Eliott Guénard, Loris Guéné & Paul Babin

Mr. Ibazizen

SAÉ – Échantillonnage et Estimation

(Provence-Alpes-Côte-d'Azur)

Lors de cette SAÉ, nous avions deux objectifs : le premier était d'estimer la population d'une région, ici la région Provence-Alpes-Côte d'Azur (que nous appellerons PACA dans le reste de ce compte rendu) à l'aide d'un intervalle de confiance réalisé à partir d'un processus d'échantillonnage, premièrement simple et ensuite par strates. Nous allons aussi comparer ces deux méthodes afin de déterminer laquelle est la plus efficace. Le deuxième objectif était de reprendre les données d'une enquête sur les étudiants et la pratique du sport afin de trouver des relations significatives entre le fait que les étudiants fassent du sport et d'autres variables qualitatives de notre choix.

Partie 1.1: Estimation du nombre d'habitants d'une région de France - Échantillonnage aléatoire simple

Nous avons commencé par importer le fichier Excel contenant les données sur la population française pour ensuite le filtrer afin d'avoir un dataframe (tableau) ne contenant que les données sur la région PACA.

```
library(sampling)
setwd("C:/Users/lguene01/OneDrive - Université de Poitiers/SAE/Stat inf")
#import des données
table = read.csv2(file ="population_francaise_communes.csv",sep = ";",dec=",",header=TRUE)

#################### -- Partie 1.1 -- ####################
#filtrage des données sur la région PACA
donnees = table[table$Nom.de.la.région == "Provence-Alpes-Côte d'Azur",c("Code.département","Commune","Population.totale")]
head(donnees)
```

Dans le dataframe obtenu, nous avions 961 communes pour un total T (calculé ci-dessous) de 5,2 millions d'habitants dans la région PACA. Pour faire le sondage aléatoire simple nécessaire afin d'estimer plus ou moins précisément la population dans la région PACA, il nous a fallu créer un échantillon aléatoire E de 100 communes, calculer le nombre moyen d'habitants dans cet échantillon afin de faire un intervalle de confiance (IDC) à 95% du nombre moyen d'habitants, estimer la taille de la population totale (T estimée) et faire un IDC pour cette population totale estimée.

```
#filtrage et calcul du nombre exact T d'habitants de la région PACA
donnees$Population.totale = as.numeric(gsub(" ","",donnees$Population.totale))
T = sum(donnees$Population.totale)
T
#création d'un échantillon(sample) de 100 communes
n = 100
E = sample(U,n)
#création d'une table données1 contenant les communes de l'échantillon aléatoire,leur département et leur nombre d'habitants
donnees1= donnees[donnees$Commune %in% E, ]
head(donnees1)
#calcul du nombre moyen d'habitants
xbar= mean(donnees1$Population.totale)
#calcul d'un IDC à 95% du nombre moyen d'habitants
idcmoy = t.test(donnees1$Population.totale)$conf.int
#valeur estimée du nombre d'habitants
T_est = N*xbar
T_est
#intervalle de confiance pour T
idcT = idcmoy*N
idcT
#marge d'erreur pour pour l'IDC de T
marge=(idcT[2]-idcT[1])/2
marge
```

Cette méthode fonctionne très bien, mais elle reste cependant très inefficace ; nous avons donc décidé d'automatiser le processus grâce à une boucle car il nous fallait répéter ce processus dix fois afin d'illustrer les résultats à l'aide d'un tableau Excel et d'un graphique représentant ce tableau.

