Projet 7
Programmation R (ISV51)

L3 Bio-Info (Université d'Evry - Paris-Saclay)

Jiou LEE
Marwa MOHAMED

Jeudi 17 décembre 2020

Partie 1 : Utilisation de 2 méthodes afin d'évaluer la qualité de l'intervalle

Méthode 1 : Théorème centrale limite avec approximation de p par p0

```
IC1 <- function(prob, u = qnorm(0.975),n=100){
  ICmin<- prob-u*sqrt(prob*(1-prob)/n)
  ICmax<- prob+u*sqrt(prob*(1-prob)/n)
  return(list(ICmin=ICmin, ICmax = ICmax))
}
```

Méthode 2 : Théorème central limite sans approximation de p par p0

```
IC2 <- function(prob, u = qnorm(0.975),n=100){
  ICmin<- (prob + ((u**2)/n)-(u/sqrt(n))*sqrt((u**2)/(4*n)+prob*(1-prob)))/(1+((u**2)/n))
  ICmax<- (prob + ((u**2)/n)+(u/sqrt(n))*sqrt((u**2)/(4*n)+prob*(1-prob)))/(1+((u**2)/n))
  return(list(ICmin=ICmin, ICmax = ICmax))
}
```

Voici ci-dessous la fonction permettant de comparer les intervalles de confiance obtenus par ces 2 méthodes.

```
IC <- function(prob, u = qnorm(0.975),n=100){
  ICmin1<- prob-u*sqrt(prob*(1-prob)/n)
  ICmax1<- prob+u*sqrt(prob*(1-prob)/n)
  ICmin2<- (prob + ((u**2)/(2*n))-(u/sqrt(n))*sqrt((u**2)/(4*n)+prob*(1-prob)))/(1+((u**2)/n))
  ICmax2<- (prob + ((u**2)/(2*n))+(u/sqrt(n))*sqrt((u**2)/(4*n)+prob*(1-prob)))/(1+((u**2)/n))
  return(list(ICmin1=ICmin1, ICmax1 = ICmax1,ICmin2=ICmin2, ICmax2 = ICmax2))
}
```

#Simulation 1.1 On va vérifier si les deux méthodes sont fiables à 95%, c'est-à-dire déterminer si 95% des éléments entrent dans l'intervalle de confiance généré par les deux méthodes.

