TP2: Statistiques descriptives bivariées

Tanguy ROUDAUT — Tadios QUINIO FIPASE 24

13 Septembre 2022

1 Statistiques descriptives bivariées sur des données d'Iris

1.1 Étude de la largeur du pétale en fonction de la longueur du pétale

Représentation graphique

Question 1 : Tracer le nuage de points de la longueur du pétale en fonction de la largeur du pétale pour les 150 iris contenus dans les données ((numpy.)plot, (numpy.)scatter). Ne pas oublier de mettre des titres sur les axes. Décrire le nuage de points.

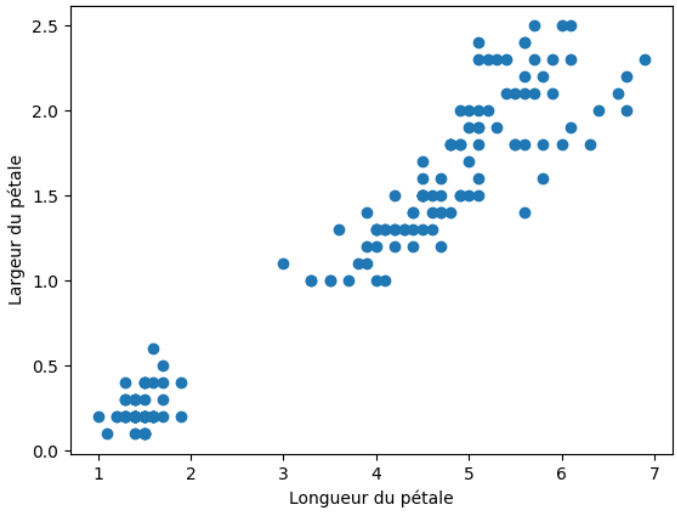


FIGURE 1 – Nuage de points de la longueur en fonction de la largeur du pétale des 150 Iris

Dans un premier temps, on remarque deux concentrations principale.
La première a une longueur de pétale qui varie entre 1 cm et 2 cm pour une largeur d'environ 0,2 cm et 0,6 cm. La seconde, une longueur variant de 3 cm à 7 cm avec une largeur de 1 cm à 2,5 cm.

On constate également que la longueur de pétale varie proportionnellement à la largeur.

```
plt.scatter(petallength, petalwidth)
plt.ylabel("Largeur du pétale")
plt.xlabel("Longueur du pétale")
plt.title("Nuage de points de la longueur en fonction de la largeur du pétale")
plt.show()
```

Listing 1 – Code Python pour calculer le coefficient de corrélation

Question 2 : Rappeler la définition du coefficient de corrélation et le calculer par la fonction ((numpy.)corrcoef)

Le coefficient de corrélation est une valeur qui permet de déterminer s'il existe une relation linéaire entre deux variables, ce qui veut dire que si ces variables sont corrélées alors elles sont liées.

```
corr_coef_matrix = np.corrcoef(petallength, petalwidth)
print("Matrice de corrélation:\n",corr_coef_matrix)
```

Listing 2 – Code Python pour calculer le coefficient de corrélation

```
Matrice de corrélation:
[[1.        0.9627571]
[0.9627571 1.       ]]
```

Listing 3 – Résultat du code

Question 3 : Donner l'équation de la droite de régression linéaire, créer une fonction permettant de la calculer à partir de deux variables X et Y et tracer la sur le même graphique

