T-검정 실습 예제

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
df=pd.read_csv("../data/(9)일표본t검정.csv")
df.head(10)
```

Out[1]:

	일련번호	과학성취도
0	1	482.7
1	2	490.8
2	3	494.6
3	4	481.2
4	5	476.9
5	6	490.6
6	7	490.9
7	8	479.0
8	9	484.2
9	10	476.3

In [2]:

```
df[df['과학성취도'].isna()] #결측치를 먼저 확인함
```

Out[2]:

	일련번호	과학성취도

In [3]:

```
x=np.array(df['과학성취도'])
```

In [4]:

```
df['과학성취도'].mean()
```

Out[4]:

486.0633

In [5]:

```
# t는 모집단과 표준편차 집단의 평균 차이
t,p = stats.ttest_1samp(x,470)
t,p
```

Out[5]:

(25.258263730196195, 2.980584542912143e-109)

In [6]:

```
# -109는 소수점 109번째 자리의 값을 나타냄
print(f'p값 = {p:.110f}')
```

p값 = 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000030

(결론)

- 1000명의 학생을 대상으로 과학성취도를 평가한 결과, 평균은 486.06점, 표준편차는 20.11이다. t통계값은 25.26이고, 이에 따른 유의확률은 0.05 미만이므로 대립가설이 채택된다
- 즉, 우리나라 학생들의 과학성취도의 평균은 470점이 아니라고 결론 내릴 수 있다.

결측지가 존재할때

In [7]:

```
df_na=pd.read_csv("../data/(9)일표본t검정_na.csv", encoding='cp949')
df_na.head(10)
```

Out[7]:

	일련번호	과학성취도
0	1	482.7
1	2	490.8
2	3	494.6
3	4	481.2
4	5	476.9
5	6	490.6
6	7	490.9
7	8	NaN
8	9	484.2
9	10	476.3

In [8]:

```
df_na[df_na['과학성취도'].isna()] #결측치를 먼저 확인함
```

Out[8]:

	일련번호	과학성취도
7	8	NaN
22	23	NaN
34	35	NaN

In [9]:

```
# 결측치를 0으로 대체하기 --> fillna()
df_filled= df_na.fillna(0)
df_filled.loc[7]#행의 값으로 접근
```

Out[9]:

```
일련번호      8.0
과학성취도     0.0
Name: 7, dtype: float64
```

In [10]:

```
# 결측치 값을 제거하기, dropna()
df_droped=df_na.dropna()
df_droped.head(10)
```

Out[10]:

	일련번호	과학성취도
0	1	482.7
1	2	490.8
2	3	494.6
3	4	481.2
4	5	476.9
5	6	490.6
6	7	490.9
8	9	484.2
9	10	476.3
10	11	484.5

과학성취도 점수를 표준화하여 분석하기(Z점수,T점수)

In [11]:

```
# 점수를 넘파이 배열로 만들기

scores = np.array(df['과학성취도'])
scores[:10]
```

Out[11]:

```
array([482.7, 490.8, 494.6, 481.2, 476.9, 490.6, 490.9, 479. , 484.2,
       476.3])
```

In [12]:

```
# 데이터에서 평균을 빼고 표준편차로 나누는 표준화 작업하여 z점수 구하기
# 표준화된 데이터는 평균이 0, 표준편차 1
z = (scores - np.mean(scores))/np.std(scores)
z[:10]
```

Out[12]:

```
array([-0.16732144,  0.23564697,  0.42469388, -0.24194522, -0.45586672,
        0.22569713,  0.24062189, -0.35139343, -0.09269766, -0.48571623])
```

In [13]:

```
# 평균이 50, 표준편차가 10이 되도록한 T점수

t=50+10*z
t[:10]
```

Out[13]:

```
array([48.32678563, 52.35646969, 54.24693876, 47.58054784, 45.44133284,
       52.25697132, 52.40621888, 46.48606575, 49.07302342, 45.14283772])
```

In [14]:

