

TD1 : Révisions

Normalité d'une variable

Test d'un paramètre

Test de comparaison d'échantillons

BIO5XX - BIOSTATISTIQUE L3

21 septembre 2016

Objectifs de la séance

À partir des jeux de données proposés et de la problématique biologique soulevée, il s'agit de choisir le bon test statistique et de s'initier avec le logiciel *R*

1 Les données peupliers

Rappels sur les données :

Des chercheurs ont planté les arbres dans deux sites différents :

- un terrain fertile et bien drainé en bordure de ruisseaux
- une crête au sol sablonneux

1.1 Manipulation des données sous *R*

Lire des données depuis un fichier

R est un langage informatique dédié à l'analyse statistique. On manipule les données sous *R* par l'intermédiaire de commandes qui doivent être saisies dans un interpréteur. Le plus souvent les données sont enregistrées dans un fichier que l'on demande à *R* de lire. *R* construit lors de cette lecture une «feuille de données» (*data.frame*) que nous devons ranger dans un objet. La commande ci dessous lit le fichier *peuplier.txt* et range le *data.frame* créé dans l'objet *peuplier*.

Exemple R - 1 :

La fonction *read.table* permet de lire un fichier. Le premier argument est le nom du fichier. L'argument *header* mit à *TRUE* indique que la première ligne du fichier contient le nom des colonnes

```
peuplier<-read.table("peuplier.txt",header=TRUE)
```

Un *data.frame* peut être vu comme un tableau constitué de colonnes et de lignes. Chaque colonne correspond à une variable et chaque ligne à un individu. Il est possible d'afficher le contenu d'un objet en écrivant son nom dans l'interpréteur.

Exemple R - 2 :

La saisie du nom d'un objet permet d'afficher son contenu. L'appel de la fonction *head* permet de n'afficher que le début de l'objet (tête). Sur votre écran les six premières lignes du jeu de données doivent s'afficher.

```
head(peuplier)
```

```
##   Site Annee Traitement Diametre Hauteur Poids Age
## 1    1     1          1      2.23    3.76  0.17  3
## 2    1     1          1      2.12    3.15  0.15  3
## 3    1     1          1      1.06    1.85  0.02  3
## 4    1     1          1      2.12    3.64  0.16  3
## 5    1     1          1      2.99    4.64  0.37  3
## 6    1     1          1      4.01    5.25  0.73  3
```

Sélectionner des variables dans un *data.frame*

Chacune des variables constituant le *data.frame* peut être obtenue individuellement. Il existe plusieurs façon pour réaliser cela. La plus simple est certainement de faire suivre le nom de l'objet contenant le *data.frame* du signe \$ et du nom de la variable souhaité : *peuplier\$Diametre*

Exemple R - 3 :

La saisie du nom d'un objet suivie du signe \$ et du nom d'une des variables contenues dans un *data.frame* permet de ne sélectionner que cette variable dans le *data.frame*.

