SAÉ - Échantillonnage et estimation

Région **Provence-Alpes-Côte d'Azur**

Table des matières

Introduction

L'objectif de cette SAE est d'explorer la manière dont l'incertitude et la précision de l'estimation d'une grandeur mesurable dans une population peuvent être appréhendées à travers la construction d'un intervalle de confiance à partir d'un processus d'échantillonnage. Dans notre étude, nous avons examiné la population de la région Provence-Alpes-Côte d'Azur (PACA) (1ère et 2ème partie). En outre, cette SAE nous permet d'examiner la corrélation entre deux variables. Dans notre cas, nous avons analysé la relation entre le prix de l'assurance et les caractéristiques d'un véhicule.

Pour commencer, nous avons utilisé le logiciel R pour réaliser un échantillonnage via un sondage aléatoire simple avec des probabilités égales (chaque individu ayant le même poids dans la population).

Ensuite, nous avons utilisé une méthode de sondage stratifié pour procéder à une estimation. Nous avons comparé ces estimations aux valeurs réelles pour évaluer la précision des résultats et les interpréter.

Nous avons alors utilisé le test du khi-deux sur R pour examiner s'il y avait une relation entre deux variables. Nos recherches visaient à vérifier si le coût d'une prime d'assurance dépendait des caractéristiques d'une voiture.

Partie 1 : Estimation du nombre d'habitants en région PACA

Dans cette partie du code, nous installons les bibliothèques nécessaires pour les analyses ultérieures et procédons à la lecture du fichier de données. Nous filtrons ensuite les informations relatives à la région PACA et sélectionnons uniquement les colonnes pertinentes pour nos calculs. De plus, nous apportons quelques transformations aux données pour les mettre au bon format.

```
#install.packages("survival")
#install.packages("sampling")
#install.packages("stringr")
library(survival)
library(sampling)
library(stringr)
table = read.csv2("SAE echantillonage\\population_francaise_communes.csv", sep =";", dec=",", header=TRUE)

paca <- subset(table, Nom.de.la.région == "Provence-Alpes-Côte d'Azur")
paca <- paca[,c("Code.département", "Commune", "Population.totale")]
head(paca)
paca$Code.département=as.numeric(paca$Code.département)
paca$Population.totale=str_remove_all(paca$Population.totale, " ")
paca$Population.totale=as.numeric(paca$Population.totale)
```

Echantillonnage aléatoire simple

Ensuite nous avons rassemblé toutes les communes de la région PACA dans une variable. Puis, nous avons déterminé le nombre total de communes (961) présentes dans cette région spécifique.

```r
#ensemble des communes en PACA
U <- paca$Commune
#Nombre de communes en PACA
N <- length(U)
```

En utilisant la colonne "Population.totale" de notre jeu de données, qui indique le nombre d'habitants dans chaque commune, nous avons calculé le nombre total d'habitants. En additionnant toute la colonne, nous avons déterminé qu'il y avait 5 174 034 habitants au total dans la région PACA.

```r
#Le nombre total d'habitants en PACA (soit 5 174 034)
T= sum(paca$Population.totale)
T
```

Par la suite, nous avons tenté d'estimer la population totale de la région en utilisant un échantillon de 100 communes sélectionnées au hasard parmi toutes celles disponibles. À partir de ce tirage, nous avons construit un tableau récapitulant le nom des communes tirées au sort ainsi que leur nombre d'habitants. En calculant la moyenne du nombre d'habitants par commune dans cet échantillon et en répétant ce processus 10 fois, nous avons cherché à estimer la taille de la population. En utilisant cette moyenne, le nombre total de communes et le nombre de communes dans l'échantillon, nous avons pu estimer le nombre total d'habitants en PACA.

```r
# Répétition du processus 10 fois
for (i in 1:10) {
  # Tirage aléatoire simple d'un échantillon de taille n=100
  n = 100
  E = sample(U, n)
  # récupération des données des communes tirées dans l'échantillon
  paca1 <- paca[paca$Commune %in% E, ]
  # nb d'habitants des communes de l'échantillon
  paca2 <- subset(paca1, select=c(Commune, Population.totale))
  xbar <- mean(paca2$Population.totale)
```

Pour mieux interpréter nos résultats, nous les avons consignés dans un tableau afin de les comparer. Nous avons remarqué que nos estimations n'étaient pas toujours très précises. En effet, notre échantillon pouvait inclure des communes à faible population ou, au contraire, des communes très peuplées, ce qui pouvait engendrer des variations significatives par rapport aux données démographiques réelles. C'est l'une des limitations du sondage aléatoire simple lorsque la population n'est pas homogène.

