[II.2313] Analyse de données - TP 1

isep
École d'ingénieurs du numérique

Rémi Biolley – Thierry Lincoln

A- Séries Discrètes

1. On génère un vecteur de taille 1000 dont les valeurs se situent entre 0 et 10 :

```
> A=round(runif(1000,0,10))
> A
   [1]  4  1  4  7  3  1  3  6  4  1  4  5  8  8  5  0  0  2  5  4  6  1  5  1
  [25]  8  6  2  4  8 10  5  1  5  4 10  1  9  2  1  5  1  5  4 10  6  4  2  2
  [49]  7  5  7  1  7  6  1  5  7  6  7  2  3  3  3  7  3  1  9  4  9  1  9  9
  [73] 10  3  0  0 10  4  6  2  7  4  9  0  6  8 10  3  1  3  5 10  9  6  3  4
  [97]  7  9  6  8 10  9 10  9  1  2  3  9  2  6  1  7  6  2  9  2  7  3  9  3
 [121]  1  2  7  4  7 10  9  9  7  6  4  6  8  5  5  9  2  3  9  6  3  6  1  7
 [145]  5  7  8  1  4  3  3  0  8  7  6  9  3  2  4  7  2  3  1  1  7  1  5  0
 [169]  7  6  2  9  2  7  5 10  1  8  9  5  7  3  7  2  2  6  2  8  9 10  3  6
 [193]  6  3  5  5  6 10  1  2  2  7  7  5  8  4  4  0  5  4  5  9  7  1  3  2
 [217]  4  7  8  9  1  0  5  8  8  7  4  9  7  4  3  2  3  7  5  2  7  2  2  2
 [241]  5  9  2  3 10  3  7  7  3  3  6  1  0  9  7  3  1  8  9 10 10  2  7  1
 [265]  6  5  4  4  8  3 10  2  9  7  8  3  7  5  4  4  8  4  7  8  5  7  9  1
 [289]  3 10  3  8  6  6 10  8  2  9  9  7  6  5  1  8  9  2  6  7  9  2  4  4
 [313]  3  6  7  1  9  5  2  6  9  6  1  7 10  0  6  3  9  4  1  0  1  6  4  0
 [337]  8  6  5  5  4  3  8  5  5  9  1  6  5  3  2  3  7  4 10  2  7  1  6  8
 [361]  2  4 10  4  6  3  9  4  3  5  0  1  3  6  7  3  5  3  8  0  7  6  6  8
 [385]  2  8  1  6  1  6  9  6  2  4  8  5 10 10  8  3  7  3  2  1  5  0  4  7
 [409]  6  5  1  0  9  1  4  5  8  5  5  5  7  3  6 10  0  8  8  1  1  6  5  5
 [433]  3  4  7  0  1  6  8  6  9  1 10  4  5  1  7  8  3  2  9  4  8  6  7  2
 [457]  1  0  1  9  7 10  4  6 10  3  5  8  3  4  1  9  6  0  0  0  9  0  8  4
 [481]  8  5 10  2  3  2  2  7 10  9  1  5  8  6  8  9  2  1  1  7  6  3  6  8
 [505]  3  0  7  0  7  1  3  3  1 10  7  1  3 10  4  6  2  6  4  8  6 10  8  5
 [529]  4  3  7  5  2  6  8  1  5  5  5  4  2  4  7  6  6  3  5  4  7  4  8  2
 [553]  8  5  9  3  2  6  2  2  8  3  9  6  0  0  8  6  8  7  1  1  3  4  2  3
 [577]  2  9  8  7  2  6  6  2  2  2  7  5  2  1  9  4  2  8  7  1  9  2  8  3
 [601]  6  0  6  8 10  9  0  7  3 10  2  7  3  2  0  9  5  8  2  4  0  0  9  7
 [625]  8  8  1  8  2  9  5  8  9 10  7  7  7  9 10  8  3  0  8  4 10  2  7  7
 [649]  9  4  1  9  2  4  7  7  3  0  9  2  4  6  4  0  3  3  9  5  8  2  2  5
 [673]  8  8  8  8  4  9  3  6  7  3  5 10  1  1  5  9 10  5  5  6  3  1  9  2
 [697]  5  7  0  1  3  6  3  6  6  7  0  9  3  9  9  8  6  6  6  1  9  4  2  9
 [721]  4  6  8  7  0  3 10  2  1  8  4  9  1  4  5  5  2  9  5  7  7  6  9  9
 [745] 10  3  0  3  0  4  4  5  7  4  1  4  3  6  6  8  9  3  0  4  8  1  6  7
 [769]  2  3  8  4  1  8  7  5  9  1  5  0  7  6  2  3  3  5  7  7  4  9  8  7
 [793]  7  6  1  7  6  4  3  0  7  9  3  9  1  3  9  2  2  5  5  1  3  7  7  4
 [817]  5  4  3  9  6  4  2  3  1  4  5  9  7  4  5  3  0  5  7  1  9  9  0  1
 [841]  2  7  9  7  3  9  5  3  0  6  9  5  5  9  6  9  1 10  5  6 10  6  6  5
 [865]  6  3  7  1 10  4  3  5  7  7  1  6  8  7  4  6  3  5  5  3  7  1  1  0
```