Attention, le nom des objets et des variables est sensible à la casse (majuscule ou minuscule) prenez garde à les saisir correctement

```
peuplier$Diametre
```

```
##   [1] 2.23 2.12 1.06 2.12 2.99 4.01 2.41 2.75 2.20 4.09 3.62 4.77 1.39 2.89
##  [15] 3.90 1.52 4.51 1.18 3.17 3.33 4.88 2.73 3.05 2.11 1.03 4.84 5.28 1.66
##  [29] 1.57 4.38 2.21 4.00 2.45 2.59 2.89 5.50 3.13 1.37 2.07 4.45 3.26 4.19
##  [43] 4.03 2.42 1.30 3.00 2.93 2.21 1.44 1.91 4.11 1.89 3.75 2.73 3.41 1.17
##  [57] 1.58 5.73 2.64 4.65 3.76 4.92 3.09 4.22 5.13 6.14 2.28 5.46 5.84 2.72
##  [71] 2.57 4.95 3.03 1.94 3.50 2.66 4.70 5.07 5.24 5.33 4.78 4.46 5.50 5.54
##  [85] 4.07 4.18 1.67 5.42 3.35 5.04 4.69 4.62 5.36 3.15 4.10 5.18 4.72 5.41
##  [99] 3.20 4.47 3.92 4.65 4.47 4.47 5.25 2.65 5.75 5.89 5.02 3.31 4.49 6.25
## [113] 3.73 4.74 3.49 5.20 3.40 3.30 2.05 5.09 4.69 4.27 4.28 4.85 3.19 3.60
## [127] 4.45 4.14 1.85 3.89 2.80 2.05 1.95 3.06 4.76 4.17 4.18 4.45 3.98 3.29
## [141] 5.52 5.57 5.73 3.76 5.78 6.10 5.69 4.37 6.57 6.13 6.04 5.22 6.06 2.55
## [155] 3.62 6.01 6.88 4.02 7.05 7.47 7.12 4.05 7.07 4.98 5.91 5.48 6.39 5.29
## [169] 2.88 6.25 4.18 4.52 5.42 7.84 6.33 7.33 5.80 7.57 7.68 8.26 7.59 6.73
## [183] 5.45 5.29 6.38 4.71 4.94 5.42 2.15 7.43 8.15 5.02 6.41 6.73 5.89 6.43
## [197] 5.17 3.48 5.16 6.24 5.23 6.19 5.95 5.64 4.26 2.83 6.51 7.28 5.33 6.05
## [211] 6.69 7.54 6.94 5.48 6.58 7.71 7.52 7.33 6.65 6.98 6.63 6.93 6.30 6.51
## [225] 6.99 6.67 6.93 6.63 6.87 6.64 6.23 6.37 6.94 6.20 7.23 6.02 6.25 4.65
## [239] 4.76 5.51 5.39 5.22 4.83 4.95 5.20 7.42 6.57 7.05 7.12 7.41 7.30 7.79
## [253] 6.99 7.43 7.75 5.65 6.09 5.54 5.32 6.00 4.93 5.09 5.85 5.64 5.44 5.83
## [267] 5.93 5.65 6.48 6.11 6.07 6.18 6.21 6.24 6.41 6.75 6.45 6.77 5.78 5.50
## [281] 6.23 6.52 5.90 6.07 6.45 6.61 6.29 5.90 6.52 6.67 6.64 5.88 5.98 6.51
## [295] 7.02
```

Sélectionner des individus dans un *data.frame*

Il est de même possible de sélectionner toutes les données relatives à un individu (sélectionner une ou plusieurs ligne du *data.frame*). Pour réaliser cette sélection il faut s'appuyer sur le fait qu'un *data.frame* peut être considéré comme un tableau. Chaque case du tableau peut être individuellement sélectionnée en indiquant ses coordonnées x et y entre crochets séparées par une virgule :

- *peuplier*[x, y] où x indique un individu (ou ligne) et y une variable (ou colonne).
- Si l'on spécifie uniquement le numéro d'individu alors toutes les valeurs associées sont retournées.
- Il est aussi possible de spécifier une série d'individus par un intervalle : *peuplier*[$x_1 : x_2,]$
- ou par une liste d'individus : *peuplier*[$c(x_1, x_2, x_3, x_4),]$.

La selection combinée de certaines variables pour certains individus s'exprime donc selon le modèle suivant :

peuplier[$c(x_1, x_2, x_3, x_4), c(v_1, v_2)$].

Exemple R - 4 :

À l'aide des commandes suivantes, il est tour à tour possible de sélectionner : les données relatives à l'individu 1, les données des individus 2, 3, 4 et 5, les données des individus 3, 7 et 9 et les valeurs des variables *Annee* et *Hauteur* pour ces même individus. Faites attention au fait que le nom dans variables est ici entourés de simples quotes «'»

```
peuplier[1,]

##   Site Annee Traitement Diametre Hauteur Poids Age
## 1    1     1           1      2.23   3.76  0.17  3

peuplier[2:5,]

##   Site Annee Traitement Diametre Hauteur Poids Age
## 2    1     1           1      2.12   3.15  0.15  3
## 3    1     1           1      1.06   1.85  0.02  3
## 4    1     1           1      2.12   3.64  0.16  3
## 5    1     1           1      2.99   4.64  0.37  3

peuplier[c(3,7,9),]

##   Site Annee Traitement Diametre Hauteur Poids Age
## 3    1     1           1      1.06   1.85  0.02  3
## 7    1     1           1      2.41   4.07  0.22  3
## 9    1     1           1      2.20   4.17  0.19  3

peuplier[c(3,7,9),c('Annee', 'Hauteur')]

##   Annee Hauteur
## 3     1     1.85
## 7     1     4.07
## 9     1     4.17
```

