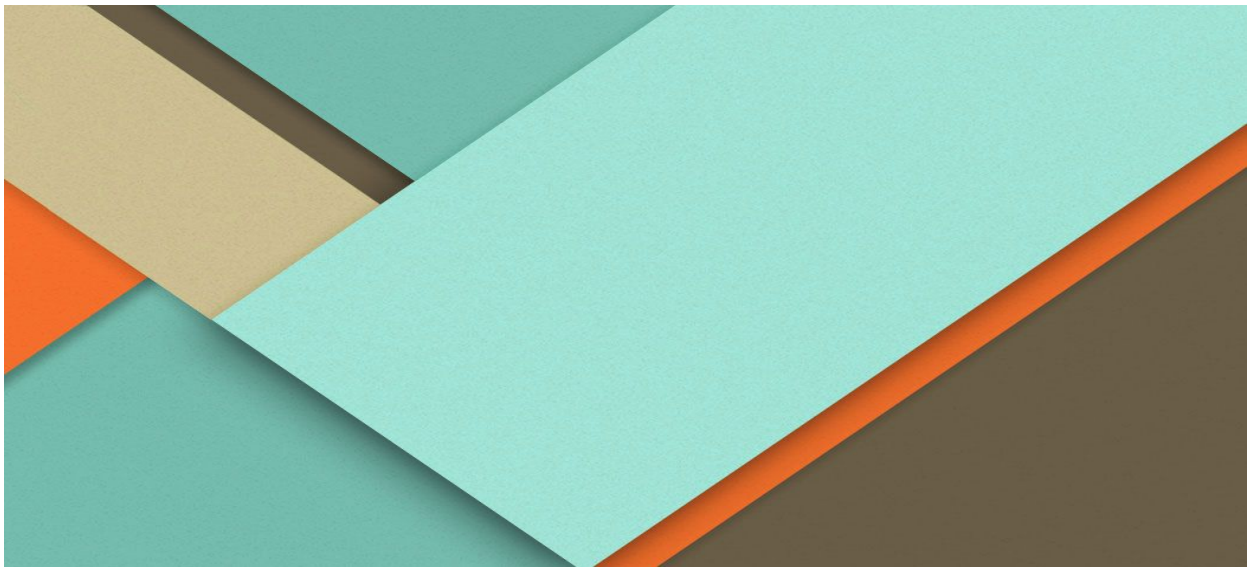


Compte rendu du projet Scoring

15.02.2017

Touzi Marwen

Beniss Selma

Benleulmi Abdelmalek

Riffault Farah

Master 2 IMDS

Paris-Saclay

Université d'Evry

Professeur : Ibrahim Touré

1. Introduction :

Le but de ce projet est de modéliser le défaut d'un portefeuille composé de plusieurs entreprises afin d'anticiper le comportement d'entités nouvelles.

Pour le faire, nous utilisons un jeu de données constitué par 5 variables numériques et 4000 observations qui présentent des caractéristiques financières des entreprises, et leur défaut (0 lors de défaut, 1 sinon).

Le but de notre étude est la prédiction de la variable défaut, ce compte rendu présentera l'application et la comparaison des différents modèles de classification supervisée, de méthode de score. Nous allons nous intéresser plus à la régression logistique vu qu'elle est essentiellement utilisée pour ce genre de données et pour répondre à des problématiques de score.

2. La régression logistique sur variables quantitatives:

2.1. Question 1:

Notre but est de prédire la variable "DEFAULT" qui est une variable qualitative. Donc, nous allons construire des modèles d'apprentissage supervisé. Il existe des modèles paramétriques et des modèles non paramétriques (le nombre de paramètres qui décrivent la loi des observations est une fonction croissant du nombre d'observations, ou le nombre de paramètres est infini).

Comme exemple de modèles paramétriques, nous pouvons citer les modèles linéaire comme **la régression logistique** (qui offre un pouvoir explicatif et prédictif très fort du fait de sa linéarité et qui pénalise les faux positifs et les faux négatifs). On peut aussi citer les modèles Probit et Tobit.

Le Support Vector Machine qui fait une redescription de nos variables dans un nouvel espace en maximisant la marge entre la frontière de décision et les observations les plus proches).

Les arbres de décisions qui construisent des règles à partir des données qui permettent de les ordonner en se basant sur de critères tels que l'indice de Gini et l'entropie

(exemple : CART, CHAID qui nécessite une transformation des variables quantitatives explicatives en variables catégoriques au préalable).
Le Random Forest (plusieurs arbres en appliquant le tree bagging et le feature sampling) et Le Gradient boosting (non linéaire, même principe que le Random Forest mais chaque itération vise à corriger l'erreur de la précédente).

Pour les modèles non paramétriques, nous pouvons utiliser les algorithme de **recherche des plus proches voisins** qui, en se basant sur des métriques de similarité, génère les prédictions.

2.2. Question 2 :

On suppose que Y prend deux valeurs (0 ou 1), il s'agit donc d'une variable binaire. Pour utiliser une régression logistique on suppose que nos variables suivent une loi logistique. Le logit se défini de cette manière :

$$logit(P[y = 1/X]) = log\left(\frac{P[y = 1/X]}{1 - P[y = 1/X]}\right)$$

Comme nous utilisons la régression logistique, voilà quelques écritures de ce modèle :

- $\log\left(\frac{P[y=1/X]}{1-P[y=1/X]}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \varepsilon$

- $P[y = 1/X] = \frac{exp^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\varepsilon}}{1+ exp^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\varepsilon}}$

- $P[y = 1/X] = \frac{1}{1+ exp^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\varepsilon)}}$

2.3. Question 3 :

Pour estimer les paramètres de notre modèle nous utilisons le maximum de vraisemblance puisque notre variable à expliquer contient que deux modalités (0 et 1).

$(Y_1, \dots, Y_n)$ *un échantillon iid (indépendemment et identiquement distribués) de la variable Y*

$(y_1, \dots, y_n)$ *la réalisation de cet échantillon*

La vraisemblance de l'échantillon est la probabilité d'observer la réalisation obtenue de l'échantillon. On suppose que la loi de Y dépend d'un paramètre β. On peut la noter ainsi :

$$L(\beta) = L(y_1, \dots, y_n;\ \beta) = P(Y_i = y_1, \dots, y_n; \beta)$$

Où P désigne la probabilité sous le modèle logistique qu'un événement se réalise.

En effet l'estimateur du maximum de vraisemblance de β est la valeur de β qui maximise la log-vraisemblance. Il s'agit dont de la valeur de β qui rend le plus vraisemblable l'échantillon observé.

On peut écrire :

$$L(\beta) = \prod_{i=1}^{n} P(Y_i = y_i / X'_i)$$

Dans notre cas y_i peut prendre deux valeurs soit 0 ou 1, dans notre cas 0 signifie que l'entreprise a fait défaut et 1 l'entreprise n'a pas fait défaut.

Ainsi on a :

$$\left(P(Y_i = y_i / X'_i) \;\middle|\; \begin{matrix} P(Y_i = 1/X'_i)\ si\ y_i = 1 \\ P(Y_i = 0/X'_i)\ si\ y_i = 0 \end{matrix} \right)$$

$$= P(Y_i = 1/\ X'_i)^{y_i}.P(Y_i = 1/\ X'_i)^{1-y_i}$$

Donc

$$L(y_1, \dots, y_n / X'_i)$$

$$= \prod_{i=1}^{n} P(Y_i = y_i / X'_i)$$

$$= \prod_{i=1}^{n} P(Y_i = 1 / X'_i)^{y_i} . P(Y_i = 0 / X'_i)^{1-y_i}$$

Or on sait que le modèle logistique s'écrit de cette manière (Cf. question 2) :

$$P\left[y = \frac{1}{X'_i}\right] = \frac{exp^{X_i'B}}{1 + exp^{X_i'B}}$$

Ainsi on aura :

$$\prod_{i=1}^{n} \left(\frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{y_i} . \left(1 - \frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{1-y_i}$$

Or ce que l'on souhaite c'est de maximiser la probabilité que notre échantillon soit vraisemblable. On aura donc :

$$\max \left(\prod_{i=1}^{n} \left(\frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{y_i} . \left(1 - \frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{1-y_i}\right)$$

Pour cela il est communément accepté de simplifier cette équation en passant par le log, on aura donc :

$$\max \left(\log \left(\prod_{i=1}^{n} \left(\frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{y_i} . \left(1 - \frac{exp^{X_i'B}}{1 + exp^{X_i'B}}\right)^{1-y_i} \right) \right) = \mathcal{L}(\beta)$$

2.4. Question 4 :

L'algorithme de Newton-Raphson qui à l'origine consiste à introduire une suite Xn qui permet d'approcher successivement f(x)=0. En régression logistique cet algorithme a le nom de IRLS ce qui signifie Iteratively Reweighted Least Squares. Il permet donc de trouver une suite de vecteur β^k capable de converger vers l'estimateur du maximum de vraisemblance. On veut donc que notre fonction atteigne un maximum, elle atteindra ce maximum (car $\mathcal{L}(\beta)$ est strictement concave, si elle admet un extremum alors il s'agit du maximum de la fonction) lorsque notre gradient (il dérive par rapport à tous les paramètres) sera égale à 0 :

$$S(\beta) = \nabla \mathcal{L}(\beta) = 0$$

Avec ∇ le gradient

Cela veut dire que notre produit de probabilité atteint son maximum pour en ce vecteur β , on a donc trouvé la meilleure estimation β .

L'agorithme IRLS est donc un algorithme itératif qui permet de converger vers l'EMV.

Cette méthode de convergence s'appuie sur le développement de Taylor.

