Introduction aux techniques d'enquête

Christophe Chesneau

▶ **To cite this version:**

Christophe Chesneau. Introduction aux techniques d'enquête. Master. France. 2022. hal-03883289

HAL Id: hal-03883289
https://cel.hal.science/hal-03883289

Submitted on 3 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

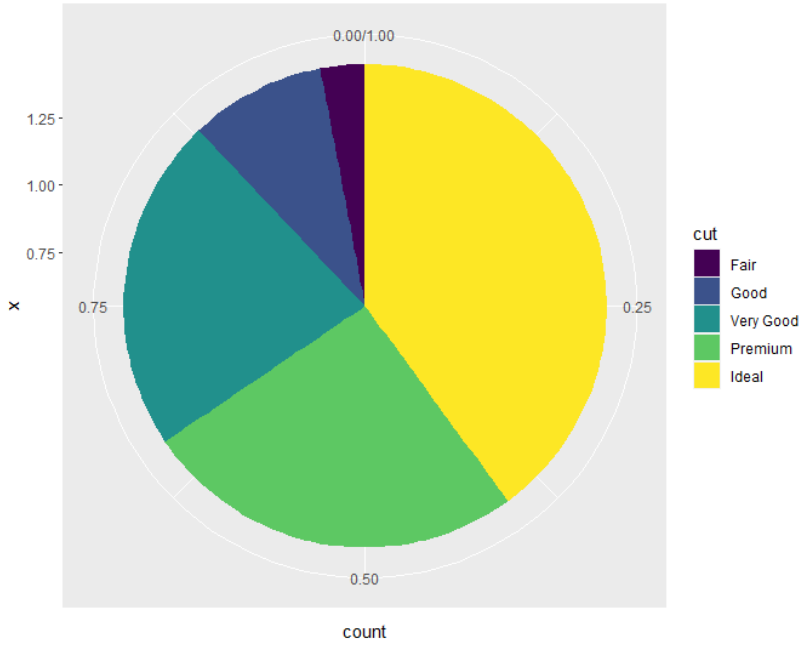
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Caen-Normandie

Introduction aux techniques d'enquête

Christophe Chesneau

https://chesneau.users.lmno.cnrs.fr/



Caen, le 10 Octobre 2022

Table des matières

~ Note ~

Ce document propose une introduction aux techniques d'enquête.

Le logiciel utilisé est R.

Je remercie chaleureusement Monsieur Bruno Dardaillon (DREAL) pour les documents de travail fournis et les projets proposés dans le dernier chapitre.

N'hésitez pas à me contacter pour tout commentaire :

christophe.chesneau@gmail.com

Bonne lecture !

1 Introduction

1.1 Exemples

Quelques exemples de résultats liés à des sondages sont donnés ci-dessous :

- Le salaire moyen pour une première embauche d'un jeunes diplômé (Bac+5) titulaire d'un diplôme en sciences technologiques est de 31700€ brut.
- 84% des français ne croient pas que leurs impôts vont baisser en 2026.
- Parmi des amateurs de bières, la question suivante a été posée : Quel est votre type de bière préféré ? Réponses : Blondes : 33.61%, Ambrées : 25.58%, Brunes : 15.92%, Blanches : 9.64%, Un peu toutes : 15.24%
- La prise de poids moyenne pour un individu fumeur est de
  - 2.26 kilogrammes après deux mois sans tabac,
  - 4.67 kilogrammes après un an sans tabac.

1.2 Concepts de base et notations

Population et individus : On appelle population un ensemble fini d'objets sur lesquels une étude se porte. Ces objets sont appelés individus/unités statistiques.

Base de sondage : On appelle base de sondage une liste qui répertorie tous les individus d'une population.

Enquête : Étude d'une question faite en réunissant des opinions, des témoignages et des expériences. Le résultat d'une enquête vise à donner une "réponse plausible" à une question.

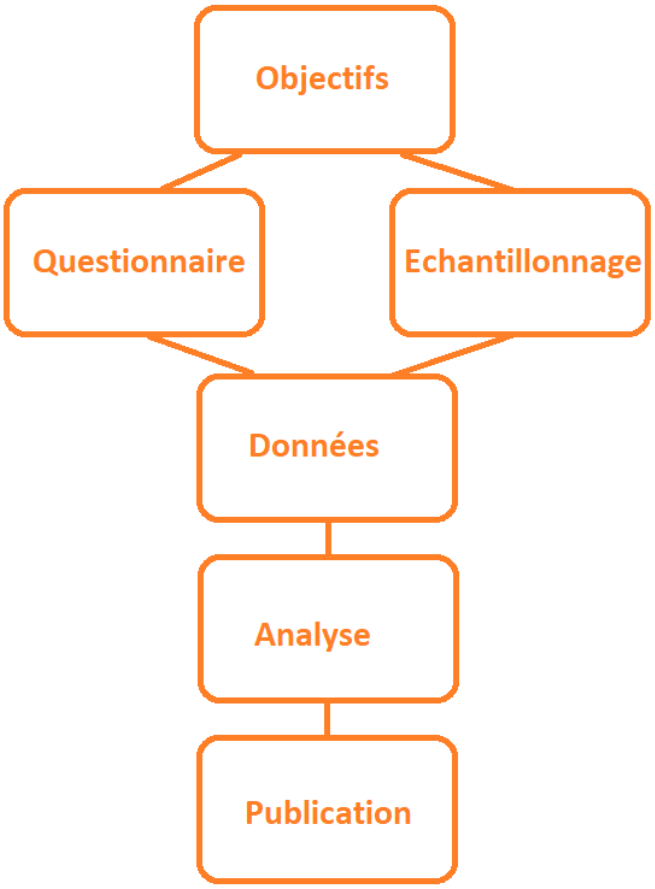
Enquête statistique : Enquête dont les principaux outils d'analyse sont de nature statistique.

Sondage : Enquête statistique dont le but est de connaître, à un moment donné, la manière dont se répartissent les opinions individuelles à propos d'une question donnée. Le résultat d'un sondage est une "photographie chiffrée" d'une actualité.

Chargé d'études : Un chargé d'études est celui qui met en œuvre une enquête statistique ou un sondage.

1.3 Étapes d'une enquête

Une enquête statistique se planifie avec les étapes suivantes :



Objectifs (et contraintes) : Dans un premier temps, il faut clarifier les objectifs principaux de l'enquête statistique. Une fois clarifiés, il ne faut pas perdre ces objectifs de vue durant tout le processus. Pour les atteindre, des contraintes pratiques doivent être prises en compte (budget modeste, temps restreint, etc.). Il s'agit de bien identifier ces contraintes et de réfléchir à comment les gérer au mieux.

Échantillonnage : Pour des raisons pratiques (budget, temps, etc.), il est souvent impossible d'accéder à tous les individus d'une population. Il faut donc sélectionner un certain nombre d'individus, constituant ainsi un échantillon d'individus, qui soit le plus représentatif possible de cette population. Dès lors, les questions suivantes se posent :

- Combien d'individus faut-il considérer dans l'échantillon ? (On parle alors de taille de l'échantillon).
- Comment constituer l'échantillon ? (On parle alors de méthodes d'échantillonnage).

Ces questions trouveront des réponses ultérieurement.

Questionnaire : Pour collecter les informations nécessaires, un questionnaire est (souvent) nécessaire. Il y a alors des règles à respecter pour optimiser la qualité du questionnaire. Il faut aussi penser à l'administration du questionnaire.

Données : Le traitement du questionnaire nous amène les données brutes. Afin qu'elles soient directement analysables avec un logiciel, il faut "nettoyer ces données", puis les mettre sous une forme arrangeante en fonction des objectifs, etc. Si l'échantillon des individus qui ont répondu n'est pas représentatif de la population, on est parfois obligé de le prendre en compte, et on peut corriger cette non-représentativité, au moins dans le calcul des mesures statistiques. On parle alors de redressement de l'échantillon.

Analyse : Une fois disponibles et prêtes, il s'agit d'analyser les données avec des outils performants (R, Python, SAS, etc.). Cette analyse doit être interprétée afin d'apporter des réponses aux objectifs de départ.

Publication : Une fois les résultats obtenus, il s'agit de les publier sous la forme de rapports, d'articles, etc., avec le support de graphiques et tableaux adaptés.

Dans le reste du document, nous allons traiter des points précédents en détail.

2 Échantillonnage

2.1 Taille de l'échantillon

Le choix d'une taille de l'échantillon est complexe. Pour résumer : "plus il y a d'individus, plus le degré de précision de l'enquête sera fort, mais les contraintes pratiques nous obligent à limiter le nombre d'individus". L'idéal est de trouver un compromis entre nombre minimum d'individus à sélectionner, le degré de précision de l'enquête et sa faisabilité pratique (budget, temps, etc.). Le dernier point pouvant être décisif.

Pour certaines méthodes d'échantillonnage, des formules mathématiques existent pour déterminer une taille acceptable de l'échantillon.

2.2 Types d'échantillonnage

On distingue deux types d'échantillonnage :

- l'échantillonnage probabiliste qui repose sur des techniques aléatoires de sélection,
- l'échantillonnage non-probabiliste qui repose sur des techniques empiriques de sélection.

Les méthodes d'échantillonnage probabiliste les plus utilisées sont les suivantes :

- l'échantillonnage aléatoire simple,
- l'échantillonnage systématique,
- l'échantillonnage aléatoire avec probabilité inégale,
- l'échantillonnage stratifié,
- l'échantillonnage en grappes (ou groupes),
- l'échantillonnage à plusieurs degrés,
- l'échantillonnage à plusieurs phases.

Les méthodes d'échantillonnage non-probabiliste les plus utilisées sont les suivantes :

- l'échantillonnage de commodité ou à l'aveuglette,
- l'échantillonnage volontaire,
- l'échantillonnage au jugé,
- l'échantillonnage par quotas.

En règle générale, les méthodes d'échantillonnage probabiliste amènent des échantillons représentatifs de la population, et avec lesquels on peut faire des estimations fiables, avec des marges d'erreur connues, voire maîtrisées. De plus, des formules nous permettent de savoir la taille d'échantillon théorique idéale ; la plus petite nous donnant des estimations suffisamment précises. En revanche, elles peuvent être compliquées à mettre en œuvre sur le terrain.

Les méthodes d'échantillonnage non-probabilistes sont souvent simples à mettre en œuvre, mais il n'y a pas moyen de savoir si l'échantillon obtenu est représentatif, et la précision des estimations qui en découlent sont incertaines.

2.3 Échantillonnages probabilistes

Les principales méthodes d'échantillonnage probabiliste sont présentées ci-dessous.

2.3.1 Échantillonnage aléatoire simple

L'échantillonnage aléatoire simple part du principe que tous les individus ont une chance égale d'être sélectionné. Pour constituer l'échantillon, on distingue l'échantillonnage aléatoire simple

- "sans remise" quand on sélectionne des individus tous différents,
- "avec remise" quand un ou plusieurs individus peuvent être sélectionnés plusieurs fois.

L'échantillonnage aléatoire simple sans remise est le plus courant ; l'échantillonnage aléatoire simple avec remise est considéré que si l'on n'a pas accès à beaucoup d'individus.

Dans la suite, on étudiera en détail la théorie sous-jacente de l'échantillonnage aléatoire simple avec remise.

Contexte "étudiants" : Pour la rentrée 2021/2022, il y a 2,81 millions d'étudiants. On souhaite savoir si les étudiants sont contents ou non de leur formation. Ne pouvant pas accéder à tous les étudiants, on décide de former un échantillon de 100000 étudiants (taille arbitraire) le plus représentatif possible.

Avec un échantillonnage aléatoire simple avec remise, on sélectionne les 100000 étudiants au hasard parmi les 2,81 millions.

2.3.2 Échantillonnage systématique

L'échantillonnage systématique consiste à numéroter et lister tous les individus de la population (dans l'ordre que l'on veut), de prendre un individu au hasard parmi les premiers numéros, disons le i-ème, de se fixer un écart de m individus, et de compléter l'échantillon en prenant les individus correspondants aux numéros : $i+m$, $i+2m$, $i+3m$, ... et ainsi de suite jusqu'à avoir un échantillon de taille jugée acceptable.

Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage systématique consiste à numéroter et lister tous les étudiants, puis de prendre, par exemple, le troisième de la liste. À partir de celui-ci, on prend se fixer un écart de 200, ce qui nous amène à prendre dans l'échantillon les étudiants numérotés 203, 403 603, ..., et ainsi de suite jusqu'au 100000-ème étudiant.

2.3.3 Échantillonnage aléatoire avec probabilités inégales

L'échantillonnage aléatoire avec probabilités inégales part du principe que certains individus ont plus ou moins de chances d'être sélectionnés.

Exemple : En reprenant le contexte “étudiants”, un exemple d’échantillonnage aléatoire avec probabilités inégales est le suivant : On peut penser à pondérer la probabilité de sélectionner un étudiant en fonction de la taille de son université. On peut alors accorder plus d’importance aux étudiants des petites universités pour augmenter leurs chances de figurer dans l’échantillon.

2.3.4 Échantillonnage stratifié

On appelle strate un groupe homogène d’individus dans une population. Si la population peut se diviser en plusieurs strates pertinentes dans le contexte, il est judicieux de les prendre en compte pour constituer un échantillon le plus représentatif possible. Dès lors, l’échantillonnage stratifié consiste à faire un échantillonnage aléatoire simple (sans remise) dans chaque strate, avec des tailles possiblement différentes, et de combiner tous les échantillons obtenus pour constituer l’échantillon final.
Exemple : En reprenant le contexte “étudiants”, un exemple d’échantillonnage stratifié est le suivant : On peut différencier les étudiants de Licence ou en Master, constituant ainsi deux strates (on peut penser que les étudiants en Master sont plus souvent contents de leur formation que les étudiants de Licence car ils se spécialisent dans un domaine choisi ; on a donc deux groupes homogènes d’étudiants quant au problème considéré). Dès lors, par un échantillonnage aléatoire simple sans remise, on peut constituer un échantillon dans la strate des étudiants en Licence, disons de taille 700000, et un autre dans la strate des étudiants en Master, disons 300000, la différence des tailles choisies s’expliquant par le fait que les étudiants en Licence étant plus nombreux. L’échantillon final est constitué de ces deux échantillons.

2.3.5 Échantillonnage en grappes

L’échantillonnage en grappes est une version “rustique et pratique” de l’échantillonnage stratifié. Si la population se divise en plusieurs groupes d’individus, souvent au sens géographique du terme, l’échantillonnage en grappes consiste à choisir au hasard plusieurs de ces groupes : une fois un groupe sélectionné, on considère tous ses individus dans l’échantillon, et tous les individus ainsi sélectionnés constituent l’échantillon final. En règle générale, il vaut mieux choisir un grand nombre de petites groupes, plutôt qu’un petit nombre de gros groupes.
Exemple : En reprenant le contexte “étudiants”, un exemple d’échantillonnage en grappes est le suivant : On sélectionne au hasard quelques universités (Université de Caen-Normandie, Université de Toulouse, Université de Lille, etc.) et on considère tous les étudiants de celles-ci. Le nombre d’universités sélectionnées est de telle sorte à ce que la taille de l’échantillon ne soit pas trop éloignée de 100000, taille fixée dans les exemples précédents.

Grappes de raisin : En termes de métaphore, la population est une vigne et les individus sont les raisins de celle-ci. On prend alors quelques grappes au hasard, et on considère tous les raisins associés.

2.3.6 Échantillonnage à deux degrés

En quelque sorte, l'échantillonnage à deux degrés est une version améliorée de l'échantillonnage en grappes, souvent utilisé quand les groupes sont de grandes tailles. Il se déroule en deux temps :

- D'abord, on choisit au hasard des groupes (comme avec un échantillonnage en grappes),
- Puis, dans chacun des groupes choisis, on sélectionne des individus avec une méthode probabiliste (échantillonnage aléatoire simple, stratifié, en grappes, etc.).

On est alors plus souple sur le nombre de groupes à choisir en premier lieu.

On peut aller à plus de deux degrés en continuant le processus de sélection des individus.

Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage à deux degrés est le suivant : On sélectionne au hasard quelques universités (Université de Caen-Normandie, Université de Toulouse, Université de Lille, etc.) et on effectue un échantillonnage aléatoire simple des étudiants dans chacune de ces universités. L'échantillon final se compose de tous les individus finalement sélectionnés. On s'arrange pour que la taille de l'échantillon ne soit pas trop éloigné de 100000, taillé fixée dans les exemples précédents.

2.3.7 Échantillonnage à deux phases

En quelque sorte, l'échantillonnage à deux phases permet de révéler des strates inconnues au premier abord, et de les utiliser pour obtenir un échantillon représentatif. Elle se déroule en deux temps :

- D'abord, on choisit au hasard des individus (comme avec un échantillonnage aléatoire simple) et leur poser une (ou plusieurs) question discriminante,
- Puis, on sélectionne les individus en fonction de leurs réponses (on choisit uniquement les individus qui ont formulé une certaine réponse, ou on crée des strates dans l'échantillon déjà formé, et on poursuit avec un sondage stratifié ou autre, etc.).

On peut aller à plus que deux phases en continuant le processus de sélection des individus en fonction de leurs réponses.

Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage à deux phases est le suivant : À l'aide d'un échantillonnage aléatoire simple, on forme un premier échantillon d'étudiants, et on leur pose une unique question : "Avez-vous été assidus pendant les cours ?" Une fois les réponses reçues, on ne garde que les étudiants qui ont répondu "oui". On peut alors concentrer l'enquête de satisfaction de la formation que sur eux (on procède à un nouvel échantillonnage aléatoire simple, si l'on veut). Là encore, difficile de respecter avec précision la taille de 100000, mais ce nombre étant subjectif à la base, ce n'est pas crucial.

2.4 Échantillonnages non-probabilistes

Les principales méthodes d'échantillonnage non-probabiliste sont présentées ci-dessous. On rappelle que celles-ci sont souvent simples et pratiques. Toutefois, il n'y a pas moyen de savoir si l'échantillon obtenu est représentatif, et la précision des estimations qui en découlent sont incertaines.

2.4.1 Échantillonnage de commodité ou à l'aveuglette

L'échantillonnage de commodité ou à l'aveuglette consiste à sélectionner les individus dont on a immédiatement accès, sans discrimination particulière.
Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage de commodité ou à l'aveuglette est le suivant : On se positionne dans les couloirs principaux de certaines universités, et on demande aux étudiants qui passent s'ils sont contents ou non de leur formation. On peut alors constituer un échantillon de taille 100000 sans trop de problème, mais cet échantillon ne sera certainement pas représentatif de l'ensemble des étudiants de France.

2.4.2 Échantillonnage volontaire

L'échantillonnage volontaire considère les individus qui sont volontaires pour participer à l'enquête (sans demander leur accord). Ainsi, les individus se sélectionnent eux-mêmes. Il peut y avoir une récompense en jeu. Si les individus sont trop nombreux, on peut faire une sélection de ceux-ci par une méthode d'échantillonnage aux choix (comme pour l'échantillonnage à deux degrés). La non-représentativité de l'échantillon est donc forte.
Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage volontaire est le suivant : Un café est offert aux 100000 premiers étudiants de France qui répondront à la question : "Êtes-vous contents de votre formation ?".

