- 집간 간의 평균값 차이가 통계적으로 유의미한 것인지 알아내는 방법

> 예를 들어 쇼핑몰의 지역별 객단가를 분석했을 때 A지역의 고객별 평균 매출은 67,000원이고 B지역은 68,500원이라고 했을 때 1,500원 차이가 우연히 나타난 것인지, 통계적으로 유의미한 것인지 알아 보기 위해서는 t-test를 진행

> A의 집단과 B 집단에서 표본을 추출하고 몸무게의 평균 차이를 비교하였을 때, 2kg의 차이가 났다. 과연, 이 몸무게의 차이는 두 집단의 몸무게가 차이가 난다고 할 수 있을까?

- t-test는 검정통계량이 귀무가설 하에서 t-분포를 따르는 통계적 가설 검정 방법
- 어느 특정 집단의 평균의 값을 추정하거나 차이를 검정할 때 사용할 수 있음
- 종속변수는 평균값을 가질 수 있는 연속형 변수여야 하고, 독립변수는 성별, 종교, 부서와 같은 범주형 변수여야 한다.
- 표본의 크기가 30이상이면 중심 극한 정리에 의해 정규분포를 따른다고 볼 수 있으며 t-test 사용 가능
- 모집단이 정규분포를 따르지만 그의 평균과 표준편차를 모르고 또한 표본크기가 작은 경우에 모평균 μ 에 대한 신뢰구간의 설정은 t 통계량을 이용해야 한다.
- 표본크기 n이 작아도 적용 가능한 t분포에는, 정규분포에서 얻은 데이터라는 가정이 필요
- 단 표본크기 n이 클 때는 중심극한정리에 따라 모집단이 정규분포가 아니더라도 표본평균을 정규분포로 근사할 수 있으므로 신뢰구간은 정확해 짐

모평균 추정법

1단계

- 얻은 n개의 표본에서 표본평균 $\bar{x}$와 표본표준편차 s를 계산한다.

2단계

- 표본평균 $\bar{x}$와 표본표준편차 s, 추정하려고 하는 모평균 μ를 사용하여 자유도 n-1인 t분포를 따르는 통계량 T를 다음과 같이 계산

$$T = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

3단계

- 자유도 n-1인 95% 예언적중구간을 선택해서 $-\alpha \leq T \leq \alpha$라 하는 95% 예언적중구간을 만든다.

$$\text{신뢰구간} = \bar{x} \pm t_{\text{critical}} \times \frac{s}{\sqrt{n}}$$

4단계

- $-\alpha \leq T \leq \alpha$를 μ에 대해서 풀면, 이것이 95% 신뢰구간

예제

어떤 가게 주인이 예상 매출액을 세우려고 한다. 주인은 매출액을 정규모집단에서 관측된 데이터로 가정하고, 이 모평균 μ를 대표적인 매출액으로 추정하려고 한다. 전표 중에서 무작위로 8장을 골라보니 다음과 같은 수가 나왔다.
45, 39, 42, 57, 28, 33, 40, 52 (만원)
모평균 μ의 구간을 추정해 보세요.

```python
import numpy as np
import scipy.stats as stats

data = np.array([45, 39, 42, 57, 28, 33, 40, 52])

# 평균과 표준편차 계산
mean = np.mean(data) # 42
# 표본의 표준편차
std_dev = np.std(data, ddof=1)  # ddof=1로 설정하여 표본표준편차를 계산
# 9.441549509633317

# 표본크기
n = len(data)

# t 분포의 임계값 계산 (95% 신뢰구간)
t_critical = stats.t.ppf(1 - 0.05 / 2, df=n-1)  # df는 자유도 (표본크기 - 1)
margin_of_error = t_critical * (std_dev / np.sqrt(n))
# 7.8933329207118055

# 신뢰구간 계산
confidence_interval = (mean - margin_of_error, mean + margin_of_error)
# (34.106667079288194, 49.893332920711806)
```

휘발유의 옥탄가(정규분포로 가정함)를 13일 연속 조사하니 다음과 같았다.
88.6, 86.4, 87.2, 88.4, 87.2, 87.6, 86.8, 86.1, 87.4, 87.3, 86.4, 86.6, 87.1
옥탄가 모평균에 대한 95% 신뢰구간을 구하여라.

