GASSE Armand-Valentin

BENOIT Nathaël

SAE - Echantillonnage et Estimation

La Bretagne

Partie 1 : Estimation du nombre d'habitants d'une région de France : La Bretagne :

Partie 1.1 : Echantillonnage aléatoire simple :

Ce travail, mené à l'aide du logiciel R, vise à développer une compréhension approfondie de l'incertitude et de la précision associées à l'estimation d'une variable mesurable au sein d'une population, et ce par le biais de la construction d'intervalles de confiance fondés sur des techniques d'échantillonnage. Dans un premier temps, l'approche adoptée repose sur un échantillonnage aléatoire simple à probabilités égales, dans lequel chaque individu de la population dispose de la même chance d'être sélectionné. Dans un second temps, l'étude se poursuit par l'application d'un échantillonnage stratifié, méthode permettant de segmenter la population en sous-groupes homogènes afin d'améliorer la précision des estimations.

Pour commencer, nous avons filtré les régions afin de ne sélectionner que celle sur laquelle nous allons travailler. Ainsi, nous allons présenter notre recherche sur la Bretagne.

Nous avons débuté par l'importation du jeu de données qui regroupe toutes les populations par départements et communes de la France. Nous avons donc stocké toutes les données du fichier dans un data frame. Nous avons également intégré la librairie « sampling » qui nous a permis de procéder à des tirages aléatoires sur les communes.

```r
# Importation de librairie
library(sampling)

# Importation des données du fichier CSV
data <- read.csv2("communes.csv",sep=";",dec=",",header=TRUE, fileEncoding = "latin1")
Estimations <- data.frame(Total= numeric(),Total_estime=numeric(), IDCinf=numeric(), IDCsup=numeric(), marge= numeric())

```

Nous avons choisi de travailler sur la région de la Bretagne, par conséquent nous avons extrait toutes les données relatives à cette région dans une table nommée données. Puis, nous avons choisi de garder uniquement trois variables (colonnes) : le code département, les communes et la population totale.

```r
## Partie 1.1
# Filtre pour avoir que les Code.département, Commune et Population.totale de la Bretagne
donnees <- data[data$Nom.de.la.région == "Bretagne",c("Code.département", "Commune", "Population.totale")]
donnees$Population.totale <- as.numeric(gsub(" ","",donnees$Population.totale))
# Affichage des 6 premières lignes
head(donnees)

```

L'utilisation du code département garantit que, même si certaines communes portent le même nom, elles seront identifiées distinctement. Ensuite nous avons converti la colonne "Population.totale" en une variable numérique, et en enlevant les espaces (1 200 devient 1200).

Par la suite, nous avons défini une variable appelée U qui représente toutes les communes de la région de Bretagne. Cette variable répertorie tous les noms des communes de cette région, qui est la plus peuplée du pays. Cette région possède 1 207 communes. Nous avons aussi calculé le nombre total d'habitants dans la région : la Bretagne abrite 12 384 734 résidents. Nous avons opté pour cette démarche car, ultérieurement, nous prévoyons d'évaluer la population de cette zone en utilisant un échantillon.

```
# Création d'une table contenant l'ensemble des communes et compte du nombre total
U = donnees$Commune
head(U)
N = length(U)

# Calcul du nombre total d'habitants
T = sum(donnees$Population.totale)
```

Par la suite, nous avons procédé à l'estimation de la population de la Bretagne en nous basant sur un échantillon aléatoire de 100 communes. Ce tirage nous a permis de constituer un data frame, contenant les informations de chaque commune tirée. Par la suite, nous avons déterminé le nombre moyen d'habitants par commune dans cet échantillon de 100 communes sélectionnées. Cela nous a donné la possibilité d'évaluer la dimension de la population dans cette région. Effectivement, étant donné que nous connaissons le nombre moyen d'habitants, le nombre total de communes et le nombre de communes incluses dans l'échantillon, nous sommes alors en mesure d'estimer le nombre d'habitants en Bretagne. Nous avons également effectué le calcul de l'IDC ainsi que de la marge d'erreur. En créant une boucle, nous avons réalisé ces calculs en utilisant 10 tirages aléatoires distincts.