Sélectionner des individus dans un *data.frame* selon un critère

La sélection d'individus peut aussi être réalisée selon des critères définis à partir des variables du *data.frame*. Nous pouvons par exemple sélectionner les peupliers d'un certain âge, plus haut qu'une

certaine taille... Pour cela, à la place d'indiquer comme précédemment des numéros d'individus, nous allons fournir la condition de sélection. Pour obtenir tous les peupliers âgés de x ans nous écrirons : `peuplier[peuplier$Age == x,]`. Les opérateurs de comparaisons sont :

Opérateur		représentation dans R
égalité	(=)	==
différent	(≠)	!=
supérieur	(>)	>
supérieur ou égale	(≥)	>=
inférieur	(<)	<
inférieur ou égale	(≤)	<=

TABLE 1 – Liste des opérateurs de comparaison

Exemple R - 5 :

La sélection d'individu selon un critère est possible en indiquant ce critère à la place du numéro des individus que l'on souhaite retenir. La première commande réalise la sélection des peupliers d'une hauteur supérieure à 10m. Il est toujours possible de combiner la sélection d'individus et de variables, comme le montre la seconde commande

```
peuplier[peuplier$Hauteur > 10,]
```

```
##      Site Annee Traitement Diametre Hauteur Poids Age
## 216     1     2          1     7.71   10.30  5.82  4
## 218     1     2          1     7.33   10.05  5.51  4
## 219     1     2          1     6.65   10.13  5.11  4
## 222     1     2          1     6.93   10.15  5.27  4
## 227     1     2          2     6.93   10.10  5.47  4
## 246     1     2          4     7.42   10.74  6.22  4
## 247     1     2          4     6.57   10.29  4.91  4
## 248     1     2          4     7.05   10.63  5.64  4
## 249     1     2          4     7.12   10.81  4.72  4
## 250     1     2          4     7.41   10.60  6.33  4
## 251     1     2          4     7.30   10.80  6.36  4
## 252     1     2          4     7.79   10.73  6.52  4
## 253     1     2          4     6.99   10.72  5.85  4
## 254     1     2          4     7.43   10.90  6.49  4
## 255     1     2          4     7.75   10.82  6.93  4
```

```
peuplier[peuplier$Hauteur > 10, c('Diametre', 'Age', 'Hauteur')]
```

```
##      Diametre Age Hauteur
## 216      7.71  4   10.30
## 218      7.33  4   10.05
## 219      6.65  4   10.13
## 222      6.93  4   10.15
## 227      6.93  4   10.10
## 246      7.42  4   10.74
```

```
## 247      6.57  4  10.29
## 248      7.05  4  10.63
## 249      7.12  4  10.81
## 250      7.41  4  10.60
## 251      7.30  4  10.80
## 252      7.79  4  10.73
## 253      6.99  4  10.72
## 254      7.43  4  10.90
## 255      7.75  4  10.82
```

Il est possible de combiner plusieurs critères de sélection grâce à des opérateurs logiques.

Opérateur		représentation dans R
Les deux conditions doivent être vraies	$(A \text{ et } B)$	<code>&</code>
L'une des deux conditions doit être vraie	$(A \text{ ou } B)$	<code> </code>
La condition doit être fausse	$(\text{non } A)$	<code>!</code>

TABLE 2 – Liste des opérateurs logiques

Exemple R - 6 :

La sélection d'individu selon plusieurs critères est possible combinant les critères à l'aide des opérateurs décrits dans la table 2. Ici nous sélectionnons simultanément sur un diamètre inférieur à 7m et une hauteur supérieure à 10m.

```
peuplier[peuplier$Diametre < 7 & peuplier$Hauteur > 10,
         c('Diametre', 'Age', 'Hauteur')]
```

```
##      Diametre Age Hauteur
## 219      6.65  4   10.13
## 222      6.93  4   10.15
## 227      6.93  4   10.10
## 247      6.57  4   10.29
## 253      6.99  4   10.72
```

1.2 Test d'un paramètre

Quel est l'intervalle de confiance de la moyenne du poids des arbres âgés de 4 ans et plantés l'année 1 ?