Fig 1 : Vecteur généré

2. On fait ensuite apparaître ce vecteur sous forme d'histogramme pour mieux visualiser la fréquence d'apparition de chaque valeur. Pour cela on utilise la fonction *hist()*.

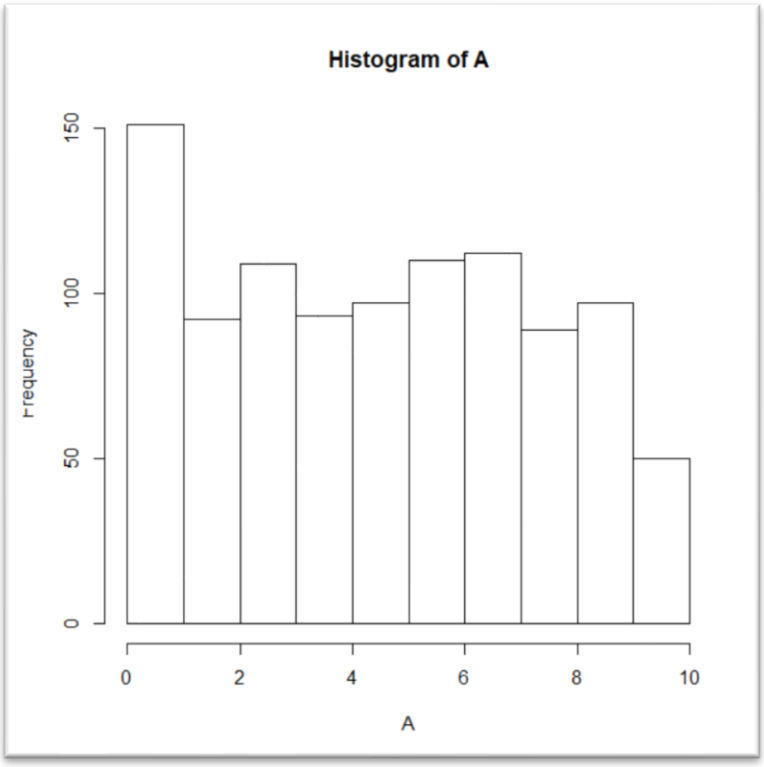

Fig 2 : Histogramme avec hist()

On remarque que la colonne la plus à gauche de l'histogramme (fig2) indique un nombre d'occurrences anormalement élevé. Cette anomalie vient du fait que cette colonne de l'histogramme rassemble les occurrences de 1 **et** de 0.

