Décrire et inférer. Une introduction intuitive à l'analyse quantitative avec R

Notes de cours

Alexandre Blanchet
McGill University

mars 2018

Résumé

Ce document contient les notes de cours de mon enseignement de l'introduction aux statistiques. Il vise principalement les étudiants en science politique, mais pourrait aussi servir aux étudiants de tous les domaines qui utilisent les méthodes quantitatives. Les exemples utilisés sont cependant tirés de la vie politique. Le document couvre d'abord l'introduction à R et les manipulations de bases avec le logiciel. Une emphase particulière est ensuite portée sur la régression linéaire. Les autres types de régression, notamment les régressions logistiques et multinomiales, seront ajoutées éventuellement. Finalement, le document aborde les questions liées à l'inférence statistique. Tout le code R utilisé est reproduit dans ce document et est facilement reproductible. L'accent est mis sur la compréhension intuitive des méthodes statistiques elles-mêmes plutôt que sur l'aspect «computationnel» de leur mise en application. Cependant, cet aspect est vu en parallèle afin d'aider les lecteurs à saisir progressivement comment sont mises en application les méthodes statistiques abordées. Ce document est une version préliminaire et les commentaires et suggestions sont bienvenus ! Je n'ai aucun problème à ce que ce matériel soit utilisé pour votre propre enseignement, en autant que les crédits me soient adéquatement attribués. Cette version est en date du **7 mars 2018**.

Table des matières

1 Objectifs **3**
1.1 Les logiciels 4

2 Introduction générale à R **5**
2.1 Les packages 6
2.2 Le «working directory» 6
2.3 Quelques manipulations initiales avec R 7
2.4 Une syntaxe qui roule : une bonne méthode de travail 12
2.5 Charger des données 13
2.6 Voir les variables 14
2.7 Tableau croisé 16

3 Décrire **18**
3.1 Une équation pour décrire une variable 18
3.2 Les types de variables et éviter de faire n'importe quoi 21
3.3 Décrire une relation entre des variables 23
3.4 La régression linéaire simple avec une variable dichotomique 25
3.5 La régression linéaire simple avec une variable continue 33
3.6 La régression linéaire avec plusieurs variables indépendantes 36
3.7 La régression linéaire, un exemple avec des données réelles 39
3.8 Les régressions logistiques et multinomiales 49

4 Inférer **50**
4.1 L'inférence statistique, quelques notions théoriques 51
4.2 L'échantillonnage 54
4.3 Du sondage à la population 62

Références **70**

1 Objectifs

L'objectif de ce document est d'abord de démystifier les approches statistiques. La plupart d'entre vous commencez votre maîtrise ou votre doctorat et aurez peut-être à apprendre une variété de méthodes statistiques pour mener à bien vos projets de recherche. Lorsque l'on commence à apprendre ces méthodes, il est facile d'être intimidé par des équations qui semblent bien complexes, ou par des logiciels qui peuvent de prime abord être difficiles à comprendre. Peut-être êtes-vous un peu plus avancé et avez déjà commencé à apprendre des méthodes plus complexes. Si c'est votre cas, cette séance vous sera utile aussi. Quand on commence à tenter de faire des choses plus complexes, il n'est pas rare que l'on se sente arriver en territoire inconnu et qu'on perde de vue l'objectif de base qui demeure pourtant simple.

Les statistiques ne sont pas de la magie et elles ne doivent jamais se substituer à votre jugement, étant entendu que celui-ci soit fondé sur une bonne compréhension de ce que les chiffres vous disent. Le fait que quelque chose soit «statistiquement significatif» ne constitue en aucun cas une sorte de sceau certifiant la véracité de ce que vous proposez. Quand on commence à faire des analyses statistiques, on est souvent obnubilé par les étoiles (ces fameuses étoiles, qui sont généralement le symbole associé à la significativité statistique). Il est excitant d'appuyer sur un bouton et de voir des étoiles apparaître à côté des chiffres. La chose entraîne une sorte de «boucle béhaviorale» : on appuie sur un bouton et on obtient une récompense. La plupart d'entre-nous y succombons. Je l'ai fait au début et je le vois très souvent chez mes collègues plus jeunes qui commencent.

La réalité est que tous les modèles statistiques, les simples comme les plus complexes, ont deux objectifs généraux : décrire et inférer. Nous voulons *décrire* des données quelconques et, si nécessaire (ce qui est le cas la plupart du temps puisque nous utilisons souvent des échantillons) en *inférer* quelque chose à une population plus large. En tout temps donc, gardez à l'esprit ces deux notions, parce que c'est systématiquement ce que toute forme de d'analyse statistique cherche à faire.

Nous utiliserons ici le logiciel R qui est gratuit et de plus en plus utilisé en science politique. Je vous suggère aussi de télécharger R Studio qui est une interface très utile facilitant l'utilisation de R. Ceci étant, ce cours ne porte pas spécifiquement sur R. Je vais d'abord vous donner quelques notions de bases et expliquer la syntaxe au fur et à mesure pour que vous puissiez la comprendre, mais je ne me lancerai pas dans de grandes explications concernant le logiciel lui-même. La raison est tout simplement que nous n'apprenons rien sans avoir la *motivation* de le faire. Dans le cas de R (et de tous les logiciels statistiques), il est donc tout à fait inutile de se concentrer sur son utilisation sans en même temps vous donner des idées sur ce que vous pourriez faire avec le logiciel.

L'apprentissage de la syntaxe (l'aspect «computationnel» des méthodes quantitatives), se fait forcément par la pratique et celle-ci vient de la nécessité. Nous apprenons à faire quelque chose parce que nous avons un objectif en tête. Je vous donnerai ici les bases pour vous aider à vous repérer au travers de l'aspect computationnel des méthodes statistiques, mais lorsque vous serez seul dans votre bureau, prêt à faire une certaine analyse, vous aurez

forcément à chercher sur internet. C'est tout à fait normal.

Je me concentrerai donc à vous donner des éléments de bases de l'analyse statistique et, au passage, nous verrons comment les faire avec R. Vous verrez alors les deux éléments conjointement et, lentement mais sûrement, la logique derrière la syntaxe de R vous apparaîtra de plus en plus claire. L'objectif central demeure cependant que vous développiez une compréhension plus intuitive des méthodes statistiques que je vais vous présenter. L'apprentissage de la syntaxe est donc un objectif secondaire, mais parallèle. Il ne fait pas de sens d'apprendre à utilisr une syntaxe sans comprendre ce qu'elle fait, et il fait de moins en moins de sens d'apprendre des méthodes statistiques sans savoir comment les appliquer dans la pratique. Il reste que la compréhension des méthodes demeure la priorité, parce que c'est cette compréhension qui vous motivera à les appliquer dans vos recherches et surtout, c'est cette compréhension qui vous permettra de bien le faire.

Si R vous intéresse, vous pouvez consulter le site web de Quick-R qui contient toutes les informations nécessaires pour vous aider à démarrer et aller plus loin avec le logiciel. Ceci étant, toute la syntaxe nécessaire pour reproduire les analyses présentées ici est inclue dans le document dans des blocs ombragé et le résultat produit par R sera précédé de ## pour chaque ligne. Vous pourrez donc vous y référer au besoin et adapter la syntaxe pour mener vos propres analyses.

1.1 Les logiciels

Une syntaxe est une série de commandes que l'on demande à un logiciel d'effectuer. Certains logiciels mettent un accent particulier sur les menus déroulants (notamment SPSS), alors que d'autres offrent un bel équilibre en la possibilité d'utiliser une syntaxe complète et intuitive, tout en ayant accès à des menus déroulant pour les tâches les plus communes (Stata). Tous les logiciels ont besoin d'une forme ou d'une autre de syntaxe parce qu'il serait tout simplement impossible d'avoir des menus déroulants pour toutes les analyses statistiques existantes. À l'inverse, la syntaxe offre des possibilités infinies. R n'offre pas de menus déroulant, mais il a l'avantage d'être gratuit et aussi complet (et même plus complet) que les logiciels payants.

Le premier avantage de la syntaxe est qu'elle est reproductible. On peut donc «rouler» une syntaxe sur des données et répéter nos analyses antérieures sans altérer les données initiales. Dans un univers où la reproductibilité de la recherche est de plus en plus important, l'utilisation de la syntaxe deviendra rapidement impératif. Par ailleurs, même les logiciels qui offrent des menus déroulant fonctionnent aussi à base de syntaxe afin que l'utilisateur puisse répéter facilement ses analyses au besoin et, comme mentionné plus haut, parce qu'il est impossible d'inclure en menus déroulant toutes les analyses statistiques possibles.

À mon avis, SPSS devrait être laissé de côté par quiconque commence à utiliser des méthodes quantitaves. Il s'agit d'un bon logiciel, mais il a été créé il y a très longtemps et pour cette raison, sa syntaxe n'a jamais réellement été conçue pour être utilisée directement. En conséquence, le langage syntaxique de SPSS est réellement pénible à utiliser. Par ailleurs, SPSS est incroyablement dispendieux.

Stata est un excelent logiciel et sa syntaxe est très complète et intuitive. Stata a aussi de bons menus déroulants qui peuvent être très utiles pour nous apprendre à utiliser la syntaxe puisque l'on peut faire une analyse d'abord en utilisant le menu déroulant, puis regarder la syntaxe produite par Stata. Il faut cependant payer la licence pour l'utiliser (entre 400$ à 1000$ suivant la version et le statut professionnel), ou ne l'utiliser que sur les postes informatiques disponibles dans les laboratoires universitaires.

R est aussi puissant que Stata (voir plus), et il est gratuit. Par ailleurs, il y a une très grande communeauté d'utilisateurs de R et il existe donc beaucoup de «packages» qu'il est possible de télécharger gratuitement pour ajouter des fonctions à R (Stata offre aussi cette possibilité, mais il y en a beaucoup moins). En conséquence, R est toujours en avance sur Stata et il permet de faire des analyses qui n'ont pas forcément encore été implémentées dans Stata. Le seul désavantage de R par rapport à Stata est que sa syntaxe est un peu plus difficile à apprendre, mais une fois qu'on la maîtrise, il n'y a vraiment plus aucune raison de s'ennuyer Stata.

Au final, l'apprentissage de n'importe quel logiciel implique forcément beaucoup d'essais-erreurs et des consultations fréquentes de notre ami Google. Stata est un peu plus facile à apprendre, mais puisqu'il faut apprendre un nouveau logiciel, aussi bien fournir le petit effort supplémentaire pour apprendre R qui est aussi complet, plus souple, gratuit, et qui offre une communeauté d'utilisateurs très réactive.

2 Introduction générale à R

Le logiciel R est gratuit et très puissant. Il est littéralement possible de faire toutes les analyses statistiques imaginables avec R, et cela est notamment dû au fait qu'il y a une très large communauté d'utilisateurs de R qui produisent des «packages» qui ajoutent des fonctions au logiciel. Le désavantage de R est que la courbe d'apprentissage est plus difficile qu'avec d'autres logiciels parce que R n'offre pas de menus déroulants. Cependant *après* avoir installé R sur votre ordinateur, il est possible d'installer R Studio qui est aussi gratuit.R Studio n'est pas un logiciel qui fonctionne seul, il nécessite que la base de R soit installée sur votre ordinateur. R Studio est une interface qui va «par-dessus» R et facilite un peu son utilisation. Il faut donc préalablement avoir installé R, puis on installe R Studio. Une fois cela fait, on peut utiliser R Studio.

R Studio propose par défaut quatre panels qui peuvent être positionnés à votre guise dans les préférences du logiciel. Si vous ne modifiez pas les préférences, le panel où nous écrivons la syntaxe est en haut à gauche. Sous ce panel, en bas à gauche, se trouve la «console» de R, qui renvoie les résultats des commandes que nous passons à R via notre syntaxe. En haut à droite se trouve le panel «Environnement» qui liste tous les objets chargé en mémoire dans R. En bas à droite se trouve le panel des fichiers, figures, packages (et autres) où seront affichés les figures et où nous pourront charger et installer des packages.

2.1 Les packages

R vient par défaut avec une série de commandes très utiles, mais il existe aussi beaucoup de «packages» qui ajoutent des fonctions au logiciel. Les packages doivent être installés sur votre ordinateur (et souvent ré-installés lorsque vous mettez à jour votre version de R), mais ensuite vous n'avez pas à les installer de nouveau. Cependant, si vous désirez utiliser les fonctions d'un package, il faut systématiquement le charger dans votre session R. R fonctionne ainsi parce que si tous les packages installés sur votre ordinateurs étaient systématiquement actifs sans avoir à les charger, R deviendrait de plus en plus lourd et lent parce qu'il exigerait énormément de mémoire RAM. Donc, on *installe* d'abord un package quand on veut pouvoir utiliser ses fonctions. Cette installation ne se fait qu'une seule fois (ou lors des mises à jour), mais ensuite on doit *charger* à chaque session les packages que l'on veut utiliser.

Par exemple, le package *ggplot2* est très utilie pour faire des graphiques. Si je ne l'ai jamais installé sur mon ordinateur, je peux le faire en cliquant sur l'onglet «packages» du panel en bas à droite, puis «install», puis je peux taper le nom du package ("ggplot2") dans le menu. R Studio cherchera la package correspondant dans la banque de package accessible via internet. En cliaquant sur install, R Studio installera le package. Alternativement, on peut aussi procéder directement par la syntaxe comme suit avec la commande "install.packages("NOM DU PACKAGE")", donc dans notre cas *install.packages("ggplot2")*.

Une fois que le package est installé, il apparaîtra dans la liste «System Library» sous l'onglet «package» du panel en bas à droite. On peut simplement cliquer sur la case correspondant au package souhaité pour l'activer, ou encore mieux le faire via la syntaxe comme ceci :

```
library("ggplot2")
```

Notez que vous ne pouvez pas utiliser les fonctions d'un package s'il n'est pas chargé dans votre session. Pour cette raison, si vous écrivez un script qui utilise des fonctions qui viennent de certains packages, la bonne pratique est de charger ceux-ci dans votre syntaxe elle-même avant d'utiliser les fonctions. Autrement, vous aurez un message d'erreur et votre script ne pourra pas rouler du début à la fin. Il peut donc être pratique au début d'utiliser les menus et les cases cliquables, mais *je vous suggère fortement* de copier et coller les lignes de code produites par ces actions dans votre syntaxe elle-même afin que vous contruisiez progressivement un script qui pourra ensuite être roulé du début à la fin sans erreurs. Une bonne méthode de travail et des syntaxes «propres» vous sauverons beaucoup de temps.

2.2 Le «working directory»

Comme tous les logiciels statistiques, R a en mémoire un endroit sur votre ordinateur qui est le «working directory». C'est à cet endroit que R placera par défaut tout ce que vous sauvegarderez durant votre session. Par défaut, R utilisera le dernier working directory qu'il a en méoire. Vous pouvez à tout moment changer le working directory comme ceci :

```
setwd("~/Google Drive/Cours_LACPOC") # setwc() pour set working directory
# puis on place le lien vers le fichier souhaité dans la parenthèse.
```

On peut passer «par-dessus» le working directory en mémoire à un moment donné en référant au lien complet vers un fichier que l'on souhaite utiliser (ou dans lequel on souhaite sauvegarder quelque chose), mais l'avantage d'établir le working directory approprié est que R assumera alors que ce à quoi vous référez s'y trouve. Il est alors possible d'utiliser seulement le nom du fichier auquel on réfère (plutôt que le lien complet vers celui-ci). Il est donc de bonne pratique d'établir au début d'une syntaxe le working directory que vous souhaitez utiliser. Cela vous évitera d'avoir à écrire constement les liens complets vers les fichiers sur votre ordinateur.

2.3 Quelques manipulations initiales avec R

Pour vous aider à mieux saisir ce que nous ferons durant le cours et pour que vous ayez un petit guide de référence lorsque vous serez seul à votre bureau, je vais ici procéder à quelques manipulations de bases avec R. La syntaxe (ou les commandes) apparaîtront dans des blocs ombragé et le résultat produit par R apparaîtra sous ces blocs sur des lignes précédés de deux dièzes. Les annotations *à l'intérieur* de la syntaxe seront précédées d'un seul dièse. Cela est nécessaire afin que R sache qu'il s'agit d'une annotation et non pas de commandes qu'il doit exécuter.

Commençons doucement en demandant à R de renvoyer le chiffre π.

```
# Ceci est une autre annonation. Nous allons entrer la commande "pi"
pi # Ceci est une autre annotation que je place à côté de la commande.
```

```
## [1] 3.141593
```

La commande «pi» est la ligne de commande et R nous renvoie le résultat sous le bloc ombragé sur la ligne précédée par deux dièses. Nous pourrions aussi demander à R de nous renvoyer toutes les couleurs qu'il a en mémoire. R contient par défaut 657 couleurs dans l'objet *colors()* qui sont utilisées notamment pour faire les graphiques. Or, puisqu'il y en a 657, nous allons simplement lui demander de nous donner seulement les premières en utilisant la commande *head()*, et en insérant *colors()* dans la parenthèse de la commande *head()* :

```
head(colors())
```

```
## [1] "white"          "aliceblue"      "antiquewhite"   "antiquewhite1"
## [5] "antiquewhite2" "antiquewhite3"
```

Ici, le premier élément contenu dans l'objet *colors()* est "white", le cinquième élément est "antiquewhite2" et le sixième "antiquewhite3". Si nous n'avions pas inséré *colors()* à l'intérieur de la commande *head()*, R nous aurait renvoyé la liste complète des 657 couleurs, ce qui aurait été interminable.

Nous pouvons aussi créer des objets qui peuvent contenir tout ce que l'on veut. Nous

pouvons nommer ces objets de n'importe quelle manière. Créons ici l'objet «fadeldwill» qui contiendra la valeur de 1 :

```
fadeldwill <- 1
```

Nous avons ici créé l'objet *fadeldwill* et nous lui avons assigné la valeur de 1. Le signe <- est équivalent à =, ou peut être lu comme «insère ce qui est à droite dans l'objet à gauche de la flèche». L'objet *fadeldwill* est maintenant en mémoire, mais R ne renvoie rien parce que nous ne lui avons pas demandé. Faisons-le.

```
fadeldwill
```

```
## [1] 1
```

Nous voyons que *fadeldwill* ne contient qu'un seul élément ([1]) qui est la valeur 1.

Nous pourrions changer le contenu de *fadeldwill* pour "abc" au lieu de 1. Notez que je commence ici à annoter le contenu de la syntaxe en faisant précéder toute annotation d'un dièse afin que R sache qu'il ne s'agit pas de code qu'il doit exécuter. Les annotations sont très utiles pour clarifier une syntaxe et mieux s'y retrouver plus tard. Pour ceux qui sont familiers avec Stata, c'est l'équivalent de mettre une étoile (*) dans la syntaxe.

```
fadeldwill <- "abc" # Pour mettre abc dans fadeldwill.
# Les guillements sont nécessaires puisque abc n'est pas numérique.
fadeldwill # Pour demander à R de nous dire ce que contient fadeldwill.
```

```
## [1] "abc"
```

Nous pourrions vouloir que *fadeldwill* contienne plutôt une série de chiffres (disons : 1, 2, 3, 4, 5). Il faudra alors utiliser la fonction *c()*, «c» pour «concatenate», et on insère ce que l'on veut dans la parenthèse, avec des virgules entres les éléments.

```
fadeldwill <- c(1, 2, 3, 4, 5)
fadeldwill
```

```
## [1] 1 2 3 4 5
```

Nous pourrions aussi vouloir mettre autre chose que des nombres dans *fadeldwill* :

```
fadeldwill <- c("A", "B", "C", "D", "E")
fadeldwill
```

```
## [1] "A" "B" "C" "D" "E"
```

Jusqu'à présent, nous avons systématiquement *remplacé* le contenu de *fadeldwill* par du nouveau contenu. Nous pourrions cependant vouloir lui *ajouter* un nouvel élément à ce qu'il contient déjà. Ajoutons la lettre F à la suite de lettres que contient *fadeldwill* :

```
fadeldwill <- c(fadeldwill, "F") # Ici, nous disons simplement que l'objet
# fadeldwill contient fadeldwill et la lettre F. Donc, ce qu'était déjà
# l'objet fadeldwill (soit "A", "B", "C", "D", "E") et on y ajoute F.
fadeldwill
```

```
## [1] "A" "B" "C" "D" "E" "F"
```

Nous pourrions aussi vouloir *retirer* un élément de *fadeldwill*. Disons que nous voulions retirer C :

```
fadeldwill <- fadeldwill[-3] # Nous voulons que fadeldwill demeure identique
# mais en excluant "C", qui est la troisième valeur, d'où le [-3].
fadeldwill
```

```
## [1] "A" "B" "D" "E" "F"
```

Nous aurions aussi pu vouloir retirer A et F :

```
fadeldwill <- fadeldwill[-c(1, 5)] # Nous retirons le premier (A) et le
# cinquième élément (F, devenu cinquième après le retrait de C).
# Nous devons les mettre dans un c() parce qu'il y a plus d'un élément
# à retirer
fadeldwill
```

```
## [1] "B" "D" "E"
```

Nous pourrions aussi vouloir que R nous renvoie seulement le deuxième élément contenu dans *fadeldwill* :

```
fadeldwill[2] # Nous demandons à R de donner le 2e élément de fadeldwill
```

```
## [1] "D"
```

R nous renvoie bien la lettre D, qui est maintenant le deuxième élément dans notre objet.

Nous pourrions aussi vouloir créer l'objet *fadeldwillidou* qui contiendrait seulement le premier et le troisième élément de fadeldwill :

```
fadeldwillidou <- fadeldwill[c(1, 3)]
fadeldwillidou
```

```
## [1] "B" "E"
```

Encore une fois, si on veut *ajouter* quelque chose à un objet qui existe déjà, il faut que l'objet lui-même soit inclu à droite de la flèche. Par exemple, si nous voulons ajouter A à *fadeldwillidou* qui contient déjà B, E, il faut impérativement inclure *fadeldwillidou* dans la parenthèse *c()* à droite de la flèche.

```
fadeldwillidou <- c(fadeldwillidou, "A")
```

Si l'on inclut pas l'objet lui-même à droite de la flèche, alors son contenu sera remplacé par ce qui est à droite de la flèche, sans préserver le contenu préalable. Par exemple, ici nous remplaçons complètement le contenu de *fadeldwillidou* (B, E, A) par les nombres de 1 à 10.

```
fadeldwillidou <- c(1,2,3,4,5,6,7,8,9,10)
fadeldwillidou
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Pour le meilleur et (surtout) pour le pire, le recodage est très certainement ce qui demande souvent le plus de travail lorsque nous faisons des analyses statistiques. Souvent, les variables ne sont pas codées comme on voudrait qu'elles le soient, ou il faut «nettoyer» les données brutes de sorte à les rendre analysables. Nous verrons cela plus en profondeur un peu plus loin, mais pour l'instant, voici quelques manipulations usuelles qu'il vous sera utile de comprendre.

Nous avons l'objet *fadeldwill* qui contient B, D, E. Imaginons que nous voulions remplacer E par Z. Nous pouvons le faire comme ceci.

