

# TP5: Tests du $\chi^2$ 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<sup>1</sup>, 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 :

$$\bar{x}_n = \frac{\sum (n_i \cdot x_i)}{n} \quad (1) \quad \left| \quad S_{n-1}^2 = \frac{1}{n-1} \sum_{k=1}^k n_k \cdot (x_k - \bar{x}_n)^2 \quad (2)$$

```
1 obs = np.array([6, 15, 9, 25, 17, 10, 8, 7, 3])
2 sat = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
3
4 nObs = 0
5 nSat = 0
6 for i in range(len(obs)):
7     nObs += obs[i]
8     nSat += sat[i]
9
10 num_mean = 0
11 for i in range(len(obs)):
12     num_mean += obs[i]*sat[i]
13 mean = num_mean / nObs
14
15 num_var = 0
16 for i in range(len(obs)):
17     num_var += (sat[i]-mean)**2
18 var = num_var/(len(sat)-1)
19
20 print("Question 1:")
21 print(" Moyenne =", mean)
22 print(' Variance =', var, end="\n\n")
```

Listing 1 – Code Python question 1

---

1. Éventuellement dans des environnements fortement métalliques ou partiellement enterrés

```

1 Question 1:
2 Moyenne = 4.47
3 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  $\lambda = 4.47$ . Déterminer 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

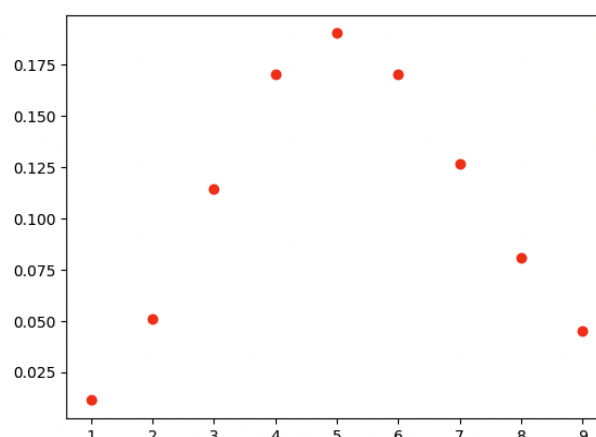


FIGURE 1 – Loi sous-jacente approcher par une loi de Poisson pour 1 à 9 satellites

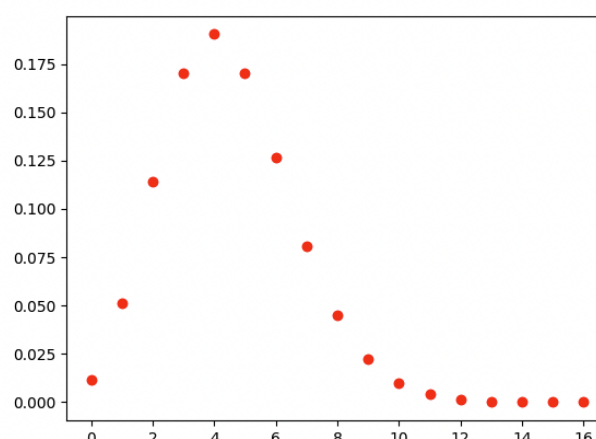


FIGURE 2 – Loi sous-jacente approcher par une loi de Poisson pour 0 à 16 satellites