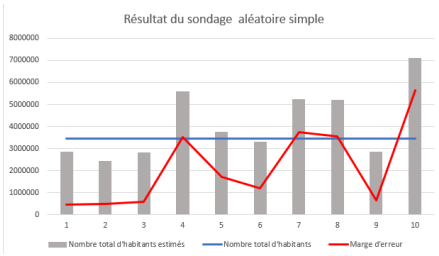
```
# Boucle pour réaliser les 10 estimations et les ajouter dans un data frame
for (i in 1:10) {
  # Tirage d'échantillon selon un sondage aléatoire simple
  n=100
  E=sample(U,n)
  length(E)
  head(E)

  # Nouvelle table contenant seulement les communes tirés
  donnees1 <- donnees[donnees$Commune %in% E,]

  # Nombre moyen d'habitants de l'échantillon E
  xbar=mean(donnees1$Population.totale)
  xbar

  #IDC à 95%
  idcmoy = t.test(donnees1$Population.totale)$conf.int
  idcmoy

  # Nombre d'habitants estimé
  T_est = N*xbar
  T_est

  # IDC de T
  idcT = idcmoy*N
  idcT

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

  # Ajouter une nouvelle ligne au data frame
  Estimations <- rbind(Estimations, data.frame(Total= T,Total_estime=T_est,
                                               IDCinf=idcT[1], IDCsup=idcT[2], marge= marge))
}
```

Les différents résultats montrent qu'il y a tout de même des écarts conséquents entre la valeur estimée et la valeur réelle, cela est dû à la taille des communes sélectionnées dans l'échantillon de 100

communes. Effectivement, on observe d'énormes disparités en termes de population entre les différentes communes.

Résultat échantillonnage aléatoire simple

	Nombre total d'habitants	Nombre total d'habitants estimés	IDC binf	IDC bsup	Marge d'erreur
1	3463439	2829317	2062076	2989125	463524
2	3463439	2411670	1920810	2902531	490860
3	3463439	2819422	2236355	3402489	583067
4	3463439	5582303	2066940	9097665	3515363
5	3463439	3736244	2037083	5435405	1699161
6	3463439	3286898	2085054	4488742	1201844
7	3463439	5221267	1483746	8958788	3737521
8	3463439	5194807	1634205	8755410	3560602
9	3463439	2830415	2176412	3484418	654003
10	3463439	7108544	1479841	12737246	5628703



Néanmoins, nous pouvons remarquer grâce au graphique que nous avons 3 estimations qui sont très proche de la réalité. Mais cela reste une petite part, par rapport aux 7 autres essais effectués.

Partie 1.2 : Echantillonnage aléatoire stratifié :

Par la suite, afin de minimiser autant que possible ces disparités, nous avons opté pour la création d'un échantillon de 100 communes en recourant à la méthode des quotas. Autrement dit, en organisant le tirage selon des normes qui reflètent les mêmes proportions que la population visée, i-e en créant des strates.

```
# Affichage des quartiles
summary(donnees$Population.totale)

# strates :
# On transfère le dataframe 'donnees' dans un autre dataframe appelé 'datastrat' pour préciser
# que l'on va manipuler les données avec les différentes strates de communes
datastrat = donnees
datastrat$strate = cut(donnees$Population.totale, breaks = c(0,716,1390,2762,230000), labels = c(1,2,3,4))
# Affichage des 6 premières lignes du nouveau data frame
head(datastrat)

# Sondage stratifié
datastrat = datastrat[order(datastrat$strate), ]
```

Nous avons ainsi débuté par la détermination des différents quartiles de notre groupe de population. Ces derniers nous ont permis de concevoir des strates entre les diverses municipalités de Bretagne, dans l'objectif d'estimer la population de manière plus précise. Nous avons donc 4 strates différentes qui sont, les communes de moins de 716 habitants, les communes entre 716 et 1390 habitants, les communes entre 1390 et 2762 habitants et pour finir toutes celles qui ont plus de 2762 habitants. Nous avons par la suite réalisé un data frame pour ordonner les communes selon leur strat respective.