```
fadeldwill
```

```
## [1] "B" "D" "E"
```

```
fadeldwill[fadeldwill == "E"] <- "Z"
# Remplace les valeurs de fadeldwillidou qui sont égales à "E"
# et remplace les par "Z".
fadeldwill
```

```
## [1] "B" "D" "Z"
```

Nous pourrions maintenant vouloir changer les valeurs de certains chiffres de *fadeldwillidou* qui contient 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Disons que nous veulons que 2 soit remplacé par 1.

```
fadeldwillidou[fadeldwillidou == 2] <- 1
# Remplace les valeurs de fadeldwillidou qui sont égales à 2
# et remplace les par 1.
fadeldwillidou
```

```
##  [1]  1  1  3  4  5  6  7  8  9 10
```

Imaginons maintenant que nous voulions que les chiffres de 3 à 7 soient remplacés par 4.

```
fadeldwillidou[fadeldwillidou > 2 & fadeldwillidou <=7] <- 4
# Recode les valeurs de fadeldwillidou plus grandes que 2 et inférieure
# ou égale à 7 et remplace par 4.
fadeldwillidou
```

```
##  [1]  1  1  4  4  4  4  4  8  9 10
```

Nous pourrions aussi vouloir changer la valeur de 9 pour celle de 10

```
fadeldwillidou[fadeldwillidou > 8] <- 10 # Recode les chiffres plus grand
# que 8 pour qu'ils prennent la valeur de 10.
fadeldwillidou
```

```
##  [1]  1  1  4  4  4  4  4  8 10 10
```

```
# alternativement, dans ce cas nous aurions aussi pu faire
# fadeldwillidou[fadeldwillidou == 9] <- 10
```

Dans la vraie vie, ces chiffres pourraient correspondre à des codes pour, par exemple, la couleur des yeux. 1 pourrait vouloir dire qu'une personne a les yeux bleus, 4 qu'elle a les yeux bruns, 8 qu'elle a les yeux pairs et 10 qu'elle a les yeux verts. Sachant cela, nous pourrions préférer avoir les labels plutôt que les chiffres. Cela pourrait être pratique plus tard.

```
fadeldwillidou <- factor(fadeldwillidou,
                         labels = c("Bleus", "Bruns",
                                    "Pairs", "Verts"))

fadeldwillidou
```

```
##  [1] Bleus Bleus Bruns Bruns Bruns Bruns Bruns Pairs Verts Verts
## Levels: Bleus Bruns Pairs Verts
```

```
summary(fadeldwillidou)
```

```
## Bleus Bruns Pairs Verts
##     2     5     1     2
```

Les objets *fadeldwill* et *fadeldwillidou* sont en mémoire dans notre environnement de travail (workspace). Nous pouvons voir le contenu de tous les objets qui se trouvent dans cet environnement en faisant la commande *objects()* :

```
objects()
```

```
## [1] "fadeldwill"     "fadeldwillidou"
```

Nous voyons que notre espace de travail ne contient pour l'instant que nos deux objets. Il y a *fadeldwill* et *fadeldwillidou*. R Studio nous permet de facilement voir tout le contenu de l'espace de travail sans passer de commande R en observant simplement le panel approprié dans l'interface.

L'espace de travail peut contenir une très grande quantité d'objets. Ceux-ci peuvent être des trivialités (comme c'est le cas ici), ou encore des choses plus importantes comme des bases de données (des matrices). Contrairement aux autres logiciels statistiques, il est donc très facile d'utiliser en même temps plusieurs bases de données avec R, ce qui est parfois très pratique et impossible (ou souvent fastidieux) avec les autres logiciels.

Ceci étant, puisque les objets *fadeldwill* et *fadeldwillidou* nous seront complètement inutiles, il vaut aussi bien les effacer :

```
rm(fadeldwill, fadeldwillidou) # rm() pour "remove". Nous n'avons pas besoin
# de mettre fadeldwill et fadeldwillidou à l'intérieur d'un c() parce que
# la commande rm() implique forcément une liste d'un ou plusieurs éléments
# et rien d'autre. Alternativement, nous pourrions faire rm(list = ls())
# pour vider complètement l'espace de travail.

objects() # objects() est maintenant vide.
```

```
## character(0)
```

2.4 Une syntaxe qui roule : une bonne méthode de travail

Il est possible d'écrire directement des commandes dans la console de R et dans certains cas cela peut être utile, mais il est beaucoup mieux d'écrire vos commandes dans un script qui pourra ensuite être réutilisé. Par ailleurs, un bon script doit pouvoir être roulé du début à la fin sans erreur. Des erreurs dans un script surviendront, par exemple, si à un moment donné dans la série de commandes on fait référence à un objet qui n'a pas préalablement été créé dans le script lui-même.

Imaginons que j'utilise la console pour créer l'objet x qui contiendra la lettre A. J'écrirais donc la commande suivante directement dans la console plutôt que dans un script :

```
x <- "A"
```

Je viens de créer l'objet x, il existe en ce moment dans ma session, mais je ne l'ai pas fait en écrivant la commande dans un scrip. Ensuite, dans un fichier de synataxe, je modifie l'objet x pour inclure aussi "B". J'écris donc la commande suivante dans un fichier de syntaxe :

```
x <- c(x, "B")
```

Remarquez cependant que notre objet x a d'abord été créé directement dans la console et contenait la lettre A, puis nous l'avons modifié par une commande écrite dans notre script de sorte à ce qu'il inclut aussi la lettre B.

```
x
```

```
## [1] "A" "B"
```

Notre objet x contient donc maintenant les lettres A et B.

Imaginons maitenant que je sauvegarde mon script qui contenait seulement la commande «x <- c(x, "B")» et que je ferme ma session R. Je vais ici simuler la fermeture de la session en vidant le contenu de l'espace de travail.

```
rm(list = ls())
```

Maintenant, imagions que je retourne à mon script deux semaines plus tard et que je veuille le rouler. Mon script contiendrait alors uniquement ceci :

```
x <- c(x, "B")
```

```
## Error in eval(expr, envir, enclos): objet 'x' introuvable
```

Nous avons un message d'erreur parce que notre script tente de modifier l'objet x qui n'est pas préalablement créé dans notre session. Donc, ma syntaxe ne peut pas être «roulée», elle n'est pas «propre» parce qu'elle génère un message d'erreur. Ce message est causé par le fait que mon script fait référence à un objet qui n'est pas créé dans le script lui-même au moment où on fait référence à l'objet. Plutôt que de créer l'objet x via une commande dans la console, il aurait donc mieux valu le faire dans un script complet comme ceci :

```
x <- "A" # Créé l'obet x qui contient A
x <- c(x, "B") # Modifie l'objet x pour lui ajouter B
```

Notez aussi que l'odre des commandes est important. Je ne pourrais pas modifier l'objet x sans d'abord l'avoir créé dans la syntaxe. Simulons encore une fois que nous fermons notre session en vidant notre espace de travail (qui contient actuellement x).

```
rm(list = ls())
```

Imaginons maintenant que ma syntaxe soit écrite en sens inverse :

```
x <- c(x, "B") # Modifie l'objet x pour lui ajouter B
```

```
## Error in eval(expr, envir, enclos): objet 'x' introuvable
```

```
x <- "A" # Créé l'obet x qui contient A
```

Nous avons un message d'erreur parce que la première ligne de la syntaxe tente de modifier l'objet x qui n'a pas encore été créé dans cette syntaxe. L'ordre des commandes est donc important.

Donc, ce qui est nécessaire à ce qu'un script roule bien doit être écrit dans le script lui-même. *N'écrivez des commandes directement dans la console que pour des choses temporaires qui ne sont pas nécessaires à ce que votre syntaxe puisse être roulée complètement.* L'installation (mais pas le chargement) des packages est un exemple de choses qui peut être écrit directement dans la console. Écrire les commandes d'installations des packages directement dans la console évitera aussi de surcharger votre script inutilement. On peut écrire ces commandes directement dans la console parce que l'installation des packages ne se fait normalement qu'une seule fois (sauf pour les mise à jour). Une fois qu'un package a été installé, il n'est pas nécessaire de le réinstaller à chaque session. Cependant, si votre script utilise des fonctions issues de packages, il faut impérativement *charger* ces packages dans votre script *avant* d'utiliser ces commandes pour qu'elles puissent être roulées.

2.5 Charger des données

Plus loin, nous créerons nous-mêmes des données fictives à des fins d'illustration, mais dans la très vaste majorité des cas vous voudrez charger de véritables données. R peut lire des données dans n'importe quel format, qu'elles aient été initialement sauvegardées en format SPSS, Stata (il y a maintenant plusieurs formats Stata depuis la version 13), SAS, csv, et autres. Il suffit d'utiliser la bonne fonction d'importation suivant le format des données que nous voulons charger.

Nous utiliserons ici des données des éltudes électorales canadiennes. Nous téléchargeons d'abord le fichier de données que nous voulons. Les données sont maintenant sur notre ordinateur, nous devons maintenant charger les données dans R. Vous pouvez facilement trouver comment importer des données de différents formats.

R-Studio inclut maintenant un importateur de données via le menu contextuel placé dans la fenêtre environement. On peut donc utiliser ce menu pour charger les données plus

facilement. L'avantage est aussi que ce menu nous indique le code nécessaire pour effectuer le chargement que l'on souhaite faire, et on peut ensuite réutiliser ce code pour charger les données sans avoir à repasser par les menus. Cela est très important quand on veut avoir une syntaxe «propre» qui peut être roulée du début à la fin sans erreur. Donc, si vous écrivez une syntaxe qui fera appel à des données, vous pouvez évidemment charger ces données avec le menu contextuel, mais prenez quelques secondes pour copier-coller le code nécessaire au chargement des données dans votre syntaxe. Elle pourra alors être «roulée» correctement.

```
#rm(list = ls())
library(haven)
ces15 <- read_sav("~/Documents/Data/CES Data/CES2015-phone-release/CES2015_CPS-PES-MBS_c
```

2.6 Voir les variables

Lorsqu'une base de données est chargée, nous voudrons évidemment y référer, et référer aux variables qu'elle contient. La base de données est en elle-même un objet, qui contient des variables. Puisque nous pouvons avoir plusieurs objets dans notre session R, il faut d'abord spécifier à quel objet nous référons avant de spécifier la variable dans cet objet. Cela ce fait avec le signe de $. Par exemple, si nous voulons faire un tableau rapide de la variable *RGENDER* (qui est le sexe des répondants) dans les données que nous venons de charger, on s'y prendrait comme suit :

```
table(ces15$RGENDER)
```

```
##
##    1    5
## 1930 2272
```

Nous voyons donc que nous avons 1930 répondants qui sont codés 1 (les hommes), et 2272 qui sont codés 5 (les femmes). Pour savoir à quoi réfèrent ces codes numériques, il faut consulter la documentation qui accompagne les données. Dans certains cas, les variables seront directement encodées avec des labels clairs, mais ce n'est pas systématiquement le cas.

Si nous voulons remplacer ces codes numériques par quelque chose de plus signifiant, on peut le faire ainsi :

```
ces15$sexe <- factor(ces15$RGENDER,
                          labels = c("Homme", "Femme"))
# Notez ici que plutôt que de modifier la variable originale RGENDER
# j'ai plutôt créé une nouvelle variable "sexe" dans l'objet ces15.
# Cette variable équivaut à la variable RGENDER, mais avec les labels
# à la place des chiffres. SI j'avais écrit "ces15$RGENDER" à gauche
# de la flèche plutôt que "ces15$sexe", alors j'aurais modifié la variable
# originale.

table(ces15$sexe)
```

```
##
## Homme Femme
##  1930  2272
```

Nous pourrions aussi faire un graphique pour visualiser la variable.

```
plot(ces15$sexe) # En utilisant la fonction plot(), R choisi ce qui est
```


```
# le plus approprié pour le type de variable.
```

Nous pourrions aussi enregistrer cette figure quelque part sur notre ordinateur.

```
jpeg('hist.jpg') # créé un fichier jpeg (ce pourrait être un autre type)
# qui sera prêt à recevoir du contenu que nous ferons à la prochaine ligne.

plot(ces15$sexe) # fait l'histogramme

dev.off() # referme la création de figure temprairement ouverte par
```

```
## pdf
##   2
```

```
# la première commande.
```

Nous aurons alors le fichier «hist.jpg» qui sera sauvegardé là où a été établi notre «working directory». Si nous avions voulu enregistrer ce fichier ailleurs, il aurait simplement fallu écrire le chemin complet vers cet endroit. Par exemple, pour sauvegarder la figure sur notre bureau, on aurait fait ceci :

```
jpeg("/Users/alexandreblanchet/Desktop/hist.jpg") # sauvegarde sur le bureau
# au lieu du working directory
plot(ces15$sexe,
```

```
     main="La figure de l'année!", # spécifier un titre à la figure
     ylab="Fréquence")  # Spécifier un label pour l'axe Y

# J'en profite pour mettre un titre et un label à l'axe Y.
# Notez l'ajouts de virgules, qui annoncent qu'il y aura autre
# chose. Notez aussi que la parenthèse ouverte à plot( se referme après
# avoir spécifié les options ajoutées.
dev.off()
```

```
## pdf
##   2
```

Évidemment, le lien est valable pour le bureau de mon ordinateur à moi et il faudrait modifier le lien pour qu'il correspondent au vôtre.

2.7 Tableau croisé

Nous voulons souvent faire des tableaux croisés poru voir comment deux variables sont liées. Imaginons que nous voulions voir dans quelle langue les hommes et les femmes ont répondu au sondage.

```
table(ces15$CPS15_INTLANG)
```

```
##
##    1    5
## 3472  730
```

Nous avons 3472 personnes qui sont codées 1 (anglais) et 730 personnes qui sont codées 5 (français). Nous pouvons modifier ces codes par les labels.

```
ces15$langue_int <- factor(ces15$CPS15_INTLANG,
                           labels = c("Anglais", "Français"))
# Notez qu'encore une fois j'ai créé une nouvelle variable "langue_int"
# plutôt que de modifier la variable originale CPS15_INTLANG.
table(ces15$langue_int)
```

```
##
##  Anglais Français
##     3472      730
```

Puis nous pouvons faire le tableau croisé avec la variable de sexe :

```
table(ces15$langue_int, ces15$sexe)
```

```
##
##            Homme Femme
##   Anglais   1592  1880
```

```
##   Français   338   392
```

Nous obtenons ici les fréquences pour chaque caractéristiques croisées. Nous pourrions cependant vouloir les pourcentages.

```
tableau_sexelangue <- table(ces15$langue_int, ces15$sexe) # Nous plaçons
# d'abord le tableau croisé prédédent dans l'objet tableau_sexelangue,
# puis nous pourrons utiliser cet objet pour demander à R de compiler
# les pourcentage via la commande prop.table()

prop.table(tableau_sexelangue, 1) # Proportions en rangées
```

```
##
##                Homme     Femme
##   Anglais  0.4585253 0.5414747
##   Français 0.4630137 0.5369863
```

```
prop.table(tableau_sexelangue, 2) # Proportions en colones
```

```
##
##                Homme     Femme
##   Anglais  0.8248705 0.8274648
##   Français 0.1751295 0.1725352
```

```
# Si nous voulions avoir les proportions sur 100 (pourcentages)
# plutôt que sur 1, nous pourrions simplement ajouter "*100".
prop.table(tableau_sexelangue, 1)*100 # Pourcentages en rangées
```

```
##
##                Homme    Femme
##   Anglais  45.85253 54.14747
##   Français 46.30137 53.69863
```

