INTRODUCTION A R

Hervé Morin

Université Laval

Opérations élémentaires, vecteurs, matrices

Pour quitter le logiciel

q()	termine la session

Opérations élémentaires		**Exemples**		
x+y	addition	35+7	1.26+3.46	
x-y	soustraction	25-14	0.6789-1.3805	
x*y	multiplication	6*5	1.57*2.5432	
x/y	division	10/7	1543/217.89	
x^y	puissance	2^3	3.67^-1.639	3*2-1/10

Fonctions élémentaires			
sqrt(x)	racine	sqrt(6.8035)	
log(x)	logarithme népérien	log(1294.56)	
exp(x)	exponentielle	exp(0.6752)	
abs(x)	valeur absolue	abs(0.6789-1.3805)	
gamma(n+1)	factorielle n	gamma(7)	
choose(n,k)	combinaison de k parmi n	choose(5,2)	choose(49,6)

Définition d'un objet			
x<-y	x "est donné par" y	x<-sqrt(3*2-1/10)	y<-10/7
x_y	"	z<-x*log(y)	z_x*log(y)

Création de vecteurs					
x<-a:b	vecteur des entiers de "a" à "b"	x<-1:10	x<--3:4	x<-25:3	x<-1:10-1
x<-c(a,b,c,...)	vecteur des données "a", "b", "c",....	x<-c(2,5,6.3,0,-2.8)			
x<-rnorm(n)	vecteur de n données aléatoires normales centrées réduites	x<-rnorm(50)			
x<-rep(x,n)	x est répété n fois	z<-1:5	t<-rep(z,3)		

Opérations arithmétiques sur les vecteurs

+ - * / ^ sqrt() log() exp() abs()	x<-1:6	y<-x-3
	z<-2*y/3	t<-sqrt(abs(z)^3)

Opérations logiques sur les vecteurs

		x<-1:6	y<-c(1,4,2,5,4,3)
x<y	x "inférieur à" y	x<y	
x>y	x "supérieur à" y	x>y	
x<=y	x "inférieur ou égal à" y	x<=y	
x>=y	x "supérieur ou égal à" y	x>=y	
x==y	x "égal à" y		x==y
x!=y	x "différent de " y	x!=y	
x&y	intersection	(x<=3)&(y>3)	
x|y	union	(x<=3)|(y>3)	

Autres fonctions utilisables sur les vecteurs

		x<-rnorm(50)	
max(x)	maximum de x	max(x)	
min(x)	minimum de x	min(x)	
round(x,n)	arrondi les éléments de x avec n chiffres après la virgule	y<-(round(x,2))*100	
length(x)	nombre d'éléments de x	length(x)	
sum(x)	somme des éléments de x	sum(x)	sum(y)/length(y)
prod(x)	produits des éléments de x	prod(x*2)	prod(1:6)
sort(x)	classe les éléments de x par ordre croissant		sort(y)
rank(x)	rangs des éléments de x	rank(y)	

Extraction d'éléments

x[i]	i^e élément de x	y[5]	y[3,6,48]	y[10:20]
x[i]<-a	remplace le i^e élément de x par a	y[2]<-0		

Vecteurs de caractères

c("X","Y","Z")	vecteur de caractères	z<-c("Hervé","Morin")
z<-paste(...,sep="")	crée une séquence de caractères	z<-paste("X",1:10,sep="")

Création de matrices

matrix(x,ncol=c)	matrice de c colonnes déployant le vecteur x	x<-1:12 Z<-matrix(x,ncol=3)
cbind(x,y)	matrice combinant par colonnes les éléments x et y	Z<-cbind(1:10,rnorm(10))
rbind(x,y)	matrice combinant par lignes les éléments x et y	Z<-rbind(1:5,rnorm(5))
colnames(...)	identifie les colonnes	colnames(Z)<-paste("X",1:5,sep="")
rownames(...)	identifie les lignes	rownames(Z)<-c("Hommes","Femmes")
c(Z)	transforme une matrice en un vecteur	c(Z)
Z[r,]	vecteur de la ligne r	Z[2,]
Z[,c]	vecteur de la colonne c	Z[,3]
Z[i,j]	élément de la ligne i et de la colonne j	Z[2,3]

Opération sur les matrices

t(Z)	matrice transposée	t(Z)
dim(Z)	dimensions (lignes, colonnes)	dim(Z)
Z%*%Y	multiplication	Y<-matrix(1:6,ncol=2) Z%*%Y
solve(X)	matrice inverse	solve(matrix(c(2,2,3,1,7,4,5,5,5),ncol=3))

Statistique descriptive

Caractéristiques de l'échantillon

		y<-(round(3+rnorm(50),2))*100
mean(x)	moyenne du vecteur x	mean(y)
var(x)	variance du vecteur x	var(y)
median(x)	médiane du vecteur x	median(y)
quantile(x)	minimum, Q_1, médiane, Q_2, maximum	quantile(y)
sd(x)	écart-type	sd(y)

