

Classification supervisée : aperçu de 3 méthodes

Analyse discriminante - Arbre décisionnel - Régression logistique

On considère ici q classes d'individus sur lesquels p variables quantitatives et ou qualitatives sont mesurées ou observées. On note Y la matrice codée des classes et T la matrice des indicatrices et $X = (X^1, \dots, X^p)$ le tableau des variables quantitatives.

Ce chapitre concerne l'objectif décisionnel, permettant de classer de nouveaux individus dans les classes préexistantes connaissant les p variables.

On souhaite construire un modèle $E(Y|X) = f(X)$ classant un nouvel objet noté x dans une des classes (x représente aussi le vecteur des p variables). L'estimation du modèle est notée \hat{Y} .

Le cours ne présente que quelques méthodes simples, approfondies en master 2 par d'autres méthodes (réseaux de neurone, forêts aléatoires, ...).

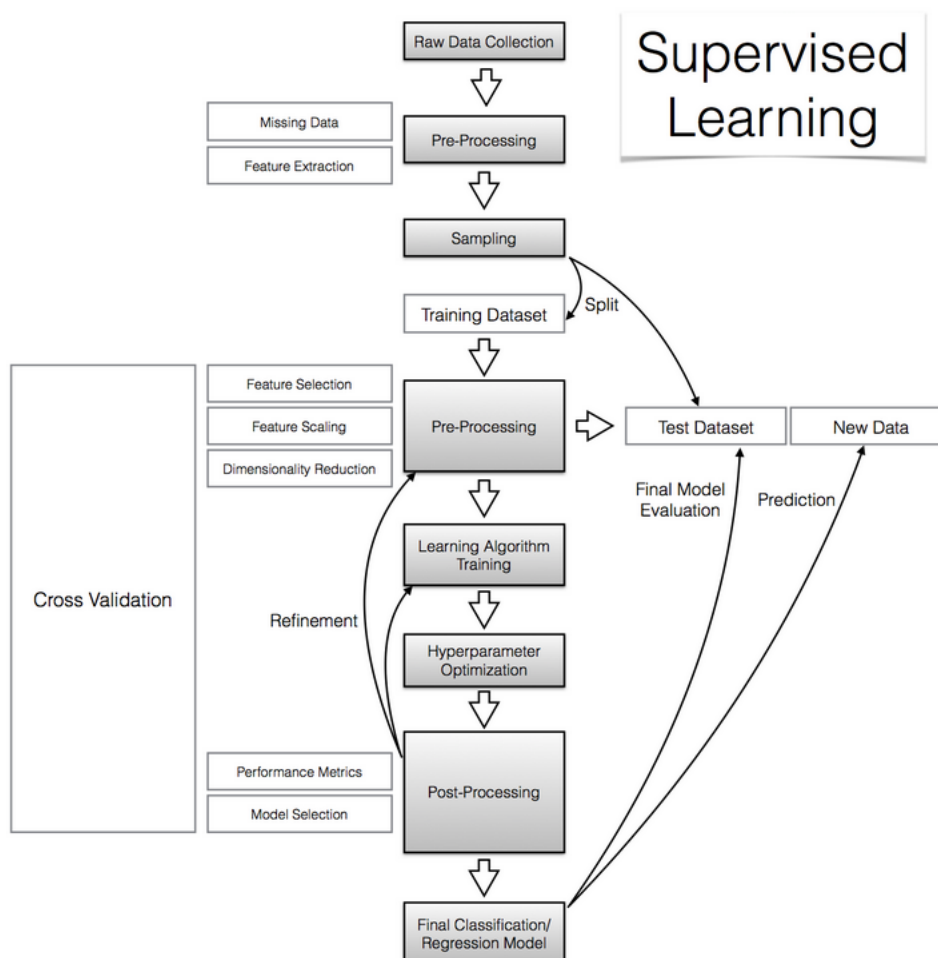


Table des matières

1	Analyse discriminante	3
1.1	Règles d'affectation	3
1.1.1	Approche géométrique	3
1.1.2	Approche probabiliste : Modèle bayésien	5
1.1.3	Modèle bayésien avec méthode paramétrique	5
1.1.4	Cas de deux classes	8
1.1.5	Modèle bayésien non paramétrique	10
1.2	Sélection de modèle	11
1.2.1	Sélection de variables	11
1.3	Qualité des règles de classement	12
1.3.1	Resubstitution	12
1.3.2	Echantillon test	12
1.3.3	Validation croisée (LOO, bootstrap)	12
2	Arbre de décision	13
2.1	Critère de sélection d'un nœud	13
2.2	exemple	14
3	Régression logistique : ML4 - CS3	15
3.1	Le modèle	15
3.2	Ajustement	17
3.3	Test du modèle	17
3.4	Exemple	18
4	Courbe ROC	19

1 Analyse discriminante

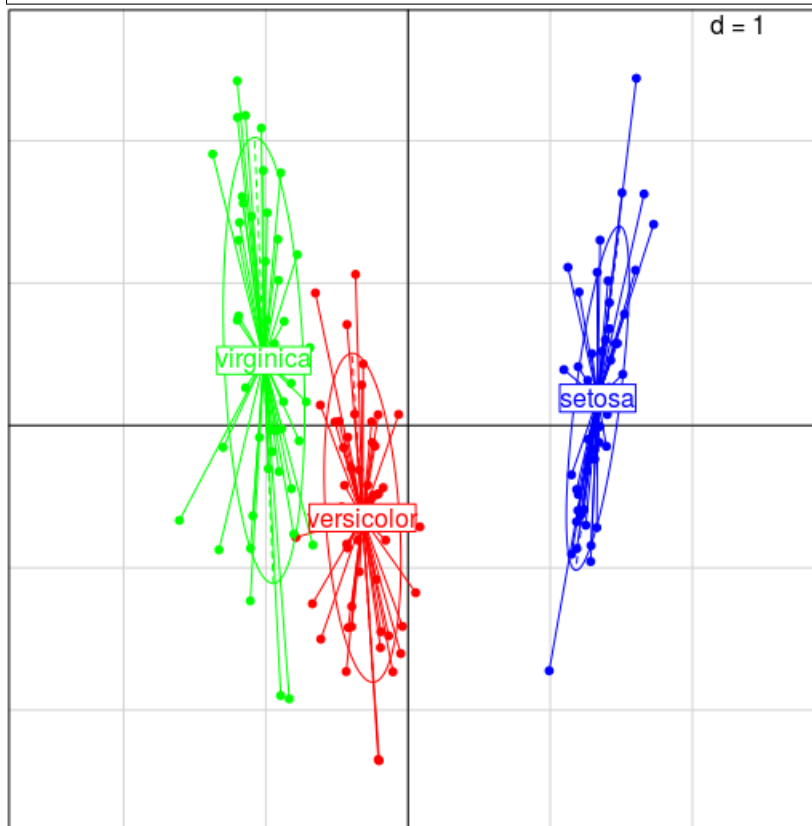
Nous reprendrons l'exemple classique iris pour illustrer le cours.

