# ESTIMATION DE LA POPULATION DE LA NOUVELLE AQUITAINE

**Octave Romer, Mathieu Lopes Moreira**

Université de Poitiers | Campus de Niort

IUT Poitiers Niort Châtellerault

BUT Science des données Niort

La Nouvelle-Aquitaine, située dans le sud-ouest de la France, est la région la plus vaste du pays en termes de superficie. Mais qu'en est-il de sa population ? Avec 6 171 721 habitants, elle se classe au 3e rang des régions françaises les plus peuplées. Dans ce rapport, nous chercherons à estimer cette population dans une première partie à l'aide d'échantillons de villes. Pour ce faire, nous utiliserons deux méthodes : d'abord un échantillon aléatoire simple de taille 100, puis un échantillon aléatoire stratifié, également de taille 100. Pour finir, dans une deuxième partie, nous allons chercher à identifier des relations significatives dans la table EnqueteSportEtudiant2024 à l'aide d'un test du khi-deux. Cette table reprend les données d'enquête sur les étudiants et leur pratique du sport, déjà traitées dans la SAÉ *Tableaux de données et analyse exploratoire* du semestre 1.

# Partie 1 : Estimation du nombre d'habitants de la Nouvelle Aquitaine.

## 1. Echantillonnage aléatoire simple

```r
setwd('C:\Users\Romer Octave\OneDrive - Université de Poitiers\But sd\s2\SAE\ibazizen')

# Lire le fichier CSV
table <- read.csv2("population_francaise_communes.csv", sep=";", dec=",", header=TRUE)

# Filtrer les données pour la Nouvelle-Aquitaine
donnees <- table[table$Code.région == "75", c("Code.département", "Commune", "Population.totale")]

library(sampling)


# Afficher les 6 premières lignes de la table
head(donnees)
```

Tout d'abord, on définit le répertoire de travail, puis on lit le fichier CSV des communes françaises. Ensuite, on filtre les données pour ne conserver que celles correspondant à la région Nouvelle-Aquitaine, en sélectionnant les colonnes d'intérêt : "Code.département", "Commune" et "Population.totale". La bibliothèque sampling est ensuite chargée pour permettre l'échantillonnage que nous allons effectuer par la suite. Enfin, on affiche les six premières lignes de la table obtenue.

```r
# Créer la table U
U <- donnees$Commune
head(U)

# Calculer le nombre total de communes
N <- length(U)

# Afficher le nombre total de communes
print(paste("Le nombre total de communes dans la région est :", N))

# Supprimer les espaces dans la colonne "Population.totale" et convertir en numérique
donnees$Population.totale <- as.numeric(gsub(" ", "", donnees$Population.totale))

# Calculer le nombre total d'habitants
T <- sum(donnees$Population.totale, na.rm = TRUE)

# Afficher le nombre total d'habitants
print(paste("Le nombre total d'habitants dans la région est :", T))
```

On commence par créer une table contenant la liste des communes de la Nouvelle-Aquitaine, puis on calcule et affiche leur nombre total. Ensuite, on nettoie la colonne "Population.totale" pour la convertir en numérique, ce qui permet de calculer le nombre total d'habitants dans la région, également affiché à la fin.

On effectue un tirage aléatoire simple d'un échantillon de taille n = 100 communes à partir de la liste complète des communes U. Les six premières communes tirées sont affichées.

Ensuite, on crée une nouvelle table donnees1 qui contient uniquement les communes sélectionnées, avec leurs codes départementaux et leur population totale. On affiche les six premières lignes de cette table.

