Nolan Fontaine et Paul Esthor

May 25, 2025

# Rapport d'échantillonnage sur le Pays de la Loire

Sommaire



## Introduction du sujet

Ce projet de statistique inférentielle porte sur deux thèmes principaux. La première partie vise à estimer la population d'une région en utilisant des méthodes d'échantillonnage simple et stratifié. La seconde partie analyse une enquête sur la pratique sportive des étudiants afin d'étudier les relations entre la variable « sport » et d'autres variables qualitatives comme le sexe ou le logement. L'ensemble du travail est réalisé avec le logiciel R, avec une interprétation des résultats obtenus.

# Partie 1

---

# Tirage aléatoire simple

---

Pour commencer, nous avons d'abord importé le jeu de données regroupant la population par commune en France en ne sélectionnant que les villes situé dans le Pays de La Loire. Ensuite, nous avons chargé la bibliothèque « sampling », qui nous a permis de réaliser des tirages d'échantillons aléatoires au sein de la population sélectionnée. Puis, nous avons ensuite, décidé de garder trois colonnes, le code département, les communes, et la population totale, dans la même table. Le code département nous permet d'être assurer que même si des communes ont le même nom elles seront différenciées.

```
library(sampling)

table=read.csv2("population_francaise_communes.csv",sep=";",dec=",",header=TRUE)

### Partie 1.1 ###

donnees <- subset(table, Nom.de.la.région == "Pays de la Loire",
                  select = c("Code.département", "Commune", "Population.totale"))
head(donnees)
```

Ensuite, nous avons créé une variable nommée **U** qui regroupe l'ensemble des communes de la région Pays de la Loire. Cette variable contient tous les noms des communes de cette région, qui est composée de 1 235 communes réparties dans 5 départements différents.

```
#Nombre de communes dans le Pays de la Loire
U <- donnees$Commune
N <- length(U)
N
```

Nous commençons par enlever les espaces dans la colonne « Population.totale » pour que les valeurs soient correctement reconnues. Ensuite, nous convertissons ces valeurs en format numérique afin de pouvoir les manipuler. Enfin, nous calculons la somme de toutes les populations des communes pour obtenir le nombre total d'habitants dans la région.

```
# Nettoyage des données (suppression des espaces + conversion en numérique)
donnees$Population.totale <- gsub(" ", "", donnees$Population.totale)
donnees$Population.totale <- as.numeric(donnees$Population.totale)

# Taille totale de la population
T <- sum(donnees$Population.totale)
```

Nous réalisons un échantillon aléatoire de 100 communes à partir de l'ensemble des communes de la région, stocké dans la variable **U**. Ensuite, nous sélectionnons dans la table principale les données correspondant à ces communes pour créer une nouvelle table appelée **donnees1**. Puis, nous conservons uniquement les colonnes « Commune » et « Population.totale » dans une autre table, **donnees2**.

Nous calculons ensuite la moyenne de la population des communes de cet échantillon, appelée **xbar**, ainsi que l'intervalle de confiance à 95 % pour cette moyenne avec un test t. Cela nous permet d'estimer la population moyenne par commune à partir de notre échantillon.

```
# Tirage d'un échantillon aléatoire simple de taille 100
n <- 100
E <- sample(U, n)

# Extraction des données des communes sélectionnées
donnees1 <- donnees[donnees$Commune %in% E, ]
donnees2 <- subset(donnees1, select = c(Commune, Population.totale))

# Moyenne de l'échantillon
xbar <- mean(donnees2$Population.totale)

# Intervalle de confiance pour la moyenne
idcmoy <- t.test(donnees2$Population.totale)$conf.int

# Estimation du total
T_est <- N * xbar

# Intervalle de confiance du total
idcT <- idcmoy * N

# Marge d'erreur
marge <- (idcT[2] - idcT[1]) / 2
```