1.1 Règles d'affectation

1.1.1 Approche géométrique

L'approche géométrique consiste à affecter un nouvel individu à la classe dont le centre de gravité est le plus proche. La distance alors utilisée est la distance de Mahalanobis globale, de métrique \hat{W}^{-1} , ou locales \hat{W}_k^{-1} si les matrices de variance-covariance sont différentes. La distance globale ne prend pas en compte la forme de chaque nuage de chaque classe.

```
library(ade4)
data(iris)
iris.dis=discrimin(dudi.pca(iris[,1:4],scan=FALSE),iris$Species,scan=FALSE)
s.class(iris.disli, fac = iris$Species,col=c("blue","red","green"))
```



La distance d'un individu x à un groupe k est donnée par :

$$(x - g_k)^T \hat{W}^{-1} (x - g_k).$$

On peut alors calculer la distance de x à tous les groupes et choisir le groupe correspondant à la plus faible distance.

On obtient en développant :

$$d_n^2(x, g_k) = \overbrace{x^T \hat{W}^{-1} x}^{\text{constante (ne change pas peut importe } g_k)} - \underbrace{2g_k^T \hat{W}^{-1} x + g_k^T \hat{W}^{-1} g_k}_{\text{on cherche le min donc le max de l'opposé}}$$

En pratique, on calcule pour chaque groupe le terme

$$S(x, k) = 2g_k^T \hat{W}^{-1} x - g_k^T \hat{W}^{-1} g_k$$

On attribue le groupe qui rend maximal ce score linéaire en x .

On lui préfère une approche bayésienne prenant en compte les probabilités a priori d'affectation.

1.1.2 Approche probabiliste : Modèle bayésien

Soit un individu x . On définit $f_k(x)$ comme la densité de cet individu sachant qu'il appartient au groupe k . On note p_k les probabilités a priori des différents groupes.

Pour un individu donné, la probabilité qu'il appartienne au groupe k est d'après la formule de Bayes :

$$P(k/x) = \frac{p_k f_k(x)}{\sum_{l=1}^q p_l f_l(x)}.$$

On affecte alors l'individu au groupe dont la probabilité est la plus forte.

Comme le dénominateur est constant pour un individu donné, il suffit de déterminer le groupe pour lequel $p_k f_k(x|k)$ est le plus grand.

Les probabilités a priori sont déterminées en général sur le groupe d'apprentissage (celui utilisé pour ajuster l'analyse).

La fonction de densité peut être déterminée soit par une méthode non-paramétrique (méthode des noyaux, méthode des plus proches voisins) soit par une méthode paramétrique.

1.1.3 Modèle bayésien avec méthode paramétrique

On considère que les individus dans une classe donnée suivent une loi multinormale $\mathcal{N}_p(\mu_k, \Sigma)$ ou $\mathcal{N}_p(\mu_k, \Sigma_k)$ si les matrices de variance covariance sont différentes.

La densité de x pour un groupe k est :

$$f_k(x) = \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp \left[-\frac{(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)}{2} \right]$$

μ_k est estimé par g_k , Σ par \hat{W} , et Σ_k par \hat{W}_k si les matrices de variance-covariance sont différentes.

Le problème revient donc à maximiser :

$$p_k \frac{1}{(2\pi)^{p/2} |\Sigma_k|^{1/2}} \exp \left[-\frac{(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)}{2} \right]$$

ou son logarithme :

$$\log(p_k) - \frac{p}{2} \log(2\pi) - \frac{1}{2} \log(|\Sigma_k|) - \frac{(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)}{2}$$

soit à maximiser 2 fois l'expression :

$$-(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) - p \log(2\pi) - \log(|\Sigma_k|) + 2 \log(p_k).$$

$$-x^T \Sigma_k^{-1} x - \mu_k^T \Sigma_k^{-1} \mu_k + 2 \mu_k^T \Sigma_k^{-1} x - p \log(2\pi) - \log(|\Sigma_k|) + 2 \log(p_k).$$

Si les matrices de variances covariances peuvent être supposées égales, on peut supprimer les termes constants $-x^T \Sigma_k^{-1} x$, et $-p \log(2\pi) - \log(|\Sigma_k|)$. On obtient le critère linéaire en x suivant :

$$S(x) = 2\mu_k^T \Sigma^{-1} x - \mu_k^T \Sigma^{-1} \mu_k + 2 \log(p_k).$$

On parle alors d'analyse discriminante linéaire (lda).

Si les matrices de variances covariances ne peuvent être supposées égales, on obtient le critère quadratique en x suivant :

$$S(X_i) = -x^T \Sigma_k^{-1} x + 2\mu_k^T \Sigma_k^{-1} x - \mu_k^T \Sigma_k^{-1} \mu_k - \log(|\Sigma_k|) + 2 \log(p_k).$$

On parle alors d'analyse discriminante quadratique (qda).

Retour à l'exemple

Utilisons le modèle lda et qda pour prédire les classes et définir une règle en fonction de X_i .

```
library(MASS)
iris.lda=lda(iris$Species, data=iris[,1:4])
iris.qda=qda(iris$Species, data=iris[,1:4])
predict(iris.lda)
```

prédiction

```
$class
[1] setosa setosa setosa setosa setosa setosa setosa
[8] setosa setosa
```

on prend toutes les variables avec.