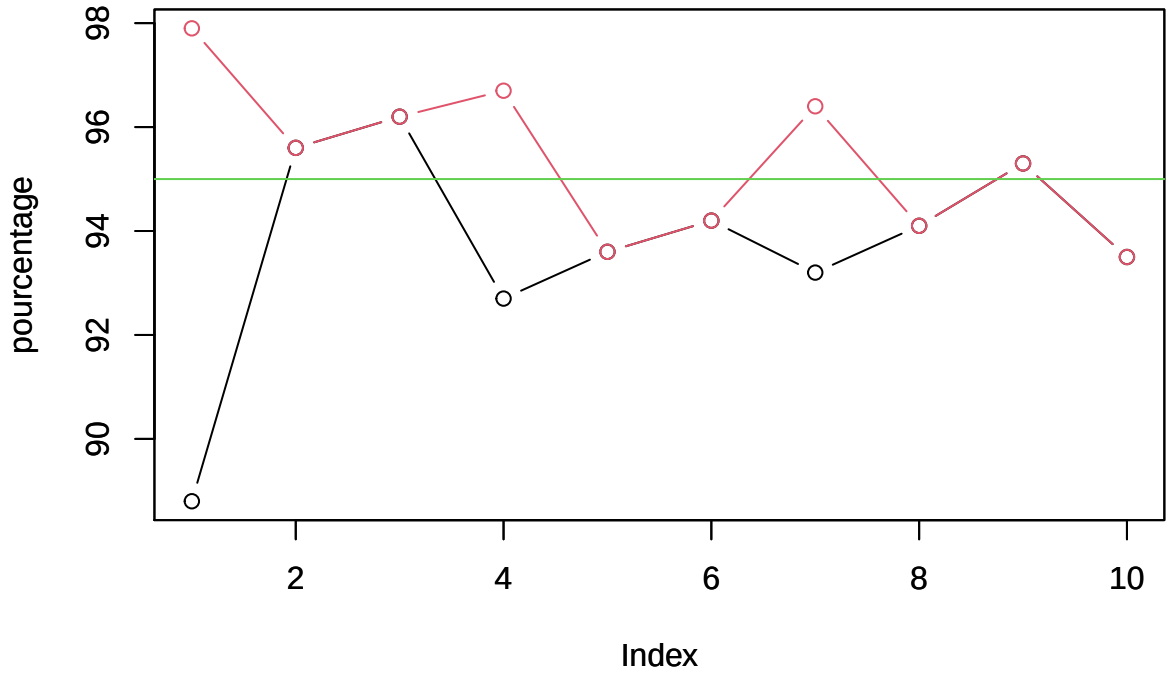
```r
simu1 <-function(N = 1000, n = 100, p = 0.5)
{
  count1<-0##Compteur pour méthode 1
  count2<-0##Compteur pour méthode 2
  u <- qnorm(0.975)
  list<-replicate(N,mean(rbinom(n,size = 1,prob = p)))##Génération de n proportions compris entre 0 et

  for(x in list)
  {
    ICtest<-IC(prob=x,u = qnorm(0.975),n=n)
    if(p>ICtest$ICmin1 && p< ICtest$ICmax1)
    {##Si le x-ième événement est dans l'intervalle généré par la méthode 1
      count1<-count1+1##On incrémente 1 au compteur de la méthode 1
    }
    if(p>ICtest$ICmin2 && p<ICtest$ICmax2)
    {##Si le x-ième événement est dans l'intervalle généré par la méthode 1
      count2<-count2+1##On incrémente 1 au compteur de la méthode 1
    }
  }##Calcul du taux d'éléments dans l'intervalle généré par les deux méthodes
  return(list(pcount1=(count1/N)*100, pcount2=(count2/N)*100))
}
##Nous simulons la précision de l'intervalle avec un échantillon grandissant afin de simuler l'infini
simu1(N = 1000, n = 10, p = 0.5)
```

```
## $pcount1
## [1] 89.7
##
## $pcount2
## [1] 98.6
```

```r
simu1(N = 1000, n = 100, p = 0.5)
```

```
## $pcount1
## [1] 94.2
##
## $pcount2
## [1] 94.2
```

```r
nb <- 10
res2 <- matrix(0,nrow = nb, ncol = 2)
for(i in 1:nb)
{
  s <- simu1(N = 1000, n = 10*i, p = 0.5)
  res2[i,] <- c(s$pcount1, s$pcount2)
}
ylim <- c(min(res2),max(res2))
plot(res2[,1], type = 'b', col = 1, ylim = ylim, ylab="pourcentage")
par(new = TRUE)
plot(res2[,2], type = 'b', col = 2, ylim = ylim, ylab="pourcentage")
abline(h=95, col = 3)
```



#Simulation 1.2

Nous comparons ensuite, grâce au code suivant, les deux méthodes pour une population de n = 1000.

```
################## Librairies
library(ggplot2)
library(parallel)
#install.packages("gridExtra") installer ce package au préalable
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.0.3
```

