






L2 Biostatistiques-Bioinformatique

Exploration graphique d'un jeu de données

A. MARY & J.R. LOBRY

L'objectif de cette séance est de vous faire utiliser le logiciel  pour procéder à l'exploration préliminaire d'un jeu de données. Le but sera essentiellement d'apprendre à utiliser les commandes graphiques pour visualiser des données. Seules les parties faisant l'objet d'un exercice sont susceptibles d'être évaluées à l'examen.




Contents

1	Lancement de  et calculs élémentaires	2
1.1	Lancement de  en salle de TP	2
1.2	Calculs élémentaires	2
2	Importation de données dans 	4
2.1	Les données sur les notes des étudiants	4
2.2	Les données sur la masse des bébés	9
3	Extraction des données utiles	9
3.1	Extraction de colonnes (variables : après la virgule)	9
3.2	Extraction de lignes (individus : avant la virgule)	12
3.3	Extraction simultanée de lignes et de colonnes : les variables que je veux sur les individus que je veux	13
4	La calculatrice 	14
4.1	Calculatrice numérique vectorielle	14
4.2	Calculatrice programmable	15
5	Visualisation d'une variable à la fois	16
5.1	Cas d'une variable qualitative	16
5.1.1	Les diagrammes en bâtons	16
5.1.2	Pour aller plus loin (hors programme)	17
5.2	Cas d'une variable quantitative	18
5.2.1	Histogramme	18
5.2.2	Boîte à moustaches	20
5.2.3	Pour aller plus loin (hors programme)	21

6	Visualisation de deux variables à la fois	25
6.1	Deux variables quantitatives	25
6.2	Deux variables quantitatives (hors programme)	26
6.3	Quantitatif-Qualitatif	28
6.4	Quantitatif-Qualitatif (hors programme)	30
6.5	Deux variables qualitatives	31
7	Visualisation de trois variables à la fois	33
7.1	Quanti-Quanti-Quali	33
7.2	Quanti-Quanti-Quali (hors programme)	34
7.3	Quanti-Quali-Quali	36
8	Sauvegarde des données	37

1 Lancement de et calculs élémentaires


1.1 Lancement de en salle de TP

IL existe plusieurs interfaces permettant de jouer avec le logiciel . Pour les débutants, nous conseillons d'utiliser `RStudio` qui est disponible en salle de TP. Le logiciel  fonctionne comme une calculatrice à laquelle on donne des ordres. Par exemple, la commande `print(pi)` donne l'ordre d'afficher la valeur approximative de π . Dans ce document, toutes les commandes  sont données en rouge, il est inutile de perdre votre temps à les recopier : vous pouvez simplement les copier/coller à partir du PDF dont l'URL est donnée en pied-de-page de ce document. Votre premier exercice consiste donc à ouvrir ce PDF puis à copier/coller la commande ci-après dans votre console (*cf.* en bas dans la figure 1 page 5) et vérifier que vous obtenez le bon résultat, en bleu dans ce document.

```
print(pi)
[1] 3.141593
```

SI vous n'êtes pas arrivés à reproduire ce résultat, faites vous aider par votre chargé de TP, par un collègue de votre groupe de TP ou par le tutorat de l'université. Pour la suite nous supposons que cette étape est acquise.

1.2 Calculs élémentaires

VOUS pouvez utiliser  comme une calculatrice. La syntaxe des opérateurs arithmétiques usuels se trouve facilement en consultant la documentation du logiciel (*cf.* figure 2 page 6). Voici quelques exemples :

```
3 + 5
[1] 8
```

```
3*5
[1] 15
```

```
9/3
[1] 3
```

```
3^2  
[1] 9
```

ON peut utiliser l'opérateur d'affectation « <- », composé des deux caractères « < » et « - », pour ranger une valeur dans un objet. Par exemple, pour mettre la valeur 6 dans l'objet de nom `w` :

```
w <- 6  
w + w  
[1] 12
```

DONNEZ le code `R` permettant de calculer 6 fois la valeur de `w`, vous devez obtenir le résultat suivant :

```
[1] 36
```

Réponse :

L'OPÉRATEUR deux points « : » permet de générer des séries d'entiers consécutifs :

```
1:12  
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

Donnez le code `R` permettant de générer la série suivante :

```
[1] -5 -4 -3 -2 -1 0 1 2 3 4 5
```

Réponse :

LA fonction `c()` permet de construire « à la main » n'importe quel vecteur, par exemple :

```
x <- c(2, 5, 8)  
x  
[1] 2 5 8
```

DONNEZ le code `R` permettant de ranger les valeurs 7, 8 et 9 dans le vecteur `y` :

```
[1] 7 8 9
```


Réponse :

LES opérations arithmétiques usuelles fonctionnent directement avec des vecteurs ; elles se font élément par élément :


```
x + y  
[1] 9 13 17
```

2 Importation de données dans



2.1 Les données sur les notes des étudiants

VOUS avez sans doute l'habitude de manipuler vos données avec des outils de type tableur. Nous allons vous montrer comment importer facilement ce type de données sous  à l'aide d'un exemple détaillé. Puis ce sera à vous de jouer avec un autre jeu de données. Dans votre butineur de toile favori, copiez/collez l'adresse suivante¹ :


<https://goo.gl/bU3nJK>

VOUS devez obtenir la figure 3 page 6. Il s'agit d'un tableau de données où, par convention universellement suivie en statistiques, les individus (ici des étudiants) sont disposés en ligne et les variables en colonne. Pour pouvoir importer ces données dans , nous allons utiliser un fichier de données brut au format « `tsv` ». Ce fichier est un simple fichier texte dans lequel chaque ligne correspond à une ligne du tableau et chaque case est séparée par une tabulation. Ce fichier est disponible à l'adresse ci-dessous :

<https://pbil.univ-lyon1.fr/R/donnees/bsbi/mathsv.tsv>

VOUS pouvez également obtenir ce fichier à partir du tableur en téléchargeant une copie de ces données « `tsv` » (*cf.* figure 4 page 7). Cependant, nous n'allons pas avoir besoin de télécharger ces données brutes puisque  est capable d'importer des données à partir d'une adresse distante (l'url du fichier qui vous intéresse). Pour importer un fichier de données, qu'il soit présent en local sur votre disque dur ou qu'il soit disponible via une adresse distante, on utilise la fonction « `read.table()` ». Cette fonction permet de créer objet  de type « `DataFrame` » à partir d'un fichier de données. Une `DataFrame` peut être vu comme un tableau similaire à celui que vous avez visualisé dans le tableur. Dans le logiciel RStudio tapez la commande suivante:

```
etudiants <- read.table("https://pbil.univ-lyon1.fr/R/donnees/bsbi/mathsv.tsv",
  header = TRUE, dec = ',')
```



LA COMMANDE PRÉCÉDENTE demande à  de créer un objet de type `DataFrame` à partir du fichier « `mathsv.tsv` » et de l'enregistrer dans un objet nommé « `etudiants` ». Vous pouvez bien sûr choisir un nom différent. En plus de l'adresse du fichier, nous renseignons également deux arguments supplémentaires à la fonction.

`header = TRUE` permet de préciser que la première ligne du fichier correspond aux noms des colonnes et ne doit pas être considérée comme une ligne de donnée normale.

`dec = ','` permet de préciser que les valeurs numériques décimales utilisent la notation française, c'est-à-dire que nous utiliserons la virgule (et non pas le point anglo-saxon) pour séparer la partie entière de la partie décimale d'un nombre.

¹C'est une version compactée de https://docs.google.com/spreadsheets/d/1800QZrseB0F6PTpCFZ0tH2rw7xIK5PU81oJJR3evY_U/edit#gid=324731599



Figure 1: Copie d'écran de la partie gauche de l'interface de RStudio. Dans la console , en bas, on peut directement coller des instructions, par exemple `print(pi)`. L'intérêt de la fenêtre du haut sera expliquée dans la section « calculatrice programmable ». En cliquant sur « Source » le script , en haut, va être exécuté et les résultats affichés dans la console, en bas.

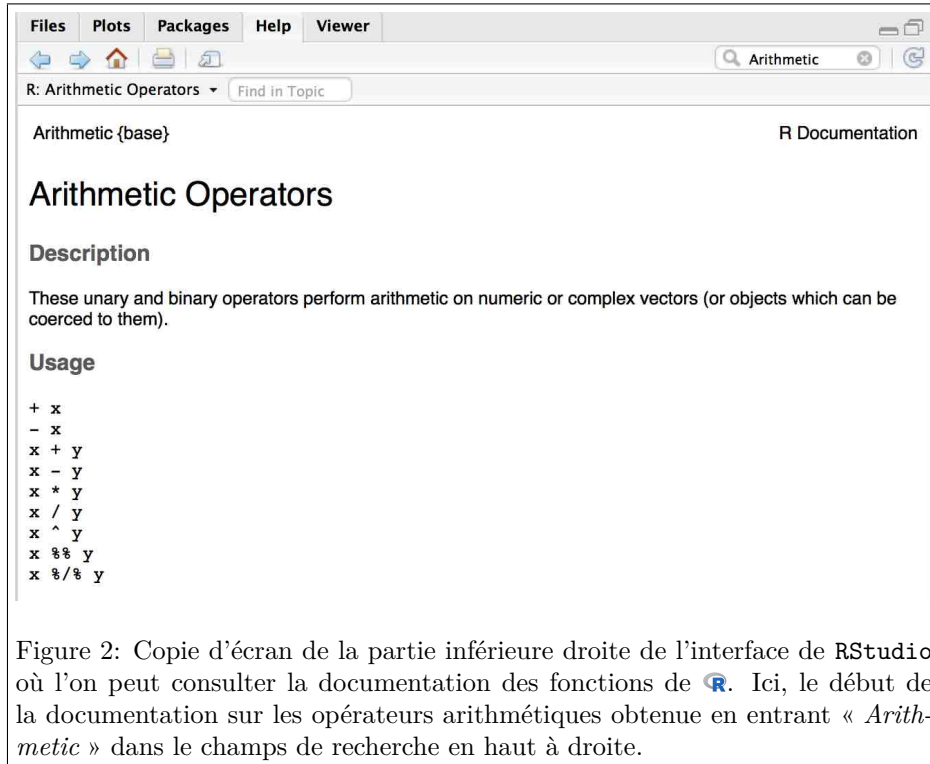


Figure 2: Copie d'écran de la partie inférieure droite de l'interface de RStudio où l'on peut consulter la documentation des fonctions de R. Ici, le début de la documentation sur les opérateurs arithmétiques obtenue en entrant « *Arithmetic* » dans le champs de recherche en haut à droite.

The screenshot shows a shared spreadsheet titled 'mathSVctp17'. The spreadsheet contains a table with 14 rows and 16 columns. The columns are labeled A through P. The data in the table represents student information, including names, scores, and various attributes.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	note	seq	ffsu	gEPS	sexe	abs	gMaths	gPhys	mBac	nMaths	nPhys	nBPSVT				
2	4,23	2	NON	+	M	>1	-	+	<12	10-12	<10					8,10
3	2,31	2	NON	-	F	0-1	-	+	<12	<10	>14					6,17
4	0,77	3	NON	-	M	>1		0	<12	12-14	10-12					10,03
5	12,31	2	NON	-	F	0-1		0	+	12-14	12-14	<10				8,25
6	9,62	2	NON	+	F	0-1		0	0	12-14	12-14	12-14				8,01
7	13,08	2	NON	-	F	0-1		0	0	<12	<10	<10				5,53
8	7,69	2	NON	-	M	>1		0	0	<12	<10	10-12				6,00
9	13,85	2	OUI	+	M	>1		0	0	<12	<10	<10				5,82
10	3,85	2	NON	+	M	0-1		0	+	<12	<10	10-12				4,00
11	7,69	2	NON	-	M	0-1			0	<12	<10	12-14				10,03
12	13,08	3	NON	-	F	>1		+	0	<12	10-12	<10				8,01
13	11,54	2	NON	+	M	0-1			0	<12	10-12	>14				8,03
14	10,00	2	NON	+	F	0-1			0	<12	<10	<10				8,53

Figure 3: Le tableau partagé contenant les données sur les étudiants.

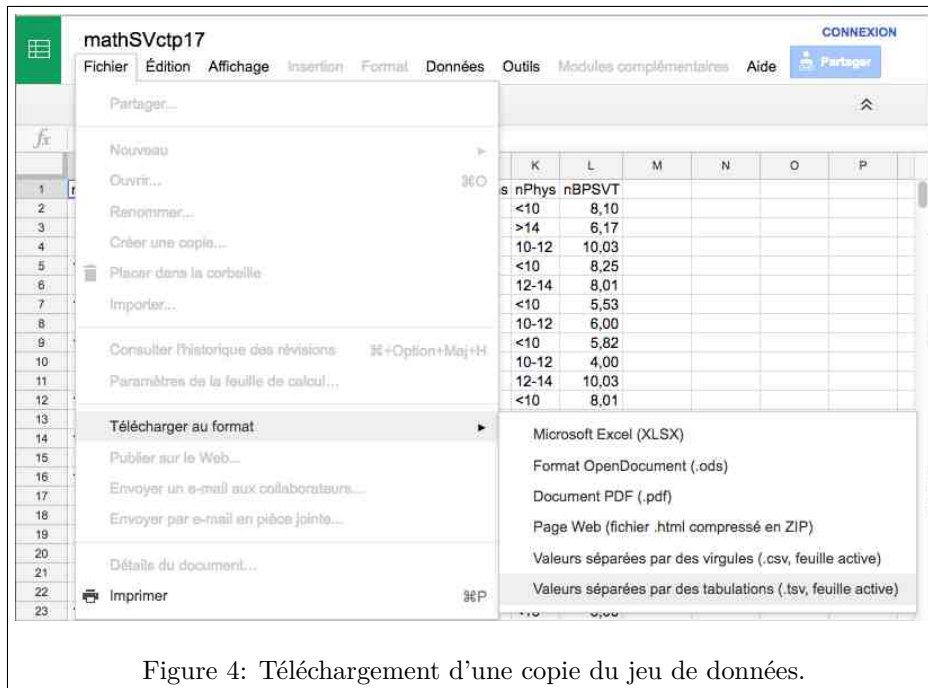


Figure 4: Téléchargement d'une copie du jeu de données.

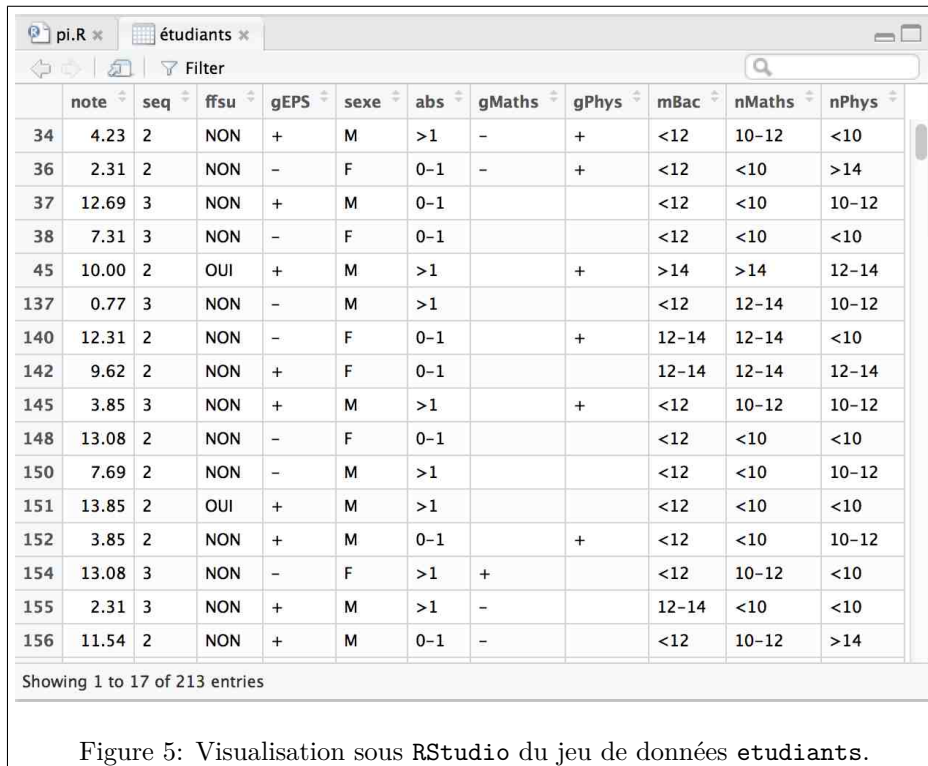


Figure 5: Visualisation sous RStudio du jeu de données étudiants.

etudiants

LE jeu de données **etudiants** est un échantillon de 220 étudiants inscrits à l'université Claude Bernard - Lyon 1 ayant eu un baccalauréat de la série S et ayant été notés à la première session du contrôle terminal de l'unité d'enseignement « MathSV » au semestre de printemps 2017. Les étudiants sont caractérisés par les 12 variables suivantes :

- 1° **note**, une variable quantitative donnant la note de l'étudiant en MathSV sur une échelle croissante allant, des moins bons résultats aux meilleurs, de 0 à 20 ;
- 2° **sequence**, une variable qualitative nominale indiquant si l'étudiant est inscrit en séquence 2 ou 3 ;
- 3° **ffsu**, une variable qualitative nominale à deux modalités OUI ou NON indiquant si l'étudiant est inscrit en FFSU^a de niveau 2 ;
- 4° **gEPS**, une variable qualitative ordonnée à deux modalités indiquant le goût autodéclaré de l'étudiant pour le sport (- ou +) ;
- 5° **sexe**, une variable qualitative nominale donnant le sexe de l'étudiant (M ou F) ;
- 6° **abs**, une variable qualitative ordonnée à deux modalités donnant le niveau d'absentéisme de l'étudiant lors des séances de TD de MathSV (0-1 ou >1) ;
- 7° **gMaths**, une variable qualitative ordonnée à trois modalités indiquant le goût autodéclaré de l'étudiant pour les mathématiques (-, 0 ou +) ;
- 8° **gPhys**, une variable qualitative ordonnée à trois modalités indiquant le goût autodéclaré de l'étudiant pour la physique (-, 0 ou +) ;
- 9° **mBac**, une variable qualitative ordonnée à trois modalités donnant la mention autodéclarée de l'étudiant au baccalauréat (<12, 12-14 ou >14) ;
- 10° **nMaths**, une variable qualitative ordonnée à quatre modalités donnant la note autodéclarée en mathématiques de l'étudiant au baccalauréat (<10, 10-12, 12-14, ou >14) ;
- 11° **nPhys**, une variable qualitative ordonnée à quatre modalités donnant la note autodéclarée en physique de l'étudiant au baccalauréat (<10, 10-12, 12-14, ou >14) ;
- 12° **nBPSVT**, une variable quantitative donnant la note de l'étudiant en « Bases de Physique pour les Sciences de la Vie et de la Terre » sur une échelle croissante allant de 0 à 20.

^aAcronyme de la « Fédération Française du Sport Universitaire ».

2.2 Les données sur la masse des bébés

Importez dans les données disponibles à l'adresse donnée ci-dessous :

<https://pbil.univ-lyon1.fr/R/donnees/bsbi/baby.tsv>

VOUS placerez ces données dans un objet nommé `bebes`. Vous devez obtenir le résultat de la figure 6 page 9. Les données sont décrites dans l'encart page 11. Vérifiez avant de passer à la suite que vous avez bien les objets `bebes` et `etudiants` présents dans votre environnement (*cf.* figure 7 page 10).

	bwt	gestation	parity	age	height	weight	smoke	tension
1	3.43	284	FALSE	27	155.0	45.30	FALSE	16.1
2	3.23	282	FALSE	33	160.0	61.16	FALSE	12.7
3	3.66	279	FALSE	28	160.0	52.09	TRUE	14.8
4	3.09	282	FALSE	23	167.5	56.62	TRUE	13.3
5	3.89	286	FALSE	25	155.0	42.13	FALSE	13.3
6	3.94	244	FALSE	33	155.0	80.63	FALSE	11.9
7	3.77	245	FALSE	23	162.5	63.42	FALSE	13.4
8	3.43	289	FALSE	25	155.0	56.62	FALSE	11.7
9	4.09	299	FALSE	30	165.0	61.61	TRUE	18.9
10	4.00	351	FALSE	27	170.0	54.36	FALSE	12.0
11	4.11	282	FALSE	32	160.0	56.17	TRUE	12.5
12	4.03	279	FALSE	23	157.5	57.98	TRUE	18.7
13	3.14	281	FALSE	36	152.5	44.85	TRUE	16.1
14	3.26	273	FALSE	30	157.5	69.76	FALSE	11.6
15	3.29	285	FALSE	38	157.5	58.89	FALSE	11.8
16	2.63	255	FALSE	25	162.5	56.62	TRUE	14.5

Showing 1 to 17 of 1,173 entries

Figure 6: Les données sur la masse des bébés.

3 Extraction des données utiles

3.1 Extraction de colonnes (variables : après la virgule)

ON peut accéder aux valeurs d'une colonne par sa position ou bien par son nom. Voici quatre façons de récupérer les données de la 5^e colonne donnant le sexe des étudiants. Notez que dans le cas de l'opérateur crochet « `[,]` », tout ce qui a trait aux colonnes se trouve à droite de la virgule :

```
etudiants[ , 5]
```

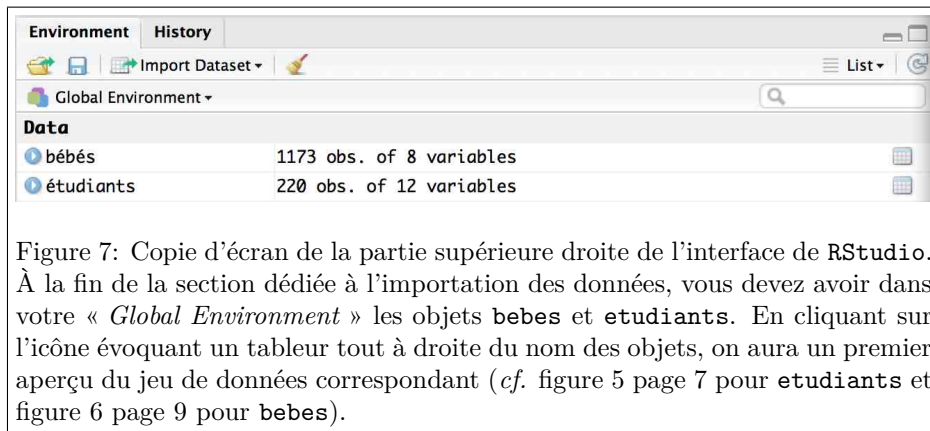


Figure 7: Copie d'écran de la partie supérieure droite de l'interface de RStudio. À la fin de la section dédiée à l'importation des données, vous devez avoir dans votre « *Global Environment* » les objets *bebes* et *etudiants*. En cliquant sur l'icône évoquant un tableau tout à droite du nom des objets, on aura un premier aperçu du jeu de données correspondant (cf. figure 5 page 7 pour *etudiants* et figure 6 page 9 pour *bebes*).