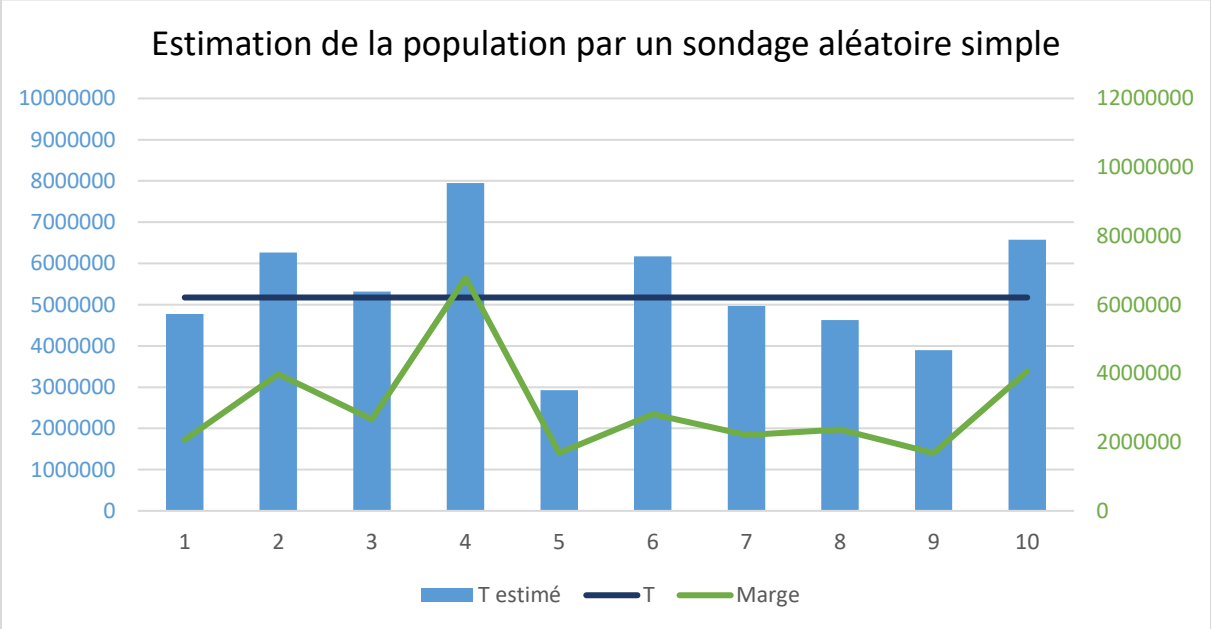
NB : vous trouverez le code complet de notre projet en fin de document

```r
for (i in 1:10) {
  E = sample(U, n)
  donnees1 = donnees[donnees$Commune %in% E, ]
  xbar = mean(donnees1$Population.totale)
  idcmoy = t.test(donnees1$Population.totale)$conf.int
  T_est = N * xbar
  idcT = idcmoy * N
  marge = (idcT[2] - idcT[1]) / 2

  resultats <- rbind(resultats, data.frame(
    T = T,
    T_est = T_est,
    Borne_inf = idcT[1],
    Borne_sup = idcT[2],
    Marge = marge
  ))
}
#création d'un fichier csv ouvrable sur excel comprenant les résultats des 10 itérations
write.csv2(resultats, file = "resultats_estimations.csv", row.names = FALSE)
```

La boucle ci-dessus répète donc le processus dix fois et mets immédiatement les résultats dans un fichier Excel que nous avons exploité afin d'obtenir le tableau et le graphique suivant :

	T	T estimé	IDC borne inférieure	IDC borne supérieure	Marge
1	5174034	4774686	2704731	6844640	2069954
2	5174034	6262637	2282918	10242356	3979719
3	5174034	5317004	2656549	7977459	2660455
4	5174034	7947937	1167238	14728635	6780699
5	5174034	2930108	1247603	4612613	1682505
6	5174034	6173651	3358585	8988716	2815066
7	5174034	4970121	2758426	7181815	2211694
8	5174034	4629821	2262925	6996717	2366896
9	5174034	3899193	2216743	5581642	1682450
10	5174034	6578297	2517119	10639475	4061178



En observant ce graphique, on peut voir que la méthode du sondage aléatoire simple n'est pas très précise. En effet, même si la première, troisième et septième estimation sont proches de la réalité, les autres en sont très loin, la plus grande différence entre la population réelle et la population estimée étant de presque trois millions d'habitants. De plus, la moyenne des estimations est de 5,4 millions d'habitants, ce qui est légèrement au-dessus de la réalité. Il est possible que l'on obtienne des résultats plus précis et proches de la réalité en utilisant une autre méthode de sondage, par exemple la méthode des sondages stratifiés.

