[베이직 반] 판다스 (1)

📎 실습 데이터

https://drive.google.com/file/d/1hZtwoWg4OuaM3rrImcYZIyE68SW50Bi0/view?usp=sharing

---

데이터 전처리(판다스)

1) 데이터 전처리가 뭐고, 왜 중요한데?

- 모델/분석 전에 **쓸 수 있는 상태**로 만드는 과정: 결측·이상치·타입·중복·스케일·인덱스 정리
- 잘한 전처리 = **노이즈↓, 해석력/성능/재현성↑**

---

2) 판다스는 뭐고, 어떻게 설치/확인?

- 표 형태의 데이터 처리 라이브러리(엑셀과 유사), 핵심 객체: **Series(1D), DataFrame(2D)**

```
pip install pandas
```

```
import pandas as pd
pd.__version__
```

---

3) 판다스 주요 데이터 타입

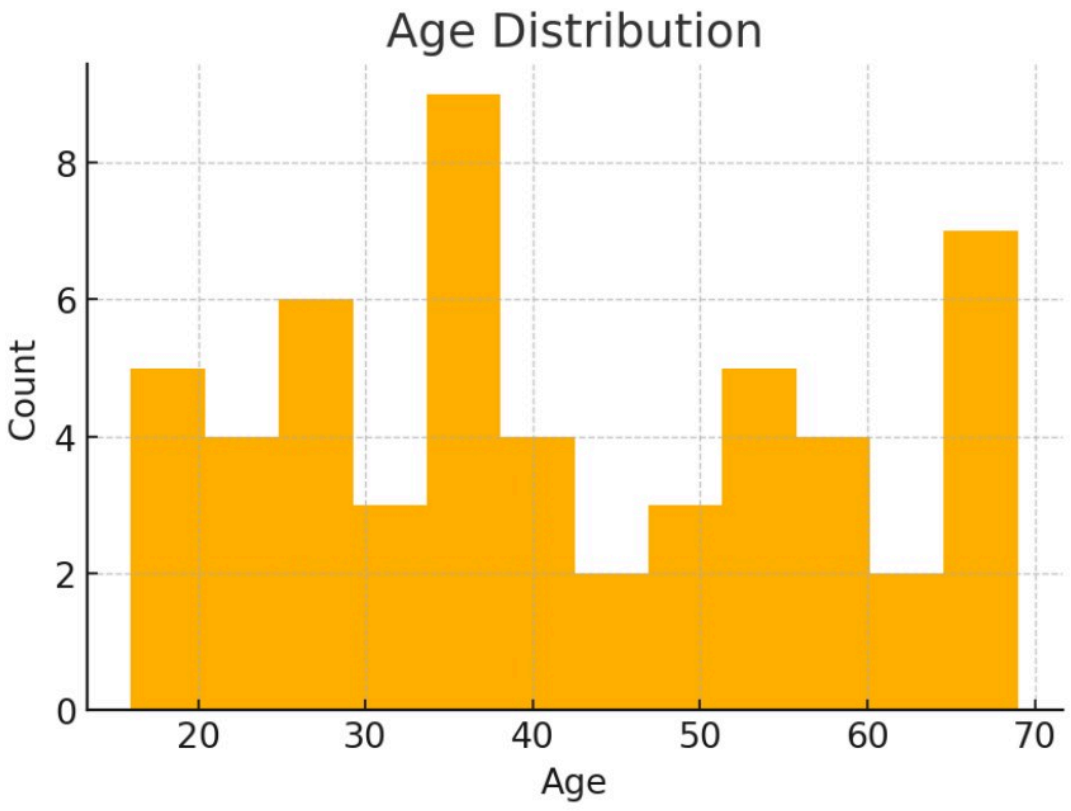
▼ 데이터 타입 정리

타입	예	핵심 포인트	유용한 도구
수치형	Age, Fare	분포/이상치 확인	`describe()`, 히스토그램
범주형	Sex, Class	값 종류·비율	`value_counts()`
불리언	IsAdult	필터링·마스킹 핵심	`df[cond]`
날짜/시각	BoardedDate	`datetime` 변환 후 시계열 연산	`to_datetime`
결측값	NaN	패턴 파악 후 전략적 처리	`isnull().sum()`, 대체/삭제

