SY09 - TP4 Discrimination

Matthieu PETIT - Pierre-Louis LACORTE

11 juillet 2017

Objectif

Le but de ce TP est de réaliser une application mettant en œuvre la l'analyse discriminante (binaire) ainsi que les régressions logistiques binaire et quadratique. Elles permettent de réaliser des prédictions à partir d'observations. Nous testerons ensuite ces fonctions sur des jeux de données fournis afin d'observer leur validité.

Table des matières

1 Programmation

Dans un premier temps, nous allons programmer les diverses fonctions qui nous sont demandées dans ce TP.

1.1 Analyse discriminante

Dans le cadre de l'implémentation de l'analyse discriminante, nous allons compléter quatre fonctions permettant d'effectuer cette étude sur un ensemble de données. Les trois premières fonctions font l'apprentissage des trois modèles, c'est à dire :

— l'analyse discriminante quadratique : elle constitue le cas général où la distribution de chaque variable dans chacune des classes est caractérisée par des paramètres μ_k et Σ_k différents. On obtient les paramètres à l'aide des formules suivantes :

$$\widehat{\pi_k} = \frac{n_k}{n}$$

$$\widehat{\mu_k} = \frac{1}{n_k} \sum_{i=1}^{n} z_{ik} xi$$

$$\widehat{\Sigma_k} = \frac{1}{n_k} \sum_{i=1}^{n} z_{ik} (x_i - \widehat{\mu_k})(x_i - \widehat{\mu_k})^t$$

— l'analyse discriminante linéaire : elle suppose elle vrai l'hypothèse d'homoscédasticité (les matrices de covariances sont égales) sur les paramètres présentés précédemment.

— le classifieur bayésien naïf : une hypothèse courante que vérifie le classifieur bayésien naïf est que les variables de X sont indépendantes conditionnelles à leur appartenances à des classes différentes et que les matrices Σ_k sont diagonales et égales entre classe.

Finalement, nous avons réalisé une fonction permettant d'effectuer la validation d'un ensemble de test à partir du modèle créé par ces trois fonctions. Cette fonction renvoie les probabilités *a posteriori* de l'ensemble de test ainsi que les étiquettes prédites pour cet ensemble.

1.2 Régression logistique

Dans le cadre de l'implémentation de la régression logistique binaire, nous allons compléter deux fonctions. La première d'entre elles permet de faire l'apprentissage du modèle et la seconde applique le modèle obtenu sur un ensemble de données test.

1.2.1 Apprentissage

Pour l'apprentissage des paramètres on utilise la méthode de Newton-Raphson qui se base sur la maximisation d'un maximum local de la log-vraissemblance. On traite dans notre cas d'exemples ne présentant que 2 classes. On utilise donc les probabilités *a priori* :

$$P(w_1|x) = p(x, w) = \frac{\exp(w^t x)}{1 + \exp(w^t x)}$$

$$P(w_2|x) = 1 - p(x, w)$$

On selectionne un vecteur de poids initial $w^{(0)}$ et on essaie de le faire converger de telle manière que $w^{(q+1)}$ et $w^{(q)}$ soit inférieur à un seuil qu'on défini (ici $10e^{-5}$). $w^{(q+1)}$ est obtenu à l'aide de :

$$w^{(q+1)} = w^{(q)} + (X^T W_{(q)} X)^{-1} X^T (\mathbf{t} - \mathbf{p}^{(q)}) \tag{1}$$

On donne aussi la possibilité de pouvoir ajouter une ordonnée à l'origine pour ajouter de la flexibilité au modèle. Finalement on récupère grâce à cette fonction l'estimateur du maximum de vraissemblance local $\widehat{\beta}$.

