Module4

t-test

◆ 학습목표

Pyhon을 이용해 t-test 분석방법을 학습한다.

I. One Sample T-test

II. Independent Sample T-test

III. Paired Sample T-test

IV. Equivalence test

V. Sample size

Ⅰ. One Sample T-test

모평균검정

One Sample t-test

❖ 문제의 정의

- B아이스크림회사에서 판매하는 아이스크림 중 파인트의 무게는 320g이다.
- 그러나 G대학 앞에 있는 점포에서 파는 아이스크림의 무게가 320g이 아니라는 소비자들의 불만이 있었다.
- 이에 따라 소비자단체에서는 B아이스크림회사에서 만든 아이스크림이 320g인지를 검사하고자 한다.
- 04_1.OST.csv

❖ 가설

- 귀무가설(H_0): 파인트의 무게는 320g이다.

$$H_0: \mu = 320$$

- 연구가설(H_1): 파인트의 무게는 320g이 아니다.

$$\left\{ \begin{array}{ll} H_1: \mu \neq 320 & \text{양측검정(two-sided test)} \\ H_1: \mu > 320 & \text{우측검정(right-sided test)} \\ H_1: \mu < 320 & \text{좌측검정(left-sided test)} \end{array} \right.$$

One Sample t-test

❖ t-test의 통계적 가정

- 모집단의 분포가 정규분포(모수통계)로 가정 → 표본이므로 t분포 가정
- 표본이 작으면서 이상점이 많을 경우: 비모수적 통계분석 사용

❖ 가설검정

- 중심극한정리: 표집분포는 평균이 $\boldsymbol{\mu}$이고 분산이 $\frac{\boldsymbol{\sigma^2}}{\boldsymbol{n}}$인 정규분포에 근사

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$$

- (가정)모집단의 표준편차 σ를 알 경우 : 표준정규분포

$$x_{critical} = \mu_0 - 1.96 \frac{\sigma}{\sqrt{n}} \qquad z_{cal} = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n}}} \sim N(0,1)$$

- (실제)모집단의 표준편차 σ를 모를 경우 : Student t 분포

$$x_{critical} = \mu_0 \pm t_{n-1} \frac{s}{\sqrt{n}} \qquad t_{cal} = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$$

One Sample t-test

❖ 통계치

- 표본 (n) : 100
- 표본평균 ($\bar{X}$) : 317.91
- 표본표준편차 (s): 6.77, 표준오차 ($\frac{s}{\sqrt{n}}$): 0.68

❖ 임계치

$$x_{critical} = \mu_0 \pm t_{n-1}\frac{s}{\sqrt{n}} = 320 \pm 1.984\frac{6.77}{\sqrt{100}} = 320 \pm 1.34 = [318.64, 321.36]$$

❖ 검정통계량 (test statistics)

$$t_{cal} = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} = \frac{317.91 - 320}{\frac{6.77}{\sqrt{100}}} = \frac{-2.09}{0.68} = -3.09$$

$$* \; t_{cal} = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$$

❖ 유의확률(p-value) 계산

$$p - value = P(|t| > 3.09) = 0.003$$

One Sample t-test

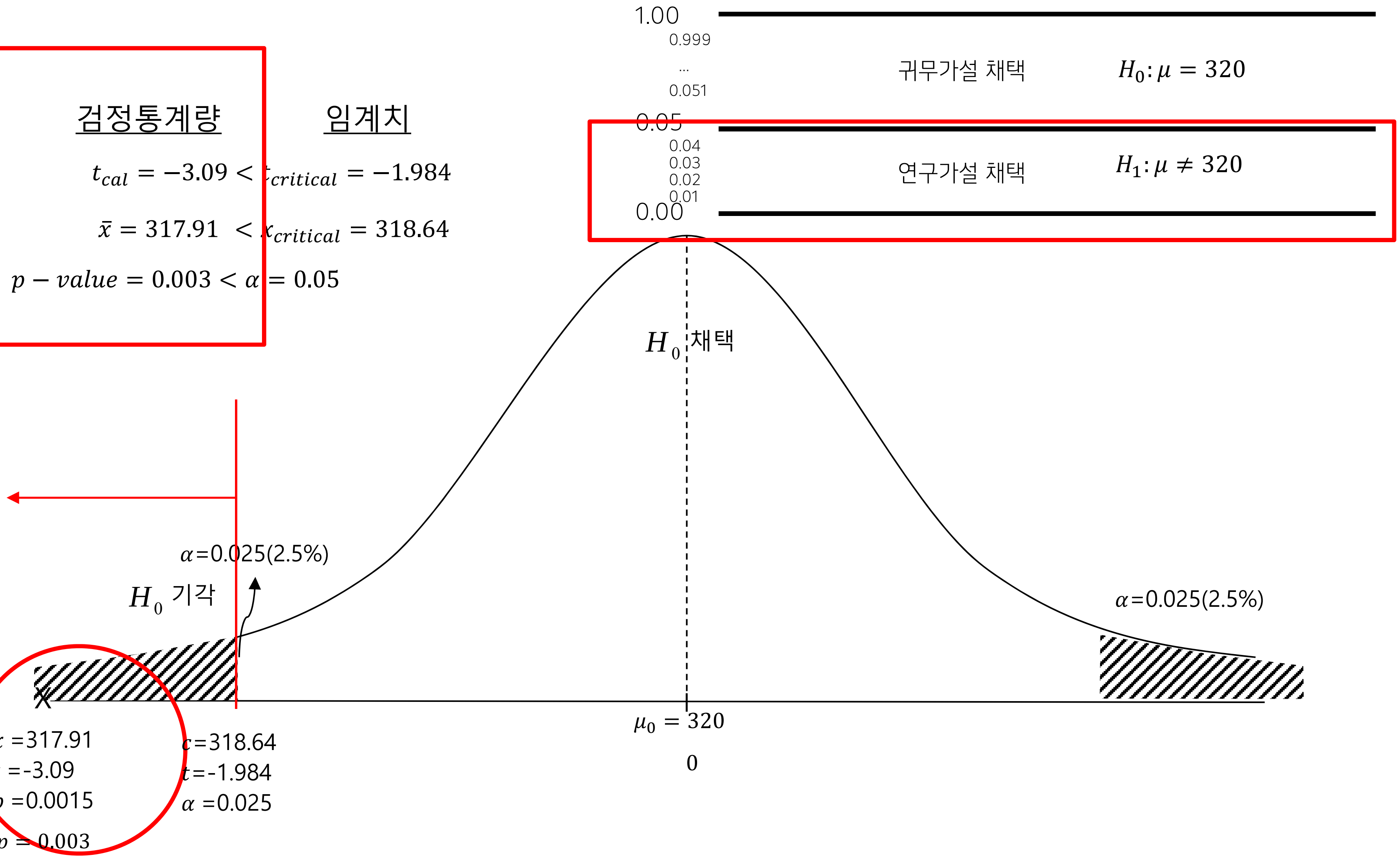
❖ 검정결과

검정통계량 임계치

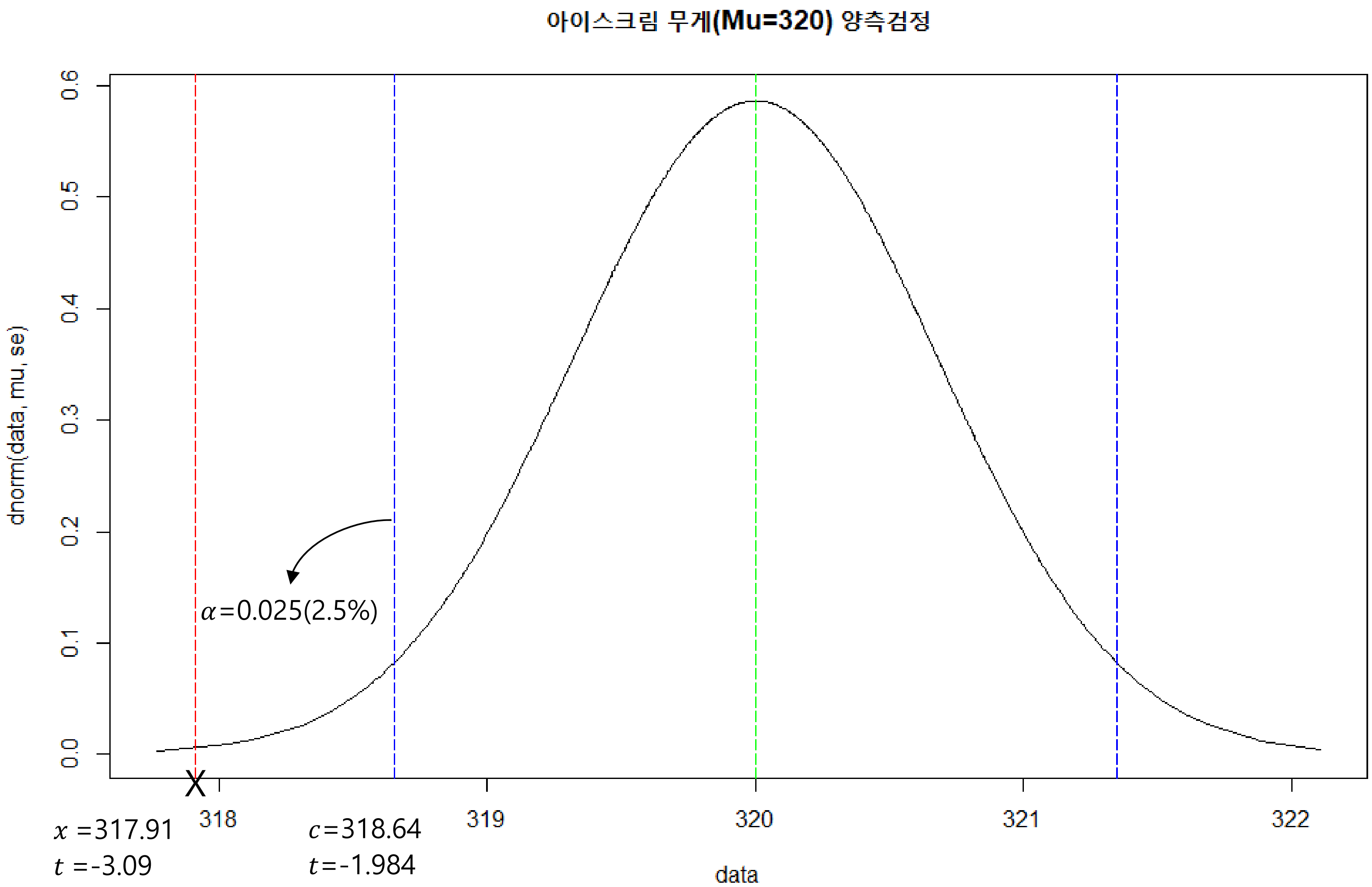
$t_{cal} = -3.09 < t_{critical} = -1.984$

$\bar{x} = 317.91 < x_{critical} = 318.64$

$p - value = 0.003 < \alpha = 0.05$



One Sample t-test



x =317.91
t =-3.09
p =0.0015

c=318.64
t=-1.984
α =0.025

$* p = 0.003$

LG Life's Good

모비율검정

모비율 가설검정

❖ 문제의 정의

- G텔레콤의 고객 이탈율은 9%이다. 500명의 고객을 샘플로 이탈가능성을 조사하였다.
- 500명 중 50명이 앞으로 이탈할 것으로 나타났다.
- 유의수준 5%로 고객 이탈율이 9%라고 할 수 있는가?

$$p = \frac{50}{500} = 0.1$$

$$np = 0.1 \times 50 = 5 \geq 5$$

* 만약, 조건이 충족되지 않으면 이항분포 또는 포아송 분포로

❖ 가설

$$H_0: \pi = 0.09$$

$$H_1: \pi \neq 0.09$$

❖ 임계치

$$\alpha = 0.05 \text{일 때}, z = 1.960$$

모비율 가설검정

❖ 임계치

$$\bar{p}_{critical} = \pi_0 + z_a \sqrt{\frac{p(1-p)}{n}} = 0.09 + 1.96 \sqrt{\frac{0.100(1-0.100)}{500}} = 0.116$$

❖ 검정통계량 (test statistics)

$$z_{cal} = \frac{p - \pi_0}{\sqrt{\frac{p(1-p)}{n}}} = \frac{0.10 - 0.09}{\sqrt{\frac{0.100(1-0.100)}{500}}} = \frac{0.01}{0.013} = 0.745$$

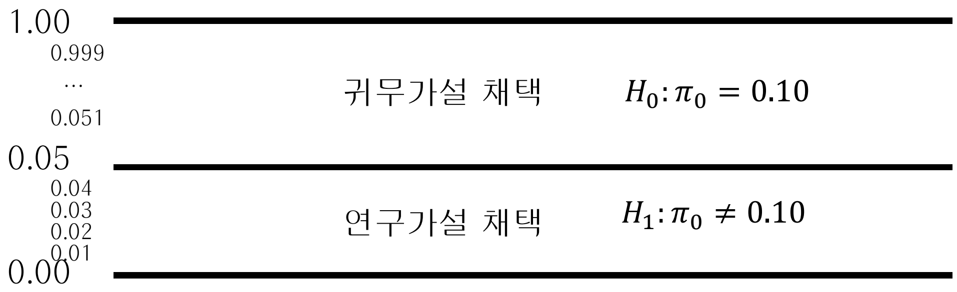
$$* \; z_{cal} = \frac{p - \pi_0}{\sqrt{\frac{\pi_0(1-\pi_0)}{n}}} \sim N(0,1)$$

❖ 유의확률(p-value) 계산

$$p - value = P(|z| > 0.745) = 0.434$$

모비율 가설검정

❖ 검정결과

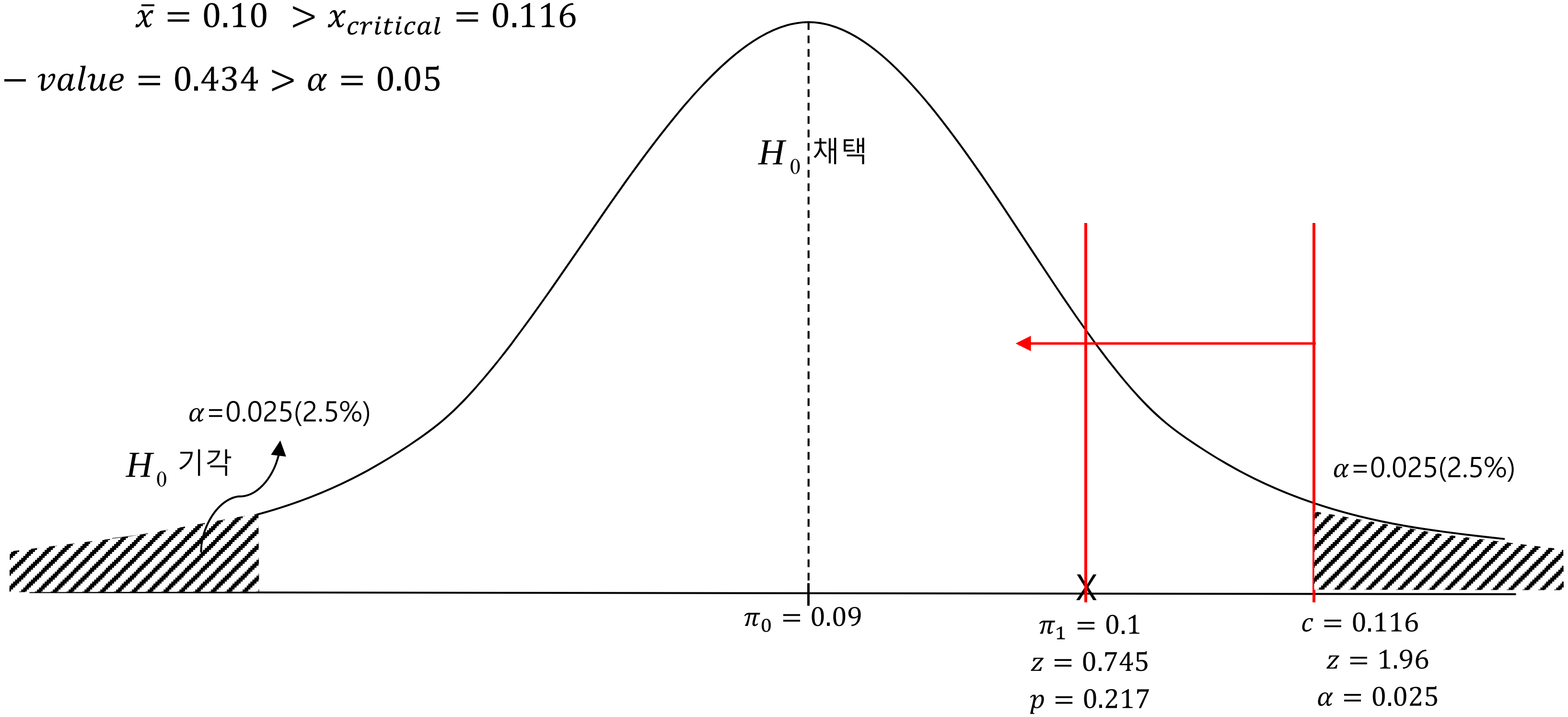


<u>검정통계량</u> <u>임계치</u>

$$z_{cal} = 0.745 > z_{critical} = 1.96$$

$$\bar{x} = 0.10 \ > x_{critical} = 0.116$$

$$p - value = 0.434 > \alpha = 0.05$$



모분산검정

LG
Life's Good

모분산 가설검정

❖ 문제의 정의

- K음료는 공장에서 병뚜껑을 제작하고 있다.
- 병뚜껑의 규격은 지름 5cm이며, 품질관리를 위해 표준편차는 0.8mm이다.
- 오전에 생산한 5,000개의 품질 검사를 위해 총 30개의 샘플을 조사하였다.
- 샘플평균은 5cm이며, 표준편차는 1.2mm로 나타났다.
- 생산을 계속 해도 괜찮은가? (품질경영 - 신뢰성 테스트)

❖ 가설

$H_0: \sigma^2 = (0.8)^2 = 0.64$

$H_1: \sigma^2 \neq 0.64$

❖ 임계치 (양측검정)

$n = 30, \alpha = 0.05$일 때, $\chi^2 = [16.04 \sim 45.72]$

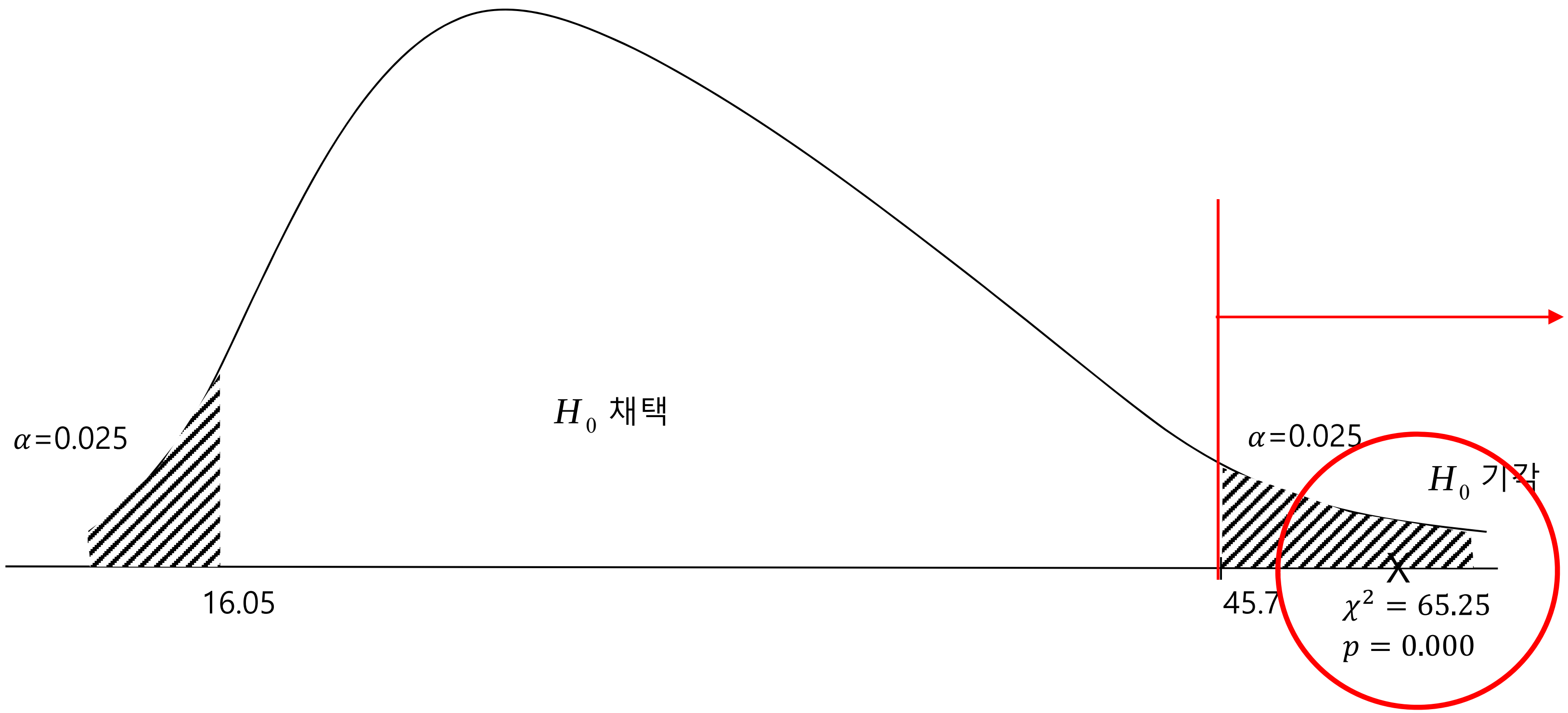
모분산 가설검정

❖ 검정결과

$$\chi^2_{cal} = \frac{(n-1)s^2}{\sigma_o^2} = \frac{(30-1)(1.2)^2}{(0.8)^2} = 65.25$$

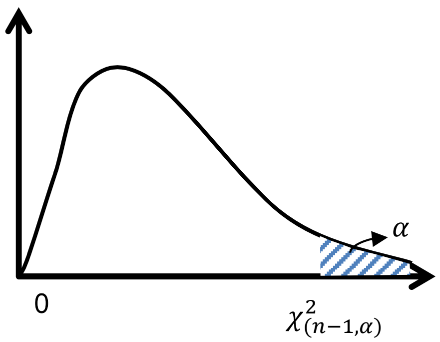
$$* \ \chi^2_{cal} = \frac{(1-n)s^2}{\sigma_o^2} \sim \chi^2_{n-1}$$

$$p - value = P\left(\chi^2 > 65.25\right) = 0.000$$



모분산 가설검정

❖ χ^2분포



df.	0.990	0.975	0.950	0.900	0.500	0.100	0.050	0.025	0.100	0.005
1	0.0002	0.0001	0.004	0.02	0.45	2.71	3.84	5.02	6.63	7.88
2	0.02	0.05	0.10	0.21	1.39	4.61	5.99	7.38	9.21	10.60
3	0.11	0.22	0.35	0.58	2.37	6.25	7.81	9.35	11.34	12.84
4	0.30	0.48	0.71	1.06	3.36	7.78	9.49	11.14	13.28	14.86
5	0.55	0.83	1.15	1.61	4.35	9.24	11.07	12.83	15.09	16.75
10	2.56	3.25	3.94	4.87	9.34	15.99	18.31	20.48	23.21	25.19
20	8.26	9.59	10.85	12.44	19.34	28.41	31.14	34.17	37.57	40.00
29	14.26	16.05	17.71	19.77	28.34	39.09	42.56	45.72	49.59	52.34
30	14.95	16.79	18.49	20.60	29.34	40.26	43.77	46.98	50.89	53.67
40	22.16	24.43	26.51	29.05	39.34	51.81	55.76	59.34	63.69	66.77
50	29.71	32.36	34.76	37.69	49.33	63.17	67.50	71.42	76.15	79.49
60	37.48	40.48	43.19	46.46	59.33	74.40	79.08	83.30	88.38	91.95
70	45.44	48.76	51.74	55.33	69.33	85.53	90.53	95.02	100.43	104.21
80	53.54	57.15	60.39	64.28	79.33	96.58	101.88	106.63	112.33	116.32
90	61.75	65.65	69.13	73.29	89.33	107.57	113.15	118.14	124.12	128.30
100	70.06	74.22	77.93	82.36	99.33	118.50	124.34	129.56	135.81	140.17

