# 2V481 – BM2

# Cours 1

LICENCE SCIENCES ET TECHNOLOGIE
MENTION SCIENCE DE LA VIE – L2

LARSEN MARTIN
DEMEYRIER VIRGINIE
RYBARCZYK HERVÉ

# Rappels concernant R

LICENCE SCIENCES ET TECHNOLOGIE
MENTION SCIENCE DE LA VIE – L2

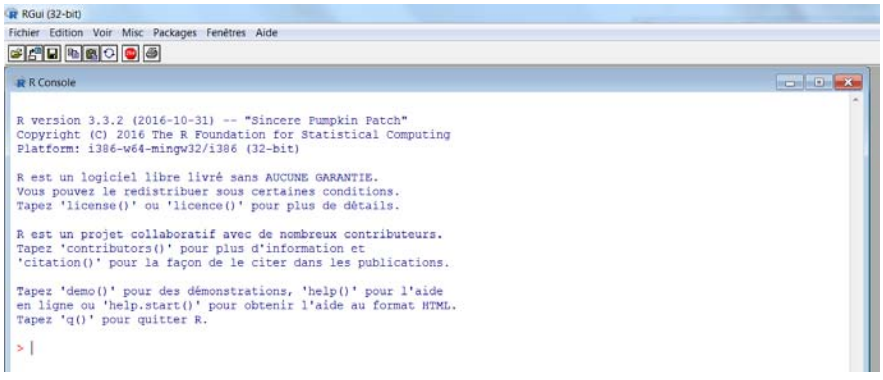
LARSEN MARTIN
DEMEYRIER VIRGINIE
RYBARCZYK HERVÉ

## Rappels sur R

- Il est construit autour d'un noyau de base contenant les principales fonctions. Le package nommé « base » est en quelque sorte le cœur de R.
- Il fait appel à des bibliothèques ou « packages » pour des utilisations plus spécifiques.
- Ces paquets sont en constant développement (communauté très dynamique).
- Il dispose d'une aide importante et pratiquement tous vos problèmes ou questions vont trouver une réponse sur un des innombrables forums.

## L'interface de R

- Une fois R lancé vous vous trouvez dans un fenêtre : la console de commande



## L'interface de R

- Du texte apparaît et un prompt « > » vous invite à saisir vos commandes.
- R manipule des objets
- MAIS ATTENTION : ceux-ci doivent IMPERATIVEMENT commencer par une lettre (A à Z ou a à z) et peuvent contenir des chiffres, des points ou des espaces soulignés.
- L'oublier est souvent source d'erreur et d'énervement.

## L'interface de R Studio

R Studio combine 4 éléments

- 1. Le terminal R
- 2. Un éditeur de texte / script lié au terminal R
- 3. Interface d'environnement et d'historique (par exemple, les valeurs de variables et de constantes)
- 4. Une interface pour les interactions entre R et les éléments externes (Fichiers, Graphiques, Packages, Aide et Viewer)

R studio représente un environnement complet facilitant l'utilisation de R.

## Jouons un peu avec R

- Créons un objet « n » et donnons lui la valeur 5 (par exemple)

>n<-5

Faire entrer pour valider la commande

Réponse de R : ?

- Voyons ce que R a fait :

>n

Réponse de R :

[1] 5

- Appliquer une fonction:

>pnorm(1.96)

[1] 0.9750021

- La fonction pnorm : N(0,1) loi Normale centrée-réduite
- ATTENTION : Les décimaux TOUJOURS avec un « . »

- Refait avec l'éditeur de script. (Comment line(s) of script: CTRL+SHIFT+C; Run line(s) of script: CTRL+ENTER)

## Jouons un peu avec R

- On peut entrer des « noms » dans des variables
  - Attention le point-virgule s'utilise pour séparer des commandes distinctes sur la même ligne

> name <- "Carmen"; n1 <- 10; n2 <- 100; m <- 0.5

- La fonction ls (pour list) permet d'afficher une liste simple des objets en mémoire :

> ls()

[1] "m" "n1" "n2" "name"

- Seuls les « noms » des objets apparaissent ….

