UNIVERSITÉ NATIONALE DU VIETNAM
INSTITUT FRANCOPHONE INTERNATIONAL

FOUILLE DE DONNÉES

RAPPORT FINAL DES TRAVAUX

Jeu de données : **"Infarctus"**

Juin 2019

Étudiants du Groupe 9 :

Mike Arley MORIN

Afi Elolo Gisèle DEKPE

Professeur : **Nguyen Thi Minh Huyen**

P23 RSC, Année Académique : 2019 - 2020

Table des matières

Table des figures

Introduction

La fouille de données est une approche analytique de données, adaptée et utilisée dans un large nombre de domaine d'activités. C'est une discipline qui vise à extraire les informations pertinentes d'un grand ensemble de données. Tout l'enjeu est de réussir à préparer, manipuler et analyser les données dans l'optique de les transformer en connaissance actionnable et en outil d'aide à la décision pour bon nombre de domaines de la vie active. Le principe de la Fouille de données se base sur trois (03) importants aspects : ***SAVOIR*** , car dans un premier de temps, il nous permet de comprendre les phénomènes ; ensuite ***PRÉVOIR*** pour enfin ***DÉCIDER***. A travers ses méthodes, nous pouvons anticiper et faire des prédictions pour des prises de décision sur notre environnement ; et ce peu importe le type d'environnement. Bien évidemment, le Data Mining s'appuie sur plusieurs méthodes. Dans le but de nous imprégner dans le sujet de la fouille de données, il nous a été demandé de choisir un jeu de données sur lequel nous avons travaillé en binôme. Pour ce faire, nous avons choisi le jeu de données **"Infarctus"**. Ce document fait office du rapport des traitements effectués sur les données. Dans les lignes suivantes, nous allons faire la description des données puis une analyse exploratoire des variables et des paires de variables.

1 Méthodologie utilisée et Logiciel utilisé

1.1 La méthodologie CRISP

Pour réaliser un projet de Fouille de données, il est important de choisir une méthode à utiliser. Il en existe plusieurs mais nous avons choisie une qui se dénomme **CRISP** (*Cross Industry Standard Process*). CRISP est né du bésoin de mettre en place une méthode de découverte de connaissance simple et efficace, applicable à tout secteur d'activité. Elle permet de rendre les projets de Fouille de données à grande échelles plus rapides, moins coûteux, plus fiables et surtout améliorer leur gestion. L'avantage de son adoption est donc évidente. Cette méthodologie ne vise pas que les grands projets car même les petits projets de Fouille de données peuvent tirer profit de son utilisation. Elle est de ce fait nommée de **modèle vertueux**. Le schéma ci-dessous montre le cycle de vie d'un projet avec CRISP, qui identifie clairement les principales phases au travers des tâches et des relations entre ces tâches.

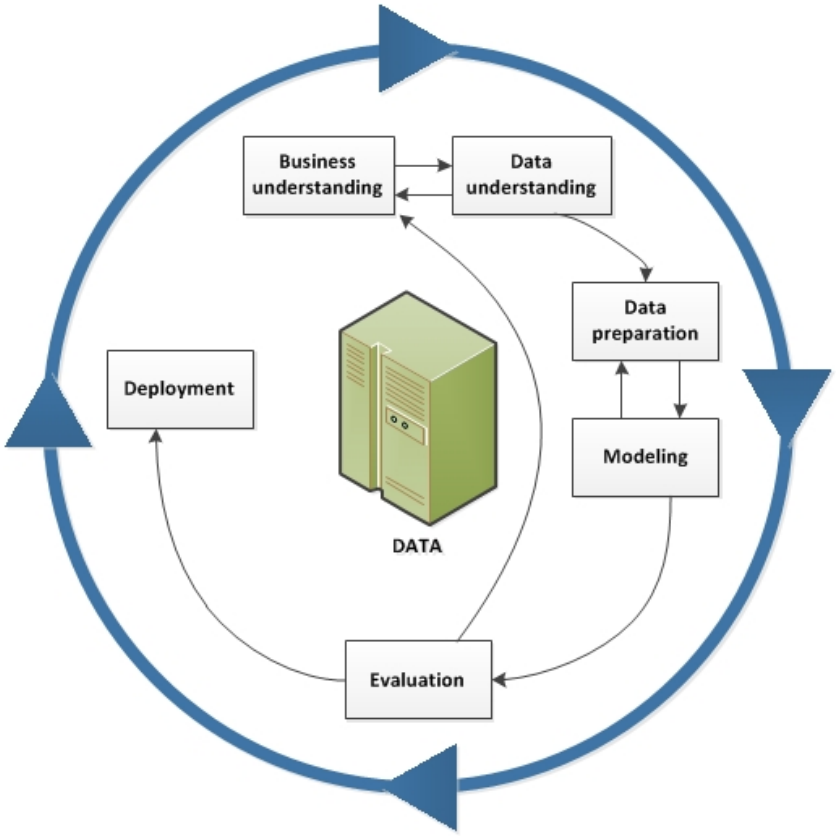


FIGURE 1 – Le processus CRISP

- **La compréhension du problème métier** : il s'agit de définir le problème d'analyse sur la base des objectifs métier qui en sont à l'origine. Cette étape consiste à bien comprendre les éléments métiers et problématiques que notre étude vise à résoudre ou à améliorer.

- **La compréhension des données** : il faut déterminer les données essentielles à analyser ainsi que leur qualité des données disponibles. Cette étape nous permet de faire le lien entre les données et leur signification d'un point de vue métier.

- **La préparation des données ou construction du DataHub** : ici, on construit un ensemble à partir de données brutes en faisant un classement des données en fonction de critères choisis, un nettoyage également et surtout leur recodage pour les rendre compatibles avec les algorithmes qui seront utilisés puis s'il le faut puis d'autres opérations encore.

- **La modélisation** : c'est la phase la plus importante dans le paramétrage et le test de différentes techniques, sachant que l'objectif est d'optimiser un modèle et ainsi les connaissances obtenues. La modélisation comprend également le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle.

- **L'évaluation** : elle vise à vérifier le modèle ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus. Elle contribue aussi à la décision de déploiement du modèle, ou si besoin est, à son amélioration. A ce stade, on teste notamment la robustesse et la précision des modèles obtenus.

- **Le déploiement** : c'est l'étape finale du processus de découverte de connaissance. Son objectif est de mettre la connaissance obtenue par la modélisation, dans une forme adaptée et l'intégrer au processus de prise de décision . Le déploiement peut aller, selon les objectifs, de la simple génération d'un rapport décrivant les connaissance obtenues jusqu'à la mise en place d'une application, permettant l'utilisation du modèle obtenu, pour la prédiction de valeurs inconnues d'un élément d'intérêt (exemple : le nombre d'étudiants en informatique à investir dans un ordinateur portable de marque MAC).

Cette méthode est agile et itérative, c'est-à-dire que chaque itération apporte de la connaissance métier supplémentaire qui permet de mieux aborder l'itération suivante. Chacune des six étapes de la méthodologie CRISP peut être assimilée à une user story, en mettant la production de valeur au centre de chaque itération. Elles sont toutes aussi

importantes l'une que l'autre pour la réussite d'un projet de Data Mining et la méthode CRISP répond bien à ce besoin ; raison pour laquelle nous l'avons choisie.

En plus, elle n'impose pas d'outil ou langage de travail ; ce qui nous conduit librement à choisir le logiciel **R** pour notre projet.

1.2 Logiciel utilisé : R

R est un langage de programmation et en même temps un logiciel libre, destiné aux statistiques et à la science des données soutenu par la **R Foundation for Statistical Computing**. Il est un logiciel libre distribué selon les termes de la licence GNU GPL et disponible sous GNU/Linux, disposant d'une multitude de bibliothèque largement utilisé par les statisticiens, les data miners, data scientists pour le développement de logiciels statistiques et l'analyse des données. Il fonctionne sous la forme d'un interpréteur de commandes. Il dispose d'une bibliothèque très large de fonctions statistiques, d'autant plus large qu'il est possible d'en intégrer de nouvelles par le système des "packages", des modules externes compilés (sous forme de DLL sous Windows) que l'on peut télécharger gratuitement sur internet. R propose également une palette étendue de fonctionnalités graphiques.

La méthodologie et l'outil de travail ainsi décrits, nous allons passer à la réalisation des étapes précitées dans la description de la méthodologie ; en commençant par l'introduction au problème que nous allons étudier.

2 Introduction au problème

Cette rubrique est dédiée à la compréhension du problème métier.

2.1 Contexte de l'étude

Dans notre contexte, il s'agit d'une étude cas-témoins sur l'infarctus du myocarde dont le but était d'évaluer l'existence d'un risque plus élevé de survenue d'un infarctus du myocarde chez les femmes qui utilisent ou ont utilisé des contraceptifs oraux. Cette étude a été menée par une équipe de l'Institut de santé publique d'épidémiologie et de développement (ISPED) de Bordeaux. Pour mener à bien notre étude, nous suivrons un certains nombres d'étapes qui nous mèneront jusqu'à la production d'un modèle de prédiction validé pour notre jeu de données.

2.2 Problématique

Nous avons à notre disposition des données relatives à des sujets classifiés comme « **témoins** » (n'ayant pas eu un infarctus du myocarde) ou « **cas** » (ayant eu un infarctus du myocarde). Ce qui nous amène à dire que nous sommes dans un problème de classification dans le domaine de l'apprentissage supervisé.

2.3 Objectif à atteindre

Il s'agit par la suite de prédire le risque de survenue d'un infarctus du myocarde de futurs patients.

2.4 Proposition de solution

Pour résoudre le problème qui se pose dans notre problématique, nous proposons de comparer plusieurs modèles afin de choisir le modèle avec le taux d'erreur le plus faible. La principale est la méthode de **Random Forests** et en secondaire la méthode de la **Classification Naïve bayésienne**.

La prochaine étape sera dédiée à comprendre les données qui seront utilisées pour la modélisation.

3 La compréhension des données

La phase de compréhension des données de CRISP concerne l'étude des données disponibles pour l'exploration de données. Cette étape revêt une importance vitale, car elle permet d'éviter les problèmes inattendus au cours de la phase suivante, la préparation des données, phase généralement la plus longue d'un projet. La compréhension des données implique l'accès aux données et leur exploration à l'aide de tables et de graphiques. Les lignes suivantes décrivent cette étape du processus.