Partie 1.2 : Estimation du nombre d'habitants d'une région de France - Échantillonnage aléatoire stratifié

Dans cette seconde partie sur l'estimation de la population totale de la région PACA, nous avons repris le travail fait à la partie 1.1 en adoptant un échantillonnage stratifié. Nous avons commencé par choisir comme strates les quartiles de la population totale de la région, que nous avons trouvé grâce à la fonction suivante :

```
summary(donnees$Population.totale)
```

Fort de notre expérience sur le sondage aléatoire simple, nous avons décidé de faire une boucle dès le départ. Cependant, il nous fallait au préalable préparer les données.

```
# Paramètres personnalisables
k <- 4              #nombre de strates
n <- 100            #taille de l'échantillon
#création des strates
bornes <- quantile(donnees$Population.totale, probs = seq(0, 1, length.out = k + 1), na.rm = TRUE)
#découpage la population en strates
donnees$strate <- cut(donnees$Population.totale, breaks = bornes, labels = 1:k, include.lowest = TRUE)
head(donnees)
# Préparation de la table ordonnée avec l'effectif des strates
datastrat <- donnees[, c("Commune", "Population.totale", "strate")]
data <- datastrat[order(datastrat$strate), ]
Nh <- table(data$strate)
N <- sum(Nh)
#poids des strates
gh <- Nh / N
#tirage d'un échantillon stratifié
nh <- round(n * Nh / N)
#taux de sondage dans les strates
fh <- nh / Nh
```

Comme le montre le code ci-dessus, nous avons créé 4 strates (avec l'option de personnaliser le nombre de strates pour de futurs tests), créé une table spécifiquement pour le sondage par strates, et tiré notre échantillon. Toutes ces étapes ont servi à préparer nos données au traitement que nous avons effectué grâce à la boucle sur la page ci-dessous :

```r
#création d'un dataframe qui servira à récolter les résultats
resultats_stratifie <- data.frame()
#on répète l'expérience dix fois grâce à une boucle
for (i in 1:10) {
  st <- strata(data, stratanames = c("strate"), size = nh, method = "srswr")
  data1 <- getdata(data, st)

  moyennes <- numeric(k)
  variances <- numeric(k)

  for (j in 1:k) {
    ech <- data1[data1$strate == j, ]
    moyennes[j] <- mean(ech$Population.totale)
    variances[j] <- var(ech$Population.totale)
  }

  # Estimation moyenne et variance
  Xbarst <- sum(Nh * moyennes) / N
  varXbarst <- sum((gh^2) * (1 - fh) * variances / nh)

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

  # Estimations totales
  Tstr <- Xbarst * N
  idcT <- idcmoy * N
  marge <- (idcT[2] - idcT[1]) / 2

  # Stockage des résultats
  resultats_stratifie <- rbind(resultats_stratifie, data.frame(
    T = T,
    T_est = Tstr,
    Borne_inf = idcT[1],
    Borne_sup = idcT[2],
    Marge = marge
  ))
}
```

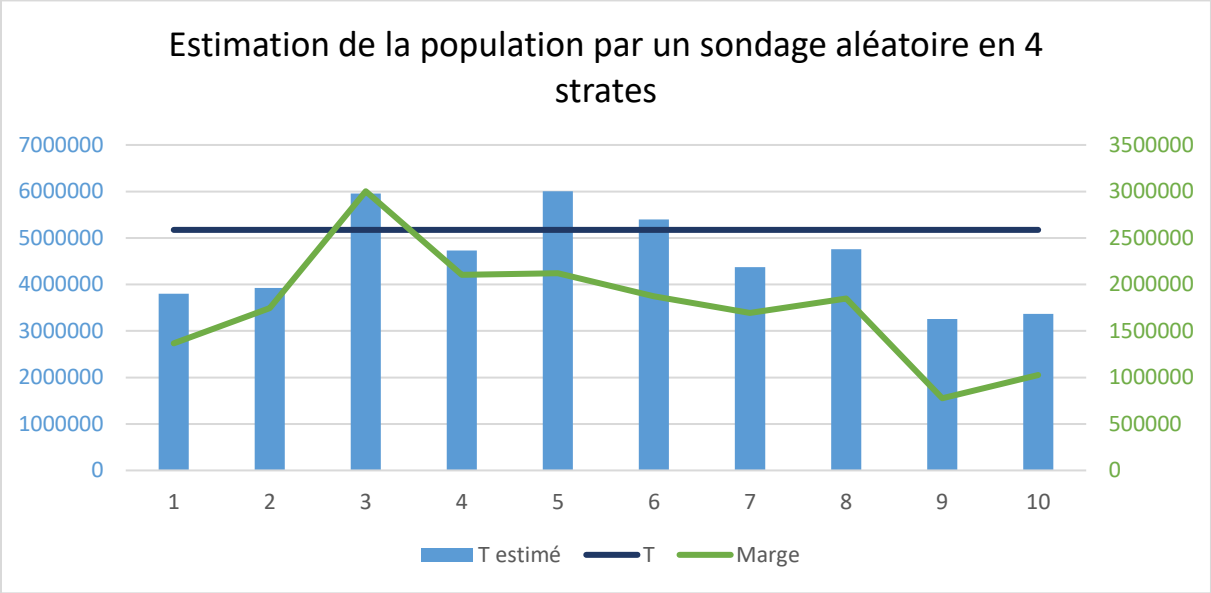
Cette boucle effectue une estimation répétée par sondage stratifié pour évaluer la population totale de la région PACA. Elle crée d'abord un dataframe vide pour stocker les résultats et répète ensuite dix fois l'expérience de sondage : à chaque itération, un échantillon est tiré de manière aléatoire avec remise dans chaque strate définie par la variable "strate". Pour chaque strate, la moyenne et la variance de la population totale sont calculées. Ces estimations sont ensuite combinées pour obtenir une moyenne stratifiée (Xbarst) et sa variance (varXbarst), à partir desquelles un intervalle de confiance à 95 % est construit pour l'estimation totale (Tstr). Les bornes de cet intervalle et la marge d'erreur sont calculées et enregistrées dans le tableau de résultats.