Construction du jeu de données

La réponse à cette question demande en premier de construire un jeu de données contenant l'ensemble des poids des arbres répondant aux deux critères énoncés.

Exemple R - 7 :

La première commande sélectionne les poids des arbres âgés de 4 ans et plantés la première année. Le résultat de cette sélection est affecté à un nouvel objet nommé *poids.4ans.annee1*. La deuxième commande affiche la liste des valeurs ainsi sélectionnées. La troisième affiche le nombre de valeurs retenues et la dernière dessine l'histogramme de ces valeurs

```
poids.4ans.annee1 <- peuplier$Poids[peuplier$Age==4 & peuplier$Annee==1]
poids.4ans.annee1

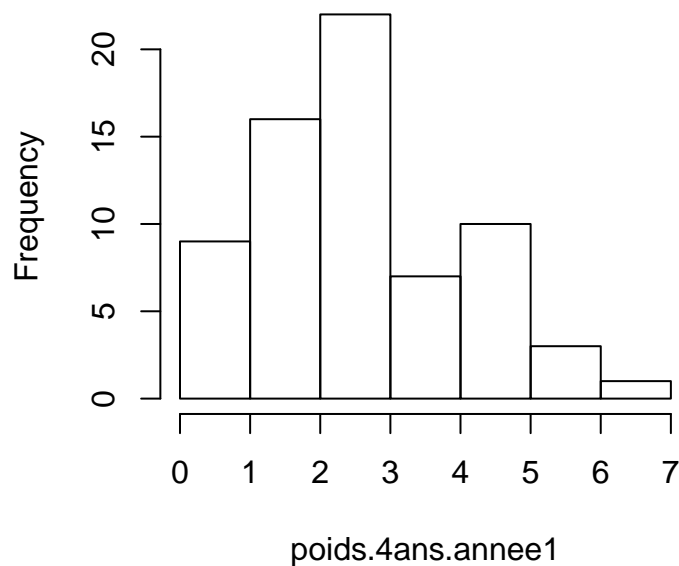
## [1] 1.24 3.24 2.51 2.68 1.71 2.61 0.25 0.82 2.27 3.82 1.10 4.16 4.92 4.38
## [15] 1.10 4.49 1.80 2.26 2.17 2.84 1.68 0.35 2.71 0.99 1.14 2.36 4.05 3.37
## [29] 4.34 2.90 5.21 5.12 6.19 4.80 2.59 1.89 1.61 1.92 1.09 1.32 0.64 0.16
## [43] 3.56 5.69 1.54 2.86 2.99 2.11 2.73 1.85 0.66 1.93 2.83 1.70 2.63 2.48
## [57] 2.34 0.99 0.39 3.24 4.28 2.36 2.85 3.83 4.79 4.33 2.12 3.83

length(poids.4ans.annee1)

## [1] 68
```

```
hist(poids.4ans.annee1)
```