실습

04_1.One Sample t-test

04_1.One Sample t-test

- https://pingouin-stats.org/build/html/generated/pingouin.ttest.html#pingouin.ttest

1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기
    !sudo apt-get install -y fonts-nanum
    !sudo fc-cache -fv
    !rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin

    #  *** 세션 다시 시작
```

```
[1] # 1.기본
    import numpy as np  # numpy 패키지 가져오기
    import matplotlib.pyplot as plt # 시각화 패키지 가져오기
    import seaborn as sns # 시각화

    # 2.데이터 가져오기
    import pandas as pd # csv -> dataframe으로 전환

    # 3.통계분석 package
    import pingouin as pg
    from scipy import stats
    import statsmodels.api as sm
```

```
[2] # 기본세팅
    # 테마 설정
    sns.set_theme(style = "darkgrid")
```

✓ 0초 오전 9:15에 완료됨

1.기본 package 설정

1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기
    !sudo apt-get install -y fonts-nanum
    !sudo fc-cache -fv
    !rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin

    #  *** 세션 다시 시작
```

```
[1] # 1.기본
    import numpy as np  # numpy 패키지 가져오기
    import matplotlib.pyplot as plt # 시각화 패키지 가져오기
    import seaborn as sns # 시각화

    # 2.데이터 가져오기
    import pandas as pd # csv -> dataframe으로 전환

    # 3.통계분석 package
    import pingouin as pg
    from scipy import stats
    import statsmodels.api as sm
```

```
[2] # 기본세팅
    # 테마 설정
    sns.set_theme(style = "darkgrid")

    # 한글 인식
    plt.rc('font', family='NanumBarunGothic')
    plt.rcParams['axes.unicode_minus'] = False # -인식
```

2.데이터 불러오기

✓ 0초   오전 9:15에 완료됨

2.데이터 불러오기

2.데이터 불러오기

2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

```
[3] ost_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/04_1.OST.csv', encoding="cp949")
    ost_df.head()
```

	무게1	무게2	무게3	무게4	무게5	무게6
0	242.0	242.0	242.0	242.0	242.0	242.0
1	244.3	244.3	244.3	244.3	244.3	244.3
2	304.9	301.0	301.0	307.9	309.9	307.9
3	305.2	304.0	304.0	305.2	310.2	305.2
4	304.0	304.0	304.0	307.0	309.0	307.0

Next steps: View recommended plots

2.2 자료구조 살펴보기

```
[4] ost_df.shape
```

```
(102, 6)
```

```
[5] ost_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   무게1     102 non-null    float64
 1   무게2     102 non-null    float64
```

✓ 0초 오전 9:15에 완료됨

2.데이터 불러오기

2.2 자료구조 살펴보기

[4]
```
ost_df.shape
```

```
(102, 6)
```

[5]
```
ost_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 102 entries, 0 to 101
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   무게1     102 non-null    float64
 1   무게2     102 non-null    float64
 2   무게3     102 non-null    float64
 3   무게4     102 non-null    float64
 4   무게5     102 non-null    float64
 5   무게6     102 non-null    float64
dtypes: float64(6)
memory usage: 4.9 KB
```

[6]
```
ost_df.columns
```

```
Index(['무게1', '무게2', '무게3', '무게4', '무게5', '무게6'], dtype='object')
```

3.기술통계

[7]
```
# 수치형 변수
ost_df.describe().round(2).T
```

	count	mean	std	min	25%	50%	75%	max
무게1	102.0	316.44	12.39	242.0	313.92	317.35	322.50	331.8
무게2	102.0	317.13	12.58	242.0	314.73	318.15	323.30	332.6
무게3	102.0	317.14	12.58	242.0	314.73	318.15	323.30	332.6

✓ 0초 오전 9:15에 완료됨

3.기술통계

[7]
```
# 수치형 변수
ost_df.describe().round(2).T
```

	count	mean	std	min	25%	50%	75%	max
무게1	102.0	316.44	12.39	242.0	313.92	317.35	322.50	331.8
무게2	102.0	317.13	12.58	242.0	314.73	318.15	323.30	332.6
무게3	102.0	317.14	12.58	242.0	314.73	318.15	323.30	332.6
무게4	102.0	319.35	12.77	242.0	316.92	320.35	325.50	334.8
무게5	102.0	321.35	12.98	242.0	318.92	322.35	327.50	336.8
무게6	102.0	315.97	12.33	242.0	312.52	317.90	320.68	334.8

[8]
```
ost_df.agg({"무게1": ["count", "mean", "std", "min", "max", "median", "skew", "kurtosis"]}).T \
    .round(2)
```

	count	mean	std	min	max	median	skew	kurtosis
무게1	102.0	316.44	12.39	242.0	331.8	317.35	-4.07	22.94

4.t-test

4.0 scipy.stats와 비교

[9]
```
# scipy.stats.ttest_1samp
stats.ttest_1samp(ost_df["무게1"], popmean = 320, alternative = "two-sided")
```

```
TtestResult(statistic=-2.899472691059131, pvalue=0.004586364436777763, df=101)
```

4.1 차이가 있는 경우(two-sided)

✓ 0초 오전 9:15에 완료됨

4.t-test

4.0 scipy.stats와 비교

[9]
```
# scipy.stats.ttest_1samp
stats.ttest_1samp(ost_df["무게1"], popmean = 320, alternative = "two-sided")
```

```
TtestResult(statistic=-2.899472691059131, pvalue=0.004586364436777763, df=101)
```

4.1 차이가 있는 경우(two-sided)

[10]
```
# two-sided
pg.ttest(ost_df["무게1"], 320, alternative = "two-sided").round(3)

# 5.정규분포 가정 검정후 다시 분석
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-2.899	101	two-sided	0.005	[314.01, 318.88]	0.287	5.612	0.819

5. 정규분포 가정 검정후 다시 분석

[ ]
```
# two-sided
pg.ttest(ost_df["무게2"], 320, alternative = "two-sided").round(4)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-2.3039	101	two-sided	0.0233	[314.66, 319.6]	0.2281	1.363	0.6263

[ ]
```
# less
pg.ttest(ost_df["무게2"], 320, alternative = "less").round(3)
```

✓ 0초 오전 9:15에 완료됨

5.가정검정

5.정규성 검정

5.1 정규성 검정

[11]
```
pg.normality(ost_df["무게1"])
```

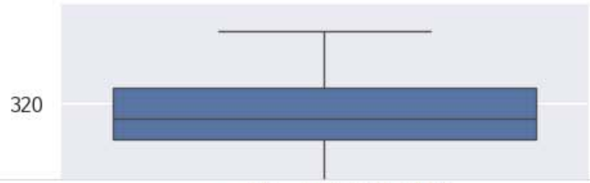
	W	pval	normal
무게1	0.63365	1.388964e-14	False

[12]
```
pg.normality(ost_df).T.round(3)
```

	무게1	무게2	무게3	무게4	무게5	무게6
W	0.63365	0.640934	0.639918	0.620722	0.605953	0.631473
pval	0.0	0.0	0.0	0.0	0.0	0.0
normal	False	False	False	False	False	False

5.2 이상치제거

[13]
```
# 한글 폰트 인식
sns.catplot(data = ost_df,
            y = "무게1",
            kind = "box")
plt.show()
```

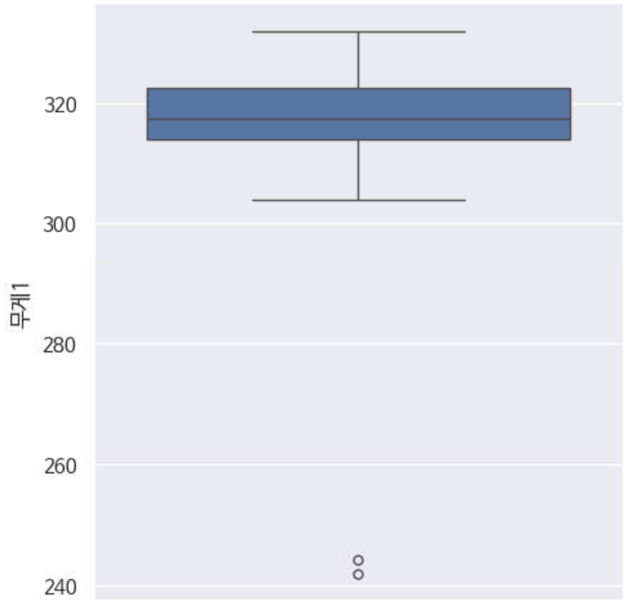


✓ 0초 오전 9:20에 완료됨

5.가정검정

5.2 이상치제거

```
[13] # 한글 폰트 인식
     sns.catplot(data = ost_df,
                 y = "무게1",
                 kind = "box")
     plt.show()
```

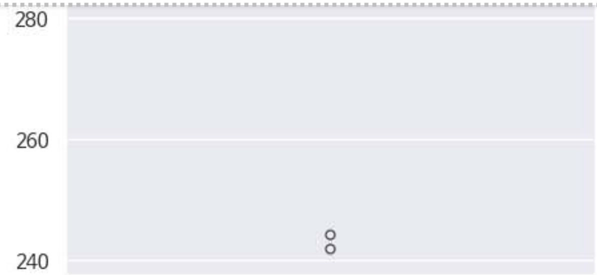



```
[14] filter = (ost_df["무게1"] <= 250)
     ost_df.loc[filter]
```

무게1	무게2	무게3	무게4	무게5	무게6

✓ 0초 오전 9:20에 완료됨

[13]



[14]
```
filter = (ost_df["무게1"] <= 250)
ost_df.loc[filter]
```

	무게1	무게2	무게3	무게4	무게5	무게6
0	242.0	242.0	242.0	242.0	242.0	242.0
1	244.3	244.3	244.3	244.3	244.3	244.3

[15]
```
ost_df.drop(ost_df[filter].index, inplace = True)
```

[16]
```
pg.normality(ost_df).T.round(3)

# 4.1로 다시 분석
```

	무게1	무게2	무게3	무게4	무게5	무게6
W	0.981564	0.985788	0.985853	0.984864	0.981564	0.964417
pval	0.175659	0.36139	0.365223	0.310514	0.175659	0.008408
normal	True	True	True	True	True	False

5.3 비모수 통계

[ ]
```
# 비모수통계
pg.wilcoxon(ost_df["무게6"] - 320, alternative = "two-sided").round(3)
```

cro...　✓ 0초　오전 9:20에 완료됨

4.t-test

4.1 차이가 있는 경우(two-sided)

```
[17] # two-sided
pg.ttest(ost_df["무게1"], 320, alternative = "two-sided").round(3)

# 5.정규분포 가정 검정후 다시 분석
```

정규성 검정후 다시 분석한 결과

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.087	99	two-sided	0.003	[316.57, 319.25]	0.309	9.318	0.864

4.2 양측과 단측 검정 비교(less)

단측검정으로 바뀌면

```
[18] # two-sided
pg.ttest(ost_df["무게2"], 320, alternative = "two-sided").round(4)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.9925	99	two-sided	0.0491	[317.22, 319.99]	0.1992	0.737	0.5053

```
[19] # less
pg.ttest(ost_df["무게2"], 320, alternative = "less").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.992	99	less	0.025	[-inf, 319.77]	0.199	1.473	0.631

```
[20] # two-sided
pg.ttest(ost_df["무게3"], 320, alternative = "two-sided").round(4)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.9827	99	two-sided	0.0502	[317.24, 320.0]	0.1983	0.723	0.5014

✓ 0초 오전 9:21에 완료됨

4.t-test

[20]
```
# two-sided
pg.ttest(ost_df["무게3"], 320, alternative = "two-sided").round(4)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.9827	99	two-sided	0.0502	[317.24, 320.0]	0.1983	0.723	0.5014

[21]
```
# less
pg.ttest(ost_df["무게3"], 320, alternative = "less").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.983	99	less	0.025	[-inf, 319.78]	0.198	1.447	0.627

4.3 차이가 없는 경우

[22]
```
# two-sided
pg.ttest(ost_df["무게4"], 320, alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	1.286	99	two-sided	0.202	[319.52, 322.24]	0.129	0.246	0.247

[23]
```
# greater
pg.ttest(ost_df["무게4"], 320, alternative = "greater").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	1.286	99	greater	0.101	[319.74, inf]	0.129	0.492	0.356

4.4 차이가 있는 경우(greater)

[24]
```
# two-sided
pg.ttest(ost_df["무게5"], 320, alternative = "two-sided").round(3)
```

T	dof	alternative	p-val	CI95%	cohen-d	BF10	power

✓ 0초 오전 9:21에 완료됨

4.t-test

[22]

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	1.286	99	two-sided	0.202	[319.52, 322.24]	0.129	0.246	0.247

[23]
```
# greater
pg.ttest(ost_df["무게4"], 320, alternative = "greater").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	1.286	99	greater	0.101	[319.74, inf]	0.129	0.492	0.356

4.4 차이가 있는 경우(greater)

[24]
```
# two-sided
pg.ttest(ost_df["무게5"], 320, alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	4.295	99	two-sided	0.0	[321.57, 324.25]	0.429	422.533	0.989

[25]
```
# greater
pg.ttest(ost_df["무게5"], 320, alternative = "greater").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	4.295	99	greater	0.0	[321.78, inf]	0.429	845.067	0.996

5.정규성 검정

5.1 정규성 검정

[11]
```
pg.normality(ost_df["무게1"])
```

	W	pval	normal
무게1	0.63365	1.388964e-14	False

✓ 0초 오전 9:21에 완료됨

One Sample t-test

❖ 결과해석

Descriptives

	N	Mean	Median	SD	SE
무게1	100	317.91	317.40	6.77	0.68

$p - value$: 귀무가설($H_0: \mu = 320$)이 맞을 확률

One Sample T-Test

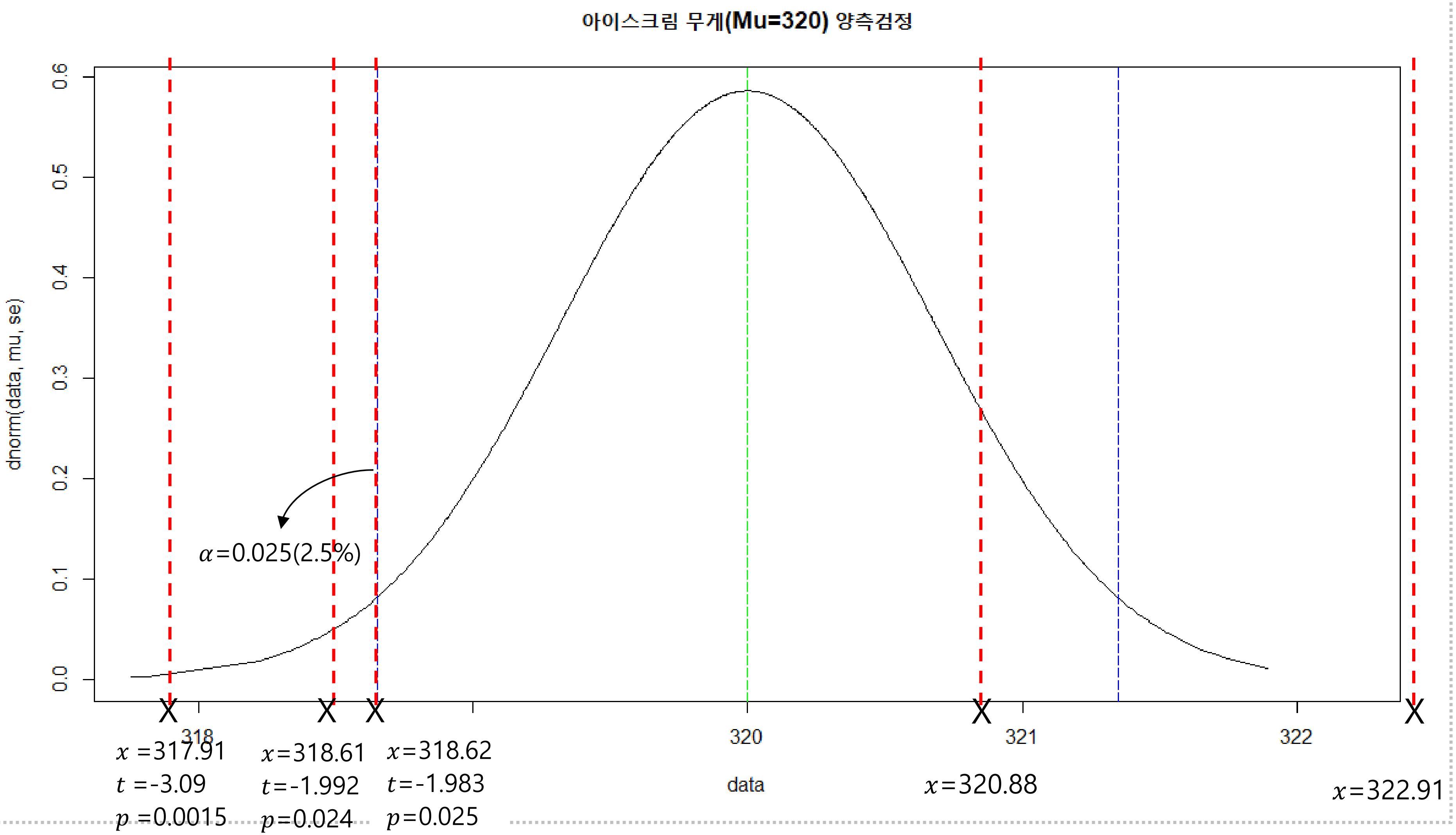
		Statistic	df	p	Mean difference
무게1	Student's t	−3.09	99.00	0.003	−2.09

Note. H_a $\mu \neq 320$

$H_0: \mu = 320$

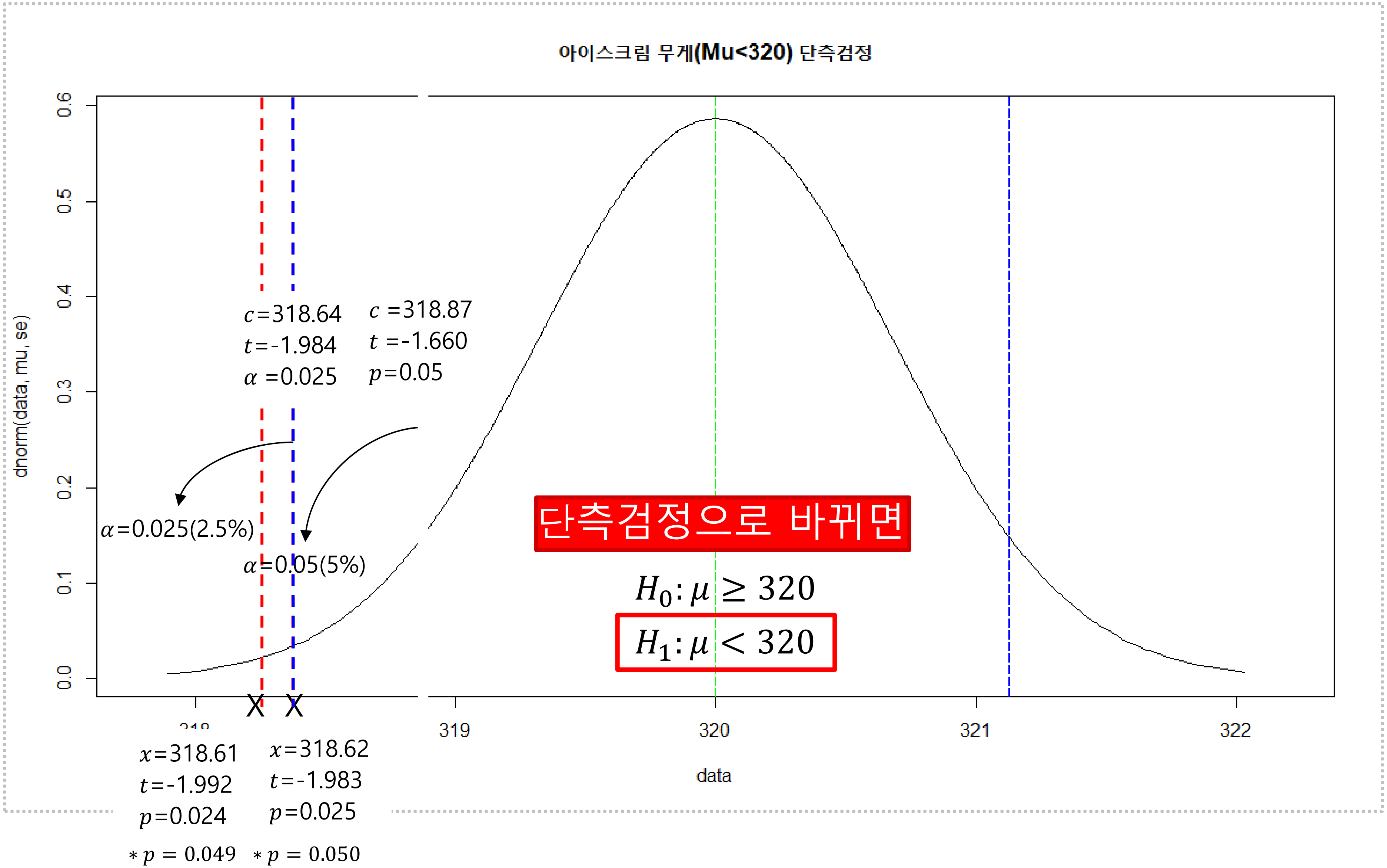
1.00
0.999
…
0.051
귀무가설 채택 $H_0: \mu = 320$
0.05
0.04
0.03
0.02
0.01
연구가설 채택 $H_1: \mu \neq 320$
0.00

One Sample t-test



$*p = 0.003$ $*p = 0.049$ $*p = 0.050$

One Sample t-test



5.가정검정

```
[15] ost_df.drop(ost_df[filter].index, inplace = True)
```

```
[16] pg.normality(ost_df).T.round(3)

     # 4.1로 다시 분석
```

	무게1	무게2	무게3	무게4	무게5	무게6
W	0.981564	0.985788	0.985853	0.984864	0.981564	0.964417
pval	0.175659	0.36139	0.365223	0.310514	0.175659	0.008408
normal	True	True	True	True	True	False

5.3 비모수 통계

```
[26] # 비모수통계
     pg.wilcoxon(ost_df["무게6"] - 320, alternative = "two-sided").round(3)
```

	W-val	alternative	p-val	RBC	CLES
Wilcoxon	1377.0	two-sided	0.0	-0.444	NaN

```
[27] # 모수통계 결과와 비교
     pg.ttest(ost_df["무게6"], 320, alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.805	99	two-sided	0.0	[316.08, 318.77]	0.381	80.297	0.965

6.검증결과 그래프

```
[28] from scipy.stats import norm # 정규분포

     x_data = np.linspace(317, 323, 200)
```

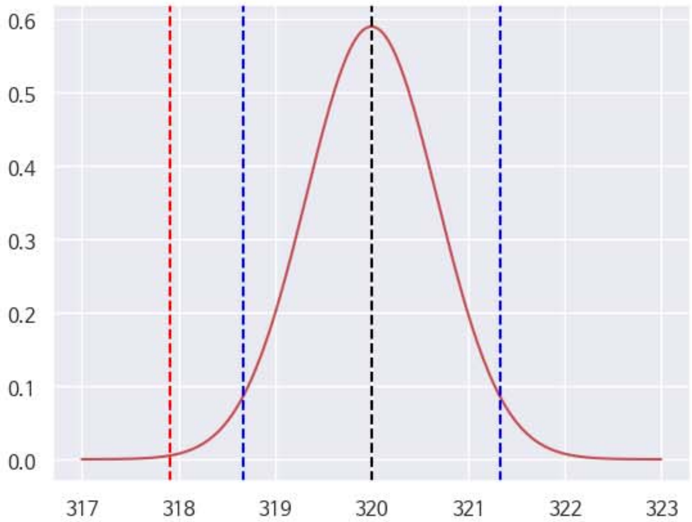
✓ 0초 오전 9:25에 완료됨

6.검증결과 그래프

6.검증결과 그래프

[28]