3.1 Choix du jeu de données

Un jeu de données est un ensemble de valeurs (ou données) où chaque valeur est associée à une variable (ou attribut) et a une observation. Une variable décrit l'ensemble des valeurs décrivant le même attribut et une observation contient l'ensemble des valeurs décrivant les attributs d'une unité (ou individu statistique). Pour ce travail, nous avons choisi « **"Infarctus"** ». Il contient non seulement des valeurs d'entrées mais aussi il contient une valeur de sortie. L'étude a été menée auprès de 148 femmes ayant eu un infarctus du myocarde (cas) et 300 femmes n'en ayant pas eu (témoins). Le facteur d'exposition principal est la prise de contraceptifs oraux, les autres facteurs recueillis sont : l'âge, le poids, la taille, la consommation de tabac, l'hypertension artérielle, les antécédents familiaux de maladies cardio-vasculaires. Notre jeu de donnée est disponible au lien suivant : **https ://github.com/GiseleIgore/DataMiningProjectIFI/blob/master/infarctus.csv**.

3.2 Description du jeu de données et résumé des données

L'ensemble de données « Infarctus » comporte 9 variables dont 5 variables qualitatives et 4 variables quantitatives ; avec pour variable de sortie **"Infarct"** pour un total de 448 observations :

N°	Type	Variables	Description	Unité ou Codage
1	Qualitative	Infarct	Infarctus du Myocarde	0: témoins 1:Cas
2		Co	Prise de contraceptifs oraux	0: Jamais 1:Oui
3		Tabac	Consommation de tabac	0: Non 1: Fumeuse 2: Ancienne fumeuse
4		Atcd	Antécédents familiaux de maladie cardio-vasculaire	0: Nom 1: Oui
5		Hta	Hypertension artérielle	0: Non 1: Oui
6	Quantitative	Age	Age	Années
7		Poids	Poids	Kg
8		Taille	Taille	Cm
9		Imc	Indice de masse corporelle	Kg/m2

FIGURE 2 – Table des variables

Les variables étant ainsi présentées, nous estimons qu'elles sont toutes importantes pour notre analyse. Nous allons passer à l'étude des variables avec le logiciel de Fouilles de Données **R**.

3.3 Analyse exploratoire des variables et représentation graphique : Cas d'une variable prise de façon unique

3.3.1 Cas des variables quantitatives

Nous présentons dans le tableau des indicateurs statistiques(Tendance centrale et Dispersion) ci-dessous, l'étude des variables quantitatives de notre jeu de données et à travers les graphiques :

(i) ***Le tableau des indicateurs statistiques des variables quantitatives***

Variables		Age	Poids	Taille	Imc
Tendance centrale	Effectif	448	448	448	448
	Moyenne	45.61	66.44	165.14	24.20
	Médiane	44	64	166	23.03
	Mode	42	45	170	16.35
Dispersion	Min	15	33	138	11.36
	Max	100	128	184	47.78
	Étendue	15 - 100	33 - 128	138 - 184	11.36 - 47.78
	Variance	261.9115	326.31	65.79	51.04
	Écart-type	16.18368	18.06	8.11	7.14

FIGURE 3 – Tableau des indicateurs statistiques des variables quantitatives

(ii) ***Les histogrammes des variables quantitatives***

— Histogramme : *Age*

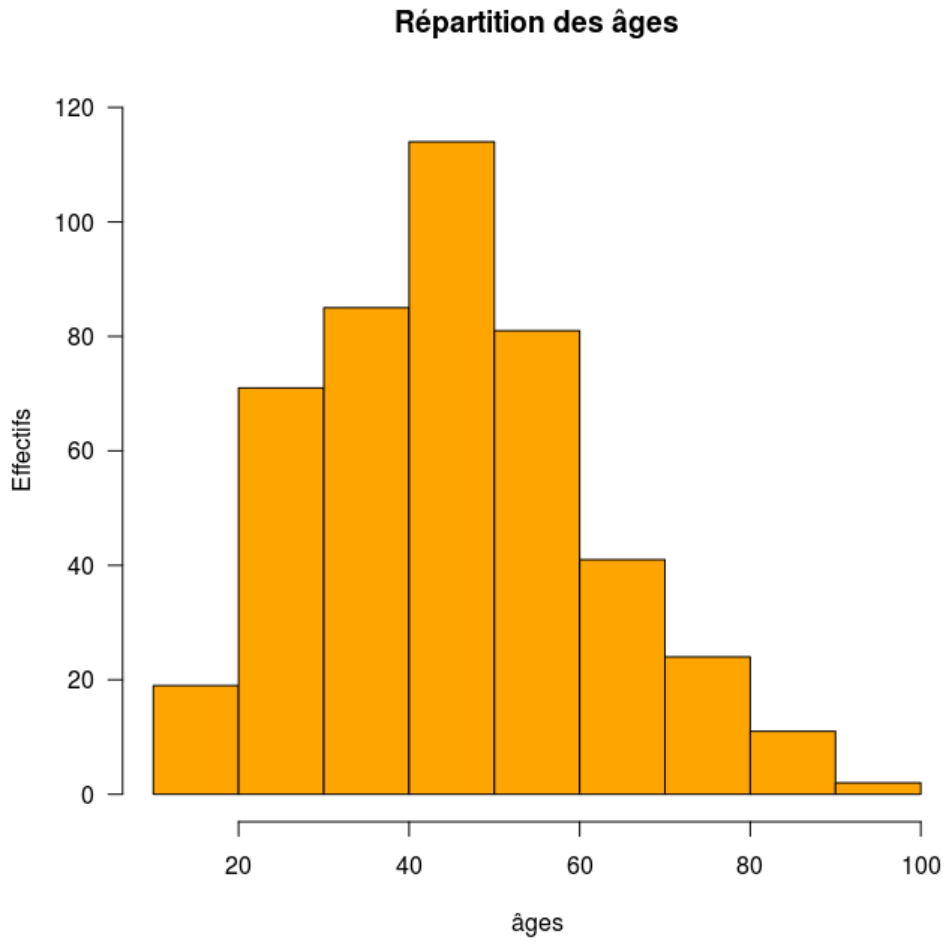


FIGURE 4 – Histogramme de la variable Age

— Histogramme : *Poids*

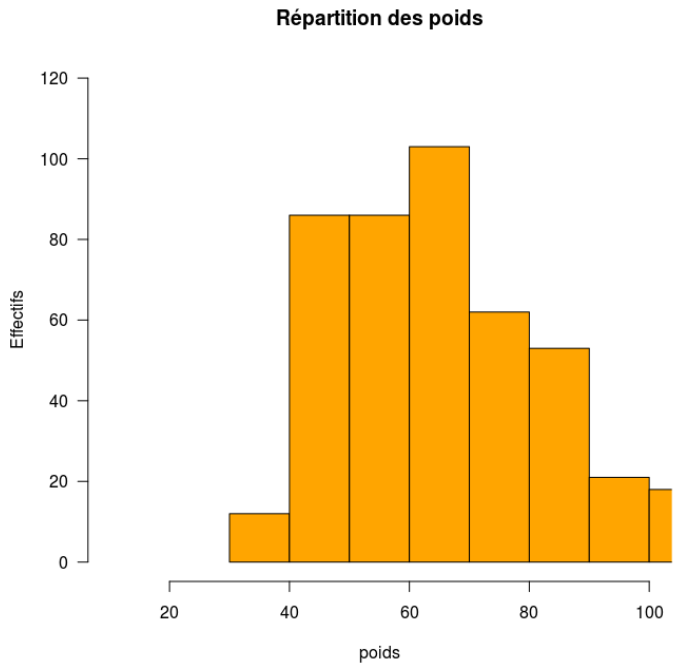


FIGURE 5 – Histogramme de la variable Poids

— Histogramme : *Taille*

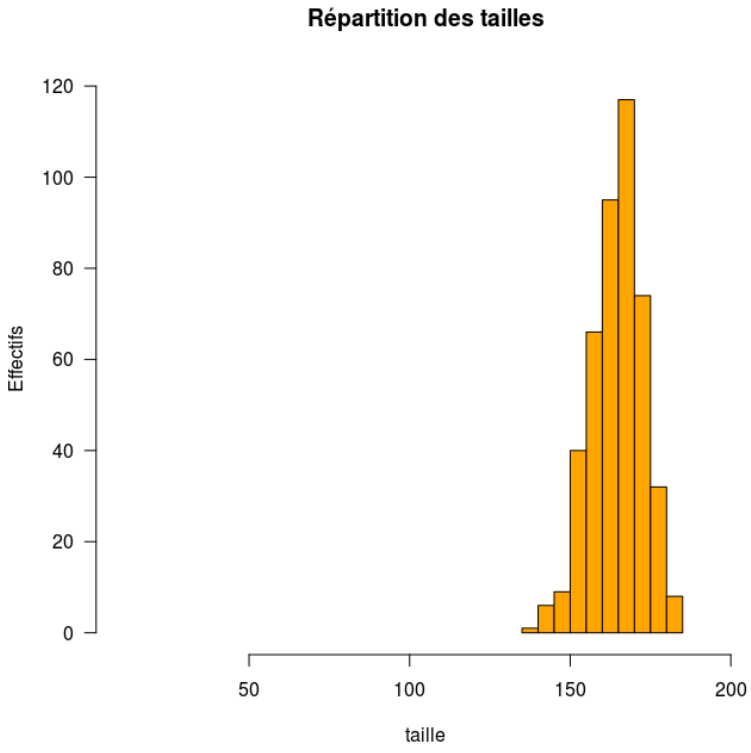


FIGURE 6 – Histogramme de la variable Taille

— Histogramme : *Imc*

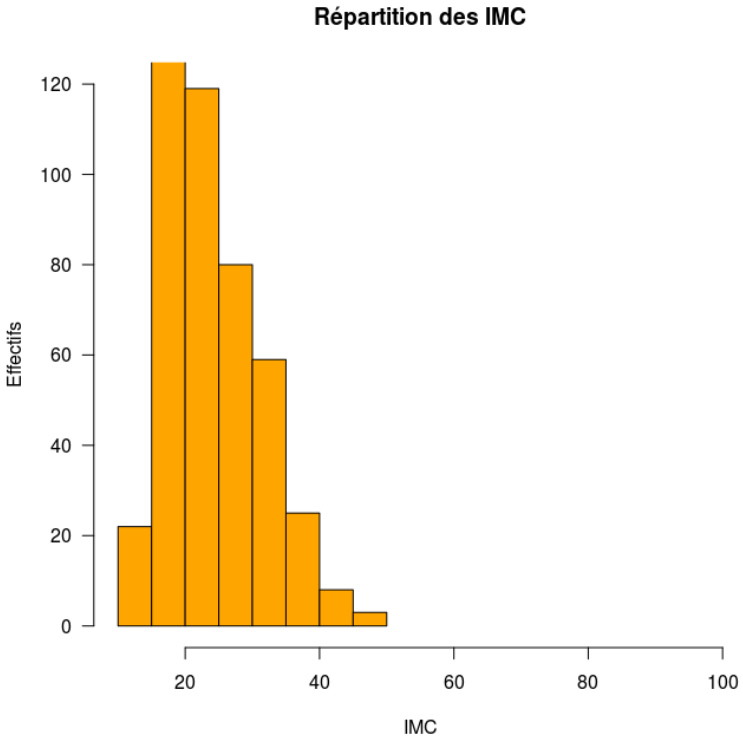


FIGURE 7 – Histogramme de la variable Imc

(iii) Interprétation :

