Code

Pratique AMMS : analyse de puissance pour les conceptions en cluster

Bob Verity

August 04, 2022

Dépendances pour Pratique

Veuillez copier et coller le morceau de code ci-dessous dans son intégralité sur votre console pour télécharger les bibliothèques de packages R nécessaires à cette pratique. Si vous rencontrez des difficultés pour installer l'un des packages R, veuillez demander à un instructeur un lecteur flash préchargé.

Hide

```
if (!("tidyverse" %in% installed.packages())) {
  install.packages("tidyverse")
}
```

Chargez maintenant toutes ces bibliothèques dans cette session en utilisant le morceau de code ci-dessous. Veuillez le copier-coller dans son intégralité.

Hide

```
library(tidyverse)
```

Enfin, sourcez les fonctions supplémentaires nécessaires à ce TP en copiant-collant cette fonction :

Hide

```
source("source_functions/power2_utils.R")
```

Introduction aux sondages groupés

Il existe de nombreuses situations où il est logique d'échantillonner en grappes, plutôt que d'échantillonner des individus. Certaines raisons incluent:

- Il est logistiquement impossible d'échantillonner au niveau individuel. Cela est vrai dans les enquêtes au niveau provincial ou national, où nous ne pouvons pas échantillonner au hasard des individus de toute la population car ils pourraient être situés n'importe où, ce qui serait un cauchemar logistique !
- La population se regroupe naturellement en grappes. Un exemple serait l'échantillonnage du paludisme dans les cliniques, qui concentrent les cas de paludisme de toute la zone desservie, ce qui rend la surveillance à ce niveau très efficace.
- Les interventions sont planifiées au niveau du cluster plutôt qu'au niveau individuel. Si une intervention sera réalisée de manière groupée, il peut être judicieux de collecter des données de référence de la même manière.

Lors de l'utilisation d'une conception en grappes, nous devons être conscients que les observations auront tendance à être plus similaires au sein des grappes qu'entre les grappes - appelées **corrélation intra-grappe** ou **corrélation intra-grappe**. Par exemple, en règle générale, les individus d'un même village auront tendance à se ressembler davantage en termes de comportements, de caractéristiques physiques et de facteurs de risque que les individus échantillonnés au hasard dans la province élargie. Dans les maladies infectieuses, nous avons une autre raison pour la corrélation intra-cluster ; transmission de la maladie. Cela peut conduire à des épidémies locales qui provoquent un grand nombre de résultats corrélés dans un seul cluster.

Heureusement, il existe des méthodes statistiques bien définies pour traiter la corrélation intra-cluster. En utilisant ces méthodes, nous pouvons estimer la force de cette corrélation, puis en tenir compte dans nos tests statistiques.

Aperçu des données

Pour cette pratique, nous travaillerons entièrement avec des ensembles de données inventés ou simulés. Celles-ci nous permettront de nous familiariser avec les concepts de base des enquêtes par grappes, que nous pourrons ensuite appliquer à des ensembles de données du monde réel à un stade ultérieur.

Objectifs pratiques

À la fin de cet exercice pratique, vous devriez être en mesure de :

- Identifier les données surdispersées

- Estimer un coefficient de corrélation intra-cluster (ICC) et un design effect
- Construire un intervalle de confiance tenant compte de la corrélation intra-cluster
- Tester une différence entre une prévalence échantillonnée en grappes et un seuil arbitraire (par exemple dans la conception de l'étude *pfhrp2/3*)
- Effectuer une analyse de puissance et un calcul de la taille de l'échantillon pour les enquêtes en grappes

Augmentation de l'incertitude due au clustering

Corrélation intra-cluster et surdispersion

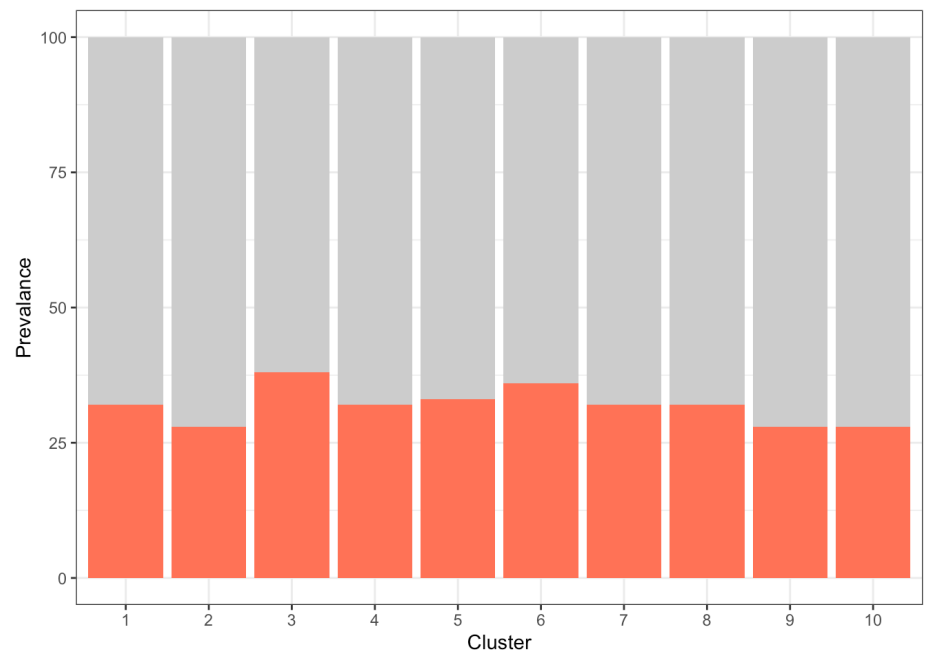
Avant d'entrer dans les idées concernant la corrélation intra-cluster, il convient d'explorer ce que nous attendrions de voir s'il n'y avait * aucune * corrélation intra-cluster. Le code suivant simule la prévalence observée dans une série de grappes, où chaque individu de chaque grappe a la même probabilité "prev" d'être positif pour le paludisme. Il produit ensuite un diagramme à barres simple de la prévalence observée.