```
prop.table(tableau_sexelangue, 2)*100 # Pourcentages en colones
```

```
##
##                Homme    Femme
##   Anglais  82.48705 82.74648
##   Français 17.51295 17.25352
```

Nous pourrions continuer à voir séparément différentes fonctions de R, mais cela deviendrait vite interminable. Vous devriez maintenant avoir les bases pour saisir la logique de la syntaxe de R et pour avoir une idée générale de ce qu'est une syntaxe. C'est ce qui est important. Il est impossible de couvrir le fonctionnement de R sans aussi apprendre à faire des statistiques. Si vous décidez d'apprendre R, vous devrez de toute manière apprendre par essai-erreur et cela viendra avec la pratique. Google sera votre plus grand ami !

Tournons-nous maintenant sur l'objectif principal de toute analyse statistique : décrire.

3 Décrire

Si nous voulions décrire la taille de Serge, nous pourrions par exemple dire qu'il mesure 180cm. Si nous voulions décrire la taille de Serge et Janette, nous pourrions dire que Serge mesure 180cm et que Janette mesure 170cm. Si nous voulions décrire la taille de Serge, Janette et Albert, nous pourrions dire que Serge mesure 180cm, Janette 170cm et Albert 160cm. Nous pourrions continuer d'ajouter des personnes et décrire leur taille, mais à mesure que le nombre de personnes augmentera, il deviendra de plus en plus difficile de décrire la taille des personnes qui nous intéressent en se référant à la taille individuelle de chaque personne. Même l'individu le plus intelligent du monde ne parviendra pas à manipuler simultanément dans sa tête les tailles individuelles de 30 personnes (voir même 10). Pour cette raison, décrire des données implique aussi nécessairement les de *synthétiser*.

Évidemment, on peut décrire un seul élément, mais aussi vouloir décrire la relation entre deux ou plusieurs éléments. En sciences sociales, nous nous intéressons à des choses qui varient. Les choses qui ne varient pas ne nous intéressent généralement pas, justement parce qu'elles ne varient pas. Par exemple, le fait que chaque personne lisant ce texte soit née sur terre (il n'y a pas d'humains né ailleurs que sur terre) est aussi rassurant qu'innintéressant. En statistiques, nous dirons donc que nous nous intéressons aux «variables» qui sont des «choses qui varient» (tadam !), par opposition aux choses qui ne varient pas, qui sont des constantes. Ainsi, lorsque nous cherchons à décrire un phénomène, nous voulons, en termes statistiques, décrire une variable ou des relations entre variables. Concentrons-nous pour l'instant sur la description d'une seule variable.

3.1 Une équation pour décrire une variable

Je voudrais maintenant que vous lisiez l'équation ci-dessous sans lire ce qui vient sous la section «La réponse». Je vais ici décrire ce que veulent dire les symboles et je voudrais que vous réfléchissiez à ce que peut bien vouloir dire cette équation. L'objectif est tout simplement de commencer à vous habituer à réfléchir en termes mathématiques, et surtout de vous faire voir que les formules n'ont rien de magique si l'on prend simplement le temps de s'y arrêter un peu.

3.1.1 Qu'est-ce que cette équation ?

$$? = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Le signe $\sum$ peut se lire comme «la somme de tout ce qui vient à droite». x_i représente la valeur d'un individu i sur la variable x. Donc, $\sum x_i$ implique que nous fassions la somme de toues les x individuels. Imaginez x comme étant le nom d'une colonne dans un fichier excel et chaque x_i représente une cellule particulière de cette colonne. Par exemple, x_1 réfère à la valeur de x à la ligne 1, x_2 la valeur de x à la ligne 2, etc. $i = 1$ sous le signe de $\sum$ signifie

que la somme qui nous intéresse commence à la ligne 1. Le n au-dessus du signe de $\sum$ signifie que nous arrêtons l'addition à n. C'est donc dire qu'ici, nous additionnons toutes les valeurs individuelles de x_1 à x_n, autrement dit les valeurs de x pour toutes les lignes.

Donc, $\sum_{i=1}^{n} x_i$ signifie que nous additionnons toutes les valeurs individuelles de x. Nous voyons ensuite que $\sum_{i=1}^{n} x_i$ est multiplié par $\frac{1}{n}$.

Réfléchissez un instant à ce que veut dire cette équation avant de passer à la section «La réponse». Qu'est-ce que peut bien nous donner le fait d'additionner toutes les lignes d'une colonne ($\sum_{i=1}^{n} x_i$) et de multiplier cette somme par $\frac{1}{n}$?

3.1.2 La réponse

L'équation ci-dessus est tout simplement celle qui nous permet de calculer la moyenne d'une variable. Il s'agit de l'équation «officielle». Si vous n'avez pas trouvé la réponse, la raison est probablement que ce n'est pas de cette manière que la plupart d'entre-nous calculons une moyenne. Habituellement, nous faisons la somme des valeurs d'une variable (donc $\sum_{i=1}^{n} x_i$) et nous divisons cette somme par le nombre d'unités de cette variable (ou n). Autrement dit, ce que vous avez probablement tendance à faire est l'équation suivante :

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Cette équation est équivalente à la première, mais réfléchissons quelques instants pour bien comprendre pourquoi elles donnent forcément le même résultat.

Les deux équations sont formées de la même somme $\sum_{i=1}^{n} x_i$ qui est soit divisée par n, soit multipliée par $\frac{1}{n}$. Si par exemple nous nous intéressions à la taille moyenne d'un groupe de 50 individus, $\sum_{i=1}^{50} x_i$ veut donc dire que nous additionnons la taille de chaque individu (x_i) pour obtenir la taille totale du groupe en entier. Lorsque vous calculez une moyenne, vous avez probablement tendance a diviser la somme totale du groupe par n, ici le nombre d'individus dans notre groupe, soit 50. En divisant la somme totale du groupe par le nombre d'individus qui le compose, on se trouve donc à «répartir» la taille totale du groupe parmi les 50 individus. C'est tout à fait correct et logique. Cependant, remarquez comment, plutôt que de «répartir» la somme totale sur n, la première équation donne plutôt un «poids» à chaque individu en divisant 1 par n.

Imaginons que la taille totale de nos 50 individus donne 9000cm ($\sum_{i=1}^{n} x_i$=9000). La formule que vous utilisez habituellement serait donc $\bar{x} = \frac{9000}{50} = 180$. Si nous procédons plutôt en suivant la première formule, nous remarquons d'abord que $\frac{1}{50} = 0.02$. Dans un groupe de 50, un individu représente donc $\frac{1}{50}$, ou 0.02 du total des individus. La première formule revient donc à faire $\bar{x} = 0.02 \times 9000 = 180$.

Avec la première formule, nous voulons donc obtenir une certaine proportion (ici 0.02, ou $\frac{1}{50}$) du total des x_i, alors qu'avec la formule «habituelle», on «réparti» ce total sur le nombre d'éléments dont il est composé. Plutôt que de «répartir» la taille totale sur le nombre d'individus, la première formule «pondère» le poids de chaque individu en remenant le «poids»

d'un seul individu sur le total du groupe sur une unité de 1 ($\frac{1}{n}$). La logique est légèrement différente, mais elle revient au même.

3.1.3 Une équation générale décrivant la variable ?

L'objectif de la section précédente était tout simplement de vous faire voir la «mécanique» d'une équation avec quelque chose que la plupart d'entre-vous utilisez régulièrement. L'exemple de la moyenne nous permet aussi de voir comment deux logiques différentes, mais équivalentes, peuvent aussi être écrites de manière différentes mathématiquement. En utilisant le concept de moyenne que la plupart d'entre-vous connaissez bien, je tente de diminuer votre angoisse (Tobias 1993) matématique face aux équations !

Ceci étant, l'un des avantages des équations est qu'elles nous permettent aussi de proposer une description claire et succincte d'une variable. Imaginons que nous voulions décrire la variable x en une seule équation et de manière la *plus complète possible*. En conservant notre exemple du groupe de 50 personnes ayant une taille moyenne de 180cm, nous pourrions dire que la moyenne de $x = 180$cm et cette information est très certainement utile. Cependant, la moyenne du groupe demeure tout de même une information générale qui n'est pas forcément vraie pour chaque personne, elle nous informe sur le groupe entier, pas sur chaque individu spécifique. Nous pourrions cependant utiliser la moyenne afin de décrire la variable à l'aide d'une équation qui sera «vraie» pour chaque individu. C'est ce que fait l'équation suivante :

$$x_i = \frac{1}{n}\sum_{i=1}^{n} x_i + e_i$$

Ici, x_i équivaut encore à la valeur de x pour l'individu i. Remarquons que la formule est pratiquement identique à la formule de la moyenne, mais nous y avons ajouté $+e_i$, où e_i signifie *l'écart* de chaque individu i par rapport à la moyenne ($\frac{1}{n}\sum_{i=1}^{n} x_i = 180$).

Donc, si par exemple l'individu 2 mesure 190cm ($x_2 = 190$), la valeur de e_2 sera de 10. Ainsi, $x_2 = 180 + 10 = 190$ et nous voyons que l'équation proposée décrit adéquatement l'individu 2. Si si l'individu 3 mesure 170cm ($x_3 = 170$), la valeur de e_3 sera de -10. e_i équivaut donc à la distance d'un individu par rapport à la moyenne du groupe.

Ces écarts individuels à la moyenne peuvent être mis au carré (pour éliminer les signes négatifs) et additionnés ensemble. Cela nous donne ce que nous appelons la *somme du carré des écarts*, c'est-à-dire le grand total de tous les e_i mis à la 2 (e_i^2). L'une des propriété de la moyenne est qu'elle est le nombre qui *minimise* cette somme. Aucun autre nombre que la moyenne d'une distribution ne peut donner une *somme du carré des écarts* plus petite. Une moyenne n'est pas forcément le meilleur nombre à utiliser pour synthétiser une variable, mais la moyenne est systématiquement le chiffre qui est le plus proche de toutes les valeurs individuelles d'une variable donnée. Nous verrons plus tard que la régression présente une caractéristique similaire et il est donc utile que vous compreniez pourquoi cette caractéristique est intéressante à partir du concept de moyenne que vous maîtrisez déjà.

Le concept d'écart individuel à la moyenne est aussi très utile parce qu'il nous permet d'écrire une règle générale qui décrira précisément toutes nos données et ce en tenant compte

de la variation individuelle autour de notre moyenne. Pour une équation générale, nous pourrions simplifier encore plus et écrire :

$$x_i = \bar{x} + e_i$$

Où $\bar{x}$ (prononcé «x-barre») a simplement remplacé l'équation plus détaillée de la moyenne. La taille précise de l'individu 2 décrit plus haut peut être écrite :

$$x_2 = \bar{x} + e_2$$

$$x_2 = 180 + 10 = 190$$

Vous voyez donc que l'on peut décrire de manière succincte une variable en établissant une caractéristique de cette variable qui s'applique à tout le groupe (ici la moyenne du groupe) et en y ajoutant la *déviation* (ou l'écart) de chaque individu par rapport à cette caractéristique commune. Ce qui varie d'un individu à l'autre (les valeurs de x et e) sont indicés par i (x_i et e_i), alors que ce qui est valable pour tout le groupe (la moyenne) n'a pas d'indice ($\bar{x}$). Les indices nous permettent de spécifier plus clairement ce à quoi nous référons. Ceci étant, il arrive fréquemment qu'ils soient omis lorsqu'il n'y a aucun doute sur l'interprétation d'une équation.

Il est fort probable que vous ne décriviez jamais la moyenne d'un groupe d'individus avec l'équation que nous venons de voir, mais l'objectif était simplement de vous faire voir comment on peut formuler une description générale valable pour tous les individus d'un groupe à partir d'une caractéristique commune au groupe. L'idée n'est donc pas ici que vous appliquiez cette formule, mais que vous compreniez le concept derrière l'idée de «déviation» individuelle par rapport à la caractéristique commune. Cette compréhension intuitive vous sera très utile lorsque nous arriverons à la régression.

Il y a bien sûr de nombreuses autres manières de décrire une variable.

— Les mesures de tendances centrales : mode, médiane, moyenne
— Les mesures de variation : étendue, variance, écart-type
— Les mesure d'asymétrie : coefficient d'asymétrie

3.2 Les types de variables et éviter de faire n'importe quoi

La manière la plus adéquate de décrire une variable est aussi liée au *type de variable*. Les variables peuvent être :

— Nominales
  — Une variable qui décrit des états différents qui ne sont pas ordonnancés hiérarchiquement.
  — Les catégories doivent être exhaustives et mutuellement exclusives.
  — Exemples : la couleur des yeux ou des cheveux. Le choix de vote (PLQ, PQ, CAQ, QS).
— Ordinales

    - Une variable qui varie en fonction de différents états qui *peuvent* être ordonnancés hiérarchiquement, mais dont la distance entre ces états ne peut être réellement établie.
    - Exemples : très satisfait, satisfait, insatisfait, très insatisfait. Un peu, beaucoup, pationnément, à la folie.
- Continues (parfois appelées «d'intervalles/ratio»)
    - Une variable qui peut être mesurée par une unité standard.
    - Exemples : le poids, la taille, la distance, le temps.
- Dichotomiques (parfois appelées «dummy»)
    - Comme les variables nominales, mais avec seulement deux catégories.
    - Exemples : homme/femme, riche/pauvre, avoir un diplôme uiversitaire/ne pas en avoir.

Contrairement aux variables nominales et ordinales, les variables dichotomiques ont cetains avantages qui viennent du fait qu'elles n'ont que deux catégories. Elles peuvent notamment être taitées numériquement (0 et 1) même si les chiffres 0 et 1 réfèrent à des catégories distinctes qui ne sont pas ordonancées hiérarchiquement. Par exemple, il ne fait aucun sens de faire la moyenne d'une variable nominale ou ordinale : calculer la moyenne de la couleur des yeux des gens dans une classe n'a pas de sens. Par contre, même s'il ne fait pas de sens de calculer la «moyenne du sexe» des gens dans une classe, cette moyenne nous donnera quand même un chiffre interprétable : la proportion d'hommes et de femmes. Par exemple, si nous avons 4 hommes (codés 0) et 6 femmes (codées 1), la moyenne nous donnera $\frac{6}{10} = 0.6$, donc la proportion de la catégorie qui est codée 1, ici les femmes. Inversement, nous saurons aussi forcément qu'il y a 40% d'hommes.

Ceci étant, lorsque l'on fait des analyses statistiques, il est important de faire attention au type de variables que nous utilisons si l'on veut les décrire adéquatement. C'est également le cas lorsque nous voulons décrire une relation entre deux variables. Peu importe le type de variable, les logiciels et les procédures statistiques traitent des colones de nombres. *Un logiciel n'a aucune idée de ce que ce que ces nombres veulent dire, ni s'il fait du sens de faire la procédure que vous lui demandez d'effectuer.*

Si vous avez un groupe d'individus et que vous vous intéressez à la couleur de leurs yeux qui peuvent être bleus (codé 1), bruns (codé 2), pairs (codé 3), ou verts (codé 4) ; le logiciel vous donnera un résultat si vous lui demandez la moyenne de la couleurs de leurs yeux. Le nombre n'aura aucun sens, mais le logiciel ne le saura pas. Il aura simplement calculé la moyenne des nombre codés de 1 à 4 dans la colone «couleur des yeux» sans savoir qu'ils réfèrent en fait à des couleurs. C'est à vous de le savoir. Nous verrons en chemin que les logiciels peuvent traiter des variables de plusieurs format – par exemple des variables numériques ou des variables de facteur (bleu, brun, pair, vert au lieu des nombres), et ces différents types de variables peuvent être traitées différemment suivant les logiciels. Au final cependant, tout est en quelque part transformé en un chiffre. C'est à vous de vous assurer que le tout fait du sens et *aucune procédure statistique ne peut se substituer à votre jugement.*

3.3 Décrire une relation entre des variables

Si bien sûr nous pouvons souvent nous intéresser à la description d'une seule variable (par exemple : pour quel parti les gens ont le plus l'intention de voter lors des prochaines élections ?), on s'intéresse aussi souvent à la *relation* entre des variables. La raison en est tout simplement que nous cherchons des *explications* à un phénomène (une variable) qui nous intéresse et nous désirons donc voir si le phénomène diffère (si la variable varie) en fonction d'un autre phénomène. Pour cette raison, nous nous intéressons souvent aux corrélations entre des variables.

Une mesure souvent utilisée pour évaluer la relation entre deux variables est le coefficient de corrélation. Ce coefficient varie en -1 et 1. Un coefficient de 1 indique une relation positive parfaite, un coefficient de -1 une relation négative parfaite et un coefficient de 0 indique une absence «parfaite» de relation entre les variables. Par exemple, imaginons que nous avons les variables suivantes pouitA, pouitB, pouitC et pouitD :

```
pouitA <- c(1, 2, 3, 4, 5)
pouitB <- c(6, 7, 8, 9, 10) # Similaire à pouitA, mais de 6 à 10
pouitC <- c(5, 4, 3, 2, 1) # Inverse de pouitA
pouitD <- c(2, 1, 2, 5, 4) # Quelques variations par rapport à pouitA
```

PouiA et PouitB auront une corrélation de 1, alors que PouitC aura une corrélation de -1 avec PouitA et PouitB. À vue d'oeil, PouitD devrait avoir une corrélation d'autour de 0.75 avec pouitA et pouitB, et d'environ -0.75 avec pouitC.

```
# Corrélation entre PouitA et PouitB
cor(pouitA, pouitB) # cor() pour "correlation"
```

```
## [1] 1
```

```
# Corrélation entre PouitA et PouitC
cor(pouitA, pouitC) # Quand puitA augmente, pouitC diminue
```

```
## [1] -1
```

```
# Corrélation entre PouitB et PouitC
cor(pouitB, pouitC) # Quand puitB augmente, pouitC diminue
```

```
## [1] -1
```

```
# Corrélation entre PouitA et PouitD
cor(pouitA, pouitD) # Quand puitA augmente, pouitD augmente
```

```
## [1] 0.7698004
```

```
# Corrélation entre PouitB et PouitD
cor(pouitB, pouitD) # Quand puitB augmente, pouitD augmente
```

```
## [1] 0.7698004
```

```r
# Corrélation entre PouitC et PouitD
cor(pouitC, pouitD) # Quand puitC augmente, pouitD diminue
```

```
## [1] -0.7698004
```

```r
# Débarassons-nous de ces pouits...
rm(pouitA, pouitB, pouitC, pouitD) # rm() pour "remove" comme vu plus haut.
```

En règle générale, un coefficient de 0.5 (ou de -0.5) ou plus en sciences sociales est généralement considéré comme relativement fort. Plus le chiffre en termes absolu (c'est-à-dire sans tenir compte du signe négatif ou positif) est élevé, plus la relation est forte. Cependant, nous ne nous étendrons pas ici sur ce coefficient ni sur son calcul puisqu'il est relativement simple et que je souhaite que nous nous concentrions sur la régression, qui est abondement utilisée en méthodes quantitatives. Aussi, faites attention de *ne pas confondre les coefficients de corrélation avec les coefficients de régression*. Un coefficient de régression de 0.25 (ou même de 0.02) peut être très fort dépendemment de l'échelle de mesure des variables concernées.

Ceci étant, une chose doit être précisée dès le début : *la corrélation n'équivaut pas à la causalité !* La question de la causalité est très complexe et controversée. Gardons-nous donc d'en discuter et contentons-nous de dire que nous cherchons à *comprendre* un phénomène qui nous intéresse[1]. Par *comprendre*, nous voulons donc dire que nous voulons *expliquer* ce qui fait en sorte que le phénomène survient ou non, ou mieux décrire comment au juste il survient. Nous avons une certaine thèse sur les facteurs qui font en sorte que le phénomène advient ou n'advient pas, ou qu'il advient d'une manière plutôt que d'une autre. Trois règles logiques doivent minimalement être respectées :

1. La chose qui explique le phénomène doit être antérieure (temporellement, logiquement, ou théoriquement) à celui-ci.
   — Temporellement : ce qui arrive aujourd'hui ne peut expliquer ce qui est arrivé hier. C'est bêtement logique.
   — Théoriquement : Même si une explication potentielle est temporellement antérieure à un phénomène, il faut avoir un argument raisonnable pour affirmer qu'ils sont liés. Cet argument (ou idéalement *ces* arguments) n'a pas grand-choses à voir avec les statistiques. Il s'agit ici de réflexion théorique et empirique.
2. Pour qu'une chose en explique une autre, elle doit lui être liée d'une manière ou d'une autre. Ceci implique donc forcément la présence d'une association entre le facteur explicatif et le phénomène que l'on veut expliquer.
   — En méthode quantitatives, cela implique que les deux éléments doivent être associés statistiquement. C'est-à-dire que les valeurs des deux éléments doivent être corrélées.
   — Une corrélation peut être *positive* : quand les valeurs d'une variable sont élevées, celles d'une autre variable ont tendance à être plus élevées.
   — Une corrélation peut être *négative* : quand les valeurs d'une variable sont élevées, celles d'une autre variable ont tendance à être plus faibles, ou inversement.

1. Pour un livre intéressant sur cette question et nuancé sur le positivisme général, voir Favre (2005).

3. Le lien entre le facteur explicatif et le phénomène d'intérêt ne doit pas être fallacieux, c'est-à-dire qu'il ne doit pas être généré par un autre facteur qui explique les deux éléments initiaux.
   — Par exemple, si l'on remarque que les dégâts lors d'un incendie sont plus importants lorsqu'il y a plus de pompiers pour le combattre, il serait évidemment fallacieux d'en conclure que les pompiers causent les dégâts. C'est *l'intensité* de l'incendie qui rend nécessaire la présence d'un plus grand nombre de pompiers pour le combattre *et* c'est aussi cette intensité qui explique que les dégâts sont plus importants. Le fait qu'il y ait plus de dégâts quand il y a plus de pompiers (une simple association statistique) n'implique absoluement pas que les pompiers produisent les dégâts.

3.4 La régression linéaire simple avec une variable dichotomique

Mon objectif est ici que vous développiez une compréhension intuitive de ce qu'est la régression linéaire et de ce que les résultats veulent dire au juste. Nous n'allons pas nous amuser à calculer à la main des régressions, mais nous allons en faire avec des données que nous créerons nous-mêmes. Pour cette section, oubliez les étoiles et concentrez-vous uniquement les coefficients. Les coefficients permettent de *décrire* les données, les étoiles ont trait à l'inférence. Or avant d'inférer quoique ce soit, encore faut-il le décrire.

3.4.1 Création des données

Nous allons ici créer des données que nous inventerons nous-mêmes pour fins d'illustration. Nous nous intéresserons aux notes qu'un groupe de 100 élèves ont reçu à leur dernier examen du cours d'introduction à la pétanque full contact (PFC 1000). Nous nous pencherons sur l'impact du sexe des élèves et du nombre d'heures d'étude sur la note qu'ils ont reçu.

La syntaxe ci-bas crée les données et quelques explications sont fournies en annotation. Vous n'avez cependant pas à comprendre cette syntaxe pour comprendre ce qui suivra.

```
# Pour réplication
set.seed(987654321) # R génère des chiffres aléatoirement et pour
# qu'il le fasse de manière constante d'une fois à l'autre, on
# peut lui dire de le faire d'une manière exacte. Si vous ne le faites pas,
# vous obtiendrez probablement des résultats différents de ceux du document
# tout simplement parce que R génèrera aléatoirement des données un peu
# différentes.

fille <- rbinom(100, 1, 1/2)# créer 100 individus qui ont une chance de
# 1/2  d'être codé 1 dans l'objet "fille", les autres seront codés 0.

heures.etude <- rnorm(100, mean=4.5 + 1.5*fille, sd=1.5 ) # Crée la variable
# heures études avec une moyenne de 4.5, les filles qui étudient en moyenne
```

```
# 1.5 heures de plus que la garçons, le tout avec une erreur standard de
# 1.5 pour mettre une peu de variation dans tout cela.

notes <- rnorm(100, mean=35 + 4*fille + 5*heures.etude, sd=6) # Crée la
# variable notes avec une moyenne de 35, avec les filles qui ont en moyenne
# 4 de plus que la garçons, les heures d'études apportant en moyenne
# 5 points de plus par heure, le tout avec une erreur standard de 6
# pour mettre une peu de variation dans tout cela.

sexe <- factor(fille, labels = c("garçon", "fille")) # crée la variable sexe
# à partir de la variable numérique "fille". Sexe sera traité comme un
# facteur ce qui ne change rien mathématiquement mais qui simplifie
# la création des graphiques que nous ferons plus tard (Voir la note de
# bas de page 3).

data <- data.frame(notes, sexe, heures.etude, fille) # place les
# variables dans une même matrice (i.e. un seul object avec 4 colones)

# Utilisons la fonction Head() histoire de mieux voir les données.
head(data) # Nous montre les 6 premières rangées des données
```

```
##      notes   sexe heures.etude fille
## 1 56.36004 garçon     3.429696     0
## 2 70.16088  fille     5.102714     1
## 3 61.71704  fille     4.780638     1
## 4 70.84951 garçon     5.656184     0
## 5 72.49611  fille     3.883307     1
## 6 84.08077 garçon     8.094016     0
```

Vous voyez qu'une fois réunie en une seule matrice (comme une feuille d'un fichier excel), nos variables ont chacune une colone et les données individuelles sont en rangées. Le tableau ci-haut ne montre que 6 de ces rangées, mais il y en a une par élève, donc 100 au total. Ayez donc en tête que nos données sont en fait très semblables à un fichier excel dans lequel nos variables sont en colones et les scores de chaque élève sur ces variables sont en rangées. *Encore une fois, les logiciels et les procédures statistiques traitent des colones de chiffres. C'est à vous de savoir ce que ces colones signifient et si ce que vous demandez au logiciel a du sens.*

Nous pouvons avoir plus de détails sur chacune des variables si nous le souhaitons.

```
summary(data)
```

```
##      notes           sexe     heures.etude          fille
##  Min.   :37.28   garçon:44   Min.   : 0.07104   Min.   :0.00
##  1st Qu.:56.74   fille :56   1st Qu.: 4.10012   1st Qu.:0.00
##  Median :62.87               Median : 5.18805   Median :1.00
##  Mean   :63.41               Mean   : 5.25523   Mean   :0.56
```

```
##  3rd Qu.:69.82                3rd Qu.: 6.61722   3rd Qu.:1.00
##  Max.   :94.28                Max.   :11.28492   Max.   :1.00
```

L'objet «data» contient quatre variables (colones) : notes, sexe, heures.etude et fille. La variable sexe est une variable de facteur («factor» ou «character» en anglais dans R) en ce sens qu'elle ne contient pas de chiffres, mais des mots. Les cellules de cette colone contiennent soit le mot «garçon», soit le mot «fille». La variable fille est identique, mais elle est numérique. Ses cellules contiennent un 0 si l'élève est un garçon et le chiffre 1 si l'élève est une fille. R sait que la variable «sexe» est un facteur (elle est «encodée» de cette manière), mais il ne sait pas que la variable fille peut aussi être considérée comme tel puisque cette colone ne contient que les chiffres 0 ou 1. En conséquence, R nous donne des infomrations différentes à propos de ces variables. Ceci étant, qu'elle soit numérique ou en facteur, la variable sexe (ou fille) est dichotomique et il n'y a aucune différence à traiter cette variable comme facteur ou comme des valeur numériques. Ce ne serait cependant pas le cas si nous avions à faire à une variable ordinale ou nominale.

Nous voyons donc qu'il y a 56 filles et 44 garçons. Ce n'est pas 50% pile puisque j'ai demandé à R qu'il y ait un élément aléatoire dans la génération des données afin qu'elles ressemblent davantage à des «vraies» données. Cependant, la loi des grands nombre fait en sorte que si nous avions créé un échantillon plus gros, la proportion se rapprocherait encore plus de 50% de filles et de garçons, puisque j'ai demandé à R d'avoir *environ* 50% d'élèves de chaque sexe.

Intéressons nous d'abord à la moyenne des notes de tous les élèves. Nous l'avons déjà vu dans le tableau ci-haut, mais demandons tout de même à R de nous donner spécifiquement cette moyenne.

```
# Puisque la variable «notes» est dans l'objet data, il faut d'abord dire à
# R l'objet, puis la variable avec un signe de $ entre les deux.
mean(data$notes) # Nous demandons la moyenne de notes dans l'objet data
```

```
## [1] 63.41256
```

Nous voyons donc que la moyenne générale est de 63.4125571. Cette moyenne est-elle différente selon le sexe ? La manière la plus simple de répondre à cette question est tout simplement de faire un tableau croisé des notes en fonction du sexe. Faisons-le.

```
# Nous chargerons ici le premier «package». R est ouvert par défaut avec
# des fonctions qui sont toujours actives, mais afin de minimiser
# l'utilisation de la mémoire ram de l'ordinateur, on doit activer
# les «packages» supplémentaires dont on a besoin dans une session. Cela
# permet d'éviter que des fonctions que nous n'utilisons pas dans une
# session soient actives inutilement. Des «packages» supplémentaires
# peuvent aussi être téléchargés et être ensuite activés au besoin.
# Les packages supplémentaires ne doivent être téléchargés qu'une seule
# fois et ils sont ensuite dans la «library» et sont traités comme
# n'importe quel autre package. Les packages n'ont besoin d'être activés
# qu'une seule fois dans une session.
```

```r
# Vous pouvez aussi consulter le menu «Packages» dans le panel
# approprié de l'interface R-Studio.
library(dplyr) # Package très utile pour la manipulation des données.

