Projet

Nadia GHERNAOUT
Philippine RENAUDIN

Table des matières

1 Introduction générale

L'idée générale du scoring est d'affecter une note (un score) globale à un individu à partir de plusieurs descripteurs, quantitatifs ou qualitatifs. À partir de cette note, on affecte l'individu à un groupe préexistant. Un score peut donc être définit comme un outil statistique ou probabiliste de détection de risque. Le scoring peut également être vu comme l'application au monde de l'entreprise de plusieurs techniques de classement. Nous en aborderons 2 dans ce rapport.

Nous pouvons déjà citer plusieurs types de score :

1. Les scores de risque :
   — risque de crédit ou credit scoring : prédire le retard de remboursement de crédit.
   — risque financier : prédire la bonne ou mauvaise santé d'une entreprise.
   — risque médical : prédire l'apparition d'une maladie chez un patient.
2. Les scores en marketing :
   — score d'attrition : prédire le risque qu'un client passe à la concurrence ou résilie son abonnement.
   — score d'appétence : prédire l'appétence d'un client à acheter tel ou tel type de produit.

La création d'un score se fait en fonction des objectifs recherchés et des moyens techniques disponibles. Par exemple, le développement d'un score comportemental nécessite de disposer de données sur au moins un an, si l'on a moins d'historique, il vaut mieux partir sur un score générique ou un score d'octroi.

Il faut aussi également choisir l'utilisation qui sera faite du score : outil d'aide à la décision ou outil de ciblage pour le marketing direct par exemple. C'est en fonction de l'utilisation que l'on en fera que la règle de décision sera ajustée.

Pour construire un score, il faut dans un premier temps disposer d'un échantillon suffisamment conséquent pour pouvoir tester plusieurs modèles prédictifs. De plus, pour éviter des problèmes de surestimation de la qualité du modèle, il est préférable de séparer l'échantillon d'étude en deux sous-échantillons : un échantillon d'apprentissage à partir duquel sera créé le modèle, et un échantillon test sur lequel sera testé la qualité du modèle par rapport à l'objectif recherché et au risque que l'on est prêt à prendre.

Ensuite, il faut élaborer un modèle prédictif à l'aide de techniques prédictives : analyse discriminante et régression logistique en l'occurrence.

Enfin, les notes de score sont découpées en plusieurs classes de valeur. Dans le domaine financier, on aura tendance à découper les notes de score en trois classes : faible, moyen, fortes. Dans le milieu médical, on préfèrera 2 classes : à risque, non à risque. La règle de classement (seuil comparatif du score) se décide en fonction du risque d'erreur que l'on souhaite prendre.

Nous présentons dans ce rapport 2 des techniques prédictives les plus utilisées en scoring : la régression logistique et l'analyse discriminante. Pour illustrer ce qu'est le scoring, nous avons utilisé ces 2 techniques sur 2 jeux de données différents. Nous présentons dans la suite la théorie de chaque technique ainsi que l'étude des données associée.

Première partie

Protocole

1 Validation croisée

Comme il a été mentionné en introduction, une habitude à prendre lors de toute analyse de données est de séparer les données en plusieurs sous échantillons pour éviter les problèmes de surestimation des capacités du modèle.

On appelle validation croisée la technique consistant à ajuster un modèle prédictif sur un échantillon d'apprentissage et à valider ce modèle sur un échantillon test. Ces échantillons peuvent provenir du même jeu de données auquel cas il est coutume que l'échantillon d'apprentissage représente entre 60% et 80% des données et que l'échantillon test représente 20% à 40%. Il est également possible, si l'on dispose de plusieurs jeux de données différents pour le sujet d'étude, de prendre un jeu de données comme échantillon d'apprentissage et de valider le modèle sur un deuxième jeu de données.

Nous avons choisi dans nos études de cas de réaliser une validation croisée à partir d'un seul jeu de données en le décomposant en un échantillon d'apprentissage représentant 80% du jeu de données et en un échantillon test représentant les 20% restant, et ce de manière aléatoire.

2 Sélection de modèles

Une fois le jeu de données séparé en deux échantillons, vient le moment de construire différents modèles prédictifs. Cependant, il n'est pas toujours évident de savoir quelles variables garder, quelles sont celles qui apportent le plus d'information, qui discriminent le mieux les groupes d'individus, etc... C'est pourquoi on s'appuie sur différents indicateurs, en plus de ceux implémentés par défaut dans les logiciels. Nous en évoquons 3 ici :

2.1 Tests entre modèles emboîtés

A l'image de ce qui est fait en régression linéaire il existe des tests entre modèles emboités, on souhaite comparer un modèle restreint de p_0 paramètres au modèle global (à p paramètres).

Soit $p_0 < p$, on compare le modèle $\mathcal{M}_0$

$$\text{logit}(p_\gamma(x)) = \gamma_1 x_1 + \cdots + \gamma_{p_0} x_{p_0}$$

avec le modèle $\mathcal{M}_1$

$$\text{logit}(p_\beta(x)) = \beta_1 x_1 + \cdots + \beta_p x_p$$

On peut ainsi faire le test d'hypothèses suivant :

$$\mathcal{H}_0 : "\beta_{p_0+1} = \cdots = \beta_p = 0" \text{ contre } \mathcal{H}_1 : "\exists j \in \{p_0 + 1, \cdots, p\} : \beta_j \neq 0"$$

Ne pas rejeter $\mathcal{H}_0$ signifie privilégier le modèle $\mathcal{M}_0$ au détriment du modèle $\mathcal{M}_1$.

Ce test peut être réalisé à l'aide du test de Wald ou test de rapport de vraisemblance(déviance).

2.1.1 Le test de Wald

2.1.2 La déviance

Nous verrons plus tard dans ce rapport que les paramètres des différents modèles sont le plus souvent estimés par la méthode du maximum de vraisemblance.

Pour savoir quels modèles garder, il est donc courant d'utiliser le critère de la déviance. En effet, la déviance est égale à -2log-vraisemblance donc le modèle maximisant la vraisemblance est celui minimisant la déviance. On utilise plutôt cet indicateur car plus simple à calculer que les expressions du maximum de vraisemblance.

$$Deviance = -2\ln\left(\frac{\text{Vraisemblance sans la variable}}{\text{Vraisemblance avec la variable}}\right)$$

On a sous $\mathcal{H}_0$:

$$2\left(\ell\ell_{\widehat{\beta}} - \ell\ell_{\widehat{\beta^0}}\right) \longrightarrow \chi^2(p)$$

Avec $\ell\ell_{\widehat{\beta}}$ la log-vraisemblance du modèle avec la variable, et $\ell\ell_{\widehat{\beta^0}}$ la log vraisemblance du modèle sans la variable.

Il existe des fonctions dans Rstudio rendant le /les modèles minimisant la déviance, il n'est donc pas nécessaire de créer les modèles au préalable et de les comparer entre eux, le logiciel fait ce travail à notre place.

2.2 Méthodes automatiques : critères AIC et BIC

L'approche du test de Wald et test du rapport de vraisemblance permet de choisir un modèle parmi deux modèles emboités. Cependant ces tests ne permettent pas de sélectionner automatiquement un sous groupe de variables explicatives.

Pour des modèles ayant un nombre de paramètres égal, l'algorithme utilisera la vraisemblance pour choisir le meilleur modèle à k variables. Cependant la vraisemblance ne pourra pas être utilisée quand le nombre de paramètres sera différent pour des modèles. En effet la vraisemblance augmente avec le nombre de paramètres, ainsi le modèle choisi sera celui avec le plus grand nombre de paramètres.

Pour pallier à cela des critères existent. Parmi les critères les plus utilisés, on retrouve, comme pour les modèles linéaires l'AIC et le BIC. Ces critères pénalisent l'opposé de la log-vraisemblance d'un modèle $\mathcal{M}$ par son nombre de paramètres k.

AIC (*Akaike Information criterion*)

$$AIC = -2\ell\ell(\widehat{\beta}) + 2k$$

BIC (*Bayesian information criterion*)

$$BIC = -2\ell\ell(\widehat{\beta}) + \ln(n)k$$

avec $\ell\ell(\widehat{\beta})$ qui désigne la log-vraisemblance maximisée du modèle logistique $\mathcal{M}$,ces critères sont basés sur deux parties :

- la composante $-2\ell\ell(\widehat{\beta})$ mesure l'ajustement du modèle aux données. Plus les valeurs sont faibles plus l'ajustement est bon.
- les composantes $2k$ pour l'AIC et $k\ln(n)$ pour le BIC mesurent la complexité du modèle

Ces critères à minimiser sélectionneront les modèles qui réalisent un bon compromis entre qualité d'ajustement et complexité.

Remarque 1. Le critère BIC aura tendance à choisir des modèles plus parcimonieux que le critère AIC. Cela arrive quand $\ln(n) > 2$ soit dès que le modèle a 8 observations ou plus.

Remarque 2. Les fonctions R correspondant à ces deux critères sont :

- `bestglm` : qui utilisera l'algorithme de *Best subset selection* on peut choisir quel critère nous voulons utiliser pour trouver le meilleur modèle (AIC, BIC, EBIC, CV pour la validation croisée, $\cdots$)
- `step` : fonction qui trouve le modèle minimisant l'AIC. Elle permet de lancer les procédures pas à pas. En effet l'algorithme nécessitant le calcul de 2^p modèles devient coûteux en temps de calcul lorsque le nombre de variables p est grand (au delà de 30).

2.3 Le lambda de Wilks

Cet indicateur est propre à l'analyse discriminante et n'est pas utilisé en régression logistique. Le Lambda de Wilks est souvent utilisé dans les logiciels comme critère pour ne garder que les variables apportant de l'information sur l'appartenance ou non d'un individu à un groupe.

Le Lambda de Wilks est une approche paramétrique permettant de tester si plusieurs variables continues distinctes $X = (X_1, \cdots, X_p)$ sont liées à une variable qualitative Y à $K \geq 2$ groupes, lorsqu'elles sont considérées avec leurs différentes interactions multivariées.

Les hypothèses d'utilisation de ce test sont : $X_{|Y=1}, \cdots, X_{|Y=k}$ suivent une loi normale et leur matrice de covariance respective sont égales (homoscédasticité).

La statistique du test du Lambda de Wilks se définie de la manière suivante :

$$\Lambda = \frac{\det(W)}{\det(B)}$$

Où W est la matrice de variance-covariance intragroupe et B la matrice de variance-covariance intergroupe.

Cette statistique de test suit une loi de Wilks à $(P, n, K-1)$ degrés de liberté et l'hypothèse H_0 est : « Indépendance entre X et $Y_{|\mu_1=\cdots=\mu_k}$ ».

Une variable a un bon pouvoir discriminant si la dispersion intra-groupe est faible et si la dispersion intergroupe est forte. Donc plus le Lambda de Wilks sera faible, plus la variable considérée est discriminante. C'est ce critère qu'utilise la commande `greedy.wilks` de Rstudio que nous avons utilisée pour trouver les

variables les plus discriminantes dans notre jeu de données et ainsi se focaliser sur un nombre de modèles plus réduit.

3 Qualité du modèle

3.1 Matrices de confusion

L'analyse factorielle discriminante probabiliste et la régression logistique sont utilisées pour affecter des scores à des individus. Mais donner un score à un individu sans contexte d'étude est assez absurde. C'est pourquoi, il faut évaluer le modèle choisi pour savoir si celui-ci permet une bonne prédiction des groupes d'appartenance.

Il est coutume de tester la qualité du modèle grâce aux matrices de confusions qui donnent le taux de bon et mauvais classement des individus dans chaque groupe. Nous construisons donc souvent plusieurs modèles et évaluons leurs capacités prédictives grâce à ces matrices.

Dans le cas de deux groupes les matrices de confusions se représentent souvent de la façon suivante :

	$Y = 0$	$Y = 1$
$Y_{pred} = 0$	VN	FN
$Y_{pred} = 1$	FP	VP

Avec :
— 0 = négatif
— 1 = positif
— VN = vrai négatif
— FP = faux positif
— FN = faux négatif
— VP = vrai positif

En fonction de ce que l'on cherche à faire grâce au scoring, on préfèrera retenir le modèle minimisant les faux négatifs ou les faux positifs, ou encore le modèle maximisant les vrais positifs ou vrais négatifs.

3.2 Courbes ROC et AUC

Une fois notre modèle choisi grâce aux comparaisons des différentes matrices de confusion, il est possible de visualiser graphiquement la qualité globale de ce modèle grâce à une courbe ROC.