→ les 10 premiers individus prédit ∈ class setosa

```
$posterior
setosa versicolor virginica
1 1.000000e+00 3.896358e-22 2.611168e-42
2 1.000000e+00 7.217970e-18 5.042143e-37
```

individu 1, proba d'∈ au grp 1, 2, 3

```
$x
LD1 LD2
1 8.0617998 0.300420621
2 7.1286877 -0.786660426
```

coordonnée de l'individu

matrice de confusion

On compare ici les classes réelles et celles prédites sur l'échantillon d'apprentissage à travers une matrice de confusion permettant d'évaluer le taux de mal classés.

```
table(predict(iris.lda)$class, iris$Species)
table(predict(iris.qda)$class, iris$Species)
```

→ on compare prédit à la réalité

LDA

Y Species->	setosa	versicolor	virginica
\hat{Y} setosa	50	0	0
\hat{Y} versicolor	0	48	1
\hat{Y} virginica	0	2	49

*→ pas d'erreur de prédiction
petite erreur de prédiction*

QDA

Y Species ->	setosa	versicolor	virginica
\hat{Y} setosa	50	0	0
\hat{Y} versicolor	0	47	1
\hat{Y} virginica	0	3	49

```
# lda2 sur LD1 et LD2 pour faire une grille
iris.lda2=lda(iris$Species .,data=data.frame(predict(iris.lda)$x))
plot(predict(iris.lda)$x,col=c("blue","red","green")[iris$Species])
plot(predict(iris.lda2)$x,col=c("blue","red","green")[iris$Species])
```

```
# Construction des frontières
```

```
plot(predict(iris.lda)$x,col=c("blue","red","green")[iris$Species])
xp1<-seq(min(predict(iris.lda2)$x[,1]),max(predict(iris.lda2)$x[,1]),length=50)
xp2<-seq(min(predict(iris.lda2)$x[,2]),max(predict(iris.lda2)$x[,2]),length=50)
grille<-expand.grid(x1=xp1,x2=xp2)
grille=data.frame(LD1=grille[,1],LD2=grille[,2]) # grille contient les coordonnées x y de la grille du plan
```

```
Zp<-predict(iris.lda2,grille)
zp<-Zp$posterior[,3]-pmax(Zp$posterior[,2],Zp$posterior[,1])
```

```
#zp>0 si on classe en 3, <0 sinon
```

```
contour(xp1,xp2,matrix(zp,50),add=TRUE,levels=0,drawlabels=FALSE)
```

```
zp<-Zp$posterior[,2]-pmax(Zp$posterior[,1],Zp$posterior[,3])
```

```
#zp >0 si on classe en 2, <0 sinon
```

```
contour(xp1,xp2,matrix(zp,50),add=TRUE,levels=0,drawlabels=FALSE)
```

```
#la même chose avec qda
```

```
iris.qda=qda(iris$Species .,data=data.frame(predict(iris.lda)$x))
```

```
table(iris$Species,predict(iris.lda)$class)
```

```
xp1<-seq(min(predict(iris.lda2)$x[,1]),max(predict(iris.lda2)$x[,1]),length=50)
```

```
xp2<-seq(min(predict(iris.lda2)$x[,2]),max(predict(iris.lda2)$x[,2]),length=50)
```

```
grille<-expand.grid(x1=xp1,x2=xp2)
```

```
grille=data.frame(LD1=grille[,1],LD2=grille[,2])
```

```
Zp<-predict(iris.qda,grille)
```

```
zp<-Zp$posterior[,3]-pmax(Zp$posterior[,2],Zp$posterior[,1])
```

```
contour(xp1,xp2,matrix(zp,50),add=TRUE,levels=0,drawlabels=FALSE,lty=2)
```

```
zp<-Zp$posterior[,2]-pmax(Zp$posterior[,1],Zp$posterior[,3])
```

```
contour(xp1,xp2,matrix(zp,50),add=TRUE,levels=0,drawlabels=FALSE,lty=2)
```

```
legend(7, -2, legend=c("lda", "qda"), lty=1 :2, cex=0.8)
```

```
title(main="Frontières avec lda et qda")
```

```
library(ade4)
```

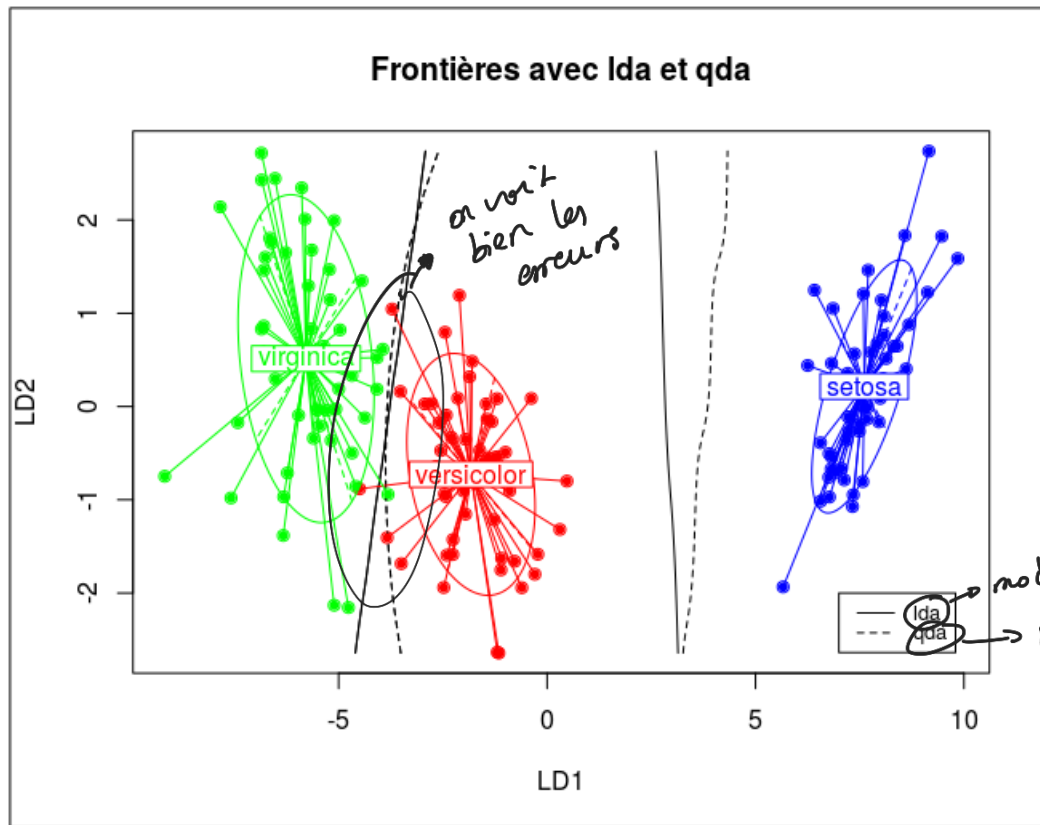
```
s.class(predict(iris.lda)$x,fac=iris$Species, col=c("blue","red","green"),add.plot=TRUE)
```