Graphiques

		Z<-cbind(1:10,(3+round(rnorm(10),2))*100)
barplot(x)	diagramme en tuyaux d'orgue	barplot(Z[,2])
Mac68k: barchart(x)		y<-round(17.4+rnorm(50),1)*10
dotplot(x)	diagramme par points	dotplot(y)
stem(x)	diagramme en tiges et feuilles	stem(y)
stem(x,2)	double le nombre de tiges	stem(y,2)
stem(x,.5)	divise par 2 le nombre de tiges	stem(y,.5)
hist(x)	histogramme des données de x	hist(y)
hist(x,n)	histogramme avec environ n colonnes	hist(y,5)
boxplot(x)	diagramme en boites	boxplot(y,horiz=T)
par(mfrow=c(m,n))	fenêtre graphique divisée en m x n cellules	
Mac68k: layout(m,n)		par(mfrow=c(2,1))
		hist(y)
		boxplot(y)
		par(mfrow=c(1,1))
		x<-round(13+rnorm(50),1)*10
plot(x,y)	grahique des points de coordonnées (x,y)	plot(x,y)
print(x)	imprime le vecteur x	print(-10:10)

Lois de probabilité usuelles

Loi binomiale

dbinom(x,n,p)	probabilité de x succès en n expériences de Bernoulli p est la probabilité de succès à chaque expérience	dbinom(12,20,0.7)
pbinom(x,n,p)	probabilité d'au plus x succès	pbinom(17,20,0.7)
qbinom(q,n,p)	nombre de succès pour avoir une probabilité d'au moins q	qbinom(.95,20,0.7)
rbinom(m,n,p)	m observations aléatoires de la binomiale (n,p)	rbinom(20,4,.4756)

Loi de Poisson

dpois(x,λ)	probabilité de x succès dans une Poisson de paramètre λ	dpois(2,1.3)
ppois(x,λ)	probabilité d'au plus x succès	ppois(2,1.3)
qpois(q,λ)	nombre de succès pour avoir une probabilité d'au moins q	qpois(.95,1.3)
rpois(m,λ)	m observations aléatoires de la Poisson (λ)	rpois(20,1.3)

Loi hypergéométrique

dhyper(x,N1,N2,n)	probabilité de x succès dans un échantillon de taille n tiré d'une population avec N1 succès et N2 échecs	dhyper(8,18,12,10)
phyper(x,N1,N2,n)	probabilité d'au plus x succès	phyper(8,18,12,10)
qhyper(q,N1,N2,n)	nombre de succès pour avoir une probabilité d'au moins q	qhyper(.95,18,12,10)
rhyper(m,N1,N2,n)	m observations aléatoires	rhyper(20,18,12,10)

Loi uniforme

punif(x,a,b)	Pr(X≤x) dans une uniforme [a,b]	punif(2.3,1.5,7.8)
qunif(q,a,b)	valeur x telle que Pr(X≤x)=q	qunif(.95,1.5,7.8)
runif(m,a,b)	m observations aléatoires d'une uniforme [a,b]	y<-runif(1000,2,8) hist(y)

Loi exponentielle

pexp(x,λ)	Pr(X≤x) dans une exponnentielle de paramètre λ	pexp(3,.2)
qexp(q,λ)	valeur x telle que Pr(X≤x)=q	qexp(.95,.2)
rexp(m,λ)	m observations aléatoires d'une exponnentielle (λ)	y<-rexp(1000,.2) hist(y)

Mac68k: 1/λ au lieu de λ

Loi normale

pnorm(z)	Pr(Z≤z) dans une loi normale centrée et réduite	pnorm(1,96)
pnorm(x,μ,σ)	Pr(X≤x) dans une normale de moyenne μ et d'écart-type σ	pnorm(180,174,25)
qnorm(q,μ,σ)	valeur x telle que Pr(X≤x)=q	qnorm(.95,174,25)
rnorm(m,μ,σ)	m observations aléatoires d'une normale (μ,σ)	y<-rnorm(1000,10,2)hist(y)

Échantillonnage

Loi de Student

pt(t,r)	Pr(T≤t) dans une loi de Student avec r degrés de liberté	pt(0.860,20)
qt(q,r)	valeur t telle que Pr(T≤t)=q	qt(0.8,20)

Loi de khi2

pchisq(u,r)	Pr(U≤u) dans une loi de khi2 avec r degrés de liberté	pchisq(70.22,49)
qchisq(q,r)	valeur u telle que Pr(U≤u)=q	qchisq(0.975,49)

Loi F de Fisher

pf(f,r_1,r_2)	Pr(F≤f) dans une loi F avec r_1 et r_2 degrés de liberté	pf(3.13,14,20)
qf(q,r_1,r_2)	valeur f telle que Pr(F≤f)=q	qf(0.99,14,20)

Tirage aléatoire d'un échantillon

sample(x,n)	éch. de taille n tiré sans remise de x	sample(1:1000,50)
sample(x,n,replace=T)	éch. de taille n tiré avec remise de x	sample(1:10,50,replace=T)

```
x<-rnorm(200,174,25)
ech<-sample(x,50)
mean(ech)
sd(ech)
```

