SAE: Sondages simples

Diallo Thierno | Imany Arango Catty

RESSOURCES: Statistiques inférentielles et Probabilités

Table des matières

Lecture fichier

```
data = read.csv("FD_NAIS_2019.csv",sep=";")
```

2.1 Génération d'échantillons

```
# cette fonction tire et retourne n élément dans le vecteur donnée, avec length(vec) > n
echantillonnage = function(vec, n){
  vec = sample(vec, size = n)
  return(vec)
}
```

2.2 Estimations ponctuelles

```
n = 1000
```

1. l'âge de la mère,

```
#tire un echantillon dans age_mere et le stocke dans la variable
age_mere = echantillonnage(data$AGEMERE, n)

# calcule moyenne
moy_age_mere = mean(age_mere)

print(paste("moyenne âge mere : ", moy_age_mere))
```

```
## [1] "moyenne âge mere :  31.08"
```

2. l'âge du père,

```
#tire un echantillon dans age_pere et le stocke dans la variable
age_pere = echantillonnage(data$AGEPERE, n)

# calcule moyenne
moy_age_pere = mean(age_pere)

print(paste("moyenne âge pere : ", moy_age_pere))
```

```
## [1] "moyenne âge pere :  33.97"
```

3. Donner une estimation ponctuelle non biaisée de la variance de l'âge de de la mère.

```
# calcule variance âge mere
var_age_mere = var(age_mere)

print(paste("variance non biaisée âge mere : ", var_age_mere))
```

```
## [1] "variance non biaisée âge mere :  28.8584584584585"
```

2.3 Estimations par intervalles de confiance

1. Créer une fonction fournissant un intervalle de confiance pour chacune des caractéristiques du paragraphe précédent (2.2).

```r
#calcule un intervalle de confiance centré en X et retourne
#un vecteur (born inf, born sup)
interv_conf = function(ech, level){
  # calcule moyenne, variance et nb_echantillon
  moy = mean(ech)
  vari = var(ech)
  n = length(ech)

  # calcul du quantile d'ordre (1 + level) / 2 : la variance est
  #inconnue donc, on utilise la loi de Student
  level = (1 + level) / 2
  q = qt(level, n - 1)

  #calcul de la borne inf et sup de l'intervalle
  born_inf =  moy - (sqrt(vari / n)) * q
  born_sup =  moy + (sqrt(vari / n)) * q

  return(c(born_inf, born_sup))

}
```

Nous avons utiliser la loi de student pour que notre fonction puisse gérer toutes les tailles d'echantillons, c'est a dire, quand `n` est petit, $\overline{X}_n$ suit une loi de student de $n-1$ degré de liberté. lorque `n` est grand, $\overline{X}_n$ suit une loi normale, or la loi de student tend vers une loi normal quand `n` grandit, donc la fonction peut gérer tous les cas d'échantillons.

2. Comment la longueur de l'intervalle de confiance varie-t-elle quand on change le niveau de confiance ? Même question quand la taille de l'échantillon augmente.

```r
# valeur de c pour p(t<Z<t) = c
conf = seq(0.1, 0.9, length = 9)

#calcule de l'intervalle de confiance pour les valeurs de C et
#affichage de leur évolution
for(i in conf){
  confiance = interv_conf(age_mere, i)
  print(paste("étendue pour intervalle de confiance de niveau ",
              i," : ", confiance[2] - confiance[1]))
}
```

```
## [1] "étendue pour intervalle de confiance de niveau  0.1  :  0.0427049944086022"
## [1] "étendue pour intervalle de confiance de niveau  0.2  :  0.0860990111665316"
## [1] "étendue pour intervalle de confiance de niveau  0.3  :  0.130952400133758"
## [1] "étendue pour intervalle de confiance de niveau  0.4  :  0.178224846845936"
## [1] "étendue pour intervalle de confiance de niveau  0.5  :  0.229245100524309"
## [1] "étendue pour intervalle de confiance de niveau  0.6  :  0.286067779764728"
## [1] "étendue pour intervalle de confiance de niveau  0.7  :  0.352316873184492"
## [1] "étendue pour intervalle de confiance de niveau  0.8  :  0.435702371385084"
## [1] "étendue pour intervalle de confiance de niveau  0.9  :  0.559366859958985"
```

