Projet Sondage

Kuassi Pierre DOVODJI

2024-11-12

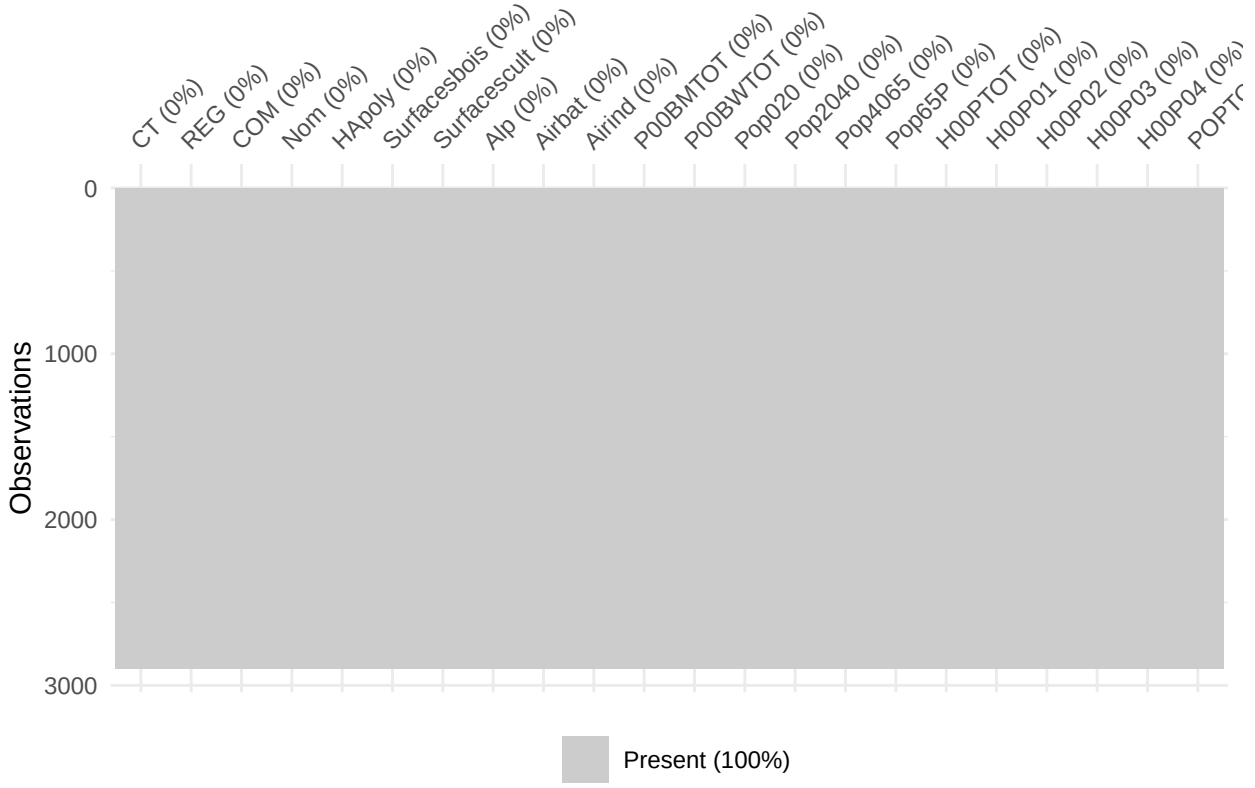
Contents

1 Introduction

Les forêts occupent une grande partie du territoire suisse et jouent un rôle important pour l'environnement. Ce projet a pour but d'estimer la superficie totale des forêts en Suisse en utilisant un échantillon de données tiré du jeu *swissmunicipalities* du package *sampling* de R. Ce jeu de données contient des informations sur 2 896 municipalités et 22 variables y compris la variable Surfacesbois, qui représente la superficie forestière notre variable d'interêt. Pour cela, nous mettrons en place un plan de sondage avec un échantillon de taille $n = 100$ et un estimateur précis pour minimiser les erreurs. En complément, nous proposerons une méthode pour estimer le ratio entre les surfaces forestières et les surfaces cultivées.