```
[1] "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "F" "F"
[21] "M" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "F" "F" "M" "F" "M" "M" "M"
[41] "F" "F" "F" "M" "M" "F" "F" "M" "F" "F" "M" "M" "F" "M" "M" "F" "M" "M" "M" "M"
[61] "F" "F" "M" "F" "F" "F" "F" "M" "F" "M" "M" "M" "M" "F" "F" "F" "M" "F" "F" "F"
[81] "F" "M" "F" "M" "F" "F" "F" "F" "M" "M" "F" "M" "M" "F" "F" "M" "F" "F" "F" "M"
[101] "F" "F" "M" "M" "F" "F" "F" "F" "M" "M" "F" "F" "F" "F" "F" "M" "M" "M" "F" "F"
[121] "F" "M" "F" "M" "F" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "M" "F" "M"
[141] "F" "F" "F" "F" "F" "F" "F" "F" "F" "F" "M" "F" "F" "F" "F" "M" "F" "F" "M" "M"
[161] "M" "M" "F" "M" "F" "F" "F" "M" "F" "M" "M" "M" "F" "M" "M" "F" "M" "F" "F" "F"
[181] "M" "F" "F" "M" "F" "M" "M" "F" "M" "F" "M" "M" "M" "F" "M" "F" "M" "F" "F" "M"
[201] "M" "F" "F" "M" "F" "M" "F" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "M" "M" "M"
```

```
etudiants[, "sexe"]
[1] "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "F" "F"
[21] "M" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "F" "F" "M" "F" "M" "M" "M"
[41] "F" "F" "F" "M" "M" "F" "F" "M" "F" "F" "M" "M" "F" "M" "M" "F" "M" "M" "M" "M"
[61] "F" "F" "M" "F" "F" "F" "F" "M" "F" "M" "M" "M" "M" "F" "F" "F" "M" "F" "F" "F"
[81] "F" "M" "F" "M" "F" "F" "F" "F" "M" "M" "F" "M" "M" "F" "F" "M" "F" "F" "F" "M"
[101] "F" "F" "M" "M" "F" "F" "F" "F" "M" "M" "F" "F" "F" "F" "F" "M" "M" "M" "F" "F"
[121] "F" "M" "F" "M" "F" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "M" "F" "M"
[141] "F" "F" "F" "F" "F" "F" "F" "F" "F" "F" "M" "F" "F" "F" "F" "M" "F" "F" "M" "M"
[161] "M" "M" "F" "M" "F" "F" "F" "M" "F" "M" "M" "M" "F" "M" "M" "F" "M" "F" "F" "F"
[181] "M" "F" "F" "M" "F" "M" "M" "F" "M" "F" "M" "M" "M" "F" "M" "F" "M" "F" "F" "M"
[201] "M" "F" "F" "M" "F" "M" "F" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "M" "M" "M"
```

```
etudiants$sexe
[1] "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "F" "F"
[21] "M" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "F" "F" "M" "F" "M" "M" "M"
[41] "F" "F" "F" "M" "M" "F" "F" "M" "F" "F" "M" "M" "F" "M" "M" "F" "M" "M" "M" "M"
[61] "F" "F" "M" "F" "F" "F" "F" "M" "F" "M" "M" "M" "M" "F" "F" "F" "M" "F" "F" "F"
[81] "F" "M" "F" "M" "F" "F" "F" "F" "M" "M" "F" "M" "M" "F" "F" "M" "F" "F" "F" "M"
[101] "F" "F" "M" "M" "F" "F" "F" "F" "M" "M" "F" "F" "F" "F" "F" "M" "M" "M" "F" "F"
[121] "F" "M" "F" "M" "F" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "M" "F" "M"
[141] "F" "F" "F" "F" "F" "F" "F" "F" "F" "F" "M" "F" "F" "F" "F" "M" "F" "F" "M" "M"
[161] "M" "M" "F" "M" "F" "F" "F" "M" "F" "M" "M" "M" "F" "M" "M" "F" "M" "F" "F" "F"
[181] "M" "F" "F" "M" "F" "M" "M" "F" "M" "F" "M" "M" "M" "F" "M" "F" "M" "F" "F" "M"
[201] "M" "F" "F" "M" "F" "M" "F" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "M" "M" "M"
```

```
with(etudiants, sexe)
[1] "M" "F" "M" "F" "F" "F" "M" "M" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "F" "F"
[21] "M" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "F" "F" "M" "F" "M" "M" "M"
[41] "F" "F" "F" "M" "M" "F" "F" "M" "F" "F" "M" "M" "F" "M" "M" "F" "M" "M" "M" "M"
[61] "F" "F" "M" "F" "F" "F" "F" "M" "F" "M" "M" "M" "M" "F" "F" "F" "M" "F" "F" "F"
[81] "F" "M" "F" "M" "F" "F" "F" "F" "M" "M" "F" "M" "M" "F" "F" "M" "F" "F" "F" "M"
[101] "F" "F" "M" "M" "F" "F" "F" "F" "M" "M" "F" "F" "F" "F" "F" "M" "M" "M" "F" "F"
[121] "F" "M" "F" "M" "F" "M" "F" "F" "F" "F" "F" "F" "F" "F" "M" "M" "F" "M" "F" "M"
[141] "F" "F" "F" "F" "F" "F" "F" "F" "F" "F" "M" "F" "F" "F" "F" "M" "F" "F" "M" "M"
[161] "M" "M" "F" "M" "F" "F" "F" "M" "F" "M" "M" "M" "F" "M" "M" "F" "M" "F" "F" "F"
[181] "M" "F" "F" "M" "F" "M" "M" "F" "M" "F" "M" "M" "M" "F" "M" "F" "M" "F" "F" "M"
[201] "M" "F" "F" "M" "F" "M" "F" "M" "M" "F" "M" "F" "F" "M" "F" "M" "M" "M" "M" "M"
```

DONNEZ le code R pour extraire le numéro de séquence à laquelle les étudiants étaient inscrits. Vous devez obtenir le résultat suivant :

3.2 Extraction de lignes (individus : avant la virgule)

ON cherche en général à extraire les individus qui satisfont certaines propriétés. Notez que, dans le cas de l'opérateur crochet « [,] », tout ce qui a trait aux lignes se trouve à gauche de la virgule. Par exemple, pour avoir le 66^e étudiant :

```
etudiants[66, ]
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
66 16.15      2  NON   -   F 0-1    +     0 >14  >14 10-12 10.85
```

POUR extraire les étudiants qui ont eu une note strictement supérieure à 19/20 :

```
etudiants[etudiants$note > 19, ]
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
40 20.00      3  NON   -   M >1    +     0 +12-14 >14 >14 8.55
47 20.00      2  NON   +   F 0-1    0     0 >14  >14 >14 9.80
61 19.62      2  NON   -   F 0-1    +     0 >14  >14 >14 12.66
67 19.62      3  NON   +   F 0-1    0     0 12-14 10-12 10-12 10.61
78 20.00      3  NON   +   M 0-1    0     0 12-14 10-12 >14 7.30
83 20.00      3  NON   -   F 0-1    +     + >14  >14 >14 10.05
110 20.00     2  OUI   +   F >1    +     + >14  >14 >14 10.07
149 20.00     3  NON   +   F 0-1    +     0 >14  >14 12-14 9.01
152 20.00     2  NON   +   F 0-1    +     0 >14  >14 >14 13.60
154 20.00     2  OUI   +   F 0-1    +     0 12-14 >14 >14 13.47
156 20.00     2  NON   +   F 0-1    +     0 12-14 12-14 >14 11.28
160 20.00     2  NON   +   M 0-1    0     0 >14  12-14 12-14 9.30
163 20.00     2  NON   -   F 0-1    0     0 >14  >14 >14 15.72
169 20.00     2  OUI   +   F 0-1    +     + >14  >14 >14 10.02
194 20.00     2  NON   -   F 0-1    0     0 12-14 12-14 >14 8.65
199 19.10     2  OUI   +   F 0-1    +     0 12-14 10-12 12-14 12.77
219 20.00     3  NON   -   M 0-1    0     + 12-14 12-14 12-14 4.31
```

DONNEZ le code R pour extraire les étudiants qui ont eu une note strictement inférieure à 1/20. Vous devez obtenir le résultat suivant :

```
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
3  0.77      3  NON   -   M >1    0     0 <12  12-14 10-12 10.03
25 0.77      3  NON   -   F >1    +     0 <12  12-14 >14 1.13
28 0.00      3  NON   -   F >1    -     0 <12  <10 12-14 1.17
44 0.00      3  NON   +   M >1    0     0 +12-14 10-12 12-14 3.61
52 0.77      3  NON   -   M >1    0     0 <12  <10 12-14 0.88
55 0.00     2  NON   +   M >1    0     - <12  <10 12-14 0.26
64 0.77     2  NON   +   F >1    -     + 12-14 <10 12-14 0.88
90 0.00     2  NON   -   M >1    -     0 <12  <10 >14 0.65
172 0.00     2  NON   +   F >1    0     0 <12  12-14 <10 7.01
182 0.00     3  NON   +   F 0-1    0     0 <12  <10 10-12 4.97
```

Réponse :

ON peut combiner plusieurs conditions avec les opérateurs logiques. Par exemple, pour avoir les étudiants de sexe mâle qui ont eu une note strictement supérieure à 19/20 :

```
etudiants[etudiants$note > 19 & etudiants$sexe == "M", ]
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
40  20      3  NON   -   M >1    +     + 12-14 >14 >14 8.55
78  20      3  NON   +   M 0-1    0     0 12-14 10-12 >14 7.30
160 20      2  NON   +   M 0-1    0     0 >14  12-14 12-14 9.30
219 20      3  NON   -   M 0-1    0     + 12-14 12-14 12-14 4.31
```

NOTEZ que la commande `with()` permet d'obtenir la même chose avec une écriture plus compacte, donc plus lisible :

```
with(etudiants, etudiants[note > 19 & sexe == "M", ])
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
40 20 3 NON - M >1 + 12-14 >14 >14 8.55
78 20 3 NON + M 0-1 0 0 12-14 10-12 >14 7.30
160 20 2 NON + M 0-1 0 0 >14 12-14 12-14 9.30
219 20 3 NON - M 0-1 0 + 12-14 12-14 12-14 4.31
```

DONNEZ le code `R` pour extraire les étudiants de sexe femelle qui ont eu une note strictement inférieure à 1/20. Vous devez obtenir le résultat suivant :

```
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
25 0.77 3 NON - F >1 + <12 12-14 >14 1.13
28 0.00 3 NON - F >1 - 0 <12 <10 12-14 1.17
64 0.77 2 NON + F >1 - + 12-14 <10 12-14 0.88
172 0.00 2 NON + F >1 0 0 <12 12-14 <10 7.01
182 0.00 3 NON + F 0-1 0 0 <12 <10 10-12 4.97
```

Réponse :

COMME dans le cas des colonnes, on peut utiliser l'opérateur deux points « : » pour extraire des lignes consécutives et la fonction `c()` pour extraire des lignes arbitraires, par exemple :

```
etudiants[1:5, ]
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
1 4.23 2 NON + M >1 - + <12 10-12 <10 8.10
2 2.31 2 NON - F 0-1 - + <12 <10 >14 6.17
3 0.77 3 NON - M >1 0 0 <12 12-14 10-12 10.03
4 12.31 2 NON - F 0-1 0 + 12-14 12-14 <10 8.25
5 9.62 2 NON + F 0-1 0 0 12-14 12-14 12-14 8.01
```

```
etudiants[c(1, 10, 144), ]
  note sequence ffsu gEPS sexe abs gMaths gPhys mBac nMaths nPhys nBPSVT
1 4.23 2 NON + M >1 - + <12 10-12 <10 8.10
10 7.69 2 NON - M 0-1 - 0 <12 <10 12-14 10.03
144 15.77 2 NON + F 0-1 + - <12 >14 <10 11.95
```

3.3 Extraction simultanée de lignes et de colonnes : les variables que je veux sur les individus que je veux

ON peut combiner les approches précédentes pour extraire directement les données qui nous intéressent. Par exemple, on aimerait connaître la séquence des étudiants de sexe femelle qui ont eu une note strictement supérieure à 19/20 :

```
with(etudiants, etudiants[note > 19 & sexe == "F", "sequence"])
[1] 2 2 3 3 2 3 2 2 2 2 2 2
```

ON peut extraire plusieurs colonnes d'un coup. Par exemple, pour vérifier que nous n'avons bien que des étudiants de sexe femelle ayant eu plus de 19/20 :