Courbe ROC Une courbe ROC (*Receiver Operating Curve*) est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs :

— Le taux de vrais positifs (TVP) (sensibilité) est défini comme : $TVP = \dfrac{VP}{VP + FN}$

— Le taux de faux positifs (TFP) (spécificité) est défini comme : $TFP = \dfrac{VN}{VN + FP}$

.

Les termes "positifs" et "négatifs" dépendent de ce que l'on aura choisi au préalable.

La *sensibilité* se définit comme le pourcentage de vrais positifs : $1 - \beta$: D'un point de vue médical cela veut dire être testé positif à un test détectant la présence de maladie quand on est bien malade .

La *spécificité* se définit quant à elle comme le pourcentage de vrais négatifs : $1 - \alpha$. D'un point de vue médical cela signifie être testé négatif à un test détectant la présence de maladie, quand on est bien sain.

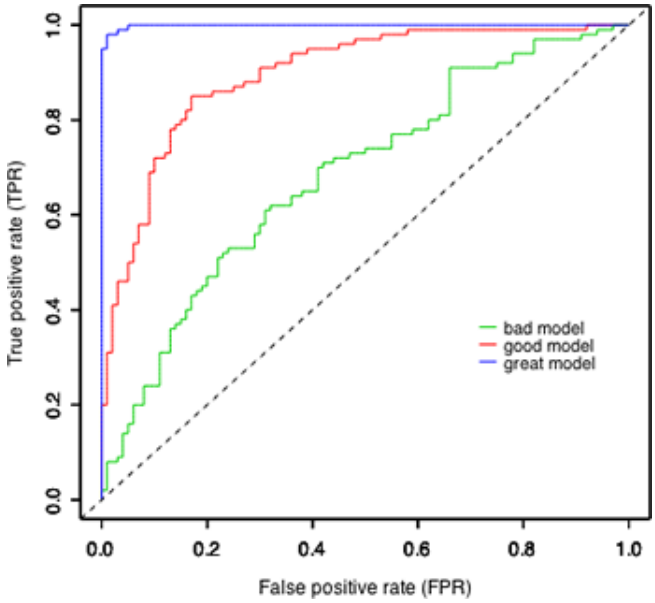


FIGURE 1 – Exemple de courbes ROC (Source : *The University of North California at Chapell Hill*)

Ainsi la courbe ROC représente le taux de vrais positifs (TVP) par rapport au taux de faux positifs (TFP). Ainsi elle représente la sensibilité sur 1-spécificité.

AUC

3.3 Histogrammes de scores

Le seuil utilisé dans l'algorithme de recherche du meilleur modèle (pour la problématique considérée) est par défaut fixé à 0.5 (dans le cas de 2 groupes) pour décider du groupe d'appartenance. Or toute l'essence du scoring est justement de trouver le seuil qui donnera un risque d'erreur le plus faible possible, tout en prenant les contraintes de coût financier en compte.

C'est pourquoi il est courant de représenter les histogrammes de score pour déterminer le seuil optimal.

On trace les histogrammes des scores des individus en fonction de leur vrai groupe d'appartenance. Dans le cas où les 2 histogrammes sont disjoints, alors il est possible de trouver un seuil qui annulera l'erreur prise, mais ce cas est plutôt rare. Les histogrammes sont toujours plus ont moins superposés et c'est donc le travail de l'analyste de choisir le seuil qui minimisera le plus possible le taux d'erreur pris, tout en prenant en compte encore une fois toutes les contraintes financières ou matérielles.

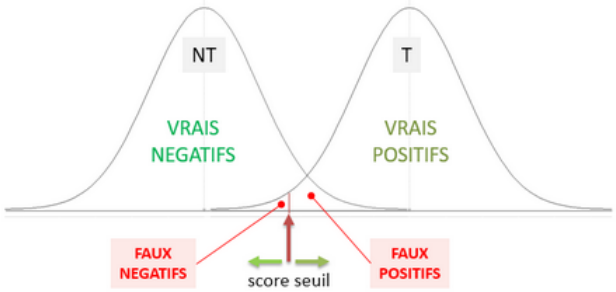


FIGURE 2 – Exemple d'histogrammes de scores (Source : Cours de Psychométrie de Mr Jean-Luc Roulin)

Deuxième partie

Régression logistique

1 Introduction

Les racines de la régression logistique plongent loin dans l'histoire de l'analyse des données. En effet c'est vers 1840 que P.F Verhulst introduit ce qu'il appelle "équation logistique" pour répondre à une problématique de dynamique des populations . La régression logistique consiste à expliquer une variable Y (variable cible), par une ou plusieurs variables explicatives X_j(qualitatives ou quantitatives). Cette méthode a été introduite en 1944 par Berkson [1] en biostatistiques.
La méthode de régression logistique est très appréciée pour sa généralité, son interprétabilité et sa robustesse.
La fonction logistique est utilisée dans de nombreux domaines :

- épidémiologie : la diffusion d'une épidémie
- marketing : ventes d'un nouveau produit
- psychologie : pour prédire des comportements
- technologie

Dans ce projet on se concentre au cas où la variable à expliquer est binaire. On suppose qu'il y a donc deux groupes à discriminer. Ainsi la variable à expliquer Y prend deux modalités 0 ou 1.
Quand le nombre de modalités de la variable à expliquer est supérieur à 2 on parle de régression logistique *polytomique* (scrutin a plus de deux candidats, degrés de satisfaction pour un produit, mention a un examen....)

Les avantages de cette méthode sont qu'il n'y a pas besoin d'hypothèses de multinormalité. On se place sous les hypothèses de normalité et égalité des matrices de variance covariance.

Notations On note :

- Y la variable à expliquer est à valeurs dans $\{0 \quad 1\}$.
- $X = \begin{pmatrix} 1 \\ X_1 \\ \vdots \\ X_p \end{pmatrix} \in \mathbb{R}^{p+1}$ un vecteur de variables explicatives $X_j \quad \forall j \in [\![1, p]\!]$
- Le vecteur des coefficients $\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \in \mathbb{R}^{p+1}$ à estimer par maximum de vraisemblance.
- $x = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_p \end{pmatrix} \in \mathbb{R}^{p+1}$ une réalisation de X, y est une réalisation de Y, y_i suit une loi de Bernoulli de paramètre $\pi_\beta(x_i)$.

1. Joseph BERKSON (1899,1982) était un physicien, médecin statisticien américain. Il a introduit la notion de régression logistique dans son article *Are there two regressions ? (1950)*

— $(X_1, Y_1), \cdots (X_n, Y_n)$ est un n-échantillon aléatoire et de même loi que le couple (X, Y)

— $(x_1, y_1), \cdots, (x_n, y_n)$ une réalisation de $(X_1, Y_1) \cdots (X_n, Y_n)$

L'objectif de la régression logistique est de modéliser l'espérance conditionnelle de Y par rapport à X : $\mathbb{E}[Y \mid X = x]$.

En régression linéaire, on a :

$$\mathbb{E}[Y \mid X = x] = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Ceci ne convient pas lorsque Y est binaire (0 ou 1) puisque le terme ci dessus est non borné alors que $\mathbb{P}(Y \mid X = x)$ est dans l'intervalle $[0, 1]$. On a alors quand Y est binaire (0 ou 1) :

$$\mathbb{E}[Y \mid X = x] = 1 \times \mathbb{P}(Y = 1 \mid X = x) + 0 \times \mathbb{P}(Y = 0 \mid X = x)$$

Ainsi :

$$\begin{aligned} \mathbb{E}[Y \mid X = x] &= \mathbb{P}(Y = 1 \mid X = x) \\ &= f_\beta(x) \end{aligned}$$

Cette expression est la probabilité a posteriori d'appartenir au premier groupe. La fonction $f_\beta(x)$ est appelée *fonction de transfert*.

Exemple avec un cas simple (une variable explicative) On va tenter d'expliquer la présence de maladie cardiovasculaire par une seule variable explicative : l'âge du patient.

Ici on va donc expliquer la variable `CHD` (0 si le patient est sain, 1 sinon) par la variable `AGE`. On dispose de 100 individus.

```
groupe_age = tapply(AGE,AGRP,mean)


ggplot(cardio, aes(x = AGE, y = CHD))
+ geom_point(aes(color=CHD))
+ labs(title = "Pr sence de maladie cardiovasculaire selon l' ge  du patient", x
    = "  ge  du patient", y="Pr sence de maladie")


ggplot(data_frame, aes(x= groupe_age,y = CHD_groupe))
+ geom_point(size = 2, aes(color=CHD_groupe))
+ labs(title = "Proportion de malades selon les groupes d' ge ", x = "Groupes d'
    ge ", y="Proportion de malades") + ylim(0, 1)
+ geom_line()
```

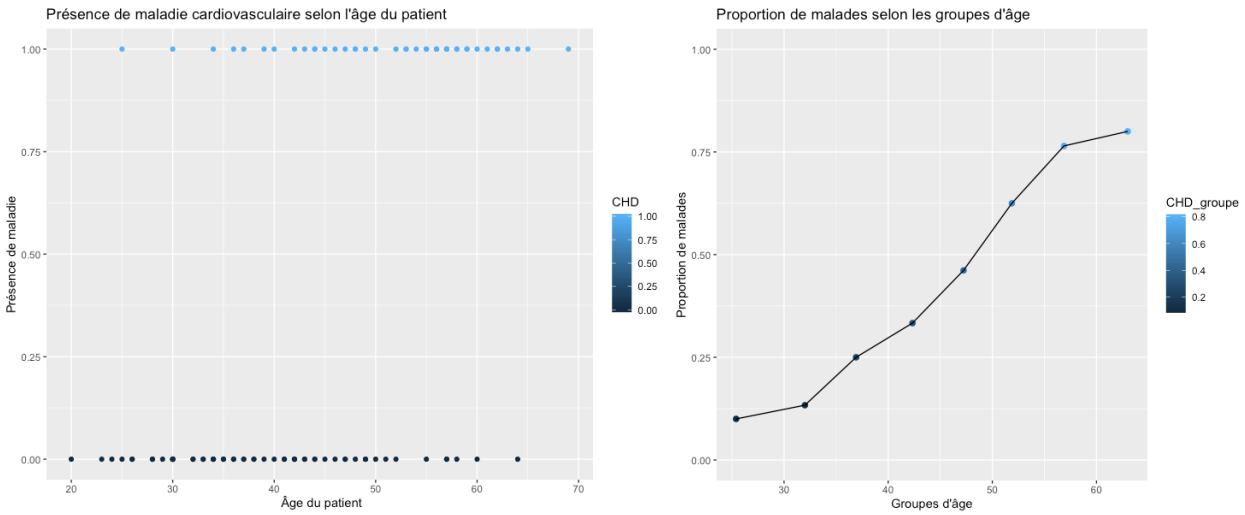


FIGURE 3 – Représentation des individus à gauche, proportion de malades selon des groupes d'âge à droite

On peut apercevoir avec le premier graphique de gauche qu'il est difficile de modéliser les données à cause de la variabilité de la variable CHD. Ainsi nous regroupons les individus par classes d'âge prédéfinies.

On remarque que la liaison entre les deux variables est plus claire sur le second graphique (à droite) grâce à cette répartition par classes d'âges. En effet plus l'âge augmente plus le risque de contracter une maladie cardiovasculaire est élevé. On remarque par ailleurs que la forme suit une courbe sigmoïde en forme de "S".