on construit
une grille

→ proba d'être en 3

→ si zp < 0 → grp 3.

→ on construit les
2 frontières



1.1.4 Cas de deux classes

On se place dans le cas de deux classes, 1 et 2, en supposant une même matrice de covariances intra $\Sigma_1 = \Sigma_2 = \Sigma$. On affecte alors un individu x au groupe 1 si $p_1 f_1(x; \mu_1, \Sigma) > p_2 f_2(x; \mu_2, \Sigma)$, soit :

$$2x^T \Sigma^{-1} \mu_1 - \mu_1^T \Sigma^{-1} \mu_1 + 2 \log(p_1) > 2x^T \Sigma^{-1} \mu_2 - \mu_2^T \Sigma^{-1} \mu_2 + 2 \log(p_2)$$

$$2x^T \Sigma^{-1} (\mu_1 - \mu_2) > \mu_1^T \Sigma^{-1} \mu_1 - \mu_2^T \Sigma^{-1} \mu_2 + 2 \log\left(\frac{p_2}{p_1}\right)$$

$$x^T \Sigma^{-1} (\mu_1 - \mu_2) > \frac{1}{2} (\mu_1 + \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2) + \underbrace{\log\left(\frac{p_2}{p_1}\right)}_{\ln !}$$

$$p_1 f_1(x) > p_2 f_2(x) \Leftrightarrow \ln(p_1) + \ln(f_1(x)) > \ln(p_2) + \ln(f_2(x))$$

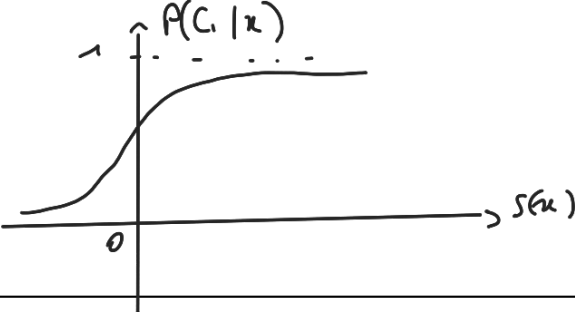
$$\Leftrightarrow \ln\left(\frac{f_1(x)}{f_2(x)}\right) > \ln\left(\frac{p_2}{p_1}\right)$$

Proposition 1 On définit le score $S(x) = x^T \Sigma^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) - \log\left(\frac{p_2}{p_1}\right)$.
On affecte alors x :

- au groupe 1 si $S(x) > 0$,
- au groupe 2 si $s(x) < 0$.

On a de plus $P(C_1|x) = \frac{1}{1 + e^{-S(x)}}$. La probabilité a posteriori que x appartienne à la classe 1 est la fonction logistique de $S(x)$.

$$S(x) = \ln(p_1 f_1(x)) - \ln(p_2 f_2(x))$$

$$\Rightarrow \frac{1}{1 + e^{-\ln(p_1 f_1(x)) - \ln(p_2 f_2(x))}} = \frac{e^{\ln(p_1 f_1(x))}}{e^{\ln(p_1 f_1(x))} + e^{\ln(p_2 f_2(x))}} = \frac{p_1 f_1(x)}{p_1 f_1(x) + p_2 f_2(x)}$$


Proposition 2 Dans le cas $p_1 = p_2$, $S(x)$ devient

$$S(X) = x^T \Sigma^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)$$

Le taux d'erreur de classement dans le groupe 1 est ainsi : $P(C_1|x \in C_2) = P(S(x) > 0 | x \sim \mathcal{N}(\mu_2, \Sigma))$.

Sous l'hypothèse $x \in C_2$, $E_{\mu_2}(S(x)) = -\frac{1}{2}\Delta_p^2$ avec $\Delta_p^2 = (\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)$ la distance de Mahalanobis entre les deux moyennes, et $\text{Var}(S(x)) = \Delta_p^2$.

On en déduit :

$$P(C_1|x \in C_2) = P(C_2|x \in C_1) = P(U > \frac{\Delta_p}{2})$$

avec U la loi normale centrée réduite. Ainsi, plus la distance de Mahalanobis est grande, plus la discrimination est bonne.

$$E_2(S(X)) = \mu_2^T \Sigma^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) = \frac{1}{2}(\mu_2 - \mu_1)^T \Sigma^{-1}(\mu_1 - \mu_2) = -\frac{1}{2}\Delta_p^2$$

$$V_2(S(X)) = V((\mu_1 - \mu_2)^T \Sigma^{-1}x - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)) = V_2(Ax + B) = A \Sigma A^T = (\mu_1 - \mu_2)^T \Sigma^{-1} \Sigma \Sigma^{-1}(\mu_1 - \mu_2) = \Delta_p^2$$

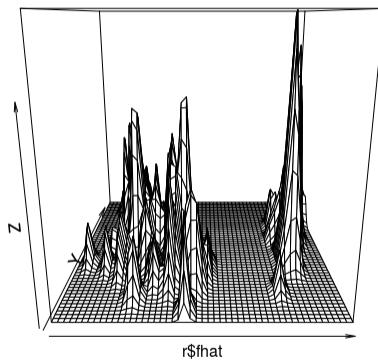
$$P(1/2) = P_2(S(X) > 0) = P_2((S(X) + \frac{1}{2}\Delta_p^2)/\Delta_p > \frac{1}{2}\Delta_p^2/\Delta_p = \frac{1}{2}\Delta_p)$$

