

1 Loi de répartition

Revenons à notre série `enfants` : nous constatons qu'il y a un ménage (et un seul) ayant 5 enfants, ce qui nous amène à formuler quelques questions :

- Quelles sont les autres valeurs possibles, les autres *modalités* de notre *caractère*? Est-ce qu'il y a par exemple un ménage à 6 enfants?
- Est-ce que les ménages à un enfant sont plus nombreux que les ménages à trois enfants?

Nous nous intéressons donc à la distribution, la *répartition*¹ des ménages sur les différentes modalités.

1.1 Les modalités

La fonction `unique` permet de répondre à la première question en construisant le relevé de toutes les valeurs, de toutes les modalités apparaissant dans la série :

```
> unique(enfants)
[1] 0 1 4 3 2 5
```

Le nombre maximal d'enfants par ménage dans ce quartier est donc 5, ce qu'on peut vérifier directement à l'aide de la fonction `max`

```
> max(enfants)
[1] 5
```

Nous savons maintenant que dans ce quartier, les ménages ont 0, 1, 2, 3, 4 ou 5 enfants. Un rapide survol de notre série nous apprend que ces valeurs apparaissent avec des *fréquences*² différentes.

1.2 Effectifs et fréquences

1.2.1 Le tableau des effectifs

Pour répondre à la deuxième question, structurons un peu les données de notre série `enfants` en les regroupant suivant les modalités communes à l'aide de la fonction `table` :

```
> table(enfants)
enfants
 0  1  2  3  4  5
 7  8 15  7  2  1
```

qui construit le *tableau des effectifs*. Ce tableau décrit la *répartition* du caractère, c'est-à-dire le nombre d'observations par modalité du caractère : ainsi 7 ménages n'ont pas d'enfant, 8 ménages ont 1 enfant, ...

Visualisons cette répartition à l'aide d'un *diagramme en bâtons*³, chaque bâton, chaque barre ayant une hauteur proportionnelle à l'effectif correspondant⁴ :

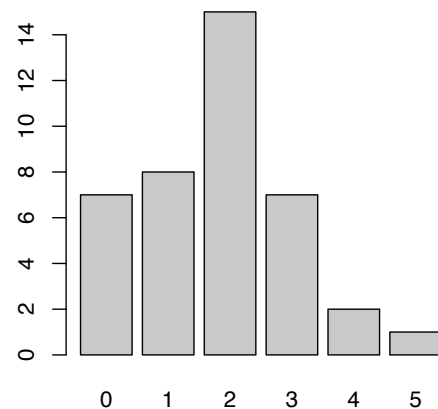
¹Verteilung

²Häufigkeiten

³Parfois appelé diagramme à tuyaux d'orgue

⁴Remarquons que `R` n'affiche aucun message ; en effet, aucun objet qu'on pourrait manipuler à l'aide des fonctions `R` n'a été construit.

```
barplot(table(enfants))
```



Le diagramme nous montre clairement que la distribution est étalée sur la gauche, que 2 est le nombre d'enfants le plus fréquent du quartier.

1.2.2 Proportions et pourcentages

L'ensemble des ménages de notre quartier se décompose donc en 6 groupes (on dit aussi : en 6 *facteurs*) dont les *effectifs* sont donnés par

```
effectifs ← table(enfants)
```

Pour comprendre l'importance *relative* de chaque groupe par rapport à la population totale, calculons le tableau des *fréquences relatives*

```
effectifs / length(enfants)
```

et en pourcent :

```
effectifs / length(enfants)*100
```

Nous obtenons le tableau :

```
enfants
 0  1  2  3  4  5
17.5 20.0 37.5 17.5 5.0 2.5
```

Remarquons qu'on pourrait remplacer l'expression `length(enfants)` par `sum(effectifs)`.

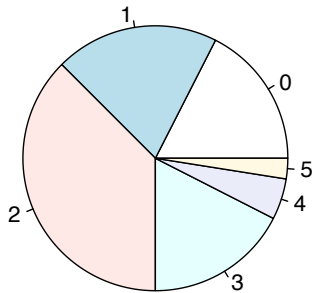
Ainsi, 5% des ménages de notre quartier ont 4 enfants. La fonction `prop.table` nous permet de calculer les proportions de manière plus directe

```
> prop.table(effectifs)*100
enfants
 0  1  2  3  4  5
17.5 20.0 37.5 17.5 5.0 2.5
```

Une représentation classique (peu appréciée des statisticiens professionnels) des proportions est le *diagramme en secteurs* où les différentes modalités sont représentées par des secteurs angulaires d'angles au centre proportionnels aux fréquences observées.

```
pie(effectifs)
```

Vous obtenez la figure (assez pâle) suivante



On verra plus loin comment modifier les couleurs de ces diagrammes en exprimant nos préférences à l'aide d' *options*.