Hide

```
# set simulation parameters
n_clusters <- 10    # number of clusters
n_samp <- 100       # number of samples per cluster
prev <- 0.3         # true prevalence

# simulate some data by drawing from binomial distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = rbinom(n_clusters, n_samp, prev) / n_samp)

# produce a simple barplot of prevalence
barplot_clusters(cluster_prev)
```



Q1. Essayez d'exécuter ce code de simulation plusieurs fois pour avoir une idée de la distribution de la prévalence entre les clusters. Avez-vous déjà vu un cluster avec une prévalence de 0 % ? Avez-vous déjà vu un cluster avec une prévalence supérieure à 50 % ?

Click For Answer

A1. Pour une taille d'échantillon de `n_samp = 100`, il serait très peu probable (bien que pas impossible) que vous voyiez un jour une prévalence de 0 % ou supérieure à 50 %. Les valeurs ont tendance à rester dans une bande assez étroite autour de la prévalence réelle, ne variant que d'environ 10 % dans chaque direction.

Dans cette situation, chaque individu a la même probabilité d'être positif, quel que soit le cluster dans lequel il se trouve. Cela signifie que nous pouvons regrouper les résultats sur les clusters sans rien perdre. Dans l'exemple ci-dessus, nous avons le même nombre d'échantillons dans chaque cluster, ce qui signifie que le nombre total d'échantillons est donné par `n_clusters * n_samp`. Grâce à l'analyse de puissance précédente, nous savons comment utiliser cette valeur pour construire un intervalle de confiance (IC) à 95 % sur notre estimation de la prévalence. Le code suivant effectue la même simulation, mais construit maintenant un IC à 95 % plutôt que de tracer les résultats. Jetez un coup d'œil aux mathématiques et assurez-vous de bien comprendre comment cet intervalle est construit.

Hide

```
# set simulation parameters
n_clusters <- 10    # number of clusters
n_samp <- 100       # number of samples per cluster
prev <- 0.3         # true prevalence

# simulate some data by drawing from binomial distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = rbinom(n_clusters, n_samp, prev) / n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error using n_samp*n_clusters as our total sample size
SE <- sqrt(p_bar*(1 - p_bar) / (n_samp*n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.2599192 0.3160808
```

Q2. Essayez d'exécuter ce code de simulation plusieurs fois. À quelle fréquence la prévalence réelle ("prev") se situe-t-elle dans votre IC à 95 % ?

Click For Answer

A2. Vous devriez constater que l'IC à 95 % contient la prévalence réelle la plupart du temps. Cependant, si vous exécutez le code suffisamment de fois, vous devriez pouvoir trouver une simulation où la prévalence réelle se situe en dehors de cet intervalle. Ainsi, même s'il est peu probable qu'il se trompe, ce n'est pas impossible.

Donc, s'il n'y a pas de corrélation intra-cluster et que tous les individus ont la même probabilité d'être positifs quel que soit leur cluster, alors notre analyse des données est assez simple. Pour estimer la prévalence sur tous les clusters (par exemple sur l'ensemble de la région géographique), nous regroupons d'abord les résultats, puis nous estimons la prévalence comme la proportion de cas positifs. Notre IC à 95 % peut être construit à l'aide de la formule standard ci-dessus.

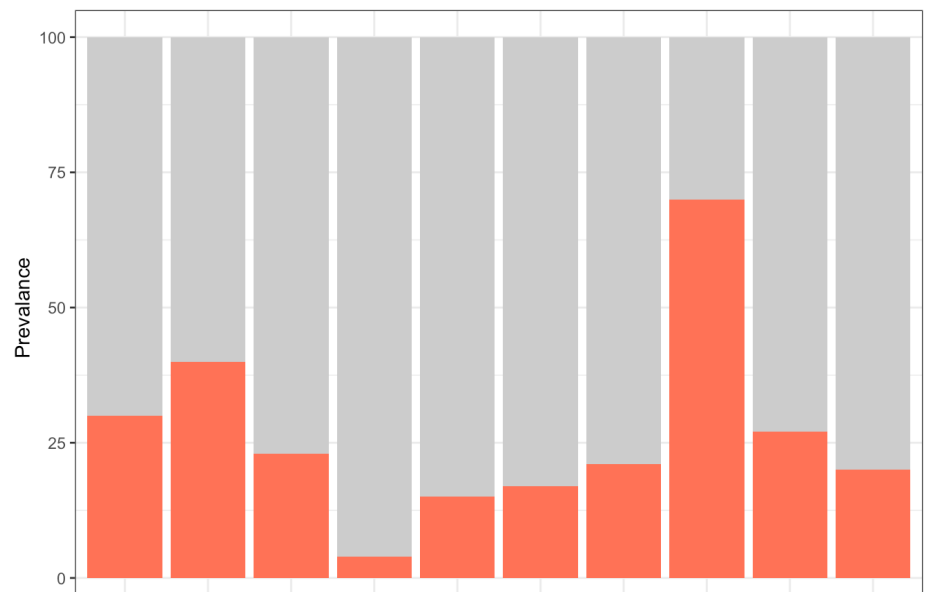
Mais que se passe-t-il si nos données sont **surdispersées**, ce qui signifie que la propagation est plus grande que ce à quoi nous nous attendrions si les probabilités étaient les mêmes partout ? Le code suivant simule les données d'une distribution surdispersée. La quantité de surdispersion est contrôlée par le coefficient de corrélation intra-cluster (ICC) qui varie entre 0 et 1. Plus l'ICC est grand, plus la propagation est importante.