Création d'une fonction

Création de la fonction "fun"

fun<-function(a,b,..) création de la fonction: fun(a,b,..)

```
moyenne<-function(x)
        {
        y<-sum(x)
        n<-length(x)
        y/n
        }
```

Modification de la fonction

fun<-edit(fun) modification de la fonction "fun"
fun1<-edit(fun) création d'une autre fonction "fun1"

Utilisation d'une condition

if (condition) {expression1} else {expression2}
si la condition est remplie, l'expression1 est exécutée
sinon l'expression2 est exécutée

```
if (n<41) {d<-qt(0.95,n-1)}
else {d<-qnorm(0.95)}
```

Boucles

for (i in 1:m) {expression} l'expression est exécutée m fois

```
for(i in 1:5) print(1:i)
```

```
simul<-function (m, n, x)
{
moyeas <- rep(0, m)
for (i in 1:m)
        {
        ech <- sample(x, n)
        moyeas[i] <- mean(ech)
        }
moyeas
}

x<-rnorm(200,174,25)
simul(10,50,x)
```

Pour imprimer un texte avec des résultats

cat("texte...", expression, "\n")
imprime "texte...",
calcule et imprime l'expression
"\n" permet de changer de ligne

```
cat("Moyenne: ", mean(x), "\n")
```

Intervalles de confiance

Intervalle de confiance pour la moyenne d'une loi normale

Création de la fonction "icmoy":

```
icmoy<-function (x, alpha = 0.05)
{
d <- qt((1 - alpha/2), length(x) - 1) * sd(x)/sqrt(length(x))
ic <- c(mean(x) - d, mean(x) + d)
ic
}
```

```
x<-rnorm(30,174,25)
icmoy(x)
icmoy(x,0.10)

x<-rnorm(200,174,25)
icmoy(x)
icmoy(x,0.10)
```

Intervalle de confiance pour la variance d'une loi normale

```
icvar<-function (x, alpha = 0.05)
{
c1 <- (length(x) - 1) * var(x)/qchisq((1 - alpha/2), length(x) - 1)
c2 <- (length(x) - 1) * var(x)/qchisq((alpha/2), length(x) - 1)
ic <- c(c1, c2)
ic
}
```

```
x<-rnorm(30,174,25)
icvar(x)
icvar(x,0.10)

x<-rnorm(200,174,25)
icvar(x)
icvar(x,0.10)
```

Intervalle de confiance pour une proportion

```
icprop<-function (m, n, alpha = 0.05)
{
p <- m/n
d <- qt((1 - alpha/2), n - 1) * sqrt(p * (1 - p)/n)
ic <- c(p - d, p + d)
ic
}
```

```
icprop(25,35)
icprop(25,35,0.10)
icprop(50,70)
icprop(50,70,0.10)
icprop(500,700)
icprop(5000,7000)
icprop(50000,70000)
icprop(50000,70000,0.25)
```

Tests statistiques

Test concernant la moyenne d'une loi normale

```
testmoy<-function (x, mu.zero)
{
t <- abs((mean(x) - mu.zero)/(sd(x)/sqrt(length(x))))
seuil <- 2 * (1 - pt(t, dl<-(length(x) - 1)))
cat("\n")
cat("Moyenne: ", mean(x), "\n")
cat("Écart-type: ", sd(x), "\n")
cat("\n")
cat("Valeur de t: ",t," avec", dl, "degrés de liberté","\n")
cat("\n")
cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
}
```

```
x<-rnorm(200,174,25)
testmoy(x,176)
```

Test concernant la variance d'une loi normale

```
testvar<-function (x, sig2.zero)
{
u <- (length(x) - 1) * var(x)/sig2.zero
if (var(x) < sig2.zero)
        {
        seuil <- 2 * pchisq(u, dl<-(length(x) - 1))
        }
else
        {
        seuil <- 2 * (1 - pchisq(u, dl<-(length(x) - 1)))
        }
        cat("\n")
        cat("Variance: ", var(x), "\n")
        cat("\n")
        cat("Valeur de U: ",u," avec", dl, "degrés de liberté","\n")
        cat("\n")
        cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
        cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
}
```

```
x<-rnorm(200,174,25)
testvar(x,550)
```

Test concernant une proportion

```
testprop<-function (m, n, pi.zero)
{
z <- abs((m/n - pi.zero)/sqrt(pi.zero * (1 - pi.zero)/n))
seuil <- 2 * (1 - pnorm(z))
cat("\n")
cat("Proportion: ", m/n, "\n")
cat("\n")
cat("Valeur de Z: ", round(z,2), "\n")
cat("\n")
cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
}
```

```
testprop(87,1069,0.10)
```