1.2.3 Fréquences cumulées

On pourrait poser la question : « Quel est le nombre de ménages ayant au plus trois enfants ? »

La réponse est fournie par le tableau des effectifs : $7 + 15 + 30 + 37$, c'est-à-dire en appliquant la fonction `cumsum` au tableau des effectifs

```
cumsum(effectifs)
```

on obtient

```
0 1 2 3 4 5
7 15 30 37 39 40
```

La plupart des ménages ont donc au plus 2 enfants. Ce tableau est appelé tableau des *effectifs cumulés croissants* : il indique, pour chaque modalité possible x du caractère, le nombre d'individus ayant une modalité inférieure ou égale à x .

Le tableau des *fréquences cumulées croissantes* s'en déduit aisément en divisant par le nombre d'observations :

```
> cumsum(effectifs/length(enfants))
0 1 2 3 4 5
0.175 0.375 0.750 0.925 0.975 1.000
```

Remarquez que le tableau des effectifs et le tableau des effectifs cumulés croissants sont des informations équivalentes : nous venons de calculer le deuxième tableau à partir du premier, mais il est facile de calculer le premier à partir du second.

En effet, il suffit de calculer les différences successives des éléments du tableau des effectifs croissants :

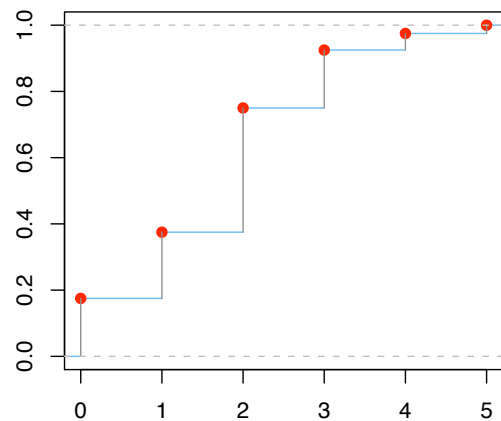
```
diff(cumsum(effectifs))
```

On recouvre le tableau `effectifs` privé de son premier terme.

La représentation graphique des fréquences cumulées se fait par l'intermédiaire de la fonction `ecdf`⁵

```
plot(ecdf(enfants))
```

⁵Acronyme de *empirical cumulative distributive function*



Nous avons obtenu une *fonction en escalier* dont les discontinuités correspondent aux modalités du caractère.

Ce graphique nous permet de faire des *interpolations* : 50 % des ménages du quartier ont un nombre d'enfants certainement inférieur ou égal 2.

Plus généralement, la fonction qui à une proportion p associe la modalité x de la série telle que la proportion d'observations inférieures ou égales à x soit p est la fonction **quantile**⁶.

Elle devrait donc, d'après le tableau des fréquences cumulées, associer à la proportion $p = 0,975$ à la modalité 2 – et effectivement

```
> quantile(enfants, 0.975, type = 1)
97.5%
4
```

L'*option* `type=1` s'explique par le fait qu'il y plusieurs possibilités de faire une interpolation⁷ entre les pourcentages qui n'interviennent pas dans le tableau cumulatif :

```
> quantile(enfants, 0.975)
97.5%
4.025
```

Ainsi, la fonction **quantile** est la réciproque de la fonction qui calcule les fréquences cumulées croissantes : elle transforme les proportions en des modalités du caractère. On a donc :

```
quantile(enfants, type = 1,
c(0.175, 0.375, 0.750, 0.925, 0.975, 1.000))
```

```
17.5% 37.5% 75% 92.5% 97.5% 100%
0 1 2 3 4 5
```

2 Infographie statistique

Les graphiques précédents diffèrent certainement des vôtres ; ils ont été construits en spécifiant soigneusement quelques *options graphiques*.

⁶Relisez cette définition au moins deux fois !

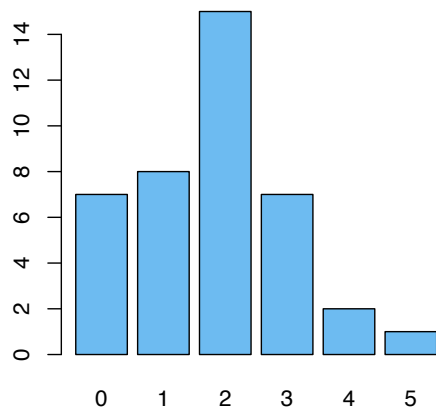
⁷Ce genre de subtilités montre bien que **R** s'adresse à un public très exigeant.