Par la suite, nous avons calculé l'effectif et le poids des strates. Nous avons fait une boucle pour procédé à une sélection aléatoire de 100 communes issues des 4 strates distinctes. Nous avons sélectionné 25 municipalités pour chaque stratification. Le tirage aléatoire étant sans remise, nous avons calculé le taux de sondage (environ 0.08 < 0.1), et ces tirages peuvent alors être assimilé à des tirages avec remise. Ainsi, chaque commune est indépendante l'une de l'autre.

```
# Boucle pour réaliser les 10 estimations et les ajouter dans un data frame
for (i in 1:10) {
  # Tirage d'un échantillon stratifié de taille n = 100
  n =100
  nh = round(c(n*Nh[1]/N, n*Nh[2]/N, n*Nh[3]/N, n*Nh[4]/N))

  # taux de sondage dans les strates
  fh = nh/Nh
  fh
```

Ensuite, nous avons réalisé des tirages aléatoires simples dans chacune des strates puis calculer leurs moyennes et leurs variances respectives. En utilisant ces moyennes de différents échantillons, nous pourrons alors établir des estimations plus précises de la population de Bretagne par rapport aux premières estimations. En utilisant chaque moyenne des échantillons, nous avons été en mesure de déterminer la moyenne globale et la variance générale.

```
# Sondage strat (sans remsie dans les srates)
st = strata(datastrat, stratanames = c("strate"),size = nh, method = "srswr")
data1 = getdata(datastrat, st)
head(data1)
length((data1$Commune))

# Sous-échantillons
ech1 = data1[data1$strate == 1, ]
ech1
ech2 = data1[data1$strate == 2, ]
ech2
ech3 = data1[data1$strate == 3, ]
ech3
ech4 = data1[data1$strate == 4, ]
ech4

# Moyenne des sous-échantillons
m1 = mean(ech1$Population.totale)
m2 = mean(ech2$Population.totale)
m3 = mean(ech3$Population.totale)
m4 = mean(ech4$Population.totale)

# Moyenne des 4 sous échantillons réunis
Xbarst = (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4)/N

# Variances des 4 sous-échantillons
var1 = var(ech1$Population.totale)
var2 = var(ech2$Population.totale)
var3 = var(ech3$Population.totale)
var4 = var(ech4$Population.totale)
```

```
# Moyenne des 4 sous échantillons réunis
Xbarst = (Nh[1]*m1 + Nh[2]*m2 + Nh[3]*m3 + Nh[4]*m4)/N
# Estimation de la variance de 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])
```

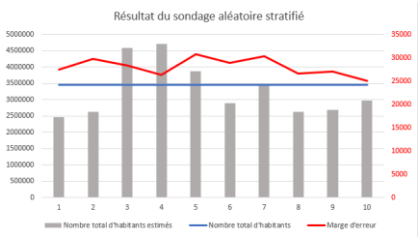
En utilisant chaque moyenne et variance des échantillons, nous avons été en mesure de calculer l'IDC pour μ à 95%. Pour finir, nous avons estimé la population de Bretagne, son IDC et la marge d'erreur à dix reprises avec une boucle.

```
# IDC pour μ à 95%
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 total de la population (T)
Tstr = N * Xbarst
Tstr

# Estimation de l'IDC pour T
binf = idcmoy[1] * N
bsup = idcmoy[2] * N
idcT = c(binf,bsup)
idcT

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

# Ajouter une nouvelle ligne au data frame
Estimations <- rbind(Estimations, data.frame(Total= T,Total_estime=Tstr,
                                             IDCinf=idcT[1], IDCsup=idcT[2], marge= marge))
}
```

On peut observer à travers le tableau et le graphique suivants que les résultats sont clairement plus précis que ceux de la première partie. Effectivement, l'établissement de strates a facilité l'équilibre entre les grandes et petites communes afin d'éviter toute disproportion.