Test de l’égalité de deux moyennes pour 2 échantillons indépendants

• Test de l’égalité des variances

```
eg.var<-function (x, y, alpha = 0.05)
{
nx <- length(x)
ny <- length(y)
f <- var(x)/var(y)
if (f > 1)
{
seuilf <- 2 * (1 - pf(f, nx - 1, ny - 1))
}
else
{
seuilf <- 2 * (1 - pf(1/f, ny - 1, nx - 1))
}
cat("\n")
cat("Test de l'égalité des variances", "\n")
cat("\n")
cat("Variance échantillon 1: ", var(x), "\n")
cat("Variance échantillon 2: ", var(y), "\n")
cat("\n")
cat("\n")
cat("Valeur du F: ", round(max(f, 1/f), 2), "\n")
cat("\n")
cat("Seuil observé: ", round((seuilf) * 100, 2), "%", "\n")
}
```

```
x<-rnorm(50,174,25)
y<-rnorm(30,185,30)
eg.var(x,y)
```

• Test de l’égalité des moyennes si les variances sont égales

```
eg.moy<-function (x, y, alpha = 0.05)
{
nx <- length(x)
ny <- length(y)
sp <- sqrt(((nx - 1) * var(x) + (ny - 1) * var(y))/(nx + ny - 2))
t <- abs((mean(x) - mean(y))/(sp * sqrt(1/nx + 1/ny)))
dl <- nx + ny - 2
seuil <- 2 * (1 - pt(t, dl))

cat("\n", "\n", "Test de Student pour l’égalité des moyennes", "\n", "\n")
cat("\n")
cat("Moyenne échantillon 1 de taille", nx, ": ", mean(x), "\n")
cat("Moyenne échantillon 2 de taille", ny, ": ", mean(y), "\n")
cat("\n")
cat("Valeur de t: ", round(t, 2), "avec", dl, "degrés de liberté", "\n")
cat("\n")
cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
cat("\n")
}
```

```
eg.moy(x,y)
```

• Test de l'egal te des moyennes s les var ances sont negales

```
welch<-function (x, y, alpha = 0.05)                                    welch(x,y)
{
        nx <- length(x)
        ny <- length(y)
        sdex2 <- var(x)/nx
        sdey2 <- var(y)/ny
        dl <- ((sdex2 + sdey2)^2)/((sdex2^2)/(nx - 1) + (sdey2^2)/(ny - 1))
        t <- abs((mean(x) - mean(y))/(sqrt(sdex2 + sdey2)))
        seuil <- 2 * (1 - pt(t, dl))
        cat("\n", "\n", "Test de Welch", "\n", "\n")
        cat("\n")
        cat("Moyenne échantillon 1 de taille", nx, ": ", mean(x), "\n")
        cat("Moyenne échantillon 2 de taille", ny, ": ", mean(y), "\n")
        cat("\n")
        cat("Valeur de t: ", round(t, 2), "avec", dl, "degrés de liberté", "\n")
        cat("\n")
        cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
        cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
}
```

Test de l'égalité de deux moyennes pour 2 échantillons appariés

```
eg.moy.p<-function (x, y)                                    x<-c(210,217,208,215,202,209,207,210,221,218)
                                                             y<-c(212,210,210,213,200,208,203,199,218,214)
{                                                            eg.moy.p(x,y)
d <- x - y
t <- abs(mean(d)/sqrt(var(d)/length(d)))
seuil <- 2 * (1 - pt(t, length(d) - 1))
cat("\n")
cat("Test de l'égalité des moyennes", "\n")
cat("\n")
cat("Moyenne des",length(d), "différences: ", mean(d),"\n")
cat("\n")
cat("Valeur de t:",round(t,2), "avec", length(d) - 1, "degrés de liberté", "\n")
cat("\n")
cat("Valeur de t: ", t, "avec", length(d) - 1, "degrés de liberté", "\n")
cat("\n")
cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
}
```

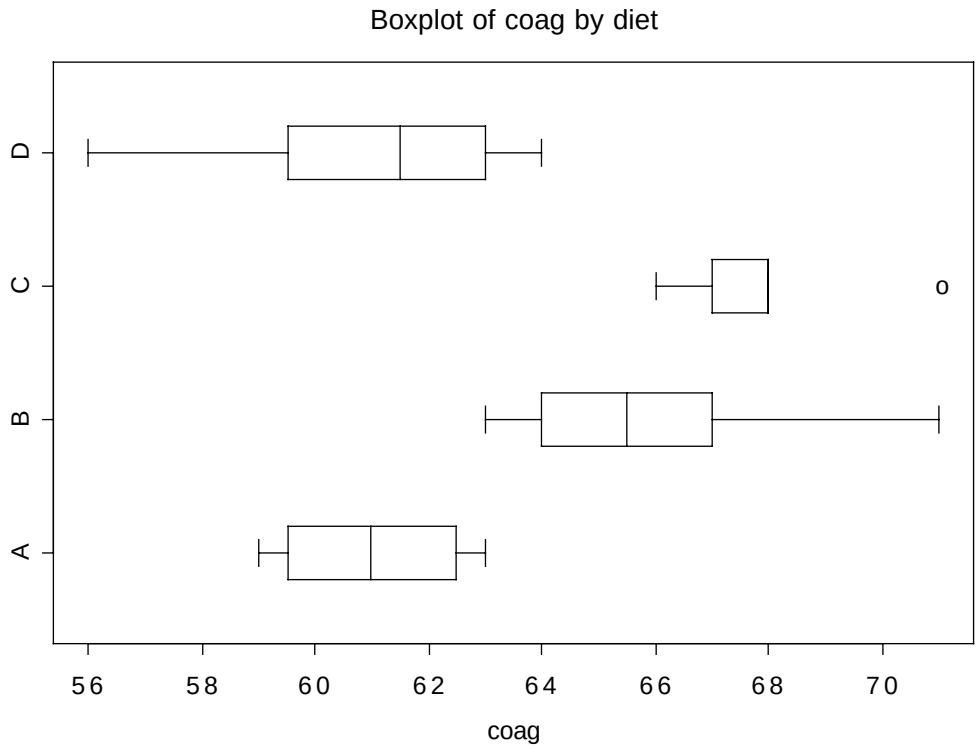
Test de l'égalité de deux proportions

```
eg.prop<-function (m1, n1, m2, n2)                                    eg.prop(211,725,361,1095)
{
p0 <- (m1 + m2)/(n1 + n2)
z <- abs((m1/n1 - m2/n2)/sqrt(p0 * (1 - p0) * (1/n1 + 1/n2)))
seuil <- 2 * (1 - pnorm(z))
cat("\n")
cat("Valeur de Z: ", z, "\n")
cat("\n")
cat("Seuil observé, test bilatéral: ", round(seuil * 100, 2), "%", "\n")
cat("Seuil observé, test unilatéral: ", round((seuil/2) * 100, 2), "%", "\n")
cat("\n")
}
```

