

NOM :

GROUPE :



R2.08 - Statistique descriptive
Contrôle Terminal - Sujet A
 2021/2022 - A. Ridard



Modalités d'évaluation

- Pour répondre aux questions, vous devez utiliser l'annexe
- Les questions possèdent une et une seule bonne réponse
- Une bonne réponse rapporte 1 point et une mauvaise retire 0.5 point à la note qui ne pourra être négative

Indiquer le code permettant d'obtenir le dataframe...

1.

option	effectif
aucune	15
conservatoire	3
foot	3
rugby	2
tennis	4

- ☐ code 1
- ☐ code 2
- ☐ code 3
- ☐ code 4

2.

clNoteMusique	idEleve
(4.985, 7.5]	1
(7.5, 10.0]	1
(10.0, 12.5]	2
(12.5, 15.0]	12
(15.0, 17.5]	2
(17.5, 20.0]	9

- ☐ code 5
- ☐ code 6
- ☐ code 7
- ☐ code 8

3.

sexe	lv2	idEleve
F	allemand	2
	chinois	5
	espagnol	8
M	allemand	7
	chinois	2
	espagnol	3

- ☐ code 9
- ☐ code 10
- ☐ code 11
- ☐ code 12

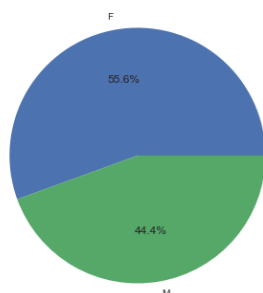
4.

sexe	lv2	allemand	chinois	espagnol
F		2	5	8
M		7	2	3

- ☐ code 13
- ☐ code 14
- ☐ code 15
- ☐ code 16

Indiquer le code permettant d'obtenir le graphique...

5.



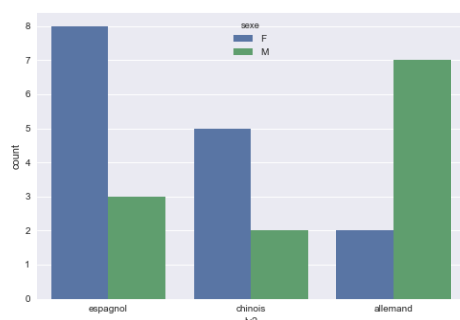
☐ code 17

☐ code 18

☐ code 19

☐ code 20

6.



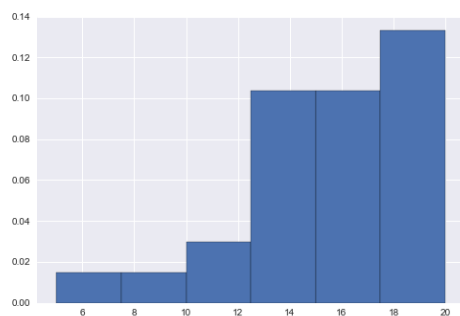
☐ code 21

☐ code 22

☐ code 23

☐ code 24

7.



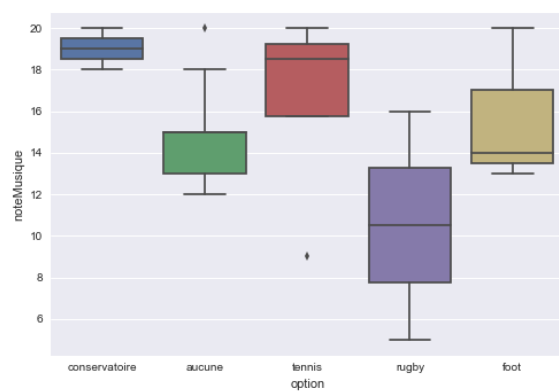
☐ code 25

☐ code 26

☐ code 27

☐ code 28

8.



☐ code 29

☐ code 30

☐ code 31

☐ code 32

Indiquer le code permettant d'obtenir le résumé numérique...

mean

std

option

aucune	14.666667	2.160247
conservatoire	19.000000	1.000000
foot	15.666667	3.785939
rugby	10.500000	7.778175
tennis	16.500000	5.066228

9.

☐ code 33

☐ code 34

☐ code 35

☐ aucun code ne convient

noteEps

count

mean

std

min

25%

50%

75%

max

sexe

F	15.0	14.566667	3.245143	8.0	13.0	15.0	17.0	19.0
M	12.0	15.416667	5.212892	5.0	11.0	18.0	19.0	20.0

10.