C'est pour cela qu'on va utiliser une fonction Dans la régression logistique la fonction de transfert est la fonction $\pi_\beta(x)$

$$\mathbb{P}(G1|x) = \mathbb{P}(Y = 1|X = x) = \pi_\beta(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

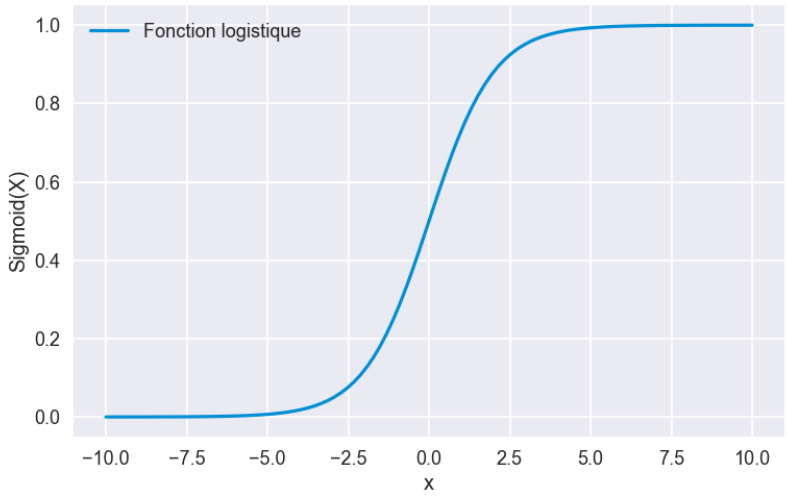


FIGURE 4 – Représentation graphique de la fonction logistique sur Python - Sigmoïde

Définition 1.1

La fonction $\pi(x)$ est appelée *fonction logistique*. Sa représentation graphique est une sigmoïde en fonction des modalités de x. La fonction $\pi(x)$ est comprise dans $]0,1[$, elle convient donc à une probabilité et donne souvent une bonne représentation des phénomènes.

$$\begin{aligned} \pi_\beta(x) : \quad & \mathbb{R}^{p+1} \longrightarrow]0,1[\\ & x \longmapsto \pi_\beta(x) = \frac{e^{\beta^T x}}{1+e^{\beta^T x}} \end{aligned}$$

On cherche à écrire l'espérance conditionnelle de la variable à expliquer Y comme combinaison linéaire de variables à expliquer X. On veut modéliser l'espérance conditionnelle $\mathbb{E}[Y|X=x]$.
On cherche la valeur moyenne de Y pour toute valeurs de X.

Définition 1.2

Soit Y une variable à valeurs dans $\{0,1\}$ à expliquer par p variables explicatives $X_1, \cdots, X_p$.
Le modèle logistique propose une modélisation de la loi de $Y \mid X = x$ par une loi de Bernoulli de paramètre $\pi_\beta(x) = \mathbb{P}_\beta(Y = 1 \mid X = x)$ telle que :

$$\ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

ou encore

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

La fonction logit est appelée *fonction de lien*, elle est bijective, dérivable de $]0,1[$ dans $\mathbb{R}$.

Utilisée avec la fonction de logarithme néperien, logit est la réciproque de $f(x) = \dfrac{1}{1+e^{-x}}$ qui est utilisée pour linéariser les fonctions logistiques.

1.1 Interprétation avec des Odds-ratio

L'odds ratio permet de mesurer l'effet d'un facteur. L'odds ratio d'une variable explicative mesure lorsque X_j passe de x à $x+1$ toutes variables étant égales par ailleurs.

$$Odds = \frac{\pi(x)}{1-\pi(x)}$$

On définit l'Odds ratio comme :

$$OR = \frac{\frac{\pi(x+1)}{1-\pi(x+1)}}{\frac{\pi(x)}{1-\pi(x)}} = e^{\beta_i}$$

— Si $\beta \leq 0 \iff OR < 1$ cela indique que la variable explicative a une influence négative sur la variable à prédire.

— Si $\beta \geq 0 \iff OR > 1$ cela indique que la variable explicative a une influence positive sur la variable à prédire.

Quand la variable explicative X_j est binaire, on a :

$$Odds = \frac{\mathbb{P}(Y = 1 \mid X_j = 1)}{\mathbb{P}(Y = 0 \mid X_j = 1)}$$

On obtient un seul odds ratio qui est :

$$OR = \frac{\dfrac{\mathbb{P}(Y = 1 \mid X_j = 1)}{\mathbb{P}(Y = 0 \mid X_j = 1)}}{\dfrac{\mathbb{P}(Y = 1 \mid X_j = 0)}{\mathbb{P}(Y = 0 \mid X_j = 0)}} = e^{\beta_j}$$

C'est le facteur par lequel on multiplie la côte lorsque x passe de 0 à 1.

Retour sur l'exemple introductif .

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

```
Call:
glm(formula = CHD ~ AGE, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9718  -0.8456  -0.4576   0.8253   2.2859

Coefficients:
            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 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    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
```

FIGURE 5 – Sortie R de la fonction glm

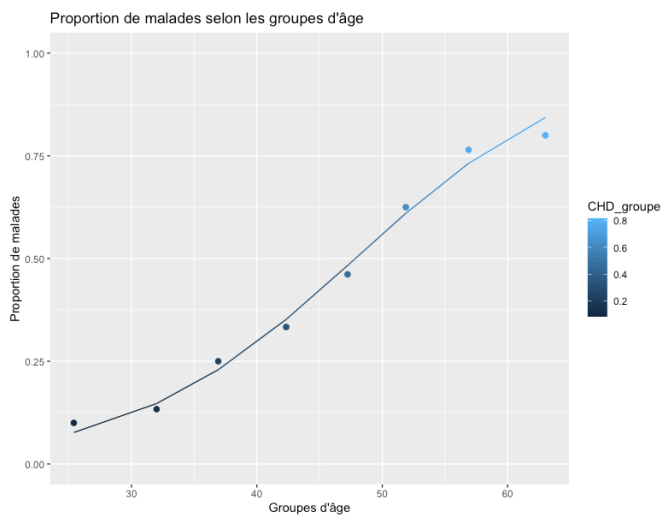
On peut lire les coefficients $\beta_0 = -5.30945$ et $\beta_1 = 0.11092$. On va utiliser ces coefficients pour tracer la sigmoïde représentant la proportion de malades selon les âges :

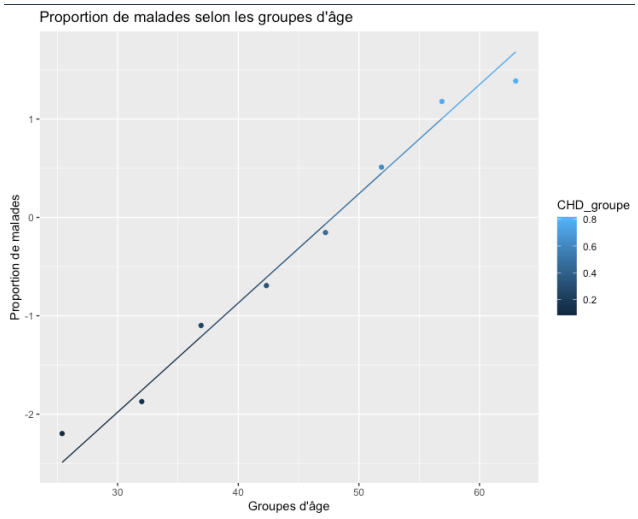
On obtient la fonction logistique suivante :

$$\pi_\beta(x) = \frac{e^{-5.31+0.11x}}{1 + e^{-5.31+0.11x}}$$

```
ggplot(data_frame, aes(x= groupe_age,y = CHD_groupe))
+ geom_point(size = 2, aes(color=CHD_groupe))
+ labs(title = "Proportion de malades selon les groupes d' ge ", x = "Groupes d'
    ge ", y="Proportion de malades") + ylim(0, 1)
+ geom_line(aes( x = groupe_age, y = exp(-5.31+0.111*groupe_age)/(1 + exp(-5.31 +
    0.111*groupe_age)), color = CHD_groupe))

ggplot(data_frame,aes(x= groupe_age,y = CHD_groupe)) + geom_point(aes(x = groupe_
    age, y = log(CHD_groupe/(1-CHD_groupe)), color = CHD_groupe)) + geom_line(aes(
    x = groupe_age, y = -5.31+0.111*groupe_age, color = CHD_groupe)) + labs(title
    = "Proportion de malades selon les groupes d' ge ", x = "Groupes d' ge ", y="
    Proportion de malades")
```





Les coefficients s'interprètent comme des logarithmes népériens d'odds ratio. Ainsi on a :

$$OR = e^{\beta_1} = e^{0.1109} = 1.12$$

Alors quand un patient vieillit d'un an, son risque de contracter une maladie cardiovasculaire est multiplié par 1.12.

2 Estimation des paramètres

Pour estimer le vecteur de paramètres β on utilise la méthode de maximum de vraisemblance à partir d'un échantillon *iid* de n observations. En effet la variable Y à expliquer étant qualitative, on ne peut pas utiliser la méthode d'estimation par les moindres carrés habituelle.

La vraisemblance La vraisemblance pour une observation (y_i, x_i) peut s'écrire :

$$\ell(\beta; y_i, x_i) = \pi_\beta(x_i)^{y_i} \times (1 - \pi_\beta(x_i))^{1-y_i}$$

Comme les observations sont *iid* on peut écrire que la vraisemblance du n-échantillon est égale au produit des vraisemblances par observation :

$$\ell(\beta; y, x) = \prod_{i=1}^{p} \pi_\beta(x_i)^{y_i} \times (1 - \pi_\beta(x_i))^{1-y_i}$$

La log vraisemblance

Proposition 2.1

La log vraisemblance s'écrit

$$\beta \longrightarrow \ell\ell_X(\beta; y, x) = \sum_{i=1}^{n} y_i \ln\left(\pi_\beta(x_i)\right) + (1 - y_i) \ln\left(1 - \pi_\beta(x_i)\right)$$

Démonstration. La vraisemblance s'écrit :

$$\ell(\beta; y, x) = \prod_{i=1}^{n} \pi_\beta(x_i)^{y_i} \times (1 - \pi_\beta(x_i))^{1-y_i}$$

Or $\pi_\beta(x_i) \in]0, 1[$. Donc la vraisemblance est strictement positive, on peut calculer la log vraisemblance.

$$\begin{aligned}
\ell\ell(\beta) = \ln \ell(\beta) &= \sum_{i=1}^{n} \ln\left(\mathbb{P}\left(Y = y_i \mid X = x_i\right)\right) \\
&= \sum_{i=1}^{n} y_i \ln\left(\pi_\beta(x_i)\right) + (1 - y_i) \ln\left(1 - \pi_\beta(x_i)\right)
\end{aligned}$$

□

Équations de vraisemblance Le vecteur gradient au point β est défini par :

$$\nabla_\beta \ell\ell(\beta) = \begin{pmatrix} \frac{\partial \ell\ell}{\partial \beta_0}(\beta) \\ \vdots \\ \frac{\partial \ell\ell}{\partial \beta_p}(\beta) \end{pmatrix}$$

Calculons $\dfrac{\partial \ell\ell}{\partial \beta_j}(\beta) \quad \forall j \in [\![0, p]\!]$. On a :

$$\begin{aligned}
\ell\ell(\beta) &= \sum_{i=1}^{n} y_i \ln\left(\pi_\beta(x_i)\right) + (1 - y_i) \ln\left(1 - \pi_\beta(x_i)\right) \\
&= \sum_{i=1}^{n} y_i \ln\left(\frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}\right) + (1 - y_i) \ln\left(1 - \frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}\right) \\
&= \sum_{i=1}^{n} y_i \ln\left(\frac{e^{\beta^T x_i}}{1 + e^{\beta^T x_i}}\right) + (1 - y_i) \ln\left(\frac{1}{1 + e^{\beta^T x_i}}\right)
\end{aligned}$$

Avec $\beta x_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \cdots + \beta_p x_{ip}$

$$\begin{aligned}
\frac{\partial}{\partial\beta_j}\left[\ln\left(\frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}\right)\right] &= \frac{x_{ij}e^{\beta x_i}\left(1+e^{\beta^T x_i}\right) - e^{\beta^T x_i}\left(x_{ij}e^{\beta^T x_i}\right)}{\left(1+e^{\beta^T x_i}\right)^2} \times \frac{1+e^{\beta^T x_i}}{e^{\beta^T x_i}} \\
&= \frac{x_{ij}e^{\beta^T x_i}}{\left(1+e^{\beta^T x_i}\right)^2} \times \frac{1+e^{\beta^T x_i}}{e^{\beta^T x_i}} \\
&= \frac{x_{ij}}{1+e^{\beta^T x_i}} \\
\frac{\partial}{\partial\beta_j}\left[\ln\left(\frac{1}{1+e^{\beta^T x_i}}\right)\right] &= -\frac{x_{ij}e^{\beta^T x_i}}{\left(1+e^{\beta^T x_i}\right)^2} \times \left(1+e^{\beta^T x_i}\right) \\
&= -\frac{x_{ij}e^{\beta^T x_i}}{1+e^{\beta^T x_i}}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\ell\ell}{\partial\beta_j}(\beta) &= \sum_{i=1}^{n} y_i \frac{x_{ij}}{1+e^{\beta^T x_i}} - (1-y_i)\frac{x_{ij}e^{\beta^T x_i}}{1+e^{\beta^T x_i}} \\
&= \sum_{i=1}^{n} \frac{x_{ij}\left(y_i - e^{\beta^T x_i} + y_i e^{\beta^T x_i}\right)}{1+e^{\beta^T x_i}} \\
&= \sum_{i=1}^{n} x_{ij}\frac{y_i\left(1+e^{\beta^T x_i}\right) - e^{\beta^T x_i}}{1+e^{\beta^T x_i}} \\
&= \sum_{i=1}^{n} x_{ij}\left(y_i - \frac{e^{\beta^T x_i}}{1+e^{\beta^T x_i}}\right) \\
&= \sum_{i=1}^{n} x_{ij}\left(y_i - \pi_\beta(x_i)\right) \qquad \forall j \in [\![0,p]\!] \text{ avec } x_{i0}=1
\end{aligned}$$

