

- Résumé
- 1 Certaines méthodes sont meilleures que d'autres
- 2 Randomisation par bloc : vous pouvez vous assurer que les groupes de traitement et de contrôle sont équilibrés
- 3 Designs factoriels : vous pouvez randomiser plusieurs traitements en même temps sans perdre de puissance statistique
- 4 Vous pouvez randomiser des grappes entières ensemble (mais plus vos grappes sont grandes, plus votre puissance statistique est faible !)
- 5 Vous pouvez randomiser d'une manière qui permet de voir plus facilement s'il y a des effets de débordement
- 6 Différentes unités peuvent être assignées au traitement avec différentes probabilités
- 7 Randomisation restreinte : si vous n'aimez pas ce que vous obtenez, vous pouvez recommencer
- 8 Écrivez un code de randomisation qui vous permet de simuler de nombreuses randomisations possibles
- 9 Vous pouvez randomiser au fur et à mesure
- 10 La randomisation peut parfois être une manière éthique d'assigner un traitement, mais parfois ce n'est pas le cas.

Résumé

Ce guide vous aidera à concevoir et à exécuter différents types de randomisation dans vos expériences. Nous nous concentrons sur les grandes idées et fournissons des exemples et des outils que vous pouvez utiliser dans R. Pour savoir pourquoi faire de la randomisation, voir ce guide des méthodes (<http://egap.org/resource/10-strategies-for-figuring-out-if-x-caused-y>).

1 Certaines méthodes sont meilleures que d'autres

Il existe de nombreuses façons de randomiser. Le plus simple est de lancer une pièce à chaque fois que vous voulez déterminer si un sujet donné reçoit un traitement ou non. Cela garantit que chaque sujet a une probabilité de 0,5 de recevoir le traitement et une probabilité de 0,5 de ne pas le recevoir. De cette façon, le fait qu'un sujet reçoive le traitement n'affecte en rien le fait que le sujet suivant reçoive le traitement, chaque sujet a une chance égale de recevoir le traitement, et le traitement ne sera pas corrélé avec tous les facteurs de confusion - du moins en espérance.

Ce n'est pas une mauvaise approche, mais elle présente des lacunes. Premièrement, en utilisant cette méthode, vous ne pouvez pas savoir à l'avance combien d'unités seront sous traitement et combien sous contrôle. Si vous voulez le savoir à l'avance, vous avez besoin d'un moyen de faire des sélections afin que les différents tirages ne soient pas statistiquement indépendants les uns des autres (comme tirer des noms d'un chapeau). Deuxièmement, vous voudrez peut-être avoir le contrôle sur la part exacte des unités assignées au traitement et au contrôle. C'est difficile à faire avec une pièce de monnaie. Troisièmement, vous voudrez peut-être pouvoir reproduire votre randomisation pour montrer qu'il n'y avait pas d'erreurs. C'est difficile à faire avec des pièces de monnaie ou des chapeaux. Enfin, comme nous le montrons ci-dessous, il existe toutes sortes de façons de faire de la randomisation pour améliorer la puissance statistique et assurer l'équilibre. C'est plus difficile en utilisant des pièces de monnaie ou des chapeaux.

Heureusement cependant, une randomisation répliquable et flexible est possible avec des logiciels disponibles gratuitement. Le code suivant en R peut, par exemple, être utilisé pour générer une assignation aléatoire, spécifiant le nombre d'unités à traiter. Ici, N (100) est le nombre d'unités dont

vous disposez et m (34) est le nombre que vous souhaitez traiter. La “graine” permet de reproduire le même tirage à chaque fois que vous exécutez le code (ou vous pouvez changer la graine pour un tirage différent).^{1 2}

```
library(randomizr)
# définir une "graine" aléatoire pour rendre cela reproductible
set.seed(343)
complete_ra(100, 34)
```

2 Randomisation par bloc : vous pouvez vous assurer que les groupes de traitement et de contrôle sont équilibrés

Il est possible, en randomisant, de spécifier la proportion pour des facteurs particuliers qui vous intéressent entre les groupes de traitement et de contrôle, même s’il n’est pas possible de spécifier quelles unités particulières sont sélectionnées pour l’un ou l’autre groupe et de maintenir l’assignation aléatoire.