On calcule la moyenne de la population totale dans cet échantillon (xbar), puis on calcule l'intervalle de confiance à 95 % pour cette moyenne à l'aide d'un test t.

```r
# Tirage aléatoire simple d'un échantillon de taille n
n <- 100
E <- sample(U, n)
head(E)  # Afficher les 6 premières valeurs de l'échantillon tiré

# Créer une nouvelle table "donnees1" contenant les communes sélectionnées,
# leur département et leur nombre d'habitants
donnees1 <- donnees[donnees$Commune %in% E, ]

# Afficher les 6 premières lignes de la table "donnees1"
head(donnees1)

# Calculer la moyenne de la population totale dans l'échantillon
xbar <- mean(donnees1$Population.totale)
xbar  # Afficher la moyenne

# Calculer l'intervalle de confiance à 95 % pour la moyenne
idcmoy <- t.test(donnees1$Population.totale)$conf.int
idcmoy  # Afficher l'intervalle de confiance

# Estimation du nombre total d'habitants dans la population totale
T_est <- N * xbar
T_est  # Afficher l'estimation du total

# Calculer l'intervalle de confiance à 95 % pour le total
idcT <- idcmoy * N
idcT  # Afficher l'intervalle de confiance pour le total

# Calcul de la marge d'erreur de l'estimation du total
marge <- (idcT[2] - idcT[1]) / 2
marge  # Afficher la marge d'erreur
```

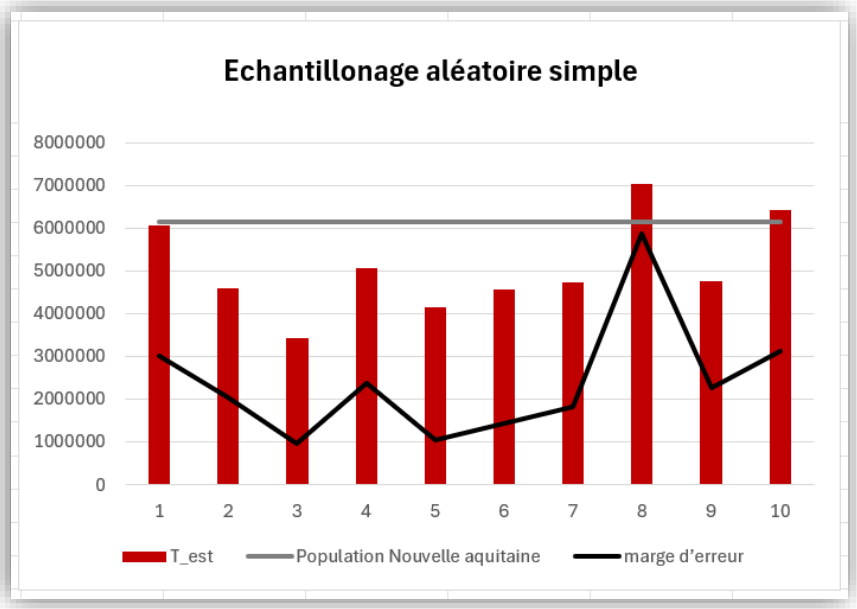
En multipliant la moyenne par le nombre total de communes N, on obtient une estimation du nombre total d'habitants dans la région (T_est). On calcule également l'intervalle de confiance à 95 % pour cette estimation en multipliant les bornes de l'intervalle de confiance de la moyenne par N.

Enfin, on calcule la marge d'erreur de cette estimation du total, qui correspond à la moitié de la largeur de l'intervalle de confiance, et on l'affiche.

Après avoir réalisé l'échantillonnage 10 fois voici nos résultats illustrés par ce tableau et ce graphique :

| Population Nouvelle aquitaine | T_est | IDCT | marge d'erreur |
|---|---|---|---|
| 6171721 | 6071666 | [3052885 ; 9090446] | 3018781 |
| 6171721 | 4595833 | [2547717 ; 6643949] | 2048116 |
| 6171721 | 3452200 | [248420 ; 20200] | 968000 |
| 6171721 | 5083463 | [2707381 ; 7459545] | 2376082 |
| 6171721 | 4167310 | [3098592 ; 5236028] | 1068718 |
| 6171721 | 4567461 | [3110673 ; 6024249] | 1456788 |
| 6171721 | 4740193 | [2909754 ; 6570632] | 1830439 |
| 6171721 | 7040329 | [1147072 ;12933586] | 5893257 |
| 6171721 | 4779883 | [2512181 ; 7047584] | 2267701 |
| 6171721 | 6450268 | [3304936 ; 9595600] | 3145332 |

Cette méthode d'échantillonnage reste assez approximative, puisque les estimations varient entre un maximum de 7 millions d'habitants et un minimum de 3,4 millions. Par ailleurs, l'analyse de la courbe de tendance des marges montre qu'elle n'est pas très linéaire et est influencée par plusieurs petites valeurs, ce qui peut s'expliquer par le tirage aléatoire simple où 100 petites villes ont pu être sélectionnées, ce qui ne reflète pas correctement la population réelle. Voyons maintenant si l'échantillonnage aléatoire stratifié permet d'obtenir des résultats plus fiables.