# Tableau croisé des notes moyennes selon le sexe
data %>% # Sélectionne l'objet data puis ...
  group_by(sexe) %>% # Regroupe les données par la variable sexe puis...
  summarise_each (funs(mean) , notes_moy = notes) # Donne la moyenne
```

```
## # A tibble: 2 x 2
##   sexe   notes_moy
##   <fct>      <dbl>
## 1 garçon      57.5
## 2 fille       68.0
```

Ce tableau croisé nous montre que les garçons ont des notes moyennes de 57.5, alors que les filles ont une moyenne de 68.0. Donc, les filles ont en moyenne de meilleures notes que les garçons dans nos données. Ici, la variable *sexe* est dichotomique. Si elle est traitée comme étant numérique, elle ne peut avoir que la valeur de 0 (lorsqu'il s'agit d'un garçon) ou de 1 (lorsqu'il s'agit d'une fille). Il n'y a donc personne entre 0 et 1 sur cette variable.

La plupart d'entre vous ont probablement une compréhension très intuitive de la signification du tableau croisé ci-haut. Il nous donne la moyenne des filles et des garçons. Je veux cependant vous démontrer que la régression linéaire peut nous offrir la même information. L'équation de base d'une régression linéaire simple peut s'écrire comme suit :

$$y = a + \beta x$$

Où a est l'intercept (aussi souvent appelé la «constante») et βx est le coefficient de régression associé à la variable x. L'intercept correspond à la valeur de notre variable dépendante Y quand toutes les autres variables de l'équation (ici seulement x) ont une valeur de 0. Une formulation plus précise pourrait aussi être écrite comme ceci :

$$y_i = a + \beta x_i + e_i$$

Ici, nous disons simplement que la valeur sur y de l'individu i est une fonction de l'intercept a, de l'effet β de la variable x qui varie selon l'individu i, et d'un terme d'erreur e spécifique à chaque individu i. Ce terme d'erreur est similaire à la distance d'un individu par rapport à la moyenne dont nous avons parlé plus tôt. [2]

Appliquons maintenant cette équation aux données.

2. Ceci étant, lors d'une simple équation de régression linéaire habituelle, les indices sont sous-entendus et l'on ne prend donc généralement pas la peine de les écrire. Bien indicer les équations est cependant essentiel quand les variables peuvent varier selon différents éléments. Notamment quand nous avons des données longitudinales où les variables peuvent varier à la fois *entre* les individus et «à l'intérieur» d'un même individu dans le temps. Avec des variables longitudinales, vous verrez souvent des variables qui varient *entre* les individus i, et à l'intérieur des individus dans le temps j. Cela peut par exemple donner des variables indicées par ij comme x_{ij}.

```r
m1 <- lm(notes ~ sexe) # Crée l'objet m1 qui contient les résultats d'un
# modèle linéaire (fonction lm() pour "linear model") dans lequel
# la variable note est une fonction (~) du sexe.
summary(m1) # Pour voir les résultats dans l'objet m1
```

```
##
## Call:
## lm(formula = notes ~ sexe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.2474  -7.3035  -0.6993   6.0207  26.5525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   57.528      1.441  39.933  < 2e-16 ***
## sexefille     10.508      1.925   5.458 3.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.556 on 98 degrees of freedom
## Multiple R-squared:  0.2331, Adjusted R-squared:  0.2253
## F-statistic: 29.79 on 1 and 98 DF,  p-value: 3.627e-07
```

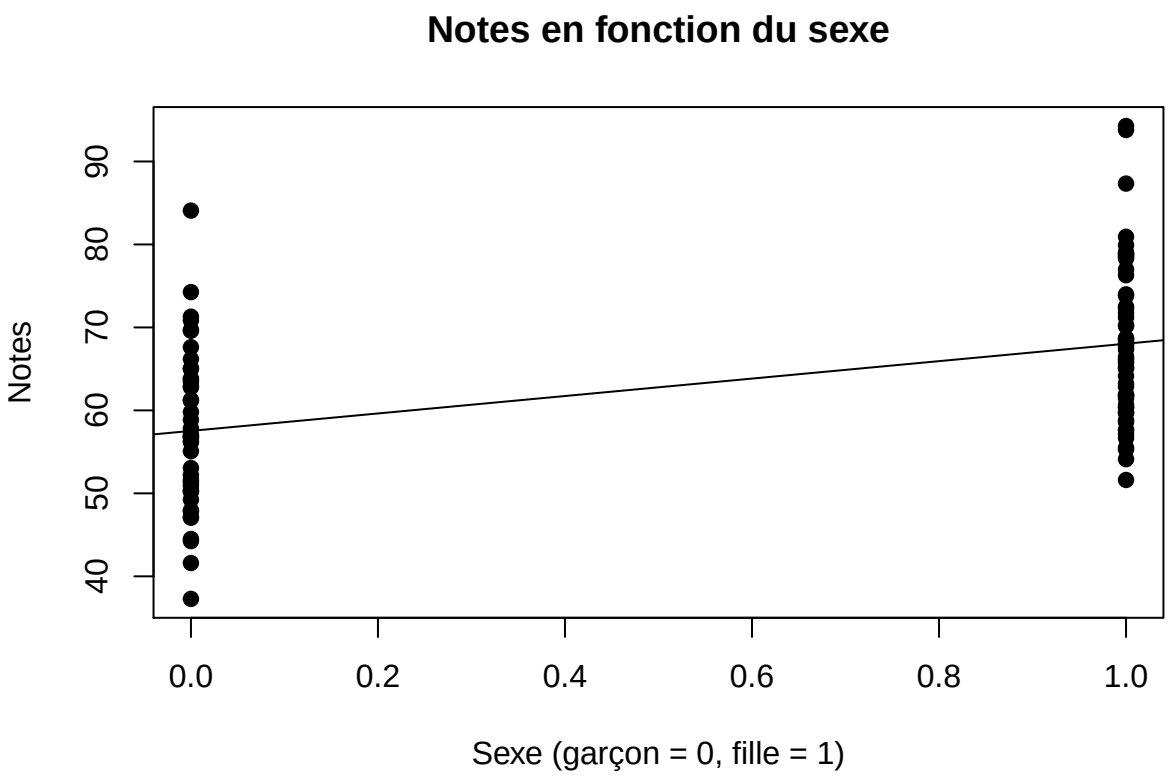
Nous voyons donc la valeur de l'intercept a est de 57.5282439 et que la valeur du coefficient β associé à la variable dichotomique *sexe* est de 10.5077022. La variable sexe est dichotomique : dans nos données un individu est soit un garçon ou une fille. Il y a donc seulement deux options. En termes mathématiques, cela revient à accorder la valeur de 1 à l'une des possibilités et de 0 à l'autre. Dans notre cas, nous avons choisi d'accorder la valeur de 1 lorsque l'élève est une fille et de 0 lorsque c'est un garçon. Nous aurions aussi pu faire l'inverse. Les résultats de la régression auraient été les mêmes, mais le coefficient de 10.5077022 aurait simplement été négatif plutôt que positif («les garçons ont de moins bonnes notes que les filles», plutôt que «les filles ont de meilleures notes que les garçons»).

Nous pourrions aussi visualiser les données.

```r
plot(sexe, notes, main="Notes en fonction du sexe",
    xlab="Sexe", ylab="Notes", data=data)
```

Ici, R traite la variable *sexe* comme un facteur, c'est-à-dire comme une variable qui distingue différents états qui n'ont pas d'ordonnancement numérique. Pour cette raison, la fonction *plot()* nous donne automatiquement un diagramme en boîte à moustache (c'est le nom !) puisque cela fait plus de sens pour des variables dichotomiques qui marquent des états distincts. Il demeure que cette variable est mathématiquement traitée comme si fille=1 et garçon = 0. Voyons ce qu'une telle figure donnerait si R «pense» que la variable est numérique.

```
# La variable fille est utilisée ici plutôt que sexe parce que cette
# variable est enregistrée dans R comme une variable de format numérique.
# R la traite donc comme telle. Cependant, gardez bien à l'esprit que
# cela revient mathématiquement au même.
plot(fille, notes, main="Notes en fonction du sexe",
    xlab="Sexe (garçon = 0, fille = 1)", ylab="Notes", pch=19, data=data,
    abline(m1))
```



Chaque point est un individu. S'il s'agit d'un garçon, le point est à gauche (à 0) sur l'axe des x, alors qu'il est à droite (à 1) s'il s'agit d'une fille. Il n'y a personne au milieu puisqu'un élève est soit un garçon, soit une fille (raison pour laquelle cette variable est normalement traitée comme un facteur). La hauteur du point nous indique la note obtenue par l'élève. La ligne est l'équation de régression. La distance entre chaque point individuel et la ligne est l'erreur e_i, c'est-à-dire ce qui n'est pas décrit parfaitement par notre équation (la ligne). Nous appelons aussi souvent cette erreur les «résidus».

L'intercept a est la valeur moyenne de notre variable dépendante lorsque notre variable dépendante $sexe = 0$. Autrement dit, a est ici la note moyenne des garçons, soit 57.5282439. La valeur du coefficient de régression β (prononcé beta) nous indique le mouvement moyen de notre variable dépendante y lorsque notre variable dépendante x augmente de 1 unité. Dans le cas qui nous occupe, «augmenter» de 1 unité signifie être une fille plutôt qu'un garçon. Vous voyez donc que même si $1 > 0$ alors que les filles ne peuvent pas être considérées «plus» que les garçons, cela ne fait pas réellement de différence mathématiquement puisqu'une unité peut ici être interprétée comme distinguant ces états qui n'ont pas d'ordonnancement hiérarchique. *Cela n'est cependant vrai que pour les variables dichotomiques*, qui sont un cas particulier des *variables nominales*[3].

3. Les variables dichotomiques sont un cas particulier des *variables nominales*. Comme les variables dichotomiques, les variables nominales distinguent des états différents qui ne peuvent être ordonnancés hiérarchiquement, mais elles comportent plus de deux possibilités. Un exemple de variable nominale est le choix de vote : PLQ, PQ, CAQ, QS. Ce sont là quatre possibilités distinctes qui ne peuvent être ordonnancées hiérarchiquement. Lorsque l'on veut utiliser une telle variable comme variable indépendante dans une régression, on ne peut pas laisser une telle variable dans un format numérique comme PLQ=1, PQ=2, CAQ=3 et QS=4 parce qu'il n'y a alors aucun moyen de distinguer mathématiquement qu'une unité de cette variable équivaut seulement à un état différent. Il faut alors «dichotomiser» toutes les options (par exemple : PLQ =

Ainsi, dans le cas qui nous occupe, notre modèle prévoit que la note moyenne des filles sera de $a + \beta x$, soit de 57.5282439 + 10.5077022= 68.0359461.

Rappelons-nous quelles étaient les moyennes des garçons et des filles dans notre tableau croisé :

```
data %>%
  group_by(sexe) %>%
  summarise_each (funs(mean) , notes_moy = notes)
```

```
## # A tibble: 2 x 2
##    sexe   notes_moy
##    <fct>      <dbl>
## 1 garçon       57.5
## 2 fille        68.0
```

Nous constatons que les chiffres sont très exactement les mêmes. La valeur de l'intercept a est égal à la moyenne des garçons (c'est-à-dire quand le sexe = 0), alors que $a + \beta x$ est égal à la moyenne des filles. Inversement, nous pourrions aussi dire que dans ce cas, la valeur du β est tout simplement la différence de moyenne entre les garçons et les filles, soit 68.03595 - 57.52824 = 10.50771. Vous voyez donc que notre modèle de régression n'a rien de magique, il *décrit* tout simplement nos données.

Vous comprendez aussi que lorsque nous proposons l'équation $y_i = a + \beta x_i + e_i$, ne sommes absolument pas en train de dire que les notes des élèves de notre cours de pétanque obéissent à une sorte de loi immuable de l'univers. Nous disons simplement que les notes de nos élèves varient autour d'une certaine valeur a et qu'elles ont tendance à être différentes (une différence moyenne de valeur β) en fonction du sexe des élèves. Nous ne savons pas si le sexe des élèves est la *cause* de la variation des notes, mais nous constatons tout simplement que, dans nos données, les filles ont tendance à avoir de meilleures notes que les garçons. Plutôt que de décrire les données en donnant la moyenne des garçons et des filles séparément (décrire les données avec deux chiffres, les moyennes par groupe), nous avons simplement proposé une équation qui est équivalente.

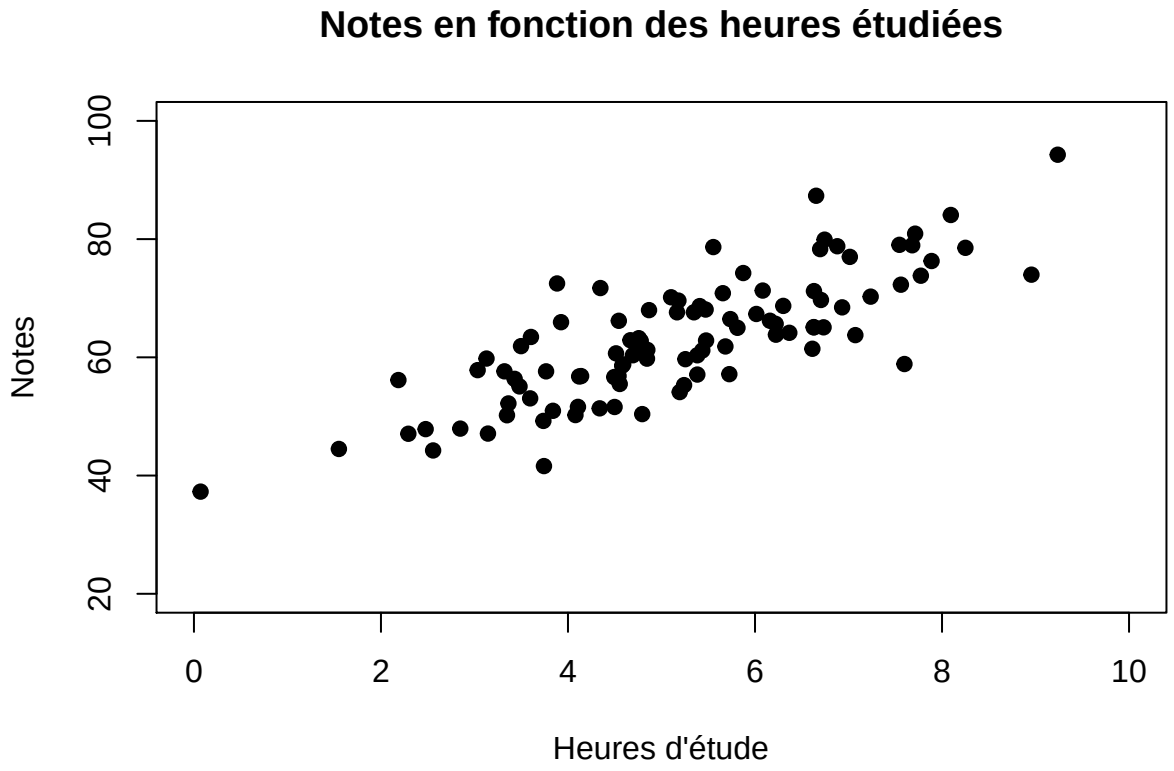
1, autres = 0 ; PQ = 0, autres = 0, etc.) et laisser une catégorie de référence à partir de laquelle les autres options seront interprétées. Nous y reviendrons plus tard. Ceci étant, R peut enregistrer des variables en différents formats, notamment en format numérique et en «facteur» (fator en anglais). Une variable en facteur est une variable qui distingue différents états qui ne sont pas hiérarchisés numériquement, comme le sexe ou le choix de vote. Lorsqu'une variable est enregistrée comme facteur, R dichotomisera automatiquement cette variable lorsqu'elle sera utilisée comme variable indépendante dans un modèle. Vous remarquerez que dans les modèles, j'ai utilisé la variable *sexe* qui est enregistrée dans R comme facteur, plutôt que la variable *fille* qui est enregistrée comme étant numérique. Puisqu'il s'agit d'une variable dichotomique, cela revient mathématiquement au même, mais cela a l'avantage de faciliter la création de certains graphiques. Si la variable sexe avait eu 3 catégories (disons garçons, filles et autres), alors la variable numérique n'aurait pas été équivalente à la variable enregistrée en facteur et il aurait impérativement fallu utiliser la variable en facteur dans les modèles.

3.5 La régression linéaire simple avec une variable continue

Jusqu'ici, notre variable dépendante a été dichotomique (garçon = 0, fille =1). Un coefficient de régression nous donne le mouvement dans la variable dépendante associée *au mouvement d'une unité de la variable indépendante*. Les coefficients des variables dichotomiques sont plus «faciles» à interpréter puisqu'ils s'interprètent tout simplement comme le fait d'être dans un groupe plutôt que dans l'autre, ou la différence entre les deux groupes. Cependant, nous utilisons aussi régulièrement des variables continues, donc des variables dont les valeurs *sont ordonnancées hiérarchiquement* et dont *l'unité de mesure peut être considérée comme standard.*[4] Faisons donc un modèle similaire, mais en utilisant cette fois le nombre d'heures d'étude de nos élèves.

Commençons par visualiser de quoi ont l'air les données en fonction du nombre d'heures étudiées par semaine.

```
plot(heures.etude, notes, main="Notes en fonction des heures étudiées",
     xlim=c(0, 10), ylim=c(20, 100),
    xlab="Heures d'étude", ylab="Notes", pch=19, data=data)
```



Visuellement, il est assez clair que ceux qui étudient davantage ont de meilleures notes. De manière analogue à ce que nous faisons lorsque nous calculons une moyenne pour synthétiser

4. Comme nous l'avons vu plus haut, il existe aussi *les variables ordinales* qui sont ordonnancées hiérarchiquement, mais qui ne sont pas mesurées sur une unité standard. La différence entre les variables numériques et la variable ordinale n'est pas toujours claire et il faut donc faire preuve de jugement lorsque l'on procède à l'analyse statistique. Lorsque l'on détermine qu'une variable doit être traitée comme étant ordinale, elle doit être dichotomisée, comme les variables nominales. Voir la note [^3] pour plus de détails.

une vaste quantité d'information, nous pourrions ici vouloir résumer le plus succinctement possible ce que nous observons chez ces 100 élèves. Bien entendu, chaque individu est intéressant, mais nous nous intéressons ici aux notes moyennes des élèves en fonction des heures d'étude. C'est précisément ce que calcule la régression linéaire.

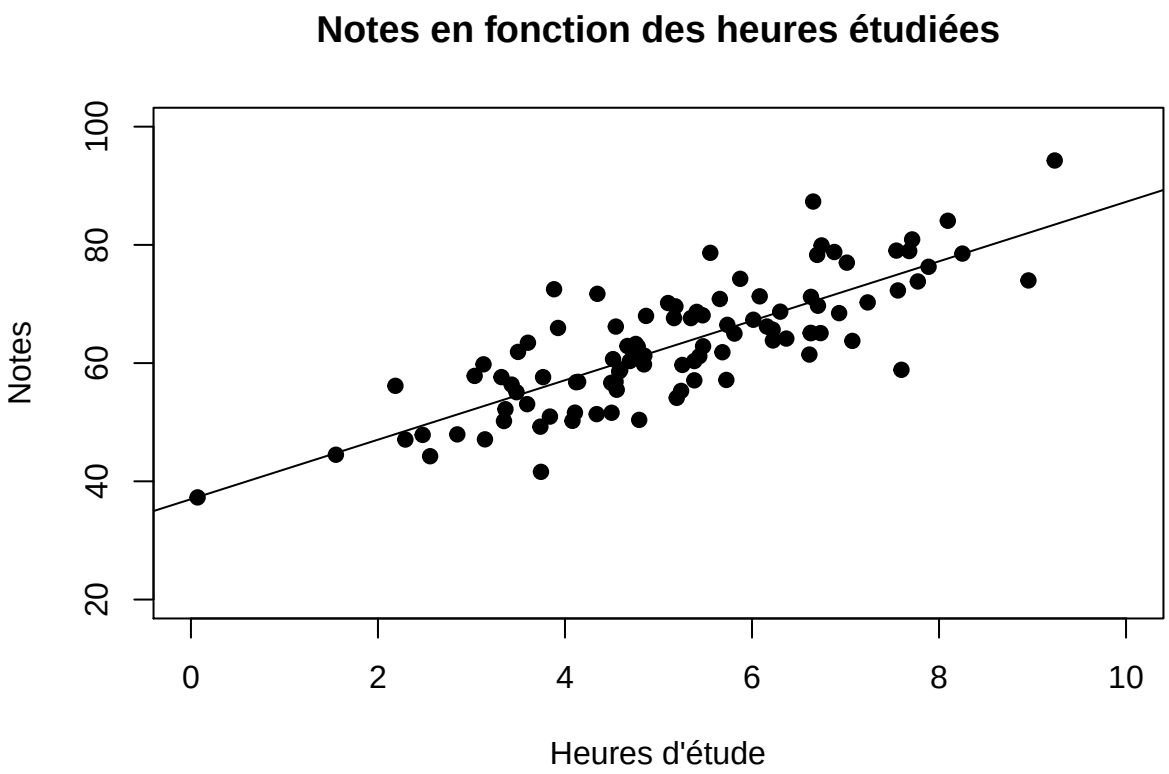
```
m2 <- lm(notes~heures.etude) # Crée l'objet m2 qui contient les résultats
# d'un modèle linéaire (fonction lm() pour "linear model") dans lequel
# la variable note est une fonction (~) de la variable heure.etude.
summary(m2) # Pour voir les résultats contenus dans l'objet m2.
```

```
##
## Call:
## lm(formula = notes ~ heures.etude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.3271  -3.7251  -0.6483   4.1989  16.8874
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    36.981      1.957   18.90   <2e-16 ***
## heures.etude    5.030      0.353   14.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.227 on 98 degrees of freedom
## Multiple R-squared:  0.6744, Adjusted R-squared:  0.6711
## F-statistic:   203 on 1 and 98 DF,  p-value: < 2.2e-16
```

Les résultats de notre modèle nous indiquent que chaque heure d'étude supplémentaire est *en moyenne* associée à des notes plus élevées de 5.0295675. Un élève qui n'a pas étudié obtient donc, en moyenne, la valeur de l'intercept a, soit de 36.9810477. Un élève qui a étudié 1 heure obtient en moyenne $a + \beta \times 1$, soit $36.9810477 + 5.0295675 = 42.0106152$. Un élève qui a étudié 2 heures obtient en moyenne $a + \beta \times 2$, soit $36.9810477 + 5.0295675 \text{ X } 2 = 47.0401827$. Un élève qui a étudié 8 heures obtient en moyenne $a + \beta \times 8$, soit $36.9810477 + 5.0295675 \text{ X } 8 = 77.2175877$.

Voici notre ligne de régression par rapport à nos points. Si vous observez où passe cette ligne, vous constaterez qu'elle est directement sur les «valeurs prédites» que nous avons calculées au paragraphe précédent.