1. 수치형 변수

- 해당 데이터들이 정수형 혹은 실수형인 변수
- `int64`, `float64`

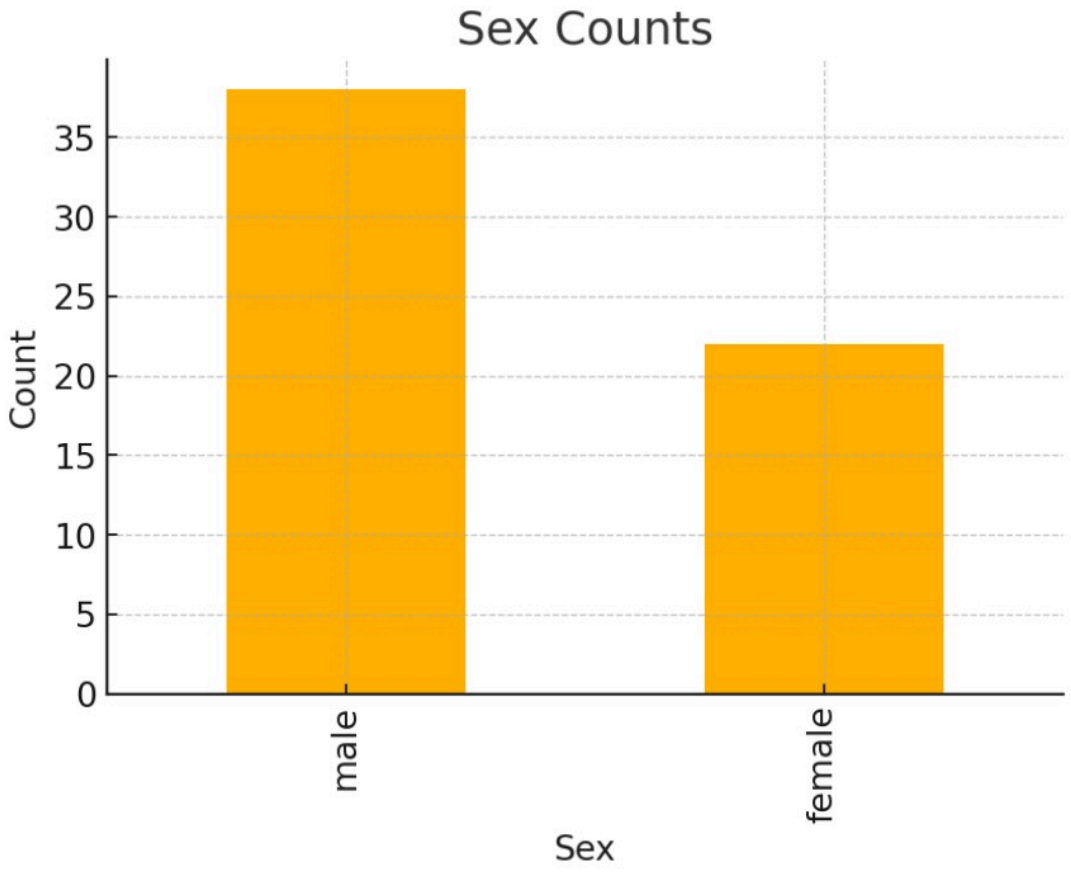
▼ 수치형 변수 그림 예시



2. 범주형 변수

- 해당 데이터들이 문자열, 혹은 혼합된 데이터(숫자+문자 등)인 변수
- `object`, `category`

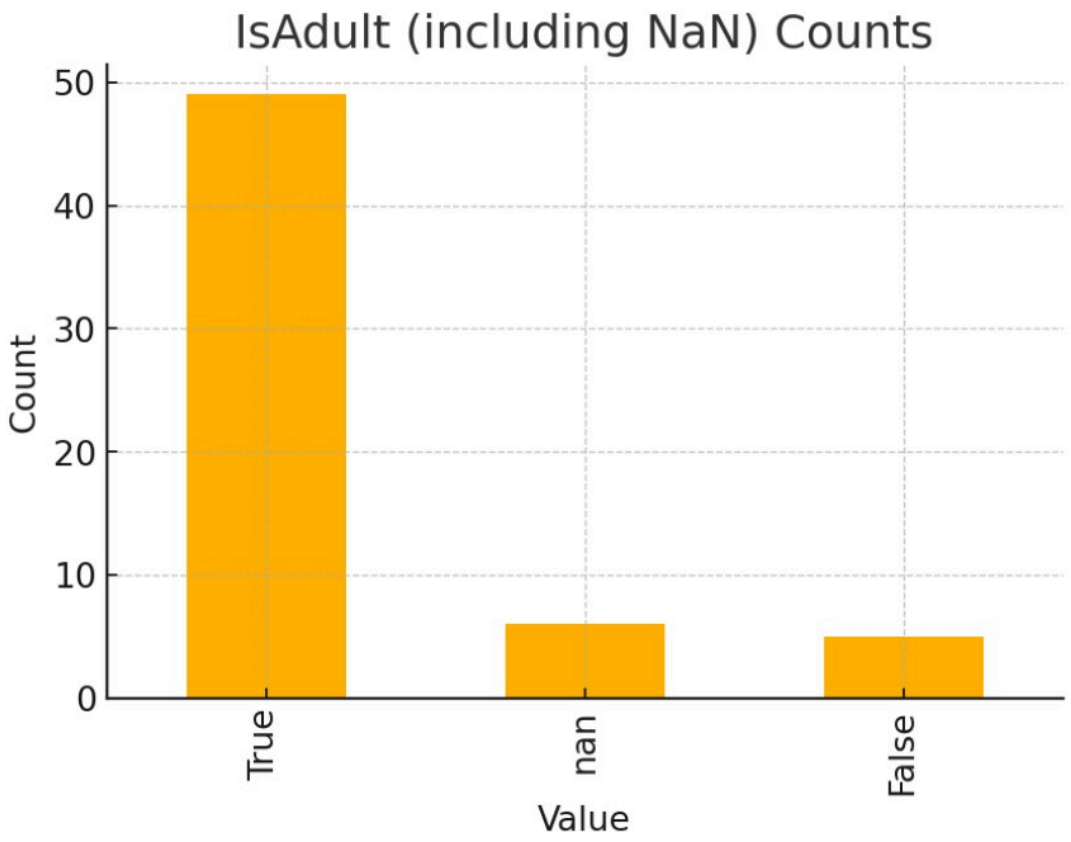
▼ 범주형 변수 그림 예시



3. 불린형 변수

- 해당 데이터들이 True(참), False(거짓)으로 이뤄진 변수
- `bool`

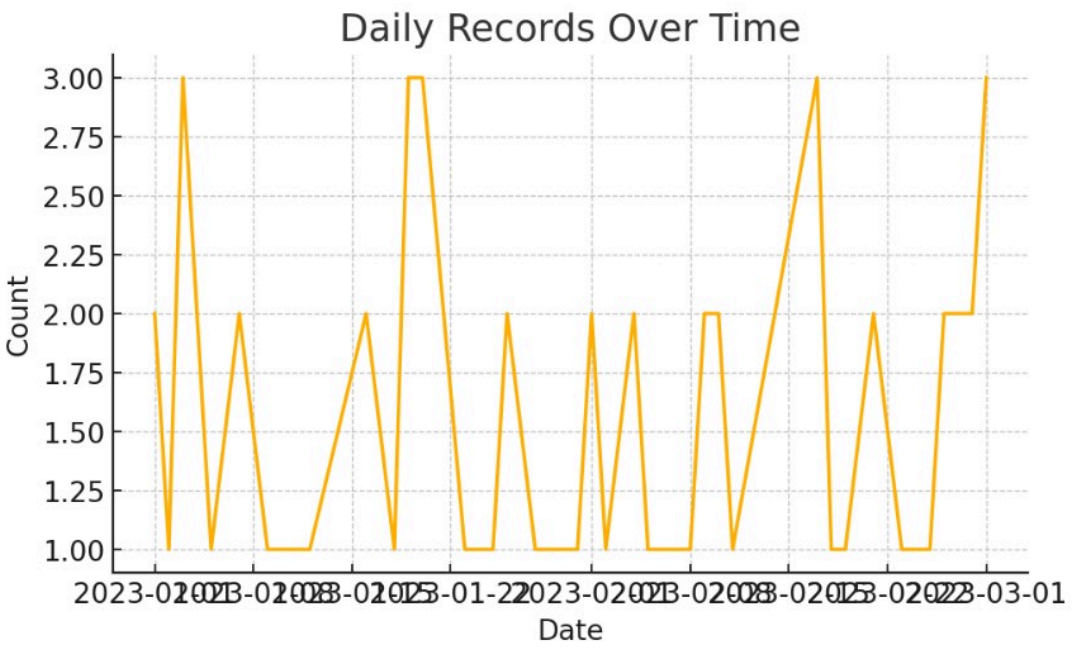
▼ 불린형 변수 그림 예시



4. 날짜/시간형 변수

- 해당 데이터들이 날짜/시간 데이터 (년, 월, 일, 시, 분, 초) 형태인 변수
- 일반적으로 object 타입으로 되어있어서, 날짜/시간형 변수로 형태 변환(`pd.to_datetime`)이 필요
- `datetime64[ns]`

▼ 날짜/시간형 변수 그림 예시



5. 결측치

- 값이 비어 있거나 데이터가 존재하지 않는 경우
- `NaN`

QUIZ.

	나이	성별	IsAdult	Boarded Date	요금	Class