Rappel le développement de Taylor du premier ordre s'écrit de cette façon :

$$f(x) \approx f(a) + f'(a)(x - a)$$

Ainsi on aura :

$$S(\widehat{\beta_n}) = 0$$

$$S(\widehat{\beta_n}) \approx S(\beta^{(k)}) + A(\beta^{(k)})(\widehat{\beta_n} - \beta^{(k)})$$

Avec A la dérivée seconde de $\nabla\mathcal{L}(\beta^k)$, soit la matrice hessienne de la log-vraisemblance :

$$A = \nabla^2\mathcal{L}(\beta^k)$$

On obtient donc $\widehat{\beta_n} = \beta^{(k)} - A^{-1}\beta^{(k)}S(\beta^{(k)})$

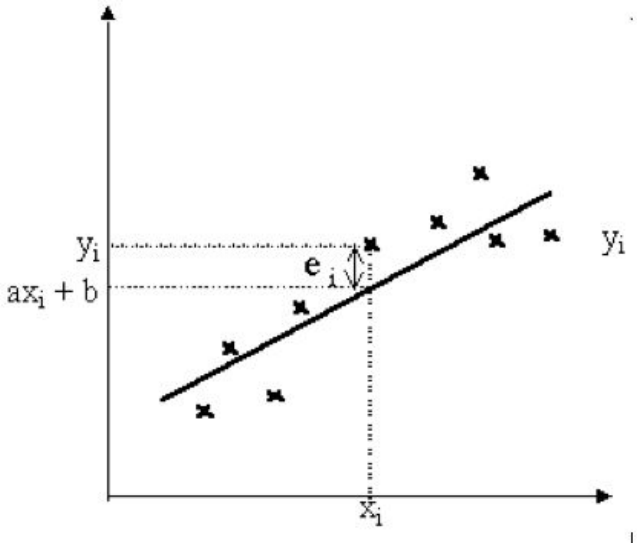
On va réitérer l'opération afin d'approximer 0 ce qui va donner la formule de récurrence suivante :

$$\beta^{(k+1)} = \beta^{(k)} - A^{-1}\beta^{(k)}S(\beta^{(k)})$$

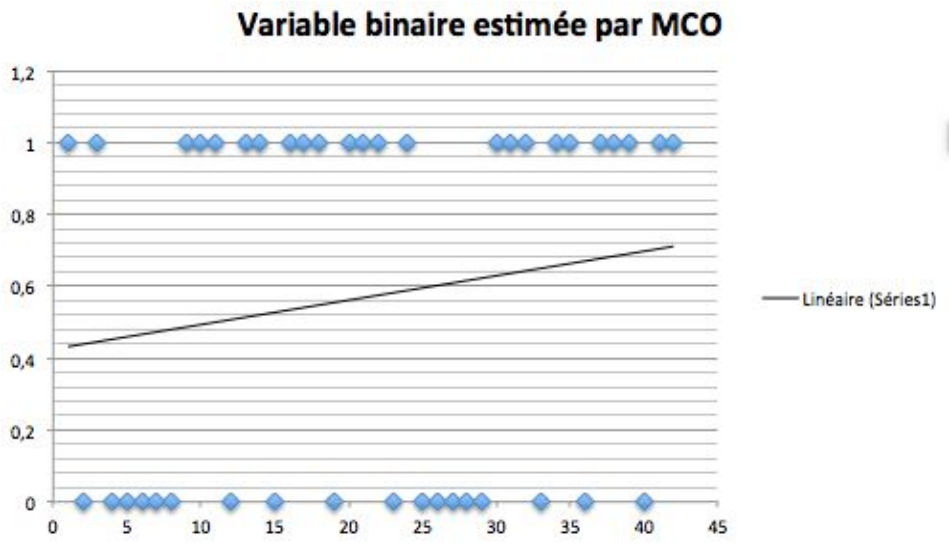
2.5. Question 5 :

Il n'est pas correcte d'utiliser la méthode des moindres carrés ordinaires (MCO) pour estimer notre modèle car les MCO consistent à construire une « droite » qui s'ajuste le mieux à un nuage de points. Pour cela on minimisera la somme des carrés des écarts entre la réalisation de notre variable à expliquer et l'estimation de notre variable à expliquer.

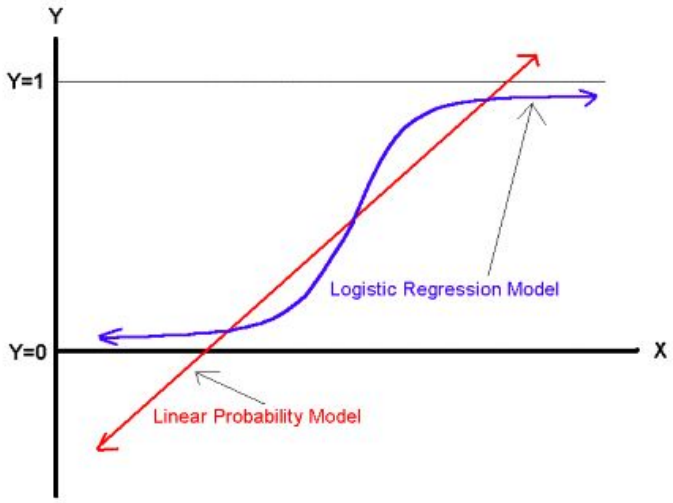
Visuellement on a quelque chose qui ressemble à ce graphique ci-dessous lorsque l'on souhaite modéliser une variable quantitative (non bornée).



Or nous ce que nous voulons c'est estimer une variable qui prend seulement deux valeurs à savoir 0 et 1. Ainsi le risque en utilisant les MCO pour estimer nos paramètres on serait de prédire des probabilités inférieures à 0 ou supérieure à 1 et d'avoir une droite d'estimation qui ressemble à ça :



Cette méthode ne nous permet pas d'estimer si la personne sera attribuée 0 ou 1. Pour cela on doit utiliser une autre méthode pour se rapprocher de la réalité et ce grâce à une régression logistique (en bleu sur le graphique). Cette méthode nous permet d'estimer notre variable cible binaire de manière plus appropriée.



2.6. Question 6:

Les données manquantes sont des données perdues pour une raison ou autre. La plupart des algorithmes de modélisation exigent le fait de n'avoir que des variables qui n'ont pas de données manquantes, pour cela, selon la fréquence de ces valeurs, soit on supprime carrément les variables ou les lignes qui contiennent ces valeurs, soit on les remplace par la moyenne ou par la modalité la plus présente (dans le cas d'une variable qualitative) ou on les estime avec des modèles d'apprentissage supervisé, parfois une valeur manquante peut être considérée comme une information, il est nécessaire de faire attention aux raisons de ce manque de données . Dans notre cas, nous n'avons pas de données manquantes dans notre base de données :

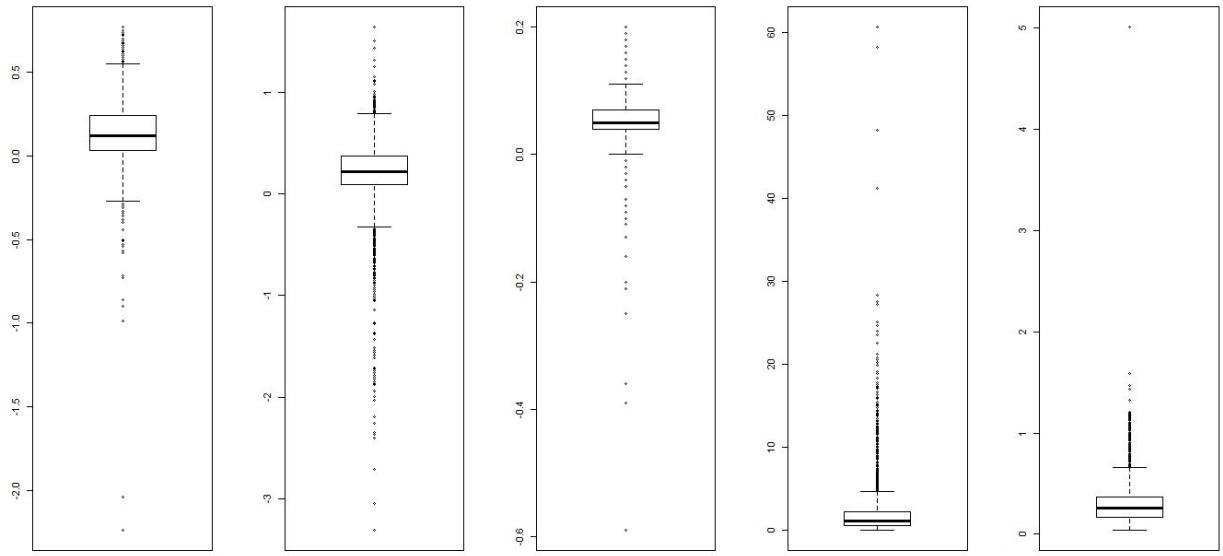
```
> na_count <-sapply(data, function(y) sum(length(which(is.na(y)))))
> na_count
     ID    Year Default    WCTA    RETA EBIT_TA    METL     STA
      0       0       0       0       0       0       0       0
```

Les données aberrantes sont des données qui ne sont pas logiques ; par exemple un âge négatif, ou encore une date d'inscription avant la date de naissance d'un utilisateur.

Dans notre cas, vu que nous n'avons pas des connaissances financières pour trouver le domaine de définition pour chaque variable, et vu que toute nos données sont numériques et paraissent raisonnables (les minimaux et les maximums ne sont pas vraiment loins des moyennes) nous avons supposé que nous avons pas des données aberrantes.

Les données extrêmes ne sont pas des données erronées mais elles ont un grand effet sur le modèle. Pour les détecter, nous utilisons souvent les boîtes à moustaches (les données aux dessous du premier quartile et celles aux dessus du troisième quartile), les histogrammes et les courbes des densités (les points qui ne suivent pas les autres), les tests d'adéquation aux loi de probabilité (les points qui ne sont pas situés sur la première bissectrice de la QQplot par exemple), les graphes d'analyse factorielle (les points isolé des regroupement des données) ou bien sur les graphes qui représentent les résidus.

La détection des valeurs extrêmes est plus facile que leurs manipulation. Avoir un grand effet sur le modèle ne signifie pas qu'il biaise toujours le modèle, mais nous devons à chaque fois les supprimer et regarder si leur effet est positif ou négatif. Voilà quelque graphiques qui présentent ces valeurs :



Les Box-Plot ci-dessus permettent de détecter les valeurs extrêmes, en effet plus les données sont loins du premier et dernier quartile plus les valeurs sont considérées comme étant extrêmes.

Nous avons également décidé de faire d'autres graphiques et tests pour détecter les valeurs extrêmes :

Exemple variable EBIT_TA :

Pour que définir si notre variable suit une loi normale il faut que le Skewness (coefficient d'asymétrie) soit égal ou proche de 0 ce qui signifie que la distribution est symétrique. Si le Skweness est négatif cela signifie que la distribution à une forte queue à gauche (toute chose égale par ailleurs).

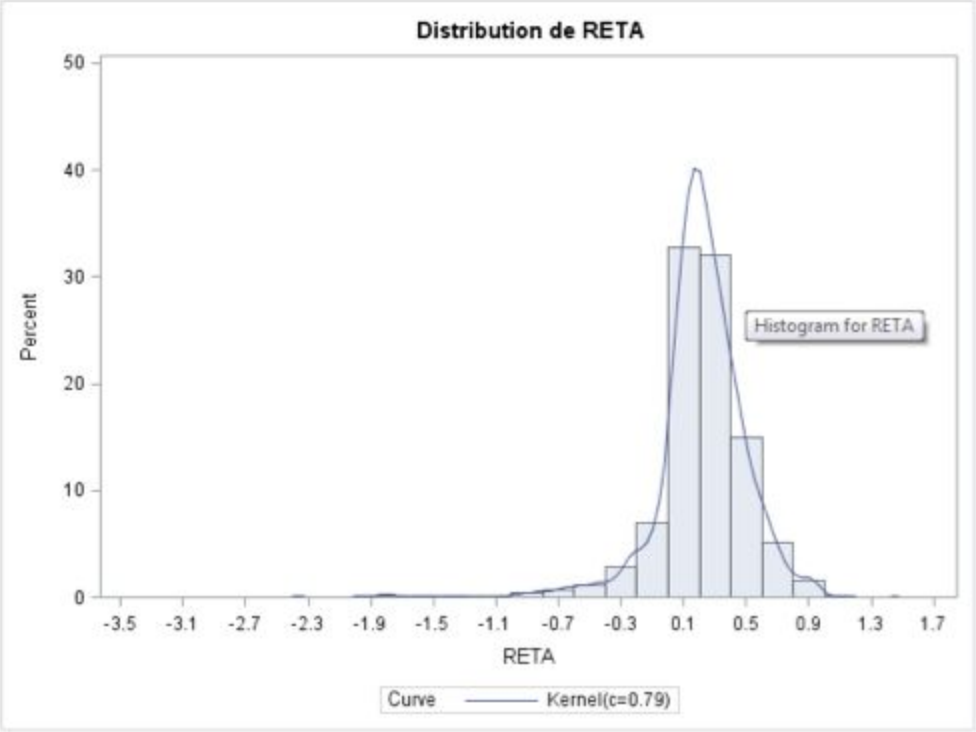
Pour le Kurtosis (coefficient d'applatissement) on doit comparer sa valeur par rapport à 3 (ou 0 pour K' avec K'= K-3, Cf. Fisher) pour déterminer s'il s'agit d'une loi normale ou pas. Si notre Kurtosis est inférieure à 3 (K'<0) cela indique que les queues contiennent plus d'informations (platikurtique), en revanche si le Kurtosis est supérieure à 3 (K'>0) cela indique que les queues contiennent moins d'information que dans une distribution gaussienne (leptokuritique).

