**Master1 : Probabilités et Statistique**

**Module : Initiation au langage R**

**Introduction au langage R (cours 1)**

Le logiciel R est un logiciel de calcul statistique, créé par *Ross Ihaka* et *Robert Gentleman*. C'est un langage dérivé de S, disponible sur

http://cran.r-project.org /

Documentation sur le logiciel R :

www.math.sciences.univ-nantes.fr/:philippe/R.html

site consacré aux graphiques :

addictedtor.free.fr/graphiques/

R est gratuit (open source), fonctionne sur plusieurs plateformes (windows,MacOSX, Linux ,…) , c'est un langage de programmation puissant.

Au démarrage

> est l'invite des commandes : apparait automatiquement en début de chaque ligne de commandes.

+ apparait en début de la ligne si la ligne précédente est incomplète

Object : sont les éléments de base au langage R qui peuvent être des données (vecteurs, matrices, …) , des fonctions, des graphiques.

Les objets des R se différencient par leur mode, qui décrit leur contenu, et leur classe qui décrit leur structure.

Les principales classes d'objets sont : vecteurs, matrices, array , factor , time series , data.frame, liste…

Les principaux types d'objets sont :

1) entiers,réel, complexe

2) caractere

3) logique : TRUE, FALSE

Pour connaitre le mode (le type) et la classe (largeur) d'un objet on peut utiliser, resp. les fonctions : mode() ; length()

Quand R est utilisé, les variables, les données, les fonctions , les résultats…sont stockés dans la mémoire de l'ordinateurs sous forme d'objets qui ont chacun un

nom. L'utilisateur va agir sur ces objets avec des opérateurs (arithmétique, logiques, de comparaison,…) et des fonctions

Exemple

>x<-2

>mode (x)

Numeric

>length(x)

1

- Avec R une fonction, pour être exécuter, s'écrit toujours avec des parenthèses, même si elles ne contiennent rien (comme par exemple Is())

>fonction(arg,arg ,…)

Les arguments peuvent être des objets (données, formules, expression) dont certains peuvent être définis par défaut dans la fonction, ces valeurs par défaut peuvent être modifiées par l'utilisateur avec les options.

Les fonctions disponibles sont stockées dans une bibliothèque localisée sur le disque dans le répertoire.

**Calcul matricielle avec R**

Soient α un scalaire, A , B deux matrices réelles et C une matrice complexe.

Opération de base

> alpha+A # l 'addition d'un scalaire α+A ; > alpha*A # la multiplication par un scalaire α.A.

>A+B #l'addition terme à terme. >A-B # la soustraction terme à terme.

>t(A) # la transposition. >cong(C) #la conjuguée.

>A*B # la multiplication terme à terme. >A% *%B # multiplication matricielle.

>A/B # la division terme à terme. >solve(B) # l'inversion matricielle.

>A% *% solve(B) # la division matricielle. > t(A) % *%B # produit avec la transposition.

>det(A) # déterminant d'une matrice. >sum(diag(A))# calcul de la trace de A.

>sqrt(t(A) % *%A) ou bien > sqrt(A % *%t(A))#la norme d'une matrice ligne ou colonne.

>eigen(A) # calcul des valeurs propres et vecteurs propres.

## Intégration avec R

Le logiciel R sait faire du calcul numérique d'intégrales à l'aide de la fonction : integrate()

**Exemple 1** : calculer l'intégrale

> f=function(x){exp(-x^2/2)/sqrt(2*pi)} # définir la fonction f.

> integrate(f,lower=-Inf,upper=Inf) #intégrer f.

La fonction integrate() renvoie aussi la précision du calcul numérique effectué , c'est-à-dire une majoration de l'erreur. Si on veut que le résultat, il faut ajouter : $value.

++++++++++++++++++++++++++++++++++++

## Dérivation avec R

R possède des fonctions de dérivation symbolique : D() et deriv().

**Exemple 2** : calcul la dérivée de la fonction :

> D(expression(sin(cos(x+y^2))),"x") #dériver par rapport à x.

-(cos(cos(x + y^2)) * sin(x + y^2))

> D(expression(sin(cos(x+y^2))),"y") #dériver par rapport à y.

-(cos(cos(x + y^2)) * (sin(x + y^2) * (2 * y)))

**Exemple 3** : calcul la dérivée de la fonction :

Il est possible d'effectuer des dérivations numériques en utilisant la fonction : grad() du package numDeriv.