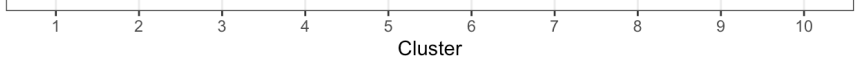
Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.2

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# produce a simple barplot of prevalence
barplot_clusters(cluster_prev)
```





Q3. Comme précédemment, essayez d'exécuter ce code de simulation plusieurs fois pour avoir une idée de la distribution de la prévalence entre les clusters. Avez-vous déjà vu un cluster avec une prévalence de 0 % ? Avez-vous déjà vu un cluster avec une prévalence supérieure à 50 % ? Que se passe-t-il lorsque vous utilisez un ICC plus grand ?

Click For Answer

A3. Contrairement au cas indépendant, il est maintenant assez courant de voir une prévalence de 0 % ou supérieure à 50 % dans certaines grappes. Alors que la prévalence moyenne globale est toujours la même, la propagation est beaucoup plus grande. Plus l'ICC est grand, plus la dispersion entre les clusters est grande.

Dans ce qui précède, nous avons vu le lien entre la *surdispersion* et la *corrélation intra-cluster*. S'il existe une corrélation intra-cluster, cela entraînera une surdispersion des observations. De même, s'il y a une surdispersion, il doit y avoir une certaine corrélation intra-cluster. Les deux idées sont étroitement liées et nous en parlerons indifféremment.

Si nous ne connaissions pas les problèmes de l'échantillonnage en grappes, nous pourrions être tentés de regrouper ces résultats et d'estimer la prévalence exactement comme nous l'avons fait dans l'exemple précédent. Ceci est implémenté dans le code suivant :

Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.5

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error using n_samp*n_clusters as our total sample size
SE <- sqrt(p_bar*(1 - p_bar) / (n_samp*n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.291073 0.348927
```

Q4. Essayez d'exécuter ce code de simulation plusieurs fois. À quelle fréquence la véritable prévalence se situe-t-elle dans votre IC à 95 % ?

Click For Answer

A4. Cette fois, vous devriez trouver assez facile de générer des simulations où la prévalence réelle se situe en dehors de l'IC à 95 %. Cela nous indique que notre IC est cassé - il ne fait pas ce que nous voulons parce que nos hypothèses ne sont pas valides. Les observations ne sont pas indépendantes sur les grappes et nous ne devrions donc pas regrouper les résultats de cette manière.

Ainsi, pour des données surdispersées, nous pouvons obtenir des résultats trompeurs si nous ne prenons pas en compte le regroupement. Et si nous abordions ce problème d'une manière complètement différente ? Nous pourrions ignorer le fait que nous avons des tailles d'échantillon pour chacune de nos estimations de prévalence et les traiter plutôt comme de simples valeurs distinctes. Nous pouvons ensuite calculer l'erreur standard et un IC à 95 % comme suit :

Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.5

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error from variance over clusters
SE <- sqrt(var(cluster_prev$p) / (n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.05798201 0.52001799
```

Q5. Essayez d'exécuter ce code de simulation plusieurs fois. À quelle fréquence la véritable prévalence se situe-t-elle dans votre IC à 95 % ?

Click For Answer

A5. Cet IC fonctionne beaucoup mieux. Nous devrions maintenant constater que la véritable prévalence se situe dans cet intervalle la plupart du temps.

Ainsi, une façon simple de traiter la corrélation intra-grappe est d'abandonner l'idée de regrouper les échantillons sur les grappes et de traiter à la place chaque grappe comme une observation unique. Nous pouvons ensuite calculer un IC à 95 % pour la prévalence au niveau de l'étude sur la base de ce nombre beaucoup plus petit de valeurs. Le CI ainsi produit est robuste à la surdispersion, et ne risque pas de donner une fausse confiance dans nos conclusions, mais cela se fait au prix d'une puissance réduite. Nous allons explorer cela ci-dessous.

Effets de conception

L'**effet de conception** est une façon de quantifier l'effet du regroupement sur notre analyse. Cela mesure à quel point la variance de notre estimation de la prévalence est supérieure à ce à quoi nous nous attendrions dans le cadre d'un échantillonnage aléatoire simple (EAS). En d'autres termes, il quantifie la mesure dans laquelle la conception de notre étude influence notre précision, d'où le nom *effet de conception*. Plus l'effet de conception est important, plus nous nous éloignons de ce à quoi nous nous attendrions sous SRS (une valeur de 1 indique qu'il n'y a pas de différence avec SRS).

Pour les enquêtes en grappes, l'effet de conception peut être défini comme :

$$D_{\text{eff}} = \frac{\text{Var}_{\text{clust}}(\bar{p})}{\text{Var}_{\text{SRS}}(\,barrep)}$$

où $\bar{p}$ est notre estimation de la prévalence globale, $\text{Var}_{\text{clust}}(\bar{p})$ est la variance réelle de cet estimateur en tenant compte du clustering, et $\text{Var}_{\text{SRS}}(\bar{p})$ est la variance à laquelle on s'attendrait sous un échantillonnage aléatoire simple. Ces deux variances sont simplement le carré des deux différents types d'erreur standard calculés ci-dessus - un au niveau de la grappe et un calculé en regroupant les résultats. Nous pouvons écrire:

$$\text{Var}_{\text{clust}}(\bar{p}) = \frac{s_c^2}{c-1}$$

où s_c^2 est la variance sur les clusters et c est le nombre de clusters. Pour le cas SRS, on peut écrire :

$$\text{Var}_{\text{SRS}}(\bar{p}) = \frac{\bar{p}(1-\bar{p})}{nc-1}$$

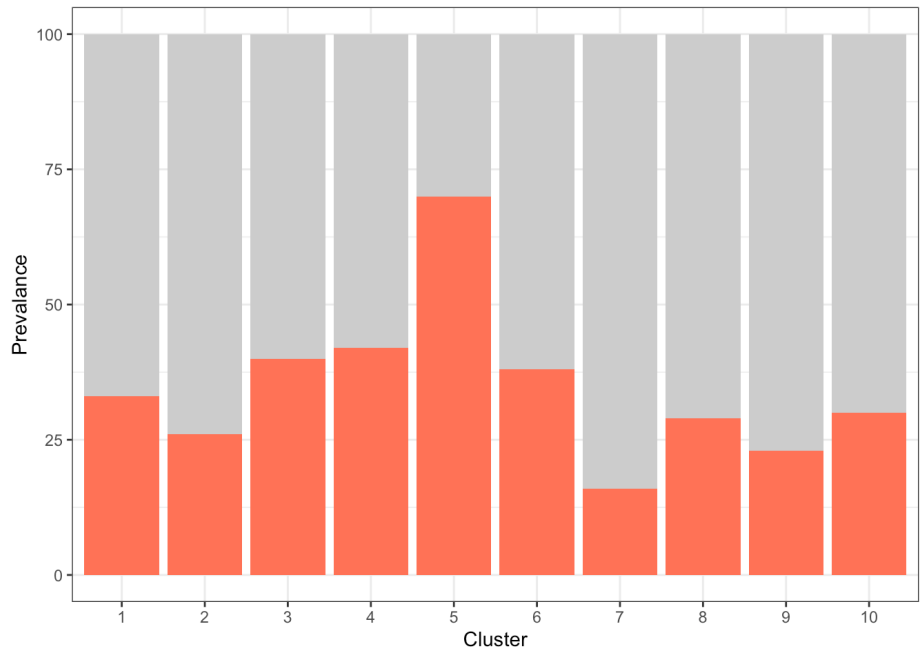
où n est le nombre d'échantillons par grappe. Une fois que nous avons ces deux valeurs, nous pouvons les combiner pour produire une estimation de l'effet de conception.

Chargeons quelques données sur lesquelles travailler :

Hide