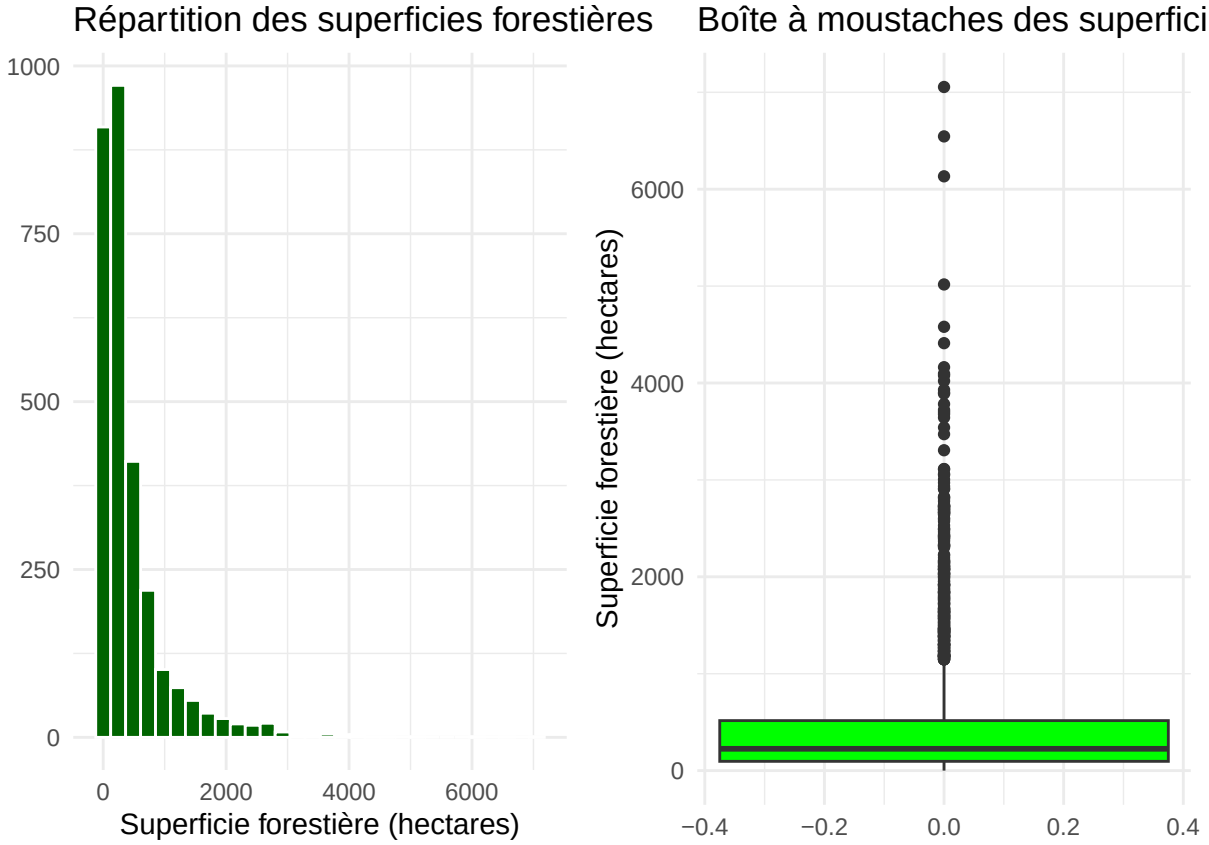
2 Statistique descriptive

2.1 Visualisation des données manquantes



On constate que les données ne contiennent aucune valeur manquante, garantissant ainsi leur intégralité pour les analyses.

2.2 Distribution de la variable d'interêt



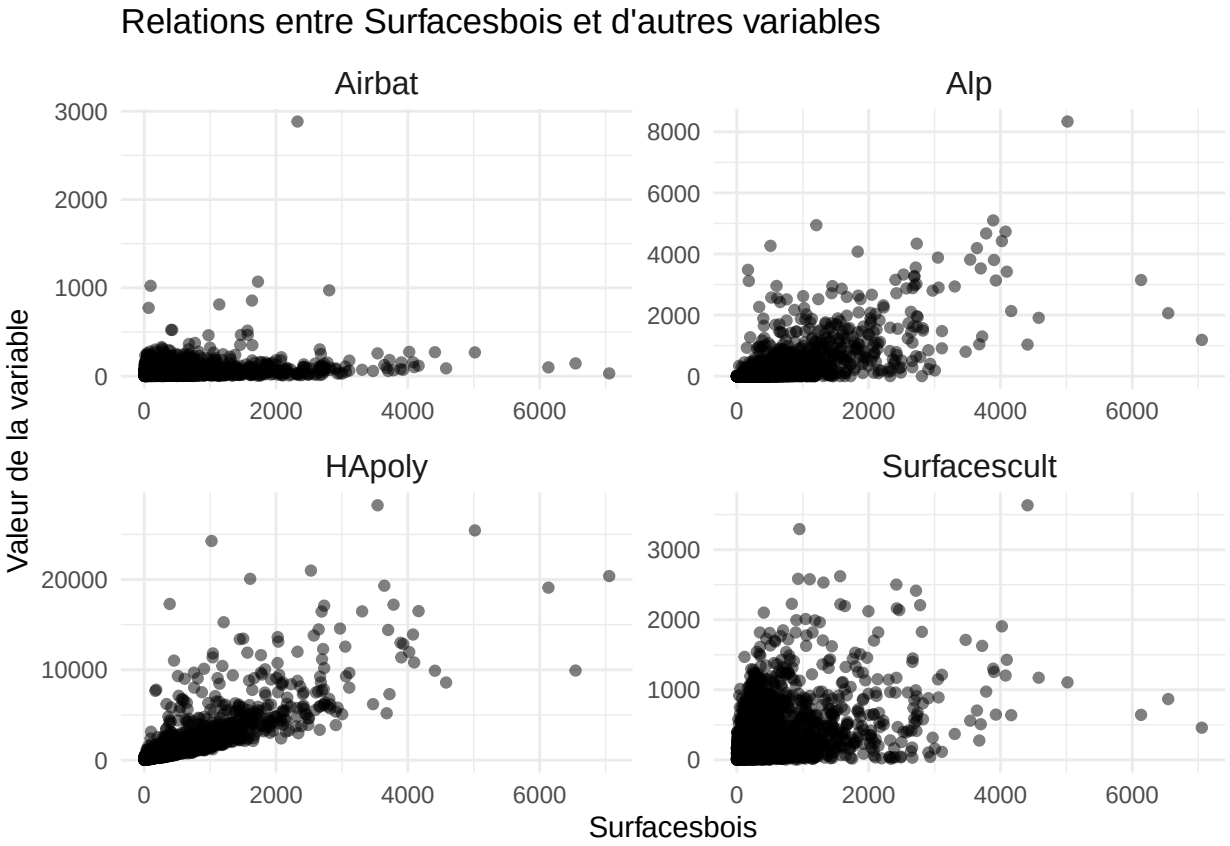
On observe ici qu'un grand nombre de municipalités possède une faible superficie de bois, et cette proportion diminue à mesure que la superficie augmente. Cette tendance pourrait indiquer des disparités dans la couverture forestière à travers les différentes municipalités. Quelles sont donc les facteurs/variables qui pourraient influencer la superficie boisée ? Nous répondrons à cette question dans la section suivante.