++++++++++++

**Racine d'une fonction**

Pour calculer la racine d'une fonction f (solution de f(x)=0) on utilise la commande : uniroot(). Dans le cas de plusieurs racines (racine d'un polynome) on utilise la commande : polyroot()

Exemple 4 :

>uniroot(f=function(x) (cos(x^2)) ,lower=0,upper=2,tol=0.0001)$root

>polyroot(c(3,-8,1)) #les racines de $3-8x+x^2=0$)

**Master1 : Probabilités et Statistique**

**Module : Initiation au langage R**

# *Statistique descriptive et Probabilités aver R*

## Graphiques avec R

D'abord comment sauvegarder les commandes, les résultats et les graphiques d'une analyse.

***1- Sauvegarder les commandes dans un script*** : pour afficher les dernières commandes que vous avez tapées, taper la commande. **« history() »** , et pour sauvegarder les commandes dans un fichier texte, suivez les étapes : premièrement, en parallèle avec R, ouvrer un éditeur de fichier texte ( par exemple bloc-note) puis copier les commandes qui font l'essentielle de l'analyse dans ce fichier et à la fin de votre session de travail, sauver ce fichier avec un nom explicite et une extension .R .

Quand vous reprendrez cette analyse plu tard, vous pouvez réutiliser ce fichier, qu'on appelle un script. R vous permettra de ré-exécuter les commandes de ce fichier script en utilisant la commande **« source() ».**

**Exemple 1** :(utiliser des donnés du data1)

***2- Sauvegarder les résultats d'une analyse :***

* Les commande et les résultats des analyses statistiques et graphiques peuvent être copiés/collés dans un document.

* Les résultats (sans les commandes) peuvent être copiés automatiquement dans un fichier texte grâce à la commande **« sink () »**

**Exemple 2** : (utiliser des donnés du data1)

***3- Construction de graphiques***

Pour avoir un aperçu des possibilités graphiques du R, on utilise la commande :
**« demo(graphics) «** . Une fois que plus rien ne s'affiche, on tape **« dev.off »**

***a) Créer un graphique :*** Vous pouvez créer un graphique en utilisant une des commandes suivantes :

- **plot**(nom de la variable,type=''l'',xlab= ''nom pour l'axe des x'',ylab=''nom pour l'axe des y'', xlim=c(a,b) ,ylim=c(d,e),main=''titre'')

- **curve**(f(x), a,b) : pour dessiner la courbe de la fonction f sur l'intervalle (a,b).

- **barplot**(x) : pour tracer le diagramme en barres.

- **boxplot**(x) : pour dessiner la boite à moustache.

- **pie**(x) : pour dessiner le diagramme circulaire.

- **hist**(x) : pour construire un histogramme

***b) Avoir plusieurs graphiques sur la même fenêtre*** : Si vous voulez voir lpusiurs graphiques sur la même fenêtre, vous tapez :

- **par**(mfrow=c(k,l)) où k et l sont deux entiers, servant à découper l'écran en k lignes et l colonnes, lorsque plusieurs commandes créant un graphique se succèdent, ces graphiques se positionnent par ligne sur les cas ainsi créées.

- **par**(mfcol=c(k,l)) : pour que les graphiques se positionnent par colonne.

- **par**(ask=TRUE) : affiche plusieurs graphiques successivement.

***c) Ajouter un graphique à un graphique existant :***

- **points**(x,y) : pour ajoiter à la figure existante un nuage de points associée à (x,y).

- **lines**(x,y) : pour ajouter une ligne reliant les points du nuage de points associée à (x,y).

- **text** () : pour ajouter un texte.

- **abline**(h=y) : pour ajouter une ligne horizontale de coordonnée y .

- **abline**(v=x) : pour ajouter une ligne verticale de coordonnée x.

- **abline**(a,b) : pour ajouter la droite d'équation y=a+bx.

-**curve**(f(x),add=TRUE) : ajouter à la fenêtre courante le graphe de la fonction f.

- **title**() : ajouter un titre.

- **legend**() : ajouter une légende.

- **grid**() : pour mettre un quadrillage.