```
# load overdispersed data from file
load("data/prev_overdispersed.RData")

# produce a simple barplot of prevalence
barplot_clusters(prev_overdispersed)
```



Hide

```
prev_overdispersed
```

```
##    cluster n_samp    p
## 1        1    100 0.33
## 2        2    100 0.26
## 3        3    100 0.40
## 4        4    100 0.42
## 5        5    100 0.70
## 6        6    100 0.38
## 7        7    100 0.16
## 8        8    100 0.29
## 9        9    100 0.23
## 10      10    100 0.30
```

Ici, nous avons des données de 10 grappes, ainsi que la même taille d'échantillon dans chaque grappe. En regardant le barplot et en utilisant notre intuition sur ce à quoi ressemblent les données sous des probabilités indépendantes, nous pouvons déjà soupçonner qu'il y a une surdispersion ici.

Tout d'abord, nous calculons la variance au niveau du cluster :

Hide

```
# calculate actual variance at cluster level
n_clusters <- nrow(prev_overdispersed)
var_clust <- var(prev_overdispersed$p) / (n_clusters - 1)
var_clust
```

```
## [1] 0.002417407
```

Deuxièmement, nous calculons la variance à laquelle nous nous attendrions sous SRS :

Hide

```
# calculate variance we would expect under SRS
p_bar <- mean(prev_overdispersed$p)
n_samp_total <- sum(prev_overdispersed$n_samp)
var_srs <- p_bar*(1 - p_bar) / (n_samp_total - 1)
var_srs
```

```
## [1] 0.0002268178
```

Enfin, nous pouvons estimer l'effet de conception comme une variance par rapport à l'autre.

Q6. Complétez le code suivant pour estimer l'effet de conception :

Hide

```
# estimate design effect
Deff <- # TO COMPLETE
Deff
```

Click For Answer

A6.

Hide

```
# estimate design effect
Deff <- var_clust / var_srs
Deff
```

```
## [1] 10.65793
```

L'effet de conception est juste supérieur à 10. Cela nous indique que la variance de notre estimation de la prévalence est 10 fois plus élevée que ce à quoi nous nous attendrions dans le cadre d'un échantillonnage aléatoire simple.

Une façon utile de penser à ce résultat est en termes de *taille d'échantillon effective*. C'est la taille de l'échantillon dont nous *aurions* besoin pour générer des données comme les nôtres si SRS était effectivement vrai. La taille effective de l'échantillon, n_{eff} est obtenue en divisant la taille réelle de l'échantillon par l'effet de conception :

Hide

