Zadanie2

May 23, 2021

Wczytamy dane z pliku *zad2.csv*.

[1]:
```
import pandas
data_frame = pandas.read_csv("zad2.csv")
data_frame
```

[1]:

	A	B	C	D
0	41	44.0	43.0	34.0
1	43	40.0	40.0	37.0
2	45	37.0	42.0	37.0
3	44	43.0	41.0	41.0
4	42	41.0	41.0	37.0
5	48	43.0	39.0	39.0
6	49	42.0	45.0	36.0
7	48	40.0	38.0	41.0
8	47	43.0	40.0	42.0
9	45	42.0	45.0	37.0
10	47	44.0	34.0	37.0
11	45	42.0	43.0	33.0
12	46	40.0	43.0	42.0
13	45	37.0	42.0	36.0
14	48	NaN	42.0	NaN
15	40	NaN	39.0	NaN
16	44	NaN	NaN	NaN
17	39	NaN	NaN	NaN

Jak widać, tym razem pracować będziemy z grupami, w których liczba próbek nie jest taka sama, dlatego zaczniemy od przekształcenia reprezentacji do wygodniejszej formy.

[2]:
```
import numpy
data = {}

groups = data_frame.columns;

for group in data_frame.columns:
    data[group] = []
    for x in data_frame[group]:
        if not numpy.isnan(x):
```

```
            data[group].append(x)
```

Przeprowadzimy analizę wariancji (ANOVA), w celu ustalenia czy **średnia liczba ogłoszeń w trzech gazetach jest taka sama**. Inaczej:

$$H_0 : x_{A\bullet} = x_{B\bullet} = x_{C\bullet}$$

$$H_a : \exists_{x_{i\bullet}, x_{j\bullet}, i \neq j} \; x_{i\bullet} \neq x_{j\bullet}$$

Wartości potrzebne do przeprowadzenia analizy można zaprezentować w postaci tabelki, w stosunku do poprzedniego zadania zmieni się wzór na stopnie swobody, gdyż w każdej grupie jest ich inna liczba.

df	SS	MS	f	
$I-1$	SSA	MSA	$\frac{MSA}{MSE}$	p-value
$\sum_{i=1}^{I}(J_i - 1)$	SSE	MSE		

gdzie I oznacza liczbę grup, a J_i obserwacji na i-tą grupę. Pierwszą i drugą kolumnę liczymy na podstawie danych, a każde kolejne na podstawie poprzednich.

Korzystając ze wzorów z wykładu, na początku ustalimy wartość zmienności międzygrupowej SSA

$$SSA = \sum_{i=1}^{I} \sum_{j=1}^{J_i} (x_{i\bullet} - \bar{x})^2$$

gdzie

$$x_{i\bullet} = \frac{1}{J_i} \sum_{j=1}^{J} x_{ij},$$

$\bar{x}$ średnią wszystkich obserwacji, czyli

$$\bar{x} = \frac{1}{\sum_{i=1}^{I} J_i} \sum_{i,j} x_{ij}.$$

Najpierw policzymy wartości potrzebnych średnich $x_{i\bullet}$ i $\bar{x}$

[3]:
```python
from pprint import pprint
means = {}

for group in groups:
    means[group] = numpy.mean(data[group])

global_header = "global"
```

```
flatten_values = sum(list(data.values()), [])

means[global_header] = numpy.mean(flatten_values)

pprint(means)
```

```
{'A': 44.77777777777778,
 'B': 41.285714285714285,
 'C': 41.0625,
 'D': 37.785714285714285,
 'global': 41.45161290322581}
```

Po czym możemy przejść do wyznaczenia SSA

[4]:
```
ssa = 0
for group in groups:
    ssa += len(data[group]) * (means[group] - means[global_header])**2

print('SSA = ', ssa)
```

```
SSA =  390.0919418842808
```

Z wykładu wiemy że z dokładnością do stałej

$$SSA \sim \chi^2(I-1)$$

Teraz możemy przejść do wyznaczenia zmienności wewnątrzgrupowej SSE

$$SSE = \sum_{i,j} (x_{ij} - x_{i\bullet})^2$$

[5]:
```
sse = 0
for group in groups:
    for x in data[group]:
        sse += (x - means[group])**2
print("SSE = ", sse)
```

```
SSE =  429.26289682539664
```

Z wykładu wiemy, że

$$SSE \sim \chi^2\left(\sum_{i=1}^{I}(J_i - 1)\right)$$

Następnie możemy przejść do wyznaczenia MSA i MSE

$$MSA = \frac{SSA}{I-1}$$

$$MSE = \frac{SSE}{\sum_{i=1}^{I}(J_i - 1)},$$

czyli odpowiednio SSA i SSE podzielone przez ich stopnie swobody.

[6]:
```
deg_of_freedom_ssa = len(groups) - 1

deg_of_freedom_sse = 0
for group in groups:
    deg_of_freedom_sse += len(data[group]) - 1

msa = ssa / deg_of_freedom_ssa
mse = sse / deg_of_freedom_sse

print("MSA = ", msa)
print("MSE = ", mse)
```

```
MSA =  130.03064729476026
MSE =  7.40108442802408
```

Kolejnym krokiem jest wyznaczenia wartości F

$$f = \frac{MSA}{MSE},$$

a stąd, że

$$MSA \sim \frac{\chi^2(I-1)}{I-1}$$

i

$$MSE \sim \frac{\chi^2\left(\sum_{i=1}^{I}(J_i - 1)\right)}{\sum_{i=1}^{I}(J_i - 1)}$$

wnioskujemy następujące (przy założeniu że H_0 jest prawdziwe)

$$f = \frac{MSA}{MSE} \sim F\left(J - 1, \sum_{i=1}^{I}(J_i - 1)\right),$$

bo rozkład Fishera-Snedecore'a może być przedstawiony jako stosunek przeskalowanych rozkładów χ^2.

[7]:
```
f = msa / mse
print("f = ", f)
```

```
f =  17.56913443689433
```

Możemy teraz policzyć wartości krytyczne dla tego rozkładu i wybranego obszaru krytycznego. Sprawdźmy jak wyglądałby on dla $\alpha = 0.1$

[8]:
```
from scipy import stats

f_critical = stats.f.interval(0.9, deg_of_freedom_ssa, deg_of_freedom_sse)
print("Wartości skrajne: ", f_critical)
```

```
Wartości skrajne:  (0.11663769974251183, 2.7635518374327885)
```

Widzimy że nasze f jest większe od wartości skrajnej, zatem odrzucilibyśmy hipotezę.