2.3 Corrélation

Ici nous analysons les corrélations linéaires entre notre variable d'intérêt et les autres variables, afin d'explorer leurs relations.

```
##              CT       REG         COM    HApoly Surfacescult       Alp
## [1,] 0.01077436 0.1762316 0.005213288 0.8094731    0.3700496 0.6991955
##         Airbat    Airind  P00BMTOT P00BWTOT    Pop020   Pop2040   Pop4065
## [1,] 0.2222185 0.1309974 0.1073639 0.104094 0.1207728 0.0995802 0.1011412
##        Pop65P    H00PTOT     H00P01     H00P02    H00P03    H00P04    P0PTOT
## [1,] 0.107509 0.09768084 0.08818849 0.09889159 0.1030843 0.1242905 0.1056776
```

On remarque que, notre variable d'interêt `Surfacesbois` présente une forte relation linéaire avec la variable `HApoly` (0.81) qui représente la superfice de la municipalité, une relation linéaire modérée avec `Alp` (0.70), et des relations modérées avec `Surfacescult` (0.37) et `Airbat` (0.22). Les autres variables, notamment démographiques et économiques, montrent des corrélations faibles ou négligeables avec Surfacesbois, indiquant qu'elles ont une influence limitée sur cette variable. Visualisons donc ses relations avec des graphiques.



Nous allons dans la suite cossidérer la variables la surface de la municipalié `HApoly` comme variable auxiliaire pour estimer la surface totale boisée qui est égale à 1270996 .

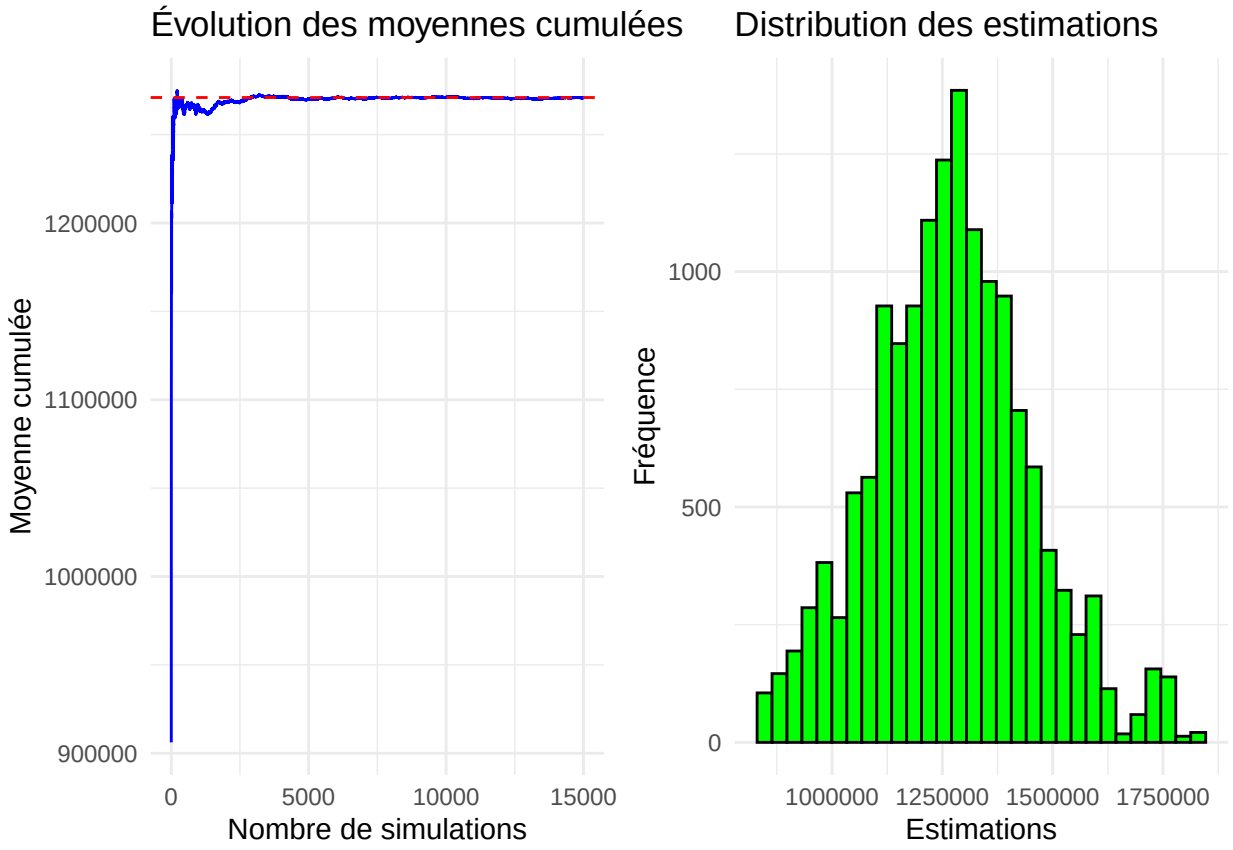
3 Estimation de la surface totale boisée

Pour estimer cette superficie, nous allons utilisé le songade systématique à probabilité égale et à probabilté inégale de taille fixe avec l'estimateur d'Horvitz-Thompson et aussi l'estimateur par ratio. Étant donné que nous avons constaté une répartition inégale, ces approches nous permettront de tenir compte des informations auxiliaires afin d'obtenir des échantillons équilibrés ce qui est très important pour obtenir une bonne estimation. Par la suite, nous évaluerons la performance et la qualité des différentes méthodes à l'aide de la variance, du coefficient de variation, du biais relatif et des simulations. Il s'agit de l'estimation d'un total. Pour ce faire, nous allons utiliser le package `survey` et sa fonction `svytotal()`, qui permet de calculer l'estimation du total d'une variable dans la population tout en tenant compte du plan d'échantillonnage. Pour réaliser l'estimation, nous devons définir un plan de sondage en créant un objet avec la fonction `svydesign()` du package `survey`, qui intègre les informations sur les unités échantillonnées, les poids, et, si nécessaire, la correction pour population finie (fpc).