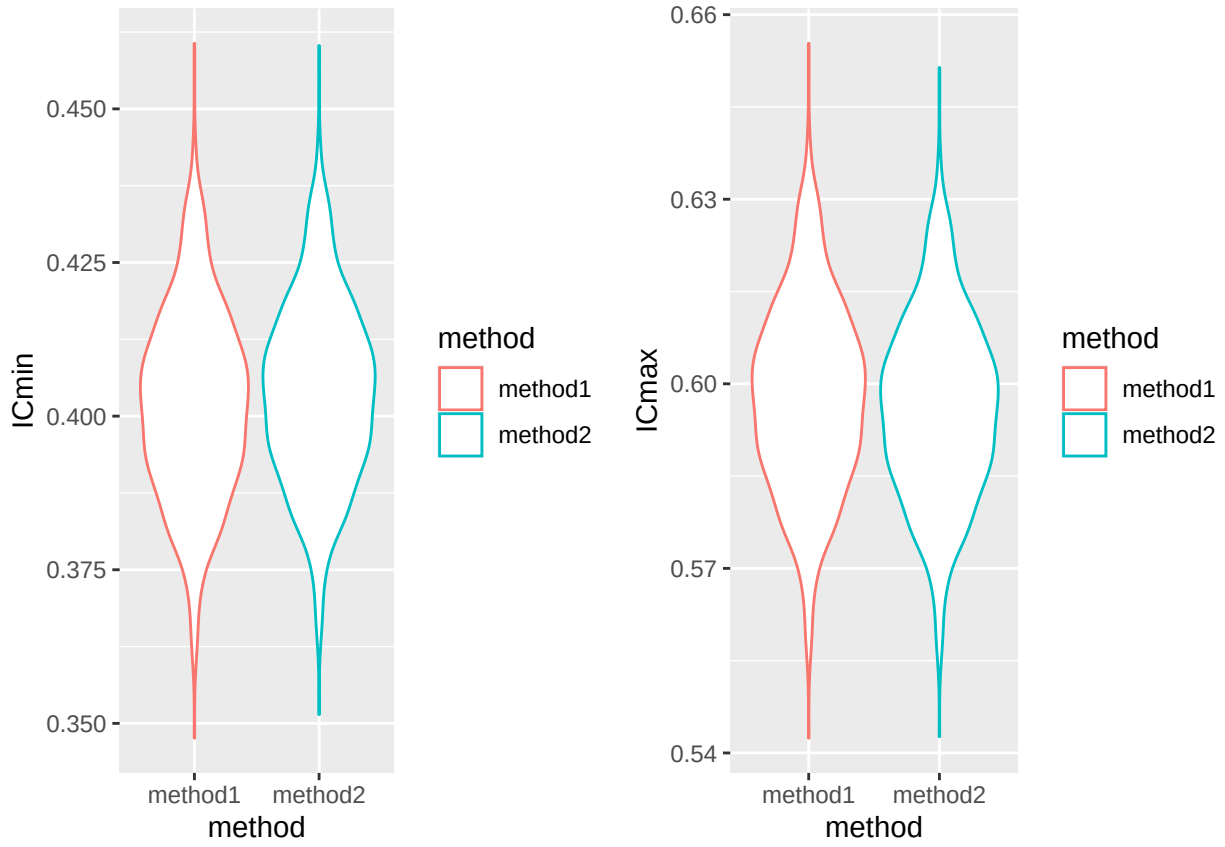
```
################## Déclaration de la fonction
simu1.2 <- function(i,n = 1000, prob)
{
  ## pour une population de taille n, chaque événement choisira au hasard 0 ou 1 avec une probabilité d
  tirage <- rbinom(n,size = 1,prob = prob)
  p <- mean(tirage) ## L'espérance d'obtenir un résultat donné
  q <- 1 - p ## L'inverse de p

  ##Calcul de l'intervalle de confiance de p avec les deux méthodes
  ICmin1.p <- IC(prob = p)$ICmin1
  ICmax1.p <- IC(prob = p)$ICmax1
  ICmin2.p <- IC(prob = p)$ICmin2
  ICmax2.p <- IC(prob = p)$ICmax2
```

```r
  ##Calcul de l'intervalle de confiance de q avec les deux méthodes
  ICmin1.q <- IC(prob = q)$ICmin1
  ICmax1.q <- IC(prob = q)$ICmax1
  ICmin2.q <- IC(prob = q)$ICmin2
  ICmax2.q <- IC(prob = q)$ICmax2

  ##Création d'un data frame
  df <- data.frame(
    Donnees = rep(c("evenement1","evenement2"),2),
    Proba = rep(c(p,q),2),
    method = rep(c("method1","method2"), each=2),
    ICmin = c(ICmin1.p,ICmin1.q,ICmin2.p,ICmin2.q),
    ICmax = c(ICmax1.p,ICmax1.q,ICmax2.p,ICmax2.q),
    stringsAsFactors = TRUE
  )
  return(df)
}

################# Application de la fonction
N<-1000

list <- mclapply(1:N, FUN = simu1.2,
                 n = 1000,
                 prob = 0.5,
                 mc.cores = 1) #si pas sous windows, remplacer 1 par nb_cores

df <- do.call(rbind, list)
head(df)
```

```
##      Donnees Proba  method     ICmin     ICmax
## 1 evenement1 0.493 method1 0.3950114 0.5909886
## 2 evenement2 0.507 method1 0.4090114 0.6049886
## 3 evenement1 0.493 method2 0.3970996 0.5894183
## 4 evenement2 0.507 method2 0.4105817 0.6029004
## 5 evenement1 0.504 method1 0.4060049 0.6019951
## 6 evenement2 0.496 method1 0.3980049 0.5939951
```