Histogram of poids.4ans.annee1



Calcul des paramètres de l'échantillon

Le calcul de la moyenne est réalisé par la fonction *mean* l'estimateur de la variance de la population est calculé par la fonction *var*. L'écart type est obtenu par la fonction *sd*

```
moyenne.p.4.1 = mean(poids.4ans.annee1)
moyenne.p.4.1

## [1] 2.613382

var(poids.4ans.annee1)

## [1] 2.031748

sd(poids.4ans.annee1)

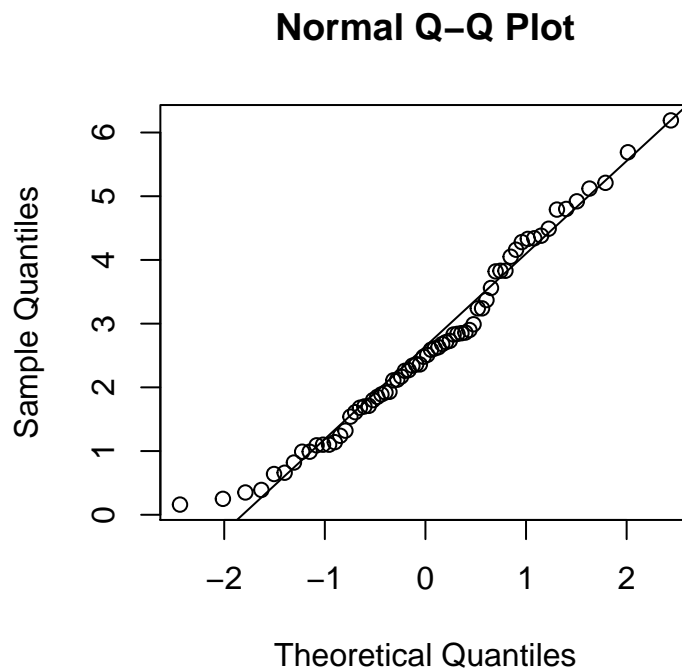
## [1] 1.425394
```

1.2.1 Test de la normalité d'un échantillon

Lorsque l'on calcule un intervalle de confiance, c'est la plupart du temps sous l'hypothèse d'une loi normale. Pour calculer notre intervalle de confiance en connaissance de cause, nous devons donc en premier tester la normalité des données.

Une méthode graphique permet de vérifier cette normalité en traçant les quantiles de l'échantillon en fonction des quantiles d'une loi normale. Sous l'hypothèse de normalité les points doivent s'aligner sur une droite.

```
qqnorm(poids.4ans.annee1)
qqline(poids.4ans.annee1)
```



Afin d’observer ce que donne un «qqplot» avec des données suivant réellement une loi normale, nous pouvons simuler un jeu de données possédant la même moyenne et le même écart type que nos données mais suivant une loi normale. La fonction `rnorm` permet de générer ces données. Elle accepte trois arguments dans l’ordre :

- le nombre de valeurs à générer
- la moyenne de ces valeurs
- l’écart type de ces valeurs

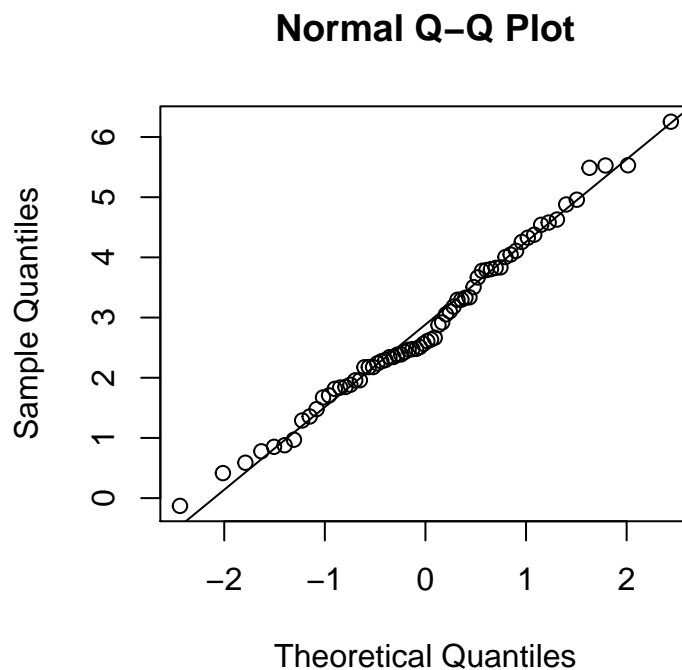
```
normal.alea = rnorm(length(poids.4ans.annee1),
                     mean(poids.4ans.annee1),
                     sd(poids.4ans.annee1))
```

```
normal.alea
```

```
## [1] 2.1775459 4.3274896 1.7069579 1.3560983 0.8522310 2.3875707
## [7] 0.7780788 2.2733455 4.1078693 3.6664103 2.2388402 1.2902732
## [13] 2.6086017 2.4748704 2.8760246 1.8370270 2.1740901 0.5882175
## [19] 1.9566746 4.9583377 3.8340956 -0.1290901 2.3412592 4.5787746
## [25] 1.4786787 2.4659835 3.3293692 4.6284173 0.4156773 2.3452896
## [31] 1.6730962 5.5306553 2.2946995 3.1038135 5.4882858 6.2543624
## [37] 2.5111437 4.8787635 3.3366600 5.5266776 1.8467607 1.8154948
## [43] 3.7894654 0.8789972 4.5416256 2.1779503 2.3826434 2.4798537
## [49] 0.9703575 4.0465823 4.0053448 2.5639160 3.8053713 3.7766749
## [55] 3.2962648 3.8247463 2.9189924 3.0534601 2.6348086 1.8839534
## [61] 3.2979476 1.9591701 2.4299095 2.6649031 3.5063204 4.2558697
## [67] 4.3768783 3.1820600
```



```
qqnorm(normal.alea)
qqline(normal.alea)
```