## Jouons un peu avec R

- Il peut être utile de « nettoyer » les variables en effaçant leur contenu.
- La commande rm() pour « remove » s'en charge :

> rm(n)

> n

Erreur : objet 'n' introuvable

- La commande > ?pnorm
  - Va vous fournir l'aide sur la fonction « pnorm » par exemple.
  - Valable pour toutes les fonctions !!

## Jouons un peu avec R

- Définissons un objet « longueur »
  - ➢ longueur <- c(32,40,42,38,41.5,43.7,39.5,41.25,38.65,40.8)
- Nous avons créé un objet, de type « VECTEUR », à une dimension contenant 10 valeurs. La commande suivante le vérifie :

> length(longueur)

> longueur

[1] 32.00 40.00 42.00 38.00 41.50 43.70 39.50 41.25 38.65 40.80

- Nous donne bien le contenu numérique du vecteur.

## R comme language de programmation

- Function:
  - Foo <- function(X){2*x + 2}
- Conditional expression
  - If (x == 7){print('n=7')} else {print('n!=7')}
  - Ifelse(x==7,print('n=7'),print('n!=7'))
- Loop (e.g For loop)
  - For (i in 1:5){print(i)}

## Les principaux types d'objets sous R

- Les vecteurs (variables avec valeurs assignées)

**un exemple de vecteur sous R** : longueur <- c(32,40,42,38,41.5,43.7,39.5,41.25,38.65,40.8)

- Les data frame (tableaux de variables)

**un exemple de data.frame sous R**

| | Dates | Fluo | Turbidity | Salinity | Temp | Chla | Pheo | PheoT | Tchla | NO2.NO3. | PO4 | SiO4 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2009-12-15 | 1.6894600 | 1.5664000 | 35.60542 | 25.16604 | 0.9481284 | 0.1636948 | 14.72310 | 1.1118233 | 0.3931333 | 0.07053333 | 0 |
| 2 | 2010-01-15 | 1.7694375 | 1.4226875 | 35.26078 | 25.53967 | 0.5415489 | 0.1337643 | 19.80774 | 0.6759132 | 0.5184000 | 0.08123333 | 0 |
| 3 | 2010-02-15 | 0.5810062 | 1.1380000 | 35.34602 | 26.84876 | 0.6401712 | 0.1104446 | 14.71386 | 0.7506156 | 0.1988333 | 0.05093333 | 0 |
| 4 | 2010-03-16 | 0.0000000 | 0.5980000 | 35.30830 | 25.13370 | 0.6636906 | 0.1803576 | 21.36816 | 0.8440481 | 0.2657667 | 0.06903333 | 0 |
| 5 | 2010-04-15 | 0.0000000 | 0.6170000 | 35.39920 | 25.27000 | 0.6343510 | 0.1034000 | 14.01502 | 0.7377798 | 0.1065000 | 0.10760000 | 0 |
| 6 | 2010-05-17 | 0.7271211 | 0.6723684 | 35.43810 | 24.78147 | 0.4321119 | 0.0800000 | 15.61877 | 0.5122043 | 0.1167333 | 0.02980000 | 0 |

- Les tables (tableaux de contingences)
- Les matrices

**un exemple de table de contingence sous R**

| | A | B | Sum |
|---|---|---|---|
| C | 70 | 150 | 220 |
| D | 100 | 350 | 450 |
| Sum | 170 | 500 | 670 |

## Quelques commandes

L'étendue : range(longueur) donne [1] 32.0 43.7
La moyenne : mean(longueur) donne [1] 39.74
La médiane : quartile(longueur,0.50) donne ????
L'écart-type : sqrt(var(longueur)) donne [1] 3.189201
L'erreur standard (L'écart-type de l'espérance):
es=sqrt(var(longueur))/(sqrt(length(longueur)))
Le coefficient de variation :
cv=(sqrt(var(longueur))/mean(longueur))*100