1.2.2 Validation

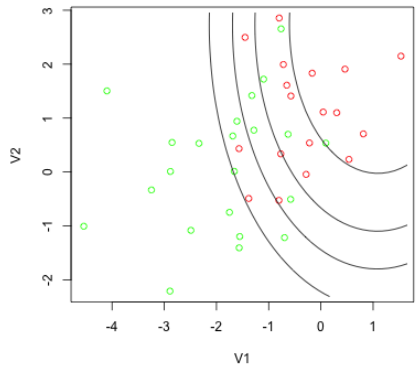
Une fois que l'on a l'estimateur du maximum de vraissemblance local on peut prédire les étiquettes des individus en comparant les probabilités *a priori* de chaque classe.

1.2.3 Généralisation

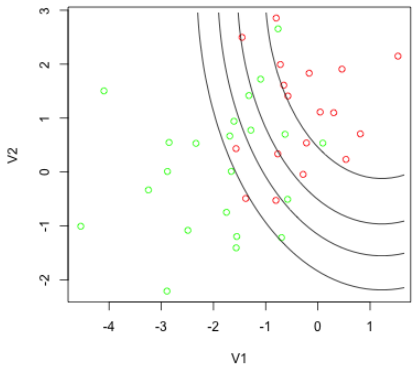
Une fois ces deux fonctions obtenue, il nous est aisé de généraliser ce modèle à la régression linéaire quadratique. Nous transformons les données en un espace plus complexe dans lesquelles les données peuvent être séparées par un hyperplan. Dans les faits cela consiste à déduire des n variables données, un nombre $p = n(n+3)/2$ qui, lorsque $n = 3$ et $X = \{X_1, X_2, X_3\}$, peuvent être définies par : $X^2 = \{X_1, X_2, X_3, X_1X_2, X_1X_3, X_2X_3, X_1^2, X_2^2, X_3^2\}$.

1.3 Vérification des fonctions

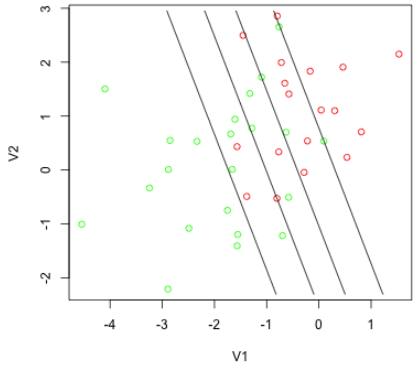
Afin de vérifier les données que nous obtenons, nous allons comparer nos graphiques avec ceux fournis dans le sujet du TP. Nous avons représenté les données avec $niveaux = \{0.2, 0.4, 0.6, 0.8\}$ dans la figure 1. Nous observons ainsi que les graphiques obtenus avec nos fonctions sont les mêmes que ceux attendus, nos fonctions semblent donc correspondre elles aussi aux fonctions attendues dans le cadre de ce TP.



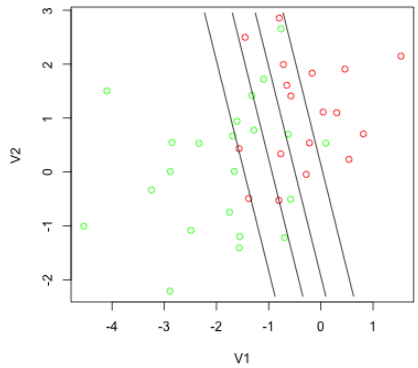
(a) Analyse discriminante quadratique



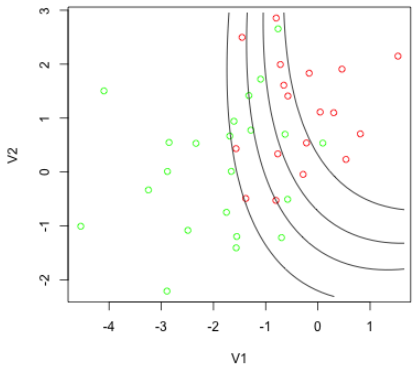
(b) Classifieur bayésien naïf



(c) Analyse discriminante linéraire



(d) Régression logistique



(e) Régression logistique quadratique

FIGURE 1 – Frontières de décision obtenues en utilisant toutes les données pour l'apprentissage (ensemble de données Synth1_40)

