TP6: Inférences statistiques pour deux échantillons

Tanguy ROUDAUT — Tadios QUINIO FIPASE 24

18 Octobre 2022

1 Bien doser l'alliage

On teste la résistance à la traction de sept éprouvettes pour quatre aciers différents.

Proportion de carbone	**Résistance à la traction**						
	n°1	n°2	n°3	n°4	n°5	n°6	n°7
0.1%	23.05	36	31.1	32.65	30.9	31.4	30.85
0.2%	41.85	25.65	46.7	34.5	36.65	31.45	36.13
0.4%	47.05	43.45	43	38.65	41.85	35.45	41.57
0.6%	49.65	73.9	66.45	74.55	62.4	63.75	65.11

Dans un premier temps, on peut préparer le programme avec les valeurs que nous avons :

```
p = 4
n = 7

resistance_01 = [23.05, 36, 31.1, 32.65, 30.9, 31.4, 30.85]
resistance_02 = [41.85, 25.65, 46.7, 34.5, 36.65, 31.45, 36.13]
resistance_04 = [47.05, 43.45, 43, 38.65, 41.85, 35.45, 41.57]
resistance_06 = [49.65, 73.9, 66.45, 74.55, 62.4, 63.75, 65.11]
p_carbonne    = [0.1, 0.2, 0.4, 0.6]

resistance = [resistance_01, resistance_02, resistance_04, resistance_06]
```

Listing 1 – Exploitation de l' énnoncé

Question 1 : Identifier la nature des deux variables aléatoires de ce problème.

- Variables qualitatives ordinales → les différentes proportions de carbone
- Variables quantitatives discrètes → les différentes résistances

Question 2 : Calculr les moyennes et les variances de chaque sous-population définie par la proportion de carbone. Représenter les boites à moustaches sur une même figure. Commenter qualitativement la figure résultante quant à l'influence de la proportion de carbone sur la résistance à la traction.

- **Calcul de la moyenne :** $\overline{y}_{i\bullet} = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij}$

```python
resistance_classe_mean = []
for i in range(p):
    sum = 0
    for j in range(n):
        sum += resistance[i][j]
    resistance_classe_mean.append(sum/n)
```

Listing 2 – Calcul de la moyenne

- **Calcul de la variance :** $S_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(y_{ij} - \overline{y}_{i\bullet})$

```python
resistance_classe_var = []
for i in range(p):
    sum = 0
    for j in range(n):
        sum += (resistance[i][j] - resistance_classe_mean[i])**2
    resistance_classe_var.append(sum/(n-1))
```

Listing 3 – Calcul de la variance

- **Résultat :**

```python
print("Question 2:")
for i in range(p):
    print("\t--> ", p_carbonne[i], "% de carbonne: moy =",
    resistance_classe_mean[i], " et var =", resistance_classe_var[i])
```

Listing 4 – Affichage du résultat

```
Question 2:
    -->  0.1 % de carbonne: moy = 30.850  et var = 15.161
    -->  0.2 % de carbonne: moy = 36.132  et var = 46.517
    -->  0.4 % de carbonne: moy = 41.574  et var = 13.611
    -->  0.6 % de carbonne: moy = 65.115  et var = 69.396
```

Listing 5 – Résultat

- **Boite à moustaches :**

Grâce à la boite à moustache (figure 1) et aux valeurs précédentes, on constate facilement que la proportion de carbone à une influence sur la résistance à la traction.
Si l'on prend les deux extrémités, soit une proportion de carbone de 0.1% et 0.6% nous obtenons les valeurs suivantes :

Proportion	min	max	moy
0.1%	23.05	36	30.850
0.6%	49.65	74.55	65.115

Ce qui nous permet de dire que la proportion de carbone à bien une influence sur la résistance à la traction, la proportion de 0.6% à une valeur max, min, quartile 1/2/3 plus important que la proportion de 0.1%.