```

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

```

```

6 print("=====")
7 poisson_array[0] = stats.poisson.pmf(0, Lambda)
8 effectifs_th[0] = nObs * poisson_array[0]
9 print(" ", 0, " | ", "N.C", " | \t", round(poisson_array[0], 3), "\t | \t", round(
    effectifs_th[0], 3))
10 print("-----")
11
12 for i in range(1, len(sat)+1):
13     poisson_array[i] = stats.poisson.pmf(sat[i-1], Lambda)
14     effectifs_th[i] = nObs * poisson_array[i]
15
16     print(" ", sat[i-1], " | ", round(obs[i-1], 2), "\t | \t", round(poisson_array
    [i], 3), "\t | \t", round(effectifs_th[i], 3))
17
18 print("-----")
19
20 plt.plot(sat, poisson_array[:9], 'ro')
21 plt.show()
22
23 for i in range(10, 17):
24     poisson_array[i] = stats.poisson.pmf(i, Lambda)
25     effectifs_th[i] = nObs * poisson_array[i]
26
27     print(" ", i, " | ", "N.C", " | \t", round(poisson_array[i], 3), "\t | \t", round(
    effectifs_th[i], 3))
28
29 print("\n\n")
30
31 nSat16 = np.arange(0, 17, 1)
32 plt.plot(nSat16, poisson_array, 'ro')
33 plt.show()

```

Listing 3 – Code Python question 2

```

1 Question 2:
2     nb Sat |  nb Obs |  Proba |  eff th
3     =====
4         0 |    N.C |    0.011 |    1.145
5     -----
6         1 |     6 |    0.051 |    5.117
7         2 |    15 |    0.114 |   11.436
8         3 |     9 |    0.17 |    17.04
9         4 |    25 |    0.19 |   19.042
10        5 |    17 |    0.17 |   17.024
11        6 |    10 |    0.127 |   12.683
12        7 |     8 |    0.081 |    8.099
13        8 |     7 |    0.045 |    4.525
14        9 |     3 |    0.022 |    2.248
15     -----
16       10 |    N.C |    0.01 |    1.005
17       11 |    N.C |    0.004 |    0.408
18       12 |    N.C |    0.002 |    0.152
19       13 |    N.C |    0.001 |    0.052
20       14 |    N.C |    0.0 |    0.017
21       15 |    N.C |    0.0 |    0.005
22       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 :

```
1 i = 0
2 while (effectifs_th[i] < 5):
3     effectifs_th[i+1] += effectifs_th[i]
4     effectifs_th[i] = 0
5     i += 1
6
7 i = len(effectifs_th)-1
8 while (effectifs_th[i] < 5):
9     effectifs_th[i-1] += effectifs_th[i]
10    effectifs_th[i] = 0
11    i -= 1
12
13 effectifs_th = np.delete(effectifs_th, np.where(effectifs_th == 0))
14 print("Les effectifs théorique après modification dû aux n.pi<5 :", effectifs_th)
```

Listing 5 –  $n.p_i < 5$

```
1 Les effectifs théorique après modification dû aux n.pi<5 :
2 [ 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 débiter 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,  $H_0$  :** 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,  $H_1$  :** 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  $\chi_{Obs}^2$  à partir de l'échantillon
6. **Rejet de  $H_0$  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

```

1 k = len(effectifs_th)
2 p = 3 # la moyenne, l'effectif th et loi poisson
3 ic = 95
4 alpha = 1 - ic / 100
5 chi2 = stats.chi2.ppf(1-alpha, k-p-1)
6 chi2_obs = 0
7
8 for i in range(k):
9     chi2_obs += ((obs[i]-effectifs_th[i])**2)/(effectifs_th[i])
10
11 p_valeur = 1 - stats.chi2.cdf(chi2_obs, k-p-1)
12
13 print("Question 3:")
14 print(" Chi2:", chi2)
15 print(" Chi2 Obs:", chi2_obs)
16 print(" p-valeur:", p_valeur)

```

Listing 7 – Code Python question 3