```
with(etudiants, etudiants[note > 19 & sexe == "F", c("note", "sexe", "sequence")])
  note sexe sequence
47 20.00 F 2
61 19.62 F 2
67 19.62 F 3
83 20.00 F 3
110 20.00 F 2
149 20.00 F 3
152 20.00 F 2
154 20.00 F 2
```

```
156 20.00 F 2
163 20.00 F 2
169 20.00 F 2
194 20.00 F 2
199 19.10 F 2
```

DONNEZ le code `R` pour extraire la note, le sexe et la mention au baccalauréat des étudiants de sexe mâle qui ont eu une note strictement inférieure à 1/20. Vous devez obtenir le résultat suivant :

```
  note sexe mBac
3 0.77 M <12
44 0.00 M 12-14
52 0.77 M <12
55 0.00 M <12
90 0.00 M <12
```

Réponse :

4 La calculatrice `R`

4.1 Calculatrice numérique vectorielle

Le logiciel `R` manipule directement des données vectorielles, par exemple pour calculer automatiquement pour chaque étudiant la moyenne entre la note de MathSV et celle BPSVT, il suffit d'écrire :


```
with(etudiants, (note + nBPSVT)/2)
 [1] 6.165 4.240 5.400 10.280 8.815 9.305 6.845 9.835 3.925 8.860 10.545
 [12] 9.785 9.265 9.310 12.690 7.965 7.465 4.675 4.325 4.940 5.370 10.155
 [23] 6.370 5.965 0.950 7.535 6.625 0.585 5.835 8.510 4.960 9.670 7.025
 [34] 9.600 9.185 9.465 7.245 13.610 9.205 14.275 5.150 8.935 9.365 1.805
 [45] 7.180 10.290 14.900 16.595 3.860 5.475 6.895 0.825 14.125 6.060 0.130
 [56] 5.440 13.695 6.180 15.225 7.190 16.140 15.770 11.430 0.825 3.490 13.500
 [67] 15.115 9.105 13.325 10.505 4.095 7.095 5.300 8.850 4.050 4.480 12.290
 [78] 13.650 6.650 11.865 9.235 3.945 15.025 8.185 10.375 10.000 10.810 9.830
 [89] 4.550 0.325 4.780 7.955 6.510 9.970 11.125 6.725 8.305 7.540 2.270
 [100] 10.750 8.705 9.550 11.435 5.335 13.765 6.680 3.805 13.485 2.615 15.035
 [111] 3.170 4.775 3.975 10.535 6.045 7.130 9.445 1.395 10.315 11.420 9.520
 [122] 10.020 4.230 6.245 6.075 9.615 13.015 9.910 16.390 4.190 5.365 13.035
 [133] 6.665 4.730 13.385 6.825 10.105 6.285 5.275 9.930 4.585 6.955 3.715
 [144] 13.860 8.980 11.375 2.410 12.590 14.505 10.545 8.140 16.800 6.320 16.735
 [155] 5.860 15.640 16.665 15.190 10.655 14.650 5.860 7.215 17.860 12.735 14.435
 [166] 10.765 13.305 4.515 15.010 5.490 10.695 3.505 5.560 7.075 10.575 4.935
 [177] 5.085 8.870 5.330 7.290 5.465 2.485 5.675 3.655 10.885 12.175 8.070
 [188] 10.380 5.765 1.190 10.145 11.575 9.690 14.325 6.840 14.905 7.875 9.805
 [199] 15.935 10.935 13.275 7.530 5.485 5.650 9.740 3.860 9.650 8.250 10.450
 [210] 7.660 9.200 4.005 4.865 7.485 2.640 8.665 16.965 4.010 12.155 7.730
```

L'OPÉRATION équivalente dans un tableur consisterait à entrer dans une nouvelle colonne la formule calculant la moyenne des deux notes, puis à étendre cette formule à l'ensemble des individus. Pour que l'analogie soit complète, il faudrait ajouter une nouvelle colonne dans l'objet `etudiants`, si on décide de l'appeler `moyenne`, le code `R` réalisant cette opération est :

```
etudiants$moyenne <- with(etudiants, (note + nBPSVT)/2)
```

L'INDICE de masse corporelle, IMC, le plus utilisé chez l'homme adulte est celui proposé par Adolphe QUÉTELET et défini par le rapport,

$$\text{IMC} = \frac{\text{masse}}{\text{taille}^2}$$

où la masse est exprimée en kg et la taille en m. Donnez le code  permettant de créer une nouvelle colonne IMC dans le jeu de données `bébés` correspondant à l'indice de masse corporelle des mères. On rappelle que la description des variables et des unités employées est documentée dans l'encart page 11. Pour les 50 premières mères, vous devez trouver les valeurs suivantes :


```
[1] 18.85536 23.89062 20.34766 20.18089 17.53590 33.56087 24.01704 23.56712 22.62994
[10] 18.80969 21.94141 23.37314 19.28514 28.12195 23.73998 21.44189 25.16444 26.64706
[19] 23.62902 21.49490 28.25752 24.12856 28.52941 20.93065 19.21515 22.83948 30.02320
[28] 21.44189 21.96511 20.45472 28.43967 20.46648 21.14222 20.58604 20.98998 20.99874
[37] 16.80222 17.32544 31.16582 32.32250 23.18087 22.30154 20.08768 24.95868 18.12000
[46] 24.45914 21.73243 24.51197 25.38793 21.42650
```


Réponse :

4.2 Calculatrice programmable

SUPPOSEZ que nous ayons envie de calculer toute une série de statistiques sur les notes des étudiants en MathSV :

```
x <- etudiants$note
print("Nombre de valeurs :")
print(length(x))
print("Valeur minimale :")
print(min(x))
print("Valeur maximale")
print(max(x))
print("Étendue :")
print(range(x))
print("Moyenne :")
print(mean(x))
print("Variance de la population :")
print(var(x))
print("Écart-type de la population :")
print(sd(x))
```

PLUTÔT que de copier/coller les commandes dans la console, créez un nouveau script  que vous sauvegardez chez vous sous le nom `script.R`. Collez dans ce document les commandes ci-dessus puis cliquez sur « Source » pour lancer leur exécution (*cf.* figure 1 page 5). Vous pouvez également sélectionner la partie du code que vous souhaitez exécuter puis tapez la combinaison de touches « Ctrl-Entrée » simultanément (raccourci du menu Code -> Exécuter les lignes sélectionnées).

DONNEZ la ligne de code  à modifier dans le script précédent pour calculer les mêmes indicateurs sur la masse des bébés à la naissance. Vous devez obtenir le résultat suivant :

```
[1] "Nombre de valeurs :"
```

```
[1] 1173
```

```
[1] "Valeur minimale :"
```

```
[1] 1.57
```

```
[1] "Valeur maximale"
```

```
[1] 5.03
```

```
[1] "Étendue :"
```

```
[1] 1.57 5.03
```

```
[1] "Moyenne :"
```

```
[1] 3.413291
```

```
[1] "Variance de la population :"
```

```
[1] 0.2744026
```

```
[1] "Écart-type de la population :"
```

```
[1] 0.5238346
```

Réponse :

L'UTILISATION de scripts est très commode dès que l'on dépasse une ligne de texte. C'est généralement le cas quand on utilise les fonctions graphiques de **R** car de nombreuses options sont disponibles. Dans les cadres réservés aux réponses dans la suite du document, vous pouvez recopier le code de la solution ou plus simplement donner le nom du fichier dans lequel vous avez sauvegardé votre solution.

5 Visualisation d'une variable à la fois

5.1 Cas d'une variable qualitative

5.1.1 Les diagrammes en bâtons

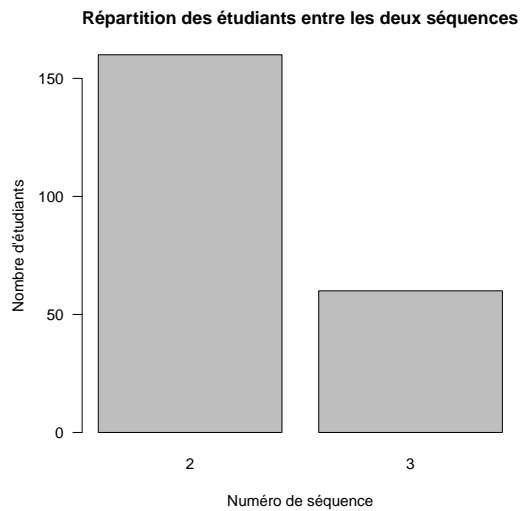
UNE variable qualitative est caractérisée par les fréquences absolues de ses modalités. La fonction `table()` permet d'effectuer cette opération, par exemple pour la séquence des étudiants :

```
table(etudiants$sequence)

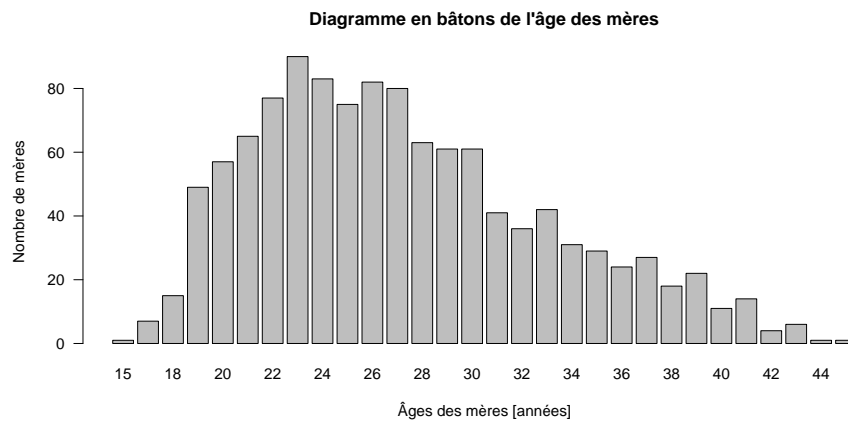
  2   3
160  60
```

UNE représentation en bâtons permet de bien visualiser la part relative des différentes modalités :

```
barplot(table(etudiants$sequence),
        main = "Répartition des étudiants entre les deux séquences",
        xlab = "Numéro de séquence",
        ylab = "Nombre d'étudiants",
        las = 1)
```

DONNEZ le code permettant de produire le diagramme en bâtons de l'âge² des mères dans le jeu de données `bébés` :



Réponse :

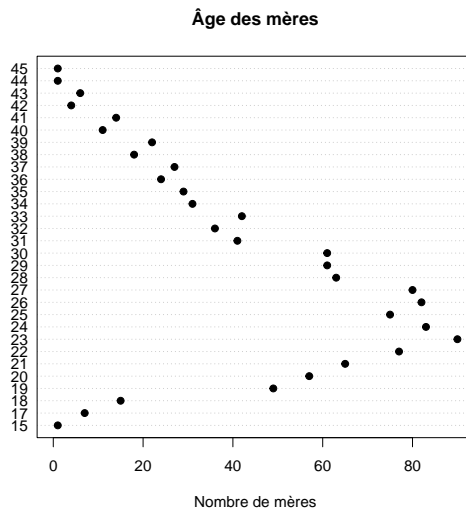
5.1.2 Pour aller plus loin (hors programme)

LE diagramme de CLEVELAND est une alternative au diagramme en bâtons qui possède l'avantage d'avoir un meilleur ratio information/encre :

```
x <- with(bebes, c(table(age)))
dotchart(x, xlim = c(0, max(x)),
```

²L'âge est considéré ici comme une variable qualitative ordonnée car exprimé par un ensemble d'entiers de faible cardinal.