1.1.5 Modèle bayésien non paramétrique

Si l'hypothèse de distribution gaussienne n'est pas réaliste, il est aussi possible d'utiliser une densité non paramétrique, \hat{f}_k . On présentera les deux méthodes les plus classiques.

méthode du noyau On étend à l'espace \mathbb{R}^p les densités à noyau que vous avez abordé en statistiques en S1. Par exemple avec un noyau uniforme, en tout point x de l'espace, la densité est proportionnelle au nombre de points du groupe k dans une sphère centrée en x . La sphère est définie par la métrique utilisée et dépend d'un paramètre son rayon. En changeant le noyau (normal...), on obtient des estimations plus fines.

```
library(KernSmooth)
iris.lda=lda(iris$Species, data=iris[,1:4])
X=predict(iris.lda)$x
r <- bkde2D(X, bandwidth=c(.2,.2),range.x=list(c(-10,10),c(-3,3)))
persp(r$fhat,col=0)
```



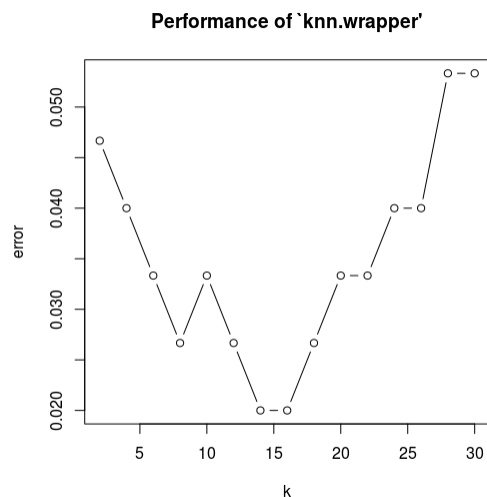
Méthode des k plus proches voisins On cherche les r points les plus proches de l'individu x et on classe x dans le groupe le plus représenté : la probabilité a posteriori d'appartenir au groupe k est égale au quotient entre le nombre d'individus du groupe k parmi les r points, et le nombre de voisins r .

```
library(class)
?knn
iris.knn=knn(iris[,1:4],iris[,1:4],iris[,5],k=10)
iris.knn[1:5]
[1] setosa setosa setosa setosa setosa
Levels : setosa versicolor virginica
#matrice de confusion
table(iris$Species,iris.knn) iris.knn[1:5]
#matrice de confusion
table(iris$Species,iris.knn)
```

	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	49	1
virginica	0	1	49

Recherche du k optimal : On réalise l'apprentissage sur différentes valeurs de k et on étudie l'erreur par validation LOO

```
library(e1071)
plot(tune.knn(iris[,1:4],iris[,5],k=seq(2,30, by=2)))
```



1.2 Sélection de modèle

Le choix d'un modèle va dépendre des variables en entrée, des variables discriminantes retenues et de la qualité de prédiction du modèle.

1.2.1 Sélection de variables

Variables en entrée Toutes les variables n'apportent pas d'information sur l'appartenance ou pas à un groupe, il existe des procédures, comme STEPDISC (SAS) pour sélectionner les variables pertinentes.

```
library(klaR)
greedy.wilks(Species ~.,data=iris, niveau = 0.01)
Species ~ Petal.Length + Sepal.Width + Petal.Width
```

Variables qualitatives Il est possible d'introduire dans le modèle des variables qualitatives en utilisant les composantes principales de leur ACM, on parle alors de la méthode DISQUAL.

Variables discriminantes Seules les premières variables discriminantes ont un fort pouvoir discriminant, il est possible avec le test de Bartlett de tester le pouvoir discriminant des dernières variables discriminantes (voir cours RD4).

1.3 Qualité des règles de classement

1.3.1 Resubstitution

On utilise le groupe d'apprentissage pour évaluer le taux d'erreur. Chacun des n individus est affecté à un groupe à l'aide des fonctions discriminantes ajustées. L'inconvénient est que le taux estimé est sous-estimé car estimé à partir des individus utilisés dans l'analyse.

1.3.2 Echantillon test

Si le nombre d'individus est très important, il est possible d'utiliser une partie des individus (75%) pour ajuster les fonctions discriminantes (groupe d'apprentissage) et une autre partie (25%) pour estimer le taux d'erreur. On peut utiliser la fonction sample ou des fonctions dédiées (voir TD).

1.3.3 Validation croisée *leave one out* (LOO, bootstrap)

Si l'échantillon est trop faible, pour chacun des n individus, on estime les fonctions discriminantes sur les $n-1$ individus restants et on teste alors la règle sur l'individu. On peut utiliser des fonctions dédiées (voir TD).

2 Arbre de décision

L'apprentissage se fait par partitionnement récursif selon des règles sur les variables explicatives. Suivant les critères de partitionnement et les données, on dispose de différentes méthodes, dont CART, CHAID ... Ces méthodes peuvent s'appliquer à une variable à expliquer qualitative ou quantitative. Deux types d'arbres de décision sont ainsi définis :

- arbres de classification : la variable expliquée est de type nominale (facteur). A chaque étape du partitionnement, on cherche à réduire l'impureté totale des deux nœuds fils par rapport au nœud père.
- arbres de régression : la variable expliquée est de type numérique et il s'agit de prédire une valeur la plus proche possible de la vraie valeur.

Construire un tel arbre consiste à définir une suite de nœud, chaque nœud permettant de faire une partition des objets en 2 groupes sur la base d'une des variables explicatives. Il convient donc :

- de définir un critère permettant de sélectionner le meilleur nœud possible à une étape donnée,
- de définir quand s'arrête le découpage, en définissant un nœud terminal (feuille),
- d'attribuer au nœud terminal la classe ou la valeur la plus probable,
- d'élaguer l'arbre quand le nombre de nœuds devient trop important en sélectionnant un sous arbre optimal à partir de l'arbre maximal,
- valider l'arbre à partir d'une validation croisée ou d'autres techniques

2.1 Critère de sélection d'un nœud

La construction d'un nœud doit réduire de façon optimale le désordre des objets. Pour mesurer ce désordre, on définit l'entropie d'une variable qualitative Y à q modalités par :

$$H(Y) = - \sum_{k=1}^q P(Y = k) \times \log(P(Y = k)) \quad \text{avec} \quad 0 \log(0) = 0.$$

$H(Y)$ est un critère d'incertitude de Y , il vaut 0 si le groupe est homogène ($P(Y = k) = 1$) pour une classe k donnée.

