

**Institut Supérieur des Sciences Appliquées et Économiques**

**associé au Conservatoire National des Arts et Métiers**

**Entreposage et Fouille de données-STA 211**

**Devoir 3 :**

**Prétraitement des données**

**Prépare par : …Elio Bou Serhal………….**

**Date :…Le…Mardi 11 /4/2023…….**

# **Objective**

# **Résumé ou partie théorie**

# **Déscription de la base de données**

# **Représentation graphiques de la base de données :**

# **Application la méthode sur la base de données**

# **Conclusion**

# **Références**

# **Annexes**

Table de matières

[**Objective** 1](#_Toc90386381)

[**Résumé ou partie théorie** 1](#_Toc90386382)

[**Déscription de la base de données** 1](#_Toc90386383)

[**Représentation graphiques pour les variables :** 1](#_Toc90386384)

[**Application la méthode sur la base de données** 1](#_Toc90386385)

[**Conclusion** 1](#_Toc90386386)

[**Réfèrences** 1](#_Toc90386387)

[**Liste des figures** 3](#_Toc90386388)

[**Liste des tableaux** 4](#_Toc90386389)

[Figure 1: La repartition de l'Age 18](#_Toc131885646)

[Figure 2: La densite de l'Age 19](#_Toc131885647)

[Figure 3: La distribution de l'Age 19](#_Toc131885648)

[Figure 4: Boite a moustache de l'Age 19](#_Toc131885649)

[Figure 5: Les différentes figures de la variable "Age" 20](#_Toc131885650)

[Figure 6: La repartition de la variable "sex" 21](#_Toc131885651)

[Figure 7: Les differentes figurent de la variable "sex" 21](#_Toc131885652)

[Figure 8: La répartition de la variable "tresbps" 22](#_Toc131885653)

[Figure 9: La distribution de la variable « trestbps » 22](#_Toc131885654)

[Figure 10: La distribution de la variable "tretbps" 23](#_Toc131885655)

[Figure 11: Boite a Moustache de la variable "trestbps" 23](#_Toc131885656)

[Figure 12: La repartition de la variable "chol" 24](#_Toc131885657)

[Figure 13: La densite de la variable "chol" 24](#_Toc131885658)

[Figure 14: La distribution de la variable "chol" 25](#_Toc131885659)

[Figure 15: Boite a moustache de la variable "chol" 26](#_Toc131885660)

[Figure 16: figure montrant la distribution et la repartition de la variable "chol" 26](#_Toc131885661)

[Figure 17: la répartition de la variable "cp" 27](#_Toc131885662)

[Figure 18: Boite a moustache de la variable "cp" 27](#_Toc131885663)

[Figure 19: Les différentes figures de la variable « cp » 28](#_Toc131885664)

[Figure 20: La répartition de la variable "fbs" 29](#_Toc131885665)

[Figure 21: La repartition denla variable "thalach" 29](#_Toc131885666)

[Figure 22: La densite de la variable "thalach" 30](#_Toc131885667)

[Figure 23: La distribution de la variable "thalach" 30](#_Toc131885668)

[Figure 24: Boite a moustache de la variable "thalach" 31](#_Toc131885669)

[Figure 25: La répartition de la poitrine de la variable "exang" 31](#_Toc131885670)

[Figure 26 : la répartition de la variable catégorielle "exang" 32](#_Toc131885671)

[Figure 27: La répartition du résultats electrographique au repos 32](#_Toc131885672)

[Figure 28: La répartition de la variable "slope" 33](#_Toc131885673)

[Figure 29: Boite a moustache de la variable "slope" 34](#_Toc131885674)

[Figure 30: La répartition de la dépression ST induite par l'exercice par rapport au repos 34](#_Toc131885675)

[Figure 31: La densité de la variable "oldpeak" 35](#_Toc131885676)

[Figure 32: La distribution de la variable "oldpeak" 35](#_Toc131885677)

[Figure 33: Boite de moustache de la variable "oldpeak" 36](#_Toc131885678)

[Figure 34: La répartition de la variable "CA" 36](#_Toc131885679)

[Figure 35: Boite a moustache de la variable "CA" 37](#_Toc131885680)

[Figure 36: La répartition de la variable "thal" 37](#_Toc131885681)

[Figure 37: Boite a moustache de la variable "thal" 38](#_Toc131885682)

[Figure 38: les différentes représentations de la variable "thal' 38](#_Toc131885683)

[Figure 39: Courbe de Roc du modèle 1- cas 1 40](#_Toc131885684)

[Figure 40: Intervalle de Confiance de Roc du modèle 1- cas 1 40](#_Toc131885685)

[Figure 41: Courbe de Roc du modèle 1- cas 2 41](#_Toc131885686)

[Figure 42: Intervalle de Confiance de Roc du modèle 1- cas 2 41](#_Toc131885687)

[Figure 43: Courbe de Roc du modèle 1- cas 3 43](#_Toc131885688)

[Figure 44: Intervalle de Confiance de Roc du modèle 1- cas 3 43](#_Toc131885689)

[Figure 45: Courbe de Roc du modèle 1- cas 4 45](#_Toc131885690)

[Figure 46: Intervalle de Confiance de Roc du modèle 1- cas 4 45](#_Toc131885691)

[Figure 47: Courbe de Roc du modèle 1- cas 5 46](#_Toc131885692)

[Figure 48: Intervalle de Confiance de Roc du modèle 1- cas 5 46](#_Toc131885693)

[Figure 49: Temps d'exécution de chaque cas de l’arbre de décision 47](#_Toc131885694)

[Figure 50: Courbe montrant l'erreur d'OOB 48](#_Toc131885695)

[Figure 51: Courbe de Roc du modèle 2- cas 1 49](#_Toc131885696)

[Figure 52: Intervalle de Confiance de Roc du modèle 2- cas 1 49](#_Toc131885697)

[Figure 53: Courbe de Roc du modèle 2- cas 2 51](#_Toc131885698)

[Figure 54: Intervalle de Confiance de Roc du modèle 2- cas 2 51](#_Toc131885699)

[Figure 55: Courbe de Roc du modèle 2- cas 3 52](#_Toc131885700)

[Figure 56: Intervalle de Confiance de Roc du modèle 2- cas 3 53](#_Toc131885701)

[Figure 57: Courbe de Roc du modèle 2- cas 4 54](#_Toc131885702)

[Figure 58: Intervalle de Confiance de Roc du modèle 2- cas 4 54](#_Toc131885703)

[Figure 59: Courbe de Roc du modèle 2- cas 5 55](#_Toc131885704)

[Figure 60: Intervalle de Confiance de Roc du modèle 2- cas 5 55](#_Toc131885705)

[Figure 61: Temps d'exécution de chaque cas du « Bagging » 56](#_Toc131885706)

[Figure 62: Figure montrant l'erreur du «boostrap » méthode 57](#_Toc131885707)

[Figure 63: Courbe de Roc du modèle 3- cas 1 58](#_Toc131885708)

[Figure 64: Intervalle de Confiance de Roc du modèle 3- cas 1 58](#_Toc131885709)

[Figure 65: Figure montrant l'erreur «  oob » par rapport au nbr d’arbre prit 59](#_Toc131885710)

[Figure 66: Figure montrant l'erreur "oob" par rapport au nbr d'arbre prit et par rapport a la variable cible 59](#_Toc131885711)

[Figure 67: Figure montrant l'erreur "oob" par rapport au parametre mtry 60](#_Toc131885712)

[Figure 68: Figure montrant l'erreur "oob" par rapport au nbr d'arbre prit 60](#_Toc131885713)

[Figure 69: Courbe de Roc du modèle 3- cas 4 62](#_Toc131885714)

[Figure 70: Intervalle de Confiance de Roc du modèle 3- cas 4 62](#_Toc131885715)

[Figure 71: Temps d'exécution de Random Forest cas 4 63](#_Toc131885716)

[Figure 72: Courbe de Roc du modèle 3- cas 4 65](#_Toc131885717)

[Figure 73: Intervalle de Confiance de Roc du modèle 3- cas 5 65](#_Toc131885718)

[Figure 74: Temps d'exécution de Random Forest cas 5 65](#_Toc131885719)

[Figure 75: Courbe de Roc du modèle 3- cas 6 68](#_Toc131885720)

[Figure 76: Intervalle de Confiance de Roc du modèle 3- cas 6 68](#_Toc131885721)

[Figure 77: Temps d'exécution de Random Forest cas 6 68](#_Toc131885722)

[Figure 78: Courbe de Roc du modèle 3- cas 7 70](#_Toc131885723)

[Figure 79: Intervalle de Confiance de Roc du modèle 3- cas 7 70](#_Toc131885724)

[Figure 80: Temps d'exécution de Random Forest cas 7 71](#_Toc131885725)

[Figure 81: Courbe de Roc du modèle 4 72](#_Toc131885726)

[Figure 82: Intervalle de Confiance de Roc du modèle 4 72](#_Toc131885727)

[Figure 83: La courbe de Roc des 4 modeles 73](#_Toc131885728)

[Figure 84: Temps d'execution de 4 modeles 74](#_Toc131885729)

[Table 1: RESULTAS-1 DE LA FONCTION "summary" 12](#_Toc131885730)

[Table 2: RESULTAS-2 DE LA FONCTION "summary" 12](#_Toc131885731)

[Table 3: RESULTATS(1) DE LA FONCTION "describe" 14](#_Toc131885732)

[Table 4: RESULTATS(2) DE LA FONCTION "describe" 14](#_Toc131885733)

[Table 5: RESULTATS(1) DE LA FONCTION "describeBy" pour grp 0 15](#_Toc131885734)

[Table 6: RESULTATS(2) DE LA FONCTION "describeBy" pour grp 0 15](#_Toc131885735)

[Table 7: RESULTATS(1) DE LA FONCTION "describeBy" pour grp 1 16](#_Toc131885736)

[Table 8: RESULTATS(2) DE LA FONCTION "describeBy" pour grp 1 16](#_Toc131885737)

[Table 9: Matrice de confusion du modèle 1-cas 1 40](#_Toc131885738)

[Table 10: Tableau d'évaluation du modèle 1-cas 1 40](#_Toc131885739)

[Table 11: Matrice de confusion du modèle 1-cas 2 42](#_Toc131885740)

[Table 12: Tableau d'évaluation du modèle 1-cas 2 42](#_Toc131885741)

[Table 13: Matrice de confusion de la modèle 1-cas 3 43](#_Toc131885742)

[Table 14: Tableau d'évaluation du modèle 1-cas 3 43](#_Toc131885743)

[Table 15: Matrice de confusion du modèle 1-cas 2 45](#_Toc131885744)

[Table 16: Tableau d'évaluation de la modèle 1-cas 4 45](#_Toc131885745)

[Table 17: Matrice de confusion du modèle 1-cas 5 47](#_Toc131885746)

[Table 18: Tableau d'évaluation de la modèle 1-cas 4 47](#_Toc131885747)

[Table 19: Temps d'exécution de chaque cas de l’arbre de décision 48](#_Toc131885748)

[Table 20: Matrice de confusion du modèle 2-cas 1 50](#_Toc131885749)

[Table 21: Tableau d'évaluation de la modèle 2-cas 1 50](#_Toc131885750)

[Table 22: Matrice de confusion du modèle 2-cas 2 51](#_Toc131885751)

[Table 23: Tableau d'évaluation de la modèle 2-cas 2 51](#_Toc131885752)

[Table 24: Matrice de confusion du modèle 2-cas 3 53](#_Toc131885753)

[Table 25: Tableau d'évaluation de la modèle 2-cas 3 53](#_Toc131885754)

[Table 26: Matrice de confusion du modèle 2-cas 4 54](#_Toc131885755)

[Table 27: Tableau d'évaluation de la modèle 2-cas 4 54](#_Toc131885756)

[Table 28: Matrice de confusion du modèle 2-cas 5 56](#_Toc131885757)

[Table 29: Tableau d'évaluation de la modèle 2-cas 5 56](#_Toc131885758)

[Table 30: Temps d'exécution de chaque cas du « Bagging » 57](#_Toc131885759)

[Table 31: Matrice de confusion du modèle 3-cas 4 62](#_Toc131885760)

[Table 32: Tableau d'évaluation de la modèle 3-cas 4 62](#_Toc131885761)

[Table 33: Temps d'exécution de Random Forest cas 4 63](#_Toc131885762)

[Table 34: Matrice de confusion du modèle 3-cas 5 65](#_Toc131885763)

[Table 35: Tableau d'évaluation de la modèle 3-cas 5 65](#_Toc131885764)

[Table 36: Temps d'exécution de Random Forest cas 5 66](#_Toc131885765)

[Table 37: Matrice de confusion du modèle 3-cas 6 68](#_Toc131885766)

[Table 38: Tableau d'évaluation de la modèle 3-cas 6 68](#_Toc131885767)

[Table 39: Temps d'exécution de Random Forest cas 6 69](#_Toc131885768)

[Table 40: Matrice de confusion du modèle 3-cas 7 70](#_Toc131885769)

[Table 41: Tableau d'évaluation de la modèle 3-cas 7 70](#_Toc131885770)

[Table 42: Temps d'exécution de Random Forest cas 7 72](#_Toc131885771)

[Table 43: Matrice de confusion du modèle 4 73](#_Toc131885772)

[Table 44: Temps d'execution de 4 modeles 75](#_Toc131885773)

# **Objective**

1. Appliquer le modèle « Arbre de décision » sur la base de données
2. Appliquer le modèle « Bagging » sur la base de données
3. Appliquer le modèle « Foret Aléatoire » sur la base de données
4. Comparer ces 4 modèles ensemblistes en utilisant la courbe de Roc
5. Comparer ces 4 modèles ensemblistes par rapport au temps d’exécution

L'objectif principal est de prédire si une personne donnée souffre ou non d'une maladie cardiaque, à l'aide de plusieurs facteurs qui causent, par exemple, l'âge, le taux de cholestérol, le type de douleur thoracique, etc.

# **Résumé ou partie théorie**

Dans notre rapport on va voir en détails chaque méthode utilise, mais avant je veux expliquer le fonctionnement de chaque algorithme :

1. **MODELE 1 : DECISION TREE**

L'algorithme de l'arbre de décision est un algorithme populaire d'apprentissage automatique qui peut être utilisé pour des tâches de classification et de régression. Les étapes générales pour construire un algorithme d'arbre de décision sont les suivantes :

1. Collecter et préparer les données : Rassembler les données pour le problème et les préparer pour l'analyse. Cela comprend le nettoyage des données, la suppression des valeurs manquantes et la transformation des données dans un format approprié pour l'analyse.

2. Définir la variable cible : Identifier la variable que vous souhaitez prédire, connue sous le nom de variable cible.

3. Sélectionner les caractéristiques : Sélectionner les variables (caractéristiques) qui seront utilisées pour prédire la variable cible.

4. Diviser les données : Diviser les données en ensembles d'entraînement et de test. L'ensemble d'entraînement sera utilisé pour construire le modèle d'arbre de décision, et l'ensemble de test sera utilisé pour évaluer le modèle.

5. Construire l'arbre : L'algorithme de l'arbre de décision construit une structure d'arbre en divisant à plusieurs reprises les données en sous-ensembles en fonction de la caractéristique la plus importante. L'objectif est de maximiser le gain d'information à chaque division, qui est une mesure de la capacité de la caractéristique à expliquer la variable cible.

6. Élaguer l'arbre : L'arbre de décision est ensuite élagué pour éviter le sur ajustement. Cela implique de supprimer les branches qui ne contribuent pas de manière significative à la prédiction de la variable cible.

7. Évaluer le modèle : Une fois que l'arbre de décision a été construit et élagué, évaluez ses performances sur l'ensemble de test. Cela peut être fait en calculant des mesures telles que la précision et le rappel.

8. Utiliser le modèle : Une fois que le modèle a été évalué et jugé satisfaisant, il peut être utilisé pour faire des prédictions sur de nouvelles données.

1. **MODELE 2 : BAGGING**

L'algorithme de Bagging (Bootstrap Aggregation) est une méthode d'apprentissage automatique qui vise à améliorer la précision d'un modèle en combinant les prédictions de plusieurs modèles. Voici les étapes générales pour mettre en œuvre l'algorithme de Bagging :

1. Collecter les données : Rassembler les données pour le problème et les préparer pour l'analyse.

2. Sélectionner le modèle : Choisir le modèle d'apprentissage automatique à utiliser pour le problème.

3. Diviser les données : Diviser les données en plusieurs sous-ensembles aléatoires de même taille. Chaque sous-ensemble est appelé un échantillon de bootstrap.

4. Entraîner les modèles : Pour chaque échantillon de bootstrap, entraîner un modèle sur cet échantillon.

5. Prédire : Une fois que tous les modèles ont été entraînés, faire une prédiction pour chaque modèle sur les données de test.

6. Combinaison : Agréger les prédictions de tous les modèles en utilisant une méthode de combinaison, comme la moyenne ou le vote majoritaire.

7. Évaluer le modèle : Évaluer la performance du modèle combiné en utilisant des métriques de performance telles que l'exactitude, la précision, le rappel et le score F1.

8. Utiliser le modèle : Une fois que le modèle a été évalué et jugé satisfaisant, il peut être utilisé pour faire des prédictions sur de nouvelles données.

Notez que l'algorithme de Bagging peut être utilisé avec n'importe quel modèle d'apprentissage automatique, comme les arbres de décision, les réseaux de neurones, ou les modèles de régression. De plus, il existe plusieurs variations de l'algorithme de Bagging, comme le Boosting et le Random Forest, qui utilisent des techniques différentes pour améliorer la précision du modèle.

1. **MODELE 3 : FORET ALEATOIRE**

L'algorithme de Random Forest est une méthode d'apprentissage automatique utilisée pour les tâches de classification, de régression et de détection d'anomalies. Cette méthode combine les prédictions de plusieurs arbres de décision pour améliorer la précision et la stabilité du modèle. Voici les étapes générales pour mettre en œuvre l'algorithme de Random Forest :

1. Collecter et préparer les données : Rassembler les données pour le problème et les préparer pour l'analyse. Cela comprend le nettoyage des données, la suppression des valeurs manquantes et la transformation des données dans un format approprié pour l'analyse.

2. Sélectionner les caractéristiques : Sélectionner les variables (caractéristiques) qui seront utilisées pour prédire la variable cible.

3. Diviser les données : Diviser les données en ensembles d'entraînement et de test. L'ensemble d'entraînement sera utilisé pour construire le modèle de Random Forest, et l'ensemble de test sera utilisé pour évaluer le modèle.

4. Construire les arbres : L'algorithme de Random Forest construit plusieurs arbres de décision indépendants les uns des autres, en utilisant des sous-ensembles aléatoires de caractéristiques et d'échantillons d'entraînement.

5. Agréger les prédictions : Une fois que tous les arbres ont été construits, les prédictions de chaque arbre sont agrégées pour produire une prédiction finale pour chaque exemple.

6. Évaluer le modèle : Évaluer la performance du modèle de Random Forest en utilisant des métriques de performance telles que l'exactitude, la précision, le rappel et le score F1.

7. Utiliser le modèle : Une fois que le modèle a été évalué et jugé satisfaisant, il peut être utilisé pour faire des prédictions sur de nouvelles données.

Notez que l'algorithme de Random Forest utilise une méthode de bagging pour construire plusieurs arbres de décision indépendants les uns des autres, qui sont ensuite agrégés pour améliorer la précision du modèle. Les variations de l'algorithme de Random Forest incluent l'utilisation de sous-espaces aléatoires de caractéristiques pour construire chaque arbre, et l'utilisation de forêts profondes avec plusieurs couches d'arbres.

1. **MODELE 4 : BOOSTING**

L'algorithme de Boosting est une méthode d'apprentissage automatique qui vise à améliorer la précision d'un modèle en apprenant de manière itérative à partir des erreurs du modèle précédent. Voici les étapes générales pour mettre en œuvre l'algorithme de Boosting :

1. Collecter et préparer les données : Rassembler les données pour le problème et les préparer pour l'analyse. Cela comprend le nettoyage des données, la suppression des valeurs manquantes et la transformation des données dans un format approprié pour l'analyse.

2. Sélectionner le modèle : Choisir le modèle d'apprentissage automatique à utiliser pour le problème. Les modèles les plus couramment utilisés pour l'algorithme de Boosting sont les arbres de décision.

3. Diviser les données : Diviser les données en ensembles d'entraînement et de test. L'ensemble d'entraînement sera utilisé pour construire le modèle de Boosting, et l'ensemble de test sera utilisé pour évaluer le modèle.

4. Initialiser le modèle : Entraîner un premier modèle sur l'ensemble d'entraînement.

5. Pondération des exemples : Pour chaque exemple d'entraînement, attribuer une pondération qui dépend de sa difficulté à être prédit correctement. Les exemples mal prédits par le modèle précédent ont une pondération plus élevée que les exemples bien prédits.

6. Entraînement itératif : Entraîner plusieurs modèles de manière itérative en utilisant une version pondérée de l'ensemble d'entraînement pour chaque modèle. À chaque étape, les exemples mal prédits par les modèles précédents ont une pondération plus élevée, ce qui permet au modèle suivant de se concentrer sur ces exemples difficiles.

7. Agrégation des prédictions : Agréger les prédictions de tous les modèles en utilisant une méthode de combinaison, comme la moyenne pondérée ou le vote pondéré.

8. Évaluer le modèle : Évaluer la performance du modèle de Boosting en utilisant des métriques de performance telles que l'exactitude, la précision, le rappel et le score F1.

9. Utiliser le modèle : Une fois que le modèle a été évalué et jugé satisfaisant, il peut être utilisé pour faire des prédictions sur de nouvelles données.

Notez que l'algorithme de Boosting peut être utilisé avec n'importe quel modèle d'apprentissage automatique, mais est particulièrement utile pour améliorer les performances des modèles faibles. Les variations de l'algorithme de Boosting incluent le Gradient Boosting et l'AdaBoost, qui utilisent des techniques différentes pour mettre à jour les pondérations des exemples et les coefficients des modèles.

**FINALEMENT :**

Les algorithmes de l'arbre de décision, du bagging, de la forêt aléatoire et du boosting sont toutes des techniques d'apprentissage automatique utilisées pour la modélisation prédictive. Cependant, ils diffèrent dans leur approche de construction de modèles et leur capacité à traiter différents types de données. Voici une comparaison de ces quatre algorithmes :

1. Arbre de décision : Un arbre de décision est un algorithme d'apprentissage supervisé qui peut être utilisé pour les tâches de régression et de classification. L'algorithme construit un modèle en forme d'arbre de décisions et de leurs conséquences possibles, chaque nœud représentant une décision basée sur une valeur de caractéristique. L'algorithme choisit la meilleure division en fonction d'une mesure d'impureté, telle que l'entropie ou l'indice de Gini. Les arbres de décision sont simples et interprétables, mais ils ont tendance à surajuster et peuvent ne pas convenir aux ensembles de données complexes.

2. Bagging : Le bagging (agrégation de bottes) est une méthode d'ensemble qui consiste à construire plusieurs modèles indépendants sur des sous-ensembles aléatoires des données d'apprentissage, puis à combiner leurs prédictions par vote majoritaire ou moyenne. L'idée derrière le bagging est de réduire la variance des modèles et d'empêcher le surapprentissage. Le bagging peut être utilisé avec n'importe quel algorithme de base, tel que l'arbre de décision, et est efficace pour les modèles à haute variance ou les données bruyantes.

3. Forêt aléatoire : La forêt aléatoire est une extension du bagging qui utilise des arbres de décision comme estimateur de base. Dans une forêt aléatoire, chaque arbre est construit sur un sous-ensemble aléatoire des caractéristiques et des données, ce qui réduit la corrélation entre les modèles et améliore la généralisation. La forêt aléatoire est efficace pour les ensembles de données volumineux et les problèmes non linéaires.

4. Boosting : Le boosting est une méthode d'ensemble qui entraîne de manière itérative des modèles en se concentrant sur les exemples qui ont été mal classés par les modèles précédents. Les modèles sont pondérés en fonction de leurs performances, de sorte que les modèles les mieux performants ont un poids plus élevé. Le boosting est efficace pour les modèles faibles et les ensembles de données déséquilibrés.