Une fois que la boucle est finie, les résultats sont compilés dans un fichier nommé « résultats_stratifiés » grâce à la fonction suivante :

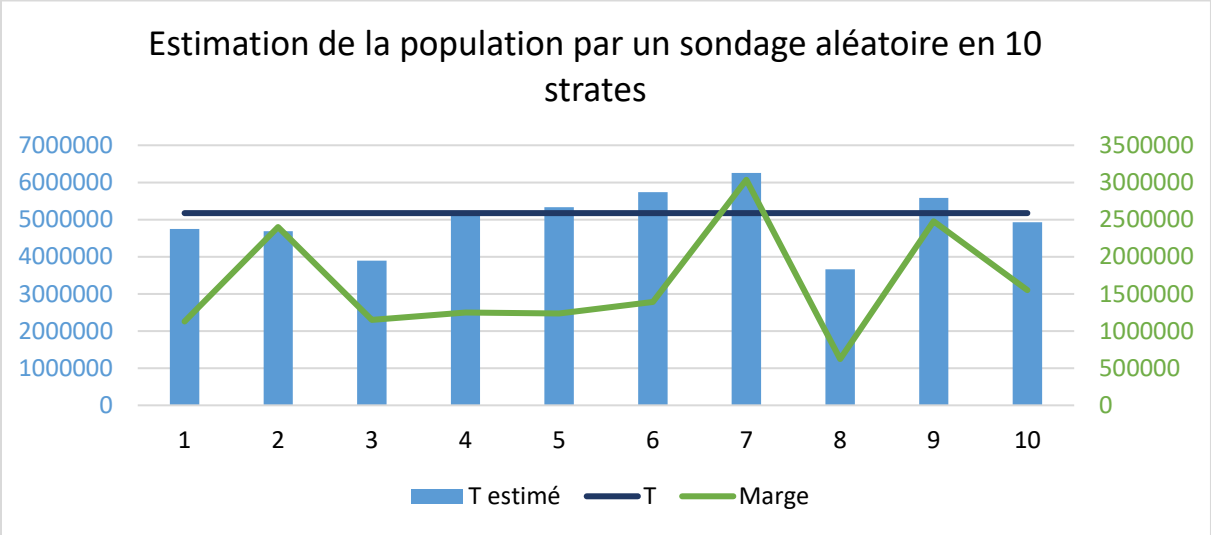
```r
# Export CSV
write.csv2(resultats_stratifie, file = paste0("resultats_stratifie_k", k, ".csv"), row.names = FALSE)
```

Ce document créé automatiquement est bien plus simple et plus rapide à exploiter, étant donné qu'il ne nécessite aucun copier-coller entre RStudio et Excel. De plus, il nous a permis de faire de nombreux tests. En effet, après avoir réalisé ce premier test avec 4 strates équivalentes aux quartiles, nous avons eu l'idée de faire ce même test avec un nombre de strates différent dans l'espoir de trouver des résultats différents voire même meilleurs et plus pertinents. Ces tests nous ont montré que la seule différence notoire au niveau de nos résultats n'apparaissait que quand nous effectuions des sondages aléatoires stratifiés à 10 strates, les tableaux et graphiques de ces deux options se trouvant ci-dessous :