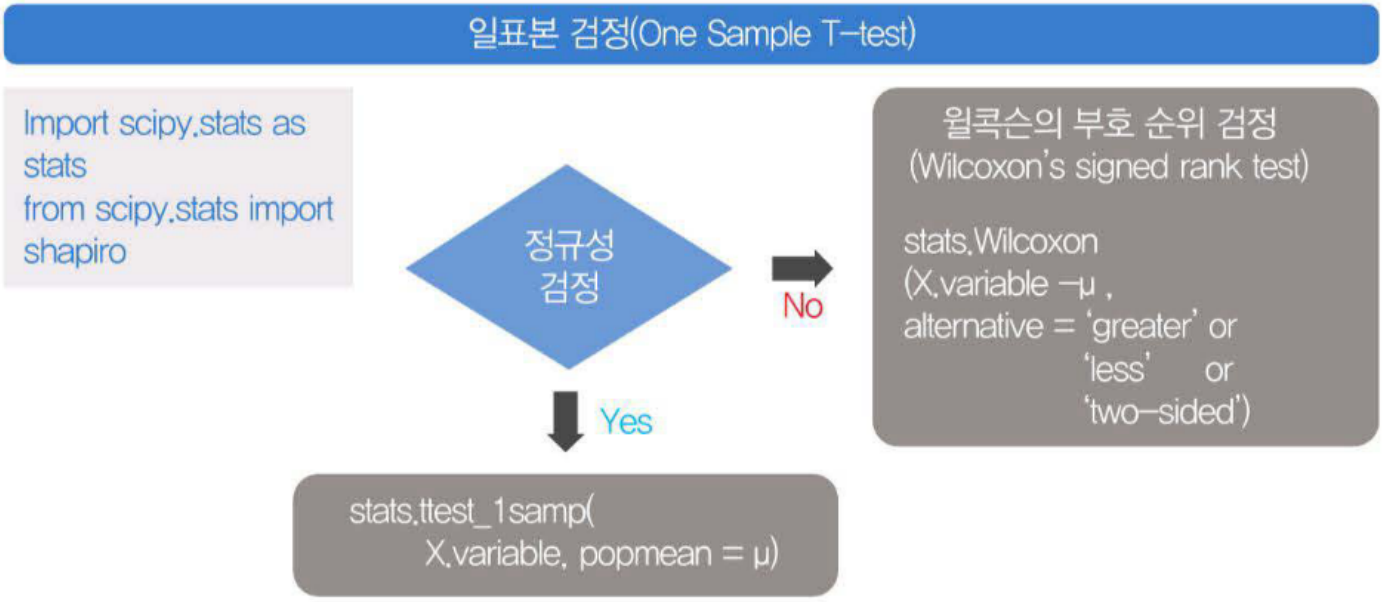
- 평균 : 87.16
- 표본 표준편차(s) = 0.74
- 분포의 임계값 : 2.179
- n : 13
- 신뢰구간 : 87.16 - 0.45 ~ 87.16 + 0.45
  - 86.71 ~ 87.61

t-test

- t-test는 데이터가 정규분포를 따른다는 가정한다.
- 두 집단 평균 차이 t-test의 귀무가설은 '집단 A와 B의 평균은 차이가 없다'
- 두 집단 간의 평균 차이를 표준오차로 나누어 검정 통계량 t를 구하는 것
- 귀무가설 : 평균이 동일
- 대립가설 : 평균이 차이가 있음

One-Sample t-test

- 일 표본 t-검정
- 한 모집단의 평균이 어떤 특정한 값과 같은지 검증하는 통계방법
- 일(단일) 표본 t-test은 가설검정의 일종으로, 하나의 모집단의 평균값을 특정값과 비교하는 경우 사용하는 통계적 분석 방법
- 단일모집단에서 관심이 있는 연속형 변수의 평균값을 특정 기준값과 비교한다.
- 가정 : 모집단의 구성요소들이 정규분포를 이룬다는 가정 - 종속변수는 연속형이어야 한다. - 검증하고자 하는 기준값이 있어야 한다.