— Pour l'*Age* : les traitements nous montre les informations suivantes :

• Indicateurs statistiques de Tendance centrale :

- la moyenne = 45,61 : ainsi la moyenne d'âge de la population étudiée est estimée à 45 ans ; - la médiane = 44 : la moitié de la population étudiée a plus de 44 ans tandis que l'autre moitié a moins de 44 ans ; - le mode est 42 : la majorité a 42 ans ;

• Indicateurs statistiques de Dispersion : - l'écartype = 16,18 ; - l'étendue = 85 : la différence d'âge maximale est de 85 ans ; - La dispersion interquartile (variance) est de 261.91.

— Pour le *Poids* :les traitements nous montre les informations suivantes :

• Indicateurs statistiques de Tendance centrale :

- la moyenne = 66.44 KG ; - la médiane = 64 : la moitié de la population

étudiée pèse plus de 44 ans tandis que l'autre moitié a moins de 44 ans ; - le mode est 45 : la majorité a 45KG comme poids ;

• Indicateurs statistiques de Dispersion : - l'écartype = 18.06 ; - l'étendue = 95 : la différence de poids maximale est de 95 Kg ; - La dispersion interquartile (variance) est de 326.31.

— Pour la *Taille* : les traitements nous montre les informations suivantes :

• Indicateurs statistiques de Tendance centrale :

- la moyenne = 165.14 : ainsi la taille moyenne de la population étudiée est estimée à 1m 65 cm ; - la médiane = 166 : la moitié de la population étudiée mesure plus de 1m 66 tandis que l'autre moitié moins de 1m 66 ; - le mode est 170 : la majorité 1m 70 ;

• Indicateurs statistiques de Dispersion : - l'écartype = 8.11 ; - l'étendue = 46 : la différence de taille maximale est de 0.46 m ; - La dispersion interquartile (variance) est de 65.79.

— Pour l'*IMC* : les traitements nous montre les informations suivantes :

• Indicateurs statistiques de Tendance centrale :

- la moyenne = 24.20 : ainsi la moyenne d'indice de masse corporelle de la population étudiée est estimée à 24.2 Kg par mètre-carré ; - la médiane = 23.03 : la moitié de la population étudiée a plus de 23.03 Kg/m tandis que l'autre moitié a moins de 23.03 Kg par mètre-carré ; - le mode est 16.35 Kg par mètre-carré : la majorité a 16.35 Kg par mètre-carré ;

• Indicateurs statistiques de Dispersion : - l'écartype = 7.14 Kg par mètre-carré ; - l'étendue = 36.42 Kg par mètre-carré : la différence d'indice de masse corporelle maximale est de 36.42 Kg par mètre-carré ; - La dispersion interquartile (variance) est de 51.04 Kg par mètre-carré.

3.3.2 Cas des variables qualitatives

L'étude des variables qualitatives est également résumée dans le tableau ci-dessous et à travers les graphiques suivants :

(i) ***Le tableau des indicateurs statistiques des variables qualitatives***

Variables	Infarct		Co		Tabac			ATCD		HTA	
Valeurs	0	1	0	1	0	1	2	0	1	0	1
Fréquence	300	148	249	199	215	135	98	395	53	290	158
Fréquence cumulée	448										
Pourcentage	66.96	33.04	55.58	44.41	47.99	30.13	21.87	88.17	11.83	64.73	35.26
Pourcentage cumulée	100										