## Statistique descriptive avec R

**TP sous R :** voici quelques commandes utiles :

**- median()** : renvoie la médiane d'un vecteur.
**- mean()** : renvoie la moyenne d'un vecteur.
**- quantile**() : renvoie les quantiles d'un vecteur.
**- summary ()** : appliquée a une série numérique, renvoie minimum, maximum, quartiles et moyenne.
**- range ()** : renvoie le minimum et le maximum.
**- IQR()** : renvoie l'intervalle interquartiles.
**- var ()** : renvoie la variance de l'échantillon.
**- sd ()** : renvoie l'écart type de l'échantillon.

**- cov ()** : renvoie la covariance.

**- cor ()** : renvoie la corrélation.

## Probabilités avec R

Pour une variable aléatoire X suivant une loi notée *loi* dans R, la syntaxe générale est la suivante :

- pour obtenir "la densité" de X, la commande est : *dloi* ; on ajoute la lettre **d** devant loi,

- pour obtenir la fonction de répartition de X, la commande est : *ploi* ; on ajoute la lettre **p** devant loi,

- pour obtenir le quantile de X, la commande est : *qloi* ; on ajoute la lettre **q** devant loi,

- pour simuler des réalisations de var suivant la même loi que X, la commande est : *rloi* ; on ajoute la lettre **r** devant loi.

**Densité :** Si la loi de X dépend d'un ou de plusieurs paramètres, disons par1 et par2, alors la densité de X en x est donnée par les commandes : *dloi*(x, par1, par2)

**Lois discrètes**

— Loi binomiale B(*n, p*) :binom(x,size=n,prob=p). Cette loi a *n* + 1 modalités distinctes : 0*,* 1*, . . . , n*.
— Loi de Bernoulli B(*p*) : binom(x,size=1,prob=p)
— Loi géométrique partant de zéro G(*p*) (nombre d'échecs avant succès) : geom(n, prob=p)
— Loi de Poisson P(_) : pois(x,lambda=lambda)

**Lois continues**

— Loi uniforme U[*a,b*] : unif(x,min=a,max=b)
— Loi normale N(*μ,* _2) : norm(x,mean=mu,sd=sigma)
— Loi exponentielle E(_) : exp(x,rate=lambda)
— Loi du Chi-deux à *r* degrés de liberté _2(*r*) : chisq(x,df=r)
— Loi de Student à *r* degrés de liberté : t(x,df=r)

**Exemples**

>dbinom(4, 8, 0.3 ) # pour calculer la densité de X ∼ B(8, 0.3) en x = 4 .

>dnorm(1.7, 2, 0.12) # pour calculer la densité de X ∼ N (2, 0.122 ) en x = 1.7 .

**Remarque** : Pour calculer la densité en plusieurs valeurs, on prend pour x le vecteur ayant pour éléments ces valeurs. On peut faire de même avec un ensemble de paramètres et les arguments correspondants.

**Exemples**

>dbinom(c(4, 6), 8, 0.3) # pour calculer la densité de X ∼ B(8, 0.3) pour x ∈ {4, 6}.

>dexp(2, c(1, 2, 3)) # pour calculer la densité de X ∼ E(λ) en x = 2, avec λ = 1, λ = 2 et λ = 3.

**Représentation graphique** : On peut représenter le graphe de la densité d'une variable aléatoire X discrète avec la commande : plot et l'option type = h.

On fait :

> plot(0:5, dbinom(0:5, 5, 0.2), type = "h", ylab = "P(X = x)") # représentation graphique de la densité de $X \sim B(5, 0.2)$.

- On peut représenter le graphe de la densité d'une variable X à densité avec la commande curve.

On fait :

> curve(dnorm(x, 5, 1.5), 0.5, 9.5, ylab = "fX(x)") # représentation graphique de la densité de $X \sim N(5, 1.5\ 2)$.

**Fonction de répartition** $F_X$

Si la loi de X dépend d'un ou de plusieurs paramètres, disons par1 et par2, alors la fonction de répartition F de X en x est donnée par les commandes :

ploi(x, par1, par2)

On peut calculer : $P(X > x) = 1 - F_X(x)$, en faisant :

ploi(x, par1, par2, lower.tail = FALSE)

**Exemples**

>pbinom(4, 8, 0.3) # pour calculer la fonction de répartition de $X \sim B(8, 0.3)$ en x = 4.

> pnorm(12, 9, 2)# pour calculer la fonction de répartition de $X \sim N(9, 2^2)$ en x = 12.