On obtient l'écriture générale :

$$\nabla_\beta \ell\ell(\beta) = \sum_{i=1}^{n} x_i(y_i - \pi_\beta(x_i)) \qquad \forall i \in [\![0,n]\!] \text{ avec } x_0 = 1$$

On peut également l'écrire sous forme matricielle :

$$X^T(Y - \Pi_\beta)$$

avec $X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n\times(p+1)}$ $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$ et $\Pi_\beta = \begin{pmatrix} \pi_\beta(x_1) \\ \vdots \\ \pi_\beta(x_n) \end{pmatrix} \in \mathbb{R}^n$

Recherche d'estimateur du maximum de vraisemblance :
Si l'estimateur de maximum de vraisemblance $\widehat{\beta}$ existe, il est solution de l'équation :

$$X^T (Y - \Pi_\beta) = 0$$

Ainsi rechercher les solutions de cette équation revient à résoudre $p+1$ équations à $p+1$ inconnues $(\beta_0, \beta_1, \cdots, \beta_p)$:

$$\begin{cases} y_1 + \cdots + y_n = \pi_\beta(x_1) + \cdots + \pi_\beta(x_n) & j = 0 \\ x_{1j} y_1 + \cdots + x_{nj} y_n = x_{1j}\pi_\beta(x_1) + \cdots + x_{nj}\pi_\beta(x_n), & \forall j \in [\![1, p]\!] \end{cases}$$

$$\iff \begin{cases} y_1 + \cdots + y_n = \dfrac{e^{\beta^T x_1}}{1 + e^{\beta^T x_1}} + \cdots + \dfrac{e^{\beta^T x_n}}{1 + e^{\beta^T x_n}} & j = 0 \\ x_{1j} y_1 + \cdots + x_{nj} y_n = x_{1j} \dfrac{e^{\beta^T x_1}}{1 + e^{\beta^T x_1}} + \cdots x_{nj} \dfrac{e^{\beta^T x_n}}{1 + e^{\beta^T x_n}}, & \forall j \in [\![1, p]\!] \end{cases}$$

Ce système d'équations n'a pas de solution analytique et se résout par des procédures de calcul numérique (Newton Raphson, algorithme IRLS, ...).

Théorème 2.1

Si X est de rang maximal, la log vraisemblance $\beta \longmapsto \ell\ell(\beta)$ est strictement concave : si $\widehat{\beta}$ existe il est unique.

Démonstration. Calculons la matrice Hessienne de la log vraisemblance

$$\nabla^2_\beta \ell\ell(\beta; y, x) = \left(\frac{\partial^2 \ell\ell}{\partial \beta_i \partial \beta_j}(\beta; y, x) \right)_{1 \le i,j \le p}$$

$$\begin{aligned} \nabla^2_\beta \ell\ell(\beta; y, x) &= \nabla_\beta \left(\nabla_\beta \ell\ell(\beta; y, x) \right) \\ &= \nabla_\beta \left(\sum_{i=1}^n x_i (y_i - \pi_\beta(x_i)) \right) \\ &= -\sum_{i=1}^n x_i^T \nabla_\beta(\pi_\beta(x_i)) \\ &= -\sum_{i=1}^n x_i^T \frac{x_i e^{\beta^T x_i}}{\left(1 + e^{\beta^T x_i}\right)^2} \\ &= -\sum_{i=1}^n x_i^T x_i \frac{e^{\beta^T x_i}}{\left(1 + e^{\beta^T x_i}\right)^2} \\ &= -\sum_{i=1}^n x_i^T x_i \pi_\beta(x_i)(1 - \pi_\beta(x_i)) \end{aligned}$$

Or $\pi_\beta(x_i)(1 - \pi_\beta(x_i)) > 0$ car $\pi_\beta(x_i) \in]0, 1[$.

De plus $x_i^T x_i = \|x_i\|^2$ donc $\|x_i\|^2 \geq 0$ et $\|x_i\|^2 = 0$ pour $x_i = 0$.

Sous forme matricielle on a : Pour alléger les notations on va poser $\pi_i = \pi_\beta(x_i)$

$$H(\beta; Y, X) = \nabla^2_\beta \ell\ell(\beta; Y, X) = -\begin{pmatrix} \sum_{i=1}^n \pi_i(1-\pi_i) & \sum_{i=1}^n x_{i1}\pi_i(1-\pi_i) & \cdots & \sum_{i=1}^n x_{ip}\pi_i(1-\pi_i) \\ \sum_{i=1}^n x_{i1}\pi_i(1-\pi_i) & \sum_{i=1}^n (x_{i1})^2\pi_i(1-\pi_i) & \cdots & \sum_{i=1}^n x_{i1}x_{ip}\pi_i(1-\pi_i) \\ \vdots & & \ddots & \vdots \\ \sum_{i=1}^n x_{ip}\pi_i(1-\pi_i) & \cdots & \cdots & \sum_{i=1}^n (x_{ip})^2\pi_i(1-\pi_i) \end{pmatrix}$$

$$= -\begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}^T \begin{pmatrix} \pi_1(1-\pi_1) & & 0 \\ & \ddots & \\ 0 & & \pi_n(1-\pi_n) \end{pmatrix} \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$= -X^T \Delta_\beta X$$

Δ_β est une matrice $n \times n$ diagonale où le k-ième terme est égal à $\pi_k(1-\pi_k) > 0$.
De plus si X est de rang maximal ($rg(X) = p+1$) alors X est injective et la matrice Δ_β est définie positive.

Ainsi la matrice hessienne de la log vraisemblance est définie négative, alors la log vraisemblance est strictement concave par rapport à β. Ceci garantit, s'il existe, l'unicité du maximum de cette fonction. Ainsi quel que soit le choix des conditions initiales ou de l'algorithme utilisé, les estimateurs du maximum de vraisemblance convergeront vers la vraie valeur $\widehat{\beta}$. □

L'algorithme le plus souvent utilisé pour le calcul de cet estimateur est l'algorithme de Newton Raphson. Enfin puisque on utilise l'estimateur du maximum de vraisemblance, il est possible de construire des intervalles de confiance asymptotiques utiles aux tests pour la sélection de variables.

3 Tests et sélection des variables

La théorie du maximum de vraisemblance nous donne la loi asymptotique des estimateurs : il est donc possible de tester la significativité des variables explicatives : Trois tests sont généralement utilisés :

— Le test de Wald
— Le test du rapport des vraisemblances ou déviance
— le test du score

4 Des Exemples avec R

4.1 Fonction GLM

On utilise avec le logiciel R la fonction `glm` modèle linéaire généralisé. L'approche de la fonction GLM consiste à :

— choisir une loi pour $Y \mid X = x$
— choisir une fonction de lien $g_\beta(x)$ bijective et dérivable
— réaliser une transformation de l'espérance conditionnelle $\mathbb{E}[Y \mid X = x]$ par la fonction g_β pour obtenir notre fonction en sigmoïde :

$$g_\beta\left(\mathbb{E}[Y \mid X = x]\right) = f_\beta(x) = x^T\beta$$

Pour estimer les coefficients $\beta_0, \cdots \beta_p$ la fonction `glm` utilisera l'algorithme de Newton Raphson.

Choix	logistique	log-linéaire	linéaire
loi de $Y \mid X = x$	Bernoulli	Poisson	Normale
modélisation de $\mathbb{E}[Y \mid X = x]$	$\text{logit}\,\mathbb{E}[Y \mid X = x] = x^T\beta$	$\log \mathbb{E}[Y \mid X = x] = x^T\beta$	$\mathrm{E}[\ \mathrm{Y} \mid X = x] = x^T\beta$

4.2 Exemple médical sur les données diabète

Cette base de données contient des observations de 768 individus. Tous les patients ici sont des femmes d'au moins 21 ans d'origine indienne Pima. Le peuple Pima est connu pour être une des communautés comportant le plus grand pourcentage d'obèses et de diabétiques au monde, et à ce titre est un sujet d'études pour les scientifiques.
L'objectif est de prédire si la patiente est diabétique ou non.
Le but est d'expliquer la variable Y ici `Outcome` par les variables explicatives quantitatives suivantes :

— X_1 = `Pregnancies` : Nombre de grossesses de la patiente.
— X_2 = `Glucose` : Concentration de glucose plasmatique après 2 heures par un test de tolérance au glucose par voie orale.
— X_3 = `BloodPressure` : Pression artérielle diastolique (mm Hg)
— X_4 = `SkinThickness` : Épaisseur du pli cutané au niveau du triceps (mm)
— X_5 = `Insulin` : mesure de l'insuline 2h après une injection d'insuline (mu U/ml)
— X_6 = `BMI` : Indice de masse corporelle : $\frac{\text{poids en kg}}{(\text{ taille en m })^2}$. Les personnes ayant un indice de masse corporelle élevé ont plus de risque de contracter un diabète de type II.
— X_7 = `DiabetesPedigreeFunction` : score qui représente la probabilité d'être diabétique selon les antécédents familiaux. Des facteurs génétiques peuvent engendrer un diabète.
— X_8 = `Age` : âge de la patiente au moment du diagnostic. Le diabète de type II a tendance à se manifester quand l'individu est âgé, alors que le diabète de type I concerne les individus plus jeunes.

Analyse du jeu de données .

```
1 > head(diabete)
```

```
  Pregnancies Glucose BloodPressure SkinThickness Insulin  BMI
1           6     148            72            35       0 33.6
2           1      85            66            29       0 26.6
3           8     183            64             0       0 23.3
4           1      89            66            23      94 28.1
5           0     137            40            35     168 43.1
6           5     116            74             0       0 25.6
  DiabetesPedigreeFunction Age Outcome
1                    0.627  50       1
2                    0.351  31       0
3                    0.672  32       1
4                    0.167  21       0
5                    2.288  33       1
6                    0.201  30       0
```

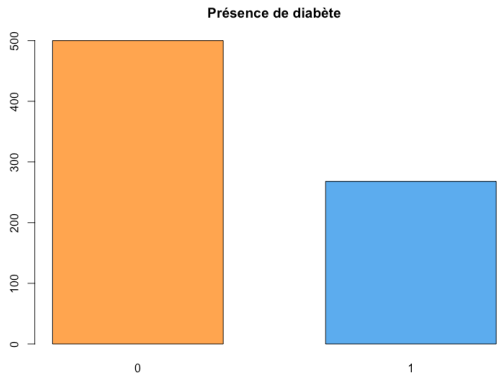


FIGURE 6 – Représentation des 6 premières patientes (à gauche) Présence de diabète ou non chez les patientes (à droite)

On voit que dans cet échantillon 268 sont diabétiques contre 500 non diabétiques

```
 Pregnancies        Glucose      BloodPressure    SkinThickness      Insulin          BMI        DiabetesPedigreeFunction
Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00   Min.   :  0.0   Min.   : 0.00   Min.   :0.0780
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437
Median : 3.000   Median :117.0   Median : 72.00   Median :23.00   Median : 30.5   Median :32.00   Median :0.3725
Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54   Mean   : 79.8   Mean   :31.99   Mean   :0.4719
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262
Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00   Max.   :846.0   Max.   :67.10   Max.   :2.4200
     Age        Outcome
Min.   :21.00   0:500
1st Qu.:24.00   1:268
Median :29.00
Mean   :33.24
3rd Qu.:41.00
Max.   :81.00
```

FIGURE 7 – Variables

On remarque des valeurs parmi ces variables assez atypiques, en effet on peut remarquer que de nombreuses valeurs parmi les variables `Skin Thickness` et `Insulin` (soit l'épaisseur de la peau et le taux d'insuline) sont nulles. C'est ce que l'on remarque en observant le minimum et premier quartile respectif de ces deux variables. Or il est impossible que l'épaisseur de la peau ou le taux d'insuline soient strictement nuls. On en conclut que ces valeurs nulles correspondent à des valeurs manquantes. Représentant plus de 30% du jeu de données, les enlever conduirait à beaucoup d'information perdue. On va donc les traiter.