1. **Résultat et analyse :**

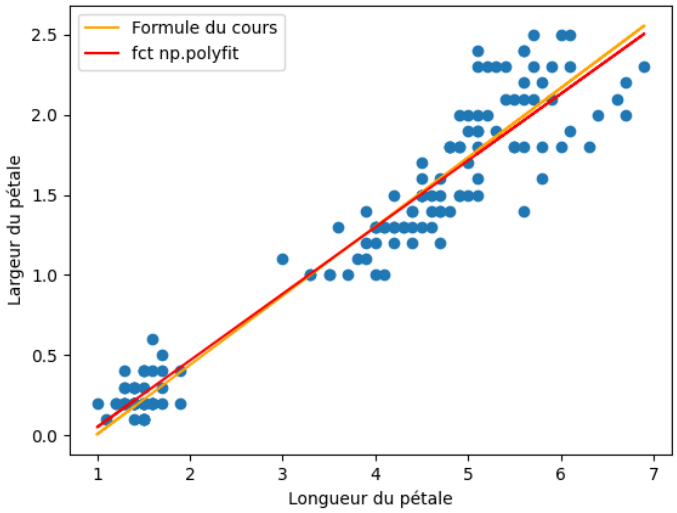


FIGURE 2 – Équation de la droite de régression linéaire sur le nuage de points

2. **Formules utilisées :**

$$\hat{a} = \rho_{X,Y} \cdot \frac{S_{n-1,Y}}{S_{n-1,X}} \qquad \text{et} \qquad \hat{b} = \overline{Y_n} \cdot \rho_{X,Y} \cdot \frac{S_{n-1,Y}}{S_{n-1,X}} . \overline{X_n} \tag{1}$$

3. **Code python :**

```
# Méthode par la formule du cours
def regression_lineaire(corr_coef, x, y):
    mean_x = np.mean(x)
    var_x = np.var(x)
    e_type_x = np.sqrt(var_x)

    mean_y = np.mean(y)
    var_y = np.var(y)
    e_type_y = np.sqrt(var_y)

    a = corr_coef * (e_type_y/e_type_x)
```

```
    b = mean_y - corr_coef * ((e_type_y/e_type_x)) * mean_x

    return a, b


a, b = regression_lineaire(corr_coef, petallength, petalwidth)

# Méthode avec polyfit de numpy
fit = np.polyfit(petallength, petalwidth, 1)
poly = petallength*fit[0]+fit[1]

plt.plot(petallength, a*petallength+b, color='orange', label='Formule du cours')
plt.plot(petallength, poly, color='red', label='fct np.polyfit')
plt.scatter(petallength, petalwidth)
plt.ylabel("Largeur du pétale")
plt.xlabel("Longueur du pétale")
plt.title("Equation de la droite de régression linéaire")
plt.legend()
plt.show()

```

Listing 4 – Code Python pour tracer la droite de régression linéaire

Question 4 : Analyser le lien entre les deux variables

On constate que les variables *petalLenght* et *petalWidth* sont liées, elles évoluent proportionnellement l'une avec l'autre.
On peut donc soumettre l'hypothèse que ces variables sont corrélées puisqu'il existe deux réels tels que $Y = aX + b$.

1.2 Étude de la longueur de pétale selon les différentes espèces

Représentations graphiques

Question 5 : Représenter sur une même figure, les trois histogrammes de la longueur de pétale : un histogramme pour chaque espèce avec une couleur. Commenter cette figure.