2 Application

Nous allons maintenant appliquer nos fonctions sur des ensembles de données fournies, afin d'en faire une analyse. Les ensembles de données concernés sont :

— *Synth1_1000, Synth2_1000, Synth3_1000* (Données simulées)
— *Pima* (Individus indiens diabétiques ou non)
— *Breastcancer* (Individus atteint du cancer du sein ou non)
— *Spam* (Email classés spam ou non)

2.1 Test sur données simulées

Nous commençons par tester les fonctions sur des ensembles de données simulées. Nous obtenons les taux d'erreurs de la table 1 pour un nombre $N = 20$ de séparations effectuées. Le protocole de traitement que nous utilisons est propre de celui utilisé durant le TP3. Dans un premier temps nous séparons le jeu de données en 2 : un jeu d'appretissage et un jeu de test. Nous réalisons l'apprentissage à partir du premier jeu et mesurons le taux d'erreur sur le jeu de test.

en %	Taux d'erreur par fonction					
Données	ADQ	ADL	NBA	LOG	LOGQ	Tree
Synth1_40	22.69231	22.30769	23.46154	19.23077	18.84615	16.11111
Synth1_1000	3.188623	4.520958	4.236527	3.637725	3.502994	2.484985
Synth2_1000	6.171171	7.747748	6.171171	6.741742	5.885886	4.287856
Synth3_1000	4.429429	4.384384	4.909910	4.084084	4.129129	3.253373

TABLE 1 – Taux d'erreur en pourcentages selon les différentes fonctions pour différents ensembles de données générés. (ADQ : Analyse discriminante quadratique, ADL : Analyse discriminante linéraire, NBA : Classifieur bayésien naïf, LOG : Régression logistique, LOGQ : Régression logistique quadratique et Tree : arbre de décision)

Nous observons ainsi qu'avec les données générées Synth1_1000, Synth2_1000 et Synth3_1000 nous avons un taux d'erreur relativement faible ($< 10\%$) alors que les données Synth_40 nous avons un taux d'erreur plus élevé (avoisinnant les 20%). Nous pouvons en déduire que plus l'ensemble de données est grand moins l'erreur sera importante. De plus on constate que l'analyse discrimitante renvoit majoritairement une frontière de décision avec moins d'erreur. Ceci vient de son hypothèse de définition même. On ne simplifie pas les variances, le modèle est donc plus complexe et ainsi plus proche de la réalité. D'où le taux d'erreur plus faible. Finalement nous ne présentons ici que les résulats des regressions logistiques avec ajout de l'ordonnée à l'origine puisque les classes ne sont pas représentées autour de l'ordonnée à l'origine du repère et cela permet donc de corriger ce problème et d'obtenir des taux d'erreur plus faible. Les résultats obtenus sur les données simulées Synth1-1000 et Synth2-1000 sont représentés respectivements sur les Figures 2, 4 et Figure 3, 5.

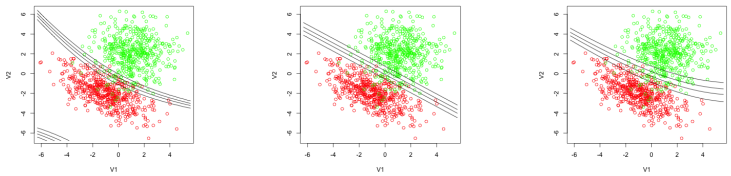


FIGURE 2 – Frontière de décision de l'analyse discriminante (ADQ, ADL, NBA) sur **Synth1-1000** avec des frontières = $\{0.2, 0.4, 0.6, 0.8\}$

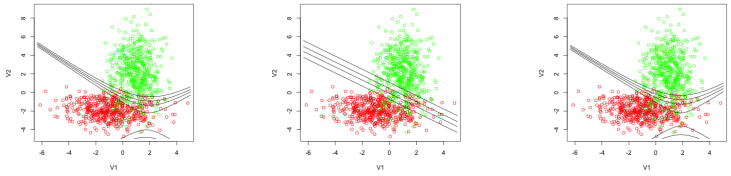


FIGURE 3 – Frontière de décision de l'analyse discriminante (ADQ, ADL, NBA) sur **Synth2-1000** avec des frontières = $\{0.2, 0.4, 0.6, 0.8\}$