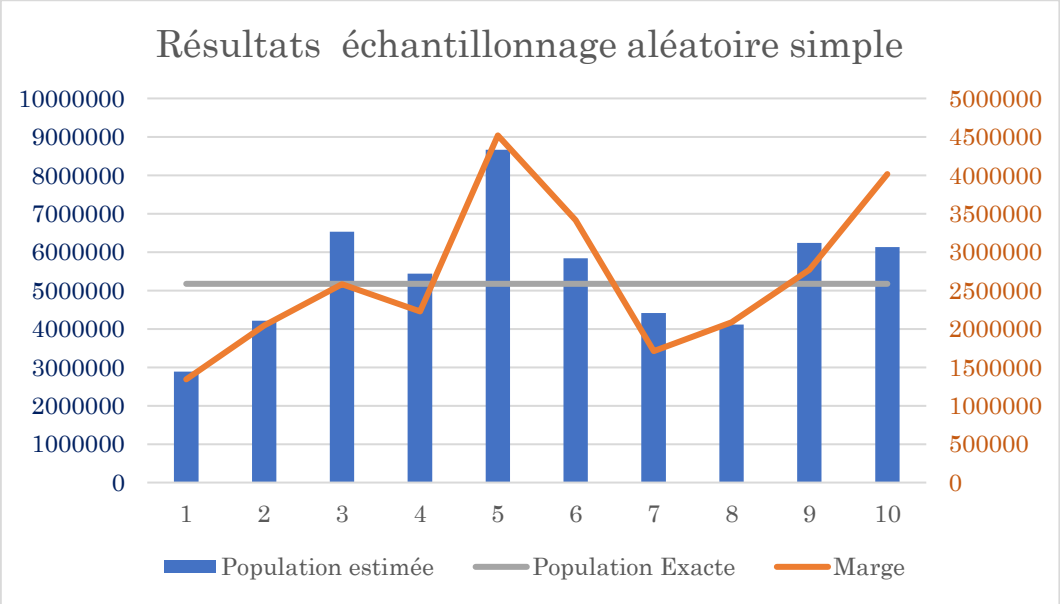
```r
  # idc de mu
  idcmoy <- t.test(paca2$Population.totale)$conf.int
  # estimation nb total hab
  T_est <- N * xbar
  # idc de T
  idcT <- idcmoy * N
  # Calcul de la marge d'erreur
  marge <- (idcT[2] - idcT[1]) / 2
  # Stockage des résultats dans le dataframe
  new_row = data.frame(Pop_estimee = T_est,
                       Binf = idcT[1],
                       bsup = idcT[2],
                       Marge = marge,
                       Pop_exacte = T)
  resultats = rbind(resultats,new_row)
}
```

En conclusion, nous avons établi un intervalle de confiance (IDC) avec un risque 5%, ce qui signifie qu'il y avait 95% de chances que la véritable valeur du paramètre estimé se trouve à l'intérieur de cet intervalle.

En d'autres termes, nous avions 95% de certitude que le véritable nombre d'habitants se situait dans l'intervalle de confiance créé à partir de notre échantillon de 100 communes. Pour notre premier essai, nous avons obtenu un intervalle de confiance égal à [2 172 506; 6 259 807], le nombre réel d'habitants en PACA (5 174 034) se trouvait bien à l'intérieur de cet intervalle. Ainsi, nous avions une certitude de 95% que la vraie valeur du nombre d'habitants en PACA était incluse dans l'intervalle que nous avions trouvé grâce à notre échantillon de 100 communes. Nous avons ensuite enregistré nos résultats dans notre data frame.

Résultats de l'échantillonnage aléatoire simple :

Pop_estimee	Binf	bsup	Marge	Pop_exacte
2887701,36	1544127,3	4231275,43	1343574,07	5174034
4216156,49	2172505,91	6259807,07	2043650,58	5174034
6529338,93	3946879,43	9111798,44	2582459,5	5174034
5436890,16	3208663,69	7665116,63	2228226,47	5174034
8663234,22	4144457,04	13182011,4	4518777,18	5174034
5835860,93	2419937,72	9251784,15	3415923,22	5174034
4411432,81	2698383,88	6124481,75	1713048,93	5174034
4115713,51	2031639,95	6199787,07	2084073,56	5174034
6238193,53	3466771,54	9009615,53	2771421,99	5174034
6132962,05	2116695,98	10149228,11	4016266,06	5174034



Echantillonnage aléatoire stratifié

À la suite de nos premiers résultats et observations, nous avons entrepris une comparaison entre un échantillonnage aléatoire simple et un échantillonnage stratifié. Nous avons donc repris les mêmes données et suivi la même démarche jusqu'à la sélection de l'échantillon.