2.4.3 Échantillonnage au jugé

L'échantillonnage au jugé consiste à sélectionner uniquement les individus qui semblent être dans le cœur de cible de l'enquête. Il repose sur des critères subjectifs et des idées préconçues.
Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage au jugé est le suivant : On se positionne dans les couloirs principaux de certaines universités, et on demande aux étudiants qui passent et qui paraissent sérieux s'il sont contents ou non de leur formation. C'est donc très subjectif. Mais ainsi, on peut alors constituer un échantillon de taille 100000, mais cet échantillon ne sera certainement pas représentatif de l'ensemble des étudiants de France.

2.4.4 Échantillonnage par quotas

On appelle quotas une quantité limitée ou réglementaire (pourcentage, contingent, nombre déterminé, etc.). L'échantillonnage par quotas repose sur un résultat statistique déjà connu sur la répartition de la population (entière), ou sur un quota d'individus que l'on impose. En général, ce résultat nous informe des pourcentages des individus appartenant à des groupes formant cette population. Partant d'une taille d'échantillon choisie, l'idée est de former un échantillon d'individus dont l'appartenance aux groupes respecte ces pourcentages. Il n'y a pas de règle sur la méthode pour choisir les individus, du moment que les quotas sont respectés.

Exemple : En reprenant le contexte "étudiants", un exemple d'échantillonnage par quotas est le suivant : On sait que, en règle général, il y a approximativement 70% étudiants et 30% d'étudiants en Master. Dès lors, si on se fixe une taille d'échantillon de 100000, la méthode par quotas suggère de sélectionner $100000 \times 70/100 = 70000$ étudiants en Licence, et 30000 étudiants en Master. La sélection de ces étudiants peut se faire comme on veut.

2.4.5 Autres

On rencontre également les échantillonnages non-probabilistes suivants :

- Échantillonnage par itinéraire : Il s'agit de convenir d'un itinéraire précis à l'avance, et d'interroger les individus que l'on croise sur le chemin. Les individus interrogés forment ainsi l'échantillon. Cela évite d'interroger trop de gens au même endroit, réduisant ainsi un peu le biais. C'est une variante "nomade" de l'échantillonnage de commodité.
- Échantillonnage boule de neige : Lorsque l'on s'intéresse à une population très spécifique, on peut utiliser l'échantillonnage boule de neige qui consiste à interroger des individus par la méthode au jugé, puis d'interroger leur connaissances (amis, collègues, etc.). On rencontre aussi cet échantillonnage avec des systèmes de parrainages.

2.5 Représentativité d'un échantillon

2.5.1 Quelques exemple de non-représentativité d'un échantillon

Quelques exemples illustrant la possible non-représentativité d'un échantillon sont présentés ci-dessous.

Exemple 1 : Pour connaître l'opinion de la population de Caen sur la construction d'une nouvelle ligne de tramway, on envoie 8 enquêteurs interroger les gens à la sortie de 8 magasins fréquentés

de la ville. Ils s'arrêtent lorsqu'ils ont collectés 150 réponses chacun.

Analyse : L'échantillon d'individus obtenu n'est certainement pas représentatif car les clients des magasins ne sont pas typiques de l'ensemble de la population. En particulier, il peut y avoir une sur représentation d'hommes ou de femmes, selon le type de magasin considéré.

Exemple 2 : Dans le cadre d'une enquête de satisfaction, on choisit au hasard 2000 numéros de téléphone de particuliers dans l'annuaire national et on les appelle entre 9h00 et 17h00. On obtient 792 réponses.

Analyse : L'échantillon d'individus obtenu n'est certainement pas représentatif car il exclut pratiquement tous les individus actifs ; il y aura une sur représentativité des retraités, mères au foyer et chômeurs, entre autres. Cette représentativité peut être améliorée en appelant entre 18h00 et 21h00, quitte à ré-appeler en cas de non décrochage.

Exemple 3 : Pour se faire une idée du niveau des 4-èmes d'un collège, un enquêteur considère les élèves du premier rang de toutes les classes et les questionne. Il obtient ainsi un échantillon de 16 élèves.

Analyse : L'échantillon d'individus obtenu n'est certainement pas représentatif car les élèves des premiers rangs ne sont certainement pas représentatifs de leurs classes respectives.

Ainsi, dans le contexte, on peut deviner qu'un échantillon est représentatif ou pas. Il existe aussi des techniques statistiques, comme les tests statistiques. Dans ce contexte, les principaux tests statistiques sont décrits ci-dessous.

2.5.2 Tests statistique : vision alternative

Tests de conformité : Lorsque l'on fait un test de conformité (test d'une proportion inconnue, test d'une moyenne inconnue, etc. voir le cours de `Statistique fondamentale 1`), on s'interroge sur le fait qu'un (ou plusieurs) paramètre inconnu associé à un caractère soit égale à une (ou plusieurs) valeur de référence. Pour ce faire, on sélectionne un échantillon d'individus et on relève les valeurs du caractère considéré sur les individus, constituant ainsi les données, puis on utilise ses données pour mettre en œuvre le test.

Si la conclusion du test est que l'on peut affirmer, avec un faible risque de se tromper, qu'il y a un problème de conformité, alors que cela est en fait peu probable dans le contexte, on peut conclure que l'échantillon d'individus considéré est peu représentatif de la population.

Ainsi, dans un certain contexte, un test de conformité peut être interprêté comme un test de représentativité d'un échantillon.

Exemple 1 : Sur l'emballage de boîtes de conserve de cassoulet industriel, on peut lire que la quantité nominale contenue dans la boîte est de 2 kilogrammes. Sur un échantillon de 16 boîtes, on obtient une quantité nominale moyenne de 2.03 kilogrammes et un écart-type corrigé de 0.045 kilogrammes. Sachant que la conformité est logique, peut-on dire, avec un faible risque

de se tromper, que l'échantillon choisi est peu représentatif de la population ? *On admettra la normalité des données.*

Pour répondre à cette question, on peut réaliser un test de conformité (comparaison d'une moyenne inconnue à une valeur de référence). Soit μ la moyenne inconnue du caractère quantitatif X = "Quantité nominale contenue dans une boîte de conserve de cassoulet industriel". La moyenne de référence est $\mu_0 = 2$ (l'unité étant le kilogramme). On souhaite montrer, avec un faible risque de se tromper, que $\mu \neq \mu_0$. On considère alors les hypothèses suivantes :

$$H_0 : \mu = \mu_0 \quad \text{contre} \quad H_1 : \mu \neq \mu_0$$

Comme la normalité des données est supposée et que H_1 est bilatérale, on applique un T-Test bilatéral. On rappelle que la p-valeur associée est définie par :

$$\text{p-valeur} = \mathbb{P}(|T| \geq |t|),$$

où $T \sim \mathcal{T}(\nu)$ avec $\nu = n - 1$, n étant le nombre d'individus dans l'échantillon (ou le nombre de données), $\overline{x}$ est la moyenne des données, s est l'écart-type corrigé des données et

$$t = \sqrt{n}\left(\frac{\overline{x} - \mu_0}{s}\right).$$

Si l'on avait accès aux données brutes, on aurait pu utiliser la commande `t.test(x, mu = mu0)$p.value`, avec `x` le vecteur contenant ces données. Ici, n'ayant pas les données brute, on est contraint de coder la p-valeur à la main en faisant :

```
n = 16
mu0 = 2
xbar = 2.03
s = 0.045
 t = sqrt(n) * ((xbar - mu0) / s)
pvaleur = 2 * (1 - pt(abs(t), n - 1))
pvaleur
```

Cela renvoie :

```
[1] 0.01759515
```

On a p-valeur $= 0.01759515$. Comme p-valeur $\in]0.01, 0.05]$, on rejette H_0 et ce rejet est significatif $\star$. Ainsi, avec un faible risque de se tromper, on peut dire qu'il y a non-conformité, contrairement à la logique. Par conséquent, l'échantillon est peu représentatif de la population.

Exemple 2 : Dans un établissement, sur un échantillon de 180 étudiants, 41 ont un quotient intellectuel (QI) supérieur à 120. Or il est admis que le taux national de personnes ayant un tel QI est de 30%. Peut-on affirmer, avec un faible risque de se tromper, que l'échantillon choisi est représentatif de la population ?

Pour répondre à cette question, on peut réaliser un test de conformité (comparaison d'une proportion inconnue à une valeur de référence). Dès lors, soit p la proportion inconnue des étudiants de cet établissement ayant un QI supérieur à 120. La proportion de référence est $p_0 = 0.30$. On considère les hypothèses suivantes :

$$H_0 : p = p_0 \quad \text{contre} \quad H_1 : p \neq p_0$$

Comme H_1 est bilatérale, on utilise le test binomial bilatéral. La p-valeur associée est calculée avec la commande `binom.test(m, n, p0)$p.value`. Ainsi, on fait :

```
m = 41
n = 180
p0 = 0.30
binom.test(m, n, p0)$p.value
```

Cela renvoie :

```
[1]  0.03445969
```

On a p-valeur $= 0.03445969$. Comme p-valeur $\in]0.01, 0.05]$, on rejette H_0 et ce rejet est significatif $\star$. Ainsi, avec un faible risque de se tromper, on peut dire que le taux national n'est pas respecté. Par conséquent, l'échantillon est peu représentatif de la population.

Tests d'adéquation à une loi de probabilité : Le même raisonnement s'applique pour les tests d'adéquation à une loi de probabilité. Si l'on est sûr de connaître la loi de probabilité sous-jacente à un caractère et que les données associées à un échantillon d'individus contredisent cette loi, on peut conclure que cet échantillon est peu représentatif de la population.

Exemple : La marque Dragicolor produit des bonbons de 6 couleurs différentes. Il est connu que la proportion de chaque couleur est très précisément de 30% pour le brun, noté B, 20% pour le jaune, noté J, 20% pour le rouge, noté R, 10% pour l'orange, noté O, 10% pour le vert, noté V et 10% pour le doré, noté D. Sur un échantillon de 355 bonbons, on obtient les résultats suivants :

couleur	B	J	R	O	V	D
nombre de bonbons	82	77	76	45	33	42

Peut-on dire, avec un faible risque de se tromper, que l'échantillon n'est pas représentatif de la population ?

Pour répondre à cette question, on peut réaliser un test du Chi-deux d'adéquation en prenant pour loi de référence celle donnée dans l'énoncé. On considère le caractère qualitatif $X =$ "Couleur d'un bonbon". Il a 6 modalités. On peut modéliser X par une *var* discrète X. Soit $\mathcal{L}$ la loi théorique. On considère alors les hypothèses suivantes :

$$H_0 : "X \sim \mathcal{L}" \quad \text{contre} \quad H_1 : "X \text{ ne suit pas la loi } \mathcal{L}"$$

On utilise le test du Chi-deux avec les probabilités théoriques : $p_1 = 0.3$, $p_2 = 0.2$, $p_3 = 0.2$, $p_4 = 0.1$, $p_5 = 0.1$ et $p_6 = 0.1$. Il vient :

```
nb = c(82, 77, 76, 45, 33, 42)
proba = c(0.3, 0.2, 0.2, 0.1, 0.1, 0.1)
chisq.test(nb, p = proba)$p.value
```

Cela renvoie :

```
[1] 0.06457068
```

Notons qu'aucun "Warning message" n'apparaît ; les conditions d'application du test sont vérifiées.

Ainsi, on a p-valeur = 0.06457068. Comme p-valeur > 0.05, on ne rejette pas H_0. Dès lors, on ne peut pas affirmer qu'il y a un problème de représentativité de l'échantillon.

2.6 Exercices corrigés

Exercice 1 : Un sondage est réalisé pour savoir le coût d'achat des livres des 4000 élèves d'une formation générale qui offre 50 programmes d'études. Identifier les méthodes d'échantillonnage correspondantes aux cas suivants :

- On attribue aux 4000 élèves les numéros $1, 2, 3, \ldots, 4000$, respectivement, et on note chaque numéro sur un bout de papier. On en tire 400 au hasard et sans remise. Les élèves associés aux numéros sélectionnés forment notre échantillon.
- On considère la liste de numéros associés aux élèves précédemment constituée. On choisit au hasard le numéro 7. Puis on poursuit notre sélection en appliquant un pas de 10 dans la numérotation : on sélectionne donc le numéro 17, puis le 27 et ainsi de suite jusqu'à ne pas dépasser la limite 4000. Les élèves associés aux numéros sélectionnés forme notre échantillon.
- On choisit 15 programmes d'études au hasard, et on prend tous les élèves y participants pour former notre échantillon.
- Pour chacun des 50 programmes, on sélectionne au hasard 20% des élèves, formant ainsi notre échantillon.
- On choisit 15 programmes d'études au hasard, et pour chacun de ces programmes, on sélectionne au hasard 15% des élèves, formant ainsi notre échantillon.
- On attribue aux 4000 élèves les numéros $1, 2, 3, \ldots, 4000$, respectivement, et on note chaque numéro sur un bout de papier. On en tire 1000 au hasard et sans remise. On interroge alors ces élèves pour savoir s'ils ont acheté des livres ou non (ils peuvent très bien les consulter à

la bibliothèque sans les acheter, ou reprendre ceux généreusement donnés par des étudiants des promotions précédentes). Parmi ceux qui répondent en avoir acheté, on en sélectionne 80% au hasard.

Solution :

- Il s'agit d'un échantillonnage aléatoire simple.
- Il s'agit d'un échantillonnage systématique.
- Il s'agit d'un échantillonnage en grappes.
- Il s'agit d'un échantillonnage stratifié.
- Il s'agit d'un échantillonnage à deux degrés.
- Il s'agit d'un échantillonnage à deux phases.

Exercice 2 : On souhaite former un échantillon de 1000 individus représentatif de la population française en termes d'âge, de genre, de catégories sociaux-professionnelles, etc. On doit respecter les caractéristiques suivantes : 51% de femmes et 49% d'hommes, 20% de population rurale et 80 % de population urbaine, 20.5% d'ouvriers, 27.5% d'employés, 25.5% de professions intermédiaires, 18.5% de cadres, 6.5% d'artisans et 1.5% d'agriculteurs. L'échantillon est ainsi fait, sans processus aléatoire particulier. Quelle méthode d'échantillonnage avons-nous utilisée ?

Solution : Il s'agit d'un échantillonage par quotas.

Exercice 3 : Des enquêteurs font un sondage dans un centre commercial afin d'avoir l'opinion des habitants sur les nouvelles règles de circulation dans une ville. Ils forment un échantillon de 100 clients pour faire l'enquête (la méthode n'est pas précisée, mais on l'a supposé adéquate). Est-ce que l'échantillon est représentatif a priori ?

Solution : L'échantillon n'est pas représentatif car il exclut plusieurs catégories d'individus, notamment ceux qui n'aiment pas se rendre dans les centres commerciaux en général, ceux qui ne peuvent pas se rendre au centre commercial considéré, et ceux qui préfèrent se rendre dans un autre centre commercial.

Exercice 4 : En France, la proportion des individus aux cheveux châtains dans la population est de 50% (environ). On a observé un échantillon de 152 individus parmi lesquels 90 ont les cheveux châtains. Est-ce que cet échantillon est représentatif de la population ?

Solution : Pour répondre à cette question, on peut réaliser un test de conformité (comparaison d'une proportion inconnue à une valeur de référence). Dès lors, soit p la proportion inconnue des individus ayant des cheveux châtains. La proportion de référence est $p_0 = 0.50$. On considère les hypothèses suivantes :

$$H_0 : p = p_0 \quad \text{contre} \quad H_1 : p \neq p_0$$

Comme H_1 est bilatérale, on utilise le test binomial bilatéral. La p-valeur associée est calculée avec la commande `binom.test(m, n, p0)$p.value`. Ainsi, on fait :

```
m = 90
n = 152
p0 = 0.5
binom.test(m, n, p0)$p.value
```

Cela renvoie :

```
[1]  0.02819284
```

On a p-valeur = 0.02819284. Comme p-valeur $\in]0.01, 0.05]$, on rejette H_0 et ce rejet est significatif $\star$. Ainsi, avec un faible risque de se tromper, on peut dire que l'échantillon est peu représentatif de la population.

Exercice 5 : Les résultats d'une élection nationale opposant 4 partis politiques ont donné les pourcentages suivants : $pourc1 = 15\%$, $pourc2 = 5\%$, $pourc3 = 60\%$, $pourc4 = 20\%$, respectivement. Dans un village de 200 habitants, le nombre de voix obtenues par chaque parti a été de : $o1 = 9$, $o2 = 5$, $o3 = 141$, $o4 = 45$, respectivement. Peut-on affirmer, avec un faible risque de se tromper, que l'échantillon (et le village en particulier) n'est pas représentatif de la tendance nationale ?

Solution : Pour répondre à cette question, on peut réaliser un test du Chi-deux d'adéquation en prenant pour loi de référence celle donnée par les pourcentages de la tendance nationale. On considère le caractère qualitatif X = "Intention de votes". Il a 4 modalités. On peut modéliser X par une *var* discrète X. Soit $\mathcal{L}$ la loi théorique. On considère alors les hypothèses suivantes :

$$H_0 : "X \sim \mathcal{L}" \quad \text{contre} \quad H_1 : "X \text{ ne suit pas la loi } \mathcal{L}"$$

On utilise le test du Chi-deux avec les probabilités théoriques : $p_1 = 0.15$, $p_2 = 0.05$, $p_3 = 0.6$ et $p_4 = 0.2$. Il vient :

```
nb = c(9, 5, 141, 45)
proba = c(0.15, 0.05, 0.6, 0.2)
chisq.test(nb, p = proba)$p.value
```

Cela renvoie :

```
[1] 8.287855e-05
```

Notons qu'aucun "Warning message" n'apparaît ; les conditions d'application du test sont vérifiées.

Ainsi, on a p-valeur $= 8.287855 \times 10^{-5}$. Comme p-valeur < 0.001, le rejet de H_0 est hautement significatif ($\star\star\star$). Ainsi, on peut affirmer, avec un faible risque de se tromper, que l'échantillon n'est pas représentatif de la tendance nationale.

3 Plan de sondage aléatoire simple sans remise (PESR)

L'objectif de ce chapitre est de présenter en détail les outils mathématiques utilisés dans le cadre d'un plan de sondage classique : le plan de sondage de type PESR. En quelques sortes, c'est une introduction à la "théorie des sondages", qui est une branche spécifique de la Statistique.

3.1 Concepts de base (rappel) et notations

Rappel : population et individus : On appelle population un ensemble fini d'objets sur lesquels une étude se porte. Ces objets sont appelés individus/unités statistiques. Une population est notée

$$U = \{u_1, \ldots, u_N\},$$

où N est le nombre d'individus dans la population et, pour tout $i \in \{1, \ldots, N\}$, u_i est le i-ème individu.

Rappel : Base de sondage : On appelle base de sondage une liste qui répertorie tous les individus d'une population.

Rappel : caractère : Un caractère est une qualité que l'on étudie chez les individus d'une population.

Un caractère est noté Y. Pour tout $i \in \{1, \ldots, N\}$, on note y_i la valeur de Y pour l'individu u_i.

Moyenne-population :

On appelle moyenne-population le réel :

$$\overline{y}_U = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

Le paramètre $\overline{y}_U$ est une valeur centrale de Y.

Écart-type corrigé-population :

On appelle écart-type corrigé-population le réel :

$$s_U = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (y_i - \overline{y}_U)^2}.$$

Le paramètre s_U mesure la dispersion de Y autour de $\overline{y}_U$.

Rappel : échantillon : On appelle échantillon un ensemble d'individus issus d'une population.

Un échantillon est noté ω. Le nombre d'individus dans un échantillon est noté n.

Deux questions centrales : Pour constituer un échantillon représentatif de la population,

- comment faut-il procéder ?
- combien d'individus faut-il choisir ?

