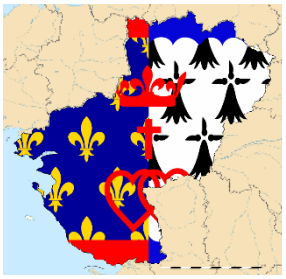
Nathan Puaud, Titouan Leblanc--Floch



SAE – Échantillonnage et Estimation

Population des Pays de la Loire

Partie 1 : Estimation du nombre d'habitants

Dans le cadre de cette SAÉ, nous avons tenté d'estimer la population de la région Pays de la Loire en utilisant deux méthodologies de sondage aléatoire : le sondage aléatoire simple et le sondage stratifié.

Les résultats obtenus via le sondage aléatoire simple sont présentés sur un graphique en fin de rapport, en comparaison avec la population réelle de la région, estimée à 3,9 millions d'habitants. On observe que la population estimée varie fortement selon les tirages, avec un minimum de 1,3 million et un maximum de 5,5 millions d'habitants. La marge d'erreur est très élevée, atteignant parfois un niveau proche de la population réelle, ce qui limite la fiabilité de cette méthode.

Cependant, lors de deux tirages, l'estimation s'est révélée très proche de la population réelle. De manière générale, la majorité des estimations présentent un écart de 500 000 à 1 000 000 d'habitants par rapport à la valeur réelle.

```r
# -------------------------
# Initialisation
# -------------------------

library(sampling)  # Chargement du package pour les fonctions d'échantillonnage

# Chargement du fichier CSV contenant les données sur les communes françaises
Commune_fr = read.csv2("population_francaise_communes.csv", sep=";", dec=",", header=TRUE)

# -------------------------
# Conception et nettoyage
# -------------------------

Commune_fr <- Commune_fr[-(1:6), ]  # Suppression des 6 premières lignes (souvent des en-têtes ou commentaires)
Commune_fr <- Commune_fr[, -9]      # Suppression de la colonne 9 (inutile pour l'analyse)
Commune_fr <- Commune_fr[, -10]     # Suppression de la colonne 10

colnames(Commune_fr) <- Commune_fr[1, ]  # Redéfinition des noms de colonnes à partir de la 1ère ligne restante
Commune_fr <- Commune_fr[-1, ]           # Suppression de cette ligne qui sert maintenant d'en-tête

# -------------------------
# Partie 1.1 : Échantillonnage aléatoire simple
# -------------------------

# Filtrer les données pour ne garder que la région "Pays de la Loire"
donnees <- Commune_fr[Commune_fr$`Nom de la région` == "Pays de la Loire", ]
donnees <- donnees[, c(3, 7, 9)]  # Sélection des colonnes d'intérêt

# Nettoyage de la colonne "Population totale"
donnees$`Population totale` <- gsub(" ", "", donnees$`Population totale`)  # Supprimer les espaces
donnees$`Population totale` <- as.numeric(donnees$`Population totale`)     # Conversion en numérique

head(donnees)  # Affichage des premières lignes

U = donnees$Commune  # Liste des communes
head(U)
```