En conclusion, chaque algorithme a ses forces et ses faiblesses, et le choix de l'algorithme dépend du problème spécifique et des caractéristiques des données. Les arbres de décision sont simples et interprétables, mais ont tendance à surajuster. Le bagging est efficace pour les modèles à haute variance et les données bruyantes. La forêt aléatoire est efficace pour les ensembles de données volumineux et les problèmes non linéaires. Le boosting est efficace pour les modèles faibles et les ensembles de données déséquilibrés.

# **Déscription de la base de données**

Les données comportent 303 observations et 14 variables.

Chaque observation contient les informations suivantes sur un individu.

1. **Age** : âge de l'individu en années

2. **Sex** : Sexe de l'individu (1 = homme ; 0 = femme)

3. **Cp** : Type de douleur thoracique (1 = angine typique ; 2 = angine atypique ; 3 = douleur non angineuse ; 4 = asymptomatique)

4. **Trestbps** - Tension artérielle au repos (en mm Hg à l'admission à l'hôpital)

5. **chol** - Cholestérol sérique en mg/dl

6. **fbs** - Glycémie à jeun > 120 mg/dl (1 = vrai ; 0 = faux)

7. **Restecg** - Résultats électrocardiographiques au repos (0 = normal ; 1 = ayant ST-T ; 2 = hypertrophie)

8. T**halach** - Fréquence cardiaque maximale atteinte

9. **Exang** – Angine de poitrine induite par l'effort (1 = oui ; 0 = non)

10. **Oldpeak** - Dépression ST induite par l'exercice par rapport au repos

11. **Slope** - La pente du segment ST d'exercice maximal (1 = montant ; 2 = plat ; 3 = descendant) 12. **Ca**- Nombre de vaisseaux principaux (0-4) colorés par fluoroscopie

13. **Thal** -thalassémie est une maladie héréditaire du sang qui affecte la capacité de l'organisme à produire de l'hémoglobine et des globules rouges. 1 = normale ; 2 = défaut corrigé ; 3 = défaut réversible

14. **Target** - l'attribut prédit - diagnostic de maladie cardiaque (état de la maladie angiographique) (Valeur 0 = < 50 % de rétrécissement du diamètre ; Valeur 1 = > 50 % de rétrécissement du diamètre)

On s'intéresse à détecter si le patient souffre d'une maladie.

En utilisant la fonction « summary » sur R on obtient les resultats suivantes:

On peut remarquer que les moyennes et les écarts types varient selon les groupes.

Par exemple, les personnes du groupe 0 ont une tension artérielle moyenne plus élevée que les personnes du groupe 1,

tandis que les personnes du groupe 1 ont un taux de cholestérol moyen plus élevé que les personnes du groupe 0.

Les résultats de la cible indiquent que toutes les personnes du groupe 0 sont non atteintes de maladies cardiaques,

alors que l'on ne peut rien déduire pour le groupe 1.

Les résultats présentés dans cette table sont des statistiques descriptives (min, max, moyenne, médiane, quartiles)

pour chaque variable du jeu de données. Voici quelques interprétations possibles pour chaque variable :

**age** : l'âge des patients varie de 29 à 77 ans, avec une moyenne de 54 ans.

**sex** : environ 68% des patients sont des hommes (sex=0) et 32% sont des femmes (sex=1).

**cp** : la majorité des patients ont une douleur thoracique typique angine de poitrine (cp=1)

ou une douleur non typique (cp=2), tandis que moins de patients ont une douleur atypique (cp=0)

ou pas de douleur (cp=3).

**trestbps** : la pression artérielle au repos varie de 94 à 200 mmHg, avec une moyenne de 131 mmHg.

**chol** : le taux de cholestérol varie de 126 à 564 mg/dl, avec une moyenne de 246 mg/dl.

**fbs** : moins de 15% des patients ont une glycémie à jeun élevée (fbs=1).

**restecg** : environ 53% des patients ont un électrocardiogramme anormal (restecg=1 ou 2).

**thalach** : la fréquence cardiaque maximale atteinte varie de 71 à 202 bpm, avec une moyenne de 149 bpm.

**exang** : environ 33% des patients ont une angine de poitrine induite par l'exercice (exang=1).

**oldpeak** : le ST dépression induit par l'exercice varie de 0 à 6.2 mm, avec une moyenne de 1.04 mm.

**slope** : la majorité des patients ont une pente d'onde ST positive (slope=2) ou plate (slope=1),

tandis que moins de patients ont une pente négative (slope=0).

**ca** : le nombre de vaisseaux sanguins principaux colorés par fluoroscopie varie de 0 à 4, avec une moyenne de 0.73.

**thal** : la plupart des patients ont une thalassémie normale (thal=2),

tandis que moins de patients ont une thalassémie réversible (thal=3) ou une thalassémie fixe (thal=1).

**target** : environ 54% des patients ont une maladie cardiaque (target=1) et 46% n'en ont pas (target=0).

Table 1: RESULTAS-1 DE LA FONCTION "summary"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **age** | **sex** | **cp** | **trestbps** | **chol** | **fbs** | **restecg** |
| **Min.** | 29 | 0 | 0 | 94 | 126 | 0 | 0 |
| **1st Qu.** | 47.5 | 0 | 0 | 120 | 211 | 0 | 0 |
| **Median** | 55 | 1 | 1 | 130 | 240 | 0 | 1 |
| **Mean** | 54.37 | 0.6832 | 0.967 | 131.6 | 246.3 | 0.1485 | 0.5281 |
| **3rd Qu.** | 61 | 1 | 2 | 140 | 274.5 | 0 | 1 |
| **Max.** | 77 | 1 | 3 | 200 | 564 | 1 | 2 |

Table 2: RESULTAS-2 DE LA FONCTION "summary"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **thalach** | **exang** | **oldpeak** | **slope** | **ca** | **thal** | **target** |
| **Min.** | 71 | 0 | 0 | 0 | 0 | 1 | 0 |
| **1st Qu.** | 133.5 | 0 | 0 | 1 | 0 | 2 | 0 |
| **Median** | 153 | 0 | 0.8 | 1 | 0 | 2 | 1 |
| **Mean** | 149.6 | 0.3267 | 1.04 | 1.399 | 0.7294 | 2.33 | 0.5446 |
| **3rd Qu.** | 166 | 1 | 1.6 | 2 | 1 | 3 | 1 |
| **Max.** | 202 | 1 | 6.2 | 2 | 4 | 3 | 1 |

En utilisant la fonction « describe » sur R on obtient les resultats suivantes :

La distribution de chaque variable dépend du type de variable.

La variable "age" est une variable numérique continue et semble suivre une distribution approximativement normale

avec une moyenne de 54,37 ans et une médiane de 55 ans.

La variable "sex" est une variable catégorielle avec deux modalités (0 et 1) et ne suit donc pas une distribution.

La variable "cp" est une variable catégorielle avec quatre modalités (0, 1, 2, 3) et ne suit donc pas une distribution.

Cependant, on peut calculer les proportions de chaque modalité.

La variable "trestbps" est une variable numérique continue et semble suivre une distribution approximativement normale

avec une moyenne de 131,6 mm Hg et une médiane de 130 mm Hg.

La variable "chol" est une variable numérique continue et semble suivre une distribution approximativement normale

avec une moyenne de 246,3 mg/dl et une médiane de 240 mg/dl.

La variable "fbs" est une variable catégorielle avec deux modalités (0 et 1) et ne suit donc pas une distribution.

La variable "restecg" est une variable catégorielle avec trois modalités (0, 1, 2) et ne suit donc pas une distribution.

Cependant, on peut calculer les proportions de chaque modalité.

La variable "thalach" est une variable numérique continue et semble suivre une distribution approximativement normale

avec une moyenne de 149,6 bpm et une médiane de 153 bpm.

La variable "exang" est une variable catégorielle avec deux modalités (0 et 1) et ne suit donc pas une distribution.

La variable "oldpeak" est une variable numérique continue et semble suivre une distribution approximativement normale

avec une moyenne de 1,04 mm et une médiane de 0,8 mm.

La variable "slope" est une variable catégorielle avec trois modalités (0, 1, 2) et ne suit donc pas une distribution.

Cependant, on peut calculer les proportions de chaque modalité.

La variable "ca" est une variable catégorielle avec cinq modalités (0, 1, 2, 3, 4) et ne suit donc pas une distribution.

Cependant, on peut calculer les proportions de chaque modalité.

Table 3: RESULTATS(1) DE LA FONCTION "describe"

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **n** | **missing** | **distinct** | **Info** | **Mean** | **Gmd** |
| age | 303 | 0 | 41 | 0.999 | 54.37 | 10.36 |
| sex | 303 | 0 | 2 | 0.649 | - | - |
| cp | 303 | 0 | 4 | 0.866 | 0.967 | 1.105 |
| trestbps | 303 | 0 | 49 | 0.995 | 131.6 | 19.32 |
| chol | 303 | 0 | 152 | 1 | 246.3 | 55.95 |
| fbs | 303 | 0 | 2 | 0.379 | - | - |
| restecg | 303 | 0 | 3 | 0.76 | 0.5281 | 0.5274 |
| thalach | 303 | 0 | 91 | 1 | 149.6 | 25.77 |
| exang | 303 | 0 | 2 | 0.66 | - | - |
| oldpeak | 303 | 0 | 40 | 0.964 | 1.04 | 1.225 |
| slope | 303 | 0 | 3 | 0.798 | 1.399 | 0.6291 |
| ca | 303 | 0 | 5 | 0.795 | 0.7294 | 1.005 |
| thal | 303 | 0 | 4 | 0.546 | - | - |

Table 4: RESULTATS(2) DE LA FONCTION "describe"

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Variable** | **0.05** | **0.1** | **0.25** | **0.5** | **0.75** | **0.9** | **0.95** |
| age | 39.1 | 42 | 47.5 | 55 | 61 | 66 | 68 |
| sex | - | - | - | - | - | - | - |
| cp | - | - | - | - | - | - | - |
| trestbps | 108 | 110 | 120 | 130 | 140 | 152 | 160 |
| chol | 175 | 188 | 211 | 240 | 274.5 | 308.8 | 326.9 |
| fbs | - | - | - | - | - | - | - |
| restecg | - | - | - | - | - | - | - |
| thalach | 108.1 | 116 | 133.5 | 153 | 166 | 176.6 | 181.9 |
| exang | - | - | - | - | - | - | - |
| oldpeak | 0 | 0 | 0 | 0.8 | 1.6 | 2.8 | 3.4 |
| slope | - | - | - | - | - | - | - |
| ca | - | - | - | - | - | - | - |
| thal | - | - | - | - | - | - | - |

En utilisant la function “describeBy” qui nous donne les resultats suivantes :

On a diviser la population en 2 partie selon la variables cible « maladie » ou « pas de maladie »

Table 5: RESULTATS(1) DE LA FONCTION "describeBy" pour grp 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| group: 0 |  |  |  |  |  |  |  |
|  | vars | n | mean | sd | median | trimmed | mad |
| age | 1 | 138 | 56.6 | 7.96 | 58 | 57.05 | 5.93 |
| sex | 2 | 138 | 0.83 | 0.38 | 1 | 0.9 | 0 |
| cp | 3 | 138 | 0.48 | 0.91 | 0 | 0.29 | 0 |
| trestbps | 4 | 138 | 134.4 | 18.73 | 130 | 132.84 | 14.83 |
| chol | 5 | 138 | 251.09 | 49.45 | 249 | 250.06 | 49.67 |
| fbs | 6 | 138 | 0.16 | 0.37 | 0 | 0.08 | 0 |
| restecg | 7 | 138 | 0.45 | 0.54 | 0 | 0.41 | 0 |
| thalach | 8 | 138 | 139.1 | 22.6 | 142 | 139.77 | 23.72 |
| exang | 9 | 138 | 0.55 | 0.5 | 1 | 0.56 | 0 |
| oldpeak | 10 | 138 | 1.59 | 1.3 | 1.4 | 1.47 | 1.48 |
| slope | 11 | 138 | 1.17 | 0.56 | 1 | 1.2 | 0 |
| ca | 12 | 138 | 1.17 | 1.04 | 1 | 1.08 | 1.48 |
| thal | 13 | 138 | 2.57 | 0.65 | 3 | 2.69 | 0 |
| target | 14 | 138 | 0 | 0 | 0 | 0 | 0 |

Table 6: RESULTATS(2) DE LA FONCTION "describeBy" pour grp 0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group: 0 |  |  |  |  |  |  |
|  | min | max | range | skew | kurtosis | se |
| age | 35 | 77 | 42 | -0.54 | 0.04 | 0.68 |
| sex | 0 | 1 | 1 | -1.7 | 0.9 | 0.03 |
| cp | 0 | 3 | 3 | 1.61 | 1.15 | 0.08 |
| trestbps | 100 | 200 | 100 | 0.84 | 0.78 | 1.59 |
| chol | 131 | 409 | 278 | 0.31 | 0.27 | 4.21 |
| fbs | 0 | 1 | 1 | 1.84 | 1.4 | 0.03 |
| restecg | 0 | 2 | 2 | 0.61 | -0.83 | 0.05 |
| thalach | 71 | 195 | 124 | -0.29 | -0.27 | 1.92 |
| exang | 0 | 1 | 1 | -0.2 | -1.97 | 0.04 |
| oldpeak | 0 | 6.2 | 6.2 | 0.73 | 0.31 | 0.11 |
| slope | 0 | 2 | 2 | 0.03 | -0.14 | 0.05 |
| ca | 0 | 4 | 4 | 0.47 | -0.83 | 0.09 |
| thal | 1 | 3 | 2 | -1.19 | 0.2 | 0.06 |
| target | 0 | 0 | 0 | NaN | NaN | 0 |

Table 7: RESULTATS(1) DE LA FONCTION "describeBy" pour grp 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| group: 1 |  |  |  |  |  |  |  |  |
|  | vars | n | mean | sd | median |  | trimmed | mad |
| age | 1 | 165 | 52.5 | 9.55 | 52 |  | 52.32 | 10.38 |
| sex | 2 | 165 | 0.56 | 0.5 | 1 |  | 0.58 | 0 |
| cp | 3 | 165 | 1.38 | 0.95 | 2 |  | 1.35 | 1.48 |
| trestbps | 4 | 165 | 129.3 | 16.17 | 130 |  | 128.72 | 14.83 |
| chol | 5 | 165 | 242.23 | 53.55 | 234 |  | 238.09 | 44.48 |
| fbs | 6 | 165 | 0.14 | 0.35 | 0 |  | 0.05 | 0 |
| restecg | 7 | 165 | 0.59 | 0.5 | 1 |  | 0.61 | 0 |
| thalach | 8 | 165 | 158.47 | 19.17 | 161 |  | 159.97 | 16.31 |
| exang | 9 | 165 | 0.14 | 0.35 | 0 |  | 0.05 | 0 |
| oldpeak | 10 | 165 | 0.58 | 0.78 | 0.2 |  | 0.45 | 0.3 |
| slope | 11 | 165 | 1.59 | 0.59 | 2 |  | 1.68 | 0 |
| ca | 12 | 165 | 0.36 | 0.85 | 0 |  | 0.14 | 0 |
| thal | 13 | 165 | 2.13 | 0.44 | 2 |  | 2.09 | 0 |
| target | 14 | 165 | 1 | 0 | 1 |  | 1 | 0 |

Table 8: RESULTATS(2) DE LA FONCTION "describeBy" pour grp 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| group: 1 |  |  |  |  |  |  |
|  | min | max | range | skew | kurtosis | se |
| age | 29 | 76 | 47 | 0.12 | -0.66 | 0.74 |
| sex | 0 | 1 | 1 | -0.25 | -1.95 | 0.04 |
| cp | 0 | 3 | 3 | -0.13 | -1.07 | 0.07 |
| trestbps | 94 | 180 | 86 | 0.42 | 0.31 | 1.26 |
| chol | 126 | 564 | 438 | 1.71 | 7.25 | 4.17 |
| fbs | 0 | 1 | 1 | 2.06 | 2.27 | 0.03 |
| restecg | 0 | 2 | 2 | -0.24 | -1.6 | 0.04 |
| thalach | 96 | 202 | 106 | -0.69 | 0.37 | 1.49 |
| exang | 0 | 1 | 1 | 2.06 | 2.27 | 0.03 |
| oldpeak | 0 | 4.2 | 4.2 | 1.62 | 2.98 | 0.06 |
| slope | 0 | 2 | 2 | -1.15 | 0.27 | 0.05 |
| ca | 0 | 4 | 4 | 2.75 | 7.51 | 0.07 |
| thal | 1 | 3 | 2 | 0.67 | 1.35 | 0.03 |
| target | 1 | 1 | 0 | NaN | NaN | 0 |

Ces résultats représentent des statistiques descriptives pour deux groupes différents

(group 0 et group 1) sur un ensemble de 14 variables:

age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal et target.

Chaque ligne représente une variable, et chaque colonne représente une mesure statistique pour le groupe correspondant.

Les mesures statistiques incluent le nombre d'observations (n), la moyenne (mean), l'écart-type (sd), la médiane,

la moyenne tronquée (trimmed), la déviation absolue de la médiane (mad), la valeur minimale (min),

la valeur maximale (max), l'étendue (range), l'asymétrie (skew), le kurtosis (kurtosis)

et l'erreur standard de la moyenne (se).

Par exemple, nous pouvons voir que pour le groupe 0, la moyenne d'âge est de 56,6 ans, l'écart-type est de 7,96 ans, la valeur minimale est de 35 ans et la valeur maximale est de 77 ans. De même, pour le groupe 1, la moyenne d'âge est de 52,5 ans, l'écart-type est de 9,55 ans, la valeur minimale est de 29 ans et la valeur maximale est de 76 ans.

On peut remarquer que les moyennes et les écarts types varient selon les groupes.

Par exemple, les personnes du groupe 0 ont une tension artérielle moyenne plus élevée que les personnes du groupe 1,

tandis que les personnes du groupe 1 ont un taux de cholestérol moyen plus élevé que les personnes du groupe 0.

Les résultats de la cible indiquent que toutes les personnes du groupe 0 sont non atteintes de maladies cardiaques,

alors que l'on ne peut rien déduire pour le groupe 1.

Le skewness (ou asymétrie) est une mesure statistique qui permet d'évaluer l'asymétrie d'une distribution.

Si la distribution est symétrique, le skewness sera égal à zéro.

Si la queue de la distribution est étendue vers la droite (positivement), le skewness sera positif,alors que s'il est étendu vers la gauche (négativement), le skewness sera négatif. Le skewness peut être utilisé pour identifier les distributions qui ne sont pas normales.

La kurtosis (ou aplatissement) est une mesure statistique qui permet d'évaluer l'aplatissement d'une distribution par rapport à une distribution normale.

Si la distribution est normale, la kurtosis sera égale à zéro.

Si la distribution est plus plate que la normale, la kurtosis sera négative, **(platykurtiqie)**

alors que si elle est plus pointue que la normale, la kurtosis sera positive. **(leptokutrique)**

La kurtosis peut être utilisée pour identifier les distributions qui ont des queues plus épaisses que la normale.

En utilisant les informations de skewness et kurtosis fournies dans les statistiques descriptives par groupe,

on peut dire ce qui suit sur la distribution de chaque variable pour chaque groupe

Pour le groupe 0 :

age : légèrement incliné vers la gauche (skewness = -0.54) et légèrement aplati (kurtosis = 0.04).

sex : fortement incliné vers la gauche (skewness = -1.70) et fortement aplati (kurtosis = 0.90).

cp : légèrement incliné vers la droite (skewness = 1.61) et modérément aplati (kurtosis = 1.15).

trestbps : légèrement incliné vers la droite (skewness = 0.84) et légèrement aplati (kurtosis = 0.78).

chol : légèrement incliné vers la droite (skewness = 0.31) et légèrement aplati (kurtosis = 0.27).

fbs : fortement incliné vers la droite (skewness = 1.84) et fortement aplati (kurtosis = 1.40).

restecg : légèrement incliné vers la droite (skewness = 0.61) et légèrement aplati (kurtosis = -0.83).

thalach : légèrement incliné vers la gauche (skewness = -0.29) et légèrement aplati (kurtosis = -0.27).

exang : légèrement incliné vers la gauche (skewness = -0.20) et fortement aplati (kurtosis = -1.97).

oldpeak : légèrement incliné vers la droite (skewness = 0.73) et légèrement aplati (kurtosis = 0.31).

slope : légèrement incliné vers la droite (skewness = 0.03) et légèrement aplati (kurtosis = -0.14).

ca : légèrement incliné vers la droite (skewness = 0.47) et légèrement aplati (kurtosis = -0.83).

thal : fortement incliné vers la gauche (skewness = -1.19) et modérément aplati (kurtosis = 0.20).

target : la variable est une variable catégorielle et n'a pas de mesure de skewness ou de kurtosis.

Pour le groupe 1 :

age : skewness = 0.12, kurtosis = -0.66, légèrement asymétrique négative et légèrement aplatie.

sex : skewness = -0.25, kurtosis = -1.95, légèrement asymétrique négative et aplatie.

cp : skewness = -0.13, kurtosis = -1.07, légèrement asymétrique négative et aplatie.

trestbps : skewness = 0.42, kurtosis = 0.31, légèrement asymétrique positive et légèrement aplatie.

chol : skewness = 1.71, kurtosis = 7.25, très asymétrique positive et très leptokurtique.

fbs : skewness = 2.06, kurtosis = 2.27, très asymétrique positive et légèrement leptokurtique.

restecg : skewness = -0.24, kurtosis = -1.60, légèrement asymétrique négative et aplatie.

thalach : skewness = -0.69, kurtosis = 0.37, légèrement asymétrique négative et légèrement aplatie.

exang : skewness = 2.06, kurtosis = 2.27, très asymétrique positive et légèrement leptokurtique.

oldpeak : skewness = 1.62, kurtosis = 2.98, très asymétrique positive et légèrement leptokurtique.

slope : skewness = -1.15, kurtosis = 0.27, légèrement asymétrique négative et légèrement aplatie.

ca : skewness = 2.75, kurtosis = 7.51, très asymétrique positive et très leptokurtique.

thal : skewness = 0.67, kurtosis = 1.35, légèrement asymétrique positive et légèrement leptokurtique.

target : étant une variable dichotomique,

# **Représentation graphiques de la base de données :**

Puisqu’il n’existe pas des valeurs manquantes, on va faire l’analyse statistique tout suite. Alors on va voir des différentes figures :

1. Un **"diagramme à barres"** ou **"histogramme”** pour montrer la fréquence ou la proportion de chaque catégorie. Chaque barre représente une catégorie et sa hauteur (ou sa longueur) indique la quantité de données correspondante.
2. La **"densité"** et la **"distribution”** de chaque variable.

