

Auto-évaluation – Stratégies d'échantillonnage et intervalles de confiance

Solutions

1. a) Parmi les stratégies d'échantillonnage suivantes, laquelle est la plus appropriée pour comparer la densité de pins gris dans cinq différents types d'habitats? *Réponse* : d) **échantillonnage stratifié aléatoire**

b) Dès qu'un échantillon comporte moins de 50 observations, il faut utiliser la distribution normale.

Réponse : **faux** – il faut plutôt utiliser la distribution du t de Student.

c) Nommez un problème qui peut survenir dans un échantillon.

Réponse : La non-représentativité de l'échantillon, les erreurs de saisies des données ou de mesure, et les erreurs liées à la probabilité de détection sont tous des problèmes potentiels dans un échantillon.

d) La distribution cunéiforme permet de sélectionner aléatoirement des unités dans une population.

Réponse : **Faux** – c'est plutôt la distribution uniforme qu'on utilise.

e) Donnez un désavantage d'utiliser une stratégie d'échantillonnage systématique.

Réponse : L'échantillonnage systématique ne donne pas un échantillon représentatif en présence de patrons périodiques et les inférences qu'on peut en tirer sont très limitées en présence d'une seule grille.

f. Quelle stratégie d'échantillonnage serait la plus appropriée pour étudier l'occurrence d'un petit champignon très rare et difficile à voir sur la litière forestière?

Réponse : La méthode d'analyse d'occupation de sites serait la plus appropriée, mais, à la limite, l'échantillonnage adaptatif pourrait aussi être approprié.

2. a) Importez le jeu de données `croissance.csv`, qui présente la croissance en kg de bovins auxquels on a donné l'une des trois types de moulées.

Réponse :

```
> ##importer jeu de données
> gain <- read.table("croissance.csv", header = TRUE)
> ##on regarde une partie du jeu de données et sa structure
> head(gain)

  diete  gain
1  ble 17.371
2  ble 16.815
3  ble 18.082
4  ble 15.782
5  ble 17.707
6  ble 18.227

> str(gain)

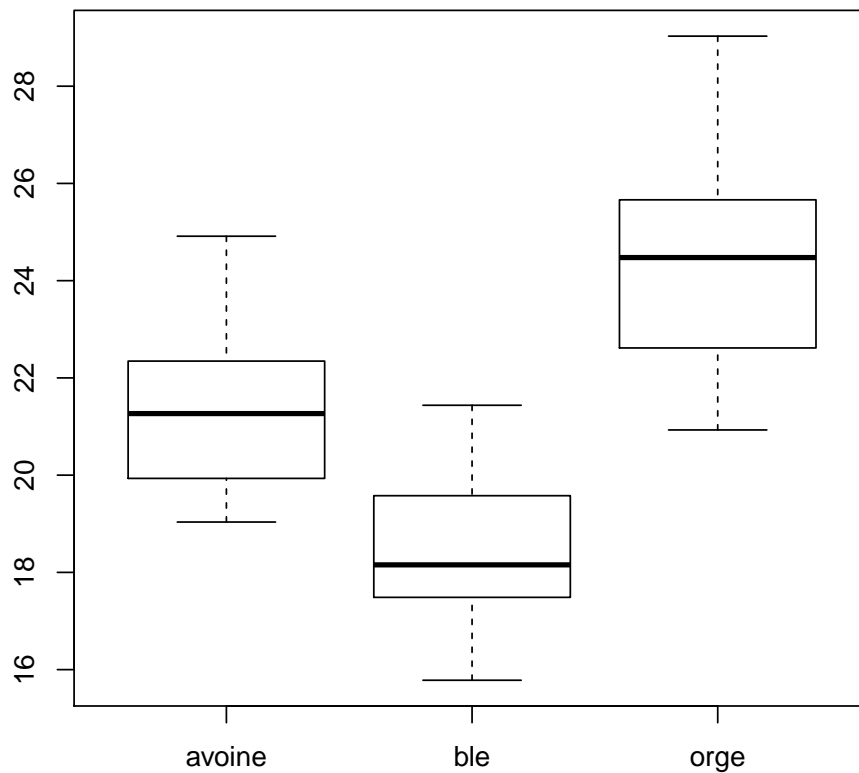
'data.frame':      48 obs. of  2 variables:
 $ diete: Factor w/ 3 levels "avoine","ble",...: 2 2 2 2 2 2 2 2 2 2 ...
 $ gain : num  17.4 16.8 18.1 15.8 17.7 ...
```

b) Calculez l'intervalle de confiance à 95 % autour de la moyenne de chacun des trois groupes définis par le type de moulée.

Vous trouverez de plus amples détails sur le script suivant dans le forum du cours à l'entrée « Explications sur le script du calcul de l'intervalle de confiance ». Vous trouverez facilement cette entrée en saisissant quelques mots du titre dans la fenêtre de recherche du forum (en haut de la page).