Résultat du sondage aléatoire stratifié

	Nombre total d'habitants	Nombre total d'habitants estimés	IDC binf	IDC bsup	Marge d'erreur
1	3463439	2458369	2430921	2485818	27448
2	3463439	2622420	2592698	2652141	29721
3	3463439	4585705	4557415	4613996	28290
4	3463439	4711364	4685063	4737665	26300
5	3463439	3866149	3835422	3896876	30726
6	3463439	2884499	2855600	2913399	28899
7	3463439	3488394	3458099	3518688	30294
8	3463439	2618047	2591401	2644694	26646
9	3463439	2684300	2657296	2711303	27003
10	3463439	2977366	2952262	3002471	25104



Partie 2 : Traitement de données d'enquête :

L'objectif de cette partie est de réaliser le test du khi-deux d'indépendance afin de trouver des relations significatives entre la pratique sportive des étudiants et certaines de leurs caractéristiques.

Tout d'abord, nous avons débuté par l'importation du jeu de données qui regroupe l'ensemble des informations sur l'activité sportive des étudiants ainsi que d'autres données les concernant, que nous avons pu récupérer antérieurement grâce à un questionnaire. Ces informations ont été stockées dans un data frame.

```
data <- read.csv2("EnqueteSportEtudiant2024.csv",sep=";",dec=",",header=TRUE)
head(data)
```

Ce dernier contient 76 variables et 375 observations / réponses, et les individus sont les étudiants de l'IUT de Niort. Quant aux variables, on y retrouve le sexe, le département géographique, le département de formation, le niveau, et d'autres informations recueillies lors du questionnaire. Par ailleurs, les variables sont toutes qualitatives.

Nous avons ensuite décidé de croiser la variable 'sport' avec les variables suivantes : la sante, si l'étudiant est fumeur ou non, le sexe, le département de formation, l'alimentation et la bonification.

```
# Tableaux croisés dynamiques de la variable 'sport' avec

TCD_sante = table(data$sport, data$sante)
TCD_sante # Afficher le tableau
TCD_fumer = table(data$sport, data$fumer)
TCD_fumer # Afficher le tableau
TCD_sexe = table(data$sport, data$sexe)
TCD_sexe # Afficher le tableau
TCD_deptformation = table(data$sport, data$deptformation)
TCD_deptformation # Afficher le tableau
TCD_alimentation = table(data$sport, data$alimentation)
TCD_alimentation # Afficher le tableau
TCD_bonif = table(data$sport, data$bonif)
TCD_bonif # Afficher le tableau
```

Nous avons ensuite effectué le test d'indépendance du khi-deux entre le sport et les variables choisies.

```
# Test d'indépendance du khi-deux entre la variable sport et les autres variables qualitatives

khideux_sante = chisq.test(TCD_sante)
khideux_sante # Afficher le khi-deux
khideux_fumer = chisq.test(TCD_fumer)
khideux_fumer # Afficher le khi-deux
khideux_sexe = chisq.test(TCD_sexe)
khideux_sexe # Afficher le khi-deux
khideux_deptformation = chisq.test(TCD_deptformation)
khideux_deptformation # Afficher le khi-deux
khideux_alimentation = chisq.test(TCD_alimentation)
khideux_alimentation # Afficher le khi-deux
khideux_bonif = chisq.test(TCD_bonif)
khideux_bonif # Afficher le khi-deux
```

Précision importante : nous avons choisi arbitrairement de poser le niveau de risque alpha à 0,05.

Pour chaque khi-deux calculé, nous avons obtenu les p-valeurs suivantes :

p-value = 0.70 pour la variable 'sante'

p-value = 0.67 pour la variable 'fumer'

p-value = 0.0006 pour la variable 'sexe'

p-value = 0.0046 pour la variable 'deptformation'

p-value = 0.0002 pour la variable 'alimentation'

p-value = 0.0000000021 pour la variable 'bonif'

Nous avons donc pu déterminer le khi-deux théorique à l'aide de la table du khi-deux, et nous avons pu arriver aux conclusions suivantes :

