TP3: Tests d'hypothèses

Hubert Charles

28 octobre 2015

Table des matières

1 Comparaison de distributions

Dans cet exemple, on utilisera le fichier "arbre.txt" disponible sur Moodle. Il s'agit de mesures de hauteurs d'arbres issues de deux terrains différents (T1 et T2).

1. Représentez les deux distributions côte à côte sur une même figure d'abord sous forme d'histogrammes, puis ensuite sous forme de boxplots sur une autre figure
2. Comparez les distributions par un test de Komolgorov-Smirnov (*ks.test*)
3. Comparez les variances avec la fonction *var.test*
4. Comparez les moyennes avec un test de t (*t.test*)

Discutez les résultats des tests.

2 Construire son test soi-même

On utilisera encore le fichier arbre.txt. On décide arbitrairement de s'intéresser à la somme des hauteurs des arbres sur un terrain pour la comparer à l'autre.

Calculer la somme de T1 (ST1), est-elle exceptionnelle ? Est-elle significativement plus grande que celle de T2 ?

Pour tester cela, "prendre tous les arbres et les mettre dans une urne" (c'est facile sous R !), puis réaliser 1000 échantillonages de T1 et calculer pour chacun la somme des T1 simulés. Construire l'histogramme des ST1 simulées et attribuer une p-value à la ST1 observée dans le cas d'un test unilatéral, puis d'un test bilatéral. Vous discuterez de ces deux situations et de ce qui les différencie.

De quelles hypothèses avez vous eu besoin pour construire votre test et de laquelle vous êtes vous dispensée ?

3 Puissance d'un test statistique

Dans cette partie, nous allons travailler sur des échantillons simulés (s1 et s2) dans des distributions normales de moyennes $\mu_1 = 0$ et $\mu_2 = 2$, de même taille ($n = 30$) et de même écart type $\sigma = 1$.

Tracer les densités de probabilités théoriques de ces deux lois sur un même graphe (l'une en trait plein l'autre en pointillé) et représenter les seuils gauche et droite à 0.025 en bleu et en vert sur chaque courbe avec des lignes verticales.

En utilisant ce même graphique jouer sur μ_2 pour estimer l'écart de moyenne δ minimum que l'on sera en mesure de détecter avec un risque $\alpha = 0.05$ et une puissance $(1 - \beta) = 0.95$.

Utiliser encore le même graphe pour déterminer la puissance du test $(1 - \beta)$ permettant de ne pas accepter HO alors que H1 "une différence de moyenne $\delta = 2$ existe" est vraie ?

Revenons à nos échantillons issus de ces deux lois, dans quelle mesure intervient la taille de l'échantillon ? Refaire le graphique des histogrammes supperposés avec une taille de 10, 100 et 1000. Sur quelle propriété du test, la taille des échantillons influe-t-elle ?

4 Illustration du théorème central limite

Dans cette exemple, vous allez simuler une moyenne (c'est-à-dire une somme) de variables aléatoires distribuées selon une loi uniforme. On observera alors que même si les variables aléatoires ne sont pas normales, leur moyenne devient normale dès que l'on observe plus de 10 tirages. Vous comencerez par effectuer un premier tirage dans une loi uniforme continue entre O et 1. Vous représenterez la distribution de votre échantillon sous forme d'un histogramme sur lequel vous supperposerez la courbe de densité théorique d'une loi normale.

Partager ensuite votre fenêtre graphique en quatre parties et repésenter les histogrammes des moyennes (Faire 10 000 simulations au moins) pour des échantillons de taille 1, 2, 10 et 100 et superposer les courbes théoriques.

5 Comparaison de variances, le test de Hartley

Le test de Hartley permet la comparaison de k variances observées issues de k échantillons de même taille n (*cf.* cours). Dans la pratique la réalisation du test est très simple :

1. on calcule les k variances empiriques
2. on sélectionne les deux variances extrêmes
3. on calcule le rapport H de la plus grande sur la plus petite
4. on lit sur une table (pour un risque α donné) la valeur de H acceptable sous l'hypothèse d'égalité des variances
5. on conclut à l'acceptation de H0 si la valeur observée est inférieure à la valeur de la table et on rejette l'hypothèse si la valeur est supérieure

On vous propose ici de créer une fonction "hartley($\nu, k, quant$) qui donne par simulation la statistique de H en fonction du degré de liberté de la variance théorique commune (ν), du nombre d'échantillons (k) et de la valeur du risque $\alpha = (1 - quant)$.

A vous de jouer... Une fois la fonction créée, vous reproduirez une table des valeur de H ($\alpha = 0.09$, $\nu = 2$ à 10 et $k = 2$ à 5) que vous pourrez comparer à la table du cours polycopié, puis vous tracerez une abaque des valeurs de H pour un risque donné (par exemple $\alpha = 0.05$), pour 2 à 6 échantillons de tailles variant entre $n = 3$ et $n = 20$ avec des couleurs et une légende.

