 반환하는 기능을 수행합니다.
1day 프로젝트	.nunique()	데이터셋 또는 DataFrame의 특정 열에서 고유(중복되지 않는) 요소의 개수를 세는 것입니다.
1day 프로젝트	.value_counts()	범주형 데이터를 요약하는 데이터 분석 작업에 자주 사용됩니다.
1day 프로젝트	.replace()	특정 값을 다른 값으로 대체 <예시> 데이터프레임의 'A' 열에서 'apple'을 'orange'로 바꿉니다.
1day 프로젝트	.split()	앞 뒤 불필요한 공백을 제거 합니다.
2day	.concat()	여러 데이터프레임이나 시리즈를 하나로 병합하는 기능을 제공합니다. 전처리 과정에서 서로 다른 데이터 소스를 하나로 합치거나, 관련된 데이터를 연결할 때 매우 유용합니다.
2day	.merge()	두 개 이상의 데이터프레임을 특정 공통 열 또는 인덱스를 기준으로 병합하는 데 사용됩니다. SQL의 JOIN과 유사한 기능을 제공합니다. pandas에서는 merge 함수를 사용하여 SQL 스타일의 조인을 수행할 수 있습니다.

2day	join	pandas에서는 merge 함수를 사용하여 SQL 스타일의 조인을 수행할 수 있습니다. -Inner Join: 두 DataFrame에서 공통 키를 가진 행만 반환합니다. *merge의 기본설 -Left Join: 원쪽 DataFrame의 모든 행과 오른쪽 DataFrame에서 일치하는 행을 반환합니다. -Right Join: 오른쪽 DataFrame의 모든 행과 원쪽 DataFrame에서 일치하는 행을 반환합니다. -Outer Join: 두 DataFrame의 모든 행을 반환합니다. 일치하는 행이 없는 경우 해당 위치는 NaN 값으로 채워집니다.
2day	.unique()	데이터셋에서 중복을 제거하고 유니크한(고유한) 값들의 배열을 반환하는 기능을 수행합니다.
2day	.replace()	특정 값을 다른 값으로 대체 <예시> 데이터프레임의 'A' 열에서 'apple'을 'orange'로 바꿉니다.
2day	.rename()	데이터 분석 과정에서 데이터의 의미를 명확히 하거나, 다른 데이터 세트와의 일관성을 유지하기 위해 사용됩니다.
2day	pd.to_numeric()	문자열이나 다른 형식의 데이터를 숫자형 데이터로 변환하는 데 사용됩니다.
2day	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
2day	.isna().sum()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 총 개수를 계산합니다.
2day	.isna().mean()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 비율을 계산합니다.
2day	.sort_values()	DataFrame 또는 Series 내의 값에 따라 데이터를 정렬하는 데 사용됩니다. 데이터를 오름차순 또는 내림차순으로 정렬할 수 있으며, 복수의 열을 기준으로 정렬하는 것도 가능합니다. -by: 정렬 기준이 되는 열의 이름(또는 열의 리스트). -axis: 정렬을 수행할 축. 0은 행을, 1은 열을 의미합니다. -ascending: 오름차순 여부. True는 오름차순, False는 내림차순 정렬을 의미합니다. -inplace: True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다. -kind: 정렬 알고리즘 선택 (예: 'quicksort', 'mergesort', 'heapsort'). -na_position: 결측치의 위치. 'first'는 결측치를 앞에 배치, 'last'는 뒤에 배치합니다.
2day	.reset_index()	새로운 연속된 정수 인덱스를 생성하고, 기존 인덱스를 데이터 프레임의 열로 변환할 수 있습니다. 데이터 분석 과정에서 인덱스를 재정렬하거나 조작한 후 원래의 인덱스 순서를 복구하고자 할 때 유용합니다.
2day	set_option	데이터 처리 및 출력 시의 다양한 옵션을 커스터마이즈할 수 있습니다. 예를 들어, 데이터 프레임의 출력 시 표시되는 최대 행 수나 열 수, 출력 포맷 등을 조정할 수 있습니다.
2day	.iloc[]	<lms 개념 안내 링크> 1. 로그란? : https://youtu.be/Z5myJ8dg_rM 2. 지수와 로그의 관계 : https://youtu.be/Obch1OP5QyA
2day	.get_dummies()	주어진 범주형 열의 각 고유 범주를 대표하는 새로운 이진(0 또는 1) 열을 생성합니다. drop_first: True로 설정하면, 각 범주형 열에서 첫 번째 범주를 제외합니다. 이는 다중 공선성 문제를 피하는 데 도움이 됩니다. (한 변수의 변동이 다른 변수와 비슷한 패턴을 보이는 경우, 이를 다중 공선성이라고 합니다. 이러한 상황은 회귀 모델의 해석과 예측의 정확성에 영향을 줄 수 있습니다.)

2day	.nunique()	데이터셋 또는 DataFrame의 특정 열에서 고유(중복되지 않는) 요소의 개수를 세는 것입니다.