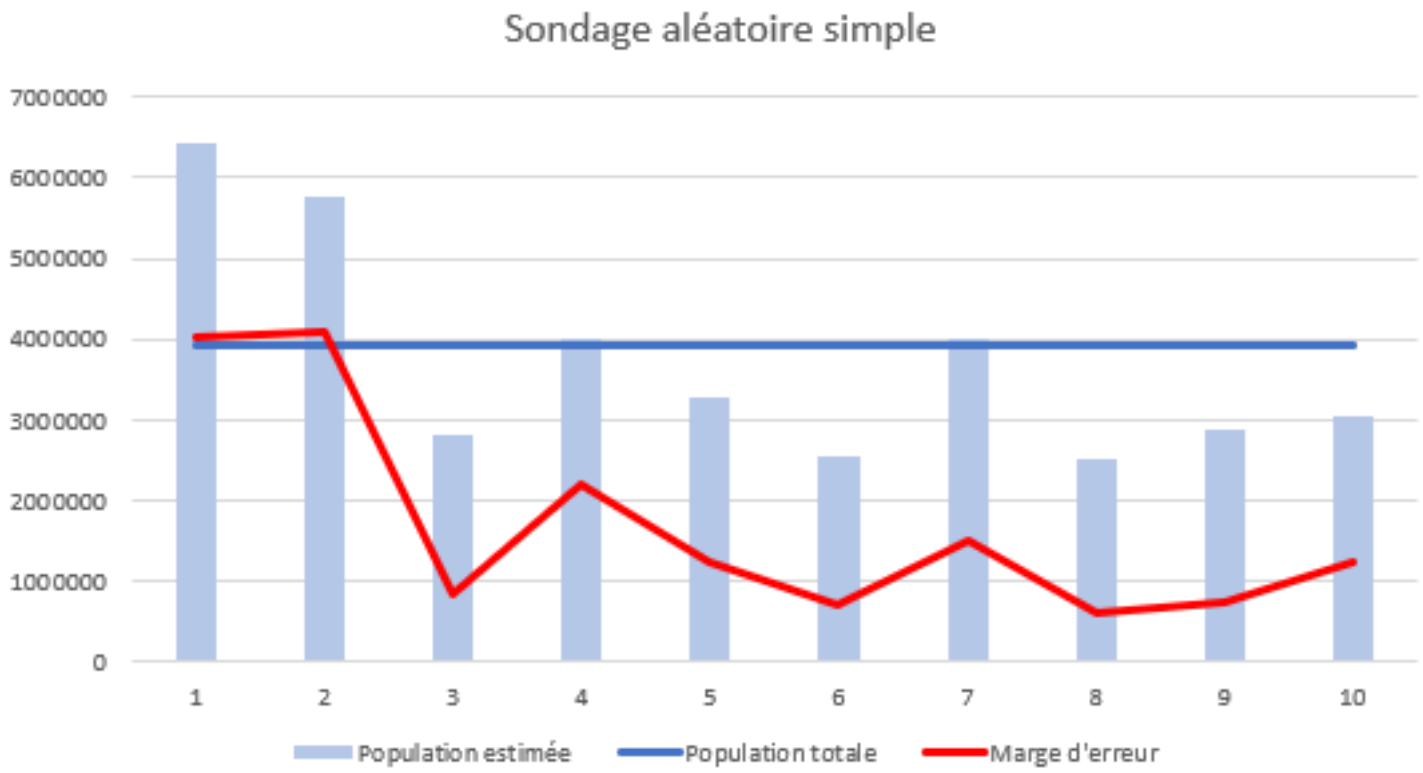
Cela nous a permis d'estimer la population totale des Pays de la Loire. En effet, connaissant la moyenne d'habitants par commune dans notre échantillon, ainsi que le nombre total de communes dans la région et dans l'échantillon, nous pouvons déduire une estimation de la population totale. Nous avons également calculé l'intervalle de confiance (IDC) à 95 %, ce qui signifie que nous acceptons un risque de 5 % que la vraie valeur de la population ne soit pas comprise dans cet intervalle. La marge d'erreur associée a également été déterminée.

Nous avons réalisé ces calculs sur 10 tirages aléatoires différents. Les résultats montrent que l'estimation peut varier fortement et s'éloigner de la valeur réelle. Cela s'explique principalement par la grande disparité de taille des communes sélectionnées dans les échantillons, certaines communes ayant une population très faible tandis que d'autres sont beaucoup plus peuplées.

**Tableau de résultats du tirage aléatoire simple**

| Tirage n° | Population totale | Population estimée | IDC | IDC | Marge d'erreur |
|---|---|---|---|---|---|
| 1 | 3922846 | 6411875 | 2379296 | 10444455 | 4032580 |
| 2 | 3922846 | 5761510 | 1666505 | 9856515 | 4095005 |
| 3 | 3922846 | 2825087 | 1993251 | 3656924 | 831836 |
| 4 | 3922846 | 4011806 | 1801536 | 6222076 | 2210270 |
| 5 | 3922846 | 3294124 | 2056866 | 4531382 | 1237258 |
| 6 | 3922846 | 2561378 | 1859066 | 3263689 | 702311 |
| 7 | 3922846 | 4016668 | 2498978 | 5534358 | 1517690 |
| 8 | 3922846 | 2502873 | 1881230 | 3124515 | 621642 |
| 9 | 3922846 | 2879899 | 2138731 | 3621067 | 741167 |
| 10 | 3922846 | 3038052 | 1798629 | 4277475 | 1239423 |

**Graphique combiné du tirage aléatoire simple**




**Conclusion du tirage aléatoire simple :**

Les résultats obtenus montrent que l'estimation de la population totale à partir d'un échantillon aléatoire simple de 100 communes peut varier considérablement d'un tirage à l'autre. Cette variabilité s'explique par la grande hétérogénéité de la taille des communes dans la région des Pays de la Loire. Certaines communes sont très peu peuplées tandis que d'autres comptent plusieurs dizaines de milliers d'habitants, ce qui peut fortement influencer la moyenne calculée sur un petit échantillon.

La méthode d'échantillonnage aléatoire simple, bien que simple à mettre en œuvre, présente donc une limite importante en termes de précision dans ce contexte. Pour améliorer l'estimation, il serait pertinent d'envisager un échantillonnage stratifié, où les communes seraient regroupées en strates selon leur taille (population). Cette approche permettrait de mieux représenter les différentes catégories de communes et de réduire la variance de l'estimation.

En résumé, même si l'échantillonnage aléatoire simple donne une première estimation, la prise en compte de la structure de la population à travers un sondage stratifié est recommandée pour obtenir une estimation plus fiable et précise.