Plan de sondage : On appelle plan de sondage une procédure permettant de sélectionner un échantillon dans une population. Un plan de sondage est dit :

- aléatoire si chaque individu de la population a une probabilité connue de se retrouver dans l'échantillon,
- simple si chaque individu a la même probabilité qu'un autre d'être sélectionné ; les probabilités sont égales (PE),
- sans remise (SR) si un même individu ne peut apparaître qu'une seule fois dans l'échantillon.

3.2 Contexte

Loi de probabilité :

On prélève un échantillon de n individus suivant un plan de sondage aléatoire simple sans remise (PESR pour Probabilités Egales Sans Remise) dans une population U. Soit W la *var* égale à l'échantillon obtenu. Alors la loi de W est donnée par

$$\mathbb{P}(W = \omega) = \frac{1}{\binom{N}{n}}, \qquad \omega \in W(\Omega),$$

où $\mathbb{P}$ désigne la probabilité uniforme et $W(\Omega)$ désigne l'ensemble de tous les échantillons de n individus possibles parmi les N avec un tel plan de sondage.

Explication : Pour fixer les idées, on considère la situation simplifiée suivante : on prélève au hasard et simultanément n individus de la population pour former un échantillon. L'univers associé à cette expérience aléatoire est $\Omega = \{\text{combinaisons de } n \text{ individus parmi } N\}$. Comme Ω est fini et qu'il y a équiprobabilité, l'utilisation de la probabilité uniforme $\mathbb{P}$ est justifiée. Il vient

$$\mathbb{P}(W = \omega) = \frac{\text{Card}(\{W = \omega\})}{\text{Card}(\Omega)}, \qquad \omega \in W(\Omega).$$

Or on a $\text{Card}(\Omega) = \binom{N}{n}$ et $\text{Card}(\{W = \omega\}) = 1$, d'où le résultat.

Situations de référence : Les différents types de prélèvements décrits ci-dessous rentrent dans le cadre d'un PESR :

- on prélève au hasard et simultanément n individus de la population pour former un échantillon,
- on prélève au hasard et un à un n individus de la population pour former un échantillon, l'ordre n'étant pas pris en compte.

Quelques commandes R : Pour illustrer un plan de sondage aléatoire de type PESR avec le logiciel R, on fait :

```
install.packages("animation")
library(animation)
sample.simple(nrow = 10, ncol = 10, size = 15, p.col = c("blue", "red"),
p.cex = c(1, 3))
```

Par exemple, pour faire un tirage sans remise de $n = 20$ individus dans une population de $N = 200$ individus, on peut utiliser

◦ la commande `sample` :

```
sample(1:200, 20, replace = F)
```

◦ la commande `srswor` de la librairie `sampling` :

```
install.packages("sampling")
library(sampling)
t = srswor(20, 200)
x = 1:200
x[t != 0]
```

L'abréviation `srswor` signifie `Simple Random Sampling WithOut Replacement`.

Précisons que `t = srswor(20, 200)` renvoie un vecteur de taille 200 constitué de 20 chiffres 1 et de 180 chiffres 0. Les 1 sont positionnés aux indices des individus prélevés et les 0 aux autres.

Un autre exemple : on considère la population U constituée de $N = 9$ garçons et on prélève un échantillon de $n = 3$ individus suivant un plan de sondage aléatoire de type PESR :

```
U = c("Bob", "Nico", "Ali", "Fabien", "Malik", "John", "Jean", "Chris", "Karl")
library(sampling)
t = srswor(3, 9)
w = U[t != 0]
w
```

◦ la commande `slice_sample` de la librairie `dplyr`. Elle sera traitée en détail plus tard dans le document.

Dans la suite :

◦ pour les résultats, on considère un plan de sondage aléatoire de type PESR et la *var* W égale à l'échantillon obtenu,

◦ pour les preuves, pour raison de simplicité, on se place dans la situation de référence I,

◦ pour les commandes R, on se focalisera dorénavant sur la librairie `sampling`.

Taux de sondage :

On appelle taux de sondage le réel :

$$f = \frac{n}{N}.$$

Probabilités d'appartenance :

- pour tout $i \in \{1, \dots, N\}$, la probabilité que l'individu u_i appartienne à W est

$$\mathbb{P}(u_i \in W) = \frac{n}{N} \qquad (= f).$$

- pour tout $(i, j) \in \{1, \dots, N\}^2$ avec $i \neq j$, la probabilité que les individus u_i et u_j appartiennent à W est

$$\mathbb{P}((u_i, u_j) \in W) = \frac{n(n-1)}{N(N-1)}.$$

Preuve :

- Par la définition de la probabilité uniforme, on a

$$\mathbb{P}(u_i \in W) = \frac{\text{Card}(\{u_i \in W\})}{\text{Card}(\Omega)}.$$

On a $\text{Card}(\Omega) = \binom{N}{n}$. Il reste à calculer $\text{Card}(\{u_i \in W\})$. Le nombre de possibilités pour que u_i soit dans l'échantillon est égal au nombre de possibilités de prélever $n-1$ individus parmi les $N-1$ autres que u_i. D'où $\text{Card}(\{u_i \in W\}) = \binom{N-1}{n-1}$. On en déduit que

$$\mathbb{P}(u_i \in W) = \frac{\binom{N-1}{n-1}}{\binom{N}{n}} = \frac{\frac{(N-1)!}{(n-1)!((N-1)-(n-1))!}}{\frac{N!}{n!(N-n)!}} = \frac{n!}{(n-1)!}\frac{(N-1)!}{N!} = \frac{n}{N}.$$

- Avec un raisonnement similaire, on a

$$\mathbb{P}((u_i, u_j) \in W) = \frac{\text{Card}(\{(u_i, u_j) \in W\})}{\text{Card}(\Omega)}.$$

On a $\text{Card}(\Omega) = \binom{N}{n}$. Il reste à calculer $\text{Card}(\{(u_i, u_j) \in W\})$.
Le nombre de possibilités pour que u_i et u_j soient dans l'échantillon est égal au nombre de possibilités pour prélever simultanément $n-2$ individus parmi les $N-2$ autres que u_i et u_j. D'où $\text{Card}(\{(u_i, u_j) \in W\}) = \binom{N-2}{n-2}$. On en déduit que

$$\mathbb{P}((u_i, u_j) \in W) = \frac{\binom{N-2}{n-2}}{\binom{N}{n}} = \frac{\frac{(N-2)!}{(n-2)!((N-2)-(n-2))!}}{\frac{N!}{n!(N-n)!}} = \frac{n!}{(n-2)!}\frac{(N-2)!}{N!} = \frac{n(n-1)}{N(N-1)}.$$

□

3.3 Estimateurs

Estimation aléatoire de $\overline{y}_U$:

Un estimateur aléatoire de $\overline{y}_U$ est

$$\overline{y}_W = \frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i \in W\}},$$

où $\mathbb{1}$ désigne la fonction indicatrice définie par : $\mathbb{1}_A = \begin{cases} 1 & \text{si l'événement } A \text{ est réalisé,} \\ 0 & \text{sinon.} \end{cases}$

Remarques : On peut également écrire cet estimateur

- sous la forme :

$$\overline{y}_W = \frac{1}{n}\sum_{i \in S} y_i,$$

où $S = \{(i_1, \ldots, i_n) \in \{1, \ldots, N\}^n,\ i_1 \neq \ldots \neq i_n;\ \ u_{i_1} \in W, \ldots, u_{i_n} \in W\}$,

- sous la forme :

$$\overline{y}_W = \frac{1}{n}\sum_{i=1}^{N} y_i \sum_{m=1}^{n} \mathbb{1}_{\{W_m = u_i\}},$$

où W_m est la *var* égale au m-ème individu de l'échantillon.

En effet, comme $W = (W_1, \ldots, W_n)$ et tous les individus sont différents, on a $\sum_{m=1}^{n} \mathbb{1}_{\{W_m = u_i\}} = \mathbb{1}_{\{u_i \in W\}}$.

On peut montrer que, pour tout $i \in \{1, \ldots, N\}$ et $m \in \{1, \ldots, n\}$, on a $\mathbb{P}(u_i \in W_m) = 1/N$.

Espérance de $\overline{y}_W$:

L'estimateur $\overline{y}_W$ est sans biais pour $\overline{y}_U$:

$$\mathbb{E}(\overline{y}_W) = \overline{y}_U.$$

Preuve : En utilisant la linéarité de l'espérance, $\mathbb{E}(\mathbb{1}_A) = \mathbb{P}(A)$ et $\mathbb{P}(u_i \in W) = n/N$, il vient

$$\begin{aligned} \mathbb{E}(\overline{y}_W) &= \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i \in W\}}\right) = \frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}\right) \\ &= \frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{P}(u_i \in W) = \frac{1}{n}\sum_{i=1}^{N} y_i \frac{n}{N} = \frac{1}{N}\sum_{i=1}^{N} y_i = \overline{y}_U. \end{aligned}$$

Variance de $\overline{y}_W$:

La variance de $\overline{y}_W$ est

$$\mathbb{V}(\overline{y}_W) = (1-f)\frac{s_U^2}{n}.$$

Preuve : Par la formule de la variance d'une somme de *var*, on obtient

$$\begin{aligned}
\mathbb{V}(\overline{y}_W) &= \mathbb{V}\left(\frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i \in W\}}\right) = \frac{1}{n^2}\mathbb{V}\left(\sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i \in W\}}\right) \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{N} \mathbb{V}\left(y_i \mathbb{1}_{\{u_i \in W\}}\right) + 2\sum_{i=2}^{N}\sum_{j=1}^{i-1} \mathbb{C}\left(y_i \mathbb{1}_{\{u_i \in W\}}, y_j \mathbb{1}_{\{u_j \in W\}}\right)\right) \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{N} y_i^2 \mathbb{V}\left(\mathbb{1}_{\{u_i \in W\}}\right) + 2\sum_{i=2}^{N}\sum_{j=1}^{i-1} y_i y_j \mathbb{C}\left(\mathbb{1}_{\{u_i \in W\}}, \mathbb{1}_{\{u_j \in W\}}\right)\right).
\end{aligned}$$

Or, en utilisant $\mathbb{P}(u_i \in W) = n/N$, on a

$$\begin{aligned}
\mathbb{V}\left(\mathbb{1}_{\{u_i \in W\}}\right) &= \mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}^2\right) - \left(\mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}\right)\right)^2 = \mathbb{P}(u_i \in W) - (\mathbb{P}(u_i \in W))^2 \\
&= \frac{n}{N} - \left(\frac{n}{N}\right)^2 = \frac{n}{N}\left(1 - \frac{n}{N}\right).
\end{aligned}$$

De plus, comme $\mathbb{P}(\{u_i \in W\} \cap \{u_j \in W\}) = \mathbb{P}((u_i, u_j) \in W) = n(n-1)/(N(N-1))$, il vient

$$\begin{aligned}
\mathbb{C}\left(\mathbb{1}_{\{u_i \in W\}}, \mathbb{1}_{\{u_j \in W\}}\right) &= \mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}\mathbb{1}_{\{u_j \in W\}}\right) - \mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}\right)\mathbb{E}\left(\mathbb{1}_{\{u_j \in W\}}\right) \\
&= \mathbb{P}(\{u_i \in W\} \cap \{u_j \in W\}) - \mathbb{P}(u_i \in W)\mathbb{P}(u_j \in W) \\
&= \frac{n(n-1)}{N(N-1)} - \left(\frac{n}{N}\right)^2 = \frac{n}{N}\left(\frac{n-1}{N-1} - \frac{n}{N}\right).
\end{aligned}$$

En combinant ces égalités, on obtient

$$\begin{aligned}
\mathbb{V}(\overline{y}_W) &= \frac{1}{n^2}\left(\frac{n}{N}\left(1 - \frac{n}{N}\right)\sum_{i=1}^{N} y_i^2 + 2\frac{n}{N}\left(\frac{n-1}{N-1} - \frac{n}{N}\right)\sum_{i=2}^{N}\sum_{j=1}^{i-1} y_i y_j\right) \\
&= \frac{1}{nN}\left(\left(1 - \frac{n}{N}\right)\sum_{i=1}^{N} y_i^2 + \left(\frac{n-1}{N-1} - \frac{n}{N}\right)\left(2\sum_{i=2}^{N}\sum_{j=1}^{i-1} y_i y_j\right)\right).
\end{aligned}$$

En utilisant la décomposition :

$$2\sum_{i=2}^{N}\sum_{j=1}^{i-1} y_i y_j = \left(\sum_{i=1}^{N} y_i\right)^2 - \sum_{i=1}^{N} y_i^2,$$

on obtient

$$\begin{aligned}
\mathbb{V}(\overline{y}_W) &= \frac{1}{nN}\left(\left(1-\frac{n}{N}\right)\sum_{i=1}^N y_i^2 + \left(\frac{n-1}{N-1}-\frac{n}{N}\right)\left(\left(\sum_{i=1}^N y_i\right)^2 - \sum_{i=1}^N y_i^2\right)\right)\\
&= \frac{1}{nN}\left(\left(1-\frac{n}{N}-\frac{n-1}{N-1}+\frac{n}{N}\right)\sum_{i=1}^N y_i^2 + \left(\frac{n-1}{N-1}-\frac{n}{N}\right)\left(\sum_{i=1}^N y_i\right)^2\right)\\
&= \frac{1}{nN}\left(\frac{N-n}{N-1}\sum_{i=1}^N y_i^2 - \frac{N-n}{N(N-1)}\left(\sum_{i=1}^N y_i\right)^2\right)\\
&= \frac{N-n}{nN}\left(\frac{1}{N-1}\left(\sum_{i=1}^N y_i^2 - N\left(\frac{1}{N}\sum_{i=1}^N y_i\right)^2\right)\right).
\end{aligned}$$

D'autre part, on a

$$\begin{aligned}
s_U^2 &= \frac{1}{N-1}\sum_{i=1}^N (y_i-\overline{y}_U)^2 = \frac{1}{N-1}\left(\sum_{i=1}^N y_i^2 - 2\overline{y}_U\sum_{i=1}^N y_i + N\overline{y}_U^2\right)\\
&= \frac{1}{N-1}\left(\sum_{i=1}^N y_i^2 - 2N\overline{y}_U^2 + N\overline{y}_U^2\right) = \frac{1}{N-1}\left(\sum_{i=1}^N y_i^2 - N\overline{y}_U^2\right)\\
&= \frac{1}{N-1}\left(\sum_{i=1}^N y_i^2 - N\left(\frac{1}{N}\sum_{i=1}^N y_i\right)^2\right).
\end{aligned}$$

Il s'ensuit

$$\mathbb{V}(\overline{y}_W) = \frac{N-n}{nN}s_U^2 = \left(1-\frac{n}{N}\right)\frac{s_U^2}{n} = (1-f)\frac{s_U^2}{n}.$$

□

Erreur quadratique moyenne de $\overline{y}_W$:

L'erreur quadratique moyenne de $\overline{y}_W$ est le réel :

$$EQM(\overline{y}_W)[PESR] = \mathbb{E}\left((\overline{y}_W-\overline{y}_U)^2\right) = (1-f)\frac{s_U^2}{n}.$$

La quantité $EQM(\overline{y}_W)[PESR]$ est une mesure de l'erreur que commet $\overline{y}_W$ dans l'estimation de $\overline{y}_U$.

On constate que :

- plus n est grand/l'échantillon est grand, plus $\overline{y}_W$ estime bien $\overline{y}_U$,
- plus U est homogène/plus s_U^2 est petit, plus $\overline{y}_W$ estime bien $\overline{y}_U$.

Estimation aléatoire de s_U :

Un estimateur aléatoire de s_U est

$$s_W = \sqrt{\frac{1}{n-1}\sum_{i=1}^{N}(y_i - \overline{y}_W)^2 \mathbb{1}_{\{u_i \in W\}}}.$$

Propriété de s_W^2 :

L'estimateur s_W^2 est sans biais pour s_U^2 :

$$\mathbb{E}\left(s_W^2\right) = s_U^2.$$

Preuve : En remarquant que $\sum_{i=1}^{N} \mathbb{1}_{\{u_i \in W\}} = n$, il vient

$$\begin{aligned}
s_W^2 &= \frac{1}{n-1}\sum_{i=1}^{N}(y_i - \overline{y}_W)^2 \mathbb{1}_{\{u_i \in W\}} \\
&= \frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{1}_{\{u_i \in W\}} - 2\overline{y}_W \sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i \in W\}} + \overline{y}_W^2 \sum_{i=1}^{N} \mathbb{1}_{\{u_i \in W\}}\right) \\
&= \frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{1}_{\{u_i \in W\}} - 2n\overline{y}_W^2 + n\overline{y}_W^2\right) = \frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{1}_{\{u_i \in W\}} - n\overline{y}_W^2\right).
\end{aligned}$$

On a $\mathbb{P}(u_i \in W) = n/N$ et

$$\mathbb{E}\left(\overline{y}_W^2\right) = \mathbb{V}\left(\overline{y}_W\right) + \left(\mathbb{E}\left(\overline{y}_W\right)\right)^2 = (1-f)\frac{s_U^2}{n} + \overline{y}_U^2.$$

D'où

$$\begin{aligned}
\mathbb{E}\left(s_W^2\right) &= \mathbb{E}\left(\frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{1}_{\{u_i \in W\}} - n\overline{y}_W^2\right)\right) = \frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{E}\left(\mathbb{1}_{\{u_i \in W\}}\right) - n\mathbb{E}\left(\overline{y}_W^2\right)\right) \\
&= \frac{1}{n-1}\left(\sum_{i=1}^{N} y_i^2 \mathbb{P}\left(u_i \in W\right) - n\mathbb{E}\left(\overline{y}_W^2\right)\right) \\
&= \frac{1}{n-1}\left(\frac{n}{N}\sum_{i=1}^{N} y_i^2 - n\left((1-f)\frac{s_U^2}{n} + \overline{y}_U^2\right)\right) \\
&= \frac{1}{n-1}\left(\frac{n}{N}\left(\sum_{i=1}^{N} y_i^2 - N\overline{y}_U^2\right) - \left(1 - \frac{n}{N}\right)s_U^2\right) \\
&= \frac{n(N-1)}{(n-1)N}\left(\frac{1}{N-1}\left(\sum_{i=1}^{N} y_i^2 - N\overline{y}_U^2\right)\right) - \frac{1}{n-1}\left(1 - \frac{n}{N}\right)s_U^2.
\end{aligned}$$

Or

$$\begin{aligned} s_U^2 &= \frac{1}{N-1}\sum_{i=1}^{N}(y_i-\overline{y}_U)^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N}y_i^2 - 2\overline{y}_U\sum_{i=1}^{N}y_i + N\overline{y}_U^2\right) \\ &= \frac{1}{N-1}\left(\sum_{i=1}^{N}y_i^2 - 2N\overline{y}_U^2 + N\overline{y}_U^2\right) = \frac{1}{N-1}\left(\sum_{i=1}^{N}y_i^2 - N\overline{y}_U^2\right). \end{aligned}$$

Par conséquent,

$$\begin{aligned} \mathbb{E}\left(s_W^2\right) &= \frac{n(N-1)}{(n-1)N}s_U^2 - \frac{1}{n-1}\left(1-\frac{n}{N}\right)s_U^2 \\ &= \frac{n(N-1)-N+n}{(n-1)N}s_U^2 = \frac{nN-n-N+n}{(n-1)N}s_U^2 = \frac{(n-1)N}{(n-1)N}s_U^2 = s_U^2. \end{aligned}$$

□

3.4 Estimations ponctuelles

Estimation ponctuelle de $\overline{y}_U$:

Soit ω un échantillon de n individus de U. Une estimation ponctuelle de $\overline{y}_U$ est la moyenne-échantillon :

$$\overline{y}_\omega = \frac{1}{n}\sum_{i=1}^{N} y_i \mathbb{1}_{\{u_i\in\omega\}}.$$

Quelques commandes R : Un exemple de calcul de $\overline{y}_\omega$ avec R est décrit ci-dessous :

```
U = c("Bob", "Nico", "Ali", "Fabien", "Malik", "John", "Jean", "Chris", "Karl")
y = c(72, 89, 68, 74, 81, 87, 76, 61, 84)
n = 3
library(sampling)
t = srswor(n, 9)
bar_y_w = (1 / n) * sum(y * t)
bar_y_w
```

Erreur d'estimation :

Soit ω un échantillon de n individus de U. L'erreur d'estimation que commet $\overline{y}_\omega$ en estimant $\overline{y}_U$ est le réel :

$$e_\omega = |\overline{y}_\omega - \overline{y}_U|.$$

Probabilité d'erreur :

La probabilité de se tromper de plus de $(100 \times \beta)\%$, $\beta \in]0, 1[$, en estimant $\overline{y}_U$ par $\overline{y}_W$ est le réel :

$$p_\beta = \frac{1}{\binom{N}{n}} \sum_{\omega \in W(\Omega)} \mathbb{1}_{\{e_\omega \geq \beta \overline{y}_U\}}.$$

Estimation ponctuelle de s_U :

Soit ω un échantillon de n individus de U. Une estimation ponctuelle de s_U est l'écart-type corrigé-échantillon :

$$s_\omega = \sqrt{\frac{1}{n-1} \sum_{i=1}^{N} (y_i - \overline{y}_\omega)^2 \mathbb{1}_{\{u_i \in \omega\}}}.$$

Tout comme la moyenne-population, on peut aussi s'intéresser à l'erreur d'estimation et la probabilité d'erreur, lesquelles se définissent de manière similaire.