Pour la variable EBIT_TA on contaste un Skewness inférieure à 0 ce qui indique une distribution avec une plus forte queue à gauche. On contaste également que le coefficient d'aplatissement est loin de la valeur correspondant à la distribution gaussienne.

Variable : EBIT_TA (EBIT_TA)

Moments			
N	4000	Somme des poids	4000
Moyenne	0.05180323	Somme des observations	207.212917
Ecart-type	0.02959551	Variance	0.00087589
Skewness	-4.8443616	Kurtosis	85.9974266
Somme des carrés non corrigée	14.2370001	Somme des carrés corrigée	3.50270189
Coeff Variation	57.1306353	Std Error Mean	0.00046795

Graphiquement on constate que les queues sont très fines ce qui indique qu'il existe des valeurs extrêmes, le plus le fait que graphiquement on observe une asymétrie cela signifie qu'il y a plus de valeurs extrêmes d'un côté que d'un autre.



Le logiciel SAS nous permet également d'observer à quelles sont exactement nos valeurs extrêmes. Ci-dessous le tableau permettant de les repérer.

Observations extrêmes			
Le plus bas		Le plus haut	
Valeur	Obs	Valeur	Obs
-0.591821	3968	0.167703	907
-0.394546	1559	0.173217	2033
-0.362083	60	0.178964	3052
-0.254492	1390	0.190057	3402
-0.210191	3395	0.198035	905

Nous avons fait cette « Proc Univariate » pour chacune des autres variables et on a constaté qu'aucune ne suit une loi normale et que toutes possèdent des valeurs extrêmes (Cf. point en annexe pour les sorties des autres variables).

Conclusion : Nous avons beaucoup de données extrêmes, nous allons les corriger dans la partie 3 de ce rapport.

2.7. Question 7 :

Pour bien comprendre la répartition de nos données et la relation entre eux et notre variable cible nous sommes passés par les statistiques descriptives.

2.7.1. Statistique univarié :

Concernant la variable défaut on peut conclure que le taux de défaut de notre portefeuille sur la période de 1995 à 2004 est de 1,18%.

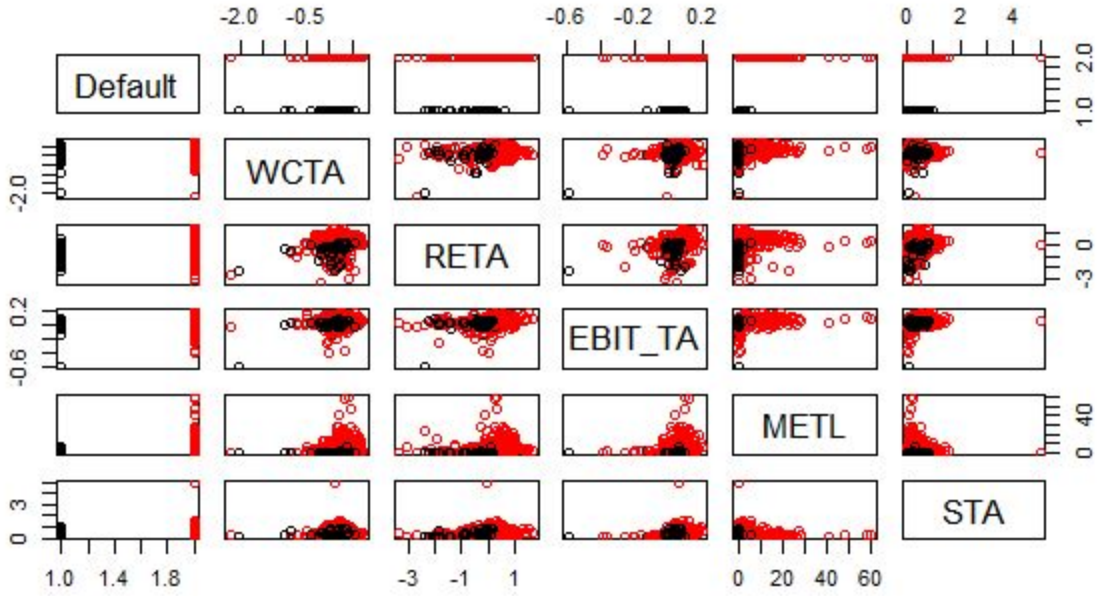
```
> summary(data)
       ID              Year       Default        WCTA              RETA              EBIT_TA        
 Min.   :  1.0   Min.   :1995   0:  72   Min.   :-2.2400   Min.   :-3.3100   Min.   :-0.59000  
 1st Qu.:192.0   1st Qu.:1998   1:3928   1st Qu.: 0.0300   1st Qu.: 0.0900   1st Qu.: 0.04000  
 Median :358.0   Median :2000            Median : 0.1200   Median : 0.2200   Median : 0.05000  
 Mean   :356.3   Mean   :2000            Mean   : 0.1426   Mean   : 0.2104   Mean   : 0.05181  
 3rd Qu.:521.0   3rd Qu.:2002            3rd Qu.: 0.2400   3rd Qu.: 0.3700   3rd Qu.: 0.07000  
 Max.   :830.0   Max.   :2004            Max.   : 0.7700   Max.   : 1.6400   Max.   : 0.20000  
      METL              STA         
 Min.   : 0.020   Min.   :0.0400  
 1st Qu.: 0.620   1st Qu.:0.1700  
 Median : 1.140   Median :0.2600  
 Mean   : 1.954   Mean   :0.3036  
 3rd Qu.: 2.240   3rd Qu.:0.3700  
 Max.   :60.610   Max.   :5.0100  
```

2.7.2. Statistique Bivarié :

Matrice de variance covariance entre les variables numériques :

```
> cor(as.matrix(data[4:8]), method = c("pearson"))
             WCTA       RETA   EBIT_TA        METL         STA
WCTA    1.0000000 0.19731068 0.1821160 0.206030895 0.247898916
RETA    0.1973107 1.00000000 0.3277421 0.236996136 0.080873128
EBIT_TA 0.1821160 0.32774212 1.0000000 0.283165019 0.251783256
METL    0.2060309 0.23699614 0.2831650 1.000000000 0.001731055
STA     0.2478989 0.08087313 0.2517833 0.001731055 1.000000000
```

Un autre outil d'analyse est aussi de réaliser des nuages de points pour toutes les variables. On peut éventuellement colorer les points selon la variable cible.



Nous remarquons que nos variables ne sont pas vraiment corrélées entre eux, ce qui limite les problèmes de multicolinéarité.

Regardons maintenant la relation entre nos variables numériques et notre variable cible. Nous allons expliquer comment nous avons raisonné avec la variable WCTA puis nous allons généraliser notre raisonnement sur les autres variables. Nous avons posé cette hypothèse nulle : ”la variable WCTA dépend de la variable Default”. C’est à dire “WCTA avec un défaut et WCTA sans défaut sont issus de la même distribution”. Pour se faire nous pouvons utiliser le test de student sur deux échantillons (si WCTA_def et WCTA_sans_def sont gaussiennes), sinon nous pouvons utiliser le test non paramétrique de Wilcoxon (si les deux échantillons sont diffus c’est à dire il y’a pas d’ex-aequos).

```
X <- data$WCTA[data$Default==0]
Y <- data$WCTA[data$Default==1]
lillie.test(X)#p-val<5% => on rejette la normalité
lillie.test(Y)#p-val<5% => on rejette la normalité
#cl: nous pouvons pas appliquer le test de Student  t.test(X,Y)
wilcox.test(X,Y)#p-value>5% => on ne rejette pas H0
#=> on ne rejette pas que les 2 echant suivent la mm loi
#conditions:
length(unique(X))==length(X)
length(unique(Y))==length(Y)
```

Conclusion : Nous pouvons pas rejeter l'hypothèse "la variable WCTA dépend de la variable Default".

```
#RETA-DEFAULT
X <- data$RETA[data$Default==0]
Y <- data$RETA[data$Default==1]
lillie.test(X)#p-val<5% => on rejette la normalité
lillie.test(Y)#p-val<5% => on rejette la normalité
#cl: nous pouvons pas appliquer le test de Student  t.test(X,Y)
wilcox.test(X,Y)#p-value<5% => on rejette  H0
#=> on ne rejette pas que les 2 echant suivent la mm loi
#conditions:
length(unique(X))==length(X)#false
length(unique(Y))==length(Y)#false
```

Conclusion : malgré que nous n'avons pas validé les hypothèses du test non paramétrique, nous pouvons rejeter l'hypothèse nulle: "la variable RETA dépend de la variable Default"

Nous avons trouvé les mêmes résultats pour pour les variables EBITA_TA, METL et STA.

2.8. Question 8 :

Etudier les relations de corrélations entre les variables est primordial ; en effet nous ne pourrons introduire des variables fortement corrélées entre elles ; le risque serait d'avoir des problèmes de multicolinéarité. La multi-colinéarité se définit comme une relation linéaire entre plusieurs variables.

Il existe ce que l'on appelle la multi-colinéarité parfaite : si nous prenons trois variables par exemple et qu'une des variables soit une combinaison parfaite des autres. Il s'agit d'un phénomène assez rare. Les cas de multi-colinéarités les plus fréquents sont imparfaits. Plusieurs moyens permettent de détecter ce cas de multi-colinéarité tels que

l'instabilité du modèle en augmentant la variance des coefficients de régression, elle peut se détecter de plusieurs façons :

- La suppression d'une des variable entraîne un changement considérable sur les autres variables (signes, coefficients estimés etc.)
- Changements dans la significativité des variables les rendant non significatives alors qu'en réalité elles le sont.
- Estimateur biaisé et donc mauvaises prédictions
- Modèle non robuste
- D'un point de vue mathématiques si deux variables sont très fortement corrélées le déterminant sera proche de 0, ce qui cause des problèmes pour le calcule de l'inverse de la matrice et donc le calcul des coefficients

2.9. Question 9:

Premier modèle : $y = f(WCTA, RETA)$:

```
> model1 <- glm(Default~WCTA+RETA,family=binomial(link='logit'), data=data
> summary(model1)

Call:
glm(formula = Default ~ WCTA + RETA, family = binomial(link = "logit"),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1700   0.1372   0.1604   0.1845   2.3508

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.8226     0.1428  26.766   <2e-16 ***
WCTA          0.7290     0.5911   1.233    0.217
RETA          1.8035     0.1865   9.668   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 721.2  on 3999  degrees of freedom
Residual deviance: 631.5  on 3997  degrees of freedom
AIC: 637.5

Number of Fisher Scoring iterations: 7
```

Pour la constante et la variable RETA on rejette l'hypothèse nulle car notre p-value est inférieure à 5%, c'est à dire que nous rejetons l'hypothèse que nos variables ne soient

pas significatives. En revanche notre variable WCTA a une p-value supérieure à 5%, ainsi nous ne rejetons pas l'hypothèse nulle.