## 2. Echantillonnage aléatoire stratifié

On commence par utiliser la fonction summary() sur la variable "Population.totale", ce qui permet d'obtenir des statistiques descriptives, notamment les quintiles (minimum, 1er quartile, médiane, 3e quartile, maximum). Ces informations servent de base pour définir les bornes des strates.

À partir de ces valeurs, on crée une nouvelle colonne "strate" dans la table donnees, en regroupant les communes selon leur population totale. Quatre strates sont définies en fonction des classes de population : 0–250, 251–500, 501–1 000, et plus de 1 000 habitants.

```r
# Quintile de la variable "Population.totale"
summary(donnees$Population.totale)

# Création d'une nouvelle colonne "strate" selon les classes de population
donnees$strate <- cut(
  donnees$Population.totale,
  breaks = c(0, 250, 500, 1000, Inf),  # Définition des bornes des strates
  labels = c("Strate 1", "Strate 2", "Strate 3", "Strate 4"),  # Noms des strates
  include.lowest = TRUE  # Inclure la valeur minimale dans le premier intervalle
)

# Création d'une nouvelle table avec les colonnes utiles : "Commune", "Population.totale" et "strate"
donneesstrat <- donnees[, c("Commune", "Population.totale", "strate")]

# Affichage des 6 premières lignes de la table "donneesstrat"
head(donneesstrat)
#Question 3

# Trier les données par strate
data <- donneesstrat[order(donneesstrat$strate), ]
head(data)
```

Puis, on crée une nouvelle table donneesstrat contenant uniquement les colonnes utiles pour la suite : le nom de la commune, sa population totale, et la strate à laquelle elle appartient. Enfin, on affiche les six premières lignes de cette table pour vérifier sa structure.

On commence par calculer le nombre de communes dans chaque strate, puis on en déduit leur poids dans la population totale. Ensuite, on détermine combien de communes on doit tirer dans chaque strate pour obtenir un échantillon de taille 100, en respectant la répartition proportionnelle. Un petit ajustement est fait si la somme n'est pas exactement égale à 100. On calcule ensuite le taux de sondage dans chaque strate. Enfin, on effectue un tirage aléatoire stratifié sans remise à partir des strates définies, et on récupère l'échantillon final dans une nouvelle table, dont on vérifie les premières lignes ainsi que le nombre total d'observations.

```r
# Effectifs par strate
Nh <- table(data$strate)
Nh

# Taille totale de la population
N <- sum(Nh)
N

# Poids des strates
gh <- Nh / N
gh

# Taille de l'échantillon total
n <- 100

# Effectifs proportionnels à tirer par strate
nh <- round(c(n * Nh[1]/N, n * Nh[2]/N, n * Nh[3]/N, n * Nh[4]/N))

# Ajustement si la somme != 100
diff_n <- n - sum(nh)
if (diff_n != 0) {
  nh[which.max(nh)] <- nh[which.max(nh)] + diff_n
}
nh

# Taux de sondage dans chaque strate
fh <- nh / Nh
fh

# Tirage de l'échantillon stratifié sans remise
st <- strata(data, stratanames = c("strate"), size = nh, method = "srswor")
data1 <- getdata(data, st)

# Afficher les premières lignes et taille de l'échantillon
head(data1)
length(data1$Commune)  # Devrait être 100
```