Traitement des valeurs manquantes .

```
1 list(Column = colSums(diabete==0))
```

```
$Column
            Pregnancies                 Glucose           BloodPressure           SkinThickness
                    111                       5                      35                     227
                    BMI DiabetesPedigreeFunction                     Age                 Outcome
                     11                       0                       0                     500
```

FIGURE 8 – Valeurs manquantes

En observant les autres valeurs de ce tableau on a remarqué que d'autres valeurs parmi les variables

`Glucose`, `BloodPressure` et `BMI` contenaient également des valeurs manquantes : en effet il est impossible d'avoir un IMC, un taux de glucose ou encore une pression artérielle diastolique nulle.

La variable `Pregnancies` contient elle aussi des valeurs nulles cependant nous ne pouvons pas avancer la nature de ces valeurs : représent-t-elles des femmes n'ayant connu aucune grossesse ou cette variable contient-elle, elle aussi des valeurs manquantes ? Il est complexe de répondre à cette question, ainsi nous décidons que ces valeurs nulles représentent l'absence de grossesse.
Ces remarques soulignent le fait qu'il faudra prendre des précautions quant aux conclusions.

On utilise pour traiter ces valeurs manquantes une méthode des plus proches voisins en utilisant l'algorithme KNN (*K nearest neighbors*) et obtenons ainsi un nouveau jeu de données diabète.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
1	6	148	72	35.00000	189.74880	33.6	0.627	50	1
2	1	85	66	29.00000	60.56782	26.6	0.351	31	0
3	8	183	64	30.64939	272.43145	23.3	0.672	32	1
4	1	89	66	23.00000	94.00000	28.1	0.167	21	0
5	0	137	40	35.00000	168.00000	43.1	2.288	33	1
6	5	116	74	21.28276	115.69694	25.6	0.201	30	0

FIGURE 9 – Représentation des 6 premières patientes après traitement des données manquantes

```
 Pregnancies        Glucose      BloodPressure    SkinThickness
Min.   : 0.000   Min.   : 44.0   Min.   : 24.00   Min.   : 7.00
1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.00   1st Qu.:22.00
Median : 3.000   Median :117.0   Median : 72.00   Median :29.00
Mean   : 3.845   Mean   :121.7   Mean   : 72.32   Mean   :29.02
3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:35.00
Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
   Insulin           BMI        DiabetesPedigreeFunction      Age
Min.   : 14.00   Min.   :18.20   Min.   :0.0780           Min.   :21.00
1st Qu.: 89.21   1st Qu.:27.50   1st Qu.:0.2437           1st Qu.:24.00
Median :130.00   Median :32.30   Median :0.3725           Median :29.00
Mean   :152.14   Mean   :32.45   Mean   :0.4719           Mean   :33.24
3rd Qu.:187.28   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
Max.   :846.00   Max.   :67.10   Max.   :2.4200           Max.   :81.00
   Outcome
Min.   :0.000
1st Qu.:0.000
Median :0.000
Mean   :0.349
3rd Qu.:1.000
Max.   :1.000
```

FIGURE 10 – Analyse exploratoire des différentes variables après traitement des données manquantes

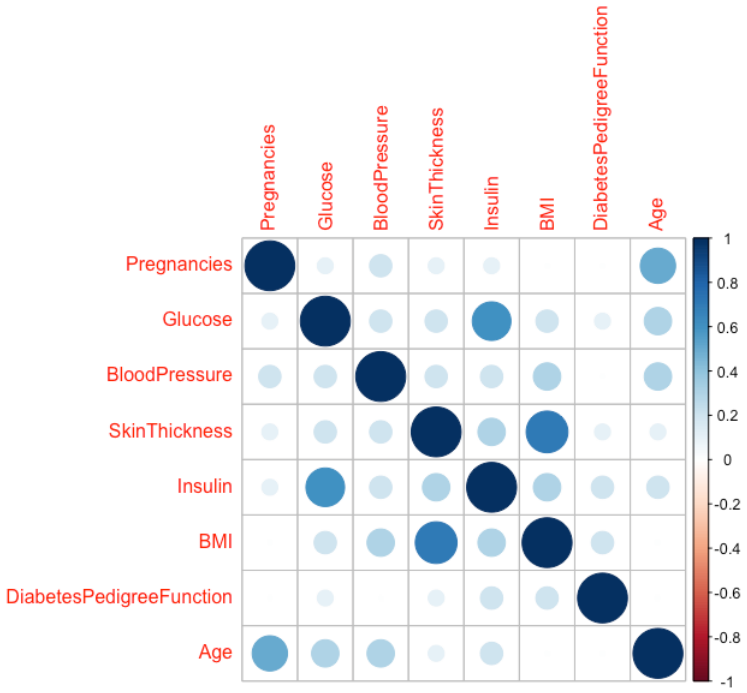


FIGURE 11 – Corrélation entre variables

— La variable `Age` est corrélée positivement à la variable `Pregnancies`. En effet plus une femme a d'enfants plus elle a tendance à être âgée.

— La variable `Insulin` est corrélée positivement à la variable `Glucose`. Cela est vrai car d'une part quand un individu sain est en hyperglycémie (i.e le taux de glucose dans le sang est élevé) des cellules du pancréas libèrent dans le sang l'hormone hypoglycémiante : Insuline. Ainsi plus il y a de glucose dans le sang plus l'hormone d'insuline est libérée. D'autre part un individu souffrant de diabète de type 2 produit bien de l'insuline mais ses cellules développent une résistance à celle ci.

— La variable `BMI` est corrélée positivement à la variable `SkinThickness`. Plus un individu a un indice de masse corporel élevé plus sa peau a tendance à être épaisse.

D'après ce diagramme des corrélations, on remarque que les variables sont en général peu corrélées entre elles. Nous remarquons tout de même les variables `Age` et `Pregnancies` qui sont corrélées entre elles. On prendra compte de cette remarque dans le choix de nos modèles.

4.2.1 Choix des modèles

Modèle 1 : Modèle complet .

```
modele_complet = glm(Outcome ~ .  , data = diabete, family = binomial(link = "
    logit"))
summary(modele_complet)
```

```
Call:
glm(formula = Outcome ~ ., family = binomial(link = "logit"),
    data = diabete)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5566  -0.7274  -0.4159   0.7267   2.9297

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)              -8.4046964  0.7166359 -11.728  < 2e-16 ***
Pregnancies               0.1231823  0.0320776   3.840 0.000123 ***
Glucose                   0.0351637  0.0037087   9.481  < 2e-16 ***
BloodPressure            -0.0132955  0.0052336  -2.540 0.011072 *
SkinThickness             0.0006190  0.0068994   0.090 0.928515
Insulin                  -0.0011917  0.0009012  -1.322 0.186065
BMI                       0.0897010  0.0150876   5.945 2.76e-09 ***
DiabetesPedigreeFunction  0.9451797  0.2991475   3.160 0.001580 **
Age                       0.0148690  0.0093348   1.593 0.111192
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 993.48  on 767  degrees of freedom
Residual deviance: 723.45  on 759  degrees of freedom
AIC: 741.45

Number of Fisher Scoring iterations: 5
```

FIGURE 12 – Variables

Nous décidons à partir de ces résultats de créer un nouveau modèle : ce nouveau modèle comportera les variables les plus significatives au seuil de 5 % soit : `Pregnancies`, `Glucose`, `BloodPressure`, `BMI` et `DiabetesPedigreeFunction`.

Modèle 2 : Variables significatives

$$\text{logit}(\pi_\beta(x)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_6 X_6 + \beta_7 X_7$$

Ce modèle vérifie également plusieurs critères comme le test de Wald et le test de la déviance.

```
Anova(modele_complet, type = 3,test.statistic = "Wald")
Anova(modele_complet, type = 3, test.statistic = "LR")
```

```
Analysis of Deviance Table (Type III tests)

Response: Outcome
                         LR Chisq Df Pr(>Chisq)
Pregnancies                15.233  1  9.505e-05 ***
Glucose                   114.927  1  < 2.2e-16 ***
BloodPressure               6.548  1   0.010502 *
SkinThickness               0.008  1   0.928500
Insulin                     1.742  1   0.186918
BMI                        40.779  1  1.704e-10 ***
DiabetesPedigreeFunction   10.340  1   0.001302 **
Age                         2.522  1   0.112253
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Analysis of Deviance Table (Type III tests)

Response: Outcome
                           Df    Chisq Pr(>Chisq)
(Intercept)                 1 137.5457  < 2.2e-16 ***
Pregnancies                 1  14.7467   0.000123 ***
Glucose                     1  89.8968  < 2.2e-16 ***
BloodPressure               1   6.4537   0.011072 *
SkinThickness               1   0.0080   0.928515
Insulin                     1   1.7485   0.186065
BMI                         1  35.3470  2.759e-09 ***
DiabetesPedigreeFunction    1   9.9829   0.001580 **
Age                         1   2.5372   0.111192
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

FIGURE 13 – Modèle avec les variables utilisant le test de déviance (à gauche) et le test de Wald(à droite)

Modèle 3 : Minimisant le critère AIC

$$\text{logit}(\pi_\beta(x)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7$$

```
bestglm(diabete, family = binomial, IC = "AIC")  # Preg Glucose Bloodpressure
   Insuline  BMI Diabetespedigreedunction  Age
step(modele_complet)
```

```
Morgan-Tatar search since family is non-gaussian.
AIC
BICq equivalent for q in (0.910337349179387, 0.965036759857444)
Best Model:
                            Estimate   Std. Error    z value     Pr(>|z|)
(Intercept)             -8.405136208 0.7167032628 -11.727498 9.214195e-32
Pregnancies              0.123172450 0.0320687734   3.840884 1.225919e-04
Glucose                  0.035112252 0.0036624713   9.587038 9.064975e-22
BloodPressure           -0.013213574 0.0051536754  -2.563913 1.034996e-02
Insulin                 -0.001157035 0.0008141589  -1.421142 1.552755e-01
BMI                      0.090088589 0.0144619078   6.229371 4.683116e-10
DiabetesPedigreeFunction 0.947595358 0.2980062755   3.179783 1.473853e-03
Age                      0.014788838 0.0092896771   1.591965 1.113926e-01
```

FIGURE 14 – Modèle avec les variables minimisant le critère de l'AIC

Les variables retenues pour ce modèle sont : `Pregnancies`, `Glucose`,`BloodPressure`, `Insulin`, `BMI`, `DiabetesPedigreeFunction` et `Age`.

Modèle 4 : Minimisant le critère : BIC

$$\text{logit}(\pi_\beta(x)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_6 X_6 + \beta_7 X_7$$

```
bestglm(diabete, family = binomial, IC = "BIC")
```

```
Morgan-Tatar search since family is non-gaussian.
BIC
BICq equivalent for q in (0.169717182537119, 0.610343314895634)
Best Model:
                            Estimate  Std. Error    z value     Pr(>|z|)
(Intercept)             -8.41585098 0.656907771 -12.811313 1.417163e-37
Pregnancies              0.14192631 0.027105325   5.236104 1.640012e-07
Glucose                  0.03382636 0.003345272  10.111690 4.903090e-24
BMI                      0.07809694 0.013770941   5.671140 1.418501e-08
DiabetesPedigreeFunction 0.90129355 0.291696408   3.089834 2.002682e-03
```

FIGURE 15 – Modèle avec les variables minimisant le critère du BIC

Les variables retenues pour ce modèle sont : `Pregnancies`, `Glucose`, `BMI` et `DiabetesPedigreeFunction`.