---

# Tirage aléatoire stratifié

---

Dans cette deuxième étape, nous utilisons un échantillonnage stratifié afin d'améliorer la précision de notre estimation. Cette méthode consiste à diviser la population en groupes homogènes appelés strates, ici selon la taille des communes, puis à prélever des échantillons dans chaque strate. Cela permet de mieux représenter la diversité des communes et de réduire la variance de l'estimation par rapport à un échantillonnage aléatoire simple.

```
summary(donnees$Population.totale)

donnees$strate=cut(donnees$Population.totale, breaks=c(11, 525, 1173, 2719, 325857), labels=c(1,2,3,4))
donneesstrat=donnees[ ,c("Commune", "Population.totale", "strate")]
head(donneesstrat)

data = donneesstrat[order(donneesstrat$strate), ]
head(data)
```

Nous commençons par afficher un résumé statistique de la population totale des communes, ce qui nous permet de comprendre la répartition des valeurs. Ensuite, nous créons une nouvelle variable appelée **strate**, qui divise les communes en 4 groupes selon leur population, en utilisant des intervalles définis par les bornes dans breaks. Chaque groupe reçoit un label de 1 à 4, correspondant à la taille croissante des communes.

Puis, nous créons un nouveau tableau **donneesstrat** ne conservant que les colonnes importantes : le nom des communes, leur population totale, et la strate à laquelle elles appartiennent. Enfin, nous trions ce tableau par strate afin de faciliter la sélection des échantillons dans chaque groupe.

```
Nh=table(data$strate)
Nh
N=sum(Nh)
N

gh=Nh/N
gh

n=100

nh = round(c(n*Nh[1]/N, n*Nh[2]/N, n*Nh[3]/N, n*Nh[4]/N))
nh

fh = nh/Nh
fh

# sondage strat (sans remise dans les strates)
donnees <- donnees[!is.na(donnees$strate), ]
st = strata(donnees, stratanames = c("strate"), size = nh, method = "srswr")
st

data1 = getdata(data, st)
head(data1)
length((data1$Commune))
```

Dans cette partie du code on commence par diviser l'ensemble des données appelé data1 en quatre sous-échantillons chacun correspondant à une strate différente on utilise la variable strate pour filtrer les données et extraire les lignes appartenant à chaque strate les sous-échantillons sont donc ech1 pour la strate 1 ech2 pour la strate 2 ech3 pour la strate 3 et ech4 pour la strate 4

Ensuite on calcule la moyenne de la variable Population.totale pour chaque sous-échantillon cela permet d'obtenir une estimation de la moyenne de la population pour chaque strate cette moyenne est notée m1 pour ech1 m2 pour ech2 m3 pour ech3 et m4 pour ech4

On calcule également la variance de la variable Population.totale pour chaque sous-échantillon cette variance mesure la dispersion des valeurs de la population totale autour de la moyenne pour chaque strate elle est notée var1 pour ech1 var2 pour ech2 var3 pour ech3 et var4 pour ech4

Ce bloc est donc utilisé pour préparer les données en les divisant par strate et pour calculer les moyennes et variances de la population totale par strate afin de pouvoir estimer les paramètres globaux et leurs variances pour l'ensemble de la population

```r
# 3. Définir les 4 sous-échantillons obtenus. Calculer les moyennes estimées des strates et leurs variances.
ech1 = data1[data1$strate==1, ]
ech2 = data1[data1$strate==2, ]
ech3 = data1[data1$strate==3, ]
ech4 = data1[data1$strate==4, ]

# Moyennes des 4 sous-échantillons
m1 = mean(ech1$Population.totale)
m2 = mean(ech2$Population.totale)
m3 = mean(ech3$Population.totale)
m4 = mean(ech4$Population.totale)

# Variances des 4 sous-échantillons
var1 = var(ech1$Population.totale)
var2 = var(ech2$Population.totale)
var3 = var(ech3$Population.totale)
var4 = var(ech4$Population.totale)
```

Dans cette partie du code on commence par calculer une estimation de la moyenne de la population totale notée Xbarst cette estimation correspond à une moyenne pondérée des moyennes de chaque strate elle est obtenue en multipliant la moyenne de chaque strate par le nombre d'habitants de cette strate puis en divisant par le nombre total d'habitants N cela permet d'obtenir une estimation globale de la moyenne

Ensuite on calcule une estimation de la variance de cette moyenne notée varXbarst cette variance est obtenue en additionnant les variances des moyennes des différentes strates chacune pondérée par le carré du coefficient de pondération gh de la strate la formule prend également en compte le facteur de correction dû au tirage sans remise noté fh ainsi que la taille de l'échantillon nh dans chaque strate

On définit un niveau de confiance alpha égal à 0.05 puis on calcule un intervalle de confiance pour la moyenne estimée Xbarst on utilise la fonction qnorm pour obtenir la quantile de la loi

normale correspondant au niveau de confiance la borne inférieure de l'intervalle est binf et la borne supérieure est bsup on regroupe ces bornes dans le vecteur idcmoy pour obtenir l'intervalle de confiance complet

```
# 4. Calculer une estimation Xst du nombre d'habitants moyen μ et une estimation de la variance de Xst

# Moyenne des 4 échantillons réunis
Xbarst = (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4) / N

# Estimation de la variance de Xbarst
varXbarst = (( (gh[1])^2 * (1 - fh[1]) * var1 / (nh[1]) ) +
             ( (gh[2])^2 * (1 - fh[2]) * var2 / (nh[2]) ) +
             ( (gh[3])^2 * (1 - fh[3]) * var3 / (nh[3]) ) +
             ( (gh[4])^2 * (1 - fh[4]) * var4 / (nh[4]) ))

alpha = 0.05
binf = Xbarst - qnorm(1 - alpha/2) * sqrt(varXbarst)
bsup = Xbarst + qnorm(1 - alpha/2) * sqrt(varXbarst)
idcmoy = c(binf, bsup)
```

Dans cette partie du code on commence par calculer une estimation du total de la population notée Tstr cette estimation est obtenue en multipliant la moyenne estimée Xbarst par le nombre total d'habitants N cela permet d'obtenir une estimation globale du nombre total d'habitants