```
main = "Âge des mères",
xlab = "Nombre de mères",
pch = 19)
```

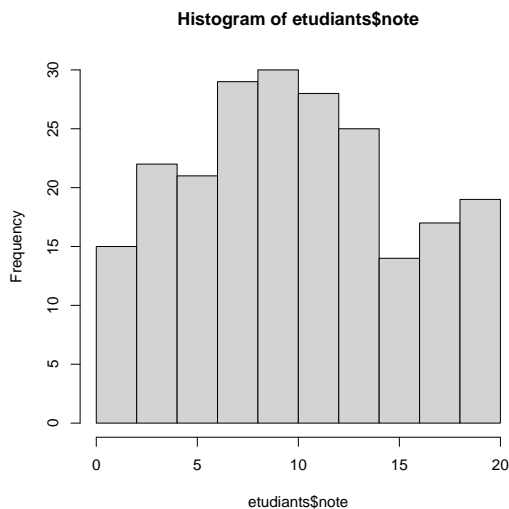


5.2 Cas d'une variable quantitative

5.2.1 Histogramme

La fonction `hist()` permet de tracer des histogrammes, par exemple pour représenter la distribution des notes en MathSV :

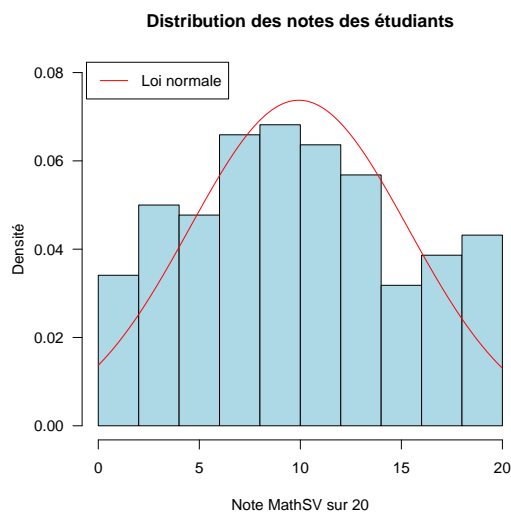
```
hist(etudiants$note)
```




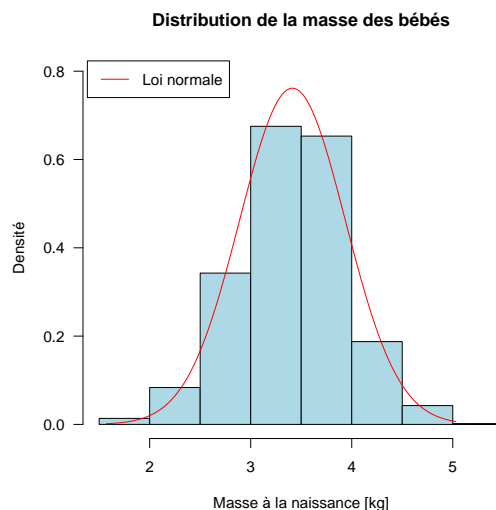
CETTE fonction possède de nombreuses options dont quelques-unes sont mises en œuvre ci-après. On consultera la documentation de cette fonction pour comprendre leur signification. Notez que l'option `proba = TRUE` permet d'avoir

une surface totale égale à 1, nous permettant de confronter les données à une fonction de densité de probabilité de référence, par exemple la loi normale $\mathcal{N}(\mu, \sigma)$. L'estimation de la moyenne de la population, $\hat{\mu}$, et l'estimation de l'écart-type de la population, $\hat{\sigma}$, sont données par les fonctions `mean()` et `sd()`, respectivement.

```
hist(etudiants$note,
     main = "Distribution des notes des étudiants",
     xlab = "Note MathSV sur 20",
     ylab = "Densité", col = "lightblue",
     las = 1, proba = TRUE, ylim = c(0, 0.08))
x <- with(etudiants, seq(from = min(note), to = max(note), length = 200))
with(etudiants, lines(x, dnorm(x, mean(note), sd(note)), col = "red"))
legend("topleft", inset = 0.01, legend = "Loi normale", lty = 1, col = "red")
```



DONNEZ le code  permettant de produire la représentation graphique suivante :

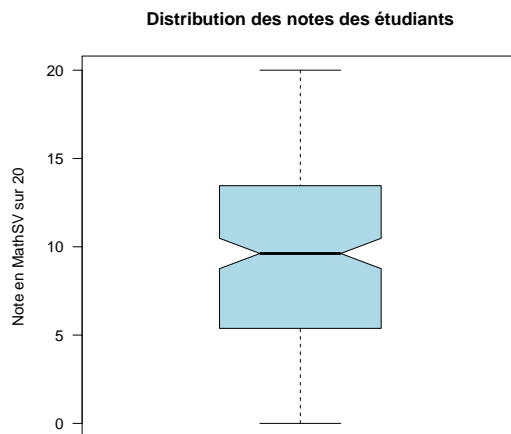



Réponse :

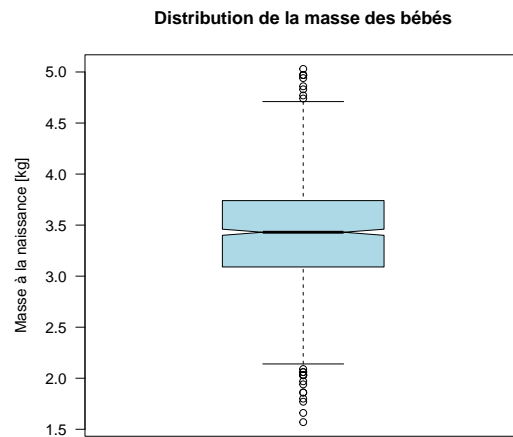
5.2.2 Boîte à moustaches

UNE autre possibilité pour représenter les variables quantitatives est celle des « boîtes à moustaches ». L'option `notch = TRUE` va tailler des encoches dans la boîte pour indiquer un intervalle de confiance à 95 % de la médiane de la valeur pour la population d'origine :

```
boxplot(etudiants$note,  
        main = "Distribution des notes des étudiants",  
        ylab = "Note en MathSV sur 20",  
        las = 1, notch = TRUE,  
        col = "lightblue")
```



DONNEZ le code  permettant de produire la représentation graphique suivante :

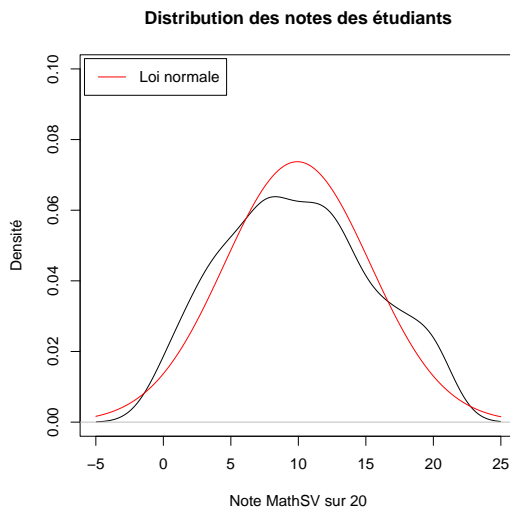


Réponse :

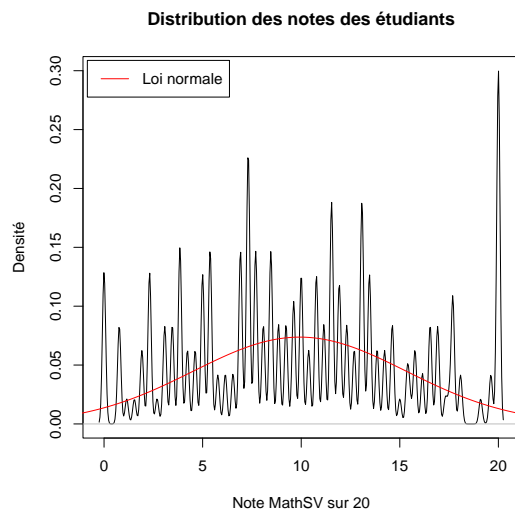
5.2.3 Pour aller plus loin (hors programme)

ON peut également utiliser un estimateur de la densité locale des notes. C'est une sorte d'histogramme à fenêtre glissante :

```
with(etudiants, plot(density(note, adjust = 1),
  main = "Distribution des notes des étudiants",
  xlab = "Note MathSV sur 20",
  ylab = "Densité",
  ylim = c(0, 0.1)))
x <- seq(from = -5, to = 25, length = 200)
with(etudiants, lines(x, dnorm(x, mean(note), sd(note)), col = "red"))
legend("topleft", inset = 0.01, legend = "Loi normale", lty = 1, col = "red")
```



Le paramètre `adjust` permet de contrôler la taille de la fenêtre glissante. Par exemple, avec une valeur de 0.05 pour ce paramètre :



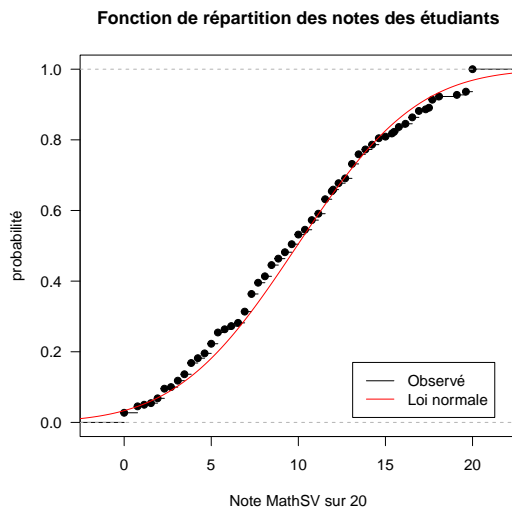
On met ici en évidence la nature discrète de la variable `note` : le nombre de notes différentes possibles n'est pas infini, la fonction `unique()` donne la liste des valeurs distinctes :

```
with(etudiants, sort(unique(note)))
[1] 0.00 0.77 1.15 1.54 1.92 2.31 2.69 3.08 3.46 3.85 4.23 4.62 5.00
[14] 5.38 5.77 6.15 6.54 6.92 7.31 7.69 8.08 8.46 8.85 9.23 9.62 10.00
[27] 10.38 10.77 11.15 11.54 11.92 12.00 12.31 12.69 13.08 13.46 13.85 14.23 14.62
[40] 15.00 15.38 15.50 15.77 16.15 16.54 16.92 17.31 17.50 17.69 18.08 19.10 19.62
[53] 20.00
```

Il y a donc 53 notes différentes possibles. Une autre façon de procéder est de représenter la fonction de répartition qui donne la probabilité pour qu'une note soit inférieure à une valeur donnée. Les notes étant bornées entre 0 et 20, la probabilité pour qu'une note soit strictement inférieure à 0 est nulle, la

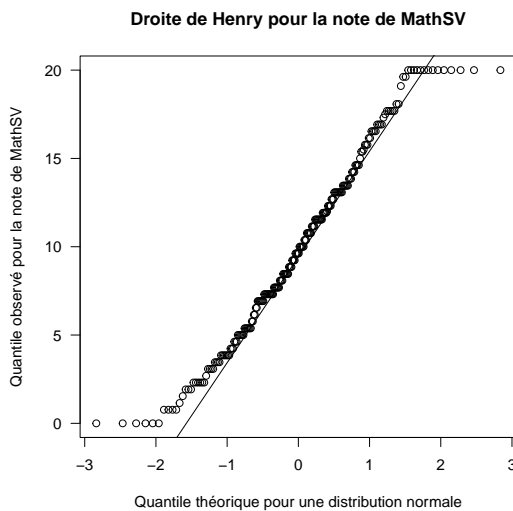
probabilité pour qu'une note soit inférieure ou égale à 20 est celle d'un événement certain, la fonction de répartition a l'allure d'une sigmoïde entre 0 et 1 :