FIGURE 8 – Tableau des indicateurs statistiques des variables qualitatives

(ii) ***Les diagrammes en secteur des variables qualitatives***

— Diagramme en bâtons : *Infarct*

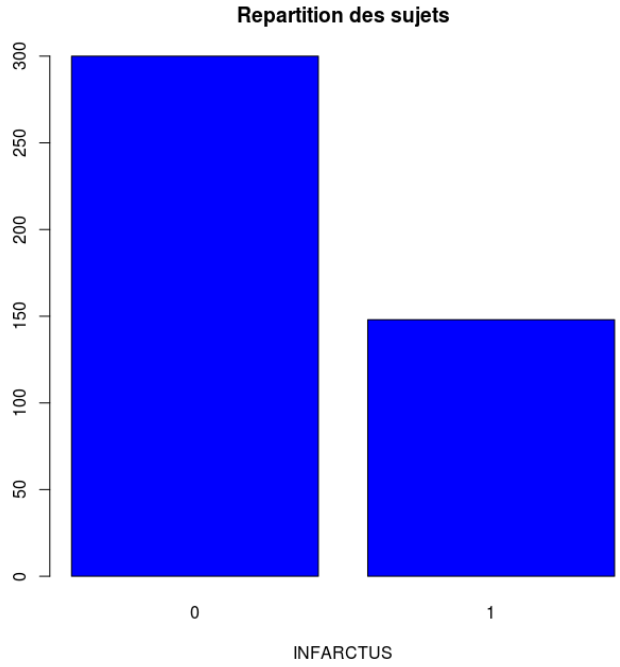


FIGURE 9 – Diagramme en bâtons de la variable Infarct

— Diagramme en bâtons : *Co*

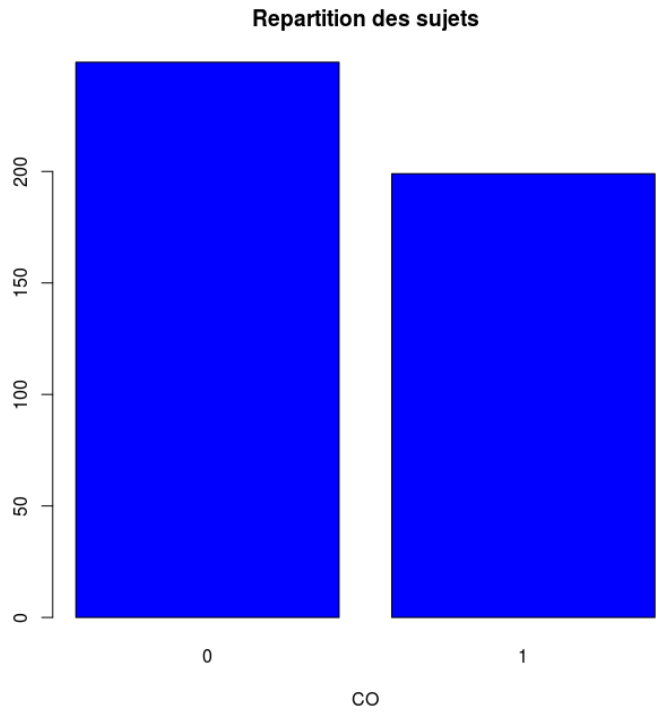


FIGURE 10 – Diagramme en bâtons de la variable CO

— Diagramme en bâtons : *Tabac*

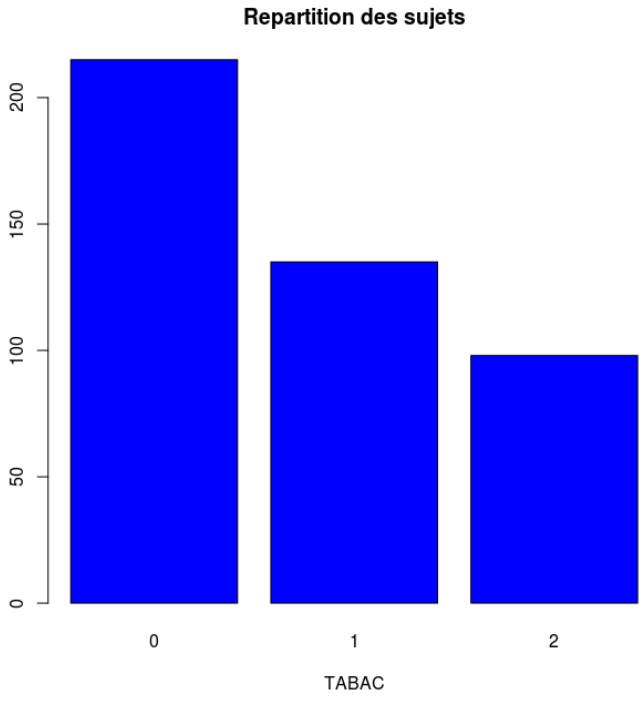


FIGURE 11 – Diagramme en bâtons de la variable Tabac

— Diagramme en bâtons : *ATCD*

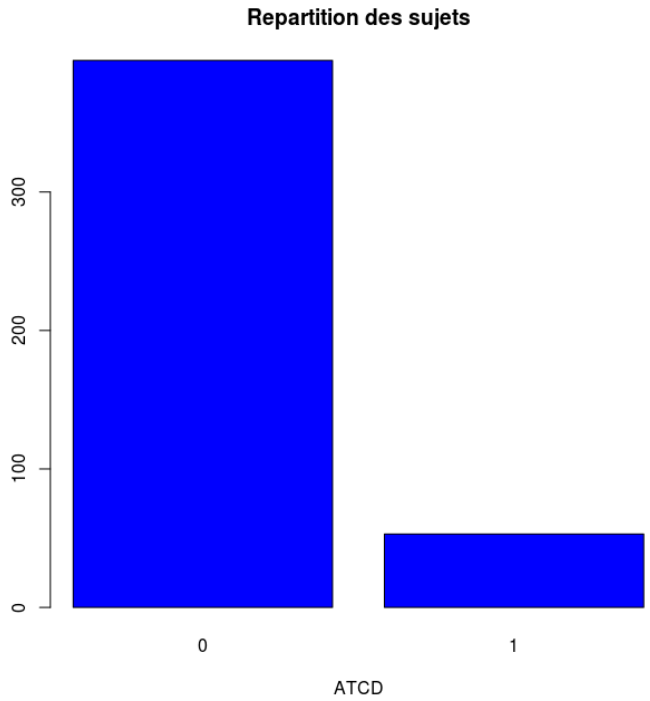


FIGURE 12 – Diagramme en bâtons de la variable ATCD

— Diagramme en bâtons : *HTA*

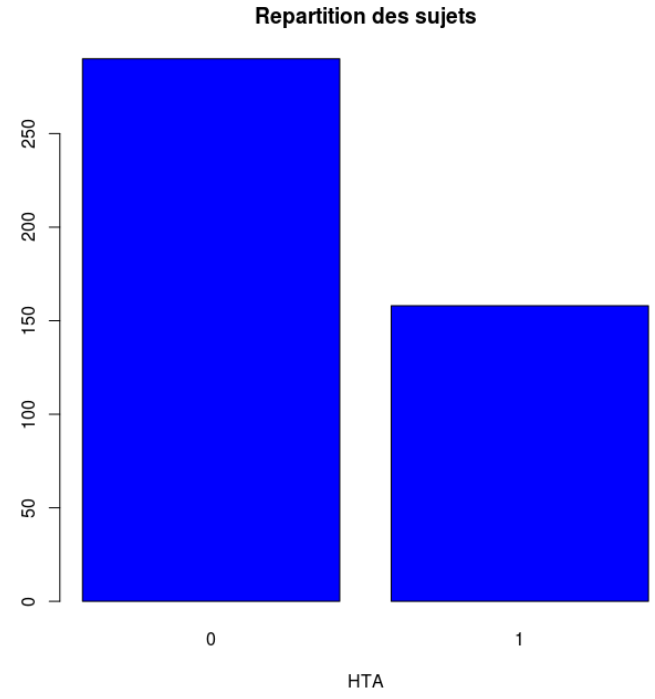


FIGURE 13 – Diagramme en bâtons de la variable HTA

(iii) Interprétation :

— Pour la variable *Infarct* : d'après le résultat des calculs et de la représentation en secteur ci-dessus, le mode de la variable « Infarct » est « 0 » puisque parmi les valeurs correspondantes à l'attribut Infarct, c'est « 0 » qui a la plus forte proportion d'invidus (**66.96%**). La plupart de la population étudiée est donc **témoins d'infarctus du myocarde**.

— Pour la variable *Co* : d'après le résultat des calculs et de la représentation en secteur ci-dessus, le mode de la variable « Co » est « 0 » puisque parmi les valeurs correspondantes à l'attribut Co, c'est « 0 » qui a la plus forte proportion d'invidus (**55.58%**). La plupart de la population étudiée **ne prend donc pas de contraceptifs oraux**.

— Pour la variable *Tabac* : d'après le résultat des calculs et de la représentation en secteur ci-dessus, le mode de la variable « Tabac » est « 0 » puisque parmi les valeurs correspondantes à l'attribut Tabac, c'est « 0 » qui a la plus forte proportion d'invidus (**47.99%**). La plupart de la population étudiée **ne fume pas**.

— Pour la variable *ATCD* : d'après le résultat des calculs et de la représentation en secteur ci-dessus, le mode de la variable « ATCD » est « 0 » puisque parmi les valeurs correspondantes à l'attribut ATCD, c'est « 0 » qui a la plus forte proportion d'invidus (**88.17%**). La plupart de la population étudiée **n'a pas d'antécédent familial de maladie cardio-vasculaire**.

— Pour la variable *HTA* : d'après le résultat des calculs et de la représentation en secteur ci-dessus, le mode de la variable « HTA » est « 0 » puisque parmi les valeurs correspondantes à l'attribut HTA, c'est « 0 » qui a la plus forte proportion d'invidus (**64.73%**). La plupart de la population étudiée **ne souffre pas d'hypertension artérielle**.

3.4 Analyse exploratoire des paires de variables

3.4.1 La corrélation entre les variables

Le tableau ci-dessous présente le résultat de la corrélation entre la variable de sortie ainsi que les autres variables. Les valeurs les plus dominantes concernant la variable de sortie sont encadrées en rouge.

```
> cor(data)
            INFARCT          CO        TABAC         AGE        POIDS
INFARCT  1.00000000  0.432279446  0.350560059  0.05680119  0.201888766
CO       0.43227945  1.000000000  0.214831667 -0.24557914 -0.009033425
TABAC    0.35056006  0.214831667  1.000000000 -0.30562133  0.122223183
AGE      0.05680119 -0.245579138 -0.305621327  1.00000000  0.094025986
POIDS    0.20188877 -0.009033425  0.122223183  0.09402599  1.000000000
TAILLE  -0.03580759 -0.025926466  0.031052111 -0.07193286 -0.014276296
IMC      0.18840600  0.002250073  0.130097816  0.11279752  0.886621517
ATCD     0.06599332  0.006364894 -0.001379484 -0.02240932  0.150232545
HTA      0.10730658 -0.076937613 -0.145540971  0.34920382  0.184369979
             TAILLE          IMC          ATCD         HTA
INFARCT -0.03580759  0.188406004  0.065993325  0.10730658
CO      -0.02592647  0.002250073  0.006364894 -0.07693761
TABAC    0.03105211  0.130097816 -0.001379484 -0.14554097
AGE     -0.07193286  0.112797516 -0.022409315  0.34920382
POIDS   -0.01427630  0.886621517  0.150232545  0.18436998
TAILLE   1.00000000 -0.331768877  0.041006714  0.02731272
IMC     -0.33176888  1.000000000  0.108626467  0.19211593
ATCD     0.04100671  0.108626467  1.000000000  0.12017451
HTA      0.02731272  0.192115930  0.120174506  1.00000000
```

FIGURE 14 – La corrélation entre les variables

Les informations représentées montrent que la variable d'intérêt **INFARCT** a une corrélation positive forte avec la variable **CO**, comparativement à **TABAC** et **POIDS**.

3.4.2 La relation entre la variable d'intérêt et quelques variables

(i) *Relation entre la variable de sortie **INFARCT** et la variable **CO***

Co	Infarct 0	1
0	212	37
1	88	111

FIGURE 15 – Tableau de contingence entre INFARCT et CO

Interprétation : La figure nous montre que sur les 148 cas d'infarctus 111 ont pris des contraceptifs oraux. Ci-dessous la représentation graphique.

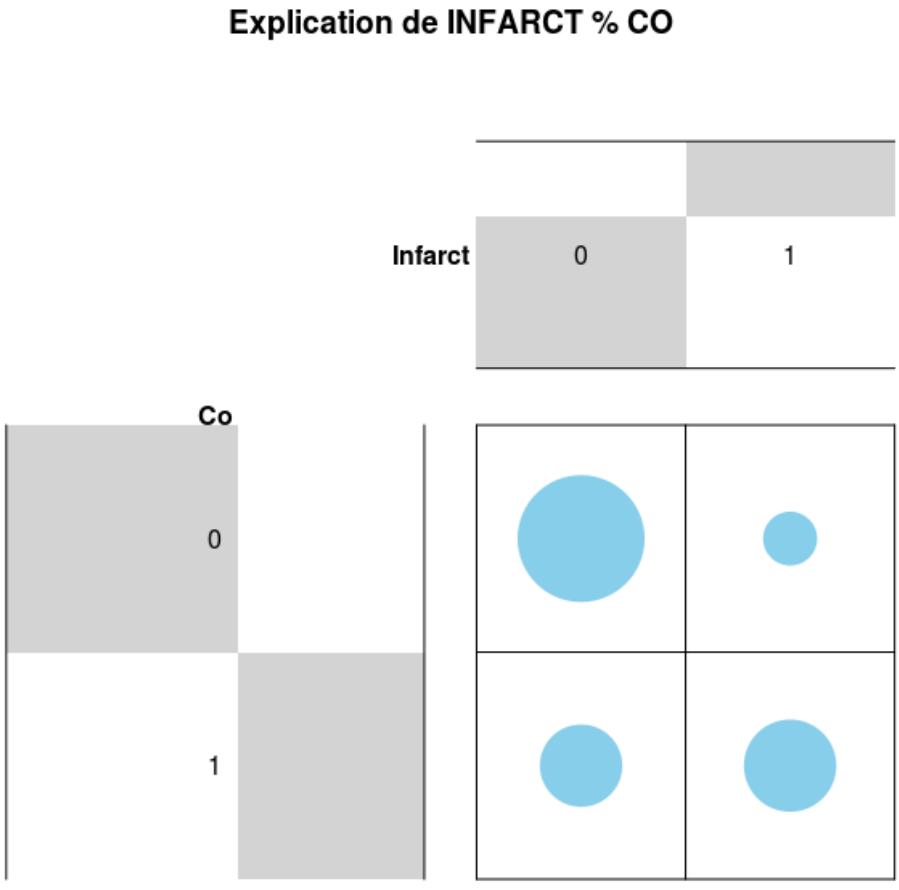


FIGURE 16 – Explication de la variable INFARCT par rapport à CO

(ii) *Relation entre la variable de sortie* ***INFARCT*** *et la variable* ***TABAC***

Tabac	Infarct 0	1
0	181	34
1	75	60
2	44	54

FIGURE 17 – Tableau de contingence entre INFARCT et TABAC

Interprétation : La figure nous montre que sur les 148 cas d'infarctus , on a 60 cas d'infarctus qui sont fumeuses de tabac et 54 qui sont anciennes fumeuses qui fait un total de 114. Ci-dessous la représentation graphique.