Analyse de variance à un facteur

Version Mac68k

```
coag<-c(62,60,63,59,63,67,71,64,65,66,68,66,71,67,68,68,56,62,60,61,63,64,63,59)
diet<-rep(c("A","B","C","D"),c(4,6,6,8))
boxplot(coag,group=diet)
```



> anova.table(coag,diet)

	Deg freedom	Sum Squares	Mean Square	F-statistic	p-value
Between Groups	3	228	76	13.57143	0
Within Groups	20	112	5.6		
Total	23	340			

> levene.test(coag,diet)

	Deg freedom	Sum Squares	Mean Square	F-statistic	p-value
Between Groups	3	4.333333	1.444444	0.6491885	0.5926
Within Groups	20	44.5	2.225		
Total	23	48.83333			

Contrastes

```
contrasts<-function (y, gr, ci)
{
ni <- by.group(y, gr, length)
si2 <- by.group(y, gr, var)
t <- length(ni)
n <- sum(ni)
mse <- sum((ni - 1) * si2)/(n - t)
ts <- abs(sum(ci * by.group(y, gr, mean))/sqrt(mse * sum((ci^2)/ni)))
cat("\n")
cat("Valeur du t: ", round(ts, 2), "\n")
cat("\n")
cat("Seuil observé: ", round(2 * (1 - pt(ts, n - t)) * 100, 2), "%", "\n")
cat("\n")
}
```

```
data1<-c(8,6,7,5,4,14,11,10,12,13,6,7,9,10,8,3,5,2,4,6)
trait1<-rep(c("T1","T2","T3","T4"),c(5,5,5,5))
>c1<-c(-1,-1,-1,3)
```

```
> contrasts(data1,trait1,c1)

Valeur du t:  5.72
Seuil observé:  0 %
```

Test de Kruskal-Wallis

```
kruskal.test<-function (y, gr)
{
n <- sum(grp.sizes <- table(gr))
kw <- (12/(n * (n + 1))) * sum(by.group(rank(y), gr, sum)^2/grp.sizes) - 3 * (n + 1)
t <- length(by.group(y, gr, length))
cat("\n")
cat("Valeur de KW: ", round(kw, 2), "\n")
cat("\n")
cat("Seuil observé: ", round((1 - pchisq(kw, t - 1)) * 100, 2), "%", "\n")
cat("\n")
}
```

```
poids<-c(100,140,165,160,220,180,95,165,140,55,175,140,160,175,115,120,105,80,200,210,
        160,185,190,195,175)
porcs<-rep(c("P1","P2","P3"),c(10,8,7))
```

```
> kruskal.test(poids,porcs)

Valeur de KW:  8.68
Seuil observé:  1.3 %
```

Analyse de variance à un facteur

Version PC et PPC

```
coag<-c(62,60,63,59,63,67,71,64,65,66,68,66,71,67,68,68,56,62,60,61,63,64,63,59)
diet<-rep(c("A","B","C","D"),c(4,6,6,8))
diet<-factor(diet)
anova(lm(coag~diet))
```

Analysis of Variance Table

	Df	Sum-Sq	Mean-Sq	F	Pr(>F)
diet	3	228	76.0	13.57	4.658e-05
Residual	20	112	5.6		

Test de Levene pour l'égalité des variances

```
levene.test<-function (y, gr)
{
mi <- tapply(y, gr, median)
z <- abs(y - mi[gr])
anova(lm(z~gr))
}
> levene.test(coag,diet)
```