Vous pouvez relancer plusieurs fois ces commandes de manière à observer la fluctuation de l’aspect du «qqplot».

Il est aussi possible d’avoir une approche basée sur un test numérique, par exemple en calculant la variable W du test de *Shapiro – Wilk*. Pour cela la fonction `shapiro.test` permet à la fois de calculer cette valeur W et de la comparer à sa distribution théorique sous l’hypothèse nulle de normalité des données.

```
shapiro.test(poids.4ans.annee1)

##
##  Shapiro-Wilk normality test
##
## data:  poids.4ans.annee1
## W = 0.97367, p-value = 0.1592

shapiro.test(poids.4ans.annee1)$p.value

## [1] 0.1592436

shapiro.test(poids.4ans.annee1)$statistic

##           W
## 0.9736659

help(shapiro.test)
```

1.2.2 Calcul de l'intervalle de confiance

La moyenne du poids m que nous avons calculé à partir de notre échantillon est une estimation de la moyenne μ du poids de la population. m fluctue en fonction de l'échantillon que nous analysons. C'est donc une variable aléatoire se distribuant selon une loi normale d'espérance $E[m]$ (voir équation 1) et de variance $V[m]$ (équation 2) si le poids se distribue lui même selon une loi normale avec une moyenne μ et une variance σ^2 ¹.

$$E[m] = \mu \quad (1)$$

$$V[m] = \frac{\sigma^2}{n} \quad (2)$$

Donc l'erreur que l'on fait en estimant la moyenne : $\Delta = m - \mu$ est une variable normale de moyenne nulle et de variance $V[m]$. Il est toujours possible de diviser une variable aléatoire par son écart-type pour ramener sa variance à 1.

$\frac{\Delta}{\sqrt{\frac{\sigma^2}{n}}}$ est donc une variable normale de moyenne nulle et de variance égale à 1. Malheureusement nous ne connaissons pas σ^2 mais seulement sont estimation sur l'échantillon S . Il est donc possible de réécrire cette variable de la manière suivante :

$$t = \frac{\Delta}{\sqrt{\frac{S^2}{n}}} \quad (3)$$

Qui est une variable se distribuant selon une loi de *student* à $n - 1$ degrés de liberté. Pour un risque global de 5% l'écart type peut se calculer de la façon suivante

$$\bar{x} + t_{\alpha/2}^{n-1} \sqrt{\frac{S^2}{n}} < \mu < \bar{x} + t_{1-\alpha/2}^{n-1} \sqrt{\frac{S^2}{n}} \quad (4)$$

où $t_{\alpha/2}^{n-1}$ est le quantile de la distribution t de *student* pour $n - 1$ degrés de liberté et une probabilité $\alpha/2$. Ce quantile est calculé par la fonction `qt` de R. Elle accepte comme paramètres :

- une probabilité en première argument
- un nombre de degrés de liberté en second argument

Attention :

Du fait de la symétrie de la loi de student

$$-t_{\alpha/2}^{n-1} = t_{1-\alpha/2}^{n-1} \quad (5)$$

De plus la loi de *student* étant définie avec une moyenne nulle :

$$t_{\alpha/2}^{n-1} < 0 \quad (6)$$

```
mm <- mean(poids.4ans.annee1)
sm <- sqrt(var(poids.4ans.annee1)/length(poids.4ans.annee1))
df <- length(poids.4ans.annee1)-1
alpha <- 0.05
t <- qt(alpha/2, df)
t
```

