

통계적 가설검정 기법

상황별 검정법 빠른 참조 가이드

이 문서는 “통계적 가설검정 입문”을 읽었다는 전제 위에서,
상황에 맞는 검정법을 빠르게 찾아 쓰기 위한 레퍼런스입니다.

0. 이 문서 사용법

이 문서는 처음부터 끝까지 읽는 문서가 아니라, 필요할 때 펼쳐서 원하는 검정법을 찾아보는 레퍼런스입니다. “통계적 가설검정 입문”에서 다룬 위험함수, 1 종·2 종 오류, p 값 같은 기본 개념은 여기서 다시 설명하지 않습니다.

다음 두 가지 입구 중 상황에 맞는 쪽에서 시작하십시오.

- 이미 어떤 검정을 써야 할지 알고 있다면 → 1 장 빠른 참조표에서 바로 함수와 통계량을 확인합니다.
- 어떤 검정을 써야 할지 모르겠다면 → 2 장 순서도를 따라가 검정법을 찾은 뒤, 해당 장으로 이동합니다.

3 장 이후의 각 검정법 페이지는 모두 같은 형식(언제 쓰는가 / 가설 / 검정통계량과 분포 / 가정 / R·Python 함수 / 주의할 점)을 따릅니다.

1. 빠른 참조표

아래 표는 이 문서가 다루는 모든 검정법을 한눈에 보여줍니다. “참조” 열의 번호를 따라가면 해당 검정법의 상세 페이지로 이동할 수 있습니다.

구분	상황	통계량	R 함수	Python 함수	참조
단일표본	모평균 검정, σ 를 아는 경우	Z	직접 계산	<code>scipy.stats.norm</code>	3-1
단일표본	모평균 검정, σ 를 모르는 경우	t	<code>t.test()</code>	<code>ttest_1samp()</code>	3-2
단일표본	모비율 검정	Z	<code>prop.test()</code>	<code>proportions_ztest()</code>	3-3
단일표본	모분산 검정	χ^2	<code>EnvStats::varTest()</code> *	직접 계산	3-4
2 표본	평균 비교, 두 σ 모두 아는 경우	Z	직접 계산	<code>scipy.stats.norm</code>	4-1
2 표본	평균 비교, 등분산 가정	t	<code>t.test(var.equal=T)</code>	<code>ttest_ind(equal_var=T)</code>	4-2
2 표본	평균 비교, 등분산 가정 없음	t (Welch)	<code>t.test()</code>	<code>ttest_ind(equal_var=F)</code>	4-3
2 표본	대응표본(전후) 비교	t	<code>t.test(paired=T)</code>	<code>ttest_rel()</code>	4-4
2 표본	비율 비교	χ^2	<code>prop.test()</code>	<code>proportions_ztest()</code>	4-5
2 표본	분산 비교	F	<code>var.test()</code>	직접 계산	4-6
3 그룹+	평균 비교(3 개 이상 집단)	F	<code>aov()</code>	<code>f_oneway()</code>	5-1
범주형	한 변수가 특정 분포를 따르는가	χ^2	<code>chisq.test()</code>	<code>chisquare()</code>	6-1
범주형	두 변수가 서로 독립인가	χ^2	<code>chisq.test()</code>	<code>chi2_contingency()</code>	6-2

* 별도 패키지 설치가 필요합니다.

2. 검정법 선택 순서도

어떤 검정을 써야 할지 막막할 때는 데이터의 형태와 상황에 대한 질문에 답하며 아래 순서도를 따라가십시오. 각 종착점(초록색 상자)에 적힌 번호가 해당 검정법의 상세 페이지입니다.