```
diabete = read.table('/Users/Nadia/Documents/Maths/DATA_SCIENCES/PROJET_SCORING/
   diabetes.csv', header = TRUE, sep= ',')

attach(diabete)
y = table(Outcome)
col.y = colors()[c(621, 617)]
```

```
barplot(y, main = "Pr sence de diab te", col = col.y, space = 0.6)
#Colonnes supp
Breaks_age = c(min(Age),24,29,41, max(Age))  #d coup s selont les diff rents quartiles
Breaks_preg = c(0,1,3,6, max(Pregnancies))

diabete$Age_classe = cut(Age, breaks = Breaks_age, include.lowest = TRUE)
diabete$Pregnancies_classe = cut(Pregnancies, breaks = Breaks_preg, include.lowest = TRUE)
diabete$sqrtInsulin = (sqrt(Insulin)-1)*2
  #boxcox
# diabete$logInsulin = log(Insulin) pose probleme, les valeurs nulles
modele_complet = glm(Outcome ~ .  , data = diabete, family = binomial(link = "logit"))
Anova(modele_complet, type = 3,test.statistic = "Wald")
Anova(modele_complet, type = 3, test.statistic = "LR")
#nous donne les m mes resultats


summary(modele_complet)
bestglm(diabete)  #critere BIC
bestglm(diabete, IC = "AIC")
mod_3 <- bestglm(diabete, family= binomial)
mod_3$BestModels
diabete
summary(diabete)

diabete[,9] <- factor(diabete[,9])
str(diabete)    # 0: pas de diabete, 1: diabete


db_cor <- round(cor(diabete[1:8]),1)

corrplot(db_cor)   #pas beaucoup de correlation entre les variables
N = 1000
mod_ret = rep(0,N)

scores_A = rep(0,10)


for (k in 1:N){
  sample = sample.split(Outcome, SplitRatio = 0.8)
  train = subset(diabete, sample == TRUE)
  test = subset(diabete, sample == FALSE)

  modele_1 = glm(Outcome ~ . - Age_classe - sqrtInsulin - Pregnancies_classe , data = train, family = binomial(link = "logit"))   #mod le complet "normal"
  modele_2 = glm(Outcome ~ . -Insulin - Age_classe - Pregnancies_classe , data = train, family=binomial(link="logit")) #modele complet avec sqrt(Insulin) au lieu de Insulin
  modele_3 = glm(Outcome ~ Pregnancies + Glucose + BMI + DiabetesPedigreeFunction + BloodPressure + Age, data = train, family = binomial(link = "logit")) # avec les variables significatives
  modele_4 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction , data = train, family = binomial(link = "logit")) #avec les variables les plus significatives
  modele_5 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction + BloodPressure + Pregnancies, data = train, family = binomial(link = "logit")) #variables significatives sans Age car correle avec Pregnancies
  modele_6 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction + BloodPressure + Age, data = train, family = binomial(link = "logit")) # variables significatives sans Pregnancies car correle avec Age
  modele_7 = glm(Outcome ~  . - Age - sqrtInsulin - Pregnancies_classe , data = train, family = binomial(link = "logit"))  #modele complet avec les classes
```

```
    d' ge
  modele_8 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction +
     BloodPressure + Age_classe, data = train, family = binomial(link = "logit"))
      # m me que
  modele_9 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction + Age_classe
     , data = train, family = binomial(link = "logit")) #les plus significatives
     + Age classe ( toiles - toiles )
  modele_10 = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction +
     BloodPressure + Pregnancies_classe, data = train, family = binomial(link = "
     logit"))  #m me que modele 8 avec Pregnancies classe au lieu de Age classe

  liste_modeles = list(modele_1, modele_2, modele_3, modele_4, modele_5, modele_6
     , modele_7, modele_8, modele_9, modele_10)
  n = length(liste_modeles)
  erreur = 1
  j = 0           #num ro du mod le retenu
  A = matrix(0, nrow = 2, ncol=n)
  A[1,]= 1:n

  for (i in 1:n){
    outcome.pred = predict(liste_modeles[[i]], newdata=test, type="response")  #
       rend un score
    erreur_pred = prop.table(table(outcome.pred>0.3, test$Outcome))[2]   #rend le
       taux de faux negatifs
    A[2,i] = erreur_pred
#    if (erreur_pred<erreur){  #on cherche    minimiser les faux n gatifs
    #  erreur<-erreur_pred
    #  j = as.integer(i) #modele i retenu
     # }
  }
  A_tri = A[,order(A[2,], decreasing = FALSE)]
  for (i in 1:n){
    scores_A[A_tri[1,i]] = scores_A[A_tri[1,i]] + (11-i)}
 #mod_ret[k]=j
}
scores_A
#attribuer des scores: matrice deux lignes, une avec les modeles une avec les
   scores

res = as.data.frame(prop.table(table(mod_ret))); res
max_occ = which.max(res$Freq)
cat("Le mod le    retenir est le mod le", max_occ)

"Mod le final"

Reg_fin = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction + BloodPressure
    + Age_classe, data = diabete, family = binomial(link = "logit"))
Reg_fin = glm(Outcome ~ . - Age_classe - sqrtInsulin - Pregnancies_classe , data
   = train, family = binomial(link = "logit"))   #modele 1
Reg_fin = glm(Outcome ~  Glucose + BMI + DiabetesPedigreeFunction + BloodPressure
    + Pregnancies, data = train, family = binomial(link = "logit")) #modele 5
outcome_pred = rep(0,dim(diabete)[1])     #vecteur de 0

for (i in 1: dim(diabete)[1]){
  if (Reg_fin$fitted.values[i] >=0.3){
    outcome_pred[i] = 1
  }   #ce qu'on a pr dit
}
score_pred = filter(data.frame(Reg_fin$fitted.values, outcome_pred), outcome_pred
    == "1")  #score predit, ceuw qu'on a predit comme  tant  malades
score = filter(data.frame(Reg_fin$fitted.values, diabete$Outcome), diabete$
   Outcome == "1")  #vrai score, ceux qui sont
score_1 = filter(data.frame(Reg_fin$fitted.values, diabete$Outcome), diabete$
   Outcome == "1")  #vrai score, ceux qui sont
```

