단변량 분석 (Univariate Analysis)

- **수치형 변수 (Numerical Variable) 분석**:
  - **기술 통계량**: `describe()` 메서드를 사용하여 평균(mean), 중앙값(median), 표준편차(std), 최소/최대값, 사분위수 등을 계산하여 데이터의 중심 경향성과 퍼진 정도를 파악합니다.
  - **분포 시각화**: 히스토그램(Histogram)이나 커널 밀도 추정(Kernel Density Estimate, KDE) 플롯을 사용하여 데이터의 분포 형태(정규분포, 치우친 분포 등)를 시각적으로 확인합니다.
- **범주형 변수 (Categorical Variable) 분석**:
  - **빈도 분석**: `value_counts()` 메서드를 사용하여 각 카테고리(범주)에 속하는 데이터의 개수(빈도)와 비율을 계산합니다.
  - **시각화**: 바 차트(Bar Chart)나 파이 차트(Pie Chart)를 사용하여 각 카테고리의 빈도를 시각적으로 비교합니다.

적용 가능한 상황

- **데이터 초기 탐색**: 데이터셋을 처음 접했을 때, 각 변수가 어떤 특성을 가지고 있는지 개별적으로 파악하는 첫 단계에서 필수적으로 수행됩니다.
- **데이터 품질 검증**: 각 변수의 분포나 빈도를 확인하여 예상치 못한 값이나 이상치를 발견하고, 데이터 정제의 필요성을 판단할 때.
- **피처 엔지니어링 아이디어 도출**: 변수의 분포가 특정 방향으로 심하게 치우쳐 있다면 로그 변환 등의 변수 변환을 고려하거나, 특정 카테고리의 빈도가 너무 낮다면 다른 카테고리와 통합하는 등의 아이디어를 얻을 수 있습니다.

예제 데이터 생성

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Seaborn 내장 'titanic' 데이터셋 사용
df = sns.load_dataset('titanic')
'''
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
```

```
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
'''
```

수치형 변수 분석

- **분석 대상**: `age`, `fare` (나이, 운임 요금)
- **용도**: 데이터의 중심 경향, 산포도, 분포 형태를 파악합니다.
- **코드 예시**:

```
# 1-1. 기술 통계량 확인
print("--- Descriptive Statistics for 'age' ---")
print(df['age'].describe())

# 1-2. 분포 시각화 (히스토그램 + KDE)
plt.figure(figsize=(10, 5))
sns.distplot(df['age'].dropna(), kde=True, bins=30)
plt.title('Distribution of Age')
plt.axvline(df['age'].mean(), color='red', linestyle='--', label=f"Mean:
{df['age'].mean():.2f}")
plt.axvline(df['age'].median(), color='green', linestyle='-',
label=f"Median: {df['age'].median():.2f}")
plt.legend()
plt.show()

# 1-3. 박스 플롯 (이상치 확인)
plt.figure(figsize=(8, 5))
sns.boxplot(df['fare'])
plt.title("BoxPlot of Fare")
plt.show()
```

- **결과 해석**:
  - **기술 통계량**: `age` 변수의 평균은 약 29.7세, 중앙값은 28세이며, 최소 0.42세부터 최대 80세까지 분포합니다. `count`가 전체 데이터 수보다 적으므로 결측치가 있음을 알 수 있습니다.
  - **시각화**: 히스토그램을 보면, 20-30대 승객이 가장 많고, 어린 아이들도 일부 포함되어 있으며, 전체적으로 약간 오른쪽으로 치우친 분포를 보입니다. 평균(빨간 점선)이 중앙값(녹색 실선)보다 오른쪽에 있는 것을 통해 이를 다시 확인할 수 있습니다.
  - **박스 플롯**: 수염 길이를 IQR(3사분위수-1사분위수)로 보고, 수염의 양 끝은 IQR 값에 1.5를 곱한 최소, 최대 범위입니다. 일반적으로 해당 범위를 넘어가면 이상치로 보고 해당 데이터에 대한 처리를 결정하는 편입니다. 이 경우 이상치가 많아 그대로 유지할 수도 있고, `fare`가 유의미한 변수가 아

니라면 변수 자체를 제거할 수도 있습니다. 단, 변수를 그대로 사용하더라도 500이 넘어가는 극단점 같은 경우는 데이터를 제거하고 보는 것이 좋을 수도 있습니다.

2. 범주형 변수 분석

- **분석 대상**: `pclass`, `sex`, `embarked` (객실 등급, 성별, 탑승 항구)
- **용도**: 각 카테고리의 빈도와 비율을 파악하여 데이터의 구성을 이해합니다.
- **코드 예시**:

```
# 2-1. 빈도 분석 (normalize=True는 비율 출력)
print("--- Frequency of 'pclass' ---")
print(df['pclass'].value_counts())

print("\n--- Ratio of 'sex' ---")
print(df['sex'].value_counts(normalize=True))

# 2-2. 시각화 (바 차트)
plt.figure(figsize=(12, 5))

# pclass에 대한 countplot
plt.subplot(1, 2, 1)
sns.countplot(x='pclass', data=df, palette='viridis')
plt.title('Count of Passengers by Pclass')

# embarked에 대한 countplot
plt.subplot(1, 2, 2)
sns.countplot(x='embarked', data=df, palette='plasma')
plt.title('Count of Passengers by Embarked Port')

plt.tight_layout()
plt.show()
```

- **결과 해석**:
  - **빈도 분석**: `pclass`는 3등석 승객이 가장 많음을 알 수 있습니다. `sex`는 남성 승객이 약 64.8%를 차지함을 보여줍니다. `value_counts(normalize=True)`는 각 카테고리의 비율을 계산해줍니다.
  - **시각화**: 바 차트(countplot)를 통해 각 카테고리의 빈도를 직관적으로 비교할 수 있습니다. 3등석(pclass=3) 승객이 압도적으로 많고, Southampton(embarked='S')에서 탑승한 승객이 가장 많다는 사실을 쉽게 파악할 수 있습니다.
  - 시각화로 확인 시, 파이 차트를 그려서 비율을 비교할 수도 있습니다. (`seaborn`에서 파이 차트는 제공되지 않아서 해당 분석에서는 생략)

장단점 및 대안

분석 방법	장점	단점	대안/보완

분석 방법	장점	단점	대안/보완
수치형 분석	데이터의 분포, 중심 경향, 이상치 등 핵심적인 통계 정보를 정량적으로 파악할 수 있음.	단일 숫자(평균, 중앙값 등)만으로는 데이터의 전체적인 모습을 오해할 수 있음.	**시각화(히스토그램, 박스 플롯)와 병행**: 기술 통계량으로 요약된 정보를 시각화를 통해 확인하여 데이터 분포에 대한 완전한 이해를 돕습니다. **왜도(Skewness)와 첨도(Kurtosis)**: `df['col'].skew()`, `df['col'].kurt()`를 통해 분포의 비대칭성과 뾰족한 정도를 정량적으로 측정할 수 있습니다.
범주형 분석	데이터의 구성을 간단하고 명확하게 파악할 수 있음.	변수 간의 관계는 전혀 알 수 없음.	**파이 차트(Pie Chart)**: 비율을 시각화하는 데 사용될 수 있지만, 카테고리가 많아지면 가독성이 급격히 떨어져 바 차트가 더 권장됩니다.

단변량 분석은 각 변수에 대한 깊은 이해를 제공하지만, 변수들 사이의 상호작용이나 관계를 설명하지는 못합니다. 따라서 단변량 분석을 통해 각 변수의 특성을 파악한 후에는, 필연적으로 두 개 이상의 변수를 함께 분석하는 **이변량 분석(Bivariate Analysis)** 및 **다변량 분석(Multivariate Analysis)**으로 나아가야 합니다.

데이터 해석 시 주의사항

- 모드 : 그래프 그렸을 때, 나타나는 봉우리 → 정규분포는 1개의 모드
  - 모드가 여러개일 경우, 멀티 모달이라고 함
- 단순한 수치적 해석이 아니라, 비즈니스적 관점에서도 살펴봐야한다.
  - 이상치가 발생할 정도로 판매 금액이 이상한 지역이 있다.
    → 해당 회사의 가격 정책이 어떻게 되는지? 등을 전부 확인
  - 멀티 모달일 경우, 균등하게 분포되어 있다고 생각할 수도 있다.
    → 전체적으로 전부 필요로 한다고 해석 가능 → 모드 별로 나눠서 구분해서 해석할 수도 있다 → 모드 별로 구별되는 특징이 있을 확률이 크다