2-1. 평균을 비교하는 경로

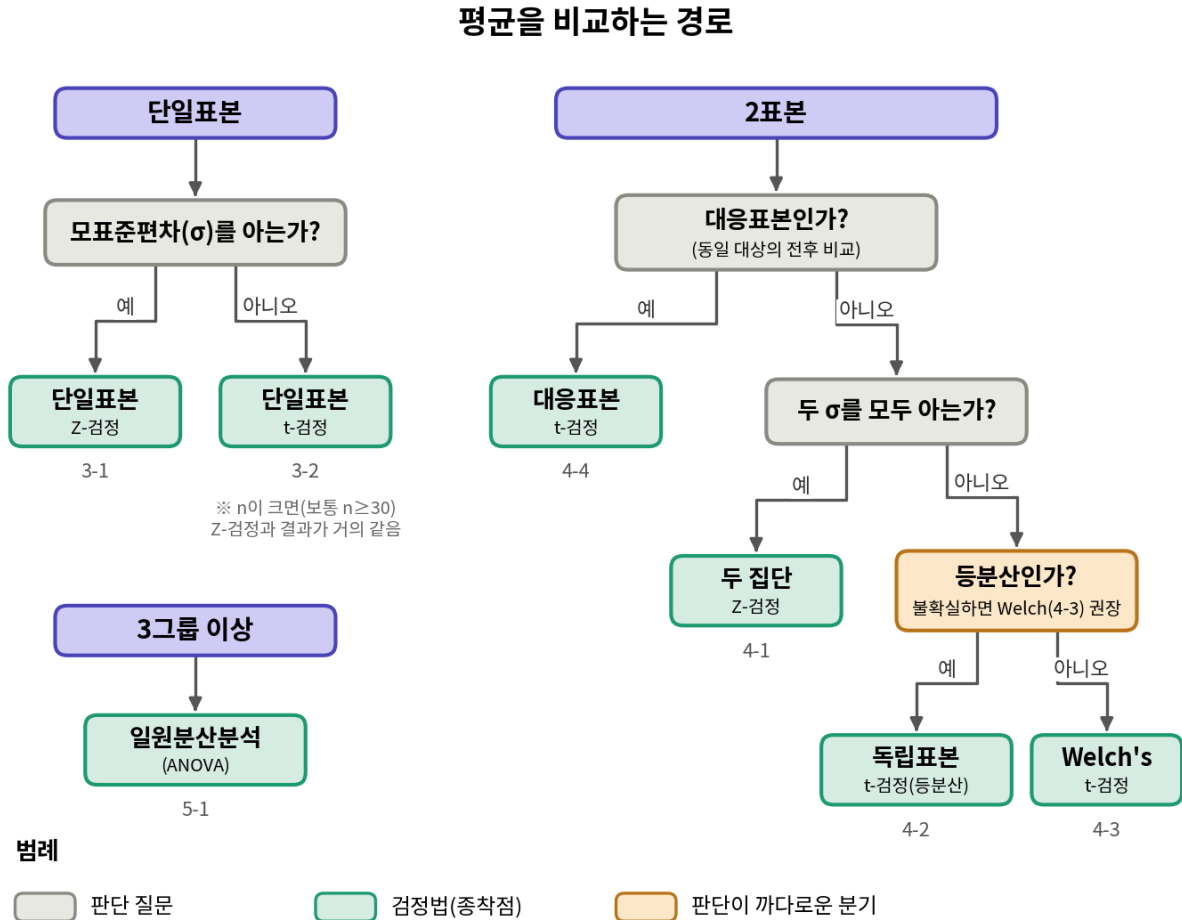


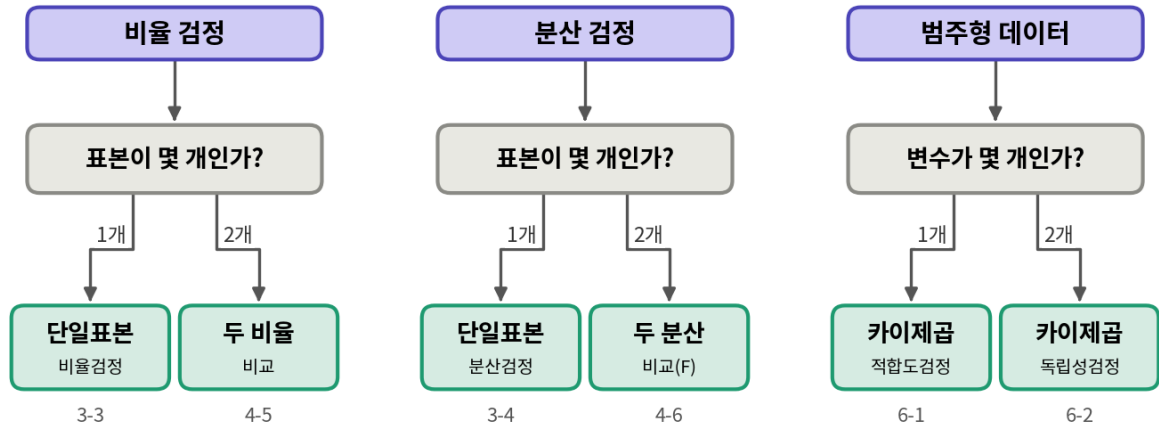
그림 1. 단일표본·2 표본·3 그룹 이상에서 평균(또는 비율과 무관한 연속형 지표)을 비교할 때의 검정법 선택 흐름.

등분산 확인 순서에 대한 안내

그림 1의 “등분산인가?” 분기는 두 집단의 분산이 같다고 가정할 근거가 있는지를 묻습니다. 근거가 없거나 불확실하면 F-검정(4-6)으로 사전 검정하기보다 바로 Welch's t-검정(4-3)을 쓰는 것이 권장됩니다. F-검정은 정규성 가정에 민감하고, 사전 검정 결과에 따라 검정법을 바꾸면 1종 오류율이 왜곡될 수 있기 때문입니다. 등분산이 실제로 성립해도 Welch의 결과는 4-2와 거의 같습니다.

2-2. 비율 · 분산 · 범주형 데이터 경로

비율 · 분산 · 범주형 데이터 경로



범례

판단 질문 검정법(종착점)

그림 2. 비율을 비교하거나, 분산 자체를 검정하거나, 범주형 데이터를 다룰 때의 검정법 선택 흐름.

3. 단일표본 검정

3-1. 단일표본 Z-검정

언제 쓰는가

모집단의 평균이 특정 값과 다른지 검정하고 싶고, 모표준편차 σ 를 정확히 알고 있는 경우. 표준화된 검사나 오래 누적되어 σ 가 잘 알려진 공정 등 실무에서는 드물게 나타나는 상황입니다.

가설

$H_0: \mu = \mu_0$ / $H_1: \mu \neq \mu_0$ (또는 $\mu > \mu_0, \mu < \mu_0$)