3.1 Echantillonnage systématique (Modèle 1)

Nous mettons en place ici, le plan de sondage systématique qui est une méthode d'échantillonnage où les unités sont sélectionnées à intervalles réguliers dans une liste ordonnée de la population. Pour la mise en œuvre de ce plan, nous allons utiliser le package `sampling` et la fonction `UPsystematic()`, qui permet d'effectuer un tirage systématique à probabilités égales ou inégales. L'argument principal est `pik`, un vecteur de probabilités d'inclusion de taille N (taille de la population), où la somme des composantes `pik` correspond à la taille de l'échantillon souhaitée n. Pour un sondage systématique à probabilités égales, on définit une variable auxiliaire $x = 1$ et on utilise `pik` avec toutes ses composantes égales à n/N. Voici les résultats obtenus:

Métrique	Modèle2
CV	0.142
Bias Relative	0.000
TauxIC	0.907
Ecart-type	177993.172
Total Estimaté	1270820.318

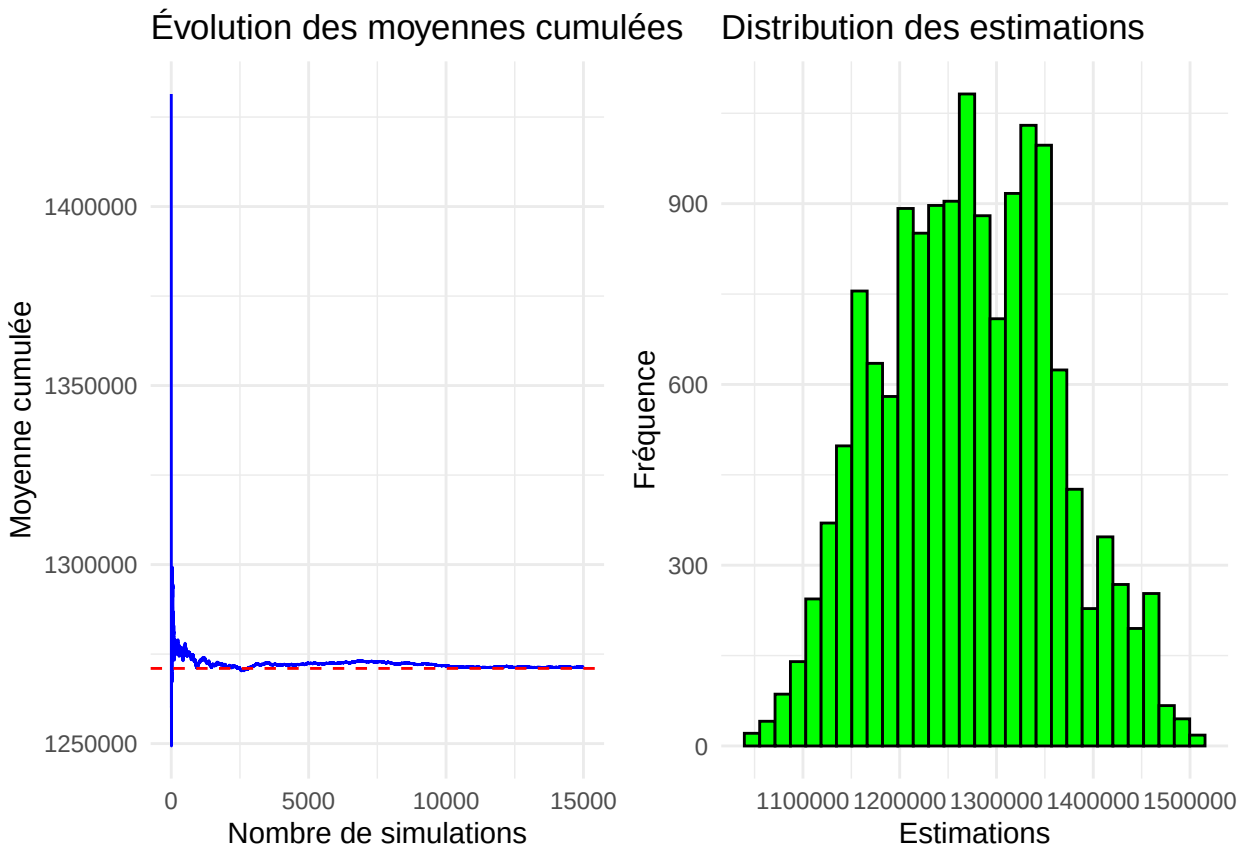


On obtient un coéficient de variation de 0.142, de biais relative nul avec le taux de couveture de l'intervalle de confiance 90.7%. Aussi la distribution de l'estimateur semble être normale ce qui est rassurant.

3.2 Echantillonnage systématique ordonné selon la variables `HApoly` (Modèle 2)

Ici nous utilisons l'échantillonnage systématique a probabilité égale mais avant d'effectuer l'échantillonage nous allons ordonné la table de donnée selon la variables `HApoly` ce qui pourrait amélioré les résultats précédents car cela nous permetra d'avoir un échantillon héterogène. Voici les résultats obtenus:

Métrique	Modèle2
CV	0.07
Bias Relative	0.00
TauxIC	1.00
Ecart-type	179004.51
Total Estimaté	1271342.43