Quelques commandes R : Un exemple de calcul de s_ω avec R est décrit ci-dessous :

```
U = c("Bob", "Nico", "Ali", "Fabien", "Malik", "John", "Jean", "Chris", "Karl")
y = c(72, 89, 68, 74, 81, 87, 76, 61, 84)
n = 3
library(sampling)
t = srswor(n, 9)
bar_y_w = (1 / n) * sum(y * t)
s_w = sqrt(sum((y - bar_y_w)^2 * t) / (n - 1))
s_w
```

Estimation ponctuelle de l'écart-type de $\overline{y}_W$:

Soit ω un échantillon de n individus de U. Une estimation ponctuelle de l'écart-type de $\overline{y}_W$ est le réel :

$$s(\overline{y}_\omega) = \sqrt{(1-f)\frac{s_\omega^2}{n}}.$$

3.5 Intervalles de confiance

Intervalle de confiance pour $\overline{y}_U$:

Soit ω un échantillon de n individus de U. Un intervalle de confiance pour $\overline{y}_U$ au niveau $100(1-\alpha)\%$, $\alpha \in]0,1[$, est

$$\begin{aligned} i_{\overline{y}_U} &= [\overline{y}_\omega - z_\alpha s(\overline{y}_\omega),\ \overline{y}_\omega + z_\alpha s(\overline{y}_\omega)] \\ &= \left[\overline{y}_\omega - z_\alpha \sqrt{(1-f)\frac{s_\omega^2}{n}},\ \overline{y}_\omega + z_\alpha \sqrt{(1-f)\frac{s_\omega^2}{n}}\right], \end{aligned}$$

où z_α est le réel vérifiant $\mathbb{P}(|Z| \geq z_\alpha) = \alpha$, $Z \sim \mathcal{N}(0,1)$.

Quelques commandes R : Un exemple de fonction R pour calculer l'intervalle de confiance pour $\overline{y}_U$ au niveau $100(1-\alpha)\%$ est décrit ci-dessous :

```
icPESR = function(y, N, niveau) {
n = length(y)
bar_y_w = mean(y)
z = qnorm(1 - (1 - niveau) / 2)
s2_w = sd(y) ^ 2
var_bar_y_w = (1 - n / N) * (s2_w / n)
a = bar_y_w - z * sqrt(var_bar_y_w)
b = bar_y_w + z * sqrt(var_bar_y_w)
print(c(a, b)) }

icPESR(y = c(2.1, 2.3, 4.1, 2.6, 7.1, 8.6), N = 100, niveau = 0.95)
```

Cela renvoie : 2.329876, 6.603457.

3.6 Taille d'échantillon

Incertitude absolue :

Soit ω un échantillon de n individus de U. On appelle incertitude absolue sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$, $\alpha \in]0,1[$, la demi-longueur de $i_{\overline{y}_U}$:

$$d_\omega = z_\alpha s(\overline{y}_\omega) = z_\alpha \sqrt{(1-f)\frac{s_\omega^2}{n}}.$$

Plus d_ω est petit, plus l'estimation de $\overline{y}_U$ par $\overline{y}_\omega$ est précise.

Incertitude relative :

Soit ω un échantillon de n individus de U et d_ω l'incertitude absolue sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$, $\alpha \in]0,1[$. On appelle incertitude relative sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$ le pourcentage $(100 \times d_\omega^*)\%$ où d_ω^* est le réel :

$$d_\omega^* = \frac{d_\omega}{\overline{y}_\omega}.$$

Taille d'échantillon :

Soit ω un échantillon prélevé lors d'une étude préliminaire. La taille d'échantillon n à choisir pour avoir :

- une incertitude absolue sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$, $\alpha \in]0,1[$, inférieure ou égale à d_0 est le plus petit n tel que

$$d_\omega \leq d_0 \qquad \Leftrightarrow \qquad n \geq \frac{N z_\alpha^2 s_\omega^2}{N d_0^2 + z_\alpha^2 s_\omega^2},$$

- une incertitude relative sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$, $\alpha \in]0,1[$, inférieure ou égale à $(100 \times d_1)\%$ est le plus petit n tel que

$$d_\omega^* \leq d_1 \qquad \Leftrightarrow \qquad n \geq \frac{N z_\alpha^2 s_\omega^2}{N(\overline{y}_\omega d_1)^2 + z_\alpha^2 s_\omega^2}.$$

Quelques commandes R : Un exemple de fonction R pour calculer la taille n d'un échantillon à partir de l'incertitude absolue sur $\overline{y}_U$ au niveau $100(1-\alpha)\%$ est décrit ci-dessous :

```
n_ech = function(N, s2, d0, niveau) {
z = qnorm(1 - (1 - niveau) / 2)
n = N * s2 * z ^ 2 / (N * d0 ^  2 + s2 * z ^ 2)
print (ceiling(n)) }

n_ech(N = 1000, s2 = 625, d0 = 3, niveau = 0.95)
```

Cela renvoie 211.

3.7 Sélection des individus

Méthode du tri aléatoire : La méthode du tri aléatoire est un un plan de sondage aléatoire de type PESR. Pour la mettre en œuvre,

- on génère N nombres $x_1, \ldots, x_N$ (indépendamment des uns des autres) suivant la loi uniforme $\mathcal{U}([0,1])$,

- pour tout $i \in \{1, \ldots, N\}$, on affecte à l'individu u_i le nombre x_i,
- on sélectionne les n individus correspondants au n plus grandes valeurs de $x_1, \ldots, x_N$.

Quelques commandes R : Un exemple de commandes R sur la méthode du tri aléatoire est décrit ci-dessous :

```
N = 100
n = 10
x = runif(N)
z = NULL
u = x
for (i in 1:10) {
z[i] = which.max(u)
u[which.max(u)] = 0 }

z
```

3.8 Exercices corrigés

Exercice 1 : *L'objectif de cet exercice est d'illustrer certains résultats théoriques du cours sur les plans de sondage aléatoire de type PESR avec un exemple.* On étudie un caractère Y dans une population de 5 individus : $U = \{u_1, \ldots, u_5\}$. Pour tout $i \in \{1, \ldots, 5\}$, soit y_i la valeur de Y pour l'individu u_i. Les résultats sont :

y_1	y_2	y_3	y_4	y_5
3	4	6	8	13

- Calculer la moyenne-population $\overline{y}_U$ et l'écart-type corrigé-population s_U.
- On prélève au hasard et simultanément 2 individus dans cette population formant ainsi un échantillon. Chaque individu a la même probabilité qu'un autre d'être sélectionné. On est donc dans le cadre PESR.
  - Quel est est le taux de sondage ? Combien d'échantillons peut-on former ? Expliciter les.
  - Pour chaque échantillon ω, calculer la moyenne-échantillon $\overline{y}_\omega$ et l'écart-type corrigé-échantillon s_ω.
  - Soit $\overline{y}_W$ la *var* égale à la moyenne-échantillon, l'aléatoire étant dans l'échantillon considéré. Déterminer sa loi, puis calculer son espérance et sa variance.
  - Soit s_W la *var* égale à l'écart-type corrigé-échantillon, l'aléatoire étant dans l'échantillon considéré. Calculer l'espérance de s_W^2.

- Retrouver les résultats des deux questions précédentes avec les formules du cours.
- Calculer les erreurs dans l'estimation de $\overline{y}_U$.
- Quelle est la probabilité de se tromper de plus de 20% dans l'estimation de $\overline{y}_U$?

Solution :

- On a

$$\overline{y}_U = 6.8, \qquad s_U = 3.9623.$$

- - Le taux de sondage est

$$f = \frac{n}{N} = \frac{2}{5} = 0.4.$$

  Vu le mode de prélèvement, le nombre d'échantillons possibles est

$$\binom{5}{2} = \frac{5!}{2!(5-2)!} = 10.$$

  Ils sont :

$\{u_1, u_2\}$	$\{u_1, u_3\}$	$\{u_1, u_4\}$	$\{u_1, u_5\}$	$\{u_2, u_3\}$
$\{u_2, u_4\}$	$\{u_2, u_5\}$	$\{u_3, u_4\}$	$\{u_3, u_5\}$	$\{u_4, u_5\}$

- On a, en prenant 4 chiffres après la virgule :

ω	Y	$\overline{y}_\omega$	s_ω
$\{u_1, u_2\}$	$\{3, 4\}$	3.5	0.7071
$\{u_1, u_3\}$	$\{3, 6\}$	4.5	2.1213
$\{u_1, u_4\}$	$\{3, 8\}$	5.5	3.5355
$\{u_1, u_5\}$	$\{3, 13\}$	8	7.0710
$\{u_2, u_3\}$	$\{4, 6\}$	5	1.4142
$\{u_2, u_4\}$	$\{4, 8\}$	6	2.8284
$\{u_2, u_5\}$	$\{4, 13\}$	8.5	6.3639
$\{u_3, u_4\}$	$\{6, 8\}$	7	1.4142
$\{u_3, u_5\}$	$\{6, 13\}$	9.5	4.9497
$\{u_4, u_5\}$	$\{8, 13\}$	10.5	3.5355

- Soit $\overline{y}_W$ la *var* égale à la moyenne-échantillon. L'ensemble des valeurs possibles pour $\overline{y}_W$ est

$$\overline{y}_W(\Omega) = \{3.5, 4.5, 5.5, 8, 5, 6, 8.5, 7, 9.5, 10.5\}.$$

  Comme il y a 10 échantillons différents et qu'ils sont équiprobables, la loi de $\overline{y}_W$ est donnée par

k	3.5	4.5	5.5	8	5	6	8.5	7	9.5	10.5
$\mathbb{P}(\overline{y}_W = k)$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

L'espérance de $\overline{y}_W$ est

$$\begin{aligned}\mathbb{E}(\overline{y}_W) &= \sum_{k\in\overline{y}_W(\Omega)} k\mathbb{P}(\overline{y}_W = k)\\ &= \frac{1}{10}(3.5+4.5+5.5+8+5+6+8.5+7+9.5+10.5)\\ &= 6.8.\end{aligned}$$

En utilisant la formule de König-Huyghens, la variance de $\overline{y}_W$ est

$$\mathbb{V}(\overline{y}_W) = \mathbb{E}(\overline{y}_W^2) - (\mathbb{E}(\overline{y}_W))^2 .$$

Or on a $\mathbb{E}(\overline{y}_W) = 6.8$ et

$$\begin{aligned}\mathbb{E}(\overline{y}_W^2) &= \sum_{k\in\overline{y}_W(\Omega)} k^2\mathbb{P}(\overline{y}_W = k)\\ &= \frac{1}{10}(3.5^2+4.5^2+5.5^2+8^2+5^2+6^2+8.5^2+7^2+9.5^2+10.5^2)\\ &= 50.95.\end{aligned}$$

D'où

$$\mathbb{V}(\overline{y}_W) = 50.95 - 6.8^2 = 4.71.$$

- Soit s_W la *var* égale à l'écart-type corrigé-échantillon. L'ensemble des valeurs possibles pour s_W est

$$s_W(\Omega) = \{0.7071, 1.4142, 2.1213, 2.8284, 3.5355, 4.9497, 6.3639, 7.0710\}.$$

Comme il y a 10 échantillons différents et qu'ils sont équiprobables, la loi de s_W est donnée par

k	0.7071	1.4142	2.1213	2.8284	3.5355	4.9497	6.3639	7.0710
$\mathbb{P}(s_W = k)$	$\frac{1}{10}$	$\frac{2}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{2}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

L'espérance de s_W^2 est

$$\begin{aligned}\mathbb{E}(s_W^2) &= \sum_{k\in s_W(\Omega)} k^2\mathbb{P}(s_W = k)\\ &= \frac{1}{10}(0.7071^2 + 2\times 1.4142^2 + 2.1213^2 + 2.8284^2 + 2\times 3.5355^2 + 4.9497^2\\ &+ 6.3639^2 + 7.0710^2)\\ &= 15.6997.\end{aligned}$$

- En utilisant les formules du cours, on retrouve les résultats précédents (en prenant en compte les approximations) :

$$\mathbb{E}(\overline{y}_W) = \overline{y}_U = 6.8, \qquad \mathbb{V}(\overline{y}_W) = (1-f)\frac{s_U^2}{n} = (1-0.4)\frac{3.9623^2}{2} = 4.71$$

et

$$\mathbb{E}(s_W^2) = s_U^2 = 15.6998.$$

- On utilise la formule d'erreur d'estimation :

$$e_\omega = |\overline{y}_\omega - \overline{y}_U| = |\overline{y}_\omega - 6.8|.$$

On a, en prenant 4 chiffres après la virgule :

ω	$\overline{y}_\omega$	e_ω
$\{u_1, u_2\}$	3.5	3.3
$\{u_1, u_3\}$	4.5	2.3
$\{u_1, u_4\}$	5.5	1.3
$\{u_1, u_5\}$	8	1.2
$\{u_2, u_3\}$	5	1.8
$\{u_2, u_4\}$	6	0.8
$\{u_2, u_5\}$	8.5	1.7
$\{u_3, u_4\}$	7	0.2
$\{u_3, u_5\}$	9.5	2.7
$\{u_4, u_5\}$	10.5	3.7

- On a $20\% = (100\times\beta)\%$ avec $\beta = 0.2$. Le nombre de e_ω dépassant $\beta\times\overline{y}_U = 0.2\times 6.8 = 1.36$ est de 6. Donc la probabilité de se tromper de plus de $(100\times\beta)\%$

dans l'estimation de $\overline{y}_U$ par $\overline{y}_W$ est

$$p = \frac{1}{\binom{N}{n}} \sum_{\omega \in W(\Omega)} \mathbb{1}_{\{e_\omega \geq \beta \times \overline{y}_U\}} = \frac{6}{10} = 0.6.$$

Il y a 60% chances de se tromper de plus de 20% en estimant $\overline{y}_U$ par $\overline{y}_W$.

Exercice 2 : On prélève 25 sacs de farine de maïs dans une usine en contenant 200 suivant un plan de sondage aléatoire de type PESR. On pèse ces 25 sacs. Les valeurs obtenues donnent une moyenne de 13.5 kilogrammes et un écart-type corrigé de 1.3 kilogrammes.

Déterminer un intervalle de confiance pour la moyenne des poids des 200 sacs de farine de maïs au niveau 95%.

Solution : On a $95\% = 100(1-\alpha)\%$ avec $\alpha = 0.05$. On a $\mathbb{P}(|Z| \geq z_\alpha) = \alpha = 0.05$, $Z \sim \mathcal{N}(0,1)$, avec $z_\alpha = 1.96$. Un intervalle de confiance pour la moyenne des poids des 200 sacs de farine $\overline{y}_U$ au niveau 95% est

$$\begin{aligned} i_{\overline{y}_U} &= \left[\overline{y}_\omega - z_\alpha \sqrt{(1-f)\frac{s_\omega^2}{n}},\ \overline{y}_\omega + z_\alpha \sqrt{(1-f)\frac{s_\omega^2}{n}}\right] \\ &= \left[13.5 - 1.96\sqrt{\left(1 - \frac{25}{200}\right)\frac{1.3^2}{25}},\ 13.5 + 1.96\sqrt{\left(1 - \frac{25}{200}\right)\frac{1.3^2}{25}}\right] \\ &= [13.0233,\ 13.9766]. \end{aligned}$$

Ainsi, il y a 95 chances sur 100 que $[13.0233,\ 13.9766]$ contienne $\overline{y}_U$, l'unité étant le kilogramme.

Exercice 3 : On dispose d'une liste de 500 foyers avec, pour chacun d'entre eux, le nombre d'individus y vivant. Sur un échantillon de 8 foyers constitué par un plan de sondage aléatoire de type PESR, les résultats sont :

3	6	1	2	4	4	1	8

- Calculer le taux de sondage.
- Donner une estimation ponctuelle de la moyenne des effectifs des 500 foyers.
- Donner une estimation ponctuelle de l'écart-type corrigé de l'estimateur de la moyenne des effectifs des 500 foyers.
- Déterminer un intervalle de confiance au niveau 95% pour la moyenne-population.
- Déterminer la taille d'échantillon à choisir pour avoir une incertitude absolue sur la moyenne-population inférieure ou égale à 1 au niveau 95%.

Solution :

- On a $n = 8$ et $N = 500$. Le taux de sondage est

$$f = \frac{n}{N} = \frac{8}{500} = 0.016.$$

- Une estimation ponctuelle de la moyenne des effectifs des 500 foyers est la moyenne échantillon :

$$\overline{y}_\omega = 3.625.$$

- Une estimation ponctuelle de l'écart-type corrigé de l'estimateur de la moyenne des effectifs des 500 foyers est

$$s(\overline{y}_\omega) = \sqrt{(1-f)\frac{s_\omega^2}{n}} = \sqrt{(1-0.016)\frac{2.4458^2}{8}} = 0.8577.$$

- On a $95\% = 100(1-\alpha)\%$ avec $\alpha = 0.05$. On a $\mathbb{P}(|Z| \geq z_\alpha) = \alpha = 0.05$, $Z \sim \mathcal{N}(0,1)$, avec $z_\alpha = 1.96$. Un intervalle de confiance pour $\overline{y}_U$ au niveau 95% est

$$\begin{aligned} i_{\overline{y}_U} &= [\overline{y}_\omega - z_\alpha s(\overline{y}_\omega),\ \overline{y}_\omega + z_\alpha s(\overline{y}_\omega)] \\ &= [3.625 - 1.96 \times 0.8577,\ 3.625 + 1.96 \times 0.8577] \\ &= [1.9439,\ 5.3060]. \end{aligned}$$

  Ainsi, il y a 95 chances sur 100 que $[1.9439,\ 5.3060]$ contienne $\overline{y}_U$.