```
# calculate effective sample size
n_samp_eff <- n_samp / Deff
n_samp_eff
```

```
## [1] 9.382689
```

Ici, nous constatons que la taille effective de l'échantillon se situe quelque part entre 9 et 10, même si la taille réelle de l'échantillon était de 100 par grappe. Cela fournit une façon assez intuitive de penser à la corrélation intra-cluster : même si nous avons échantillonné 100 individus d'un cluster, ils ont tendance à se ressembler beaucoup, donc c'est vraiment comme si nous n'avions échantillonné que 9 ou 10 personnes.

Comparer la prévalence à un seuil

Nous passons maintenant à la motivation principale de cette pratique - la conception d'une étude de suppression *pfhrp2/3*. En 2020, l'OMS a publié un *protocole principal pour la surveillance des délétions pfhrp2/3 et la biobanque pour soutenir la recherche future* (https://apps.who.int/iris/handle/10665/331197). Ce document est un excellent outil de référence pour quiconque envisage de mener une étude sur la prévalence de *pfhrp2/3*, et couvre tout, des techniques de laboratoire à l'élaboration d'un plan d'analyse. Il suggère une conception d'enquête en grappes avec les caractéristiques suivantes :

- L'étude est menée au niveau de la province. Plusieurs cliniques (suggérées au moins 10) sont recrutées dans la province, ce qui en fait une **conception d'enquête en grappes**.
- Les cas suspects de paludisme se présentant dans les cliniques sont testés par un test de diagnostic rapide (TDR) basé sur HRP2 et une autre méthode, par ex. microscopie. Les résultats discordants sont traités comme des TDR présumés faussement négatifs, qui sont ensuite suivis par séquençage du gène pour déterminer si la délétion du gène *pfhrp2/3* est présente.
- Le résultat principal est la prévalence des délétions *pfhrp2/3* dans chaque cluster. Celles-ci doivent ensuite être résumées par grappes pour produire une estimation de la prévalence globale des délétions au niveau de la province.
- Nous nous intéressons particulièrement à savoir si la prévalence est **au-dessus ou au-dessous du seuil de 5 %**. Si c'est le cas, il est conseillé au pays d'envisager de passer à un TDR non basé sur HRP2 pour réduire le risque de manquer des épisodes cliniques.

Ici, nous apprendrons une approche pour analyser ce type de données et comment concevoir une étude en tenant compte des considérations de puissance et de logistique.

Tests statistiques

Commençons par charger des données à partir du fichier :

Hide

```
load("data/cluster_onegroup.RData")

cluster_onegroup
```

```
##    cluster n_samp    p
## 1        1    100 0.23
## 2        2    100 0.07
## 3        3    100 0.00
## 4        4    100 0.05
## 5        5    100 0.29
## 6        6    100 0.06
## 7        7    100 0.03
## 8        8    100 0.19
## 9        9    100 0.00
## 10      10    100 0.06
```

Cet ensemble de données simulées est destiné à représenter les résultats d'une étude de suppression *pfhrp2/3*. Pour chaque cluster, nous avons le nombre d'échantillons testés et la prévalence des délétions au niveau du cluster.

Comme dans les exemples précédents, il est utile de réfléchir à ce qui se passerait si nous devions ignorer le clustering et simplement regrouper nos résultats. On obtient l'IC à 95% suivant sur la prévalence des délétions :

Hide

```
# pool results over clusters
n_samp_total <- sum(cluster_onegroup$n_samp)
n_samp_pos <- sum(cluster_onegroup$n_samp * cluster_onegroup$p)

# estimate prevalence
p_bar <- n_samp_pos / n_samp_total

# calcualte standard error
SE <- sqrt(p_bar*(1 - p_bar) / n_samp_total)

p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.07957225 0.11642775
```

Nous obtenons une estimation assez précise de la prévalence, avec notre IC allant d'environ 8% à 12%. Et si on comparait cela au seuil de 5% ? Le test approprié ici est le test z à un échantillon, qui peut être implémenté dans R à l'aide de la fonction `prop.test()` :

Hide

```
# carry out 1-proportion z-test
prop.test(n_samp_pos, n_samp_total, p = 0.05)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  n_samp_pos out of n_samp_total, null probability 0.05
## X-squared = 47.5, df = 1, p-value = 5.5e-12
## alternative hypothesis: true p is not equal to 0.05
## 95 percent confidence interval:
##  0.08062551 0.11853423
## sample estimates:
##     p
## 0.098
```

Q7. Ce résultat est-il significatif ? Comment le pays devrait-il réagir sur la base de ces résultats ?

Click For Answer

A7. Ce résultat est hautement significatif, avec une valeur de p inférieure à 1 sur un million. Sur la base de ce résultat, nous conclurions que la prévalence était supérieure au niveau de 5 % et que le pays devrait donc envisager de changer de TDR.

Mais, cette analyse groupée ignore la nature groupée de la conception. Tout comme dans les exemples précédents, nous devons tenir compte de la corrélation intra-cluster, sinon nous risquons de tirer de mauvaises conclusions. En utilisant l'IC à 95 % au niveau du cluster, nous obtenons ce qui suit :

Hide

```
# estimate prevalence
n_clusters <- nrow(cluster_onegroup)
p_bar <- mean(cluster_onegroup$p)

# calcualte standard error
SE <- sqrt(var(cluster_onegroup$p) / (n_clusters - 1))

p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.031744 0.164256
```

Q8. L'IC à 95 % s'étend-il sur 5 % ? Qu'est-ce que cela suggère quant à la possibilité que la prévalence soit nettement supérieure à cette valeur ?

Click For Answer

A8. Oui, l'IC à 95 % s'étend sur 5 %. Cela suggère qu'un test de signification par rapport à ce seuil ne sera pas significatif, car nous ne pouvons pas exclure la possibilité que la prévalence réelle soit aussi faible ou même inférieure.

Le test statistique approprié lors de la comparaison avec un seuil est le test t à un échantillon. Cela peut être implémenté dans R comme suit :

Hide

```
# one-sample t-test against 5% threshold
t.test(x = cluster_onegroup$p, mu = 0.05)
```