On commence par séparer l'échantillon stratifié en quatre sous-échantillons, un pour chaque strate.

```r
# Séparation de l'échantillon selon les strates
ech1 <- data1[data1$Stratum == 1, ]  # Sous-échantillon de la strate 1
ech2 <- data1[data1$Stratum == 2, ]  # Sous-échantillon de la strate 2
ech3 <- data1[data1$Stratum == 3, ]  # Sous-échantillon de la strate 3
ech4 <- data1[data1$Stratum == 4, ]  # Sous-échantillon de la strate 4

# Moyennes des 4 sous-échantillons
m1 <- mean(ech1$Population.totale)  # Moyenne pour la strate 1
m2 <- mean(ech2$Population.totale)  # Moyenne pour la strate 2
m3 <- mean(ech3$Population.totale)  # Moyenne pour la strate 3
m4 <- mean(ech4$Population.totale)  # Moyenne pour la strate 4

# Variances des 4 sous-échantillons
var1 <- var(ech1$Population.totale)  # Variance pour la strate 1
var2 <- var(ech2$Population.totale)  # Variance pour la strate 2
var3 <- var(ech3$Population.totale)  # Variance pour la strate 3
var4 <- var(ech4$Population.totale)  # Variance pour la strate 4
```

Ensuite, on calcule la moyenne de la population totale dans chaque strate, puis la variance de cette même variable dans chaque groupe. Ces valeurs serviront à estimer plus précisément la moyenne et le total pour l'ensemble de la population.

```r
# Moyenne stratifiée pondérée des 4 sous-échantillons
Xbarst <- (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4) / N

# Estimation de la variance de la moyenne stratifiée (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]

# Intervalle de confiance à 95 % pour la moyenne stratifiée
alpha <- 0.05
binf <- Xbarst - qnorm(1 - alpha / 2) * sqrt(varXbarst)
bsup <- Xbarst + qnorm(1 - alpha / 2) * sqrt(varXbarst)
idcmoy <- c(binf, bsup)  # Intervalle de confiance pour la moyenne

# Estimation du total de la population à partir de la moyenne stratifiée
Tstr <- N * Xbarst
Tstr  # Affiche l'estimation du total

# Intervalle de confiance pour le total de la population
binf <- idcmoy[1] * N
bsup <- idcmoy[2] * N
idcT <- c(binf, bsup)
idcT  # Affiche l'intervalle de confiance pour le total

# Marge d'erreur de l'estimation du total
marge <- (idcT[2] - idcT[1]) / 2
marge  # Affiche la marge d'erreur
```

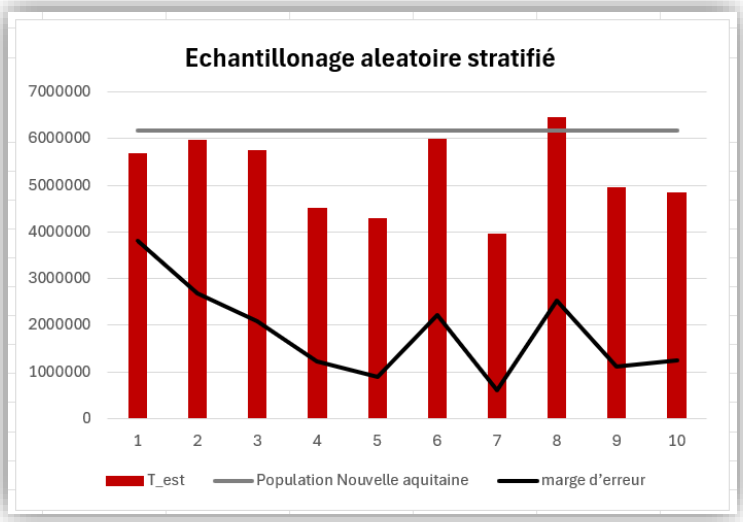
On commence par calculer la moyenne stratifiée pondérée, en tenant compte du poids de chaque strate dans la population. Ensuite, on estime la variance de cette moyenne, en intégrant les taux de sondage, les variances par strate, et la taille des sous-échantillons.

À partir de cela, on calcule l'intervalle de confiance à 95 % pour la moyenne stratifiée, puis on estime le total de la population en multipliant la moyenne par la taille totale de la population. L'intervalle de confiance pour ce total est obtenu en multipliant les bornes de l'intervalle de la moyenne par N.

Enfin, on calcule la marge d'erreur de cette estimation du total, en prenant la moitié de la largeur de l'intervalle de confiance.