Avec 4 strates					
	T	T estimé	IDC borne inférieure	IDC borne supérieure	Marge
1	5174034	3798568	2432686	5164450	1365882
2	5174034	3922240	2177799	5666682	1744441
3	5174034	5958120	2955731	8960509	3002389
4	5174034	4731604	2629070	6834138	2102534
5	5174034	6001307	3880359	8122255	2120948
6	5174034	5395194	3522378	7268011	1872816
7	5174034	4371269	2677539	6064998	1693730
8	5174034	4758239	2909093	6607385	1849146
9	5174034	3259943	2483822	4036065	776122
10	5174034	3366342	2341922	4390763	1024421



Avec 10 strates					
	T	T estimé	IDC borne inférieure	IDC borne supérieure	Marge
1	5174034	4747544	3620090	5874997	1127453
2	5174034	4688667	2290138	7087195	2398529
3	5174034	3896529	2746513	5046545	1150016
4	5174034	5199498	3952155	6446842	1247343
5	5174034	5332818	4095098	6570538	1237720
6	5174034	5735840	4344706	7126973	1391133
7	5174034	6251162	3215555	9286769	3035607
8	5174034	3663244	3040013	4286475	623231
9	5174034	5580958	3105225	8056692	2475734
10	5174034	4926045	3376793	6475297	1549252



En comparant les deux graphiques obtenus, on observe que les estimations les plus proches de la réalité (T estimé) ont été obtenues en faisant un sondage à 10 strates, même si la différence n'est pas des plus flagrantes. Dans le meilleur graphique, la plus grande différence entre la réalité et la population totale estimée est aux alentours de 1,4 millions d'habitants, ce qui est beaucoup moins que la différence de 3 millions du sondage aléatoire simple. De plus, la moyenne des estimations est de 5 millions d'habitants, ce qui est légèrement en dessous de la réalité mais qui reste meilleur que la moyenne de 5,4 millions d'habitants obtenue par le sondage aléatoire simple.

Dans l'ensemble, il reste clair que le sondage aléatoire stratifié nous permet d'obtenir de meilleurs résultats que le sondage aléatoire simple, même si notre constat est que l'on manque de précision. L'idée de changer le nombre de strates était bonne et nous a permis d'obtenir des estimations plus précises, mais trouver d'autres strates/ d'autres manières de les faire autrement qu'en se reposant sur les quantiles aurait aussi pu nous permettre d'obtenir des résultats intéressants.

Partie 2 : Traitement de données d'enquête

Lors de cette partie, nous avions comme données les réponses à l'enquête sur le sport de la SAÉ « Tableaux de données et analyse exploratoire » du premier semestre. Le but était de trouver des relations significatives entre le sport et plusieurs variables qualitatives présentes dans l'étude.

```
#################### -- Partie 2 -- ####################

tablesport = read.csv2("C:/Users/eguena01/OneDrive - Université de Poitiers/SEMESTRE 2/Stat inf/SAÉ/EnqueteSportEtudiant2024.csv",
                       sep = ";", dec = ";", header = TRUE)
head(tablesport)
```

Nous avons commencé par importer le fichier afin de d'explorer le fichier et de déterminer quelles variables pourraient être intéressantes à croiser. Étant donné que certaines questions du questionnaire étaient conditionnelles, c'est-à-dire qu'uniquement une partie des répondants y ont répondu, énormément de cases vides ou avec des NA se trouvaient dans les données et rendaient certaines variables qualitatives inutilisables. Après élimination de ces variables et études des variables restantes, nous avons décidé d'essayer de trouver des relations significatives entre la variables sport et les variables suivantes : le sexe, le fait d'être alternant ou non, le département de formation, le niveau d'études, le type de logement, le fait d'être fumeur ou non, le fait d'avoir une bonne qualité d'alimentation ou non et le fait d'être en bonne santé ou non.

```
# Tableaux croisés dynamiques entre le sport et chaque variable qualitative
TCD_Sexe = table(tablesport$sport, tablesport$sexe)
TCD_Alternant = table(tablesport$sport, tablesport$alternant)
TCD_Dept = table(tablesport$sport, tablesport$deptformation)
TCD_niveau = table(tablesport$sport, tablesport$niveau)
TCD_logement = table(tablesport$sport, tablesport$logement)
TCD_fumer = table(tablesport$sport, tablesport$fumer)
TCD_alimentation = table(tablesport$sport, tablesport$alimentation)
TCD_sante = table(tablesport$sport, tablesport$sante)
```