검정통계량과 분포

$$Z = (\bar{X} - \mu_0) / (\sigma / \sqrt{n})$$

귀무가설이 참이면 Z는 표준정규분포를 따릅니다.

가정

- 모집단이 정규분포를 따르거나, 표본크기가 충분히 큼(중심극한정리).
- 모표준편차 σ 를 정확히 알고 있습니다.

R 함수

```
# R에는 Z-검정 표준 함수가 없어 직접 계산합니다
xbar <- mean(x); n <- length(x)
z <- (xbar - mu0) / (sigma / sqrt(n))
p_value <- 2 * (1 - pnorm(abs(z))) # 양측검정
```

Python 함수

```
import numpy as np
from scipy import stats

z = (xbar - mu0) / (sigma / np.sqrt(n))
p_value = 2 * (1 - stats.norm.cdf(abs(z))) # 양측검정
```

주의할 점

- σ 를 “안다고 가정”하는 경우와 “표본으로 추정”하는 경우를 혼동하지 않도록 주의해야 합니다. 표본표준편차 s 를 σ 자리에 그대로 대입하면 안 됩니다 (그 경우는 3-2 참조).
- 표본크기가 충분히 크면(보통 $n \geq 30$) t-검정(3-2)과 결과가 사실상 동일해집니다.

3-2. 단일표본 t-검정

언제 쓰는가

모집단의 평균이 특정 값과 다른지 검정하고 싶지만, 모표준편차 σ 를 모르는 경우. 실무에서 가장 흔하게 쓰이는 단일표본 평균 검정입니다.