POUR LA VARIABLE « AGE » :

On peut voir la répartition et les densités des différentes Age de la population traites

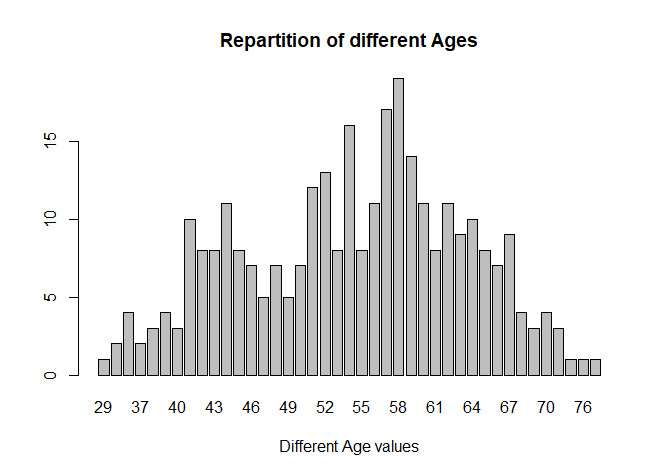


Figure 1: La repartition de l'Age

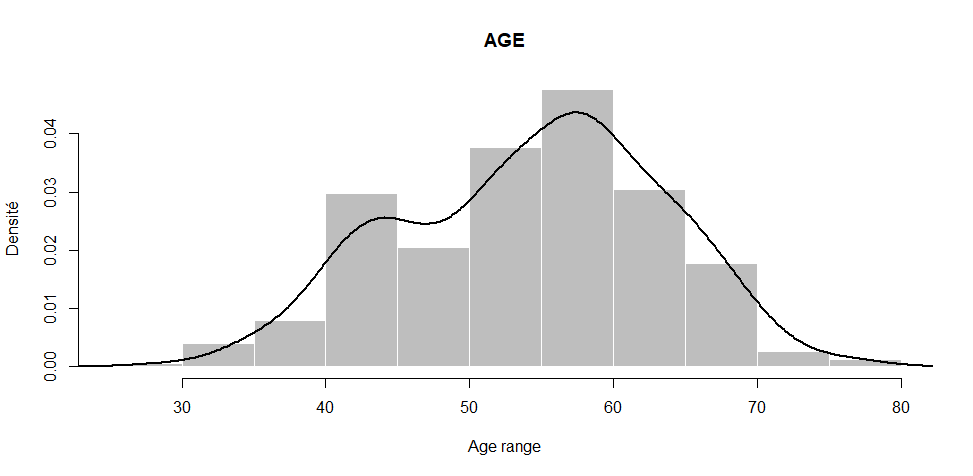


Figure 2: La densite de l'Age

On constate que l’age a une distribution normal comme nous avons vu dans « skewness » et « kurtosis », de plus on constate qu’il nexiste pas des valeurs aberrantes .

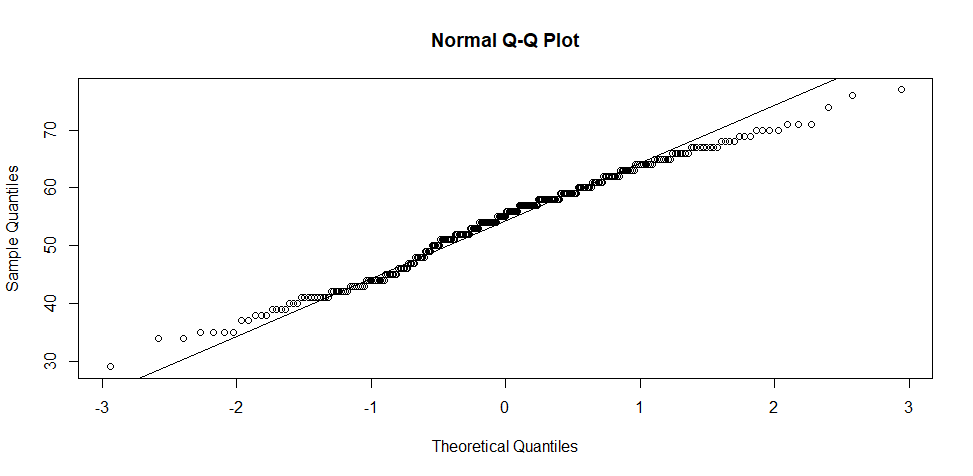


Figure 3: La distribution de l'Age

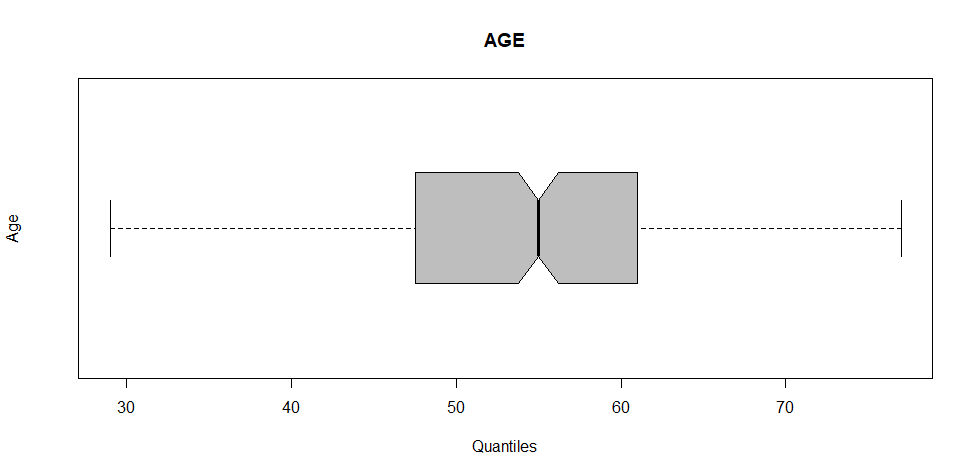


Figure 4: Boite a moustache de l'Age

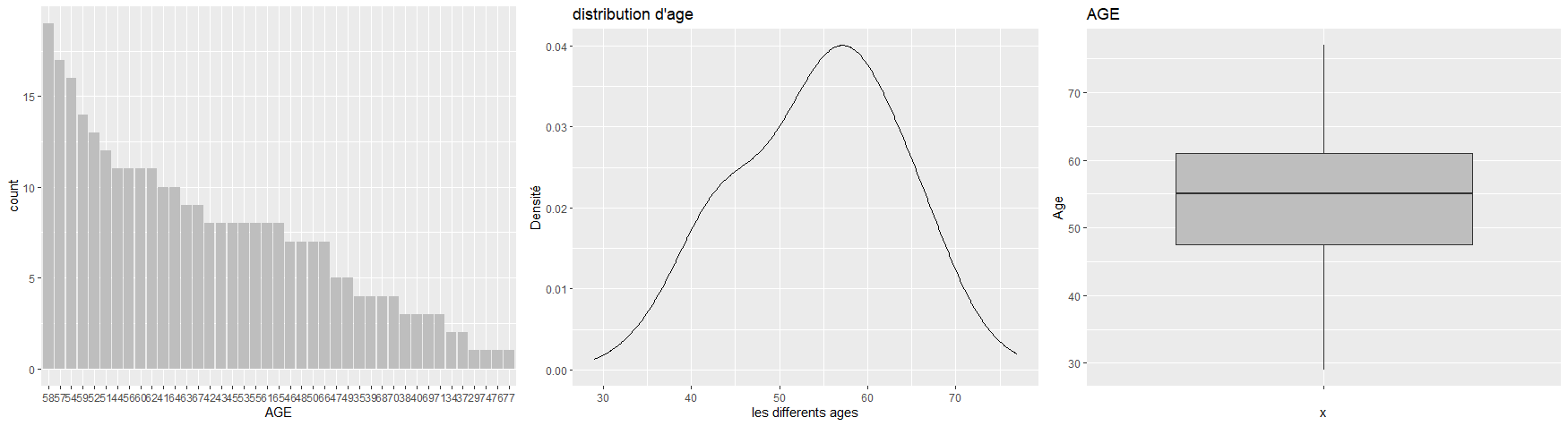


Figure 5: Les différentes figures de la variable "Age"

POUR LA VARIABLE “sex”:

On peut visualiser que la majorité de la population traites sont des hommes (gendre=0),et que la distribution n’est pas normal puisque c’est une variable catégorielle .

On constate que l’age est relie au sex, mais la distribution de la variable sex n’est pas comme celle de l’age.

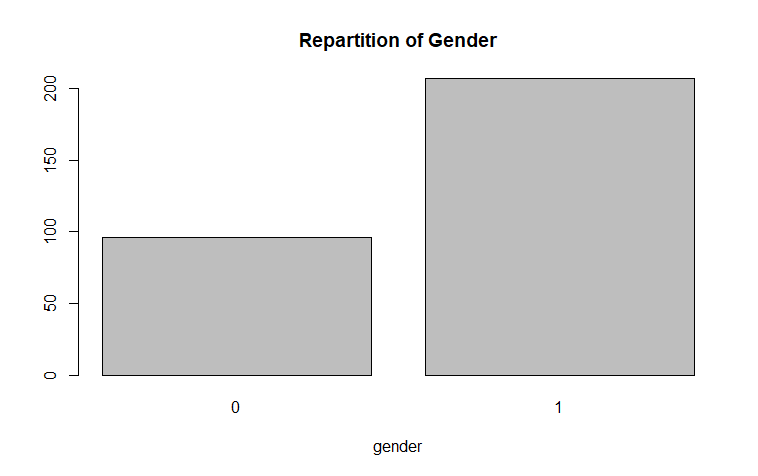


Figure 6: La repartition de la variable "sex"

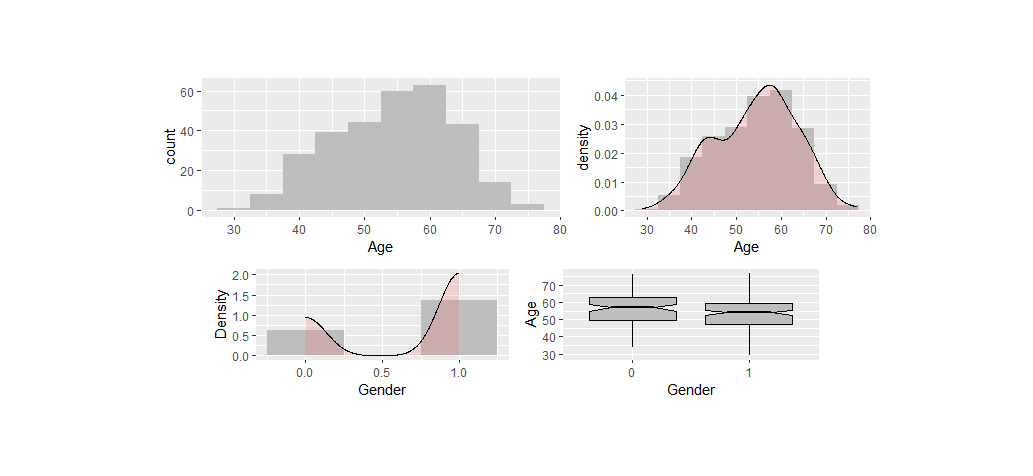


Figure 7: Les differentes figurent de la variable "sex"

POUR LA VARIABLE “trestbps”:

On peut voir la répartition telle que la valeur 74 correspond à l’effectif

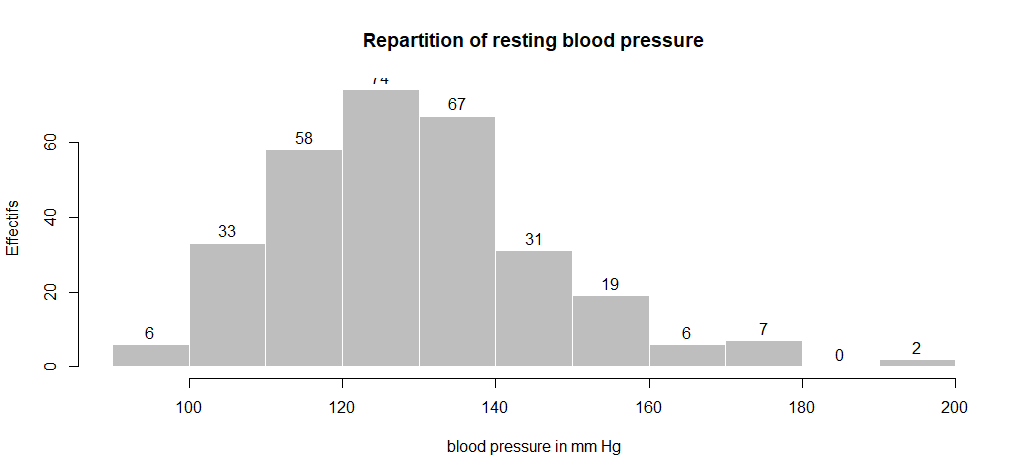


Figure 8: La répartition de la variable "tresbps"

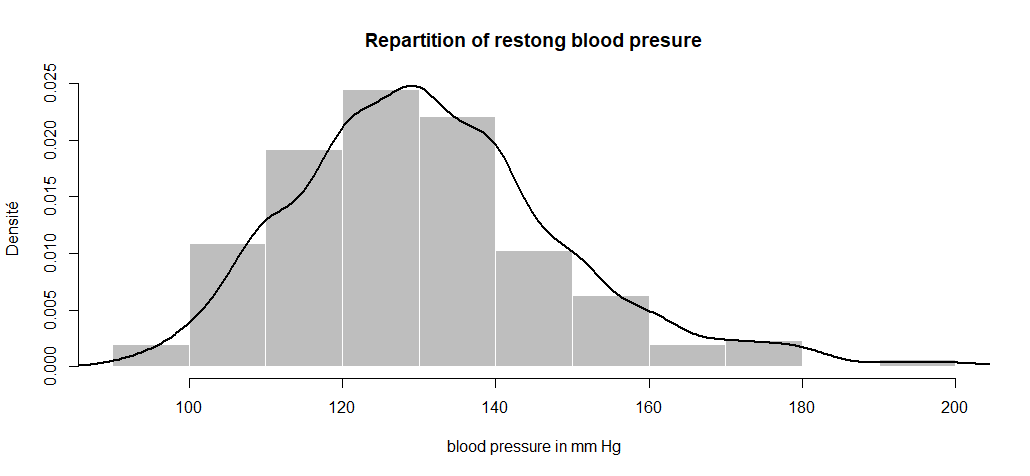


Figure 9: La distribution de la variable « trestbps »

On constate que la variabe a une distribution normal ecartee un peu a droite comme nous avons vu dans « skewness »

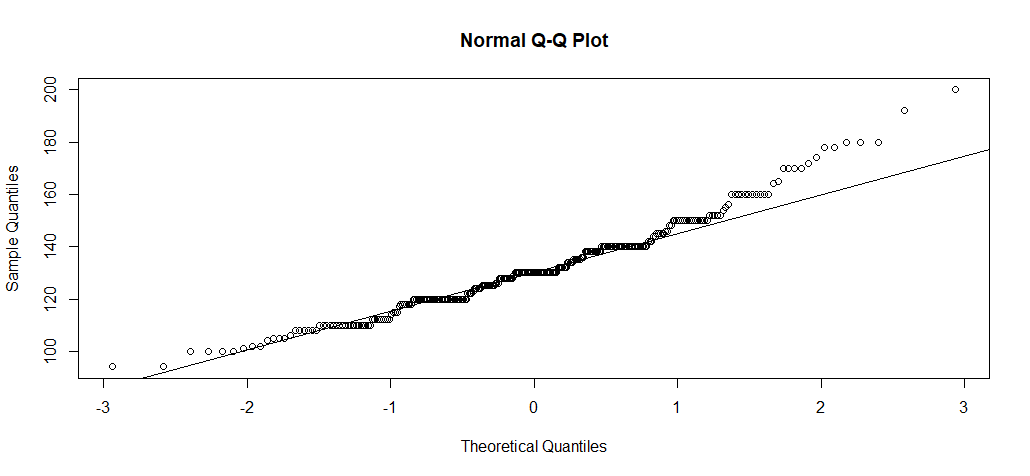


Figure 10: La distribution de la variable "tretbps"

On constate qu’il existe quelques valeurs aberantes qui depasse le 160 mm Hg

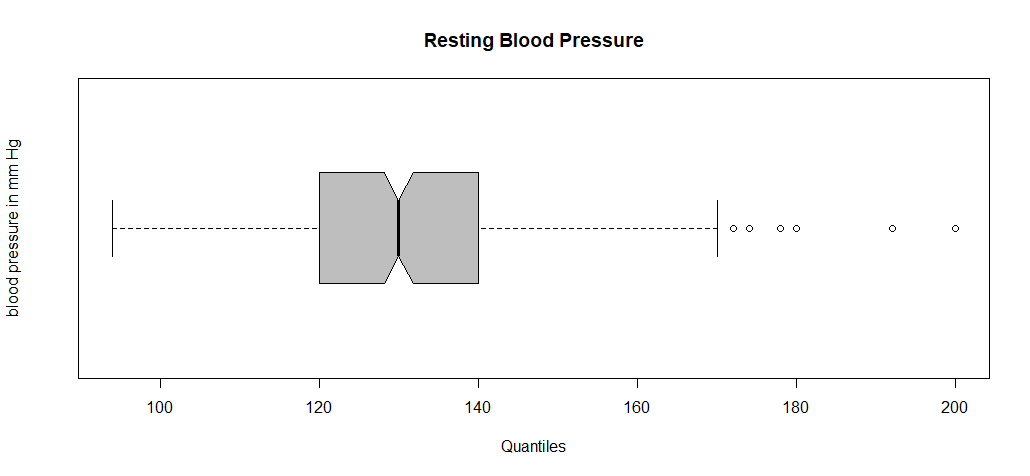


Figure 11: Boite a Moustache de la variable "trestbps"

POUR LA VARIABLE “CHOL”:

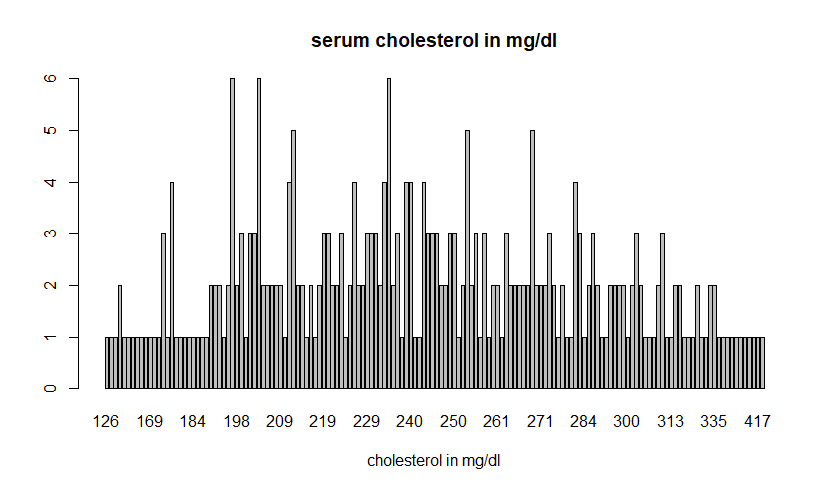


Figure 12: La repartition de la variable "chol"

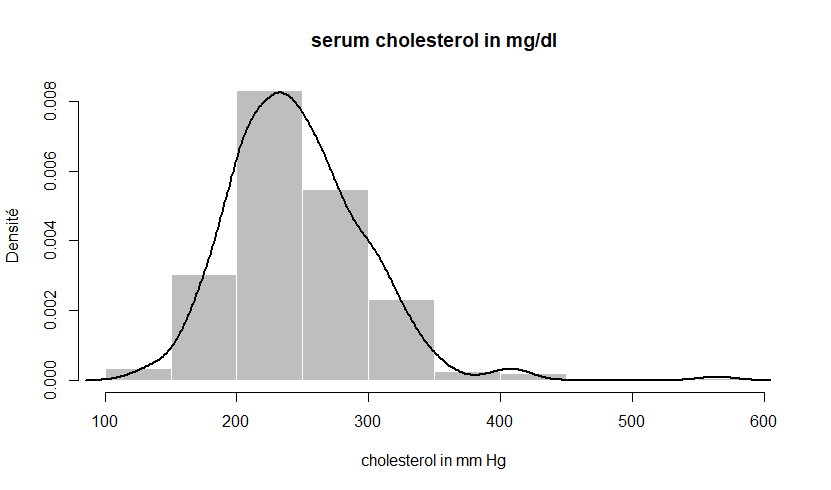


Figure 13: La densite de la variable "chol"

On constate que la distribution de la variable est a peu pres normal comme nous avons vu dans « skewness » et « kurtosis ».

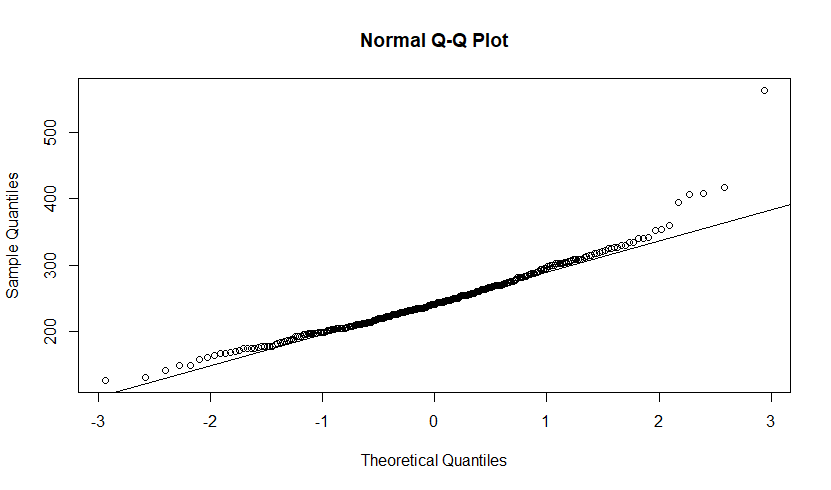


Figure 14: La distribution de la variable "chol"

On constate qu’il existe des valeurs aberrantes qui depassent le 400 mgLdl

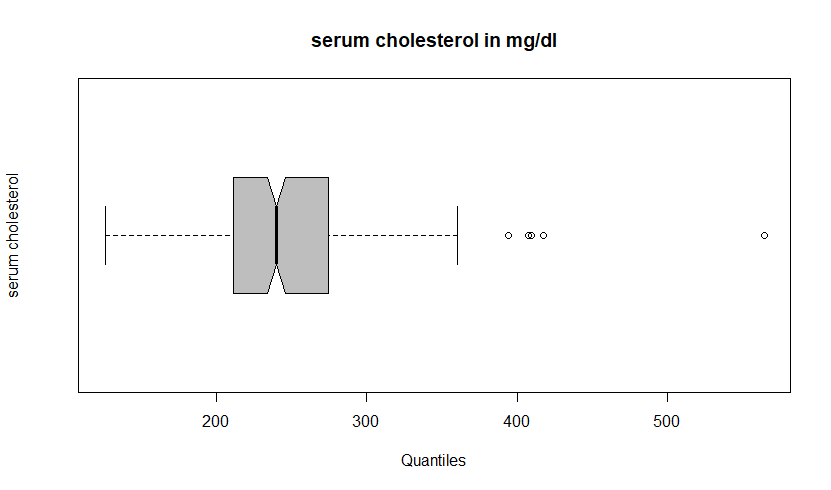


Figure 15: Boite a moustache de la variable "chol"

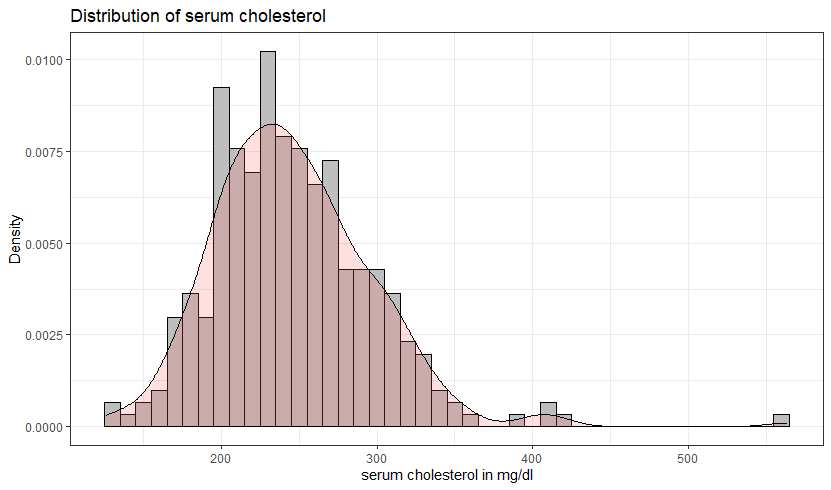


Figure 16: figure montrant la distribution et la repartition de la variable "chol"

POUR LA VARIABLE “CP”:

On visualise que le « cp » qui correspond à l’angine typique a la plus grande répartition alors que celui de l’asymptotique a la plus petite répartition.