Nous visualisons en bas, les intervalles de confiance obtenus par les deux méthodes pour un événement p.

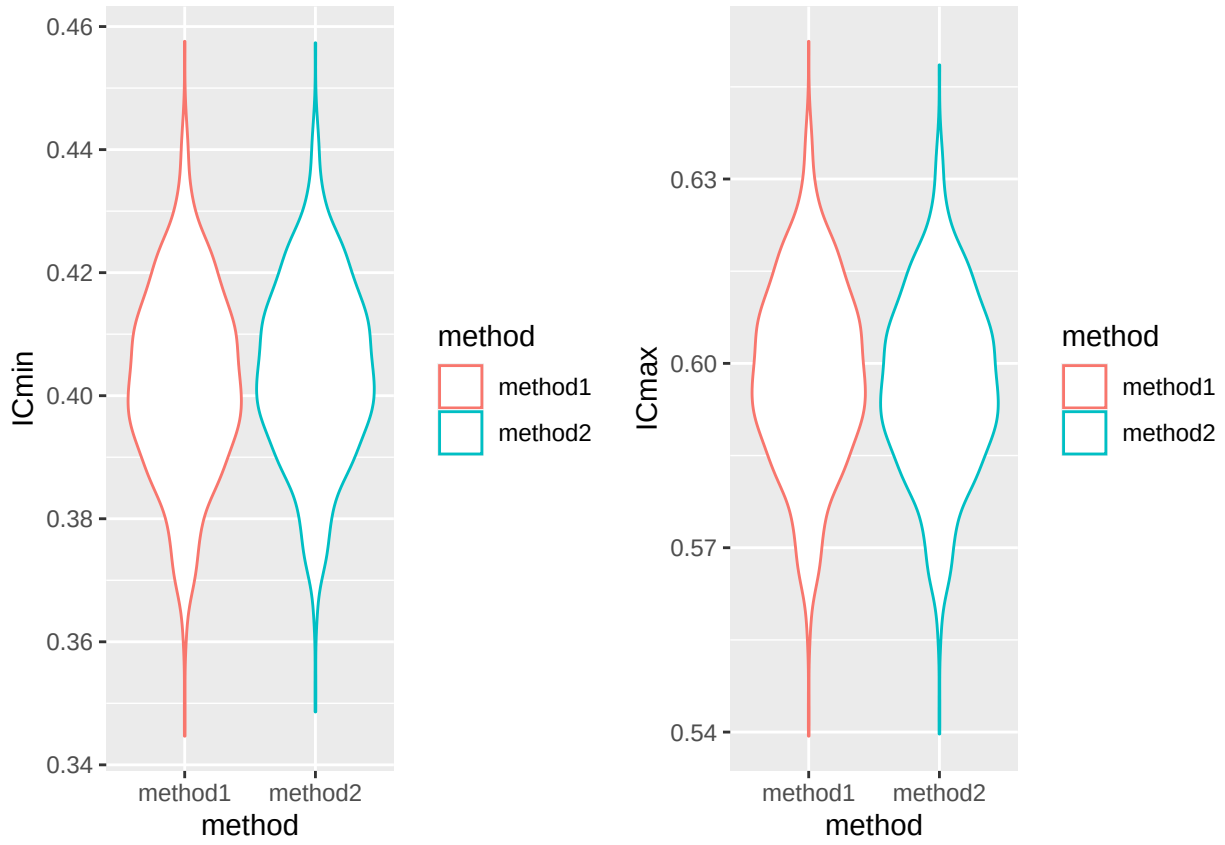
```r
ev1 <- df[df[,1] == "evenement1",1:5]

pICmin.1 <- ggplot(ev1, aes(x = method, y = ICmin, color=method)) + geom_violin()
pICmax.1 <- ggplot(ev1, aes(x = method, y = ICmax, color=method)) + geom_violin()
grid.arrange(pICmin.1, pICmax.1, ncol=2)
```