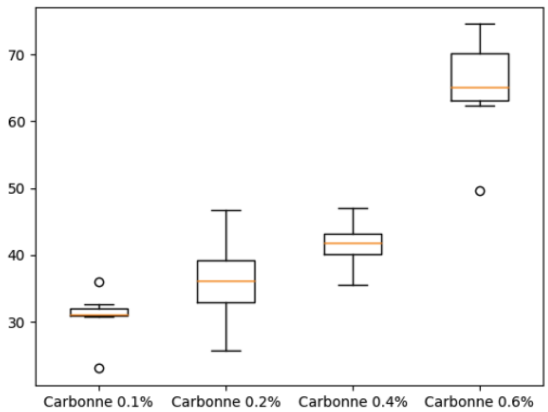

FIGURE 1 – Boite à moustache de la résistance à la traction de sept éprouvettes pour différent acier

Question 3 : Mener ce test en adoptant la procédure générale des tests. Répondre à la question : la proportion de carbone a-t-elle une influence sur la résistance à la traction ? On utilisera *scipy.stats.f.ppf* et *scipy.stats.f.cdf*

— **Calcul de la moyenne globale :** $\overline{y}_{\bullet\bullet} = \frac{1}{N}\sum_{i=1}^{p}\sum_{j=1}^{n} y_{ij}$

```
N = p * n
resistance_global_mean = 0
for i in range(p):
    for j in range(n):
        resistance_global_mean += resistance[i][j]
resistance_global_mean /= N
```

Listing 6 – Calcul de la moyenne globale

— **Calcul de la dispersion intraclasse totale :** $S_W^2 = \frac{1}{N}\sum_{i=1}^{p} n_i S_i^2$

```
disp_intraclasse_tot = 0
for i in range(p):
    disp_intraclasse_tot += (n*resistance_classe_var[i])
disp_intraclasse_tot /= N
```

Listing 7 – Calcul de la dispersion intraclasse totale

— **Calcul de la dispersion interclasse :** $S_B^2 = \frac{1}{N}\sum_{i=1}^{p} n_i(\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2$

```
disp_interclasse = 0
for i in range(p):
    disp_interclasse += n*(resistance_classe_mean[i]-resistance_global_mean)**2
disp_interclasse /= N
```

Listing 8 – Calcul de la dispersion interclasse

— **Résultat :**

```
print("Question 3 - CALCUL:")
print("\t--> Moyenne global=", resistance_global_mean)
print("\t--> Dispersion intraclasse totale=", disp_intraclasse_tot)
print("\t--> Dispersion interclasse=", disp_interclasse, end="\n\n")
```

Listing 9 – Affichage du résultat

```
Question 3 - CALCUL:
    --> Moyenne global= 43.418214285714285
    --> Dispersion intraclasse totale= 36.171686904761906
    --> Dispersion interclasse= 171.30450446428577
```

Listing 10 – Résultat

1. **Grandeur d'intérêt :** Proportion de carbone.
2. **Hypothèse nulle,** $H0$ **:** La proportion de carbone n'a pas d'influence sur la résistance à la traction.
3. **Hypothèse alternative,** $H1$ **:** La proportion de carbone a une influence sur la résistance à la traction.
4. **Niveau de confiance :** 95%
5. **Test statistique :** $F_0 = \frac{S_B^2/(p-1)}{S_W^2/(N-p)}$ estimée par f_0 à partir de l'échantillon.

6. **Rejet de $H0$ si :**
   - Région critique : $f_0 > f_{\alpha,(p-1),(N-p)}$
   - p-valeur : $p - valeur < 0.05$
7. **Calculs :**
   - Formules utilisées :

$$f_0 = \frac{S_B^2/(p-1)}{S_W^2/(N-p)} \qquad (1)$$

$$f_{\alpha,(p-1),(N-p)} = F_{F_0}^{-1}(1-\alpha) \qquad (2)$$

$$p\text{-}valeur = 1 - F_{f_{\alpha,(p-1),(N-p)}}(f_0) \qquad (3)$$

```python
ic = 95
alpha = 1 - (ic/100)
f = stats.f.ppf(1-alpha, (p-1), (N-p))
f0 = ((disp_interclasse)/(p-1))/((disp_intraclasse_tot)/(N-p))
p_valeur = 1 - stats.f.cdf(f0, (p-1), (N-p))

print("Question 3 - TEST:")
print("\t--> f=", f)
print("\t--> f0=", f0)
print("\t--> p-valeur=", p_valeur)
```

Listing 11 – Code Python question 3

```
Question 3 - TEST:
    --> f= 3.0087865704473615
    --> f0= 37.88698158652568
    --> p-valeur= 2.910348628759607e-09
```