Pour passer outre ce problème, on peut utiliser l'argument optionnel *breaks* de la fonction *hist*() ou utiliser la fonction *barplot*(). *Voir fig 3*

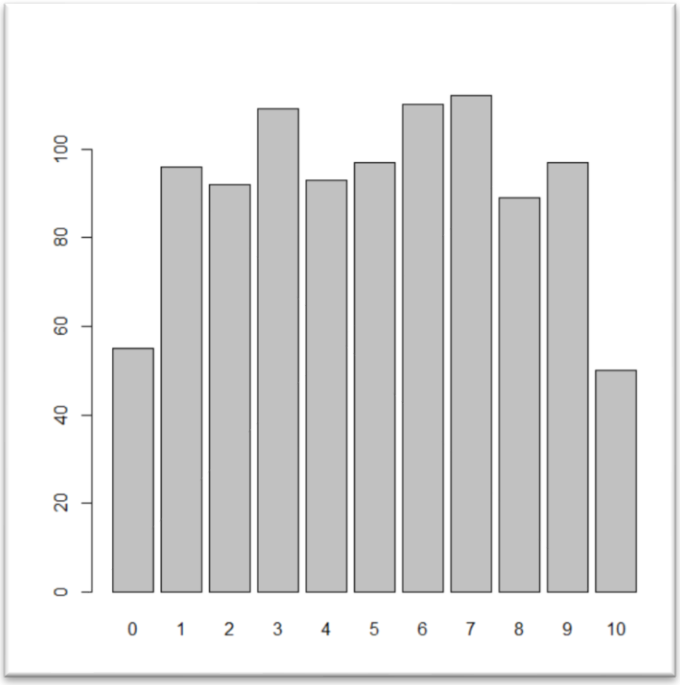

Fig 3 : Histogramme avec barplot

Il faut bien faire attention au fait que pour obtenir un histogramme tel que celui-ci, l'argument de *barplot*() doit être un tableau d'éléments.

```
> barplot(table(A))
```

3. La moyenne s'obtient en sommant les éléments du vecteur A puis en divisant par la taille du vecteur.

```
> b=sum(A)
> b
[1] 4993
> c=b/1000
> c
[1] 4.993
>
```

Pour obtenir la médiane des éléments du vecteur A on trie dans un premier temps les valeurs de A grâce à la fonction $sort()$. Ainsi il ne nous reste plus qu'à regarder la valeur de l'élément au milieu. On repère qu'il s'agit de la valeur 5.

Pour le mode, il suffit d'observer l'histogramme précédemment obtenu et de regarder les valeurs dont le nombre d'occurrences est plus élevé que les autres. Le mode est ici 7. Le nombre d'occurrences de chaque valeur est très proche donc le mode n'a pas vraiment de signification.

4. On obtient bien les mêmes valeurs de moyenne et de médiane en utilisant les fonctions $mean()$ et $median()$.
5. La génération aléatoire ne garantit pas que les valeurs de 0 à 10 aient le même nombre d'occurrences. Ceci implique que certaines valeurs apparaissent plus nombreuses que les autres, décalant la médiane par rapport à la moyenne.
6.

   a. Pour la variance : $\boldsymbol{Var(A) = mean(A^2) - mean(A)^2 = 8.42895}$

Pour l'écart-type :

   b.

```
> var(A)
[1] 8.437388
```

```
> sd(A)
[1] 2.904718
```

   c. On remarque une légère différence entre les valeurs calculées "manuellement" et les valeurs calculées directement grâce aux fonctions proposées par R. Cette différence est probablement due aux approximations de valeurs faites par le langage R dans les étapes intermédiaires.

PS : les sommes et moyennes effectuées pour le calcul sans var() et sd() ont été réalisées avec R.

B- Séries discrètes par fréquence.