Voici le code utilisé pour installer la librairie sampling ainsi que pour lire et trier les données nécessaires à l'analyse.

```r
N = length(U)  # Taille de la population
print(N)

T = sum(donnees$`Population totale`)  # Somme totale des populations
print(T)

n = 100  # Taille de l'échantillon
E = sample(U, n)  # Tirage aléatoire simple
head(E)

donnees1 = donnees[donnees$Commune %in% E, ]  # Sous-échantillon
head(donnees1)

xbar = mean(donnees1$`Population totale`)  # Moyenne estimée
print(xbar)

idcmoy = t.test(donnees1$`Population totale`)$conf.int  # Intervalle de confiance
print(idcmoy)

T_test = N * xbar  # Estimation du total
print(T_test)

idcT = idcmoy * N  # Intervalle de confiance pour le total
print(idcT)

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

La méthode utilisée consiste à effectuer une estimation de la population à partir d'un échantillon de 100 villes tirées aléatoirement dans la région des Pays de la Loire. Pour chaque tirage, la population estimée, l'intervalle de confiance (IDC) et la marge d'erreur ont ensuite été calculés.

Les résultats obtenus via le sondage aléatoire stratifié sont présentés dans le graphique en fin de rapport, montrant la population estimée, la population réelle (3,9 millions d'habitants) et la marge d'erreur associée.

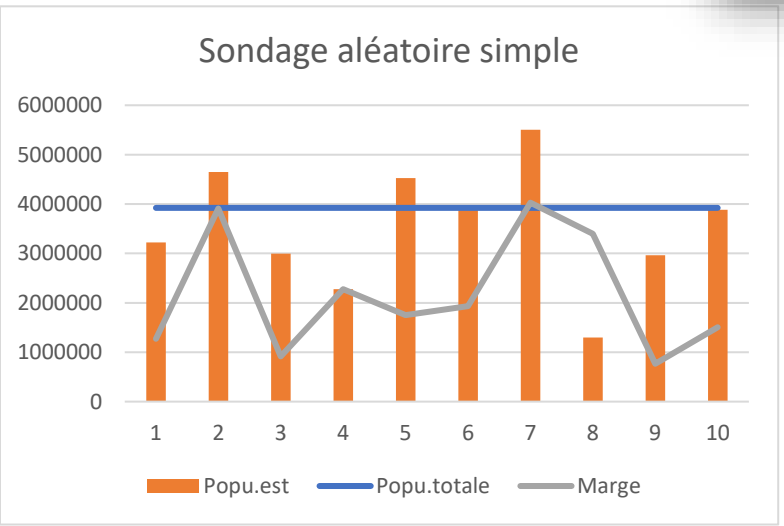
La population estimée maximale atteint 4,3 millions d'habitants, avec seulement deux sondages dépassant les 4 millions. Le minimum observé est de 2,8 millions d'habitants. On constate que la marge d'erreur est, dans l'ensemble, nettement plus faible que celle obtenue avec le sondage aléatoire simple.

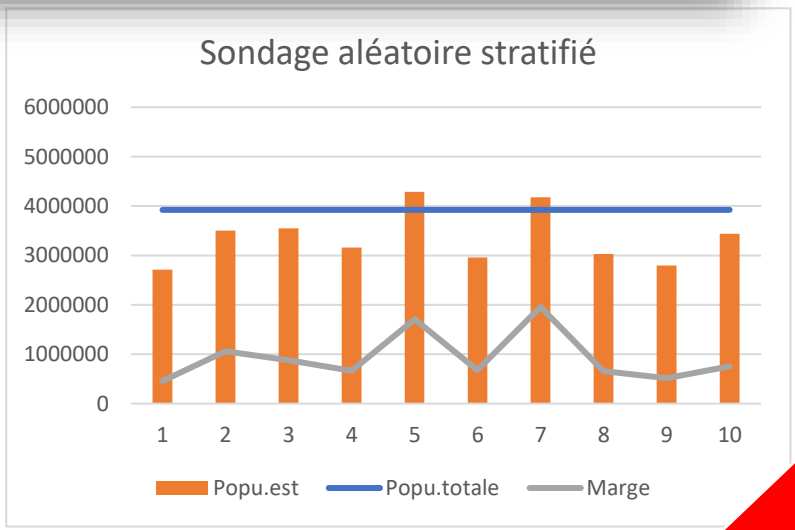
```r
# -------------------------
# Partie 1.2 : Échantillonnage aléatoire stratifié
# -------------------------

summary(donnees$`Population totale`)  # Statistiques descriptives

# Création des strates
donnees$strate = cut(donnees$`Population totale`, breaks=c(0, 525, 1173, 2719, 325857), labels=c(1, 2, 3, 4))
datastrat = donnees[, c("Commune", "Population totale", "strate")]
head(datastrat)

data = datastrat[order(donnees$strate), ]
head(data)

Nh = table(data$strate)  # Taille de chaque strate
print(Nh)

N = sum(Nh)
print(N)

gh = Nh / N  # Proportions de chaque strate
print(gh)

n = 100
nh = c(25, 25, 25, 25)  # Taille d'échantillon égale par strate
print(nh)

fh = nh / Nh  # Taux de sondage
print(fh)

# Tirage aléatoire stratifié
st = strata(data, stratanames = c("strate"), size = nh, method = "srswr")
data1 = getdata(data, st)
head(data1)
print(length(data1$Commune))  # Taille totale de l'échantillon

# Extraction des sous-échantillons
ech1 = data1[data1$strate == 1, ]
ech2 = data1[data1$strate == 2, ]
ech3 = data1[data1$strate == 3, ]
ech4 = data1[data1$strate == 4, ]
```

En moyenne, l'écart entre la population réelle et la population estimée est significativement réduit. La plupart des estimations se situent entre 200 000 et 600 000 habitants d'écart par rapport à la valeur réelle.

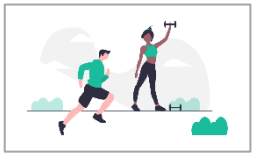
```r
# Moyennes par strate
m1 = mean(ech1$`Population totale`)
m2 = mean(ech2$`Population totale`)
m3 = mean(ech3$`Population totale`)
m4 = mean(ech4$`Population totale`)
print(m1); print(m2); print(m3); print(m4)

# Variances par strate
var1 = var(ech1$`Population totale`)
var2 = var(ech2$`Population totale`)
var3 = var(ech3$`Population totale`)
var4 = var(ech4$`Population totale`)
print(var1); print(var2); print(var3); print(var4)

# Moyenne stratifiée estimée
xbarst = (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4) / N
print(xbarst)

# Variance de l'estimateur de moyenne
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])
print(varXbarst)

# Intervalle de confiance pour la moyenne
alpha = 0.05
binf = xbarst - qnorm(1 - alpha/2) * sqrt(varXbarst)
bsup = xbarst + qnorm(1 - alpha/2) * sqrt(varXbarst)
idcmoy = c(binf, bsup)
print(idcmoy)

# Estimation du total
Tstr = N * xbarst
print(Tstr)

# Intervalle de confiance pour le total
idcT = c(idcmoy[1]*N, idcmoy[2]*N)
print(idcT)

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

Cette seconde partie du code permet de définir les différentes strates à partir des quartiles, puis de calculer la population estimée, l'intervalle de confiance (IDC) ainsi que la marge d'erreur pour chaque tirage.