```
plot(ecdf(etudiants$note),
     main = "Fonction de répartition des notes des étudiants",
     xlab = "Note MathSV sur 20",
     ylab = "probabilité",
     las = 1)
with(etudiants, lines(x, pnorm(x, mean(note), sd(note)), col = "red"))
legend("bottomright", inset = 0.05, legend = c("Observé", "Loi normale"),
      lty = 1, col = c("black", "red"))
```

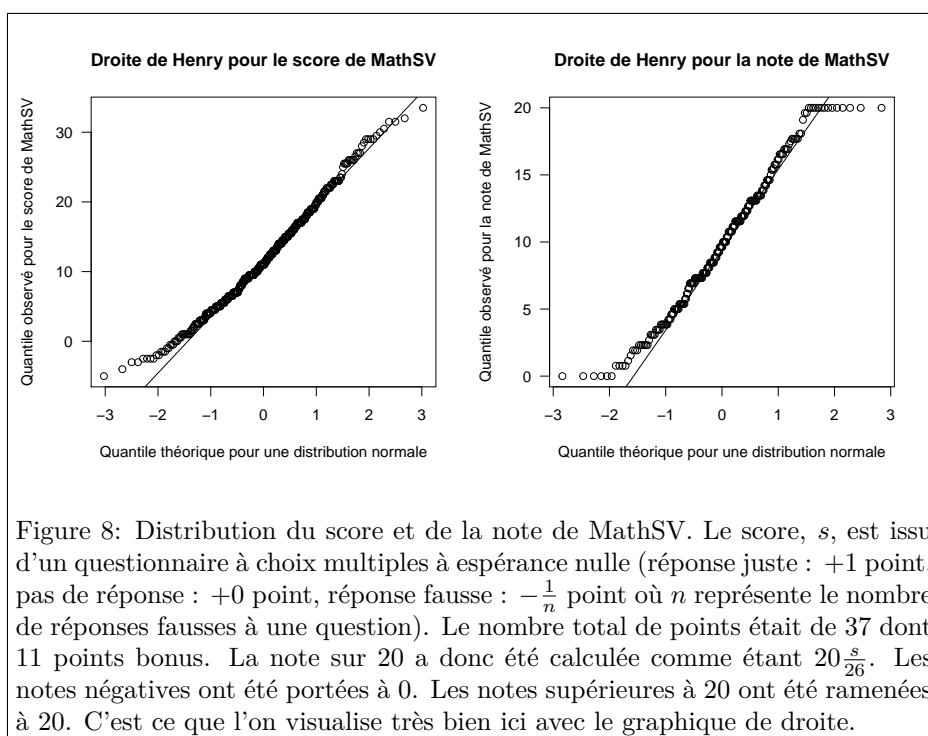


ENFIN, on peut confronter directement les valeurs des courbes des fonctions de répartition observée et théorique pour produire le graphique « quantiles-quantiles », dit aussi droite de HENRY :

```
with(etudiants, qqnorm(note, las = 1, main = "Droite de Henry pour la note de MathSV",
  ylab = "Quantile observé pour la note de MathSV",
  xlab = "Quantile théorique pour une distribution normale"))
qqline(etudiants$note)
```



DANS cette représentation graphique, les points sont alignés dans le cas d'une distribution normale. Notez sa puissance puisqu'elle nous permet de détecter un phénomène qui n'était pas évident avec les graphiques précédents. La distribution est raisonnablement normale dans sa partie centrale mais on observe un décrochage aux deux extrémités pour les notes 0 et 20. C'est assez logique quand on y pense puisque la note est bornée entre 0 et 20 alors qu'une variable normale est supposée pouvoir gambader librement entre $-\infty$ et $+\infty$. Ce que l'on visualise en fait ici, c'est le procédé employé pour construire la note de MathSV (*cf.* figure 8 page 24).

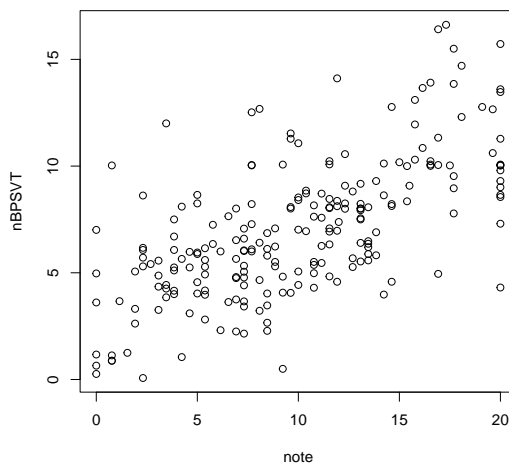


6 Visualisation de deux variables à la fois

6.1 Deux variables quantitatives

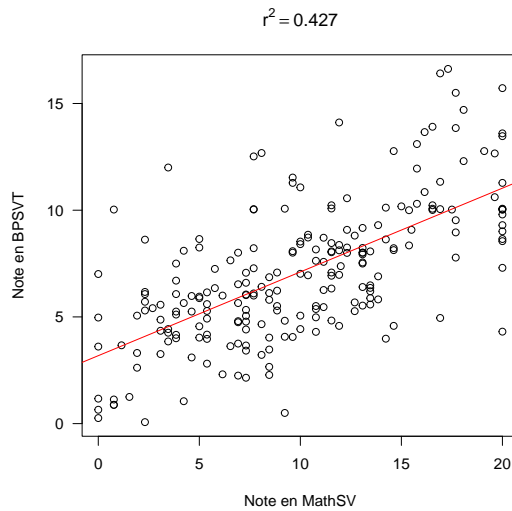
La fonction `plot(x, y)` permet de produire des nuages de points dont les coordonnées en abscisse sont données par `x` et celles en ordonnée par `y`, par exemple pour comparer la note de MathSV avec celle de BPSVT :

```
with(etudiants, plot(note, nBPSVT))
```

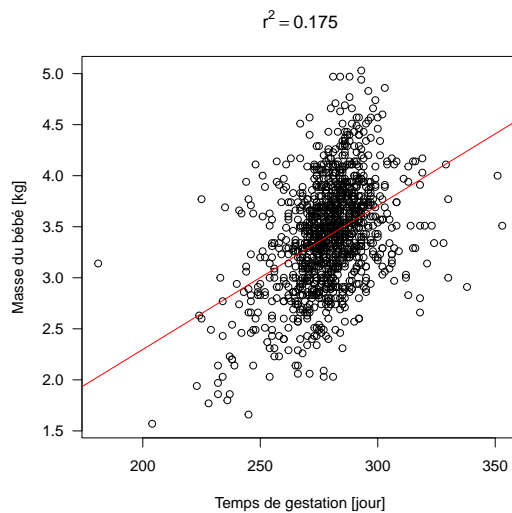


On peut enrichir ce graphique en tirant parti des options de la fonction `plot()` et en utilisant la fonction `abline()` pour ajouter la droite de régression linéaire :

```
#  
# Calculons le carré du coef de corrélation linéaire  
#  
r2 <- summary(lm(nBPSVT~note, etudiants))$r.squared  
#  
# Ne gardons que trois chiffres significatifs  
#  
r2 <- signif(r2, 3)  
#  
# Dessinons le nuage de points  
#  
with(etudiants, plot(note, nBPSVT, las = 1,  
  xlab = "Note en MathSV", ylab = "Note en BPSVT",  
  main = bquote(r^2 == .(r2))))  
#  
# Ajoutons la droite de régression  
#  
abline(lm(nBPSVT~note, etudiants), col = "red")
```



DONNEZ le code permettant de produire la représentation graphique suivante :

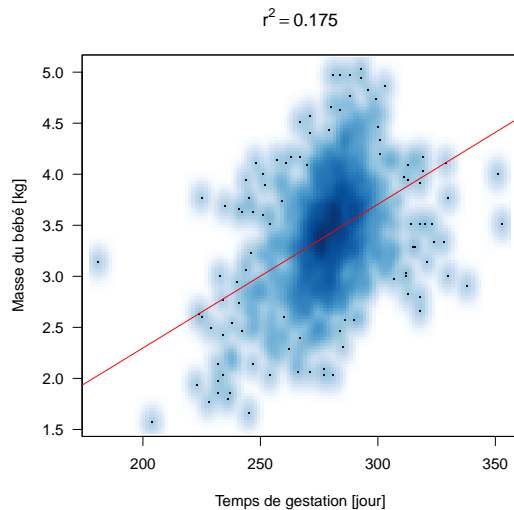


Réponse :

6.2 Deux variables quantitatives (hors programme)

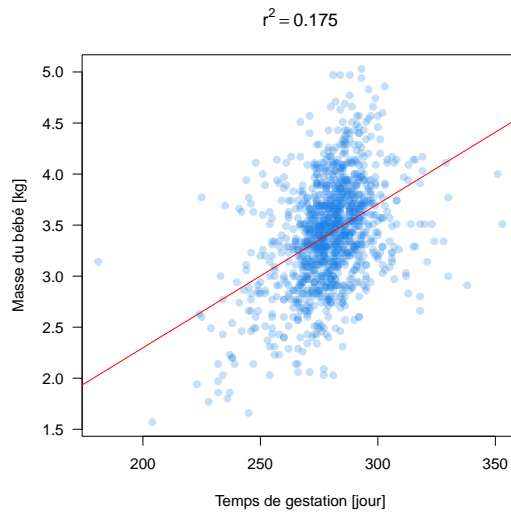
QUAND le nombre de points à représenter devient trop important (des milliers de points) il y a risque de superposition des points et donc d'un graphique trompeur. La fonction `smoothScatter()` offre une alternative utile en représentant la densité locale des points dans le plan :

```
r2 <- round(summary(lm(bwt~gestation, bebes))$r.squared, 3)
#
with(bebes, smoothScatter(gestation, bwt, las = 1,
  xlab = "Temps de gestation [jour]", ylab = "Masse du bébé [kg]",
  main = bquote(r^2 == .(r2))))
#
abline(lm(bwt~gestation, bebes), col = "red")
```



UNE autre possibilité est d'utiliser des couleurs transparentes pour les points de sorte que les superpositions sont mieux mises en évidence. On peut générer des couleurs transparentes en jouant sur le paramètre `alpha` de la fonction `rgb()` :

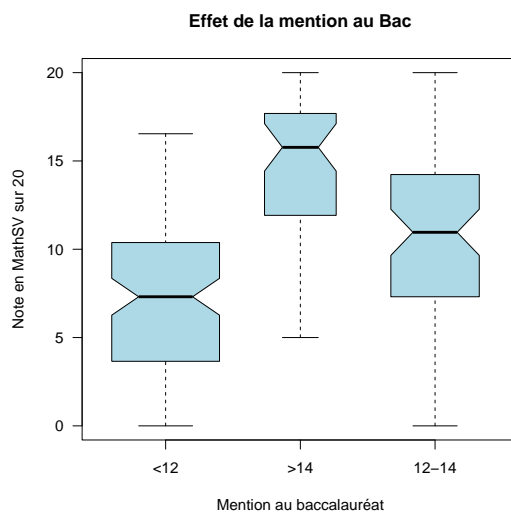
```
r2 <- round(summary(lm(bwt~gestation, bebes))$r.squared, 3)
#
with(bebes, plot(gestation, bwt, las = 1,
  pch = 19, col = rgb(0.1, 0.5, 0.9, 0.25),
  xlab = "Temps de gestation [jour]", ylab = "Masse du bébé [kg]",
  main = bquote(r^2 == .(r2))))
#
abline(lm(bwt~gestation, bebes), col = "red")
```



6.3 Quantitatif-Qualitatif

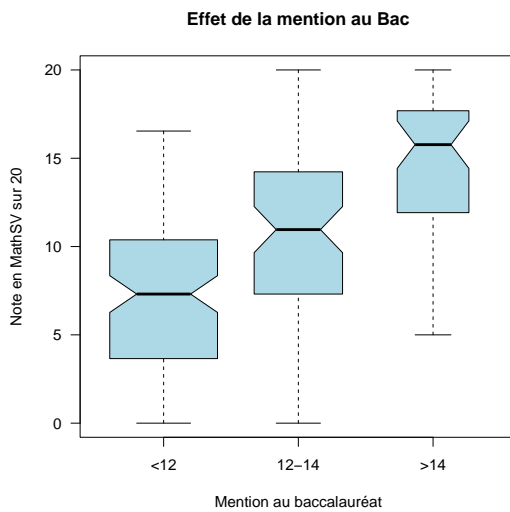
La fonction `boxplot()` permet très facilement de visualiser l'effet d'une variable qualitative sur une variable quantitative. Voyons par exemple l'effet de la mention au baccalauréat sur la note en MathSV. La notation `note~mBac` se lit comme « la note en fonction de la mention au baccalauréat ». L'option `varwidth = TRUE` va imposer que la surface des boîtes soit proportionnelle aux effectifs des étudiants, ce qui permet de détecter s'il y a des déséquilibres entre les groupes.