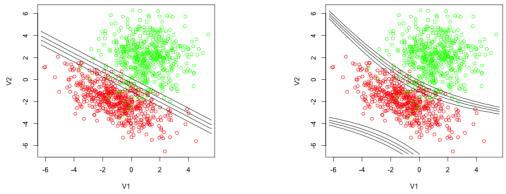


FIGURE 4 – Frontière de décision obtenu par regression linéaire et regression linéaire quadratique **Synth1-1000** avec des frontières = $\{0.2, 0.4, 0.6, 0.8\}$

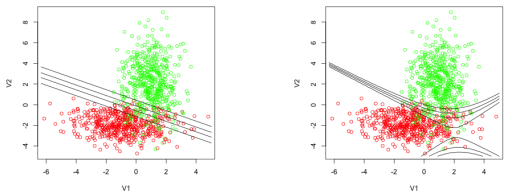


FIGURE 5 – Frontière de décision obtenu par regression linéaireet regression linéaire quadratique **Synth2-1000** avec des frontières = $\{0.2, 0.4, 0.6, 0.8\}$

Les taux d'erreurs obtenus avec les différentes méthodes de discrimination sont "constants" entre les différents jeux de données. C'est-à-dire que l'ADQ sera plus précise que l'ADL dans chacun des cas, etc... Une autre remarque qui peut être soulevée est que le jeu Synth2-1000 présente un taux d'erreur plus élévé que les autres. Comment le justifier ? On remarque avec les Figures 2 et 3 que les variances des 2 classes sont très différentes dans le cas de Synth2 alors qu'elles sont proches pour Synth1. Il est donc plus complexe pour les modèles d'avoir une discrimination de qualité, et donc de ne pas réaliser d'erreur. On notera que dans le cas de Synth2 l'utilisation des méthodes de regression logistique sont plus justifiées puisque les hypothèses d'égalité de variance sont moins vérifiées que sur les autres jeux de données.

2.2 Test sur données réelles

Nous allons maintenant appliquer les fonctions que nous avons définies à trois jeux de données réels.

en %	Taux d'erreur par fonction					
Données	ADQ	ADL	NBA	LOG	LOGQ	Tree
Pima	22.23164	21.11864	23.15254	21.37288	21.05650	14.93220
Breastcancer	4.798246	4.087719	3.837719	3.565789	X	3.653509
Spam	X	X	X	X	X	X

TABLE 2 – Taux d'erreur en pourcentages selon les différentes fonctions pour différents ensembles de données réels. (ADQ : Analyse discriminante quadratique, ADL : Analyse discriminante linéraire, NBA : Classifieur bayésien naïf, LOG : Régression logistique, LOGQ : Régression logistique quadratique et Tree : arbre de décision)

2.2.1 Données Pima

Le jeu de données Pima contient 532 individus décrits par 7 variables. Nous observons, pour 100 répétitions de l'expérience, lors de l'analyse discriminante et de la régression logistique que nous obtenons des taux d'erreurs plus élevés que pour les données théoriques (données Synth générées). En effet, la plupart d'entre eux sont situés entre 21% et 23% d'erreur. Pour le calcul du taux d'erreur sur les arbres de décisions, celui-ci est de 14,93%, soit légèrement plus faible. Ces taux d'erreurs importants peuvent être expliqués par la Figure 6 qui montre bien que les deux classes ne sont facilement distinguable pour aucun couple de paramètres.