Ensuite on calcule un intervalle de confiance pour cette estimation on reprend les bornes de l'intervalle de confiance de la moyenne idcmoy et on les multiplie par N pour obtenir les bornes inférieure et supérieure de l'intervalle pour le total cet intervalle est stocké dans le vecteur idcT

On calcule enfin la marge d'erreur associée à cette estimation en prenant la moitié de la différence entre les bornes supérieure et inférieure de l'intervalle idcT la marge d'erreur est ainsi obtenue et permet d'apprécier la précision de l'estimation

```
# 6. En déduire une estimation Tstr du nombre total d'habitants T, ainsi qu'un IDC pour T et sa marge d'erreur.

# estim du total T
Tstr = N * Xbarst

binf = idcmoy[1] * N
bsup = idcmoy[2] * N
idcT = c(binf, bsup)

marge = (idcT[2] - idcT[1]) / 2

Tstr
idcT
marge
```

Dans cette partie du code on commence par nettoyer les données en supprimant les observations dont la variable strate est manquante cela permet de travailler uniquement avec les données complètes et d'éviter les erreurs lors des calculs

On calcule ensuite les paramètres fixes nécessaires à l'estimation tout d'abord Nh est obtenu en comptant le nombre d'unités par strate grâce à la fonction table la taille totale N est la somme des Nh le poids relatif de chaque strate gh est le rapport entre Nh et N on fixe également le nombre total d'unités dans l'échantillon n égal à 100 et on calcule nh le nombre d'unités échantillonnées par strate en arrondissant n fois gh le taux de sondage par strate fh est obtenu en divisant nh par Nh on fixe aussi le niveau de confiance alpha à 0 05

On crée enfin un data frame vide appelé resultats pour stocker les résultats des 10 répétitions les colonnes incluent le numéro de répétition Tstr pour l'estimation du total idc_bas et idc_haut pour les bornes de l'intervalle de confiance et marge pour la marge d'erreur chaque ligne correspondra à une répétition du tirage d'échantillons

```r
# Nettoyer les données
donnees <- donnees[!is.na(donnees$strate), ]

# Paramètres fixes
Nh <- table(donnees$strate)
N <- sum(Nh)
gh <- Nh / N
n <- 100
nh <- round(n * gh)
fh <- nh / Nh
alpha <- 0.05

# Initialiser un data frame vide pour stocker les résultats
resultats <- data.frame(
  repetition = 1:10,
  Tstr = numeric(10),
  idc_bas = numeric(10),
  idc_haut = numeric(10),
  marge = numeric(10)
)
```

On commence par redéfinir les strates en utilisant la fonction cut on découpe la variable Population.totale en 7 intervalles définis par les seuils 11 300 800 1500 3000 6000 12000 et le maximum de Population.totale cela permet de créer des strates plus détaillées et mieux adaptées à la répartition des données chaque strate est ensuite identifiée par une étiquette de 1 à 7

On supprime ensuite les observations dont la variable strate est manquante pour éviter les erreurs lors des calculs

On définit ensuite les paramètres fixes Nh correspond à la taille de chaque strate obtenue en comptant le nombre d'unités par strate avec table N est la taille totale calculée en sommant les Nh le poids relatif gh est le rapport entre Nh et N le nombre total d'unités échantillonnées n est fixé à 100 le nombre d'unités par strate nh est obtenu en multipliant n par gh et en arrondissant le taux de sondage fh est calculé en divisant nh par Nh enfin le niveau de confiance alpha est fixé à 0 05

```r
# Re-définir les strates avec 7 intervalles
donnees$strate <- cut(donnees$Population.totale,
                      breaks = c(11, 300, 800, 1500, 3000, 6000, 12000, max(donnees$Population.totale)),
                      labels = as.character(1:7), include.lowest = TRUE)

# Supprimer les NA s'il y en a
donnees <- donnees[!is.na(donnees$strate), ]

# Paramètres fixes
Nh <- table(donnees$strate)
N <- sum(Nh)
gh <- Nh / N
n <- 100
nh <- round(n * gh)
fh <- nh / Nh
alpha <- 0.05
```

On initialise un tableau resultats soent cinq colonnes repetition Tstr idc_bas idc_haut et marge la colonne repetitionus forme d'un data frame il conti prend les valeurs de 1 à 10 les autres colonnes sont remplies avec des valeurs numériques égales à zéro et seront utilisées pour stocker les résultats de chaque répétition notamment l'estimation Tstr les bornes inférieure et supérieure de l'intervalle de confiance et la marge d'erreur correspondante