Réponse :

```
> ##on visualise les données  
> boxplot(gain$gain ~ gain$diete)
```



```
> ##on crée un sous-jeu de données pour chaque diète
> avoine <- gain[gain$diete == "avoine", ]
> ble <- gain[gain$diete == "ble", ]
> orge <- gain[gain$diete == "orge", ]
> ##avoine - IC à 95%
> n.avoine <- nrow(avoine)
> moy.avoine <- mean(avoine$gain)
> sd.avoine <- sd(avoine$gain)
> SE.avoine <- sd.avoine/sqrt(n.avoine)
> IC.inf.avoine <- moy.avoine + qt(p = 0.025, df = n.avoine - 1) * SE.avoine
> IC.sup.avoine <- moy.avoine - qt(p = 0.025, df = n.avoine - 1) * SE.avoine
> ##ble - IC à 95%
> n.ble <- nrow(ble)
> moy.ble <- mean(ble$gain)
> sd.ble <- sd(ble$gain)
> SE.ble <- sd.ble/sqrt(n.ble)
> IC.inf.ble <- moy.ble + qt(p = 0.025, df = n.ble - 1) * SE.ble
> IC.sup.ble <- moy.ble - qt(p = 0.025, df = n.ble - 1) * SE.ble
> ##avoine - IC à 95%
> n.orge <- nrow(orge)
> moy.orge <- mean(orge$gain)
> sd.orge <- sd(orge$gain)
> SE.orge <- sd.orge/sqrt(n.orge)
> IC.inf.orge <- moy.orge + qt(p = 0.025, df = n.orge - 1) * SE.orge
> IC.sup.orge <- moy.orge - qt(p = 0.025, df = n.orge - 1) * SE.orge
> ##assembler le tout dans un petit tableau
> out <- data.frame(Groupe = c("avoine", "ble", "orge"),
```

```
Moyenne = c(moy.avoine, moy.ble, moy.orge),  
IC.inf = c(IC.inf.avoine, IC.inf.ble, IC.inf.orge),  
IC.sup = c(IC.sup.avoine, IC.sup.ble, IC.sup.orge))  
  
> out  
  
  Groupe Moyenne IC.inf IC.sup  
1  avoine  21.329 20.441 22.216  
2    ble   18.431 17.579 19.283  
3   orge   24.422 23.213 25.631
```

c) Présentez graphiquement les moyennes avec des barres d'erreur à l'aide des fonctions `plot()`, `axis()`, `points()` et `arrows`. À noter que le plus simple est de créer le graphique par étape :

1. Utiliser `plot()` pour créer un graphique vide avec les étiquettes des axes et les bonnes limites, mais en supprimant l'axe des x's avec l'argument `axt = "n"`. La raison en est simple : à chaque fois que l'on spécifie un graphique d'une variable numérique en fonction d'une variable catégorique (i.e., comme `diete`), R donne un diagramme de boîtes et moustaches – ce n'est pas ce qu'on veut ici.
2. Ajouter l'axe des x's à l'aide de `axis()`.
3. Ajouter les points avec `points()`.
4. Ajouter les barres d'erreurs avec `arrows()`. Cette dernière fonction nécessite un point de départ (`x0`, `y0`) et un point d'arrivée (`x1`, `y1`). Bien qu'elle puisse créer des flèches, on peut obtenir des barres d'erreurs en utilisant `angle = 90`. L'argument `code = 3` spécifie que l'on veut une flèche au point d'arrivée et au point de départ.

Vous trouverez de plus amples détails sur le script suivant dans le forum du cours à l'entrée « Explications sur le script du graphique de la moyenne et des barres d'erreur ».

Réponse :

```
> ##présenter les résultats dans un graphique
> ##créer un graphique vide
> ##voir ?par pour les paramètres graphiques
> plot(y = 0, x = 0, xlab = "Type de moulée",
      ylab = "Moyenne (kg)", main = "Moyennes ± IC à 95 %",
      ylim = c(min(out$IC.inf), max(out$IC.sup)),
      ##détermine les limites sur le graphique
      xlim = c(0, 4),
      xaxt = "n") #on n'affiche pas l'axe des x's tout de suite
> ##on ajoute l'axe des x's
> axis(side = 1, at = c(1, 2, 3), labels = c("avoine", "ble", "orge"))
> ##ajouter les points
> points(y = out$Moyenne, x = c(1, 2, 3))
> ##ajouter les barres d'erreurs
> arrows(x0 = c(1, 2, 3), x1 = c(1, 2, 3), y0 = out$IC.inf,
      y1 = out$IC.sup, length = 0.05, code = 3, angle = 90)
> ##les barres d'erreurs ont un point de départ x0, y0 et
> ##d'arrivée x1, y1
> ##length donne la longueur des flèches
> ##angle = 90 demande des flèches à 90°
> ##code = 3 demande une flèche au point de départ et d'arrivée
```