☐ code 36

☐ code 37

☐ code 38

☐ aucun code ne convient

Indiquer le graphique le plus adapté pour représenter la distribution...	
11. d'une variable qualitative ordinale	<input type="checkbox"/> diagramme circulaire <input type="checkbox"/> histogramme <input type="checkbox"/> diagramme en barres <input type="checkbox"/> boîte à moustaches
12. d'une variable quantitative discrète	<input type="checkbox"/> diagramme circulaire <input type="checkbox"/> histogramme <input type="checkbox"/> diagramme en barres <input type="checkbox"/> boîte à moustaches
13. d'une variable quantitative continue	<input type="checkbox"/> diagramme circulaire <input type="checkbox"/> histogramme <input type="checkbox"/> diagramme en barres <input type="checkbox"/> boîte à moustaches

Indiquer ce que permet d'afficher...	
14. le code 39	<input type="checkbox"/> une moyenne <input type="checkbox"/> une variance <input type="checkbox"/> une médiane <input type="checkbox"/> un premier quartile <input type="checkbox"/> un troisième quartile
15. le code 40	<input type="checkbox"/> une moyenne <input type="checkbox"/> une variance <input type="checkbox"/> une covariance <input type="checkbox"/> un rapport de corrélation
16. le code 41	<input type="checkbox"/> une moyenne <input type="checkbox"/> une variance <input type="checkbox"/> une covariance <input type="checkbox"/> un rapport de corrélation

Exercice.

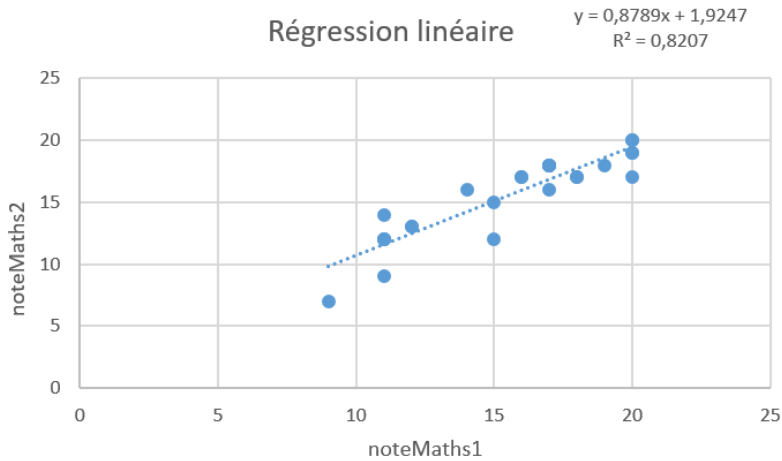
1. Voici une partie du résultat obtenu en réalisant, avec Excel, l'analyse de la variance de **noteMaths1** par rapport à **lv2**.

Groupes	Nombre d'échantillons	Somme	Moyenne	Variance
allemand	9	125	13,88888889	14,86111111
chinois	7	103	14,71428571	13,23809524
espagnol	11	193	17,54545455	3,872727273

Quels sont les calculs à effectuer pour vérifier la décomposition suivante :

Variance de Y = Variance **inter-catégories** + Variance **intra-catégories**

2. Voici le résultat obtenu en réalisant, avec Excel, la régression linéaire de **noteMaths2** en fonction de **noteMaths1**.



Dans quelle mesure peut-on dire que la note de maths 2 dépend linéairement de celle de maths 1 ?

Annexe

Importations

```
# importation des packages
import numpy as np
import pandas as pd
import seaborn as sns

# importation des modules
import matplotlib.pyplot as plt
import scipy.stats as sps

# importation du fichier Excel
data = pd.read_excel("QCM_Data.xlsx", sheetname = "data")
```

Contenu de "data" :

	idEleve	sexe	lv2	option	noteMaths1	noteMaths2	noteMusique	noteEps
0	1	F	espagnol	conservatoire	20	19	20	18.0
1	2	F	espagnol	aucune	18	17	20	15.0
2	3	F	chinois	aucune	14	16	18	16.0
3	4	M	allemand	tennis	17	18	20	18.0
4	5	M	chinois	rugby	20	17	16	19.0
5	6	M	espagnol	foot	15	15	14	19.0
6	7	F	espagnol	aucune	16	17	15	13.0
7	8	F	chinois	aucune	18	17	15	13.0
8	9	M	allemand	rugby	9	7	5	5.0
9	10	M	espagnol	tennis	16	17	18	19.0
10	11	F	allemand	conservatoire	20	19	19	15.0
11	12	F	espagnol	conservatoire	20	20	18	18.5
12	13	F	espagnol	tennis	20	20	19	19.0
13	14	M	allemand	foot	11	12	13	17.0
14	15	M	espagnol	foot	19	18	20	20.0
15	16	M	allemand	aucune	17	18	16	20.0
16	17	M	chinois	aucune	12	13	15	11.0
17	18	F	chinois	aucune	17	18	15	14.0
18	19	M	allemand	aucune	11	12	14	11.0
19	20	F	chinois	aucune	11	14	12	12.0
20	21	F	espagnol	aucune	15	12	13	15.0
21	22	M	allemand	aucune	17	16	14	18.0
22	23	F	chinois	aucune	11	9	12	8.0
23	24	F	espagnol	aucune	17	18	15	18.0
24	25	M	allemand	tennis	12	13	9	8.0
25	26	F	espagnol	aucune	17	18	13	9.0
26	27	F	allemand	aucune	11	12	13	15.0