```
plot(heures.etude, notes, main="Notes en fonction des heures étudiées",
     xlim=c(0, 10), ylim=c(20, 100),
    xlab="Heures d'étude", ylab="Notes", pch=19, data=data,
    abline(lm(notes~heures.etude)))
```



Tout comme une moyenne résume en un seul chiffre la valeur centrale d'une variable, l'équation de régression linéaire résume en quelques coefficients la relation entre deux ou plusieurs variables. Cette équation correspond tout simplement à une ligne que l'on trace au milieu des points. Encore une fois, l'équation ne signifie absolument pas que le phénomène qui nous intéresse obéit à une loi immuable de l'univers. L'équation ne fait que *résumer* et *décrire* ce que nous observons dans les données.

La ligne de régression est calculée de manière à *minimiser* la *somme des erreurs au carré*. La somme des erreurs au carré réfère à la somme totale des *résidus*, soit la distance de chaque point par rapport à la ligne de régression. Cette distance est ensuite mise au carré. Puis, nous faisons la même chose pour tous les points et nous additionnons ces distances pour obtenir la *somme des erreurs au carré*. Le fait que la régression linéaire minimise cette somme est importante parce que cela veut dire qu'il n'existe pas de meilleure ligne qui puisse décrire ces données. Il n'y a pas de meilleure ligne parce que c'est celle qui est le plus près possible de tous les points à la fois. Vous vous rappelez peut-être que nous disions plus haut que la moyenne a une propriété similaire : une moyenne minimise la somme des *écarts à la moyenne au carré*. Le principe est ici le même.

Évidemment, l'équation de régression ne décrit pas parfaitement la relation, il y a des points qui ne sont pas directement sur la ligne (il y a des «résidus», des écarts entre chaque point et la ligne), mais souvenez-vous qu'il en était de même pour la moyenne d'un groupe et chaque individu. La régression résume la relation entre nos deux variables de manière à ce que nous soyons capables de manipuler cette information. Encore une fois, il n'y a rien de magique à cela, nous ne faisons que *décrire* nos données à l'aide d'une équation. J'espère qu'au passage vous commencez à arrêter de faire de l'angoisse devant les équations !

3.6 La régression linéaire avec plusieurs variables indépendantes

Jusqu'à présent, nous n'avons utilisé qu'une seule variable indépendante à la fois. Nous avons vu qu'une régression linéaire avec une variable dichotomique comme prédicteur nous permettait facilement de retrouver les moyennes de nos deux groupes (garçons et filles). Nous avons aussi vu comment une régression similaire qui utilise une variable continue comme prédicteur nous permet aussi de résumer simplement le lien entre une unité de ce prédicteur et la variable dépendante. Nous pourrions cependant nous intéresser au lien spécifique d'une, mais variable en tenant compte d'une ou de plusieurs autres variables. C'est ce que permet la régression multiple.

Cela est important parce que, si par exemple les filles ont tendance à étudier plus d'heures que les garçons et que le nombre d'heures passées à étudier est lié aux notes obtenues, on peut alors se demander si c'est réellement le fait d'être une fille qui fait en sorte qu'elles ont de meilleures notes, ou si cela est simplement lié au fait qu'elles étudient plus. Le problème que nous nous posons ici est similaire à l'exemple de l'ampleur des dégâts lors d'un incendie, du nombre de pompiers qui le combattent et de l'intensité d'un incendie donné plus haut. Nous nous demandons si la relation entre l'une ou autre de nos variables d'intérêts est réelle ou fallacieuse.

Voyons dans nos données le nombre d'heures d'étude en fonction du sexe.

```
# Tableau croisé des heures d'étude moyennes selon le sexe
data %>%
  group_by(sexe) %>%
  summarise_each (funs(mean) , h.etude_moy = heures.etude)
```

```
## # A tibble: 2 x 2
##   sexe   h.etude_moy
##   <fct>        <dbl>
## 1 garçon        4.36
## 2 fille         5.96
```

Nous voyons dans nos données que les filles étudient davantage. Les garçons étudient en moyenne 4.36 heures alors que les filles étudient 5.95 heures. Ainsi, on pourra se demander si, dépendamment du nombre d'heures étudiées, les filles obtiennent quand même de meilleurs résultats. Entre d'autres mots, nous pouvons nous demander si une fille qui a étudié 5 heures a en moyenne la même note qu'un garçon qui a étudié lui aussi 5 heures, ou si elle a quand même une meilleure note du simple fait d'être une fille. Pour répondre à cette question, nous allons estimer l'équation de régression linéaire suivante :

$$Y = a + \beta_1 X_1 + \beta_2 X_2 + e$$

Ici, a est encore une fois l'intercept, β_1 est le coefficient de régression de la variable dichotomique *sexe*, β_2 est le coefficient de régression de la variable continue _heures.etude et e est le terme d'erreur individuel, ou ce qui n'est pas expliqué par nos autres variables. Pour aider le lecteur, on pourrait écrire l'équation comme suit :

$$Y = a + \beta_1 sexe + \beta_2 heure.etude + e$$

β_1 et β_2 sont indicés de 1 et 2 pour bien les distinguer, mais les autres variables ne sont pas indicées de i tout simplement parce que l'indice est sous-entendu et ne porte pas à confusion. Il est évident que le sexe et les heures d'étude varient entre les individus.

Estimons maintenant cette équation dans R.

```
m3 <- lm(notes ~ sexe + heures.etude) # Créer l'objet m1 qui contient
# les résultats # d'un modèle linéaire (fonction lm() pour "linear model")
# dans lequel la variable note est une fonction (~) du sexe
# ET de heure.etude.
summary(m3)
```

```
##
## Call:
## lm(formula = notes ~ sexe + heures.etude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6705  -4.2615  -0.6981   3.9705  16.0712
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.3057     1.9227  19.403   <2e-16 ***
## sexefille      3.1023     1.3758   2.255   0.0264 *
## heures.etude   4.6372     0.3872  11.977   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.101 on 97 degrees of freedom
## Multiple R-squared:  0.6906, Adjusted R-squared:  0.6843
## F-statistic: 108.3 on 2 and 97 DF,  p-value: < 2.2e-16
```

Nos résultats nous indiquent donc qu'indépendamment du nombre d'heures d'étude, les filles ont en moyenne des notes de 3.1023461 points supérieurs aux garçons. Par ailleurs, en tenant compte de la différence entre les garçons et les filles, chaque heure d'étude supplémentaire apporte en moyenne 4.6372014 points de plus. À partir de ces résultats, il est possible d'estimer la note d'un garçon qui a étudié 5 heures en complétant simplement notre équation :

$$Y = a + \beta_1 sexe + \beta_2 heure.etude$$

$$Y = 37.3057061 + 3.1023461 sexe + 4.6372014 heure.etude$$

$$Y = 37.3057061 + 3.1023461 \times 0 + 4.6372014 \times 5$$

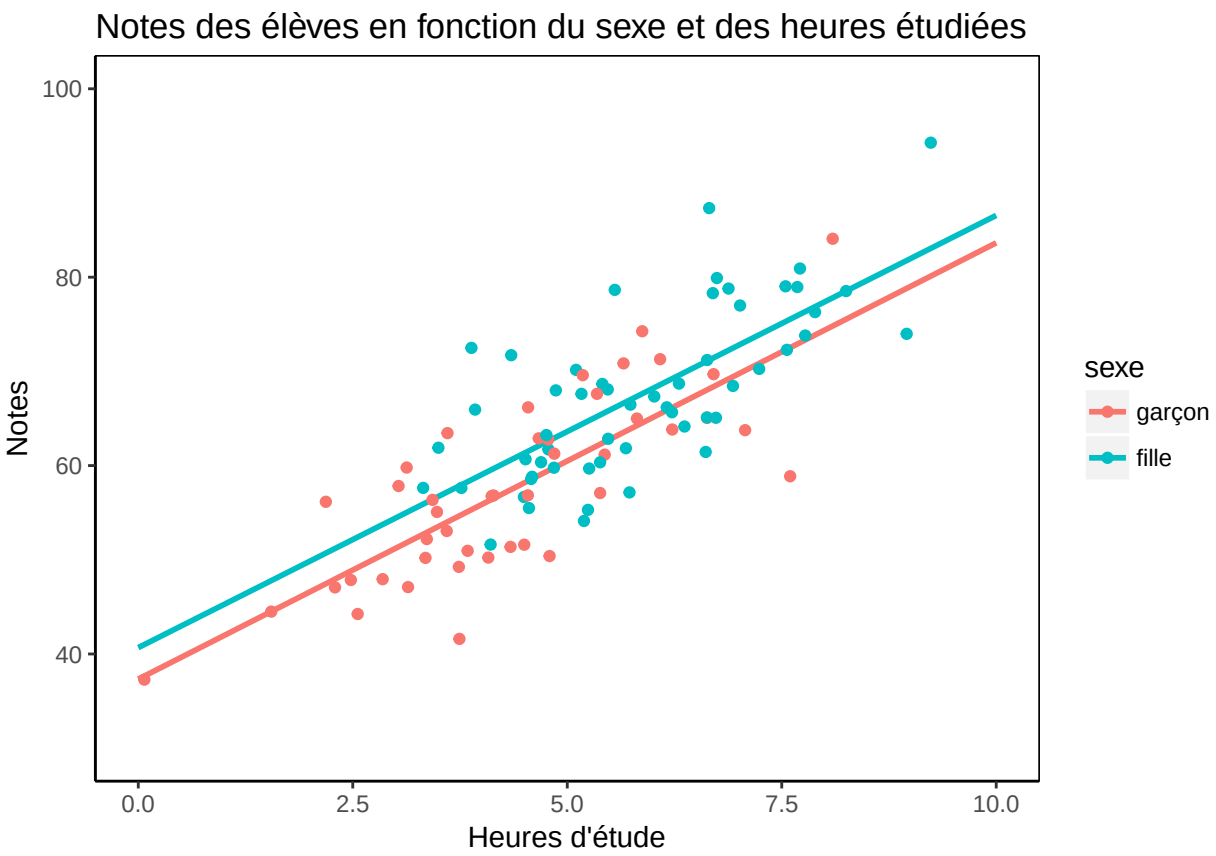
En résolvant l'équation, on arrive à 60.4917131. Si au lieu d'un garçon on s'intéressait à la note moyenne d'une fille qui a elle aussi étudié 5 heures, il suffit encore de résoudre l'équation en ajustant les chiffres.

$$Y = 37.3057061 + 3.1023461 \times 1 + 4.6372014 \times 5$$

Nous obtenons alors 63.5940592. La différence entre le garçon et la fille qui ont étudié le même nombre d'heures (63.5940592 -60.4917131 = 3.1023461) équivaut alors précisément à notre coefficient β_1 de 3.1023461.

Il peut être difficile de visualiser des résultats de régression multiple sur une surface en deux dimensions. Nous pourrions faire un graphique en trois dimensions, mais la surface sur laquelle il est projeté (du papier ou un écran) demeure bidimensionnelle. La meilleure manière de s'y prendre est d'ajouter de la couleur.

```
library(ggplot2) # charger ggplot2, super package pour les graphiques
# Notez que le package ggplot2 permet l'ajout d'un +
# en fin de ligne pour lui dire qu'une autre ligne ajoutant des
# éléments à la figure suit. Un + n'est pas nécessaire si
# une parenthèse est déjà ouverte et n'a pas encore été fermée.

# Notez aussi comment la syntaxe est écrite. Les indentations n'ont aucun
# effet sur le résultat, mais ils améliorent la lisibilité. Il est
# beaucoup plus facile de lire une longue parenthèse dans une
# syntaxe lorsque celle-ci est indentée à partir de la première
# parenthèse. Voir après «theme(». Toutes les options dans cette
# parenthèse sont indentées sous celle-ci. Cela est simplement une
# question de lisibilité. Le tout pourrait être écrit en une seule ligne
# interminable (et difficile à lire). Mieux vaut l'écrire en plusieurs
# lignes.
ggplot(data, aes(heures.etude, notes, colour=sexe)) +
  ggtitle("Notes des élèves en fonction du sexe et des heures étudiées") +
  xlim(0, 10) +
  ylim(30, 100) +
  ylab("Notes") +
  xlab("Heures d'étude") +
  geom_smooth(method = lm, se=FALSE, fullrange=TRUE) + # lignes de régression
  geom_point() + # les points individuels
  theme(panel.grid.major = element_blank(), # Ce qui suit est esthétique
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black"),
        panel.border = element_rect(fill=NA,color="black",
                                    size=0.3, linetype="solid")
        )
```



Nous voyons dans cette figure le résultat de notre régression. Le bleu correspond aux filles et le rouge aux garçons. Nous voyons que leurs droites de régression ont la même pente (β_2) et augmentent de 4.6372014 pour chaque heure d'étude supplémentaire. Cependant, la droite des filles est plus élevée de 3.1023461 par rapport à celle des garçons (β_1) qui débute à l'intercept a, soit à 37.3057061. En observant les points bleus, nous voyons qu'il n'y a aucune fille dans nos données qui a étudié 0 heure (en fait, aucune fille n'a étudié moins de 3 heures), mais nos résultats nous permettent *d'estimer* la note éventuelle d'une fille qui n'aurait pas étudié. Encore une fois, on trouverait cette valeur précise en résolvant notre équation de régression :

$$Y = 37.3057061 + 3.1023461 \times 1 + 4.6372014 \times 0$$

Puisque les heures étudiées sont de 0 (4.6372014 X 0 = 0), cela se simplifie en :

$$Y = 37.3057061 + 3.1023461 \times 1$$

Ce qui nous donne 37.3057061. Évidemment, il faut être prudent lorsque nous prédisons de la sorte des valeurs en dehors de nos données, mais l'objectif est ici de vous montrer la logique.

3.7 La régression linéaire, un exemple avec des données réelles

Nous avons jusqu'ici utilisé des données fictives que nous avons nous-mêmes créées afin de mieux comprendre l'outil de la régression. Cependant, dans la vraie vie nous utilisons

évidemment des «vraies» données. Nous allons ici voir comment une analyse de régression linéaire se fait «dans la vraie vie». Ce faisant, nous verrons qu'il faut préalablement «nettoyer» un peu les données en recodant convenablement les variables avant de procéder. J'utiliserai ici les données des *Études électorales canadiennes* de 2015. Ces données sont produites lors de chaque élection fédérale et elles sont rendues disponibles gratuitement sur le site internet de l'étude.

Comme nous l'avons fait plus haut, chargeons maintenant ces données.

```
#rm(list = ls())

# Pour charger des données du format Stata, nous avons besoin du package
# "readstata13" qui inclut les fonctions de transformation nécessaires.
#library(readstata13)

# Chargeons les données.
#ces15 <- read.dta13("CES2015-phone-release/CES2015_CPS-PES-MBS_complete-v2.dta")

library(haven)
ces15 <- read_sav("~/Documents/Data/CES Data/CES2015-phone-release/CES2015_CPS-PES-MBS_c
```

Nous nous intéresserons à l'appréciation de Justin Trudeau et nous utiliserons ce que nous appelons sa «mesure thermométrique». Nous demandons simplement aux répondants de dire, sur une échelle de 0 à 100 où 0 veut dire qu'ils ne l'aiment pas du tout et 100 veut dire qu'ils l'aiment beaucoup. La question est formulée comme suit :

« Maintenant les chefs de partis. Utiliser la même échelle, où zéro veut dire que vous N'AIMEZ VRAIMENT PAS DU TOUT un chef, et cent veut dire que vous L'AIMEZ VRAIMENT BEAUCOUP.

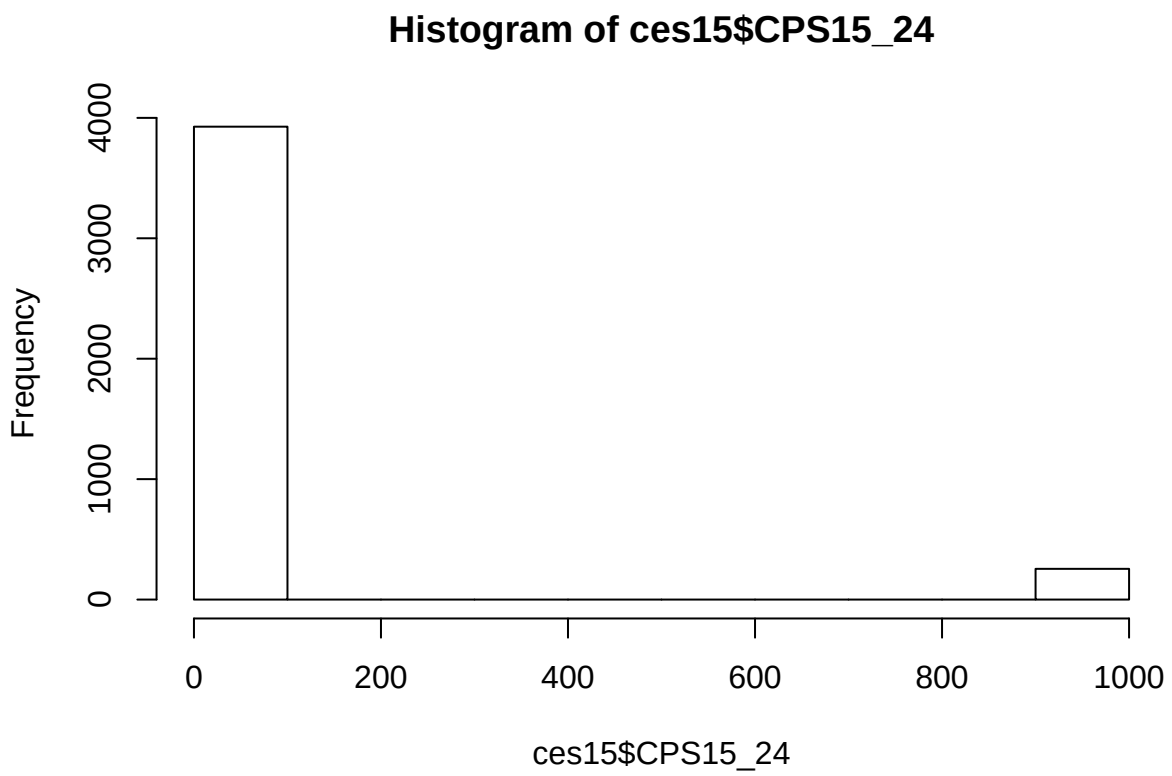
Que pensez_vous de JUSTIN TRUDEAU ?»

Dans les données, cette variable est nommée «CPS15_24», qui est un titre très informatif... Voyons voir de quoi la variable a l'air.

```
summary(ces15$CPS15_24)
```

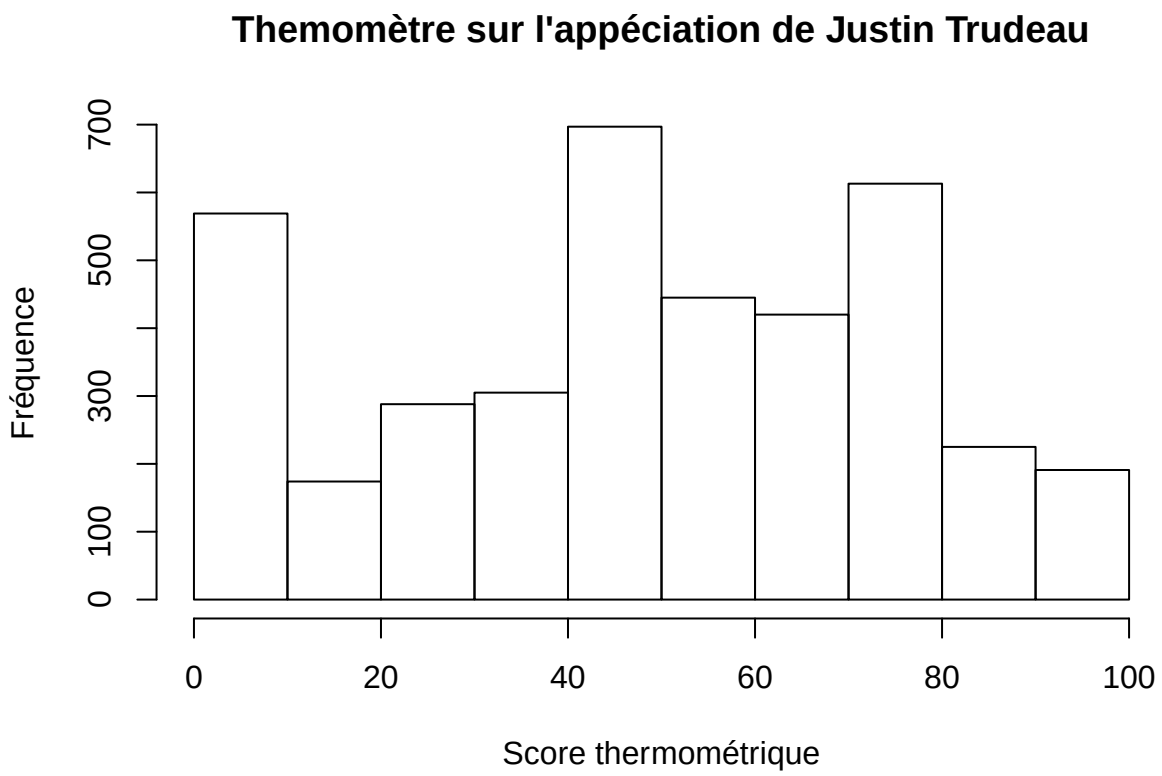
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       0      35      60     109      75     999      20
```

```
# Faire un histograme
hist(ces15$CPS15_24)
```



On remarque que pour une variable qui est censée aller de 0 à 100, nous avons des valeurs étrangement élevées. Cela arrive fréquemment. Un coup d'oeil à la documentation fournie avec les données nous indique que les «Ne sais pas» ont été codés 998, et les «Refus» ont été codés 999. R assume que ces réponses sont «valables» et il faut donc nettoyer cette variable en codant ces réponses de manières appropriées. Ici, nous traiterons ces réponses comme des valeurs manquantes, qui sont codées dans R par «NA».

```
ces15$thermo.trudeau <- as.numeric(ces15$CPS15_24) # Créé la variable
# "thermo.truceau" dans la matrice ces15 en copiant le contenu
# de la variable CPS15_24 dans la matrice data.ces
# J'ai ici utilisé l'option as.numeric() parce que CPS15_24 était
# encodé dans un format "labelled" et nous préférons ici avoir
# simplement les valeurs numériques.

# Recoder les valeurs de la variable thermo.trudeau pour que ce qui
# est plus grand que 100 soit transormé en valeur manquante.
ces15$thermo.trudeau[ces15$thermo.trudeau > 100] <- NA

# Faire un histogramme et préciser les titres
hist(ces15$thermo.trudeau,
  main = "Themomètre sur l'appéciation de Justin Trudeau",
  xlab = "Score thermométrique",
  ylab = "Fréquence"
)
```




```
# On aurait pu faire la même chose sans placer la nouvelle variable
# thermo.trudeau dans la matrice de données initiale ces15.
# thermo.trudeau serait alors un nouvel objet en lui-même. Comme ceci:

thermo.trudeau <- ces15$CPS15_24
thermo.trudeau[thermo.trudeau > 100] <- NA

# Cependant, il vaut souvent mieux garder les variables à l'intérieur
# de matrices pour garder de l'ordre dans notre session. Éliminons
# donc l'objet "thermo.trudeau" seul, et utilisons sa copie située dans
# la matrice ces15

rm(thermo.trudeau)
```

Imaginons maintenant que nous nous intéressions à la perception de Trudeau chez les répondants en fonction de leur niveau d'éducation. La variable d'éducation dans les données est nommée «CPS15_79», encore une fois un nom très utile...

```
table(ces15$CPS15_79)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   98   99
##    5   30   65  321  765  233  930  285 1007  360  165   11   25
```