단계

1. 가설수립

- 귀무가설 : 모평균의 값은 x 이다.
- 대립가설 : 모평균의 값은 x가 아니다.

2. 유의수준

- 기본적으로 0.05

3. 검정통계량 계산

- 검정통계량의 값 및 p-value(유의확률) 계산

귀무가설 판단

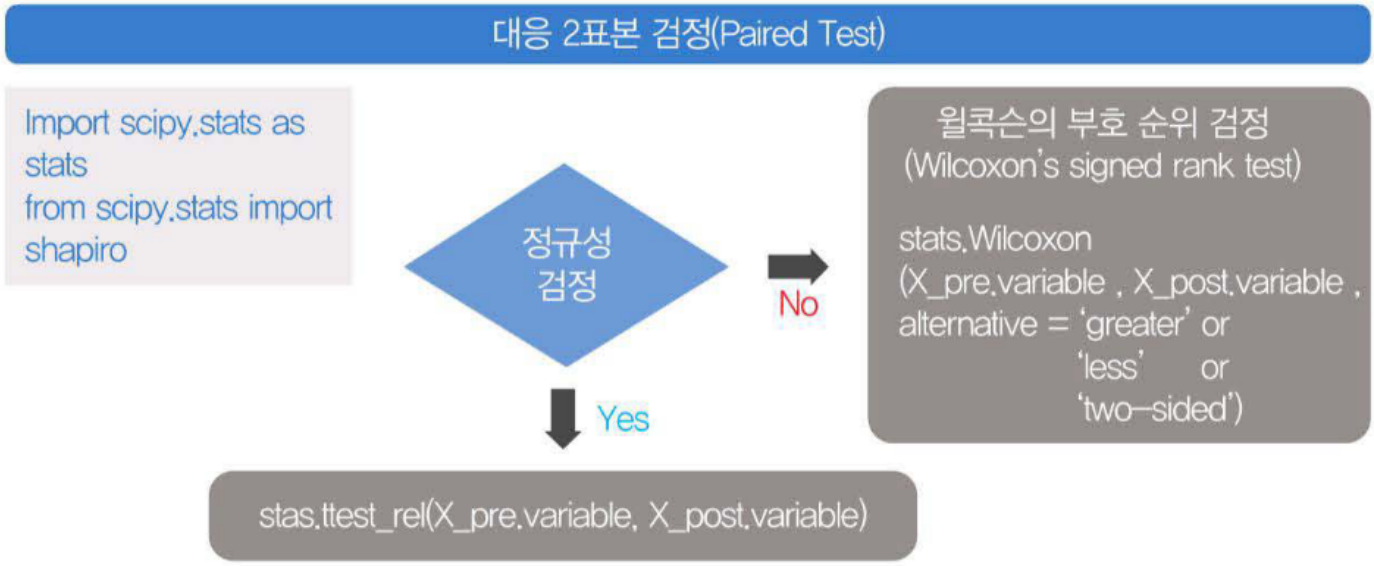
- 귀무가설의 기각여부 판단 및 해석

```
import pandas as pd
cats=pd.read_csv('../data/cats.csv')
#  cats 데이터에서 고양이들의 평균몸무게가 2.6kg인지 아닌지 통계적 검정을 수행하고, 결과를 해석해보자(양측검정, 유의수준 : 0.05).

# shapiro test를 통해 데이터의 정규성을 검정한다. 고양이의 몸무게를 검정하므로 고양이의 몸무게만 추출하여 shapiro test를 진행해야 한다.

import scipy.stats as stats
from scipy.stats import shapiro
mu =2.6
shapiro(cats['Bwt'])

# 정규성이 나왔다면
stats.ttest_1samp(cats.Bwt, popmean=mu)

# 위의 결과에서 정규성이 나오지 않았다면

stats.wilcoxon(cats.Bwt - mu , alternative='two-sided')

# 정규성 테스트에서 정규성이 아니라면 시각화로 확인
import matplotlib.pyplot as plt
cats_Bwt_cnt = pd.value_counts(cats['Bwt'].values, sort=False)
width =0.4
plt.bar(cats_Bwt_cnt.index, cats_Bwt_cnt.values,width)
plt.title('Bwt')
plt.ylabel('Count')
```