Cela signifie qu’il est possible de spécifier, par exemple, que vos groupes de traitement et de contrôle contiennent des proportions égales d’hommes et de femmes. En d’autres termes, cela évite toute randomisation qui pourrait produire un groupe de traitement distinctement masculin et un groupe de contrôle distinctement féminin, ou vice-versa.

Pourquoi est-ce souhaitable ? Non pas parce que notre estimation de l’effet moyen du traitement serait autrement biaisée, mais parce qu’elle pourrait produire beaucoup de bruit. Supposons qu’une assignation aléatoire ait généré un groupe de traitement très masculin et un groupe de contrôle très féminin. Nous observerions une corrélation entre le genre et le statut de traitement. Si nous devions estimer un effet de traitement, cet effet de traitement serait toujours non biaisé car le genre n’a pas causé le statut de traitement. Cependant, il serait plus difficile de rejeter l’hypothèse nulle selon laquelle ce n’était pas notre traitement mais le genre qui produisait l’effet. Bref, le déséquilibre produit une estimation bruitée, ce qui rend plus difficile pour nous d’avoir confiance en nos estimations.

La randomisation par bloc (parfois appelée stratifiée) aide à truquer notre expérience de manière à ce que nos groupes de traitement et de contrôle soient équilibrés selon des dimensions importantes, mais toujours assignés de manière aléatoire. Essentiellement, ce type de design de randomisation construit de multiples mini-expériences : par exemple, il pourrait prendre des femmes et assigner de manière aléatoire la moitié au traitement et l’autre moitié au contrôle, puis il assignerait la moitié des hommes au traitement et l’autre moitié au contrôle. Cela garantit un équilibre entre les genres lorsque les groupes de traitement et de contrôle sont mis en commun.

Le package `blockTools` (<https://cran.r-project.org/web/packages/blockTools/index.html>) est un package utile pour effectuer la randomisation par bloc. Commençons par générer un faux ensemble de données pour 60 sujets, dont 36 hommes et 24 femmes.

Supposons que nous voudrions découper par bloc selon le genre. Sur la base de nos données, `blockTools` générera les blocs les plus petits possibles, chaque bloc contenant deux unités du même sexe, dont l’une sera assignée au traitement et l’autre au contrôle.

```
rm(list = ls())

library(blockTools)
library(dplyr)
library(randomizr)

dat <-
  tibble(
    id = seq(1:60),
    female = c(rep(0, 36), rep(1, 24)),
    age = sample(18:65, size = 60, replace = TRUE)
  )

# un bloc de covariables

dat <-
  dat %>%
  mutate(Z_block_1 = block_ra(female))

with(dat, table(female, Z_block_1))

# groupes assortis de 4
out <- block(dat, n.tr = 4, id.vars = "id",
             block.vars = c("female", "age"))

dat <-
  dat %>%
  mutate(
    block_id = createBlockIDs(out, dat, id.var = "id"),
    Z_block_2 = block_ra(block_id))

with(dat, table(Z_block_2, block_id))
```

Vous pouvez vérifier la moyenne de la variable sur laquelle vous avez découpé en bloc pour voir que les groupes de traitement et de contrôle sont en fait parfaitement équilibrés selon le sexe.

3 Designs factoriels : vous pouvez randomiser plusieurs traitements en même temps sans perdre de puissance statistique

Supposons qu'il existe plusieurs composants d'un traitement que vous souhaitez tester. Par exemple, vous voudrez peut-être évaluer l'impact d'un programme de microfinance. Deux traitements spécifiques pourraient consister à prêter de l'argent aux femmes et à leur fournir une formation. Un design factoriel examine toutes les combinaisons possibles de ces traitements : (1) Prêts, (2) Formation, (3) Prêts + Formation et (4) Contrôle. Les sujets sont ensuite assignés de manière aléatoire à l'une de ces quatre conditions.

Table 1: Factorial design

	Loans	No loans
Training	Loans + Training	Training only
No training	Loans only	Pure control group

Les designs factoriels sont particulièrement utiles lors de l'évaluation d'interventions qui incluent un ensemble de traitements. Comme dans l'exemple ci-dessus, de nombreuses interventions de développement comportent plusieurs bras de traitement, et il est parfois difficile de dire quels bras produisent l'effet observé. Un design factoriel sépare ces différents traitements et permet également de voir l'interaction entre eux.