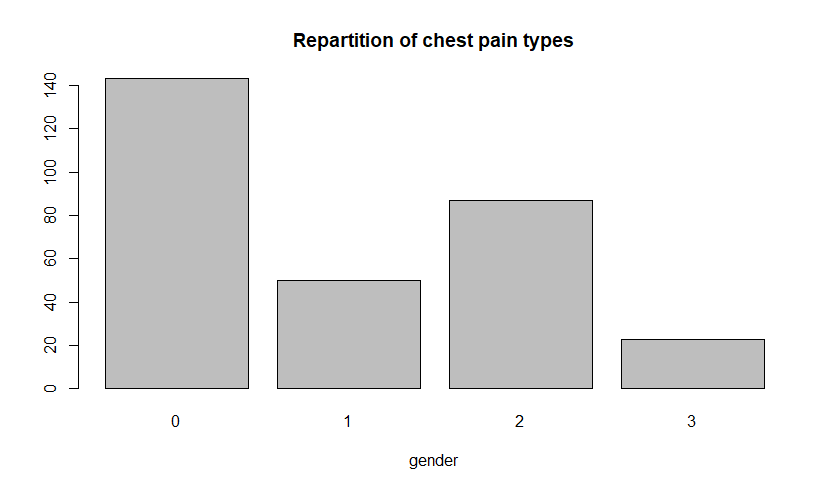


Figure 17: la répartition de la variable "cp"

On constate qu’il n’existe pas des valeurs aberrantes pour la variables « cp »

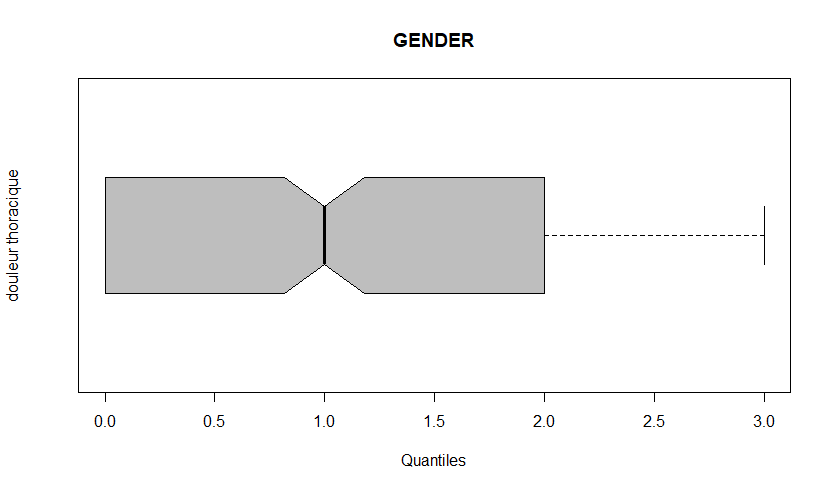


Figure 18: Boite a moustache de la variable "cp"

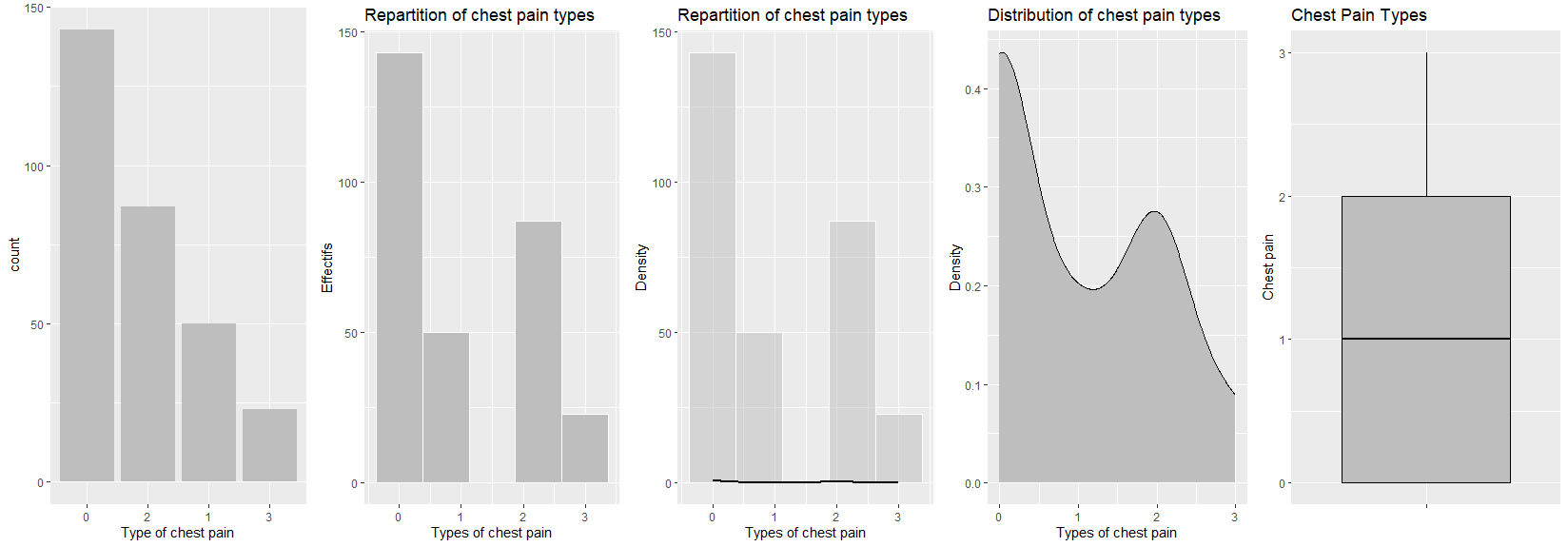


Figure 19: Les différentes figures de la variable « cp »

POUR LA VARIABLE “FBS”:

La repartition de la variable “FBS » nous montre que la plus part de la population n’ont pas la glycemine>120 mg/dl.

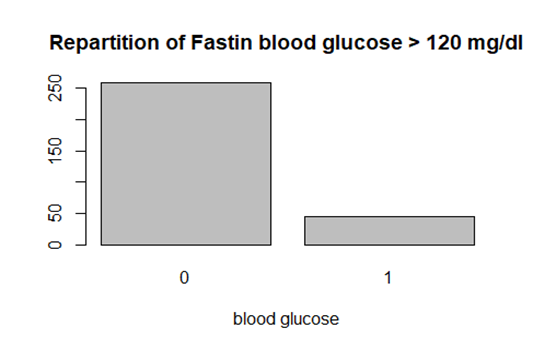


Figure 20: La répartition de la variable "fbs"

FOR THE VARIABLE “THALACH”:

On peut visualiser la répartition et la densité des fréquences cardiaques maximales atteinte de chaque observation.

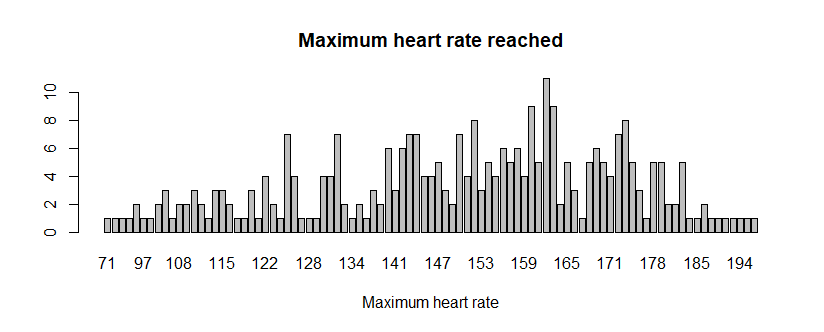


Figure 21: La repartition denla variable "thalach"

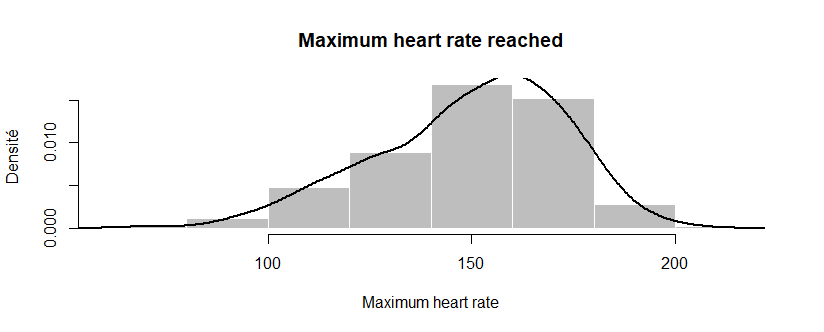


Figure 22: La densite de la variable "thalach"

Comme nous avons vu dans « kurtosis » et « skewness » que la distribution est normale.

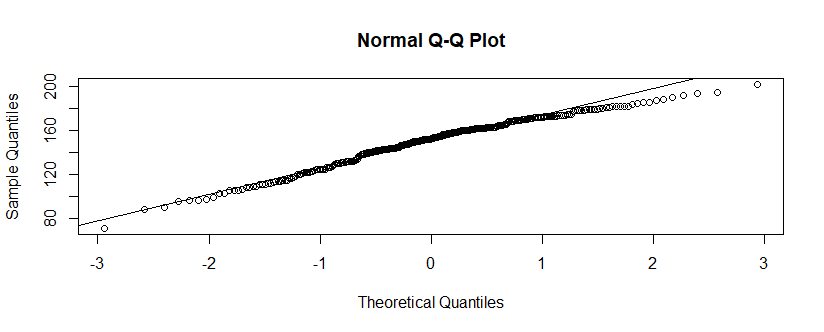


Figure 23: La distribution de la variable "thalach"

Om remarque qu’il n’existe qu’un seul point aberrant.

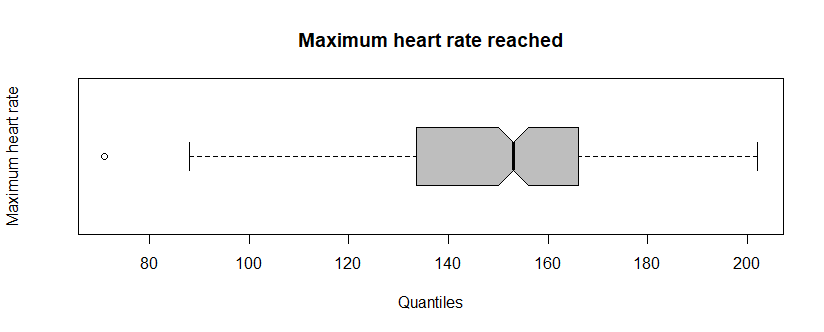


Figure 24: Boite a moustache de la variable "thalach"

POUR LA VARIALE “EXANG”:

On remarque que la plus part de la population n’ont pas la poitrine induite par l’effort.

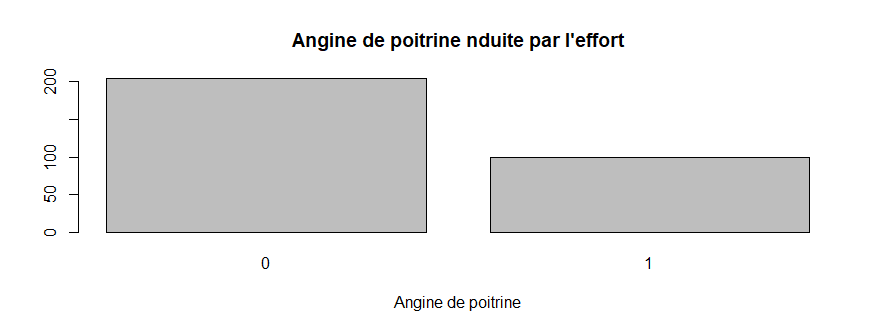


Figure 25: La répartition de la poitrine de la variable "exang"

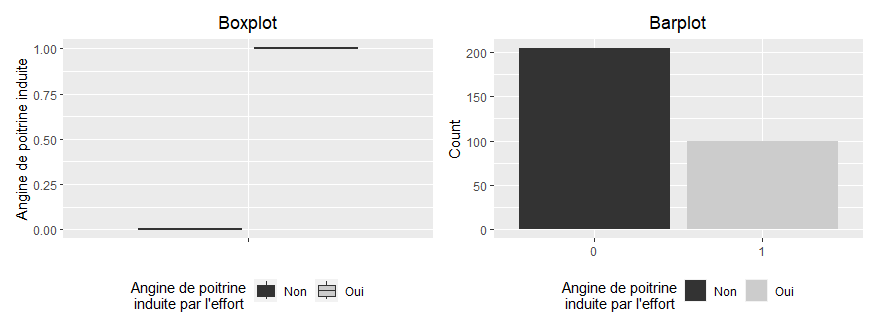


Figure 26 : la répartition de la variable catégorielle "exang"

POUR LA VARIABLE “RESTECG”:

On constate que la plus part des individus ayant un electrographique au repos normal ou ayant ST-T, et la minorités ayant un hypertrophie.

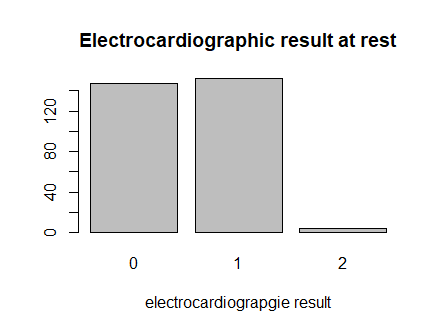
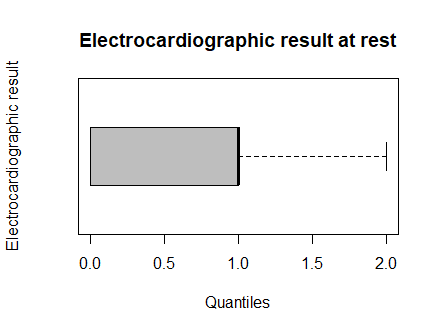


Figure 27: La répartition du résultats electrographique au repos

On constate qu’il n’existe aucun point aberrantes puisque c’est une variable catégorielle factorielle.



POUR LA VARIABLE “SLOPE”:

On constate que la plus part de la population ont une pente de segmentation plat ou descendant, alors que la minorités ont une segmentations montant.

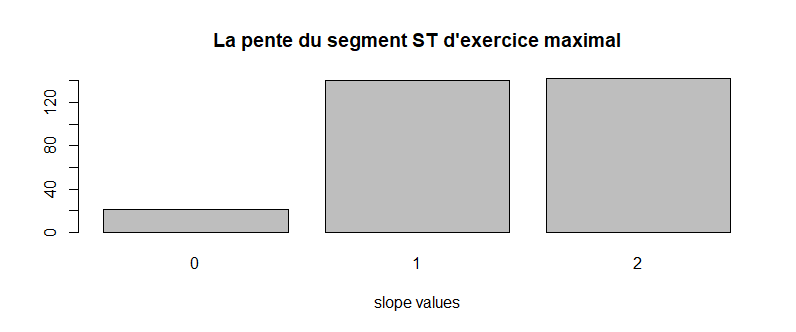


Figure 28: La répartition de la variable "slope"

On constate qu’il n’existe aucuns points aberrants puisque c’est une variable catégorielle.

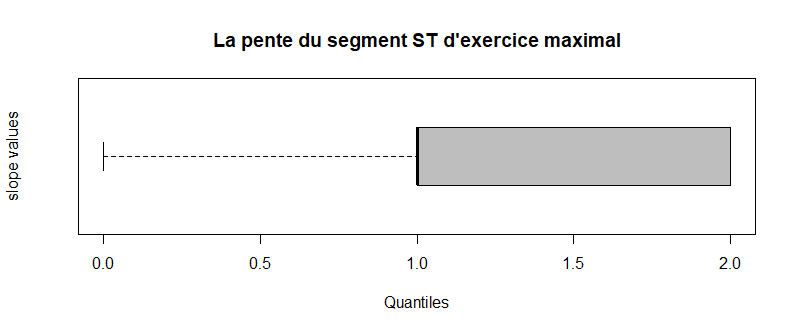


Figure 29: Boite a moustache de la variable "slope"

POUR LA VARIABLE “OLDPEAK”:

On Remarque que la plus part de la population on une dépression zero.

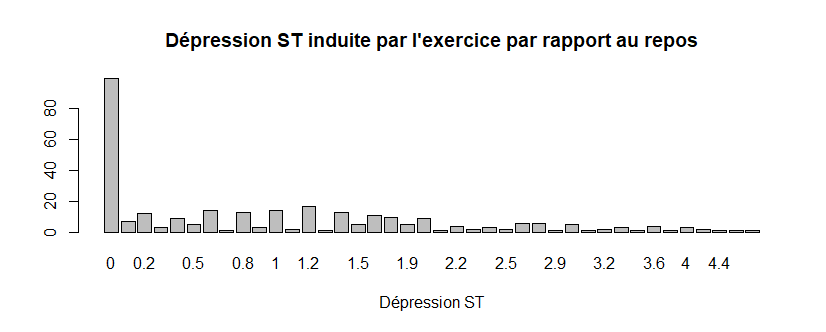


Figure 30: La répartition de la dépression ST induite par l'exercice par rapport au repos

Comme nous avons vu que la distribution de la variable « oldpeak » n’est pas normale, plutôt c’est écarte a droite.

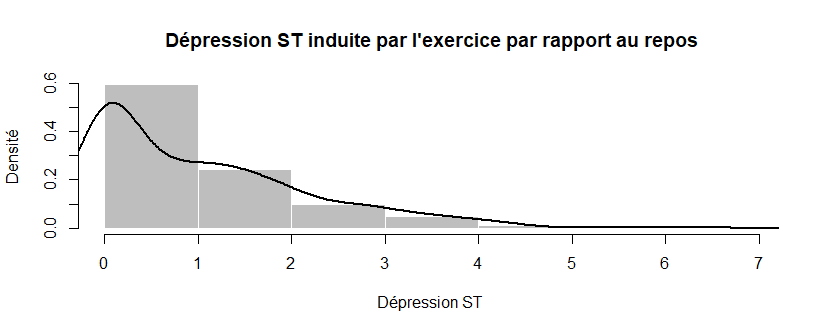


Figure 31: La densité de la variable "oldpeak"

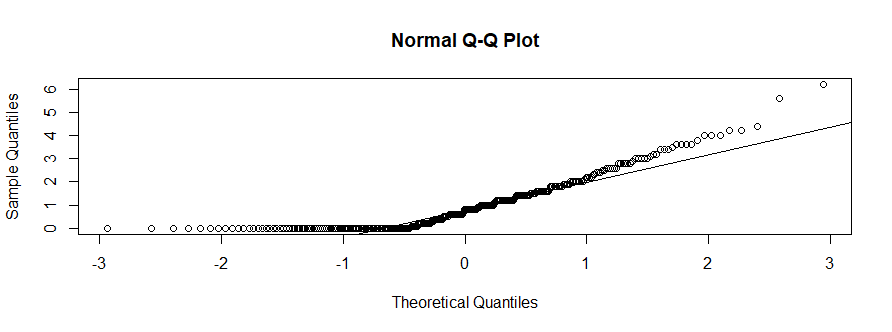


Figure 32: La distribution de la variable "oldpeak"

On constate que cette variable a 4 points aberrantes.

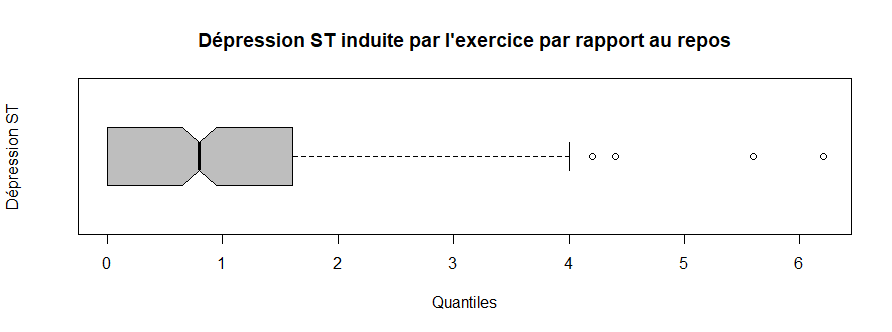


Figure 33: Boite de moustache de la variable "oldpeak"

POUR LA VARIABLE “CA”:

On constate que la plus part de la population ont les vaisseaux principaux de couleur 0 par fluoroscopie.

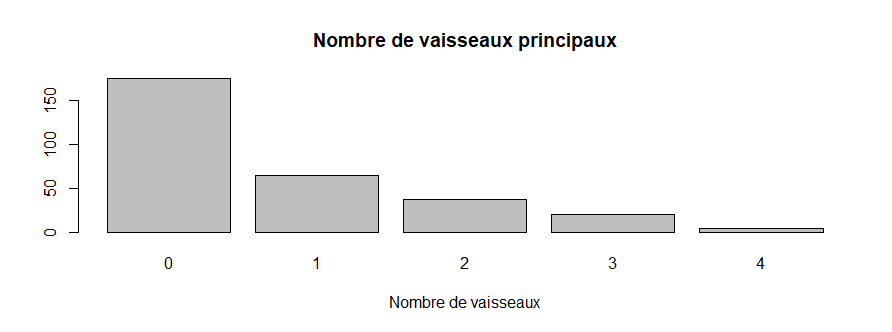


Figure 34: La répartition de la variable "CA"

On constate qu’il existe deux points aberrants.

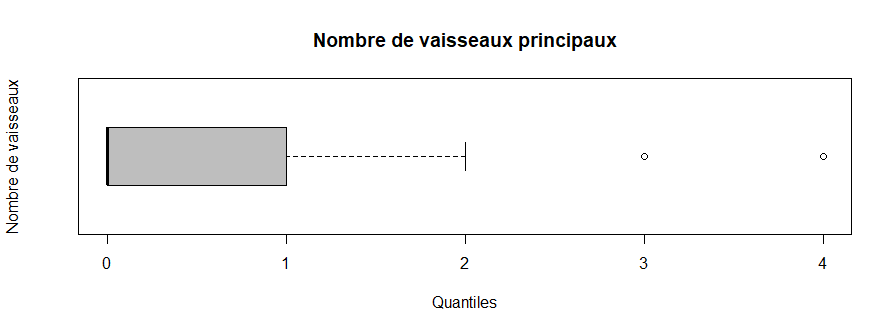


Figure 35: Boite a moustache de la variable "CA"

POUR LA VARIABLE “THAL”:

On constate que la plus part de la population on tune répartition default corrige et default réversible, lors que les minorités ont la thalassémie normale. De plus on constate qu’il n’existe aucuns points aberrants.

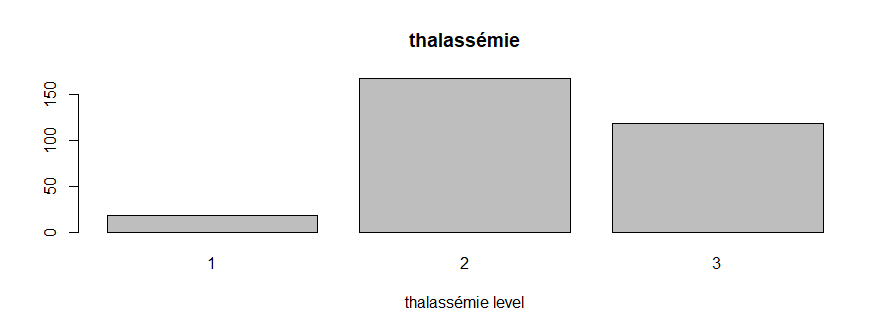


Figure 36: La répartition de la variable "thal"

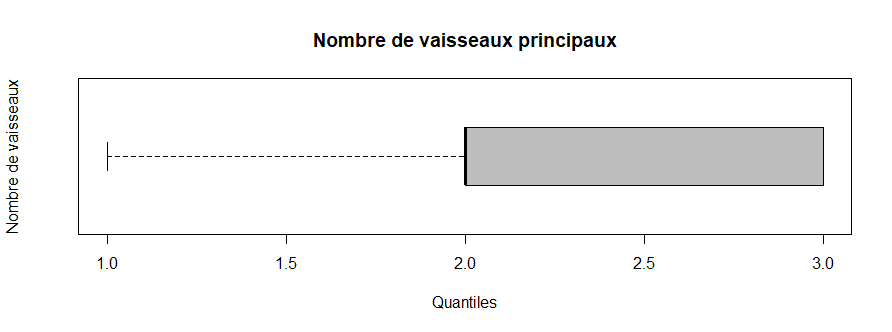


Figure 37: Boite a moustache de la variable "thal"

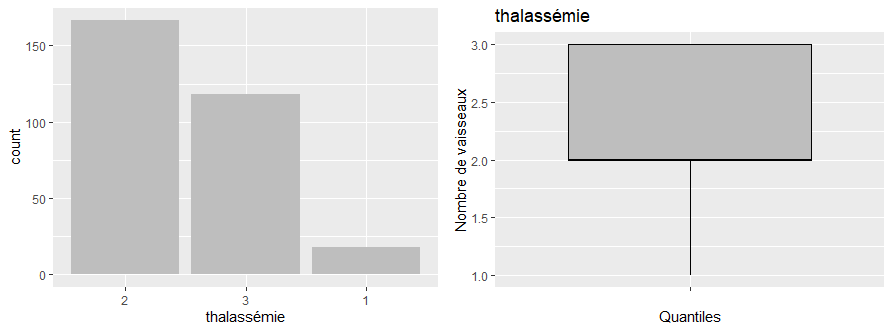


Figure 38: les différentes représentations de la variable "thal'

# **Application la méthode sur la base de données**

modèle 1 : « ARBRE DE DECISION » 🡪 cas1 : sans élagage ensuite meilleur CP

1. Tout d'abord, en utilisant un arbre de décision sans élagage (c'est-à-dire avec un paramètre de complexité nul), nous obtenons un arbre complet qui a été entraîné sur les données d'entraînement.
2. Ensuite, en utilisant la fonction printcp, nous avons pu sélectionner la meilleure valeur de cp (paramètre de complexité) pour élaguer l'arbre. Cette valeur est utilisée pour entraîner un nouvel arbre (dendrp\_cas1) en utilisant la fonction prune.
3. L'arbre élagué a une profondeur plus faible que l'arbre complet, ce qui signifie qu'il est plus simple et plus facile à interpréter.
4. Les variables les plus importantes pour la classification sont "ca", "thal" et "cp".
5. En utilisant l'ensemble de test, nous avons obtenu une **erreur de prédiction de 22,77%** pour le modèle d'arbre de décision.
6. Nous avons également tracé la courbe ROC et calculé l'AUC pour évaluer la performance du modèle. **L'AUC est de 0,812**, ce qui indique que le modèle est raisonnablement précis.

