

T-검정 (t-test)

T-검정(t-test):

- 모집단의 분산이나 표준편차를 알지 못할 때, 표본 데이터(sample)를 이용하여 모집단의 평균(μ)이 특정 값과 같은지, 또는 두 집단의 평균이 서로 같은지를 검정하는 통계적 방법
- 표본의 크기가 작거나(일반적으로 $n < 30$) 모집단의 표준편차를 모를 때, 정규분포 대신 t-분포(t-distribution)를 사용하여 가설을 검정

t-분포의 특징:

- 정규분포와 같이 종 모양의 대칭적인 분포이지만, 양쪽 꼬리가 더 두꺼움
- 표본 크기가 작을 때의 불확실성을 반영
- **자유도(degree of freedom, df)**라는 모수에 따라 모양이 변하며, 자유도가 커질수록(즉, 표본 크기가 커질수록) 표준정규분포에 근사

T-검정의 종류:

1. 단일표본 T-검정 (One-sample t-test):

- 하나의 표본 그룹의 평균이 우리가 알고 있는 특정 기준값(모집단 평균)과 같은지를 검정
- $H_0: \mu = \mu_0$ (표본의 평균은 기준값 μ_0 와 같다)
- $H_1: \mu \neq \mu_0$ (표본의 평균은 기준값 μ_0 와 다르다)

2. 독립표본 T-검정 (Independent two-sample t-test):

- 서로 독립적인 두 표본 그룹의 평균이 같은지를 검정
- $H_0: \mu_1 = \mu_2$ (두 그룹의 평균은 같다)
- $H_1: \mu_1 \neq \mu_2$ (두 그룹의 평균은 다르다)
- 등분산성 가정에 따라 두 가지로 나뉩니다:
 - **Student's t-test:** 두 그룹의 분산이 같다고 가정. (등분산성 만족 시)
 - **Welch's t-test:** 두 그룹의 분산이 다르다고 가정. (등분산성 불만족 시, 더 안정적이고 일반적으로 권장됨)

3. 대응표본 T-검정 (Paired-samples t-test):

- 동일한 대상에 대해 어떤 처치(treatment) 전후의 값을 측정하여, 그 차이가 유의미한지를 검정
- 두 변수가 서로 독립이 아닌 쌍(pair)을 이루고 있을 때 사용
- $H_0: \mu_d = 0$ (대응되는 값들의 차이의 평균은 0이다)
- $H_1: \mu_d \neq 0$ (대응되는 값들의 차이의 평균은 0이 아니다)

T-검정의 기본 가정:

1. **독립성:** 각 관측치는 서로 독립적이어야 합니다. (대응표본 t-검정은 예외)
2. **정규성:** 데이터(또는 대응표본의 경우 차이값)는 정규분포를 따라야 합니다. 하지만 표본 크기가 충분히 크면($n \geq 30$) 중심 극한 정리에 의해 이 가정이 다소 완화될 수 있습니다.
3. **등분산성:** 독립표본 t-검정의 경우, 두 그룹의 분산이 동일해야 합니다. (Welch's t-test는 이 가정이 필요 없음)

1. 단일표본 T-검정 (One-sample t-test)

- **문제:** 어떤 공장에서 생산하는 과자 한 봉지의 평균 무게는 150g으로 알려져 있다. 새로 생산된 과자 25 봉지를 표본으로 뽑아 무게를 재어보니 평균 155g, 표준편차 10g이었다. 이 새로운 생산분의 평균 무게가 150g과 유의미하게 다른가? (유의수준 $\alpha=0.05$)
- **함수:** `stats.ttest_1samp(a, popmean)`

```
from scipy import stats
import numpy as np

# 표본 데이터 생성
np.random.seed(0)
sample_data = np.random.normal(loc=155, scale=10, size=25)

# 기준값 (모집단 평균)
pop_mean = 150

# 정규성 검정 (Shapiro-Wilk)
shapiro_stat, shapiro_p = stats.shapiro(sample_data)
print(f"Shapiro-Wilk p-value: {shapiro_p:.4f}") # 0.4065, 정규성 만족

# 단일표본 t-검정 수행
t_statistic, p_value = stats.ttest_1samp(sample_data, popmean=pop_mean)
print(f"\n--- One-sample t-test ---")
print(f"T-statistic: {t_statistic:.4f}")
print(f"P-value: {p_value:.4f}")

if p_value < 0.05:
    print("귀무가설 기각: 표본 평균은 150g과 유의미하게 다릅니다.")
else:
    print("귀무가설 기각 실패: 표본 평균은 150g과 다르다고 할 수 없습니다.")
...

--- One-sample t-test ---
T-statistic: 4.4596
P-value: 0.0002
귀무가설 기각: 표본 평균은 150g과 유의미하게 다릅니다.
...
```

- **결과 해석**
 - `p-value(0.0002)`가 0.05보다 작으므로, 귀무가설을 기각합니다.
 - 즉, 새로 생산된 과자의 평균 무게는 150g과 통계적으로 유의미하게 다르다고 결론 내릴 수 있습니다.