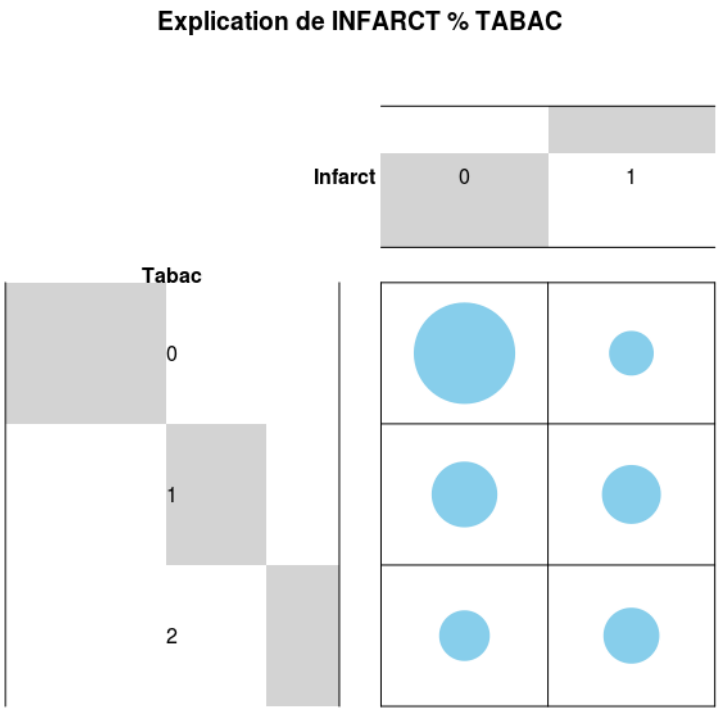


FIGURE 18 – Explication de la variable INFARCT par rapport à TABAC

3.4.3 Cas de deux (02) variables qualitatives : Test de Khi2

Nous allons chercher à trouver l'indépendance entre deux variables qualitatives. Nous allons effectuer un test khi-deux (2) à cet effet.
Les variables choisies sont l'infarctus du Myocarde "*Infarct*" et la prise de contraceptifs oraux "*Co*". L'objectif est de vérifier si les femmes qui prennent des contraceptifs sont sujettes à l'infarctus du myocarde.

Hypothèse nulle (H0) : La prise des contraceptifs oraux n'ont pas d'effet sur la contraction de l'infarctus du myocarde.
Si l'hypothèse est rejetée, alors il y a un lien entre la prise des contraceptifs et l'infarctus du myocarde. Avec **R**, nous avons fait un tableau de contingence avec les variables choisies ; puis déterminé la valeur **p-value** :**2,2exp(-16)** qui équivaut à **0.00000332075**.

	khi²	dF	p-Value
Infarct & Co	81.876	1	< 2.2e-16
Infarct & Tabac	58.338	2	2.148e-13
Infarct & ATCD	1.5408	1	0.2145
Infarct & HTA	4.6921	1	0.0303
Co & Tabac	20.808	2	3.032e-05
Co & ATCD	3.8584e-30	1	1
Co & HTA	2.3377	1	0.1263
Tabac & ATCD	0.11813	2	0.9426
Tabac & HTA	11.563	2	0.00308
ATCD & HTA	5.7146	2	0.01682

FIGURE 19 – Résultat du test de Khi-deux sur les variables qualitatives

Interprétation du résultat : khi-deux = 81,876 et p-value = 0.00000332075. p-value est très petite et inférieure au seuil significatif (0,05). Les variables présentent

donc une association statistiquement significative. L'hypothèse H0 est rejetée.

Conclusion : Le test d'association du khi2 entre les variables (**Co**, **TABAC**, **HTA**) et la variable de sortie (**INFARCT**) ont un p-Value inférieur au seuil de signification on conclut que ces variables présentent une association statistiquement significative. Nous les retenons pour la constitution du nouveau jeux de données.

3.4.4 Cas de deux (02) variables dont l'une quantitative et l'autre quantitative : Test d'ANOVA

A ce niveau, nous allons faire une analyse de variance entre les variables quantitatives et la variable qualitative "Infarct" afin de déterminer là où il y a plus de lien. Nous posons une hypothèse pour chacune des variables quantitatives.

L'hypothèse nulle **H0** veut que les moyennes de population soient toutes égales

Les résultats du test ANOVA se présentent comme suit :

* Test d'ANOVA sur l'Age :

```
Analysis of Variance Table

Response: inf
             Df  Sum Sq Mean Sq F value    Pr(>F)
factor(age)   2 2236547 1118273  94.672 < 2.2e-16 ***
Residuals   445 5256365   11812
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

* Test d'ANOVA sur le Poids :

```
Analysis of Variance Table

Response: inf
               Df  Sum Sq Mean Sq F value Pr(>F)
factor(poids)  72 1196791   16622    0.99 0.5055
Residuals     375 6296121   16790
```

* Test d'ANOVA sur la Taille :

```
Analysis of Variance Table

Response: inf
              Df  Sum Sq Mean Sq F value Pr(>F)
factor(tay)   78 1177613   15098  0.8821 0.7462
Residuals    369 6315299   17115
```

* Test d'ANOVA sur l'IMC :

```
Analysis of Variance Table

Response: inf
              Df  Sum Sq Mean Sq F value Pr(>F)
factor(imc)   41  583918   14242  0.8369 0.7534
Residuals    406 6908994   17017
```

Interprétation : De toutes les valeurs calculées, la valeur de **p** liée à l'âge (**AGE**) (**2.2e-16**) est la seule inférieure au seuil de signification(**0.05**). Nous pouvons rejeter l'hypothèse nulle et conclure à la signification statistique.

Les données ainsi explorées, nous passons à la préparation des données pour les étapes prochaines.

4 La préparation du DataHub

La préparation des données est l'une des étapes les plus importantes et qui vraisemblablement prend plus de temps. Le fait de consacrer une énergie suffisante aux phases initiales de compréhension du problème métier et de compréhension des données permet de réduire cette étape. La préparation des données comporte plusieurs tâches, et en fonction du type de projet et des objectifs à atteindre, on choisit quelle tâche exécuter. Pour notre cas d'étude, nous allons nous concentrer sur le nettoyage et la réduction des variables.

Dans un premier temps, nous allons vérifier l'existence de valeurs manquantes et/ou de doublons dans notre jeu de données ; ensuite nous passerons à la réduction des variables puis à l'échantillonnage.

4.1 Élimination de doublons

Nous avons vérifié l'existence de doublons afin de pouvoir les éliminer. Pour ce faire, nous avons utilisé la fonction **duplicated** de R et créé une variable **dup** qui indique quelles lignes sont dupliquées. Lorsqu'une ligne est dupliquée, la valeur de sa colonne **dup** affiche **TRUE**.

Opération : *data$dup <- duplicated(data)*

```
> data3
    INFARCT CO TABAC AGE POIDS TAILLE      IMC ATCD HTA   dup
1         0  0     0  47    48    173 16.03796    0   0 FALSE
2         0  0     0  17    49    162 16.35931    0   0 FALSE
3         0  0     0  35    53    163 19.94806    0   0 FALSE
4         0  0     0  82    78    157 31.64429    0   0 FALSE
5         0  0     0  50    52    172 17.57707    0   0 FALSE
6         0  0     0  31    47    184 13.88233    0   0 FALSE
7         0  0     0  60    60    169 21.00767    0   0 FALSE
8         0  0     0  30    75    174 24.77210    0   0 FALSE
9         0  0     0  44    68    164 25.28257    0   0 FALSE
10        0  0     0  38    64    167 16.35931    0   0 FALSE
11        0  0     0  33    59    173 19.71332    0   0 FALSE
12        0  0     0  62    55    146 25.80221    0   0 FALSE
```

FIGURE 20 – Exemple de sortie pour test de duplicata

C'est le résultat que nous avons eu pour toutes les 448 lignes, ce qui nous amène à dire qu'on a pas d'enregistrement dupliqué.

4.2 Les valeurs manquantes

Pour vérifier l'existence des valeurs manquantes dans le jeu de données, nous avons utilisé la fonction **is.na** (pour dire *is non available*) afin de déterminer toutes les valeurs manquantes.

Opération : *is.na(data)*

```
> is.na(data)
      INFARCT    CO TABAC   AGE POIDS TAILLE   IMC  ATCD   HTA
 [1,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [2,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [3,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [4,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [5,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [6,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [7,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [8,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
 [9,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
[10,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
[11,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
[12,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
[13,]   FALSE FALSE FALSE FALSE FALSE  FALSE FALSE FALSE FALSE
```

FIGURE 21 – Exemple de sortie pour test de valeurs manquantes

Opération : *which(is.na(data))*

```
> which(is.na(data))
integer(0)
```

FIGURE 22 – Sortie pour vérification de la position d'une éventuelle valeur manquante

L'exécution de la fonction nous retourne **FALSE** pour chaque enregistrement, ce qui prouve que nous n'avons pas de valeur manquante. Une éventuelle recherche de position de valeur manquante nous retourne bien évidemment **0** pour confirme le précédent test.

4.3 Réduction des variables

Suite à l'analyse exploratoire faite dans la section précédente, nous pensons qu'il serait bien de réduire nos variables et d'en retenir que cinq (05) ; c'est-à-dire **INFARCT**, **CO**, **HTA**, **TABAC** et **AGE**. Le dataset réduit se trouve à ce lien : **https ://github.com/GiseleIgore/DataMiningProjectIFI/blob/master/newInfarctus.csv**
Ci-dessous une sommarisation du nouveau jeu de données :

```
> str(newdataset)
'data.frame':   448 obs. of  5 variables:
 $ INFARCT: Factor w/ 2 levels "cas","temoin": 2 2 2 2 2 2 2 2 2 2 ...
 $ HTA    : Factor w/ 2 levels "non","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ CO     : Factor w/ 2 levels "jamais","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ TABAC  : Factor w/ 3 levels "ancienne fumeuse",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ AGE    : num  47 17 35 82 50 31 60 30 44 38 ...
```

FIGURE 23 – Sommarisation du nouveau dataset

Nous pouvons maintenant passer à l'échantillonnage de notre nouveau dataset.

4.4 Fractionnement en sous-ensembles d'apprentissage et de test : échantillonage

4.4.1 Le concept

Un échantillon est une partie d'un ensemble choisi pour représenter une ou plusieurs propriétés caractéristiques de cet ensemble. Dans cette partie, nous nous focalisons sur la répartition de nos données en données d'apprentissage et de test. L'échantillon d'apprentissage permet de créer des modèles pour un bon entraînement du modèle. L'échantillon de test permet de valider les modèles créés.

4.4.2 La taille de l'échantillon

Dans notre cas, nous allons utilisé un modèle d'échantillonnage aléatoire pour répartir nos données en deux (02) échantillons :
- d'une part des données qui serviront à la construction du modèle, avec des données d'apprentissage et
- d'autre part des données pour le test.

Nous nous proposons des ration de **70 :30** et **80 :20**; autrement dit nous spécifions que nous souhaitons que l'échantillon de test représente 30% puis 20% des données et celui de l'apprentissage constituent donc 70% et 80% des données.