2day	.value_counts()	범주형 데이터를 요약하는 데이터 분석 작업에 자주 사용됩니다.
2day	.tail()	DataFrame 또는 Series 객체의 끝에서부터 지정된 수의 행을 반환합니다. 최신데이터셋을 확인할때 유용한데 데이터 크기가 큰 경우에 전체 데이터를 로드하지 않고도 빠른 데이터 검토를 할수 있도록 도와 줍니다.
2day	.merge	두 개 이상의 데이터프레임을 특정 공통 열 또는 인덱스를 기준으로 병합하는 데 사용됩니다. SQL의 JOIN과 유사한 기능을 제공합니다. pandas에서는 merge 함수를 사용하여 SQL 스타일의 조인을 수행할 수 있습니다. -Inner Join: 두 DataFrame에서 공통 키를 가진 행만 반환합니다. *merge의 기본설정 -Outer Join: 두 DataFrame의 모든 행을 반환합니다. 일치하는 행이 없는 경우 해당 위치는 NaN 값으로 채워집니다.
2day	.quantile()	통계학에서 데이터 세트를 동일한 크기의 연속 구간으로 나누는 데 사용되는 값입니다. 데이터 세트의 특정 백분위수를 나타내는 위치 값으로 생각할 수 있습니다. 예를 들어, 50번째 백분위수(또는 중앙값)는 데이터 세트를 절반으로 나누는 값입니다.
2day	round()	파이썬에서 숫자를 특정 소수점 자리에서 반올림하는 데 사용되는 내장 함수입니다.
2day	pca.explained_variance_ratio_	주로 주성분 분석(Principal Component Analysis, PCA)과 같은 차원 축소 기법에서 사용되는 속성입니다. 이 속성은 각 주성분이 원본 데이터의 분산(variance)을 얼마나 설명하는지에 대한 비율을 나타냅니다.
2day	.corr()	데이터프레임 내의 모든 변수 간의 상관관계를 계산합니다. 상관관계는 두 변수 간의 선형 관계의 강도와 방향을 나타내는 통계적 지표입니다.
2day 프로젝트	str.split()	파이썬에서 문자열을 분할하는 데 사용되는 메소드입니다. 이 메소드는 주어진 구분자(delimiter)에 따라 문자열을 여러 부분으로 나누고, 결과를 문자열 리스트로 반환합니다
2day 프로젝트	str.upper()	파이썬의 문자열 메소드 중 하나로, 문자열에 포함된 모든 문자를 대문자로 변환하는 기능을 수행합니다
2day 프로젝트	.merge()	두 개 이상의 데이터프레임을 특정 공통 열 또는 인덱스를 기준으로 병합하는 데 사용됩니다. SQL의 JOIN과 유사한 기능을 제공합니다. pandas에서는 merge 함수를 사용하여 SQL 스타일의 조인을 수행할 수 있습니다.
2day 프로젝트	.inplace()	True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다.
2day 프로젝트	.rename()	데이터 분석 과정에서 데이터의 의미를 명확히 하거나, 다른 데이터 세트와의 일관성을 유지하기 위해 사용됩니다.
2day 프로젝트	.str.split()	파이썬에서 문자열을 분할하는 데 사용되는 메소드입니다. 이 메소드는 주어진 구분자(delimiter)에 따라 문자열을 여러 부분으로 나누고, 결과를 문자열 리스트로 반환합니다
2day 프로젝트	pd.to_numeric()	문자열이나 다른 형식의 데이터를 숫자형 데이터로 변환하는 데 사용됩니다.
2day 프로젝트	.isna.mean()	결측치의 비율을 계산할 때 일반적으로 사용됩니다.
2day 프로젝트	.unique()	데이터셋에서 중복을 제거하고 유니크한(고유한) 값들의 배열을 반환하는 기능을 수행합니다.
2day 프로젝트	.groupby.mean()	서로 다른 범주의 데이터를 그룹별로 나누어 각 그룹의 평균 값을 얻기 위해 사용됩니다.
2day 프로젝트	.fillna()	DataFrame 또는 Series 객체 내의 결측치(누락된 데이터)를 채우는 데 사용됩니다.
2day 프로젝트	.isna().sum()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 총 개수를 계산합니다.

2day 프로젝트	.sort_values()	DataFrame 또는 Series 내의 값에 따라 데이터를 정렬하는 데 사용됩니다. 데이터를 오름차순 또는 내림차순으로 정렬할 수 있으며, 복수의 열을 기준으로 정렬하는 것도 가능합니다. -by: 정렬 기준이 되는 열의 이름(또는 열의 리스트). -axis: 정렬을 수행할 축. 0은 행을, 1은 열을 의미합니다. -ascending: 오름차순 여부. True는 오름차순, False는 내림차순 정렬을 의미합니다. -inplace: True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다. -kind: 정렬 알고리즘 선택 (예: 'quicksort', 'mergesort', 'heapsort'). -na_position: 결측치의 위치. 'first'는 결측치를 앞에 배치, 'last'는 뒤에 배치합니다.