```
from scipy.stats import norm # 정규분포

x_data = np.linspace(317, 323, 200)

mu = 320   # 모집단 평균
x = 317.91 # 표본평균
se = 6.77/np.sqrt(100) # 표준오차(표준편차/sqrt(n)))

plt.plot(x_data, norm.pdf(x_data, loc = mu, scale = se), 'r-')
plt.axvline(x = mu, color='black', linestyle='--')
plt.axvline(x = mu - 1.96 * se, color='blue', linestyle='--')
plt.axvline(x = mu + 1.96 * se, color='blue', linestyle='--')
plt.axvline(x = 317.91, color='red', linestyle='--')
plt.show()
```



✓ 0초 오전 9:25에 완료됨

7.단일모집단 비율검정(proportion)

7.단일모집단 비율검정(proportion)

[29]
```python
# One Sample T Test of Proportion
from statsmodels.stats.proportion import proportions_ztest

z, p = proportions_ztest(count = 50,
                         nobs = 500,
                         value = 0.09)
print('z : {}, p : {}'.format(z, p))
```

```
z : 0.7453559924999305, p : 0.456056540250255566
```

[30]
```python
# 이항분포로 검정 n*p < 5 일때
stats.binom_test([50, 450], p = 0.09, alternative="two-sided")
```

```
<ipython-input-30-1e62f8be38b0>:2: DeprecationWarning: 'binom_test' is deprecated in favour of 'binomtest' from version 1.7.0 and will be removed in Scipy 1.12.0.
  stats.binom_test([50, 450], p = 0.09, alternative="two-sided")
0.4341018177288992
```

8.동등성(Equivalence test)

[31]
```python
pg.tost(ost_df["무게1"],
        y = 320,
        bound = 3)
```

	bound	dof	pval
TOST	3	99	0.091325

[32]
```python
pg.tost(ost_df["무게4"],
        y = 320,
        bound = 3)
```

bound	dof	pval

✓ 0초 오전 9:26에 완료됨

One Sample t-test

❖ G대학 앞 점포에서 파는 아이스크림의 무게(317.91g)는 B아이스크림회사에서 발표한 파인트의 무게(320g)보다 통계적으로 유의하게 적었다(t=-3.09, p= 0.003).(무게1)

	M(SD)	t	p
무게	317.91 (6.77)	-3.09	0.003

One Sample t-test

❖ G대학 앞 점포에서 파는 아이스크림의 무게(320.88g)는 B아이스크림회사에서 발표한 파인트의 무게(320g)와 차이가 없었다(t=1.286, p= 0.202). (무게4)

	M(SD)	t	p
무게	320.88(6.84)	1.286	0.202

One Sample t-test (비모수일때)

- 먼저, Shapiro test를 한 결과, 정규분포가 아닌 것으로 나타나(w=0.964, p=0.008), 비모수통계분석인 Wilcoxon Rank 분석을 실시하였다.
- G대학 앞 점포에서 파는 아이스크림의 무게가 320g인지 검증할 결과, 평균은 317.426, 중앙값은 317.9로 나타났으며, 파인트의 무게(320g)보다 통계적으로 유의하게 적었다(w=5,050, p= 0.000).

	M(SD)	W	p
무게	317.43(6.76)	5,050	0.000

연습문제

연습문제1

❖ 문제의 정의

- G식품에서는 햄버거의 칼로리를 연구하고 있다.
- 햄버거의 칼로리가 500kcal인지 검증해 보세요.
- 04_3.calorie.csv
- 1. 칼로리1, 칼로리2, 칼로리3은 500kcal인가?
- 2. 칼로리1,2,3은 정규분포가정을 만족하는가? 만족하지 않다면 이상치를 제거하고 다시 분석하세요.
- 3. 만약 칼로리2, 칼로리3은 500kcal보다 큰지 검증한다면 크다고 할 수 있는가?

	칼로리1	칼로리2	칼로리3
1	509	511	512
2	491	493	494
3	501	503	504
4	502	504	505
5	498	500	501
6	503	505	506
7	497	499	500
8	490	492	493
9	504	506	507
10	502	504	505
11	499	501	502
12	508	510	511
13	504	506	507
14	500	502	503
15	494	496	497
16	502	504	505
17	501	503	504
18	501	503	504

연습문제2

❖ 문제의 정의

- G제약회사에서는 새로운 진통제를 개발하였다.
- 새로운 진통제의 지속효과가 300분인지 검증해 보세요.
- 04_4.painkiller.csv
- 1. 지속시간1, 지속시간2, 지속시간3은 300분인가?
- 2. 지속시간1, 지속시간2, 지속시간3 은 정규분포가정을 만족하는가? 만족하지 않다면 비모수통계로 다시 분석하세요.
- 3. 만약, 지속시간3이 300분보다 작은지 검증한다면 작다고 할 수 있는가?

	지속시간1	지속시간2	지속시간3
1	299	295	294
2	300	296	295
3	294	290	294
4	294	290	295
5	296	292	296
6	297	293	292
7	298	294	293
8	298	294	293
9	299	295	294
10	300	296	295
11	301	297	296
12	301	297	296
13	301	297	296
14	301	297	296
15	302	298	297
16	302	298	297
17	302	298	297
18	303	299	298

연습문제3

❖ 문제의 정의

- G대학에서는 재학생 만족도 조사를 실시하였다.
- 교양만족도, 전공만족도, 비교과만족도, 전체만족도가 50점인지 검증해 보세요.
- 04_5.Education.csv
- 1. 교양만족도, 전공만족도, 비교과만족도, 전체만족도는 50점인가?
- 2. 정규분포가정을 만족하는가? 만족하지 않다면 비모수 통계를 사용하세요.
- 3. 전공만족도는 50점 보다 큰지 검증한다면 크다고 할 수 있는가?
- 4. 비교과만족도는 50점 보다 작은지 검증한다면 작다고 할 수 있는가?

	교양만족도	전공만족도	비교과만...	전체만족도
1	47.6	40.5	40.0	46.7
2	33.3	35.7	33.3	33.6
3	50.0	52.4	50.0	50.4
4	35.7	28.5	40.0	36.1
5	54.7	92.8	43.3	56.2
6	39.3	53.6	55.0	48.9
7	46.4	46.4	45.0	46.9
8	42.9	67.9	35.0	53.2
9	42.9	50.0	25.0	37.4
10	32.1	28.6	30.0	31.5
11	50.0	50.0	50.0	48.7
12	50.0	50.0	50.0	50.9
13	78.6	64.3	70.0	73.9
14	42.9	53.6	25.0	48.7
15	42.8	73.8	46.6	49.3
16	80.5	90.5	76.6	85.1
17	50.0	46.4	25.0	34.3
18	25.0	32.1	40.0	34.8
19	39.3	50.0	35.0	46.8
20	39.3	57.1	20.0	49.3
21	28.6	39.3	25.0	30.0
22	50.0	75.0	45.0	53.3
23	85.7	92.9	20.0	59.3
24	25.0	46.4	30.0	44.8
25	50.0	50.0	50.0	48.1

II. Independent Sample T-test

두 모평균 검정

Independent Sample t-test

❖ 문제의 정의

– 이교수는 이번에 자동차 타이어를 교체하려고 하는데 수명이 긴 타이어로 교체하려고 한다.

– 시중에는 A회사의 타이어와 B회사의 타이어가 있는데, 이 교수는 이 중에서 어느 타이어를 골라야 하는가?

– 05_1.IST.csv

❖ 가설

– 귀무가설(H_0): A타이어회사와 B타이어회사의 타이어수명은 차이가 없다.

$$H_0: \mu_1 = \mu_2 \qquad H_0: \mu_1 - \mu_2 = 0$$

– 연구가설(H_1): A타이어회사와 B타이어회사의 타이어수명은 차이가 있다.

$$H_1: \mu_1 \neq \mu_2 \qquad H_1: \mu_1 - \mu_2 \neq 0$$

Independent Sample t-test

❖ Independent Sample t-test의 통계적 가정

– 두 집단 분포가 모두 정규분포(모수통계)로 가정 → 표본이므로 t분포 가정

$$\bar{X}_1 \sim N(\mu_1, \frac{\sigma^2}{n}) \qquad \bar{X}_2 \sim N(\mu_2, \frac{\sigma^2}{n})$$

– 두 집단의 분포가 같다고 가정: 등분산 → 분산이 같지 않으면 이분산 분석방법(Welch's test)로 분석

$$t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t_{(n_1+n_2-2, \frac{0.05}{2})}$$

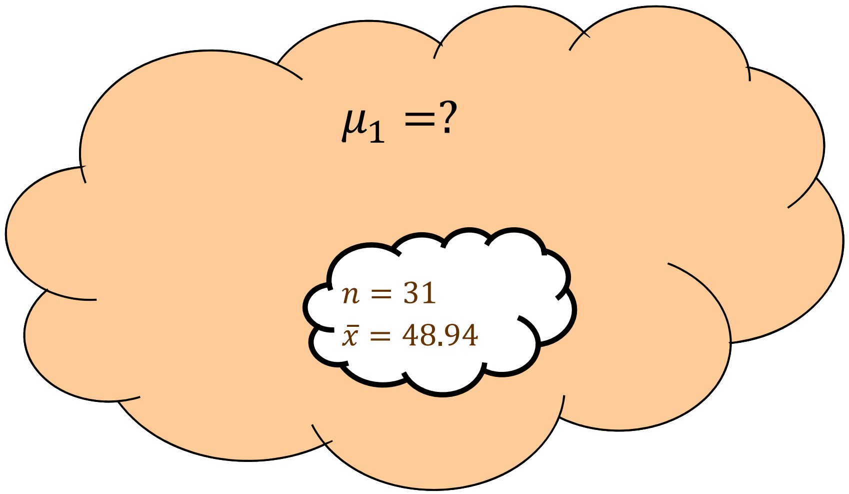
$$* \ t_{cal} = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \sim t_{n-1}$$

– 표본이 작으면서 이상점이 많을 경우: 비모수적 통계분석 사용

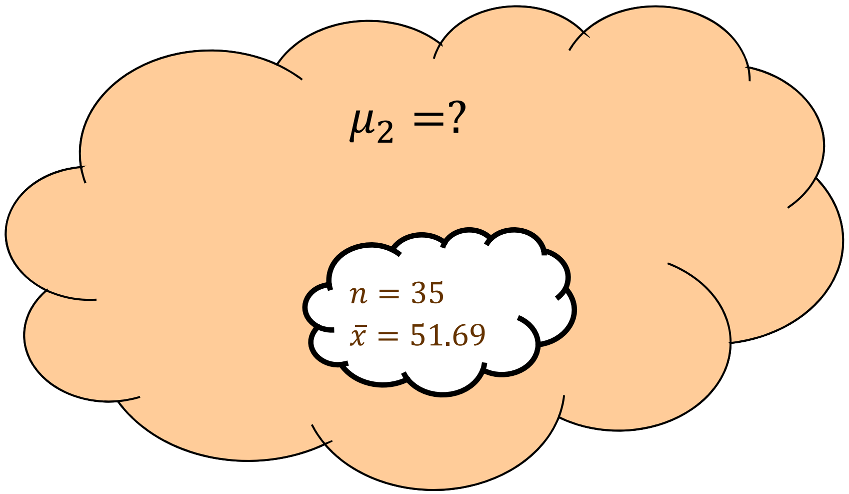
Independent Sample t-test

❖ 모집단에서 표본추출

A타이어



B타이어



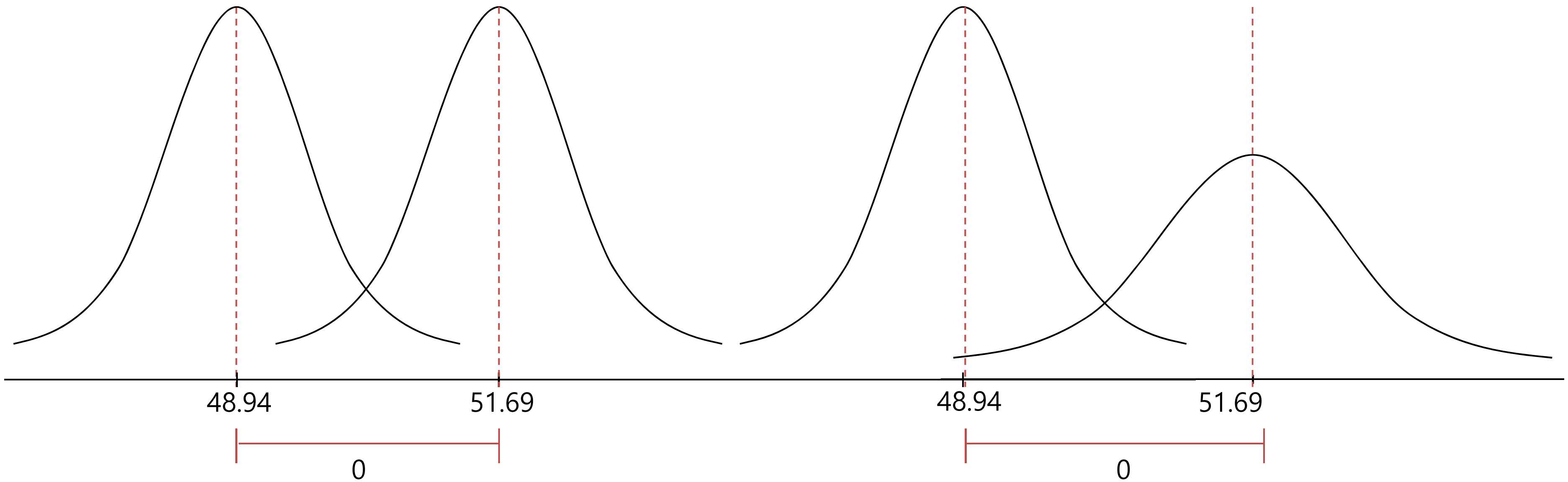
$$\bar{X}_1 \sim N(\mu_1, \frac{\sigma^2}{n})$$

$$\bar{X}_2 \sim N(\mu_2, \frac{\sigma^2}{n})$$

Independent Sample t-test

$\bar{X}_1 \sim N(\mu_1, \frac{\sigma^2}{n})$ $\bar{X}_2 \sim N(\mu_2, \frac{\sigma^2}{n})$

등분산 이분산



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

Independent Sample t-test

❖ 검정통계량 (등분산(equal variances)일 경우)

$$t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$* \, s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_1 - 1)s_2^2}{n_1 + n_2 - 2}, (\text{공통분산})$$

$$* \, d.f = n_1 + n_2 - 2, (\text{자유도})$$

❖ 검정통계량 (이분산(unequal variances)일 경우)

$$t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

$$* \, d.f = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\sqrt{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}}$$

Independent Sample t-test

- 통계치 ($\bar{X}_1$)
  - 표본 (n) : 31
  - 표본평균 ($\bar{X}_1$): 48.94
  - 표본표준편차 (s): 3.33, 표준오차 ($\frac{s}{\sqrt{n}}$): 0.60

- 통계치 ($\bar{X}_2$)
  - 표본 (n) : 35
  - 표본평균 ($\bar{X}_2$): 51.69
  - 표본표준편차 (s): 3.77, 표준오차 ($\frac{s}{\sqrt{n}}$): 0.64

Independent Sample t-test

❖ 임계치 (등분산(equal variances)일 경우)

$$x_{critical} = (\mu_1 - \mu_2) \pm t_{(n_1+n_2-2,\frac{0.05}{2})} \times s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

$$= 0 \pm 2.00 \times 3.57 \times \sqrt{\frac{1}{31} + \frac{1}{35}}$$

$$= 0 \pm (2.00)(3.57)(0.247) = \pm 1.76$$

$$= [-1.76, +1.76]$$

$$* \, t_{\left(n_1+n_2-2,\frac{0.05}{2}\right)} = t_{(64,0.025)} = 2.00$$

$$* \, s_p^2 = \frac{(n_1-1)s_1^2 + (n_1-1)s_2^2}{n_1+n_2-2} = \frac{(31-1)(3.33)^2 + (35-1)(3.77)^2}{31+35-2} = 12.75$$

$$* \, s_p = \sqrt{12.75} = 3.57$$

Independent Sample t-test

❖ 검정통계량 (등분산(equal variances)일 경우)

$$t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} = \frac{(48.94 - 51.69) - (0)}{0.88} = -3.12 < -2.0$$

$$* \, t_{critical} = t_{\left(n_1+n_2-2, \frac{0.05}{2}\right)} = \pm 2.00$$

$$* \, t_{cal} = \frac{(\bar{x}_1 - \bar{x}_2) - (\mu_1 - \mu_2)}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \sim t_{\left(n_1+n_2-2, \frac{0.05}{2}\right)}$$

❖ 유의확률(p-value)

$$p - value = P(|t| > 3.12) = 0.003 < 0.05$$

Independent Sample t-test

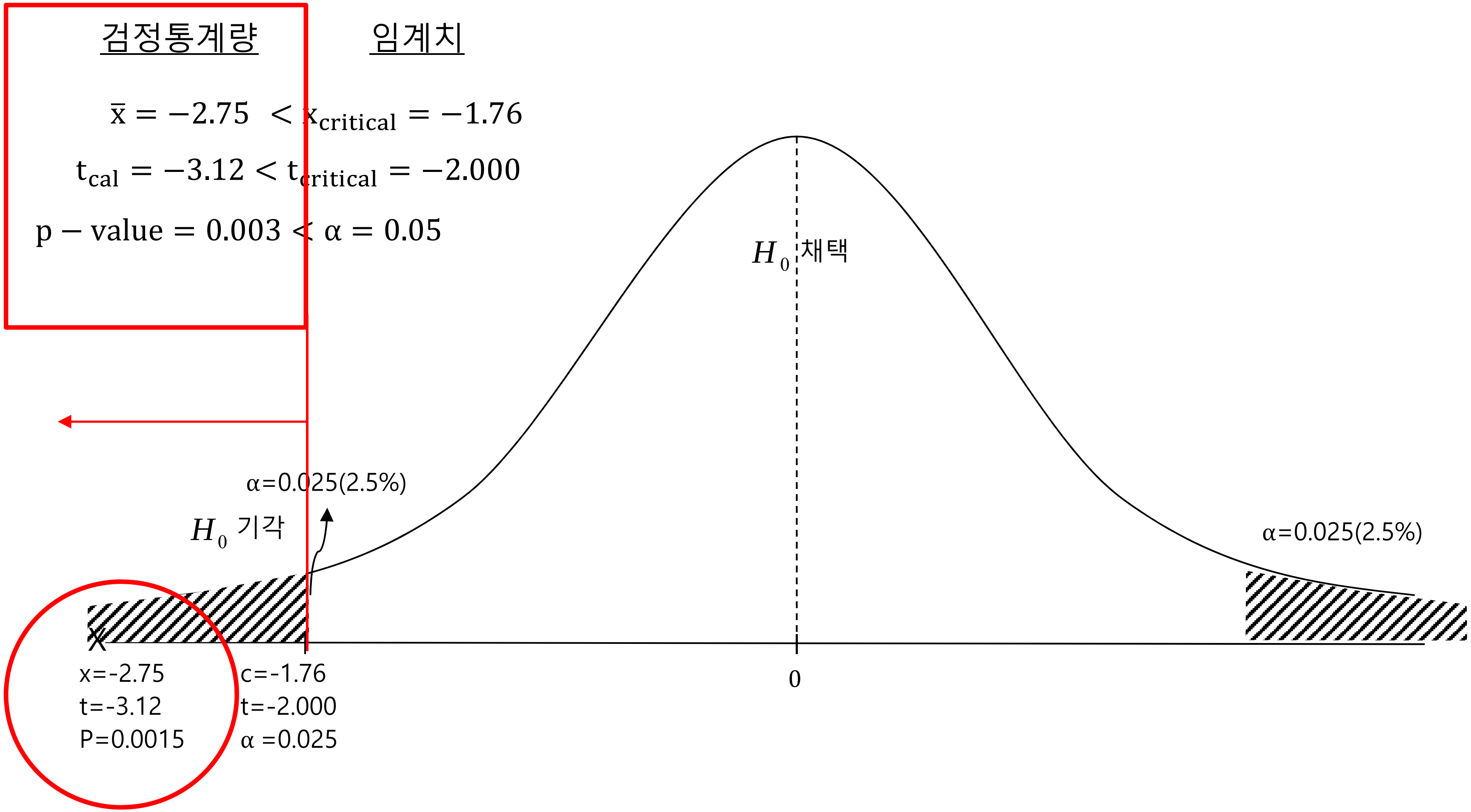
❖ 검정결과

검정통계량 　　 임계치

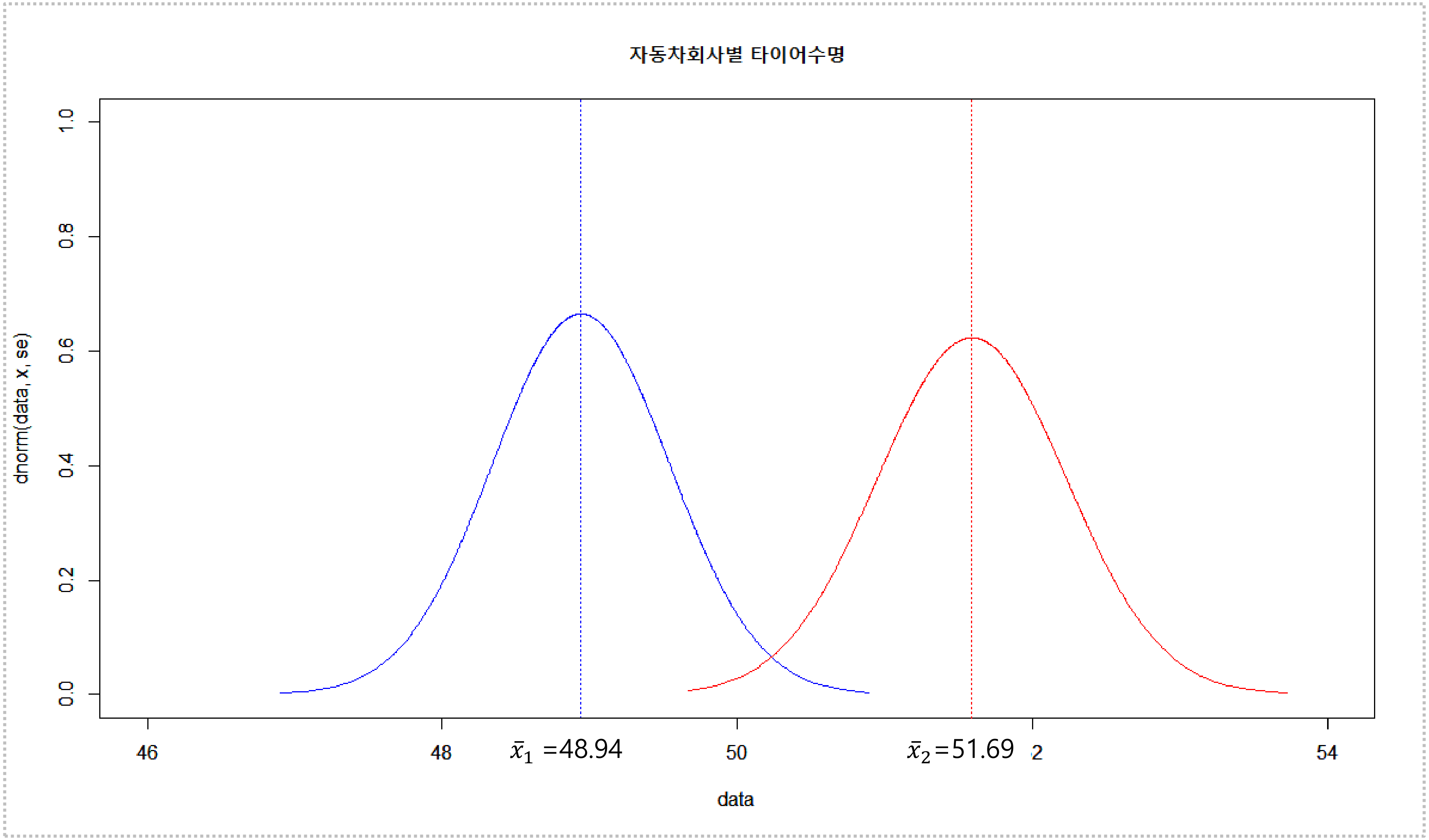
$$\bar{x} = -2.75 < x_{critical} = -1.76$$

$$t_{cal} = -3.12 < t_{critical} = -2.000$$

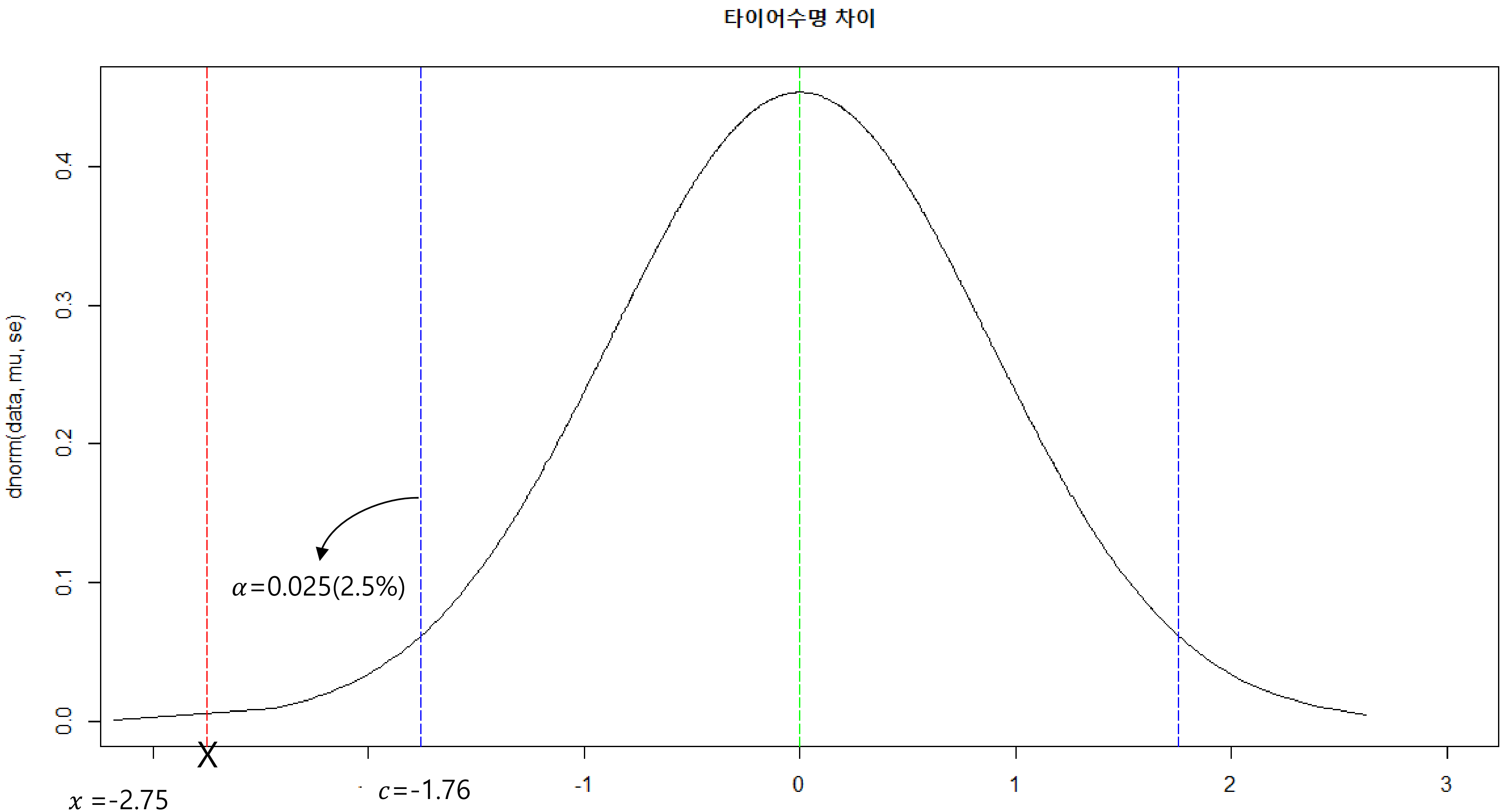
$$p - value = 0.003 < \alpha = 0.05$$



Independent Sample t-test



Independent Sample t-test



x =-2.75
t =-3.12
p =0.0015

c=-1.76
t=-2.00
α =0.025

$* \, p = 0.003$

두개 모비율 비교

모비율 가설검정

❖ 두 개의 모비율 검정

– 당뇨병환자에서 아스피린 사용현황 및 동반질환에 대한 분석결과이다. 아스피린 복용한 사람과 그렇지 않은 사람간에 뇌경색(Cerebral infarct) 발생비율이 차이가 있었는가?

Table 3. Comparison of Clinical Characteristics between Aspirin User and Non-user in 2001

	Aspirin user (2,065)	Aspirin non-user (27,949)	*P* value
Sex			
Men (%)	50.6	57.1	< 0.001
Women (%)	49.4	42.9	< 0.001
Mean age (S.D.)(yrs)	60.4 ± 9.6	56.4 ± 10.4	< 0.001
DM treatment (%)			
OHA	95.8	95.4	ns
Insulin alone	0.9	1.5	
OHA + insulin	3.3	3.1	
Associated CVD (%)			
Hypertension	66.0	27.9	< 0.001
Hypercholesterolemia	20.1	9.7	< 0.001
Cerebral infarct	27.2	10.7	< 0.001
Cerebral hemorrhage	4.2	2.4	< 0.001
Coronary disease	23.3	2.4	< 0.001
No associated CVD	12.1	58.8	< 0.001

* CVD, cardiovascular disease.

고혈압(Hypertension), 고지혈증(Hypercholesterolemia), 및 뇌출혈(Cerebral hemorrhage)

출처: 박이병외(2006), 당뇨병환자에서 아스피린 사용현황 및 동반질환:건강보험자료 분석결과, 당로병, 제30권, 제5호

모비율 가설검정

❖ 검정통계량 (test statistics)

$$\bar{p} = \frac{x_1 + x_2}{n_1 + n_2}$$

$$t_{cal} = \frac{(p_1 - p_2) - (\pi_1 - \pi_2)}{\sqrt{\frac{\bar{p}(1-\bar{p})}{n_1}} + \sqrt{\frac{\bar{p}(1-\bar{p})}{n_2}}}$$

$$(\pi_1 - \pi_2) = 0\text{이면},\ t_{cal} = \frac{(p_1 - p_2)}{\sqrt{\bar{p}(1-\bar{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

$$* \, t_{cal} = \frac{(p_1 - p_2)}{\sqrt{\bar{p}(1-\bar{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim t_{n-1}$$

❖ 신뢰구간 계산

$$\bar{p}_{conf} = (p_1 - p_2) \pm t_{\frac{a}{2}} \sqrt{\frac{p_1(1-p_1)}{n_1} + \frac{p_2(1-p_2)}{n_2}}$$

LG Life's Good

모비율 가설검정

❖ 임계치

$$\bar{p} = \frac{x_1 + x_2}{n_1 + n_2} = \frac{87 + 671}{2{,}065 + 27{,}949} = 0.025$$

$$\bar{p}_{critical} = (p_1 - p_2) \pm t_{\frac{\alpha}{2}} \sqrt{\bar{p}(1-\bar{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)} = 0 \pm 1.96\sqrt{0.025(1-0.025)\left(\frac{1}{2{,}065} + \frac{1}{27{,}949}\right)}$$
$$= 0 \pm 1.96\sqrt{0.025(0.0005)} = 0 \pm 1.96(0.036) = [-0.007, 0.007]$$

❖ 검정통계량 (test statistics)

$$t_{cal} = \frac{(p_1 - p_2)}{\sqrt{\bar{p}(1-\bar{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} = \frac{(0.042 - 0.024)}{\sqrt{0.025(1-0.025)\left(\frac{1}{2{,}065} + \frac{1}{27{,}949}\right)}} = \frac{(0.018)}{\sqrt{0.025(0.0005)}} = 5.030$$

❖ 유의확률(p-value) 계산

$$p - value = P(|t| > 5.030) = 0.000$$

모비율 가설검정

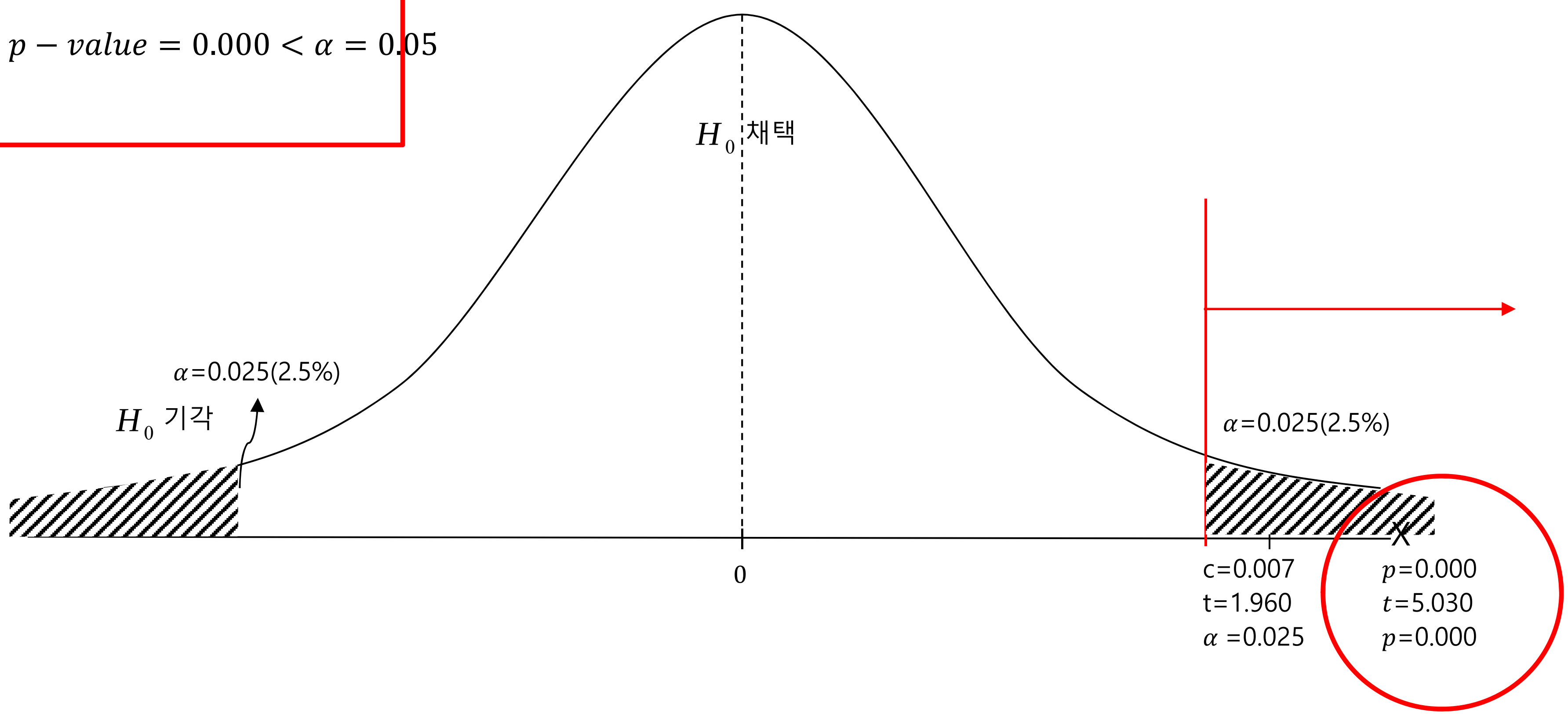
❖ 검정결과

검정통계량 임계치

$$\bar{x} = 0.025 > x_{critical} = 0.007$$

$$t_{cal} = 5.030 > t_{critical} = 1.960$$

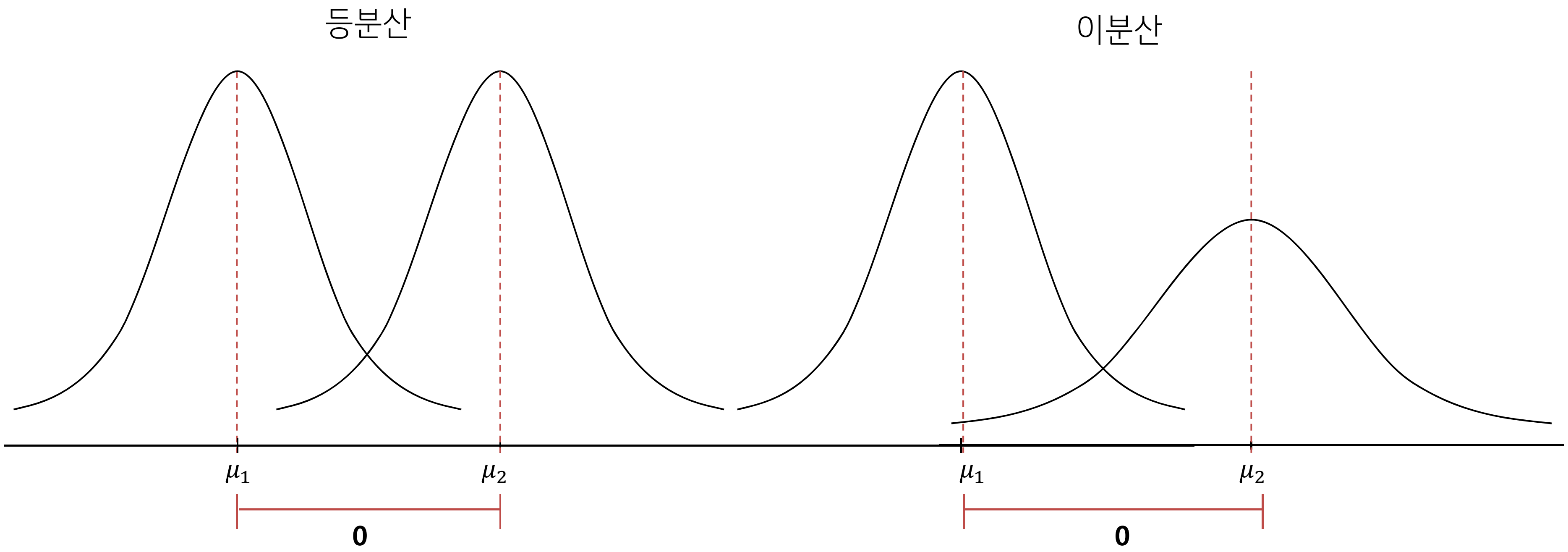
$$p - value = 0.000 < \alpha = 0.05$$



두개 모분산 비교

모분산 가설검정

$$\bar{X}_1 \sim N(\mu_1, \frac{\sigma^2}{n}) \qquad \bar{X}_2 \sim N(\mu_2, \frac{\sigma^2}{n})$$



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

모분산 가설검정

❖ 문제의 정의

- Independent t-test를 실시하기 위해 두 집단 분산의 동질성을 검정하고자 한다.
- 두 집단의 분포가 같다고 가정: 등분산 → 분산이 같지 않으면 이분산 분석방법(Welch's test)로 분석해야 한다.

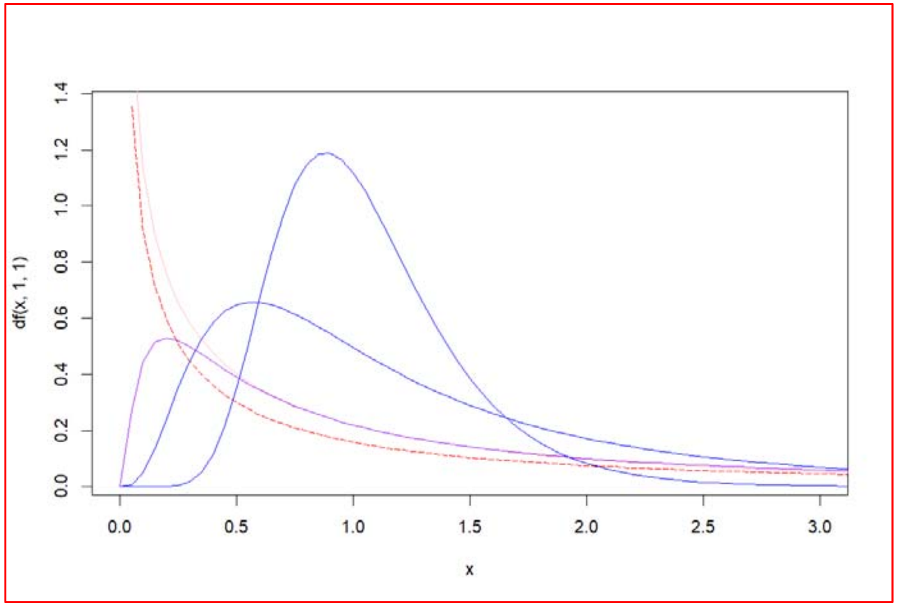
- 가설

$$H_0: \sigma_1^2 = \sigma_2^2 \text{ or } \frac{\sigma_1^2}{\sigma_2^2} = 1$$

$$H_1: \sigma_1^2 \neq \sigma_2^2 \text{ or } \frac{\sigma_1^2}{\sigma_2^2} \neq 1$$

- 임계치 (양측검정)

$$F_{cal} = \frac{s_1^2}{s_2^2}, df_1 = n_1 - 1, df_2 = n_2 - 1$$



모분산 가설검정

❖ 통계치 ($\bar{X}_1$)

- 표본 (n) : 31
- 표본평균 ($\bar{X}_1$): 48.94
- 표본표준편차 (s): 3.33, 표준오차 ($\frac{s}{\sqrt{n}}$): 0.60

❖ 통계치 ($\bar{X}_2$)

- 표본 (n) : 35
- 표본평균 ($\bar{X}_2$): 51.69
- 표본표준편차 (s): 3.77, 표준오차 ($\frac{s}{\sqrt{n}}$): 0.64

❖ 임계치 (양측검정)

$$F_L = 0.489 < F_{cal} < F_R = 2.011$$

모분산 가설검정

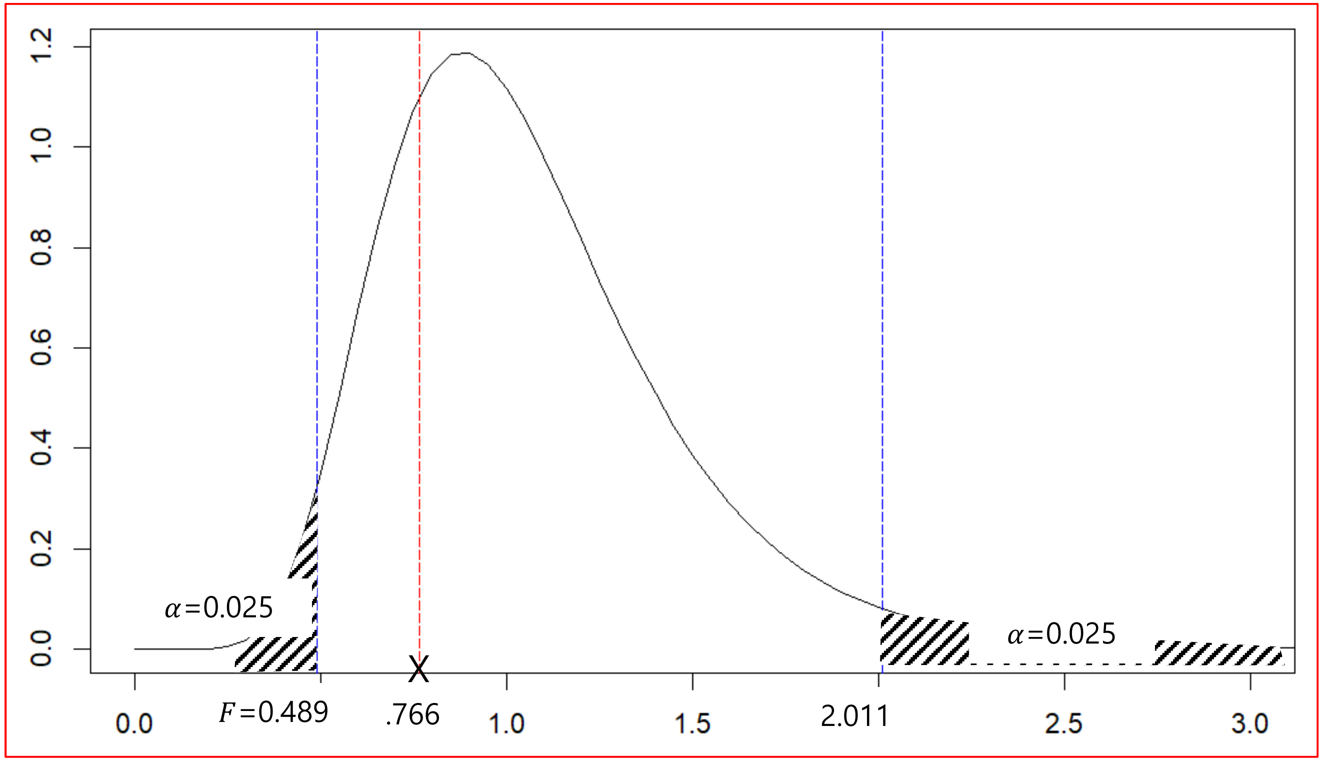
❖ 검정통계량

$$F_{cal} = \frac{s_1^2}{s_2^2} = \frac{(3.33)^2}{(3.77)^2} = \frac{10.89}{14.22} = 0.766$$

$$F_L = 0.489 < F_{cal} = 0.766 < F_R = 2.011$$

❖ 유의수준

$$p - value = 0.76$$



Independent Sample t-test 분석절차

실습

05_1.Independent Sample t-test

05_1.Independent Sample t-test

- https://pingouin-stats.org/build/html/generated/pingouin.ttest.html#pingouin.ttest

1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기
    !sudo apt-get install -y fonts-nanum
    !sudo fc-cache -fv
    !rm ~/.cache/matplotlib -rf
```

```
[ ] !pip install pingouin

    #  *** 세션 다시 시작
```

```
[1] # 1.기본
    import numpy as np  # numpy 패키지 가져오기
    import matplotlib.pyplot as plt # 시각화 패키지 가져오기
    import seaborn as sns # 시각화

    # 2.데이터 가져오기
    import pandas as pd # csv -> dataframe으로 전환

    # 3.통계분석 package
    import pingouin as pg
    from scipy import stats
    import statsmodels.api as sm
```

```
[2] # 기본세팅
    # 테마 설정
    sns.set_theme(style = "darkgrid")
```

