Université de Poitiers

SAÉ - Échantillonnage et estimation

Nouvelle-Aquitaine

Introduction :

Pour cette saé, la région qui nous a été attribuée est la Nouvelle-Aquitaine. Le fichier “population-francaise-communes.xlsx” concerne toutes les communes, nous avons donc lancer les lignes suivantes pour obtenir dans un data frame seules les données concernant la Nouvelle-Aquitaine.

```r
library(stringr)
setwd ← ('C:/Users/hp/OneDrive - Université de Poitiers/REMI SAE STAT IONF')
table ← read.csv2('population_francaise_communes.csv',sep=";",dec=",",header=TRUE,encoding="latin1")

#Création du data frame
donnees ← data.frame(table)

donnees ← subset(donnees0, select=c(Code.département, Commune, Population.totale), Code.région == 75)
```

Nous devions après afficher les 6 premières lignes de ce data frame :

```r
#6 premières ligne de la table
head(donnees)
```

	Code.département	Commune	Population.totale
5201	16	Abzac	508
5202	16	Les Adjots	542
5203	16	Agris	869
5204	16	Aigre	1 615
5205	16	Alloue	480
5206	16	Ambérac	308

Nous devions ensuite créer la variable U contenant l’ensemble des communes de la région pour ensuite créer la variable N qui nous donnera le nombre total de communes avec la fonction length(). Ensuite à l’aide de la variable Population.totale nous avons pu récupérer le nombre exact d’habitant en Nouvelle-Aquitaine qui est de 6 171 721 habitants. Nous avons après cela commencer l’estimation en prenant un échantillon n de 100.

```r
#Population U (ensemble des communes de la région)
U = donnees$Commune
head(U)

#Nombre total N de communes dans U
N = length(U)
N

#nombre T exact d'habitants de la région Nouvelle-Aquitaine
donnees$Population.totale = str_remove_all(donnees$Population.totale," ")
donnees$Population.totale = as.numeric(donnees$Population.totale)
T = sum(donnees$Population.totale)
T

#Estimation
n=100
E=sample(U, n)
head(E)
```

Le résultat du head(E) renvoie ceci :

```
[1] "Bugeat"                 "Montignac-Lascaux"      "Coulonges-Thouarsais"    "Bélis"                  "Saint-Merd-la-Breuille"
[6] "Mouleydier"
```

Nous avons après coup calculé xbar, le nombre moyen d'habitant qui est d'environ 1431.

```
xbar = mean(donnees$Population.totale)
xbar

#idc de mumu
idcmoy = t.test(donnees$Population.totale)$conf.int
idcmoy

#nb d'habitant total estimé
T_est = N*xbar
T_est

#IDC DE T
idcT = idcmoy*N
idcT

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

Notre IDC contient bien T et notre marge d'erreur est de 780 356.

```
# Vérification si la vraie valeur T est dans l'intervalle de confiance
IDC_contains_T = T ≥ idcT[1] && T ≤ idcT[2]
IDC_contains_T
```

```
> IDC_contains_T
[1] TRUE
```

```
> marge
[1] 780356
```

Nous avons également calculé l'IDC de mu pour un niveau de confiance de 0.95.