2day 프로젝트	.groupby().agg()	데이터를 그룹화하고 이러한 그룹에 하나 이상의 연산을 적용할 수 있습니다.
2day 프로젝트	.nunique()	데이터셋 또는 DataFrame의 특정 열에서 고유(중복되지 않는) 요소의 개수를 세는 것입니다.
2day 프로젝트	pca.explained_variance_ratio_.sum()	explained_variance_ratio_ 배열에 있는 모든 값을 합산합니다.
2day 프로젝트	.corr()	데이터프레임 내의 모든 변수 간의 상관관계를 계산합니다. 상관관계는 두 변수 간의 선형 관계의 강도와 방향을 나타내는 통계적 지표입니다.
3day	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
3day	.startswith()	프로그래밍 언어들에서 문자열이 특정 문자 또는 문자열로 시작하는지 여부를 확인하는 데 사용됩니다.
3day	.endswith()	프로그래밍 언어에서 문자열이 특정 문자 또는 문자열로 끝나는지 여부를 확인하는 데 사용됩니다.
3day	.replace()	특정 값을 다른 값으로 대체 <예시> 데이터프레임의 'A' 열에서 'apple'을 'orange'로 바꿉니다.
3day	.rename()	데이터 분석 과정에서 데이터의 의미를 명확히 하거나, 다른 데이터 세트와의 일관성을 유지하기 위해 사용됩니다.
3day	.isna().mean()	데이터 분석에서 자주 사용되는 코드 조합으로 데이터 프레임이나 시리즈 내의 결측치(NA 또는 NaN 값)의 비율을 계산합니다.
3day	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
3day	.sort_values()	DataFrame 또는 Series 내의 값에 따라 데이터를 정렬하는 데 사용됩니다. 데이터를 오름차순 또는 내림차순으로 정렬할 수 있으며, 복수의 열을 기준으로 정렬하는 것도 가능합니다. -by: 정렬 기준이 되는 열의 이름(또는 열의 리스트). -axis: 정렬을 수행할 축. 0은 행을, 1은 열을 의미합니다. -ascending: 오름차순 여부. True는 오름차순, False는 내림차순 정렬을 의미합니다. -inplace: True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다. -kind: 정렬 알고리즘 선택 (예: 'quicksort', 'mergesort', 'heapsort'). -na_position: 결측치의 위치. 'first'는 결측치를 앞에 배치, 'last'는 뒤에 배치합니다.
3day	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
3day	abs()	주어진 숫자의 절대값을 반환합니다.
3day	.value_counts()	범주형 데이터를 요약하는 데이터 분석 작업에 자주 사용됩니다.

3day	rating	일반적으로 평가, 등급 또는 순위를 나타내는데 사용됩니다.
3day	for in zip	zip 함수는 여러 시퀀스의 요소를 하나씩 묶어서 튜플로 반환하며, for 루프를 사용하여 이 튜플들을 순차적으로 처리할 수 있습니다.
3day 프로젝트	pd.set_option('display.max_columns', 5)	이 코드는 pandas가 데이터를 화면에 출력할 때 최대 50개의 열을 보여주도록 설정하는 것입니다. 기본적으로 pandas는 출력할 때 더 적은 수의 열을 보여주도록 설정되어 있기 때문에, 이 설정을 변경하면 데이터프레임의 더 많은 열을 한눈에 볼 수 있습니다.
3day 프로젝트	.describe()	DataFrame 또는 Series 객체에 대한 요약 통계를 제공합니다. 이 메소드는 데이터의 중심 경향, 분포 및 형태 등을 빠르게 이해하는 데 도움이 됩니다.
3day 프로젝트	.nunique()	데이터셋 또는 DataFrame의 특정 열에서 고유(중복되지 않는) 요소의 개수를 세는 것입니다.
3day 프로젝트	.groupby().agg().reset_index()	데이터를 그룹화하고 집계한 후, 인덱스를 다시 설정하는 데 사용됩니다.
3day 프로젝트	.value_counts()	범주형 데이터를 요약하는 데이터 분석 작업에 자주 사용됩니다.
3day 프로젝트	.merge()	두 개 이상의 데이터프레임을 특정 공통 열 또는 인덱스를 기준으로 병합하는 데 사용됩니다. SQL의 JOIN과 유사한 기능을 제공합니다. pandas에서는 merge 함수를 사용하여 SQL 스타일의 조인을 수행할 수 있습니다.
3day 프로젝트	.inplace()	True로 설정하면 원본 DataFrame이 변경되고, False (기본값)이면 새로운 DataFrame 객체가 반환됩니다.
3day 프로젝트	.apply()	함수를 데이터프레임의 열 또는 행에 적용 합니다. <예시> 데이터프레임의 모든 수치 데이터에 2를 곱합니다.
3day 프로젝트	.loc[].sum()	데이터프레임의 특정 행이나 열을 선택한 후 그 선택된 부분의 합을 계산하는 데 사용됩니다.