On peut ensuite définir l'entropie de Y conditionnée par une variable qualitative X ayant q' modalités :

$$H(Y|X) = - \sum_{k,k'} P(Y = k, X = k') \times \log(P(Y = k|X = k')) = - \sum_{k'} P(X = k') \times H(Y|X = k').$$

Par exemple, considérons pour deux modalités de Y et de X , on observe le tableau. Calculer $H(Y)$ puis $H(Y|X)$.

	$X = 0$	$X = 1$
$Y = 0$	20	5
$Y = 1$	3	10

Plus X nous renseigne sur Y plus l'entropie conditionnelle diminue. A l'extrême, $H(Y|Y) = 0$. On choisira le nœud de façon à réduire au maximum le désordre. Il existe d'autres critères :

- critère de Gini : $G = 1 - \sum_{k,k'} p_{kk'}$,
- pour un arbre de régression, on utilise le test de Fisher comme critère et dans l'algorithme CHAID, le test du χ^2 .

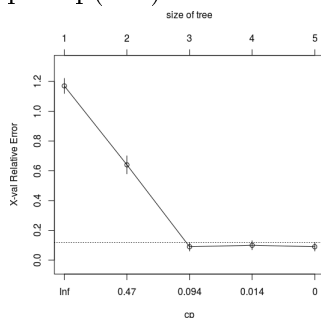
Pour éviter le sur-apprentissage rendant l'arbre insuffisamment robuste, il faut souvent élaguer l'arbre maximal mais ces arbres sont souvent assez instables (règles peuvent changer facilement selon les données d'apprentissage).

2.2 exemple

```
library(rpart)# Pour l'arbre de décision
library(rpart.plot) # Pour la représentation de l'arbre de décision
TR <- rpart(Species~.,data=iris,method='class', control=rpart.control(minsplit=5,cp=0))
plot(TR, uniform=TRUE, branch=0.5, margin=0.1); text(TR, cex=1,all=FALSE,
use.n=TRUE)
```



```
#Elagage de l'arbre avec le cp optimal
plotcp(TR)
```



```
OTR <- prune(TR,cp=0.094) # coupure de l'arbre
plot(OTR, uniform=TRUE, branch=0.5, margin=0.15); text(OTR, cex=1,all=FALSE,
use.n=TRUE)
```

3 Régression logistique : ML4 - CS3

La méthode porte sur une variable binaire Y (0 ou 1) que l'on cherche à expliquer ou prédire par des variables X_i qualitatives ou quantitatives. L'objectif est d'expliquer Y à partir de relations linéaires avec les variables X_i pour pouvoir analyser leur contribution respective. Le terme aléatoire ne peut plus être gaussien et on parle alors de modèle linéaire généralisé.

3.1 Le modèle

Nous cherchons à expliquer Y connaissant $X = x$ mais Y ne prenant que la valeur 0 ou 1, nous cherchons en fait à estimer :

$$\mathbb{P}(Y = 1|X = x) = \pi(x).$$

Prenons X une variable qualitative à q modalités. Par exemple le tableau suivant :

X/Y	Sciences	Lettre
Fille	5	10
Garçon	20	10

Il est intuitif de proposer pour chaque valeur possible x_i de X pour $\pi(x_i)$ comme estimation :

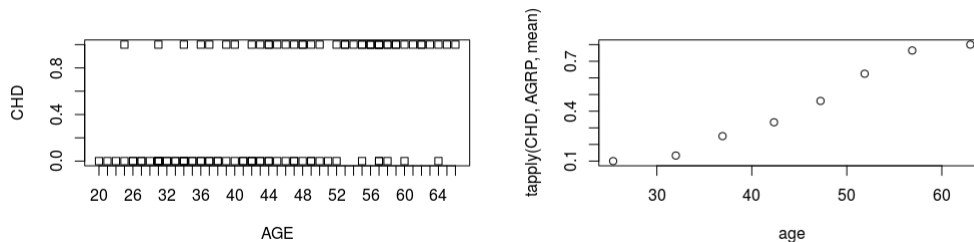
$$\pi(\hat{x}_i) = \frac{\#(Y = 1, X = x_i)}{\#(X = x_i)}$$

représentant la fréquence de $Y = 1$ pour la modalité x_i .

On pourrait alors écrire le modèle avec q paramètres sous la forme $Y = X\beta + \epsilon$, avec X le tableau disjonctif, β les probabilités conditionnelles, Y suivant une loi de bernoulli de paramètre $\pi(x_i)$ et ϵ_i une loi uniforme discrète.

Prenons maintenant une variable continue X . En représentant Y en fonction de X , aucune relation semble adaptée, encore moins une relation linéaire. Une relation apparaît plus clairement en transformant X en variable qualitative par classe.

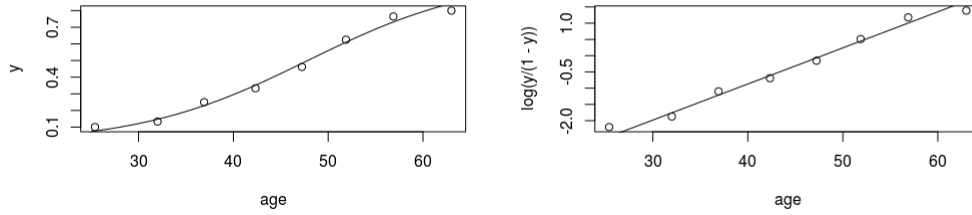
```
cardio=read.table("cardio.txt",h=T);attach(cardio);stripchart(CHD AGE,vertical=T,xlab="AGE")
age=tapply(AGE,AGRP,mean);age;y=tapply(CHD,AGRP,mean);
plot(age,tapply(CHD,AGRP,mean))
```