Plus le niveau de confiance grandit, logiquement plus l'intervalle de confiance grandit également afin d'avoir plus de chance que la valeur se trouve dans cet intervalle. On le remarque avec le résultat ci-dessus l'intervalle de confiance du niveau 90 est plus étendue que celui de 10 %.

```r
#creation de longeur d'echantillon
nb = c(10, 100, 1000, 10000)
```

```r
for(n in nb){
  confiance = interv_conf(echantillonnage(data$AGEMERE, n), 0.9)
  print(paste("intervale de confiance de 0.9 pour ",
              n," individu : ", confiance[2] - confiance[1]))
}
```

```
## [1] "intervale de confiance de 0.9 pour  10  individu :  5.04466994963865"
## [1] "intervale de confiance de 0.9 pour  100  individu :  1.82220503287245"
## [1] "intervale de confiance de 0.9 pour  1000  individu :  0.562165839729239"
## [1] "intervale de confiance de 0.9 pour  10000  individu :  0.178008915766355"
```

Lorsque nous augmentons la longeur de l'échantillon, l'étendue du niveau de confiance diminue, donc l'intervalle de confiance devient de plus en plus précis. Cependant plus les valeurs de n seront grandes plus la variation de l'intervalle de confiance sera lente.

3. Créer une nouvelle fonction qui renvoit un intervalle de confiance de la forme $[a; +\infty[$ et un autre de la forme $]-\infty; b]$ pour l'âge de la mère.

```r
interv_conf_droite = function(ech, level){
  # P( -t < X ) = level
  moy = mean(ech)
  vari = var(ech)
  n = length(ech)

  level = (1 - level)
  q = qt(level, n - 1)

  born_inf =  - Inf
  born_sup =  moy + (sqrt(vari / n)) * q

  return(c(born_inf, born_sup))

}


interv_conf_gauche <- function(ech,level){
  # P( X < t ) = level
  moy = mean(ech)
  vari = var(ech)
  n = length(ech)

  q = qt(level, n - 1)

  born_inf = moy - q * (sqrt(vari/n))

  return(c(born_inf,+Inf))

}

interv_conf_droite(age_mere, 0.05)
```

```
## [1]     -Inf 31.35968
```

```r
interv_conf_gauche(age_mere, 0.95)
```

```
## [1] 30.80032      Inf
```

2.4 Etude de la qualité des sondages

Protocole : (1) générer un grand nombre d'échantillons de même taille, (2) déterminer les intervalles de confiance associés aux échantillons tirés (α est fixe), (3) calculer la proportion des intervalles de confiance obtenus qui contiennent la vraie valeur (celle de la population).

— De quelle valeur théorique devrait s'approcher la proportion des intervalles de confiance contenant la vraie moyenne ?

La valeur théorique de la proportion des intervalles de confiance contenant la vraie moyenne devrait se rapprocher du niveau de confiance `level` (ici ce qui correspond à 95%)}

— Créer le code permettant de réaliser le protocole ci-dessus.

```r
# n     : taille de l'echantillon
# len   : nombre d'echantillon
# vec   : donnée sur lesquelles on test
# level : niveau de confiance

prop_in_ech = function(vec = data$AGEMERE, n = 1000, len = 1000, level = 0.95){
  mean_vec = mean(vec)

  in_ech = numeric()

  for(i in 1:len){
    ech = echantillonnage(vec, n)
    borne = interv_conf(ech, level)
    if(borne[1] < mean_vec & borne[2] > mean_vec){
      in_ech[i] = 1
    } else {
      in_ech[i] = 0
    }

  }
  return(mean(in_ech))
}
```

— Faire varier la caractéristique à étudier, la taille de l'échantillon, le niveau de confiance ... et commentez les résultats.

```r
val = c(2, 10, 50, 100, 500, 1000, 2000)

print("test de variation de n : len = 500, level = 0.95, vec = data$AGEMERE")
```

```
## [1] "test de variation de n : len = 500, level = 0.95, vec = data$AGEMERE"
```

```r
for(i in val){
  print(paste("pour n = ", i, " : ", "proportion = ",
              prop_in_ech(n = i, len = 500)))
}
```

```
## [1] "pour n =  2  :  proportion =  0.936"
## [1] "pour n =  10  :  proportion =  0.94"
## [1] "pour n =  50  :  proportion =  0.968"
## [1] "pour n =  100  :  proportion =  0.95"
## [1] "pour n =  500  :  proportion =  0.954"
```

```
## [1] "pour n =  1000  :  proportion =  0.932"
## [1] "pour n =  2000  :  proportion =  0.952"
```