1. On prend les valeurs du tableau fourni et on les représente sous forme de 2 vecteurs.

```
> notes=c(5,8,9,10,11,12,13,14,16)
> notes
[1]  5  8  9 10 11 12 13 14 16
> number=c(10,12,48,23,24,48,9,7,13)
> number
[1] 10 12 48 23 24 48  9  7 13
```

On affiche alors l'histogramme correspondant à l'aide de la fonction plot() :

```
> plot(notes,number,type="h")
```

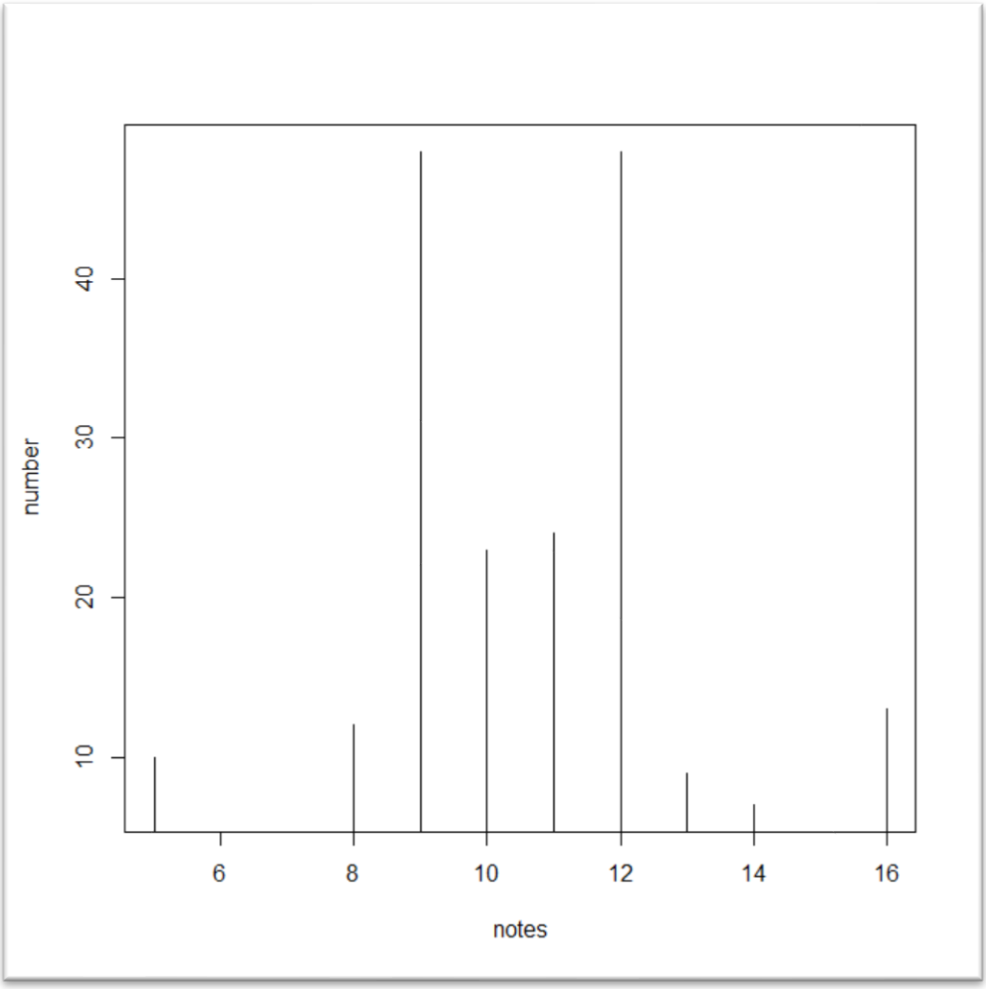

Fig 4 : Histogramme correspondant aux deux vecteurs

2. /! \ ici, ayant deux vecteurs, nous utilisons la méthode suivante pour pouvoir obtenir médiane et moyenne plus facilement :

```
> table1=rep(notes,number)
> table1
  [1]  5  5  5  5  5  5  5  5  5  5  8  8  8  8  8  8  8  8  8  8  8  9  9  9
 [26]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9
 [51]  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10
 [76] 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11
[101] 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12
[126] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
[151] 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 14
[176] 14 14 14 14 14 14 16 16 16 16 16 16 16 16 16 16 16 16 16 16
```