Le code suivant vous montre comment randomiser pour un design factoriel.

```
dat <-
  tibble(
    Z_loan = complete_ra(80, 40),
    Z_training = block_ra(blocks = Z_loan)
  )

with(dat, table(Z_loan, Z_training))
```

4 Vous pouvez randomiser des grappes entières ensemble (mais plus vos grappes sont grandes, plus votre puissance statistique est faible !)

Parfois, il est impossible de randomiser au niveau de l'individu. Par exemple, un appel à la radio pour amener des individus à un bureau de vote doit par nature être diffusé à l'ensemble d'un marché médiatique ; il est impossible de diffuser uniquement à certains individus mais pas à d'autres. Que ce soit par nécessité ou par choix, vous allez parfois randomiser des grappes (clusters) au lieu d'individus.

L'inconvénient de la randomisation par grappe est qu'elle réduit votre puissance statistique, puisque le nombre d'unités assignées de manière aléatoire reflète désormais le nombre de grappes et non pas simplement votre nombre total de sujets. Si vous aviez deux grappes assignées de manière aléatoire de 1 000 individus chacune, le nombre fonctionnel d'unités pourrait être plus proche de 2, et non de 2 000. Pour cette raison, il est préférable de faire des grappes aussi petites que possible.

De même, il est également souhaitable d'avoir une hétérogénéité au sein de vos grappes afin qu'elles soient aussi représentatives que possible de votre population plus large. Si les individus au sein d'une grappe donnée sont très similaires les uns aux autres, ils peuvent avoir des résultats potentiels similaires, et cette grappe avec des résultats potentiels similaires sera assignée au traitement ou au contrôle en tant que grappe. Dans l'ensemble, cela augmentera votre variance si cette grappe avait des résultats potentiels particulièrement élevés ou faibles, car cela augmente la corrélation globale entre les résultats potentiels et l'assignation au traitement. En bref, si vos grappes sont plus représentatives de la population plus large, vos estimations de l'effet moyen du traitement seront plus précises.

Une question fréquemment posée est de savoir en quoi la randomisation par grappe (cluster) diffère de la randomisation par bloc. La randomisation par bloc est effectuée afin d'atteindre un équilibre basé sur les covariables de pré-traitement. Par exemple, une intervention éducative peut bloquer la randomisation sur les résultats des tests de l'année précédente afin de suivre les progrès des élèves les moins performants et les plus performants. La randomisation par grappe se produit lorsque plusieurs unités sont traitées comme un groupe : elles reçoivent ensemble le même statut de traitement ou de contrôle. Par exemple, la même intervention éducative peut être randomisée au niveau de la classe, de sorte que les salles de classe constituent les grappes. Il est possible de randomiser par bloc et par grappe simultanément. Dans notre exemple, vous pouvez calculer la note moyenne au test pour chaque classe et effectuer une randomisation par bloc en fonction de la note moyenne de la classe.

Le graphique suivant montre à quoi pourraient ressembler vos données dans les cas de randomisation par bloc, par grappe et par bloc + grappe, par rapport à un cas simple de randomisation sans bloc ni grappe. Dans les deux cas où le découpage par grappe se produit, vous voyez que l'assignation du traitement (représentée par la couleur) apparaît en petits groupes. Dans les deux cas où le découpage par bloc se produit, il y a une distribution uniforme des couleurs dans les quatre quadrants du tracé, les blocs de cette assignation aléatoire.