✓ 0초 오전 9:31에 완료됨

2.데이터 불러오기

2.데이터 불러오기

2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

```
[3] ist_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/05_1.IST.csv', encoding="cp949")
    ist_df.head()
```

	회사	수명1	수명2	수명3	수명4	수명5
0	1	50	52	51	50	51
1	1	52	54	53	52	53
2	2	51	51	51	51	51
3	2	52	52	52	52	52
4	1	52	54	53	50	53

Next steps: View recommended plots

2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

```
[4] ist_df['회사'].replace({1:'A타이어', 2:'B타이어',}, inplace=True)
    ist_df['회사'] = ist_df['회사'].astype('category')

    ist_df.head()
```

	회사	수명1	수명2	수명3	수명4	수명5
0	A타이어	50	52	51	50	51
1	A타이어	52	54	53	52	53

✓ 0초 오전 9:31에 완료됨

2.데이터 불러오기

[4]

0	A타이어	50	52	51	50	51
1	A타이어	52	54	53	52	53
2	B타이어	51	51	51	51	51
3	B타이어	52	52	52	52	52
4	A타이어	52	54	53	50	53

Next steps: View recommended plots

2.3 자료구조 살펴보기

[5]
```
ist_df.shape
```
```
(66, 6)
```

[6]
```
ist_df.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66 entries, 0 to 65
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   회사      66 non-null     category
 1   수명1     66 non-null     int64
 2   수명2     66 non-null     int64
 3   수명3     66 non-null     int64
 4   수명4     66 non-null     int64
 5   수명5     66 non-null     int64
dtypes: category(1), int64(5)
memory usage: 2.9 KB
```

[7]
```
ist_df.columns
```
```
Index(['회사', '수명1', '수명2', '수명3', '수명4', '수명5'], dtype='object')
```

3.기술통계

✓ 0초 오전 9:31에 완료됨

3.기술통계

3.기술통계

[8]
```
# 그룹별 기술통계
ist_df.groupby('회사')["수명1"].describe().round(2)
```

회사	count	mean	std	min	25%	50%	75%	max
A타이어	31.0	48.94	3.33	42.0	47.0	49.0	51.0	56.0
B타이어	35.0	51.69	3.77	44.0	50.0	52.0	55.0	59.0

