TP5: Tests du χ^2 sur une distribution d'un échantillon

Tanguy ROUDAUT — Tadios QUINIO FIPASE 24

11 Octobre 2022

1 Retrouver la loi

À l'aide de 100 antennes GPS, en des points différents du globe [1], le nombre de satellites visibles a été compté :

Nombre de satellites	1	2	3	4	5	6	7	8	9
Nombre d'observations	6	15	9	25	17	10	8	7	3

Question 1 : Estimer la moyenne et la variance de l'échantillon. Indication : on utilisera les opérateurs terme à terme python : ** et * ?

Formules utilisées :

$$\overline{x}_n = \frac{\sum (n_i.x_i)}{n} \qquad (1)$$

$$S^2_{n-1} = \frac{1}{n-1} \sum_{k=1}^{k} n_k.(x_k - \overline{x}_n)^2 \qquad (2)$$

```python
obs = np.array([6, 15, 9, 25, 17, 10, 8, 7, 3])
sat = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

nObs = 0
nSat = 0
for i in range(len(obs)):
    nObs += obs[i]
    nSat += sat[i]

num_mean = 0
for i in range(len(obs)):
    num_mean += obs[i]*sat[i]
mean = num_mean / nObs

num_var = 0
for i in range(len(obs)):
    num_var += (sat[i]-mean)**2
var = num_var/(len(sat)-1)

print("Question 1:")
print(" Moyenne =", mean)
print(’ Variance =’, var, end="\n\n")
```

Listing 1 – Code Python question 1

1. Éventuellement dans des environnements fortement métalliques ou partiellement enterrés

```
Question 1:
    Moyenne = 4.47
    Variance = 7.816012499999999
```

Listing 2 – Résultat du code

Question 2 : Approcher la loi sous-jacente à l'aide d'une loi de Poisson de paramètre $\lambda = 4.47$. Déter- miner les effectifs théoriques pour 0 à 16 satellites. Indication : on utilisera la fonction *poisson.pmf*.

Grâce au code python ci-dessous, nous avons obtenus les effectifs théoriques suivants pour les satellites visibles de 0 à 16 :

Donnés		**Estimés**	
Nb sat	**Nb d'obs**	**Proba**	**eff th**
0	N.C	0.011	1.145
1	6	0.051	5.117
2	15	0.114	11.436
3	9	0.17	17.04
4	25	0.19	19.042
5	17	0.17	17.024
6	10	0.127	12.683
7	8	0.081	8.099
8	7	0.045	4.525
9	3	0.022	2.248
10	N.C	0.01	1.005
11	N.C	0.004	0.408
12	N.C	0.002	0.152
13	N.C	0.001	0.052
14	N.C	0.0	0.017
15	N.C	0.0	0.005
16	N.C	0.0	0.001

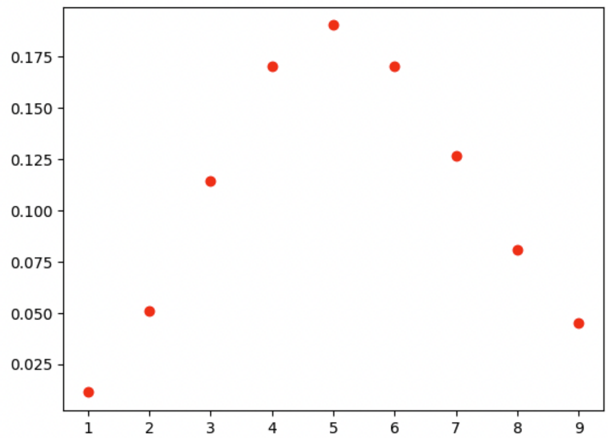

FIGURE 1 – Loi sous-jacente approcher par une loi de Poisson pour 1 à 9 satellites