Listing 12 – Résultat du code

8. **Décision :**

Critéres de rejet de H0	
pour α fixé	avec *p-valeur*
$f_0 > f_{\alpha,(p-1),(N-p)}$	$p\text{-}valeur < 0.05$

$$\left.\begin{aligned} f_0 &= 37,886 \\ f_{\alpha,(p-1),(N-p)} &= 3,008 \\ p\text{-}valeur &= 2,910.10^{-09} \end{aligned}\right\} \quad \begin{aligned} f_0 &> f_{\alpha,(p-1),(N-p)} \\ p\text{-}valeur &< 0.01 \end{aligned}$$

Les résultats du test statistique sont hautement significatifs, ils montrent que *H0* peut être rejeté puisque toutes les conditions sont validées.
La VA ne suit pas une loi de Fisher, soit la proportion de carbone a une influence sur la résistance à la traction.

2 Code complet

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt


# question 1
# Proportion de carbonne --> qualitative
# Résistance à la traction  --> quantitative

# question 2
p = 4
n = 7

resistance_01 = [23.05, 36, 31.1, 32.65, 30.9, 31.4, 30.85]
resistance_02 = [41.85, 25.65, 46.7, 34.5, 36.65, 31.45, 36.13]
resistance_04 = [47.05, 43.45, 43, 38.65, 41.85, 35.45, 41.57]
resistance_06 = [49.65, 73.9, 66.45, 74.55, 62.4, 63.75, 65.11]
p_carbonne    = [0.1, 0.2, 0.4, 0.6]

resistance = [resistance_01, resistance_02, resistance_04, resistance_06]

resistance_classe_mean = []
for i in range(p):
    sum = 0
    for j in range(n):
        sum += resistance[i][j]
    resistance_classe_mean.append(sum/n)

resistance_classe_var = []
for i in range(p):
    sum = 0
    for j in range(n):
        sum += (resistance[i][j] - resistance_classe_mean[i])**2
    resistance_classe_var.append(sum/(n-1))


print("Question 2:")
for i in range(p):
    print("\t--> ", p_carbonne[i], "% de carbonne: moy =", resistance_classe_mean[i], " et var =", resistance_classe_var[i])
print("\n")

plt.boxplot(resistance, labels=['Carbonne 0.1%', 'Carbonne 0.2%', 'Carbonne 0.4%', 'Carbonne 0.6%'])
plt.title("Boite à moustache de la résistance à la traction de \nsept éprouvettes pour différent acier")
plt.show()



# question 3
N = p * n
resistance_global_mean = 0
for i in range(p):
    for j in range(n):
        resistance_global_mean += resistance[i][j]
resistance_global_mean /= N


disp_intraclasse_tot = 0
for i in range(p):
    disp_intraclasse_tot += (n*resistance_classe_var[i])
disp_intraclasse_tot /= N

```

```python

disp_interclasse = 0
for i in range(p):
    disp_interclasse += n*(resistance_classe_mean[i]-resistance_global_mean)**2
disp_interclasse /= N


#1. Paramètre d intérêt : proportion de carbone
#2. Hypothèse nulle H0 : La proportion de carbone n'a pas d'influence sur la résistance à la traction
#3. Hypothèse alternative H1 : La proportion de carbone a une influence sur la résistance à la traction
#4. Tests statistique: test de fisher
#5. Niveau de confiance: 95%
#6. Rejet de H0 si f0 > f ou si p-valeur < 0.05

ic = 95
alpha = 1 - (ic/100)
f = stats.f.ppf(1-alpha, (p-1), (N-p))
f0 = ((disp_interclasse)/(p-1))/((disp_intraclasse_tot)/(N-p))
p_valeur = 1 - stats.f.cdf(f0, (p-1), (N-p))

print("Question 3:")
print("\t--> Moyenne global=", resistance_global_mean)
print("\t--> Dispersion intraclasse totale=", disp_intraclasse_tot)
print("\t--> Dispersion interclasse=", disp_interclasse, end="\n\n")
print("\t----------- TEST -----------\n")
print("\t--> f=", f)
print("\t--> f0=", f0)
print("\t--> p-valeur=", p_valeur)
```

Listing 13 – Code Python complet TP6