Afin de comparer chaque variable, nous avons créé des tableaux croisés dynamiques sur lesquels nous avons commencé par faire les tests du khi-deux d'indépendance ainsi que le test du V de Cramer avec le code suivant :

```
# Test du khi2 pour chaque variable/TCD
khideux_Sexe= chisq.test(TCD_Sexe)
khideux_Sexe #p-value = 0.0006292 --> relation significative
```

Nous avons répété ce test 8 fois pour chacune des variables qualitatives que nous avons décidé de croiser avec la variable sport, et de même pour le V de Cramer (exemple avec le croisement sport/sexe) :

```
# V de Cramer pour chaque variable
n<-dim(tablesport)[1]
p <- nrow(TCD_Sexe)
q <- ncol(TCD_Sexe)
m <- min(p-1, q-1)
V_Sexe =sqrt(khideux_Sexe$statistic/(n*m))
V_Sexe
```

Pour comparer nos résultats sur chaque croisement, nous avons pris les données obtenues et les avons mises dans un tableau Excel, dans lequel nous trouvons les données suivantes :

	Sexe	Alternant	Département	Niveau	Logement	Fumer	**Alimentation**	Sante
X^2	14.742	6.9227	18.777	12.668	4.8007	0.81111	16.661	0.70524
P-valeur	0.0006292	0.14	0.004557	0.1238	0.3084	0.6666	**0.000241**	0.7028
V de Cramer	0.198274	0.0960743	0.1582276	0.129963	0.08000623	0.04650774	**0.2107832**	0.04650774

En observant ces résultats, on peut voir que parmi les 8 p-valeurs obtenues, trois d'entre elles sont bien plus petites que les autres et nous indiquent qu'il y a une relation significative entre le sport et ces trois variables : le sexe, le département de formation ainsi que l'alimentation. Les autres p-valeurs sont bien trop grandes (certaines sont même très proches de 1) et constituent une preuve plus ou moins modérée envers l'hypothèse d'indépendance ; ces autres variables n'expliquent pas la variable sport.

En allant plus loin et en comparant le V de Cramer, les résultats obtenus sur ces trois mêmes variables sont là encore meilleurs que ceux des autres variables, cependant la seule liaison modérée et ainsi acceptable est la liaison entre la variable sport et la variable alimentation avec un V de Cramer à 0,21. Le V de Cramer obtenu sur la variable Sexe est proche, mais il reste entre 0,10 et 0,20 ce qui indique une liaison faible.

À l'issue de cette analyse exploratoire des données issues de l'enquête sur le sport, nous avons pu identifier certaines relations significatives entre la pratique sportive et des variables qualitatives précises. Grâce aux tests du khi-deux et au calcul du V de Cramer réalisés sur les croisements entre la variable « sport » et huit autres variables qualitatives, il ressort que seules trois d'entre elles — le sexe, le département de formation et la qualité de l'alimentation — présentent une relation statistiquement significative avec la pratique sportive.

Cependant, l'intensité de ces relations varie. Si le lien entre sport et alimentation s'avère modéré (V de Cramer ≈ 0,21), les relations avec le sexe et le département, bien que significatives, restent faibles. Les autres variables, telles que le niveau d'études, le statut d'alternant, le type de logement, le tabagisme ou la perception de la santé, ne montrent pas de lien notable avec la pratique sportive dans ce contexte.