```

1 Question 3:
2   Chi2: 9.487729036781154
3   Chi2 Obs: 7.585020824300105
4   p-valeur: 0.10801814641379404

```

Listing 8 – Résultat du code

## 8. Décision :

Critères de rejet de $H_0$	
pour $\alpha$ fixé	avec $p$ -valeur
$\chi_{Obs}^2 > \chi_{k-p-1,\alpha}^2$	$p$ -valeur $< 0.05$

$$\left. \begin{array}{l} \chi_{Obs}^2 = 7.585 \\ \chi_{k-p-1,\alpha}^2 = 9.487 \\ p\text{-valeur} = 0.1 \end{array} \right\} \begin{array}{l} \chi_{Obs}^2 < \chi_{k-p-1,\alpha}^2 \\ p\text{-valeur} > 0.05 \end{array}$$

Les résultats du test statistique montrent que  $H_0$  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. Totalelement en désaccord
2. En désaccord
3. Sans commentaire
4. D'accord
5. Totalelement 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  $n_j$  :

```

1 nij = np.array([[3, 9, 18, 36, 29],
2 [3, 3, 1, 14, 13]])
3
4 sum_ni = np.array([np.sum(nij[0, :]), np.sum(nij[1, :])])
5 sum_nj = np.array([np.sum(nij[:, 0]), np.sum(nij[:, 1]),
6 np.sum(nij[:, 2]), np.sum(nij[:, 3]), np.sum(nij[:, 4])])
7 sum_nij = int(np.array([np.sum(sum_ni[:])]))
8
9 print("Nous avons les données suivantes :")
10 print(" nij:\n", nij)
11 print(" sum_ni:", sum_ni)
12 print(" sum_nj:", sum_nj)
13 print(" sum nij:", sum_nij, end="\n\n")

```

Listing 9 – Code Python pour  $n_i$  et  $n_j$

```

1 Nous avons les données suivantes :
2   nij:
3   [[ 3  9 18 36 29]
4    [ 3  3  1 14 13]]
5   sum_ni: [95 34]
6   sum_nj: [ 6 12 19 50 42]
7   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

- Grandeur d'intérêt :** Le comportement masculin et féminin est indépendant/dépendant
- Hypothèse nulle,  $H_0$  :** Le comportement masculin et féminin est dépendant
- Hypothèse alternative,  $H_1$  :** Le comportement masculin et féminin est indépendant
- Niveau de confiance :** 95%
- 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  $\chi_{Obs}^2$  à partir de l'échantillon
- Rejet de  $H_0$  si :**
  - Région critique :  $\chi_{Obs}^2 > \chi_{(p-1)(q-1), \alpha}^2$
  - p-valeur :  $p - valeur < 0.05$

## 7. Calculs :

— Formules utilisées :

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

$$\chi_{(p-1)(q-1),\alpha}^2 = F_{\chi_{(p-1)(q-1)}^2}^{-1}(\alpha) \quad (7)$$

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

— Paramètres à prendre en compte :

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

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

```

1 p = 2
2 q = 5
3 ic = 95
4 alpha = 1 - ic / 100
5 chi2 = stats.chi2.ppf(1-alpha, ((p-1)*(q-1)))
6 chi2_obs = 0
7
8 for i in range(p):
9     for j in range(q):
10         chi2_obs += ((nij[i][j] - ((sum_ni[i]*sum_nj[j])/(sum_nij))))**2/((
11             sum_ni[i]*sum_nj[j])/(sum_nij))
12 p_valeur = 1 - stats.chi2.cdf(chi2_obs, ((p-1)*(q-1)))
13
14 print("Question 4:")
15 print(" Chi2:", chi2)
16 print(" Chi2 Obs:", chi2_obs)
17 print(" p-valeur:", p_valeur)

```

Listing 11 – Code Python question 4

```

1 Question 4:
2   Chi2: 9.487729036781154
3   Chi2 Obs: 6.621370399683419
4   p-valeur: 0.1573019095765884

```

Listing 12 – Résultat du code

## 8. Décision :

Critères de rejet de $H_0$	
pour $\alpha$ fixé	avec $p\text{-valeur}$
$\chi_{Obs}^2 > \chi_{(p-1)(q-1),\alpha}^2$	$p\text{-valeur} < 0.05$

$$\left. \begin{array}{l} \chi_{Obs}^2 = 6.621 \\ \chi_{(p-1)(q-1),\alpha}^2 = 9.487 \\ p\text{-valeur} = 0.15 \end{array} \right\} \begin{array}{l} \chi_{Obs}^2 < \chi_{(p-1)(q-1),\alpha}^2 \\ p\text{-valeur} > 0.05 \end{array}$$

Les résultats du test statistique montrent que  $H_0$  ne peut être rejeté puisque aucune des conditions n'est validée.