```
score_0 = filter(data.frame(Reg_fin$fitted.values, diabete$Outcome), diabete$
    Outcome == "0")  #vrai score, ceux qui sont
103
105 hgA = hist(score_pred$Reg_fin.fitted.values, breaks=10, plot=F)
hgB1 = hist(score_1$Reg_fin.fitted.values, breaks=20, plot=F)
107 hgB2 = hist(score_0$Reg_fin.fitted.values, breaks=20, plot=F)
109
col_1 = rgb(1,0,0,0.5)
111 col_2 = rgb(0,0,1,0.5)
113 plot(hgB1, col= col_1, freq=FALSE, xlim=c(0,1), ylim= c(0,3))
plot(hgB2, col=col_2, freq=FALSE, xlim=c(0,1), add=T)
115 lines(density(score_0$Reg_fin.fitted.values), lwd=1.5)
lines(density(score_1$Reg_fin.fitted.values), lwd=1.5)
```

4.3 Exemple économique avec des données sur les exploitations fermières

Troisième partie

Analyse Factorielle Discriminante

1 Théorie de la méthode probabiliste

Le but premier de cette méthode est de prédire au mieux les valeurs d'une variable Y qualitative à K modalités, à partir de p variables explicatives $X = (X_1, \cdots, X_p)$ quantitatives.

Dans cette section, nous allons définir des règles de décision bayésiennes qui vont permettre d'affecter un nouvel individu à la classe "la plus probable" et non pas au groupe « le plus proche » comme c'est le cas pour l'analyse discriminante géométrique. Pour cela, il est nécessaire de faire des hypothèses probabilistes sur les données, d'où le nom de la méthode.

On suppose maintenant que les données sont issues d'une population regroupant des individus de K groupes prédéfinis différents $G_1, \cdots, G_k$ et que :

- Y est une variable aléatoire qui prend ses valeurs dans $\{1, \cdots, K\}$
- $X = (X_1, \cdots, X_p)$ est un vecteur de variables aléatoires réelles

On notera :

- $p_k = \mathbb{P}(Y = k)$ la probabilité à priori d'appartenir au groupe G_k.
- $f_k = \mathbb{R}^p \longrightarrow [0, 1]$ la densité de X dans le groupe k
- w_i le poids d'un individu x_i
- $w_k = \dfrac{n_k}{n}$ le poids du groupe G_k où n_k représente le nombre d'individus dans le groupe G_k.
- $g_k = \displaystyle\sum_{i \in I(k)} \frac{w_I}{w_k} x_i$ le centre de gravité du sous nuage formé par les individus du groupe G_k
- $W_k = \displaystyle\sum_{k=1}^{q} w_k W_k$ la matrice de variance covariance du nuage N_k représentant le groupe G_k

On supposera que l'on dispose d'un échantillon i.i.d $(X_1, Y_1), \cdots, (X_n, Y_n)$ de même loi que (X, Y).

1.1 Approche paramétrique : modèle Bayésien

Avec les notations précédentes, on obtient d'après la formule de Bayes que la probabilité qu'un individu x appartienne au groupe G_k s'écrit :

$$\mathbb{P}(G_k \mid x) = \frac{p_k f_k(x \mid k)}{\sum_{u=1}^{K} p_u f_u(x \mid u)}$$

On affecte alors l'individu x au groupe G_k pour lequel la probabilité $\mathbb{P}(G_k \mid x)$ est la plus forte (c'est la règle du minimum a posteriori).

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

La fonction de densité f_k introduite ici peut être déterminée soit :

1. Par des approches non paramétriques : on cherche à estimer directement à partir des données les densités (méthode des noyaux, des plus proches voisins).

2. Par des méthodes paramétriques : on suppose $f_k(x)$ d'une forme paramétrique particulière et on estime les paramètres grâce à l'échantillon d'apprentissage.

1.2 Modèle Bayésien avec méthode paramétrique

On considère dans cette section que X suit une loi normale $\mathcal{N}(\mu_k; \Sigma_k)$ dans chaque groupe G_k. On se place donc dans le cas paramétrique Gaussien.

On a donc

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

où μ_k appartient à $\mathbb{R}^p$ est le vecteur des moyennes théoriques et Σ_k la matrice des variances-covariances théoriques. La règle de classement énoncée précédemment (également appelée règle de Bayes) revient donc à maximiser

$$p_k \times \frac{1}{(2\pi)^{\frac{p}{2}} (\det(\Sigma_k))^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

Par soucis pratique, on préfère maximiser le logarithme de cette expression, c'est-à-dire maximiser :

$$\begin{aligned}\ln(p_k f_k(x)) &= \ln(p_k) + \ln(f_k(x)) \\ &= \ln(p_k) - \frac{p}{2}\ln(2\pi) - \frac{1}{2}\ln(\det(\Sigma_k)) - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\end{aligned}$$

Cela revient à maximiser

$$\begin{aligned}&-(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - p\ln(2\pi) - \ln(\det(\Sigma_k)) + 2\ln(p_k) \\ =&-x^T\Sigma_k^{-1}x + \underbrace{x^T\Sigma_k^{-1}\mu_k + \mu_k^T\Sigma_k^{-1}x}_{=2\mu_k^T\Sigma^{-1}x \text{ car symétrique}} - \mu_k^T\Sigma_k^{-1}\mu_k - p\ln(2\pi) - \ln(\det(\Sigma_k)) + 2\ln(p_k)\end{aligned}$$

Donc avec $\frac{p}{2}\ln(2\pi)$ qui est indépendant de k, maximiser $\ln(p_k f_k(x))$ est équivalent à minimiser :

1.2.1 Analyse discriminante linéaire

Dans le cas où les matrices de variance- covariance Σ_k peuvent être supposées égales, on obtient le critère linéaire en x (d'affectation de l'individu x au groupe G_k) :

$$SL_k(x) = 2\mu_k^T\Sigma^{-1}x - \mu_k^T\Sigma^{-1}\mu_k + 2\ln(p_k)$$

car $\det(\Sigma)$ et $x^T\Sigma^{-1}x$ est constant pour un individu x donné.
On affecte donc x au groupe G_k qui donne une valeur de $SL_k(x)$ maximale.

1.2.2 Analyse discriminante quadratique

Dans le cas où les matrices de variance-covariance Σ_k ne peuvent pas être supposées égales, on obtient le critère quadratique en x.

$$SQ_k(x) = -x^T\Sigma_k^{-1}x + 2\mu_k^T\Sigma_k^{-1}\mu_k - \ln(\det(\Sigma_k)) + 2\ln(p_k)$$

1.3 Estimation des paramètres

Le calcul de $SL_k(x)$ et $SQ_k(x)$ n'est possible qu'en estimant les paramètres μ_k, p_k et Σ_k non connus en pratique. Ces paramètres peuvent être estimés par maximum de vraisemblance auquel cas on obtient :

$$\begin{aligned}
\widehat{\mu_k} &= g_k \\
\widehat{p_k} &= \frac{n_k}{n} = w_k \\
\widehat{\Sigma_k} &= \begin{cases} \widehat{\Sigma} = \dfrac{n}{n-K}W & \text{(cas homoscédastique)} \\ \widehat{\Sigma_k} = \dfrac{n_k}{n_k-1}W_k & \text{(cas hétéroscédastique)} \end{cases}
\end{aligned}$$

1.3.1 Analyse discriminante linéaire (LDA)

Dans le cas homoscédastique, on a alors :

$$L_k(x) = 2g_k^T\widehat{\Sigma}^{-1}x - g_k^T\widehat{\Sigma}^{-1}g_k + 2\ln(\widehat{p_k})$$

Cette fonction est appelée fonction linéaire discriminante du groupe G_k. Chaque fonction linéaire discriminante définie une fonction de score et un nouvel individu sera donc affecté au groupe G_k pour lequel le score sela le plus élevé.

On retrouve la fonction linéaire discriminante

$$L_k(x) = x^TW^{-1}g_k - \frac{1}{2}g_k^TW^{-1}g_k$$

de l'analyse discriminante géométrique avec le terme $\ln(p_k)$ en plus. Dans le cas où l'on fait l'hypothèse d'égalité des probabilités à priori ($p_1 = \cdots = p_K$), la règle de l'analyse discriminante linéaire (LDA) est équivalente à la méthode de l'analyse discriminante géométrique.

1.3.2 Analyse discriminante quadratique (QDA)

Dans le cas hétéroscédastique, on a alors :

$$Q_k(x) = -(x-g_k)^T\widehat{\Sigma}^{-1}(x-g_k) - \ln(\det(\widehat{\Sigma_k})) + 2\ln(\widehat{p_k})$$

Cette fonction est appelée fonction quadratiue discriminante du groupe G_k. Chaque fonction quadratique discriminante définie une fonction score et un nouvel individu sera donc affecté au groupe G_k pour lequel le score sera le plus élevé.

1.4 Cas particulier de 2 groupes

On se place dans le cas où $K = 2$ et $\Sigma_1 = \Sigma_2 = \Sigma$ (cas homoscédastique).
On affecte donc un individu x au groupe 1 si $p_1 f_1(x; \mu_1, \Sigma) > p_2 f_2(x; \mu_2, \Sigma)$. Ce qui donne après calculs :

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

On pose donc

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

La nouvelle fonction discriminante est

$$L(x) = x^T \widehat{\Sigma}^{-1}(g_1 - g_2) - \frac{1}{2}(g_1 + g_2)^T \widehat{\Sigma}^{-1}(g_1 - g_2) - \ln\left(\frac{\widehat{p_2}}{\widehat{p_1}}\right)$$

On compare donc cette fonction à 0 pour affecteur x dans un des deux groupes :

— si $L(x) \geq 0 \Longrightarrow x$ est affecté au groupe 1

— si $L(x) \leq 0 \Longrightarrow x$ est affecté au groupe 2

Dans ce cas de figure précis, on a

$$\mathbb{P}(G_1 \mid x) = \frac{p_1 f_1(x)}{p_1 f_1(x) + p_2 f_2(x)} = \frac{1}{1 + \frac{p_2 f_2(x)}{p_1 f_1(x)}} = \frac{1}{1 + e^{-S(x)}}$$

Donc la probabilité à posteriori que $x \in G_1$ est la fonction logistique de $S(x)$.

2 ACP préliminaire (présentation du jeu de données)

2.1 Présentation du jeu de données

Notre jeu de données traite des exploitations fermières entre

Le contexte de l'époque se résumait par des réformes politiques sur l'agriculture, l'agrandissement de l'Union Européenne mais aussi des durcissements des contraintes économiques dans le secteur agricole dans les années 80. Tout cela a eu un impact sur les exploitations fermières françaises et s'est résulté par une multiplication des crises agricoles. c'est à partir de ce moment que s'est posé la question d'estimer les risques financiers en agriculture

L'expérience a démontré que les mesures de redressement financier pouvaient être efficaces pourvu que des actions préventives débutaient tôt. Donc il est important d'avoir une méthode pour la détection tôt et rapide des risques financiers en agriculture.

basé sur le concept de la viabilité des exploitations fermières

Donc par la suite nous avons quelques définitions sur la *viabilité* et l' *insolvabilité* :

— une ferme viable peut être définie comme

— une ferme qui assure au fermier un revenu équivalent à celui des autres catégories socio professionnelles

L' *insolvabilité* est définie comme la situation dans laquelle une exploitation agricole n'est pas en mesure d'honorer les obligations générées par la dette existante, à savoir le paiement des intérêts et le paiement des prêts la méthode du « credit scoring » promet de diagnostiquer de manière préventive les soucis financiers

des exploitations.

Puis on nous donne plusieurs critères qui contribueraient à la déstabilisation des fermes d'un point de vue financier : le déclin des prix des produits agricoles l'augmentation des crédits affaiblissement financier dû à : une augmentation des dépenses une baisse du chiffre d'affaire une recrudescence des incidents et des retards de paiement

2.2 Données

Les données concernent 1260 exploitations fermières qui sont réparties en 2 groupes (décrits par la variable DIFF). Le premier groupe rassemble les fermes saines (653) et le second groupe rassemble les fermes défaillantes (607). La variable à expliquer est donc cette variable DIFF.

- `CNTY` : code de département
- `DIFF` : variable à expliquer, est ce que la ferme a déjà eu un incident de paiement (1= ferme saine, 2= ferme défaillande)
- `STATUS` : statut légal (1 = propriétaire indépendant, 0= entreprise)
- `HECTARE:` aire de la ferme en hectares
- `ToF` : index de type de ferme
- `OWNLAND` : owned land (O= Oui, N= non)
- `AGE` : l'âge du propriétaire des terres
- `HARVEST` : année de récolte concernée

De plus, pour calculer les risques financiers plusieurs ratios sont présentés.
On nous définit ainsi un ensemble de critères micro économiques qui calculent le degré de faillite des exploitations fermières. Ces ratios sont les variables explicatives de notre variable `DIFF`

- Capitalisation : R1, R2, R3, R4, R5
- Poids des dettes : R6,R7,R8
- Liquidité :R11, R12, R14
- Debt servicing : R17, R18, R19, R21, R22
- Capital Profitability : R24
- Earnings : R28, R30, R32
- Produtive activity : R36, R37

La première méthode nous permettant de construire une score utilisera une Analyse en Composantes Principales (ACP). Nous allons voir que nous pouvons faire du scoring avec une Analyse en composantes principales.

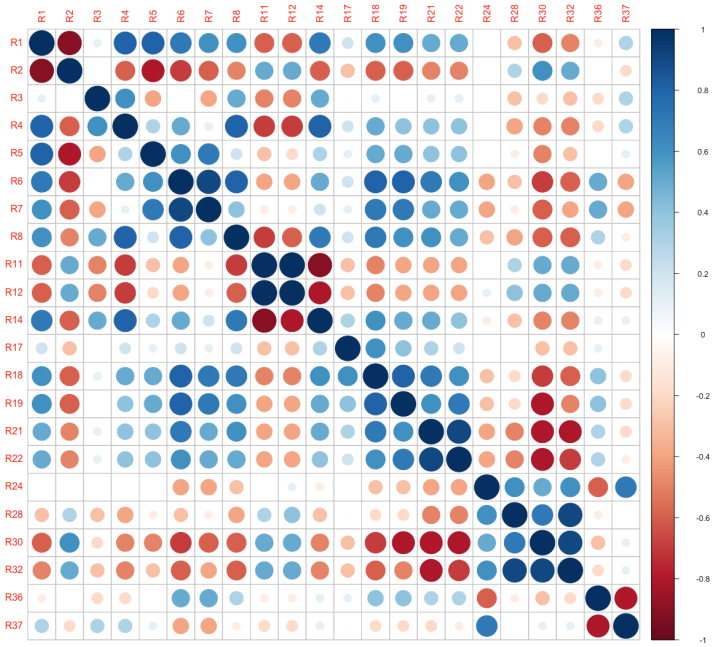


FIGURE 16 – Corrélation entre ratios financiers

Cette figure nous permet de remarquer les différentes corrélations entre ratios financiers on remarque alors qu'ils sont rassemblés par groupes.

Analyse en composantes principales Pour résumer l'information contenue dans ces ratios financiers on fait une Analyse en composante principale.

On peut lire la corrélation entre les différentes variables (ratios) On a également représenté des variables supplémentaires (celles qui ne sont pas des ratios)

```
acp = PCA(farms2, quali.sup = c(1,2,3,5,6), quanti.sup = c(4,7,8))