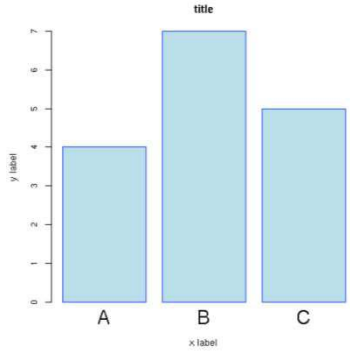
Le coefficient de variation permet de comparer des évolutions de variables n'ayant pas la même échelle ex: croissance d'une souris et d'un éléphant …

## Représentation graphique

- Variable aléatoire discrète : Diagramme en barre barplot()

vect <- c(a = 4, b = 7, c = 5)

**barplot(**vect, col = "lightblue", border = "blue", names.arg =toupper(names(vect)), cex.names = 2, cex.axis = 0.8, main = "title", xlab = "x label", ylab = "y label", axes = TRUE**)**



## Représentation graphique

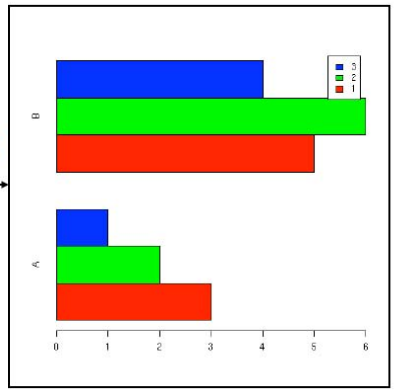
- mat <- matrix(c(3, 2, 1, 5, 6, 4), nrow = 3, dimnames = list(c("1", "2", "3"), c("a", "b")))

Ou

- mat <- matrix(c(3, 2, 1, 5, 6, 4), nrow = 3)
- row.names(mat) <- c("1", "2", "3")
- colnames(mat) <- c("a", "b")

| | a | b |
|---|---|---|
| 1 | 3 | 5 |
| 2 | 2 | 6 |
| 3 | 1 | 4 |

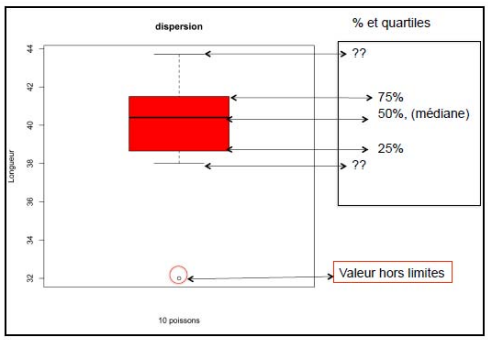
barplot(mat)




## Représentation graphique

- Variable aléatoire continue : Boîte à moustache

longueur <- c(32,40,42,38,41.5,43.7,39.5,41.25,38.65,40.8)
plotTmp <- boxplot(longueur,col="red")
Essai plotTmp$




Ajoutons la moyenne
> points(1,mean(longueur),col="black")
> points(1,median(longueur),col="green")

- Les valeurs des moustaches s'obtiennent par la formule suivante :

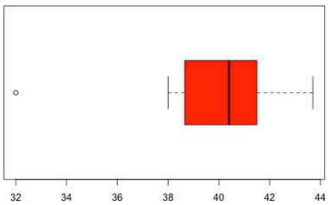
Upper whisker = min(max, max(non outliers), Q3 + IQR*1.5)
Lower whisker = max(min, min(non outliers), Q1 - IQR*1.5)

Une valeur « hors limite » on l'appelle un « outlier » (=plotTmp$out)