On observe alors une relation de forme sigmoïde entre $\pi(c_i)$ et c_i le centre des classes que nous cherchons à modéliser sous forme linéaire. On rappelle qu'une fonction sigmoïde $Sig(x)$ s'écrit :

$$Sig(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}.$$

```
plot(age,y); x=seq(25,63,length=50);lines(x,exp(-5.31+0.111*x)/(1 + exp(-5.31 + 0.111*x)))
plot(age,log(y/(1-y))); lines(x,-5.31+0.111*x)
```



On pose ainsi sa fonction réciproque appelée fonction logit définie par :

$$\text{logit}(x) = \log \frac{x}{1-x}$$

que nous appliquons à $\pi(x_i)$. La relation entre x_i et $\pi(x_i)$ devient alors linéaire. Cette fonction est appelée fonction de lien. On obtient ainsi :

$$\text{logit}(\pi(x_i)) = \beta_0 + \beta_1 x_i.$$

Interprétation

Pour cela on définit le rapport des chances de $Y = 1$ conditionnellement à $X = x$ appelé cote ou Odds :

cote de $x = \text{Odd}(x) = \frac{\pi_x}{1-\pi_x}$ s'interprétant comme le rapport des chances de $Y = 1$ par rapport aux chances de $Y = 0$ conditionnellement à $X = x$. Par exemple un garçon a $20/10=2$ fois plus de chance de faire sciences que lettre et les filles $5/10$ deux fois moins de chance.

On définit alors l'Odds Ratio pour deux valeurs x et x' comme le rapport des cotes :

$$\text{OR}(x', x) = \frac{\text{cote}(x')}{\text{cote}(x)}.$$

Par exemple, le rapport des cotes garçon fille est 4 soit les garçons ont 4 fois plus de chance de faire sciences par rapport aux filles.

La cote de x est une fonction à valeur dans $[0, +\infty[$, et cote de x n'est toujours pas linéaire. On obtient le résultat par passage au logarithme, $\log(\text{cote})$ à valeur dans \mathbb{R} et $\text{logit}(\pi(x))$ devient bien linéaire en x .

Définition 1 *Modèle logistique*

Soit une variable dichotomique Y et des variables qualitatives ou quantitative $X = (X_1, \dots, X_p)$. Le modèle logistique modélise la loi de $Y|X = x$ par une loi de Bernoulli de paramètre $\pi_\beta(x) = \mathbb{P}(Y = 1|X = x)$ telle que :

$$\log \frac{\pi_\beta(x)}{1 - \pi_\beta(x)} = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

soit

$$\text{logit}(\pi_\beta(x)) = x^t \beta.$$

On a ainsi :

- $\pi_\beta(x) = \frac{\exp(x^t \beta)}{1 + \exp(x^t \beta)}$
- $\mathbb{E}_\beta(Y|X = x) = \pi_\beta(x)$

- Y suit une loi de Bernoulli de paramètre $\pi_\beta(x)$

Remarque

Nous ne traitons que ce cas, mais suivant la nature de Y , dichotomique, à q valeurs ou à valeur dans \mathbb{N} , et suivant la fonction de lien (probit, log) , on obtient des modèles linéaires généralisés différents (Poisson, polytomique...)

Remarque

On peut ainsi paramétrer le modèle pour les variables quantitatives et qualitatives ainsi que leurs interactions. Par exemple :

- X_1 une variable qualitative : $\text{logit}(\pi_\beta(x_i)) = \beta_0 + \beta_2 I_{x_2}(x_i) + \dots + \beta_q I_{x_q}(x_i)$

On retrouve un même paramétrage qu'en ANOVA.

- X_1 une variable quantitative : $\text{logit}(\pi_\beta(x_i)) = \beta_0 + \beta_1 x_i$

On retrouve le paramétrage en régression simple et multiple

Interprétation

- Pour une variable qualitative ou quantitative, $\beta_i \approx 0$ indique l'absence d'influence significative de X sur Y .
- Si β_i est grand pour une variable quantitative X_i , la réponse passera rapidement de $Y = 0$ à $Y = 1$. On peut interpréter aussi β avec les OR, une variation de X_i de 1 conduit à un OR de $\exp(\beta_i)$.
- Soit β_i et β'_i les coefficients de 2 modalités, l'OR(x, x') est alors $\exp(\beta_i - \beta'_i)$.

3.2 Ajustement

Reprenons l'exemple. Le paramétrage pourrait être $\text{logit}(\pi_\beta(G)) = \beta_0 + \beta_1 * 0$ et $\text{logit}(\pi_\beta(F)) = \beta_0 + \beta_1 * 1$ identifiable et intuitivement poser : $\hat{\beta}_0 = \text{Sig}(20/30) \simeq 0.67$ et $\hat{\beta}_0 + \hat{\beta}_1 = \text{Sig}(5/15) \simeq 0.33$.

On utilise en fait l'estimateur du maximum de vraisemblance.

Proposition 3 Soit Y une variable binaire et $X = (X_1, \dots, X_p)$ p variables explicatives. On suppose les observations indépendantes. Avec le modèle logistique, on suppose que $Y_i \sim \mathcal{B}(\pi_\beta(x_i))$ et la vraisemblance s'écrit :

$$l(\underline{X}, \beta) = \prod_{i=1}^n \mathbb{P}_\beta(Y = y_i | X = x_i) = \prod_{i=1}^n \pi_\beta(x_i)^{y_i} (1 - \pi_\beta(x_i))^{1-y_i}$$

Pour estimer les coefficients, on utilise le plus souvent la méthode de Newton-Raphson

3.3 Test du modèle

Test sur les coefficients

Proposition 4 On utilise le comportement asymptotique des estimateurs du maximum de vraisemblance :

$$\sqrt{n}(\hat{\beta} - \beta) \rightarrow \mathcal{N}(0, I(\beta)^{-1})$$

avec $I(\beta) = H$ l'information de Fisher.

On estime H avec les estimations des coefficients β .