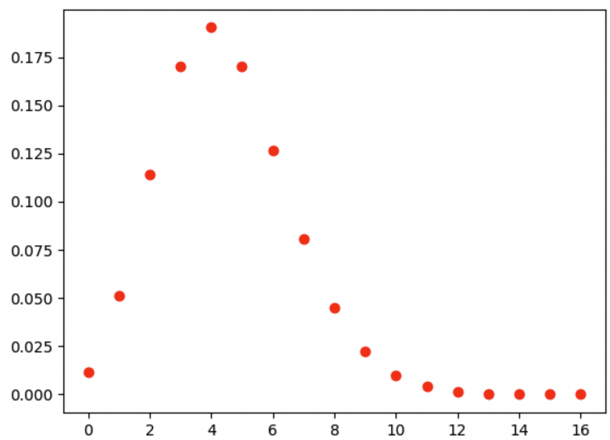

FIGURE 2 – Loi sous-jacente approcher par une loi de Poisson pour 0 à 16 satellites

```python
Lambda = 4.47
poisson_array = np.zeros((17,))
effectifs_th = np.zeros((17,))
print("Question 2:")
print("nb Sat |  nb Obs   |  Proba   |    eff th")
```

```python
print("========================================")
poisson_array[0] = stats.poisson.pmf(0, Lambda)
effectifs_th[0] = nObs * poisson_array[0]
print("   ", 0, "  |   ", "N.C", "   |\t", round(poisson_array[0], 3), "\t|\t", round(effectifs_th[0], 3))
print("----------------------------------------")

for i in range(1, len(sat)+1):
    poisson_array[i] = stats.poisson.pmf(sat[i-1], Lambda)
    effectifs_th[i] = nObs * poisson_array[i]

    print("   ", sat[i-1], "  |    ", round(obs[i-1], 2), "\t |\t", round(poisson_array[i], 3), "\t|\t", round(effectifs_th[i], 3))

print("----------------------------------------")

plt.plot(sat, poisson_array[:9], 'ro')
plt.show()

for i in range(10, 17):
    poisson_array[i] = stats.poisson.pmf(i, Lambda)
    effectifs_th[i] = nObs * poisson_array[i]

    print(" ", i, "  |   ", "N.C", "   |\t", round(poisson_array[i], 3), "\t|\t", round(effectifs_th[i], 3))

print("\n\n")

nSat16 = np.arange(0, 17, 1)
plt.plot(nSat16, poisson_array, 'ro')
plt.show()
```

Listing 3 – Code Python question 2

```
Question 2:
    nb Sat |  nb Obs   |  Proba   |   eff th
    ========================================
       0   |   N.C    |   0.011  |  1.145
    ----------------------------------------
       1   |    6     |   0.051  |  5.117
       2   |    15    |   0.114  |  11.436
       3   |    9     |   0.17   |  17.04
       4   |    25    |   0.19   |  19.042
       5   |    17    |   0.17   |  17.024
       6   |    10    |   0.127  |  12.683
       7   |    8     |   0.081  |  8.099
       8   |    7     |   0.045  |  4.525
       9   |    3     |   0.022  |  2.248
    ----------------------------------------
      10   |   N.C    |   0.01   |  1.005
      11   |   N.C    |   0.004  |  0.408
      12   |   N.C    |   0.002  |  0.152
      13   |   N.C    |   0.001  |  0.052
      14   |   N.C    |   0.0    |  0.017
      15   |   N.C    |   0.0    |  0.005
      16   |   N.C    |   0.0    |  0.001
```

Listing 4 – Résultat du code

Question 3 : Peut-on, à un seuil de 95%, considérer que l'échantillon a été produit par cette loi de Poisson ?
Indication : Attention aux hypothèses, on utilisera les fonctions *np.sum* et *chi2.ppf*, les opérateurs terme à terme **** et */*, ainsi que l'extraction d'une tranche d'un vecteur par *tab[i :j]*