2.1 Couleurs

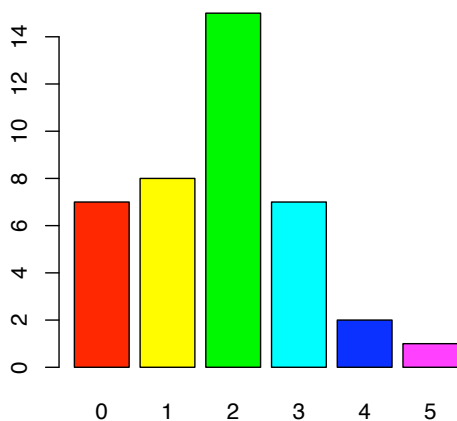
Si vous n'aimez pas le gris standard du diagramme en bâtons, vous pouvez changer la couleur en utilisant l'option graphique `col` :

```
barplot(table(enfants), col = "steelblue2")
```

ce qui donne le graphique nettement moins triste



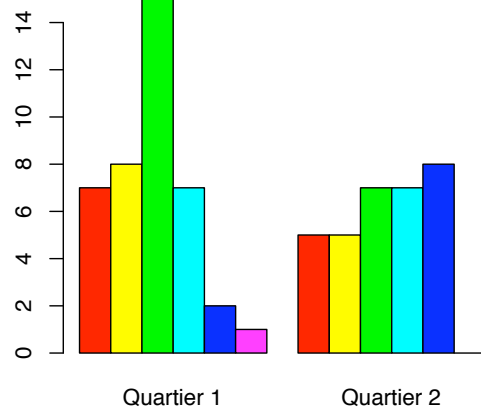
D'aucuns parmi vous préfèrent peut-être un diagramme encore plus gai, utilisant les couleurs de l'arc-en-ciel,



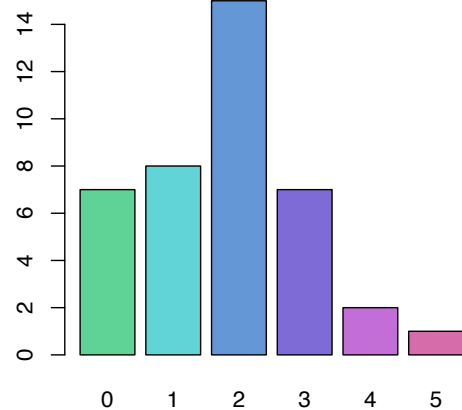
Le graphique précédent a été obtenu à l'aide de l'option `col = rainbow(6)`.

Mais attention : ces couleurs n'ont à priori aucune signification et peuvent donc empêcher une interprétation objective des données.

Il n'empêche : les couleurs facilitent parfois les comparaisons :



Une description très fine des couleurs s'obtient en utilisant le codage HSV⁸ :



Chaque couleur utilisée étant définie par la fonction `hsv` ; la première colonne par exemple est définie par l'expression `hsv(0.4, 0.6, 0.8)`, la seconde par `hsv(0.5, 0.6, 0.8)` etc.⁹

Comment communiquer à **R** cette séquence de couleurs ? En construisant un vecteur à l'aide de l'opérateur de concaténation `c`

```
col = c( hsv (0.4,0.6,0.8), hsv (0.5,0.6,0.8),
        hsv (0.6,0.6,0.8), hsv (0.7,0.6,0.8),
        hsv (0.8,0.6,0.8), hsv (0.9,0.6,0.8))
```

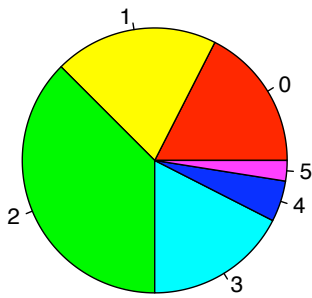
Colorions aussi le diagramme en secteurs

```
pie(table(enfants), col = rainbow(6))
```

ce qui donne

⁸Hue Saturation Value

⁹Vous avez même la possibilité de spécifier l'exposant de correction γ et le paramètre de transparence α .