1	20	Male	True	2025-09-16	635.2$	FIRST
2	27	Female	True	2025-09-14	325.4$	BUSINESS
3	13	Male		2025-09-13	122.7$	ECONOMY
4	8	Male	False	2025-09-15		FIRST
5	54	Female	True	2025-09-11	325.4$	BUSINESS

---

4) 시리즈(Series)가 뭔데?

- **값(values) + 인덱스(index)**가 있는 1차원 자료. DataFrame의 **한 컬럼**이 Series.
- 직접 생성 시 `pd.Series`

```
s = pd.Series([10, 20, 30], index=['a','b','c'], name='score')
print(s.index) # 시리즈의 인덱스를 확인
```

```
print(s.values) # 시리즈의 값을 확인
```

5) 데이터프레임(DataFrame)이 뭔데?

- **행(row)/열(column)이 있는 2차원 표**, 여러 Series의 모음
- 직접 생성 시 `pd.DataFrame`

```
df = pd.DataFrame({'name':['Kim','Lee'], 'age':[20,22]})
print(df.shape) # 해당 데이터의 행(데이터 수)과 열(컬럼 수)
print(df.head())  # 데이터프레임의 첫 5행을 출력(간략히 보기)
print(df.info())  # 각 컬럼의 타입, null 개수 등 정보 출력
print(df.describe()) # 수치형 컬럼들의 통계 요약
print(df.value_counts()) # 데이터프레임의 값 분포 확인
```

6) 외부 데이터를 어떻게 불러오고, 저장하는데?

1. 불러오기(.csv)

- `read_csv`로 시작 → `pd.read_csv('파일명')` → 파일 명은 확장자 포함 문자열('') 형태로!
- `csv` 파일 뿐만 아니라, `excel` / `json` 다양한 형태 불러오고 저장 가능

```
df = pd.read_csv('train.csv')
df
```

2. 저장하기(.csv)

- `데이터프레임.to_csv` 로 시작 → 전처리가 끝난 데이터프레임 변수 명을 적고 `.to_csv('저장할 파일명)` → 저장할 파일 명은 확장자 포함 문자열('') 형태로!

```
df.to_csv('preprocessed_train.csv')
```

7) 데이터 인덱싱과 슬라이싱은 뭔데?