Dans un premier temps, on constate que les $n.p_i$ ne sont pas tous supérieurs à 1.
Les extrémités ont des effectifs plus faibles, on va donc les sommer jusqu'à dépasser 5, tout en réduisant de 1 la valeur de k à chaque somme :

```python
i = 0
while (effectifs_th[i] < 5):
    effectifs_th[i+1] += effectifs_th[i]
    effectifs_th[i] = 0
    i += 1

i = len(effectifs_th)-1
while (effectifs_th[i] < 5):
    effectifs_th[i-1] += effectifs_th[i]
    effectifs_th[i] = 0
    i -= 1

effectifs_th = np.delete(effectifs_th, np.where(effectifs_th == 0))
print("Les effectifs théorique après modification dû aux n.pi<5 :", effectifs_th)
```

Listing 5 – $n.p_i < 5$

```
Les effectifs théorique après modification dû aux n.pi<5 :
[ 6.26168177 11.43638366 17.04021166 19.04243653 17.02393826 12.682834 8.09889543
  8.41313506 ]
```

Listing 6 – Résultat du code

Maintenant que le test est conforme, on peut débuter les hypothèses et les calculs :

1. **Grandeur d'intérêt :** La distribution du nombre de satellites visibles par rapport aux nombres de points d'observation.
2. **Hypothèse nulle,** $H0$ **:** La distribution du nombre de satellites par rapport aux nombres de points d'observation a été produite par cette loi de poisson.
3. **Hypothèse alternative,** $H1$ **:** La distribution du nombre de satellites par rapport aux nombres de points d'observation n'a pas été produite par cette loi de poisson.
4. **Niveau de confiance :** 95%
5. **Test statistique :** $\chi_0^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$ estimée par χ_{Obs}^2 à partir de l'échantillon
6. **Rejet de** $H0$ **si :**
   — Région critique : $\chi_{Obs}^2 > \chi_{k-p-1,\alpha}^2$
   — p-valeur : $p - valeur < 0.05$
7. **Calculs :**
   — Formules utilisées :

$$\chi_{Obs}^2 = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i} \quad (3)$$

$$\chi_{k-p-1,\alpha}^2 = F_{\chi_{k-p-1}^2}^{-1}(\alpha) \quad (4)$$

$$p\text{-}valeur = 1 - F_{\chi_{k-p-1}^2}(\chi_{Obs}^2) \quad (5)$$

- Paramètres à prendre en compte :
  $p \rightarrow$ Nombre de paramètres estimés : 3 (moyenne, effectif théorique, loi poisson)
  $k \rightarrow$ Nombre de classe : $\neq 16$, la condition $n.p_i > 5$ n'était pas respectée

```python
k = len(effectifs_th)
p = 3 # la moyenne, l'effectif th et loi poisson
ic = 95
alpha = 1 - ic / 100
chi2 = stats.chi2.ppf(1-alpha, k-p-1)
chi2_Obs = 0

for i in range(k):
    chi2_Obs += ((obs[i]-effectifs_th[i])**2)/(effectifs_th[i])

p_valeur = 1 - stats.chi2.cdf(chi2_Obs, k-p-1)

print("Question 3:")
print(" Chi2:", chi2)
print(" Chi2 Obs:", chi2_Obs)
print(" p-valeur:", p_valeur)
```

Listing 7 – Code Python question 3

```
Question 3:
    Chi2: 9.487729036781154
    Chi2 Obs: 7.585020824300105
    p-valeur: 0.10801814641379404
```

Listing 8 – Résultat du code

8. **Décision :**

Critéres de rejet de H0	
pour α fixé	avec *p-valeur*
$\chi^2_{Obs} > \chi^2_{k-p-1,\alpha}$	*p-valeur* < 0.05

$$\left.\begin{aligned} \chi^2_{Obs} &= 7.585 \\ \chi^2_{k-p-1,\alpha} &= 9.487 \\ p\text{-}valeur &= 0.1 \end{aligned}\right\} \quad \begin{aligned} \chi^2_{Obs} &< \chi^2_{k-p-1,\alpha} \\ p\text{-}valeur &> 0.05 \end{aligned}$$

Les résultats du test statistique montrent que *H0* ne peut être rejeté puisqu'aucune des conditions n'est validée.
La distribution du nombre de satellites par rapport aux nombres de points d'observations a été produite par cette loi de poisson.