Le comportement masculin et féminin est dépendant.

### 3 Code complet

```

1 import numpy as np
2 from scipy import stats
3 import matplotlib.pyplot as plt
4
5
6 # exercice 1
7 # question 1
8 # xi satellite
9 # ni obs
10 obs = np.array([6, 15, 9, 25, 17, 10, 8, 7, 3])
11 sat = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
12
13 nObs = 0
14 nSat = 0
15 for i in range(len(obs)):
16     nObs += obs[i]
17     nSat += sat[i]
18
19 num_mean = 0
20 for i in range(len(obs)):
21     num_mean += obs[i]*sat[i]
22 mean = num_mean / nObs
23
24 num_var = 0
25 for i in range(len(obs)):
26     num_var += (sat[i]-mean)**2
27 var = num_var/(len(sat)-1)
28
29 print("Question 1:")
30 print(" Moyenne =", mean)
31 print(' Variance =', var, end="\n\n")
32
33
34 #question 2
35 Lambda = 4.47
36 poisson_array = np.zeros((17,))
37 effectifs_th = np.zeros((17,))
38 print("Question 2:")
39 print("nb Sat | nb Obs | Proba | eff th")
40 print("=====")
41 poisson_array[0] = stats.poisson.pmf(0, Lambda)
42 effectifs_th[0] = nObs * poisson_array[0]
43 print(" ", 0, " | ", "N.C", " | \t", round(poisson_array[0], 3), "\t | \t", round(
    effectifs_th[0], 3))
44 print("-----")
45
46 for i in range(1, len(sat)+1):
47     poisson_array[i] = stats.poisson.pmf(sat[i-1], Lambda)
48     effectifs_th[i] = nObs * poisson_array[i]
49
50     print(" ", sat[i-1], " | ", round(obs[i-1], 2), "\t | \t", round(poisson_array
    [i], 3), "\t | \t", round(effectifs_th[i], 3))
51
52 print("-----")
53
54 plt.plot(sat, poisson_array[:9], 'ro')
55 plt.show()
56
57 for i in range(10, 17):
58     poisson_array[i] = stats.poisson.pmf(i, Lambda)
59     effectifs_th[i] = nObs * poisson_array[i]
60
61     print(" ", i, " | ", "N.C", " | \t", round(poisson_array[i], 3), "\t | \t", round(
    effectifs_th[i], 3))