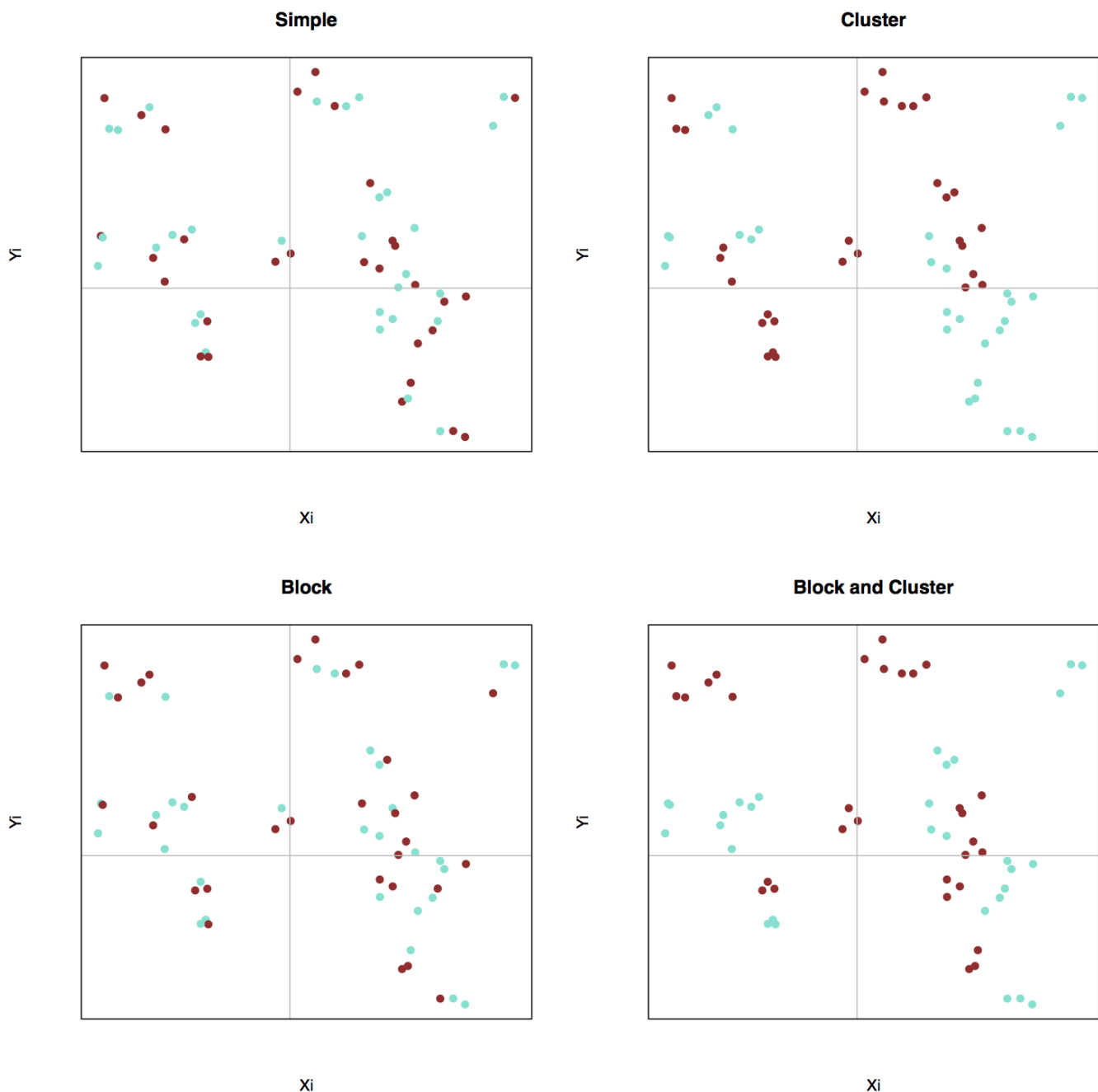


Illustration des patterns pour les unités de traitement et de contrôle que vous pourriez voir sous différents types de designs par bloc et par grappe.

5 Vous pouvez randomiser d'une manière qui permet de voir plus facilement s'il y a des effets de débordement

Lors du design de votre expérience, demandez-vous de manière critique si les “effets de débordement” constituent une menace pour votre capacité à identifier l'effet causal de votre traitement. Des débordements surviennent si le résultat d'une unité est affecté par le statut de traitement d'une autre unité. Cela peut être délicat si les unités ont la capacité d'interagir les unes avec les autres : un membre d'un village peut apprendre qu'un autre membre de son village a reçu une subvention en espèces et peut changer son comportement en conséquence.

Une façon de rendre les effets de débordement plus évidents consiste à utiliser la double randomisation. Vous assignez d'abord de manière aléatoire certains groupes au traitement et d'autres au contrôle, et au sein des groupes, vous assignez certains individus au traitement et d'autres au contrôle. La comparaison des individus témoins de votre grappe de traitement avec les individus de votre grappe de contrôle vous permettra d'évaluer le rôle des effets de débordement dans votre expérience.