2 Participation volontaire ou non ?

Le tableau suivant résume les résultats d'une enquête menée à la Faculté d'Ingénierie de l'Université de Porto, auprès de 129 étudiants de première année. L'objectif de l'enquête était d'évaluer l'attitude des étudiants de première année envers les "rites d'initiation des étudiants de première année". Une des questions était : "J'ai participé à l'initiation de mon plein gré".
Les réponses ont été notées comme suit :

1. Totalement en désaccord
2. En désaccord
3. Sans commentaire
4. D'accord
5. Totalement d'accord

Le tableau d'effectifs pour les variables SEXE et réponses est présenté dans le tableau ci-dessous.

REPONSE		1	2	3	4	5
GENRE	Homme	3	9	18	36	29
	Femme	3	3	1	14	13

Question 4 : Peut-on conclure que les étudiants masculins et féminins ont un comportement différent lors de l'"initiation" ? Quel test allez vous effectuer pour le prouver ?

Nous pouvons dans un premier temps établir le tableau avec les paramètres n_i et n_j :

```python
nij = np.array([[3, 9, 18, 36, 29],
[3, 3, 1, 14, 13]])

sum_ni = np.array([np.sum(nij[0, :]), np.sum(nij[1, :])])
sum_nj = np.array([np.sum(nij[:, 0]), np.sum(nij[:, 1]),
    np.sum(nij[:, 2]), np.sum(nij[:, 3]), np.sum(nij[:, 4])])
sum_nij = int(np.array([np.sum(sum_ni[:])]))

print("Nous avons les données suivantes :")
print(" nij:\n", nij)
print(" sum_ni:", sum_ni)
print(" sum_nj:", sum_nj)
print(" sum nij:", sum_nij, end="\n\n")
```

Listing 9 – Code Python pour n_i et n_j

```
Nous avons les données suivantes :
    nij:
    [[ 3  9 18 36 29]
    [ 3  3  1 14 13]]
    sum_ni: [95 34]
    sum_nj: [ 6 12 19 50 42]
    sum nij: 129
```

Listing 10 – Résultat du code

REPONSE		1	2	3	4	5	n_i
GENRE	Homme	3	9	18	36	29	95
	Femme	3	3	1	14	13	34
n_j		6	12	19	50	42	129

1. **Grandeur d'intérêt :** Le comportement masculin et féminin est indépendant/dépendant
2. **Hypothèse nulle,** $H0$ **:** Le comportement masculin et féminin est dépendant
3. **Hypothèse alternative,** $H1$ **:** Le comportement masculin et féminin est indépendant
4. **Niveau de confiance :** 95%
5. **Test statistique :** $\chi_0^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{(n_{ij} - \frac{n_i \cdot n_j}{n})^2}{\frac{n_i \cdot n_j}{n}}$ estimée par χ_{Obs}^2 à partir de l'échantillon
6. **Rejet de** $H0$ **si :**
   — Région critique : $\chi_{Obs}^2 > \chi_{(p-1)(q-1),\alpha}^2$
   — p-valeur : $p - valeur < 0.05$

7. **Calculs :**

— Formules utilisées :

$$\chi^2_{Obs} = \sum_{i=1}^{p}\sum_{j=1}^{q} \frac{(n_{ij} - \frac{n_i.n_j}{n})^2}{\frac{n_i.n_j}{n}} \quad (6)$$

$$\chi^2_{(p-1)(q-1),\alpha} = F^{-1}_{\chi^2_{(p-1)(q-1)}}(\alpha) \quad (7)$$

$$p\text{-}valeur = 1 - F_{\chi^2_{(p-1)(q-1)}}(\chi^2_{Obs}) \quad (8)$$

— Paramètres à prendre en compte :

$p \rightarrow$ Le nombre d'intervalles : 2 (Homme, Femme)