Concernant la variable santé et la variable fumer, elles sont toutes les deux indépendantes de la variable sport au risque de 5%, car leur khi-deux lue (5,99) est supérieur à leur khi-deux calculé. En revanche, on rejette l'hypothèse d'indépendance pour la variable sexe au risque de 5%, car le khi-deux calculé (14,7) est supérieur au khi-deux lu (5,9). C'est aussi le cas pour le département de formation avec un khi-deux calculé de 18,8 et un khi-deux lue de 12,6. Le test est aussi significatif pour les variables alimentation et bonification, leur khi-deux calculé sont respectivement de 16,7 et de 46,3 tandis que leur khi-deux lu respectif sont 5,9 et 9,5. Ces 4 derniers tests sont donc significatifs, car nous pouvons rejeter l'hypothèse d'indépendance au risque alpha de 5%.

Nous avons ensuite souhaité déterminer le niveau de dépendance de ces 4 relations significatives. Ainsi nous avons calculé le V de Cramer pour la variable 'sexe', 'deptformation', 'alimentation' et 'bonif'.

```r
# Calcul du V de Cramer de la variable 'sexe', 'deptformation', 'alimentation' et 'bonif'

# Effectif total
n<-nrow(data)

# Nb de lignes
p <- nrow(TCD_sexe)
# Nb de colonnes
q <- ncol(TCD_sexe)
# Détemriner le minimum entre le nb de lignes et le nb de colonnes
m <- min(p-1,q-1)
# V de Cramer
V_sexe <- sqrt(khideux_sexe$statistic/(n*m))
V_sexe

p <- nrow(TCD_deptformation)
q <- ncol(TCD_deptformation)
m <- min(p-1,q-1)
# V de Cramer
V_deptformation <- sqrt(khideux_deptformation$statistic/(n*m))
V_deptformation

p <- nrow(TCD_alimentation)
q <- ncol(TCD_alimentation)
m <- min(p-1,q-1)
# V de Cramer
V_alimentation <- sqrt(khideux_alimentation$statistic/(n*m))
V_alimentation

p <- nrow(TCD_bonif)
q <- ncol(TCD_bonif)
m <- min(p-1,q-1)
# V de Cramer
V_bonif <- sqrt(khideux_bonif$statistic/(n*m))
V_bonif

```

Nous avons réalisé un tableau regroupant tous les V de Cramer de chaque test significatif, et avons rajouté une colonne dans laquelle est inscrit la valeur du V de Cramer le plus élevé.

```r
# Tableau des V de Cramer

# Création d'un tableau de données
dataCramer = data.frame()
# Insertion des valeurs des V de Cramer
dataCramer = rbind(dataCramer,c(V_sexe, V_deptformation, V_alimentation, V_bonif))
# On renomme les colonnes
colnames(dataCramer) <- c("V_sexe", "V_deptformation", "V_alimentation", "V_bonif")
# On ajoute une colonne affichant le V de Cramer le plus élevé
dataCramer$V_Cramer_Max = max(c(V_sexe, V_deptformation, V_alimentation, V_bonif))
# On affiche le tableau de données
dataCramer
```

Voici donc le contenu du tableau :

```
> # On affiche le tableau de données
> dataCramer
    V_sexe V_deptformation V_alimentation   V_bonif V_Cramer_Max
1 0.198274       0.1582276      0.2107832 0.2485422    0.2485422
> View(dataCramer)
```

Comme nous pouvons le constater, le sexe et le département de formation de l'étudiant ont tous les deux une faible liaison avec sa pratique sportive, puisque leur V de Cramer sont compris entre 0,10 et 0,20. Cependant, le V de Cramer de la variable 'alimentation' et 'bonif' est compris entre 0,20 et 0,25. Cela signifie que la pratique sportive des étudiants est modérément liée à leur alimentation, et au fait qu'ils aient choisi de réaliser une activité sportive en étant bonifié ou non.

Pour conclure, la pratique sportive des étudiants dépend légèrement du sexe de l'étudiant et de son département de formation, mais elle dépend davantage du choix de bonification de l'étudiant ainsi que de son alimentation.