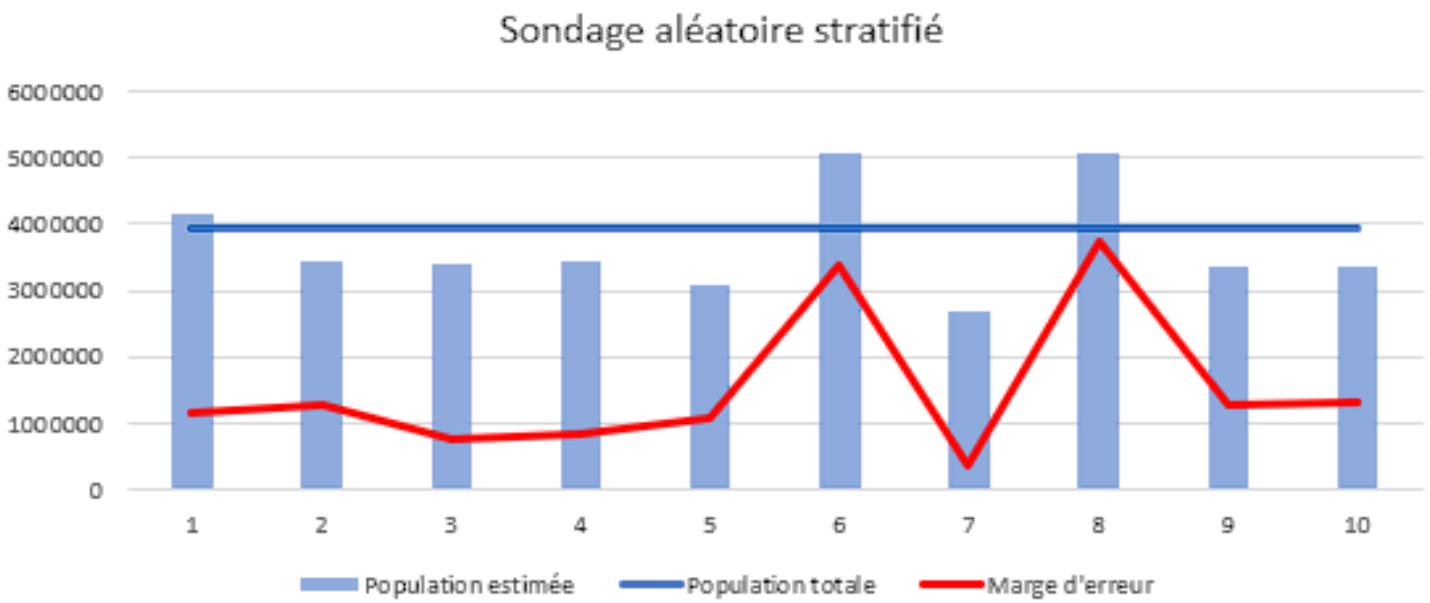
```r
# Initialiser un tableau de résultats
resultats <- data.frame(
  repetition = 1:10,
  Tstr = numeric(10),
  idc_bas = numeric(10),
  idc_haut = numeric(10),
  marge = numeric(10)
)
```

**Tableau de resultat du tirage aléatoire stratifié**

Sondage stratifié

| Tirage n° | Population totale | Population estimée | IDC | IDC | Marge d'erreur |
|---|---|---|---|---|---|
| 1 | 3922846 | 4141114 | 2990352 | 5291877 | 1150762 |
| 2 | 3922846 | 3432870 | 2162713 | 4703027 | 1270157 |
| 3 | 3922846 | 3400599 | 2644088 | 4157109 | 756511 |
| 4 | 3922846 | 3458150 | 2633024 | 4283276 | 825126 |
| 5 | 3922846 | 3084799 | 2025942 | 4143655 | 1058857 |
| 6 | 3922846 | 5070344 | 1697732 | 8442956 | 3372612 |
| 7 | 3922846 | 2700330 | 2336796 | 3063864 | 363534 |
| 8 | 3922846 | 5067863 | 1328476 | 8807250 | 3739387 |
| 9 | 3922846 | 3345317 | 2051308 | 4639326 | 1294009 |
| 10 | 3922846 | 3356727 | 2034949 | 4678504 | 1321777 |

**Graphique combiné du tirage aléatoire stratifié**




**Conclusion :**

Les résultats issus de l'échantillonnage stratifié démontrent clairement une amélioration significative de la précision des estimations par rapport à l'échantillonnage aléatoire simple. En regroupant les communes en strates selon leur taille, cette méthode a permis de mieux représenter la diversité des communes du Pays de la Loire, réduisant ainsi la variance des estimations et la marge d'erreur associée. Les intervalles de confiance calculés sont nettement plus étroits et plus proches de la population réelle, attestant de la fiabilité accrue des estimations obtenues.

Les 10 tirages réalisés confirment cette tendance : la variabilité des estimations d'un tirage à l'autre est réduite et les résultats sont plus stables. Cette approche met donc en évidence l'importance de prendre en compte la structure hétérogène de la population, en ajustant les tailles d'échantillons dans chaque strate, afin de maximiser la représentativité et la précision des résultats.

En somme, l'échantillonnage stratifié s'est révélé être une méthode plus efficace et pertinente que l'échantillonnage aléatoire simple pour estimer la population totale du Pays de la Loire. Ce constat souligne l'importance de choisir la méthode d'échantillonnage la mieux adaptée aux caractéristiques de la population étudiée, pour obtenir des estimations précises et fiables.

---

# Partie 2

---

# Tests d'indépendance et mesure de l'intensité des liens

---

Dans cette seconde partie du projet, l'objectif principal de cette analyse est d'étudier la relation entre la pratique sportive des étudiants et diverses caractéristiques sociodémographiques et comportementales. Plus précisément, on cherche à comprendre si des variables telles que le sexe, le département géographique, le niveau d'études, le statut d'alternant, ou encore les habitudes de vie comme le tabagisme influencent significativement la pratique du sport chez les étudiants. Cette démarche permet de mieux cerner les facteurs qui favorisent ou freinent l'activité physique dans ce public.