```
> idcmoy = t.test(donnees$Population.totale)$conf.int
> idcmoy
[1] 1251.187 1613.385
attr(,"conf.level")
[1] 0.95
```

L'étape suivante est le calcul du nombre total d'habitant estimé ainsi que l'IDC estimé de ce nombre.

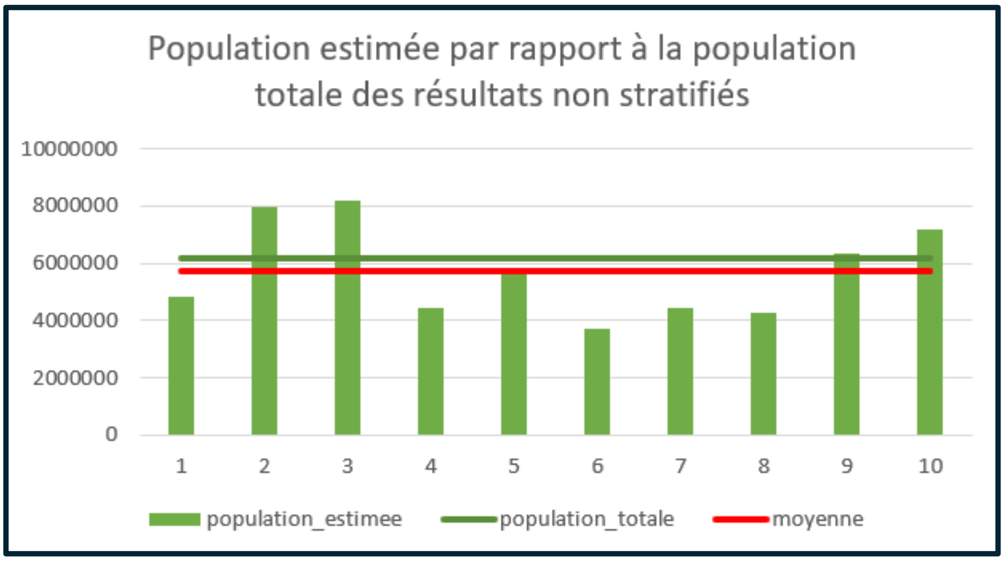
```
> T_est = N*xbar
> T_est
[1] 6171721
```

```
> idcT = idcmoy*N
> idcT
[1] 5391365 6952077
attr(,"conf.level")
[1] 0.95
```

Nous avons ensuite programmé une boucle afin de répéter ce processus 10 fois pour pouvoir le mettre dans un fichier csv que nous avons ouvert sur excel. Nous avons pu produire un graphique illustrant ces résultats.

```r
# Initialisation d'un dataframe pour stocker les résultats
resultats <- data.frame(population_totale = numeric(), population_estimee = numeric(), idc = numeric(), marge_erreur = numeric())

for (i in 1:10) {
  # Question 4 : Estimation
  n <- 100
  E <- sample(U, n)
  xbar <- mean(donnees$Population.totale[donnees$Commune %in% E])

  # Question 5 : IDC de T
  idcT <- t.test(donnees$Population.totale[donnees$Commune %in% E])$conf.int * N

  # Concaténation des deux valeurs de l'IDC dans une même chaîne de caractères
  idc <- paste("[", round(idcT[1], 0), ";", round(idcT[2], 0), "]", sep = "")

  # Question 6 : Marge d'erreur
  marge <- (idcT[2] - idcT[1]) / 2

  # Stockage des résultats dans le dataframe
  resultats <- rbind(resultats, data.frame(population_totale = T, population_estimee = N * xbar, idc = idc, marge_erreur = marge))
}

write.csv2(resultats, "resultats.csv", row.names = FALSE)
```

Voici le graphique obtenu :



Les données utilisées pour faire ce graphique sont les suivantes :

population_totale	population_estimee	idc	marge_erreur	moyenne
6171721	4825582,808	[3504045;6147120]	1321537,322	5720385,019
6171721	7969942,66	[341818;15598067]	7628124,275	5720385,019
6171721	8196610,856	[3628797;12764424]	4567813,418	5720385,019
6171721	4455545,532	[2404519;6506572]	2051026,584	5720385,019
6171721	5729641,056	[2757391;8701891]	2972249,813	5720385,019
6171721	3721898,75	[2827877;4615920]	894021,2558	5720385,019
6171721	4458240,558	[3090350;5826131]	1367890,608	5720385,019
6171721	4298368,449	[3088688;5508049]	1209680,929	5720385,019
6171721	6342525,832	[3310938;9374113]	3031587,498	5720385,019
6171721	7205493,692	[3605870;10805118]	3599623,943	5720385,019

Conclusion :

Bien que l'échantillonnage aléatoire simple soit une méthode efficace et pratique pour estimer la population totale de la région Nouvelle-Aquitaine, l'utilisation de techniques plus sophistiquées, telles que l'échantillonnage stratifié, pourrait fournir des estimations plus précises et fiables.

À la suite de cela, nous avons effectué les mêmes opérations mais sur un échantillonnage stratifié. Cet échantillonnage a été mise en place à l'aide de quantiles.