- On a $95\% = 100(1-\alpha)\%$ avec $\alpha = 0.05$. On souhaite déterminer le plus petit n tel que :

$$d_\omega = z_\alpha\sqrt{(1-f)\frac{s_\omega^2}{n}} \leq d_0 \qquad \Leftrightarrow \qquad n \geq \frac{N z_\alpha^2 s_\omega^2}{N d_0^2 + z_\alpha^2 s_\omega^2},$$

  avec $d_0 = 1$, $z_\alpha = 1.96$, ω est l'échantillon considéré précédemment, $s_\omega = 2.4458$ et $N = 500$. On a

$$\frac{500 \times 1.96^2 \times 2.4458^2}{500 \times 1^2 + 1.96^2 \times 2.4458^2} = 21.97044.$$

  Donc $n = 22$ convient.

4 Questionnaire

4.1 La base

Questionnaire : Un questionnaire est une liste de questions posées en vue d'une enquête.

Principes de base : Pour mettre en place un questionnaire efficace, il ne faut pas perdre de vue les objectifs à atteindre, et articuler les questions autour de cet objectif. Le bon sens est de mise.

Le questionnaire doit

- avoir une structure claire et fluide,
- être relativement court (pour augmenter les chances de réponses).

Pour ce faire, il faut classer les questions par thème et toujours formuler des questions courtes et compréhensibles. De plus, l'ordre des questions agit sur le résultat de l'enquête. Pour chaque thème, il est donc nécessaire de structurer le questionnaire.

Méthode de l'entonnoir : Souvent, "la méthode de l'entonnoir" est appropriée. Cette méthode consiste à

- dans un premier temps, poser des questions d'ordre général (présenter le sujet de l'enquête, clarifier le lien entre l'individu interrogé et le sujet, etc.)
- puis enchaîner les questions afin de préciser les choses, jusqu'à poser des questions d'ordre personnel.

Question de contrôle : Si l'enquête traite d'un sujet plutôt sensible, pour vérifier le sérieux ou la cohérence des réponses des individus interrogés, il convient d'insérer une "question de contrôle" qui aborde un sujet déjà abordé plus tôt dans le questionnaire. Si la réponse à cette question n'est pas en adéquation avec une précédente réponse, on ne considère pas le questionnaire.

Test : Avant diffusion, il est conseillé de tester le questionnaire auprès de proches pour identifier d'éventuels problèmes.

Remerciement : À la fin du questionnaire, penser à remercier l'individu interrogé.

4.2 Type de questions

Les questions peuvent être de plusieurs types.

Question fermée à choix unique : La question fermée à choix unique a l'avantage d'être rapide et simple à traiter mais, elle ne laisse place à aucune nuance.

Exemple : En reprenant le contexte "étudiants" : Êtes-vous content de votre formation ? [Oui ou Non]

Question fermée à choix multiples : La question fermée à choix multiples a l'avantage d'être rapide et simple à traiter, mais les réponses proposées peuvent induire un biais. Les choix ne doivent pas être trop nombreux, sinon il y a risque de réponse au hasard ou non-réponse.

Exemple : En reprenant le contexte "étudiants" : Êtes-vous content de votre formation ? [Oui, Partiellement, ou Non]

Question ouverte : La question ouverte permet une spontanéité et une richesse des réponses. En revanche, la multiplicité des réponses possibles entraîne une difficulté dans leur traitement. Aussi, beaucoup de non-réponses sont possibles.

Exemple : En reprenant le contexte "étudiants" : Que faudrait-il changer pour améliorer votre formation ? Réponses possibles : "Plus d'heures de cours", "Moins de sévérité dans les notations", etc.

Question offrant une échelle de réponses : La question offrant une échelle de réponses a l'avantage de nuancer quantitativement les réponses. La graduation reste toutefois subjective.

Exemple : En reprenant le contexte "étudiants" : Dans quelle mesure êtes-vous satisfait des services de votre garagiste ? [Très déçu 0 1 2 3 4 5 6 7 8 9 10 très satisfait]

4.3 Biais

Dans la formulation des questions ou des réponses, il faut éviter d'introduire des biais qui peuvent fausser la précision d'un questionnaire.

Complexité : Il faut éviter les mots trop techniques, ou vague.

Exemple : En reprenant le contexte "étudiants" : Un exemple de question compliquée est : "Qu'est-ce qui vous messiez le plus dans votre formation ?" au lieu de "Qu'est-ce qui vous déplait le plus dans votre formation ?"

Il faut éviter les questions complexes (double négation, etc.) : elles sont peu claires et vont inciter l'individu interview à abandonner.

Exemple : En reprenant le contexte "étudiants" : Un exemple de question à double négation est : "Ne pensez-vous pas que votre formation n'est pas assez complète ?"

Appréciation : Évitez les réponses de type appréciation : souvent, régulièrement, rarement, etc. Ces réponses sont en fait subjectives. Préférez les réponses chiffrées de manière claire (comptage, fréquence, intervalles de valeurs, etc.)

Réponse forcée : Ne posez pas des questions dont le contexte ou la bonne conscience influence une réponse (questions orientées, effet d'halo, etc.).

Exemple : En reprenant le contexte "étudiants" : Un exemple de question "sensible" est : "Pensez-vous que votre formation est améliorable ?" Il est probable que l'étudiant interrogé réponde "oui".

Il y a beaucoup d'autres exemples, mais avec du bon sens, on outrepasse le problème du biais.

4.4 Réalisation de l'enquête

La réalisation de l'enquête dépend principalement des moyens mis en œuvre (moyens humains, budget, etc.). Les principaux types d'enquêtes sont les suivants :

L'enquête face à face : L'enquête face à face consiste à interroger les individus dans la rue, à leur domicile, à la sortie de leur travail, zone de chalandise, etc. En cas d'enquête face à face, l'attitude de l'enquêteur peut influencer l'individu interrogé : il faut éviter les formes d'"intimidation, ou gestes, conscients ou non, qui trahissent les opinions : lever les yeux au ciel, souffler, etc.

L'enquête téléphonique : L'enquête téléphonique consiste à interroger les individus par téléphone. Elle est économique, mais beaucoup d'appels n'aboutissent pas et le risque que l'individu mette fin subitement au questionnaire est réel.

L'enquête par courrier : L'enquête par courrier consiste à interroger les individus par courrier. Il faut prévoir à mettre une enveloppe déjà affranchie pour espérer avoir plus de retours, et, éventuellement, promettre un lot aux répondants.

L'enquête par internet : L'enquête par internet consiste à remplir un questionnaire en ligne. Les avantages sont nombreux (simplicité, rapidité de traitement, etc.). Par contre, le risque de réponses à la va-vite est réel.

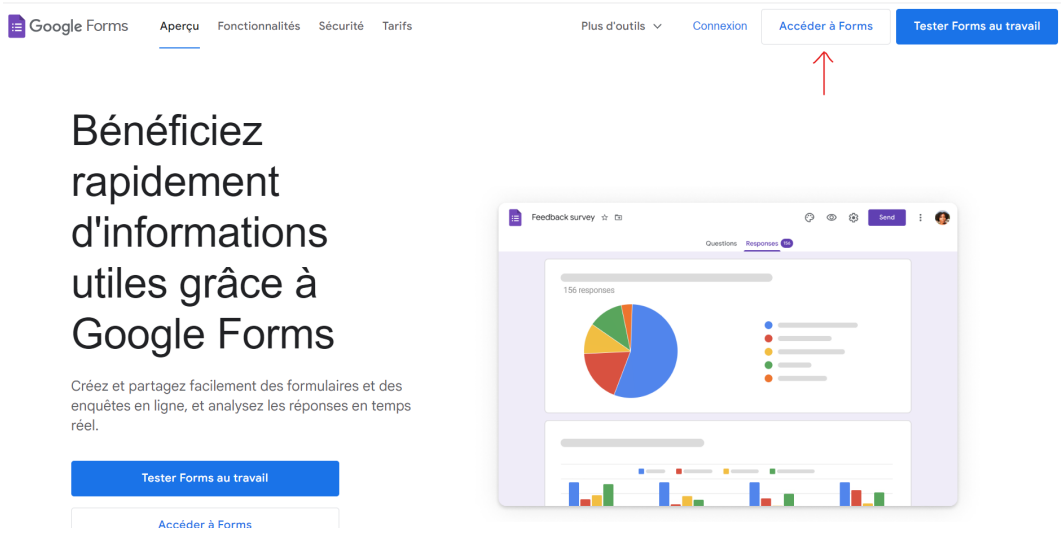
4.5 Google Forms

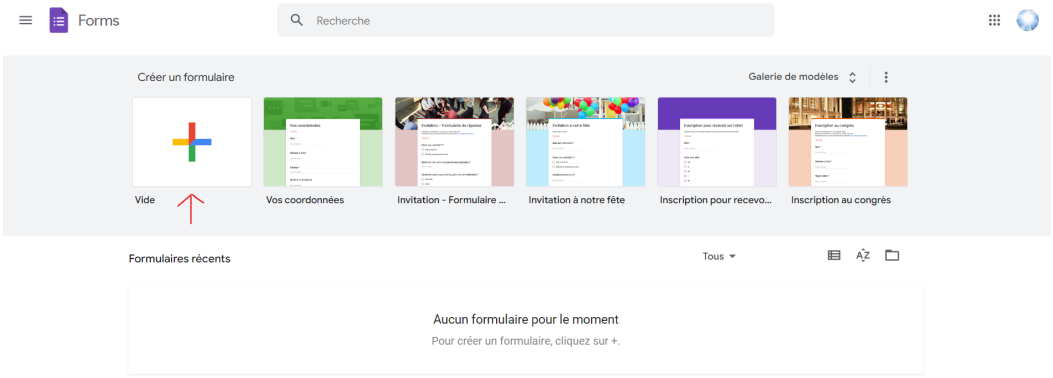
L'outil (gratuit) le plus performant pour réaliser un questionnaire est Google Forms. Il est simple à utiliser et permet la collecte des données dans un fichier que l'on peut exploiter par la suite. On peut le laisser en ligne ou l'imprimer pour une enquête terrain.

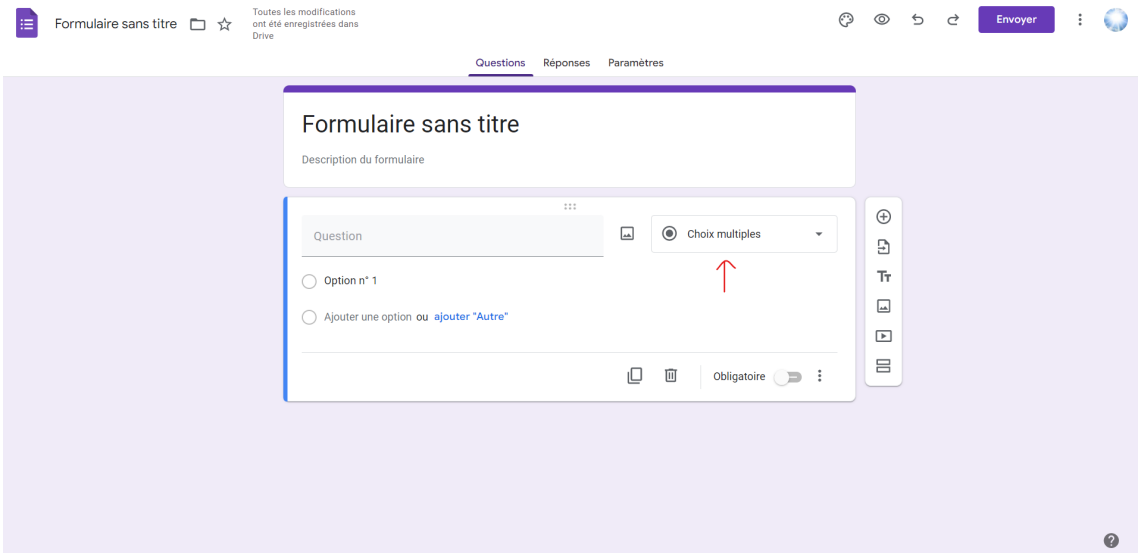
Pour l'utiliser, il faut

- avoir un compte google et s'y connecter,
- se rendre à l'adresse internet : `https://www.google.fr/intl/fr/forms/about/`

Dès lors, il faut suivre les flèches rouges comme indiquées :


Google Forms
Aperçu
Fonctionnalités
Sécurité
Tarifs
Plus d'outils
Connexion
Accéder à Forms
Tester Forms au travail
Bénéficiez rapidement d'informations utiles grâce à Google Forms
Créez et partagez facilement des formulaires et des enquêtes en ligne, et analysez les réponses en temps réel.
Tester Forms au travail
Accéder à Forms
Feedback survey
Send
Questions
Responses
156 responses


Forms
Recherche
Créer un formulaire
Galerie de modèles
Vide
Vos coordonnées
Invitation - Formulaire ...
Invitation à notre fête
Inscription pour recevo...
Inscription au congrès
Formulaires récents
Tous
Aucun formulaire pour le moment
Pour créer un formulaire, cliquez sur +.


Formulaire sans titre
Toutes les modifications ont été enregistrées dans Drive
Envoyer
Questions
Réponses
Paramètres
Formulaire sans titre
Description du formulaire
Question
Choix multiples
Option n° 1
Ajouter une option ou ajouter "Autre"
Obligatoire

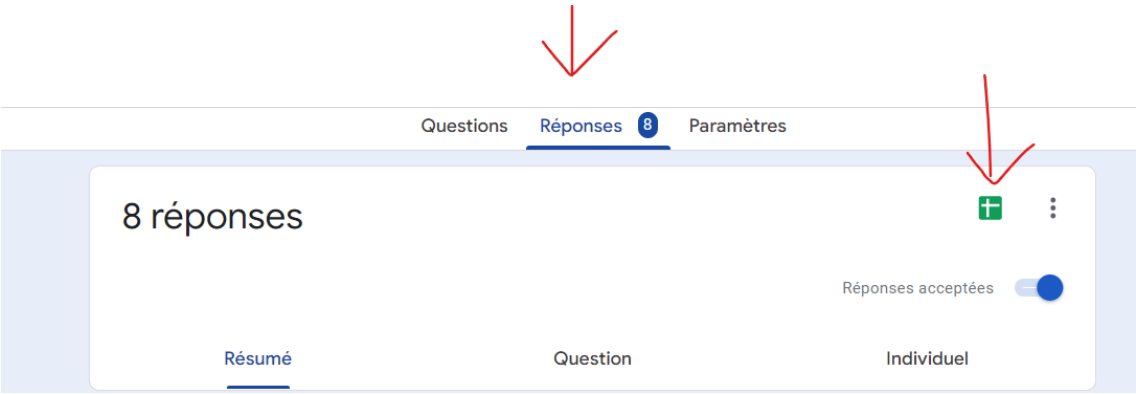
Les différents types de réponses disponibles sont les suivants : réponse courte, paragraphe, choix multiples, cases à cocher, liste déroulante, importer un fichier : (il permet aux répondants d'ajouter une pièce jointe en guise de réponse), échelle linéaire (elle permet de demander une note sur une échelle), grille à choix multiples, date, et heure.

Il y a de nombreuses autres options qui sont assez intuitives à utiliser (barre de progression, modification du design, etc.).

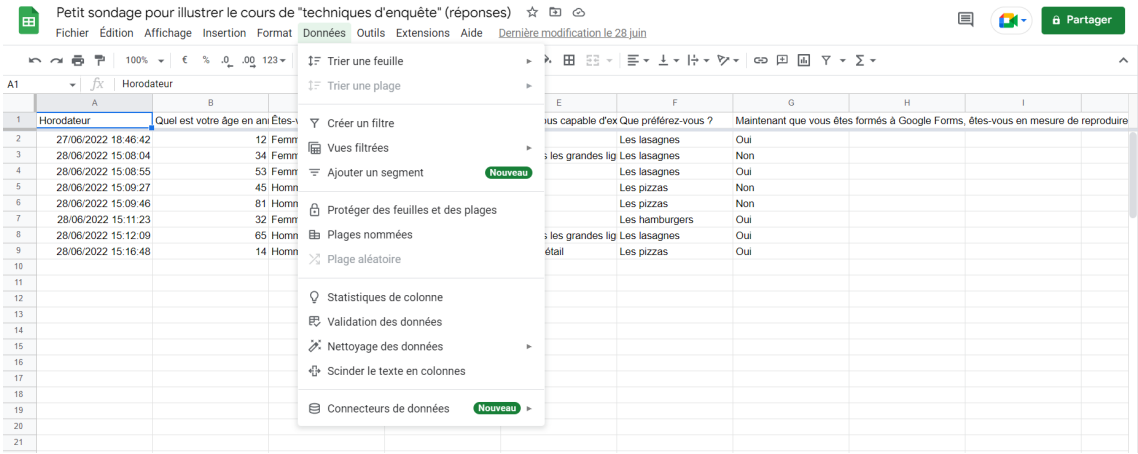
Pour une prise en main simple et rapide, on peut s'entraîner avec le sondage tutorial disponible à cette adresse :

`https://docs.google.com/forms/d/e/1FAIpQLSctd4DTS_uWVeTO2SMbufp7jSQe6KjXu9HdRm6JMWe8DSr0tQ/viewform?usp=sf_link`

Une fois le questionnaire rempli de tous, on accède aux réponses en cliquant sur "Réponses", puis en cliquant sur l'icône verte en haut à droite.



On obtient alors une feuille de calcul Google Forms qui se manipule comme une feuille Excel classique. Entre autres, on peut y faire du nettoyage de données basique et de la statistique (description des données, tableau croisé, etc.).



Si l'on souhaite faire un traitement des données approfondi, on privilégiera l'utilisation de R, qui offre une panoplie plus large d'outils.

4.6 Sur l'analyse des résultats

Partant du fichier de données, on distingue trois niveaux d'analyse possibles :

Les tris à plat : Les tris à plat consistent à analyser les données des caractères pris à un à un (calculs des moyennes, écart-types, intervalles de confiance, test sur un paramètre inconnu, etc.).

Les tris croisés : Les tris croisés consistent à analyser les données de couples de caractères simultanément, quand cela à du sens (calculs des coefficients de corrélations, test de dépendance, régression, etc.).

Analyse multivariée : L'analyse multivariée consiste à analyser les données de plusieurs caractères simultanément (plus que deux) (analyse en composantes principales, régression multiple, analyse des correspondance multiples, analyse de la variance, etc.)

Il est recommandé de faire un maximum de tableaux et de graphiques de qualité au fur et à mesure. Les plus intéressants seront gardés en vue de compléter le rapport final.

4.7 Complément : Biais des non-réponses

Les non-réponses peuvent gêner l'exploitation statistique d'un questionnaire. Pour éviter un grand nombre de non-réponses, il est conseillé d'utiliser dans la mesure du possible un mode de collecte qui réduise le taux de non-réponses (par exemple, l'enquête face à face minimise le nombre de non-réponses par rapport à l'enquête par courrier). Bien évidemment, l'idéal étant de n'avoir aucune non-réponse.