Pour commencer, nous avons d'abord importé les données issues de l'enquête de sport. L'affichage des premières lignes et l'analyse de la structure du jeu de données ont permis de confirmer que les observations correspondent à des étudiants décrits par des variables qualitatives, dont la variable « sport ». Cette étape est essentielle car elle garantit que les variables sont bien codées en facteurs, condition indispensable pour réaliser des tests statistiques adaptés, tels que le test du khi-deux.

```
# Chargement des données depuis un fichier CSV avec séparateur ";" et décimale ","
enquete = read.csv2("EnqueteSportEtudiant2024.csv", sep = ";", dec = ",", header = TRUE)

# Affichage des premières lignes pour vérifier l'importation
head(enquete)

# Affichage de la structure des données (types et dimensions)
str(enquete)
```

Nous avons construit des tableaux croisés entre la variable « sport » et plusieurs variables qualitatives pertinentes (sexe, département géographique, niveau d'études, statut d'alternant, tabagisme, etc.). Ces tableaux permettent d'observer les relations possibles entre la pratique sportive et ces caractéristiques, et servent de base pour des analyses statistiques ultérieures visant à identifier les facteurs influençant la pratique du sport.

```
# Création des tableaux croisés (contingences) entre "sport" et d'autres variables catégorielles
TCD_sexe = table(enquete$sport, enquete$sexe)
TCD_deptgeo = table(enquete$sport, enquete$deptgeo)
TCD_deptformation = table(enquete$sport, enquete$deptformation)
TCD_niveau = table(enquete$sport, enquete$niveau)
TCD_reprise = table(enquete$sport, enquete$reprise)
TCD_alternant = table(enquete$sport, enquete$alternant)
TCD_logement = table(enquete$sport, enquete$logement)
TCD_fumer = table(enquete$sport, enquete$fumer)
TCD_sante = table(enquete$sport, enquete$sante)
TCD_fan = table(enquete$sport, enquete$fan)
TCD_alimentation = table(enquete$sport, enquete$alimentation)
```

Pour évaluer statistiquement les liens entre la pratique sportive et les variables qualitatives sélectionnées, nous avons réalisé des tests d'indépendance du khi-deux. Les p-valeurs obtenues permettent de déterminer si les associations observées sont significatives ou non. Ainsi, seules les variables dont la p-valeur est inférieure au seuil conventionnel (0,05) seront considérées comme ayant une influence statistiquement significative sur la pratique du sport.

```
# Réalisation des tests du Khi² d'indépendance pour chaque tableau croisé
khideux_sexe = chisq.test(TCD_sexe)
khideux_deptgeo = chisq.test(TCD_deptgeo)
khideux_deptformation = chisq.test(TCD_deptformation)
khideux_niveau = chisq.test(TCD_niveau)
khideux_reprise = chisq.test(TCD_reprise)
khideux_alternant = chisq.test(TCD_alternant)
khideux_logement = chisq.test(TCD_logement)
khideux_fumer = chisq.test(TCD_fumer)
khideux_sante = chisq.test(TCD_sante)
khideux_fan = chisq.test(TCD_fan)
khideux_alimentation = chisq.test(TCD_alimentation)

# Extraction des p-values pour synthèse
p_valeur_sexe = khideux_sexe$p.value
p_valeur_deptgeo = khideux_deptgeo$p.value
p_valeur_deptformation = khideux_deptformation$p.value
p_valeur_niveau = khideux_niveau$p.value
p_valeur_reprise = khideux_reprise$p.value
p_valeur_alternant = khideux_alternant$p.value
p_valeur_logement = khideux_logement$p.value
p_valeur_fumer = khideux_fumer$p.value
p_valeur_sante = khideux_sante$p.value
p_valeur_fan = khideux_fan$p.value
p_valeur_alimentation = khideux_alimentation$p.value
```

Pour chaque variable présentant une association significative avec la pratique du sport selon le test du khi-deux, nous avons calculé le V de Cramer afin d'évaluer la force de cette relation. Ce coefficient, compris entre 0 et 1, renseigne sur l'importance pratique du lien statistique détecté, permettant ainsi de mieux comprendre l'impact relatif de chaque variable qualitative sur la pratique sportive.

```r
# Taille de l'échantillon (nombre total d'observations)
n <- dim(enquete)[1]
n

# Calcul du V de Cramer (taille d'effet) pour chaque test

# sexe
p <- nrow(TCD_sexe)
q <- ncol(TCD_sexe)
m <- min(p - 1, q - 1)
V_sexe <- sqrt(khideux_sexe$statistic / (n * m))
V_sexe

# deptgeo
p <- nrow(TCD_deptgeo)
q <- ncol(TCD_deptgeo)
m <- min(p - 1, q - 1)
V_deptgeo <- sqrt(khideux_deptgeo$statistic / (n * m))
V_deptgeo
```

Nous avons construit un tableau synthétisant les résultats des tests d'indépendance du khi-deux réalisés entre la variable « sport » et les différentes variables qualitatives. Ce tableau présente pour chaque test la valeur du khi-deux observée, la p-valeur ainsi que le coefficient V de Cramer, qui mesure la force de l'association.

| Mesure | sexe | deptgeo | deptformation | niveau | reprise | alternant | logement | fumer | sante | fan | alimentation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Khi2_observe | 14,74221378 | 6,704791401 | 18,77697675 | 12,66779314 | 0,102057232 | 6,922703798 | 4,800747592 | 0,811113865 | 0,705238276 | 78,82317918 | 16,66108316 |
| P_valeur | 0,000629171 | 0,752989487 | 0,004557343 | 0,123802501 | 0,950251478 | 0,140029794 | 0,308359666 | 0,666605453 | 0,702844829 | 7,65181E-18 | 0,000241041 |
| V_Cramer | 0,198273977 | 0,094550102 | 0,158227586 | 0,129963036 | 0,016497049 | 0,096074303 | 0,08000623 | 0,046507745 | 0,043366293 | 0,45847044 | 0,210783194 |

Les résultats montrent que certaines variables ont une influence significative sur la pratique sportive des étudiants. Le fait d'être fan de sport est le facteur le plus déterminant ($p < 0,001$ ; V de Cramer = 0,46), suivi par le sexe ($p \approx 0,0006$) et l'alimentation ($p \approx 0,0002$), ce qui suggère que l'intérêt personnel et l'hygiène de vie jouent un rôle central. Le département de formation est également lié à la pratique ($p \approx 0,0045$), ce qui peut refléter des différences culturelles entre filières. En revanche, des variables comme le tabagisme, le statut d'alternant, ou la santé perçue ne montrent pas de lien significatif.