6 Différentes unités peuvent être assignées au traitement avec différentes probabilités

Parfois, les gens pensent que “aléatoire” signifie que deux événements sont également probables, mais en fait, l'assignation aléatoire est “aléatoire” tant que la probabilité d'assignation au traitement est strictement comprise entre 0 et 1. Si un sujet a 0 ou 100 % de chances d'être assigné au traitement, ce sujet doit être exclu de votre analyse expérimentale car il n'y a pas de randomisation. Cependant, tous les sujets ayant une probabilité d'assignation au traitement strictement comprise entre 0 et 1 peuvent être inclus, même si leurs probabilités diffèrent, pour autant que leurs probabilités soient connues.

Pourquoi voudriez-vous attribuer différentes probabilités d'assignation au traitement ? Supposons que vous travailliez avec un partenaire pour randomiser l'assignation des observateurs électoraux afin de mesurer l'effet sur la fraude électorale. Votre partenaire pour la mise en œuvre ne peut se permettre financièrement d'envoyer beaucoup d'observateurs électoraux dans une partie rurale ou éloignée du pays. Vous pouvez résoudre cette contrainte en découpant par bloc géographique et en attribuant une probabilité plus élevée d'assignation au traitement à des villages plus proches vers lesquels il est moins coûteux de se déplacer. Tant que la probabilité d'assignation au traitement pour les villages les plus accessibles est inférieure à 1, la probabilité d'assignation au traitement pour les villages moins accessibles est supérieure à zéro, et que ces probabilités sont connues, il est possible d'estimer l'effet du traitement.

Cependant, lorsque les sujets ont des probabilités différentes d'assignation au traitement, vous ne pouvez plus simplement fusionner tous les sujets dans l'analyse de vos données. Si vous le faites, l'assignation du traitement sera corrélée aux caractéristiques de base sur lesquelles vous avez découpé par bloc. Il y a deux façons de gérer cela.

La première consiste à estimer l'effet moyen du traitement bloc par bloc, puis à faire la moyenne des effets de traitement, pondérée par la taille du bloc par rapport à l'ensemble de l'échantillon.

La deuxième méthode est la pondération de probabilité inverse (inverse probability weighting, IPW). Pour l'IPW, les poids sont définis comme $1/p$ pour les unités traitées et $1/(1-p)$ pour les unités de contrôle, où p fait référence à la probabilité d'assignation de traitement. Cette méthode vous permet d'exécuter une régression pondérée de Y sur l'assignation de traitement.

```
N <- 100000
dat <- tibble(Y0 = 1:N,
              Y1 = Y0 + 10000)

dat <-
  dat %>%
  mutate(
    p = seq(0.25, 0.75, length.out = N),
    Z = simple_ra(N, prob_unit = p),
    Y = Z * Y1 + (1 - Z) * Y0)

dat %>%
  summarise(
    naive_estimate = mean(Y[Z == 1]) - mean(Y[Z == 0]),
    ipw_estimate = weighted.mean(Y[Z == 1], 1 / p[Z == 1]) -
      weighted.mean(Y[Z == 0], 1 / (1 - p[Z == 0]))
  )
```

7 Randomisation restreinte : si vous n'aimez pas ce que vous obtenez, vous pouvez recommencer

Cela peut sembler incompatible avec l'idée même de la randomisation de ne pas choisir une assignation aléatoire parce que vous n'aimez pas ce qui est choisi. Mais parfois, cela a du sens. Parfois, vous voudrez peut-être vous assurer que la randomisation ne produit pas de schémas particuliers (par exemple, trop de personnes qui se connaissent sont traitées). Mais les schémas qui vous intéressent peuvent être trop difficiles à mettre en place en avance. Ce que vous pouvez ensuite faire, c'est faire un tirage au sort et voir ensuite si le tirage répond aux critères qui vous intéressent ou non, si ce n'est pas le cas, refaites le tirage. Soyez averti, cependant, que si vous faites cela, vous créez quelques complications : (1) chaque unité ne sera pas nécessairement assignée avec la même probabilité, (2) les unités peuvent ne pas être assignées indépendamment au traitement. Vous devez prendre en compte ces deux faits dans votre analyse, par exemple, en générant des poids de probabilité inverse comme nous l'avons fait au point 6, mais en utilisant le même code de randomisation restreinte pour déterminer la probabilité que chaque sujet soit assigné au traitement sous ces restrictions. Ensuite, vous utilisez la distribution des assignations de traitement possibles pour implémenter l'inférence de randomisation. Ces analyses étant complexes, procédez avec prudence.