Lorsque nous faisont changer la taille de notre échantillon la proportion des intervalles de confiance contenant notre vraie moyenne se rapproche du niveau de confiance `level`, c'est a dire plus n est grand plus on est précis.

```
val = c(2, 10, 50, 100, 500, 1000, 2000)

print("test de variation de len : n = 500, level = 0.95, vec = data$AGEMERE")
```

```
## [1] "test de variation de len : n = 500, level = 0.95, vec = data$AGEMERE"
```

```
for(i in val){
  print(paste("pour len = ", i, " : ", "proportion = ",
              prop_in_ech(len = i, n = 500)))
}
```

```
## [1] "pour len =  2  :  proportion =  1"
## [1] "pour len =  10  :  proportion =  0.9"
## [1] "pour len =  50  :  proportion =  0.98"
## [1] "pour len =  100  :  proportion =  0.94"
## [1] "pour len =  500  :  proportion =  0.952"
## [1] "pour len =  1000  :  proportion =  0.953"
## [1] "pour len =  2000  :  proportion =  0.944"
```

Lorsque nous faisons varier le nombre d'échantillons, la proportion d'intervalle contenant notre vraie moyenne varie, quand le nombre d'échantillons augmente on devient encore plus précis (on se rapproche de plus en plus de "level") et pour cela nous n'avons pas besoin de prendre un grand nombre d'échantillons car on voit qu'à partir de 100, la valeur est déjà suffisamment précise et qu'en augmentant encore cela ne nous rendra pas plus précis.

```
val = c(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95)

print("test de variation de level : len = 500, n = 500, vec = data$AGEMERE")
```

```
## [1] "test de variation de level : len = 500, n = 500, vec = data$AGEMERE"
```

```
for(i in val){
  print(paste("pour level = ", i, " : ", "proportion = ",
              prop_in_ech(level = i, len = 1000, n = 1000)))
}
```

```
## [1] "pour level =  0.1  :  proportion =  0.111"
## [1] "pour level =  0.2  :  proportion =  0.184"
## [1] "pour level =  0.3  :  proportion =  0.288"
## [1] "pour level =  0.4  :  proportion =  0.412"
## [1] "pour level =  0.5  :  proportion =  0.5"
## [1] "pour level =  0.6  :  proportion =  0.588"
## [1] "pour level =  0.7  :  proportion =  0.694"
## [1] "pour level =  0.8  :  proportion =  0.787"
## [1] "pour level =  0.9  :  proportion =  0.889"
## [1] "pour level =  0.95  :  proportion =  0.956"
```

Ici on remarque que la proportion d'intervalle de confiance contenant la vraie moyenne est peu près proportionelle au niveau de confiance `level` pour `n` et `len` suffisamment grand.

2.5 Naît-il plus de filles ou de garçons ?

Servez-vous de ce théorème pour déterminer un intervalle de confiance de la probabilité qu'un nouveau né soit de sexe masculin. Faîtes varier le niveau de confiance : $\alpha = 0.95, 0.99, 0.999...$ Quelles sont vos conclusions ?

```
data_sexe = data$SEXE == 1


mean_data_sexe = mean(data_sexe)
var_data_sex = var(data_sexe)

val = c(0.95, 0.99, 0.999)

for(i in val){
  level = i
  q = qnorm((1 + level) / 2)
  born_inf = mean_data_sexe - q * sqrt(var_data_sex / length((data_sexe)))
  born_sup = mean_data_sexe + q * sqrt(var_data_sex / length((data_sexe)))

  b = c(born_inf, born_sup)
  print(paste("interval de confance de niveau ", i, " : [", b[1], ", ", b[2], "]"))
}
```

```
## [1] "interval de confance de niveau  0.95  : [ 0.509949928573241 ,  0.512207460165289 ]"
## [1] "interval de confance de niveau  0.99  : [ 0.509595244654282 ,  0.512562144084248 ]"
## [1] "interval de confance de niveau  0.999  : [ 0.509183642187333 ,  0.512973746551197 ]"
```

D'après les intervalles de confiances que nous avons eu nous pouvons en conclure qu'il y a une lègere différence entre la proportion d'hommes et de femmes en faveur des hommes. Environ 51% d'hommes pour logiquement, environ 49% de femmes . Cela rejoint la problématique de la question qui émettait que les bébés humains sont plus souvent masculin, en revanche nous n'avons aucune preuve qu'il s'agisse d'un facteur biologique il est beaucoup plus probable qu'il s'agisse d'un facteur socio-économique.