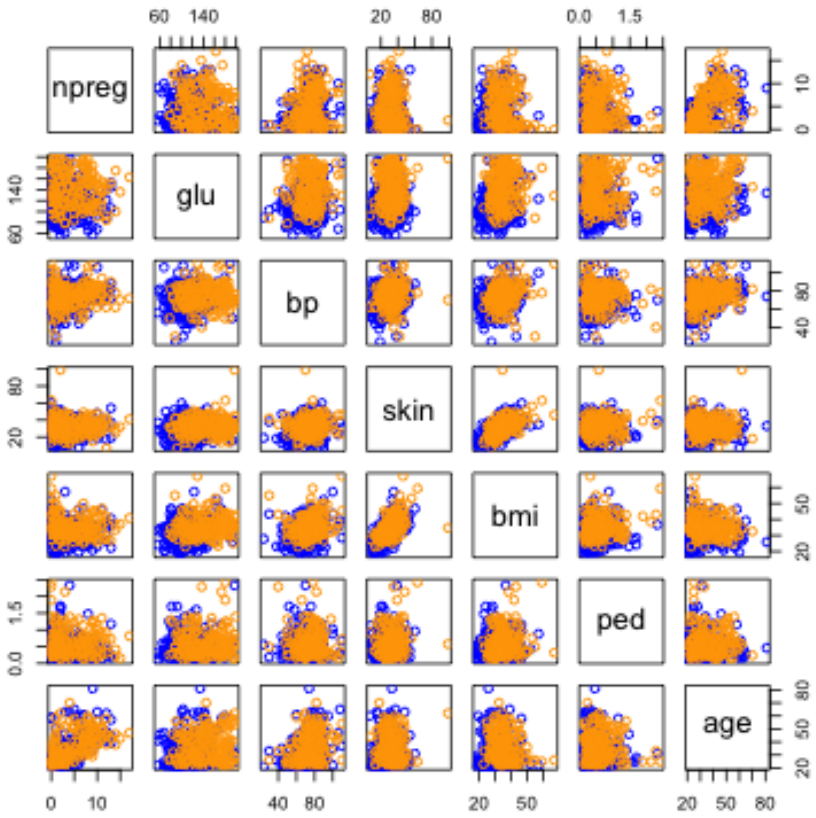


FIGURE 6 – Matrice des graphs des représentations des variables de Pima

2.2.2 Données Breast cancer "Wisconsin"

Le jeu de données BreastCancer contient 683 individus décrits par 9 variables. Nous observons, pour 100 répétitions de l'expérience, lors de l'analyse discriminante, la régression logistique et les arbres de décisions que nous obtenons des taux d'erreurs similaires aux données théoriques (données Synth générées). En effet, la plupart d'entre eux sont situés entre 3% et 5% d'erreur. En ce qui concerne les regressions logistiques on constate que pour ce jeu de données la prise en compte de l'ajout de l'ordonnée à l'origine est primordiale puisque sans ajout de l'ordonnée à l'origine on remarque un taux d'erreur autour de 15% tandis que avec nous sommes à 3%. On remarque en effet sur la Figure 7 que les variables ne sont pas dispersées autour de l'origine puisqu'elles prennent leur valeur entre 1 et 10 (11 pour la 7^{eme} variable) et que l'ajout d'un point d'ordonnée à l'origine corrige ce problème.

```
       V2              V3              V4              V5              V6
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00   Min.   : 1.000
 1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 1.00   1st Qu.: 2.000
 Median : 4.000   Median : 1.000   Median : 1.000   Median : 1.00   Median : 2.000
 Mean   : 4.442   Mean   : 3.151   Mean   : 3.215   Mean   : 2.83   Mean   : 3.234
 3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 4.00   3rd Qu.: 4.000
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00   Max.   :10.000
       V7              V8              V9             V10
 Min.   : 2.000   Min.   : 1.000   Min.   : 1.00   Min.   : 1.000
 1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 1.00   1st Qu.: 1.000
 Median : 2.000   Median : 3.000   Median : 1.00   Median : 1.000
 Mean   : 3.217   Mean   : 3.445   Mean   : 2.87   Mean   : 1.603
 3rd Qu.: 3.000   3rd Qu.: 5.000   3rd Qu.: 4.00   3rd Qu.: 1.000
 Max.   :11.000   Max.   :10.000   Max.   :10.00   Max.   :10.000
```