Pour commencer, nous avons utilisé la fonction "summary" pour obtenir les quantiles de la population des communes de la PACA afin de créer des strates regroupant les communes en quartiles en fonction de leur population. Cependant, comme le résultat ne nous semblait pas suffisamment précis, nous avons décidé de répartir les communes en déciles, formant ainsi 10 strates de population. La première strate comprend les 10% des communes les moins peuplées, tandis que la 10ème strate regroupe les 10% des communes les plus peuplées de la région.

```r
#quantiles de la variable Population.totale
summary(paca$Population.totale)
#création de déciles de la variable Population.totale
deciles <- quantile(paca$Population.totale, probs = seq(0, 1, by = 0.1))
paca$strate <- cut(paca$Population.totale, breaks = deciles, labels = 1:10, include.lowest = TRUE)
pacastrat <- paca[, c("Commune", "Population.totale", "strate")]
Nh <- table(pacastrat$strate)
```

Une fois que nous avions défini nos strates, nous avons procédé au calcul des moyennes et des variances pour chacune des dix strates. Ensuite, après avoir effectué ces calculs, nous avons également déterminé la moyenne et la variance globale de chacune des strates. Ces calculs ont ensuite été utilisés pour estimer la population de la région PACA.

```r
for (i in 1:10) {
  st <- strata(pacastrat, stratanames = "strate", size = nh, method = "srswor")
  paca1 <- getdata(pacastrat, st)

  gh <- Nh / N
  fh <- nh / Nh

  # moyenne et variance de chaque strate
  ech1 <- paca1[paca1$strate == 1, ]
  ech2 <- paca1[paca1$strate == 2, ]
  ...
  ech10 <- paca1[paca1$strate == 10, ]

  m1 <- mean(ech1$Population.totale)
  m2 <- mean(ech2$Population.totale)
  ...
  m10 <- mean(ech10$Population.totale)

  var1 <- var(ech1$Population.totale)
  var2 <- var(ech2$Population.totale)
  ...
  var10 <- var(ech10$Population.totale)

  # moyenne géénérale des 10 échantillons
  Xbarst <- sum(Nh * c(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10)) / N
  # variance de la moyenne générale
  varXbarst <- sum((gh^2) * (1 - fh) * c(var1, var2, var3, var4, var5, var6, var7, var8, var9, var10) / nh)
```