가설

$H_0: \mu = \mu_0$ / $H_1: \mu \neq \mu_0$ (또는 $\mu > \mu_0, \mu < \mu_0$)

검정통계량과 분포

$$T = (\bar{X} - \mu_0) / (s / \sqrt{n})$$

귀무가설이 참이면 T 는 자유도 $n-1$ 인 t 분포를 따릅니다.

가정

- 모집단이 정규분포를 따르거나, 표본크기가 충분히 큼니다.
- 표본은 모집단에서 독립적으로 추출되었습니다.

R 함수

```
t.test(x, mu = mu0)
```

x: 표본 데이터(벡터), mu0: 비교하려는 모평균 값

Python 함수

```
from scipy import stats
```

```
t_stat, p_value = stats.ttest_1samp(x, mu0)
```

x: 표본 데이터(array-like), mu0: 비교하려는 모평균 값

주의할 점

- 기본적으로 양측검정 결과를 반환합니다. 단측검정이 필요하면 R 은 `alternative="greater"/"less"`, Python 은 `alternative='greater'/'less'` 인자를 추가합니다.
- 표본크기가 작고 모집단이 정규분포에서 크게 벗어나 있으면 결과가 부정확할 수 있습니다.

3-3. 단일표본 비율검정

언제 쓰는가

모집단의 비율(성공률, 불량률 등)이 특정 값과 다른지 검정하고 싶은 경우.

가설

$H_0: p = p_0$ / $H_1: p \neq p_0$ (또는 $p > p_0, p < p_0$)

검정통계량과 분포

$$Z = (\hat{p} - p_0) / \sqrt{(p_0(1-p_0)/n)}$$

귀무가설이 참이고 표본크기가 충분히 크면 Z는 표준정규분포에 근사합니다.

가정

- $np_0 \geq 5$ 이고 $n(1-p_0) \geq 5$ 정도로 표본크기가 충분히 큼(정규근사 조건).
- 각 시행은 독립적이며 성공확률이 동일합니다(베르누이 시행).

R 함수

```
prop.test(x, n, p = p0)
```

x: 성공 횟수, n: 전체 시행 횟수, p0: 비교하려는 모비율

Python 함수

```
from statsmodels.stats.proportion import proportions_ztest  
  
stat, p_value = proportions_ztest(count=x, nobs=n, value=p0)
```

주의할 점

- R의 prop.test()는 기본적으로 연속성 수정(continuity correction)을 적용합니다. Python의 proportions_ztest()는 적용하지 않으므로 두 결과가 미세하게 다를 수 있습니다.
- 또한 표준오차를 계산할 때 R의 prop.test()는 귀무가설 값 p_0 를, Python의 proportions_ztest()는 기본적으로 표본비율 $\hat{p}$ 를 사용합니다. 두 결과를 맞추려면 Python에서 proportions_ztest(x, n, value=p0, prop_var=p0)로, R에서 prop.test(x, n, p=p0, correct=FALSE)로 호출합니다.
- 표본크기가 작으면 정규근사가 부정확해집니다. 이 경우 이항검정(R의 binom.test(), Python의 scipy.stats.binomtest())을 고려해야 합니다.

3-4. 단일표본 분산검정

언제 쓰는가

모집단의 분산(또는 표준편차)이 특정 값과 다른지 검정하고 싶은 경우. 품질관리에서 공정의 변동성이 기준치를 벗어났는지 확인할 때 주로 사용합니다.

가설

$H_0: \sigma^2 = \sigma_0^2$ / $H_1: \sigma^2 \neq \sigma_0^2$ (또는 $\sigma^2 > \sigma_0^2, \sigma^2 < \sigma_0^2$)

검정통계량과 분포

$$\chi^2 = (n-1)s^2 / \sigma_0^2$$

귀무가설이 참이면 이 통계량은 자유도 $n-1$ 인 카이제곱분포를 따릅니다.

가정

- 모집단이 정규분포를 따릅니다(이 검정은 정규성 가정에 특히 민감합니다).