[9]
```
# 분석변수가 여러개 일 때
num_feature = ['수명1', '수명2', '수명3', '수명4', '수명5']
for num in num_feature:
    print("----", num, "----")
    results = ist_df.groupby('회사')[num].describe().round(2)
    print(results, "\n")
```

```
---- 수명1 ----
       count   mean   std   min   25%   50%   75%   max
회사
A타이어   31.0  48.94  3.33  42.0  47.0  49.0  51.0  56.0
B타이어   35.0  51.69  3.77  44.0  50.0  52.0  55.0  59.0

---- 수명2 ----
       count   mean   std   min   25%   50%   75%   max
회사
A타이어   31.0  50.94  3.33  44.0  49.0  51.0  53.0  58.0
B타이어   35.0  51.69  3.77  44.0  50.0  52.0  55.0  59.0

---- 수명3 ----
       count   mean   std   min   25%   50%   75%   max
회사
A타이어   31.0  50.03  3.19  42.0  48.0  50.0  52.0  57.0
B타이어   35.0  51.69  3.77  44.0  50.0  52.0  55.0  59.0

---- 수명4 ----
       count   mean   std   min   25%   50%   75%   max
회사
A타이어   31.0  48.71  1.97  44.0  47.0  49.0  50.0  52.0
```

✓ 0초 오전 9:31에 완료됨

4.t-test

4.t-test

4.1 차이가 있는 경우(two-sided)

[10]
```
x = ist_df['수명1'][ist_df['회사'] == 'A타이어']
y = ist_df['수명1'][ist_df['회사'] == 'B타이어']
```

[11]
```
# paired = True : paired sample t-test
# correction = False : 등분산일때
pg.ttest(x, y,
         paired = False,
         alternative = "two-sided",
         correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.124	64	two-sided	0.003	[-4.51, -0.99]	0.77	13.517	0.868

[12]
```
# 그래프
sns.catplot(x = "회사",
            y = "수명1",
            kind = "point",
            data = ist_df)
plt.show()
```

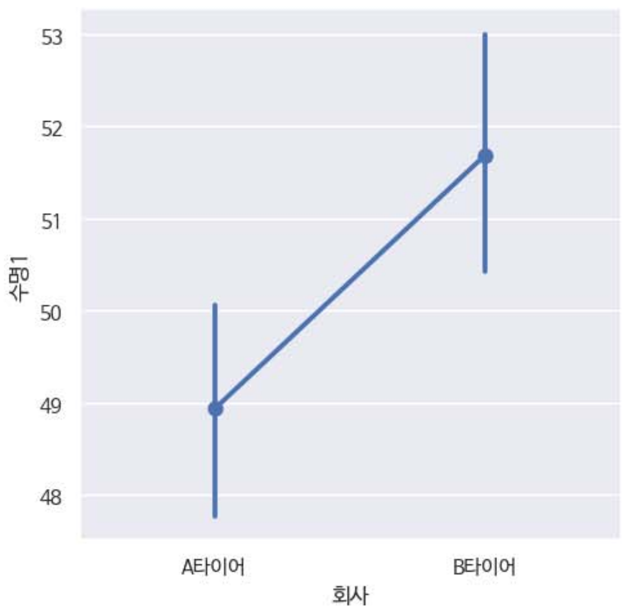
✓ 0초 오전 9:32에 완료됨

4.t-test

[11]

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.124	64	two-sided	0.003	[-4.51, -0.99]	0.77	13.517	0.868

[12]
```
# 그래프
sns.catplot(x = "회사",
            y = "수명1",
            kind = "point",
            data = ist_df)
plt.show()
```



4.2 차이가 없는 경우

```
[ ] x = ist_df['수명2'][ist_df['회사'] == 'A타이어']
```

✓ 0초 오전 9:32에 완료됨

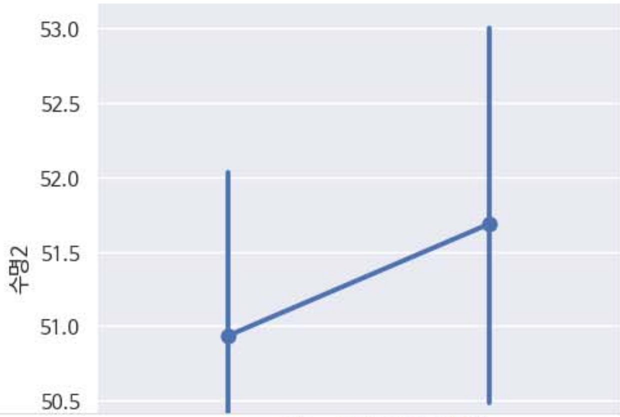
4.t-test

4.2 차이가 없는 경우

[13]
```
x = ist_df['수명2'][ist_df['회사'] == 'A타이어']
y = ist_df['수명2'][ist_df['회사'] == 'B타이어']

pg.ttest(x, y,
         paired = False,
         alternative = "two-sided",
         correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-0.852	64	two-sided	0.397	[-2.51, 1.01]	0.21	0.344	0.134

[14]
```
# 그래프
sns.catplot(x = "회사",
            y = "수명2",
            kind = "point",
            data = ist_df)
plt.show()
```



✓ 0초 오전 9:35에 완료됨

4.t-test

4.3 양측과 단측 검정 비교(less)

[15]
```python
# two-sided
x = ist_df['수명3'][ist_df['회사'] == 'A타이어']
y = ist_df['수명3'][ist_df['회사'] == 'B타이어']

pg.ttest(x, y,
         paired = False,
         alternative = "two-sided",
         correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.91	64	two-sided	0.061	[-3.38, 0.08]	0.471	1.166	0.469

[16]
```python
# less
pg.ttest(x, y,
         paired = False,
         alternative = "less",
         correction = False).round(3)
```

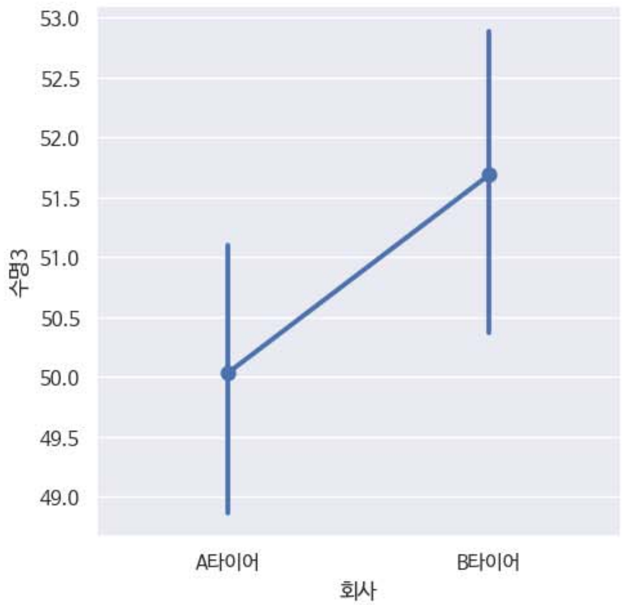
	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.91	64	less	0.03	[-inf, -0.21]	0.471	2.333	0.597

[17]
```python
# 그래프
sns.catplot(x = "회사",
            y = "수명3",
            kind = "point",
            data = ist_df)
plt.show()
```

✓ 0초 오전 9:35에 완료됨

4.t-test

```
[17] # 그래프
     sns.catplot(x = "회사",
                 y = "수명3",
                 kind = "point",
                 data = ist_df)
     plt.show()
```



5.등분산 검정

5.1 등분산 검정

```
[ ] # 등분산이면 지금까지 분석한 것이 문제 없음
```

✓ 0초 오전 9:35에 완료됨

Independent Sample t-test

❖ 결과해석

Group Descriptives

	Group	N	Mean	Median	SD	SE
수명1	A타이어	31	48.94	49.00	3.33	0.60
	B타이어	35	51.69	52.00	3.77	0.64

$p - value$: 귀무가설($H_0: \mu_1 = \mu_2$)이 맞을 확률

Independent Samples T-Test

		Statistic	df	p	Mean difference	SE difference
수명1	Student's t	−3.12	64.00	0.003	−2.75	0.88

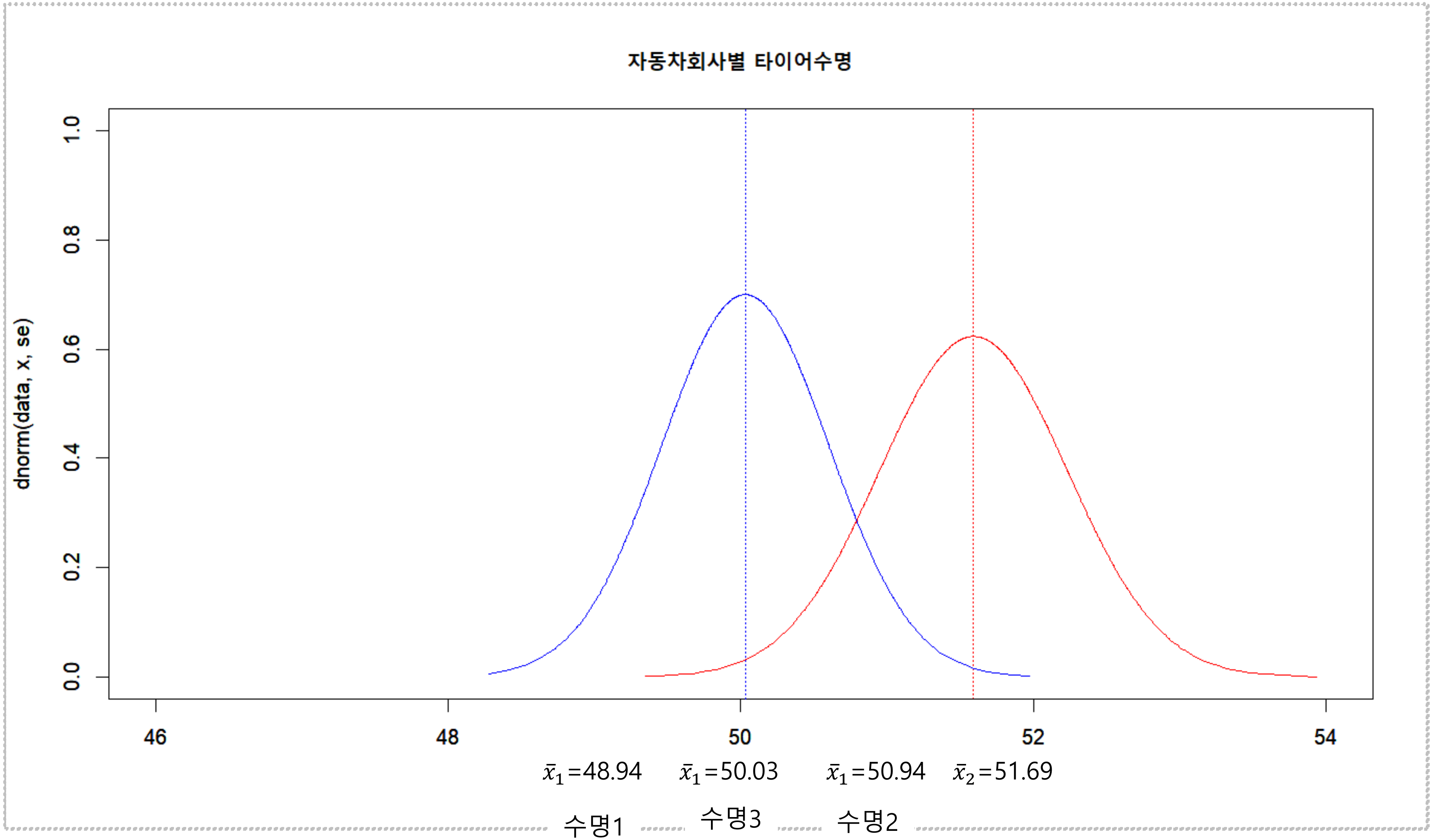
Note. Hₐ μ A타이어 ≠ μ B타이어

$H_0: \mu_1 = \mu_2$

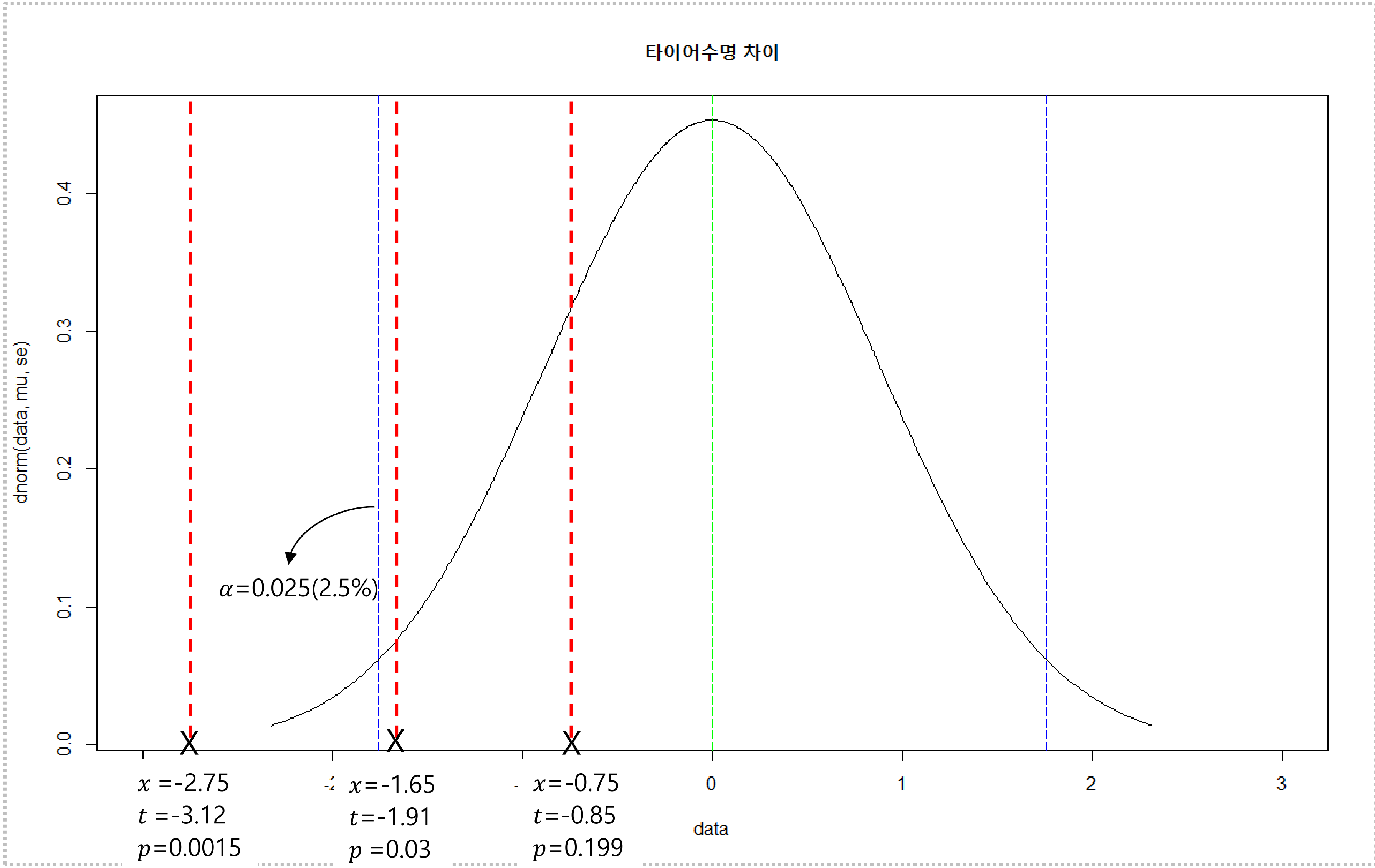
1.00
0.999
…
0.051
귀무가설 채택 $H_0: \mu_1 = \mu_2$
0.05
0.04
0.03
0.02
0.01
연구가설 채택 $H_1: \mu_1 \neq \mu_2$
0.00

LG Life's Good

Independent Sample t-test

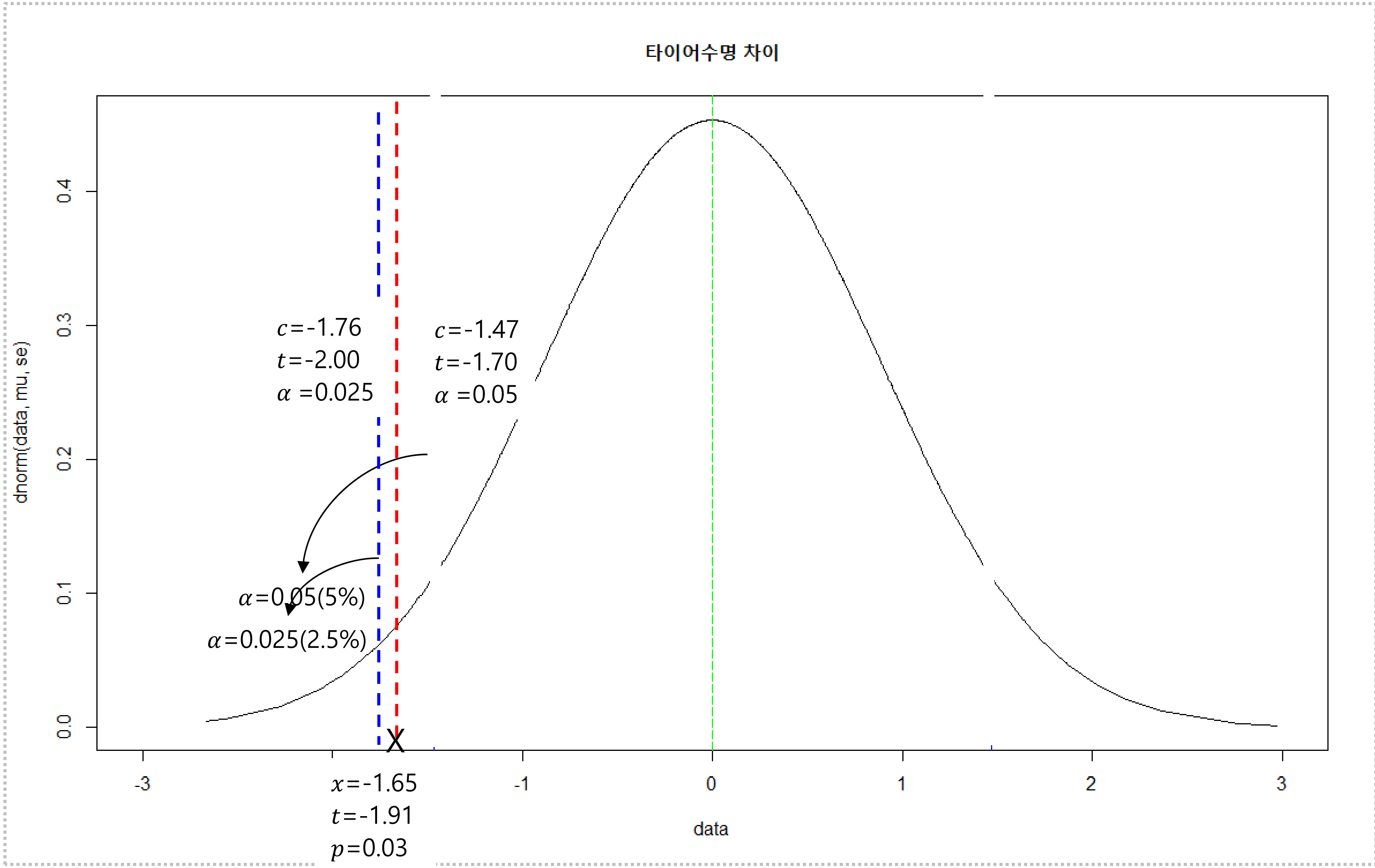


Independent Sample t-test



$*p = 0.003$ $*p = 0.061$ $*p = 0.397$

Independent Sample t-test



$* \, p = 0.061$

5.등분산 검정

5.등분산 검정

5.1 등분산 검정

[18]
```
# 등분산이면 지금까지 분석한 것이 문제 없음
pg.homoscedasticity(ist_df,
                    dv = "수명1",
                    group = "회사")
```

	W	pval	equal_var
levene	0.195988	0.659471	True

[19]
```
num_feature = ['수명1', '수명2', '수명3', '수명4', '수명5']
for num in num_feature:
    print("----", num, "----")
    results = pg.homoscedasticity(ist_df,
                                  dv = num,
                                  group = "회사")
    print(results, "\n")
```

```
---- 수명1 ----
               W      pval  equal_var
levene  0.195988  0.659471       True

---- 수명2 ----
               W      pval  equal_var
levene  0.195988  0.659471       True

---- 수명3 ----
               W      pval  equal_var
levene  0.400108  0.529287       True

---- 수명4 ----
              W      pval  equal_var
levene  7.02041  0.010141      False

---- 수명5 ----
```

✓ 0초 오전 9:36에 완료됨

5.등분산 검정

5.2 이분산일때(Welch's test)

```
[20] pg.homoscedasticity(ist_df,
                         dv = "수명4",
                         group = "회사")
```

	W	pval	equal_var
levene	7.02041	0.010141	False

```
[21] x = ist_df['수명4'][ist_df['회사'] == 'A타이어']
     y = ist_df['수명4'][ist_df['회사'] == 'B타이어']
```

```
[22] # paired = True : paired sample t-test
     # correction = True : 이분산일때
     pg.ttest(x, y,
              paired = False,
              alternative = "two-sided",
              correction = True).round(3) # 이분산일때
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-4.082	52.522	two-sided	0.0	[-4.44, -1.51]	0.972	181.419	0.973

```
[23] # correction = False : 등분산일때와 비교
     pg.ttest(x, y,
              paired = False,
              alternative = "two-sided",
              correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.941	64	two-sided	0.0	[-4.48, -1.47]	0.972	119.98	0.973

```
[24] # 그래프
```

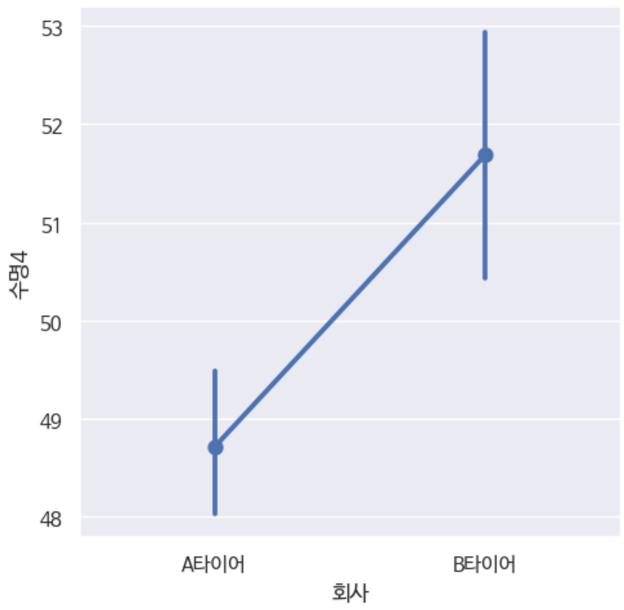
✓ 0초 오전 9:36에 완료됨

5.등분산 검정

```
[23]          correction = False).round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-3.941	64	two-sided	0.0	[-4.48, -1.47]	0.972	119.98	0.973

```
[24] # 그래프
     sns.catplot(x = "회사",
                 y = "수명4",
                 kind = "point",
                 data = ist_df)
     plt.show()
```



6.정규성 검정

✓ 0초 오전 9:36에 완료됨

6.정규성 검정

6.1 정규분포 검정

[25]
```
pg.normality(ist_df,
             dv = '수명1',
             group = '회사')
```

회사	W	pval	normal
A타이어	0.976345	0.705439	True
B타이어	0.959988	0.228475	True

[26]
```
num_feature = ['수명1', '수명2', '수명3', '수명4', '수명5']
for num in num_feature:
    print("----", num, "----")
    results = pg.normality(ist_df,
                           dv = num,
                           group='회사')
    print(results, "₩n")
```

```
---- 수명1 ----
              W      pval  normal
회사
A타이어  0.976345  0.705439    True
B타이어  0.959988  0.228475    True

---- 수명2 ----
              W      pval  normal
회사
A타이어  0.976345  0.705439    True
B타이어  0.959988  0.228475    True

---- 수명3 ----
              W      pval  normal
회사
```

✓ 0초 오전 9:38에 완료됨

6.정규성 검정

[25]