```
##
##  One Sample t-test
##
## data:  cluster_onegroup$p
## t = 1.4968, df = 9, p-value = 0.1687
## alternative hypothesis: true mean is not equal to 0.05
## 95 percent confidence interval:
##  0.02545405 0.17054595
## sample estimates:
## mean of x
##     0.098
```

Q9. Ce résultat est-il significatif ? Comment le pays devrait-il réagir sur la base de ces résultats ?

Click For Answer

A9. Cette fois, la valeur de p n'est pas significative au niveau de signification de 5 %. Par conséquent, nous n'avons pas suffisamment de preuves pour rejeter l'hypothèse nulle selon laquelle la prévalence est égale au seuil. Sur la base de ce résultat, le pays n'envisagerait pas de changer de TDR.

Nous avons ici un exemple où nos conclusions générales, et l'action qu'un pays prend en réponse aux suppressions de *pfhrp2/3*, sont différentes selon notre méthode d'analyse. Si nous regroupons de manière incorrecte les résultats sans tenir compte de la corrélation intra-cluster, nous pouvons finir par faire une recommandation sur l'utilisation des TDR à l'échelle nationale qui n'est pas bien prise en charge. Cela souligne l'importance de comprendre le regroupement dans l'analyse des données d'enquête.

Analyse de puissance et calcul de la taille de l'échantillon

Comment pouvons-nous alimenter une étude de suppression *pfhrp2/3*, compte tenu de ce que nous savons sur le clustering ? Nous commençons par examiner la valeur théorique de notre statistique de test pour le test t à un échantillon :

$$t = \frac{p-\mu}{\sqrt{\frac{\sigma^2}{c}}}$$

où p est la prévalence réelle, σ_c^2 est la variance entre les grappes et c est le nombre de grappes. Mais rappelez-vous que l'effet de conception est défini comme la variance entre les grappes divisée par la variance à laquelle nous nous attendrions sous SRS. Ainsi, en réorganisant cela, nous pouvons exprimer la variance entre les clusters comme suit :

$$\sigma_c^2 = D_{\text{eff}} \frac{p(1-p)}{n}$$

Si nous avions une prévalence supposée p, une taille d'échantillon n par cluster et un effet de conception supposé D_{eff}, nous pourrions alors déterminer la variance appropriée au niveau du cluster à utiliser dans notre t -statistique.

Q10. Si la prévalence réelle des délétions est de 10 %, et en supposant 100 échantillons par cluster et un effet de conception de 1,5, à quoi vous attendriez-vous en termes de variance entre les clusters ?

Click For Answer

A10. Nous nous attendrions à une variance de 0,00135 entre les clusters :

Hide

```
1.5 * 0.1*(1 - 0.1)/100
```

```
## [1] 0.00135
```

Parfois, il est plus facile de travailler en termes de coefficient de corrélation intra-cluster (ICC) plutôt qu'en termes d'effet de conception. L'ICC reçoit souvent le symbole mathématique ρ et est lié à l'effet de conception par la formule :

$$D_{\text{eff}} = 1 + (n-1)\rho$$

Cela conduit à la formule alternative suivante pour la variance entre les clusters :

$$\sigma_c^2 = (1 + (n-1)\rho)\frac{p(1-p)}{n}$$

Q11. Si la prévalence réelle des délétions était de 10 % et en supposant 100 échantillons par cluster et un ICC de 0,2, à quoi vous attendriez-vous en termes de variance entre les clusters ?

Click For Answer

A11. Nous nous attendrions à une variance de 0,01872 entre les clusters :

Hide

```
(1 + (100 - 1)*0.2) * 0.1*(1 - 0.1)/100
```

```
## [1] 0.01872
```

Enfin, nous pouvons remplacer cette valeur dans notre formule pour la statistique du test t :

$$t = \frac{p-\mu}{\sqrt{(1+(n-1)\rho)\frac{p(1-p)}{nc}}}$$

Nous pouvons utiliser cette formule pour calculer la distribution de la statistique de test sous l'hypothèse alternative, et donc pour calculer la puissance. La fonction `get_pow_ttest_thresh()` effectue ce calcul de puissance pour vous :

Hide

```
# calculer la puissance compte tenu de certaines entrées
get_pow_ttest_thresh(p = 0.1, n_samp = 100, n_clusters = 10, ICC = 0.1, p_thresh =
        0.05)
```

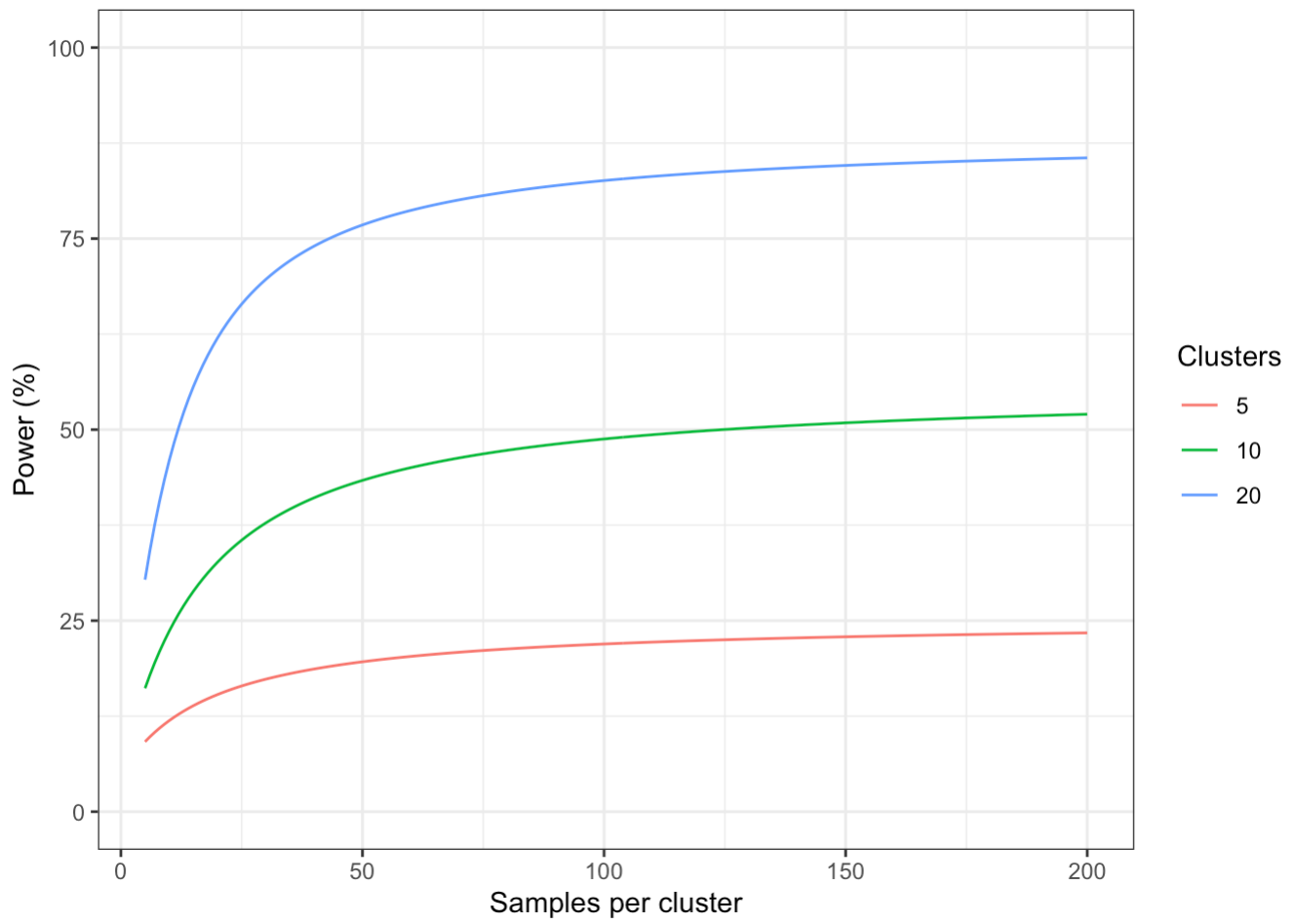
```
## [1] 0.2978355
```

Nous pouvons utiliser cette fonction pour produire des courbes de puissance pour une série de tailles d'échantillons et de nombres de grappes :

Hide