1. 컬럼(열) 고르기

- 한 컬럼만 `df['Age']`는 **Series,** 리스트(`df[['PassengerId','Survived','Pclass','Sex','Age']]`)로 넘기면 **DataFrame**

```
print(df['Age'])
print(df[['PassengerId','Survived','Pclass','Sex','Age']].head()
```

2. 로우(행) 고르기 - 인덱스

- **라벨 기반(.loc):** .loc[row_label, col_label]

```
df.loc[2, 'Name']
```

  - **인덱스 값이 2인 행, 'Name' 열**의 데이터를 가져와라
  - 열 부분은 생략 가능 → 전체 열 선택
- **정수 위치(.iloc):** .iloc[row_idx, col_idx]

```
df.iloc[2, 3]
```

  - **인덱스 번호가 2인 행, 인덱스 번호가 3인 열**의 데이터를 가져와라
  - 열 부분은 생략 가능 → 전체 열 선택

3. 로우(행) 고르기 - 슬라이싱

- **라벨 기반(.loc):** .loc[row_label:row_label, col_label:col_label] - 끝(슬라이싱 end) 포함

```
df.loc[0:5, ['PassengerId','Name','Sex','Age']]
```

  - **인덱스 값이 0부터 5(포함)인 행, 컬럼명이 'PassengerId','Name','Sex','Age'인 열**의 데이터를 가져와라
  - 열 부분은 생략 가능 → 전체 열 선택
- **정수 위치(.iloc):** .iloc[row_idx:row_idx, col_idx:col_idx] - 끝(슬라이싱 end) 제외

```
df.iloc[0:6, [0,3,4,5]]
```

  - **인덱스 번호가 0부터 6(불포함 → 즉 5까지)인 행, 인덱스 번호가 0/3/4/5 → 'PassengerId','Name','Sex','Age'인 열**의 데이터를 가져와라
  - 열 부분은 생략 가능 → 전체 열 선택

4. 불리언 필터 (맛보기)

```
adults = df.loc[df['Age'] >= 18, ['PassengerId','Sex','Age','Survived']]
adults
```

- **인덱스 값이 Age가 18이상인 행, 컬럼명이 'PassengerId','Sex','Age','Survived'인 열**의 데이터를 가져와라

5. 복합 조건 (맛보기)

- `&` : AND → 모든 조건이 True이면 True(=조건에 만족한다)
- `|` : OR → 여러 조건 중 한 조건만 True이면 True(=조건에 만족한다)
- `&`와 `|` 사용 시 **괄호 필수**

```
mask = (df['Pclass']==1) & (df['Sex']=='female') & (df['Age']<30)
df.loc[mask, ['PassengerId','Pclass','Sex','Age','Survived']].head(10)
```

- **인덱스 값이 Pclass가 1이고, Sex가 Female이고 Age가 30미만인 행, 컬럼명이 'PassengerId','Pclass','Sex','Age','Survived'인 열**의 데이터를 가져와라

[데이터와 실습 파일은 같은 폴더/디렉토리에서 과제를 수행하시는 것을 권장드립니다!]

📎 미니 과제

1. H&M
    a. 데이터 : https://drive.google.com/file/d/1QhBueLd6STJpID3SKxN9nFFE5JfEHwhE/view?usp=drive_link
    b. 실습 파일 : https://drive.google.com/file/d/1oNTaRkqB-5Tg5q7Scg3CTN1cNAwb5tYp/view?usp=drive_link
    c. 정답 : https://drive.google.com/file/d/1QYz-kcBjuu7i0kP3QE1BaQLfL98-yL7r/view?usp=drive_link
2. 서울시 부동산 데이터
    a. 데이터 : https://drive.google.com/file/d/13tFuo7b6NLKnQtqVXE43bYWXhlAym9Za/view?usp=drive_link
    b. 실습 파일 : https://drive.google.com/file/d/1uKKLlS1WaQekfHGsY8hvMt-OF1Nrx3Km/view?usp=drive_link
    c. 정답 : https://drive.google.com/file/d/1RkHx0L9KI9HZ096IJIybVGm5FURJXxd9/view?usp=sharing

---