>pexp(2, 3, lower.tail = FALSE)# pour calculer : $P(X > 2)$, $X \sim E(3)$ .

**Représentation graphique** On peut représenter le graphe de la fonction de répartition d'une variable X discrète avec la commande : stepfun.

> plot(stepfun(0:15, c(0, pbinom(0:15, 15, 0.6))), ylab = "FX(x)", main = "")# représentation graphique de la fonction de répartition de $X \sim B(15, 0.6)$.

On peut représenter le graphe de la fonction de répartition d'une var X à densité avec la commande curve.

> curve(pnorm(x, 5, 1.5), 0.5, 9.5, ylab = "FX(x)")# représentation graphique de la fonction de répartition de $X \sim N(5, 1.5^2)$.

**Master1 : Probabilités et Statistique**

**Module : Initiation au langage R**

## Estimation ponctuelle et par intervalle de confiance avec R

**1- Notions de base**

**1-1- Population statistique et individus (unité statistique**) : Une population est un ensemble d'objets sur lesquels une étude se porte. Ces objets sont appelés individus ou unité statistique.

**1-2- Caractère ou variable statistique** : Toute propriété étudiée chez les individus d'une population est appelée caractère.

Un caractère est dit :

- **quantitatif** s'il mesure une quantité ou un nombre ; par exemple : le nombre de personnes dans une salle, le temps de réalisation d'un travail en heures. . .
- **qualitatif ou catégoriel** s'il mesure une catégorie (la couleur des yeux, la marque du téléphone portable d'un étudiant.

Les valeurs sont appelées **modalités**.

**1-3- Échantillon :** Un échantillon est un sous ensemble de la population statistique.

**1-4-Données :** Les données sont les observations de caractères sur les individus d'un échantillon.

**2- Estimation paramétrique (estimation ponctuelle)**

Estimer un paramètre inconnu c'est chercher une valeur approchée de la valeur exacte de ce paramètre.

Supposons que le caractère étudier est X , sur la population la loi de X est de moyenne $\mu$ et de variance $\sigma^2$ inconnus. De cette population on extrait un échantillon de taille n (le nombre d'individus d'échantillon) et $x_1$ , $x_2$,…,$x_k$ sont les données de cette échantillon, on définit

a) La moyenne des $x_1$ , $x_2$,…,$x_k$ par

C'est une estimation ponctuelle de la valeur moyenne $\mu$ de X.

b) La variance corrigée des $x_1$ , $x_2$,…,$x_k$ par

c) Soit P la proportion théorique d'individus présentant le caractère A dans la population. La proportion observée dans un ´échantillon de taille n est notée f. f est une estimation ponctuelle de P.

**3- Estimation par intervalle de confiance**

Soit X une variable aléatoire continue de moyenne $\mu$ et de variance $\sigma^2$. Soit n observations de X. La moyenne de cet ´échantillon $\bar{x}$ est distribuée autour de $\mu$.

L'intervalle de confiance de $\mu$ est donné, quand n est supérieur à 30, par

dont Z est la variable aléatoire suivant la loi normale centrée réduite et $\alpha$ est le risque que $\mu$

n'appartienne pas à cet intervalle. Pour n petit, inferieur à 30, on remplace Z par T la loi de Student.

Sous R la commande qui permet d'obtenir l'intervalle de confiance de la moyenne $\mu$ est
`t.test(x)$conf`

L'intervalle de confiance de $\sigma^2$ est donné par

L'intervalle de confiance de P est donné par

Sous R la commande qui permet d'obtenir l'intervalle de confiance de la proportion P est :
`prop.test(n,N)$conf`

**Les principaux tests d'hypothèses en R**

**Objectif :** Il s'agit de faire le choix entre deux hypothèses, $H_0$ l'hypothèse nulle et $H_1$ l'hypothèse alternative

- t.test(x,...) test de Student sur l'espérance d'une loi normale
- binom.test() test sur une proportion
- var.test(x,y,...) test de Fisher sur la variance de 2 ´echantillons gaussiens indépendants

t.test(x,y,...) test de Student sur l'esp´erance de 2 ´echantillons gaussiens ind´ependants
prop.test() test de comparaison de proportions
chisq.test(x,...) test du X2 sur les probabilités d''ev`enements et tables de contingence