Après avoir réalisé l'échantillonnage 10 fois voici nos résultats illustrés par ce tableau et ce graphique :

| Population Nouvelle aquitaine | T_est | IDCT | marge d'erreur |
|---|---|---|---|
| 6171721 | 5677890 | [1878831 ; 9476949] | 3799059 |
| 6171721 | 5972676 | [3295025 ; 8650327] | 2677651 |
| 6171721 | 5762758 | [3687428 ; 7838088] | 2075330 |
| 6171721 | 4509593 | [3289990 ; 5729196] | 1219603 |
| 6171721 | 4285119 | [3383919 ; 5186320] | 901200 |
| 6171721 | 5993802 | [3764965 ; 8222639] | 2228837 |
| 6171721 | 3958995 | [3344317 ; 4573673] | 614678 |
| 6171721 | 6460518 | [3936861 ; 8984174] | 2523657 |
| 6171721 | 4951481 | [3830416 ; 6072546] | 1121065 |
| 6171721 | 4856895 | [3610981 ; 6102810] | 1245914 |




En conclusion, l'échantillonnage aléatoire stratifié est meilleur que l'échantillonnage aléatoire simple, comme on l'a vu avec les 10 tirages. En divisant la population en groupes similaires (appelés strates) et en tirant au hasard dans chaque groupe, cette méthode permet de choisir un échantillon plus précis et représentatif.

Cela aide à mieux refléter la population réelle, à éviter les résultats trop éloignés de la réalité, et à obtenir des estimations plus fiables.

Donc, pour faire des tirages dans une population, il vaut mieux utiliser l'échantillonnage stratifié. Mais il faut bien choisir comment créer les groupes, car cela a un impact important sur la qualité des résultats. Par exemple, utiliser seulement 4 strates avec de larges intervalles peut cacher des différences importantes à l'intérieur de chaque groupe. Pour améliorer la précision, il serait préférable de créer 6 à 8 strates, ce qui permettrait de mieux capter la variabilité interne et d'obtenir des estimations encore plus précises.

# Partie 2 : Traitement de données d'une enquête

Le fichier EnqueteSportEtudiant2024.csv contient les réponses de plusieurs étudiants à une enquête sur leurs habitudes de vie. Chaque ligne représente un individu, et chaque colonne une variable. On y trouve des variables qualitatives (sexe, sport pratiqué, motivation, niveau d'étude, etc.) ainsi que des variables quantitatives (âge, taille, poids, heures de sommeil, fréquence du sport…). Cette base de données comprend au total 76 variables et 376 lignes. La variable qui nous intéresse particulièrement est « sport ». Nous cherchons à déterminer s'il existe une relation d'indépendance entre cette variable « sport » et les autres variables du fichier.

Autrement dit, nous voulons savoir si certaines variables dépendent de la pratique du sport. Logiquement, on s'attend à ce qu'il y ait des dépendances, mais il s'agit d'identifier lesquelles précisément. Pour cela, nous allons réaliser un test du khi-deux d'indépendance afin de détecter des relations significatives, puis utiliser le V de Cramer pour mesurer la force de ces liens. La p-value obtenue lors du test nous permettra de savoir si les relations observées sont statistiquement significatives ou non.

```r
# Lire le fichier CSV
data  <- read.csv2("EnqueteSportEtudiant2024.csv", sep=";", dec=",", header=TRUE)

head(data)

# Sexe vs Sport
table_sexe_sport <- table(data$sport, data$sexe)
print(table_sexe_sport)

# Niveau d'étude vs Sport
table_niv_sport <- table(data$sport, data$niveau)
print(table_niv_sport)


# Fumeur vs Sport
table_fumeur_sport <- table(data$sport, data$fumer)
print(table_fumeur_sport)


# Sante vs Sport
table_transport_sport <- table(data$sport, data$sante)
print(table_transport_sport)
```

Le code commence par lire le fichier EnqueteSportEtudiant2024.csv avec read.csv2, en utilisant le point-virgule comme séparateur et la virgule comme séparateur décimal. Il affiche ensuite les premières lignes du tableau pour avoir un aperçu des données.