```
# input parameters
p <- 0.1
ICC <- 0.05
n_samp <- 5:200
n_clusters <- c(5, 10, 20)

# plot power curves
expand_grid(n_clusters, n_samp) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = n_samp, y = 100*pow, color = as.factor(n_clusters), group = n_
        clusters)) +
  ylim(c(0, 100)) + xlab("Samples per cluster") + ylab("Power (%)") +
  scale_color_discrete(name = "Clusters")
```



Sans surprise, nous constatons que la puissance augmente à mesure que nous obtenons plus d'échantillons et que nous recrutons plus de grappes. Cependant, il y a aussi des choses très intéressantes à remarquer à propos de ces courbes de puissance. Tout d'abord, contrairement aux courbes de puissance que nous avons vues dans la pratique précédente, il ne semble pas que ces courbes s'approchent de 100 %. Nous pouvons le vérifier à l'aide de la fonction suivante, qui calcule la puissance théorique maximale pouvant être atteinte :

Hide

```
# calculate maximum possible power
get_max_pow_ttest_thresh(p = p, n_clusters = n_clusters, ICC = ICC)
```

```
## [1] 0.2511925 0.5568322 0.8850831
```

Pour les paramètres ci-dessus et pour 5 clusters, la puissance maximale que nous pourrions atteindre serait d'environ 25 %. Pour 10 clusters, nous pouvons obtenir jusqu'à 56 % de puissance, et pour 20 clusters jusqu'à 89 %. Cela nous dit que si notre objectif est d'atteindre 80% de puissance alors il y a un nombre minimum de clusters que nous *devons* recruter pour qu'il soit *possible* que nous atteignions cette puissance.

Q12. Quel est le nombre minimum de clusters que nous devons recruter dans l'exemple ci-dessus pour avoir une chance d'atteindre 80 % de puissance ?

Click For Answer

A12. Nous avons besoin d'au moins 17 clusters pour qu'il nous soit théoriquement possible d'atteindre 80 % de puissance.

Hide

```
data.frame(clusters = 5:20,
           max_power = get_max_pow_ttest_thresh(p = p, n_clusters = 5:20, ICC = IC
        C))
```

```
##    clusters max_power
## 1         5 0.2511925
## 2         6 0.3174829
## 3         7 0.3822125
## 4         8 0.4441344
## 5         9 0.5024881
## 6        10 0.5568322
## 7        11 0.6069494
## 8        12 0.6527850
## 9        13 0.6944024
## 10       14 0.7319490
## 11       15 0.7656304
## 12       16 0.7956894
## 13       17 0.8223907
## 14       18 0.8460082
## 15       19 0.8668165
## 16       20 0.8850831
```

La prochaine chose intéressante à noter à partir de ces courbes de puissance concerne le nombre total d'échantillons sur tous les clusters. Le code suivant calcule la taille d'échantillon requise pour atteindre une puissance cible donnée, jusqu'à une valeur maximale donnée :

Hide

```
# define parameters
p <- 0.1
ICC <- 0.01
n_clusters <- c(5, 10, 20)
target_power <- 0.8