A타이어	0.976345	0.705439	True
B타이어	0.959988	0.228475	True

[26]

```
num_feature = ['수명1', '수명2', '수명3', '수명4', '수명5']
for num in num_feature:
    print("----", num, "----")
    results = pg.normality(ist_df,
                           dv = num,
                           group='회사')
    print(results, "₩n")
```

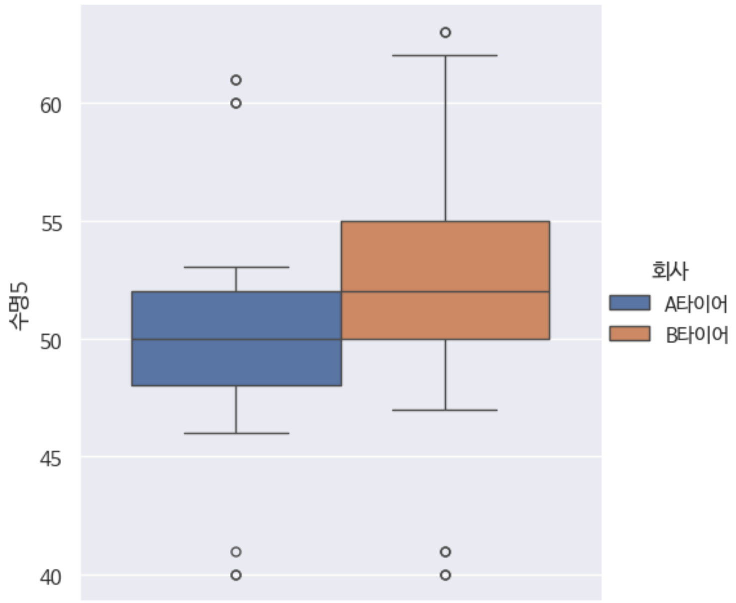
```
---- 수명1 ----
               W      pval  normal
회사
A타이어  0.976345  0.705439    True
B타이어  0.959988  0.228475    True

---- 수명2 ----
               W      pval  normal
회사
A타이어  0.976345  0.705439    True
B타이어  0.959988  0.228475    True

---- 수명3 ----
               W      pval  normal
회사
A타이어  0.977802  0.749155    True
B타이어  0.959988  0.228475    True

---- 수명4 ----
               W      pval  normal
회사
A타이어  0.930326  0.044749   False
B타이어  0.959988  0.228475    True

---- 수명5 ----
               W      pval  normal
회사
A타이어  0.917587  0.020380   False
B타이어  0.914812  0.010135   False
```

6.2 이상치제거(필요시)

✓ 0초 오전 9:38에 완료됨

6.2 이상치제거(필요시)

[27]
```
# 한글 폰트 인식
sns.catplot(data = ist_df,
            y = "수명5",
            hue = "회사",
            kind = "box")
plt.show()
```



6.3 비모수일때

[28]
```
pg.normality(ist_df,
```

✓ 0초 오전 9:38에 완료됨

6.정규성 검정

6.3 비모수일때

[28]
```
pg.normality(ist_df,
             dv = '수명5',
             group = '회사')
```

회사	W	pval	normal
A타이어	0.917587	0.020380	False
B타이어	0.914812	0.010135	False

[29]
```
x = ist_df['수명5'][ist_df['회사'] == 'A타이어']
y = ist_df['수명5'][ist_df['회사'] == 'B타이어']
```

[30]
```
# Mann-Whitney U test
pg.mwu(x, y,
       alternative='two-sided')
```

	U-val	alternative	p-val	RBC	CLES
MWU	434.5	two-sided	0.164555	0.199078	0.400461

[31]
```
# 모수통계(t-test)와 비교
pg.ttest(x, y,
         paired = False,
         alternative = "two-sided",
         correction = True).round(3)
```

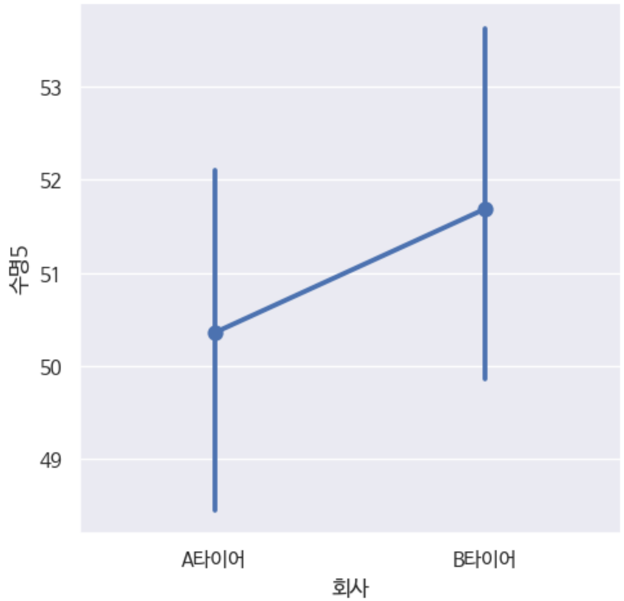
	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-0.98	63.948	two-sided	0.331	[-4.04, 1.38]	0.24	0.38	0.16

[32]
```
# 그래프
```

✓ 0초 오전 9:38에 완료됨

T-test	-0.98	63.948	two-sided	0.331	[-4.04, 1.38]	0.24	0.38	0.16

[32]
```
# 그래프
sns.catplot(x = "회사",
            y = "수명5",
            kind = "point",
            data = ist_df)
plt.show()
```



7.검증결과 그래프

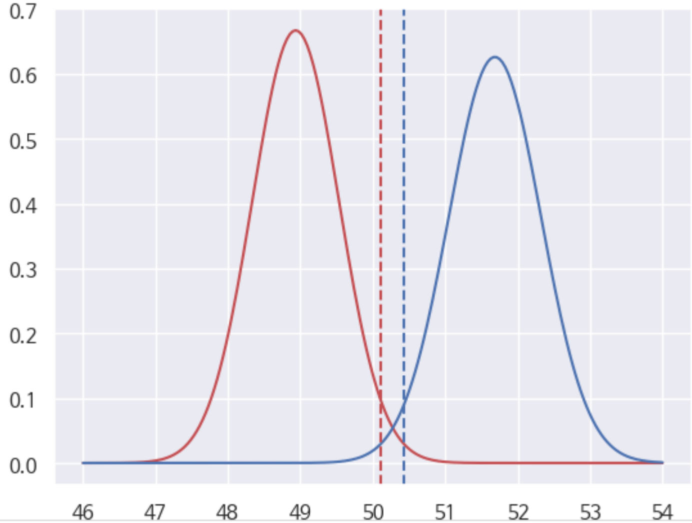
[33]
```
from scipy.stats import norm # 정규분포
```

✓ 0초 오전 9:38에 완료됨

7.검증결과 그래프

7.검증결과 그래프

[33]
```
from scipy.stats import norm # 정규분포

x_data = np.linspace(46, 54, 200)

x1 = 48.935
x2 = 51.686
se1 = 3.33/np.sqrt(31) # 표준오차(표준편차/sqrt(n)))
se2 = 3.77/np.sqrt(35) # 표준오차(표준편차/sqrt(n)))

plt.plot(x_data, norm.pdf(x_data, loc = x1, scale = se1), 'r-')
plt.plot(x_data, norm.pdf(x_data, loc = x2, scale = se2), 'b-')
plt.axvline(x = x1+1.96 * se1, color='r', linestyle='--')
plt.axvline(x = x2-1.96 * se2, color='b', linestyle='--')
plt.show()
```



✓ 0초 오전 11:07에 완료됨

8.두모집단 비율검정(proportion)

8.두모집단 비율검정(proportion)

[34]
```
from statsmodels.stats.proportion import proportions_ztest

count = np.array([87, 671])       # x1, x2
nobs = np.array([2065, 27949])    # n1, n2

z, p = proportions_ztest(count = count,
                         nobs = nobs,
                         value = 0)
print('z : {}, p : {}'.format(z, p))
```

```
z : 5.065085626514842, p : 4.082168195162829 3e-07
```

[35]
```
# chi-square test로 분석한 결과
count = np.array([87, 2065-87])       # x1, x2
nobs = np.array([671, 27949-671])    # n1, n2

tab = [count, nobs]
result = sm.stats.Table(tab)
rslt = result.test_nominal_association()
print(rslt)
```

```
df          1
pvalue      4.0821681956959566e-07
statistic   25.655092403927 25
```

[36]
```
# z값과 비교
np.sqrt(rslt.statistic)
```

```
5.065085626514842
```

9.동등성(Equivalence test)

✓ 0초 오전 9:42에 완료됨

8.두모집단 비율검정(proportion)

8.두모집단 비율검정(proportion)

[34]
```
from statsmodels.stats.proportion import proportions_ztest

count = np.array([87, 671])      # x1, x2
nobs = np.array([2065, 27949])   # n1, n2

z, p = proportions_ztest(count = count,
                         nobs = nobs,
                         value = 0)
print('z : {}, p : {}'.format(z, p))
```

```
z : 5.065085626514842, p : 4.082168195162829 3e-07
```

[35]
```
# chi-square test로 분석한 결과
count = np.array([87, 2065-87])      # x1, x2
nobs = np.array([671, 27949-671])   # n1, n2

tab = [count, nobs]
result = sm.stats.Table(tab)
rslt = result.test_nominal_association()
print(rslt)
```

```
df          1
pvalue      4.082168195695956 6e-07
statistic   25.655092403927 25
```

[36]
```
# z값과 비교
np.sqrt(rslt.statistic)
```

```
5.065085626514842
```

9.동등성(Equivalence test)

✓ 0초 오전 9:42에 완료됨

Independent Sample t-test

❖ <표>에 의하면, A타이어회사의 타이어수명(M=48.94)과 B타이어회사의 타이어수명(M=51.69)간에는 통계적으로 유의한 차이가 있었으며, B타이어회사의 타이어수명이 더 높게 나타났다(t=-3.12, p=0.003).

	A타이어회사 (n=31)	B타이어회사 (n=35)	t	Sig
타이어수명	48.94	51.69	-3.12	0.003

Independent Sample t-test(이분산일때)

❖ 먼저, Levene's test를 한 결과, 분산이 동질하지 않은 것으로 나타났다(F= 6.98, p = 0.01). 따라서 이분산으로 가정하고 Welch's test를 진행하였다.

❖ <표>에 의하면, A타이어회사의 타이어수명(M=48.71)과 B타이어회사의 타이어수명(M=51.69)간에는 통계적으로 유의한 차이가 있었으며, B타이어회사의 타이어수명이 더 높게 나타났다(t=-3.12, p=0.003).

	A타이어회사 (n=31)	B타이어회사 (n=35)	t	Sig
타이어수명	48.71	51.69	-4.08	0.000

Independent Sample t-test(비모수일때)

❖ 먼저, Shapiro test를 한 결과, 정규분포가 아닌 것으로 나타나(w=0.93, p=0.000), 비모수통계분석인 Mann Whitney U test를 실시하였다.

❖ <표>에 의하면, A타이어회사의 타이어수명(Md=50.00)과 B타이어회사의 타이어수명(Md=52.00)간에는 통계적으로 유의한 차이가 없는 것으로 나타났다 (w=434.5, p=0.165).

	A타이어회사 (n=31)	B타이어회사 (n=35)	w	Sig
타이어수명	50.00	52.00	434.5	0.165

연습문제

연습문제1

❖ 문제의 정의

- K대학에서는 재학생(1)과 교원(2)을 대상으로 교육과정에 대한 현황 조사를 실시하였다.
- 1:재학생, 2:교원
- 1. 종합점수는 재학생과 교원이 차이가 있는 가?
- 2. 등분산 가정이 만족하는가? 만족하지 않다면 Welch's test를 사용하세요.
- 3. 정규분포가정을 만족하는가? 만족하지 않다면 비모수통계를 사용하세요.
- 05_2.Education.csv

	구분	종합점수
1	재학생	0.0
2	재학생	5.0
3	재학생	70.8
4	재학생	71.6
5	재학생	71.7
6	재학생	71.7
7	재학생	73.3
8	재학생	73.3
9	재학생	78.3
10	재학생	79.1
11	재학생	70.0
12	재학생	70.0
13	재학생	70.0
14	재학생	70.0
15	재학생	70.0
16	재학생	51.7
17	재학생	53.3
18	재학생	54.2
19	재학생	54.2
20	재학생	56.6
21	재학생	56.6
22	재학생	57.5
23	재학생	58.3
24	재학생	58.3
25	재학생	61.6

III. Paired Sample T-test

Paired Sample t-test

❖ 문제의 정의

- K제약회사의 신제품 개발부서에서는 3개월 안에 살이 빠지는 다이어트 약을 개발하였다. 회사 경영진에게 새롭게 개발한 다이어트약이 효과가 있는지를 보고하기 위하여 약의 효능을 검증하였다. 약을 먹기 전의 체중과 약을 먹은 후 3개월 후의 체중을 조사하였다.
- 과연 새로운 약은 다이어트에 효과가 있는가
- 06_1.PST.csv

❖ 가설1

- 귀무가설 (H_0): 다이어트약을 먹기 전과 후의 체중은 변화가 없다.

  $H_0: \mu_d = 0$

- 연구가설 (H_1): 다이어트약을 먹기 전과 후의 체중은 변화가 있다.

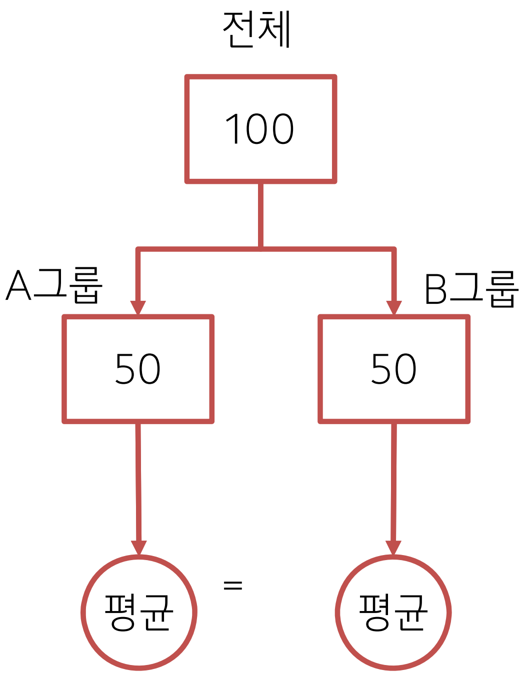
  $H_1: \mu_d \neq 0$

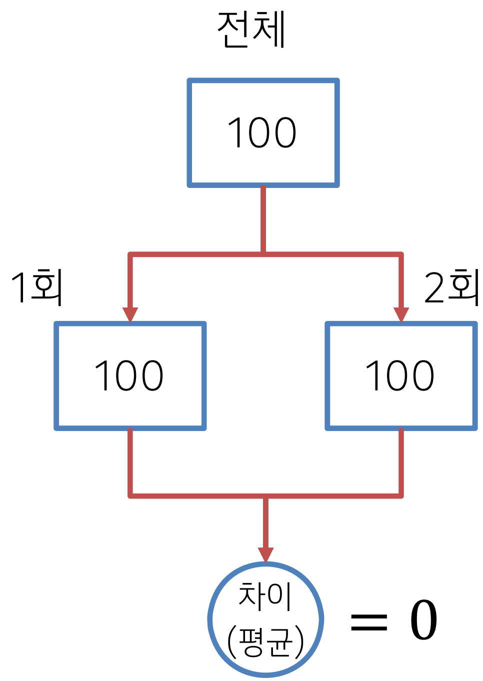
Paired Sample t-test

❖ 독립표본과 대응표본의 차이점

– 독립표본 : 대상에서 1번만 측정

– 대응표본 : 동일 대상에서 반복해서 측정





Paired Sample t-test

반복 1(x_{i1})	반복 2(x_{i2})	$\bar{d} = x_{i1} - x_{i2}$
x_{11} x_{21} $\vdots$ x_{n1}	x_{12} x_{22} $\vdots$ x_{n2}	$\bar{d} = x_{11} - x_{12}$ $\bar{d} = x_{21} - x_{22}$ $\vdots$ $\bar{d} = x_{n1} - x_{n2}$

반복 1(x_{i1})	반복 2(x_{i2})	$\bar{d} = x_{i1} - x_{i2}$
83.69 71.80 $\vdots$ 54.03	77.01 69.03 $\vdots$ 50.44	-6.68 -2.77 $\vdots$ -359

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}$$

$$s_d = \frac{\sum_{i=1}^{n} (d_i - \bar{d})^2}{n - 1}$$

Paired Sample t-test

❖ 통계치

- 표본 (n) : 50
- 표본평균 $(\bar{d})$: -2.79
- 표본표준편차 (s_d): 2.74 , 표준오차 $(\frac{s}{\sqrt{n}})$: 0.387

❖ 임계치

$$x_{critical} = \mu_d \pm 2.01\frac{s}{\sqrt{n}} = 0 \pm 2.01\frac{2.74}{\sqrt{50}} = 0 \pm 2.01(0.387) = 0 \pm 0.784$$

$$* \, t_{(19,\frac{0.05}{2})} = -2.01$$

❖ 검정통계량 (test statistics)

$$t_{cal} = \frac{\bar{d} - \mu_d}{\frac{s_d}{\sqrt{n}}} = \frac{-2.79 - 0}{\frac{2.74}{\sqrt{50}}} = \frac{-2.79}{0.39} = -7.19 < -2.01$$

$$* \, t_{cal} = \frac{\bar{d} - \mu_d}{\frac{s_d}{\sqrt{n}}} \sim t_{n-1}$$

❖ 유의확률(p-value)

$$p - value = P(|t| \geq 7.19) = 0.000 < 0.05$$

Paired Sample t-test

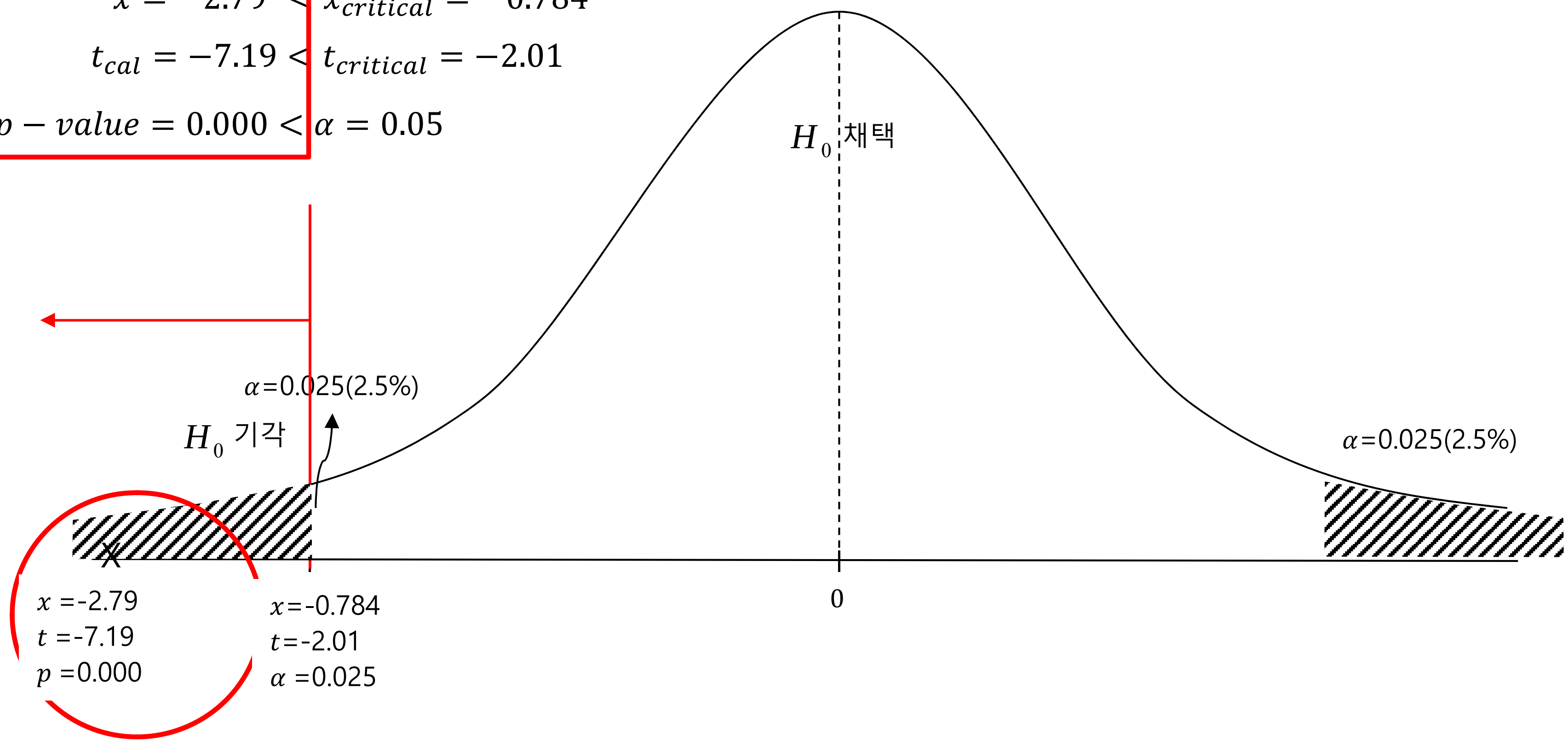
❖ 검정결과

<u>검정통계량</u> <u>임계치</u>

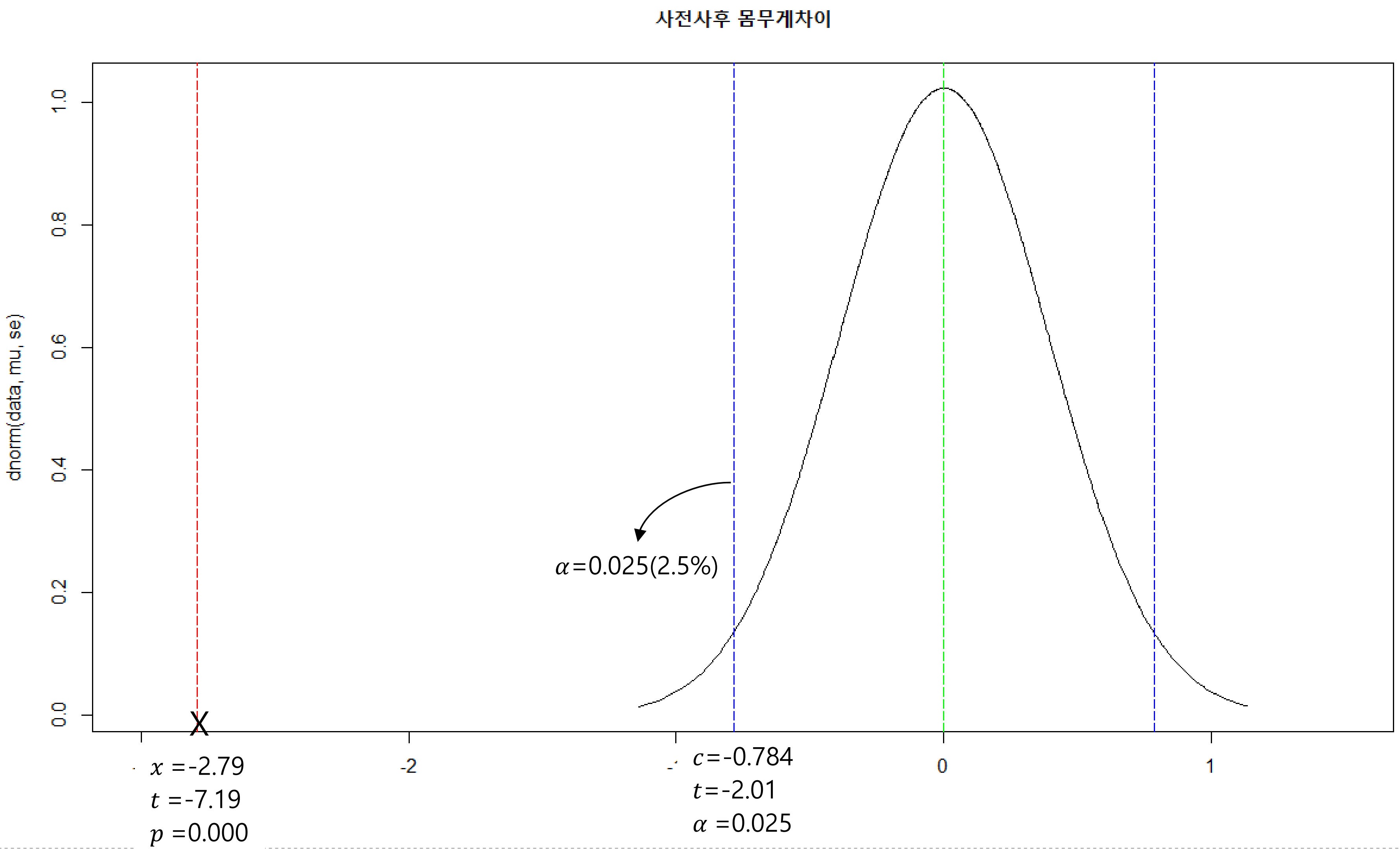
$$\bar{x} = -2.79 < x_{critical} = -0.784$$

$$t_{cal} = -7.19 < t_{critical} = -2.01$$

$$p - value = 0.000 < \alpha = 0.05$$



Paired Sample t-test



Paired Sample 분석절차

06_1.PairedSample t-test

06_1.PairedSample t-test

- https://pingouin-stats.org/build/html/generated/pingouin.ttest.html#pingouin.ttest