R 함수

```
# 기본 패키지에는 없어 EnvStats 패키지를 사용합니다
# install.packages("EnvStats")
library(EnvStats)
varTest(x, sigma.squared = sigma0_sq)
```

Python 함수

```
from scipy import stats

n = len(x)
s2 = x.var(ddof=1)
chi2_stat = (n - 1) * s2 / sigma0_sq
p_value = 2 * min(stats.chi2.cdf(chi2_stat, n-1),
                    1 - stats.chi2.cdf(chi2_stat, n-1))
```

주의할 점

- 정규성 가정이 깨지면 이 검정의 1종 오류율이 크게 왜곡됩니다. t-검정류보다 정규성 위반에 훨씬 민감하므로, 정규성 검정(Shapiro-Wilk 등)을 먼저 수행하는 것이 안전합니다.

4. 2 표본 검정

4-1. 두 집단 Z-검정

언제 쓰는가

두 독립적인 집단의 평균을 비교하고 싶고, 두 모표준편차 σ_1, σ_2 를 모두 정확히 아는 경우. 단일표본 Z-검정과 마찬가지로 실무에서는 드뭅니다.

가설

$H_0: \mu_1 = \mu_2$ / $H_1: \mu_1 \neq \mu_2$ (또는 $\mu_1 > \mu_2, \mu_1 < \mu_2$)

검정통계량과 분포

$$Z = (\bar{X}_1 - \bar{X}_2) / \sqrt{(\sigma_1^2/n_1 + \sigma_2^2/n_2)}$$

귀무가설이 참이면 Z는 표준정규분포를 따릅니다.

가정

- 두 모집단이 각각 정규분포를 따르거나 표본크기가 충분히 큼니다.
- 두 모표준편차 σ_1, σ_2 를 정확히 알고 있습니다.
- 두 표본은 서로 독립입니다.

R 함수

```
# 직접 계산
z <- (mean(x1) - mean(x2)) / sqrt(sigma1^2/n1 + sigma2^2/n2)
p_value <- 2 * (1 - pnorm(abs(z)))
```

Python 함수

```
import numpy as np
from scipy import stats

z = (x1.mean() - x2.mean()) / np.sqrt(sigma1**2/n1 + sigma2**2/n2)
p_value = 2 * (1 - stats.norm.cdf(abs(z)))
```

주의할 점

- σ_1, σ_2 를 모르면서 표본표준편차로 대체하면 이 검정이 아니라 4-2 또는 4-3 을 사용해야 합니다.

4-2. 독립표본 t-검정 (등분산 가정)

언제 쓰는가

두 독립적인 집단의 평균을 비교하고 싶고, 두 모분산이 같다고 가정할 근거가 있는 경우 (근거가 없으면 4-3 권장).

가설

$H_0: \mu_1 = \mu_2$ / $H_1: \mu_1 \neq \mu_2$ (또는 $\mu_1 > \mu_2, \mu_1 < \mu_2$)

검정통계량과 분포

$$T = (\bar{X}_1 - \bar{X}_2) / (s_p \sqrt{1/n_1 + 1/n_2})$$

s_p 는 두 표본을 합동(pooled)하여 추정한 표준편차이며, 귀무가설이 참이면 T 는 자유도 n_1+n_2-2 인 t 분포를 따릅니다.

가정

- 두 모집단이 각각 정규분포를 따르거나 표본크기가 충분히 큼니다.
- 두 모집단의 분산이 같습니다(등분산성).
- 두 표본은 서로 독립입니다.

R 함수

```
t.test(x1, x2, var.equal = TRUE)
```

Python 함수

```
from scipy import stats  
  
t_stat, p_value = stats.ttest_ind(x1, x2, equal_var=True)
```

주의할 점

- 등분산 가정이 실제로 깨져 있으면 1 종 오류율이 왜곡됩니다. 등분산 여부가 불확실하면 F-검정으로 사전 검정하기보다 처음부터 4-3(Welch's t-검정)을 쓰는 것이 권장됩니다. 사전 검정 결과에 따라 검정법을 바꾸면 전체 1 종 오류율이 오히려 왜곡될 수 있기 때문입니다.

