

1 Paramètres de tendance centrale

Nous allons maintenant essayer de répondre – de préférence à l'aide d'un nombre – à des questions du type :

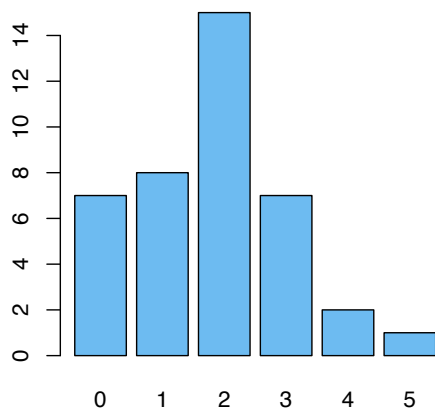
- « Où » se trouvent les valeurs de la série ?
- Autour de quelle valeur oscillent les observations ?
- Quel est le « centre » de la série ?

Les nombres qui permettent de répondre à ces questions sont appelés : *paramètres de position*, ou aussi : *paramètres de tendance centrale*, parce qu'ils permettent de localiser le centre de la répartition.

1.1 Le mode

Le *tableau des effectifs* ainsi que ses représentations graphiques permettent de voir que *la valeur la plus fréquente* est 2 : nous dirons que 2 est le *mode* de notre série, sa *valeur dominante*.

Le diagramme en bâtons



indique que notre série statistique est *unimodale*, qu'elle n'a qu'un seul mode.

R ne semble pas disposer d'une fonction qui permet de calculer directement les modes d'une série statistique. Nous allons donc écrire notre propre fonction :

```
modes ← fonction(xs) {
  txs ← table(xs)
  txs[txs == max(txs)]}
```

En appliquant cette fonction à la série `enfants` nous obtenons le résultat escompté

```
> modes(enfants)
2
15
```

1.2 La médiane

Dans notre cas, le mode 2 a aussi la propriété de subdiviser l'ensemble de nos observations en deux sous-groupes : les unités du premier groupe sont inférieures ou égales à 2, tandis que les individus du deuxième groupe sont supérieurs ou égaux à 2.

2 vérifie donc la condition de définition d'un autre paramètre statistique important : la *médiane*, qui précisément est une valeur telle que la moitié des observations soit inférieure ou égale à cette valeur.

R met à notre disposition la fonction **median** :

```
median(enfants)
[1] 2
```

Vérifions, à l'aide de quelques fonctions **R**, que 2 est bien la médiane de notre série.

En effet, considérons la série triée

```
senfants ← sort(enfants)
```

On peut déterminer les 20 premiers éléments de la série à l'aide de l'expression

```
senfants[1:20]
```

qui donne

```
[1] 0 0 0 0 0 0 0 1 1 1 1 1 1 1 2 2 2 2
```

et les 20 derniers éléments de la série sont alors

```
> senfants[21:40]
[1] 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 4 4 5
```

1.3 La moyenne

Quel est d'ailleurs le nombre d'enfants dans le quartier ? La fonction **sum**, appliquée à notre vecteur de données,

```
sum(enfants)
```

nous fournit la réponse

```
[1] 72
```

Dans notre quartier, le nombre d'enfants présents dans un ménage varie. Si maintenant il n'y avait aucune fluctuation aléatoire dans le nombre d'enfants, si chaque ménage avait le même nombre d'enfants, quel devrait être ce nombre pour que le nombre total d'enfants dans le quartier soit le même ? Le nombre de ménages du quartier étant donné par

```
length(enfants)
```

nous en déduisons la *moyenne* : $\frac{72}{40} = 1,8$, autre paramètre de position, qui est proche de la médiane et du mode.

R permet évidemment d'éviter ces calculs intermédiaires :

```
mean(enfants)
```

L'interprétation cependant de ce nombre est, dans notre contexte discret, plus difficile. Que pourrait bien signifier l'expression : 1,8 enfants ?

Adoptons un point de vue purement formel et retenons pour le moment que ce nombre, comme le mode et la médiane, permet de déterminer un « centre » de notre série statistique : les termes de la série, les observations, oscillent autour de ce centre.

Ainsi par exemple un quartier de moyenne 1,8 enfants par ménage sera considéré comme moins riche en enfants qu'un quartier de 2,25 enfants par ménage.

2 Paramètres de dispersion

Mais même si les observations ont tendance à se regrouper autour du centre, il est indéniable qu'elles s'écartent plus ou moins. Notre suite n'est pas constante – on pourrait même dire qu'elle serait constante s'il n'y avait pas les perturbations aléatoires.

On essaie alors de déterminer des *paramètres de dispersion* qui indiquent le degré de dispersion des observations autour du centre : est-ce que les observations s'en écartent beaucoup ou plutôt peu ?

2.1 L'étendue

S'il y a peu de variations, la différence entre les valeurs observées les plus élevées et les plus faibles dans notre ensemble de données sera petite.

Cette différence nous donne donc déjà un paramètre de dispersion très simple : c'est l'*étendue*.

La fonction **range**

```
range(enfants)
```

nous fournit l'encadrement

```
[1] 0 5
```

c'est-à-dire le minimum et le maximum de la série.

La fonction **diff**

```
diff(range(enfants))
```

nous fournit alors l'étendue

```
[1] 5
```

Définissons alors une fonction

```
spread ← fonction(xs){ diff(range(xs))}
```

de sorte que l'appel

```
spread(enfants)
```

nous donne le résultat 5.