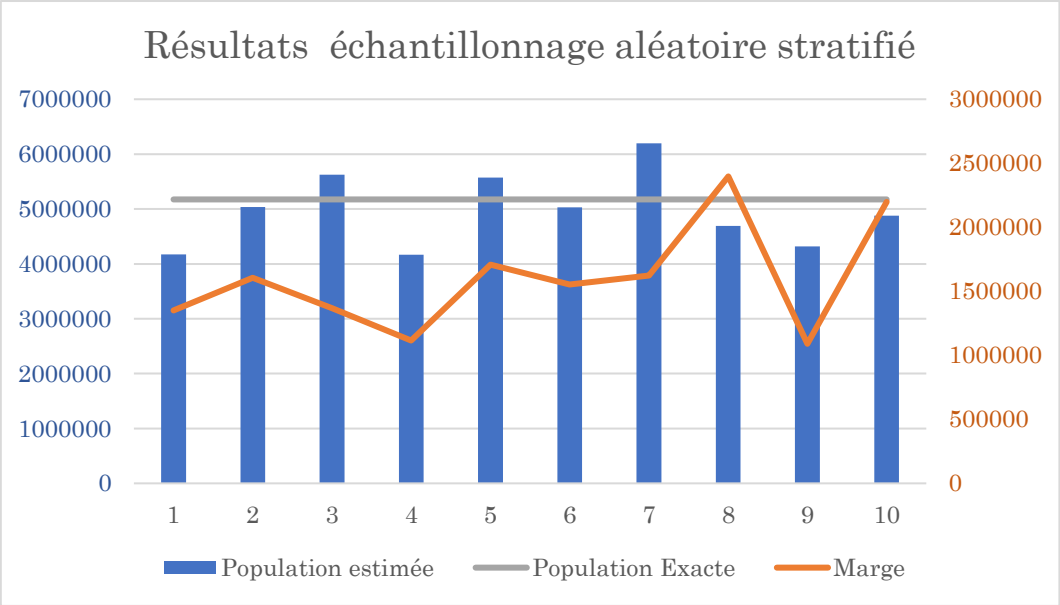
Pour conclure, nous avons établi un intervalle de confiance (IDC) avec une fiabilité de 95%, ce qui signifie qu'il y avait 95% de chances que la vraie valeur du paramètre estimé (en l'occurrence, le nombre d'habitants en région PACA) se trouve à l'intérieur de cet intervalle, ainsi que la marge d'erreur associée à cet IDC. En d'autres termes, nous étions certains à 95% de trouver le vrai nombre d'habitants dans l'intervalle de confiance construit à partir de l'échantillon de 100 communes stratifié.

```
# calcul de l'intervalle de confiance à 95% pour Xbarst
alpha <- 0.05
binf <- Xbarst - qnorm(1 - alpha / 2) * sqrt(varXbarst)
bsup <- Xbarst + qnorm(1 - alpha / 2) * sqrt(varXbarst)
idcmoy <- c(binf, bsup)

# estimation du nombre total d'habitants
Tstr <- N * Xbarst

# estimation du nombre total T grâce à un idc
binf <- idcmoy[1] * N
bsup <- idcmoy[2] * N
idcT <- c(binf, bsup)

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

Pour notre premier essai, nous avons obtenu un intervalle de confiance égal à [2 820 189 ; 5 519 066]. Ainsi, le nombre réel d'habitants en région PACA (5 174 034) était bien inclus dans cet intervalle. Nous avions donc une certitude de 95% que la vraie valeur du nombre d'habitants en région PACA était incluse dans l'intervalle que nous avions déterminé grâce à notre échantillon de 100 communes. Enfin, nous avons enregistré les résultats de nos 10 tirages d'échantillons dans un tableau.

Résultats de l'échantillonnage aléatoire stratifié :

Population estimée	Binf	bsup	Marge	Population Exacte
4169627,7	2820189,32	5519066,08	1349438,38	5174034
5035288,7	3429296,75	6641280,65	1605991,95	5174034
5625865	4257872,84	6993857,16	1367992,16	5174034
4164042,8	3049137,23	5278948,37	1114905,57	5174034
5573074	3865497,36	7280650,64	1707576,64	5174034
5032143,5	3478909	6585378	1553234,5	5174034
6195011,8	4572190,36	7817833,24	1622821,44	5174034
4689611,7	2292475,79	7086747,61	2397135,91	5174034
4316585,1	3226589,82	5406580,38	1089995,28	5174034
4876424,7	4557049,37	7073560,61	2197135,91	5174034

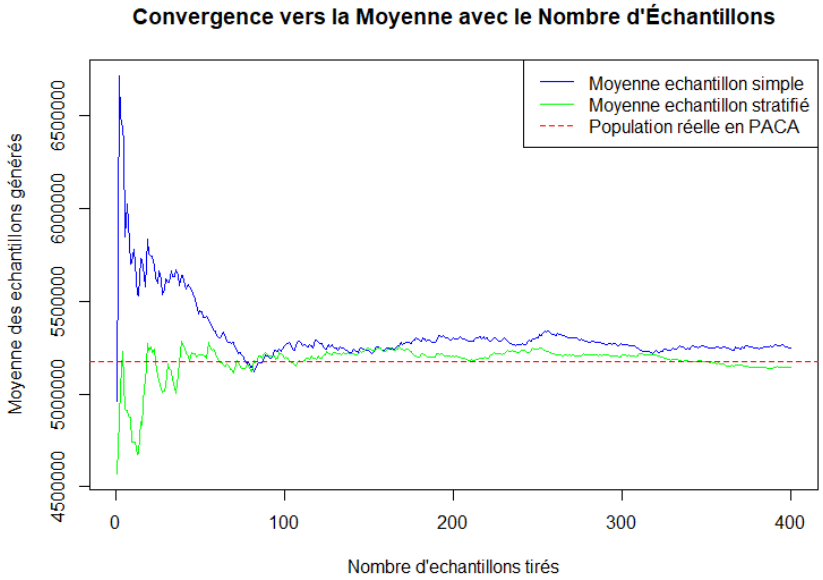


Conclusion partie 1

Pour conclure, l'échantillonnage stratifié s'est montré plus efficace dans notre étude car il prend en compte les différences entre les sous-groupes de la population, assurant ainsi une représentation équilibrée de chaque strate et améliorant la précision de nos résultats. Contrairement à l'échantillonnage aléatoire simple, l'échantillonnage stratifié permet une analyse plus précise de la population dans son ensemble. Bien que cela demande des données supplémentaires et une étape de stratification, les avantages en termes de précision et de représentativité sont significatifs.

Cependant, il est important de noter que nos résultats ne sont pas encore parfaitement précis. Pour obtenir une estimation encore plus exacte de la population réelle, une division en strates de centiles aurait pu fournir des résultats bien plus représentatifs.

Autre graphique : loi des grands nombres



La loi des grands nombres affirme que plus on tire d'échantillons, plus la moyenne de ces échantillons se rapproche de la moyenne de la population totale. Les échantillons stratifiés, en regroupant la population selon des critères spécifiques, convergent plus rapidement vers cette moyenne que les échantillons aléatoires simples. Cette efficacité accrue résulte de la représentativité plus équilibrée des différentes strates de la population. Ainsi, les échantillons stratifiés captent plus efficacement la variabilité de la population, améliorant la précision des estimations statistiques. Cette méthodologie combinée à la loi des grands nombres assure des résultats fiables et significatifs dans les études statistiques, favorisant une meilleure compréhension des phénomènes étudiés.

Partie 2 : Test du khi-deux d'indépendance

Dans cette partie du code, nous avons importé les données depuis le fichier "voitures.csv" en utilisant la fonction `read.csv2`. Nous avons également créé une nouvelle variable qualitative appelée `Prime2` à partir de la variable quantitative `Prime`. La fonction `ifelse` nous a permis d'attribuer les valeurs "faible", "moyenne" ou "élevée" à `Prime2` en fonction des valeurs de `Prime`, afin de catégoriser les primes d'assurance en classes distinctes pour les analyses ultérieures.