Ces résultats mettent en lumière l'importance de certaines dimensions, notamment l'alimentation, dans la compréhension des comportements sportifs. Ils suggèrent également que d'autres facteurs, potentiellement non mesurés dans cette enquête, pourraient jouer un rôle déterminant. Une analyse plus poussée avec des données complémentaires permettrait d'enrichir cette première exploration.

```r
library(sampling)
setwd("C:/Users/eguena01/OneDrive - Université de Poitiers/SEMESTRE 2/Stat inf/SAÉ")
#import des données
table = read.csv2(file ="population_francaise_communes.csv",sep = ";",dec=",",header=TRUE)

#################### -- Partie 1.1 -- ####################
#filtrage des données sur la région PACA
donnees = table[table$Nom.de.la.région == "Provence-Alpes-Côte d'Azur",c("Code.département","Commune","Population.totale")]
head(donnees)

#ensemble des communes de la région
U = donnees$Commune
head(U)
N = length(U)

#filtrage et calcul du nombre exact T d'habitants de la région PACA
donnees$Population.totale = as.numeric(gsub(" ","",donnees$Population.totale))
T = sum(donnees$Population.totale)
T
#création d'un échantillon(sample) de 100 communes
n = 100
E = sample(U,n)
#création d'une table données1 contenant les communes de l'échantillon aléatoire,leur département et leur nombre d'habitants
donnees1= donnees[donnees$Commune %in% E, ]
head(donnees1)
#calcul du nombre moyen d'habitants
xbar= mean(donnees1$Population.totale)
#calcul d'un IDC à 95% du nombre moyen d'habitants
idcmoy = t.test(donnees1$Population.totale)$conf.int
#valeur estimée du nombre d'habitants
T_est = N*xbar
T_est
#intervalle de confiance pour T
idcT = idcmoy*N
idcT
#marge d'erreur pour pour l'IDC de T
marge=(idcT[2]-idcT[1])/2
marge

#création d'un dataframe qui servira à récolter les résultats pour plusieurs échantillons
#(ici dix échantillons)
resultats <- data.frame()
```

```r
#boucle qui permet d'automatiser les calculs présents au dessus pour un gain de temps
#pour 10 itérations
for (i in 1:10) {
  E = sample(U, n)
  donnees1 = donnees[donnees$Commune %in% E, ]
  xbar = mean(donnees1$Population.totale)
  idcmoy = t.test(donnees1$Population.totale)$conf.int
  T_est = N * xbar
  idcT = idcmoy * N
  marge = (idcT[2] - idcT[1]) / 2

  resultats <- rbind(resultats, data.frame(
    T = T,
    T_est = T_est,
    Borne_inf = idcT[1],
    Borne_sup = idcT[2],
    Marge = marge
  ))
}
#création d'un fichier csv ouvrable sur excel comprenant les résultats des 10 itérations
write.csv2(resultats, file = "resultats_estimations.csv", row.names = FALSE)
```

```r
#################### -- Partie 1.2 -- ####################
summary(donnees$Population.totale)

# Paramètres personnalisables
k <- 4              #nombre de strates
n <- 100            #taille de l'échantillon
#création des strates
bornes <- quantile(donnees$Population.totale, probs = seq(0, 1, length.out = k + 1), na.rm = TRUE)
#découpage la population en strates
donnees$strate <- cut(donnees$Population.totale, breaks = bornes, labels = 1:k, include.lowest = TRUE)
head(donnees)
# Préparation de la table ordonnée avec l'effectif des strates
datastrat <- donnees[, c("Commune", "Population.totale", "strate")]
data <- datastrat[order(datastrat$strate), ]
Nh <- table(data$strate)
N <- sum(Nh)
#poids des strates
gh <- Nh / N
#tirage d'un échantillon stratifié
nh <- round(n * Nh / N)
#taux de sondage dans les strates
fh <- nh / Nh
```

```r
#création d'un dataframe qui servira à récolter les résultats
resultats_stratifie <- data.frame()
#on répète l'expérience dix fois grâce à une boucle
for (i in 1:10) {
  st <- strata(data, stratanames = c("strate"), size = nh, method = "srswr")
  data1 <- getdata(data, st)

  moyennes <- numeric(k)
  variances <- numeric(k)

  for (j in 1:k) {
    ech <- data1[data1$strate == j, ]
    moyennes[j] <- mean(ech$Population.totale)