‘

Nous visualisons en bas, les intervalles de confiance obtenus par les deux méthodes pour un événement q, l'inverse de p.

```
ev2 <- df[df[,1] == "evenement2",1:5]
pICmin.2 <- ggplot(ev2, aes(x = method, y = ICmin, color=method)) + geom_violin()
pICmax.2 <- ggplot(ev2, aes(x = method, y = ICmax, color=method)) + geom_violin()
grid.arrange( pICmin.2, pICmax.2, ncol=2)
```



Partie 2 : Reproduire le tableau dans le cadre d'un sondage d'une population finie

Nous avons décidé d'appliquer les deux méthodes à un exemple concret, les élections américaines entre Trump et Biden.

```
simu2 <- function(i, n = 100, etat, prob)
{
  B <- prob[i]
  Tr <- 1 - B

  ICmin1.B<- IC(prob = B)$ICmin1
  ICmax1.B<- IC(prob = B)$ICmax1
  ICmin2.B <- IC(prob = B)$ICmin2
  ICmax2.B <- IC(prob = B)$ICmax2

  ICmin1.Tr<- IC(prob = Tr)$ICmin1
  ICmax1.Tr<- IC(prob = Tr)$ICmax1
  ICmin2.Tr <- IC(prob = Tr)$ICmin2
  ICmax2.Tr <- IC(prob = Tr)$ICmax2

  df <- data.frame(
    Etat = etat[i],
    Candidats = rep(c("Biden","Trump"),2),
    Votes = rep(c(B,Tr),2),
```

```
    method = rep(c("method1","method2"),each=2),
    ICmin = c(ICmin1.B,ICmin1.Tr,ICmin2.B,ICmin2.Tr),
    ICmax = c(ICmax1.B,ICmax1.Tr,ICmax2.B,ICmax2.Tr),
    stringsAsFactors = TRUE
  )
  return(df)
}

nEtats <- 51
table<-read.table("prBiden.txt", col.names = c("Etat","pr.Biden","tendance"))
head(table)
```

```
##   Etat pr.Biden tendance
## 1   CA    0.635    0.590
## 2   NV    0.501    0.499
## 3   OR    0.569    0.530
## 4   WA    0.584    0.549
## 5   ID    0.331    0.312
## 6   MT    0.406    0.396
```