Mode	9,12
Moyenne	10.67526
Variance	5.878452
Écart-type	2.424552
Médiane	11

3. La distribution est dite bimodale car elle possède deux modes.

C-Distributions gaussiennes

1. On souhaite utiliser la fonction curv() afin de représenter la distribution des valeurs de QI en France (moyenne de 100 et variance de 225). Il faut préciser le domaine de définition de notre représentation grâce aux arguments optionnels from et to.

```
curve(dnorm(x,100,15), from = 0, to= 200)
```

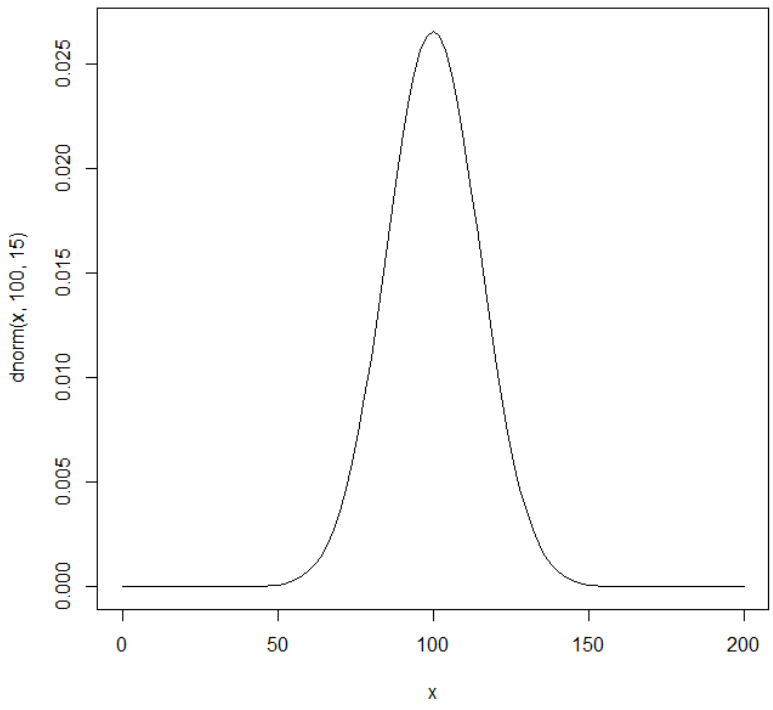

Fig 5 : Gaussienne associée à notre modélisation

2. On génère un échantillon de taille 100000 suivant une loi normale de moyenne 100 et d'écart-type 15.

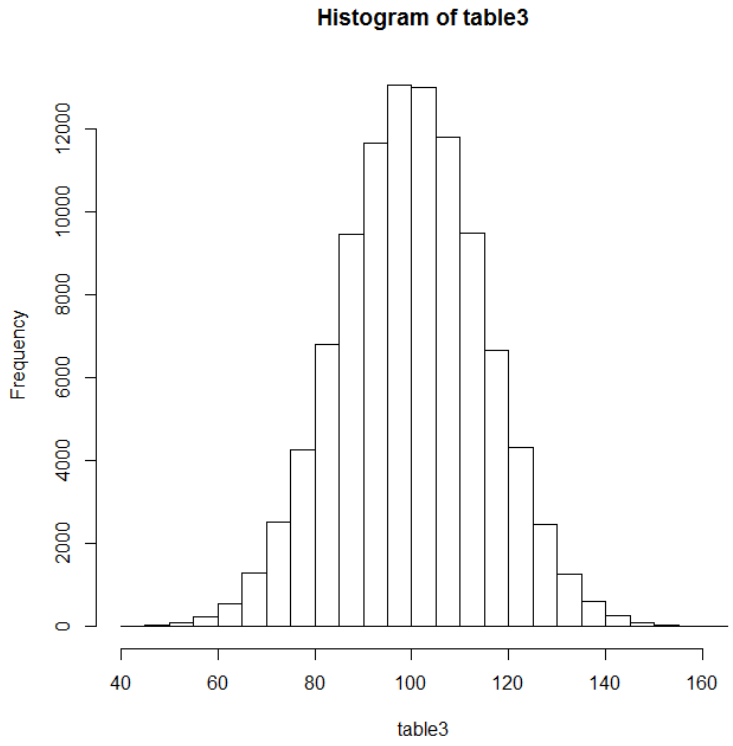

Fig 6: Histogramme associé à la modélisation gausienne.

On obtient cet histogramme gâce aux lignes de code suivantes :

```
> table3 = rnorm(100000, mean=100, sd=15)
> hist(table3)
```

Moyenne	100.0166
Variance	224.0603
QI inférieur à 60	$\frac{373}{100000} = 0{,}00373 = 0.373\%$
QI supérieur à 130	$\frac{2271}{100000} = 0{,}02271 = 2{,}271\%$
95% des valeurs autours de la moyenne de vos données	[70.70579,129.35537]

Détermination de l'intervalle du tableau :

```
> sum(table3 < 60)
[1] 373
> sum(table3 > 130)
[1] 2271
> quantile(table3, probs=c(0.025, 0.975))
     2.5%     97.5%
 70.70579 129.35537
```