Puis, il crée plusieurs tableaux de contingence qui comptent le nombre d'observations (effectifs) pour chaque combinaison entre la variable sport et d'autres variables qualitatives :

- table_sexe_sport montre le nombre d'étudiants selon leur pratique du sport et leur sexe.
- table_niv_sport affiche le nombre d'étudiants selon leur pratique du sport et leur niveau d'étude.
- table_fumeur_sport donne le nombre d'étudiants selon leur pratique du sport et leur statut de fumeur.
- table_transport_sport indique le nombre d'étudiants selon leur pratique du sport et leur état de santé.

```
> # Sexe vs Sport
> table_sexe_sport <- table(data$sport, data$sexe)
> print(table_sexe_sport)

     Un homme Une femme
            1         0
  Non      48        43
  Oui     209        74
>
> # Niveau d'étude vs Sport
> table_niv_sport <- table(data$sport, data$niveau)
> print(table_niv_sport)

         BUT1 BUT2 BUT3 Licence Pro
       0    1    0    0           0
  Non  1   46   26   15           3
  Oui  0  107   81   66          29
>
>
> # Fumeur vs Sport
> table_fumeur_sport <- table(data$sport, data$fumer)
> print(table_fumeur_sport)

      Non Oui
        1   0
  Non  73  18
  Oui 237  46
>
>
> # Sante vs Sport
> table_transport_sport <- table(data$sport, data$sante)
> print(table_transport_sport)

      Non Oui
        0   1
  Non   8  83
  Oui  18 265
```

Ces tableaux permettent d'analyser les effectifs croisés entre la variable sport et d'autres caractéristiques pour étudier d'éventuelles relations.

Ce code crée une fonction pour calculer le test du khi-deux d'indépendance, la p-value associée, et le V de Cramér (qui mesure la force de l'association) à partir d'un tableau croisé entre deux variables qualitatives. Cette fonction est ensuite utilisée pour analyser les relations entre la variable « sport » et quatre autres variables : « sexe », « niveau d'étude », « fumeur » et « santé ». Enfin, les résultats p-values et V de Cramér sont affichés pour chaque paire de variables afin d'évaluer la significativité et la force des liens entre « sport » et ces variables.

```
# Création de la fonction pour calculer le Khi², la p-value et le V de Cramér
calculer_khi2_vcramer <- function(table_croisee) {
  test_result <- chisq.test(table_croisee)
  khi2 <- test_result$statistic
  p_value <- test_result$p.value

  # Calcul du V de Cramér
  n <- sum(table_croisee)
  q <- nrow(table_croisee)
  r <- ncol(table_croisee)
  m <- min(q - 1, r - 1)
  v_cramer <- sqrt(khi2 / (n * m))

  return(list(khi2 = khi2, p = p_value, v = v_cramer))
}

# Appliquer à toutes les variables croisées avec sport

# Sexe
tc_sexe <- table(data$sport, data$sexe)
res_sexe <- calculer_khi2_vcramer(tc_sexe)

# Niveau
tc_niveau <- table(data$sport, data$niveau)
res_niveau <- calculer_khi2_vcramer(tc_niveau)

# Fumeur
tc_fumer <- table(data$sport, data$fumer)
res_fumer <- calculer_khi2_vcramer(tc_fumer)

# Santé
tc_sante <- table(data$sport, data$sante)
res_sante <- calculer_khi2_vcramer(tc_sante)

# Affichage des résultats
cat("Résultats Khi² et V de Cramér\n")
cat("Sport vs Sexe :      p =", res_sexe$p, ", V de Cramér =", res_sexe$v, "\n")
cat("Sport vs Niveau :    p =", res_niveau$p, ", V de Cramér =", res_niveau$v, "\n")
cat("Sport vs Fumer :     p =", res_fumer$p, ", V de Cramér =", res_fumer$v, "\n")
cat("Sport vs Santé :     p =", res_sante$p, ", V de Cramér =", res_sante$v, "\n")
```

Ce passage de code rassemble les résultats des tests (khi-deux, p-value, V de Cramér) pour chaque variable (« Sexe », « Niveau », « Fumer », « Santé ») dans une liste nommée tests. Cela permet de centraliser les résultats pour faciliter leur manipulation et analyse ultérieure.

```r
# Regrouper les résultats dans une liste
tests <- list(
  "Sexe" = res_sexe,
  "Niveau" = res_niveau,
  "Fumer" = res_fumer,
  "Santé" = res_sante
)
```

Voici les tableaux de résultats obtenus pour les différents tests d'association entre la variable sport et les autres variables étudiées.

