

## ZAD 1

**Przykład pracy badawczej:** Kelter, R. Analysis of Bayesian posterior significance and effect size indices for the two-sample t-test to support reproducible medical research. BMC Med Res Methodol 20, 88 (2020).

<https://doi.org/10.1186/s12874-020-00968-2>

Artykuł omawia między innymi możliwość zastąpienia p-value metodami bayesowskimi (ze względu na to, że p-value ma swoje ograniczenia, powoduje powszechne nieporozumienia i fałszywą interpretację badań). Autorzy wykonują liczne symulacje, by porównać metody bayesowskie z powszechnymi miarami jak p-value dla testu t-Studenta i t-Welcha.

## ZAD 2

### 1. Test t- Studenta dla próbek niezależnych

```
from math import sqrt
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from scipy.stats import sem
from scipy.stats import t

# t-test for independent samples
print('Student's t-Test for Independent Samples')
# function for calculating the t-test for two independent samples
new *
def independent_ttest_student(data1, data2, alpha):
    # calculate means
    mean1, mean2 = mean(data1), mean(data2)
    # calculate standard errors
    se1, se2 = sem(data1), sem(data2)
    # standard error on the difference between the samples
    sed = sqrt(se1**2.0 + se2**2.0)
    # calculate the t statistic
    t_stat = (mean1 - mean2) / sed
    # degrees of freedom
    df = len(data1) + len(data2) - 2
    # calculate the critical value
    cv = t.ppf(1.0 - alpha, df)
    # calculate the p-value
    p = (1.0 - t.cdf(abs(t_stat), df)) * 2.0
    # return everything
    return t_stat, df, cv, p

# seed the random number generator
seed(1)
# generate two independent samples
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
# calculate the t test
alpha = 0.05
t_stat, df, cv, p = independent_ttest_student(data1, data2, alpha)
print('t=%.3f, df=%d, cv=%.3f, p=%.3f' % (t_stat, df, cv, p))
# interpret via critical value
if abs(t_stat) <= cv:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')
# interpret via p-value
if p > alpha:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')
```

Uruchomienie przykładu najpierw oblicza test. Wyniki przedstawiono poniżej.

```
C:\Users\Mo\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\Mo\Documents\Studia\SN_lab\SN\PWR_2023_SN\lab4\independent t-Student.py"
Student's t-Test for Independent Samples
t=-2.262, df=198, cv=1.653, p=0.025
Reject the null hypothesis that the means are equal.
Reject the null hypothesis that the means are equal.

Process finished with exit code 0
```

Drukowane są wyniki testu, w tym statystyka t, stopnie swobody, wartość krytyczna i wartość p.

Widzimy, że zarówno statystyka t, jak i wartość p odpowiadają wynikom funkcji SciPy. Wydaje się, że test został poprawnie zaimplementowany.

Statystyka t i wartość p są następnie wykorzystywane do interpretacji wyników testu. Stwierdzamy, że zgodnie z naszymi oczekiwaniami istnieją wystarczające dowody, aby odrzucić hipotezę zerową, stwierdzając, że średnie próbki są prawdopodobnie różne.

## 2. Test t- Studenta dla próbek zależnych

```
from math import sqrt
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from scipy.stats import t
print('Student's t-Test for Dependent Samples')#t-test for dependent samples
new *
def dependent_ttest_student(data1, data2, alpha): # function for calculating the t-test for two dependent samples
    # calculate means
    mean1, mean2 = mean(data1), mean(data2)
    # number of paired samples
    n = len(data1)
    # sum squared difference between observations
    d1 = sum([(data1[i]-data2[i])**2 for i in range(n)])
    # sum difference between observations
    d2 = sum([data1[i]-data2[i] for i in range(n)])
    # standard deviation of the difference between means
    sd = sqrt((d1 - (d2**2 / n)) / (n - 1))
    # standard error of the difference between the means
    sed = sd / sqrt(n)
    # calculate the t statistic
    t_stat = (mean1 - mean2) / sed
    # degrees of freedom
    df = n - 1
    # calculate the critical value
    cv = t.ppf(1.0 - alpha, df)
    # calculate the p-value
    p = (1.0 - t.cdf(abs(t_stat), df)) * 2.0
    # return everything
    return t_stat, df, cv, p

# seed the random number generator
seed(1)
# generate two independent samples (pretend they are dependent)
data1 = 5 * randn(100) + 50
data2 = 5 * randn(100) + 51
# calculate the t test
alpha = 0.05
t_stat, df, cv, p = dependent_ttest_student(data1, data2, alpha)
print('t=%.3f, df=%d, cv=%.3f, p=%.3f' % (t_stat, df, cv, p))
# interpret via critical value
if abs(t_stat) <= cv:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')
# interpret via p-value
if p > alpha:
    print('Accept null hypothesis that the means are equal.')
else:
    print('Reject the null hypothesis that the means are equal.')
```

Uruchomienie przykładu oblicza sparowany test t na przykładowym problemie. Otrzymano następujące wyniki

```
C:\Users\Mo\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\Mo\Documents\Studia\SN_lab\SN\PWR_2023_SN\lab4\depent t-Student.py"
Student's t-Test for Dependent Samples
t=-2.372, df=99, cv=1.660, p=0.020
Reject the null hypothesis that the means are equal.
Reject the null hypothesis that the means are equal.

Process finished with exit code 0
```

Obliczona statystyka t i wartość p odpowiadają temu, czego oczekujemy od implementacji biblioteki SciPy. Sugeruje to, że implementacja jest prawidłowa.

Interpretacja statystyki testu t z wartością krytyczną i wartości p z poziomem istotności daje znaczący wynik, odrzucając hipotezę zerową, że średnie są równe