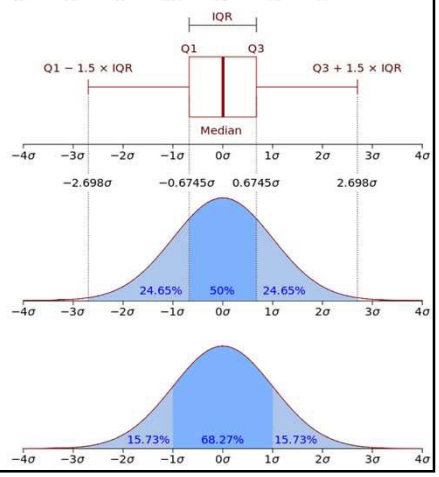
## Représentation graphique

- Variable aléatoire continue : Boîte à moustache

longueur <- c(32,40,42,38,41.5,43.7,39.5,41.25,38.65,40.8)
plotTmp <- boxplot(longueur,col="red")
Essai plotTmp$




Ajoutons la moyenne
> points(1,mean(longueur),col="black")
> points(1,median(longueur),col="green")




- Les valeurs des moustaches s'obtiennent par la formule suivante :

Upper whisker = min(max, max(non outliers), Q3 + IQR*1.5)
Lower whisker = max(min, min(non outliers), Q1 - IQR*1.5)

Une valeur « hors limite » on l'appelle un « outlier » (=plotTmp$out)




## Représentation graphique

- Ces quantiles peuvent s'obtenir par la formule suivante :
> quantile(longueur,0.25)
où 0.25 correspond au 1er quartile soit 25% de la distribution
et cette commande retourne comme résultat :
25%
38.8625
Cela veut dire que 25% des valeurs de longueur la précédent

- Quelles bornes pour 10,50,75,90% de la distribution?

## Représentation graphique

- Une commande simple résume les différents paramètres de position d'une distribution des valeurs d'une V.A.C.

> summary(longueur)

qui donne

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max |
|---|---|---|---|---|---|
| 32.00 | 38.86 | 40.40 | 39.74 | 41.44 | 43.70 |

## Représentation graphique

- On retire la valeur considérée comme aberrante et on recommence l'analyse des données.

- On forme un nouveau vecteur à partir de longueur auquel on a retiré la 1ère valeur
  - new <- longueur[-1]
  - new <- longueur[!(longueur %in% plotTmp$out)] # Alternative plus general que longueur != plotTmp$out (Ajout 32.5 au « longueur »)

On retire le premier terme (si c'est lui qui est à retirer)
  - length(new)

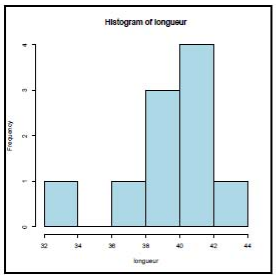
- Résumons ce que nous venons de voir et enrichissons le:
  - new <- c(40,42,38,41.5,43.7,39.5,41.25,38.65,40.8)
  - boxplot(new,horizontal=T,col="red",main="dispersion",xlab="9 poissons", ylab="Longueur", add=FALSE)

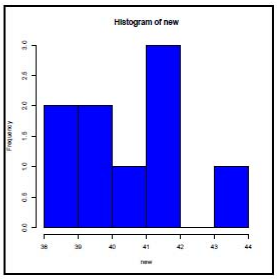
## Représentation graphique

- Traçons maintenant l'histogramme des données (effectifs ou fréquences):

> hist(longueur,col="lightblue",lwd=3,nclass=5)

> hist(new,col="blue",lwd=3,nclass=5)




## Représentation graphique

- Pour afficher plusieurs graphes dans la même fenêtre graphique, on peut la diviser en plusieurs sous parties avec la commande

par(mfrow=c(2,2))  # mfrow=c(nrows, ncols) crée une matrix de nrows x ncols remplis par ligne.

par exemple qui divise la fenêtre en 2 rangs et 2 colonnes (2x2) soit 4 sous graphes

- Essayez :

> par(mfrow=c(2,2))

> hist(longueur, col="lightblue", lwd=3)

> hist(new, col="blue", lwd=3)

## Représentation graphique

- L'histogramme (de longueur ou de new) n'est pas très représentatif d'une distribution spécifique.
- Il apparait plutôt incomplet par rapport à ce que l'on pourrait attendre d'une distribution de la taille (variable aléatoire continue dans la population de poissons de cette espèce).
- Simulons la distribution de plusieurs échantillons dont les valeurs suivent une **loi normale** avec n = 20, 50, 100, 1000 avec les mêmes paramètres :
  - moyenne de new = 40.6
  - écart-type de new =1.76