Paried t-test

- 대응 표본 t-검정
- 단일모집단에 대해 어떠한 처리르 가했을 때, 처리 전후에 따른 평균의 차이를 비교할 때 사용
- 즉, 동일한 대상에 대해 두 가지 관측치가 있는 경우 이를 비교하여 차이가 있는지 검정할 때 사용
- 주로 실험 전후의 효과를 비교하기 위해 사용
- 가장 흔한 예로는 한 집단을 대상으로 어떤 개입의 효과를 보기 위해 개입 전-후 값을 비교하여 개입의 효과를 측정하는 것
- 약의 복욕, 수술 혹은 치료, 새로운 교육방법 도입등이 있다.
- 표본 내에 개체들에 대해 두 번의 측정을 한다.(같은 집단이므로 등분산성 만족)

- 모집단의 관측값이 정규성을 만족해야 한다는 가정이 있다.



- 서로 연관성이 있는 두 대상으로부터 측정된 값이어서 비교하는 두 변수들 간에 상관관계가 존재
- 서로 독립적인 두 집단으로부터 측정한 두 값을 비교하는 two-sample t-test와 근본적으로 다른 점

1. 가설수립

- 귀무가설 : 두 모평균 사이의 차이는 없다.
- 대립가설 : 두 모평균 사이의 차이는 있다.

2. 유의수준

- 기본적으로 0.05

3. 검정통계량

- 검정통계량 및 p-value 계산

4. 귀무가설 판단

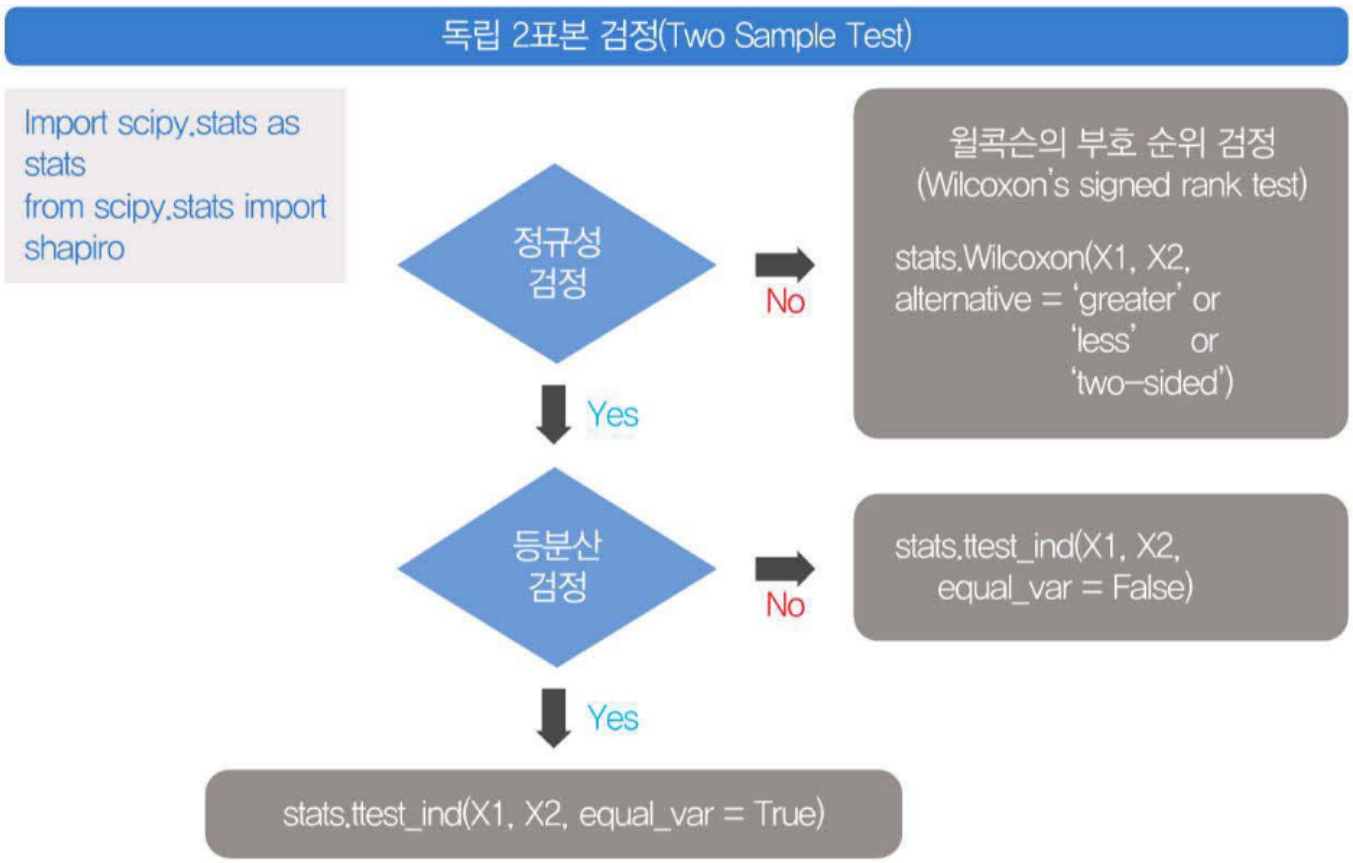
- 귀무가설의 기각여부 판단 및 해석

```
# 10명의 환자 대상 수면영양제 복용 전과 후의 수면시간을 측정하였다.
# 영양제의 효과가 있는지를 판단해보자.

# 귀무가설 : 수면제 복용 전과 후의 수면시간 차이는 없다.
# 대립가설 : 수면제 복용 전과 후의 수면시간 차이는 있다.
import pandas as pd
data = {'before':[7,3,4,5,2,1,6,6,5,4],
        'after':[8,4,5,6,2,3,6,8,6,5]}
data = pd.DataFrame(data)

stats.ttest_rel(data['after'],data['before'],alternative='greater')
```

```
# p-값이 0.05보다 작다면 귀무가설 기각, 즉 영양제 복용한 후 수면시간이 늘었다라고 결론
```