```
voitures = read.csv2("voitures.csv")
head(voitures)

voitures$Prime2 <- ifelse(voitures$Prime < 200, "faible", ifelse(voitures$Prime >= 200 & voitures$Prime < 400, "moyenne", "élevée"))
head(voitures)
```

Nous avons ensuite construit des tableaux croisés entre la nouvelle variable `Prime2` et les autres variables qualitatives du jeu de données, à l'exception de la variable "ville". Ces tableaux nous permettent de visualiser la répartition des différentes classes de primes en fonction des caractéristiques des voitures. Cette étape est essentielle pour préparer les données aux tests d'indépendance du khi-deux qui suivent, en fournissant une vue d'ensemble des relations possibles entre les primes d'assurance et les caractéristiques des véhicules.

```
Tabl_Puissfisc <- table(voitures$PuissFisc, voitures$Prime2)
Tabl_Puissfisc

Tabl_Categorie <- table(voitures$Categorie, voitures$Prime2)
Tabl_Categorie

Tabl_Anciennete <- table(voitures$Anciennete, voitures$Prime2)
Tabl_Anciennete

Tabl_Formule <- table(voitures$Formule, voitures$Prime2)
Tabl_Formule

Tabl_Marque <- table(voitures$Marque, voitures$Prime2)
Tabl_Marque
```

Nous avons ensuite effectué des tests d'indépendance du khi-deux pour examiner les relations entre les différentes variables qualitatives et la nouvelle variable Prime2. Pour chaque variable nous avons calculé le test du khi-deux et extrait la p valeur correspondante. Ces tests nous ont permis de déterminer si les différences observées dans les tableaux croisés étaient significatives, indiquant ainsi une éventuelle dépendance entre les primes d'assurance et les caractéristiques des véhicules. En analysant les valeurs p, nous avons pu identifier quelles relations étaient statistiquement significatives, ce qui est crucial pour les étapes d'analyse suivantes.

```
khi2_Puissfisc <- chisq.test(Tabl_Puissfisc)
khi2_Puissfisc
khi2_Puissfisc$p.value

khi2_Categorie <- chisq.test(Tabl_Categorie)
khi2_Categorie
khi2_Categorie$p.value

khi2_Anciennete <- chisq.test(Tabl_Anciennete)
khi2_Anciennete
khi2_Anciennete$p.value

khi2_Formule <- chisq.test(Tabl_Formule)
khi2_Formule
khi2_Formule$p.value

khi2_Marque <- chisq.test(Tabl_Marque)
khi2_Marque
khi2_Marque$p.value
```

Nous avons également défini une fonction pour calculer le coefficient de V de Cramer, un indice de force de l'association entre deux variables. Cette fonction prend en entrée un tableau croisé de deux variables qualitatives et renvoie le coefficient de V de Cramer correspondant. En utilisant cette fonction, nous avons calculé le coefficient de V de Cramer pour chaque paire de variables croisées avec la variable Prime2. Ces coefficients nous permettent d'évaluer la force de la relation entre les variables et d'identifier les associations les plus pertinentes pour notre analyse.