```
> set.seed(100)
> train <- sample(nrow(newdataset), 0.7*nrow(newdataset), replace = FALSE)
> TrainSet <- newdataset[train,]
> TestSet <- data1[-train,]
Error: object 'data1' not found
> TestSet <- newdataset[-train,]
> summary(TrainSet)
   INFARCT     HTA          CO                   TABAC          AGE
 cas   :111   non:207   jamais:164   ancienne fumeuse: 71   Min.   :15.00
 temoin:202   oui:106   oui   :149   fumeuse         : 98   1st Qu.:33.00
                                     non             :144   Median :43.00
                                                            Mean   :44.99
                                                            3rd Qu.:55.00
                                                            Max.   :98.00

> str(TrainSet)
'data.frame':   313 obs. of  5 variables:
 $ INFARCT: Factor w/ 2 levels "cas","temoin": 1 2 2 1 1 2 1 1 2 2 ...
 $ HTA    : Factor w/ 2 levels "non","oui": 2 1 2 2 1 1 1 1 2 1 ...
 $ CO     : Factor w/ 2 levels "jamais","oui": 2 1 1 2 1 1 2 2 1 2 ...
 $ TABAC  : Factor w/ 3 levels "ancienne fumeuse",..: 3 1 3 3 1 3 2 2 3 1 ...
 $ AGE    : num  69 28 24 73 41 82 26 25 67 21 ...
> str(TestSet)
'data.frame':   135 obs. of  5 variables:
 $ INFARCT: Factor w/ 2 levels "cas","temoin": 2 2 2 2 2 2 2 2 2 2 ...
 $ HTA    : Factor w/ 2 levels "non","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ CO     : Factor w/ 2 levels "jamais","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ TABAC  : Factor w/ 3 levels "ancienne fumeuse",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ AGE    : num  50 31 30 44 38 39 50 46 17 20 ...
```

FIGURE 24 – Opération d'échantillonnage 70 : 30

```
> set.seed(100)
> train <- sample(nrow(newdataset), 0.8*nrow(newdataset), replace = FALSE)
> TrainSet <- newdataset[train,]
> TestSet <- data1[-train,]
Error: object 'data1' not found
> TestSet <- newdataset[-train,]
> str(TrainSet)
'data.frame':   358 obs. of  5 variables:
 $ INFARCT: Factor w/ 2 levels "cas","temoin": 1 2 2 1 1 2 1 1 2 2 ...
 $ HTA    : Factor w/ 2 levels "non","oui": 2 1 2 2 1 1 1 1 2 1 ...
 $ CO     : Factor w/ 2 levels "jamais","oui": 2 1 1 2 1 1 2 2 1 2 ...
 $ TABAC  : Factor w/ 3 levels "ancienne fumeuse",..: 3 1 3 3 1 3 2 2 3 1 ...
 $ AGE    : num  69 28 24 73 41 82 26 25 67 21 ...
> str(TestSet)
'data.frame':   90 obs. of  5 variables:
 $ INFARCT: Factor w/ 2 levels "cas","temoin": 2 2 2 2 2 2 2 2 2 2 ...
 $ HTA    : Factor w/ 2 levels "non","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ CO     : Factor w/ 2 levels "jamais","oui": 1 1 1 1 1 1 1 1 1 1 ...
 $ TABAC  : Factor w/ 3 levels "ancienne fumeuse",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ AGE    : num  30 44 46 17 20 40 55 46 41 75 ...
```

FIGURE 25 – Opération d'échantillonnage 80 : 20

Après la phase de préparation des données, nous passons maintenant à la modélisation.

5 La modélisation

Les données que nous avons préparées minutieusement vont être utilisées en vue de donner des résultats qui pourront éclaircir la problématique posée lorsr de la compréhension du métier. Il s'agira de sélectionner et de paramétrer les techniques de modélisation à utiliser. Les techniques utilisées à cet effet sont multiples. Le Datamining met en œuvre un ensemble de technique issues des méthodes statistiques, des analyses de données, des algorithmes et de l'informatique. Dans notre cas, nous utiliserons la modélisation supervisée (méthode prédictive), c'est-à-dire l'extrapolation de nouvelles données à partir d'une base brute.

La raison d'être des méthodes prédictives est d'expliquer ou de prévoir un ou plusieurs phénomènes observables et effectivement mesurés. Concrètement, elles vont s'intéresser à une ou plusieurs variables définies comme étant les cibles de l'analyse.

5.1 La classification par Forêts Aléatoires

5.1.1 Définition et Concept

Les forêts aléatoires sont composées d'un ensemble d'arbres décisionnels. Ces arbres se distinguent les uns des autres par le sous-échantillon de données sur lequel ils sont entraînés. Ces sous-échantillons sont tirés au hasard (d'où le terme "aléatoire") dans le jeu de données initial. Random Forest est un algorithme qui est particulièrement efficace pour repérer des liens entre une variable à expliquer et des variables explicatives. Random Forest va classer les variables explicatives en fonction de leurs liens avec la variable à expliquer.
Les forêts d'arbres décisionnels ou forêts aléatoires (Random Forest) font partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de bagging L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données. L'objectif est donc de rendre plus indépendants les arbres de l'agrégation en ajoutant du hasard dans le choix des variables qui interviennent dans les modèles.

Soit (X, Y) de loi P, un échantillon $\mathcal{L} = (x_i, y_i)_{1 \leq i \leq n}$ et un prédicteur **individuel** $\widehat{y} = \phi(x, \mathcal{L})$.

Le prédicteur **baggé** associé est, en supposant qu'on effectue un grand nombre de tirage aléatoire:

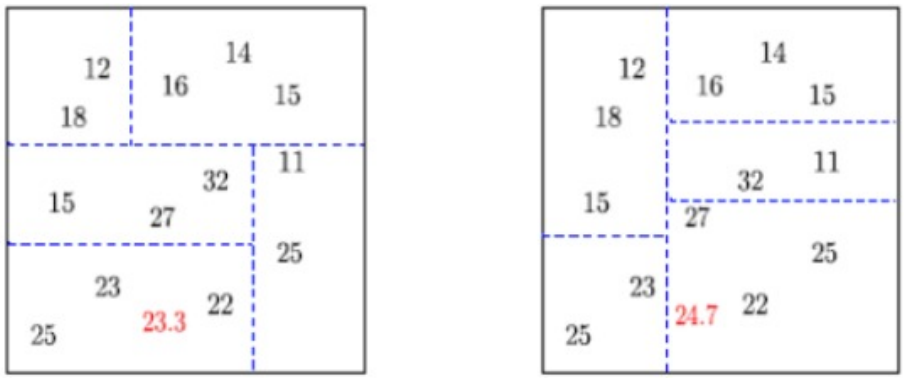
$$\phi_a(x, P) = E_{\mathcal{L}}(\phi(x, \mathcal{L}))$$

Le bagging crée des sous-ensembles d'entraînement à l'aide d'échantillonnage bootstrap. Pour créer un nouveau "base learner" :

- on tire aléatoirement avec remise n observations de l'ensemble d'entraînement.
- on entraîne notre méthode sur cet ensemble d'observations chaque base learner contient un sous ensemble des observations de l'ensemble d'entraînement.

La performance d'un "base learner" est obtenu par l'erreur out-of-bag. Les forêts aléatoires consistent à faire tourner en parallèle un grand nombre (plusieurs centaines) d'arbres de

décisions construits aléatoirement, avant de les moyenner. En termes statistiques, si les arbres sont décorrélés, cela permet de réduire la variance des prévisions.



$$\text{prévoit } \frac{24.7+23.3}{2} = 24$$

Algorithme Le bagging est appliqué à des arbres binaires de décision en ajoutant un tirage aléatoire de m variables explicatives parmi les p.

Soit $\mathbf{x}_0$ à prévoir et
$\mathbf{z} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ un échantillon
for $b = 1$ à B **do**
Tirer un échantillon bootstrap $\mathbf{z}_b^*$
Estimer un arbre sur cet échantillon avec **randomisation** des variables : la recherche de chaque division optimale est précédée d'un tirage aléatoire d'un sous-ensemble de m prédicteurs.
end for
Calculer l'estimation moyenne $\widehat{f}_B(\mathbf{x}_0) = \frac{1}{B}\sum_{b=1}^{B} \widehat{f}_{\mathbf{z}_b}(\mathbf{x}_0)$ ou le résultat du vote.

Pour notre cas d'exercice, nous allons maintenant créer un modèle de forêt aléatoire avec des paramètres par défaut, puis ajuster le modèle en modifiant «**mtry**». Nous pouvons ajuster le modèle de forêt aléatoire en modifiant le nombre d'arbres (**ntree**) et le nombre de variables échantillonnées aléatoirement à chaque étape (**mtry**). Selon la description du paquet Random Forest :

a) Choix du mtry : le nombre de variables échantillonnées au hasard en tant que

candidats à chaque division. Il correspond à la racine carrée du nombre de variables. Notons que les valeurs par défaut sont différentes pour la classification et la régression.
b) Choix du ntree : le nombre d'arbres à développer ; il ne doit pas être trop petit pour que toutes les lignes en entrée soient prédites au moins quelques fois.

5.1.2 Création du modèle de Random Forest : Cas du ration 70/30

Dans un premier temps, nous créons un modèle sans précision de paramètres. Le résulat est le suivant :

```
> model1 <- randomForest(INFARCT ~ ., data = TrainSet, importance = TRUE)
> model1

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 21.09%
Confusion matrix:
       cas temoin class.error
cas     69     42   0.3783784
temoin  24    178   0.1188119
```

FIGURE 26 – Premier version du modèle de RF

Par défaut, le nombre d'arbres est de 500 et le nombre de variables essayées à chaque division est de 2 dans ce cas. Le taux d'erreur est de **21,09%**.

On essaie un second modèle en précisant la valeur de *ntree* et *mtry*.