    variances[j] <- var(ech$Population.totale)
  }

  # Estimation moyenne et variance
  Xbarst <- sum(Nh * moyennes) / N
  varXbarst <- sum((gh^2) * (1 - fh) * variances / nh)

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

  # Estimations totales
  Tstr <- Xbarst * N
  idcT <- idcmoy * N
  marge <- (idcT[2] - idcT[1]) / 2

  # Stockage des résultats
  resultats_stratifie <- rbind(resultats_stratifie, data.frame(
    T = T,
    T_est = Tstr,
    Borne_inf = idcT[1],
    Borne_sup = idcT[2],
    Marge = marge
  ))
}

# Export CSV
write.csv2(resultats_stratifie, file = paste0("resultats_stratifie_k", k, ".csv"), row.names = FALSE)

resultats$Methode <- "Simple"
resultats_stratifie$Methode <- "Stratifie"
resultats_fusionnes <- rbind(resultats, resultats_stratifie)

#création du fichier contenant les réslutats des sondages simple et stratifés afin de faire les graphiques
write.csv2(resultats_fusionnes, file = "comparaison_methodes.csv", row.names = FALSE)
```

```
#################### -- Partie 2 -- ####################

tablesport = read.csv2("C:/Users/eguena01/OneDrive - Université de Poitiers/SEMESTRE 2/Stat inf/SAÉ/EnqueteSportEtudiant2024.csv",
                       sep = ";", dec = ";", header = TRUE)
head(tablesport)

#4 : on croise la variable sport avec 8 variables qualitatives
# Tableaux croisés dynamiques entre le sport et chaque variable qualitative
TCD_Sexe = table(tablesport$sport, tablesport$sexe)
TCD_Alternant = table(tablesport$sport, tablesport$alternant)
TCD_Dept = table(tablesport$sport, tablesport$deptformation)
TCD_niveau = table(tablesport$sport, tablesport$niveau)
TCD_logement = table(tablesport$sport, tablesport$logement)
TCD_fumer = table(tablesport$sport, tablesport$fumer)
TCD_alimentation = table(tablesport$sport, tablesport$alimentation)
TCD_sante = table(tablesport$sport, tablesport$sante)

TCD_Sexe
TCD_Alternant
TCD_Dept
TCD_niveau
TCD_logement
TCD_fumer
TCD_alimentation
TCD_sante
```

```
# Test du khi2 pour chaque variable/TCD
khideux_Sexe= chisq.test(TCD_Sexe)
khideux_Sexe #p-value = 0.0006292 --> relation significative

khideux_Alternant= chisq.test(TCD_Alternant)
khideux_Alternant #p-value = 0,14 --> preuve modérée contre l'hypothèse nulle, peut-être que c'est dû au hasard

khideux_Dept= chisq.test(TCD_Dept)
khideux_Dept #p-value = 0.004557 --> relation significative

khideux_niveau= chisq.test(TCD_niveau)
khideux_niveau #p-value = 0.1238 --> preuve modérée contre l'hypothèse nulle

khideux_logement= chisq.test(TCD_logement)
khideux_logement #p-value = 0.3084 --> preuve plus que modérée pour l'hypothèse nulle

khideux_fumer= chisq.test(TCD_fumer)
khideux_fumer #p-value = 0.6666 --> très élevé, il n'y a quasiment pas de lien

khideux_alimentation= chisq.test(TCD_alimentation)
khideux_alimentation #p-value = 0.000241 --> relation significative

khideux_sante= chisq.test(TCD_sante)
khideux_sante #p-value = 0.7028 --> très élevé, il n'y a quasiment pas de lien
```

```
# V de Cramer pour chaque variable
n<-dim(tablesport)[1]
p <- nrow(TCD_Sexe)
q <- ncol(TCD_Sexe)
m <- min(p-1, q-1)
V_Sexe =sqrt(khideux_Sexe$statistic/(n*m))
V_Sexe

n<-dim(tablesport)[1]
p <- nrow(TCD_alimentation)
q <- ncol(TCD_alimentation)
m <- min(p-1, q-1)
V_alimentation =sqrt(khideux_alimentation$statistic/(n*m))
V_alimentation

n<-dim(tablesport)[1]
p <- nrow(TCD_Dept)
q <- ncol(TCD_Dept)
m <- min(p-1, q-1)
V_Dept =sqrt(khideux_Dept$statistic/(n*m))
V_Dept

n<-dim(tablesport)[1]
p <- nrow(TCD_Alternant)
q <- ncol(TCD_Alternant)
m <- min(p-1, q-1)
V_Alternant =sqrt(khideux_Alternant$statistic/(n*m))
V_Alternant

n<-dim(tablesport)[1]
p <- nrow(TCD_niveau)
q <- ncol(TCD_niveau)
m <- min(p-1, q-1)
V_niveau =sqrt(khideux_niveau$statistic/(n*m))
V_niveau
```

```r
n<-dim(tablesport)[1]
p <- nrow(TCD_sante)
q <- ncol(TCD_sante)
m <- min(p-1, q-1)
V_sante =sqrt(khideux_sante$statistic/(n*m))
V_sante
```