```
max(scores), max(z),max(t) # t점수는 평균이 50이라는 점이다
```

Out[14]:

```
(537.0, 2.534059361905425, 75.34059361905425)
```

In [15]:

```
# scipy 라이브러리에 zscore()함수를 이용하여 Z점수 구하기

stats.zscore(scores)[:10]
```

Out[15]:

```
array([-0.16732144,  0.23564697,  0.42469388, -0.24194522, -0.45586672,
        0.22569713,  0.24062189, -0.35139343, -0.09269766, -0.48571623])
```

In [16]:

```
df['Z점수']=z
df['T점수']=t
df
```

Out[16]:

	일련번호	과학성취도	Z점수	T점수
0	1	482.7	-0.167321	48.326786
1	2	490.8	0.235647	52.356470
2	3	494.6	0.424694	54.246939
3	4	481.2	-0.241945	47.580548
4	5	476.9	-0.455867	45.441333
...	...	...	...	...
995	996	486.3	0.011776	50.117756
996	997	418.6	-3.356244	16.437557
997	998	442.1	-2.187138	28.128616
998	999	494.1	0.399819	53.998193

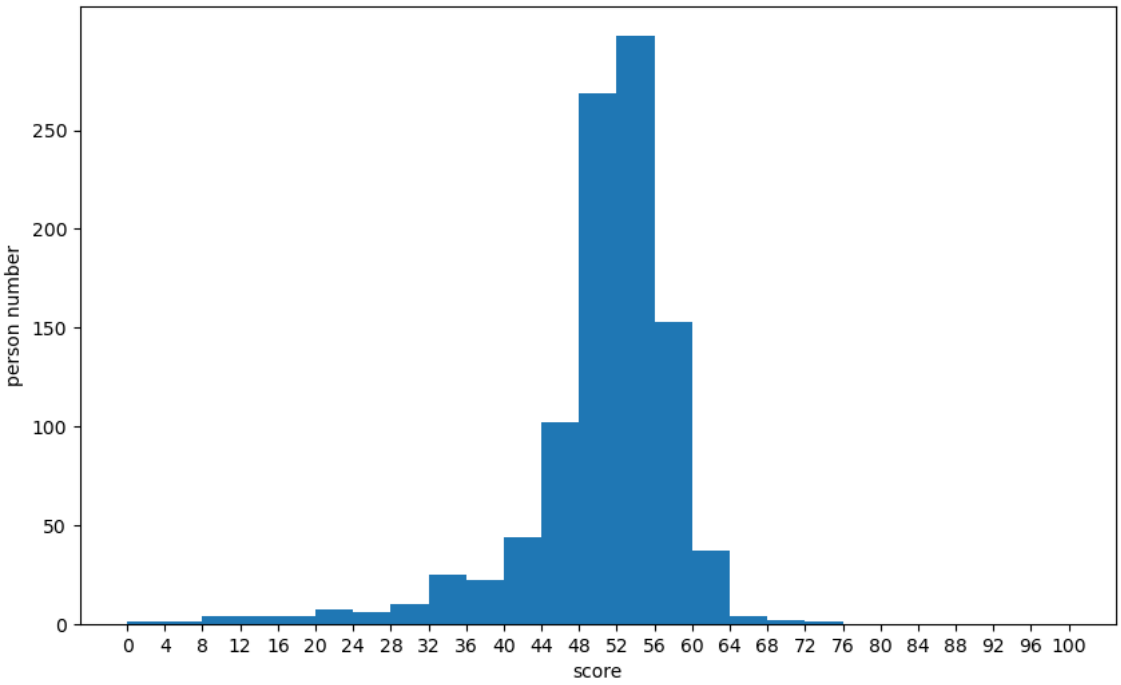
(결론)

- T점수가 50점이면 평균적인 결과이고, 60점이면 상위 결과라는 통일된 평가를 할 수 있다.
- T점수가 60점이라면 평균에서 표준편차가 한 단위 만큼 높은 점수에 해당한다.

In [17]:

```
fig = plt.figure(figsize=(10,6))
ax = fig.add_subplot(111)

freq, *_ = ax.hist(t, bins=25, range=(0,100))
ax.set_xlabel("score")
ax.set_ylabel("person number")
ax.set_xticks(np.linspace(0,100,26))
ax.set_yticks(np.arange(0,freq.max()+1,50))
plt.show()
```



In [18]:

```
df['과학성취도'].plot(kind='hist',bins=10,range=(0,540))
```

Out[18]:

<Axes: ylabel='Frequency'>

In [19]:

```
np.histogram(t, bins=25, range=(0,100))
```

Out[19]:

```
(array([  1,   1,   4,   4,   4,   7,   6,  10,  25,  22,  44, 102, 269,
        298, 153,  37,   4,   2,   1,   0,   0,   0,   0,   0,   0],
       dtype=int64),
 array([  0.,   4.,   8.,  12.,  16.,  20.,  24.,  28.,  32.,  36.,  40.,
         44.,  48.,  52.,  56.,  60.,  64.,  68.,  72.,  76.,  80.,  84.,
         88.,  92.,  96., 100.]))
```

In [20]:

```
np.histogram(scores,bins=10,range=(0,540))
```

Out[20]:

```
(array([  0,   0,   0,   0,   0,   1,   5,  20, 323, 651], dtype=int64),
 array([  0.,  54., 108., 162., 216., 270., 324., 378., 432., 486., 540.]))
```

2. 대응표본 t-검정 : 같은 모집단에서 추출된 표본들의 평균비교(전-후)

- 체력증진을 위해 개발된 새로운 프로그램을 고등학교 2학년 학생 150명을 대상으로 한학기동안 적용하였다. 체력증진 프로그램을 적용하기 전후 학생들의 체력에 차이가 있는가?
- 귀무가설(영가설) : 전후 차이가 없다. 즉, 프로그램의 효과가 없다.
- 대립가설 : 전후 차이가 있다. 프로그램의 효과가 있다.

In [21]:

```
df3=pd.read_csv("../data/(10)대응표본t검정.csv")
df3
```

Out[21]:

	일련번호	사전체력	사후체력
0	1	50.0	36.67
1	2	50.0	61.67
2	3	67.5	85.00
3	4	95.0	75.00
4	5	67.5	75.00
...	...	...	...
145	146	90.0	83.33
146	147	72.5	75.00
147	148	57.5	66.67
148	149	80.0	88.33
149	150	37.5	65.00

150 rows × 3 columns

In [22]:

```
# 기술통계
df3.describe()
```

Out[22]:

	일련번호	사전체력	사후체력
count	150.000000	150.000000	150.000000
mean	75.500000	64.066667	68.244867
std	43.445368	21.608060	16.007447
min	1.000000	10.000000	31.670000
25%	38.250000	48.125000	56.670000
50%	75.500000	67.500000	66.670000
75%	112.750000	82.500000	80.000000
max	150.000000	97.500000	96.670000

In [23]:

```
# 대응표본 t-검정 실행

t3,p3=stats.ttest_rel(df3['사후체력'],df3['사전체력'])
p3
```

Out[23]:

0.003766155401630092

(결론)

- 체력증진 프로그램의 효과를 알아보기 위하여 사전체력과 사후체력의 기술통계와 대응표본 t-검정 결과는 다음과 같다.
- 사전체력의 평균은 64.07, 표준편차는 21.61이며, 사후체력의 평균은 68.25, 표준편차는 16.01이다.
- 사전체력과 사후체력의 차이에 대한 통계적 유의성을 검정한 결과 t 통계값은 2.94, 유의확률(p값)은 0.004로서 유의수준 0.05보다 미만이다.
- 그러므로 체력증진 프로그램에 의한 학생들의 사전과 사후 체력에 차이가 있는 것으로 분석되었다. ---> 귀무가설 기각. 대립가설 채택.

독립표본 t-검정 : 다른 모집단에서 추출된 두 표본의 평균비교

- 성별에 따라 고등학교 2학년 학생들의 외국어 능력에 차이가 있는가?
- 귀무가설(영가설) : 성별에 따라 외국어 능력에 차이가 없다.
- 대립가설 : 차이가 있다.

In [24]:

```
df4=pd.read_csv("../data/(11)독립표본t검정.csv")
df4
```

Out[24]:

	일련번호	성별	외국어
0	1	2	22
1	2	1	37
2	3	2	51
3	4	1	52
4	5	2	55
...	...	...	...
223	224	1	45
224	225	2	40
225	226	1	46
226	227	2	53
227	228	1	39

228 rows × 3 columns

In [25]:

```
m_scores = df4.loc[df4['성별']==1,['외국어']]
m_scores
```

Out[25]:

	외국어
1	37
3	52
6	21
13	45
15	25
...	...
220	42
221	36
223	45
225	46
227	39

105 rows × 1 columns

In [26]:

```
f_scores = df4.loc[df4['성별']==2,['외국어']]
f_scores
```

Out[26]:

	외국어
0	22
2	51
4	55
5	37
7	45
...	...
217	41
218	50
222	36
224	40
226	53

123 rows × 1 columns

In [27]:

```
m_scores.describe()
```

Out[27]:

	외국어
count	105.000000
mean	39.714286
std	10.149568
min	19.000000
25%	31.000000
50%	39.000000
75%	48.000000
max	59.000000

In [28]:

```
f_scores.describe()
```

Out[28]:

	외국어
count	123.000000
mean	42.439024
std	9.115962
min	22.000000
25%	36.500000
50%	41.000000
75%	50.500000
max	59.000000

In [29]:

```
m=np.array(m_scores['외국어'])
f=np.array(f_scores['외국어'])

stats.levene(m,f,center='mean')
```

Out[29]:

```
LeveneResult(statistic=1.1634057041850445, pvalue=0.2819101981266899)
```

In [30]:

```
# 독립표본 t-검정 수행
# 기본적으로 equal_var=True로 설정되어 있음. 디폴트

t4,p4=stats.ttest_ind(m,f)
t4,p4
```

Out[30]:

```
(-2.1349509532860065, 0.033841284363902845)
```

In [31]:

```
# equal_var=False를 지정하면 웰치의 방법으로 독립표본 t-검정이 수행됨.
# 웰치의 t검정은 두 표본의 분산이 다르다는 것을 가정하고 검정을 수행한다.

t4, p4 = stats.ttest_ind(m, f, equal_var= False )
t4, p4
```

Out[31]:

```
(-2.1169131152610396, 0.035437530133271736)
```

(결론)

- Levene의 등분산 검정결과, 유의확율(p값)이 0.282로 유의수준(0.05) 이상이므로 두 집단의 분산이 같다는 귀무가설이 채택된다. 그러므로 독립표본 t-검정시 equal_var 인수에'True' 또는 생략하여 등분산을 가정하는 p값을 구한다.
- 고등학교 2학년 학생들의 성별에 따른 외국어 점수의 기술통계와 차이를 알아보기 위하여 두 독립표본 t 검정을 실시한 결과는 다음과 같다.
- 남학생들의 외국어 능력의 평균은 39.71, 표준편차는 10.15이며,
- 여학생들의 외국어 능력의 평균은 42.44, 표준편차는 9.12이다.
- 남여 학생들의 외국어 능력에 차이가 있는지에 대한 t통계값은 2.14, 유의확율(p값)dms 0.034로서 유의수준 0.05에서 성별에 따라 외국어 능력에 유의한 차이가 있는 것으로 분석되었다.