FIGURE 7 – Résumés des variables caractérisant les données Breast Cancer

2.2.3 Challenge : données Spam

Les données fournies dans le fichier spam.csv comportent trop de variables par rapport au nombre d'individus fournis (4601 individus pour 57 variables) ; nous allons donc appliquer l'algorithme random forest à ces données. Nous obtenons ainsi la matrice de confusion (cf. table 3) qui nous permet de visualiser les taux d'erreurs de prédictions de classes (par l'algorithme).

z	1 (prédit)	2 (prédit)	taux d'erreur (%)
1	1680	133	7.335907
2	78	2710	2.797704

TABLE 3 – Matrice de confusion des données Spam (algorithme random forest)

Nous obtenons ensuite dans la table 4 et la figure 8 le classement des variables selon si elles sont plus ou moins explicatives. Dans le modèle représenté par les données spams nous observons donc que les variables V52 et V53 sont celles qui nous permettent le mieux de distinguer un spam d'un courriel désiré. Nous pouvons donc les analyser graphiquement, nous obtenons les graphiques de la figure 9, nous y observons que la quasi-totalité des individus se répartissent en proportions 1/3 - 2/3 et de façon uniforme selon ces deux variables.

Nous analysons ensuite le taux d'erreurs selon le nombre d'arbres (cf. figure 10), nous observons qu'a partir de 1500 arbres l'erreur se stabilise.

Pour conclure, nous pouvons observer que l'erreur aue nous avons obtenu sur les données spams grâce à l'algorithme random forest est de 2.8% pour la reconnaissance des individus de la classe 2 alors qu'elle est plus élevée (7.3%) pour les individus de la classe 1.

Variable	Mean Decrease Gini
V52	252.8463157
V53	204.8876276
V7	176.7147973
V16	147.8342008
V55	142.6446466
V21	125.8198107
V56	117.8785585
V25	102.8969686
V57	87.8351430
V24	78.7664839