On peut vérifier que l'intervalle obtenu correspond bien à l'intervalle dans lequel 95% des valeurs de QI se trouvent :

```
> sum(table3 > 70.7 & table3 < 129.35)
 [1] 94998
```

D- Analyse de QI

1.

	Echantillon de taille 10	Echantillon de taille 1000	Echantillon de taille 100000
Moyenne	104.8758	99.37854	100.0419
Ecart type	14.46032	15.17239	14.99065
Erreur type moyenne	4.572755	0.4797932	0.04740459
Intervalle de confiance à 95%	[95.91321 , 113.8384]	[98.43814, 100.3189]	[99.949, 100.1348]

```
> D10 = rnorm(10, 100, 15)
> D1000 = rnorm(1000, 100 , 15)
> D100000 = rnorm(100000, 100, 15)
> mean(D10)
[1] 104.8758
> mean(D1000)
[1] 99.37854
> mean(D100000)
[1] 100.0419
> sd(D10)
[1] 14.46032
> sd(D1000)
[1] 15.17239
> sd(D100000)
[1] 14.99065
> sd(D10)/sqrt(  10     )
[1] 4.572755
> sd(D1000)/sqrt(   1000    )
[1] 0.4797932
> sd(D100000)/sqrt(   100000    )
[1] 0.04740459
> DE10=sd(D10)/sqrt(mean(D10))
> DE1000=sd(D1000)/sqrt(mean(D1000))
> DE100000=sd(D100000)/sqrt(mean(D100000))
> mean(D10)-1.96*DE10
[1] 95.91321
> mean(D10)+1.96*DE10
[1] 113.8384
> mean(D1000)-1.96*DE1000
[1] 98.43814
> mean(D1000)+1.96*DE1000
[1] 100.3189
> mean(D100000)-1.96*DE100000
[1] 99.949
> mean(D100000)+1.96*DE100000
[1] 100.1348
```

2. On ouvre le fichier *malnutrition.csv* de la façon qui suit :

```
test=read.table('C:/Users/Thier/OneDrive/ISEP 2019/malnutrition.csv')
test=read.table('C:/Users/Thier/OneDrive/ISEP 2019/malnutrition.csv')
```