# search through sample sizes until reach target power
n_samp_optimal <- expand_grid(n_clusters, n_samp = 5:1e3) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  group_by(n_clusters) %>%
  summarise(n_samp = n_samp[which(pow > target_power)[1]])

n_samp_optimal
```

```
## # A tibble: 3 × 2
##   n_clusters n_samp
##        <dbl>  <int>
## 1          5     NA
## 2         10     55
## 3         20     19
```

Dans ce cas, nous constatons que pour 5 clusters, nous ne pouvons pas atteindre la puissance cible dans la plage de recherche (et en fait, c'est théoriquement impossible pour n'importe quelle taille d'échantillon). Pour 10 grappes, nous avons besoin de 55 échantillons par grappe, et pour 20 grappes, nous avons besoin de 19 échantillons par grappe. Notez que le nombre total d'échantillons dans le cas de 10 clusters est de `10 * 55 = 550`, et dans le cas de 20 clusters est de `20 * 19 = 380` Donc, quelque peu contre-intuitif, il est ** plus efficace statistiquement * * pour recruter un plus grand nombre de grappes, car cela signifie que le nombre total d'échantillons dans l'ensemble de l'étude diminue.

Q13. Supposons que la prévalence réelle des délétions *pfhrp2/3* est de 7 % et que la corrélation intra-cluster est de 0,01. Quelle est la taille totale de l'échantillon nécessaire si nous recrutons 20 cliniques dans notre province? Quelle amélioration obtiendrons-nous si nous parvenons à recruter 10 cliniques supplémentaires ?

Click For Answer

A13. Pour 20 cliniques, nous avons besoin de 122 échantillons par clinique, pour un total de 2440 échantillons. Pour 30 cliniques, nous avons besoin de 42 échantillons par clinique pour un total de 1260 échantillons. Ainsi, les 10 cliniques supplémentaires signifient que nous avons besoin d'environ la moitié du nombre d'échantillons au total.

Hide

```
# define parameters
p <- 0.08
ICC <- 0.02
n_clusters <- c(20, 30)
target_power <- 0.8

# search through sample sizes until reach target power
n_samp_optimal <- expand_grid(n_clusters, n_samp = 5:1e3) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  group_by(n_clusters) %>%
  summarise(n_samp = n_samp[which(pow > target_power)[1]])

n_samp_optimal
```

```
## # A tibble: 2 × 2
##   n_clusters n_samp
##        <dbl>  <int>
## 1         20    122
## 2         30     42
```

En règle générale, c'est toujours une bonne idée de **recruter plus de clusters, pas plus d'individus par cluster**. Non seulement cela a tendance à être une stratégie plus efficace, comme nous l'avons vu ci-dessus, mais cela rend également notre échantillonnage plus ** représentatif ** de la population. Si nous ne réussissions à recruter que 3 cliniques, nous *pourrions* être en mesure d'atteindre notre puissance souhaitée, mais 3 cliniques peuvent-elles vraiment capturer une province entière ? Il est tout à fait possible que ce que nous voyons dans ces cliniques ne soit pas une représentation exacte de la province dans son ensemble. C'est l'une des raisons pour lesquelles le protocole principal de l'OMS recommande de viser au moins 10 cliniques par district.

Une mise en garde importante à ce qui précède est que le recrutement d'un grand nombre de clusters peut entraîner des coûts. Créer un cluster, former du personnel, etc. signifient qu'il peut y avoir des coûts fixes à prendre en compte dans notre budget global. Ainsi, bien que le recrutement de plus de grappes puisse être plus * statistiquement * efficace, il n'est pas nécessairement plus efficace en termes de conception globale de l'étude.

Q14. En utilisant les paramètres de la question 12, supposons que nous concevons un essai dans le cadre d'un budget strict. Le coût de mise en place d'un cluster est estimé à 500 USD. Le coût par échantillon du traitement et du séquençage est estimé à 20 USD par échantillon. Quel nombre de clusters fournit le moyen le plus rentable de concevoir l'étude, en supposant que nous visons une puissance de 80 % ?

Click For Answer

A14. L'option la plus rentable consiste à recruter 34 grappes, avec 33 échantillons par grappe. Cela a un coût de 39 440 USD. Tout ce qui se situe entre 31 et 35 clusters aurait également des coûts similaires.

Hide

```
# define parameters
p <- 0.08
ICC <- 0.02
n_clusters <- 30:40
target_power <- 0.8

# search through sample sizes until reach target power
n_samp_optimal <- expand_grid(n_clusters, n_samp = 5:1e3) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  group_by(n_clusters) %>%
  summarise(n_samp = n_samp[which(pow > target_power)[1]])

# convert sample sizes to costs
n_samp_optimal %>%
  mutate(total_samples = n_clusters * n_samp,            # calculate total sample
        size
         total_cost = 20*total_samples + 500*n_clusters)  # calculate total cost
```

```
## # A tibble: 11 × 4
##    n_clusters n_samp total_samples total_cost
##         <int>  <int>         <int>      <dbl>
##  1         30     42          1260      40200
##  2         31     39          1209      39680
##  3         32     37          1184      39680
##  4         33     35          1155      39600
##  5         34     33          1122      39440
##  6         35     32          1120      39900
##  7         36     30          1080      39600
##  8         37     29          1073      39960
##  9         38     28          1064      40280
## 10         39     26          1014      39780
## 11         40     25          1000      40000
```

Nous avons vu comment l’échantillonnage en grappes peut conduire à des considérations statistiques supplémentaires. Si nous ne tenons pas compte de ces problèmes, nous pouvons obtenir des résultats trompeurs qui, dans le pire des cas, peuvent conduire à des recommandations incorrectes au niveau national. Lorsque nous tenons compte de ces problèmes, nous constatons qu’il est souvent souhaitable de recruter un grand nombre de grappes, plutôt que d’échantillonner continuellement plus d’individus de la même grappe. Cependant, cela doit être mis en balance avec les contraintes logistiques et budgétaires, qui pourraient finalement limiter le nombre de clusters disponibles. Enfin, notons que les méthodes d’analyse présentées ici ne sont qu’un moyen simple d’analyser des données d’enquête en grappes. Il existe de nombreuses autres techniques, par exemple la modélisation multi-niveaux, qui peuvent présenter certains avantages mais dépassent le cadre de ce TP.