$k \rightarrow$ Le nombre de valeurs : 5 par intervalles

```python
p = 2
q = 5
ic = 95
alpha = 1 - ic / 100
chi2 = stats.chi2.ppf(1-alpha, ((p-1)*(q-1)))
chi2_Obs = 0

for i in range(p):
    for j in range(q):
        chi2_Obs += ((nij[i][j] - ((sum_ni[i]*sum_nj[j])/(sum_nij)))**2)/((
    sum_ni[i]*sum_nj[j])/(sum_nij))

p_valeur = 1 - stats.chi2.cdf(chi2_Obs, ((p-1)*(q-1)))

print("Question 4:")
print(" Chi2:", chi2)
print(" Chi2 Obs:", chi2_Obs)
print(" p-valeur:", p_valeur)
```

Listing 11 – Code Python question 4

```
Question 4:
    Chi2: 9.487729036781154
    Chi2 Obs: 6.621370399683419
    p-valeur: 0.1573019095765884
```

Listing 12 – Résultat du code

8. **Décision :**

Critéres de rejet de H0	
pour α fixé	avec *p-valeur*
$\chi^2_{Obs} > \chi^2_{(p-1)(q-1),\alpha}$	$p\text{-}valeur < 0.05$

$$\left.\begin{aligned} \chi^2_{Obs} &= 6.621 \\ \chi^2_{(p-1)(q-1),\alpha} &= 9.487 \\ p\text{-}valeur &= 0.15 \end{aligned}\right\} \quad \begin{aligned} \chi^2_{Obs} &< \chi^2_{(p-1)(q-1),\alpha} \\ p\text{-}valeur &> 0.05 \end{aligned}$$

Les résultats du test statistique montrent que *H0* ne peut être rejeté puisqu'aucune des conditions n'est validée.
Le comportement masculin et féminin est dépendant.

3 Code complet

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


# excercice 1
# question 1
# xi satellite
# ni obs
obs = np.array([6, 15, 9, 25, 17, 10, 8, 7, 3])
sat = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

nObs = 0
nSat = 0
for i in range(len(obs)):
    nObs += obs[i]
    nSat += sat[i]

num_mean = 0
for i in range(len(obs)):
    num_mean += obs[i]*sat[i]
mean = num_mean / nObs

num_var = 0
for i in range(len(obs)):
    num_var += (sat[i]-mean)**2
var = num_var/(len(sat)-1)

print("Question 1:")
print(" Moyenne =", mean)
print(' Variance =', var, end="\n\n")


#question 2
Lambda = 4.47
poisson_array = np.zeros((17,))
effectifs_th = np.zeros((17,))
print("Question 2:")
print("nb Sat |  nb Obs  |  Proba  |   eff th")
print("=========================================")
poisson_array[0] = stats.poisson.pmf(0, Lambda)
effectifs_th[0] = nObs * poisson_array[0]
print("  ", 0, "  |  ", "N.C", "  |\t", round(poisson_array[0], 3), "\t|\t", round(effectifs_th[0], 3))
print("-----------------------------------------")

for i in range(1, len(sat)+1):
    poisson_array[i] = stats.poisson.pmf(sat[i-1], Lambda)
    effectifs_th[i] = nObs * poisson_array[i]

    print("  ", sat[i-1], "  |   ", round(obs[i-1], 2), "\t |\t", round(poisson_array[i], 3), "\t|\t", round(effectifs_th[i], 3))

print("-----------------------------------------")

plt.plot(sat, poisson_array[:9], 'ro')
plt.show()

for i in range(10, 17):
    poisson_array[i] = stats.poisson.pmf(i, Lambda)
    effectifs_th[i] = nObs * poisson_array[i]

    print(" ", i, "  |  ", "N.C", "  |\t", round(poisson_array[i], 3), "\t|\t", round(effectifs_th[i], 3))
```

```python

print("\n\n")

nSat16 = np.arange(0, 17, 1)
plt.plot(nSat16, poisson_array, 'ro')
plt.show()


