

03 데이터 분석 기초 - 판다스

AI 에이전트 개발

데이터 분석 기초

원티드캡

- [소개](#)
- [1. Series](#)
 - [1\) Series](#)
 - [2\) DataFrame](#)
- [2. 주요 기능 정리](#)
- [3. 외부 데이터 처리](#)
 - [1\) CSV 파일](#)
 - [2\) Excel 파일](#)
 - [3\) SPSS 파일](#)

소개



- 데이터 분석에 관련된 기능을 제공하는 파이썬 라이브러리
- 데이터 분석, 가공, 시각화 전처리 등에서 매우 많이 사용됨
- 외부 데이터(csv, txt, excel 등)의 처리가 용이함
- Numpy 기반으로 만들어져 수치 계산이 빠름

```
uv add pandas
```



1. Series

1) Series

- 1차원 배열로, 데이터와 인덱스를 가지며 리스트와 유사함
- 각 데이터는 고유한 인덱스를 가지고 있어 접근 및 조작이 쉬움

2) DataFrame

- 2차원 배열로, 행과 열로 구성된 테이블 형태의 데이터 구조
- 각 열은 Series 객체로 구성되며, 다양한 데이터 타입을 가질 수 있음

2. 주요 기능 정리

분류	메서드 / 속성	설명
생성	pd.Series(data)	1차원 시리즈 생성 (리스트, 배열 등)

분류	메서드 / 속성	설명
	<code>pd.DataFrame(data)</code>	2차원 데이터프레임 생성 (딕셔너리, 리스트 등)
조회	<code>.head()</code> , <code>.tail()</code>	처음/끝에서 5개 데이터 조회
검색	<code>.loc[]</code> , <code>.iloc[]</code>	라벨 기반 / 번호 기반 데이터 조회
속성 조회	<code>.shape</code> , <code>.size</code>	(행, 열) 구조 및 전체 원소 개수 확인
	<code>.index</code> , <code>.columns</code>	인덱스, 열 이름 확인 (<code>columns</code> 는 DataFrame 전용)
	<code>.dtypes</code> , <code>.dtype</code>	각 열 또는 전체의 데이터 타입 확인
	<code>.values</code>	실제 값만 NumPy 배열로 추출
편집	<code>.drop()</code> , <code>.rename()</code>	행/열 삭제, 이름 변경
	<code>.astype()</code>	데이터 타입 변경
	<code>.fillna()</code> , <code>.dropna()</code>	결측치 채우기 / 제거
통계	<code>.mean()</code> , <code>.sum()</code>	평균, 합계 등 요약 통계
	<code>.value_counts()</code>	값 빈도수 세기 (Series 전용, 범주형 분석에 자주 사용)
	<code>.describe()</code>	전체 요약 통계 제공
	<code>.groupby()</code> , <code>.agg()</code>	그룹별 집계 / 사용자 지정 통계 (DataFrame 전용)
정렬	<code>.sort_values()</code>	값 기준 정렬(내림차순, <code>ascending=False</code>)
	<code>.sort_index()</code>	인덱스 기준 정렬

3. 외부 데이터 처리

1) CSV 파일

불러오기

```
df = pd.read_csv("파일명.csv", 옵션들...)
```

매개변수	설명
encoding	문자 인코딩 (예: "utf-8", "cp949")
sep	구분자 (기본: ",", 탭 파일이면 "\t")
header	헤더 행 지정 (데이터에 헤더가 없으면 <code>header=None</code> 으로 설정)
names	열 이름 수동 지정 (헤더 없음과 함께 사용)
usecols	사용할 열만 선택 (["이름", "나이"])
na_values	결측치로 처리할 값 지정 (["NA", "-", ""])
skiprows	앞에서 건너뛸 행 수 / 리스트

저장하기

```
df.to_csv("저장경로.csv", 옵션들...)
```

매개변수	설명
index	인덱스를 저장할지 여부 (False 권장)
encoding	인코딩 지정 ("utf-8-sig": Excel 한글 대응)

2) Excel 파일

```
uv add openpyxl
```

불러오기

```
df = pd.read_excel("파일명.xlsx", 옵션들...)
```

매개변수	설명
sheet_name	읽을 시트 이름 (기본은 첫 시트)
header	헤더 행 지정
usecols	사용할 열만 선택
na_values	결측치 처리 값 지정

저장하기

```
df.to_excel("저장경로.xlsx", 옵션들...)
```

매개변수	설명
index	인덱스 저장 여부
sheet_name	시트 이름 지정

3) SPSS 파일

```
uv add pyreadstat
```

불러오기

```
df = pd.read_spss("data/mydata.sav")
```