Encore une fois, cette variable a des codes étranges (98 et 99). Un coup d'oeil à la documentation nous informe sur comment cette variable a été codée :

Quel est le plus haut niveau d'éducation que vous avez complété ?
1 aucune scolarité 2 quelques années à l'élémentaire
3 école élémentaire terminée 4 quelques années d'école secondaire 5 école secondaire terminée
6 quelques études au collège, au cégep, au collège classique 7 études terminées au collège, au cégep, au collège classique
8 quelques études universitaires
9 baccalauréat
10 maîtrise
11 diplôme professionnel ou doctorat 98 Ne sais pas 99 Refus

Nous devons donc encore une fois «nettoyer» cette variable, minimalement en recodeant les 98 et 99 en tant que valeurs manquantes.

```
ces15$educ <- ces15$CPS15_79 # Je créé encore une fois une nouvelle variable
# "educ" dans l'objet ces15qui est une copie de CPS15_79. Elle aura un nom
# plus informatif
ces15$educ[ces15$educ > 11] <- NA # Change les valeurs plus grandes que 11 pour NA
table(ces15$educ)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11
##    5   30   65  321  765  233  930  285 1007  360  165
```

Nous pourrions maintenant vouloir une variable d'éducation qui regroupe certains niveaux d'études. Nous pourrions vouloir avoir quatre niveaux d'éducation plutôt que 11.

```
ces15$educ.gp <- ces15$educ # Copie la variable "educ" dans la nouvelle
# variable "educ.gp"

# Créé les différentes catégories
attach(ces15)
ces15$educ.gp[educ.gp < 5] <- "Moins que secondaire"
ces15$educ.gp[educ.gp == 5] <- "Secondaire complété"
ces15$educ.gp[educ.gp > 5 & educ.gp <= 8] <- "Post-secondaire"
ces15$educ.gp[educ.gp > 8] <- "Diplôme universitaire"
detach(ces15)

table(ces15$educ.gp)
```

```
##
## Diplôme universitaire  Moins que secondaire       Post-secondaire
##                  1532                   421                  1448
##   Secondaire complété
##                   765
```

```
# On voit que R ordonnance les catégories par ordre alphabétique.

# Cependant, nous pouvons aussi voir que la variable est encore une
```

```
# de de classe "labelled" et cela posera problème plus tard.
summary(ces15$educ.gp)
```

```
##    Length     Class      Mode
##      4202  labelled character
```

```
# Nous préférons avoir une variable de classe facteur.
# Je la modifie donc d'abord.
ces15$educ.gp <- as.factor(ces15$educ.gp)

# Si on veut les replacer en ordre "logique" de niveau d'étude
# on fait ceci:

ces15$educ.gp <- factor(ces15$educ.gp,levels(ces15$educ.gp)[c(2,4,3,1)])
table(ces15$educ.gp)
```

```
##
##  Moins que secondaire   Secondaire complété       Post-secondaire
##                   421                   765                  1448
## Diplôme universitaire
##                  1532
```

Finalement, nous pourrions aussi vouloir tenir compte du sexe des répondants. Comme nous l'avons vu plus haut, la variable est RGENDER, que nous avons déjà «nottoyé» plus haut comme suit :

```
table(ces15$RGENDER)
```

```
##
##    1    5
## 1930 2272
```

```
ces15$sexe <- factor(ces15$RGENDER,
                          labels = c("Homme", "Femme"))

table(ces15$sexe)
```

```
##
## Homme Femme
##  1930  2272
```

Nous avons maintenant nettoyé plusieurs variables. Nous avons notre variable dépendante, la mesure thermométrique de Justin Trudeau, la variable de sexe, et deux versions de la variable d'éducation. L'une des versions («educ») va de 1 à 11 avec tous les niveaux d'étude, l'autre a quatre catégories qui regroupent ces différents niveaux («educ.gp»).

Commençons par faire un modèle linéaire où nous incluons notre variable d'éducation de 0 à 11 comme variable indépendante à la mesure thermométrique de Justin Trudeau.

```r
m1.ces15 <- lm(data = ces15, thermo.trudeau ~ educ)
summary(m1.ces15)
```

```
##
## Call:
## lm(formula = thermo.trudeau ~ educ, data = ces15)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.615 -18.808   1.923  21.231  57.615
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   40.962      1.606   25.50  < 2e-16 ***
## educ           1.423      0.213    6.68 2.73e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.57 on 3899 degrees of freedom
##   (301 observations deleted due to missingness)
## Multiple R-squared:  0.01131,    Adjusted R-squared:  0.01106
## F-statistic: 44.62 on 1 and 3899 DF,  p-value: 2.733e-11
```

Nous voyons que, suivant ces résultats, une unité de plus sur notre variable d'éducation est associée à un score thermométrique de 1.4 plus élevé. Suivant le codage de notre variable d'éducation, une unité correspond ici à chacun des 11 niveaux de notre variable.

Faisons le même modèle, mais avec notre variable d'éducation en quatre groupes.

```r
m2.ces15 <- lm(data = ces15, thermo.trudeau ~ educ.gp)
summary(m2.ces15)
```

```
##
## Call:
## lm(formula = thermo.trudeau ~ educ.gp, data = ces15)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.632 -19.543   0.948  20.368  55.457
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     44.543      1.429  31.173  < 2e-16 ***
## educ.gpSecondaire complété       4.509      1.765   2.555  0.01067 *
## educ.gpPost-secondaire           6.145      1.612   3.813  0.00014 ***
## educ.gpDiplôme universitaire    10.089      1.601   6.301 3.28e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.56 on 3897 degrees of freedom
##   (301 observations deleted due to missingness)
## Multiple R-squared:  0.01242,    Adjusted R-squared:  0.01166
## F-statistic: 16.34 on 3 and 3897 DF,  p-value: 1.503e-10
```

Ici, comme notre variable d'éducation a 4 groupes distincts et que nous avons encodé notre variable comme un facteur, R dichotomise automatiquement chacune des catégories de cette variable et laisse l'une des catégories en dehors du modèle. Cette catégorie, ici ceux qui ont un niveau inférieur aux études secondaires, devient le groupe de référence, qui sera capturé par l'intercept.

Notre modèle nous indique donc que ceux qui ont un niveau inférieur aux études secondaires donnent en moyenne un score de 44.5 à Trudeau, ceux qui ont un secondaire complété donnent 4.5 points de plus (donc 44.5+4.5= 49), ceux qui ont un niveau post-secondaire donnent 6.1 points de plus (donc 44.5+6.1= 50.6), et ceux qui ont un diplôme universitaire donnent en moyenne 10 points de plus (donc 44.5+10= 54.5).

Faisons maintenant un modèle qui inclut la variable de sexe, à des fins de simplicité, nous utiliserons la variable d'éducation allant de 0 à 11.

```r
m3.ces15 <- lm(data = ces15, thermo.trudeau ~ educ + sexe)
summary(m3.ces15)
```

```
##
## Call:
## lm(formula = thermo.trudeau ~ educ + sexe, data = ces15)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.922 -19.059   2.589  21.625  60.258
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  38.3031     1.6733  22.891  < 2e-16 ***
## educ          1.4389     0.2123   6.778 1.40e-11 ***
## sexeFemme     4.7914     0.8813   5.437 5.76e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.47 on 3898 degrees of freedom
##   (301 observations deleted due to missingness)
## Multiple R-squared:  0.01875,    Adjusted R-squared:  0.01825
## F-statistic: 37.25 on 2 and 3898 DF,  p-value: < 2.2e-16
```

Dans ce dernier modèle, nous voyons d'abord que, en tenant compte du sexe, chaque niveau d'éducation supplémentaire est associé à une hausse de 1.4 point d'appréciation de Trudeau. Par ailleurs, les femmes apprécient Trudeau plus que les hommes, de 4.78 points en moyenne.

L'intercept correspond ici au score moyen donné par les individus qui ont un score de 0 sur les deux variables indépendantes. Avoir un score de 0 sur la variable sexe signifie qu'on est un homme, mais notez qu'il n'est pas possible d'avoir un score de 0 sur la variable d'éducation que nous avons utilisé ici, puisque celle-ci va de 0 à 11. Il arrive fréquemment que l'intercept ne correspondent à rien de possible et cela n'a pas réellement de conséquence mathématiquement parlant. Il faut simplement faire plus attention lorsque l'on interprète les résultats.

Cela illustre aussi qu'il faut bien réfléchir à comment on code nos variables. Nous aurions pu ici décider de coder notre variable d'éducation pour qu'elle varie entre 0 et 10 (plutôt que 1 à 11) et alors la valeur de 0 sur cette variable (et donc l'intercept) aurait pu correspondre à des cas réels dans les données. Encore une fois, ce codage n'a pas de conséquence autre que la facilité avec laquelle on peut interpréter les résultats. Les résultats ne sont pas soudainement «faux» ou «mauvais» pour autant, ils sont simplement moins intuitifs à interpréter. J'espère aussi que vous voyez pourquoi il vaut souvent mieux coder les variables pour que 0 soit possible.

Vous avez certainement remarqué que R nous donne aussi plusieurs informations quant à l'erreur standard, aux statistiques t, aux valeurs de P et qu'il y a parfois des étoiles à côté des coefficients. Ces informations ont trait à *l'inférence* statistique et nous traiterons de cela plus loin. Nos données viennent d'un échantillon, et nous voulons inférer quelque chose à la population entière. Or avant d'inférer, il faut décrire adéquatement. Donc pour l'instant, assurez-vous de bien saisir ce que les coefficients de régression veulent dire.

Finalement, on pourrait vouloir créer un beau tableau de régression avec ces trois modèles qui pourrait être intégré à un document MS Word. Le package *texreg* nous permettra de le faire.

```
library(texreg)
```

```
## Version:  1.36.23
## Date:     2017-03-03
## Author:   Philip Leifeld (University of Glasgow)
##
## Please cite the JSS article in your publications -- see citation("texreg").
```

```
htmlreg(list(m1.ces15, m2.ces15, m3.ces15), # On inclut nos 3 modèles
        file = "tableau1.doc", # On veut créer cela dans le fichier tableau.doc
        inline.css = FALSE, doctype = TRUE, # Nécessaire pour WORD
        head.tag = TRUE, body.tag = TRUE, html.tag = TRUE, #pour WORD
       stars = 0.05, # Niveau de significativité désiré pour les étoiles
       digits=2, # Combien de chiffres après la virgule
       caption="La perception de Justin Trudeau", # Titre
```

Table 1 – La perception de Justin Trudeau

	Model 1	Model 2	Model 3
Éducation (1-11)	1.42*		1.44*
	(0.21)		(0.21)
Groupe d'éducation			
Secondaire complété		4.51*	
		(1.77)	
Post-secondaire		6.14*	
		(1.61)	
Diplôme universitaire		10.09*	
		(1.60)	
Sexe (Femme)			4.79*
			(0.88)
Intercept	40.96*	44.54*	38.30*
	(1.61)	(1.43)	(1.67)
R^2	0.01	0.01	0.02
Adj. R^2	0.01	0.01	0.02
Num. obs.	3901	3901	3901
RMSE	27.57	27.56	27.47

*$p < 0.05$

```
       caption.above = T, # Titre en haut du tableau
       custom.coef.names = c("Intercept",   # Adapter les noms des variables
                             "Éducation (1-11)",
                             "Secondaire complété",
                             "Post-secondaire",
                             "Diplôme universitaire",
                             "Sexe (Femme)"),
      reorder.coef=c(2,3,4,5,6,1),  # Réordonner les variables dans le tableau
      groups = list("Groupe d'éducation" = 2:4), # Créé un groupe
      include.rsquared = T, # Inclu le R-carré
      model.names=c("Modèle 1", "Modèle 2", "Modèle 3") # Nom des modèles
      )
```

```
## The table was written to the file 'tableau1.doc'.
```

Le tableau sera alors sauvegardé dans le document «tableau1.doc» et placé dans notre «working directory». Si comme moi vous utilisez Latex, il est aussi possible avec de légères modifications au code de produire un tableau Latex.

3.8 Les régressions logistiques et multinomiales

Nous avons présenté ici seulement le modèle de régression *linéaire*. Celui-ci ne peut être utilisé que si la *variable dépendante* est une variable continue. Comme nous l'avons déjà dit, il arrive que des variables ordinales soient traitées comme des variables continues. C'est acceptable dans certains cas, et il est aussi possible que cela ne le soit pas dans d'autres cas. Il revient à vous d'en juger et de voir ce qui est accepté dans votre champ.

Ceci étant, les variables dichotomiques ne peuvent pas être modélisées avec un modèle linéaire. Il faut utiliser un modèle logistique ou un modèle probit (qui est très similaire au modèle logistique). Nous ne verrons pas ces modèles ici. Encore une fois, le choix du modèle adéquat se fonde sur la nature de la variable *dépendante*, pas des variables indépendantes.

Les variables nominales doivent quant à elles être modélisées avec un modèle multinomial logistique. Ces modèles peuvent être un peu pénibles à interpréter et pour cette raison, plusieurs préfèrent dichotomiser les variables nominales (une des catégories prend la valeur de 1 et toutes les autres sont codées 0) pour ensuite utiliser un modèle logistique. Cependant, cette procédure n'est pas toujours adéquate et elle est critiquée dans plusieurs situations, notamment quand il est question du choix de vote. C'est à vous de juger du modèle adéquat et de ce qui est accepté dans votre champ de recherche.

Les modèles linaire, logistique, probit et multinomial sont des modèles de base en analyse de régression. Tous les bons livres d'introduction aux statistiques les aborderont. Le lecteur qui souhaite approfondir le sujet peut consulter W. Fox (1999) et Pétry and Gélineau (2003) qui sont en français. En anglais, Tabachnick, Fidell, and Osterlind (2001) et surtout Gelman and Hill (2006) offrent d'excellents tours d'horizon.

J'espère maintenant que vous saisissez bien qu'il n'y a rien de magique dans la régression. Son rôle principal consiste essentiellement à décrire les données de manière à rendre l'information intelligible. Nous sommes incapables de gérer les données de plusieurs variables en même temps (dans l'exemple utilisé ici, cela revient à 3 colones de 100 lignes si vous imaginez un fichier Excel). Nous allons maintenant passer à *l'inférence statistique*, c'est-à-dire le processus par lequel on applique à une population entière quelque chose que l'on observe dans un échantillon.

4 Inférer

Durant les vacances de Noël, j'étais dans un souper familial et nous discutions des élections américaines. Ma tante était debout devant le poêle et, pendant qu'elle faisait réchauffer une soupe, se disait surprise de l'élection de Trump. Après quelques soupirs, elle déclara :

> «En tout cas, moi les sondages, je commence à ne plus trop y croire. Tu ne peux pas prendre juste 1000 personnes sur une population de plusieurs millions et penser prévoir comment ils vont voter».

Puis, après avoir bien brassé sa soupe, elle prit une petite cuillérée pour y goûter et, semblant juger qu'elle était adéquatement assaisonnée, lança avec assurance : «Ouin, elle va être bonne cette soupe-là !». Votre humble serviteur n'a pu s'empêcher de remarquer l'ironie de la situation : ma tante qui procédait avec assurance à l'échantillonnage de sa soupe, à peine quelques secondes après avoir déclaré ne pas croire à l'échantillonnage.

Ma tante a bien brassé la soupe pour que les éléments solides ne soient pas tous dans le fond du chaudron. Autrement dit, elle voulait, consciemment ou non, s'assurer que tous les éléments de sa soupe aient une chance égale d'aboutir dans sa cuiller. Puis elle en a pris une cuillérée et généralisa que ce que sa soupe goûtait dans sa cuiller devait être suffisamment semblable à ce que le reste la soupe dans le chaudron devait goûter. Dans le doute, ou si elle avait besoin de plus de précision, elle aurait pu prendre un second échantillon, mais elle ne l'a pas fait.

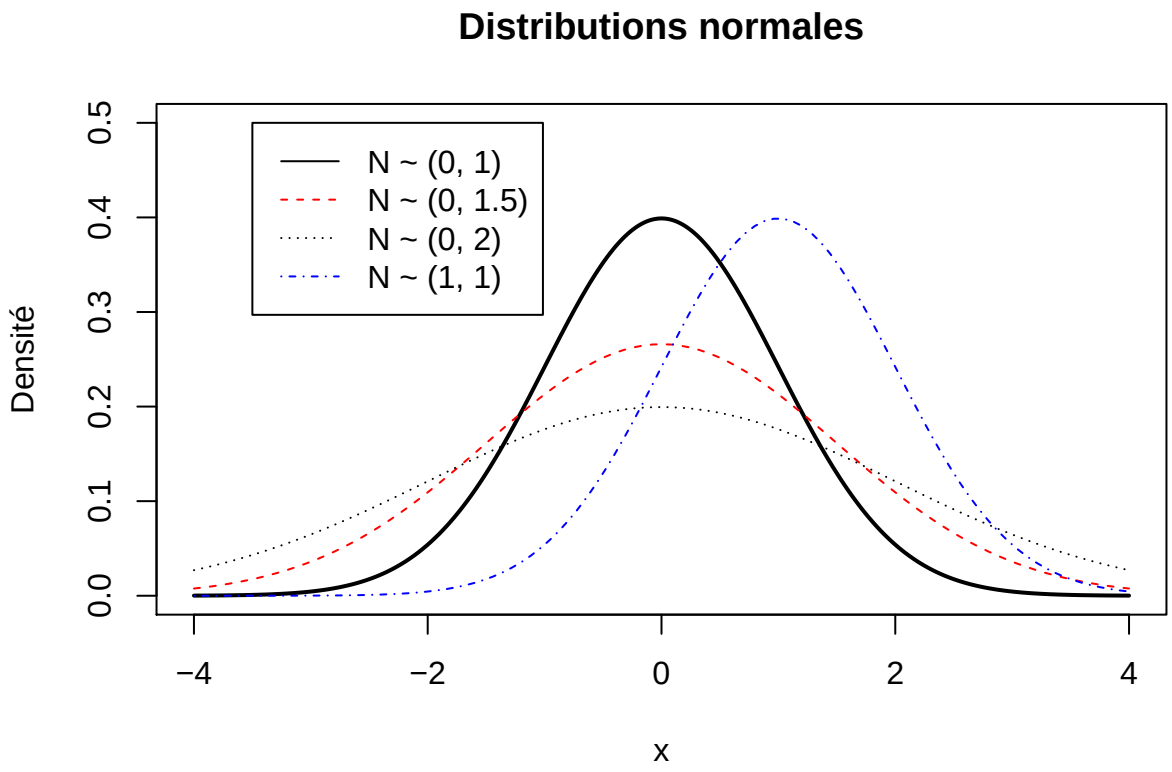
Pourtant, quelqu'un qui ne croit pas à l'échantillonnage aurait littéralement dû goûter à tout le chaudron pour savoir si sa soupe était correctement assaisonnée. Évidemment, aucun invité n'aurait eu de soupe et il est fort probable que ma tante n'aurait alors pas mangé de dinde, mais elle n'aurait alors pas eu à passer par l'échantillonnage. Cette petite anecdote peut vous faire sourire, mais réalisez que vous appliquez tous des techniques d'échantillonnage dans votre quotidien, autrement il vous serait tout simplement impossible d'accomplir plusieurs tâches pourtant simples et banales.

Évidemment, les êtres humains ne sont pas des molécules et il faut donc forcément en tenir compte lorsque nous analysons des statistiques qui sont influencées par le libre arbitre des individus. Cependant, en gardant en tête ces précautions, il demeure que les notions d'inférence statistiques peuvent aussi être appliquées aux comportements humains. Bien comprendre comment fonctionne l'inférence statistique vous permettra ensuite de mieux juger d'une variété de résultats qui, bien que «statistiquement adéquats», peuvent être critiquables étant donné que nous nous intéressons à des humains plutôt qu'à des molécules. Cependant, vous ne pourrez pas réellement poser un jugement informé sur ces situations sans bien comprendre l'inférence statistique. Malheuseuement, plusieurs comprennent mal ce qu'est l'inférence statistique et ont tendance à prendre la «significativité statistique» pour un absolu objectif qui peut se substituer au jugement du chercheur. Ce n'est *jamais* le cas. C'est à vous d'interpréter adéquatement ce que disent les chiffres et le fait que quelque chose soit «statistiquement significatif» ne constitue pas un sceau de véracité.

Afin de mieux saisir comment au juste fonctionne l'inférence statistique, nous allons utiliser l'exemple totalement fictif d'une élection dans laquelle le parti Rouge, dirigé par Gros-Cochon LeTapon-Tâteur, est en compétition avec le parti Bleu, dirigé par Madame Black-Berry duCourriel-Qui-Coule. Nous allons simuler les intentions de vote d'une population de 10 millions d'électeurs et nous prendrons ensuite aléatoirement des sondages de 1000 personnes dans cette population. Nous pourrons donc voir comment les sondages peuvent varier autour des véritables intentions de vote des 10 millions d'électeurs.

4.1 L'inférence statistique, quelques notions théoriques

Toutes les variables ont une certaine distribution qui a une certaine forme. L'une des distributions les plus importantes en statistiques est la distribution normale. Une distribution normale a une moyenne de μ (prononcé «mu») et un écart-type de σ (pronconé «sigma»). Vous verrez souvent l'expression : $N \sim (\mu, \sigma)$ pour dire qu'une variable est normalement distribuée avec une moyenne de μ et un écart-type de σ. Bien sûr, la moyenne et l'écart-type d'une distribution peuvent varier. Voici quatre exemples de distributions normales. Les trois premières distributions ont une moyenne de 0, mais elles varient par leur écart-type ; la distribution en bleu a quant à elle une moyenne de 1 et un écart-type de 1.