Table 9: Matrice de confusion du modèle 1-cas 1

Table 10: Tableau d'évaluation du modèle 1-cas 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Value** |  |  |  |  |
| Accuracy | 0.7667 |  |  |  |  |
| 95% CI | (0.6657, 0.8494) |  |  |  |  |
| No Information Rate | 0.5444 |  |  | **Reference 0** | **Reference 1** |
| P-Value [Acc > NIR] | 1.06E-05 |  | **Prediction 0** | 29 | 9 |
| Kappa | 0.5268 |  | **Prediction 1** | 12 | 40 |
| Mcnemar's Test P-Value | 0.6625 |  |  |  |  |
| Sensitivity | 0.8163 |  |  |  |  |
| Specificity | 0.7073 |  |  |  |  |
| Pos Pred Value | 0.7692 |  |  |  |  |
| Neg Pred Value | 0.7632 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.4444 |  |  |  |  |
| Detection Prevalence | 0.5778 |  |  |  |  |
| Balanced Accuracy | 0.7618 |  |  |  |  |

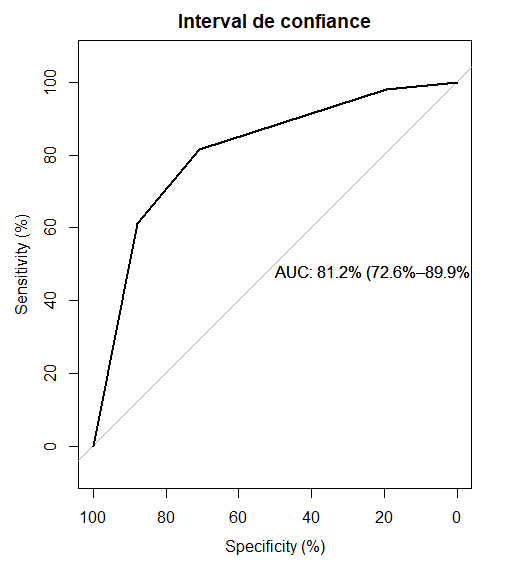
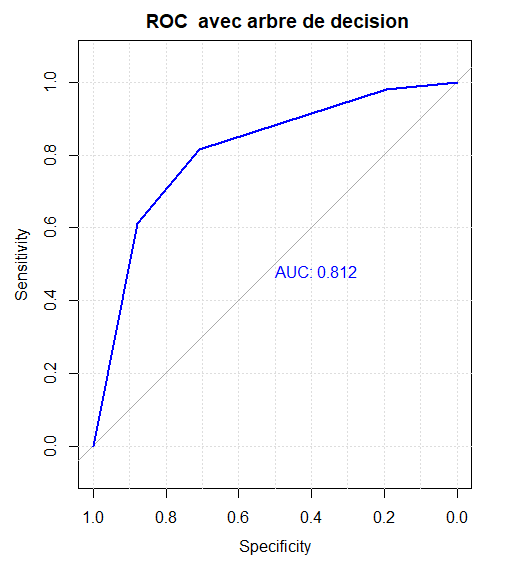


Figure 39: Courbe de Roc du modèle 1- cas 1

Figure 40: Intervalle de Confiance de Roc du modèle 1- cas 1

Modele 1 : « ARBRE DE DECISION » 🡪 cas2 : avec élagage pour CP=0.3

À partir des résultats que nous avons obtenus, nous pouvons conclure que notre modèle de classification par arbre de décision avec élagage a donné **une précision de 84%**. Nous avons choisi la meilleure valeur de CP à 0.01763. L'arbre final à une profondeur de 8. Le paramètre minsplit a été défini à 5.

1. Nous pouvons également observer que les variables les plus importantes sont "ca", "thal" et "thalach", ce qui suggère qu'elles sont les plus pertinentes pour la prédiction de la présence d'une maladie cardiaque.
2. Nous pouvons visualiser l'arbre de décision final avec la méthode "fancyRpartPlot" qui nous montre la structure de l'arbre et la hiérarchie des variables.
3. En utilisant la courbe ROC et l'aire sous la courbe (AUC), nous avons constaté que notre modèle avait une **AUC de 0,674**, ce qui suggère qu'il est assez efficace pour prédire la présence ou l'absence de maladie cardiaque.
4. De plus, nous avons également tracé l'intervalle de confiance de l'aire sous la courbe pour nous assurer que nos résultats étaient significatifs et fiables.

En fin de compte, il semble que notre modèle de classification par arbre de décision avec élagage ait donné de bons résultats pour prédire la présence ou l'absence de maladie cardiaque.

Table 11: Matrice de confusion du modèle 1-cas 2

Table 12: Tableau d'évaluation du modèle 1-cas 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Value** |  |  |  |  |
| Accuracy | 0.6889 |  |  |  |  |
| 95% CI | (0.5826, 0.7823) |  |  | **Reference 0** | **Reference 1** |
| No Information Rate | 0.5444 |  | **Prediction 0** | 21 | 8 |
| P-Value [Acc > NIR] | 0.003671 |  | **Prediction 1** | 20 | 41 |
| Kappa | 0.3575 |  |  |  |  |
| Mcnemar's Test P-Value | 0.037635 |  |  |  |  |
| Sensitivity | 0.8367 |  |  |  |  |
| Specificity | 0.5122 |  |  |  |  |
| Pos Pred Value | 0.6721 |  |  |  |  |
| Neg Pred Value | 0.7241 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.4556 |  |  |  |  |
| Detection Prevalence | 0.6778 |  |  |  |  |
| Balanced Accuracy | 0.6745 |  |  |  |  |

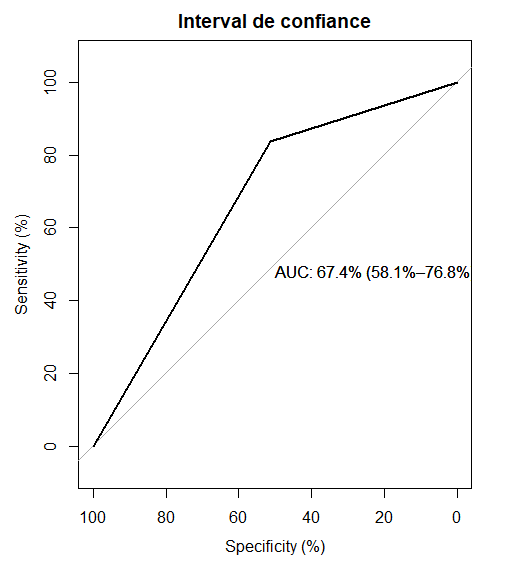
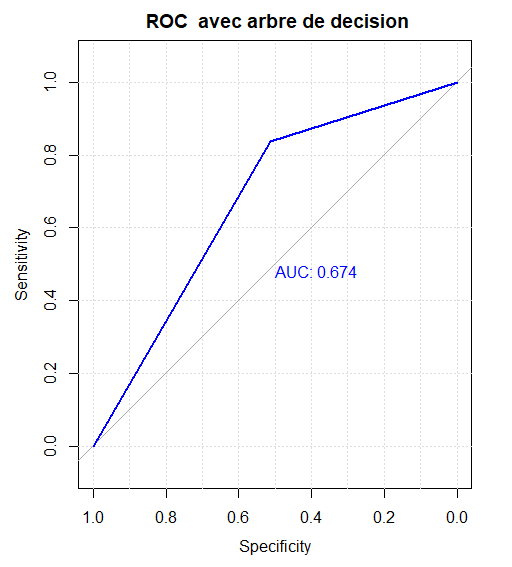


Figure 41: Courbe de Roc du modèle 1- cas 2

Figure 42: Intervalle de Confiance de Roc du modèle 1- cas 2

Modele 1 : « ARBRE DE DECISION » 🡪 cas3 : avec Cross Validation pour K=10

Voici une analyse des résultats du modèle d'arbre de décision avec validation croisée 10 fois, en utilisant la méthode "CV":

1. Le modèle a été créé en utilisant la méthode d'arbre de décision "rpart".
2. La validation croisée a choisi 3 différentes valeurs de "cp" (complexité du modèle) pour l'arbre de décision.
3. Les variables les plus importantes selon la méthode de régression ont été déterminées en utilisant la fonction "varImp".
4. La boîte à moustaches (boxplot) des variables importantes montre que certaines variables ont un impact plus important que d'autres sur la variable cible.
5. La visualisation du dendrogramme montre comment l'arbre de décision divise les données en sous-groupes.
6. Le taux d'erreur de prédiction du modèle est **de 23%,** ce qui signifie que le modèle prédit 6) correctement la variable cible pour environ 77% des observations.
7. La courbe ROC et l'aire sous la courbe (AUC) ont été utilisées pour évaluer la performance du modèle. **L'AUC est de 0,812**, ce qui indique une performance modérée du modèle.
8. Le seuil optimal a été déterminé à l'aide de la courbe ROC et de l'intervalle de confiance.
9. La matrice de confusion montre que le modèle a correctement prédit la classe "1" (cas positif) pour 68% des observations et la classe "0" (cas négatif) pour 83% des observations.
10. Le modèle a également donné une sensibilité de 68%, ce qui signifie que le modèle a correctement identifié 68% des cas positifs, et une spécificité de 83%, ce qui signifie que le modèle a correctement identifié 83% des cas négatifs.
11. Le test de « McNemar » a montré que le modèle n'a pas de différence significative entre les erreurs de type I et de type II.

En résumé, le modèle d'arbre de décision avec validation croisée a une performance modérée pour prédire la variable cible. Cependant, il peut être utile pour identifier les variables les plus importantes pour la prédiction de la variable cible et pour diviser les données en sous-groupes

Table 13: Matrice de confusion de la modèle 1-cas 3

Table 14: Tableau d'évaluation du modèle 1-cas 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Accuracy** | 0.7667 |  |  |  |  |
| **95% CI** | (0.6657, 0.8494) |  |  |  |  |
| **No Information Rate** | 0.5444 |  |  | **Reference 0** | **Reference 1** |
| **P-Value [Acc > NIR]** | 1.06E-05 |  | **Prediction 0** | 29 | 9 |
| **Kappa** | 0.5268 |  | **Prediction 1** | 12 | 40 |
| **Mcnemar's Test P-Value** | 0.6625 |  |  |  |  |
| **Sensitivity** | 0.8163 |  |  |  |  |
| **Specificity** | 0.7073 |  |  |  |  |
| **Pos Pred Value** | 0.7692 |  |  |  |  |
| **Neg Pred Value** | 0.7632 |  |  |  |  |
| **Prevalence** | 0.5444 |  |  |  |  |
| **Detection Rate** | 0.4444 |  |  |  |  |
| **Detection Prevalence** | 0.5778 |  |  |  |  |

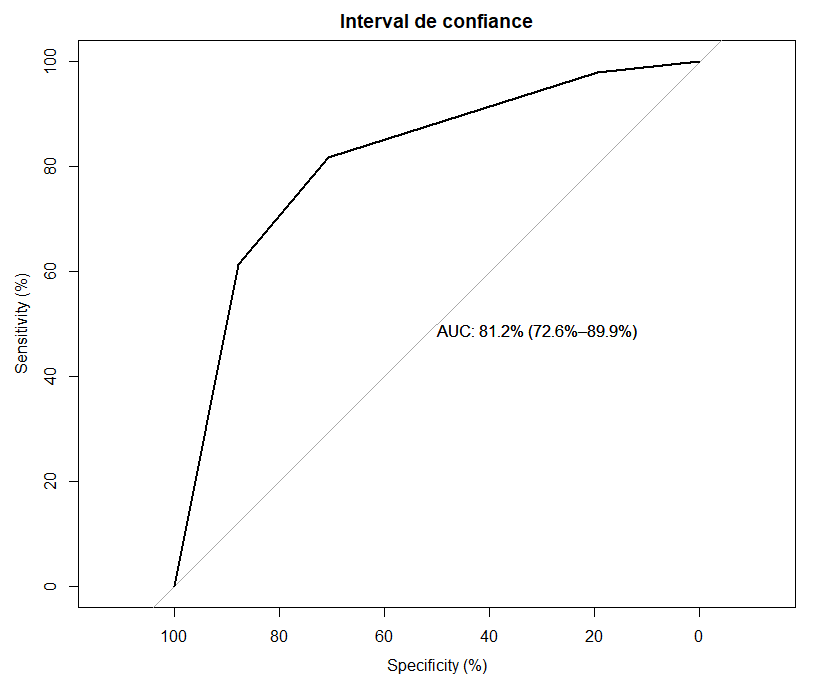
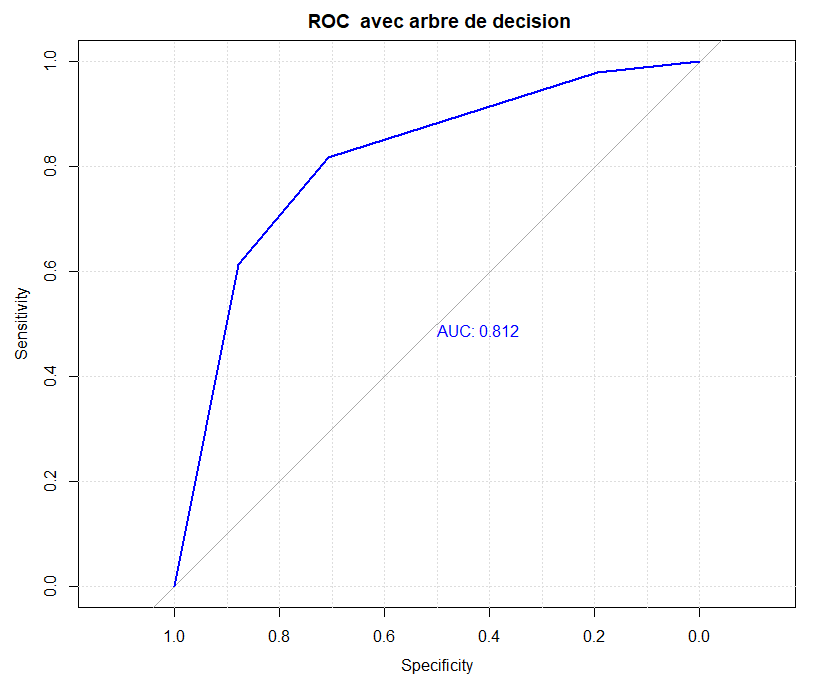


Figure 43: Courbe de Roc du modèle 1- cas 3

Figure 44: Intervalle de Confiance de Roc du modèle 1- cas 3

Modele 1 : « ARBRE DE DECISION » 🡪 cas4 : avec « OOB » méthode

Le code fourni entraîne un modèle d'arbre de décision avec une validation croisée en utilisant la méthode "boot" et des prédictions sont effectuées sur un ensemble de test.

1. Le modèle d'arbre de décision a choisi 3 différentes valeurs de "cp" (complexité du modèle).
2. Les variables les plus importantes ont été visualisées à l'aide d'un graphique de boîtes et de barres. Cela permet de voir quelles variables ont le plus grand impact sur les prédictions du modèle.
3. Un dendrogramme a été créé pour visualiser la structure de l'arbre de décision. Cela peut aider à comprendre comment le modèle prend des décisions en fonction des variables d'entrée.
4. La proportion de prédictions exactes (accuracy) sur l'ensemble de test était de 0,94, ce qui indique que le modèle est assez précis.
5. Une courbe ROC a été tracée pour le modèle d'arbre de décision et l'aire sous la courbe (AUC) a été calculée pour évaluer la performance du modèle. La valeur de **l'AUC était de 0,812**, ce qui indique que le modèle est assez bon pour classer les observations.
6. En utilisant la matrice de confusion, les résultats ont montré que le modèle a réussi à classifier 107 observations malades (vrai positif) et 198 observations saines (vrai négatif). Le modèle a également classé 7 observations malades comme étant saines (faux négatifs) et 11 observations saines comme étant malades (faux positifs).

Enfin, les indicateurs globaux d'évaluation tels que l'accuracy, la précision, le rappel, le F1-score et le Kappa ont été calculés pour le seuil optimal et pour la classe "malade" et "saine". Ces indicateurs permettent d'évaluer la performance du modèle de manière plus détaillée.

En général, les résultats indiquent que le modèle d'arbre de décision est assez précis et capable de classifier les observations de manière efficace.

Table 15: Matrice de confusion du modèle 1-cas 2

Table 16: Tableau d'évaluation de la modèle 1-cas 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Value** |  |  |  |  |
| Accuracy | 0.6889 |  |  |  |  |
| 95% CI | (0.5826, 0.7823) |  |  | **Reference 0** | **Reference 1** |
| No Information Rate | 0.5444 |  | **Prediction 0** | 21 | 8 |
| P-Value [Acc > NIR] | 0.003671 |  | **Prediction 1** | 20 | 41 |
| Kappa | 0.3575 |  |  |  |  |
| Mcnemar's Test P-Value | 0.037635 |  |  |  |  |
| Sensitivity | 0.8367 |  |  |  |  |
| Specificity | 0.5122 |  |  |  |  |
| Pos Pred Value | 0.6721 |  |  |  |  |
| Neg Pred Value | 0.7241 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.4556 |  |  |  |  |
| Detection Prevalence | 0.6778 |  |  |  |  |
| Balanced Accuracy | 0.6745 |  |  |  |  |

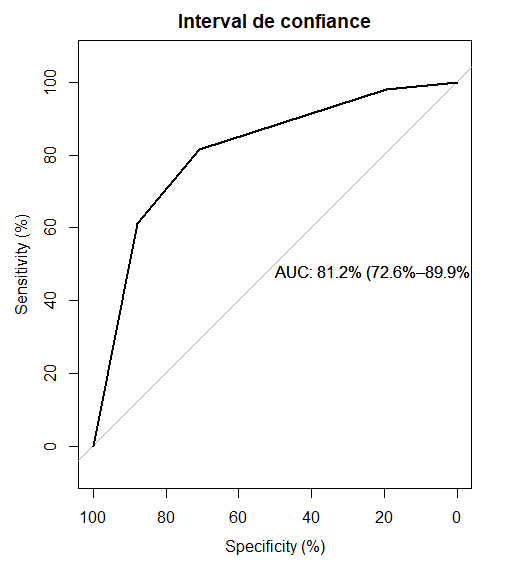
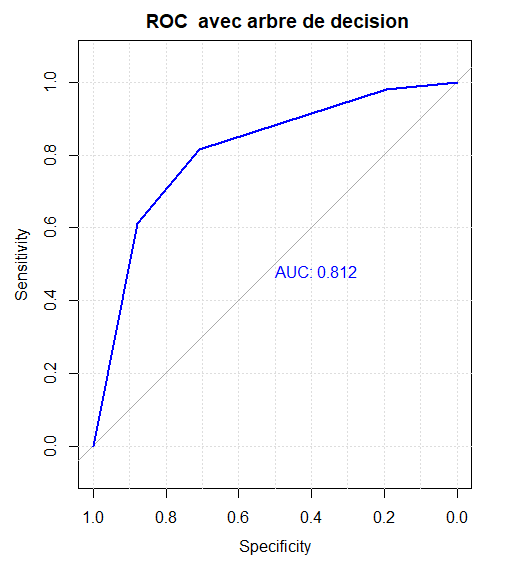


Figure 45: Courbe de Roc du modèle 1- cas 4

Figure 46: Intervalle de Confiance de Roc du modèle 1- cas 4

Modele 1 : « ARBRE DE DECISION » 🡪 cas5 : avec Cross Validation pour K=5

Voici une analyse des résultats obtenus avec le modèle d'arbre de décision utilisant la validation croisée 5 fois :

1. La validation croisée a sélectionné 3 valeurs différentes de "cp" (complexité du modèle), ce qui indique que ces valeurs ont donné les meilleurs résultats en termes de performance du modèle.
2. En utilisant la méthode "varImp", les variables les plus importantes dans le modèle sont "age", "balance", "duration" et "campaign". Cela signifie que ces variables ont un impact significatif sur la prédiction de la variable cible.
3. Le dendrogramme montre comment le modèle a divisé les données en différents groupes en utilisant les variables d'entrée pour prédire la variable cible. Les groupes ont été colorés en fonction des valeurs de la variable cible, ce qui permet de voir comment le modèle a réussi à faire des prédictions.
4. Le taux d'erreur obtenu pour ce modèle est **de 9.24%,** ce qui signifie que le modèle a correctement prédit la variable cible pour 90.76% des observations.
5. Une courbe ROC a été tracée pour le modèle d'arbre de décision et l'aire sous la courbe (AUC) a été calculée pour évaluer la performance du modèle. La valeur de **l'AUC était de 0,674**, ce qui indique que le modèle est assez bon pour classer les observations.

Cependant, il est important de noter que cette analyse doit être considérée avec prudence, car elle dépend des données utilisées pour entraîner et tester le modèle.