Deuxième modèle : $y = f(METL, EBITTA)$:

```
> model2<- glm(Default~METL+EBIT_TA,family=binomial(link='logit'), data=data)
Warning message:
glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> summary(model2)

Call:
glm(formula = Default ~ METL + EBIT_TA, family = binomial(link = "logit"),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.7133   0.0296   0.1188   0.2237   1.4532

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.8487     0.2181   8.475  < 2e-16 ***
METL          2.2515     0.3453   6.520 7.04e-11 ***
EBIT_TA       8.5452     2.6715   3.199  0.00138 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 721.20  on 3999  degrees of freedom
Residual deviance: 599.43  on 3997  degrees of freedom
AIC: 605.43

Number of Fisher Scoring iterations: 10
```

Nous remarquons que toutes nos variables ont une p-value inférieure à 5%, ainsi nous rejetons l'hypothèse nulle que celles-ci ne soient pas significatives.

Troisième modèle : $y = f(WCTA, RETA, EBITTA, STA)$:

```
> model3<- glm(Default~WCTA+RETA+EBIT_TA+STA,family=binomial(link='logit'), data=data)
> summary(model3)

Call:
glm(formula = Default ~ WCTA + RETA + EBIT_TA + STA, family = binomial(link = "logit"),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1243   0.1294   0.1551   0.1777   2.3826

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   3.7244     0.1999  18.628  < 2e-16 ***
WCTA          0.6062     0.6499   0.933  0.35092
RETA          1.7143     0.1965   8.722  < 2e-16 ***
EBIT_TA       9.5760     2.4344   3.934 8.37e-05 ***
STA          -0.9607     0.3545  -2.710  0.00674 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 721.20  on 3999  degrees of freedom
Residual deviance: 614.76  on 3995  degrees of freedom
AIC: 624.76

Number of Fisher Scoring iterations: 7
```

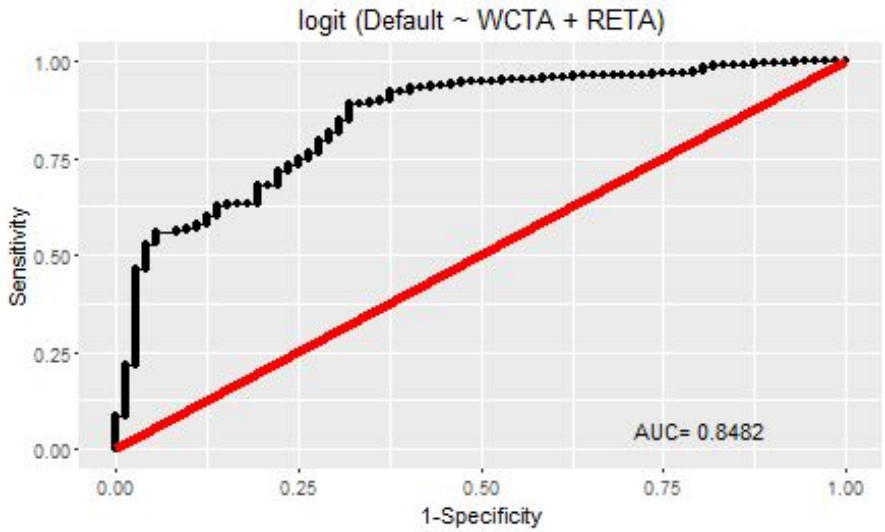
Dans ce modèle seule la variable WCTA a une p-value supérieure à 5%, ainsi pour cette variable nous ne rejetons pas l'hypothèse nulle de non significativité.

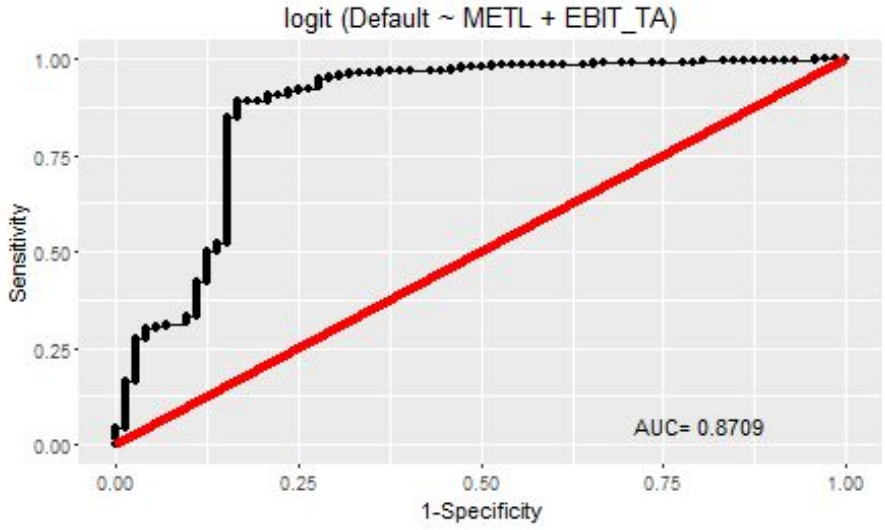
Quatrième modèle : $y = f(WCTA, RETA, EBITTA, METL, STA)$:

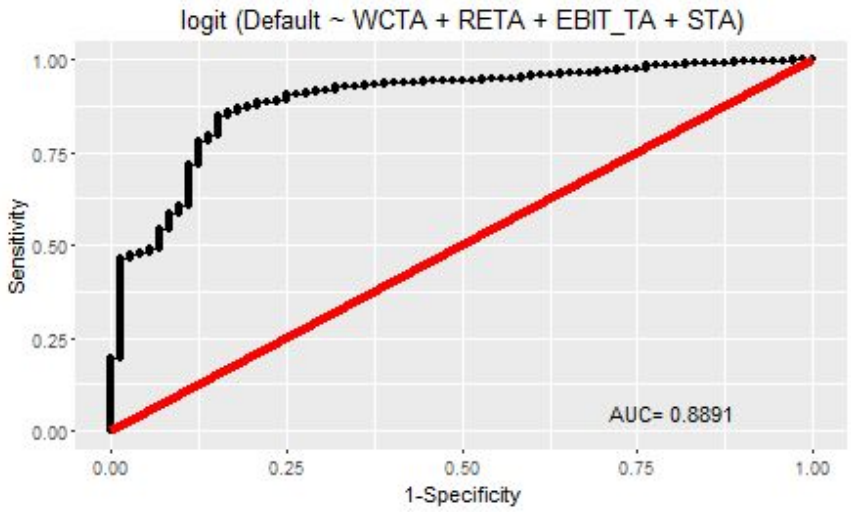
```
> model4<- glm(Default~WCTA+RETA+EBIT_TA+METL+STA,family=binomial(link='logit'), data=data)
Warning message:
glm.fit: des probabilités ont été ajustées numériquement à 0 ou 1
> summary(model4)

Call:
glm(formula = Default ~ WCTA + RETA + EBIT_TA + METL + STA, family = binomial(link = "logit"),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.0377   0.0407   0.1194   0.1971   1.9855

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.5508     0.2660   9.588  < 2e-16 ***
WCTA         -0.4188     0.5735  -0.730  0.46525
RETA          1.4533     0.2293   6.339 2.32e-10 ***
EBIT_TA       7.8839     2.7127   2.906  0.00366 **
METL          1.5917     0.3232   4.925 8.44e-07 ***
STA          -0.6253     0.3490  -1.792  0.07314 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 721.20  on 3999  degrees of freedom
Residual deviance: 561.39  on 3994  degrees of freedom
AIC: 573.39

Number of Fisher Scoring iterations: 10
```

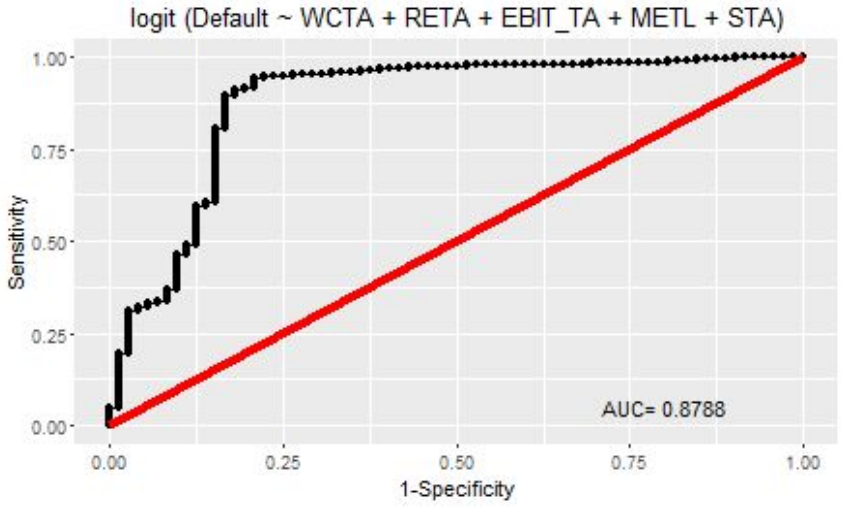
Dans ce modèle les variables WCTA et STA ont des p-value supérieures à 5%, on constate donc qu'en ajoutant METL, la variable STA devient non significatives.

Comparaison des 4 modèles avec la courbe de ROC :









Voilà les les indices AUC "Area under the curve" des quatres modèles :

Modele :	AUC :
$y = f(WCTA, RETA)$:	0.84
$y = f(METL, EBITTA)$:	0.870
$y = f(WCTA, RETA, EBITTA, STA)$:	0.88
$y = f(WCTA, RETA, EBITTA, METL, STA)$:	0.878

Plus AUC est grand, meilleur est le modèle.

On peut conclure que le 3eme modèle est le meilleur.

2.10. Question 10 :

Le **Akaike Information Criterion** (AIC) mesure l'adéquation entre une distribution observée sur un échantillon et une loi de probabilité qui est supposée décrire la population. Le AIC évalue le modèle en fonction de sa vraisemblance tout en pénalisant les modèles avec trop de variables.

Ci-dessous sa formule mathématiques :

$$AIC = 2k - 2\log(L)$$

Avec k le nombre de paramètre du modèle et L la log-vraisemblance des paramètres du modèle.

Bayesian Information Criterion (BIC) a le même principe mais pénalise les modèles trop complexes.

Ci-dessous sa formule mathématiques :

$$BIC = 2\log(L) + klog(n)$$

Avec n le nombre d'observations

	WCTA, RETA	*METL, EBITTA*	*WCTA, RETA, EBITTA, STA*	*WCTA, RETA, EBITTA, METL, STA*
AIC	637.5001	605.4296	624.7637	573.3933
BIC	656.3822	624.3117	656.2339	611.1576

Plus les critères AIC et BIC sont petits, meilleur est le modèle.