#question 3
#1. Paramètre d intérêt : La distribution du nombre de satellites visibles par rapport aux nombre de point d'observation
#2. Hypothèse nulle H0 : La distribution du nombre de satellites par rapport aux nombre de point d'observation  a été produit par cette loi de
#   Poisson
#3. Hypothèse alternative H1 : La distribution du nombre de satellites par rapport aux nombre de point d'observation
#   n'a pas été produit par cette loi de Poisson
#4. Tests statistique: page 201
#5. Niveau de confiance: 95%
#6. Rejet de H0 si ki2Obs > ki2,alpha,n-p-1 ou si p-valeur < 0.05

# On add les effectifs où npi < 5
i = 0
while (effectifs_th[i] < 5):
    effectifs_th[i+1] += effectifs_th[i]
    effectifs_th[i] = 0
    i += 1

i = len(effectifs_th)-1
while (effectifs_th[i] < 5):
    effectifs_th[i-1] += effectifs_th[i]
    effectifs_th[i] = 0
    i -= 1

effectifs_th = np.delete(effectifs_th, np.where(effectifs_th == 0))


k = len(effectifs_th)
p = 3 # la moyenne, l'effectif th et loi poisson
ic = 95
alpha = 1 - ic / 100
chi2 = stats.chi2.ppf(1-alpha, k-p-1)
chi2_Obs = 0

for i in range(k):
    chi2_Obs += ((obs[i]-effectifs_th[i])**2)/(effectifs_th[i])

p_valeur = 1 - stats.chi2.cdf(chi2_Obs, k-p-1)

print("Question 3:")
print(" Les effectifs théorique après modification dû aux n.pi<5 :", effectifs_th)
print(" Chi2:", chi2)
print(" Chi2 Obs:", chi2_Obs)
print(" p-valeur:", p_valeur)
print(" On ne peut pas rejeter H0, La distribution du nombre de satellites par rapport aux nombre de points d'observations à été produit par cette loi de poisson.", end="\n\n\n")


# On ne peut pas rejeter H0 donc La distribution du nombre de satellites par rapport aux nombre de point d'observation  a été produit par cette loi de
# Poisson

print("----------------------- EXERCICE 2 -----------------------\n\n")


```

```python
# exercice 2
# question 1
nij = np.array([[3, 9, 18, 36, 29],
                [3, 3, 1, 14, 13]])

sum_ni = np.array([np.sum(nij[0, :]), np.sum(nij[1, :])])
sum_nj = np.array([np.sum(nij[:, 0]), np.sum(nij[:, 1]),
                   np.sum(nij[:, 2]), np.sum(nij[:, 3]), np.sum(nij[:, 4])])
sum_nij = int(np.array([np.sum(sum_ni[:])]))

print("Nous avons les données suivantes :")
print(" nij:\n", nij)
print(" sum_ni:", sum_ni)
print(" sum_nj:", sum_nj)
print(" sum nij:", sum_nij, end="\n\n")

#1. Paramètre d intérêt : Le comportement masculins et féminins est indépendant/dépendant
#2. Hypothèse nulle H0 : Le comportement masculins et féminins est dépendant
#3. Hypothèse alternative H1 : Le comportement masculins et féminins est indépendant
#4. Tests statistique: page 207
#5. Niveau de confiance: 95%
#6. Rejet de H0 si ki2Obs > ki2,alpha,n-p-1 ou si p-valeur < 0.05

p = 2  #2 intervalles homme/femme
q = 5  #5 valeurs dans chaques intervalles
ic = 95
alpha = 1 - ic / 100
chi2 = stats.chi2.ppf(1-alpha, ((p-1)*(q-1)))
chi2_Obs = 0

for i in range(p):
    for j in range(q):
        chi2_Obs += ((nij[i][j] - ((sum_ni[i]*sum_nj[j])/(sum_nij)))**2)/((sum_ni[i]*sum_nj[j])/(sum_nij))

p_valeur = 1 - stats.chi2.cdf(chi2_Obs, ((p-1)*(q-1)))

print("Question 4:")
print(" Chi2:", chi2)
print(" Chi2 Obs:", chi2_Obs)
print(" p-valeur:", p_valeur)
print(" On ne peut pas rejeter H0, le comportement masculins et féminins est dépendant", end="\n\n\n")
```

Listing 13 – Code Python complet TP5