```
# Création des strates en utilisant les quantiles
quantiles <- quantile(donnees$Population.totale, probs = c(0, 0.25, 0.5, 0.75, 1))
donnees$strate <- cut(donnees$Population.totale, breaks = quantiles, include.lowest = TRUE, labels = FALSE)

# Affichage des 6 premières lignes de la nouvelle table
head(donnees)

# Conversion en data frame nommé "datastrat"
datastrat <- data.frame(donnees)

# Affichage des 6 premières lignes de "datastrat"
head(datastrat)
```

Voici Les 6 premières lignes de datastrat :

```
> head(datastrat)
     Code.département    Commune Population.totale strate
5201                16      Abzac               508      3
5202                16 Les Adjots               542      3
5203                16      Agris               869      3
5204                16      Aigre              1615      4
5205                16     Alloue               480      2
5206                16    Ambérac               308      2
```

Les échantillons stratifiés ont ensuite été placé dans un data frame. Nous avons programmé une boucle pour tirer des échantillons dans chaque strate.

```
# Tirage d'un échantillon stratifié de taille n = 100 avec effectifs égaux dans les strates
n <- 100
num_strates <- length(unique(donnees$strate))
taille_echantillon_par_strate <- n / num_strates

# Initialisation de l'échantillon
echantillon_stratifie <- data.frame()

# Boucle pour tirer des échantillons dans chaque strate
for (strate in unique(donnees$strate)) {
  communes_strate <- subset(donnees, strate == strate)
  echantillon_strate <- communes_strate[sample(1:nrow(communes_strate), taille_echantillon_par_strate), ]
  echantillon_stratifie <- rbind(echantillon_stratifie, echantillon_strate)
}
```

Nous avons effectué après cela des calculs de la moyenne de chaque strate ainsi que leur variance.

```r
# Calcul des moyennes et variances des strates
moyennes_strates <- tapply(echantillon_stratifie$Population.totale, echantillon_stratifie$strate, mean)
variances_strates <- tapply(echantillon_stratifie$Population.totale, echantillon_stratifie$strate, var)
moyennes_strates
variances_strates

# Calcul de Xbarstr (moyenne pondérée des moyennes des strates)
Xbarstr <- mean(moyennes_strates)

# Calcul de l'estimation de la variance de Xbarstr
n_h <- taille_echantillon_par_strate
N_h <- table(donnees$strate)  # Effectifs totaux par strate

variance_Xbarstr <- sum((N_h / sum(N_h))^2 * variances_strates / n_h)
```

Ensuite, nous avons calculé la moyenne pondérée des moyennes des strates puis nous avons estimé la variance ainsi que les effectifs totaux par strate.

Voici les moyennes de chaque strate ainsi que leurs variances :

```
> moyennes_strates
        1         2         3         4 
 161.2800  324.2500  723.3125 2942.4737 
> variances_strates
          1           2           3           4 
   2178.543    4010.717   35163.512 9332943.152 
```

Sur le bout de code ci-dessous nous avons calculé un intervalle de confiance pour xbarstr et nous avons définis les bornes inférieures et supérieures.

```r
# Calcul d'un intervalle de confiance pour Xbarstr
alpha <- 0.05  # Niveau de confiance à 95%
t_value <- qt(1 - alpha/2, df=n-1)
marge_erreur <- t_value * sqrt(variance_Xbarstr)

borne_inf <- Xbarstr - marge_erreur
borne_sup <- Xbarstr + marge_erreur

# Affichage des résultats
list(
  Xbarstr = Xbarstr,
  variance_Xbarstr = variance_Xbarstr,
  IDC = c(borne_inf, borne_sup)
)
```

Voici les résultats :

```
$Xbarstr
[1] 1037.829

$variance_Xbarstr
[1] 23423.84

$IDC
[1]  734.1476 1341.5105
```

Notre IDC contient bien encore la valeur T.

```r
IDC_T <- T >= borne_inf_T && T <= borne_sup_T
IDC_T
```

```
> IDC_T
[1] TRUE
```

Nous avons ensuite estimé T à l'aide des estimations de la variance de xbarstr puis nous avons calculé un intervalle de confiance pour ce dernier ainsi que les bornes inférieures et supérieures comme fait précédemment.