On peut conclure que le 4eme modèle est le meilleur modèle.

2.11. Question 11 :

Voilà le code que nous avons utilisé pour obtenir nos prédictions :

```
> x=as.data.frame(t(c(0.6,.25,.45,0.05,0.72)))
> colnames(x) <- c("WCTA","RETA","EBIT_TA","METL","STA")
> predlog=predict(model4,x,type='response')
> res <- ifelse(predlog > 0.5,1,0)
> cat ("la proba de ne pas avoir un défaut est:",predlog ,"Donc notre prédiction est:",res)
la proba de ne pas avoir un défaut est: 0.9970994 Donc notre prédiction est: 1
```

3. La régression logistique sur variables discrétisées :

3.1. Question 1 :

Le but est de construire une nouvelle modalité pour avoire une vue concrète sur les modalités, avec le discrétisation des variables nous donnera un modèle plus robuste.

La régression logistique sur variables discrétisées permet d'éliminer les données extrêmes qui pourraient biaiser le modèle.

La discrétisation est le fait de transformer une variable quantitative en variable qualitative, c'est-à-dire qu'on va créer des classes sur notre variable d'origine. Il faut choisir le nombre de classes et les bornes de classes. Ainsi il faut jauger entre la « perte » d'informations et le nombre de classes (si celui-ci est trop grand il n'y a pas d'intérêt à discrétiser).

L'intérêt de cette démarche est de pouvoir atténuer l'effet des valeurs extrêmes, harmonise les données hétérogènes, facilité d'interprétation lors des résultats, appréhende mieux les relations non linéaires.

Il existe plusieurs méthodes de discrétisations de variables, nous avons décidés de faire des arbres de décisions.

3.2. Question 2 :

L'identifiabilité est une étape primordiale, en effet c'est une hypothèse en modélisation statistique qui permet de "garantir" la consistance des estimateurs.

Lorsque l'on discrétise des variables l'étape suivante est de créer des variables binaire or si nous entrons toutes nos variables issues d'une seule variable il est mathématiquement impossible d'inverser la matrice et donc de trouver des estimateurs (cause : colinéarité parfaite). L'identification est donc un des piliers en statistiques pour pouvoir trouver des estimateurs qui ont du sens et donner une interprétation en faisant référence à notre variable identifiée.

Exemple si nous décidons de binariser la variable genre, nous aurons deux variables une femme qui prendra la valeur 1 lorsque la condition est vérifiée et 0 sinon, la variable homme prendra donc les valeurs inverses. Si nous choisissons de conserver la variable femme le coefficient de la régression sera interprété par rapport à notre variable de référence (homme), ainsi on pourra dire par exemple qu'une femme à N% moins de chance de mourir avant 80 ans qu'un homme.

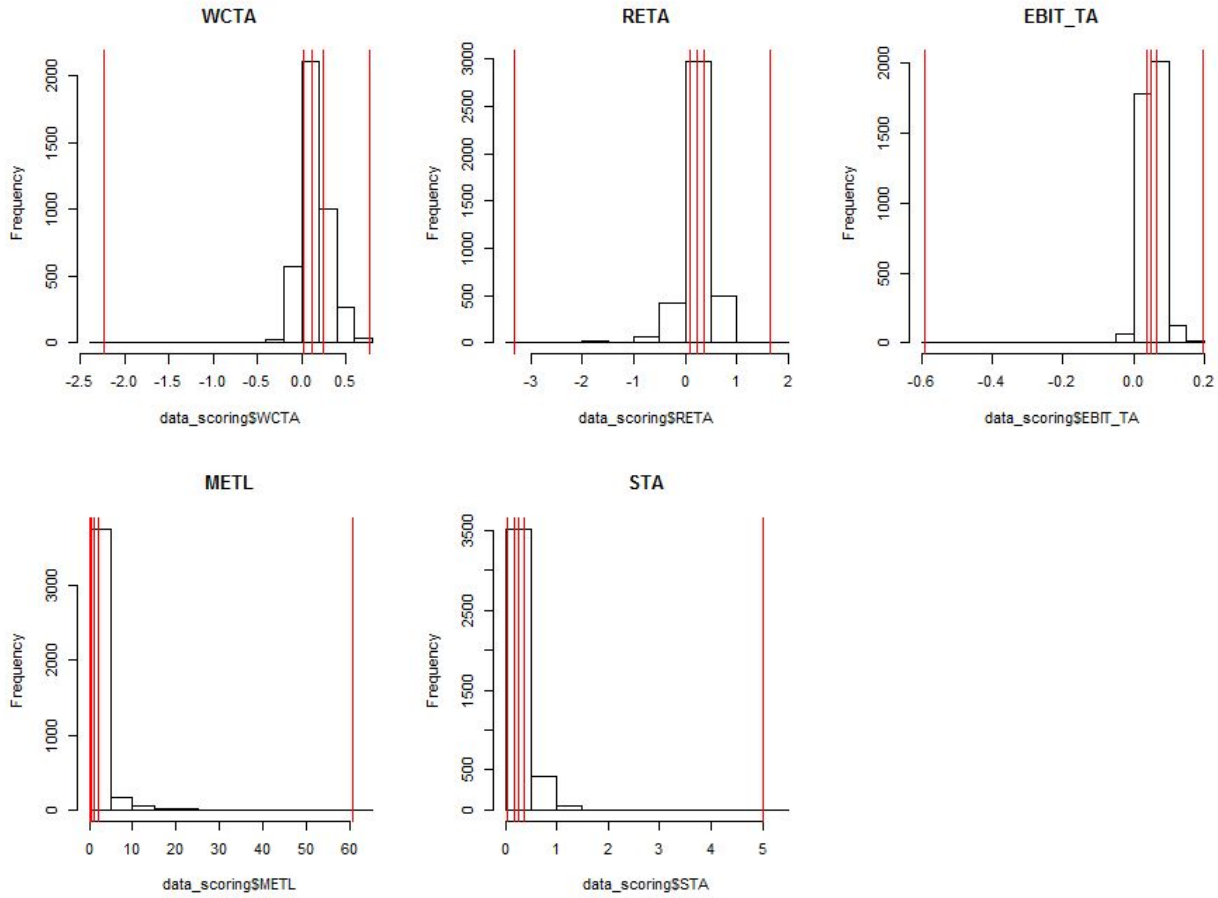
3.3. Question 3 : Discrétisation des variables

Discrétiser une variable quantitative c'est transformer un vecteur de nombres réels en un vecteur d'indices des classes, tel que chaque valeur réelle va appartenir à une classe. La problématique ici est évidente, c'est de trouver le nombre des classes et les seuils pour faire la transformation adéquate.

Première méthode : En fonction de la distribution

Dans un premier temps nous avons décidé de visualiser la distribution de nos variables et d'essayer de détecter visuellement les différentes classes.

Ci dessous les histogrammes représentant la distribution de nos 5 variables, les traits en rouge représentent les quantiles :



On a fait le choix ci dessous

Pour la variable WCTA :

```
> summary(data_scoring$WCTA.d)
 [-2.24,0]     (0,0.4] (0.4,0.766]
       603        3099         298
```

Pour la variable RETA :

```
> summary(data_scoring$RETA.d)
 [-3.31,0]     (0,0.5] (0.5,1.64]
       530        2967         503
```

Pour la variable EBIT_TA :

```
> summary(data_scoring$EBIT_TA.d)
 [-0.592,0]     (0,0.1] (0.1,0.198]
         82        3782         136
```

Pour la variable METL :

```
> summary(data_scoring$METL.d)
[0.0237,5]   (5,60.6]
      3753         247
```

Pour la variable STA :

```
> summary(data_scoring$STA.d)
[0.0359,0.5]   (0.5,5.01]
        3518          482
```

Deuxième méthode : Avec les arbres de décisions

Sachant que les arbres de décisions se basent sur le principe de l'entropie pour faire la séparation des noeuds, nous avons décidé de nous baser sur ce principe pour définir les seuils de discrétisation.
Pour chaque variable nous allons construire un arbre de décision, pour définir le seuil de discrétisation.

Par exemple: pour la variable WCTA :

```
> arbre_WCTA=rpart(Default~WCTA,data =data_scoring)
> arbre_WCTA
n= 4000

node), split, n, deviance, yval
      * denotes terminal node

1) root 4000 70.704000 0.9820000
  2) WCTA< -0.6513918 7  1.714286 0.5714286 *
  3) WCTA>=-0.6513918 3993 67.807660 0.9827198 *
```

Le problème avec cette séparation c'est que si on choisit la valeur -0.6513918 de WCTA comme valeur de séparation, on aura deux classes : la première va contenir uniquement 7 individus, la deuxième 3993, ce qui nous laisse penser que cette séparation n'est peut être pas la plus pertinente

Troisième méthode : Clustering Hiérarchique

#Dans cette partie nous avons fait le choix de séparer chacune de nos variables en 3 classes.

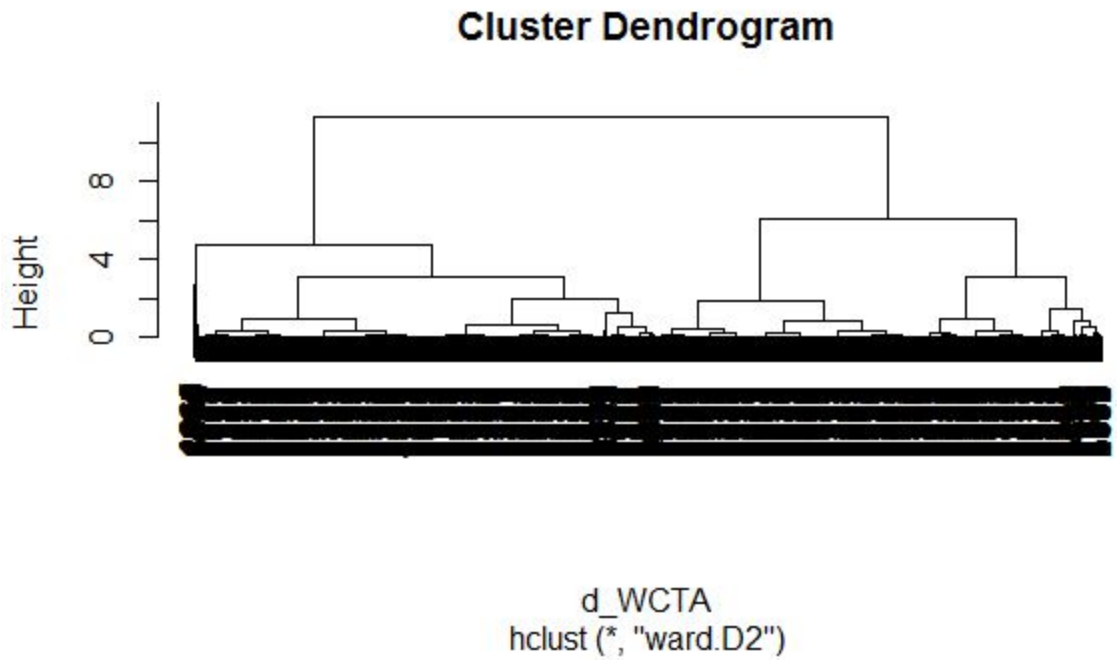
La classification ascendante hiérarchique (CAH) est une méthode de classification itérative dont le principe est simple.