1.기본 package 설정

```
[ ] # 그래프에서 한글 폰트 인식하기
    !sudo apt-get install -y fonts-nanum
    !sudo fc-cache -fv
    !rm ~/.cache/matplotlib -rf

    #  *** 런타임 다시 시작
```

```
[ ] !pip install pingouin
```

```
[1] # 1.기본
    import numpy as np  # numpy 패키지 가져오기
    import matplotlib.pyplot as plt # 시각화 패키지 가져오기
    import seaborn as sns # 시각화

    # 2.데이터 가져오기
    import pandas as pd # csv -> dataframe으로 전환

    # 3.통계분석 package
    import pingouin as pg
    from scipy import stats
    import statsmodels.api as sm
```

```
[2] # 기본세팅
    # 테마 설정
    sns.set_theme(style = "darkgrid")
```

zEg#scrollTo...   ✓ 0초   오후 8:33에 완료됨

2.데이터 불러오기

2.데이터 불러오기

2.1 데이터 프레임으로 저장

- 원본데이터(csv)를 dataframe 형태로 가져오기(pandas)

[3]
```
pst_df = pd.read_csv('https://raw.githubusercontent.com/leecho-bigdata/statistics-python/main/06_1.PST.csv', encoding="cp949")
pst_df.head()
```

	사전	사후1	사후2	사후3
0	83.69	77.01	79.11	69.11
1	71.80	69.03	71.13	61.13
2	78.45	71.03	73.13	63.13
3	75.11	71.04	73.14	73.14
4	78.19	71.06	73.16	73.16

Next steps: View recommended plots

2.2 범주형 변수 처리

- 가변수 처리시 문자로 처리를 해야 변수명 구분이 쉬움

2.2 자료구조 살펴보기

[4]
```
pst_df.shape
```

(99, 4)

[5]
```
pst_df.info()
```

✓ 0초 오후 8:33에 완료됨

3.기술통계

3.기술통계

```
[7] # 그룹별 기술통계
pst_df.describe().round(2).T
```

	count	mean	std	min	25%	50%	75%	max
사전	50.0	73.04	7.00	54.03	71.72	75.27	76.12	83.69
사후1	50.0	70.25	6.91	50.12	69.05	71.20	74.31	77.33
사후2	50.0	72.35	6.91	52.22	71.15	73.30	76.41	79.43
사후3	50.0	71.75	7.14	52.22	70.32	73.24	76.25	79.43

4.통계분석

4.1 차이가 있는 경우(two-sided)

```
[8] # paired = True : paired sample t-test
pg.ttest(pst_df['사후1'], pst_df['사전'],
         paired = True,
         alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-7.193	49	two-sided	0.0	[-3.57, -2.01]	0.401	3.564e+06	0.794

```
[9] # one sample로 분석할 때와 비교
pst_df['차이1'] = pst_df['사후1'] - pst_df['사전']
pg.ttest(pst_df['차이1'], 0, alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-7.193	49	two-sided	0.0	[-3.57, -2.01]	1.017	3.564e+06	1.0

✓ 0초 오후 8:33에 완료됨

4.t-test

4.통계분석

4.1 차이가 있는 경우(two-sided)

[8]
```
# paired = True : paired sample t-test
pg.ttest(pst_df['사후1'], pst_df['사전'],
         paired = True,
         alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-7.193	49	two-sided	0.0	[-3.57, -2.01]	0.401	3.564e+06	0.794

[9]
```
# one sample로 분석할 때와 비교
pst_df['차이1'] = pst_df['사후1'] - pst_df['사전']
pg.ttest(pst_df['차이1'], 0, alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-7.193	49	two-sided	0.0	[-3.57, -2.01]	1.017	3.564e+06	1.0

4.2 양측과 단측 검정 비교(less)

[10]
```
# two-sided 차이가 없는 경우
pg.ttest(pst_df['사후2'], pst_df['사전'],
         paired = True,
         alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.779	49	two-sided	0.081	[-1.47, 0.09]	0.099	0.663	0.106

[11]
```
# one-side로 바뀌면 차이가 있음
```

✓ 0초 오후 8:33에 완료됨

4.t-test

4.2 양측과 단측 검정 비교(less)

[10]
```
# two-sided 차이가 없는 경우
pg.ttest(pst_df['사후2'], pst_df['사전'],
         paired = True,
         alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.779	49	two-sided	0.081	[-1.47, 0.09]	0.099	0.663	0.106

[11]
```
# one-side로 바뀌면 차이가 있음
pg.ttest(pst_df['사후2'], pst_df['사전'],
         paired = True,
         alternative = "less").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-1.779	49	less	0.041	[-inf, -0.04]	0.099	1.325	0.17

5.등분산 검정

- paired sample t-test는 등분산 검정 없음

6.정규성 검정

6.1 정규분포 검정

[12]
```
# 정규분포일때
pg.normality(pst_df)
```

W	pval	normal

✓ 0초 오후 8:33에 완료됨

Paired Sample t-test

❖ 결과해석

Descriptives

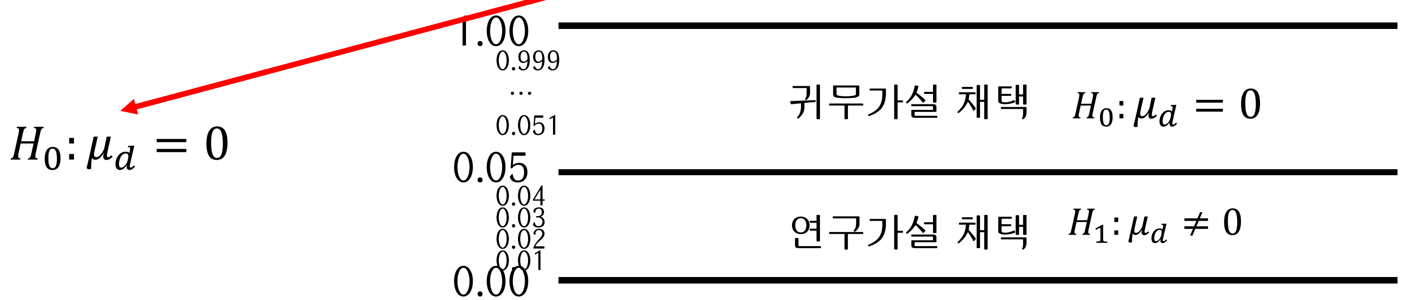
	N	Mean	Median	SD	SE
사후1	50	70.25	71.20	6.91	0.98
사전	50	73.04	75.27	7.00	0.99

$p - value$: 귀무가설$(H_0: \mu_1 = \mu_2)$이 맞을 확률

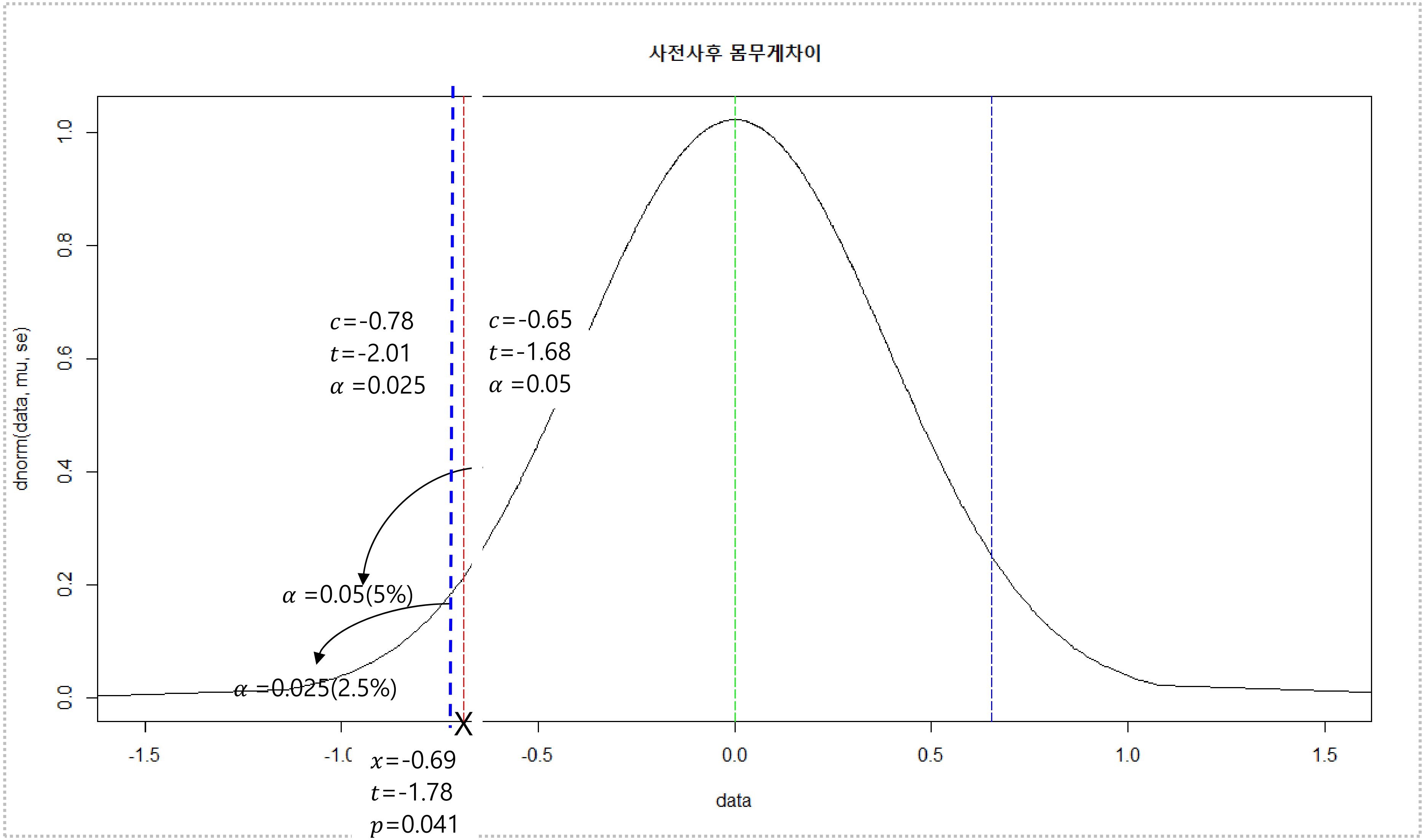
Paired Samples T-Test

			statistic	df	p	Mean difference	SE difference
사후1	사전	Student's t	−7.19	49.00	< .001	−2.79	0.39

Note. H_a $\mu_{\text{Measure 1 - Measure 2}} \neq 0$



Paired Sample t-test



$* p = 0.081$

6.정규성 검정

6.정규성 검정

6.1 정규분포 검정

[12]
```
# 정규분포일때
pg.normality(pst_df)
```

	W	pval	normal
사전	0.824520	3.381514e-06	False
사후1	0.766022	1.572257e-07	False
사후2	0.766022	1.572257e-07	False
사후3	0.810415	1.536419e-06	False
차이1	0.974911	3.620991e-01	True

[13]
```
# 정규분포일때
pg.normality(pst_df['차이1'])
```

	W	pval	normal
차이1	0.974911	0.362099	True

6.2 이상치제거(필요시)

6.3 비모수일때

[14]
```
pst_df['차이2'] = pst_df['사후2'] - pst_df['사전']
pst_df['차이3'] = pst_df['사후3'] - pst_df['사전']
```

[15]
```
pg.normality(pst_df)
```

0초 오전 9:48에 완료됨

6.정규성 검정

6.3 비모수일때

[14]
```
pst_df['차이2'] = pst_df['사후2'] - pst_df['사전']
pst_df['차이3'] = pst_df['사후3'] - pst_df['사전']
```

[15]
```
pg.normality(pst_df)
```

	W	pval	normal
사전	0.824520	3.381514e-06	False
사후1	0.766022	1.572257e-07	False
사후2	0.766022	1.572257e-07	False
사후3	0.810415	1.536419e-06	False
차이1	0.974911	3.620991e-01	True
차이2	0.974911	3.620991e-01	True
차이3	0.857319	2.456339e-05	False

[16]
```
# Wilcoxon Rank test
pg.wilcoxon(pst_df['사후3'], pst_df['사전'],
            alternative='two-sided').round(3)
```

	W-val	alternative	p-val	RBC	CLES
Wilcoxon	474.5	two-sided	0.116	-0.256	0.446

[17]
```
# 모수통계(t-test)와 비교
pg.ttest(pst_df['사후3'], pst_df['사전'],
         paired = True,
         alternative = "two-sided").round(3)
```

	T	dof	alternative	p-val	CI95%	cohen-d	BF10	power
T-test	-2.229	49	two-sided	0.03	[-2.45, -0.13]	0.182	1.467	0.244

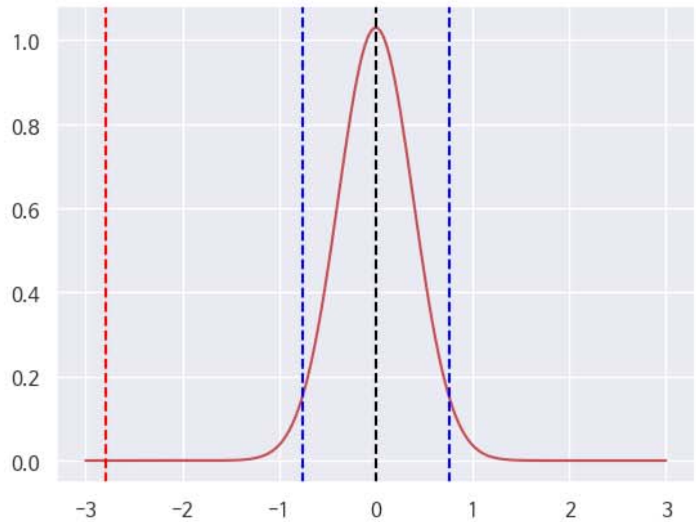
0초 오전 9:48에 완료됨

7.검증결과 그래프

7.검증결과 그래프

[18]

```
from scipy.stats import norm # 정규분포

x_data = np.linspace(-3, 3, 200)

mu = 0 # 평균
x = -2.79 # 표본평균
se = 2.74/np.sqrt(50) # 표준편차(표준오차)