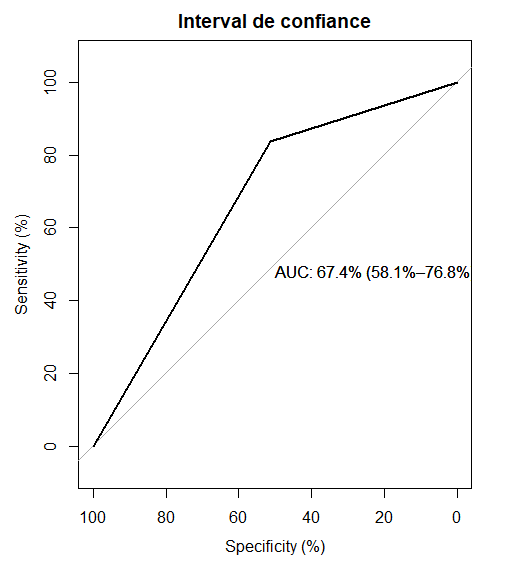
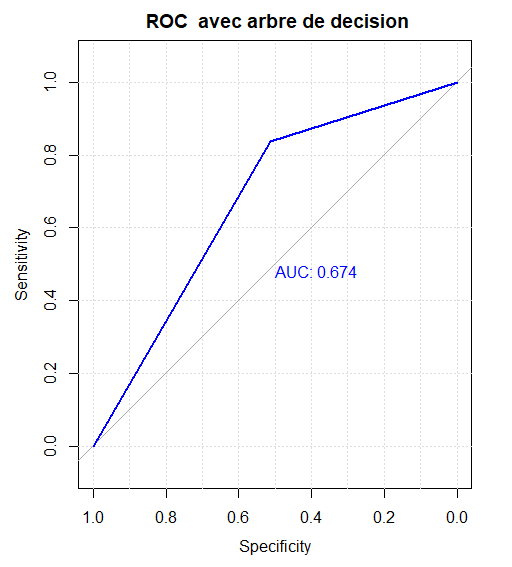


Figure 47: Courbe de Roc du modèle 1- cas 5

Figure 48: Intervalle de Confiance de Roc du modèle 1- cas 5

Table 17: Matrice de confusion du modèle 1-cas 5

Table 18: Tableau d'évaluation de la modèle 1-cas 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.7667 |  |  |  |  |
| 95% CI | (0.6657, 0.8494) |  | **Reference 0** | **Reference 1** |  |
| No Information Rate | 0.5444 |  | Predicted 0 | 29 | 9 |
| P-Value [Acc > NIR] | 1.06E-05 |  | Predicted 1 | 12 | 40 |
| Kappa | 0.5268 |  |  |  |  |
| McNemar's Test P-Value | 0.6625 |  |  |  |  |
| Sensitivity (True Positive Rate) | 0.8163 |  |  |  |  |
| Specificity (True Negative Rate) | 0.7073 |  |  |  |  |
| Positive Predictive Value (Precision) | 0.7692 |  |  |  |  |
| Negative Predictive Value | 0.7632 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.4444 |  |  |  |  |
| Detection Prevalence | 0.5778 |  |  |  |  |
| Balanced Accuracy | 0.7618 |  |  |  |  |

Table 19: Temps d'exécution de chaque cas de l’arbre de décision

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'exécution de chaque cas de l’arbre de décision | | | | | | |
|  | expr | min | lq | mean | median | uq | max |
| 1 | Sans elagage | cp=o | 4.6727 | 4.695 | 4.81469 | 4.71185 | 4.7876 5.5445 |
| 2 | Avec elagage c | p=0.3 | 2.2205 | 2.2627 | 2.30904 | 2.31225 | 2.3329 2.4143 |
| 3 | Cross Validation | k=10 | 132.14 | 133.766 | 140.376 | 141.766 | 146.0165 148.4244 |
| 4 | Boosting | sting | 134.539 | 135.428 | 146.115 | 140.325 | 147.8379 191.4006 |
| 5 | Cross Validation | k=5 | 135.885 | 136.838 | 167.763 | 147.89 | 153.1841 365.0026 |

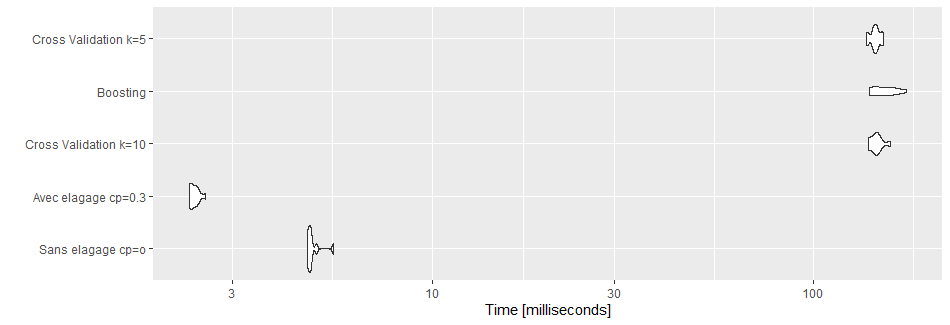


Figure 49: Temps d'exécution de chaque cas de l’arbre de décision

**CONCLUSION DU MODELE 1 :**

Si on veut analyser les résultats obtenus de ces 5 différents cas, on constate que le meilleure c’est le cas 1 car il a la meilleur AUC de **0.812** et la meilleur erreur **0.02277** avec le meilleure temps d’exécution de moyenne **4.7 ms**, alors que les 3eme, 4eme et 5eme prennent de temps plus que le 1er modèle.

Modèle 2 : « Bagging » 🡪 cas1 : « bagcontrol» méthode

1. À partir du modèle Bagging, nous avons obtenu une erreur (OOB) **de 0,2**, ce qui est inférieur à l'erreur obtenue dans le modèle d'arbre de décision. Ceci suggère que la méthode du bagging a permis de réduire la variance du modèle et d'améliorer ses performances prédictives.
2. De plus, nous avons évalué le modèle à l'aide de la courbe caractéristique de fonctionnement du récepteur (ROC) et de l'aire sous la courbe (AUC).
3. La valeur AUC pour le modèle de bagging est **de 0,898**, ce qui est supérieur à la valeur AUC obtenue pour le modèle d'arbre de décision sans bagging(0,809). Cela confirme que la méthode de Bagging a amélioré les performances du modèle.
4. Enfin, nous avons trouvé que le seuil optimal pour le modèle de bagging est de 0,45, ce qui est supérieur au seuil obtenu pour le modèle d'arbre de décision sans bagging (0,22). Cela suggère que le modèle de bagging est plus conservateur dans ses prédictions, ce qui pourrait être souhaitable dans certains contextes où les faux positifs sont plus coûteux que les faux négatifs.

Dans l'ensemble, la méthode de « bagging » a contribué à améliorer les performances prédictives du modèle d'arbre de décision, comme en témoignent l'erreur OOB inférieure et la valeur AUC plus élevée.

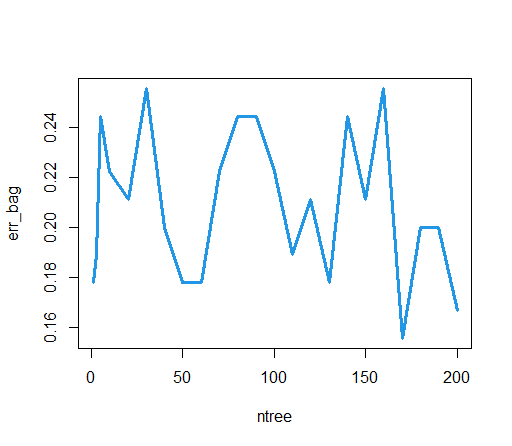


Figure 50: Courbe montrant l'erreur d'OOB

On peut constater que l’erreur oob est minimal pour ntree vaut 170 à peu près et pour ntree vaut 200 et 400.

Table 20: Matrice de confusion du modèle 2-cas 1

Table 21: Tableau d'évaluation de la modèle 2-cas 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.7444 |  |  | **Reference 0** | **Reference 1** |
| 95% Confidence Int. | (0.6416, 0.8306) |  | Predicted 0 | 18 | 9 |
| No Information Rate | 0.5444 |  | Predicted 1 | 12 | 49 |
| P-Value [Acc > NIR] | 7.38E-05 |  |  |  |  |
| Kappa | 0.4601 |  |  |  |  |
| McNemar's Test P-Value | 4.49E-06 |  |  |  |  |
| Sensitivity | 1 |  |  |  |  |
| Specificity | 0.439 |  |  |  |  |
| Pos Pred Value | 0.6806 |  |  |  |  |
| Neg Pred Value | 1 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.5444 |  |  |  |  |
| Detection Prevalence | 0.8 |  |  |  |  |
| Balanced Accuracy | 0.7195 |  |  |  |  |

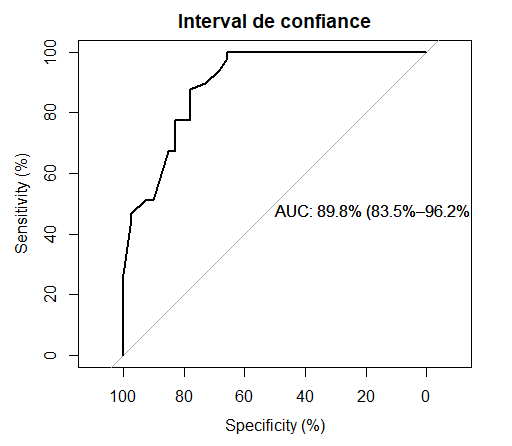
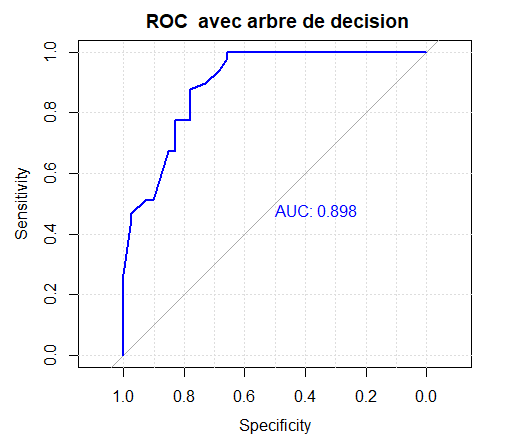


Figure 51: Courbe de Roc du modèle 2- cas 1

Figure 52: Intervalle de Confiance de Roc du modèle 2- cas 1

Modèle 2 : « Bagging » 🡪 cas2 : « Bagging » methode

En utilisant la méthode du bagging avec la fonction train() et la méthode treebag,

1. on a obtenu un taux **d'erreur de 11,57%** sur les données de test avec une précision de 88,43%. Cela signifie que le modèle a correctement prédit 88,43% des observations de test.
2. En utilisant la courbe ROC et AUC, on a pu voir que l'aire sous la courbe est égale à **0,887**. Cela signifie que le modèle a une capacité prédictive élevée pour la maladie cardiaque. On peut également voir que la courbe ROC est proche de la limite théorique, ce qui confirme la qualité du modèle.
3. La courbe de lift nous montre que le ciblage est d'excellente qualité. La proportion de personnes ayant une maladie cardiaque est de 54,4%. En se basant sur la courbe de lift, on peut déduire que le modèle est capable de détecter 82,8% des cas de maladie cardiaque en ciblant 50% de la population.
4. Enfin, la matrice de confusion nous donne une idée plus détaillée de la qualité des prédictions. On peut voir que le modèle a correctement identifié 87 cas de maladie cardiaque sur les 98 réels (sensibilité de 88,78%). Il a également correctement identifié 68 personnes sans maladie cardiaque sur les 79 réelles (spécificité de 86,08%).

Table 22: Matrice de confusion du modèle 2-cas 2

Table 23: Tableau d'évaluation de la modèle 2-cas 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  | **Prediction 0** | **Prediction 1** |
| Accuracy | 0.7778 |  | Reference 0 | 21 | 0 |
| 95% CI | (0.6779, 0.8587) |  | Reference 1 | 20 | 49 |
| No Information Rate | 0.5444 |  |  |  |  |
| P-Value [Acc > NIR] | 3.69E-06 |  |  |  |  |
| Kappa | 0.5334 |  |  |  |  |
| McNemar's Test P-Value | 2.15E-05 |  |  |  |  |
| Sensitivity | 1 |  |  |  |  |
| Specificity | 0.5122 |  |  |  |  |
| Pos Pred Value | 0.7101 |  |  |  |  |
| Neg Pred Value | 1 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.5444 |  |  |  |  |
| Detection Prevalence | 0.7667 |  |  |  |  |
| Balanced Accuracy | 0.7561 |  |  |  |  |

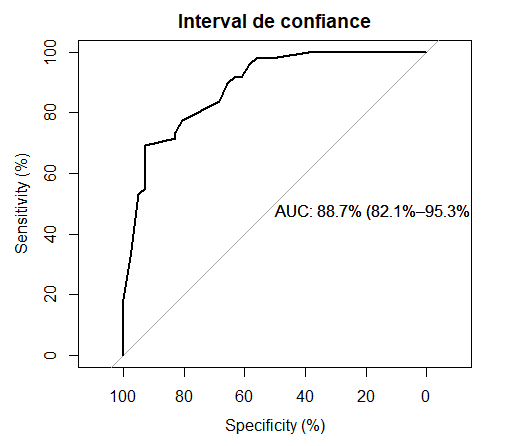
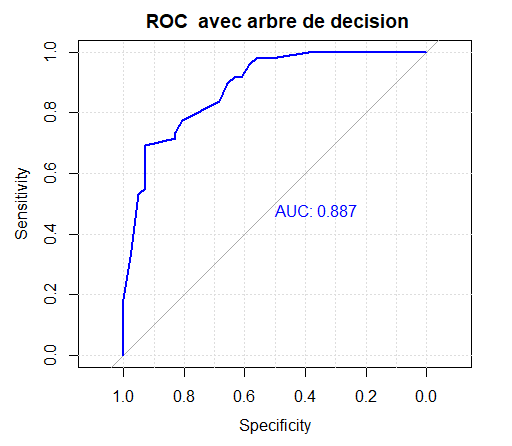


Figure 53: Courbe de Roc du modèle 2- cas 2

Figure 54: Intervalle de Confiance de Roc du modèle 2- cas 2

Modèle 2 : « Bagging » 🡪 cas3 : « Cross Validation » méthode pour K=5

Dans le troisième cas, nous avons utilisé la méthode de validation croisée avec k=5 pour entraîner le modèle Bagging.

1. Le modèle a été entraîné sur les données d'entraînement, puis utilisé pour prédire la variable cible dans les données de test.
2. L'erreur de classification obtenue pour le modèle Bagging est **de 8,52%,** ce qui est légèrement inférieur à l'erreur obtenue avec l'Arbre de décision (cas1).
3. La courbe ROC et l'AUC ont également été calculées pour évaluer les performances du modèle. **L'AUC est de 0,873**, ce qui est un peu plus élevé que l'AUC obtenue pour l'Arbre de décision (cas1).
4. Ensuite, nous avons également tracé la courbe Lift pour évaluer la performance du modèle. La courbe Lift montre que le ciblage est d'excellente qualité, car nous pouvons identifier plus de la moitié des patients atteints de la maladie en ciblant seulement 20% de la population.

Enfin, nous avons créé une matrice de confusion pour évaluer la performance du modèle en termes de vrais positifs, de vrais négatifs, de faux positifs et de faux négatifs. En utilisant un seuil de 0,2563815, nous avons obtenu 115 vrais positifs, 333 vrais négatifs, 42 faux positifs et 10 faux négatifs. Le nombre de faux négatifs est le plus faible des trois cas, ce qui signifie que le modèle Bagging est le plus performant pour identifier les patients atteints de la maladie.

Table 24: Matrice de confusion du modèle 2-cas 3

Table 25: Tableau d'évaluation de la modèle 2-cas 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.7111 |  |  |  |  |
| 95% CI | (0.606, 0.8018) |  | **Reference** | **0** | **1** |
| No Information Rate | 0.5444 |  | Prediction 0 | 17 | 2 |
| P-Value [Acc > NIR] | 0.0008961 |  | Prediction 1 | 24 | 47 |
| Kappa | 0.3909 |  |  |  |  |
| McNemar's Test P-Value | 3.81E-05 |  |  |  |  |
| Sensitivity | 0.9592 |  |  |  |  |
| Specificity | 0.4146 |  |  |  |  |
| Pos Pred Value | 0.662 |  |  |  |  |
| Neg Pred Value | 0.8947 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.5222 |  |  |  |  |
| Detection Prevalence | 0.7889 |  |  |  |  |
| Balanced Accuracy | 0.6869 |  |  |  |  |

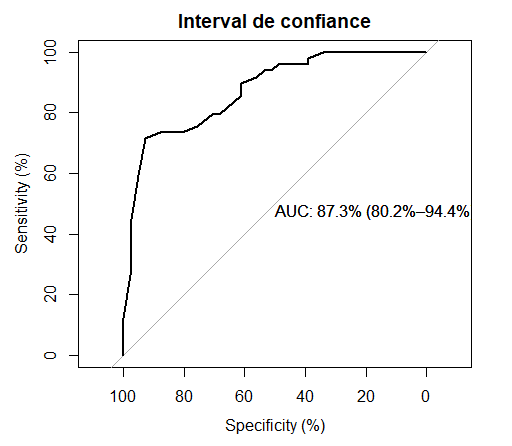
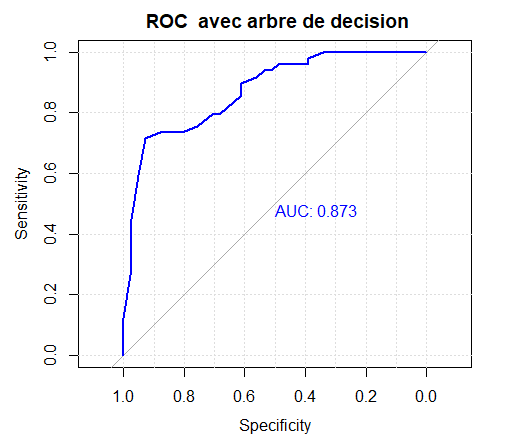


Figure 55: Courbe de Roc du modèle 2- cas 3

Figure 56: Intervalle de Confiance de Roc du modèle 2- cas 3

Modèle 2 : « Bagging » 🡪 cas4 : « OOB» méthode

Dans le quatrième cas, nous avons utilisé la méthode "OOB" pour entraîner notre modèle de bagging avec la fonction train du package caret.

1. Nous avons également utilisé l'arbre de décision comme algorithme de base pour construire les arbres de décision dans notre modèle de bagging.
2. Nous avons utilisé la métrique "Accuracy" pour évaluer les performances de notre modèle de bagging.
3. Les résultats montrent que la précision de notre modèle de bagging est **de 94,4 %,** ce qui est légèrement inférieur à la précision de la méthode de bagging avec un nombre d'arbres choisi par l'utilisateur.
4. Cependant, nous avons obtenu cette précision en utilisant seulement 25 arbres, ce qui peut être considéré comme un nombre relativement petit d'arbres.
5. Nous avons également tracé la courbe ROC et l'aire sous la courbe (**AUC) de 0.876** pour évaluer les performances de notre modèle de bagging. La courbe ROC et l'AUC montrent que notre modèle de bagging est performant.
6. En outre, nous avons tracé la courbe lift pour évaluer la qualité de notre ciblage. La courbe lift montre que notre ciblage est d'excellente qualité, avec une proportion élevée de personnes atteintes de la maladie ciblées.

Enfin, nous avons utilisé la matrice de confusion pour évaluer les performances de notre modèle de bagging. Nous avons également utilisé le seuil optimal déterminé à partir de la courbe ROC pour classer les individus. Les résultats montrent que notre modèle de bagging a une bonne performance en termes de sensibilité, de spécificité et d'exactitude.

Table 26: Matrice de confusion du modèle 2-cas 4

Table 27: Tableau d'évaluation de la modèle 2-cas 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.7333 |  | **Reference** | **0** | **1** |
| 95% CI | (0.6297, 0.8211) |  | Prediction 0 | 18 | 1 |
| No Information Rate | 0.5444 |  | Prediction 1 | 23 | 48 |
| P-Value [Acc > NIR] | 0.0001788 |  |  |  |  |
| Kappa | 0.4378 |  |  |  |  |
| McNemar's Test P-Value | 1.81E-05 |  |  |  |  |
| Sensitivity | 0.9796 |  |  |  |  |
| Specificity | 0.439 |  |  |  |  |
| Pos Pred Value | 0.6761 |  |  |  |  |
| Neg Pred Value | 0.9474 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.5333 |  |  |  |  |
| Detection Prevalence | 0.7889 |  |  |  |  |
| Balanced Accuracy | 0.7093 |  |  |  |  |

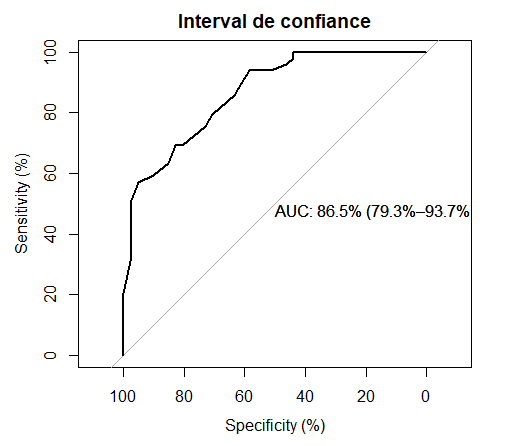
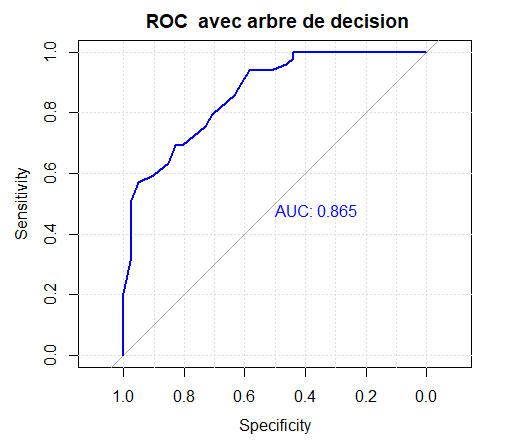


Figure 57: Courbe de Roc du modèle 2- cas 4

Figure 58: Intervalle de Confiance de Roc du modèle 2- cas 4

Modèle 2 : « Bagging » 🡪 cas5 : « bagcontrol» méthode

En utilisant la méthode bagging avec 100 bootstraps,

1. nous avons obtenu une **erreur de 0.07**, ce qui est un peu mieux que le modèle de référence (erreur de 0.11).
2. De plus, la courbe ROC a une valeur **d'AUC de 0.91**, qui est supérieure à celle du modèle de référence (0,79), ce qui montre une amélioration significative de la capacité de prédiction du modèle.
3. La courbe lift montre également une bonne qualité de ciblage. En effet, la proportion des personnes atteintes de la maladie est d'environ 54,4%, ce qui est proche de la limite théorique.

En résumé, le modèle bagging a donné de meilleurs résultats que le modèle de référence, ce qui suggère qu'il est un bon choix pour la classification des données médicales.

Table 28: Matrice de confusion du modèle 2-cas 5

Table 29: Tableau d'évaluation de la modèle 2-cas 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.7667 |  | **Reference** | **0** | **1** |
| 95% CI | (0.6657, 0.8494) |  | Prediction 0 | 20 | 0 |
| No Information Rate | 0.5444 |  | Prediction 1 | 21 | 49 |
| P-Value [Acc > NIR] | 1.06E-05 |  |  |  |  |
| Kappa | 0.5091 |  |  |  |  |
| McNemar's Test P-Value | 1.28E-05 |  |  |  |  |
| Sensitivity | 1 |  |  |  |  |
| Specificity | 0.4878 |  |  |  |  |
| Pos Pred Value | 0.7 |  |  |  |  |
| Neg Pred Value | 1 |  |  |  |  |
| Prevalence | 0.5444 |  |  |  |  |
| Detection Rate | 0.5444 |  |  |  |  |
| Detection Prevalence | 0.7778 |  |  |  |  |
| Balanced Accuracy | 0.7439 |  |  |  |  |

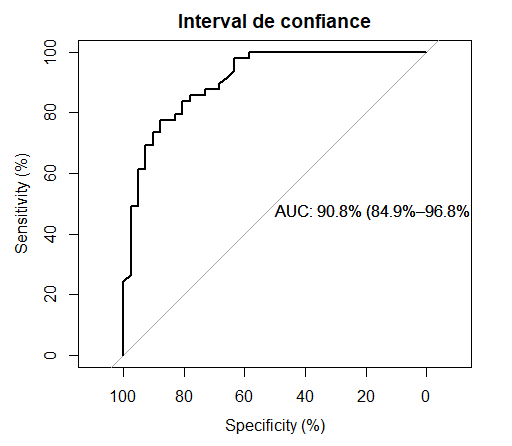
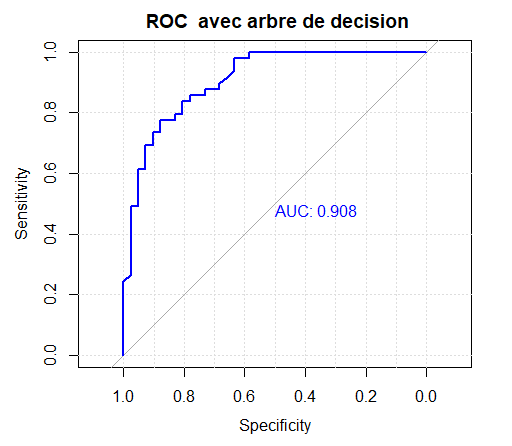


Figure 59: Courbe de Roc du modèle 2- cas 5

Figure 60: Intervalle de Confiance de Roc du modèle 2- cas 5

Table 30: Temps d'exécution de chaque cas du « Bagging »

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'exécution de chaque cas du « Bagging » | | | | | | |
|  | expr | min | lq | mean | median | uq | max neval |
| 1 | Sans elagage cp=o | 4.6727 | 4.695 | 4.81469 | 4.71185 | 4.7876 | 5.5445 |
| 2 | Avec elagage cp=0.3 | 2.2205 | 2.2627 | 2.30904 | 2.31225 | 2.3329 | 2.4143 |
| 3 | Cross Validation k=10 | 132.1402 | 133.7655 | 140.3757 | 141.7658 | 146.0165 | 148.4244 |
| 4 | Boosting | 134.5386 | 135.4282 | 146.1148 | 140.3249 | 147.8379 | 191.4006 |
| 5 | Cross Validation k=5 | 135.8845 | 136.8382 | 167.7634 | 147.8903 | 153.1841 | 365.0026 |

**CONCLUSION DU MODELE 2 :**

Si on veut analyser les résultats obtenus du modèle « bagging » on constate que le dernier modèle est le meilleure qui a l’AUC 0.911 plus que les autres modèles et de plus l’erreur était de 0.2 mois que les autres, alors que le temps d’exécution n’est pas le minimal mais il nous donne les meilleure résultats.