```r
# Définir une fonction pour calculer le V de Cramer
v_cramer <- function(table) {
  n <- sum(table)
  p <- nrow(table)
  q <- ncol(table)
  m <- min(p - 1, q - 1)
  chisq_result <- chisq.test(table)
  cramer_v <- sqrt(chisq_result$statistic / (n * m))
  return(cramer_v)
}

# Tester le V de Cramer sur chaque variable
v_Puissfisc <- v_cramer(Tabl_Puissfisc)
v_Categorie <- v_cramer(Tabl_Categorie)
v_Anciennete <- v_cramer(Tabl_Anciennete)
v_Formule <- v_cramer(Tabl_Formule)
v_Marque <- v_cramer(Tabl_Marque)
```

Enfin, nous avons créé un tableau récapitulatif contenant les valeurs du coefficient de V de Cramer ainsi que les p valeurs associées à chaque test du khi-deux. Ce tableau nous permet de visualiser aisément les résultats de nos analyses et de déterminer les associations significatives entre les variables, voici nos resultats :

```r
# Construire un tableau récapitulatif avec les valeurs de V de Cramer et de p
tableau_v_cramer <- data.frame(
  Test = c("Marque", "Puiss.fisc", "Catégorie", "Ancienneté", "Formule"),
  V_de_Cramer = c(v_Puissfisc, v_Marque, v_Categorie, v_Anciennete, v_Formule),
  P_value = c(khi2_Puissfisc$p.value, khi2_Marque$p.value, khi2_Categorie$p.value, khi2_Anciennete$p.value, khi2_Formule$p.value)
)
```

```
> tableau_v_cramer
   Variables V_de_Cramer     P_valeur
1     Marque  0.05335323 4.636742e-01
2 Puiss.fisc  0.30956085 1.773080e-21
3  Catégorie  0.08914343 7.243220e-02
4 Ancienneté  0.19816182 1.371920e-08
5    Formule  0.15320650 1.768714e-03
```

Conclusion test du khi2

Grace à nos résultats, nous pouvons affirmer que les trois variables les moins significatives, sont `Marque`, `Catégorie`, et `Formule`. Ces trois variables ont des valeurs de V de Cramer relativement faibles, ce qui indique une faible association avec la prime. De plus, leurs valeurs p sont soit bien au-dessus du seuil de 0.05 (`Marque` et `Catégorie`), soit légèrement en dessous (`Formule`). Cela signifie que, statistiquement, il n'y a pas de relation significative entre ces variables et le niveau de prime d'assurance, que les différences observées dans les données pourraient être dues au hasard plutôt qu'à une réelle association entre les variables, ou que cette relation est très faible. En pratique, cela suggère que les assureurs ne prennent pas autant en compte la marque, la catégorie des voitures, ou la formule d'assurance lorsqu'ils déterminent le prix des primes.

Les deux variables les plus significatives sont `Puiss.fisc` et `Ancienneté`. La puissance fiscale des voitures a une association forte avec le prix de la prime, comme le montre la

valeur relativement élevée de V de Cramer. La valeur p extrêmement faible indique que cette relation est assez significative. Cela signifie que les assureurs prennent en compte la puissance fiscale des voitures de manière notable lors de la détermination des primes. Une puissance fiscale plus élevée peut indiquer une voiture plus performante et potentiellement plus coûteuse à réparer ou à remplacer, ce qui justifie le prix d'une prime plus élevée

L'ancienneté des voitures a également une association modérée avec les niveaux de prime, avec une valeur de V de Cramer significative. La p valeur très faible montre que cette relation est statistiquement significative. Cela suggère que les assureurs considèrent l'ancienneté des voitures comme un facteur important lors de la fixation des primes. En général, les voitures plus anciennes peuvent être plus sujettes à des pannes ou nécessiter plus d'entretien, ce qui peut augmenter le risque pour l'assureur, ce qui justifie une prime plus élevée.

Pour conclure, les assureurs semblent accorder plus d'importance à la puissance fiscale et à l'ancienneté des voitures lorsqu'ils déterminent le prix des primes d'assurance, que à la marque, la catégorie, ou la formule d'assurance. Cela peut s'expliquer par le fait que la puissance fiscale et l'ancienneté sont plus directement liées aux coûts potentiels de réparation, de remplacement et aux risques globaux associés à l'assurance de ces véhicules.

Annexe 1 : code R partie 1