1. On commence par calculer la dissimilarité entre les N objets.
2. Puis on regroupe les deux objets dont le regroupement minimise un critère d'agrégation donné, créant ainsi une classe comprenant ces deux objets.
3. On calcule ensuite la dissimilarité entre cette classe et les N-2 autres objets en utilisant le critère d'agrégation. Puis on regroupe les deux objets ou classes d'objets dont le regroupement minimise le critère d'agrégation.

On continue ainsi jusqu'à ce que tous les objets soient regroupés.

Pour la variable WCTA :

```
d_WCTA=dist(data_scoring$WCTA)
clust_WCTA=hclust(d_WCTA,method="ward.D2")
plot(clust_WCTA)
```

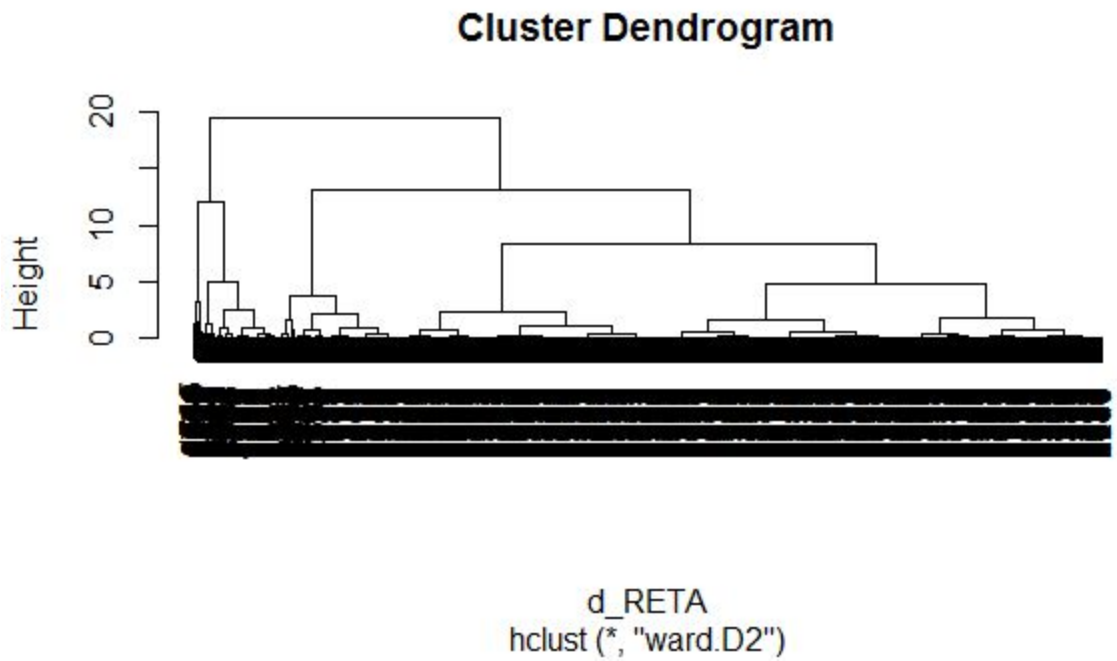


Graphiquement on peut dire que le nombre optimal de clusters est 2 ou 3.

On choisit de séparer notre variable en 3 classes :

```
> length(which(data_scoring$WCTA_CAH==1))
[1] 793
> length(which(data_scoring$WCTA_CAH==2))
[1] 2026
> length(which(data_scoring$WCTA_CAH==3))
[1] 1181
```

Pour la variable RETA :

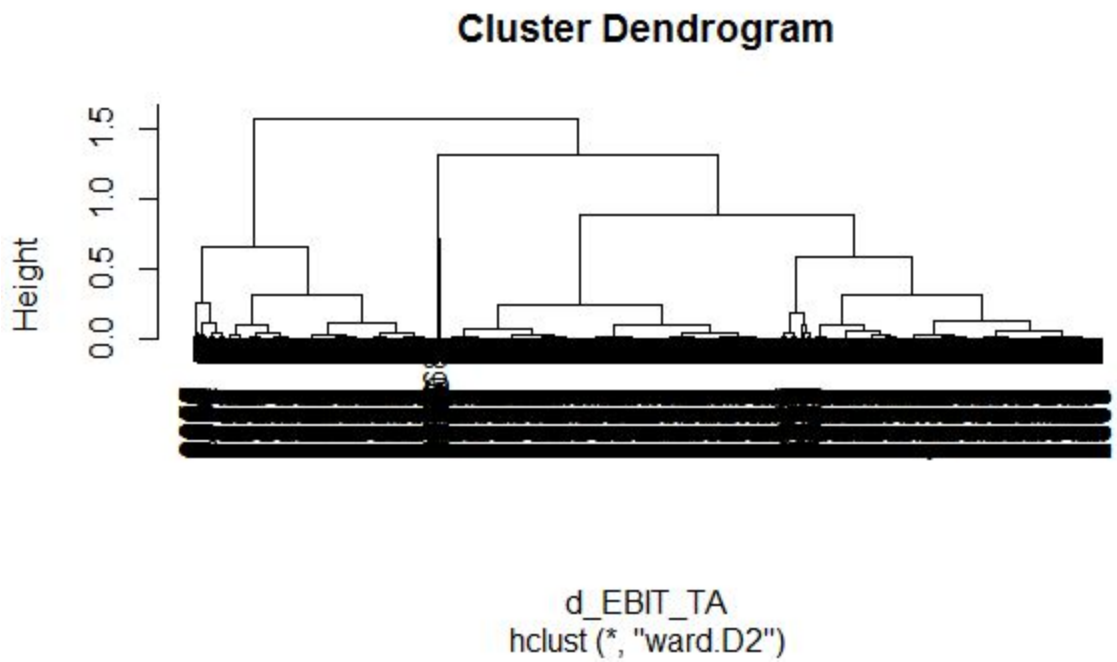


Graphiquement on peut dire que le nombre optimal de clusters est 3 ou 4.

On choisit de séparer notre variable en 3 classes :

```
> length(which(data_scoring$RETA_CAH==1))
[1] 3085
> length(which(data_scoring$RETA_CAH==2))
[1] 375
> length(which(data_scoring$RETA_CAH==3))
[1] 540
```

Pour la variable EBIT_TA :

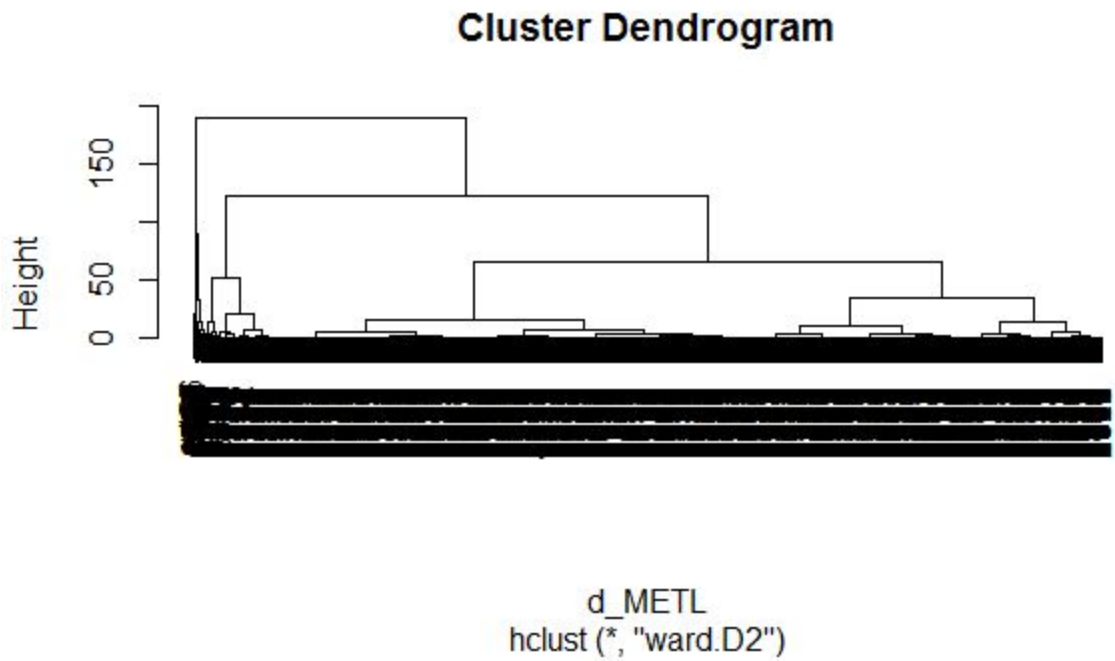


Graphiquement on peut dire que le nombre optimal de clusters est 2 ou 3.

On choisit de séparer notre variable en 3 classes :

```
> length(which(data_scoring$EBIT_TA_CAH==3))
[1] 17
> length(which(data_scoring$EBIT_TA_CAH==2))
[1] 1069
> length(which(data_scoring$EBIT_TA_CAH==1))
[1] 2914
```

Pour la variable METL :

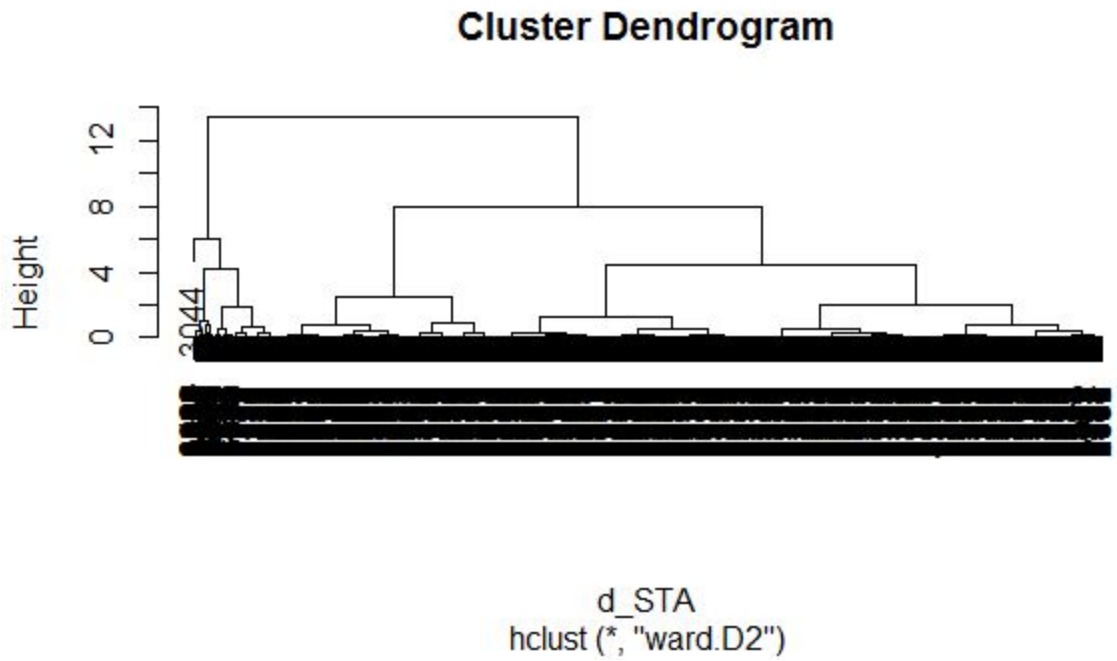


Graphiquement on peut dire que le nombre optimal de clusters est 2 ou 3.