Proposition 5 Pour tester $\mathcal{H}_0 : \beta_j = 0$, on utilise la statistique de Wald :

$$W_j = \frac{\hat{\beta}_j^2}{\hat{\sigma}_j}$$

avec $\hat{\sigma}_j$ l'estimation de l'écart-type obtenue avec H^{-1} . Sous $\mathcal{H}_0 : \beta_j = 0$, $W_j \sim \chi_{(1)}^2$.
On peut également utiliser ce résultat pour la construction d'un IC.

Nullité de plusieurs coefficients- Modèles emboîtés

On souhaite tester maintenant :

$$\mathcal{H}_0 : \beta_{k_1} = \dots = \beta_{k_p} = 0$$

Proposition 6 Pour tester $\mathcal{H}_0 : \beta_{k_1} = \dots = \beta_{k_p} = 0$, on peut utiliser le test du rapport de vraisemblance ou de la déviance, la déviance étant $-2 \times (l_{\text{mod}} - l_{\text{sat}})$. l_{sat} représente le ll du modèle complet (un paramètre par combinaison de x , 0 en absence de répétitions de x)

On obtient alors sous \mathcal{H}_0 :

$$2(l_{\hat{\beta}} - l_{\hat{\beta}^0}) \rightarrow \chi_{(p)}^2$$

avec $l_{\hat{\beta}_0}$ la log-vraisemblance du modèle contraint.

Il existe d'autres tests : test de Wald, test du Score.

Sélection de modèles Pour choisir le meilleur modèle, il faut comparer un nombre très importants de modèles non forcément emboîtés. On utilise le critère AIC permettant de comparer les modèles en pénalisant le nombre de paramètres utilisé aussi en régression multiple.

3.4 Exemple

```
cardio.glm=glm(CHD AGE,family=binomial)

summary(cardio.glm)

Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.30945 1.13365 -4.683 2.82e-06 ***
AGE 0.11092 0.02406 4.610 4.02e-06 ***

Null deviance : 136.66 on 99 degrees of freedom
Residual deviance : 107.35 on 98 degrees of freedom
AIC : 111.35
Number of Fisher Scoring iterations : 4

table(cardio.glm$fitted.values>0.5,CHD)
CHD
0 1
FALSE 45 14
TRUE 12 29

library(car)
Anova(cardio.glm, type="II", test="LR")
Response : CHD
LR Chisq Df Pr(>Chisq)
AGE 29.31 1 6.168e-08 ***
```

4 Courbe ROC

On se place dans le cas d'une réponse binaire $+/-$. On obtient une matrice de confusion de la forme

	$Y = 0$	$Y = 1$
$\hat{Y} = 0$		
$\hat{Y} = 1$		

avec

- TP (true positives) : les prédits positifs qui le sont vraiment.
- FP (false positives) : les prédits positifs qui sont en fait négatifs.
- TN (true negatives) : les prédits négatifs qui le sont vraiment.
- FN (false negatives) : les prédits négatifs qui sont en fait positifs.
- P (positives) : tous les positifs quelque soit l'état de leur prédiction. $P = TP + FN$.
- N (negatives) : tous les négatifs quelque soit l'état de leur prédiction. $N = TN + FP$.

On définit alors la spécificité et la sensibilité :

- la sensibilité est : $TP / (TP + FN) = TP / P$.
- la spécificité est : $TN / (TN + FP) = TN / N$.

Principe de la courbe ROC :

On dispose d'une fonction score, $S(x)$, construite avec un modèle en fonction de X , avec laquelle prédire Y . La règle de classement dépend alors d'un seuil s tel que :

- si $S(x) > s$, alors \hat{Y} est positif (1),
- si $S(x) < s$, alors \hat{Y} est négatif (0),

Au fur et à mesure que s augmente :

- la spécificité augmente.
- mais la sensibilité diminue.

La courbe ROC représente l'évolution de la sensibilité (taux de vrais positifs) en fonction de 1 - spécificité (taux de faux positifs) quand on fait varier le seuil t .

- C'est une courbe croissante entre le point (0,0) et le point (1, 1) et en principe au-dessus de la première bissectrice.
- Une prédiction random donnerait la première bissectrice.
- Meilleure est la prédiction, plus la courbe est au-dessus de la première bissectrice.
- Une prédiction idéale est l'horizontale $y=1$ sur $]0,1]$ et le point (0,0).
- L'aire sous la courbe ROC (AUC, Area Under the Curve) donne un indicateur de la qualité de la prédiction (1 pour une prédiction idéale, 0.5 pour une prédiction random).

Example

```
library(ROCR)
library(MASS)
cancer=read.table("cancerprostate.txt",h=T,sep=';')
Y=cancer$Y;X=cancer[,c(1:5,7)]
cancer.lda=lda(Y.,data=X)
cancer.glm=glm(Y.,data=X,family=binomial)
library(rpart)#Pour l'arbre de décision
cancer.TR <- rpart(Y.,data=X,method='class', control=rpart.control(minsplit=5,cp=0))
#Matrice de confusion
table(Y,predict(cancer.lda,type='response')$class); table(Y,pglm>0.5); table(Y,pTR==1)
```

LDA	= Y = 0	Y = 1	GLM	Y = 0	Y = 1	TR	= Y = 0	Y = 1
$\hat{Y} = 0$	29		4	28	5		33	0
$\hat{Y} = 1$	6	14		7	13		5	15

```
plda=predict(cancer.lda,type='response')$x #P(Y=1|X)
pglm=predict(cancer.glm,type='response') #P(Y=1|X)
pTR=predict(cancer.TR)[,2]
p=prediction(plda,Y)
cbind(p@cutoffs[[1]],p@tn[[1]],p@fn[[1]],p@tp[[1]],p@fp[[1]])[1:5,]
s TN FN TP FP
[1,] Inf 33 20 0 0
[2,] 2.660766 33 19 1 0
[3,] 2.463575 33 18 2 0
perf.lda <- performance(prediction(plda,Y), "tpr", "fpr")
plot(perf.lda,col=2)
perf.glm<- performance(prediction(pglm,Y), "tpr", "fpr")
plot(perf.glm,col=3,lty=2,add=TRUE)
perf.TR<- performance(prediction(pTR,Y), "tpr", "fpr")
plot(perf.TR,col=4,add=TRUE,lty=3)
legend(0.6,0.2,c('lda','glm','TR'),lty=1:3,col=2:4)
```