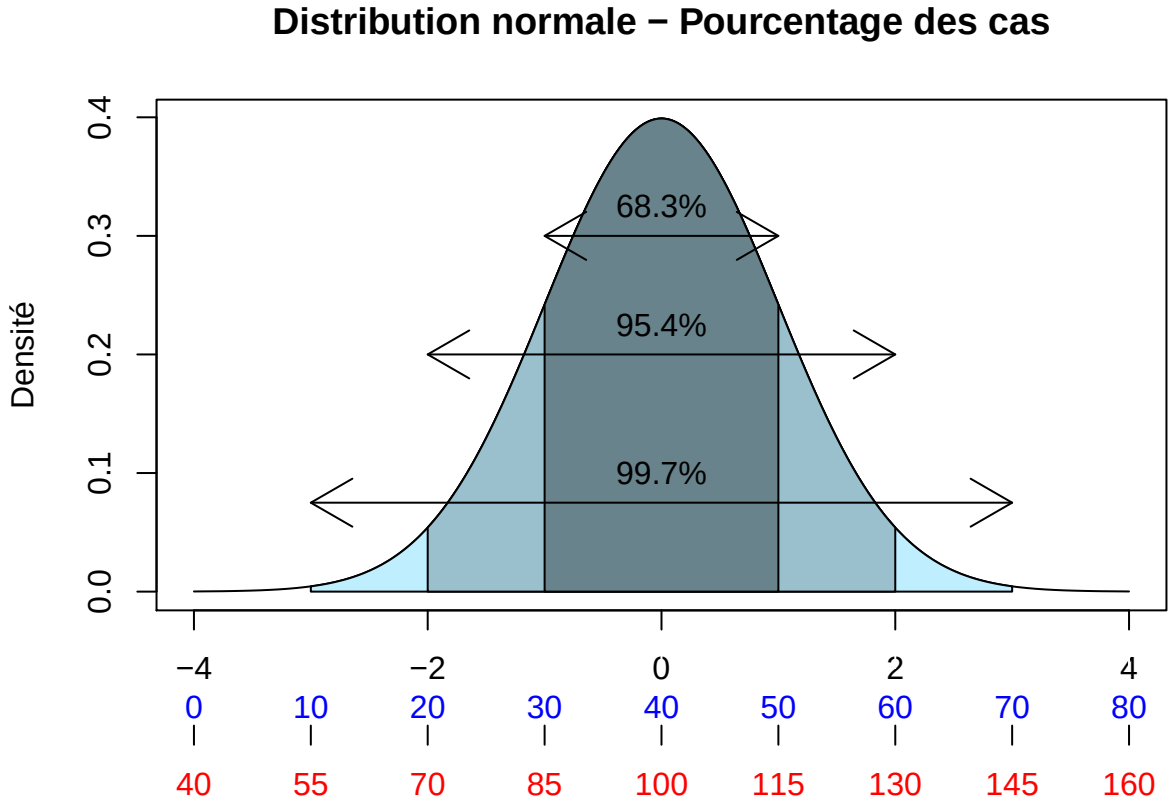


La mesure en unité d'écart-type est standardisée, c'est-à-dire que peu importe l'échelle de mesure originale, la distribution d'une variable peut être ramenée à une échelle mesurée en unité d'écart-type afin de pouvoir adéquatement comparer la distribution d'une variable à une autre variable mesurée différemment [5].

5. L'écart-type est calculé par la racine carrée de la moyenne des carrés des écarts entre chaque observation

Par exemple, la figure suivante montre la distribution de deux variables $N \sim (\mu, \sigma)$, une variable bleue et une variable rouge. L'échelle en noir est en unités d'écarts-types, alors que les échelles en bleu et en rouge peuvent être pensées comme deux possibilités d'échelles «originales» sur lesquelles les variables ont été mesurées. L'échelle en bleu va de 0 à 80 avec une moyenne de 40, celle en rouge de 40 à 160 avec une moyenne de 100 (on peut par exemple songer à la mesure du QI qui a aussi une moyenne de 100 et un écart-type de 15). Mais lorsque ces échelles sont ramenées en unité d'écarts-types, elles peuvent alors être adéquatement comparées. On remarque qu'en plaçant les échelles originales sur une échelle comparable, leur distribution est identique.

La raison tient à l'une des propriétés fondamentales de la loi normale : pour toute combinaison de moyenne et d'écart-type, il y aura toujours une proportion constante de cas entre la moyenne et un point quelconque situé à une certaine distance de la moyenne exprimée en écart-type. En observant la figure, on pourrait donc dire que, dans le cas de l'échelle bleue, il y a 95.4% des cas qui se situent entre 20 et 60, alors que 95% des cas se situent entre 85 et 115 dans le cas de la variable rouge.



Nous savons qu'il y aura toujours exactement 68.3% des cas situés à $\pm 1\sigma$ (plus ou moins 1 écart-type). Il y aura aussi toujours 95.4% des cas situés à $\pm 2\sigma$ et 99.7% des cas situés à $\pm 3\sigma$. Cela sera vrai peu importe l'échelle de mesure originale utilisée pour mesurer une variable.

et la moyenne. Soit : $\sqrt{\frac{\Sigma(x_i-\mu)^2}{N}}$, où χ représente une observation i, μ la moyenne des observations χ_i et N Le nombre de cas.

Cette propriété est vraie pour la distribution illustrée ici, mais elle le serait tout autant pour une distribution qui aurait une apparence plus aplatie. On pourrait «aplatir» la distribution de la figure, mais l'échelle des écarts-types bougerait avec elle. Imaginez que, pour aplatir la distribution, vous tiriez de chaque côté de l'axe des abscisses (x), cela aurait pour effet d'étirer chaque côté de la distribution et donc de l'aplatir, mais l'échelle suivrait elle aussi.

4.1.1 Le théorème de la limite centrale

Nous venons de voir rapidement l'importance de la distribution normale, mais pour comprendre l'inférence statistique, il est important de saisir qu'il existe trois types de distributions.

1. La distribution d'une population (N)
   — Lors d'un rescencement
   — Avec une moyenne μ et un écart-type σ
2. La distribution d'un échantillon (n)
   — Lors d'un sondage
   — Avec une moyenne $\bar{x}$ (prononcé «x-barre») et un écart-type s
3. La distribution de *tous les échantillons possibles* (distribution d'échantillonnage)
   — Toutes les combinaisons possibles des cas qu'il est possible de tirer pour produire un échantillon n d'une population N.

Les deux premiers types de distributions sont relativement intuitifs, le troisième est cependant plus difficile à saisir. Imaginons la population formée des éléments A, B, C, D, E (N=5) et que nous voulions tirer un échantillon de taille 2 (n=2). Dans ce cas, tous les échantillons possibles correspondent à toutes les paires de lettres qu'il est possible de tirer à partir de la population. Dans notre cas, il a a 10 échantillons possibles : AB, AC, AD, AE, BC, BD, BE, CD, CE et DE. Le nombre total d'échantillons possibles peut être calculé à partir du NcN :

$$Ncn = \frac{N!}{n!(N-n)!}$$

Le signe ! signifie «factoriel», il c'est-à-dire que nous multiplions entre eux la suite de nombre qui vient avec le premier nombre jusqu'à arriver à 1. Dans notre exemple, cela donne :

$$\frac{5!}{2!(5-2)!} = \frac{5 \times 4 \times 3 \times 2 \times 1}{2 \times 1 \quad (3 \times 2 \times 1)} = \frac{120}{12} = 10$$

Dans notre exemple, il y a donc 10 échantillons possibles de 2 lettres dans notre population de 5 lettres. Imaginons maintenant une situation habituelle avec un sondage de 1000 répondants (n=1000) pour une population de 5 millions d'électeurs (N=5 000 000). Tous les échantillons possibles signifient alors toutes les possibilités d'échantillons de n=1000 qu'il est possible de tirer d'une population de 5 millions d'individus. Deux échantillons ne différant

que par 1 seul individu sont ici considérés comme deux échantillons possibles distincts. Dans ce cas, le nombre d'échantillons possibles donne :

$$Ncn = \frac{N!}{n!(N-n)!}$$

$$Ncn = \frac{5000000!}{1000!(5000000-1000)!}$$

$$Ncn = 27000000000000000000000000000000000000000000$$

Le nombre d'échantillons possibles est tellement gros que nous ne savons même pas comment au juste nous devrions dire ce nombre. Vous comprenez donc que cette idée de «tous les échantillons possibles» n'est que *théorique* et ne peut jamais réellement être appliquée. Cependant, sans pour autant que nous fassions la démonstration mathématique, sachez simplement que le théorème de la limite centrale nous apprend qu'au fur et à mesure que *le nombre d'échantillons augmente*, la moyenne d'une variété de paramètres de ces échantillons se rapprochera de plus en plus de la «vraie» valeur des paramètres dans la population.

Attention ici, nous ne parlons pas de la moyenne d'une variable *dans* un échantillon, mais de la *moyenne de la moyenne des échantillons*. Si par exemple nous nous intéressons à la taille moyenne des femmes d'une population donnée, nous prenons un premier échantillon de 100 femmes et nous obtenons une moyenne de 167cm. Nous prenons ensuite un second échantillon et nous obtenons une moyenne de 172cm. Un troisième échantillon nous donne une moyenne de 169 cm. À mesure que nous augmentons le nombre d'échantillons, la moyenne de ceux-ci (ici, la moyenne de 167, 172 et 169), qui revient à la moyenne des moyennes *dans* les échantillons_ se rapprochera de plus en plus de la «vraie» valeur dans la population. Mathématiquement, il pourrait être démontré que la moyenne de tous les échantillons possibles (la «moyenne échantillonnale») est égale à la moyenne de la population. Par ailleurs, la *disrtribution* de ces moyennes d'échantillons tendra à devenir normale et cela sera d'autant plus vrai que la taille des échantillons individuels sera grande.

Si la taille de l'échantillon est de 30 ou plus, les trois choses suivantes seront vraies : - La moyenne de la distribution d'échantillonnage est identique à la moyenne de la population ($\mu_{\bar{x}}$=μ) - La distribution d'échantillonnage suit une distribution normale. - Son écart-type sera de $\frac{\sigma}{\sqrt{N}}$

4.2 L'échantillonnage

Afin d'illustrer le théorème de la limite centrale, qui est plus facile à saisir en le «voyant» à l'oeuvre qu'en tentant de le comprendre abstraitement, nous allons simuler des données d'une population de 10 millions de personnes et allons ensuite faire des «sondages» aléatoires dans cette population. Nous utiliserons deux scénarios. D'abord une élection entre deux partis, l'un étant réellement à 55% et l'autre à 45% d'intentions de vote dans la population. Puis nous simulerons une situation où la course entre ces deux partis est serrée, l'un étant réellement à 51% et l'autre à 49%. Nous observerons ce qui se produit lorsque nous augmentons la quantité de «sondages», et aussi de quoi ont l'air ces sondages lorsque le véritable écart entre les deux partis est relativement petit.

4.2.1 Exemple d'une élection à deux partis (55-45)

Nous simulons d'abord les données de la population de 10 millions d'individus

```
# Créons les données de la population entière de 10M d'électeurs.
population <- sample(c(1, 0), size=10000000, rep=TRUE, prob=c(.55,.45))
# L'objet population contient donc 5 millions de lignes (individus) qui votent
# pour les différents partis dans les proportions décrites.

ftable(population) # Voir le nombre d'individus qui votent pour chaque parti.
```

```
## population       0       1
##
##            4499648 5500352
```

```
# Bien sûr, les proportions seraient plus informatives. Demandons-les à R.
prop.table(table(population)) # Pour en voir les proportions
```

```
## population
##         0         1
## 0.4499648 0.5500352
```

Nous voyons donc qu'il y a effectivement 55.00352% des gens qui ont l'entention de voter pour le parti bleu et 44.99648% qui ont l'intention de voter pour le parti rouge. Ce sont ici les *vraies* intentions de vote dans notre *population*. Imaginons maintenant que nous fassions un sondage aléatoire de 1000 individus dans cette population.

```
# Faisons un premier sondage aléatoire de cette population
# avec un échantillon de 1000 personnes.
sondage.1 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.1))
```

```
## sondage.1
##     0     1
## 0.448 0.552
```

Nous obtenons ici qu'une proportion de 0.552 des «répondants» affirment vouloir voter pour le parti bleu et 0.448 pour le parti rouge. Faisons un deuxième sondage aléatoire.

```
# Un deuxième sondage
sondage.2 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.2))
```

```
## sondage.2
##     0     1
## 0.444 0.556
```

Nous obtenons 55.6% pour le parti bleu et 44.4% pour le parti rouge. Faisons un troisième sondage.

```r
# Un troisième sondage
sondage.3 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.3))
```

```
## sondage.3
##     0     1
## 0.452 0.548
```

Ici, nous obtenons que 54.8% des répondants affirment vouloir voter pour le parti bleu, alors que 45.2 disent vouloir voter pour le parti rouge. Tentons un quatrième sondage.

```r
# Un quatrième sondage
sondage.4 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.4))
```

```
## sondage.4
##    0    1
## 0.46 0.54
```

Avec ce quatrième sondage, nous obtenons que 54% des répondants affirment vouloir voter pour le parti bleu, alors que 46. Essayons un dernier sondage.

```r
# Un cinquième sondage
sondage.5 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.5))
```

```
## sondage.5
##     0     1
## 0.437 0.563
```

Avec ce cinquième sondage, nous obtenons que 56.3% des répondants affirment vouloir voter pour le parti bleu, alors que 43.7.

Nous pourrions continuer ainsi à faire des sondages, mais remarquez deux choses : les résultats de chaque sondage ne sont par forcément «pile» sur les «vrais» chiffres dans la population (55% pour les bleus et 45% pour les rouges), mais ils tournent globalement autour de ceux-ci. Aussi, si nous faisons la moyenne des résultats obtenus dans ces cinq sondages, nous obtiendrions un résultat qui se rapprocherait encore plus de la vraie valeur. Essayons-le.

```r
# Moyenne des résultats des 5 sondages pour le parti bleu
mean(prop.table(table(sondage.1))[2],
     prop.table(table(sondage.2))[2],
     prop.table(table(sondage.3))[2],
     prop.table(table(sondage.4))[2],
     prop.table(table(sondage.5))[2]
     )
```

```
## [1] 0.552
```

```r
# Moyenne des résultats des 5 sondages pour le parti rouge
mean(prop.table(table(sondage.1))[1],
     prop.table(table(sondage.2))[1],
     prop.table(table(sondage.3))[1],
     prop.table(table(sondage.4))[1],
     prop.table(table(sondage.5))[1]
     )
```

```
## [1] 0.448
```

Nous n'avons que cinq sondages et notre moyenne n'est donc pas pile sur les vrais résultats de la population, mais nous nous en approchons beaucoup. Imaginons que nous fassions 50 sondages plutôt que seulement 5.

```r
data.sondages <- list()
for(i in 1:50){ # Ceci est un "loop", nous faisons la même commande 50 fois
    data.sondages[[i]] <- sample(population, 1000, replace=FALSE)
}

prop.sondages = vector()
for(i in 1 :50){
    prop.sondages[i] <- mean(data.sondages[[i]])
}
mean(prop.sondages)
```

```
## [1] 0.54922
```

On pourrait difficilement faire mieux. Ceci dit, ici notre exemple posait que les vraies valeurs étaient à 55% et 45%, soit une différence de 10 points de pourcentage entre les deux partis. Dans la réalité, il arrive souvent que les vraies intentions de vote soient beaucoup plus serrées. Imaginons donc que nous une course très serrée dans laquelle le parti Bleu ne mène que par 2 points de pourcentage.

4.2.2 Exemple d'une élection serrée (51-49)

```r
# Créons les données de la population entière de 10M d'électeurs.
population.serre <- sample(c(1, 0), size=10000000, rep=TRUE, prob=c(.51,.49))
# L'objet population contient donc 5 millions de lignes (individus) qui votent
# pour les différents partis dans les proportions décrites.

prop.table(table(population)) # Pour en voir les proportions
```

```
## population
##         0         1
## 0.4499648 0.5500352
```

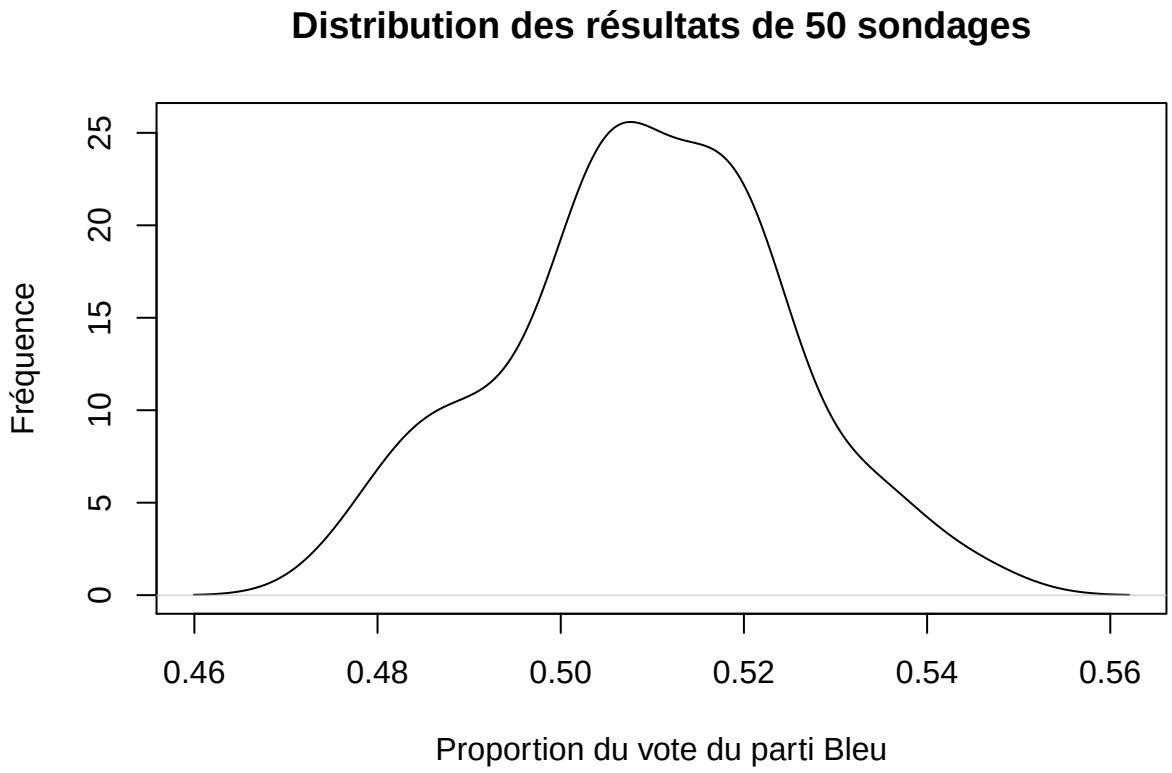
Faisons 50 sondages.

```
data.sondages_serre <- list()
for(i in 1:50){ # Ceci est un "loop", même commande 50 fois i=1:50
    data.sondages_serre[[i]] <- sample(population.serre, 1000, replace=FALSE)
}

prop.sondages_serre = vector()
for(i in 1 :50){
    prop.sondages_serre[i] <- mean(data.sondages_serre[[i]])
}
mean(prop.sondages_serre)
```

```
## [1] 0.50922
```

Vous voyez que nous obtenons une moyenne de nos 50 sondages qui ressemble à ce qui devrait être la vraie valeur de 0.51 pour le parti Bleu. Ceci étant, tous nos sondages n'ont pas, individuellement la même moyenne. Visualisons la distribution de nos sondages pour voir comment se sont répartis les résultats de tous les sondages.

```
densite_50 <- density(prop.sondages_serre)

plot(densite_50,
     main="Distribution des résultats de 50 sondages",
     ylab="Fréquence",
     xlab="Proportion du vote du parti Bleu")
```



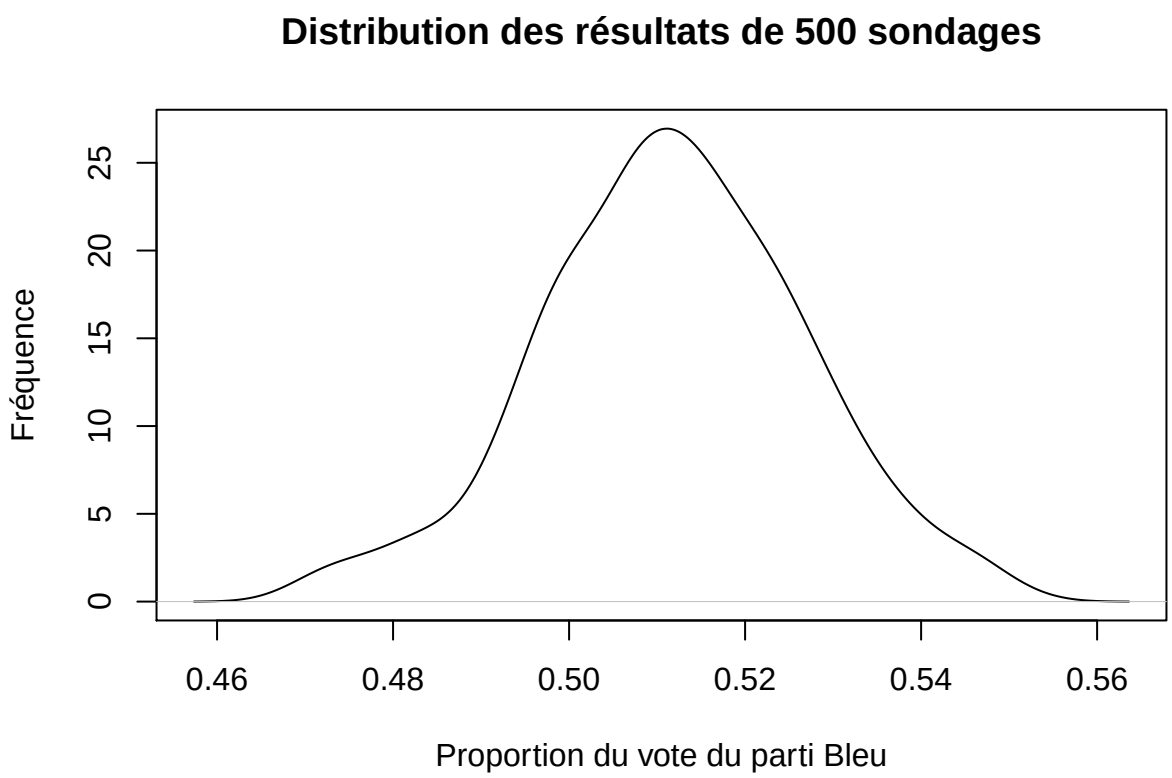
Nous voyons que la distribution des résultats est presque normale, mais pas complètement. La plupart des sondages sont autour de la vraie valeur de 0.51, mais certains en sont plus éloignés. Imaginons maintenant que nous fassions 500 sondages plutôt que 50. Voyons voir.

```
data.sondages_serre_500 <- list()
for(i in 1:500){
    data.sondages_serre_500[[i]] <- sample(population.serre, 1000, replace=FALSE)
}

prop.sondages_serre_500 = vector()
for(i in 1:500){
    prop.sondages_serre_500[i] <- mean(data.sondages_serre_500[[i]])
}
mean(prop.sondages_serre_500)
```

```
## [1] 0.511752
```