## Représentation graphique

- On va utiliser la commande « rnorm(**a,b,c**) » qui permet de tirer au hasard

**a** : nombre de valeurs d'une distribution **normale** de paramètres

**moyenne** =**b**

et **écart-type** =**c**

- Ce qui dans notre exemple nous donne

**>X=rnorm(9,40.6,1.7)**

**>hist(X)**

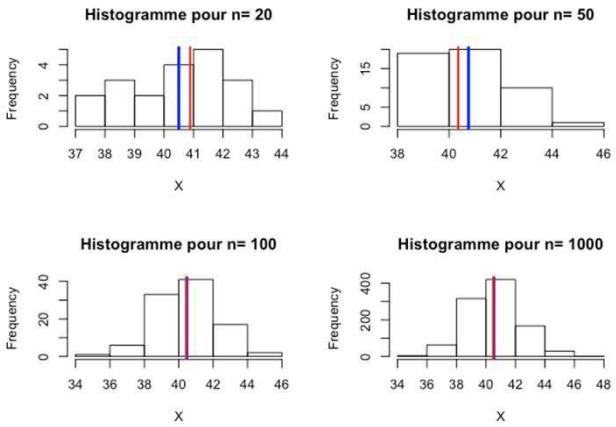
La commande **abline(v=mean(X))** qui permet de tracer une droite v pour verticale et de constante **mean(X)** et on recommence pour la médiane.

## Représentation graphique

- # simulation de data Echantillon / Population

```
par(mfrow=c(2,2))

X=rnorm(20,40.6,1.76); hist(X,main="n=20")
abline(v=mean(X),lwd=3);abline(v=median(X),lwd=2,col="red")

X=rnorm(50,40.6,1.76);hist(X,main="n=50")
abline(v=mean(X),lwd=3);abline(v=median(X),lwd=2,col="red")

X=rnorm(100,40.6,1.76);hist(X,main="n=100")
abline(v=mean(X),lwd=3);abline(v=median(X),lwd=2,col="red")

X=rnorm(1000,40.6,1.76);hist(X,main="n=1000")
abline(v=mean(X),lwd=3);abline(v=median(X),lwd=2,col="red")
```

## Représentation graphique




Conclusion des simulations ?

Effectifs, Echantillons, Population, …

## Représentation graphique

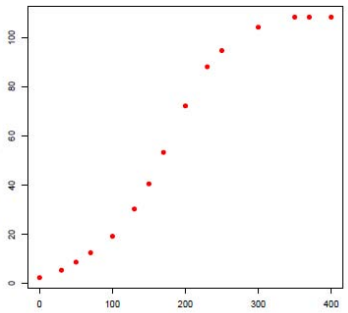
- Diagrammes binaires pour les variables continues : nuage de points ou « scatter plot » plot()

- **Exemple de croissance bactérienne Y en fonction du temps X**

**x <- c(0, 30, 50, 70, 100, 130, 150, 170, 200, 230, 250, 300, 350,370, 400)**

**y <- c(2.4, 5.4, 8.7, 12.5, 19.2, 30.3, 40.5, 53.3, 72.2, 88.1,94.7, 104.2, 108.3, 108.3, 108.3)**

**plot(y~x,pch=19,col="red")**




## Résumé des principales commandes

- > signifie que R attend vos instructions
- <-, -> ou = (correspond a <-) assigne une valeur ou du texte à un objet
- c(…,…) permet d'assigner plusieurs éléments dans un objet
- le . sépare les décimales des unités
- la , sépare les éléments d'un vecteur ou d'une commande
- le ; sépare les commandes dans une même ligne
- barplot()
- boxplot()
- hist()
- plot(x,y) ou plot(y~x)