Moyenne	87.98
Écart-type	9.677611

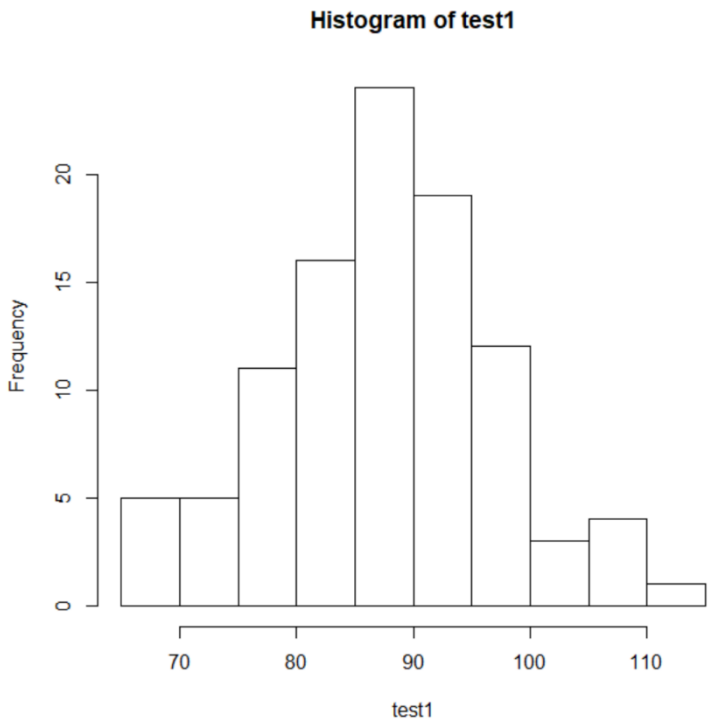

Fig 7 : Histogramme issu de malnutrition.csv

3.

	Malnutrition	Normal
Moyenne	87.98	100.0419
Écart-type	9.677611	14.99065
Intervalle de confiance à 95%	[86.08319 , 89.87681]	[99.949, 100.1348]

On peut voir qu'en moyenne le QI est plus bas, et que l'on s'écarte moins de la moyenne dans le cas mal nourri. Ainsi nous ne sommes pas dans cas de données réparties de manières de manière "étrange" [comme ci-dessous fig8] (cela conforte plus nos résultats).

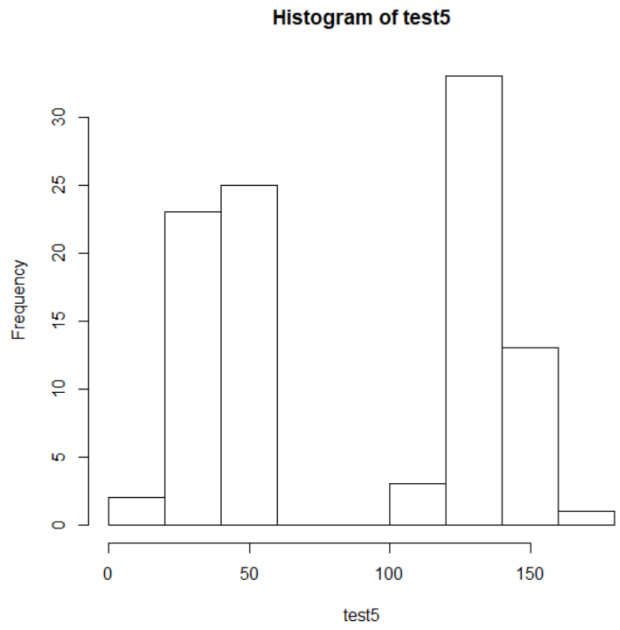

Fig 8 : Histogramme avec des valeurs déplacées vers la gauche et la droite.

La moyenne ici [fig.8] est toujours à 87.98, mais l'interprétation possible serait : la malnutrition a pour effet de rendre la moitié des gens idiots et de faire de l'autre moitié des génies.

L'intervalle de confiance à 95% montre que la moyenne des QI dans les 2 populations a de très fortes chances de ne pas être la même, même dans le cas où l'on prendrait d'autre échantillons (les intervalles de confiance sont disjoints).

Il semblerait donc que la malnutrition ait un effet néfaste sur le QI, même si cela n'est pas sûr à 100%.