On traite la non-réponse en fonction de sa nature :

La non-réponse totale : L'individu n'a pas du tout répondu à l'enquête. Dans ce cas, on traite la non-réponse par repondération : on augmente le poids des réponses des individus répondants de sorte à ce que, dans les formules des grandeurs associées (moyennes, écart-types, etc.), elles compensent celles des non-répondants.

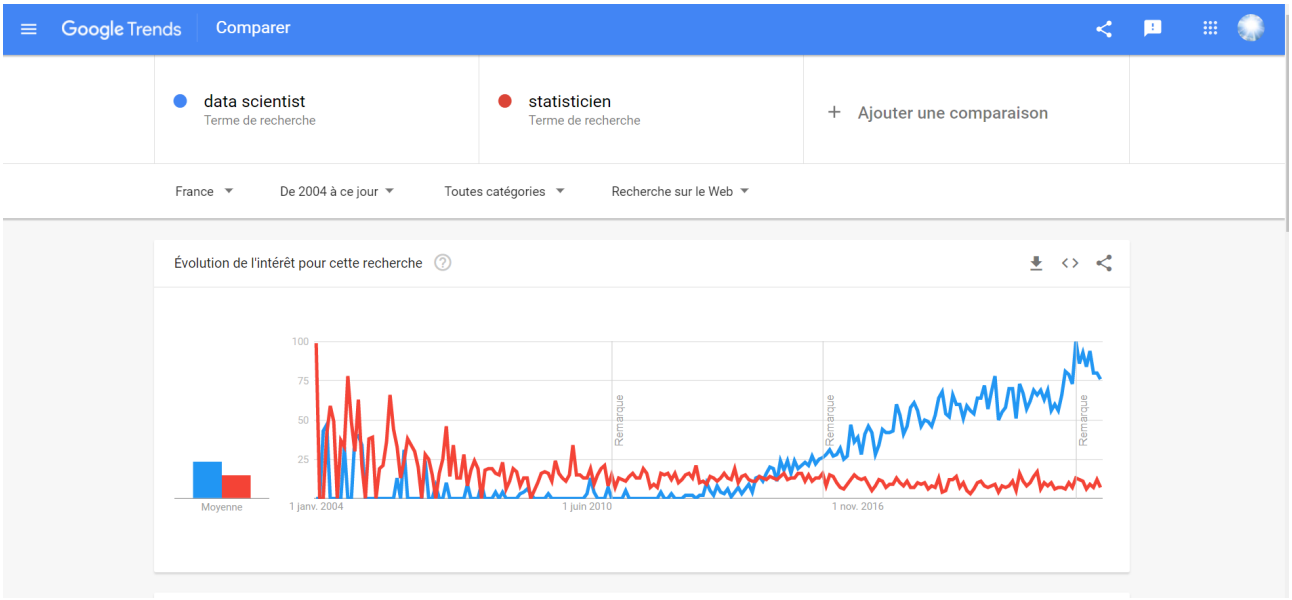
La non-réponse partielle : L'individu n'a pas répondu à certaines questions. Dans ce cas, on peut traiter la non-réponse en excluant toutes les données associées à l'individu, ou utiliser l'imputation : on remplace les réponses manquantes par des réponses plausibles (de nombreuses techniques probabilistes existent).

4.8 Complément : Google Trends

Dans la conception des questions ou l'analyse et interprétation des résultats, il peut être intéressant de connaître "les tendances" du sujet que l'on aborde. À cet égard, un outil potentiellement intéressant est Google Trends disponible à l'adresse internet : `https://trends.google.fr/trends/?geo=FR`

Google Trends indique ce que les gens recherchent dans Google, en temps réel. Nous pouvons utiliser ces données pour mesurer l'intérêt de la recherche pour un sujet particulier, à un endroit particulier et à un moment particulier. On peut aussi voir l'évolution dans le temps de la recherche d'un sujet particulier, et aussi de le comparer à un autre sujet. Les donnés sont disponibles en format csv, et l'on peut enregistrer les graphiques

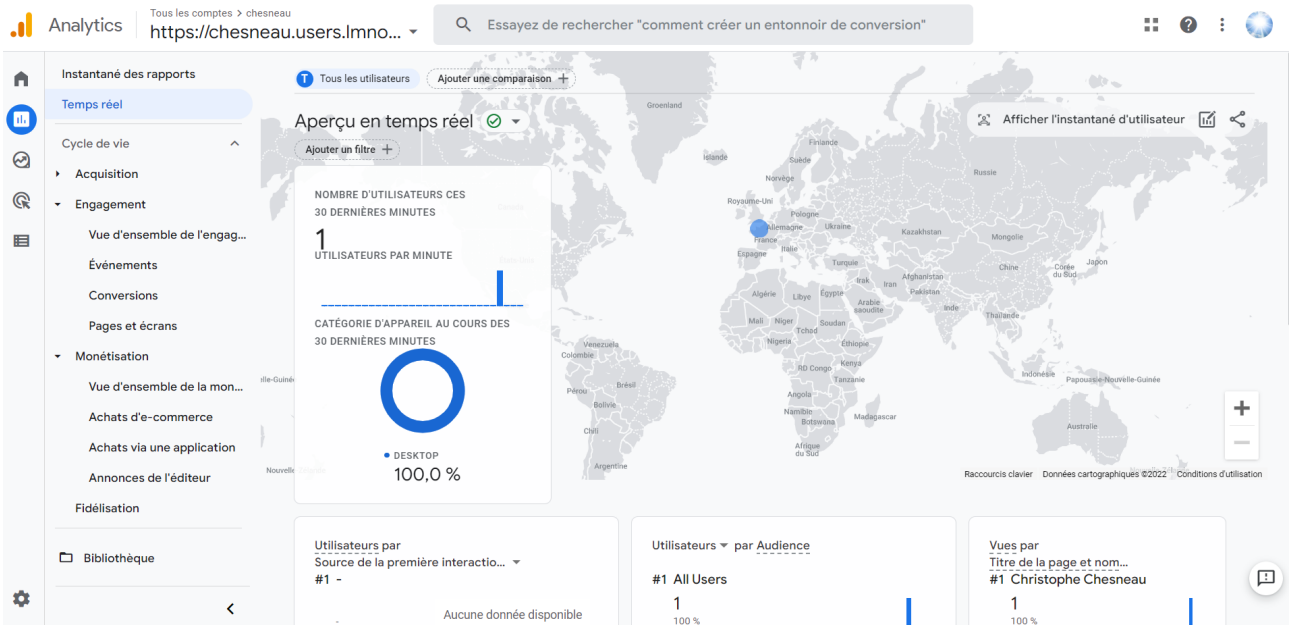
Par exemple, l'image suivante montre l'évolution dans le temps de la recherche des mots "statisticien" et "data scientist".



4.9 Complément : Google Analytics

Si l'on dispose d'un site internet, il peut être intéressant d'avoir des informations sur les visiteurs, en vue d'une enquête notamment. À cet égard, un outil potentiellement intéressant est Google Analytics disponible à l'adresse internet : `https://analytics.google.com` Il faut alors insérer un code html (fourni dans Google Analytics, code appelé "tags") dans les codes du site, puis remplir les informations demandées (identité, url du site, etc.). Entre autres, Google Analytics donne permet à la fois le nombre de visiteurs et leurs origines géographiques. Cela peut donner des informations importantes sur les individus à ciblés pour une étude en lien avec le site internet.

Un visuel du tableau de bord de Google Analytics est disponible dans l'image ci-dessous.



Dans ce visuel, le site est visité “en temps réel” par une personne en France. Les données sont ainsi collectées dans le temps.

5 Nettoyage les données

5.1 Présentation

Bien souvent, les données brutes issues d'un questionnaire ou autre possèdent des petites erreurs de saisie ou d'uniformisation qui empêchent leur traitement direct par un logiciel. C'est particulièrement vrai pour les gros volume de données. Il s'agit donc de résoudre ces problèmes avant toute chose. On parle alors de nettoyage des données. Les points classiques à vérifier sont :

- les valeurs ou réponses hors des normes fixées,
- les valeurs extrêmes (détectables avec des boîtes à moustaches, entre autres),
- les valeurs manquantes (sujet déjà abordé dans la sous-section "Complément : Biais des non-réponses"),
- les doublons (plusieurs réponses identiques d'un même individu due à de multiples validations malencontreuses, ou autre).

La plupart du temps, on peut régler ses problèmes un à un avec les options permettant de lire ou manipuler le jeu de données. Autrement, des solutions simples et automatiques existent. Notamment, il y a une option "Nettoyage des données" dans la feuille de calcul des résultats de Google Forms (Google Sheets), mais celle-ci reste limitée. Ci-dessous, on se focalise sur les possibilités offertes par R.

Exemple : Si l'on demande à cinq personnes différentes d'écrire 12345 euros et 67 centimes, on peut avoir les résultats suivants : 12,345.67, €12.345,67, 12,345.67, 12345,67, 12345.67 euros. Si ces réponses sont saisies telles quelles, aucun logiciel ne va comprendre qu'il s'agit d'une même valeur répétée 5 fois, à savoir 12345. Il y a évidement un grand nombre de petits problèmes de ce genre qui peut se greffer dans les données. Il faut donc pouvoir nettoyer de manière ces données de manière simple et automatique, de sorte à ce que le logiciel puisse travailler avec les "vraies" données.

Exemple : On s'intéresse au prix d'un jeu vidéo sur le site internet *LaBonneAffaire*. Sur un échantillon de 30 annonces obtenu, les résultats, en euros, sont les suivants :

26	33	23	30	24	36	27	29	22	28	37	31	34	28	31
28	35	30	33	27	32	39	28	33	32	28	31	30	11	37

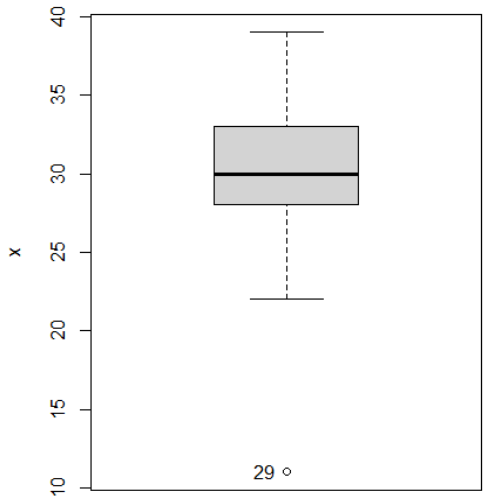
On peut voir que la valeur 11 est anormalement basse par rapport aux autres. Cela peut-être une erreur de saisie ou autre. On aurait pu la détecter en faisant une boîte à moustaches :

```
x = c(26, 33, 23, 30, 24, 36, 27, 29, 22, 28, 37, 31, 34, 28, 31, 28, 35, 30, 33,
27, 32, 39, 28, 33, 32, 28, 31, 30, 11, 37)
library(car)
Boxplot(x, id = TRUE)
```

Cela renvoie :

```
[1] 29
```

ainsi que le graphique :



De plus, on fait :

```
x[29]
```

Cela renvoie :

```
[1] 11
```

Ainsi, le 29 ième individu ressort comme étant associé à la valeur anormale 11.

5.2 Données manquantes avec R

Par défaut, le logiciel R propose de gérer les valeurs manquantes en excluant toutes les données associées aux individus concernés (en quelque sorte, si un individus possède au moins une donnée manquante, il est retiré des données). Les données manquantes sont représentées sous le logiciel R par `NA` signifiant "Not Available". Pour les retrouver, On détecte la présence de données manquantes à l'aide des fonctions `is.na` ou `summary`, et on peut les enlèver d'une traite à l'aide de la fonction `na.omit`.

Des solutions plus fines sont possibles, comme l'imputation (voir les packages).

Exemple : On considère le jeu de données `Auto`, lequel possèdent des valeurs manquantes notées par `?`. Dès lors, on lit le jeu de données en précisant la présence de ces valeurs manquantes en faisant :

```
Auto = read.table(file = "https://chesneau.users.lmno.cnrs.fr/auto-mpg.txt", header = F,
sep = "", dec = ".", na.strings = "?"
)
```

On vérifie la présence des valeurs manquantes en faisant :

```
which(is.na(Auto), arr.ind = TRUE)
```

Cela renvoie :

```
     row col
[1,]  33   4
[2,] 127   4
[3,] 331   4
[4,] 337   4
[5,] 355   4
[6,] 375   4
```

Ainsi, il y a 6 données manquantes situées sur la 4 ième colonne, et aux lignes 33, 127, 331, 337, 355 et 375.

On aurait pu utilisaer la fonction `summary` :

```
summary(Auto)[, 4]
# ou summary(Auto) tout court
```

Cela renvoie :

```
"Min.   : 46.0  " "1st Qu.: 75.0  " "Median : 93.5  " "Mean   :104.5  " "3rd Qu.:126.0  "
"Max.   :230.0  "     "NA's   :6  "
```

Ainsi, la 4 ième colonne contient 6 valeurs manquantes.

On créée un nouveau jeu de données `Autonew` en enelevant toutes les lignes (individus) contenant des données manquantes en faisant :

```
Autonew = na.omit(Auto)
```

On peut vérifier que cela a bien marché en faisant :

```
which(is.na(Autonew), arr.ind = TRUE)
```

Cela renvoie :

```
     row col
```

Les données manquantes ont bien été enlevées.

5.3 Package `cleaner`

Le librairie `cleaner` offre des fonctions qui rende le nettoyage des données simple et rapide avec des fonctions qui "corrigent les problèmes évidents par elles-mêmes".

La fonction clé est `clean` qui permet un nettoyage globale des données, au format directement traitable avec le logiciel R. Cependant, il faut souvent utiliser des variantes précises de `clean` pour obtenir ce que l'on veut.

Les fonctions principales sont :

- `clean_logical` : fonction permettant de définir les valeurs TRUE/FALSE.
- `clean_factor` : fonction permettant de définir et redéfinir un facteur.
- `clean_numeric` : fonction permettant de supprimer tous les non-numériques du texte d'entrée encombré.
- `clean_Date` : fonction permettant de définir et redéfinir tout type de dates.
- `clean_character` : fonction permettant de supprimer tous les non-caractères évidents.
- `clean_percentage` : fonction permettant d'utiliser la nouvelle classe de pourcentage fournie avec le librairie `cleaner`.
- `clean_currency` : fonction permettant d'utiliser la nouvelle classe monétaire fournie avec le librairie `cleaner`.

Exemple : Pour une prise en main rapide, on va travailler sur plusieurs petits exemples, sans trop sapeusantir ; le librairie `cleaner` étant complémentaire dans l'analyse.

Dans un premier temps, on installe et charge le librairie `cleaner` en faisant :

```
install.packages("cleaner")
library(cleaner)
```

Pour revenir à nos données relatives aux “12345 euros et 67 centimes”, on fait :

```
a = c("$ 12,345.67", "eur 12.345,67", "12,345.67", "12345,67", "12345.67 euro")
```

On nettois les données en faisant :

```
clean(a)
# ou clean_numeric(a)
```

Cela renvoie :

```
Note: Assuming class 'numeric'
[1] 12345.67 12345.67 12345.67 12345.67 12345.67
```

On constate que les valeurs ont été uniformiser en écriture, et sont donc exploitabes directement.

Un exemple de nettoyage avec des données de type logique est le suivant :

```
b = c("YES", "Yes", "No", "no", "NO")
clean(b)
# ou clean_logical(b)
```

Cela renvoie :

```
Note: Assuming class 'logical'
[1]  TRUE  TRUE FALSE FALSE FALSE
```

Un exemple de nettoyage avec des données de type qualitative est le suivant :

```
c = c("Male 23", "Female 36", "Male 48", "Male 29", "YY Female")
clean_factor(c, levels = c("M", "F"))
```

Cela renvoie :

```
[1] M F M M F
Levels: M F
```

Un exemple de nettoyage avec des données de type prix est le suivant :

```
d = c("Received $25", "Received $31.40", "Received $28", "Received $11.2",
 "Received $25")
clean_currency(d)
```

Cela renvoie :

```
[1] 'USD 25.00' 'USD 31.40' 'USD 28.00' 'USD 11.20' 'USD 25.00'
```

Un exemple de nettoyage avec des données de type charactère est le suivant :

```
e = c("Admis 20", "Admis 15", "Non-Admis 02", "Non-Admis 07", "Admis 12")
clean_character(e)
```

Cela renvoie :

```
[1] "Admis"    "Admis"    "NonAdmis" "NonAdmis" "Admis"
```

6 Le “tidyverse”

6.1 Présentation

Les données d'une enquête doivent être traitées avec efficacité et précision, aussi bien dans leur mise en forme que dans leur analyse statistique. On peut alors utiliser le logiciel R, lequel propose, entre autres, le “tidyverse”. Le tidyverse est une collection de librairies R conçus pour la manipulation de données en tout genre. Ils sont, entre autres, bien adaptés pour de gros volumes de données au sens large du terme (le fameux Big Data, aussi appelé 3Vs pour Volumétrie, Variété, et Vitesse). Toutes les librairies partagent une philosophie de conception, une grammaire et des structures de données sous-jacentes. Les principales librairies du tidyverse sont les suivants :

ggplot2 : `ggplot2` propose un système de création déclarative de graphiques, basé sur une grammaire particulière (appelée The Grammar of Graphics).

dplyr : `dplyr` offre une grammaire de manipulation de données, fournissant un ensemble cohérent de verbes qui résolvent les problèmes de manipulation de données les plus courants.

tidyr : `tidyr` fournit un ensemble de fonctions qui vous aident à ranger vos données. Les données ordonnées sont des données avec une forme cohérente : en bref, chaque caractère va dans une colonne, et chaque colonne est un caractère.

readr : `readr` fournit un moyen rapide et convivial de lire des données rectangulaires (comme csv, tsv et fwf). Il est conçu pour analyser de manière flexible de nombreux types de données.

tibble : `tibble` est une réinvention moderne du célèbre `data.frame`. En quelques sortes, les tibbles sont des dataframes “paresseux” et “hargneux” : ils en font moins et se plaignent davantage, ce qui vous oblige à affronter les problèmes plus tôt, mais qui conduit généralement à un code plus propre et plus fonctionnel.

stringr : `stringr` fournit un ensemble cohérent de fonctions conçues pour rendre le travail avec les chaînes aussi simple que possible. Il est construit sur stringi, qui utilise la bibliothèque ICU C pour fournir des implémentations rapides et correctes des manipulations de chaînes courantes.

forcats : `forcats` fournit des outils pour manipuler les modalités (ou facteurs) de caractères qualitatifs (forcats étant l'anagramme de factors).

Il existe d'autres outils associés au tidyverse, mais ceux-ci sont plus spécifiques (`purr`, `readxl`, `haven`, etc.).