Two-sample t-test

- 이 표본 t-검정 (독립표본 t-test, Independent Sample t-test)
- 서로 독립적인 두 개의 집단에 대하여 모수(모평균)의 값이 같은 값을 갖는지 통계적으로 검정하는 방법
- 독립이란 두 모집단에서 각각 추출된 두 표본이 서로 관계가 없는 것을 의미
- 두 모집단의 분산이 같음을 의미하는 등분산성을 만족해야 한다.
- 따라서 t-test를 수행하기 전에 등분산 검정(F 검정)을 먼저 수행햐야 함
- 가장 흔한 실험 연구는 실험군과 대조군에 서로 다른 개입을 적용시킨 후 두 집단의 평균이 같은지를 비교하여 개입 효과의 차이를 평가하는 것
- 이런 경우에 two-sample t-test를 사용하는 데, 쌍을 이룬 두 변수간의 차이의 평균이 0인지 검정하는 Paried t-test와는 달리 서로 독립적인 두 집단의 평균의 차이가 0인지를 검정



1. 가설수립

- 귀무가설 : 두 모평균 사이의 차이는 없다.
- 대립가설 : 두 모평균 사이의 차이는 있다.

2. 유의수준

- 기본적으로 0.05

3. 정규성, 등분산성

- 정규성 및 등분산성 가설검정

4. 검정통계량

- 검정통계량 및 p-value 계산

5. 귀무가설 판단

- 귀무가설의 기각여부 판단 및 해석

```
# H0 : 수컷과 암컷 고양이의 몸무게 차이는 없다.
# H1 : 수컷과 암컷 고양이의 몸무게 차이는 있다.


import pandas as pd
cats=pd.read_csv('../data/cats.csv')


female = cats.loc[cats.Sex =='F', 'Bwt']
male = cats.loc[cats.Sex =='M', 'Bwt']

# 등분산성 검정(stats.levene)를 진행
stats.levene(female, male)

# 성별에 따른 몸무게가 등분산성을 만족하지 않으면 아래 함수의 파라미터의 값을
# equal_var=False로 진행

stats.ttest_ind(female, male, equal_var=False)

# p-값을 확인하여 귀무가설을 기각 및 선택을 한다.


# 두 집단간의 평균에 대한 시각화
female_Bwt_cnt = pd.value_counts(female.values, sort=False)
male_Bwt_cnt = pd.value_counts(male.values, sort=False)
fig, axs = plt.subplots(1, 2,figsize=(20,5))
fig.suptitle('Bar plot')
width =0.4
axs[0].bar(female_Bwt_cnt.index, female_Bwt_cnt.values)
axs[0].set_title('female_Bwt')
axs[0].set_ylabel('Count')
axs[1].bar(male_Bwt_cnt.index, male_Bwt_cnt.values)
axs[1].set_title('male_Bwt')
axs[1].set_ylabel('Count')
plt.show()
```