```r
# Estimation de Tstr
Tstr <- sum(N_h) * Xbarstr

# Calcul de l'intervalle de confiance pour T
variance_Tstr <- sum(N_h^2 * variances_strates / n_h)
marge_erreur_T <- t_value * sqrt(variance_Tstr)

borne_inf_T <- Tstr - marge_erreur_T
borne_sup_T <- Tstr + marge_erreur_T

# Initialisation du data frame pour stocker les résultats
resultats <- data.frame(
  population_totale = numeric(),
  population_estimee = numeric(),
  IDC = character(),
  marge_erreur = numeric()
)
```

Nous avons ensuite effectué une boucle pour obtenir ces résultats 10 fois. Ces résultats seront plus tard intégrés dans un fichier qui sera ouvert sous Excel afin de produire un graphique.

```r
# Répétition des estimations 10 fois
for (i in 1:10) {
  # Tirage d'un échantillon stratifié
  echantillon_stratifie <- data.frame()
  for (strate in unique(donnees$strate)) {
    communes_strate <- subset(donnees, strate == strate)
    echantillon_strate <- communes_strate[sample(1:nrow(communes_strate), taille_echantillon_par_strate), ]
    echantillon_stratifie <- rbind(echantillon_stratifie, echantillon_strate)
  }
```

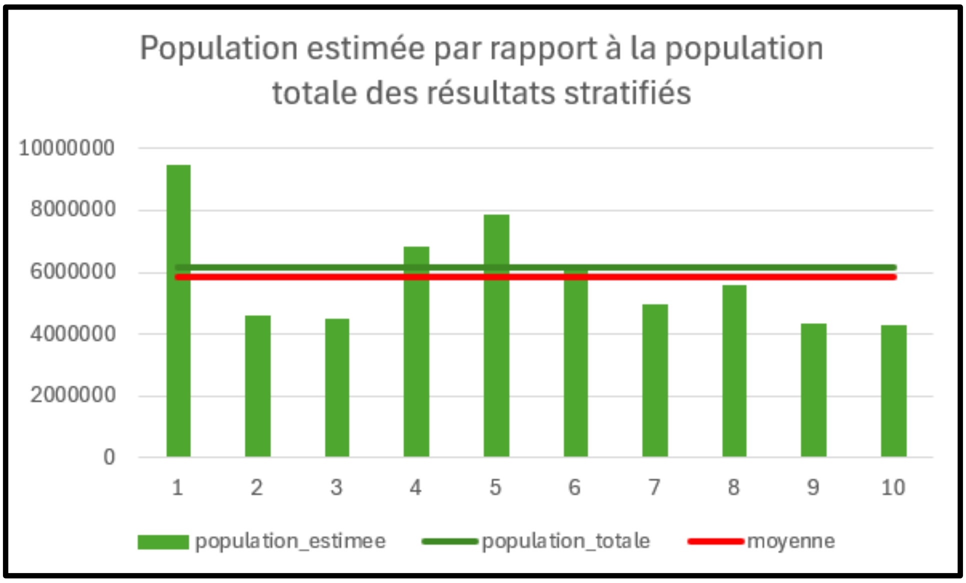
Nous avons ensuite calculé les moyennes et variances des strates et nous avons ensuite calculé les estimations pour les strates.

```r
# Calcul des moyennes et variances des strates
moyennes_strates <- tapply(echantillon_stratifie$Population.totale, echantillon_stratifie$strate, mean)
variances_strates <- tapply(echantillon_stratifie$Population.totale, echantillon_stratifie$strate, var)

# Estimation de la moyenne stratifiée
Xbarstr <- mean(moyennes_strates)
variance_Xbarstr <- sum((N_h / sum(N_h))^2 * variances_strates / n_h)
marge_erreur <- t_value * sqrt(variance_Xbarstr)

# Estimation de Tstr
Tstr <- sum(N_h) * Xbarstr
variance_Tstr <- sum(N_h^2 * variances_strates / n_h)
marge_erreur_T <- t_value * sqrt(variance_Tstr)

# Intervalle de confiance pour T
borne_inf_T <- Tstr - marge_erreur_T
borne_sup_T <- Tstr + marge_erreur_T
IDC_T <- paste("[", round(borne_inf_T, 0), ";", round(borne_sup_T, 0), "]", sep = "")
```

Les résultats sont ensuite stockés et mis dans un fichier csv.