```
library(survival)
library(sampling)
library(stringr)
table = read.csv2("population_francaise_communes.csv", sep =";", dec=",", header=TRUE)

paca <- subset(table, Nom.de.la.région == "Provence-Alpes-Côte d'Azur")
paca <- paca[,c("Code.département", "Commune", "Population.totale")]
head(paca)
paca$Code.département=as.numeric(paca$Code.département)
paca$Population.totale=str_remove_all(paca$Population.totale, " ")
paca$Population.totale=as.numeric(paca$Population.totale)

#ensemble des communes en PACA
U <- paca$Commune
#Nombre de communes en PACA
N <- length(U)
N
#Le nombre total d'habitants en PACA (soit 5 174 034)
T= sum(paca$Population.totale)
T

# Echantillonage aleatoire simple
# Initialisation du dataframe pour stocker les résultats
resultats <- data.frame(Pop_estimee = numeric(),
                        Binf = numeric(),
                        Bsup = numeric(),
                        Marge = numeric(),
                        Pop_exacte = numeric())

# Répétition du processus 10 fois
for (i in 1:10) {
  # Tirage aléatoire simple d'un échantillon de taille n=100
  n = 100
  E = sample(U, n)
```

```
 # récupération des données des communes tirées dans l'échantillon
 paca1 <- paca[paca$Commune %in% E, ]
 # nb d'habitants des communes de l'échantillon
 paca2 <- subset(paca1, select=c(Commune, Population.totale))
 xbar <- mean(paca2$Population.totale)
 # idc de mu
 idcmoy <- t.test(paca2$Population.totale)$conf.int
 # estimation nb total hab
 T_est <- N * xbar
 # idc de T
 idcT <- idcmoy * N
 # Calcul de la marge d'erreur
 marge <- (idcT[2] - idcT[1]) / 2
 # Stockage des résultats dans le dataframe
 new_row = data.frame(Pop_estimee = T_est,
                      Binf = idcT[1],
                      bsup = idcT[2],
                      Marge = marge,
                      Pop_exacte = T)
 resultats = rbind(resultats,new_row)
}

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

###Echantillonage aleatoire stratifié

#quantiles de la variable Population.totale
summary(paca$Population.totale)
#création de déciles de la variable Population.totale
deciles <- quantile(paca$Population.totale, probs = seq(0, 1, by = 0.1))
paca$strate <- cut(paca$Population.totale, breaks = deciles, labels = 1:10, include.lowest = TRUE)
pacastrat <- paca[, c("Commune", "Population.totale", "strate")]
Nh <- table(pacastrat$strate)

resultats2 <- data.frame(Tstr = numeric(),
```

```
                Binf = numeric(),
                Bsup = numeric(),
                Marge = numeric(),
                Pop_exacte = numeric())

nh <- rep(10, 10)

for (i in 1:10) {
  st <- strata(pacastrat, stratanames = "strate", size = nh, method = "srswor")
  paca1 <- getdata(pacastrat, st)

  gh <- Nh / N
  fh <- nh / Nh

  # moyenne et variance de chaque strate
  ech1 <- paca1[paca1$strate == 1, ]
  ech2 <- paca1[paca1$strate == 2, ]
  ech3 <- paca1[paca1$strate == 3, ]
  ech4 <- paca1[paca1$strate == 4, ]
  ech5 <- paca1[paca1$strate == 5, ]
  ech6 <- paca1[paca1$strate == 6, ]
  ech7 <- paca1[paca1$strate == 7, ]
  ech8 <- paca1[paca1$strate == 8, ]
  ech9 <- paca1[paca1$strate == 9, ]
  ech10 <- paca1[paca1$strate == 10, ]

  m1 <- mean(ech1$Population.totale)
  m2 <- mean(ech2$Population.totale)
  m3 <- mean(ech3$Population.totale)
  m4 <- mean(ech4$Population.totale)
  m5 <- mean(ech5$Population.totale)
  m6 <- mean(ech6$Population.totale)
  m7 <- mean(ech7$Population.totale)
  m8 <- mean(ech8$Population.totale)
  m9 <- mean(ech9$Population.totale)
  m10 <- mean(ech10$Population.totale)
```