4-3. Welch's t-검정 (등분산 가정 없음)

언제 쓰는가

두 독립적인 집단의 평균을 비교하고 싶고, 두 모분산이 같다고 가정하기 어려운 경우. 등분산 여부가 불확실할 때 기본으로 선택할 수 있는 안전한 대안입니다.

가설

$H_0: \mu_1 = \mu_2$ / $H_1: \mu_1 \neq \mu_2$ (또는 $\mu_1 > \mu_2, \mu_1 < \mu_2$)

검정통계량과 분포

$$T = (\bar{X}_1 - \bar{X}_2) / \sqrt{(s_1^2/n_1 + s_2^2/n_2)}$$

귀무가설이 참이면 T는 근사적으로 t분포를 따르며, 자유도는 Welch-Satterthwaite 근사식으로 별도 계산됩니다(공식 생략, 소프트웨어가 자동 계산).

가정

- 두 모집단이 각각 정규분포를 따르거나 표본크기가 충분히 큼니다.
- 두 표본은 서로 독립입니다.
- 등분산 가정이 필요하지 않습니다.

R 함수

```
t.test(x1, x2) # var.equal의 기본값이 FALSE이므로 그대로 사용
```

Python 함수

```
from scipy import stats  
  
t_stat, p_value = stats.ttest_ind(x1, x2, equal_var=False)
```

주의할 점

- R의 t.test() 기본값이 이미 Welch 방식입니다. 등분산 가정 검정(4-2)을 쓰려면 var.equal=TRUE를 명시해야 합니다.
- 등분산이 실제로 성립하는 경우에도 Welch's t-검정을 쓰면 결과가 크게 달라지지 않으므로, 확신이 없다면 이 검정을 기본으로 사용하는 것이 안전합니다.

4-4. 대응표본 t-검정

언제 쓰는가

같은 대상을 두 번 측정한 경우(치료 전후, 처리 전후 등)의 평균 차이를 검정하고 싶은 경우.

가설

$H_0: \mu_d = 0$ (차이의 평균이 0) / $H_1: \mu_d \neq 0$ (또는 $\mu_d > 0, \mu_d < 0$)

검정통계량과 분포

$$T = \bar{d} / (s_d / \sqrt{n})$$

$\bar{d}$ 는 쌍별 차이의 표본평균, s_d 는 차이의 표본표준편차이며, 귀무가설이 참이면 T는 자유도 $n-1$ 인 t분포를 따릅니다.

가정

- 측정값의 쌍별 차이가 정규분포를 따르거나 표본크기가 충분히 큼니다.
- 측정은 동일한 대상에서 짝을 이루어 이루어집니다(독립표본이 아님).

R 함수

```
t.test(x1, x2, paired = TRUE)

# x1, x2: 같은 대상의 전/후 측정값 (순서가 일치해야 함)
```

Python 함수

```
from scipy import stats

t_stat, p_value = stats.ttest_rel(x1, x2)
```

주의할 점

- 대응표본을 독립표본(4-2, 4-3)으로 잘못 처리하면 검정력이 손실됩니다. 같은 대상을 반복 측정했다면 반드시 대응표본 방법을 사용해야 합니다.
- x1과 x2의 데이터 순서가 같은 대상끼리 정확히 짝지어져 있는지 확인해야 합니다.

4-5. 두 비율 검정

언제 쓰는가

두 독립적인 집단의 비율(전환율, 성공률 등)을 비교하고 싶은 경우. A/B 테스트에서 흔히 사용됩니다.

가설

$H_0: p_1 = p_2$ / $H_1: p_1 \neq p_2$ (또는 $p_1 > p_2$, $p_1 < p_2$)

검정통계량과 분포

$$\chi^2 = \sum (\text{관측빈도} - \text{기대빈도})^2 / \text{기대빈도}$$

2×2 분할표에 대한 카이제곱검정과 수학적으로 동치이며, 귀무가설이 참이면 자유도 1 인 카이제곱분포를 따릅니다. Z 통계량으로 동등하게 표현되는 경우도 혼합니다.