2.2 Les 5 nombres de John Tukey

Nous venons de voir que les individus de la série appartiennent à l'intervalle $[0; 5]$, la première moitié des individus (c'est-à-dire `senfants [1:20]`) étant située dans l'intervalle $[0; 2]$, la seconde moitié (`senfants [21:40]`)

étant dans le sous-intervalle $[2; 5]$. La répartition des observations n'est donc pas uniforme.

Appliquons de manière récursive les calculs de médiane aux sous-suites précédentes. La répartition des éléments de la moitié gauche `enfants [1:20]` est décrite par

```
median(senfants[1:20])
```

ce qui donne 1 et la répartition de la moitié droite est donnée par

```
median(senfants[-(1:20)])
```

ce qui donne 2.5.

Nous obtenons encore une fois un nombre « imaginaire » – on ne peut parler de 2,5 enfants.

Examinons la structure de cette dernière série de 20 observations et subdivisons-la en deux sous-groupes de 10 ; le premier groupe de 10 éléments se termine par 2, tandis que le second groupe commence par 3 : la médiane est alors, par définition, la moyenne de ces deux entiers, donc 2,5.

Ces calculs, qui conduisent à une description plus fine de la série, sont résumés à l'aide de la fonction **fiveum**

```
fiveum(enfants)
```

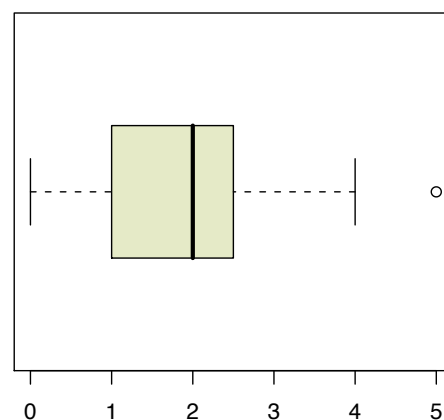
```
[1] 0.0 1.0 2.0 2.5 5.0
```

0.0 et 5.0 sont les extrêmes, 1.0 et 2,5 sont les nombres que l'on vient de calculer et 2.0 est la médiane.

On peut en déduire que 50 % environ des observations sont situées dans l'intervalle $[1, 2.5]$.

Un résumé visuel est donné par le diagramme « à moustaches » (*boxplot*, en anglais)

```
boxplot(enfants,
  col = hsv(0.18,0.20,0.9),
  horizontal = TRUE)
```

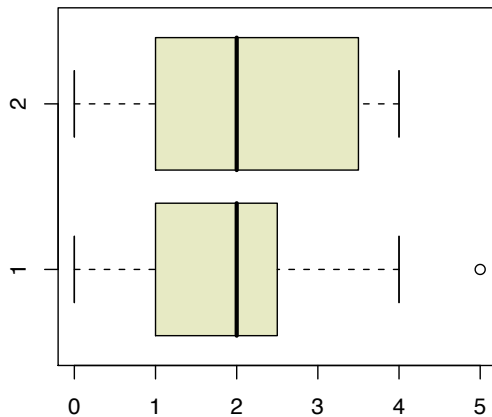


R nous suggère que l'observation 5 (le ménage ayant 5 enfants) est peut-être une anomalie.

Ces diagrammes constituent aussi un outil excellent de *comparaison*. Reprenons les données `enfnats2` du deuxième quartier :

```
boxplot(enfants, enfants2,
        col = hsv(0.18,0.20,0.9),
        horizontal=T)
```

Nous représentons les deux diagrammes « à moustaches » l'un au-dessus de l'autre



Les deux séries ont la même médiane, essentiellement les mêmes bornes, mais des distributions différentes : on pourrait supposer une composition sociologique différente des deux quartiers.

2.3 Les quartiles

Les *quartiles* définissent quatre sous-séries de même effectif.

2.4 Intervalle interquartile

R permet de calculer directement cet intervalle à l'aide de la fonction **IQR** (*interquartile range*)

```
IQR(enfants)
```

```
[1] 1.25
```

2.5 Écarts par rapport aux paramètres de position

On peut associer à notre série initiale la série des *écarts* par rapport à la *médiane* ou la *moyenne*. Ensuite, nous caractérisons cette nouvelle série à l'aide des paramètres de position précédents.

2.5.1 MAD

Commençons par étudier la série des écarts par rapport à la *médiane*

```
abs(enfants - median(enfants))
```

```
[1] 2 1 2 2 2 1 0 1 2 0 1 0 0 0 1 0 1 0 ...
```

La médiane de cette série s'obtient directement à l'aide de la fonction **mad** (acronyme de *median absolute deviation*)

```
mad(enfants, constant = 1)
```

```
[1] 1
```

l'option **constant** s'expliquant par des considérations qui ne nous intéressent pas pour le moment.

2.5.2 L'écart moyen

On peut aussi déterminer la moyenne de la série

```
abs(enfants - mean(enfants))
```

des écarts par rapport à la *moyenne* :

```
mean(abs(enfants - mean(enfants)))
```

```
[1] 0.95
```

Cette fonction n'existe pas dans **R** – excellente occasion de la définir soi-même :

```
mecart <- function(xs) mean(abs(xs - mean(xs)))
```

Nous pouvons appliquer notre fonction **mecart** comme une fonction prédéfinie

```
mecart(enfants)
```

```
[1] 0.95
```