```
    # Stockage des résultats
    resultats <- rbind(resultats, data.frame(
      population_totale = T,
      population_estimee = Tstr,
      IDC = IDC_T,
      marge_erreur = marge_erreur_T
    ))
  }

  # Écriture des résultats dans un fichier CSV
  write.csv2(resultats, "resultats_stratifies.csv", row.names = FALSE)
```

Voici le graphique représentant ces résultats :



Les données utilisées pour ce tableau sont les suivantes :

population_totale	population_estimee	IDC	marge_erreur	moyenne
6171721	9480988,365	[1172207;17789770]	8308781,761	5870743,12
6171721	4578340,714	[3480904;5675777]	1097436,634	5870743,12
6171721	4519320,321	[3506793;5531848]	1012527,57	5870743,12
6171721	6813975,37	[1524291;12103659]	5289684,082	5870743,12
6171721	7886907,701	[2537905;13235911]	5349003,136	5870743,12
6171721	6208666,261	[2113325;10304008]	4095341,268	5870743,12
6171721	4972934,122	[3661828;6284040]	1311106,021	5870743,12
6171721	5602886,706	[3976314;7229459]	1626572,769	5870743,12
6171721	4343757,401	[3182803;5504711]	1160954,035	5870743,12
6171721	4299654,215	[3279816;5319493]	1019838,344	5870743,12

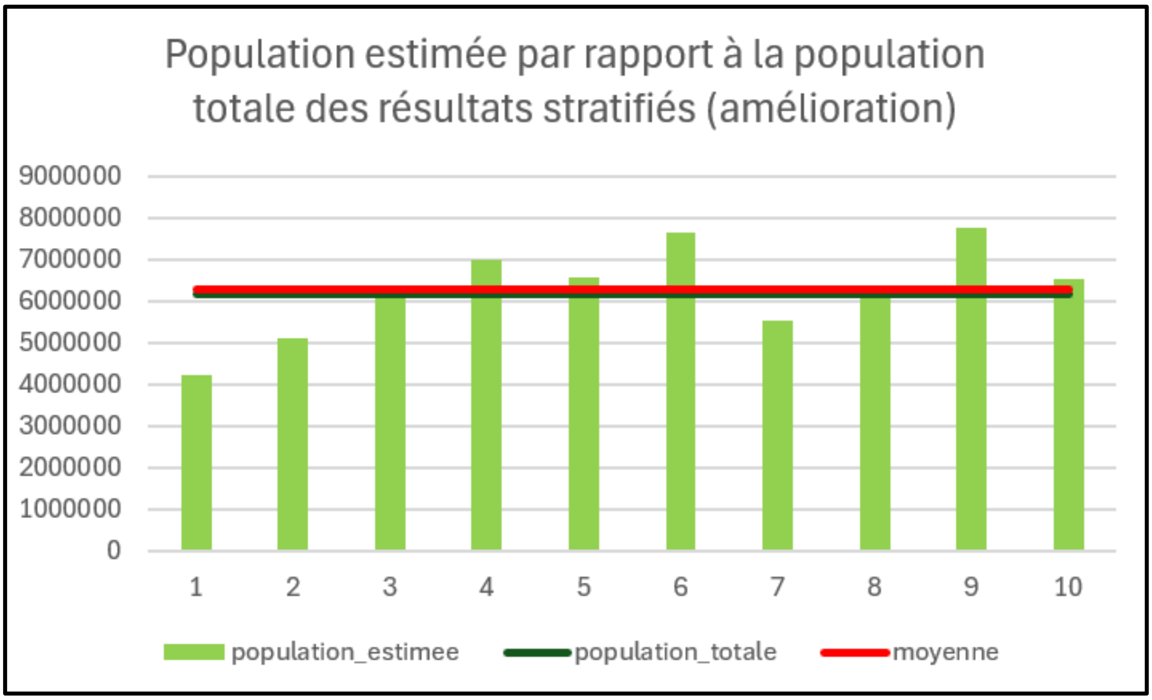
Pour des résultats encore plus précis, nous avons décidé d’effectuer ces calculs avec 6 strates au lieu de 4.