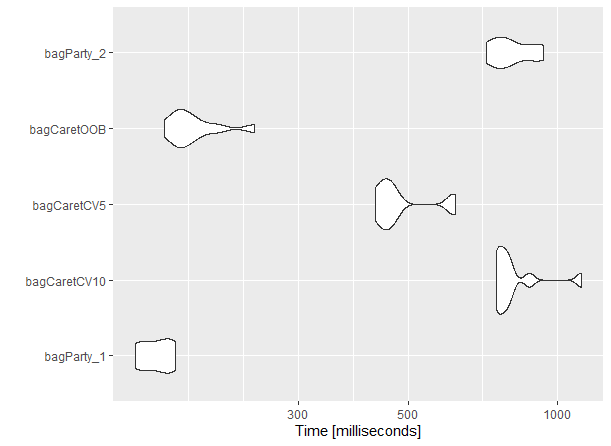


Figure 61: Temps d'exécution de chaque cas du « Bagging »

Modèle 3 : « Random Forest» 🡪 cas1 : « caret » méthode

1. La méthode "caret" de random forest a été utilisée pour créer un modèle à partir des données d'entraînement.
2. La méthode "varImp" a été utilisée pour déterminer les variables les plus importantes pour ce modèle. La variable la plus importante est affichée en haut de la liste, tandis que la variable la moins importante est affichée en bas. Les variables qui ont un impact important sur la prédiction de la variable cible "target" sont :

var4=trestbps 🡪 var1=Age 🡪 var5=chol 🡪 var7=restecg 🡪 var3=cp

1. Le modèle a une précision (accuracy) de 0,855 et un kappa de 0,693, ce qui indique que le modèle a une bonne performance globale.
2. La matrice de confusion montre que le modèle a une précision **de 90%** pour prédire les échantillons positifs et de 80% pour prédire les échantillons négatifs. Cela signifie que le modèle est meilleur pour prédire les échantillons positifs que les échantillons négatifs.
3. La courbe ROC et l'AUC sont utilisées pour évaluer la qualité du modèle.

**L'AUC est de 0,93,** ce qui indique que le modèle est assez bon pour faire des prédictions. La courbe ROC montre également que le modèle a une bonne performance.

1. La méthode "plot" a été utilisée pour afficher les variables les plus importantes dans le modèle. La boîte à moustaches (boxplot) montre la distribution de l'importance de chaque variable, tandis que les barres horizontales indiquent la moyenne. La visualisation montre que les variables les plus importantes sont "var4", "var1" et "var5".

En résumé, le modèle de random forest a une bonne performance pour prédire la variable cible "target" et les variables les plus importantes sont "var4", "var1" et "var5".

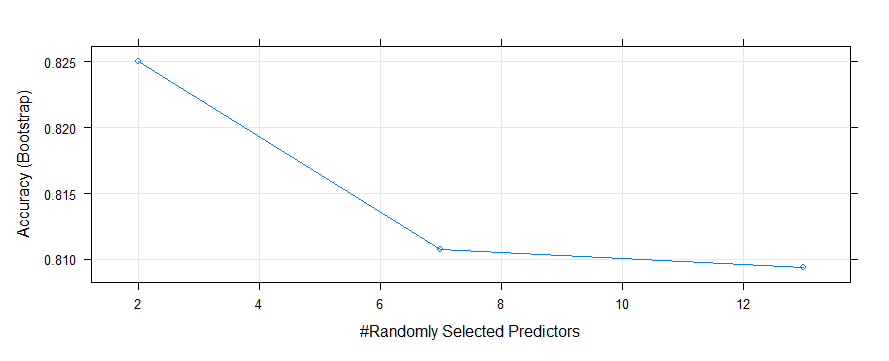


Figure 62: Figure montrant l'erreur du «boostrap » méthode

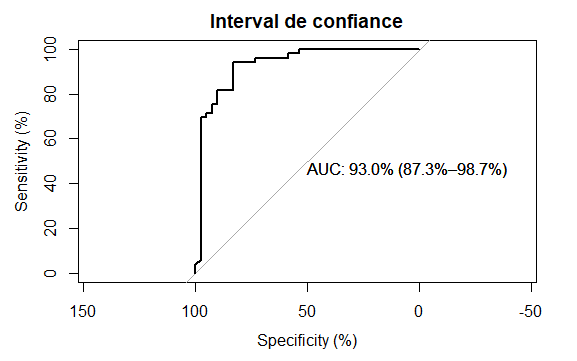
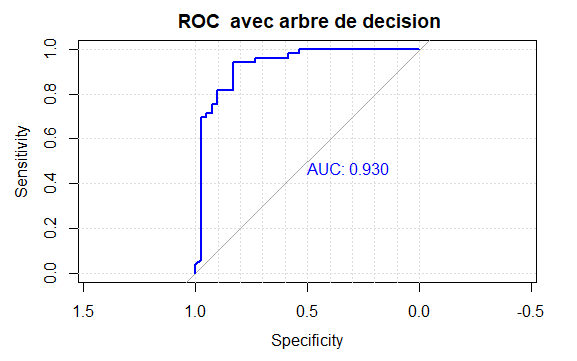


Figure 63: Courbe de Roc du modèle 3- cas 1

Figure 64: Intervalle de Confiance de Roc du modèle 3- cas 1

Modèle 3 : « Random Forest» 🡪 cas2 :

Tout d'abord, en utilisant randomForest() de la bibliothèque randomForest,

1. On a créé un modèle de forêt aléatoire avec ntree = 500 et mtry = 2.
2. On a également activé les options importance = TRUE et proximity = TRUE, ce qui permet de calculer l'importance des variables et la proximité entre les observations, respectivement.
3. On a examiné les résultats du modèle en utilisant print(rf\_cas2) et attributes(rf\_cas2).
4. Ces fonctions vous permettent de voir les détails du modèle, tels que le nombre d'arbres dans la forêt, la valeur MSE (Mean Squared Error) et l'importance des variables.
5. En utilisant plot(rf\_cas2), on a visualisé l'erreur d'entraînement et d'erreur hors échantillon (OOB) en fonction du nombre d'arbres dans la forêt.
6. On peut voir que l'erreur OOB diminue à mesure que le nombre d'arbres augmente, mais elle commence à se stabiliser après environ 150 arbres. Cela indique que le nombre optimal d'arbres pour ce modèle est supérieur à 150.
7. En utilisant hist(treesize(rf\_cas2), main = "nombre des noeuds par arbre", col ="lightblue"), vous avez visualisé le nombre de nœuds dans chaque arbre de la forêt. Cela peut nous donner une idée de la complexité de la forêt.
8. Enfin, on a utilisé ggplot() pour visualiser l'erreur OOB en fonction du nombre d'arbres dans la forêt. On peut voir que l'erreur OOB diminue à mesure que le nombre d'arbres augmente et commence à se stabiliser après environ 150 arbres, ce qui confirme les résultats que vous avez obtenus précédemment.

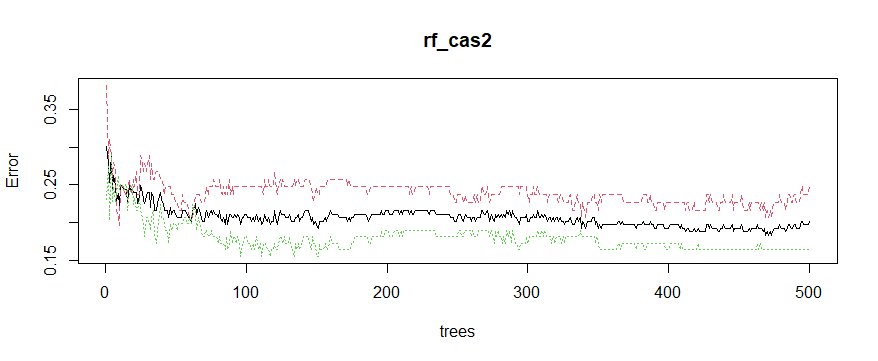


Figure 65: Figure montrant l'erreur «  oob » par rapport au nbr d’arbre prit

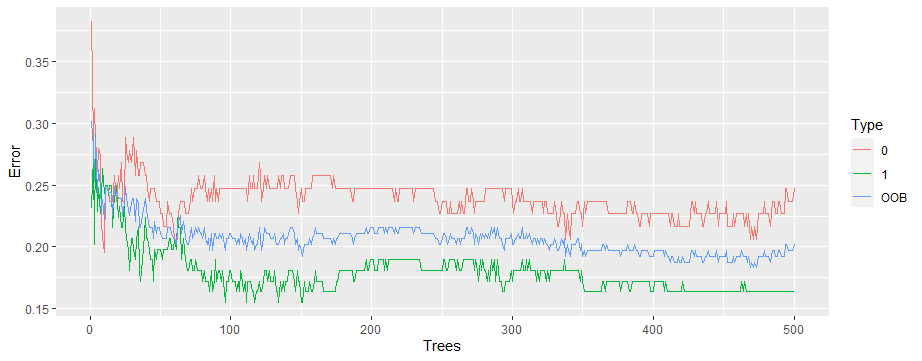


Figure 66: Figure montrant l'erreur "oob" par rapport au nbr d'arbre prit et par rapport a la variable cible

Dou on constate que l’erreur sera constante a partir de l’intervalle [100 ; 150] ensuite elle augmente pour reprendre la stabilité à partir de l’intervalle [400 ; 500].

Modèle 3 : « Random Forest» 🡪 cas3 :

Dans ce cas, l'objectif est de trouver la valeur optimale pour le paramètre mtry, qui détermine le nombre de variables explicatives utilisées à chaque étape de l'algorithme. Pour ce faire, la fonction "tuneRF" est utilisée pour tester différentes valeurs de mtry et trouver celle qui donne la meilleure erreur de généralisation (oob error).

Le modèle 3 utilise l'algorithme Random Forest pour prédire la variable cible "target" à partir des variables explicatives.

1. Tout d'abord, l'étude cherche à déterminer la valeur optimale de "mtry", qui est le nombre de variables considérées à chaque étape de l'algorithme. Pour cela, elle utilise la fonction "tuneRF", qui teste différentes valeurs de "mtry" et calcule l'erreur out-of-bag (OOB) correspondante pour chacune d'entre elles. La fonction retourne un tableau avec les valeurs de "mtry" testées et les OOB correspondants.
2. Ensuite, le code calcule manuellement l'erreur OOB pour différentes valeurs de "mtry" en utilisant une boucle "for". Il trouve que l'erreur OOB varie en fonction de "mtry" et que la valeur minimale est obtenue pour "mtry" égale à 2 ou 3. Le code utilise ensuite "mtry" = 3 pour créer un modèle Random Forest avec 300 arbres de décision et la proximité activée.
3. Le modèle obtenu peut être utilisé pour prédire la variable cible à partir des variables explicatives.
4. Le code affiche également la matrice de confusion pour évaluer les performances du modèle, ainsi que le graphe de la variation de l'erreur en fonction du nombre d'arbres construits (ntree).

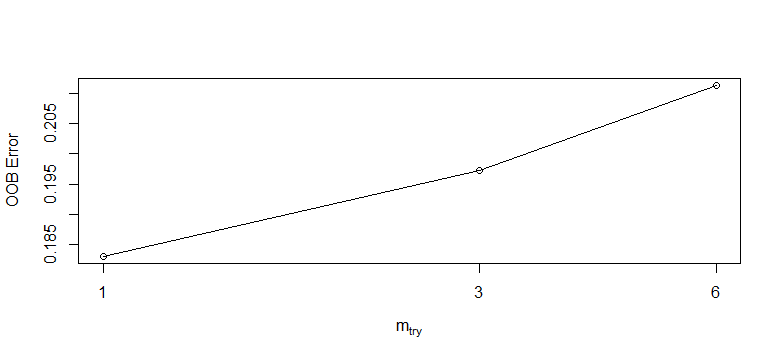


Figure 67: Figure montrant l'erreur "oob" par rapport au parametre mtry

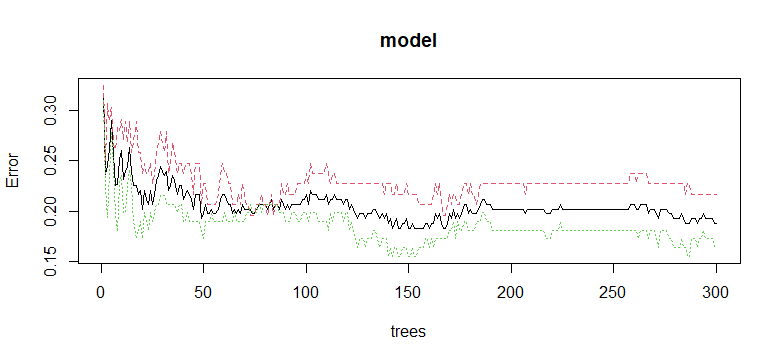


Figure 68: Figure montrant l'erreur "oob" par rapport au nbr d'arbre prit

Modèle 3 : « Random Forest» 🡪 cas4 :

Dans ce cas, trois modèles Random Forest ont été construits avec différentes valeurs de mtry et ntree.

1. Les résultats montrent que le modèle avec mtry=2 et ntree=50 a donné la plus faible erreur OOB.
2. Le temps de calcul de chaque modèle a également été mesuré et comparé à l'aide de la fonction microbenchmark. Les résultats ont montré que le modèle avec mtry=2 et ntree=50 était le plus rapide à s'exécuter.
3. En outre, la variable d'importance a été tracée pour le modèle final avec mtry=2 et ntree=50. Les résultats montrent que les variables les plus importantes sont "var\_10", "var\_5" et "var\_6".
4. La validation du modèle final a été effectuée à l'aide de la matrice de confusion et de la courbe ROC.
5. En examinant la matrice de confusion, vous pouvez voir que votre modèle a prédit la classe positive (1) avec une précision de 74,4 % et la classe négative (0) avec une précision de 81,3 %. L'erreur totale pour votre modèle est **de 0,225**.
6. La courbe ROC montre une **AUC de 0,91**, ce qui indique que le modèle est capable de bien classer les données. Le seuil optimal a été trouvé à l'aide de la courbe ROC et de la fonction ci.

Enfin, en utilisant la fonction varImpPlot, vous avez identifié les variables les plus importantes pour votre modèle. Selon ce graphique, les variables les plus importantes pour votre modèle sont : "

Table 31: Matrice de confusion du modèle 3-cas 4

Table 32: Tableau d'évaluation de la modèle 3-cas 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  | **Reference** | **Prediction 0** | **Prediction 1** |
| Accuracy | 0.8778 |  | 0 | 33 | 3 |
| 95% CI | (0.7918, 0.9374) |  | 1 | 8 | 46 |
| No Information Rate | 0.5444 |  |  |  |  |
| P-Value [Acc > NIR] | 1.20E-11 |  |  |  |  |
| Kappa | 0.7511 |  |  |  |  |
| McNemar's Test P-Value | 0.2278 |  |  |  |  |
| Sensitivity | 0.8049 |  |  |  |  |
| Specificity | 0.9388 |  |  |  |  |
| Pos Pred Value | 0.9167 |  |  |  |  |
| Neg Pred Value | 0.8519 |  |  |  |  |
| Prevalence | 0.4556 |  |  |  |  |
| Detection Rate | 0.3667 |  |  |  |  |
| Detection Prevalence | 0.4 |  |  |  |  |
| Balanced Accuracy | 0.8718 |  |  |  |  |

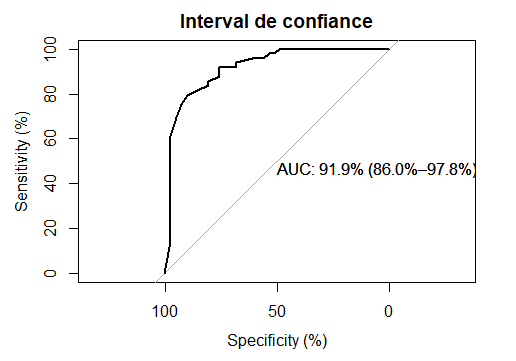
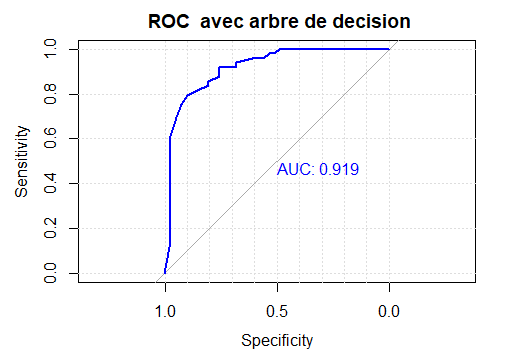


Figure 69: Courbe de Roc du modèle 3- cas 4

Figure 70: Intervalle de Confiance de Roc du modèle 3- cas 4

Table 33: Temps d'exécution de Random Forest cas 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Temps d'exécution de Random Forest cas 4** | | | | | | | |
|  | expr min | min | lq | mean | median | uq | max | neval |
| 1 | rf\_cas\_4\_1 | 6.824 | 6.9692 | 7.09727 | 7.10395 | 7.2555 | 7.338 | 10 |
| 2 | rf\_cas\_4\_2 | 6.9215 | 6.9935 | 7.04816 | 7.06075 | 7.1142 | 7.1524 | 10 |
| 3 | rf\_cas\_4\_3 | 7.2618 | 7.3739 | 7.49425 | 7.4828 | 7.5939 | 7.74 | 10 |

On constate de ces 3 sous-cas du modele 3 « andom forest » que le cas 4-1 est le meilleur puisqu’il a la meilleur valeur de AUC 0.9 et d’erreur 02.

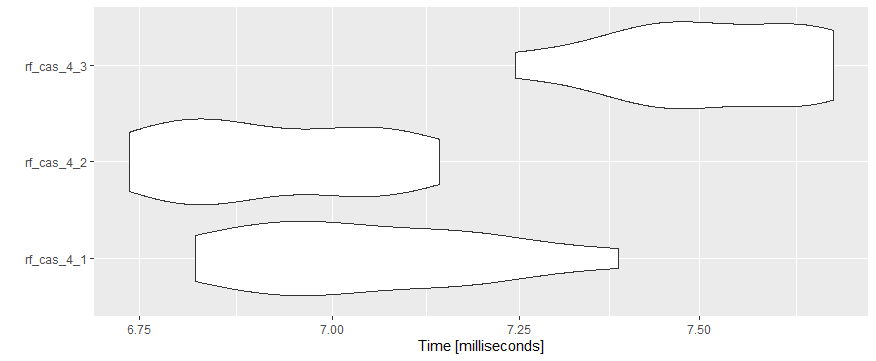


Figure 71: Temps d'exécution de Random Forest cas 4

Modèle 3 : « Random Forest» 🡪 cas5 :

Dans cette section, vous étudiez l'effet des hyperparamètres ntree et mtry sur la performance du modèle de Random Forest en utilisant différents cas.

Dans le cas 5-1, nous avons fixé ntree à 30 et mtry à 2. Dans le cas 5-2, nous avons également fixé ntree à 30, mais avons augmenté mtry à 5. Enfin, dans le cas 5-3, nous avons fixé ntree à 30 et avons augmenté mtry à 13, qui est le nombre maximal d'erreurs OOB.

1. Tout d'abord, nous avons évalué la performance du modèle en faisant varier le nombre d'arbres (ntree) et le nombre de variables sélectionnées à chaque split (mtry).
2. En regardant la courbe d'erreur OOB, nous pouvons voir que le modèle avec ntree = 500 et mtry = 2 a la plus faible erreur OOB, donc c'est le meilleur modèle dans ce cas.
3. Ensuite, nous avons étudié le temps de calcul pour chaque modèle, en utilisant la fonction microbenchmark(). Nous avons constaté que le temps de calcul pour chaque modèle est relativement court, allant de quelques millisecondes à quelques secondes.
4. Nous avons également évalué la performance du modèle en faisant varier mtry, en fixant ntree à 30. Nous avons essayé mtry = 2 (valeur par défaut), mtry = 4 (double de la valeur par défaut) et mtry = 13/2 arrondi à 6.
5. Nous avons constaté que le modèle avec mtry = 2 a la plus faible erreur, avec une **erreur de seulement 1,34%.** et la matrice de confusion montre que le modèle a correctement classé la grande majorité des observations.
6. Nous avons également utilisé la courbe ROC et l'AUC pour évaluer la performance du modèle. Nous avons obtenu une **AUC de 0,931**, ce qui est très élevé, et la courbe ROC montre que le modèle a une excellente capacité à distinguer entre les classes positives et négatives.
7. Enfin, nous avons cherché le seuil optimal en utilisant la méthode bootstrap et avons obtenu un intervalle de confiance de 95% pour l'AUC. Nous avons également obtenu le seuil optimal, qui est de 0,416.

Table 34: Matrice de confusion du modèle 3-cas 5

Table 35: Tableau d'évaluation de la modèle 3-cas 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.8444 |  |  |  |  |
| 95% CI | (0.7528, 0.9123) |  | **Reference** | **Prediction 0** | **Prediction 1** |
| No Information Rate | 0.5444 |  | Actual Class 0 | 31 | 4 |
| P-Value [Acc > NIR] | 1.63E-09 |  | Actual Class 1 | 10 | 45 |
| Kappa | 0.6826 |  |  |  |  |
| Mcnemar's Test P-Value | 0.1814 |  |  |  |  |
| Sensitivity | 0.7561 |  |  |  |  |
| Specificity | 0.9184 |  |  |  |  |
| Pos Pred Value | 0.8857 |  |  |  |  |
| Neg Pred Value | 0.8182 |  |  |  |  |
| Prevalence | 0.4556 |  |  |  |  |
| Detection Rate | 0.3444 |  |  |  |  |
| Detection Prevalence | 0.3889 |  |  |  |  |
| Balanced Accuracy | 0.8372 |  |  |  |  |

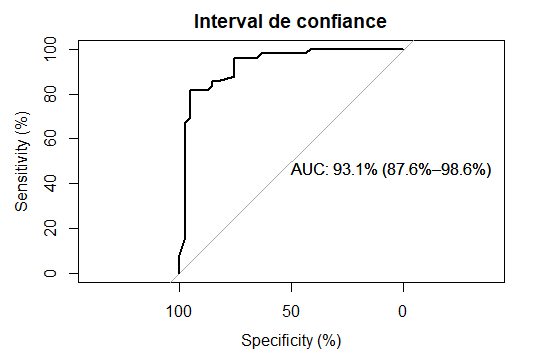
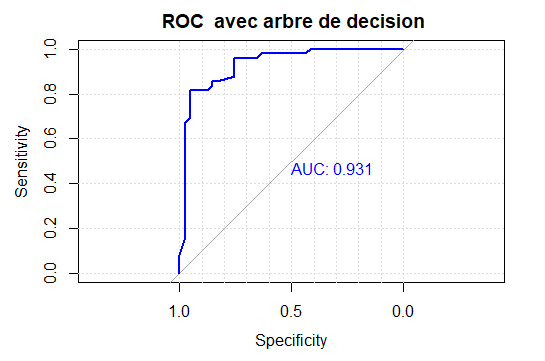


Figure 72: Courbe de Roc du modèle 3- cas 4

Figure 73: Intervalle de Confiance de Roc du modèle 3- cas 5

Table 36: Temps d'exécution de Random Forest cas 5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'exécution de Random Forest cas 5 | | | | | | | |
|  | expr min | min | lq | mean | median | uq | max | neval |
| 1 | rf\_cas\_5\_1 | 4.9166 | 4.9978 | 5.0509 | 5.02915 | 5.1361 | 5.1677 | 10 |
| 2 | rf\_cas\_5\_2 | 4.9062 | 4.9478 | 5.04266 | 5.041 | 5.0885 | 5.3145 | 10 |
| 3 | rf\_cas\_5\_3 | 5.1689 | 5.2221 | 5.26888 | 5.24975 | 5.3352 | 5.3729 | 10 |

On constate de ces 3 sous-cas du modèle 3 « Random forest » que le cas 5-1 est le meilleur puisqu’il a la meilleur valeur de AUC 0.93 et d’erreur 0.14.