6.2 Package `tibble`

L'objectif de cette section est d'appréhender la librairie `tibble`. Comme déjà dit, les tibbles sont des versions modernes des dataframes. Ils conservent les fonctionnalités qui ont résisté à l'épreuve du

temps et abandonnent les fonctionnalités qui étaient auparavant pratiques mais qui sont maintenant frustrantes (comme, par exemple, convertir des vecteurs de caractères en facteurs). On installe et charge la librairie `tibble` en faisant :

```
install.packages("tibble")
library(tibble)
% ou, plus global: install.packages("tidyverse")
% library(tidyverse)
```

Parmi les différences entre `tibble` et le classique `dataframe`, il y a

- `tibble` ne modifie pas le nom des colonnes. Pour s'en convaincre, on fait :
  `names(data.frame("Dusty Bun" = 1))`
  Cela renvoie : `"Dusty.Bun"` (un point a été ajouté sans raison).
  Puis :
  `names(tibble("Dusty Bun" = 1))`
  Cela renvoie : `"Dusty Bun"`
- Lorsque l'on affiche un tableau `tibble`, il n'affiche que les dix premières lignes et toutes les colonnes qui tiennent sur un seul écran. Il imprime également une description abrégée du type de colonne et utilise des styles de police et des couleurs pour la mise en surbrillance. Par exemple, on fait :

```
w = tibble(
  x = 1:10,
  y = rnorm(10),
  z = x ** 2 + y
)
```

Cela renvoie :

```
# A tibble: 106 x 2
       x         y
   <int>     <dbl>
 1    -5     0.508
 2    -4     1.52
 3    -3     4.57
 4    -2    13.7
 5    -1    41.2
 6     0   123.
 7     1   370.
 8     2  1111.
 9     3  3333.
10     4 10000.
```

```
# ... with 96 more rows
```

Une fois `w` définie, on peut alors faire les manipulations basiques suivantes :

```
w$x
w["x"]
w[c("x","y")]
w[1:5, "y"]
```

- un tableau `tibble` ne supporte pas les noms de lignes.

Pour finir cette partie, on convertit un `data.frame` ou autre en format `tible` à l'aide de la commande `as_tibble`.

6.3 Package `dplyr`

L'objectif de cette section est d'appréhender la librairie `dplyr`.

En étant bref, la librairie `dplyr` est dédiée à la manipulation de données ; il permet de faire des manipulations complexes avec simplicité et rapidité en temps de calcul. C'est une librairie membre d'un ensemble de librairies appelé `tidyverse`. On l'installe et charge en faisant :

```
install.packages("dplyr")
library(dplyr)
% ou, plus global: install.packages("tidyverse")
% library(tidyverse)
```

Les outils de base de `dplyr` sont des tableaux `tibble`. Les fonctions de `dplyr` peuvent s'appliquer à des tableaux de données de `type data.frame` ou `tibble`, mais elles retournent systématiquement des tableaux au format `tibble`.

Les principales fonctions de `dplyr` à retenir sont :

- `slice` : fonction permettant de sélectionner des lignes en fonction de leurs indices,
- `filter` : fonction permettant de sélectionner des lignes en fonction de conditions,
- `select` : fonction permettant de sélectionner des colonnes,
- `rename` : fonction permettant de renommer les colonnes,
- `arrange` : fonction permettant de réarranger les données en fonction de l'ordre des valeurs d'une ou plusieurs colonnes,
- `mutate` : fonction permettant de créer des colonnes dans le jeu de données,
- `group_by` : fonction permettant de regrouper les données,
- `summarise` et `summarize_all` : fonctions permettant d'avoir un résumé statistique des données.

On peut combiner ces fonctions à l'aide de la commande `%>%` (s'appelant “pipe” en anglais), signifiant “tuyau” en anglais. Le pipe est pratique car il permet l'édition simple de fonctions compliquées (sans multiplier les paranthèses), et il permet d'éviter des boucles, donc de réduire du temps de calcul.

Pour une vue plus complète sur les possibilités de la librairie `dplyr`, voir ici (entre autre) :

https://chesneau.users.lmno.cnrs.fr/data.transformation.pdf

Exemple : Pour les premières utilisations, on va considérer le jeu de données `iris`.

On rappelle que `iris` comporte quatre caractères quantitatifs intitulés : `Sepal.Length`, `Sepal.Width`, `Petal.Length` et `Petal.Width`, et un caractère qualitatif (nominal) intitulé `Species`.

Pour un aperçu rapide des données, on peut demander l'entête du jeu de données `iris` :

```
library(dplyr)
library(datasets)
head(iris)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

Une description plus globale s'obtient en faisant :

```
str(iris)
```

Cela renvoie :

```
'data.frame': 150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

On retrouve les noms des colonnes, la nature des caractères considérés, le nombre d'observations, etc.

Le reste de la section est dédiée à des manipulations simples de ce jeu de données.

6.3.1 Commande `slice`

La fonction `slice` permet de sélectionner des lignes (correspondantes aux individus) en fonction de leurs indices.

Par exemple, à partir de `iris`, si l'on veut créer un jeu de données contenant les 10 premières lignes, on fait :

```
slice(iris, 1:10)
```

Cela renvoie :

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1           5.1         3.5          1.4         0.2  setosa
2           4.9         3.0          1.4         0.2  setosa
3           4.7         3.2          1.3         0.2  setosa
4           4.6         3.1          1.5         0.2  setosa
5           5.0         3.6          1.4         0.2  setosa
6           5.4         3.9          1.7         0.4  setosa
7           4.6         3.4          1.4         0.3  setosa
8           5.0         3.4          1.5         0.2  setosa
9           4.4         2.9          1.4         0.2  setosa
10          4.9         3.1          1.5         0.1  setosa
```

On peut mettre un nom à ce jeu de données en vue de l'utiliser à des fins statistiques, ou le manipuler par la suite. Dès lors, on fait :

```
datanew = slice(iris, 1:10)
datanew
```

Deux fonctions dérivées de `slice` peuvent avoir un certain intérêt : `slice_min` permet de sélectionner les lignes avec les valeurs les plus petites d'une colonne donnée, et `slice_max` permet de sélectionner les lignes avec les valeurs les plus grandes d'une colonne donnée. Avec cette dernière, on peut tenter d'identifier des valeurs anormalement élevées, entre autres.

Par exemple, à partir de `iris`, pour sélectionner les 4 lignes ayant les plus grandes valeurs de `Petal.Length`, on fait :

```
slice_max(iris, Petal.Length, n = 4)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          7.7         2.6          6.9         2.3 virginica
2          7.7         3.8          6.7         2.2 virginica
3          7.7         2.8          6.7         2.0 virginica
4          7.6         3.0          6.6         2.1 virginica
```

Plus intéressant dans le contexte des techniques d'enquêtes, la fonction `slice_sample` permet de sélectionner aléatoirement les lignes (individus plus les valeurs associées) d'un jeu de données. Par défaut, la sélection se fait sans remise. Pour activer l'option “avec remise”, on met `replace = TRUE`. Par exemple, à partir de `iris`, pour sélectionner au hasard 5 lignes sans remise, on fait :

```
slice_sample(iris, n = 5)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1          6.3         2.9          5.6         1.8  virginica
2          5.4         3.0          4.5         1.5 versicolor
3          4.7         3.2          1.3         0.2     setosa
4          7.9         3.8          6.4         2.0  virginica
5          5.4         3.4          1.7         0.2     setosa
```

Si l’on veut garder les lignes sélectionnées pendant toute l’étude, on fait `set.seed(123)` une ligne avant la commande avec le `slice_sample`.

6.3.2 Commande `filter`

La fonction `filter` permet de sélectionner des lignes en fonction de conditions.

Pour une première manipulation, on peut sélectionner les lignes dont `Species` affichant la modalité `versicolor`. On fait :

```
filter(iris, Species == "versicolor")
```

Cela renvoie :

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           7.0         3.2          4.7         1.4 versicolor
2           6.4         3.2          4.5         1.5 versicolor
3           6.9         3.1          4.9         1.5 versicolor
4           5.5         2.3          4.0         1.3 versicolor
5           6.5         2.8          4.6         1.5 versicolor
6           5.7         2.8          4.5         1.3 versicolor
7           6.3         3.3          4.7         1.6 versicolor
8           4.9         2.4          3.3         1.0 versicolor
9           6.6         2.9          4.6         1.3 versicolor
10          5.2         2.7          3.9         1.4 versicolor
11          5.0         2.0          3.5         1.0 versicolor
12          5.9         3.0          4.2         1.5 versicolor
13          6.0         2.2          4.0         1.0 versicolor
14          6.1         2.9          4.7         1.4 versicolor
15          5.6         2.9          3.6         1.3 versicolor
16          6.7         3.1          4.4         1.4 versicolor
17          5.6         3.0          4.5         1.5 versicolor
```

```
18          5.8         2.7          4.1         1.0 versicolor
19          6.2         2.2          4.5         1.5 versicolor
20          5.6         2.5          3.9         1.1 versicolor
21          5.9         3.2          4.8         1.8 versicolor
22          6.1         2.8          4.0         1.3 versicolor
23          6.3         2.5          4.9         1.5 versicolor
24          6.1         2.8          4.7         1.2 versicolor
25          6.4         2.9          4.3         1.3 versicolor
26          6.6         3.0          4.4         1.4 versicolor
27          6.8         2.8          4.8         1.4 versicolor
28          6.7         3.0          5.0         1.7 versicolor
29          6.0         2.9          4.5         1.5 versicolor
30          5.7         2.6          3.5         1.0 versicolor
31          5.5         2.4          3.8         1.1 versicolor
32          5.5         2.4          3.7         1.0 versicolor
33          5.8         2.7          3.9         1.2 versicolor
34          6.0         2.7          5.1         1.6 versicolor
35          5.4         3.0          4.5         1.5 versicolor
36          6.0         3.4          4.5         1.6 versicolor
37          6.7         3.1          4.7         1.5 versicolor
38          6.3         2.3          4.4         1.3 versicolor
39          5.6         3.0          4.1         1.3 versicolor
40          5.5         2.5          4.0         1.3 versicolor
41          5.5         2.6          4.4         1.2 versicolor
42          6.1         3.0          4.6         1.4 versicolor
43          5.8         2.6          4.0         1.2 versicolor
44          5.0         2.3          3.3         1.0 versicolor
45          5.6         2.7          4.2         1.3 versicolor
46          5.7         3.0          4.2         1.2 versicolor
47          5.7         2.9          4.2         1.3 versicolor
48          6.2         2.9          4.3         1.3 versicolor
49          5.1         2.5          3.0         1.1 versicolor
50          5.7         2.8          4.1         1.3 versicolor
```

On aurait aussi pu utiliser le pipe (mais cela rallonge un peu la taille des commandes), et faire :

```
iris %>%
  filter(Species == "versicolor")
```

Si l’on veut rajouter une condition, on peut utiliser les symboles “et”, “ou (inclusif)” et/ou “ou (exclusif)”,

représentées par `&`, `|` et `xor(,)`, respectivement.

Par exemple, pour sélectionner les lignes dont `Species` affiche la modalité `versicolor`, et `Petal.Length` est strictement supérieur à 4.5, on fait :

```
filter(iris, Species == "versicolor" &  Petal.Length > 4.5)
```

Cela renvoie :

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           7.0         3.2          4.7         1.4 versicolor
2           6.9         3.1          4.9         1.5 versicolor
3           6.5         2.8          4.6         1.5 versicolor
4           6.3         3.3          4.7         1.6 versicolor
5           6.6         2.9          4.6         1.3 versicolor
6           6.1         2.9          4.7         1.4 versicolor
7           5.9         3.2          4.8         1.8 versicolor
8           6.3         2.5          4.9         1.5 versicolor
9           6.1         2.8          4.7         1.2 versicolor
10          6.8         2.8          4.8         1.4 versicolor
11          6.7         3.0          5.0         1.7 versicolor
12          6.0         2.7          5.1         1.6 versicolor
13          6.7         3.1          4.7         1.5 versicolor
14          6.1         3.0          4.6         1.4 versicolor
```

On pourra également expérimenter les sélections suivantes :

```
filter(iris, Sepal.Length < 5.5 & Petal.Length > 4)
```

et

```
filter(iris, xor(Sepal.Length < 3, Petal.Length > 5))
```

et

```
filter(iris, Sepal.Length <= 3 | Petal.Length >= 5)
```

La commande `%in%` peut être utile pour sélectionner les lignes en fonction des caractères quantitatifs, pour éviter les commandes trop longues. Pour s’en convaincre, on peut faire :

```
filter(iris, Species == "versicolor" |  Species == "setosa")
```

et comparer avec :

```
filter(iris, Species %in% c("versicolor", "setosa"))
```

On obtient exactement la même chose.

On peut combiner les sélections avec `%>%` et `filter`.

Par exemple, pour sélectionner les lignes telles que `Species` affiche `versicolor` et `setosa`, et `Sepal.Length` strictement plus petit que 5, on fait :

```
filter(iris, Species %in% c("versicolor", "setosa")) %>%
  filter(Sepal.Length < 5)
```

Cela renvoie :

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1           4.9         3.0          1.4         0.2     setosa
2           4.7         3.2          1.3         0.2     setosa
3           4.6         3.1          1.5         0.2     setosa
4           4.6         3.4          1.4         0.3     setosa
5           4.4         2.9          1.4         0.2     setosa
6           4.9         3.1          1.5         0.1     setosa
7           4.8         3.4          1.6         0.2     setosa
8           4.8         3.0          1.4         0.1     setosa
9           4.3         3.0          1.1         0.1     setosa
10          4.6         3.6          1.0         0.2     setosa
11          4.8         3.4          1.9         0.2     setosa
12          4.7         3.2          1.6         0.2     setosa
13          4.8         3.1          1.6         0.2     setosa
14          4.9         3.1          1.5         0.2     setosa
15          4.9         3.6          1.4         0.1     setosa
16          4.4         3.0          1.3         0.2     setosa
17          4.5         2.3          1.3         0.3     setosa
18          4.4         3.2          1.3         0.2     setosa
19          4.8         3.0          1.4         0.3     setosa
20          4.6         3.2          1.4         0.2     setosa
21          4.9         2.4          3.3         1.0 versicolor
```

6.3.3 Commande `select`

La fonction `select` permet de sélectionner des colonnes (correspondantes aux caractères).

Par exemple, à partir de `iris`, pour créer un jeu de données composé des deux premières colonnes, on fait :

```
iris2c = select(iris, c(1,2))
head(iris2c)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width
1          5.1         3.5
2          4.9         3.0
3          4.7         3.2
4          4.6         3.1
5          5.0         3.6
6          5.4         3.9
```

Pour sélectionner les colonnes faisant référence à la taille, on fait :

```
iristaille = select(iris, Sepal.Length, Petal.Length)
head(iristaille)
```

Cela renvoie :

```
  Sepal.Length Petal.Length
1          5.1          1.4
2          4.9          1.4
3          4.7          1.3
4          4.6          1.5
5          5.0          1.4
6          5.4          1.7
```

On peut aussi supprimer les colonnes que l’on souhaite. Par exemple, pour créer un jeu de données composé de `iris` et privé de la colonne `Petal.Length`, on fait :

```
irissanspetallength = select(iris, -Petal.Length)
head(irissanspetallength)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Width Species
1          5.1         3.5         0.2  setosa
2          4.9         3.0         0.2  setosa
3          4.7         3.2         0.2  setosa
4          4.6         3.1         0.2  setosa
5          5.0         3.6         0.2  setosa
6          5.4         3.9         0.4  setosa
```

On peut aussi sélectionner un bloc de colonnes, avec des colonnes qui se suivent, uniquement avec les noms de la première colonne et dernière colonne du bloc qui nous intéressent. Par exemple, à partir de `iris`, pour créer un jeu de données contenant les colonnes allant de `Sepal.Length` à `Petal.Width`, on fait :

```
irisnames = select(iris, Sepal.Length : Petal.Width)
head(irisnames)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
```

On peut aussi faire une sélection des colonnes en fonction de mots récurrents dans leur noms avec les commandes `starts_with` et `ends_with`. Par exemple, à partir de `iris`, pour créer un jeu de données contenant les colonnes ayant `Petal` au début de leurs noms, on fait :

```
irispetal = select(iris, starts_with("Petal"))
head(irispetal)
```

Cela renvoie :

```
  Petal.Length Petal.Width
1          1.4         0.2
2          1.4         0.2
3          1.3         0.2
4          1.5         0.2
5          1.4         0.2
6          1.7         0.4
```

Ou bien, dans le même registre, pour créer un jeu de données contenant les colonnes ayant `Length` à la fin de leurs noms, on fait :

```
irislength = select(iris, ends_with("Length"))
head(irislength)
```

Cela renvoie :

```
  Sepal.Length Petal.Length
1          5.1          1.4
2          4.9          1.4
3          4.7          1.3
4          4.6          1.5
5          5.0          1.4
6          5.4          1.7
```

Également, on peut sélectionner les colonnes en fonction de la nature des caractères associés avec la commande `select_if` combinée avec `is.numeric` ou `is.factor`. Par exemple, à partir de `iris`, pour créer un jeu de données contenant que les colonnes associées au caractères quantitatifs, on fait :

```
irisquant = select_if(iris, is.numeric)
head(irisquant)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1         3.5          1.4         0.2
2          4.9         3.0          1.4         0.2
3          4.7         3.2          1.3         0.2
4          4.6         3.1          1.5         0.2
5          5.0         3.6          1.4         0.2
6          5.4         3.9          1.7         0.4
```

6.3.4 Commande `rename`

La fonction `rename` permet de renommer les colonnes (ou caractères) dans le jeu de données.

Par exemple, à partir de `iris`, si l'on veut renommer `Sepal.Length` par `SL`, on fait :

```
irisrename = rename(iris,  SL = Sepal.Length)
head(irisrename)
```

Cela renvoie :

```
   SL Sepal.Width Petal.Length Petal.Width Species
1 5.1         3.5          1.4         0.2  setosa
2 4.9         3.0          1.4         0.2  setosa
3 4.7         3.2          1.3         0.2  setosa
4 4.6         3.1          1.5         0.2  setosa
5 5.0         3.6          1.4         0.2  setosa
6 5.4         3.9          1.7         0.4  setosa
```

On constate que le changement de nom a bien été fait.

Si les noms des colonnes ont des espaces ou des caractères spéciaux, ceux-ci sont gérables en utilisant des guillemets.

6.3.5 Commande `arrange`

La fonction `arrange` permet de rearranger les données en fonction de l'ordre des valeurs d'une ou plusieurs colonnes.

Par exemple, si l'on veut réarranger les données de `iris` selon l'ordre croissant des valeurs de `Petal.Length`, on fait :

```
irisarrange = arrange(iris, Petal.Length)
head(irisarrange)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.6         3.6          1.0         0.2  setosa
2          4.3         3.0          1.1         0.1  setosa
3          5.8         4.0          1.2         0.2  setosa
4          5.0         3.2          1.2         0.2  setosa
5          4.7         3.2          1.3         0.2  setosa
6          5.4         3.9          1.3         0.4  setosa
```

Autrement, par exemple, si l'on veut réarranger les données de `iris` selon l'ordre décroissant des valeurs de `Sepal.Length`, on fait :

```
irisarrange2 = arrange(iris, desc(Sepal.Length))
head(irisarrange2)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          7.9         3.8          6.4         2.0 virginica
```

```
2          7.7         3.8          6.7         2.2 virginica
3          7.7         2.6          6.9         2.3 virginica
4          7.7         2.8          6.7         2.0 virginica
5          7.7         3.0          6.1         2.3 virginica
6          7.6         3.0          6.6         2.1 virginica
```

Si cela s’y prête, on peut trier selon plusieurs colonnes, les unes après les autres, suivant un ordre défini. Par exemple, on peut trier suivant un caractère qualitatif, et une fois fait, suivant un caractère quantitatif.

6.3.6 Commande `mutate`

La fonction `mutate` permet de créer des colonnes dans le jeu de données.