8 Écrivez un code de randomisation qui vous permet de simuler de nombreuses randomisations possibles

Un avantage de R pour randomiser est que vous pouvez effectuer des milliers de randomisations possibles en quelques secondes. Pourquoi est-ce bénéfique ?

1. Cela peut être utile pour vérifier si votre code de randomisation a fonctionné. Par exemple, si un ou plusieurs sujets de votre expérience n'ont jamais reçu de traitement avec plus de 10000 assignations aléatoires possibles, vous suspecterez alors une faille dans votre code de randomisation.
2. Vous pouvez utiliser la re-randomisation pour calculer la probabilité exacte d'assignation au traitement pour chaque individu de votre expérience. Ceci est particulièrement utile si votre code de randomisation est plus complexe. Peut-être que vous employez à la fois la randomisation par bloc et par grappe, ce qui entraîne des probabilités d'assignation au traitement très différentes pour les individus dans une grande expérience. Ces probabilités seraient difficiles à calculer à la main, mais une solution simple consiste à exécuter votre code de randomisation de multiples fois et à générer une variable représentant la proportion de fois où chaque individu a été assigné au traitement : cela représente sa probabilité individuelle d'assignation au traitement. Cette variable peut ensuite être utilisée dans une régression pondérée lors du calcul de l'effet moyen du traitement.
3. La simulation des randomisations possibles est une approche pour calculer le niveau de signification statistique. Cette approche, appelée inférence de randomisation, génère une p-valeur exacte en calculant les effets moyens du traitement possibles qui seraient observés dans le cadre d'assignations aléatoires hypothétiques si le traitement n'avait aucun effet. La p-valeur est alors la proportion des effets estimés du traitement qui est au moins aussi importante que celle observée par votre expérience. Cette approche est préférable aux calculs standard de signification statistique lorsque la distribution de vos données n'est pas normale, car l'inférence de randomisation évite de faire des hypothèses de distribution et utilise à la place la distribution des données observées dans votre expérience. Pour plus d'informations sur l'inférence de randomisation, y compris des exemples de code, voir 10 choses à savoir sur l'inférence de randomisation (<https://egap.org/resource/10-things-to-know-about-randomization-inference/>).

9 Vous pouvez randomiser au fur et à mesure

Dans de nombreuses expériences, vous ne connaissez peut-être pas l'intégralité de votre échantillon au début de l'expérience ; certains sujets peuvent se joindre à l'expérience au fil du temps. Cela présente une complication lorsque nous voulons utiliser un algorithme par bloc simple car l'ajout de sujets à notre pool peut modifier la composition de nos blocs et donc leurs probabilités d'assignation de traitement.

Pour maintenir les blocs et donc le contrôle sur l'équilibre entre les groupes de traitement et de contrôle, vous pouvez utiliser des covariables pour calculer la similarité d'un nouveau sujet avec d'autres sujets précédemment assignés et assigner le nouveau sujet à la condition de traitement avec le moins d'unités similaires.^{3 4}

10 La randomisation peut parfois être une manière éthique d'assigner un traitement, mais parfois ce n'est pas le cas.

La randomisation est l'ingrédient clé pour isoler l'effet causal d'un traitement du point de vue du design de recherche, mais il est également important de tenir compte des implications éthiques de la randomisation. Lorsque nous pensons aux effets à long terme d'une expérience, la randomisation nous permet de tester quels programmes sont les plus efficaces afin que les ressources puissent être dirigées

vers les programmes qui font le plus de différences dans la vie future des populations. A court terme, la randomisation de l'accès à un programme (par opposition à une distribution basée sur des caractéristiques arbitraires) peut être une manière particulièrement éthique de distribuer des biens rares qui ne peuvent pas être étendus à tout le monde.

Cependant, parfois, ce sont les populations les plus démunies qui ont besoin d'être servies par une intervention dans une expérience. Un design aléatoire qui offre des prêts à un nombre égal de participants à faible revenu et à revenu élevé soustrait en partie ces ressources aux personnes dans le besoin. Si nous pensons que le prêt a des effets bénéfiques, cela soulève des inquiétudes quant à l'éthique de ce détournement d'allocation de ressources envers les plus nécessiteux.⁵ Il faudrait déployer un argumentaire solide en faveur des avantages sociaux de la recherche et également tendre vers des designs qui offrent finalement aussi les avantages au groupe de contrôle.