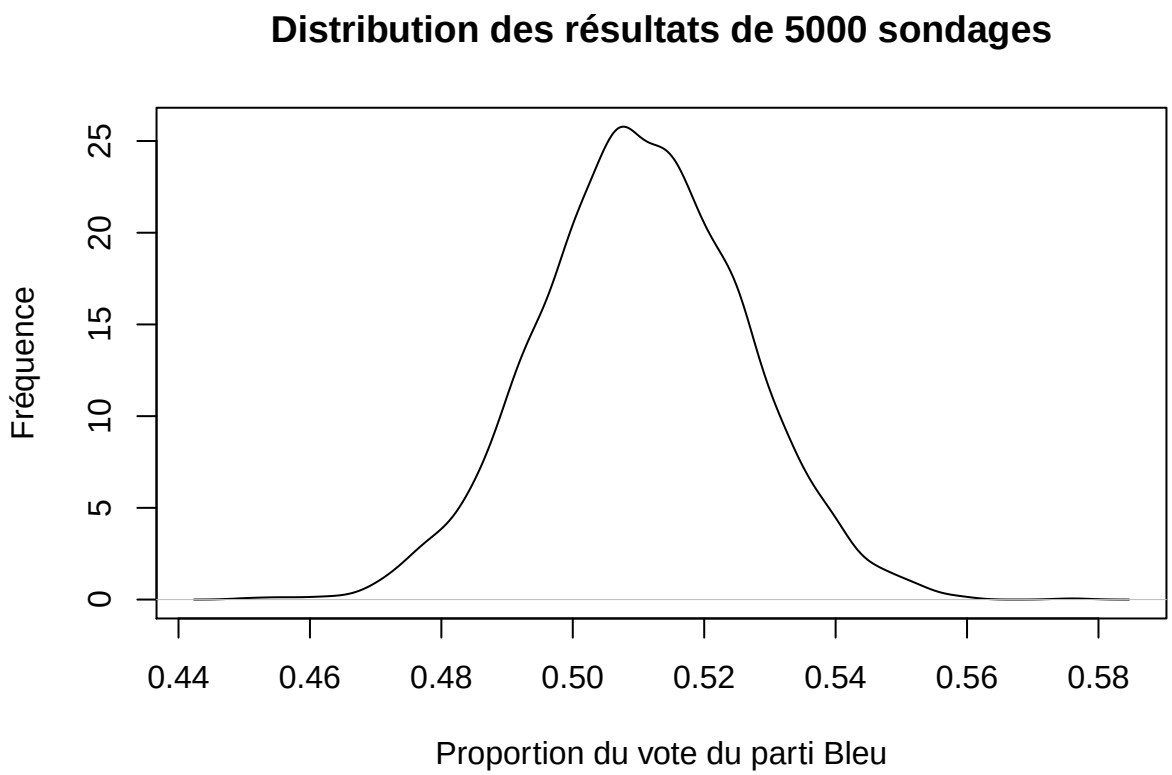
Encore une fois, vous voyez que nous obtenons une moyenne très près de ce qui devrait être la vraie valeur de 0.51 pour le parti Bleu. Visualisons encore la distribution pour voir comment se sont répartis les résultats de tous les sondages.

```
densite_500 <- density(prop.sondages_serre_500)

plot(densite_500,
     main="Distribution des résultats de 500 sondages",
     ylab="Fréquence",
     xlab="Proportion du vote du parti Bleu")
```



Clairement, cette distribution est beaucoup plus près d'une distribution normale que ce que nous observions avec «seulement» 50 sondages. Essayons avec 5000.

```
data.sondages_serre_5000 <- list()
for(i in 1:5000){
    data.sondages_serre_5000[[i]] <- sample(population.serre, 1000, replace=FALSE)
}

prop.sondages_serre_5000 = vector()
for(i in 1:5000){
    prop.sondages_serre_5000[i] <- mean(data.sondages_serre_5000[[i]])
}

densite_5000 <- density(prop.sondages_serre_5000)
plot(densite_5000,
     main="Distribution des résultats de 5000 sondages",
     ylab="Fréquence",
     xlab="Proportion du vote du parti Bleu")
```



Clairement, la distribution de tous les sondages commence à prendre la forme d'une distribution normale avec une moyenne centrée autour de la vraie valeur de 0.51. Remarquez aussi que quelques sondages s'écartent de cette valeur, il y en a par exemple qui placent le parti Bleu à 0.46 et d'autres à 0.56, mais la très vaste majorité des sondages sont rapprochés de la vraie valeur de 0.51. Remarquez aussi qu'avec une différence de seulement 2 points de pourcentage entre les deux partis (dans la population), il y a quand même une grande quantité de sondages qui donnent moins de 50% des voix au parti Bleu, même si nous savons que ce dernier devrait être à 51%. Également, si nous comparons cette distribution avec 5000 sondages avec celle que nous avons obtenu avec 500 sondages, nous voyons aussi que la distribution avec 5000 sondages est plus resserrée, son écart-type est plus faible.

Cette démonstration devrait vous aider à saisir pourquoi il ne faut pas trop s'affoler avec des mouvements de 1, 2 ou 3 points de pourcentages d'un sondage à l'autre. Deux sondages menés la même journée et exactement de la même manière pourront donner des résultats différents. Ces variations inévitables sont le simple fait des aléas de l'échantillonnage statistique. Si l'on prend également en compte que, contrairement à nos 10 millions d'individus fictifs, les vrais citoyens peuvent décider de ne pas répondre (donc l'échantillonnage n'est jamais parfaitement aléatoire parce que la non-réponse peu être plus probable chez certains groupes) et que les maisons de sondage peuvent poser les questions de manière légèrement différente (ce qui peut également entraîner de légères variations dans les résultats), on comprendra qu'il n'y a vraiment pas de quoi écrire à sa mère pour des variations de quelques points de pourcentage.

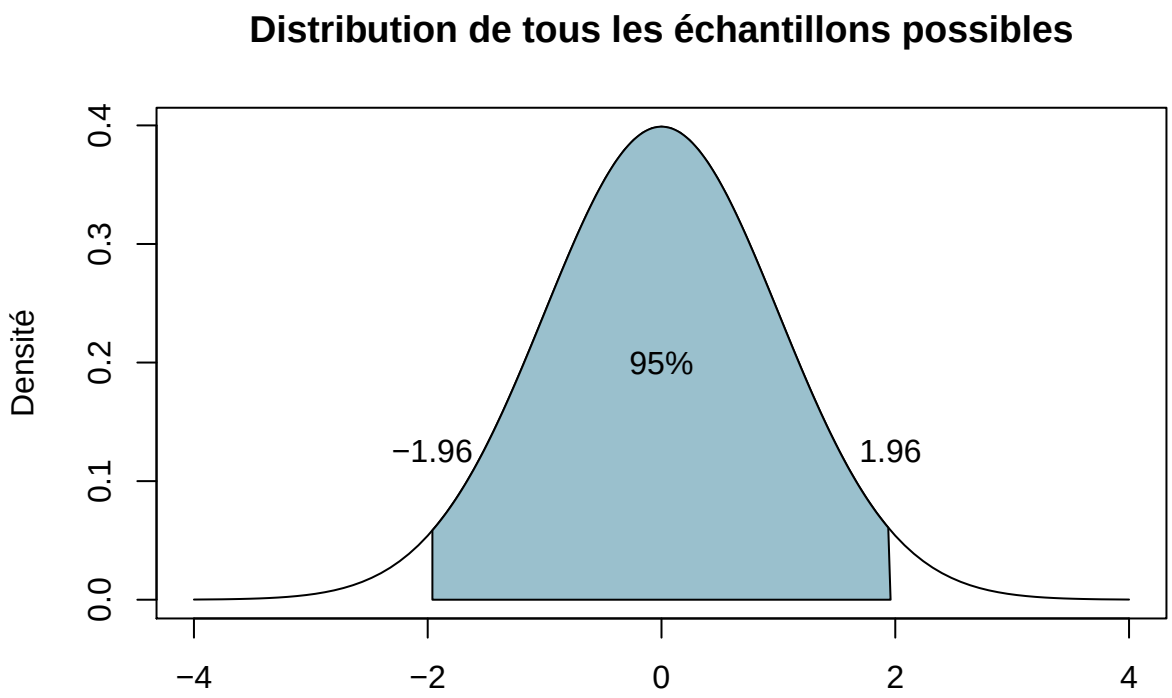
4.3 Du sondage à la population

Évidemment, on ne fait jamais 5000 sondages, ni même 500. La plupart du temps, nous n'avons qu'un seul sondage à partir duquel on veut tirer des inférences sur une population entière. Comme nous l'avons vu, bien que les résultats des sondages tendent à se rapprocher des véritables valeurs d'une population, les aléas de l'échantillonnage font en sorte que les valeurs que nous obtenons dans un sondage donné dévient des vraies valeurs. Ainsi, il est plus prudent de construire une marge d'erreur autour des résultats du sondage.

4.3.1 La marge d'erreur

Heureusement, grâce au théorème de la limite centrale, il est possible calculer avec précision un intervalle à l'intérieur duquel le véritable paramètre de la population devrait se retrouver, ce à un degré de confiance désiré. Généralement, le degré de confiance que nous visons est de 95%, c'est-à-dire que nous acceptons qu'il y ait 5% des chances que l'on se trompe en affirmant que le véritable paramètre de la population se retrouve à l'intérieur de l'interval que nous aurons calculé. Il convient cependant de remarquer que cette «ligne de démarcation» n'est que conventionnelle et a été fortement contestée récemment (voir notamment Benjamin et al. (2017) et la réponse de Lakens et al. (2017)).

Grâce au théorème de la limite centrale, nous savons que 95% des échantillons possibles donneront un pourcentage situé à plus ou moins 1.96 écart-types du vrai pourcentge. Nous savons cela grâce à la propriété de la distribution normale discutée plus haut : pour chaque combinaison de moyenne et d'écart-type, il y aura toujours une proportion constante de cas entre la moyenne et un point quelconque situé à une certaine distance de la moyenne exprimée en écart-type. Ici, nous voulons être certain à 95% de certidue, et donc nous savons que 95% des cas sont situés à l'intérieur de plus ou moins 1.96 écart-types. La figure suivante illustre la chose pour une distribution échantillonale ayant une moyenne de 0. 95% de tous les échantillons possibles nous donnerons un résultat situé dans la zone bleue.



Suivant cela, si nous voulons estimer le pourcentage π d'une population, nous pouvons le faire en utilisant le pourcentage p obtenu dans un sondage.[6] L'équation suivante permet de calculer la marge d'erreur autour de notre pourcentage p à l'intérieur de laquelle nous avons 95% des chances de trouver le véritable pourcentge π dans la population.

$$\pi = p \pm 1,96\sigma_p$$

Où l'écart-type de la distribution échantillonnale σ_p est estimé à partir de l'écart-type s dans notre sondage.

$$\sigma_p = \frac{s}{\sqrt{n}}$$

4.3.1.1 Test de la marge d'erreur du pourcentage (55-45)

```
# avec un échantillon de 1000 personnes.
sondage.1 <- sample(population, 1000, replace=FALSE)
prop.table(table(sondage.1))
```

```
## sondage.1
##     0     1
## 0.454 0.546
```

```
# Pour avoir l'écart-type
sd(sondage.1) # sd() pour standard deviation
```

6. Les équations pour l'estimation d'une moyenne sont similaires. Il suffit de remplacer les signes appropriés, ce qui donne $\mu = \bar{x} \pm 1.96\sigma_{\bar{x}}$, où $\sigma_{\bar{x}} = \frac{s}{\sqrt{n}}$

```
## [1] 0.4981286
```

```
# Calculon Sigma_p
sigma_p <- sd(sondage.1)/sqrt(1000)

# Calculons l'intervalle de confiance à 95%
low_bound <- mean(sondage.1)-1.96*sigma_p
low_bound
```

```
## [1] 0.5151257
```

```
hi_bound <- mean(sondage.1)+1.96*sigma_p
hi_bound
```

```
## [1] 0.5768743
```

Donc, suivant notre calcul sur un sondage dans lequel le parti bleu avait 56% des intentions de vote, l'interval bas est de 0.5151257 et l'interval haut de 0.5768743. Suivant notre scénario, nous savons que le pourcentage π réel d'intention de vote dans la population est de 55%, ce qui est bel et bien à l'intérieur de notre interval de confiance. Évidemment, ici nous avons l'avantage de connaître la véritable valeur du pourcentage de la population puisque nous l'avons nous-mêmes créé, ce n'est évidemment normalement pas le cas.

4.3.1.2 Test de la marge d'erreur du pourcentage (51-49)

Juste histoire de s'amuser, tentons aussi l'expérience avec notre scénario d'élection serrée.

```
# avec un échantillon de 1000 personnes.
sondage.1_s <- sample(population.serre, 1000, replace=FALSE)
prop.table(table(sondage.1_s))
```

```
## sondage.1_s
##     0     1
## 0.453 0.547
```

```
# Pour avoir l'écart-type
sd(sondage.1_s) # sd() pour standard deviation
```

```
## [1] 0.4980352
```

```
sigma_p <- sd(sondage.1_s)/sqrt(1000)
mean(sondage.1_s)
```

```
## [1] 0.547
```

```
# Calculons l'intervalle de confiance à 95%
low_bound.s <- mean(sondage.1_s)-1.96*sigma_p
low_bound.s
```

```
## [1] 0.5161315
```

```
hi_bound.s <- mean(sondage.1_s)+1.96*sigma_p
hi_bound.s
```

```
## [1] 0.5778685
```

Ici, suivant notre calcul sur un sondage dans lequel le parti codé 1 avait 54.7 % des intentions de vote, l'intval bas est de 0.5161315 et l'interval haut de 0.5778685. Encore une fois, la vraie valeur du parti codé 1 est bien dans notre interval. Cependant, on remarque aussi que cet interval inclu la vraie valeur des intentions de vote du parti codé 0. Nous voyons donc que lorsque nous sommes devant une situation où les intentions de votes réelles dans la population sont très serrée, il est statistiquement très difficile d'établir avec confiance qu'un parti est réellement en avant d'un autre avec un seul sondage. la raison n'a ici rien à voir avec la fiabilité des méthodes statistiques elles-mêmes, mais est tout simplement lié à la nature même de la précision d'un échantillon et au fait que la précision requise pour «prédire» adéquatement le gagant d'une élection est beaucoup plus élevée quand une élection est serrée que lorsque'elle ne l'est pas.

4.3.2 La différence entre deux groupes : Le Test t

Le test-t est un test qui nous permet de déterminer si une différence observée entre deux groupes dans un échantillon peut être généralisée à l'ensemble de la population. Ici, plutôt que d'établir un intervalle à l'intérieur duquel on devrait raisonnablement s'attendre à trouver un paramètre d'intérêt dans notre population, on cherche plutôt à vérifier si la différence que l'on observe entre nos deux groupes dans le sondage existe bel et bien aussi dans la population. Autrement dit, nous nous intéressons à la *signification statistique* de la différence que nous observons dans l'échantillon.

Dans le cas de l'estimation de l'intervalle de confiance autour d'un pourcentage vu plus haut, nous n'avions pas réellement d'hypothèse à tester. Nous savions que le résultat d'un échantillon ne nous donne qu'une approximation de la valeur du paramètre dans la population et avons donc voulu avoir une approche plus rigoureuse pour estimer cette vraie valeur. Ici cependant, nous observons une différence entre deux groupes et nous voulons savoir si elle existe vraiment dans la population.

Intuitivement, puisque nous observons une différence entre nos deux groupes, nous pourrions dire que notre hypothèse est que les deux groupes doivent réellement être différents dans la population. Donc, mathématiquement :

$$\mu_1 \neq \mu_2$$

Or, il n'y a pas de manière formelle de tester réellement cette hypothèse directement. Par contre, si nous l'inversons en formulant l'hypothèse nulle selon laquelle il n'y a pas de différence entre nos deux groupes, alors nous pouvons tester *l'improbabilité* de $\mu_1 \neq \mu_2$. Donc, mathématiquement nous cherchons à vérifier si :

$$\mu_1 = \mu 2 \quad ou \quad \mu_1 - \mu 2 = 0$$

Si, suivant nos données, il est jugé trop *improbable* que $\mu_1 - \mu 2 = 0$ (moins de 5% des chances), alors inversement il sera crédible de croire que $\mu_1 \neq \mu_2$.

Poser l'hypothèse nulle de cette manière est utile parce que, suivant notre connaissance de la distribution normale et du théorème de la limite centrale, nous savons de quoi devraient avoir l'air 95% des échantillons possibles dans une situation ou l'hypothèse nulle est vraie dans la population. Nous savons que les aléas de l'échantillnage produiront de la variation autour de la vraie valeur, mais nous savons de quoi devraient avoir l'air cette variation. En ce sens, si ce que nous observons dans nos données n'est *pas* conforme à ce que nous pourrions raisonnablement nous attendre d'observer dans 95% des échantillons possibles tirés d'une population où il n'y a *pas* de différence entre les deux groupes, alors nous jugerons crédible qu'il y ait réellement une différence dans la population.

Pour renforcer encore ce point : nous savons que dans une population ou $mu_1 = mu_2$, 95% des échantillons possibles nous donneront un résultat qui sera dans la zone bleue de la figure de distribution présentée plus haut. Si le résultat que nous observons se trouve à l'extérieur de cette zone bleue, dans les extrémités gauche et droite de la distribution, alors nous jugerons **qu'il est trop improbable de tomber sur un échantillon qui nous montre une telle différence entre μ_1 et μ_2 s'il n'y en a pas dans la population. Nous jugerons alors que l'hypothèse nulle est trop improbable, donc peu crédible, et nous la rejetterons. Nous dirons donc que la différence que nous observons dans notre échantillon est «statistiquement significative»**.

Revenons maintenant au tout premier exemple que nous avons utilisé au tout début à propos des notes de 100 étudiants. Imaginons que ces 100 étudiants soient un échantillon provenant d'une population de 10 000 étudiants ayant complété une épreuve ministérielle. Nous pourrions alors vouloir vérifier si la différence que nous avons observé entre les garçons et les filles était le seul fait de cet échantillon ou si cette différence existe réellement dans la population entière. Nous pourrions alors faire le tests-t.

```
# Histoire de se raffraîchire la mémoire
head(data)
```

```
##      notes   sexe heures.etude fille
## 1 56.36004 garçon     3.429696     0
## 2 70.16088  fille     5.102714     1
## 3 61.71704  fille     4.780638     1
## 4 70.84951 garçon     5.656184     0
## 5 72.49611  fille     3.883307     1
## 6 84.08077 garçon     8.094016     0
```

```
# Différence fille-garçon dans les notes?
t.test(data$notes~data$sexe) # ~ pcq sexe est facteur binaire
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  data$notes by data$sexe
## t = -5.4541, df = 92.215, p-value = 4.102e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -14.333948  -6.681456
## sample estimates:
## mean in group garçon  mean in group fille
##             57.52824             68.03595
```

```
# Différence fille-garçon dans le temps d'étude?
t.test(data$heures.etude~data$sexe)
```

```
##
##  Welch Two Sample t-test
##
## data:  data$heures.etude by data$sexe
## t = -4.9518, df = 90.354, p-value = 3.399e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.2376079 -0.9562823
## sample estimates:
## mean in group garçon  mean in group fille
##             4.360936             5.957881
```

Un test-t vérifie l'improbabilité d'une valeur d'un échantillon en faisant l'hypothèse que cette valeur est *tirée d'une population dans laquelle cette valeur est de 0* (hypothèse nulle). La logique est très eaxtement la même lorsque nous évaluons la signification statistique des coefficients en régression linéaire. Visualisons les trois modèles que nous avions fait plus haut.

```
summary(m1)
```

```
##
## Call:
## lm(formula = notes ~ sexe)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.2474  -7.3035  -0.6993   6.0207  26.5525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   57.528      1.441  39.933  < 2e-16 ***
## sexefille     10.508      1.925   5.458 3.63e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 9.556 on 98 degrees of freedom
## Multiple R-squared:  0.2331, Adjusted R-squared:  0.2253
## F-statistic: 29.79 on 1 and 98 DF,  p-value: 3.627e-07
```

```
summary(m2)
```

```
##
## Call:
## lm(formula = notes ~ heures.etude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.3271  -3.7251  -0.6483   4.1989  16.8874
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    36.981      1.957   18.90   <2e-16 ***
## heures.etude    5.030      0.353   14.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.227 on 98 degrees of freedom
## Multiple R-squared:  0.6744, Adjusted R-squared:  0.6711
## F-statistic:   203 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(m3)
```

```
##
## Call:
## lm(formula = notes ~ sexe + heures.etude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.6705  -4.2615  -0.6981   3.9705  16.0712
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.3057     1.9227  19.403   <2e-16 ***
## sexefille      3.1023     1.3758   2.255   0.0264 *
## heures.etude   4.6372     0.3872  11.977   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.101 on 97 degrees of freedom
## Multiple R-squared:  0.6906, Adjusted R-squared:  0.6843
```

```
## F-statistic: 108.3 on 2 and 97 DF,  p-value: < 2.2e-16
```

Notez que R nous renvoie une valeur t. Un coefficient est jugé significatif s'il est trop peu probale que le coefficient obtenu dans un échantillon soit tiré d'une population dans lequel il a la valeur de 0.

Vous voyez donc que le fait que quelque chose soit «statistiquement significatif» n'est absoluement pas une sorte de sceau magique démontrant la véracité absolu de quelque chose. Il s'agit tout simplement d'un critère conventionnel (95%) fondé sur l'improbabilité d'observer quelque chose dans un échantillon tiré d'une population dans laquelle nous faisons l'hypothèse que ce que nous pensons observer est faux. Au final, il s'agit donc simplement de savoir si ce que nous décrivons dans notre échantillon est aussi probablement vrai dans la population qui nous intéresse. Que ce soit dans un échantillon ou dans une population, ce n'est pas parce que deux groupes sont différents l'un de l'autre que cette différence est réellement *causée* par l'appartenance de groupe. Le fait que quelque chose soit «statistiquement significatif» veut simplement dire qu'il est très improbable que nous observions un certaine chose dans un échantillon si cela n'est pas aussi vrai dans la population. Au-delà d'établir l'association entre les deux phénomènes dans la population, la signification statistique ne nous dit rien de la causalité à proprement parler. En ce sens, il ne faut jamais interpréter la signification statistique comme établissant un rapport causal.

Références

Benjamin, Daniel J., James O. Berger, Magnus Johannesson, Brian A. Nosek, E.-J. Wagenmakers, Richard Berk, Kenneth A. Bollen, et al. 2017. "Redefine Statistical Significance." *Nature Human Behaviour* 2 (1). Springer Nature : 6–10. doi :10.1038/s41562-017-0189-z.

Favre, Pierre. 2005. *Comprendre Le Monde Pour Le Changer*. Paris : Presses de Sciences Po.

Fox, William. 1999. *Statistiques Sociales*. Presses Université Laval.

Gelman, Andrew, and Jennifer Hill. 2006. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge university press.

Lakens, Daniel, Federico Adolfi, Casper Albers, Farid Anvari, Matthew Apps, Shlomo Argamon, Marcel van Assen, et al. 2017. "Justify Your Alpha." PsyArXiv, –. doi :10.17605/osf.io/9s3y6.

Pétry, François, and François Gélineau. 2003. *Guide Pratique d'introduction à La Régression En Sciences Sociales*. Presses Université Laval.

Tabachnick, Barbara G, Linda S Fidell, and Steven J Osterlind. 2001. *Using Multivariate Statistics*. Allyn ; Bacon Boston.

Tobias, Sheila. 1993. *Overcoming Math Anxiety*. WW Norton.