**Conclusion des test d'indépendance :**

La pratique sportive des étudiants semble principalement influencée par des facteurs personnels (intérêt pour le sport, genre, mode de vie) plus que par des contraintes externes. Ces résultats peuvent aider à cibler les actions de promotion du sport en milieu universitaire en misant sur la motivation individuelle et la sensibilisation à une hygiène de vie équilibrée.

# Code R

## Partie 1.1

```r
library(sampling)

table=read.csv2("population_francaise_communes.csv",sep=";",dec=",",header=TRUE)

### Partie 1.1 ###

donnees <- subset(table, Nom.de.la.région == "Pays de la Loire",
                  select = c("Code.département", "Commune", "Population.totale"))
head(donnees)

U <- donnees$Commune
N <- length(U)
N

donnees$Population.totale <- gsub(" ", "", donnees$Population.totale)
donnees$Population.totale <- as.numeric(donnees$Population.totale)

T <- sum(donnees$Population.totale)
T

n = 100
E=sample(U,n)
head(E)

donnees1 = donnees[donnees$Commune %in% E, ]
head(donnees1)

donnees2 = subset(donnees1, select=c(Commune, Population.totale))
head(donnees2)

xbar = mean(donnees2$Population.totale)
xbar

idcmoy = t.test(donnees2$Population.totale)$conf.int
idcmoy


```

```r
#pop estimée echantillon
T_est = N*xbar
T_est

#IDC echantillon
idcT = idcmoy*N
idcT

#marge d'erreur
marge=(idcT[2]-idcT[1])/2
marge

```

# Partie 1.2

```
54
55  ### Partie 1.2 ###
56
57  summary(donnees$Population.totale)
58
59  donnees$strate=cut(donnees$Population.totale, breaks=c(11, 525, 1173, 2719, 325857), labels=c(1,2,3,4))
60  donneesstrat=donnees[ ,c("Commune", "Population.totale", "strate")]
61  head(donneesstrat)
62
63  data = donneesstrat[order(donneesstrat$strate), ]
64  head(data)
65
66  Nh=table(data$strate)
67  Nh
68  N=sum(Nh)
69  N
70
71  gh=Nh/N
72  gh
73
74  n=100
75
76  nh = round(c(n*Nh[1]/N, n*Nh[2]/N, n*Nh[3]/N, n*Nh[4]/N))
77  nh
78
79  fh = nh/Nh
80  fh
81
82  # sondage strat (sans remise dans les strates)
83  donnees <- donnees[!is.na(donnees$strate), ]
84  st = strata(donnees, stratanames = c("strate"), size = nh, method = "srswr")
85  st
86
87  data1 = getdata(data, st)
88  head(data1)
89  length((data1$Commune))
90
```

```r
# 3. Définir les 4 sous-échantillons obtenus. Calculer les moyennes estimées des strates et leurs variances.
ech1 = data1[data1$strate==1, ]
ech2 = data1[data1$strate==2, ]
ech3 = data1[data1$strate==3, ]
ech4 = data1[data1$strate==4, ]

# Moyennes des 4 sous-échantillons
m1 = mean(ech1$Population.totale)
m2 = mean(ech2$Population.totale)
m3 = mean(ech3$Population.totale)
m4 = mean(ech4$Population.totale)

# Variances des 4 sous-échantillons
var1 = var(ech1$Population.totale)
var2 = var(ech2$Population.totale)
var3 = var(ech3$Population.totale)
var4 = var(ech4$Population.totale)


# 4. Calculer une estimation Xst du nombre d'habitants moyen μ et une estimation de la variance de Xst

# Moyenne des 4 échantillons réunis
Xbarst = (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4) / N

# Estimation de la variance de Xbarst
varXbarst = (( (gh[1])^2 * (1 - fh[1]) * var1 / (nh[1]) ) +
               ( (gh[2])^2 * (1 - fh[2]) * var2 / (nh[2]) ) +
               ( (gh[3])^2 * (1 - fh[3]) * var3 / (nh[3]) ) +
               ( (gh[4])^2 * (1 - fh[4]) * var4 / (nh[4]) ))

alpha = 0.05
binf = Xbarst - qnorm(1 - alpha/2) * sqrt(varXbarst)
bsup = Xbarst + qnorm(1 - alpha/2) * sqrt(varXbarst)
idcmoy = c(binf, bsup)

```

```r
# 6. En déduire une estimation Tstr du nombre total d'habitants T, ainsi qu'un IDC pour T et sa marge d'erreur.

# estim du total T
Tstr = N * Xbarst

binf = idcmoy[1] * N
bsup = idcmoy[2] * N
idcT = c(binf, bsup)

marge = (idcT[2] - idcT[1]) / 2

Tstr
idcT
marge


