# TP4: Tests d'hypothèses à partir d'un échantillon

Tanguy ROUDAUT — Tadios QUINIO FIPASE 24

03 Octobre 2022

## 1 Fitness

Regardless of age, about 20% of American adults participate in fitness activities at least twice a week. However, these fitness activities change as the people get older, and occasionally participants become nonparticipants as they age. In a local survey of n = 100 adults over 40 years old, a total of 15 people indicated that they participated in a fitness activity at least twice a week.

**Question 1 :** Do these data indicate that the participation rate for adults over 40 years of age is significantly less than the 20% figure ? Calculate the p-value and use it to draw the appropriate conclusions.

1. **Grandeur d'intérêt :** Proportion de personne de plus de 40 ans participent à du fitness.
2. **Hypothèse nulle, H0 :** $p = p0 = 20\%$, soit la proportion de personnes de plus de 40 ans qui participent à du fitness est de 20%.
3. **Hypothèse alternative, H1 :** $p < p0$, soit la proportion de personnes de plus de 40 ans qui participent à du fitness est inférieur à 20%.
4. **Niveau de confiance :** 95%
5. **Test statistique :** Test statistique unilatérale pour une proportion d'un grand échantillon :

$$n = 100 \text{ et } p_0 = 0.2 \qquad \begin{cases} n(1-p_0) = 80 < 5 \\ n.p_0 = 20 < 5 \end{cases}$$

$$z_0 = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0 * (1-p_0)}{n}}} \quad (1)$$

$$z_\alpha = erf^{-1}(1-\alpha) \quad (2)$$

$$p\text{-}valeur = erf(t_0) \quad (3)$$

```python
n = 100
p0 = .2
p = 15/n
ic = 95
alpha = 1 - ic / 100
z = stats.norm.ppf(1 - alpha)
z0 = (p - p0)/np.sqrt(p0*(1-p0)/n)
p_valeur = stats.norm.cdf(z0)

print("Question 1:")
print(" z:", round(z, 3))
print(" z0:", round(z0, 3))
print(" p-valeur:", p_valeur, end="\n\n")
```

Listing 1 – Code Python question 1

```
Question 1:
 z: 1.645
 z0: -1.25
 p-valeur: 0.10564977366685518
```

Listing 2 – Résultat du code

6. **Rejet de H0 ?**

| Critéres de rejet de H0 | |
|---|---|
| pour $\alpha$ fixé | avec *p-valeur* |
| $z_0 < -z_\alpha$ | *p-valeur* $< 0.05$ |

$$\left.\begin{aligned} z_0 &= -1,25 \\ -z_\alpha &= -1,645 \\ p\text{-}valeur &= 0,1 \end{aligned}\right\} \quad \begin{aligned} z_0 &> -z_\alpha \\ p\text{-}valeur &> 0.05 \end{aligned}$$

Les résultats du test statistique montrent que *H0* ne peut être rejeté puisqu'aucune des conditions n'est validée.
On ne peut pas dire que le taux de personnes âgé de plus de 40 ans qui font du fitness est inférieur à 20%.

## 2 Météo

Les données de la variable T 16 représentent les températures maximales enregistrées en 2016 dans plusieurs stations météo du Portugal.
$T16 = [39, 39, 40, 33, 36, 40, 37, 41, 39, 34, 42, 41, 42, 44, 42, 42, 39, 42, 41, 40, 43, 43, 40, 39, 37]$
Un rapport du centre de Météorologie portugais précise que la moyenne des températures maximales enregistrées au Portugal au fil des ans est de 37.5 °C.

**Question 2 :** L'année 2016 est-elle exceptionnelle ? Détaillez votre démarche et les hypothèses du test que vous utilisez.

1. **Grandeur d'intérêt :** Moyenne de température
2. **Hypothèse nulle, H0 :** $\mu = \mu_0 = 37,5$, soit la moyenne de température pour l'année 2016 est de 37.5 °C.
3. **Hypothèse alternative, H1 :** $\mu \neq \mu_0 = 37,5$, soit la moyenne de température pour l'année 2016 est différente de 37.5 °C.
4. **Niveau de confiance :** 95%
5. **Test statistique :** Test statistique bilatéral pour la moyenne d'une distribution normale de variance inconnue :

$$t_0 = \frac{\overline{Xn} - \mu_0}{\frac{S_{n-1}}{\sqrt{n}}} \tag{4}$$

$$t_{n-1,\alpha/2} = F^{-1}_{Student,n-1}(1 - \frac{\alpha}{2}) \tag{5}$$

$$p\text{-}valeur = 2 - 2erf(\mid t_0 \mid) \tag{6}$$

```
x = np.array([39, 39, 40, 33, 36, 40, 37, 41, 39, 34, 42, 41, 42, 44, 42, 42,
              39, 42, 41, 40, 43, 43, 40, 39, 37])
n = len(x)
u0 = 37.5
```

```python
xn = np.mean(x)
ecart_type_x = np.std(x, ddof=1)
ic = 95
alpha = 1 - ic / 100
t = stats.t.ppf(1 - (alpha / 2), n - 1)
t0 = (xn - u0)/(ecart_type_x/np.sqrt(n))
p_valeur = 2 - 2 * stats.norm.cdf(np.abs(t0))

print("Question 2:")
print(" t:", round(t, 3))
print(" t0:", round(t0, 3))
print(" p-valeur:", p_valeur, end="\n\n")
```

Listing 3 – Code Python question 2

```
Question 2:
 t: 2.064
 t0: 4.199
 p-valeur: 2.678522342547396e-05
```