1. Le théorème centrale limite indique que la variable m peut être considérée comme normale quand la taille de l'échantillon est grand

```
## [1] -1.996008

erreur <- -t * sm
erreur

## [1] 0.3450189
```

L'erreur calculée doit ensuite être ajoutée ou retranchée à la moyenne estimée pour obtenir respectivement la borne supérieure et la borne inférieure de l'intervalle de confiance.

```
borne.min <- mm - erreur
borne.max <- mm + erreur
borne.min

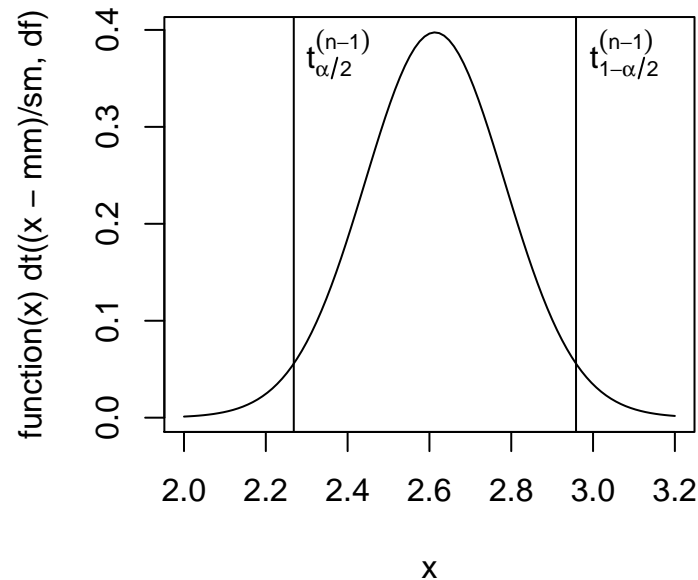
## [1] 2.268363

borne.max

## [1] 2.958401
```

Nous pouvons enfin visualiser graphiquement la distribution de la variable aléatoire *mm*

```
plot(function(x)dt((x - mm) / sm ,df),2,3.2)
abline(v=borne.min)
abline(v=borne.max)
text(borne.min,0.35,expression(t[alpha/2]^(n-1)),adj = c(-0.2,0))
text(borne.max,0.35,expression(t[1-alpha/2]^(n-1)),adj = c(-0.2,0))
```



Comparaison d'une moyenne observé à une moyenne théorique

On voudrait savoir si le poids moyen des arbres âgés de 4 ans et plantés l'année 1 est égal à 2 kg.
Faire les calculs dans les deux sites.

