데이터 다루기 (2)

Pandas를 활용한 데이터 통계 입문

GOAL

- Pandas로 주요 기초 통계를 계산할 수 있다.
- 데이터의 분포를 요약해 이해할 수 있다.
- 숫자형/범주형 변수의 차이를 구분할 수 있다.

목차

숫자형 vs 범주형 변수

구분	예시	설명
숫자형(Numerical)	혈당, 나이, BMI	수치로 계산 가능
범주형(Categorical)	성별, Outcome(당뇨 여부)	그룹이나 종류 구분용

비유로 이해하기

- 숫자형: **줄자로 잴 수 있는 값** (키, 체중)
- 범주형: **라벨이 붙은 값** (남/여, 예/아니오)

> 둘 다 중요하지만 “계산 가능한가?”가 기준!

숫자형 변수의 예시

변수명	설명
Glucose	혈당 수치
BMI	체질량 지수
Age	나이

이 값들은 평균, 중앙값, 표준편차를 계산할 수 있다

왜 범주형 변수가 중요한가?

- 현실 데이터의 다수는 **라벨/종류** 형태다: 성별, 지역, 직업, 병력, 결혼여부 등
- 숫자형만으로는 담기 어려운 **질적 차이**를 제공한다.
- 모델이 학습해야 하는 **결정 경계**를 풍부하게 만든다.
- 예: Titanic의 `sex`, `embarked`, `class`는 생존률에 큰 차이를 만든다.

숫자로 바꾸는 근본 이유

- 선형모델, SVM, k-NN, 신경망 등은 **벡터 계산**이 핵심
- 범주형을 그대로 두면 **거리/내적**이 정의되지 않음
- 따라서 **인코딩 → 수치 벡터화**가 필요

잘못된 인코딩의 위험

- 임의의 정수 부여(Label Encoding)를 선형모델에 사용하면 **가짜 순서/거리**가 생긴다.
- 예: `embarked: C=0, Q=1, S=2` → 모델이 S > Q > C 로 해석
- 트리 계열은 덜 민감하지만, 여전히 **분할 기준**에 영향을 준다.

범주형 변수의 예시

변수명	설명
Outcome	0: 비당뇨, 1: 당뇨

- 이 변수는 평균을 계산할 수 없고,
  고유값 개수(`nunique()`), 빈도(`value_counts()`)로 요약

예제 데이터 미리보기

```
import pandas as pd

df = pd.read_csv("diabetes.csv")
print(df.head())
```

Pregnancies	Glucose	BloodPressure	BMI	Age	Outcome
2	138	62	33.6	42	1
1	85	66	26.6	31	0

표준편차(std)

> 값들이 평균으로부터 얼마나 퍼져 있는지를 나타냄

```
df["Glucose"].std()
```

- 값이 클수록 데이터가 흩어져 있음
- 값이 작을수록 데이터가 비슷비슷함

비유로 이해하기

- 표준편차는 **“학생들의 성적 차이”**
  - 모두 90점이면 → std 작음 (비슷함)
  - 어떤 학생 100점, 어떤 학생 40점 → std 큼 (차이 큼)

데이터 요약 describe()

```
df.describe()
```

통계량	의미
count	데이터 개수
mean	평균
std	표준편차
min/max	최소/최대값
25%, 50%, 75%	사분위수

describe() 활용 팁

- `include='all'` : 범주형까지 모두 표시
- `df.describe(include='object')` : 문자열 변수만 요약

실습 1: 기본 통계 구하기

```
mean_glucose = df["Glucose"].mean()
median_glucose = df["Glucose"].median()
std_glucose = df["Glucose"].std()

print(mean_glucose, median_glucose, std_glucose)
```

실습 2: Outcome 빈도 확인

```
df["Outcome"].value_counts()
```

Outcome	Count
0	500
1	268

실습 3: 범주형 변수 고유값 확인

```
df["Outcome"].nunique()
```

👉 2 (0 또는 1)

실습 4: describe() 직접 적용

```
df.describe()
```

결과 예시:

| count | mean | std | min | 25% | 50% | 75% | max |

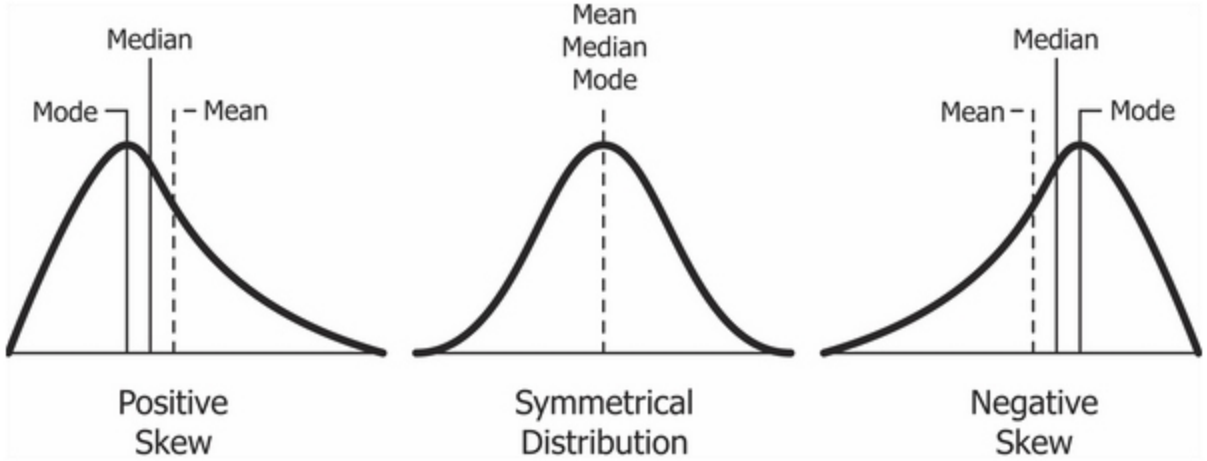
시각화로 분포 보기

```
import seaborn as sns
sns.histplot(df["Glucose"], kde=True)
```

> 히스토그램으로 데이터 분포를 확인하면 평균이 왜곡되어 있는지도 볼 수 있습니다.

평균과 중앙값의 관계

분포 형태	평균 vs 중앙값
정규분포	거의 같음
오른쪽으로 긴 꼬리	평균 > 중앙값
왼쪽으로 긴 꼬리	평균 < 중앙값



통계로 보는 당뇨병 데이터

변수	해석 예시
Glucose	평균 120 이상이면 고혈당 경향
BMI	평균 30 이상 → 비만 가능성
Age	50대 이상일수록 위험도 상승

직접 계산

```
glucose_values = df["Glucose"].dropna()

mean = sum(glucose_values) / len(glucose_values)
variance = sum((glucose_values - mean)**2) / len(glucose_values)
std = variance ** 0.5

print(mean, std)
```

👉 Pandas가 해주는 계산을 직접 구현해보기!

통계의 핵심 요약

- 평균: 데이터의 중심
- 중앙값: 극단값에 덜 민감한 중심
- 표준편차: 데이터의 다양성

정리 요약

개념	설명
평균(mean)	전체 데이터의 중심
중앙값(median)	데이터 중간 위치
표준편차(std)	퍼져 있는 정도
describe()	데이터 전체 요약
value_counts()	범주형 빈도 계산