Par exemple, si l’on veut créer un nouveau jeu de données composé de `iris` avec une nouvelle colonne correspondante au carré des valeurs de `Sepal.Length`, on fait :

```
irisnewcol = mutate(iris, Sepcarre = Sepal.Length ** 2)
head(irisnewcol)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species Sepcarre
1          5.1         3.5          1.4         0.2  setosa    26.01
2          4.9         3.0          1.4         0.2  setosa    24.01
3          4.7         3.2          1.3         0.2  setosa    22.09
4          4.6         3.1          1.5         0.2  setosa    21.16
5          5.0         3.6          1.4         0.2  setosa    25.00
6          5.4         3.9          1.7         0.4  setosa    29.16
```

Une autre possibilité, moins élégante, mais donnant le même résultat est d’utiliser la fonction `bind_cols` qui permet de rajouter des colonnes. On fait :

```
S = tibble(Sepcarre = iris$Petal.Length ** 2)
irisnewcol2 = bind_cols(iris, S)
head(irisnewcol2)
```

En complément de `bind_cols`, `bind_rows` permet de rajouter des lignes.

6.3.7 Commande `group_by`

La fonction `group_by` permet de préciser les groupes d’individus (lignes) qui nous intéressent en fonction des modalités d’un caractère qualitatif, et de faire des analyses sur chacun de ces groupes

simultanément. Ainsi, `group_by` est souvent suivi du pipe, lequel mène à une certaine action. Une action particulièrement utile est une analyse statistique, notamment avec la commande `summarize` qui sera présentée par la suite.

Par exemple, si l'on veut créer un nouveau jeu de données composé de `iris` avec les quatre premières lignes associées aux modalités de `Species` (donc 12 lignes au total), on fait :

```
irisgroup = group_by(iris, Species) %>% slice(1:4)
head(irisgroup)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
         <dbl>       <dbl>        <dbl>       <dbl> <fct>
1          5.1         3.5          1.4         0.2 setosa
2          4.9         3            1.4         0.2 setosa
3          4.7         3.2          1.3         0.2 setosa
4          4.6         3.1          1.5         0.2 setosa
5          7           3.2          4.7         1.4 versicolor
6          6.4         3.2          4.5         1.5 versicolor
```

On peut également créer un nouveau jeu de données composé de `iris` avec les lignes associées aux modalités de `Species` ordonnées par ordre croissant selon le caractère `Petal.Length`. On fait :

```
irisgroupa = group_by(iris, Species) %>% arrange(Petal.Length)
head(irisgroupa)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
         <dbl>       <dbl>        <dbl>       <dbl> <fct>
1          4.6         3.6          1           0.2 setosa
2          4.3         3            1.1         0.1 setosa
3          5.8         4            1.2         0.2 setosa
4          5           3.2          1.2         0.2 setosa
5          4.7         3.2          1.3         0.2 setosa
6          5.4         3.9          1.3         0.4 setosa
```

6.3.8 Commandes `summarize` et `summarize_all`

La fonction `summarize` permet de faire des analyses statistiques ciblées, et la fonction `summarize_all` permet de faire des analyses statistiques globales.

Par exemple, à partir de `iris`, si l'on veut déterminer la moyenne des valeurs du caractère `Sepal.Length`, on fait :

```
summarize(iris, mean(Sepal.Length))
```

Cela renvoie :

```
  mean(Sepal.Length)
1           5.843333
```

Si l'on veut la moyenne de tous les caractères quantitatifs de `iris`, on fait :

```
irismean = select_if(iris, is.numeric) %>% summarize_all(mean)
head(irismean)
```

Cela renvoie :

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.843333    3.057333        3.758    1.199333
```

Les fonctions `summarize` et `summarize_all` peuvent être combinées avec `group_by`.

Par exemple, à partir de `iris`, si l'on veut déterminer les moyennes des valeurs des caractères quantitatifs selon les trois modalités du caractère qualitatif `Species`, on fait :

```
irisgroupanalyse = group_by(iris, Species) %>% summarise_all(mean)
head(irisgroupanalyse)
```

Cela renvoie :

```
  Species    Sepal.Length Sepal.Width Petal.Length Petal.Width
  <fct>             <dbl>       <dbl>        <dbl>       <dbl>
1 setosa             5.01        3.43         1.46       0.246
2 versicolor         5.94        2.77         4.26       1.33
3 virginica          6.59        2.97         5.55       2.03
```

En plus des moyennes, on pourrait s'intéresser aux minimums avec `min`, aux maximum avec `max`, aux médianes avec `median`, et aux écart-types avec `sd`, entre autres.

En complément, on peut aussi présenter la fonction `count`. Elle consiste à compter le nombre de fois que les modalités d'un caractère qualitatif apparaissent dans le jeu de données.

Par exemple, à partir de `iris`, si l'on veut compter le nombre de fois où les trois modalités du caractère qualitatif `Species` apparaissent (bien qu'on le sache déjà), on fait :

```
count(iris, Species)
```

Cela renvoie :

```
     Species  n
1     setosa 50
2 versicolor 50
3  virginica 50
```

6.4 Package `forcats`

Tout d'abord, `forcats` est l'anagramme de “factors”, et ainsi nommé, il fournit une suite d'outils utiles qui résolvent les problèmes courants liés aux modalités (ou facteurs) des caractères qualitatifs.

Les principales fonctions de `forcats` à retenir sont :

- `fct_recode` : fonction permettant de recoder les modalités,
- `fct_rev` : fonction permettant d'inverser l'ordre des modalités,
- `fct_relevel` : fonction permettant de spécifier le rang d'une ou plusieurs modalités spécifiques,
- `fct_inorder` : fonction permettant de ranger les modalités selon leur ordre d'apparition dans les données,
- `fct_infreq` : fonction permettant de ranger les modalités selon leur ordre (décroissant) de fréquence,
- `fct_lump` : fonction permettant d'agréger les modalités moins fréquentes en une seule modalité.

Pour les premières utilisations, on va considérer le jeu de données `airquality`.

On rappelle que `airquality` comporte 6 caractères, dont le caractère `Month` codé par des entiers correspondants aux mois de l'année (il est donc qualitatif).

Pour un aperçu rapide des données, on peut demander l'entête du jeu de données `airquality` :

```
library(dplyr)
library(forcats)
% ou, plus global: install.packages("tidyverse")
% library(tidyverse)
library(datasets)
head(airquality)
```

Cela renvoie :

```
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

Une description plus globale s’obtient en faisant :

```
str(airquality)
```

Cela renvoie :

```
'data.frame': 153 obs. of  6 variables:
 $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
 $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
 $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
 $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
 $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
 $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

On retrouve les noms des colonnes, la nature des caractères considérés, le nombre d’observations, etc.

Le reste de la section est dédiée au traitement du caractère `Month`. En effet, il est considéré comme un caractère quantitatif (entier) mais en fait, c’est un caractère qualitatif et les chiffres associés codes les mois de l’année.

Dans un premier temps, on le précise comme étant un caractère qualitatif en faisant :

```
airquality$Month = factor(airquality$Month)


str(airquality$Month)
```

Cela renvoie :

```
Factor w/ 5 levels "5","6","7","8",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Dès lors, la librairie `forcats` va nous permettre de manipuler ce caractère.

6.4.1 Commandes `fct_recode`

La fonction `fct_recode` permet de recoder les modalités d’un caractère qualitatif comme on le souhaite.

Par exemple, si l’on veut recoder les chiffres de `Month` avec les mois de l’année associés, on fait :

```
airquality$Month = fct_recode(airquality$Month, Mai = '5', Juin = '6', Juillet = '7',
Aout = '8', Septembre = '9')


str(airquality$Month)
```

Cela renvoie :

```
 Factor w/ 5 levels "Mai","Juin","Juillet",..: 1 1 1 1 1 1 1 1 1 1 ...
```

On a donc réussit le recodage.

6.4.2 Commandes `fct_rev`

La fonction `fct_rev` permet d'inverser l'ordre des modalités.

Par exemple, si l'on veut inverser l'ordre des modalités du (nouveau) caractère `Month`, on fait :

```
airquality$Month = fct_rev(airquality$Month)
```

```
str(airquality$Month)
```

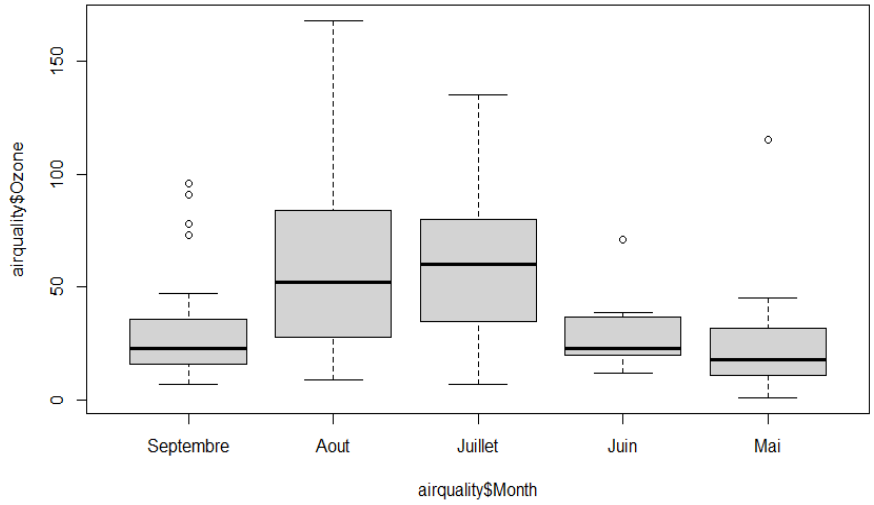
Cela renvoie :

```
 Factor w/ 5 levels "Septembre","Aout",..: 5 5 5 5 5 5 5 5 5 5 ...
```

Cela impacte surtout sur les graphiques. Par exemple, si l'on fait :

```
boxplot(airquality$Ozone ~ airquality$Month)
```

Cela renvoie :



Commençant par `Septembre`, puis `Aout`, etc., l'ordre des modalités prises en compte est donc bien inversé.

6.4.3 Commandes `fct_relevel`

La fonction `fct_relevel` permet de spécifier le rang d'une ou plusieurs modalités spécifiques.

Par exemple, si l'on veut positionner les modalités du (nouveau) caractère `Month` suivant un ordre précis, on fait :

```
airquality$Month = fct_relevel(airquality$Month, "Juin", "Septembre", "Juillet",
```

```
 "Mai", "Aout")

levels(airquality$Month)
```

Cela renvoie :

```
[1] "Juin"      "Septembre" "Juillet"   "Mai"       "Aout"
```

6.4.4 Commandes `fct_inorder`

La fonction `fct_inorder` permet de ranger les modalités selon leur ordre d'apparition dans les données.

Pour `airqyuality`, cela n'a pas d'intérêt car l'ordre d'apparition est déjà considéré.

6.4.5 Commandes `fct_infreq`

La fonction `fct_infreq` permet de ranger les modalités selon leur ordre (décroissant) de fréquence.

Par exemple, si l'on veut ranger les modalités selon leur ordre (décroissant) de fréquence de `Month`, on fait :

```
airquality$Month = fct_infreq(airquality$Month)

levels(airquality$Month)
```

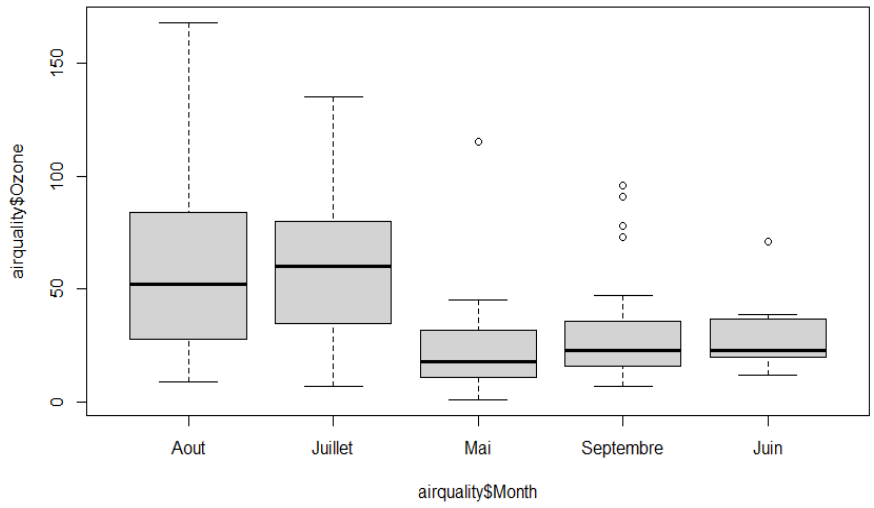
Cela renvoie :

```
[1] "Aout"      "Juillet"   "Mai"       "Septembre" "Juin"
```

Cela impacte surtout sur les graphiques. Par exemple, si l'on fait :

```
boxplot(airquality$Ozone ~ airquality$Month)
```

Cela renvoie :



6.4.6 Commandes `fct_lump`

La fonction `fct_lump` permet d’agréger les modalités moins fréquentes en une seule modalité (“lump” signifie grumeau en français). Cette fonction comporte plusieurs options à moduler.

Par exemple, si l’on veut ne garder que les 3 modalités principales de `Month` (les 3 plus fréquentes), on fait :

```
airquality$Month = fct_lump(airquality$Month, n = 3)


levels(airquality$Month)
```

Cela renvoie :

```
[1] "Aout"    "Juillet" "Mai"     "Other"
```

Le `"Other"` contient `Juin` et `Septembre`, qui s’avèrent être les moins courantes. On peut vérifier que cela s’est bien appliqué sur les données de `airquality`. Par exemple, pour afficher les dernières lignes de `airquality`, on fait :

```
tail(airquality)
```

Cela renvoie :

```
    Ozone Solar.R Wind Temp Month Day
148    14      20 16.6   63 Other  25
149    30     193  6.9   70 Other  26
```

```
150    NA     145 13.2   77 Other  27
151    14     191 14.3   75 Other  28
152    18     131  8.0   76 Other  29
153    20     223 11.5   68 Other  30
```

Le `Other` est bien présent.

7 Publication : le rapport

7.1 Outils informatiques

Les deux outils informatiques les plus utilisés pour écrire un rapport “digne de ce nom” sont Word et LaTex. Pour un résultat le plus professionnel possible, on privilégiera LaTex, surtout si le rapport s'annonce long et/ou s'il y a des formules mathématiques. Ce document a été écrit avec LaTex (avec l'interface TeXworks).

7.2 Les premières pages

La première page porte le titre, le ou les auteurs, la date, le lieu et les commanditaires de l'étude. Éventuellement, une page est dédiée aux remerciements de collaborateurs (ou proches). Une autre page concerne le résumé de l'étude, et met l'accent sur les points importants. Une autre page donne un sommaire détaillé du rapport.

7.3 L'introduction

Ce chapitre donne les raisons pour lesquelles l'enquête a été menée. On y détaille quelle problématique a été suivie, en quoi l'étude se positionne par rapport aux autres travaux qui ont déjà été faits, quels sont ces autres travaux, etc. Une partie plus théorique pourra définir les termes clés de l'étude. Les questions spécifiques de recherche et les hypothèses à tester sont ici énoncées.

7.4 Les données et la méthode

On définit

- les caractéristiques de l'enquête : plan, population soumise à l'étude, échantillonnage (taille), mode de collecte, etc.,
- le contenu du questionnaire (organisation, nombre et justification des questions, etc.), et des exemples de questions posées (le questionnaire complet se trouve en annexe),
- éventuellement, les caractéristiques démographiques des répondants,
- les méthodes et logiciels utilisés en vue de l'analyse (caractéristiques des indices élaborés, présentation des méthodes d'analyse plus complexe (ACP, ACM,...).

7.5 Les résultats

Ils sont décrits dans différents chapitres. Selon les objectifs de l'enquête, on étudie l'effet des principales variables explicatives prises successivement ou l'on peut prendre comme point de départ les

questions ou les hypothèses de recherche. On commence souvent par des tris à plats, puis on fait des tris croisés et enfin, des analyses multivariées. Il faut se montrer prudent dans l'exposé et l'interprétation des résultats, surtout si l'échantillon n'est pas représentatif : on évite des générations abusives (généraliser les résultats obtenus dans le cadre de l'université de Caen à l'ensemble des universités). Les résultats sont illustrés par des tableaux numériques et des graphiques qu'il faut mettre aux endroits appropriés.

7.6 Conclusion

La dernière partie décrit ce que les données révèlent en les situant par rapport aux autres études. Noter les résultats inattendus, les recommandations, les perspectives d'études futures, etc.

7.7 Bibliographie

La bibliographie doit contenir tous les documents consultés (articles, polycopiés, etc.), y compris les sites internet et autres ressources.

7.8 Les annexes

On y trouve :

- le questionnaire entièrement reproduit,
- une fiche résumant les caractéristiques de l'enquête,
- les informations techniques sur la méthodologie statistique,
- le dictionnaire des codes,
- la description des réponses aux questions ouvertes,
- éventuellement, un glossaire.

8 Projets

Projet 1 : Suivi des étudiants des Masters de l'université de Caen-Normandie. Proposer un projet d'analyse pour étudier le devenir des étudiants issus des Masters de l'université de Caen-Normandie. Vous présenterez les enjeux de l'étude, les différentes hypothèses testées, la méthode d'échantillonnage choisi ainsi que la méthode de recueil de l'information. Vous rédigerez le questionnaire. Une bibliographie devra accompagner votre travail (il n'est pas demandé d'administrer votre questionnaire, ni de l'analyser).

Projet 2 : Synthèse autour du Big Data. À partir des articles de François Robin, « Utilisation de Google Trends dans les enquêtes mensuelles sur le Commerce de détail de la Banque de France » et de Pete Richardson, « L'apport des big data pour les prévisions macroéconomiques à court terme et en temps réel » issus du numéro d'Économie et Statistique 505-506 de 2018 (1ère partie), vous présenterez de manière structurée les apports et les limites de l'utilisation des big data. Vous veillerez à ne pas paraphraser les textes.

Projet 3 : Le secret statistique dans les enquêtes socio-économiques. À partir des différentes sources de votre choix (recensement de la population, sirène, clap, ...) que vous trouverez sur le site Insee.fr, vous rappellerez l'intérêt du secret statistique dans les enquêtes (et notamment les divergences entre producteurs et utilisateurs de données), expliquerez à partir d'exemples précis sa mise en application pour deux sources de votre choix. Vous exposerez ensuite un programme R qui permet de gérer pour une source que vous choisirez les questions de secret statistique.

Projet 4 : La gestion des données manquantes. À partir des différentes sources de votre choix (recensement de la population, sirène, clap, ...) que vous trouverez sur le moteur de recherche google, vous exposerez les différentes méthodes mathématiques existantes permettant la gestions des données manquantes. Des applications avec l'utilisation du logiciel R et des librairies adaptées sont demandées.

Projet 5 : Activités, emploi et chômage en 2019 et en séries longues. À partir des données que vous récupérez sur Insee.fr, `https://www.insee.fr/fr/statistiques/4498680?sommaire=4498692&q=enqu%C3%Aate#documentation`,

vous présenterez la source utilisée (enquête emploi en continu), puis zoomerez sur les formes particulières d'emplois par sexe et âges regroupés entre 1982 et 2019. Vous présenterez une analyse structurée en identifiant les principaux messages et les grandes tendances.

Projet 6 : Ressenti des salariés de France métropolitaine suite à la mise en place du télétravail en période de Covid 19. Le télétravail a été généralisé dans la plupart des entreprises françaises. Après avoir rappelé le contexte juridique autour du télétravail, vous préparez un questionnaire auprès d'un échantillon de 1000 personnes pour connaître les principales difficultés que les salariés ont rencontrées durant cette période. Vous justifierez les principales hypothèses testées et le mode d'administration du questionnaire. Une bibliographie est attendue.