```
> model2 <- randomForest(INFARCT ~ ., data = TrainSet, mtry=4, importance = TRUE)
> model2

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, mtry = 4,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 28.75%
Confusion matrix:
       cas temoin class.error
cas     71     40   0.3603604
temoin  50    152   0.2475248
```

FIGURE 27 – Deuxième version du modèle de RF

Lorsque nous avons augmenté le nombre de cas de 2 à 4, le taux d'erreur a été augmenté de 21,09% à 28,75%. Ce qui ne nous avantage pas.

Il est conseillé de choisir le nombre d'arbres (ntree) en gardant la valeur par défaut de mtry et de tester plusieurs valeurs en les évaluant par exemple avec la commande suivante qui affiche un graphique montrant comment réduire l'OOB (Out-Of-Bag error) en fonction du nombre d'arbres générés.

```
> plot(fit$err.rate[, 1], type = "l", xlab = "nombre d'arbres", ylab = "erreur OOB")
```

Ce qui nous donne le graphique suivant :

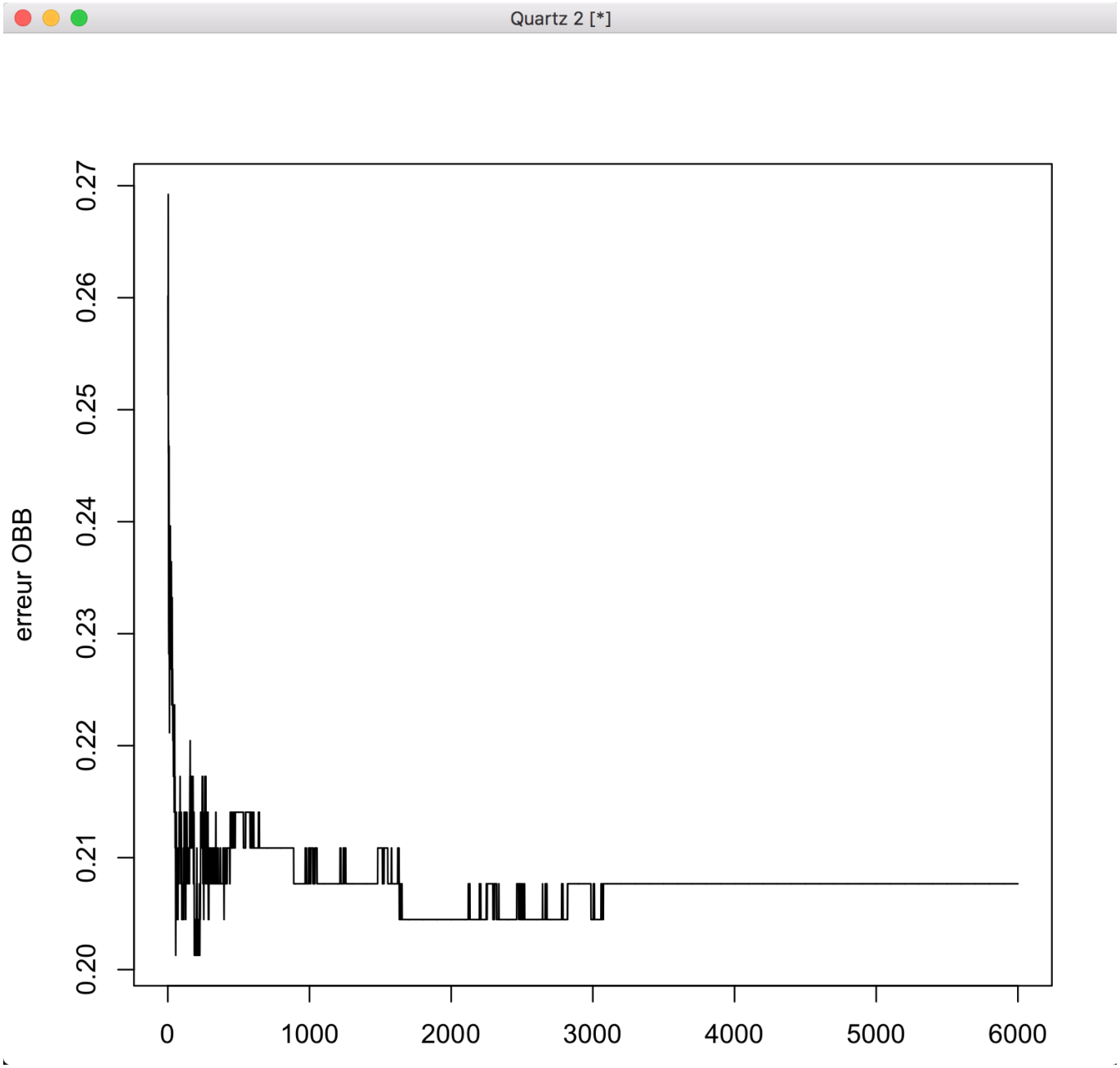


FIGURE 28 – Nombre d'arbres

On choisit ensuite le nombre d'arbres lorsque la valeur se stabilise au minimum. Nous remarquons que le graphe se stabilise à 3200. Nous avons réessayé une autre version du modèle avec **3200** arbres.

```
> model4 <- randomForest(INFARCT ~ ., data = TrainSet, mtry=2, ntree=3200, importance = TRUE)
> model4

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, mtry = 2,      ntree = 3200, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 3200
No. of variables tried at each split: 2

        OOB estimate of  error rate: 20.45%
Confusion matrix:
       cas temoin class.error
cas     70     41   0.3693694
temoin  23    179   0.1138614
```

FIGURE 29 – Troisième version du modèle RF

Ici, nous avons un taux d'erreur de **20,45%**. Ce qui est nettement mieux que les taux précédents. Nous pensons pouvoir évoluer avec ça.

Prédiction avec random Forest : Cas du ration 70/30

Nous allons maintenant prédire d'abord sur l'échantillon d'apprentissage, puis sur le jeu de données de validation.

```
> predValid <- predict(model4, TestSet, type = "class")
> mean(predValid == TestSet$INFARCT)
[1] 0.837037
```

FIGURE 30 – Prédiction avec le modèle RF

La précision obtenue est de **83,70%**.

5.1.3 Création du modèle de Random Forest : Cas du ration 80/20

Dans un premier temps, nous créons un modèle sans précision de paramètres. Le résulat est le suivant :

```
> model1 <- randomForest(INFARCT ~ ., data = TrainSet, importance = TRUE)
> model1

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 19.83%
Confusion matrix:
       cas temoin class.error
cas     81     43   0.3467742
temoin  28    206   0.1196581
```

FIGURE 31 – Premier version du modèle de RF

Par défaut, le nombre d'arbres est de 500 et le nombre de variables essayées à chaque division est de 2 dans ce cas. Le taux d'erreur est de **19,83%**.

On essaie un second modèle en précisant la valeur de *ntree* et *mtry*.

```
> model2 <- randomForest(INFARCT ~ ., data = TrainSet, mtry=4, importance = TRUE)
> model2

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, mtry = 4,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 27.37%
Confusion matrix:
       cas temoin class.error
cas     80     44   0.3548387
temoin  54    180   0.2307692
>
```

FIGURE 32 – Deuxième version du modèle de RF

Lorsque nous avons augmenté le nombre de cas de 2 à 4, le taux d'erreur a été augmenté de 19,83% à 27,37%. Ce qui ne nous avantage pas.

Ce qui nous donne le graphique suivant :

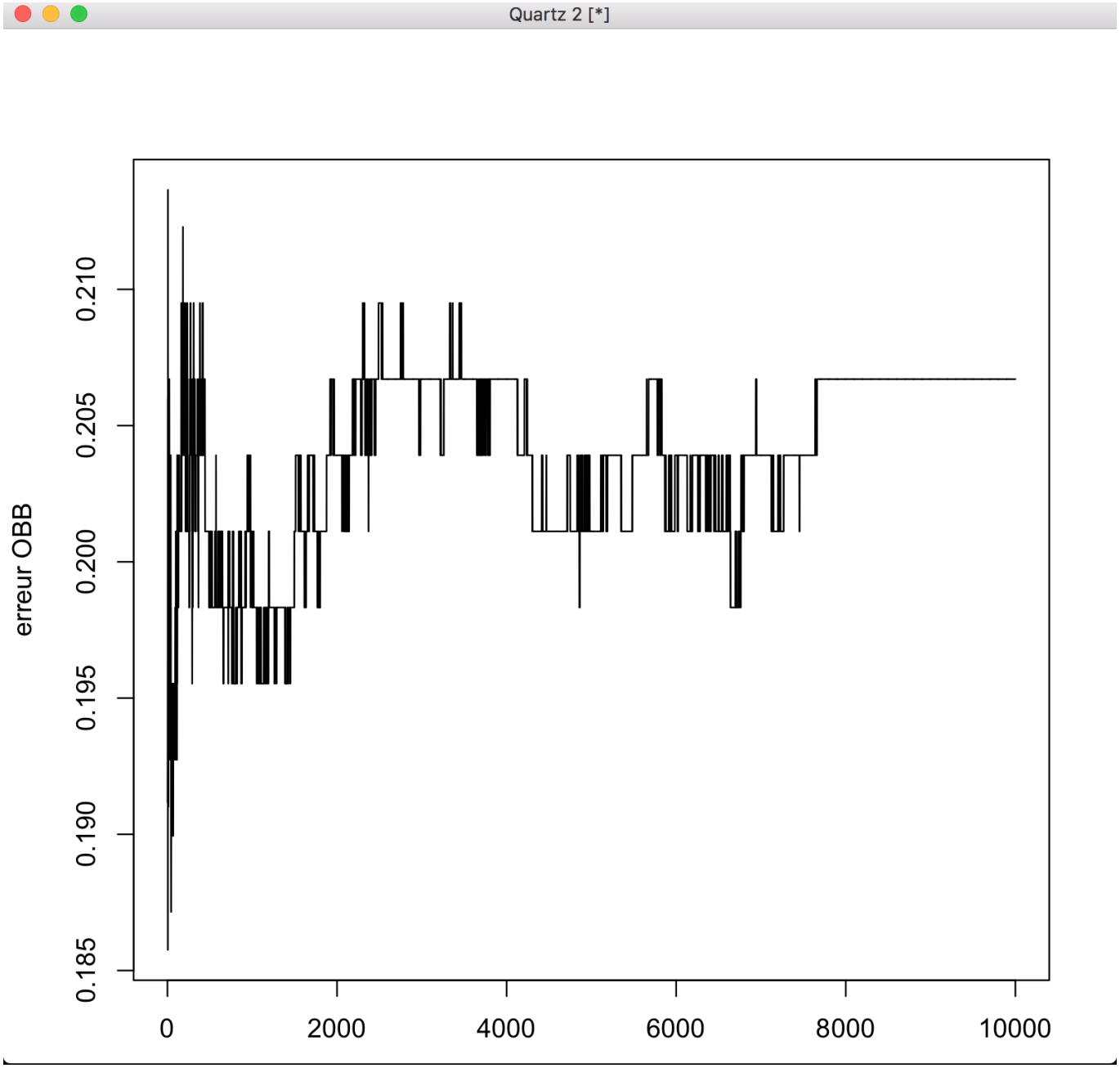


FIGURE 33 – Nombre d'arbres

On choisit ensuite le nombre d'arbres lorsque la valeur se stabilise au minimum. Nous remarquons que le graphe se stabilise à 8000. Nous avons réessayé une autre version du modèle avec **8000** arbres.