2. 독립표본 T-검정 (Independent two-sample t-test)

- **문제:** A반과 B반 학생들의 수학 시험 점수가 있다. 두 반의 평균 점수에 유의미한 차이가 있는가?
- **함수:** `stats.ttest_ind(a, b, equal_var=True/False)`

```

# 표본 데이터 생성
class_a_scores = [85, 90, 78, 92, 88, 76, 89, 95, 81, 79]
class_b_scores = [72, 80, 68, 75, 71, 82, 70, 65, 77, 74]

# 1. 정규성 검정 (두 그룹 모두 수행)
print(f"Class A Shapiro p-value: {stats.shapiro(class_a_scores).pvalue:.4f}") #
0.6263
print(f"Class B Shapiro p-value: {stats.shapiro(class_b_scores).pvalue:.4f}") #
0.9882, 모두 정규성 만족

# 2. 등분산성 검정 (Levene)
levene_stat, levene_p = stats.levene(class_a_scores, class_b_scores)
print(f"\nLevene's test p-value: {levene_p:.4f}") # 0.3562, 등분산성 만족 (p >
0.05)

# 3. 독립표본 t-검정 수행
# 등분산성 가정이 만족되었으므로 equal_var=True (Student's t-test)
t_statistic, p_value = stats.ttest_ind(class_a_scores, class_b_scores,
equal_var=True)
print(f"\n--- Independent two-sample t-test ---")
print(f"T-statistic: {t_statistic:.4f}")
print(f"P-value: {p_value:.4f}")

if p_value < 0.05:
    print("귀무가설 기각: 두 반의 평균 점수는 유의미하게 다릅니다.")
else:
    print("귀무가설 기각 실패: 두 반의 평균 점수는 차이가 없습니다.")
...

--- Independent two-sample t-test ---
T-statistic: 4.4889
P-value: 0.0003
귀무가설 기각: 두 반의 평균 점수는 유의미하게 다릅니다.
...

```

• 결과 해석

- `p-value(0.0003)`가 0.05보다 작으므로, 귀무가설을 기각합니다.
- 즉, A반과 B반의 평균 수학 점수에는 통계적으로 유의미한 차이가 있다고 결론 내릴 수 있습니다.
- 만약 등분산성 검정에서 `p-value`가 0.05보다 작았다면, `equal_var=False`로 설정하여 **Welch's t-test**를 수행해야 합니다.

3. 대응표본 T-검정 (Paired-samples t-test)

- **문제:** 다이어트 약 복용 전후의 체중 변화가 유의미한지 알아보고 싶다. 10명의 참가자를 대상으로 약 복용 전과 후의 체중을 측정했다.
- **함수:** `stats.ttest_rel(a, b)`

```

# 표본 데이터 (동일한 사람에 대한 전/후 측정값)
weight_before = [78, 82, 75, 68, 90, 85, 79, 72, 88, 81]
weight_after = [75, 79, 74, 67, 85, 82, 77, 70, 84, 78]

```

```

# 차이값 계산
differences = np.array(weight_before) - np.array(weight_after)

# 1. 차이값의 정규성 검정
shapiro_stat, shapiro_p = stats.shapiro(differences)
print(f"Shapiro-Wilk p-value on differences: {shapiro_p:.4f}") # 0.4365, 정규성 만족

# 2. 대응표본 t-검정 수행
t_statistic, p_value = stats.ttest_rel(weight_before, weight_after)
print(f"\n--- Paired-samples t-test ---")
print(f"T-statistic: {t_statistic:.4f}")
print(f"P-value: {p_value:.4f}")

if p_value < 0.05:
    print("귀무가설 기각: 다이어트 약 복용 전후 체중 변화는 유의미합니다.")
else:
    print("귀무가설 기각 실패: 체중 변화가 유의미하지 않습니다.")
...

--- Paired-samples t-test ---
T-statistic: 6.8214
P-value: 0.0001
귀무가설 기각: 다이어트 약 복용 전후 체중 변화는 유의미합니다.
...

```

• 결과 해석

- `p-value(0.0001)`가 0.05보다 매우 작으므로, 귀무가설을 기각합니다.
- 즉, 다이어트 약 복용 전후의 체중 변화는 통계적으로 매우 유의미하다고 결론 내릴 수 있습니다.