Le sondage aléatoire simple ne fonctionne correctement que lorsque la population des villes est homogène. Or, dans notre cas, elle est très hétérogène, ce qui rend cette méthode peu adaptée. Le sondage aléatoire stratifié est donc plus pertinent. Cette SAÉ nous a permis de mieux utiliser le logiciel R pour appliquer des techniques d'échantillonnage, et de comprendre que la méthodologie choisie influence fortement les résultats obtenus.



Partie 2 : Traitement des données d'une enquête

Dans cette seconde partie, nous travaillons à partir d'un fichier Excel issu d'une enquête sur les habitudes sportives des étudiants.

Voici le tableau présentant les variables qualitatives que nous avons sélectionnées (Sexe, Fumeur, Santé, Fan de sport et Logement), accompagné des résultats des tests statistiques réalisés avec R. Parmi ces variables, seuls les tests portant sur le sexe et sur le fait d'être fan de sport se révèlent statistiquement significatifs.

Retour des valeurs trouvées sur l'enquête Sport

	Sexe	Fumer	Sante	Fan	logement
khideux	14.742	0.81111	0.70524	78.823	4.8007
p-valeur	0.0006292	0.6666	0.7028	< 2.2e-16	0.3084
V-Cramer	0.198274			0.4584704	

Voici le script R commenté correspondant à l'ensemble de la deuxième partie. Il comprend la création de tableaux croisés dynamiques, ainsi que les différents calculs liés au test du Khi² et au V de Cramer pour analyser les relations entre les variables qualitatives.

```r
# -------------------------
# Partie 2 : Traitement de données d'enquête
# -------------------------

Sport <- read.csv2("EnqueteSportEtudiant2024.csv", sep = ";", dec = ",", header = TRUE)
head(Sport)  # Aperçu


# Cette base contient :
# - Des **individus** : chaque ligne représente un étudiant ayant répondu à l'enquête
# - Des **variables** : chaque colonne est une question ou caractéristique (ex : sexe, fumer, santé, logement, sport…)
# - Types de variables :
# - **Catégorielles** (qualitatives) : sexe, fumer, sante, logement, fan, sport
#    => ce sont des réponses sous forme de catégories ou modalités (ex: "Oui", "Non", "Homme", "Femme", etc.)
# - **Possiblement numériques** si d'autres variables (non affichées ici) contiennent des quantités (ex : âge, nombre d'heures de sport, etc.)

# Création des tableaux croisés
TCD_Sexe <- table(Sport$sport, Sport$sexe)
TCD_Fumer <- table(Sport$sport, Sport$fumer)
TCD_Sante <- table(Sport$sport, Sport$sante)
TCD_Fan <- table(Sport$sport, Sport$fan)
TCD_Logement <- table(Sport$sport, Sport$logement)

# Affichage des tableaux croisés
print(TCD_Sexe)
print(TCD_Fumer)
print(TCD_Sante)
print(TCD_Fan)
print(TCD_Logement)
```

```r
# Tests du khi-deux
khideux_Sexe = chisq.test(TCD_Sexe)
print(khideux_Sexe)

khideux_Fumer = chisq.test(TCD_Fumer)
print(khideux_Fumer)

khideux_Sante = chisq.test(TCD_Sante)
print(khideux_Sante)

khideux_Fan = chisq.test(TCD_Fan)
print(khideux_Fan)

khideux_Logement = chisq.test(TCD_Logement)
print(khideux_Logement)

# V de Cramer (Sexe)
n <- dim(Sport)[1]
p <- nrow(TCD_Sexe)
q <- ncol(TCD_Sexe)
m <- min(p - 1, q - 1)
V_Sexe = sqrt(khideux_Sexe$statistic / (n * m))
print(V_Sexe)

# V de Cramer (Fan de sport)
p <- nrow(TCD_Fan)
q <- ncol(TCD_Fan)
m <- min(p - 1, q - 1)
V_Fan = sqrt(khideux_Fan$statistic / (n * m))
print(V_Fan)
```

Pour conclure, les résultats montrent qu'il existe un lien significatif entre la pratique sportive, le sexe et le fait d'être fan de sport. Le V de Cramer obtenu pour la relation entre la pratique sportive et le sexe est de 0.198, indiquant une association faible. En revanche, la relation entre la pratique sportive et le fait d'être fan présente un V de Cramer de 0.458, ce qui traduit une association modérée à forte.