plt.plot(x_data, norm.pdf(x_data, loc = mu, scale = se), 'r-')
plt.axvline(x = mu, color='black', linestyle='--')
plt.axvline(x = mu + 1.96 * se, color='blue', linestyle='--')
plt.axvline(x = mu - 1.96 * se, color='blue', linestyle='--')
plt.axvline(x = x, color='red', linestyle='--')
plt.show()
```



✓ 0초 오전 9:55에 완료됨

Paired Sample t-test

❖ 다이어트약의 효과를 검증한 결과 <표>에서 나타나듯이, 섭취전(M=73.15)과 섭취후(M=70.6)는 통계적으로 유의한 차이가 있는 것으로 나타났으며, 다이어트약을 섭취한 후에 몸무게가 감소한 것으로 나타났다(t=3.636, p=0.002).

	섭취전(n=20)	섭취후(n=20)	t	p
몸무게	73.15	70.6	3.636	0.002

연습문제

연습문제2

❖ 문제의 정의

- 다음은 호흡과 뇌파와의 관계를 연구한 자료이다.
- 총 4개 채널이 있는데, 채널별로 알파파(al)와 베타파(be) 간에는 각각 차이가 있는가?
- 1. Ch1al-Ch1be, Ch2al-Ch2be, Ch3al-Ch3be, Ch4al-Ch4be간에 차이가 있는 채널은 어디인가?
- 06_2.EEG.csv

	ch1al	ch2al	ch3al	ch4al	ch1be
1	0.03	0.03	0.02	0.02	0.18
2	0.05	0.04	0.07	0.07	0.09
3	0.05	0.02	0.06	0.06	0.13
4	0.01	0.01	0.02	0.02	0.08
5	0.04	0.04	0.05	0.06	0.18
6	0.02	0.02	0.02	0.02	0.17
7	0.04	0.05	0.05	0.03	0.18
8	0.03	0.03	0.02	0.03	0.08
9	0.03	0.05	0.08	0.08	0.10
10	0.04	0.02	0.07	0.07	0.08
11	0.01	0.04	0.07	0.05	0.13
12	0.06	0.06	0.06	0.06	0.15
13	0.04	0.05	0.05	0.04	0.10
14	0.03	0.03	0.04	0.04	0.16
15	0.03	0.02	0.04	0.03	0.13
16	0.04	0.03	0.04	0.04	0.12
17	0.05	0.04	0.04	0.05	0.12
18	0.03	0.04	0.05	0.05	0.10
19	0.08	0.06	0.09	0.10	0.08
20	0.03	0.02	0.06	0.06	0.15
21	0.05	0.05	0.08	0.07	0.11
22	0.02	0.02	0.03	0.02	0.11
23	0.03	0.03	0.04	0.03	0.12
24	0.05	0.04	0.04	0.04	0.13
25	0.02	0.02	0.02	0.01	0.05

Ⅳ. Equivalence test

동등성 검정

❖ 차이 검정(t-test) vs 동등성 검정

	차이 검정	동등성 검정
목적	• 차이가 있다	• 동등하다
한계유무	• 없음	• 사용자가 차이에 대해 허용 가능한 값 (*equivalence bound*: 동등한계) 있음
가설	• $H_0: \mu = 320$ • $H_1: \mu \neq 320$	• $H_{01}: \Delta \leq \Delta_L$, $H_{02}: \Delta \geq \Delta_u$ • $H_1: \Delta_L \leq \Delta \leq \Delta_U$
방법	• t-test	• two-one-sided t-tests (TOST)

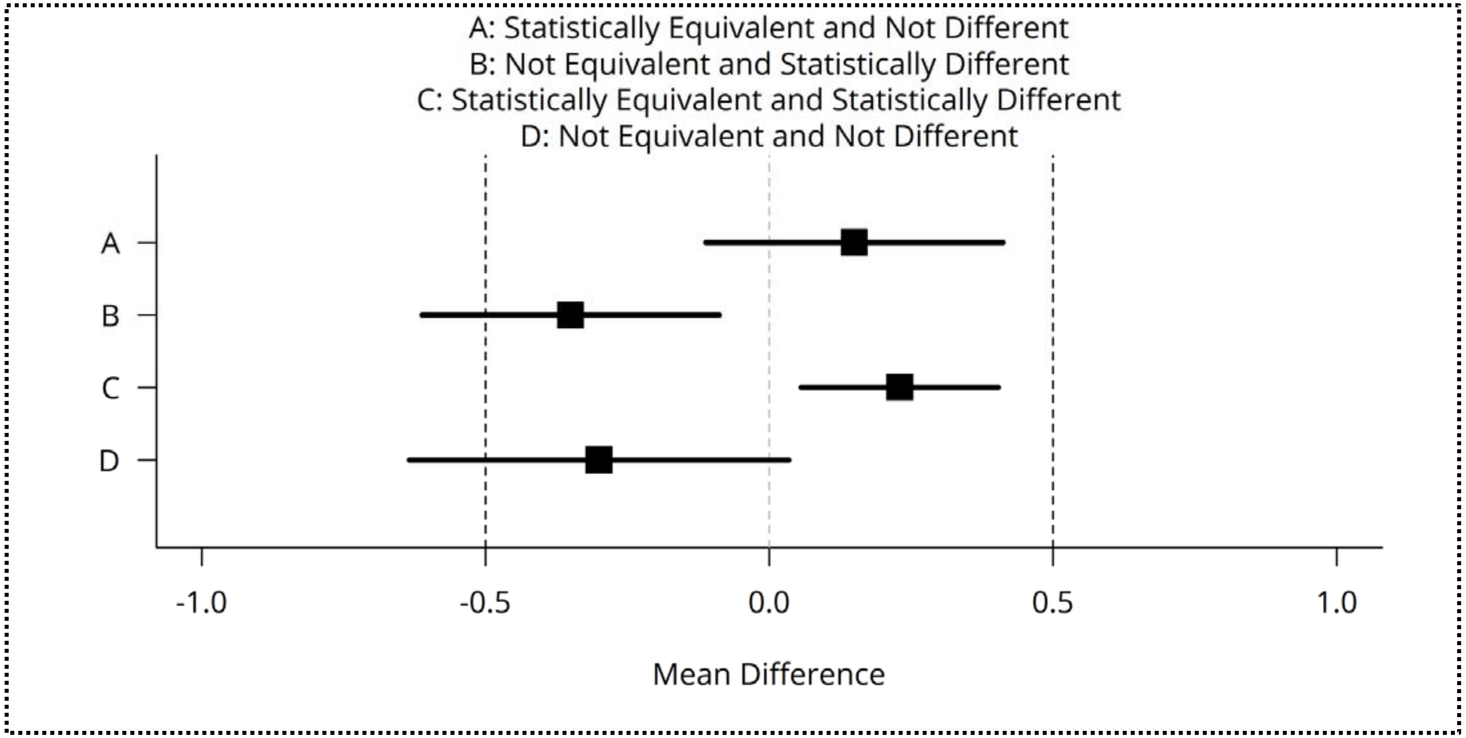
❖ 동등한계(*equivalence bound*)

– 기관별 기준이 있을 때: Annex, ICH, USP 등

– 기준이 없을 때: the smallest effect size: d = 0.3

동등성 검정

- ❖ A: difference(X), equivalence(O)
- ❖ B: difference(O), equivalence(X)
- ❖ C: difference(O), equivalence(O)
- ❖ D: difference(X), equivalence(X)



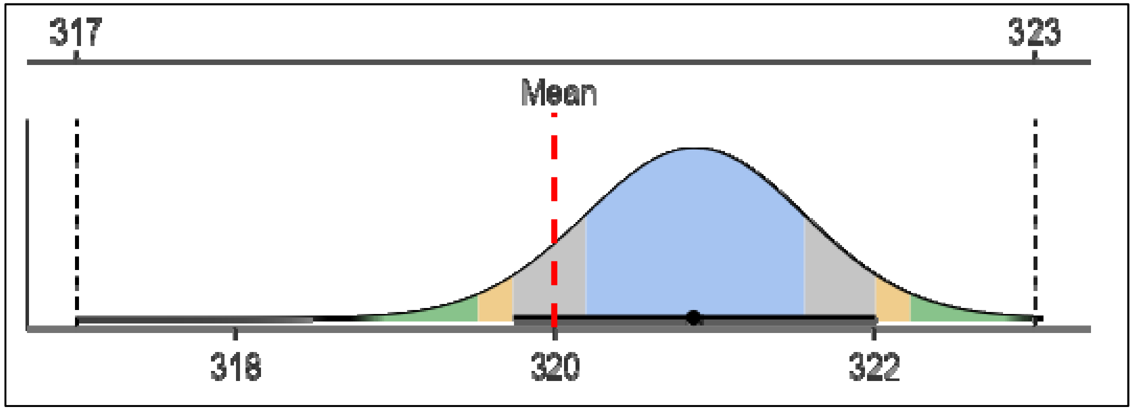
https://en.wikipedia.org/wiki/Equivalence_test

동등성 검정(One sample)

A: difference(X), equivalence(O)

TOST Results

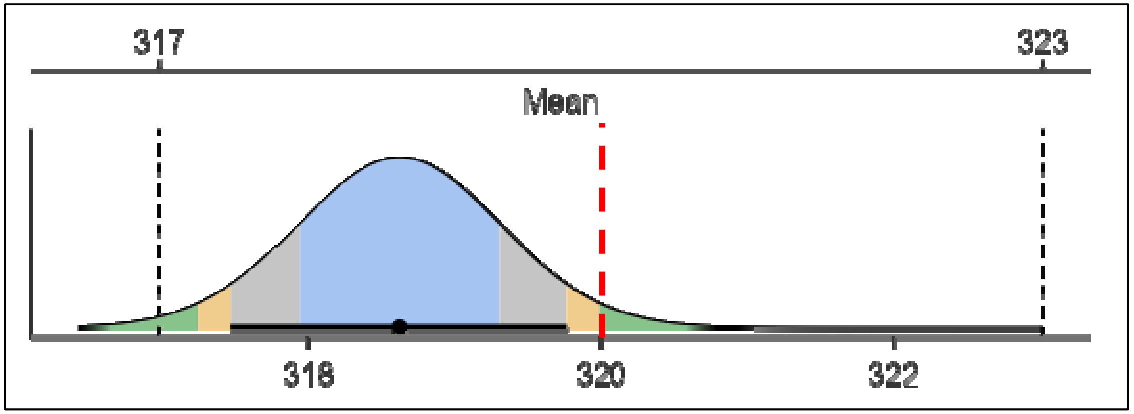
		t	df	p
무게4	t-test	469	99	< .001
	TOST Lower	5.67	99	< .001
	TOST Upper	−3.10	99	0.001



C: difference(O), equivalence(O)

TOST Results

		t	df	p
무게2	t-test	464	99	< .001
	TOST Lower	2.37	99	0.010
	TOST Upper	−6.36	99	< .001

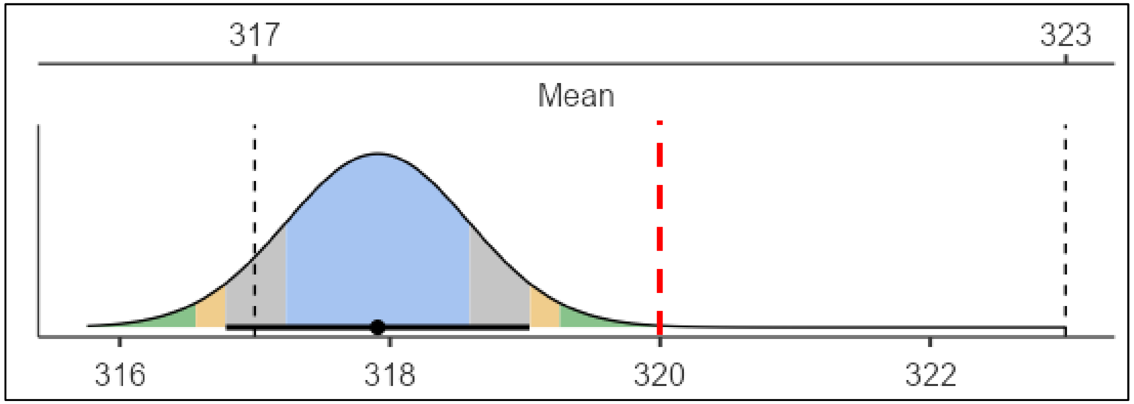


동등성 검정(One sample)

B: difference(O), equivalence(X)

TOST Results

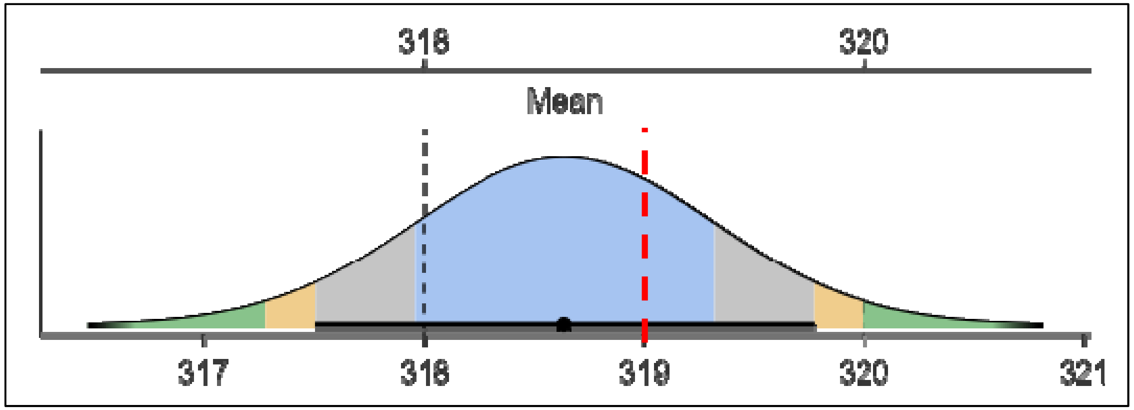
		t	df	p
무게1	t-test	469	99	< .001
	TOST Lower	1.34	99	0.091
	TOST Upper	−7.52	99	< .001



D: difference(X), equivalence(X)

TOST Results

		t	df	p
무게3	t-test	466	99	< .001
	TOST Lower	0.937	99	0.175
	TOST Upper	−1.99	99	0.025



동등성 검정(Independent sample)

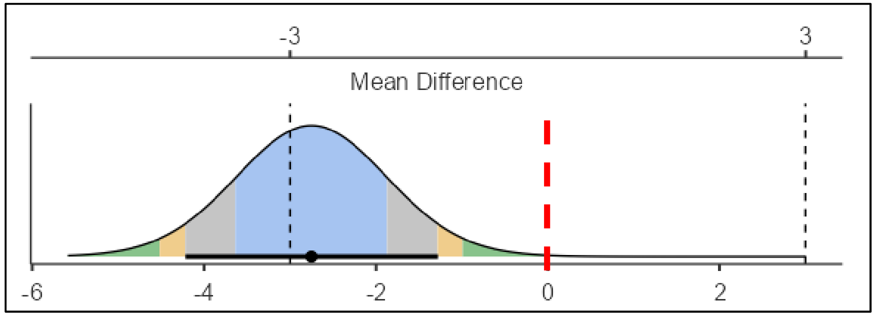
❖ Independent sample

- 동등한계($equivalence\ bound$): $\Delta: \mu_1 - \mu_2$
- $H_{01}: \mu_1 - \mu_2 \leq \Delta_L$
- $H_{02}: \mu_1 - \mu_2 \geq \Delta_u$
- $H_1: \Delta_L \leq \mu_1 - \mu_2 \leq \Delta_U$

B: difference(O), equivalence(X)

TOST Results

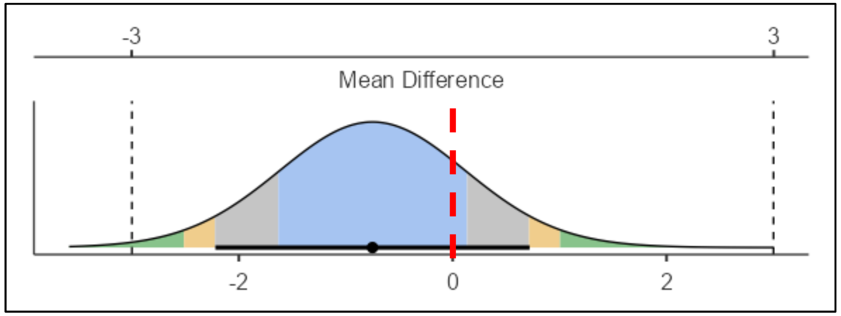
		t	df	p
수명1	t-test	−3.12	64.0	0.003
	TOST Upper	0.284	64.0	0.389
	TOST Lower	−6.53	64.0	< .001



C: difference(O), equivalence(O)

TOST Results

		t	df	p
수명2	t-test	−0.852	64.0	0.397
	TOST Upper	2.56	64.0	0.006
	TOST Lower	−4.26	64.0	< .001



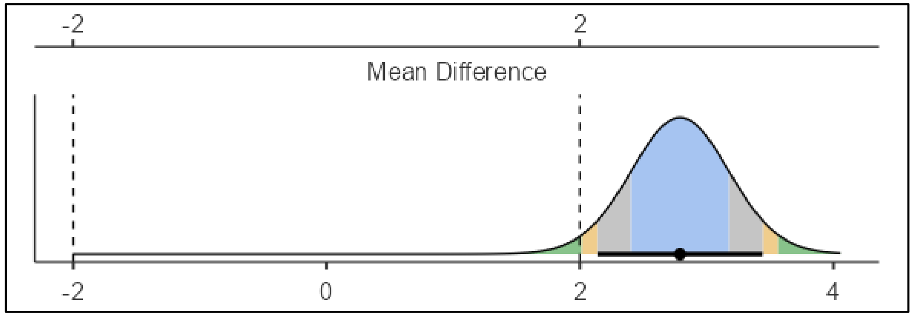
동등성 검정(Paired sample)

- Paired sample
  - 동등한계($equivalence\ bound$): $\Delta: \mu_d$
  - $H_{01}: \mu_d \leq \Delta_L$
  - $H_{02}: \mu_d \geq \Delta_u$
  - $H_1: \Delta_L \leq \mu_d \leq \Delta_U$

B: difference(O), equivalence(X)

TOST Results

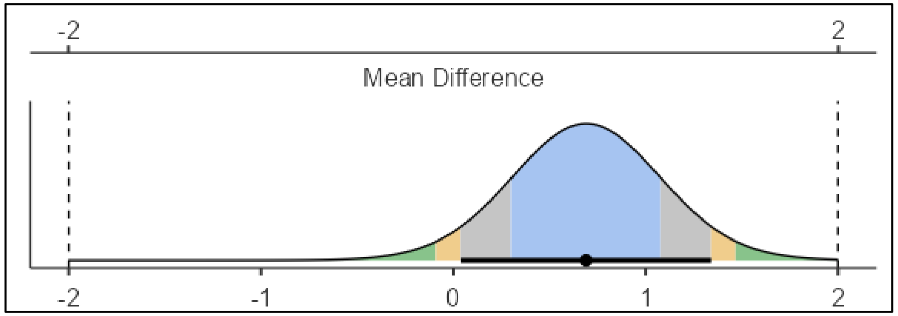
			t	df	p
사후1	사전	t-test	7.19	49	< .001
		TOST Lower	12.35	49	< .001
		TOST Upper	2.04	49	0.976



C: difference(O), equivalence(O)

TOST Results

			t	df	p
사후2	사전	t-test	1.78	49	0.081
		TOST Lower	6.93	49	< .001
		TOST Upper	−3.38	49	< .001



8.동등성(Equivalence test – One sample)

8.동등성(Equivalence test)

[31]
```
pg.tost(ost_df["무게1"],
        y = 320,
        bound = 3)
```

	bound	dof	pval
TOST	3	99	0.091325

[32]
```
pg.tost(ost_df["무게4"],
        y = 320,
        bound = 3)
```

	bound	dof	pval
TOST	3	99	0.00125

[33]
```
from scipy.stats import norm # 정규분포

x_data = np.linspace(315, 323, 200)

mu = 320   # 모집단 평균
eb = 3
x = 317.91 # 표본평균
se = 6.77/np.sqrt(100) # 표준오차(표준편차/sqrt(n)))

plt.plot(x_data, norm.pdf(x_data, loc = x, scale = se), 'r-')
plt.axvline(x = mu, color='black', linestyle='--')
plt.axvline(x = x - 1.96 * se, color='red', linestyle='--')
plt.axvline(x = x + 1.96 * se, color='red', linestyle='--')
plt.axvline(x = mu - 3, color='blue', linestyle='--')
plt.axvline(x = mu + 3, color='blue', linestyle='--')
plt.show()
```

✓ 0초 오전 9:26에 완료됨

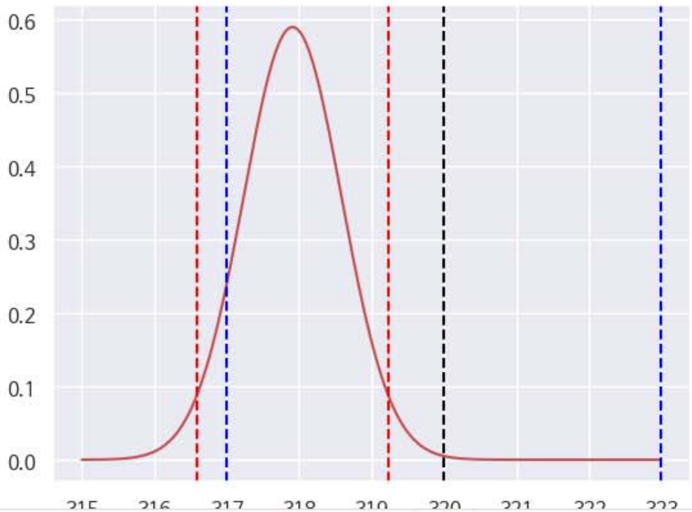
8.동등성(Equivalence test – One sample)

[32]

TOST	3	99	0.00125

[33]

```
from scipy.stats import norm # 정규분포

x_data = np.linspace(315, 323, 200)

mu = 320   # 모집단 평균
eb = 3
x = 317.91 # 표본평균
se = 6.77/np.sqrt(100) # 표준오차(표준편차/sqrt(n)))

plt.plot(x_data, norm.pdf(x_data, loc = x, scale = se), 'r-')
plt.axvline(x = mu, color='black', linestyle='--')
plt.axvline(x = x - 1.96 * se, color='red', linestyle='--')
plt.axvline(x = x + 1.96 * se, color='red', linestyle='--')
plt.axvline(x = mu - 3, color='blue', linestyle='--')
plt.axvline(x = mu + 3, color='blue', linestyle='--')
plt.show()
```



✓ 0초 오전 9:26에 완료됨

9.동등성(Equivalence test - Independent sample)

9.동등성(Equivalence test)

[37]
```
x = ist_df['수명1'][ist_df['회사'] == 'A타이어']
y = ist_df['수명1'][ist_df['회사'] == 'B타이어']
```

[38]
```
pg.tost(x, y,
        bound = 2,
        paired = False)
```

	bound	dof	pval
TOST	2	64	0.801358

[39]
```
x = ist_df['수명2'][ist_df['회사'] == 'A타이어']
y = ist_df['수명2'][ist_df['회사'] == 'B타이어']
```

[40]
```
pg.tost(x, y,
        bound = 2,
        paired = False)
```

	bound	dof	pval
TOST	2	64	0.080285

cro... ✓ 0초 오전 9:42에 완료됨

8.동등성(Equivalence test - Paired sample)

8.동등성(Equivalence test)

[19]
```
pg.tost(x = pst_df['사후1'],
        y = pst_df['사전'],
        bound = 2,
        paired = True)
```

	bound	dof	pval
TOST	2	49	0.976447

[20]
```
pg.tost(x = pst_df['사후2'],
        y = pst_df['사전'],
        bound = 2,
        paired = True)
```

	bound	dof	pval
TOST	2	49	0.000721

cro... ✓ 0초 오전 9:55에 완료됨

V. Sample size

Sample size

Sample size

❖ 검정력을 이용한 표본크기

- 의학연구나 실험연구의 경우에 유의수준($\alpha = 0.05$)과 검정력(80%)을 이용해서 표본수 추출

- 유효효과: $$\delta = \mu_1 - \mu_0$$

- 효과크기(effect size: ES): $$ES = \frac{\mu_1 - \mu_0}{\sigma} \qquad ES = \frac{p_1 - p_2}{\sqrt{p(1-p)}}$$

- 표본크기:

단일표본일 때 $$n_1 = \frac{\sigma^2 (z_\alpha + z_\beta)^2}{\delta^2} = \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2$$

두 표본일 때 $$n_2 = 2 \times n_1$$

출처: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7745163/

Sample size(One sample t-test)

❖ One Sample t-test

- 사례) K병원에서는 새로운 진통제를 개발하고자 한다. 사전 연구를 통해 진통제의 효과크기가 5시간 이었으며, 표준편차는 20이었다. 5% 유의수준과 80% 검정력으로 평가하려면 몇 개의 표본을 이용해야 하는가?

- 효과크기(effect size: ES): $$ES = \frac{\mu_1 - \mu_0}{\sigma} = \frac{5}{20} = 0.25$$

- 표본크기: $$n = \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2 = \left(\frac{1.96 + 0.842}{0.25}\right)^2 = 127.52 \cong 128$$

- 실험군: 128명

- 감소율 감안(20%) $$n = \frac{128}{(1 - 0.2)} = 160$$

Sample size(Independent sample t-test)

- Independent Sample t-test
  - 사례) K병원에서는 새로운 진통제를 개발하고자 한다. 기존약을 투약할 대조군과 신약을 투약할 실험군으로 나누어서 실험을 하려고 한다. 기존 약에 비해 신약의 효과크기가 5시간 더 지속되어야 하며, 표준편차는 20이었다. 5% 유의수준과 80% 검정력으로 평가하려면 몇 개의 표본을 이용해야 하는가?
  - 효과크기(effect size: ES): $$ES = \frac{\mu_1 - \mu_0}{\sigma} = \frac{5}{20} = 0.25$$
  - 표본크기: $$n = 2 \times \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2 = 2 \times \left(\frac{1.96 + 0.842}{0.25}\right)^2 = 2 \times 126 \cong 252$$
  - 대조군: 252명, 실험군: 252명
  - 감소율 감안(20%) $$n = \frac{252}{(1 - 0.2)} = 316$$

Sample size(Paired sample t-test)

❖ Paired sample t-test

– 사례) 만성 편두통 환자의 통증 감소를 위한 침술 치료의 효능을 평가하고자 한다. 사전에 통증을 측정하고, 침을 맞은 이후에 통증을 평가하고 한다. 침을 맞은 이후에 기존에 비해 통증이 10 만큼 줄어 들었는지 확인하고자 하며, 표준편차는 20이었다. 5% 유의수준과 80% 검정력으로 평가하려면 몇 개의 표본을 이용해야 하는가?

– 효과크기(effect size: ES): $ES = \frac{\mu_d}{\sigma_d} = \frac{10}{20} = 0.5$

– 표본크기: $n = \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2 = \left(\frac{1.96 + 0.842}{0.5}\right)^2 = 33.37 \cong 34$

– 34명

– 감소율 감안(20%) $n = \frac{34}{(1 - 0.2)} = 42$

Sample size(One proportion test)

❖ One proportions test

– 사례) 당뇨병환자에서 아스피린 복용한 사람과 그렇지 않은 사람간에 뇌경색(Cerebral infarct) 발생비율이 차이가 있었는가? 아스피린은 복용하지 않은 사람이 뇌경색이 일어날 가능성은 2.4%였다. 아스피린을 복용하면 2%이상 상승하는 것을 확인하고자 한다. 5% 유의수준과 80% 검정력으로 평가하려면 몇 개의 표본을 이용해야 하는가?

$$H_0:\ \theta_0 = 0.024 \qquad H_1:\ \theta_1 = \theta_0 + \delta = 0.024 + 0.02 = 0.044$$

– 효과크기(effect size: ES):

$$ES = \frac{p_1 + p_2}{\sqrt{p(1-p)}} = \frac{0.044 - 0.024}{\sqrt{0.034(1-0.034)}} = \frac{0.02}{0.181} = 0.11$$

$$*\ p = \frac{p_1 + p_2}{2} = \frac{0.044 + 0.024}{2} = 0.034$$

– 표본크기:

$$n = \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2 = \left(\frac{1.96 + 0.842}{0.11}\right)^2 = 630.52 \cong 631$$

– 631명

– 감소율 감안(20%) $n = \dfrac{631}{(1-0.2)} = 789$

Sample size(Two proportion test)

❖ Two Proportion test

- 사례) 당뇨병환자에서 아스피린 복용한 사람과 그렇지 않은 사람간에 뇌경색(Cerebral infarct) 발생비율이 차이가 있었는가? 아스피린은 복용하지 않은 사람이 뇌경색이 일어날 가능성은 2.4%였다. 실험군과 대조군으로 나누어서 아스피린을 복용하면 2%이상 상승하는 것을 확인하고자 한다. 5% 유의수준과 80% 검정력으로 평가하려면 몇 개의 표본을 이용해야 하는가?

- 효과크기(effect size: ES): $$ES = \frac{p_1 + p_2}{\sqrt{p(1-p)}} = \frac{0.044 - 0.024}{\sqrt{0.034(1-0.034)}} = \frac{0.02}{0.181} = 0.11$$

- 표본크기: $$n = 2 \times \left(\frac{z_{\alpha/2} + z_\beta}{ES}\right)^2 = 2 \times \left(\frac{1.96 + 0.842}{0.11}\right)^2 = 2 \times 630 \cong 1262$$

- 감소율 감안(20%) $$n = \frac{1{,}262}{(1-0.2)} = 1{,}577$$

- 대조군: 1,577명, 실험군: 1,577명

06_03.sample size

06_03.sample size

1.기본 package 설정

[1]
```
# 1.기본
import numpy as np  # numpy 패키지 가져오기
import matplotlib.pyplot as plt # 시각화 패키지 가져오기
import seaborn as sns # 시각화

# 2.데이터 가져오기
import pandas as pd # csv -> dataframe으로 전환

# 3.통계분석 package
from scipy import stats
import statsmodels.api as sm
```

2.One Sample t-test

[2]
```
# 3.power package
import statsmodels.stats.power as smp
from statsmodels.stats.power import TTestIndPower
from statsmodels.stats.power import TTestPower
```

[3]
```
effect_size = 5/20
power = 0.8
alpha = 0.05

pa = smp.TTestPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
```

0초 오전 9:58에 완료됨

2.One Sample t-test

2.One Sample t-test

[2]
```
# 3.power package
import statsmodels.stats.power as smp
from statsmodels.stats.power import TTestIndPower
from statsmodels.stats.power import TTestPower
```

[3]
```
effect_size = 5/20
power = 0.8
alpha = 0.05

pa = smp.TTestPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

```
127.51583288422903
```

[4]
```
sz/(1-0.2)
```

```
159.3947911052863
```

3.Independent Sample t-test

[5]
```
effect_size = 5/20
power = 0.8
alpha = 0.05

pa = smp.TTestIndPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
```

0초 오전 9:58에 완료됨

3.Independent Sample t-test

LGE Internal Use Only

3.Independent Sample t-test

[5]
```
effect_size = 5/20
power = 0.8
alpha = 0.05

pa = smp.TTestIndPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

252.12750515434277

[6]
```
sz/(1-0.2)
```

315.15938144292846

4.Paired Sample t-test

[7]
```
effect_size = 10/20
power = 0.8
alpha = 0.05

pa = smp.TTestPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

33.3671314275208

[8]
```
sz/(1-0.2)
```

scrollTo... 0초 오전 9:58에 완료됨

4.Paired Sample t-test

4.Paired Sample t-test

```
[7] effect_size = 10/20
    power = 0.8
    alpha = 0.05

    pa = smp.TTestPower()
    sz = pa.solve_power(effect_size = effect_size,
                        alpha = alpha,
                        power = power)
    sz
```

33.3671314275208

```
[8] sz/(1-0.2)
```

41.708914284401004

5.One proportion test

```
[9] import statsmodels.stats.api as sms
    from statsmodels.stats.power import GofChisquarePower

    effect_size = sms.proportion_effectsize(0.044, 0.024)
    power = 0.8
    alpha = 0.05

    pa = sms.GofChisquarePower()
    sz = pa.solve_power(effect_size = effect_size,
                        alpha = alpha,
                        power = power)
    sz
```

0초 오전 9:58에 완료됨

5.One proportion test

5.One proportion test

[9]
```
import statsmodels.stats.api as sms
from statsmodels.stats.power import GofChisquarePower

effect_size = sms.proportion_effectsize(0.044, 0.024)
power = 0.8
alpha = 0.05

pa = sms.GofChisquarePower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

630.5273608951413

[10]
```
sz/(1-0.2)
```

788.1592011189266

6.Two proportion test

[11]
```
effect_size = sms.proportion_effectsize(0.044, 0.024)
power = 0.8
alpha = 0.05

pa = sms.NormalIndPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

1261.0547193616262

✓ 0초 오전 10:01에 완료됨

6.Two proportion test

6.Two proportion test

[11]
```
effect_size = sms.proportion_effectsize(0.044, 0.024)
power = 0.8
alpha = 0.05

pa = sms.NormalIndPower()
sz = pa.solve_power(effect_size = effect_size,
                    alpha = alpha,
                    power = power)
sz
```

```
1261.0547193616262
```

[12]
```
sz/(1-0.2)
```

```
1576.3183992020327
```

✓ 0초 오전 10:01에 완료됨