```
var1 <- var(ech1$Population.totale)
var2 <- var(ech2$Population.totale)
var3 <- var(ech3$Population.totale)
var4 <- var(ech4$Population.totale)
var5 <- var(ech5$Population.totale)
var6 <- var(ech6$Population.totale)
var7 <- var(ech7$Population.totale)
var8 <- var(ech8$Population.totale)
var9 <- var(ech9$Population.totale)
var10 <- var(ech10$Population.totale)

# moyenne géénérale des 10 échantillons
Xbarst <- sum(Nh * c(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10)) / N
# variance de la moyenne générale
varXbarst <- sum((gh^2) * (1 - fh) * c(var1, var2, var3, var4, var5, var6, var7, var8, var9, var10) /
nh)
# calcul de l'intervalle de confiance à 95% pour Xbarst
alpha <- 0.05
binf <- Xbarst - qnorm(1 - alpha / 2) * sqrt(varXbarst)
bsup <- Xbarst + qnorm(1 - alpha / 2) * sqrt(varXbarst)
idcmoy <- c(binf, bsup)

# estimation du nombre total d'habitants
Tstr <- N * Xbarst

# estimation du nombre total T grâce à un idc
binf <- idcmoy[1] * N
bsup <- idcmoy[2] * N
idcT <- c(binf, bsup)

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

new_row <- data.frame(Tstr = Tstr,
                      Binf = idcT[1],
                      Bsup = idcT[2],
```

```
                    Marge = marge,
                    Pop_exacte = T)
  resultats2 <- rbind(resultats2, new_row)
}

print(resultats2)
write.csv(resultats2, "resultats_stratified_sampling.csv", row.names = FALSE)
```

Annexe 2 : code 2 partie 2

```
voitures = read.csv2("voitures.csv")
head(voitures)
voitures$Prime2 <- ifelse(voitures$Prime < 200, "faible", ifelse(voitures$Prime >= 200 & voitures$Prime < 400, "moyenne", "élevée"))
head(voitures)

Tabl_Puissfisc <- table(voitures$PuissFisc, voitures$Prime2)
Tabl_Puissfisc
Tabl_Categorie <- table(voitures$Categorie, voitures$Prime2)
Tabl_Categorie
Tabl_Anciennete <- table(voitures$Anciennete, voitures$Prime2)
Tabl_Anciennete
Tabl_Formule <- table(voitures$Formule, voitures$Prime2)
Tabl_Formule
Tabl_Marque <- table(voitures$Marque, voitures$Prime2)
Tabl_Marque



khi2_Puissfisc <- chisq.test(Tabl_Puissfisc)
khi2_Puissfisc
khi2_Puissfisc$p.value
khi2_Categorie <- chisq.test(Tabl_Categorie)
khi2_Categorie
khi2_Categorie$p.value
khi2_Anciennete <- chisq.test(Tabl_Anciennete)
khi2_Anciennete
```

```
khi2_Anciennete$p.value
khi2_Formule <- chisq.test(Tabl_Formule)
khi2_Formule
khi2_Formule$p.value
khi2_Marque <- chisq.test(Tabl_Marque)
khi2_Marque
khi2_Marque$p.value

# Définir une fonction pour calculer le V de Cramer
v_cramer <- function(table) {
  n <- sum(table)
  p <- nrow(table)
  q <- ncol(table)
  m <- min(p - 1, q - 1)
  chisq_result <- chisq.test(table)
  cramer_v <- sqrt(chisq_result$statistic / (n * m))
  return(cramer_v)
}

# Tester le V de Cramer sur chaque variable
v_Puissfisc <- v_cramer(Tabl_Puissfisc)
v_Categorie <- v_cramer(Tabl_Categorie)
v_Anciennete <- v_cramer(Tabl_Anciennete)
v_Formule <- v_cramer(Tabl_Formule)
v_Marque <- v_cramer(Tabl_Marque)

# Construire un tableau récapitulatif avec les valeurs de V de Cramer et de p
tableau_v_cramer <- data.frame(
  Test = c("Marque", "Puiss.fisc", "Catégorie", "Ancienneté", "Formule"),
  V_de_Cramer = c(v_Puissfisc, v_Marque, v_Categorie, v_Anciennete, v_Formule),
  P_value = c(khi2_Puissfisc$p.value, khi2_Marque$p.value, khi2_Categorie$p.value,
khi2_Anciennete$p.value, khi2_Formule$p.value)
)

# Afficher le tableau récapitulatif
tableau_v_cramer
```