가정

- 각 집단의 표본크기가 충분히 큼니다(정규근사 조건).
- 두 표본은 서로 독립입니다.

R 함수

```
prop.test(x = c(x1, x2), n = c(n1, n2))
```

```
# x1, x2: 각 집단의 성공 횟수, n1, n2: 각 집단의 전체 시행 횟수
```

Python 함수

```
from statsmodels.stats.proportion import proportions_ztest
```

```
stat, p_value = proportions_ztest(count=[x1, x2], nobs=[n1, n2])
```

주의할 점

- 표본크기가 작으면 정규근사가 부정확해질 수 있습니다. 이 경우 R 의 `fisher.test()`, Python 의 `scipy.stats.fisher_exact()`(피셔의 정확검정)를 고려해야 합니다.
- R 과 Python 함수가 기본적으로 적용하는 연속성 수정 여부가 다를 수 있어 결과가 미세하게 다를 수 있습니다.

4-6. 두 분산 비교 (F-검정)

언제 쓰는가

두 독립적인 집단의 분산(변동성)이 같은지 검정하고 싶은 경우. 주로 4-2(등분산 가정 t-검정)을 쓸지 4-3(Welch's t-검정)을 쓸지 결정하기 위한 사전 확인 단계로 사용됩니다.

가설

$H_0: \sigma_1^2 = \sigma_2^2$ / $H_1: \sigma_1^2 \neq \sigma_2^2$

검정통계량과 분포

$$F = S_1^2 / S_2^2$$

귀무가설이 참이면 F는 자유도 (n_1-1 , n_2-1)인 F 분포를 따릅니다.

가정

- 두 모집단이 각각 정규분포를 따릅니다(정규성 가정에 민감합니다, 3-4 와 마찬가지로).

R 함수

```
var.test(x1, x2)
```

Python 함수

```
from scipy import stats

f_stat = x1.var(ddof=1) / x2.var(ddof=1)
df1, df2 = len(x1) - 1, len(x2) - 1
p_value = 2 * min(stats.f.cdf(f_stat, df1, df2),
                  1 - stats.f.cdf(f_stat, df1, df2))
```

주의할 점

- 정규성 가정이 깨지면 이 검정 자체의 신뢰도가 떨어집니다. 정규성이 의심스러운 상태에서 등분산 여부를 판단하려 한다면, 차라리 등분산을 가정하지 않는 4-3(Welch's t-검정)을 바로 사용하는 편이 안전할 수 있습니다.

5. 3 그룹 이상 검정

5-1. 일원분산분석 (ANOVA)

언제 쓰는가	3 개 이상의 독립적인 집단의 평균을 동시에 비교하고 싶은 경우. 여러 번의 2 표본 t-검정을 반복하는 대신 한 번에 비교합니다.
가설	$H_0: \mu_1 = \mu_2 = \dots = \mu_k$ (모든 집단의 평균이 같다) / H_1 : 적어도 한 집단의 평균이 다르다

검정통계량과 분포

$$F = (\text{집단 간 분산}) / (\text{집단 내 분산})$$

귀무가설이 참이면 F는 자유도 (k-1, n-k)인 F 분포를 따릅니다(k는 집단 수, n은 전체 표본크기).

가정

- 각 집단이 정규분포를 따르거나 표본크기가 충분히 큼니다.
- 모든 집단의 분산이 동일합니다(등분산성).
- 각 집단의 관측값은 서로 독립입니다.