On choisit de séparer notre variable en 3 classes :

```
> length(which(data_scoring$METL_CAH==1))
[1] 3629
> length(which(data_scoring$METL_CAH==2))
[1] 324
> length(which(data_scoring$METL_CAH==3))
[1] 47
```

Pour la variable STA :



Graphiquement on peut dire que le nombre optimal de clusters est 2 ou 3.

On choisit de séparer notre variable en 3 classes :

```
> length(which(data_scoring$STA_CAH==1))
[1] 970
> length(which(data_scoring$STA_CAH==2))
[1] 2674
> length(which(data_scoring$STA_CAH==3))
[1] 356
```

On décide de garder la discrétisation des variables faite par la méthode du clustering.

3.4. Question 4 :

On souhaite étudier la corrélation entre nos variables discrétisées. Pour ceci le simple test de corrélation de **Pearson** n'est pas adéquat car nos variables sont catégorielles.

Pour réaliser notre étude, on va utiliser :

Le test de **KHI-deux** : si l'on suppose que deux variables sont indépendantes, les valeurs du tableau de contingence pour ces variables doivent être réparties uniformément.
L'hypothèse nulle : les variables sont indépendants.
L'hypothèse alternative : il existe une corrélation entre le variables.

Il existe également le **V de Crammer** qui est une mesure de corrélation qui vient du test de Khi-deux.

Ainsi, nous exécutons le test Khi-deux et la p-valeur résultante, peut être considérée comme une mesure de la corrélation entre ces deux variables.

On réalise le test entre chaque pair de variable, et on obtient les p-values suivantes :

	WCTA_CAH	RETA_CAH	EBIT_TA_CAH	METL_CAH
WCTA_CAH				
RETA_CAH	2.2e-16			
EBIT_TA_CAH	1.318e-12	2.2e-16		
METL_CAH	2.2e-16	2.2e-16	2.2e-16	
STA_CAH	2.2e-16	3.09e-09	2.2e-16	0.0505

On rejette l'hypothèse d'indépendance de toute paire de variable (p_valeur <0.05) sauf pour la paire : STA_CAH et METL_CAH

3.5. Question 5 :

Avant de travailler avec les nouvelles variables discrétisées, il est important d'appliquer la fonction factor de R, pour que par la suite ces variables soient interprétés comme des variables qualitatives :

Pour connaître l'ordre des modalités d'une variable de type facteur, on peut utiliser la fonction **frequencies**, prenons par exemple la variable WCTA_CAH :

```
-----------------------------------------------------------------
--                         Frequencies                         --
--                                                             --
  Value # of Cases       % Cumulative %
1     1         793    19.8         19.8
2     2        2026    50.6         70.5
3     3        1181    29.5        100.0
--                                                             --
```

Dans notre exemple, la modalité « 2 » n'est pas la première modalité, il est donc nécessaire de modifier notre variable, il faudra modifier la modalité de référence avec la fonction **relevel**

En effet, dans un modèle, tous les coefficients sont calculés par rapport à la modalité de référence. Il est important de choisir une modalité de référence qui fasse sens afin de faciliter l'interprétation. De manière générale on évitera de choisir comme référence une modalité peu représentée dans l'échantillon ou bien une modalité correspondant à une situation atypique.

```
> data_scoring$WCTA_CAH <- relevel(data_scoring$WCTA_CAH, "2")
```

Et on obtient :

```
-----------------------------------------------------------------
--                         Frequencies                         --
--                                                             --
  Value # of Cases       % Cumulative %
1     2        2026    50.6         50.6
2     1         793    19.8         70.5
3     3        1181    29.5        100.0
--                                                             --
```

En appliquant la méthode de step wise pour les 3 directions: backwards, forwards et bothways, on obtient les modèle suivants :

```
> formula(backwards)
Default ~ RETA_CAH + EBIT_TA_CAH + STA_CAH
> formula(forwards)
Default ~ RETA_CAH + STA_CAH + EBIT_TA_CAH
> formula(bothways)
Default ~ RETA_CAH + STA_CAH + EBIT_TA_CAH
```

On remarque clairement que pour les 3 directions, l'algorithme ne prend pas les variables WCTA et METL discrétisées, nous pouvons donc dire que ces variables n'expliquent pas notre variable cible: Default.

Les 3 modèles retenus pour chaque direction représentent bien les modèles qui réduisent au plus la valeur de AIC et de BIC.

On obtient finalement donc notre meilleur modèle avec les meilleurs valeurs de AIC et BIC :

AIC = 593.819, et BIC =637.8773

3.6. Question 6 :

Dans cette partie nous allons nous intéresser aux modèles estimés lors de la méthode stepwise pour la direction backwards au fil des itérations :

```
> backwards=step(fullmod,direction = "backward")
Start:  AIC=597.7
Default ~ WCTA_CAH + RETA_CAH + EBIT_TA_CAH + METL_CAH + STA_CAH

              Df Deviance    AIC
- WCTA_CAH     2   575.93 593.93
- METL_CAH     2   579.68 597.68
<none>             575.70 597.70
- EBIT_TA_CAH  2   579.94 597.94
- STA_CAH      2   581.24 599.24
- RETA_CAH     2   691.18 709.18

Step:  AIC=593.93
Default ~ RETA_CAH + EBIT_TA_CAH + METL_CAH + STA_CAH

              Df Deviance    AIC
- METL_CAH     2   579.82 593.82
<none>             575.93 593.93
- EBIT_TA_CAH  2   580.13 594.13
- STA_CAH      2   581.50 595.50
- RETA_CAH     2   691.74 705.74

Step:  AIC=593.82
Default ~ RETA_CAH + EBIT_TA_CAH + STA_CAH

              Df Deviance    AIC
<none>             579.82 593.82
- EBIT_TA_CAH  2   584.70 594.70
- STA_CAH      2   585.74 595.74
- RETA_CAH     2   695.33 705.33
```

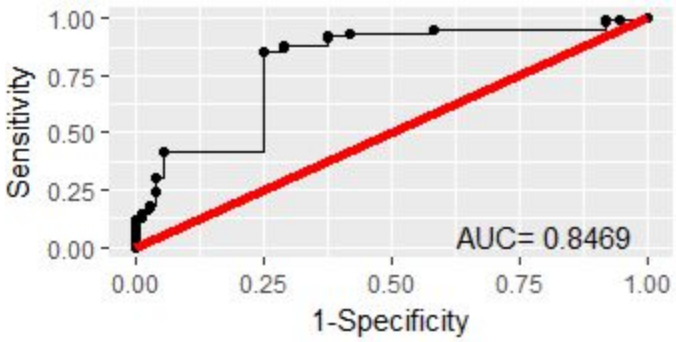
Pour la direction backward, l'algorithme part du modèle full : c'est-à-dire le modèle contenant toutes les variables, et au fur des itérations il supprime les variables,

```
> AIC(iter1)  > BIC(iter1)
[1] 597.6993  [1] 666.9338
> AIC(iter2)  > BIC(iter2)
[1] 593.9292  [1] 650.5756
> AIC(iter3)  > BIC(iter3)
[1] 593.819   [1] 637.8773
```

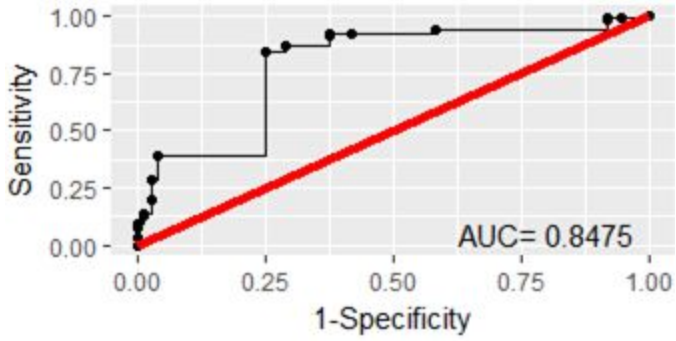
dont la suppression minimise la valeur de l'AIC et BIC :

Intéressons-nous maintenant aux courbes de ROC calculées à partir des modèles obtenus au fil des itérations :

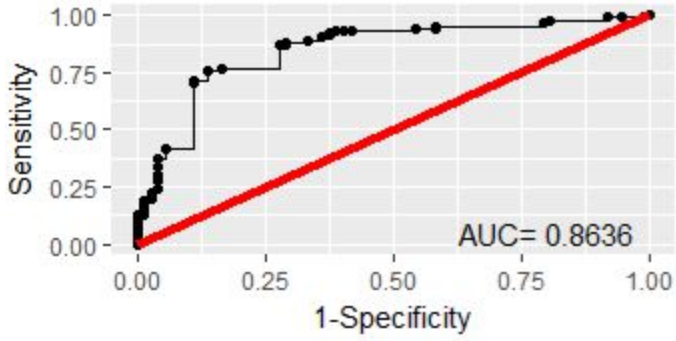
Itération 1 :



Itération 2:



Itération 3:



On note bien l'évolution de la valeur de l'AUC au fil des itérations.

Faisons maintenant la comparaison avec le meilleur modèle obtenu avec les variables quantitatives avant discrétisation (Partie A) :

	Modèle avec variables Quantitative	Modèle avec variables discrétisées
AIC	573.3933	593.819
BIC	611.1576	637.8773
AUC	0.8636	0.878

3.7. Question 7 :

Dans cette partie nous allons travailler avec le meilleur modèle obtenu précédemment :

Intercept	**4.7132110**
RETA_CAH2	-2.7000127
RETA_CAH3	1.2132688
EBIT_TA_CAH2	0.8035541
EBIT_TA_CAH3	-0.5579324
STA_CAH1	0.1138155
STA_CAH3	-0.8292011

EXP(Coef) = Odds-ratio => surcroit de chances d'être positif

Par rapport à ceux qui ont la modalité 1 pour la variable RETA_CAH, ceux qui ont la modalité 2 ont
14.87973 (1/exp(-2.70)) fois plus de chance de faire défaut.

Etape 1 :

On fait apparaître les variables de référence :

RETA_CAH1	**0**
RETA_CAH2	-2.7000127
RETA_CAH3	1.2132688
EBIT_TA_CAH1	0
EBIT_TA_CAH2	0.8035541
EBIT_TA_CAH3	-0.5579324
STA_CAH1	0
STA_CAH1	0.1138155
STA_CAH3	-0.8292011

Etape 2 :

Détection des valeurs min des coefficients par variable (la constante est mise de côté)

Min_RETA_CAH=-2.7000127

Min_EBIT_TA_CAH=-0.5579324

Min STA_CAH=-0.8292011

Etape 3 :

Correction par variable pour rendre positifs tous les coefficients

Coef + |Min_variable|

RETA_CAH1	**2.7000127**
RETA_CAH2	0
RETA_CAH3	3.913282
EBIT_TA_CAH1	0.5579324
EBIT_TA_CAH2	1.361487
EBIT_TA_CAH3	0
STA_CAH2	0.8292011
STA_CAH1	0.9430166

STA_CAH3	0

Ainsi, les points attribués seront toujours positifs.