```



```

62
63 print("\n\n")
64
65 nSat16 = np.arange(0, 17, 1)
66 plt.plot(nSat16, poisson_array, 'ro')
67 plt.show()
68
69
70 #question 3
71 #1. Paramètre d'intérêt : La distribution du nombre de satellites visibles par
    rapport au nombre de point d'observation
72 #2. Hypothèse nulle H0 : La distribution du nombre de satellites par rapport au
    nombre de point d'observation a été produite par cette loi de
73 # Poisson
74 #3. Hypothèse alternative H1 : La distribution du nombre de satellites par rapport
    au nombre de point d'observation
75 # n'a pas été produite par cette loi de Poisson
76 #4. Tests statistiques: page 201
77 #5. Niveau de confiance: 95%
78 #6. Rejet de H0 si  $\chi^2_{obs} > \chi^2_{1-\alpha, n-p-1}$  ou si p-valeur < 0.05
79
80 # On ajoute les effectifs où  $n_{pi} < 5$ 
81 i = 0
82 while (effectifs_th[i] < 5):
83     effectifs_th[i+1] += effectifs_th[i]
84     effectifs_th[i] = 0
85     i += 1
86
87 i = len(effectifs_th)-1
88 while (effectifs_th[i] < 5):
89     effectifs_th[i-1] += effectifs_th[i]
90     effectifs_th[i] = 0
91     i -= 1
92
93 effectifs_th = np.delete(effectifs_th, np.where(effectifs_th == 0))
94
95
96 k = len(effectifs_th)
97 p = 3 # la moyenne, l'effectif th et loi poisson
98 ic = 95
99 alpha = 1 - ic / 100
100 chi2 = stats.chi2.ppf(1-alpha, k-p-1)
101 chi2_obs = 0
102
103 for i in range(k):
104     chi2_obs += ((obs[i]-effectifs_th[i])**2)/(effectifs_th[i])
105
106 p_valeur = 1 - stats.chi2.cdf(chi2_obs, k-p-1)
107
108 print("Question 3:")
109 print(" Les effectifs théoriques après modification dû aux  $n_{pi} < 5$  :", effectifs_th)
110 print(" Chi2:", chi2)
111 print(" Chi2 Obs:", chi2_obs)
112 print(" p-valeur:", p_valeur)
113 print(" On ne peut pas rejeter H0, La distribution du nombre de satellites par
    rapport au nombre de points d'observations a été produite par cette loi de
    poisson.", end="\n\n\n")
114
115
116 # On ne peut pas rejeter H0 donc La distribution du nombre de satellites par rapport
    au nombre de point d'observation a été produite par cette loi de
117 # Poisson
118
119 print("----- EXERCICE 2 ----- \n\n")
120
121

```

```

122 # exercice 2
123 # question 1
124 nij = np.array([[3, 9, 18, 36, 29],
125                [3, 3, 1, 14, 13]])
126
127 sum_ni = np.array([np.sum(nij[0, :]), np.sum(nij[1, :])])
128 sum_nj = np.array([np.sum(nij[:, 0]), np.sum(nij[:, 1]),
129                  np.sum(nij[:, 2]), np.sum(nij[:, 3]), np.sum(nij[:, 4])])
130 sum_nij = int(np.array([np.sum(sum_ni[:])]))
131
132 print("Nous avons les données suivantes :")
133 print(" nij:\n", nij)
134 print(" sum_ni:", sum_ni)
135 print(" sum_nj:", sum_nj)
136 print(" sum nij:", sum_nij, end="\n\n")
137
138 #1. Paramètre d'intérêt : Le comportement masculins et féminins est indépendant/dé
    pendant
139 #2. Hypothèse nulle H0 : Le comportement masculins et féminins est dépendant
140 #3. Hypothèse alternative H1 : Le comportement masculins et féminins est indépendant
141 #4. Tests statistique: page 207
142 #5. Niveau de confiance: 95%
143 #6. Rejet de H0 si  $\chi^2_{obs} > \chi^2_{1-\alpha, n-p-1}$  ou si p-valeur < 0.05
144
145 p = 2 #2 intervalles homme/femme
146 q = 5 #5 valeurs dans chaque intervalles
147 ic = 95
148 alpha = 1 - ic / 100
149 chi2 = stats.chi2.ppf(1-alpha, ((p-1)*(q-1)))
150 chi2_obs = 0
151
152 for i in range(p):
153     for j in range(q):
154         chi2_obs += ((nij[i][j] - ((sum_ni[i]*sum_nj[j])/(sum_nij))))**2)/((sum_ni[i]*
            sum_nj[j])/(sum_nij))
155
156 p_valeur = 1 - stats.chi2.cdf(chi2_obs, ((p-1)*(q-1)))
157
158 print("Question 4:")
159 print(" Chi2:", chi2)
160 print(" Chi2 Obs:", chi2_obs)
161 print(" p-valeur:", p_valeur)
162 print(" On ne peut pas rejeter H0, le comportement masculins et féminins est dé
    pendant", end="\n\n\n")

```

Listing 13 – Code Python complet TP5