	Deg freedom	Sum Squares	Mean Square	F-statistic	p-value
Between Groups	3	4.333333	1.444444	0.6491885	0.5926
Within Groups	20	44.5	2.225		
Total	23	48.83333			

Contrastes

```
contrasts<-function (y, gr, ci)
{
      ni <- tapply(y, gr, length)
      si2 <- tapply(y, gr, var)
      t <- length(ni)
      n <- sum(ni)
      mse <- sum((ni - 1) * si2)/(n - t)
      ts <- abs(sum(ci * tapply(y, gr, mean))/sqrt(mse * sum((ci^2)/ni)))
      cat("\n")
      cat("Valeur du t: ", round(ts, 2), "\n")
      cat("\n")
      cat("Seuil observé: ", round(2 * (1 - pt(ts, n - t)) *100, 2), "%", "\n")
      cat("\n")
}
```

```
data1<-c(8,6,7,5,4,14,11,10,12,13,6,7,9,10,8,3,5,2,4,6)
trait1<-rep(c("T1","T2","T3","T4"),c(5,5,5,5))
>c1<-c(-1,-1,-1,3)
```

```
> contrasts(data1,trait1,c1)

Valeur du t:  5.72
Seuil observé:  0 %
```

Test de Kruskal-Wallis

```
kruskal.test<-function (y, gr)
{
n <- sum(grp.sizes <- table(gr))
kw <- (12/(n * (n + 1))) * sum(tapply(rank(y), gr, sum)^2/grp.sizes) - 3 * (n + 1)
t <- length(tapply(y, gr, length))
cat("\n")
cat("Valeur de KW: ", round(kw, 2), "\n")
cat("\n")
cat("Seuil observé: ", round((1 - pchisq(kw, t - 1)) * 100, 2), "%", "\n")
cat("\n")
}
```

```
> kruskal.test(poids,porcs)


Valeur de KW:  8.68
Seuil observé:  1.3 %
```

Analyse de variance à deux facteurs

Version Mac68k

```
> insecticides<-c(2.05,1.56,1.68,1.69,1.25,1.73,1.82,1.31,1.95,2.00,1.83,1.81)
> trait2<-rep(c("T1","T2","T3"),c(4,4,4))
> blocs2<-rep(c("B1","B2","B3","B4"),3)

> levene.test(insecticides,trait2)
```

	Deg freedom	Sum Squares	Mean Square	F-statistic	p-value
Between Groups	2	0.06155	0.030775	2.941848	0.104
Within Groups	9	0.09415	0.01046111		
Total	11	0.1557			

```
> anova.table(insecticides,blocs2,trait2)
```

	Deg freedom	Sum Squares	Mean Square	F-statistic	p-value
blocs2	3	0.05866667	0.01955556	0.3331598	0.8023
trait2	2	0.2766167	0.1383083	2.356301	0.1757
Residual	6	0.3521833	0.05869722		
Total	11	0.6874667	0.01955556		

Analyse de variance à deux facteurs

Version PC et PPC

```
> insecticides<-c(2.05,1.56,1.68,1.69,1.25,1.73,1.82,1.31,1.95,2.00,1.83,1.81)
> trait2<-rep(c("T1","T2","T3"),c(4,4,4))
> blocs2<-rep(c("B1","B2","B3","B4"),3)
> trait2<-factor(trait2)
> blocs2<-factor(blocs2)

> levene.test(insecticides,trait2)
```

	Df	Sum Sq	Mean Sq	F	Pr(>F)
insecticides	2	0.06155	0.030775	2.941848	0.104
Residual	9	0.09415	0.01046111		

```
> anova(lm(insecticides~blocs2+trait2))
```

Analysis of Variance Table

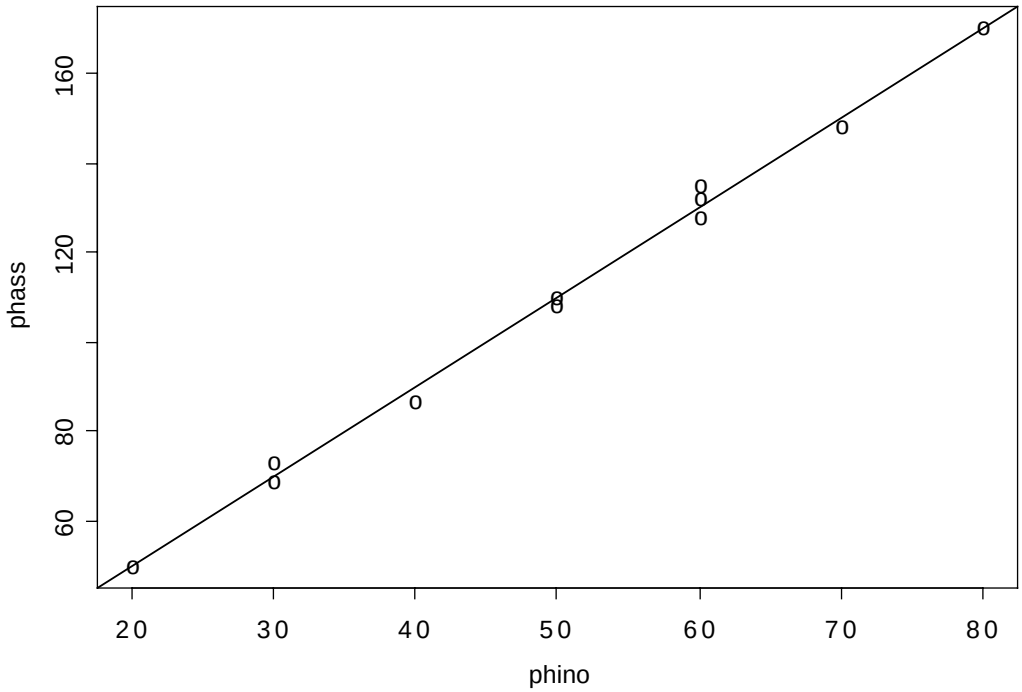
	Df	Sum Sq	Mean Sq	F	Pr(>F)
blocs2	3	0.05867	0.01956	0.3332	0.8023
trait2	2	0.27662	0.13831	2.3563	0.1757
Residual	6	0.35218	0.05870		