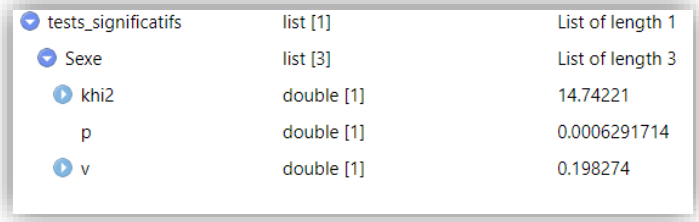
| | | |
|---|---|---|
| tests | list [4] | List of length 4 |
| Sexe | list [3] | List of length 3 |
| khi2 | double [1] | 14.74221 |
| p | double [1] | 0.0006291714 |
| v | double [1] | 0.198274 |
| Niveau | list [3] | List of length 3 |
| khi2 | double [1] | 12.66779 |
| p | double [1] | 0.1238025 |
| v | double [1] | 0.129963 |
| Fumer | list [3] | List of length 3 |
| khi2 | double [1] | 0.8111139 |
| p | double [1] | 0.6666055 |
| v | double [1] | 0.04650774 |
| Santé | list [3] | List of length 3 |
| khi2 | double [1] | 0.7052383 |
| p | double [1] | 0.7028448 |
| v | double [1] | 0.04336629 |

Ce code filtre les résultats des tests pour ne garder que ceux dont la p-value est inférieure à 0,05, ce qui signifie que la relation est statistiquement significative. Ensuite, il extrait les valeurs du V de Cramér pour ces tests significatifs afin de mesurer la force de l'association entre les variables et la variable « sport ».

Le code identifie ensuite la variable qui a la liaison la plus forte avec « sport », c'est-à-dire celle dont le V de Cramér est le plus élevé. Enfin, il crée un tableau regroupant ces résultats significatifs avec leurs valeurs de V de Cramér et affiche ce tableau, ainsi qu'un message indiquant quelle variable est la plus liée à « sport ».

```r
# Extraire uniquement les tests significatifs
tests_significatifs <- lapply(tests, function(x) if (x$p < 0.05) x else NULL)
tests_significatifs <- tests_significatifs[!sapply(tests_significatifs, is.null)]

# Construire un tableau des résultats significatifs
v_cramer_values <- sapply(tests_significatifs, function(x) x$v)

# Trouver la liaison la plus forte (max V de Cramér)
max_v <- max(v_cramer_values)
nom_max_v <- names(which.max(v_cramer_values))

# Construire un tableau en data.frame
resultats_df <- data.frame(
  Test = names(v_cramer_values),
  V_de_Cramer = v_cramer_values
)

# Afficher le tableau avec soulignement (en console on peut juste afficher un message)
print(resultats_df)
cat("\nLa liaison la plus forte est pour :", nom_max_v, "avec un V de Cramér de", max_v, "\n")
```

Ce tableau présente les liaisons significatives entre la variable « sport » et d'autres variables, mesurées par le V de Cramér. La liaison la plus forte est observée entre le sport et le sexe, ce qui indique une association notable entre ces deux variables.



| | | |
|---|---|---|
| tests_significatifs | list [1] | List of length 1 |
| Sexe | list [3] | List of length 3 |
| khi2 | double [1] | 14.74221 |
| p | double [1] | 0.0006291714 |
| v | double [1] | 0.198274 |

En résumé, nos analyses montrent que plusieurs variables sont liées à la pratique du sport, mais c'est la variable sexe qui présente la relation la plus marquée. Cela suggère que les habitudes sportives varient en fonction du sexe des étudiants. Ces résultats soulignent l'importance de prendre en compte les différences démographiques pour mieux comprendre les comportements sportifs. Ces analyses permettent de savoir si deux variables sont liées. La p-value du test du khi² nous indique si cette relation est statistiquement significative (généralement si elle est inférieure à 0,05). Le V de Cramér, lui, mesure la force de cette liaison. Ces outils sont utiles pour mieux comprendre les comportements des personnes à partir des données.