

## ADP\_Red\stats\1.ttest.py

```

1  # %% 0. Import Libraries
2  # !conda install numpy
3  # !conda install pandas
4  import numpy as np
5  import pandas as pd
6  import seaborn as sns
7  import matplotlib.pyplot as plt
8
9  import scipy.stats as stats
10
11 # %% 1. One-sample t-test
12 df = pd.read_csv('../ADP_Python/data/cats.csv')
13 dv = 'Bwt'
14
15 print(f'{df.info()}')
16 dv_mean = df[dv].mean()
17 pvalue = stats.ttest_1samp(df[dv], popmean=2.6)[1]
18
19 print(f'''
20 df 데이터는 총 {df.shape[0]}개의 데이터로 구성되어 있다.
21 {dv} 데이터 평균은 {dv_mean:.2f}이며
22 one-sample t-test 결과
23 p-value = {pvalue:.2f}으로 통계적으로 유의미하다.
24 ''')
25
26
27 # 1-2. Data Visualization
28 fig, axes = plt.subplots(nrows=2, ncols=1)
29
30 sns.histplot(df[dv], bins=15, ax=axes[0])
31 sns.barplot(data=df, y=dv, ax=axes[1])
32 plt.tight_layout
33 plt.show()
34
35
36 # 1-3. Statistical Test
37 mu = 2.6
38
39 # 정규성 검증: Shapiro-Wilks test
40 # Shapiro-Wilks test가 p-value < 0.05로 기각된다면
41 # 데이터가 정규분포를 따르지 않는다는 뜻이다.
42 val, pval = stats.shapiro(df[dv])
43
44 if pval > 0.05:
45     # 데이터가 정규분포를 따를 경우 평범하게 one sample t-test를 진행하면 된다.
46     print(f'p-value:{pval:.2f}, Normality Assumption Satisfied!')
47     print(f'Try Simple t-test!')
48     print(stats.ttest_1samp(df[dv], popmean=mu))
49 else:
50     # 데이터가 정규분포를 따르지 않을 경우 비모수 검정인 Wilcoxon test를 사용한다.
51     # Wilcoxon test가 p-value < 0.05로 기각된다면
52     print(f'p-value:{pval:.2f}, Normality Assumption Unsatisfied!')
53     print(f'Try Wilcoxon Test!')
54     print(stats.wilcoxon(df[dv]-mu, alternative='two-sided'))
55
56 # 어느 쪽이든 고양이의 몸무게는 2.6이 아니라는 결과가 나오지만
57 # one sample t-test가 아닌 wilcoxon test를 사용해야 한다.

```

```

58
59
60 # %% 2. paired-sample t-test
61 data = {'before':[7,3,4,5,2,1,6,6,5,4],
62         'after':[8,4,5,6,2,3,6,8,6,5]}
63
64 df = pd.DataFrame(data)
65 dv = 'diff'
66 df[dv] = df.after - df.before
67
68 # 2-2. Visualization
69 fig, axes = plt.subplots(nrows=2, ncols=1)
70
71 sns.histplot(df[dv], bins=15, ax=axes[0])
72 sns.barplot(data=df, y=dv, ax=axes[1])
73 plt.tight_layout
74 plt.show()
75
76 # 2-3. Statistical Test
77 val, pval = stats.shapiro(df[dv])
78
79 if pval < 0.05: # Normality assumption not satisfied
80     print(f'p-value:{pval:.2f}, Normality Assumption Unsatisfied!')
81     print(f'Try Wilcoxon Test!')
82 else: #
83     print(f'p-value:{pval:.2f}, Normality Assumption Satisfied!')
84     print(f'Try Simple t-test!')
85
86 # print(stats.wilcoxon(df.before, df.after, alternative='two-sided'))
87 print(stats.ttest_rel(df.after, df.before, alternative='greater'))
88
89 # %% 3. independent-sample t-test
90 # [Welch's t-test](https://en.wikipedia.org/wiki/Welch%27s_t-test)
91 # Independent sample t-test를 시행할 때
92 # 등분산성 확인을 위해 Levene test 등을 수행할 필요는 없다.
93 # 등분산성 가정을 만족하든, 만족하지 않든 Welch's t-test를 수행하면 된다.
94 # (오히려 등분산성을 미리 확인하고 Student's t-test와 Welch's t-test를 고르는 것을 경계해야 한다.)
95
96 # ★ 단, 정규성 조건은 여전히 유효하다.
97
98 df = pd.read_csv('../ADP_Python/data/cats.csv')
99 dv = 'Bwt'
100 iv = 'Sex'
101 conditions = {'A':'M', 'B':'F'}
102
103 group_A = df[df[iv] == conditions['A']]
104 group_B = df[df[iv] == conditions['B']]
105
106 # 3-2. Visualization
107 fig, axes = plt.subplots(nrows=2, ncols=1)
108
109 sns.histplot(group_A[dv], bins=15, ax=axes[0])
110 sns.histplot(group_B[dv], bins=15, ax=axes[1])
111 plt.tight_layout
112 plt.show()
113
114 # 3-3. Statistical Test
115 print(stats.ttest_ind(group_A[dv], group_B[dv], equal_var=False)) # Welch's t-test
116

```

```
117 | print(f'Group A Mean: {group_A[dv].mean()}')  
118 | print(f'Group B Mean: {group_B[dv].mean()}')
```