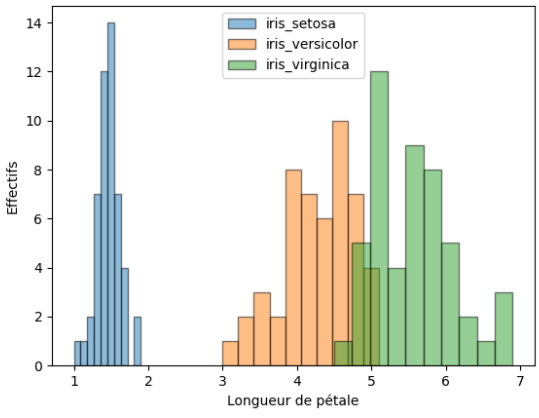


FIGURE 3 – Histogrammes de la longueur de pétale pour chaque espèce

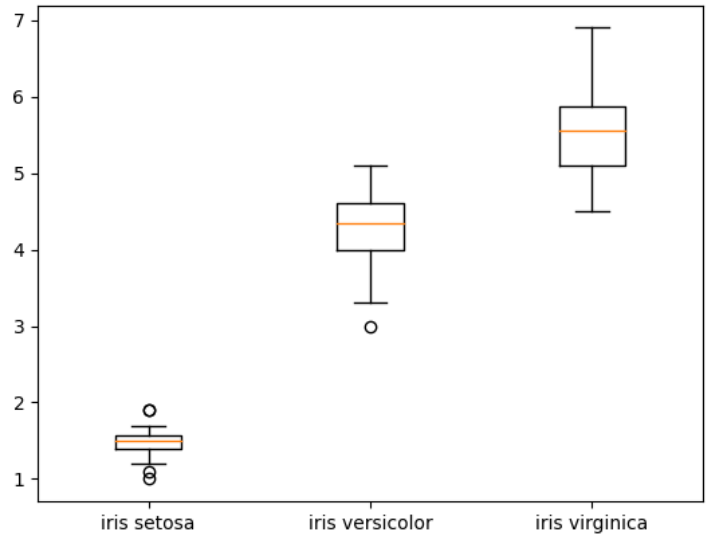
Lors du *TP1 question 7*, nous avions pu remarquer que l'une des espèces a une taille de pétale plus faible, mais qui varie également moins que les deux autres. Le problème était que c'était seulement une hypothèse, puisque nous n'avions pas cette répartition entre les différentes espèces.
L'histogramme de la figure 3, reprend le même schéma que celui de la *question 7 du TP1*, mais cette fois si en prenant en compte les différentes espèces.
On peut donc conclure que notre hypothèse est vraie, l'iris setosa est plus petite que les deux autres espèces.

```
iris_setosa_petallength = []
iris_versicolor_petallength = []
iris_virginica_petallength = []

```

```python
for i in range(len(species)):
    if species[i] == 'Iris-setosa':
        iris_setosa_petallength.append(petallength[i])
    elif species[i] == 'Iris-versicolor':
        iris_versicolor_petallength.append(petallength[i])
    elif species[i] == 'Iris-virginica':
        iris_virginica_petallength.append(petallength[i])

plt.hist(iris_setosa_petallength, edgecolor='black', alpha=0.5, label='iris_setosa')
plt.hist(iris_versicolor_petallength, edgecolor='black', alpha=0.5, label='iris_versicolor')
plt.hist(iris_virginica_petallength, edgecolor='black', alpha=0.5, label='iris_virginica')
plt.xlabel('Longueur de pétale')
plt.ylabel('Effectifs')
plt.title("Histogrammes de la longueur de pétale pour chaque espèce")
plt.legend()
plt.show()
```

Listing 5 – Code Python pour calculer le coefficient de corrélation

Question 6 : Représenter sur une même figure. une boite à moustache par espèce. Commenter cette figure.



On remarque encore une fois que l'iris setosa à un intervalle de longueur de pétales plus faible que les deux autres espèces.

FIGURE 4 – Boite à moustache de la longueur de pétale pour chaque espèce

```python
boxplot_petallength = [iris_setosa_petallength, iris_versicolor_petallength, iris_virginica_petallength]
plt.boxplot(boxplot_petallength, labels=['iris setosa', 'iris versicolor', 'iris virginica'])
plt.title("Boite à moustache de la longueur de pétale pour chaque espèce")
plt.show()
```

Listing 6 – Code Python pour calculer le coefficient de corrélation

Représentations graphiques

Question 7 : Calculer le rapport de corrélation lié à la décomposition de la variance en variance intraclasse et interclasse. Qu'en concluez-vous ?