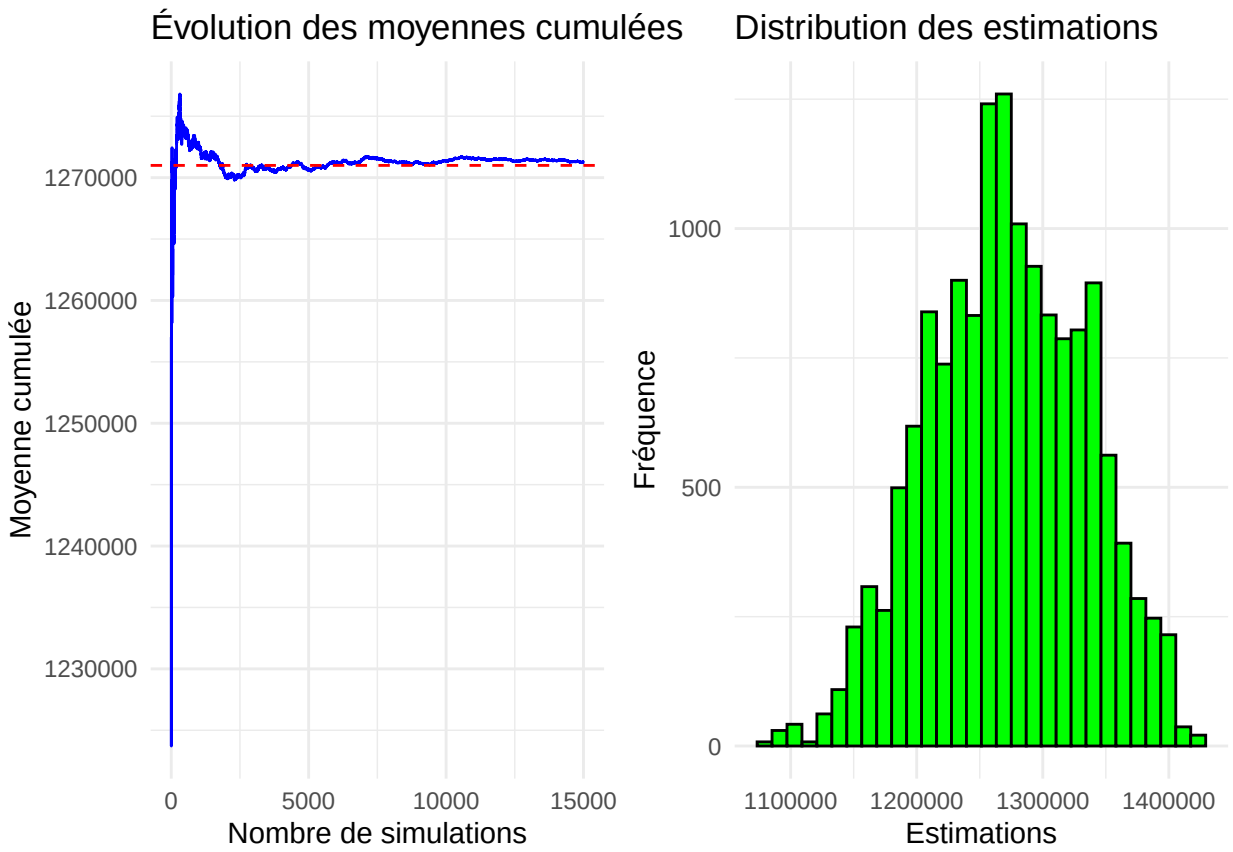


On obtient un coéficient de variation de 0.07, de biais relative nul avec le taux de couveture de l'intervalle de confiance 100%. Aussi la distribution de l'estimateur semble être normale. Ceci signifie que cette méthode est meilleur que la précédente. Néanoins on ne gagne quasiment rien en variance.

3.3 Echantillonnage systématique à probabilité inégale (Modèle 3)

Ici, nous utilisons l'échantillonnage systématique à probabilité inégale. Avant de réaliser l'échantillonnage, nous allons considéré la variable HApoly. Cette démarche nous permettra d'obtenir un échantillon plus hétérogène, car l'échantillonnage sera effectué proportionnellement aux valeurs de cette variable auxiliaire, offrant ainsi des probabilités d'inclusion inégales basées sur `HApoly`. Les probabilités d'inclusion proportionnelles à la variable `HApoly` sont données par $\pi_k = \frac{n * HApoly_k}{\sum_k^N HApoly_k}$ qui seront calculer l'aide de la fonction `inclusionprobabilities()`. Voici les résultats obtenus:

Metrique	Modèle3
CV	0.049
Bias Relative	0.000
TauxIC	0.975
Ecart-type	65456.841
Total Estimaté	1271254.482



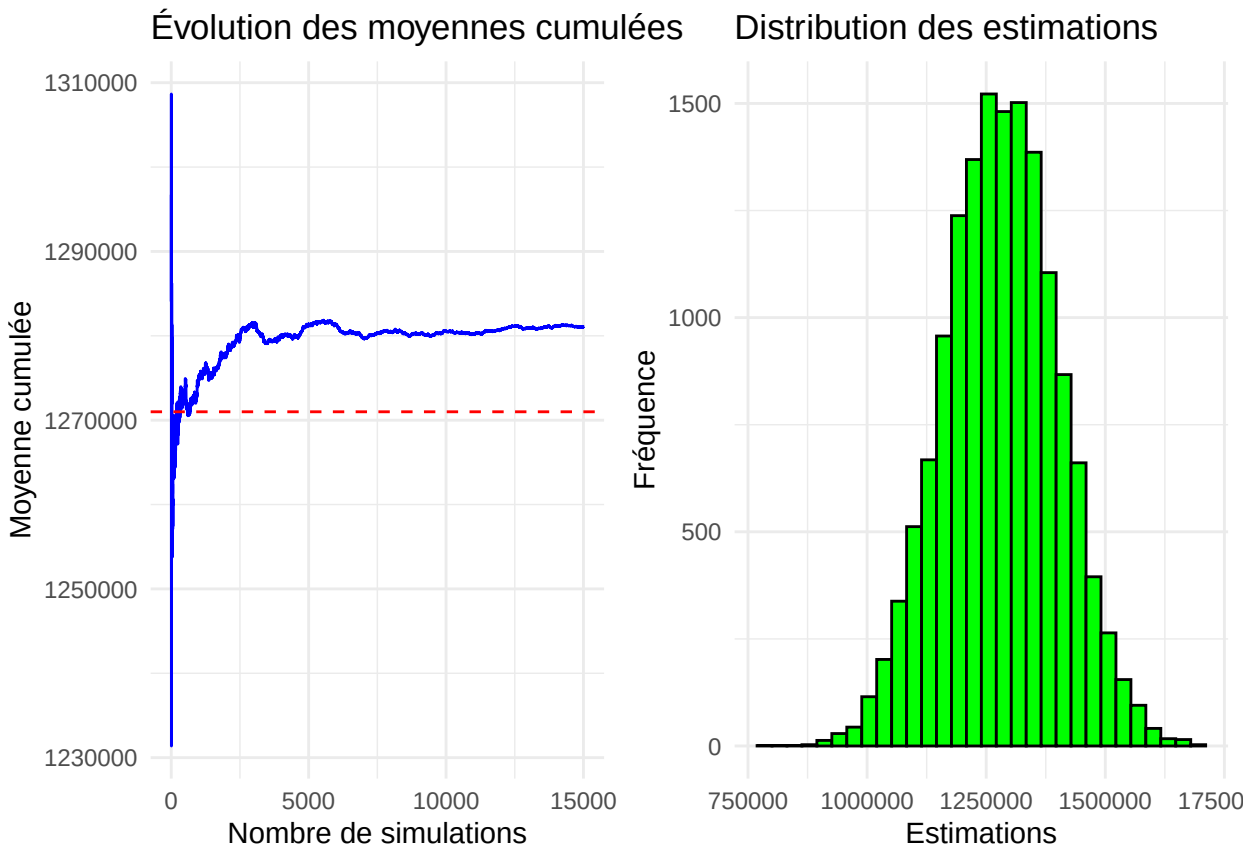
On obtient un coéficient de variation de 0.049, de biais relative nul avec le taux de couveture de l'intervalle de confiance 97.5%. Aussi la distribution de l'estimateur semble être normale. Nous gagnons énormément en variance. Ceci signifie que cette méthode est meilleur que les précédentes.