TABLE 4 – Classement des 10 premières variables explicatives des données Spam

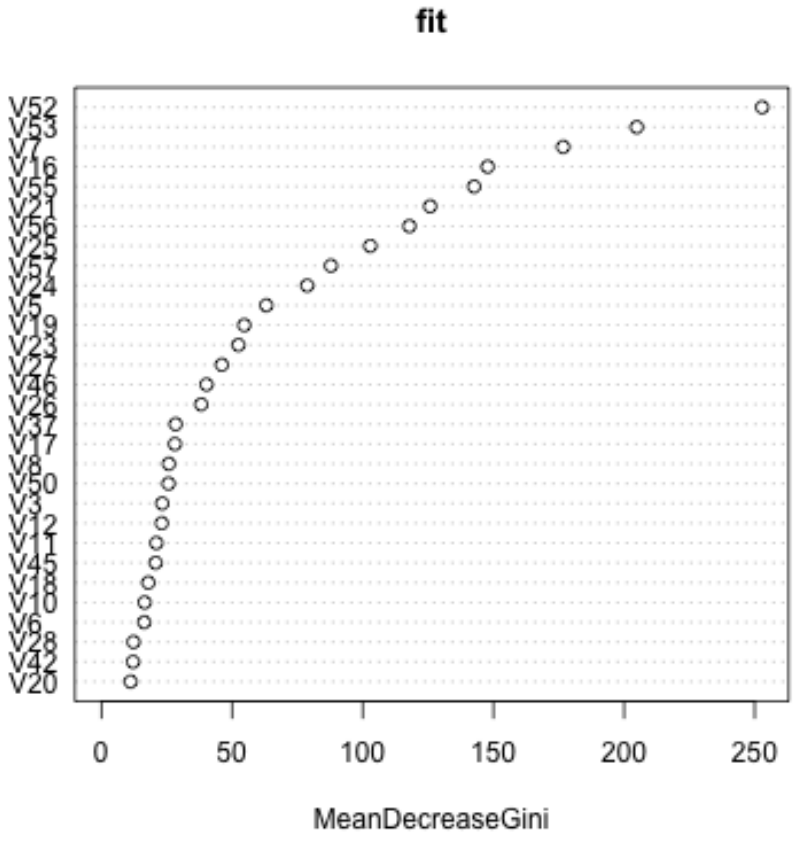


FIGURE 8 – Classement des variables de Spam selon leur degré d'explicativité

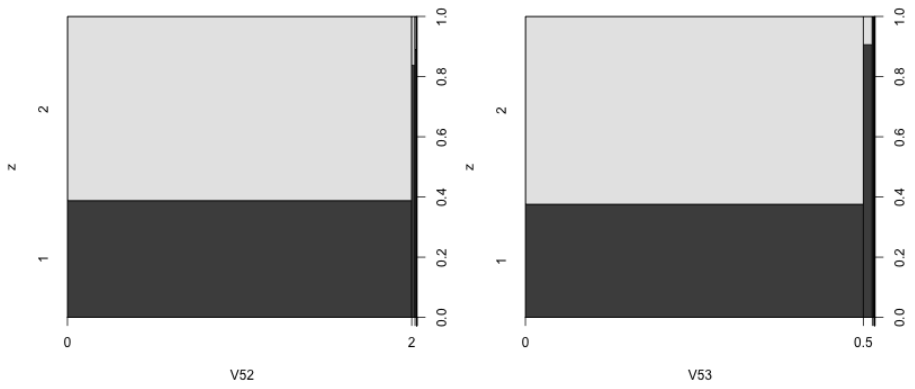


FIGURE 9 – Graphiques représentant les variables V52 et V53 en fonction de z pour les données spam

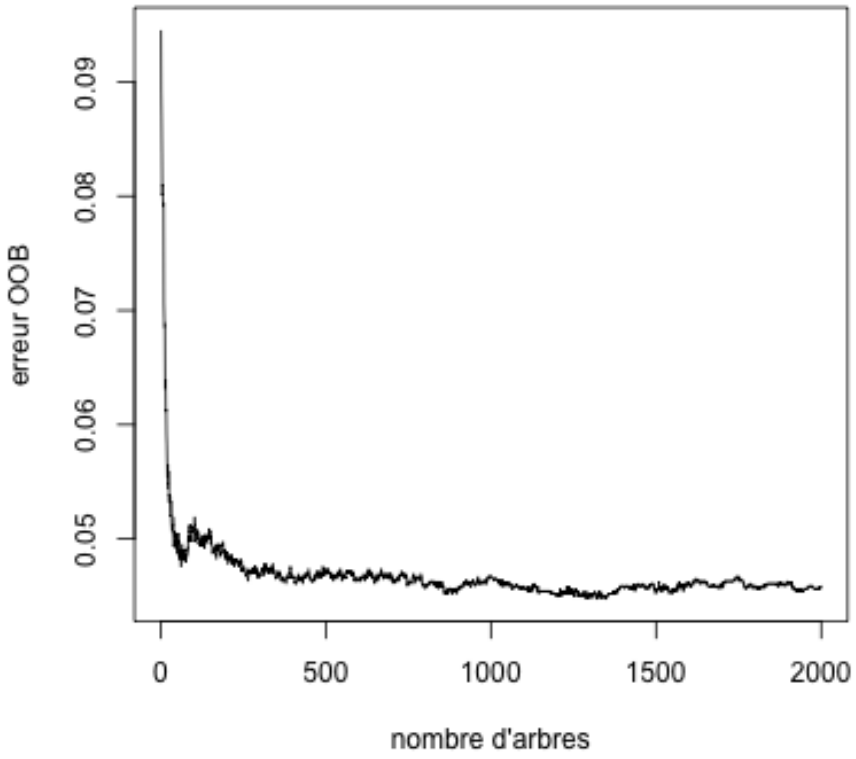


FIGURE 10 – Erreur sur random forest selon le nombre d'arbre