1. **Formules utilisées :**

   – Variance interclasse :

   $$S_B^2 = \frac{1}{n}\sum_{i=1}^{p} n_i(\overline{Y_i} - \overline{Y})^2 \qquad (2)$$

   – Variance intraclasse :

   $$S_W^2 = \frac{1}{n}\sum_{i=1}^{p} n_i.S_{n-1,Y_i}^2 \qquad (3)$$

   – Variance totale :

   $$S_{n-1,Y}^2 = S_B^2 + S_W^2 \qquad (4)$$

   – Rapport de corrélation :

   $$S_{\frac{Y}{X}} = \sqrt{\frac{S_B^2}{S_{n-1,Y}^2}} \in [0,1] \qquad (5)$$

2. **Valeurs obtenues :**

Variance interclasse	2.910
Variance intraclasse	0.181
Variance total	3.092
Rapport de corrélation	0.970

3. **Conclusion :**

   La variance totale trouvée grâce à la formule 4, correspond avec celle trouvée avec *numpy*. Cette constatation nous permet de nous assurer que nos résultats sont corrects et de trouver par la suite le rapport de corrélation qui est de 0,970.
   Le rapport de corrélation étant proche de 1 nous permet de confirmer notre hypothèse établie à la *question 4* concernant la corrélation des variables *petalLength* et *petalWidth*.

4. **Code python :**

```
n_cumul = len(species)

mean_species_petallength = [np.mean(iris_setosa_petallength), np.mean(
    iris_versicolor_petallength), np.mean(iris_virginica_petallength)]

mean_petallength = np.mean(petallength)

var_species_petallength = [np.var(iris_setosa_petallength), np.var(
    iris_versicolor_petallength), np.var(iris_virginica_petallength)]

n_species = [len(iris_setosa_petallength), len(iris_versicolor_petallength),
    len(iris_virginica_petallength)]



def var_inter_classe_petallength(n_cumul, n_species, mean_species, mean):
    ret = 0
    for i in range(len(speciesname)):
        ret += n_species[i] * (mean_species[i] - mean)**2

    return ret/n_cumul

```

```python

def var_intra_classe_petallength(n_cumul, n_species, var_species):
    ret = 0
    for i in range(len(speciesname)):
        ret += n_species[i] * var_species[i]

    return ret / n_cumul



var_inter_classe = var_inter_classe_petallength(n_cumul, n_species,
    mean_species_petallength, mean_petallength)

var_intra_classe = var_intra_classe_petallength(n_cumul, n_species,
    var_species_petallength)

var_totale_formule = var_inter_classe + var_intra_classe

var_totale = np.var(petallength)

rapport_corr = np.sqrt(var_inter_classe/var_totale_formule)

print("Variance interclasse: ", var_inter_classe)
print("Variance intraclasse: ", var_intra_classe)
print("Variance total trouvé avec la formule: ", var_totale_formule)
print("Variance total trouvé avec numpy: ", var_totale)
print("Rapport de corrélation: ", rapport_corr)

```

Listing 7 – Code Python pour calculer le coefficient de corrélation

```
Variance interclasse:  2.910958222222223
Variance intraclasse:  0.1814666666666667
Variance total trouvé avec la formule:  3.0924248888888894
Variance total trouvé avec numpy:  3.092424888888889
Rapport de corrélation:  0.9702159417163924

```

Listing 8 – Résultat du code

1.3 Code complet

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('iris.csv', sep=',')

sepallength = df["sepallength"].values
sepalwidth = df["sepalwidth"].values
petallength = df["petallength"].values
petalwidth = df["petalwidth"].values
species = df["class"].values
speciesname = np.unique(species)
variablename = df.keys().values[0:-1]


# question 1
plt.scatter(petallength, petalwidth)
plt.ylabel("Largeur du pétale")
plt.xlabel("Longueur du pétale")
plt.title("Nuage de points de la longueur en fonction de la largeur du pétale")
plt.show()


# question 2
corr_coef_matrix = np.corrcoef(petallength, petalwidth)
print(corr_coef_matrix)

corr_coef = corr_coef_matrix[1][1]