3.4 Estimation par le ratio (Modèle 4)

L'estimation par le ratio est une méthode couramment utilisée pour améliorer l'estimation d'une variable d'intérêt en utilisant une variable auxiliaire fortement corrélée ce qui est le cas ici des variables `Surfacesbois` et '`HApoly`. Cette approche permet d'estimer le total de la variable d'intérêt en exploitant le rapport entre les totaux de la variable d'intérêt et de la variable auxiliaire dans la population. Également, la visualisation de la relation linéaire entre ces deux variables a révélé un nuage de points en forme d'entonnoir, ce qui suggère que cette méthode pourrait être plus appropriée.

Pour estimer le total de SuperficieBois par le ratio, nous commençons par effectuer un échantillonnage aléatoire simple sans remise de taille n=100 à l'aide de la fonction `srswor()`. Ensuite, nous créons un objet de sondage avec `svydesign()`, en spécifiant les poids(même poids = N/n) d'échantillonnage et les corrections pour population finie. L'estimation du ratio est réalisée avec la fonction `svyratio()`, qui calcule la pente `R` entre `SuperficieBois` (numérateur) et `HApoly` (dénominateur). En utilisant predict(), nous obtenons l'estimation du total de SuperficieBois avec son écart-type associé. Voici les résultats obtenus:

Métrique	Modèle4
CV	0.093
Bias Relative	0.008
TauxIC	0.606
Ecart-type	111927.520
Total Estimaté	1281115.677



On voit que la distribution de l'estimateur est normale. Un coéficient de variation de 0.09, de biais relative quasi nul avec le taux de couveture de l'intervalle de confiance 61% qui est un peut faible. Aussi nous gagnons en variance comparativement au modèle 1 et 2. Ceci signifie que cette méthode est meilleur que les deux première en terme de variance mais l'estimateur est biaisé ce qui justifie l'éloignement de la vrai valeur de l'estimation.

3.5 Comparaison des quatres méthodes

Nous allons procceder ici à l'analyse des quatres modèles mis en place. Le tableau suivant résume les métriques des quatres modèles.

Métrique	Modèle1	Modèle2	Modèle3	Modèle4
CV	0.142	0.07	0.049	0.093
Bias Relative	0.000	0.00	0.000	0.008
TauxIC	0.907	1.00	0.975	0.606
Ecart-type	177993.172	179004.51	65456.841	111927.520
Total Estimaté	1270820.318	1271342.43	1271254.482	1281115.677

On remarque que le :

- Modèle 3 est le meilleur, combinant un CV faible, un biais nul, un excellent TauxIC et un écart-type minimal.
- Modèle 2 est une alternative solide, mais légèrement moins précis que le Modèle 3.
- Modèle 4 à une bonne précision (CV et écart-type), mais son TauxIC très faible (0,61) rend ses intervalles peu fiables qui est normale car c'est une méthode qui donne un estimateur biaisé.

- Modèle 1 à la précision la plus faible (CV élevé, écart-type important) et des intervalles de confiance insuffisants, ce qui en fait le moins performant.

En terme de variance le modele 3 est le meilleur en suite le model modele 4 et il s'en suit les modele 1 et 2 qui on pratiquement le même perfomence. Nous retenons donc le modèle 3 pour notre estimation.

4 Estimation du ratio

Pour estimer le ratio entre la surface occupée par des forêts et la surface cultivée, nous allons estimer les totaux de la surface occupée par des forêts et la surface cultivée et en suite faite le ratio. Pour le calcule de la variance on poura en supponsons que n est grand l'approximer en utilisant le variable linéarisé u_k, qui repose sur l'approximation de Taylor. Après avoir tiré un échantillon aléatoire simple sans remise, le plan d'échantillonnage est défini à l'aide de la fonction `svydesign()`. L'estimation du ratio est réalisée avec la fonction `svyratio()` du package `survey`, qui fournit à la fois la valeur estimée du ratio, son ecart type et un intervalle de confiance. Voici les résultats obtenus :

```
## Ratio estimator: svyratio.survey.design2(~Surfacesbois, ~Surfacescult, design)
## Ratios=
##              Surfacescult
## Surfacesbois     1.516765
## SEs=
##              Surfacescult
## Surfacesbois      0.18455
```

```
##                               2.5 %   97.5 %
## Surfacesbois/Surfacescult 1.155054 1.878476
```

```
## La valeur réelle est :  1.287323
```

Le ratio estimé entre les surfaces boisées (Surfacesbois) et les surfaces cultivées (Surfacescult) est de 1,5167, ce qui signifie qu'en moyenne, il y a environ 1,52 unités de surface boisée pour chaque unité de surface cultivée. L'écart type, qui mesure la précision de cette estimation, est de 0,18, ce qui indique que le résultat est fiable. L'intervalle de confiance à 95 % est de [1,15 ; 1,88], ce qui veut dire que le vrai ratio dans la population se trouve probablement dans cette fourchette. La valeur exacte, calculée comme 1,2873, est très proche de l'estimation, ce qui confirme la solidité du résultat. On pourra confirmé ses résultats en faisant des simulations.