# Principaux tests univariés sous R

LICENCE SCIENCES ET TECHNOLOGIE

MENTION SCIENCE DE LA VIE – L2

## Saisie des données pour R

- Deux échantillons où X :

VA représentant la taille d'individus d'une espèce de poissons

- 1. Création de la table sous Excel
- (2. On s 'assure des points au lieu des virgules)
- 3. On enregistre « sous » au format « .txt » avec tabulation

- Pour récupérer le fichier créé à partir d'excel …dans un script sous R

# nouveau script : test de comparaisons de moyennes

>donnees<-read.table(file.choose(),header=TRUE)

>attach(donnees)

| adu1 | adu2 |
|---|---|
| 46.01668 | 45.31913 |
| 49.47434 | 47.04900 |
| 50.60362 | 35.20589 |
| 42.30919 | 48.79993 |
| 51.14163 | 40.33424 |
| 49.3686 | 42.51932 |
| 42.94062 | 49.57548 |
| 49.82174 | 52.00621 |
| 48.73526 | 40.59082 |
| 64.07578 | 43.75296 |
| 57.55783 | 51.04380 |
| 59.44963 | 35.72993 |
| 52.63457 | 44.08565 |
| 54.87269 | 42.33216 |
| 42.50207 | 43.10860 |
| 39.98742 | 46.59471 |
| 47.96577 | 43.49734 |
| 45.51302 | 46.27165 |
| 57.83556 | 49.00498 |
| 46.34471 | 41.97896 |

## Histogramme et normalité

- Comme nous avons 2 échantillons on va réaliser des graphiques avec les deux échantillons et des couleurs différentes pour chaque : rouge pour adu1 et bleue pour adu2

- Script (histogramme des distributions)

hist(adu1,col='red')

hist(adu2,col='blue')

- Une fois les graphes et les premières statistiques obtenues, on va vérifier la normalité des deux séries

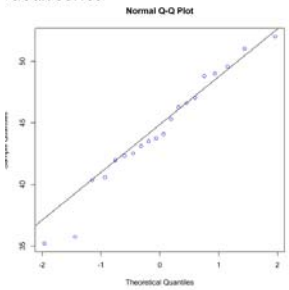
Traçons la distribution des quartiles sous forme linéaire….

qqnorm(adu1,col='red')

qqline(adu1,col='black')

qqnorm(adu2,col='blue')

qqline(adu2,col='black')




## Normalité et homoscédasticité

- On va appliquer le test de Shapiro et Wilk :

Shapiro.test (adu1)

Shapiro.test (adu2)

  - Valeur a 5% pour n=20 : 0,905
  - pas de rejet de la normalité

- Pour l'égalité des variances, on va appliquer le test de Fisher

var.test (adu1,adu2)

p.value ?????

## Test statistique

- On peut faire un test t classique de comparaison de moyennes sous la condition de normalité et d'égalité des variances (homoscedasticity) que l'on indique comme argument
  - test <- t.test (adu1,adu2,var.equal=TRUE )

- Et on récupère le ddl entier avec Two Sample t-test

data: adu1 and adu2

t = 3.1683, **df = 38**, p-value = 0.003023

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

1.992086 9.042911

sample estimates:

mean of x mean of y

49.95754 44.44004

## Test statistique

- Hypothèses unilatérales

>?t.test

## Default S3 method:

  - test <- t.test(x,y , alternative = ***c("two.sided", "less", "greater")***, mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)
  - Explore test$....

- Les autres arguments possibles

mu =0

Paired =FALSE ou TRUE

Var.equal=FALSE (Welch's t-test) ou TRUE

Conf.level=0.95

## Test statistique

- Comparaison moyenne à une norme ou une référence :

>Vec<-c(2.4,2.5,3.1,1.9,3.5)

Création du vecteur de données

>t.test(vec,mu=2)

Les sorties possibles et extractions avec $

tes<- t.test(vec,mu)

tes$p.value

tes$parameter

tes$conf.int

tes$statistic

## Test statistique

- Imaginons maintenant que les deux échantillons sont appariés …

t.test(adu1,adu2,paired=T)

CONCLUSION ?