fviz_pca_var(acp,
             col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
)
```

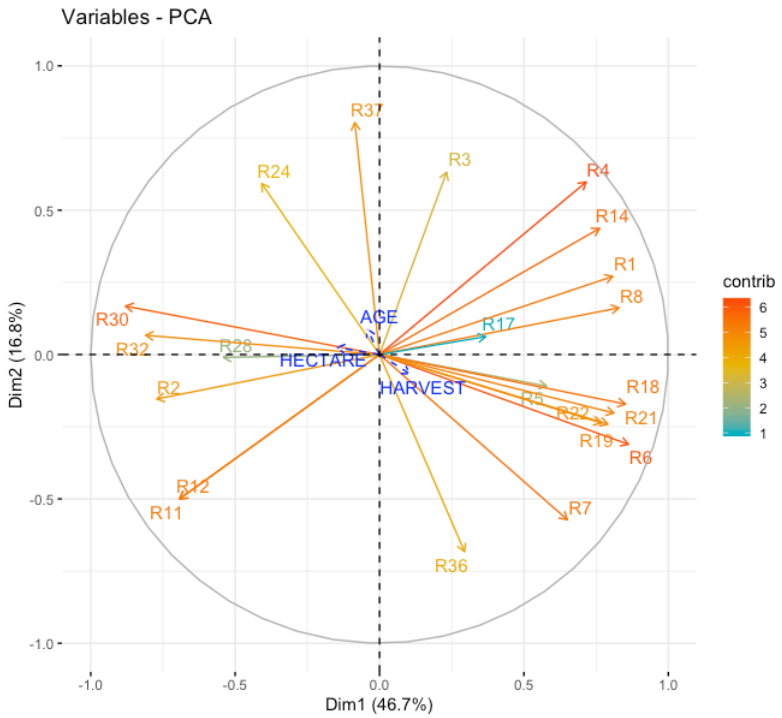


FIGURE 17 – ACP

L'axe F1 représente 47% de l'information totale. Cet axe montre l'opposition entre deux groupes de ratios :

Les variables quantitatives supplémentaires comme AGE, HECTARE sont trop peu corrélées avec les deux premiers facteurs. Ainsi leur projection sur le premier plan factoriel ne peut pas être interprétée.

L'axe F2 représente 17% de l'information totale, il peut être interprété comme

Positivement corrélé	Négativement corrélé
R37	R36
R3	R7
R4	R11
R24	R12

L'axe F2 représente la relation entre

Nous avons ci dessus la représentation des différentes fermes sur les plans F1-F2 :

FIGURE 18 – Représentation des fermes sur les plans F1-F2

On peut ainsi voir que les fermes saines sont bien séparées des fermes défaillantes sur ces axes. On remarque qu'une méthode de classification serait de créer une droite de regression :

$$y = -1.07578x + 0.00005$$

Nous représentons ci dessous les différentes fermes sur les plans F1-F3 et les plans F2-F3.

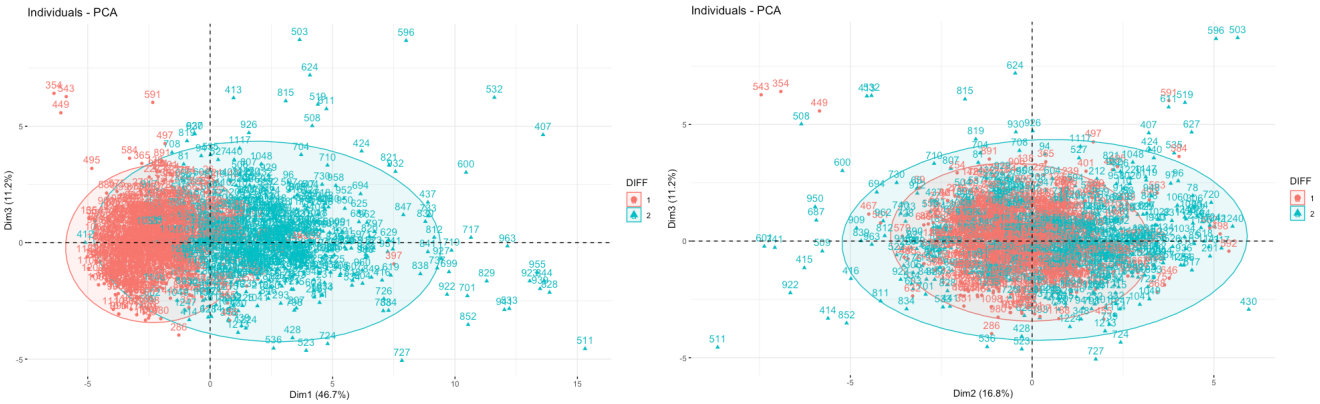


FIGURE 19 – Représentation des fermes sur les plans F1-F3 (à gauche) et F2-F3 (à droite)

De ces graphiques on en conclue que c'est l'axe 1 qui est indispensable à une bonne distinction des deux groupes de fermes. On va ainsi utiliser seulement cet axe

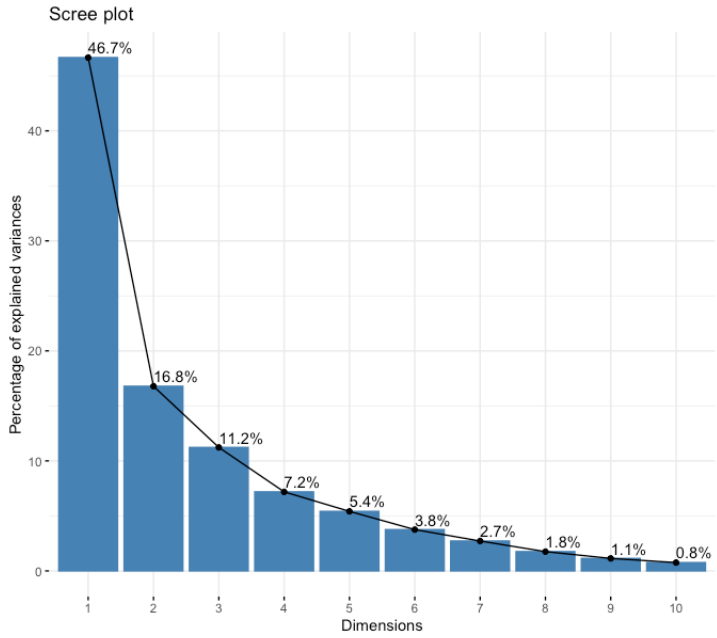


FIGURE 20 – Histogramme des variances expliquées

Cette ACP montre que la première composante principale s'avère être un facteur discriminant acceptable quand on cherche à privilégier l'intérêt général. On peut cependant utiliser une méthode conçue pour maximiser la note globale des exploitations classées correctement pour améliorer les performances de classement.

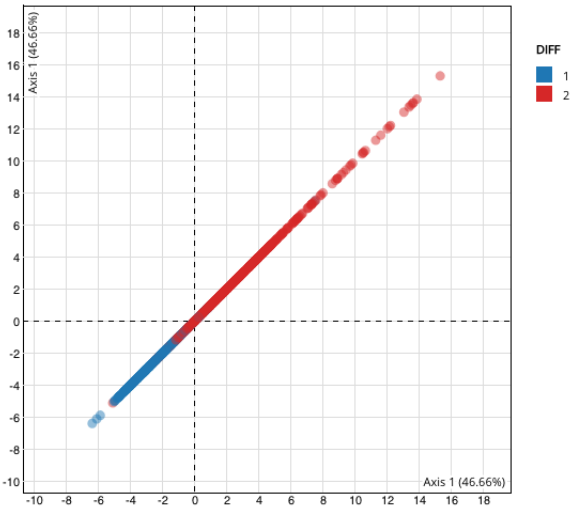


FIGURE 21 – Représentations des fermes sur le premier axe factoriel

La moyenne des individus du groupe des fermes saines pour l'axe F1 est : $\mu_0 = -2.16474485$. La moyenne des individus du groupe des fermes défaillantes pour l'axe F1 est $\mu_1 = 2.32879471$. Ainsi un point pivot est la moyenne de ces deux valeurs.

$$p = \frac{\mu_0 + \mu_1}{2} = 0.08202495$$

```
acp_dim1[["ind"]][["coord"]]   #pour coordonnees des points
acp_dim1[["call"]][["quali.sup"]][["quali.sup"]][["DIFF"]]  #vrais resulats DIFF
DIFF_prevu_dim1 = rep(0,n)
for (i in 1:n){
```

```
  x = acp_dim1[["ind"]][["coord"]][i]
  point_pivot = 0.08202495
  if (x>point_pivot){
    DIFF_prevu_dim1[i] <- 2
  }
  if (x<point_pivot){
    DIFF_prevu_dim1[i] <- 1
  }
}

obs_dim1 = DIFF_prevu_dim1
pred = acp_dim1[["call"]][["quali.sup"]][["quali.sup"]][["DIFF"]]
```

Nous obtenons à partir de cet algorithme le vecteur `obs_dim1` indiquant par des 1 : les fermes considérés comme saines et des 2 les fermes considérées comme défaillantes (par le point pivot).
On va comparer ce vecteur prédit à l'actuel vecteur `DIFF` à l'aide de l'algorithme ci dessous créant la matrice de confusion associée.

```
A_dim1 = matrix(0, nrow = 2, ncol=4)
A_dim1[1,] = 1:4

for (i in 1:n){
  if (obs_dim1[i] == 1 & pred[i] == 1){      # vrai: sain     observe: sain
    A_dim1[2,1] <- (A_dim1[2,1] + 1)
  } else if (obs_dim1[i] == 2 & pred[i] == 2){  #vrai: risque  observe: risque
    A_dim1[2,2] <- (A_dim1[2,2] + 1)
  } else if (obs_dim1[i] == 1 & pred[i] == 2){   #vrai:risque  observe: sain
    A_dim1[2,3] <- (A_dim1[2,3] + 1)
  } else { A_dim1[2,4] <- (A_dim1[2,4] +1)} # vrai: sain  observe: risque
}
```

	$Y = 0$	$Y = 1$
$Y_{pred} = 0$	599	124
$Y_{pred} = 1$	54	483

Ainsi on obtient un taux de fermes saines bien classées de 91.7 %, et un taux de fermes défaillantes bien classées de 79.6 %.

L'utilisation de l'axe F1 donne alors de meilleurs résultats que l'utilisation des axes F1 et F2. Cependant avec cette méthode le taux de faux négatifs soit "observer une ferme saine alors qu'elle est défaillante" est trop élevé. Si on adopte le point de vue d'un banquier, ce modèle ne sera donc pas satisfaisant. En effet on veut éviter les situations de non remboursement des prêts.

3 Exemple sur R

La méthode de construction de score l'analyse discriminante probabiliste

3.1 Choix des modèles et des variables

Modèle 1 : modèle complet

```
greedy.wilks(DIFF~., data=farms, niveau=0.01)
```

```
Formula containing included variables:

DIFF ~ R1 + R32 + R14 + R17 + R2 + R3 + R36 + R21

Values calculated in each step of the selection procedure:

  vars Wilks.lambda F.statistics.overall
1   R1    0.5804430             909.3100
2  R32    0.5017071             624.2229
3  R14    0.4667025             478.4073
4  R17    0.4534737             378.1314
5   R2    0.4451038             312.6641
6   R3    0.4371952             268.8328
7  R36    0.4292739             237.7932
8  R21    0.4234460             212.9165
```

FIGURE 22 – Lambda de Wilks = 0.01

```
greedy.wilks(DIFF~., data=farms, niveau=0.05)
```

```
Formula containing included variables:

DIFF ~ R1 + R32 + R14 + R17 + R2 + R3 + R36 + R21 + R7 + R18 +
    R19

Values calculated in each step of the selection procedure:

   vars Wilks.lambda F.statistics.overall
1    R1    0.5804430             909.3100
2   R32    0.5017071             624.2229
3   R14    0.4667025             478.4073
4   R17    0.4534737             378.1314
5    R2    0.4451038             312.6641
6    R3    0.4371952             268.8328
7   R36    0.4292739             237.7932
8   R21    0.4234460             212.9165
9    R7    0.4217347             190.4387
10  R18    0.4188155             173.3220
11  R19    0.4172071             158.4837
```

FIGURE 23 – Lambda de Wilks = 0.05

3.2 Analyse disriminante linéaire (LDA)

```
#Etude statistique
#greedy.wilks(DIFF~., data=farms, niveau=0.01)

N = 1000
mod_ret = rep(0,N)
```

```
scores_A = rep(0,8)

for (k in 1:N){
  sample = sample.split(DIFF, SplitRatio = 0.8)
  train = subset(farms, sample == TRUE)
  test = subset(farms, sample == FALSE)

  modele_1 = lda(DIFF~.,data=train)   #ceux de la r?gression logistique
  modele_2 = lda(DIFF ~ R1 + R3 + R14 + R17 + R36, data = train)
  modele_3 = lda(DIFF ~ R1 + R3 + R17 + R36, data = train)
  modele_4 = lda(DIFF ~ R1 + R14 + R17 + R36, data = train)
  modele_5 = lda(DIFF ~ R1 + R12 + R14 + R17 + R32 + R36,data=train)
  modele_6 = lda(DIFF ~ R2 + R7 + R17 + R32,data=train) #ceux du TP
  modele_7 = lda(DIFF ~ R1 + R2 + R3 + R7 + R14 + R17 + R18 + R19 + R21 + R32 + R
     36,data=train) #crit?re de Wilks lambda ? 0.05
  modele_8 = lda(DIFF ~ R1 + R2 + R3 + R14 + R17 + R21 + R32 + R36,data=train) #
     idem mais ? 0.01

  liste_modeles = list(modele_1, modele_2, modele_3, modele_4,modele_5,modele_6,
     modele_7,modele_8)
  n = length(liste_modeles)

  A = matrix(0, nrow = 2, ncol=n)
  A[1,]= 1:n

  for (i in 1:n){
    diff.pred = predict(liste_modeles[[i]],test[,-1],method="predictive")$class #
       rend les classes pr?dites
    erreur_pred = prop.table(table(diff.pred, test$DIFF))[2]   #rend le taux de
       faux negatifs (0=sain, 1=defaillant)
    A[2,i] = erreur_pred
  }
  A_tri = A[,order(A[2,], decreasing = FALSE)]
  for (i in 1:n){
    scores_A[A_tri[1,i]] = scores_A[A_tri[1,i]] + (9-i)}
}

scores_A
#attribuer des scores: matrice deux lignes, une avec les modeles une avec les
   scores


"Mod le final"
#C'est le mod?le 5 qui gagne

afd_fin = lda(DIFF ~ R1 + R12 + R14 + R17 + R32 + R36 ,data=farms)

pred = predict(afd_fin,farms[,-1])
prob_post = pred$posterior    #Quelle proba choisir ici --> voir avec Mr Pro?a

#point de vue on minimise les d?faillantes mal class?es (FN)
score_1 = filter(data.frame(prob_post[,2], farms$DIFF), farms$DIFF == "saine")  #
   vrai score, ceux qui sont 0 (1)
score_2 = filter(data.frame(prob_post[,2], farms$DIFF), farms$DIFF == "
   d faillante")  #vrai score, ceux qui sont 1 (2)

#score_1bis=filter(data.frame(prob_post[,1], farms$DIFF), farms$DIFF == "saine")
#score_2bis=filter(data.frame(prob_post[,1], farms$DIFF), farms$DIFF == "d?
   faillante")

hgB1 = hist(score_1$prob_post...2., breaks=100, plot=F) #histogramme des scores
   des vraies fermes saines
hgB2 = hist(score_2$prob_post...2., breaks=100, plot=F) #histogramme des scores
```

```
   des vraies fermes d?faillantes


col_1 = rgb(1,0,0,0.5)
col_2 = rgb(0,0,1,0.5)

plot(hgB1, col= col_1, freq=FALSE, xlim=c(0,1), ylim= c(0,20))
plot(hgB2, col=col_2, freq=FALSE, xlim=c(0,1), add=T)
lines(density(score_2$prob_post...2.), lwd=1.5)
lines(density(score_1$prob_post...2.), lwd=1.5)

ApproxQuantile(hgB2, 0.05)
ApproxQuantile(hgB2, 0.1)
```

3.3 Analyse discriminante quadratique (QDA)

bibliographie$_s$*coring*