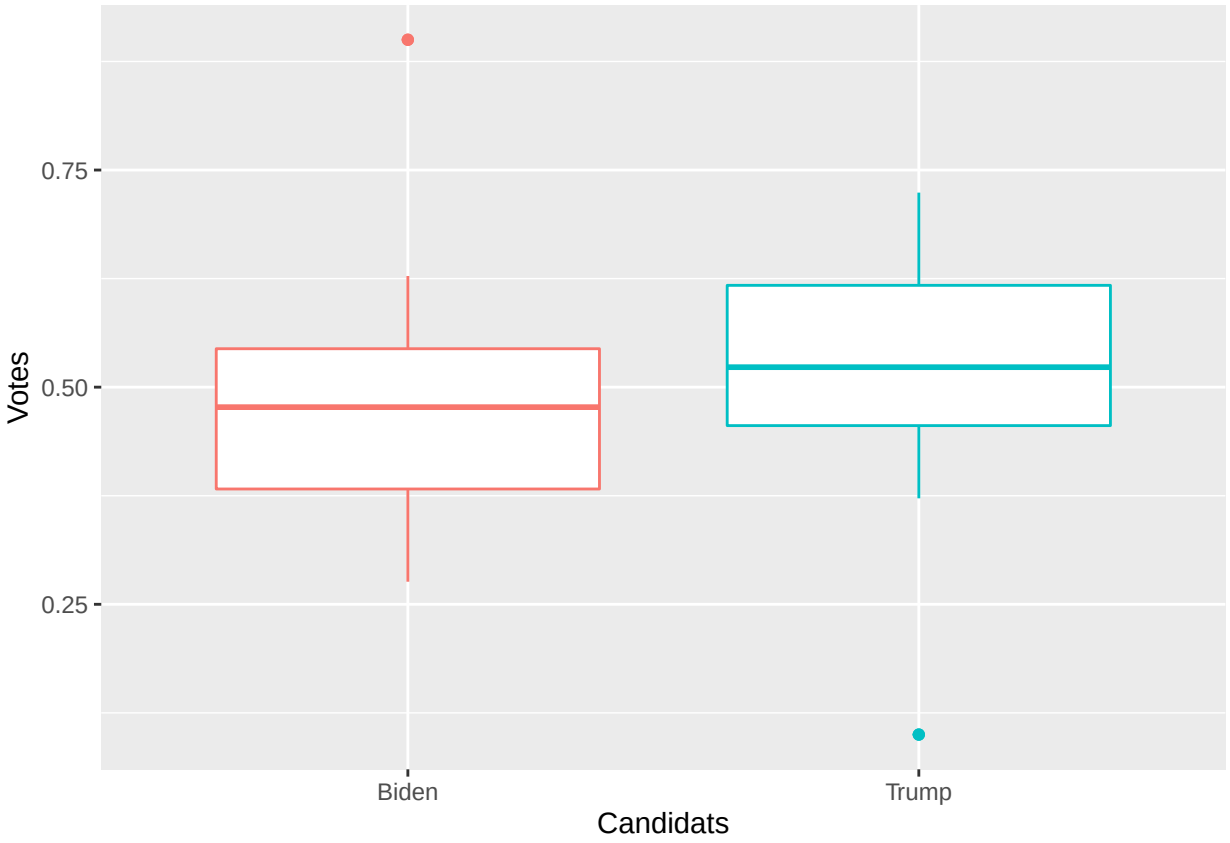
```
list_results <- mclapply(1:nEtats, FUN = simu2,
                         n = 1000,
                         etat = table$Etat,
                         prob = table$tendance,
                         mc.cores = 1) #si pas sous windows, remplacer 1 par nb_cores

df.BvsT <- do.call(rbind, list_results)
head(df.BvsT)
```

```
##   Etat Candidats Votes  method     ICmin     ICmax
## 1   CA     Biden 0.590 method1 0.4936024 0.6863976
## 2   CA     Trump 0.410 method1 0.3136024 0.5063976
## 3   CA     Biden 0.590 method2 0.4920143 0.6813269
## 4   CA     Trump 0.410 method2 0.3186731 0.5079857
## 5   NV     Biden 0.499 method1 0.4010020 0.5969980
## 6   NV     Trump 0.501 method1 0.4030020 0.5989980
```

```
real_list_results <- mclapply(1:nEtats, FUN = simu2,
                         n = 1000,
                         etat = table$Etat,
                         prob = table$pr.Biden,
                         mc.cores = 1) #si pas sous windows, remplacer 1 par nb_cores

df.BvsT.real <- do.call(rbind, real_list_results)
head(df.BvsT.real)
```

```
##   Etat Candidats Votes  method     ICmin     ICmax
## 1   CA     Biden 0.635 method1 0.5406414 0.7293586
## 2   CA     Trump 0.365 method1 0.2706414 0.4593586
## 3   CA     Biden 0.635 method2 0.5372745 0.7227373
## 4   CA     Trump 0.365 method2 0.2772627 0.4627255
## 5   NV     Biden 0.501 method1 0.4030020 0.5989980
## 6   NV     Trump 0.499 method1 0.4010020 0.5969980
```

```r
bp <- ggplot(df.BvsT, aes(x=Candidats, y=Votes, color=Candidats)) +
  geom_boxplot() +
  theme(legend.position = "none")
bp
```



##Sources votes réels : https://www.huffpost.com/elections tendance vote : https://www.270towin.com/states/ TD5 2019-2020 L2SDV-MSV31 TD5-Estimation par intervalle et théorème central limite https://fr.wikipedia.org/wiki/Intervalle_de_confiance