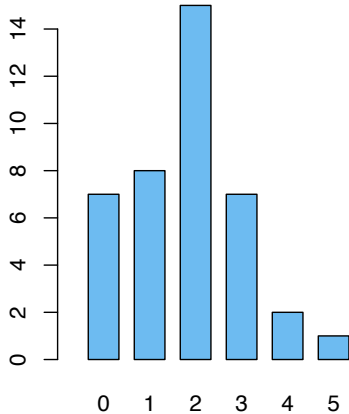


2.2 Largeur des colonnes

On peut considérer que pour un diagramme à barres, les colonnes sont trop larges; en précisant notre choix esthétique à l'aide de quelques options

```
barplot(table(enfants),
        width = 0.4,
        xlim = c(0,5),
        space = 0.5,
        col = "steelblue2")
```

nous obtenons des colonnes plus minces



Expliquons les autres options utilisées :

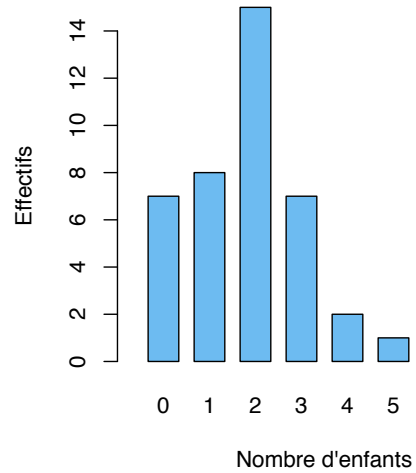
- **width** décrit la largeur des colonnes
 - **xlim** décrit l'ensemble des abscisses
 - **space** décrit l'espace entre les colonnes
- L'option **xlim** a été spécifiée à l'aide d'un vecteur à deux éléments (en utilisant l'opérateur de concaténation **c**).

2.3 Étiquettes

Remarquez que le graphique n'explique pas la signification des nombres qui figurent sur les axes. Les options **xlab** et **ylab** permettent de remédier à cet inconvénient :

```
barplot(table(enfants),
        width = 0.4,
        xlim = c(0,5),
        space = 0.5,
        col = "steelblue2",
        xlab = "Nombre_d'enfants",
        ylab = "Effectifs ")
```

et nous obtenons le graphique suivant :



Si vous voulez supprimer l'étiquette sur l'axe des abscisses, vous écrivez : **xlab=""**

2.4 Pareto

Lorsque les modalités sont ordonnées par effectifs décroissants, on obtient un diagramme dit *de Pareto*

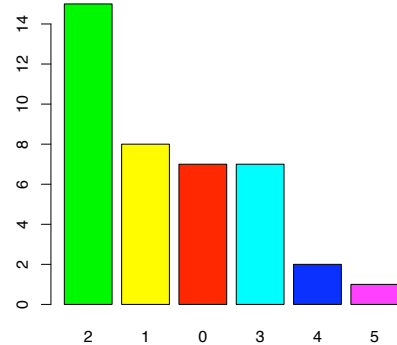
```
barplot(sort(table(enfants), decreasing = TRUE))
```

Essayons de remplacer le graphique grisâtre par un graphique plus gai - mais en conservant le codage par les couleurs utilisé plus haut. Nous construisons d'abord la *permutation*

```
oe ← order(table(enfants), decreasing = TRUE)
```

ensuite nous appliquons cette permutation au vecteur **table(enfants)** et au vecteur des couleurs **rainbow(6)**

```
barplot(table(enfants)[oe], col = rainbow(6)[oe])
```

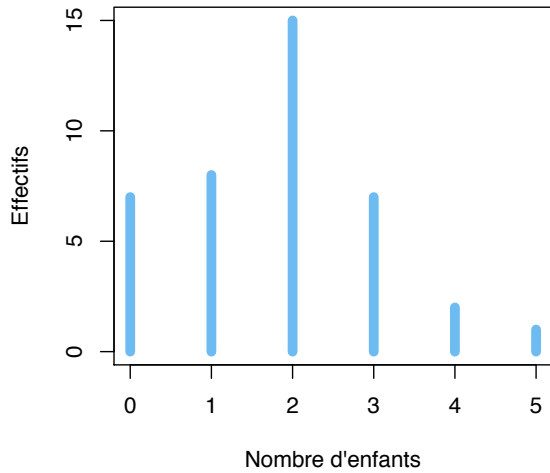


2.4.1 Autres possibilités

R offre encore de nombreuses autres possibilités (objets et options) de construire des graphiques; la fonction graphique la plus générale est **plot**. Utilisons cette fonction pour recréer un diagramme en bâtons :

```
plot(table(enfants),  
      col = "steelblue2", lw = 7,  
      xlab = "Nombre_d'enfants",  
      ylab = "Effectifs ")
```

L'option `lw=7` fixe l'épaisseur¹⁰ du trait. Nous obtenons alors le graphique



¹⁰line width