- Supposons que la normalité et que l'égalité des variances ne sont pas respectées…

- On doit alors faire un test non paramétrique :

≪ Test des médianes ≫ qui va comparer comment les valeurs sont rangées entre les deux échantillons (tests des rangs)

>wilcox.test(adu1,adu2)

CONCLUSION ?

## Comparaison de pourcentages

- Prenons le cas de deux grossistes qui obtiennent après un contrôle qualité le tableau suivant :

| | A | B |
|---|---|---|
| Total | 96 | 55 |
| Défaut | 12 | 15 |

- On cherche à savoir si les deux grossistes se fournissent chez le même fabricant ?

  - defauts <- c( 12, 15 )
  - pieces <- c( 96, 55 )
  - prop.test(defauts, pieces)

On crée deux vecteurs defauts et pieces et on va comparer le nombre de défauts dans les deux échantillons

## Test du Chi 2 de conformité

- Monsieur GRAIN semencier de Monsieur LEON (cultivateur) affirme que la variété de semences achetées par Mr LEON donne

50% de pieds à épis ≪ jaune ≫

30% de pieds à épis ≪ jaune/rouge ≫ (ou orange plus simple en R)

20% de pieds à épis ≪ rouge ≫

- Lors de la récolte Mr LEON observe :

48 épis ≪ jaune ≫, 22 épis ≪ jaune/rouge ≫ et 29 épis ≪ rouge ≫

## Test du Chi 2 de conformité

**Hypothese:**
**H0:** La distribution observée lors de la récolte est celle annoncée par Mr GRAIN
**H1:** La distribution observée lors de la récolte n'est pas celle annoncée par Mr GRAIN

- Script :

```
> ##### CHI CARRE #####
> #### Monsieur Leon et son Mais ###
> ###################################
> # créons un vecteur avec les effectifs observés
> obs<-c(48,22,29)
># Transformons le en table pour avoir des entêtes de colonnes
> obs<-as.table(obs)
> obs
> # affichons les noms de colonnes
> names(obs)
```

## Test du Chi 2 de conformité

- Script :

```
> # changeons les entêtes de colonnes avec les couleurs de grains
> names(obs)<-c("Jaune","Jaune.Rouge","Rouge")
> obs
>addmargins(obs) # tableau de contingence
># calcul des prop observées #
> pi.obs<-obs/margin.table(obs)
# Proportions théoriques
pi.theo<-c(0.5,0.3,0.2)
```

**chisq.test(obs,p=pi.theo)**

## Test du Chi 2 d'Independence

- En plus de la couleur, on travaille avec le caractère d'enracinement.

| | Faible | Fort | Moyen | Tresfort |
|---|---|---|---|---|
| Jaune | 13 | 6 | 17 | 12 |
| Orange | 2 | 7 | 3 | 10 |
| Rouge | 3 | 13 | 8 | 5 |

```
> ## On rentre une matrice avec 3 lignes
> mat<-matrix(c(13,2,3,6,7,13,17,3,8,12,10,5),nrow=3)
> ## noms des lignes
> rownames(mat)<-c("Jaune","Orange","Rouge")
> ## noms des colonnes
> colnames(mat)<-c("Faible","Fort","Moyen","Tresfort")
```

## Test du Chi 2 d'Independence

- Script :

```
> ## Tableau de contingence
> addmargins(mat)
        Faible Fort Moyen Tresfort Sum
Jaune 13 6 17 12 48
Orange 2 7 3 10 22
Rouge 3 13 8 5 29
Sum 18 26 28 27 99
> ## graphe du tableau de contingence
> mosaicplot(mat)
> ## test du chi 2 d'Independence
> chisq.test(mat)
        Pearson's Chi-squared test
data: mat
X-squared = 17.9626, df = 6, p-value = 0.006326
```

**Hypothese:**
**H0:** Couleur et enracinement sont independent
**H1:** Couleur et enracinement ne sont pas independent