```
> model4 <- randomForest(INFARCT ~ ., data = TrainSet, mtry=2, ntree=8000, importance = TRUE)
> model4

Call:
 randomForest(formula = INFARCT ~ ., data = TrainSet, mtry = 2,      ntree = 8000, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 8000
No. of variables tried at each split: 2

        OOB estimate of  error rate: 20.67%
Confusion matrix:
       cas temoin class.error
cas     79     45   0.3629032
temoin  29    205   0.1239316
> predValid <- predict(model4, TestSet, type = "class")
> mean(predValid == TestSet$INFARCT)
[1] 0.8222222
```

FIGURE 34 – Troisième version du modèle RF

Ici, nous avons un taux d'erreur de **20,45%**. Ce qui est nettement mieux que les taux précédents. Nous pensons pouvoir évoluer avec ça.

La précision obtenue est de **82,22%**.

5.2 Classification naïve bayésienne

5.2.1 Définition et Concept

La classification naïve bayésienne est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Elle met en œuvre un classifieur bayésien naïf, ou classifieur naïf de Bayes, appartenant à la famille des classifieurs linéaires. Un terme plus approprié pour le modèle probabiliste sous-jacent pourrait être « modèle à caractéristiques statistiquement indépendantes ». Selon la nature de chaque modèle probabiliste, les classifieurs bayésiens naïfs peuvent être entraînés efficacement dans un contexte d'apprentissage supervisé. Dans beaucoup d'applications pratiques, l'estimation des paramètres pour les modèles bayésiens naïfs repose sur le maximum de vraisemblance. Autrement dit, il est possible de travailler avec le modèle bayésien naïf sans se préoccuper de probabilité bayésienne ou utiliser les méthodes bayésiennes.

Malgré leur modèle de conception « naïf » et ses hypothèses de base extrêmement simplistes, les classifieurs bayésiens naïfs ont fait preuve d'une efficacité plus que suffisante dans beaucoup de situations réelles complexes. En termes simples, un classifieur bayésien naïf suppose que l'existence d'une caractéristique pour une classe, est indépendante de l'existence d'autres caractéristiques.

5.2.2 Les étapes de la classification naïve de Bayes

Un classifieur est un algorithme permettant de définir la classe d'un objet suivant certaines de ses propriétés. Pour atteindre son objectif, cette méthode suit les étapes suivantes : - on construit le modèle avec la fonction **naiveBayes** de R. Lors du paramétrage de la procédure naiveBayes, on précise le type de dataset à utiliser. R affiche l'estimation laplacienne des probabilités.
- on prévoir la classe en fonction des probabilités conditionnelles, avec la fonction **predict** de R. -ensuite on pourra afficher le taux de confusion pour dire si le classement a été bien fait ou non.
Cette méthode permet de gagner du temps et ne fait pas intervenir la subjectivité ou le jugement du chercheur.

5.2.3 Mise en oeuvre avec notre jeu de données

Nous utiliserons l'échantillonnage 70/30 pour faire ce modèle. Nous passons à la modélisation dans un premier temps.

```
> library(e1071)
> nb.model <- naiveBayes(INFARCT ~ ., data = TrainSet)
> nb.model
```

```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
      cas    temoin
0.3322684 0.6677316

Conditional probabilities:
        HTA
Y              non       oui
  cas    0.5865385 0.4134615
  temoin 0.7033493 0.2966507

        CO
Y           jamais       oui
  cas    0.2403846 0.7596154
  temoin 0.6889952 0.3110048

        TABAC
Y        ancienne fumeuse   fumeuse       non
  cas           0.3750000 0.3942308 0.2307692
  temoin        0.1387560 0.2727273 0.5885167

        AGE
Y            [,1]     [,2]
  cas    46.77885 15.66454
  temoin 44.85646 16.59670
```

FIGURE 35 – Résultat de l'implémentation du modèle

La qualité du modèle dépend de sa capacité à bien classer dans le jeu de données test.

```
> predset <- predict(object = nb.model, newdata = ValidSet)
> valMod <- cbind(ValidSet, predset)
> head(valMod, 5)
   INFARCT HTA     CO TABAC AGE predset
1   temoin non jamais   non  47  temoin
2   temoin non jamais   non  17  temoin
5   temoin non jamais   non  50  temoin
7   temoin non jamais   non  60  temoin
12  temoin non jamais   non  62  temoin
```

FIGURE 36 – Vérification de l'implémentation du modèle

Voyons à présent comment le classement a été bien fait

```
> Confusion = table(valMod$INFARCT, valMod$predset)
> Confusion

         cas temoin
  cas     26     18
  temoin   8     83
> round(prop.table(Confusion), 2)

          cas temoin
  cas    0.19   0.13
  temoin 0.06   0.61
```

FIGURE 37 – Vérification du classement

Soit un taux de bon classement de 80%.

6 L'évaluation

Pour chaque modèle étudié, il est judicieux d'effectuer une évaluation méthodique. Nous allons comparer nos différents modèles que nous avons construit afin de choisir le modèle qui résout efficacement notre problématique. Nos remarquons que le taux de classement des méthodes utilisées sont pratiquement égaux. Ceci nous permet de conclure que la première méthode (Random Forest) est bonne et on peut la garder.
La comparaison des modèles ci-dessus montre le véritable pouvoir de l'ensemencement

et l'importance d'utiliser Random Forest sur les arbres de décision. Bien que Random Forest ait ses propres limites inhérentes (en termes de nombre de niveaux de facteurs qu'une variable catégorique peut avoir), il reste néanmoins l'un des meilleurs modèles pouvant être utilisés pour la classification. Il est facile à utiliser et à ajuster par rapport à certains autres modèles complexes, tout en nous offrant un bon niveau de précision dans le scénario commercial. Vous pouvez également comparer Random Forest avec d'autres modèles et voir comment il se comporte par rapport à d'autres techniques.

7 Le déploiement

Le déploiement est le processus consistant à utiliser les nouvelles connaissances issues de cette étude pour apporter des améliorations pour de futurs cas d'étude. En ce qui concerne notre projet, cette phase sera exécutée lors que le besoin se fera sentir auprès d'une organisation qui en aura besoin.

Conclusion

Pour ce travail de recherche, nous avons choisi le domaine médical surtout les données sur l'infarctus. Grâce au logiciel R, nous avons pu faire les études statistiques sur les variables qualitatives et quantitatives du jeu de données choisi. Le travail que nous avons effectué nous a permis de réaliser l'importance de l'apprentissage supervisé notamment dans la prédiction. Pour ce faire, nous avons d'abord réalisé un modèle qui s'appuie sur la méthode de Random Forest. Après, nous l'avons évaluer avec une seconde méthode nommée méthode Naïve Bayésienne. Les résultats obtenus ont révélé que les méthodes ne présentaient pas le même taux de classement. Néanmoins, nous gardons notre modèle Random Forest vu ses avantages et la qualité de ses résultats. Au regard de notre faible maîtrise en mécanique, notre étude pourrait donc être plus approfondie dans le futur. En perspective de ce travail, on pourrait chercher à faire cet apprentissage avec d'autres méthodes d'apprentissage surpervisé pour voir la possibilité d'avoir de meilleurs résultats tels que SVM, Knn, etc... Cependant, aucune méthodologie n'est parfaite, et la clé de la réussite résidera toujours dans l'implication constante des métiers pour une amélioration continue du produit final. Après tout, comme disait le statisticien George E.P.BOX : "All models are wrong, but some are useful » (« Tous les modèles sont faux mais certains sont utiles »).

Références

— Source de données : http ://www.biostatisticien.eu/springeR/jeuxDonnees5.html

— https ://www.solutions-numeriques.com/expertise-la-methode-crisp-une-solution-pour-reussir-vos-projets-big-data-alianor-sibai-mc2i-groupe/

— https ://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/

— https ://maximilienandile.github.io/2016/09/29/Machine-Learning-comment-fonctionne-la-classification-naive-Bayesienne/

— https ://www.captaineconomics.fr/-classification-naive-bayesienne-supervisee

— http ://eric.univ-lyon2.fr/ ricco/tanagra/fichiers/fr_Tanagra_Naive_Bayes_Classifier_Explained.pdf

— https ://www.r-bloggers.com/naive-bayes-classification-in-r-part-2/

— https ://www.r-bloggers.com/how-to-implement-random-forests-in-r/