Codes pour les questions 1 à 4

```
#####

# code 1
data.groupby("option")["idEleve"].count().rename(columns={"idEleve": "effectif"})

# code 2
data.groupby("option")["effectif"].count()

# code 3
data.groupby("option")[["idEleve"]].count().rename(columns={"idEleve": "effectif"})

# code 4
data.groupby("option")[["effectif"]].count()

#####

# code 5
data["clNoteMusique"] = pd.cut(data["noteMusique"], bins=6)
data.groupby("clNoteMusique")[["idEleve"]].count()

# code 6
data["clNoteMusique"] = pd.cut(data["noteMusique"], bins=6)
data.groupby("clNoteMusique")[["idEleve"]].sum()

# code 7
data["clNoteMusique"] = pd.cut(data["noteMusique"], bins=2.5)
data.groupby("clNoteMusique")[["idEleve"]].sum()

# code 8
data["clNoteMusique"] = pd.cut(data["noteMusique"], bins=2.5)
data.groupby("clNoteMusique")[["idEleve"]].count()

#####

# code 9
pd.crosstab(data["sexe"], data["lv2"])

# code 10
data.groupby(["lv2", "sexe"])[["idEleve"]].count()

# code 11
data.groupby(["sexe", "lv2"])[["idEleve"]].count()

# code 12
pd.crosstab(data["lv2"], data["sexe"])

#####

# code 13
pd.crosstab(data["sexe"], data["lv2"], normalize="index")

# code 14
pd.crosstab(data["sexe"], data["lv2"], normalize="columns")

# code 15
pd.crosstab(data["sexe"], data["lv2"], normalize="all")

# code 16
pd.crosstab(data["sexe"], data["lv2"], normalize=False)
```



Codes pour les questions 5 à 7

```
#####

# code 17
dist = data.groupby("sexe")["idEleve"].count()
plt.pie(dist["idEleve"], labels=dist.index, autopct='%1.1f%%')
plt.show()

# code 18
dist = data.groupby("sexe")["idEleve"].count()
plt.pie(dist["sexe"], labels=dist.index, autopct='%1.1f%%')
plt.show()

# code 19
dist = data.groupby("lv2")["idEleve"].count()
plt.pie(dist["idEleve"], labels=["F", "M"], autopct='%1.1f%%')
plt.show()

# code 20
dist = data.groupby("lv2")["idEleve"].count()
plt.pie(dist["lv2"], labels=["F", "M"], autopct='%1.1f%%')
plt.show()

#####

# code 21
dist = data.groupby("sexe")["id_eleve"].count()
plt.bar(dist["id_eleve"], labels=dist.index, autopct='%1.1f%%')
plt.show()

# code 22
sns.countplot(x="lv2", hue="sexe", data=data)
plt.show()

# code 23
dist = data.groupby(["sexe", "lv2"])["idEleve"].count()
x = range(len(dist))
plt.bar(x, dist["idEleve"])
plt.xticks(dist.index)
plt.show()

# code 24
sns.countplot(x="sexe", hue="lv2", data=data)
plt.show()

#####

# code 25
plt.hist(data["noteMusique"], bins=6, density=False, edgecolor="black")
plt.show()

# code 26
dist = data.groupby(["noteMusique"])["idEleve"].count()
x = range(len(dist))
plt.bar(x, dist["idEleve"], edgecolor="black")
plt.show()

# code 27
plt.hist(data["noteMusique"], bins=6, density=True, edgecolor="black")
plt.show()

# code 28
x = range(len(data["noteMusique"]))
plt.bar(x, data["noteMusique"], edgecolor="black")
plt.show()
```



Codes pour les questions 8 à 10 et 14 à 16

```
#####

# code 29
plt.boxplot(data["option", "note_musique"])
plt.show()

# code 30
plt.boxplot(data["option"], data["note_musique"])
plt.show()

# code 31
sns.boxplot(data["option", "noteMusique"])
plt.show()

# code 32
sns.boxplot(data["option"], data["noteMusique"])
plt.show()

#####

# code 33
data.groupby("option")["noteMusique"].agg(["mean", "std"])

# code 34
data.groupby("noteMusique")["option"].agg(["mean", "std"])

# code 35
data.groupby("noteMusique")["option"].agg([mean, std])

#####

# code 36
data.groupby("noteEps")[[ "sexe" ]].describe()

# code 37
data.groupby("sexe")[[ "noteEps" ]].describe()

# code 38
data["noteEps"].groupby("sexe").describe()

#####

# code 39
np.percentile(data["noteMusique"],50)

# code 40
sum(np.array(data["noteMusique"])/len(data)

# code 41
sum((np.array(data["noteMusique"])-data["noteMusique"].mean())**2)/len(data)
```