```
boxplot(note~mBac, etudiants, main = "Effet de la mention au Bac",
        xlab = "Mention au baccalauréat",
        ylab = "Note en MathSV sur 20",
        las = 1, col = "lightblue", varwidth = TRUE,
        notch = TRUE)
```

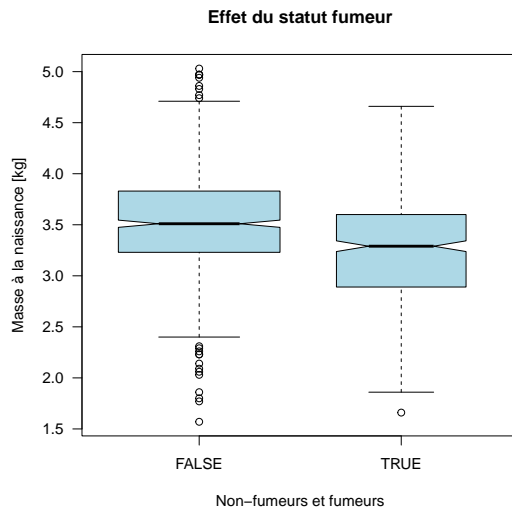


CETTE représentation graphique n'est pas correcte parce que la mention au baccalauréat est une variable qualitative *ordonnée*. Pour indiquer que c'est une variable qualitative ordinale, il faut le faire explicitement :

```
etudiants$mBac <- ordered(etudiants$mBac, levels = c("<12", "12-14", ">14"))
boxplot(note~mBac, etudiants, main = "Effet de la mention au Bac",
        xlab = "Mention au baccalauréat",
        ylab = "Note en MathSV sur 20",
        las = 1, col = "lightblue", varwidth = TRUE,
        notch = TRUE)
```



DONNEZ le code permettant de représenter l'effet du statut fumeur ou non de la mère sur la masse des bébés.

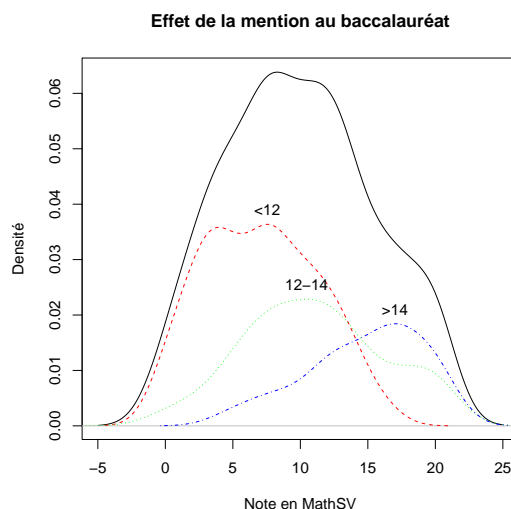


Réponse :

6.4 Quantitatif-Qualitatif (hors programme)

UNE autre possibilité est d'utiliser des fonctions de densité de probabilité. Un avantage par rapport aux histogrammes est que l'on peut facilement les superposer sur un même graphique sans le rendre illisible. Il n'y a cependant pas de fonction standard pour cette représentation graphique. Il faut la programmer soi-même. Dans le script suivant vous pouvez vous amuser à modifier la valeur du paramètre `adjust` de la fonction `density()` pour comprendre son effet sur la représentation obtenue.

```
adjust <- 1
nc <- length(levels(etudiants$mBac)) # Nombre de modalités
#
plot(density(etudiants$note, adjust = adjust),
     main = "Effet de la mention au baccalauréat",
     xlab = "Note en MathSV", ylab = "Densité")
#
mylty <- 2:(2 + nc - 1) # Mes types de lignes
mycol <- rainbow(nc)     # Mes types de couleurs
i <- 1 # Compteur d'appel de la fonction
#
mafonction <- function(x, ...){
  dst <- density(x, adjust = adjust)
  dst$y <- length(x)*dst$y/nrow(etudiants)
  lines(dst$x, dst$y, lty = mylty[i], col = mycol[i])
  imax <- which.max(dst$y)
  label <- as.character(levels(etudiants$mBac)[i])
  text(dst$x[imax], dst$y[imax], label = label, pos = 3)
  i <- i + 1
}
with(etudiants, tapply(note, mBac, mafonction))
```



6.5 Deux variables qualitatives

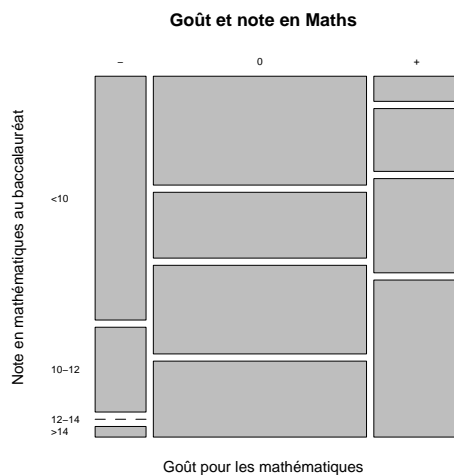
La fonction `table()` permet de prendre en compte simultanément deux variables qualitatives. Intéressons-nous par exemple au goût pour les mathématiques des étudiants (`gMaths`) et la note qu'ils ont obtenu en mathématiques au baccalauréat (`nMaths`).

```
with(etudiants, table(gMaths, nMaths))
      nMaths
gMaths <10 >14 10-12 12-14
-      23  1  8  0
+      4 25 10 15
0     43 30 26 35

# Ordonnons les modalités :
etudiants$gMaths <- ordered(etudiants$gMaths, levels = c("-", "0", "+"))
etudiants$nMaths <- ordered(etudiants$nMaths, levels = c("<10", "10-12", "12-14", ">14"))
with(etudiants, table(gMaths, nMaths))
      nMaths
gMaths <10 10-12 12-14 >14
-      23  8  0  1
0     43 26 35 30
+      4 10 15 25
```

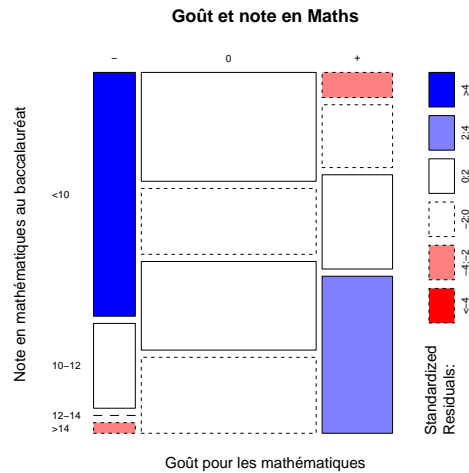
UN tableau tel que celui-ci, ventilant des individus entre les modalités croisées de deux variables qualitatives s'appelle une table de contingence. La fonction `mosaicplot()` donne une représentation graphique des tables de contingences :

```
with(etudiants, mosaicplot(table(gMaths, nMaths),
                             main = "Goût et note en Maths",
                             xlab = "Goût pour les mathématiques",
                             ylab = "Note en mathématiques au baccalauréat",
                             las = 1))
```



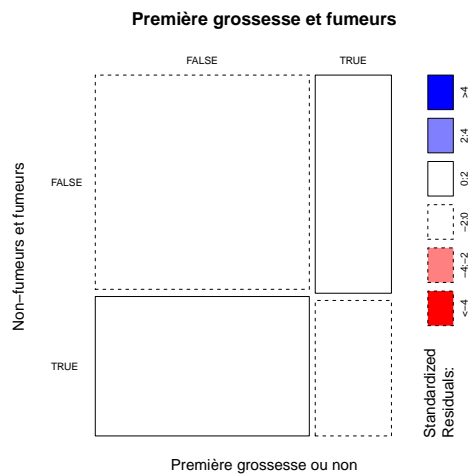
DANS cette représentation, la surface des rectangles est proportionnelle aux effectifs des étudiants concernés. Par exemple, le plus grand rectangle ici correspond aux 43 étudiants qui ont un goût modéré pour les mathématiques et une note <10 en mathématiques au baccalauréat. Une option intéressante de cette fonction est `shade = TRUE` :

```
with(etudiants, mosaicplot(table(gMaths, nMaths),
  main = "Goût et note en Maths",
  xlab = "Goût pour les mathématiques",
  ylab = "Note en mathématiques au baccalauréat",
  las = 1, shade = TRUE))
```



CETTE option permet de mettre évidence les couples de modalités qui présentent un excès (en bleu) ou un défaut (en rouge) par rapport aux effectifs qui seraient attendus si les deux variables étaient indépendantes. On constate ici qu'il y a un excès d'étudiants qui n'aiment pas les mathématiques et qui ont eu moins de 10, un excès d'étudiants qui aiment les mathématiques et qui ont eu plus de 14, un défaut d'étudiants qui n'aiment pas les mathématiques et qui ont eu plus de 14, et un défaut d'étudiants qui aiment les mathématiques et qui ont eu moins de 10.

DONNEZ le code permettant de représenter la relation entre le fait de fumer ou non pour les mères et le fait que cela soit leur première grossesse ou non :



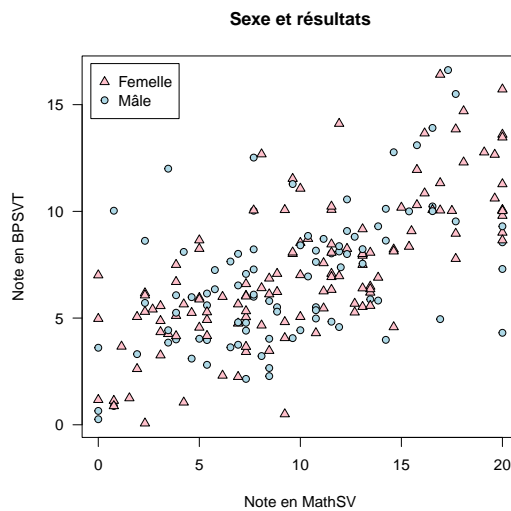
Réponse :


7 Visualisation de trois variables à la fois

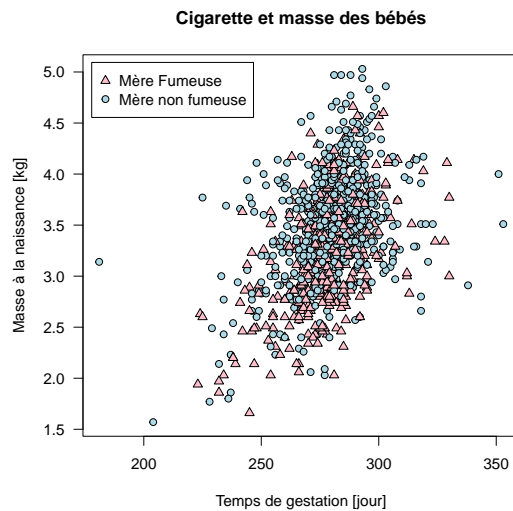
7.1 Quanti-Quanti-Quali

IL s'agit ici d'un nuage de points sur lequel on porte une information supplémentaire à l'aide d'un code graphique (type et couleur des points). La liste des types de points disponibles est donnée dans la documentation de la fonction `points()`. Un aperçu des couleurs pré-définies est donné en invoquant `demo("colors")` dans la console. Par exemple, on aimerait connaître le sexe des étudiants quand on croise la note en MathSV et la note en BPSVT :

