Zadanie9

May 24, 2021

[1]:
```
import pandas

data_frame = pandas.read_csv("zad9.csv")
data = data_frame["Czas"]

data_frame
```

[1]:

	Czas
0	3150
1	2669
2	2860
3	3033
4	2364
5	2423
6	2862
7	2575
8	2843
9	2827
10	3161
11	3134
12	3124
13	2570
14	2959

Mamy niewiele obserwacji i nie znamy wariancji, więc przybliżać będziemy rozkładem t-studenta.

$$t = \frac{\bar{X} - \mu}{S} \sqrt{n-1}$$

Z wykładu wiemy, że

$$t \sim T(n-1)$$

chcemy żeby wartość znalazła się w przedziale ufności. Zatem dla ustalonego α

$$t_{\frac{\alpha}{2}} < \frac{\bar{X} - \mu}{S} \sqrt{n-1} < t_{1-\frac{\alpha}{2}}$$

czyli

$$\bar{X} - t_{1-\frac{\alpha}{2}} \frac{S}{\sqrt{n-1}} < \mu < \bar{X} - t_{\frac{\alpha}{2}} \frac{S}{\sqrt{n-1}}$$

.

Policzymy wszyskie potrzebne wartości, najpierw n i $\bar{X}$

[2]:
```
import numpy

n = len(data)
print("n = ", n);

mean = numpy.mean(data)
print("Średnia = ", mean)
```

```
n =  15
Średnia =  2836.9333333333334
```

Następnie wartość S

$$S = \sqrt{\frac{1}{n} \sum_{k=1}^{n} (X_k - \bar{X})^2}$$

[3]:
```
import math

S = 0
for x in data:
    S += (x - mean)**2
S = S / n
S = math.sqrt(S)

print("S = ", S)
```

```
S =  257.0855802170856
```

Kolejnym krokiem jest wyznaczenie wartości krytycznych $t_{\frac{\alpha}{2}}$ oraz $t_{1-\frac{\alpha}{2}}$. Ustalmy poziom istotności $\alpha = 0.1$.

[4]:
```
from scipy import stats

a = 0.1
critical_values = stats.t.interval(1-a, n-1)
critical_values
```

[4]:
```
(-1.7613101357748566, 1.7613101357748562)
```

Zatem $t_{\frac{\alpha}{2}} \approx -1.76$ oraz $t_{1-\frac{\alpha}{2}} \approx 1.76$

[5]:
```
fl, fp = critical_values
left_bound = mean - ( fp * ( S / math.sqrt(n-1) ) )
rigt_bound = mean - ( fl * ( S / math.sqrt(n-1) ) )

print("Lewe ograniczenie: ", left_bound)
print("Prawe ograniczenie: ", rigt_bound)
```

```
Lewe ograniczenie:  2715.915455053376
Prawe ograniczenie:  2957.951211613291
```

Czyli nasz przedział ufności dla średniej μ przy poziomie ufności $\alpha = 0.1$ to w przybliżeniu $(2715.91, 2957.95)$

W kolejnej części zadania obliczymy *p_value* dla wybranych μ_0. Wzór na test statystyczny to

$$t = \frac{\bar{X} - \mu_0}{S}\sqrt{n-1}$$

$$t \sim T(n-1)$$

[6]:
```
mu_0 = 2840
mu_1 = 2850
mu_2 = 2875

t_0 = ( (mean - mu_0) * math.sqrt(n-1) ) / S
t_1 = ( (mean - mu_1) * math.sqrt(n-1) ) / S
t_2 = ( (mean - mu_2) * math.sqrt(n-1) ) / S

print("t_0 = ", t_0)
print("t_1 = ", t_1)
print("t_2 = ", t_2)
```

```
t_0 =  -0.04463267047656822
t_1 =  -0.1901739872479892
t_2 =  -0.5540272791765417
```

Otrzymaliśmy same ujemne wartości zatem licząc *p_value* musimy policzyć pole "na lewo" od t i pomnożyć razy dwa, bo rozkład t-studenta jest symetryczny

$$p\_value = 2 * F(t)$$

[11]:
```
p_value0 = 2 * stats.t.cdf(t_0, n-1)
p_value1 = 2 * stats.t.cdf(t_1, n-1)
p_value2 = 2 * stats.t.cdf(t_2, n-1)

print("p_value dla t_0 = ", p_value0)
print("p_value dla t_1 = ", p_value1)
```

```python
print("p_value dla t_2 = ", p_value2)
```

```
p_value dla t_0 =  0.9650304721854002
p_value dla t_1 =  0.8519027195830752
p_value dla t_2 =  0.5883008855237514
```

Wszystkie te wartości mieściły się w naszym przedziale ufności dla średniej i ich *p_value* również wskazuje na to że nie zostałyby odrzucone przy takim poziomie pewności.