```
# Création des strates en utilisant plus de quantiles (6 strates)
quantiles <- quantile(donnees$Population.totale, probs = seq(0, 1, length.out = 7))
donnees$strate <- cut(donnees$Population.totale, breaks = quantiles, include.lowest = TRUE, labels = FALSE)
```

Voici les moyennes et les variances de ces 6 strates.

```
> moyennes_strates
        1         2         3         4         5         6 
 134.9091  240.5625  384.9375  650.1905 1151.0588 3090.0667 
> variances_strates
          1           2           3           4           5           6 
   1857.691    1155.329    2504.329    7075.162   25307.934 4907917.638 
```

Voici le graphique obtenu à l’issue de ces calculs.



Voici les données utilisées pour ce graphique :

population_totale	population_estimee	IDC	marge_erreur	moyenne
6171721	4242914,417	[3589376;4896453]	653538,7158	6283336,02
6171721	5120338,959	[3338065;6902613]	1782273,711	6283336,02
6171721	6364378,945	[3515713;9213045]	2848666,078	6283336,02
6171721	6988365,289	[3071711;10905019]	3916653,86	6283336,02
6171721	6562768,199	[3637152;9488384]	2925615,839	6283336,02
6171721	7644459,918	[3038684;12250235]	4605775,548	6283336,02
6171721	5535160,954	[3229636;7840686]	2305524,71	6283336,02
6171721	6104629,72	[2492677;9716582]	3611952,248	6283336,02
6171721	7742224,204	[2465088;13019360]	5277135,92	6283336,02
6171721	6528119,611	[4062444;8993795]	2465675,333	6283336,02

Conclusion :

La méthode d'échantillonnage stratifié tend à donner des estimations plus précises avec des marges d'erreur plus faibles. Comme nous avons pu le voir ici sur chaque graphique, dès que nous avons stratifié nos échantillons la moyenne de la population estimée se rapprochait de la population totale réelle. De même lorsque que nous avons décidé d'augmenter le nombre de strate. Une proposition d'amélioration serait donc d'utiliser un plus grand nombre de strates pour mieux représenter la population.

Partie 2 : Test d'indépendance du Khi-deux

L'objectif de la seconde partie de la SAE est d'effectuer un test du khi deux d'indépendance. Nous disposons du fichier voitures.xlsx qui contient un jeu de données sur les voitures ainsi que leurs primes d'assurance. Nous effectuons ces manipulations à l'aide du langage de programmation statistique R.

La première étape consiste à convertir le fichier XLSX en format CSV. Une fois cela fait, nous pouvons le manipuler avec R. Nous affichons ensuite les six premières lignes afin de mieux comprendre l'organisation de la table.

```
setwd("/Users/remipierron/Library/Mobile Documents/com~apple~CloudDocs/SAE stat inf")
table <- read.csv2('Voitures.csv',sep=";",dec=",",header=TRUE,encoding="latin1")
head(table, 6)
```

```
> head(table, 6)
   Marque PuissFisc Categorie Anciennete       Formule           Ville Prime
1 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi     Chantecorps   171
2 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi      Le Chillou   176
3 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi       LoubillÃ©   316
4 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi Pougne-Herisson   310
5 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi     Villefollet   316
6 Citroen de_7_a_10   BERLINE    4-7 ans Tous Risques     Chantecorps   444
```

La table contient les variables suivantes :

- Marque : la marque de la voiture (par exemple, Citroen)
- PuissFisc : la puissance fiscale de la voiture (par exemple, de 7 à 10)
- Categorie : la catégorie de la voiture (par exemple, BERLINE)
- Anciennete : l'ancienneté de la voiture (par exemple, 4-7 ans)
- Formule : le type de formule d'assurance choisi (par exemple, Tiers Maxi)
- Ville : la ville où la voiture est assurée (par exemple, Chantecorps)
- Prime : la prime d'assurance annuelle de la voiture (par exemple, 171 euros)

Marque, PuissFisc, Categorie, Anciennete, Formule et Ville sont des variables qualitatives, tandis que prime est une variable quantitative.