6 Tests multiples et contrôle du risque α

Nous allons simuler les notes (20 examens) d'une promotion de 30 élèves avec l'idée de comparer ensuite les élèves entre eux. Lorsqu'on réalise un grand nombre de tests sur un même jeu de données, nous verrons qu'il est nécessaire de controler le risque α pour ne pas conclure à tord trop souvent (seulement par "malchance"). On vous propose le script ci-dessous. Le test d'ANOVA (cf. cours) permet de tester s'il existe globalement des différences de moyennes entre les élèves (H0 : tous les élèves ont la même moyenne) :

1. Expliquez ce qui est fait ligne par ligne,
2. selon, la simulation que doit donner le test d'ANOVA ?

```
> notes.m=matrix(nr=30,nc=20)
> for (i in 1:20){notes.m[,i]=round(rnorm(30,10,abs(rnorm(1,mean=1,sd=0.3))), digit=2)}
> colnames(notes.m)= c("E1","E2", "E3", "E4", "E5","E6","E7","E8","E9","E10","E11",
+                     "E12","E13","E14","E15","E16","E17","E18","E19","E20")
> notes=as.vector(t(notes.m))
> etu=sort(rep(as.factor(1:30),20))
> plot(notes~etu)
> anova(lm(notes~etu))
```

On souhaite comparer tous les étudiants entre eux, soit réaliser $30*29/2 = 435$ tests différents. Pour cela, vous ferez un script R avec 2 boucles imbriquées pour comparer au moyen de la fonction *t.test* tous les élèves entre eux. Vous récupèrerez dans un vecteur les 435 pvalues correspondantes et vous compterez le nombre de tests significatifs au risque individuel $\alpha = 5\%$ et discuterez du résultat (faites tourner plusieurs fois votre simulation pour bien comprendre son comportement). Combien de fois avez-vous conclu à tord dans votre simulation ?

Afin d'aller plus vite et de vérifier vos calculs, on vous propose de faire la même chose en utilisant la fonction *pairewise.t.test* de la librairie *multcomp*, selon le script ci-dessous. Décrivez ligne à ligne ce qui est fait et discuter des résultats.

```
> library(multcomp)
> test.none <- pairwise.t.test(notes, etu,p.adj="none")
> summary(as.vector(test.none$p.value<0.05))
```

Le Family-wise error rate (FWER), c'est-à-dire le risque cumulé de conclure au moins une fois à tord est de $(1 - (1 - \alpha)^c)$ avec c le nombre de comparaisons réalisées. Calculer le FWER dans votre cas que peut-on en dire ?

Supposons maintenant que l'étudiant 5 à une moyenne de 13. Vous ferez tourner plusieurs fois la simulation ci-dessous et vous commenterez les résultats. Il existe différentes façons de controler le FWER selon si les tests sont réalisés séquentiellement ou simultanément, selon s'ils sont dépendants ou indépendants (cf. cours). Il est également possible de contrôler, non pas le risque α, mais le nombre total de faux positif de notre série de test : c'est le FDR (False Discovery Rate). La procédure de contrôle de FWER la plus simple et celle énnoncée par Bonféreni. A l'aide du script ci-dessous et d'informations que vous irez chercher sur internet, essayer de comprendre ce qui est fait pour réaliser ce contrôle de risque. C'est parfait non, on ne conclue plus du tout à tord ! Vous commenterez cette affirmation.

```
> notes.m=matrix(nr=30,nc=20)
> for (i in 1:20){notes.m[,i]=round(rnorm(30,10,abs(rnorm(1,mean=1,sd=0.3))), digit=2)}
> colnames(notes.m)= c("E1","E2", "E3", "E4", "E5","E6","E7","E8","E9","E10","E11",
+                      "E12","E13","E14","E15","E16","E17","E18","E19","E20")
> notes.m[5,]=notes.m[5,]+1
> notes=as.vector(t(notes.m))
> etu=sort(rep(as.factor(1:30),20))
> plot(notes~etu)
> anova(lm(notes~etu))
> test.none <- pairwise.t.test(notes, etu,p.adj="none")
> summary(as.vector(test.none$p.value<0.05))
> test.bonf <- pairwise.t.test(notes, etu,p.adj="bonferroni")
> summary(as.vector(test.bonf$p.value<0.05))
> ((test.bonf$p.value/test.none$p.value)[test.bonf$p.value!=1])
```

Voici enfin, une dernière simulation. Vous expliquerez ce qui est fait et commenterez les résultats obtenus (n'hésitez pas à faire tourner plusieurs fois la simulation pour comprendre son comportement).

```
> notes.m=matrix(nr=30,nc=20)
> for (i in 1:20){notes.m[,i]=round(rnorm(30,10,abs(rnorm(1,mean=1,sd=0.3))), digit=2)}
> colnames(notes.m)= c("E1","E2", "E3", "E4", "E5","E6","E7","E8","E9","E10","E11",
+                      "E12","E13","E14","E15","E16","E17","E18","E19","E20")
> notes.m[1:30,]=notes.m[1:30,]+seq(-1, +1, le=30)
> notes.m=notes.m[sample(1:30,30, rep=F),]
> notes=as.vector(t(notes.m))
> etu=sort(rep(as.factor(1:30),20))
> plot(notes~etu)
> anova(lm(notes~etu))
> test.none <- pairwise.t.test(notes, etu,p.adj="none")
> summary(as.vector(test.none$p.value<0.05))
> test.bonf <- pairwise.t.test(notes, etu,p.adj="bonferroni")
> summary(as.vector(test.bonf$p.value<0.05))
> test.holm <- pairwise.t.test(notes, etu,p.adj="holm")
> summary(as.vector(test.holm$p.value<0.05))
> test.BH <- pairwise.t.test(notes, etu,p.adj="BH")
> summary(as.vector(test.BH$p.value<0.05))
```