# question 3
def regression_lineaire(corr_coef, x, y):
    mean_x = np.mean(x)
    var_x = np.var(x)
    e_type_x = np.sqrt(var_x)

    mean_y = np.mean(y)
    var_y = np.var(y)
    e_type_y = np.sqrt(var_y)

    a = corr_coef * (e_type_y/e_type_x)
    b = mean_y - corr_coef * ((e_type_y/e_type_x)) * mean_x

    return a, b


a, b = regression_lineaire(corr_coef, petallength, petalwidth)

fit = np.polyfit(petallength, petalwidth, 1)
poly = petallength*fit[0]+fit[1]

plt.plot(petallength, a*petallength+b, color='orange', label='Formule du cours')
plt.plot(petallength, poly, color='red', label='fct np.polyfit')
plt.scatter(petallength, petalwidth)
plt.ylabel("Largeur du pétale")
plt.xlabel("Longueur du pétale")
plt.title("Equation de la droite de régression linéaire")
plt.legend()
plt.show()


#question 5
iris_setosa_petallength = []
iris_versicolor_petallength = []
```

```
iris_virginica_petallength = []

for i in range(len(species)):
    if species[i] == 'Iris-setosa':
        iris_setosa_petallength.append(petallength[i])
    elif species[i] == 'Iris-versicolor':
        iris_versicolor_petallength.append(petallength[i])
    elif species[i] == 'Iris-virginica':
        iris_virginica_petallength.append(petallength[i])

plt.hist(iris_setosa_petallength, edgecolor='black', alpha=0.5, label='iris_setosa')
plt.hist(iris_versicolor_petallength, edgecolor='black', alpha=0.5, label='
    iris_versicolor')
plt.hist(iris_virginica_petallength, edgecolor='black', alpha=0.5, label='
    iris_virginica')
plt.xlabel('Longueur de pétale')
plt.ylabel('Effectifs')
plt.title("Histogrammes de la longueur de pétale pour chaque espèce")
plt.legend()
plt.show()


# question 6
boxplot_petallength = [iris_setosa_petallength, iris_versicolor_petallength,
    iris_virginica_petallength]
plt.boxplot(boxplot_petallength, labels=['iris setosa', 'iris versicolor', 'iris
    virginica'])
plt.title("Boite à moustache de la longueur de pétale pour chaque espèce")
plt.show()


#question 7
n_cumul = len(species)

mean_species_petallength = [np.mean(iris_setosa_petallength), np.mean(
    iris_versicolor_petallength), np.mean(iris_virginica_petallength)]

mean_petallength = np.mean(petallength)

var_species_petallength = [np.var(iris_setosa_petallength), np.var(
    iris_versicolor_petallength), np.var(iris_virginica_petallength)]

n_species = [len(iris_setosa_petallength), len(iris_versicolor_petallength), len(
    iris_virginica_petallength)]



def var_inter_classe_petallength(n_cumul, n_species, mean_species, mean):
    ret = 0
    for i in range(len(speciesname)):
        ret += n_species[i] * (mean_species[i] - mean)**2

    return ret/n_cumul



def var_intra_classe_petallength(n_cumul, n_species, var_species):
    ret = 0
    for i in range(len(speciesname)):
        ret += n_species[i] * var_species[i]

    return ret / n_cumul



```

```
123 var_inter_classe = var_inter_classe_petallength(n_cumul, n_species,
        mean_species_petallength, mean_petallength)
124
125 var_intra_classe = var_intra_classe_petallength(n_cumul, n_species,
        var_species_petallength)
126
127 var_totale_formule = var_inter_classe + var_intra_classe
128
129 var_totale = np.var(petallength)
130
131 rapport_corr = np.sqrt(var_inter_classe/var_totale_formule)
132
133 print("Variance interclasse: ", var_inter_classe)
134 print("Variance intraclasse: ", var_intra_classe)
135 print("Variance total trouvé avec la formule: ", var_totale_formule)
136 print("Variance total trouvé avec numpy: ", var_totale)
137 print("Rapport de corrélation: ", rapport_corr)
```

Listing 9 – Code Python complet TP2