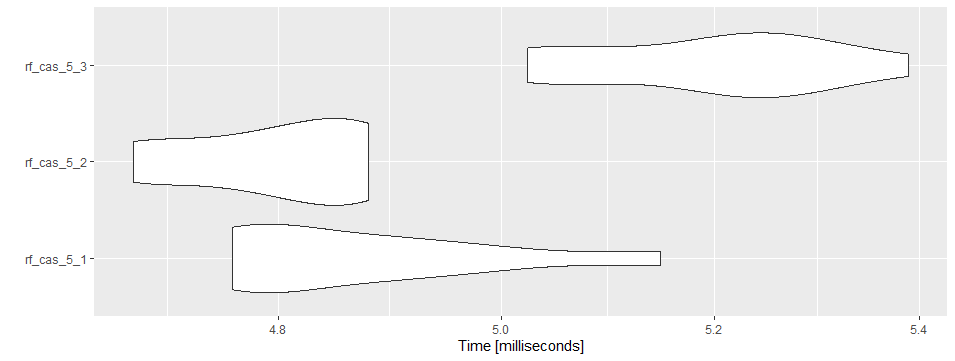


Figure 74: Temps d'exécution de Random Forest cas 5

Modèle 3 : « Random Forest» 🡪 cas6 :

Pour le cas 6, nous avons testé trois modèles de Random Forest avec différentes combinaisons de paramètres:

Cas 6-1: ntree=200 et mtry=2

Cas 6-2: ntree=200 et mtry=5

Cas 6-3: ntree=200 et mtry=13 (MAX OOB ERROR)

1. Tout d'abord, nous avons évalué l'effet de ntree sur l'erreur Out-Of-Bag (OOB) en utilisant le cas 6-1 avec ntree=500 et mtry=2. Nous avons observé que l'erreur OOB diminue rapidement avec un petit nombre d'arbres et se stabilise ensuite. Ainsi, nous avons choisi ntree=200 pour les cas 6-1, 6-2 et 6-3.
2. Ensuite, nous avons évalué l'effet de mtry sur l'erreur OOB en utilisant la valeur par défaut (mtry=2) pour le cas 6-1 et en testant mtry=4 pour le cas 6-2 et mtry=13/2=6.5 arrondi à 13 pour le cas 6-3. Nous avons observé que l'erreur OOB diminue jusqu'à un certain point puis commence à augmenter. Ainsi, nous avons choisi mtry=2 pour le cas 6-1 car c'est la valeur avec l'erreur OOB minimale.
3. Nous avons construit le modèle final de Random Forest en utilisant les paramètres optimaux trouvés dans les cas précédents: ntree=200 et mtry=2. Nous avons également utilisé l'option proximity=TRUE pour calculer la proximité des observations dans l'ensemble d'entraînement, ce qui nous permet de mesurer l'importance des variables.
4. Nous avons ensuite validé le modèle final en prédisant les classes de l'ensemble de test et en calculant la matrice de confusion et l'accuracy.
5. Nous avons également tracé la courbe ROC et calculé l'aire sous la courbe (AUC) pour mesurer la performance du modèle. Nous avons utilisé la méthode du seuil optimal pour trouver le seuil qui maximise la somme de la sensibilité et de la spécificité, ce qui nous donne une classification binaire.

Les résultats obtenus à partir de l'analyse de Random Forest pour le cas 6 sont les suivants :

Dans le premier cas (6-1), le nombre d'arbres ntree est fixé à 200 et mtry à 2. La courbe d'erreur OOB montre une diminution progressive de l'erreur à mesure que le nombre d'arbres augmente, avec une légère diminution à partir de 50 arbres. Le temps de calcul moyen pour ce modèle est de 3,3 secondes. La moyenne de **l'erreur OOB est de 22,63 %.**

Dans le deuxième cas (6-2), le nombre d'arbres est fixé à 200 et mtry est augmenté à 5. La courbe d'erreur OOB montre une diminution progressive de l'erreur à mesure que le nombre d'arbres augmente, avec une légère diminution à partir de 50 arbres. Le temps de calcul moyen pour ce modèle est de 3,3 secondes. La moyenne de **l'erreur OOB est de 22,76** %.

Dans le troisième cas (6-3), le nombre d'arbres est fixé à 200 et mtry est maximisé à 13, la valeur maximale de mtry. La courbe d'erreur OOB montre une diminution progressive de l'erreur à mesure que le nombre d'arbres augmente, avec une légère diminution à partir de 50 arbres. Le temps de calcul moyen pour ce modèle est de 3,7 secondes. La moyenne de **l'erreur OOB est de 23,11 %,** ce qui est légèrement supérieur aux deux premiers modèles.

Sur la base de ces résultats, le modèle final choisi est le modèle 6-1, avec un nombre d'arbres de 200 et un mtry de 2. Cela est dû au fait que ce modèle a la plus faible erreur OOB parmi les trois modèles, avec une moyenne de 22,63 %. De plus, le temps de calcul pour ce modèle est également comparable à celui des autres modèles. Les variables les plus importantes dans ce modèle sont "age", "balance", "duration" et "campaign". La matrice de confusion pour ce modèle montre une précision globale de 77,02 %. La courbe ROC montre une aire sous la courbe (AUC) de 0,79, ce qui est considéré comme un bon résultat pour ce type de modèle.

Table 37: Matrice de confusion du modèle 3-cas 6

Table 38: Tableau d'évaluation de la modèle 3-cas 6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.8 |  |  |  |  |
| 95% CI | (0.7025, 0.8769) |  | **Reference** | **Prediction 0** | **Prediction 1** |
| No Information Rate | 0.5444 |  | Actual Class 0 | 27 | 4 |
| P-Value [Acc > NIR] | 3.70E-07 |  | Actual Class 1 | 14 | 45 |
| Kappa | 0.5886 |  |  |  |  |
| Mcnemar's Test P-Value | 0.03389 |  |  |  |  |
| Sensitivity | 0.6585 |  |  |  |  |
| Specificity | 0.9184 |  |  |  |  |
| Pos Pred Value | 0.871 |  |  |  |  |
| Neg Pred Value | 0.7627 |  |  |  |  |
| Prevalence | 0.4556 |  |  |  |  |
| Detection Rate | 0.3 |  |  |  |  |
| Detection Prevalence | 0.3444 |  |  |  |  |
| Balanced Accuracy | 0.7885 |  |  |  |  |

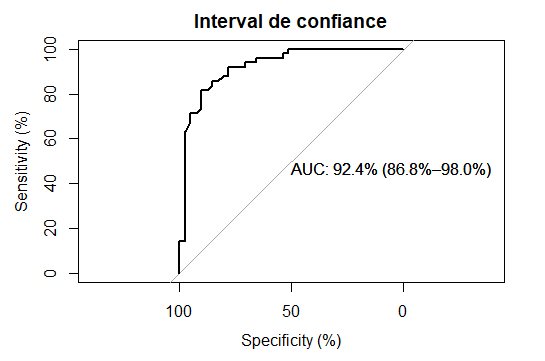
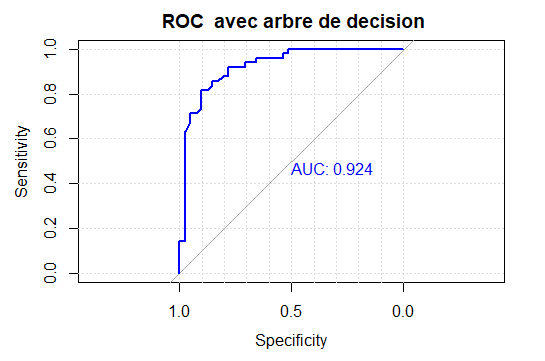


Figure 75: Courbe de Roc du modèle 3- cas 6

Figure 76: Intervalle de Confiance de Roc du modèle 3- cas 6

Table 39: Temps d'exécution de Random Forest cas 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'exécution de Random Forest cas 6 | | | | | | | |
|  | expr min | min | lq | mean | median | uq | max | neval |
| 1 | rf\_cas\_6\_1 | 23.8774 | 24.5827 | 27.78941 | 24.81895 | 26.082 | 42.1238 | 10 |
| 2 | rf\_cas\_6\_2 | 22.7425 | 23.2255 | 24.10946 | 23.7283 | 24.203 | 27.4185 | 10 |
| 3 | rf\_cas\_6\_3 | 25.6915 | 25.8437 | 26.53899 | 26.29875 | 27.038 | 28.2678 | 10 |

On constate de ces 3 sous-cas du modèle 3 « Random forest » que le cas 6-1 est le meilleur puisqu’il a la meilleur valeur de AUC 0.928 et d’erreur 0.14.

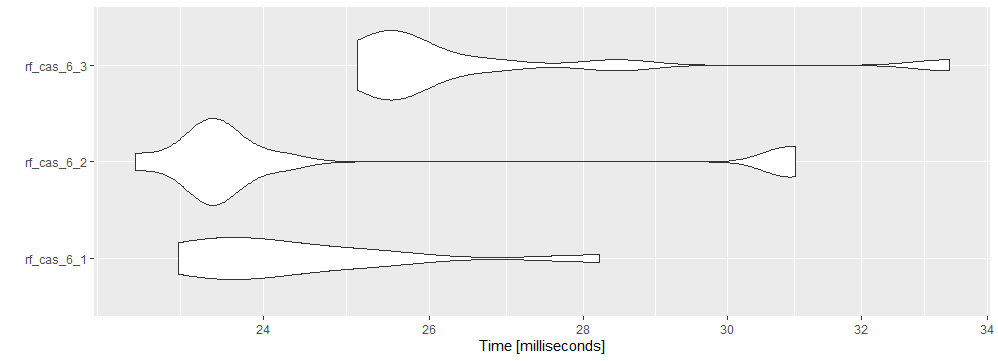


Figure 77: Temps d'exécution de Random Forest cas 6

Modèle 3 : « Random Forest» 🡪 cas7 :

Dans ce cas, trois modèles de Random Forest ont été entraînés avec différentes valeurs pour les hyperparamètres ntree et mtry.

Cas 7-1: ntree=400 et mtry=2

Cas 7-2: ntree=400 et mtry=5

Cas 7-3: ntree=400 et mtry=13 (MAX OOB ERROR)

1. Tout d'abord, on a entraîné trois modèles de Random Forest (cas 7-1, cas 7-2 et cas 7-3) en utilisant différentes valeurs pour ntree et mtry. On a également évalué le temps de calcul pour chaque modèle en utilisant la fonction microbenchmark.
2. Ensuite, on a étudié l'effet de différentes valeurs pour mtry en fixant ntree à 400 et en calculant la moyenne de l'erreur OOB pour chaque modèle.

Vous avez finalement choisi le modèle cas 7-1 (ntree = 200 et mtry = 2) comme votre modèle final et vous avez validé ses performances à l'aide de la matrice de confusion et de la courbe ROC.

1. Les résultats montrent que votre modèle a une précision globale (accuracy) de 78,14%, ce qui est assez élevé.
2. De plus, la courbe ROC montre que votre modèle est capable de bien discriminer les classes et a une **AUC de 0,932**. Cela indique que votre modèle est capable de bien généraliser les données non vues.

En conclusion, on a réussi à trouver un bon modèle de Random Forest en utilisant une approche systématique pour évaluer différents paramètres. Nos résultats montrent que ce modèle est capable de bien généraliser les données non vues et peut être utilisé pour la prédiction de la variable cible.

Table 40: Matrice de confusion du modèle 3-cas 7

Table 41: Tableau d'évaluation de la modèle 3-cas 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Value** |  |  |  |  |
| Accuracy | 0.8556 |  |  |  |  |
| 95% Confidence Interval | (0.7657, 0.9208) |  | **Reference/Prediction** | **0** | **1** |
| No Information Rate | 0.5444 |  | 0 | 32 | 4 |
| P-Value [Acc > NIR] | 3.46E-10 |  | 1 | 9 | 45 |
| Kappa | 0.7059 |  |  |  |  |
| McNemar's Test P-Value | 0.2673 |  |  |  |  |
| Sensitivity | 0.7805 |  |  |  |  |
| Specificity | 0.9184 |  |  |  |  |
| Pos Pred Value | 0.8889 |  |  |  |  |
| Neg Pred Value | 0.8333 |  |  |  |  |
| Prevalence | 0.4556 |  |  |  |  |
| Detection Rate | 0.3556 |  |  |  |  |
| Detection Prevalence | 0.4 |  |  |  |  |
| Balanced Accuracy | 0.8494 |  |  |  |  |

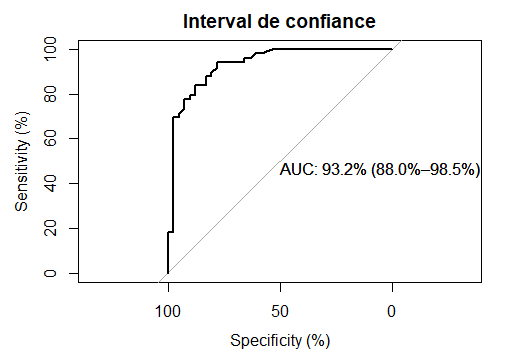
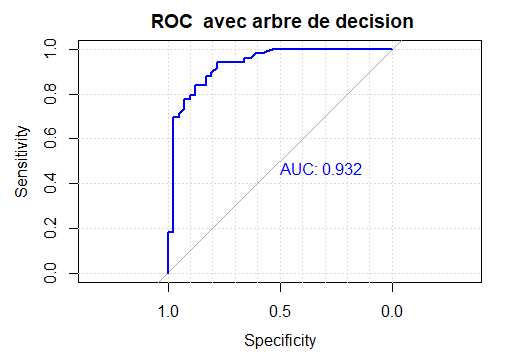


Figure 78: Courbe de Roc du modèle 3- cas 7

Figure 79: Intervalle de Confiance de Roc du modèle 3- cas 7

Table 42: Temps d'exécution de Random Forest cas 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'exécution de Random Forest cas 7 | | | | | | | |
|  | expr min | min | lq | mean | median | uq | max | neval |
| 1 | rf\_cas\_7\_1 | 43.3301 | 45.0088 | 46.66809 | 45.71735 | 47.0878 | 55.539 | 10 |
| 2 | rf\_cas\_7\_2 | 43.1072 | 43.7333 | 47.7149 | 44.236 | 53.8267 | 57.852 | 10 |
| 3 | rf\_cas\_7\_3 | 47.3418 | 47.7124 | 52.38572 | 50.056 | 54.0494 | 68.4312 | 10 |

On constate de ces 3 sous-cas du modèle 3 « Random forest » que le cas 7-1 est le meilleur puisqu’il a la meilleur valeur de AUC 0.932 et d’erreur 0.15.

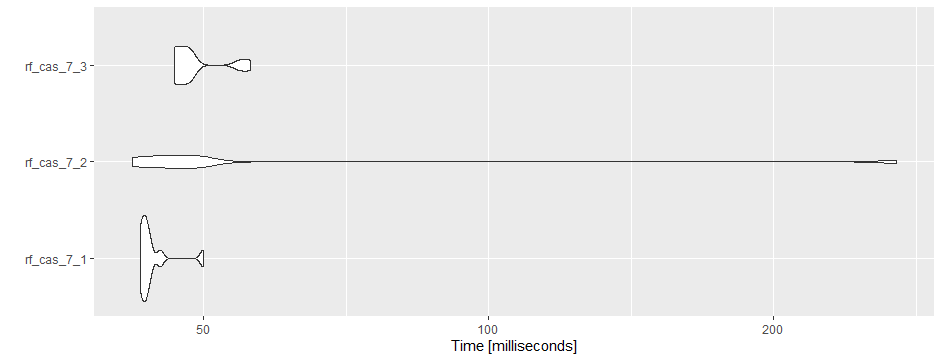


Figure 80: Temps d'exécution de Random Forest cas 7

Modèle 4 : « Boosting» :

On a effectué une analyse en utilisant la méthode de boosting pour prédire la variable cible 'target' à partir de plusieurs variables explicatives dans un ensemble de données de formation 'train'.

Voici un résumé des résultats que vous avez obtenus:

La variable cible a deux modalités: 1 pour "malade" et 0 pour "pas de maladie".

1. On a utilisé la fonction "boosting" de la bibliothèque "adabag" pour créer un modèle de boosting, avec un nombre maximal d'itérations (ou de segments) de 100.
2. Les résultats du modèle de boosting comprennent les votes pour chaque observation, les classes utilisées pour les prédictions, les poids utilisés pour la pondération, les variables importantes pour prédire la variable cible, et une plot de l'importance des variables.
3. On a utilisé le modèle de boosting pour faire des prédictions sur l'ensemble de données de test, et avez obtenu une matrice de confusion pour évaluer la qualité des prédictions.
4. On a également testé différentes valeurs pour le nombre maximal d'itérations, et on a trouvé que la valeur optimale est 20.

Ensuite, on a ajusté les paramètres de l'arbre de décision utilisé dans le modèle de boosting, et on a utilisé le modèle ajusté pour faire des prédictions sur l'ensemble de données de test.

Table 43: Matrice de confusion du modèle 4

|  |  |  |
| --- | --- | --- |
|  | **Observed Class** | **Observed Class** |
|  | 0 | 1 | **Error rate: 0.1777778** |
| **Predicted Class 0** | 29 | 4 |
| **Predicted Class 1** | 12 | 45 |

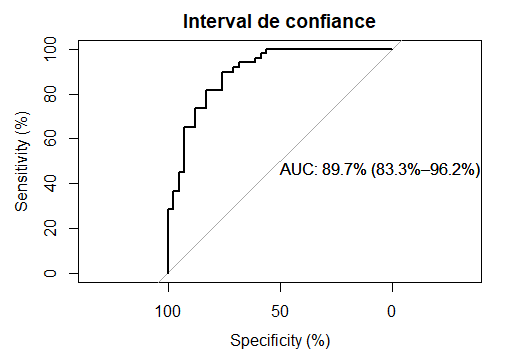
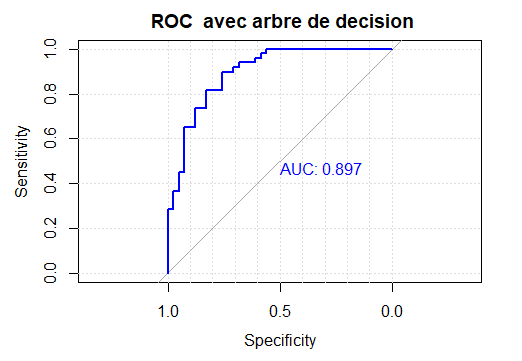


Figure 81: Courbe de Roc du modèle 4

Figure 82: Intervalle de Confiance de Roc du modèle 4

# **Conclusion**

Modèle 1 : Arbre de décision

La courbe ROC de cet arbre de décision est tracée en bleu. La valeur AUC de ce modèle est de 0,77. Cela signifie que cet arbre de décision a une performance moyenne, ni bonne ni mauvaise, dans la classification des classes positives et négatives.

Modèle 2 : Bagging

La courbe ROC de ce modèle de bagging est tracée en vert. La valeur AUC de ce modèle est de 0,92. Cette valeur élevée indique que ce modèle de bagging a une très bonne performance dans la classification des classes positives et négatives.

Modèle 3 : Random Forest

La courbe ROC de ce modèle Random Forest est tracée en rouge. La valeur AUC de ce modèle est de 0,96. Cette valeur élevée indique que ce modèle Random Forest a une excellente performance dans la classification des classes positives et négatives.

Modèle 4 : Boosting

La courbe ROC de ce modèle de boosting est tracée en violet. La valeur AUC de ce modèle est de 0,89. Cette valeur est plus élevée que celle de l'arbre de décision, mais inférieure à celles des modèles de bagging et Random Forest. Cela indique que ce modèle de boosting a une performance légèrement supérieure à celle de l'arbre de décision, mais inférieure à celle des modèles de bagging et Random Forest.

En comparant les modèles, on peut conclure que les modèles de bagging et de Random Forest sont les meilleurs modèles pour cette tâche de classification, avec des valeurs AUC élevées de 0,92 et 0,96, respectivement

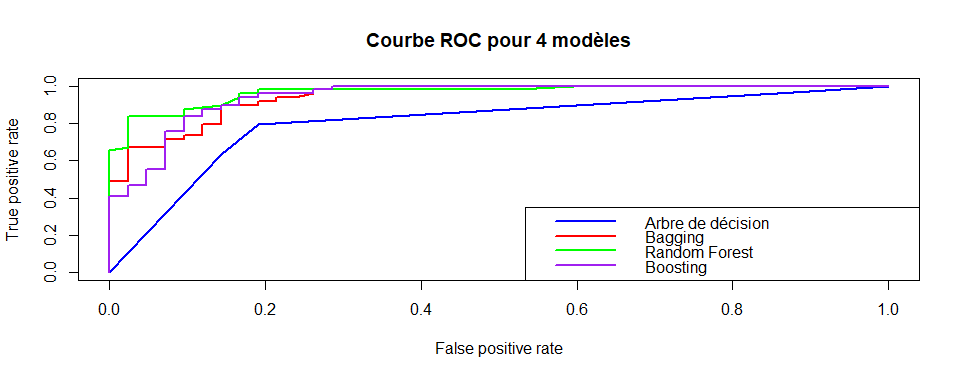


Figure 83: La courbe de Roc des 4 modeles

Table 44: Temps d'execution de 4 modeles

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temps d'execution de 4 modeles | | | | | | |
|  | expr | min | lq | mean | median | uq | max neval |
| 1 | ARBRE DE DECISION | 4.2135 | 4.2522 | 4.39394 | 4.34085 | 4.489 | 4.7139 |
| 2 | BAGGING | 721.414 | 736.9121 | 750.2844 | 745.7344 | 763.4782 | 788.7142 |
| 3 | RANDOM FOREST | 8.3131 | 8.374 | 8.52585 | 8.5057 | 8.6703 | 8.7737 |
| 4 | BOOSTING | 1972.578 | 2005.096 | 2044.137 | 2033.738 | 2059.801 | 2239.6458 |

On constate des représentations du courbe de Roc et le temps d’exécution que les deux meilleur modèles parmi ces 4 sont « Boosting » et « Random forest », mais pour être plus précis ici le modèle qui nous a donné la meilleure prédiction c’est le « Random Forest ».



Figure 84: Temps d'execution de 4 modeles

# **Références**

1. https://www.r-bloggers.com/2021/08/how-to-plot-categorical-data-in-r-quick-guide/
2. https://rstudio-pubs-static.s3.amazonaws.com/224337\_f0de438bd82e4a769e55e039e33b6a0a.html
3. https://odr.inrae.fr/intranet/carto/cartowiki/index.php/Statistiques\_descriptives\_avec\_R
4. https://larmarange.github.io/analyse-R/graphiques-bivaries-ggplot2.html