Listing 4 – Résultat du code

6. **Rejet de H0 ?**

| **Critéres de rejet de H0** | |
|---|---|
| pour $\alpha$ fixé | avec *p-valeur* |
| $t_0 > t_{n-1,\alpha/2}$ ou $t_0 < -t_{n-1,\alpha/2}$ | *p-valeur* $< 0.05$ |

$$\left.\begin{aligned} t_0 &= 4.199 \\ t_{n-1,\alpha/2} &= 2.064 \\ p\text{-}valeur &= 2,6.10^{-5} \end{aligned}\right\} \quad \begin{aligned} &t_0 > t_{n-1,\alpha/2} \\ &p\text{-}valeur < 0.01 \end{aligned}$$

Les résultats du test statistique sont hautement significatifs, *H0* peut être rejeté puisque les deux conditions sont validées avec une *p-valeur* $< 0.01$.
On peut dire avec un niveau de confiance de 95% que l'année 2016 a été exceptionnelle pour la météo du Portugal.

## 3 Potency of an Antibiotic

A drug manufacturer claimed that the mean potency of one of its antibiotics was 80%. A random sample of $n = 100$ capsules were tested and produced a sample mean of $\overline{Xn} = 79.7\%$ with a standard deviation of $S_{n-1} = .8\%$.

**Question 3 :** Do the data present sufficient evidence to refute the manufacturer's claim ? Let $\alpha = 0.05$.

1. **Grandeur d'intérêt :** La puissance moyenne des antibiotiques.
2. **Hypothèse nulle, H0 :** $\mu = \mu_0 = 80\%$, soit la puissance moyenne des antibiotiques est de 80%.
3. **Hypothèse alternative, H1 :** $\mu \neq \mu_0 = 80\%$, soit la puissance moyenne des antibiotiques est différente de 80%.
4. **Niveau de confiance :** 95%
5. **Test statistique :** Test statistique bilatéral pour la moyenne d'un grand échantillon de variance connue :

$$z_0 = \frac{\overline{x_n} - \mu_0}{\frac{\sigma}{\sqrt{n}}} \quad (7)$$

$$z_{\alpha/2} = erf^{-1}(1 - \frac{\alpha}{2}) \quad (8)$$

$$p\text{-}valeur = 2 - 2erf(\mid z_0 \mid) \quad (9)$$

```python
n = 100
u0 = 80/100
xn = 79.7/100
ecart_type_x = .8/100
ic = 95
alpha = 1 - ic / 100
z = stats.norm.ppf(1 - (alpha / 2))
z0 = (xn - u0)/(ecart_type_x/np.sqrt(n))
p_valeur = 2 - 2 * stats.norm.cdf(np.abs(z0))

print("Question 3:")
print(" z:", round(z, 3))
print(" z0:", round(z0, 3))
print(" p-valeur:", p_valeur)
```

Listing 5 – Code Python question 3

```
Question 3:
 z: 1.96
 z0: -3.75
 p-valeur: 0.00017683457040162942
```

Listing 6 – Résultat du code

6. **Rejet de H0 ?**

| **Critéres de rejet de H0** | |
|---|---|
| pour $\alpha$ fixé | avec *p-valeur* |
| $z_0 > z_{\alpha/2}$ ou $z_0 < -z_{\alpha/2}$ | *p-valeur* $< 0.05$ |

$$\left.\begin{aligned} z_0 &= 1.96 \\ z_{\alpha/2} &= -3.75 \\ p\text{-}valeur &= 0.0001 \end{aligned}\right\} \quad \begin{aligned} z_0 &> z_{\alpha/2} \\ p\text{-}valeur &< 0.01 \end{aligned}$$

Les résultats du test statistique sont hautement significatifs, *H0* peut être rejeté puisque les deux conditions sont validées avec une *p-valeur* $< 0.01$.
On peut dire avec un niveau de confiance de 95% que la puissance moyenne des antibiotiques est différente de 80%.

# 4 Code complet

```python
import numpy as np
from scipy import stats

# question 1
n = 100
p0 = .2
p = 15/n
ic = 95
alpha = 1 - ic / 100
z = stats.norm.ppf(1 - alpha)
z0 = (p - p0)/np.sqrt(p0*(1-p0)/n)
p_valeur = stats.norm.cdf(z0)
```

```python

print("Question 1:")
print(" z:", round(z, 3))
print(" z0:", round(z0, 3))
print(" p-valeur:", p_valeur, end="\n\n")


# question 2
x = np.array([39, 39, 40, 33, 36, 40, 37, 41, 39, 34, 42, 41, 42, 44, 42, 42,
              39, 42, 41, 40, 43, 43, 40, 39, 37])
n = len(x)
u0 = 37.5
xn = np.mean(x)
ecart_type_x = np.std(x, ddof=1)
ic = 95
alpha = 1 - ic / 100
t = stats.t.ppf(1 - (alpha / 2), n - 1)
t0 = (xn - u0)/(ecart_type_x/np.sqrt(n))
p_valeur = 2 - 2 * stats.norm.cdf(np.abs(t0))

print("Question 2:")
print(" t:", round(t, 3))
print(" t0:", round(t0, 3))
print(" p-valeur:", p_valeur, end="\n\n")


# question 3
n = 100
u0 = 80/100
xn = 79.7/100
ecart_type_x = .8/100
ic = 95
alpha = 1 - ic / 100
z = stats.norm.ppf(1 - (alpha / 2))
z0 = (xn - u0)/(ecart_type_x/np.sqrt(n))
p_valeur = 2 - 2 * stats.norm.cdf(np.abs(z0))

print("Question 3:")
print(" z:", round(z, 3))
print(" z0:", round(z0, 3))
print(" p-valeur:", p_valeur)
```

Listing 7 – Code Python complet TP4