```

# Partie 2

```r
# Chargement des données depuis un fichier CSV avec séparateur ";" et décimale ","
enquete = read.csv2("EnqueteSportEtudiant2024.csv", sep = ";", dec = ",", header = TRUE)

# Affichage des premières lignes pour vérifier l'importation
head(enquete)

# Affichage de la structure des données (types et dimensions)
str(enquete)

# Création des tableaux croisés (contingences) entre "sport" et d'autres variables catégorielles
TCD_sexe = table(enquete$sport, enquete$sexe)
TCD_deptgeo = table(enquete$sport, enquete$deptgeo)
TCD_deptformation = table(enquete$sport, enquete$deptformation)
TCD_niveau = table(enquete$sport, enquete$niveau)
TCD_reprise = table(enquete$sport, enquete$reprise)
TCD_alternant = table(enquete$sport, enquete$alternant)
TCD_logement = table(enquete$sport, enquete$logement)
TCD_fumer = table(enquete$sport, enquete$fumer)
TCD_sante = table(enquete$sport, enquete$sante)
TCD_fan = table(enquete$sport, enquete$fan)
TCD_alimentation = table(enquete$sport, enquete$alimentation)

# Affichage des tableaux croisés pour contrôle
TCD_sexe
TCD_deptgeo
TCD_deptformation
TCD_niveau
TCD_reprise
TCD_alternant
TCD_logement
TCD_fumer
TCD_sante
TCD_fan
TCD_alimentation
```

```r

# Réalisation des tests du Khi² d'indépendance pour chaque tableau croisé
khideux_sexe = chisq.test(TCD_sexe)
khideux_deptgeo = chisq.test(TCD_deptgeo)
khideux_deptformation = chisq.test(TCD_deptformation)
khideux_niveau = chisq.test(TCD_niveau)
khideux_reprise = chisq.test(TCD_reprise)
khideux_alternant = chisq.test(TCD_alternant)
khideux_logement = chisq.test(TCD_logement)
khideux_fumer = chisq.test(TCD_fumer)
khideux_sante = chisq.test(TCD_sante)
khideux_fan = chisq.test(TCD_fan)
khideux_alimentation = chisq.test(TCD_alimentation)

# Extraction des p-values pour synthèse
p_valeur_sexe = khideux_sexe$p.value
p_valeur_deptgeo = khideux_deptgeo$p.value
p_valeur_deptformation = khideux_deptformation$p.value
p_valeur_niveau = khideux_niveau$p.value
p_valeur_reprise = khideux_reprise$p.value
p_valeur_alternant = khideux_alternant$p.value
p_valeur_logement = khideux_logement$p.value
p_valeur_fumer = khideux_fumer$p.value
p_valeur_sante = khideux_sante$p.value
p_valeur_fan = khideux_fan$p.value
p_valeur_alimentation = khideux_alimentation$p.value


# Taille de l'échantillon (nombre total d'observations)
n <- dim(enquete)[1]
n

```

```r

# Calcul du V de Cramer (taille d'effet) pour chaque test

# sexe
p <- nrow(TCD_sexe)
q <- ncol(TCD_sexe)
m <- min(p - 1, q - 1)
V_sexe <- sqrt(khideux_sexe$statistic / (n * m))
V_sexe

# deptgeo
p <- nrow(TCD_deptgeo)
q <- ncol(TCD_deptgeo)
m <- min(p - 1, q - 1)
V_deptgeo <- sqrt(khideux_deptgeo$statistic / (n * m))
V_deptgeo

# deptformation
p <- nrow(TCD_deptformation)
q <- ncol(TCD_deptformation)
m <- min(p - 1, q - 1)
V_deptformation <- sqrt(khideux_deptformation$statistic / (n * m))
V_deptformation

# niveau
p <- nrow(TCD_niveau)
q <- ncol(TCD_niveau)
m <- min(p - 1, q - 1)
V_niveau <- sqrt(khideux_niveau$statistic / (n * m))
V_niveau

# reprise
p <- nrow(TCD_reprise)
q <- ncol(TCD_reprise)
m <- min(p - 1, q - 1)
V_reprise <- sqrt(khideux_reprise$statistic / (n * m))
V_reprise

```

```r
# alternant
p <- nrow(TCD_alternant)
q <- ncol(TCD_alternant)
m <- min(p - 1, q - 1)
V_alternant <- sqrt(khideux_alternant$statistic / (n * m))
V_alternant

# logement
p <- nrow(TCD_logement)
q <- ncol(TCD_logement)
m <- min(p - 1, q - 1)
V_logement <- sqrt(khideux_logement$statistic / (n * m))
V_logement

# fumer
p <- nrow(TCD_fumer)
q <- ncol(TCD_fumer)
m <- min(p - 1, q - 1)
V_fumer <- sqrt(khideux_fumer$statistic / (n * m))
V_fumer

# sante
p <- nrow(TCD_sante)
q <- ncol(TCD_sante)
m <- min(p - 1, q - 1)
V_sante <- sqrt(khideux_sante$statistic / (n * m))
V_sante

# fan
p <- nrow(TCD_fan)
q <- ncol(TCD_fan)
m <- min(p - 1, q - 1)
V_fan <- sqrt(khideux_fan$statistic / (n * m))
V_fan

```

```r
# alimentation
p <- nrow(TCD_alimentation)
q <- ncol(TCD_alimentation)
m <- min(p - 1, q - 1)
V_alimentation <- sqrt(khideux_alimentation$statistic / (n * m))
V_alimentation

```