t-분포표

df \ α	0.4	0.25	0.1	0.05	0.025	0.01	0.005	0.0025	0.001	0.0005
1	0.325	1.000	3.078	6.314	12.706	31.821	63.657	127.32	318.31	636.62
2	0.289	0.816	1.886	2.920	4.303	6.965	9.925	14.089	22.327	31.599
3	0.277	0.765	1.638	2.353	3.182	4.541	5.841	7.453	10.215	12.924
4	0.271	0.741	1.533	2.132	2.776	3.747	4.604	5.598	7.173	8.610
5	0.267	0.727	1.476	2.015	2.571	3.365	4.032	4.773	5.893	6.869
6	0.265	0.718	1.440	1.943	2.447	3.143	3.707	4.317	5.208	5.959
7	0.263	0.711	1.415	1.895	2.365	2.998	3.499	4.029	4.785	5.408
8	0.262	0.706	1.397	1.860	2.306	2.896	3.355	3.833	4.501	5.041
9	0.261	0.703	1.383	1.833	2.262	2.821	3.250	3.690	4.297	4.781
10	0.260	0.700	1.372	1.812	2.228	2.764	3.169	3.581	4.144	4.587
11	0.260	0.697	1.363	1.796	2.201	2.718	3.106	3.497	4.025	4.437
12	0.259	0.695	1.356	1.782	2.179	2.681	3.055	3.428	3.930	4.318
13	0.259	0.694	1.350	1.771	2.160	2.650	3.012	3.372	3.852	4.221
14	0.258	0.692	1.345	1.761	2.145	2.624	2.977	3.326	3.787	4.140
15	0.258	0.691	1.341	1.753	2.131	2.602	2.947	3.286	3.733	4.073
16	0.258	0.690	1.337	1.746	2.120	2.583	2.921	3.252	3.686	4.015
17	0.257	0.689	1.333	1.740	2.110	2.567	2.898	3.222	3.646	3.965
18	0.257	0.688	1.330	1.734	2.101	2.552	2.878	3.197	3.610	3.922
19	0.257	0.688	1.328	1.729	2.093	2.539	2.861	3.174	3.579	3.883
20	0.257	0.687	1.325	1.725	2.086	2.528	2.845	3.153	3.552	3.850
21	0.257	0.686	1.323	1.721	2.080	2.518	2.831	3.135	3.527	3.819
22	0.256	0.686	1.321	1.717	2.074	2.508	2.819	3.119	3.505	3.792
23	0.256	0.685	1.319	1.714	2.069	2.500	2.807	3.104	3.485	3.768
24	0.256	0.685	1.318	1.711	2.064	2.492	2.797	3.091	3.467	3.745
25	0.256	0.684	1.316	1.708	2.060	2.485	2.787	3.078	3.450	3.725
26	0.256	0.684	1.315	1.706	2.056	2.479	2.779	3.067	3.435	3.707
27	0.256	0.684	1.314	1.703	2.052	2.473	2.771	3.057	3.421	3.690
28	0.256	0.683	1.313	1.701	2.048	2.467	2.763	3.047	3.408	3.674
29	0.256	0.683	1.311	1.699	2.045	2.462	2.756	3.038	3.396	3.659
30	0.256	0.683	1.310	1.697	2.042	2.457	2.750	3.030	3.385	3.646
40	0.255	0.681	1.303	1.684	2.021	2.423	2.704	2.971	3.307	3.551
60	0.254	0.679	1.296	1.671	2.000	2.390	2.660	2.915	3.232	3.460
120	0.254	0.677	1.289	1.658	1.980	2.358	2.617	2.860	3.160	3.373
∞	0.253	0.674	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291