R 함수

```
result <- aov(value ~ group, data = df)
summary(result)

# df: value(측정값)와 group(집단 구분) 열을 가진 데이터프레임
```

Python 함수

```
from scipy import stats

f_stat, p_value = stats.f_oneway(group1, group2, group3)
# group1, group2, group3: 각 집단의 데이터 배열
```

주의할 점

- ANOVA가 유의하다는 결과는 “적어도 한 집단이 다르다”는 것만 알려줄 뿐, 어느 집단끼리 다른지는 알려주지 않습니다. 어느 집단인지 확인하려면 사후검정(Tukey HSD 등)이 별도로 필요하며, 이는 이 문서의 범위를 벗어납니다.
- 여러 번의 t-검정을 반복하는 대신 ANOVA를 쓰는 이유는 다중검정으로 인한 1종 오류 누적을 피하기 위함입니다.

6. 범주형 데이터 검정

6-1. 카이제곱 적합도검정

언제 쓰는가	하나의 범주형 변수가 특정 분포(비율)를 따르는지 검정하고 싶은 경우. 예: 주사위가 공정한지, 특정 비율로 선호도가 나뉘는지.
가설	H_0 : 각 범주의 비율이 특정 값과 같다 / H_1 : 적어도 한 범주의 비율이 다르다

검정통계량과 분포

$$\chi^2 = \sum (\text{관측빈도} - \text{기대빈도})^2 / \text{기대빈도}$$

귀무가설이 참이면 이 통계량은 자유도 (범주 수 - 1)인 카이제곱분포를 따릅니다.

가정

- 각 범주의 기대빈도가 5 이상입니다(일반적인 경험칙).
- 관측값들은 서로 독립입니다.

R 함수

```
chisq.test(x = observed, p = expected_props)

# observed: 각 범주의 관측빈도, expected_props: 귀무가설상의 비율
```

Python 함수

```
from scipy import stats

chi2_stat, p_value = stats.chisquare(f_obs=observed, f_exp=expected)
# expected: 기대빈도(비율이 아니라 빈도 값)
```

주의할 점

- 기대빈도가 5 미만인 범주가 많으면 카이제곱 근사가 부정확해집니다. 이 경우 범주를 통합하거나 정확검정을 고려해야 합니다.
- Python의 stats.chisquare()는 기대빈도를, R의 chisq.test()는 기대비율을 인자로 받는 차이가 있어 혼동하기 쉽습니다.

6-2. 카이제곱 독립성검정

언제 쓰는가	두 범주형 변수가 서로 독립인지(관련이 있는지) 검정하고 싶은 경우. 예: 성별과 제품 선호도 사이에 관련이 있는가.
가설	H_0 : 두 변수는 서로 독립이다 / H_1 : 두 변수는 서로 독립이 아니다(관련이 있다)

검정통계량과 분포

$$\chi^2 = \sum (\text{관측빈도} - \text{기대빈도})^2 / \text{기대빈도}$$

귀무가설이 참이면 이 통계량은 자유도 (행 수-1)×(열 수-1)인 카이제곱분포를 따릅니다.

가정

- 분할표 각 칸의 기대빈도가 5 이상입니다(일반적인 경험칙).
- 관측값들은 서로 독립입니다.

R 함수

```
chisq.test(table(var1, var2))  
  
# var1, var2: 두 범주형 변수 / table()로 분할표를 생성
```

Python 함수

```
from scipy.stats import chi2_contingency  
  
chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)  
# contingency_table: 행 × 열 형태의 분할표 (2차원 배열)
```

주의할 점

- 분할표가 2×2 이면 R 과 Python 모두 표본크기와 관계없이 기본적으로 연속성 수정(Yates' correction)을 적용합니다. 이를 끄려면 R 은 correct=FALSE, Python 은 correction=False 를 지정해야 합니다.
- 기대빈도가 5 미만인 칸이 많으면(특히 2×2 표에서) 피셔의 정확검정을 대신 고려해야 합니다.

이 문서는 단일표본·2 표본·3 그룹 이상의 평균 비교, 비율·분산 검정, 범주형 데이터 검정을 다루었습니다.
ANOVA 에서 유의한 결과가 나온 뒤 어느 집단이 다른지 확인하는 사후검정, 정규성이나 등분산 가정이 깨졌을
때 쓰는 비모수 검정, 다중검정 보정 등은 이 문서의 범위를 벗어나며 후속 가이드에서 다룰 예정입니다.