```
with(etudiants,
  plot(note, nBPSVT, las = 1,
        xlab = "Note en MathSV", ylab = "Note en BPSVT",
        pch = ifelse(sexe == "F", 24, 21),
        main = "Sexe et résultats",
        bg = ifelse(sexe == "F", "pink", "lightblue")))
#
legend("topleft", inset = 0.02, legend = c("Femelle", "Mâle"), pch = c(24, 21),
      pt.bg = c("pink", "lightblue"))
```



DONNEZ le code  permettant de repérer les mères fumeuses et non fumeuses quand on croise le temps de gestation avec la masse à la naissance des bébés :

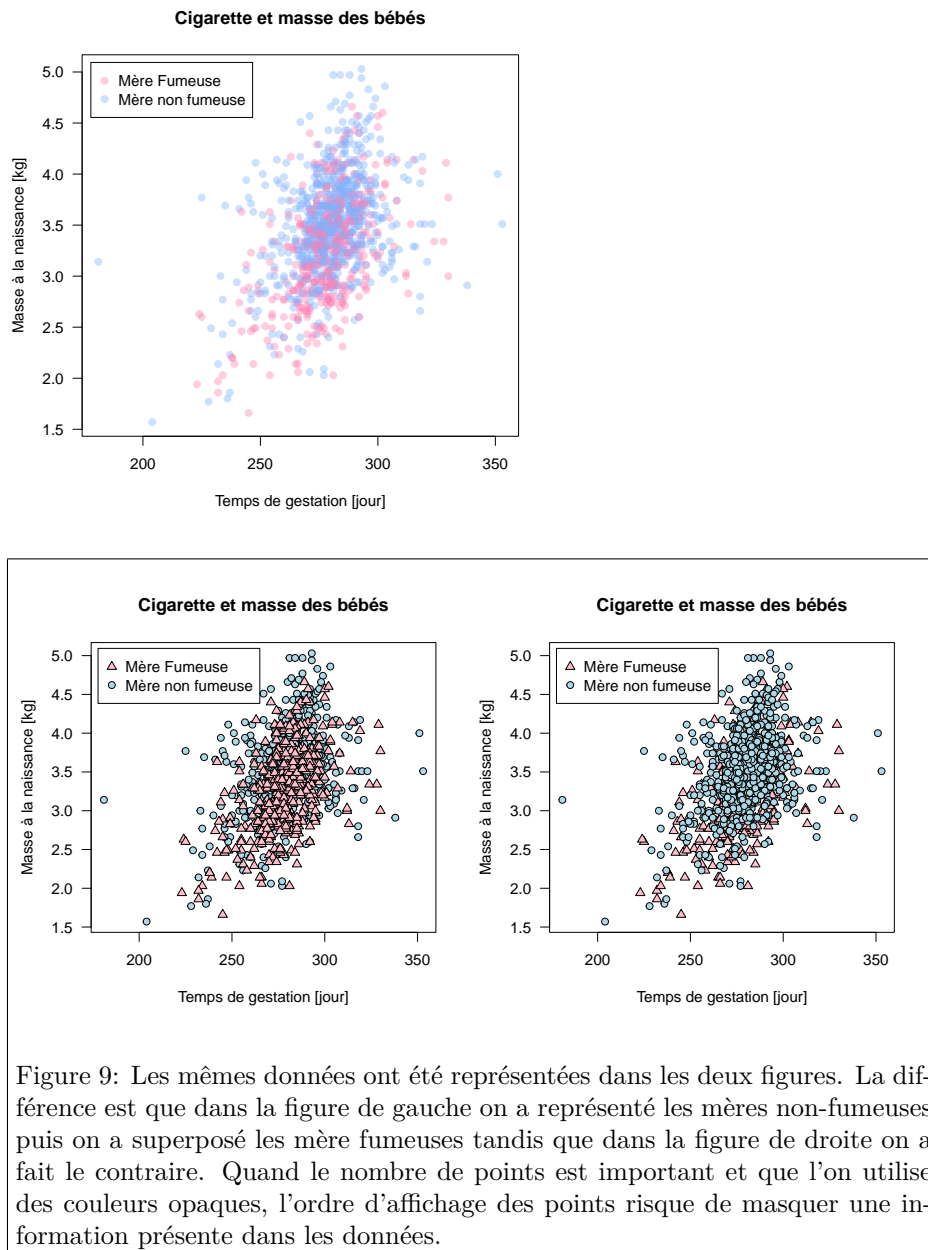


Réponse :

7.2 Quanti-Quanti-Quali (hors programme)

QUAND dans un nuage de points le nombre d'iceux est élevé (de l'ordre de plus de 10^3), vouloir représenter une variable qualitative supplémentaire à titre illustratif devient délicat (*cf.* figure 9 page 35). L'utilisation de couleurs transparentes permet de gagner un ou deux ordres de grandeur sur le seuil de 10^3 , mais n'y voyez pas une panacée, il finira toujours par avoir trop de points par rapport à votre capacité de perception des couleurs.

```
colF <- rgb(1.0, 0.5, 0.7, 0.4) # Couleur des fumeuses
colNF <- rgb(0.5, 0.7, 1.0, 0.4) # Couleur des non fumeuses
with(bebes,
  plot(gestation, bwt, las = 1, pch = 19,
       xlab = "Temps de gestation [jour]",
       ylab = "Masse à la naissance [kg]",
       main = "Cigarette et masse des bébés",
       col = ifelse(smoke == TRUE, colF, colNF))
  legend("topleft", inset = 0.02, legend = c("Mère Fumeuse", "Mère non fumeuse"),
       pch = 19, col = c(colF, colNF))
```

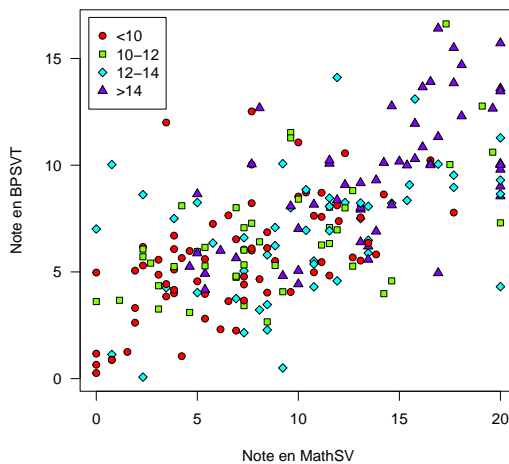


La situation se complique également lorsque la variable qualitative illustrative possède plus de 2 modalités. Voici un exemple qui fonctionnera jusqu'à quatre modalités (pour plus de 4 modalités il est préférable de représenter les groupes par des ellipses ou des enveloppes convexes, ce qui est encore plus hors-programme). On aimerait représenter en variable supplémentaire la classe de note en mathématiques au baccalauréat quand on croise la note en MathSV avec celle en BPSVT :

```
nc <- length(levels(etudiants$Maths)) # Le nombre de classes
if(nc > 4) stop("Ça ne marchera pas")
```

```
mycol <- rainbow(nc) # Mes couleurs
mypch <- 21:(21 + nc - 1) # Mes types de points
with(etudiants,
  plot(note, nBPSVT, las = 1,
        xlab = "Note en MathSV", ylab = "Note en BPSVT",
        pch = mypch[nMaths],
        main = "Note en mathématiques au baccalauréat et résultats en L1",
        bg = mycol[nMaths]))
legend("topleft", inset = 0.02, legend = levels(etudiants$nMaths), pch = mypch,
      pt.bg = mycol)
```

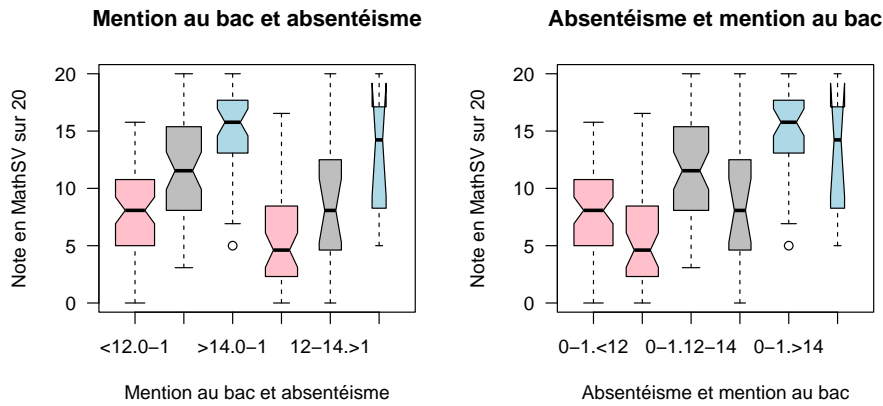
Note en mathématiques au baccalauréat et résultats en L1



7.3 Quanti-Quali-Quali

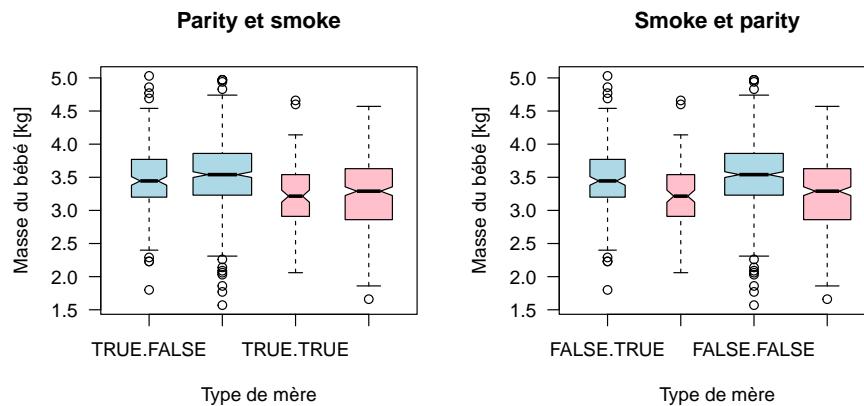
ON cherche dans ce cas à analyser l'effet de deux variables qualitatives sur une variable quantitative, par exemple, quel est l'effet de la mention au baccalauréat et de l'absentéisme en TD sur la note de MathSV ? La notation `note~mBac+abs` se lit « la note en fonction de la mention au bac et de l'absentéisme ». Le graphique produit va dépendre de l'ordre des variables explicatives. Pour faciliter la lecture on va colorier en rose les mentions <12, en gris les mentions 12-14 et en bleu les mentions >14.

```
# Ordonnons les modalités d'absentéisme
etudiants$abs <- ordered(etudiants$abs, levels = c("0-1", ">1"))
par(mfrow = c(1, 2)) # Pour avoir deux graphiques côte à côte
boxplot(note~mBac+abs, etudiants, main = "Mention au bac et absentéisme",
        xlab = "Mention au bac et absentéisme",
        ylab = "Note en MathSV sur 20",
        las = 1, col = c("pink", "grey", "lightblue"),
        varwidth = TRUE, notch = TRUE)
boxplot(note~abs+mBac, etudiants, main = "Absentéisme et mention au bac",
        xlab = "Absentéisme et mention au bac",
        ylab = "Note en MathSV sur 20",
        las = 1, col = rep(c("pink", "grey", "lightblue"), each = 2),
        varwidth = TRUE, notch = TRUE)
```



Le graphique de gauche permet de visualiser facilement l'effet de la mention au baccalauréat pour les deux modalités d'absentéisme, le graphique de droite celui de l'absentéisme pour chaque modalité de mention au baccalauréat. DONNEZ le code `R` permettant de représenter l'effet simultané du tabagisme et de la parité sur la masse des bébés à la naissance. Pour faciliter la lecture des graphiques, on pourra utiliser le recodage suivant :

```
bebes$smoke <- ordered(ifelse(bebes$smoke, TRUE, FALSE), levels = c(FALSE, TRUE))
bebes$parity <- ordered(ifelse(bebes$parity, TRUE, FALSE), levels = c(TRUE, FALSE))
```



Réponse :

8 Sauvegarde des données

VOUS allez utiliser les données `etudiants` et `bebes` lors du prochain TP. Pour sauvegarder d'un coup tous les objets définis dans votre environnement, cliquez sur l'icône qui ressemble à une disquette dans l'onglet « *Environment* »



en haut à droite de l'interface de **RStudio**. Pour restaurer tout votre environnement à la prochaine séance, il suffira de cliquer sur l'icône juste à gauche de la précédente (elle représente une flèche qui sort d'un dossier).