Régression simple

Version Mac68k

```
> phino<-c(30,20,60,80,40,50,60,30,70,60,50)
> phass<-c(73,50,128,170,87,108,135,69,148,132,110)

> plot(phino,phass)
> fit.line(phino,phass,draw.plot=T)
```

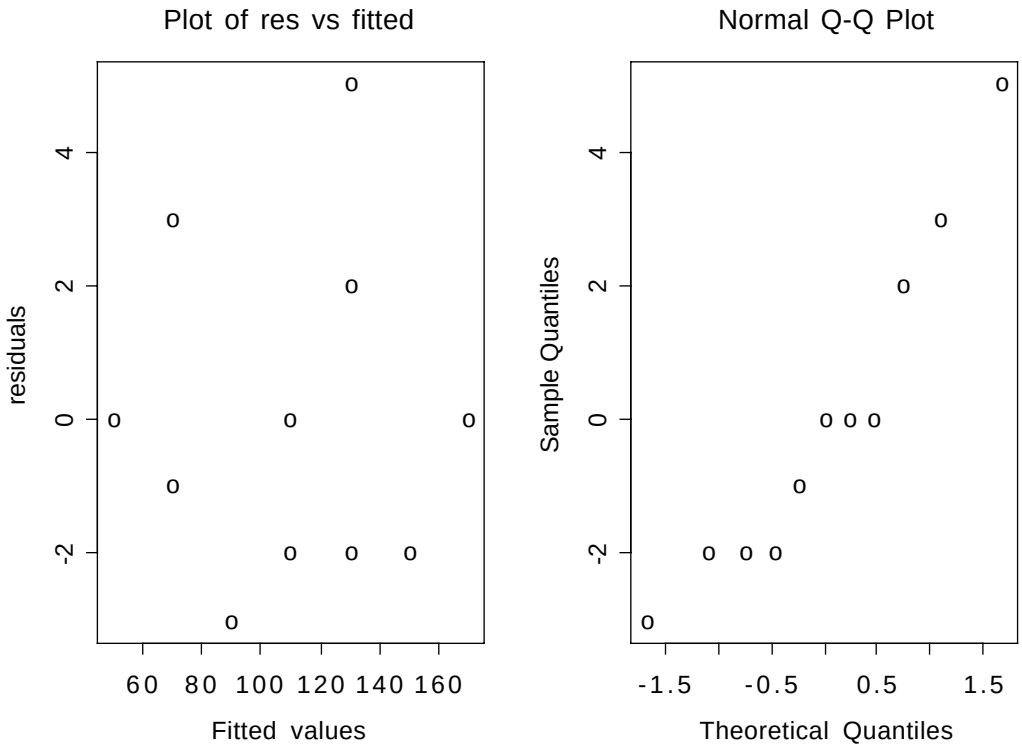



```
> regress(phino,phass)
```

	Coef	Std Err	t-value	p-value	C.I.lower	C.I.upper
Intercept	10	2.34691742	4.260908	2.1083e-03	4.690904	15.30910
phino	2	0.04428074	45.166359	6.3931e-12	1.899830	2.10017

Percent of variation explained: 99.56
Estimate of error Std dev: 2.581989
Error df: 9

	Deg-freedom	Sum-squares	Mean-square	F-statistic	p-value
Regression	1	13600	13600	2040	0
Residual	9	60	6.666667		
Total	10	13660			



> predict(regress(phino,phass),65)

	Predicted	Conf-lower	Conf-upper	Pred-lower	Pred-upper
P1	140	137.6850	142.3150	133.7171	146.2829

Régression simple

Version PC et PPC

```
> phino<-c(30,20,60,80,40,50,60,30,70,60,50)
> phass<-c(73,50,128,170,87,108,135,69,148,132,110)

> reg<-lm(phass~phino)
> summary(reg)
```

Call:
lm(formula = phass ~ phino)

Residuals:

Min	1Q	Median	3Q	Max
-3.000e+00	-2.000e+00	-2.753e-14	1.000e+00	5.000e+00

Coefficients:
Coefficients:

	Estimate	Std-Error	t-Value	Pr(>|t|)
(Intercept)	10	2.3469	4.2609	0.0021
phino	2	0.0443	45.1664	0.0000

Residual standard error: 2.582 on 9 degrees of freedom
Multiple R-Squared: 0.9956, Adjusted R-squared: 0.9951
F-statistic: 2040 on 1 and 9 degrees of freedom, p-value: 6.393e-12

```
> predict(reg,data.frame(x=c(65,100)))
```

	predictor	conf.l	conf.u	pred.l	pred.u
[1,]	140	137.6850	142.3150	133.7171	146.2829
[2,]	210	204.6909	215.3091	202.1068	217.8932

```
> plot(fitted(reg),residuals(reg))
> qqnorm(residuals(reg))
> qqline(residuals(reg))
```

Régression multiple

> phorg<-c(16,10,32,40,20,28,30,16,35,30,29)

Version Mac68k

> regress(phino,phorg,phass)

	Coef	Std-Err	t-value	p-value	C.I.lower	C.I.upper
Intercept	10.5207101	2.5420344	4.1386970	0.00325921842	4.658768	16.382652
phino	2.1988166	0.2987397	7.3603101	0.00007915061	1.509922	2.887711
phorg	-0.4023669	0.5974793	-0.6734406	0.51964479382	-1.780157	0.975423

Percent of variation explained: 99.58
Estimate of error Std dev: 2.664138
Error df: 8

	Deg freedom	Sum squares	Mean square	F-statistic	p-value
Regression	2	13603.22	6801.61	958.2926	0
Residual	8	56.78107	7.097633		
Total	10	13660			

Version PC et PPC

> reg<-lm(phass~phino+phorg)
> summary(reg)

Call:
lm(formula = phass ~ phino + phorg)

Residuals:

Min	1Q	Median	3Q	Max
-3.4260	-1.3846	-0.4734	1.4142	4.6213

Coefficients:

	Estimate	Std-Error	t-Value	Pr(>|t|)
(Intercept)	10.5207	2.5420	4.1387	0.0033
phino	2.1988	0.2987	7.3603	0.0001
phorg	-0.4024	0.5975	-0.6734	0.5196

Residual standard error: 2.664 on 8 degrees of freedom
Multiple R-Squared: 0.9958, Adjusted R-squared: 0.9948
F-statistic: 958.3 on 2 and 8 degrees of freedom, p-value: 2.985e-10

Corrélation

```
> ans<-c(25,36,22,25,48,39,42,31,28,23)
> dommage<-c(55,60,50,30,75,70,70,55,30,35)
```

Coefficient de corrélation de Pearson

```
> cor(ans,dommage)
[1] 0.818533
```

Coefficient de corrélation de Spearman

```
> cor(rank(ans),rank(dommage))
[1] 0.8098312
```

Test d'un coefficient de corrélation nul

```
cor.test<-function(x,y,pearson=T)
{
if(pearson)
{r<-cor(x,y)}
else{r<-cor(rank(x),rank(y))}
t<-abs(r*sqrt((length(x)-2)/(1-r^2)))
seuil <- 2 * (1 - pt(t, length(x) - 2))
cat("\n")
cat("Coefficient de corrélation: ", r, "\n")
cat("\n")
cat("Valeur de t: ", round(t,2), "\n")
cat("\n")
cat("Seuil observé: ", round(seuil * 100, 2), "%", "\n")
}
```

```
cor.test(x,y)
cor.test(x,y,pearson=F)
```

Tests de khi2

Tests d'ajustement

Test d'ajustement pour une loi aux paramètres spécifiés

Mac68k:

```
> chisquare.test(o,p)
```

o: valeurs observées, p: vecteur des pronbabilités (si absent, probabilités égales)

PC et PPC:

```
> chisq.test(o,p)
```

```
o<-c(80,79,75,86,81,79)
chisq.test(o)
```

Test d'ajustement pour une loi aux paramètres estimés

```
> khi2.test<-function (o, p, m)
{
e <- sum(o) * p
u <- sum(((o - e)^2)/e)
dl <- length(o) - 1 - m
seuil <- (1 - pchisq(u, dl))
cat("\n")
cat("Valeur du khi2: ", round(u, 2), "avec", dl, "degrés de liberté", "\n")
cat("\n")
cat("Seuil observé: ", round(seuil * 100, 2), "%", "\n")
}
```

```
o<-c(1,9,14,2,4,4)
p<-c(0.0934,0.1809,0.2774,0.2506,0.1395,0.0582)
khi2.test(o,p,2)
```

Tables de contingence

```
X<-matrix(c(135,15,440,110,100,400,10,90),ncol=4)
```

Mac68k:

```
> chisquare.test(X)
X = 535.7842  df = 3   p-value = 0
```

PC et PPC:

```
> chisq.test(X,correct=F)

        Pearson's Chi-square test

data:  X
X-squared = 535.7842, df = 3, p-value = 0
```