5 Conclusion

Ce projet nous a permis d'explorer différentes méthodes d'échantillonnage et d'estimation pour analyser les relations entre des variables d'intérêt, comme les surfaces boisées et cultivées, à partir de données partielles. Nous avons comparé des approches telles que l'estimation par le ratio et le plan systématique dans ses différent variétés, en évaluant leur performance à travers des métriques comme le biais relatif, le coefficient de variation, le taux de couverture des intervalles de confiance, l'écart-type et des simulation. Les résultats montrent que le choix de la méthode dépend de la nature des données et de la relation entre les variables. L'estimation par le ratio avec un plan SAS s'est révélée un peu efficace avec une faible variance mais l'estimation est un peu biaisé. Par contre l'estimateur de HT obtenu avec le plan systématique avec probabilité d'inclusion proportionnelle à la variable `HApoly` donne des résultats exceptionnels. Ce travail met en lumière l'importance d'un choix judicieux du plan d'échantillonnage et des méthodes d'estimation pour garantir des résultats fiables.

6 Annexe - Code R

```r
library(sampling)
library(visdat)
library(ggplot2)
library(tidyr)
library(dplyr)
require(survey)
require(tidyverse)

data(swissmunicipalities)
df <- swissmunicipalities
vis_miss(df)

## Distribution de la variable d'interêt

hist <- ggplot(data = df, aes(x = Surfacesbois)) +
  geom_histogram(bins = 30, fill = "darkgreen", color = "white") +
  labs(title = "Répartition des superficies forestières",
       x = "Superficie forestière (hectares)",
       y = " ") +
  theme_minimal()

# Graphique de la boîte à moustaches des superficies forestières
box <- ggplot(data = df, aes(y = Surfacesbois)) +
  geom_boxplot(fill = "green") +
  labs(title = "Boîte à moustaches des superficies forestières",
       y = "Superficie forestière (hectares)") +
  theme_minimal()

# Disposition 1x2 avec grid.arrange()
gridExtra::grid.arrange(hist, box, nrow = 1)

corr <- cor(df[,-4])
t(corr[5,-5])
# corrplot::corrplot(corr)

# Convertir les données au format long pour ggplot
df_long <- df %>%
  select(Surfacesbois, HApoly, Alp, Airbat, Surfacescult) %>%
  gather(key = "Variable", value = "Value", -Surfacesbois)


ggplot(df_long, aes(x = Surfacesbois, y = Value)) +
  geom_point(alpha = 0.5) +
  #geom_smooth(method = "lm", se = FALSE, color = "blue") +
  facet_wrap(~Variable, nrow = 2, ncol = 2, scales = "free") +
  theme_minimal() +
  labs(
    title = "Relations entre Surfacesbois et d'autres variables",
    x = "Surfacesbois",
    y = "Valeur de la variable"
  ) +
```

```r
  theme(strip.text = element_text(size = 12))




plot_cum_moy_avec_histogram <- function(estimates, total, n.sim) {
  x <- seq(1, n.sim, by = 1)
  m_cum <- sapply(x, function(i) mean(estimates[1:i]))

  data_cum <- data.frame(Simulation = x, MeanCumulative = m_cum)

  plot_c <- ggplot(data_cum, aes(x = Simulation, y = MeanCumulative)) +
    geom_line(color = "blue") +
    geom_hline(yintercept = total, color = "red", linetype = "dashed") +
    labs(
      title = "Évolution des moyennes cumulées",
      x = "Nombre de simulations",
      y = "Moyenne cumulée"
    ) +
    theme_minimal()

  plot_hist <- ggplot(data.frame(Estimates = estimates), aes(x = Estimates)) +
    geom_histogram(bins = 30, fill = "green", color = "black") +
    labs(
      title = "Distribution des estimations",
      x = "Estimations",
      y = "Fréquence"
    ) +
    theme_minimal()
  gridExtra::grid.arrange(plot_c, plot_hist, ncol = 2)
}
verifIC <- function(IC,total,n.sim=15000){
  verif <- rep(total,n.sim) >= IC[,1] & rep(total,n.sim) <= IC[,2]
  return(mean(verif))
}

set.seed(sample(1,10^6,1))
total <- sum(df$Surfacesbois)
n <- 100          # Taille de l'échantillon
N <- nrow(df)  # Taille de la population
pik <- rep(n / N, N)

# Tirage de l'échantillon systématique
sy <- UPsystematic(pik)
ech.sy <- df[sy == 1, ]

# Estimation du total pour la variable Surfacesbois
design.sy <- svydesign(ids = ~1, weights = rep(N / n, n), data = ech.sy)
est_total_sy <- svytotal(~Surfacesbois, design.sy)
#est_total_sy

#Étude par simulations
# Paramètres pour les simulations
```

```r
n.sim <- 15000
est.sy <- numeric(n.sim)
IC.sy <- matrix(1,n.sim,2)
var.sy <- numeric(n.sim)
# Simulations

for (i in 1:n.sim) {