Un design de randomisation avec liste d'attente est une façon de traiter tous les sujets tout en permettant au chercheur de tester expérimentalement l'efficacité du traitement. Dans ce design, on peut déployer l'intervention par phases et assigner de manière aléatoire les unités à la phase dans laquelle elles seront traitées. Par exemple, pour un total de 90 villages, on pourrait traiter 30 villages chaque année et mesurer les résultats à la fin de chaque année. Si vous vouliez comparer les résultats dans les villages de traitement et de contrôle, vous compareriez les 30 villages traités aux 60 villages pas encore traités à la fin de la première année. A la fin de la deuxième année, vous pourriez comparer les 30 villages qui ont été traités l'année précédente aux 30 villages qui ne le sont pas encore. Essentiellement, cela crée deux expériences, identiques sauf pour l'année qui les sépare. Dans le tableau ci-dessous, vous pouvez voir qu'au cours de la première année, nous avons pu comparer le groupe de traitement bleu foncé aux deux groupes de contrôle bleu clair. Au cours de la deuxième année, nous pouvons comparer le groupe de traitement rouge foncé au groupe de contrôle rouge clair, mais nous voulons éviter de regrouper les deux groupes de traitement car l'un a été traité plus longtemps que l'autre. Vous pouvez voir qu'après la troisième année, plus aucune comparaison ne peut être faite car toutes les unités ont été traitées.

Table 2: Wait list randomization design

Year	Villages 1-30	Villages 31-60	Villages 61-90
1	T	C	C
2	T	T	C
3	T	T	T

La seule exigence est que l'assignation d'un sujet au traitement dans une phase particulière soit assignée de manière aléatoire et sans rapport avec ses résultats potentiels d'intérêt. Un design dans lequel des participants plus enthousiastes recevraient un traitement plus tôt violerait cette hypothèse et ne donnerait pas une estimation non biaisée de l'effet du traitement, car des facteurs non observés qui les prédisposent à rechercher un traitement peuvent influencer leurs résultats potentiels. Le design avec liste d'attente est un exemple de design de randomisation créatif qui pourrait répondre aux préoccupations éthiques concernant une distribution limitée d'un traitement précieux.

L'éthique est souvent étroitement liée aux designs randomisés, en particulier dans les sciences sociales et la recherche médicale. En tant que chercheur, vous devez examiner attentivement les implications possibles de la randomisation d'un traitement donné. Vous devrez également solliciter l'approbation de votre design de recherche auprès du comité d'éthique pour la recherche de votre établissement.

1. Les générateurs de nombres aléatoires sont en fait pseudo-aléatoires car ils génèrent un vecteur de nombres aléatoires basé sur un petit ensemble de valeurs initiales, appelé état de départ. Les générateurs de nombres aléatoires fonctionnent de cette manière afin d'améliorer la vitesse de calcul. Cependant, la série de nombres aléatoires générés est aussi aléatoire que nécessaire pour une assignation aléatoire, car elle n'a aucun rapport avec les résultats potentiels de vos sujets.↵
2. Le code a été mis à jour par Alex Coppock le 25 novembre 2020.↵
3. Voir Moore, Ryan T. et Sally A. Moore. "Blocking for sequential political experiments." *Political Analysis* 21.4 (2013): 507-523.↵
4. Pour une présentation plus détaillée des procédures de randomisation disponibles dans le package R randomizr, voir https://declaredesign.org/r/randomizr/articles/randomizr_vignette.html (https://declaredesign.org/r/randomizr/articles/randomizr_vignette.html)↵
5. Mais si nous sommes certains des effets du prêt, alors pourquoi mener une expérience pour le tester ? Dans la recherche médicale, les essais contrôlés randomisés s'arrêtent souvent s'il devient clair dès le début qu'un médicament guérit sans aucun doute des maladies potentiellement mortelles, et donc le refuser aux sujets témoins est dangereux. (De même, un essai s'arrêterait également s'il était clair dès le début qu'un médicament provoque sans aucun doute des effets négatifs et nocifs.)↵