```
str(table$Prime)
table$prime <- as.numeric(table$Prime)
table$Prime2 <- cut(table$Prime,
                    breaks = c(-Inf, 200, 400, Inf),
                    labels = c("faible", "moyenne", "elevee"),
                    include.lowest = TRUE)
```

Le type de la variable "Prime" dans le jeu de données est vérifié à l'aide de la fonction str(). Si cette variable n'est pas de type numérique, elle est convertie en type numérique à l'aide de la fonction as.numeric(). Une nouvelle variable "Prime2" est créée à partir de la variable "Prime" en utilisant la fonction cut(). Cette nouvelle variable est un facteur avec trois niveaux : "faible" pour les primes inférieures à 200, "moyenne" pour les primes entre 200 et 400, et "élevée" pour les primes supérieures à 400. Nous affichons par la suite le début de la nouvelle table.

```
> head(table)
   Marque PuissFisc Categorie Anciennete       Formule             Ville Prime  Prime2
1 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi       Chantecorps   171  faible
2 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi        Le Chillou   176  faible
3 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi         LoubillÃ©   316 moyenne
4 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi Pougne-Herisson   310 moyenne
5 Citroen de_7_a_10   BERLINE    4-7 ans    Tiers Maxi       Villefollet   316 moyenne
6 Citroen de_7_a_10   BERLINE    4-7 ans Tous Risques      Chantecorps   444  elevee
```

Plusieurs tableaux croisés sont construits entre la variable "Prime2" et d'autres variables du jeu de données, comme "Marque", "PuissFisc", "Categorie", "Anciennete" et "Formule". Cela est fait à l'aide de la fonction table(). Le premier tableau croisé entre "Prime2" et "Marque" est affiché à l'aide de la fonction print().

```
tableau_croise_Marque <- table(table$Prime2, table$Marque)
tableau_croise_PuissFisc <- table(table$Prime2, table$PuissFisc)
tableau_croise_Categorie <- table(table$Prime2, table$Categorie)
tableau_croise_Anciennete <- table(table$Prime2, table$Anciennete)
tableau_croise_Formule <- table(table$Prime2, table$Formule)
```

```
> print(tableau_croise_Marque)

          Citroen Peugeot Renault
  faible       28       4      28
  moyenne      47     128      47
  elevee      105      48     105
```

Des tests du Khi-deux d'indépendance sont effectués sur les tableaux croisés à l'aide de la fonction chisq.test(). Les résultats de ces tests sont stockés dans des variables et affichés à l'aide de la fonction print().

```r
khi2_Marque <- chisq.test(tableau_croise_Marque)
khi2_PuissFisc <- chisq.test(tableau_croise_PuissFisc)
khi2_Categorie <- chisq.test(tableau_croise_Categorie)
khi2_Anciennete <- chisq.test(tableau_croise_Anciennete)
khi2_Formule <- chisq.test(tableau_croise_Formule)

print(khi2_Marque)
print(khi2_PuissFisc)
print(khi2_Categorie)
print(khi2_Anciennete)
print(khi2_Formule)
```

```
> print(khi2_Marque)

        Pearson's Chi-squared test

data:  tableau_croise_Marque
X-squared = 103.49, df = 4, p-value < 2.2e-16

> print(khi2_PuissFisc)

        Pearson's Chi-squared test

data:  tableau_croise_PuissFisc
X-squared = 1.5371, df = 2, p-value = 0.4637

> print(khi2_Categorie)

        Pearson's Chi-squared test

data:  tableau_croise_Categorie
X-squared = 8.5823, df = 4, p-value = 0.07243

> print(khi2_Anciennete)

        Pearson's Chi-squared test

data:  tableau_croise_Anciennete
X-squared = 42.41, df = 4, p-value = 1.372e-08

> print(khi2_Formule)