  sy <- UPsystematic(pik)
  ech.sy <- df[sy == 1,]
  design.sy <- svydesign(ids = ~1, weights = rep(N / n, n) , data = ech.sy)
  est <- svytotal(~Surfacesbois, design.sy)
  est.sy[i] <- est[1]
  IC.sy[i,] <- confint(est)
  var.sy[i] <- (SE(est))^2
}

cv.sy <- sd(est.sy) / mean(est.sy)
bias_rel.sy <- (mean(est.sy) - sum(df$Surfacesbois)) / sum(df$Surfacesbois)
verif.sy <- verifIC(IC.sy, total)
total.sy<- mean(est.sy)
var.sy <- mean(var.sy)

m.sy = round(c(cv.sy, bias_rel.sy, verif.sy, var.sy^0.5, total.sy),3)

plot_cum_moy_avec_histogram(est.sy, total, n.sim)

# Tri des données par la variable auxiliaire HApoly
order_HAp <- order(df$HApoly)
swiss_sorted <- df[order_HAp, ]
est.sytri.HAp <- numeric(n.sim)
IC.sytri.HAp <- matrix(1,n.sim, 2)
var.sytri.HAp <- numeric(n.sim)

# Simulations après tri
for (i in 1:n.sim) {
  sy <- UPsystematic(pik)
  ech.sy <- swiss_sorted[sy == 1, ]
  design.sy <- svydesign(ids = ~1, weights =rep(N / n, n), data = ech.sy)
  est <- svytotal(~Surfacesbois, design.sy)
  est.sytri.HAp[i] <- est[1]
  IC.sytri.HAp[i,] <- confint(est)
  var.sytri.HAp[i] <- (SE(est))^2
}

# CV pour le plan trié
cv.sytri.HAp <- sd(est.sytri.HAp) / mean(est.sytri.HAp)
# Biais relatif
bias_rel.HAp <- (mean(est.sytri.HAp) - sum(df$Surfacesbois)) / sum(df$Surfacesbois)
verif.sytri.HAp <- verifIC(IC.sytri.HAp, total)
total.sytri.HAp <- mean(est.sytri.HAp)
var.sytri.HAp <- mean(var.sytri.HAp)

m.sytri.HAp<- round(c(cv.sytri.HAp, bias_rel.HAp, verif.sytri.HAp, var.sytri.HAp^0.5, total.sytri.HAp),
```

```r
## Proba inégale
piki <- inclusionprobabilities(df$HApoly, n)

est.in <- numeric(n.sim)
IC.in <- matrix(1, n.sim, 2)
var.in <- numeric(n.sim)

# Simulation
for (i in 1:n.sim) {
  # Tirage systématique
  sy <- UPsystematic(piki)

  # Sélectionner les échantillons
  ech.in <- swiss_sorted[sy == 1, ]
  piki_in <- piki[sy == 1]
  design.in <- svydesign(ids = ~1, weights = 1/piki_in, data = ech.in)
  est <- svytotal(~Surfacesbois, design.in)
  est.in[i] <- est[1]
  IC.in[i,] <- confint(est)
  var.in[i] <- (SE(est))^2
}

cv.in <- sd(est.in) / mean(est.in)
# Biais relatif
bias_rel.in <- (mean(est.in) - sum(df$Surfacesbois)) / sum(df$Surfacesbois)
verif.in <- verifIC(IC.in, total)
total.in <- mean(est.in)
var.in <- mean(var.in)

m.in <- round(c(cv.in, bias_rel.in, verif.in, var.in^0.5, total.in),3)

verif3 <- data.frame(
  Metrique = c("CV", "Bias Relative","TauxIC","Ecart-type" , "Total Estimaté"),
  Modèle3 = m.in
)

# Affichage du tableau en format Markdown
knitr::kable(verif3, format = "markdown")

plot_cum_moy_avec_histogram(est.in, total, n.sim)
## par la ratio
plot_cum_moy_avec_histogram(est.sytri.HAp, total, n.sim)
est.ratio <- matrix(1,n.sim,1)
IC.ratio <- matrix(1,n.sim,2)
var.ratio <- numeric(n.sim)

for (i in 1:n.sim)
{
  #set.seed(sample(1:10^6,1))
  si.rec <- srswor(n,N)
  ech.si <- svydesign(id=~COM,
                      weights=rep(N/n,n),
                      fpc=rep(n/N,n),
```

```
                      data=df[si.rec==1,])
  R.est <-svyratio(~Surfacesbois,~HApoly,ech.si)
  est.rat<-predict(R.est, total=sum(df$HApoly))
  est.ratio[i] <- est.rat$total[1,1]
  IC.ratio[i,1] <- est.rat$total[1,1] - 0.96*est.rat$se[1,1]
  IC.ratio[i,2] <- est.rat$total[1,1] + 0.96*est.rat$se[1,1]
  var.ratio[i] <- est.rat$se[1,1]^2
}
# CV pour le plan trié
cv.ratio <- sd(est.ratio) / mean(est.ratio)
# Biais relatif
bias_rel.ratio <- (mean(est.ratio) - sum(df$Surfacesbois)) / sum(df$Surfacesbois)
verif.ratio<- verifIC(IC.ratio, total)
total.ratio <- mean(est.ratio)
var.ratio <- mean(var.ratio)

m.ratio <- round(c(cv.ratio, bias_rel.ratio, verif.ratio, var.ratio^0.5, total.ratio),3)

plot_cum_moy_avec_histogram(est.ratio, total, n.sim)

#Métrique
verif.sy <- data.frame(
  Metrique = c("CV", "Bias Relative","TauxIC","Ecart-type" , "Total Estimaté"),
  Modèle1 = m.sy,
  Modèle2 = m.sytri.HAp,
  Modèle3 = m.ratio
)
# Affichage du tableau en format Markdown
knitr::kable(verif.sy, format = "markdown")

######################
#### Ratio #########
####################

set.seed(sample(1,10^6,1))
si <- srswor(n, N)
ech <- df[si == 1,]
design <- svydesign(ids = ~1, data = ech, weights = rep(N/n, n))
ratio_est <- svyratio(~Surfacesbois, ~Surfacescult, design)
print(ratio_est)
confint(ratio_est)
val <- sum(df$Surfacesbois)/sum(df$Surfacescult)
cat("La valeur réelle est : ",val)
```