Nous devons en premier lieu préparer les échantillons

```
poids.4ans.annee1.site1 <- peuplier$Poids[peuplier$Age==4
                                         & peuplier$Annee==1
                                         & peuplier$Site==1]
poids.4ans.annee1.site2 <- peuplier$Poids[peuplier$Age==4
                                         & peuplier$Annee==1
                                         & peuplier$Site==2]

length(poids.4ans.annee1.site1)
## [1] 34

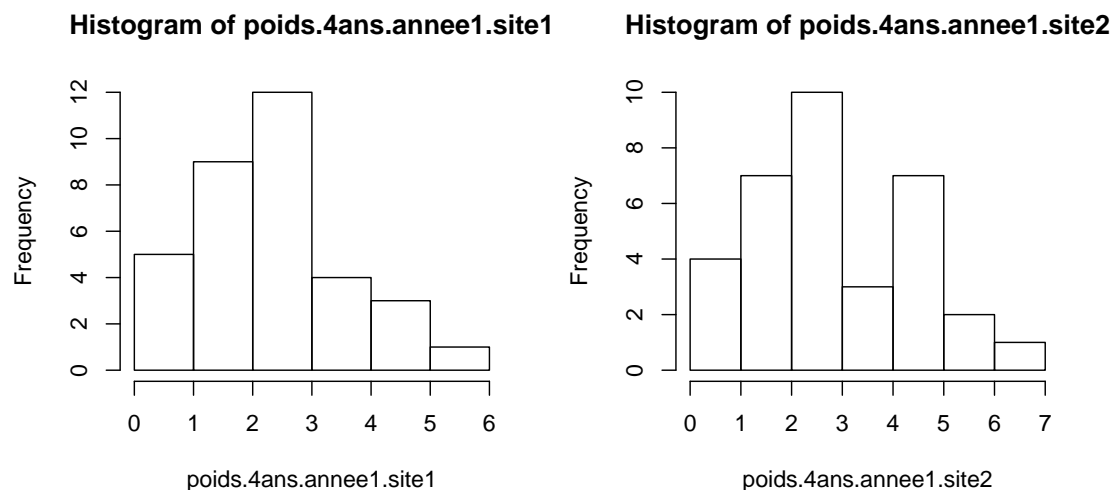
length(poids.4ans.annee1.site2)
## [1] 34

poids.4ans.annee1.site1
## [1] 2.59 1.89 1.61 1.92 1.09 1.32 0.64 0.16 3.56 5.69 1.54 2.86 2.99 2.11
## [15] 2.73 1.85 0.66 1.93 2.83 1.70 2.63 2.48 2.34 0.99 0.39 3.24 4.28 2.36
## [29] 2.85 3.83 4.79 4.33 2.12 3.83
```

```
poids.4ans.annee1.site2

## [1] 1.24 3.24 2.51 2.68 1.71 2.61 0.25 0.82 2.27 3.82 1.10 4.16 4.92 4.38
## [15] 1.10 4.49 1.80 2.26 2.17 2.84 1.68 0.35 2.71 0.99 1.14 2.36 4.05 3.37
## [29] 4.34 2.90 5.21 5.12 6.19 4.80

par(mfrow=c(1,2))
hist(poids.4ans.annee1.site1)
hist(poids.4ans.annee1.site2)
```



```
mean(poids.4ans.annee1.site1)

## [1] 2.415588

mean(poids.4ans.annee1.site2)

## [1] 2.811176
```

Nous devons réaliser un test de *student* comparant la moyenne estimée par la fonction *mean* avec la moyenne théorique escomptée $E = 2$ dans notre cas. Nous savons pour réaliser le test t que la variable auxiliaire T

$$T_{poids} = \frac{\overline{poids} - E_{poids}}{\sqrt{\sigma_{poids}^2/n}} \quad (7)$$

suit une loi de *student*.

```
m1 = mean(poids.4ans.annee1.site1)
s1 = sqrt(var(poids.4ans.annee1.site1)/length(poids.4ans.annee1.site1))
df1 = length(poids.4ans.annee1.site1) - 1
m1
```

```
## [1] 2.415588
s1
## [1] 0.2197236
df1
## [1] 33
t1 = (m1 - 2)/s1
t1
## [1] 1.891414
```

La fonction $pt(x, df)$ retourne la probabilité d'observer une valeur de la variable t de *student* supérieure à x avec df degrés de liberté.

Si nous considérons que l'hypothèse alternative est : la moyenne observée est différente de 2 alors la $pvalue$ du test est :

```
(1 - pt(t1, df1))*2
## [1] 0.06737366
```

Pour le site 1 la $pvalue$ est supérieur au risque $\alpha = 0.05$ donc nous ne pouvons pas rejeter l'hypothèse H_0 .

Refaites le même calcul pour le site 2.

Il existe une fonction *t.test* qui réalise un test de *student* complet à un ou deux échantillons. Elle nous permet de réaliser plus facilement ce même test de *student*.

```
help(t.test)
t.test(poids.4ans.annee1.site1, mu=2)

##
## One Sample t-test
##
## data:  poids.4ans.annee1.site1
## t = 1.8914, df = 33, p-value = 0.06737
## alternative hypothesis: true mean is not equal to 2
## 95 percent confidence interval:
##  1.968557 2.862619
## sample estimates:
## mean of x
##  2.415588

t.test(poids.4ans.annee1.site2, mu=2)

##
## One Sample t-test
##
## data:  poids.4ans.annee1.site2
```

```
## t = 3.0513, df = 33, p-value = 0.004475
## alternative hypothesis: true mean is not equal to 2
## 95 percent confidence interval:
##  2.270302 3.352051
## sample estimates:
## mean of x
##  2.811176
```