Le minimum des points est égal à 0.

Etape 4 :

Identifier le maximum des points :

Max_RETA_CAH= 3.913282

Max_EBIT_TA_CAH= 1.361487

Max STA_CAH=0.9430166

Max_point = 3.913282+1.361487+0.9430166 =6.217786

Etape 5 :

Calculer le facteur de correction h (echelle 1000)

h =1000/Max_point= 160.829

Etape 6 :

Multiplier les points modalités par le facteur de correction h

	SCORE
RETA_CAH1	434
RETA_CAH2	0
RETA_CAH3	629
EBIT_TA_CAH1	90
EBIT_TA_CAH2	219
EBIT_TA_CAH3	0
STA_CAH2	133
STA_CAH1	152
STA_CAH3	0

160.829*2.7000127 = 434.2403

Les notes par modalité sont arrondies pour faciliter la lecture.

Le score est calibré, il est compris entre 0 et 1000

Annexes :

1. Question 6 : (sortie SAS proc univariate)

Code SAS :

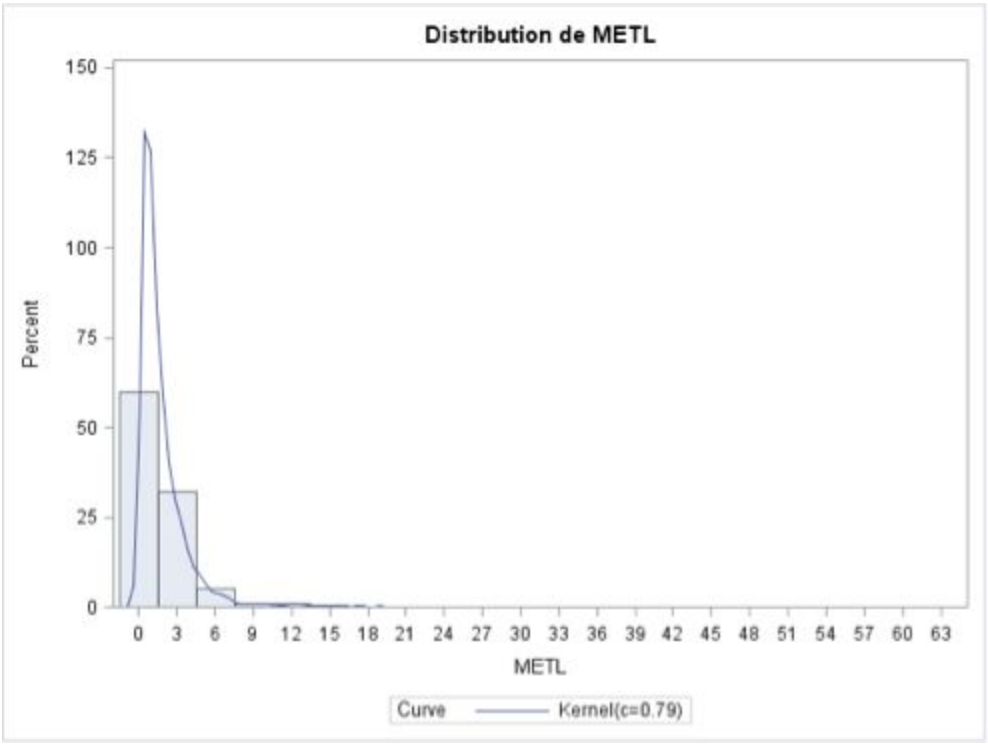
```
proc univariate data=scoring;
var wcta reta ebit_ta metl sta;
histogram wcta reta ebit_ta metl sta / normal ;
run;
```

Sorites SAS :

METL :

Observations extrêmes			
Le plus bas		Le plus haut	
Valeur	Obs	Valeur	Obs
0.0237178	1615	28.3772	2730
0.0272822	3992	41.2630	1201
0.0280297	1083	48.1661	196
0.0297572	734	58.2408	194
0.0307504	3991	60.6072	195

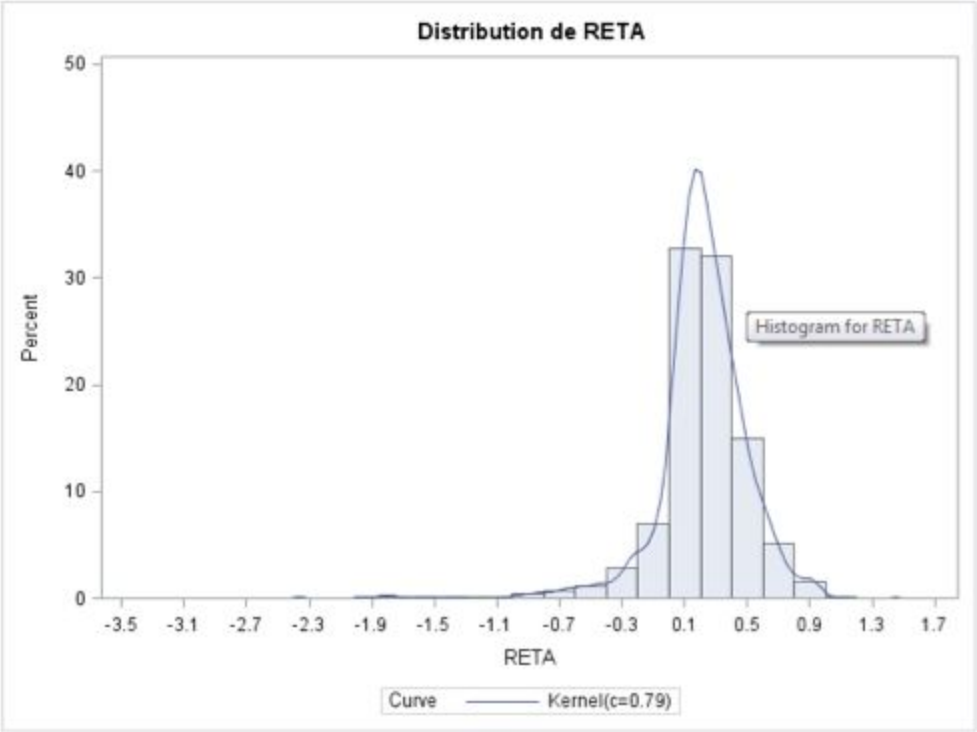
Moments			
N	4000	Somme des poids	4000
Moyenne	1.95396271	Somme des observations	7815.85084
Ecart-type	2.994475	Variance	8.96688051
Skewness	7.75072812	Kurtosis	103.127537
Somme des carrés non corrigée	51130.4363	Somme des carrés corrigée	35858.5552
Coeff Variation	153.251389	Std Error Mean	0.04734681



RETA :

Variable : RETA (RETA)

Moments			
N	4000	Somme des poids	4000
Moyenne	0.21046709	Somme des observations	841.868347
Ecart-type	0.33254945	Variance	0.11058913
Skewness	-2.5548915	Kurtosis	17.441196
Somme des carrés non corrigée	619.431527	Somme des carrés corrigée	442.245949
Coeff Variation	158.00544	Std Error Mean	0.00525807



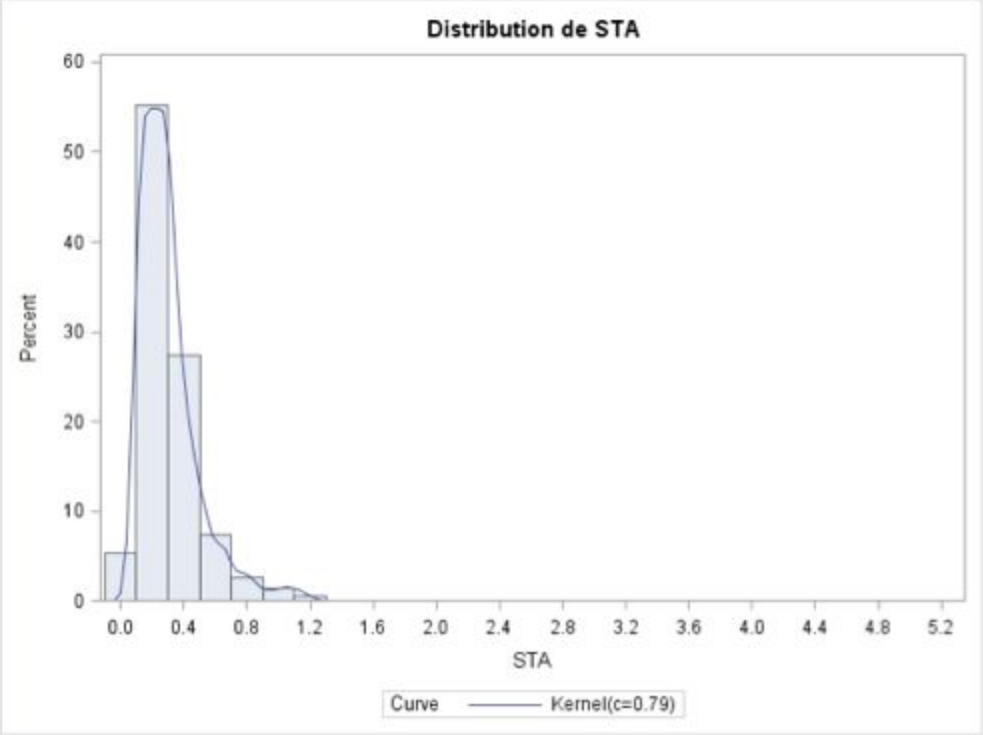
Observations extrêmes			
Le plus bas		Le plus haut	
Valeur	Obs	Valeur	Obs
-3.31242	710	1.32474	237
-3.05114	631	1.42506	234
-2.70772	1615	1.42696	236
-2.40186	3968	1.51017	2033
-2.37359	632	1.63965	235

STA :

Variable : STA (STA)

Moments			
N	4000	Somme des poids	4000
Moyenne	0.30365176	Somme des observations	1214.60703
Ecart-type	0.20579405	Variance	0.04235119
Skewness	4.48100441	Kurtosis	71.2152414
Somme des carrés non corrigée	538.179965	Somme des carrés corrigée	169.362405
Coeff Variation	67.773046	Std Error Mean	0.00325389

Observations extrêmes			
Le plus bas		Le plus haut	
Valeur	Obs	Valeur	Obs
0.0358583	1046	1.31931	391
0.0362116	60	1.43186	1528
0.0381004	1326	1.47393	392
0.0386720	3820	1.58570	1527
0.0417286	3972	5.00778	3044



WCTA :

Procédure UNIVARIATE
Variable : WCTA (WCTA)

Moments			
N	4000	Somme des poids	4000
Moyenne	0.14251368	Somme des observations	570.054705
Ecart-type	0.17076261	Variance	0.02915987
Skewness	-1.0145317	Kurtosis	17.6822686
Somme des carrés non corrigée	197.850908	Somme des carrés corrigée	116.610316
Coeff Variation	119.821911	Std Error Mean	0.00269999

Observations extrêmes			
Le plus bas		Le plus haut	
Valeur	Obs	Valeur	Obs
-2.240268	1615	0.725125	3237
-2.041123	3968	0.736556	3235
-0.994230	3999	0.744054	3232
-0.899241	3992	0.753184	3231
-0.861508	734	0.766040	3230