        Pearson's Chi-squared test

data:  tableau_croise_Formule
X-squared = 12.675, df = 2, p-value = 0.001769
```

Voici les résultats des tests du khi deux :

- Il y a une relation significative entre la variable "Prime2" et la variable "Marque" (X-squared = 103.49, df = 4, p-value < 2.2e-16). Cela signifie que la marque de la voiture est associée à la prime d'assurance.

- Il n'y a pas de relation significative entre la variable "Prime2" et la variable "PuissFisc" (X-squared = 1.5371, df = 2, p-value = 0.4637). Cela signifie que la puissance fiscale de la voiture n'est pas associée à la prime d'assurance.

- Il n'y a pas de relation significative entre la variable "Prime2" et la variable "Categorie" (X-squared = 8.5823, df = 4, p-value = 0.07243). Cela signifie que la catégorie de la voiture n'est pas associée à la prime d'assurance.

- Il y a une relation significative entre la variable "Prime2" et la variable "Anciennete" (X-squared = 42.41, df = 4, p-value = 1.372e-08). Cela signifie que l'ancienneté de la voiture est associée à la prime d'assurance.

- Il y a une relation significative entre la variable "Prime2" et la variable "Formule" (X-squared = 12.675, df = 2, p-value = 0.001769). Cela signifie que le type de formule d'assurance choisi est associé à la prime d'assurance.

En résumé, les variables "Marque", "Anciennete" et "Formule" sont significativement associées à la prime d'assurance, tandis que les variables "PuissFisc" et "Categorie" ne le sont pas.

Les valeurs de V de Cramer sont calculées pour les tests du Khi-deux effectués précédemment. Ces valeurs sont une mesure de l'association entre les variables dans les tableaux croisés. Elles sont stockées dans une variable et affichées à l'aide de la fonction print(). Les valeurs de V de Cramer sont calculées pour les tableaux croisés entre "Prime2" et "Marque", "Prime2" et "Anciennete", et "Prime2" et "Formule", qui ont tous des tests du Khi-deux significatifs. Les valeurs de V de Cramer ne sont pas calculées pour les tableaux croisés entre "Prime2" et "PuissFisc" et "Prime2" et "Categorie", qui n'ont pas de tests du Khi-deux significatifs.

```
V_values <- c()
V_Marque <- sqrt(khi2_Marque$statistic/(sum(tableau_croise_Marque)*min(nrow(tableau_croise_Marque)-1, ncol(tableau_croise_Marque)-1)))
V_Anciennete <- sqrt(khi2_Anciennete$statistic/(sum(tableau_croise_Anciennete)*min(nrow(tableau_croise_Anciennete)-1, ncol(tableau_croise_Anciennete)-1)))
V_Formule <- sqrt(khi2_Formule$statistic/(sum(tableau_croise_Formule)*min(nrow(tableau_croise_Formule)-1, ncol(tableau_croise_Formule)-1)))

V_values <- c(V_Marque, V_Anciennete, V_Formule)

V_table <- data.frame(Test = c("Marque", "Anciennete", "Formule"),V_de_Cramer = V_values)
print(V_table)
```

Le tableau `V_table` contient les valeurs de V de Cramer pour les tests du Khi-deux significatifs entre la variable "Prime2" et les variables "Marque", "Anciennete" et "Formule".

```
> print(V_table)
        Test V_de_Cramer
1     Marque   0.3095609
2 Anciennete   0.1981618
3    Formule   0.1532065
```

La première ligne du tableau indique que la valeur de V de Cramer pour le test du Khi-deux entre "Prime2" et "Marque" est de 0,3095609. Cela signifie qu'il y a une association modérée entre la marque de la voiture et la prime d'assurance.

La deuxième ligne du tableau indique que la valeur de V de Cramer pour le test du Khi-deux entre "Prime2" et "Anciennete" est de 0,1981618. Cela signifie qu'il y a une association faible à modérée entre l'ancienneté de la voiture et la prime d'assurance.

La troisième ligne du tableau indique que la valeur de V de Cramer pour le test du Khi-deux entre "Prime2" et "Formule" est de 0,1532065. Cela signifie qu'il y a une association faible à modérée entre le type de formule d'assurance choisi et la prime d'assurance.

En résumé, le tableau `V_table` montre que la marque de la voiture a la plus forte association avec la prime d'assurance, suivie par l'ancienneté de la voiture et le type de formule d'assurance choisi. Cependant, il convient de noter que ces associations sont toutes faibles à modérées et que d'autres facteurs peuvent également influencer la prime d'assurance.